

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

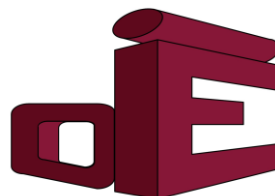
Diseño y programación de una aplicación móvil para
la gestión de información en bares y restaurantes

Autor: Alejandro Mora López

Tutor: Juan Antonio Sánchez Segura

Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



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

Diseño y programación de una aplicación móvil para la gestión de información en bares y restaurantes

Autor:

Alejandro Mora López

Tutor:

Juan Antonio Sánchez Segura

Profesor colaborador

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021

Proyecto Fin de Grado: Diseño y programación de una aplicación móvil para la gestión de información en bares y restaurantes

Autor: Alejandro Mora López

Tutor: Juan Antonio Sánchez Segura

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal

A mi familia
A mis amigos

Resumen

En la actualidad, la transformación digital crece sin pausas en todos los sectores profesionales, la implementación de tecnología para nuevas herramientas y procesos digitales hacen que las empresas y organizaciones obtenga mayor eficiencia y valor para los clientes. Este trabajo presenta una aplicación para dispositivos con sistema operativo Android enfocada al sector hostelero, dónde los clientes de los bares y restaurantes podrán acceder a través de la aplicación a información de interés, como puede ser la ocupación en tiempo real, la información sobre lo que ofrece el establecimiento, las reservas y los datos de carácter general. Esta información podrá se gestionada por el administrador del establecimiento, también a través de la aplicación.

Abstract

Currently, digital transformation grows without pauses in all professional sectors, the implementation of technology for new tools and digital processes make companies and organizations obtain greater efficiency and value for customers. This paper presents an application for devices with Android operating system focused on the hospitality sector, where customers of bars and restaurants can access information of interest through the application, such as occupation in real time, information on what offers the establishment, booking and general data. This information may be managed by the administrator of the establishment, also through the application.

Índice

Resumen	ix
Abstract	xi
Índice	xii
Índice de Tablas	xiv
Índice de Figuras	xvi
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	1
1.3 Alcance	2
1.3.1 Servidor	3
1.3.2 Cliente	3
1.4 Planificación	4
2 Estado del arte	7
2.1 Modelo de negocio y mercado actual	7
2.1.1 Ocupación en tiempo real	7
2.1.2 Herramientas para la gestión de cartas	9
2.1.3 Herramientas para la gestión de reservas	10
2.2 Tecnologías actuales	12
2.2.1 Base de datos	12
2.2.2 Servidor de aplicaciones web	16
2.2.3 Lenguajes de programación del lado del servidor	16
2.2.4 Lenguajes de programación para la aplicaciones móviles	17
3 Tecnología para el proyecto	19
3.1 Alojamiento	19
3.2 Base de datos	19
3.2.1 Administración de la base de datos: phpMyAdmin	20
3.3 Servidor web	20
3.4 Lenguaje de programación en el servidor	20
3.4.1 Administración de desarrollos en el servidor: Atom vía SFTP	20
3.4.2 Pruebas de las diferentes funcionalidades contra el servidor: Postman	20
3.5 Aplicación móvil	21
3.5.1 IDE para el desarrollo: Android Studio	21
3.5.2 Lenguaje de programación: Kotlin	21
4 Diseño e implementación	23
4.1 Servidor	23
4.1.1 Modelo de datos	23
4.1.2 API	36
4.2 Aplicación de usuario	60
4.2.1 Análisis funcional	60
5 Conclusiones	75

6	Líneas futuras	77
6.1	<i>Desarrollo de la aplicación móvil en el sistema operativo iOS</i>	77
6.2	<i>Implementación de seguridad en la comunicación con los servicios</i>	77
6.3	<i>Sincronización y escalabilidad</i>	77
6.4	<i>Despliegue en producción y mantenimiento</i>	78
	Bibliografía	79

ÍNDICE DE TABLAS

Tabla 2–1. Funcionalidades de <i>AppForo Limitado</i>	8
Tabla 2–2. Funcionalidades de <i>Clicker&GO</i>	9
Tabla 2–3. Funcionalidades de <i>QuickTendr</i>	9
Tabla 2–4. Funcionalidades de <i>Sírveme</i>	10
Tabla 2–5. Funcionalidades de <i>Foodeo</i>	10
Tabla 2–6. Funcionalidades de <i>El tenedor</i>	11
Tabla 2–7. Funcionalidades de <i>Opentable</i>	11
Tabla 4–1. Modelo de datos – Tabla de restaurantes y bares	26
Tabla 4–2. Modelo de datos – Tabla de usuarios	28
Tabla 4–3. Modelo de datos – Tabla de horarios	29
Tabla 4–4. Modelo de datos – Tabla de intervalos de las reservas	30
Tabla 4–5. Modelo de datos – Tabla de reservas	31
Tabla 4–6. Modelo de datos – Tabla de cartas	32
Tabla 4–7. Modelo de datos – Tabla de categorías de un plato	33
Tabla 4–8. Modelo de datos – Tabla de platos de una carta	33
Tabla 4–9. Modelo de datos – Tabla de relación de productos fuera de carta	35
Tabla 4–10. API - Servicio registrar usuario	37
Tabla 4–11. API - Servicio autenticar usuario	38
Tabla 4–12. API - Servicio obtener productos carta – Administrador	39
Tabla 4–13. API - Servicio crear producto carta	40
Tabla 4–14. API - Servicio obtener producto	41
Tabla 4–15. API - Servicio editar producto	42
Tabla 4–16. API - Servicio eliminar producto	44
Tabla 4–17. API - Servicio obtener fuera de carta	44
Tabla 4–18. API - Servicio editar fuera de carta	45
Tabla 4–19. API - Servicio obtener horarios	46
Tabla 4–20. API - Servicio actualizar horarios	46
Tabla 4–21. API - Servicio obtener ocupación	47
Tabla 4–22. API - Servicio actualizar ocupación	48
Tabla 4–23. API - Servicio obtener reservas - Administrador	48
Tabla 4–24. API - Servicio crear intervalo	49
Tabla 4–25. API – Servicio obtener intervalo	50
Tabla 4–26. API - Servicio editar intervalo	50
Tabla 4–27. API - Servicio eliminar intervalo	51

Tabla 4–28. API - Servicio crear reserva	52
Tabla 4–29. API – Servicio obtener reserva	52
Tabla 4–30. API - Servicio editar reserva	53
Tabla 4–31. API - Servicio eliminar reserva	53
Tabla 4–32. API – Servicio obtener servicios disponibles	54
Tabla 4–33. API - Servicio editar servicios disponibles	55
Tabla 4–34. API - Servicio obtener establecimientos	56
Tabla 4–35. API - Servicio obtener información	57
Tabla 4–36. API - Servicio obtener productos de carta - Cliente	58
Tabla 4–37. API - Servicio obtener reservas - Cliente	59

ÍNDICE DE FIGURAS

Figura 1-1. Esquema general	2
Figura 1-2. Esquema de bloques de planificación	5
Figura 1-3. Gráfico conceptual del tiempo invertido aproximado	6
Figura 2-1. Gráfico de horas punta <i>Google My Business</i>	8
Figura 2-2. Interfaz de <i>AppForo Limitado</i>	8
Figura 2-3. Interfaz de <i>Clicker&GO</i>	9
Figura 2-4. Interfaz de <i>QuickTendr</i>	9
Figura 2-5. Interfaz de <i>Sírveme</i>	10
Figura 2-6. Interfaz de <i>Foodeo</i>	10
Figura 2-7. Interfaz de <i>El tenedor</i>	11
Figura 2-8. Interfaz de <i>Opentable</i>	11
Figura 2-9. Logo MySQL	12
Figura 2-10. Logo MariaDB	13
Figura 2-11. Logo SQLite	13
Figura 2-12. Logo postgresQL	14
Figura 2-13. Logo Microsoft SQLServer	14
Figura 2-14. Logo Oracle	14
Figura 2-15. Logo Mongo	15
Figura 2-16. Logo Redis	15
Figura 2-17. Logo Cassandra	15
Figura 2-18. Comparación de uso de servidores web	16
Figura 2-19. Comparación de uso de lenguajes en lado del servidor	17
Figura 3-1. Esquema Atom-SFTP-Hosting	20
Figura 3-2. Interfaz de Postman	21
Figura 4-1. Esquema de relación base de datos – Usuarios	24
Figura 4-2. Esquema de relación base de datos – Información de horarios	25
Figura 4-3. Esquema de relación base de datos – Carta	26
Figura 4-4. Esquema de relación base de datos – Reservas	26
Figura 4-5. Diagrama de flujo general de los ficheros de servicios en la API.	37
Figura 4-6. Pantalla principal de OcuSite	61
Figura 4-7. Pantalla de autenticación	61
Figura 4-8. Pantalla de alta de usuario	62
Figura 4-9. Pantalla principal del menú de administración – administrador	62
Figura 4-10. Pantalla de inicio – administrador	63
Figura 4-11. Pantalla de gestión de carta – administrador	63

Figura 4-12. Pantalla de creación de producto – administrador	64
Figura 4-13. Pantalla de edición de producto – administrador	65
Figura 4-14. Pantalla de gestión fuera de carta – administrador	65
Figura 4-15. Pantalla de gestión de ocupación – administrador	66
Figura 4-16. Pantalla de gestión de horarios – administrador	66
Figura 4-17. Pantalla de gestión de reservas – administrador	67
Figura 4-18. Pantalla de creación de intervalo – administrador	68
Figura 4-19. Pantalla de edición de intervalo – administrador	68
Figura 4-20. Pantalla de creación de reserva – administrador	69
Figura 4-21. Pantalla de edición de reserva – administrador	70
Figura 4-22. Pantalla de gestión de servicios disponibles – administrador	70
Figura 4-23. Pantalla de localización - cliente	71
Figura 4-24. Leyenda de marcadores de la pantalla de localización	71
Figura 4-25. Pantalla de información - cliente	72
Figura 4-26. Pantalla de la carta - cliente	72
Figura 4-27. Pantalla de reservas- cliente	73

1 INTRODUCCIÓN

Cada día sabemos más y entendemos menos.

- Albert Einstein -

En este primer capítulo se exponen la motivación, los objetivos, el alcance y la planificación seguida que han servido de guía para la elaboración del proyecto.

1.1 Motivación

España es uno de los países con mayor calidad gastronómica a nivel internacional, muchas personas se dedican a este sector, intentando dar el mejor potencial de la amplia variedad de los productos que existen aquí y una gran experiencia turística y de interior. Son muchas personas las que con frecuencia salen a consumir a bares y restaurantes.

El proyecto actual, pretende ofrecer servicios digitales para facilitar algunos aspectos tanto del cliente a nivel de información, como del administrador del bar o restaurante a nivel de gestión.

1.2 Objetivos

El objetivo principal del proyecto es el diseño, el desarrollo y la puesta en marcha en un entorno de pruebas de una aplicación móvil (android) que abarque de menor a mayor importancia las siguientes funcionalidades:

- Información sobre la ocupación del establecimiento en tiempo real
- Herramienta para la creación de una carta de productos que ofrece el establecimiento.
- Sistema de gestión de reservas en intervalos de tiempos establecidos por el encargado del negocio.
- Información general del establecimiento, horarios, localización y servicios disponibles.

1.3 Alcance

Este trabajo abarcará el funcionamiento correcto de todos los objetivos expuestos en el subapartado anterior. Para ello se implementarán los siguientes componentes y requisitos.

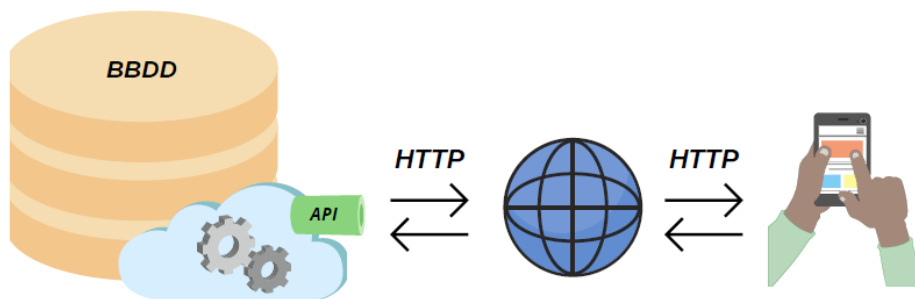


Figura 1-1. Esquema general

Existirán dos tipos de usuario en la aplicación, usuario cliente o usuario común, que será la persona que busca un establecimiento para comer, y usuario administrador, que será el encargado de actualizar la información que le llega al usuario cliente.

El usuario administrador a través de una autenticación en un formulario de usuario y contraseña podrá entrar en su panel de administración para actualizar toda la información.

Una vez dicho esto los requisitos para cumplir los objetivos serán los siguientes:

- Ocupación en tiempo real: el usuario administrador a través de su panel de administración tendrá botones para aumentar y disminuir el número de mesas que hay disponibles en ese momento. A su vez, el usuario cliente tendrá una interfaz basada en el mapa de Google con una serie de marcadores que indicarán, según el color verde, naranja o rojo, si hay suficiente, poco o ningún sitio respectivamente. Los marcadores se establecen en el mapa con las coordenadas de latitud y longitud almacenadas en la base de datos.
- Herramienta para la creación de una carta de productos: el usuario administrador desde su panel de administración tendrá acceso a una pantalla para añadir productos a su carta personalizada, con un campo de categoría que será el que clasifique el producto. Además, el administrador podrá añadir información extra para productos u ofertas a través de una pantalla con varios espacios de área de texto. El cliente común, podrá acceder a la información de la carta a través de la ventana del marcador del mapa, dónde encontrará entre otros, la carta categorizada.
- Sistema de gestión de reservas en intervalos de tiempos: el usuario administrador desde su panel de administración tendrá acceso a una pantalla para añadir intervalos de reservas y reservas dentro de ese intervalo con los datos necesarios. El usuario cliente, dentro de la información del establecimiento podrá ver los espacios disponibles en los intervalos de reserva.
- Información general del establecimiento: el usuario administrador desde su panel de administración tendrá acceso a una pantalla donde podrá activar los servicios que quiera que los usuarios clientes puedan ver, además de una pantalla para actualizar los horarios de apertura del establecimiento. Los servicios que se podrán poner como disponibles, o no, serán:
 - Establecimiento activo: si este no está activado no se podrá acceder a ninguna información.
 - Ocupación activa: si no está activada el marcador en el mapa aparecerá con un color diferente a los anteriores, y mostrará en la ventana de información que la ocupación en tiempo real está

desactivada.

- Carta activa: si no está activada el usuario común cuando haga click en la ventana de información no aparecerá la carta.
- Fuera de carta activa: en principio esta información no será relevante en la interfaz del cliente común para el alcance del proyecto, pero su función más adelante será informar de ofertas u otros productos que no se quieran añadir a la carta.
- Reservas activas: al igual que la carta, si el usuario administrador deshabilita este servicio no estará disponible la sección de información en la interfaz del usuario cliente.
- Servicios relacionados con el pedido del cliente común: es información de interés, indicará la disponibilidad de estos.
 - Comer en local
 - Domicilio
 - Recoger en local

Además de los requisitos mencionados que tendrá la aplicación, se define a continuación el alcance de los componentes del sistema, servidor y cliente.

1.3.1 Servidor

El servidor se encargará de poner a disposición del cliente todos los servicios y recursos necesarios para el correcto funcionamiento de la aplicación final. Puede dividirse a groso modo en tres módulos:

- *Servidor web*. Software encargado de la interpretación de los ficheros escritos en algún lenguaje de programación para exportar recursos concretos a través de la red.
- *Base de datos*. La base de datos del sistema central contendrá los registros de los establecimientos afiliados a la aplicación. Se diseñará un modelo de datos para tratar todos los tipos de datos de la aplicación y las relaciones existentes entre ellos.
- *API*. La API (del original en inglés “*Application Programming Interfaces*”) tratará la lógica de los datos de la aplicación con parámetros recibidos a través del protocolo HTTP (del original en inglés “*Hypertext Transfer Protocol*”) actualizando la información en cada momento. Ofrecerá la puerta de acceso a la base de datos e intercambiará la información necesaria con la aplicación final.

