

Proyecto Fin de Grado

Grado en Ingeniería de las Tecnologías de
Telecomunicación

Plataforma Web para la creación de rutinas de
entrenamiento para la App RutinaApp

Autor: Francisco José Díaz Romero

Tutor: María Teresa Ariza Gómez

Departamento de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2017



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Plataforma Web para la creación de rutinas de entrenamiento para la App RutinaApp

Autor:

Francisco José Díaz Romero

Tutor:

María Teresa Ariza Gómez

Profesor titular

Departamento de Ingeniería Telemática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017

Trabajo Fin de Grado: Plataforma Web para la creación de rutinas de entrenamiento
para la App RutinaApp

Autor: Francisco José Díaz Romero

Tutor: María Teresa Ariza Gómez

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal

A mi familia

A mis maestros

A mis amigos

Agradecimientos

Son muchísimas las personas que me han acompañado a lo largo de esta etapa, en cada uno de los momentos vividos, tanto buenos como malos, y es por eso por lo que quiero agradecerles y dedicarles cada grano de esfuerzo que he realizado para llegar a conseguir mis objetivos, porque sin ellos no hubiese sido posible.

En primer lugar, agradecer a mis padres, Francisco Jesús y Carmen por toda una vida llena de sacrificio y trabajo para que mi hermana y yo tuviésemos la oportunidad de formarnos, invirtiendo en nuestra educación y formación. También agradecer a mi tía Conso por todo el apoyo recibido, la cual se ha comportado como una segunda madre en todo momento para mí y mi hermana. Nunca os ha faltado implicación hacia nosotros, cuidado, atención y cariño, lo cual puedo y podré sentirme orgulloso de vosotros toda mi vida. Gracias a mi hermana, Carmen, la pequeña de la casa, que para ella habré sido un ejemplo a seguir, pero ella para mí también lo ha sido, y le agradezco mucho el apoyo recibido por ella en todo momento.

En segundo lugar, agradecer a mis abuelos todo el apoyo que también me han aportado, y en general a la familia por todo lo recibido por ellos y todas las experiencias vividas, las cuales sirven para aprender y las cuales me han llevado a ser como soy. En especial, nombrar a mis primos Jesús y Silvia, los cuales han estado ahí en todo momento.

Agradecer a mis compañeros incondicionales de batalla y de convivencia, con los que he compartido momentos, vivencias, experiencias y convivencias, Alejandro, Jesús, Lorenzo y Daniel. Me llevo de ellos en esta etapa a grandes personas, grandes amigos y únicas personas las cuales me siento afortunado por haberlas conocido. También quiero agradecer aquí a Mario y Manuel, mis mentores, mis amigos incondicionales junto con Alejandro, los cuales me han enseñado muchísimas cosas y a los cuales les debo gran parte de mi trayectoria como persona y como profesional.

Mi más sincero y profundo agradecimiento también para mi pareja, Patricia, la cual ha sido una persona que a pesar conocerme poco a día de hoy, me ha apoyado en todo momento desde el primer día que la conocí, y que a día de hoy es una persona muy especial para mí, y que cada día que pasa me siento más afortunado de haberte conocido.

Gracias a todas aquellas personas que han aparecido en mi vida en el final de mi etapa universitaria y todos los que decidieron algún día marcharse de mi vida, por todo el apoyo y asesoramiento recibido. En especial a mis compañeros de trabajo en Everis, Andrés, Jesús Rodríguez, Alejandro e Iván.

Gracias a mis amigos de infancia y de toda la vida y compañeros de universidad y residencia de estudiantes, Manuel Ruíz, Manuel Jesús Campos, José Israel, Rafa, Pablo y Antonio Ángel, por todas las anécdotas vividas, momentos disfrutados, viajes y experiencias vividas.

Gracias también a mi tutora, María Teresa, por brindarme esta oportunidad de investigar sobre este campo de las tecnologías de la información por la que tengo gran vocación e interés profesional. Gracias por su implicación en todo momento, su asesoramiento y corrección.

Por último, en especial, quiero dedicarle este proyecto y título de Ingeniería a una persona muy especial para mí, Mi Abuela Laura. Un día te fuiste sin poder despedirme de tí porque estaba de exámenes y no pude llegar a tiempo para poder verte por una última vez, pero quiero que sepas que estás donde estás esto va por tí, porque para mí has sido como mi madre y quiero que te sientas orgullosa de ello estás donde estás.

Te quiero Abuela.

Francisco José Díaz Romero.

Sevilla, 2017

Resumen

La revolución tecnológica que vivimos hoy día, enfocada principalmente en el mundo de Internet, y con este las aplicaciones Web, está permitiendo que todas las herramientas que tenemos a nuestro alrededor se encuentren en un proceso de evolución y cambio continuo.

En este proyecto se ha desarrollado una aplicación Web que permite gestionar rutinas de entrenamiento, cuya finalidad es la creación de dichas rutinas y su personalización por el usuario que las crea, de la forma que crea oportuna, para su posterior uso y reproducción en una aplicación Móvil desarrollada en Android, llamada Rutina App.

Con esto, podemos proporcionar al usuario, una herramienta con la que poder planificar sus entrenamientos de forma interactiva y sencilla, favoreciéndole a llevar un estilo de vida saludable. También destacar que con estos objetivos conseguimos que el usuario disponga en todo momento de la información relativa a las rutinas, y destacar que también tiene la posibilidad de compartir sus rutinas de entrenamiento y ejercicios de forma que se pueda disponer de un repositorio de información publicada por todos los usuarios y disponibles para todos ellos.

Agradecimientos	9
Resumen	11
Índice	13
Índice de Figuras	17
1 Introducción	22
1.1 <i>Motivación y Objetivos</i>	22
1.2 <i>Contexto</i>	23
1.3 <i>Presentación Del Problema</i>	24
1.4 <i>Antecedentes</i>	25
1.5 <i>Descripción de la Solución</i>	28
1.5.1 <i>Objetivos específicos</i>	28
1.5.2 <i>Funcionalidades</i>	28
1.5.3 <i>Arquitectura</i>	29
1.6 <i>Estructura de la memoria</i>	30
2 Tecnologías utilizadas	32
2.1 <i>Servicio Web REST</i>	32
2.1.1 <i>Arquitectura REST</i>	32
2.1.2 <i>Uso en nuestro proyecto (REST) y ventajas respecto a otras arquitecturas</i>	33
2.2 <i>Formato para el intercambio de datos (JSON)</i>	34
2.2.1 <i>Introducción</i>	34
2.2.2 <i>Otros formatos (XML)</i>	34
2.2.3 <i>Uso en el proyecto</i>	34
2.3 <i>Tecnologías en el Servidor: Java Spring Framework</i>	35
2.3.1 <i>Introducción: Spring</i>	35
2.3.2 <i>Spring Boot</i>	36
2.3.3 <i>Spring Security</i>	36
2.3.4 <i>Spring Data</i>	37
2.3.5 <i>Spring Beans</i>	38
2.3.6 <i>Construcción y Gestión de Software (Maven)</i>	40
2.3.7 <i>Inversión de Control (IoC)</i>	41
2.3.8 <i>Inyección de dependencias (DI)</i>	41
2.4 <i>Persistencia: Base de datos</i>	42
2.4.1 <i>Introducción: Modelo Relacional</i>	42
2.4.2 <i>Gestor MySQL y uso en nuestro proyecto</i>	42
2.5 <i>Tecnologías para la interfaz web de usuario</i>	43
2.5.1 <i>Patrón Modelo-Vista-Controlador (MVC)</i>	43
2.5.2 <i>Bootstrap (HTML5 + CSS3 + JavaScript)</i>	44
2.5.3 <i>Controlador de cliente: JQuery/Ajax</i>	47

3	Herramientas Utilizadas	51
3.1	<i>Spring Tool Suite (STS)</i>	51
3.2	<i>MySQLWorkbench</i>	53
3.3	<i>XAMPP</i>	54
3.4	<i>Advanced REST Client</i>	55
3.5	<i>Google Chrome</i>	56
3.6	<i>EditThisCookie</i>	57
3.7	<i>Git / GitHub</i>	58
3.8	<i>Waffle.io</i>	60
3.8.1	Uso en nuestro proyecto (Metodología Scrum)	61
4	Arquitectura y análisis	63
4.1	<i>Diagramas de Clases</i>	64
4.1.1	Gestión de Usuarios	65
4.1.2	Gestión de Rutinas	66
4.1.3	Gestión de Ejercicios	67
4.1.4	Gestión de Vídeos	68
4.1.5	Gestión de la Seguridad (Spring Security)	69
4.2	<i>Diagramas de Casos de Uso</i>	70
4.2.1	Identificación de los actores del sistema	70
4.2.2	Usuario no Autenticado	71
4.2.3	Usuario Autenticado	72
4.2.4	Gestión de Rutinas	74
4.2.5	Gestión de Ejercicios	78
4.2.6	Ajustes	82
4.3	<i>Diagramas de Secuencia</i>	83
4.3.1	Identificación de elementos en los diagramas	83
4.3.2	Inicio de Sesión	84
4.3.3	Obtención de datos	85
4.3.4	Inserción de datos	86
4.3.5	Modificación de Datos	87
4.3.6	Eliminación de Datos	87
4.3.7	Descarga de Rutina	88
4.4	<i>Diagrama EERR de la BBDD</i>	89
4.4.1	Detalle de los elementos del diagrama EERR	90
5	Interfaz de usuario y funcionalidad	95
5.1	<i>Introducción</i>	96
5.2	<i>Autenticación o Inicio de Sesión</i>	96
5.3	<i>Creación de una cuenta de usuario</i>	99
5.4	<i>Gestión de Rutinas</i>	101
5.4.1	Crear Nueva Rutina	101
5.4.2	Rutinas Disponibles	103
5.5	<i>Gestión de Ejercicios</i>	107
5.5.1	Crear Nuevo Ejercicio	107
5.5.2	Ejercicios Disponibles	109
5.6	<i>Ajustes</i>	112
5.6.1	Modificar Datos de Usuario	112
5.6.2	Dar de Baja al Usuario	113
5.7	<i>Ayuda y Contacto (Envío de Email)</i>	114
5.8	<i>Cerrar sesión de usuario</i>	115
6	Líneas de Mejora y Conclusiones	117
6.1	<i>Líneas de Continuación del Proyecto</i>	118

6.1.1	API entre Aplicación Android y Servicio Web	118
6.1.2	Mejoras de Funcionalidad de la Aplicación Web	118
6.2	<i>Líneas de Mejora</i>	120
6.2.1	JQuery VS AngularJS	120
6.2.2	JDBC VS JPA, Hibernate	120
6.2.3	Compresión y cifrado del Código JS	121
6.2.4	Spring Security	122
6.2.5	OAuth2 (Autenticación)	122
6.2.6	Documentación del Servicio Web REST: Swagger	123
6.3	<i>Conclusiones finales</i>	124
Referencias		126
Anexo A: API REST Documentation		129
A.1	<i>Recurso Usuario</i>	129
A.2	<i>Recurso Rutina</i>	130
A.3	<i>Recurso Ejercicio</i>	131
A.4	<i>Recurso Asociaciones Rutinas/Ejercicios</i>	132
A.5	<i>Recurso Vídeo</i>	133
Anexo B: Manual de Instalacion y Despliegue de la aplicacion		135
B.1	<i>Instalación de Java JDK 8</i>	135
B.2	<i>Instalación de Spring Tools Suite (STS)</i>	136
B.3	<i>Instalación de XAMPP</i>	137
B.4	<i>Despliegue de la aplicación</i>	139
Anexo C: Diagrama de Gantt		145

ÍNDICE DE FIGURAS

Figura 1.1: Fuente de datos, Morgan Stanley.	23
Figura 1.2: Aplicación Móvil Rutina App	25
Figura 1.3: Aplicación Web de Runtastic.	26
Figura 1.4: Aplicación Móvil de Runtastic.	26
Figura 1.5: Aplicación Web de Strava.	27
Figura 1.6: Aplicación Móvil de Strava.	27
Figura 1.7: Arquitectura de la aplicación Web para RutinaApp.	29
Figura 2.1: Logo HTTP REST.	32
Figura 2.2: Logo de JSON.	34
Figura 2.3: Logo de XML	34
Figura 2.4: Logo de Spring Framework	35
Figura 2.5: Módulos de Spring Usados en este proyecto.	35
Figura 2.6: Logo de Spring Boot.	36
Figura 2.7: Logo de Spring Security.	36
Figura 2.8: Logo de Spring Data	37
Figura 2.9: Logo de JDBC	38
Figura 2.10: Logo de Java Beans.	38
Figura 2.11: Logo de Maven.	40
Figura 2.12: Formato de fichero POM.xml.	40
Figura 2.13: Logo de MySQL.	42
Figura 2.14: Esquema del Modelo MVC.	43
Figura 2.15: Logo de Bootstrap.	44
Figura 2.16: Logo de HTML5.	45
Figura 2.17: Logo de CSS3.	45
Figura 2.18: Logo de JavaScript.	46
Figura 2.19: Logo de DOM.	46
Figura 2.20: Logo de JQuery.	47
Figura 2.21: Logo de AJAX.	47
Figura 2.22: Esquema de tecnologías aplicadas en AJAX.	48
Figura 2.23: Esquema de aplicación Web tradicional y aplicación Web con AJAX.	48
Figura 2.24: Logo de JQuery AJAX.	49
Figura 2.25: Ejemplo de sintaxis de una consulta simple en AJAX.	49
Figura 2.26: Logo de JQuery Plugin Validation.	49
Figura 3.1: Logo de Spring Tool Suite.	51

Figura 3.2: Interfaz Gráfica de STS.	52
Figura 3.3: Logo de MySQL WorkBench.	53
Figura 3.4: Logo de XAMPP.	54
Figura 3.5: Logo de phpMyAdmin.	54
Figura 3.6: Logo de Advanced REST Client.	55
Figura 3.7: Interfaz Gráfica de Advanced REST Client.	55
Figura 3.8: Logo de Google Chrome.	56
Figura 3.9: Interfaz Gráfica del depurador en modo consola de Google Chrome.	56
Figura 3.10: Logo de EditThisCookie.	57
Figura 3.11: Esquema de la extensión de EditThisCookie en Google Chrome.	57
Figura 3.12: Interfaz Gráfica de EditThisCookie.	57
Figura 3.13: Logo de Git.	58
Figura 3.14: Logo de GitHub.	58
Figura 3.15: Interfaz Gráfica de GitHub.	59
Figura 3.16: Logo de Waffle.io.	60
Figura 3.17: Interfaz Gráfica de Waffle.io.	60
Figura 3.18: Esquema Gráfico de Metodología Scrum.	61
Figura 4.1: Diagrama de clases de la Gestión de Usuarios.	65
Figura 4.2: Diagrama de Clases de la Gestión de Rutinas	66
Figura 4.3: Diagrama de Clases de la Gestión de Ejercicios.	67
Figura 4.4: Diagrama de Clases de Gestión de Vídeos.	68
Figura 4.5: Diagrama de Clases de la Gestión de Seguridad con Spring Security.	69
Figura 4.6: Diagrama de Casos de Uso de Acciones para Usuario No Autenticado.	71
Figura 4.7: Diagrama de Casos de Uso con Operaciones Disponibles para Usuario Autenticado.	72
Figura 4.8: Diagrama de Casos de Uso de Gestión de Rutinas de UA.	74
Figura 4.9: Diagrama de Casos de Uso de la Gestión de las Rutinas propias de un UA.	75
Figura 4.10: Diagrama de Casos de Uso de la Gestión de las Rutinas públicas de otros usuarios para un UA.	77
Figura 4.11: Diagrama de Casos de Uso de Gestión de Ejercicios.	78
Figura 4.12: Diagrama de Casos de Uso de la Gestión de los Ejercicios propios de un UA.	79
Figura 4.13: Diagrama de Casos de Uso de la Gestión de los Ejercicios públicos de otros usuarios para un UA.	81
Figura 4.14: Diagrama de Casos de Uso de Ajustes de la cuenta de UA.	82
Figura 4.15: Diagrama de Secuencia del Inicio de Sesión.	84
Figura 4.16: Diagrama de Secuencia de la obtención de datos.	85
Figura 4.17: Diagrama de Secuencia de la inserción de datos	86
Figura 4.18: Diagrama de Secuencia de la eliminación de datos.	87
Figura 4.19: Diagrama de Secuencia de la Descarga de una Rutina.	88

Figura 4.20: Diagrama EERR de la Base de Datos, para la Gestión de Rutinas de Entrenamiento.	89
Figura 5.1: Página de Inicio de Sesión.	96
Figura 5.2: Inicio de Sesión de Usuario ya Registrado.	97
Figura 5.3: Mensaje de Aviso de Credenciales Introducidas Erróneas.	97
Figura 5.4: Página de Inicio de la aplicación Web tras la Autenticación correcta del Usuario.	98
Figura 5.5: Menú desplegable de Navegación de Usuario.	98
Figura 5.6: Página de Registro Gratuito de Usuario.	99
Figura 5.7: Registro Correcto de Nuevo Usuario.	99
Figura 5.8: Registro erróneo de nuevo Usuario.	100
Figura 5.9: Opciones para la Gestión de Rutinas en el menú desplegable.	101
Figura 5.10: Formulario de Registro de Nueva Rutina.	101
Figura 5.11: Mensaje de aviso de Registro de Rutina Correcto.	102
Figura 5.12: Lista de Rutinas Disponibles de un Usuario Autenticado.	103
Figura 5.13: Menú desplegable para selección de Rutinas a Visualizar.	103
Figura 5.14: Lista de Rutinas Públicas de otros Usuarios Disponibles para un Usuario Autenticado.	103
Figura 5.15: Botón correspondiente a la Modificación de una Rutina.	104
Figura 5.16: Formulario para Modificación de Rutina.	104
Figura 5.17: Lista de Ejercicios Asociados a una Rutina.	104
Figura 5.18: Lista de Ejercicios Disponibles para Asociar a una Rutina.	105
Figura 5.19: Mensajes de Aviso para Asociación de Ejercicios a una Rutina.	105
Figura 5.20: Botón correspondiente a la descarga de una Rutina.	105
Figura 5.21: Mensajes de Aviso para la Descarga de una Rutina.	105
Figura 5.22: Mensajes de Aviso para la Eliminación de una Rutina.	106
Figura 5.23: Opciones para la Gestión de Rutinas en el menú desplegable.	107
Figura 5.24: Formulario de Registro de Nuevo Ejercicio.	107
Figura 5.25: Mensaje de aviso de Registro de Ejercicio Correcto.	108
Figura 5.26: Lista de Ejercicios Disponibles de un Usuario Autenticado.	109
Figura 5.27: Menú desplegable para selección de Ejercicios a Visualizar.	109
Figura 5.28: Lista de Ejercicios Públicos de otros Usuarios Disponibles para un Usuario Autenticado.	109
Figura 5.29: Botón correspondiente a la Modificación de un Ejercicio.	110
Figura 5.30: Formulario para Modificación de Ejercicio.	110
Figura 5.31: Botón correspondiente a la Subida de un Vídeo para un Ejercicio.	110
Figura 5.32: Formulario para la Subida de un Vídeo y Asignación a un Ejercicio.	110
Figura 5.33: Botón correspondiente a la Visualización de un Vídeo Asociado a un Ejercicio.	111
Figura 5.34: Vista Correspondiente a la Visualización de un Vídeo Asociado a un Ejercicio.	111
Figura 5.35: Botón de Opciones de Eliminar Vídeo/Ejercicio.	111
Figura 5.36: Mensajes de Aviso de Borrado de Vídeo/Ejercicio con Éxito.	111
Figura 5.37: Opciones de Ajustes en el menú desplegable	112
Figura 5.38: Formulario para la Modificación de Datos de Perfil de Usuario.	112

Figura 5.39: Mensaje de aviso de Modificación Correcta de Datos de Perfil de Usuario.	113
Figura 5.40: Formulario para Dar de Baja al Usuario.	113
Figura 5.41: Mensaje de confirmación de Usuario dado de Baja.	113
Figura 5.42: Formulario de Ayuda al Usuario.	114
Figura 5.43: Ventana del Cliente de Correo del Usuario.	114
Figura 5.44: Mensaje de Aviso en el caso de que el Usuario Cierre su Sesión.	115
Figura 6.1: Logo de AngularJS.	120
Figura 6.2: Logo de JPA, Hibernate.	120
Figura 6.3: Logo de OAuth.	122
Figura 6.4: logo de Swagger.	123
Figura B.1: Descarga de Java JDK 8.	135
Figura B.2: Versión de Java Instalado.	135
Figura B.3: Pagina de descarga de STS.	136
Figura B.4: Ejecutable de STS.	136
Figura B.5: Inicio de STS.	136
Figura B.6: Pagina Web de XAMPP.	137
Figura B.7: Proceso de Instalación de XAMPP.	137
Figura B.8: Navegador Web de PHPMyAdmin para la Gestión de BBDD.	138
Figura B.9: Gestor de BBDD en consola.	138
Figura B. 10: Interfaz para Importar proyecto.	139
Figura B. 11: Directorio Raíz del Proyecto.	139
Figura B.12: Proyecto importado al entorno STS.	140
Figura B. 13: Creación de la BBDD.	140
Figura B. 14: Modificaciones del pom.xml.	141
Figura B. 15: Clase Principal de la Aplicación.	141
Figura B.16: Directorio del proyecto.	141
Figura B.17: Crear nueva instancia de Pivotal tc Server.	142
Figura B.18: Directorio donde escucha la instancia del servidor.	143
Figura B.19: Servicio Web Desplegado y publicado en el Servidor.	143
Figura B.20: Botón de “Play” para iniciar la aplicación.	143
Figura B.21: Proceso de inicialización del Servicio Web.	143

1 INTRODUCCIÓN

Hemos establecido la civilización de manera que los más cruciales elementos dependen de la ciencia y la tecnología.

Carl Sagan.

Actualmente, el desarrollo software se encuentra en un proceso revolucionario cuyo principal objetivo y tendencia es la implantación y desarrollo de software distribuido, donde se marcan como principales objetivos, la disponibilidad, usabilidad y funcionalidad de cualquier aplicación a través de internet. Es aquí donde entran en juego, entre otros, el desarrollo de aplicaciones o servicios Web, sustituyendo en gran medida al desarrollo de aplicaciones y servicios de escritorio.

Muchos son los aspectos a tener en cuenta, entre los que cabe destacar la eficiencia, la velocidad de respuesta y de navegación, los cuales para algunos servicios Web son excelentes, pero para otros aún se encuentran en un proceso de mejora. Esto es debido a que gran cantidad de aplicaciones requieren de un buen diseño donde se optimicen el uso de los recursos disponibles, como el ancho de banda, memoria, carga del servidor, respaldo, seguridad, etc.

1.1 Motivación y Objetivos

El proyecto que nos ocupa, pretende desarrollar una aplicación Web para gestión y creación de rutinas de entrenamiento, proporcionando las funcionalidades necesarias para ello, las cuales serán usadas por una aplicación móvil en Android.

Dicha aplicación Web estará compuesta por un conjunto de servicios Web que proporcionarán las funcionalidades necesarias para tal fin, cuyo desarrollo web se implementará con el uso de un framework llamado Spring, que permitirá el despliegue, tanto en máquinas locales como en la Cloud, de nuestra aplicación web desde un solo fichero, que detallaremos más adelante.

Atendiendo a la mención anterior, dicha aplicación seguirá la arquitectura de un servicio Web REST desarrollado con Spring framework, usando como lenguaje de programación Java. Durante el desarrollo de este servicio Web, se han hecho uso de ciertas librerías nativas de Java, además del manejo de tipos de datos simples, aprovechando al máximo las capacidades que nos proporciona Java y su framework de Spring.

Finalmente, son muchas las razones por las cuales he decidido adentrarme en el mundo de las aplicaciones y servicios Web, ya que a lo largo del grado, ha sido lo que más curiosidad e interés me ha generado, viendo en este proyecto una oportunidad para aprender e investigar en tecnologías orientadas a los servicios Web. Otra de las razones es mi pasión y dedicación hacia el deporte en mi vida, lo cual ha hecho que este interés sea mayor debido a la relación directa que guarda este proyecto con el mismo.

Una vez mostradas las motivaciones y objetivos que han llevado a realizar este proyecto, detallaremos a continuación el problema a resolver y la solución que se propone, además de los antecedentes relacionados con dicha tarea.

1.2 Contexto

Durante los últimos años, la evolución tecnológica gira principalmente en torno a Internet, hasta tal punto de que dicha herramienta está presente cada vez más en nuestras vidas y en todas las actividades que desempeñamos en el día a día.

Poco a poco, la necesidad de disponer de toda la diversidad funcional y tecnológica que esta herramienta nos proporciona, ligado a la necesidad de comunicarnos y estar conectados continuamente, evoluciona en paralelo, entre otras, la tecnología móvil. Es aquí donde ocupa la mayor parte del foco de atención, los dispositivos móviles, principalmente los Smartphones.

Ambas herramientas en conjunto nos permiten el desarrollo de muchas de nuestras actividades desde cualquier lugar, ya que a día de hoy muchas de estas actividades están ligadas a la tecnología y principalmente a Internet, ya sean actividades de ocio, de trabajo, de docencia, etc.

Atendiendo a estas necesidades y expectativas de la sociedad, nace el concepto de aplicación Web o servicio Web, donde se permite la interacción con servicios o herramientas tecnológicas ligadas a nuestras actividades cotidianas desde cualquier lugar y con una disponibilidad de recursos continua y eficiente.

Observando la tendencia y los intereses sociales en la actualidad, podemos observar que la imagen juega un papel primordial en muchos aspectos sociales, lo cual nos lleva a tender hacia un hábito de vida e imagen física saludables, donde como medio para alcanzar dicho fin aparecen la alimentación y el deporte.

Por otro lado, el ritmo de vida de la sociedad actual, cada vez es más acelerado y exigente, y la precariedad económica que se ha sufrido durante los últimos años a nivel mundial nos impide a veces poder acudir a centros de entrenamiento y de nutrición especializados. Estos obstáculos nos llevan a buscar alternativas.

Aquí nacen entre muchas otras, las aplicaciones móviles o plataformas web para el seguimiento de nuestro entrenamiento físico, las cuales permiten personalizar nuestros entrenamientos y poderlos realizar en casa sin necesidad de acudir al gimnasio.

Además, estas aplicaciones nos enseñan a seguir un hábito de entrenamiento saludable, ya que la correcta ejecución de los ejercicios físicos es muy importante para evitar lesiones futuras. También aportan funcionalidades de monitorización y seguimiento del ejercicio físico, de forma que para conseguir un mínimo estado de forma no es necesaria la presencia de un entrenador o monitor.

Global Mobile vs. Desktop Internet User Projection, 2007 - 2015

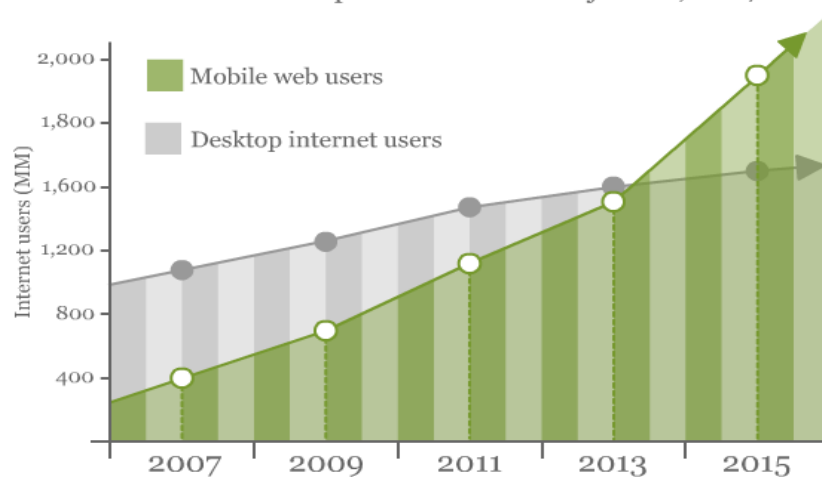


Figura 1.1: Fuente de datos, Morgan Stanley.

1.3 Presentación Del Problema

Como principal problema que se plantea en este proyecto es proponer una solución Web a una aplicación móvil desarrollada en Android para la gestión y creación de rutinas de entrenamiento.

La gran mayoría de aplicaciones existentes en el mercado, dedicadas al seguimiento de tareas físicas, imponen ciertas limitaciones en el número de rutinas de entrenamiento y de ejercicios, además de que son aplicaciones muy genéricas en cuanto al área de entrenamiento.

Es por tanto que se desea mejorar o enfatizar en estas carencias de otras aplicaciones para intentar mejorarlas en nuestra solución.

Persiguiendo lo mencionado previamente, en nuestra plataforma Web, el propio usuario que hace uso de la plataforma es el encargado de crear, elegir y descargar las rutinas que desee. Puede elegir si sólo desea rutinas especializadas en una zona muscular o en varias, el nivel de forma del usuario y otros aspectos descriptivos del ejercicio, como una descripción detallada o vídeos relacionados con el ejercicio físico.

En resumen, la idea de esta página sería que cualquier usuario pueda ser productor y consumidor de rutinas, el límite en el número sólo lo imponen los propios usuarios. Además, en esta plataforma se permitirá al usuario la descarga de las rutinas que desee para poder reproducirlas en la aplicación móvil, en formato .zip.

Con la combinación de la aplicación móvil y la plataforma Web, conseguimos que mediante una aplicación Web podamos crear nuestras propias rutinas y personalizarlas, para posteriormente descargarlas y reproducirlas en la aplicación móvil.

En comparación con otras opciones existentes en el mercado, la mayoría de las aplicaciones móviles requieren de conexión a internet para reproducir las rutinas de entrenamiento desde un servidor externo, por lo que si no se dispone de buena conexión a internet, no se consigue una calidad mínima del servicio.

El objetivo es que la aplicación funcione siempre en tiempo real y en cualquier lugar donde se encuentre el usuario, además de poder crear sus propias rutinas de entrenamiento y que estas estén disponible siempre y desde cualquier lugar para poder ser descargadas a través de la aplicación Web, y reproducidas localmente desde cualquier dispositivo móvil Android.

1.4 Antecedentes

Como antecedentes principales a nuestro proyecto, debemos destacar principalmente la siguiente línea de trabajo:

Aplicación Android de rutinas de entrenamiento adaptadas al usuario usando SQLite y JSON

Este es un proyecto desarrollado en la Escuela Técnica Superior de Ingeniería de Sevilla por la alumna Mirian Franco Maireles en el año 2015.

El objetivo principal de este proyecto es el desarrollo de una aplicación móvil en Android que permita al usuario tener sus propias rutinas de entrenamiento en el lugar y el momento que lo desee, con la mayor calidad y exactitud posible, y con la ausencia de un profesional especializado, tales como un entrenador personal o fisioterapeuta.

Se tienen en cuenta además especificaciones tales como la forma física del usuario, además de una descripción detallada de las rutinas y los ejercicios que agrupan, destacando como principal característica la disponibilidad de vídeos explicativos para los ejercicios de dichas rutinas.

El funcionamiento de dicha aplicación está totalmente desvinculado a la existencia de una red de datos, lo cual no permite la gestión de los recursos que maneja dicha aplicación a través de Internet mediante alguna plataforma o servicio Web, dejándose esta como línea de desarrollo.

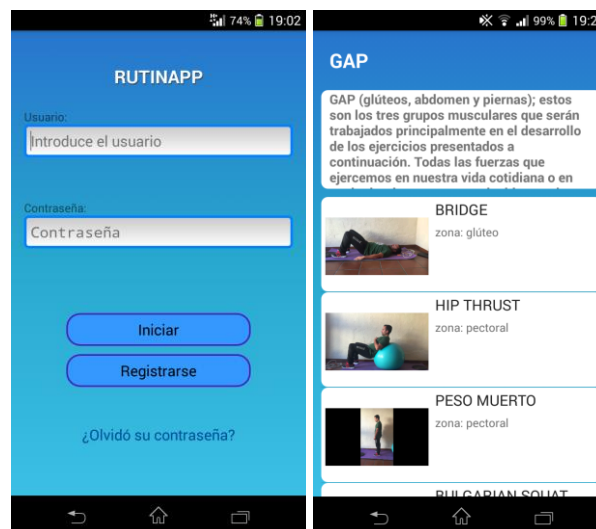


Figura 1.1: Aplicación Móvil Rutina App

Otras Referencias

Son muchas las aplicaciones móviles que disponen de una plataforma web para su gestión remota o para su gestión y almacenamiento de datos en la Cloud, permitiendo tener la información distribuida y disponible siempre, además de poder compartirla con más usuarios de la plataforma si se desea. Existen aplicaciones móviles orientadas al deporte que disponen además de plataformas web, las cuales algunas de estas son:

Runtastic

Es una plataforma dedicada al seguimiento de la actividad física, la cual permite multitud de funcionalidades, como seguimiento de la actividad física, estadísticas, almacenamiento de rutas para actividades como el running o ciclismo, red social y sincronización con Facebook, etc. A día de hoy es una de las más completas existentes en el mercado, y cuenta tanto con aplicación web como aplicación móvil.

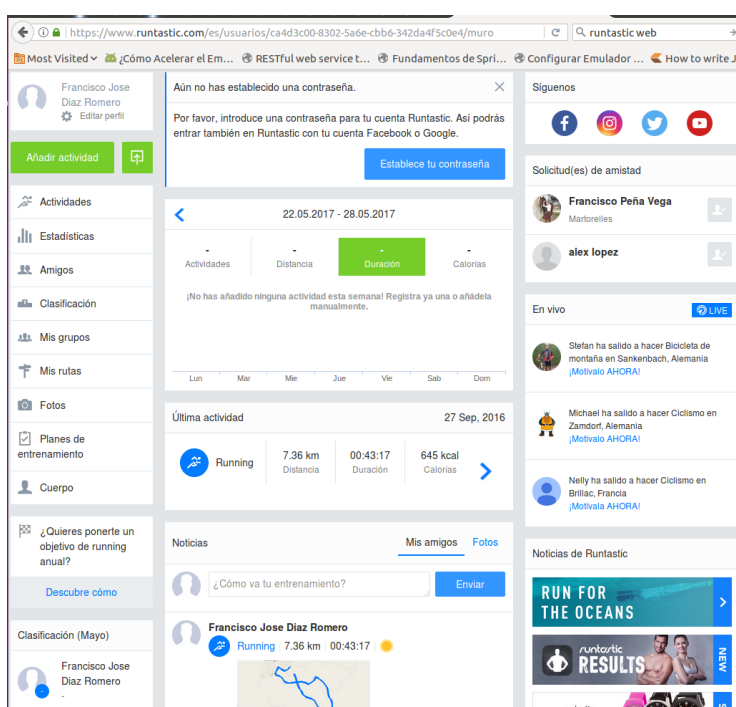


Figura 1.2: Aplicación Web de Runtastic.

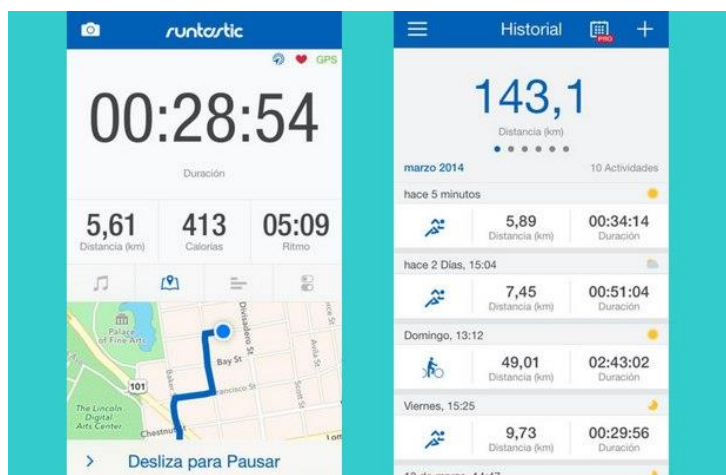


Figura 1.3: Aplicación Móvil de Runtastic.

Strava

Esta se trata de otra plataforma, la cual está dedicada solamente a dos actividades de fondo como el running y el ciclismo, mientras que la anterior permite su uso para muchos otros deportes. Esta aplicación destaca porque en el trazado de rutas en el mapa, reconoce tramos en el mapa transitados por más usuarios, y permite realizar rankings y estadísticas de tiempos.

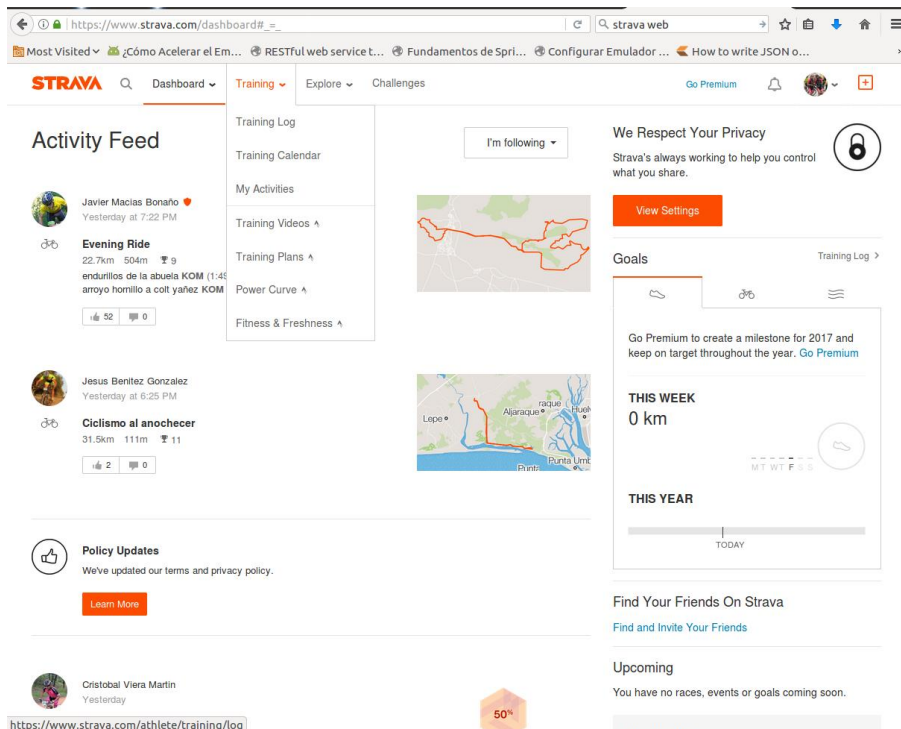


Figura 1.4: Aplicación Web de Strava.

