

Trabajo Fin de Carrera

Ingeniería de Telecomunicación

Módulo de Inventario de GTI

Autor: Álvaro Marín Sánchez

Tutor: Antonio Luis Delgado González

Dpto. Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



Proyecto Fin de Carrera
Ingeniería de Telecomunicación

Módulo de Inventario de GTI

Autor:
Álvaro Marín Sánchez

Tutor:
Antonio Luis Delgado González
Profesor asociado

Dpto. Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2021



Proyecto Fin de Carrera: Módulo de Inventario de GTI

Autor: Álvaro Marín Sánchez

Tutor. Antonio Luis Delgado González.

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal



A mi familia

A mis profesores

A mis amigos y compañeros



Agradecimientos

Hay muchas personas que han contribuido a que yo me encuentre en este momento, desde mis profesores de colegio, instituto y universidad, hasta mi familia. Sin embargo, me gustaría agradecer personalmente a personas que sin duda me han sabido guiar en la dirección correcta hasta llegar a convertirme en quién soy ahora.

En primer lugar, no puedo no mencionar a mis padres y mis hermanos que han estado desde el principio, en los momentos difíciles de la carrera y en los momentos de celebración cuando se requería. En particular, me gustaría dar las gracias a mi hermana que ha estado siempre en los momentos difíciles para alegrarme el día con cualquier plan.

En segundo lugar, me gustaría agradecer a todas las personas que he conocido en la carrera que me han motivado y ayudado a seguir adelante. En especial, Pablo y José Antonio que han hecho que estos últimos años tan complicados fueran mucho más llevaderos.

Por otro lado, mencionar a Paula que siempre ha estado conmigo desde que tenía tres años, y, sin duda, ha sido responsable en parte de ser quien soy hoy en día.

Finalmente, me gustaría agradecer a mi tutor del proyecto, Antonio Luis Delgado, por toda la ayuda que me ha ofrecido a la hora de realizar este proyecto y por su paciencia en momentos que no se esperaban.

Álvaro Marín Sánchez
Sevilla, 2021



Resumen

Muchas empresas tienen la necesidad de controlar sus activos físicos y digitales. Gestionar de forma correcta las existencias de estos puede conllevar un importante ahorro económico ya que permite a la organización anticiparse a sus necesidades sin tener que realizar una elevada inversión en el stock.

Existen multitud de aplicaciones que dan respuesta a este problema. Sin embargo, son escasas las que recogen todas las funcionalidades que una empresa necesitaría para la total administración de sus activos (ordenadores, licencias software, etc).

En este trabajo de fin de grado se ha diseñado una aplicación web modular llamada GTI (Gestión TIC) que permite la gestión TIC completa de una organización. Dicha aplicación se compone de los siguientes módulos: módulo de gestión de proyectos, módulo de gestión de servicios, módulo de gestión de activos, módulo de gestión de proveedores y módulo común, siendo los tres últimos los desarrollados en este proyecto.

El módulo de gestión de activos da respuesta a la problemática del inventario TIC, ofreciendo la total gestión de los activos durante su ciclo de vida: su inclusión en el inventario, posibles transferencias y finalmente su baja. El módulo de gestión de adquisiciones y proveedores está directamente ligado al módulo de inventario ya que permite realizar transacciones de compra y venta de modelos. Además, se completa el módulo común, base para la construcción del proyecto GTI.

Finalmente, referente al desarrollo del proyecto, se ha optado por la técnica llamada Desarrollo de Interfaces de Usuario Basado en Modelos (MBUID), apoyada por el entorno WAINE.



Abstract

Nowadays, companies have found the necessity of having control of their physical and digital assets since this not only means the classification and storage of their assets, but also large amounts of economic savings by being able to anticipate their needs without having to invest in stock.

There are multiple applications that give an answer to this problem. However, there are few that include all the functionalities that a company could expect for the total administration of their assets.

Therefore, this Final Degree Project is part of a bigger web application called GTI that provides a clear solution to the management of ICT inventory, projects and processes and services within an organization, providing functionalities that improve the main options in the market. This application is currently built by five modules: processes and services management module, projects management module, assets management module, providers and purchases management module and common module, the last three being the topic of this document.

Firstly, asset management module responds to the ICT inventory problem. In addition to that, this project involves two additional modules called common module and providers and purchases module that complete aspects of the inventory module to offer a total management of assets in their life cycle, throughout their acquisition, maintenance, possible transfers and, finally, their withdrawal.

As regards the development of this project, the technique called Module-Based User Interface Development (MBUIDE) has been chosen, which is supported in the WAINE environment in its application development. The use of this environment will be valued as a conclusion alongside the improvements made and possible lines of progress.



Índice

Agradecimientos	7
Resumen.....	9
Abstract	11
1 Introducción	29
2 Análisis de los módulos de activos, adquisiciones y común de GTI	32
2.1 Introducción	32
2.1.1 El módulo de gestión de activos	32
2.1.2 Objetivos del proyecto	34
2.2 Situación actual.....	34
2.2.1 Modelo físico de datos	34
2.2.2 Interfaces de usuario desarrolladas	35
2.2.2.1 Sedes y Localizaciones	35
2.2.2.2 Unidades Organizativas	36
2.2.2.3 Unidades y tipos de unidades	37
2.2.2.4 Marcas, modelos y atributos	38
2.2.2.5 Tipos de elementos inventariables, atributos de modelo y atributos de elemento	39
2.3 Catálogo de requisitos.....	40
2.3.1 Catálogo de requisitos modulo am.....	40
2.3.1.1 Funcionalidades	41
2.3.1.2 Usuarios	44
2.3.2 Catálogo de requisitos módulo pp.....	45
2.3.2.1 Funcionales	45
2.3.2.2 Usuario.....	46
2.4 Modelo conceptual de datos.....	47
2.4.1 Entidades.....	47
2.4.1.1 Módulo común.....	47
2.4.1.2 Módulo de inventario.....	48
2.4.1.3 Módulo de adquisiciones	49
2.4.1.4 Módulo de entradas y salidas del inventario de activos	50
2.4.2 Relaciones	51
2.5 Interfaces de usuario y acciones.....	52
2.5.1 Usuario administrador del módulo común.....	54
2.5.1.1 Gestión de ámbitos	54
2.5.1.2 Gestión de unidades organizativas.....	55
2.5.1.3 Gestión de localizaciones	55
2.5.2 Usuario administrador del módulo de inventario.....	56
2.5.2.1 Administración de permisos	56

2.5.2.2	Tipos de entrada.....	57
2.5.2.3	Tipos de salida	57
2.5.2.4	Estados de los recursos	58
2.5.2.5	Tipos de unidades y unidades	58
2.5.2.6	Tipos de elemento y atributos	59
2.5.3	Gestor de adquisiciones	59
2.5.3.1	Gestión de proveedores.....	59
2.5.3.2	Gestión de ofertas	60
2.5.3.3	Gestión de adquisiciones	61
2.5.3.4	Marcas, modelos y atributos	61
2.5.4	Gestor de inventario	62
2.5.4.1	Gestión de entradas.....	62
2.5.4.2	Gestión de salidas	64
2.5.4.3	Gestión de inventario	65
2.5.4.4	Edición de activos.....	65
2.5.4.4.1	Pestaña recursos hijo	67
2.5.4.4.2	Pestaña Histórico	67
2.5.4.4.3	Pestaña Traza	68
2.5.4.4.4	Pestaña Adjuntos	68
2.5.4.5	Gestión de activos.....	68
2.5.5	Técnico Avanzado	69
2.5.5.1	Gestión de inventario reducida	69
2.5.5.2	Edición de activos reducida	70
2.5.6	Técnico.....	70
2.5.7	Usuario regular.....	71
2.5.7.1	Consulta de recursos asignados	71
2.5.8	Usuario sin acceso.....	72
2.5.8.1	Visualización de recurso	72
2.5.8.2	Visualización de adquisición	72
2.5.8.3	Visualización de entrada	73
2.5.8.4	Visualización de salida	73
2.6	Anexos	73
2.6.1	Normas de codificación SQL.....	74
2.6.2	Normas de codificación ASL	74
2.6.3	Código SQL de la base de datos del prototipo inicial.....	75
2.6.4	Código ASL del prototipo inicial	78
2.6.5	Winter.....	86
2.6.5.1	Introducción.....	86
2.6.5.2	El problema original	87



2.6.5.3	La solución propuesta	87
2.6.5.3.1	El proceso de desarrollo	88
2.6.5.3.2	La aplicación Winter	89
2.6.5.3.2.1	Tabla Rol/Funcionalidad	89
2.6.5.3.2.2	Modelo de dominio	90
2.6.5.3.2.3	Especificación de las unidades de interacción	90
2.6.5.3.2.4	Asignación de paneles a un usuario	91
2.6.5.3.2.5	Visualización de todos los paneles	91
2.6.6	Módulo fdescr	93
2.6.6.1	Funcionalidades	93
2.6.6.2	Ejemplos	93
2.6.7	Módulo ffkdescr	93
2.6.7.1	Funcionalidades	93
2.6.7.2	Ejemplos	94
2.6.8	Paquete percon	94
2.6.9	Funcionalidades	95
2.6.10	Paquete attach	96
2.6.10.1	Funcionalidades	96
2.6.10.2	Ejemplos	97
3	Diseño de los módulos de activos, adquisiciones y común de GTI	102
3.1	Introducción	102
3.2	Arquitectura del sistema	102
3.3	Modelo físico de datos	103
3.3.1	Modelo físico de datos modulo am	103
3.3.2	Modelo físico de datos módulo gti	108
3.3.3	Modelo físico de datos módulo pp	110
3.4	Arquitectura de módulos del sistema	112
3.4.1	Arquitectura de módulos del sistema módulo am	112
3.4.1.1	Eventos	112
3.4.1.1.1	Eventos implementados en ASL	112
3.4.1.1.1.1	Evento “Eliminación de un tipo de entrada predefinida”	112
3.4.1.1.1.2	Evento “Eliminación de un tipo de salida predefinida”	113
3.4.1.1.1.3	Evento “Eliminación de un estado de los recursos”	113
3.4.1.1.1.4	Evento “Asignación de permisos de gestor de adquisiciones”	113
3.4.1.1.1.5	Evento “Asignación de permisos de gestor de inventario”	113
3.4.1.1.1.6	Evento “Asignación de permisos de técnico avanzado”	113
3.4.1.1.1.7	Evento “Asignación de permisos de técnico”	113
3.4.1.1.1.8	Evento “Inserción de un nuevo activo en una entrada”	113

3.4.1.1.1.9	Evento “Inserción de un activo en una salida”	113
3.4.1.1.1.10	Evento “Actualización de un activo”	114
3.4.1.1.1.11	Evento “Actualización de un número de serie en un activo”	114
3.4.1.1.2	Eventos implementados en SQL	114
3.4.1.1.2.1	Evento TRI_ModelTypeAttr_after	114
3.4.1.1.2.2	Evento TRI_AssetTypeAttr_after.....	114
3.4.1.1.2.3	Evento TRI_Model_after	114
3.4.1.1.2.4	Evento TRI_Asset_after	114
3.4.1.1.2.5	Evento TRI_Input_after	114
3.4.1.1.2.6	Evento TRI_Asset_update.....	114
3.4.1.1.2.7	Evento TRI_Asset_history	114
3.4.1.1.2.8	Evento TRI_Purchase_after.....	114
3.4.1.2	Funciones.....	115
3.4.1.2.1	Función PROC_ModelTypeAttr_after().....	115
3.4.1.2.2	Función PROC_AssetTypeAttr_after()	115
3.4.1.2.3	Función PROC_Model_after()	115
3.4.1.2.4	Función PROC_Asset_after().....	115
3.4.1.2.5	Función PROC_Input_after()	115
3.4.1.2.6	Función FUNC_Input_aux(int,int,int,int,varchar,varchar).....	115
3.4.1.2.7	Función FUNC_Asset_aux(int,int,varchar)	116
3.4.1.2.8	Función PROC_Asset_update()	116
3.4.1.2.9	Función PROC_Asset_history().....	116
3.4.1.2.10	Función PROC_Purchase_after()	117
3.4.2	Arquitectura de módulos del sistema módulo pp.....	117
3.4.2.1	Eventos	117
3.4.2.1.1	Eventos implementados en ASL.....	117
3.4.2.1.1.1	Evento “Inserción de un nuevo contacto para un proveedor”	117
3.4.2.1.1.2	Evento “Inserción de un nuevo modelo en una oferta”	117
3.4.2.1.1.3	Evento “Inserción de un nuevo modelo en una compra”	117
3.4.2.1.1.4	Evento “Cálculo de la imputación total de una oferta”	117
3.4.2.1.1.5	Evento “Cálculo de la imputación total de una compra”	117
3.5	Diseño de las interfaces de usuario	118
3.5.1	Diseño de las interfaces de usuario módulo am.....	118
3.5.1.1	Administrador del módulo de inventario	119
3.5.1.1.1	Administración de permisos – <i>gti.am.cpermmanag1</i>	119
3.5.1.1.2	Tipos de entrada – <i>gti.am.cinputtype</i>	119
3.5.1.1.3	Tipos de salida – <i>gti.am.coutputtype</i>	119
3.5.1.1.4	Estados de los recursos – <i>gti.am.cstatus</i>	120
3.5.1.1.5	Tipos de unidades y unidades – <i>gti.am.cunit</i>	120



3.5.1.1.6	Tipos de elementos y atributos – <i>gti.am.casstype1</i>	120
3.5.1.1.7	Consulta de recursos asignados – <i>cassignresourcequery1</i>	121
3.5.1.2	Gestor de inventario.....	122
3.5.1.2.1	Marcas, modelos y atributos – <i>gti.am.cmarkmodell</i>	122
3.5.1.2.2	Gestión de entradas – <i>gti.am.cinputmanag1</i>	122
3.5.1.2.3	Gestión de salidas – <i>gti.am.coutputmanag1</i>	123
3.5.1.2.4	Gestión de inventario – <i>gti.am.cassetmanag</i>	123
3.5.1.2.5	Gestión de inventario por tipo recurso – <i>gti.am.cassetmanagtype</i>	124
3.5.1.2.6	Edición de activos – <i>gti.am.cassetedit1</i>	124
3.5.1.2.7	Gestión de activos – <i>gti.am.cassetmanag1</i>	125
3.5.1.2.8	Estadísticas de tipos de activos por sede - <i>gti.am.cgraphtype</i>	125
3.5.1.2.9	Estadísticas de estados y marcas de activos por sede y tipo - <i>gti.am.cgraphstatusmark1</i>	126
3.5.1.2.10	Consulta de recursos asignados – <i>cassignresourcequery1</i>	126
3.5.1.3	Técnico avanzado	126
3.5.1.3.1	Gestión de entradas – <i>gti.am.cinputmanagat1</i>	126
3.5.1.3.2	Gestión de salidas – <i>gti.am.coutputmanagat1</i>	127
3.5.1.3.3	Gestión de inventario reducida – <i>gti.am.cassetmanagredat1</i>	127
3.5.1.3.4	Edición de activos reducida – <i>casseteditredat1</i>	128
3.5.1.3.5	Consulta de recursos asignados – <i>cassignresourcequery1</i>	128
3.5.1.4	Técnico	128
3.5.1.4.1	Gestión de inventario reducida – <i>gti.am.cassetmanagredtec</i>	128
3.5.1.4.2	Edición de activos reducida – <i>gti.am.casseteditredtec1</i>	129
3.5.1.4.3	Consulta de recursos asignados – <i>gti.am.cassignresourcequery1</i>	129
3.5.1.5	Usuario regular	129
3.5.1.5.1	Consulta de recursos asignados – <i>gti.am.cassignresourcequery1</i>	129
3.5.1.6	Usuario sin acceso	130
3.5.1.6.1	Visualización de recurso – <i>gti.am.cresourceview</i>	130
3.5.1.6.2	Visualización de entrada – <i>gti.am.cinputview</i>	130
3.5.1.6.3	Visualización de salida – <i>gti.am.coutputview</i>	131
3.5.2	Diseño de las interfaces de usuario módulo gti.....	131
3.5.2.1	Administrador del módulo común	131
3.5.2.1.1	Gestión de ámbitos – <i>gti.cscope</i>	132
3.5.2.1.2	Gestión de unidades organizativas – <i>gti.coull</i>	132
3.5.2.1.3	Gestión de localizaciones – <i>gti.csite1</i>	132
3.5.2.1.4	Datos personales – <i>percon.struct.edit</i>	133
3.5.2.1.5	Datos personales usuario – <i>upercon.struct.edit</i>	134
3.5.3	Diseño de las interfaces de usuario módulo pp.....	134

3.5.3.1	Gestor de adquisiciones	134
3.5.3.1.1	Gestión de proveedores – <i>gti.am.cprovmanag1</i>	135
3.5.3.1.2	Gestión de ofertas – <i>gti.am.coffermanag1</i>	135
3.5.3.1.3	Gestión de adquisiciones – <i>gti.am.cpurchaseview</i>	136
3.5.3.1.4	Marcas, modelos y atributos – <i>gti.am.cmarkmodell</i>	137
3.5.3.1.5	Consulta de recursos asignados – <i>gti.am.cassignresourcequery1</i>	138
3.5.3.2	Usuario sin acceso	138
3.5.3.2.1	Visualización de adquisición – <i>gti.pp.cpurchaseview</i>	138
3.6	Casos de uso reales	139
3.6.1	Casos de uso reales módulo am	139
3.6.1.1	Asignación de permisos a un usuario	139
3.6.1.2	Gestión de los activos	140
3.6.1.3	Tramitación de una entrada de activos para una adquisición	142
3.6.1.4	Consulta de los recursos asignados.....	143
3.6.1.5	Visualización del recurso	144
3.6.2	Casos de uso reales módulo pp	145
3.6.2.1	Tramitación de una oferta dada una solicitud	145
3.6.2.2	Visualización de adquisición	146
3.7	Especificaciones de Construcción.....	147
3.8	Carga inicial de datos	149
3.8.1	Carga inicial de datos módulo am.....	149
3.8.2	Carga inicial de datos módulo gti.....	149
3.8.3	Carga inicial de datos módulo pp.....	150
3.9	Plan de pruebas técnico	150
3.9.1	Plan de pruebas técnico módulo am.....	150
3.9.1.1	Tablas de pruebas.....	150
3.9.1.1.1	Tabla de pruebas administrador del módulo de inventario.....	150
3.9.1.1.2	Tabla de pruebas gestor de inventario	152
3.9.1.1.3	Tabla de pruebas técnico avanzado	156
3.9.1.1.4	Tabla de pruebas técnico	159
3.9.1.1.5	Tabla de pruebas usuario regular	160
3.9.1.1.6	Tabla de pruebas usuario sin acceso	160
3.9.1.2	Matriz de trazabilidad	161
3.9.1.2.1	Matriz de trazabilidad administrador del módulo	161
3.9.1.2.2	Matriz de trazabilidad gestor de inventario	162
3.9.1.2.3	Matriz de trazabilidad técnico avanzado	162
3.9.1.2.4	Matriz de trazabilidad técnico	163
3.9.1.2.5	Matriz de trazabilidad usuario regular.....	163
3.9.1.2.6	Matriz de trazabilidad usuario sin acceso.....	163



3.9.2	Plan de pruebas técnico módulo gti	164
3.9.2.1	Tablas de pruebas	164
3.9.2.1.1	Tabla de pruebas administrador del módulo común.....	164
3.9.3	Plan de pruebas técnico módulo pp.....	165
3.9.3.1	Tablas de pruebas.....	165
3.9.3.1.1	Tabla de pruebas gestor de adquisiciones.....	165
3.9.3.1.2	Tabla de pruebas usuario sin acceso	167
3.9.3.2	Matriz de trazabilidad	167
3.9.3.2.1	Matriz de trazabilidad gestor de adquisiciones	167
3.9.3.2.2	Matriz de trazabilidad usuario regular.....	168
3.10	Requisitos de implantación	168
4	Construcción de los módulos de activos, adquisiciones y común de GTI.....	169
4.1	Introducción	169
4.2	Construcción modelo físico de datos	169
4.2.1	Construcción del modelo físico de datos módulo am	169
4.2.2	Construcción del modelo físico de datos módulo gti.....	182
4.2.3	Construcción del modelo físico de datos módulo pp	184
4.3	Construcción de los módulos del sistema	186
4.3.1	Construcción de los módulos del sistema módulo am	186
4.3.1.1	Construcción de eventos	186
4.3.1.1.1	Construcción de eventos en SQL	186
4.3.1.1.2	Construcción de eventos en ASL.....	187
4.3.1.2	Construcción de funciones.....	190
4.3.2	Construcción de los módulos del sistema módulo pp.....	195
4.3.2.1	Construcción de eventos	195
4.3.2.1.1	Construcción de eventos en ASL.....	195
4.4	Construcción de las interfaces de usuario	196
4.4.1	Construcción de las interfaces de usuario módulo am.....	196
4.4.1.1	Interfaces comunes	196
4.4.1.1.1	Consulta de recursos asignados.....	196
4.4.1.2	Administrador del módulo de inventario	200
4.4.1.2.1	Administración de permisos.....	202
4.4.1.2.2	Tipos de entrada	203
4.4.1.2.3	Tipos de salida.....	204
4.4.1.2.4	Estados de los recursos.....	205
4.4.1.2.5	Tipos de unidades y unidades.....	205
4.4.1.2.6	Tipos de elementos y atributos	206
4.4.1.3	Gestor de inventario.....	207

4.4.1.3.1	Marcas, modelos y atributos.....	209
4.4.1.3.2	Gestión de entradas	210
4.4.1.3.3	Gestión de salidas	213
4.4.1.3.4	Gestión de inventario.....	216
4.4.1.3.5	Gestión de inventario por tipo de recurso	218
4.4.1.3.6	Gestión de activos	219
4.4.1.3.7	Edición de activos	224
4.4.1.3.8	Estadísticas de tipos de activos por sede	227
4.4.1.3.9	Estadísticas de estados y marcas de activos por sede y tipo	228
4.4.1.4	Técnico avanzado	229
4.4.1.4.1	Gestión de entradas	230
4.4.1.4.2	Gestión de salidas.....	232
4.4.1.4.3	Gestión de inventario reducida.....	233
4.4.1.4.4	Edición de activos reducida.....	235
4.4.1.5	Técnico	238
4.4.1.5.1	Gestión de inventario reducida.....	238
4.4.1.5.2	Edición de activos reducida.....	238
4.4.1.6	Usuario regular	239
4.4.1.7	Usuario sin acceso	239
4.4.1.7.1	Visualización de recurso.....	240
4.4.1.7.2	Visualización de entrada.....	241
4.4.1.7.3	Visualización de salida	242
4.4.2	Construcción de las interfaces de usuario módulo gti.....	243
4.4.2.1	Administrador del módulo común	243
4.4.2.1.1	Gestión de ámbitos	244
4.4.2.1.2	Gestión de unidades organizativas	244
4.4.2.1.3	Gestión de localizaciones	246
4.4.2.1.4	Datos personales.....	248
4.4.2.1.5	Datos personales de usuario	249
4.4.3	Construcción de las interfaces de usuario módulo pp.....	250
4.4.3.1	Gestor de adquisiciones	250
4.4.3.1.1	Gestión de proveedores	251
4.4.3.1.2	Gestión de ofertas.....	253
4.4.3.1.3	Gestión de adquisiciones	256
4.4.3.1.4	Marcas, modelos y atributos.....	258
4.4.3.1.5	Consulta de recursos asignados.....	258
4.4.3.2	Usuario sin acceso	258
4.4.3.2.1	Visualización de adquisición	259
4.5	Manuales de usuario.....	259



4.5.1	Manuales de usuario de GTI.AM.....	259
4.5.1.1	Introducción.....	259
4.5.1.2	Roles	259
4.5.1.3	Manual de usuario del administrador del módulo GTI.AM	260
4.5.1.3.1	Funcionalidades.....	260
4.5.1.3.2	Permisos	260
4.5.1.3.3	Tipos de entradas	263
4.5.1.3.4	Tipos de salidas	263
4.5.1.3.5	Estados de los recursos.....	264
4.5.1.3.6	Tipos de unidades y unidades.....	264
4.5.1.3.7	Tipos de elementos y atributos	265
4.5.1.3.8	Recursos asignados.....	266
4.5.1.4	Manual de usuario del gestor de inventario GTI.AM.....	269
4.5.1.4.1	Funcionalidades.....	269
4.5.1.4.2	Marcas, modelos y atributos.....	269
4.5.1.4.3	Gestión de entradas	270
4.5.1.4.4	Gestión de salidas	274
4.5.1.4.5	Gestión de inventario.....	277
4.5.1.4.6	Gestión de inventario por tipo de recurso	278
4.5.1.4.7	Gestión de activos	279
4.5.1.4.8	Edición de activos	282
4.5.1.4.9	Tipos de activos por sede	282
4.5.1.4.10	Estados y marcas de activos por sede y tipo	283
4.5.1.4.11	Recursos asignados.....	284
4.5.1.5	Manual de usuario del técnico avanzado GTI.AM.....	284
4.5.1.5.1	Funcionalidades.....	284
4.5.1.5.2	Gestión de entradas	285
4.5.1.5.3	Gestión de salidas.....	285
4.5.1.5.4	Gestión de inventario reducida.....	285
4.5.1.5.5	Edición de activos reducida.....	287
4.5.1.5.6	Recursos asignados.....	288
4.5.1.6	Manual de usuario del técnico GTI.AM.....	288
4.5.1.6.1	Funcionalidades.....	288
4.5.1.6.2	Gestión de inventario reducida.....	288
4.5.1.6.3	Edición de activos reducida.....	288
4.5.1.6.4	Recursos asignados.....	289
4.5.1.7	Manual de usuario del usuario regular GTI.AM	289
4.5.1.7.1	Funcionalidades.....	289

4.5.1.7.2	Recursos asignados.....	289
4.5.1.8	Manual de usuario del usuario sin acceso GTI.AM	289
4.5.1.8.1	Funcionalidades.....	289
4.5.1.8.2	Visualización de recurso.....	289
4.5.1.8.3	Visualización de entrada.....	290
4.5.1.8.4	Visualización de salida	291
4.5.2	Manuales de usuario de GTI.BASE.....	292
4.5.2.1	Introducción.....	292
4.5.2.2	Roles	292
4.5.2.3	Manual de usuario del administrador del módulo GTI.....	292
4.5.2.3.1	Funcionalidades.....	292
4.5.2.3.2	Ámbitos	292
4.5.2.3.3	Unidades organizativas.....	293
4.5.2.3.4	Localizaciones	294
4.5.2.3.5	Datos personales.....	295
4.5.2.3.6	Datos personales de usuario	297
4.5.3	Manuales de usuario de GTI.PP.....	299
4.5.3.1	Introducción.....	299
4.5.3.2	Roles	299
4.5.3.3	Manual de usuario del gestor de adquisiciones GTI.PP	300
4.5.3.3.1	Funcionalidades.....	300
4.5.3.3.2	Gestión de proveedores	300
4.5.3.3.3	Gestión de ofertas.....	302
4.5.3.3.4	Gestión de adquisiciones	304
4.5.3.3.5	Marcas, modelos y atributos.....	306
4.5.3.3.6	Recursos asignados.....	306
4.5.3.4	Manual de usuario del usuario sin acceso GTI.PP	306
4.5.3.4.1	Funcionalidades.....	306
4.5.3.4.2	Visualización de adquisición.....	307
5	Implantación de los módulos de activos, adquisiciones y común.....	308
5.1	Introducción	308
5.2	Procedimiento de despliegue	308
5.3	Paquete gti-base	311
5.3.1	Instalador del paquete gti-base.....	314
5.4	Paquete gti-pp	315
5.5	Paquete gti-am.....	317
5.6	Informe de despliegue.....	319
5.6.1	Informe de despliegue de gti.install	319
5.6.2	Informe de instalación del paquete gti-pp.....	335



5.6.3	Informe de instalación del paquete gti-am	336
5.7	Anexos	342
5.7.1	winter.install	342
6	Conclusiones	344
6.1	Líneas de mejora	344
7	Anexos	345
7.1	Fases de trabajo	345
7.2	Reutilización de elementos	346
8	Bibliografía	347
9	Glosario	348

Índice de figuras

Ilustración 1:	Diagrama informal del sistema	32
Ilustración 2:	Diagrama informal del módulo de gestión de activos	33
Ilustración 3:	Modelo físico de datos actual	35
Ilustración 4:	Interfaz de usuario final de sedes y localizaciones	36
Ilustración 5:	Interfaz de usuario final de unidades organizativas	37
Ilustración 6:	Interfaz de usuario final de tipos de unidades y unidades	38
Ilustración 7:	Interfaz de usuario final de marcas, modelos y atributos	39
Ilustración 8:	Interfaz de usuario final de Tipos de elementos y atributos	40
Ilustración 9:	Requisitos del módulo de inventario	41
Ilustración 10:	Requisitos del módulo de adquisiciones	45
Ilustración 11:	Diagrama entidad-relación	47
Ilustración 12:	Unidad de interacción para gestión de ámbitos	54
Ilustración 13:	Interfaz de usuario final de unidades organizativas	55
Ilustración 14:	Interfaz de usuario final de sedes y localizaciones	56
Ilustración 15:	Unidad de interacción para la administración de permisos	57
Ilustración 16:	Unidad de interacción tipos de entrada	57
Ilustración 17:	Unidad de interacción tipos de entrada	58
Ilustración 18:	Unidad de interacción estados de los recursos	58
Ilustración 19:	Interfaz de usuario final de tipos de unidades y unidades	59
Ilustración 20:	Interfaz de usuario final de Tipos de elementos y atributos	59
Ilustración 21:	Unidad de interacción de gestión de proveedores	60
Ilustración 22:	Unidad de interacción de gestión de ofertas	60
Ilustración 23:	Unidad de interacción de gestión de adquisiciones	61
Ilustración 24:	Interfaz de usuario final de marcas, modelos y atributos	62
Ilustración 25:	Unidad de interacción para la gestión de entradas	63
Ilustración 26:	Informe de entrada	63
Ilustración 27:	Unidad de interacción para la gestión de salidas	64
Ilustración 28:	Informe de salida	65
Ilustración 29:	Unidad de interacción de gestión de inventario	65
Ilustración 30:	Unidad de interacción para la edición de activos	66
Ilustración 31:	Informe de activo	67
Ilustración 32:	Pestaña recursos hijo	67
Ilustración 33:	Pestaña histórico	67
Ilustración 34:	Pestaña traza	68
Ilustración 35:	Pestaña Adjuntos	68

Ilustración 36: Unidad de interacción para la gestión de activos.....	69
Ilustración 37: Unidad de interacción de gestión de inventario reducida	70
Ilustración 38: Unidad de interacción para la edición de activos reducida.....	70
Ilustración 39: Unidad de interacción consulta de recursos asignados	71
Ilustración 40: Unidad de interacción para la visualización de recursos	72
Ilustración 41: Unidad de interacción para la visualización de adquisiciones.....	72
Ilustración 42: Unidad de interacción para la visualización de entradas	73
Ilustración 43: Unidad de interacción para la visualización de salidas.....	73
Ilustración 44: Diagrama de despliegue.....	102
Ilustración 45: Modelo físico de datos módulo am.....	104
Ilustración 46: Modelo físico de datos módulo gti	109
Ilustración 47: Modelo físico de datos módulo pp.....	111
Ilustración 48: ERD anotado Administración de permisos	119
Ilustración 49: ERD anotado Tipos de entrada	119
Ilustración 50: ERD anotado Tipos de salida.....	120
Ilustración 51: ERD anotado Estados de los recursos.....	120
Ilustración 52: ERD anotado Tipos de unidades y unidades.....	120
Ilustración 53: ERD anotado Tipos de elementos y atributos	121
Ilustración 54: ERD anotado Consulta de recursos asignados.....	121
Ilustración 55: ERD anotado Consulta de recursos asignados - Recursos hijos.....	122
Ilustración 56: ERD anotado Gestión de entradas	123
Ilustración 57: ERD anotado Gestión de salidas.....	123
Ilustración 58: ERD anotado Gestión de inventario	124
Ilustración 59: ERD anotado Gestión de inventario por tipo de recurso	124
Ilustración 60: ERD anotado Edición de activos	125
Ilustración 61: ERD anotado Gestión de activos	125
Ilustración 62: ERD anotado Estadísticas de tipos de activos por sede.....	125
Ilustración 63: ERD anotado Estadísticas de estados y marcas de activos por sede y tipo	126
Ilustración 64: ERD anotado Gestión de entradas	127
Ilustración 65: ERD anotado Gestión de salidas.....	127
Ilustración 66: ERD anotado Gestión de inventario reducida.....	128
Ilustración 67: ERD anotado Edición de activos reducida	128
Ilustración 68: ERD anotado Gestión de inventario reducida.....	129
Ilustración 69: ERD anotado Edición de activos reducida	129
Ilustración 70: ERD anotado Visualización de recurso.....	130
Ilustración 71: ERD anotado Visualización de entrada.....	130
Ilustración 72: ERD anotado Visualización de salida	131
Ilustración 73: ERD anotado Gestión de ámbitos	132
Ilustración 74: ERD anotado Gestión de unidades organizativas	132
Ilustración 75: ERD anotado Gestión de localizaciones	133
Ilustración 76: ERD anotado Datos personales.....	133
Ilustración 77: ERD anotado Datos personales usuario.....	134
Ilustración 78: ERD anotado Gestión de proveedores	135
Ilustración 79: ERD anotado Gestión de ofertas.....	136
Ilustración 80: ERD anotado Gestión de adquisiciones.....	137
Ilustración 81: ERD anotado Marcas, modelos y atributos	138
Ilustración 82: ERD anotado Visualización de adquisición.....	139
Ilustración 83: Diagrama de caso de uso CU-01: asignación de permisos a un usuario.....	139
Ilustración 84: Diagrama de caso de uso CU-02: gestión de los activos	140
Ilustración 85: Diagrama de caso de uso CU-03: tramitación de una entrada de activos para una adquisición	142
Ilustración 86: Diagrama de caso de uso CU-04: consulta de los recursos asignados.....	143



I Introducción

Ilustración 87: Diagrama de caso de uso CU-05: visualización del recurso.....	144
Ilustración 88: Diagrama de caso de uso CU-01: tramitación de una oferta dada una solicitud	145
Ilustración 89: Diagrama de caso de uso CU-02: visualización del recurso.....	147
Ilustración 90: Diagrama de paquetes.....	148
Ilustración 91: Interfaz de usuario final Consulta de recursos asignados.....	198
Ilustración 92: Interfaz de usuario final Consulta de recursos asignados. Visualización de activos.....	200
Ilustración 93: Esquema menú Administrador del módulo de inventario.....	201
Ilustración 94: Interfaz de usuario final Administración de permisos	203
Ilustración 95: Interfaz de usuario final Tipos de entrada.....	204
Ilustración 96: Interfaz de usuario final Tipos de salida	205
Ilustración 97: Interfaz de usuario final Estados de los recursos	205
Ilustración 98: Interfaz de usuario final Tipos de unidades y unidades	206
Ilustración 99: Interfaz de usuario final Tipos de elementos y atributos	207
Ilustración 100: Esquema menú Gestor de inventario	208
Ilustración 101: Interfaz de usuario final Marcas, modelos y atributos.....	210
Ilustración 102: Interfaz de usuario final Gestión de entradas. Filtros	211
Ilustración 103: Interfaz de usuario final Gestión de entradas. Formulario.....	212
Ilustración 104: Interfaz de usuario final Gestión de entradas. Pestaña Activos	212
Ilustración 105: Interfaz de usuario final Gestión de entradas. Pestaña Resumen	213
Ilustración 106: Interfaz de usuario final Gestión de entradas. Pestaña Adjuntos.....	213
Ilustración 107: Interfaz de usuario final Gestión de salidas. Filtros.....	214
Ilustración 108: Interfaz de usuario final Gestión de salidas. Formulario	215
Ilustración 109: Interfaz de usuario final Gestión de salidas. Pestaña Activos.....	215
Ilustración 110: Interfaz de usuario final Gestión de salidas. Pestaña Resumen	216
Ilustración 111: Interfaz de usuario final Gestión de salidas. Pestaña Adjuntos.....	216
Ilustración 112: Interfaz de usuario final Gestión de inventario. Filtros	217
Ilustración 113: Interfaz de usuario final Gestión de inventario. Activos.....	218
Ilustración 114: Interfaz de usuario final Gestión de inventario por tipo de recurso. Filtros	219
Ilustración 115: Interfaz de usuario final Gestión de inventario por tipo de recurso. Activos	219
Ilustración 116: Interfaz de usuario final Gestión de activos. Combos	220
Ilustración 117: Interfaz de usuario final Gestión de activos. Formulario.....	221
Ilustración 118: Interfaz de usuario final Gestión de activos. Pestaña Propiedades.....	221
Ilustración 119: Interfaz de usuario final Gestión de activos. Pestaña Propiedades Modelo	222
Ilustración 120: Interfaz de usuario final Gestión de activos. Pestaña Recursos hijo	222
Ilustración 121: Interfaz de usuario final Gestión de activos. Pestaña Histórico	223
Ilustración 122: Interfaz de usuario final Gestión de activos. Pestaña Traza	223
Ilustración 123: Interfaz de usuario final Gestión de activos. Pestaña Adjuntos.....	224
Ilustración 124: Interfaz de usuario final Edición de activos.....	226
Ilustración 125: Interfaz de usuario final Edición de activos. Pestaña Detalles	226
Ilustración 126: Interfaz de usuario final Edición de activos. Pestaña Foto	227
Ilustración 127: Interfaz de usuario final Estadísticas de tipos de activos por sede	228
Ilustración 128: Interfaz de usuario final Estadísticas de estados y marcas de activos por sede y tipo	229
Ilustración 129: Esquema menú Técnico avanzado	229
Ilustración 130: Interfaz de usuario final Gestión de inventario reducida. Filtros	234
Ilustración 131: Interfaz de usuario final Gestión de inventario reducida. Formulario	235
Ilustración 132: Interfaz de usuario final Edición de activos reducida. Formulario.....	237
Ilustración 133: Interfaz de usuario final Edición de activos reducida. Pestaña Recursos hijo.....	237
Ilustración 134: Esquema menú Técnico	238
Ilustración 135: Esquema menú Usuario sin acceso	239
Ilustración 136: Interfaz de usuario final Visualización de recurso	241
Ilustración 137: Interfaz de usuario final Visualización de entrada.....	242

Ilustración 138: Interfaz de usuario final Visualización de salida	242
Ilustración 139: Esquema menú Administrador del módulo común.....	243
Ilustración 140: Interfaz de usuario final Gestión de ámbitos	244
Ilustración 141: Interfaz de usuario final Gestión de unidades organizativas	246
Ilustración 142: Interfaz de usuario final Gestión de localizaciones	248
Ilustración 143: Interfaz de usuario final Datos personales.....	249
Ilustración 144: Interfaz de usuario final Datos personales de usuario	250
Ilustración 145: Esquema menú Gestor de adquisiciones.....	251
Ilustración 146: Interfaz de usuario final Gestión de proveedores	253
Ilustración 147: Interfaz de usuario final Gestión de proveedores. Pestaña Adjuntos.....	253
Ilustración 148: Interfaz de usuario final Gestión de ofertas. Formularios	255
Ilustración 149: Interfaz de usuario final Gestión de ofertas. Pestaña Elementos.....	255
Ilustración 150: Interfaz de usuario final Gestión de ofertas. Pestaña Adjuntos	255
Ilustración 151: Interfaz de usuario final Gestión de adquisiciones. Formularios.....	257
Ilustración 152: Interfaz de usuario final Gestión de adquisiciones. Pestaña Elementos.....	257
Ilustración 153: Interfaz de usuario final Gestión de adquisiciones. Pestaña Adjuntos	258
Ilustración 154: Esquema menú Usuario sin acceso	258
Ilustración 155: Interfaz de usuario final Visualización de adquisición	259
Ilustración 156: Interfaz de usuario Permisos.....	261
Ilustración 157: Interfaz de usuario Permisos. Pestaña Gestores de adquisiciones.....	261
Ilustración 158: Interfaz de usuario Permisos. Pestaña Gestores de inventario.....	262
Ilustración 159: Interfaz de usuario Permisos. Pestaña Técnicos avanzados.....	262
Ilustración 160: Interfaz de usuario Permisos. Pestaña Técnicos	262
Ilustración 161: Interfaz de usuario Tipos de entradas	263
Ilustración 162: Interfaz de usuario Tipos de salidas.....	264
Ilustración 163: Interfaz de usuario Estado de los recursos.....	264
Ilustración 164: Interfaz de usuario Tipos de unidades y unidades	265
Ilustración 165: Interfaz de usuario Tipos de elementos y atributos.....	266
Ilustración 166: Interfaz de usuario Recursos asignados	268
Ilustración 167: Interfaz de usuario Recursos asignados. Pestaña Propiedades	268
Ilustración 168: Interfaz de usuario Recursos asignados. Pestaña Recursos hijos	268
Ilustración 169: Interfaz de usuario Marcas, modelos y atributos	270
Ilustración 170: Interfaz de usuario Gestión de entradas.....	272
Ilustración 171: Interfaz de usuario Gestión de entradas. Filtro avanzado.....	273
Ilustración 172: Interfaz de usuario Gestión de entradas. Pestaña Activos	273
Ilustración 173: Interfaz de usuario Gestión de entradas. Pestaña Resumen.....	273
Ilustración 174: Interfaz de usuario Gestión de entradas. Pestaña Adjuntos	273
Ilustración 175: Interfaz de usuario Gestión de salidas	275
Ilustración 176: Interfaz de usuario Gestión de salidas. Filtro avanzado	276
Ilustración 177: Interfaz de usuario Gestión de salidas. Pestaña Activos.....	276
Ilustración 178: Interfaz de usuario Gestión de salidas. Pestaña Resumen	276
Ilustración 179: Interfaz de usuario Gestión de salidas. Pestaña Adjuntos.....	276
Ilustración 180: Interfaz de usuario Gestión de inventario	277
Ilustración 181: Interfaz de usuario Gestión de inventario. Filtro avanzado.....	277
Ilustración 182: Interfaz de usuario Gestión de inventario. Filtro unidad organizativa.....	278
Ilustración 183: Interfaz de usuario Gestión de inventario. Tabla Activo.....	278
Ilustración 184: Interfaz de usuario Gestión de inventario por tipo de recurso.....	279
Ilustración 185: Interfaz de usuario Gestión de inventario por tipo de recurso. Tabla Activo	279
Ilustración 186: Interfaz de usuario Gestión de activos.....	280
Ilustración 187: Interfaz de usuario Gestión de activos. Pestaña Propiedades	281
Ilustración 188: Interfaz de usuario Gestión de activos. Pestaña Recursos Hijo.....	281
Ilustración 189: : Interfaz de usuario Gestión de activos. Pestaña Histórico.....	281



I Introducción

Ilustración 190: : Interfaz de usuario Gestión de activos. Pestaña Traza.....	281
Ilustración 191: Interfaz de usuario Gestión de activos. Pestaña Adjuntos	282
Ilustración 192: Interfaz de usuario Edición de activos.....	282
Ilustración 193: Interfaz de usuario Tipos de activos por sede	283
Ilustración 194: Interfaz de usuario Estados y marcas de activos por sede y tipo	284
Ilustración 195: Interfaz de usuario Gestión de inventario reducida	286
Ilustración 196: Interfaz de usuario Gestión de inventario reducida. Filtro avanzado	286
Ilustración 197: Interfaz de usuario Gestión de inventario reducida. Filtro unidad organizativa....	286
Ilustración 198: Interfaz de usuario Gestión de inventario reducida. Tabla Activo.....	287
Ilustración 199: Interfaz de usuario Edición de activos reducida	288
Ilustración 200: Interfaz de usuario Visualización de recurso	290
Ilustración 201: Interfaz de usuario Visualización de entrada	291
Ilustración 202: Interfaz de usuario Visualización de salida.....	291
Ilustración 203: Interfaz de usuario Ámbitos.....	293
Ilustración 204: Interfaz de usuario Unidades organizativas	294
Ilustración 205: Interfaz de usuario Localizaciones	295
Ilustración 206: Interfaz de usuario Datos personales	296
Ilustración 207: Interfaz de usuario Datos personales. Pestaña Teléfono	296
Ilustración 208: Interfaz de usuario Datos personales. Pestaña Dirección	297
Ilustración 209: Interfaz de usuario Datos personales. Pestaña Email	297
Ilustración 210: Interfaz de usuario Datos personales. Pestaña Notas.....	297
Ilustración 211: Interfaz de usuario Datos personales de usuario	298
Ilustración 212: Interfaz de usuario Datos personales de usuario. Pestaña Teléfono	298
Ilustración 213: Interfaz de usuario Datos personales de usuario. Pestaña Dirección.....	299
Ilustración 214: Interfaz de usuario Datos personales de usuario. Pestaña Email	299
Ilustración 215: Interfaz de usuario Datos personales de usuario. Pestaña Notas	299
Ilustración 216: Interfaz de usuario Gestión de proveedores.....	301
Ilustración 217: Interfaz de usuario Gestión de proveedores. Pestaña Contactos	301
Ilustración 218: Interfaz de usuario Gestión de proveedores. Pestaña Adjuntos	302
Ilustración 219: Interfaz de usuario Gestión de ofertas	303
Ilustración 220: Interfaz de usuario Gestión de ofertas. Pestaña Elementos	304
Ilustración 221: Interfaz de usuario Gestión de ofertas. Pestaña Adjuntos.....	304
Ilustración 222: Interfaz de usuario Gestión de adquisiciones	305
Ilustración 223: Interfaz de usuario Gestión de adquisiciones. Pestaña Elementos	306
Ilustración 224: Interfaz de usuario Gestión de adquisiciones. Pestaña Adjuntos.....	306
Ilustración 225: Interfaz de usuario Visualización de adquisición.....	307

Índice de tablas

Tabla 1: Tabla rol/funcionalidad	54
Tabla 2: Tabla Rol - Unidad de interacción.....	118
Tabla 3: Tabla Rol - Unidad de interacción.....	131
Tabla 4: Tabla Rol - Unidad de interacción.....	134
Tabla 5: Caso de uso CU-01: asignación de permisos a un usuario	140
Tabla 6: Caso de uso CU-02: gestión de los activos	142
Tabla 7: Caso de uso CU-03: tramitación de una entrada de activos para una adquisición.....	143
Tabla 8: Caso de uso CU-04: consulta de los recursos asignados	144
Tabla 9: Caso de uso CU-05: visualización del recurso.....	145
Tabla 10: Caso de uso CU-01: tramitación de una oferta dada una solicitud	146
Tabla 11: Caso de uso CU-02: visualización del recurso.....	147
Tabla 12: Pruebas del administrador del módulo de inventario.....	152

Tabla 13: Pruebas del gestor de inventario	156
Tabla 14: Pruebas del técnico avanzado	159
Tabla 15: Pruebas del técnico.....	160
Tabla 16: Pruebas del usuario regular	160
Tabla 17: Pruebas del usuario sin acceso	161
Tabla 18: Matriz de trazabilidad del administrador del módulo	162
Tabla 19: Matriz de trazabilidad del gestor de inventario.....	162
Tabla 20: Matriz de trazabilidad del técnico avanzado	163
Tabla 21: Matriz de trazabilidad del técnico	163
Tabla 22: Matriz de trazabilidad del usuario regular	163
Tabla 23: Matriz de trazabilidad del usuario sin acceso	163
Tabla 24: Pruebas del administrador del módulo común.....	165
Tabla 25: Pruebas del gestor de adquisiciones.....	167
Tabla 26: Pruebas del usuario sin acceso	167
Tabla 27: Matriz de trazabilidad del gestor de adquisiciones	167
Tabla 28: Matriz de trazabilidad del usuario regular	168



1 Introducción

Este documento presenta la memoria del trabajo de fin de grado titulada *Gestión de inventario de GTI* para la obtención del título de Graduado en Ingeniería de las Tecnologías de Telecomunicación por la Universidad de Sevilla del alumno Álvaro Marín Sánchez.

En la actualidad, la mayoría de las organizaciones e industrias son conscientes de que una óptima gestión de sus activos ofrece grandes ventajas competitivas. La organización internacional de normalización ISO recoge en su normativa ISO55000 [1], ISO55001 [2] e ISO55002 [3] aspectos relacionados con la creación de un sistema de gestión de activos:

- La **ISO55000**, lanzada en enero de 2014, no solo afirma la alta relevancia de la parte de desarrollo e implementación para un sistema de gestión de activos, sino de la parte documental entre la que destaca el **plan estratégico de gestión de activos (PEGA)**. **Campbell** en 2016 [4] ampliaría este concepto añadiendo que la gestión de activos es la interacción de la áreas de una organización (finanzas, logística, compra, etc) a través de actividades que permiten gestionar los riesgos correctamente para generar valor a la organización.
- La **ISO55001** expone que *“La organización debe utilizar el PEGA para guiar la definición de sus objetivos organizacionales y para describir el rol del sistema de gestión de activos para alcanzar dichos objetivos. Esto incluye las estructuras, roles y responsabilidades necesarias para establecer el sistema de gestión de activos y operarlo eficazmente”*.
- Finalmente, la **ISO55002** define las **directrices para la aplicación** de la ISO55001 como, por ejemplo, una guía del contenido PEGA.

Respecto al objetivo de este proyecto, se han diseñado tres módulos adicionales a los dos existentes que formarán parte de la aplicación web GTI:

- El módulo de **gestión de proyectos** [11] permite la definición de distintos tipos de proyectos para un departamento y el seguimiento de su ejecución.
- El módulo de **gestión de servicios y procesos** [12] permite gestionar los procedimientos que realiza un departamento y los servicios ofrecidos tanto a los usuarios como a las distintas áreas del departamento.
- El módulo de **gestión de inventario** ofrece la gestión de los activos, control de stock, adquisiciones y las entradas y salidas del material.
- El módulo de **gestión de adquisiciones** complementa a la gestión de inventario, permitiendo la gestión de proveedores, ofertas y compras.
- El módulo **común** utilizado por todos los módulos que posibilita asignar la gestión en localizaciones, ubicaciones y a los diferentes usuarios definidos.

Los tres últimos módulos mencionados serán los que se detallarán a la largo de este documento.

Referente al tipo de activo, en la aplicación propuesta se trabaja principalmente con la gestión de activos de software y hardware. Esto incluye la gestión de compras, licencias y todo tipo de activos físicos como equipamiento de ordenadores, dispositivos, cableado, etc.

La implementación se ha desarrollado utilizando el MBUIDE **WAINÉ** [13]. Esta herramienta ofrece gran independencia entre el sistema operativo, servidor web y navegador usado, flexibilidad a la hora de añadir sistemas de autenticación y widgets y seguridad ya que incorpora métodos de autenticación y restricciones de acceso, entre otros.

Por otra parte, analizando el mercado, algunas de las herramientas destacadas para la gestión de activos son:

- **ServiceNow**: ha sido líder en el mercado ITSM durante 5 años. Esta herramienta ofrece un producto llamado ServiceNow Asset Management que proporciona un conjunto de funciones extendidas para ITAM. Entre estas se encuentran la gestión de auditorías, inventarios o mantenimiento. Sin embargo, esta herramienta no ofrece una gestión de proveedores o adquisiciones.
- **AssetCloud**: es un sistema de gestión de activos que está diseñado para pequeñas empresas, Gobierno y Educación. Permite realizar auditorías rápidamente y tiene alcance tanto en móviles Android como iOS. Su principal inconveniente es que carece de gestión de adquisiciones.
- **ManageEngine AssetExplorer**: es una herramienta muy potente que se encuentra en uso por más de 100.000 empresas de todo el mundo. Ofrece el seguimiento y la gestión de los activos en una única vista. Además, permite la gestión de licencias software y realizar auditorías. A pesar de que presenta una gestión de activos muy completa, no tiene un soporte en móviles iOS.
- **Asset Panda**: se trata de una plataforma simple y personalizada para la administración de activos. Sus principales características son: escaneo de código de barras, creación de informes, flujo de trabajo mediante tickets de soporte, administración del ciclo de vida completo de los activos, seguridad basada en roles, entre otros. No obstante, no ofrece una gestión de licencias ni un seguimiento de costes.

En lo que concierne al ciclo de vida para el desarrollo de una aplicación, **WAINE** ofrece una metodología que da soporte a este proceso [15]. Las fases indicadas son las siguientes:

- **Análisis**: incluye la obtención de requisitos y la redacción del documento de análisis con las funcionalidades, roles de los usuarios y bocetos de las principales interfaces de usuario e informes empleados para cada rol.
- **Modelo de conceptos**: incluye la tabla rol-funcionalidad que contiene las acciones e IU necesarias para cada rol y los DER que se requieran.
- **Anotación del diagrama Entidad-Relación**: se trata de, a partir del DER que afecta a la unidad de interacción que se está definiendo, anotar sobre él las partes que lo forman (formularios y contenedores).
- **Especificación en ASL**: se trata del documento en lenguaje ASL que define las instancias de las clases de los modelos de la IU de WAINE.
- **Generación de repositorios y personalización**: concluye con el proceso de desarrollo e incluye los 3 repositorios: Repositorio de la IU (a partir del documento ASL), Repositorio de configuraciones (aspectos concretos de la IU como colores, estilo de letra, etc) y la Base de Datos (a partir de los DER y del código SQL). Además, como se mencionó anteriormente, se le pueden añadir widgets, layouts y métodos de autenticación.

Finalmente, como resultado de este proceso de desarrollo, se han elaborado una serie de documentos que se detallan a lo largo de este trabajo de fin de grado:



1 Introducción

- En la siguiente sección se presenta la fase de análisis donde se describen los requisitos, el estado inicial y los bocetos de las interfaces necesarias junto a sus informes.
- Posteriormente, se detalla la descripción de la fase de Diseño que incluye el diseño de las interfaces de usuario y de la arquitectura de la aplicación.
- Se prosigue con la fase de Implementación de las interfaces y de la base de datos de la aplicación y su implantación.
- Para finalizar el documento, se presentan las conclusiones del proyecto y se proponen diversas mejoras al mismo.

2 Análisis de los módulos de activos, adquisiciones y común de GTI

2.1 Introducción

El proyecto GTI (Gestión TIC) pretende desarrollar una aplicación web modular que permita controlar los activos TIC de una empresa u organismo y gestionar los procesos, proyectos y servicios ofrecidos por el departamento TIC. Inicialmente se han propuesto cuatro módulos:

1. El módulo común, que será responsable de la gestión y mantenimiento de todos los elementos compartidos entre los diversos módulos de la aplicación.
2. El módulo de gestión de procesos, desde el cual se gestionan los distintos procedimientos que realiza el departamento de informática y los servicios ofrecidos a los usuarios y a las distintas áreas del departamento.
3. El módulo de gestión de proyectos, que permite la definición de los proyectos tipo del departamento y el seguimiento de su ejecución.
4. El módulo de gestión de activos, encargado entre otros aspectos del inventario, control de stock, adquisiciones y las entradas y salidas de material. Este Trabajo de fin de Grado (TFG) trata precisamente sobre este último módulo.



Ilustración 1: Diagrama informal del sistema

La aplicación GTI debe tener un diseño modular, es decir, cada sección de esta será un módulo, que será empaquetado e integrado en un marco de integración base, la aplicación Winter, (ver sección 86). El entorno de desarrollo empleado para todos los módulos es **WAINE**.

2.1.1 El módulo de gestión de activos

El módulo de gestión de activos viene a dar respuesta a una de las necesidades de gestión básicas dentro de la gestión TIC: las adquisiciones y el control e inventario de activos. Este módulo debe



permitir la gestión de las compras en distintos ámbitos (territoriales, departamentales, de administración, etc.). También debe permitir la definición de diferentes tipos de elementos inventariables con sus respectivas propiedades. Por último debe permitirse controlar y seguir las adquisiciones, controlando su ubicación y asignación, es decir, quién es responsable del activo concreto.

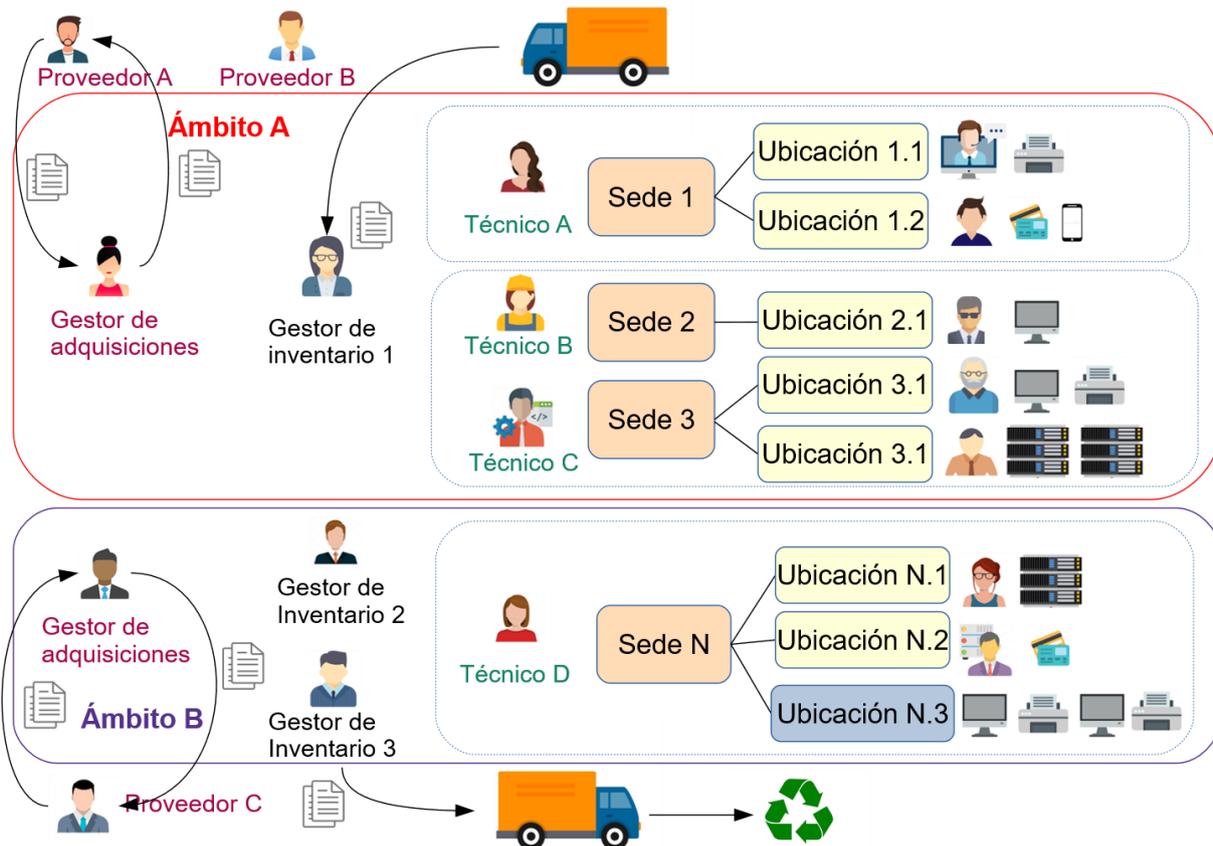


Ilustración 2: Diagrama informal del módulo de gestión de activos

En la ilustración anterior se presentan un par de ámbitos. En ellos pueden existir varios gestores de adquisiciones encargados de los procesos de compra. Estos gestores realizan solicitudes de adquisiciones a distintos proveedores a las que los proveedores responden con sus correspondientes ofertas. Una vez que se decide realizar la compra de un determinado bien éste es recepcionado por el gestor de inventario. La entrega del material representa una entrada en el sistema de inventario y normalmente vendrá acompañada de cierta documentación: albaranes, documentos de garantía, etc. Las sedes de los distintos ámbitos de gestión están organizadas en ubicaciones. En las distintas ubicaciones existirán usuarios que tienen asignados diverso equipamiento y del que son responsables. Pueden existir ubicaciones en las que no haya usuarios pero que tengan una asignación de equipamiento como por ejemplo un centro de proceso de datos o un almacén. Aún así, ese equipamiento debe tener un responsable asignado. En las distintas sedes, los técnicos dedicados a la instalación y mantenimiento del equipamiento hardware y software mantienen actualizado el inventarios de activos.

2.1.2 Objetivos del proyecto

1. Elaborar una aplicación para la gestión de activos TIC.
2. Contribuir a la normalización de la gestión de activos en los organismos en los que se implante esta aplicación.
3. Contribuir a la construcción de una aplicación que dé respuesta a la problemática de la gestión TIC en una empresa u organismo.
4. Desarrollar un aplicación de gran tamaño modularizada que pueda servir como ejemplo al desarrollo de otros proyectos similares en tamaño y complejidad dentro de la plataforma **WAINE**.

2.2 Situación actual

Actualmente existen desarrollos preeliminares del modelo de datos y de algunas interfaces de usuario.

2.2.1 Modelo físico de datos

A continuación, se muestra el modelo físico de datos existente para el módulo de inventario.

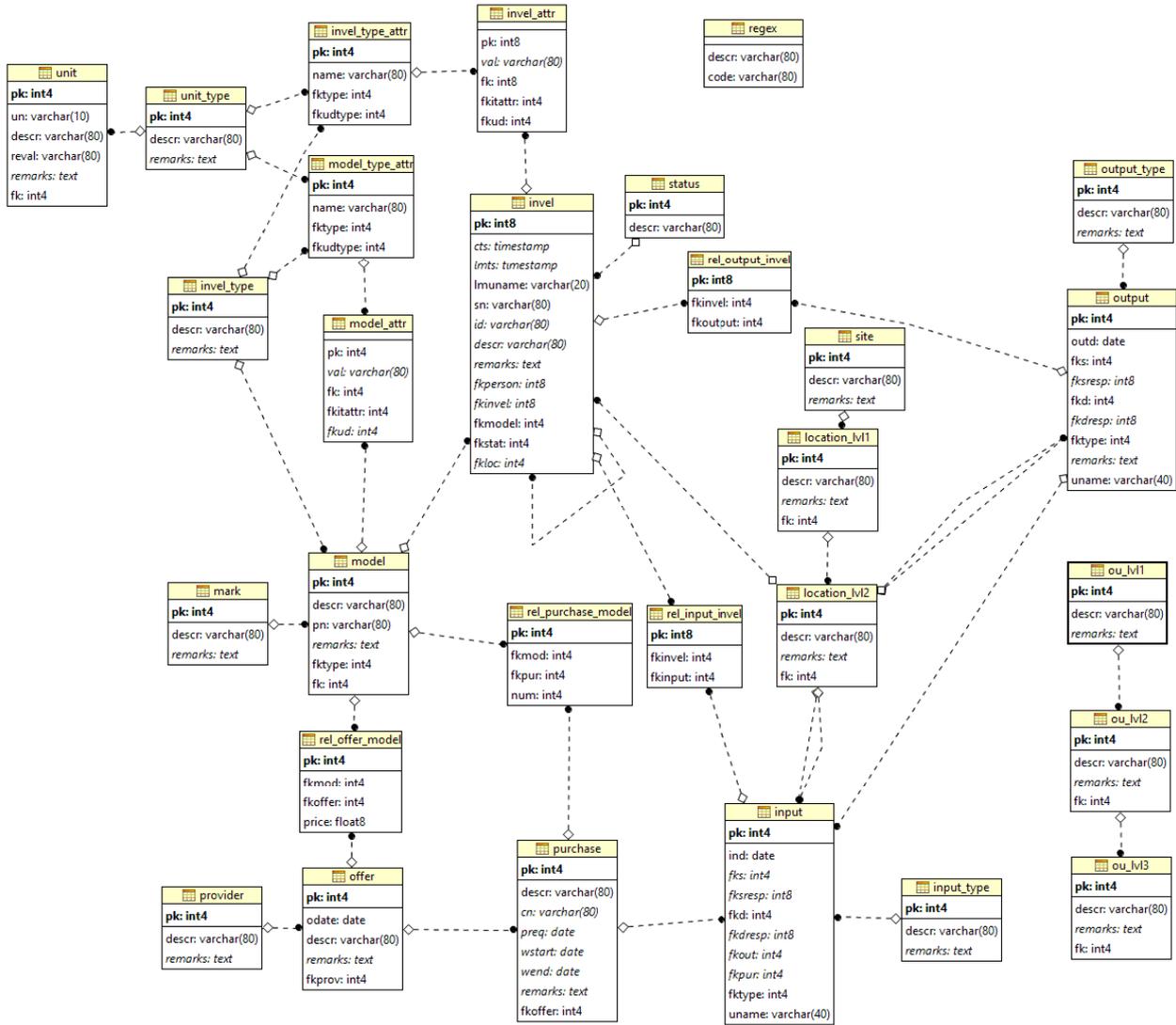


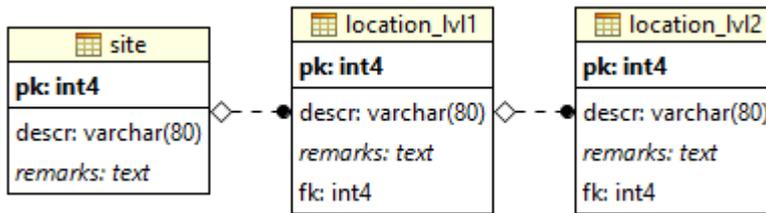
Ilustración 3: Modelo físico de datos actual

2.2.2 Interfaces de usuario desarrolladas

En esta sección se describen las distintas interfaces de usuario que ya han sido desarrolladas en su versión preliminar.

2.2.2.1 Sedes y Localizaciones

Esta interfaz de usuario permite definir las sedes y sus localizaciones, elementos comunes a varios módulos.



No debería formar parte de este módulo, se trata más bien de una interfaz de administración del sistema común a todos los módulos de GTI.

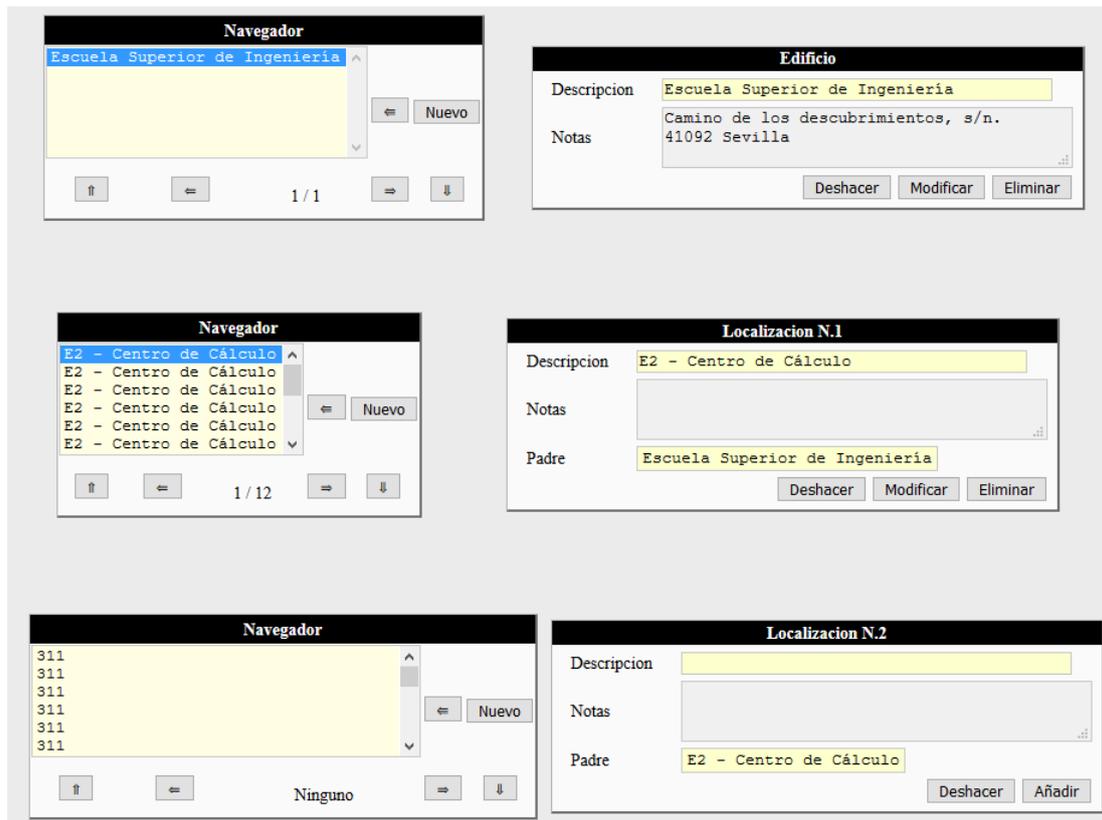


Ilustración 4: Interfaz de usuario final de sedes y localizaciones

2.2.2.2 Unidades Organizativas

La interfaz de usuario de unidades administrativas permite gestionar una organización jerárquica de tres niveles para la organización.



Es una interfaz de usuario para la administración del sistema común a todos los módulos de GTI.

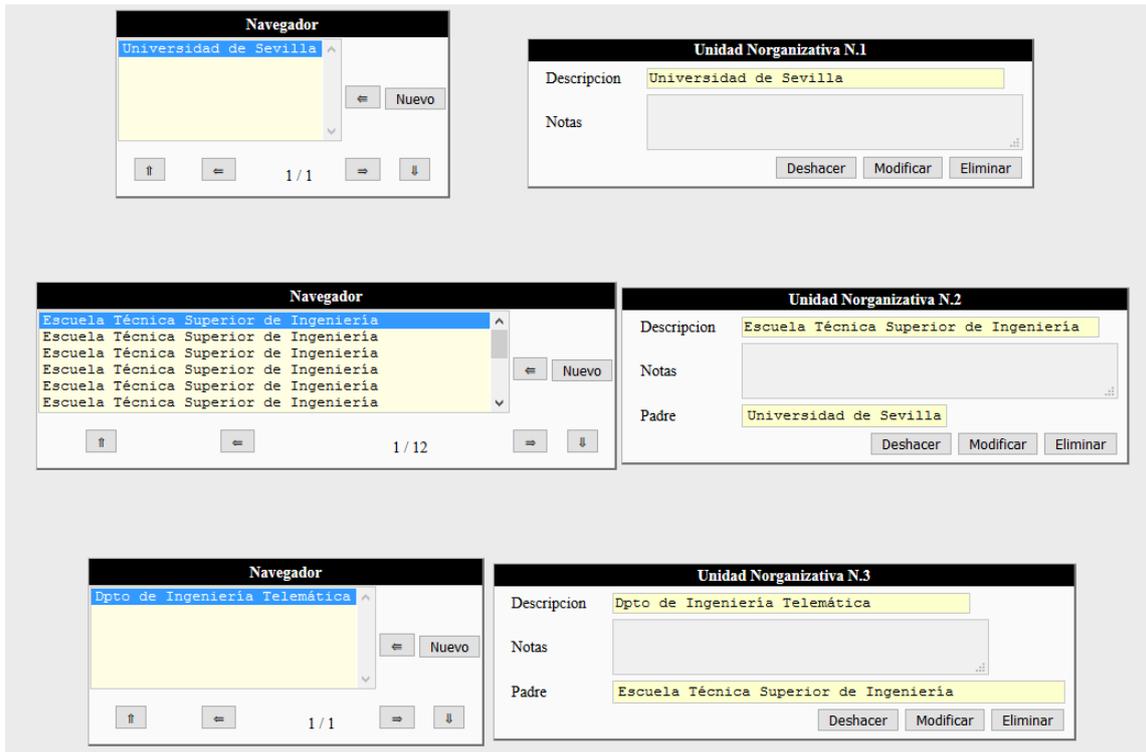
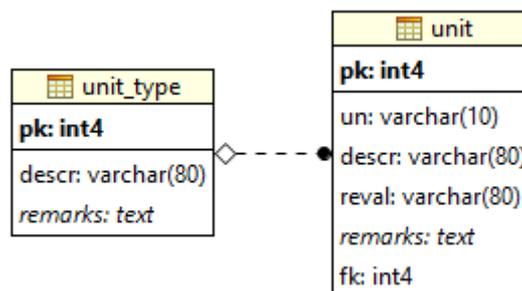


Ilustración 5: Interfaz de usuario final de unidades organizativas

2.2.2.3 Unidades y tipos de unidades

Esta interfaz de usuario permite gestionar las unidades y los tipos de unidades de las propiedades de los activos y de los modelos, así como una validación de los valores de los mismos.



Su interfaz es la siguiente:

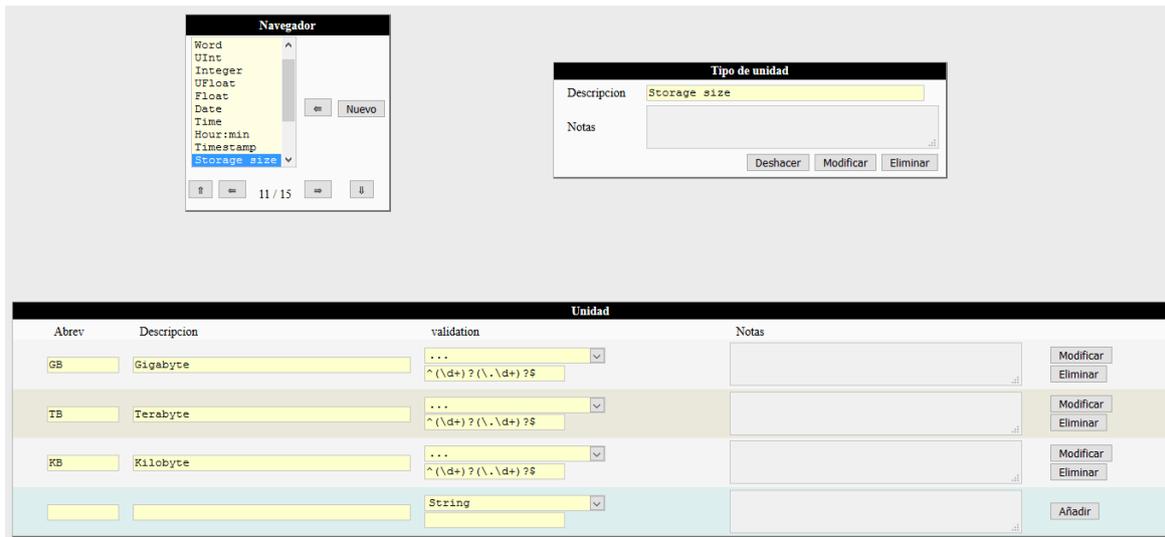
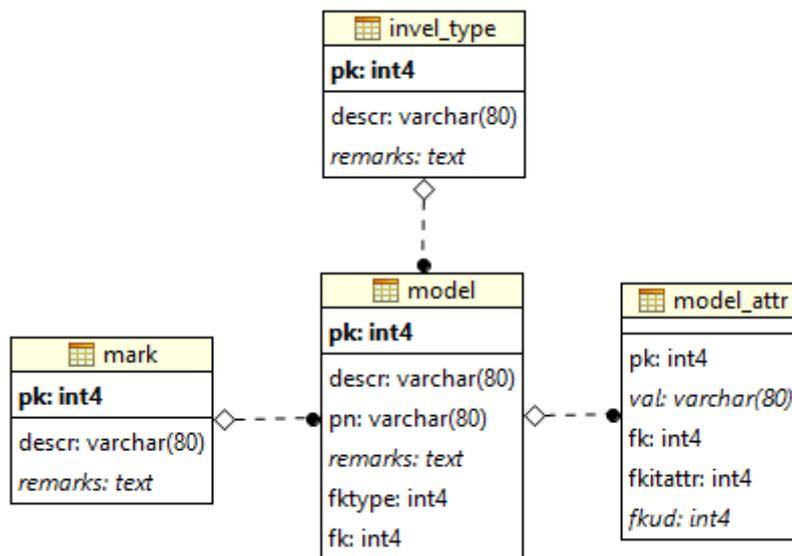


Ilustración 6: Interfaz de usuario final de tipos de unidades y unidades

2.2.2.4 Marcas, modelos y atributos

Esta interfaz de usuario permite crear marcas, modelos y atributos para los activos.



Su interfaz es la siguiente:

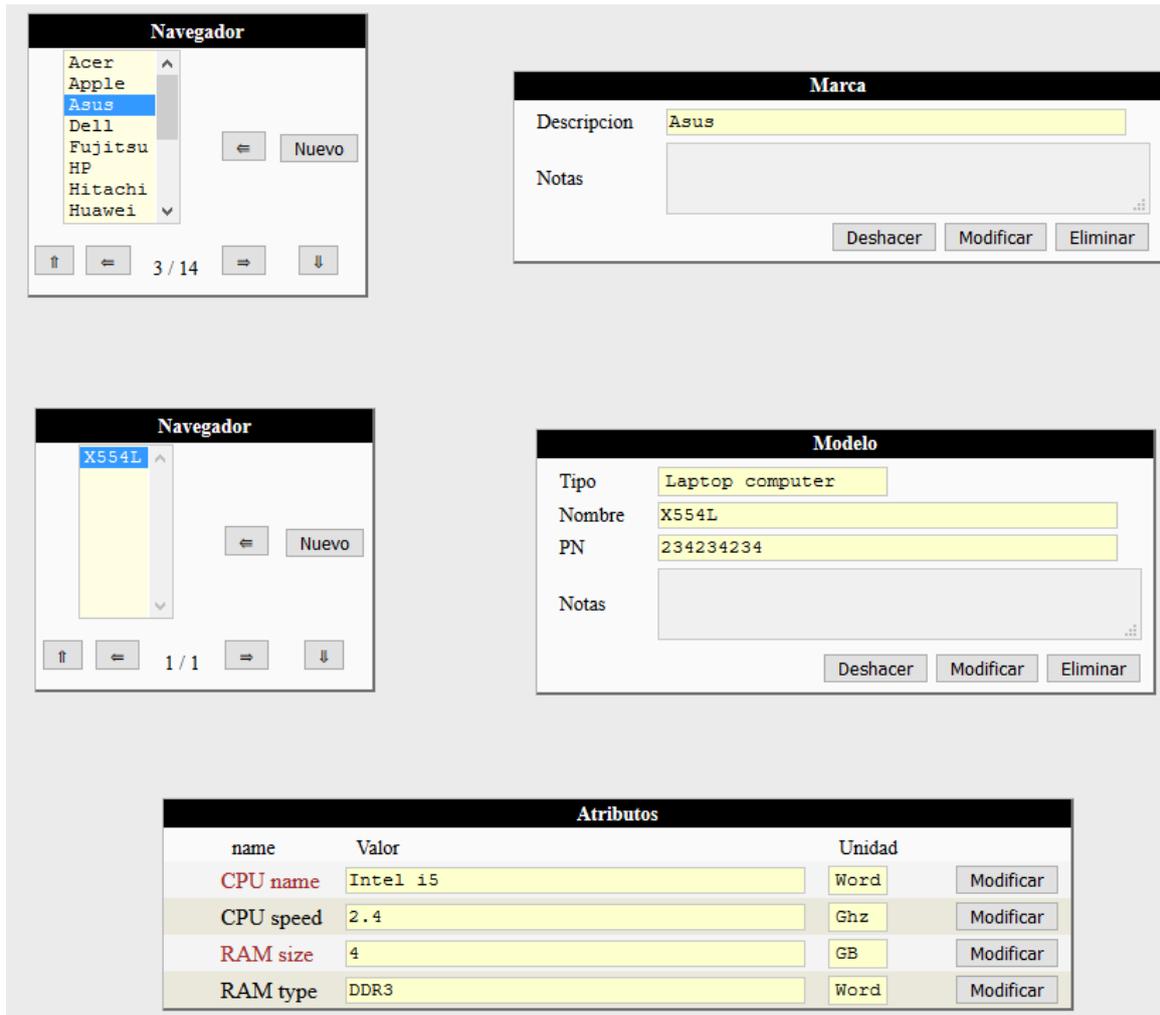


Ilustración 7: Interfaz de usuario final de marcas, modelos y atributos

2.2.2.5 Tipos de elementos inventariables, atributos de modelo y atributos de elemento

Esta interfaz de usuario permite crear tipos de elementos y asignarles atributos relacionados con el modelo o con el propio activo.

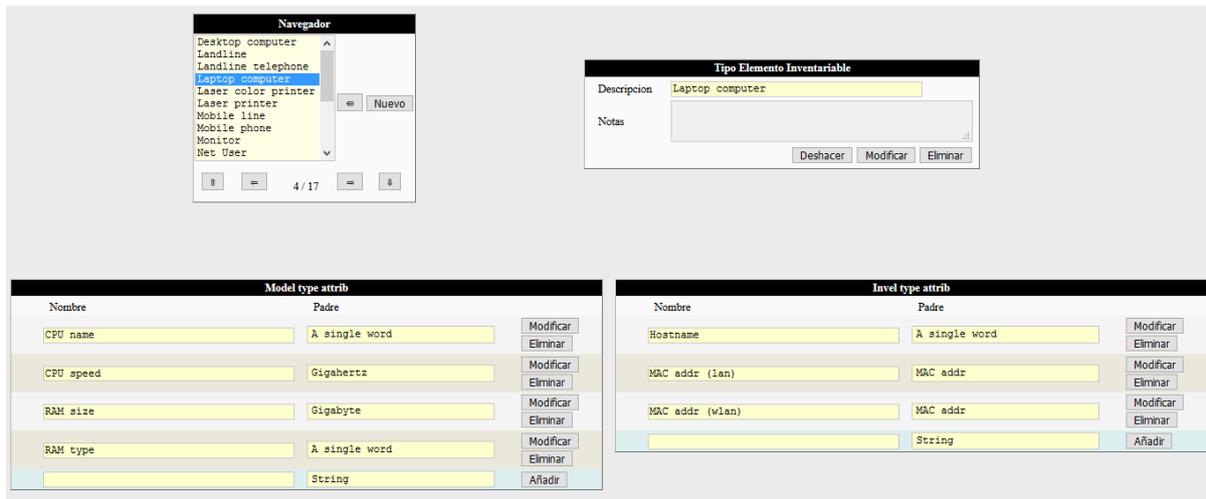


Ilustración 8: Interfaz de usuario final de Tipos de elementos y atributos

2.3 Catálogo de requisitos

2.3.1 Catálogo de requisitos modulo am

En esta sección se enumeran los requisitos identificados para el módulo de gestión de activos.



Ilustración 9: Requisitos del módulo de inventario

2.3.1.1 Funcionalidades

Entre los requisitos funcionales del sistema están los siguientes:

- **R1.1 Ámbitos**
 - *R1.1.1 Las sedes deben contemplar ámbitos.* El módulo de gestión de activo debe contemplar ámbitos. Normalmente serán provinciales, regionales o funcionales (departamentos, áreas, etc.)
 - *R1.1.2 Siempre existirá un ámbito global.* Existirá un ámbito global por defecto. Estará identificado por un código especial y no podrá ser eliminado.

- *R1.1.3 Los gestores y técnicos se definen por ámbito.* El administrador del módulo define para cada ámbito los perfiles de gestores y técnicos autorizados.

- **R1.2 Sedes**

- *R1.2.1 Las sedes disponen de ubicaciones.* En las distintas sedes existentes en el sistema aparecen diversas ubicaciones o localizaciones. Una jerarquía de dos niveles es suficiente para adaptarse a las necesidades de localización de los diferentes activos.
- *R1.2.2 Algunas ubicaciones son consideradas almacenes.* Existen algunas localizaciones que tienen consideración de almacenes.

- **R1.3 Movimientos de activos**

- *R1.3.1 El sistema debe contemplar entradas y salidas de activos.* Los movimientos de activos deben quedar reflejado en el sistema.
- *R1.3.2 Los movimientos tendrán un origen y un destino.* Se trata de información relevante en el movimiento. Hay que tener en cuenta que pueden existir movimientos internos a la organización, por ejemplo, de un departamento a otro o movimientos de activos entre sedes, o bien movimientos externos cuando bien el origen o el destino del movimiento son entidades externas a la organización (adquisiciones, reciclaje, etc.)
- *R1.3.3 Tendrán también asignados responsables.* Existe siempre al menos un responsable del movimiento.
- *R1.3.4 Existirán tipos de entradas.* Se podrán clasificar las entradas de activos.
- *R1.3.5 Tipos de entradas predefinidas: Adquisición, Interna, devolución.* Algunos de los tipos de entrada del sistema estarán predefinidos en el sistema y no podrán ser modificados ni eliminados.
 - Adquisición: entrada externa provocada por una compra.
 - Interna: entrada provocada por un movimiento interno.
 - Devolución: entrada provocada por un movimiento interno asociado a un préstamo

Cualquier otro tipo de entrada definida por el administrador del módulo será considerado como una entrada interna.

- *R1.3.6 Existirán tipos de salidas.* Se podrán clasificar las salidas de activos.
- *R1.3.7 Tipos de salidas predefinidas: Interno, préstamo, baja.* Algunos de los tipos de salida del sistema estarán predefinidos en el sistema y no podrán ser modificados ni eliminados.
 - Interna: salida provocada por un movimiento interno.
 - Préstamo: salida provocada por una cesión temporal de material a un
 - baja: salida externa provocada por una baja de material.

Cualquier otro tipo de salida definida por el administrador del módulo será considerado como una salida interna.

- *R1.3.8 Se debe poder generar la documentación asociada al movimiento concreto.* Tanto entradas como salidas de material tendrán asociada una documentación que será generada de forma automática por la aplicación.



- *RI.3.9 Se debe poder realizar el alta o baja de los activos relacionados en el movimiento en función del tipo.*
 - Una entrada de tipo adquisición debe permitir realizar el alta de los elementos de la compra.
 - Una salida de tipo baja debe realizar la baja efectiva de los elementos que la componen.
- *RI.3.10 El tipo de movimiento puede determinar la copia de datos de una adquisición.* Una entrada de tipo adquisición debería copiar al movimiento los activos relacionados en la adquisición.
- **R1.4 Activos**
 - *RI.4.1 Siempre están asignados a una ubicación.* Un activo siempre se encontrará en una localización.
 - *RI.4.2 Están asignados a un usuario.* Un activo siempre tendrá asignado a alguien responsable del mismo. Un activo siempre tendrá un responsable asignado. Normalmente este responsable será la persona que tiene asignado el equipamiento, pero por ejemplo en el caso de tratarse de activos que se encuentren en un almacén será responsable del mismo el responsable del almacén.
 - *RI.4.3 Los activos son de un tipo.* La pertenencia a un tipo determina unos campos personalizados. Se debe permitir la definición de tipos de activos inventariables, así como la definición de campos personalizados asociados a los mismos.
 - *RI.4.4 Existirán estados de los recursos.* Los activos tendrán un estado asignado que deberá estar definido previamente.
 - *RI.4.5 Estados de los recursos predefinidos: Activo, baja.* Los estados de los recursos predefinidos serán Activo y Baja
 - *RI.4.6 Los activos son de un modelo determinado.* El sistema debe contemplar la existencia de marcas y modelos. Los modelos también podrán tener campos personalizados.
 - *RI.4.7 Tienen fecha de alta y fecha de baja en el sistema.* La fecha de alta se corresponderá con la fecha de entrada de la adquisición en el sistema. La fecha de baja se corresponderá con la fecha de salida en baja del sistema.
 - *RI.4.8 Se debe poder trazar los cambios durante la vida del activo.* El sistema trazará la historia del activo, detallando los cambios realizados sobre el mismo a lo largo de su existencia en el sistema.
 - *RI.4.9 Se deben poder realizar anotaciones sobre el activo.* Los gestores y técnicos deben poder realizar anotaciones sobre un activo.
 - *RI.4.10 Se deben poder adjuntar documentos a un activo.* Los gestores y técnicos deben poder adjuntar documentos sobre un activo: órdenes de trabajo relativas a reparaciones, revisiones, etc.

- *R1.4.11 Debe permitirse generar relaciones de activos filtrando por sus características.* Debe ser posible extraer relaciones de activos filtrando por los campos principales de los mismos.
- *R1.4.12 Debe permitirse generar relaciones de activos filtrando por sus campos personalizados.* Debe ser posible extraer relaciones de activos filtrando por los campos principales de los mismos.

2.3.1.2 Usuarios

- **R2.1 Administrador del módulo**
 - *R2.1.1 Establece gestores de adquisiciones por ámbitos.* El administrador del módulo establece los permisos de acceso de los gestores de adquisiciones por ámbito.
 - *R2.1.2 Establece gestores de inventario y técnicos por sede.* El administrador del módulo establece la visibilidad de los gestores de inventario y de los técnicos por ámbito.
 - *R2.1.3 Define tipos de entradas y salidas.* Los administradores del módulo pueden definir nuevos tipos de entradas y salidas.
 - *R2.1.4 Define tipos de recursos y sus campos personalizados.* Los administradores del módulo definen los tipos de recursos y los campos personalizados de activo.
- **R2.3 Gestor de inventario**
 - *R2.3.1 Gestiona marcas y modelos con sus atributos.* Los gestores de inventario también pueden definir marcas y modelos y pueden especificar los campos personalizados de estos últimos.
 - *R2.3.2 Gestiona de forma completa los recursos de las sedes en las que está autorizado.* Un gestor de inventario puede realizar cualquier modificación sobre los activos en las sedes de los ámbitos que gestiona.
 - *R2.3.3 Recepciona material por adquisiciones, movimientos o devoluciones.* Puede realizar diversas recepciones y completar la información o documentación relacionada con la misma.
 - *R2.3.4 Realiza salidas de recursos en las sedes que gestiona.* Puede autorizar salidas de material completando la información y documentación relacionada con la misma.
 - *R2.3.5 Extrae relaciones de equipamiento en base a sus propiedades.* El gestor de inventario puede generar listados de activos en base a consultas sobre sus atributos en aquellos ámbitos en los que está autorizado.
 - *R2.3.6 Extrae estadísticas relativas al equipamiento.* También puede generar estadísticas de los activos del sistema dentro de los ámbitos que gestiona.
- **R2.4 Técnico Avanzado**
 - *R2.4.1 Recepciona material por adquisiciones, movimientos o devoluciones.* Un técnico avanzado puede realizar diversas recepciones y completar la información o



documentación relacionada con la misma.

- R2.4.2 Realiza salidas de recursos en las sedes en las que está autorizado. Puede autorizar salidas de material completando la información y documentación relacionada con la misma.
- R2.4.3 Puede modificar información sobre los recursos en las sedes en las que está autorizado. Los técnicos avanzados pueden realizar ciertas modificaciones sobre los activos en las sedes de los ámbitos en los que está autorizado.
- R2.4.4 Genera informes sobre el equipamiento de usuarios determinados. El técnico avanzado puede generar informes sobre el equipamiento asignado a un usuario concreto.
- **R2.4 Técnico**
 - R2.4.1 Puede modificar información sobre los recursos en las sedes en las que está autorizado. Los técnicos pueden realizar ciertas modificaciones sobre los activos en las sedes de los ámbitos en los que están autorizados.
 - R2.4.2 Genera informes sobre el equipamiento de usuarios determinados. Un técnico puede generar informes sobre el equipamiento asignado a un usuario concreto.
- **R2.5 Usuario**
 - R2.5.1 Puede consultar los recursos que tiene asignados. Los usuarios pueden visualizar el equipamiento que tienen asignado y generar un informe del mismo.
- **R2.6 Usuario sin autenticar**
 - R2.6.1 Puede acceder a información básica sobre activos
 - R2.6.2 Puede acceder a información básica sobre movimientos

2.3.2 Catálogo de requisitos módulo pp

En esta sección se enumeran los requisitos identificados para el módulo de gestión de adquisiciones.



Ilustración 10: Requisitos del módulo de adquisiciones

2.3.2.1 Funcionales

- **R1.1 Adquisiciones**

- *R1.1.1 Las adquisiciones se inician con una petición de oferta.* Los gestores de adquisiciones inician un proceso de compra solicitando una oferta concreta a un proveedor.
- *R1.1.2 A las peticiones de ofertas pueden responder varios proveedores.* Diversos proveedores pueden dar respuesta a una necesidad de compra por parte de la organización.
- *R1.1.3 Se debe poder detallar el contenido de una oferta, precio unitario, cantidad, etc.* Especificar completamente la oferta facilitará el conocimiento del detalle de las adquisiciones y de las futuras entradas de activos en el sistema originadas por la oferta de un proveedor.
- *R1.1.4 Se debe poder adjuntar documentación relativa a la oferta.* Con ello se persigue documentar las ofertas de los proveedores.
- *R1.1.5 Una adquisición se vinculará a una oferta concreta de un proveedor.* Cuando se realice una adquisición ésta debe quedar asociada a la oferta concreta del proveedor contra la que se lanzó.
- *R1.1.6 En el momento de crear la adquisición se copiarán todos los elementos de la oferta a la misma.* El detalle de la oferta (elementos, cantidad, precio unitario, etc.) se copiará al detalle de la adquisición, permitiéndose eliminar objetos de la oferta.
- *R1.1.7 Se debe poder adjuntar documentación relativa a la adquisición.* Con ello se persigue documentar las adquisiciones que se vayan realizando.

2.3.2.2 Usuario

- **R2.1 Gestor de adquisiciones**

- *R2.1.1 Gestiona marcas y modelos con sus atributos.* Los gestores de adquisiciones definen marcas y modelos del sistema y pueden establecer campos personalizados de estos últimos.
- *R2.1.2 Gestiona proveedores y sus contactos.* Los gestores de adquisiciones manejan los distintos proveedores del módulo, así como sus contactos.
- *R2.1.3 Solicita ofertas a los proveedores.* Los gestores de adquisiciones solicitan ofertas a sus distintos proveedores. Estas solicitudes deben ser registradas en la aplicación.
- *R2.1.4 Recibe ofertas a sus solicitudes.* Como resultado de las solicitudes comentadas en el punto anterior, los gestores recibirán diversas ofertas que deben ser registradas en la aplicación.
- *R2.1.5 Realiza adquisiciones contra una oferta de un proveedor.* Como culminación de este proceso que se comenta desde los dos puntos anteriores, el gestor de adquisiciones registra las compras que realiza contra una oferta de un proveedor.

- **R2.2 Usuario sin autenticar**

- *R2.2.1 Puede acceder a información básica sobre adquisiciones*



2.4 Modelo conceptual de datos

En esta sección se presenta y describe el modelo de datos del módulo de gestión de inventario.

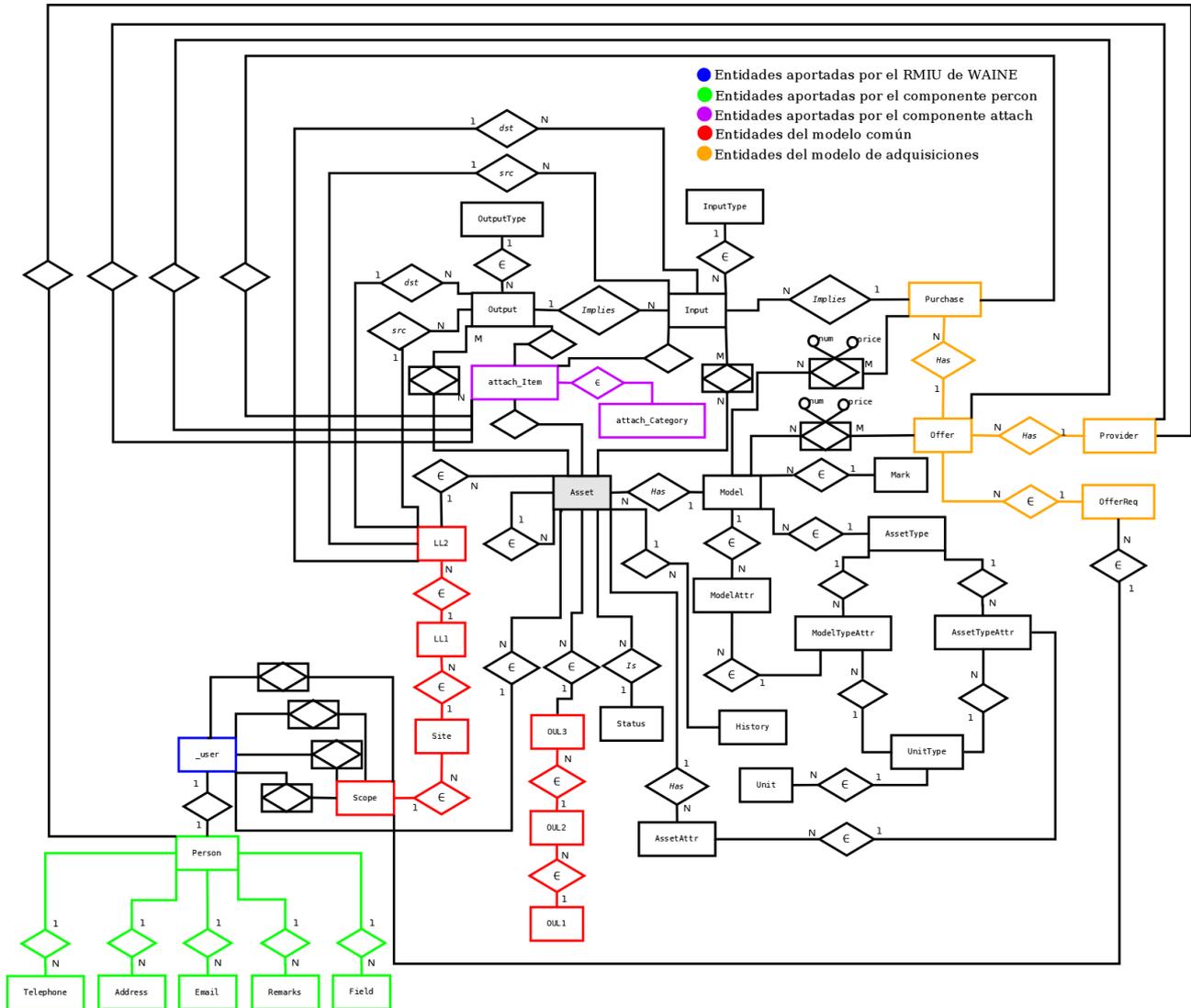


Ilustración 11: Diagrama entidad-relación

2.4.1 Entidades

En esta sección se describen las entidades del modelo de datos agrupadas según la funcionalidad que ofrecen.