Para cumplir los objetivos marcados el servidor tiene que ser capaz de ofrecer los datos que se soliciten a través del cliente (aplicación móvil en este caso):

- Datos de registro y autenticación de usuario administrador.
- Datos generales del establecimiento, servicios disponibles y horarios.
- Datos de ocupación en tiempo real, las mesas disponibles y las mesas totales.
- Datos relacionados con la carta y los productos de la carta.
- Datos relacionados con intervalos de reserva y las reservas que puedan existir dentro de ese intervalo.

1.3.2 Cliente

Aplicación móvil con una interfaz práctica e intuitiva que conectará con el servidor a través de librerías de programación encargadas de gestionar los diferentes métodos de petición soportados en el protocolo HTTP. Los métodos que se utilizarán son:

- GET. Recuperará datos almacenados en el servidor para ofrecer al usuario algo concreto.
- POST. Enviará una serie de datos causando en el servidor un cambio de estado.

El conjunto de estos métodos hace posible las operaciones de CRUD (de “Crear, Leer, Actualizar y Borrar” del original en inglés “*Create, Read, Update and Delete*”) en la base de datos del servidor.

La aplicación tendrá dos tipos de usuarios, cliente o usuario común y administrador. El primero de ellos no necesitará autenticación y tendrá acceso a la información de los establecimientos que estén registrados y disponibles. El administrador una vez registrado en la base de datos, podrá gestionar la información y hacerla visible, o no, a los clientes.

La aplicación tiene que ser capaz de conseguir la comunicación con el servidor y mostrar los datos solicitados por la persona que maneja la aplicación móvil.

1.4 Planificación

A continuación se enumeran los bloques de trabajo a seguir para la elaboración ordenada y lógica del proyecto:

1. Elección de las tecnologías para el desarrollo.

Para realizar el proyecto se buscarán tecnologías que se usen en la actualidad, a nivel académico y profesional, que sean de libre acceso para cualquier persona. Aunque podrá hacerse uso de servicios de *hosting* para agilizar el proceso de la puesta a punto de un servidor web.

2. Diseño e implementación de un modelo datos.

Para saber en todo momento y tener claro cuáles son y cómo se relacionan las entidades presentes en la aplicación, se hará un modelo de datos lo más simple posible que se amolde correctamente a los casos de uso.

3. Desarrollo e implementación de la aplicación.

Una vez se tengan los dos puntos anteriores se pasará a desarrollar el sistema completo intentando conseguir una arquitectura limpia y clara.

4. Pruebas

Una vez tengamos la aplicación funcionando se harán una serie de pruebas para verificar el correcto funcionamiento de todos los casos de uso que pueden darse para cumplir los objetivos.

Los puntos tres y cuatro entrarán en bucle hasta que se consiga el objetivo final y la aplicación cumpla con lo cometido.

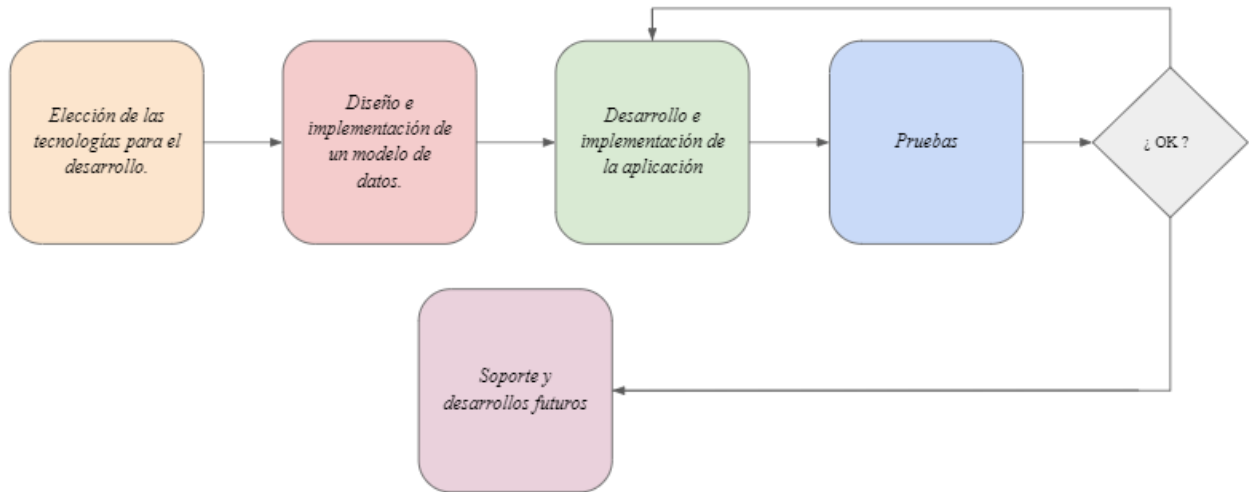


Figura 1-2. Esquema de bloques de planificación

Una aproximación de los tiempos invertidos en cada uno de los bloques en el plan se refleja en el siguiente gráfico circular.

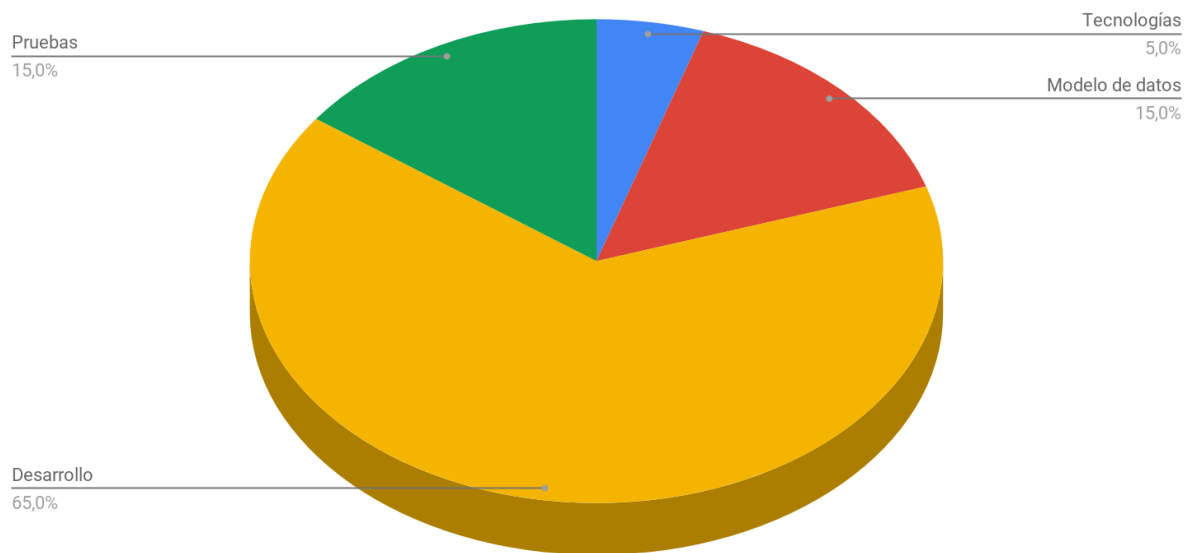


Figura 1-3. Gráfico conceptual del tiempo invertido aproximado

2 ESTADO DEL ARTE

Las máquinas me sorprenden con mucha frecuencia.

- Alan Mathison Turing -

En este capítulo se estudiará y analizará los modelos de negocios, la posición en el mercado y las tecnologías de algunas de las aplicaciones y algunos de los servicios más exitosos en la actualidad.

2.1 Modelo de negocio y mercado actual

El mundo de la gastronomía y la hostelería es muy amplio, es por eso por lo que hay muchas opciones a la hora de implantar un plan de negocio en este sector. La transformación tecnológica está cambiando el sector de forma acelerada. Se hablará en esta sección de los modelos que marcan tendencia actualmente y que están directamente relacionados con la tecnología.

Para dar una estructura a la sección, se evaluarán las aplicaciones o servicios que tengan funcionalidades similares a las marcadas en los objetivos del proyecto, estas aplicaciones o servicios entrarán en la subsección correspondiente dependiendo del propósito principal de las mismas, pero siempre podrá tener más opciones. Por ejemplo puede existir una aplicación que ofrezca servicios de ocupación en tiempo real como propósito principal, pero también tenga otro tipo de servicios.

2.1.1 Ocupación en tiempo real

Lo cierto es que hay pocas aplicaciones que puedan ofrecer este servicio, y más aún en este sector. La dificultad reside en la necesidad de algún mecanismo de actualización que controle el flujo de entrada y salida, ya sean sensores, tornos de entrada y salida o una persona actualizando constantemente el número de persona que hay en el lugar. Con la situación actual en todo el mundo debido a la crisis de la pandemia, este tipo de servicios son demandados por las personas. A continuación se exponen algunos ejemplos que lo ofrecen completa o parcialmente.

Empezaremos por el gigante Google, cuando los usuarios buscan el establecimiento concreto en el buscador, los resultados de las búsquedas te dan una cantidad de datos y referencias difícil de superar gracias a las técnicas de *big data* e inteligencia artificial, y también, a una comunidad que aporta información de los sitios. Concretamente una de las secciones de este grupo de datos se trata una serie de indicaciones acerca de la ocupación y tiempos.



Figura 2-1. Gráfico de horas punta *Google My Business*

Google utiliza la información de usuarios anónimos que tiene habilitado en su móvil el historial de ubicaciones, indicando:

- Gráfico de horas punta: basadas en la popularidad de los últimos meses.
- Datos actualizados de las visitas: muestra el nivel de actividad de la ubicación en ese momento.
- Duración de las visitas: muestra la cantidad de tiempo que los clientes pasan en la ubicación.
- Estimaciones de tiempo de espera: el tiempo de espera aproximado de la clientes.

Una empresa puede tener este tipo de información publicada formando parte de *Google My Business*. Crear una ficha en esta aplicación es gratis, pretende que las empresas físicas también tengan la posibilidad de estar en internet y tener su visibilidad. Como de costumbre *Google* posiciona más alto si la empresa paga por sus anuncios en *Google Ads*.

Otras aplicaciones para la gestión del aforo en España han salido a la luz durante 2020, ofrecen un servicio gratuito pero con limitaciones, de manera que si quieres más funcionalidades tendrás que pagar por ello. Algunas de estas aplicaciones son:

- AppForo Limitado: es una aplicación móvil y web para el registro y control de los aforos en lugares de ocio, creada para hacer frente a la situación generada por la pandemia de la COVID-19. Este sistema de control de accesos en tiempo real está destinado a profesionales y requiere de una licencia y registro previo. Existe una plataforma web donde consultar los aforos de los sitios registrados.

Tabla 2–1. Funcionalidades de *AppForo Limitado*

Opciones	Disponible
Ocupación en tiempo real	✓
Gestión de carta	✗
Gestión de reservas	✗

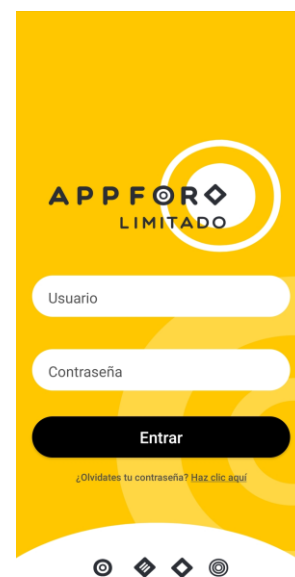


Figura 2-2. Interfaz de *AppForo Limitado*

- *Clicker&GO*: la aplicación tiene el mismo objetivo que la anterior, además te da la opción de contratar sistemas automáticos de control basados en láseres para prescindir de una persona que lo controle.

Tabla 2–2. Funcionalidades de *Clicker&GO*

Opciones	Disponible
Ocupación en tiempo real	✓
Gestión de carta	✗
Gestión de reservas	✗



Figura 2-3. Interfaz de *Clicker&GO*

2.1.2 Herramientas para la gestión de cartas

La necesidad de reducir al máximo el contacto y la expansión de bacterias a través de cartas físicas ha impulsado la digitalización de las mismas. La mayoría de los restaurantes han optado por utilizar la tecnología *QR* para acceder a sus productos, de manera que cuando el cliente llega a la mesa escanea el código y accede a un PDF (normalmente) alojado en una nube, ya sea *Google Drive*, *Microsoft SharePoint* o alguna otra plataforma que permita compartir archivos.

Paralelamente existen algunas aplicaciones impulsadas también por la situación, a continuación se mencionan algunos ejemplos que se encuentra en el mercado:

- *QuickTendr*: las mesas disponen de un código *QR* (*original del inglés "Quick Response"*) que al ser escaneado con la aplicación abre la carta del establecimiento en el dispositivo del cliente, permitiéndoles hacer el pedido y pagarlo al momento. Además, tiene opción de hacer pedidos para llevar o domicilio y revisar el historial de pedidos.

Tabla 2–3. Funcionalidades de *QuickTendr*

Opciones	Disponible
Ocupación en tiempo real	✗
Gestión de carta	✓
Gestión de reservas	✗

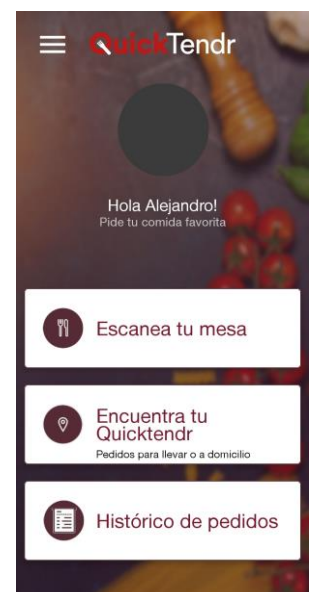


Figura 2-4. Interfaz de *QuickTendr*

- *Sírveme*: ofrece servicios parecidos a *QuickTendr*, como cliente puedes hacer el pedido directamente sin esperar al camarero y los pedidos saldrán a través alguna impresora o pantalla en la cocina. Como administrador del negocio puedes gestionar la carta en tiempo real, recibir los pedidos por impresora, gestión de las mesas y conocer los platos más valorados por los clientes.

Tabla 2-4. Funcionalidades de *Sírveme*

Opciones	Disponible
Ocupación en tiempo real	✗
Gestión de carta	✓
Gestión de reservas	✗

Figura 2-5. Interfaz de *Sírveme*

- *Foodeo*: esta aplicación no ofrece la opción de pagar desde la mesa con el móvil ni tampoco ofrece encargar pedidos para llevar o a domicilio, no obstante, puedes gestionar tu carta y también puedes tener un sistema gestión de reservas, además de comanderos para que los camareros apunten los productos de cada mesa.

Tabla 2-5. Funcionalidades de *Foodeo*

Opciones	Disponible
Ocupación en tiempo real	✗
Gestión de carta	✓
Gestión de reservas	✓

Figura 2-6. Interfaz de *Foodeo*

2.1.3 Herramientas para la gestión de reservas

Reservar antes de ir a un establecimiento es una forma de asegurar a un cliente que va a tener mesa. Con las medidas que deben adoptar la mayoría de los establecimientos referente a la reducción del aforo, la necesidad de reservar es aún mayor, y con ello, la demanda de aplicaciones que ofrezcan este tipo de servicios.

Este tipo de aplicaciones llevan mucho tiempo en el mercado, se hablará de dos aplicaciones con bastante éxito en la gestión de reservas:

- *El tenedor*: es una aplicación que pertenece a *Tripadvisor* especializada en bares y restaurantes, es muy utilizada sobre todo por ofertas especiales que ofrece a los clientes cuando reservan. Por un lado, ofrece un usuario de gestión a cambio de una cuota mensual, y por otro una interfaz amigable a los usuarios clientes con datos masivos de todos los restaurantes que estén registrados en ella. Los clientes tienen muchas referencias a la mano para poder elegir un lugar donde comer.