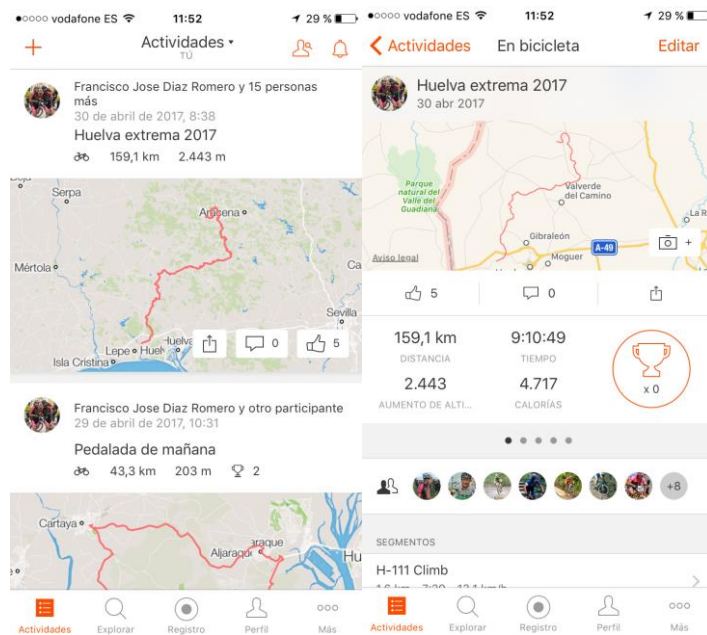


Figura 1.5: Aplicación Móvil de Strava.

1.5 Descripción de la Solución

En esta sección abordaremos las características principales, funcionalidades y arquitectura que seguiremos en el desarrollo del proyecto.

1.5.1 Objetivos específicos

Como objetivo principal se busca facilitar al usuario la creación y gestión de un auto-entrenamiento a través de una plataforma web, y su posterior reproducción y realización desde cualquier lugar y en cualquier momento a través de la aplicación móvil que ya se proporciona.

A través de la aplicación Web, el usuario podrá descargar tanto sus rutinas como las rutinas que el resto de usuarios hayan decidido publicar, de forma que estas se podrán reproducir en la aplicación móvil Rutina App.

1.5.2 Funcionalidades

Al iniciar el navegador del usuario y acceder a la aplicación Web, en la página principal se proporcionan varias alternativas:

- ✓ Autenticación de un usuario (Usuario, Contraseña) que previamente se ha registrado.
- ✓ Acceso a la vista Web donde se permitirá el registro de un nuevo usuario.

Una vez iniciada la sesión con un usuario, se accederá a la vista principal de usuario, en la que se dispondrá de un menú con diversas funcionalidades:

- ✓ Acceso a la modificación de los datos de usuario.
- ✓ Acceso a la creación, modificación y listado de rutinas de entrenamiento. Además, se permitirá la asociación de ejercicios ya existentes a las rutinas propias del usuario. Por otro lado, se permitirá la descarga de las rutinas existentes en formato “.Zip” para su posterior interpretación en la aplicación Móvil Rutina App.
- ✓ Acceso a la creación, modificación y listado de ejercicios. Además, se permitirá añadir vídeos en formato “.MP4” a los ejercicios permitiendo la subida desde la máquina local del usuario. Se permitirá la visualización de los vídeos disponibles a través de la propia plataforma web.
- ✓ Acceso a soporte de ayuda en línea, en la que como opciones se dan la baja del usuario en el sistema y una vía de contacto a través de correo electrónico con los responsables del mantenimiento y soporte de la Web.
- ✓ Cabe destacar, que los usuarios podrán hacer uso de las rutinas y ejercicios que hayan creado, además de darse la opción de poder listar y descargar las rutinas y ejercicios de otros usuarios que hayan decidido hacer públicos. Además, se da la opción de filtrar la búsqueda por la coincidencia de la búsqueda en algunos de los principales campos de rutinas y ejercicios.
- ✓ Opción para el cierre de sesión por parte del usuario.

1.5.3 Arquitectura

Como la mayoría de servicios Web REST, para el diseño y desarrollo de nuestro servicio hemos seguido una arquitectura basada en el modelo vista controlador (MVC), el cual separa los datos de la aplicación de la lógica de negocio que implementa y de la interfaz de usuario para acceder al servicio.

Esto permite el desarrollo por separado de cada parte, incrementando entre otras, su reutilización y flexibilidad.

Aplicando este patrón de diseño a nuestra solución, podemos visualizar el diseño de nuestro servicio en la siguiente ilustración, donde a continuación se detallará cada parte con las tecnologías y modelos de datos utilizados para ello.

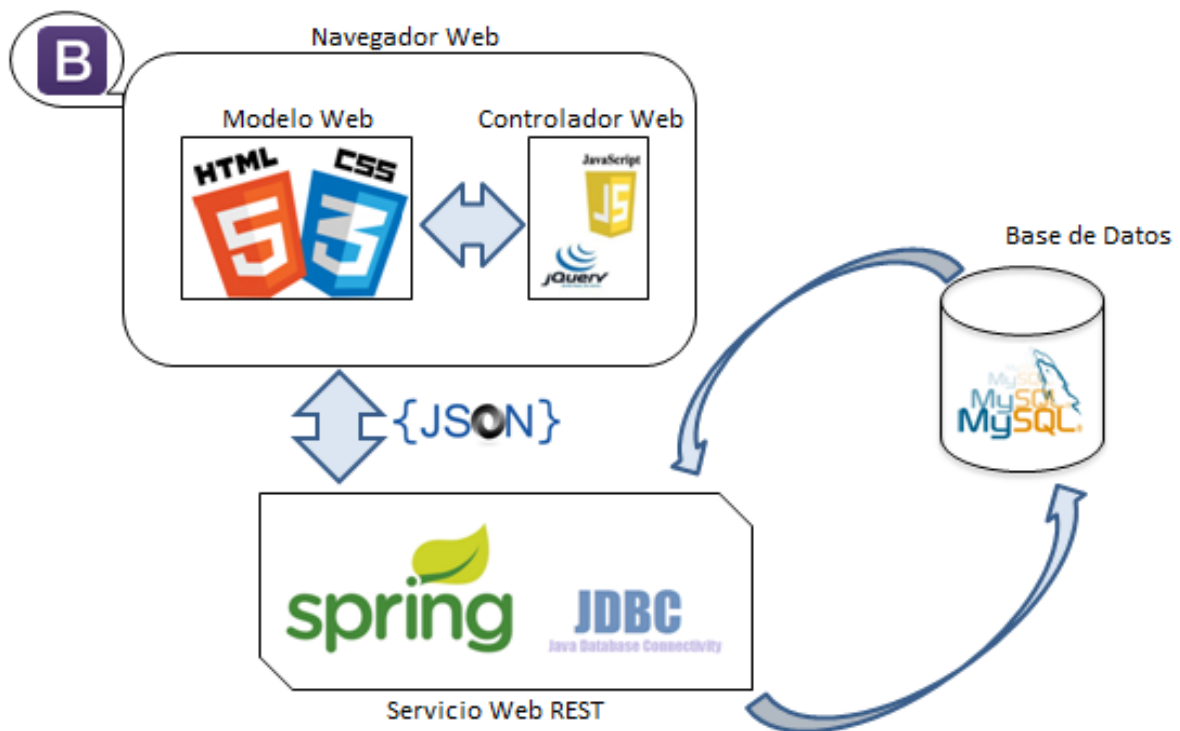


Figura 1.6: Arquitectura de la aplicación Web para RutinaApp.

El usuario interactúa con la aplicación Web a través del Navegador, el cual obtiene datos del servidor mediante el servicio Web y los muestra en la vista, que se genera modificando el modelo Web. Este modelo se implementa con tecnologías de desarrollo y diseño Web (HTML5 y CSS3), con los controladores correspondientes para obtener y enviar datos con servicio Web REST compuestos por ficheros JavaScript. Se hará por tanto uso en su conjunto de Bootstrap, el cual es un framework que proporciona este conjunto de tecnologías.

Se implementará un servicio Web tipo REST, el cual se desarrolla mediante Java Spring framework. La función principal de este servicio es la de gestionar la autenticación de usuarios, seguridad de los datos y la realización de operaciones CRUD (Create, Read, Update, Delete), sobre la información de usuarios, rutinas, ejercicios y vídeos. Como formato para intercambiar datos entre el servicio Web y el controlador del cliente se hará uso de JSON, ya que su estructura es estándar y permite una buena integración entre ambas partes.

Este servicio implementará la persistencia de los datos mediante una base de datos relacional. Esta se ha diseñado siguiendo un modelo Entidad-Relación, usando como gestor de BBDD MySQL. Cabe destacar que, el servicio Web REST accederá a los datos mediante la implantación del patrón DAO (Data Access Model), que detallaremos más adelante.

1.6 Estructura de la memoria

El siguiente esquema resume de forma genérica los puntos tratados en esta memoria para que el lector pueda tener una idea orientativa de la organización de este documento:

1. **Introducción:** En este punto se abordan temas genéricos como los objetivos, ámbito y contexto, antecedentes o funcionalidades, de forma que pueda adquirirse una breve aproximación a los detalles del proyecto.
2. **Tecnologías Utilizadas:** A lo largo de esta sección se entrará en detalle sobre todas las tecnologías utilizadas e implementadas para la realización de este proyecto, con un breve razonamiento de por qué se usan.
3. **Herramientas Utilizadas:** Consta de un resumen de las herramientas que se han usado para implementar y llevar a cabo las tecnologías usadas, y funcionalidades de nuestra aplicación Web.
4. **Arquitectura y Análisis:** Se realiza un análisis de diseño de la aplicación Web, mediante el uso de diagramas UML de diversos tipos.
5. **Interfaz de usuario y Funcionalidad:** Consta de un detalle sobre la funcionalidad visual de la aplicación, de forma que el usuario pueda entender como interactuar con el servicio.
6. **Líneas de Mejora y Conclusiones:** Se mencionarán algunas tecnologías para mejorar este servicio y un razonamiento de las mismas. Además, se abarcan mejoras en cuanto a la propia funcionalidad de la aplicación que podrían ser implementadas y las ventajas de las mismas. También se incluye una conclusión final tras la realización del proyecto y una valoración del mismo a nivel general y personal.

Además, se realizará una sección de anexos, en el que se tratarán puntos adicionales sobre el proyecto:

- **Anexo A:** API REST Documentation.
- **Anexo B:** Manual de Instalación y Despliegue de la Aplicación.
- **Anexo C:** Diagrama de Gantt.

2 TECNOLOGÍAS UTILIZADAS

La tecnología no es en sí el fin sino el medio entre la sociedad del conocimiento y el desarrollo mundial.
Anónimo

En este punto, mostraremos en detalle el conjunto de tecnologías usadas para llevar a cabo la realización de la aplicación Web. Realizaremos una explicación de cada tecnología y un razonamiento de la finalidad con la que se ha usado. Será en el próximo capítulo donde se entrará en detalle sobre la forma en la que se han implementado dichas tecnologías en nuestra solución.

2.1 Servicio Web REST

2.1.1 Arquitectura REST

La arquitectura REST o Transferencia de Estado Representacional es un estilo de arquitectura introducido y definido en el año 2000 por Roy Fielding en su tesis doctoral. Fielding es uno de los principales autores del protocolo de transferencia de hipertexto (HTTP).

Es por tanto, que REST usa dicho protocolo de transferencia, lo que le da gran facilidad para poder ser usada prácticamente por cualquier lenguaje de programación y que sea fácil de testear. Además es un requisito de un servicio REST que el cliente y el servidor sean independientes entre sí.

Esta arquitectura hace énfasis en que las interacciones entre los clientes y los servicios se mejoran al tener un número limitado de operaciones (verbos). La flexibilidad se obtiene asignando recursos a sus propios identificadores de recursos universales únicos (URI). Debido a que cada verbo tiene un significado específico (GET, POST, PUT y DELETE), REST evita la ambigüedad:

- ✓ GET: Obtener un recurso.
- ✓ POST: Crear un recurso en el servidor.
- ✓ PUT: Cambiar el estado de un recurso o actualizarlo.
- ✓ DELETE: Eliminar un recurso.



Figura 2.1: Logo HTTP REST.

REST se ha convertido en una de las tecnologías más importantes para las aplicaciones Web. En el futuro es probable que su importancia siga creciendo rápidamente a medida que las empresas buscan proporcionar interfaces abiertas y bien definidas para los servicios de aplicaciones e infraestructura.

2.1.2 Uso en nuestro proyecto (REST) y ventajas respecto a otras arquitecturas

En nuestro proyecto, hemos hecho uso de esta arquitectura como modelo de diseño para el servicio Web a implantar, ya que hoy día es el más usado por los desarrolladores, lo cual se dispone de gran cantidad de documentación en Internet para aprender a usarlo

El uso de REST es preferible a la arquitectura SOAP (Simple Object Access Protocol) que en comparación es más pesado, porque REST no aprovecha tanto ancho de banda, lo que hace que sea un mejor ajuste para su uso a través de Internet. Por tanto, REST nos proporciona soluciones más simples, sacrificando a su vez la potencia de dar soluciones para todo, como es el caso de SOAP.

2.2 Formato para el intercambio de datos (JSON)

2.2.1 Introducción

Cuando intercambiamos datos entre el navegador Web del cliente y el servidor necesitamos gestionar de forma sencilla y eficaz mucha información. JSON es un estándar para el intercambio de datos siguiendo una sintaxis específica y definida de forma que podemos intercambiar información entre diferentes aplicaciones independientemente de la plataforma o lenguaje en el que hayan sido desarrolladas. JSON nació como una alternativa a XML, debido a su gran facilidad de uso con JavaScript.

Se destaca principalmente porque este formato puede ser leído por cualquier lenguaje de programación, lo cual nos proporciona mayor integridad y flexibilidad.

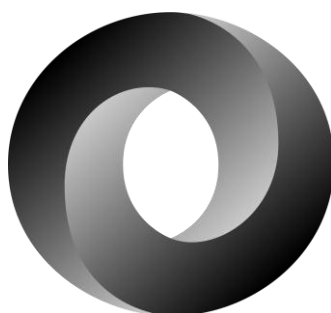


Figura 2.2: Logo de JSON.

2.2.2 Otros formatos (XML)

XML es un lenguaje de marcado que define un conjunto de reglas para codificar información de forma que sea legible por un ser humano o por un ordenador. Gracias a la flexibilidad que proporciona, carece de las limitaciones que tiene HTML y puede ser usada para representar cualquier estructura de datos. Como ventaja principal de este lenguaje, cabe destacar que proporciona soporte Unicode, lo cual permite escribir la información en cualquier idioma del mundo. También cabe mencionar que su complejidad hace que JSON vaya ganando terreno respecto a XML.



Figura 2.3: Logo de XML

2.2.3 Uso en el proyecto

En nuestro proyecto, principalmente se ha hecho uso de JSON como formato para el intercambio de datos entre el cliente y el servicio web. Esto se debe principalmente a que la generación e interpretación de estructuras JSON a partir de objetos JavaScript es sumamente sencilla en el lado del cliente.

Por otro lado, cabe destacar que Spring framework también soporta XML para la representación de los datos, aunque solo hemos hecho uso de este para las directivas y ficheros de configuración tanto del servicio Web Spring REST como para el despliegue del mismo en el servidor de Tomcat.

2.3 Tecnologías en el Servidor: Java Spring Framework

2.3.1 Introducción: Spring

Este es un framework que se usa para el desarrollo de Servicios y aplicaciones, y como contenedor de inversión de control de código abierto para la plataforma Java.

La primera versión de este framework fue escrita por Rod Johnson en Octubre de 2002, el cual fue lanzado inicialmente bajo licencia de Apache 2.0 en Junio de 2003. En los años posteriores, se han ido lanzando más versiones, hasta llegar a la versión 4.0, la cual se encuentra en desarrollo desde enero de 2013.

Spring fue escrito para la plataforma J2EE de Java, plataforma orientada al desarrollo de aplicaciones web, y ha ido evolucionando rápidamente hasta el día de hoy, donde podemos encontrar diferentes ramas de desarrollo de la mano de SpringSource y todo su equipo de desarrolladores.



Figura 2.4: Logo de Spring Framework

Spring Framework nos provee de un conjunto de conceptos, técnicas y una metodología de programación para el desarrollo de aplicaciones informáticas de alta calidad, pudiendo ser usada en cualquier aplicación desarrollada en Java. Se trata pues de un framework que impulsa una metodología de trabajo ágil y eficiente, dando como resultado la creación de software de elevada calidad.

Además, tiene amplia compatibilidad para la integración con otros frameworks y librerías de uso común para la creación de aplicaciones Web, desde Composite Views como Velocity o Tiles, APIs en capa de persistencia como Hibernate o JDO, y otros muchos recursos como JavaMail, Quartz, etc...

En la actualidad, es la referencia mundial como framework de programación orientado al desarrollo Web, y su éxito se basa en la constante labor de investigación e innovación que lleva a cabo su equipo de desarrollo.

En el desarrollo de nuestro proyecto hemos hecho uso de este framework para la construcción de nuestro servicio Web REST, debido a la agilidad y diversidad de herramientas que provee, lo cual hace que su implementación y uso sea más sencillo de llevar a cabo. De todos los módulos que posee este framework, hemos hecho uso de los siguientes: Spring Web Services, Spring Boot, Spring Security, Spring Data.

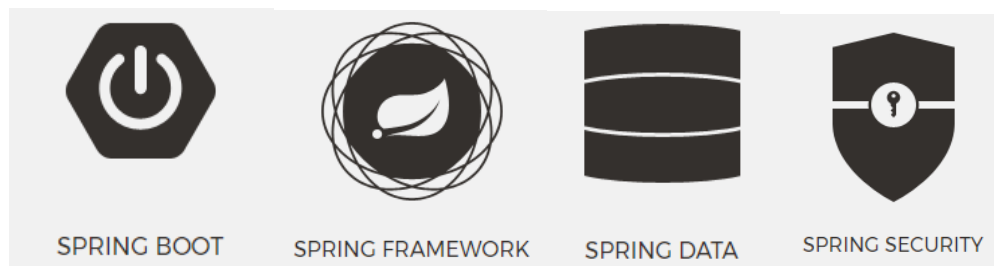


Figura 2.5: Módulos de Spring Usados en este proyecto.

2.3.2 Spring Boot

Spring Boot es un módulo de Spring que nos permite crear diferentes tipos de aplicaciones autónomas de una manera rápida y sencilla, donde simplemente hay que ejecutarlas, simplificando la creación de dependencias y el despliegue en un servidor de aplicaciones.

Principalmente se trata de un mecanismo centrado en una clase principal, la cual tiene una configuración por defecto y que podemos personalizar. Este es compatible con Maven y Gradle, y su formato por defecto para el despliegue es “.JAR”, aunque también soporta “.WAR”.



Figura 2.6: Logo de Spring Boot.

Haremos uso de este módulo para la creación y despliegue de nuestra aplicación Web, configurada mediante una clase principal que se encargará de configurar el resto de parámetros del despliegue de la aplicación de forma automática.

Con esto se consigue que la aplicación completa quede contenida en un fichero “.JAR” o “.WAR” y sólo ejecutando dicho fichero como un fichero normal de clases Java, conseguimos que la aplicación se lance de forma automática, incluyéndose un servidor Web Tomcat embebido.

2.3.3 Spring Security

Spring Security es un módulo que permitirá gestionar todo lo relativo a la seguridad de nuestra aplicación web, desde el protocolo de seguridad, hasta los roles que necesitan los usuarios para acceder a los diferentes recursos de la aplicación. Desde el punto de vista de la autenticación, nos permite comprobar que un usuario exista y permitir el acceso a los recursos de la aplicación, dando autorización a unos recursos u otros según sus credenciales y sus privilegios.

Este módulo consta de una serie de elementos, como clases y parámetros en Java, los cuales hay que personalizar para que se adapte de la mejor manera posible al comportamiento y finalidad que queremos llegar a conseguir en nuestra aplicación Web.



Figura 2.7: Logo de Spring Security.

2.3.4 Spring Data

Este módulo tiene como propósito principal el de unificar y facilitar el acceso a distintos tipos de tecnologías de persistencia, tanto a bases de datos relacionales como no relacionales.

Por defecto, Spring Framework ya proporcionaba soporte para el acceso a bases de datos como JDBC, Hibernate o JPA, simplificando la implementación de la capa de acceso a datos, unificando la configuración y creando una jerarquía de excepciones común para todas ellas.

Spring Data permite cubrir el soporte necesario para todas las tecnologías de persistencia y además, integra las tecnologías de acceso tradicionales, simplificando el trabajo a la hora de crear las implementaciones concretas.

Con cada tipo de tecnología de persistencia, los DAO (Data Access Object) ofrecen funcionalidades típicas de un CRUD (Create, Read, Update y Delete) para objetos de dominio propios, métodos de búsqueda, ordenación y paginación. Además, proporciona interfaces genéricas para estos aspectos e implementaciones específicas para cada tipo de tecnología de persistencia.



Figura 2.8: Logo de Spring Data

2.3.4.1 Patrón DAO (Data Access Object)

Este patrón de diseño software pertenece al catálogo de Core J2EE Patterns de Java, por lo que no es un patrón exclusivo de Spring, aunque es el utilizado para realizar aplicaciones con este framework que hagan uso de la persistencia.

El patrón DAO muestra una forma de envolver el conocimiento sobre el acceso a datos, mediante el uso de objetos de acceso a datos para conseguir abstraer y encapsular el acceso a dichos datos.

Un objeto DAO permite obtener y guardar datos, manejando la conexión con el sistema de persistencia que se haya implementado. Todo ello lo realiza mediante operaciones atómicas (Creación, actualización, borrado, obtención de registros) en la base de datos, por lo que nunca son necesarias las transacciones. Normalmente, se crea un objeto DAO por cada Objeto que tengamos en nuestra aplicación.

Esto nos permite que nuestra aplicación sea lo más transparente posible al sistema de almacenamiento usado, permitiendo la migración entre diferentes sistemas de diferentes fabricantes, diversos tipos de almacenamiento y diferentes almacenamientos de datos.

En nuestro proyecto, hemos hecho uso de este patrón para generar objetos DAO por cada objeto disponible en nuestra aplicación (Usuarios, Rutinas, Ejercicios y Vídeos), de forma que se usarán para realizar operaciones sobre los datos almacenados en la base de datos.

Cabe destacar que el uso de este patrón hace posible que si en un futuro deseáramos migrar nuestra base de datos a otro tipo, las modificaciones que serían necesarias en nuestra implementación no afectarían a la implementación de este modelo, sino que simplemente sería necesario ajustar la sintaxis de las consultas que realizamos para la implementación de las operaciones atómicas, hacia la sintaxis del nuevo modelo a implantar.

2.3.4.2 JDBC (Java Database Connectivity)

JDBC es una API que describe o define una librería estándar para el acceso a Fuentes de datos desde el lenguaje de programación Java, independientemente del Sistema operativo donde se ejecute o de la base de datos a la que se acceda, principalmente orientado a bases de datos relacionales.



Figura 2.9: Logo de JDBC

Esta API consiste en un conjunto de interfaces y clases escritas en Java. Con estas interfaces y clases estándar, los programadores pueden escribir aplicaciones que conecten con la base de datos, enviar consultas escritas en el lenguaje de consulta estructurada “SQL” y posteriormente procesar los resultados obtenidos.

Nosotros usamos esta para acceder a nuestro sistema de gestión de bases de datos (MySQL), y así permitir que desde nuestra aplicación Java podamos realizar operaciones sobre dicha base de datos, consiguiendo la persistencia necesaria y la disponibilidad de los datos que gestiona la aplicación.

2.3.5 Spring Beans

Un bean se define como un componente de software que tiene la particularidad de ser reutilizable. Estos deben cumplir ciertos criterios en java:

- Implementación serializable.
- Poseer todos sus atributos privados (private).
- Poseer métodos “set” y “get” públicos de los atributos privados que nos interese.
- Poseer un constructor público por defecto.

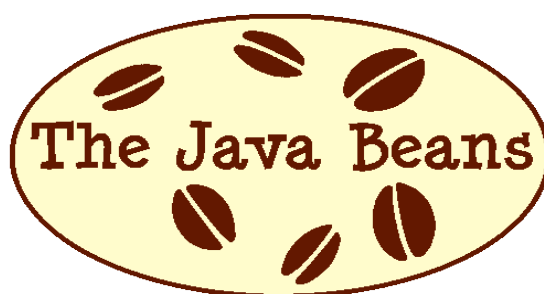


Figura 2.10: Logo de Java Beans.

A diferencia de los beans convencionales que representan una clase, la particularidad de los beans en Spring es que son objetos creados y manejados por el contenedor de Spring.

2.3.5.1 Contenedor de Spring

Este se encuentra en el núcleo del marco de trabajo de Spring y utiliza la inyección de dependencias para gestionar los componentes que forman la aplicación. Se encarga de diversas tareas, como crear, conectar y alojar los objetos definidos por los beans. También hace de repartidor, proporcionando los beans por petición. Este contenedor, carga las definiciones de beans escritas en archivos “XML” estructurados de forma ordenada. Existen dos tipos de contenedor en Spring:

- **Fábrica de Beans (Bean Factory):** Es un contenedor simple con soporte básico de inyección de dependencias.
- **Contexto de Aplicación (Application Context):** es una implementación del contenedor anterior que proporciona opciones avanzadas, como la publicación de beans registrados como receptores o formas genéricas de abrir recursos de archivo.

2.3.5.2 Ciclo de vida de un bean en Spring

Otra diferencia respecto a los beans convencionales es que a los beans de Spring se les añade un ciclo de vida nuevo para que el bean sepa cuál es su contexto de aplicación. Las fases de este ciclo de vida son las siguientes:

1. Instanciación.
2. Inyección de propiedades.
3. Nombre del Bean.
4. Nombre de la fábrica.
5. Postprocesado previo a la inicialización.
6. Inicialización.
7. Postprocesado tras la inicialización.
8. Bean listo para su uso en el contexto de aplicación.
9. Destrucción del bean.

2.3.5.3 Formas de crear un Bean

- **Bean Simple:** Es un bean sin atributos.
- **Bean con inyección por constructor:** Se pasan los atributos a través del constructor del objeto.
- **Bean con referencias de objeto de constructores:** Cuando pasamos un bean como atributo del constructor de otro objeto.
- **Bean con inyección de propiedades:** Cuando en vez del método constructor utilizamos “setters” de atributos. Existen a su vez diferentes tipos de realizarlo, pero nosotros en nuestro proyecto hemos hecho uso de la inyección de propiedades por referencia de otro objeto.

En definitiva, en nuestro proyecto se han usado los beans de Spring para definir los objetos DAO relacionados con la base de datos, como son los ejercicios, rutinas, usuarios, vídeos y el dataSource para la conexión con la base de datos.

2.3.6 Construcción y Gestión de Software (Maven)

Esta herramienta tiene un funcionamiento similar a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en formato XML. En principio, estuvo integrado dentro del proyecto Jakarta pero a día de hoy es un proyecto de nivel superior de la Apache Software Foundation.

Maven posee la capacidad de realizar operaciones y tareas definidas, como la compilación del código de la aplicación y su empaquetado junto con las dependencias necesarias para su funcionamiento. Para ello, es necesario definir todas las dependencias del proyecto (librerías externas usadas) en un fichero propio de todo proyecto Maven, el fichero POM.xml, el cual contiene todo lo necesario para que a la hora de generar el fichero ejecutable de la aplicación este contenga todo lo que necesita para su ejecución en su interior.

Otras de las características a destacar de Maven es su capacidad de trabajar en red. Cuando definimos dependencias en Maven, este sistema se encarga de ubicar las librerías que deseamos utilizar en Maven Central, el cual es un repositorio que contiene cientos de librerías constantemente actualizadas por sus creadores. Maven permite incluso buscar versiones más recientes o más antiguas de un código dado y agregarlas a nuestro proyecto.



Figura 2.11: Logo de Maven.

Todo ello se hará de forma automática sin que el usuario tenga que hacer nada más que definir las dependencias en el formato que mostramos a continuación:

```
*pom.xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>org.springframework</groupId>
  <artifactId>rutina_app</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.2.RELEASE</version>
  </parent>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>

    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web-services</artifactId>
    </dependency>

  </dependencies>

  <build>
    <plugins>
      <plugin>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
      </plugin>
    </plugins>
  </build>
```

Figura 2.12: Formato de fichero POM.xml.

Hemos usado esta herramienta para la producción de nuestra aplicación Web en un paquete de formato “.WAR” que contendrá todo el servicio web. Hemos hecho uso de Maven debido a la gran facilidad en su configuración y la automatización que proporciona para la construcción de dependencias y obtención de librerías. Además, presenta una buena integración con el IDE de Spring (Spring Tools Suite).

2.3.7 Inversión de Control (IoC)

Es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se lleva a cabo haciendo llamadas a procedimientos o funciones. En la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario, y para el conjunto de sucesos que tengan que ocurrir.

Las aplicaciones que utilicen el Framework Spring están utilizando Inversión de Control, ya que implementa un “Contenedor” que se encarga de gestionar las instancias (así como sus creaciones y destrucciones) de los objetos del usuario. Por tanto, esta funcionalidad se lleva a cabo en cualquier proyecto con Spring framework de forma transparente al usuario.

2.3.8 Inyección de dependencias (DI)

Es uno de los principales temas a tener en cuenta a la hora de realizar un buen desarrollo de una aplicación con tecnologías orientadas a objetos, como es el caso de Java. Una de las principales características de la programación orientada a objetos es que los objetos que componen una aplicación sean lo más independientes posibles, lo cual significa que queremos que las clases que definen a nuestros objetos sean autónomas a pesar de los cambios de los objetos con los que interactúa.

DI es un patrón de diseño en el que suministran objetos a una clase en lugar de ser la propia clase la que cree el objeto. Estos objetos necesitan de esa clase para funcionar, por lo que tienen una dependencia directa con las clases que lo usan, ya que se inyectará la implementación deseada a nuestro proyecto para poder funcionar, según la clase que haga uso de ellos.

Esta característica puede realizarse referenciando directamente las clases de dichas dependencias, aunque no es una buena práctica, dado que sus componentes tienen una fuerte relación entre sí y eso dificultará el mantenimiento de nuestro software. Normalmente, los frameworks usan en su lugar la inyección de dependencias mediante interfaces entre sus componentes.

El poco acoplamiento que presenta la implementación de este patrón permite que la relación entre las clases de nuestra aplicación esté separada por capas de abstracción o interfaces, consiguiendo una reutilización modular de nuestros componentes.

Este patrón ha sido aplicado a nuestro proyecto junto con la propiedad de la inversión de control mediante la definición de los beans de Spring, de una forma más transparente e interna de nuestra aplicación.

2.4 Persistencia: Base de datos

2.4.1 Introducción: Modelo Relacional

El modelo relacional es una alternativa para la organización y representación de la información que se pretende almacenar en una base de datos. Se trata de un modelo teórico matemático que, además de proporcionarnos los elementos básicos de modelado (las relaciones), incluye un conjunto de operadores (definidos en forma de un álgebra relacional) para su manipulación, sin ambigüedad posible.

Este modelo, hace relativamente sencilla su representación y gestión por medio de herramientas informáticas, siendo elegido como referencia para la gran mayoría de los sistemas de Gestión de Bases de Datos disponibles en el mercado. Además, también es seleccionado como referencia para la elaboración del esquema lógico de una base de datos en su diseño, junto con herramientas como el álgebra relacional.

2.4.2 Gestor MySQL y uso en nuestro proyecto

Es un sistema de gestión de bases de datos relacional desarrollado por Oracle Corporation, considerado como el más popular del mundo, además de ser desarrollado y distribuido libremente (Open Source).

MySQL es multiplataforma, lo cual da soporte a una amplia lista de sistemas operativos y trae soporte para aproximadamente diez motores de almacenamiento, en los cuales se encuentran InnoDB que es el motor predeterminado, que soporta transacciones y bloqueo de registros. Esto hace de este gestor que posea un alto rendimiento comparado con sistemas similares.



Figura 2.13: Logo de MySQL.

En este proyecto se ha hecho uso de MySQL para implementar la persistencia de los datos de la aplicación, es decir, para gestionar el almacenamiento de usuarios, rutinas, ejercicios y vídeos de nuestro servicio Web.

Principalmente hemos hecho uso de esta tecnología debido a la gran cantidad de información disponible en Internet para el aprendizaje de su uso, ya que es el más usado en el mundo de la informática. Esta tecnología también provee de una interfaz de conexión a una base de datos MySQL compatible con Spring, llamada JDBC, la cual se ha descrito anteriormente.

Por último, este sistema de gestión de bases de datos provee de varias herramientas de gestión y de diseño de bases de datos de forma interactiva y gráfica, como son XAMPP y MySQLWorkbench. Estas herramientas nos han facilitado el diseño y montaje de la base de datos, además de su gestión.

2.5 Tecnologías para la interfaz web de usuario

2.5.1 Patrón Modelo-Vista-Controlador (MVC)

Este patrón de diseño software fue inventado en el contexto de “Smalltalk” (Lenguaje de programación orientado a objetos) durante los años 80. El objetivo principal de este patrón es el de realizar una separación de los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello, este patrón define tres componentes o módulos básicos:

- **Modelo:** el modelo representa la parte de la aplicación que implementa la lógica de negocio. Esto significa que es responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos.
- **Vista:** es una presentación de los datos del modelo al usuario, estando separada de los objetos del modelo. Es responsable del uso de la información de la cual se dispone para producir cualquier interfaz de presentación de cualquier petición que se presente.
- **Controlador:** gestiona las peticiones de los usuarios. Es responsable de responder a la información solicitada con la ayuda tanto del modelo como de la vista.

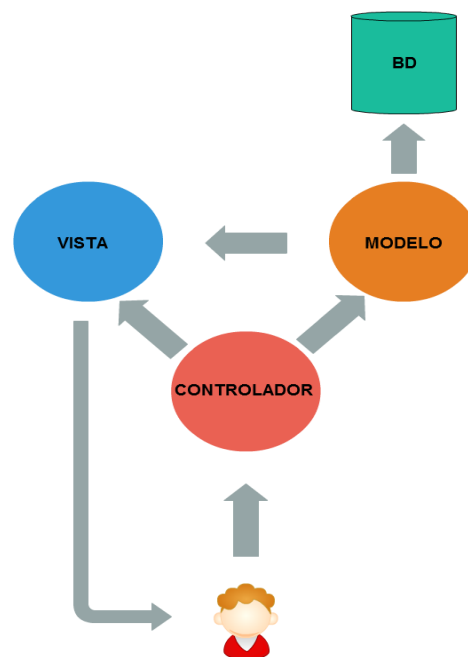


Figura 2.14: Esquema del Modelo MVC.

La ideas y objetivos principales de este modelo software son la reutilización de código y la separación de conceptos a través de una implementación modular, permitiendo el desarrollo y mantenimiento de aplicaciones de una forma más simple y eficiente. Sin embargo, para llevar a cabo este modelo se requiere una mayor dedicación en los tiempos iniciales de diseño y elaboración para su posterior desarrollo, además de ser un modelo orientado a objetos, por lo que su implementación es sumamente costosa y difícil en lenguajes que no siguen este paradigma.

Nosotros usamos este modelo en el diseño e implementación de la interfaz web del usuario, donde el modelo se identifica como el código HTML que pasa a ser la vista inicial, sobre la que se solicitarán y representarán los datos a través del controlador JavaScript correspondiente.

2.5.2 Bootstrap (HTML5 + CSS3 + JavaScript)

Bootstrap fue desarrollado por Mark Otto y Jacobd Thornton de Twitter, como un marco de trabajo para fomentar la consistencia entre las herramientas internas de diseño Web. Antes de Bootstrap, se usaban varias librerías para el desarrollo de interfaces de usuario, las cuales llevaban a inconsistencias y a una gran carga de trabajo en su mantenimiento.

Este es un framework o conjunto de herramientas de código abierto para el diseño de sitios y aplicaciones Web. Contiene plantillas de diseño de tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basados en HTML y CSS, así como extensiones de JavaScript adicionales. Además es compatible con la mayoría de los navegadores web.



Figura 2.15: Logo de Bootstrap.

Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo que implementan la variedad de componentes de la herramienta. Una hoja de estilo llamada `bootstrap.less` incluye los componentes de las hojas de estilo. Los desarrolladores pueden adaptar el mismo archivo de Bootstrap, seleccionando los componentes que deseen usar en su proyecto.

Es por tanto que hemos hecho uso de este framework en la implementación de la vista Web del usuario, junto con otras tecnologías como JQuery y Ajax para la comunicación con el servidor de la aplicación Web.

2.5.2.1 HTML5

HTML es un lenguaje de marcado utilizado para definir la estructura y contenido de una página o documento Web. La idea es utilizar un lenguaje para hacer referencia a otros documentos, como archivos, imágenes, vídeos, audio, etc.

Es por tanto que HTML5 se trata de una nueva versión de HTML en la cual se especifican nuevos elementos, atributos y comportamientos. Además, contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance.



Figura 2.16: Logo de HTML5.

2.5.2.2 CSS3

CSS es un lenguaje usado para definir el estilo o apariencia de las páginas Web, escritas con HTML o de los documentos XML. Este lenguaje se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.

A partir del año 2005 se comenzó a definir el sucesor de esta versión, al cual se conoce como CSS. Esta versión nos ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación, CSS3 es el estándar que dominará la web en cuanto a diseño gráfico.



Figura 2.17: Logo de CSS3.

La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales con la etiqueta `<style>` o en cada etiqueta particular mediante el atributo "style".

2.5.2.3 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Este estándar es también un lenguaje de programación interpretado, definido como orientado a objetos y basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado cliente, implementado como una parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cuál fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript. El cambio de nombre coincidió aproximadamente con el momento en que Netscape agregó soporte para la tecnología Java en su navegador web Netscape Navigator en la versión 2.003 en diciembre de 1995.

La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java, y se ha caracterizado por muchos como una estrategia de mercadotecnia de Netscape para obtener prestigio e innovar en lo que eran los nuevos lenguajes de programación web.

Este lenguaje fue diseñado con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.



Figura 2.18: Logo de JavaScript.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

El modelo DOM es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.



Figura 2.19: Logo de DOM.

Tradicionalmente, JavaScript se venía utilizando en páginas Web HTML para realizar operaciones y únicamente en el marco de la aplicación del cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con la ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

2.5.3 Controlador de cliente: JQuery/Ajax

2.5.3.1 JQuery

JQuery es la biblioteca ECMAScript más extendida y usada en el mundo. Su autor original es John Resig que la presentó en 2006 en el BarCamp. Aunque como sucede con todas las librerías exitosas, actualmente recibe contribuciones de decenas de programadores. JQuery ha sido programada de forma muy eficiente y su versión compacta apenas ocupa 90 KB. La documentación de JQuery es muy completa e incluye muchos ejemplos. Además, también existen algunos recursos útiles para aprender su funcionamiento básico.

Esta biblioteca de software libre y de código abierto permite simplificar la forma de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción mediante AJAX a páginas Web.