2.4.1.1 Módulo común

A continuación, se enumeran las entidades del módulo común.

- **Scope:** Ámbito
 - *descr:* Descripción del ámbito
 - *rem:* Notas

- **Site:** Localización
 - *descr:* Descripción de la localización
 - *rem:* Notas
- **LL1:** Nivel de localización 1. Primer nivel en la jerarquía de localizaciones
 - *descr:* Descripción del nivel 1 de la localizaciones
 - *rem:* Notas
- **LL2:** Nivel de localización 2. Segundo nivel en la jerarquía de localizaciones
 - *descr:* Descripción del nivel 2 de localizaciones
 - *rem:* Notas
- **OUL1:** Nivel de unidades organizativas 1. Primer nivel en la jerarquía de unidades organizativas
 - *descr:* Descripción del nivel 1 de unidades organizativas
 - *rem:* Notas
- **OUL2:** Nivel de unidades organizativas 2. Segundo nivel en la jerarquía de unidades organizativas
 - *descr:* Descripción del nivel 2 de unidades organizativas
 - *rem:* Notas
- **OUL3:** Nivel de unidades organizativas 3. Tercer nivel en la jerarquía de unidades organizativas
 - *descr:* Descripción del nivel 3 de unidades organizativas
 - *rem:* Notas

2.4.1.2 Módulo de inventario

En este apartado se presentan las entidades y relaciones del módulo de inventario.

- **Asset:** Activo
 - *SN:* Número de serie
 - *ID:* Identificador. Existen elementos que además de disponer de un número de serie poseen un identificador único como por ejemplo un IMEI en un terminal móvil o el identificador de SIM. También a veces se pueden emplear una codificación interna de la organización.
 - *Responsable:* UID del usuario responsable del equipo



- *cun*: Nombre del usuario que dió de alta el equipo
- *run*: Nombre del usuario responsable del equipo
- *rem*: Notas

- **AssetType**: Tipo de recurso
 - *descr*: Descripción del tipo; SIM, celular, portátil, impresora laser, etc.
 - *rem*: Notas

- **Status**: Estado del activo. Al menos existirán los valores Activo y Baja. Estos dos no serán modificables ni eliminables.
 - *descr*: Descripción
 - *rem*: Notas

- **History**: Histórico del activo
 - *mts*: Instante de la modificación
 - *mun*: Nombre del usuario que realiza la modificación
 - *descr*: Descripción de la modificación. Por ejemplo:

Inserción	SN: 9999999 OU: Central → Contabilidad → Responsable: Juan España España
Modificación	Responsable: Juan Sevilla Sevilla

- **Mark**: Marca
 - *descr*: Descripción
 - *rem*: Notas

- **Model**: Modelo
 - *descr*: Descripción
 - *PN*: Part Number (número de modelo)
 - *rem*: Notas

2.4.1.3 Módulo de adquisiciones

En esta sección se especifican las entidades del módulo de adquisiciones.

- **Provider**: Proveedor

- *name*: Nombre del proveedor
- *tel*: Teléfono
- *email*: Dirección de correo electrónico
- *addr*: Dirección postal
- *rem*: Notas

- **OfferReq**: Solicitud de oferta
 - *descr*: Descripción del tipo, p.e.: SIM, celular, portátil, impresora laser, etc.
 - *rem*: Notas

- **Offer**: Oferta
 - *od*: Fecha de la oferta
 - *total*: Precio
 - *rem*: Notas

- **Purchase**: Adquisición
 - *pd*: Fecha de la adquisición
 - *wsd*: Fecha de inicio de garantía
 - *esd*: Fecha de fin de garantía
 - *pd*: Fecha de la adquisición
 - *CN*: Número de contrato
 - *rem*: Notas
 - *total*: Precio

2.4.1.4 Módulo de entradas y salidas del inventario de activos

Para finalizar la descripción de las entidades se describen aquellas que modelan las entradas y salidas de activos.

- **InputType**: Tipo de entrada
 - *descr*: Descripción

- **Input**: Entrada
 - *sun*: Nombre del usuario emisor
 - *run*: Nombre del usuario destinatario
 - *iun*: Nombre del usuario que da de alta la entrada



- *its*: Instante de la entrada
 - *src*: Origen
 - *esrc*: Origen externo (sólo para movimientos externos)
 - *dst*: Destino
 - *descr*: Descripción
 - *rem*: Notas
- **OutputType**: Tipo de salida
 - *descr*: Descripción
 - **Output**: Salida
 - *sun*: Nombre del usuario emisor
 - *run*: Nombre del usuario destinatario
 - *iu*: Nombre del usuario que da de alta la entrada
 - *ots*: Instante de la salida
 - *src*: Origen
 - *dst*: Destino
 - *edst*: Destino externo (sólo para movimientos externos)
 - *descr*: Descripción
 - *rem*: Notas

2.4.2 Relaciones

En este apartado se describen las relaciones más importantes del modelo conceptual de datos.

- **REL_gad_scope**: Relación entre un usuario (*_user*) y los ámbitos en los que está autorizado como gestor de adquisiciones
- **REL_gin_scope**: Relación entre un usuario (*_user*) y los ámbitos en los que está autorizado como gestor de inventario
- **REL_ate_scope**: Relación entre un usuario (*_user*) y los ámbitos en los que está autorizado como técnico avanzado
- **REL_tec_scope**: Relación entre un usuario (*_user*) y los ámbitos en los que está autorizado como técnico

- **REL_offer_model:** Relación entre una oferta y los modelos que comprende
 - *num*: Cantidad de elementos de cada modelo
 - *price*: Precio unitario

- **REL_purchase_model:** Relación entre una adquisición y los modelos que comprende
 - *num*: Cantidad de elementos de cada modelo
 - *price*: Precio unitario

- **REL_input_asset:** Relación entre una entrada y los activos que comprende

- **REL_output_asset:** Relación entre una salida y los activos que comprende

2.5 Interfaces de usuario y acciones

Las principales interfaces y acciones de los usuarios del sistema son las siguientes:

	Administrador del módulo común	Administrador del módulo de inventario	Gestor de adquisiciones	Gestor de inventario	Técnico avanzado	Técnico	Usuario regular	Usuario sin acceso
Gestión de ámbitos	✓							
Gestión de unidades organizativas	✓							
Gestión de localizaciones	✓							
Datos personales	✓							
Datos personales de usuarios	✓							
Administración de permisos		✓						
Tipos de entrada		✓						
Tipos de salida		✓						
Estados de los recursos		✓						
Tipos de unidades y unidades		✓						
Tipos de elementos y atributos		✓						
Gestión de proveedores			✓					



2 Análisis de los módulos de activos, adquisiciones y común de GTI

Gestión de ofertas			✓					
Gestión de adquisiciones			✓					
Marcas, modelos y atributos			✓	✓				
Gestión de entradas				✓	✓			
Gestión de salidas				✓	✓			
Gestión de inventario				✓				
Edición de activos				✓				
Gestión de activos				✓				
Estadísticas tipos por sede				✓				
Estadísticas Estado / Marcas por sede y tipo				✓				

	Administrador del módulo común	Administrador del módulo de inventario	Gestor de adquisiciones	Gestor de inventario	Técnico avanzado	Técnico	Usuario regular	Usuario sin acceso
Gestión de inventario reducida					✓	✓		
Edición de activos reducida					✓	✓		
Consulta de recursos asignados		✓	✓	✓	✓	✓	✓	
Visualización recurso								✓
Visualización adquisición								✓
Visualización de entrada								✓
Visualización de salida								✓

Tabla 1: Tabla rol/funcionalidad

En las siguientes secciones se pasa a describir en detalle estas interfaces de usuario.

2.5.1 Usuario administrador del módulo común

El administrador del módulo común configura y gestiona la información compartida por todos los módulos de GTI. las unidades de interacción que puede utilizar este usuario son las siguientes.

2.5.1.1 Gestión de ámbitos

En varios módulos de GTI se emplea el concepto de ámbito. Los ámbitos representan una unidad lógica de agrupación y pueden ser territoriales (Álava, León, Madrid) , departamentales (Contabilidad, Investigación, Personal) o cualquier otro conjunto de valores dependiendo de las necesidades de cada instancia de la aplicación.

Ámbitos

Descripción:	Notas:
Scope.descr (*)	Scope.rem Eliminar
Scope.descr (*)	Scope.rem Eliminar
:	:
Scope.descr (*)	Scope.rem Eliminar
Scope.descr (*)	Scope.rem Añadir

Ilustración 12: Unidad de interacción para gestión de ámbitos



2.5.1.2 Gestión de unidades organizativas

Esta unidad de interacción permite al usuario administrador del módulo común definir la jerarquía de unidades organizativas. Se empleará para ello la interfaz existente presentada en la sección 36 modificada añadiendo un check en el último nivel jerárquico para indicar si es ese nivel sigue activo o no.

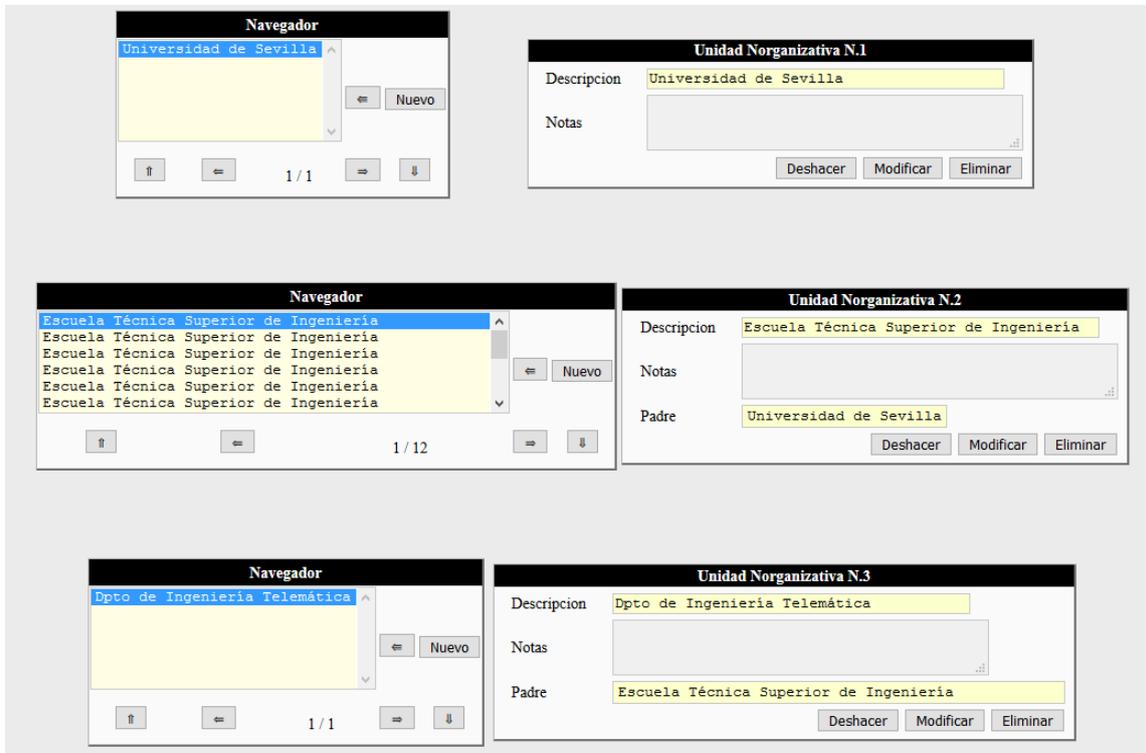


Ilustración 13: Interfaz de usuario final de unidades organizativas

2.5.1.3 Gestión de localizaciones

Esta interfaz permite describir las distintas sedes con sus localizaciones. Se empleará para ello una versión modificada de la interfaz existente presentada en la sección 35. En esta versión modificada los edificios deben poder asignarse a ámbitos y las localizaciones de nivel 2 incluirán un par de checks: uno para indicar si esa localización debe ser considerada un almacén y otro para indicar si sigue activa.

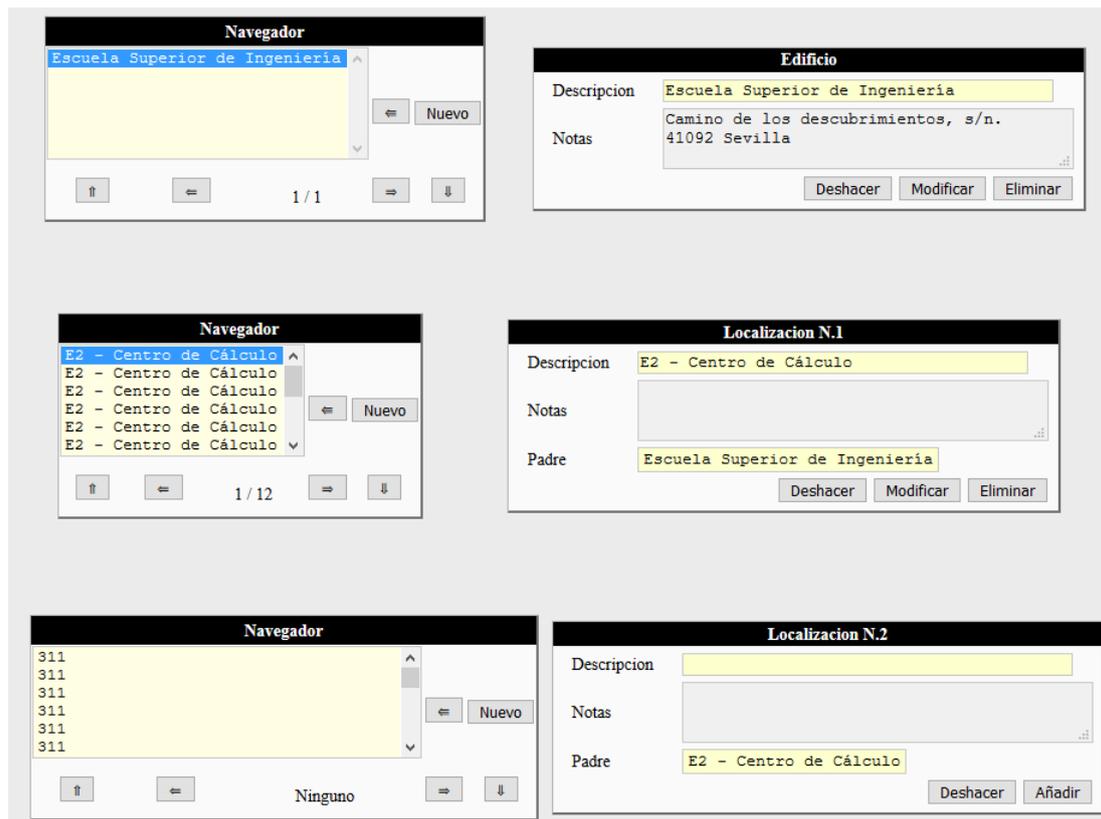


Ilustración 14: Interfaz de usuario final de sedes y localizaciones

2.5.2 Usuario administrador del módulo de inventario

Este usuario es el encargado de configurar el módulo de inventario. En las siguientes secciones se describen las principales unidades de interacción que el administrador del módulo de inventario emplea en su trabajo habitual.

2.5.2.1 Administración de permisos

La interfaz de usuario que se muestra a continuación permite al administrador del módulo gestionar para cada ámbito quiénes son los usuario gestores y quiénes son los técnicos que trabajan en ella.



Ilustración 15: Unidad de interacción para la administración de permisos

2.5.2.2 Tipos de entrada

Este formulario permite al administrador del módulo definir los tipos de entrada que se permitirán en el sistema. Existen tipos predefinidos que no podrán ser eliminados. Los tipos predefinidos son:

1. Adquisición
2. Interna
3. Devolución

Ilustración 16: Unidad de interacción tipos de entrada

2.5.2.3 Tipos de salida

Este formulario permite al administrador del módulo definir los tipos de salida que se permitirán en el sistema. Existen tipos predefinidos que no podrán ser eliminados. Son los siguientes:

1. Interna
2. Préstamo
3. Baja

Tipos de salidas

OutputType.descr	Eliminar
OutputType.descr	Eliminar
:	:
OutputType.descr	Eliminar
OutputType.descr	Añadir

Ilustración 17: Unidad de interacción tipos de entrada

2.5.2.4 Estados de los recursos

Con esta interfaz el administrador del módulo puede establecer cuáles son los diferentes estados en los que se puede encontrar un determinado recurso inventariable. Existen estados que no pueden ser eliminados. Son los siguientes:

1. Activo
2. Baja

Estado de los recursos

Status.descr	Eliminar
Status.descr	Eliminar
:	:
Status.descr	Eliminar
Status.descr	Añadir

Ilustración 18: Unidad de interacción estados de los recursos

2.5.2.5 Tipos de unidades y unidades

Esta unidad de interacción permite definir cuáles son las unidades admitidas para los distintos atributos de los elementos inventariables. Se trata de una interfaz que ya se ha presentado en la sección 37

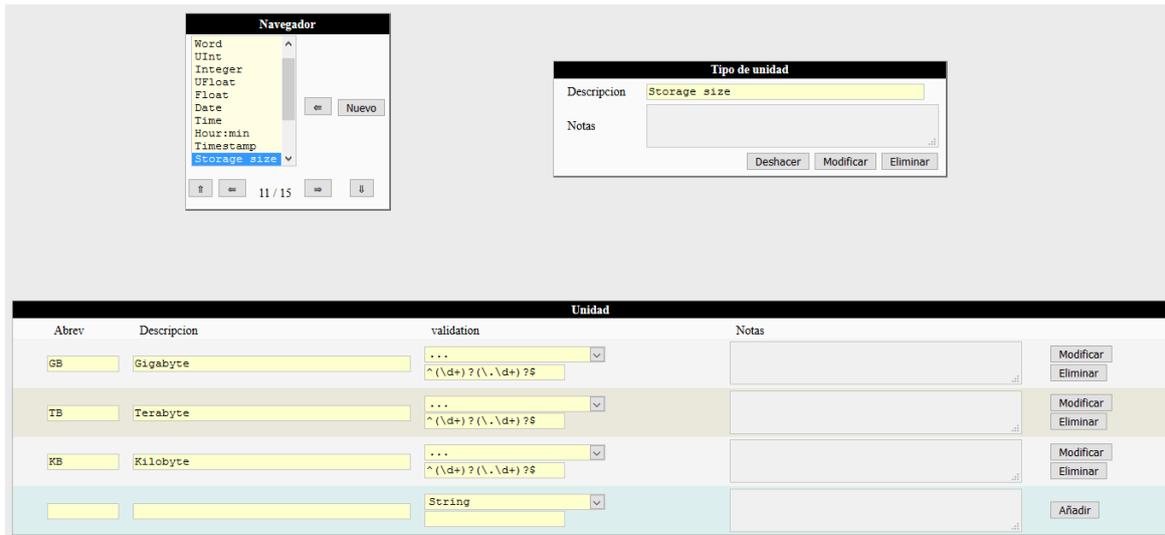


Ilustración 19: Interfaz de usuario final de tipos de unidades y unidades

2.5.2.6 Tipos de elemento y atributos

A través de esta interfaz el administrador puede establecer cuáles son los atributos que aplican a un tipo de elemento inventariable. Esta unidad de interacción se presentó previamente en la sección 39

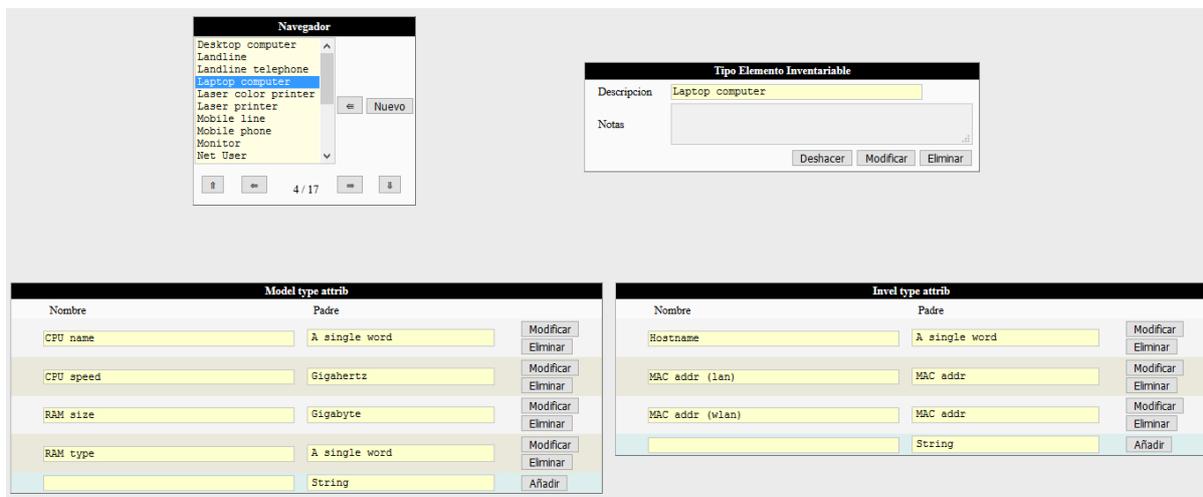


Ilustración 20: Interfaz de usuario final de Tipos de elementos y atributos

2.5.3 Gestor de adquisiciones

Se especifican a continuación las unidades de interacción propias del perfil gestor de adquisiciones.

2.5.3.1 Gestión de proveedores

La gestión de proveedores habilita al gestor de adquisiciones para insertar toda la información relevante de un proveedor, así como del personal de contacto del mismo.

Provider.name

Nombre: Provider.name

Teléfono: Provider.tel

Email: Provider.email

Dirección: Provider.addr

Notas: Provider.rem

Modificar Eliminar

Contactos Adjuntos

Nombre

Person.surname + ' ' + Person.name ▼ Modificar Ver

Person.surname + ' ' + Person.name ▼ Modificar Ver

Person.surname + ' ' + Person.name ▼ Añadir

Ilustración 21: Unidad de interacción de gestión de proveedores

2.5.3.2 Gestión de ofertas

La interfaz de gestión de ofertas permite dar de alta solicitudes de ofertas por parte del gestor de adquisiciones, así como vincularle las ofertas recibidas por parte de uno o varios proveedores. También permite definir los elementos ofertados por el proveedor en su número y precio.

OfferReq.descr

Descripción: OfferReq.descr

Ámbito: Scope.descr ▼

Notas: OfferReq.rem

Modificar Eliminar

Offer.odate Offer.prov

Proveedor: Offer.prov ▼

Fecha: Offer.odate

Notas: Offer.rem

Total: Offer.price

Modificar Eliminar

Elementos Adjuntos

Modelo	Cantidad	Precio Unitario	Total	
Mark.descr - Model.descr ▼	Offer.num	Offer.price	f)=price*num	Modificar
Mark.descr - Model.descr ▼	Offer.num	Offer.price	f)=price*num	Modificar
:	:	:	:	:
Mark.descr - Model.descr ▼	Offer.num	Offer.price	f)=price*num	Añadir
			f)=Σ total	

Ilustración 22: Unidad de interacción de gestión de ofertas



2.5.3.3 Gestión de adquisiciones

La gestión de adquisiciones permite registrar las compras que se realizarán contra una oferta de un proveedor. Si existen datos de elementos (modelos, cantidad, precios, etc.) en la oferta se copiarán a la adquisición.

Solicitud

Oferta

Fecha	<input type="text" value="Purchase.pd"/>
I. Garantía	<input type="text" value="Purchase.wsd"/>
F. Garantía	<input type="text" value="Purchase.wed"/>
CN	<input type="text" value="Purchase.cn"/>
Notas:	<input type="text" value="Purchase.rem"/>
Total:	<input type="text" value="Purchase.price"/>

Elementos	Adjuntos			
Modelo	Cantidad	Precio Unitario	Total	
<input type="text" value="Mark.descr - Model.descr"/>	<input type="text" value="Offer.num"/>	<input type="text" value="Offer.price"/>	<input type="text" value="f()=price*num"/>	<input type="button" value="Modificar"/>
<input type="text" value="Mark.descr - Model.descr"/>	<input type="text" value="Offer.num"/>	<input type="text" value="Offer.price"/>	<input type="text" value="f()=price*num"/>	<input type="button" value="Modificar"/>
:	:	:	:	
<input type="text" value="Mark.descr - Model.descr"/>	<input type="text" value="Offer.num"/>	<input type="text" value="Offer.price"/>	<input type="text" value="f()=price*num"/>	<input type="button" value="Añadir"/>
			<input type="text" value="f()=Σ total"/>	

Ilustración 23: Unidad de interacción de gestión de adquisiciones

2.5.3.4 Marcas, modelos y atributos

Esta unidad de interacción permite administrar las distintas marcas y modelos existentes, así como establecer los atributos propios a cada uno de los modelos según se definió en la unidad de interacción que se muestra en la sección 2.2.4.

Navegador

Acer
Apple
Asus
Dell
Fujitsu
HP
Hitachi
Huawei

3 / 14

Marca

Descripcion: Asus

Notas

Deshacer Modificar Eliminar

Navegador

X554L

1 / 1

Modelo

Tipo: Laptop computer

Nombre: X554L

PN: 234234234

Notas

Deshacer Modificar Eliminar

Atributos

name	Valor	Unidad	
CPU name	Intel i5	Word	Modificar
CPU speed	2.4	Ghz	Modificar
RAM size	4	GB	Modificar
RAM type	DDR3	Word	Modificar

Ilustración 24: Interfaz de usuario final de marcas, modelos y atributos

2.5.4 Gestor de inventario

Esta sección recoge las unidades de interacción propias del usuario gestor de inventario.

2.5.4.1 Gestión de entradas.

La gestión de entradas permite gestionar las entradas de material inventariable en la organización.

El único campo destacable del informe es el campo etiquetado como *Relacionado* que indica el movimiento con el que estaría relacionada la entrada, por ejemplo "Préstamo 34213" o "Adquisición 94721".

2.5.4.2 Gestión de salidas

La gestión de salidas permite administrar las salidas de material inventariable en la organización.

Filtros básicos Filtros avanzados

Origen: Output:src: gti.Site -> gti.LL1 -> gti.LL2 (*) ID Salida: Output.pk Descripción: Output.descr Tipo: Output.t

Salida

Información Notas Detalles

ID: Output.pk Descripción: Output.descr (*) Tipo: Output.t Instante: Output.iots Emisor: Output.sun (*) Receptor: Output.nun (*) Origen: Output:src: gti.Site -> gti.LL1 -> gti.LL2 (*) Destino ext: Output.edst Destino: Output:dst: gti.Site -> gti.LL1 -> gti.LL2 (*)

Imprimir Insertar Modificar

Activos Resumen Adjuntos

NS - ID - Marca - Modelo

Asset.NS - Asset.ID - Mark.descr - Model.descr	Eliminar	Ver
Asset.NS - Asset.ID - Mark.descr - Model.descr	Eliminar	Ver
Asset.fkmodel: AssetType.descr - Mark.descr - Model.descr	Añadir	Ver

f()=Count pk

Ilustración 27: Unidad de interacción para la gestión de salidas

Desde esta unidad de interacción se puede generar el informe de salida asociado al movimiento de material. Este informe está ordenado por Tipo, Marca, Modelo y Número de Serie.

Activo

Información Detalles

ID

NS

IDR

Estado ▼

Tipo ▼

Modelo ▼

UO ▼

Localización ▼

Responsable ▼

Padre ▼

Informe Modificar

propiedades Recursos Hijo Histórico Adjuntos

Nombre	Valor	Unidad	
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un	Modificar
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un	Modificar
:	:	:	
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un	Modificar

Ilustración 30: Unidad de interacción para la edición de activos

Desde esta unidad de interacción se puede generar el informe del activo.

2.5.4.4.3 Pestaña Traza

Esta pestaña muestra una traza cronológica de las modificaciones realizadas sobre el elemento inventariado. Estas entradas se realizan automáticamente sin intervención del usuario.

Fecha y hora	Usuario	Descripción
History.mts	History.mun	History.descr
:	:	:
History.mts	History.mun	History.descr

Ilustración 34: Pestaña traza

2.5.4.4.4 Pestaña Adjuntos

La pestaña adjuntos posibilita adjuntar documentos a un activo. Se hará uso en este caso del formulario `attach.form.catattach` del paquete `attach` (ver sección 6.10).

Adjuntos				
Descripcion	Archivo	Tipo	Creador por	Modificado por
	Descargar <input type="checkbox"/> Borrar <input type="button" value="Examinar..."/> No se ha seleccionado ningún archivo.	pm.proj	admproyl	admproyl
				<input type="button" value="Añadir"/>

Ilustración 35: Pestaña Adjuntos

2.5.4.5 Gestión de activos

La interfaz de usuarios de *gestión de activos* permite al gestor del inventario una gestión completa de los activos en diversas localizaciones incluyendo la funcionalidad de inserción y eliminación de activos.



Ámbito

Localización

Activo

Información Detalles

ID

NS

IDR

Estado

Tipo

Modelo

UO

Localización

Responsable

Padre

Informe Modificar Eliminar

propiedades Recursos Hijo Histórico Adjuntos

Nombre	Valor	Unidad	
<input type="text" value="AssetAttr.fkatattr: AssetTypeAttr.name"/>	<input type="text" value="AssetAttr.val"/>	<input type="text" value="AssetAttr.fkud: Unit.un"/>	<input type="button" value="Modificar"/>
<input type="text" value="AssetAttr.fkatattr: AssetTypeAttr.name"/>	<input type="text" value="AssetAttr.val"/>	<input type="text" value="AssetAttr.fkud: Unit.un"/>	<input type="button" value="Modificar"/>
:	:	:	
<input type="text" value="AssetAttr.fkatattr: AssetTypeAttr.name"/>	<input type="text" value="AssetAttr.val"/>	<input type="text" value="AssetAttr.fkud: Unit.un"/>	<input type="button" value="Modificar"/>

Ilustración 36: Unidad de interacción para la gestión de activos

Desde esta unidad de interacción se puede generar el informe de Activo.

2.5.5 Técnico Avanzado

Los técnicos avanzados tienen la capacidad de realizar entradas y salidas de activos con unas interfaces de usuario idéntica a las mostradas en las secciones 5.4.1 y 5.4.2. Pero además tienen a su disposición las siguientes interfaces de usuario.

2.5.5.1 Gestión de inventario reducida

La *gestión de inventario reducida* es una versión limitada de la interfaz de usuario *gestión de inventario* (descrita en la sección 5.4.3). En este formulario se impide la modificación del tipo, modelo, número de serie e identificador del activo.

Filtros basicos		Filtros avanzados						
Localización	OU	Tipo de recurso	Marca y Modelo	Responsable	NS	IDR		
gti.Scope → gti.Site → gti.LL1 → gti.LL2 (*)	gti.OUL1 → gti.OUL2 → gti.OUL3	AssetType.descr	Mark.descr → Model.descr	_user.name	Asset.SN	Asset.ID		
gti.Scope → gti.Site → gti.LL1 → gti.LL2 (*)	gti.OUL1 → gti.OUL2 → gti.OUL3	AssetType.descr	Mark.descr → Model.descr	_user.name	Asset.SN	Asset.ID	Ver	Modificar
gti.Scope → gti.Site → gti.LL1 → gti.LL2 (*)	gti.OUL1 → gti.OUL2 → gti.OUL3	AssetType.descr	Mark.descr → Model.descr	_user.name	Asset.SN	Asset.ID	Ver	Modificar
gti.Scope → gti.Site → gti.LL1 → gti.LL2 (*)	gti.OUL1 → gti.OUL2 → gti.OUL3	AssetType.descr	Mark.descr → Model.descr	_user.name	Asset.SN	Asset.ID	Ver	Modificar

f()=Count

Ilustración 37: Unidad de interacción de gestión de inventario reducida

Desde esta unidad de interacción se puede acceder a la interfaz *edición de activos reducida*.

2.5.5.2 Edición de activos reducida

Esta unidad de interacción es una versión limitada de la interfaz *edición de activos* (presentada en la sección 5.4.4). Las restricciones en este formulario impiden la modificación del tipo, modelo, número de serie e identificador de un activo.

Activo

Información Detalles

ID: Asset.pk

NS: Asset.SN

IDR: Asset.ID

Estado: Asset.fks: Status.descr

Tipo: Asset.fkt: AssetType.descr

Modelo: Asset.fkmodel: Mark.descr → Model.descr

UO: Asset.fkou3: gti.OUL1 → gti.OUL2 → gti.OUL3

Localización: Asset.fkll2: gti.Scope → gti.Site → gti.LL1 → gti.LL2

Responsable: Asset.run

Padre: Asset.fkasset: Asset.SN

Informe
Modificar

propiedades Recursos Hijo Histórico Traza Adjuntos

Nombre	Valor	Unidad
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un

Ilustración 38: Unidad de interacción para la edición de activos reducida

2.5.6 Técnico

Los usuarios con perfil técnico no tienen, a diferencia de los que poseen perfil técnico avanzado (ver sección 5.5), la capacidad de realizar entradas y salidas de activos. Sin embargo, comparten con el perfil técnico avanzado las interfaces de usuario *gestión de inventario reducida* (sección 5.5.1) y *edición de activos reducida* (sección 5.5.2). Hay que tener en cuenta que se trata de interfaces similares, pero en las que los filtros deben actuar limitando la visibilidad a aquellos ámbitos, etc. a los que se ha dado permiso al técnico.



2.5.7 Usuario regular

Los usuarios regulares, así como el resto de usuarios del módulo de inventario tienen a su disposición la posibilidad de conocer los recursos que tienen asignados.

2.5.7.1 Consulta de recursos asignados

Esta consulta permite visualizar los elementos "padre" inventariados en el sistema y asignados al usuario actual.

Activo

Información
Detalles

ID	Asset.pk
NS	Asset.SN
IDR	Asset.ID
Estado	Asset.fks: Status.descr ▼
Tipo	Asset.fkt: AssetType.descr ▼
Modelo	Asset.fkmodel: Mark.descr → Model.descr ▼
UO	Asset.fkou3: gti.OUL1 → gti.OUL2 → gti.OUL3 ▼
Localización	Asset.fkll2: gti.Scope → gti.Site → gti.LL1 → gti.LL2 ▼
Responsable	Asset.run ▼
Padre	Asset.fkasset: Asset.SN ▼

propiedades
Recursos Hijo

Nombre	Valor	Unidad
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un ▼
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un ▼
:	:	:
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un ▼

Ilustración 39: Unidad de interacción consulta de recursos asignados

Desde la pestaña de recursos hijos el usuario puede navegar por los hijos de elemento actual pero siempre utilizando esta misma unidad de interacción.

2.5.8 Usuario sin acceso

Como en otros módulos de la aplicación GTI, existe la posibilidad de que usuarios de otros módulos puedan acceder a información del módulo de inventario sin ser usuario del mismo. La interfaces que se comentan a continuación tienen esta finalidad.

2.5.8.1 Visualización de recurso

Este formulario permite una visualización de la información mínima sobre un elemento inventariable.

The image shows a web form titled "Activo". It contains several input fields and dropdown menus. Below the main form is a section titled "Propiedades" with three columns: "Nombre", "Valor", and "Unidad".

Activo		
ID	Asset.pk	
NS	Asset.SN	
IDR	Asset.ID	
Estado	Asset.fks: Status.descr ▼	
Tipo	Asset.fkt: AssetType.descr ▼	
Modelo	Asset.fkmodel: Mark.descr → Model.descr ▼	

Propiedades		
Nombre	Valor	Unidad
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un ▼
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un ▼
:	:	:
AssetAttr.fkatattr: AssetTypeAttr.name	AssetAttr.val	AssetAttr.fkud: Unit.un ▼

Ilustración 40: Unidad de interacción para la visualización de recursos

2.5.8.2 Visualización de adquisición

La unidad de interacción *visualización de adquisición* ofrece la información básica sobre una adquisición.

The image shows a web form titled "Adquisición". It contains four input fields.

Adquisición	
Fecha	Purchase.pd
I. Garantía	Purchase.wsd
F. Garantía	Purchase.wed
CN	Purchase.cn

Ilustración 41: Unidad de interacción para la visualización de adquisiciones



2.5.8.3 Visualización de entrada

El formulario para la visualización de entradas habilita a los usuarios sin acceso a tener una visión mínima sobre los datos de una entrada.

Entrada	
Descripción	Input.descr (*)
Tipo	Input.t: InputType.descr ▼
Instante	Input.its
Emisor	Input.sun (*)
Receptor	Input.run (*) ▼
Origen	Input.src: gti.Site → gti.LL1 → gti.LL2 ▼
Destino	Input.dst: gti.Site → gti.LL1 → gti.LL2 (*) ▼

Ilustración 42: Unidad de interacción para la visualización de entradas

2.5.8.4 Visualización de salida

Este contenedor posibilita el acceso a los datos limitados sobre una salida a los usuarios de otros módulos.

Salida	
Descripción	Output.descr (*)
Tipo	Output.t: OutputType.descr ▼
Instante	Output.iots
Emisor	Output.sun (*)
Receptor	Output.run (*) ▼
Origen	Output.src: gti.Site → gti.LL1 → gti.LL2 (*) ▼
Destino	Output.dst: gti.Site → gti.LL1 → gti.LL2 ▼

Ilustración 43: Unidad de interacción para la visualización de salidas

2.6 Anexos

2.6.1 Normas de codificación SQL

A la hora de codificar SQL se cumplirán las siguientes normas:

Esquemas

El identificador de un esquema debe ser breve, normalmente se empleará algún tipo de acrónimo. Se antepondrá siempre al identificador de cualquier objeto que pertenezca al esquema.

ejemplos:

```
am ps
```

Tablas

Los identificadores para las tablas estarán formados por letras y números, pero comenzarán siempre por una letra mayúscula. Debe ser un identificador breve pero que permita identificar claramente a la entidad que representa. Si el identificador es una composición de varias palabras ambas deben comenzar por mayúsculas.

ejemplos:

```
am.Input pm.ProjType
```

Tablas de relación

Las tablas que sintetizan una relación tendrán un identificador especial. Este identificador comenzará por la palabra **REL** y le seguirán los nombres de las tablas que relaciona en orden alfabético y todo ello estará separado por el caracter _

ejemplos:

```
ps.REL_Group_Serv am.REL_Asset_Input
```

Vistas

El identificador de las vistas comenzará por la palabra **VIEW**, le seguirá un caracter separador _ y finalizará un nombre descriptivo de la vista con todos sus caracteres en minúsculas

ejemplos:

```
ps.VIEW_fullproc gti.VIEW_ou
```

Procedimientos y funciones

Su identificador comenzará por la palabra **PROC** o **FUNC**, seguido de un caracter separador _ y finalizará un nombre descriptivo siguiendo la norma del id de las tablas.

ejemplos:

```
ps.PROC_NewYear gti.FUNC_Value
```

2.6.2 Normas de codificación ASL

En el código ASL se seguirán las siguientes convenciones:

Contenedores y formularios



El identificador de un formulario se compondrá de la siguiente manera:

- acrónimo de la aplicación
- caracter separador.
- acrónimo del módulo
- caracter separador.
- caracter identificador del tipo de objeto: **c** contenedor, **f** formulario
- identificador del elemento

ejemplos:

```
gti.pm.fproc
```

```
gti.am.cinput
```

Cuando se trate de un elemento del módulo común se omitirá el acrónimo del módulo

ejemplo:

```
gti.fentry
```

```
gti.csite
```

2.6.3 Código SQL de la base de datos del prototipo inicial

```
-- ===== --
-- Units --
-- ===== --

DROP TABLE Unit_type CASCADE;

CREATE TABLE Unit_type (
  pk serial PRIMARY KEY ,
  descr character varying(80) NOT NULL,
  remarks text
);

DROP TABLE Unit CASCADE;

CREATE TABLE Unit(
  pk serial PRIMARY KEY ,
  un character varying(10) NOT NULL,
  descr character varying(80) NOT NULL,
  reval character varying(80) NOT NULL, -- One of regex or free
  remarks text,
  fk integer NOT NULL REFERENCES Unit_type(pk)
);

DROP TABLE regex CASCADE;

CREATE TABLE regex(
  descr varchar(80) NOT NULL,
  code varchar(80) NOT NULL
);
```

```

-- ===== --
-- Invel_type --
-- ===== --

DROP TABLE Invel_type CASCADE;

CREATE TABLE Invel_type (
  pk serial PRIMARY KEY ,
  descr varchar(80) NOT NULL,
  remarks text
);

DROP TABLE Model_type_attr CASCADE;

CREATE TABLE Model_type_attr (
  pk serial PRIMARY KEY ,
  name varchar(80) NOT NULL,
  fktype integer NOT NULL REFERENCES Invel_type(pk) ON UPDATE CASCADE,
  fkudtype integer NOT NULL REFERENCES Unit_type(pk) ON UPDATE CASCADE
);

CREATE OR REPLACE FUNCTION tproc_model_type_attr_ains() RETURNS TRIGGER
AS '
BEGIN
INSERT INTO Model_attr(fk,fkitattr,fkud)
SELECT pk, NEW.pk, 1 from Model where fktype=NEW.fktype;
RETURN NEW;
END;
' LANGUAGE plpgsql;

CREATE TRIGGER TPROC_model_type_attr_ains AFTER INSERT ON Model_type_attr
FOR EACH ROW EXECUTE PROCEDURE TPROC_model_type_attr_ains();

DROP TABLE Invel_type_attr CASCADE;

CREATE TABLE Invel_type_attr (
  pk serial PRIMARY KEY ,
  name varchar(80) NOT NULL,
  fktype integer NOT NULL REFERENCES Invel_type(pk) ON UPDATE CASCADE,
  fkudtype integer NOT NULL REFERENCES Unit_type(pk) ON UPDATE CASCADE
);

CREATE OR REPLACE FUNCTION tproc_invel_type_attr_ains() RETURNS TRIGGER
AS '
BEGIN
INSERT INTO Invel_attr(fk,fkitattr,fkud)
SELECT pk, NEW.pk, 1 from Invel where fkmodel IN (Select pk from Model where fktype=NEW.fktype);
RETURN NEW;
END;
' LANGUAGE plpgsql;

CREATE TRIGGER TPROC_invel_type_attr_ains AFTER INSERT ON Invel_type_attr
FOR EACH ROW EXECUTE PROCEDURE TPROC_invel_type_attr_ains();

-- ===== --
-- Marks --

```



```
-- ===== --
```

```
DROP TABLE Mark CASCADE;
```

```
CREATE TABLE Mark (  
  pk serial PRIMARY KEY ,  
  descr varchar(80) NOT NULL,  
  remarks text  
);
```

```
DROP TABLE Model CASCADE;
```

```
CREATE TABLE Model (  
  pk serial PRIMARY KEY ,  
  descr varchar(80) NOT NULL,  
  pn varchar(80) NOT NULL,  
  remarks text,  
  fktype integer NOT NULL REFERENCES Invel_type(pk) ON UPDATE CASCADE,  
  fk integer NOT NULL REFERENCES Mark(pk) ON UPDATE CASCADE  
);
```

```
DROP TABLE Model_attr CASCADE;
```

```
CREATE TABLE Model_attr (  
  pk serial NOT NULL,  
  val character varying(80),  
  fk integer NOT NULL REFERENCES Model(pk) ON UPDATE CASCADE ON DELETE CASCADE,  
  fkitattr integer NOT NULL REFERENCES Model_type_attr(pk) ON UPDATE CASCADE ON DELETE CASCADE,  
  fkud integer  
);
```

```
CREATE OR REPLACE FUNCTION tproc_model_ains() RETURNS TRIGGER  
AS '  
BEGIN  
INSERT INTO Model_attr(fk,fkitattr,fkud)  
SELECT NEW.pk, pk, 1 from Model_type_attr where fktype=NEW.fktype;  
RETURN NEW;  
END;  
' LANGUAGE plpgsql;
```

```
CREATE TRIGGER TPROC_model_ains AFTER INSERT ON Model FOR EACH ROW EXECUTE PROCEDURE  
TPROC_model_ains();
```

```
DROP VIEW VIEW_model CASCADE;
```

```
CREATE VIEW VIEW_model AS  
  SELECT Model.pk,  
    Model.fk,  
    Model.fktype,  
    Mark.descr || ' - ' ||  
    Model.descr AS descr  
  FROM Mark, Model  
  WHERE Model.fk=Mark.pk;
```

```
-- ===== --
```

```
-- Invel --
```

```
-- ===== --
```

```
DROP TABLE Status CASCADE;
```

```
CREATE TABLE Status (  
  pk serial PRIMARY KEY ,  
  descr character varying(80) NOT NULL  
);
```

```
DROP TABLE Invel CASCADE;
```

```
CREATE TABLE Invel (  
  pk bigserial PRIMARY KEY,  
  cts timestamp DEFAULT Now(),  
  lmts timestamp DEFAULT Now(),  
  lmuname varchar(20) NOT NULL,  
  sn character varying(80) NOT NULL,  
  id character varying(80),  
  descr character varying(80),  
  remarks text,  
  fkperson bigint,  
  fkinvel bigint REFERENCES invel(pk),  
  fkmodel integer NOT NULL REFERENCES Model(pk),  
  fkstat integer NOT NULL REFERENCES Status(pk),  
  fkloc integer REFERENCES gti.LL2(pk)  
);
```

```
DROP TABLE Invel_attr CASCADE;
```

```
CREATE TABLE invel_attr (  
  pk bigserial NOT NULL,  
  val character varying(80),  
  fk bigint NOT NULL REFERENCES Invel(pk) ON UPDATE CASCADE ON DELETE CASCADE,  
  fkitattr integer NOT NULL REFERENCES Invel_type_attr(pk) ON UPDATE CASCADE,  
  fkud integer NOT NULL  
);
```

```
CREATE OR REPLACE FUNCTION tproc_invel_ains() RETURNS TRIGGER  
AS '  
BEGIN  
INSERT INTO Invel_attr(fk,fkitattr,fkud)  
SELECT NEW.pk, pk, 1 from Invel_type_attr WHERE fktype IN (SELECT fktype from Model WHERE pk=NEW.fkmodel);  
RETURN NEW;  
END;  
' LANGUAGE plpgsql;
```

```
CREATE TRIGGER TPROC_invel_ains AFTER INSERT ON Invel  
FOR EACH ROW EXECUTE PROCEDURE TPROC_invel_ains();
```

2.6.4 Código ASL del prototipo inicial

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<!DOCTYPE asl PUBLIC "-//ITI//DTD XWF 0.6 //EN"  
  "/usr/local/lib/waine-0.4.3/lib/asl.dtd"  
>
```



```
<asl>
```

```
<include href="/usr/local/lib/waine-0.4.3/include/meta.asl"/>
```

```
<head>
```

```
<meta class="AppInfo" name="appname" value="gedeti.incid"/>
```

```
<meta class="AppInfo" name="ver" value="0.1"/>
```

```
<meta class="AppInfo" name="abstract" value="Destión de incidencias"/>
```

```
<meta class="AppInfo" name="author" value="Antonio Delgado"/>
```

```
<meta class="AppInfo" name="email" value="aldelgado@us.es"/>
```

```
<meta class="AppInfo" name="org" value="Area de telematica"/>
```

```
<meta class="AppInfo" name="date" value="2012"/>
```

```
</head>
```

```
<!-- Grupo de administradores -->
```

```
<group gid="0" name="root">
```

```
<doc>Administration group</doc>
```

```
<user uid="0" name="root" passwd="root" mainid="main_adm"
```

```
descr="Administrator"/>
```

```
</group>
```

```
<!-- ===== -->
```

```
<!-- Elementos reutilizados -->
```

```
<!-- ===== -->
```

```
<form id="frm_rel_nn" caption="XXXX">
```

```
<fields>
```

```
<key source="pk"/>
```

```
<string source="fk1" caption="en=Father|es=Padre" len="0"
```

```
canbenull="N" msg="Seleccione"/>
```

```
<string source="fk2" caption="en=Father|es=Padre" len="0"
```

```
canbenull="N" msg="Seleccione"/>
```

```
<string source="fk3" caption="en=Father|es=Padre" len="0"
```

```
canbenull="N" msg="Seleccione"/>
```

```
<string source="fk4" caption="en=Father|es=Padre" len="0"
```

```
canbenull="N" msg="Seleccione"/>
```

```
</fields>
```

```
</form>
```

```
<form id="frm_fk_descr" caption="XXXX">
```

```
<orderby>descr</orderby>
```

```
<fields>
```

```
<key source="pk"/>
```

```
<string source="descr" caption="es=Descripcion|en=Description"
```

```
len="40" maxlen="80" />
```

```
<text source="remarks" caption="en=Remarks|es=Notas" canbenull="Y"
```

```
tooltip="Indicaciones">
```

```
<widget>fstextbox</widget>
```

```
<width>40</width>
```

```
<height>2</height>
```

```
</text>
```

```
<string source="fk" caption="en=Father|es=Padre" len="0"
```

```
canbenull="N" msg="Seleccione">
```

```
<search>XXX</search>
```

```
<searchfield>pk,descr</searchfield>
```

```
</string>
```

```
</fields>
```

```
</form>
```

```
<!-- ===== -->
```

```
<!-- Locations -->
```

```
<!-- ===== -->
```

```
<struct id="st_site" type="relation">  
  <param name="form_split" value="rows=33%,*"/>  
  
  <param name="formid" value="frm_fk_descr"/>  
  <param name="form_source" value="Site"/>  
  <param name="form_type" value="form"/>  
  <param name="navigator_position" value="W"/>  
  <param name="navigator_fields" value="1"/>  
  <param name="navigator_size" value="6"/>  
  <param name="button_data" value="1"/>  
  <param name="form_caption" value="en=Site|es=Edificio"/>  
  <param name="fields_remove" value="3"/>
```

```
  <param ord="2" name="structid" value="st_site_n"/>  
</struct>
```

```
<struct id="st_site_n" type="split">  
  <param name="form_split" value="rows=50%,*"/>  
  
  <param name="formid" value="frm_fk_descr"/>  
  <param name="form_source" value="Location_lv1"/>  
  <param name="form_type" value="form"/>  
  <param name="form_caption" value="en=Location Lv1|  
    es=Localizacion N.1"/>  
  <param name="navigator_position" value="W"/>  
  <param name="navigator_fields" value="1"/>  
  <param name="navigator_size" value="6"/>  
  <param name="button_data" value="1"/>  
  <param name="fields_modify[0]" value="search#DATA:Site;;;#3"/>  
  <param name="source_filter_field" value="fk"/>
```

```
  <param ord="2" name="formid" value="frm_fk_descr"/>  
  <param ord="2" name="form_source" value="Location_lv2"/>  
  <param ord="2" name="form_type" value="form"/>  
  <param ord="2" name="form_caption" value="en=Location Lv1|  
    es=Localizacion N.2"/>  
  <param ord="2" name="navigator_position" value="W"/>  
  <param ord="2" name="navigator_fields" value="1"/>  
  <param ord="2" name="navigator_size" value="6"/>  
  <param ord="2" name="button_data" value="1"/>  
  <param ord="2" name="fields_modify[0]"  
    value="search#DATA:Location_lv1;;;#3"/>  
  <param ord="2" name="source_filter_field" value="fk"/>  
</struct>
```

```
<!-- ===== -->
```

```
<!-- Organizational units -->
```

```
<!-- ===== -->
```

```
<struct id="st_ous" type="relation">  
  <param name="form_split" value="rows=33%,*"/>
```



```

<param name="formid" value="frm_fk_descr"/>
<param name="form_source" value="OU_lv1"/>
<param name="form_type" value="form"/>
<param name="navigator_position" value="W"/>
<param name="navigator_fields" value="1"/>
<param name="navigator_size" value="6"/>
<param name="button_data" value="1"/>
<param name="form_caption" value="en=Organizative Unit Lvl. 1|
es=Unidad Norganizativa N.1"/>
<param name="fields_remove" value="3"/>

<param ord="2" name="structid" value="st_ous_n"/>
</struct>

<struct id="st_ous_n" type="split">
  <param name="form_split" value="rows=50%,*"/>

  <param name="formid" value="frm_fk_descr"/>
  <param name="form_source" value="OU_lv2"/>
  <param name="form_type" value="form"/>
  <param name="form_caption" value="en=Organizative Unit Lvl. 2|
es=Unidad Norganizativa N.2"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="6"/>
  <param name="button_data" value="1"/>
  <param name="fields_modify[0]" value="search#DATA:OU_lv1;;;#3"/>
  <param name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="frm_fk_descr"/>
  <param ord="2" name="form_source" value="OU_lv3"/>
  <param ord="2" name="form_type" value="form"/>
  <param ord="2" name="form_caption"
value="en=Organizative Unit Lvl. 3|es=Unidad Norganizativa N.3"/>
  <param ord="2" name="navigator_position" value="W"/>
  <param ord="2" name="navigator_fields" value="1"/>
  <param ord="2" name="navigator_size" value="6"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="fields_modify[0]"
value="search#DATA:OU_lv2;;;#3"/>
  <param ord="2" name="source_filter_field" value="fk"/>
</struct>

<!-- ===== -->
<!-- Invel_type -->
<!-- ===== -->

<form id="frm_invel_type_attr" caption="XXXX">
  <orderby>name</orderby>
  <fields>
    <key source="pk"/>
    <string source="name" caption="es=Nombre|en=Name" len="40"
maxlen="80" />
    <string source="fkudtype" caption="en=Father|es=Padre" len="0"
canbenull="N" msg="Seleccione">
    <search>DATA:Unit_type;;;</search>

```

```

        <searchfield>pk,descr</searchfield>
    </string>
    <fkey source="fktype"/>
</fields>
</form>

<struct id="st_invel" type="relation">
    <param name="form_split" value="rows=340,*"/>

    <param name="formid" value="frm_fk_descr"/>
    <param name="form_source" value="invel_type"/>
    <param name="form_type" value="form"/>
    <param name="navigator_position" value="W"/>
    <param name="navigator_fields" value="1"/>
    <param name="navigator_size" value="10"/>
    <param name="button_data" value="1"/>
    <param name="form_caption" value="en=Inventory Element Type|
        es=Tipo Elemento Inventariable"/>
    <param name="fields_remove" value="3"/>

    <param ord="2" name="structid" value="st_invel_n"/>
</struct>

<struct id="st_invel_n" type="split">
    <param name="form_split" value="cols=50%,*"/>

    <param ord="1" name="formid" value="frm_invel_type_attr"/>
    <param ord="1" name="form_source" value="Model_type_attr"/>
    <param ord="1" name="form_type" value="table"/>
    <param ord="1" name="form_caption" value="Model type attrib"/>
    <param ord="1" name="button_data" value="1"/>
    <param ord="1" name="source_filter_field" value="fktype"/>

    <param ord="2" name="formid" value="frm_invel_type_attr"/>
    <param ord="2" name="form_source" value="Invel_type_attr"/>
    <param ord="2" name="form_type" value="table"/>
    <param ord="2" name="form_caption" value="Invel type attrib"/>
    <param ord="2" name="button_data" value="1"/>
    <param ord="2" name="source_filter_field" value="fktype"/>
</struct>

<!-- ===== -->
<!-- Units -->
<!-- ===== -->

<form id="frm_unit" source="Unit" caption="en=Unit|es=Unidad">
    <orderby>pk</orderby>
    <fields>
        <key source="pk"/>
        <string source="un" caption="Abrev" len="10" maxlen="10" />
        <string source="descr" caption="es=Descripcion|en=Description"
            len="40" maxlen="80" />
        <string source="reval" caption="validation" len="40" maxlen="80">
            <search>DATA:regex;;;</search>

```



2 Análisis de los módulos de activos, adquisiciones y común de GTI

```

<searchfield>code,descr</searchfield>
<widget>editcombobox</widget>
</string>
<text source="remarks" caption="en=Remarks|es=Notas" canbenull="Y"
  tooltip="Indicaciones">
  <widget>fstextbox</widget>
  <width>40</width>
  <height>2</height>
</text>
<fkey source="fk"/>
</fields>
</form>

```

```

<struct id="st_units" type="relation">
  <param name="form_split" value="rows=340,*"/>

  <param name="formid" value="frm_fk_descr"/>
  <param name="form_source" value="Unit_type"/>
  <param name="form_type" value="form"/>
  <param name="source_order" value="pk"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="10"/>
  <param name="button_data" value="1"/>
  <param name="form_caption" value="en=Unit Type|es=Tipo de unidad"/>
  <param name="fields_remove" value="3"/>

  <param ord="2" name="formid" value="frm_unit"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="source_filter_field" value="fk"/>
</struct>

```

```

<!-- ===== -->
<!-- Mark & Model -->
<!-- ===== -->

```

```

<form id="frm_attr" caption="en=Attributes|es=Atributos">
  <orderby>pk</orderby>
  <fields>
    <key source="pk"/>
    <string source="fkitattr" caption="name" len="0"
      canbenull="Y" attr="R">
      <search>DATA:Model_type_attr;;;</search>
      <searchfield>pk,name</searchfield>
    </string>
    <string source="val" caption="es=Valor|en=Value" len="40"
      maxlen="80" />

```

```

    <string source="fkud" caption="en=Unit|es=Unidad" len="4"
        canbenull="N" msg="Seleccione"/>
    <fkey source="fk"/>
</fields>
</form>

<form id="frm_model" source="Model" caption="es=Modelo|en=Model">
  <orderby>descr</orderby>
  <fields>
    <key source="pk"/>
    <string source="fktype" caption="en=Type|es=Tipo" len="0"
        canbenull="N" msg="Seleccione">
      <search>DATA:Invel_type;;;</search>
      <searchfield>pk,descr</searchfield>
    </string>
    <string source="descr" caption="es=Nombre|en=Name" len="40"
        maxlen="80" />
    <string source="pn" caption="PN" len="40" maxlen="80" />
    <text source="remarks" caption="en=Remarks|es=Notas" canbenull="Y"
        tooltip="Indicaciones">
      <widget>fstextbox</widget>
      <width>40</width>
      <height>2</height>
    </text>
    <fkey source="fk"/>
  </fields>
</form>

<struct id="st_mark" type="relation">
  <param name="form_split" value="rows=280,*"/>

  <param name="formid" value="frm_fk_descr"/>
  <param name="form_source" value="Mark"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="8"/>
  <param name="button_data" value="1"/>
  <param name="form_caption" value="en=Mark|es=Marca"/>
  <param name="fields_remove" value="3"/>

  <param ord="2" name="structid" value="st_mark_n"/>
</struct>

<struct id="st_mark_n" type="relation">
  <param name="form_split" value="rows=280,*"/>

  <param ord="1" name="formid" value="frm_model"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="2"/>
  <param name="navigator_size" value="8"/>
  <param name="button_data" value="1"/>
  <param name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="frm_attr"/>
  <param ord="2" name="form_source" value="Model_attr"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="button_update" value="1"/>

```



2 Análisis de los módulos de activos, adquisiciones y común de GTI

```
<param ord="2" name="source_filter_field" value="fk"/>
</struct>
```

```
<!-- ===== -->
<!-- Providers -->
<!-- ===== -->
```

```
<form id="frm_group" source="Provision_group" caption="en=Group|
es=Grupo compra">
  <orderby>pk</orderby>
  <fields>
    <key source="pk"/>
    <string source="fkprov" caption="" len="40" maxlen="80">
      <search>DATA:Provider;;;</search>
      <searchfield>pk,descr</searchfield>
    </string>
    <string source="descr" caption="es=Descripcion |en=Description"
      len="40" maxlen="80" />
    <string source="cn" caption="es=Contract Number |en=Cod. Contrato"
      len="40" maxlen="80" canbenull="Y"/>
    <date source="preq" caption="es=Request date. |en=F.Peticion"
      canbenull="Y"/>
    <date source="prec" caption="es=Receive date. |en=F.Recepcion"
      canbenull="Y"/>
    <date source="wstart" caption="es=Warranty start |en=Inicio Garantia"
      canbenull="Y"/>
    <date source="wend" caption="es=Warranty end |en=Fin Garantia"
      canbenull="Y"/>
    <text source="remarks" caption="en=Remarks |es=Notas" canbenull="Y"
      tooltip="Indicaciones">
      <widget>fstextbox</widget>
      <width>40</width>
      <height>2</height>
    </text>
  </fields>
</form>
```

```
<struct id="st_prov" type="relation">
  <param name="form_split" value="rows=340,*"/>

  <param name="formid" value="frm_fk_descr"/>
  <param name="form_source" value="Provider"/>
  <param name="form_type" value="form"/>
  <param name="source_order" value="descr"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="8"/>
  <param name="button_data" value="1"/>
  <param name="form_caption" value="en=Provider |es=Proveedor"/>
  <param name="fields_remove" value="3"/>

  <param ord="2" name="formid" value="frm_group"/>
  <param ord="2" name="navigator_position" value="W"/>
  <param ord="2" name="navigator_fields" value="2"/>
```

```

<param ord="2" name="navigator_size" value="8"/>
<param ord="2" name="button_data" value="1"/>
<param ord="2" name="source_filter_field" value="fkprov"/>
</struct>

<!-- ===== -->
<!-- Administrador -->
<!-- ===== -->

<main id="main_admin" caption="Admin menu">

  <menu caption="Procesos" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Unidades | en=Units" call="st_units"/>
    <option caption="es=Tipo de Elemento | en=Elemt type" call="st_invel"/>
    <option caption="es=Marcas y Modelos | en=Marks and Models"
      call="st_mark"/>
    <option caption="es=Proveedores y compras | en=Providers and Groups"
      call="st_prov"/>
    <option caption="es=Unidades Organizativas | en=Organizative Units"
      call="st_ous"/>
    <option caption="es=Localizaciones | en=Sites and Locations"
      call="st_site"/>
  </menu>

  <menu caption="Misc" tooltip="Procesos, actividades y Tareas">
    <option caption="logout" url="logout.php"/>
  </menu>

</main>

</asl>

```

2.6.5 Winter

2.6.5.1 Introducción

La gestión de los Sistemas de Información en las grandes organizaciones puede llegar a ser muy compleja. En casos extremos, un organismo puede manejar cientos de aplicaciones que son utilizadas por miles de usuarios. En este tipo de entornos puede ser realmente difícil responder en este a preguntas como: ¿qué sistema de información requiere una inversión estratégica? ¿qué equipo de desarrollo mantiene la aplicación X? ¿a qué información tiene acceso el usuario U?. Sin embargo, disponer de respuestas a estas preguntas de forma ágil es fundamental para la gobernanza TIC de la organización.

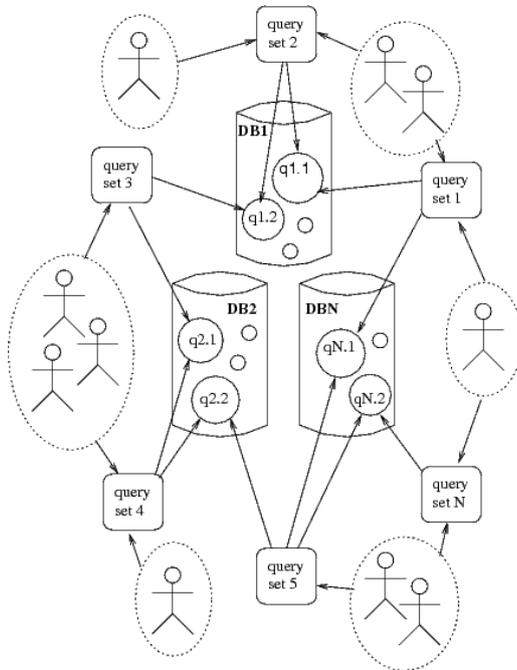
En este artículo se propone un modelo de hyper-integración para los sistemas de información. En el modelo propuesto, la información, las aplicaciones, los procesos y las personas (usuarios, desarrolladores, gestores, etc.) conviven en un único entorno de información. La aplicación de este modelo conllevaría interesantes consecuencias que son analizadas en este artículo.

Los autores realizan esta propuesta tras la experiencia en el desarrollo de un sistema que pretende dar solución a un problema particular, pero evolucionó de forma radical y fué capaz de adaptarse a nuevas necesidades del cliente convirtiéndose finalmente en un entorno en el que se fue integrando una parte importante del conocimiento del organismo.



2.6.5.2 El problema original

Desde un organismo público del gobierno regional nos solicitaron asesoramiento para la resolución de un problema. En este organismo disponían de un elevadísimo número de bases de datos relacionales y deseaban explotar su información. El cliente deseaba atacar directamente a las bases de datos empleando consultas realizadas a medida. Estas consultas se debían clasificar según su temática y las distintas categorías definidas se pondrían a disposición de los usuarios en función de su perfil. El objetivo final era desarrollar una única aplicación a través de la cual los usuarios tuvieran visibilidad sobre la información ofrecida por los distintos grupos de consultas.



En esta figura se presenta un resumen de la problemática planteada. Sobre las bases de datos implicadas (DB1, DB2, etc.) se pueden definir diversas consultas de interés (q1.1, q1.2, q2.1, etc.). Estas consultas son agrupadas en conjuntos según su función (query set 1, query set 2, etc.). Cada usuario o grupo de usuarios puede acceder a varios de estos conjuntos en función de su perfil.

Algunos condicionantes de importancia del sistema eran:

- El número de consultas a crear era enorme (cientos), así como las bases de datos implicadas.

1. Los plazos para llevar a cabo el desarrollo eran muy ajustados y los módulos finalizados debían ponerse a disposición de los usuarios de forma inmediata

2. Las bases de datos origen sufrirían modificaciones continuas durante su vida, por lo que la solución propuesta debía ser flexible y capaz de adaptarse a cambios en los datos fácilmente.
3. El sistema pasaría a ser mantenido por personal del propio organismo una vez fuera entregado, por lo tanto, el mantenimiento de la solución debía ser sencillo y ágil.

2.6.5.3 La solución propuesta

En este escenario se desechó el proceso de desarrollo tradicional formado por fases secuenciales de análisis, diseño y construcción:

- Un desarrollo de este tipo concluiría con la obtención y entrega del producto software finalizado, pero en un plazo de tiempo inadmisibles (tras realizar el análisis de cientos de bases de datos y crear una enorme cantidad de consultas e interfaces de usuario). Además, dilatar tanto la entrega del producto final generaría una gran incertidumbre sobre el éxito del proyecto.
- Por otra parte, los productos monolíticos de gran tamaño presentan un mantenimiento complejo. Esto es un inconveniente importante cuando sabemos con antelación que la estará sometida a evoluciones y cambios continuos. Es necesario también tener en cuenta que el mantenimiento sería realizado por personal del propio organismo que no habría participado en el desarrollo del proyecto, por lo tanto, era fundamental asegurarse de que el mantenimiento del proyecto sería un éxito.

Desde nuestro punto de vista era primordial proponer un modelo más ágil que permitiera poner a disposición del cliente una aplicación mínima de forma casi inmediata y en la que paulatinamente se irían integrando diferentes módulos. Esto minimizaría la incertidumbre tanto del cliente como del equipo de desarrollo y a su vez permitiría tratar a la aplicación como un conjunto de componentes integrados en lugar de como un programa monolítico. La aplicación debería gestionar los

módulos desarrollados junto con su documentación asociada. Esta información sería la base del proceso de mantenimiento una vez que la aplicación fuera entregada.

La naturaleza del problema, en la que los datos provienen de una plétora de orígenes, donde sólo sería necesaria la presentación de los mismos, junto a la necesidad de disponer rápidamente la aplicación en funcionamiento nos impulsó a emplear nuestro MB-UIDE para su construcción. WAINE (Web Application INterface Engine) es un MB-UIDE que genera aplicaciones web basadas en el paradigma del formulario. Aunque originalmente fue diseñado para facilitar el desarrollo sistemático de la interfaz de usuario a estudiantes de ingeniería, ha sido empleado con éxito para desarrollar varios proyectos.

Dado que el producto se construiría con WAINE y conceptualmente se trataba de un sistema integrador de módulos decidimos llamar al proyecto WAINE INTEgratoR o WINTER.

2.6.5.3.1 El proceso de desarrollo

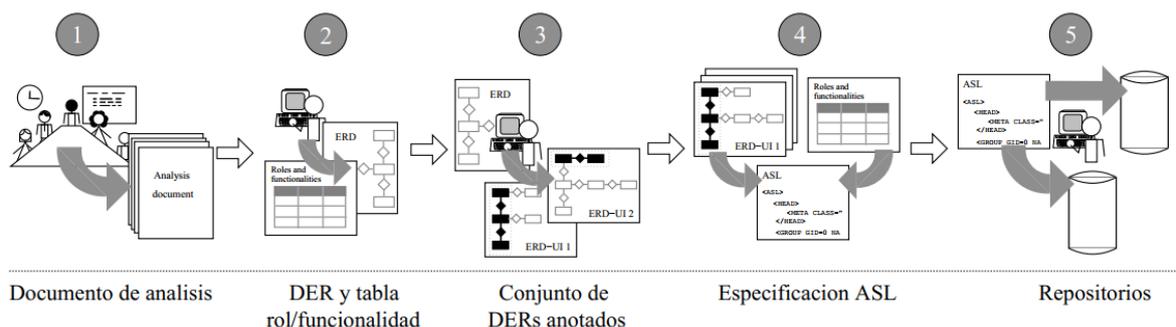
Para conseguir nuestro propósito era necesario trabajar en dos líneas de desarrollo paralelas:

- La construcción de las consultas necesarias junto a su presentación
 1. El desarrollo de Winter

En la primera línea, para cada consulta a integrar sería necesario realizar un estudio profundo de las bases de datos involucradas. El proceso de desarrollo propuesto requeriría de las siguientes actividades:

- Identificar junto con los usuarios expertos la base de datos afectada y la información a extraer para la consulta.
 - a. Se identificaría la base de datos implicada: nombre, tipo, parámetros de conexión, etc.
 - b. Si no se dispone de documentación de la base de datos a tratar, se procedería a su análisis y documentación.
 - c. Por último, se documentaría la consulta a realizar, tablas implicadas, columnas seleccionadas, orden, agrupación, etc.
- 2. Construcción de la consulta. El producto de esta actividad sería el código SQL necesario para la ejecución de la consulta.
- 3. Establecer los perfiles/usuarios con acceso a la información. Para definir este aspecto se construye un catálogo de usuarios con acceso a los datos.
- 4. Especificación de las interfaces de usuario que mostrarán los datos de la consulta. Para ello empleamos Diagramas ER-anotados.
- 5. Construcción de las Interfaces de usuario. Describiendo cada uno de los contenedores y formularios necesarios en lenguaje ASL.

Para el desarrollo de winter (segunda línea) se seguiría el proceso habitual empleado en los proyectos basados en WAINE:





2 Análisis de los módulos de activos, adquisiciones y común de GTI

- Análisis. En esta fase se elabora un documento en el que se describen los roles de los usuarios, las funcionalidades de la aplicación y los bocetos de las principales interfaces de usuario e informes empleados por cada rol.
- 1. Modelado de conceptos. Tomando como base el documento de análisis se elaboran dos documentos: (a) la tabla de roles/funcionalidades que indica las acciones e interfaces de usuario necesarias para cada rol y (b) el modelo conceptual de la base de datos que se plasma en un Diagrama Entidad-Relación.
- 2. Anotación del Diagrama Entidad-Relación. El objetivo de esta fase es especificar cada unidad de interacción de la interfaz de usuario. Para ello se toma la porción de Diagrama Entidad-Relación afectada por la unidad de interacción que se está definiendo y se enriquece anotando sobre él los componentes que tomarán parte en su construcción.
- 3. Especificación en ASL. Los desarrolladores partiendo de la tabla rol/funcionalidad y los Diagramas Entidad-Relación anotados crean un documento codificado en lenguaje ASL.
- 4. Generación de repositorios y personalización. El proceso finaliza con la generación de los repositorios usados por el Run-time de WAINÉ y especificando aspectos concretos de la interfaz de usuario (colores, bordes, fuentes, etc.) si fuera necesario.

2.6.5.3.2 La aplicación Winter

Pasamos en este apartado a describir los aspectos más relevantes de la aplicación Winter.

2.6.5.3.2.1 Tabla Rol/Funcionalidad

La aplicación define tres roles posibles:

- usuario: Accede a los distintos conjuntos de consultas que tiene asignados empleando un elemento que denominamos panel. A través de los paneles los usuarios tienen acceso a la visualización de las consultas, así como a la ejecución de acciones.
- 1. administrador: Administra a los distintos usuarios del sistema y les concede acceso a los paneles disponibles según su perfil. El administrador también aprueba nuevas versiones de la aplicación.
- 2. desarrollador: Pueden visualizar cualquier panel, así como definir nuevos paneles, consultas, interfaces de usuario, etc.

En la siguiente tabla se presenta un resumen las funcionalidades de principales asociadas a cada rol:

Funcionalidad	Usuario	Desarrollador	Administrador
Listado de paneles definidos	✓	✓	✓
Visualización paneles asignados	✓		
Visualización de todos los paneles		✓	
Edición de paneles		✓	
Edición de unidades de interacción		✓	
Edición de acciones		✓	
Configuración de orígenes de datos		✓	
Desarrollo ASL		✓	

Desarrollo SQL		✓	
Definición de versión		✓	
Gestión de usuarios			✓
Aprobar versiones			✓

2.6.5.3.2 Modelo de dominio

De forma resumida, el modelo conceptual de datos para soportar la aplicación winter es el siguiente:

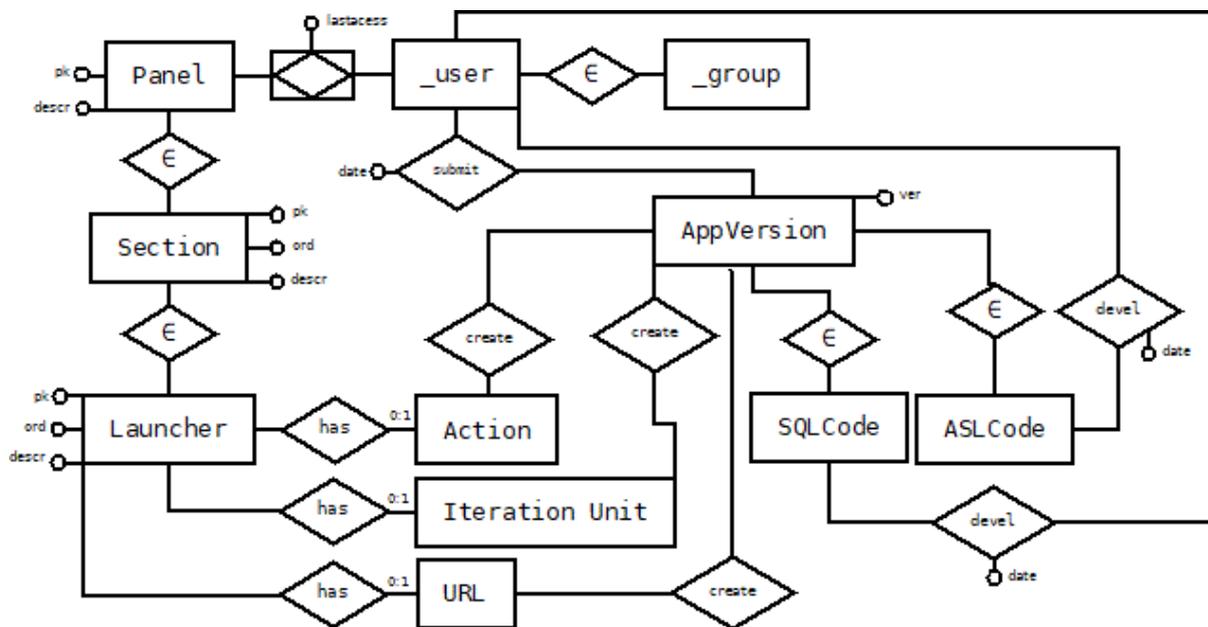


Ilustración 38: Modelo de dominio de la aplicación winter

Los usuarios se clasifican en grupos de acuerdo con su rol. Cada usuario puede tener acceso a varios paneles y un mismo panel puede ser accedido por varios usuarios (la fecha de último acceso de cada usuario a un panel queda recogida en el modelo). Los paneles tienen diversas secciones y en ellas podemos encontrar varios lanzadores. Un lanzador puede tener asociada una acción, una unidad de interacción o una URL. El resto del diagrama sintetiza el proceso de desarrollo. Los desarrolladores, también usuarios del sistema, construyen ficheros ASL, SQL o de cualquier otro tipo que son vinculados a una determinada versión. Las versiones son aprobadas por el administrador del sistema y en ese momento se generan unidades de interacción, acciones, etc.

2.6.5.3.2.3 Especificación de las unidades de interacción

Para concluir esta sección presentamos la definición de alguna de las unidades de interacción más interesantes de la aplicación. Para la especificación de estas unidades empleamos diagramas ER-anotados.



2.6.5.3.2.4 Asignación de paneles a un usuario

Se trata de la interfaz de usuario empleada por los administradores para asignar paneles a un usuario. En primer lugar, se selecciona el usuario (1). Posteriormente se añaden nuevos paneles o bien se modifican o eliminan los ya asignados (2).

Ilustración 39: Asignación de paneles a usuarios

2.6.5.3.2.5 Visualización de todos los paneles

Esta unidad de interacción es de las más complejas y relevantes. Permite a los desarrolladores seleccionar un panel dentro de los definidos en el sistema (1). Una vez seleccionado el panel se muestra con sus secciones y launchers (2). Finalmente, si se activa un lanzador se muestra a la derecha el formulario asociado (3).

	Column 1	Column 2	..	Column N
	Cell11	Cell21	:	CellN1
	Cell12	Cell22	:	CellN2
	:	:	:	:
	:	:	:	:
	:	:	:	:
	:	:	:	:
	Cell1M	Cell2M	..	CellNM

Ilustración 40: Visualización de paneles

Los usuarios regulares disponen de una unidad de interacción análoga en la cual los paneles que aparecen en el desplegable (1) están limitados a aquellos a los que el usuario tiene acceso. A continuación, se muestra una imagen con la interfaz de usuario final correspondiente a esta unidad de interacción:

The screenshot displays a web application interface with a light blue background featuring a winter scene with evergreen trees. The interface is divided into several sections:

- Desarrollo (Development):** A sidebar menu on the left containing categories like 'Datos', 'Repositorio SQL', 'Paneles', 'Edición de Paneles', 'Modo experto', and 'Misc'.
- Menu principal (Main Menu):** A dropdown menu at the top center showing 'Título NAVEX: Administración' and a 'Filtrar' button.
- Menu (Menu):** A secondary menu box with options for 'Administración Usuarios', 'Técnicos', and 'Consultas'.
- Usuarios de Navex (Navex Users):** A table on the right side of the screen listing user details.

The 'Usuarios de Navex' table contains the following data:

#	user_id	Login	Nombre	Delegación	Rol	Permisos	Gestion
1	1	josec.pacheco.ext	José Carlos Pacheco	Todas	Administrador	No	Si
2	2	francisco.silveira	Paco Silveira	Todas	Administrador	No	Si
3	3	alfredo.bolanos	Alfredo Bolaños	Servicios Centrales	Usuario	No	No
4	4	antonio.lopez	Antonio López	Todas	Usuario	No	No
5	5	jfelipe.cara	Jorge Felipe Cara	Todas	Usuario	No	No
6	6	julian.cadenas.ext	Julían Cadenas	Todas	Usuario	No	Si
7	7	marial.sanchez.delgado	Maria Luz Sánchez	Servicios Centrales	Usuario	No	Si
8	8	carmen.arias.gil	Carmen Arias	Servicios Centrales	Usuario	No	Si
9	9	mmarperez	Maria del Mar Pérez Muñoz	Todas	Usuario	No	Si
10	10	rosario.ponce.bernal	Rosario Ponce	Servicios Centrales	Administrador	No	Si
11	11	mcarmen.rodriguez.rubio.ext	M. Carmen Rodríguez	Servicios Centrales	Usuario	No	No
12	12	maria.d.sierra	Maria Dolores Sierra	Servicios Centrales	Usuario	No	Si
13	13	dgfpe.ced	DGFPE	Servicios Centrales	Usuario	No	No
14	14	rosa.pires	Rosa Pires	Servicios Centrales	Usuario	No	No
15	15	marial.bomez	Maria Isabel Bómezz	Servicios Centrales	Usuario	No	Si

Ilustración 41: Interfaz de usuario final para la visualización de paneles



2.6.6 Módulo fdescr

El módulo `fdescr.asl` define un formulario que permite gestionar tablas que disponen únicamente de una clave primaria identificada por `pk` y un campo descripción con identificador `descr`.

2.6.6.1 Funcionalidades

Este módulo define únicamente un formulario que permite manipular el tipo de tabla descrito con anterioridad: `fdescr.form.descr`

fdescr.form.descr (descr1)		
Descripcion		
descr1 - 1	Modificar	Eliminar
descr1 - 2	Modificar	Eliminar
descr1 - 3	Modificar	Eliminar
	Añadir	

Ilustración 42: formulario `fdescr.form.descr`

2.6.6.2 Ejemplos

En este ejemplo el formulario `fdescr.form.descr` se emplea en el contenedor `st_tipo` (de tipo `form`) que permite manipular la tabla `Tipo`:

```
<struct id="st_tipo" type="form">
  <param name="formid" value="fdescr.form.descr"/>
  <param name="form_source" value="Tipo"/>
  <param name="form_type" value="table"/>
  <param name="form_caption" value="Tipos permitidos"/>
  <param name="button_data" value="1"/>
</struct>
```

2.6.7 Módulo ffkdescr

El módulo `ffkdescr.asl` define un formulario que permite gestionar tablas que tienen la siguiente estructura:

1. clave primaria identificada por **pk**
2. campo descripción con identificador **descr**
3. clave ajena a otra tabla con identificador **fk**

2.6.7.1 Funcionalidades

Este módulo define únicamente un formulario que permite manipular el tipo de tabla descrito con anterioridad: `ffkdescr.form.descr`

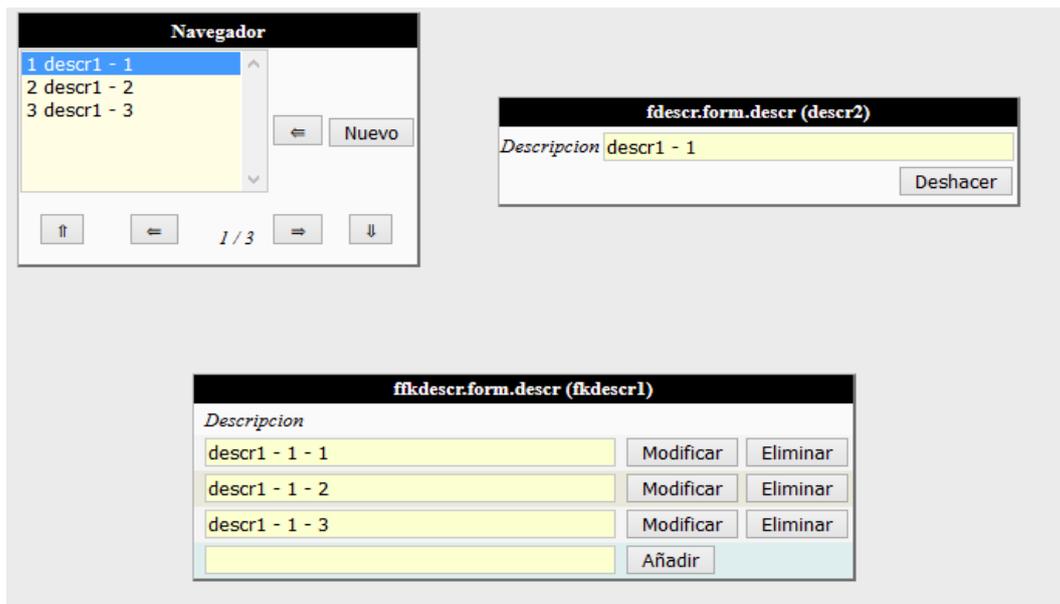


Ilustración 43: formulario ffkdescr.form.descr

2.6.7.2 Ejemplos

En este ejemplo el formulario `ffkdescr.form.descr` se emplea en el contenedor `st_Tipo_SubTipo` (de tipo `relation`) para manipular las tablas `Tipo` y `SubTipo`. Como se puede apreciar en el ejemplo, el formulario `ffkdescr.form.descr` se está empleando en combinación con el formulario `fdescr.form.descr`.

