

# Gestor de negocios hosteleros en Android 2.0

Lorena Aguilar Briz

Máster en Ingeniería Informática



MÁSTERES  
DE LA UAM  
2017 - 2018

Escuela Politécnica Superior

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



**TRABAJO FIN DE MÁSTER**

Gestor de negocios hosteleros en Android 2.0

**Máster Universitario en Ingeniería Informática**

**Autor: AGUILAR BRIZ, Lorena**

**Tutor: MARTÍNEZ MUÑOZ, Gonzalo**  
**Departamento de Ingeniería Informática**

**FECHA: febrero, 2018**

**Todos los derechos reservados.**

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

**DERECHOS RESERVADOS**

© 2018 por UNIVERSIDAD AUTÓNOMA DE MADRID  
Francisco Tomás y Valiente, nº 1  
Madrid, 28049  
Spain

**Lorena Aguilar Briz**

*Gestor de negocios hosteleros en Android 2.0*

**Lorena Aguilar Briz**

Calle Mosquilona 59, 4ºA, Colmenar Viejo, Madrid

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

*Dedicado a mi abuelo.  
Estés donde estés, te quiero.*

*“Todas las piezas deben unirse sin ser forzadas.  
Debe recordar que los componentes que está  
reensamblando fueron desmontados por usted,  
por lo que si no puede unirlos debe existir una  
razón. Pero sobre todo, no use un martillo”*

*– Manual de mantenimiento de IBM, año 1925*



# AGRADECIMIENTOS

---

Hace aproximadamente un año y medio que daba por concluido el Grado en Ingeniería Informática, presentando mi trabajo final de grado el 11 de Julio de 2016. No mucho antes decidí seguir formándome y empezar los estudios de máster que me han conducido hasta aquí. Aún recuerdo a todas aquellas personas que me llamaron loca por seguir estudiando, a todas aquellas que, a día de hoy, me siguen diciendo que lo del máster es una tontería. Sin embargo, personalmente, no sé si me he formado más en este tiempo, no sé si he aprendido muchas más cosas que me fueran a ser necesarias, pero sé que me siento más completa de algún modo y mucho más preparada para el mundo laboral que me espera ahora.

Mucho tiempo ha pasado desde que emprendí el camino que me llevaría a este día. Pisé por primera vez la escuela el 9 de Septiembre de 2012 y ya me da miedo no volver a pisarla, pues al final de todo esto se ha convertido en mi segunda casa o, mas bien en la primera, y mi casa pasó a un segundo plano. Se que mis padres estarán orgullosos pero no más que dichosos por volver a verme paseando por casa. Cuatro largos años de carrera y un año y medio de máster en los que la informática me ha dado tanto, y también me ha quitado mucho. Pero llegados al final, solo queda recoger los frutos de ese esfuerzo. Sabiendo lo que se hoy, no habría tomado una decisión diferente hace casi 6 años, pese a las pérdidas.

Quiero empezar este punto agradeciéndoles a mis abuelos y mis padres que hoy este aquí. Mi abuelo ya no podrá verlo y lo mínimo que puedo hacer es mencionarle como merece. Ellos me han dado siempre el apoyo moral y económico que he necesitado para dedicarme enteramente a la carrera y no tener otras preocupaciones que no fuesen las propias de un estudiante. También me han aguantado en las épocas de exámenes, en los horribles días de entregas de prácticas y proyectos, y en las noches interminables tecleando y dando paseos por la casa, con todo esto nunca han dejado de apoyarme, de fingir que no les despertaba por las noches y que no les preocupaban mis suspensos. A ellos prácticamente se lo debo todo. Yo nunca quise estudiar una carrera, no me veía capaz, decía que era para gente "lista" y se que hoy es una superación personal para mi y también para ellos. No sabría decir de cual de las partes nace mayor orgullo hacia la otra.

No me puedo dejar al resto de personas que me han rodeado durante estos años, mi pareja, amigos, compañeros y profesores. No tengo palabras de agradecimiento para el apoyo que he recibido de todos ellos en los peores momentos, cuando les cancelaba los planes porque tenía que estudiar o cuando me pasaba meses sin verlos por el mismo motivo. Los incesantes correos con dudas y tutorías, en los que muchas veces pesaba más nuestro agobio que la propia carga de los exámenes o las prácticas, pero por suerte nunca que me encontrado un profesor que me negase la más mínima ayuda. Mis compañeros, se han acabado convirtiendo en una segunda familia, aunque solo sea por el número de horas que los he estado viendo diariamente, siempre he pensado que el camino ha sido mucho más fácil gracias a que hemos hecho piña piña, nos apoyábamos y ayudábamos. Compañeros de llanto, de alegrías, de revisiones, de suspensos, de competencia. Todos con un mismo fin hemos luchado en esta carrera y muchos, por desgracia, sin comprender que nosotros mismos eramos nuestro mejor apoyo. Yo he tenido la suerte de contar con compañeros excelentes que me han explicado cosas, me han ayudado en prácticas, etc. Y espero poder haberles ayudado del mismo modo, pues al final creo que como ingenieros informáticos, el trabajo en equipo es fundamental.

Todos han constituido un apoyo fundamental.

Por último agradecer al lector de este documento que dedique su tiempo en ello, pues para mi ha sido un orgullo realizar este proyecto, desde mi trabajo final de grado hasta aquí. Un proyecto que se ha extendido por casi 3 años, que es mi mayor dolor de cabeza y mi mayor orgullo hoy en día. De nuevo gracias, y solo espero que pueda disfrutarlo tanto como he disfrutado yo escribiendo cada una de las líneas.



# RESUMEN

---

En la actualidad existen todo tipo de herramientas en el mercado que, de una forma u otra, ayudan a los negocios de hostelería a gestionar su personal, productos, proveedores, etc. Aun así, aunque gran parte de estas herramientas mejoran la gestión de los establecimientos, están orientadas a los empleados y no a mejorar la experiencia de los clientes.

Hace aproximadamente dos años fue presentada en el marco de un TFG la herramienta **The Waiter**, que daba una solución parcial a esta carencia. En este trabajo **The Waiter** no se presenta como una herramienta, sino como una solución compuesta por varias herramientas que permiten mejorar la gestión de los negocios de hostelería de forma integral, tanto del lado de los clientes como del lado del negocio. Esta nueva versión del proyecto incluye dos herramientas nuevas, **The Waiter TPV** y **The Waiter Wear**, además de conservar la herramienta anterior, la cual es igual en esencia pero ha sufrido muchas mejoras, de cara, principalmente, a incluirla en el nuevo sistema.

**The Waiter Plus**, una herramienta diseñada en Android para dispositivos móviles y tabletas. Los usuarios objetivo de esta herramienta son tanto los clientes como los empleados del establecimiento. En el caso de los clientes nos permite integrar en los negocios una tecnología con el fin de reducir esperas, independizando sus peticiones de aquel personal destinado a atenderlas. Además permite una abierta comunicación entre todos los usuarios conectados a la red local del establecimiento. Esta herramienta es intuitiva y fácil de usar: no genera dudas en los usuarios. En el caso de los empleados, la aplicación permite que almacenen las peticiones de los clientes en su dispositivo móvil o tableta, con el objetivo de agilizar así este servicio.

Para completar esta herramienta, el sistema incorpora **The Waiter Wear**, una aplicación diseñada para smartwatches con sistema operativo Android Wear que permite realizar algunas de las acciones que realiza **The Waiter Plus** desde la muñeca de los clientes. No añade funcionalidades adicionales, se incorpora en el proyecto por comodidad para los clientes al realizar peticiones y recibir avisos.

**The Waiter TPV** tendrá un papel principal en el sistema, puesto que sin ella, el resto de aplicativos no funcionarán de manera correcta. Su característica más importante es que aloja la base de datos que contiene toda la información del establecimiento, información que será configurada por el administrador desde la propia aplicación. Esta herramienta permite gestionar elementos de los restaurantes tales como empleados, proveedores, clientes, inventario, valoraciones, estadísticas, carta, etc. En cuanto a los empleados, el administrador podrá configurar las contraseñas de acceso y tener control sobre los accesos al sistema de cada uno de ellos. Permitirá crear una lista de proveedores con varios datos sobre los mismos, estos proveedores después podrán ser asignados a los productos del inventario, lo que permitirá automatizar los pedidos si se desea. Los clientes, desde **The Waiter Plus**, podrán listar la carta realizando una petición a la base de datos que estará configurada en esta herramienta, también podrán enviar sus pedidos, valoraciones y comentarios, los cuales también serán almacenados en la base de datos de esta herramienta. El administrador tendrá acceso a estadísticas que le permitirán conocer datos relevantes, como por ejemplo el producto más pedido en un mes, o el camarero que está mejor valorado por los clientes, entre otras muchas opciones.

Todas estas funcionalidades se implementan con un protocolo de comunicación diseñado especialmente para este proyecto. Esto requiere únicamente la configuración de una red local en el establecimiento para que los clientes puedan solicitar a **The Waiter TPV** la información necesaria para realizar sus pedidos. Por ello la implantación de esta herramienta conlleva unos costes muy bajos, debido a que los terminales se mueven en un rango de precios variado y asequible si lo comparamos con herramientas utilizadas actualmente y adicionalmente sólo requiere de una red configurada adecuadamente, lo que facilita su integración en el mercado.

El sistema se ha implementado en Android para dispositivos móviles, tabletas y smartwatches por ser uno de los medios más utilizados. Entre los motivos más importantes por los que se ha elegido esta tecnología destacan los bajos costes y la facilidad de ser implantada en diferentes negocios. **The Waiter** es un sistema para agilizar la gestión de los negocios destinados a la hostelería y más concretamente al área de restauración.

**The Waiter** se presenta como una solución real y original a los problemas cotidianos de los establecimientos de hostelería, así como un medio al alcance de cualquier individuo para mejorar su experiencia en los negocios de este tipo.

## PALABRAS CLAVE

---

Aplicación móvil, Android, Android Wear, Java, Código QR, Smartphone, Smartwatch, Hostelería, Mensajería, Chat, Multicast, TPV.

# ABSTRACT

---

These days there are different kind of tools in the market that, in one way or another, help the hostelry business to manage their employees, products, suppliers, etc. Despite this, a big portion of these tools aim to improve only the management of the business, leaving aside the customers and not helping them to improve their experience while at the establishment.

About two years ago, in the context of a TFG, **The Waiter** was presented to provide a solution to this problem. In this project, **The Waiter** is not only a tool, but a complete solution composed by many tools to provide an end to end way to manage this kind of establishments. This new version of the project includes two new solutions, **The Waiter TPV** and "The waiter Wear", keeping also the old project, which is the same in essence, but with a lot of improvements in order to integrate it in the new system.

**The Waiter Plus** is a tool designed in Android for mobile phones and tablets. The target users of this tool are both the customers of the establishment and its own workers. In the case of the customers, they will be able to reduce waiting times by separating their requests of the employee that is destined to attend them. Also, it offers an open chat between all the customers that are connected to the local network of the establishment. This tool is intuitive and easy to use: the users won't encounter any trouble using it. In the case of the own workers of the establishment, the application offers a way to store the requests of the customers in order to streamline this service.

To complement this tool, the system also incorporates **The Waiter Wear**, an application designed for smartwatches whose operating system is Android Wear. This solution allows the customers to perform some of the actions that they would be able to perform using **The Waiter Plus**, but from their own wrists. This solution doesn't add new functionalities, it's just incorporated to make it easier for the customers to send their requests and receive some notifications.

**The Waiter TPV** plays the main role in the system, since, without it, the rest of the applications wouldn't work correctly. Its main characteristic is that it holds the database containing all the information required by the solution, information that will be configured by the establishment administrator in the own application. This will allow the administrator to manage all the elements related to their establishment, such as the employees, suppliers, customers, stock, customer ratings, statistics, menu, etc. Regarding the workers, the administrator will be able to configure their credentials and he will have control over all the accesses of every worker. The application will also allow the manager to create a list of suppliers with their related information. These providers will also be linked with the products of the stock, allowing the application to place replacement orders automatically, if the manager wishes so. The customers will be able to request the foods menu from **The Waiter Plus** by launching a request directly to the database (allocated in **The Waiter TPV**). In the same way, they will also be able to send their orders, ratings and comments, which will also be stored in the already mentioned database. The administrator will have access to different statistics showing relevant data for his business, such as the best-selling product per month, or the best rated worker, among many other things.

All these functionalities are implemented through a communication protocol exclusively designed for this project. This protocol only requires that a local network is configured within the establishment, allowing the customers to request **The Waiter TPV** the required information to place their orders. This is why the implementation of this solution has an associated low cost, because the required terminals have an affordable and wide range of prices, compared to those of the currently existing tools, and, as mentioned before, it only requires a correctly configured local network, which, all together, facilitates its integration in the market.

The system is designed for Android mobile phones, tablets and smartwatches, since Android is one of the

most widespread operating system for the most used devices of nowadays. Another important reason why this was the selected framework is because of the low costs associated to it, and because of the facilities that it offers to be implemented in different kind of businesses. **The Waiter** is a system to speed up the management process of hostelry businesses, more specifically, those in the area of restaurant business.

**The Waiter** is presented as a real and novel solution to everyday problems of hostelry business, as well as a medium within the reach of any individual to improve their experience in this type of business.

## KEYWORDS

---

Mobile application, Android, Android Wear, Java, QR code, Smartphone, Smartwatch, Hostelry, Messaging, Chat, Multicast, TPV.

# ÍNDICE

---

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contexto .....	1
1.2	Marco del proyecto .....	1
1.3	Motivación .....	2
1.4	Objetivos .....	2
1.5	Estructura del documento .....	3
<b>2</b>	<b>Estado del arte</b>	<b>5</b>
2.1	Negocios de hostelería .....	5
2.2	Aplicaciones de gestión hostelera y sus funciones típicas .....	5
2.3	Android .....	9
2.4	Estudio de mercado .....	9
2.5	Opiniones y conclusiones .....	12
<b>3</b>	<b>Objetivos y funcionalidades</b>	<b>17</b>
3.1	Introducción .....	17
3.2	Objetivos específicos y funcionalidades .....	17
<b>4</b>	<b>Definición del proyecto</b>	<b>21</b>
4.1	Metodología .....	21
4.2	Planificación orientativa .....	22
4.3	Herramientas utilizadas .....	25
<b>5</b>	<b>Análisis</b>	<b>29</b>
5.1	Introducción .....	29
5.2	Roles de usuario .....	29
5.3	Casos de uso .....	30
5.4	Catálogo y definición de requisitos .....	30
<b>6</b>	<b>Diseño e implementación</b>	<b>39</b>
6.1	Introducción .....	39
6.2	Arquitectura de la aplicación .....	39
6.3	Códigos QR .....	47
6.4	Interfaz de usuario .....	48
6.5	Estructura de las aplicaciones .....	48
<b>7</b>	<b>Pruebas</b>	<b>53</b>
7.1	Alcance de las pruebas .....	53
7.2	Estrategia y desarrollo .....	56
7.3	Resultados y conclusiones .....	57
<b>8</b>	<b>Evaluación</b>	<b>59</b>
8.1	Evaluación de los usuarios .....	59
8.2	Beneficios de la aplicación .....	60
8.3	Modularidad .....	60
<b>9</b>	<b>Conclusiones y líneas futuras</b>	<b>63</b>

9.1	Conclusiones .....	63
9.2	Trabajo futuro .....	64
	<b>Bibliografía</b>	<b>68</b>
	<b>Definiciones</b>	<b>69</b>
	<b>Acrónimos</b>	<b>71</b>
<b>I</b>	<b>Anexos</b>	<b>73</b>
A	Android	75
B	Resultados del Estudio de Mercado	79
C	Planificación	103
D	Generación de códigos QR	105
E	Envío de mensajes por la red	107
F	Recepción de mensajes de la red	109
G	Firebase	113
H	Interfaz <i>The Waiter Plus</i>	115
I	Interfaz <i>The Waiter Wear</i>	121
J	Interfaz <i>The Waiter TPV</i>	123
K	Test de alcoholemia	131

# LISTAS

---

## Lista de códigos

D.1	Generación de códigos QR. ....	105
E.1	Envío de mensajes. ....	107
F.1	Envío de mensajes. ....	109
F.2	Envío de mensajes. ....	110
F.3	Envío de mensajes. ....	111

## Lista de figuras

2.1	Versiones Android desde 2008 a 2016. ....	9
2.2	Gráficos de resultados del formulario en relación con el contexto para los clientes ....	11
2.3	Gráficos de resultados del formulario en relación con el impacto para los clientes ....	13
2.4	Gráficos de resultados del formulario en relación con el contexto para el negocio ....	14
2.5	Gráficos de resultados del formulario en relación con el impacto para el negocio ....	15
4.1	Ciclo de vida en cascada con realimentación. ....	22
4.2	Cuota de mercado de sistemas operativos para móviles ....	26
5.1	Diagrama casos de uso número 1 ....	31
5.2	Diagrama casos de uso número 2 ....	32
5.3	Diagrama casos de uso número 3 ....	33
6.1	Diagrama general de la arquitectura del sistema. ....	40
6.2	Diagrama de los usuarios del sistema y sus correspondientes aplicativos. ....	41
6.3	Diagrama Relacional de la base de datos. ....	45
6.4	Diagrama Entidad-Relación de la base de datos. ....	46
6.5	Códigos QR generados por la aplicación ....	47
6.6	Diagrama de navegación entre pantallas <b>The Waiter Plus</b> . ....	50
6.7	Diagrama de navegación entre pantallas <b>The Waiter Wear</b> . ....	51
6.8	Diagrama de navegación entre pantallas <b>The Waiter TPV</b> . ....	52
7.1	Crash Reporting. ....	54
7.2	Crash Reporting. ....	54
7.3	Estabilidad del sistema. ....	58
A.1	Uso de versiones Android en Julio de 2017. ....	76
A.2	Distribución de las distintas versiones Android en Enero de 2017. ....	77
B.1	Configuración del formulario. ....	79
B.2	Gráfico 1 de resultados del formulario. ....	80
B.3	Gráfico 2 de resultados del formulario. ....	80
B.4	Gráfico 3 de resultados del formulario. ....	81
B.5	Gráfico 4 de resultados del formulario. ....	81