Figura 2.20: Logo de JQuery.

2.5.3.2 AJAX, Asynchronous JavaScript And XML

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas completamente, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.



Figura 2.21: Logo de AJAX.

Es una técnica asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. No es una tecnología en sí mismo, sino se trata de varias tecnologías que se unen para colaborar.

Las tecnologías que pueden formar parte de una “aplicación” AJAX son las mostradas en la siguiente figura:

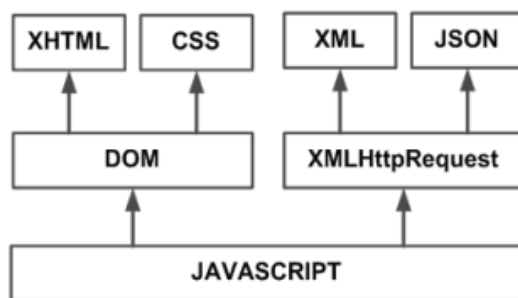


Figura 2.22: Esquema de tecnologías aplicadas en AJAX.

AJAX permite mejorar la interacción del usuario con la aplicación, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

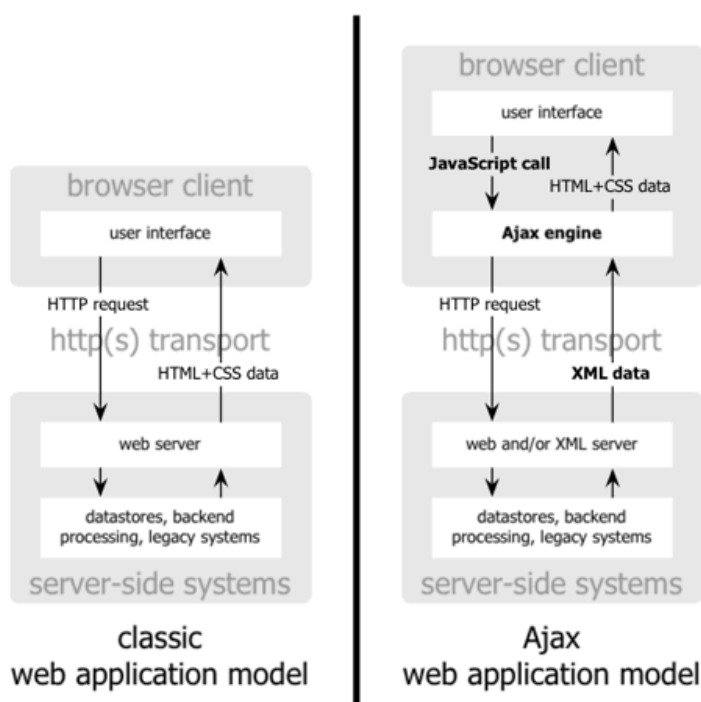


Figura 2.23: Esquema de aplicación Web tradicional y aplicación Web con AJAX.

El esquema anterior muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX.

2.5.3.3 JQuery AJAX

JQuery facilita el trabajo con AJAX mediante métodos que simplifican todo el proceso anterior. La más importante es “.ajax()”, que realiza una petición asíncrona y tiene multitud de opciones.



Figura 2.24: Logo de JQuery AJAX.

En nuestro caso, usaremos AJAX a través del método \$.ajax() que es la implementación de dicha tecnología a través de JavaScript, concretamente de la biblioteca JQuery, que es configurado a través de un objeto, el cual contiene todas las instrucciones que necesita JQuery para completar una petición de este tipo. Dicho método es muy útil debido a que ofrece la posibilidad de especificar acciones en caso de que la petición haya fallado o no. Además, al estar configurado a través de un objeto, es posible definir sus propiedades de forma separada, haciendo que sea más fácil la reutilización del código.

```
$.ajax({
  url: '/ruta/hasta/pagina.php',
  type: 'POST',
  async: true, // Valor por defecto, no necesario
  data: 'parametro1=valor1&parametro2=valor2',
  // Datos que se envían al servidor
  success: procesaRespuesta,
  // función que procesa los datos devueltos por el servidor
  error: muestraError // función invocada si la petición falla
});
```

Figura 2.25: Ejemplo de sintaxis de una consulta simple en AJAX.

2.5.3.4 JQuery Validation Plugin

Este es un Plugin de JQuery cuya finalidad es realizar la validación de formularios HTML en el lado del cliente, ofreciendo una gran cantidad de opciones personalizables. Por defecto, el idioma que incluye y que muestra por tanto en los mensajes de alerta al realizar la validación de un formulario es el inglés, aunque pueden configurarse otros.



Figura 2.26: Logo de JQuery Plugin Validation.

3 HERRAMIENTAS UTILIZADAS

La tecnología no es nada. Lo importante es que tengas fé en la gente, que sean básicamente buenas e inteligentes, y si les das herramientas, harán cosas maravillosas con ellas.

Steve Jobs.

Esta sección describirá las herramientas de desarrollo utilizadas para llevar a cabo este proyecto. Cabe destacar que la elección de las herramientas adecuadas es crucial a la hora de conseguir el fin que se persigue, además de ser más eficientes y compactos, con la elección de herramientas de desarrollo lo más completas posible. Pero no sirve de nada el uso de estas herramientas si no se tiene conocimiento previo sobre ellas y sobre su funcionalidad y uso en el ámbito del proyecto.

3.1 Spring Tool Suite (STS)

Spring Tool Suite proporciona un entorno *ready-to-use* para implementar, depurar, ejecutar y desplegar las aplicaciones Spring, incluyendo integraciones para Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ, y viene encima de las últimas versiones de Eclipse.

Este entorno de desarrollo incluye la edición para desarrolladores de Pivotal tc Server, una versión de Apache Tomcat optimizado para Spring. Con su consola Spring Insight, tc Server Developer Edition ofrece una visión en tiempo real gráfica de los parámetros de rendimiento de aplicaciones que permite a los desarrolladores identificar y diagnosticar los problemas desde sus escritorios.

Además, soporta el despliegue de aplicaciones tanto en servidores locales, virtuales y en la nube. Es de libre acceso para el desarrollo y uso en operaciones internas sin límite de tiempo, completamente de código abierto y licenciada bajo los términos de la Licencia Pública Eclipse.



Figura 3.1: Logo de Spring Tool Suite.

Algunas de las características que podemos destacar son las siguientes:

- ✓ **Validaciones para la configuración de Spring:** ofrece un amplio conjunto de validaciones que se están aplicando de forma automática. Esas validaciones indican errores en las configuraciones directamente en el IDE, mucho antes de que sea ejecutada la aplicación. Encontrar problemas y errores de configuración es mucho más fácil.
- ✓ **Soporte Refactoring para su aplicación Spring:** el término refactorización en ingeniería del Software es usado para describir la modificación del código fuente sin alterar el comportamiento del mismo. Es una de las partes más importantes de la ingeniería de software de hoy. El IDE agrega nuevas refactorizaciones para los elementos de Spring (como el cambio de nombre de los beans de Spring, por ejemplo).
- ✓ **Code Assistant:** proporciona contenido de ayuda significativo en todos lados, junto con soluciones rápidas para los errores y problemas comunes.
- ✓ **Soporte AOP (Programación Orientada a Aspectos):** la Spring Tool Suite se integra con las herramientas de Eclipse y proporciona el soporte más completo para AOP disponible hoy en día.
- ✓ **Integrado con Cloud Foundry y Pivotal tc Server:** permite el despliegue de las aplicaciones directamente en Cloud Foundry o una instancia tc Server (incluyendo el soporte para la depuración, creación de la instancia, Spring Insight, servicios y más).

En nuestro proyecto, hemos elegido este entorno (IDE) para el desarrollo completo de nuestra aplicación Web, es decir, tanto para la programación Web en cliente como en Servidor.

Este IDE nos ha facilitado mucho las tareas de configuración e implementación de nuestro servicio WebREST, ya que es un entorno especializado y adaptado para el desarrollo de este tipo de aplicaciones.

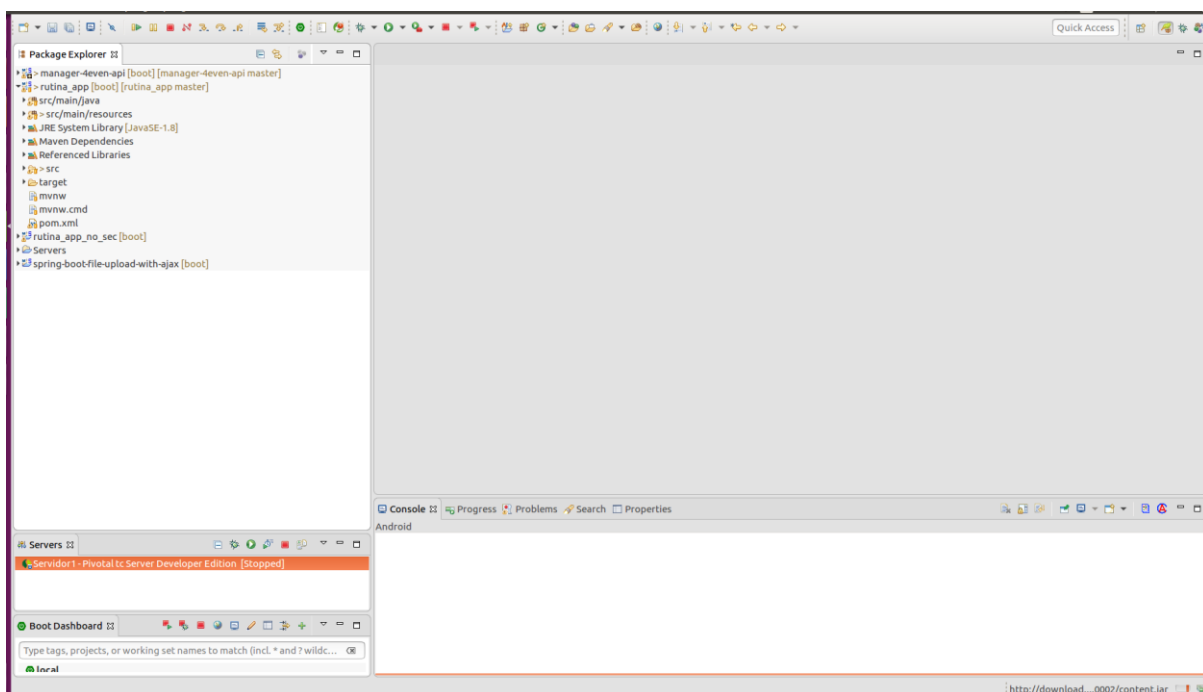


Figura 3.2: Interfaz Gráfica de STS.

3.2 MySQLWorkbench

Esta es una herramienta gráfica de diseño de bases de datos para trabajar con servidores y bases de datos MySQL. Es el sucesor de DBDesigner 4, y reemplaza el anterior conjunto de software, MySQL GUI Tools Bundle.

La funcionalidad de esta herramienta abarca cinco tareas fundamentales:

- ✓ **Desarrollo de SQL:** Permite crear y gestionar las conexiones a los servidores de bases de datos y configurar los parámetros de conexión. MySQL Workbench proporciona la capacidad de ejecutar consultas SQL en las conexiones de base de datos utilizando el built-in editor de SQL.
- ✓ **Modelado de datos (Diseño):** Permite crear modelos de bases de datos de forma gráfica, y editar todos los aspectos de la base de datos con el Editor de la tabla completa. El Editor de tablas ofrece instalaciones de fácil uso para la edición de tablas, columnas, índices, triggers, creación de particiones, opciones, inserciones y privilegios, Rutinas y vistas.
- ✓ **Administración del Servicio:** Permite administrar instancias del servidor MySQL mediante la administración de usuarios, la realización de copias de seguridad y recuperación, la inspección de los datos, la visualización del estado de la base de datos, y el seguimiento de su rendimiento.
- ✓ **Migración de datos:** Da soporte para la migración a otros gestores de bases de datos como Microsoft SQL Server, Microsoft Access, Sybase ASE, SQLite, SQL Anywhere, PostgreSQL. La migración también es compatible con la migración desde versiones anteriores de MySQL a las últimas versiones.
- ✓ **MySQL Enterprise Support:** Soporte para productos empresariales como MySQL Enterprise Backup, MySQL Firewall y auditoría de MySQL.



Figura 3.3: Logo de MySQL WorkBench.

En el desarrollo de este proyecto se ha hecho uso de esta herramienta para diseñar la base de datos que establecerá la persistencia de los datos de nuestra aplicación, con un modelo de datos relacional. Este diseño se exportará y desplegará con XAMPP, otra herramienta que veremos a continuación.

3.3 XAMPP

XAMPP es un paquete de instalación independiente de plataforma, software libre, que consiste en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

El nombre proviene del acrónimo de X (Multiplataforma), Apache, MariaDB, PHP, Perl. Desde la versión "5.6.15", XAMPP cambió la base de datos de MySQL a MaríaDB. El cual es un fork de MySQL con licencia GPL.

El programa se distribuye bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.



Figura 3.4: Logo de XAMPP.

XAMPP provee de una configuración totalmente funcional desde el momento de su instalación, sin embargo, es bueno decir que la seguridad de datos no es su punto fuerte, por lo cual no es suficientemente seguro para ambientes grandes o de producción. La filosofía de XAMPP, como lo indican en su sitio web, es crear una distribución fácil de instalar, de tal manera que los desarrolladores web principiantes cuenten con todo lo necesario ya configurado. XAMPP solamente requiere descargar y ejecutar un archivo “.zip”, “.tar”, o “.exe”, con unas pequeñas configuraciones en alguno de sus componentes que el servidor Web necesitará.

Es por tanto, que hemos hecho uso de esta herramienta para administrar y ejecutar la base de datos de nuestra aplicación Web, accediendo de forma gráfica a un panel de administración que provee, llamado phpMyAdmin.



Figura 3.5: Logo de phpMyAdmin.

3.4 Advanced REST Client

Advanced Rest Client es una extensión de Chrome que nos permite lanzar peticiones a servicios o APIs REST. Puedes hacer cualquier tipo de petición además de las habituales GET, POST, PUT y DELETE. Permite pasar parámetros a las peticiones y muestra el resultado devuelto por el servicio que queremos probar.



Figura 3.6: Logo de Advanced REST Client.

Su funcionamiento es muy simple y no requiere ningún tipo de configuración especial, ya que solo tenemos que definir la operación a realizar, así como la URL del recurso al que queremos realizar la petición. Tras la petición, podemos obtener la respuesta en varios formatos de intercambio de datos, como puede ser XML o JSON.

En nuestro proyecto hemos hecho uso de esta herramienta para llevar a cabo la depuración del servicio Web REST, comprobando que todos los métodos del servicio están disponibles y su comportamiento es el correcto. Tras esta comprobación con dicha herramienta, se ha procedido a implantar los controladores con JQuery/Ajax que realizarán dichas peticiones en el navegador Web del cliente.

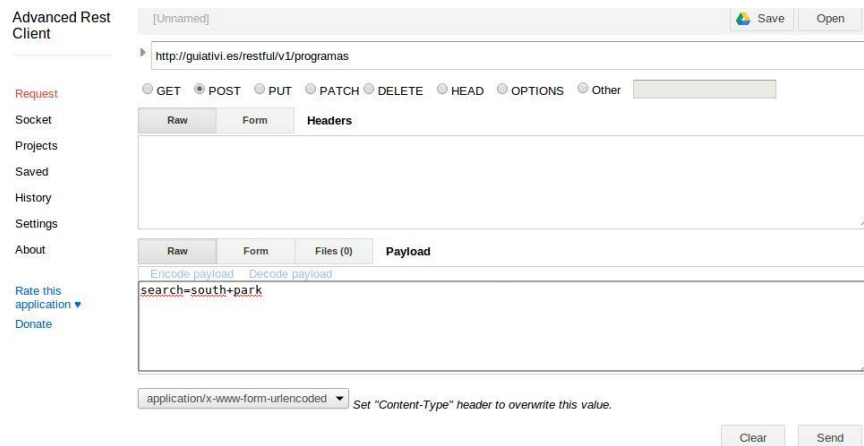


Figura 3.7: Interfaz Gráfica de Advanced REST Client.

3.5 Google Chrome

Google Chrome es un navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones (frameworks) de código abierto, como el motor de renderizado Blink (bifurcación o *fork* de WebKit).

Está disponible gratuitamente bajo condiciones específicas del software. El nombre del navegador deriva del término en inglés usado para el marco de la interfaz gráfica de usuario («chrome»).



Figura 3.8: Logo de Google Chrome.

Chromium es el proyecto de software libre con el que se ha desarrollado Google Chrome y es de participación comunitaria (bajo el ámbito de *Google Code*) para fundamentar las bases del diseño y desarrollo del navegador Chrome (junto con la extensión Chrome Frame), además del sistema operativo Google Chrome OS.

Cuenta con más de 750 millones de usuarios, y dependiendo de la fuente de medición global, puede ser considerado el navegador más usado de la Web variando hasta el segundo puesto, algunas veces logrando la popularidad mundial en la primera posición.

Entre todas sus funcionalidades y características cabe destacar que cuenta con un soporte en modo de tienda Web para agregar extensiones (por ejemplo, la descrita anteriormente, Advanced Rest Client), con más de 12000 extensiones disponibles hasta el momento.

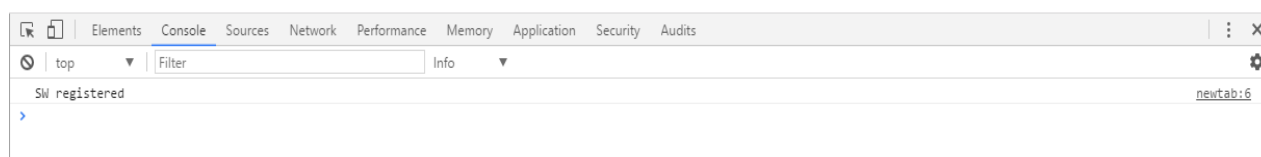


Figura 3.9: Interfaz Gráfica del depurador en modo consola de Google Chrome.

Se ha hecho uso de Google Chrome para el uso de herramientas de depuración del código del cliente (MVC Web), y para la interacción del usuario con la aplicación Web y los distintos servicios que presta al usuario.

3.6 EditThisCookie

Edit This Cookie es una extensión para Google Chrome con la que podremos gestionar las cookies que se generan durante nuestra navegación usando Google Chrome y, por tanto, podremos añadir, eliminar, borrar o modificar *cookies* además de realizar búsquedas entre las que tengamos, bloquearlas (evitar que se generen) o ponerlas en modo “solo lectura”.



Figura 3.10: Logo de EditThisCookie.

Todas estas funciones estarán accesibles desde la esquina superior derecha del interfaz de Chrome, es decir, donde se concentran todos los iconos de las extensiones (justo al lado de la barra de búsquedas, el *Omnibox*).



Figura 3.11: Esquema de la extensión de EditThisCookie en Google Chrome.

Este tipo de utilidades es algo que todo usuario debería tener y utilizar con frecuencia, de hecho, en el caso de trabajar con aplicaciones web, nos puede ayudar a eliminar sesiones abiertas y, por ejemplo, ahorrarnos algún que otro reinicio del explorador. En nuestro proyecto la hemos usado para depurar las cookies de sesión y para la protección CSRF de la aplicación Web.

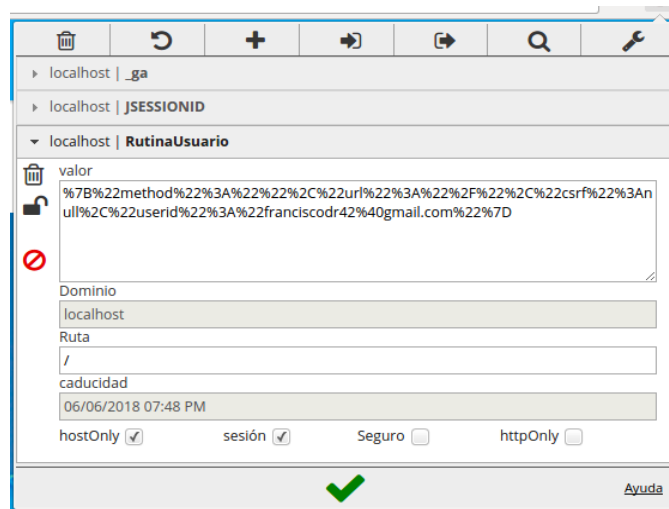


Figura 3.12: Interfaz Gráfica de EditThisCookie.

3.7 Git / GitHub

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.



Figura 3.13: Logo de Git.

GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Utiliza el framework Ruby on Rails por *GitHub, Inc.* (anteriormente conocida como *Logical Awesome*). Desde enero de 2010, GitHub opera bajo el nombre de *GitHub, Inc.* El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

Esta plataforma cuenta con una Wiki para cada proyecto y una página Web, un gráfico para ver como los desarrolladores trabajan en sus repositorios y ramas o bifurcaciones del proyecto, funcionalidades de una red social (seguidores), herramientas para trabajo colaborativo entre programadores y un gestor de proyectos del estilo kanban.



Figura 3.14: Logo de GitHub.

Hemos hecho uso de estas herramientas para tener un control de versiones sobre nuestro proyecto, de forma que así podemos tener siempre salvados todos los cambios realizados en el código con posibilidad de contrastarlos con versiones anteriores y recuperar dichos cambios.

A continuación, mostramos la sincronización del código de la aplicación Web en GitHub, donde podemos observar algunas características como el número de versiones de nuestro proyecto, las ramas, tareas pendientes, etc.

The screenshot shows the GitHub repository page for 'fjdr778 / TFG'. The repository has 13 commits, 1 branch, 0 releases, and 1 contributor. The latest commit is from 2 days ago. The repository is currently on the 'master' branch. The file list includes: .mvn/wrapper (Segundo Commit, 22 days ago), src/main (02-06-2017/17:26h Videos y datos de usuario arreglados, 4 days ago), .gitignore (Segundo Commit, 22 days ago), TFG_FcoJoseDiazRomero.docx (Add files via upload, 2 days ago), mvnw (Segundo Commit, 22 days ago), mvnw.cmd (Segundo Commit, 22 days ago), and pom.xml (24/05/2017:12:00h Solucionado el DownLoad y Despliegue con .war en Se..., 13 days ago). There is a button to 'Add a README'.

File	Commit	Time
.mvn/wrapper	Segundo Commit	22 days ago
src/main	02-06-2017/17:26h Videos y datos de usuario arreglados	4 days ago
.gitignore	Segundo Commit	22 days ago
TFG_FcoJoseDiazRomero.docx	Add files via upload	2 days ago
mvnw	Segundo Commit	22 days ago
mvnw.cmd	Segundo Commit	22 days ago
pom.xml	24/05/2017:12:00h Solucionado el DownLoad y Despliegue con .war en Se...	13 days ago

Figura 3.15: Interfaz Gráfica de GitHub.

3.8 Waffle.io

Waffle es una herramienta Open Source de seguimiento de trabajo automatizado para desarrolladores de software y jefes de proyecto construido en GitHub.



Figura 3.16: Logo de Waffle.io.

Esta herramienta coloca una nota dinámica en el archivo README.txt de GitHub de un proyecto, para que todos vean donde pueden ayudar. En otras palabras, hace uso de las tareas (Issues) de GitHub como modelo de datos, así que cuando se realiza un cambio en el tablero de Waffle, se actualiza automáticamente como un issue de GitHub. De la misma forma ocurre cuando creas el issue en GitHub, creándose este automáticamente en el tablero de Waffle.

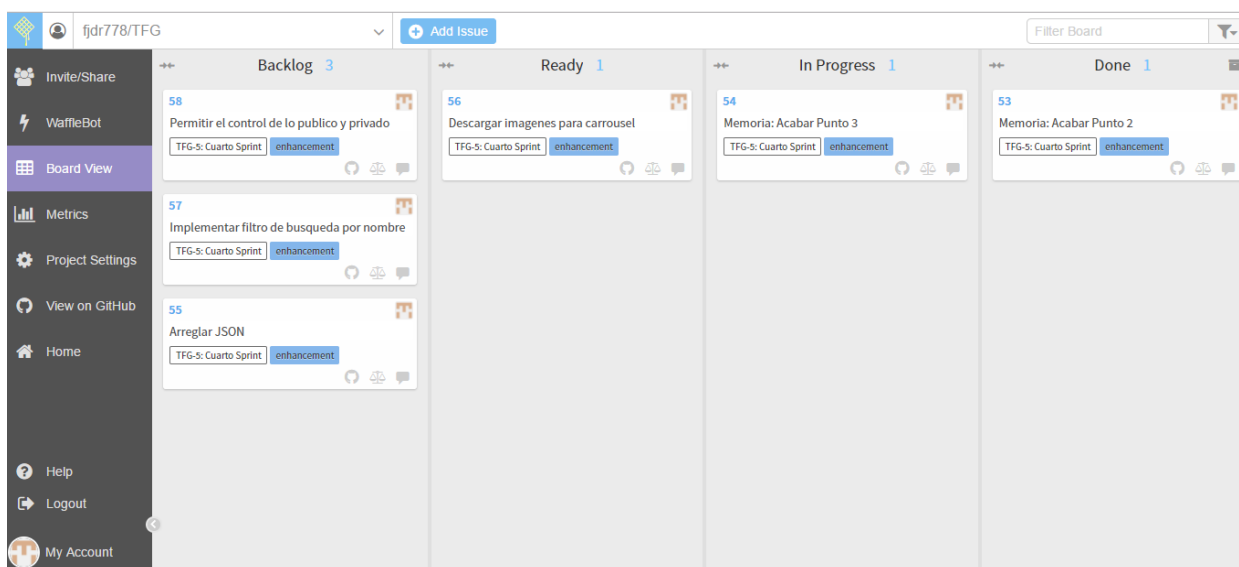


Figura 3.17: Interfaz Gráfica de Waffle.io.

3.8.1 Uso en nuestro proyecto (Metodología Scrum)

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Esta metodología divide el trabajo en una lista de pequeñas funcionalidades ordenadas por prioridad y estimadas en esfuerzo. Estas pequeñas tareas son abordadas en iteraciones cortas de longitud fija de 1 a 4 semanas (en este caso se ha optado por iteraciones de 1 o 2 semanas), obteniendo código potencialmente entregable, siendo demostrado al final de cada iteración.

El cliente (en este caso la tutora de proyecto) de esta forma se ve involucrado en el proceso de entregas y demostraciones, para ir así siguiendo el estado del proyecto en cada iteración e ir actualizando las prioridades en base a la inspección de los entregables.

Por último, se va realizando un seguimiento periódico del estado del proyecto y de la situación de cada integrante (en este caso solo es uno) mediante una actividad previa diaria de unos 15 minutos para situar el punto de partida de trabajo diario, específica, que serían reuniones diarias de apenas 15 minutos denominadas “Daylis”.

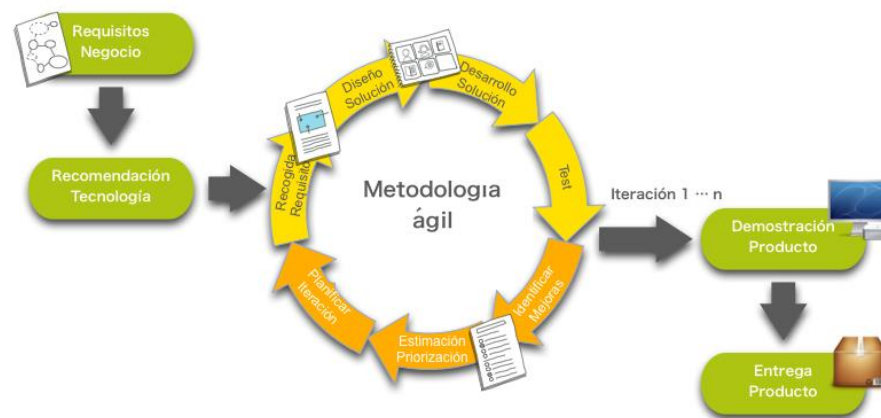


Figura 3.18: Esquema Gráfico de Metodología Scrum.

4 ARQUITECTURA Y ANÁLISIS

La simplicidad llevada al extremo se convierte en
elegancia

Jon Franklin.

En esta sección se detallará la arquitectura de la aplicación que se ha llevado a cabo, haciendo uso de un conjunto de diagramas basados en el estándar UML, permitiendo analizarla de forma detallada. Para ello, haremos uso de tres tipos de diagramas UML: Diagramas de Casos de uso, Diagramas de secuencia y Diagramas de Clases. Además, se incluirá un diagrama de Entidad-Relacion para mostrar el diseño de la base de datos.

4.1 Diagramas de Clases

Los diagramas de Clases son un tipo de diagramas estáticos que describen la estructura de un sistema mostrando sus clases, atributos, métodos y las relaciones entre clases.

La estructura de estos diagramas es muy similar para la gestión de los distintos tipos de datos que maneja la aplicación Web (Usuarios, Rutinas, Ejercicios y Vídeos). Esta estructura se caracteriza por las siguientes clases:

- **UriConstants:** Clase donde se definen las constantes que describen las URI's para acceder a los recursos de la aplicación Web.
- **SQLConstants:** Clase que contiene el conjunto de consultas realizadas en el servicio Web para acceder a la base de datos y operar sobre ella.
- **UsuarioController, RutinaController, EjercicioController y VideoController:** estas clases contienen los controladores REST para realizar operaciones sobre la base de datos a partir de las operaciones CRUD (GET, POST, PUT, DELETE) recibidas por el cliente Web a través del servicio Web. Para cada uno de ellos se establece una URI de distinto tipo ya que cada uno maneja unos recursos determinados y realiza unas operaciones determinadas sobre ese recurso.
- **UsuarioDaoImpl, RutinaDaoImpl, EjercicioDaoImpl y VideoDaoImpl:** En estas clases se implementan los métodos que realizan las operaciones para interactuar entre la BBDD y el Controlador Web. Para ello se utiliza el modelo DAO descrito anteriormente. Estas clases son implementaciones de las siguientes interfaces: **UsuarioDao, RutinaDao, EjercicioDao y VideoDao.**
- **Usuario, Rutina, Ejercicio y Video:** Estas clases definen los objetos que maneja el servicio Web, y en ellas se define el constructor, los atributos y métodos correspondientes a cada objeto.
- **UsuarioRowMapper, RutinaRowMapper, EjercicioRowMapper y VideoRowMapper:** Estas clases permiten la realización de un mapeo entre los datos obtenidos de la BBDD a través de un objeto DAO a un objeto Java determinado.

A continuación, mostraremos los diferentes diagramas de clases en los que se estructura nuestra aplicación, pudiendo observar la similitud existente entre los distintos tipos de objetos que gestiona este servicio Web.

4.1.1 Gestión de Usuarios



Figura 4.1: Diagrama de clases de la Gestión de Usuarios.

4.1.2 Gestión de Rutinas

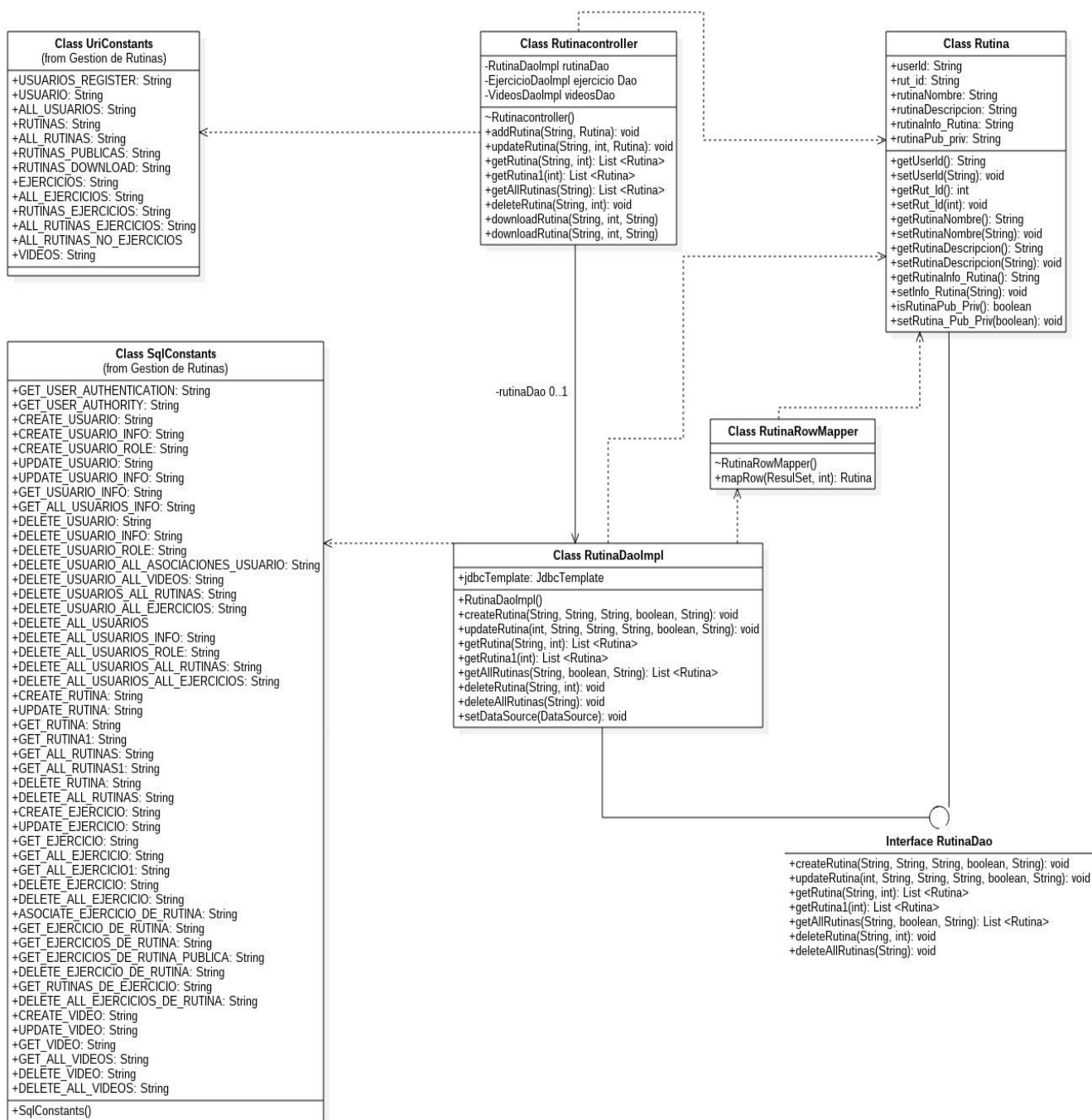


Figura 4.2: Diagrama de Clases de la Gestión de Rutinas

4.1.3 Gestión de Ejercicios

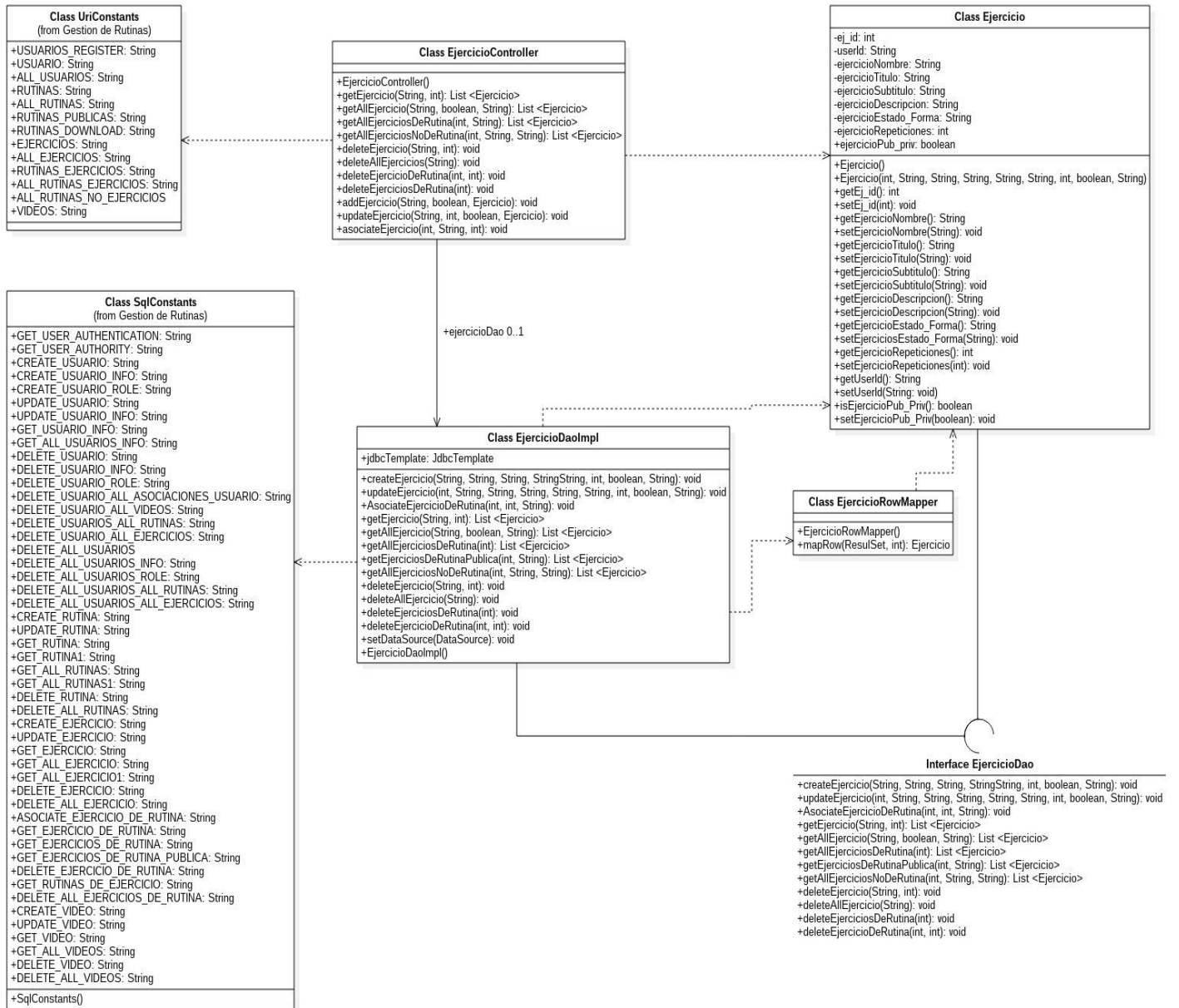


Figura 4.3: Diagrama de Clases de la Gestión de Ejercicios.