Tabla 2-6. Funcionalidades de *El tenedor*

Opciones	Disponible
Ocupación en tiempo real	✗
Gestión de carta	✓
Gestión de reservas	✓

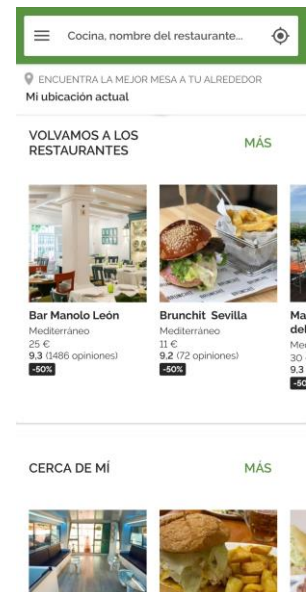


Figura 2-7. Interfaz de *El tenedor*

- *Opentable*: ofrece servicios similares a *El tenedor*, se considera la más utilizada a nivel mundial con una gestión mensual de 29 millones de comensales. Expone la carta, la posibilidad de hacer pedidos para llevar y a domicilio. Además ofrece servicios de informes de estadísticas y guías para mejorar el negocio.

Tabla 2-7. Funcionalidades de *Opentable*

Opciones	Disponible
Ocupación en tiempo real	✗
Gestión de carta	✓
Gestión de reservas	✓

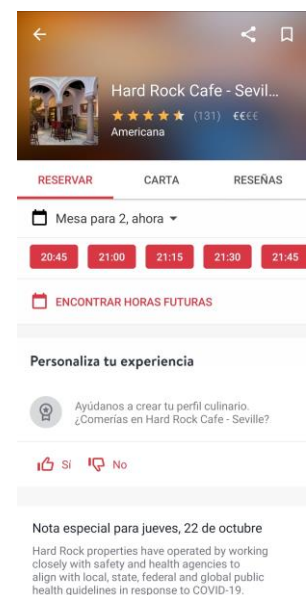


Figura 2-8. Interfaz de *Opentable*

La primera de estas aplicaciones usa un modelo de negocio de cara a las empresas basado en el ofrecimiento de servicios que con ciertos criterios definen unos valores económicos, de cara a los usuarios normales, el modelo de negocio es la publicidad, un usuario que entra en el *El tenedor* para buscar un sitio o alguna información sobre un sitio, se encuentra con anuncios inyectados. *El tenedor* se llevará una comisión si algún cliente pulsa en alguno de estos anuncios que están en la misma interfaz.

Por otro lado, *Opentable* tiene su modelo de negocio orientado únicamente a la prestación de servicios intentando aportar al restaurante clientes de valor, una gran visibilidad de cara a la audiencia mundial y la

inclusión del restaurante en el mundo tecnológico en auge. Las empresas pueden planificar cambios de forma efectiva, gracias a una serie de informes y análisis se pueden tomar mejores decisiones.

2.2 Tecnologías actuales

En esta sección se hace una comparativa general de las tecnologías que existen actualmente en el mercado y que podrían utilizarse en el proyecto.

2.2.1 Base de datos

Existen muchos servidores de bases de datos, se puede hacer una primera clasificación en función de lenguaje, SQL o NoSQL.

A continuación, se citan las diferencias más notables entre ambos tipos de base de datos:

- Aunque pueden hacerlos, las bases de datos NoSQL no suelen utilizar el lenguaje de programación SQL para realizar consultas. Por ejemplo, MongoDB, utiliza JSON.
- Las bases de datos SQL tienen tablas como estructura de almacenamiento, mientras que NoSQL, permite usar otros modelos, como por ejemplo objetos o grafos, es flexible en este aspecto.
- NoSQL tiene limitaciones para realizar operaciones JOIN, debido a la cantidad de datos que puede soportar, a veces puede ser un proceso muy costoso.
- SQL tiene un motor de ejecución por el que pasan todas las operaciones, NoSQL puede usar sistemas distribuidos mediante diferentes mecanismos.

Otro aspecto para tener en cuenta la hora de clasificar un servidor de base de datos es si se trata de un software de *open source*, o bien de un software de pago.

En los siguientes apartados se definen las características más generales de los sistemas gestores de bases de datos más usados hoy en día.

MySQL (SQL)

Se trata de un sistema gestor de base de datos multihilo y multiusuario, el más usado generalmente en las páginas web, es "*open source*" pero también se distribuye en el ámbito empresarial desde que Oracle realizó su compra en 2010.

Es de uso fácil, tiene gran rendimiento, pero su escalabilidad está más limitada cuando hay un gran volumen de datos y ese rendimiento se ve afectado. Soporta SSL y es multiplataforma.



Figura 2-9. Logo MySQL

MariaDB (SQL y NoSQL)

Es una derivación de MySQL después de la compra de Oracle, tiene muchas funcionalidades con las mismas

características más algunas extensiones. Tiene mejoras en motores de búsqueda, gran escalabilidad, es segura, rápida y tiene extensiones con características para base de datos no relacionales. No presenta desventajas notorias frente a MySQL y es de libre distribución.



Figura 2-10. Logo MariaDB

SQLite (SQL)

Se trata de librería desarrollada en C, que implementa un SGBD pero que no lo es. Tiene la ventaja de que es muy portable por la razón de ser una librería, pero por otro lado es poco escalable. Aún así guarda los criterios de atomicidad, consistencia, aislamiento y durabilidad. Es open source y se usa bastante en aplicaciones pequeñas.



Figura 2-11. Logo SQLite

PostgreSQL (SQL)

Se trata de un sistema gestor de base de datos orientado a objetos y de libre distribución. Es muy robusto, eficiente y estable, además es multiplataforma y flexible a diversos lenguajes de programación. Tiene herramientas de administración de base de datos fáciles e intuitivas. La principal desventaja es la administración de las bases de datos que contienen pocos datos a tratar, puesto que se hace pesada la administración debido a que PostgreSQL está optimizada para bases de datos con gran volumen de datos.



Figura 2-12. Logo postgresSQL

Microsoft SQL Server (SQL)

Basado en el lenguaje Transact-SQL, es capaz de soportar grandes volúmenes de datos y servirlos a los usuarios de manera simultánea. Se trata de un software de pago, aunque tiene alguna licencia de libre acceso, pero con muchas limitaciones. Es escalable, estable y seguro, sólo lo soportaba el sistema operativo de Windows hasta hace poco pero ya es compatible con otros sistemas operativos y con Docker. Su sistema gestor de base de datos sólo soporta estructura de tablas relacionales.



Figura 2-13. Logo Microsoft SQLServer

Oracle (SQL)

Software de pago más utilizado por excelencia en el ámbito empresarial, su estabilidad, escalabilidad y gran cantidad de transacciones que soporta las hace estar en el top. Es muy completo y robusto, la gran desventaja al igual que ocurre con la distribución de Microsoft es el precio. Aunque existe alguna versión gratuita, las grandes ventajas las tiene en los formatos de pagos, ofreciendo mejores características en función del tipo de licencia. También trata los datos de forma relacional.



Figura 2-14. Logo Oracle

MongoDB (NoSQL)

Se trata del sistema de base de datos no relacionales más utilizado, la información se trata en ficheros con una estructura llamada BSON (Binary JSON). Empresas como Google o Facebook usan MongoDB para el procesamiento de los datos. Es un sistema con una gran indexación que acelera las búsquedas, es escalable horizontalmente y balancea carga con cierta facilidad. Es un software “open source” con la desventaja de que no se pueden realizar transacciones con cierto nivel de dificultad.



Figura 2-15. Logo Mongo

Redis (NoSQL)

Se basa en un software con almacenamiento de tipo clave-valor, es muy rápido debido a que el uso principal es almacenamiento en caché. Se caracteriza por ser simple, atómico, persistente y multiplataforma. Es open source y no relacional.



Figura 2-16. Logo Redis

Cassandra (NoSQL)

Al igual que Redis tiene un sistema de almacenamiento clave-valor. Utiliza su propio lenguaje llamado CQL (Cassandra Query Language), es un sistema gestor base de datos distribuido, con escalabilidad lineal y horizontal, no relacional y multiplataforma. Es “open source” y lo utilizan plataformas como Netflix o Facebook.



Figura 2-17. Logo Cassandra

2.2.2 Servidor de aplicaciones web

Actualmente el mercado de los servidores está liderado por Apache y Nginx, aunque es cierto que Cloudflare Server está creciendo rápidamente.

La siguiente gráfica sacada de “w3tech” muestra la situación actual de uso de los servidores web.

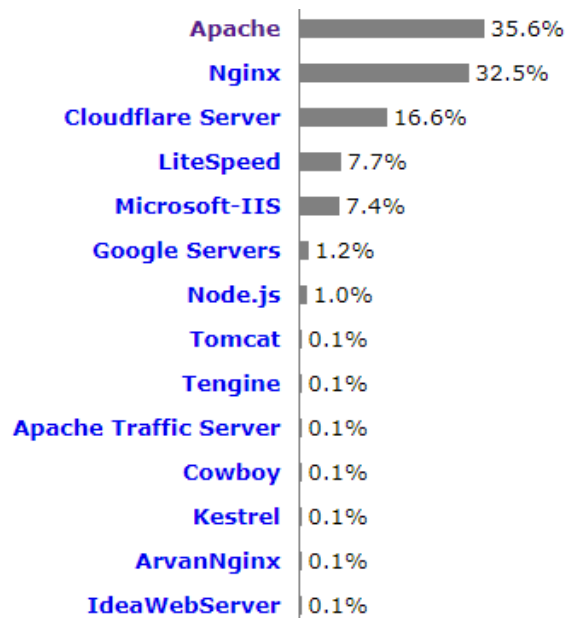


Figura 2-18. Comparación de uso de servidores web

Apache y Nginx no leen aplicaciones escritas en lenguaje Java, pero tienen extensiones o módulos que pueden adaptar para conectarse con un servidor que si pueda interpretar Java. De esta forma, se encargarían de leer script de PHP u otros tipos de lenguaje, mientras que los ficheros que requieran de Java se mandarían al servidor configurado para hacerlo.

Es habitual tener por ejemplo un servidor Apache conectado a un servidor Tomcat, para que este último lea e interprete las fuentes de Java.

2.2.3 Lenguajes de programación del lado del servidor

Aquí existe un claro ganador a la hora de comparar el uso en el mercado de los lenguajes que se ejecutan en el servidor, los ficheros están escritos en este tipo de lenguajes que los servidores interpretan y ejecutan para que el cliente tenga una respuesta en el navegador.

Esta gráfica, también sacada de “w3tech” muestra los lenguajes más utilizados actualmente.

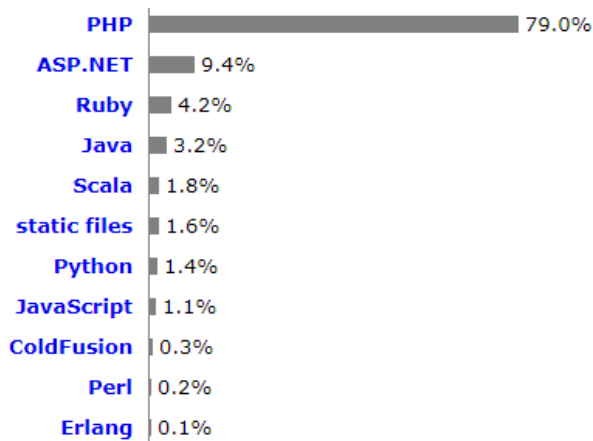


Figura 2-19. Comparación de uso de lenguajes en lado del servidor

Además de los lenguajes de programación de forma nativa, existe gran cantidad de *frameworks* y sistemas de gestión de contenido (CMS, original del inglés “*Content Manage System*”), que te facilitan muchas funcionalidades. Algunos de ellos son muy utilizados:

- CMS:
 - o WordPress
 - o Drupal
 - o Joomla

- Frameworks:
 - o Laravel
 - o Symfony
 - o CodeIgniter

2.2.4 Lenguajes de programación para la aplicaciones móviles

Los principales lenguajes de programación para dispositivos móviles que existen en el mercado son los siguientes:

- Java: es el primer lenguaje que se utilizaba para aplicaciones Android.
- Kotlin: tiene una sintaxis más simple y es compatible completamente con Java, además desde hace poco, el lenguaje principal de Android.
- Swift: es el lenguaje principal para desarrollar aplicaciones en el sistema iOS.
- Dart: usa Flutter para crear aplicaciones nativas en iOS y Android desde una sola base de código.
- JavaScript: con ayuda de frameworks como React o Ionic también se pueden crear aplicaciones móviles con este lenguaje.
- C#: creado por Microsoft, usa Xamarin para el desarrollo móvil y ejecuta el código para iOS o Android.

3 TECNOLOGÍA PARA EL PROYECTO

Antes de que un software sea reutilizable debería ser utilizable.

- Ralph Johnson -

En este capítulo se indican las tecnologías utilizadas para el desarrollo de la aplicación, *IDE* o *IDEs* utilizados, lenguajes de programación, software para la realización de las pruebas o cualquier otro software u otra aplicación utilizada en el proceso de desarrollo en cada uno de los módulos en los que se ha dividido.

3.1 Alojamiento

El alojamiento, *hosting* en inglés, es un servicio que provee a usuario un sistema o entorno para poder almacenar todo tipo de información y dar acceso a la misma a través de la red. Hay muchas compañías en internet que ofrecen alojamientos con muchísimas utilidades ya creadas y facilitan mucho al desarrollador, normalmente se busca en función de las necesidades del proyecto.

Existen otros tipos de alternativas, como puede ser la configuración de todo el entorno que sostiene la aplicación, sin necesidad de tener que recurrir a terceros para ello. Por ejemplo se podría utilizar una raspberry e instalar y configurar todos los servicios necesarios “*open source*” para poner en marcha el alojamiento del proyecto.

Independientemente del alojamiento que se escoja será indiferente para el proyecto actual, puesto que las tecnologías seleccionadas son perfectamente migrables a cualquier entorno.

Para realizar las pruebas de la aplicación se hará uso de un hosting, cuyos servicios fueron adquiridos por la persona que realiza el proyecto para fines diversos, entre ellos las pruebas de la aplicación de este proyecto.

3.2 Base de datos

La elección de la base de datos se centra en tres de los nueve SGBD que se han visto en el capítulo anterior. Por ser los más utilizados y por ser de libre distribución, el debate para la elección ha sido entre MongoDB, MariaDB y PostgreSQL.

En este caso el *hosting* trae por defecto la base de datos de MariaDB, es por eso que se usará este software, como el modelo de datos será relacional, la base de datos de MongoDB no es posible usarla, pero PostgreSQL es otra opción posible, no obstante se usará MariaDB.

3.2.1 Administración de la base de datos: phpMyAdmin

El *hosting* también incorpora phpMyAdmin con la base de datos, esto hace más sencillo la administración de la misma. Se trata de una interfaz sencilla e intuitiva que te muestra la base de datos, sus tablas y sus registros, además se pueden gestionar bastantes funciones.

3.3 Servidor web

El servidor de aplicaciones web se encarga de exportar el contenido del usuario propietario de la web, interpretando los diferentes ficheros que se encuentren en el directorio principal y directorios hijos. En el caso de este proyecto el contenido que tendrá que exportar serán objetos en formato JSON, que dependiendo de la función solicitada a la API, será de una manera o de otra.

El *hosting* también ofrece por defecto el servidor web *LiteSpeed*, suficiente para el propósito del Proyecto, no obstante también se usará *Apache* para hacer pruebas en un entorno local.

3.4 Lenguaje de programación en el servidor

Para el desarrollo de la API se utilizará PHP. Este lenguaje está muy extendido, hay muchas herramientas y documentación disponible para poder solucionar de manera rápida cualquier problema que pueda surgir.

3.4.1 Administración de desarrollos en el servidor: Atom vía SFTP

Existen muchos editores y entornos para programar, algunos que te ayudan más y otros que son más simples. En este proyecto se utilizará Atom para realizar los desarrollos de la API, tiene una extensión sencilla para la conexión con el repositorio remoto a través del protocolo SFTP (original del inglés “*Secure File Transfer Protocol*”) y hace la experiencia para las pruebas de los desarrollos bastante cómoda.