B.6	Gráfico 5 de resultados del formulario. ....	81
B.7	Gráfico 6 de resultados del formulario. ....	82
B.8	Gráfico 7 de resultados del formulario. ....	82
B.9	Gráfico 8 de resultados del formulario. ....	83
B.10	Gráfico 9 de resultados del formulario. ....	83
B.11	Gráfico 10 de resultados del formulario. ....	84
B.12	Gráfico 11 de resultados del formulario. ....	84
B.13	Gráfico 12 de resultados del formulario. ....	85
B.14	Gráfico 13 de resultados del formulario. ....	85
B.15	Gráfico 14 de resultados del formulario. ....	86
B.16	Gráfico 15 de resultados del formulario. ....	86
B.17	Gráfico 16 de resultados del formulario. ....	87
B.18	Gráfico 17 de resultados del formulario. ....	87
B.19	Gráfico 18 de resultados del formulario. ....	88
B.20	Gráfico 19 de resultados del formulario. ....	88
B.21	Gráfico 20 de resultados del formulario. ....	89
B.22	Información sobre la segunda parte del formulario. ....	89
B.23	Gráfico 21 de resultados del formulario. ....	89
B.24	Gráfico 22 de resultados del formulario. ....	90
B.25	Gráfico 23 de resultados del formulario. ....	90
B.26	Gráfico 24 de resultados del formulario. ....	91
B.27	Gráfico 25 de resultados del formulario. ....	91
B.28	Gráfico 26 de resultados del formulario. ....	92
B.29	Gráfico 27 de resultados del formulario. ....	92
B.30	Gráfico 28 de resultados del formulario. ....	93
B.31	Gráfico 29 de resultados del formulario. ....	93
B.32	Gráfico 30 de resultados del formulario. ....	93
B.33	Gráfico 31 de resultados del formulario. ....	94
B.34	Gráfico 32 de resultados del formulario. ....	94
B.35	Gráfico 33 de resultados del formulario. ....	95
B.36	Gráfico 34 de resultados del formulario. ....	95
B.37	Gráfico 35 de resultados del formulario. ....	95
B.38	Gráfico 36 de resultados del formulario. ....	96
B.39	Gráfico 37 de resultados del formulario. ....	96
B.40	Gráfico 38 de resultados del formulario. ....	97
B.41	Gráfico 39 de resultados del formulario. ....	97
B.42	Gráfico 40 de resultados del formulario. ....	98
B.43	Gráfico 41 de resultados del formulario. ....	98
B.44	Gráfico 42 de resultados del formulario. ....	99
B.45	Gráfico 43 de resultados del formulario. ....	99
B.46	Gráfico 44 de resultados del formulario. ....	100
B.47	Gráfico 45 de resultados del formulario. ....	100
B.48	Gráfico 46 de resultados del formulario. ....	101
C.1	Diagrama de Gantt con la planificación del proyecto. ....	104
G.1	Email recibido cuando se genera un informe de fallos. ....	113
G.2	Reporte. ....	114
G.3	Detalle del reporte. ....	114
H.1	Parte 1. ....	115
H.2	Parte 2. ....	116
H.3	Parte 3. ....	117

H.4	Parte 4. ....	118
H.5	Parte 5. ....	119
I.1	Parte 1. ....	121
J.1	Parte 1. ....	123
J.2	Parte 2. ....	124
J.3	Parte 3. ....	125
J.4	Parte 4. ....	126
J.5	Parte 5. ....	127
J.6	Parte 6. ....	128
J.7	Parte 7. ....	129

## Lista de tablas

A.1	Versiones de Android .....	75
K.1	Daros para el cálculo de la tasa de alcoholemia .....	131



# INTRODUCCIÓN

---

## 1.1. Contexto

Este proyecto, que se presenta como Trabajo Final de Máster (TFM) , tiene su origen en un trabajo final de grado que fue presentado a la comisión evaluadora de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid el día 11 de Julio de 2016 [1].

El trabajo que se presenta en este documento cubre las mejoras que fueron propuestas en su momento, así como gran parte del trabajo futuro que se planteó. La solución que se presentó como una aplicación para clientes, se presenta ahora como un sistema completo para la gestión de los negocios de hostelería, más concretamente los que pertenecen al área de restauración. En una primera versión se implementó una aplicación cuya funcionalidad se concentraba en el lado del cliente y ligeramente en los empleados. El nuevo sistema **The Waiter**, incluye la misma aplicación para clientes y empleados con multitud de mejoras que se detallarán a lo largo del documento, además de una nueva aplicación en Android que permite completar el ciclo aportando las funcionalidades correspondientes del lado del negocio.

Estas herramientas se han denominado **The Waiter Plus** y **The Waiter TPV**. Además, algunas funciones sencillas que se realizaban desde **The Waiter Plus**, se han introducido en los relojes inteligentes con otra herramienta conocida como **The Waiter Wear**.

## 1.2. Marco del proyecto

La situación actual de los negocios dedicados a la hostelería se centra en una atención personalizada al cliente mediante recursos humanos. El uso de tecnologías por parte de estos establecimientos es mínimo.

La atención personalizada que es proporcionada por estos negocios tiene una gran acogida entre sus clientes pero a veces el número de personal frente al número de solicitudes resulta insuficiente para atender todas estas de manera rápida y eficiente.

**The Waiter** se centra en solventar los problemas que surgen en los negocios más tradicionales puesto que son, por lo general, los que menos tecnologías utilizan, centrándose más en la atención personalizada. Estos negocios normalmente siguen un sistema simple para atender a los clientes: en la primera atención se recoge la solicitud del cliente y en la segunda se atiende esta solicitud de la forma correspondiente. Las herramientas tecnológicas que utilizan abarcan la gestión de los pedidos y los cobros.

Este sistema resulta lento cuando el número de clientes supera en demasía al número de empleados. El problema nace de la necesidad de los empleados de acudir, como mínimo, dos veces a la mesa para realizar un único servicio, lo que claramente provoca efectos muy negativos en el tiempo que transcurre entre la petición y la respuesta.

Además, la gestión del negocio es muy limitada, con la herramienta **The Waiter TPV** se pretende proporcionar al administrador del establecimiento un mayor número de información relevante sobre todo lo que ocurre en el mismo, así como información sobre las opiniones, valoraciones y comentarios de los clientes, información que en la gran mayoría de los sistemas existentes que han sido objeto de estudio por mantener similitudes con **The Waiter**, es inexistente.

### 1.3. Motivación

Escoger un tema sobre el que realizar un trabajo de esta magnitud no fue una tarea fácil en su día. El motivo por el que se decidió continuar con el proyecto presentado como TFG en lugar de empezar otro nuevo, es principalmente la inquietud por todas aquellas cosas que no pudieron llegar a formar parte de dicho proyecto, como las mejoras y el trabajo futuro.

En este caso el objetivo principal ha sido mejorar la aplicación Android existente, que permitía las gestiones propias de los clientes y empleados en un negocio de restauración. Además de incorporar lo que sin duda acapará la mayor parte de este trabajo, una herramienta Android destinada a la administración del negocio, lo que termina de conformar un sistema completo para la mejora de los negocios de restauración.

El objetivo del sistema es, por una lado, conseguir mayor rendimiento de los usuarios en las actividades cotidianas de mayor frecuencia y menor relevancia, optimizando así el tiempo que requieren determinadas tareas, básicamente, ahorrando a los usuarios esperas innecesarias. Por otro lado, se pretende agilizar y facilitar la tarea de gestión que lleva por detrás un establecimiento de este tipo.

En lo que se refiere al sistema, se podría pensar que es “otro más”, sin embargo, hay ciertos detalles que permiten distinguir a este árbol en medio del bosque. Estos detalles fundamentalmente se deben al enfoque del sistema, como se explicará a lo largo de este documento.

En los últimos años el número de tecnologías dedicadas a mejorar la gestión de los negocios de restauración ha aumentado considerablemente. Gracias en gran parte al crecimiento del número de negocios “de comida rápida” que son conocidos así por la agilidad en el servicio, debido en su mayoría al uso de herramientas electrónicas. Siendo así, se podría pensar que este proyecto propone una solución ya existente. Sin embargo mostraremos que esto no es así. El sistema que se presenta permite evitar determinadas esperas en los negocios dedicados a la hostelería que tanto se frecuentan a diario. Esperas como por ejemplo, aquellas que se producen cuando los clientes esperan a ser atendidos para realizar una solicitud. Además de facilitar las gestiones de un administrador aumentando la cantidad de información sobre el negocio que se podrá almacenar en un dispositivo móvil de manera automática.

El resultado de este planteamiento ha sido el desarrollo de un sistema que pueda llevarse a la práctica obteniendo resultados que denoten la mejora del rendimiento esperado. Esto supone una mejora sobre las funciones que pueden desempeñarse en los negocios de restauración, haciendo que las herramientas sean lo más sencillas, accesibles y atractivas posible. Además de esto, el sistema **The Waiter** presenta costes muy bajos para su implantación que pueden potenciar su uso en detrimento de otros medios mucho más costosos y actualmente en uso.

### 1.4. Objetivos

Los objetivos de este proyecto son, por tanto, la incorporación de mejoras sobre el proyecto presentado como Trabajo Final de Grado (TFG) y el diseño e implementación de una herramienta que permita realizar una gestión más eficiente por parte de los administradores de los negocios de restauración.

Todos estos objetivos se concentran en **The Waiter**. **The Waiter** es el sistema de “Gestión de negocios de hostelería” desarrollado para este trabajo de fin de máster.

El sistema es capaz de gestionar todas las peticiones, valoraciones y comentarios de los clientes. A su vez, los camareros pueden recibir en sus dispositivos todas estas peticiones lanzadas por los clientes para ir atendiéndolas. Ambos perfiles obtienen la información necesaria para utilizar los aplicativos y unirse a la red del establecimiento de unos códigos Quick Response (QR) que son proporcionados por el administrador. El resto de información necesaria sobre el negocio debe ser configurada por el administrador en el correspondiente aplicativo, esta información será almacenada en una base de datos SQLite, y el resto de usuarios del sistema harán las solicitudes correspondientes haciendo uso del protocolo de comunicación. Todo esto con una interfaz gráfica desarrollada en Android que permite a los usuarios utilizar las funcionalidades que ofrece la aplicación de forma intuitiva.

## 1.5. Estructura del documento

**Capítulo 2.** En este capítulo se realiza un análisis del estado del arte, en el que se detalla la situación actual de los negocios de hostelería, las herramientas que existen en la actualidad similares a la que se presenta en este proyecto, una breve introducción sobre Android, la base fundamental de este trabajo y un pequeño estudio de mercado que nos permite conocer la situación del mercado actual y evaluar la necesidad que existe de este proyecto.

**Capítulo 3.** En este capítulo se detallan los objetivos y funcionalidades del proyecto.

**Capítulo 4.** En este capítulo se define el sistema que se ha diseñado. Para ello se detalla la metodología que se ha utilizado en el desarrollo así como todas las herramientas que han formado parte del mismo.

**Capítulo 5.** En este capítulo se hace un detalle del análisis realizado incluyendo el catálogo completo de requisitos y los casos de uso para los usuarios de la aplicación.

**Capítulo 6.** En el capítulo 6 se desarrollan el diseño y la implementación. Para ello se detallan la arquitectura utilizada y sus componentes, el diseño de la base de datos, el protocolo de comunicación, la navegación entre pantallas, etc.

**Capítulo 7.** En este capítulo se hace un análisis de las pruebas que se han realizado sobre la aplicación.

**Capítulo 8.** En este capítulo se evalúa el resultado final por parte del equipo de desarrollo y los usuarios escogidos y se identifican los beneficios que aporta la aplicación.

**Capítulo 9.** En este capítulo, como punto final, se detallan las conclusiones obtenidas del proyecto así como un breve planteamiento sobre el trabajo futuro.



## ESTADO DEL ARTE

---

### 2.1. Negocios de hostelería

Negocios de hostelería es el nombre genérico que reciben las actividades económicas consistentes en la presentación de servicios ligados al alojamiento y la alimentación, usualmente prácticas ligadas al turismo.

Se ha hecho una distinción de estos negocios en dos tipos según el número de tecnologías que utilizan para realizar los servicios que ponen a disposición de los clientes.

#### 2.1.1. Tradicionales

Los negocios tradicionales se valen de muy pocas tecnologías para mejorar los servicios que ofrecen, en su lugar se utilizan muchos recursos humanos. **The Waiter** es una solución orientada a aquellos negocios de hostelería que ofertan servicios de alimentación, como pueden ser los restaurantes o los bares. La mayoría de estos negocios incorporan una tecnología denominada TPV (Terminal Punto de Venta) que agiliza las tareas de atención a los clientes mediante la acumulación de los pedidos, la elaboración de facturas rápidas y el almacenaje de datos relevantes para el establecimiento, por ejemplo los productos más solicitados.

#### 2.1.2. De comida rápida

Este tipo de negocios incorpora cada vez un mayor número de tecnologías pero se orientan en que los clientes hagan su propio pedido ahorrándose así recursos humanos pero dificultando en muchas ocasiones la acción de los usuarios, como sucede en el caso de las personas mayores con dificultades para entender la tecnología y utilizar herramientas tecnológicas. Benefician la agilidad del negocio pero ponen en duda la comodidad de los clientes. Es por ello que aunque muchos negocios han incorporado los auto-servicios, siguen manteniendo el servicio tradicional.

**The Waiter** en la actualidad está más enfocado a los negocios tradicionales puesto que, por lo general, en los establecimientos donde se ofrece un "servicio rápido" no existe atención personalizada en las mesas.

### 2.2. Aplicaciones de gestión hostelera y sus funciones típicas

En la actualidad existe una amplia gama de aplicaciones que comparten algunos de los objetivos de este proyecto. Se citan las que se han considerado más relevantes y cuya funcionalidad tiene más similitudes con la

funcionalidad de **The Waiter**.

**The Waiter** se implementó en su primera versión con un enfoque directo hacia el cliente, sin embargo, en la actualidad, la solución presenta diferentes herramientas enfocadas tanto a la parte del cliente como a la parte del negocio y sus empleados. Por ello las herramientas que se citan a continuación se dividen en dos enfoques, en función de si están orientadas a los clientes o al negocio.

### 2.2.1. Destinadas al cliente

#### Dual Link B&R Comandero



Esta aplicación permite que los camareros de un restaurante, bar, pub o cualquier otro negocio de hostelería tengan una herramienta eficaz para realizar los pedidos así como sincronizarlos automáticamente con la aplicación de Terminal Punto de Venta (TPV), agilizando de este modo su trabajo y permitiéndoles mandar el pedido de forma seleccionada, parcial y completa.

Con Dual Link B&R Comandero los camareros tienen la oportunidad de solicitar los pedidos de los clientes a la cocina y a la barra, además, pueden seleccionar si la comida que se va a servir ha de mantenerse en frío o en caliente. También puede recibir y marcar los pedidos realizados por los clientes que estén usando la aplicación "Carta" de la Suite de Ocio y Restauración y efectuar el pago directamente desde el propio terminal (con tarjeta o móvil).

#### iD Bares



Esta aplicación sirve para interactuar con los locales de ocio iD. A través de ella, el usuario puede pedir comandas o bebidas desde la mesa utilizando un iPad incluido en la misma o desde su propio smartphone.

Desde la aplicación, el cliente, además, puede realizar reservas, consultar el menú, seleccionar cada ingrediente de sus consumiciones, conocer el gasto acumulado, individualizar los pagos y obtener descuentos en función de su participación en el local. Además dispone de una sección de juegos y retos.

#### MyOrder



Desarrollada por la empresa Atecresa en Canarias, con ella los usuarios disponen de un sistema para evitar incómodas y largas esperas en los bares y restaurantes, ya que con esta aplicación se puede realizar un pedido a través de códigos QR.

Gracias a la aplicación "MyOrder" se puede realizar el pedido desde el móvil o Smartphone, sin tener que esperar a que el camarero tome nota.

#### qlikBar



Esta aplicación es capaz de conocer el local en el que se encuentra el usuario y darle la posibilidad de consultar el menú, interactuar con el resto de clientes, dejar comentarios y opiniones y realizar pedidos. Ello permite a los clientes que el tiempo que pasen en el bar o restaurante sea más social y divertido, además de que la aplicación es también una guía de locales para ayudar a descubrir sitios nuevos y un juego en el que los usuarios acumulan puntos por visitas y consumos en los locales. Los usuarios de smartphone se puede descargar de forma gratuita la aplicación.

## Coaster



Coaster, es una startup, idea de Inderpal Singh y de Kevin Callaghan, que cansados de esperar por sus bebidas cuando salían con sus amigos, decidieron crear un aplicación para pedir bebidas, que solucionara su problema, agilizando el servicio en el bar.

## 2.2.2. Destinadas al negocio

### GlopDroid



La app es un sistema de toma de comandas muy funcional e intuitivo, ya que aprovecha la calidad gráfica de smartphones y tablets para ofrecer una forma de trabajar más cómoda y ágil en restaurantes, cafeterías y otros locales de hostelería. Permite a los camareros tomar nota de las comandas en dispositivos móviles en lugar de en papel.

Puede gestionar comentarios a cocina, enviar mensajes con sugerencias, seleccionar los platos para un menú o realizar traspasos entre mesas, entre otras muchas opciones. Además controla los salones colocando mesas o cobrando las mismas por pago fraccionado si así lo desean los clientes y gestiona tantas impresoras de cocina como sean necesarias. Equipado tan sólo con una red wifi y la aplicación de Glop para hostelería, proporciona una solución de cara a la atención al cliente.

### Dual Link B&R TPV



Implementa las funcionalidades más importantes de un TPV pero ofreciendo una interfaz sencilla y usable que puede ser incorporada en cualquier dispositivo con sistema iOS.

Esta aplicación permite a los usuarios gestionar los pedidos por secciones: localizados y no localizados. También es capaz, por ejemplo, de asignar puntos VIP para los clientes más importantes.