4.1.4 Gestión de Vídeos

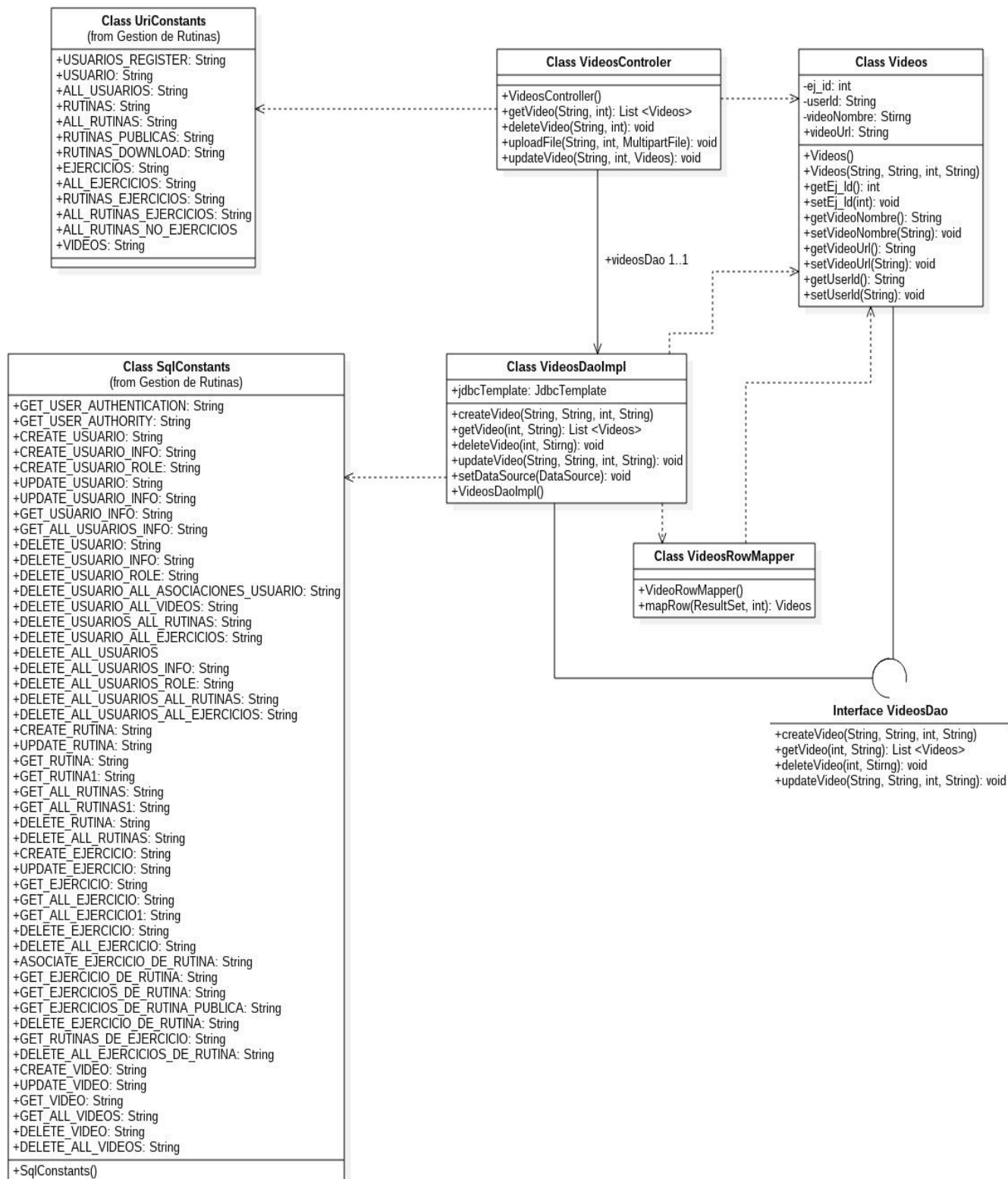


Figura 4.4: Diagrama de Clases de Gestión de Vídeos.

4.1.5 Gestión de la Seguridad (Spring Security)

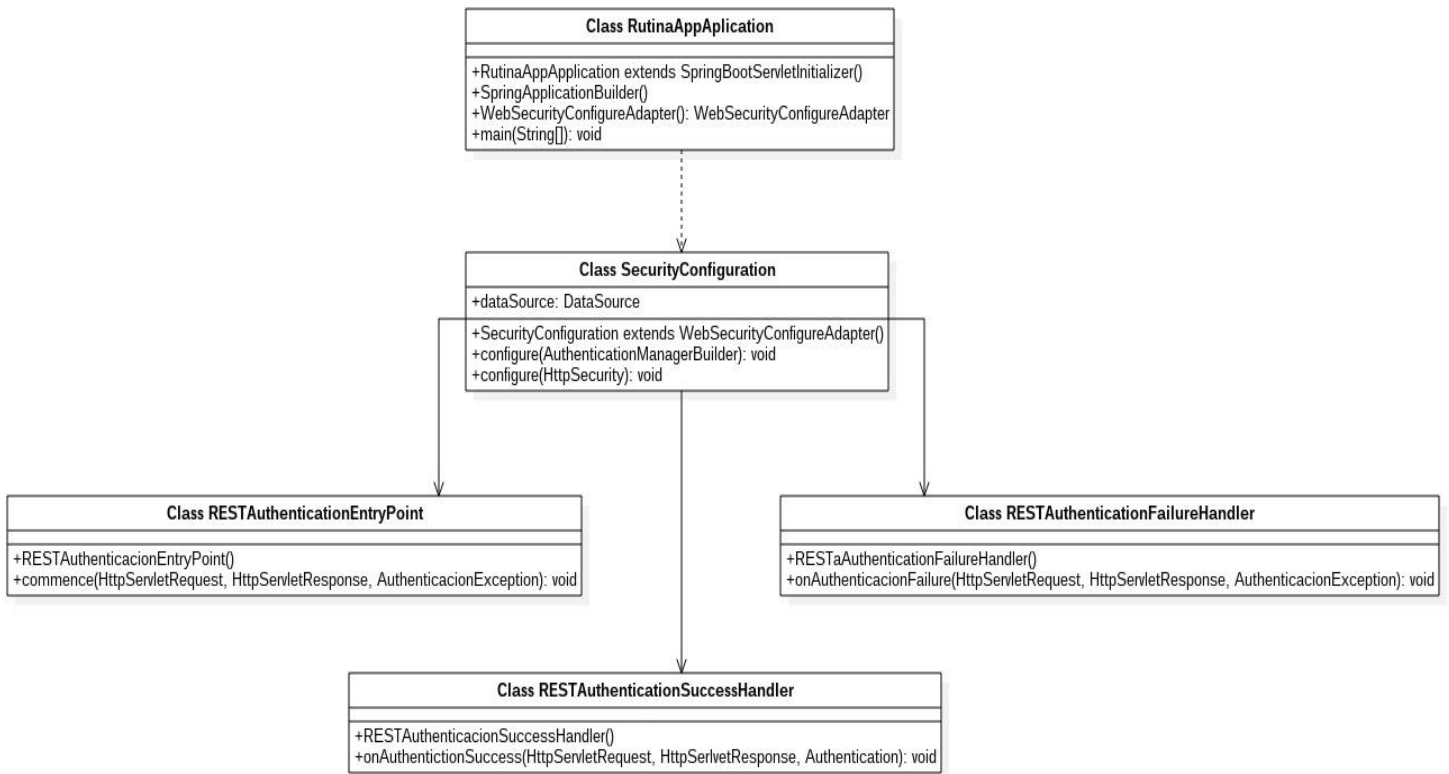


Figura 4.5: Diagrama de Clases de la Gestión de Seguridad con Spring Security.

4.2 Diagramas de Casos de Uso

En este punto detallaremos los casos de uso de la aplicación Web y sus diagramas para comprender su funcionamiento. Por consiguiente, se detallará en una tabla los actores del sistema, además de las acciones de dichos actores y su descripción, y la respuesta del sistema.

4.2.1 Identificación de los actores del sistema

En nuestra aplicación, pueden identificarse dos actores:

- ✓ **Usuario Autenticado:** Este usuario está
- ✓
- ✓ ya dado de alta en la aplicación y dispone de un perfil de usuario en la misma, creado en la plataforma de Gestión de Rutinas de Entrenamiento. Tras Autenticarse introduciendo sus credenciales de usuario (Email/Contraseña), dicho usuario puede acceder a todas las funciones que le ofrece la aplicación, para la gestión de rutinas de entrenamiento. Cabe destacar que el usuario tiene disponibles funcionalidades distintas según si está accediendo a las Rutinas y Ejercicios Propias, o a las Rutinas y Ejercicios Públicos de otros Usuarios.
- ✓ **Usuario no Autenticado:** En este caso, se trataría de un usuario que aún no está registrado en la aplicación, por lo que no dispone aún de un perfil de usuario, y debe darse de alta en el servicio, o bien, de un usuario que está dado de alta en el servicio y desea autenticarse. Además, en ambos casos se permite la descarga de la aplicación en Android de Rutina App.

Se hará uso de las siguientes abreviaciones para referirnos a los distintos actores y elementos en los diagramas, con el fin de ser eficientes en cuanto a espacio, principalmente en los casos de uso:

- **UA:** Usuario Autenticado.
- **UNA:** Usuario No Autenticado.
- **SGRE:** Sistema de Gestión de Rutinas de Entrenamiento.
- **BBDD:** Base de Datos del SGRE.

4.2.2 Usuario no Autenticado

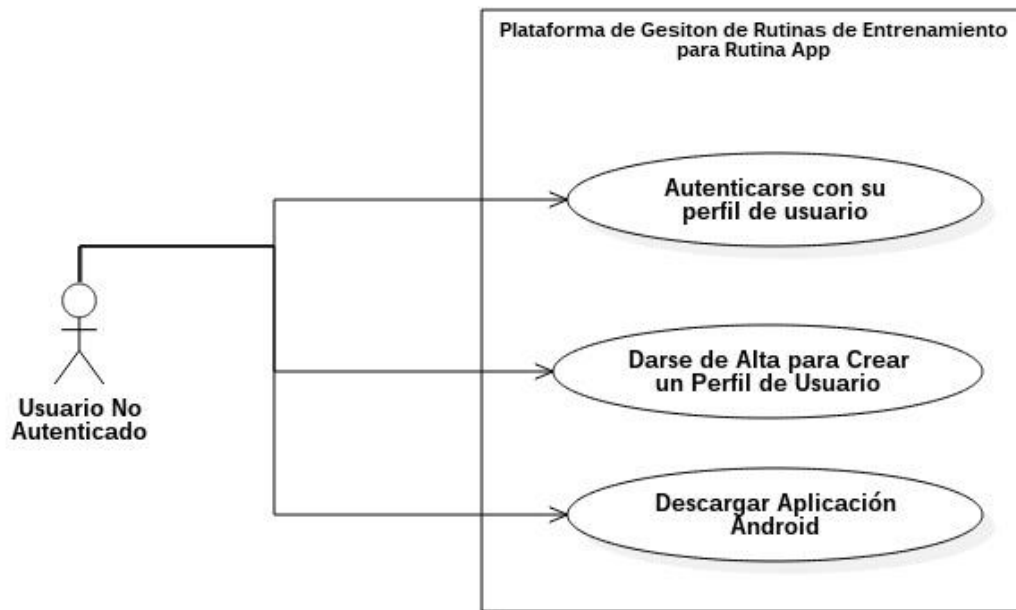


Figura 4.6: Diagrama de Casos de Uso de Acciones para Usuario No Autenticado.

Caso de Uso	<i>Operaciones disponibles para un UNA.</i>
Objetivo	<i>Realizar operaciones sobre el SGRE.</i>
Actores	<i>Usuario No Autenticado (UNA)</i>
Disparador	<i>Este caso de uso comienza cuando un usuario quiere acceder al servicio Web. Carga de la página Web de inicio</i>
Precondiciones	<i>El UNA no debe haberse autenticado previamente o no debe disponer de una cuenta de usuario</i>
Descripción	<i>El UNA accede a las operaciones que no necesitan autenticación de usuario, respecto del SGRE.</i>
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
<i>UNA se autentica en el SGRE (Ya sería UA)</i>	<i>El SGRE devuelve una cookie de sesión para dar autorización al UA a acceder a sus datos.</i>
<i>UNA se da de alta en el SGRE</i>	<i>El SGRE registra sus datos y le da la bienvenida</i>
<i>UNA descarga la aplicación Android Rutina App</i>	<i>El SGRE permite al UNA dicha descarga</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.2.3 Usuario Autenticado

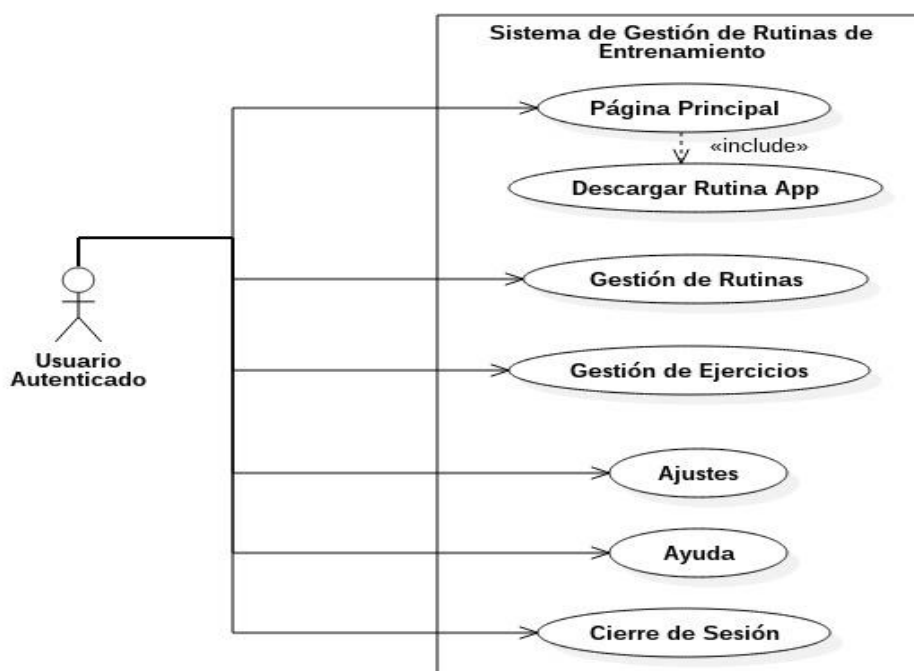


Figura 4.7: Diagrama de Casos de Uso con Operaciones Disponibles para Usuario Autenticado.

Caso de Uso	<i>Operaciones disponibles para un UA</i>
Objetivo	<i>Realizar operaciones desde su cuenta sobre el SGRE</i>
Actores	<i>Usuario Autenticado (UA)</i>
Disparador	<i>El UA se autentica con Éxito. Página Principal</i>
Precondiciones	<i>El UA debe haberse autenticado correctamente en el SGRE</i>
Descripción	<i>El UA accede a las operaciones del SGRE que requieren Autenticación de Usuario</i>
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
<i>El UA accede a la Página de Inicio del SGRE</i>	<i>El SGRE muestra un Carrusel de imágenes</i>
<i>El UA Gestiona sus Rutinas, y las Rutinas Publicadas por otros usuarios, pero con menos operaciones disponibles</i>	<i>El SGRE muestra las rutinas registradas por el usuario y las operaciones que puede realizar sobre los mismos</i>
<i>El UA Gestiona sus Ejercicios, y los Ejercicios Publicados por otros usuarios, pero con menos operaciones disponibles</i>	<i>El SGRE muestra los ejercicios registrados por el usuario y las operaciones que puede realizar sobre los mismos</i>

<i>El UA accede a los Ajustes</i>	<i>El SGRE permite dos opciones: Modificar los datos de Perfil de usuario, o Dar de Baja al Usuario.</i>
<i>El UA accede a la opción de ayuda.</i>	<i>El SGRE muestra información de ayuda al usuario y da la posibilidad al usuario de mandar un Email al soporte técnico de la aplicación</i>
<i>El UA cierra la sesión de usuario (Pasa a ser UNA).</i>	<i>El SGRE procede a invalidar la cookie de sesión del usuario</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.2.4 Gestión de Rutinas

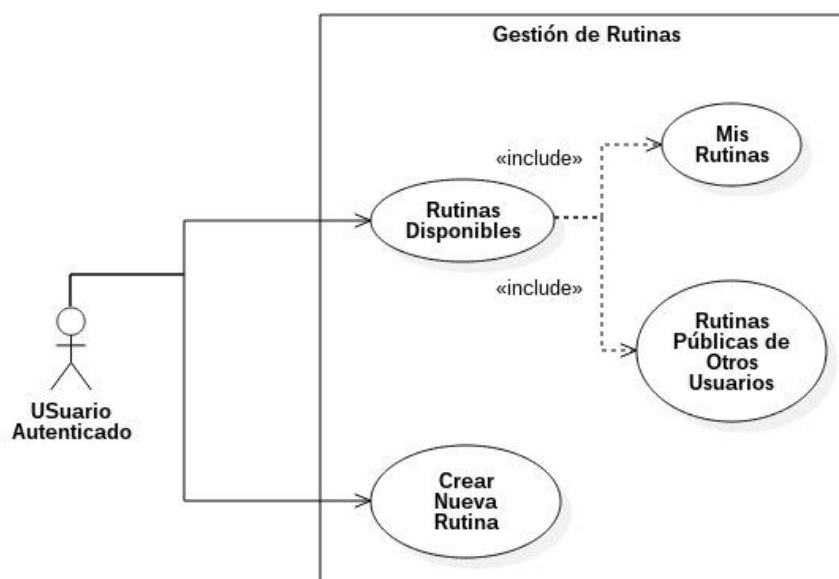


Figura 4.8: Diagrama de Casos de Uso de Gestión de Rutinas de UA.

Caso de Uso	<i>Gestión de Rutinas para UA</i>
Objetivo	<i>Realizar operaciones sobre la información de las Rutinas tanto del propio usuario como de las Rutinas publicadas por otros usuarios.</i>
Actores	<i>Usuario Autenticado (UA)</i>
Disparador	<i>El UA hace clic sobre la pestaña Rutinas</i>
Precondiciones	<i>El UA debe haberse autenticado correctamente en el SGRE</i>
Descripción	<i>El UA puede acceder crear una Rutina, decidiendo si publicarlo o no. Además, puede visualizar las Rutinas propias o las públicas de otros usuarios.</i>
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
<i>El UA accede a las Rutinas disponibles</i>	<i>El SGRE muestra, por defecto, las Rutinas del propio usuario, dándose la opción de poder acceder a las Rutinas Públicas de Otros Usuarios</i>
<i>El UA accede a crear una nueva Rutina</i>	<i>El SGRE inserta una nueva Rutina en la BBDD</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.2.4.1 Mis Rutinas

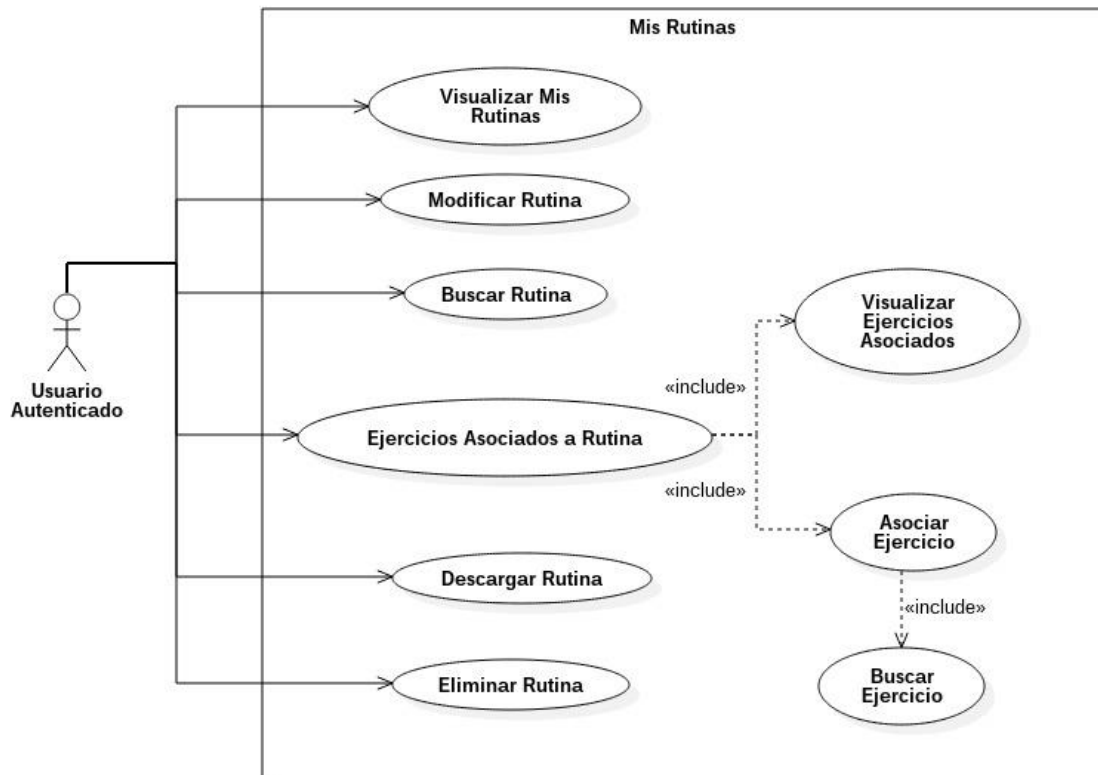


Figura 4.9: Diagrama de Casos de Uso de la Gestión de las Rutinas propias de un UA.

Caso de Uso	Mis Rutinas
Objetivo	Realizar operaciones sobre las Rutinas del propio UA
Actores	Usuario Autenticado (UA)
Disparador	El UA hace clic sobre la pestaña Mis Rutinas
Precondiciones	El UA debe haberse autenticado correctamente en el SGRE
Descripción	<p>El UA puede acceder a sus Rutinas, permitiendo la modificación, visualización, descarga o eliminación de la rutina cuyo propietario sea el UA</p> <p>Además, se permite visualizar los ejercicios asociados a una rutina, permitiendo asociar más ejercicios de los que haya disponibles</p>
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
El UA visualiza sus rutinas	El SGRE muestra las rutinas del propio usuario
El UA modifica sus rutinas	El SGRE modifica los datos de una rutina en la BBDD
El UA elimina sus rutinas	El SGRE elimina los datos de una rutina en la BBDD.

<i>El UA busca rutinas</i>	<i>El SGRE busca rutinas a partir de una cadena de caracteres de entrada y muestra los resultados obtenidos.</i>
<i>El UA descarga el paquete de rutina</i>	<i>El SGRE permite la descarga de la rutina correspondiente, creando el ".zip" previamente</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.2.4.2 Rutinas Públicas de Otros Usuarios

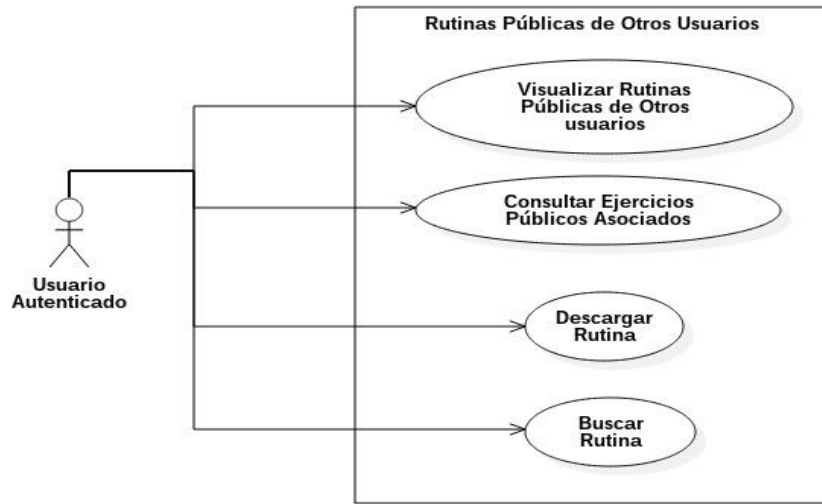


Figura 4.10: Diagrama de Casos de Uso de la Gestión de las Rutinas públicas de otros usuarios para un UA.

Caso de Uso	<i>Rutinas Públicas de Otros Usuarios</i>
Objetivo	<i>Realizar operaciones sobre las rutinas públicas de otros usuarios para el UA</i>
Actores	<i>Usuario Autenticado (UA)</i>
Disparador	<i>El UA hace clic sobre la pestaña Rutinas Públicas de Otros Usuarios</i>
Precondiciones	<i>El UA debe haberse autenticado correctamente en el SGRE</i>
Descripción	<i>El UA puede acceder a las rutinas públicas de otros usuarios, permitiendo la visualización o descarga. Además, se permite visualizar los ejercicios públicos asociados a una rutina</i>
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
<i>El UA visualiza las rutinas públicas de otros usuarios</i>	<i>El SGRE muestra los Ejercicios públicos de otros usuarios</i>
<i>El UA consulta los ejercicios asociados a una rutina</i>	<i>El SGRE muestra los ejercicios públicos asociados</i>
<i>El UA descarga el paquete de rutina</i>	<i>El SGRE permite la descarga de la rutina correspondiente</i>
<i>El UA busca rutinas</i>	<i>El SGRE busca rutinas a partir de una cadena de caracteres de entrada y muestra los resultados obtenidos.</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.2.5 Gestión de Ejercicios

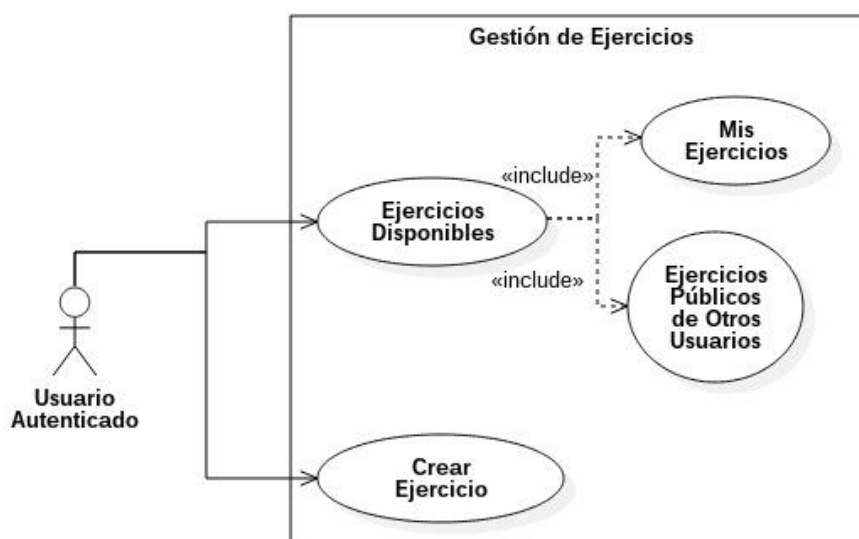


Figura 4.11: Diagrama de Casos de Uso de Gestión de Ejercicios.

Caso de Uso	<i>Gestión de Ejercicios para UA</i>
Objetivo	<i>Realizar operaciones sobre la información de los ejercicios tanto del propio usuario como de los ejercicios publicados por otros usuarios.</i>
Actores	<i>Usuario Autenticado (UA)</i>
Disparador	<i>El UA hace clic sobre la pestaña Ejercicios.</i>
Precondiciones	<i>El UA debe haberse autenticado correctamente en el SGRE</i>
Descripción	<i>El UA puede acceder crear un ejercicio, decidiendo si publicarlo o no. Además, puede visualizar los ejercicios propios o los públicos de otros usuarios.</i>
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
<i>El UA accede a los Ejercicios disponibles</i>	<i>El SGRE muestra, por defecto, los Ejercicios del propio usuario, dándose la opción de poder ver los ejercicios públicos de otros usuarios.</i>
<i>El UA accede a crear un nuevo Ejercicio</i>	<i>El SGRE inserta un nuevo Ejercicio en la Base de Datos.</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.2.5.1 Mis Ejercicios

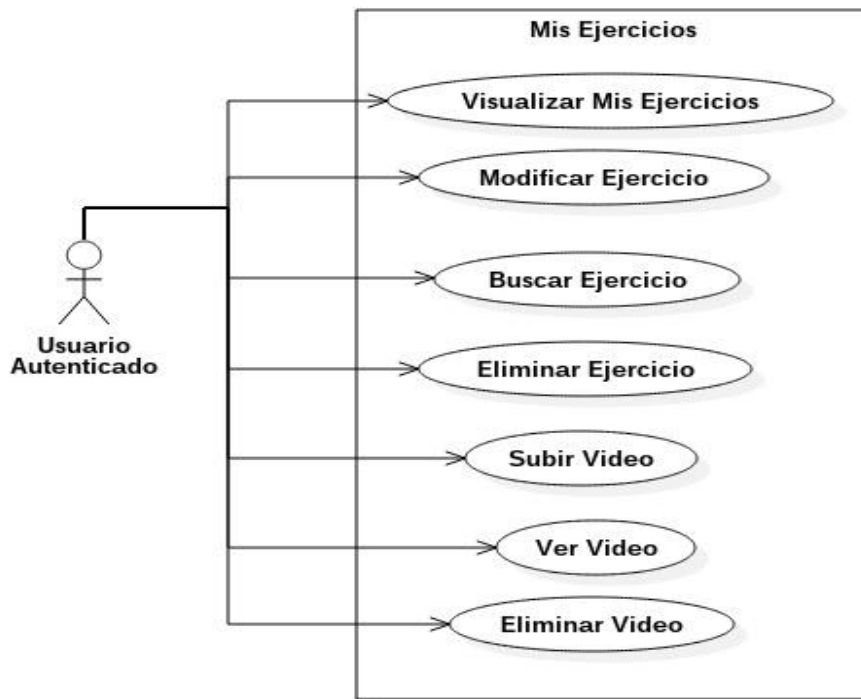


Figura 4.12: Diagrama de Casos de Uso de la Gestión de los Ejercicios propios de un UA.

Caso de Uso	Mis ejercicios
Objetivo	Realizar operaciones sobre los ejercicios del propio UA
Actores	Usuario Autenticado (UA)
Disparador	El UA hace clic sobre la pestaña Mis Ejercicios
Precondiciones	El UA debe haberse autenticado correctamente en el SGRE
Descripción	El UA puede acceder a visualizar, modificar, eliminar, subir o ver un Ejercicio/Vídeo cuyo propietario sea el UA
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
El UA visualiza sus ejercicios	El SGRE muestra los Ejercicios del propio usuario
El UA modifica sus ejercicios	El SGRE modifica los datos de un ejercicio en la BBDD
El UA elimina sus ejercicios	El SGRE elimina los datos de un ejercicio en la BBDD.
El UA Busca ejercicios	El SGRE busca ejercicios a partir de una cadena de caracteres de entrada y muestra los resultados obtenidos.
El UA sube vídeos	El SGRE permite la subida de un vídeo por ejercicio
El UA visualiza vídeos	El SGRE permite la visualización de los vídeos disponibles

<i>El UA elimina vídeos</i>	<i>El SGRE permite eliminar los vídeos que haya disponibles</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.2.5.2 Ejercicios Públicos de otros usuarios

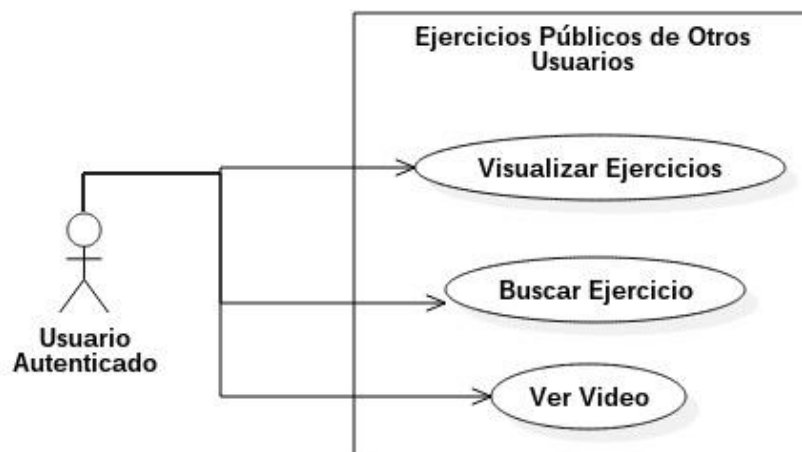


Figura 4.13: Diagrama de Casos de Uso de la Gestión de los Ejercicios públicos de otros usuarios para un UA.

Caso de Uso	<i>Ejercicios Públicos de Otros Usuarios</i>
Objetivo	<i>Realizar operaciones sobre los ejercicios públicos de otros usuarios para el UA</i>
Actores	<i>Usuario Autenticado (UA)</i>
Disparador	<i>El UA hace clic sobre la pestaña Ejercicios Públicos de Otros Usuarios</i>
Precondiciones	<i>El UA debe haberse autenticado correctamente en el SGRE</i>
Descripción	<i>El UA puede acceder a visualizar o ver aquellos ejercicios publicados por otros usuario y sus vídeos</i>
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
<i>El UA visualiza los ejercicios públicos de otros usuarios</i>	<i>El SGRE muestra los Ejercicios públicos de otros usuarios</i>
<i>El UA visualiza los vídeos de los ejercicios públicos de otros usuarios</i>	<i>El SGRE muestra los vídeos que estén disponibles de los ejercicios públicos de otros usuarios</i>
<i>El UA busca un ejercicio público de otro usuario</i>	<i>El SGRE busca ejercicios a partir de una cadena de caracteres de entrada y muestra los resultados obtenidos.</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.2.6 Ajustes

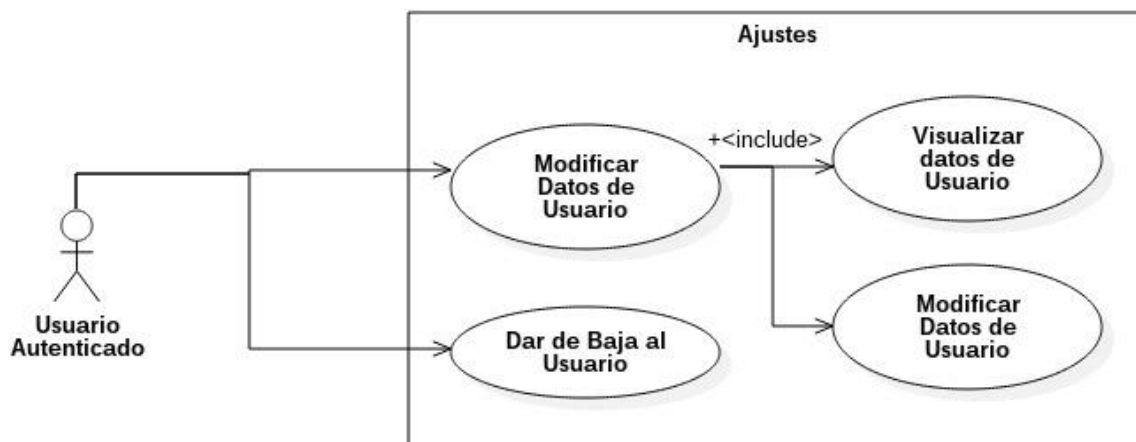


Figura 4.14: Diagrama de Casos de Uso de Ajustes de la cuenta de UA.

Caso de Uso	<i>Ajustes de Perfil para UA</i>
Objetivo	<i>Dos opciones: Modificar datos del UA o Dar de Baja al UA</i>
Actores	<i>Usuario Autenticado (UA)</i>
Disparador	<i>El UA hace clic sobre la pestaña Ajustes.</i>
Precondiciones	<i>El UA debe haberse autenticado correctamente en el SGRE</i>
Descripción	<i>El UA puede acceder a modificar el perfil de usuario o darse de baja en el servicio</i>
Curso Normal de los Eventos	
Acción de los Actores	Respuesta del Sistema
<i>El UA Modifica los datos del Perfil de Usuario</i>	<i>El SGRE muestra por defecto los datos de Usuario, y modifica los datos de usuario en la Base de Datos</i>
<i>El UA da de Baja al Usuario en el SGRE</i>	<i>El SGRE elimina todos los datos relacionados con el usuario de la Base de datos</i>
Cursos Alternativos	
<i>Ninguno.</i>	

4.3 Diagramas de Secuencia

En los siguientes diagramas podemos ver la interacción que se produce entre los distintos elementos de la aplicación Web para su funcionamiento. Se presentarán varios diagramas que agrupan toda la funcionalidad de la aplicación en cuatro acciones coincidentes con las operaciones CRUD de un servicio WebREST: Obtención de datos, Inserción de datos, Modificación de datos y Eliminación de datos.

Además, se presentará un último diagrama donde se detallarán las operaciones realizadas cuando un usuario descarga una rutina.

4.3.1 Identificación de elementos en los diagramas

Los diagramas mostrados a continuación disponen de 5 elementos principales:

- ✓ **Vista Web:** Es la apariencia que el usuario ve desde su navegador y mediante la cual interactúa con el servicio Web. Esta es generada por el controlador Web a partir del modelo definido, atendiendo al patrón MVC.
- ✓ **Modelo Web:** Es el modelo HTML sobre el que se representan los datos obtenidos mediante el controlador Web, o los datos insertados por el usuario.
- ✓ **Controlador Web:** Agrupa el conjunto de librerías y funciones en JavaScript que permiten la comunicación del modelo Web con el servicio Web para gestionar así los datos y ajustarlos al modelo para entregar la vista final al usuario.
- ✓ **WS (Web Service):** Es la aplicación WebREST desarrollada con Java Spring Framework, con la implementación de todas las funcionalidades y operaciones que requiere la aplicación Web.
- ✓ **BBDD:** Es la base de datos, la cual permite el almacenamiento de los datos que gestiona la aplicación web, consiguiendo así la persistencia de los mismos. Estos datos son la información de usuario, ejercicios, rutinas y vídeos.