```
<struct id="st_Tipo_SubTipo" type="relation">
  <param name="form_split" value="rows=30%,*" />

  <param name="formid" value="fdescr.form.descr" />
  <param name="form_source" value="Tipo" />
  <param name="form_type" value="form" />
  <param name="navigator_position" value="W" />
  <param name="form_caption" value="Tipo" />

  <param ord="2" name="formid" value="ffkdescr.form.descr" />
  <param ord="2" name="form_source" value="SubTipo" />
  <param ord="2" name="form_type" value="table" />
  <param ord="2" name="form_caption" value="SubTipo" />
  <param ord="2" name="source_filter_field" value="fk" />
  <param ord="2" name="button_data" value="1" />
</struct>
```

2.6.8 Paquete `percon`

percon (*PER*sonal *CON*tacts) es un componente que permite gestionar información de contacto de personas. Más concretamente permite mantener información sobre direcciones, teléfonos, emails y notas.



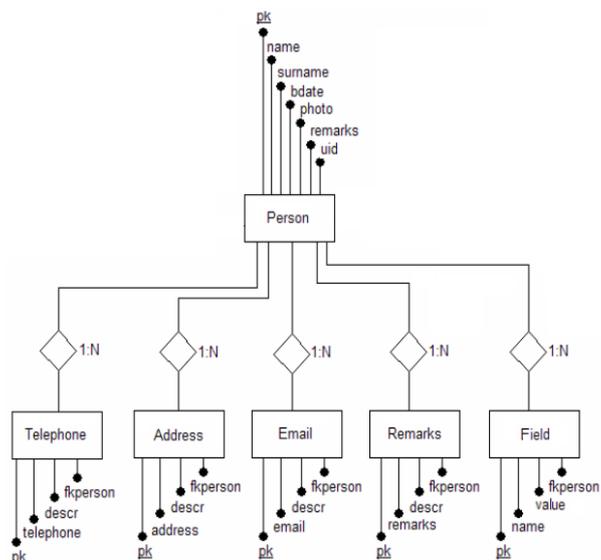
The image shows two screenshots of a web application interface. The top screenshot displays a 'Person' form with a sidebar 'Navegador' (Navigator) showing 'Doe John'. The form fields include 'Nombre' (John), 'Apellidos' (Doe), 'Fecha de Nacimiento' (with a calendar icon), and 'Usuario' (john). There are buttons for 'IMPRIMIR', '?', 'Personal', 'Photo', 'Others', 'DESHACER', 'MODIFICAR', 'AÑADIR', and 'ELIMINAR'. Below the form is a horizontal tab bar with 'Telephone', 'Address', 'Email', and 'Remarks'. The bottom screenshot shows a 'Telephone' table with columns 'Telefono' and 'Descripcion'. It contains one entry: '555 55 55 55' with description 'Unreal telephone'. There are buttons for 'MODIFICAR', 'ELIMINAR', and 'AÑADIR'.

2.6.9 Funcionalidades

El paquete provee elementos de los modelos de presentación y dominio. En cuanto a la presentación ofrece los siguientes elementos:

- percon.form.person
- percon.form.telephone
- percon.form.address
- percon.form.email
- percon.form.remarks
- percon.struct.person
- percon.struct.person_n

En cuanto al modelo de dominio tiene la siguiente estructura:



- percon_Person: datos de la persona
- percon_Telephone: teléfonos y su descripción
- percon_Address: direcciones y su descripción
- percon_Email: correos electrónicos y su descripción
- percon_Field: pares nombre-valor
- percon_Remarks: anotaciones sobre el usuario

2.6.10 Paquete attach

El **paquete attach** ofrece interfaces de usuario para el uso de adjuntos en cualquier aplicación WAINE. Los adjuntos pueden estar clasificados y por defecto tienen protección de modificación y borrado, es decir, sólo puede borrar o eliminar un adjunto quien lo creó. Estas restricciones pueden ser eliminadas (ver sección de ejemplos).

2.6.10.1 Funcionalidades

El paquete provee elementos de los modelos de presentación y dominio. En cuanto a la presentación ofrece los siguientes elementos:

- **attach.struct.category**: Administración de categorías de adjuntos

Categoria	
Descripcion	
A Type	Modificar Eliminar
B Type	Modificar Eliminar
None	Modificar Eliminar
	Añadir

- **attach.struct.search**: Búsqueda de adjuntos



Buscar adjuntos

Descripción
 Tipo
 Creador por
 F.Creacion
 Modificado por
 F.Modificacion

Adjuntos

Descripcion	Archivo	Tipo	Creador por	F.Creacion	Modificado por	F.Modificacion
-------------	---------	------	-------------	------------	----------------	----------------

- **attach.struct.nocatsearch:** Búsqueda de adjuntos sin categoría (para los casos en que no se empleen categoría)

Buscar adjuntos

Descripción
 Creador por
 F.Creacion
 Modificado por
 F.Modificacion

Adjuntos

Descripcion	Archivo	Creador por	F.Creacion	Modificado por	F.Modificacion
-------------	---------	-------------	------------	----------------	----------------

- **attach.struct.listall:** Lista de todos los adjuntos del sistema

Adjuntos						
Descripcion	Archivo	Tipo	Creador por	F.Creacion	Modificado por	F.Modificacion
aal	Descargar	A Type	demo	2018-08-01 23:23:04	demo	2018-08-01 23:23:04

- **attach.form.catattach:** Formulario para la gestión de adjuntos. En la sección de ejemplos puede ver cómo emplearlo en sus aplicaciones

Adjuntos

Descripción Archivo Tipo

 Descargar **Borrar**

 No se ha seleccionado ningún archivo.

- **attach.form.category:** Formulario para la manipulación de categorías

2.6.10.2 Ejemplos

- Adjunto con todas sus características: categorías, protección de borrado y actualización

```

<struct id="s_catattachpud" type="relation">
  <param name="form_split" value="rows=18%,*"/>

  <param ord="1" name="formid" value="f_entry"/>

  <!-- Formulario con datos genéricos -->
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_readonly" value="1"/>
  <param ord="1" name="source_filter_field" value="pk"/>
    
```

```

    <param ord="1" name="source_filter_value" value="1"/>
    <param ord="1" name="source_row" value="0"/>

    <param ord="2" name="formid" value="attach.form.catattach"/>
<!-- Formulario con adjuntos a los datos anteriores -->
    <param ord="2" name="form_type" value="table"/>
    <param ord="2" name="button_data" value="1"/>
    <param ord="2" name="fields_remove" value="5:7"/>
    <param ord="2" name="source_filter_field" value="fk"/>
</struct>

```

- Adjunto con categoría prefijada, protección de borrado y actualización

```

<struct id="s_fixcatattachpud" type="relation">
    <param name="form_split" value="rows=18%,*"/>

    <param ord="1" name="formid" value="f_entry"/>

<!-- Formulario con datos genéricos -->
    <param ord="1" name="form_type" value="table"/>
    <param ord="1" name="fields_readonly" value="1"/>
    <param ord="1" name="source_filter_field" value="pk"/>
    <param ord="1" name="source_filter_value" value="2"/>
    <param ord="1" name="source_row" value="0"/>

    <param ord="2" name="formid" value="attach.form.catattach"/>
<!-- Formulario con adjuntos a los datos anteriores -->
    <param ord="2" name="form_type" value="table"/>
    <param ord="2" name="fields_modify[0]" value="defvalue#2#3"/>
<!-- Asignando categoría -->
    <param ord="2" name="fields_remove" value="5:7"/>
    <param ord="2" name="fields_readonly" value="3"/>
    <param ord="2" name="button_data" value="1"/>
    <param ord="2" name="source_filter_field" value="fk"/>
</struct>

```

- Adjunto sin categoría pero con protección de borrado y actualización

```

<struct id="s_unicatattachpud" type="relation">
    <param name="form_split" value="rows=18%,*"/>

    <param ord="1" name="formid" value="f_entry"/>
<!-- Formulario con datos genéricos -->
    <param ord="1" name="form_type" value="table"/>
    <param ord="1" name="fields_readonly" value="1"/>
    <param ord="1" name="source_filter_field" value="pk"/>
    <param ord="1" name="source_filter_value" value="3"/>
    <param ord="1" name="source_row" value="0"/>

    <param ord="2" name="formid" value="attach.form.catattach"/>
<!-- Formulario con adjuntos a los datos anteriores -->
    <param ord="2" name="form_type" value="table"/>
    <param ord="2" name="fields_remove" value="3:5:7"/>
<!-- Se elimina también el campo categoría -->
    <param ord="2" name="button_data" value="1"/>
    <param ord="2" name="source_filter_field" value="fk"/>
</struct>

```



- Adjunto concategorías y protección de borrado

```
<struct id="s_catattachpd" type="relation">
  <param name="form_split" value="rows=18%,*"/>

  <param ord="1" name="formid" value="f_entry"/>
  <!-- Formulario con datos genéricos -->
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_readonly" value="1"/>
  <param ord="1" name="source_filter_field" value="pk"/>
  <param ord="1" name="source_filter_value" value="1"/>
  <param ord="1" name="source_row" value="0"/>

  <param ord="2" name="formid" value="attach.form.catattach"/>
  <!-- Formulario con adjuntos a los datos anteriores -->
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="fields_hidden" value="6"/>
  <param ord="2" name="fields_remove" value="5:7"/>
  <param ord="2" name="event_beforeupdate" value="0"/>
  <param ord="2" name="source_filter_field" value="fk"/>
</struct>
```

- Adjunto con categoría prefijada y protección de borrado

```
<struct id="s_fixcatattachpd" type="relation">
  <param name="form_split" value="rows=18%,*"/>

  <param ord="1" name="formid" value="f_entry"/>
  <!-- Formulario con datos genéricos -->
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_readonly" value="1"/>
  <param ord="1" name="source_filter_field" value="pk"/>
  <param ord="1" name="source_filter_value" value="2"/>
  <param ord="1" name="source_row" value="0"/>

  <param ord="2" name="formid" value="attach.form.catattach"/>
  <!-- Formulario con adjuntos a los datos anteriores -->
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_modify[0]" value="defvalue#2#3"/>
  <param ord="2" name="fields_hidden" value="6"/>
  <param ord="2" name="fields_remove" value="5:7"/>
  <param ord="2" name="fields_readonly" value="3"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="event_beforeupdate" value="0"/>
  <param ord="2" name="source_filter_field" value="fk"/>
</struct>
```

- Adjunto sin categoría con protección de eliminación

```
<struct id="s_uncatattachpd" type="relation">
  <param name="form_split" value="rows=18%,*"/>
```

```

    <param ord="1" name="formid" value="f_entry"/>
    <!-- Formulario con datos genéricos -->
    <param ord="1" name="form_type" value="table"/>
    <param ord="1" name="fields_readonly" value="1"/>
    <param ord="1" name="source_filter_field" value="pk"/>
    <param ord="1" name="source_filter_value" value="3"/>
    <param ord="1" name="source_row" value="0"/>

    <param ord="2" name="formid" value="attach.form.catattach"/>
    <!-- Formulario con adjuntos a los datos anteriores -->
    <param ord="2" name="form_type" value="table"/>
    <param ord="2" name="fields_hidden" value="6"/>
    <param ord="2" name="fields_remove" value="3:5:7"/>
    <param ord="2" name="button_data" value="1"/>
    <param ord="2" name="event_beforeupdate" value="0"/>
    <param ord="2" name="source_filter_field" value="fk"/>
</struct>

```

- Adjunto concategorías sin protección

```

<struct id="s_catattach" type="relation">
    <param name="form_split" value="rows=18%,*"/>

    <param ord="1" name="formid" value="f_entry"/>
    <!-- Formulario con datos genéricos -->
    <param ord="1" name="form_type" value="table"/>
    <param ord="1" name="fields_readonly" value="1"/>
    <param ord="1" name="source_filter_field" value="pk"/>
    <param ord="1" name="source_filter_value" value="1"/>
    <param ord="1" name="source_row" value="0"/>

    <param ord="2" name="formid" value="attach.form.catattach"/>
    <!-- Formulario con adjuntos a los datos anteriores -->
    <param ord="2" name="form_type" value="table"/>
    <param ord="2" name="button_data" value="1"/>
    <param ord="2" name="fields_hidden" value="4:6"/>
    <param ord="2" name="fields_remove" value="5:7"/>
    <param ord="2" name="event_beforeupdate" value="0"/>
    <param ord="2" name="event_beforedetelete" value="0"/>
    <param ord="2" name="source_filter_field" value="fk"/>
</struct>

```

- Adjunto con categoría prefijada sin protección

```

<struct id="s_fixcatattach" type="relation">
    <param name="form_split" value="rows=18%,*"/>

    <param ord="1" name="formid" value="f_entry"/>
    <!-- Formulario con datos genéricos -->
    <param ord="1" name="form_type" value="table"/>
    <param ord="1" name="fields_readonly" value="1"/>
    <param ord="1" name="source_filter_field" value="pk"/>
    <param ord="1" name="source_filter_value" value="2"/>
    <param ord="1" name="source_row" value="0"/>

    <param ord="2" name="formid" value="attach.form.catattach"/>
    <!-- Formulario con adjuntos a los datos anteriores -->

```



2 Análisis de los módulos de activos, adquisiciones y común de GTI

```
<param ord="2" name="form_type" value="table"/>
<param ord="2" name="fields_modify[0]" value="defvalue#2#3"/>
<param ord="2" name="fields_hidden" value="4:6"/>
<param ord="2" name="fields_remove" value="5:7"/>
<param ord="2" name="fields_readonly" value="3"/>
<param ord="2" name="button_data" value="1"/>
<param ord="2" name="event_beforeupdate" value="0"/>
<param ord="2" name="event_beforedelete" value="0"/>
<param ord="2" name="source_filter_field" value="fk"/>
</struct>
```

- Adjunto sin categorías ni protección

```
<struct id="s_uncatattach" type="relation">
  <param name="form_split" value="rows=18%,*"/>

  <param ord="1" name="formid" value="f_entry"/>
  <!-- Formulario con datos genéricos -->
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_readonly" value="1"/>
  <param ord="1" name="source_filter_field" value="pk"/>
  <param ord="1" name="source_filter_value" value="3"/>
  <param ord="1" name="source_row" value="0"/>