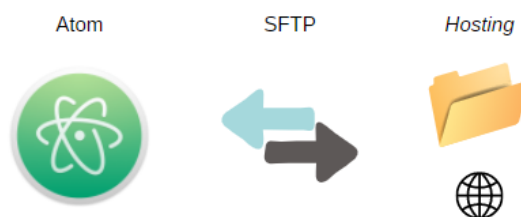


Figura 3-1. Esquema Atom-SFTP-Hosting

3.4.2 Pruebas de las diferentes funcionalidades contra el servidor: Postman

Para probar los diferentes servicios desarrollados en el servidor y corroborar las respuestas que se van a recibir en la aplicación móvil, primero se hacen pruebas contra el servidor. Para ello se utiliza Postman, dónde se podrán construir fácilmente las solicitudes necesarias para probar las funciones de CRUD.

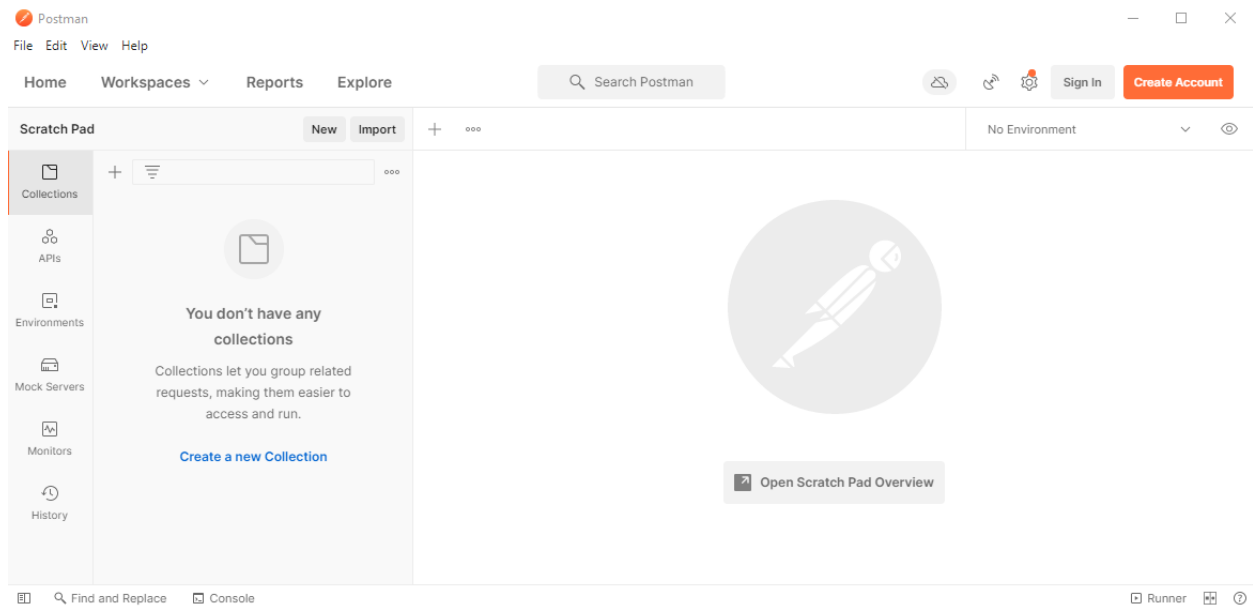


Figura 3-2. Interfaz de Postman

3.5 Aplicación móvil

El objetivo del proyecto es una aplicación móvil para dispositivos Android, por lo tanto esto filtra y deja la elección del lenguaje entre los oficiales de Android, Java y Kotlin.

3.5.1 IDE para el desarrollo: Android Studio

El entorno de desarrollo será Android Studio, que es por excelencia el entorno oficial de los desarrolladores de aplicaciones en los sistemas operativos de Android.

3.5.2 Lenguaje de programación: Kotlin

Kotlin será el lenguaje de programación que se utilizará para el desarrollo de la aplicación. El principal motivo es el aprendizaje del mismo, además es el lenguaje preferido por Google para las aplicaciones en Android, es un lenguaje basado en Java y con una estructura sintáctica más atractiva.

4 DISEÑO E IMPLEMENTACIÓN

El ordenador nació para resolver problemas que antes no existían.

- Bill Gates -

En este capítulo se detallarán los procedimientos que se realizarán para cumplir los objetivos del proyecto. En primer lugar, se comenzará comentando la parte del servidor, cómo se impletará el modelo de datos diseñado previamente analizando las relaciones entre las tablas y su contenido. Seguidamente, se explicará la lógica de los datos en función de los posibles parámetros de entrada a través de los métodos http. Y por último en la parte del servidor, se aclararán de forma resumida la forma de impletar los diferentes servicios ofrecidos por la API para su uso en la parte del cliente.

Una vez haya terminado la parte correspondiente al servidor, se detallará el análisis funcional y el diseño técnico de cada una de las actividades disponibles en la aplicación.

Para terminar este apartado se informará al lector de los pasos seguidos para una prueba de integración completa de todos los componentes necesarios.

4.1 Servidor

4.1.1 Modelo de datos

Esta subsección contiene el modelo gráfico y las características en forma de tabla de cada una de las tablas del modelo, valga la redundancia.

4.1.1.1 MySQL o MariaDB

Puntos para tener en cuenta para un buen rendimiento:

- *Redundancia de datos.* Se trata de la duplicidad de registros o datos que ya se encuentran en base de datos.
- *Inconsistencia de datos.* Producida por lo anterior, por ejemplo, cuando alguna información es ambigua en más de un registro.
- *Integridad de los datos.* Cuando se trabaja con bases de datos, generalmente los datos se reparten entre varios ficheros. Sí, como pasa con MariaDB, la base de datos está disponible para varios usuarios de forma simultánea, deben existir mecanismos que aseguren que las interrelaciones entre registros se mantienen coherentes, que se respetan las dependencias de existencia y que las claves únicas no se repitan.

Los problemas de integridad se suelen producir cuando varios usuarios están editando datos de la misma base de datos de forma simultánea. Por ejemplo, un usuario crea un nuevo registro, mientras

otro edita uno de los existentes, y una tercera borra otro. El *DBMS* debe asegurar que se pueden realizar estas tareas sin que se produzcan errores que afecten a la integridad de la base de datos.

Antes de modelar hay que tener una serie de requisitos para saber qué vamos a desarrollar. Los objetivos de funcionamiento ya están claros, dejando de lado la parte de la interfaz de usuario.

4.1.1.2 Esquemas de relación

- Un establecimiento sólo puede tener un usuario administrador.

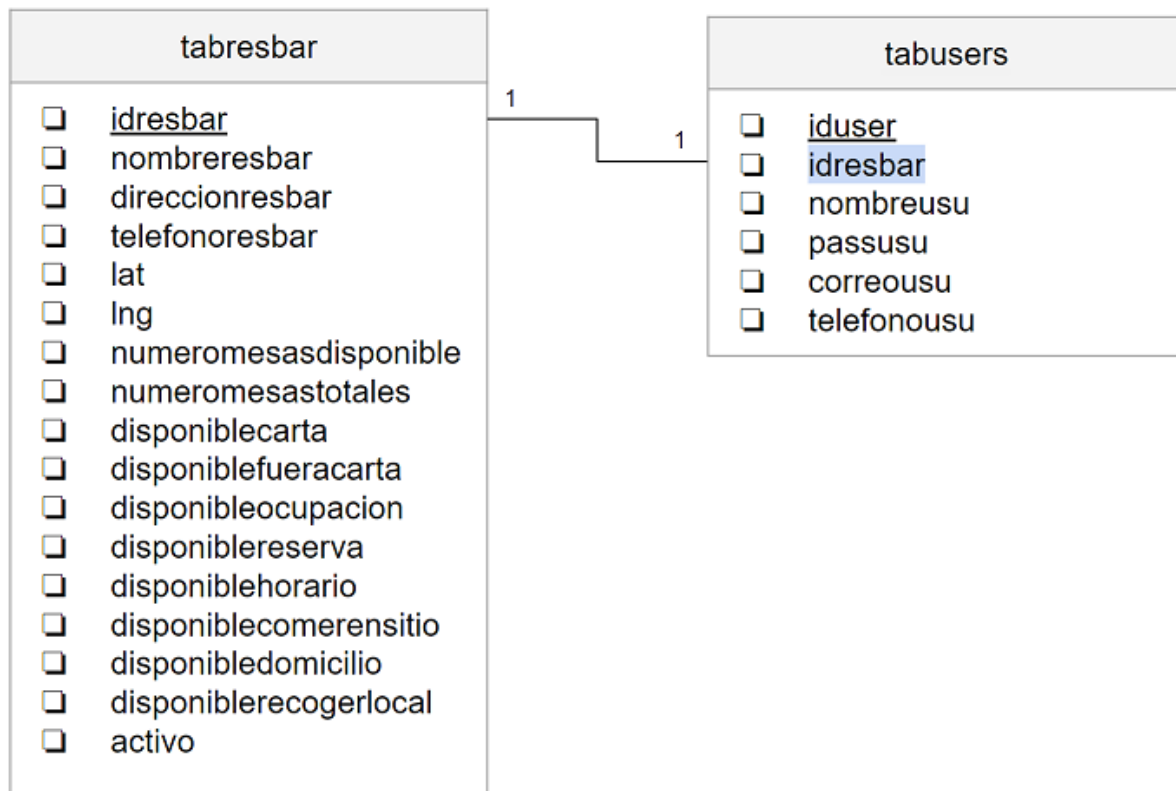


Figura 4-1. Esquema de relación base de datos – Usuarios

- Un establecimiento solo puede tener asociado un registro de información sobre los horarios de apertura y cierre del mismo.

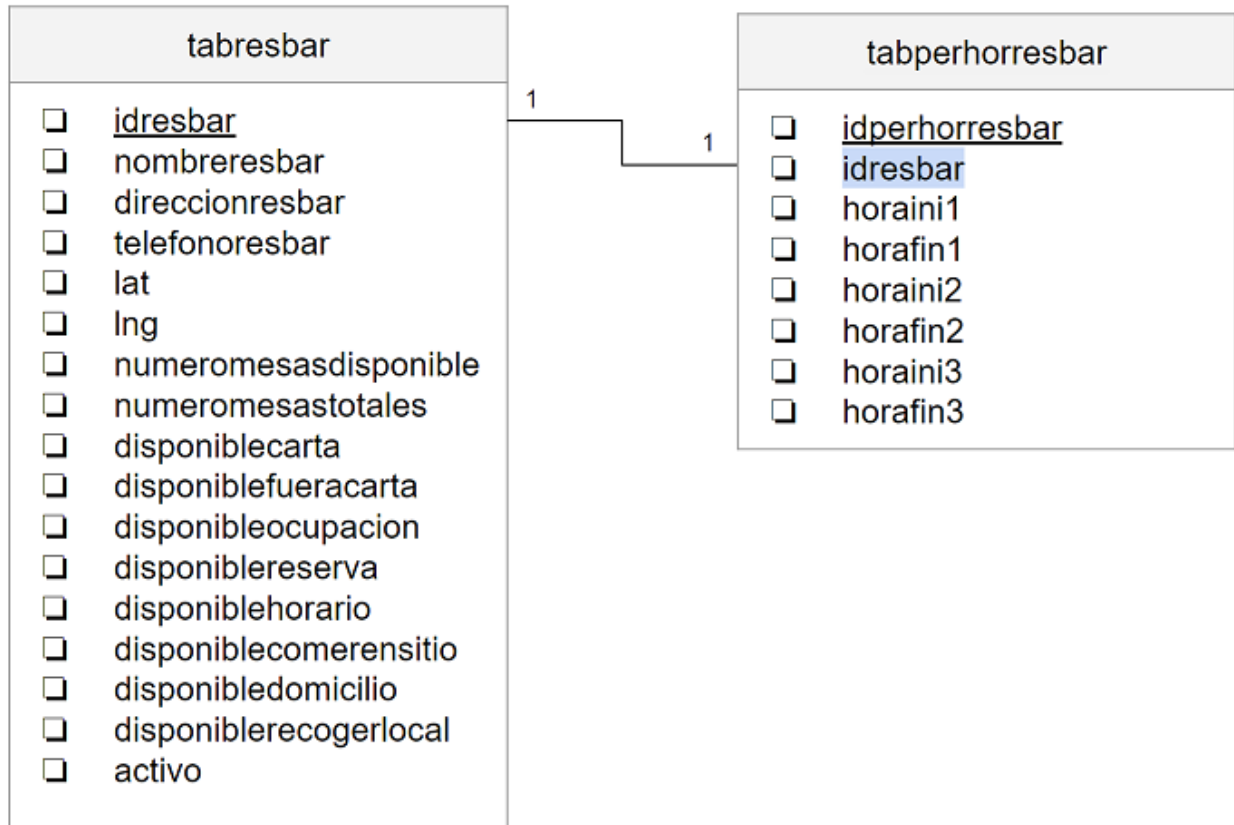


Figura 4-2. Esquema de relación base de datos – Información de horarios

- Un establecimiento puede tener una carta.
- Una carta puede tener muchos platos o productos.
- Un plato o producto pertenece a una categoría.

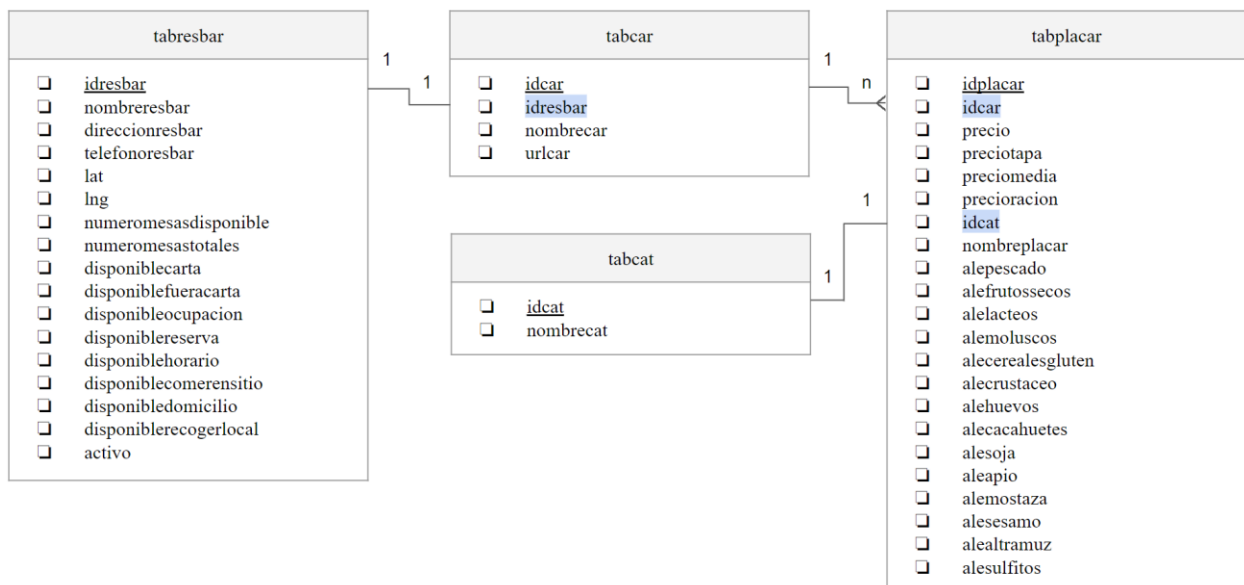


Figura 4-3. Esquema de relación base de datos – Carta

- Un restaurante puede tener varios intervalos horarios habilitados para la reserva.
- Un intervalo horario puede tener varias reservas.