4.3.2 Inicio de Sesión

El siguiente diagrama muestra la secuencia realizada para la autenticación de un usuario en su inicio de sesión:

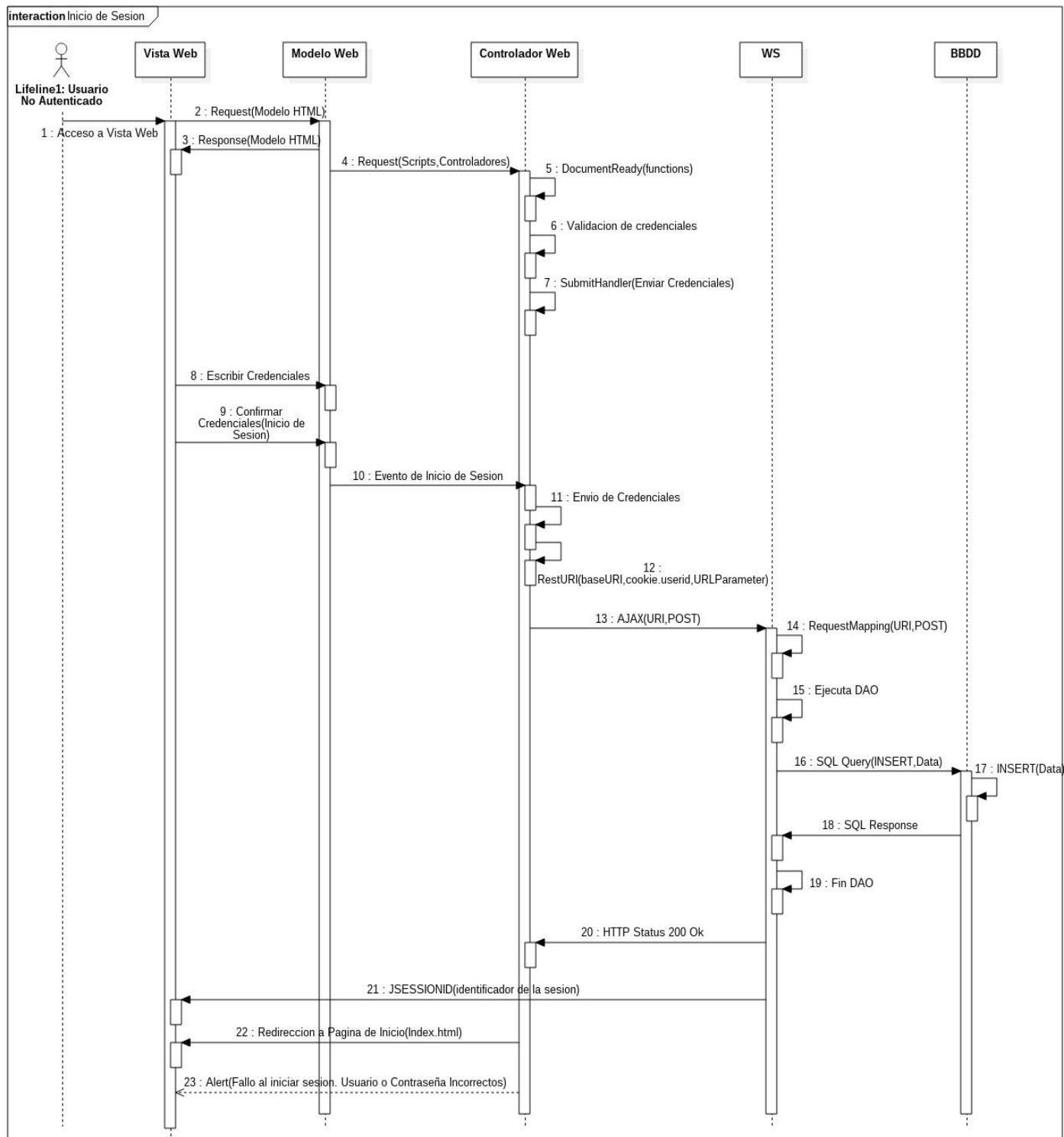


Figura 4.15: Diagrama de Secuencia del Inicio de Sesión.

4.3.3 Obtención de datos

El siguiente diagrama muestra de forma genérica la secuencia seguida para la obtención de datos en la aplicación Web. Esta secuencia es válida para la obtención de datos de usuarios, rutinas, ejercicios y vídeos, ya que lo único que varía entre ellos estaría identificado en el servidor como un recurso Web, el cual se identifica por su URI REST.

Además, cabe destacar que la aplicación Web obtiene datos en función de la privacidad de los mismos, es decir, permite pedir unos recursos u otros en función de los ejercicios o rutinas que se quieran obtener. Esto es posible porque se busca que los usuarios puedan decidir compartir tanto sus rutinas como sus ejercicios y vídeos. De esta forma, los usuarios autenticados podrán obtener datos tanto de ejercicios privados como de ejercicios públicos de otros usuarios, indicándose esta mediante una variable de control presente desde la vista Web hasta el propio servicio Web. En función de si los datos que se desean obtener son públicos o privados, el servicio web, realizaría una consulta diferente a la BBDD, devolviendo los datos acordes a la petición.

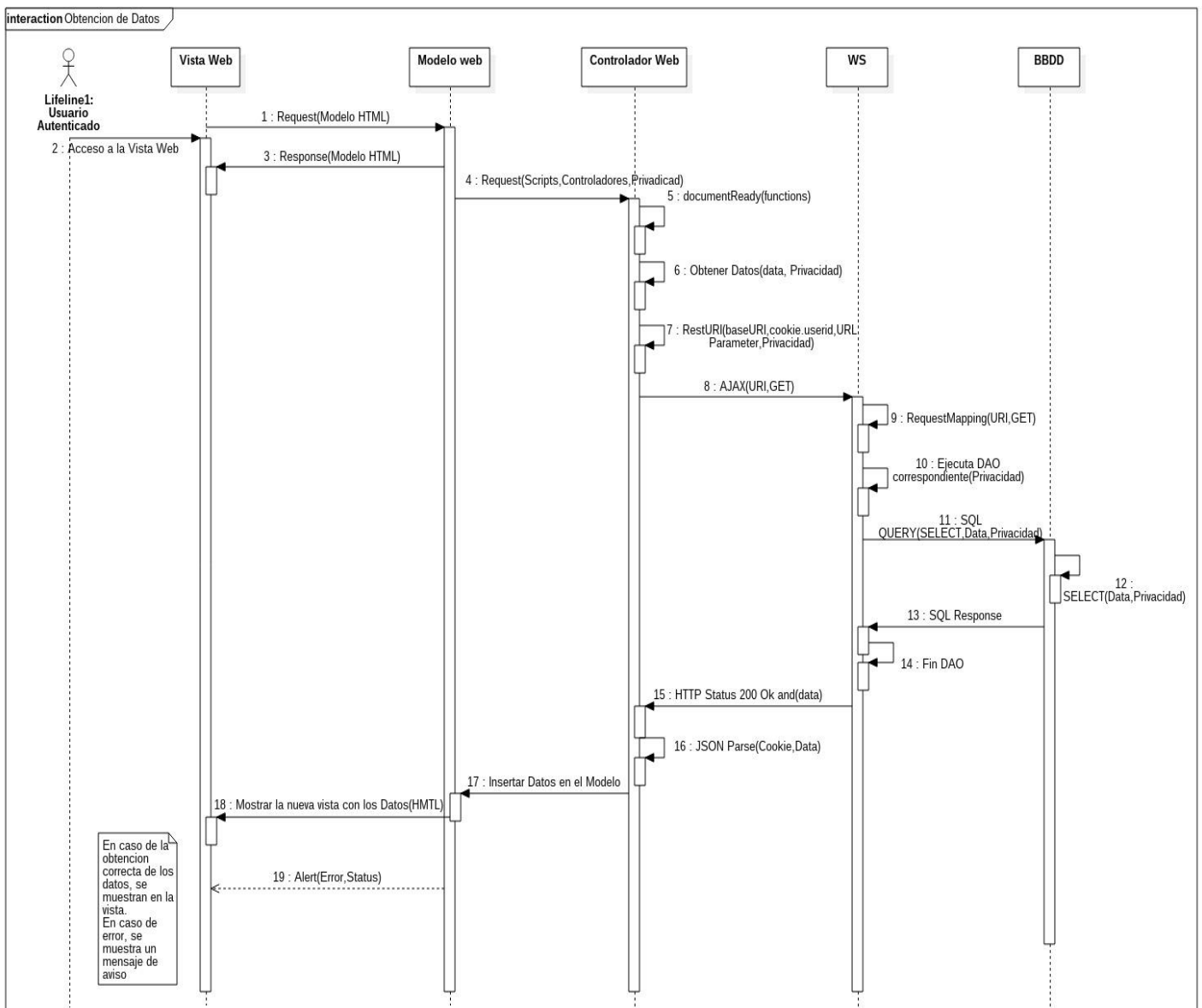


Figura 4.16: Diagrama de Secuencia de la obtención de datos.

4.3.4 Inserción de datos

El siguiente diagrama muestra de forma genérica la secuencia seguida para la inserción de datos en la aplicación Web. Esta secuencia es válida para la inserción de datos de usuarios, rutinas, ejercicios y vídeos, ya que lo único que varía entre ellos estaría identificado en el servidor como un recurso Web, el cual se identifica por su URI REST.

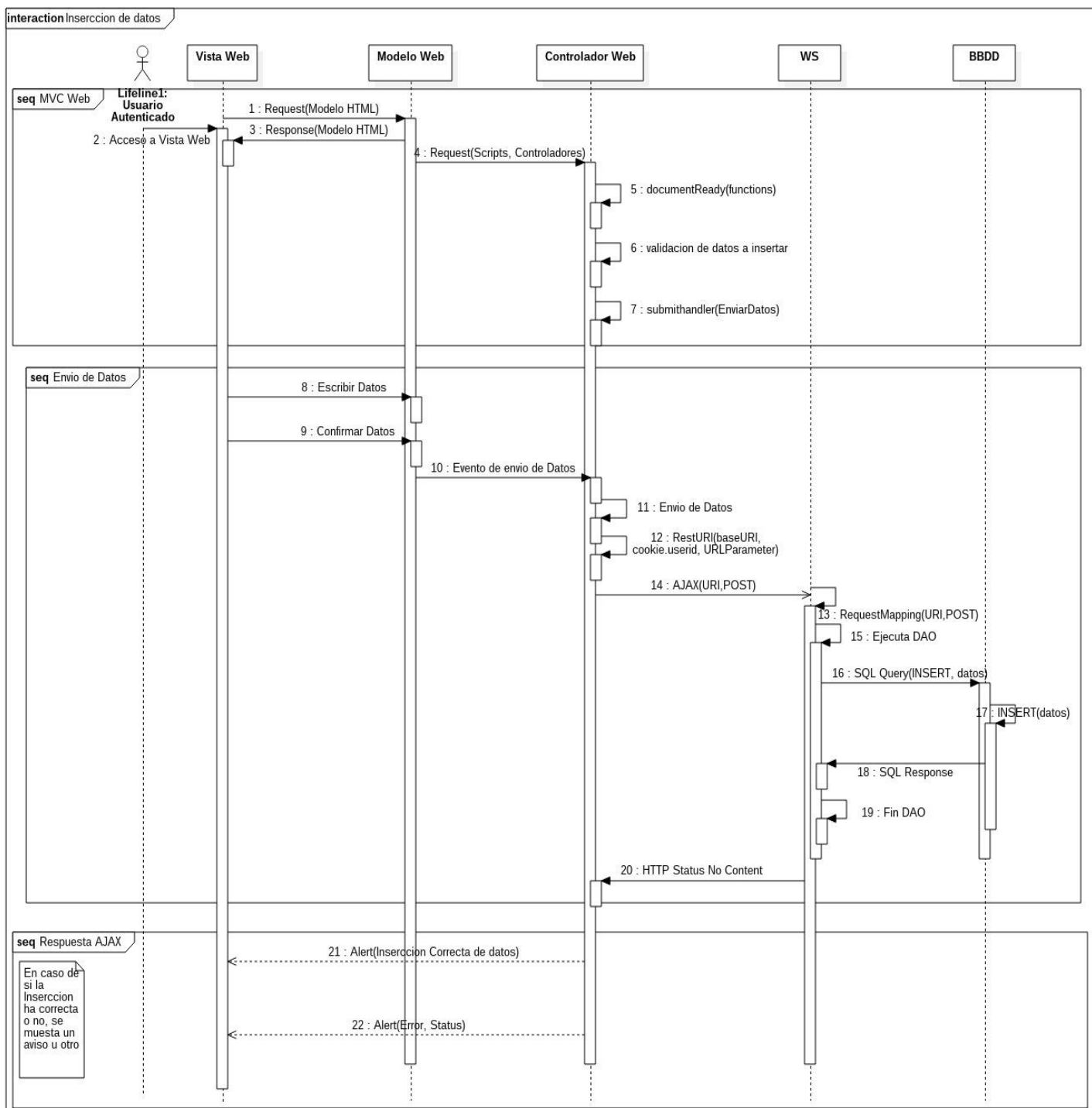


Figura 4.17: Diagrama de Secuencia de la inserción de datos

4.3.5 Modificación de Datos

Para la modificación de datos de la aplicación Web, la secuencia a seguir sería, primero la obtención de los datos para mostrarlos al usuario para que de esta forma pueda saber qué dato quiere modificar, y posteriormente la inserción de los nuevos datos. Por tanto, sería la secuencia seguida por los dos diagramas anteriores, cambiando tanto las URI para acceder a los recursos, como los DAO de los objetos en el Servicio Web en función de los datos que queremos modificar.

4.3.6 Eliminación de Datos

Para la eliminación de datos de la aplicación Web, la secuencia a seguir sería, primero la obtención de los datos para identificar cuál de ellos se quiere eliminar, y posteriormente la eliminación de dichos datos. Esta secuencia es válida para la eliminación de datos de usuarios, rutinas, ejercicios y vídeos, ya que lo único que varía entre ellos estaría identificado en el servidor como un recurso Web, el cual se identifica por su URI REST.

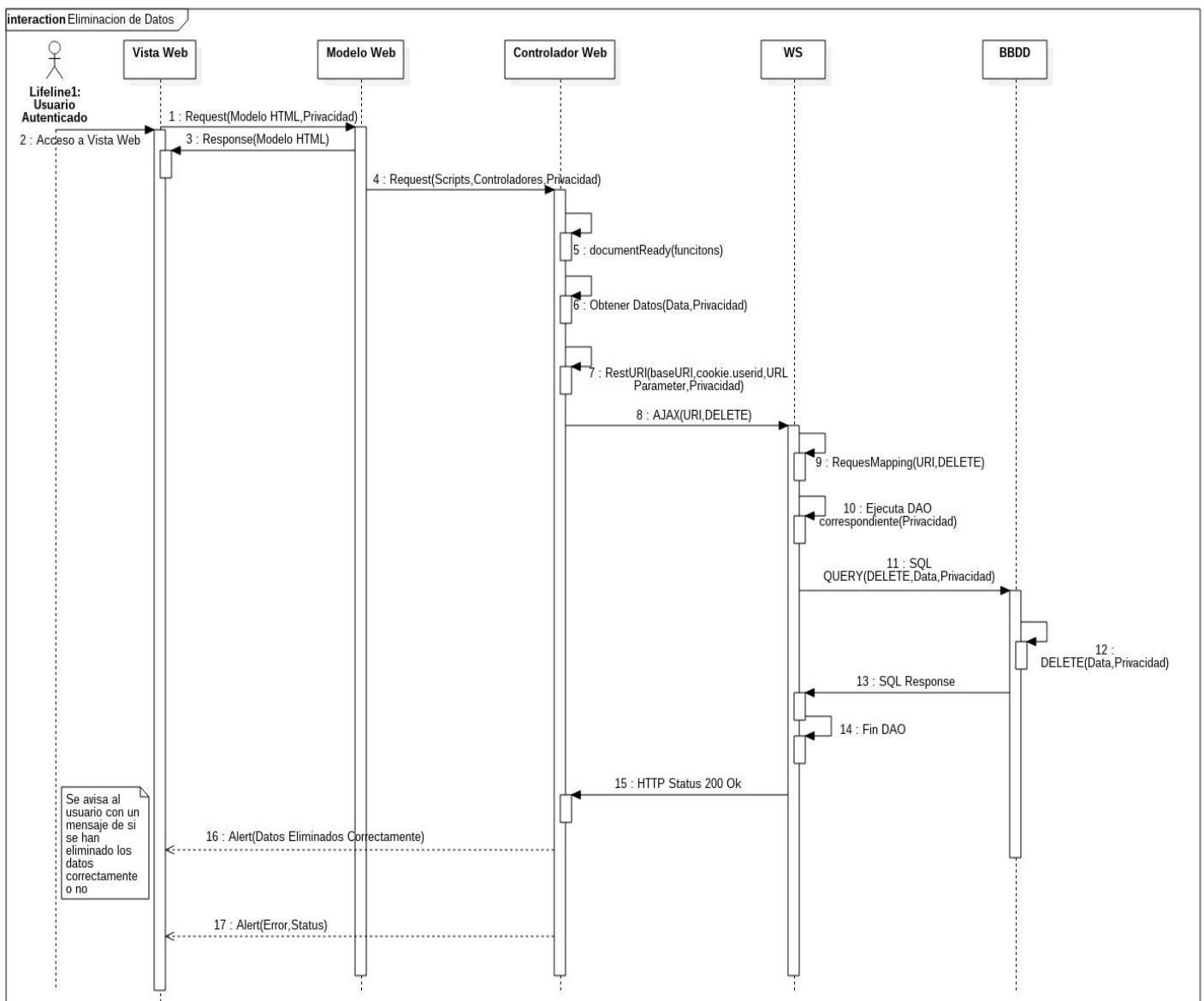


Figura 4.18: Diagrama de Secuencia de la eliminación de datos.

4.3.7 Descarga de Rutina

El siguiente diagrama muestra de forma genérica la secuencia seguida para la descarga de una rutina desde la aplicación Web. En esta secuencia se detallarán tanto la descarga como las operaciones realizadas para preparar el paquete de la rutina que se va a descargar el usuario, y que son transparentes para el mismo, es decir, el “.Zip” que recibirá la aplicación Android de Rutinas de entrenamiento “Rutina App”.

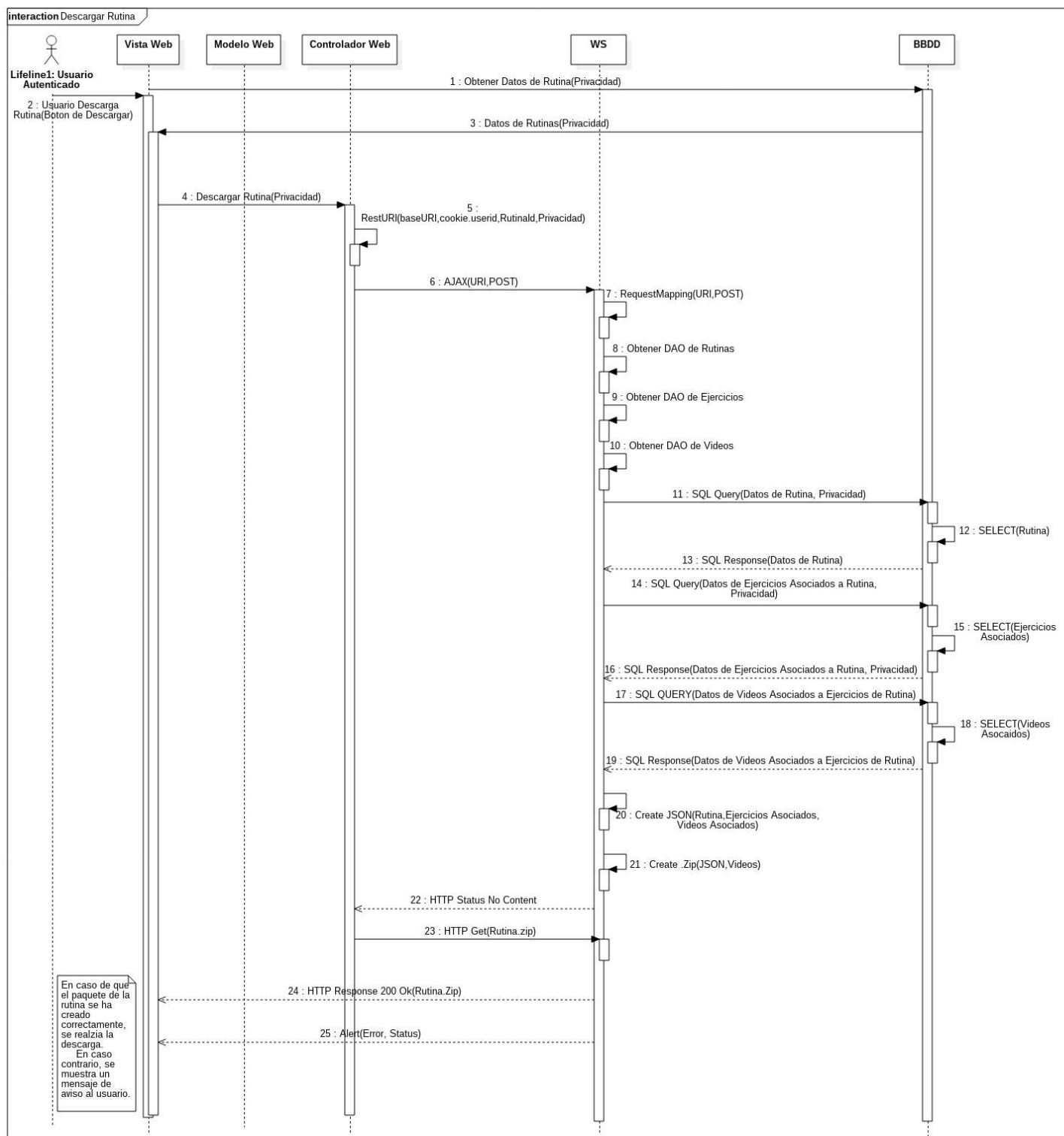


Figura 4.19: Diagrama de Secuencia de la Descarga de una Rutina.

4.4 Diagrama EERR de la BBDD

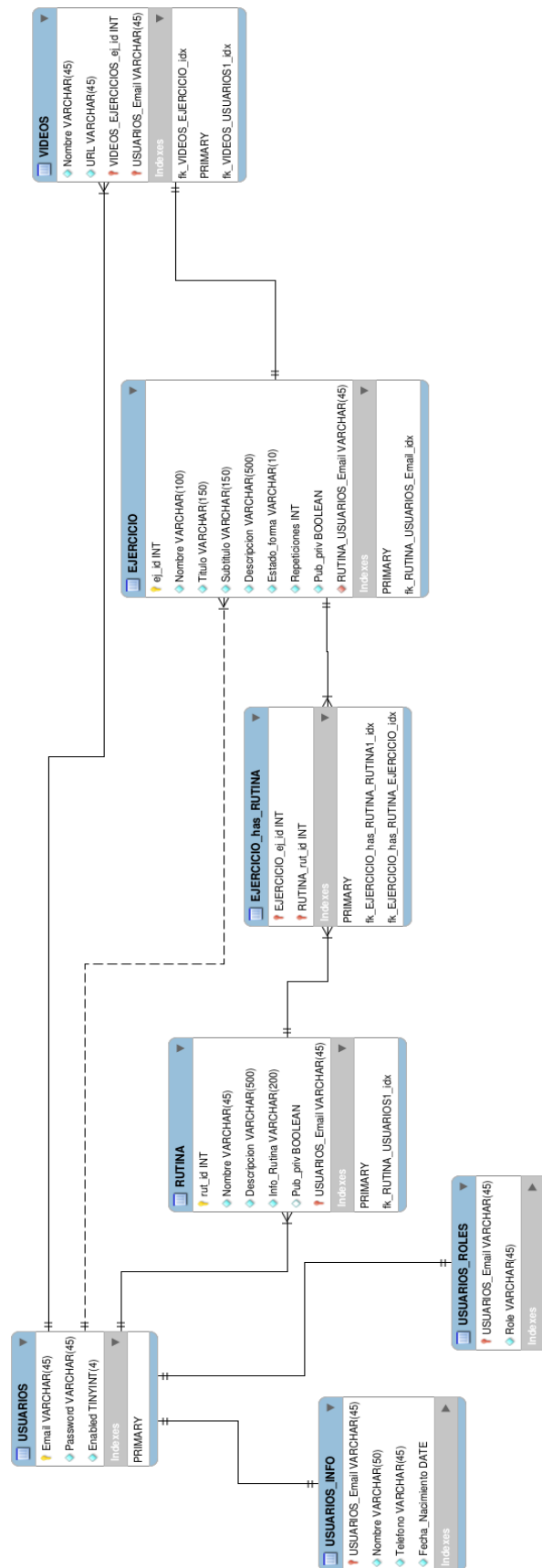


Figura 4.20: Diagrama EERR de la Base de Datos, para la Gestión de Rutinas de Entrenamiento.

4.4.1 Detalle de los elementos del diagrama EERR

4.4.1.1 Tabla USUARIOS

Descripción	En esta tabla, cada fila representa la información de acceso al servicio de un usuario que se ha registrado en el sistema. El usuario, puede añadir rutinas y ejercicios, y puede personalizar sus rutinas asociándole los ejercicios que desee entre los que tenga disponibles.
Email	<i>Es el correo electrónico del usuario registrado en la plataforma, el cual actúa como clave primaria de los datos del usuario en la base de datos, y de todo el sistema en sí, ya que el resto de tablas poseen un campo de este tipo como clave externa con este valor. No puede tener valor nulo.</i>
Password	<i>Es la contraseña que usan los usuarios para el acceso a la plataforma con su usuario. No puede tener valor nulo.</i>
Enabled	<i>Es un valor booleano que indica si el usuario está activo o no en el sistema. Lo usa Spring Security para su sistema de autenticación basado en formularios. No puede tener valor nulo.</i>

4.4.1.2 Tabla USUARIOS_INFO

Descripción	Es una tabla en la que cada fila se representa la información general sobre los usuarios registrados.
USUARIOS_Email	<i>Es la clave externa que apunta al campo Email de la tabla USUARIOS. No puede tener valor nulo.</i>
Nombre	<i>Este campo representa el nombre completo de un usuario. No puede tener valor nulo.</i>
Teléfono	<i>Este campo representa el número de teléfono de un usuario. No es un campo obligatorio.</i>
Fecha_Nacimiento	<i>Este campo representa la fecha de nacimiento de un usuario. No puede tener valor nulo.</i>

4.4.1.3 Tabla USUARIOS_ROLES

Descripción	En esta tabla, cada fila representa la información sobre los roles de un usuario en el sistema. Su existencia está marcada por el uso de Spring Security.
USUARIOS_Email	<i>Es la clave externa que apunta al campo Email de la tabla USUARIOS. No puede tener valor nulo.</i>
Role	<i>Para un propietario, es su ROL cuando se autentica en la aplicación Web. Lo usa Spring Security para su sistema de autorizaciones. No puede tener valor nulo.</i>

4.4.1.4 Tabla RUTINA

Descripción	En esta tabla, cada fila representa una rutina de entrenamiento con toda la información asociada a la misma. A una rutina puede asociarse diversos Ejercicios.
rut_id	<i>Este es el identificador único de una rutina de entrenamiento en la base de datos. Es clave primaria de esta tabla y es un entero autoincremental. No puede tener valor nulo.</i>
Nombre	<i>Es el nombre completo de una rutina. No puede tener valor nulo.</i>
Descripción	<i>Es la descripción completa de una rutina. No puede tener valor nulo.</i>
Info_Rutina	<i>Este campo permite tener información adicional de una rutina. No puede tener valor nulo.</i>
Pub_priv	<i>Este campo de valor booleano, permite controlar que una rutina pueda ser pública o privada. De esta forma el usuario decide la privacidad de sus rutinas de entrenamiento pudiendo compartirla con otros usuarios o no. No puede tener valor nulo.</i>
USUARIOS_Email	<i>Es la clave externa que apunta al campo Email de la tabla USUARIOS. No puede tener valor nulo.</i>

4.4.1.5 Tabla EJERCICIO

Descripción	En esta tabla, cada fila representa un ejercicio con toda la información asociada al mismo. Un ejercicio puede estar asociado a varias rutinas.
ej_id	<i>Este es el identificador único de un ejercicio en la base de datos. Es clave primaria de esta tabla y es un entero autoincremental. No puede tener valor nulo.</i>
Nombre	<i>Es el nombre completo de un ejercicio. No puede tener valor nulo.</i>
Título	<i>Es el título de un ejercicio. No puede tener valor nulo.</i>
Subtítulo	<i>Es el subtítulo de un ejercicio. No puede tener valor nulo.</i>
Descripción	<i>Es la descripción completa de un ejercicio. No puede tener valor nulo.</i>
Estado_forma	<i>Este campo permite establecer el estado de forma que se requiere para la realización de un ejercicio determinado. Por defecto, los estados de forma disponibles son Bajo, Medio y Alto. No puede tener valor nulo.</i>
Repeticiones	<i>Este campo es un entero que permite indicar el número de repeticiones requeridas para la realización de un ejercicio. No puede tener valor nulo.</i>
Pub_priv	<i>Este campo de valor booleano, permite controlar que un ejercicio pueda ser público o privado. De esta forma el usuario decide la privacidad de sus ejercicios, pudiendo compartirlos con otros usuarios o no. No puede tener valor nulo.</i>
RUTINA_USUARIOS_Email	<i>Es la clave externa que apunta al campo Email de la tabla USUARIOS. No puede tener valor nulo.</i>

4.4.1.6 Tabla EJERCICIO_has_RUTINA

Descripción	Esta es una tabla intermedia que permite la relación existente entre ejercicios y rutinas. Esta relación es de “Many to Many”, ya que una rutina puede contener múltiples ejercicios y un ejercicio puede estar contenido en múltiples rutinas.
EJERCICIO_ej_id	Clave externa que apunta al campo ej_id de la tabla EJERCICIO. No puede tener valor nulo.
RUTINA_rut_id	Clave externa que apunta al campo rut_id de la tabla RUTINA. No puede tener valor nulo.
USUARIOS_Email	<i>Es la clave externa que apunta al campo Email de la tabla USUARIOS. No puede tener valor nulo.</i>

4.4.1.7 Tabla VIDEO

Descripción	En esta tabla se almacena la información relativa a los vídeos que se asocian a los diferentes ejercicios.
Nombre	<i>Este campo almacena el nombre completo con el que se almacena el vídeo en el repositorio del servidor. Es usado para identificarlo de forma unívoca en dicho repositorio. No puede tener valor nulo.</i>
URL	<i>Este campo almacena la URL completa de un vídeo en el repositorio del servidor. Es usado para saber la ubicación absoluta del vídeo en el repositorio del servidor.</i>
VIDEOS_EJERCICIOS_ej_id	<i>Este campo es una clave externa que apunta al campo ej_id de la tabla EJERCICIO. No puede tener valor nulo.</i>
USUARIOS_Email	<i>Es la clave externa que apunta al campo Email de la tabla USUARIOS. No puede tener valor nulo.</i>

5 INTERFAZ DE USUARIO Y FUNCIONALIDAD

El diseño es el alma de todo lo creado por el hombre.

Steve Jobs.

En este capítulo veremos el diseño de la interfaz de usuario de nuestra aplicación Web, de qué elementos consta dicha interfaz y de qué manera están distribuidos los distintos elementos que la componen, además de especificar la funcionalidad que tiene cada uno de ellos. Cabe destacar que la interfaz gráfica es lo que percibe el usuario que hará uso de la aplicación Web, lo cual es lo más importante de cara a una aplicación como esta. Se busca por tanto originalidad y sencillez de uso para que esta aplicación pueda ser usada por usuarios sin experiencia previa.

5.1 Introducción

A continuación se intentará describir detalladamente la funcionalidad de la aplicación descrita en el Capítulo 1 de esta memoria. Los aspectos gráficos en los que nos centraremos a lo largo de este capítulo, los analizaremos mediante ejemplos de uso.

5.2 Autenticación o Inicio de Sesión

El usuario que hará uso de la aplicación Web, deberá acceder al mismo mediante un navegador Web, preferiblemente Google Chrome.

Una vez que el usuario accede al servicio, a través de su dirección Web (URL), se mostrará la página de inicio mostrada en la siguiente figura:

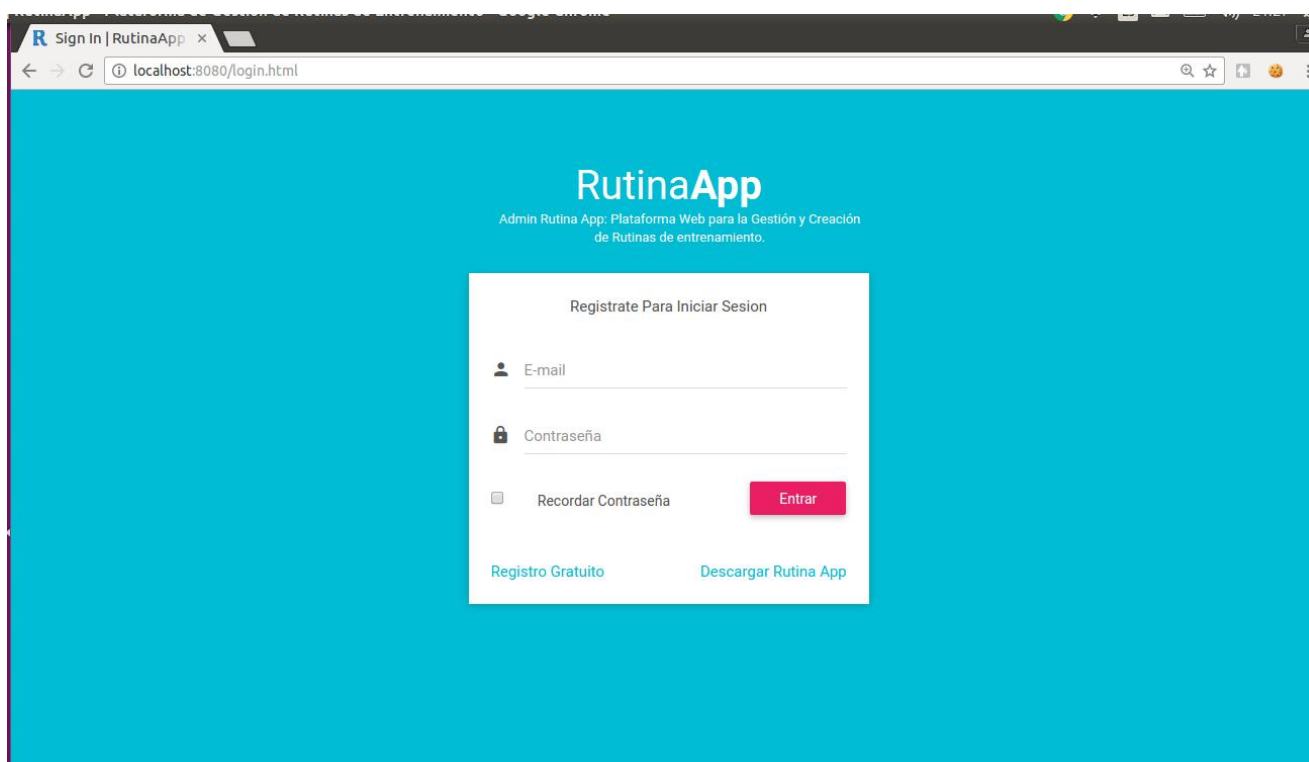


Figura 5.1: Página de Inicio de Sesión.

Para que un usuario previamente registrado pueda acceder a la aplicación, debe rellenar correctamente los campos del formulario de la Figura 5.1, en el cual realizan las comprobaciones oportunas antes de enviar los datos al Servicio Web. Además, se avisará al usuario si los datos introducidos son erróneos.

Por otro lado, se da también al usuario la opción de recordar su contraseña, la cual se almacenará en la cache del navegador para el próximo inicio de sesión.

A continuación, se mostrará un ejemplo de inicio de sesión con un usuario ya registrado, en el cual, primero se introducirán los datos correctamente y luego de forma errónea, mostrando esta casuística con varias ilustraciones:

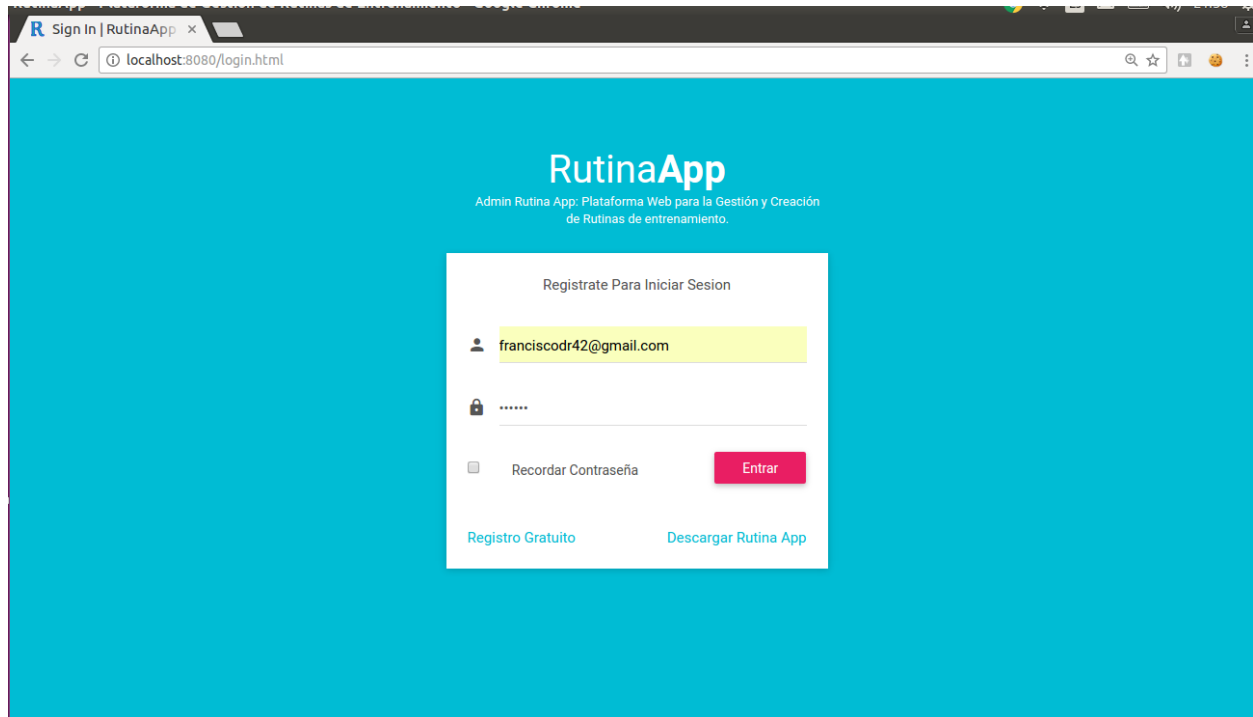


Figura 5.2: Inicio de Sesión de Usuario ya Registrado.

Si el usuario introduce el Email o Contraseña de usuario de forma errónea, se muestra el siguiente mensaje de aviso:

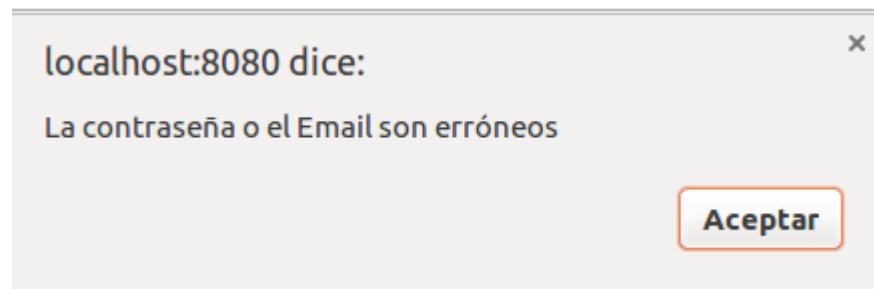


Figura 5.3: Mensaje de Aviso de Credenciales Introducidas Erróneas.