Los pedidos localizados se administran gracias a un plano personalizable de las mesas, con el cual se pueden juntar mesas o distribuir las a semejanza del local. Los pedidos no localizados conciernen a los taburetes y puestos en la barra, cuyos clientes estarán atendidos gracias a esta aplicación.

### Philomarket



Es una app gratuita para hacer los pedidos a proveedores desde cualquier móvil o tablet. También se puede acceder a la web desde cualquier dispositivo.

### QR Question



Con QR Question los hosteleros tienen acceso de forma instantánea a los comentarios y valoraciones de sus clientes. Los comensales, por su parte, acceden a una breve encuesta a través de una URL o un código QR, y desde el mismo restaurante valoran el servicio.

## 7Shifts



7Shifts es una app para administrar restaurantes. En este caso, 7shifts viene perfecta para organizar turnos de trabajo entre los empleados del lugar. Con la aplicación se puede: organizar horarios y turnos de trabajo con facilidad, mejorar la comunicación entre el equipo, reducir costes y horas extra y permite solicitar turnos u horas de trabajo y aprobarlas o rechazarlas.

Genera los turnos de trabajo de forma semiautomática en apenas unos minutos, y estos se envían a quien corresponde a través del e-mail y mensajes. Con esta aplicación, se puede conocer la disponibilidad de los empleados, y ellos por su parte, podrán comunicarse de forma mucho más eficaz con el administrador y sus compañeros.

## Deluxrest



Deluxrest se ha convertido en una de las mejores aplicaciones para restaurantes del momento. Es una app completamente personalizable que permite al dueño del restaurante muchas opciones, como por ejemplo tomar reservas, enviar ofertas, mostrar el menú o la carta, enseñar las facilidades y ubicación, entre otras.

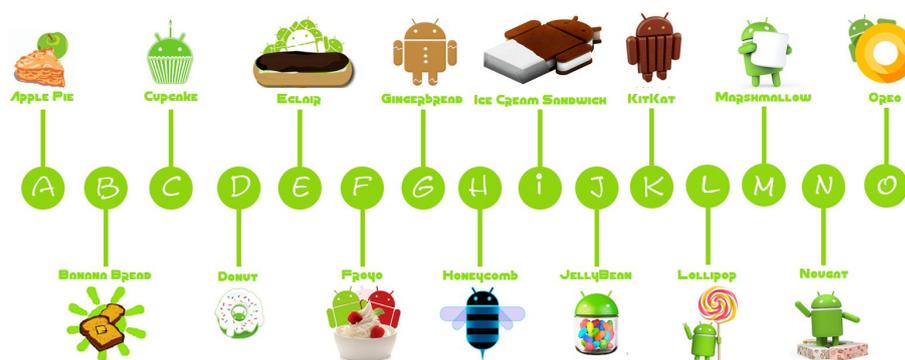
Tiene notificaciones push, una ventaja para estar al momento actualizado de las peticiones de los clientes.

## 2.3. Android

Android, entre sus muchas definiciones, es un sistema operativo para dispositivos móviles basado en Linux.

El SDK de Android proporciona las herramientas necesarias para desarrollar aplicaciones Android utilizando el lenguaje de programación Java. Android fue desarrollado originalmente por una startup del mismo nombre que fue comprada por Google en 2005. Cualquiera puede empezar a utilizar Android descargando su código completo. Además, los fabricantes pueden añadir sus extensiones propietarias, lo cual ha hecho crecer considerablemente la aceptación de la plataforma.

La historia de Android comenzó con la publicación de Android beta en noviembre de 2007. La primera versión comercial apareció en febrero de 2009. Desde abril de 2009, las versiones reciben un nombre inspirado en un postre o un dulce, en orden alfabético (tabla A.1, figura A.1).



**Figura 2.1:** Versiones Android desde 2008 a 2016.

## 2.4. Estudio de mercado

El estudio de mercado [2] es una parte importante de la investigación de mercado que consiste en la medición de los sentimientos y las preferencias de los clientes en un mercado determinado. El estudio de mercado, el cual varía en tamaño, diseño y propósito, es una de las principales piezas de información que las organizaciones utilizan para determinar que productos y servicios ofrecerán, y cómo los comercializarán.

### 2.4.1. Objetivo

En el caso de este proyecto, el objetivo del estudio es conocer la necesidad real que existe en el mercado de incorporar un sistema de estas características, así como la aceptación que recibiría y la disposición para implantarlo en un negocio y usarlo al visitar dichos negocios.

### 2.4.2. Naturaleza, magnitud y tamaño del mercado

Es fundamental definir el mercado hacia el cual se enfoca el estudio. En este caso se ha seleccionado como área geográfica España, prestando especial atención en su capital, Madrid.

### 2.4.3. Aspectos del mercado

The Waiter se trata de un producto nuevo para el mercado por lo que uno de los aspectos más interesantes del estudio consiste en averiguar la necesidad que existe en el mercado actual del producto o que tan deseado es.

### 2.4.4. Dónde y cuándo llegar a los clientes del mercado

Este apartado supone una parte fundamental del estudio, en el se concentran las claves para hacer llegar este proyecto a los usuarios objetivo.

En nuestro caso nos hemos concentrado en dar a conocer el sistema en negocios de restauración muy frecuentados en determinadas franjas horarias. Como son, por ejemplo, los restaurantes cercanos a zonas industriales en la hora de la comida, bares cercanos a las universidades en los horarios de finalización de las clases, etc.

Además, se ha hecho uso de una gran variedad de redes sociales que han permitido captar a un gran número de usuarios potenciales.

### 2.4.5. Tipo de estudio empleado

Para llevarlo a cabo, se ha elaborado una encuesta de Google Forms, publicada el 7 de Junio de 2017. El formulario fue cerrado el 7 de Diciembre de 2017, 6 meses más tarde, con un total de 168 respuestas únicas, para ello el formulario fue configurado para solicitar el inicio de sesión con una cuenta de Google, lo que permitía asegurar que cada participante emitiría una única respuesta.

En este enlace se puede encontrar el formulario de Google utilizado para realizar la encuesta: <https://goo.gl/forms/4dGAfWKgPPdWGDRx2>.

### 2.4.6. Resultados

Finalmente el volumen de información recogida ha sido suficiente para analizar la carencia del mercado actual y por lo tanto la necesidad de un sistema como el que se propone en este trabajo, así como lo que se esperaría de dicho sistema.

En este apartado se expondrá aquella información que ha resultado más importante para el objeto de estudio, pero el informe completo de resultados puede consultarse en el anexo B.

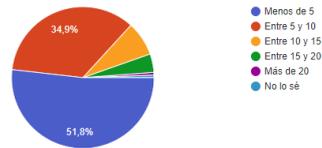
## Contexto de los clientes

En la **figura 2.2** podemos observar un resumen de los gráficos que mejor se ajustan para describir el contexto actual de los clientes objetivo de este proyecto.

Como se puede observar, el número de visitas semanales a estos negocios es alto. Por otro lado, el tiempo de servicio resulta elevado si lo multiplicamos por el número de visitas en una semana. Casi un 70 % de los encuestados afirma que en los establecimientos a los que acude no le permiten realizar peticiones desde un dispositivo móvil.

Indique el número de visitas que realizó a negocios de restauración en los 7 días anteriores a la realización de esta encuesta

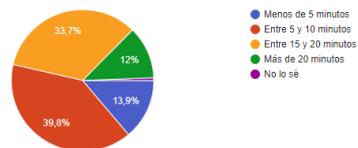
166 respuestas



(a) Gráfico 1

Indique el tiempo medio hasta que el servicio le entregó su pedido completo

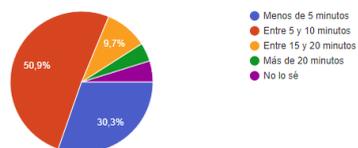
166 respuestas



(b) Gráfico 2

Indique el tiempo medio que esperó para pedir la cuenta más el tiempo que tardó, una vez pedida, en realizar el pago y recibir su cambio si fuera necesario

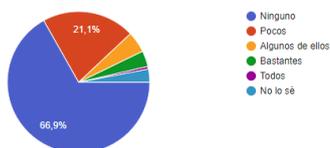
165 respuestas



(c) Gráfico 3

¿Los sitios que frecuenta permiten que realice las peticiones desde su dispositivo móvil?

166 respuestas



(d) Gráfico 4

**Figura 2.2:** Gráficos de resultados del formulario en relación con el contexto para los clientes

## Impacto de The Waiter para los clientes

En la **figura 2.3** podemos observar un resumen de los gráficos que mejor se ajustan para describir el impacto que se pretende que este proyecto cause en los clientes objetivo de este proyecto.

Un 50 % de la muestra garantiza que les gustaría disponer de la posibilidad de realizar pedidos desde su dispositivo móvil, mejorando su experiencia. Además de poder enviar otros datos relevantes como valoraciones y comentarios. Una destacable mayoría de los usuarios que han recibido información sobre el sistema afirman que implantar el sistema supondría una mejora y harían uso del mismo.

## Contexto del negocio

En la **figura 2.4** podemos observar un resumen de los gráficos que mejor se ajustan para describir el contexto actual de los negocios objetivo de este proyecto.

El dispositivo que, digamos, sustituye al aplicativo principal de este proyecto se considera “difícil de usar” por casi un 30 % de los usuarios, además, no permite gestionar información sobre el inventario, los proveedores o las valoraciones y comentarios de los clientes.

## Impacto de The Waiter para los negocios y sus empleados

En la **figura 2.5** podemos observar un resumen de los gráficos que mejor se ajustan para describir el impacto que se pretende que este proyecto cause en los negocios objetivo de este proyecto.

De nuevo, la mayoría de los usuarios que han participado, corroboran que la implantación del sistema supondría una mejora para los establecimientos en los que trabajan, y aquellas personas con un rol de administrador se notan predispuestas a usar dicho sistema. Considerando, en rasgos generales, que esto es algo positivo para el negocio. A su vez, un porcentaje muy elevado de empleados aseguran que les gustaría recibir las peticiones del modo que les ofrece esta solución.

### 2.4.7. Conclusiones

En definitiva, pese al pequeño tamaño del estudio, se ha obtenido información verdaderamente útil para este proyecto. Pues bien, el objetivo de este apartado no era elaborar un estudio de mercado completo, si no aproximarnos a un estudio de mercado real para entender la necesidad que pudiera existir en el mercado de una solución como **The Waiter**.

## 2.5. Opiniones y conclusiones

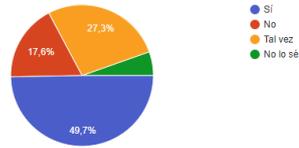
Las aplicaciones citadas anteriormente son solo unas pocas de las que existen actualmente en el mercado, en concreto las más similares que se han encontrado al proyecto que se presenta.

Tras una larga investigación sobre la aplicación de estas herramientas se ha advertido que pese a sus múltiples ventajas no han tenido apenas éxito y es una idea muy poco explotada por los negocios de restauración.

En su mayoría, esto se debe a la escasa aceptación por parte de los negocios de introducir nuevas herramientas, bien por los gastos económicos o bien por un temor frecuente a las nuevas tecnologías. Por ello, estas aplicaciones, que en cierto modo pretenden cambiar todo el modelo de gestión del negocio y aunque resulten una mejora, se enfrentan al rechazo frente a métodos más tradicionales.

¿Considera que utilizar el teléfono móvil para las funciones descritas anteriormente agilizaría el servicio de los establecimientos y mejoraría su experiencia como cliente?

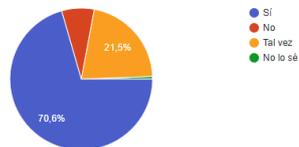
165 respuestas



(a) Gráfico 1

¿Le gustaría poder enviar al establecimiento sus valoraciones y comentarios sobre el servicio recibido, los productos consumidos, etc.?

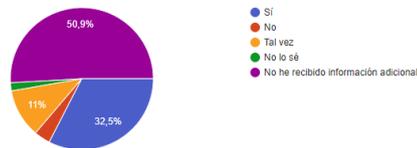
163 respuestas



(b) Gráfico 2

Si ha recibido información sobre el sistema que se ha implementado para este proyecto, ¿cree que supondría una mejora implantarlo en los negocios?

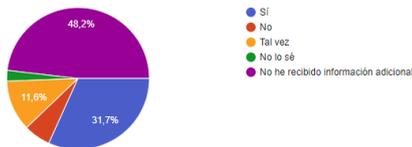
163 respuestas



(c) Gráfico 3

De nuevo, si ha recibido información, en su caso personal, ¿haría uso del sistema?

164 respuestas

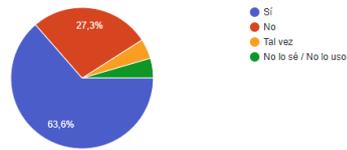


(d) Gráfico 4

**Figura 2.3:** Gráficos de resultados del formulario en relación con el impacto para los clientes

¿Considera que el dispositivo principal que utilizan (caja registradora, TPV...) es sencillo de usar?

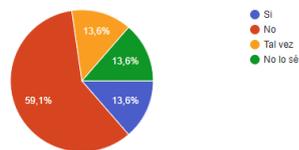
22 respuestas



(a) Gráfico 1

El dispositivo, ¿permite llevar una gestión detallada del inventario del establecimiento?

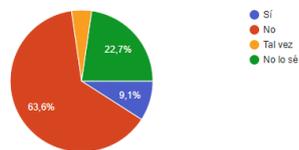
22 respuestas



(b) Gráfico 2

El dispositivo, ¿puede gestionar los proveedores de los productos del establecimiento?

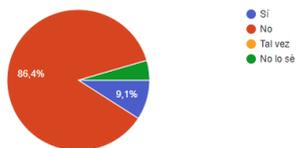
22 respuestas



(c) Gráfico 3

El dispositivo, ¿puede gestionar las valoraciones y comentarios de los clientes?

22 respuestas

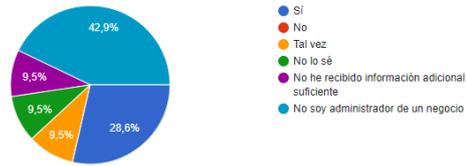


(d) Gráfico 4

**Figura 2.4:** Gráficos de resultados del formulario en relación con el contexto para el negocio

En caso de ser el administrador de un negocio de este tipo y haber recibido la información suficiente, ¿integraría este sistema en su establecimiento?

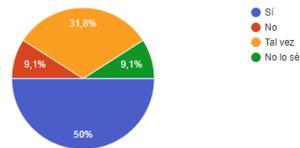
21 respuestas



(a) Gráfico 1

¿Considera que sería más rápido si pudiera recibir todas las peticiones de los clientes en su teléfono móvil?

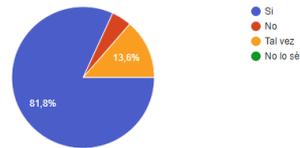
22 respuestas



(b) Gráfico 2

¿Considera que incorporar tecnología en un negocio de este tipo es algo positivo para el mismo?

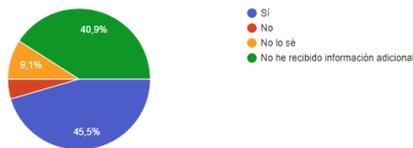
22 respuestas



(c) Gráfico 3

Si ha recibido información sobre el sistema que se ha implementado para este proyecto, ¿cree que supondría una mejora implantarlo en su negocio?

22 respuestas



(d) Gráfico 4

**Figura 2.5:** Gráficos de resultados del formulario en relación con el impacto para el negocio

Es por ello que la solución **textitThe Waiter** requiere una inversión inicial prácticamente nula, el camarero podría usar su propio smartphone y los clientes utilizan sus propios dispositivos también. En cuanto a los temores por las nuevas tecnologías, se ha hecho un esfuerzo considerable por hacer una interfaz simple y usable que permita que tanto la parte del negocio como los clientes puedan utilizarla de forma intuitiva.

# OBJETIVOS Y FUNCIONALIDADES

---

## 3.1. Introducción

Como se ha descrito en el capítulo 1, el propósito de este Trabajo de Fin de Máster es definir e implementar una solución software para la gestión de los negocios de restauración y la comunicación entre estos negocios y sus clientes. Para cumplir con estos objetivos, se detallan a continuación las funcionalidades que dan forma a **The Waiter**.

## 3.2. Objetivos específicos y funcionalidades

### 3.2.1. Funcionalidades

En este apartado se definen las funciones básicas del proyecto. El propósito de estas funcionalidades es dar solución a las necesidades que originan el proyecto y cubrir los objetivos planteados para el mismo. En la solución se incluyen tres aplicativos que permiten solucionar las necesidades de los diferentes perfiles que existen en el sistema, esto es: cliente, camarero y administrador. Por ello las funcionalidades se dividen según estos perfiles, para que podamos comprender por separado la funcionalidad de las diferentes herramientas que conforman el sistema.

#### Cliente

##### F-1.– Envío de peticiones.

**F-1.1.– Limpieza en mesa.** Se solicitará que el camarero acuda para limpiar el espacio de los clientes.

**F-1.2.– Atención.** Se solicitará que el camarero acuda a la mesa para realizar una petición personalizada.

**F-1.3.– Cuenta.** El cliente solicitará la cuenta.

**F-1.4.– Pago en metálico.** El cliente solicitará al camarero que acuda a recoger el pago en metálico.

**F-1.5.– Pago electrónico.** El cliente solicitará al camarero que acuda a la mesa con los medios necesarios para llevar a cabo un pago con tarjeta.

**F-1.6.– Pedido.** Tras ir añadiendo productos al carrito, el usuario podrá solicitar el pedido completo al camarero.

**F-1.7.– Pedido personalizado.** En este pedido se enviará al camarero un mensaje personalizado por si fuese necesaria la modificación de alguno de los productos de la carta o bien éste no estuviese incluido.

**F-1.8.– Hoja de reclamaciones.** Los clientes podrán solicitar la hoja de reclamaciones, servicio que debe estar disponible en todos los establecimientos.