  <param ord="2" name="formid" value="attach.form.catattach"/>
  <!-- Formulario con adjuntos a los datos anteriores -->
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_hidden" value="4:6"/>
  <param ord="2" name="fields_remove" value="3:5:7"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="event_beforeupdate" value="0"/>
  <param ord="2" name="event_beforedelete" value="0"/>
  <param ord="2" name="source_filter_field" value="fk"/>
</struct>
```

3 Diseño de los módulos de activos, adquisiciones y común de GTI

3.1 Introducción

En este documento se explica el proceso de diseño referente al módulo de inventario, módulo común y módulo de adquisiciones de la aplicación GTI. Este proceso se compone de los siguientes puntos:

1. Definición de la arquitectura del sistema.
2. Diseño del modelo físico de datos que se emplea en la Base de Datos del módulo.
3. Definición de eventos y funciones que complementan al módulo.
4. Diseño de interfaces de usuario haciendo uso de los diagramas entidad-relación anotados.
5. Descripción de los casos de uso que muestran las principales funcionalidades del módulo.
6. Explicación de las especificaciones de construcción.
7. Exposición de la carga de datos iniciales.
8. Definición del plan de pruebas.
9. Descripción de los requisitos de implantación.

En los siguientes apartados, se va a detallar cada uno de los puntos mencionados.

3.2 Arquitectura del sistema

En este apartado se detalla la disposición física de los elementos hardware y software que componen el sistema. Concretamente, se despliega el módulo de inventario y el de adquisiciones. Para ello, se presenta un diagrama de despliegue que muestra esta arquitectura:

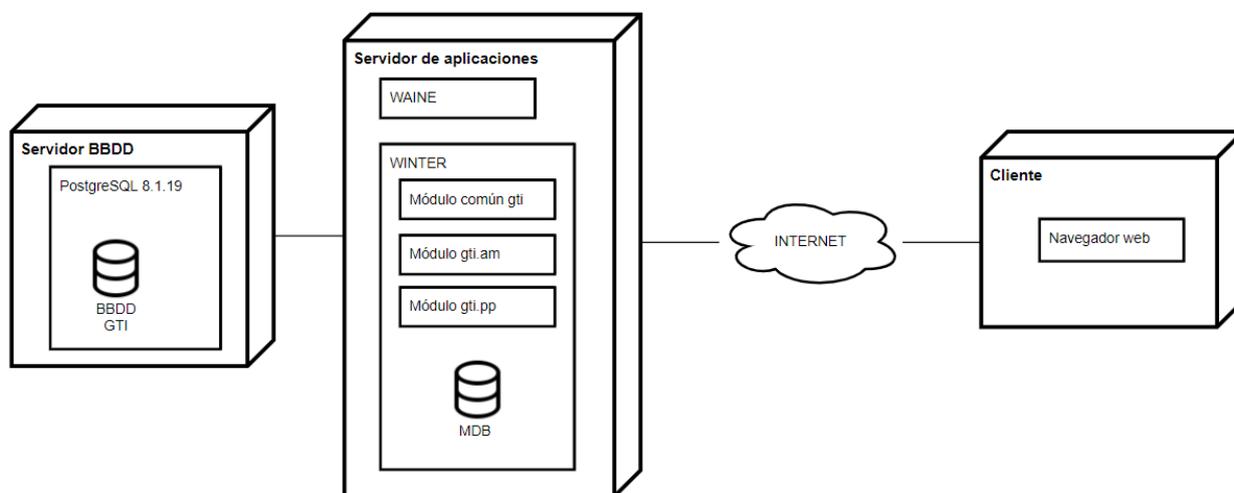


Ilustración 44: Diagrama de despliegue

Los elementos que se distinguen en la arquitectura son los siguientes:

- Servidor BBDD: incluye la base de datos de la aplicación desarrollado con el sistema de gestión de bases datos PostgreSQL. Esta BBDD contiene todos los datos relacionados con los módulos común, inventario y adquisiciones que serán mostrados al cliente a través de las interfaces de usuario.
- Servidor de aplicaciones: se compone por el integrador de WINTER, que permite la



3 Diseño de los módulos de activos, adquisiciones y común de GTI

integración de los módulos común, inventario y adquisiciones; los repositorios de las interfaces de usuario (MDB), que se almacenan en los modelos de diálogo, presentación y usuario; y el run-time de **WAINE** que es el encargado de generar las interfaces de usuario a partir de todos los componentes que el desarrollador haya definido.

- Cliente: es el navegador web utilizado por el usuario que se encarga de hacer las peticiones al servidor y mostrar sus respuestas a través de las interfaces de usuario.

La comunicación existente entre estos componentes es la siguiente:

1. El cliente hace un petición de interfaz de usuario a través del navegador web.
2. Esta petición es atendida por el run-time de **WAINE** que genera la interfaz de usuario en tiempo de ejecución, haciendo uso de los repositorios de las interfaces de usuario, RC y la BBDD GTI.

Por último, esta IU generada es enviada al navegador web del cliente, donde será interpretada y mostrada al cliente.

3.3 Modelo físico de datos

3.3.1 Modelo físico de datos modulo am

En este apartado se describe el modelo físico de datos utilizado para el módulo de gestión de activos de GTI. Para ello, se presenta el diagrama del modelo y se detallan cada una de las tablas y campos que las componen.

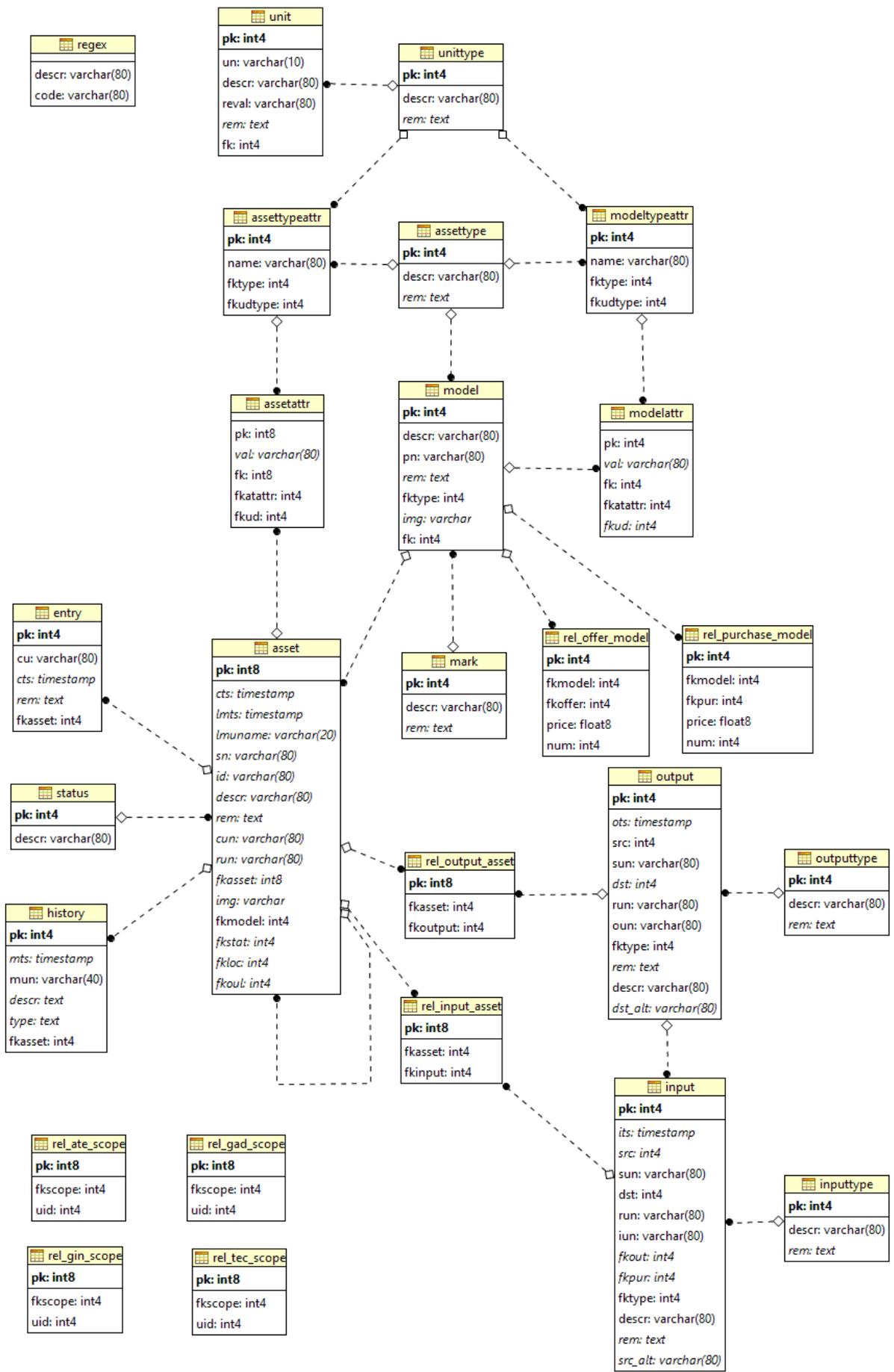


Ilustración 45: Modelo físico de datos módulo am



Las tablas que lo componen son las siguientes:

- **Unittype:** modela un tipo de unidad. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a un tipo de unidad.
 - *descr*: descripción del tipo de unidad. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre el tipo de unidad.
- **Unit:** modela una unidad. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una unidad.
 - *un*: nombre abreviado de la unidad. Hasta 10 caracteres.
 - *descr*: descripción de la unidad. Hasta 80 caracteres.
 - *reval*: formato del valor que la unidad debe seguir. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la unidad.
 - *fk*: entero que hace referencia al tipo de unidad.
- **Regex:** modela una expresión regular. Los campos que lo componen son:
 - *descr*: descripción de la expresión regular. Hasta 80 caracteres.
 - *code*: formato que deberá cumplir la expresión regular. Hasta 80 caracteres.
- **AssetType:** modela un tipo de activo. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a un tipo de activo.
 - *descr*: descripción del tipo activo. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre el tipo de activo.
- **ModelTypeAttr:** modela un atributo de tipo modelo para un tipo de activo con un tipo de unidad. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a un atributo de tipo modelo.
 - *name*: descripción del atributo de tipo modelo. Hasta 80 caracteres.
 - *fktype*: entero que hace referencia al tipo de activo.
 - *fkudtype*: entero que hace referencia al tipo de unidad.
- **AssetTypeAttr:** modela un atributo de tipo activo para un tipo de activo con un tipo de unidad. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a un atributo de tipo activo.
 - *name*: descripción del atributo de tipo activo. Hasta 80 caracteres.
 - *fktype*: entero que hace referencia al tipo de activo.
 - *fkudtype*: entero que hace referencia al tipo de unidad.
- **Mark:** modela una marca. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una marca.
 - *descr*: descripción de la marca. Hasta 80 caracteres.
 - *pn*: descripción del número de modelo. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la marca.
- **ModelAttr:** modela un atributo de un modelo. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a un atributo de un modelo.
 - *val*: valor asignado al atributo de modelo. Hasta 80 caracteres.

- *fk*: entero que hace referencia al modelo.
- *fkattr*: entero que hace referencia al atributo de tipo modelo.
- *fkud*: entero que representa la unidad elegida para el atributo de modelo.
- **Status**: modela un estado de recurso. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a un estado de recurso.
 - *descr*: descripción del estado de recurso. Hasta 80 caracteres.
- **Asset**: modela un activo. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente al activo.
 - *cts*: instante de creación del activo. El actual por defecto.
 - *lmts*: instante de última modificación realizar sobre el activo. El actual por defecto.
 - *lmuname*: nombre del equipo. Hasta 20 caracteres.
 - *sn*: número de serie. Hasta 80 caracteres.
 - *id*: número identificador del activo. Hasta 80 caracteres.
 - *descr*: descripción del activo. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre el activo.
 - *cun*: usuario que dio de alta el activo. Hasta 80 caracteres.
 - *run*: usuario responsable del activo. Hasta 80 caracteres.
 - *fkasset*: entero que hace referencia al activo padre del activo.
 - *img*: imagen adjuntada al activo.
 - *fkmodel*: entero que hace referencia al modelo del activo.
 - *fkstat*: entero que hace referencia al estado de recurso.
 - *fkloc*: entero que hace referencia al nivel de localización 2 (LL2) del activo.
 - *fkoul*: entero que hace referencia al nivel de ubicación 3 (OUL3) del activo.
- **AssetAttr**: modela un atributo de activo. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente al atributo de activo.
 - *val*: valor asignado al atributo de activo. Hasta 80 caracteres.
 - *fk*: entero que hace referencia al activo.
 - *fkattr*: entero que hace referencia al atributo de tipo activo.
 - *fkud*: entero que representa la unidad elegida para el atributo de activo.
- **OutputType**: modela un tipo de salida. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente al tipo de salida.
 - *descr*: descripción del tipo de salida. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre el tipo de salida.
- **Output**: modela una salida. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la salida.
 - *ots*: instante de creación de la salida. El actual por defecto.
 - *src*: entero que hace referencia al nivel de localización 2 (LL2) de la salida. Representa el origen.
 - *sun*: persona emisora de la salida. Hasta 80 caracteres.
 - *dst*: entero que hace referencia al nivel de localización 2 (LL2) de la salida. Representa el destino.
 - *run*: persona receptora de la salida. Hasta 80 caracteres.
 - *oun*: persona responsable de la salida. Hasta 80 caracteres.
 - *fktype*: entero que hace referencia al tipo de salida.
 - *rem*: campo de texto para realizar notas adicionales sobre la salida.
 - *descr*: descripción de la salida. Hasta 80 caracteres.



3 Diseño de los módulos de activos, adquisiciones y común de GTI

- *dst_alt*: destino alternativo de la salida. Hasta 80 caracteres.
- **InputType**: modela un tipo de entrada. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente al tipo de entrada.
 - *descr*: descripción del tipo de entrada. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre el tipo de entrada.
- **Input**: modela una entrada. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la entrada.
 - *its*: instante de creación de la entrada. El actual por defecto.
 - *src*: entero que hace referencia al nivel de localización 2 (LL2) de la entrada. Representa el origen.
 - *sun*: persona emisora de la entrada. Hasta 80 caracteres.
 - *dst*: entero que hace referencia al nivel de localización 2 (LL2) de la entrada. Representa el destino.
 - *run*: persona receptora de la entrada. Hasta 80 caracteres.
 - *iun*: persona responsable de la entrada. Hasta 80 caracteres.
 - *fkout*: entero que hace referencia a la salida.
 - *fkpur*: entero que hace referencia a la adquisición.
 - *fktype*: entero que hace referencia al tipo de entrada.
 - *rem*: campo de texto para realizar notas adicionales sobre la entrada.
 - *descr*: descripción de la entrada. Hasta 80 caracteres.
 - *src_alt*: origen alternativo de la entrada. Hasta 80 caracteres.
- **History**: modela una traza. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente la traza.
 - *mts*: instante de generación de la traza. El actual por defecto.
 - *mun*: usuario responsable de la generación de la traza. Hasta 40 caracteres.
 - *descr*: descripción de la traza.
 - *type*: tipo de la traza. Puede ser de creación o de modificación.
 - *fkasset*: entero que hace referencia al activo.
- **Entry**: modela una anotación. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la anotación.
 - *cu*: usuario responsable de generar la anotación. Hasta 80 caracteres.
 - *cts*: instante de creación de la anotación. El actual por defecto.
 - *rem*: campo de texto para realizar notas adicionales sobre la anotación.
 - *fkasset*: entero que hace referencia al activo.

El modelo físico de datos se compone además de una serie de relaciones:

- **REL_Offer_Model**: modela una relación entre una oferta (*Offer*) y un modelo (*Model*) que indica los modelos que incluye una determinada oferta. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkmodel*: entero que hace referencia al modelo.
 - *fkoffer*: entero que hace referencia a la oferta.
 - *price*: precio unitario del modelo.
 - *num*: número total de elementos del modelo.
- **REL_Purchase_Model**: modela una relación entre una adquisición (*Purchase*) y un modelo (*Model*) que indica los modelos que incluye una determinada adquisición. Los campos que

lo componen son:

- *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkmodel*: entero que hace referencia al modelo.
 - *fkpur*: entero que hace referencia a la adquisición.
 - *price*: precio unitario del modelo.
 - *num*: número total de elementos del modelo.
- **REL_Output_Asset**: modela una relación entre una salida (*Output*) y un activo (*Asset*) que indica los activos que incluye una determinada salida. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkasset*: entero que hace referencia al activo.
 - *fkoutput*: entero que hace referencia a la salida.
 - **REL_Input_Asset**: modela una relación entre una entrada (*Input*) y un activo (*Asset*) que indica los activos que incluye una determinada entrada. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkasset*: entero que hace referencia al activo.
 - *fkinput*: entero que hace referencia a la entrada.
 - **REL_Gad_Scope**: modela una relación entre un grupo de permisos (gestores de compras) y un ámbito (*Scope*). Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkscope*: entero que hace referencia al ámbito.
 - *uid*: identificador que representa al grupo de permisos.
 - **REL_Gin_Scope**: modela una relación entre un grupo de permisos (gestores de inventario) y un ámbito (*Scope*). Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkscope*: entero que hace referencia al ámbito.
 - *uid*: identificador que representa al grupo de permisos.
 - **REL_Ate_Scope**: modela una relación entre un grupo de permisos (técnicos avanzados) y un ámbito (*Scope*). Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkscope*: entero que hace referencia al ámbito.
 - *uid*: identificador que representa al grupo de permisos.
 - **REL_Tec_Scope**: modela una relación entre un grupo de permisos (técnicos) y un ámbito (*Scope*). Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkscope*: entero que hace referencia al ámbito.
 - *uid*: identificador que representa al grupo de permisos.

3.3.2 Modelo físico de datos módulo gti

En este apartado se describe el modelo físico de datos utilizado para el módulo común de GTI. Para ello, se presenta el diagrama del modelo y se detallan cada una de las tablas y campos que las componen.

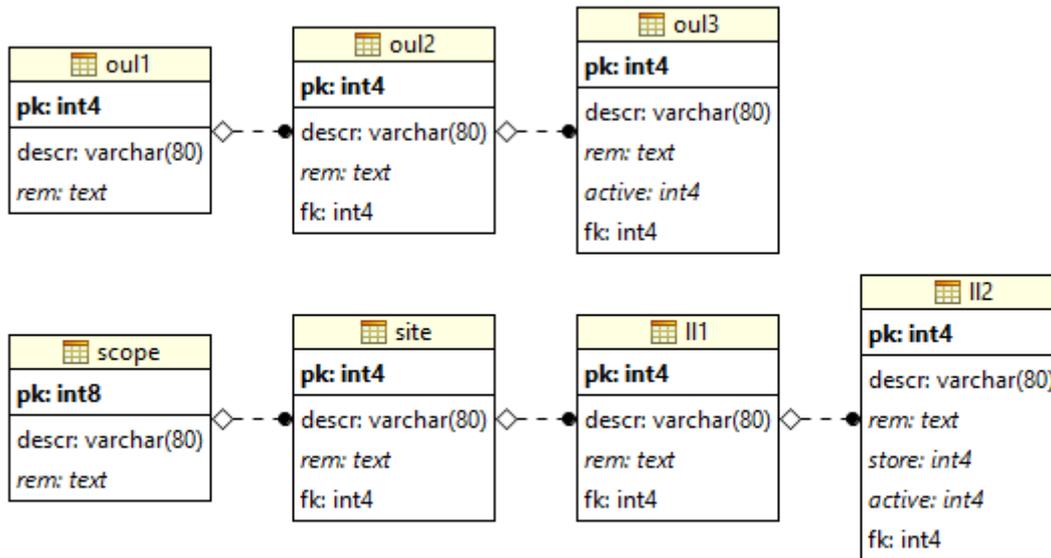


Ilustración 46: Modelo físico de datos módulo gti

Las tablas que lo componen son las siguientes:

- **OUL1:** modela una unidad organizativa de nivel 1. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una unidad organizativa de nivel 1.
 - *descr*: descripción de la unidad organizativa de nivel 1. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la unidad organizativa nivel 1.
- **OUL2:** modela una unidad organizativa de nivel 2. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una unidad organizativa de nivel 2.
 - *descr*: descripción de la unidad organizativa de nivel 2. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la unidad organizativa nivel 2.
 - *fk*: entero que hace referencia a la unidad organizativa de nivel 1.
- **OUL3:** modela una unidad organizativa de nivel 3. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una unidad organizativa de nivel 3.
 - *descr*: descripción de la unidad organizativa de nivel 3. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la unidad organizativa nivel 3.
 - *fk*: entero que hace referencia a la unidad organizativa de nivel 2.
 - *active*: entero que representa la actividad de la unidad organizativa de nivel 3. Puede ser 1 ó 0.
- **Site:** modela un edificio. Los campos que lo componen son:

- *pk*: entero generado automáticamente que identifica únicamente a un edificio.
 - *descr*: descripción del edificio. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre el edificio.
 - *fk*: entero que hace referencia al ámbito.
- **LL1**: modela una localización de nivel 1. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una localización nivel 1.
 - *descr*: descripción de la localización nivel 1. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la localización nivel 1.
 - *fk*: entero que hace referencia al edificio.
- **LL2**: modela una localización de nivel 2. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una localización nivel 2.
 - *descr*: descripción de la localización nivel 2. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la localización nivel 2.
 - *fk*: entero que hace referencia a la localización de nivel 1.
 - *active*: entero que representa la actividad de la localización nivel 2. Puede ser 1 ó 0.
- **Scope**: modela un ámbito. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a un ámbito.
 - *descr*: descripción del ámbito. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre el ámbito.

3.3.3 Modelo físico de datos módulo pp

En este apartado se describe el modelo físico de datos utilizado para el módulo de gestión de adquisiciones de GTI. Para ello, se presenta el diagrama del modelo y se detallan cada una de las tablas y campos que las componen.

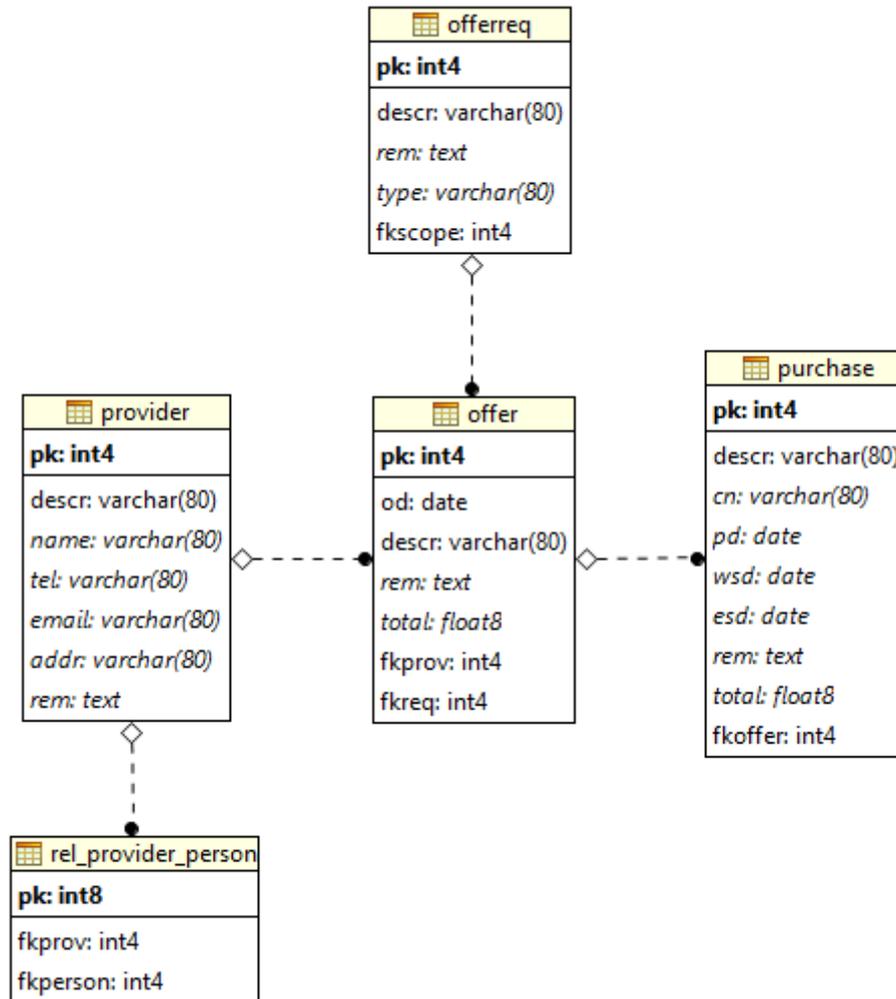


Ilustración 47: Modelo físico de datos módulo pp

Las tablas que lo componen son las siguientes:

- **Provider:** modela un proveedor. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a un proveedor.
 - *descr*: descripción del proveedor. Hasta 80 caracteres.
 - *name*: nombre del proveedor. Hasta 80 caracteres.
 - *tel*: teléfono de contacto del proveedor. Hasta 80 caracteres.
 - *email*: email de contacto del proveedor. Hasta 80 caracteres.
 - *addr*: dirección del proveedor. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre el proveedor.
- **OfferReq:** modela una petición de oferta. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una petición de oferta.
 - *descr*: descripción de la petición de oferta. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la petición de oferta.
 - *type*: tipo de la petición de oferta. En este TFG será “am”.
 - *fkscope*: entero que hace referencia al ámbito.

- **Offer:** modela una oferta. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una oferta.
 - *od*: fecha de creación de la oferta. La actual por defecto.
 - *descr*: descripción de la oferta. Hasta 80 caracteres.
 - *rem*: campo de texto para realizar notas adicionales sobre la oferta.
 - *total*: precio total de la oferta según sus elementos.
 - *fkprov*: entero que hace referencia al proveedor.
 - *fkreq*: entero que hace referencia a la petición de oferta.

- **Purchase:** modela una adquisición. Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a una adquisición.
 - *descr*: descripción de la adquisición. Hasta 80 caracteres.
 - *cn*: número de contrato. Hasta 80 caracteres.
 - *pd*: fecha de creación de la adquisición. La actual por defecto.
 - *wsd*: fecha de inicio de garantía de la adquisición.
 - *esd*: fecha de fin de garantía de la adquisición.
 - *rem*: campo de texto para realizar notas adicionales sobre la adquisición.
 - *total*: precio total de la adquisición según sus elementos.
 - *fkoffer*: entero que hace referencia a la oferta.

- **REL_Provider_Person:** modela la relación entre un proveedor (*Provider*) y un usuario (*percon_person*). Los campos que lo componen son:
 - *pk*: entero generado automáticamente que identifica únicamente a la relación.
 - *fkprov*: entero que hace referencia al proveedor.
 - *fkperson*: entero que hace referencia al usuario.

3.4 Arquitectura de módulos del sistema

3.4.1 Arquitectura de módulos del sistema módulo am

Para hacer cumplir todos los requisitos y funcionalidades propuestos, se han creado los siguientes eventos y funciones.

3.4.1.1 Eventos

Entre los eventos, podemos distinguir aquellos programados en el código ASL (llamados acciones) y otros programados en el código SQL.

3.4.1.1.1 Eventos implementados en ASL

En este apartado, se van a describir aquellos eventos que han sido codificados en el lenguaje ASL. Estos eventos son tratados directamente desde la interfaz de usuario.

3.4.1.1.1.1 Evento “Eliminación de un tipo de entrada predefinida”

Este evento se encarga de comprobar que no se elimine un tipo de entrada del tipo “Adquisición”, “Interna” o “Devolución”. Para ello, se comprueba que las claves primarias asociadas a los tipos predefinidos no son las que se están intentando eliminar. En caso de ser uno de estos tres tipos, se muestra un mensaje de error.



3.4.1.1.1.2 Evento “Eliminación de un tipo de salida predefinida”

Este evento se encarga de comprobar que no se elimine un tipo de salida del tipo “Interna”, “Préstamo” o “Baja”. Para ello, se comprueba que las claves primarias asociadas a los tipos predefinidos no son las que se están intentando eliminar. En caso de ser uno de estos tres tipos, se muestra un mensaje de error.

3.4.1.1.1.3 Evento “Eliminación de un estado de los recursos”

Este evento se encarga de comprobar que no se elimine un estado de los recursos del tipo “Activo” o “Baja”. Para ello, se comprueba que las claves primarias asociadas a los estados predefinidos no son las que se están intentando eliminar. En caso de ser uno de estos dos estados, se muestra un mensaje de error.

3.4.1.1.1.4 Evento “Asignación de permisos de gestor de adquisiciones”

Este evento se encarga de asegurar que un usuario, al otorgarle permisos en un ámbito para el gestor de adquisiciones, no aparezca más de una vez. Para ello, se comprueba que dicho usuario no se encuentre en la tabla que relaciona el ámbito con los permisos para el gestor de adquisiciones. En caso de que ese usuario ya exista en dicho ámbito para el gestor de adquisiciones, se muestra un mensaje de error.

3.4.1.1.1.5 Evento “Asignación de permisos de gestor de inventario”

Este evento se encarga de asegurar que un usuario, al otorgarle permisos en un ámbito para el gestor de inventario, no aparezca más de una vez. Para ello, se comprueba que dicho usuario no se encuentre en la tabla que relaciona el ámbito con los permisos para el gestor de inventario. En caso de que ese usuario ya exista en dicho ámbito para el gestor de inventario, se muestra un mensaje de error.

3.4.1.1.1.6 Evento “Asignación de permisos de técnico avanzado”

Este evento se encarga de asegurar que un usuario, al otorgarle permisos en un ámbito para el técnico avanzado, no aparezca más de una vez. Para ello, se comprueba que dicho usuario no se encuentre en la tabla que relaciona el ámbito con los permisos para el técnico avanzado. En caso de que ese usuario ya exista en dicho ámbito para el técnico avanzado, se muestra un mensaje de error.

3.4.1.1.1.7 Evento “Asignación de permisos de técnico”

Este evento se encarga de asegurar que un usuario, al otorgarle permisos en un ámbito para el técnico, no aparezca más de una vez. Para ello, se comprueba que dicho usuario no se encuentre en la tabla que relaciona el ámbito con los permisos para el técnico. En caso de que ese usuario ya exista en dicho ámbito para el técnico, se muestra un mensaje de error.

3.4.1.1.1.8 Evento “Inserción de un nuevo activo en una entrada”

Este evento se encarga de llamar a la función `FUNC_Asset_aux(int,int,varchar)` (ver apartado 4.2.7).

3.4.1.1.1.9 Evento “Inserción de un activo en una salida”

Este evento se encarga de comprobar que un mismo activo no pueda aparecer por duplicado en una misma salida. Para ello, se comprueba que la clave primaria de dicho activo no se encuentre para la salida elegida. En caso de que ese activo ya exista en dicha salida, se muestra un mensaje de error.

3.4.1.1.1.10 Evento “Actualización de un activo”

Este evento se encarga de actualizar el usuario que hace la modificación de un activo para la traza de dicho activo con el usuario actual. Esto se hace desde la aplicación, ya que desde la base de datos no se puede recuperar el valor del usuario actual a través de un disparador.

3.4.1.1.1.11 Evento “Actualización de un número de serie en un activo”

Este evento se encarga de comprobar que el número de serie esté completo para un determinado activo antes de crearlo o de actualizarlo. Esta comprobación posibilita eliminar un activo sin necesidad de asignarle un número de serie previamente.

3.4.1.1.2 Eventos implementados en SQL

En este apartado, se van a describir aquellos eventos que han sido codificados en el lenguaje SQL. Estos eventos son tratados en la Base de Datos del módulo.

3.4.1.1.2.1 Evento TRI_ModelTypeAttr_after

Este evento se encarga de lanzar la función **PROC_ModelTypeAttr_after()** después de producirse una inserción en la tabla *ModelTypeAttr*.

3.4.1.1.2.2 Evento TRI_AssetTypeAttr_after

Este evento se encarga de lanzar la función **PROC_AssetTypeAttr_after()** después de producirse una inserción en la tabla *AssetTypeAttr*.

3.4.1.1.2.3 Evento TRI_Model_after

Este evento se encarga de lanzar la función **PROC_Model_after()** después de producirse una inserción en la tabla *Model*.

3.4.1.1.2.4 Evento TRI_Asset_after

Este evento se encarga de lanzar la función **PROC_Asset_after()** después de producirse una inserción en la tabla *Asset*.

3.4.1.1.2.5 Evento TRI_Input_after

Este evento se encarga de lanzar la función **PROC_Input_after()** después de producirse una inserción en la tabla *Input*.

3.4.1.1.2.6 Evento TRI_Asset_update

Este evento se encarga de lanzar la función **PROC_Asset_update()** antes de producirse una actualización en la tabla *Asset*.

3.4.1.1.2.7 Evento TRI_Asset_history

Este evento se encarga de lanzar la función **PROC_Asset_history()** después de producirse una actualización en la tabla *Asset*.

3.4.1.1.2.8 Evento TRI_Purchase_after

Este evento se encarga de lanzar la función **PROC_Purchase_after()** después de producirse una



3.4.1.2 Funciones

En este apartado, se van a describir las funciones creadas en la Base de Datos del módulo.

3.4.1.2.1 Función PROC_ModelTypeAttr_after()

Esta función se ejecuta tras producirse una inserción en la tabla *ModelTypeAttr*. En su ejecución, se inserta una nueva entrada en la tabla *ModelAttr* con la clave externa con *ModelTypeAttr*, la nueva clave primaria. De esta forma, se asegura que, si se crea el tipo de atributo, exista su atributo correspondiente en los modelos ya creados.

3.4.1.2.2 Función PROC_AssetTypeAttr_after()

Esta función se ejecuta tras producirse una inserción en la tabla *AssetTypeAttr*. En su ejecución, se inserta una nueva entrada en la tabla *AssetAttr* con la clave externa con *AssetTypeAttr*, la nueva clave primaria. De esta forma, se asegura que, si se crea el tipo de atributo, exista su atributo correspondiente en los activos ya creados.

3.4.1.2.3 Función PROC_Model_after()

Esta función se ejecuta tras producirse una inserción en la tabla *Model*. En su ejecución, se inserta una nueva entrada en la tabla *ModelAttr* con la clave externa con *Model*, la nueva clave primaria. De esta forma, se asegura que, si se crea un modelo, exista su atributo correspondiente.

3.4.1.2.4 Función PROC_Asset_after()

Esta función se ejecuta tras producirse una inserción en la tabla *Asset*. En su ejecución, se inserta una nueva entrada en la tabla *AssetAttr* con la clave externa con *Asset*, la nueva clave primaria. De esta forma, se asegura que, si se crea un activo, exista su atributo correspondiente.

3.4.1.2.5 Función PROC_Input_after()

Esta función se ejecuta tras producirse una inserción en la tabla *Input*. En su ejecución, se van a distinguir dos tipos de entradas:

- Tipo de entrada es igual a “Adquisición”: se llamará a la función **FUNC_Input_aux(int,int,int,int,varchar)** tantas veces como entradas haya en la tabla *REL_Purchase_Model* para la adquisición seleccionada.
- Tipo de entrada es distinto de “Adquisición”: se añaden tantas entradas en la tabla *REL_Input_Asset* como activos haya en la salida seleccionada.

3.4.1.2.6 Función FUNC_Input_aux(int,int,int,int,varchar,varchar)

Esta función se ejecuta tras producirse una inserción en la tabla *Input* de tipo “Adquisición”. En su ejecución, se van a distinguir cinco argumentos:

- Modelo: argumento de tipo Integer que determina el modelo del activo.
- Número: argumento de tipo Integer que determina la cantidad de activos del mismo modelo.
- Input: argumento de tipo Integer que determina la clave primaria de la tabla *Input* a trabajar.
- Localización: argumento de tipo Integer que determina la localización de los nuevos activos.
- Responsable: argumento de tipo Varchar que determina el responsable de los nuevos activos.
- Usuario: argumento de tipo Varchar que determina el responsable de la creación de los nuevos activos.

Con estos argumentos, se van a añadir tantas entradas en la tabla *Asset* del modelo “argumento-modelo” como el “argumento-número” indique, poniendo como localización “argumento-localización” y como responsable “argumento-responsable”, dejando a “null” los campos “Nuevo de serie” e “Identificador del activo”.

Posteriormente, se referenciarán los nuevos activos en la tabla *REL_Input_Model* para la entrada indicada en el “argumento-input” del modelo indicado en el “argumento-modelo”.

Finalmente, se añadirá una entrada en la tabla *History* por cada nuevo activo creado con los datos correspondientes con el responsable “argumento-usuario”.

3.4.1.2.7 Función FUNC_Asset_aux(int,int,varchar)

Esta función se ejecuta tras producirse una inserción de un activo en la interfaz “Gestión de Entradas”. En su ejecución, se van a distinguir tres argumentos:

- Asset: argumento de tipo Integer que determina la clave primaria de la tabla *Asset*.
- Input: argumento de tipo Integer que determina la clave primaria de la tabla *Input*.
- Responsable: argumento de tipo Varchar que determina el responsable de la inserción.

Con estos argumentos, se va a añadir una entrada en la tabla *REL_Input_Asset* para la entrada indicada en el “argumento-input” del activo indicado en el “argumento-asset”.

Posteriormente, se van a modificar de la tabla *Asset* los atributos: localización, responsable, estado del activo y ubicación con los indicados en la entrada.

Finalmente, se añadirá una entrada en la tabla *History* para el activo creado con los datos correspondientes.

3.4.1.2.8 Función PROC_Asset_update()

Esta función se ejecuta antes de producirse una actualización en la tabla *Asset*. En su ejecución, se va a modificar el campo “lmts” (last modified timestamp) con la hora actual.

3.4.1.2.9 Función PROC_Asset_history()

Esta función se ejecuta tras producirse una actualización en la tabla *Asset*. En su ejecución, se copiarán los atributos que han sido modificados en el activo actualizado de la tabla *Asset*, creando una nueva entrada en la tabla *History*.



3.4.1.2.10 Función PROC_Purchase_after()

Esta función se ejecuta tras producirse una inserción en la tabla *Purchase*. En su ejecución, se van a copiar todas las entradas de la *REL_Offer_Model* en la tabla *REL_Purchase_Model*, sabiendo que compra se ha hecho para una determinada oferta.

3.4.2 Arquitectura de módulos del sistema módulo pp

Para hacer cumplir todos los requisitos y funcionalidades propuestos, se han creado los siguientes eventos y funciones.

3.4.2.1 Eventos

Entre los eventos, se van a definir aquellos programados en el código ASL (llamados acciones).

3.4.2.1.1 Eventos implementados en ASL

En este apartado, se van a describir aquellos eventos que han sido codificados en el lenguaje ASL. Estos eventos son tratados directamente desde la interfaz de usuario.

3.4.2.1.1.1 Evento “Inserción de un nuevo contacto para un proveedor”

Este evento se encarga de asegurar que un contacto no pueda aparecer por duplicado para un mismo proveedor. Para ello, se comprueba que la clave primaria del contacto a añadir no se encuentre para el proveedor elegido. En caso de que ese contacto ya exista en dicho proveedor, se muestra un mensaje de error.

3.4.2.1.1.2 Evento “Inserción de un nuevo modelo en una oferta”

Este evento se encarga de asegurar que un modelo no pueda aparecer por duplicado para una misma oferta. Para ello, se comprueba que la clave primaria del modelo a añadir no se encuentre para la oferta elegida. En caso de que ese modelo ya exista en dicha oferta, se muestra un mensaje de error.

3.4.2.1.1.3 Evento “Inserción de un nuevo modelo en una compra”

Este evento se encarga de asegurar que un modelo no pueda aparecer por duplicado para una misma compra. Para ello, se comprueba que la clave primaria del modelo a añadir no se encuentre para la compra elegida. En caso de que ese modelo ya exista en dicha compra, se muestra un mensaje de error.

3.4.2.1.1.4 Evento “Cálculo de la imputación total de una oferta”

Este evento se encarga de calcular el total de dinero que supone todos los elementos relacionados a una determinada oferta. Esto se comprobará después de añadir, borrar o actualizar un elemento para dicha oferta.

3.4.2.1.1.5 Evento “Cálculo de la imputación total de una compra”

Este evento se encarga de calcular el total de dinero que supone todos los elementos relacionados a una determinada compra. Esto se comprobará después de añadir, borrar o actualizar un elemento para dicha compra.

3.5 Diseño de las interfaces de usuario

3.5.1 Diseño de las interfaces de usuario módulo am

En este apartado, se van a detallar las unidades de interacción para cada rol del módulo de inventario de GTI, junto al diseño de dichas unidades de interacción. Hay que tener que este módulo se compone de dos módulos auxiliares: módulo común y el módulo de adquisiciones.

A continuación, se muestra la tabla que recoge qué unidades de interacción son accedidas para cada rol:

	Administrador del módulo de inventario	Gestor de inventario	Técnico avanzado	Técnico	Usuario regular	Usuario sin acceso
Administración de permisos	cpermmnag1					
Tipos de entrada	cinputtype					
Tipos de salida	coutputtype					
Estados de los recursos	cstatus					
Tipos de unidades y unidades	cunit					
Tipos de elementos y atributos	cassettype1					
Marcas, modelos y atributos		cmarkmodell				
Gestión de entradas		cinputmanag1	cinputmanagate1			
Gestión de salidas		coutputmanag1	coutputmanagate1			
Gestión de inventario		cassetmanag				
Gestión de inventario por tipo de recurso		cassetmanagtype				
Edición de activos		cassetedit1				
Gestión de activos		cassetmanag1				
Estadísticas de tipos de activos por sede		cgraphtype				
Estadísticas de estados y marcas de activos por sede y tipo		cgraphstatusmark1				
Gestión de inventario reducida			cassetmanagredate	cassetmanagredtec		
Edición de activos reducida			casseteditredate1	casseteditredtec1		
Consulta de recursos asignados	cassignresourcequery1	cassignresourcequery1	cassignresourcequery1	cassignresourcequery1	cassignresourcequery1	
Visualización recurso						cresourceview1
Visualización de entrada						cinputview
Visualización de salida						coutputview

Tabla 2: Tabla Rol - Unidad de interacción



3.5.1.1 Administrador del módulo de inventario

El administrador del módulo de inventario puede realizar las siguientes tareas:

- Establecer gestores de adquisiciones, de inventario y técnicos por ámbitos.
- Definir tipos de entradas y salidas.
- Definir tipos de recursos y sus campos personalizados.
- Consultar sus recursos asignados.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.1.1.1 Administración de permisos – *gti.am.cpermmanag1*

La interfaz *administración de permisos* se encuentra implementada a través de la estructura *gti.am.cpermmanag1*. A continuación, se muestra su diagrama ER anotado:

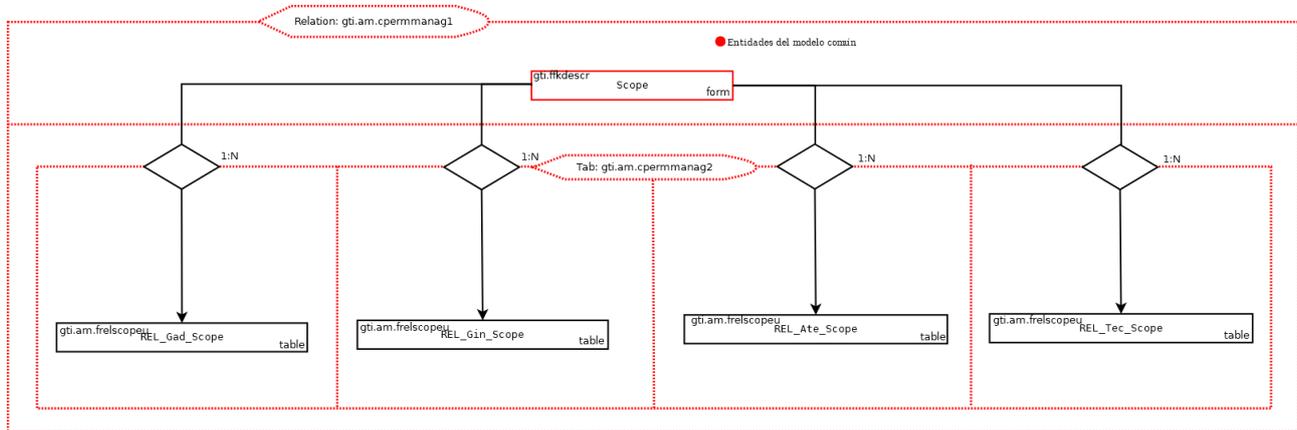


Ilustración 48: ERD anotado Administración de permisos

3.5.1.1.2 Tipos de entrada – *gti.am.cinputtype*

La interfaz *tipos de entrada* se encuentra implementada a través de la estructura *gti.am.cinputtype*. A continuación, se muestra su diagrama ER anotado:

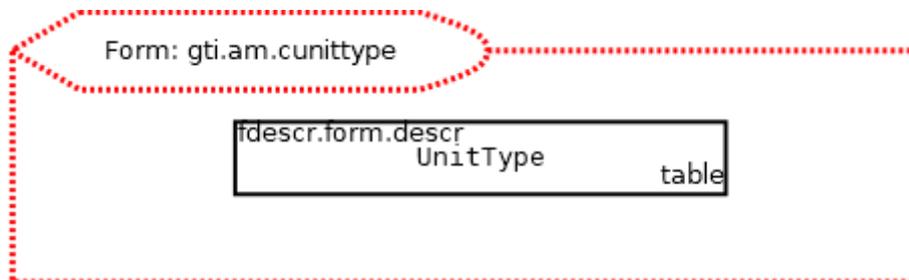


Ilustración 49: ERD anotado Tipos de entrada

3.5.1.1.3 Tipos de salida – *gti.am.coutputtype*

La interfaz *tipos de salida* se encuentra implementada a través de la estructura *gti.am.coutputtype*. A continuación, se muestra su diagrama ER anotado:

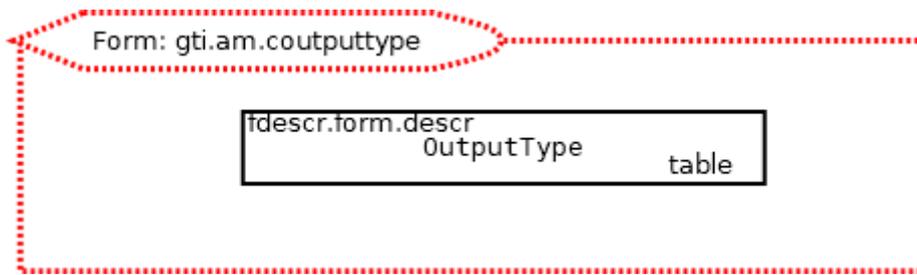


Ilustración 50: ERD anotado Tipos de salida

3.5.1.1.4 Estados de los recursos – *gti.am.cstatus*

La interfaz *estados de los recursos* se encuentra implementada a través de la estructura *gti.am.cstatus*. A continuación, se muestra su diagrama ER anotado:

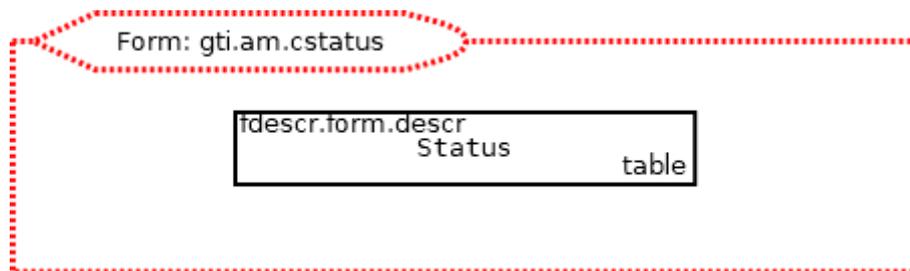


Ilustración 51: ERD anotado Estados de los recursos

3.5.1.1.5 Tipos de unidades y unidades – *gti.am.cunit*

La interfaz *tipos de unidades y unidades* se encuentra implementada a través de la estructura *gti.am.cunit*. A continuación, se muestra su diagrama ER anotado:

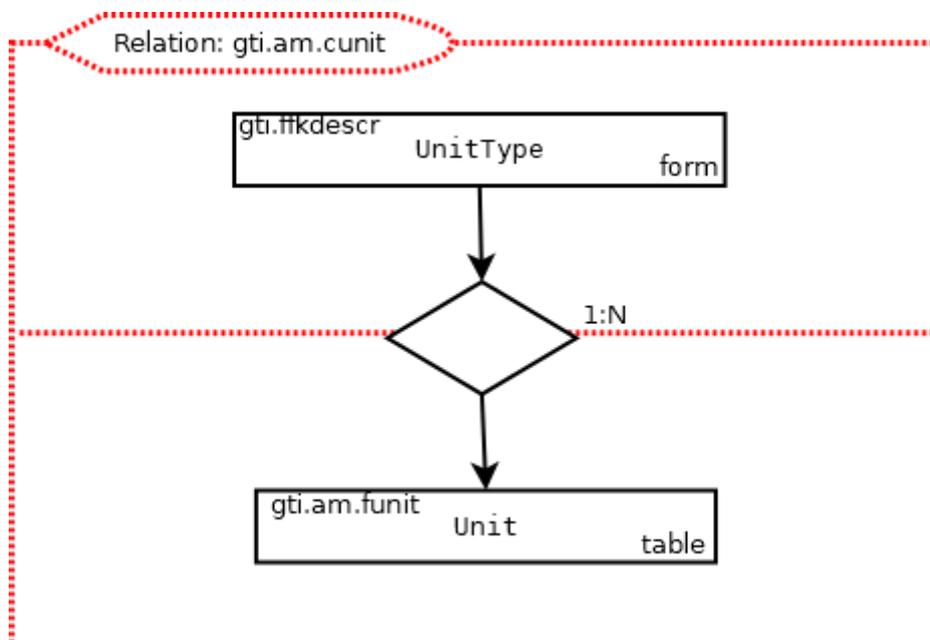


Ilustración 52: ERD anotado Tipos de unidades y unidades

3.5.1.1.6 Tipos de elementos y atributos – *gti.am.cassettype1*

La interfaz *tipos de elementos y atributos* se encuentra implementada a través de la estructura *gti.am.cassettype1*. A continuación, se muestra su diagrama ER anotado:

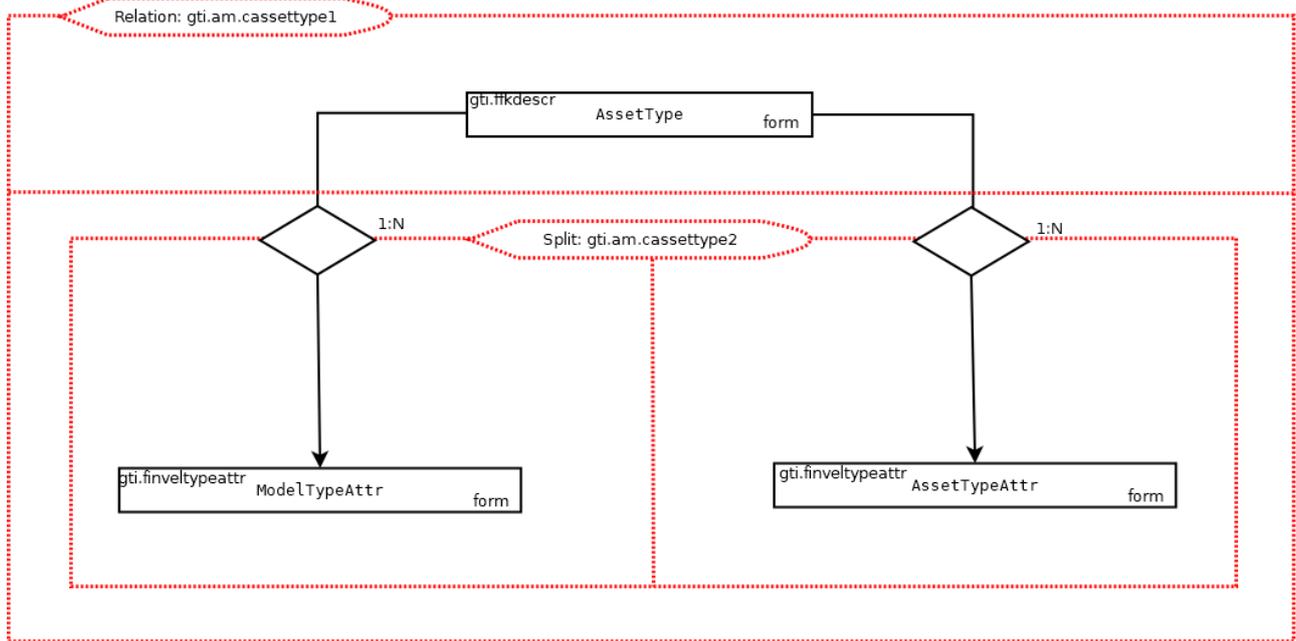


Ilustración 53: ERD anotado Tipos de elementos y atributos

3.5.1.1.7 Consulta de recursos asignados – *cassignresourcequery1*

La interfaz *consulta de recursos asignados* se encuentra implementada a través de la estructura *cassignresourcequery1*. A continuación, se muestra su diagrama ER anotado:

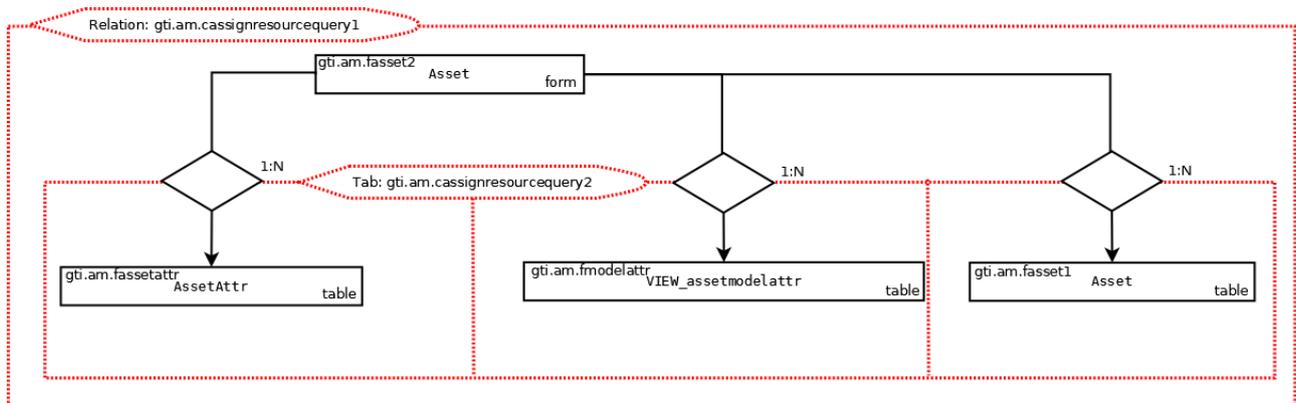


Ilustración 54: ERD anotado Consulta de recursos asignados

Esta interfaz, a su vez, en su pestaña *Recursos Hijos*, llama a otra interfaz igual que la mostrada anteriormente, pero sin tener en cuenta las restricciones de usuario o de ser activo padre. Su diagrama ER anotado es el siguiente:

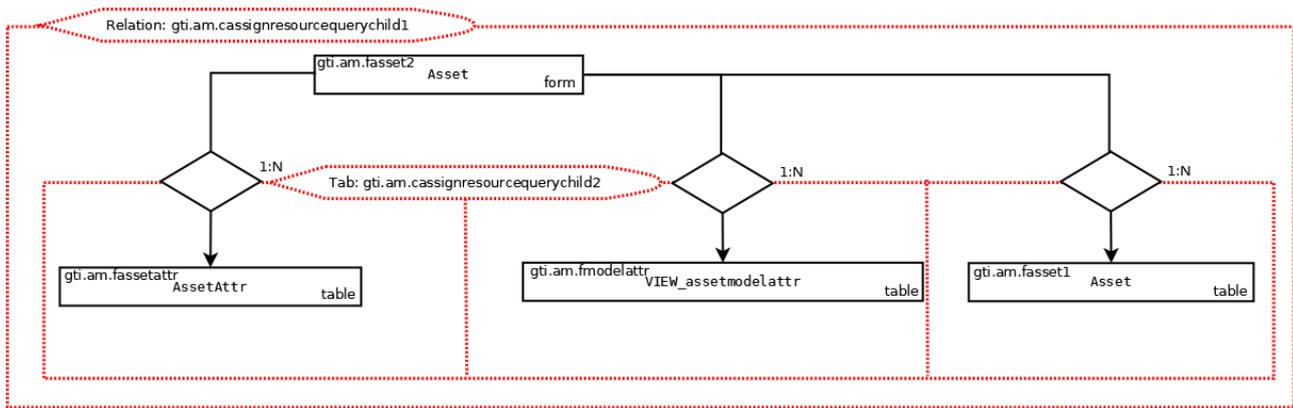


Ilustración 55: ERD anotado Consulta de recursos asignados - Recursos hijos

3.5.1.2 Gestor de inventario

El gestor de inventario puede realizar las siguientes tareas:

- Gestionar marcas y modelos con sus atributos.
- Gestionar proveedores y sus contactos.
- Solicitar ofertas a los proveedores.
- Recibir ofertas a sus solicitudes.
- Realizar adquisiciones contra una oferta de un proveedor.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.1.2.1 Marcas, modelos y atributos – *gti.am.cmarkmodel1*

La interfaz *marcas, modelos y atributos* se encuentra implementada a través de la estructura *gti.am.cmarkmodel1*. Su diagrama ER anotado se encuentra en el apartado 5.3.4.

3.5.1.2.2 Gestión de entradas – *gti.am.cinputmanag1*

La interfaz *gestión de entradas* se encuentra implementada a través de la estructura *gti.am.cinputmanag1*. A continuación, se muestra su diagrama ER anotado:



3 Diseño de los módulos de activos, adquisiciones y común de GTI

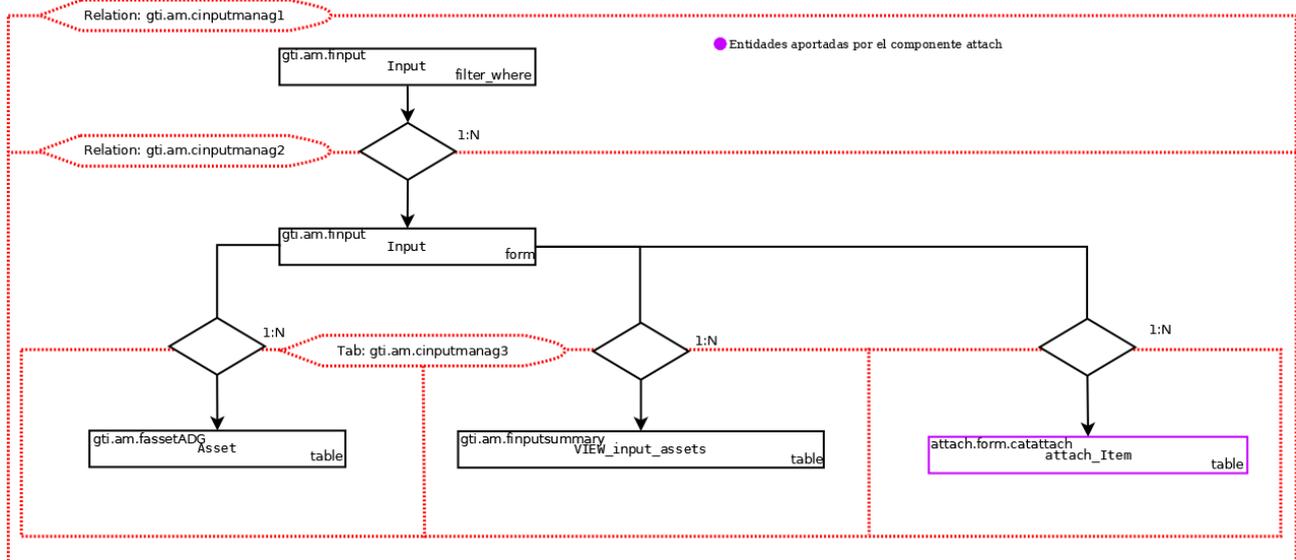


Ilustración 56: ERD anotado Gestión de entradas

3.5.1.2.3 Gestión de salidas – gti.am.coutputmanag1

La interfaz *gestión de salidas* se encuentra implementada a través de la estructura *gti.am.cgestsall*. A continuación, se muestra su diagrama ER anotado:

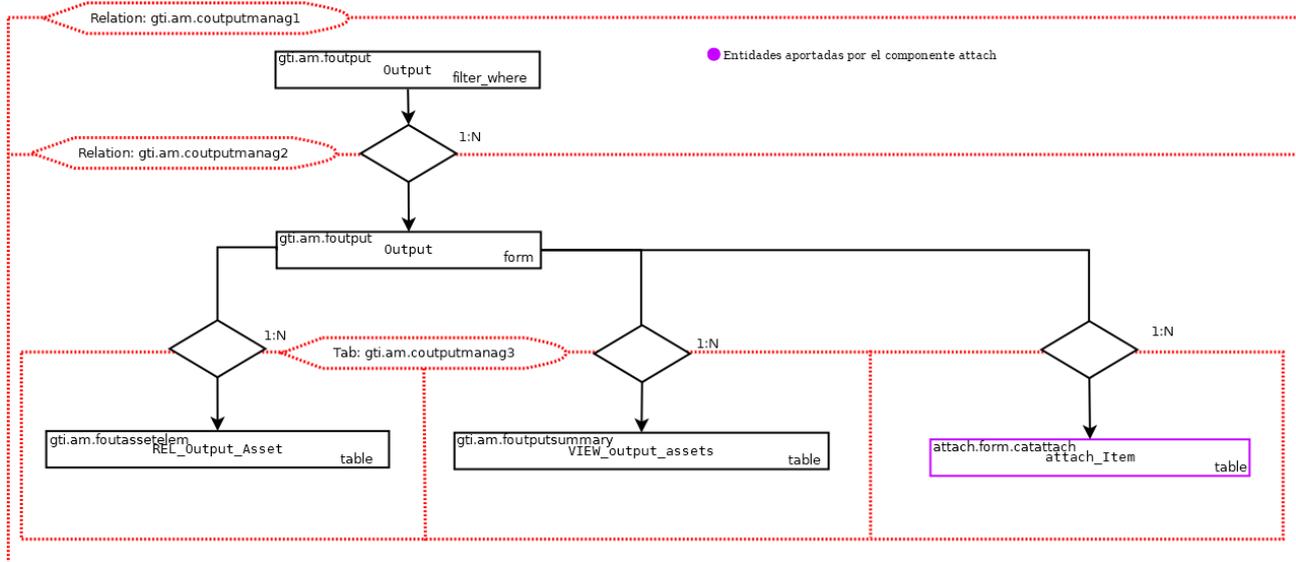


Ilustración 57: ERD anotado Gestión de salidas

3.5.1.2.4 Gestión de inventario – gti.am.cassetmanag

La interfaz *gestión de inventario* se encuentra implementada a través de la estructura *gti.am.cassetmanag*. A continuación, se muestra su diagrama ER anotado:

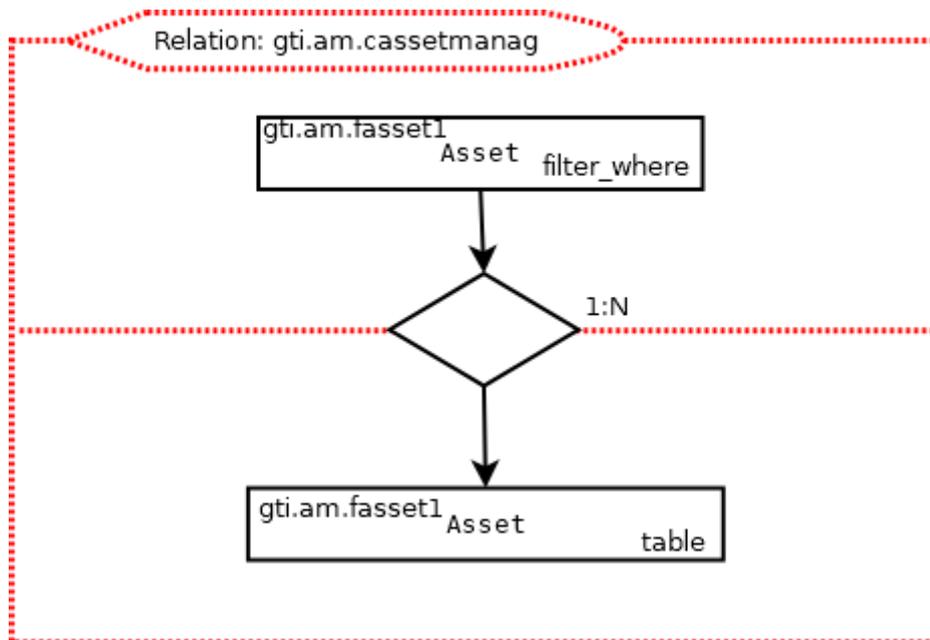


Ilustración 58: ERD anotado Gestión de inventario

3.5.1.2.5 Gestión de inventario por tipo recurso – *gti.am.cassetmanagtype*

La interfaz *gestión de inventario por tipo de recurso* se encuentra implementada a través de la estructura *gti.am.cassetmanagtype*. A continuación, se muestra su diagrama ER anotado:

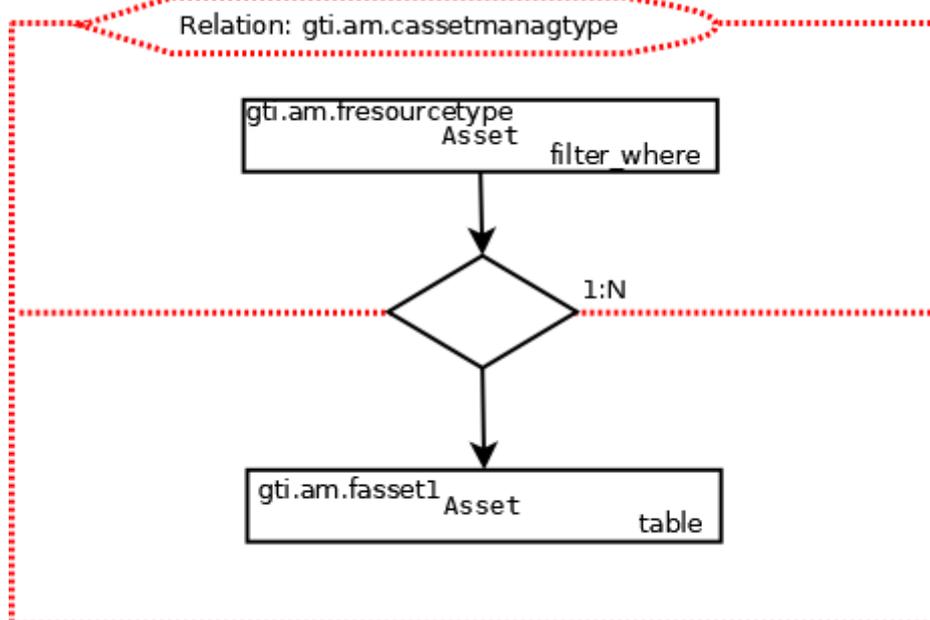


Ilustración 59: ERD anotado Gestión de inventario por tipo de recurso

3.5.1.2.6 Edición de activos – *gti.am.cassetedit1*

La interfaz *edición de activos* se encuentra implementada a través de la estructura *gti.am.cassetedit1*. A continuación, se muestra su diagrama ER anotado:



3 Diseño de los módulos de activos, adquisiciones y común de GTI

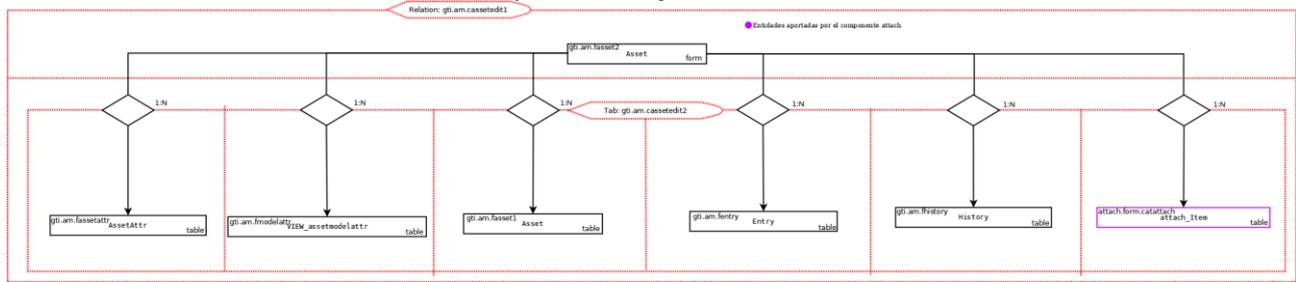


Ilustración 60: ERD anotado Edición de activos

3.5.1.2.7 Gestión de activos – gti.am.cassetmanag1

La interfaz *gestión de activos* se encuentra implementada a través de la estructura *gti.am.cassetmanag1*. A continuación, se muestra su diagrama ER anotado:

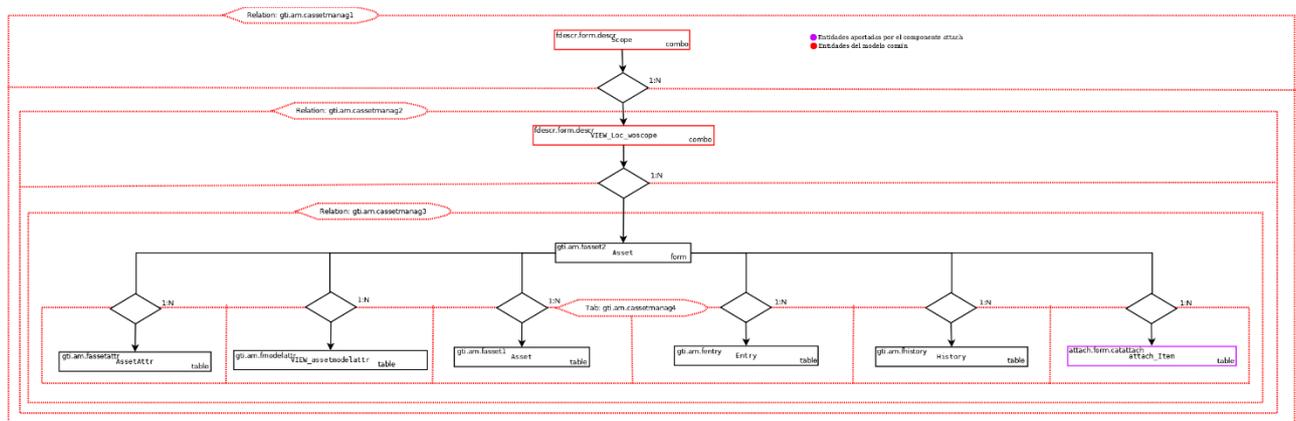


Ilustración 61: ERD anotado Gestión de activos

3.5.1.2.8 Estadísticas de tipos de activos por sede - gti.am.cgraphtype

La interfaz *estadísticas de tipos de activos por sede* se encuentra implementada a través de la estructura *gti.am.cgraphtype*. A continuación, se muestra su diagrama ER anotado:

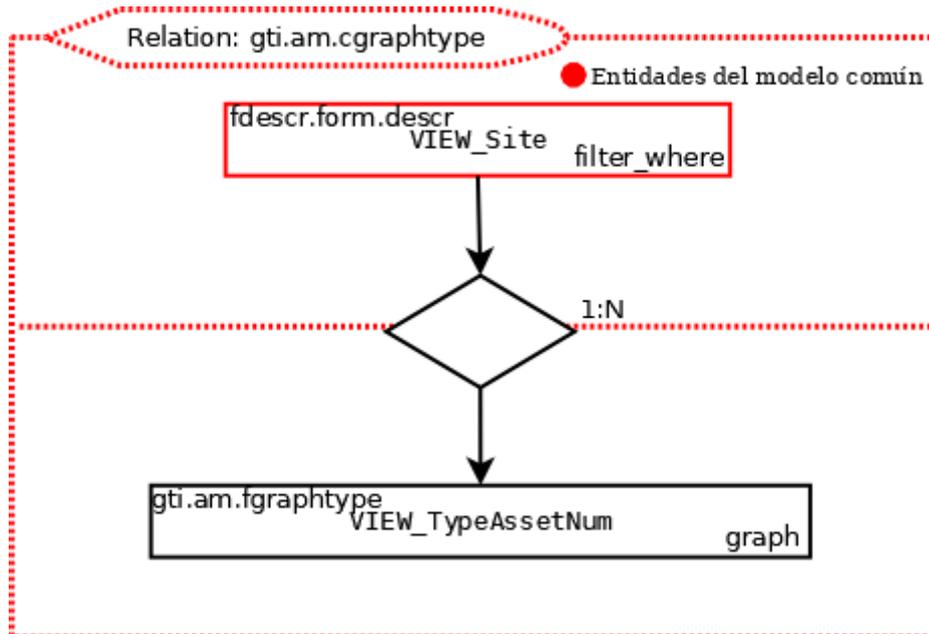


Ilustración 62: ERD anotado Estadísticas de tipos de activos por sede

3.5.1.2.9 Estadísticas de estados y marcas de activos por sede y tipo - *gti.am.cgraphstatusmark1*

La interfaz *estadísticas de estados y marcas de activos por sede y tipo* se encuentra implementada a través de la estructura *gti.am.cgraphstatusmark1*. A continuación, se muestra su diagrama ER anotado:

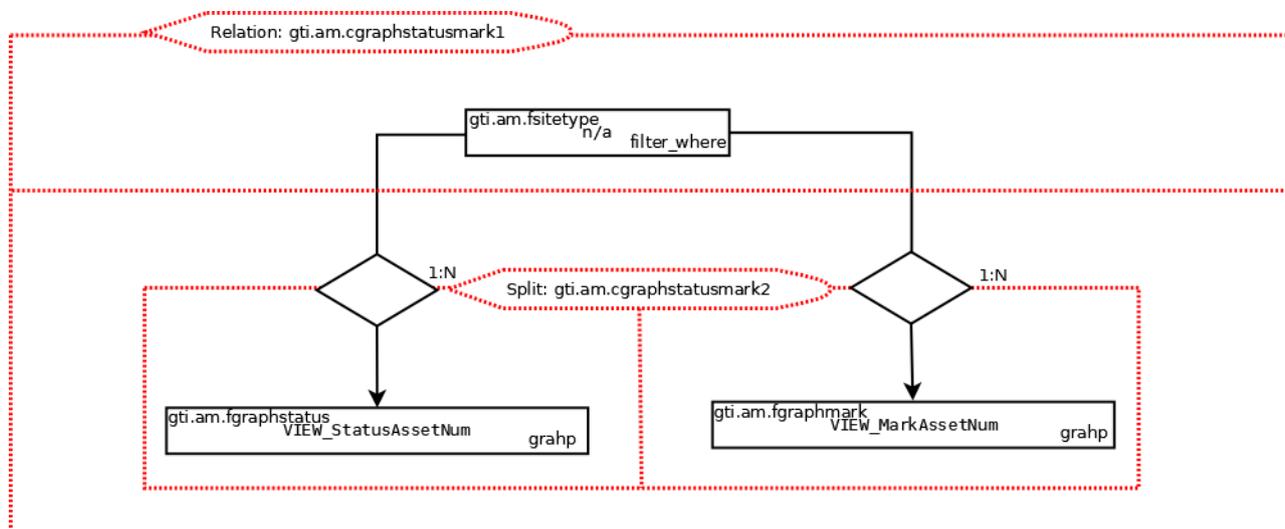


Ilustración 63: ERD anotado Estadísticas de estados y marcas de activos por sede y tipo

3.5.1.2.10 Consulta de recursos asignados – *cassignresourcequery1*

La interfaz *consulta de recursos asignados* se encuentra implementada a través de la estructura *cassignresourcequery1*. Su diagrama ER anotado se encuentra en el apartado 5.1.7.

3.5.1.3 Técnico avanzado

El técnico avanzando puede realizar las siguientes tareas:

- Recepcionar material por adquisiciones, movimientos o devoluciones.
- Realizar salidas de los recursos en las sedes que gestiona.
- Puede modificar información sobre los recursos en las sedes en las que está autorizado.
- Generar informes sobre el equipamiento de usuarios determinados.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.1.3.1 Gestión de entradas – *gti.am.cinputmanagete1*

La interfaz *gestión de entradas* se encuentra implementada a través de la estructura *gti.am.cinputmanagete1*. Esta interfaz es igual que la del apartado 5.2.2, pero está filtrada para los permisos del técnico avanzado. A continuación, se muestra su diagrama ER anotado:



3 Diseño de los módulos de activos, adquisiciones y común de GTI

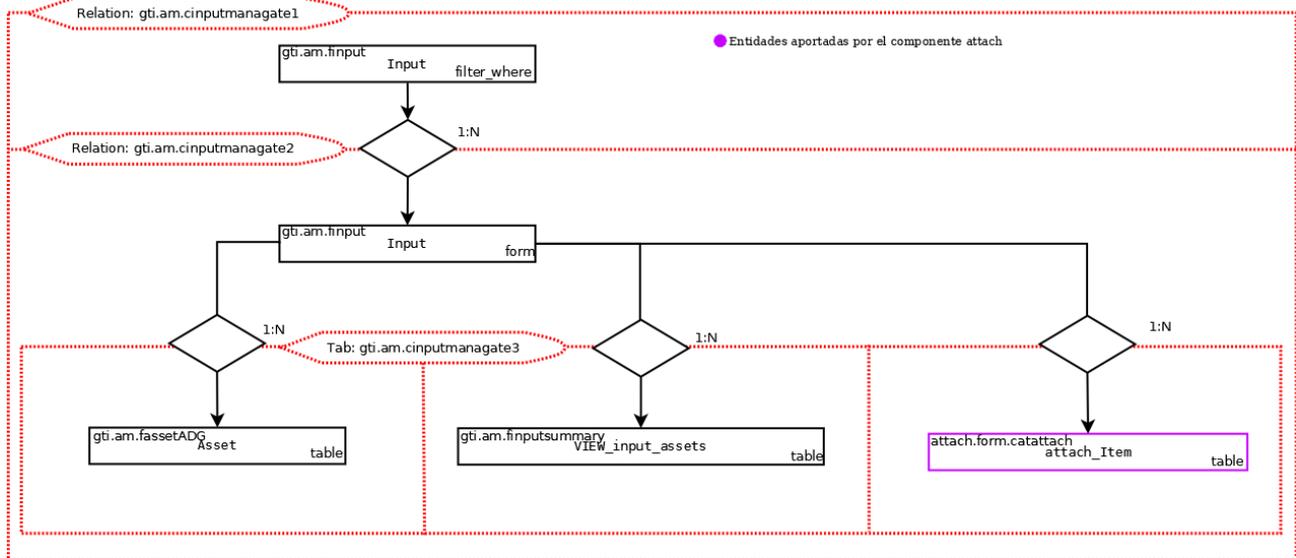


Ilustración 64: ERD anotado Gestión de entradas

3.5.1.3.2 Gestión de salidas – gti.am.coutputmanagate1

La interfaz *gestión de salidas* se encuentra implementada a través de la estructura *gti.am.coutputmanagate1*. Esta interfaz es igual que la del apartado 5.2.3, pero está filtrada para los permisos del técnico avanzado. A continuación, se muestra su diagrama ER anotado:

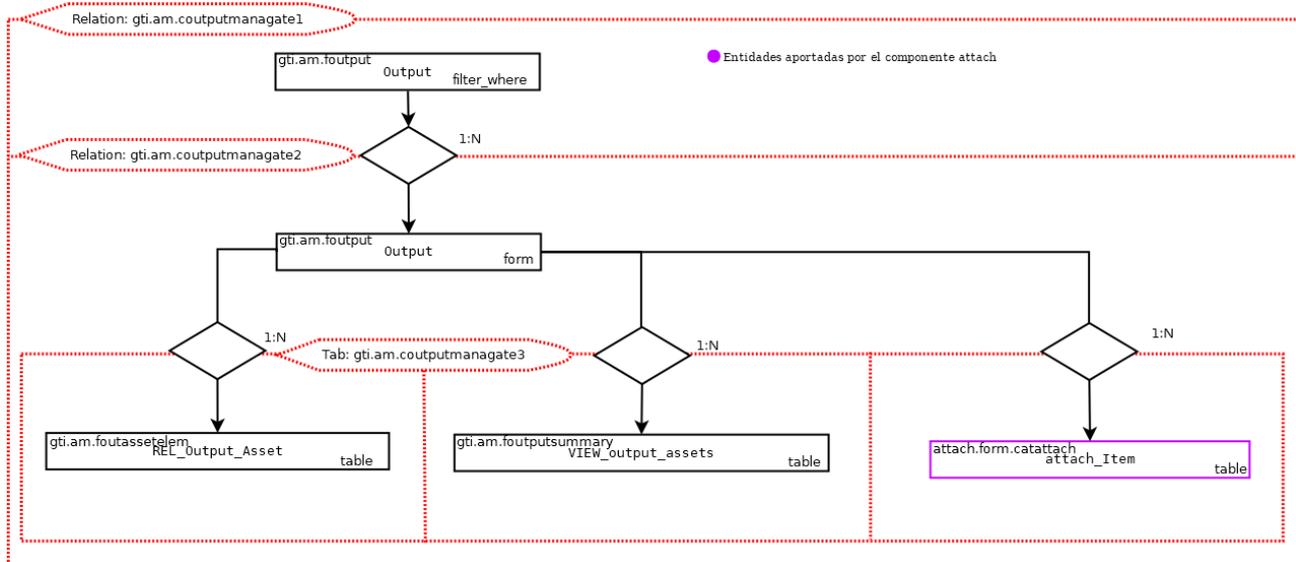


Ilustración 65: ERD anotado Gestión de salidas

3.5.1.3.3 Gestión de inventario reducida – gti.am.cassetmanagredate

La interfaz *gestión de inventario reducida* se encuentra implementada a través de la estructura *gti.am.cassetmanagredate*. A continuación, se muestra su diagrama ER anotado:

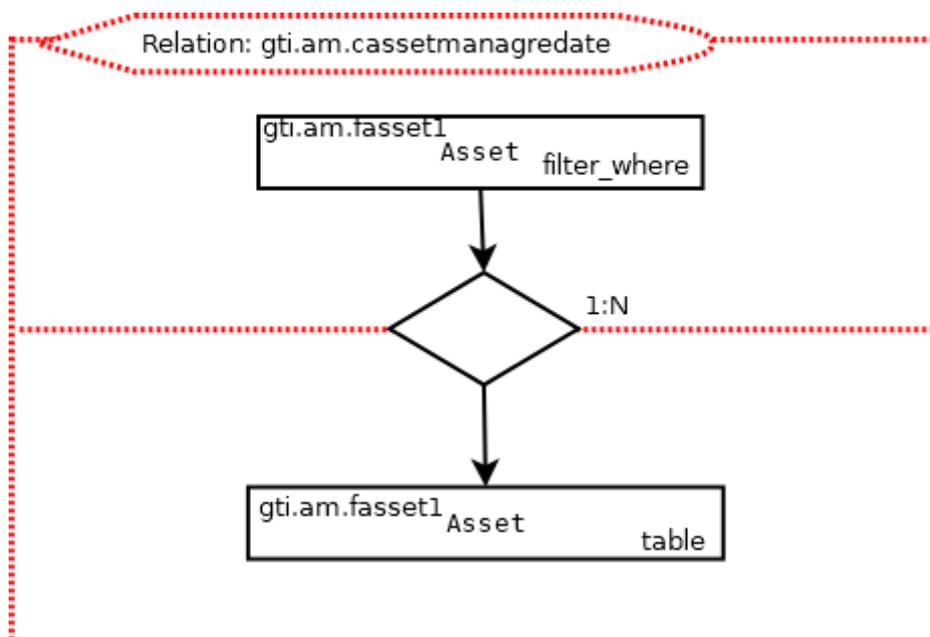


Ilustración 66: ERD anotado Gestión de inventario reducida

3.5.1.3.4 Edición de activos reducida – casseteditredate1

La interfaz *gestión de activos reducida* se encuentra implementada a través de la estructura *casseteditredate1*. A continuación, se muestra su diagrama ER anotado:

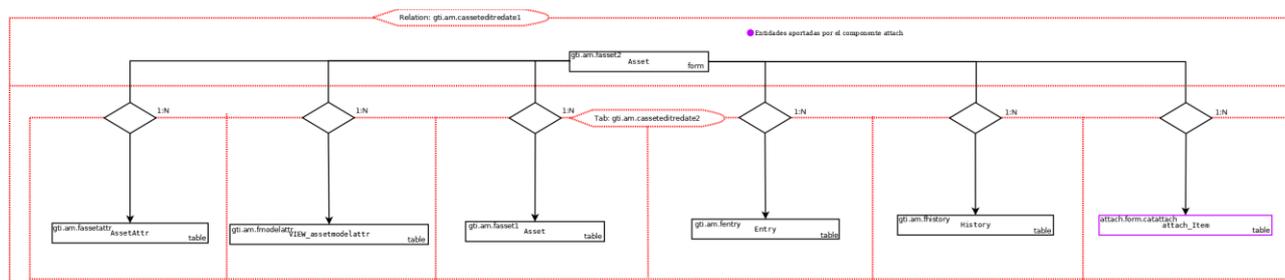


Ilustración 67: ERD anotado Edición de activos reducida

3.5.1.3.5 Consulta de recursos asignados – cassignresourcequery1

La interfaz *consulta de recursos asignados* se encuentra implementada a través de la estructura *cassignresourcequery1*. Su diagrama ER anotado se encuentra en el apartado 5.1.7.

3.5.1.4 Técnico

El técnico podrá realizar las mismas tareas que el técnico avanzado excepto por la gestión de entradas y salidas. Estas tareas son las siguientes:

- Puede modificar información sobre los recursos en las sedes en las que está autorizado.
- Generar informes sobre el equipamiento de usuarios determinados.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.1.4.1 Gestión de inventario reducida – gti.am.cassetmanagredtec

La interfaz *gestión de inventario reducida* se encuentra implementada a través de la estructura *gti.am.cassetmanagredtec*. Esta interfaz es igual que la del apartado 5.3.3, pero está filtrada para los permisos del técnico. A continuación, se muestra su diagrama ER anotado:

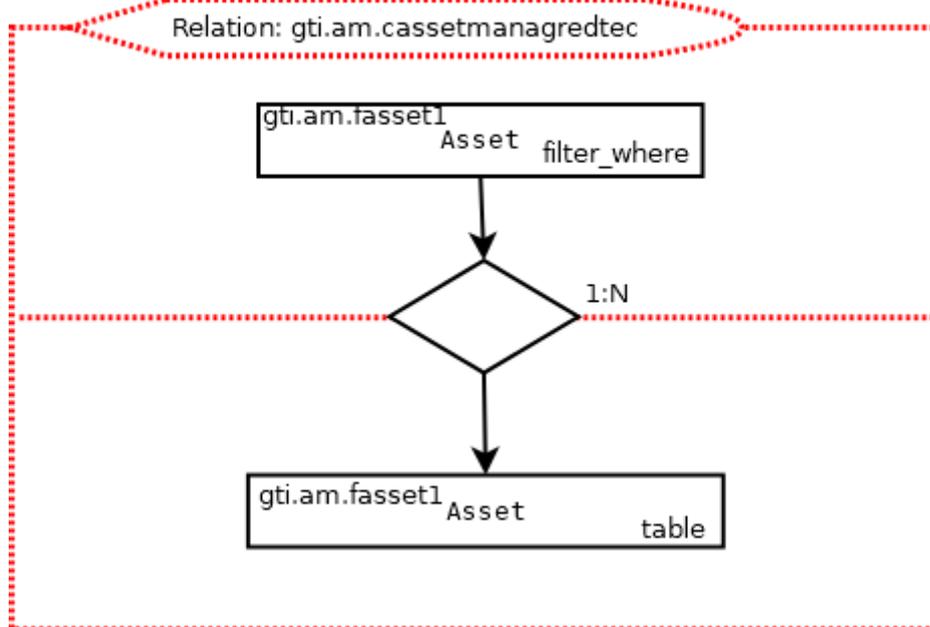


Ilustración 68: ERD anotado Gestión de inventario reducida

3.5.1.4.2 Edición de activos reducida – *gti.am. casseteditredtec1*

La interfaz *edición de activos reducida* se encuentra implementada a través de la estructura *gti.am. casseteditredtec1*. Esta interfaz es igual que la del apartado 5.3.4, pero está filtrada para los permisos del técnico. A continuación, se muestra su diagrama ER anotado:

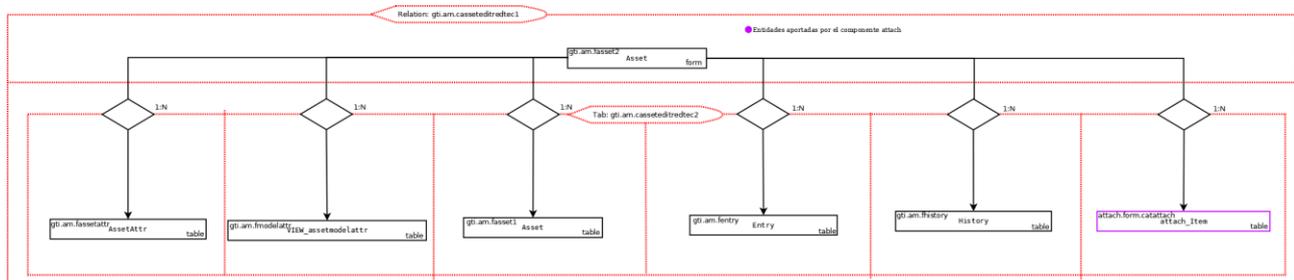


Ilustración 69: ERD anotado Edición de activos reducida

3.5.1.4.3 Consulta de recursos asignados – *gti.am.cassignresourcequery1*

La interfaz *consulta de recursos asignados* se encuentra implementada a través de la estructura *gti.am.cassignresourcequery1*. Su diagrama ER anotado se encuentra en el apartado 5.1.7.

3.5.1.5 Usuario regular

El usuario regular puede realizar la siguiente tarea:

- Consultar los recursos que tiene asignados.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.1.5.1 Consulta de recursos asignados –

gti.am.cassignresourcequery1

La interfaz *consulta de recursos asignados* se encuentra implementada a través de la estructura *gti.am.cassignresourcequery1*. Su diagrama ER anotado se encuentra en el apartado 5.1.7.

3.5.1.6 Usuario sin acceso

El usuario sin acceso puede realizar las siguientes tareas:

- Acceder a información básica sobre activos.
- Acceder a información básica sobre movimientos.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.1.6.1 Visualización de recurso – *gti.am.cresourceview*

La interfaz *visualización de recurso reducida* se encuentra implementada a través de la estructura *gti.am.cresourceview*. A continuación, se muestra su diagrama ER anotado:

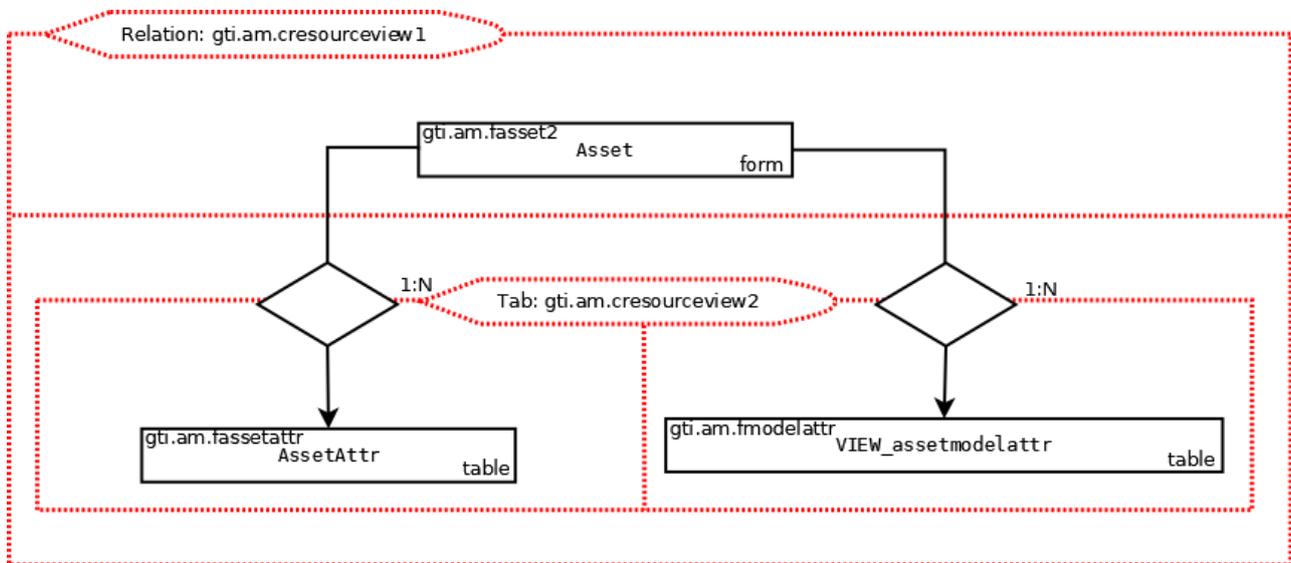


Ilustración 70: ERD anotado Visualización de recurso

3.5.1.6.2 Visualización de entrada – *gti.am.cinputview*

La interfaz *visualización de entrada* se encuentra implementada a través de la estructura *gti.am.cinputview*. A continuación, se muestra su diagrama ER anotado:

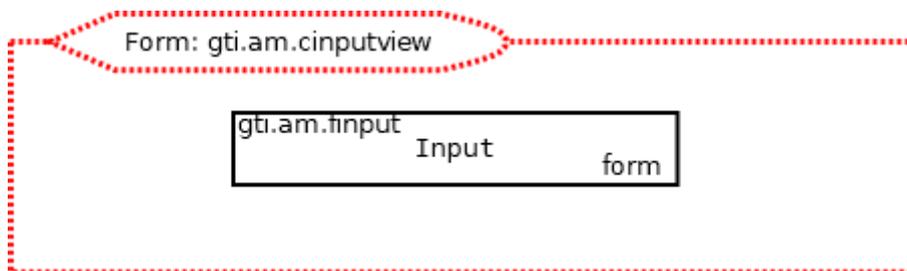


Ilustración 71: ERD anotado Visualización de entrada



3.5.1.6.3 Visualización de salida – *gti.am.coutputview*

La interfaz *visualización de salida* se encuentra implementada a través de la estructura *gti.am.coutputview*. A continuación, se muestra su diagrama ER anotado:

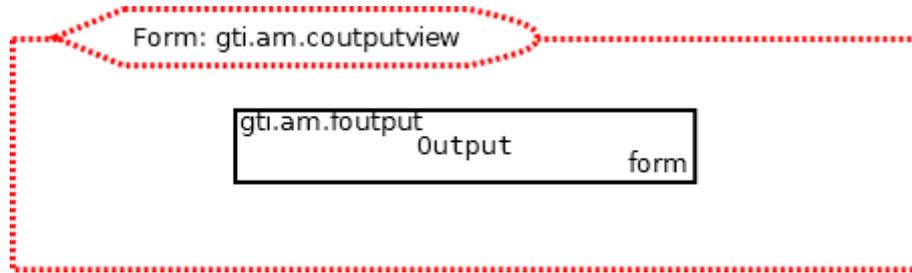


Ilustración 72: ERD anotado Visualización de salida

3.5.2 Diseño de las interfaces de usuario módulo gti

En este apartado, se van a detallar las unidades de interacción para cada rol del módulo común de GTI, junto al diseño de dichas unidades de interacción.

A continuación, se muestra la tabla que recoge qué unidades de interacción son accedidas para cada rol:

	Administrador del módulo común
Gestión de ámbitos	cscope
Gestión de unidades organizativas	cou11
Gestión de localizaciones	csite1
Datos personales	percon.struct.edit
Datos personales usuario	upercon.struct.edit

Tabla 3: Tabla Rol - Unidad de interacción

3.5.2.1 Administrador del módulo común

El administrador del módulo común puede realizar las siguientes tareas:

- Crear, añadir, modificar y añadir comentarios a ámbitos.
- Gestionar todos los niveles de localizaciones.
- Gestionar todos los niveles de unidades organizativas.
- Crear, modificar y añadir perfiles para personas.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.2.1.1 Gestión de ámbitos – *gti.cscope*

La interfaz *gestión de ámbitos* se encuentra implementada a través de la estructura *gti.cscope*. A continuación, se muestra su diagrama ER anotado:

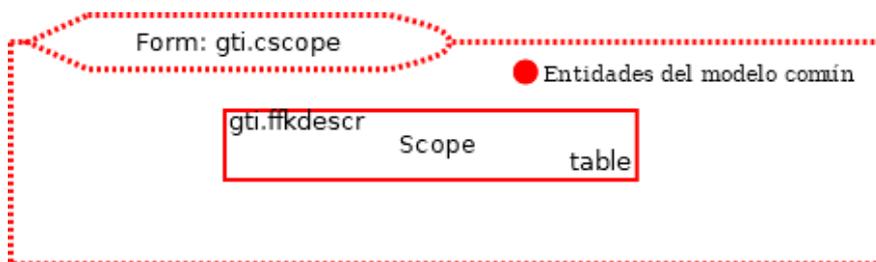


Ilustración 73: ERD anotado Gestión de ámbitos

3.5.2.1.2 Gestión de unidades organizativas – *gti.coul1*

La interfaz *gestión de unidades organizativas* se encuentra implementada a través de la estructura *gti.coul1*. A continuación, se muestra su diagrama ER anotado:

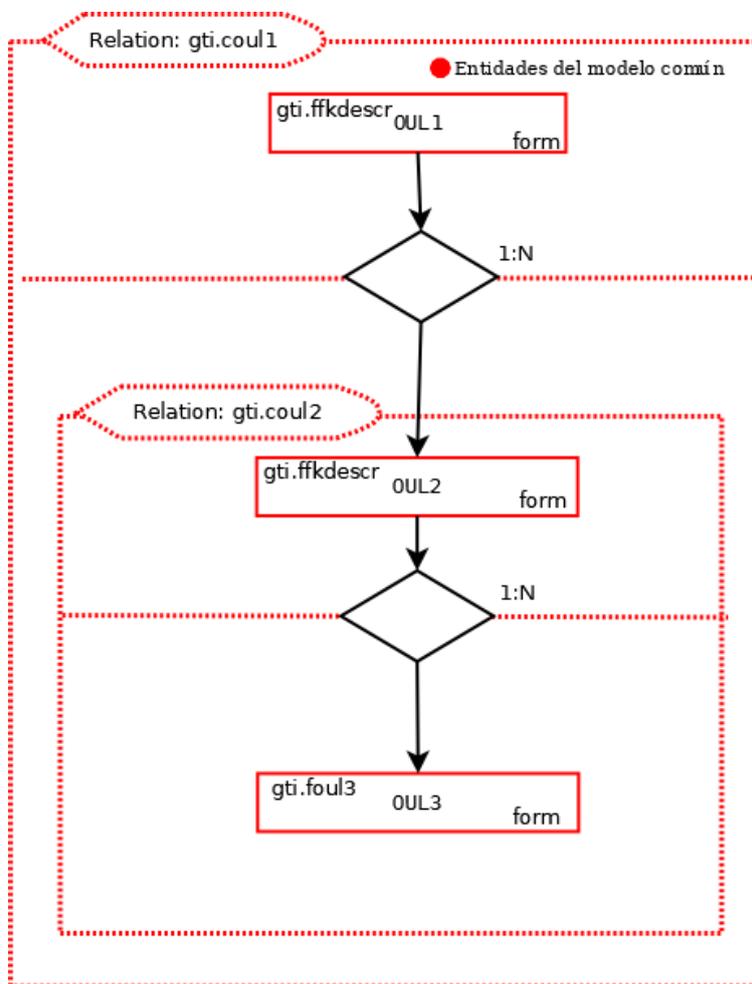


Ilustración 74: ERD anotado Gestión de unidades organizativas

3.5.2.1.3 Gestión de localizaciones – *gti.csite1*

La interfaz *gestión de localizaciones* se encuentra implementada a través de la estructura *gti.csite1*. A continuación, se muestra su diagrama ER anotado:

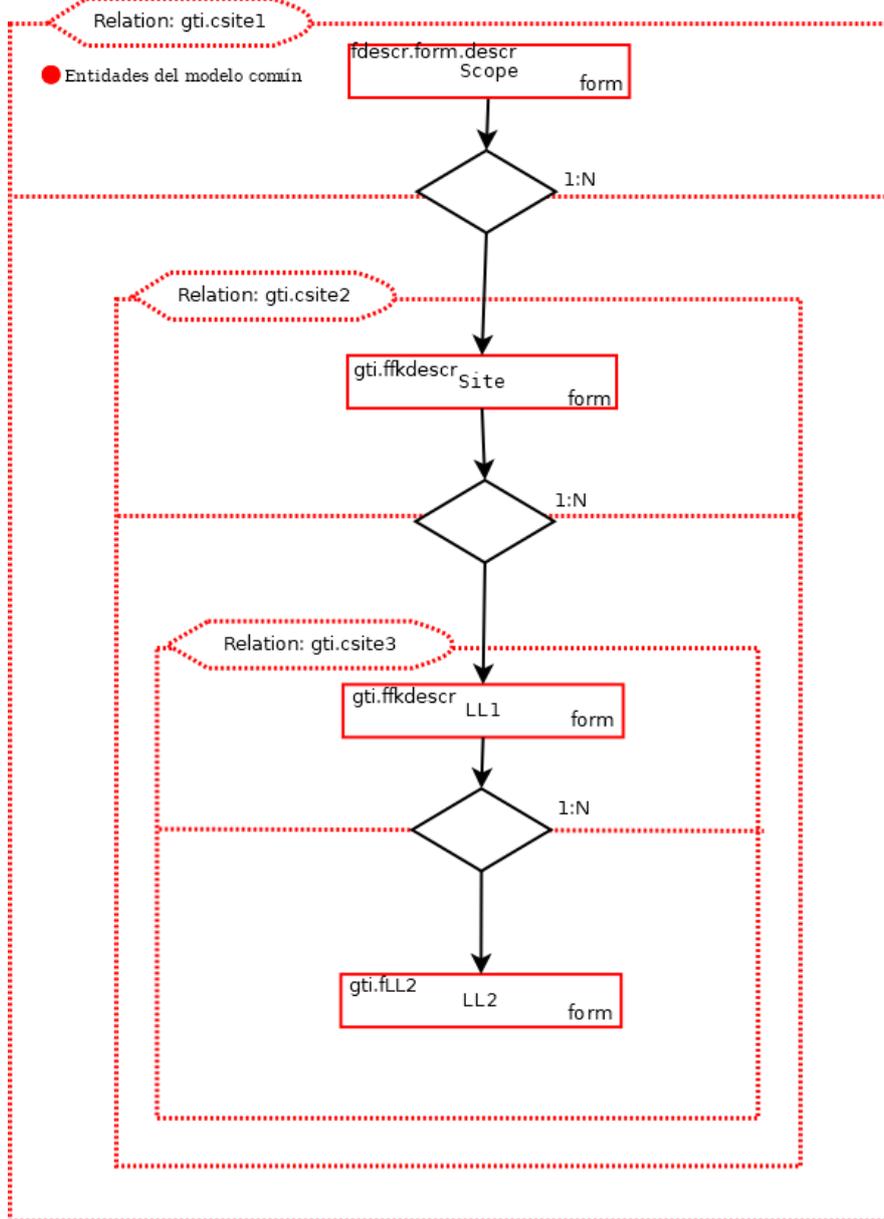


Ilustración 75: ERD anotado Gestión de localizaciones

3.5.2.1.4 Datos personales – *percon.struct.edit*

La interfaz *datos personales* se encuentra implementada a través de la estructura *percon.struct.edit*. Esta estructura se encuentra definida en paquete **percon**. A continuación, se muestra su diagrama ER anotado:

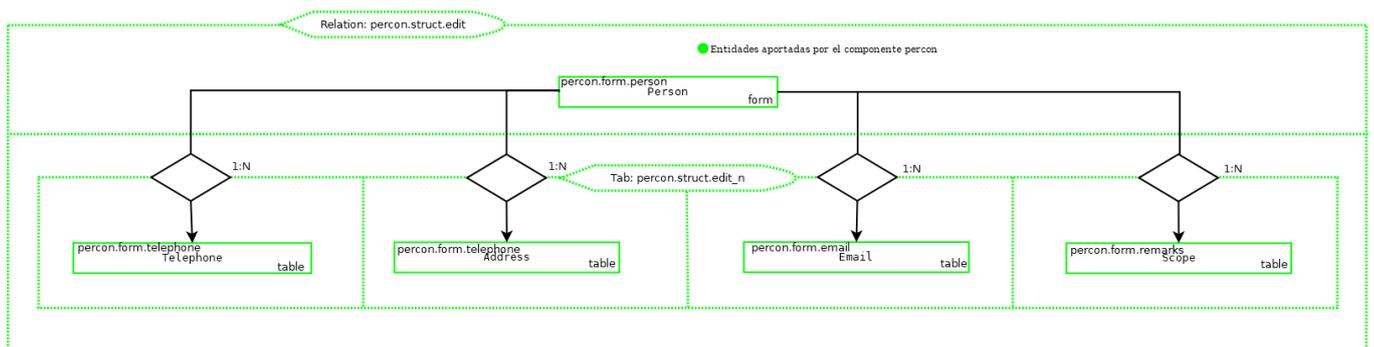


Ilustración 76: ERD anotado Datos personales

3.5.2.1.5 Datos personales usuario – *upercon.struct.edit*

La interfaz *datos personales* se encuentra implementada a través de la estructura *upercon.struct.edit*. Esta estructura se encuentra definida en paquete **percon**. A continuación, se muestra su diagrama ER anotado:

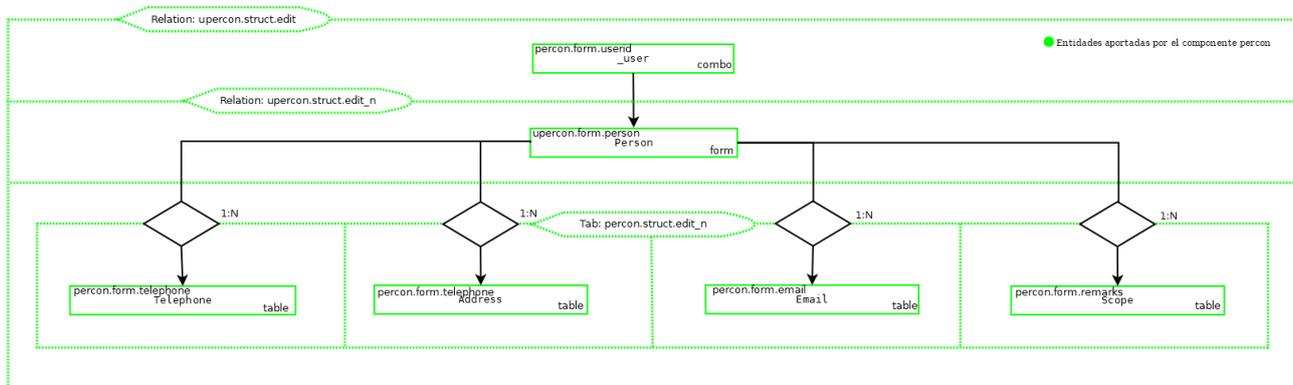


Ilustración 77: ERD anotado Datos personales usuario

3.5.3 Diseño de las interfaces de usuario módulo pp

En este apartado, se van a detallar las unidades de interacción para cada rol del módulo de adquisiciones de GTI, junto al diseño de dichas unidades de interacción.

A continuación, se muestra la tabla que recoge qué unidades de interacción son accedidas para cada rol:

	Gestor de adquisiciones	Usuario sin acceso
Gestión de proveedores	cprovmanag1	
Gestión de ofertas	coffermanag1	
Gestión de adquisiciones	cpurchmanag1	
Marcas, modelos y atributos	cmarkmodell	
Consulta de recursos asignados	cassignresourcequery1	
Visualización adquisición		cpurchaseview

Tabla 4: Tabla Rol - Unidad de interacción

3.5.3.1 Gestor de adquisiciones

El gestor de adquisiciones puede realizar las siguientes tareas:

- Gestionar marcas y modelos con sus atributos.



3 Diseño de los módulos de activos, adquisiciones y común de GTI

- Gestionar de forma completa los recursos de las sedes en las que está autorizado.
- Recepcionar material por adquisiciones, movimientos o devoluciones.
- Realizar salidas de los recursos en las sedes que gestiona.
- Extraer relaciones de equipamiento base a sus propiedades.
- Extraer estadísticas relativas al equipamiento.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.3.1.1 Gestión de proveedores – gti.am.cprovmanag1

La interfaz *gestión de proveedores* se encuentra implementada a través de la estructura *gti.am.cprovmanag1*. A continuación, se muestra su diagrama ER anotado:

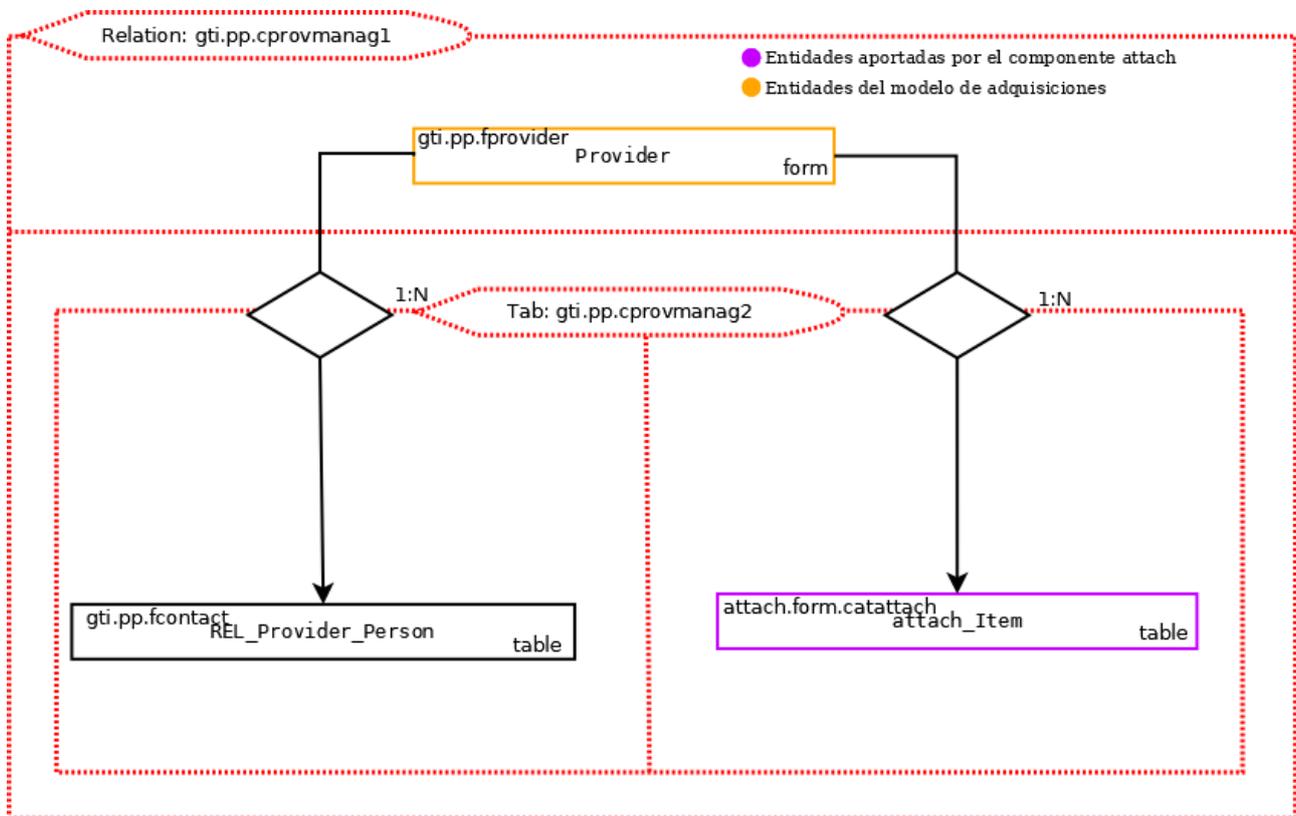


Ilustración 78: ERD anotado Gestión de proveedores

3.5.3.1.2 Gestión de ofertas – gti.am.coffermanag1

La interfaz *gestión de ofertas* se encuentra implementada a través de la estructura *gti.am.coffermanag1*. A continuación, se muestra su diagrama ER anotado:

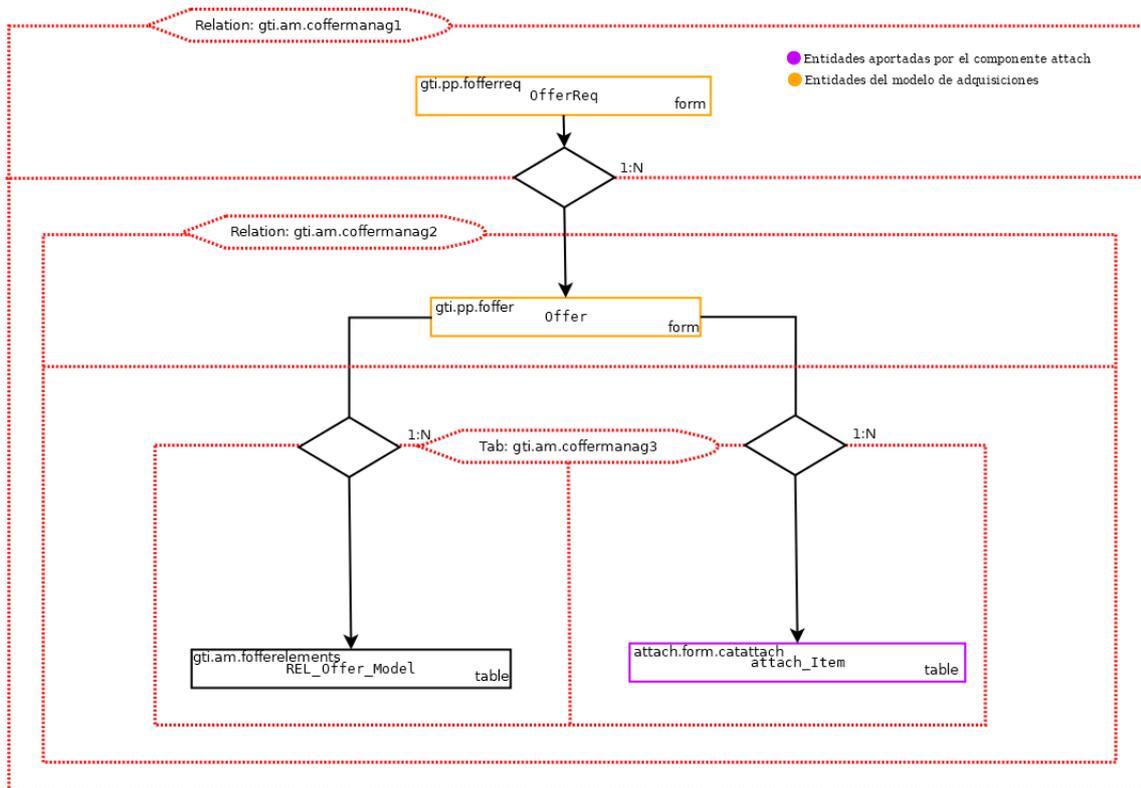


Ilustración 79: ERD anotado Gestión de ofertas

3.5.3.1.3 Gestión de adquisiciones – gti.am.cpurchase

La interfaz *gestión de adquisiciones* se encuentra implementada a través de la estructura *gti.am.cpurchase*. A continuación, se muestra su diagrama ER anotado:

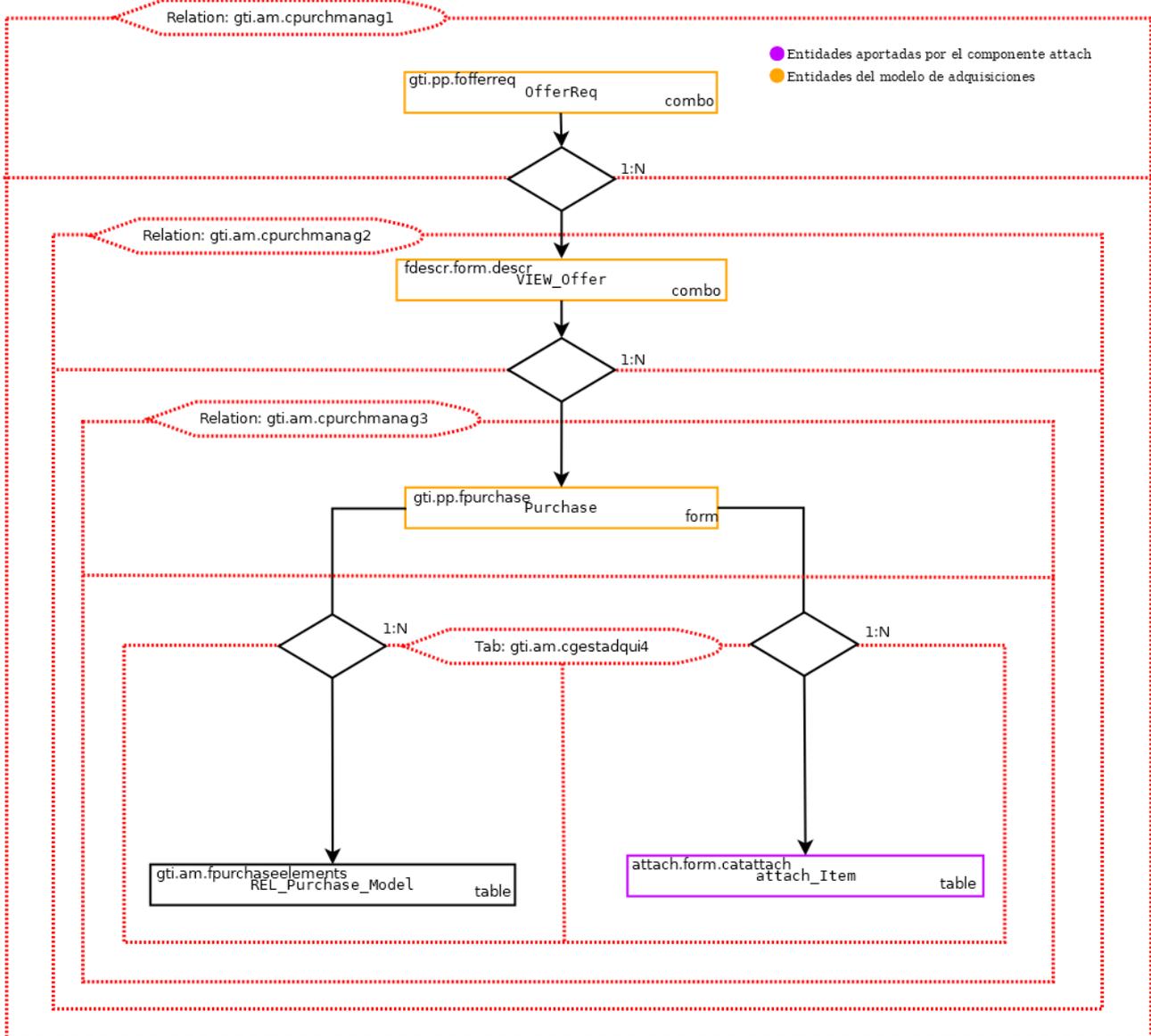


Ilustración 80: ERD anotado Gestión de adquisiciones

3.5.3.1.4 Marcas, modelos y atributos – gti.am.cmarkmodell

La interfaz *marcas, modelos y atributos* se encuentra implementada a través de la estructura *gti.am.cmarkmodell*. A continuación, se muestra su diagrama ER anotado:

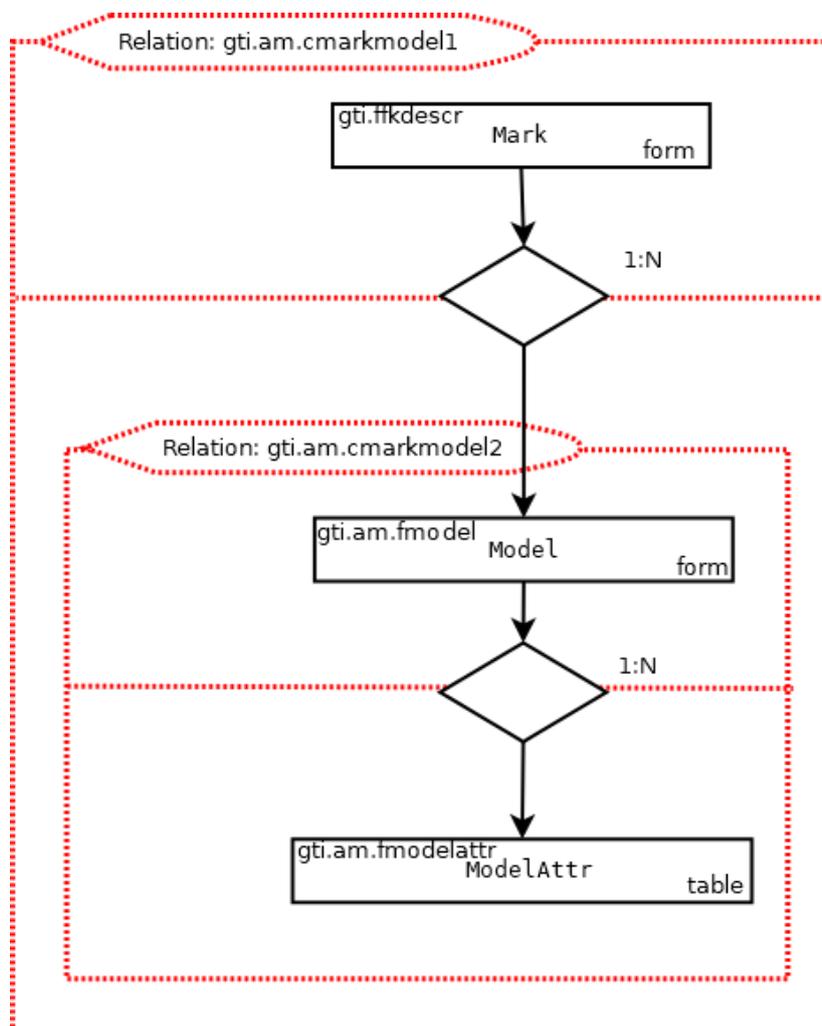


Ilustración 81: ERD anotado Marcas, modelos y atributos

3.5.3.1.5 Consulta de recursos asignados – *gti.am.cassignresourcequery1*

La interfaz *consulta de recursos asignados* se encuentra implementada a través de la estructura *gti.am.cassignresourcequery1*. Su diagrama ER anotado se encuentra en el apartado 5.2.7 del Documento de Diseño del módulo de inventario.

3.5.3.2 Usuario sin acceso

El usuario sin acceso puede realizar las siguientes tareas:

- Acceder a información básica sobre activos.
- Acceder a información básica sobre movimientos.
- Acceder a información básica sobre adquisiciones.

A continuación, se van a detallar los diagramas Entidad-Relación de las interfaces utilizadas para lograr las tareas mencionadas.

3.5.3.2.1 Visualización de adquisición – *gti.pp.cpurchaseview*

La interfaz *visualización de adquisición* se encuentra implementada a través de la estructura *gti.pp.cpurchaseview*. A continuación, se muestra su diagrama ER anotado:



Ilustración 82: ERD anotado Visualización de adquisición

3.6 Casos de uso reales

3.6.1 Casos de uso reales módulo am

A continuación, se van a detallar algunos casos de uso que reflejan las funcionalidades del módulo de inventario de GTI. Para ello, se va a exponer un caso de uso para cada uno de los distintos roles.

3.6.1.1 Asignación de permisos a un usuario

El administrador del módulo de inventario asigna permisos a los gestores de adquisiciones, de inventario, técnicos y técnicos avanzados según el usuario. El diagrama de caso de uso es el siguiente:

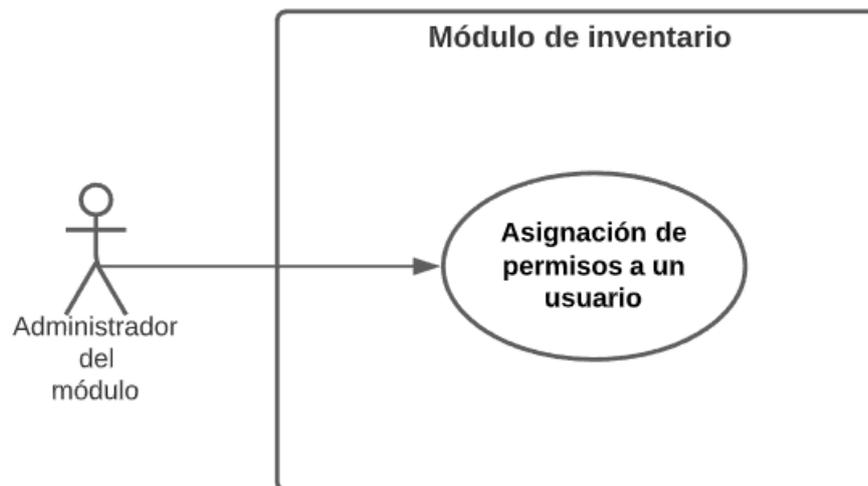


Ilustración 83: Diagrama de caso de uso CU-01: asignación de permisos a un usuario

CU-01	Asignación de permisos a un usuario	
Dependencias	<ul style="list-style-type: none"> R1.1.3 Los gestores y técnicos se definen por ámbito. R2.1.1 Establece gestores de adquisiciones por ámbitos. R2.1.2 Establece gestores de inventario y técnicos por sede. 	
Precondición	El usuario al que se le van a otorgar los permisos debe encontrarse registrado en el módulo.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando <i>el administrador del módulo asigna permisos a un determinado usuario.</i>	
Secuencia normal	Paso	Acción

	1	El administrador inicia sesión en la aplicación.
	2	El administrador selecciona en la sección <i>Gestión módulo inventario</i> la opción <i>Administración de permisos</i> .
	3	El administrador selecciona en el navegador el ámbito sobre el que quiere asignar los permisos.
	4	El administrador del módulo elegirá entre las pestañas Gest. Adquisiciones, Gest. Inventario, Tec. Avanzados y Técnicos, la que quiera utilizar.
	5	El administrador busca el usuario que quiere añadir al grupo en el desplegable de la tabla que se muestra y pulsa <i>Añadir</i> .
	6	El usuario queda registrado en la tabla junto a los otros usuarios que pudiera haber en el grupo.
Postcondición	El usuario queda registrado como nuevo miembro del grupo pasando a tener los permisos que tienen asignados los usuarios de ese grupo.	
Excepciones	Paso	Acción
	5	Si el usuario ya es miembro del grupo,
	E.1	El sistema informa de una situación de error mostrando el mensaje: <i>Inserción abortada. Ese usuario ya existe en este ámbito.</i>
E.2	Se cancela el caso de uso.	
Comentarios	Un usuario puede ser miembro de más de un grupo de permisos.	

Tabla 5: Caso de uso CU-01: asignación de permisos a un usuario

3.6.1.2 Gestión de los activos

El gestor de inventario modifica las características de un activo determinado. Para ello, tendrá que hacer una búsqueda del activo deseado y acceder a la interfaz de edición de los activos. El diagrama de caso de uso es el siguiente:

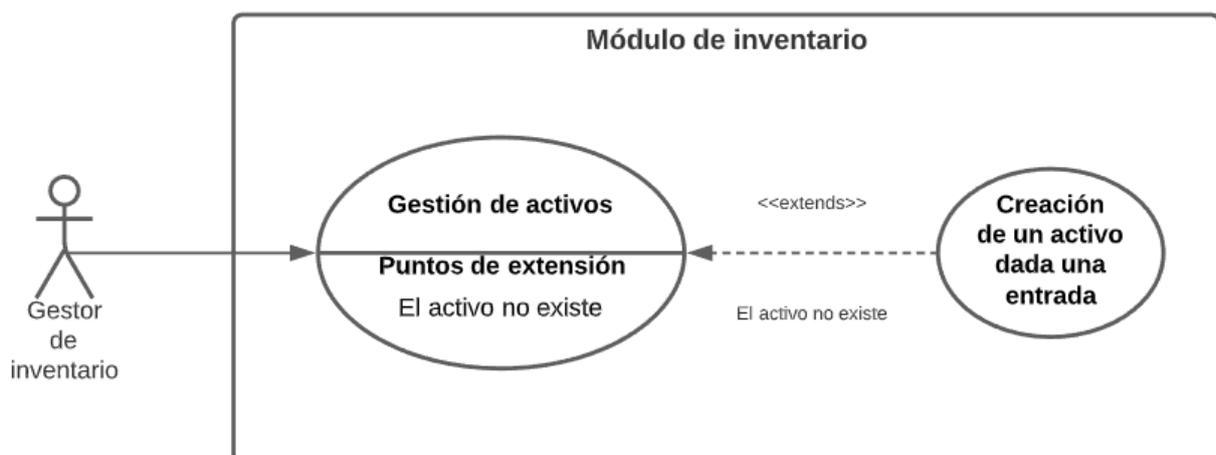


Ilustración 84: Diagrama de caso de uso CU-02: gestión de los activos

CU-02	Gestión de los activos
--------------	-------------------------------



3 Diseño de los módulos de activos, adquisiciones y común de GTI

<p>Dependencias</p>	<ul style="list-style-type: none"> • R1.4.1 Siempre están asignados a una ubicación. • R1.4.2 Están asignados a un usuario. Un activo siempre tendrá asignado a alguien responsable del mismo. • R1.4.3 Los activos son de un tipo. La pertenencia a un tipo determina unos campos personalizados. • R1.4.4 Los activos son de un modelo determinado. • R1.4.7 Tienen fecha de alta y fecha de baja en el sistema. • R1.4.8 Se debe poder trazar los cambios durante la vida del activo. • R1.4.9 Se deben poder realizar anotaciones sobre el activo. • R1.4.10 Se deben poder adjuntar documentos a un activo. • R1.4.11 Debe permitirse generar relaciones de activos filtrando por sus características. • R1.4.12 Debe permitirse generar relaciones de activos filtrando por sus campos personalizados. • R2.2.2 Gestiona de forma completa los recursos de las sedes en las que está autorizado. • R2.2.5 Extrae relaciones de equipamiento en base a sus propiedades. • R2.2.6 Extrae estadísticas relativas al equipamiento. 	
<p>Precondición</p>	<p>El gestor de inventario debe tener permisos sobre el ámbito del recurso a modificar.</p>	
<p>Descripción</p>	<p>El sistema deberá comportarse como se describe en el siguiente caso de uso cuando <i>el gestor de inventario modifica las características de un activo</i>.</p>	
<p>Secuencia normal</p>	<p>Paso</p>	<p>Acción</p>
	<p>1</p>	<p>El gestor de inventario inicia sesión en la aplicación.</p>
	<p>2</p>	<p>El gestor de inventario selecciona en la sección <i>Gestión inventario</i> la opción <i>Gestión de activos</i>.</p>
	<p>3</p>	<p>El gestor de inventario elige en el combo <i>Ámbito y Localización</i>, los valores correspondientes al recurso a modificar.</p>
	<p>4</p>	<p>El gestor de inventario elige en el navegador el activo a modificar según el identificador y el número de serie.</p>
	<p>5</p>	<p>El gestor de inventario modifica los valores deseados y pulsa <i>Modificar</i>.</p>
<p>Postcondición</p>	<p>El sistema actualiza los valores que han sido modificado y crea un registro en la pestaña <i>Traza</i> para el activo actualizado. El gestor de inventario podrá visualizar las propiedades, recursos hijos y traza del activo seleccionado, así como añadir anotaciones y adjuntos sobre dicho activo.</p>	
<p>Excepciones</p>	<p>Paso</p>	<p>Acción</p>
	<p>4</p>	<p>Si el activo a modificar no existe,</p>
	<p>E.1</p>	<p>Se realiza el caso de uso <i>Creación de un activo dada una entrada</i>.</p>

	E.2	El usuario permitido continúa la secuencia normal del caso de uso.
4	Si no existe ningún activo para dicha ubicación,	
	E.1	En el navegador no aparecerá ningún activo.
	E.2	Se cancela el caso de uso.

Tabla 6: Caso de uso CU-02: gestión de los activos

3.6.1.3 Tramitación de una entrada de activos para una adquisición

El gestor de inventario o técnico avanzado reciben una entrada de activos para una determinada adquisición. Con ello, se copiarán todos los elementos que se incluyan en dicha adquisición a la entrada creada y dando de alta a los nuevos activos. El diagrama de caso de uso es el siguiente:

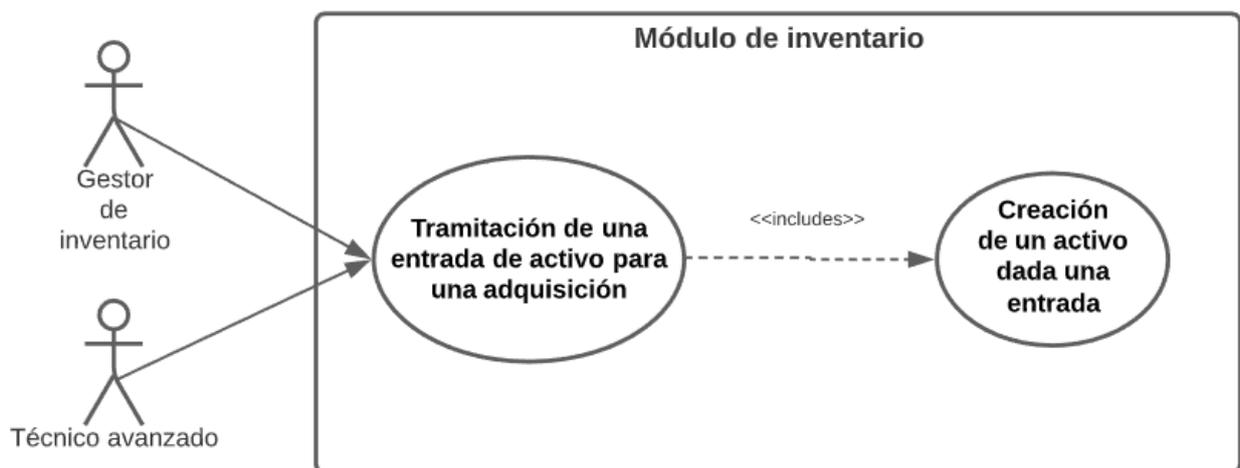


Ilustración 85: Diagrama de caso de uso CU-03: tramitación de una entrada de activos para una adquisición

CU-03	Tramitación de una entrada de activos para una adquisición
Dependencias	<ul style="list-style-type: none"> • R1.3.1 El sistema debe contemplar entradas y salidas de activos. • R1.3.2 Los movimientos tendrán un origen y un destino. • R1.3.3 Tendrán también asignados responsables. • R1.3.5 Tipos de entradas predefinidas: Adquisición, Interno, devolución. • R1.3.8 Se debe poder generar la documentación asociada al movimiento concreto. • R1.3.9 Se debe poder realizar el alta o baja de los activos relacionados en el movimiento en función del tipo. • R1.3.10 El tipo de movimiento puede determinar la copia de datos de una adquisición. • R2.2.3 Recepciona material por adquisiciones, movimientos o



3 Diseño de los módulos de activos, adquisiciones y común de GTI

	devoluciones. <ul style="list-style-type: none"> • R2.3.1 Recepciona material por adquisiciones, movimientos o devoluciones. 	
Precondición	La adquisición que se desea elegir debe estar registrada en la aplicación con antelación.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando <i>el usuario permitido recepciona una entrada de activos para una determinada adquisición.</i>	
Secuencia normal	Paso	Acción
	1	El usuario permitido (gestor de inventario o técnico avanzado) inicia sesión en la aplicación.
	2	El usuario permitido selecciona en la sección <i>Gestión inventario o Gestión técnico avanzado</i> la opción <i>Gestión de entradas.</i>
	3	El usuario permitido rellena los campos necesarios en la tabla <i>Entrada</i> , eligiendo en el Tipo “Adquisición” y en la Adquisición, la adquisición que se desea y pulsa <i>Añadir.</i>
Postcondición	Los elementos incluidos en la adquisición serán creados como activos del sistema y deberán ser otorgados de un número de serie y un identificador. Se pueden añadir activos nuevos asociados a dicha entrada.	

Tabla 7: Caso de uso CU-03: tramitación de una entrada de activos para una adquisición

3.6.1.4 Consulta de los recursos asignados

Cualquier usuario con acceso a la aplicación del módulo de inventario de GTI consulta los recursos de los que son responsables. En esta interfaz, solo podrá visualizar los valores de los recursos, no modificar. El diagrama de caso de uso es el siguiente:

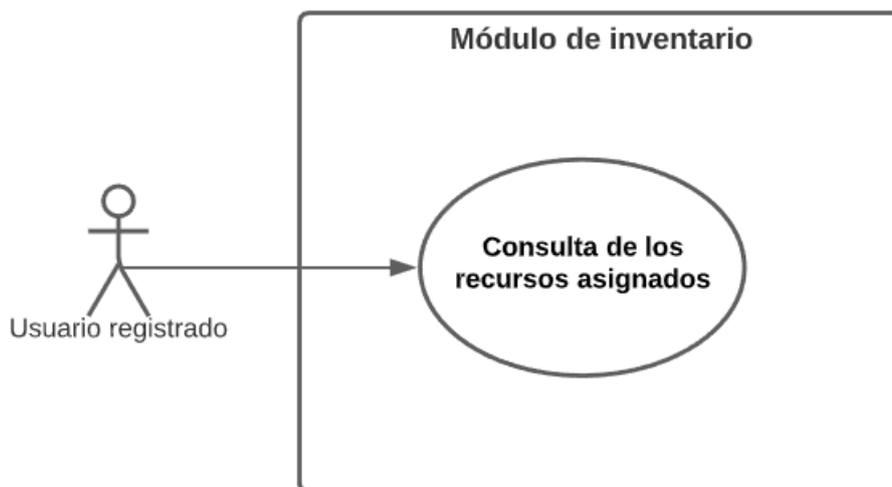


Ilustración 86: Diagrama de caso de uso CU-04: consulta de los recursos asignados

CU-04	Consulta de los recursos asignados
Dependencias	<ul style="list-style-type: none"> • R1.4.2 Están asignados a un usuario. Un activo siempre tendrá

	asignado a alguien responsable del mismo. <ul style="list-style-type: none"> • R2.5.1 Puede consultar los recursos que tiene asignados. 	
Precondición	El usuario debe tener al menos un recurso asignado.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando <i>el usuario permitido receptiona una entrada de activos para una determinada adquisición.</i>	
Secuencia normal	Paso	Acción
	1	El usuario con acceso inicia sesión en la aplicación.
	2	El usuario con acceso selecciona en el menú la opción <i>Consulta de recursos asignados.</i>
	3	El usuario elige, a través del navegador, el recurso a visualizar mediante el identificador y el número de serie.
Postcondición	El usuario con acceso puede además ver las propiedades y los recursos hijos del activo seleccionado en las pestañas <i>Propiedades y Recursos Hijos</i> , respectivamente.	
Excepciones	Paso	Acción
	3	Si el usuario no es responsable de ningún recurso,
	E.1	En el navegador no aparecerá ningún recurso.
	E.2	Se cancela el caso de uso.
Comentarios	En esta interfaz, solo se podrán visualizar los valores característicos del recurso.	

Tabla 8: Caso de uso CU-04: consulta de los recursos asignados

3.6.1.5 Visualización del recurso

Un usuario sin acceso a la aplicación del módulo de inventario de GTI consulta un recurso del que tiene enlace. En esta interfaz, solo podrá visualizar los valores de los recursos, no modificar. El diagrama de caso de uso es el siguiente:

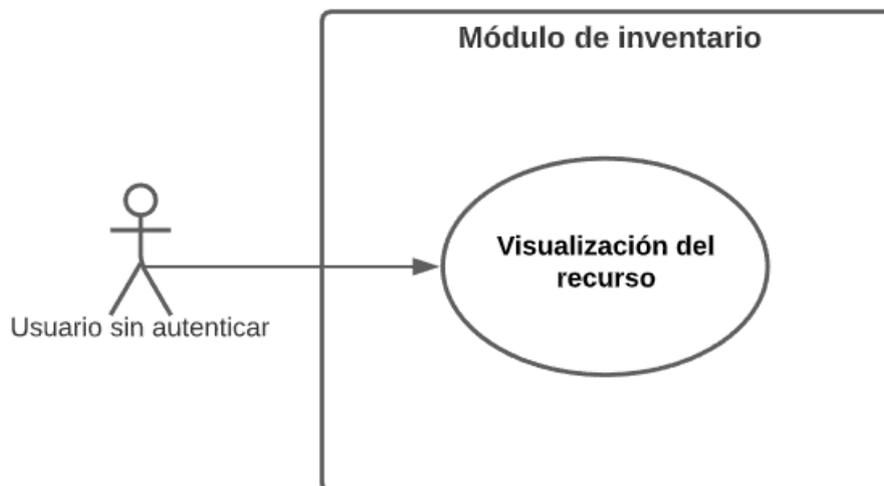


Ilustración 87: Diagrama de caso de uso CU-05: visualización del recurso

CU-05	Visualización del recurso
--------------	----------------------------------



3 Diseño de los módulos de activos, adquisiciones y común de GTI

Dependencias	<ul style="list-style-type: none"> R2.6.1 Puede acceder a información básica sobre activos. 	
Precondición	El usuario deberá acceder a esta interfaz mediante un enlace del activo.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando <i>el usuario sin autenticar visualiza la información básica de un recurso</i> .	
Secuencia normal	Paso	Acción
	1	El usuario accede a la interfaz mediante el enlace del activo.
Postcondición	El usuario puede además ver las propiedades del activo en la pestaña <i>Propiedades</i> .	
Comentarios	En esta interfaz, solo se podrán visualizar los valores característicos del recurso.	

Tabla 9: Caso de uso CU-05: visualización del recurso

3.6.2 Casos de uso reales módulo pp

A continuación, se van a detallar algunos casos de uso que reflejan las funcionalidades del módulo de adquisición de GTI. Para ello, se va a exponer un caso de uso para cada uno de los distintos roles.

3.6.2.1 Tramitación de una oferta dada una solicitud

El gestor de adquisiciones efectúa una oferta dada una solicitud de oferta. Para ello, recibe ofertas a través de los proveedores y determina los elementos que las componen. El diagrama de caso de uso es el siguiente:

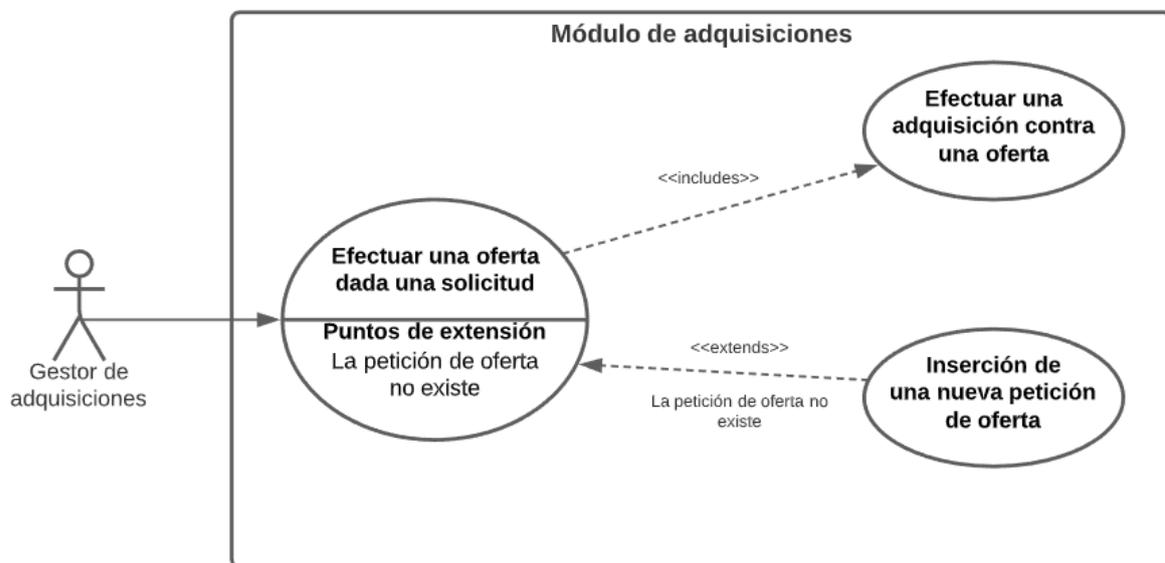


Ilustración 88: Diagrama de caso de uso CU-01: tramitación de una oferta dada una solicitud

CU-01	Tramitación una oferta dada una solicitud
Dependencias	<ul style="list-style-type: none"> R1.1.1 Las adquisiciones se inician con una petición de oferta R1.1.2 A las peticiones de ofertas pueden responder varios

	<p>proveedores.</p> <ul style="list-style-type: none"> • R1.1.3 Se debe poder detallar el contenido de una oferta, precio unitario, cantidad, etc. • R1.1.4 Se debe poder adjuntar documentación relativa a la oferta. • R2.1.3 Solicita ofertas a los proveedores. • R2.1.4 Recibe ofertas a sus solicitudes. • R2.1.5 Realiza adquisiciones contra una oferta de un proveedor. 	
Precondición	El gestor de adquisiciones que efectúa la oferta debe tener permisos en el ámbito de la petición de oferta a trabajar y en el grupo <i>Gest. Adquisiciones</i> .	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando <i>el gestor de adquisiciones efectúa una oferta para una determinada petición de oferta</i> .	
Secuencia normal	Paso	Acción
	1	El gestor de adquisiciones inicia sesión en la aplicación.
	2	El gestor de adquisiciones selecciona en la sección <i>Gestión adquisiciones</i> la opción <i>Gestión de ofertas</i> .
	3	El gestor de adquisiciones selecciona en el navegador la petición de oferta sobre la que quiere efectuar la oferta.
	4	El gestor de adquisiciones rellena los campos necesarios y pulsa <i>Añadir</i> .
	5	El gestor de adquisiciones añade los elementos necesarios en la pestaña <i>Elementos</i> , rellenando los campos necesarios y pulsa <i>Añadir</i> .
Postcondición	El gestor de adquisiciones podrá realizar adquisiciones contra una oferta de un proveedor siempre que tenga los permisos en dicho grupo.	
Excepciones	Paso	Acción
	3	Si la petición de oferta no existe,
	E.2.1	Se realiza el caso de uso <i>Inserción de una nueva petición de oferta</i> .
	E.2.2	El gestor de adquisiciones continúa la secuencia normal del caso de uso.
Comentarios	Se pueden realizar más de una oferta dada una misma petición de oferta.	

Tabla 10: Caso de uso CU-01: tramitación de una oferta dada una solicitud

3.6.2.2 Visualización de adquisición

Un usuario sin acceso a la aplicación del módulo de adquisición de GTI consulta una adquisición del que tiene enlace. En esta interfaz, solo podrá visualizar los valores de la adquisición, no modificar. El diagrama de caso de uso es el siguiente:

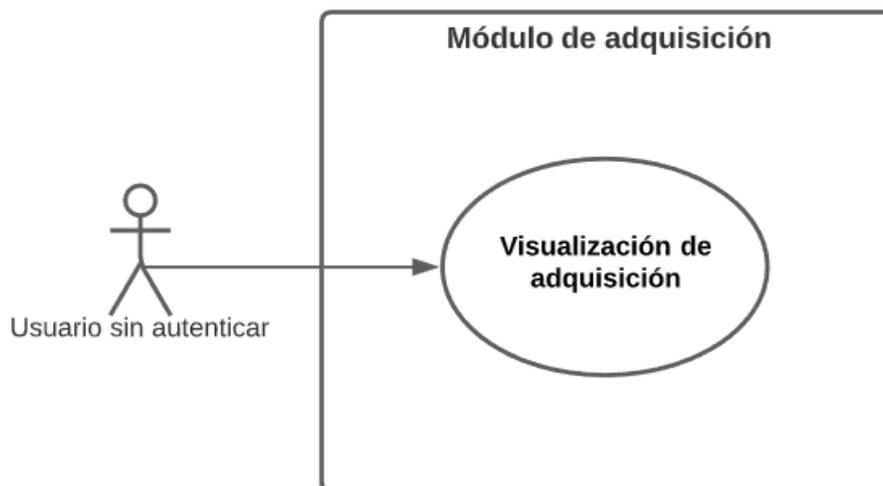


Ilustración 89: Diagrama de caso de uso CU-02: visualización del recurso

CU-02	Visualización del recurso	
Dependencias	<ul style="list-style-type: none"> R2.2.1 Puede acceder a información básica sobre adquisiciones. 	
Precondición	El usuario deberá acceder a esta interfaz mediante un enlace de la adquisición.	
Descripción	El sistema deberá comportarse como se describe en el siguiente caso de uso cuando <i>el usuario sin autenticar visualiza la información básica de una adquisición.</i>	
Secuencia normal	Paso	Acción
	1	El usuario accede a la interfaz mediante el enlace de la adquisición.
Comentarios	En esta interfaz, solo se podrán visualizar los valores característicos de la adquisición.	

Tabla 11: Caso de uso CU-02: visualización del recurso

3.7 Especificaciones de Construcción

En este apartado, se describen todos los componentes que se hacen uso en el módulo de gestión de inventario. Estos componentes son los paquetes, integrados y widgets que se han utilizado y están recogidos en el siguiente diagrama de paquetes:

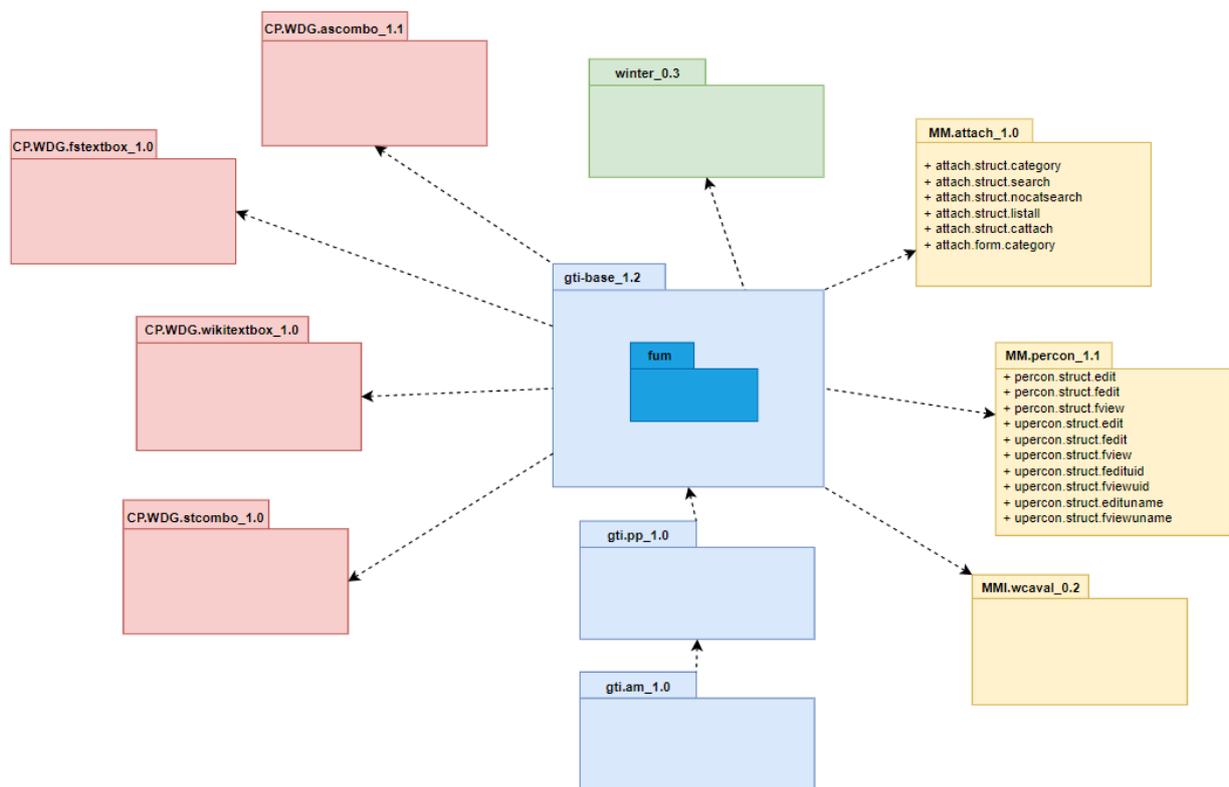


Ilustración 90: Diagrama de paquetes

A continuación, se describen los componentes definidos:

- *CP.WDG.fstextbox_1.0*: widget de cuadro de texto que muestra siempre el mismo ancho y alto.
- *CP.WDG.wikitextbox_1.0*: widget de cuadro de texto configurable para la wiki.
- *CP.WDG.stcombo_1.0*: widget de tipo combo que permite establecer transiciones entre los distintos elementos del combo.
- *CP.WDG.ascombo_1.0*: widget de tipo combo que permite ofrecer una búsqueda alternativo para hacer un "join visual".
- *MM.attach_1.0*: paquete que proporciona adjuntos categorizados genéricos.
- *MMI.percon_1.0*: paquete que permite gestionar información de contacto de personas y usuarios de la aplicación.
- *MMI.wcaval_0.2*: paquete que permite campos personalizado para cualquier entidad en el Modelo de Dominio.
- *Winter_0.3*: integrador de módulos de **WAINE**.
- *Gti-base_1.2*: paquete que permite usar los componentes definidos en el módulo común.
 - *Fum*: paquete que provee los formularios para manipular la información sobre usuarios almacenada en el repositorio de modelos.
- *Gti.am_1.0*: paquete que permite usar los componentes definidos en el módulo de gestión de inventario.
- *Gti.pp_1.0*: paquete que permite usar los componentes definidos en el módulo de adquisiciones.



3.8 Carga inicial de datos

3.8.1 Carga inicial de datos módulo am

La carga inicial de datos permite al módulo de inventario que pueda funcionar correctamente. Este proceso está compuesto por:

- Tabla Attach_Category: deben añadirse las siguientes categorías de adjuntos:
 - am.asset (con pk=9).
 - am.input (con pk=10).
 - am.output (con pk=11).
- Tabla Status del módulo de inventario. Deben añadirse los siguientes estados de los recursos predefinidos:
 - Activo.
 - Baja.
- Tabla InputType del módulo de inventario. Deben añadirse los siguientes tipos de entradas predefinidos:
 - Adquisición.
 - Interna.
 - Devolución.
- Tabla OutputType del módulo de inventario. Deben añadirse los siguientes tipos de salidas predefinidos:
 - Interna.
 - Préstamo.
 - Baja.

Esto se recoge en el fichero “am_DATA_simple.sql”.

Por otro lado, para la verificación de las pruebas, se ha creado además un fichero llamado “am.DATA_pruebas.sql”. Este fichero añade entradas en las siguientes tablas:

- am.Unit y am.UnitType.
- am.AssetType, am.AssetTypeAttr y am.ModelTypeAttr.

am.Mark y am.Model.

3.8.2 Carga inicial de datos módulo gti

La carga inicial de datos permite al módulo común y sus extensiones que pueda funcionar correctamente. Este proceso está compuesto principalmente por añadir localizaciones y unidades organizativas en las tablas:

- Localizaciones:
 - gti.Scope.
 - gti.Site.
 - gti.LL1.
 - gti.LL2.
- Unidades organizativas:
 - gti.OUL1.
 - gti.OUL2.
 - gti.OUL3.

En este caso, se han añadido tanto localizaciones, como unidades organizativas para las siguientes universidades:

- Universidad de Sevilla.
- Universidad de Córdoba.
- Universidad de Cádiz.

Esto se encuentra en el fichero “gti_DATA.sql”.

3.8.3 Carga inicial de datos módulo pp

La carga inicial de datos permite al módulo de adquisiciones que pueda funcionar correctamente. Este proceso está compuesto por:

- Tabla *Attach_Category*: deben añadirse las siguientes categorías de adjuntos:
 - pp.offer (con pk=3).
 - pp.prov (con pk=7).
 - pp.purch (con pk=8).

Además, existe el fichero llamado “pp_DATA_pruebas”, que se ha utilizado para realizar las pruebas del sistema que crea tres proveedores en la tabla “pp.Purchase”.

3.9 Plan de pruebas técnico

3.9.1 Plan de pruebas técnico módulo am

En este apartado se describen las pruebas que permiten verificar el correcto funcionamiento del módulo de inventario en todas sus interfaces según cada rol del sistema. Además, se incluye las matrices de trazabilidad en las que se relacionan cada una de las pruebas con los requisitos descritos en el documento de Análisis.

3.9.1.1 Tablas de pruebas

A continuación, se muestra una tabla para cada rol del sistema, con todas las pruebas a realizar.

3.9.1.1.1 Tabla de pruebas administrador del módulo de inventario

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol administrador del módulo de inventario.

Código de prueba	Prueba	Descripción
PRU_AMI-01	Asignación de permisos a un usuario.	El administrador del módulo accede a la interfaz <i>Administración de permisos</i> . En el formulario, elige el ámbito sobre el que quiere dar permisos y en las pestañas inferiores, elige el grupo al que quiere otorgarle los permisos (<i>Gestor adquisiciones, Gestor de inventario, Técnico avanzado o Técnico</i>). Una vez hecho esto, elige el usuario deseado y pulsa <i>Añadir</i> .
PRU_AMI-02	Denegación de permisos a un usuario.	El administrador del módulo accede a la interfaz <i>Administración de permisos</i> . En el formulario, elige el ámbito sobre el que quiere denegar permisos y en las pestañas



3 Diseño de los módulos de activos, adquisiciones y común de GTI

		inferiores, elige el grupo que pertenece el usuario al quiere denegarle los permisos (<i>Gestor adquisiciones, Gestor de inventario, Técnico avanzado o Técnico</i>). Una vez hecho esto, elige el usuario deseado y pulsa <i>Eliminar</i> .
PRU_AMI-03	Definición de tipos de entradas.	El administrador del módulo accede a la interfaz <i>Tipos de entrada</i> . En la tabla, escribe el tipo de entrada a crear y pulsa <i>Añadir</i> .
PRU_AMI-04	Eliminación de un tipo de entrada predefinido.	El administrador del módulo accede a la interfaz <i>Tipos de entrada</i> . En la tabla, elige uno de los tipos: <i>Adquisición, Interna o Devolución</i> y pulsa <i>Eliminar</i> . La aplicación responde con un mensaje de error “ <i>Eliminación abortada. Este tipo de entrada esta predefinido</i> ”.
PRU_AMI-05	Eliminación de un tipo de entrada no predefinido.	El administrador del módulo accede a la interfaz <i>Tipos de entrada</i> . En la tabla, elige el tipo de entrada no predefinido y pulsa <i>Eliminar</i> .
PRU_AMI-06	Definición de tipos de salidas.	El administrador del módulo accede a la interfaz <i>Tipos de salida</i> . En la tabla, escribe el tipo de salida a crear y pulsa <i>Añadir</i> .
PRU_AMI-07	Eliminación de un tipo de salida predefinido.	El administrador del módulo accede a la interfaz <i>Tipos de salida</i> . En la tabla, elige uno de los tipos: <i>Interna, Préstamo o Baja</i> y pulsa <i>Eliminar</i> . La aplicación responde con un mensaje de error “ <i>Eliminación abortada. Este tipo de salida esta predefinido</i> ”.
PRU_AMI-08	Eliminación de un tipo de salida no predefinido	El administrador del módulo accede a la interfaz <i>Tipos de salida</i> . En la tabla, elige el tipo de salida no predefinido y pulsa <i>Eliminar</i> .
PRU_AMI-09	Definición de estados de los recursos.	El administrador del módulo accede a la interfaz <i>Estados de los recursos</i> . En la tabla, escribe el estado del recurso a crear y pulsa <i>Añadir</i> .
PRU_AMI-10	Eliminación de un estado de los recursos predefinido.	El administrador del módulo accede a la interfaz <i>Estados de los recursos</i> . En la tabla, elige uno de los tipos: <i>Activo o Baja</i> y pulsa <i>Eliminar</i> . La aplicación responde con un mensaje de error “ <i>Eliminación abortada. Este estado esta predefinido</i> ”.
PRU_AMI-11	Eliminación de un estado de los recursos no predefinido.	El administrador del módulo accede a la interfaz <i>Estados de los recursos</i> . En la tabla, elige el estado de recurso no predefinido y pulsa <i>Eliminar</i> .

PRU_AMI-12	Definición de tipos de unidades y unidades.	El administrador del módulo accede a la interfaz <i>Tipos de unidades y unidades</i> . En la tabla superior, crea un tipo de unidad y pulsa <i>Añadir</i> . Una vez creado, en la tabla inferior, crea diferentes formatos de unidades para dicho tipo de unidad y pulsa <i>Añadir</i> .
PRU_AMI-13	Eliminación de tipos de unidades y unidades.	El administrador del módulo accede a la interfaz <i>Tipos de unidades y unidades</i> . En el formulario superior, elige un tipo de unidad y pulsa <i>Eliminar</i> . Por otra parte, en la tabla inferior, elige entre diferentes formatos de unidades para dicho tipo de unidad y pulsa <i>Eliminar</i> .
PRU_AMI-14	Definición de tipos de elementos inventariable y de sus atributos.	El administrador del módulo accede a la interfaz <i>Tipos de elementos y atributos</i> . En la tabla superior, crea un tipo de elemento inventariable y pulsa <i>Añadir</i> . Una vez creado, en las dos tablas inferiores, crea tantos atributos de tipo modelo como atributos de tipo activo y pulsa <i>Añadir</i> .
PRU_AMI-15	Eliminación de tipos de elementos inventariable y de sus atributos.	El administrador del módulo accede a la interfaz <i>Tipos de elementos y atributos</i> . En el formulario superior, elige un tipo de elemento inventariable y pulsa <i>Eliminar</i> . Por otra parte, en las dos tablas inferiores, elige tantos atributos de tipo modelo como atributos de tipo activo y pulsa <i>Eliminar</i> .
PRU_AMI-16	Consulta de los recursos asignados.	El administrador del módulo accede a la interfaz <i>Consulta de recursos asignados</i> . En el navegador, elige el activo que desea visualizar según el identificador y número de serie. En las pestañas inferiores (<i>Propiedades y Recurso Hijo</i>), puede ver las propiedades y los recursos hijos de dicho activo.

Tabla 12: Pruebas del administrador del módulo de inventario

3.9.1.1.2 Tabla de pruebas gestor de inventario

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol gestor de inventario.

Código de prueba	Prueba	Descripción
PRU_GI-01	Definición de marcas y modelos que determinan a un activo.	El gestor de inventario accede a la interfaz <i>Marcas, modelos y atributos</i> . En el formulario superior, crea una nueva marca y pulsa <i>Añadir</i> . Posteriormente, en el formulario inferior, crea un nuevo modelo,



3 Diseño de los módulos de activos, adquisiciones y común de GTI

		<p>eligiendo el tipo de modelo y pulsa <i>Añadir</i>. Una vez hecho esto, se copiarán todos los atributos del tipo de modelo elegido creados por el gestor del módulo de inventario en PRU_AMI-14 y puede visualizarlo en la tabla inferior llamada <i>Atributos</i>.</p>
PRU_GI-02	Eliminación de marcas y modelos que definen a un activo.	<p>El gestor de inventario accede a la interfaz <i>Marcas, modelos y atributos</i>. En el formulario superior, elige la marca a eliminar y pulsa <i>Eliminar</i>.</p> <p>Por otro lado, en el formulario inferior, elige el modelo a eliminar y pulsa <i>Eliminar</i>.</p>
PRU_GI-03	Tramitación de una entrada para una adquisición.	<p>El gestor de inventario accede a la interfaz <i>Gestión de entradas</i>. En la tabla <i>Entrada</i>, rellena los campos necesarios, eligiendo “Adquisición” en el campo <i>Tipo</i> y la adquisición deseada en el campo <i>Adquisicion</i> y pulsa <i>Añadir</i>. Posteriormente, se copiarán todos los elementos de dicha adquisición sin número de serie o identificador que podrá visualizarlos en la pestaña <i>Activos</i>. En esta misma pestaña, crea otros activos que van a estar relacionados con dicha entrada.</p> <p>Por otro lado, añade un adjunto en la pestaña <i>Adjuntos</i>.</p>
PRU_GI-04	Tramitación de una entrada para una salida.	<p>El gestor de inventario accede a la interfaz <i>Gestión de entradas</i>. En la tabla <i>Entrada</i>, rellena los campos necesarios, eligiendo “Interna” o “Devolución” en el campo <i>Tipo</i> y la salida deseada (creada en PRU_GI-08) en el campo <i>Salida</i> y pulsa <i>Añadir</i>. En la pestaña <i>Activos</i>, crea un activo, asignándole un número de serie y un identificador, que va a estar relacionado con dicha entrada.</p> <p>Por otro lado, añade un adjunto en la pestaña <i>Adjuntos</i>.</p>
PRU_GI-05	Visualización de los activos asociados a una entrada.	<p>El gestor de inventario accede a la interfaz <i>Gestión de entradas</i>. En tabla superior, rellena los campos para la búsqueda de la entrada según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la entrada y, en la pestaña <i>Resumen</i>, visualiza la cantidad de activos según el tipo de activo.</p>

		<p>Por otro lado, se puede ver más detalladamente un activo en particular, en la pestaña <i>Activos</i>, pulsando el botón <i>Ver</i>.</p>
PRU_GI-06	Generación de un informe de una entrada.	<p>El gestor de inventario accede a la interfaz <i>Gestión de entradas</i>. En tabla superior, rellena los campos para la búsqueda de la entrada según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la entrada y pulsa <i>Informe</i>.</p>
PRU_GI-07	Eliminación de una entrada.	<p>El gestor de inventario accede a la interfaz <i>Gestión de entradas</i>. En tabla superior, rellena los campos para la búsqueda de la entrada según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la entrada y pulsa <i>Eliminar</i>.</p> <p>Por otro lado, en la pestaña <i>Activos</i>, elige un activo y pulsa <i>Eliminar</i>. De esta forma, estaremos eliminando el activo de manera permanente.</p>
PRU_GI-08	Tramitación de una salida.	<p>El gestor de inventario accede a la interfaz <i>Gestión de salidas</i>. En la tabla <i>Salida</i>, rellena los campos necesarios y pulsa <i>Añadir</i>. En la pestaña <i>Activos</i>, añade activos que van a estar relacionados con dicha salida.</p> <p>Por otro lado, añade un adjunto en la pestaña <i>Adjuntos</i>.</p>
PRU_GI-09	Visualización de los activos asociados a una salida.	<p>El gestor de inventario accede a la interfaz <i>Gestión de salidas</i>. En tabla superior, rellena los campos para la búsqueda de la salida según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la salida y, en la pestaña <i>Resumen</i>, visualiza la cantidad de activos según el tipo de activo.</p> <p>Por otro lado, se puede ver más detalladamente un activo en particular, en la pestaña <i>Activos</i>, pulsando el botón <i>Ver</i>.</p>
PRU_GI-10	Generación de un informe de una salida.	<p>El gestor de inventario accede a la interfaz <i>Gestión de salidas</i>. En tabla superior, rellena los campos para la búsqueda de la salida según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la salida y pulsa <i>Informe</i>.</p>
PRU_GI-11	Eliminación de una salida.	<p>El gestor de inventario accede a la interfaz <i>Gestión de salida</i>. En tabla superior, rellena los campos para la búsqueda de la salida</p>



3 Diseño de los módulos de activos, adquisiciones y común de GTI

		<p>según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la salida y pulsa <i>Eliminar</i>.</p> <p>Por otro lado, en la pestaña <i>Activos</i>, elige un activo y pulsa <i>Eliminar</i>. De esta forma, eliminaremos solo la asociación del activo con la salida seleccionada.</p>
PRU_GI-12	Búsqueda de un activo según sus propiedades.	El gestor de inventario accede a la interfaz <i>Gestión de inventario</i> . En la tabla <i>Filtros</i> , elige los campos de los atributos sobre los que filtrar los activos y pulsa <i>Filtrar</i> . En la tabla inferior llamada <i>Activo</i> , aparecerán los activos que cumplan con los filtros marcados. Además, en esta tabla se podrán hacer cambios en los atributos de dichos activos filtrados y podrá acceder a la interfaz <i>Edición de activos</i> (descrito en PRU_GI-14), pulsando el <i>Ver</i> .
PRU_GI-13	Búsqueda de un activo según el tipo de recurso.	El gestor de inventario accede a la interfaz <i>Gestión de inventario por tipo de recurso</i> . En la tabla <i>Filtros</i> , elige el tipo de recurso en el campo <i>Tipo de recurso</i> sobre el que filtrará los activos y pulsa <i>Filtrar</i> . En la tabla inferior llamada <i>Activo</i> , aparecerán los activos que cumplan con el tipo de recurso marcado. Además, en esta tabla se podrán hacer cambios en los atributos de dichos activos filtrados y podrá acceder a la interfaz <i>Edición de activos</i> (descrito en PRU_GI-14), pulsando el <i>Ver</i> .
PRU_GI-14	Edición de un activo.	<p>Esta interfaz será accedida a través de las pruebas PRU_GI-12 y PRU_GI-13, pulsando el botón <i>Ver</i>, con se indica en dichas pruebas. Una vez hecho esto, se podrá modificar los atributos del activo, pulsando <i>Modificar</i>.</p> <p>Además, se visualizará las propiedades del activo en la pestaña <i>Propiedades</i>, los recursos hijos en la pestaña <i>Recursos Hijos</i>, se podrá añadir anotaciones sobre posibles incidencias en la pestaña <i>Histórico</i>, se podrá visualizar todas las modificaciones realizadas sobre dicho activo en la pestaña <i>Traza</i> y se podrá añadir un adjunto en la pestaña <i>Adjuntos</i>.</p>
PRU_GI-15	Gestión de un activo.	El gestor de inventario accede a la interfaz <i>Gestión de activos</i> . En los dos combos superiores, elige el ámbito y la localización

		<p>que desea filtrar los atributos. Posteriormente, en el navegador, elige el activo que desea gestionar, pudiendo modificar sus atributos, pulsando <i>Modificar</i>.</p> <p>Además, se visualizará las propiedades del activo en la pestaña <i>Propiedades</i>, los recursos hijos en la pestaña <i>Recursos Hijos</i>, se podrá añadir anotaciones sobre posibles incidencias en la pestaña <i>Histórico</i>, se podrá visualizar todas las modificaciones realizadas sobre dicho activo en la pestaña <i>Traza</i> y se podrá añadir un adjunto en la pestaña <i>Adjuntos</i>.</p>
PRU_GI-16	Generación de un informe de un activo.	En las pruebas PRU_GI-14 y PRU_GI-15 , en la tabla <i>Activo</i> , habrá un botón <i>Informe</i> , que genera un informe con los datos más relevantes del activo seleccionado.
PRU_GI-17	Visualización de estadísticas según los tipos de activos por sede.	El gestor de inventario accede a la interfaz <i>Estadísticas de tipos de activos por sede</i> . En el combo superior, elige la sede que desea filtrar los activos. Posteriormente, en las tablas inferiores, podrá visualizar un resumen tanto en gráfica como en tabla de todos los activos según los distintos tipos de activos.
PRU_GI-18	Visualización de estadísticas según los estados y marcas de los activos por sede y tipo de activo.	El gestor de inventario accede a la interfaz <i>Estadísticas de estados y marcas de activos por sede y tipo</i> . En el combo superior, elige la sede que desea filtrar los activos. Posteriormente, en las tablas inferiores, podrá visualizar un resumen, dos a dos, tanto en gráfica como en tabla de todos los activos según los distintos estados y marcas, respectivamente.
PRU_GI-19	Consulta de los recursos asignados.	El gestor de inventario accede a la interfaz <i>Consulta de recursos asignados</i> . En el navegador, elige el activo que desea visualizar según el identificador y número de serie. En las pestañas inferiores (<i>Propiedades y Recurso Hijo</i>), puede ver las propiedades y los recursos hijos de dicho activo.

Tabla 13: Pruebas del gestor de inventario

3.9.1.1.3 Tabla de pruebas técnico avanzado

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol técnico avanzado.

Código de prueba	Prueba	Descripción
------------------	--------	-------------



3 Diseño de los módulos de activos, adquisiciones y común de GTI

PRU_TCAV-01	Tramitación de una entrada para una adquisición.	<p>El técnico avanzado accede a la interfaz <i>Gestión de entradas</i>. En la tabla <i>Entrada</i>, rellena los campos necesarios, eligiendo “Adquisición” en el campo <i>Tipo</i> y la adquisición deseada en el campo <i>Adquisicion</i> y pulsa <i>Añadir</i>. Posteriormente, se copiarán todos los elementos de dicha adquisición sin número de serie o identificador que podrá visualizarlos en la pestaña <i>Activos</i>. En esta misma pestaña, crea otros activos que van a estar relacionados con dicha entrada.</p> <p>Por otro lado, añade un adjunto en la pestaña <i>Adjuntos</i>.</p>
PRU_TCAV-02	Tramitación de una entrada para una salida.	<p>El técnico avanzado accede a la interfaz <i>Gestión de entradas</i>. En la tabla <i>Entrada</i>, rellena los campos necesarios, eligiendo “Interna” o “Devolución” en el campo <i>Tipo</i> y la salida deseada (creada en PRU_GI-08) en el campo <i>Salida</i> y pulsa <i>Añadir</i>. En la pestaña <i>Activos</i>, crea un activo, asignándole un número de serie y un identificador, que va a estar relacionado con dicha entrada.</p> <p>Por otro lado, añade un adjunto en la pestaña <i>Adjuntos</i>.</p>
PRU_TCAV-03	Visualización de los activos asociados a una entrada.	<p>El técnico avanzado accede a la interfaz <i>Gestión de entradas</i>. En tabla superior, rellena los campos para la búsqueda de la entrada según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la entrada y, en la pestaña <i>Resumen</i>, visualiza la cantidad de activos según el tipo de activo.</p> <p>Por otro lado, se puede ver más detalladamente un activo en particular, en la pestaña <i>Activos</i>, pulsando el botón <i>Ver</i>.</p>
PRU_TCAV-04	Generación de un informe de una entrada.	<p>El técnico avanzado accede a la interfaz <i>Gestión de entradas</i>. En tabla superior, rellena los campos para la búsqueda de la entrada según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la entrada y pulsa <i>Informe</i>.</p>
PRU_TCAV-05	Eliminación de una entrada.	<p>El técnico avanzado accede a la interfaz <i>Gestión de entradas</i>. En tabla superior, rellena los campos para la búsqueda de la entrada según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la entrada y pulsa <i>Eliminar</i>.</p>

		<p>Por otro lado, en la pestaña <i>Activos</i>, elige un activo y pulsa <i>Eliminar</i>. De esta forma, estaremos eliminando el activo de manera permanente.</p>
PRU_TCAV-06	Tramitación de una salida.	<p>El técnico avanzado accede a la interfaz <i>Gestión de salidas</i>. En la tabla <i>Salida</i>, rellena los campos necesarios y pulsa <i>Añadir</i>. En la pestaña <i>Activos</i>, añade activos que van a estar relacionados con dicha salida.</p> <p>Por otro lado, añade un adjunto en la pestaña <i>Adjuntos</i>.</p>
PRU_TCAV-07	Visualización de los activos asociados a una salida.	<p>El técnico avanzado accede a la interfaz <i>Gestión de salidas</i>. En tabla superior, rellena los campos para la búsqueda de la salida según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la salida y, en la pestaña <i>Resumen</i>, visualiza la cantidad de activos según el tipo de activo.</p> <p>Por otro lado, se puede ver más detalladamente un activo en particular, en la pestaña <i>Activos</i>, pulsando el botón <i>Ver</i>.</p>
PRU_TCAV-08	Generación de un informe de una salida.	<p>El técnico avanzado accede a la interfaz <i>Gestión de salidas</i>. En tabla superior, rellena los campos para la búsqueda de la salida según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la salida y pulsa <i>Informe</i>.</p>
PRU_TCAV-09	Eliminación de una salida.	<p>El técnico avanzado accede a la interfaz <i>Gestión de salida</i>. En tabla superior, rellena los campos para la búsqueda de la salida según sus atributos (<i>ID, Tipo, Emisor, Receptor, ...</i>) y pulsa <i>Filtrar</i>. En el navegador, elige la salida y pulsa <i>Eliminar</i>.</p> <p>Por otro lado, en la pestaña <i>Activos</i>, elige un activo y pulsa <i>Eliminar</i>. De esta forma, eliminaremos solo la asociación del activo con la salida seleccionada.</p>
PRU_TCAV-10	Búsqueda de un activo según sus propiedades.	<p>El técnico avanzado accede a la interfaz <i>Gestión de inventario reducida</i>. En la tabla <i>Filtros</i>, elige los campos de los atributos sobre los que filtrar los activos y pulsa <i>Filtrar</i>. En la tabla inferior llamada <i>Activo</i>, aparecerán los activos que cumplan con los filtros marcados. Además, en esta tabla se podrán hacer cambios en los atributos de</p>



3 Diseño de los módulos de activos, adquisiciones y común de GTI

		dichos activos filtrados y podrá acceder a la interfaz <i>Edición de activos</i> (descrito en PRU_TCAV-11), pulsando el <i>Ver</i> . Esta interfaz es igual a la descrita en PRU_GI-12 , pero con algunos campos solo lectura.
PRU_TCAV-11	Edición de un activo.	Esta interfaz será accedida a través de la prueba PRU_TCAV-10 , pulsando el botón <i>Ver</i> , con se indica en dicha prueba. Una vez hecho esto, se podrá modificar los atributos del activo, pulsando <i>Modificar</i> . Además, se visualizará las propiedades del activo en la pestaña <i>Propiedades</i> , los recursos hijos en la pestaña <i>Recursos Hijos</i> , se podrá añadir anotaciones sobre posibles incidencias en la pestaña <i>Histórico</i> , se podrá visualizar todas las modificaciones realizadas sobre dicho activo en la pestaña <i>Traza</i> y se podrá añadir un adjunto en la pestaña <i>Adjuntos</i> . Esta interfaz es igual a la descrita en PRU_GI-14 , pero con algunos campos solo lectura.
PRU_TCAV-12	Generación de un informe de un activo.	En las pruebas PRU_TCAV-10 y PRU_TCAV-11 , en la tabla <i>Activo</i> , habrá un botón <i>Informe</i> , que genera un informe con los datos más relevantes del activo seleccionado.
PRU_TCAV-13	Consulta de los recursos asignados.	El técnico avanzando accede a la interfaz <i>Consulta de recursos asignados</i> . En el navegador, elige el activo que desea visualizar según el identificador y número de serie. En las pestañas inferiores (<i>Propiedades</i> y <i>Recurso Hijo</i>), puede ver las propiedades y los recursos hijos de dicho activo.

Tabla 14: Pruebas del técnico avanzado

3.9.1.1.4 Tabla de pruebas técnico

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol técnico.

Código de prueba	Prueba	Descripción
PRU_TC-01	Búsqueda de un activo según sus propiedades.	El técnico accede a la interfaz <i>Gestión de inventario reducida</i> . En la tabla <i>Filtros</i> , elige los campos de los atributos sobre los que filtrar los activos y pulsa <i>Filtrar</i> . En la tabla inferior llamada <i>Activo</i> , aparecerán los activos que cumplan con los filtros marcados. Además, en esta tabla se podrán hacer cambios en los atributos de dichos activos filtrados y podrá acceder a la interfaz <i>Edición de activos</i> (descrito en PRU_TC-02),

		pulsando el <i>Ver</i> . Esta interfaz es igual a la descrita en PRU_GI-12 , pero con algunos campos solo lectura.
PRU_TC-02	Edición de un activo.	Esta interfaz será accedida a través de la prueba PRU_TC-01 , pulsando el botón <i>Ver</i> , con se indica en dicha prueba. Una vez hecho esto, se podrá modificar los atributos del activo, pulsando <i>Modificar</i> . Además, se visualizará las propiedades del activo en la pestaña <i>Propiedades</i> , los recursos hijos en la pestaña <i>Recursos Hijos</i> , se podrá añadir anotaciones sobre posibles incidencias en la pestaña <i>Histórico</i> , se podrá visualizar todas las modificaciones realizadas sobre dicho activo en la pestaña <i>Traza</i> y se podrá añadir un adjunto en la pestaña <i>Adjuntos</i> . Esta interfaz es igual a la descrita en PRU_GI-14 , pero con algunos campos solo lectura.
PRU_TC-03	Generación de un informe de un activo.	En las pruebas PRU_TC-01 y PRU_TC-02 , en la tabla <i>Activo</i> , habrá un botón <i>Informe</i> , que genera un informe con los datos más relevantes del activo seleccionado.
PRU_TC-04	Consulta de los recursos asignados.	El técnico accede a la interfaz <i>Consulta de recursos asignados</i> . En el navegador, elige el activo que desea visualizar según el identificador y número de serie. En las pestañas inferiores (<i>Propiedades</i> y <i>Recurso Hijo</i>), puede ver las propiedades y los recursos hijos de dicho activo.

Tabla 15: Pruebas del técnico

3.9.1.1.5 Tabla de pruebas usuario regular

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol usuario regular.

Código de prueba	Prueba	Descripción
PRU_US-01	Consulta de los recursos asignados.	El usuario regular accede a la interfaz <i>Consulta de recursos asignados</i> . En el navegador, elige el activo que desea visualizar según el identificador y número de serie. En las pestañas inferiores, puede ver las propiedades y los recursos hijos de dicho activo.

Tabla 16: Pruebas del usuario regular

3.9.1.1.6 Tabla de pruebas usuario sin acceso

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol usuario sin acceso.



Código de prueba	Prueba	Descripción
PRU_USSA-01	Visualización de un recurso.	El usuario sin acceso busca un enlace a un recurso que no tenga acceso o lo crea en su defecto (R#ID). Una vez obtenido, hace click sobre el enlace y comprueba que se le muestra la información mínima, en modo lectura, del recurso seleccionada de acuerdo con la interfaz <i>Visualización de recurso</i> .
PRU_USSA -02	Visualización de una entrada.	El usuario sin acceso busca un enlace a una entrada que no tenga acceso o lo crea en su defecto (E#ID). Una vez obtenido, hace click sobre el enlace y comprueba que se le muestra la información mínima, en modo lectura de la entrada seleccionada de acuerdo con la interfaz <i>Visualización de entrada</i> .
PRU_USSA -03	Visualización de una salida.	El usuario sin acceso busca un enlace a una salida que no tenga acceso o lo crea en su defecto (S#ID). Una vez obtenido, hace click sobre el enlace y comprueba que se le muestra la información mínima, en modo lectura de la salida seleccionada de acuerdo con la interfaz <i>Visualización de salida</i> .

Tabla 17: Pruebas del usuario sin acceso

3.9.1.2 Matriz de trazabilidad

A continuación, se muestra una matriz de trazabilidad para cada usuario de la aplicación para relacionar los requisitos definidos en el Documento de Análisis con las pruebas detalladas en el apartado anterior.

3.9.1.2.1 Matriz de trazabilidad administrador del módulo

En la siguiente tabla se recogen los requisitos que relacionan las pruebas realizadas en el apartado 9.1.1 para un usuario con rol administrador del módulo.

REQUISITOS DE USUARIO	PRUEBAS															
	PRU_AMI-01	PRU_AMI_02	PRU_AMI-03	PRU_AMI-04	PRU_AMI-05	PRU_AMI-06	PRU_AMI-07	PRU_AMI-08	PRU_AMI-09	PRU_AMI-10	PRU_AMI-11	PRU_AMI-12	PRU_AMI-13	PRU_AMI-14	PRU_AMI-15	PRU_AMI-16
R2.1.1	X	X														



Tabla 20: Matriz de trazabilidad del técnico avanzado

3.9.1.2.4 Matriz de trazabilidad técnico

En la siguiente tabla se recogen los requisitos que relacionan las pruebas realizadas en el apartado 9.1.4 para un usuario con rol técnico.

REQUISITOS DE USUARIO	PRUEBAS			
	PRU_TC-01	PRU_TC_02	PRU_TC-03	PRU_TC-04
R2.4.1	X	X		
R2.4.2			X	
R2.5.1				X

Tabla 21: Matriz de trazabilidad del técnico

3.9.1.2.5 Matriz de trazabilidad usuario regular

En la siguiente tabla se recogen los requisitos que relacionan las pruebas realizadas en el apartado 9.1.5 para un usuario con rol usuario regular.

REQUISITOS DE USUARIO	PRUEBAS
	PRU_US-01
R2.5.1	X

Tabla 22: Matriz de trazabilidad del usuario regular

3.9.1.2.6 Matriz de trazabilidad usuario sin acceso

En la siguiente tabla se recogen los requisitos que relacionan las pruebas realizadas en el apartado 9.1.6 para un usuario con rol usuario sin acceso.

REQUISITOS DE USUARIO	PRUEBAS		
	PRU_USSA-01	PRU_USSA_02	PRU_USSA-03
R2.6.1	X		
R2.6.2		X	X

Tabla 23: Matriz de trazabilidad del usuario sin acceso

3.9.2 Plan de pruebas técnico módulo gti

En este apartado se describen las pruebas que permiten verificar el correcto funcionamiento del módulo de inventario en todas sus interfaces según cada rol del sistema.

3.9.2.1 Tablas de pruebas

A continuación, se muestra una tabla para cada rol del sistema, con todas las pruebas a realizar.

3.9.2.1.1 Tabla de pruebas administrador del módulo común

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol administrador del módulo común.

Código de prueba	Prueba	Descripción
PRU_AMC-01	Definición de un nuevo ámbito.	El administrador del módulo común accede a la interfaz <i>Gestión de ámbitos</i> . En la tabla, escribe el ámbito a crear y pulsa <i>Añadir</i> .
PRU_AMC-02	Definición de una nueva unidad organizativa.	El administrador del módulo común accede a la interfaz <i>Gestión de unidades organizativas</i> . En la tabla superior, escribe el nivel 1 y pulsa <i>Añadir</i> . Una vez añadido, elige el nivel añadido y, en la tabla central, escribe el nivel 2 y pulsa <i>Añadir</i> . Una vez añadido, elige el nivel añadido y, en la tabla inferior, escribe el nivel 3 y pulsa <i>Añadir</i> .
PRU_AMC-03	Definición de una nueva localización.	El administrador del módulo común accede a la interfaz <i>Gestión de localizaciones</i> . En la tabla superior, escribe la sede y pulsa <i>Añadir</i> . Una vez añadido, elige la sede añadida y, en la tabla central, escribe el nivel 1 y pulsa <i>Añadir</i> . Una vez añadido, elige el nivel añadido y, en la tabla inferior, escribe el nivel 2, pudiendo indicar si se trata de un almacén o no, y pulsa <i>Añadir</i> .
PRU_AMC-04	Definición de un nuevo perfil.	El administrador del módulo común accede a la interfaz <i>Datos personales</i> . En la tabla superior, escribe los datos relativos al perfil y pulsa <i>Añadir</i> . Posteriormente, puede rellenar los datos relativos a teléfono, dirección, email y añadir notas en las pestañas <i>Telefono</i> , <i>Direccion</i> , <i>Email</i> y <i>Notas</i> , respectivamente.
PRU_AMC-05	Definición de un nuevo perfil para un usuario.	El administrador del módulo común accede a la interfaz <i>Datos personales de usuario</i> . En el combo superior, elige el usuario del sistema y pulsa <i>Filtrar</i> . A continuación, en la tabla superior, escribe los datos relativos al perfil y pulsa <i>Añadir</i> . Posteriormente, puede rellenar los datos relativos a teléfono, dirección, email y añadir notas en las



		pestañas <i>Telefono, Direccion, Email y Notas</i> , respectivamente
--	--	--

Tabla 24: Pruebas del administrador del módulo común

3.9.3 Plan de pruebas técnico módulo pp

En este apartado se describen las pruebas que permiten verificar el correcto funcionamiento del módulo de adquisición en todas sus interfaces según cada rol del sistema. Además, se incluye las matrices de trazabilidad en las que se relacionan cada una de las pruebas con los requisitos descritos en el documento de Análisis.

3.9.3.1 Tablas de pruebas

A continuación, se muestra una tabla para cada rol del sistema, con todas las pruebas a realizar.

3.9.3.1.1 Tabla de pruebas gestor de adquisiciones

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol gestor de adquisiciones.

Código de prueba	Prueba	Descripción
PRU_GA-01	Definición de un proveedor.	El gestor de adquisiciones accede a la interfaz <i>Gestión de proveedores</i> . En la tabla superior, rellena los campos necesarios para definir al proveedor y pulsa <i>Añadir</i> . Por otro lado, procede a añadir un contacto para dicho proveedor en la pestaña <i>Contactos</i> y, un nuevo adjunto en la pestaña <i>Adjuntos</i> .
PRU_GA-02	Eliminación de un perfil de proveedor.	El gestor de adquisiciones accede a la interfaz <i>Gestión de proveedores</i> . En formulario superior, elige al proveedor y pulsa <i>Eliminar</i> . Por otro lado, en la pestaña <i>Contactos</i> , elige un contacto para dicho proveedor y pulsa <i>Eliminar</i> .
PRU_GA-03	Creación de una solicitud de oferta.	El gestor de adquisiciones accede a la interfaz <i>Gestión de ofertas</i> . En el formulario superior, rellena la petición de oferta, indicando el ámbito y pulsa <i>Añadir</i> .
PRU_GA-04	Eliminación de una solicitud de oferta.	El gestor de adquisiciones accede a la interfaz <i>Gestión de ofertas</i> . En el formulario superior, elige la petición de oferta y pulsa <i>Eliminar</i> .
PRU_GA-05	Tramitación una oferta dada una solicitud.	El gestor de adquisiciones accede a la interfaz <i>Gestión de entradas</i> . En el formulario superior, elige la petición de oferta sobre la que quiere realizar una oferta. Posteriormente, en el formulario inferior,

		rellena los campos necesarios para crear una oferta y pulsa <i>Añadir</i> . Una vez hecho esto, procede a añadir nuevos elementos para dicha oferta en la pestaña <i>Elementos</i> y, un nuevo adjunto en la pestaña <i>Adjuntos</i> .
PRU_GA-06	Eliminación de una oferta dada una solicitud.	El gestor de adquisiciones accede a la interfaz <i>Gestión de entradas</i> . En el formulario superior, elige la petición de oferta sobre la que quiere eliminar una oferta. Posteriormente, en el formulario inferior, elige la oferta a eliminar y pulsa <i>Eliminar</i> .
PRU_GA-07	Tramitación de una adquisición dada una oferta.	El gestor de adquisiciones accede a la interfaz <i>Gestión de adquisiciones</i> . En los dos combos superiores, elige la petición de oferta y oferta sobre las que quiere realizar la compra. Posteriormente, en la tabla superior, rellena los campos necesarios para llevar a cabo la compra y pulsa <i>Añadir</i> . Una vez hecho esto, procede a añadir nuevos elementos para dicha compra, asignándoles un precio en la pestaña <i>Elementos</i> y, un nuevo adjunto en la pestaña <i>Adjuntos</i> .
PRU_GA-08	Eliminación de una adquisición dada una oferta.	El gestor de adquisiciones accede a la interfaz <i>Gestión de adquisiciones</i> . En los dos combos superiores, elige la petición de oferta y oferta de la compra a eliminar. Posteriormente, en la tabla superior, elige la compra y pulsa <i>Eliminar</i> .
PRU_GA-09	Definición de marcas y modelos que definen a un activo.	El gestor de adquisiciones accede a la interfaz <i>Marcas, modelos y atributos</i> . En el formulario superior, crea una nueva marca y pulsa <i>Añadir</i> . Posteriormente, en el formulario inferior, crea un nuevo modelo, eligiendo el tipo de modelo y pulsa <i>Añadir</i> . Una vez hecho esto, se copiarán todos los atributos del tipo de modelo elegido creados por el gestor del módulo de inventario en PRU_AMI-14 y puede otorgarles un valor en la tabla inferior llamada <i>Atributos</i> .
PRU_GA-10	Eliminación de marcas y modelos que definen a un activo.	El gestor de adquisiciones accede a la interfaz <i>Marcas, modelos y atributos</i> . En el formulario superior, elige la marca a eliminar y pulsa <i>Eliminar</i> . Por otro lado, en el formulario inferior, elige el modelo a eliminar y pulsa <i>Eliminar</i> .
PRU_GA-11	Consulta de los recursos asignados.	El gestor de adquisiciones accede a la interfaz <i>Consulta de recursos asignados</i> . En el navegador, elige el activo que desea



3 Diseño de los módulos de activos, adquisiciones y común de GTI

		visualizar según el identificador y número de serie. En las pestañas inferiores (<i>Propiedades y Recurso Hijo</i>), puede ver las propiedades y los recursos hijos de dicho activo.
--	--	--

Tabla 25: Pruebas del gestor de adquisiciones

3.9.3.1.2 Tabla de pruebas usuario sin acceso

En la siguiente tabla se describen las pruebas a realizar en las interfaces para un usuario con rol usuario sin acceso.

Código de prueba	Prueba	Descripción
PRU_USSA -01	Visualización de una adquisición.	El usuario sin acceso busca un enlace a una adquisición que no tenga acceso o lo crea en su defecto (A#ID). Una vez obtenido, hace click sobre el enlace y comprueba que se le muestra la información mínima, en modo lectura, de la adquisición seleccionada de acuerdo con la interfaz <i>Visualización de adquisición</i> .

Tabla 26: Pruebas del usuario sin acceso

3.9.3.2 Matriz de trazabilidad

A continuación, se muestra una matriz de trazabilidad para cada usuario de la aplicación para relacionar los requisitos definidos en el Documento de Análisis con las pruebas detalladas en el apartado anterior.

3.9.3.2.1 Matriz de trazabilidad gestor de adquisiciones

En la siguiente tabla se recogen los requisitos que relacionan las pruebas realizadas en el apartado 9.1.1 para un usuario con rol gestor de adquisiciones.

REQUISITOS DE USUARIO	PRUEBAS										
	PRU_GA-01	PRU_GA_02	PRU_GA-03	PRU_GA-04	PRU_GA-05	PRU_GA-06	PRU_GA-07	PRU_GA-08	PRU_GA-09	PRU_GA-10	PRU_GA-11
R2.1.1									X	X	
R2.1.2	X	X									
R2.1.3			X	X							
R2.1.4					X	X					
R2.1.5							X	X			
R2.5.1											X

Tabla 27: Matriz de trazabilidad del gestor de adquisiciones

Nota: el requisito R2.5.1 proviene del módulo de inventario.

3.9.3.2.2 Matriz de trazabilidad usuario regular

En la siguiente tabla se recogen los requisitos que relacionan las pruebas realizadas en el apartado 9.1.2 para un usuario con rol usuario sin acceso.

REQUISITOS DE USUARIO	PRUEBAS
	PRU_USSA-01
R2.2.1	X

Tabla 28: Matriz de trazabilidad del usuario regular

3.10 Requisitos de implantación

A continuación, se detallan los requisitos de implantación necesarios en el módulo de inventario de GTI:

- Versión Waine 0.5.6.
- Versión PostgreSQL superior o igual a 8.1.19.
- Versión PHP superior o igual a 4.3.10.
- Versión Apache superior o igual a 1.3.34.
- Winter 0.3.



4 Construcción de los módulos de activos, adquisiciones y común de GTI

4.1 Introducción

En este documento se explica el proceso de implementación referente al módulo de inventario, módulo común y módulo de adquisiciones de la aplicación GTI. Para este proceso, debido al entorno de desarrollo de interfaces de usuario de **WAINE**, se definen los siguientes modelos:

- Modelo de dominio: describe los datos de usuarios que los usuarios manejan empleando la interfaz de usuario y se especifica haciendo uso de los Diagramas Entidad-Relación. Sus principales componentes son entidades y relaciones. Se implementa en el lenguaje SQL.
- Modelo de presentación: describe la interfaz abstracta mediante Objetos Abstractos de Interacción. Sus principales componentes son formularios (eventos y acciones) y contenedores. Se implementa en el lenguaje ASL.
- Modelo de usuario: describe la organización jerárquica de los usuarios que se agrupan en un grupo de acuerdo con sus roles. Sus principales componentes son los grupos y los usuarios. Se implementa en el lenguaje ASL.
- Modelo de diálogo: describe las opciones de menú accesibles a los usuarios de acuerdo con el grupo. Sus principales componentes son menús, opciones y acciones. Se implementa en el lenguaje ASL.

En los siguientes apartados, se va a detallar la implementación de los modelos mencionados. Además, se incluye un manual de uso para cada uno de los roles de aplicación desarrollada.

4.2 Construcción modelo físico de datos

4.2.1 Construcción del modelo físico de datos módulo am

En esta sección, se detalla la implementación del modelo físico de datos. Esto se corresponde con el modelo de dominio de **WAINE**.

Como ya se mencionó en el apartado 1, para este modelo se ha hecho uso del lenguaje SQL para la definición de las distintas tablas y vistas relativas al módulo de inventario. Estas tablas y vistas forman parte del esquema *am* de la base de datos.

A continuación, se muestra la implementación de las tablas en base al apartado 3 del documento de diseño del módulo de inventario:

- **UnitType**: define un tipo de unidad (*Unit*). Su código SQL es el siguiente:

```
CREATE TABLE am.UnitType (
    pk serial PRIMARY KEY ,
    descr character varying(80) NOT NULL,
    rem text
);
```

- **Unit**: define una unidad. Su código SQL es el siguiente:

```
CREATE TABLE am.Unit(
    pk serial PRIMARY KEY ,
```

```

un character varying(10) NOT NULL,
descr character varying(80) NOT NULL,
reval character varying(80) NOT NULL,
rem text,
fk integer NOT NULL REFERENCES am.UnitType(pk)
);

```

- **Regex:** define una expresión regular. Su código SQL es el siguiente:

```

CREATE TABLE am.Regex(
    descr varchar(80) NOT NULL,
    code varchar(80) NOT NULL
);

```

- **AssetType:** define un tipo de activo (*Asset*). Su código SQL es el siguiente:

```

CREATE TABLE am.AssetType (
    pk serial PRIMARY KEY ,
    descr varchar(80) NOT NULL,
    rem text
);

```

- **ModelTypeAttr:** define un atributo de tipo modelo para un tipo de activo (*AssetType*) con un tipo de unidad (*UnitType*). Su código SQL es el siguiente:

```

CREATE TABLE am.ModelTypeAttr (
    pk serial PRIMARY KEY ,
    name varchar(80) NOT NULL,
    fktype integer NOT NULL REFERENCES am.AssetType(pk) ON UPDATE CASCADE,
    fkudtype integer NOT NULL REFERENCES am.UnitType(pk) ON UPDATE CASCADE
);

```

- **AssetTypeAttr:** define un atributo de tipo activo para un tipo de activo (*AssetType*) con un tipo de unidad (*UnitType*). Su código SQL es el siguiente:

```

CREATE TABLE am.AssetTypeAttr (
    pk serial PRIMARY KEY ,
    name varchar(80) NOT NULL,
    fktype integer NOT NULL REFERENCES am.AssetType(pk) ON UPDATE CASCADE,
    fkudtype integer NOT NULL REFERENCES am.UnitType(pk) ON UPDATE CASCADE
);

```

- **Mark:** define una marca. Su código SQL es el siguiente:

```

CREATE TABLE am.Mark (
    pk serial PRIMARY KEY ,
    descr varchar(80) NOT NULL,
    rem text
);

```

- **Model:** define un modelo. Su código SQL es el siguiente:

```

CREATE TABLE am.Model (
    pk serial PRIMARY KEY ,
    descr varchar(80) NOT NULL,
    pn varchar(80) NOT NULL,
    rem text,
    fktype integer NOT NULL REFERENCES am.AssetType(pk) ON UPDATE CASCADE,
    img varchar,
    fk integer NOT NULL REFERENCES am.Mark(pk) ON UPDATE CASCADE
);

```

- **ModelAttr:** define un atributo de un modelo (*Model*). Su código SQL es el siguiente:



```
CREATE TABLE am.ModelAttr (
    pk serial NOT NULL,
    val character varying(80),
    fk integer NOT NULL REFERENCES am.Model(pk) ON UPDATE CASCADE ON DELETE CASCADE,
    fkatattr integer NOT NULL REFERENCES am.ModelTypeAttr(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    fkud integer
);
```

- **Status:** define un estado de recurso. Su código SQL es el siguiente:

```
CREATE TABLE am.Status (
    pk serial PRIMARY KEY ,
    descr character varying(80) NOT NULL
);
```

- **Asset:** define un activo. Su código SQL es el siguiente:

```
CREATE TABLE am.Asset (
    pk bigserial PRIMARY KEY,
    cts timestamp DEFAULT Now(),
    lmts timestamp DEFAULT Now(),
    lmuname varchar(20),
    sn character varying(80),
    id character varying(80),
    descr character varying(80),
    rem text,
    cun varchar(80),
    run varchar(80),
    fkasset bigint REFERENCES am.Asset(pk),
    img varchar,
    fkmodel integer NOT NULL REFERENCES am.Model(pk),
    fkstat integer REFERENCES am.Status(pk),
    fkloc integer REFERENCES gti.LL2(pk),
    fkoul integer REFERENCES gti.OUL3(pk),
    UNIQUE(sn),
    UNIQUE(id)
);
```

- **AssetAttr:** define un atributo de un activo (*Asset*). Su código SQL es el siguiente:

```
CREATE TABLE am.AssetAttr (
    pk bigserial NOT NULL,
    val character varying(80),
    fk bigint NOT NULL REFERENCES am.Asset(pk) ON UPDATE CASCADE ON DELETE CASCADE,
    fkatattr integer NOT NULL REFERENCES am.AssetTypeAttr(pk) ON UPDATE CASCADE,
    fkud integer NOT NULL
);
```

- **OutputType:** define un tipo de salida (*Output*). Su código SQL es el siguiente:

```
CREATE TABLE am.OutputType (
    pk serial PRIMARY KEY ,
    descr varchar(80) NOT NULL,
    rem text
);
```

- **Output:** define una salida. Su código SQL es el siguiente:

```
CREATE TABLE am.Output (
    pk serial PRIMARY KEY,
    ots timestamp DEFAULT Now(),
```

```

src integer NOT NULL REFERENCES gti.LL2(pk),
sun varchar(80) NOT NULL,
dst integer REFERENCES gti.LL2(pk),
run varchar(80) NOT NULL,
oun varchar(80) NOT NULL,
fktype integer NOT NULL REFERENCES am.OutputType(pk),
rem text,
descr varchar(80) NOT NULL,
dst_alt varchar(80)
);

```

- **InputType:** define un tipo de entrada (*Input*). Su código SQL es el siguiente:

```

CREATE TABLE am.InputType (
    pk serial PRIMARY KEY ,
    descr varchar(80) NOT NULL,
    rem text
);

```

- **Input:** define una entrada. Su código SQL es el siguiente:

```

CREATE TABLE am.Input (
    pk serial PRIMARY KEY ,
    its timestamp DEFAULT Now(),
    src integer REFERENCES gti.LL2(pk),
    sun varchar(80) NOT NULL,
    dst integer NOT NULL REFERENCES gti.LL2(pk),
    run varchar(80) NOT NULL,
    iun varchar(80) NOT NULL,
    fkout integer REFERENCES am.Output(pk),
    fkpur integer REFERENCES pp.Purchase(pk),
    fktype integer NOT NULL REFERENCES am.InputType(pk),
    descr varchar(80) NOT NULL,
    rem text,
    src_alt varchar(80)
);

```

- **History:** define una traza. Su código SQL es el siguiente:

```

CREATE TABLE am.History (
    pk serial PRIMARY KEY,
    mts TIMESTAMP DEFAULT Now(),
    mun varchar(40) NOT NULL,
    descr text,
    type text,
    fkasset INTEGER NOT NULL REFERENCES am.Asset(pk) ON UPDATE CASCADE
    ON DELETE CASCADE
);

```

- **Entry:** define una anotación. Su código SQL es el siguiente:

```

CREATE TABLE am.Entry (
    pk SERIAL PRIMARY KEY,
    cu VARCHAR(80) NOT NULL,
    cts TIMESTAMP DEFAULT Now(),
    rem TEXT,
    fkasset INTEGER NOT NULL REFERENCES am.Asset(pk) ON UPDATE CASCADE
    ON DELETE CASCADE
);

```



- **REL_Offer_Model:** define la relación entre una oferta (*Offer*) y un modelo (*Model*) que indica los modelos que incluye una determinada oferta. Su código SQL es el siguiente:

```
CREATE TABLE am.REL_Offer_Model (
    pk serial PRIMARY KEY,
    fkmodel integer NOT NULL REFERENCES am.Model(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    fkoffer integer NOT NULL REFERENCES pp.Offer(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    price float NOT NULL,
    num integer NOT NULL,
    UNIQUE(fkmodel,fkoffer)
);
```

- **REL_Purchase_Model:** define la relación entre una adquisición (*Purchase*) y un modelo (*Model*) que indica los modelos que incluye una determinada adquisición. Su código SQL es el siguiente:

```
CREATE TABLE am.REL_Purchase_Model (
    pk serial PRIMARY KEY ,
    fkmodel integer NOT NULL REFERENCES am.Model(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    fkpur integer NOT NULL REFERENCES pp.Purchase(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    price float NOT NULL,
    num integer NOT NULL,
    UNIQUE(fkmodel,fkpur)
);
```

- **REL_Output_Asset:** define la relación entre una salida (*Output*) y un activo (*Asset*) que indica los activos que incluye una determinada salida. Su código SQL es el siguiente:

```
CREATE TABLE am.REL_Output_Asset (
    pk bigserial PRIMARY KEY ,
    fkasset integer NOT NULL REFERENCES am.Asset(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    fkoutput integer NOT NULL REFERENCES am.Output(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    UNIQUE(fkasset,fkoutput)
);
```

- **REL_Input_Asset:** define la relación entre una entrada (*Input*) y un activo (*Asset*) que indica los activos que incluye una determinada entrada. Su código SQL es el siguiente:

```
CREATE TABLE am.REL_Input_Asset (
    pk bigserial PRIMARY KEY ,
    fkasset integer NOT NULL REFERENCES am.Asset(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    fkinput integer NOT NULL REFERENCES am.Input(pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    UNIQUE(fkasset,fkinput)
);
```

- **REL_Gad_Scope:** define la relación entre un grupo de permisos (gestores de compras) y un ámbito (*Scope*). Su código SQL es el siguiente:

```
CREATE TABLE am.REL_Gad_Scope (
    pk BIGSERIAL PRIMARY KEY,
    fkscope INTEGER NOT NULL REFERENCES gti.Scope (pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
```

```
uid INTEGER NOT NULL,
UNIQUE(fkscope,uid)
);
```

- **REL_Gin_Scope:** define la relación entre un grupo de permisos (gestores de inventario) y un ámbito (*Scope*). Su código SQL es el siguiente:

```
CREATE TABLE am.REL_Gin_Scope (
    pk BIGSERIAL PRIMARY KEY,
    fkscope INTEGER NOT NULL REFERENCES gti.Scope (pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    uid INTEGER NOT NULL,
    UNIQUE(fkscope,uid)
);
```

- **REL_Ate_Scope:** define la relación entre un grupo de permisos (técnicos avanzados) y un ámbito (*Scope*). Su código SQL es el siguiente:

```
CREATE TABLE am.REL_Ate_Scope (
    pk BIGSERIAL PRIMARY KEY,
    fkscope INTEGER NOT NULL REFERENCES gti.Scope (pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    uid INTEGER NOT NULL,
    UNIQUE(fkscope,uid)
);
```

- **REL_Tec_Scope:** define la relación entre un grupo de permisos (técnicos) y un ámbito (*Scope*). Su código SQL es el siguiente:

```
CREATE TABLE am.REL_Tec_Scope (
    pk BIGSERIAL PRIMARY KEY,
    fkscope INTEGER NOT NULL REFERENCES gti.Scope (pk) ON UPDATE CASCADE
    ON DELETE CASCADE,
    uid INTEGER NOT NULL,
    UNIQUE(fkscope,uid)
);
```

Además de las tablas, se han creado las siguientes vistas:

- **VIEW_model:** contiene la descripción de un modelo indicando su modelo y marca. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW am.VIEW_model AS
SELECT am.Model.pk,
       am.Model.fk,
       am.Model.fktype,
       am.Mark.descr || ' - ' ||
       am.Model.descr AS descr
FROM
    am.Mark,
    am.Model
WHERE
    am.Model.fk=am.Mark.pk;
```

- **VIEW_fullmodel:** contiene la descripción completa de un modelo indicando su modelo, marca y tipo. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW am.VIEW_fullmodel AS
SELECT DISTINCT am.Model.pk,
               am.Model.fk,
```



```

am.Model.fktype,
am.Mark.descr || '-' ||
am.Model.descr || '-' ||
am.AssetType.descr AS descr
FROM
am.Asset,
am.AssetType,
am.Mark,
am.Model
WHERE
am.Model.fktype=am.AssetType.pk
AND
am.Model.fk=am.Mark.pk
ORDER BY
pk;

```

- **VIEW_asset**: contiene la descripción de un activo indicando su tipo, marca y modelo. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW am.VIEW_asset AS
SELECT am.Asset.pk,
am.Asset.fkmodel || ':' ||
am.AssetType.descr || '-' ||
am.Mark.descr || '-' ||
am.Model.descr AS descr
FROM
am.Asset,
am.AssetType,
am.Mark,
am.Model
WHERE
am.Asset.fkmodel=am.Model.pk
AND
am.Model.fktype=am.AssetType.pk
AND
am.Model.fk=am.Mark.pk
ORDER BY
descr;

```

- **VIEW_asset_2**: contiene la descripción completa de un activo indicando su número de serie, identificador, marca y modelo. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW am.VIEW_asset_2 AS
SELECT am.Asset.pk,
am.Asset.sn || '-' ||
am.Mark.descr || '-' ||
am.Model.descr AS descr,
am.Asset.fkloc
FROM
am.Asset,
am.Mark,
am.Model
WHERE
am.Asset.fkmodel=am.Model.pk
AND
am.Model.fk=am.Mark.pk
AND
am.Asset.sn is not null
AND
am.Asset.id is null

```

```

UNION ALL
SELECT am.Asset.pk,
        am.Asset.sn || '-' ||
        am.Asset.id || '-' ||
        am.Mark.descr || '-' ||
        am.Model.descr AS descr,
        am.Asset.fkloc
FROM
        am.Asset,
        am.Mark,
        am.Model
WHERE
        am.Asset.fkmodel=am.Model.pk
AND
        am.Model.fk=am.Mark.pk
AND
        am.Asset.sn is not null
AND
        am.Asset.id is not null
ORDER BY
        descr;

```

- **VIEW_input_assets:** contiene el número de activos por tipo dada una entrada. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW am.VIEW_input_assets AS
SELECT COUNT(rm.fktype),
        ria.fkinput,
        rm.fktype as pk
FROM
        am.asset as ra,
        am.REL_Input_Asset as ria,
        am.Model as rm
WHERE
        ra.pk=ria.fkasset
AND
        rm.pk=ra.fkmodel
GROUP BY
        ria.fkinput,rm.fktype;

```

- **VIEW_output_assets:** contiene el número de activos por tipo dada una salida. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW am.VIEW_output_assets AS
SELECT COUNT(rm.fktype),
        roa.fkoutput,
        rm.fktype as pk
FROM
        am.asset as ra,
        am.REL_Output_Asset as roa,
        am.Model as rm
WHERE
        ra.pk=roa.fkasset
AND
        rm.pk=ra.fkmodel
GROUP BY
        roa.fkoutput,rm.fktype;

```

- **VIEW_input_report:** contiene la información necesaria para generar el informe de entrada con la descripción de los activos. Su código SQL es el siguiente:



```

CREATE OR REPLACE VIEW am.VIEW_input_report AS
SELECT DISTINCT am.REL_Input_Asset.fkasset as pk,
am.REL_Input_Asset.fkinput,
am.Input.descr,
am.Input.its,
(SELECT descr FROM am.InputType where pk=am.Input.fktype) as input_type,
CASE am.Input.fktype
WHEN 1 THEN (SELECT descr FROM pp.Purchase WHERE pk=am.Input.fkpur)
ELSE (SELECT descr FROM am.Output WHERE pk=am.Input.fkout)
END as rel,
am.Input.sun,
(SELECT descr FROM gti.VIEW_loc WHERE pk=am.Input.src) as src,
am.Input.run,
(SELECT descr FROM gti.VIEW_loc WHERE pk=am.Input.dst) as dst,
am.AssetType.descr as asset_type,
am.Mark.descr as mark,
am.Model.descr as model,
am.Asset.sn,
am.Asset.id
FROM
am.Asset,
am.AssetType,
am.Mark,
am.Model,
am.Input,
am.REL_Input_Asset
WHERE
am.REL_Input_Asset.fkasset=am.Asset.pk
AND
am.REL_Input_Asset.fkinput=am.Input.pk
AND
am.Asset.fkmodel=am.Model.pk
AND
am.Model.fktype=am.AssetType.pk
AND
am.Model.fk=am.Mark.pk
ORDER BY
am.REL_Input_Asset.fkasset;

```

- **VIEW_output_report:** contiene la información necesaria para generar el informe de salida con la descripción de los activos. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW am.VIEW_output_report AS
SELECT DISTINCT am.REL_Output_Asset.fkasset as pk,
am.REL_Output_Asset.fkoutput,
am.Output.descr,
am.Output.ots,
(SELECT descr FROM am.OutputType WHERE pk=am.Output.fktype) as output_type,
am.Output.sun,
(SELECT descr FROM gti.VIEW_loc WHERE pk=am.Output.src) as src,
am.Output.run,
(SELECT descr FROM gti.VIEW_loc WHERE pk=am.Output.dst) as dst,
am.AssetType.descr as asset_type,
am.Mark.descr as mark,
am.Model.descr as model,
am.Asset.sn,
am.Asset.id
FROM
am.Asset,

```

```

am.AssetType,
am.Mark, am.Model,
am.Output,
am.REL_Output_Asset
WHERE
am.REL_Output_Asset.fkasset=am.Asset.pk
AND
am.REL_Output_Asset.fkoutput=am.Output.pk
AND
am.Asset.fkmodel=am.Model.pk
AND
am.Model.fktype=am.AssetType.pk
AND
am.Model.fk=am.Mark.pk
ORDER BY
am.REL_Output_Asset.fkasset;

```

- **VIEW_asset_report:** contiene la información necesaria para generar el informe de activo con la descripción de las trazas. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW am.VIEW_asset_report AS
SELECT DISTINCT am.History.pk,
am.History.fkasset,
am.Asset.sn,
am.Asset.id,
am.AssetType.descr as asset_type,
am.Mark.descr as mark,
am.Model.descr as model,
am.Status.descr as stat,
(SELECT descr FROM gti.VIEW_ou WHERE pk=am.Asset.fkoul) as oul,
am.Asset.cun,
(SELECT descr FROM gti.VIEW_loc WHERE pk=am.Asset.fkloc) as loc,
(SELECT pd FROM am.VIEW_assetpurchase WHERE pk=am.Asset.pk) as pd,
(SELECT wsd FROM am.VIEW_assetpurchase WHERE pk=am.Asset.pk) as wsd,
(SELECT esd FROM am.VIEW_assetpurchase WHERE pk=am.Asset.pk) as esd,
am.History.mts,
am.History.mun,
(SELECT substring(split_part(am.History.descr,'\n',1),1,70)) as history_one,
(SELECT substring(split_part(am.History.descr,'\n',2),1,70)) as history_two,
(SELECT substring(split_part(am.History.descr,'\n',3),1,70)) as history_three

FROM
am.Asset,
am.AssetType,
am.Mark,
am.Model,
am.History,
am.Status
WHERE
am.History.fkasset=am.Asset.pk
AND
am.Asset.fkstat=am.Status.pk
AND
am.Asset.fkmodel=am.Model.pk
AND
am.Model.fktype=am.AssetType.pk
AND
am.Model.fk=am.Mark.pk
ORDER BY
am.History.pk;

```



- **VIEW_typeassetnum:** contiene el número de activos por tipo dado un edificio. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW am.VIEW_typeassetnum AS
```

```
SELECT gti.Site.pk,  
        am.AssetType.descr as assettype,  
        COUNT(am.AssetType.descr) as total
```

```
FROM
```

```
gti.Site,  
gti.LL1,  
gti.LL2,  
am.Asset,  
am.Model,  
am.AssetType
```

```
WHERE
```

```
gti.Site.pk=gti.LL1.fk  
AND  
gti.LL1.pk=gti.LL2.fk  
AND  
am.Asset.fkloc=gti.LL2.pk  
AND  
am.AssetType.pk=am.Model.fktype  
AND  
am.Model.pk=am.Asset.fkmodel
```

```
GROUP BY
```

```
gti.Site.pk,  
am.AssetType.descr
```

```
ORDER BY
```

```
gti.Site.pk;
```

- **VIEW_sitetype:** contiene los tipos de activos para cada edificio. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW am.VIEW_sitetype AS
```

```
SELECT DISTINCT am.AssetType.pk as typepk,  
                gti.Site.pk as sitepk,  
                gti.Scope.descr || ' - ' ||  
                gti.Site.descr || ' - ' ||  
                am.AssetType.descr AS descr
```

```
FROM
```

```
gti.Scope,  
gti.Site,  
gti.LL1,  
gti.LL2,  
am.Asset,  
am.Model,  
am.AssetType
```

```
WHERE
```

```
gti.Scope.pk=gti.Site.fk  
AND  
gti.Site.pk=gti.LL1.fk  
AND  
gti.LL1.pk=gti.LL2.fk  
AND  
am.Asset.fkloc=gti.LL2.pk  
AND  
am.AssetType.pk=am.Model.fktype  
AND  
am.Model.pk=am.Asset.fkmodel;
```

- **VIEW_statusassetnum:** contiene el número de activos por estado del recurso dado un tipo de activo. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW am.VIEW_statusassetnum AS
```

```
SELECT am.AssetType.pk as typepk,  
      gti.Site.pk as sitepk,  
      am.Status.descr as statusasset,  
      COUNT(am.Status.descr) as total
```

```
FROM
```

```
      gti.Site,  
      gti.LL1,  
      gti.LL2,  
      am.Asset,  
      am.Model,  
      am.AssetType,  
      am.Status
```

```
WHERE
```

```
      gti.Site.pk=gti.LL1.fk  
      AND  
      gti.LL1.pk=gti.LL2.fk  
      AND  
      am.Asset.fkloc=gti.LL2.pk  
      AND  
      am.AssetType.pk=am.Model.fktype  
      AND  
      am.Model.pk=am.Asset.fkmodel  
      AND  
      am.Asset.fkstat=am.Status.pk
```

```
GROUP BY
```

```
      am.AssetType.pk,  
      gti.Site.pk,  
      am.Status.descr
```

```
ORDER BY
```

```
      am.AssetType.pk;
```

- **VIEW_markassetnum:** contiene el número de activos por marca dado un tipo de activo. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW am.VIEW_markassetnum AS
```

```
SELECT am.AssetType.pk as typepk,  
      gti.Site.pk as sitepk,  
      am.VIEW_model.descr as markasset,  
      COUNT(am.VIEW_model.descr) as total
```

```
FROM
```

```
      gti.Site,  
      gti.LL1,  
      gti.LL2,  
      am.Asset,  
      am.Model,  
      am.AssetType,  
      am.VIEW_model
```

```
WHERE
```

```
      gti.Site.pk=gti.LL1.fk  
      AND  
      gti.LL1.pk=gti.LL2.fk  
      AND  
      am.Asset.fkloc=gti.LL2.pk  
      AND  
      am.AssetType.pk=am.Model.fktype
```



```

AND
am.Model.pk=am.Asset.fkmodel
AND
am.VIEW_model.pk=am.Model.pk
GROUP BY
am.AssetType.pk,
gti.Site.pk,
am.VIEW_model.descr
ORDER BY
am.AssetType.pk;

```

- **VIEW_assetunit:** contiene las unidades dado un atributo de activo.

```

CREATE OR REPLACE VIEW am.VIEW_assetunit AS
SELECT am.Unit.pk,
       am.Unit.un,
       am.AssetAttr.pk as attrpk
FROM
  am.Unit,
  am.AssetTypeAttr,
  am.AssetAttr
WHERE
  am.Unit.fk=am.AssetTypeAttr.fkudtype
AND
  am.AssetTypeAttr.pk=am.AssetAttr.fkatattr
ORDER BY
  am.Unit.pk;

```

- **VIEW_assetmodelattr:** contiene los atributos de tipo activo de un activo. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW am.VIEW_assetmodelattr AS
SELECT am.ModelAttr.pk,
       am.ModelAttr.val,
       am.Asset.pk AS fk,
       am.ModelAttr.fkatattr,
       am.ModelAttr.fkud
FROM
  am.ModelAttr , am.Asset
WHERE
  am.ModelAttr.fk=am.Asset.fkmodel;

```

- **VIEW_assetpurchase:** contiene la descripción de una adquisición para un activo. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW am.VIEW_assetpurchase AS
SELECT am.REL_Input_Asset.fkasset as pk,
       pp.Purchase.pd,
       pp.Purchase.wsd,
       pp.Purchase.esd
FROM
  am.REL_Input_Asset,
  am.Input,
  pp.Purchase
WHERE
  am.REL_Input_Asset.fkinput=am.Input.pk
AND
  am.Input.fktype=1
AND

```

```
am.Input.fkpur=pp.Purchase.pk
ORDER BY
am.REL_Input_Asset.fkasset;
```

4.2.2 Construcción del modelo físico de datos módulo gti

En esta sección, se detalla la implementación del modelo físico de datos. Esto se corresponde con el modelo de dominio de **WAINE**.

Como ya se mencionó en el apartado 1, para este modelo se ha hecho uso del lenguaje SQL para la definición de las distintas tablas y vistas relativas al módulo común. Estas tablas y vistas forman parte del esquema *am* de la base de datos.

A continuación, se muestra la implementación de las tablas en base al apartado 3 del documento de diseño del módulo común:

- **OUL1:** define una unidad organizativa de nivel 1. Su código SQL es el siguiente:

```
CREATE TABLE gti.OUL1 (
  pk serial PRIMARY KEY ,
  descr character varying(80) NOT NULL,
  rem text
);
```

- **OUL2:** define una unidad organizativa de nivel 2. Su código SQL es el siguiente:

```
CREATE TABLE gti.OUL2 (
  pk serial PRIMARY KEY ,
  descr character varying(80) NOT NULL,
  rem text,
  fk integer NOT NULL REFERENCES gti.OUL1(pk) ON UPDATE CASCADE ON DELETE CASCADE
);
```

- **OUL3:** define una unidad organizativa de nivel 3. Su código SQL es el siguiente:

```
CREATE TABLE gti.OUL3 (
  pk serial PRIMARY KEY ,
  descr character varying(80) NOT NULL,
  rem text,
  active integer,
  fk integer NOT NULL REFERENCES gti.OUL2(pk) ON UPDATE CASCADE ON DELETE CASCADE
);
```

- **Site:** define un edificio. Su código SQL es el siguiente:

```
CREATE TABLE gti.Site (
  pk serial PRIMARY KEY ,
  descr character varying(80) NOT NULL,
  rem text,
  fk integer NOT NULL REFERENCES gti.Scope(pk) ON UPDATE CASCADE ON DELETE CASCADE
);
```

- **LL1:** define una localización de nivel 1. Su código SQL es el siguiente:

```
CREATE TABLE gti.LL1 (
  pk serial PRIMARY KEY ,
  descr character varying(80) NOT NULL,
  rem text,
  fk integer NOT NULL REFERENCES gti.Site(pk) ON UPDATE CASCADE ON DELETE CASCADE
);
```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

- **LL2:** define una localización de nivel 2. Su código SQL es el siguiente:

```
CREATE TABLE gti.LL2 (
  pk serial PRIMARY KEY ,
  descr character varying(80) NOT NULL,
  rem text,
  store integer,
  active integer,
  fk integer NOT NULL REFERENCES gti.LL1(pk) ON UPDATE CASCADE ON DELETE CASCADE
);
```

- **Scope:** define un ámbito. Su código SQL es el siguiente:

```
CREATE TABLE gti.Scope (
  pk BIGSERIAL PRIMARY KEY,
  descr VARCHAR(80) NOT NULL,
  rem TEXT
);
```

Además de las tablas, se han creado las siguientes vistas:

- **VIEW_OU:** contiene todos los niveles de unidades organizativas. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW gti.VIEW_OU AS
SELECT gti.OUL3.pk,
       gti.OUL1.descr || ' - ' ||
       gti.OUL2.descr || ' - ' ||
       gti.OUL3.descr AS descr
FROM
  gti.OUL1,
  gti.OUL2,
  gti.OUL3
WHERE
  gti.OUL3.fk=gti.OUL2.pk
AND
  gti.OUL2.fk=gti.OUL1.pk;
AND
  gti.OUL3.active=1;
```

- **VIEW_Loc:** contiene todos los niveles de localizaciones, incluidos el ámbito y el edificio. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW gti.VIEW_Loc AS
SELECT gti.LL2.pk,
       gti.Scope.descr || ' - ' ||
       gti.Site.descr || ' - ' ||
       gti.LL1.descr || ' - ' ||
       gti.LL2.descr AS descr,
       gti.Scope.pk as scopepk
FROM
  gti.Scope,
  gti.Site,
  gti.LL1,
  gti.LL2
WHERE
  gti.LL2.fk=gti.LL1.pk
AND
  gti.LL1.fk=gti.Site.pk
AND
  gti.Site.fk=gti.Scope.pk;
```

```
AND
gti.LL2.active=1;
```

- **VIEW_Loc_woscope:** contiene todos los niveles de localizaciones, excepto el ámbito. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW gti.VIEW_Loc_woscope AS
SELECT gti.LL2.pk,
       gti.Site.fk,
       gti.Site.descr || ' - ' ||
       gti.LL1.descr || ' - ' ||
       gti.LL2.descr AS descr
FROM
       gti.Site,
       gti.LL1,
       gti.LL2
WHERE
       gti.LL2.fk=gti.LL1.pk
AND
       gti.LL1.fk=gti.Site.pk;
AND
gti.LL2.active=1;
```

- **VIEW_Site:** contiene la descripción de todos los edificios, junto a su ámbito. Su código SQL es el siguiente:

```
CREATE OR REPLACE VIEW gti.VIEW_Site AS
SELECT gti.Site.pk,
       gti.Scope.descr || ' - ' ||
       gti.Site.descr AS descr,
       gti.Site.fk
FROM
       gti.Scope,
       gti.Site
WHERE
       gti.Site.fk=gti.Scope.pk;
```

4.2.3 Construcción del modelo físico de datos módulo pp

En esta sección, se detalla la implementación del modelo físico de datos. Esto se corresponde con el modelo de dominio de **WAINE**.

Como ya se mencionó en el apartado 1, para este modelo se ha hecho uso del lenguaje SQL para la definición de las distintas tablas y vistas relativas al módulo de adquisición. Estas tablas y vistas forman parte del esquema *pp* de la base de datos.

A continuación, se muestra la implementación de las tablas en base al apartado 3 del documento de diseño del módulo de adquisición:

- **Provider:** define un proveedor. Su código SQL es el siguiente:

```
CREATE TABLE pp.Provider (
  pk serial PRIMARY KEY,
  descr character varying(80) NOT NULL,
  name character varying(80),
  tel character varying(80),
  email character varying(80),
  addr character varying(80),
  rem text
```



```
);
```

- **OfferReq:** define una petición de oferta. Su código SQL es el siguiente:

```
CREATE TABLE pp.OfferReq (
  pk serial PRIMARY KEY,
  descr character varying(80) NOT NULL,
  rem text,
  type character varying(80),
  fkscope integer NOT NULL REFERENCES gti.Scope(pk)
);
```

- **Offer:** define una oferta. Su código SQL es el siguiente:

```
CREATE TABLE pp.Offer (
  pk serial PRIMARY KEY,
  od date NOT NULL,
  descr character varying(80) NOT NULL,
  rem text,
  total float, -- Estimation of the total price of the offer
  fkprov integer NOT NULL REFERENCES pp.Provider(pk),
  fkreq integer NOT NULL REFERENCES pp.OfferReq(pk)
);
```

- **Purchase:** define una adquisición. Su código SQL es el siguiente:

```
CREATE TABLE pp.Purchase (
  pk serial PRIMARY KEY,
  descr character varying(80) NOT NULL,
  cn character varying(80), -- Contract number
  pd date,
  wsd date,
  esd date,
  rem text,
  total float, -- Estimation of the total price of the purchase
  fkoffer integer NOT NULL REFERENCES pp.Offer(pk)
);
```

- **REL_Provider_Person:** define la relación entre un proveedor (*Provider*) y un usuario (*percon_person*). Su código SQL es el siguiente:

```
CREATE TABLE pp.REL_Provider_Person (
  pk BIGSERIAL PRIMARY KEY,
  fkprov INTEGER NOT NULL REFERENCES pp.Provider (pk) ON UPDATE CASCADE ON DELETE CASCADE,
  fkperson INTEGER NOT NULL REFERENCES percon_person (pk) ON UPDATE CASCADE ON DELETE CASCADE,
  UNIQUE(fkprov,fkperson)
);
```

Además de las tablas, se han creado las siguientes vistas:

- **VIEW_offer:** contiene la descripción de una oferta indicando todos sus atributos, además del proveedor. Su código SQL es el siguiente:

```
CREATE VIEW pp.VIEW_offer AS
SELECT pp.Offer.pk,
  pp.Provider.descr || '-' ||
  pp.Offer.od AS descr,
  pp.Offer.od,
  pp.Offer.rem,
```

```

pp.Offer.total,
pp.Offer.fkprov,
pp.Offer.fkreq
FROM
pp.Offer,
pp.Provider
WHERE
pp.Offer.fkprov=pp.Provider.pk
ORDER BY
descr;

```

- **VIEW_person:** contiene la descripción completa de un usuario, indicando nombre y apellidos. Su código SQL es el siguiente:

```

CREATE OR REPLACE VIEW pp.VIEW_person AS
SELECT pk,
surname || ', ' ||
name AS descr
FROM
percon_person;

```

4.3 Construcción de los módulos del sistema

4.3.1 Construcción de los módulos del sistema módulo am

En esta sección se incluye la implementación de los eventos y funciones necesarias para el módulo de inventario de GTI.

4.3.1.1 Construcción de eventos

En base al apartado 4.1 del Documento de Diseño, se diferencian la implementación de los eventos en el lenguaje SQL, como en el lenguaje ASL.

4.3.1.1.1 Construcción de eventos en SQL

Los eventos implementados en el lenguaje SQL son los siguientes:

- **Evento TRI_ModelTypeAttr_after:** ocurre tras una inserción en la tabla *ModelTypeAttr*. Este evento lanza la función **PROC_ModelTypeAttr_after()**. Su código SQL es el siguiente:

```

CREATE TRIGGER TRI_ModelTypeAttr_after AFTER INSERT ON am.ModelTypeAttr FOR EACH ROW EXECUTE
PROCEDURE am.PROC_ModelTypeAttr_after();

```

- **Evento TRI_AssetTypeAttr_after:** ocurre tras una inserción en la tabla *AssetTypeAttr*. Este evento lanza la función **PROC_AssetTypeAttr_after()**. Su código SQL es el siguiente:

```

CREATE TRIGGER TRI_AssetTypeAttr_after AFTER INSERT ON am.AssetTypeAttr FOR EACH ROW EXECUTE
PROCEDURE am.PROC_AssetTypeAttr_after();

```

- **Evento TRI_Model_after:** ocurre tras una inserción en la tabla *Model*. Este evento lanza la función **PROC_Model_after()**. Su código SQL es el siguiente:

```

CREATE TRIGGER TRI_Model_after AFTER INSERT ON am.Model FOR EACH ROW EXECUTE PROCEDURE
am.PROC_Model_after();

```

- **Evento TRI_Asset_after:** ocurre tras una inserción en la tabla *Asset*. Este evento lanza la función **PROC_Asset_after()**. Su código SQL es el siguiente:

```

CREATE TRIGGER TRI_Asset_after AFTER INSERT ON am.Asset FOR EACH ROW EXECUTE PROCEDURE
am.PROC_Asset_after();

```



- **Evento TRI_Input_after:** ocurre tras una inserción en la tabla *Input*. Este evento lanza la función **PROC_Input_after()**. Su código SQL es el siguiente:

```
CREATE TRIGGER TRI_Input_after AFTER INSERT ON am.Input FOR EACH ROW EXECUTE PROCEDURE
am.PROC_Input_after();
```

- **Evento TRI_Asset_update:** ocurre antes de una actualización en la tabla *Asset*. Este evento lanza la función **PROC_Asset_update()**. Su código SQL es el siguiente:

```
CREATE TRIGGER TRI_Asset_update BEFORE UPDATE ON am.Asset FOR EACH ROW EXECUTE PROCEDURE
am.PROC_Asset_update();
```

- **Evento TRI_Asset_history:** ocurre tras una actualización en la tabla *Asset*. Este evento lanza la función **PROC_Asset_history()**. Su código SQL es el siguiente:

```
CREATE TRIGGER TRI_Asset_history AFTER UPDATE ON am.Asset FOR EACH ROW EXECUTE PROCEDURE
am.PROC_Asset_history();
```

- **Evento TRI_Purchase_after:** ocurre tras una inserción en la tabla *Purchase*. Este evento lanza la función **PROC_Purchase_after()**. Su código SQL es el siguiente:

```
CREATE TRIGGER TRI_Purchase_after AFTER INSERT ON pp.Purchase FOR EACH ROW EXECUTE PROCEDURE
am.PROC_Purchase_after();
```

4.3.1.1.2 ¹Construcción de eventos en ASL

Los eventos implementados en el lenguaje ASL son los siguientes:

- **Evento “Eliminación de un tipo de entrada predefinida”:** asegura que no se elimine un tipo de entrada del tipo “Adquisición”, “Interna” o “Devolución”. Su código ASL es el siguiente:

```
<action id="gti.am.actinptype" type="php">
  <code>
    return %values[0]>3;
  </code>
  <msg>
    es=Eliminacion abortada. Este tipo de entrada esta predefinido |en=Elimination abort. This input
    type is predefined
  </msg>
</action>
```

- **Evento “Eliminación de un tipo de salida predefinida”:** asegura que no se elimine un tipo de salida del tipo “Interna”, “Préstamo” o “Baja”. Su código ASL es el siguiente:

```
<action id="gti.am.actouttype" type="php">
  <code>
    return %values[0]>3;
  </code>
  <msg>
    es=Eliminacion abortada. Este tipo de salida esta predefinido |en=Elimination abort. This output
    type is predefined
  </msg>
```

```
</action>
```

- **Evento “Eliminación de un estado de los recursos”**: asegura que no se elimine un estado de los recursos del tipo “Activo” o “Baja”. Su código ASL es el siguiente:

```
<action id="gti.am.actstatus" type="php">
  <code>
    return %values[0]>2;
  </code>
  <msg>
    es=Eliminacion abortada. Este estado esta predefinido |en=Elimination abort. This status is
    predefined
  </msg>
</action>
```

- **Evento “Asignación de permisos de gestor de adquisiciones”**: asegura que un mismo usuario no aparezca por duplicado en el grupo de gestores de adquisiciones. Su código ASL es el siguiente:

```
<action id="gti.am.actgad" type="php">
  <code>
    return !waine_DsGetNumRows("am.REL_Gad_Scope","uid=%values[1]' and
    fkscope=%values[2]");
  </code>
  <msg>
    es=Insercion abortada. Ese usuario ya existe en este ambito |en=Insert abort. This user already
    exists in this scope
  </msg>
</action>
```

- **Evento “Asignación de permisos de gestor de inventario”**: asegura que un mismo usuario no aparezca por duplicado en el grupo de gestores de inventario. Su código ASL es el siguiente:

```
<action id="gti.am.actgin" type="php">
  <code>
    return !waine_DsGetNumRows("am.REL_Gin_Scope","uid=%values[1]' and
    fkscope=%values[2]");
  </code>
  <msg>
    es=Insercion abortada. Ese usuario ya existe en este ambito |en=Insert abort. This user already
    exists in this scope
  </msg>
</action>
```

- **Evento “Asignación de permisos de técnico avanzado”**: asegura que un mismo usuario no aparezca por duplicado en el grupo de técnicos avanzados. Su código ASL es el siguiente:

```
<action id="gti.am.actate" type="php">
  <code>
    return !waine_DsGetNumRows("am.REL_Ate_Scope","uid=%values[1]' and
    fkscope=%values[2]");
  </code>
  <msg>
    es=Insercion abortada. Ese usuario ya existe en este ambito |en=Insert abort. This user already
    exists in this scope</msg>
</action>
```



- **Evento “Asignación de permisos de técnico”**: asegura que un mismo usuario no aparezca por duplicado en el grupo de técnicos. Su código ASL es el siguiente:

```
<action id="gti.am.acttec" type="php">
  <code>
    return !waine_DsGetNumRows("am.REL_Tec_Scope","uid='%values[1]' and
    fkscope='%values[2]'");
  </code>
  <msg>
    es=Insercion abortada. Ese usuario ya existe en este ambito |en=Insert abort. This user already
    exists in this scope
  </msg>
</action>
```

- **Evento “Inserción de un nuevo activo en una entrada”**: se encarga de llamar a la función **FUNC_Asset_aux(int,int,vchar)**. Su código ASL es el siguiente:

```
<action id="gti.am.actinput" type="source">
  <code>
    SELECT am.FUNC_Asset_aux(%values[0],%filtervalue, "%username");
  </code>
</action>
```

- **Evento “Inserción de un activo en una salida”**: asegura que un mismo activo no aparezca por duplicado para una misma salida. Su código ASL es el siguiente:

```
<events>
  <beforeinsert>
    <action type="php">
      <code>
        return !waine_DsGetNumRows("am.REL_Output_Asset","fkasset='%values[1]' and
        fkoutput='%values[2]'");
      </code>
      <msg>
        es=Insercion abortada. Este activo ya existe para esta salida |en=Insert aborted. Asset al
        ready exists for this output
      </msg>
    </action>
  </beforeinsert>
</events>
```

- **Evento “Actualización de un activo”**: actualiza la traza del activo que se modifica con el usuario actual como usuario. Su código ASL es el siguiente:

```
<events>
  <afterupdate>
    <action type="source">
      <code>
        UPDATE am.History SET mun="%username" WHERE fkasset=%pk and pk=(SELECT MAX(pk)
        FROM am.History);
      </code>
    </action>
  </afterupdate>
</events>
```

- **Evento “Actualización de un número de serie en un activo”:** comprueba que el número de serie esté completo para un determinado activo antes de crearlo o de actualizarlo. Su código ASL es el siguiente:

```
<action id="gti.am.actns" type="php">
  <code>return "%values[2]"!="";</code>
  <msg>es=Este activo requiere un numero de serie | en=This asset requires a serial number</msg>
</action>
```

4.3.1.2 Construcción de funciones

Este apartado incluye las funciones que se hacen uso en los eventos descritos en el apartado anterior.

En base al apartado 4.2 del Documento de Diseño, se diferencian la implementación de las siguientes funciones en el código SQL. Estas funciones son las siguientes:

- **Función PROC_ModelTypeAttr_after():** implementa las funcionalidades descritas en el apartado 4.2.1 del Documento de Diseño. Su código SQL es el siguiente:

```
CREATE OR REPLACE FUNCTION am.PROC_ModelTypeAttr_after() RETURNS TRIGGER
AS '
BEGIN
  INSERT INTO am.ModelAttr(fk,fkatattr,fkud)
    SELECT pk, NEW.pk, 1 from am.Model where fktype=NEW.fktype;
  RETURN NEW;
END;
' LANGUAGE 'plpgsql';
```

- **Función PROC_AssetTypeAttr_after():** implementa las funcionalidades descritas en el apartado 4.2.2 del Documento de Diseño. Su código SQL es el siguiente:

```
CREATE OR REPLACE FUNCTION am.PROC_AssetTypeAttr_after() RETURNS TRIGGER
AS '
BEGIN
  INSERT INTO am.AssetTypeAttr(fk,fkatattr,fkud)
    SELECT pk, NEW.pk, 1 from am.AssetType where fkmodel IN (Select pk from am.Model where
fktype=NEW.fktype);
  RETURN NEW;
END;
' LANGUAGE 'plpgsql';
```

- **Función PROC_Model_after():** implementa las funcionalidades descritas en el apartado 4.2.3 del Documento de Diseño. Su código SQL es el siguiente:

```
CREATE OR REPLACE FUNCTION am.PROC_Model_after() RETURNS TRIGGER
AS '
BEGIN
  INSERT INTO am.ModelAttr(fk,fkatattr,fkud)
    SELECT NEW.pk, pk, 1 from am.ModelTypeAttr where fktype=NEW.fktype;
  RETURN NEW;
END;
' LANGUAGE 'plpgsql';
```

- **Función PROC_Asset_after():** implementa las funcionalidades descritas en el apartado 4.2.4 del Documento de Diseño. Su código SQL es el siguiente:

```
CREATE OR REPLACE FUNCTION am.PROC_Asset_after() RETURNS TRIGGER
AS '

```



```
BEGIN
  INSERT INTO am.AssetAttr(fk,fkatattr,fkud)
    SELECT NEW.pk, pk, 1 from am.AssetTypeAttr WHERE fktype IN (SELECT fktype from am.Model WHERE
pk=NEW.fkmodel);
  RETURN NEW;
END;
' LANGUAGE 'plpgsql';
```

- **Función PROC_Input_after():** implementa las funcionalidades descritas en el apartado 4.2.5 del Documento de Diseño. Su código SQL es el siguiente:

```
CREATE OR REPLACE FUNCTION am.PROC_Input_after() RETURNS trigger AS '
DECLARE
  adquisicion INTEGER := 1;
  r am.REL_Purchase_Model%rowtype;
  sr am.REL_Output_Asset%rowtype;
BEGIN
  IF (NEW.fktype = adquisicion) THEN
    FOR r IN SELECT * from am.REL_purchase_model where fkpur=NEW.fkpur
    LOOP
      PERFORM am.FUNC_Input_aux(r.fkmodel, r.num, NEW.pk, NEW.dst, NEW.run, NEW.iun);
    END LOOP;
    UPDATE am.Input SET fkout=null where pk=NEW.pk;
  ELSE
    FOR sr IN SELECT * from am.REL_Output_Asset where fkoutput=NEW.fkout
    LOOP
      INSERT INTO am.REL_Input_Asset(fkasset,fkinput) VALUES (sr.fkasset, NEW.pk);
      UPDATE am.Asset SET fkloc=NEW.dst,run=NEW.run,fkoul=null WHERE pk=sr.fkasset;
    END LOOP;
    UPDATE am.Input SET fkpur=null where pk=NEW.pk;
  END IF;

  RETURN NEW;
END;
' LANGUAGE 'plpgsql';
```

- **Función FUNC_Input_aux(int,int,int,int,varchar,varchar):** implementa las funcionalidades descritas en el apartado 4.2.6 del Documento de Diseño. Su código SQL es el siguiente:

```
CREATE OR REPLACE FUNCTION am.FUNC_Input_aux(model int,num int,input int,loc int,resp varchar,usuario
varchar) RETURNS boolean AS '
DECLARE
  stat INTEGER := 1;
  count INTEGER := 1;
  time TIMESTAMP := current_timestamp;
BEGIN
  WHILE count <= num
  LOOP
    INSERT INTO am.Asset (cts,fkmodel,fkstat,fkloc,cun,run,fkoul) VALUES (time, model, stat, loc,
resp, resp, null);
    INSERT INTO am.History(mts,mun,type,descr,fkasset) VALUES (current_timestamp(0),usuario,"Inser-
cion",
      "Modelo: " || (SELECT descr FROM am.VIEW_model WHERE pk=model) || E"\n" ||
      "SN: " || "" || E"\n" ||
      "ID: " || "" || E"\n" ||
      "Responsable: " || resp || E"\n" ||
```

```

        "Localizacion: " || (SELECT descr FROM gti.VIEW_loc WHERE pk=loc), (SELECT MAX(pk) FROM
am.Asset));
        count := count + 1;
    END LOOP;
    INSERT INTO am.REL_Input_Asset(fkasset,fkinput) SELECT pk,input from am.Asset where
fkmodel=model and cts=time;

    RETURN true;
END;
' LANGUAGE 'plpgsql'

```

- **Función FUNC_Asset_aux(int,int,varchar):** implementa las funcionalidades descritas en el apartado 4.2.7 del Documento de Diseño. Su código SQL es el siguiente:

```

CREATE OR REPLACE FUNCTION am.FUNC_Asset_aux(asset int,input int,resp_insert varchar) RETURNS bool-
ean AS '
DECLARE
    r_input    am.Input%rowtype;
    r_asset    am.Asset%rowtype;
BEGIN
    SELECT * INTO r_input from am.Input where pk=input;

    INSERT INTO am.REL_Input_Asset(fkasset,fkinput) VALUES (asset,input);
    UPDATE am.Asset SET fkloc=r_input.dst,cun=r_input.run,run=r_input.run,fkstat=1,fkoul=null WHERE
pk=asset;

    SELECT * INTO r_asset from am.Asset where pk=asset;

    DELETE FROM am.History where fkasset=asset;
    INSERT INTO am.History(mts,mun,type,descr,fkasset) VALUES (current_timestamp(0),resp_insert,"Inser-
cion",
        "Modelo: " || (SELECT descr FROM am.VIEW_model WHERE pk=r_asset.fkmodel) || E"\n" ||
        "SN: " || COALESCE(r_asset.sn, "") || E"\n" ||
        "ID: " || COALESCE(r_asset.id, "") || E"\n" ||
        "Responsable: " || r_asset.run || E"\n" ||
        "Localizacion: " || (SELECT descr FROM gti.VIEW_loc WHERE pk=r_asset.fkloc), r_asset.pk);

    RETURN true;
END;
' LANGUAGE 'plpgsql'

```

- **Función PROC_Asset_update():** implementa las funcionalidades descritas en el apartado 4.2.8 del Documento de Diseño. Su código SQL es el siguiente:

```

CREATE OR REPLACE FUNCTION am.PROC_Asset_update() RETURNS trigger AS '
BEGIN
    NEW.lmts = current_timestamp;

    RETURN NEW;
END;
' LANGUAGE 'plpgsql'

```

- **Función PROC_Asset_history():** implementa las funcionalidades descritas en el apartado 4.2.9 del Documento de Diseño. Su código SQL es el siguiente:

```

CREATE OR REPLACE FUNCTION am.PROC_Asset_history() RETURNS trigger AS '
DECLARE
    mod    VARCHAR;

```



```

BEGIN
  IF (NEW.sn != OLD.sn OR (NEW.sn is null AND OLD.sn is not null) OR (NEW.sn is not null AND OLD.sn is
null)) THEN
    mod := "SN: " || NEW.sn;
  END IF;
  IF (NEW.id != OLD.id OR (NEW.id is null AND OLD.id is not null) OR (NEW.id is not null AND OLD.id is null))
THEN
  IF (mod is null) THEN
    mod := "ID: " || COALESCE(NEW.id, "");
  ELSE
    mod := mod || E"\n" || "ID: " || COALESCE(NEW.id, "");
  END IF;
  END IF;
  IF (NEW.fkstat != OLD.fkstat OR (NEW.fkstat is null AND OLD.fkstat is not null) OR (NEW.fkstat is not null
AND OLD.fkstat is null)) THEN
  IF (mod is null) THEN
    mod := "Estado: " || (SELECT descr FROM am.Status WHERE pk=NEW.fkstat);
  ELSE
    mod := mod || E"\n" || "Estado: " || (SELECT descr FROM am.Status WHERE pk=NEW.fkstat);
  END IF;
  END IF;
  IF (NEW.fkmodel != OLD.fkmodel OR (NEW.fkmodel is null AND OLD.fkmodel is not null) OR
(NEW.fkmodel is not null AND OLD.fkmodel is null)) THEN
  IF (mod is null) THEN
    mod := "Modelo: " || (SELECT descr FROM am.VIEW_model WHERE pk=NEW.fkmodel);
  ELSE
    mod := mod || E"\n" || "Modelo: " || (SELECT descr FROM am.VIEW_model WHERE
pk=NEW.fkmodel);
  END IF;
  END IF;
  IF (NEW.fkoul != OLD.fkoul OR (NEW.fkoul is null AND OLD.fkoul is not null) OR (NEW.fkoul is not null
AND OLD.fkoul is null)) THEN
  IF (mod is null) THEN
    mod := "OU: " || COALESCE((SELECT descr FROM gti.VIEW_ou WHERE pk=NEW.fkoul), "");
  ELSE
    mod := mod || E"\n" || "OU: " || COALESCE((SELECT descr FROM gti.VIEW_ou WHERE
pk=NEW.fkoul), "");
  END IF;
  END IF;
  IF (NEW.fkloc != OLD.fkloc OR (NEW.fkloc is null AND OLD.fkloc is not null) OR (NEW.fkloc is not null AND
OLD.fkloc is null)) THEN
  IF (mod is null) THEN
    mod := "Localizacion: " || (SELECT descr FROM gti.VIEW_loc WHERE pk=NEW.fkloc);
  ELSE
    mod := mod || E"\n" || "Localizacion: " || (SELECT descr FROM gti.VIEW_loc WHERE
pk=NEW.fkloc);
  END IF;
  END IF;
  IF (NEW.run != OLD.run OR (NEW.run is null AND OLD.run is not null) OR (NEW.run is not null AND
OLD.run is null)) THEN
  IF (mod is null) THEN
    mod := "Responsable: " || NEW.run;
  ELSE
    mod := mod || E"\n" || "Responsable: " || NEW.run;
  END IF;
  END IF;
  IF (NEW.fkasset != OLD.fkasset OR (NEW.fkasset is null AND OLD.fkasset is not null) OR (NEW.fkasset is
not null AND OLD.fkasset is null)) THEN
  IF (mod is null) THEN

```

```

        mod := "Activo padre: " || COALESCE((SELECT sn FROM am.asset WHERE pk=NEW.fkasset), "");
    ELSE
        mod := mod || E"\n" || "Activo padre: " || COALESCE((SELECT sn FROM am.asset WHERE
pk=NEW.fkasset), "");
    END IF;
    END IF;
    IF (NEW.descr != OLD.descr OR (NEW.descr is null AND OLD.descr is not null) OR (NEW.descr is not null
AND OLD.descr is null)) THEN
        IF (mod is null) THEN
            mod := "Descripcion: " || NEW.descr;
        ELSE
            mod := mod || E"\n" || "Descripcion: " || NEW.descr;
        END IF;
    END IF;
    IF (NEW.rem != OLD.rem OR (NEW.rem is null AND OLD.rem is not null) OR (NEW.rem is not null AND
OLD.rem is null)) THEN
        IF (mod is null) THEN
            mod := "Notas: " || COALESCE(NEW.rem, "");
        ELSE
            mod := mod || E"\n" || "Notas: " || COALESCE(NEW.rem, "");
        END IF;
    END IF;
    IF (NEW.lmuname != OLD.lmuname OR (NEW.lmuname is null AND OLD.lmuname is not null) OR
(NEW.lmuname is not null AND OLD.lmuname is null)) THEN
        IF (mod is null) THEN
            mod := "Nombre equipo: " || NEW.lmuname;
        ELSE
            mod := mod || E"\n" || "Nombre equipo: " || NEW.lmuname;
        END IF;
    END IF;

    IF (mod is not null) THEN
        INSERT INTO am.History(mts,mun,type,descr,fkasset) VALUES (current_timestamp(0), "Unknown ",
"Modificacion", mod, OLD.pk);
    END IF;

    RETURN NEW;
END;
' LANGUAGE 'plpgsql'

```

- **Función PROC_Purchase_after():** implementa las funcionalidades descritas en el apartado 4.2.10 del Documento de Diseño. Su código SQL es el siguiente:

```

CREATE OR REPLACE FUNCTION am.PROC_Purchase_after() RETURNS trigger AS '
DECLARE
    r am.REL_Offer_Model%rowtype;
BEGIN
    FOR r IN SELECT * from am.REL_Offer_Model where fkoffer=NEW.fkoffer
    LOOP
        INSERT INTO am.REL_Purchase_Model(fkmodel,fkpur,price,num) VALUES
(r.fkmodel,NEW.pk,r.price,r.num);
    END LOOP;

    UPDATE pp.Purchase SET total=(SELECT pp.Offer.total FROM pp.Offer where pp.Offer.pk=NEW.fkoffer)
where pk=NEW.pk;
    RETURN NEW;
END;
' LANGUAGE 'plpgsql'

```



4.3.2 Construcción de los módulos del sistema módulo pp

En esta sección se incluye la implementación de los eventos y funciones necesarias para el módulo de inventario de GTI.

4.3.2.1 Construcción de eventos

En base al apartado 4.1 del Documento de Diseño, se diferencian la implementación de los eventos en el lenguaje ASL.

4.3.2.1.1 Construcción de eventos en ASL

Los eventos implementados en el lenguaje ASL son los siguientes:

- **Evento “Inserción de un nuevo contacto para un proveedor”:** asegura que un mismo contacto no aparezca por duplicado para un mismo proveedor. Su código ASL es el siguiente:

```
<action id="gti.pp.actprovmanag" type="php">
  <code>
    return !waine_DsGetNumRows("pp.REL_Provider_Person","fkprov='%values[1]' and
    fkperson='%values[2]'");
  </code>
  <msg>
    es=Insercion abortada. Este usuario ya existe en este proveedor|en=Insert abort. This user
    already exists in this provider
  </msg>
</action>
```

- **Evento “Inserción de un nuevo modelo en una oferta”:** asegura que un mismo modelo no aparezca por duplicado para una misma oferta. Su código ASL es el siguiente:

```
<events>
  <beforeinsert>
    <action type="php">
      <code>
        return !waine_DsGetNumRows("am.REL_Offer_Model","fkmodel='%values[1]' and
        fkoffer='%values[2]'");
      </code>
      <msg>
        es=Insercion abortada. Esa modelo ya existe en esta oferta|en=Insert aborted. Model al
        ready exists in this offer
      </msg>
    </action>
  </beforeinsert>
</events>
```

- **Evento “Inserción de un nuevo modelo en una compra”:** asegura que un mismo modelo no aparezca por duplicado para una misma compra. Su código ASL es el siguiente:

```
<events>
  <beforeinsert>
    <action type="php">
      <code>
        return !waine_DsGetNumRows("am.REL_Purchase_Model","fkmodel='%values[1]' and
        fkpur='%values[2]'");
      </code>
      <msg>

```

```

es=Insercion abortada. Esa modelo ya existe en esta compra|en=Insert aborted. Model
already exists in this purchase
</msg>
</action>
</beforeinsert>
</events>

```

- **Evento “Cálculo de la imputación total de una oferta”:** calcula el total de dinero que supone todos los elementos relacionados a una determinada oferta. Su código ASL es el siguiente:

```

<action id="gti.am.actsumoffer" type="source">
  <code>
    UPDATE pp.Offer SET total=(SELECT sum(price*num) FROM am.REL_Offer_Model WHERE
fkoffer=%values[2]) WHERE pk=%values[2];
  </code>
</action>

```

- **Evento “Cálculo de la imputación total de una compra”:** calcular el total de dinero que supone todos los elementos relacionados a una determinada compra. Su código ASL es el siguiente:

```

<action id="gti.am.actsumpur" type="source">
  <code>
    UPDATE pp.Purchase SET total=(SELECT sum(price*num) FROM am.REL_Purchase_Model WHERE
fkpur=%values[2]) WHERE pk=%values[2];
  </code>
</action>

```

4.4 Construcción de las interfaces de usuario

4.4.1 Construcción de las interfaces de usuario módulo am

En esta sección, se detalla el proceso de construcción de las interfaces de usuario para cada uno de los roles definidos en la aplicación. Esta construcción se ha hecho en base a los diagramas entidad-relación anotados mostrados en el apartado 5 del Documento de Diseño.

La implementación de estas interfaces se hecho a través del lenguaje ASL propio del MBUIDE de **WAINE**. En este lenguaje, se definen tres modelos: Modelo de Usuario, Modelo de Presentación y Modelo de Diálogo.

En los siguientes apartados, se incluirá cada uno de los modelos para cada usuario, interfaz y el resultado gráfico en forma de captura de la interfaz de usuario.

4.4.1.1 Interfaces comunes

En este apartado, se describen las interfaces que van a ser comunes para todos los usuarios autenticados en el módulo que permite tener una visión de los activos que están asignados al usuario actual. En el módulo de inventario, esta sección está constituida por la interfaz: *Consulta de recursos asignados*.

4.4.1.1.1 Consulta de recursos asignados

El código ASL correspondiente al Modelo de Presentación para la interfaz *Consulta de recursos asignados* es el siguiente:



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```

<struct id="gti.am.cassignresourcequery1" type="relation">
  <param name="form_split" value="rows=50%,*"/>

  <param name="formid" value="gti.am.fasset2"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1:2"/>
  <param name="navigator_size" value="6"/>
  <param name="form_caption" value="es=Activo|en=Asset"/>
  <param name="form_subtype" value="tab"/>
  <param name="form_tabs" value="es=Informacion|en=Information:1,2,3,4,5,6,7,8,9;es=Detalles|
  en=Details:10,11,12,13,14,15;es=Foto|en=Photo:16"/>
  <param name="fields_readonly" value="1-16"/>
  <param name="button_insert" value="0"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="form_modify[0]" value="filter#pk in (SELECT pk FROM am.Asset WHERE run='%username' AND
  fkasset is null)"/>

  <param ord="2" name="structid" value="gti.am.cassignresourcequery2"/>
</struct>

```

El código que contiene a sus tres pestañas Propiedades, Propiedades Modelo y Recursos Hijo es el siguiente:

```

<struct id="gti.am.cassignresourcequery2" type="tab">
  <param name="form_split" value="rows=70,*"/>

  <param ord="1" name="formid" value="gti.am.fassetattr"/>
  <param ord="1" name="tab_name" value="es=Propiedades|en=Properties"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_readonly" value="1-3"/>
  <param ord="1" name="button_insert" value="0"/>
  <param ord="1" name="button_update" value="0"/>
  <param ord="1" name="button_delete" value="0"/>
  <param ord="1" name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="gti.am.fmodelattr"/>
  <param ord="2" name="form_source" value="am.VIEW_assetmodelattr"/>
  <param ord="2" name="tab_name" value="es=Propiedades Modelo|en=Model properties"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_readonly" value="1-3"/>
  <param ord="2" name="button_data" value="0"/>
  <param ord="2" name="button_misc" value="0"/>
  <param ord="2" name="source_filter_field" value="fk"/>

  <param ord="3" name="formid" value="gti.am.fasset1"/>
  <param ord="3" name="tab_name" value="es=Recursos Hijo|en=Child Resources"/>
  <param ord="3" name="form_type" value="table"/>
  <param ord="3" name="fields_hidden" value="1:2"/>
  <param ord="3" name="fields_readonly" value="3:4:5:6"/>
  <param ord="3" name="button_insert" value="0"/>
  <param ord="3" name="button_update" value="0"/>
  <param ord="3" name="button_delete" value="0"/>
  <param ord="3" name="fields_remove" value="7:8:9:10:11:12:13:15"/>
  <param ord="3" name="fields_modify[0]" value="wdgparam#struct.php?structid=gti.am.cassignre-
  sourcequerychild1&source_filter_field=pk&source_filter_value=;_blank;;Ver#14"/>

```

```

<param ord="3" name="fields_modify[1]" value="caption#Vis.Activos#14" />
<param ord="3" name="source_filter_field" value="fkasset"/>
</struct>

```

Y la interfaz en la aplicación es la siguiente:

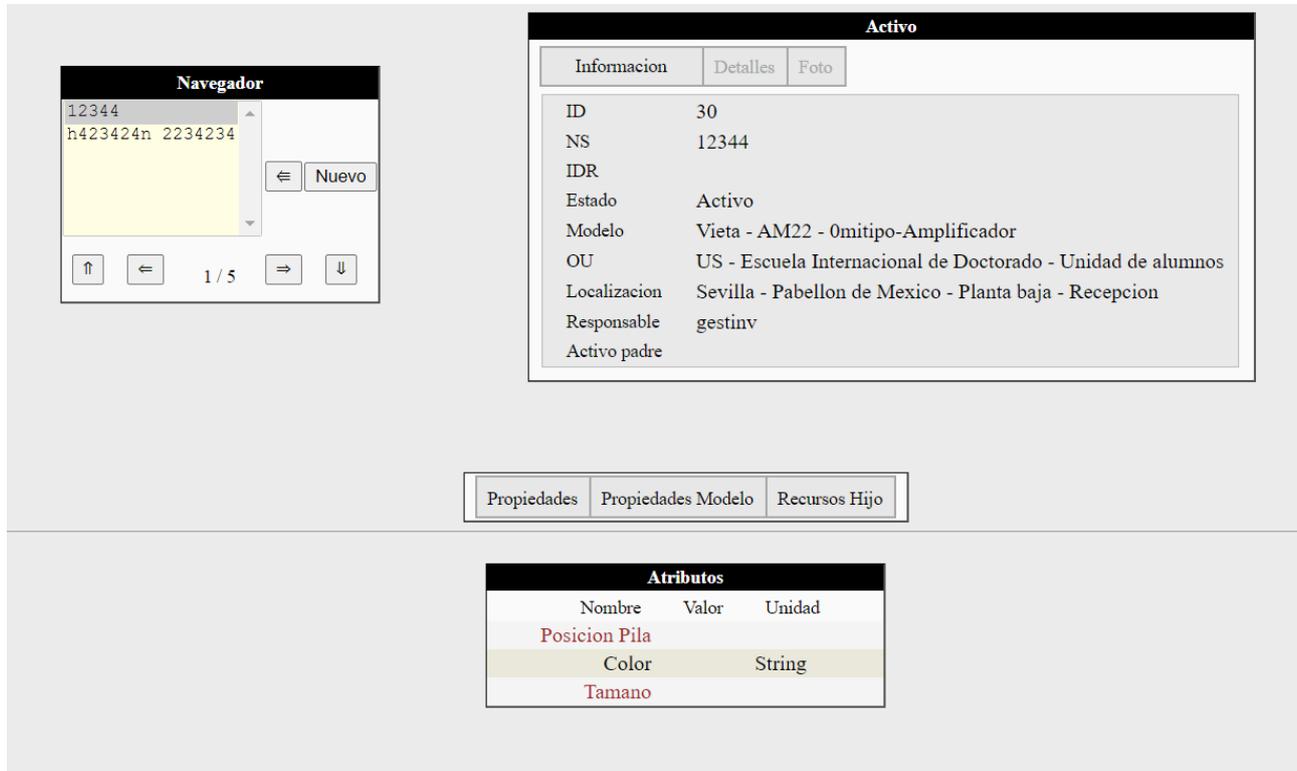


Ilustración 91: Interfaz de usuario final Consulta de recursos asignados

En la pestaña Recursos Hijo, se puede tener una vista más detallada del activo en cuestión. Su código ASL correspondiente al Modelo de Presentación para esta interfaz es:

```

<struct id="gti.am.cassignresourcequerychild1" type="relation">
  <param name="form_split" value="rows=50%,*"/>

  <param name="formid" value="gti.am.fasset2"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="none"/>
  <param name="source_row" value="0"/>
  <param name="form_subtype" value="tab"/>
  <param name="form_tabs" value="es=Informacion|en=Information:1,2,3,4,5,6,7,8,9;es=Detalles|
en=Details:10,11,12,13,14,15;es=Foto|en=Photo:16"/>
  <param name="form_caption" value="es=Activo|en=Asset"/>
  <param name="fields_readonly" value="1" />
  <param name="fields_modify[0]" value="widget#combonull#9"/>
  <param name="button_insert" value="0"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="source_filter_field" value="pk" />
  <param name="fields_readonly" value="1-16" />

  <param ord="2" name="structid" value="gti.am.cassignresourcequerychild2"/>
</struct>

```



El código que contiene a sus tres pestañas Propiedades, Propiedades Modelo y Recursos Hijo es el siguiente:

```
<struct id="gti.am.cassignresourcequerychild2" type="tab">
  <param name="form_split" value="rows=70,*"/>

  <param ord="1" name="formid" value="gti.am.fassetatr"/>
  <param ord="1" name="tab_name" value="es=Propiedades|en=Properties"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_readonly" value="1-3"/>
  <param ord="1" name="button_insert" value="0"/>
  <param ord="1" name="button_update" value="0"/>
  <param ord="1" name="button_delete" value="0"/>
  <param ord="1" name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="gti.am.fmodelatr"/>
  <param ord="2" name="form_source" value="am.VIEW_assetmodelatr"/>
  <param ord="2" name="tab_name" value="es=Propiedades Modelo|en=Model properties"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_readonly" value="1-3"/>
  <param ord="2" name="button_data" value="0"/>
  <param ord="2" name="button_misc" value="0"/>
  <param ord="2" name="source_filter_field" value="fk"/>

  <param ord="3" name="formid" value="gti.am.fasset1"/>
  <param ord="3" name="tab_name" value="es=Recursos Hijo|en=Child Resources"/>
  <param ord="3" name="form_type" value="table"/>
  <param ord="3" name="fields_hidden" value="1:2"/>
  <param ord="3" name="fields_readonly" value="3:4:5:6"/>
  <param ord="3" name="button_insert" value="0"/>
  <param ord="3" name="button_update" value="0"/>
  <param ord="3" name="button_delete" value="0"/>
  <param ord="3" name="fields_remove" value="7:8:9:10:11:12:13:15"/>
  <param ord="3" name="fields_modify[0]" value="wdgparam#struct.php?structid=gti.am.cassignre-
sourcequerychild1&source_filter_field=pk&source_filter_value=;_blank;;Ver#14"/>
  <param ord="3" name="fields_modify[1]" value="caption#Vis.Activos#14" />
  <param ord="3" name="source_filter_field" value="fkasset"/>
</struct>
```

Y la interfaz final en la aplicación es la siguiente:

Activo

Informacion	Detalles	Foto
-------------	----------	------

ID	25
NS	3123213
IDR	
Estado	Activo
Modelo	Mark 1 - Model 1 - Laptop computer
OU	US - Escuela Internacional de Doctorado - Unidad de alumnos
Localizacion	Sevilla - Pabellon de Mexico - Planta baja - Recepcion
Responsable	admincom - Antonio Delgado
Activo padre	12344

Propiedades	Propiedades Modelo	Recursos Hijo
-------------	--------------------	---------------

Atributos		
Nombre	Valor	Unidad

Ilustración 92: Interfaz de usuario final Consulta de recursos asignados. Visualización de activos

4.4.1.2 Administrador del módulo de inventario

Las funcionalidades a las que tiene acceso un usuario con rol *administrador del módulo de inventario* son las siguientes:

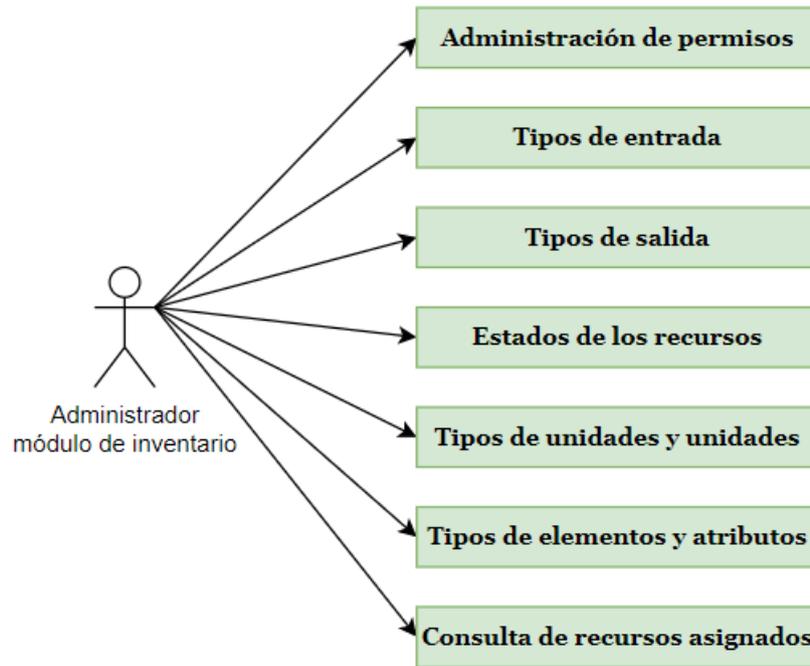


Ilustración 93: Esquema menú Administrador del módulo de inventario

El código ASL correspondiente al Modelo de Diálogo del administrador del módulo de inventario es el siguiente:

```
<main id="gti.am.madmininv" caption="es=AM.Administrador del modulo inventario|en=AM.Asset Module Administrator">

  <menu caption="es=Gestion modulo inventario|en=Asset Module Management" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Administracion de permisos|en=Permission Management" call="gti.am.cpermmanag1"/>
    <option caption="es=Tipos de entrada|en=Input Types" call="gti.am.cinputtype"/>
    <option caption="es=Tipos de salida|en=Output Types" call="gti.am.coutputtype"/>
    <option caption="es=Estados de los recursos|en=Resource States" call="gti.am.cstatus"/>
    <option caption="es=Tipos de unidades y unidades|en=Unit Types and Units" call="gti.am.cunit"/>
    <option caption="es=Tipos de elementos y atributos|en=Element Types and Attributes" call="gti.am.cassettype1"/>
  </menu>

  <menu caption="es=Recursos asignados|en=Assigned Resources" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Consulta de recursos asignados|en=Assigned Resources Query" call="gti.am.cassignresourcequery1"/>
  </menu>

  <menu caption="Misc" tooltip="Otras cosas">
    <option caption="es=Sobre este programa|en=About" call="gti.am.cmodinfo" />
    <option caption="es=Ayuda|en=Help" url="http://wayne.us.es/dokuwiki/doku.php?id=wayne:help:app:gti:am:menu:administrador_modulo_inventario" />
    <option caption="es=Sobre WAINE|en=About WAINE" url="http://www.wayne.org" />
    <option caption="es=Salir|en=Logout" url="logout.php" />
  </menu>

</main>
```

En los siguientes apartados se describirán cada una de las funcionalidades.

4.4.1.2.1 Administración de permisos

El código ASL correspondiente al Modelo de Presentación para la interfaz *Administración de permisos* es el siguiente:

```
<struct id="gti.am.cpermmanag1" type="relation">
  <param name="form_split" value="rows=40%,*"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="gti.Scope"/>
  <param name="form_type" value="form"/>
  <param name="form_caption" value="es=Ambitos|en=Scopes"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="button_insert" value="0"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="fields_readonly" value="1:2"/>
  <param name="fields_remove" value="3"/>

  <param ord="2" name="structid" value="gti.am.cpermmanag2"/>
</struct>
```

Esta, a su vez, está relacionada con otra estructura que contiene cuatro pestañas: Gest. Adquisiciones, Gest. Inventario, Tec. Avanzados y Tecnicos. Estas pestañas incluirán los usuarios que tienen permisos sobre los activos de las sedes que se encuentran en los ámbitos descritos en el caso de los gestores de inventarios y técnicos. En el caso de los gestores de adquisiciones, indicarán los permisos sobre las ofertas de los ámbitos a las que tengan permisos dichos usuarios. Su código ASL es el siguiente:

```
<struct id="gti.am.cpermmanag2" type="tab" >
  <param name="form_split" value="rows=100,*"/>

  <param ord="1" name="formid" value="gti.am.frelscope"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="form_source" value="am.REL_Gad_Scope"/>
  <param ord="1" name="tab_name" value="es=Gest. Adquisiciones|en=Purchase Manag."/>
  <param ord="1" name="form_caption" value="es=Gestores de Adquisiciones|en=Purchase Managers"/>
  <param ord="1" name="button_insert" value="1"/>
  <param ord="1" name="button_update" value="0"/>
  <param ord="1" name="button_delete" value="1"/>
  <param ord="1" name="fields_hidden" value="2"/>
  <param ord="1" name="source_filter_field" value="fkscope"/>
  <param ord="1" name="form_modify[0]" value="beforeinsert#gti.am.actgad"/>

  <param ord="2" name="formid" value="gti.am.frelscope"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="form_source" value="am.REL_Gin_Scope"/>
  <param ord="2" name="tab_name" value="es=Gest. Inventario|en=Asset Manag."/>
  <param ord="2" name="form_caption" value="es=Gestores de Inventario|en=Asset Managers"/>
  <param ord="2" name="button_insert" value="1"/>
  <param ord="2" name="button_update" value="0"/>
  <param ord="2" name="button_delete" value="1"/>
  <param ord="2" name="fields_hidden" value="2"/>
  <param ord="2" name="source_filter_field" value="fkscope"/>
  <param ord="2" name="form_modify[0]" value="beforeinsert#gti.am.actgin"/>
```



```

<param ord="3" name="formid" value="gti.am.frelscopeu"/>
<param ord="3" name="form_type" value="table"/>
<param ord="3" name="form_source" value="am.REL_Ate_Scope"/>
<param ord="3" name="tab_name" value="es=Tec. Avanzandos|en=Advanced Tech."/>
<param ord="3" name="form_caption" value="es=Tecnicos avanzados|en=Advanced Technicians"/>
<param ord="3" name="button_insert" value="1"/>
<param ord="3" name="button_update" value="0"/>
<param ord="3" name="button_delete" value="1"/>
<param ord="3" name="fields_hidden" value="2"/>
<param ord="3" name="source_filter_field" value="fkscope"/>
<param ord="3" name="form_modify[0]" value="beforeinsert#gti.am.actate"/>

<param ord="4" name="formid" value="gti.am.frelscopeu"/>
<param ord="4" name="form_type" value="table"/>
<param ord="4" name="form_source" value="am.REL_Tec_Scope"/>
<param ord="4" name="tab_name" value="es=Tecnicos|en=Technicians"/>
<param ord="4" name="form_caption" value="es=Tecnicos|en=Technicians"/>
<param ord="4" name="button_insert" value="1"/>
<param ord="4" name="button_update" value="0"/>
<param ord="4" name="button_delete" value="1"/>
<param ord="4" name="fields_hidden" value="2"/>
<param ord="4" name="source_filter_field" value="fkscope"/>
<param ord="4" name="form_modify[0]" value="beforeinsert#gti.am.acttec"/>
</struct>
    
```

Finalmente, la interfaz en la aplicación es la siguiente:

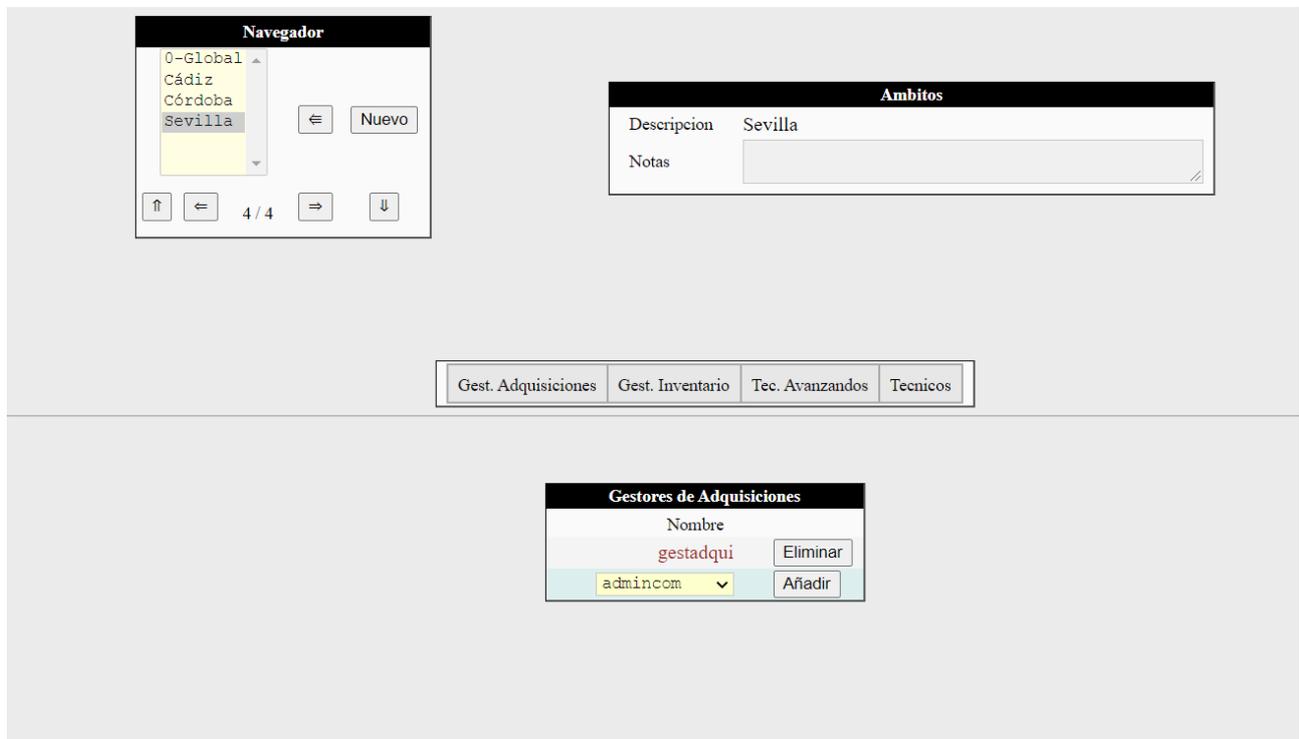


Ilustración 94: Interfaz de usuario final Administración de permisos

4.4.1.2.2 Tipos de entrada

El código ASL correspondiente al Modelo de Presentación para la interfaz *Tipos de entrada* es el siguiente:

```

<struct id="gti.am.cinputtype" type="form">
    
```

```

<param name="form_type" value="table"/>

<param name="formid" value="fdescr.form.descr"/>
<param name="form_source" value="am.InputType"/>
<param name="form_caption" value="es=Tipos de entradas|en=Input Types"/>
<param name="button_insert" value="1"/>
<param name="button_update" value="0"/>
<param name="button_delete" value="1"/>
<param name="form_modify[0]" value="beforedelete#gti.am.actinptype"/>
</struct>

```

Y su interfaz en la aplicación es la siguiente:

Tipos de entradas	
Descripcion	
Adquisicion	Eliminar
Interna	Eliminar
Devolucion	Eliminar
	Añadir

Ilustración 95: Interfaz de usuario final Tipos de entrada

4.4.1.2.3 Tipos de salida

El código ASL correspondiente al Modelo de Presentación para la interfaz *Tipos de salida* es el siguiente:

```

<struct id="gti.am.coutputtype" type="form">
  <param name="form_type" value="table"/>

  <param name="formid" value="fdescr.form.descr"/>
  <param name="form_source" value="am.OutputType"/>
  <param name="form_caption" value="es=Tipos de salidas|en=Output Types"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="1"/>
  <param name="form_modify[0]" value="beforedelete#gti.am.actouttype"/>
</struct>

```

Y su interfaz en la aplicación es la siguiente:



Tipos de salidas	
Descripcion	
Interna	Eliminar
Prestamo	Eliminar
Baja	Eliminar
	Añadir

Ilustración 96: Interfaz de usuario final Tipos de salida

4.4.1.2.4 Estados de los recursos

El código ASL correspondiente al Modelo de Presentación para la interfaz *Estados de los recursos* es el siguiente:

```
<struct id="gti.am.cstatus" type="form">
  <param name="form_type" value="table"/>

  <param name="formid" value="fdescr.form.descr"/>
  <param name="form_source" value="am.Status"/>
  <param name="form_caption" value="es=Estado de los recursos|en=Resource status"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="1"/>
  <param name="form_modify[0]" value="beforedelete#gti.am.actstatus"/>
</struct>
```

Y su interfaz es la siguiente:

Estado de los recursos	
Descripcion	
Activo	Eliminar
Baja	Eliminar
	Añadir

Ilustración 97: Interfaz de usuario final Estados de los recursos

4.4.1.2.5 Tipos de unidades y unidades

El código ASL correspondiente al Modelo de Presentación para la interfaz *Tipos de unidades y unidades* es el siguiente:

```
<struct id="gti.am.cunit" type="relation">
  <param name="form_split" value="rows=340,*"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="am.UnitType"/>
  <param name="form_type" value="form"/>
  <param name="source_order" value="pk"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="10"/>
```

```

<param name="button_data" value="1"/>
<param name="form_caption" value="es=Tipo de unidad |en=Unit Type"/>
<param name="fields_remove" value="3"/>

<param ord="2" name="formid" value="gti.am.funit"/>
<param ord="2" name="form_type" value="table"/>
<param ord="2" name="button_data" value="1"/>
<param ord="2" name="source_filter_field" value="fk"/>
</struct>

```

Y su interfaz en la aplicación es la siguiente:

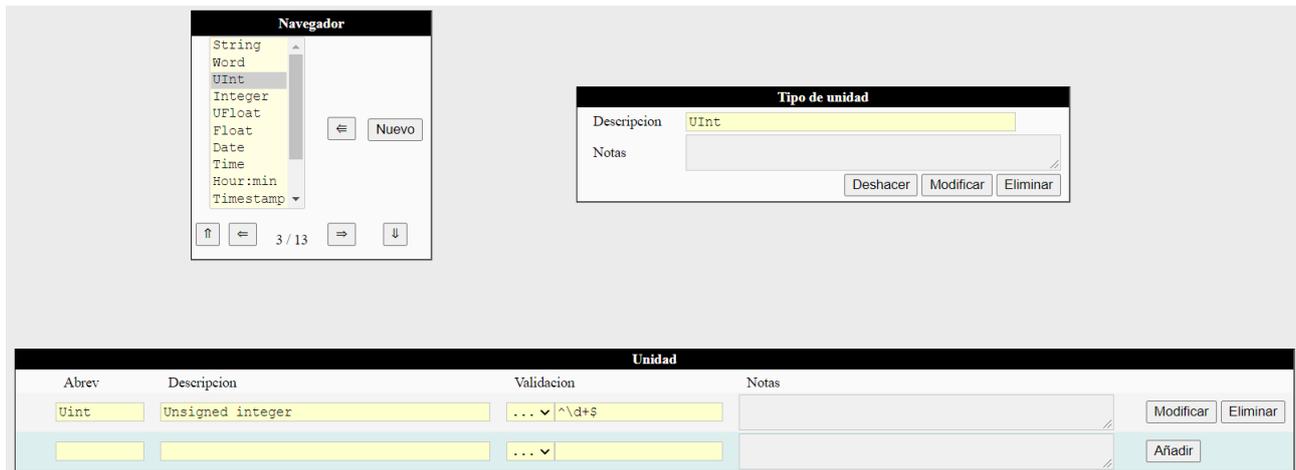


Ilustración 98: Interfaz de usuario final Tipos de unidades y unidades

4.4.1.2.6 Tipos de elementos y atributos

El código ASL correspondiente al Modelo de Presentación para la interfaz *Tipos de elementos y atributos* es el siguiente:

```

<struct id="gti.am.casstype1" type="relation">
  <param name="form_split" value="rows=340,*"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="am.AssetType"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="10"/>
  <param name="button_data" value="1"/>
  <param name="form_caption" value="es=Tipo Elemento Inventariable |en=Inventory Element Type"/>
  <param name="fields_remove" value="3"/>

  <param ord="2" name="structid" value="gti.am.casstype2"/>
</struct>

```

Esta estructura, a su vez, está relacionada con otra que contiene dos tablas con los atributos de tipo modelo y los atributos de tipo activo. Su código ASL correspondiente al Modelo de Presentación para estas pestañas es el siguiente:

```

<struct id="gti.am.casstype2" type="split">
  <param name="form_split" value="cols=50%,*"/>

  <param name="formid" value="gti.am.finveltypeattr"/>
  <param name="form_source" value="am.ModelTypeAttr"/>

```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```
<param name="form_type" value="table"/>
<param name="form_caption" value="es=Atributo Tipo Modelo|en=Model Type Attribute"/>
<param name="button_data" value="1"/>
<param name="source_filter_field" value="fktype"/>

<param ord="2" name="formid" value="gti.am.finveltypeattr"/>
<param ord="2" name="form_source" value="am.AssetTypeAttr"/>
<param ord="2" name="form_type" value="table"/>
<param ord="2" name="form_caption" value="es=Atributo Tipo Activo|en=Asset Type Attribute"/>
<param ord="2" name="button_data" value="1"/>
<param ord="2" name="source_filter_field" value="fktype"/>
</struct>
```

Y su interfaz final en la aplicación es la siguiente:

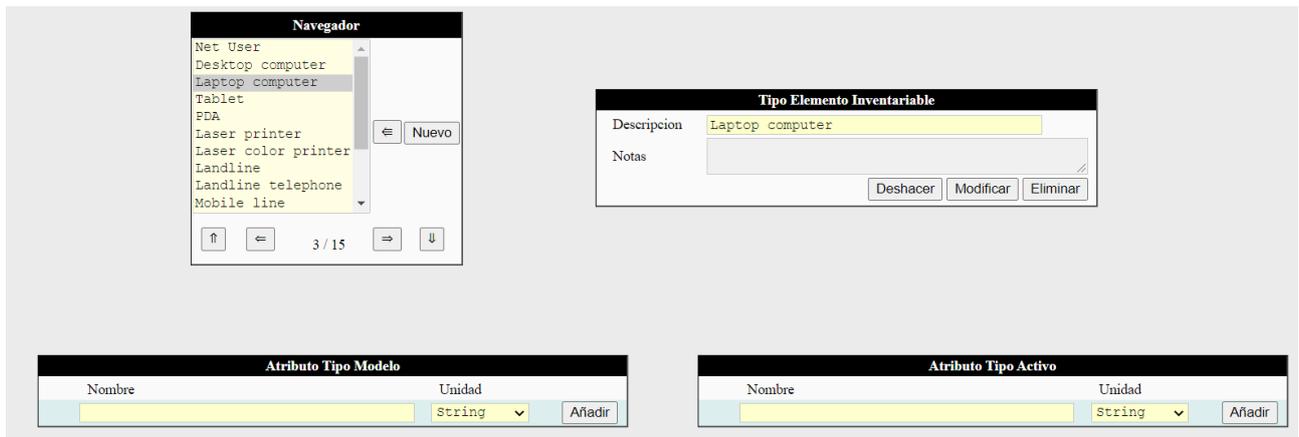


Ilustración 99: Interfaz de usuario final Tipos de elementos y atributos

4.4.1.3 Gestor de inventario

Las funcionalidades a las que tiene acceso un usuario con rol *gestor de inventario* son las siguientes:

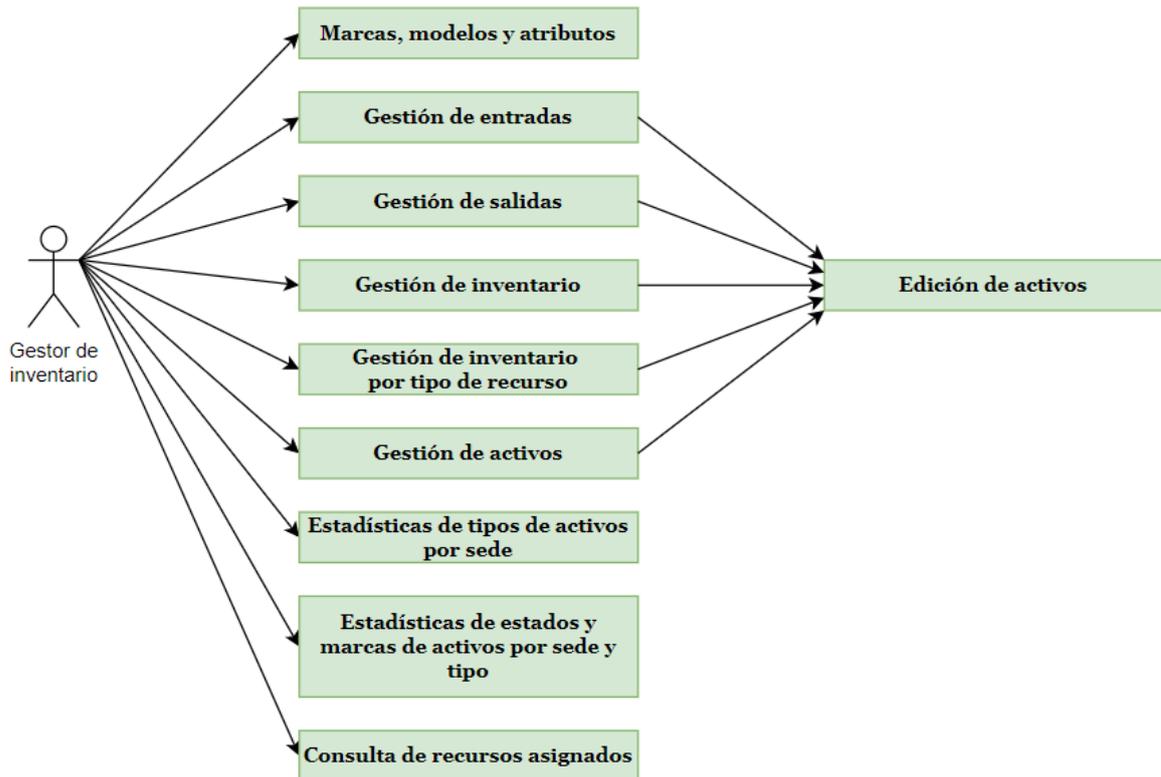


Ilustración 100: Esquema menú Gestor de inventario

El código ASL correspondiente al Modelo de Diálogo del gestor de inventario es el siguiente:

```

<main id="gti.am.mgestinv" caption="es=AM.Gestor de inventario|en=AM.Asset Manager">
  <menu caption="es=Gestion inventario|en=Asset Management" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Marcas, modelos y atributos|en=Marks, Models and Attributes"
      call="gti.am.cmarkmodel1"/>
    <option caption="es=Gestion de entradas|en=Input Management" call="gti.am.cinputmanag1"/>
    <option caption="es=Gestion de salidas|en=Output Management" call="gti.am.coutputmanag1"/>
    <option caption="es=Gestion de inventario|en=Asset Management" call="gti.am.cassetmanag"/>
    <option caption="es=Gestion de inventario por tipo de recurso|en=Asset Management by Resource Type"
      call="gti.am.cassetmanagtype" />
    <option caption="es=Gestion de activos|en=Asset Management" call="gti.am.cassetmanag1"/>
    <option caption="es=Estadísticas de tipos de activos por sede|en=Statistics of types of assets by sites"
      call="gti.am.cgraphtype" />
    <option caption="es=Estadísticas de estados y marcas de activos por sede y tipo|
      en=Statistics of status and mark of assets by sites and types" call="gti.am.cgraphstatusmark1" />
  </menu>

  <menu caption="es=Recursos asignados|en=Assigned Resources" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Consulta de recursos asignados|en=Assigned Resources Query" call="gti.am.cassignre-
sourcequery1"/>
  </menu>

  <menu caption="Misc" tooltip="Otras cosas">
    <option caption="es=Sobre este programa|en=About" call="gti.am.cmodinfo" />
    <option caption="es=Ayuda|en=Help" url="http://waine.us.es/dokuwiki/doku.php?
id=waine:help:app:gti:am:menu:gestor_inventario" />
    <option caption="es=Sobre WAINE|en=About WAINE" url="http://www.waine.org" />
    <option caption="es=Salir|en=Logout" url="logout.php" />
  </menu>

```



```
</menu>
```

```
</main>
```

En los siguientes apartados se describirán cada una de las funcionalidades.