En caso de que los datos sean introducidos correctamente, se reenvía a la siguiente página de inicio, donde se ofrece un carrusel de imágenes de presentación de la aplicación Web, además de un menú que permite al usuario desplazarse por las principales opciones que ofrece la aplicación Web y la descarga de la aplicación móvil en Android.

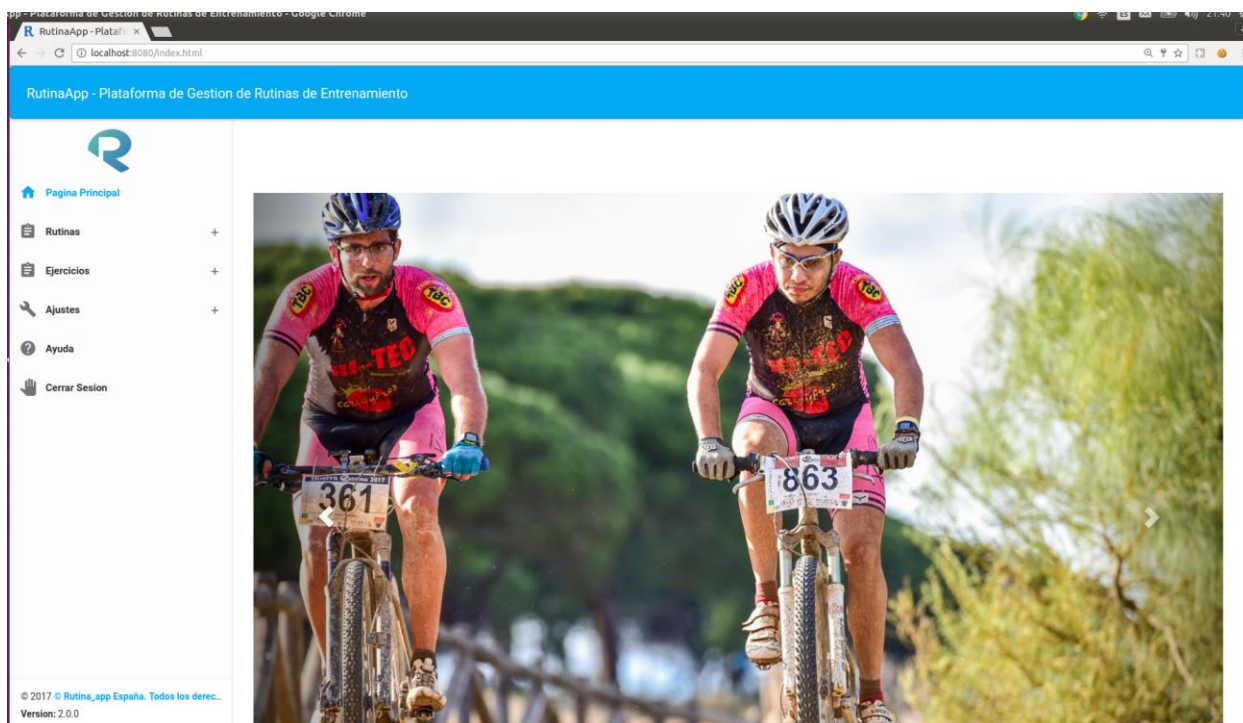


Figura 5.4: Página de Inicio de la aplicación Web tras la Autenticación correcta del Usuario.

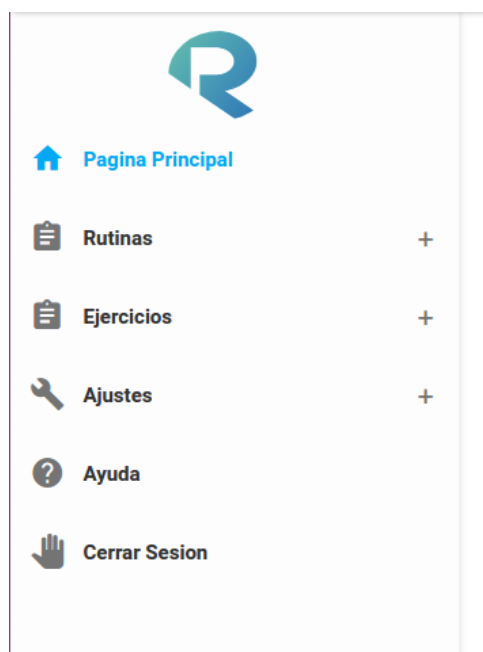


Figura 5.5: Menú desplegable de Navegación de Usuario.

5.3 Creación de una cuenta de usuario

Si el usuario selecciona la opción de “Registro Gratuito” en la página de inicio antes de autenticarse, se ofrecerá la posibilidad de registrarse como nuevo usuario en el sistema. Para ello, se presenta al usuario un formulario que deberá rellenar con sus datos de registro y una clave mediante la que accederá a la aplicación Web.

Una vez rellenos los datos, hay dos opciones posibles:

- **Confirmar:** Envía los datos introducidos y reenvía al usuario a la página principal de la aplicación (Figura 5.1). En el caso de que el registro tenga éxito o no, se informará al usuario con mensajes de aviso (Figura 5.7 y 5.8).
- **Cancelar:** Cancela la operación de registro y redirige al usuario a la pantalla de inicio (Figura 5.1).

A continuación, mostramos un ejemplo para la creación de una cuenta de usuario:

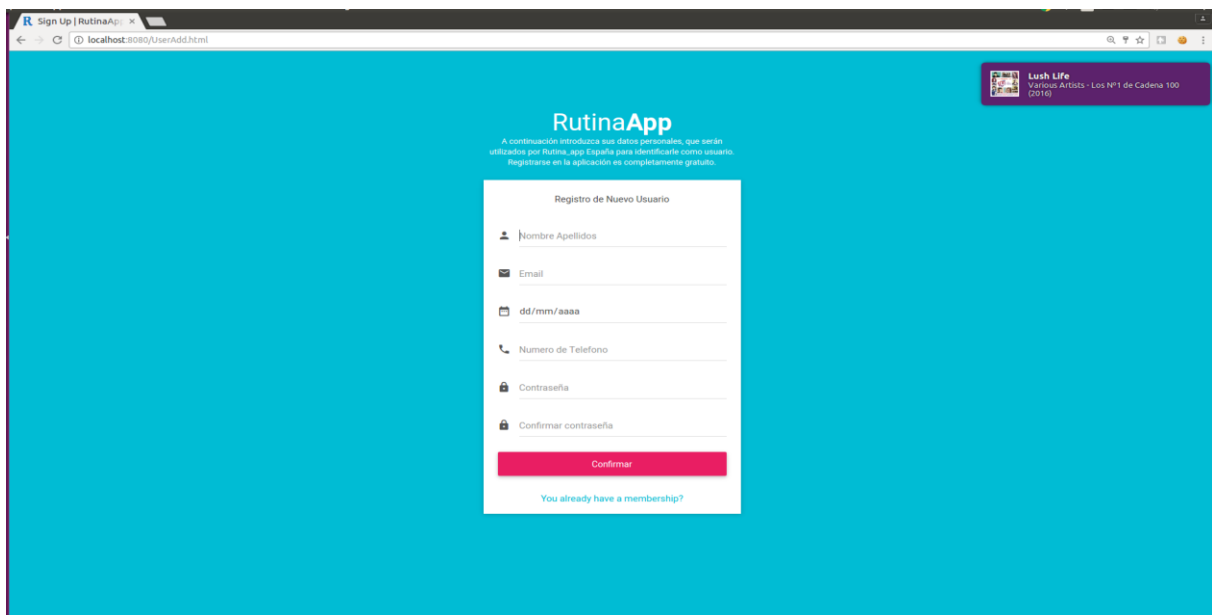


Figura 5.6: Página de Registro Gratuito de Usuario.

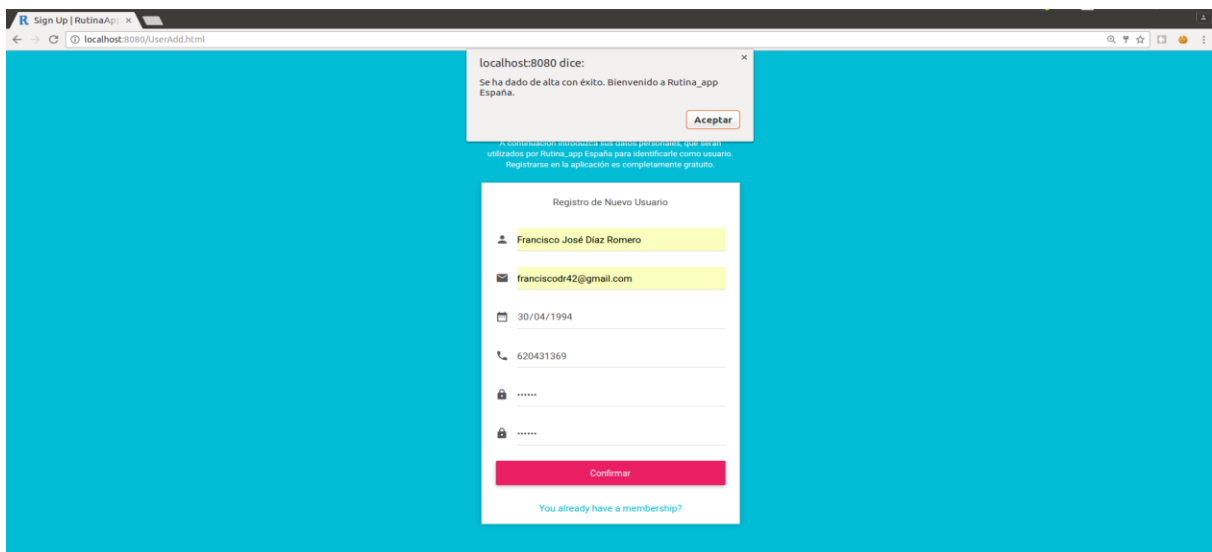


Figura 5.7: Registro Correcto de Nuevo Usuario.

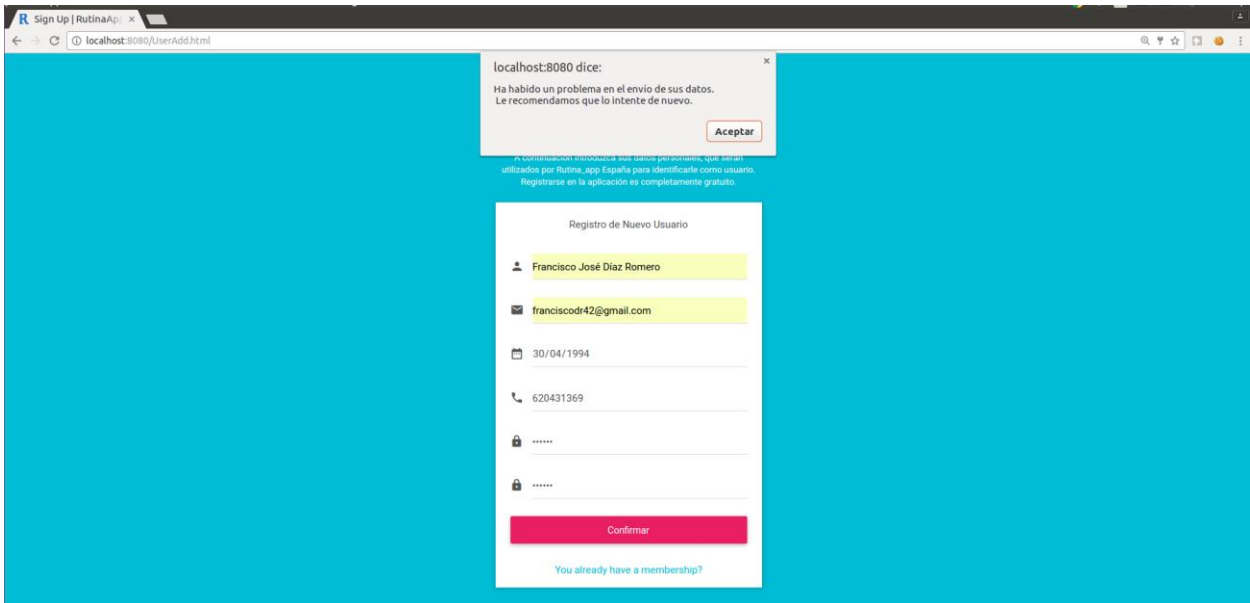


Figura 5.8: Registro erróneo de nuevo Usuario.

5.4 Gestión de Rutinas

En este apartado, explicaremos las opciones posibles por parte del usuario autenticado en el caso de que quiera gestionar Rutinas de entrenamiento, seleccionando las opciones disponibles en la pestaña de “Rutinas” del menú desplegable.

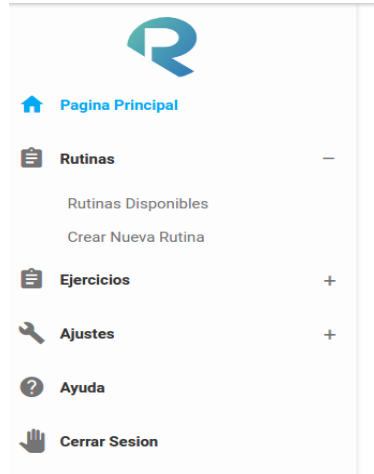


Figura 5.9: Opciones para la Gestión de Rutinas en el menú desplegable.

5.4.1 Crear Nueva Rutina

Si seleccionamos esta opción, se ofrecerá la posibilidad al usuario de crear una nueva rutina en la aplicación Web, permitiendo la posibilidad de publicarla o no. De esta forma, la rutina podrá ser visible para el resto de usuarios registrados en el sistema o sólo para el usuario que la ha creado.

Un formulario web para crear una nueva rutina. El título es 'Crear Nueva Rutina' y hay un subtítulo que indica que es obligatorio rellenar todos los campos correctamente. El formulario tiene cuatro campos de texto: 'Nombre de la Rutina' con el valor 'Circuito de Iniciación al Fitness', 'Descripción de la Rutina' con el texto 'Esta rutina de entrenamiento esta orientad a aquellas personas que deseen iniciarse en el mundo de la musculación y del Fitness.', 'Informacion Adicional de la Rutina:' con el texto 'Resaltar, que las pautas a seguir deben ser respetadas en todo momento para evitar posibles lesiones con repercusión en la vida diaria.', y 'Visibilidad:' con un menú desplegable que muestra 'Privada (Solo Podrá Consultar esta Rutina su Propietario)'. En la parte inferior hay dos botones: 'Confirmar' y 'Cancelar'.

Figura 5.10: Formulario de Registro de Nueva Rutina.

Una vez introducidos los datos de la rutina, si pulsamos el botón de confirmar, la rutina quedará registrada en la base de datos del sistema, lanzándose un mensaje de aviso, redirigiendo al usuario a la página de rutinas disponibles, donde se podrán visualizar las rutinas que hay disponibles.

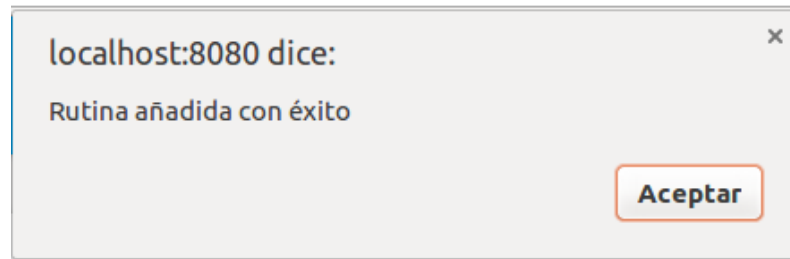


Figura 5.11: Mensaje de aviso de Registro de Rutina Correcto.

5.4.2 Rutinas Disponibles

Si seleccionamos la opción “Rutinas Disponibles”, por defecto se mostrarán las rutinas propias del usuario autenticado, tanto las públicas como las privadas, dándose la opción de filtrar la búsqueda de rutinas en todo momento.

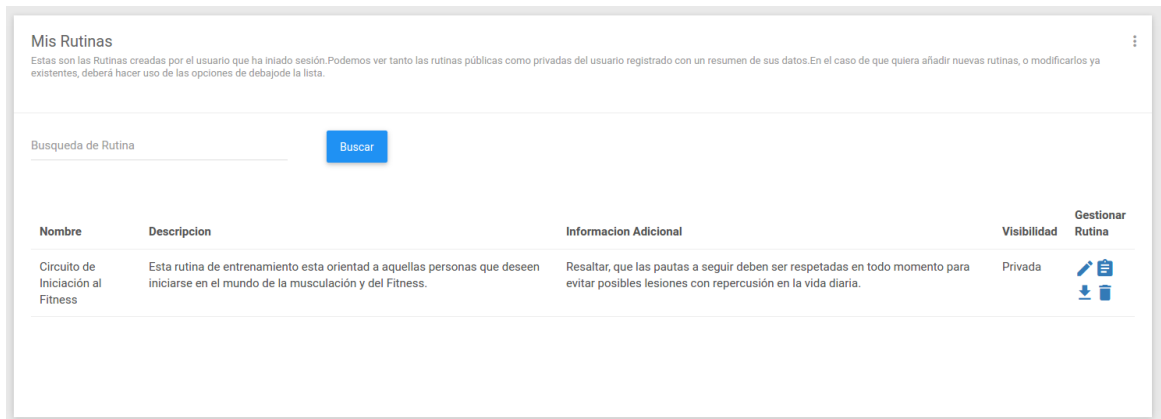


Figura 5.12: Lista de Rutinas Disponibles de un Usuario Autenticado.

Además, si pulsamos sobre el icono desplegable superior de la derecha (☰), se dará la opción de poder visualizar también las rutinas públicas de otros usuarios. Destacar, que sobre las distintas rutinas se tendrán unos privilegios u otros en función de su privacidad.

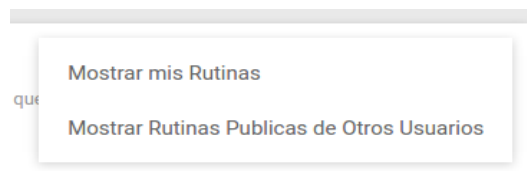


Figura 5.13: Menú desplegable para selección de Rutinas a Visualizar.

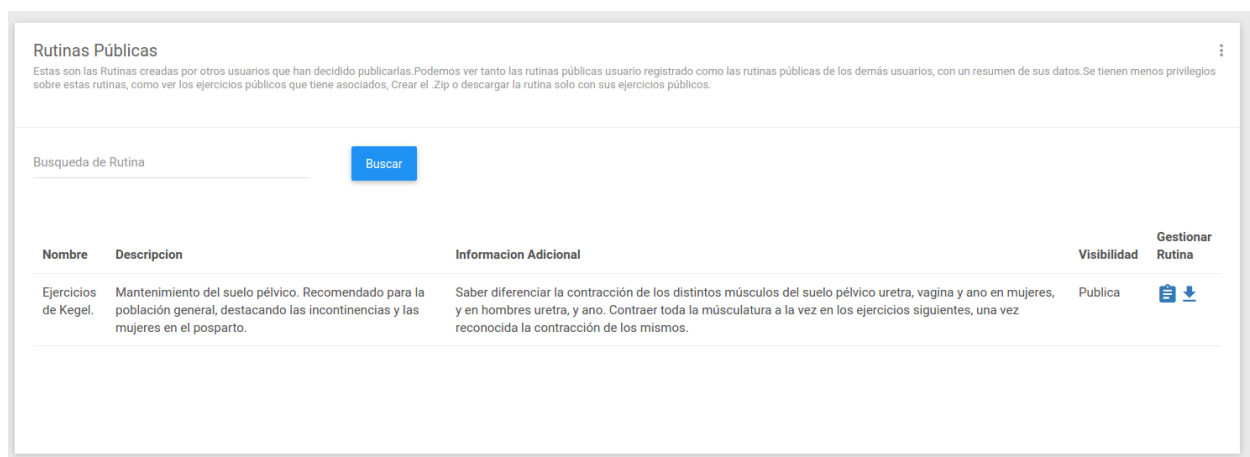


Figura 5.14: Lista de Rutinas Públicas de otros Usuarios Disponibles para un Usuario Autenticado.

Las siguientes acciones se podrán realizar sobre las rutinas disponibles en función de la privacidad de las mismas. Si la rutina es propia del usuario autenticado, se tendrán todas las acciones disponibles, pero si la rutina es pública de otro usuario, solo se permitirá visualizar los ejercicios asociados que sean públicos y la descarga de la misma.

5.4.2.1 Modificar Rutina

Esta acción permite al usuario propietario de la rutina modificar cualquier campo de la misma, modificándose en la base de datos del sistema, siempre que la modificación sea posible, y mostrando un mensaje de aviso.



Figura 5.15: Botón correspondiente a la Modificación de una Rutina.

Modificar Rutina

A continuación se muestran los datos actuales de la rutina que se desea modificar. Recuerde que el hecho de modificar rutinas es completamente gratuito, por lo que puede modificar todas las rutinas que desee sin ningún coste. Por otra parte, es obligatorio rellenar todos los campos de manera correcta, si no, no se completará la modificación de la rutina.

Nombre de la Rutina: Circuito de Iniciación al Fitness

Descripción de la Rutina: Esta rutina de entrenamiento esta orientada a aquellas personas que deseen iniciarse en el mundo de la musculación y del Fitness.

Información Adicional de la Rutina: Resaltar, que las pautas a seguir deben ser respetadas en todo momento para evitar posibles lesiones con repercusión en la vida diaria.

Visibilidad: Publica

Figura 5.16: Formulario para Modificación de Rutina.

5.4.2.2 Ejercicios Asociados

Si seleccionamos esta opción, se ofrecerá al usuario la posibilidad de visualizar los ejercicios asociados a una rutina, permitiendo la posibilidad de eliminar la asociación de ese ejercicio a la rutina seleccionada.

Ejercicios Asociados a Rutinas

Estos son los Ejercicios que tiene Asociados una Rutina, con un resumen de sus datos. En el caso de que quiera asociar nuevos ejercicios, o modificar los ya existentes, deberá hacer uso de las opciones de debajo de la lista.

Nombre	Título	Subtítulo	Descripción	Estado de Forma	Repeticiones	Visibilidad	Gestionar Ejercicios
Pecho	Ejercicio para Pechos	Maquina 11	Este ejercicio permite el fortalecimiento de la parte superior del pectoral. Su realización consisten en extensiones continuas de los brazos de forma que el peso recae sobre el pectoral ejercitandolo.	Bajo	15	Privada	<input type="button" value="ELIMINAR"/>
Biceps	Ejercicio de Biceps	Maquina 2	Ejercicio esencial para fortalecer el biceps, consintiendo en la contracción y extensión de los brazos. Precaución a la hora de realizar este ejercicio para evitar lesiones en las articulaciones, como los codos por ejemplo.	Bajo	15	Publica	<input type="button" value="ELIMINAR"/>
Triceps	Ejercicio de Triceps	Maquina 7	Este ejercicio permite el fortalecimiento de la zona anterior del brazo, es decir, permite ejercitar el triceps mediante el estiramiento inverso de los brazos.	Bajo	15	Privada	<input type="button" value="ELIMINAR"/>
Dorsales	Ejercicio de Dorsal Inferior	Maquina 9	Este ejercicio es uno de los mas completos y esenciales a la hora de ejercitar el dorsal, simulando lo conocido como el remo.	Bajo	15	Publica	<input type="button" value="ELIMINAR"/>
Gemelos	Ejercicio de Gemelos	Maquina 4	Elevaciones de talón sentado normales: Busca la máquina específica para hacer este ejercicio y céntrate más en seguir un rango de recorrido completo y no tanto en usar un peso alto. En caso de que no la tengas existen opciones alternativas como hacerlo con mancuernas sobre las rodillas, aunque personalmente me parecen muy incómodas, por lo que trataría de buscar ejercicios alternativos.	Bajo	15	Publica	<input type="button" value="ELIMINAR"/>

Figura 5.17: Lista de Ejercicios Asociados a una Rutina.

Además, se ofrecerá la posibilidad al usuario de asociar ejercicios a una rutina de los que haya disponibles. Los ejercicios que hay disponibles serán los propios del usuario autenticado y los públicos de otros usuarios, que no hayan sido asociados ya a la rutina seleccionada.

Asociación de Ejercicios

Estos son los Ejercicios disponibles para asociar a una rutina, con un resumen de sus datos. En el caso de que quiera añadir nuevos ejercicios, o modificar los ya existentes, deberá hacer uso de las opciones de debajo de la lista.

Busqueda de Rutina

Nombre	Título	Subtítulo	Descripción	Estado de Forma	Repeticiones	Visibilidad	Gestionar Ejercicios
Contracciones rápidas	2ª parte ejercicios de Kegell	Suelo pélvico	Hacer contracciones rápidas del suelo pélvico (aprox. 10 segundo) y relajar el doble de tiempo (20 segundos). Repetir 10 veces.	Bajo	10	Publica	<input type="button" value="ASOCIAR"/>
Calentamiento Previo	Cardio de calentamiento	15 minutos	Se requiere un calentamiento previo a la iniciación del circuito de ejercicios musculares mediante la realización de Cardio en cinta, elíptica o bicicleta estática.	Bajo	1	Publica	<input type="button" value="ASOCIAR"/>
Aductor	Ejercicio de Aductores	Maquina 5	La máquina de abductores/ductores es la que permite trabajar la musculatura separadora/aproximadora de la cadera. Puede que una misma máquina permita alterar la posición para trabajar estos dos movimientos, o bien que haya una máquina para abductores y otra para trabajar los abductores de cadera.	Bajo	15	Publica	<input type="button" value="ASOCIAR"/>
Cuadriceps	Ejercicio de Cuadriceps	Maquina 3	Una vez estemos sentado en la máquina debes ajustar los apoyos: el final del asiento debe coincidir con el hueco popliteo (detrás de la rodilla) y la zona lumbar en contacto con el respaldo, finalmente regula el rodillo del brazo de palanca colocándolo al final de la tibia, en la articulación del tobillo. Inicia el movimiento de extensión desde una flexión de La rodilla de 90°, no menos y realiza la extensión de la pierna hasta llegar de forma controlada a la extensión completa de la rodilla.	Bajo	15	Publica	<input type="button" value="ASOCIAR"/>
Abdominales	Ejercicio de Abdominales	Maquina 14	Se puede encontrar en todos los gimnasios y es una de las máquinas para abdominales más utilizadas por lo fácil que resulta trabajar con ella. El peso que debes colocar irá en función de la fuerza con la que puedas "tirar" de los agarres. Una recomendación para usar esta máquina y hacer abdominales en general es que a la hora de ejecutar el ejercicio hay que apretar bien el abdomen, sobre todo, para no lastimar la espalda concentrando la tracción sobre ella.	Bajo	13	Publica	<input type="button" value="ASOCIAR"/>

Figura 5.18: Lista de Ejercicios Disponibles para Asociar a una Rutina.

Se mostrarán mensajes de aviso cuando un ejercicio haya sido asociado correctamente, y además, si un ejercicio que se desea asociar no tiene vídeo asociado, permite su asociación a la rutina pero avisando de la carencia de un vídeo para el mismo (Figura 5.19).

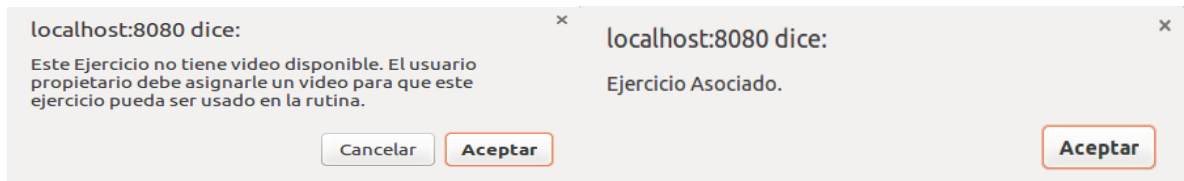


Figura 5.19: Mensajes de Aviso para Asociación de Ejercicios a una Rutina.

5.4.2.3 Descargar Rutina

Esta opción permite la descarga de una rutina en formato “.zip”, tanto del usuario que se haya autenticado como las rutinas públicas de otros usuarios. La rutina contendrá todo lo necesario para que la aplicación Android pueda interpretarla correctamente.

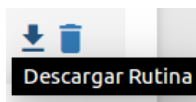


Figura 5.20: Botón correspondiente a la descarga de una Rutina.

Cabe destacar que, si la rutina tiene ejercicios asociados que no tienen vídeos disponibles y se intenta descargar la rutina, se mostrará un mensaje avisando de que todos aquellos ejercicios sin vídeos disponibles no se añadirán a la rutina, permitiendo la descarga de dicha rutina solo con los ejercicios que estén completados (Figura 5.21).

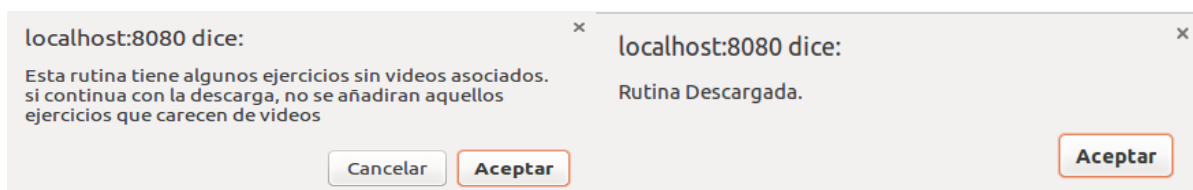


Figura 5.21: Mensajes de Aviso para la Descarga de una Rutina.

5.4.2.4 Eliminar Rutina

Esta opción permite eliminar una rutina de un usuario que se ha autenticado. Si la rutina no tiene ejercicios asociados y pulsamos dicha opción, se mostrará un mensaje de aviso indicándolo, requiriéndose la confirmación de esta acción, ya que si se confirma, la rutina será borrada junto con la asociación de los ejercicios que tenía.

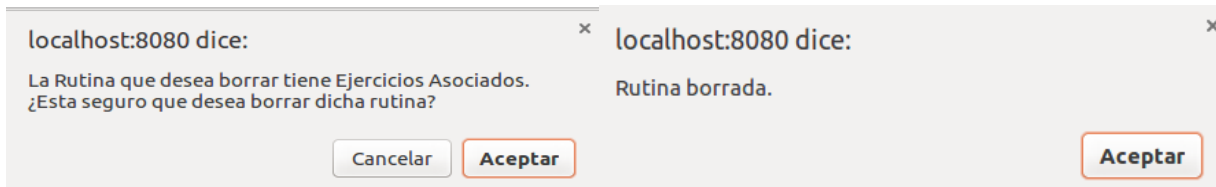


Figura 5.22: Mensajes de Aviso para la Eliminación de una Rutina.

5.5 Gestión de Ejercicios

En este apartado, explicaremos las opciones posibles por parte del usuario autenticado en el caso de que quiera gestionar Ejercicios, seleccionando las opciones disponibles en la pestaña de “Ejercicios” del menú desplegable.

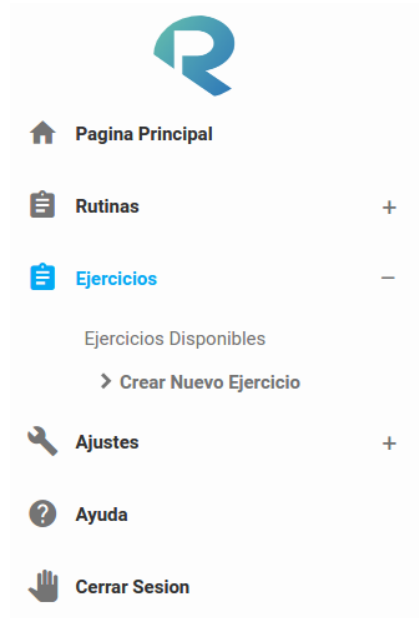


Figura 5.23: Opciones para la Gestión de Rutinas en el menú desplegable.

5.5.1 Crear Nuevo Ejercicio

Si seleccionamos esta opción, se ofrecerá la posibilidad al usuario de crear un nuevo ejercicio en la aplicación Web, permitiendo la posibilidad de publicarlo o no. De esta forma, el ejercicio podrá ser visible para el resto de usuarios registrados en el sistema o sólo para el usuario que lo ha creado.

Crear Nuevo Ejercicio

A continuación introduzca los de un ejercicio que desea registrar. Recuerde que el hecho de introducir ejercicios es completamente gratuito, por lo que puede añadir todos los ejercicios que desee sin ningún coste. Por otra parte, es obligatorio rellenar todos los campos de manera correcta, si no, no se registrará el evento.

Nombre del Ejercicio:

Título:

Subtítulo:

Descripción del Ejercicio:

Estado de Forma:

Repeticiones del Ejercicio:

Visibilidad:

Figura 5.24: Formulario de Registro de Nuevo Ejercicio.

Una vez introducidos los datos del ejercicio, si pulsamos el botón de confirmar, el ejercicio quedará registrado en la base de datos del sistema, lanzándose un mensaje de aviso, redirigiendo al usuario a la página de ejercicios disponibles, donde se podrán visualizar los ejercicios que hay disponibles.

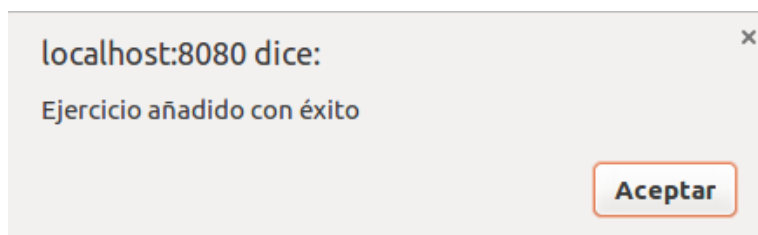
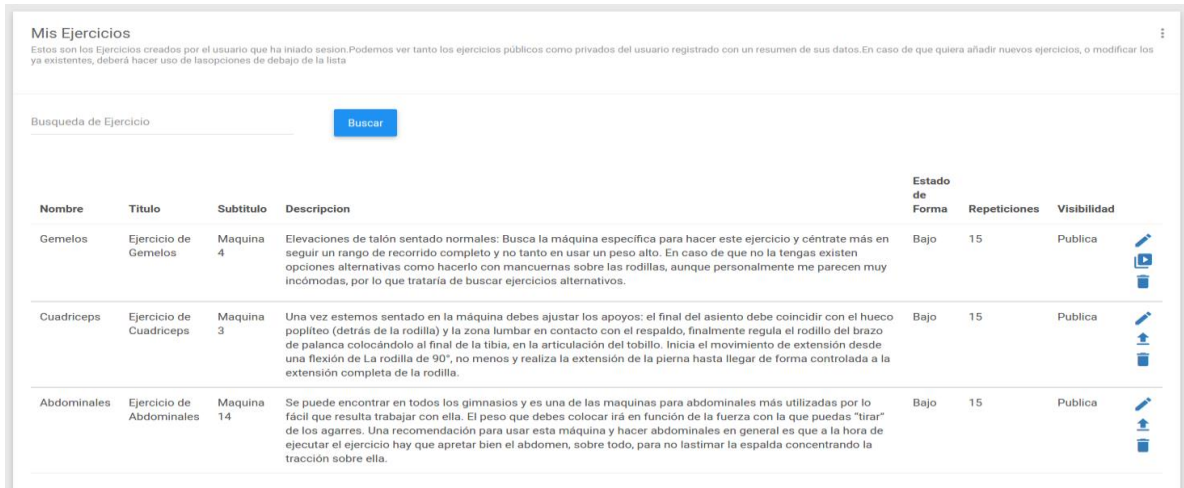


Figura 5.25: Mensaje de aviso de Registro de Ejercicio Correcto.

5.5.2 Ejercicios Disponibles

Si seleccionamos la opción “Ejercicios Disponibles”, por defecto se mostrarán los ejercicios propios del usuario autenticado, tanto los públicos como los privados, dándose la opción de filtrar la búsqueda de ejercicios.



Nombre	Título	Subtítulo	Descripción	Estado de Forma	Repeticiones	Visibilidad
Gemelos	Ejercicio de Gemelos	Maquina 4	Elevaciones de talón sentado normales: Busca la máquina específica para hacer este ejercicio y céntrate más en seguir un rango de recorrido completo y no tanto en usar un peso alto. En caso de que no la tengas existen opciones alternativas como hacerlo con mancuernas sobre las rodillas, aunque personalmente me parecen muy incómodas, por lo que trataría de buscar ejercicios alternativos.	Bajo	15	Publica
Cuadriceps	Ejercicio de Cuadriceps	Maquina 3	Una vez estemos sentado en la máquina debes ajustar los apoyos: el final del asiento debe coincidir con el hueco popliteo (detrás de la rodilla) y la zona lumbar en contacto con el respaldo, finalmente regula el rodillo del brazo de palanca colocándolo al final de la tibia, en la articulación del tobillo. Inicia el movimiento de extensión desde una flexión de La rodilla de 90°, no menos y realiza la extensión de la pierna hasta llegar de forma controlada a la extensión completa de la rodilla.	Bajo	15	Publica
Abdominales	Ejercicio de Abdominales	Maquina 14	Se puede encontrar en todos los gimnasios y es una de las máquinas para abdominales más utilizadas por lo fácil que resulta trabajar con ella. El peso que debes colocar irá en función de la fuerza con la que puedas “tirar” de los agarres. Una recomendación para usar esta máquina y hacer abdominales en general es que a la hora de ejecutar el ejercicio hay que apretar bien el abdomen, sobre todo, para no lastimar la espalda concentrando la tracción sobre ella.	Bajo	15	Publica

Figura 5.26: Lista de Ejercicios Disponibles de un Usuario Autenticado.

Además, si pulsamos sobre el icono desplegable superior de la derecha (☰), se dará la opción de poder visualizar también los ejercicios públicos de otros usuarios. Destacar, que sobre los distintos ejercicios se tendrán unos privilegios u otros en función de su privacidad.