**F-2.– Listado de la carta [3–5].** El cliente tendrá acceso a un listado clasificado de los productos que pertenecen a la carta del negocio. En estos listados podrá ir añadiendo y eliminando productos al pedido.

**F-3.– Gestión del pedido.** En el pedido se mostrarán aquellos elementos que se han ido añadiendo desde la funcionalidad de carrito. Cuando se muestre el listado de pedido también el usuario podrá añadir o eliminar elementos y finalmente podrá enviar este pedido al camarero.

**F-4.– Gestión del perfil de usuario.** Los usuarios podrán modificar sus datos personales. Alguno de ellos, como el alias, serán utilizados en ciertas funcionalidades públicas de la aplicación, como el chat.

**F-5.– Envío de valoraciones y sugerencias.** Los clientes podrán valorar y hacer sugerencias que ayuden al negocio a mejorar sus servicios.

**F-6.– Escaneo de códigos QR .** La aplicación necesita conocer datos sobre el negocio así como las notificaciones que recibe un camarero necesitan incluir información sobre la localización del cliente y para ello deben escanear un código QR situado, por ejemplo, en la mesa en la que se sienten.

**F-7.– Comunicación en red local [6, 7].** Los usuarios conectados a la red local del establecimiento podrán iniciar con los otros usuarios conectados, un chat público o una mensajería instantánea.

**F-8.– Realización del test de alcoholemia [8, 9].** Los clientes podrán rellenar un test que permite estimar la tasa de alcoholemia y así poder garantizar, en la medida de lo posible, la seguridad de los mismos.

## Camarero

**F-1.– Escaneo de códigos QR .** La aplicación necesita conocer datos sobre el negocio, como los datos para el acceso a la red. Para ello los camareros reciben del administrador un código con la información necesaria para trabajar correctamente con el aplicativo.

**F-2.– Atención de peticiones.** Las peticiones realizadas por los clientes se añadirán a un listado al que sólo tienen acceso los camareros. De este modo el camarero conocerá la naturaleza de cada petición para poder atenderla adecuadamente, así como la mesa desde la que se envió la petición y la prioridad de la misma en función del orden en el listado y el número de veces que el cliente haya realizado la misma petición.

## Administrador

**F-1.– Configuración.**

**F-1.1.– Usuarios y contraseñas.** Este usuario podrá crear, modificar y eliminar aquellos usuarios que pertenezcan a los grupos empleados o administradores.

**F-1.2.– Red.** El administrador podrá configurar aquellos parámetros de red necesarios

para que tanto los trabajadores como los clientes estén conectados a la red del establecimiento.

**F-1.3.– Creación de códigos QR .** Los clientes y empleados de un establecimiento deberán escanear estos códigos para conocer información necesaria para desempeñar correctamente su actividad con la aplicación. Para ello es necesario que el administrador genere estos códigos, añadiendo la información necesaria en cada momento.

#### **F-2.– Gestión de productos.**

**F-2.1.– Proveedores.** El administrador podrá añadir todos los proveedores que necesite para realizar los pedidos. Así mismo, todos los productos del inventario estarán asociados al proveedor que los proporciona para facilitar los pedidos cuando sea necesario.

**F-2.2.– Inventario.** En la funcionalidad de inventario se creará un listado configurable por el administrador que permitirá añadir, editar y eliminar todos los productos base que formarán parte de los productos de la carta. Además, cada uno de los productos almacenará información sobre el precio, alérgenos, proveedor que lo facilita, foto, etc.

**F-2.3.– Carta.** En la funcionalidad de carta el administrador podrá configurar un listado de productos a partir de los productos del inventario. Cada uno de estos productos incluirá un listado de los productos del inventario que contiene, alérgenos, cantidades, precio, foto, etc.

Este listado de productos será el que verá el cliente (ver funcionalidades del Cliente 2 y 3).

#### **F-3.– Gráficos.**

**F-3.1.– Valoraciones.** El administrador tendrá acceso a una serie de gráficos que le permitirán conocer las valoraciones que se han recibido sobre el establecimiento, el servicio y los empleados.

### **3.2.2. Objetivos**

Las funcionalidades citadas anteriormente tienen como propósito cumplir los objetivos que se detallan en este apartado. Los objetivos del sistema son muchos pero pueden agruparse en los objetivos genéricos que se citan a continuación:

**O-1.– Facilitar la administración de un establecimiento.** La herramienta *The Waiter TPV* está pensada para facilitar las tareas que un administrador puede llevar a cabo en un establecimiento de restauración, además de añadir funcionalidades adicionales para permitir la gestión de sus empleados y el conocimiento de las opiniones y valoraciones de sus clientes.

**O-2.– Minimizar el tiempo de espera de los clientes.** Esto se consigue haciendo que los clientes puedan realizar peticiones desde el teléfono móvil sin esperar a que un empleado del establecimiento les atienda para escuchar su solicitud y poder atenderla.

**O-3.– Facilitar la realización de los pedidos.** *The Waiter Plus* y *The Waiter WEAR* son los dos

aplicativos incorporados en el sistema que permiten a los clientes realizar los pedidos de una forma rápida y sencilla, desde su dispositivo móvil o smartwatch.

**O-4.– Permitir la comunicación privada entre los clientes que se encuentren conectados a la red local del establecimiento.** El protocolo de comunicación interno que permite el envío y recepción de peticiones también se utiliza para ofrecerle a los clientes un chat público para todos aquellos que estén conectados a la red local del establecimiento. A parte de esto, **The Waiter** ofrece la posibilidad de iniciar hilos de comunicación privada uno-a-uno entre clientes que estén utilizando el servicio.

**O-5.– Agilizar las tareas del camarero.** El camarero no tendrá que ir mesa por mesa anotando las peticiones, sino que las recibirá en un dispositivo móvil. Para atender estas peticiones **The Waiter** ofrece un sistema de prioridades pero la decisión final de como gestionarlás recae en los empleados. En posteriores capítulos se explicará esto en más detalle.

**O-6.– Evitar las pérdidas de las peticiones.** Las peticiones realizadas por los clientes se almacenan en un listado al que tienen acceso los camareros de forma permanente, de modo que hasta que el camarero no marque una petición como satisfecha, ésta no será eliminada de la lista. Es decir, el camarero tendrá el control sobre todas las peticiones que hayan sido realizadas en todo momento, recibéndolas en tiempo real.

**O-7.– Conseguir una interfaz sencilla y atractiva.** Éste es uno de los objetivos más importantes. Puesto que no se aporta manual para los usuarios, sólo una breve comunicación con aquellos negocios que quieran implantar este sistema en sus establecimientos, se ha perseguido que la interfaz sea lo más intuitiva posible para que los usuarios conozcan sus posibilidades en todo momento y puedan hacer una navegación fácil entre las pantallas de la aplicación.

# DEFINICIÓN DEL PROYECTO

---

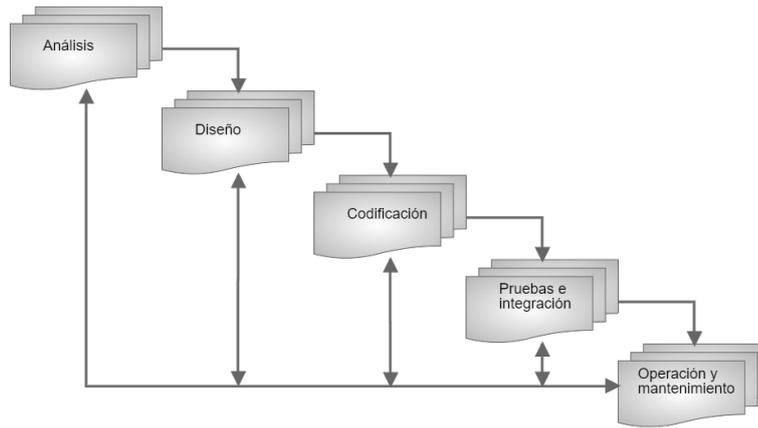
## 4.1. Metodología

En este apartado se definirán en detalle, y en orden de ejecución, cada una de las etapas que se han seguido durante el desarrollo del proyecto. Las fases del proyecto coinciden con las que normalmente se utilizan en la mayoría de proyectos software:

- 1.– **Análisis.** En esta fase la tarea principal es la captura de requisitos. Estos deben definir con exactitud el funcionamiento de la aplicación. Esto también incluye requisitos no funcionales.
- 2.– **Diseño.** Esta etapa está dedicada a realizar el diseño de las aplicaciones. Esto incluye el diseño de pantallas, navegación, base de datos, protocolo de comunicación y otros elementos menos relevantes para las aplicaciones. También se ha hecho el diseño de la estructura de clases, puesto que, en su base, las aplicaciones se han desarrollado en Java.
- 3.– **Implementación.** Esta etapa es, con diferencia, la más larga de las que componen el proyecto. En ella se codifican los diseños que se han definido en la fase anterior. También se realizan en esta etapa las pruebas correspondientes a la implementación de cada tarea, aunque el modelo de ciclo de vida en cascada con realimentación utilizado nos ha permitido volver a esta fase para realizar las modificaciones que se hayan considerado oportunas.
- 4.– **Pruebas.** En la etapa anterior se han realizado muchas de las pruebas previstas puesto que nos permitía cerrar la fase de implementación con el menor número de errores posible. En esta etapa se han completado las pruebas restantes y se han revisado las que ya se habían realizado para asegurar la funcionalidad. También se ha aprovechado esta fase para documentar las pruebas realizadas y los módulos que por su completo y correcto funcionamiento se han considerado cerrados.
- 5.– **Documentación.** En las etapas anteriores se ha ido realizando un gran trabajo de documentación pero es en esta fase cuando hemos reunificado todo para dar lugar a la presente documentación. En ella se recogen los aspectos más importantes del proyecto y se crea una relación entre los elementos de cada una de las fases. Finalmente el documento recoge la evaluación, extensibilidad, trabajo futuro y conclusiones obtenidas del proyecto.
- 6.– **Mantenimiento.** Esta fase es difícil de definir en el tiempo, puesto que depende de múltiples factores. Durante ella se mejora, actualiza y modifica aquello que se considere necesario en la aplicación. En el caso de este proyecto no se ha trabajado sobre esta etapa, aunque resultará necesaria para el trabajo futuro y por ello se detallarán algunas tareas de mantenimiento que se han previsto.

En el diagrama de Gantt del apartado 4.2.2 pueden observarse estas etapas divididas en tareas. También se puede observar la distribución temporal de estas tareas así como las relaciones que existen entre ellas, las cuales permiten solaparlas en las etapas de desarrollo.

Cómo ya hemos mencionado ligeramente en la fase de implementación, se ha seguido un modelo de ciclo de vida en cascada [10, Cap. 1]. Al constar únicamente de un recurso humano se ha decidido utilizar una variante que permite solapar el desarrollo de varias tareas pudiendo volver después a abrir cualquiera de las fases cuando se necesiten realizar modificaciones sobre ella. Esta variante es una variante iterativa (Figura 4.1) y con ella hemos conseguimos evitar limitaciones que se encontraron en el modelo clásico.



**Figura 4.1:** Ciclo de vida en cascada con realimentación.

## 4.2. Planificación orientativa

La planificación [11, Cap. 10] que se aporta a continuación es orientativa puesto que se realizó en la fase de planificación del proyecto y, aunque se ha hecho todo lo posible por seguirla, en ocasiones ha sufrido alguna desviación.

Los cambios, planteados en el diagrama de Gantt, han surgido tras las revisiones y es por ello que no se tuvieron en cuenta en la planificación inicial, sino que se incorporaron cuando fueron necesarios y así quedan especificados en el diagrama.

### 4.2.1. Puntos de revisión

Puesto que el equipo de desarrollo estaba compuesto únicamente por un recurso humano, las revisiones que vale la pena mencionar son las que se han realizado en conjunto con el tutor del proyecto.

- **Semanas 20-21.** En este punto de revisión se comprueban las definiciones y los diseños para dar paso a la fase de implementación de los diferentes componentes de la aplicación.  
Tras esta revisión fue necesario incluir modificaciones en la definición y la planificación del proyecto.
- **Semanas 33-35.** En esta revisión el objetivo es la comprobación de la correcta implementación de la base de datos así como la correcta construcción del protocolo de comunicación. Una vez comprobado esto, se realizan las revisiones oportunas sobre la implementación de la aplicación destinada al negocio, sucedidas de unos tests sobre la aplicación para clientes y empleados.

Después de esta revisión se han llevado a cabo modificaciones sobre las implementaciones de la funcionalidad para la gestión del negocio.

- **Semanas 43-45.** En esta tercera y última revisión el objetivo es comprobar el correcto funcionamiento de los elementos que componen el sistema. Por claridad, en el diagrama se hace únicamente referencia a tareas genéricas, las cuales engloban multitud de subtareas.

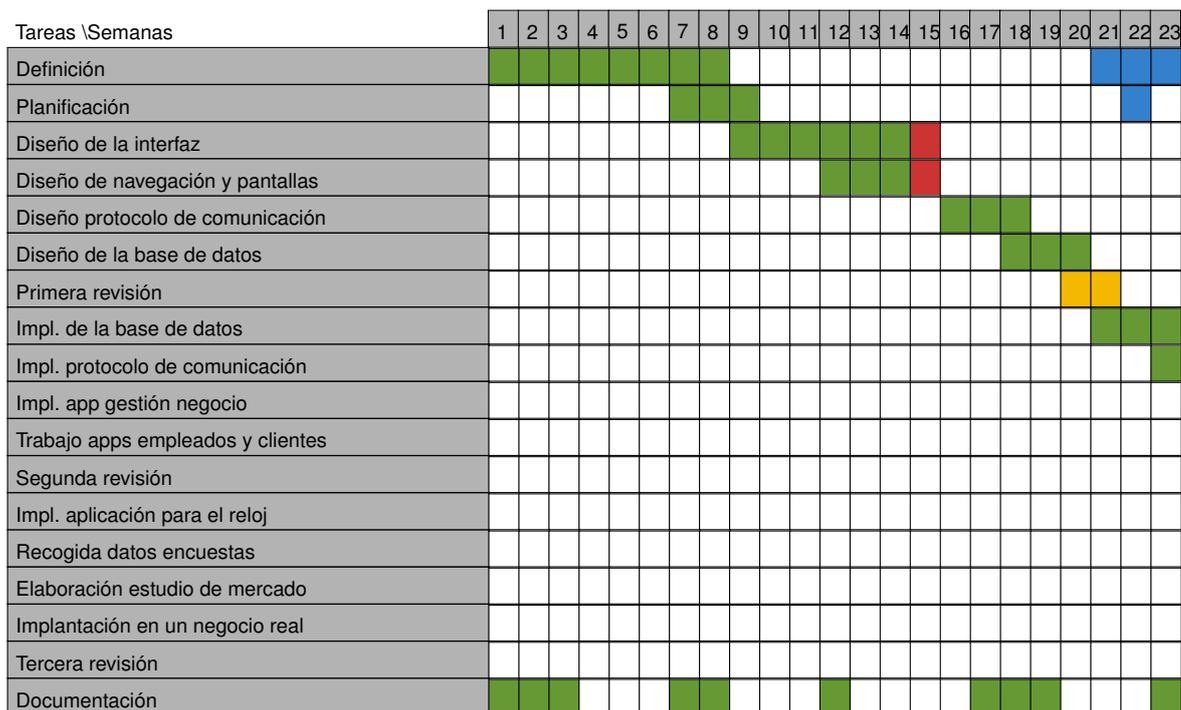
Tras esta revisión no han sido necesarias modificaciones y se procede a finalizar la etapa de documentación.

#### 4.2.2. Diagrama de Gantt

El diagrama [11, Cap. 8] se ha estructurado en dos partes que, además de facilitar la lectura, permiten distinguir dos fases que han sido claves en el proyecto:

- Análisis, planificación y diseño. En esta fase se produce la primera revisión.
- Implementación, recogida de resultados, pruebas y documentación. En esta fase se producen la segunda y última revisión.

Pese a esto, el hecho de tener 2 fases bien diferenciadas no implica que no haya habido interacción y relación entre ellas.



**Leyenda:** Desarrollo Revisión Cambios Pruebas

Tareas \ Semanas	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
Definición																							
Planificación																							
Diseño de la interfaz																							
Diseño de navegación y pantallas																							
Diseño protocolo de comunicación																							
Diseño de la base de datos																							
Primera revisión																							
Impl. de la base de datos																							
Impl. protocolo de comunicación																							
Impl. de las funcionalidades de gestión del negocio																							
Trabajo con la aplicación para empleados y clientes																							
Segunda revisión																							
Implementación aplicación para el reloj																							
Recogida de los datos de las encuestas																							
Elaboración estudio de mercado																							
Implantación en un negocio real																							
Tercera revisión																							
Documentación																							

**Leyenda:** Desarrollo Revisión Cambios Pruebas

## 4.3. Herramientas utilizadas

Para la implementación de los elementos que forman la aplicación se han utilizado entornos de trabajo, herramientas para la gestión de versiones, simuladores y lenguajes de programación. Estas herramientas se detallan a continuación.

### 4.3.1. Plataformas

#### Android Studio

Android Studio [12–18] es el entorno de desarrollo integrado que se ha escogido en este proyecto para desarrollar las aplicaciones. La elección de esta herramienta frente a otras se debe a que está desarrollada en su base para programar en Android, a diferencia de otras que permiten múltiples lenguajes.

#### Firebase

Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles desarrollada por James Tamplin y Andrew Lee en 2011 y adquirida por Google en 2014.

En este proyecto se ha utilizado esta herramienta para recibir los informes de fallos que se produzcan en los terminales en los que se instala la aplicación. También permite obtener una información general de las localizaciones en las que se instalan las aplicaciones, el número de instalaciones que se han hecho, el sistema operativo de los terminales en los que se ha instalado, etc.