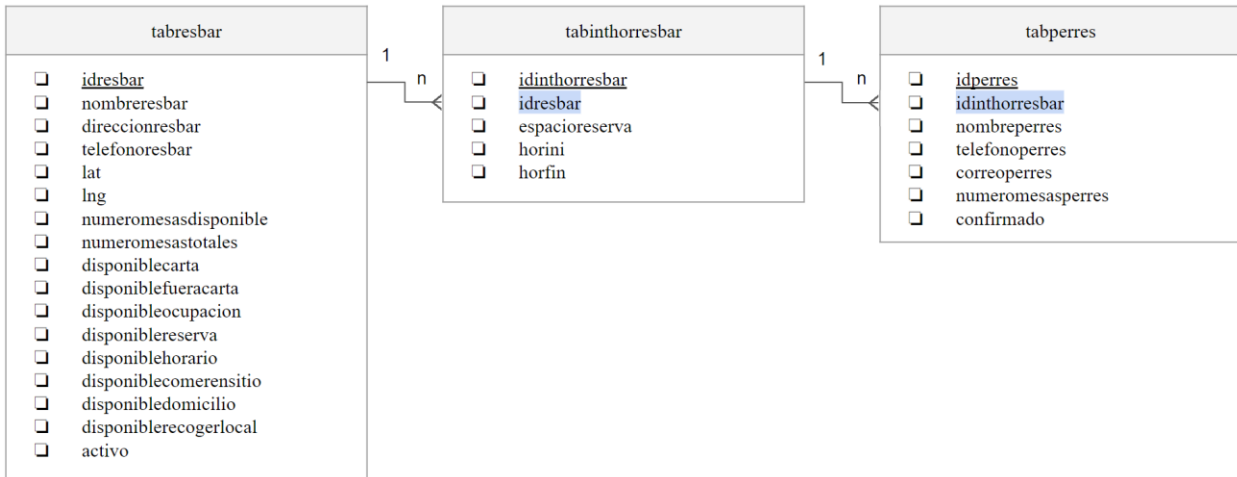


Figura 4-4. Esquema de relación base de datos – Reservas

Las tablas que tienen relación uno a uno pueden fusionarse reduciendo así el número de tablas, no se ha hecho de esa manera para que la estructura de la base de datos tenga un orden ligado a las funcionalidades, además en una supuesta base de datos con tablas de muchos registros la búsqueda de datos concretos sería más rápida de esta manera.

De esta forma el modelo esquemático de la base de datos refleja el bloque de la relación de usuarios, el bloque de los horarios de apertura y cierre, el bloque de la carta del establecimiento y el bloque de las reservas. Toda la información de servicios disponibles y ocupación en tiempo real está en la tabla principal (tabresbar).

4.1.1.3 Tablas

En este apartado se detallan todos los campos de cada una de las tablas que aparecen en el modelo de datos. Para orientar un poco al lector, la nomenclatura seguida a la hora de dar nombre en algunas de las tablas, algunos de los campos y algunas de las relaciones o restricciones es la siguiente:

Se cogen las tres primeras letras de cada uno de los nombres de las entidades que participan en la tabla, por ejemplo en el caso de que la entidad fuera un gato, su tabla se nombraría tabgat. Además, todos los nombres irán en minúsculas, que será la forma utilizada en las consultas a la hora de escribir código.

La columna con el nombre PK (clave primaria, del original “Primary Key”) indica la clave primaria de la tabla.

4.1.1.3.1 Tabla de restaurantes o bares

Tabla 4-1. Modelo de datos – Tabla de restaurantes y bares

Tabla	
Nombre	tabresbar

Alias		resbar			
Descripción		Entidad que gestiona la información referente a los restaurantes o bares que se den de alta en la aplicación.			
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	idresbar	int	Identificador del restaurante o bar.	11	✗
	nombresbar	varchar	Nombre del establecimiento.	80	✗
	direccionresbar	varchar	Dirección del establecimiento.	100	✗
	telefonoresbar	varchar	Teléfono del establecimiento	12	✗
	lat	varchar	Latitud geográfica.	9	✓
	lng	varchar	Longitud geográfica.	9	✓
	numeromesasdisponibles	varchar	Número de mesas disponibles en tiempo real.	5	✗
	numeromesastotales	varchar	Número de mesas totales para ocupar.	5	
	disponiblecarta	varchar	Flag para la visibilidad de la opción de carta S/N.	1	✗
	disponiblefuera de carta	varchar	Flag para la visibilidad de la opción fuera de carta S/N.	1	✗
	disponibleocupacion	varchar	Flag para la visibilidad de la ocupación S/N.	1	✗
	disponiblereserva	varchar	Flag para la visibilidad de las opciones de reservas S/N.	1	✗
	disponiblehorario	varchar	Flag para la visibilidad del horario del establecimiento S/N.	1	✗
	disponiblecomerensitio	varchar	Flag para el servicio de comer en el establecimiento S/N.	1	✗
	disponibledomicilio	varchar	Flag para el servicio de domicilio S/N.	1	✗
	disponiblerecogerlocal	varchar	Flag para el servicio de recoger pedido en el establecimiento S/N.	1	✗
	activo	varchar	Flag para indicar si el restaurante o bar está activo en la aplicación S/N.	1	✗
Otras restricciones					

Nombre	Campo/s	Tipo de restricción	Efecto
mapa	lat, lng	Unicidad	No puede haber más de un punto geográfico con las mismas coordenadas.

4.1.1.3.2 Tabla de usuarios

Tabla 4-2. Modelo de datos – Tabla de usuarios

Tabla					
Nombre	tabusers				
Alias	users				
Descripción	Contiene información referente a los usuarios registrados en los establecimientos.				
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	iduser	int	Identificador del usuario.	11	×
	idresbar	int	Identificador del restaurante o bar al que pertenece el intervalo de reservar.	11	×
	nombreusu	varchar	Nombre de usuario.	15	×
	passusu	varchar	Contraseña de usuario.	15	×
	correousu	varchar	Correo electrónico del usuario.	50	×
	telefonousu	varchar	Teléfono del usuario.	12	×
Relaciones					
Campo	Campo referenciado	Efecto			
idresbar	tabresbar.idresbar	Para crear un usuario tiene que existir un establecimiento al que asignarlo.			
Otras restricciones					
Nombre	Campo/s	Tipo de restricción	Efecto		
userforidresbar	idresbar	Unicidad	Sólo puede existir un usuario por establecimiento (el		

			administrador).
userforresbar	nombreusu	Unicidad	El nombre de usuario es único.
idx_idresbarinusers	idresbar	Índice	Acelera la búsqueda en base de datos.

4.1.1.3.3 Tabla de horarios

Tabla 4-3. Modelo de datos – Tabla de horarios

Tabla					
Nombre		tabperhorresbar			
Alias		perhor			
Descripción		Entidad que contiene la información referente a los horarios de los restaurantes o bares.			
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	idperhorresbar	int	Identificador del resgistro de horarios del establecimiento.	11	✗
	idresbar	int	Identificador del restaurante o bar al que pertenece el resgistro del horario	11	✗
	horaini1	varchar	Hora de inicio del primer periodo.	5	✓
	horafin1	varchar	Hora de fin del primer periodo.	5	✓
	horaini2	varchar	Hora de inicio del segundo periodo.	5	✓
	horafin2	varchar	Hora de fin del segundo periodo.	5	✓
	horaini3	varchar	Hora de inicio del tercer periodo.	5	✓
	horafin3	varchar	Hora de fin del tercer periodo.	5	✓
Relaciones					
Campo	Campo referenciado	Efecto			
idresbar	tabresbar.idresbar	Para crear un registro horario tiene que existir un			

		establecimiento al que asignarlo.	
Otras restricciones			
Nombre	Campo/s	Tipo de restricción	Efecto
perhorforresbar	idresbar	Unicidad	Sólo puede existir un registro de horario por establecimiento.
idx_idresbarperhorresbar	idresbar	Índice	Acelera la búsqueda en base de datos.

4.1.1.3.4 Tabla de intervalos de las reservas

Tabla 4-4. Modelo de datos – Tabla de intervalos de las reservas

Tabla					
Nombre	tabinthorresresbar				
Alias	inthorres				
Descripción	Entidad que contiene la información referente a los intervalos horarios de reservas disponibles en el establecimiento.				
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	idinthorresresbar	int	Identificador del registro de horarios del establecimiento.	11	✗
	idresbar	int	Identificador del restaurante o bar al que pertenece el intervalo de reservar.	11	✗
	espaciosreserva	varchar	Número de espacios disponibles para este intervalo horario.	5	✗
	horini	varchar	Hora de inicio del intervalo.	5	✗
	horfin	varchar	Hora de fin del intervalo.	5	✗
Relaciones					
Campo	Campo referenciado	Efecto			

idresbar	tabresbar.idresbar	Para crear un registro de reserva tiene que existir un establecimiento al que asignarlo.		
Otras restricciones				
Nombre	Campo/s	Tipo de restricción	Efecto	
inthorresforresbar	Idresbar, horini, horfin	Unicidad	Sólo puede existir un registro de reserva en un mismo intervalo horario.	
idx_idresbarperhorresbar	idresbar	Índice	Acelera la búsqueda en base de datos.	

4.1.1.3.5 Tabla de reservas

Tabla 4-5. Modelo de datos – Tabla de reservas

Tabla					
Nombre	tabperres				
Alias	perres				
Descripción	Entidad que contiene la información referente a las reservas realizadas en un intervalo horario.				
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	idperres	int	Identificador de la reserva realizada.	11	✗
	idinthorresresbar	int	Identificador del intervalo horario al que pertenece la reserva.	11	✗
	nombperres	varchar	Número de espacios disponibles para este intervalo horario.	50	✗
	telefonperres	varchar	Teléfono de la persona que reserva.	12	✗
	correoperres	varchar	Correo electrónico de la persona que reserva.	50	✗
	numeromesasperres	varchar	Número de mesas reservadas.	2	✗

	confirmado	varchar	Flag para confirmar la reserva S/N.	1	×
Relaciones					
Campo	Campo referenciado		Efecto		
idinthorresresbar	tabinthorresresbar.idinthorresresbar		Para crear un registro de reserva tiene que existir un intervalo al que asignarlo.		
Otras restricciones					
Nombre	Campo/s	Tipo de restricción	Efecto		
telperresforinthorres	idinthorresresbar, telefonoperres	Unicidad	Sólo puede existir un registro de reserva para el mismo teléfono.		
idx_idinthorresresbarinperres	idinthorresresbar	Índice	Acelera la búsqueda en base de datos.		

4.1.1.3.6 Tabla de cartas

Tabla 4-6. Modelo de datos – Tabla de cartas

Tabla					
Nombre	tabcar				
Alias	car				
Descripción	Entidad que relaciona la carta con el establecimiento.				
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	idcar	int	Identificador de la carta del establecimiento.	11	×
	idresbar	int	Identificador del restaurante o bar al que pertenece la carta.	11	×
	nombrecar	varchar	Nombre de la carta.	60	✓
	urlcar	varchar	URL de la carta (si requiere).	100	✓
Relaciones					

Campo	Campo referenciado	Efecto	
idresbar	tabresbar.idresbar	Para crear una carta tiene que existir un establecimiento al que asignarla.	
Otras restricciones			
Nombre	Campo/s	Tipo de restricción	Efecto
carforres	idresbar	Unicidad	Sólo puede existir crear una carta por establecimiento.
idx_idresbarincar	idresbar	Índice	Acelera la búsqueda en base de datos.

4.1.1.3.7 Tabla de categorías de un producto

Tabla 4–7. Modelo de datos – Tabla de categorías de un plato

Tabla					
Nombre	tabcat				
Alias	cat				
Descripción	Entidad que contiene la información de las categorías a las que puede pertenecer un plato de una carta.				
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	idcat	int	Identificador de la categoría.	11	✗
	nombrecat	varchar	Nombre de la categoría	20	✗

4.1.1.3.8 Tabla de platos de una carta

Tabla 4–8. Modelo de datos – Tabla de platos de una carta

Tabla	
Nombre	tabplacar
Alias	placar
Descripción	Entidad que contiene la información referente a un producto que contiene la carta

		del establecimiento.			
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	idplacar	int	Identificador del plato de la carta.	11	✗
	idcar	int	Identificador del restaurante o bar al que pertenece el intervalo de reservar.	11	✗
	precio	varchar	Precio general del plato	6	✓
	preciotapa	varchar	Precio del plato de tamaño ‘tapa’ (si requiere).	6	✓
	preciomedia	varchar	Precio del plato de tamaño ‘media’ (si requiere).	6	✓
	precioracion	varchar	Precio del plato de tamaño ‘ración’ (si requiere).	6	✓
	idcat	int	Identificador de la categoría.	11	✗
	nombrepalcar	varchar	Nombre del plato.	200	✗
	alepescado	varchar	Flag para el alérgeno - pescado S/N	1	✗
	alefrutossecos	varchar	Flag para el alérgeno – frutos secos S/N	1	✗
	alelacteos	varchar	Flag para el alérgeno – lácteos S/N	1	✗
	alemoluscos	varchar	Flag para el alérgeno – moluscos S/N	1	✗
	alecerealesgluten	varchar	Flag para el alérgeno – cereales con gluten S/N	1	✗
	alecrustaceo	varchar	Flag para el alérgeno - crustáceos S/N	1	✗
	alehuevos	varchar	Flag para el alérgeno - huevos S/N	1	✗
	alecacahuetes	varchar	Flag para el alérgeno - cacahuetes S/N	1	✗
	alesoja	varchar	Flag para el alérgeno - soja S/N	1	✗
	aleapio	varchar	Flag para el alérgeno - apio S/N	1	✗
	alemostaza	varchar	Flag para el alérgeno - mostaza S/N	1	✗
	alesesamo	varchar	Flag para el alérgeno - sésamo S/N	1	✗
	alealtramuz	varchar	Flag para el alérgeno - altramuz S/N	1	✗
	alesulfitos	varchar	Flag para el alérgeno - sulfitos S/N	1	✗

Relaciones			
Campo	Campo referenciado	Efecto	
idcar	tabcar.idcar	Para crear un plato tiene que existir una carta a la que asignarle ese plato.	
idcat	tabcat.idcat	El plato tiene que pertenecer a una categoría para su clasificación.	
Otras restricciones			
Nombre	Campo/s	Tipo de restricción	Efecto
idx_idcarinplacar	idcar	Índice	Acelera la búsqueda en base de datos.
idx_idcatinplacar	idcat	Índice	Acelera la búsqueda en base de datos.

4.1.1.3.9 Tabla de relación de productos fuera de carta

Tabla 4-9. Modelo de datos – Tabla de relación de productos fuera de carta

Tabla					
Nombre	tabfuecar				
Alias	fuecar				
Descripción	Entidad que relaciona la información fuera de carta con el restaurante o bar.				
Campos					
PK	Nombre	Tipo	Descripción	Tamaño	Nulo
✓	idfuecar	int	Identificador del registro de información fuera de carta.	11	✗
	idresbar	int	Identificador del restaurante o bar al que pertenece la información fuera de carta.	11	✗
	space1	varchar	Espacio para informar.	100	✓
	space2	varchar	Espacio para informar.	100	✓
	space3	varchar	Espacio para informar.	100	✓
	space4	varchar	Espacio para informar.	100	✓

	space5	varchar	Espacio para informar.	100	✓
	space6	varchar	Espacio para informar.	100	✓
	space7	varchar	Espacio para informar.	100	✓
Relaciones					
Campo		Campo referenciado	Efecto		
idresbar		tabresbar.idresbar	Para crear información independiente de la carta tiene que existir un establecimiento al que asignarla.		
Otras restricciones					
Nombre	Campo/s	Tipo de restricción	Efecto		
fuecarforresbar	idresbar	Unicidad	Sólo puede existir un tipo de registro fuera de carta en cada restaurante o bar.		
idx_idresbarincar	idresbar	Índice	Acelera la búsqueda en base de datos.		

4.1.2 API

En la siguiente subsección se detalla la funcionalidad de cada uno de los servicios desarrollados en la *API*. Cada uno de los servicios de la *API* está definido en un fichero con extensión *php*. Los servicios se clasificarán según el usuario, por lo tanto primero se detallan los servicios relacionados con los usuarios administradores y después los servicios de los usuarios clientes, este orden es debido a que los administradores gestionan el sitio para los clientes, y en el flujo de trabajo de la aplicación en general primero está la creación de un usuario-restaurante y los datos correspondientes. Dividimos por tanto en dos apartados esta subsección, *API* administrador y *API* cliente.

Como se ha comentado en el capítulo 3, la información que devuelve la *API* va en formato *JSON*, teniendo en cuenta esto y los métodos que se van a usar en las solicitudes *HTTP* (GET y POST), los ficheros de todos los servicios seguirán la misma estructura. A continuación se expone un diagrama de flujo que representa esta estructura.