4.4.1.3.1 Marcas, modelos y atributos

El código ASL correspondiente al Modelo de Presentación para la interfaz *Marcas, modelos y atributos* es el siguiente:

```
<struct id="gti.am.cmarkmodel1" type="relation">
  <param name="form_split" value="rows=280,*"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="am.Mark"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="8"/>
  <param name="button_data" value="1"/>
  <param name="form_caption" value="es=Marca|en=Mark"/>
  <param name="fields_remove" value="3"/>

  <param ord="2" name="structid" value="gti.am.cmarkmodel2"/>
</struct>

<struct id="gti.am.cmarkmodel2" type="relation">
  <param name="form_split" value="rows=280,*"/>

  <param name="formid" value="gti.am.fmodel"/>
  <param name="form_type" value="form"/>
  <param name="form_subtype" value="tab"/>
  <param name="form_tabs" value="es=Informacion|en=Information:1,2,3,6;es=Notas|en=Re-
marks:4;es=Foto|en=Photo:5"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="2"/>
  <param name="navigator_size" value="8"/>
  <param name="button_data" value="1"/>
  <param name="button_misc" value="0"/>
  <param name="fields_modify[0]" value="attr#l#1" />
  <param name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="gti.am.fmodelattr"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="button_update" value="1"/>
  <param ord="2" name="source_filter_field" value="fk"/>
</struct>
```

Y su interfaz final en la aplicación es la siguiente:

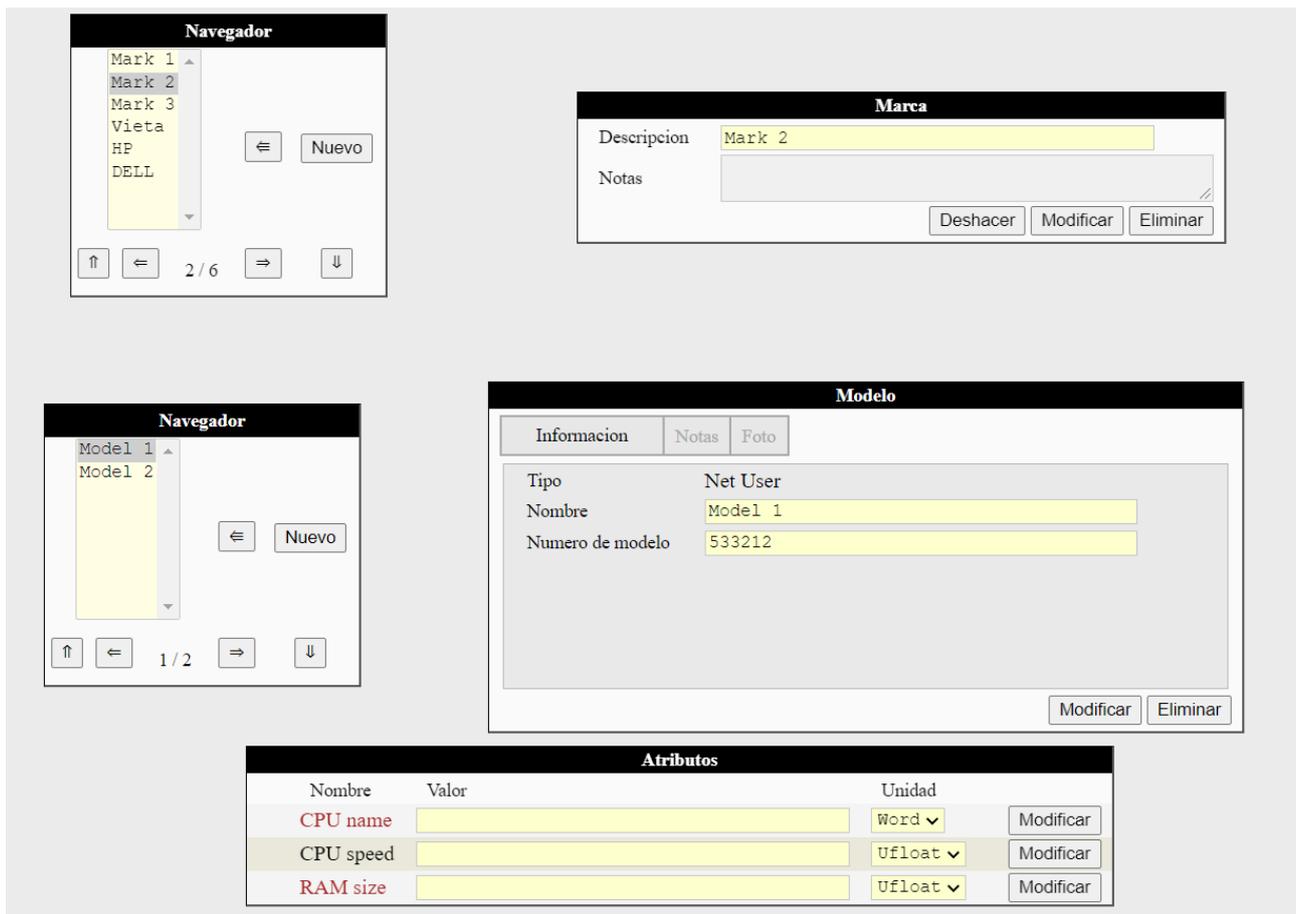


Ilustración 101: Interfaz de usuario final Marcas, modelos y atributos

4.4.1.3.2 Gestión de entradas

El código ASL correspondiente al Modelo de Presentación para la interfaz *Gestión de entradas* se divide en un filtro superior:

```
<struct id="gti.am.cinputmanag1" type="relation">
  <param name="form_split" value="rows=165,*"/>

  <param name="formid" value="gti.am.finput"/>
  <param name="tab_name" value="Filtros basicos"/>
  <param name="form_type" value="filter-where"/>
  <param name="form_tabs" value="Basicos:1,2,3;Destino:7;Origen:6;Origen Alternativo:8;Avanzados:4,5"/>
  <param name="fields_modify[0]" value="canbenull#Y#1-13"/>
  <param name="form_filterw_oper"
value="|LIKE;|LIKE;|LIKE;|LIKE;|LIKE;|LIKE;|LIKE;|LIKE;|LIKE;|LIKE;|LIKE;|LIKE;"/>
  <param name="form_subtype" value="wtab"/>
  <param name="button_action" value="0"/>
  <param name="form_caption" value="Filtros"/>
  <param name="fields_remove" value="4:7:8:11:12"/>
  <param name="fields_modify[1]" value="widget#combonull#3:6:7-10"/>
  <param name="fields_modify[2]" value="source#pk#1"/>
  <param name="form_modify[0]" value="filter#dst in (select pk from gti.VIEW_Loc where scopepk in (select
fkscope from am.REL_Gin_Scope where uid=%userid))"/>
  <param name="fields_modify[3]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Gin_Scope where uid=%userid);;#10"/>
  <param name="form_filterw_nexus" value="and"/>
  <param name="button_misc" value="0"/>

  <param ord="2" name="structid" value="gti.am.cinputmanag2"/>
```



</struct>

En la aplicación, esto sería:

Ilustración 102: Interfaz de usuario final Gestión de entradas. Filtros

Posteriormente, se mostrarán las entradas correspondientes al filtrado:

```
<struct id="gti.am.cinputmanag2" type="relation">
  <param name="form_split" value="rows=395,*"/>

  <param name="formid" value="gti.am.finput"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="0:1"/>
  <param name="navigator_size" value="6"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="1"/>
  <param name="button_misc" value="0"/>
  <param name="fields_readonly" value="1"/>
  <param name="form_tabs"
    value="es=Informacion|en=Information:1,2,3,4,5,6,7,8,9,13,10;es=Notas|en=Re-
marks:11;es=Detalles|en=Details:12"/>
  <param name="form_subtype" value="tab"/>
  <param name="fields_modify[0]" value="canbenull#Y#1:9"/>
  <param name="fields_modify[1]" value="widget#combonull#7-9"/>
  <param name="fields_modify[2]" value="attr#l#3"/>
  <param name="form_modify[0]" value="filter#dst in (select pk from gti.VIEW_Loc where scopepk in (select
fkscope from am.REL_Gin_Scope where uid=%userid))"/>
  <param name="fields_modify[3]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Gin_Scope where uid=%userid);;#10"/>
  <param name="fields_modify[4]" value="search#DATA:pp.Purchase;fkoffer IN (SELECT pp.Offer.pk FROM
pp.Offer,pp.OfferReq WHERE pp.Offer.fkreq=pp.OfferReq.pk AND pp.OfferReq.type='am');;#7"/>
  <param name="fields_modify[5]" value="defvalue#%username#6" />
  <param name="__nofill" value="1"/>

  <param ord="2" name="structid" value="gti.am.cinputmanag3"/>
</struct>
```

En la aplicación:

Ilustración 103: Interfaz de usuario final Gestión de entradas. Formulario

Finalmente, hay tres pestañas Activos, Resumen y Adjuntos, relacionados con la entrada.

- Pestaña Activos: los activos relacionados con la entrada. Su código ASL es el siguiente:

```
<param ord="1" name="formid" value="gti.am.fassetADG"/>
<param ord="1" name="tab_name" value="es=Activos|en=Assets"/>
<param ord="1" name="form_type" value="table"/>
<param ord="1" name="form_modify[0]" value="beforeupdate#gti.am.actns"/>
<param ord="1" name="form_modify[1]" value="beforeinsert#gti.am.actns"/>
<param ord="1" name="form_modify[2]" value="afterinsert#gti.am.actinput"/>
<param ord="1" name="button_insert" value="1"/>
<param ord="1" name="button_update" value="1"/>
<param ord="1" name="button_delete" value="1"/>
<param ord="1" name="source_filter_where" value="pk in (select fkasset from am.REL_Input_Asset
where fkinput=%filtervalue)"/>
```

Esto se muestra en la aplicación de la siguiente forma:

Marca y Modelo	NS	IDR	Ed. Activos
Mark 1 - Model 1	ns1	id1	Ver
Mark 1 - Model 1	ns2	id2	Ver
Mark 1 - Model 2	ns3	id3	Ver
Mark 1 - Model 2	ns4		Ver
Mark 1 - Model 2	ns5		Ver

Ilustración 104: Interfaz de usuario final Gestión de entradas. Pestaña Activos

- Pestaña Resumen: muestra un resumen de los activos incluidos en la entrada elegida según el tipo. Su código ASL correspondiente es el siguiente:

```
<param ord="2" name="tab_name" value="es=Resumen|en=Summary"/>
<param ord="2" name="formid" value="gti.am.finputssummary"/>
<param ord="2" name="form_type" value="table"/>
<param ord="2" name="fields_hidden" value="3"/>
<param ord="2" name="button_data" value="0"/>
```



```
<param ord="2" name="source_filter_field" value="fkinput"/>
```

Y su diseño en la aplicación es de la siguiente forma:

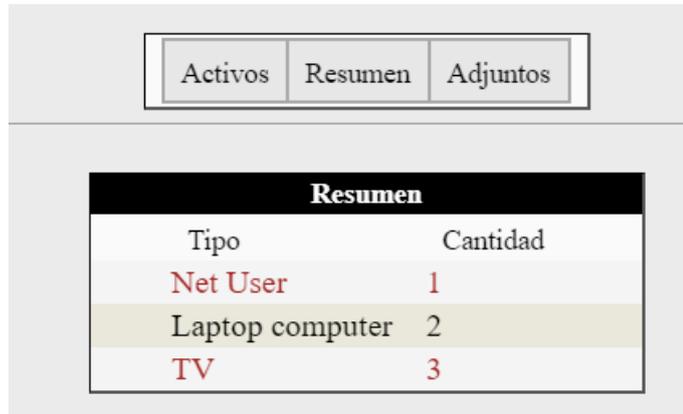


Ilustración 105: Interfaz de usuario final Gestión de entradas. Pestaña Resumen

- Pestaña Adjuntos: muestra los adjuntos relacionados con la entrada. Su código ASL relacionado es el siguiente:

```
<param ord="3" name="tab_name" value="es=Adjuntos|en=Attachments"/>
<param ord="3" name="formid" value="attach.form.catattach"/>
<param ord="3" name="form_type" value="table"/>
<param ord="3" name="fields_modify[0]" value="defvalue#10#3"/> <!-- Asignando categoría -->
<param ord="3" name="fields_remove" value="5:7"/>
<param ord="3" name="fields_hidden" value="3:4:6"/>
<param ord="3" name="button_data" value="1"/>
<param ord="3" name="source_filter_field" value="fk"/>
<param ord="3" name="form_modify[0]" value="filter#fkcat=10"/>
```

Esto se muestra en la aplicación de la siguiente forma:



Ilustración 106: Interfaz de usuario final Gestión de entradas. Pestaña Adjuntos

4.4.1.3.3 Gestión de salidas

El Modelo de Presentación para la interfaz *Gestión de salidas* es muy similar al descrito en el apartado 4.3.2 de esta mismo documento. Está compuesto por un filtro superior cuyo código ASL es:

```
<struct id="gti.am.coutputmanag1" type="relation">
  <param name="form_split" value="rows=165,*"/>

  <param name="formid" value="gti.am.foutput"/>
  <param name="tab_name" value="Filtros basicos"/>
  <param name="form_type" value="filter-where"/>
  <param name="form_tabs" value="Basicos:1,2,3;Origen:6;Destino:7;Destino Alternativo:8;Avanzados:4,5"/>
```

```

<param name="fields_modify[0]" value="canbenull#Y#1-11"/>
<param name="form_filterw_oper" value="ILIKE;ILIKE;ILIKE;ILIKE;ILIKE;ILIKE;ILIKE;ILIKE;ILIKE;"/>
<param name="form_subtype" value="wtab"/>
<param name="button_action" value="0"/>
<param name="form_caption" value="Filtros"/>
<param name="fields_remove" value="4:9:10"/>
<param name="fields_modify[1]" value="widget#combonull#3:5:7:8"/>
<param name="fields_modify[2]" value="source#pk#1"/>
<param name="form_modify[0]" value="filter#src in (select pk from gti.VIEW_Loc where scopepk in (select
fkscope from am.REL_Gin_Scope where uid=%userid))"/>
<param name="fields_modify[3]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Gin_Scope where uid=%userid);;#7"/>
<param name="form_filterw_nexus" value="and"/>
<param name="button_misc" value="0"/>

<param ord="2" name="structid" value="gti.am.coutputmanag2"/>
</struct>

```

Y su representación en la aplicación:

Ilustración 107: Interfaz de usuario final Gestión de salidas. Filtros

Posteriormente, se muestran las salidas correspondientes al filtrado:

```

<struct id="gti.am.coutputmanag2" type="relation">
  <param name="form_split" value="rows=50%,*"/>

  <param name="formid" value="gti.am.foutput"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="0:1"/>
  <param name="navigator_size" value="6"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="1"/>
  <param name="button_misc" value="0"/>
  <param name="fields_readonly" value="1"/>
  <param name="form_tabs"
value="es=Informacion|en=Information:1,2,3,4,5,6,7,8,11;es=Notas|en=Remarks:9;es=Detalles|en=De-
tails:10"/>
  <param name="form_subtype" value="tab"/>
  <param name="fields_modify[0]" value="canbenull#Y#1:8"/>
  <param name="fields_modify[1]" value="widget#combonull#8"/>
  <param name="form_modify[0]" value="filter#src in (select pk from gti.VIEW_Loc where scopepk in (select
fkscope from am.REL_Gin_Scope where uid=%userid))"/>
  <param name="fields_modify[2]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Gin_Scope where uid=%userid);;#7"/>
  <param name="fields_modify[3]" value="defvalue##username#5" />
  <param name="__nofill" value="1"/>

```



```
<param ord="2" name="structid" value="gti.am.coutputmanag3"/>
</struct>
```

En la aplicación:

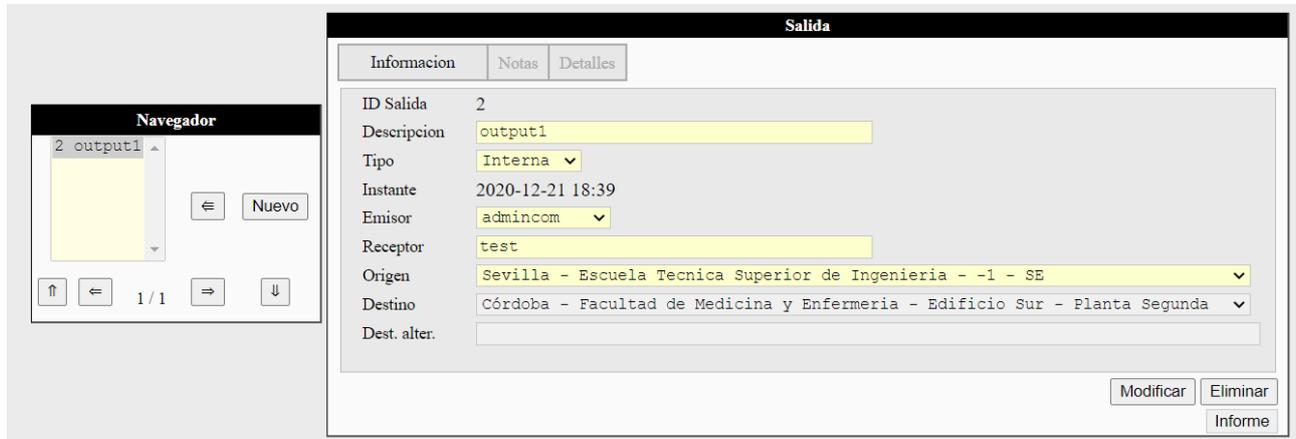


Ilustración 108: Interfaz de usuario final Gestión de salidas. Formulario

Finalmente, sus pestañas relacionadas: Activos, Resumen y Adjuntos.

- Pestaña Activos: contiene los activos relacionados con la salida:

```
<param ord="1" name="formid" value="gti.am.foutassetelem"/>
<param ord="1" name="tab_name" value="es=Activos|en=Assets"/>
<param ord="1" name="form_type" value="table"/>
<param ord="1" name="fields_hidden" value="1:4"/>
<param ord="1" name="button_insert" value="1"/>
<param ord="1" name="button_update" value="0"/>
<param ord="1" name="button_delete" value="1"/>
<param ord="1" name="source_filter_field" value="fkoutput"/>
```



Ilustración 109: Interfaz de usuario final Gestión de salidas. Pestaña Activos

- Pestaña Resumen: contiene un resumen de los activos relacionados con la salida según el tipo:

```
<param ord="2" name="tab_name" value="es=Resumen|en=Summary"/>
<param ord="2" name="formid" value="gti.am.foutputsummary"/>
```


Marca y Modelo	Responsable	NS	IDR	Ed.Activos	
Mark 2 - Model 1	admininv	ns2	id2	Ver	Modificar
Mark 1 - Model 2	admininv	ns3	id3	Ver	Modificar
Mark 1 - Model 2	admininv	ns4		Ver	Modificar
Mark 1 - Model 2	admininv	ns5		Ver	Modificar
Vieta - AM22	admininv	ns6	345345345	Ver	Modificar
Mark 3 - Model 1	gestadqui	ns9	id9	Ver	Modificar
Vieta - AM22	admincom	34534534		Ver	Modificar
Vieta - AM22	admincom	0931233		Ver	Modificar
Mark 1 - Model 1	admincom	231		Ver	Modificar
Mark 1 - Model 1	admincom	3123213		Ver	Modificar
Mark 1 - Model 1	admininv	43434		Ver	Modificar
Vieta - AM22	gestinv	12344		Ver	Modificar
Vieta - AM22	gestinv	h423424n	2234234	Ver	Modificar
Vieta - AM22	gestinv			Ver	Modificar
Mark 1 - Model 1	gestinv			Ver	Modificar
Mark 1 - Model 1	gestinv			Ver	Modificar

16

Ilustración 113: Interfaz de usuario final Gestión de inventario. Activos

4.4.1.3.5 Gestión de inventario por tipo de recurso

El código ASL correspondiente al Modelo de Presentación para la interfaz *Gestión de inventario por tipo de recurso* es el siguiente:

```
<struct id="gti.am.cassetmanagtype" type="relation">
  <param name="form_split" value="rows=25%,*"/>

  <param name="formid" value="gti.am.fresourcetype"/>
  <param name="form_type" value="filter-where"/>
  <param name="form_filterw_tpl" value="fkmodel IN (SELECT pk from am.Model where fktype=%values[2])
  AND fkloc=%values[1]" />
  <param name="fields_modify[0]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
  am.REL_Gin_Scope where uid=%userid);;#1"/>
  <param name="form_caption" value="Filtros"/>
  <param name="form_filterw_nexus" value="and"/>
  <param name="button_misc" value="0"/>
  <param name="button_action" value="0"/>

  <param ord="2" name="formid" value="gti.am.fasset1"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="button_insert" value="0"/>
  <param ord="2" name="button_update" value="1"/>
  <param ord="2" name="button_delete" value="0"/>
  <param ord="2" name="fields_hidden" value="7-13:15"/>
  <param ord="2" name="fields_modify[0]" value="widget#combonull#1:2"/>
  <param ord="2" name="fields_modify[1]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
  am.REL_Gin_Scope where uid=%userid);;#1"/>
  <param ord="2" name="__nofill" value="1"/>
</struct>
```



Y su interfaz final en la aplicación es:

- Filtro: contiene el conjunto de filtros:

Ilustración 114: Interfaz de usuario final Gestión de inventario por tipo de recurso. Filtros

- Los activos: contiene todos los activos que cumplen el filtrado superior:

Ilustración 115: Interfaz de usuario final Gestión de inventario por tipo de recurso. Activos

4.4.1.3.6 Gestión de activos

El Modelo de Presentación para la interfaz *Gestión de activos* está compuesto por dos combos superiores que filtran según el ámbito y la localización. Su código ASL es el siguiente:

```
<struct id="gti.am.cassetmanag1" type="relation">
  <param name="form_split" value="rows=90,*"/>

  <param name="formid" value="fdescr.form.descr"/>
  <param name="form_type" value="combo"/>
  <param name="form_source" value="gti.Scope"/>
  <param name="fields_hidden" value="2"/>
  <param name="source_filter_where" value="pk in (select fkscope from am.REL_Gin_Scope where
  uid=%userid)"/>
  <param name="form_caption" value="es=Ambito|en=Scope"/>

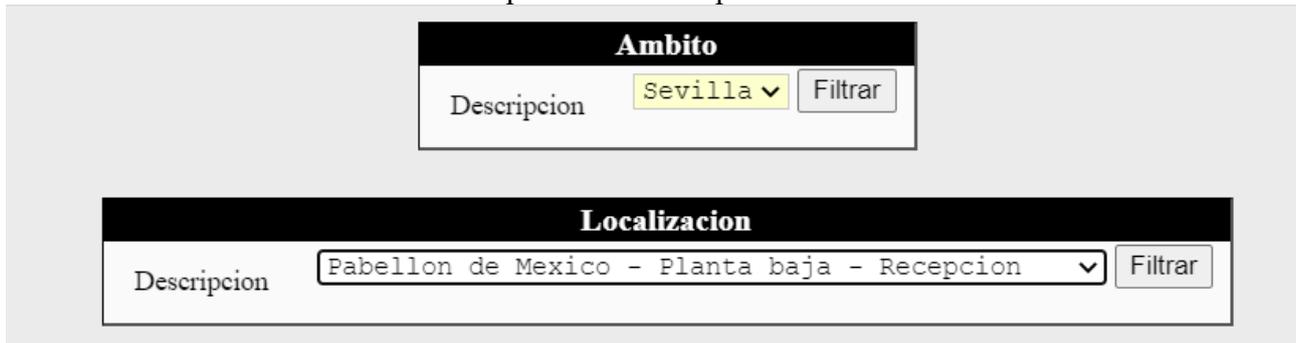
  <param ord="2" name="structid" value="gti.am.cassetmanag2"/>
</struct>

<struct id="gti.am.cassetmanag2" type="relation">
  <param name="form_split" value="rows=90,*"/>

  <param name="formid" value="fdescr.form.descr"/>
  <param name="form_type" value="combo"/>
  <param name="form_source" value="gti.VIEW_Loc_woscope"/>
  <param name="form_caption" value="es=Localizacion|en=Location"/>
  <param name="source_filter_field" value="fk"/>

  <param ord="2" name="structid" value="gti.am.cassetmanag3"/>
</struct>
```

Y su apariencia en la aplicación es:



The image shows two filter sections in a web application. The top section is titled "Ambito" and contains a label "Descripcion", a dropdown menu with "Sevilla" selected, and a "Filtrar" button. The bottom section is titled "Localizacion" and contains a label "Descripcion", a dropdown menu with "Pabellon de Mexico - Planta baja - Recepcion" selected, and a "Filtrar" button.

Ilustración 116: Interfaz de usuario final Gestion de activos. Combos

Además, está compuesto por un formulario con los activos que cumplen los requisitos del filtrado de ámbito y localización. Su código ASL correspondiente es:

```
<struct id="gti.am.cassetmanag3" type="relation">
  <param name="form_split" value="rows=380,*"/>

  <param name="formid" value="gti.am.fasset2"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1:2"/>
  <param name="navigator_size" value="6"/>
  <param name="form_caption" value="es=Activo|en=Asset"/>
  <param name="form_subtype" value="tab"/>
  <param name="form_tabs" value="es=Informacion|en=Information:1,2,3,4,5,6,7,8,9;
es=Detalles|en=Details:10,11,12,13,14,15;es=Foto|en=Photo:16"/>
  <param name="button_insert" value="0"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="fields_modify[0]" value="widget#combonull#6:7:9"/>
  <param name="fields_modify[1]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Gin_Scope where uid=%userid);;#7"/>
  <param name="source_filter_field" value="fkloc"/>

  <param ord="2" name="structid" value="gti.am.cassetmanag4"/>
</struct>
```

Y su apariencia en la aplicación es:



Ilustración 117: Interfaz de usuario final Gestión de activos. Formulario

Finalmente, esta interfaz contiene seis pestañas relacionadas con el activo seleccionado: Propiedades, Propiedades Modelo, Recursos Hijo, Histórico, Traza y Adjuntos:

- Pestaña Propiedades: contiene las propiedades relacionadas con el tipo de activo. Su código ASL es el siguiente:

```
<param ord="1" name="formid" value="gti.am.fassetattr"/>
<param ord="1" name="tab_name" value="es=Propiedades|en=Properties"/>
<param ord="1" name="form_type" value="table"/>
<param ord="1" name="fields_hidden" value="2"/>
<param ord="1" name="button_insert" value="0"/>
<param ord="1" name="button_update" value="1"/>
<param ord="1" name="button_delete" value="0"/>
<param ord="1" name="source_filter_field" value="fk"/>
```

Y su despliegue en la aplicación:

Ilustración 118: Interfaz de usuario final Gestión de activos. Pestaña Propiedades

- Pestaña Propiedades Modelo: contiene las propiedades relacionadas con el tipo de modelo. Su código ASL es el siguiente:

```
<param ord="2" name="formid" value="gti.am.fmodelattr"/>
<param ord="2" name="form_source" value="am.VIEW_assetmodelattr"/>
<param ord="2" name="tab_name" value="es=Propiedades Modelo|en=Model properties"/>
<param ord="2" name="form_type" value="table"/>
<param ord="2" name="fields_readonly" value="1-3"/>
<param ord="2" name="button_data" value="0"/>
<param ord="2" name="button_misc" value="0"/>
<param ord="2" name="source_filter_field" value="fk"/>
```

Y su despliegue en la aplicación:

Propiedades	Propiedades Modelo	Recursos Hijo	Historico	Traza	Adjuntos
Atributos					
Nombre	Valor	Unidad			
Num. Vías	123	Uint			
Tipo	Stereo	String			
Potencia	100	Uint			
Memoria	12.5	GB			
Oscilador	125	Tb			
Fecha salida mercado	2005-01-31	Date			

Ilustración 119: Interfaz de usuario final Gestión de activos. Pestaña Propiedades Modelo

- Pestaña Recursos Hijo: contiene aquellos activos cuyo activo padre es el actual. Su código ASL correspondiente es:

```
<param ord="3" name="formid" value="gti.am.fasset1"/>
<param ord="3" name="tab_name" value="es=Recursos Hijo|en=Child Resources"/>
<param ord="3" name="form_type" value="table"/>
<param ord="3" name="fields_hidden" value="1:2"/>
<param ord="3" name="button_insert" value="0"/>
<param ord="3" name="button_update" value="1"/>
<param ord="3" name="button_delete" value="0"/>
<param ord="3" name="fields_remove" value="7:8:9:10:11:12:13:15"/>
<param ord="3" name="source_filter_field" value="fkasset"/>
```

Y su apariencia en el aplicación es:

Propiedades	Propiedades Modelo	Recursos Hijo	Historico	Traza	Adjuntos
Activo					
Marca y Modelo	Responsable	NS	IDR	Ed.Activos	
Mark 2 - Model 1	admininv	ns2	id2	Ver	Modificar
1					

Ilustración 120: Interfaz de usuario final Gestión de activos. Pestaña Recursos hijo

- Pestaña Histórico: contiene las anotaciones pertinentes al activo seleccionado. Su código ASL es:

```
<param ord="4" name="tab_name" value="es=Historico|en=History"/>
<param ord="4" name="formid" value="gti.am.fentry"/>
<param ord="4" name="form_type" value="table"/>
<param ord="4" name="fields_readonly" value="1:2"/>
<param ord="4" name="button_insert" value="1"/>
<param ord="4" name="button_update" value="0"/>
<param ord="4" name="button_delete" value="0"/>
<param ord="4" name="source_filter_field" value="fkasset"/>
```

Y su apariencia en la aplicación es:



Historico		
Fecha y Hora	Usuario	Anotacion
2021-01-08 17:44:22	gestinv	cable roto
2021-04-09 18:06:45	gestinv	

Añadir

Ilustración 121: Interfaz de usuario final Gestión de activos. Pestaña Histórico

- Pestaña Traza: contiene un historial de todas las modificaciones del activo seleccionado. Su código ASL es:

```
<param ord="5" name="tab_name" value="es=Traza|en=Trace"/>
<param ord="5" name="formid" value="gti.am.fhistory"/>
<param ord="5" name="form_type" value="table"/>
<param ord="5" name="fields_readonly" value="1:2"/>
<param ord="5" name="button_insert" value="0"/>
<param ord="5" name="button_update" value="0"/>
<param ord="5" name="button_delete" value="0"/>
<param ord="5" name="source_filter_field" value="fkasset"/>
```

Y su apariencia en la aplicación es:

Traza			
Fecha y Hora	Usuario	Tipo	Descripcion
2020-12-21 18:25:02	gestinv	Insercion	Modelo: Mark 2 - Model 1 SN: ID: Responsable: admininv Localizacion: Sevilla - Pabellon de Mexico - Planta baja - Recepcion
2020-12-21 18:35:06	gestinv	Modificacion	SN: ns6
2021-01-28 15:17:51	gestinv	Modificacion	OU: Universidad de Sevilla - Escuela Internacional de Doctorado - Unidad de alumnos
2021-03-21 10:30:29	gestinv	Modificacion	Responsable: gestinv

Ilustración 122: Interfaz de usuario final Gestión de activos. Pestaña Traza

- Pestaña Adjuntos: contiene los adjuntos relativos al activo seleccionado. Su código ASL es:

```
<param ord="6" name="tab_name" value="es=Adjuntos|en=Attachments"/>
<param ord="6" name="formid" value="attach.form.catattach"/>
<param ord="6" name="form_type" value="table"/>
<param ord="6" name="fields_modify[0]" value="defvalue#9#3"/>
<param ord="6" name="fields_remove" value="5:7"/>
<param ord="6" name="fields_hidden" value="3:4:5"/>
<param ord="6" name="button_data" value="1"/>
<param ord="6" name="source_filter_field" value="fk"/>
<param ord="6" name="form_modify[0]" value="filter#fkcat=9"/>
```

Y su apariencia en la aplicación es:

Propiedades	Propiedades Modelo	Recursos Hijo	Historico	Traza	Adjuntos
-------------	--------------------	---------------	-----------	-------	----------

Adjuntos	
Descripcion	Archivo
Activo 6	Descargar <input type="checkbox"/> Borrar Seleccionar archivo Ningún archi... seleccionado
	Descargar <input type="checkbox"/> Borrar Seleccionar archivo Ningún archi... seleccionado

Ilustración 123: Interfaz de usuario final Gestión de activos. Pestaña Adjuntos

4.4.1.3.7 Edición de activos

Esta interfaz es accedida mediante un botón (enlace) desde las interfaces: *Gestión de entradas*, *Gestión de salidas*, *Gestión de inventario*, *Gestión de inventario por tipo de recurso* y *Gestión de activos*. El código ASL correspondiente al Modelo de Presentación para la interfaz *Edición de activos* es el siguiente:

```
<struct id="gti.am.cassetedit1" type="relation">
  <param name="form_split" value="rows=360,*"/>

  <param name="formid" value="gti.am.fasset2"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="none"/>
  <param name="source_row" value="0"/>
  <param name="form_subtype" value="tab"/>
  <param name="form_tabs" value="es=Informacion|en=Information:1,2,3,4,5,6,7,8,9;
es=Detalles|en=Details:10,11,12,13,14,15;es=Foto|en=Photo:16"/>
  <param name="form_caption" value="es=Activo|en=Asset"/>
  <param name="fields_modify[0]" value="widget#combonull#6:7:9"/>
  <param name="fields_modify[1]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Gin_Scope where uid=%userid);;#7"/>
  <param name="button_insert" value="0"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="source_filter_field" value="pk" />

  <param ord="2" name="structid" value="gti.am.cassetedit2"/>
</struct>

<struct id="gti.am.cassetedit2" type="tab">
  <param name="form_split" value="rows=70,*"/>

  <param ord="1" name="formid" value="gti.am.fassetattr"/>
  <param ord="1" name="tab_name" value="es=Propiedades|en=Properties"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_hidden" value="2"/>
  <param ord="1" name="button_insert" value="0"/>
  <param ord="1" name="button_update" value="1"/>
  <param ord="1" name="button_delete" value="0"/>
  <param ord="1" name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="gti.am.fmodelattr"/>
  <param ord="2" name="form_source" value="am.VIEW_assetmodelattr"/>
  <param ord="2" name="tab_name" value="es=Propiedades Modelo|en=Model properties"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_readonly" value="1-3"/>
</struct>
```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```

<param ord="2" name="button_data" value="0"/>
<param ord="2" name="button_misc" value="0"/>
<param ord="2" name="source_filter_field" value="fk"/>

<param ord="3" name="formid" value="gti.am.fasset1"/>
<param ord="3" name="tab_name" value="es=Recursos Hijo|en=Child Resources"/>
<param ord="3" name="form_type" value="table"/>
<param ord="3" name="fields_hidden" value="1:2"/>
<param ord="3" name="button_insert" value="0"/>
<param ord="3" name="button_update" value="1"/>
<param ord="3" name="button_delete" value="0"/>
<param ord="3" name="fields_remove" value="7:8:9:10:11:12:13:15"/>
<param ord="3" name="source_filter_field" value="fkasset"/>

<param ord="4" name="tab_name" value="es=Historico|en=History"/>
<param ord="4" name="formid" value="gti.am.fentry"/>
<param ord="4" name="form_type" value="table"/>
<param ord="4" name="fields_readonly" value="1:2"/>
<param ord="4" name="button_insert" value="1"/>
<param ord="4" name="button_update" value="0"/>
<param ord="4" name="button_delete" value="0"/>
<param ord="4" name="source_filter_field" value="fkasset"/>

<param ord="5" name="tab_name" value="es=Traza|en=Trace"/>
<param ord="5" name="formid" value="gti.am.fhistory"/>
<param ord="5" name="form_type" value="table"/>
<param ord="5" name="fields_readonly" value="1:2"/>
<param ord="5" name="button_insert" value="0"/>
<param ord="5" name="button_update" value="0"/>
<param ord="5" name="button_delete" value="0"/>
<param ord="5" name="source_filter_field" value="fkasset"/>

<param ord="6" name="tab_name" value="es=Adjuntos|en=Attachments"/>
<param ord="6" name="formid" value="attach.form.catattach"/>
<param ord="6" name="form_type" value="table"/>
<param ord="6" name="fields_modify[0]" value="defvalue#9#3"/> <!-- Asignando categoría -->
<param ord="6" name="fields_remove" value="5:7"/>
<param ord="6" name="fields_hidden" value="3:4:6"/>
<param ord="6" name="button_data" value="1"/>
<param ord="6" name="source_filter_field" value="fk"/>
<param ord="6" name="form_modify[0]" value="filter#fkcat=9"/>
</struct>

```

Y su interfaz final en la aplicación es la siguiente:

Activo

Informacion	Detalles	Foto
ID	6	
NS	<input type="text" value="ns6"/>	
IDR	<input type="text" value="345"/>	
Estado	<input type="text" value="Activo"/>	
Modelo	Vieta - AM22 - Omitipo-Amplificador	
OU	<input type="text" value="US - Escuela Internacional de Doctorado - Unidad de alumnos"/>	
Localizacion	<input type="text" value="Sevilla - Pabellon de Mexico - Planta baja - Recepcion"/>	
Responsable	<input type="text" value="admininv - Álvaro Marín Sánchez"/>	
Activo padre	<input type="text" value="ns4"/>	

Atributos			
Nombre	Valor	Unidad	
Tamano	<input type="text"/>	<input type="text" value="Integer"/>	<input type="button" value="Modificar"/>

Ilustración 124: Interfaz de usuario final Edición de activos

La pestaña Detalles del formulario Activo, contiene información adicional sobre el activo actual:

Activo

Informacion	Detalles	Foto
Descripcion	<input type="text"/>	
Notas	<div style="border: 1px solid gray; height: 40px;"></div>	
Nombre equipo	<input type="text"/>	
Instante creacion	2020-12-21 18:25	
Instante modificacion	2021-04-09 17:49	
Usuario dio de alta	admininv - Álvaro Marín Sánchez	

Ilustración 125: Interfaz de usuario final Edición de activos. Pestaña Detalles

La pestaña Foto del formulario Activo, permite adjuntar una imagen sobre el activo actual:



Ilustración 126: Interfaz de usuario final Edición de activos. Pestaña Foto

Esta interfaz contiene seis pestañas: Propiedades, Propiedades Modelo, Recursos Hijo, Histórico, Traza y Adjuntos. Estas están ya descritas en el apartado 4.3.6 de este mismo documento.

4.4.1.3.8 Estadísticas de tipos de activos por sede

El código ASL correspondiente al Modelo de Presentación para la interfaz *Estadísticas de tipos de activos por sede* es el siguiente:

```
<struct id="gti.am.cgraphtype" type="relation">
  <param name="form_split" value="rows=100,*"/>

  <param name="formid" value="fdescr.form.descr"/>
  <param name="form_type" value="filter-where"/>
  <param name="form_source" value="gti.VIEW_Site" />
  <param name="form_caption" value="es=Sede |en=Site" />
  <param name="fields_modify[0]" value="search#DATA:gti.VIEW_Site;fk in (select fkscope from
  am.REL_Gin_Scope where uid=%userid);;#1"/>
  <param name="fields_modify[1]" value="caption#es=Sede |en=Site#1" />
  <param name="fields_modify[2]" value="len#0#1" />
  <param name="form_filterw_tpl" value="pk=%values[1]" />
  <param name="button_misc" value="0"/>
  <param name="fields_remove" value="2" />

  <param ord="2" name="formid" value="gti.am.fgraphtype"/>
  <param ord="2" name="form_type" value="graph"/>
  <param ord="2" name="graph_width" value="520"/>
  <param ord="2" name="graph_height" value="180"/>
  <param ord="2" name="button_data" value="0"/>
  <param ord="2" name="button_misc" value="0"/>
  <param ord="2" name="graph_table" value="1"/>
  <param ord="2" name="__nofill" value="1"/>
</struct>
```

Y su interfaz final en la aplicación es la siguiente:

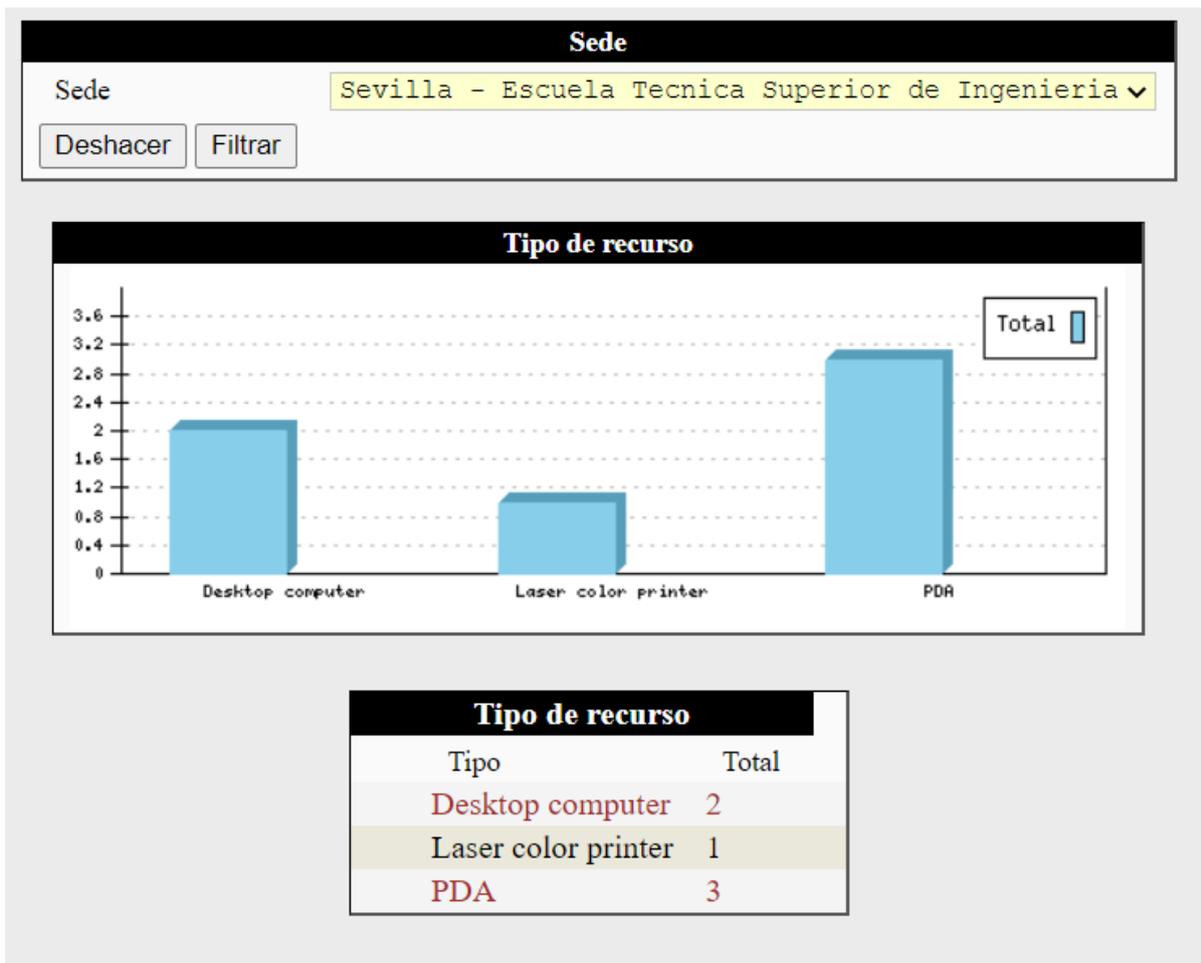


Ilustración 127: Interfaz de usuario final Estadísticas de tipos de activos por sede

4.4.1.3.9 Estadísticas de estados y marcas de activos por sede y tipo

El código ASL correspondiente al Modelo de Presentación para la interfaz *Estadísticas de estados y marcas de activos por sede y tipo* es el siguiente:

```
<struct id="gti.am.cgraphstatusmark1" type="relation">
  <param name="form_split" value="rows=150,*"/>

  <param name="formid" value="gti.am.fsitetype"/>
  <param name="form_type" value="filter-where"/>
  <param name="form_filterw_tpl" value="sitepk=%values[1] AND typepk=%values[2]" />
  <param name="fields_modify[0]" value="search#DATA:gti.VIEW_Site;fk in (select fkscope from
  am.REL_Gin_Scope where uid=%userid);;#1"/>
  <param name="button_misc" value="0"/>
  <param name="form_filterw_nexus" value="and"/>

  <param ord="2" name="structid" value="gti.am.cgraphstatusmark2"/>
</struct>

<struct id="gti.am.cgraphstatusmark2" type="split">
  <param name="form_split" value="cols=50%,*"/>

  <param name="formid" value="gti.am.fgraphstatus"/>
  <param name="form_type" value="graph"/>
  <param name="graph_width" value="520"/>
  <param name="graph_height" value="180"/>
  <param name="button_data" value="0"/>
  <param name="button_misc" value="0"/>
```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```

<param name="graph_table" value="1"/>
<param name="__nofill" value="1"/>

<param ord="2" name="formid" value="gti.am.fgraphmark"/>
<param ord="2" name="form_type" value="graph"/>
<param ord="2" name="graph_width" value="520"/>
<param ord="2" name="graph_height" value="180"/>
<param ord="2" name="button_data" value="0"/>
<param ord="2" name="button_misc" value="0"/>
<param ord="2" name="graph_table" value="1"/>
<param ord="2" name="__nofill" value="1"/>
</struct>
    
```

Y su interfaz final en la aplicación es la siguiente:



Ilustración 128: Interfaz de usuario final Estadísticas de estados y marcas de activos por sede y tipo

4.4.1.4 Técnico avanzado

Las funcionalidades a las que tiene acceso un usuario con rol *técnico avanzado* son las siguientes:

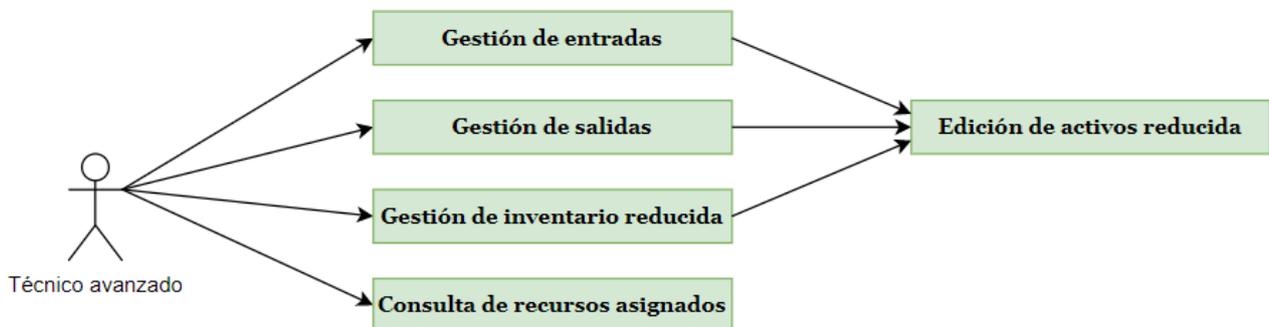


Ilustración 129: Esquema menú Técnico avanzado

El código ASL correspondiente al Modelo de Diálogo del técnico avanzado es el siguiente:



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```

<param name="button_misc" value="0"/>

<param ord="2" name="structid" value="gti.am.cinputmanagate2"/>
</struct>

<struct id="gti.am.cinputmanagate2" type="relation">
  <param name="form_split" value="rows=395,*"/>

  <param name="formid" value="gti.am.finput"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="0:1"/>
  <param name="navigator_size" value="6"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="1"/>
  <param name="button_misc" value="0"/>
  <param name="fields_readonly" value="1"/>
  <param name="form_tabs" value="es=Informacion|en=Information:1,2,3,4,5,6,7,8,9,13,10;es=Notas|en=Re-
marks:11;es=Detalles|en=Details:12"/>
  <param name="form_subtype" value="tab"/>
  <param name="fields_modify[0]" value="canbenull#Y#1:9"/>
  <param name="fields_modify[1]" value="widget#combonull#7-9"/>
  <param name="fields_modify[2]" value="attr#l#3"/>
  <param name="form_modify[0]" value="filter#dst in (select pk from gti.VIEW_Loc where scopepk in (select
fkscope from am.REL_Ate_Scope where uid=%userid))"/>
  <param name="fields_modify[3]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Ate_Scope where uid=%userid);;#10"/>
  <param name="fields_modify[5]" value="defvalue#%username#6" />
  <param name="__nofill" value="1"/>

  <param ord="2" name="structid" value="gti.am.cinputmanagate3"/>
</struct>

<struct id="gti.am.cinputmanagate3" type="tab">
  <param name="form_split" value="rows=70,*"/>

  <param ord="1" name="formid" value="gti.am.fassetADG"/>
  <param ord="1" name="tab_name" value="es=Activos|en=Assets"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_modify[0]" value="wdgparam#struct.php?structid=gti.am.casseteditre-
date1&source_filter_field=pk&source_filter_value=;_blank;;Ver#4"/>
  <param ord="1" name="form_modify[0]" value="beforeupdate#gti.am.actns"/>
  <param ord="1" name="form_modify[1]" value="beforeinsert#gti.am.actns"/>
  <param ord="1" name="form_modify[2]" value="afterinsert#gti.am.actinput"/>
  <param ord="1" name="fields_modify[1]" value="attr#R#1-3" />
  <param ord="1" name="button_insert" value="0"/>
  <param ord="1" name="button_update" value="0"/>
  <param ord="1" name="button_delete" value="1"/>
  <param ord="1" name="source_filter_where" value="pk in (select fkasset from am.REL_Input_Asset where
fkinput=%filtervalue)" />

  <param ord="2" name="tab_name" value="es=Resumen|en=Summary"/>
  <param ord="2" name="formid" value="gti.am.finputsummary"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_hidden" value="3"/>
  <param ord="2" name="button_data" value="0"/>
  <param ord="2" name="source_filter_field" value="fkinput"/>

```




4 Construcción de los módulos de activos, adquisiciones y común de GTI

```

<param name="form_subtype" value="tab"/>
<param name="fields_modify[0]" value="canbenull#Y#1:8"/>
<param name="fields_modify[1]" value="widget#combonull#8"/>
<param name="form_modify[0]" value="filter#src in (select pk from gti.VIEW_Loc where scopepk in (select
fkscope from am.REL_Ate_Scope where uid=%userid))"/>
<param name="fields_modify[2]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Ate_Scope where uid=%userid);#7"/>
<param name="fields_modify[3]" value="defvalue#%username#5" />
<param name="__nofill" value="1"/>

<param ord="2" name="structid" value="gti.am.coutputmanagate3"/>
</struct>

<struct id="gti.am.coutputmanagate3" type="tab">
  <param name="form_split" value="rows=70,*"/>

  <param ord="1" name="formid" value="gti.am.foutassetelem"/>
  <param ord="1" name="tab_name" value="es=Activos|en=Assets"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_hidden" value="3"/>
  <param ord="1" name="button_insert" value="1"/>
  <param ord="1" name="button_update" value="0"/>
  <param ord="1" name="button_delete" value="1"/>
  <param ord="1" name="fields_modify[0]" value="wdgparam#struct.php?structid=gti.am.casseteditre-
date1&source_filter_value=;_blank;;Ver#2"/>
  <param ord="1" name="source_filter_field" value="fkoutput"/>

  <param ord="2" name="tab_name" value="es=Resumen|en=Summary"/>
  <param ord="2" name="formid" value="gti.am.foutputsummary"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_hidden" value="3"/>
  <param ord="2" name="button_data" value="0"/>
  <param ord="2" name="source_filter_field" value="fkoutput"/>

  <param ord="3" name="tab_name" value="es=Adjuntos|en=Attachments"/>
  <param ord="3" name="formid" value="attach.form.catattach"/>
  <param ord="3" name="form_type" value="table"/>
  <param ord="3" name="fields_modify[0]" value="defvalue#11#3"/> <!-- Asignando categoría -->
  <param ord="3" name="fields_remove" value="5:7"/>
  <param ord="3" name="fields_hidden" value="3:4:6"/>
  <param ord="3" name="button_data" value="1"/>
  <param ord="3" name="source_filter_field" value="fk"/>
  <param ord="3" name="form_modify[0]" value="filter#fkcat=11"/>
</struct>

```

4.4.1.4.3 Gestión de inventario reducida

Esta interfaz es parecida a la descrita en el apartado 4.3.4 de este documento, con la diferencia de que algunos campos serán de modo lectura y que los permisos se van a filtrar esta vez para el técnico avanzado. Además, el botón “Ver” que muestra la interfaz de *Edición de activos*, esta vez será para la *Edición de activos reducida*. Esto se refleja en el código ASL de la siguiente forma (resaltado en negrita):

```

<struct id="gti.am.cassetmanagredate" type="relation">
  <param name="form_split" value="rows=25%,*"/>

  <param name="formid" value="gti.am.fasset1"/>
  <param name="tab_name" value="Filtros basicos"/>

```




Activo	
Localizacion	OU
Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE	US - Escuela Internacional de Doctorado - Unidad de alumnos
Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE	
Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE	US - Escuela Internacional de Doctorado - Unidad de tesis
Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE	
Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE	

Marca y Modelo	Responsable	NS	IDR	Ed.Activos	
Mark 2 - Model 2	gestadqui	ns7	id7	Ver	Modificar
Mark 2 - Model 2	gestadqui	ns8	id8	Ver	Modificar
Mark 3 - Model 1	gestadqui	ns10	id10	Ver	Modificar
Mark 3 - Model 1	gestadqui	ns11	id11	Ver	Modificar
Mark 3 - Model 2	gestadqui	ns12	id12	Ver	Modificar

Ilustración 131: Interfaz de usuario final Gestión de inventario reducida. Formulario

4.4.1.4.4 Edición de activos reducida

Esta interfaz es accedida mediante un botón (enlace) desde las interfaces: *Gestión de entradas*, *Gestión de salidas* y *Gestión de inventario reducida* del rol de técnico avanzado. Esta interfaz es parecida a la descrita en el apartado 4.3.7 de este documento, con la diferencia de que algunos campos serán de modo lectura (se impide la modificación del tipo, modelo, número de serie e identificador de un activo). Esto se refleja en el código ASL de la siguiente forma (resaltado en negrita):

```
<struct id="gti.am.casseteditredate1" type="relation">
  <param name="form_split" value="rows=360,*"/>

  <param name="formid" value="gti.am.fasset2"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="none"/>
  <param name="source_row" value="0"/>
  <param name="form_subtype" value="tab"/>
```

```
<param name="form_tabs" value="es=Informacion|en=Information:1,2,3,4,5,6,7,8,9;es=Detalles|en=De-
tails:10,11,12,13,14,15;es=Foto|en=Photo:16"/>
<param name="form_caption" value="es=Activo|en=Asset"/>
<param name="fields_readonly" value="1" />
<param name="fields_modify[0]" value="widget#combonull#6:7:9"/>
<param name="fields_modify[1]" value="search#DATA:gti.VIEW_Loc;scopepk in (select fkscope from
am.REL_Ate_Scope where uid=%userid);#7"/>
<param name="button_insert" value="0"/>
<param name="button_update" value="1"/>
<param name="button_delete" value="0"/>
<param name="button_misc" value="0"/>
<param name="source_filter_field" value="pk" />
<param name="fields_readonly" value="1:2:3:4:5"/>
```

```
<param ord="2" name="structid" value="gti.am.casseteditredate2"/>
</struct>
```

```
<struct id="gti.am.casseteditredate2" type="tab">
  <param name="form_split" value="rows=70,*"/>
```

```
  <param ord="1" name="formid" value="gti.am.fassetattr"/>
  <param ord="1" name="tab_name" value="es=Propiedades|en=Properties"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="button_insert" value="0"/>
  <param ord="1" name="button_update" value="1"/>
  <param ord="1" name="button_delete" value="0"/>
  <param ord="1" name="source_filter_field" value="fk"/>
```

```
  <param ord="2" name="formid" value="gti.am.fmodelattr"/>
  <param ord="2" name="form_source" value="am.VIEW_assetmodelattr"/>
  <param ord="2" name="tab_name" value="es=Propiedades Modelo|en=Model properties"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_readonly" value="1-3"/>
  <param ord="2" name="button_data" value="0"/>
  <param ord="2" name="button_misc" value="0"/>
  <param ord="2" name="source_filter_field" value="fk"/>
```

```
  <param ord="3" name="formid" value="gti.am.fasset1"/>
  <param ord="3" name="tab_name" value="es=Recursos Hijo|en=Child Resources"/>
  <param ord="3" name="form_type" value="table"/>
  <param ord="3" name="fields_hidden" value="1:2"/>
  <param ord="3" name="button_insert" value="0"/>
  <param ord="3" name="button_update" value="1"/>
  <param ord="3" name="button_delete" value="0"/>
  <param ord="3" name="fields_readonly" value="3:5:6" />
  <param ord="3" name="fields_remove" value="7:8:9:10:11:12:13:15"/>
  <param ord="3" name="source_filter_field" value="fkasset"/>
  <param ord="3" name="fields_modify[0]" value="wdgparam#struct.php?structid=gti.am.casseteditre-
date1&source_filter_field=pk&source_filter_value=;_blank;;Ver#14"/>
```

```
  <param ord="4" name="tab_name" value="es=Historico|en=History"/>
  <param ord="4" name="formid" value="gti.am.fentry"/>
  <param ord="4" name="form_type" value="table"/>
  <param ord="4" name="fields_readonly" value="1:2"/>
  <param ord="4" name="button_insert" value="1"/>
  <param ord="4" name="button_update" value="0"/>
  <param ord="4" name="button_delete" value="0"/>
  <param ord="4" name="source_filter_field" value="fkasset"/>
```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```

<param ord="5" name="tab_name" value="es=Traza|en=Trace"/>
<param ord="5" name="formid" value="gti.am.fhistory"/>
<param ord="5" name="form_type" value="table"/>
<param ord="5" name="fields_readonly" value="1:2"/>
<param ord="5" name="button_insert" value="0"/>
<param ord="5" name="button_update" value="0"/>
<param ord="5" name="button_delete" value="0"/>
<param ord="5" name="source_filter_field" value="fkasset"/>

<param ord="6" name="tab_name" value="es=Adjuntos|en=Attachments"/>
<param ord="6" name="formid" value="attach.form.catattach"/>
<param ord="6" name="form_type" value="table"/>
<param ord="6" name="fields_modify[0]" value="defvalue#9#3"/> <!-- Asignando categoría -->
<param ord="6" name="fields_remove" value="5:7"/>
<param ord="6" name="fields_hidden" value="3:4:6"/>
<param ord="6" name="button_data" value="1"/>
<param ord="6" name="source_filter_field" value="fk"/>
<param ord="6" name="form_modify[0]" value="filter#fkcat=9"/>
</struct>
    
```

Y los cambios en su interfaz se muestran de la siguiente forma en la aplicación:

Ilustración 132: Interfaz de usuario final Edición de activos reducida. Formulario

Y en su pestaña Recursos Hijo:

Ilustración 133: Interfaz de usuario final Edición de activos reducida. Pestaña Recursos hijo

4.4.1.5 Técnico

Las funcionalidades a las que tiene acceso un usuario con rol *técnico* son las siguientes:

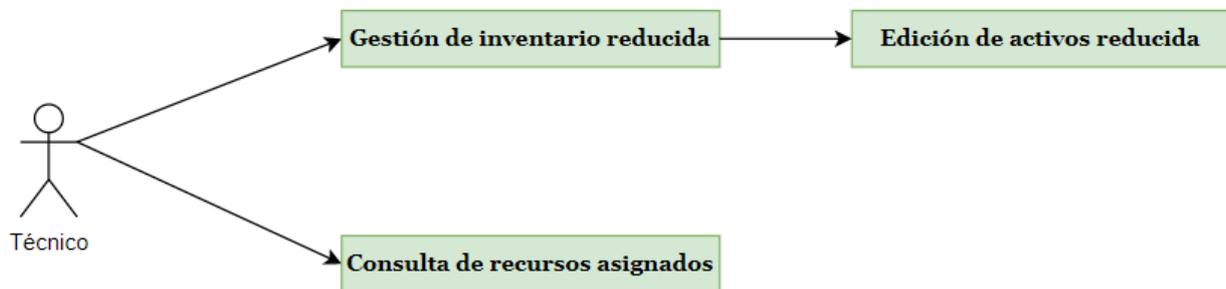


Ilustración 134: Esquema menú Técnico

El código ASL correspondiente al Modelo de Diálogo del técnico es el siguiente:

```
<main id="gti.am.mtec" caption="es=AM.Tecnico|en=AM.Technician">

  <menu caption="es=Gestion tecnico|en=Technician Management" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Gestion de inventario reducida|en=Reduced Asset Management"
      call="gti.am.cassetmanagredtec"/>
  </menu>

  <menu caption="es=Recursos asignados|en=Assigned Resources" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Consulta de recursos asignados|en=Assigned Resources Query" call="gti.am.cassignre-
sourcequery1"/>
  </menu>

  <menu caption="Misc" tooltip="Otras cosas">
    <option caption="es=Sobre este programa|en=About" call="gti.am.cmodinfo" />
    <option caption="es=Ayuda|en=Help" url="http://wayne.us.es/dokuwiki/doku.php?
id=wayne:help:app:gti:am:menu:tecnico" />
    <option caption="es=Sobre WAINE|en=About WAINE" url="http://www.wayne.org" />
    <option caption="es=Salir|en=Logout" url="logout.php" />
  </menu>

</main>
```

En los siguientes apartados se describirán cada una de las funcionalidades.

4.4.1.5.1 Gestión de inventario reducida

Esta interfaz es idéntica a la descrita en el apartado 4.4.3 de este documento, con la diferencia de que los permisos se van a filtrar esta vez para el técnico.

4.4.1.5.2 Edición de activos reducida

Esta interfaz es accedida mediante un botón (enlace) desde la interfaz *Gestión de inventario reducida* del rol de técnico. Además, es idéntica a la descrita en el apartado 4.4.4 de este documento, con la diferencia de que los permisos se van a filtrar esta vez para el técnico.



4.4.1.6 Usuario regular

Las funcionalidades a las que tiene acceso un usuario con rol *usuario regular* serán únicamente las de consultar los recursos asignados. Esta interfaz ya se encuentra detallada en el apartado 4.1.1 de este mismo documento.

El código ASL correspondiente al Modelo de Diálogo del usuario regular es el siguiente:

```
<main id="gti.am.muser" caption="es=AM.Usuario regular|en=AM.Regular User">

  <menu caption="es=Gestion usuario regular|en=Regular User Management"
  tooltip="Procesos, actividades y Tareas">
    <option caption="es=Consulta de recursos asignados|en=Assigned Resources Query"
    call="gti.am.cassignresourcequery1"/>
  </menu>

  <menu caption="Misc" tooltip="Otras cosas">
    <option caption="es=Sobre este programa|en=About" call="gti.am.cmodinfo" />
    <option caption="es=Ayuda|en=Help" url="http://waine.us.es/dokuwiki/doku.php?
    id=waine:help:app:gti:am:menu:usuario_regular" />
    <option caption="es=Sobre WAINE|en=About WAINE" url="http://www.waine.org" />
    <option caption="es=Salir|en=Logout" url="logout.php" />
  </menu>

</main>
```

4.4.1.7 Usuario sin acceso

Las siguientes interfaces son aquellas que son accedidas por usuarios que no tienen los permisos necesarios para acceder al módulo de forma autenticada. Sin embargo, se les permite obtener información general sobre los recursos, las entradas y las salidas. Las funcionalidades de un usuario sin acceso son las siguientes:

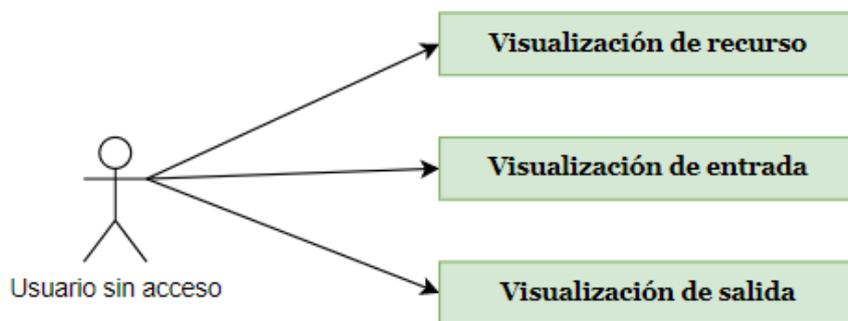


Ilustración 135: Esquema menú Usuario sin acceso

El código ASL correspondiente al Modelo de Diálogo del usuario sin acceso es el siguiente:

```
<main id="gti.am.muser_na" caption="es=AM.Usuario sin acceso|en=AM.No Access User">

  <menu caption="es=Gestion usuario sin acceso|en=No Access User Management"
  tooltip="Procesos, actividades y Tareas">
    <option caption="es=Visualizacion de recurso|en=Asset Visualization" call="gti.am.cresourceview1"/>
    <option caption="es=Visualizacion de adquisicion|en=Purchase Visualization" call="gti.pp.cpurchaseview"/>
    <option caption="es=Visualizacion de entrada|en=Input Visualization" call="gti.am.cinputview"/>
    <option caption="es=Visualizacion de salida|en=Output Visualization" call="gti.am.coutputview"/>
  </menu>
```

```

<menu caption="Misc" tooltip="Otras cosas">
  <option caption="es=Sobre este programa|en>About" call="gti.am.cmodinfo" />
  <option caption="es=Ayuda|en=Help" url="http://waine.us.es/dokuwiki/doku.php?id=waine:help:app:gti:am:menu:usuario_sin_acceso" />
  <option caption="es=Sobre WAINE|en>About WAINE" url="http://www.waine.org" />
  <option caption="es=Salir|en=Logout" url="logout.php" />
</menu>

```

```
</main>
```

En los siguientes apartados se describirán cada una de las funcionalidades.

4.4.1.7.1 Visualización de recurso

El código ASL correspondiente al Modelo de Presentación para la interfaz *Visualización de recurso* es el siguiente:

```

<struct id="gti.am.cresourceview1" type="relation">
  <param name="form_split" value="rows=30%,*"/>

  <param name="formid" value="gti.am.fasset2"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="none"/>
  <param name="source_row" value="0"/>
  <param name="form_caption" value="es=Activo|en=Asset"/>
  <param name="fields_remove" value="6-16"/>
  <param name="fields_readonly" value="1-5" />
  <param name="button_action" value="0"/>
  <param name="button_insert" value="0"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="source_filter_field" value="pk" />

  <param ord="2" name="structid" value="gti.am.cresourceview2"/>
</struct>

<struct id="gti.am.cresourceview2" type="tab">
  <param name="form_split" value="rows=70,*"/>

  <param ord="1" name="formid" value="gti.am.fassetattr"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="tab_name" value="es=Propiedades|en=Properties"/>
  <param ord="1" name="form_caption" value="es=Propiedades|en=Properties"/>
  <param ord="1" name="fields_readonly" value="1-3" />
  <param ord="1" name="button_insert" value="0"/>
  <param ord="1" name="button_update" value="0"/>
  <param ord="1" name="button_delete" value="0"/>
  <param ord="1" name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="gti.am.fmodelattr"/>
  <param ord="2" name="form_source" value="am.VIEW_assetmodelattr"/>
  <param ord="2" name="tab_name" value="es=Propiedades Modelo|en=Model properties"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_readonly" value="1-3"/>
  <param ord="2" name="button_data" value="0"/>
  <param ord="2" name="button_misc" value="0"/>

```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```
<param ord="2" name="source_filter_field" value="fk"/>
</struct>
```

Y su interfaz final en la aplicación es la siguiente:

Activo	
ID	6
NS	ns6
IDR	345
Estado	Activo
Modelo	Vieta - AM22 - 0mitipo-Amplificador

Propiedades Propiedades Modelo

Propiedades		
Nombre	Valor	Unidad
Tamano		

Ilustración 136: Interfaz de usuario final Visualización de recurso

4.4.1.7.2 Visualización de entrada

El código ASL correspondiente al Modelo de Presentación para la interfaz *Visualización de entrada* es el siguiente:

```
<struct id="gti.am.cinputview" type="form">
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="none"/>
  <param name="source_row" value="0"/>

  <param name="formid" value="gti.am.finput"/>
  <param name="button_insert" value="0"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="fields_remove" value="1:7:8:11:12"/>
  <param name="fields_readonly" value="2-6:9:10" />
  <param name="source_filter_field" value="pk" />
</struct>
```

Y su interfaz final en la aplicación es la siguiente:

Entrada	
Descripcion	input2
Tipo	Adquisicion
Instante	2020-12-21 18:33
Emisor	test
Receptor	gestadqui
Origen	Sevilla - Pabellon de Mexico - Planta baja - Recepcion
Destino	Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE

Ilustración 137: Interfaz de usuario final Visualización de entrada

4.4.1.7.3 Visualización de salida

El código ASL correspondiente al Modelo de Presentación para la interfaz *Visualización de salida* es el siguiente:

```
<struct id="gti.am.coutputview" type="form">
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="none"/>
  <param name="source_row" value="0"/>

  <param name="formid" value="gti.am.foutput"/>
  <param name="button_insert" value="0"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="fields_remove" value="1:9"/>
  <param name="fields_readonly" value="2-8" />
  <param name="source_filter_field" value="pk" />
</struct>
```

Y su interfaz final en la aplicación es la siguiente:

Salida	
Descripcion	output1
Tipo	Interna
Instante	2020-12-21 18:39
Emisor	test
Receptor	test
Origen	Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE
Destino	Córdoba - Facultad de Medicina y Enfermeria - Edificio Sur - Planta Segunda
Responsable salida	root

Ilustración 138: Interfaz de usuario final Visualización de salida



4.4.2 Construcción de las interfaces de usuario módulo gti

En esta sección, se detalla el proceso de construcción de las interfaces de usuario para cada uno de los roles definidos en la aplicación. Esta construcción se ha hecho en base a los diagramas entidad-relación anotados mostrados en el apartado 4 del Documento de Diseño del módulo común.

La implementación de estas interfaces se hizo a través del lenguaje ASL propio del MBUIDE de **WAINE**. En este lenguaje, se definen tres modelos: Modelo de Usuario, Modelo de Presentación y Modelo de Diálogo.

En los siguientes apartados, se incluirá cada uno de los modelos para cada usuario, interfaz y el resultado gráfico en forma de captura de la interfaz de usuario.

4.4.2.1 Administrador del módulo común

Las funcionalidades a las que tiene acceso un usuario con rol *administrador del módulo de común* son las siguientes:

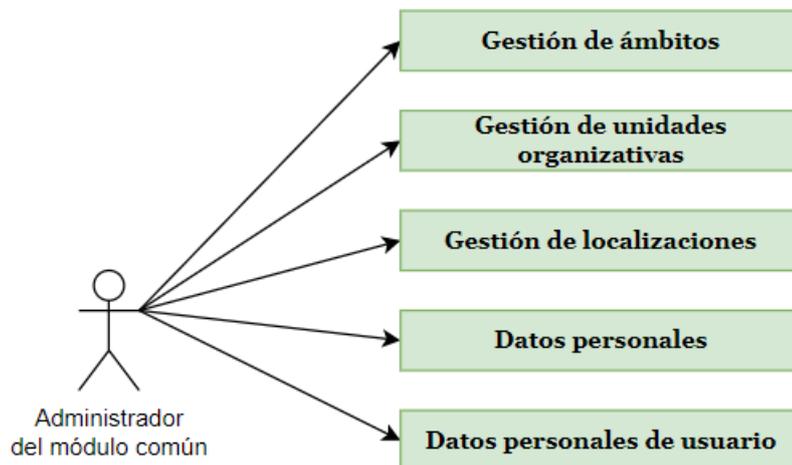


Ilustración 139: Esquema menú Administrador del módulo común

El código ASL correspondiente al Modelo de Diálogo del administrador del módulo común es el siguiente:

```

<main id="gti.am.madmincom" caption="es= AM.Administrador del modulo comun|en= AM.Common Module Administrator">
  <menu caption="es=Gestion modulo comun|en=Common Module Management" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Gestion de ambitos|en=Scope Management" call="gti.cscope"/>
    <option caption="es=Gestion de unidades organizativas|en=Organizative Units Management" call="gti.coul1"/>
    <option caption="es=Gestion de localizaciones|en=Location Management" call="gti.csite1"/>
    <option caption="es=Datos personales|en=People Management" call="percon.struct.edit"/>
    <option caption="es=Datos personales de usuario|en=User Management" call="upercon.struct.edit"/>
  </menu>
  <menu caption="Misc" tooltip="Otras cosas">
    <option caption="es=Sobre este programa|en=About" call="gti.am.cmodinfo" />
    <option caption="es=Ayuda|en=Help"
  
```

```

url="http://waine.us.es/dokuwiki/doku.php?id=waine:help:app:gti:base:menu:administrador_modulo_co-
mun"
/>
<option caption="es=Sobre WAINE|en>About WAINE" url="http://www.waine.org" />
<option caption="es=Salir|en=Logout" url="logout.php" />
</menu>

</main>

```

En los siguientes apartados se describirán cada una de las funcionalidades.

4.4.2.1.1 Gestión de ámbitos

El código ASL correspondiente al Modelo de Presentación para la interfaz *Gestión de ámbitos* es el siguiente:

```

<struct id="gti.cscope" type="form">
  <param name="form_type" value="table"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="gti.Scope"/>
  <param name="form_caption" value="es=Ámbitos|en=Scopes"/>
  <param name="fields_remove" value="3"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="1"/>
</struct>

```

Y su interfaz final en la aplicación es la siguiente:

Ámbitos	
Descripcion	Notas
0-Global	
Cádiz	
Córdoba	
Sevilla	

Ilustración 140: Interfaz de usuario final Gestión de ámbitos

4.4.2.1.2 Gestión de unidades organizativas

El código ASL correspondiente al Modelo de Presentación para la interfaz *Gestión de unidades organizativas* es el siguiente:

```

<struct id="gti.coul1" type="relation">
  <param name="form_split" value="rows=33%,*"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="gti.OUL1"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="6"/>
  <param name="button_data" value="1"/>

```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```

<param name="form_caption" value="es=Unidad Norganizativa N.1|en=Organizative Unit Lvl. 1"/>
<param name="fields_remove" value="3"/>

<param ord="2" name="structid" value="gti.coul2"/>
</struct>

<struct id="gti.coul2" type="relation">
  <param name="form_split" value="rows=50%,*"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="gti.OUL2"/>
  <param name="form_type" value="form"/>
  <param name="form_caption" value="es=Unidad Norganizativa N.2|en=Organizative Unit Lvl. 2"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="6"/>
  <param name="button_data" value="1"/>
  <param name="fields_modify[0]" value="search#DATA:gti.OUL1;;;#3"/>
  <param name="fields_modify[1]" value="caption#es=Unidad Norganizativa N.1|
  en=Organizative Unit Lvl. 1;;;#3"/>
  <param name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="gti.foul3"/>
  <param ord="2" name="form_type" value="form"/>
  <param ord="2" name="navigator_position" value="W"/>
  <param ord="2" name="navigator_fields" value="1"/>
  <param ord="2" name="navigator_size" value="6"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="source_filter_field" value="fk"/>
</struct>

```

Y su interfaz final en la aplicación es la siguiente:

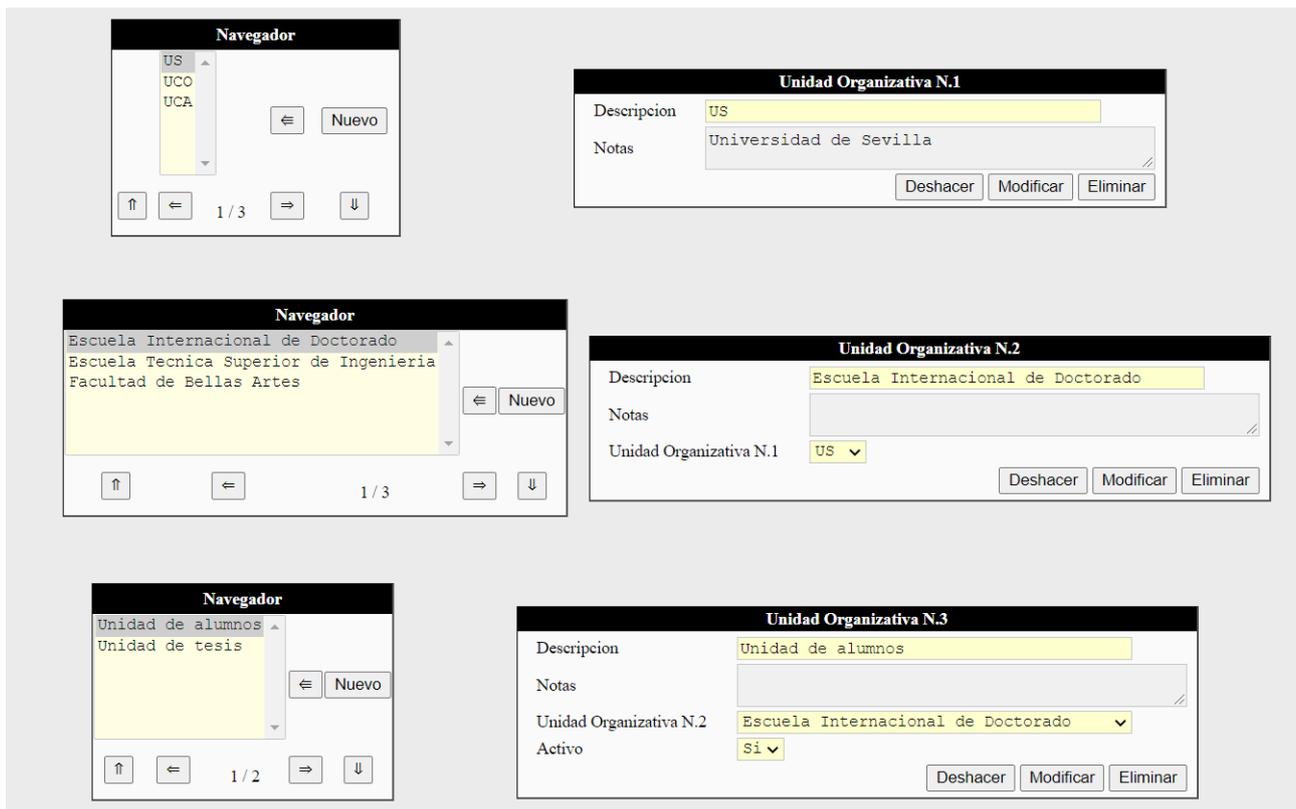


Ilustración 141: Interfaz de usuario final Gestión de unidades organizativas

4.4.2.1.3 Gestión de localizaciones

El código ASL correspondiente al Modelo de Presentación para la interfaz *Gestión de localizaciones* es el siguiente:

```
<struct id="gti.csite1" type="relation">
  <param name="form_split" value="rows=12%,*"/>

  <param name="formid" value="fdescr.form.descr"/>
  <param name="form_type" value="combo"/>
  <param name="form_source" value="gti.Scope"/>
  <param name="fields_hidden" value="2"/>
  <param name="form_caption" value="es=Ambito|en=Scope"/>

  <param ord="2" name="structid" value="gti.csite2"/>
</struct>

<struct id="gti.csite2" type="relation">
  <param name="form_split" value="rows=218,*"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="gti.Site"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="6"/>
  <param name="button_data" value="1"/>
  <param name="form_caption" value="es=Sede|en=Site"/>
  <param name="fields_modify[0]" value="search#DATA:gti.Scope;;;#3"/>
  <param name="fields_modify[1]" value="caption#es=Ambito|en=Scope;;;#3"/>
  <param name="source_filter_field" value="fk"/>

  <param ord="2" name="structid" value="gti.csite3"/>
```



```
</struct>
```

```
<struct id="gti.csite3" type="relation">
  <param name="form_split" value="rows=218,*"/>

  <param name="formid" value="gti.ffkdescr"/>
  <param name="form_source" value="gti.LL1"/>
  <param name="form_type" value="form"/>
  <param name="form_caption" value="es=Localizacion N.1|en=Location Lvl1"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="6"/>
  <param name="button_data" value="1"/>
  <param name="fields_modify[0]" value="search#DATA:gti.Site;;;#3"/>
  <param name="fields_modify[1]" value="caption#es=Sede|en=Site;;;#3"/>
  <param name="source_filter_field" value="fk"/>

  <param ord="2" name="formid" value="gti.fll2"/>
  <param ord="2" name="form_type" value="form"/>
  <param ord="2" name="navigator_position" value="W"/>
  <param ord="2" name="navigator_fields" value="1"/>
  <param ord="2" name="navigator_size" value="6"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="source_filter_field" value="fk"/>
</struct>
```

Y su interfaz final en la aplicación es:

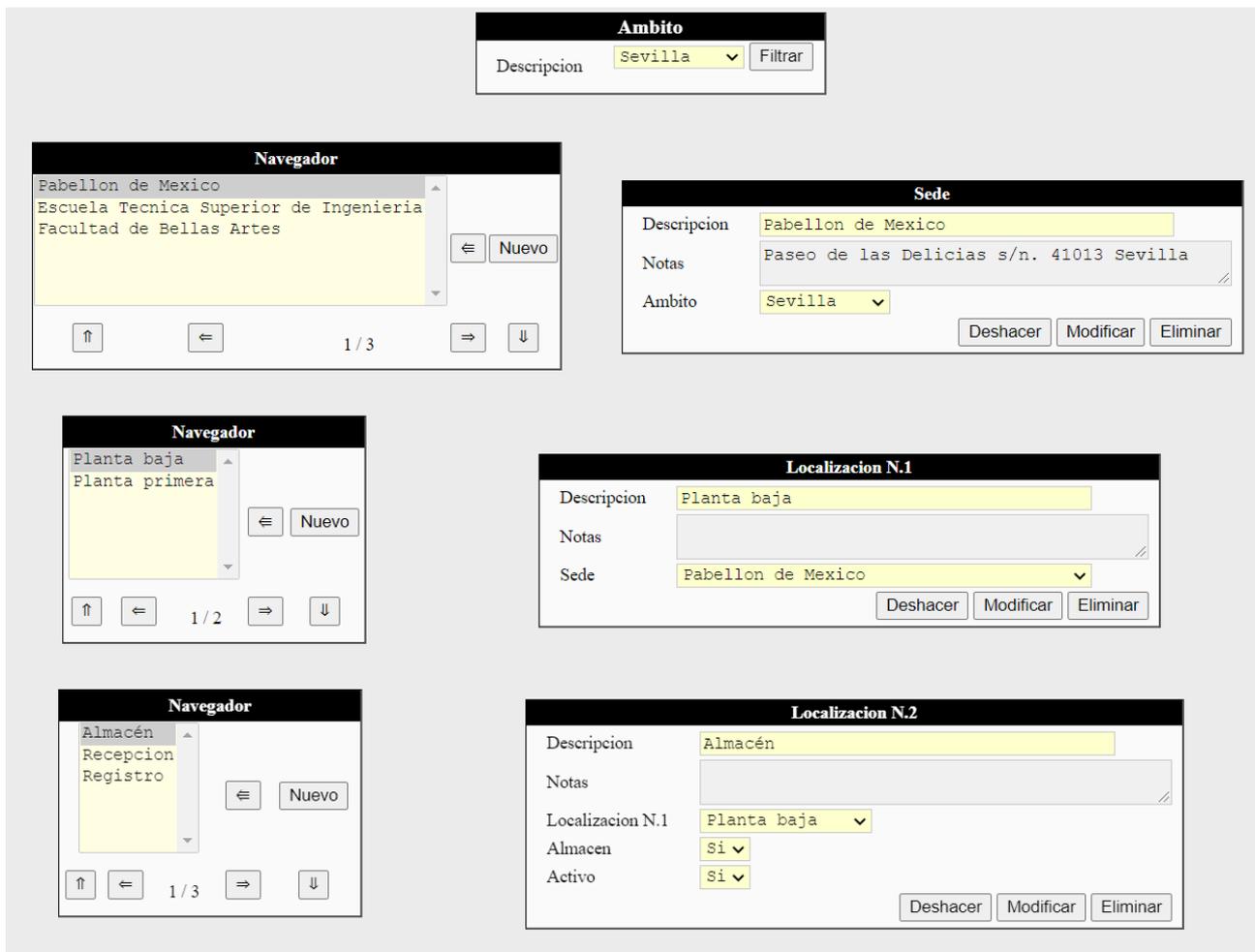


Ilustración 142: Interfaz de usuario final Gestión de localizaciones

4.4.2.1.4 Datos personales

Esta interfaz está sacada del paquete *percon*. Su código ASL es el siguiente:

```
<struct id="percon.struct.edit" type="relation">
  <param name="form_split" value="rows=360,*"/>

  <param name="formid" value="percon.form.person"/>
  <param name="form_subtype" value="tab"/>
  <param name="form_tabs" value="es=Personales|en=Personal:1,2,3,6;es=Foto|en=Photo:4;es=Otros|
en=Others:5"/>
  <param name="fields_remove" value="6:7"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="1"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="2:1"/>
  <param name="navigator_width" value="24"/>

  <param ord="2" name="structid" value="percon.struct.edit_n"/>
</struct>
```

Y su interfaz es la siguiente:



The screenshot shows a web-based user interface for entering personal data. On the left, a 'Navegador' (navigator) window displays a list of names: 'Gomez Ruiz Maria' and 'Martin Sanchez Pedro'. Below the list are navigation arrows and a 'Nuevo' (New) button. The main area is titled 'Person' and contains a form with the following fields: 'Nombre' (Name) with the value 'Maria', 'Apellidos' (Surnames) with the value 'Gomez Ruiz', and 'Fecha nacimiento' (Date of birth) with a date picker set to 'dd/mm/aaaa'. Above the form are buttons for 'Imprimir' (Print), 'Refrescar' (Refresh), 'Ayuda' (Help), and 'Vista' (View). Below the form are buttons for 'Deshacer' (Undo), 'Modificar' (Modify), and 'Eliminar' (Delete). At the bottom of the interface, there is a section titled 'Telefono' (Phone) with a table for adding phone numbers. The table has two columns: 'Telefono' and 'Descripcion' (Description), and an 'Añadir' (Add) button.

Ilustración 143: Interfaz de usuario final Datos personales

Para más información, visitar la wiki de la página oficial de **WAINE** (www.waine.us.es).

4.4.2.1.5 Datos personales de usuario

Esta interfaz está sacada del paquete *upercon*. Su código ASL es el siguiente:

```
<struct id="upercon.struct.edit" type="relation">
  <param name="form_split" value="rows=100,*"/>

  <param name="formid" value="percon.form.userid"/>
  <param name="form_type" value="combo"/>
  <param name="fields_remove" value="2"/>

  <param ord="2" name="structid" value="upercon.struct.edit_n"/>
</struct>
```

Y su interfaz es la siguiente:

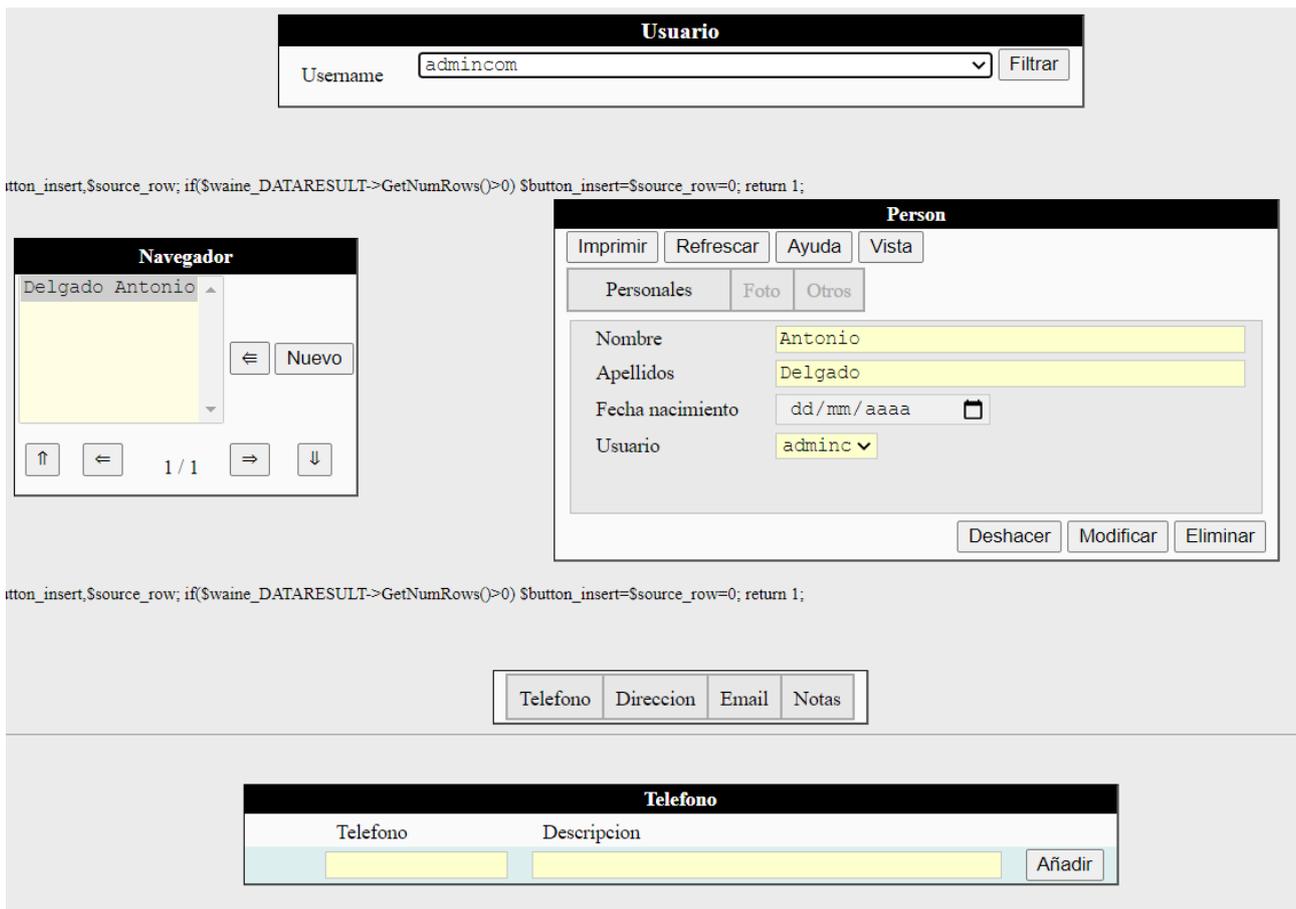


Ilustración 144: Interfaz de usuario final Datos personales de usuario

Para más información, visitar la wiki de la página oficial de **WAINE** (www.waine.us.es).

4.4.3 Construcción de las interfaces de usuario módulo pp

En esta sección, se detalla el proceso de construcción de las interfaces de usuario para cada uno de los roles definidos en la aplicación. Esta construcción se ha hecho en base a los diagramas entidad-relación anotados mostrados en el apartado 5 del Documento de Diseño.

La implementación de estas interfaces se hecho a través del lenguaje ASL propio del MBUIDE de **WAINE**. En este lenguaje, se definen tres modelos: Modelo de Usuario, Modelo de Presentación y Modelo de Diálogo.

En los siguientes apartados, se incluirá cada uno de los modelos para cada usuario, interfaz y el resultado gráfico en forma de captura de la interfaz de usuario.

4.4.3.1 Gestor de adquisiciones

Las funcionalidades a las que tiene acceso un usuario con rol *gestor de adquisiciones* son las siguientes:

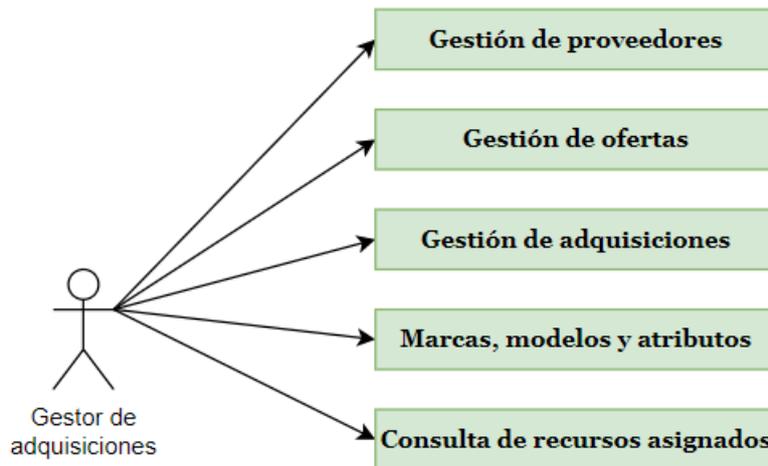


Ilustración 145: Esquema menú Gestor de adquisiciones

El código ASL correspondiente al Modelo de Diálogo del gestor de adquisiciones es el siguiente:

```

<main id="gti.am.mgestadqui" caption="es=AM.Gestor de adquisiciones|en=AM.Purchase Manager">

  <menu caption="es=Gestion adquisiciones|en=Purchase Management" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Gestion de proveedores|en=Provider Management" call="gti.pp.cprovmanag1"/>
    <option caption="es=Gestion de ofertas|en=Offer Management" call="gti.am.coffermanag1"/>
    <option caption="es=Gestion de adquisiciones|en=Purchase Management" call="gti.am.cpurchmanag1"/>
    <option caption="es=Marcas, modelos y atributos|en=Marks, Models and Attributes"
      call="gti.am.cmarkmodel1"/>
  </menu>

  <menu caption="es=Recursos asignados|en=Assigned Resources" tooltip="Procesos, actividades y Tareas">
    <option caption="es=Consulta de recursos asignados|en=Assigned Resources Query" call="gti.am.cassignre-
sourcequery1"/>
  </menu>

  <menu caption="Misc" tooltip="Otras cosas">
    <option caption="es=Sobre este programa|en=About" call="gti.am.cmodinfo" />
    <option caption="es=Ayuda|en=Help" url="http://waine.us.es/dokuwiki/doku.php?
id=waine:help:app:gti:pp:menu:gestor_adquisiciones" />
    <option caption="es=Sobre WAINE|en=About WAINE" url="http://www.waine.org" />
    <option caption="es=Salir|en=Logout" url="logout.php" />
  </menu>

</main>
  
```

En los siguientes apartados se describirán cada una de las funcionalidades.