#### Genymotion

Genymotion [19] es un completo emulador para aplicaciones Android. Ha permitido ejecutar la aplicación hasta en 5 dispositivos sin alterar el rendimiento de la misma. Esto ha resultado muy útil para simular situaciones lo más cercanas posibles a la realidad de los entornos en los que se utilizará la aplicación. También ha permitido comprobar compatibilidades de interfaz y versiones Android con 7 tipos diferentes de dispositivos y 5 versiones.

#### Bitbucket

Es un sistema de alojamiento basado en web para aquellos proyectos que utilizan el sistema de control de versiones Mercurial y Git [20].

En el caso de este proyecto se ha utilizado el sistema de revisiones Git. Esto ha permitido controlar y versionar todos los elementos del proyecto. Ha resultado muy útil para las revisiones y las modificaciones puesto que se ha tenido constancia de todos aquellos instantes en los que se ha subido una versión del proyecto con la correspondiente descripción. Esto ha permitido detectar rápidamente los módulos con errores, los que requerían modificaciones o los que se han expuesto en las correspondientes revisiones.

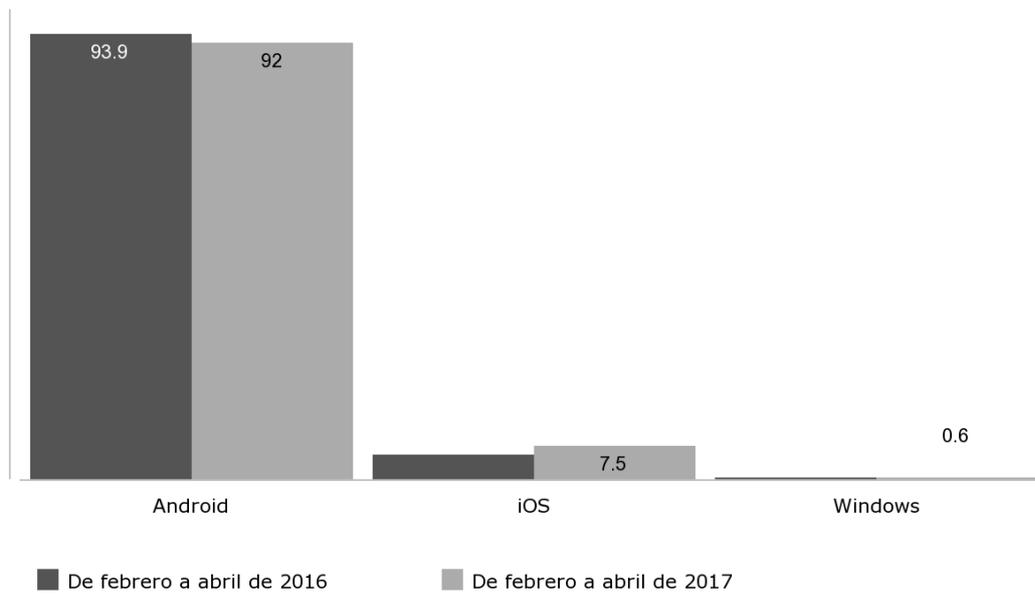
### 4.3.2. Frameworks

#### Android

La razón por la cual se escoge este framework en lugar de iOS o WindowsPhone es meramente práctica, en el sentido en que esta plataforma es la más extendida entre los Smartphones alcanzando un 92 % de la cuota de

mercado en abril de 2017 (figura 4.2).

Cuota de mercado de sistemas operativos en España (%)



**Figura 4.2:** Cuota de mercado de sistemas operativos para dispositivos móviles en los últimos años.

### 4.3.3. Lenguajes de programación

#### JAVA

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. En la actualidad es el lenguaje escogido por Google para el desarrollo nativo de aplicaciones Android.

#### SQLite

SQLite es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público creado por D. Richard Hipp. Implementa un sistema de gestión de bases de datos transaccionales Structured Query Language (SQLite) auto-contenido, sin servidor y sin configuración.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB.

El autor de SQLite ofrece formación, contratos de soporte técnico y características adicionales como compresión y cifrado.



# ANÁLISIS

---

## 5.1. Introducción

En este capítulo se presenta en detalle la fase de análisis del proyecto [10, Cap. 4, Cap. 5]. Para ello se exponen los casos de uso y el catálogo de requisitos. El catálogo de requisitos se elabora como una guía a seguir para completar la funcionalidad que se espera del proyecto y las pruebas se realizan en base a ello. Esto convierte a la fase de análisis en una de las más importantes y críticas del proceso de desarrollo software. En este punto ha sido fundamental el trabajo de detección de las necesidades de los clientes. Para ello se ha planteado a 14 usuarios la motivación de este proyecto y se les ha preguntado sobre lo que esperarían del mismo. Los resultados obtenidos conforman una sólida base de los requisitos que se exponen en este apartado. Se podría decir que es en este capítulo donde asentamos las bases del proyecto y, a partir de ellas, el proyecto va tomando forma.

Para empezar se ha hecho una definición de las acciones que podrá realizar cada tipo de usuario en la aplicación. Esto puede verse gracias a los casos de uso del siguiente apartado. Una vez se hayan definido las acciones y, basándonos en ellas, se detallarán los requisitos específicos de la aplicación.

## 5.2. Roles de usuario

**Cliente.** Los usuarios de este tipo utilizarán el aplicativo **The Waiter Plus** en sus dispositivos móviles o bien **The Waiter Wear** en sus smartwatches. La app podrá ser descargada de manera gratuita desde Google Play Store en sus terminales; así mismo podrá ser utilizada en cualquier establecimiento que haga uso del sistema **The Waiter**. La autenticación para estos usuarios no es necesaria pero pueden elegir identificarse, lo que sencillamente sirve para aumentar la información de su perfil. Para tener acceso a todas las funcionalidades será necesario que los clientes se conecten con éxito a la red local del establecimiento y lean un QR correcto, generado por el administrador del negocio.

**Camarero.** Los camareros son el único tipo de empleado que reciben funcionalidad en esta versión de la aplicación. Utilizarán el aplicativo **The Waiter Plus** que, al igual que los clientes, lo podrán descargar de forma gratuita desde Google Play Store. La autenticación para estos usuarios es obligatoria, las credenciales deberán ser proporcionadas por el administrador.

Los camareros, al igual que los clientes, para tener acceso a sus funcionalidades deberán escanear un código QR válido generado por el administrador.

**Administrador.** El administrador utiliza un aplicativo diferente al resto de roles, **The Waiter TPV**, que de nuevo podrá descargarlo de forma gratuita desde Google Play Store, pero en este caso

se limitará el aplicativo bajo diferentes criterios que se detallarán más adelante, impidiendo que pueda usarse completamente de forma gratuita. Este aplicativo está diseñado para tabletas, ya que se requiere un dispositivo de 7 pulgadas de pantalla como mínimo. A diferencia del resto de aplicativos, no requiere la lectura de códigos QR ni estar conectado a la red para funcionar correctamente, sin embargo este elemento juega un papel fundamental en el sistema, pues en el se alojará la base de datos de la solución, por lo que se deberá mantener conectado a la red del local para que los demás aplicativos puedan realizar las consultas oportunas, tales como la autenticación, el listado de la carta, etc.

El detalle de los tipos de usuarios así como las acciones asociadas y/o restringidas para cada rol se expone más adelante.

### 5.3. Casos de uso

Los casos de uso definidos para el proyecto se presentan en 3 diagramas según la siguiente clasificación:

#### Diagrama de caso de uso para los clientes.

La conexión con la base de datos y la lectura del código QR son requisitos para acceder a la mayoría de las funcionalidades de la aplicación. Cumplir uno de estos requisitos sin el otro no aumenta ni disminuye el número de acciones para un usuario, ambos deben darse juntos.

También influye la conexión del dispositivo a la red local del establecimiento pero esto no le impide al usuario llevar a cabo determinadas acciones, sólo implica que estas no tendrán el resultado esperado, es decir, podrá usar el chat pero no recibirá mensajes ni podrá enviarlos, también podrá elaborar solicitudes para el camarero pero éstas no serán enviadas. Estas acciones “inútiles” se incluyen en el diagrama (figura 5.1) junto con aquellas que no pueden llevarse a cabo sin realizar la lectura de un código QR correcto, y para poder distinguir las aparecen con el fondo blanco, a diferencia de aquellas acciones que pueden realizarse offline y sin la información de un código QR, que aparecen con el fondo gris.

#### Diagrama de caso de uso para los camareros.

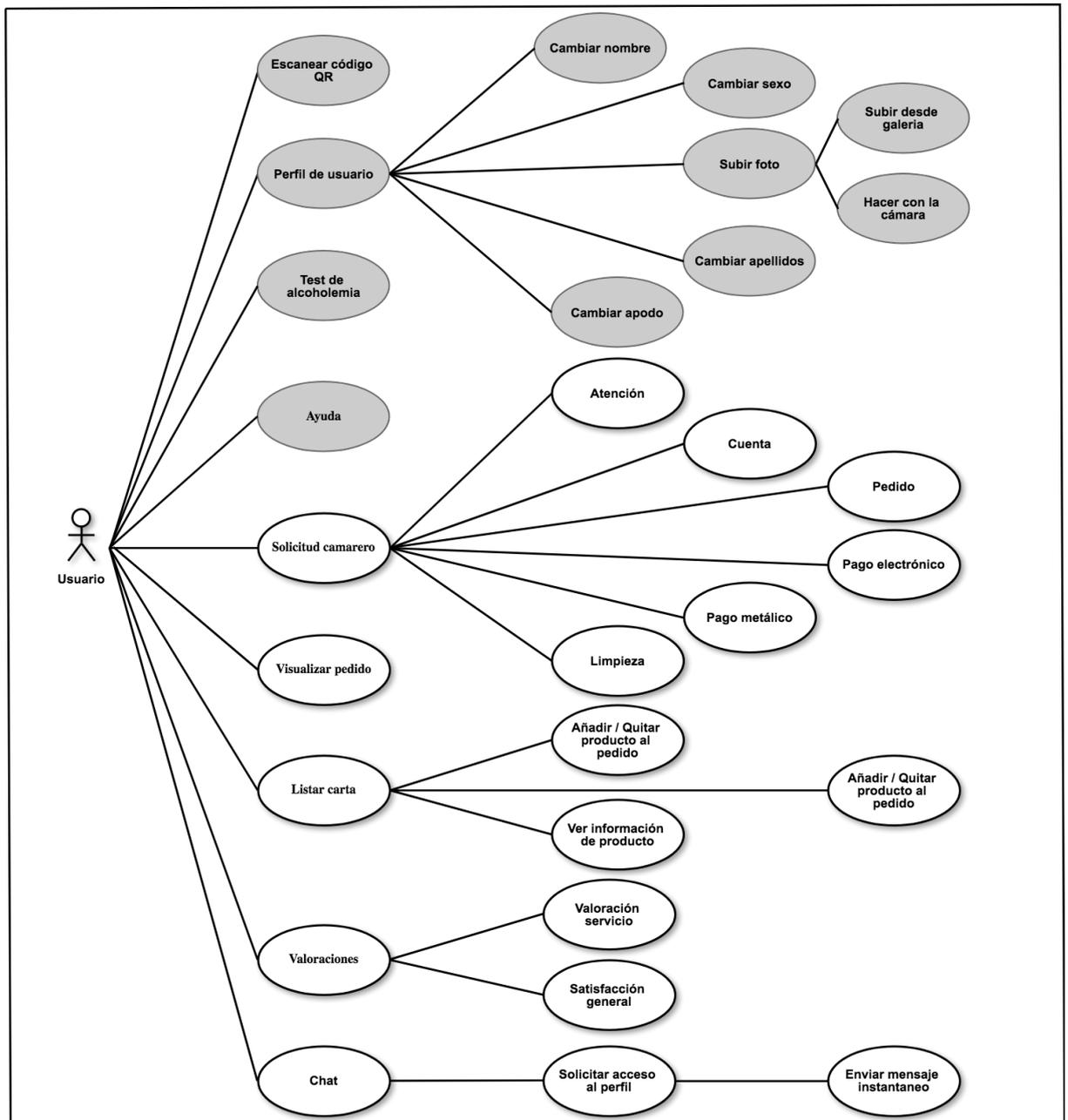
En este apartado se incluye el diagrama de comportamiento de los camareros de un establecimiento. Pese a que la herramienta utilizada por los camareros es la misma que para los clientes, se pueden diferenciar dos flujos de comportamiento completamente distintos y por eso se ha dado lugar a este otro diagrama (figura 5.2).

#### Diagrama de caso de uso para los administradores.

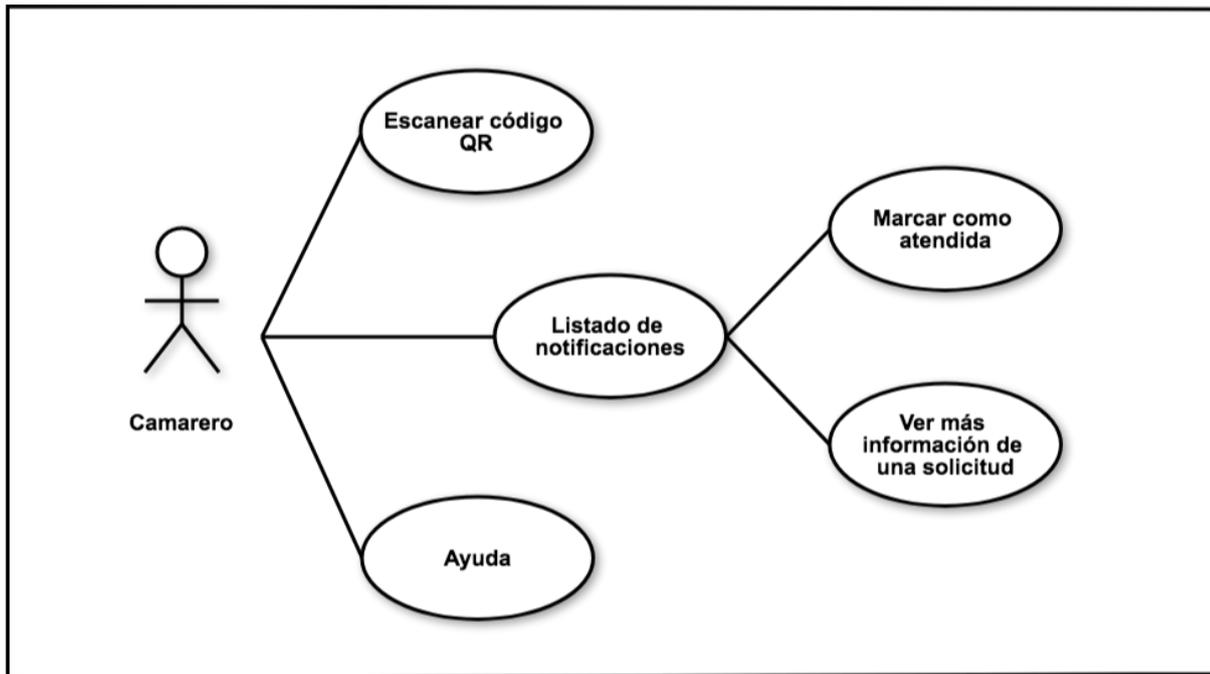
Los administradores tienen acceso a cualquier herramienta del sistema, ostentando el rol correspondiente. Pese a esto, y por simplicidad, en el diagrama de este apartado el diagrama (figura 5.3) se detallará únicamente el comportamiento restringido para este rol.

### 5.4. Catálogo y definición de requisitos

Los requisitos definidos para esta aplicación se pueden clasificar en dos grupos:



**Figura 5.1:** Diagrama de caso de uso para los clientes.



**Figura 5.2:** Diagrama de caso de uso para los camareros.

**Requisitos funcionales:** especifican las funcionalidades propias del sistema.

**Requisitos no funcionales:** en este grupo se clasifican aquellos objetivos que no describen funcionalidades de la aplicación. Se tienen en cuenta por ejemplo, aspectos de diseño, usabilidad, rendimiento, etc.

A continuación se define el catálogo de requisitos para detallar la funcionalidad de la aplicación y sus características.

### 5.4.1. Requisitos funcionales

Puesto que el sistema se compone de 3 aplicativos diferentes, los requisitos citados a continuación se agrupan para facilitar su entendimiento.

#### Requisitos comunes

**RF-1.– Compatibilidad.** La aplicación será compatible con las APIs de Android más distribuidas en el mercado en los últimos años; para ser más exactos es compatible desde la 19 hasta la 26, que es la última conocida a fecha de impresión de este documento.

**RF-2.– Conexión a la red local.**

**RF-2.1.– Cliente.** Los clientes necesitarán una red local en el establecimiento para comunicarse con los camareros u otros clientes. Además, esta conexión será necesaria si el cliente decide autenticarse con una cuenta de usuario de un servicio como Facebook [21].

**RF-2.2.– Camarero.** Los camareros necesitan acceso a la red para poder recibir las peticiones de los clientes, así como para su correcta autenticación.