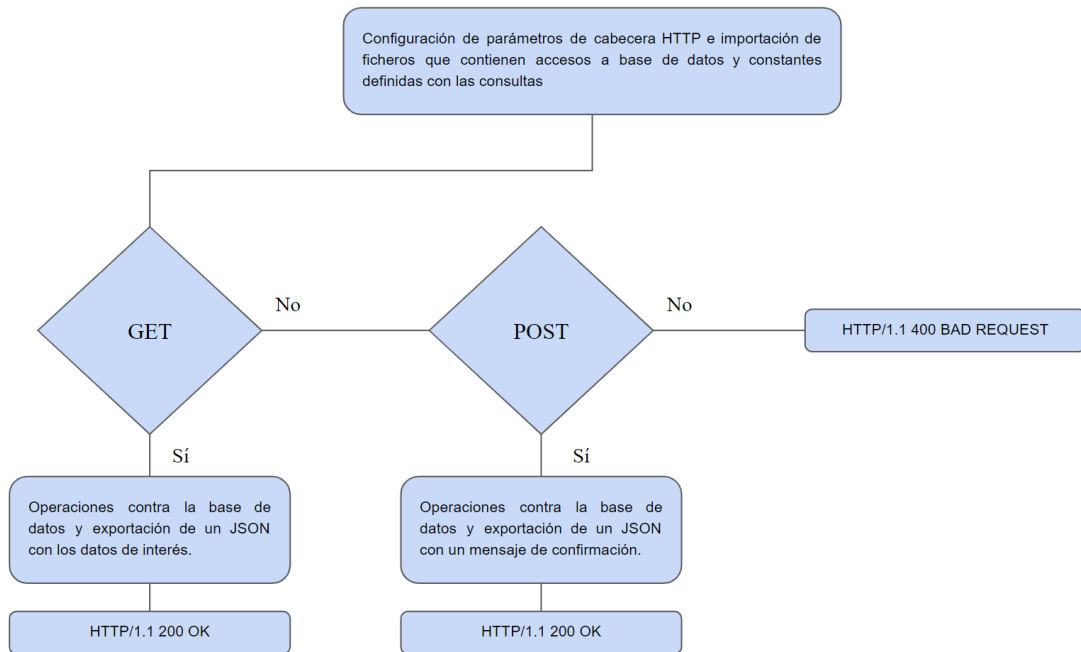


Figura 4-5. Diagrama de flujo general de los ficheros de servicios en la API.

El siguiente conjunto de tablas recoge los siguientes datos relacionados con el servicio:

- Nombre del servicio.
- Fichero dónde se ubica.
- Funcionalidad del servicio.
- Método HTTP que utiliza.
- Endpoint, es la URL del servicio.
- Solicitud, tendrá los parámetros necesarios y la descripción de cada uno.
- Respuesta, las posibles respuestas del servidor utilizando formato JSON.

4.1.2.1 API usuario administrador

4.1.2.1.1 Servicio de alta de usuario

Tabla 4–10. API - Servicio registrar usuario

Servicio	
Nombre del fichero	registroservice.php
Funcionalidad	Registra un establecimiento con una serie de campos de información por defecto y un usuario con las credenciales que haya introducido en los campos del formulario
Método	POST

Endpoint	localhost/ocusite/admin/registroservice.php
Solicitud	
Parámetros	Descripción
nombresbar	Nombre del establecimiento
direccionresbar	Dirección del establecimiento
telefonoresbar	Teléfono del establecimiento
nombreusu	Nombre del usuario que usará el administrador para autenticarse
passusu	Contraseña del usuario que usará el administrador para autenticarse
Respuesta	
Campo	Descripción
message	Indica si los datos se han registrado correctamente

4.1.2.1.2 Servicio de autenticación

Tabla 4–11. API - Servicio autenticar usuario

Servicio	
Nombre del fichero	authservice.php
Funcionalidad	Comprueba que exista el usuario en base de datos
Método	GET
Endpoint	localhost/ocusite/admin/authservice.php
Solicitud	
Parámetros	Descripción
nombreusu	Nombre del usuario
passusu	Contraseña del usuario
Respuesta	
Campos	Descripción
ok	Devuelve 1 si la autenticación es correcta, 0 si no la es
idresbar	Devuelve el identificador del restaurante al que pertenece el usuario o null

	si la autenticación no es correcta
--	------------------------------------

4.1.2.1.3 Servicios para la gestión de la carta

Tabla 4–12. API - Servicio obtener productos carta – Administrador

Servicio		
Nombre del fichero	cartaservice.php	
Funcionalidad	Devuelve los elementos de la carta de un bar o restaurante específico	
Método	GET	
Endpoint	localhost/ocusite/cartaservice.php	
Solicitud		
Parámetros	Descripción	
idresbar	Identificador del establecimiento	
Respuesta		
Campos	Descripción	
Lista	Lista de productos con la información de cada uno	
	nombrecat	Nombre de la categoría del producto
	idplacar	Identificador del producto de la carta
	idcat	Identificador de la categoría
	precio	Precio general
	preciotapa	Precio de la tapa
	preciomedia	Precio de media ración
	precioracion	Precio de la ración
	nombrelacar	Nombre del producto
	alepescado	Alérgeno de pescado
	alefrutossecos	Alérgeno de frutos secos
	alelacteos	Alérgeno de lácteos
	alemoluscos	Alérgeno de moluscos

	alecerealesgluten	Alérgeno de gluten
	alecrustaceo	Alérgeno de crustáceo
	alehuevos	Alérgeno de huevo
	alecacahuetes	Alérgeno de cacahuetes
	alesoja	Alérgeno de soja
	aleapio	Alérgeno de apio
	alemostaza	Alérgeno de mostaza
	alesesamo	Alérgeno de sésamo
	alealtramuz	Alérgeno de altramuz
	alesulfitos	Alérgeno de sulfitos

Tabla 4–13. API - Servicio crear producto carta

Servicio	
Nombre del fichero	cartaservice.php
Funcionalidad	Crea un producto de la carta
Método	POST
Endpoint	localhost/ocusite/cartaservice.php
Solicitud	
Parámetros	Descripción
idcar	Identificador de la carta del establecimiento
nombrequitar	Nombre del producto de la carta
idcat	Identificador de la categoría
precio	Precio general del producto
preciotapa	Precio de la tapa
preciomedia	Precio de la media ración
precioracion	Precio de la ración
alepescado	Alérgeno de pescado

alefrutossecos	Alérgeno de frutos secos
alelacteos	Alérgeno de lácteos
alemoluscos	Alérgeno de moluscos
alecerealesgluten	Alérgeno de gluten
alecrustaceos	Alérgeno de crustáceos
alehuevos	Alérgeno de huevos
alecacahuetes	Alérgeno de cacahuetes
alesoja	Alérgeno de soja
aleapio	Alérgeno de apio
alemostaza	Alérgeno de mostaza
alesesamo	Alérgeno de sésamo
alealtramuz	Alérgeno de altramuz
alesulfitos	Alérgeno de sulfitos
Respuesta	
Campo	Descripción
message	Indice si el producto se ha creado correctamente

Tabla 4–14. API - Servicio obtener producto

Servicio	
Nombre del fichero	cartaservice.php
Funcionalidad	Obtiene la información de un producto concreto de la carta
Método	GET
Endpoint	localhost/ocusite/cartaservice.php
Solicitud	
Parámetros	Descripción
idplacar	Identificador del producto de la carta
Respuesta	

Campos	Descripción
idplacar	Identificador del producto de la carta
idcar	Identificador de la carta del establecimiento
nombreplacar	Nombre del producto de la carta
idcat	Identificador de la categoría
precio	Precio general del producto
preciotapa	Precio de la tapa
preciomedia	Precio de la media ración
precioracion	Precio de la ración
alepescado	Alérgeno de pescado
alefrutossecos	Alérgeno de frutos secos
alalacteos	Alérgeno de lácteos
alemoluscos	Alérgeno de moluscos
alecerealesgluten	Alérgeno de gluten
alecrustaceos	Alérgeno de crustáceos
alehuevos	Alérgeno de huevos
alecacahuetes	Alérgeno de cacahuetes
alesoja	Alérgeno de soja
aleapio	Alérgeno de apio
alemostaza	Alérgeno de mostaza
alesesamo	Alérgeno de sésamo
alealtramuz	Alérgeno de altramuz
alesulfitos	Alérgeno de sulfitos

Tabla 4–15. API - Servicio editar producto

Servicio	
Nombre del fichero	cartaservice.php

Funcionalidad	Edita la información de un producto ya existente
Método	POST
Endpoint	localhost/ocusite/cartaservice.php
Solicitud	
Parámetros	Descripción
idplacar	Identificador del producto de la carta
nombrepacar	Nombre del producto de la carta
idcat	Identificador de la categoría
precio	Precio general del producto
preciotapa	Precio de la tapa
preciomedia	Precio de la media ración
precioracion	Precio de la ración
alepescado	Alérgeno de pescado
alefrutossecos	Alérgeno de frutos secos
alelacteos	Alérgeno de lácteos
alemoluscos	Alérgeno de moluscos
alecerealesgluten	Alérgeno de gluten
alecrustaceos	Alérgeno de crustáceos
alehuevos	Alérgeno de huevos
alecacahuetes	Alérgeno de cacahuetes
alesoja	Alérgeno de soja
aleapio	Alérgeno de apio
alemostaza	Alérgeno de mostaza
alesesamo	Alérgeno de sésamo
alealtramuz	Alérgeno de altramuz
alesulfitos	Alérgeno de sulfitos
Respuesta	
Campos	Descripción

message	Indica si se ha actualizado el producto o ha habido algún problema
---------	--

Tabla 4–16. API - Servicio eliminar producto

Servicio	
Nombre del fichero	cartaservice.php
Funcionalidad	Edita la información de un producto ya existente
Método	POST
Endpoint	localhost/ocusite/cartaservice.php
Solicitud	
Parámetros	Descripción
idplacar	Identificador del producto de la carta
action	Cadena que identifica la acción de borrar
Respuesta	
Campos	Descripción
message	Indica si se ha borrado el producto o ha habido algún problema

4.1.2.1.4 Servicios para la gestión de información fuera de carta

Tabla 4–17. API - Servicio obtener fuera de carta

Servicio	
Nombre del fichero	fueracartaservice.php
Funcionalidad	Obtiene la información de los siete registros disponibles en un establecimiento
Método	GET
Endpoint	localhost/ocusite/fueracartaservice.php
Solicitud	
Parámetros	Descripción
idresbar	Identificador del establecimiento

Respuesta	
Campos	Valor
idfuecar	Identificador de la información fuera de carta
space1	Información del registro 1
space2	Información del registro 2
space3	Información del registro 3
space4	Información del registro 4
space5	Información del registro 5
space6	Información del registro 6
space7	Información del registro 7

Tabla 4–18. API - Servicio editar fuera de carta

Servicio	
Nombre del fichero	fueracartaservice.php
Funcionalidad	Actualiza la información de los espacios de fuera de carta del establecimiento
Método	POST
Endpoint	localhost/ocusite/fueracartaservice.php
Solicitud	
Parámetros	Descripción
idfuecar	Identificador de la información fuera de carta
space1	Información del registro 1
space2	Información del registro 2
space3	Información del registro 3
space4	Información del registro 4
space5	Información del registro 5
space6	Información del registro 6
space7	Información del registro 7

Respuesta	
Campo	Descripción
message	Indica si la información se ha actualizado correctamente o ha habido algún problema

4.1.2.1.5 Servicios para la gestión de horarios

Tabla 4–19. API - Servicio obtener horarios

Servicio	
Nombre del fichero	horarioservice.php
Funcionalidad	Obtiene la información de horarios del establecimiento
Método	GET
Endpoint	localhost/ocusite/horarioservice.php
Solicitud	
Parámetros	Descripción
idresbar	Identificador del establecimiento
Respuesta	
Campos	Descripción
idperhorresbar	Identificador de los horarios del establecimiento
horaini1	Hora inicial del periodo de mañana
horafin1	Hora final del periodo de mañana
horaini2	Hora inicial del periodo de mañana
horafin2	Hora final del periodo de mañana
horaini3	Hora inicial del periodo de mañana
horafin3	Hora final del periodo de mañana

Tabla 4–20. API - Servicio actualizar horarios

Servicio

Nombre del fichero	horarioservice.php
Funcionalidad	Actualiza la información de los horarios
Método	POST
Endpoint	localhost/ocusite/horarioservice.php
Solicitud	
Parámetros	Descripción
idperhorresbar	Identificador de los horarios del establecimiento
horaini1	Hora inicial del periodo de mañana
horafin1	Hora final del periodo de mañana
horaini2	Hora inicial del periodo de mañana
horafin2	Hora final del periodo de mañana
horaini3	Hora inicial del periodo de mañana
horafin3	Hora final del periodo de mañana
Respuesta	
Parámetros	Valor
message	Indica si la información de horarios se ha actualizado o ha habido algún problema

4.1.2.1.6 Servicios para la gestión de ocupación

Tabla 4–21. API - Servicio obtener ocupación

Servicio	
Nombre del fichero	ocupacionservice.php
Funcionalidad	Obtiene la información de la ocupación disponible en tiempo real
Método	GET
Endpoint	localhost/ocusite/ocupacionservice.php
Solicitud	
Parámetros	Descripción

idresbar	Identificador del establecimiento
Respuesta	
Campos	Valor
numeromesasdisponibles	Número de mesas disponibles en el establecimiento
numeromesastotales	Número de mesas totales en el establecimiento

Tabla 4–22. API - Servicio actualizar ocupación

Servicio	
Nombre del fichero	ocupacionservice.php
Funcionalidad	Actualiza la ocupación del establecimiento
Método	POST
Endpoint	localhost/ocusite/ocupacionservice.php
Solicitud	
Parámetros	Descripción
idresbar	Identificador del
numeromesasdisponibles	Número de mesas disponibles en el establecimiento
numeromesastotales	Número de mesas totales en el establecimiento
Respuesta	
Parámetros	Valor
message	Indica si la información de ocupación se ha actualizado correctamente

4.1.2.1.7 Servicios para la gestión de reservas

Tabla 4–23. API - Servicio obtener reservas - Administrador

Servicio	
Nombre del fichero	reservasservice.php
Funcionalidad	Obtiene la información de los intervalos de reservas y de las reservas de cada intervalo.