4.4.3.1.1 Gestión de proveedores

El código ASL correspondiente al Modelo de Presentación para la interfaz *Gestión de proveedores* es el siguiente:

```

<struct id="gti.pp.cprovmanag1" type="relation">
  <param name="form_split" value="rows=228,*"/>

  <param name="formid" value="gti.pp.fprovider"/>
  
```

```

<param name="form_type" value="form"/>
<param name="navigator_position" value="W"/>
<param name="navigator_fields" value="2"/>
<param name="navigator_size" value="6"/>
<param name="button_insert" value="1"/>
<param name="button_update" value="1"/>
<param name="button_delete" value="1"/>
<param name="button_misc" value="0"/>

<param ord="2" name="structid" value="gti.pp.cprovmanag2"/>
</struct>

```

Esta, a su vez, está relacionada con otra estructura que contiene dos pestañas: Contactos y Adjuntos. Su código ASL es el siguiente:

```

<struct id="gti.pp.cprovmanag2" type="tab">
  <param name="form_split" value="rows=100,*"/>

  <param ord="1" name="formid" value="gti.pp.fcontact"/>
  <param ord="1" name="form_source" value="pp.REL_Provider_Person"/>
  <param ord="1" name="tab_name" value="es=Contactos|en=Contacts"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_hidden" value="1"/>
  <param ord="1" name="button_insert" value="1"/>
  <param ord="1" name="button_update" value="0"/>
  <param ord="1" name="button_delete" value="1"/>
  <param ord="1" name="source_filter_field" value="fkprov"/>
  <param ord="1" name="form_modify[0]" value="beforeinsert#gti.pp.actprovmanag"/>

  <param ord="2" name="tab_name" value="es=Adjuntos|en=Attachments"/>
  <param ord="2" name="formid" value="attach.form.catattach"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_modify[0]" value="defvalue#7#3"/> <!-- Asignando categoría -->
  <param ord="2" name="fields_remove" value="5:7"/>
  <param ord="2" name="fields_hidden" value="3:4:6"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="source_filter_field" value="fk"/>
  <param ord="2" name="form_modify[0]" value="filter#fkcat=7"/>
</struct>

```

Finalmente, la interfaz en la aplicación es la siguiente:

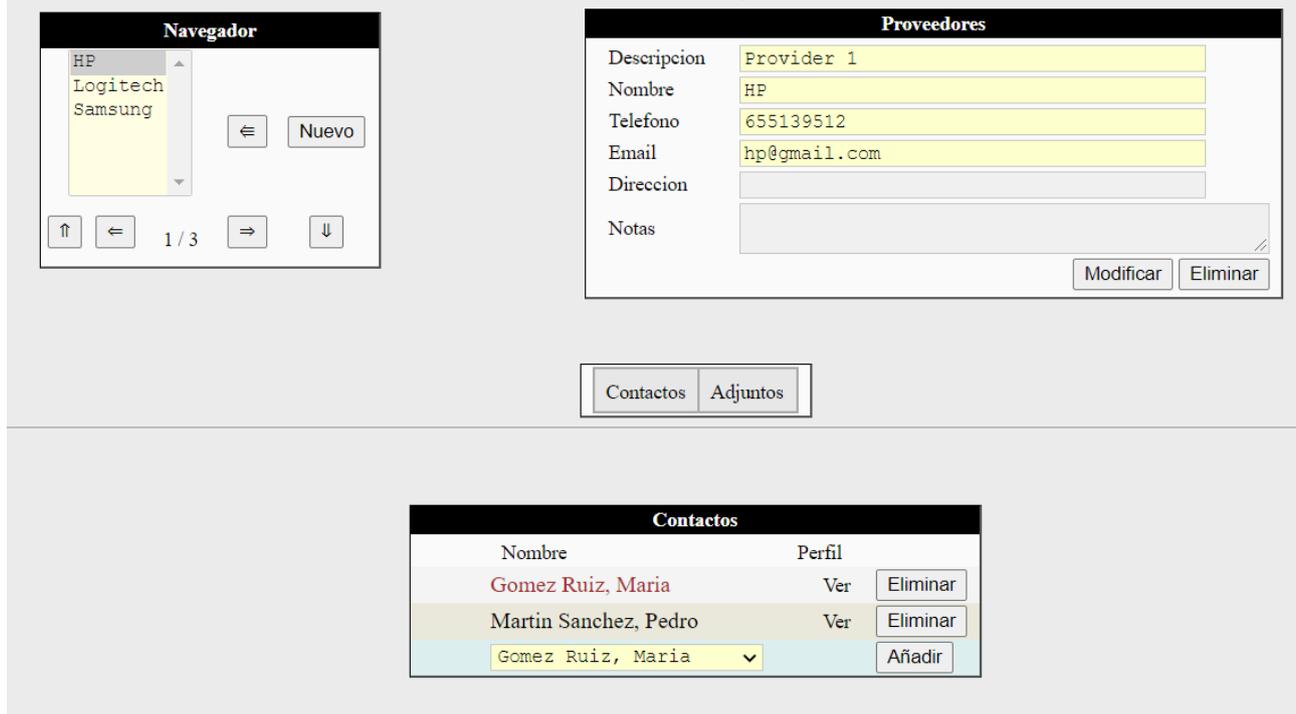


Ilustración 146: Interfaz de usuario final Gestión de proveedores

Y su pestaña adjuntos:



Ilustración 147: Interfaz de usuario final Gestión de proveedores. Pestaña Adjuntos

4.4.3.1.2 Gestión de ofertas

El código ASL correspondiente al Modelo de Presentación para la interfaz *Gestión de ofertas* es el siguiente:

```
<struct id="gti.am.cofferanag1" type="relation">
  <param name="form_split" value="rows=225,*"/>

  <param name="formid" value="gti.pp.fofferreq"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="6"/>
  <param name="form_modify[0]" value="filter#fkscope in (select fkscope from am.REL_Gad_Scope where uid=%userid)"/>
  <param name="fields_modify[0]" value="search#DATA:gti.Scope:pk in (select fkscope from am.REL_Gad_Scope
```

```

where uid=%userid);;#2"/>
<param name="button_insert" value="1"/>
<param name="button_update" value="1"/>
<param name="button_delete" value="1"/>
<param name="button_misc" value="0"/>

<param ord="2" name="structid" value="gti.am.cofferanag2"/>
</struct>

<struct id="gti.am.cofferanag2" type="relation">
  <param name="form_split" value="rows=240,*"/>

  <param name="formid" value="gti.pp.foffer"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="2:1"/>
  <param name="navigator_size" value="6"/>
  <param name="fields_hidden" value="5"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="1"/>
  <param name="button_misc" value="0"/>
  <param name="source_filter_field" value="fkreq"/>

  <param ord="2" name="structid" value="gti.am.cofferanag3"/>
</struct>

```

Esta interfaz, incluye dos pestañas: Elementos y Adjuntos. La pestaña Elementos contiene los modelos relativos a la oferta seleccionada. Su código ASL es el siguiente:

```

<struct id="gti.am.cofferanag3" type="tab">
  <param name="form_split" value="rows=100,*"/>

  <param ord="1" name="formid" value="gti.am.fofferelements"/>
  <param ord="1" name="tab_name" value="es=Elementos|en=Elements"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_hidden" value="2"/>
  <param ord="1" name="button_insert" value="1"/>
  <param ord="1" name="button_update" value="1"/>
  <param ord="1" name="button_delete" value="1"/>
  <param ord="1" name="form_modify[0]" value="afterinsert#gti.am.actsumoffer" />
  <param ord="1" name="form_modify[1]" value="afterdelete#gti.am.actsumoffer" />
  <param ord="1" name="form_modify[2]" value="afterupdate#gti.am.actsumoffer" />
  <param ord="1" name="source_filter_field" value="fkoffer"/>

  <param ord="2" name="tab_name" value="es=Adjuntos|en=Attachments"/>
  <param ord="2" name="formid" value="attach.form.catattach"/>
  <param ord="2" name="form_type" value="table"/>
  <param ord="2" name="fields_modify[0]" value="defvalue#3#3"/> <!-- Asignando categoría -->
  <param ord="2" name="fields_remove" value="5:7"/>
  <param ord="2" name="fields_hidden" value="3:4:6"/>
  <param ord="2" name="button_data" value="1"/>
  <param ord="2" name="source_filter_field" value="fk"/>
  <param ord="2" name="form_modify[0]" value="filter#fkcat=3"/>
</struct>

```

Finalmente, su interfaz en la aplicación es la siguiente:



The screenshot displays four panels from the 'Gestión de ofertas' interface:

- Navegador (top left):** A sidebar with a list containing 'orcor1' and 'orsev1'. It includes navigation arrows and a 'Nuevo' button.
- Petición de oferta (top right):** A form with fields for 'Descripción' (value: orcor1), 'Ambito' (value: Córdoba), and 'Notas'. It has 'Modificar' and 'Eliminar' buttons.
- Navegador (bottom left):** A sidebar with a list containing '2020-12-21 Provider 1'. It includes navigation arrows and a 'Nuevo' button.
- Ofertas (bottom right):** A form with fields for 'Proveedor' (value: Provider 1), 'Fecha' (value: 2020-12-21), 'Descripción' (value: orcor1), and 'Total' (value: 0.00). It has 'Modificar' and 'Eliminar' buttons.

Ilustración 148: Interfaz de usuario final Gestión de ofertas. Formularios

Y sus pestañas:

The screenshot shows the 'Elementos' tab with the following table:

Modelo	Cantidad	Precio Unitario	Total		
Mark 1 - Model 1	2	2.00	4.00	...	Modificar Eliminar
Mark 1 - Model 2	3	3.00	9.00	...	Modificar Eliminar
Mark 2 - Model 1	1	5.00	5.00	...	Modificar Eliminar
				18	
Mark 1 - Model 1	1	0.00		...	Añadir

Ilustración 149: Interfaz de usuario final Gestión de ofertas. Pestaña Elementos

The screenshot shows the 'Adjuntos' tab with the following table:

Descripción	Archivo	
offer cor1	Descargar <input type="checkbox"/> Borrar Seleccionar archivo Ningún archi... seleccionado	Modificar Eliminar
	Descargar <input type="checkbox"/> Borrar Seleccionar archivo Ningún archi... seleccionado	Añadir

Ilustración 150: Interfaz de usuario final Gestión de ofertas. Pestaña Adjuntos

4.4.3.1.3 Gestión de adquisiciones

El código ASL correspondiente al Modelo de Presentación para la interfaz *Gestión de adquisiciones* es el siguiente:

```
<struct id="gti.am.cpurchmanag1" type="relation">
  <param name="form_split" value="rows=90,*"/>

  <param name="formid" value="gti.pp.fofferreq"/>
  <param name="form_type" value="combo"/>
  <param name="fields_hidden" value="2:3"/>
  <param name="source_filter_where" value="fkscope in (select fkscope from am.REL_Gad_Scope where uid=%userid) and type='am'"/>

  <param ord="2" name="structid" value="gti.am.cpurchmanag2"/>
</struct>

<struct id="gti.am.cpurchmanag2" type="relation">
  <param name="form_split" value="rows=90,*"/>

  <param name="formid" value="fdescr.form.descr"/>
  <param name="form_type" value="combo"/>
  <param name="form_source" value="pp.VIEW_Offer"/>
  <param name="form_caption" value="es=Ofertas|en=Offers"/>
  <param name="source_filter_field" value="fkreq"/>

  <param ord="2" name="structid" value="gti.am.cpurchmanag3"/>
</struct>

<struct id="gti.am.cpurchmanag3" type="relation">
  <param name="form_split" value="rows=50%,*"/>

  <param name="formid" value="gti.pp.fpurchase"/>
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="W"/>
  <param name="navigator_fields" value="1"/>
  <param name="navigator_size" value="6"/>
  <param name="fields_hidden" value="7"/>
  <param name="button_insert" value="1"/>
  <param name="button_update" value="1"/>
  <param name="button_delete" value="1"/>
  <param name="button_misc" value="0"/>
  <param name="source_filter_field" value="fkoffer"/>

  <param ord="2" name="structid" value="gti.am.cpurchmanag4"/>
</struct>
```

Esta interfaz, contiene dos pestañas: Elementos y Adjuntos, que siguen el mismo patrón al descrito en la interfaz anterior. Su código ASL es el siguiente:

```
<struct id="gti.am.cpurchmanag4" type="tab">
  <param name="form_split" value="rows=100,*"/>

  <param ord="1" name="formid" value="gti.am.fpurchaseelements"/>
  <param ord="1" name="tab_name" value="es=Elementos|en=Elements"/>
  <param ord="1" name="form_type" value="table"/>
  <param ord="1" name="fields_hidden" value="2"/>
  <param ord="1" name="button_insert" value="1"/>
  <param ord="1" name="button_update" value="1"/>
```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```

<param ord="1" name="button_delete" value="1"/>
<param ord="1" name="form_modify[0]" value="afterinsert#gti.am.actsumpur"/>
<param ord="1" name="form_modify[1]" value="afterdelete#gti.am.actsumpur"/>
<param ord="1" name="form_modify[2]" value="afterupdate#gti.am.actsumpur"/>
<param ord="1" name="source_filter_field" value="fkpur"/>

<param ord="2" name="tab_name" value="es=Adjuntos|en=Attachments"/>
<param ord="2" name="formid" value="attach.form.catattach"/>
<param ord="2" name="form_type" value="table"/>
<param ord="2" name="fields_modify[0]" value="defvalue#8#3"/> <!-- Asignando categoría -->
<param ord="2" name="fields_remove" value="5:7"/>
<param ord="2" name="fields_hidden" value="3:4:6"/>
<param ord="2" name="button_data" value="1"/>
<param ord="2" name="source_filter_field" value="fk"/>
<param ord="2" name="form_modify[0]" value="filter#fkcat=8"/>

</struct>
    
```

Finalmente, su interfaz en la aplicación es la siguiente:

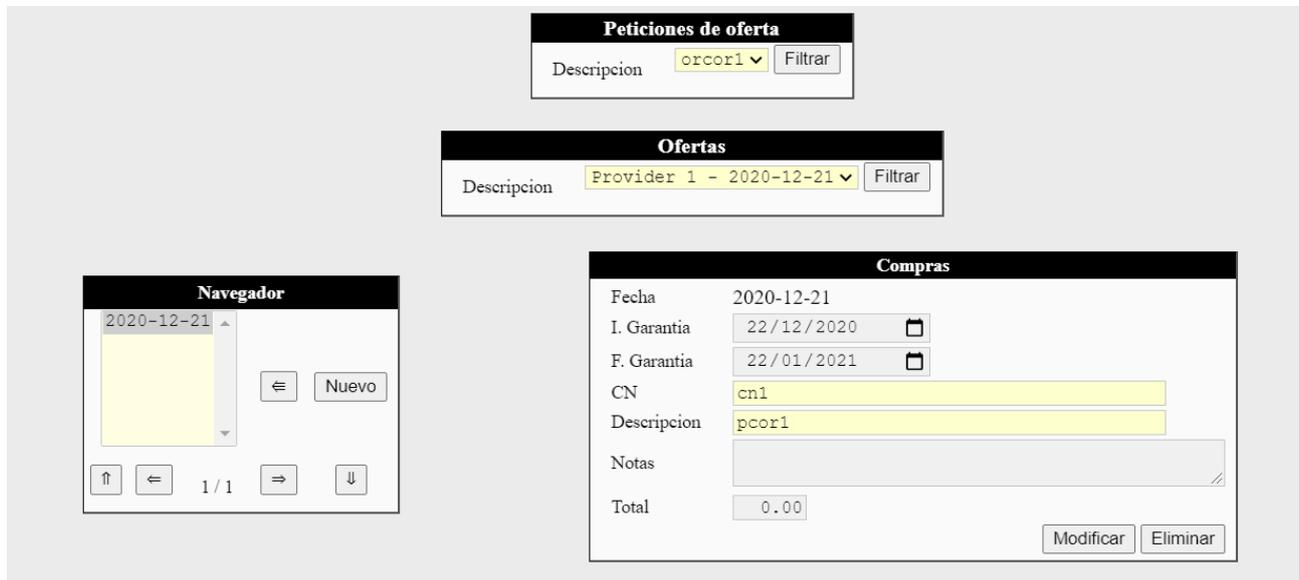


Ilustración 151: Interfaz de usuario final Gestión de adquisiciones. Formularios

Y sus pestañas:



Ilustración 152: Interfaz de usuario final Gestión de adquisiciones. Pestaña Elementos



Ilustración 153: Interfaz de usuario final Gestión de adquisiciones. Pestaña Adjuntos

4.4.3.1.4 Marcas, modelos y atributos

Esta interfaz está ya descrita en el apartado 4.3.1 del Documento de Implementación del módulo de inventario de GTI.

4.4.3.1.5 Consulta de recursos asignados

Esta interfaz está ya descrita en el apartado 4.1.1 del Documento de Implementación del módulo de inventario de GTI.

4.4.3.2 Usuario sin acceso

Las funcionalidades a las que tiene acceso un usuario con rol *usuario sin acceso* son las siguientes:

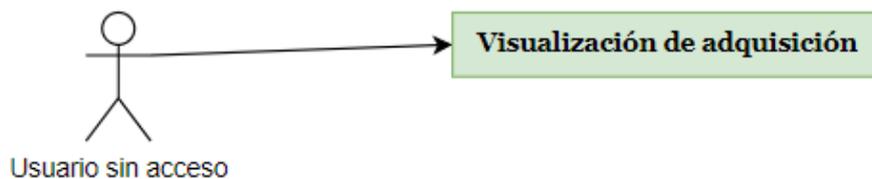


Ilustración 154: Esquema menú Usuario sin acceso

El código ASL correspondiente al Modelo de Diálogo del usuario sin acceso es el siguiente:

```
<main id="gti.am.muser_na" caption="es=GTI.AM.Usuario sin acceso|en=GTI.AM.No Access User">
  <menu caption="es=Gestion usuario sin acceso|en=No Access User Management"
  tooltip="Procesos, actividades y Tareas">
    <option caption="es=Visualizacion de recurso|en=Asset Visualization" call="gti.am.cresourceview1"/>
    <option caption="es=Visualizacion de adquisicion|en=Purchase Visualization" call="gti.pp.cpurchaseview"/>
    <option caption="es=Visualizacion de entrada|en=Input Visualization" call="gti.am.cinputview"/>
    <option caption="es=Visualizacion de salida|en=Output Visualization" call="gti.am.coutputview"/>
  </menu>
  <menu caption="Misc" tooltip="Otras cosas">
    <option caption="es=Sobre este programa|en=About" call="gti.am.cmodinfo" />
  </menu>
</main>
```



4 Construcción de los módulos de activos, adquisiciones y común de GTI

```
<option caption="es=Ayuda|en=Help" url="http://waine.us.es/dokuwiki/doku.php?id=waine:help:app:gti:am:menu:usuario_sin_acceso" />
<option caption="es=Sobre WAINE|en>About WAINE" url="http://www.waine.org" />
<option caption="es=Salir|en=Logout" url="logout.php" />
</menu>
```

```
</main>
```

En los siguientes apartados se describirán cada una de las funcionalidades.

4.4.3.2.1 Visualización de adquisición

El código ASL correspondiente al Modelo de Presentación para la interfaz *Visualización de adquisición* es el siguiente:

```
<struct id="gti.pp.cpurchaseview" type="form">
  <param name="form_type" value="form"/>
  <param name="navigator_position" value="none"/>
  <param name="source_row" value="0"/>

  <param name="formid" value="gti.pp.fpurchase"/>
  <param name="form_caption" value="es=Adquisicion|en=Purchase"/>
  <param name="fields_remove" value="5:6:7:8"/>
  <param name="fields_readonly" value="1-4" />
  <param name="button_insert" value="0"/>
  <param name="button_update" value="0"/>
  <param name="button_delete" value="0"/>
  <param name="button_misc" value="0"/>
  <param name="source_filter_field" value="pk" />
</struct>
```

Y su interfaz en la aplicación es la siguiente:

Adquisición	
Fecha	2020-12-21
I. Garantía	2020-12-22
F. Garantía	2021-01-22
CN	cn1

Ilustración 155: Interfaz de usuario final Visualización de adquisición

4.5 Manuales de usuario

4.5.1 Manuales de usuario de GTI.AM

En esta sección, se recogen los manuales de usuario del módulo de gestión de inventario de GTI. Estos contienen las distintas funcionalidades que se ofrecen para cada rol definido con sus correspondientes interfaces de usuario.

4.5.1.1 Introducción

GTI.AM (Asset Management) es el módulo de gestión de inventario de GTI.

4.5.1.2 Roles

Los roles de usuario definidos en el módulo de gestión de inventario son:

- Administrador del módulo de inventario.
- Gestor de inventario.
- Técnico avanzado.
- Técnico.
- Usuario regular.
- Usuario sin acceso.

4.5.1.3 Manual de usuario del administrador del módulo GTI.AM

En este apartado, se describe el manual de usuario del administrador del módulo de gestión de inventario (GTI.AM).

4.5.1.3.1 Funcionalidades

Las funcionalidades a las que tiene acceso el administrador del módulo son:

- **Administración**
 1. Permisos.
- **Configuración**
 1. Tipos de entradas.
 2. Tipos de salidas.
 3. Estados de los recursos.
 4. Tipos de unidades y unidades.
 5. Tipos de elementos y atributos.
- **Consulta de recursos**
 1. Recursos asignados.

4.5.1.3.2 Permisos

Tipo

- Unidad de interacción.

Descripción

La interfaz *Permisos* permite al administrador del módulo asignar la accesibilidad de la aplicación a los usuarios de la aplicación a través de permisos diferenciados en distintos grupos.

En la parte superior de la interfaz, se permite elegir entre los ámbitos creados, el ámbito al que se desean aplicar los permisos. En ella se distinguen dos campos:

- **Descripción:** nombre del ámbito.
- **Notas (opcional):** comentarios adicionales sobre el ámbito elegido. Hasta 80 caracteres.

En la parte inferior, se permite configurar los permisos para las distintas áreas. Esta está compuesta por cuatro pestañas:

- **Gestores de adquisiciones:** permite a los usuarios con este permiso realizar



4 Construcción de los módulos de activos, adquisiciones y común de GTI adquisiciones en el ámbito seleccionado.

- **Gestores de inventario:** permite a los usuarios con este permiso realizar la total gestión de los activos y sus movimientos en el ámbito seleccionado.
- **Técnicos avanzados:** permite a los usuarios con este permiso realizar la gestión de activos reducida y sus movimientos en el ámbito seleccionado.
- **Técnicos:** permite a los usuarios con este permiso realizar la gestión de activos reducida en el ámbito seleccionado.

IU Final

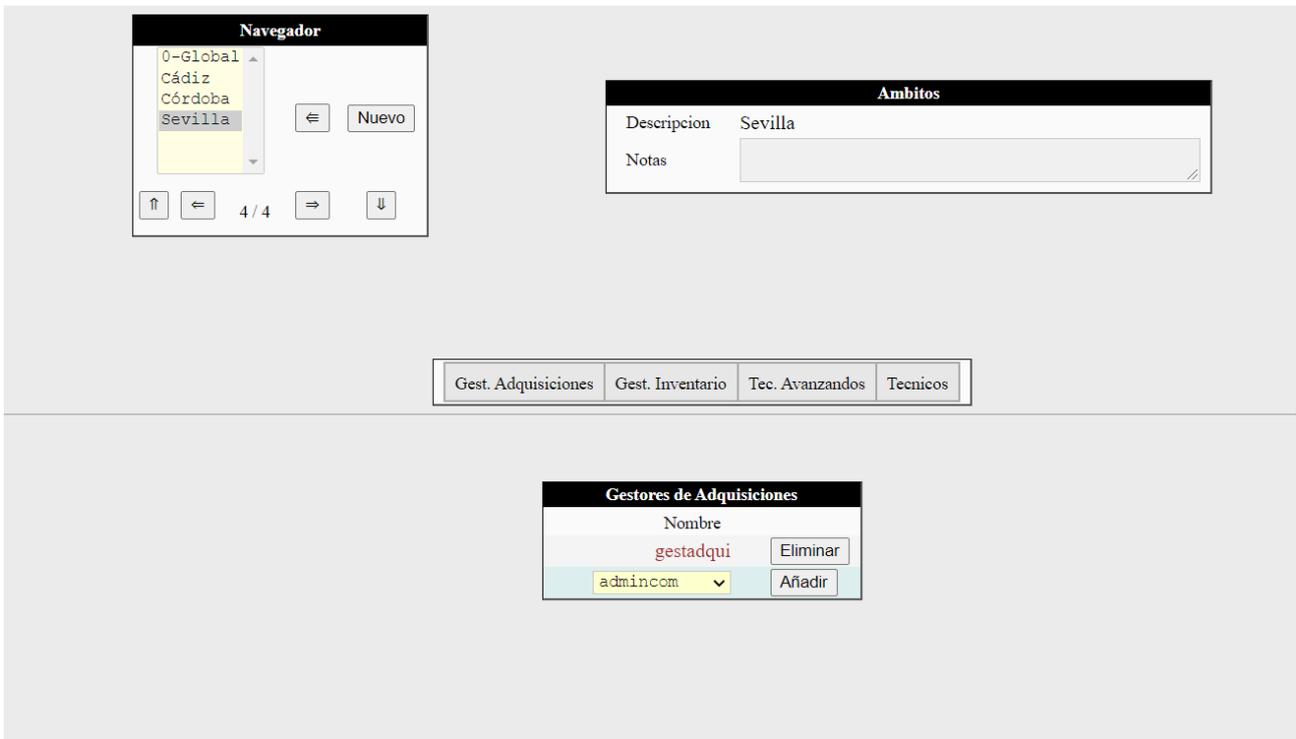


Ilustración 156: Interfaz de usuario Permisos

- Pestaña Gestores de adquisiciones:

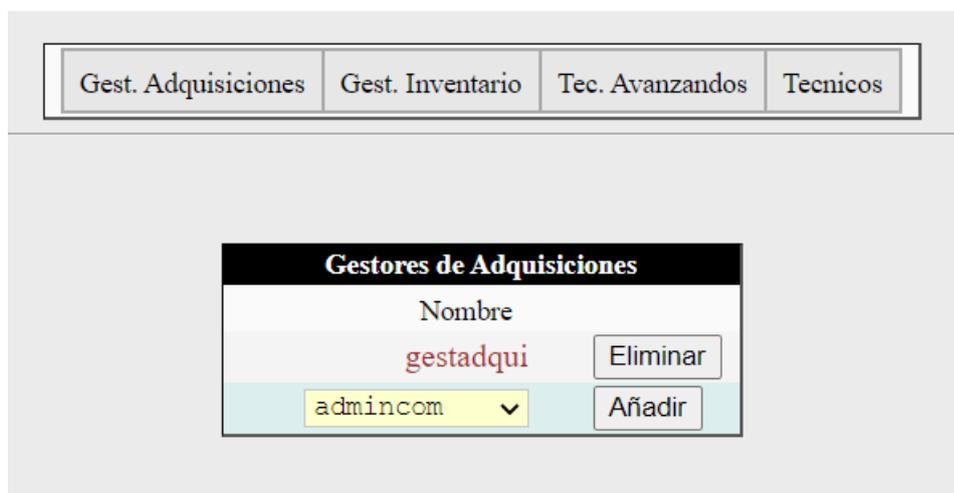


Ilustración 157: Interfaz de usuario Permisos. Pestaña Gestores de adquisiciones

- Pestaña Gestores de inventario:



Ilustración 158: Interfaz de usuario Permisos. Pestaña Gestores de inventario

- Pestaña Técnicos avanzados:



Ilustración 159: Interfaz de usuario Permisos. Pestaña Técnicos avanzados

- Pestaña Técnicos:

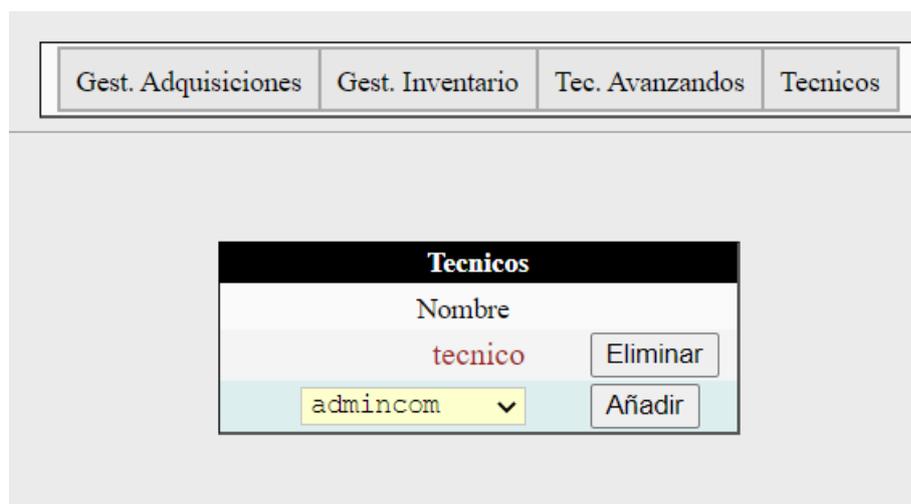


Ilustración 160: Interfaz de usuario Permisos. Pestaña Técnicos

Notas

- Un usuario puede pertenecer a distintos grupos al mismo tiempo.



4 Construcción de los módulos de activos, adquisiciones y común de GTI

- Un usuario no puede pertenecer al mismo grupo por duplicado. La aplicación generará un mensaje de error.

4.5.1.3.3 Tipos de entradas

Tipo

- Unidad de interacción.

Descripción

La interfaz *Tipos de entradas* permite al administrador del módulo añadir y eliminar los tipos de entradas.

En la tabla Tipos de entradas se escribe la descripción y se pulsa el botón Añadir. Además, se permite eliminar un tipo de entrada creado.

IU Final

Tipos de entradas	
Descripcion	
Adquisicion	Eliminar
Interna	Eliminar
Devolucion	Eliminar
	Añadir

Ilustración 161: Interfaz de usuario Tipos de entradas

Notas

- Los tipos de entradas *Adquisición*, *Interna* y *Devolución* son tipos predefinidos y, por tanto, no podrán ser eliminados.

4.5.1.3.4 Tipos de salidas

Tipo

- Unidad de interacción.

Descripción

La interfaz *Tipos de salidas* permite al administrador del módulo añadir y eliminar los tipos de salidas.

En la tabla Tipos de salidas se escribe la descripción y se pulsa el botón Añadir. Además, se permite eliminar un tipo de salida creado.

IU Final

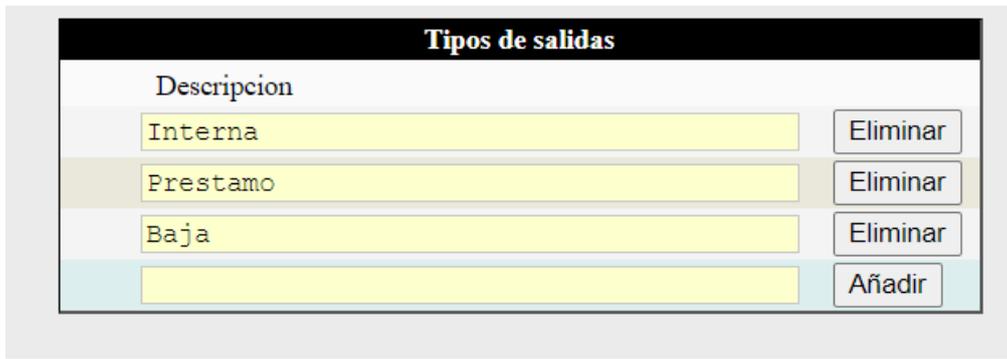


Ilustración 162: Interfaz de usuario Tipos de salidas

Notas

- Los tipos de salidas *Interna*, *Préstamo* y *Baja* son tipos predefinidos y, por tanto, no podrán ser eliminados.

4.5.1.3.5 Estados de los recursos

Tipo

- Unidad de interacción.

Descripción

La interfaz *Estados de los recursos* permite al administrador del módulo añadir y eliminar los estados de los recursos.

En la tabla Estado de los recursos se escribe la descripción y se pulsa el botón Añadir. Además, se permite eliminar un estado de los recursos creado.

IU Final

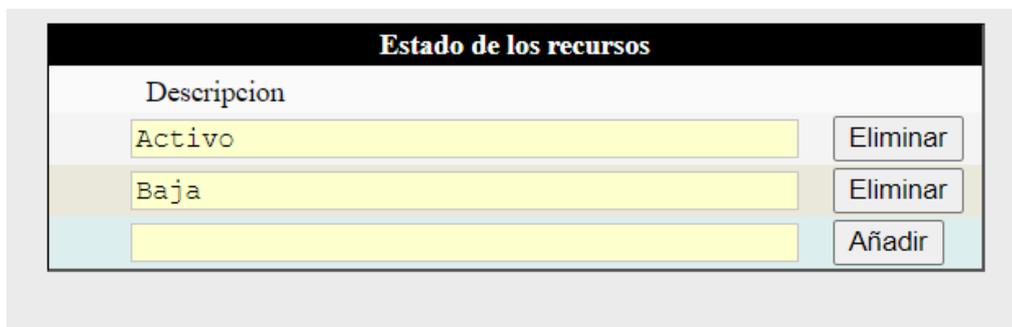


Ilustración 163: Interfaz de usuario Estado de los recursos

Notas

- Los estados de los recursos *Activo* y *Baja* son tipos predefinidos y, por tanto, no podrán ser eliminados.

4.5.1.3.6 Tipos de unidades y unidades

Tipo

- Unidad de interacción.

Descripción



La interfaz *Tipos de unidades y unidades* permite al administrador del módulo añadir, modificar y eliminar los tipos de unidades y sus formatos asociados.

En la parte superior, se permite definir, modificar y eliminar un tipo de unidad o elegir uno entre los ya creados. Se diferencian los siguientes campos:

- **Descripción:** nombre del tipo de unidad.
- **Notas (opcional):** comentarios adicionales sobre el tipo de unidad. Hasta 80 caracteres.

Una vez rellenado estos campos, se pulsa el botón Añadir.

En la parte inferior, se permite definir, modificar y eliminar un formato de unidad para la unidad elegida. Se distinguen los siguientes campos:

- **Abrev:** abreviatura del formato de unidad.
- **Descripción:** nombre de la unidad.
- **Validación:** formato de unidad que se comprobará cuando se le aplique un valor a la unidad.
- **Notas (opcional):** comentarios adicionales sobre la unidad. Hasta 80 caracteres.

Una vez rellenado estos campos, se pulsa el botón Añadir.

IU Final

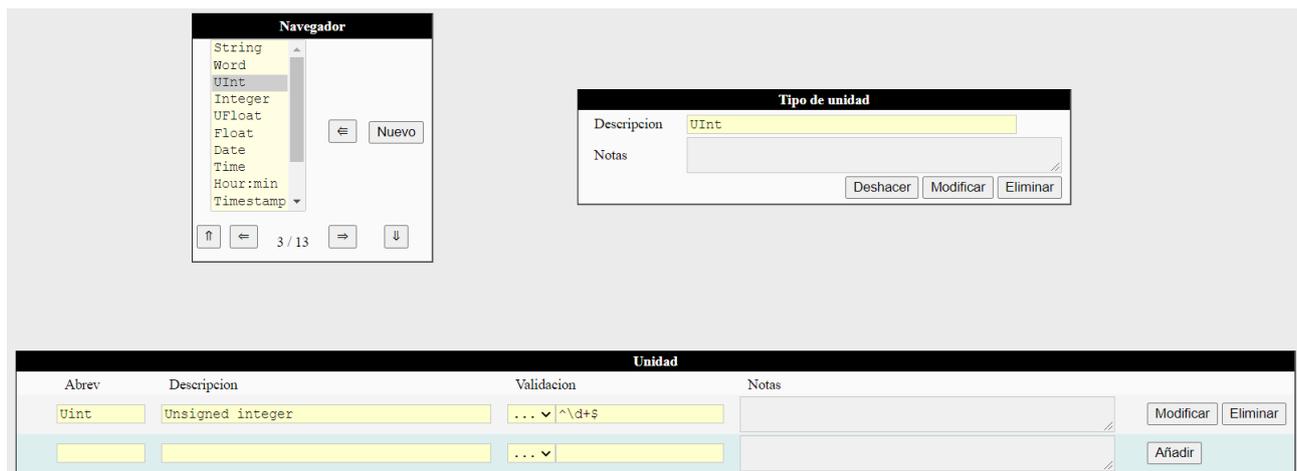


Ilustración 164: Interfaz de usuario Tipos de unidades y unidades

Notas

- Un tipo de unidad no debe tener unidades asociados para su eliminación.

4.5.1.3.7 Tipos de elementos y atributos

Tipo

- Unidad de interacción.

Descripción

La interfaz *Tipos de elementos y atributos* permite al administrador del módulo definir, modificar y eliminar categorías y sus atributos asociados para la clasificación de los activos.

En la parte superior, se permite definir, modificar y eliminar un tipo de elemento inventariable o elegir uno entre los ya creados. Se diferencian los siguientes campos:

- **Descripción:** nombre del tipo de elemento inventariable.

- **Notas (opcional):** comentarios adicionales sobre el tipo de elemento inventariable. Hasta 80 caracteres.

Una vez rellenado estos campos, se pulsa el botón Añadir.

En la parte inferior, se permite añadir, modificar y eliminar atributos tanto de tipo modelo, como de tipo activo. Se distinguen los siguientes campos:

- **Nombre:** nombre del atributo.
- **Unidad:** tipo de unidad asociado al atributo.

Una vez rellenado estos campos, se pulsa el botón Añadir.

IU Final

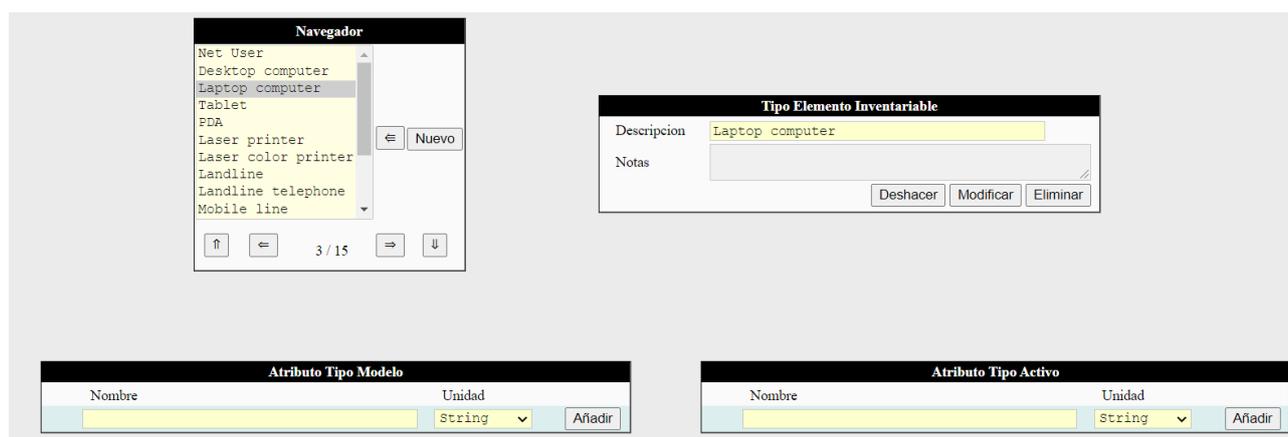


Ilustración 165: Interfaz de usuario Tipos de elementos y atributos

Notas

- Un tipo de elemento inventariable no debe tener atributos asociados para su eliminación.

4.5.1.3.8 Recursos asignados

Tipo

- Unidad de interacción.

Descripción

La interfaz *Recursos asignados* permite al usuario consultar los aquellos activos de los que es responsable.

En la parte superior, se permite elegir para visualizar un activo de los que el usuario es responsable. Este formulario se divide en dos pestañas: Información y Detalles y se diferencian los siguientes campos:

Pestaña Información:

- **ID:** identificador del activo dentro de la base de datos.
- **NS:** número de serie del activo.
- **IDR:** identificador del activo.
- **Estado:** estado del recurso.
- **Modelo:** marcas, modelo y tipo del activo.
- **OU:** unidad organizativa a la que pertenece el activo.



- **Localización:** localización a la que pertenece el activo.
- **Responsable:** responsable del activo.
- **Activo padre:** IDR del activo padre al que está asociado el activo.

Pestaña Detalles:

- **Descripción:** nombre asociado al activo.
- **Notas:** comentarios adicionales sobre el activo. Hasta 80 caracteres.
- **Nombre equipo:** nombre definido al equipo.
- **Instante creación:** momento en el que se crea el activo en la aplicación. YYYY-MM-DD HH:MM.
- **Instante modificación:** momento en el que se hace la última modificación del activo. Formato YYYY-MM-DD HH:MM.
- **Usuario dio de alta:** usuario que crea el activo en la aplicación.

En la parte inferior, se diferencian dos pestañas:

- **Propiedades:** se permite visualizar los valores de los atributos de tipo activo relacionados con el activo.
- **Recursos hijos:** se permite visualizar algunos campos de los activos hijos relacionados con el activo elegido.

Pulsando en el botón “Ver”, se permite acceder a la interfaz *Recursos asignados: recursos hijos*, que es igual la actualmente descrita, pero sin la necesidad de un navegador ya que el activo se elige de la fila seleccionada.

IU Final

The screenshot shows a web interface for managing assets. On the left is a sidebar titled 'Navegador' with a tree view containing nodes: ns2 id2, ns3 id3, ns4, ns5, and ns6. Below the tree are navigation arrows and a 'Nuevo' button. The main area is titled 'Activo' and has two tabs: 'Informacion' (selected) and 'Detalles'. The 'Informacion' tab displays the following details:

ID	6
NS	ns6
IDR	
Estado	Activo
Modelo	Mark 2 - Model 1 - Net User
OU	
Localizacion	Sevilla - Pabellon de Mexico - Planta baja - Recepcion
Responsable	admininv
Activo padre	

Below the details panel are two tabs: 'Propiedades' and 'Recursos Hijo'. At the bottom of the interface is a table with the following data:

Activo				
Marca y Modelo	Responsable	NS	IDR	Vis.Activos
Mark 1 - Model 1	admininv	ns1	id1	Ver
1				

Ilustración 166: Interfaz de usuario Recursos asignados

- Pestaña Propiedades:

This screenshot shows the 'Propiedades' tab selected. It features two tabs at the top: 'Propiedades' (selected) and 'Recursos Hijo'. Below them is a table titled 'Atributos' with the following data:

Nombre	Valor	Unidad
Hostname		1
Alias		1

Ilustración 167: Interfaz de usuario Recursos asignados. Pestaña Propiedades

- Pestaña Recursos hijos:

This screenshot shows the 'Recursos Hijo' tab selected. It features two tabs at the top: 'Propiedades' and 'Recursos Hijo' (selected). Below them is a table titled 'Activo' with the following data:

Activo				
Marca y Modelo	Responsable	NS	IDR	Vis.Activos
Mark 1 - Model 1	admininv	ns1	id1	Ver
1				

Ilustración 168: Interfaz de usuario Recursos asignados. Pestaña Recursos hijos

Notas



- Si un usuario no tiene ningún recurso asignado, el navegador estará vacío y no podrá interactuar con esta interfaz de usuario.

4.5.1.4 Manual de usuario del gestor de inventario GTI.AM

En este apartado, se describe el manual de usuario del gestor de inventario de gestión de inventario (GTI.AM).

4.5.1.4.1 Funcionalidades

Las funcionalidades a las que tiene acceso el gestor de inventario son:

- **Configuración**
 1. Marcas, modelos y atributos.
- **Gestión de movimientos**
 1. Gestión de entradas.
 2. Gestión de salidas.
- **Activos**
 1. Gestión de inventario.
 2. Gestión de inventario por tipo de recurso.
 3. Gestión de activos.
 4. Edición de activos.
- **Estadísticos**
 1. Tipos de activos por sede.
 2. Estados y marcas de activos por sede y tipo.
- **Consulta de recursos**
 1. Recursos asignados.

4.5.1.4.2 Marcas, modelos y atributos

Tipo

- Unidad de interacción.

Descripción

La interfaz *Marcas, modelos y atributos* permite al gestor de inventario y gestor de adquisiciones crear, modificar y eliminar una marca, un modelo y sus atributos.

En la parte superior, se permite añadir, cambiar y eliminar una marca. Se diferencian los siguientes campos:

- **Descripción:** nombre de la marca.
- **Notas (opcional):** comentarios adicionales acerca de la marca. Hasta 80 caracteres.

Una vez relleno estos campos, se pulsa el botón Añadir.

En la parte central, se permite añadir, cambiar y eliminar un modelo de la marca elegida en el formulario superior. Se diferencian los siguientes campos:

- **Tipo:** tipo del modelo creado.
- **Nombre:** nombre del modelo.
- **Número de modelo:** número identificativo del modelo.
- **Notas (opcional):** comentarios adicionales acerca del modelo. Hasta 80 caracteres.

Una vez relleno estos campos, se pulsa el botón Añadir.

Finalmente, en la parte inferior, se permite asignar un valor a los atributos inherentes al modelo creado/elegido. Se diferencian los siguientes campos:

- **Nombre:** nombre del atributo de tipo modelo.
- **Valor:** valor de dicho atributo.
- **Unidad:** unidad elegida para complementar el valor.

Una vez relleno estos campos, se pulsa el botón Modificar.

IU Final

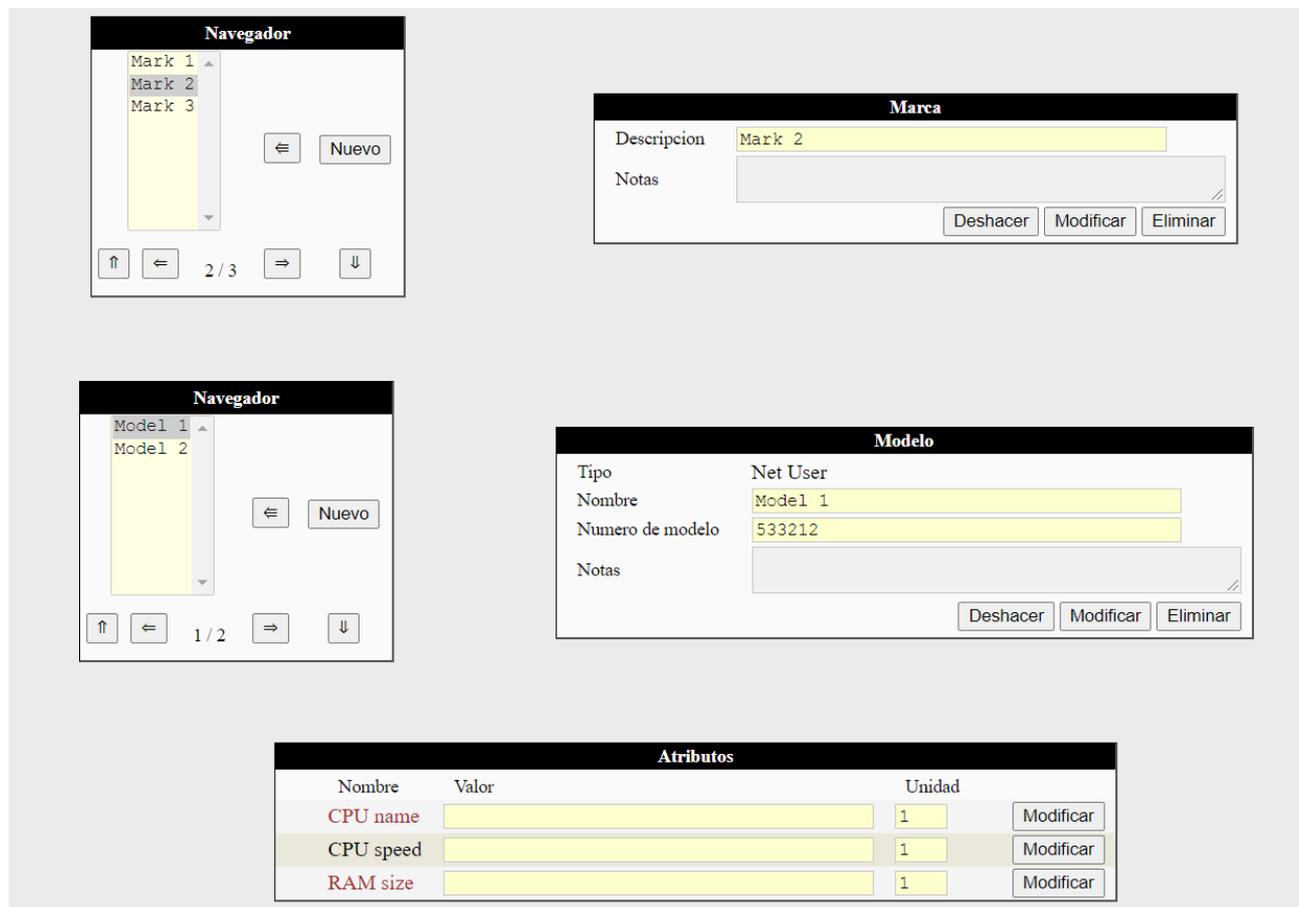


Ilustración 169: Interfaz de usuario Marcas, modelos y atributos

Notas

- Los atributos que aparecen están ligados al modelo elegido y solo podrán ser añadidos o eliminados desde la interfaz *Tipos de elementos y atributos*.
- Una marca no puede ser eliminada si tienes modelos relacionados.

4.5.1.4.3 Gestión de entradas

Tipo



- Unidad de interacción.

Descripción

La interfaz *Gestión de entradas* permite al gestor de inventario y técnico avanzado filtrar, añadir, modificar, eliminar y generar informes de las entradas y añadir, modificar y eliminar los activos que las componen.

En la parte superior, se permite filtrar las entradas por sus campos más relevantes. Se diferencian dos pestañas de filtros: Básicos y Avanzados.

Los campos del filtro básico son:

- **Destino:** localización destino de la entrada.
- **ID Entrada:** identificador de la entrada.
- **Descripción:** nombre de la entrada.
- **Tipo:** tipo de entrada.

Los campos del filtro avanzado son:

- **Emisor:** usuario que gestiona los activos de la entrada.
- **Receptor:** usuario que recibe los activos de la entrada.

Una vez relleno alguno de estos campos, se pulsa el botón Filtrar.

Posteriormente, según los campos del filtrado, aparecerán en el navegador las entradas que cumplen dichos filtros. En este formulario, se distinguen tres pestañas: Información, Notas, Detalles. Se diferencian los siguientes campos:

Pestaña Información:

- **ID Entrada:** identificador de la entrada.
- **Descripción:** nombre de la entrada.
- **Tipo:** tipo de entrada.
- **Instante:** instante de creación de la entrada. Se pone la hora actual por defecto.
- **Emisor:** nombre del usuario del emisor de la entrada.
- **Receptor:** nombre del usuario del receptor de la entrada.
- **Adquisición (semi opcional):** nombre de la adquisición. De aquí se copiarán los activos de la adquisición marcada en caso de que el tipo de entrada sea igual a “Adquisición”.
- **Salida (semi opcional):** nombre de la salida. De aquí se copiarán los activos de la salida marcada en caso de que el tipo de entrada sea distinto a “Adquisición”.
- **Origen(opcional):** localización origen de la entrada.
- **Destino:** localización destino de la entrada.

Pestaña Notas:

- **Notas(opcional):** comentarios adicionales acerca de la entrada. Hasta 80 caracteres.

Pestaña Detalles:

- **Creador entrada:** usuario que crea la entrada. Se pone el usuario actual por defecto.

Una vez relleno estos campos, se pulsa el botón Añadir.

Por último, en la parte inferior, se diferencian tres pestañas relacionadas con la entrada seleccionada: Activos, Resumen y Adjuntos:

- **Activos:** se permite añadir, modificar y eliminar los activos relacionados. Se diferencian los siguientes campos:
 - **Marca y modelo:** marca y modelo del activo.
 - **NS:** número de serie del activo.
 - **IDR:** identificador único de un activo.
 - **Ed. Activos:** botón que redirecciona a la interfaz de *Edición de activos* si se trata del gestor de inventario y a la interfaz *Edición de activos reducida* si se trata del técnico avanzado.
- **Resumen:** listado con la cantidad de cada tipo de activo de los activos relacionados. Se diferencian los siguientes campos:
 - **Tipo:** tipo del activo.
 - **Cantidad:** número de veces que se encuentra dicho tipo de activo en la entrada.
- **Adjuntos:** se permite añadir un adjunto a la entrada o ver los que ya se han incluido.

IU Final

The screenshot displays the 'Gestión de entradas' (Entry Management) interface. At the top, there is a 'Filtros' (Filters) section with tabs for 'Basicos' and 'Avanzados'. Below this, there are input fields for 'Destino', 'ID Entrada' (set to 3), 'Descripcion', and 'Tipo', along with 'Deshacer' and 'Filtrar' buttons.

The main area features a 'Navegador' (Navigator) on the left and a large 'Entrada' (Entry) details window in the center. The 'Entrada' window has tabs for 'Informacion', 'Notas', and 'Detalles'. It displays the following information:

- ID Entrada: 3
- Descripcion: input2
- Tipo: Adquisicion
- Instante: 2020-12-21 18:33
- Emisor: test
- Receptor: gestadqui
- Adquisicion: psev1
- Salida: [empty]
- Origen: Sevilla - Pabellon de Mexico - Planta baja - Recepcion
- Destino: Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE

Buttons for 'Modificar', 'Eliminar', and 'Informe' are located at the bottom right of the 'Entrada' window.

Below the 'Entrada' window, there are three tabs: 'Activos', 'Resumen', and 'Adjuntos'. The 'Activos' tab is active, showing a table of related assets:

Marca y Modelo	NS	IDR	Ed. Activos		
Mark 2 - Model 2	ns7	id7	Ver	Modificar	Eliminar
Mark 2 - Model 2	ns8	id8	Ver	Modificar	Eliminar
Mark 3 - Model 1	ns9	id9	Ver	Modificar	Eliminar

Ilustración 170: Interfaz de usuario Gestión de entradas

- Filtro avanzado:



Ilustración 171: Interfaz de usuario Gestión de entradas. Filtro avanzado

- Pestaña Activos:



Ilustración 172: Interfaz de usuario Gestión de entradas. Pestaña Activos

- Pestaña Resumen:

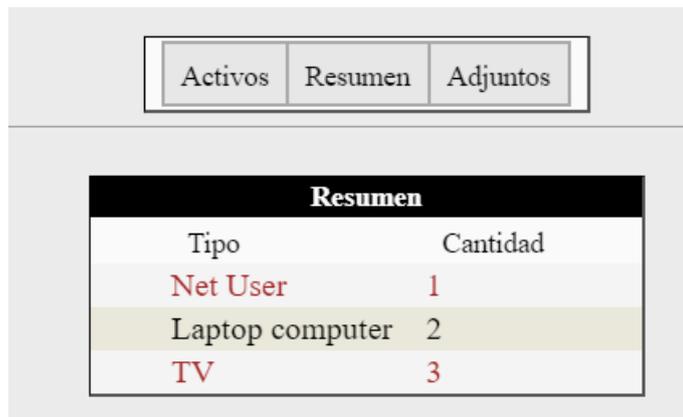


Ilustración 173: Interfaz de usuario Gestión de entradas. Pestaña Resumen

- Pestaña Adjuntos:



Ilustración 174: Interfaz de usuario Gestión de entradas. Pestaña Adjuntos

Notas

- Solo se mostrarán aquellos destinos cuyas localizaciones tenga permisos el usuario actual acorde con la interfaz Permisos.

- Eliminar una entrada no supondrá la eliminación de los activos relacionados.

4.5.1.4.4 Gestión de salidas

Tipo

- Unidad de interacción.

Descripción

La interfaz *Gestión de salidas* permite al gestor de inventario o técnico avanzado filtrar, añadir, modificar, eliminar y generar informes de las salidas y añadir y eliminar los activos que las componen.

En la parte superior, se permite filtrar las salidas por sus campos más relevantes. Se diferencian dos pestañas de filtros: Básicos y Avanzados.

Los campos del filtro básico son:

- **Origen:** localización origen de la salida.
- **ID Salida:** identificador de la salida.
- **Descripción:** nombre de la salida.
- **Tipo:** tipo de salida.

Los campos del filtro avanzado son:

- **Emisor:** usuario que gestiona los activos de la salida.
- **Receptor:** usuario que recibe los activos de la salida.

Una vez relleno alguno de estos campos, se pulsa el botón Filtrar.

Posteriormente, según los campos del filtrado, aparecerán en el navegador las salidas que cumplen dichos filtros. En este formulario, se distinguen tres pestañas: Información, Notas, Detalles. Se diferencian los siguientes campos:

Pestaña Información:

- **ID Salida:** identificador de la salida.
- **Descripción:** nombre de la salida.
- **Tipo:** tipo de salida.
- **Instante:** instante de creación de la salida. Se pone la hora actual por defecto.
- **Emisor:** nombre del usuario del emisor de la salida.
- **Receptor:** nombre del usuario del receptor de la salida.
- **Origen:** localización origen de la salida.
- **Destino(opcional):** localización destino de la salida.

Pestaña Notas:

- **Notas(opcional):** comentarios adicionales acerca de la salida. Hasta 80 caracteres.

Pestaña Detalles:

- **Responsable salida:** usuario que crea la salida. Se pone el usuario actual por defecto.

Una vez relleno estos campos, se pulsa el botón Añadir.



Por último, en la parte inferior, se diferencian tres pestañas relacionadas con la salida seleccionada: Activos, Resumen y Adjuntos:

- **Activos:** se permite añadir y eliminar los activos relacionados. Se diferencian los siguientes campos:
 - **NS – ID - Marca - modelo:** número de serie, identificador, marca y modelo del activo.
 - **Ed. Activos:** botón que redirecciona a la interfaz de *Edición de activos* si se trata del gestor de inventario y a la interfaz *Edición de activos reducida* si se trata del técnico avanzado.
- **Resumen:** listado con la cantidad de cada tipo de activo de los activos relacionados. Se diferencian los siguientes campos:
 - **Tipo:** tipo del activo.
 - **Cantidad:** número de veces que se encuentra dicho tipo de activo en la salida.
- **Adjuntos:** se permite añadir un adjunto a la salida o ver los que ya se han incluido.

IU Final

Ilustración 175: Interfaz de usuario Gestión de salidas

- Filtro avanzado:



Ilustración 176: Interfaz de usuario Gestión de salidas. Filtro avanzado

- Pestaña Activos:



Ilustración 177: Interfaz de usuario Gestión de salidas. Pestaña Activos

- Pestaña Resumen:

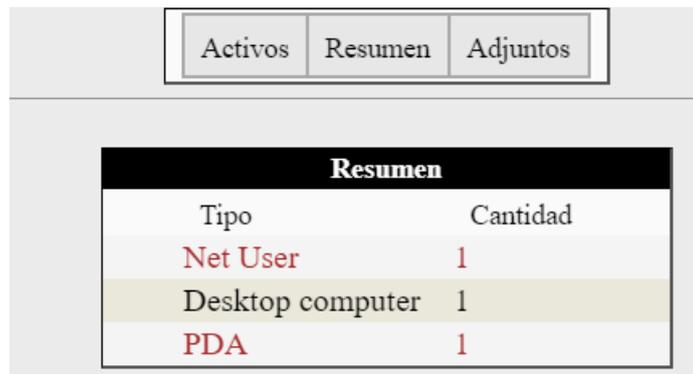


Ilustración 178: Interfaz de usuario Gestión de salidas. Pestaña Resumen

- Pestaña Adjuntos:



Ilustración 179: Interfaz de usuario Gestión de salidas. Pestaña Adjuntos

Notas

- Solo se mostrarán aquellos orígenes cuyas localizaciones tenga permisos el usuario actual acorde con la interfaz *Permisos*.
- Eliminar una salida no supondrá la eliminación de los activos relacionados.



4.5.1.4.5 Gestión de inventario

Tipo

- Unidad de interacción.

Descripción

La interfaz *Gestión de inventario* permite al gestor de inventario filtrar y modificar los activos.

En la parte superior, se permite filtrar los activos que se desean buscar. Se diferencian tres pestañas de filtros: Básicos, Avanzados y Unidad Organizativa.

Los campos del filtro básico son:

- **Localización:** localización vinculada al activo.
- **Marca y Modelo:** marca y modelo del activo.
- **Responsable:** usuario responsable del activo.
- **NS:** número de serie del activo.
- **IDR(opcional):** identificador único del activo.

Los campos del filtro avanzado son:

- **Nombre equipo:** nombre del equipo asociado al activo.
- **Usuario dio de alta:** usuario que dio de alta al activo.

Los campos del filtro unidad organizativa son:

- **OU:** unidad organizativa vinculada al activo.

Una vez rellenado algunos de estos campos, se pulsa el botón Filtrar.

En la parte inferior, se muestran todos los activos que cumplen los requisitos del filtrado. Aquí, se permite realizar modificaciones de sus campos y acceder a la interfaz *Edición de activos* a través del botón “Ver”. Los campos que se describen los mismos a los mencionados en el filtrado.

IU Final

Activo							
Cant	Localización	OU	Marca y Modelo	Responsable	NS	IDR	Ed.Activos

Ilustración 180: Interfaz de usuario Gestión de inventario

- Filtro avanzado:

Activo	
Nombre equipo	Usuario dio de alta

Ilustración 181: Interfaz de usuario Gestión de inventario. Filtro avanzado

- Filtro unidad organizativa:

Ilustración 182: Interfaz de usuario Gestión de inventario. Filtro unidad organizativa

- Tabla Activo:

Cant		Localizacion	OU
1		Sevilla - Pabellon de Mexico - Planta baja - Recepcion	
1		Sevilla - Pabellon de Mexico - Planta baja - Recepcion	
1		Sevilla - Pabellon de Mexico - Planta baja - Recepcion	
1		Sevilla - Pabellon de Mexico - Planta baja - Recepcion	
1		Sevilla - Pabellon de Mexico - Planta baja - Recepcion	
1		Sevilla - Pabellon de Mexico - Planta baja - Recepcion	
6			

Activo							
	Marca y Modelo	Responsable	NS	IDR	Ed.Activos		
	Mark 1 - Model 1	admininv	ns1	id1	Ver		Modificar
	Mark 1 - Model 1	admininv	ns2	id2	Ver		Modificar
	Mark 1 - Model 2	admininv	ns3	id3	Ver		Modificar
	Mark 1 - Model 2	admininv	ns4		Ver		Modificar
	Mark 1 - Model 2	admininv	ns5		Ver		Modificar
	Mark 2 - Model 1	admininv	ns6		Ver		Modificar

Ilustración 183: Interfaz de usuario Gestión de inventario. Tabla Activo

Notas

- Solo se mostrarán aquellas localizaciones que tenga permisos el gestor de inventario acorde con la interfaz *Permisos*.

4.5.1.4.6 Gestión de inventario por tipo de recurso

Tipo

- Unidad de interacción.

Descripción

La interfaz *Gestión de inventario por tipo de recurso* permite al gestor de inventario filtrar y modificar los activos.

En la parte superior, se permite filtrar los activos que se desean buscar. Se diferencian los siguientes campos:

Los campos del filtro básico son:

- **Localización:** localización vinculada al activo.
- **Tipo de recurso:** tipo del activo.

Una vez rellenado algunos de estos campos, se pulsa el botón Filtrar.

En la parte inferior, se muestran todos los activos que cumplen los requisitos del filtrado. Aquí, se permite realizar modificaciones de sus campos y acceder a la interfaz *Edición de activos* a través del botón “Ver”. Los campos que se describen los mismos a los mencionados en el filtrado.

IU Final



Filtros

Localización Sevilla - Escuela Tecnica Superior de Ingeniería - 0 - SO

Tipo de recurso Net User

Activo

Cant	Localizacion	OU	Marca y Modelo	Responsable	NS	IDR	Ed.Activos
------	--------------	----	----------------	-------------	----	-----	------------

Ilustración 184: Interfaz de usuario Gestión de inventario por tipo de recurso

- Tabla Activo:

Activo						
Cant	Localizacion	OU	Marca y Modelo	Responsable	NS	IDR
1	Sevilla - Pabellon de Mexico - Planta baja - Recepcion	Universidad de Sevilla - Escuela Internacional de Doctorado				
1						

Activo						
Localizacion	Marca y Modelo	Responsable	NS	IDR	Ed.Activos	
Doctorado - Unidad de alumnos	Mark 2 - Model 1	adminiv	ns6		Ver	<input type="button" value="Modificar"/>

Ilustración 185: Interfaz de usuario Gestión de inventario por tipo de recurso. Tabla Activo

Notas

- Solo se mostrarán aquellas localizaciones que tenga permisos el gestor de inventario acorde con la interfaz *Permisos*.

4.5.1.4.7 Gestión de activos

Tipo

- Unidad de interacción.

Descripción

La interfaz *Gestión de activos* permite al gestor de inventario filtrar, modificar y generar un informe de los activos a los que el usuario tiene permisos.

En la parte superior, se presentan dos combos para la elección del ámbito y localización sobre la que se quieren filtrar los activos.

En la parte central, se muestran todos los activos que tienen como localización, la localización seleccionada en el combo superior. Se permite modificar y generar un informe de estos.

En la parte inferior, se muestran cinco pestañas relacionadas con el activo seleccionado: Propiedades, Recursos Hijo, Histórico, Traza y Adjuntos.

- **Propiedades:** se permite añadir un valor a los atributos inherentes al tipo de activo. Sus campos son:
 - **Nombre:** nombre del atributo.
 - **Valor:** el valor dado al atributo.
 - **Unidad:** unidad elegida para complementar al valor.
- **Recursos Hijo:** se permite modificar los activos cuyo activo padre es el seleccionado. Sus

campos son:

- **Marca y Modelo:** marca y modelo del activo hijo.
- **Responsable:** responsable del activo hijo.
- **NS:** número de serie del activo hijo.
- **IDR(opcional):** identificador único del activo hijo.

A través del botón “Ver”, se permite acceder a la interfaz de *Edición de activos* del activo hijo.

- **Histórico:** se permite añadir anotaciones con respecto al activo seleccionado. Sus campos son:
 - **Fecha y Hora:** fecha y hora del momento de creación de la anotación. Se pone el día y hora actual por defecto.
 - **Usuario:** usuario que crea la anotación. Se pone el usuario actual por defecto.
 - **Anotación:** breve descripción de lo que se quiere anotar.
- **Traza:** historial de todos los cambios que se han realizado sobre el activo seleccionado. Se diferencian los siguientes campos:
 - **Fecha y Hora:** fecha y hora del momento de creación de la traza. Se pone el día y hora actual por defecto.
 - **Usuario:** usuario que realiza la modificación del activo. Se pone el usuario actual por defecto.
 - **Tipo:** tipo de traza. Existen dos tipos: creación e inserción.
 - **Descripción:** campos que han sido modificados/añadidos que han provocado la traza.
- **Adjuntos:** se permite añadir un adjunto al activo o ver los que ya se han incluido.

IU Final

The screenshot displays the 'Gestión de activos' user interface. At the top, there are two filter boxes: 'Ambito' with 'Sevilla' selected and 'Localizacion' with 'Pabellon de Mexico - Planta baja - Recepcion' selected. Below these is the 'Activo' form, which is divided into 'Informacion' and 'Detalles' tabs. The 'Informacion' tab is active, showing fields for ID (6), NS (ns6), IDR, Estado (Activo), Modelo (Mark 2 - Model 1 - Net User), OU, Localizacion (Sevilla - Pabellon de Mexico - Planta baja - Recepcion), Responsable (admininv), and Activo padre. To the left of the 'Activo' form is a 'Navegador' panel with a tree view of assets (ns1 id1 to ns6) and navigation buttons. Below the 'Activo' form is a menu with buttons for 'Propiedades', 'Recursos Hijo', 'Historico', 'Traza', and 'Adjuntos'. At the bottom, there is an 'Atributos' table with columns for 'Nombre', 'Valor', and 'Unidad'. The table contains two rows: 'Hostname' with a value field and '1' in the 'Unidad' column, and 'Alias' with a value field and '1' in the 'Unidad' column. Each row has a 'Modificar' button.

Nombre	Valor	Unidad	
Hostname	<input type="text"/>	1	Modificar
Alias	<input type="text"/>	1	Modificar

Ilustración 186: Interfaz de usuario Gestión de activos



- Pestaña Propiedades:

Nombre	Valor	Unidad	
Hostname		1	Modificar
Alias		1	Modificar

Ilustración 187: Interfaz de usuario Gestión de activos. Pestaña Propiedades

- Pestaña Recursos Hijo:

Marca y Modelo	Responsable	NS	IDR	Ed.Activos	
Mark 1 - Model 1	admininv	ns1	id1	Ver	Modificar

Ilustración 188: Interfaz de usuario Gestión de activos. Pestaña Recursos Hijo

- Pestaña Histórico:

Fecha y Hora	Usuario	Anotacion	
2021-01-08 17:44:22	gestinv	cable roto	
2021-01-08 17:44:30	gestinv		Añadir

Ilustración 189: : Interfaz de usuario Gestión de activos. Pestaña Histórico

- Pestaña Traza:

Fecha y Hora	Usuario	Tipo	Descripcion
2020-12-21 18:25:02	gestinv	Insercion	Modelo: Mark 2 - Model 1 SN: ID: Responsable: admininv Localizacion: Sevilla - Pabellon de Mexico - Planta baja - Recepcion
2020-12-21 18:35:06	gestinv	Modificacion	SN: ns6

Ilustración 190: : Interfaz de usuario Gestión de activos. Pestaña Traza

- Pestaña Adjuntos:



Ilustración 191: Interfaz de usuario Gestión de activos. Pestaña Adjuntos

Notas

- Solo se mostrarán aquellos ámbitos que tenga permisos el gestor de inventario acorde con la interfaz *Permisos*.

4.5.1.4.8 Edición de activos

Tipo

- Unidad de interacción.

Descripción

La interfaz *Edición de activos* permite al gestor de inventario modificar y generar un informe de un activo. Esta interfaz es accedida mediante enlaces a través de las interfaces: *Gestión de entradas*, *Gestión de salidas*, *Gestión de inventario*, *Gestión de inventario por tipo de recurso* y *Gestión de activos*.

El diseño de esta interfaz es igual al descrito en la interfaz *Gestión de activos*, pero sin los combos ya que es accedida mediante enlaces.

IU Final

Ilustración 192: Interfaz de usuario Edición de activos

Notas

- Solo se mostrarán aquellas localizaciones que tenga permisos el gestor de inventario acorde con la interfaz *Permisos*.

4.5.1.4.9 Tipos de activos por sede

Tipo



- Unidad de interacción.

Descripción

La interfaz *Tipos de activos por sede* permite al gestor de inventario ver en gráfica los distintos tipos de activos que existen en una sede.

En la parte superior, se elige la sede sobre la que quiere realizar la búsqueda.

En la parte inferior, se muestran tantos en gráfica como en tabla, el número de activos que existen por tipo de activo según la sede seleccionada.

IU Final

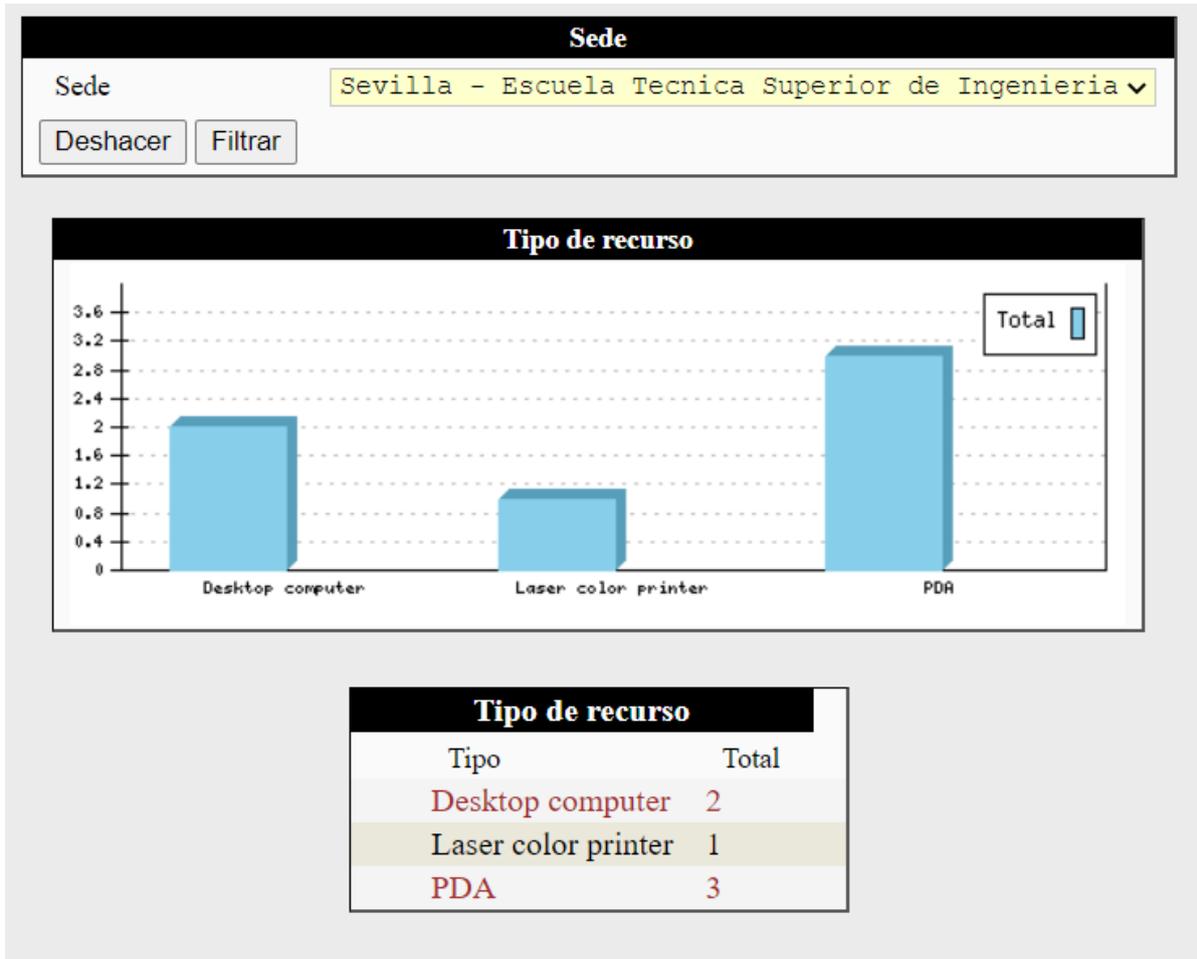


Ilustración 193: Interfaz de usuario Tipos de activos por sede

Notas

- Solo se mostrarán aquellas sedes que tenga permisos el gestor de inventario acorde con la interfaz *Permisos*.

4.5.1.4.10 Estados y marcas de activos por sede y tipo

Tipo

- Unidad de interacción.

Descripción

La interfaz *Estados y marcas de activos por sede y tipo* permite al gestor de inventario ver en dos gráficas los distintos estados y marcas de activos que existen de un tipo de recurso dada una sede.

En la parte superior, se elige el tipo de recurso y la sede sobre la que quiere realizar la búsqueda.

En la parte inferior, se muestran tanto en gráficas como en tablas, el número de activos que existen por estado y por marca de activo según el tipo de recurso y la sede seleccionada.

IU Final



Ilustración 194: Interfaz de usuario Estados y marcas de activos por sede y tipo

Notas

- Solo se mostrarán aquellas sedes que tenga permisos el gestor de inventario acorde con la interfaz *Permisos*.

4.5.1.4.11 Recursos asignados

Ver en el apartado 5.3.8 del Documento de Implementación del módulo de gestión de inventario.

4.5.1.5 Manual de usuario del técnico avanzado GTI.AM

En este apartado, se describe el manual de usuario del técnico avanzado de gestión de inventario (GTI.AM).

4.5.1.5.1 Funcionalidades

Las funcionalidades a las que tiene acceso el técnico avanzado son:

- **Gestión de movimientos**
 1. Gestión de entradas.
 2. Gestión de salidas.
- **Activos**
 1. Gestión de inventario reducida.
 2. Edición de activos reducida.
- **Consulta de recursos**
 1. Recursos asignados.



4.5.1.5.2 Gestión de entradas

Ver en el apartado 5.4.3 del Documento de Implementación del módulo de gestión de inventario.

4.5.1.5.3 Gestión de salidas

Ver en el apartado 5.4.4 del Documento de Implementación del módulo de gestión de inventario.

4.5.1.5.4 Gestión de inventario reducida

Tipo

- Unidad de interacción.

Descripción

La interfaz *Gestión de inventario reducida* permite al técnico y técnico avanzado filtrar y modificar los activos.

En la parte superior, se permite filtrar los activos que se desean buscar. Se diferencian tres pestañas de filtros: Básicos, Avanzados y Unidad Organizativa.

Los campos del filtro básico son:

- **Localización:** localización vinculada al activo.
- **Marca y Modelo:** marca y modelo del activo.
- **Responsable:** usuario responsable del activo.
- **NS:** número de serie del activo.
- **IDR(opcional):** identificador único del activo.

Los campos del filtro avanzado son:

- **Nombre equipo:** nombre del equipo asociado al activo.
- **Usuario dio de alta:** usuario que dio de alta al activo.

Los campos del filtro unidad organizativa son:

- **OU:** unidad organizativa vinculada al activo.

Una vez rellenado algunos de estos campos, se pulsa el botón Filtrar.

En la parte inferior, se muestran todos los activos que cumplen los requisitos del filtrado. Aquí, se permite realizar modificaciones de sus campos y acceder a la interfaz *Edición de activos reducida* a través del botón “Ver”. Los campos que se describen los mismos a los mencionados en el filtrado, pero con los campos Marca y Modelo, NS y IDR esta vez de lectura.

IU Final

Filtros

Localizacion
 Marca y Modelo
 Responsable
 NS
 IDR

Activo

Cant	Localizacion	OU	Marca y Modelo	Responsable	NS	IDR	Ed.Activos
------	--------------	----	----------------	-------------	----	-----	------------

Ilustración 195: Interfaz de usuario Gestión de inventario reducida

- Filtro avanzado:

Filtros

Nombre equipo
 Usuario dio de alta

Ilustración 196: Interfaz de usuario Gestión de inventario reducida. Filtro avanzado

- Filtro unidad organizativa:

Filtros

OU

Ilustración 197: Interfaz de usuario Gestión de inventario reducida. Filtro unidad organizativa

- Tabla Activo:

Activo							
Cant	Localizacion	OU	Marca y Modelo	Responsable	NS	IDR	Ed.Activos
1	Sevilla - Pabellon de Mexico - Planta baja - Recepcion	Universidad de Sevilla - Escuela Internacional de Doctorado - Unidad de alumnos					
1	Sevilla - Pabellon de Mexico - Planta baja - Recepcion						
1	Sevilla - Pabellon de Mexico - Planta baja - Recepcion						
1	Sevilla - Pabellon de Mexico - Planta baja - Recepcion						
1	Sevilla - Pabellon de Mexico - Planta baja - Recepcion						
1	Sevilla - Pabellon de Mexico - Planta baja - Recepcion						
6							



Activo	Marca y Modelo	Responsable	NS	IDR	Ed.Activos	
Doctorado - Unidad de alumnos	Mark 1 - Model 1	admininv	ns1	id1	Ver	Modificar
	Mark 1 - Model 1	admininv	ns2	id2	Ver	Modificar
	Mark 1 - Model 2	admininv	ns3	id3	Ver	Modificar
	Mark 1 - Model 2	admininv	ns4		Ver	Modificar
	Mark 1 - Model 2	admininv	ns5		Ver	Modificar
	Mark 2 - Model 1	admininv	ns6		Ver	Modificar

Ilustración 198: Interfaz de usuario Gestión de inventario reducida. Tabla Activo

Notas

- Solo se mostrarán aquellas localizaciones que tenga permisos el usuario actual acorde con la interfaz *Permisos*.

4.5.1.5.5 Edición de activos reducida

Tipo

- Unidad de interacción.

Descripción

La interfaz *Edición de activos reducida* permite al técnico y técnico avanzado modificar y generar un informe de un activo. Esta interfaz es accedida mediante enlaces a través de las interfaces: *Gestión de entradas*, *Gestión de salidas* y *Gestión de inventario reducida*.

El diseño de esta interfaz es igual al descrito en la interfaz *Edición de activos*, pero con los campos NS, IDR, Estado y Modelo en modo lectura esta vez.

IU Final

Activo

Informacion

Detalles

ID	6
NS	ns6
IDR	
Estado	Activo
Modelo	Mark 2 - Model 1 - Net User
OU	<input type="text" value=""/>
Localizacion	<input type="text" value="Sevilla - Pabellon de Mexico - Planta baja - Recepcion"/>
Responsable	<input type="text" value="admininv"/>
Activo padre	<input type="text" value=""/>

Propiedades
Recursos Hijo
Historico
Traza
Adjuntos

Activo

	Marca y Modelo	Responsable	NS	IDR	Ed.Activos
1	Mark 1 - Model 1	<input type="text" value="admininv"/>	ns1	idl	Ver <input type="button" value="Modificar"/>

Ilustración 199: Interfaz de usuario Edición de activos reducida

Notas

- Solo se mostrarán aquellas localizaciones que tenga permisos el usuario actual acorde con la interfaz *Permisos*.

4.5.1.5.6 Recursos asignados

Ver en el apartado 5.3.8 del Documento de Implementación del módulo de gestión de inventario.

4.5.1.6 Manual de usuario del técnico GTI.AM

En este apartado, se describe el manual de usuario del técnico de gestión de inventario (GTI.AM).

4.5.1.6.1 Funcionalidades

Las funcionalidades a las que tiene acceso el técnico son:

- **Activos**
 1. Gestión de inventario reducida.
 2. Edición de activos reducida.
- **Consulta de recursos**
 1. Recursos asignados.

4.5.1.6.2 Gestión de inventario reducida

Ver en el apartado 5.5.4 del Documento de Implementación del módulo de gestión de inventario.

4.5.1.6.3 Edición de activos reducida

Ver en el apartado 5.5.5 del Documento de Implementación del módulo de gestión de inventario.



4.5.1.6.4 Recursos asignados

Ver en el apartado 5.3.8 del Documento de Implementación del módulo de gestión de inventario.

4.5.1.7 Manual de usuario del usuario regular GTI.AM

En este apartado, se describe el manual de usuario del usuario regular de gestión de inventario (GTI.AM).

4.5.1.7.1 Funcionalidades

Las funcionalidades a las que tiene acceso el usuario regular son:

- **Consulta de recursos**
 1. Recursos asignados.

4.5.1.7.2 Recursos asignados

Ver en el apartado 5.3.8 del Documento de Implementación del módulo de gestión de inventario.

4.5.1.8 Manual de usuario del usuario sin acceso GTI.AM

En este apartado, se describe el manual de usuario del usuario sin acceso de gestión de inventario (GTI.AM).

4.5.1.8.1 Funcionalidades

Las funcionalidades a las que tiene acceso el usuario sin acceso son:

- **Visualización**
 1. Visualización de recurso.
 2. Visualización de entrada.
 3. Visualización de salida.

4.5.1.8.2 Visualización de recurso

Tipo

- Unidad de interacción.

Descripción

La interfaz *Visualización de recurso* permite a un usuario sin acceso ver, en modo lectura, los campos más significativos de un recurso y sus atributos.

En la parte superior, se permite ver los campos relacionados con el activo. Estos son:

- **ID**: identificador del activo en la base de datos.
- **NS**: número de serie del activo.
- **IDR**: identificador único del activo.
- **Estado**: estado del activo.
- **Modelo**: modelo del activo.

En la parte inferior, se permite ver los campos relacionados con los atributos del activo. Estos son:

- **Nombre**: nombre del atributo.

- **Valor:** valor del atributo.
- **Unidad:** unidad que complementa al valor.

IU Final

Activo	
ID	6
NS	ns6
IDR	
Estado	Activo
Modelo	Mark 2 - Model 1 - Net User

Propiedades		
Nombre	Valor	Unidad
Hostname		1
Alias		1

Ilustración 200: Interfaz de usuario Visualización de recurso

Notas

- Esta interfaz es totalmente de lectura y es accedida mediante un enlace.

4.5.1.8.3 Visualización de entrada

Tipo

- Unidad de interacción.

Descripción

La interfaz *Visualización de entrada* permite a un usuario sin acceso ver, en modo lectura, los campos más significativos de una entrada. Estos son:

- **Descripción:** nombre de la entrada.
- **Tipo:** tipo de entrada.
- **Instante:** instante de creación de la entrada.
- **Emisor:** usuario emisor de la entrada.
- **Receptor:** usuario receptor de la entrada.
- **Origen:** localización origen de la entrada.
- **Destino:** localización destino de la entrada.

IU Final



Entrada	
Descripción	input2
Tipo	Adquisición
Instante	2020-12-21 18:33
Emisor	test
Receptor	gestadqui
Origen	Sevilla - Pabellon de Mexico - Planta baja - Recepcion
Destino	Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE

Ilustración 201: Interfaz de usuario Visualización de entrada

Notas

- Esta interfaz es totalmente de lectura y es accedida mediante un enlace.

4.5.1.8.4 Visualización de salida

Tipo

- Unidad de interacción.

Descripción

La interfaz *Visualización de salida* permite a un usuario sin acceso ver, en modo lectura, los campos más significativos de una salida. Estos son:

- **Descripción:** nombre de la salida.
- **Tipo:** tipo de salida.
- **Instante:** instante de creación de la salida.
- **Emisor:** usuario emisor de la salida.
- **Receptor:** usuario receptor de la salida.
- **Origen:** localización origen de la salida.
- **Destino:** localización destino de la salida.
- **Responsable salida:** responsable de la salida.

IU Final

Salida	
Descripción	output1
Tipo	Interna
Instante	2020-12-21 18:39
Emisor	test
Receptor	test
Origen	Sevilla - Escuela Tecnica Superior de Ingenieria - -1 - SE
Destino	Córdoba - Facultad de Medicina y Enfermería - Edificio Sur - Planta Segunda
Responsable salida	root

Ilustración 202: Interfaz de usuario Visualización de salida

Notas

- Esta interfaz es totalmente de lectura y es accedida mediante un enlace.

4.5.2 Manuales de usuario de GTI.BASE

En esta sección, se recogen los manuales de usuario del módulo común de GTI. Estos contienen las distintas funcionalidades que se ofrecen para cada rol definido con sus correspondientes interfaces de usuario.

4.5.2.1 Introducción

GTI es el módulo común de GTI.

4.5.2.2 Roles

Los roles de usuario definidos en el módulo de gestión de inventario son:

- Administrador del módulo común.

4.5.2.3 Manual de usuario del administrador del módulo GTI

En este apartado, se describe el manual de usuario del administrador del módulo común.

4.5.2.3.1 Funcionalidades

Las funcionalidades a las que tiene acceso el administrador del módulo común son:

- **Gestión**
 1. Ámbitos.
 2. Unidades organizativas.
 3. Localizaciones.
- **Datos personales**
 1. Datos personales.
 2. Datos personales de usuario.

4.5.2.3.2 Ámbitos

Tipo

- Unidad de interacción.

Descripción

La interfaz *Ámbitos* permite al administrador del módulo común añadir, modificar y eliminar ámbitos. Los campos que se diferencian son:

- **Descripción:** nombre del ámbito.
- **Notas (opcional):** comentarios adicionales acerca del ámbito. Hasta 80 caracteres.

Una vez completado estos campos, se pulsa el botón Añadir.

IU Final



Ámbitos	
Descripción	Notas
0-Global	
Cádiz	
Córdoba	
Sevilla	

Ilustración 203: Interfaz de usuario Ámbitos

Notas

- La eliminación de un ámbito supone la eliminación de todos sus niveles inferiores.

4.5.2.3.3 Unidades organizativas

Tipo

- Unidad de interacción.

Descripción

La interfaz *Unidades organizativas* permite al administrador del módulo común añadir, modificar y eliminar unidades organizativas en sus tres niveles. Los campos que se diferencian son:

- **Descripción:** nombre de la unidad organizativa.
- **Notas (opcional):** comentarios adicionales acerca de la unidad organizativa. Hasta 80 caracteres.
- **Unidad Organizativa N.1/2:** unidad organizativa superior a la que pertenece la unidad organizativa actual.

Una vez completados estos campos, se pulsa el botón Añadir.

IU Final

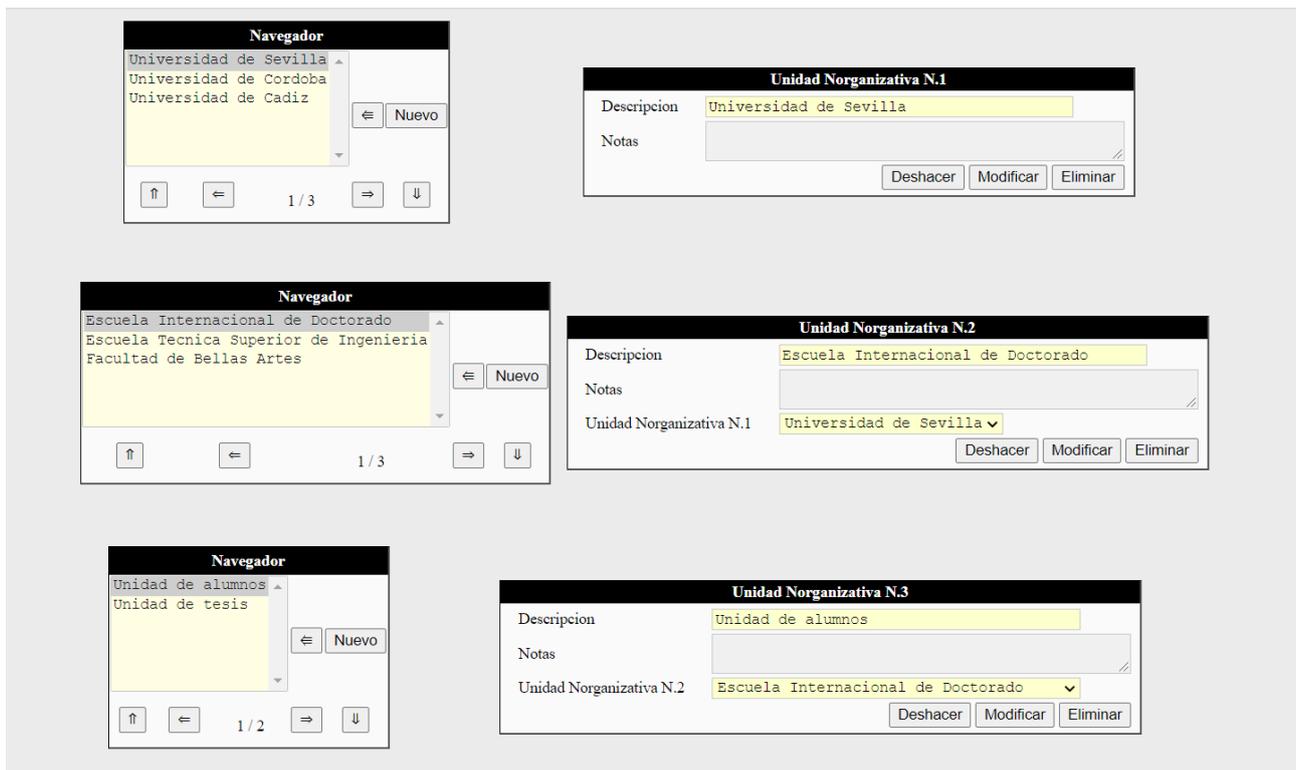


Ilustración 204: Interfaz de usuario Unidades organizativas

Notas

- La eliminación de un nivel de unidad organizativa supondrá la eliminación de todos sus niveles inferiores (entendiendo el nivel 1 como el mayor).

4.5.2.3.4 Localizaciones

Tipo

- Unidad de interacción.

Descripción

La interfaz *Localizaciones* permite al administrador del módulo común añadir, modificar y eliminar sedes y localizaciones en sus dos niveles dado un ámbito.

En el combo superior, se permite elegir el ámbito sobre el que sea trabajar.

En el resto, se permite tratar con las sedes y localizaciones. Los campos que se definen son los siguientes:

- **Descripción:** nombre de la sede/localización.
- **Notas (opcional):** comentarios adicionales acerca de la sede y localización. Hasta 80 caracteres.
- **Ámbito/Sede/Localización N.1:** nivel superior de la localización actual.

Una vez relleno estos campos, se pulsa el botón Añadir.

Adicionalmente, para la localización N.1, se permite elegir si se trata de un almacén.

IU Final

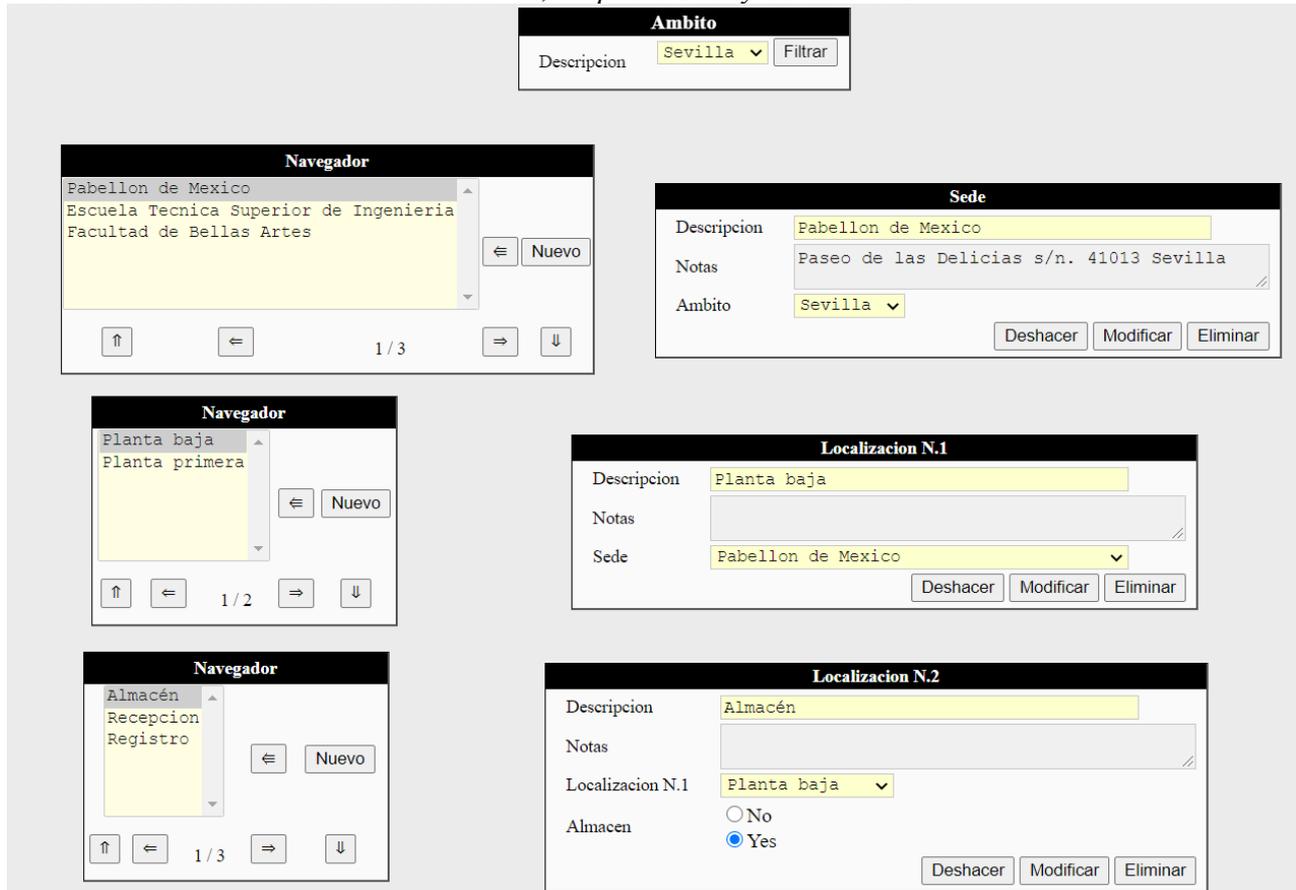


Ilustración 205: Interfaz de usuario Localizaciones

Notas

- La eliminación de un nivel superior supone la eliminación de todos sus niveles inferiores.

4.5.2.3.5 Datos personales

Tipo

- Unidad de interacción.

Descripción

La interfaz *Datos personales* permite al administrador del módulo común crear, modificar y eliminar datos personales.

En la parte superior, se permite añadir, modificar y eliminar los datos personales principales del usuario. Estos datos se dividen en tres pestañas: Personales, Foto y Otros.

La pestaña Personales contiene los siguientes campos:

- **Nombre:** nombre de la persona.
- **Apellidos:** apellidos de la persona.
- **Fecha de nacimiento (opcional):** día, mes y año de nacimiento de la persona.

La pestaña Foto contiene los siguientes campos:

- **Foto:** imagen de perfil de la persona.

La pestaña Otros contiene los siguientes campos:

- **Notas (opcional):** comentarios adicionales acerca de la persona. Hasta 80 caracteres.

Una vez relleno estos campos, se pulsa el botón Añadir.

En la parte inferior, se distinguen cuatro pestañas: Teléfono, Dirección, Email y Notas.

- **Teléfono:** permite añadir el número de teléfono de contacto de la persona
- **Dirección:** permite añadir la dirección de la persona.
- **Email:** permite añadir el correo electrónico de la persona.
- **Notas:** permite añadir comentarios adicionales acerca de la persona. Hasta 80 caracteres.

IU Final

Telefono	Descripcion	Modificar	Eliminar
655230143	teléfono móvil	Modificar	Eliminar
		Añadir	

Ilustración 206: Interfaz de usuario Datos personales

- Pestaña Teléfono:

Telefono	Descripcion	Modificar	Eliminar
655230143	teléfono móvil	Modificar	Eliminar
		Añadir	

Ilustración 207: Interfaz de usuario Datos personales. Pestaña Teléfono

- Pestaña Dirección:



Dirección	
Dirección	Descripción
Calle San Fernando, 4, 41004 Sevilla	dirección de residencia

Ilustración 208: Interfaz de usuario Datos personales. Pestaña Dirección

- Pestaña Email:

Email	
Email	Descripción
maria.gomez@gmail.com	correo electrónico personal

Ilustración 209: Interfaz de usuario Datos personales. Pestaña Email

- Pestaña Notas:

Ilustración 210: Interfaz de usuario Datos personales. Pestaña Notas

Notas

- Esta interfaz está tomada del paquete *percon* (Ver en [19]).

4.5.2.3.6 Datos personales de usuario

Tipo

- Unidad de interacción.

Descripción

La interfaz *Datos personales de usuario* permite al administrador del módulo común crear, modificar y eliminar datos personales de los usuarios registrados en la aplicación.

En la parte superior, se permite elegir el perfil a definir entre los usuarios existentes en la aplicación.

En el resto, la interfaz es igual a *Datos personales* (Ver en el apartado 4.3.5 del Documento de Implementación del módulo común).

IU Final

The screenshot shows a web application interface. At the top, there is a 'Usuario' section with a dropdown menu for 'Username' containing 'admininv' and a 'Filtrar' button. Below this, there is a 'Person' section with a header 'Person' and buttons for 'Imprimir', 'Refrescar', 'Ayuda', and 'Vista'. Underneath are tabs for 'Personales', 'Foto', and 'Otros'. The 'Personales' tab is active, showing a form with fields for 'Nombre' (Álvaro), 'Apellidos' (Marín Sánchez), 'Fecha nacimiento' (29/10/1997), and 'Usuario' (admini). At the bottom of this form are buttons for 'Deshacer', 'Modificar', and 'Eliminar'. To the left of the 'Person' section is a 'Navegador' (browser) window showing 'Marín Sánchez Álvaro' and a 'Nuevo' button. Below the 'Person' section is a navigation bar with buttons for 'Telefono', 'Direccion', 'Email', and 'Notas'. At the bottom of the screenshot is a 'Telefono' section with a table:

Telefono	Descripcion		
650543433	teléfono personal	Modificar	Eliminar
		Añadir	

Ilustración 211: Interfaz de usuario Datos personales de usuario

- Pestaña Teléfono:

This screenshot shows a close-up of the 'Telefono' tab. It features a navigation bar with buttons for 'Telefono', 'Direccion', 'Email', and 'Notas'. Below the navigation bar is a table with the following content:

Telefono	Descripcion		
650543433	teléfono personal	Modificar	Eliminar
		Añadir	

Ilustración 212: Interfaz de usuario Datos personales de usuario. Pestaña Teléfono

- Pestaña Dirección:



Dirección	
Dirección	Descripción
Calle San Fernando, 4, 41004 Sevilla	dirección de residencia

Ilustración 213: Interfaz de usuario Datos personales de usuario. Pestaña Dirección

- Pestaña Email:

Email	
Email	Descripción
alvaro.marin@gmail.com	correo electrónico personal

Ilustración 214: Interfaz de usuario Datos personales de usuario. Pestaña Email

- Pestaña Notas:

Navegador

Prueba

← Nuevo

↑ 1 / 1 ↓

Notas

Descripción: Prueba

Notas: Esto es una nota.

Deshacer Modificar Eliminar

Ilustración 215: Interfaz de usuario Datos personales de usuario. Pestaña Notas

Notas

- Esta interfaz está tomada del paquete *upercon* (Ver en [20]).

4.5.3 Manuales de usuario de GTI.PP

En esta sección, se recogen los manuales de usuario del módulo de adquisiciones de GTI. Estos contienen las distintas funcionalidades que se ofrecen para cada rol definido con sus correspondientes interfaces de usuario.

4.5.3.1 Introducción

GTI.PP (Providers and Purchases) es el módulo de adquisiciones de GTI.

4.5.3.2 Roles

Los roles de usuario definidos en el módulo de adquisiciones son:

- Gestor de adquisiciones.

- Usuario sin acceso.

4.5.3.3 Manual de usuario del gestor de adquisiciones GTI.PP

En este apartado, se describe el manual de usuario del gestor de adquisiciones.

4.5.3.3.1 Funcionalidades

Las funcionalidades a las que tiene acceso el gestor de adquisiciones son:

- **Proveedores**
 1. Gestión de proveedores.
- **Compras**
 1. Gestión de ofertas.
 2. Gestión de adquisiciones.
- **Configuración**
 1. Marcas, modelos y atributos.
- **Consulta de recursos**
 1. Recursos asignados.

4.5.3.3.2 Gestión de proveedores

Tipo

- Unidad de interacción.

Descripción

La interfaz *Gestión de proveedores* permite al gestor de adquisiciones añadir, modificar y eliminar proveedores y añadir y eliminar sus contactos relacionados.

En la parte superior, se permite crear, modificar y eliminar los proveedores. Los campos se diferencian son los siguientes:

- **Descripción:** breve descripción del proveedor.
- **Nombre:** nombre del proveedor.
- **Teléfono:** teléfono de contacto del proveedor.
- **Email:** email de contacto del proveedor.
- **Dirección (opcional):** dirección del proveedor.
- **Notas (opcional):** comentarios adicionales acerca del proveedor. Hasta 80 caracteres.

Una vez relleno estos campos, se pulsa el botón Añadir.

En la parte inferior, se encuentran dos pestañas: Contactos y Adjuntos.

- **Contactos:** permite añadir y eliminar los contactos relacionados con el proveedor seleccionado. Se diferencian los siguientes campos:
 - **Nombre:** nombre del contactos.
 - **Perfil:** botón que permite ver con detalle al contacto elegido (interfaz *upercon.struct.fview*).
- **Adjuntos:** permite añadir información en cualquier formato relacionado con el proveedor



IU Final

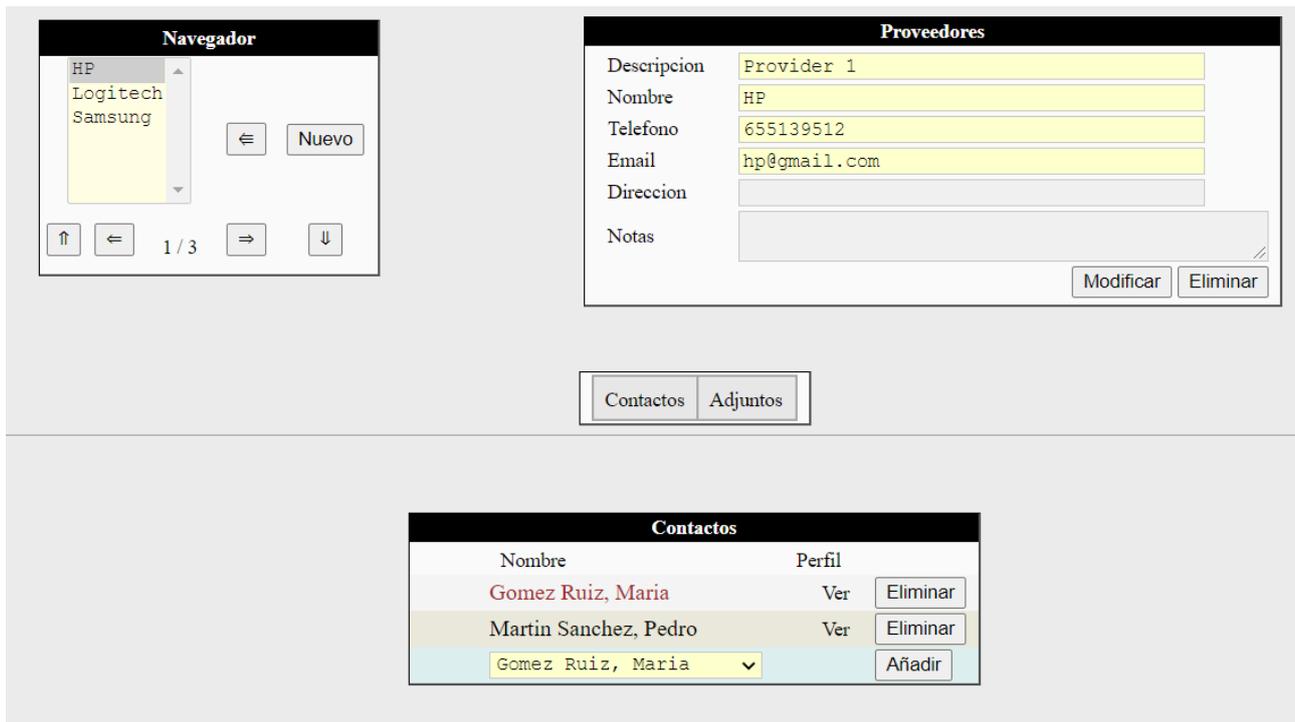


Ilustración 216: Interfaz de usuario Gestión de proveedores

- Pestaña Contactos:



Ilustración 217: Interfaz de usuario Gestión de proveedores. Pestaña Contactos

- Pestaña Adjuntos:



Ilustración 218: Interfaz de usuario Gestión de proveedores. Pestaña Adjuntos

Notas

- Los usuarios para elegir en los contactos deberán estar predefinidos para poder ser seleccionados.

4.5.3.3 Gestión de ofertas

Tipo

- Unidad de interacción.

Descripción

La interfaz *Gestión de ofertas* permite al gestor de adquisiciones añadir, modificar y eliminar peticiones de ofertas, sus ofertas y los elementos relacionados con las ofertas.

En la parte superior, se permite crear, modificar y eliminar una petición de oferta. Los campos que se definen son:

- **Descripción:** breve descripción de la petición de oferta.
- **Ámbito:** ámbito al que pertenece la petición de oferta.
- **Notas (opcional):** comentarios adicionales acerca de la petición de oferta. Hasta 80 caracteres.

Una vez completado estos campos, se pulsa el botón Añadir.

En la parte central, se permite crear, modificar y eliminar una oferta dada una petición de oferta. Los campos que se diferencian son:

- **Proveedor:** proveedor relacionado con la oferta.
- **Fecha:** día, mes y año en el que se produce la oferta. Se pone por defecto la fecha actual.
- **Descripción:** breve descripción de la oferta.
- **Notas (opcional):** comentarios adicionales acerca de la oferta. Hasta 80 caracteres.
- **Total (opcional):** estimación económica de la oferta.

Una vez relleno estos campos, se pulsa el botón Añadir.

Finalmente, en la parte inferior, se permite añadir, eliminar y modificar elementos con su correspondiente imputación económica a la oferta seleccionada. Se diferencian dos pestañas: Elementos y Adjuntos.

- **Elementos:** permite añadir elementos en modo modelos, asignándoles una cantidad junto a su precio unitario. Se diferencian los siguientes campos:



4 Construcción de los módulos de activos, adquisiciones y común de GTI

- **Modelo:** modelo del elemento que se desea añadir a la oferta.
- **Cantidad:** número de unidades del modelo seleccionado.
- **Precio Unitario:** precio por unidad del modelo seleccionado.
- **Total:** cálculo de cantidad*precio unitario. Se hace automáticamente.
- **Adjuntos:** permite añadir información en cualquier formato relacionado con la oferta seleccionada.

IU Final

The screenshot displays the 'Gestión de ofertas' interface. It includes a sidebar with a 'Navegador' (Navigator) showing a tree view with 'orcor1' and 'orsev1'. The main area contains a 'Petición de oferta' (Offer Request) form with fields for 'Descripcion' (orcor1), 'Ambito' (Córdoba), and 'Notas'. Below it is an 'Ofertas' (Offers) form with fields for 'Proveedor' (Provider 1), 'Fecha' (2020-12-21), 'Descripcion' (ocor1), and 'Total' (0.00). At the bottom, there is a 'Elementos' (Elements) table and a 'Pestaña Elementos' (Elements Tab) button.

Modelo	Cantidad	Precio Unitario	Total		
Mark 1 - Model 1	2	2.00	4.00	...	Modificar Eliminar
Mark 1 - Model 2	3	3.00	9.00	...	Modificar Eliminar
Mark 2 - Model 1	1	5.00	5.00	...	Modificar Eliminar
			18		
Mark 1 - Model 1	1	0.00		...	Añadir

Ilustración 219: Interfaz de usuario Gestión de ofertas

- Pestaña Elementos:

Elementos		Adjuntos			
Elementos					
Modelo	Cantidad	Precio Unitario	Total		
Mark 1 - Model 1 ▾	2	2.00	4.00	...	Modificar Eliminar
Mark 1 - Model 2 ▾	3	3.00	9.00	...	Modificar Eliminar
Mark 2 - Model 1 ▾	1	5.00	5.00	...	Modificar Eliminar
				18	
Mark 1 - Model 1 ▾	1	0.00		...	Añadir

Ilustración 220: Interfaz de usuario Gestión de ofertas. Pestaña Elementos

- Pestaña Adjuntos:

Elementos		Adjuntos			
Adjuntos					
Descripcion	Archivo	Tipo	Creador por	Modificado por	
offer cor1	Descargar <input type="checkbox"/> Borrar Seleccionar archivo Ningún archi... seleccionado	pp.offer	gestadqui	gestadqui	Modificar Eliminar
	Descargar <input type="checkbox"/> Borrar Seleccionar archivo Ningún archi... seleccionado	pp.offer	gestadqui	gestadqui	Añadir

Ilustración 221: Interfaz de usuario Gestión de ofertas. Pestaña Adjuntos

Notas

- Las peticiones de ofertas que aparecen están filtradas según los permisos del usuario actual en el ámbito de esta.

4.5.3.3.4 Gestión de adquisiciones

Tipo

- Unidad de interacción.

Descripción

La interfaz Gestión de adquisiciones permite al gestor de adquisiciones añadir, modificar y eliminar adquisiciones y sus elementos relacionados dadas una oferta.

En los combos superiores, se permite elegir la petición de oferta y oferta sobre la que se desea realizar la adquisición.

En la parte central, se permite añadir, modificar y eliminar una adquisición. Se distinguen los siguientes campos:

- **Fecha:** día, mes y año cuando se produce la adquisición. Se pone por defecto la fecha actual.
- **I. Garantía (opcional):** día, mes y año de inicio de garantía de la adquisición.
- **F. Garantía (opcional):** día, mes y año de fin de garantía de la adquisición.
- **CN:** número de contrato de la adquisición.



4 Construcción de los módulos de activos, adquisiciones y común de GTI

- **Descripción:** breve descripción de la adquisición.
- **Notas (opcional):** comentarios adicionales acerca de la adquisición. Hasta 80 caracteres.
- **Total (opcional):** estimación económica de la adquisición.

Una vez relleno estos campos, se pulsa el botón Añadir.

En la parte inferior, se permite añadir, modificar y eliminar elementos relacionados de la adquisición. Se diferencian dos pestañas: Elementos y Adjuntos.

- **Elementos:** permite añadir elementos en modo modelos, asignándoles una cantidad junto a su precio unitario. Se diferencian los siguientes campos:
 - **Modelo:** modelo del elemento que se desea añadir a la adquisición.
 - **Cantidad:** número de unidades del modelo seleccionado.
 - **Precio Unitario:** precio por unidad del modelo seleccionado.
 - **Total:** cálculo de cantidad*precio unitario. Se hace automáticamente.
- **Adjuntos:** permite añadir información en cualquier formato relacionado con la adquisición seleccionada.

IU Final

The screenshot displays the 'Gestión de adquisiciones' interface. At the top, there are three main sections: 'Petición de oferta' with a description field and a filter button; 'Ofertas' showing a list of offers with a filter button; and 'Compras' providing details for a specific purchase, including date, warranty periods, CN, description, notes, and total amount. A 'Navegador' sidebar on the left shows the current date and navigation controls. At the bottom, there are two tabs: 'Elementos' and 'Adjuntos'. The 'Elementos' tab is active, showing a table with columns for Model, Quantity, Unit Price, and Total, along with 'Modificar' and 'Eliminar' buttons for each row. A 'Añadir' button is at the bottom of the table.

Modelo	Cantidad	Precio Unitario	Total		
Mark 1 - Model 1	2	2.00	4.00	...	Modificar Eliminar
Mark 1 - Model 2	3	3.00	9.00	...	Modificar Eliminar
Mark 2 - Model 1	1	5.00	5.00	...	Modificar Eliminar
Mark 1 - Model 1	1	0.00		...	Añadir

Ilustración 222: Interfaz de usuario Gestión de adquisiciones

- Pestaña Elementos:

Elementos					
Modelo	Cantidad	Precio Unitario	Total		
Mark 1 - Model 1	2	2.00	4.00	...	Modificar Eliminar
Mark 1 - Model 2	3	3.00	9.00	...	Modificar Eliminar
Mark 2 - Model 1	1	5.00	5.00	...	Modificar Eliminar
				18	
Mark 1 - Model 1	1	0.00		...	Añadir

Ilustración 223: Interfaz de usuario Gestión de adquisiciones. Pestaña Elementos

- Pestaña Adjuntos:

Adjuntos				
Descripcion	Archivo	Tipo	Creador por	Modificado por
purchase cor1	<input type="checkbox"/> Descargar <input type="checkbox"/> Borrar Seleccionar archivo Ningún archi... seleccionado	pp.purch	gestadqui	gestadqui
	<input type="checkbox"/> Descargar <input type="checkbox"/> Borrar Seleccionar archivo Ningún archi... seleccionado	pp.purch	gestadqui	gestadqui
				Añadir

Ilustración 224: Interfaz de usuario Gestión de adquisiciones. Pestaña Adjuntos

Notas

- Las peticiones de ofertas que aparecen están filtradas según los permisos del usuario actual en el ámbito de esta.
- Al hacer una adquisición de una oferta, sus elementos se copiarán en la dicha adquisición.

4.5.3.3.5 Marcas, modelos y atributos

Ver en el apartado 5.4.2 del Documento de Implementación del módulo de gestión de inventario.

4.5.3.3.6 Recursos asignados

Ver en el apartado 5.3.8 del Documento de Implementación del módulo de gestión de inventario.

4.5.3.4 Manual de usuario del usuario sin acceso GTI.PP

En este apartado, se describe el manual de usuario del usuario sin acceso.

4.5.3.4.1 Funcionalidades

Las funcionalidades a las que tiene acceso el usuario sin acceso son:



- **Visualización**

1. Visualización de adquisición.

4.5.3.4.2 Visualización de adquisición

Tipo

- Unidad de interacción.

Descripción

La interfaz *Visualización de adquisición* permite a un usuario sin acceso ver, en modo lectura, los campos más significativos de una adquisición. Estos son:

- **Fecha:** fecha de creación de la adquisición.
- **I. Garantía:** fecha de inicio de garantía de la adquisición.
- **F. Garantía:** fecha de fin de garantía de la adquisición.
- **CN:** número de contrato de la adquisición.

IU Final

Adquisicion	
Fecha	2020-12-21
I. Garantia	2020-12-22
F. Garantia	2021-01-22
CN	cn1

Ilustración 225: Interfaz de usuario Visualización de adquisición

Notas

- Esta interfaz es totalmente de lectura y es accedida mediante un enlace.

5 Implantación de los módulos de activos, adquisiciones y común

5.1 Introducción

Este documento presenta el procedimiento de despliegue, así como el contenido de los paquetes y scripts necesarios para la correcta instalación del módulo de gestión de activos de GTI.

5.2 Procedimiento de despliegue

En esta sección se detalla el procedimiento de despliegue completo para la instalación del módulo de gestión de activos. Los requisitos que se están teniendo en cuenta para esta instalación son los siguientes:

- BD winter: BD (sqlite2)
- RMIU winter: MBD (sqlite2)
- BD gti: gti (PostgreSQL 8.4)

En caso de desplegar utilizando un software base distinto el procedimiento podría diferir ligeramente y requeriría la modificación de los scripts que se detallan en este documento.

Los pasos para realizar el despliegue son los siguientes:

1. Creación de la base de datos
2. Creación de la instancia de la aplicación
3. Configuración de orígenes de datos y `scripts` para paquetes
 - `db.cfg`: usando driver `sqlite2` y base de datos DB
 - `mdb.cfg`: usando driver `sqlite2` y base de datos MDB
 - `dbupd`: usando `sqlite2` y base de datos DB
 - `mdbupd`: usando `sqlite2` y base de datos MDB
4. Instalación de la aplicación `winter` y sus dependencias
5. Reconfiguración de orígenes de datos y `scripts` para paquetes
 - `mdbupd`: usando driver `sqlite2` y base de datos DB
 - `dbupd`: usando driver `pgsql` y base de datos `gtidb`
6. Instalación del paquete `gti-base` y sus dependencias
7. Instalación del paquete `gti-pp`
8. Instalación del paquete `gti-am`



5 Implantación de los módulos de activos, adquisiciones y común

Los pasos del uno al seis del procedimiento anterior, que se corresponderían con la instalación del sistema base de GTI junto con sus dependencias, se han implementado en el script `gti.install` que se detalla a continuación:

```
PPDIR=gti.stuff

DBOPTS="-U waine"
DATABASE=gtidb

WDIR=/usr/local/lib/waine-0.5.10
INSTANCE=gti

function message(){
    echo -e "\n=====
    echo $1 - `date +%F %T.%N`
    echo -e "=====
}

#####
# 1. Creacion de la base de datos
#####

message "Creating database $DATABASE"

dropdb $DBOPTS $DATABASE
createdb $DBOPTS $DATABASE

(
cat << EOF
    CREATE LANGUAGE plpgsql;
EOF
) | psql $DBOPTS $DATABASE

#####
# 2. Creacion de la instancia de la aplicaion
#####

message "Creating application instance $INSTANCE"
rm -rf $INSTANCE
$WDIR/bin/mkapp $INSTANCE
```

```
#####  
# 3. Configuracion de origenes de datos y utilidades para paquetes  
#####
```

```
message "Setting sqlite DB and MDB for winter in $INSTANCE"
```

```
sed -i 's/sqlite3/sqlite/g' $INSTANCE/_CONF/DOM/db.cfg  
sed -i 's/sqlite3/sqlite/g' $INSTANCE/_CONF/DOM/mdb.cfg  
rm $INSTANCE/packages/bin/dbupd  
rm $INSTANCE/packages/bin/mdbupd  
pushd .  
cd $INSTANCE/packages/bin/  
ln -s dbupd.sqlite2 dbupd  
ln -s mdbupd.sqlite2 mdbupd  
popd
```

```
#####  
# 4. Instalacion de la aplicacion winter y sus dependencias  
#####
```

```
message "Installing winter on $INSTANCE"
```

```
winter.install
```

```
#####  
# 5. Reconfiguracion de origenes de datos y utilidades para paquetes  
#####
```

```
message "Setting sqlite DB and MDB for gti in $INSTANCE"
```

```
sed -i 's-/MDB-/DB-g' $INSTANCE/packages/bin/mdbupd  
sed -i s/"OPTIONS=""/OPTIONS=""$DBOPTS"/g $INSTANCE/packages/bin/dbupd.pgsql  
sed -i s/"DATABASE=""/DATABASE=""$DATABASE"/g $INSTANCE/packages/bin/dbupd.pgsql  
rm $INSTANCE/packages/bin/dbupd  
pushd .  
cd $INSTANCE/packages/bin/  
ln -s dbupd.pgsql dbupd  
popd
```

```
#####  
# 6. Instalacion del paquete gti-base y sus dependencias  
#####
```

```
message "Installing gti-base on $INSTANCE"
```

```
gti-base.install
```

Una vez se ha instalado el paquete base se realiza la instalación del paquete gti-pp.



```
/usr/local/lib/waine-0.5.10/bin/wpkg install APP.gti-pp_1.0.wpk gti
```

Finalmente se realizaría el último paso, la instalación del paquete `gti-am`.

```
/usr/local/lib/waine-0.5.10/bin/wpkg install APP.gti-am_1.0.wpk gti
```

Como se puede apreciar en código del script `gti.install`, se invoca a otros dos scripts: `winter.install` y `gti-base.install`. El primero de ellos dado que no presenta relevancia para este proyecto se presenta en el anexo 7.1. El script `gti-base.install` se muestra en la siguiente sección.

5.3 Paquete `gti-base`

El paquete `gti-base` versión 1.2 contiene los elementos básicos y dependencias comunes del sistema GTI. La relación de artefactos suministrados con el paquete es la siguiente:

- `ASL/gti.asl`: código ASL con elementos e inclusiones comunes del APM.
- `doc/gti_CREATE.sql`: código SQL con definiciones de elementos comunes del DOM.
- `doc/gti_DATA.sql`: código SQL con inserciones de datos básicos.
- `doc/gti_DATA_sample.sql`: código SQL con inserciones de datos de ejemplo.
- `files/blob/DS/GTI.cfg`: datasource de la aplicación GTI.
- `files/_CONF/CPM/reports/gti.png`: Logo de la aplicación para informes.
- `files/g.cfg`: archivo de configuración para la página de redirección `g.php`.
- `files/g.php`: página para redirecciones a IUs de la aplicación (empleado conjuntamente con `wikitextbox`).
- `files/packages/bin/gti-mdb-update`: script de actualización tras la instalación de paquetes `gti`. Este script se encarga de eliminar opciones (como la de abandonar el programa) que ya no tienen sentido al integrarse dentro de la aplicación `winter`, así como de asignar el origen de datos `files/blob/DS/GTI.cfg` (que no se ha empleado durante el desarrollo) a formularios y otros elementos.

```
CMDPATH=`echo $0 | cut -d/ -f1`
```

```
echo GTI Updating $CMDPATH/DB
```

```
(
cat <<EOF
```

```

DELETE FROM _option WHERE url='logout.php';
DELETE FROM _option WHERE url='http://www.waine.org';

UPDATE _form SET dsource='blob/DS/GTI.cfg'
  WHERE formid LIKE 'wcaval%' AND dsource='';
UPDATE _form SET dsource='blob/DS/GTI.cfg'
  WHERE formid LIKE 'attach%' AND dsource='';
UPDATE _form SET dsource='blob/DS/GTI.cfg'
  WHERE formid LIKE 'percon%' AND dsource='';
UPDATE _form SET dsource='blob/DS/GTI.cfg'
  WHERE formid LIKE 'upercon%' AND dsource='';
UPDATE _form SET dsource='blob/DS/GTI.cfg'
  WHERE formid LIKE 'gti.%' AND dsource='';

UPDATE _field SET search=substr(search,1,length(search)-3)||';;;blob/DS/GTI.cfg'
  WHERE search LIKE 'DATA:%;';
UPDATE _field SET search=substr(search,1,length(search)-2)||';;;blob/DS/GTI.cfg'
  WHERE search LIKE 'DATA:%;%';
UPDATE _field SET search=substr(search,1,length(search)-1)||';;;blob/DS/GTI.cfg'
  WHERE search LIKE 'DATA:%;%;';

UPDATE _parameter SET value=substr(value,1,length(value)-2)||'blob/DS/GTI.cfg' || substr(value,length(value)-1,2)
  WHERE value LIKE '%DATA:%#_' AND value NOT LIKE '%DATA:%GTI.cfg%#_';

UPDATE _parameter SET value=substr(value,1,length(value)-3)||'blob/DS/GTI.cfg' || substr(value,length(value)-2,3)
  WHERE value LIKE '%DATA:%#__' AND value NOT LIKE '%DATA:%GTI.cfg%#__';

INSERT INTO _parameter(parameterid,structid,ord,name,value)
  SELECT parameterid||'b',structid,ord,'form_datasource','blob/DS/GTI.cfg'
  FROM _parameter WHERE name='form_source';

EOF
) | sqlite "$CMDPATH/DB"

```

- meta/meta.xml: Archivo con metainformación sobre el paquete: dependencias, funcionalidades ofrecidas, etc. A continuación se muestra el contenido de este fichero

```

<wpkg>
  <package>
    <name>APP.gti-base</name>
    <ver>1.2</ver>
    <author>aldelgado</author>
    <date>2020-05-10</date>
    <description>
      GTI base models and artifacts
    </description>
  </package>

  <provides>
    <func>APP.gti-base_1.2</func>
    <func>gti.ffkdescr</func>
    <func>gti.username</func>
    <func>gti.fmeta</func>
    <func>gti.OUL1</func>
  </provides>

```



5 Implantación de los módulos de activos, adquisiciones y común

```

<func>gti.OUL2</func>
<func>gti.OUL3</func>
<func>gti.VIEW_OU</func>
<func>gti.Site</func>
<func>gti.LL1</func>
<func>gti.LL2</func>
<func>gti.VIEW_Loc</func>
<func>gti.Scope</func>
<func>g.php</func>
<func>g.cfg</func>
<func>blob/DS/GTI.cfg</func>
<func>packages/bin/gti-mdb-update</func>
<func>_CONF/CPM/reports/gti.png</func>
</provides>
<depends>
<package>WAINE-0.4.8</package>
<package>APP.winter_0.3</package>
<package>CP.WDG.ascombo_1.1</package>
<package>CP.WDG.stcombo_1.0</package>
<package>CP.WDG.fstextbox_1.0</package>
<package>CP.WDG.wikitextbox_1.0</package>
<package>MM.percon_1.1</package>
<package>MM.attach_1.0</package>
<package>MMI.wcaval_0.2</package>
</depends>
</wpkg>

```

- `meta/postins.sh`: script que se ejecuta de forma posterior a la instalación del paquete. En `gti-base` este archivo se encarga de instalar los objetos básicos del modelo de datos y de configurar el origen de datos `DSRCGTI` en el fichero `_CONF/DOM/dsource.inc`. El contenido de este fichero es el siguiente:

```

echo 'Updating UIMs'
$WPKGDIR/bin/gti-mdb-update

addline '_CONF/DOM/dsource.cfg' '/// Added by wpkg $PKGNAME `date +%Y-%m-%d`'
addline '_CONF/DOM/dsource.cfg' '$DSRCGTI="blob/DS/GTI.cfg";'

echo 'Setting DB'
echo '1) install DB'
echo '2) install DB + sample data'
echo '0) not install DB'
echo
echo -n 'Select an option (1,0): '
read RESP
case "$RESP" in
  1) echo "Executing DB creation script" ;
    $WPKGDIR/bin/dbupd $SRCDIR/doc/gti_CREATE.sql ;
    echo "Executing DB data script" ;
    $WPKGDIR/bin/dbupd $SRCDIR/doc/gti_DATA.sql ;
    echo "Executing DB sample data" ;
    $WPKGDIR/bin/dbupd $SRCDIR/doc/gti_DATA_sample.sql ;
    echo "Executing wcaval creation script" ;
    $WPKGDIR/bin/dbupd $SRCDIR/doc/wcaval_CREATE.sql ;;
  2) echo "Executing DB creation script" ;

```

```

$WPKGDIR/bin/dbupd $SRCDIR/doc/gti_CREATE.sql ;
echo "Executing DB data script" ;
$WPKGDIR/bin/dbupd $SRCDIR/doc/gti_DATA.sql ;
echo "Executing wcaval creation script" ;
$WPKGDIR/bin/dbupd $SRCDIR/doc/wcaval_CREATE.sql ;;

0) echo 'DB creation scripts will be available in the doc directory' ;;
*) echo 'Unknown option. Skipping...' ;;
esac

```

- meta/preins.sh: script de ejecución previa a la instalación del paquete. En el caso concreto del paquete gti-base este archivo está vacío.

5.3.1 Instalador del paquete gti-base

A continuación se presenta el instalador que se ha implementado para el despliegue del paquete gti-base.

```

WDIR=/usr/local/lib/waine-0.5.10
PDIR=gti-base.stuff
INSTANCE=gti

function pkginst ()
{
    local TMPFILE
    TMPFILE=`mktemp` || exit 1

    echo -e "\n-----"
    echo -e Instalando paquete $1 - `date '+%F %T.%N'`
    echo -e "-----"
    echo -e "ejecutando: $WDIR/bin/wpkg install $PDIR/$1 $INSTANCE \n\n"

    cat - > $TMPFILE
    echo wpkg inputs
    echo -----
    cat $TMPFILE
    echo -e "\n\n"

    cat $TMPFILE | $WDIR/bin/wpkg install $PDIR/$1 $INSTANCE --force

    echo -e "\n\n-----"
    echo -e Fin Instalacion paquete $1 - `date '+%F %T.%N'`
    echo -e "-----\n\n"

    rm -rf $TMPFILE || exit 2
}

echo | pkginst CP.WDG.ascombo_1.1.wpk
echo | pkginst CP.WDG.stcombo_1.0.wpk
echo | pkginst CP.WDG.rowcombo_1.0.wpk
echo | pkginst CP.WDG.fstextbox_1.0.wpk
echo | pkginst CP.WDG.wikitextbox_1.0.wpk

(
cat <<EOF
y
1

```



```
EOF
) | pkginst MM.percon_1.1.wpk

(
cat <<EOF
y
1
EOF
) | pkginst MM.attach_1.0.wpk

(
cat <<EOF
y

0
EOF
) | pkginst MMI.wcaval_0.2.wpk

(
cat <<EOF
1
1
EOF
) | pkginst APP.gti-base_1.2.wpk
```

5.4 Paquete gti-pp

El paquete `gti-pp` contiene los elementos del módulo de gestión de proveedores y compras. Su contenido es el siguiente:

- `ASL/gti.pp.asl`: Código ASL del módulo.
- `doc/gtipp_CREATE.sql`: Script de creación de la BD para el módulo de proveedores.
- `doc/gtipp_DATA_sample.sql`: Script con una carga de datos de ejemplo.
- `files/_CONF/DOM/GTI/g.pp.inc`: Definición de las funciones necesarias para la redirección de enlaces de adquisiciones (empleado por `wikitextbox`).

```
<?php
function redirPurchase($IN_u,$IN_k){
return redirStruct('gti.pp.purchaseview',$IN_k);
}
```

?>

- meta/meta.xml: Archivo con metainformación sobre el paquete: dependencias, funcionalidades ofrecidas, etc. A continuación se muestra el contenido de este fichero:

```
<wpkg>
<package>
  <name>APP.gti-pp</name>
  <ver>1.0</ver>
  <author>aldelgado</author>
  <date>2021-04-03</date>
  <description>
    GTI providers and purchases module
  </description>
</package>
<provides>
  <func>APP.gti-pp_1.0</func>
  <func>gti.pp.cprovmanag1</func>
  <func>gti.pp.cpurchaseview</func>
  <func>gti.pp.fprovider</func>
  <func>gti.pp.fcontact</func>
  <func>gti.pp.fpurchase</func>
  <func>pp.Provider</func>
  <func>pp.OfferReq</func>
  <func>pp.Offer</func>
  <func>pp.Purchase</func>
  <func>pp.REL_Provider_Person</func>
  <func>pp.VIEW_offer</func>
  <func>pp.VIEW_person</func>
  <func>_CONF/DOM/GTI/g.pp.inc</func>
</provides>
<depends>
  <package>WAINE-0.4.8</package>
  <package>APP.gti-base_1.2</package>
</depends>
</wpkg>
```

- meta/postins.sh: script de ejecución posterior a la instalación del paquete. En gti-ps este archivo ejecuta el script gti-mdb-update (ver sección 3), añade la configuración para redirección de enlaces a tareas y peticiones al widget wikitextbox e instala los objetos del modelo de datos. El contenido de este fichero es el siguiente:

```
echo 'Updating UIMs'
$WPKGDIR/bin/gti-mdb-update

addline '_CONF/CPM/widgets/wikitextbox.cfg' '// Added by wpkg $PKGNAME `date +%Y-%m-%d`'
addline '_CONF/CPM/widgets/wikitextbox.cfg' '$ $TAGREPLACE[]=array(\'/([Pp][Pp].[Pp]#)([0-9]+)/\',\'<a
target=\'_blank\' href=\'g?p=pp.P${2}\''>${1}${2}</a>\');'

addline 'g.cfg' '// Added by wpkg $PKGNAME `date +%Y-%m-%d`'
addline 'g.cfg' '$REDIRFUNC->register("pp.P", "redirPurchase");'
addline 'g.cfg' "require_once '_CONF/DOM/GTI/g.pp.inc';"
```



5 Implantación de los módulos de activos, adquisiciones y común

```

echo 'Setting DB'
echo '1) install DB + sample data'
echo '2) install DB'
echo '0) not install DB'
echo
echo -n 'Select an option (1,0): '
read RESP
case "$RESP" in
  1) $WPKGDIR/bin/dbupd $SRCDIR/doc/gtipp_CREATE.sql ;
     $WPKGDIR/bin/dbupd $SRCDIR/doc/gtipp_DATA_sample.sql ;;

  2) $WPKGDIR/bin/dbupd $SRCDIR/doc/gtipp_CREATE.sql ;;

  0) echo 'DB creation scripts will be available in the doc directory' ;;

  *) echo 'Unknown option. Skipping...' ;;
esac

```

- meta/preins.sh: script de ejecución previa a la instalación del paquete. En el paquete gti-ps el archivo está vacío.

5.5 Paquete gti-am

El paquete gti-am contiene los elementos del módulo de gestión de activos. Su contenido es el siguiente:

- ASL/gti.am.asl: Código ASL del módulo de gestión de activos.
- doc/_origcode: directorio con la versión original del código ASL del módulo.
- doc/gtiam_CREATE.sql: Script de creación de BD del módulo de gestión de activos.
- doc/gtiam_DATA.sql: Script con la carga de datos básica del módulo.
- doc/gtiam_DATA_sample.sql: Script con la carga de datos de ejemplo.
- files/_CONF/CPM/reports/am-asset.fig: Informe de activo.
- files/_CONF/CPM/reports/am-input.fig: Informe de entrada.
- files/_CONF/CPM/reports/am-output.fig: Informe de salida.
- files/_CONF/DOM/GTI/g.am.inc: Definición de las funciones necesarias para la redirección de enlaces de entradas y salidas (empleado por wikitextbox).

```

<?php

function redirAsset($IN_u,$IN_k){
    return redirStruct('gti.am.cresourceview1',$IN_k);
}

function redirInput($IN_u,$IN_k){
    return redirStruct('gti.am.cinputview',$IN_k);
}

function redirOutput($IN_u,$IN_k){
    return redirStruct('gti.am.coutputview',$IN_k);
}

```

?>

- meta/meta.xml: Archivo con metainformación sobre el paquete: dependencias, funcionalidades ofrecidas, etc. A continuación se muestra el contenido de este fichero:

```
<wpkg>
<package>
  <name>APP.gti-am</name>
  <ver>1.0</ver>
  <author>aldelgado</author>
  <date>2021-04-08</date>
  <description>
  </description>
</package>
<provides>
  <func>APP.gti-am_1.0</func>
  <func>_CONF/DOM/GTI/g.am.inc</func>
  <func>_CONF/CPM/reports/am-asset.fig</func>
  <func>_CONF/CPM/reports/am-input.fig</func>
  <func>_CONF/CPM/reports/am-output.fig</func>
</provides>
<depends>
  <package>WAINE-0.4.8</package>
  <package>APP.gti-base_1.2</package>
  <package>APP.gti-pp_1.0</package>
</depends>
</wpkg>
```

- meta/postins.sh: script de ejecución posterior a la instalación del paquete. En gti-ps este archivo ejecuta el script gti-mdb-update (ver sección 3), añade la configuración para redirección de enlaces a tareas y peticiones al widget wikitextbox e instala los objetos del modelo de datos. El contenido de este fichero es el siguiente:

```
echo 'Updating UIMs'
$WPKGDIR/bin/gti-mdb-update

addline '_CONF/CPM/userwdg.inc' '/// Added by wpkg $PKGNAME `date +%Y-%m-%d`'
addline '_CONF/CPM/userwdg.inc' "require_once('widgets/gti.am.unitvalue.inc');"

addline '_CONF/CPM/widgets/wikitextbox.cfg' '/// Added by wpkg $PKGNAME `date +%Y-%m-%d`'
addline '_CONF/CPM/widgets/wikitextbox.cfg' '$ $TAGREPLACE[]=array(\'/[([Aa][Mm].[Aa])#]([0-9]+)/\',\'<a
target="_blank\" href="g?p=am.A${2}\">${1}${2}</a>\'); '
addline '_CONF/CPM/widgets/wikitextbox.cfg' '$ $TAGREPLACE[]=array(\'/[([Aa][Mm].[li])#]([0-9]+)/\',\'<a
target="_blank\" href="g?p=am.I${2}\">${1}${2}</a>\'); '
addline '_CONF/CPM/widgets/wikitextbox.cfg' '$ $TAGREPLACE[]=array(\'/[([Aa][Mm].[Oo])#]([0-9]+)/\',\'<a
target="_blank\" href="g?p=am.O${2}\">${1}${2}</a>\'); '

addline 'g.cfg' '/// Added by wpkg $PKGNAME `date +%Y-%m-%d`'
addline 'g.cfg' '$REDIRFUNC->register("am.A", "redirAsset");'
addline 'g.cfg' '$REDIRFUNC->register("am.I", "redirInput");'
addline 'g.cfg' '$REDIRFUNC->register("am.O", "redirOutput");'
addline 'g.cfg' "require_once '_CONF/DOM/GTI/g.am.inc';"

echo 'Setting DB'
```



5 Implantación de los módulos de activos, adquisiciones y común

```

echo '1) install DB + Demo data '
echo '2) install DB'
echo '0) not install DB'
echo
echo -n 'Select an option (1,0): '
read RESP
case "$RESP" in
  1) $WPKGDIR/bin/dbupd $SRCDIR/doc/gtiam_CREATE.sql ;
     $WPKGDIR/bin/dbupd $SRCDIR/doc/gtiam_DATA.sql ;
     $WPKGDIR/bin/dbupd $SRCDIR/doc/gtiam_DATA_sample.sql ;;

  2) $WPKGDIR/bin/dbupd $SRCDIR/doc/gtiam_CREATE.sql ;
     $WPKGDIR/bin/dbupd $SRCDIR/doc/gtips_DATA.sql ;;

  0) echo 'DB creation scripts will be available in the doc directory' ;;

  *) echo 'Unknown option. Skipping...' ;;
esac

```

- meta/preins.sh: script de ejecución previa a la instalación del paquete. En el paquete gti-am el archivo está vacío.

5.6 Informe de despliegue

Como culminación de este documento se presenta el informe de despliegue generado por la aplicación `gti.install` y por la instalación de los paquete `gti-pp` (sección 4) y paquete `gti-am` (sección 5).

5.6.1 Informe de despliegue de gti.install

En esta sección se presenta la salida de la aplicación `gti.install` (sección 2). Como se puede apreciar por los tiempos de ejecución ésta se ha realizado sobre un entorno prácticamente ideal, un disco RAM.

```

=====
Creating database gtidb - 2021-04-13 14:05:33.445133080
=====

DROP DATABASE
CREATE DATABASE
CREATE LANGUAGE

=====
Creating application instance gti - 2021-04-13 14:05:42.710466837
=====

creating dir gti
making link struct.php
making link elstruct.php
making link workflow.php
making link form.php
making link login.php
making link logout.php
making link loginev.php
making link menu.php

```

making link plot.php
making link print.php
making link action.php
making link js
creating new dir tmp
creating new dir blob
creating new dir etc
creating _CONF from defaults
creating hlp from defaults
creating packages from defaults
Insufficient privilege to change app owner to www-data:www-data
Changing app og+w

=====
Setting sqlite DB and MDB for winter in gti - 2021-04-13 14:05:42.798536798
=====

/mnt/ramdisk/gtiins /mnt/ramdisk/gtiins
/mnt/ramdisk/gtiins

=====
Installing winter on gti - 2021-04-13 14:05:42.805508307
=====

Instalando paquete CP.WDG.codemirror_1.0.wpk - 2021-04-13 14:05:42.810168958

ejecutando: /usr/local/lib/waine-0.5.6/bin/wpkg install winter.stuff/CP.WDG.codemirror_1.0.wpk gti

wpkg inputs

Installing winter.stuff/CP.WDG.codemirror_1.0.wpk
modified _CONF/CPM/userwdg.inc: addline // Added by wpkg CP.WDG.codemirror_1.0 2021-04-13
modified _CONF/CPM/userwdg.inc: addline require_once('widgets/codemirror-5.31.0.inc');
done !

Javascript based syntax highlight code editor built over CodeMirror 5.31.0

Fin Instalacion paquete CP.WDG.codemirror_1.0.wpk - 2021-04-13 14:05:42.982165186

Instalando paquete MMI.coderepo_0.1.wpk - 2021-04-13 14:05:42.987696460



5 Implantación de los módulos de activos, adquisiciones y común

```
ejecutando: /usr/local/lib/waine-0.5.6/bin/wpkg install winter.stuff/MMI.coderepo_0.1.wpk gti
```

```
wpkg inputs
```

```
-----
```

```
y  
ASL  
ASL  
1
```

```
Installing winter.stuff/MMI.coderepo_0.1.wpk
```

```
This package requires functional gti/packages/bin/mdbupd , gti/packages/bin/dbupd, gti/packages/bin/wmcoderpl and gti/packages/bin/wmcoderpl
```

```
Are these scripts ready ? (y/N): Found instantiable file /tmp/tmp.IXoOZR2063/MMI.coderepo_0.1/ASL/coderepo.aslt
```

```
Creating instance ASL
```

```
Found instantiable file /tmp/tmp.IXoOZR2063/MMI.coderepo_0.1/doc/coderepo_CREATE.sqlt
```

```
Creating instance ASL
```

```
working with ASL in /tmp/tmp.IXoOZR2063/MMI.coderepo_0.1/ASL/coderepo.ASL.asl
```

```
Creating gti/MDB with /tmp/tmp.IXoOZR2063/MMI.coderepo_0.1/ASL/coderepo.ASL.asl
```

```
asl2mdb.gensqlite Init
```

```
asl2mdb.Message: gti/MDB does not exist. Creating.
```

```
asl2mdb.Message: Generating sql /tmp/tmp.zSstgp2104.
```

```
asl2mdb.Message: Generating sqlite file gti/MDB.
```

```
modified _CONF/DOM/userfunc.inc: addline // Added by wpkg MMI.coderepo_0.1 2021-04-13
```

```
modified _CONF/DOM/userfunc.inc: addline require_once('_CONF/DOM/coderepo.inc');
```

```
Available options:
```

```
1) install coderepo database
```

```
0) not install DB
```

```
Select an option (1,2,0): Processing /tmp/tmp.IXoOZR2063/MMI.coderepo_0.1/doc/coderepo_CREATE.ASL.sql
```

```
Updating gti/DB with /tmp/tmp.IXoOZR2063/MMI.coderepo_0.1/doc/coderepo_CREATE.ASL.sql
```

```
done !
```

```
Instantiable code repository package
```

```
-----  
Fin Instalacion paquete MMI.coderepo_0.1.wpk - 2021-04-13 14:05:43.167108109  
-----
```

```
-----  
Instalando paquete MMI.dscoderepo_0.1.wpk - 2021-04-13 14:05:43.172843811  
-----
```

```
ejecutando: /usr/local/lib/waine-0.5.6/bin/wpkg install winter.stuff/MMI.dscoderepo_0.1.wpk gti
```

```
wpkg inputs
```

```
-----
```

```
y  
SQL
```

SQL
blob/DS
cfg
1

Installing winter.stuff/MMI.dscoderepo_0.1.wpk

This package requires functional gti/packages/bin/mdbupd , gti/packages/bin/dbupd, gti/packages/bin/wmcodeprl and gti/packages/bin/wmcodeprl

Are these scripts ready ? (y/N): Found instantiable file

/tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/ASL/dscoderepo.aslt

Creating instance SQL

Found instantiable file /tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/doc/dscoderepo_CREATE.sqlt

Creating instance SQL

Processing macros in /tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/ASL/dscoderepo.SQL.asl

Found macro DIR (Directory to store datasource files) in

/tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/ASL/dscoderepo.SQL.asl

Found macro EXT (Default extension for datasource files) in

/tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/ASL/dscoderepo.SQL.asl

working with ASL in /tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/ASL/dscoderepo.SQL.asl

Updating gti/MDB with /tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/ASL/dscoderepo.SQL.asl

asl2mdb.gensqlite Init

asl2mdb.Message: gti/MDB already exists. Appending.

asl2mdb.Message: Generating sql /tmp/tmp.VSwjhf2198.

asl2mdb.Message: Generating sqlite file gti/MDB.

modified _CONF/DOM/userfunc.inc: addline // Added by wpkg MMI.dscoderepo_0.1 2021-04-13

modified _CONF/DOM/userfunc.inc: addline require_once('_CONF/DOM/dscoderepo.inc');

Available options:

1) install dscoderepo database

0) not install DB

Select an option (1,0): Processing /tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/doc/dscoderepo_CREATE.SQL.sql

Updating gti/DB with /tmp/tmp.jRETr2151/MMI.dscoderepo_0.1/doc/dscoderepo_CREATE.SQL.sql

done !

Instantiable code repository with datasource selection package

Fin Instalacion paquete MMI.dscoderepo_0.1.wpk - 2021-04-13 14:05:43.352920162

Instalando paquete APP.winter_0.3.wpk - 2021-04-13 14:05:43.357588134

ejecutando: /usr/local/lib/waine-0.5.6/bin/wpkg install winter.stuff/APP.winter_0.3.wpk gti

wpkg inputs



5 Implantación de los módulos de activos, adquisiciones y común

1
1

```
Installing winter.stuff/APP.winter_0.3.wpk
working with ASL in /tmp/tmp.Qoeqk2244/APP.winter_0.3/ASL/winter_0.3.asl
Updating gti/MDB with /tmp/tmp.Qoeqk2244/APP.winter_0.3/ASL/winter_0.3.asl
asl2mdb.gensqlite Init
asl2mdb.Message: gti/MDB already exists. Appending.
asl2mdb.Message: Generating sql /tmp/tmp.Hazrns2269.
asl2mdb.Message: Generating sqlite file gti/MDB.
modified _CONF/CPM/userwdg.inc: addline // Added by wpkg APP.winter_0.3 2021-04-13
modified _CONF/CPM/userwdg.inc: addline $WDGPOOL->add("fum.pwbox",$WDGPOOL->get("passwordbox"));
Setting DB
1) install DB
0) not install DB

Select an option (1,0): Updating gti/DB with /tmp/tmp.Qoeqk2244/APP.winter_0.3/doc/mdb_0.5.6.db.sql
Updating gti/DB with /tmp/tmp.Qoeqk2244/APP.winter_0.3/doc/winter_0.2.db.sql

Codemirror widget config
1) use codemirror widget
0) do not use codemirror

Select an option (1,0):
  sqlite command (default:sqlite ):
  sqlite command (default:/MDB):
updating sqlite gti//MDB - sqlite gti//MDB /tmp/tmp.Qoeqk2244/APP.winter_0.3/doc/mdb.codemirror.upd.sql

Setting permissions

Setting fdm and fpm datasources

done !

Waine INTEgrator (0.5.6 MDB)

-----
Fin Instalacion paquete APP.winter_0.3.wpk - 2021-04-13 14:05:43.660930740
-----

=====
Setting sqlite DB and MDB for gti in gti - 2021-04-13 14:05:43.664127346
=====

/mnt/ramdisk/gtiins /mnt/ramdisk/gtiins
/mnt/ramdisk/gtiins
```

=====
Installing gti-base on gti - 2021-04-13 14:05:43.670546552
=====

Instalando paquete CP.WDG.ascombo_1.1.wpk - 2021-04-13 14:05:43.675188697

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/CP.WDG.ascombo_1.1.wpk gti

wpkg inputs

Installing gti-base.stuff/CP.WDG.ascombo_1.1.wpk
modified _CONF/CPM/userwdg.inc: addline // Added by wpkg CP.WDG.ascombo_1.1 2021-04-13
modified _CONF/CPM/userwdg.inc: addline require_once('widgets/ascombo.inc');
done !

Alternate Search Combo

Fin Instalacion paquete CP.WDG.ascombo_1.1.wpk - 2021-04-13 14:05:43.719743472

Instalando paquete CP.WDG.stcombo_1.0.wpk - 2021-04-13 14:05:43.724344718

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/CP.WDG.stcombo_1.0.wpk gti

wpkg inputs

Installing gti-base.stuff/CP.WDG.stcombo_1.0.wpk
modified _CONF/CPM/userwdg.inc: addline // Added by wpkg CP.WDG.stcombo_1.0 2021-04-13
modified _CONF/CPM/userwdg.inc: addline require_once('widgets/stcolcombo.inc');
modified _CONF/CPM/userwdg.inc: addline require_once('widgets/stcombo.inc');
done !

A combo box with allowed transitions

Fin Instalacion paquete CP.WDG.stcombo_1.0.wpk - 2021-04-13 14:05:43.769235758



Instalando paquete CP.WDG.rowcombo_1.0.wpk - 2021-04-13 14:05:43.774250544

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/CP.WDG.rowcombo_1.0.wpk gti

wpkg inputs

Installing gti-base.stuff/CP.WDG.rowcombo_1.0.wpk
modified _CONF/CPM/userwdg.inc: addline // Added by wpkg CP.WDG.rowcombo_1.0 2021-04-13
modified _CONF/CPM/userwdg.inc: addline require_once('widgets/rowcombo.inc');
done !

A combo with diferent values **for** earch row

Fin Instalacion paquete CP.WDG.rowcombo_1.0.wpk - 2021-04-13 14:05:43.821238128

Instalando paquete CP.WDG.fstextbox_1.0.wpk - 2021-04-13 14:05:43.825910356

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/CP.WDG.fstextbox_1.0.wpk gti

wpkg inputs

Installing gti-base.stuff/CP.WDG.fstextbox_1.0.wpk
modified _CONF/CPM/userwdg.inc: addline // Added by wpkg CP.WDG.fstextbox_1.0 2021-04-13
modified _CONF/CPM/userwdg.inc: addline require_once('widgets/fstextbox.inc');
done !

A textbox widget displaying always the same width and height

Fin Instalacion paquete CP.WDG.fstextbox_1.0.wpk - 2021-04-13 14:05:43.871115133

Instalando paquete CP.WDG.wikitextbox_1.0.wpk - 2021-04-13 14:05:43.875705133

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/CP.WDG.wikitextbox_1.0.wpk gti

wpkg inputs

Installing gti-base.stuff/CP.WDG.wikitextbox_1.0.wpk

modified _CONF/CPM/userwdg.inc: addline // Added by wpkg CP.WDG.wikitextbox_1.0 2021-04-13

modified _CONF/CPM/userwdg.inc: addline require_once('widgets/wikitextbox.inc');

done !

Wiki configurable textbox

Fin Instalacion paquete CP.WDG.wikitextbox_1.0.wpk - 2021-04-13 14:05:43.920372767

Instalando paquete MM.percon_1.1.wpk - 2021-04-13 14:05:43.925022170

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/MM.percon_1.1.wpk gti

wpkg inputs

y
1

Installing gti-base.stuff/MM.percon_1.1.wpk

This package requires functional gti/packages/bin/mdbupd and gti/packages/bin/dbupd

Are these scripts ready ? (y/N): working with ASL in /tmp/tmp.FHlgFO2513/MM.percon_1.1/ASL/percon.asl

Updating gti/DB with /tmp/tmp.FHlgFO2513/MM.percon_1.1/ASL/percon.asl

asl2mdb.gensqlite Init

asl2mdb.Message: gti/DB already exists. Appending.

asl2mdb.Message: Generating sql /tmp/tmp.WflyRp2538.

asl2mdb.Message: Generating sqlite file gti/DB.

Available options:

1) percon_CREATE.pgsql.sql

2) percon_CREATE.sqlite.sql

0) not install DB

Select an option (1,2,0): Updating gtidb with /tmp/tmp.FHlgFO2513/MM.percon_1.1/doc/percon_CREATE.pgsql.sql

NOTICE: CREATE TABLE will create implicit sequence "percon_person_pk_seq" for serial column "percon_person.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "percon_person_pkey" for table "percon_person"



5 Implantación de los módulos de activos, adquisiciones y común

```

CREATE TABLE
NOTICE: CREATE TABLE will create implicit sequence "percon_telephone_pk_seq" for serial column
"percon_telephone.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "percon_telephone_pkey" for table
"percon_telephone"
CREATE TABLE
NOTICE: CREATE TABLE will create implicit sequence "percon_address_pk_seq" for serial column "percon_address.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "percon_address_pkey" for table "percon_address"
CREATE TABLE
NOTICE: CREATE TABLE will create implicit sequence "percon_email_pk_seq" for serial column "percon_email.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "percon_email_pkey" for table "percon_email"
CREATE TABLE
NOTICE: CREATE TABLE will create implicit sequence "percon_field_pk_seq" for serial column "percon_field.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "percon_field_pkey" for table "percon_field"
CREATE TABLE
NOTICE: CREATE TABLE will create implicit sequence "percon_remarks_pk_seq" for serial column "percon_remarks.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "percon_remarks_pkey" for table "percon_remarks"
CREATE TABLE
CREATE VIEW
CREATE VIEW
CREATE VIEW
done !
    
```

Person contact data

 Fin Instalacion paquete MM.percon_1.1.wpk - 2021-04-13 14:05:44.892417758

 Instalando paquete MM.attach_1.0.wpk - 2021-04-13 14:05:44.897503976

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/MM.attach_1.0.wpk gti

wpkg inputs

 y
 1

Installing gti-base.stuff/MM.attach_1.0.wpk
 This package requires functional gti/packages/bin/mdbupd and gti/packages/bin/dbupd
 Are these scripts ready ? (y/N): working with ASL in /tmp/tmp.GGjojA2578/MM.attach_1.0/ASL/attach.asl
 Updating gti/DB with /tmp/tmp.GGjojA2578/MM.attach_1.0/ASL/attach.asl
 asl2mdb.gensqlite Init
 asl2mdb.Message: gti/DB already exists. Appending.
 asl2mdb.Message: Generating sql /tmp/tmp.gRhobW2603.
 asl2mdb.Message: Generating sqlite file gti/DB.
 Available options:
 1) attach_CREATE.pgsql.sql
 2) attach_CREATE.sqlite.sql

0) not install DB

```
Select an option (1,2,0): Updating gtidb with /tmp/tmp.GGjojA2578/MM.attach_1.0/doc/attach_CREATE.pgsql.sql
NOTICE: CREATE TABLE will create implicit sequence "attach_category_pk_seq" for serial column "attach_category.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "attach_category_pkey" for table "attach_category"
CREATE TABLE
NOTICE: CREATE TABLE will create implicit sequence "attach_item_pk_seq" for serial column "attach_item.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "attach_item_pkey" for table "attach_item"
CREATE TABLE
CREATE FUNCTION
CREATE TRIGGER
done !
```

A package providing generic categorized attachments

Fin Instalacion paquete MM.attach_1.0.wpk - 2021-04-13 14:05:45.239070010

Instalando paquete MMI.wcaval_0.2.wpk - 2021-04-13 14:05:45.244245999

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/MMI.wcaval_0.2.wpk gti

wpkg inputs

y

0

Installing gti-base.stuff/MMI.wcaval_0.2.wpk

This package requires functional gti/packages/bin/mdbupd , gti/packages/bin/dbupd, gti/packages/bin/wmcodeinst and gti/packages/bin/wmcodepl

Are these scripts ready ? (y/N):

wcaval data model:

```
+-----+ *          1 +-----+
| Entity +-----+ | Entity_type |
+---+---+          +---+---+
| 1 |              | 1 |
|   |              |   |
| * |              | * |
+-----+ *          1 +-----+
| Entity_wcaval_value +-----+ | Entity_type_wcaval_attr |
+-----+          +-----+
```



5 Implantación de los módulos de activos, adquisiciones y común

Use an instance **for** each Entity with custom attributes **in** your schema

Found instantiable file /tmp/tmp.yUeWoW2643/MMI.wcaval_0.2/doc/wcaval_CREATE.sqlt

```
Processing macros in /tmp/tmp.yUeWoW2643/MMI.wcaval_0.2/doc/wcaval_CREATE.sql
  Found macro ENTITY_TYPE_TABLE_NAME (Entity type table name) in
/tmp/tmp.yUeWoW2643/MMI.wcaval_0.2/doc/wcaval_CREATE.sql
  Found macro ENTITY_TYPE_TABLE_PK (Entity type table primary key) in
/tmp/tmp.yUeWoW2643/MMI.wcaval_0.2/doc/wcaval_CREATE.sql
  Found macro ENTITY_TABLE_PK (Entity table primary key) in
/tmp/tmp.yUeWoW2643/MMI.wcaval_0.2/doc/wcaval_CREATE.sql
working with ASL in /tmp/tmp.yUeWoW2643/MMI.wcaval_0.2/ASL/wcaval.asl
Updating gti/DB with /tmp/tmp.yUeWoW2643/MMI.wcaval_0.2/ASL/wcaval.asl
asl2mdb.gensqlite Init
asl2mdb.Message: gti/DB already exists. Appending.
asl2mdb.Message: Generating sql /tmp/tmp.AjTKAn2684.
asl2mdb.Message: Generating sqlite file gti/DB.
modified _CONF/CPM/userwdg.inc: addline // Added by wpkg MMI.wcaval_0.2 2021-04-13
modified _CONF/CPM/userwdg.inc: addline require_once('_CONF/CPM/widgets/wcaval_0.1/wcaval.inc');
Available options:
1) install wcaval database
0) not install DB
```

Select an option (1,0): DB creation scripts will be available **in** the doc directory
done !

Waine Custom Attribute-VALUE. A package allowing custom fields **for** any Entity **in** the Domain Model

Fin Instalacion paquete MMI.wcaval_0.2.wpk - 2021-04-13 14:05:45.382144558

Instalando paquete APP.gti-base_1.2.wpk - 2021-04-13 14:05:45.387741170

ejecutando: /usr/local/lib/waine-0.5.10/bin/wpkg install gti-base.stuff/APP.gti-base_1.2.wpk gti

wpkg inputs

1
1

```
Installing gti-base.stuff/APP.gti-base_1.2.wpk
working with ASL in /tmp/tmp.jKFvyr2725/APP.gti-base_1.2/ASL/gti.asl
Updating gti/DB with /tmp/tmp.jKFvyr2725/APP.gti-base_1.2/ASL/gti.asl
asl2mdb.gensqlite Init
asl2mdb.Message: gti/DB already exists. Appending.
```

```
asl2mdb.Message: Generating sql /tmp/tmp.tbSMSR2751.
asl2mdb.Message: Generating sqlite file gti/DB.
Updating UIMs
GTI Updating gti/DB
modified _CONF/DOM/dsource.cfg: addline // Added by wpkg APP.gti-base_1.2 2021-04-13
modified _CONF/DOM/dsource.cfg: addline $DSRCGTI="blob/DS/GTI.cfg";
Setting DB
1) install DB
2) install DB + sample data
0) not install DB

Select an option (1,0): Executing DB creation script
Updating gtidb with /tmp/tmp.jKFvyr2725/APP.gti-base_1.2/doc/gti_CREATE.sql
ERROR: language "plpgsql" already exists
gti_CREATE: Creating schema
-----

(1 row)

ERROR: schema "gti" does not exist
CREATE SCHEMA
COMMENT
gti_CREATE: Creating OUs
-----

(1 row)

ERROR: table "oul1" does not exist
NOTICE: CREATE TABLE will create implicit sequence "oul1_pk_seq" for serial column "oul1.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "oul1_pkey" for table "oul1"
CREATE TABLE
COMMENT
ERROR: table "oul2" does not exist
NOTICE: CREATE TABLE will create implicit sequence "oul2_pk_seq" for serial column "oul2.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "oul2_pkey" for table "oul2"
CREATE TABLE
COMMENT
ERROR: table "oul3" does not exist
NOTICE: CREATE TABLE will create implicit sequence "oul3_pk_seq" for serial column "oul3.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "oul3_pkey" for table "oul3"
CREATE TABLE
COMMENT
gti_CREATE: Creating Scope
-----

(1 row)

ERROR: table "scope" does not exist
NOTICE: CREATE TABLE will create implicit sequence "scope_pk_seq" for serial column "scope.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "scope_pkey" for table "scope"
CREATE TABLE
gti_CREATE: Creating Locations
-----

(1 row)

ERROR: table "site" does not exist
NOTICE: CREATE TABLE will create implicit sequence "site_pk_seq" for serial column "site.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "site_pkey" for table "site"
```



5 Implantación de los módulos de activos, adquisiciones y común

```

CREATE TABLE
COMMENT
ERROR: table "II1" does not exist
NOTICE: CREATE TABLE will create implicit sequence "II1_pk_seq" for serial column "II1.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "II1_pkey" for table "II1"
CREATE TABLE
COMMENT
ERROR: table "II2" does not exist
NOTICE: CREATE TABLE will create implicit sequence "II2_pk_seq" for serial column "II2.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "II2_pkey" for table "II2"
CREATE TABLE
COMMENT
gti_CREATE: Creating Entry
-----

(1 row)

ERROR: table "entry" does not exist
NOTICE: CREATE TABLE will create implicit sequence "entry_pk_seq" for serial column "entry.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "entry_pkey" for table "entry"
ERROR: schema "pm" does not exist
ERROR: table "sptimetype" does not exist
NOTICE: CREATE TABLE will create implicit sequence "sptimetype_pk_seq" for serial column "sptimetype.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "sptimetype_pkey" for table "sptimetype"
CREATE TABLE
ERROR: table "sptime" does not exist
NOTICE: CREATE TABLE will create implicit sequence "sptime_pk_seq" for serial column "sptime.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "sptime_pkey" for table "sptime"
ERROR: relation "gti.entry" does not exist
gti_CREATE: Creating Views
-----

(1 row)

ERROR: view "view_ou" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_loc" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_loc_woscope" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_site" does not exist
CREATE VIEW
COMMENT
Executing DB data script
Updating gtidb with /tmp/tmp.jkFvyr2725/APP.gti-base_1.2/doc/gti_DATA.sql
gti_DATA: Inserting Global Location
-----

(1 row)

INSERT 1150468 1
Executing DB sample data
Updating gtidb with /tmp/tmp.jkFvyr2725/APP.gti-base_1.2/doc/gti_DATA_sample.sql
gti_DATA: Inserting OUs
-----

```

(1 row)

```
INSERT 1150469 1
INSERT 1150470 1
INSERT 1150471 1
INSERT 1150472 1
INSERT 1150473 1
INSERT 1150474 1
INSERT 1150475 1
INSERT 1150476 1
INSERT 1150477 1
INSERT 1150478 1
INSERT 1150479 1
INSERT 1150480 1
INSERT 1150481 1
INSERT 1150482 1
INSERT 1150483 1
INSERT 1150484 1
INSERT 1150485 1
INSERT 1150486 1
INSERT 1150487 1
INSERT 1150488 1
INSERT 1150489 1
INSERT 1150490 1
INSERT 1150491 1
INSERT 1150492 1
INSERT 1150493 1
INSERT 1150494 1
INSERT 1150495 1
INSERT 1150496 1
INSERT 1150497 1
INSERT 1150498 1
INSERT 1150499 1
ALTER SEQUENCE
INSERT 1150500 1
INSERT 1150501 1
INSERT 1150502 1
ALTER SEQUENCE
INSERT 1150503 1
INSERT 1150504 1
INSERT 1150505 1
INSERT 1150506 1
INSERT 1150507 1
INSERT 1150508 1
INSERT 1150509 1
INSERT 1150510 1
INSERT 1150511 1
INSERT 1150512 1
INSERT 1150513 1
INSERT 1150514 1
gti_DATA: Inserting Locations
```

(1 row)

```
INSERT 1150515 1
INSERT 1150516 1
INSERT 1150517 1
INSERT 1150518 1
```



5 Implantación de los módulos de activos, adquisiciones y común

INSERT 1150519 1
INSERT 1150520 1
INSERT 1150521 1
INSERT 1150522 1
INSERT 1150523 1
INSERT 1150524 1
INSERT 1150525 1
INSERT 1150526 1
INSERT 1150527 1
INSERT 1150528 1
INSERT 1150529 1
INSERT 1150530 1
INSERT 1150531 1
INSERT 1150532 1
INSERT 1150533 1
INSERT 1150534 1
INSERT 1150535 1
INSERT 1150536 1
INSERT 1150537 1
INSERT 1150538 1
INSERT 1150539 1
INSERT 1150540 1
INSERT 1150541 1
INSERT 1150542 1
INSERT 1150543 1
INSERT 1150544 1
INSERT 1150545 1
INSERT 1150546 1
INSERT 1150547 1
INSERT 1150548 1
INSERT 1150549 1
INSERT 1150550 1
INSERT 1150551 1
INSERT 1150552 1
INSERT 1150553 1
INSERT 1150554 1
INSERT 1150555 1
INSERT 1150556 1
INSERT 1150557 1
INSERT 1150558 1
INSERT 1150559 1
INSERT 1150560 1
INSERT 1150561 1
INSERT 1150562 1
INSERT 1150563 1
INSERT 1150564 1
INSERT 1150565 1
INSERT 1150566 1
INSERT 1150567 1
INSERT 1150568 1
INSERT 1150569 1
INSERT 1150570 1
INSERT 1150571 1
INSERT 1150572 1
INSERT 1150573 1
INSERT 1150574 1
INSERT 1150575 1
INSERT 1150576 1
INSERT 1150577 1

```
INSERT 1150578 1
INSERT 1150579 1
INSERT 1150580 1
INSERT 1150581 1
INSERT 1150582 1
INSERT 1150583 1
INSERT 1150584 1
INSERT 1150585 1
INSERT 1150586 1
ALTER SEQUENCE
INSERT 1150587 1
INSERT 1150588 1
INSERT 1150589 1
INSERT 1150590 1
INSERT 1150591 1
INSERT 1150592 1
INSERT 1150593 1
INSERT 1150594 1
INSERT 1150595 1
INSERT 1150596 1
INSERT 1150597 1
INSERT 1150598 1
ALTER SEQUENCE
INSERT 1150599 1
INSERT 1150600 1
INSERT 1150601 1
INSERT 1150602 1
INSERT 1150603 1
INSERT 1150604 1
INSERT 1150605 1
```

Executing wcaval creation script

Updating gtidb with /tmp/tmp.jKFvyr2725/APP.gti-base_1.2/doc/wcaval_CREATE.sql

ERROR: language "plpgsql" already exists

ERROR: table "wcaval_regexp" does not exist

NOTICE: CREATE TABLE will create implicit sequence "wcaval_regexp_pk_seq" for serial column "wcaval_regexp.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "wcaval_regexp_pkey" for table "wcaval_regexp"

CREATE TABLE

```
INSERT 1150613 1
INSERT 1150614 1
INSERT 1150615 1
INSERT 1150616 1
INSERT 1150617 1
INSERT 1150618 1
INSERT 1150619 1
INSERT 1150620 1
INSERT 1150621 1
INSERT 1150622 1
INSERT 1150623 1
INSERT 1150624 1
```

done !

GTI base models and artifacts



5.6.2 Informe de instalación del paquete gti-pp

En esta sección se presenta la salida de la instalación del paquete `gti-pp` (sección 4).

```
Installing APP.gti-pp_1.0.wpk
Verifying dependences
  WAINE-0.4.8 ok
  APP.gti-base_1.2 ok
Dependencies ok
Checking conflicts
No conflicts
working with ASL in /tmp/tmp.HGowyd2838/APP.gti-pp_1.0/ASL/gti.pp.asl
Updating gti/DB with /tmp/tmp.HGowyd2838/APP.gti-pp_1.0/ASL/gti.pp.asl
asl2mdb.gensqlite Init
asl2mdb.Message: gti/DB already exists. Appending.
asl2mdb.Message: Generating sql /tmp/tmp.eMUSHJ2892.
asl2mdb.Message: Generating sqlite file gti/DB.
Updating UIMs
GTI Updating gti/DB
modified _CONF/CPM/widgets/wikitextbox.cfg: addline // Added by wpkg APP.gti-pp_1.0 2021-04-13
modified _CONF/CPM/widgets/wikitextbox.cfg: addline $TAGREPLACE[]=array('/([Pp])[Pp].[Pp]#)([0-9]+)/','<a
target="_blank" href="g?p=pp.P${2}">${1}${2}</a>');
modified g.cfg: addline // Added by wpkg APP.gti-pp_1.0 2021-04-13
modified g.cfg: addline $REDIRFUNC->register("pp.P", "redirPurchase");
modified g.cfg: addline require_once '_CONF/DOM/GTI/g.pp.inc';
Setting DB
1) install DB + sample data
2) install DB
0) not install DB

Select an option (1,0): Updating gtidb with /tmp/tmp.HGowyd2838/APP.gti-pp_1.0/doc/gtipp_CREATE.sql
ERROR: schema "pp" does not exist
CREATE SCHEMA
ERROR: schema "am" does not exist
ERROR: table "provider" does not exist
NOTICE: CREATE TABLE will create implicit sequence "provider_pk_seq" for serial column "provider.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "provider_pkey" for table "provider"
CREATE TABLE
COMMENT
ERROR: table "offerreq" does not exist
NOTICE: CREATE TABLE will create implicit sequence "offerreq_pk_seq" for serial column "offerreq.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "offerreq_pkey" for table "offerreq"
CREATE TABLE
COMMENT
ERROR: table "offer" does not exist
NOTICE: CREATE TABLE will create implicit sequence "offer_pk_seq" for serial column "offer.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "offer_pkey" for table "offer"
CREATE TABLE
COMMENT
ERROR: table "purchase" does not exist
NOTICE: CREATE TABLE will create implicit sequence "purchase_pk_seq" for serial column "purchase.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "purchase_pkey" for table "purchase"
CREATE TABLE
```

```

COMMENT
ERROR: table "rel_provider_person" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_provider_person_pk_seq" for serial column
"rel_provider_person.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_provider_person_pkey" for table
"rel_provider_person"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_provider_person_fkprov_key" for table
"rel_provider_person"
CREATE TABLE
COMMENT
ERROR: view "view_offer" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_person" does not exist
CREATE VIEW
COMMENT
Updating gtidb with /tmp/tmp.HGowyd2838/APP.gti-pp_1.0/doc/gtipp_DATA_sample.sql
INSERT 1150711 1
INSERT 1150712 1
INSERT 1150713 1
done !

```

GTI providers and purchases module

5.6.3 Informe de instalación del paquete gti-am

En esta sección se presenta la salida de la instalación del paquete gti-am (sección 5).

```

Installing APP.gti-am_1.0.wpk
Verifying dependences
  WAINE-0.4.8 ok
  APP.gti-base_1.2 ok
  APP.gti-pp_1.0 ok
Dependences ok
Checking conflicts
No conflicts
working with ASL in /tmp/tmp.KqumUR2964/APP.gti-am_1.0/ASL/gti.am.asl
Updating gti/DB with /tmp/tmp.KqumUR2964/APP.gti-am_1.0/ASL/gti.am.asl
asl2mdb.gensqlite Init
asl2mdb.Message: gti/DB already exists. Appending.
asl2mdb.Message: Generating sql /tmp/tmp.loQGML3010.
asl2mdb.Message: Generating sqlite file gti/DB.
Updating UIMs
GTI Updating gti/DB
modified _CONF/CPM/userwdg.inc: addline // Added by wpkg APP.gti-am_1.0 2021-04-13
modified _CONF/CPM/userwdg.inc: addline require_once('widgets/gti.am.unitvalue.inc');
modified _CONF/CPM/widgets/wikitextbox.cfg: addline // Added by wpkg APP.gti-am_1.0 2021-04-13
modified _CONF/CPM/widgets/wikitextbox.cfg: addline $TAGREPLACE[]=array('/([Aa][Mm].[Aa])#([0-9]+)/','<a
target="_blank" href="g?p=am.A${2}">${1}${2}</a>');
modified _CONF/CPM/widgets/wikitextbox.cfg: addline $TAGREPLACE[]=array('/([Aa][Mm].[li])#([0-9]+)/','<a
target="_blank" href="g?p=am.I${2}">${1}${2}</a>');
modified _CONF/CPM/widgets/wikitextbox.cfg: addline $TAGREPLACE[]=array('/([Aa][Mm].[Oo])#([0-9]+)/','<a
target="_blank" href="g?p=am.O${2}">${1}${2}</a>');
modified g.cfg: addline // Added by wpkg APP.gti-am_1.0 2021-04-13
modified g.cfg: addline $REDIRFUNC->register("am.A", "redirAsset");
modified g.cfg: addline $REDIRFUNC->register("am.I", "redirInput");

```



5 Implantación de los módulos de activos, adquisiciones y común

modified g.cfg: addline \$REDIRFUNC->register("am.O", "redirOutput");

modified g.cfg: addline require_once '_CONF/DOM/GTI/g.am.inc';

Setting DB

1) install DB + Demo data

2) install DB

0) not install DB

Select an option (1,0): Updating gtidb with /tmp/tmp.KqumUR2964/APP.gti-am_1.0/doc/gtiam_CREATE.sql

ERROR: schema "am" does not exist

CREATE SCHEMA

COMMENT

ERROR: table "unittype" does not exist

NOTICE: CREATE TABLE will create implicit sequence "unittype_pk_seq" for serial column "unittype.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "unittype_pkey" for table "unittype"

CREATE TABLE

COMMENT

ERROR: table "unit" does not exist

NOTICE: CREATE TABLE will create implicit sequence "unit_pk_seq" for serial column "unit.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "unit_pkey" for table "unit"

CREATE TABLE

COMMENT

ERROR: table "regex" does not exist

CREATE TABLE

COMMENT

ERROR: table "assettype" does not exist

NOTICE: CREATE TABLE will create implicit sequence "assettype_pk_seq" for serial column "assettype.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "assettype_pkey" for table "assettype"

CREATE TABLE

COMMENT

ERROR: table "modeltypeattr" does not exist

NOTICE: CREATE TABLE will create implicit sequence "modeltypeattr_pk_seq" for serial column "modeltypeattr.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "modeltypeattr_pkey" for table "modeltypeattr"

CREATE TABLE

COMMENT

ERROR: table "assettypeattr" does not exist

NOTICE: CREATE TABLE will create implicit sequence "assettypeattr_pk_seq" for serial column "assettypeattr.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "assettypeattr_pkey" for table "assettypeattr"

CREATE TABLE

COMMENT

ERROR: table "mark" does not exist

NOTICE: CREATE TABLE will create implicit sequence "mark_pk_seq" for serial column "mark.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "mark_pkey" for table "mark"

CREATE TABLE

COMMENT

ERROR: table "model" does not exist

NOTICE: CREATE TABLE will create implicit sequence "model_pk_seq" for serial column "model.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "model_pkey" for table "model"

CREATE TABLE

COMMENT

ERROR: table "modelattr" does not exist

NOTICE: CREATE TABLE will create implicit sequence "modelattr_pk_seq" for serial column "modelattr.pk"

CREATE TABLE

COMMENT

ERROR: table "status" does not exist

NOTICE: CREATE TABLE will create implicit sequence "status_pk_seq" for serial column "status.pk"

NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "status_pkey" for table "status"

CREATE TABLE

COMMENT

ERROR: table "asset" does not exist

```
NOTICE: CREATE TABLE will create implicit sequence "asset_pk_seq" for serial column "asset.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "asset_pkey" for table "asset"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "asset_sn_key" for table "asset"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "asset_id_key" for table "asset"
CREATE TABLE
COMMENT
ERROR: table "assetattr" does not exist
NOTICE: CREATE TABLE will create implicit sequence "assetattr_pk_seq" for serial column "assetattr.pk"
CREATE TABLE
COMMENT
ERROR: table "rel_offer_model" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_offer_model_pk_seq" for serial column "rel_offer_model.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_offer_model_pkey" for table "rel_offer_model"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_offer_model_fkmodel_key" for table
"rel_offer_model"
CREATE TABLE
COMMENT
ERROR: table "rel_purchase_model" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_purchase_model_pk_seq" for serial column
"rel_purchase_model.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_purchase_model_pkey" for table
"rel_purchase_model"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_purchase_model_fkmodel_key" for table
"rel_purchase_model"
CREATE TABLE
COMMENT
ERROR: table "outputtype" does not exist
NOTICE: CREATE TABLE will create implicit sequence "outputtype_pk_seq" for serial column "outputtype.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "outputtype_pkey" for table "outputtype"
CREATE TABLE
COMMENT
ERROR: table "output" does not exist
NOTICE: CREATE TABLE will create implicit sequence "output_pk_seq" for serial column "output.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "output_pkey" for table "output"
CREATE TABLE
COMMENT
ERROR: table "rel_output_asset" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_output_asset_pk_seq" for serial column
"rel_output_asset.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_output_asset_pkey" for table "rel_output_asset"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_output_asset_fkasset_key" for table
"rel_output_asset"
CREATE TABLE
COMMENT
ERROR: table "inputtype" does not exist
NOTICE: CREATE TABLE will create implicit sequence "inputtype_pk_seq" for serial column "inputtype.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "inputtype_pkey" for table "inputtype"
CREATE TABLE
COMMENT
ERROR: table "input" does not exist
NOTICE: CREATE TABLE will create implicit sequence "input_pk_seq" for serial column "input.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "input_pkey" for table "input"
CREATE TABLE
COMMENT
ERROR: table "rel_input_asset" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_input_asset_pk_seq" for serial column "rel_input_asset.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_input_asset_pkey" for table "rel_input_asset"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_input_asset_fkasset_key" for table "rel_input_asset"
CREATE TABLE
```



5 Implantación de los módulos de activos, adquisiciones y común

```

COMMENT
ERROR: table "rel_gad_scope" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_gad_scope_pk_seq" for serial column "rel_gad_scope.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_gad_scope_pkey" for table "rel_gad_scope"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_gad_scope_fkscope_key" for table "rel_gad_scope"
CREATE TABLE
COMMENT
ERROR: table "rel_gin_scope" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_gin_scope_pk_seq" for serial column "rel_gin_scope.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_gin_scope_pkey" for table "rel_gin_scope"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_gin_scope_fkscope_key" for table "rel_gin_scope"
CREATE TABLE
COMMENT
ERROR: table "rel_ate_scope" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_ate_scope_pk_seq" for serial column "rel_ate_scope.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_ate_scope_pkey" for table "rel_ate_scope"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_ate_scope_fkscope_key" for table "rel_ate_scope"
CREATE TABLE
COMMENT
ERROR: table "rel_tec_scope" does not exist
NOTICE: CREATE TABLE will create implicit sequence "rel_tec_scope_pk_seq" for serial column "rel_tec_scope.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "rel_tec_scope_pkey" for table "rel_tec_scope"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "rel_tec_scope_fkscope_key" for table "rel_tec_scope"
CREATE TABLE
COMMENT
ERROR: table "history" does not exist
NOTICE: CREATE TABLE will create implicit sequence "history_pk_seq" for serial column "history.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "history_pkey" for table "history"
CREATE TABLE
COMMENT
ERROR: table "entry" does not exist
NOTICE: CREATE TABLE will create implicit sequence "entry_pk_seq" for serial column "entry.pk"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "entry_pkey" for table "entry"
CREATE TABLE
COMMENT
ERROR: view "view_model" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_fullmodel" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_asset" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_asset_2" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_input_assets" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_output_assets" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_input_report" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_output_report" does not exist
CREATE VIEW
COMMENT

```

```
ERROR: view "view_asset_report" does not exist
ERROR: relation "am.view_assetpurchase" does not exist
ERROR: relation "am.view_asset_report" does not exist
ERROR: view "view_typeassetnum" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_sitetype" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_statusassetnum" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_markassetnum" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_assetunit" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_assetmodelattr" does not exist
CREATE VIEW
COMMENT
ERROR: view "view_assetpurchase" does not exist
CREATE VIEW
COMMENT
CREATE FUNCTION
COMMENT
CREATE TRIGGER
CREATE FUNCTION
COMMENT
Updating gtidb with /tmp/tmp.KqumUR2964/APP.gti-am_1.0/doc/gtiam_DATA.sql
INSERT 1151205 1
INSERT 1151206 1
INSERT 1151207 1
INSERT 1151208 1
INSERT 1151209 1
INSERT 1151210 1
```



5 Implantación de los módulos de activos, adquisiciones y común

```
INSERT 1151211 1
INSERT 1151212 1
INSERT 1151213 1
INSERT 1151214 1
INSERT 1151215 1
INSERT 1151216 1
INSERT 1151217 1
INSERT 1151218 1
INSERT 1151219 1
INSERT 1151220 1
INSERT 1151221 1
INSERT 1151222 1
INSERT 1151223 1
INSERT 1151224 1
INSERT 1151225 1
INSERT 1151226 1
INSERT 1151227 1
INSERT 1151228 1
INSERT 1151229 1
INSERT 1151230 1
INSERT 1151231 1
INSERT 1151232 1
INSERT 1151233 1
INSERT 1151234 1
INSERT 1151235 1
INSERT 1151236 1
INSERT 1151237 1
INSERT 1151238 1
INSERT 1151239 1
INSERT 1151240 1
INSERT 1151241 1
INSERT 1151242 1
INSERT 1151243 1
ALTER SEQUENCE
INSERT 1151244 1
INSERT 1151245 1
ERROR: value too long for type character varying(10)
INSERT 1151246 1
INSERT 1151247 1
INSERT 1151248 1
INSERT 1151249 1
INSERT 1151250 1
INSERT 1151251 1
INSERT 1151252 1
INSERT 1151253 1
INSERT 1151254 1
INSERT 1151255 1
INSERT 1151256 1
INSERT 1151257 1
INSERT 1151258 1
INSERT 1151259 1
INSERT 1151260 1
INSERT 1151261 1
INSERT 1151262 1
INSERT 1151263 1
INSERT 1151264 1
INSERT 1151265 1
INSERT 1151266 1
ALTER SEQUENCE
```

```
Updating gtidb with /tmp/tmp.KqumUR2964/APP.gti-am_1.0/doc/gtiam_DATA_sample.sql
INSERT 1151267 1
INSERT 1151268 1
INSERT 1151269 1
INSERT 1151270 1
INSERT 1151271 1
INSERT 1151272 1
INSERT 1151273 1
INSERT 1151274 1
INSERT 1151275 1
INSERT 1151276 1
INSERT 1151277 1
INSERT 1151278 1
INSERT 1151279 1
INSERT 1151280 1
INSERT 1151281 1
ALTER SEQUENCE
INSERT 1151282 1
INSERT 1151283 1
INSERT 1151284 1
INSERT 1151285 1
INSERT 1151286 1
INSERT 1151287 1
INSERT 1151288 1
INSERT 1151289 1
INSERT 1151290 1
INSERT 1151291 1
INSERT 1151292 1
INSERT 1151293 1
INSERT 1151294 1
INSERT 1151295 1
INSERT 1151299 1
INSERT 1151300 1
INSERT 1151301 1
done !
```

5.7 Anexos

5.7.1 winter.install

En este anexo se muestra el código del programa de instalación de la aplicación `winter` en su versión 0.3

```
WDIR=/usr/local/lib/waine-0.5.10
PDIR=winter.stuff
INSTANCE=gti

function pkginst ()
{
    local TMPFILE
    TMPFILE=`mktemp` || exit 1

    echo -e "\n-----"
    echo -e Instalando paquete $1 - `date '+%F %T.%N'`
    echo -e "-----"
    echo -e "ejecutando: $WDIR/bin/wpkg install $PDIR/$1 $INSTANCE \n\n"
```



5 Implantación de los módulos de activos, adquisiciones y común

```

cat - > $TMPFILE
echo wpkg inputs
echo -----
cat $TMPFILE
echo -e "\n\n"

cat $TMPFILE | $WDIR/bin/wpkg install $PDIR/$1 $INSTANCE --force

echo -e "\n\n-----"
echo Fin Instalacion paquete $1 - `date '+%F %T.%N'`
echo -e "-----\n\n"

rm -rf $TMPFILE || exit 2
}

echo | pkginst CP.WDG.codemirror_1.0.wpk

(
cat <<EOF
y
ASL
ASL
1
EOF
) | pkginst MMI.coderepo_0.1.wpk

(
cat <<EOF
y
SQL
SQL
blob/DS
cfg
1
EOF
) | pkginst MMI.dscoderepo_0.1.wpk

(
cat <<EOF
1
1
EOF
) | pkginst APP.winter_0.3.wpk

```

6 Conclusiones

Durante el desarrollo de los módulos de inventario, adquisiciones y el módulo común de la aplicación GTI, he reforzado conocimientos adquiridos durante el transcurso del grado y también he aprendido algunos nuevos.

En primer lugar, la construcción de estos módulos haciendo uso de las fases del ciclo de vida de desarrollo de aplicaciones WAINE ha facilitado notoriamente el desarrollo del proyecto. Además, me ha permitido profundizar en los conocimientos que ya tenía del ciclo de vida del desarrollo software.

Respecto a la codificación, he extendido mi conocimiento aún más en las múltiples funcionalidad que SQL ofrece. Asimismo, el uso de la técnica de desarrollo de interfaces de usuario basado en modelos (MBUID) y el entorno de desarrollo WAINE, junto con su lenguaje ASL, ha resultado ser muy efectivo a la hora de obtener el código fuente final.

Finalmente, cabe mencionar el uso de herramientas que han sido de utilidad durante el proceso de desarrollo. Este es el caso de “Bitwise SSH client”, que integra los protocolos SSH y SFTP para la conexión remota y la transferencia de datos con el servidor de la universidad. Además, he usado “Visual Studio Code” para el manejo de código, “Draw.io” para la generación de diagramas, “Tohtml” para una visualización correcta de código fuente en los documentos, entre otros.

Con lo que respecta a WAINE, caben destacar varios conceptos:

- Es importante mencionar dos conceptos que se introducen en este entorno: tablas rol/funcionalidad y los modelos Entidad-Relación anotados. Como se puede observar en el Documento de Diseño, las tablas rol/funcionalidad recogen las unidades de interacción que han sido diseñadas para cada rol. En una primera instancia, estas interfaces se crean a partir de los modelos Entidad-Relación anotados que incluyen, para cada unidad de interacción, todos los formularios y estructuras que contiene junto a sus relaciones. Una vez hecho esto, resulta muy sencillo obtener el código ASL correspondiente.
- El lenguaje de especificación ASL ha sido clave para la creación de la aplicación web. Cabría pensar que se requiere de un conocimiento muy extenso de HTML, CSS, PHP o JavaScript para la creación de la aplicación, sin embargo, el uso del lenguaje ASL facilita considerablemente el desarrollo de una aplicación web ya que simplifica el proceso haciendo uso de conceptos más simples.

WAINE además ofrece mecanismos de reutilización de gran potencial. Durante el desarrollo de los módulos se han hecho uso, en concreto, de dos técnicas de reutilización de lenguaje: paquetes y parametrización.

Finalmente, considerando todos los aspectos mencionados en este apartado, se puede afirmar que el MBUIDE WAINE es una opción adecuada para el desarrollo de aplicaciones web gracias a su gran capacidad de reutilización, los modelos que facilitan desarrollo de interfaces y la gran capacidad de adaptación ante la aparición de nuevos escenarios en el uso de su lenguaje.

6.1 Líneas de mejora

En lo que se refiere a posibles mejoras de WAINE, sería de gran utilidad para los desarrolladores una mayor cantidad de documentación acerca del lenguaje ASL. La página oficial de WAINE ya ofrece una wiki donde se recogen algunas de las funcionalidades del lenguaje de WAINE. Sin embargo, al ser un entorno de desarrollo bastante flexible y adaptable a cambios, se requiere que esos cambios se recojan



en la wiki. Esto mejoraría notablemente la agilidad de resolución de problemas en la plataforma. Una posible solución sería ofertar un trabajo de fin de grado que documente todos los avances relacionados con el lenguaje ASL.

Para terminar, se describen algunos aspectos de posibles mejoras de los módulos desarrollados:

- **Estilo de las interfaces de usuario:** el estilo actual de interfaces es el establecido por defecto. Sería interesante cambiarlo accediendo al repositorio de configuraciones de **WAINE** para así mejorar su aspecto.
- **Chat de grupos:** se plantea un chat donde los distintos administradores, gestores y técnicos se puedan comunicarse de forma más intuitiva y sencilla.
- **Notificaciones al usuario mediante correo electrónico:** podría ser interesante notificar a los distintos usuarios cuando se le asignen un nuevo activo, o si uno de sus activos (del que es responsable) ha sufrido una modificación mediante un correo electrónico.
- **Nuevos documentos de informe:** resultaría útil poder crear informes de no sólo activos, entradas y salidas. Se sugiere, por ejemplo, la creación de un informe para el módulo de adquisiciones: las ofertas. Para cada petición de oferta, qué ofertas existen y qué elementos incluyen.
- **Extensión del módulo de adquisiciones:** de la misma forma que para el módulo común, se podría extender el módulo de adquisiciones si se quiera ofrecer esta aplicación a empresas privadas, haciendo que los modelos tengan un coste asociado en lugar de estimaciones, haciendo que las entradas y salidas se les asocie un valor económico.
- **Búsquedas por propiedades de los recursos:** añadir la posibilidad de hacer búsquedas con filtros de las propiedades tanto de tipo modelos como de tipo activos de los recursos existentes podría resultar de gran utilidad.
- **Generación de etiquetas para los recursos:** poner etiquetas en los recursos cuando se trasladen de una localización con los datos más relevantes como el ID, número de serie, responsable, etc.

7 Anexos

7.1 Fases de trabajo

Con respecto a las fases de trabajo, este proyecto ha supuesto en su mayor medida de: *Formación y Aprendizaje, Diseño e Implementación.*

La fase de *Formación y aprendizaje* ha tenido una duración de en torno a las 22 horas. Como se observa, se trata de una duración muy inferior a lo que se tardaría en tener un conocimiento básico de cualquier otro entorno de desarrollo de páginas web.

En la fase de *Diseño* se han empleado un total de 71 horas. Esto implica todo lo relacionado con los diseños de diagramas Entidad-Relación anotados, como los propios documentos de diseño realizados para cada módulo.

En la fase de *Implementación* se han empleado un total de 184 horas. Esto incluye tanto la implementación de la aplicación, como su apropiada verificación de su funcionamiento y sus manuales de usuario, para facilitar su uso.

7.2 Reutilización de elementos

La técnica de paquetes ha permitido la reutilización de elementos existentes de proyectos desarrollados con anterioridad. Esto se refleja, principalmente, en el uso del paquete *percon* y *upercon*, para el manejo de datos personales y en el uso del paquete *attach*, para el manejo de adjuntos. Esta reutilización queda reflejada en la siguiente tabla:

Nombre del formulario	Veces utilizado	Líneas	Líneas ahorradas totales
attach.form.catattach	11	42	462
percon.form.person	2	21	42
gti.ffkdescr	9	15	135
gti.username	8	8	64
fmeta.form.meta	1	13	13
fum.form.user	1	57	57
fdescr.form.descr	8	6	48
TOTAL			821

Tabla 1: Número de líneas ahorradas a través de la técnica de inclusión.

La técnica de parametrización ha permitido reutilizar muchos de los formularios definidos con solo adaptar sus parámetros en función del contenedor en el que se usan. Teniendo en cuenta que las líneas añadidas para la parametrización de un formulario son de 3 líneas de media para esta aplicación y que un formulario tiene que ser usado al menos una vez para que tenga sentido su creación, se muestra en la siguiente tabla el número de líneas ahorradas por formulario:

Nombre del formulario	Veces utilizado	Líneas	Líneas ahorradas totales
gti.am.finvetypeattr	2-1	12-3	9
gti.am.frelscopeu	4-1	10-3	21
gti.am.finput	5-1	55-3	208
gti.am.finputsummary	2-1	15-3	12
gti.am.foutput	5-1	44-3	164
gti.am.foutassetelem	2-1	29-3	26
gti.am.foutputsummary	2-1	15-3	12
gti.am.fasset1	13-1	65-3	744
gti.am.fasset2	7-1	71-3	408
gti.am.fassetADG	2-1	27-3	24
gti.pp.fofferreq	2-1	13-3	10
gti.pp.fpurchase	2-1	22-3	19
TOTAL			1657

Tabla 2: Número de líneas ahorradas a través de la técnica de parametrización.

TOTAL DE LÍNEAS REUTILIZADAS: $821 + 1657 = 2478$ líneas ahorradas.

Teniendo en cuenta que el total del líneas de código en los tres módulos supone 3133 líneas (considerando comentarios), sumando las líneas ahorradas, supondría un total de 5611 líneas. Por



tanto, supone un ahorro del **44,2%** de líneas de código con respecto al total. El resultado obtenido parece concordar con el 46,2% que se estima como porcentaje de reutilización general en una aplicación desarrollada con WAINE [14]. Esto pone de manifiesto la gran capacidad de **WAINE** para la reutilización de código que facilita y agiliza el desarrollo de una aplicación y ayuda a justificar su elección.

8 Bibliografía

- [1] ISO 55000:2014 Asset Management. Overview, principles and terminology (2014).
- [2] ISO 55001:2014 Asset Management. Management systems – Requirements (2014).
- [3] ISO 55002:2018 Asset Management. Management systems – Guidelines for the application of ISO 55001 (2018).
- [4] Amendola, L. (2016). *Gestión Integral de Activos Físicos*. 3ra Edición, Valencia, España. Editorial PMM Institute for Learning;2013. 155p-186p ISBN: 978-84-935668-8-3. Campbell, J. J., A.
- [5] Rosa Fernández (2020). *Activos bajo gestión a nivel mundial 2007-2019*. Recuperado de <https://es.statista.com/estadisticas/636429/activos-bajo-gestion-a-nivel-mundial/>.
- [6] Rosa Fernández (2020). *Principales empresas de gestión de activos por activos gestionados en el mundo 2020*. Recuperado de <https://es.statista.com/estadisticas/637106/principales-empresas-de-gestion-de-activos-en-el-mundo--por-activos-gestionados/>.
- [7] ServiceNow (2021). *ServiceNow IT Asset Management*. Recuperado de <https://www.servicenow.es/products/it-asset-management.html>.
- [8] Waspbarcode (2021). *AssetCloud*. Recuperado de <https://www.waspbarcode.com/asset-tracking>.
- [9] ManageEngine (2021). *AssetExplorer*. Recuperado de <https://www.manageengine.com/es/asset-explorer/>.
- [10] AssetPanda (2021). Recuperado de <https://www.assetpanda.com/>.
- [11] Mora Galindo, P. (2020). *Módulo de gestión de proyectos de GTI* (Trabajo Fin de Grado).
- [12] Romero Gutiérrez, J.A. (2020). *Módulo de gestión de servicios y procesos de GTI* (Trabajo Fin de Grado).
- [13] Delgado González, A.L, Wiki WAINE, Website. Recuperada de <http://waine.us.es/dokuwiki/doku.php>.
- [14] Delgado González, A.L. (2016). *Propuestas para la reutilización en el Desarrollo de Interfaces de Usuario Basado en Modelos* (Tesis Doctoral). Recuperado de <http://hdl.handle.net/11441/34239>.
- [15] Delgado González, A.L., Estepa Alonso A., Estepa Alonso R., Troyano Jiménez, J.A. (2015). *Reusing UI elements with Model-Based User Interface Development*. *Int. J. Human-Computer Studies*.
- [16] Delgado González A.L. WAINE Wiki. WAINE: *Model based web UI development*. Recuperado de <http://waine.us.es/dokuwiki/doku.php>
- [17] Delgado González A.L. WAINE: *Un sistema de desarrollo de interfaces web basado en modelos para aplicaciones de gestión*. WAINE: Model based web UI development. Recuperado de http://waine.us.es/portal/blob/frm_docs-2-iVj1Sc/DOC_mbuid_waine_intro.pdf
- [18] Delgado González, A.L., Estepa Alonso A., Estepa Alonso R (2007). WAINE: *Automatic generator of web based applications*.
- [19] Delgado González, A.L, Wiki WAINE: *Percon package*. Recuperada de <http://waine.us.es/dokuwiki/doku.php?id=waine:package:percon>
- [20] Delgado González, A.L, Wiki WAINE: *Upercon package*. Recuperada de <http://waine.us.es/dokuwiki/doku.php?id=waine:package:upercon>

9 Glosario

WAINE: Web Application Interface Engine.

MBUIDE: Model Based User Interface Development Environment.

ASL: Application Specification Language.

WINTER: Waine Integrator.

RC: Repositorio de Configuraciones.

ITSM: Information Technology Service Management.

ITAM: Information Technology Asset Management

BBDD: Base de datos.

GTI: Gestión TIC.

DER: Diagrama Entidad-Relación.

WBS: Work Breakdown Language.

IU: Interfaz de Usuario.

ISO: International Organization for Standardization.

WINTER: Waine Integrator.