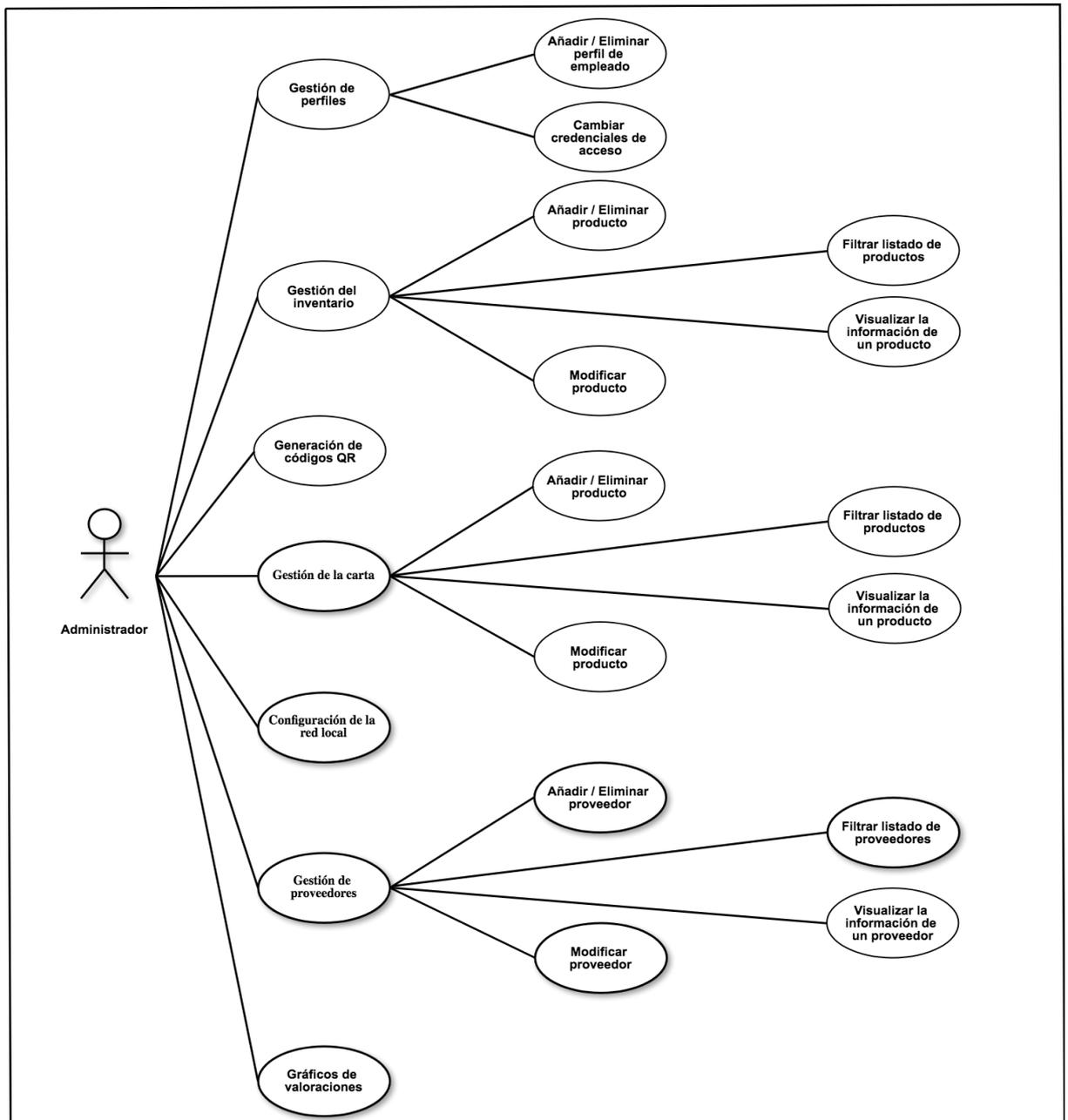


Figura 5.3: Diagrama de caso de uso para los administradores.

**RF-2.3.– Administrador.** La aplicación con la que trabaja el administrador del establecimiento será la que aloje la base de datos, por lo que es prioritaria una conexión continua a la red del establecimiento para recibir las peticiones a la base de datos y responderlas si fuera necesario.

### **RF-3.– Autenticación.**

**RF-3.1.– Cliente.** El cliente no se autenticará de forma obligatoria. Podrá elegir, por ejemplo, iniciar sesión con su cuenta de Google o Facebook. Esta opción simplemente la dará la posibilidad de tener un perfil más completo.

**RF-3.2.– Camarero.** El acceso de los camareros se realizará desde la misma aplicación utilizada por los clientes, por ello es necesaria la autenticación con las credenciales proporcionadas por el administrador, así se podrá distinguir entre clientes y empleados.

**RF-3.3.– Administrador.** El administrador deberá identificarse en la aplicación **The Waiter TPV**, únicamente a través de un código o contraseña, puesto que el usuario de este tipo es único para esta aplicación.

### **Requisitos The Waiter Plus**

**RF-4.– Gestión de roles.** Debido a que este es el único aplicativo que podrá ser utilizado por dos perfiles diferentes, se utilizará la autenticación citada anteriormente para darle acceso al usuario a las funcionalidades correspondientes.

**RF-5.– Configuración del perfil de usuario.** El usuario podrá configurar los datos que serán utilizados para la comunicación interna entre clientes del establecimiento. Estos datos estarán almacenados en las preferencias [22] del teléfono, lo que permite que se mantenga la información al salir de la aplicación e incluso al apagar el dispositivo.

- **Nombre.**
- **Apellidos.**
- **Apodo.**
- **Foto.**
  - **Seleccionar de la galería.**
  - **Hacer una foto.**
- **Sexo.**

**RF-6.– Carrito.** El carrito es la forma en la que denominaremos la acción de interactuar con los productos de la carta para modificar el pedido. Los productos de la carta se listan clasificados por categorías, y sobre este listado se pueden aplicar determinados filtros de búsqueda. Cada uno de los elementos que componen estos listados tendrá una pareja de botones y un contador, estos botones permitirán “añadir productos al carrito.” “eliminar productos del carrito”, en el contador se mostrará el número de unidades de un producto que están añadidas al pedido.

**RF-7.– Pedido.** Los elementos se añaden al pedido desde la funcionalidad de carrito y, una vez verificado por el usuario, podrá enviárselo al camarero. En el pedido también tiene opciones de gestión que le permiten añadir o quitar productos.

**RF-8.– Pedido personalizado.** El pedido personalizado tiene como objetivo que un cliente pueda solicitar modificaciones sobre los productos existentes o bien realizar un pedido de algún producto que no se encuentre en la carta. Las modificaciones que se pueden realizar no tienen restricciones, es decir, pueden variar desde detalles en la condimentación de la comida hasta variaciones de los elementos de un plato o un menú. El texto de esta solicitud es completamente libre.

**RF-9.– Listado de notificaciones.** Entre las funcionalidades de los camareros se incluye el acceso a un listado en el que se irán añadiendo las solicitudes de los clientes en orden de llegada. Se ha impedido la repetición de solicitudes a través de la gestión de prioridades. Una solicitud repetida aumenta la prioridad de la solicitud que ya se encuentra en el dispositivo del camarero, hasta un límite de 4 solicitudes repetidas.

**RF-9.1.– Atención de una solicitud.** El camarero podrá pulsar sobre una solicitud para marcarla como atendida y eliminarla del listado. Por el riesgo que conlleva este movimiento, se mostrará un diálogo de confirmación antes de eliminar la solicitud.

**RF-9.2.– Desglose de una solicitud.** El camarero podrá mantener pulsada una de las peticiones para tener acceso a la información adicional que dicha petición pudiera contener, como ocurre en el caso de los pedidos: el mensaje que se encola en el listado de notificaciones es "Recepción de pedido - Mesa X".

**RF-10.– Gestión de la prioridad de las peticiones.** Una solicitud realizada por un usuario aparece en la lista de notificaciones únicamente una vez, independientemente del número de veces que se realice. Para que esto sea posible, cada petición, tiene asociada una prioridad, cuando la petición llega al listado de un camarero para ser mostrada, si su prioridad es 0, se añadirá directamente a dicho listado, en caso contrario, aumentará la prioridad de la petición que ya exista y sea idéntica a esta nueva que ha sido generada.

**RF-11.– Prioridad.** Hay 4 niveles de prioridad que aumentarán a medida que se reciban peticiones de un mismo tipo y origen. El camarero será responsable de evaluar estas prioridades como considere oportuno.

**RF-12.– Comunicación interna.**

**RF-12.1.– Chat general del local.** En este chat pueden comunicarse todas las personas que estén conectadas a la misma red local de un establecimiento y tengan esta funcionalidad abierta en la aplicación. Desde aquí podrán solicitar una información más detallada del perfil de cualquiera de los participantes y así podrán establecer una comunicación privada.

**RF-12.2.– Mensajería instantánea privada.** Cuando un cliente solicita información detallada del perfil de otro y obtiene respuesta puede iniciar un hilo de mensajería instantánea a través de eventos y diálogos de la aplicación. Este hilo solo permanecerá funcional mientras un cliente reciba un mensaje y lo conteste. Si al recibir un mensaje utiliza la opción cerrar, el hilo de comunicación se cerrará a la espera de que se inicie de nuevo con una solicitud de información ampliada del perfil.

Este concepto puede resultar enrevesado pero se explica en detalle en posteriores apartados.

**RF-13.– Envío de valoraciones.** El cliente podrá enviar valoraciones sobre la atención y el servicio recibido durante su estancia en el establecimiento.

**RF-14.– Envío de sugerencias.** El cliente podrá enviar sugerencias que ayuden a mejorar los servicios que ofrece un establecimiento.

**RF-15.– Solicitud de hoja de reclamaciones.** El usuario podrá solicitar la hoja de reclamaciones del establecimiento.

**RF-16.– Test de alcoholemia.** El test de alcoholemia permitirá que los clientes estimen la tasa de alcohol ingerido para garantizar su seguridad si se disponen a conducir al abandonar el establecimiento.

### Requisitos *The Waiter Plus* y *The Waiter WEAR*

**RF-17.– Peticiones al camarero.** El cliente podrá realizar diferentes peticiones al camarero de entre las siguientes:

- **Atención.**
- **Limpieza.**
- **Solicitud personalizada.**
- **Cuenta.**
- **Pago en metálico.**
- **Pago electrónico.**

### Requisitos *The Waiter TPV*

**RF-18.– Gestión de empleados.** El administrador podrá crear los perfiles de empleado que considere necesarios, asignándoles un nombre de usuario y una contraseña para que éstos tengan acceso a las funciones correspondientes desde la aplicación *The Waiter Plus*. Los campos configurables de un empleado son:

- **Nombre.**
- **Puesto.**
- **Nombre de usuario.**
- **Contraseña.**

**RF-19.– Gestión de proveedores.** El administrador tendrá acceso a una pantalla con dos vistas: por un lado podrá visualizar el listado de proveedores, y por otro lado, podrá visualizar toda la información del proveedor seleccionado en la lista. Además, esta pantalla de información será completamente editable, para que el usuario pueda configurar los parámetros de un proveedor.

**RF-19.1.– Filtros de búsqueda.** El administrador podrá ordenar el listado de proveedores en orden alfabético, ascendente y descendente. También podrá utilizar la búsqueda por coincidencias, escribiendo el texto o parte de lo que desee buscar.

**RF-19.2.– Configuración del proveedor.** Los productos podrán ser añadidos, eliminados o modificados. Los campos de información que contienen los proveedores son:

- **Nombre.**
- **Dirección 1.**
- **Dirección 2.**

- **Teléfono 1.**
- **Teléfono 2.**
- **Correo electrónico.**
- **Página web.**

**RF-20.– Generación de códigos QR .** Los códigos QR se utilizan para que los clientes e incluso los camareros puedan conocer datos del local que son necesarios para el correcto funcionamiento de la aplicación. El encargado de generarlos es el administrador.

Los camareros y los clientes necesitaban prácticamente la misma información en el código pero se ha introducido una variable que permite distinguirlos.

- **Camarero.** El número de mesa es -1.
- **Cliente.** El número de mesa es asignado por el administrador al generar el QR . El objetivo es que cada mesa cuente con un QR diferente para poder localizar las peticiones, puesto que el cliente almacenará esta información sobre la mesa al leer el código QR , y la enviará como cabecera de todas sus solicitudes.

**RF-21.– Configuración de la red.** El administrador podrá configurar los parámetros necesarios para la conexión a la red de los empleados y los clientes.

**RF-22.– Configuración del inventario.** El administrador tendrá acceso a una pantalla con dos vistas. Por un lado podrá visualizar el listado de los productos del inventario, y por otro lado, podrá visualizar toda la información del producto seleccionado en la lista. Además, esta pantalla de información será completamente editable, para que el usuario pueda configurar los parámetros de un producto.

**RF-22.1.– Filtros de búsqueda.** El administrador podrá ordenar el listado de productos en orden alfabético, ascendente y descendente. También podrá utilizar la búsqueda por coincidencias, escribiendo el texto o parte de lo que desee buscar.

**RF-22.2.– Configuración del producto.** Los productos podrán ser añadidos, eliminados o modificados. Los campos de información que contienen los productos del inventario son:

- **Foto.**
- **Nombre.**
- **Unidad de medida.**
- **Proveedor.**
- **Cantidad.**
- **Información sanitaria.**
- **Más información.**
- **Observaciones.**
- **Precio base.**

**RF-23.– Configuración de la carta.** El administrador tendrá acceso a una pantalla con dos vistas. Por un lado podrá visualizar el listado de los productos de la carta, y por otro lado, podrá visualizar toda la información del producto seleccionado en la lista. Además, esta pantalla de información será completamente editable, para que el usuario pueda configurar los parámetros de un producto.

**RF-23.1.– Filtros de búsqueda.** El administrador podrá ordenar el listado de platos en orden alfabético, ascendente y descendente. También podrá utilizar la búsqueda por coincidencias, escribiendo el texto o parte de lo que desee buscar.

**RF-23.2.– Configuración del producto.** Los productos podrán ser añadidos, eliminados o modificados. Los campos de información que contienen los productos de la carta son:

- **Foto.**
- **Nombre.**
- **Listado de productos.** Aquí se listan los elementos del inventario que contiene un producto de la carta, pero también puede existir un producto de la carta que contenga otros productos de la carta, como sucedería en el caso de los menús.

**RF-24.– Gráficos de valoraciones.** El administrador tendrá acceso a un panel con diferentes gráficos que le permitirán conocer las valoraciones que los clientes han hecho acerca del establecimiento, el servicio y los productos.

## 5.4.2. Requisitos no funcionales

**RNF-1.– Interfaz de usuario intuitiva.** El número de acciones se minimiza en cada pantalla para conseguir que el usuario encuentre la información que busca fácilmente.

**RNF-1.1.– Menús de opciones.** Puesto que no se aportan manuales con la aplicación se ha perseguido que los menús fuesen lo suficientemente intuitivos como para que los clientes entiendan en todo momento las acciones que pueden realizar a través de la misma.

**RNF-2.– Privacidad.** Los mensajes que son lanzados a la red no son almacenados en ninguna estructura. Tampoco pasan a través de un servidor, los usuarios funcionan como clientes y servidores simultáneamente.

# DISEÑO E IMPLEMENTACIÓN

---

## 6.1. Introducción

En esta sección se detallan los aspectos más importantes de las fases del desarrollo, diseño e implementación [10, Cap. 6]. Para aportar el mayor detalle posible sobre el diseño, se han definido la arquitectura general del sistema con sus correspondientes elementos, la utilización de códigos QR, la interfaz de usuario y la estructura completa del proyecto incluyendo el diagrama de navegación de las aplicaciones.

Sobre estos diseños se detallará la codificación utilizada y algunas de las decisiones más importantes tomadas sobre la implementación. El objetivo de esta codificación es hacer funcional el diseño especificado.

## 6.2. Arquitectura de la aplicación

Dados los requisitos del apartado anterior, se ha propuesto una arquitectura para el sistema que permite darles solución.

De este modo **The Waiter** se compone de varios elementos, que combinados, aportan la funcionalidad planteada. La solución está basada en una arquitectura peer-to-peer (P2P) que permite que no existan clientes o servidores fijos, sino que todos los usuarios conectados se comporten como iguales entre sí, adquiriendo las funciones de un cliente o un servidor cuando sea necesario e intercambiando continuamente estos roles.

Con esto se definen los elementos principales en los que se basa la arquitectura de la aplicación (figura 6.1):

- **Base de datos.** Para la base de datos se utiliza un modelo relacional. Está diseñada para almacenar los datos de los negocios y ponerlos a disposición de los clientes. Esto permite que los usuarios tengan acceso, por ejemplo, al listado de productos y precios del establecimiento. La base de datos es almacenada en el aplicativo **The Waiter TPV**, y a través de la red y el protocolo de comunicación, el resto de aplicativos realizan las consultas necesarias.
- **Red.** Para el correcto funcionamiento de la aplicación, los establecimientos deberán poseer dispositivos que permitan crear una red local.
- **Usuarios.** El dispositivo de cada usuario cuando se instala la aplicación se convierte en un nodo de una red multicast sobre User Datagram Protocol (UDP). Los usuarios pueden utilizar 3 aplicativos y ostentar 4 roles diferentes, a saber:
  - **The Waiter TPV.** Este aplicativo es utilizado únicamente por los usuarios del tipo administrador. En él, como se ha mencionado, se almacena la base de datos y mantiene una conexión constante a la red para que el resto de componentes tengan acceso a

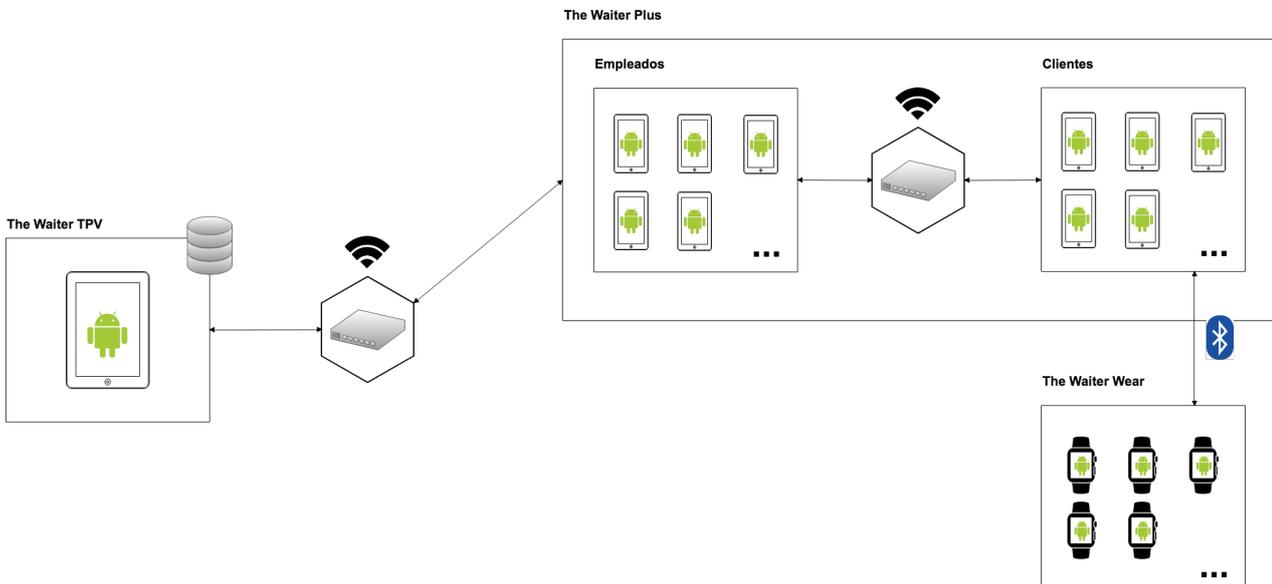
dicha base de datos, bien para consultar información o almacenarla.