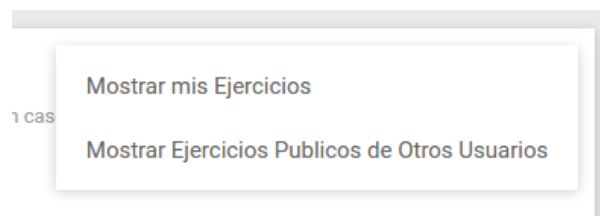
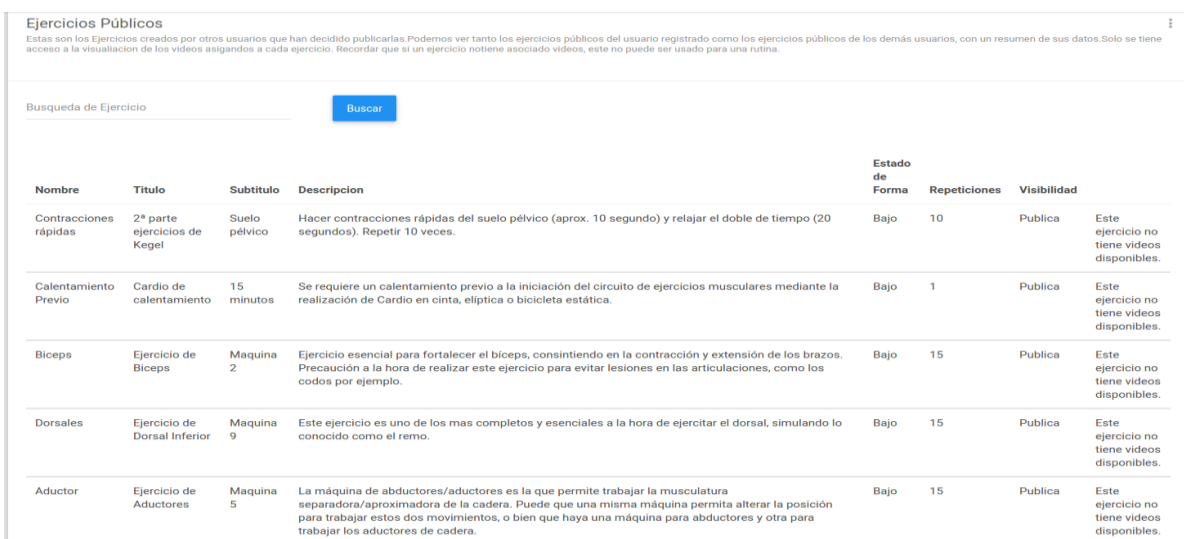


Figura 5.27: Menú desplegable para selección de Ejercicios a Visualizar.



Nombre	Título	Subtítulo	Descripción	Estado de Forma	Repeticiones	Visibilidad
Contracciones rápidas	2ª parte ejercicios de Kegel	Suelo pélvico	Hacer contracciones rápidas del suelo pélvico (aprox. 10 segundos) y relajar el doble de tiempo (20 segundos). Repetir 10 veces.	Bajo	10	Publica
Calentamiento Previo	Cardio de calentamiento	15 minutos	Se requiere un calentamiento previo a la iniciación del circuito de ejercicios musculares mediante la realización de Cardio en cinta, elíptica o bicicleta estática.	Bajo	1	Publica
Biceps	Ejercicio de Biceps	Maquina 2	Ejercicio esencial para fortalecer el bíceps, consiguiendo en la contracción y extensión de los brazos. Precaución a la hora de realizar este ejercicio para evitar lesiones en las articulaciones, como los codos por ejemplo.	Bajo	15	Publica
Dorsales	Ejercicio de Dorsal Inferior	Maquina 9	Este ejercicio es uno de los mas completos y esenciales a la hora de ejercitar el dorsal, simulando lo conocido como el remo.	Bajo	15	Publica
Aductor	Ejercicio de Aductores	Maquina 5	La máquina de abductores/ductores es la que permite trabajar la musculatura separadora/aproximadora de la cadera. Puede que una misma máquina permita alterar la posición para trabajar estos dos movimientos, o bien que haya una máquina para abductores y otra para trabajar los aductores de cadera.	Bajo	15	Publica

Figura 5.28: Lista de Ejercicios Públicos de otros Usuarios Disponibles para un Usuario Autenticado.

Para los ejercicios propios de un usuario se pueden realizar las siguientes acciones:

5.5.2.1 Modificar Ejercicio

Esta acción permite al usuario propietario del ejercicio modificar cualquier campo del mismo, modificándose en la base de datos del sistema, siempre que la modificación sea posible, y mostrando un mensaje de aviso.

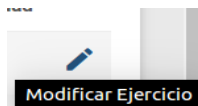


Figura 5.29: Botón correspondiente a la Modificación de un Ejercicio.

Modificar Ejercicio

A continuación introduzca los datos del local que desea registrar. Recuerde que el hecho de introducir locales es completamente gratuito, por lo que puede añadir todos los locales que desee sin ningún coste. Por otra parte, es obligatorio rellenar todos los campos de manera correcta, si no, no se completará el registro del local.

Nombre del Ejercicio: Gemelos

Título: Ejercicio de Gemelos

Subtítulo: Máquina 4

Descripción del Ejercicio: Elevaciones de talón sentado normales: Busca la máquina específica para hacer este ejercicio y céntrate más en seguir un rango de recorrido completo y no tanto en usar un peso alto. En caso de que no la tengas existen opciones alternativas como hacerlo con mancuernas sobre las rodillas, aunque personalmente me parecen muy incómodas, por lo que trataría de buscar ejercicios alternativos.

Estado de Forma: -- Seleccione una Opcion --

Repeticiones del Ejercicio: 15

Visibilidad: -- Seleccione una Opcion --

Confirmar **Cancelar**

Figura 5.30: Formulario para Modificación de Ejercicio.

5.5.2.2 Subir Vídeo

Esta opción permite asignar un vídeo, en formato “.mp4” a los ejercicios propios de un usuario.

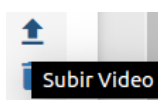


Figura 5.31: Botón correspondiente a la Subida de un Vídeo para un Ejercicio.

Asociación de video a Ejercicio

En esta vista, se permite la subida de un video por cada ejercicio que tengamos creado en nuestro servicio web.

Upload your Video

Seleccionar archivo Ningún archivo seleccionado

Confirmar **Cancelar**

Figura 5.32: Formulario para la Subida de un Vídeo y Asignación a un Ejercicio.

5.5.2.3 Ver Vídeo

Esta opción permite visualizar un vídeo que ya se haya asignado a un ejercicio en la misma aplicación Web. Sólo podrá visualizar el vídeo si ya está asociado al ejercicio, lo cual esta opción aparece sólo en este caso. Para los ejercicios que no tengan vídeos asociados, sólo se permite la opción de subir vídeo.



Figura 5.33: Botón correspondiente a la Visualización de un Vídeo Asociado a un Ejercicio.

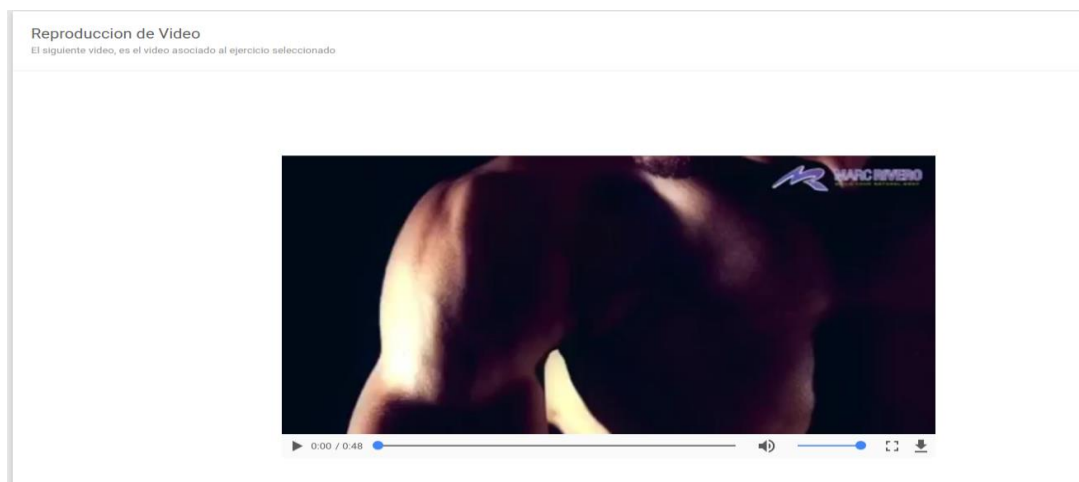


Figura 5.34: Vista Correspondiente a la Visualización de un Vídeo Asociado a un Ejercicio.

5.5.2.4 Eliminar Vídeo/Ejercicio

Este botón tiene una doble funcionalidad. Si el ejercicio tiene un vídeo asociado y pulsamos dicha opción, el vídeo se elimina del ejercicio, mostrando un mensaje de aviso. Por otro lado, si el ejercicio no tiene vídeo asociado y pulsamos dicha opción, el ejercicio será eliminado del sistema, borrando sus datos de la base de datos, mostrando un mensaje de aviso.



Figura 5.35: Botón de Opciones de Eliminar Vídeo/Ejercicio.

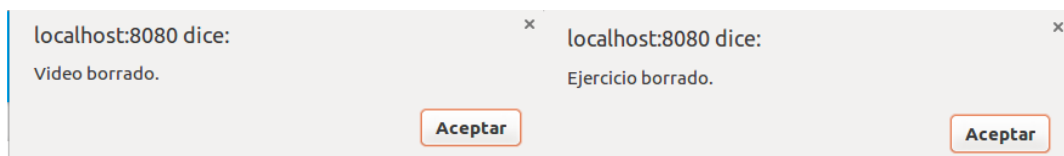


Figura 5.36: Mensajes de Aviso de Borrado de Vídeo/Ejercicio con Éxito.

5.6 Ajustes

En este apartado, explicaremos las opciones posibles por parte del usuario autenticado en el caso de que quiera realizar ajustes en su perfil de usuario o dar de baja el usuario, dándose como opciones disponibles en la pestaña de “Ajustes” del menú desplegable.

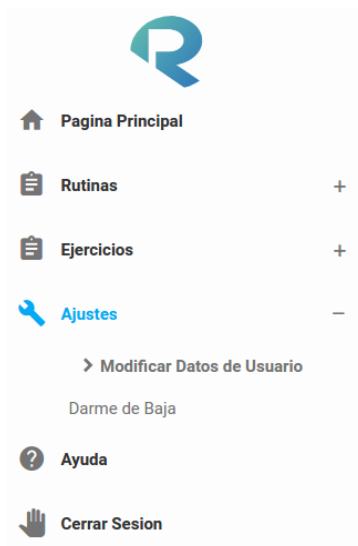


Figura 5.37: Opciones de Ajustes en el menú desplegable

5.6.1 Modificar Datos de Usuario

En esta opción, primero se muestra al usuario los datos con los que se ha registrado en la aplicación Web en un formulario (Figura 5.16).

Un formulario web con el título 'Modificar mi perfil' y un subtítulo que indica que se muestran los datos de usuario para su modificación. El formulario contiene los siguientes campos: 'E-mail' con el valor 'franciscodr42@gmail.com'; 'Nombre' con el valor 'Francisco José Díaz Romero'; 'Fecha de nacimiento' con el valor '30/04/1994'; 'Número de teléfono (opcional)' con el valor '620431369'; 'Contraseña' con caracteres ocultos por puntos; y 'Confirmar Contraseña' con caracteres ocultos por puntos. En la parte inferior del formulario hay dos botones: 'Confirmar' y 'Cancelar'.

Figura 5.38: Formulario para la Modificación de Datos de Perfil de Usuario.

Sobre este formulario, el usuario podrá modificar los datos de su perfil, pero siempre con la confirmación de la contraseña de acceso a su perfil. Si el usuario ha realizado la modificación correctamente y ha introducido su contraseña de forma correcta, la modificación realizada se llevará a cabo, mostrándose un mensaje de aviso (Figura 5.39).

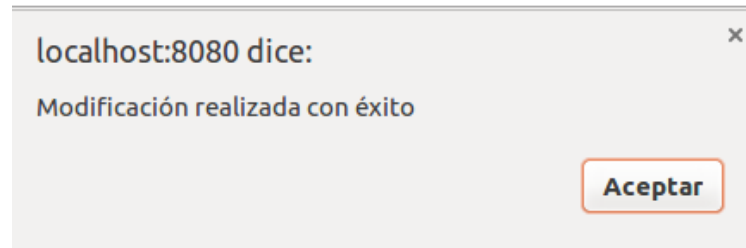


Figura 5.39: Mensaje de aviso de Modificación Correcta de Datos de Perfil de Usuario.

5.6.2 Dar de Baja al Usuario

En este apartado, aplicaremos las opciones disponibles por parte del usuario en caso de que seleccione la opción “Darse de Baja” de la figura 5.15.

Hay dos opciones disponibles:

- **Confirmar:** Elimina al usuario de la aplicación, así como todas sus Rutinas y Ejercicios. En el caso de que se produzca con éxito, aparecerá un mensaje de confirmación como el de la figura 5.19, y en el caso opuesto, es decir, si se da un error, aparecerá un mensaje informando de dicho error.
- **Cancelar:** Cancela la operación y redirige al usuario a la pantalla de la Figura 5.1.

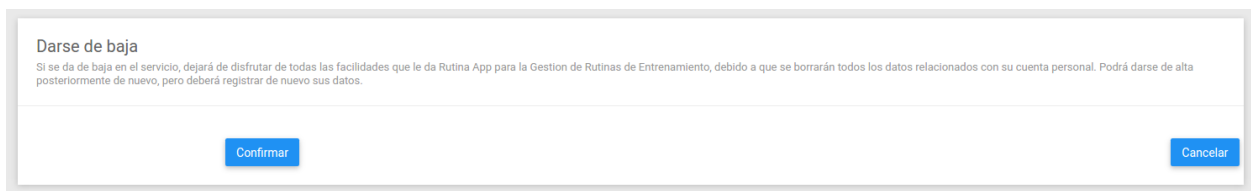


Figura 5.40: Formulario para Dar de Baja al Usuario.

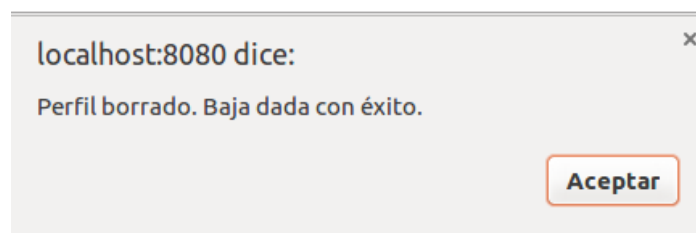
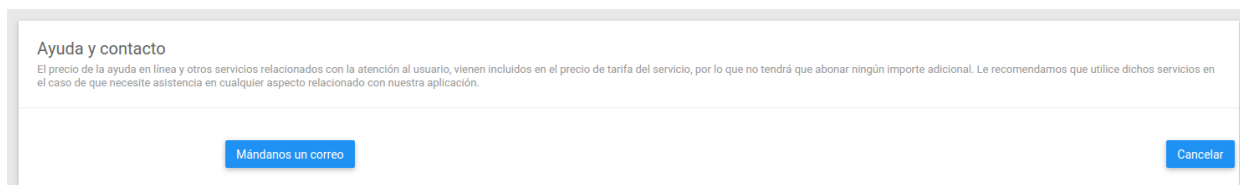


Figura 5.41: Mensaje de confirmación de Usuario dado de Baja.

5.7 Ayuda y Contacto (Envío de Email)

En este apartado, explicaremos las opciones disponibles por parte del usuario en el caso de que seleccione la opción “Ayuda” de la Figura 5.5. En la siguiente ilustración, se mostrará la pantalla que se muestra por defecto en el caso de que se seleccione dicha opción.



Ayuda y contacto

El precio de la ayuda en línea y otros servicios relacionados con la atención al usuario, vienen incluidos en el precio de tarifa del servicio, por lo que no tendrá que abonar ningún importe adicional. Le recomendamos que utilice dichos servicios en el caso de que necesite asistencia en cualquier aspecto relacionado con nuestra aplicación.

Mándanos un correo Cancelar

Figura 5.42: Formulario de Ayuda al Usuario.

Hay dos opciones disponibles:

- **Mándanos un correo:** Abre una ventana del cliente de correo por defecto en el ordenador del usuario, especificando la dirección de correo del servicio de Atención al cliente (Figura 5.43). Este servicio dispone de una cuenta de correo electrónico en la que se recibirán todas las sugerencias por parte de los usuarios al soporte de la aplicación.
- **Cancelar:** Redirige al usuario a la pantalla de la Figura 5.4.

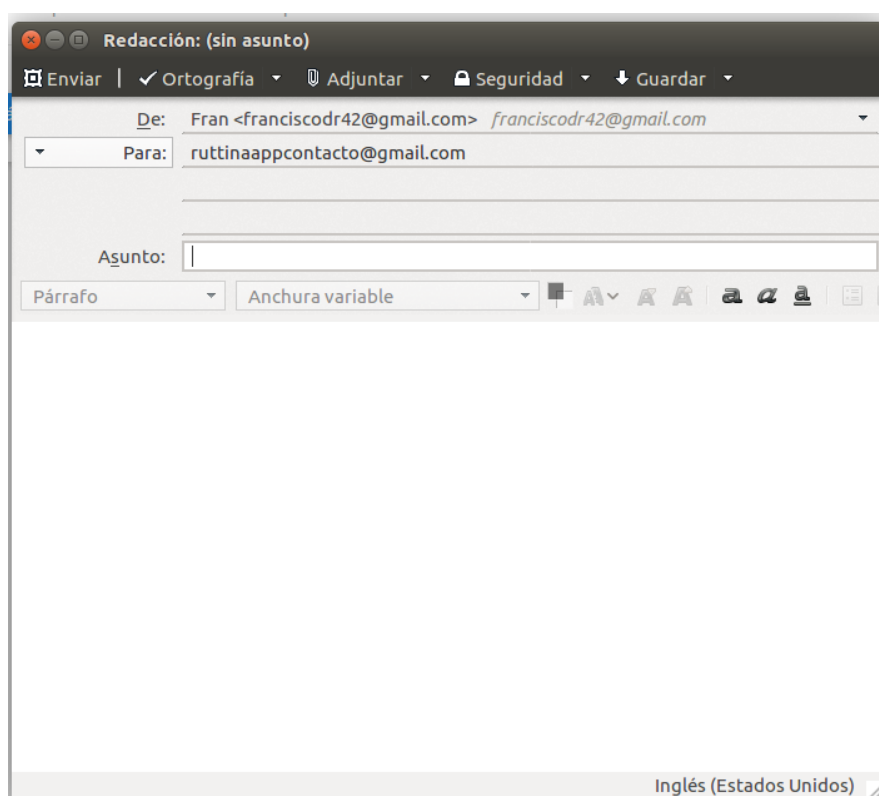


Figura 5.43: Ventana del Cliente de Correo del Usuario.

5.8 Cerrar sesión de usuario

En el caso de que el usuario autenticado seleccione la opción de “Cerrar Sesión” de la Figura 5.5, aparecerá un mensaje avisando de que la sesión se ha cerrado correctamente (Figura 5.44). En caso de que la sesión no se haya cerrado con éxito, se mostrará otro mensaje informando del error.

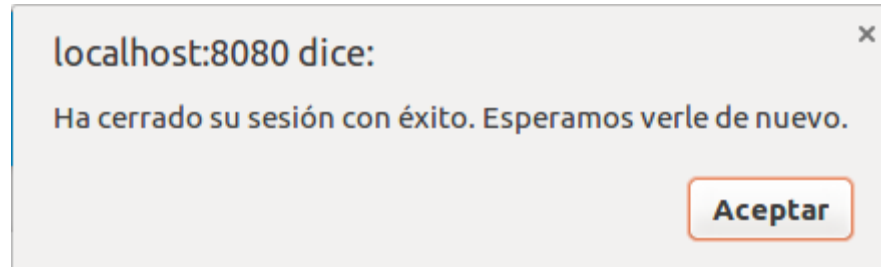


Figura 5.44: Mensaje de Aviso en el caso de que el Usuario Cierre su Sesión.

6 LÍNEAS DE MEJORA Y CONCLUSIONES

La única parte donde el “éxito” aparece antes que el "trabajo" es en el diccionario.

Vidal Sasoon.

En este último capítulo se describirán algunas líneas de mejora sobre el proyecto desarrollado, tanto en funcionalidad como en el uso de otras tecnologías más innovadoras y eficientes. Con estas mejoras, se busca como objetivo enriquecer la aplicación en todos sus aspectos, ya que siempre hay algo que se puede mejorar para conseguir un producto más completo y útil.

Por último, se incluirán las conclusiones sobre el proyecto y toda la trayectoria llevada a cabo para conseguir la solución que se proponía. Se detallarán también, aspectos de aprendizaje, opiniones, experiencias, y en definitiva todas las conclusiones sacadas sobre este proyecto, incluyendo también las dificultades encontradas para el desarrollo del mismo.

6.1 Líneas de Continuación del Proyecto

6.1.1 API entre Aplicación Android y Servicio Web

Son muchas las mejoras que pueden añadirse a una aplicación de este tipo, además de nuevas funcionalidades que la enriquezcan de usabilidad, por los usuarios que la requieran. Analizando este proyecto desarrollado, en conjunto con la aplicación Android proporcionada, podemos observar que la aplicación móvil no accede directamente al Servicio Web para gestionar rutinas de entrenamiento, cosa que sería posible.

Para ello, sería necesario implementar una API de acceso al Servicio Web Spring desarrollado desde la aplicación Android. De esta forma, el usuario, una vez registrado en la aplicación móvil, podrá acceder a la aplicación web desde la misma aplicación móvil sin necesidad de recurrir a un navegador web. Desde la aplicación móvil podrá realizar las mismas funciones que en la aplicación Web, permitiendo la creación, gestión y descarga de las rutinas de entrenamiento que desee.

Por otro lado, se puede observar que la independencia entre la aplicación Web y la aplicación móvil conlleva a que ambas tengan su propio sistema de autenticación y gestión de usuarios, lo cual se traduce a dos Bases de datos. Si implementamos esta API, sería conveniente integrar una única base de datos para la aplicación móvil y la aplicación Web de forma centralizada en el servicio Web. De esta forma, conseguimos que las credenciales de usuario sean las mismas para ambas aplicaciones y que la información esté disponible de forma distribuida y en todo momento.

6.1.2 Mejoras de Funcionalidad de la Aplicación Web

Son muchas las mejoras funcionales que pueden añadirse a una aplicación Web, ya que el desarrollo de las mismas consta de una mejora y evolución continua a medida que los usuarios de la misma la usan y manifiestan las carencias que van encontrando.

A continuación, vamos a detallar algunas mejoras que han comentado algunos usuarios que han hecho uso de esta aplicación Web durante su despliegue, y que podrían incluirse para dotarla de más características funcionales:

- **Más Opciones para Rutinas y Ejercicios:** Pueden incluirse más características sobre ellos, como por ejemplo, un sistema de puntuaciones, donde cada usuario que disponga de una misma rutina con ejercicios pueda evaluar las rutinas de entrenamiento y sus ejercicios asociados, permitiendo establecer así un ranking de las mejores rutinas. También puede incluirse un cuadro de comentarios sobre las rutinas y ejercicios, donde los usuarios pueden dejar escritos sus comentarios sobre las rutinas y ejercicios.

Por otro lado, cabe destacar que también sería interesante permitir la sincronización de la aplicación con alguna red social como Facebook, permitiendo compartir y publicar a través de ella tu actividad física y tus rutinas y ejercicios creados, además de otras muchas funcionalidades. Este es un tema que hoy día está de moda en el desarrollo tanto de aplicaciones móviles como aplicaciones Web.

- **Eliminar Limitaciones:** Principalmente, destacar que para que un ejercicio pueda ser reproducido por la aplicación móvil es necesario que tenga un vídeo asociado, por lo que en la aplicación Web se obliga a este mismo requisito. Sería interesante y mucho más sencillo para el usuario, que no se requiera un vídeo para cada ejercicio, dejándose como una opción. Además, se podría dar la opción de incluir imágenes en lugar de vídeos, o ambas opciones.
- **Filtros de Búsqueda más Detallados:** Actualmente, solo se dispone de un solo filtro de búsqueda, en el que se busca por la coincidencia de una cadena en algunos de los campos de las rutinas o ejercicios (Nombre, Título o Descripción). Podrían añadirse otros filtros más detallados, como por ejemplo, filtros por orden de fecha, por popularidad en cuanto al número de descargas, por valoraciones de usuarios, por número de ejercicios en la rutina, etc.

También sería de gran interés la agrupación de ejercicios atendiendo a ciertos criterios, como por ejemplo, atendiendo a la zona muscular o corporal, la intensidad o nivel de forma requerido para el ejercicio, e incluso podría incluirse un sistema de geolocalización con Google Maps, donde se permita ver las rutinas en función de la localización desde la que se hayan creado.

- **Sistema de Gestión de Alertas para la Realización de Ejercicios:** esta función permite a los usuarios planificar sus entrenamientos de forma que la aplicación les avisará de cuándo deben realizar los entrenamientos y de qué forma. Esto ayudará a que los usuarios puedan seguir mejor el entrenamiento, y también puede darse para muchos otros usos, ya que esta aplicación puede ser usada en otros campos, como por ejemplo, para la preinscripción del ejercicio físico o de rehabilitación de un fisioterapeuta.
- **Gestión de Estadísticas sobre los Entrenamientos Realizados:** La aplicación Web carece de esta funcionalidad, aunque la aplicación móvil sí permite obtener gráficas sobre el rendimiento de la actividad física. Sería de gran interés permitir el volcado de esa información hacia la aplicación Web, además, de permitir la conexión de la aplicación móvil con alguna pulsera deportiva o elemento que permita tomar datos como las pulsaciones o ritmo cardiaco.
- **Planificación de la Alimentación Adaptado a la Rutina de Entrenamiento:** Otra de las características que se podrían incluir sería la de que se permitiera la realización de una dieta adaptada al entrenamiento a realizar por el usuario, acorde a su nivel de forma y de las calorías que se deberían consumir para conseguir resultado.

6.2 Líneas de Mejora

6.2.1 JQuery VS AngularJS

Dado que JQuery proporciona una mayor sencillez de uso para la implementación de controlador en el cliente, cabe destacar que esta librería fue construida para abstraer al programador de las diversas complejidades que presentaba la programación de estos para adaptarlos y hacerlos funcionar con el navegador. Esta librería permite trabajar directamente con el DOM, la cual nos permite conseguir multitud de funcionalidades.

AngularJS es otro framework destinado a la creación de páginas y aplicaciones Web más pesadas y complejas, basándose en el Modelo-Vista-Controlador (MVC), permitiendo que el desarrollo y las pruebas sean más sencillos. Proporciona características como el enrutamiento, directivas, modelos de inyección de dependencias, pruebas unitarias, etc. Además, se consigue mayor rapidez y eficiencia de ejecución, aunque su curva de aprendizaje es mucho más pronunciada.



Figura 6.1: Logo de AngularJS.

Sería de gran interés el uso de esta tecnología en lugar de JQuery, o la integración de ambas, ya que Bootstrap hace uso de JQuery como principal elemento de control de su diseño. En este proyecto, se ha hecho uso de JQuery, ya que su aprendizaje es más rápido e intuitivo, y debido a la gran cantidad de tecnologías que han sido necesarias de aprender para conseguir nuestros objetivos, se deja como mejora del mismo.

6.2.2 JDBC VS JPA, Hibernate

JDBC nos proporciona total libertad a la hora de diseñar y consultar una base de datos, sin ningún tipo de limitación sobre las mismas. Nos permite realizar consultas nativas para cada base de datos, lo que favorece a la velocidad de respuesta y los resultados son devueltos en un "ResultSet", de los cuales podemos extraer solamente aquellos datos que requerimos, sin necesidad de extraer toda la Entidad de datos, como en el caso de JPA o Hibernate.

Por otro lado, JPA o Hibernate nos permite desarrollar de una forma mucho más rápida y con muchos menos errores en tiempo de ejecución, ya que nos permite modelar nuestra entidad de datos como Clases Java, las cuales serán convertidas a las directivas de SQL, según sea la operación a realizar (INSERT, UPDATE, SELECT). Sin embargo, estas tecnologías empeoran la eficiencia y el tiempo de respuesta de las consultas a nuestra base de datos, ya que deben realizar una gran cantidad de conversiones entre las entidades de datos a consultas y los "ResultSet" pasarlos a clases, además de que cada registro representa un objeto en memoria que tendrá que ser administrado.



Figura 6.2: Logo de JPA, Hibernate.

6.2.2.1 Ventajas y Desventajas

	Ventajas	Desventajas
JDBC	<p>Ofrece mayor rapidez y eficiencia, ya que es la forma más directa de consultar a la BBDD.</p> <p>Permite explotar al máximo las funcionalidades de la BBDD</p>	<p>Mantenimiento mucho más costoso.</p> <p>Introduce muchos errores en tiempo de ejecución</p> <p>El desarrollo es mucho más lento.</p>
JPA, Hibernate	<p>Permite un desarrollo más rápido y eficiente.</p> <p>Permite trabajar con las BBDD por medio de Entidades en vez de Querys.</p> <p>Ofrece un paradigma orientado a Objetos.</p> <p>Elimina errores en tiempo de ejecución.</p> <p>Mejora el Mantenimiento del Software.</p>	<p>No ofrece toda la funcionalidad que permite JDBC, ya que no permite acceder directamente a la BBDD.</p> <p>Ofrece menos rapidez y eficiencia que JDBC.</p> <p>Su aprendizaje es más complejo y requiere más tiempo.</p>

6.2.3 Compresión y cifrado del Código JS

6.2.3.1 Compresión

Esta técnica puede realizarse para reducir las líneas, espacios y aquellos caracteres no visibles para el programador con el fin de obtener un código que ocupe menos. Hay herramientas online que realizan esta labor, la cual permite reducir el ancho de banda consumido por los usuarios, haciendo a su vez que la interfaz Web se cargue más rápido, dando mayor sensación de fluidez. Destacar que esta característica es usada únicamente con esta finalidad, ya que el código que se servirá será el comprimido, mientras que el que se mantiene es el original, debido a la difícil lectura del código comprimido.

6.2.3.2 Cifrado

Para cifrar el código JavaScript, de forma que sea ilegible, se usan algoritmos de codificación sobre los ficheros JavaScript para evitar ataques y vulnerabilidades de seguridad.

6.2.4 Spring Security

En este proyecto no se han implementado todas las funcionalidades que ofrece este módulo de Spring Framework, principalmente por la dificultad que presenta el mismo para una correcta implementación y adaptación del mismo a cualquier aplicación. En general, los temas de seguridad son bastante complejos y requieren de un estudio específico y orientado únicamente al ámbito de la seguridad, lo cual la implementación de seguridad en Servicios Web REST requiere de otro Trabajo de Fin de Grado para su estudio.

6.2.5 OAuth2 (Autenticación)

Open Authorization (OAuth) es un estándar abierto que permite flujos simples de autorización para sitios web o aplicaciones informáticas. Se trata de un protocolo propuesto por Blaine Cook y Chris Messina, que permite autorización segura de una API de modo estándar y simple para aplicaciones de escritorio, móviles y web.

OAuth permite a un usuario del sitio A compartir su información en el sitio A (proveedor de servicio) con el sitio B (llamado consumidor) sin compartir toda su identidad. Para desarrolladores de consumidores, OAuth es un método de interactuar con datos protegidos y publicarlos. Para desarrolladores de proveedores de servicio, OAuth proporciona a los usuarios un acceso a sus datos al mismo tiempo que protege las credenciales de su cuenta. Este mecanismo es utilizado por compañías como Google, Facebook, Microsoft, Twitter y Github para permitir a los usuarios compartir información sobre sus cuentas con aplicaciones de terceros o sitios web.



Figura 6.3: Logo de OAuth.

OAuth tiene dos grandes ventajas, soluciona el problema de la confianza entre un usuario y aplicaciones de terceros, y a su vez permite a un proveedor de servicios/API facilitar a aplicaciones de terceros a que amplíen sus servicios con aplicaciones que hacen uso de los datos de sus usuarios de manera segura y dejando al usuario la decisión de cuándo y a quien, revocar o facilitar acceso a sus datos, creando así un ecosistema de aplicaciones alrededor del proveedor de servicios/API.

La confianza, o la falta de ella, es el motivo principal para querer implementar un protocolo de autorización como OAuth. La falta de confianza de un usuario con una aplicación de terceros puede propiciar que el usuario quiera permitir a la aplicación realizar tareas y obtener datos en su nombre pero sin darle las credenciales de autenticación a dicha aplicación.

6.2.6 Documentación del Servicio Web REST: Swagger

Swagger es una nueva especificación para documentar Servicios REST. Swagger proporciona un cliente web para poder acceder de forma muy cómoda y vistosa a la documentación de nuestros servicios. Además, permite probar el servicio desarrollado de forma extremadamente sencilla.



Figura 6.4: logo de Swagger.

Incluir este sistema de documentación sería de gran ayuda y utilidad en cuanto al trabajo de varios programadores sobre el mismo servicio Web, ya que es más sencillo y amigable gráficamente que otros sistemas de documentación como Javadoc, en el caso del desarrollo en Java.

6.3 Conclusiones finales

Iniciaremos la conclusión a este proyecto recordando todo el ámbito del mismo y su finalidad. Una vez el usuario ha iniciado sesión, puede acceder a su lista de ejercicios creados por el mismo y los publicados por otros usuarios, y de esta misma forma para las rutinas. Además, el usuario tendrá la posibilidad de crear sus propias rutinas y configurarlas añadiendo y eliminando ejercicios de los que tenga disponibles. Una vez configurada la rutina, el usuario podrá descargarla, aunque también podrá descargar rutinas publicadas por otros usuarios.

La descarga de la rutina en formato “.Zip” es el paquete que la aplicación Android leerá del directorio Downloads del dispositivo móvil, tal y como se especifica en el proyecto que describe dicha aplicación, realizado por Miriam Franco Maireles. Este paquete contiene una rutina de entrenamiento que deberá contener una carpeta que contenga un archivo JSON, donde se recaudará toda la información sobre la rutina y los ejercicios que tiene asociados, todo ello en un formato específico para que la aplicación pueda reconocerlos. También se incluirán en el mismo directorio, aquellos vídeos que estén asociados a los ejercicios de la rutina.

Tras conseguir los principales objetivos de este proyecto, los cuales serían los detallados anteriormente, podemos sacar algunas conclusiones finales sobre la trayectoria seguida, dificultades encontradas y satisfacción en cuanto al aprendizaje.

Se ha desarrollado una aplicación sencilla de usar y entender, destinada a cualquier usuario que quiera hacer uso de ella, y con cierta proyección hacia un futuro comercial de la misma. Su diseño atractivo y sencillez de uso hacen de esta aplicación una plataforma en la que cualquier usuario puede crear sus propios entrenamientos, e incluso un entrenador personal podría preinscribir rutinas de entrenamientos a los deportistas que tuviese a su cargo.

Por consiguiente, valorando la dedicación y esfuerzo para la realización de este trabajo, destacar que “las claves del fin son el trabajo y la constancia”, y que con esto se puede llegar a conseguir aquello que nos proponíamos. Son muchas las dificultades encontradas a lo largo de la realización de este proyecto, debido a la gran cantidad de tecnologías que se necesita aprender previamente para realizar un Servicio Web de este tipo.

He aprendido multitud de conceptos, tecnologías y herramientas que hasta ahora desconocía cuál era su finalidad o funcionamiento, como por ejemplo, el propio Framework de Spring, sobre el cual he necesitado gran cantidad de formación previa antes de comenzar a ponerlo en práctica para el Servicio Web que he desarrollado en este proyecto.

Son muchos los errores cometidos por el camino, los cuales creo que son los sucesos más importantes, pues de ellos se aprenden y son lo que nos hacen avanzar cuando los resolvemos.

Finalmente, destacar que este proyecto me ha dado la oportunidad de abrir mi conocimiento en un campo de las tecnologías de la información el cual me interesa muchísimo, ya que se adapta perfectamente a mis curiosidades, intereses y vocación, tanto en el sector de las Telecomunicaciones e Informática, como en el ámbito deportivo. Dar las Gracias a mi Tutora María Teresa Ariza Gómez por brindarme esta oportunidad, además de ayudarme y asesorarme, estando siempre disponible y con dedicación hacia este proyecto para que haya podido llevarse a cabo.

REFERENCIAS

[1] En relación a MySQL y Módulos relacionados:

[1][1] Web Oficial de MySQL: <https://www.mysql.com/>

[1][2] Documentación de MySQLWorkBench: <https://dev.mysql.com/doc/workbench/en/>

[1][3] Documentación de XAMPP: <https://www.apachefriends.org/community.html>

[2] En relación a Spring Framework y Módulos Relacionados:

[2][1] Documentación de Spring: <https://spring.io/docs>

[2][2] Guías de Spring: <https://spring.io/guides>

[2][3] Guías para API REST: <https://openehr.atlassian.net/wiki/display/spec/openEHR+REST+APIs>
<https://bocoup.com/blog/documenting-your-api>
<https://gist.github.com/iros/3426278>

[2][4] Tutoriales de Propósito General: <http://www.mkyong.com/tutorials/>

[2][5] Tutoriales Específicos en Spring: <http://www.baeldung.com/>

<http://www.java2blog.com/2013/04/restful-web-service-tutorial.html>

<http://blog.hcosta.info/fundamentos-de-spring-los-beans/>

<http://blog.netgloo.com/>

[3] En relación a Java:

[3][1] Tutoriales para Gestión de ficheros y paquetes ".Zip" en Java:

http://chuwiki.chuidiang.org/index.php?title=Manejar_ficheros_zip_con_java

<http://www.avajava.com/tutorials/lessons/how-can-i-create-a-zip-file-from-a-set-of-files.html?page=1>

<https://www.scribd.com/document/308407510/Como-Comprimir-Archivos-Zip-en-Java>

[3][2] Creación de JSON en Java: <https://www.java2novice.com/java-json/javax.json/create-json-array/>

[4] En relación a BootStrap(HTML5, CSS3 y JS):

[4][1] Referencia del W3C de HTML5: <https://dev.w3.org/html5/html-author/>

[4][2] Guía de HTML5 en MDN: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>

[4][3] Tutoriales de HTML5 de W3C: <http://www.w3schools.com/html/>

[4][4] Tutoriales de CSS3 de W3C: http://www.w3schools.com/css/css3_intro.asp

[4][5] Tutoriales y Documentación de BootStrap: <http://getbootstrap.com/getting-started/>

[5] En relación a JavaScript y librerías relacionadas:

[5][1] Tutoriales de JavaScript de W3C: <http://www.w3schools.com/js/>

[5][2] Web Oficial de JQuery: <https://jquery.com/>

[5][3] Web Oficial de JQuery Validation Plugin: <https://jqueryvalidation.org/>

[5][4] Perfil de Github de Klaus Hartl (librerías de cookies): <https://github.com/carhartl/>

[6] En relación al Despliegue de la Aplicación:

[6][1] Documentación de Pivotal tc Runtime Server:

<https://pubs.vmware.com/vfabric5/index.jsp?topic=/com.vmware.vfabric.tc-server.2.6/getting-started/install-directories-variables-configfiles.html>

[6][2] Documentación de Eclipse:

<http://www.eclipse.org/documentation/?sess=32267642b71e61a5209ee01bb85d8fdb>

ANEXO A: API REST DOCUMENTATION

En este anexo, se incluirá la documentación sobre la API REST desarrollada. En las siguientes plantillas se detallarán los distintos recursos disponibles en la aplicación y cómo acceder a ellos en función de la URI que los referencia. Se detallarán también las acciones u operaciones que se realizan sobre cada recurso, además de una descripción de los mismos.