Método	GET	
Endpoint	localhost/ocusite/reservasservice.php	
Solicitud		
Parámetros	Descripción	
idresbar	Identificador del establecimiento	
Respuesta		
Campos	Descripción	
Lista	Lista de reservas	
	idinthorresresbar	Identificador del intervalo de reserva
	horaini	Hora inicial del intervalo
	horafin	Hora final del intervalo
	espacioreserva	Número de mesas disponible en el intervalo
	idperres	Identificador del registro de reserva en el intervalo
	nombrepres	Nombre de la persona registrada en la reserva
	telefonopres	Teléfono de la persona registrada en la reserva
	correopres	Correo de la persona registrada en la reserva
	numeromesapres	Número de personas para la reserva

Tabla 4-24. API - Servicio crear intervalo

Servicio	
Nombre del fichero	reservasservice.php
Funcionalidad	Crea un intervalo de reserva nuevo
Método	POST
Endpoint	localhost/ocusite/fueracartaservice.php
Solicitud	
Parámetros	Descripción
idresbar	Identificador del establecimiento

espacioreserva	Número de mesas que se permiten reservar en el intervalo
horaini	Hora inicial del intervalo
horafin	Hora final del intervalo
Respuesta	
Campos	Descripción
message	Indica si el intervalo se ha creado correctamente o ha habido algún problema

Tabla 4–25. API – Servicio obtener intervalo

Servicio	
Nombre del fichero	reservasservice.php
Funcionalidad	Obtiene la información de un intervalo concreto
Método	GET
Endpoint	localhost/ocusite/ reservasservice.php
Solicitud	
Parámetros	Descripción
idinthorresbar	Identificador del intervalo
Respuesta	
Campos	Descripción
idinthorresbar	Identificador del intervalo
idresbar	Identificador del establecimiento
espacioreserva	Número de mesas que se permiten reservar en el intervalo
horaini	Hora inicial del intervalo
horafin	Hora final del intervalo

Tabla 4–26. API - Servicio editar intervalo

Servicio	
Nombre del fichero	reservasservice.php

Funcionalidad	Actualiza la información de un intervalo concreto
Método	POST
Endpoint	localhost/ocusite/reservasservice.php
Solicitud	
Parámetros	Descripción
idinthorresresbar	Identificador del establecimiento
espaciosreserva	Número de mesas que se permiten reservar en el intervalo
horaini	Hora inicial del intervalo
horafin	Hora final del intervalo
Respuesta	
Campos	Descripción
message	Indica si se ha actualizado el intervalo o ha habido algún problema

Tabla 4-27. API - Servicio eliminar intervalo

Servicio	
Nombre del fichero	reservasservice.php
Funcionalidad	Elimina el intervalo
Método	POST
Endpoint	localhost/ocusite/reservasservice.php
Solicitud	
Parámetros	Descripción
idinthorresresbar	Identificador del intervalo
action	Cadena que identifica la acción de borrar
Respuesta	
Campos	Descripción
message	Indica si el intervalo se ha borrado o si ha habido algún problema

Tabla 4–28. API - Servicio crear reserva

Servicio	
Nombre del fichero	reservasservice.php
Funcionalidad	Crea una reserva nueva para un intervalo concreto
Método	POST
Endpoint	localhost/ocusite/reservasservice.php
Solicitud	
Parámetros	Descripción
idinthorresresbar	Identificador del establecimiento
nombreperres	Nombre de la persona a registrar la reserva
telefonoperres	Teléfono de la persona a registrar la reserva
correoperres	Correo de la persona a registrar la reserva
numeromesasperres	Número de personas para la reserva
Respuesta	
Campos	Descripción
message	Indica si la reserva se ha creado correctamente o ha habido algún problema

Tabla 4–29. API – Servicio obtener reserva

Servicio	
Nombre del fichero	reservasservice.php
Funcionalidad	Obtiene información de una reserva concreta
Método	GET
Endpoint	localhost/ocusite/reservasservice.php
Solicitud	
Parámetros	Descripción
idperres	Identificador de la reserva
Respuesta	

Campos	Descripción
idperres	Identificador de la reserva
idinthorresresbar	Identificador del intervalo al que pertenece la reserva
nombrequerres	Nombre de la persona a registrar la reserva
telefonoperres	Teléfono de la persona a registrar la reserva
correoperres	Correo de la persona a registrar la reserva
numeromesasperres	Número de personas para la reserva

Tabla 4–30. API - Servicio editar reserva

Servicio	
Nombre del fichero	reservasservice.php
Funcionalidad	Actualiza la información de una reserva concreta
Método	POST
Endpoint	localhost/ocusite/reservasservice.php
Solicitud	
Parámetros	Descripción
idperres	Identificador de la reserva
nombrequerres	Nombre de la persona a registrar la reserva
telefonoperres	Teléfono de la persona a registrar la reserva
correoperres	Correo de la persona a registrar la reserva
numeromesasperres	Número de personas para la reserva
Respuesta	
Campos	Descripción
message	Indica si la reserva se ha actualizado o si ha habido algún problema

Tabla 4–31. API - Servicio eliminar reserva

Servicio

Nombre del fichero	reservasservice.php
Funcionalidad	Elimina la reserva
Método	POST
Endpoint	localhost/ocusite/reservasservice.php
Solicitud	
Parámetros	Descripción
idperres	Identificador de la reserva
action	Cadena que identifica la acción de borrar
Respuesta	
Campos	Descripción
message	Indica si la reserva se ha borrado o si ha habido algún problema

4.1.2.1.8 Servicios para la gestión de servicios disponibles

Tabla 4–32. API – Servicio obtener servicios disponibles

Servicio	
Nombre del fichero	infoservice.php
Funcionalidad	Obtiene información de los servicios disponibles en el establecimiento y algunos datos más relacionados con el usuario
Método	GET
Endpoint	localhost/ocusite/infoservice.php
Solicitud	
Parámetros	Descripción
idresbar	Identificador del establecimiento
Respuesta	
Campos	Descripción
idresbar	Identificador del establecimiento
nombresbar	Nombre del establecimiento

direccionresbar	Dirección del establecimiento
telefonoresbar	Teléfono del establecimiento
lat	Latitud de localización
lng	Longitud de localización
disponiblecarta	Disponibilidad de la carta
disponibleocupacion	Disponibilidad de la ocupación
disponiblefuera de carta	Disponibilidad de fuera de carta
disponiblereserva	Disponibilidad de reserva
disponiblehorario	Disponibilidad de horario
disponiblecomerensitio	Disponibilidad de comer en el establecimiento
disponibledomicilio	Disponibilidad de domicilio
disponiblerecojerlocal	Disponibilidad de recoger la comida en el establecimiento

Tabla 4-33. API - Servicio editar servicios disponibles

Servicio	
Nombre del fichero	infoservice.php
Funcionalidad	Actualiza la información de los servicios disponibles del establecimiento
Método	POST
Endpoint	localhost/ocusite/infoservice.php
Solicitud	
Parámetros	Descripción
idresbar	Identificador del establecimiento
disponiblecarta	Disponibilidad de la carta
disponibleocupacion	Disponibilidad de la ocupación
disponiblefuera de carta	Disponibilidad de fuera de carta
disponiblereserva	Disponibilidad de reserva
disponiblehorario	Disponibilidad de horario

disponiblecomerensitio	Disponibilidad de comer en el establecimiento
disponibledomicilio	Disponibilidad de domicilio
disponiblerecogerlocal	Disponibilidad de recoger la comida en el establecimiento
Respuesta	
Campos	Descripción
message	Indica si la información se ha actualizado correctamente o ha habido algún problema

4.1.2.2 API usuario cliente

4.1.2.2.1 Servicio para la localización

Tabla 4–34. API - Servicio obtener establecimientos

Servicio	
Nombre del fichero	resbarservice.php
Funcionalidad	Obtiene latitud y longitud de los establecimientos para mostrarlos en el mapa, junto a algunos datos de interés
Método	GET
Endpoint	localhost/ocusite/resbarservice.php
Solicitud	
Parámetros	Descripción
Ninguno	
Respuesta	
Campos	Descripción
Lista	Contiene información de los establecimientos registrados en el sistema

	nombresbar	Nombre del establecimiento
	lat	Latitud de localización
	lng	Longitud de localización
	disponibleocupacion	Disponibilidad de la ocupación
	activo	Establecimiento activo en el sistema

4.1.2.2.2 Servicio para la información general

Tabla 4-35. API - Servicio obtener información

Servicio	
Nombre del fichero	resbarservice.php
Funcionalidad	Obtiene la información de un establecimiento concreto
Método	GET
Endpoint	localhost/ocusite/resbarservice.php
Solicitud	
Parámetros	Descripción
idresbar	Identificador del establecimiento
Respuesta	
Parámetros	Valor
idresbar	Identificador del establecimiento
nombresbar	Nombre del establecimiento
direccionresbar	Dirección del establecimiento
telefonoresbar	Teléfono del establecimiento
disponiblecarta	Disponibilidad de la carta
disponibleocupacion	Disponibilidad de la ocupación
disponiblefuera de carta	Disponibilidad de fuera de carta
disponiblereserva	Disponibilidad de reserva
disponiblehorario	Disponibilidad de horario

disponiblecomerensitio	Disponibilidad de comer en el establecimiento
disponibledomicilio	Disponibilidad de domicilio
disponiblerecogerlocal	Disponibilidad de recoger la comida en el establecimiento
horaini1	Hora inicial del periodo de mañana
horafin1	Hora final del periodo de mañana
horaini2	Hora inicial del periodo de tarde
horafin2	Hora final del periodo de tarde
horaini3	Hora inicial del periodo de noche
horafin3	Hora final del periodo de noche

4.1.2.2.3 Servicio para la carta

Tabla 4–36. API - Servicio obtener productos de carta - Cliente

Servicio		
Nombre del fichero	cartaservice.php	
Funcionalidad	Devuelve los elementos de la carta de un bar o restaurante específico	
Método	GET	
Endpoint	localhost/ocusite/cartaservice.php	
Solicitud		
Parámetros	Descripción	
idresbar	Identificador del establecimiento	
Respuesta		
Campos	Descripción	
Lista	Lista de productos con la información de cada uno	
	nombrecat	Nombre de la categoría del producto
	idplacar	Identificador del producto de la carta
	idcat	Identificador de la categoría
	precio	Precio general

	preciotapa	Precio de la tapa
	preciomedia	Precio de media ración
	precioracion	Precio de la ración
	nombrequitar	Nombre del producto
	alepescado	Alérgeno de pescado
	alefrutossecos	Alérgeno de frutos secos
	alelacteos	Alérgeno de lácteos
	alemoluscos	Alérgeno de moluscos
	alecerealesgluten	Alérgeno de gluten
	alecrustaceo	Alérgeno de crustáceo
	alehuevos	Alérgeno de huevo
	alecacahuetes	Alérgeno de cacahuetes
	alesoja	Alérgeno de soja
	aleapio	Alérgeno de apio
	alemostaza	Alérgeno de mostaza
	alesesamo	Alérgeno de sésamo
	alealtramuz	Alérgeno de altramuz
	alesulfitos	Alérgeno de sulfitos

4.1.2.2.4 Servicio para las reservas

Tabla 4–37. API - Servicio obtener reservas - Cliente

Servicio	
Nombre del fichero	reservasservice.php
Funcionalidad	Obtiene la información de los intervalos de reservas y de las reservas de cada intervalo.
Método	GET
Endpoint	localhost/ocusite/reservasservice.php

Solicitud		
Parámetros	Descripción	
idresbar	Identificador del establecimiento	
Respuesta		
Campos	Descripción	
Lista	Lista de intervalos	
	idinthorresresbar	Identificador del intervalo de reserva
	horaini	Hora inicial del intervalo
	horafin	Hora final del intervalo
	espacioreserva	Número de mesas disponible en el intervalo

4.2 Aplicación de usuario

4.2.1 Análisis funcional

Como hemos hecho en la sección anterior en la subsección de la *API*, dividimos el análisis en usuario administrador y usuario cliente, a excepción de la primera pantalla, dónde se determina el tipo de usuario.

4.2.1.1 Pantalla principal

Es la primera pantalla que se ve al entrar en la aplicación, existe un menú con tres opciones posibles. En función del tipo de usuario se elegirá una opción u otra.



Figura 4-6. Pantalla principal de OcuSite

4.2.1.2 Usuario administrador

Se analizan las diferentes pantallas por las que puede navegar un usuario administrador.

4.2.1.2.1 Pantalla de autenticación

Se trata de una pantalla clásica de autenticación, se mostrará el mensaje de error “Error en la autenticación” si los datos no son correctos, de lo contrario entrará en las pantallas de administración del establecimiento. Además, existe una opción que llevará a un formulario alta para que los usuarios interesados.



Figura 4-7. Pantalla de autenticación

4.2.1.2.2 Pantalla de alta de usuario

Contiene un formulario con una serie de campos para rellenar, nombre del usuario, contraseña para entrar a la administración, correo electrónico, teléfono, nombre del establecimiento y dirección completa.

OCUSITE

Nombre usuario

Contraseña

Correo electrónico

Teléfono

Nombre establecimiento

Dirección completa

Estoy de acuerdo con los términos del servicio

ENVIAR SOLICITUD

VOLVER

Figura 4-8. Pantalla de alta de usuario

4.2.1.2.3 Pantalla principal del menú de administración

La pantalla principal de administrador dispone de una “*toolbar*” con varios botones y un texto que indica la pantalla de gestión o información en la que estamos. Los botones se representan con iconos intuitivos y las funciones son:

- Desplegar menú de navegación.
- Abrir la pantalla de gestión de servicios disponibles.
- Salir de la administración del establecimiento.

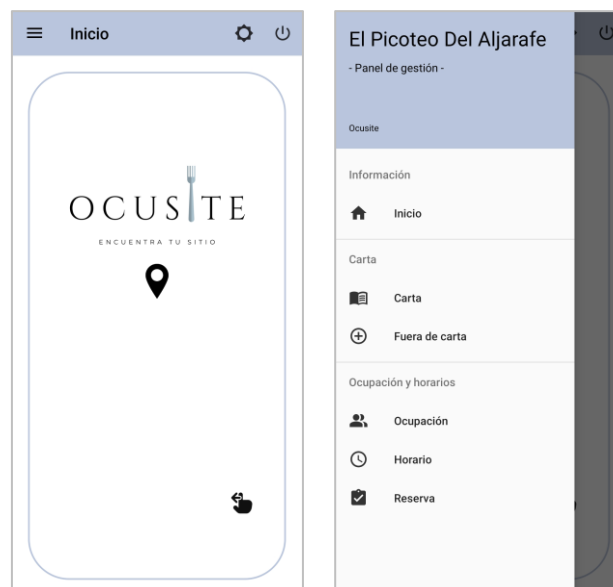


Figura 4-9. Pantalla principal del menú de administración – administrador

4.2.1.2.4 Pantalla de inicio

Es la pantalla por defecto cuando el usuario se loguea. Contiene información con las diferentes secciones que

puede gestionar, el usuario podrá arrastrar con el dedo la pantalla para pasar las páginas.

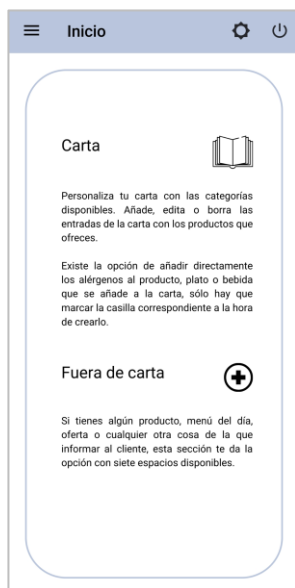


Figura 4-10. Pantalla de inicio – administrador

4.2.1.2.5 Pantalla de gestión de carta

La pantalla muestra la carta existente en el establecimiento si el administrador ha añadido productos, existe una cabecera con las categorías y seleccionando una categoría despliega la lista de productos correspondientes a la misma. Las opciones disponibles son:

- Añadir un nuevo producto, conduce a la pantalla de creación de producto (4.2.1.2.6)
- Editar un producto existente tocando el elemento de la lista que se quiere editar, conduce a la pantalla de edición de producto (4.2.1.2.7)

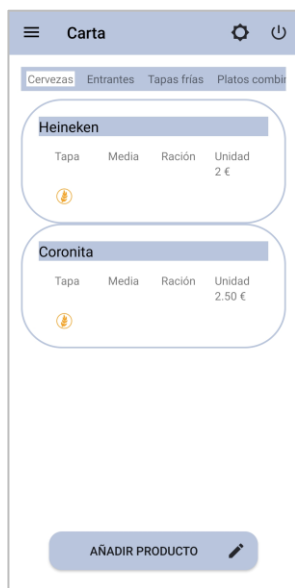


Figura 4-11. Pantalla de gestión de carta – administrador

4.2.1.2.6 Pantalla de creación de producto

La pantalla tiene una serie de campos para rellenar relacionados con un producto que se pueda ofrecer en un establecimiento, contando con “checkboxs” para indicar si contiene un alérgeno concreto. El usuario podrá:

- Volver atrás, pantalla de gestión de carta (4.2.1.2.5).
- Guardar el producto después de rellenar los campos, que aparecerá de forma inmediata en la pantalla de gestión de carta.

The screenshot shows a mobile application interface for creating a new product. At the top, there's a header 'Nuevo producto' with a checkmark and a back arrow. Below the header, there's a section 'Plato o bebida' with two input fields: 'Nombre' and 'Categoría'. Underneath, there are four price-related fields: 'Unidad' and 'Tapa' (both with a Euro symbol), and 'Media' and 'Ración' (both with a Euro symbol). The bottom section is titled 'Alérgenos' and contains a grid of 16 allergen categories, each with a small icon and a checkbox: Gluten, Lácteos, Pescados, Frutos secos, Moluscos, Crustáceos, Sésamo, Sulfitos, Altramuzes, Mostaza, Apio, Cacahuates, Soja, and Huevos.