- **The Waiter Plus.**

- ◇ **Empleados.** Los empleados sólo contactan con el TPV a través de la red en el momento de la autenticación. Su conexión debe ser continua, pues a través de ella reciben las peticiones enviadas por los clientes.

- ◇ **Clientes.** Los clientes consultan continuamente información de la base de datos a través de la red y, de la misma forma, envían información para que se grabe, como sucede en el caso de las valoraciones. Además, utilizan la red constantemente para enviar las solicitudes a los camareros.

- **The Waiter Wear.** La aplicación para smartwatches se utiliza únicamente por los clientes y hace uso del dispositivo móvil al que esté conectado para el envío de peticiones a los camareros a través de su conexión a la red. Es el único aplicativo del sistema que no hace uso de la base de datos.



**Figura 6.1:** Diagrama general de la arquitectura del sistema.

Para la comunicación entre los usuarios de la red se ha implementado un protocolo que se detallará en posteriores apartados. Como ya se ha mencionado, para permitir dicha comunicación es necesario que los usuarios estén conectados a una misma red local, es decir, la aplicación está preparada para ser utilizada en multitud de establecimientos con sus correspondientes redes locales pero un usuario sólo podrá comunicarse con los usuarios de la red que pertenezca al establecimiento en el que se encuentre.

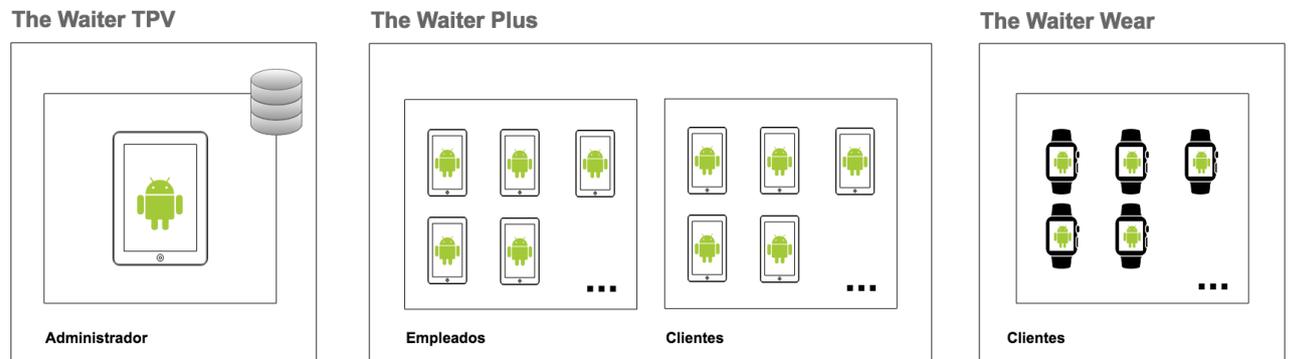
La base de datos se encuentra situada en el aplicativo principal del sistema, **The Waiter TPV**, por lo que siempre que el dispositivo se mantenga encendido, todos los usuarios conectados a la red local podrán acceder a ella.

### 6.2.1. Usuarios

No es necesario mencionar la importancia que tienen los usuarios para este proyecto. Los roles que pueden ostentar estos usuarios se han detallado en el apartado 5.2, y la arquitectura de los diferentes aplicativos, así

como las funciones que pueden desempeñar en ellos cada tipo de usuario, se explican a lo largo de este capítulo.

Para aclarar este concepto, se introduce en esta sección un diagrama (figura 6.2) que relaciona los aplicativos del sistema con los tipos de usuario para los que están diseñados.



**Figura 6.2:** Diagrama de los usuarios del sistema y sus correspondientes aplicativos.

## 6.2.2. Protocolo de comunicación

Pese a los múltiples protocolos de comunicación que existen en la actualidad, se ha decidido implementar uno propio. Esto se debe principalmente a que dichos protocolos son muy completos, y en este caso, se necesitaba una funcionalidad tan reducida que resultó más fácil hacer un diseño que se adaptase a las necesidades del sistema objeto de este trabajo.

El protocolo define un formato de mensajes que permite establecer la comunicación entre dos puntos. Para que exista comunicación ambos extremos deben implementar tareas de envío y recepción para lanzar mensajes a la red y capturarlos.

El protocolo se ha implementado sobre multicast para evitar la dependencia de servidores externos (que almacenen los mensajes), por lo que el receptor además deberá implementar un filtro para saber qué mensajes están dirigidos a su dispositivo.

### Implementación de las estructuras de envío y recepción

En la versión anterior del sistema, la recepción era un proceso que se encapsulaba en una tarea asíncrona de la aplicación [23]. Esto provocaba que la recepción de mensajes sólo fuera posible cuando la aplicación estaba corriendo.

En esta versión se ha introducido un elemento en el sistema que aloja la base de datos, por lo que resultó necesario garantizar que el protocolo funcionase pese a que el aplicativo no estuviera corriendo en el terminal. Pese a esto, la implementación no es eficiente al 100 %, pues si el terminal está apagado, se perderá completamente el acceso a la base de datos, pero de igual modo nos pasaría si apagásemos un servidor externo que contuviera dicha base de datos.

En un principio, esta mejora sólo se incorporaría en **The Waiter TPV**, pero observamos que también sería útil introducirla en **The Waiter Plus**, lo que permitía también que los usuarios pudieran recibir la totalidad de los mensajes cuando el dispositivo estuviera conectado a la red.

En el caso del aplicativo **The Waiter Wear**, no se reciben mensajes de la red, únicamente se envían.

- **Envío.**

El proceso de envío es el mismo en todos los casos, el terminal lanza a la red una cadena de texto, y en este punto entra en juego el protocolo, puesto que la cadena de texto lleva asociado un formato para que así el receptor pueda entender el mensaje. Es por esto que cada aplicativo construye un mensaje siguiendo unas determinadas pautas, como se verá en el siguiente apartado.

Código disponible en el anexo F.

- **Recepción.**

El proceso de recepción se implementa mediante un servicio configurado para arrancar cuando el dispositivo se encienda. Para que esto suceda, uno de los aplicativos se debe haber ejecutado previamente en el terminal.

Código disponible en el anexo E.

Una vez recibido el mensaje, éste se parsea según las pautas indicadas en el protocolo, como se verá en el siguiente apartado. Los pasos que tienen lugar cuando se parsea el mensaje son los siguientes:

- 1.— Se separa el mensaje por el token "|". Se comprueba que el primer item resultante de la división sea el código de verificación. En caso contrario se descarta el mensaje.
- 2.— Se comprueba el segundo token resultante de la separación. En él se almacena la clave primaria, la cual nos indica a quién va dirigido el mensaje.
- 3.— Se comprueba el tercer token de la separación. En él se almacena la clave secundaria, la cual nos informa sobre la naturaleza del mensaje.
- 4.— Una vez realizado este filtro, no se puede generalizar la actuación del protocolo, puesto que cada mensaje tiene una información diferente. En este caso el protocolo actúa diferente según le indiquen las claves citadas anteriormente. El detalle de cada tipo de mensaje se explica en los posteriores apartados.

Entre el envío y la recepción tiene lugar la implementación del protocolo, como se ha dicho, para dar lugar a la comunicación.

## Estructura del protocolo

- **Código de verificación.** Todos los mensajes tienen un código de verificación que permite comprobar que dicho mensaje ha sido lanzado por el sistema. Al tratarse de un protocolo multicast, cualquier mensaje que se lance a la red del establecimiento será capturado por los aplicativos, por lo que este código es necesario para identificar aquellos mensajes que deben procesarse dentro de las aplicaciones y cuáles no.
- **Clave primaria.** La clave primaria es el segundo token resultante de separar la cadena por el carácter "|" e indica el destinatario del mensaje.
- **Clave secundaria.** La clave secundaria es el tercer token resultante de separar la cadena por el carácter "|" e indica la naturaleza del mensaje.
- **Cuerpo.** El cuerpo del mensaje es el cuarto token resultante de separar la cadena por el carácter "|". Está compuesto de otros dos parámetros separados por los caracteres "::::". Estos parámetros son interpretados por las funciones correspondientes para ejecutar las acciones oportunas. Algunos de estos parámetros son:

- El usuario que envía el mensaje.
- El usuario destinatario del mensaje.
- El mensaje.
- El número de la mesa en la que se encuentra el usuario que envía el mensaje.
- Una valoración.
- ...

No todos los mensajes tienen la misma configuración de parámetros en el cuerpo, cada mensaje según su función incorpora únicamente los necesarios.

## Definición del protocolo

En este apartado se define de forma más concreta la construcción de los mensajes que son lanzados a la red. Se han clasificado por la clave primaria, para posteriormente mostrar las claves secundarias asociadas y un ejemplo del cuerpo de un mensaje del tipo correspondiente.

### CLIENTE

- **GLOBALCHAT.** Los mensajes con esta clave están destinados al chat público en el que participan todos los clientes del establecimiento.

Puesto que la red es multicast, los mensajes enviados por un usuario también los recibe de vuelta. Es por esto que todos los mensajes del chat se muestran en el momento de la recepción.

```
COD.VERIF|CLIENTE|GLOBALCHAT|emisor::mensaje
```

- **CATEGORIAS.** Los mensajes con esta clave contienen el listado de categorías devuelto por la base de datos.

```
COD.VERIF|CLIENTE|CATEGORIAS|mesa::id_categoria-categoria:: ...
```

- **CARTA.** Esta clave se incorpora en los mensajes acompañada del listado de productos de la carta.

```
COD.VERIF|CLIENTE|CARTA|mesa::id_producto-producto:: ...
```

- **INFO\_PRODUCTO.** En estos mensajes se encapsula la información solicitada de un producto en concreto de la carta.

```
COD.VERIF|CLIENTE|INFO_PRODUCTO|mesa::id_producto-producto::detalles_producto
```

- **PERFIL.** Esta clave se incorpora en los mensajes entre usuarios del chat cuando uno desea conocer el perfil completo de otro.

```
COD.VERIF|CLIENTE|PERFIL|mesa::emisor::receptor::mensaje
```

- **RESPUESTA\_PERFIL.** Este mensaje se genera si un usuario acepta compartir su perfil con otro que así lo haya solicitado.

```
COD.VERIF|CLIENTE|RESPUESTA_PERFIL|emisor::receptor::informacion_perfil
```

- **PRIVADO.** Estos mensajes transportan mensajes privados entre un par de usuarios que se han puesto de acuerdo previamente con el envío de su perfil.

```
COD.VERIF|CLIENTE|PRIVADO|emisor::receptor::mensaje
```

### CAMARERO

- **AUTENTICACION.** En este mensaje se recibe la respuesta de autenticación de un empleado desde

la base de datos.

COD.VERIF|CAMARERO|AUTENTICACION|1 - En caso de autenticación correcta

COD.VERIF|CAMARERO|AUTENTICACION|0 - En caso de autenticación incorrecta

- **PETICION.** Los mensajes con esta clave son solicitudes de los clientes dirigidas al camarero por lo que, de ser correcto el cuerpo, se muestran automáticamente en el listado de notificaciones.

COD.VERIF|CAMARERO|PETICION|mesa::producto-cantidad:: ... ::pedido\_personalizado

## ADMIN

- **AUTENTICACION.** En este mensaje se reciben los parámetros de un intento de autenticación para corroborar dicha información contra la base de datos.

COD.VERIF|ADMIN|AUTENTICACION|usuario::contraseña

- **CATEGORIAS.** Con este mensaje los usuarios solicitan el listado de categorías a la base de datos.

COD.VERIF|ADMIN|CATEGORIAS|mesa

- **CARTA.** Con este mensaje los usuarios solicitan el listado de productos de la carta a la base de datos.

COD.VERIF|ADMIN|CARTA|mesa::categoria

- **INFO\_PRODUCTO.** Con este mensaje los usuarios solicitan la información de uno de los productos de la carta.

COD.VERIF|ADMIN|INFO\_PRODUCTO|mesa:producto

- **VAL\_CAMARERO.** Los mensajes con esta clave contienen una valoración realizada por un usuario sobre un camarero en concreto, y de esta forma puede ser insertada en la base de datos.

COD.VERIF|ADMIN|VAL\_CAMARERO|mesa::empleado::valoracion\_numerica::comentarios

- **VAL\_PLATO.** Los mensajes con esta clave contienen una valoración realizada por un usuario sobre un plato en concreto, y de esta forma puede ser insertada en la base de datos.

COD.VERIF|ADMIN|VAL\_CAMARERO|mesa::plato::valoracion\_numerica::comentarios

- **VAL\_LOCAL** Los mensajes con esta clave contienen una valoración realizada por un usuario sobre el establecimiento, y de esta forma puede ser insertada en la base de datos.

COD.VERIF|ADMIN|VAL\_CAMARERO|mesa::valoracion\_numerica::comentarios

El protocolo se ha definido en función de los mensajes recibidos, pero las pautas seguidas para el envío de los mensajes son las mismas. Un mensaje recibido en un punto ha sido enviado desde otro punto y viceversa.

### 6.2.3. Base de datos

La base de datos tiene un diseño complejo para capturar todos los requisitos del sistema. Está compuesta por 14 tablas y 5 relaciones.

#### Diagrama relacional

El diagrama Relacional se corresponde con la Figura 6.3 de este documento.

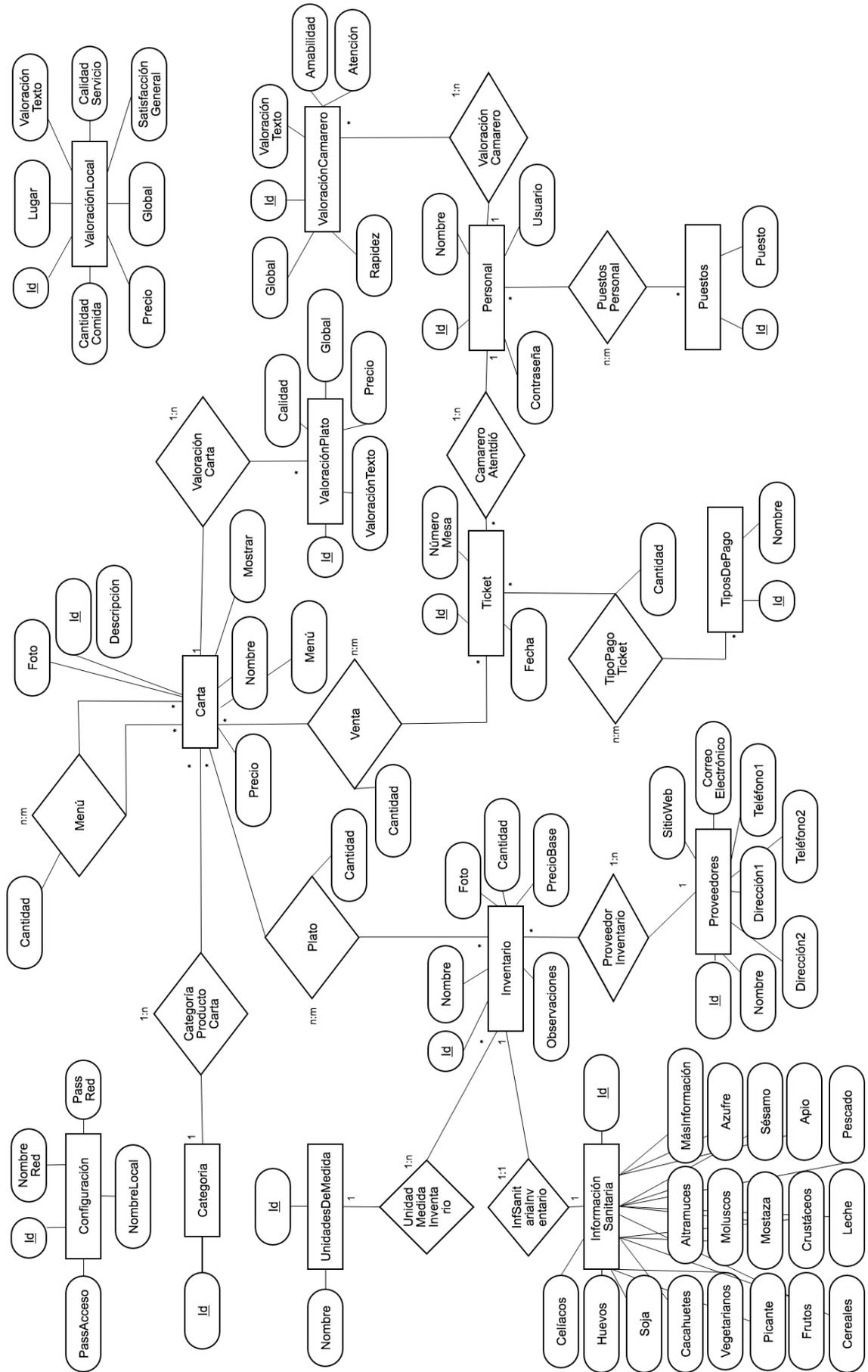


Figura 6.3: Diagrama Relacional de la base de datos.

## Diagrama entidad-relación

El diagrama Entidad-Relación se corresponde con la Figura 6.4 de este documento.