A.1 Recurso Usuario

Método	URI	Parámetros	Descripción
POST	/RutinaRegister//	Parámetros de un Usuario	Esta acción permite añadir un Usuario quedando registrado en el Servicio Web.
POST	/Rutina_app/{user_id}	Id. Del usuario: user_id Parámetros de un Usuario	Esta acción permite modificar un Usuario quedando registrado en el Servicio Web.
GET	/Rutina_app//	Sin parámetros	Esta acción permite obtener una lista con los datos de todos los usuarios registrados en el sistema.
GET	/Rutina_app/{user_id}	Id. del usuario: user_id	Esta acción permite obtener una lista con los datos de un usuario registrado en el sistema.
DELETE	/Rutina_app//	Sin parámetros	Esta acción permite eliminar los datos de todos los usuarios registrados en el sistema.
DELETE	/Rutina_app/{user_id}	Id. del usuario: user_id	Esta acción permite eliminar los datos de un usuario registrado en el sistema.

A.2 Recurso Rutina

Método	URI	Parámetros	Descripción
POST	/Rutina_app/rutinas/{user_id}/	Id. del usuario propietario de la rutina: user_id Parámetros de una Rutina	Esta acción permite añadir una Rutina a un usuario, quedando registrada en el Servicio Web.
POST	/Rutina_app/rutinas/{user_id}/{rut_id}	Id. del usuario propietario de la rutina: user_id Id. de la rutina: rut_id Parámetros de una Rutina	Esta acción permite actualizar una Rutina a un usuario, quedando registrada en el Servicio Web.
GET	/Rutina_app/rutinas/{user_id}/{rut_id}	Id. del usuario propietario de la rutina: user_id Id. de la rutina: rut_id	Esta acción permite obtener una Rutina a un usuario registrado en el sistema.
GET	/Rutina_app/rutinas/{user_id}/	Id. del usuario propietario de la rutina: user_id	Esta acción permite obtener todas las Rutinas de un Usuario registrado en el sistema.
GET	/Rutina_app/rutinas/publicas/{rut_id}	Id. de la rutina: rut_id	Esta acción permite obtener una Rutina a través de su Identificador de Rutina en el sistema.
DELETE	/Rutina_app/rutinas/{user_id}/{rut_id}	Id. del usuario propietario de la rutina: user_id Id. de la rutina: rut_id	Esta acción permite eliminar una Rutina a un usuario registrado en el sistema.
DELETE	/Rutina_app/rutinas/{user_id}/	Id. del usuario propietario de la rutina: user_id	Esta acción permite eliminar todas las Rutinas de un Usuario registrado en el sistema.

POST	/Rutina_app/downloads/{user_id}/{rut_id}	Id. del usuario propietario de la rutina: user_id Id. de la rutina: rut_id	Esta acción permite descargar una Rutina a un usuario registrado en el sistema.
-------------	--	---	---

A.3 Recurso Ejercicio

Método	URI	Parámetros	Descripción
POST	/Rutina_app/ejercicios/{user_id}/	Id. del usuario propietario de la rutina: user_id	Esta acción permite añadir un Ejercicio a un usuario, quedando registrado en el Servicio Web.
POST	/Rutina_app/ejercicios/{user_id}/{ej_id}	Id. del usuario propietario de la rutina: user_id Id. del ejercicio: ej_id	Esta acción permite actualizar un Ejercicio a un usuario registrado en el Servicio Web.
GET	/Rutina_app/ejercicios/{user_id}/{ej_id}	Id. del usuario propietario de la rutina: user_id Id. del ejercicio: ej_id	Esta acción permite obtener un Ejercicio a un usuario, quedando registrado en el Servicio Web.
GET	/Rutina_app/ejercicios/{user_id}/	Id. del usuario propietario de la rutina: user_id	Esta acción permite obtener todos los Ejercicios a un usuario registrado en el Servicio Web.
DELETE	/Rutina_app/ejercicios/{user_id}/{ej_id}	Id. del usuario propietario de la rutina: user_id Id. del ejercicio: ej_id	Esta acción permite eliminar un Ejercicio a un usuario, quedando registrado en el Servicio Web.
DELETE	/Rutina_app/ejercicios/{user_id}/	Id. del usuario propietario de la rutina: user_id	Esta acción permite eliminar todos los Ejercicios a un usuario registrado en el Servicio Web.

A.4 Recurso Asociaciones Rutinas/Ejercicios

Método	URI	Parámetros	Descripción
POST	/Rutina_app/rutinas/asociaciones/{rut_id}/{ej_id}	Id. de la rutina: rut_id Id. del ejercicio: ej_id	Esta acción permite asociar un ejercicio a una rutina, ambos registrados en el sistema.
GET	/Rutina_app/rutinas/asociaciones/{rut_id}/	Id. de la rutina: rut_id	Esta acción permite obtener todos los ejercicios asociados a una rutina registrada en el sistema.
GET	/Rutina_app/rutinas/noasociaciones/{rut_id}/	Id. de la rutina: rut_id	Esta acción permite obtener todos los ejercicios que no estén asociadas a una rutina registrada en el sistema.
DELETE	/Rutina_app/rutinas/asociaciones/{rut_id}/{ej_id}	Id. de la rutina: rut_id Id. del ejercicio: ej_id	Esta acción permite eliminar la asociación de un ejercicio a una rutina, ambos registrados en el sistema.
DELETE	/Rutina_app/rutinas/asociaciones/{rut_id}/	Id. de la rutina: rut_id	Esta acción permite eliminar todas las asociaciones de ejercicios a una rutina, ambos registrados en el sistema.

A.5 Recurso Vídeo

Método	URI	Parámetros	Descripción
POST	/Rutina_app/videos/{user_id}/{ej_id}	Id. de la rutina: rut_id Id. del ejercicio: ej_id	Esta acción permite crear un vídeo en la BBDD y asociarlo a un ejercicio. Además, permite la subida del vídeo en formato “.mp4”, quedando almacenado en el repositorio del servidor.
PUT	/Rutina_app/videos/{user_id}/{ej_id}	Id. de la rutina: rut_id Id. del ejercicio: ej_id	Esta acción permite actualizar un vídeo en la BBDD y asociarlo a un ejercicio.
GET	/Rutina_app/videos/{user_id}/{ej_id}	Id. de la rutina: rut_id Id. del ejercicio: ej_id	Esta acción permite obtener un vídeo de la BBDD.
DELETE	/Rutina_app/videos/{user_id}/{ej_id}	Id. de la rutina: rut_id Id. del ejercicio: ej_id	Esta acción permite eliminar un vídeo de la BBDD y del repositorio del servidor

ANEXO B: MANUAL DE INSTALACIÓN Y DESPLIEGUE DE LA APLICACIÓN

En este anexo se desarrollará un manual de instalación de todas las herramientas que son necesarias para que la aplicación Web pueda funcionar, además de todos los pasos necesarios para el despliegue de la aplicación. Las herramientas necesarias son: Spring Tools Suite, XAMPP y mvn.

Destacar, que todo el procedimiento se ha hecho sobre un sistema operativo Ubuntu de 64 Bits.

B.1 Instalación de Java JDK 8

Para que el entorno que vamos a usar de Eclipse (STS) pueda funcionar es necesario tener instalado Java en el sistema, mas específicamente el kit de desarrollo JDK cuya última versión estable es la 1.8. En caso de no tener instalado el JDK, bastaría con ir a la siguiente web:

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

De esta, descargarnos la versión de 64 bits para Linux:

Java SE Development Kit 8u131		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.87 MB	jdk-8u131-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.81 MB	jdk-8u131-linux-arm64-vfp-hflt.tar.gz
Linux x86	164.66 MB	jdk-8u131-linux-i586.rpm
Linux x86	179.39 MB	jdk-8u131-linux-i586.tar.gz
Linux x64	162.11 MB	jdk-8u131-linux-x64.rpm
Linux x64	176.95 MB	jdk-8u131-linux-x64.tar.gz
Mac OS X	226.57 MB	jdk-8u131-macosx-x64.dmg
Solaris SPARC 64-bit	139.79 MB	jdk-8u131-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.13 MB	jdk-8u131-solaris-sparcv9.tar.gz
Solaris x64	140.51 MB	jdk-8u131-solaris-x64.tar.Z
Solaris x64	96.96 MB	jdk-8u131-solaris-x64.tar.gz
Windows x86	191.22 MB	jdk-8u131-windows-i586.exe
Windows x64	198.03 MB	jdk-8u131-windows-x64.exe

Figura B.1: Descarga de Java JDK 8.

Una vez descargado, descomprimos el paquete e instalamos dicha versión de java, y lo situamos en el directorio “/usr/local/java”.

También podríamos instalar esta versión de java con el siguiente comando:

```
$ sudo apt-get install oracle-java9-installer
```

Una vez instalado, comprobamos que la versión de Java existente en el sistema es la previamente instalada:

```
fran@fran: ~
fran@fran:~$ java -version
java version "1.8.0_121"
Java(TM) SE Runtime Environment (build 1.8.0_121-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.121-b13, mixed mode)
fran@fran:~$
```

Figura B.2: Versión de Java Instalado.

B.2 Instalación de Spring Tools Suite (STS)

Para comenzar la instalación de la Suite descargamos desde la página oficial de Spring la aplicación, y para ello vamos a la página Web oficial que a día de hoy se encuentra en el siguiente enlace:

<https://spring.io/tools>

Pulsamos sobre el botón de Download para descargar esta suite para Linux:

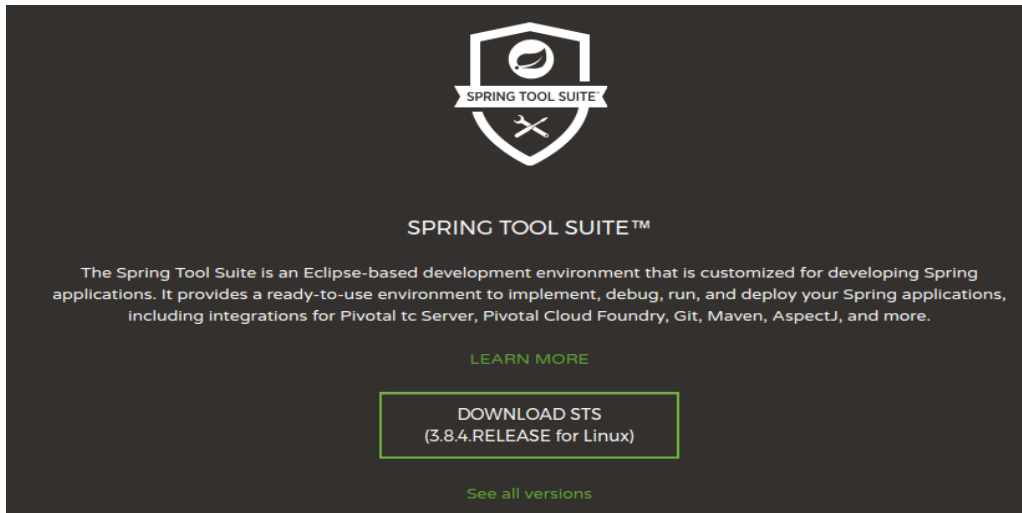


Figura B.3: Página de descarga de STS.

Una vez descargado, vamos a moverlo desde el sitio donde ha sido descargado al subdirectorio /opt. Posteriormente, descomprimos el paquete de STS en dicho directorio, y ya tendremos instalado el entorno.

Para ejecutar el entorno directamente desde el directorio descomprimido, accedemos al directorio “./sts-Bundle/sts-3.8.3.RELEASE” e iniciamos el ejecutable ubicado en ese directorio:

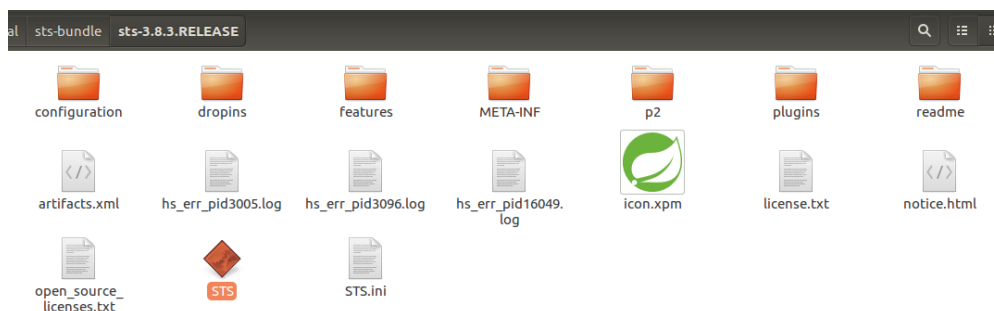


Figura B.4: Ejecutable de STS.

Una vez ejecutado, se iniciará el entorno, permitiendo configurar un directorio de trabajo o “workspace”:

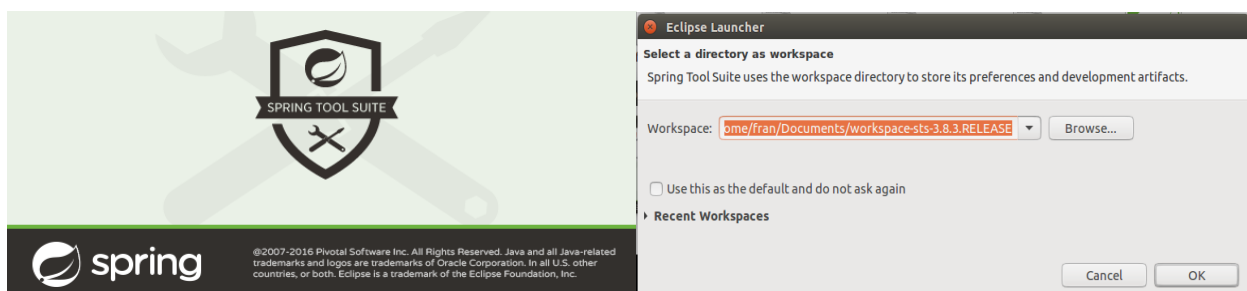


Figura B.5: Inicio de STS.

B.3 Instalación de XAMPP

Nos dirigimos a la web de descarga de XAMPP y descargamos el paquete en versión Linux:

<https://www.apachefriends.org/es/index.html>



Figura B.6: Pagina Web de XAMPP.

Nos posicionamos en la carpeta donde hemos descargado XAMPP. Vemos que se trata de un archivo ejecutable con extensión .run. Pero para poder ejecutarlo necesitamos darle permisos de ejecución. Para ello ejecutamos el siguiente comando sobre el ejecutable de XAMPP descargado:

```
$ sudo chmod 777 xampp-linux-x64-7.1.6-0-installer.run
```

Ahora abrimos una terminal en la carpeta y nos autenticamos como root con el comando “sudo su” y escribimos “./nombre_del_archivo para ejecutarlo”:

```
$/xampp-linux-x64-5.6.20-0-installer.run
```

Tras la ejecución de este comando, comenzará la instalación de XAMPP. Seguiremos el proceso de instalación dándole a siguiente en las interfaces que vayan apareciendo:

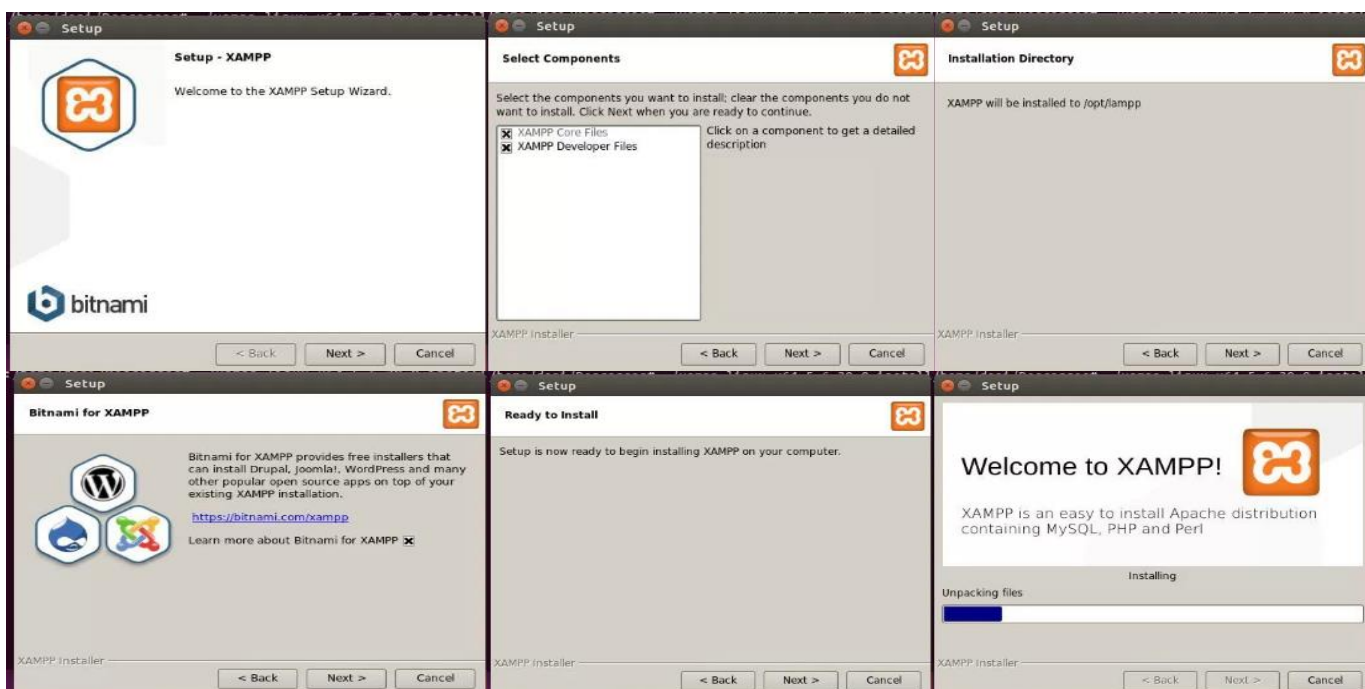


Figura B.7: Proceso de Instalación de XAMPP.

Una vez instalado XAMPP, procedemos a arrancar el servidor desde una terminal con el siguiente comando, el cual también sirve para detener o reiniciar el servidor (stop, restart):

```
$ sudo /opt/lampp/lampp start
```

Cuando el servidor se encuentra funcionando, podemos acceder al gestor de bases de datos a través de un navegador accediendo a la siguiente URL:

<http://localhost/phpmyadmin/>

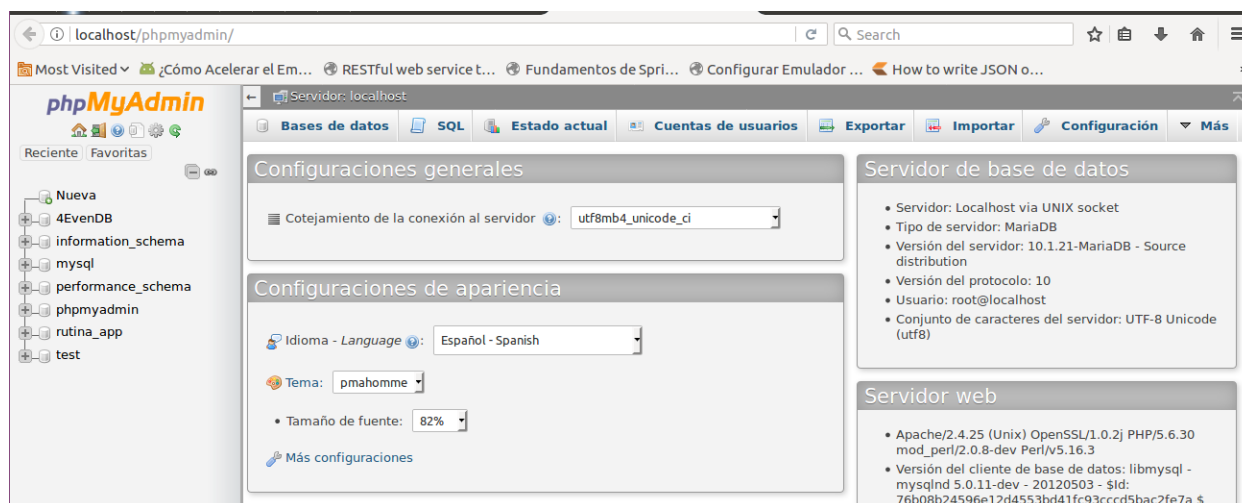


Figura B.8: Navegador Web de PHPMyAdmin para la Gestión de BBDD.

También se puede acceder al servidor de BBDD a través de una terminal con el siguiente comando:

```
$ sudo /opt/lampp/bin/mysql -u root -p
```

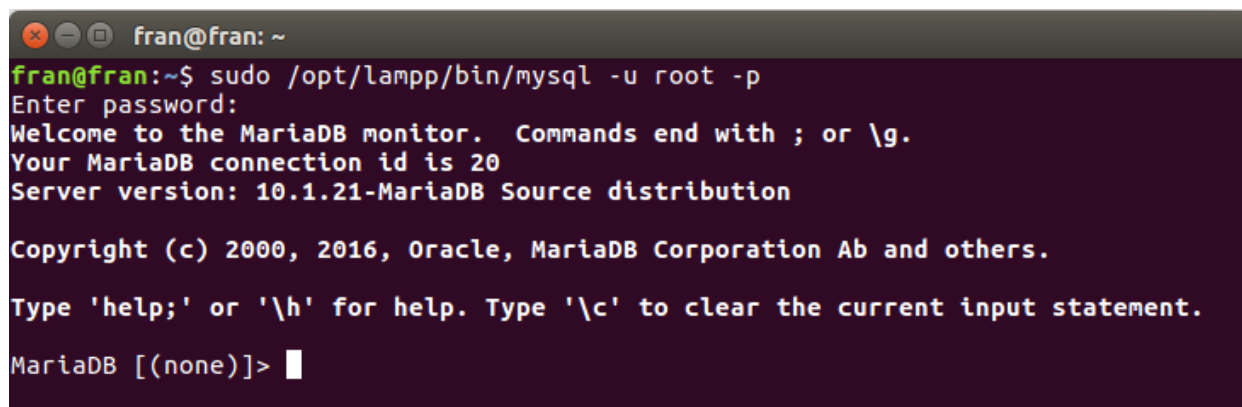


Figura B.9: Gestor de BBDD en consola.

B.4 Despliegue de la aplicación

Una vez iniciado el entorno de STS, procedemos a importar el proyecto que contiene la aplicación Web que se ha desarrollado en este proyecto. Para ello, seguimos los siguientes pasos:

1. Situamos el proyecto “rutina_app.zip” en el directorio de trabajo o “Workspace” del entorno STS.
2. Abrimos la pestaña “File->import” de STS, y aparecerá la siguiente ventana, en la cual seleccionamos la opción “Existing Maven Projects” y pulsamos “Next”:

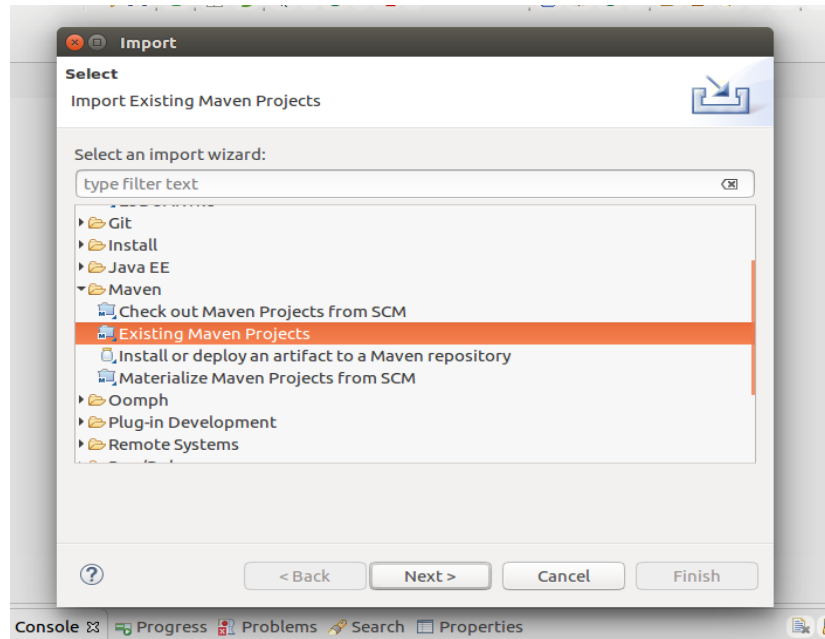


Figura B. 10: Interfaz para Importar proyecto.

3. En la siguiente ventana, buscamos el “Root Directory” de nuestro proyecto pulsando el Botón “Browser”, y finalmente pulsamos en el botón “Finish2 para importar el proyecto:

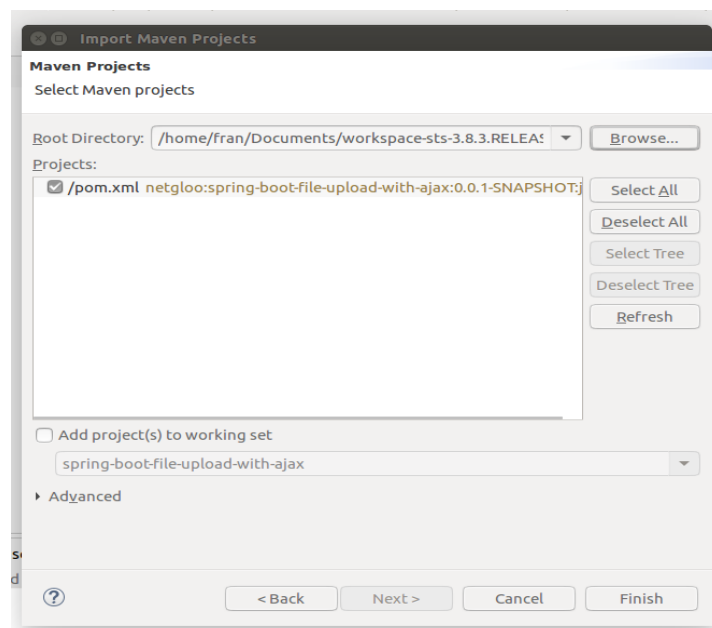


Figura B. 11: Directorio Raíz del Proyecto.

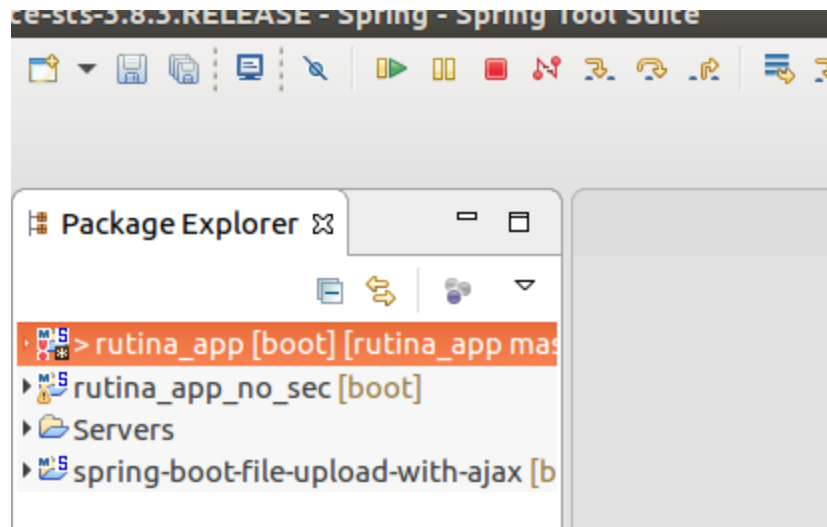


Figura B.12: Proyecto importado al entorno STS.

Teniendo el proyecto importado en el entorno STS, podríamos ejecutarlo directamente desde el mismo entorno, pero vamos a detallar los siguientes pasos para publicar el Servicio Web en un servidor Tomcat que proporciona STS, llamado Pivotal Tc Runtime Server:

1. Despliegue de la Base de Datos: Para que la aplicación tenga acceso a la base de datos, debemos crear esta a través de nuestro gestor. Para ello accedemos a la URL “<http://localhost/phpmyadmin/>” y pulsamos sobre la pestaña “nueva” para crear una nueva base de datos. Posteriormente, importamos el fichero “rutina_app.sql” de nuestro proyecto, el cual contiene las directivas SQL para la creación de las tablas y atributos necesarios.



Figura B.13: Creación de la BBDD.

- Configurar pom.xml: Para poder desplegar el servicio Web y que se genere un “.war”, es necesario añadir en el fichero pom.xml:

```
<groupId>org.springframework</groupId>
<artifactId>rutina_app</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>

<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-tomcat</artifactId>
<scope>provided</scope>
</dependency>
```

Figura B. 14: Modificaciones del pom.xml.

Además, deberá aparecer en la clase principal de la aplicación “RutinaAppApplication.java”, el siguiente contenido:

```
*pom.xml | *RutinaAppApplication.java
package src.rutina.app.Main;
import org.springframework.boot.SpringApplication;
@SpringBootApplication
public class RutinaAppApplication extends SpringBootServletInitializer {
    @Override
    protected SpringApplicationBuilder configure(SpringApplicationBuilder application) {
        return application.sources(RutinaAppApplication.class);
    }
    @Bean
    public WebSecurityConfigurerAdapter webSecurityConfigurerAdapter() {
        return new SecurityConfiguration();
    }
    public static void main(String[] args) throws Exception {
        SpringApplication.run(RutinaAppApplication.class, args);
    }
}
```

Figura B. 15: Clase Principal de la Aplicación.

- Generar paquete “.war” de la aplicación: Una vez tenemos configuradas las dependencias del pom y la clase principal de la aplicación, procedemos a la creación del paquete de la aplicación en formato “.war”. Esto se realizará mediante la terminal, con el comando “mvn”. Si no lo tenemos instalado, debemos instalarlo con el comando “apt-get install” desde la terminal. Los pasos a seguir para generar el “.War” son los siguientes:

- Nos situamos en la carpeta del proyecto importado, la cual se sitúa en el workspace de STS:

```
fran@fran: ~/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app
fran@fran:~$ cd Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app/
fran@fran:~/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app$
```

Figura B.16: Directorio del proyecto.

- Limpiamos el target del proyecto, en el cual se generan todas las dependencias y ficheros para crear el “.war”. Esto se hace ejecutando el siguiente comando:

```
fran@fran: ~/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app
fran@fran:~$ cd Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app/
fran@fran:~/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building rutina_app 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-clean-plugin:2.6.1:clean (default-clean) @ rutina_app ---
[INFO] Deleting /home/fran/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app/target
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 0.623 s
[INFO] Finished at: 2017-07-05T14:35:35+02:00
[INFO] Final Memory: 10M/150M
[INFO] -----
```

Figura B.17: Ejecución del comando “mvn clean”.

- Generamos el paquete “.war” con el comando “mvn package”. Este comando, generará un directorio “target” y dentro de este se ubicará el “.war” de la aplicación:

```
[INFO]
fran@fran:~/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app$ mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building rutina_app 0.0.1-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ rutina_app ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO] Copying 3402 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ rutina_app ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 27 source files to /home/fran/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app/target/classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ rutina_app ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory /home/fran/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ rutina_app ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.18.1:test (default-test) @ rutina_app ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-war-plugin:2.6:war (default-war) @ rutina_app ---
[INFO] Packaging webapp
[INFO] Assembling webapp [rutina_app] in [/home/fran/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app/target/rutina_app-0.0.1-SNAPSHOT]
[INFO] Processing war project
[INFO] Webapp assembled in [642 msecs]
[INFO] Building war: /home/fran/Documents/workspace-sts-3.8.3.RELEASE/rutina_app/rutina_app/target/rutina_app-0.0.1-SNAPSHOT.war
[INFO]
[INFO] --- spring-boot-maven-plugin:1.5.2.RELEASE:repackage (default) @ rutina_app ---
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time: 10.199 s
[INFO] Finished at: 2017-07-05T14:38:27+02:00
[INFO] Final Memory: 49M/431M
[INFO] -----
```

Figura B.18: Ejecucion del comando “mvn package”.

4. Publicar aplicación en el servidor: para realizar la publicación de la aplicación en una instancia del servidor, debemos crear una instancia si no tenemos ninguna creada. Esta acción creará un directorio para dicha instancia, y dentro de este será donde ubicaremos nuestra aplicación.

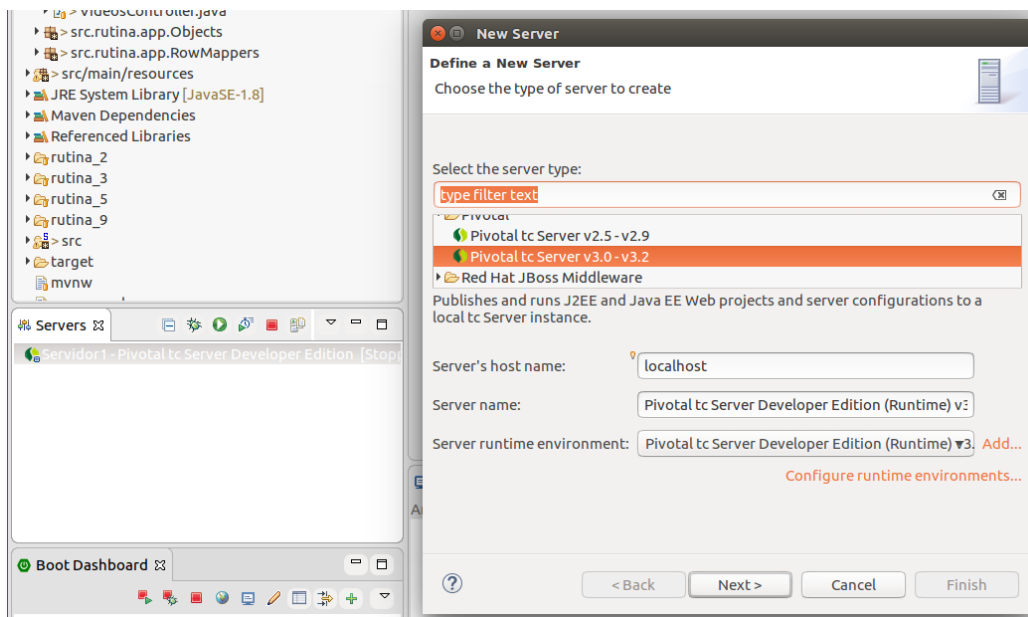


Figura B.17: Crear nueva instancia de Pivotal tc Server.

Una vez creada la instancia del servidor, ubicamos el “.war” de la aplicación, y lo descomprimos en el directorio donde esté escuchando por defecto el servidor. En este caso, sería en el directorio:

```
fran@fran:~/sts-bundle/pivotal-tc-server-developer-3.2.2.RELEASE/Servidor1/webapps/ROOT$ ls
META-INF  org  properties  rutina_app  WEB-INF
fran@fran:~/sts-bundle/pivotal-tc-server-developer-3.2.2.RELEASE/Servidor1/webapps/ROOT$
```

Figura B.18: Directorio donde escucha la instancia del servidor.

Descomprimos el “.war” en este directorio, el cual contiene la siguiente estructura de directorios:

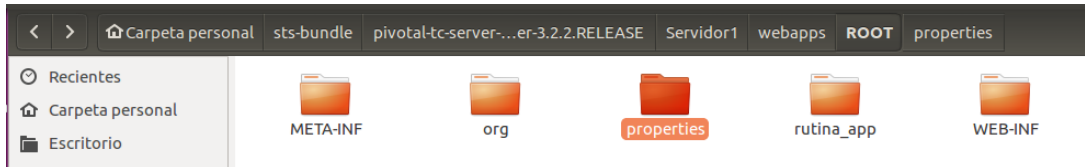


Figura B.19: Servicio Web Desplegado y publicado en el Servidor.

5. Iniciar aplicación desde el servidor: Una vez desplegada la aplicación, podemos iniciar el servidor desde el entorno STS, iniciándose la aplicación y quedando accesible a su página de inicio desde la URL <http://localhost:8080>:

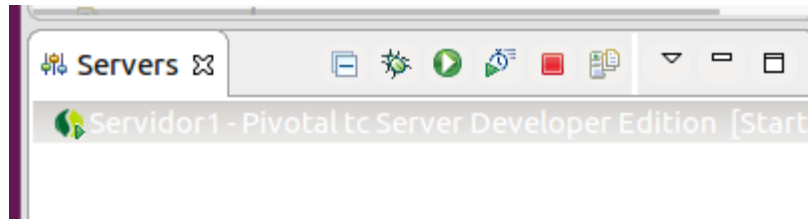


Figura B.20: Botón de “Play” para iniciar la aplicación.

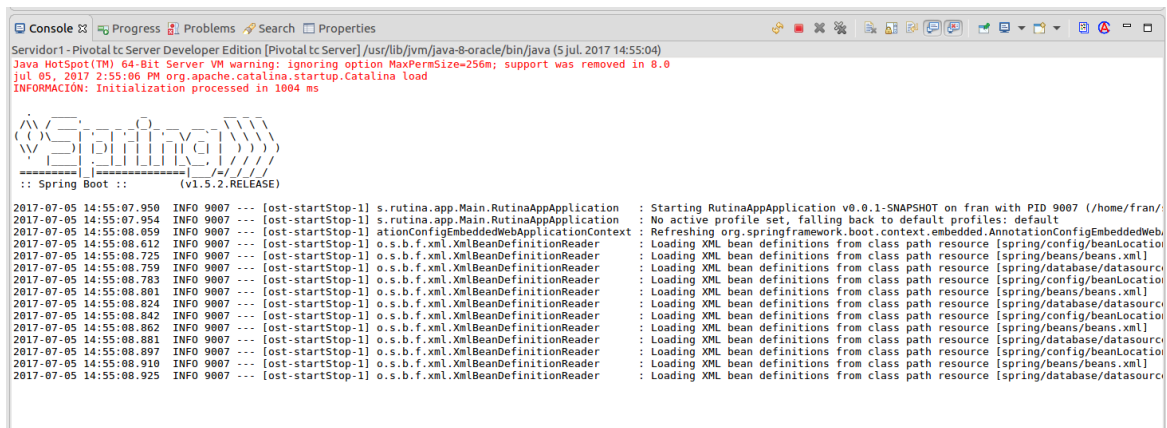


Figura B.21: Proceso de inicialización del Servicio Web.

ANEXO C: DIAGRAMA DE GANTT

En la siguiente imagen, se muestra la planificación del proyecto, donde se observan las siguientes tareas principales: Investigación Teórica, Aplicación Práctica, Documentación.