Figura 4-12. Pantalla de creación de producto – administrador

4.2.1.2.7 Pantalla de edición de producto

La pantalla es casi igual que la pantalla de creación de producto, sólo incluye un nuevo icono y el texto de información cambia. Opciones:

- Volver atrás, pantalla de gestión de carta (4.2.1.2.5).
- Guardar el producto con el cambio realizado.
- Eliminar el producto de la carta, volverá a la pantalla de gestión de carta (4.2.1.2.5). El producto ya no se reflejará en la carta.

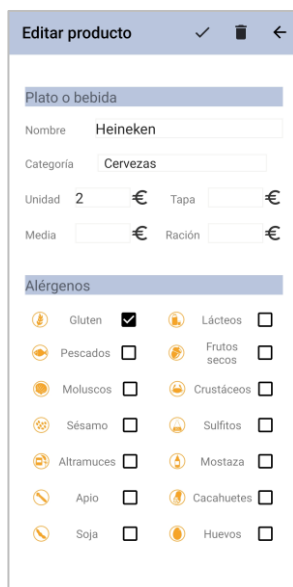


Figura 4-13. Pantalla de edición de producto – administrador

4.2.1.2.8 Pantalla de gestión fuera de carta

Contiene una serie de campos de texto dónde el administrador del establecimiento puede informar al cliente sobre productos que no quiera categorizar, o sobre ofertas que tenga en ese momento. Opciones:

- Actualizar la información.



Figura 4-14. Pantalla de gestión fuera de carta – administrador

4.2.1.2.9 Pantalla de gestión de ocupación

La pantalla tiene la función de actualizar en tiempo real la ocupación del establecimiento, el administrador puede actualizar las mesas totales que tiene disponible y la ocupación en ese momento. Opciones:

- Actualizar la información.

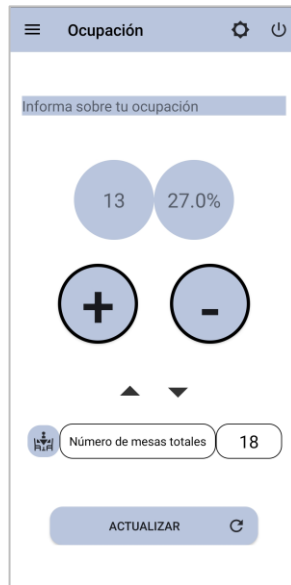


Figura 4-15. Pantalla de gestión de ocupación – administrador

4.2.1.2.10 Pantalla de gestión de horarios

El administrador podrá cambiar los horarios desde esta pantalla. Opciones:

- Actualizar la información.

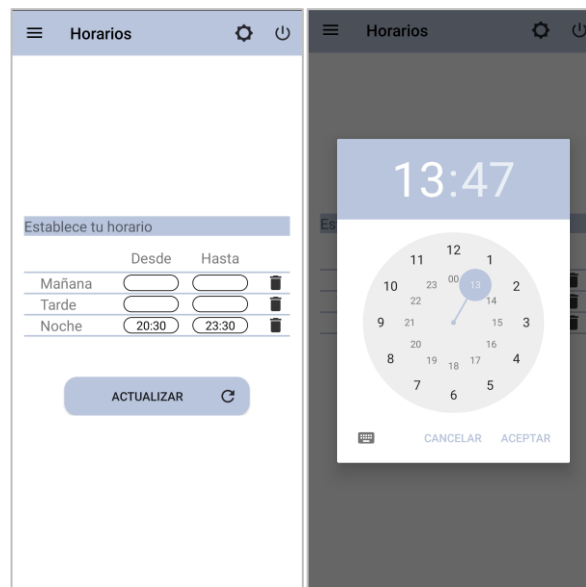


Figura 4-16. Pantalla de gestión de horarios – administrador

4.2.1.2.11 Pantalla de gestión de reservas

La pantalla muestra los intervalos de reservas que hay y dentro de cada intervalo, las reservas existentes. Tiene dos cabeceras, una cabecera superior que despliega la lista de intervalos y una cabecera inferior con los intervalos que despliega la lista de reservas del intervalo. Las opciones dependen la lista que esté desplegada, si está desplegada la lista de intervalos las opciones son:

- Añadir un nuevo intervalo de reserva, conduce a la pantalla de creación de intervalo (4.2.1.2.).
- Editar un intervalo existente tocando el elemento de la lista que se quiere editar, conduce a la pantalla de edición de intervalo (4.2.1.2.)

Las otras opciones aparecen con la lista de reservas de un intervalo desplegada:

- Añadir una nueva reserva, conduce a la pantalla de creación de reserva (4.2.1.2.).
- Editar una reserva existente tocando el elemento de la lista que se quiere editar, conduce a la pantalla de edición de reserva (4.2.1.2.)



Figura 4-17. Pantalla de gestión de reservas – administrador

4.2.1.2.12 Pantalla de creación de intervalo

La pantalla tiene una serie de campos para rellenar, hora de inicio, hora de fin y mesas que el administrador quiera poner disponible para reserve en ese intervalo. Opciones:

- Volver atrás, pantalla de gestión de reservas (4.2.1.2.11).
- Guardar el intervalo después de rellenar los campos, que aparecerá de forma inmediata en la pantalla de gestión de reservas.

Figura 4-18. Pantalla de creación de intervalo – administrador

4.2.1.2.13 Pantalla de edición de intervalo

La pantalla es casi igual que la pantalla de creación de intervalo, sólo incluye un nuevo icono y el texto de información cambia. Opciones:

- Volver atrás, pantalla de gestión de reservas (4.2.1.2.11).
- Guardar el intervalo con el cambio realizado.
- Eliminar el intervalo de la lista de intervalos, volverá a la pantalla de gestión de reservas (4.2.1.2.11). El intervalo ya no se reflejará en la lista.

Figura 4-19. Pantalla de edición de intervalo – administrador

4.2.1.2.14 Pantalla de creación de reserva

La pantalla tiene una serie de campos para rellenar, nombre, teléfono y correo electrónico, además indicar el número de mesas que esa persona ocupará. Opciones:

- Volver atrás, pantalla de gestión de reservas (4.2.1.2.11).
- Guardar el intervalo después de rellenar los campos, que aparecerá de forma inmediata en la pantalla de gestión de reservas.




Figura 4-20. Pantalla de creación de reserva – administrador

4.2.1.2.15 Pantalla de edición de reserva

La pantalla es casi igual que la pantalla de creación de reserva, sólo incluye un nuevo icono y el texto de información cambia. Opciones:

- Volver atrás, pantalla de gestión de reservas (4.2.1.2.11).
- Guardar el intervalo con el cambio realizado.
- Eliminar la reserva de la lista de intervalo, volverá a la pantalla de gestión de reservas (4.2.1.2.11). La reserva ya no se reflejará en la lista.

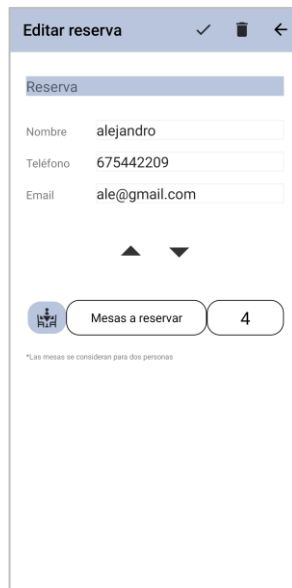


Figura 4-21. Pantalla de edición de reserva – administrador

4.2.1.2.16 Pantalla de gestión de servicios disponibles

Se puede activar y desactivar la disponibilidad de los servicios del establecimiento y de manera general desactivar el establecimiento. Estos cambios afectan a la información que el cliente del establecimiento podrá ver directamente. Opciones:

- Actualizar la información.

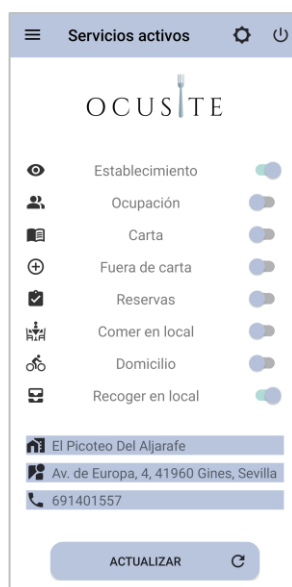


Figura 4-22. Pantalla de gestión de servicios disponibles – administrador

4.2.1.3 Usuario cliente

Cualquier persona que entre en la aplicación para buscar un sitio.

4.2.1.3.1 Pantalla de localización

Se trata de una pantalla con un mapa, para una mejor experiencia el usuario deberá activar los permisos de localización del dispositivo. El mapa muestra marcadores dónde se encuentran los establecimientos registrados, el color del marcador es significativo. Opciones:

- Tocar el marcador para tener información del nombre y la ocupación en tiempo real, si están disponibles los servicios necesarios.
- Tocar la ventana para navegar en la pantalla principal de información (4.2.1.3.2)

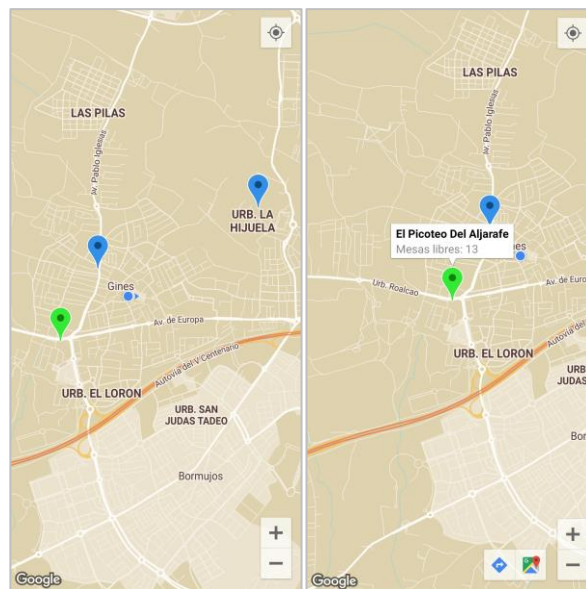


Figura 4-23. Pantalla de localización - cliente

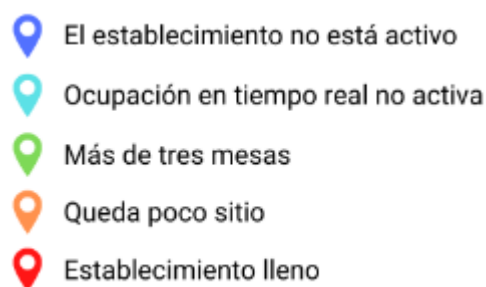


Figura 4-24. Leyenda de marcadores de la pantalla de localización

4.2.1.3.2 Pantalla de información del establecimiento

Refleja la información general del establecimiento junto con los servicios que tiene disponible.



Figura 4-25. Pantalla de información - cliente

4.2.1.3.3 Pantalla de la carta

Si el servicio de carta está disponible se muestra una cabecera con categorías para seleccionar que despliega la lista de elementos que pertenecen a la misma.

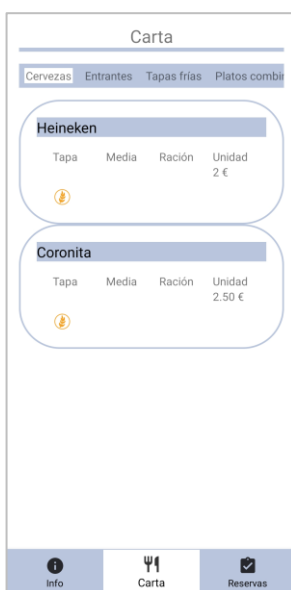


Figura 4-26. Pantalla de la carta - cliente

4.2.1.3.4 Pantalla de reservas

Si el servicio estuviera disponible aparecerían los intervalos de reservas con la información. A diferencia de la lista de reservas de administrador, el cliente no puede ver las personas que han reservado.

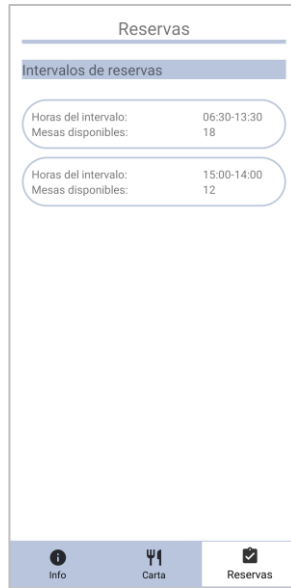


Figura 4-27. Pantalla de reservas- cliente

5 CONCLUSIONES

El valor de una idea radica en el uso de la misma.

- Tomas A Edison -

El proyecto ha sido una idea personal con el fin de facilitar y agilizar algunas de las tareas relacionadas con el mundo de la hostelería.

El administrador de un establecimiento gana una herramienta sencilla para digitalizar y poner a disposición de los clientes lo que ofrece y el espacio que tiene disponible.

El cliente tiene una serie de ventajas también, ya que entre más establecimientos estén adheridos en el sistema, éste podrá hacer una mejor comparación de lo que está buscando concretamente en ese momento, podrá ahorrar tiempo evitando desplazamientos si sabe que el establecimiento está ocupado completamente, podrá ver con anterioridad si la carta y los productos que ofrece el bar o restaurante le convence y podrá reservar en el intervalo de tiempo que le convenga.

6 LÍNEAS FUTURAS

La cantidad de aplicaciones instaladas en un dispositivo es inversamente proporcional a su uso.

- Anónimo -

Con todo esto el alcance del proyecto ya estaría cubierto y los objetivos propuestos cumplidos. No obstante, queda más desarrollo y otros tipos de conceptos que abordar. A continuación, se exponen los puntos que se quedan para seguir trabajando.

6.1 Desarrollo de la aplicación móvil en el sistema operativo iOS

El sistema operativo de Apple abarca una parte considerable de público objetivo al que está destinado la aplicación, hay muchas personas que utilizan iPhone como dispositivo móvil. Por lo tanto crear la aplicación para este tipo de dispositivos es uno de los objetivos futuros.

Por un lado, la arquitectura de la base de datos y el servidor que contiene la API no se modificarán, no habrá que crear servicios nuevos.

Por otra parte, la aplicación a nivel de interfaz y arquitectura será lo más similar posible a la desarrollada en el proyecto, en la algoritmia y el diseño.

6.2 Implementación de seguridad en la comunicación con los servicios

La seguridad cada vez es más importante en el mundo digital. Aunque los datos que se tratan no son datos sensibles, si hablamos del administrador de un establecimiento, es deseable para cualquier usuario estar cubierto de cualquier tipo de suplantación y alteración en la gestión de su negocio. Es por eso por lo que uno de los temas pendientes sería la aplicación de buenas prácticas en el sistema de autenticación, o crear un mecanismo seguro, tanto para el usuario como para la aplicación en general.

6.3 Sincronización y escalabilidad

Estos son dos conceptos muy importantes a tener en cuenta a la hora de un aumento de usuarios en el sistema.

Una mala sincronización de las solicitudes contra el servidor puede hacer que la aplicación no responda ante servicios demandados o tener comportamientos inciertos. Implementar un sistema de colas en la API queda pendiente.

La sincronización es uno de los aspectos que influyen en la escalabilidad, de manera que si el número de usuarios crece el número de solicitudes también lo hace, y las probabilidades de que haya conflicto con dos solicitudes o más que llegan al servidor aumenta. Por lo tanto es importante hacer pruebas de rendimiento para asegurarnos que el sistema completo es consistente y es capaz de afrontar un crecimiento del número de usuarios.

6.4 Despliegue en producción y mantenimiento

El último paso será subir las aplicaciones a las plataformas correspondientes para que estén disponibles para los usuarios. Después de esto lo normal es hacer correcciones en funcionalidades y en la interfaz para mejorar la aplicación.

Bibliografía

- [1] Conceptos de Kotlin: <https://kotlinlang.org/docs/home.html>
- [2] Conceptos de Android y guía de desarrollo: <https://developer.android.com/guide?hl=es-419>
- [3] Guía de maps Google: <https://developers.google.com/maps/documentation/android-sdk/start?hl=es>
- [4] Soluciones de algunos problemas: <https://es.stackoverflow.com/>
- [5] Información sobre el uso de tecnología web: <https://w3techs.com/>
- [6] Soluciones para API en PHP: <https://www.php.net/docs.php>
- [7] Simulaciones de código PHP: <https://www.w3schools.com/>