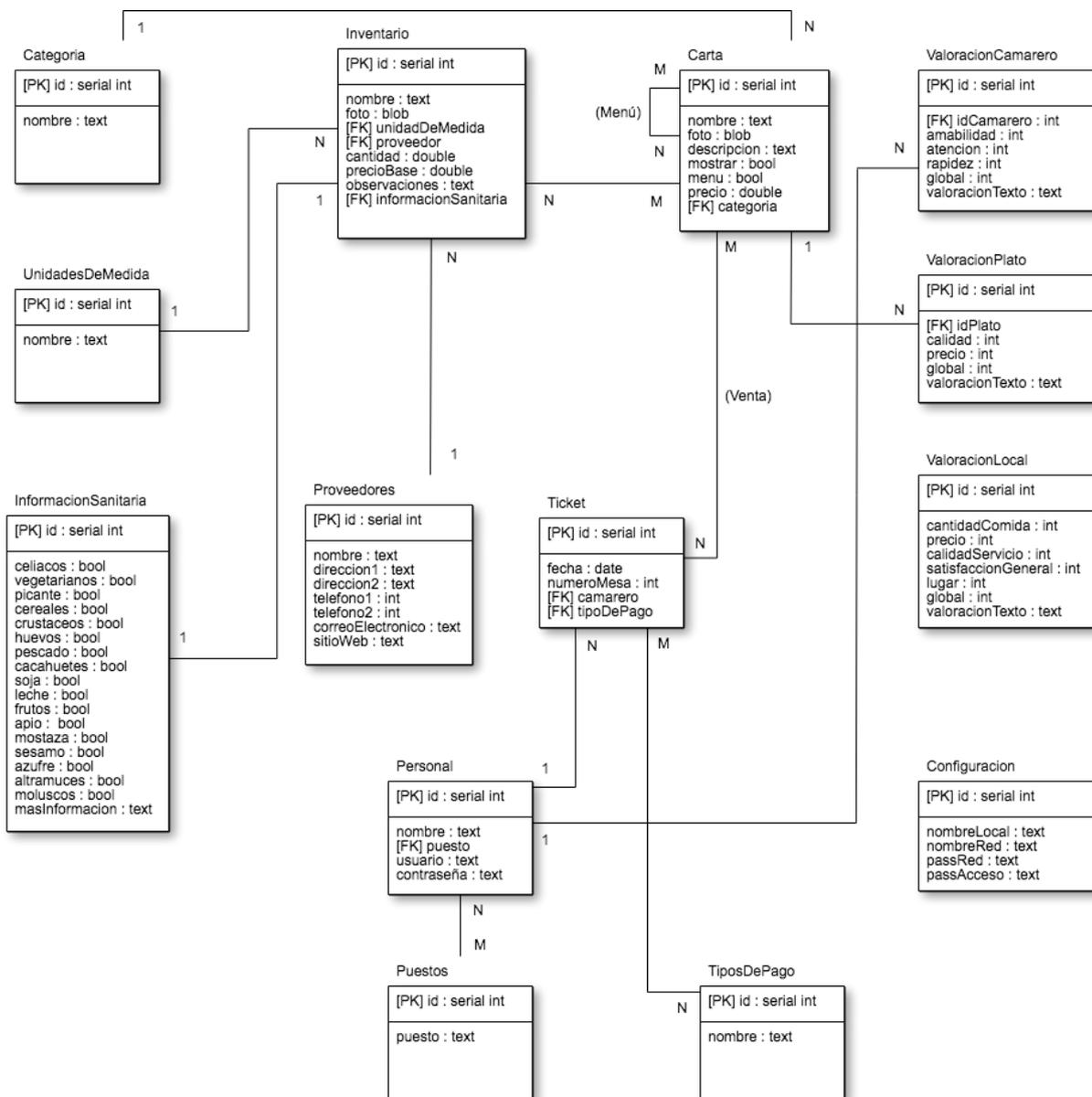


Figura 6.4: Diagrama Entidad-Relación de la base de datos.

## Conexión

En este proyecto la base de datos se almacena en el dispositivo elegido para instalar el aplicativo **The Waiter TPV**.

## 6.3. Códigos QR

Los códigos QR se han utilizado para que la aplicación tenga acceso a determinados datos. Estos códigos se generan desde la aplicación **The Waiter TPV** y sirven para que los empleados y los clientes conozcan datos sobre la red, el establecimiento en el que se encuentran o trabajan, etc. A continuación se explican más en detalle los datos que contiene cada código y con qué objetivo.

Para completar la funcionalidad entre la aplicación y los códigos han sido necesarios dos elementos:

- **Generador [24].** Este elemento utiliza una matriz de bits que se colorean en blanco o negro según lo indique una condición. Esta condición resulta de codificar los datos con una clase especial de Java. El código utilizado está disponible en el anexo D.

Aunque no es una funcionalidad prevista para este proyecto, en lugar de usar una herramienta existente se decidió implementar en parte por su simplicidad, como puede observarse en el código.

- **Lector [25].** Para el lector, se ha incorporado la librería nativa de Android “BarcodeDetector”. En la versión anterior del proyecto no existía esta funcionalidad de forma nativa y por ello se abría, desde la aplicación, otra aplicación con esta funcionalidad, sin embargo se planteó como trabajo futuro y al investigar y darnos cuenta de que existía una solución; se incorporó para esta nueva versión del proyecto.

La información que es almacenada en los códigos QR es la siguiente:

- **Clave de verificación.** Con esta clave se comprueba, al leer el código, que ha sido un código válido generado por la aplicación.
- **Nombre de la red local.**
- **Contraseña de la red local.**
- **Nombre del local.**
- **Número de mesa.** -1 al tratarse de un QR para los empleados del negocio.



(a) Empleado



(b) Cliente

**Figura 6.5:** Códigos QR generados por la aplicación con el código del anexo D

## 6.4. Interfaz de usuario

La interfaz de usuario diseñada no sigue ninguna plantilla ni se han utilizado herramientas para su elaboración. Es una interfaz sencilla, puesto que uno de los requisitos para este proyecto era proporcionar la mayor facilidad de uso a todos sus usuarios.

Siguiendo el estándar de estilo de Android, para mantener cierta coherencia entre todas las pantallas, se han utilizado estilos [26], declarados en un fichero llamado `styles.xml` en los diferentes proyectos Android. Estos estilos permiten agrupar elementos en torno a un mismo diseño, es decir, por ejemplo, permiten que todos los botones sean iguales. Estos estilos se han aplicado en la mayoría, sino en su totalidad, de los elementos de la solución aportando, como decíamos, coherencia y sencillez en todas las pantallas de la aplicación.

En anteriores apartados, y en concreto en el siguiente, se puede observar esta interfaz.

## 6.5. Estructura de las aplicaciones

En Android, cada una de las pantallas que componen una aplicación se denomina actividad. En nuestro caso, usamos una única actividad principal en cada aplicativo, sobre la que se muestran todas las pantallas del mismo. Estas pantallas que se muestran sobre una actividad principal se conocen como fragmentos, y la forma de mostrarse se llama anclaje, pues se muestran anclándose a la actividad principal.

Las actividades y los fragmentos son prácticamente el mismo concepto, al menos para el contexto en el que son utilizados en este proyecto. Están formados por dos partes: la parte lógica y la parte gráfica. La parte lógica es un archivo `.java`, que es la clase que se crea para poder manipular, interactuar y escribir el código de esa actividad. A pesar de utilizar un lenguaje orientado a objetos (Java en nuestro caso), el diagrama de clases no nos aporta información para entender el funcionamiento de la aplicación. En este sentido, el diagrama de actividades (fragmentos) resulta mucho más útil. También influye el hecho de que el diagrama de clases para este proyecto constaría de más de 40 clases y, más que ayudar a visualizar la estructura, dificultaría el entendimiento de la misma.

Teniendo en cuenta que la solución esta compuesta por 3 aplicativos diferentes, mostraremos el flujo de pantallas en 3 diagramas diferentes. Además, uno de los aplicativos puede ser utilizado por dos perfiles diferentes, y el diagrama de navegación de pantallas no tendría una parte común a ambos perfiles, por lo que en este caso se derivarán 2 diagramas diferentes, uno para cada rol, aunque se trate del mismo aplicativo.

### 6.5.1. The Waiter Plus

En la figura 6.6 se puede observar el diagrama de navegación entre pantallas del aplicativo **The Waiter Plus**. Por simplificar el diagrama y facilitar su lectura, no se incluyen todas las pantallas existentes, sólo aquellas más relevantes para entender el flujo de la aplicación. Sin embargo, se hace referencia a las pantallas que no están incluidas, y todas ellas están disponibles en el anexo H.

Cada pantalla del diagrama tiene un número para identificarla y poder describir brevemente su funcionamiento a continuación:

- 1.– **Inicio**. Pantalla de inicio de la aplicación.
- 2.– **Código QR**. En esta pantalla los clientes podrán escanear el código QR necesario para conectarse a la red del establecimiento y tener acceso en su totalidad a las funciones del aplicativo.

- 3.– **Peticiones camarero.** Desde esta pantalla el usuario puede solicitar la atención en mesa de un camarero y la limpieza de la misma.
- 4.– **Pedido y pago.** Desde esta pantalla el usuario tendrá acceso al listado de la carta para añadir productos a su pedido, y desde la pantalla que resumen dicho pedido podrá enviar la solicitud correspondiente al camarero. También, desde esta pantalla el usuario tendrá acceso las opciones de petición de cuenta y pago.
- 5.– **Chat.** Desde esta pantalla el usuario tendrá acceso al chat global del establecimiento.
- 6.– **Valoraciones.** En esta pantalla el usuario puede acceder al envío de valoraciones.
- 7.– **Test de alcoholemia.** El acceso al test de alcoholemia estará disponible desde esta pantalla.
- 8.– **Login empleado.** Desde la pantalla de inicio, el usuario puede elegir identificarse como un empleado desde el menú lateral, donde se le solicitarán sus credenciales para darle acceso a las funcionalidades de empleado.
- 9.– **Listado de notificaciones.** En esta pantalla el camarero podrá visualizar el listado de peticiones enviadas por los clientes.

### 6.5.2. The Waiter Wear

En la figura 6.7 se puede observar el diagrama de navegación entre pantallas del aplicativo **The Waiter Wear**. En este caso, por el pequeño tamaño de la aplicación, se han incluido la totalidad de las pantallas. En cualquier caso, las imágenes están disponibles en el anexo I, donde se pueden visualizar en mayor tamaño.

Cada pantalla del diagrama tiene un número para identificarla y poder describir brevemente su funcionamiento a continuación:

- 1.– **Inicio.** Pantalla de inicio de la aplicación.
- 2.– **Mesa.** En esta pantalla el usuario rellena su número de mesa para que el camarero sepa de dónde llegan las peticiones.
- 3.– **Peticiones.** En esta pantalla el usuario puede solicitar la atención en mesa de un camarero y la limpieza de la misma.
- 4.– **Pago.** Desde esta pantalla el usuario puede solicitar la cuenta y el pago de la misma.
- 5.– **Satisfacción.** En esta pantalla el usuario puede enviar su satisfacción general del servicio.
- 6.– **Valoración.** Desde esta pantalla el usuario podrá enviar su valoración personal sobre el establecimiento y su servicio.

### 6.5.3. The Waiter TPV

En la figura 6.8 se puede observar el diagrama de navegación entre pantallas del aplicativo **The Waiter TPV**. Por simplificar el diagrama y facilitar su lectura, no se incluyen todas las pantallas existentes, sólo aquellas más relevantes para entender el flujo de la aplicación, sin embargo se detallan las pantallas que no están incluidas, y todas ellas están disponibles en el anexo J.

Cada pantalla del diagrama tiene un número para identificarla y poder describir brevemente su funcionamiento a continuación:

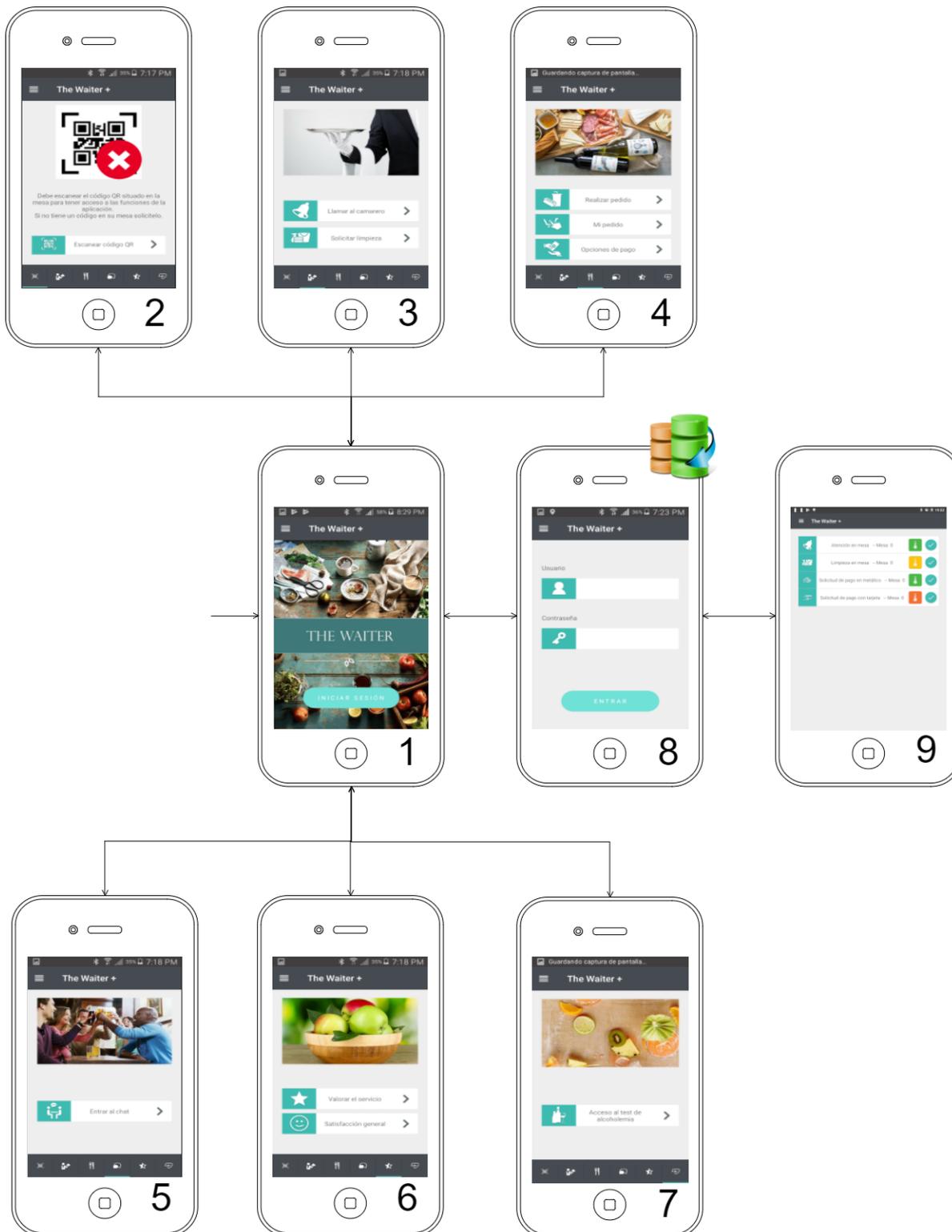
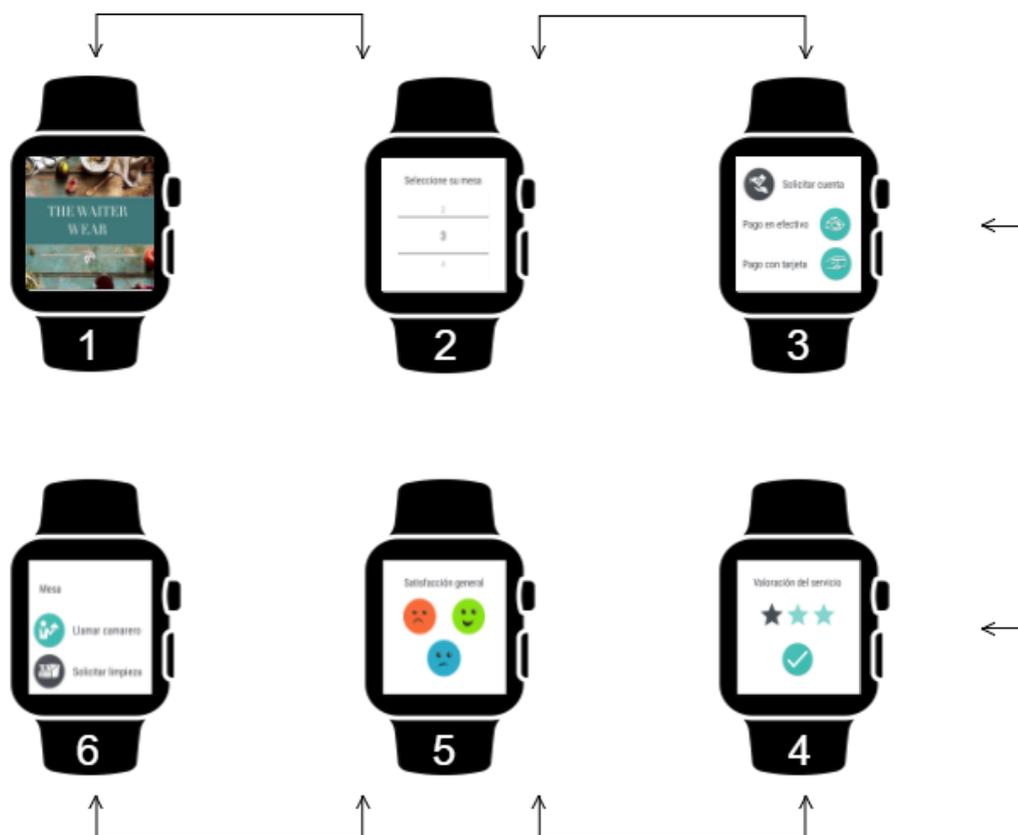


Figura 6.6: Diagrama de navegación entre pantallas *The Waiter Plus*.

4  
4

**Figura 6.7:** Diagrama de navegación entre pantallas *The Waiter Wear*.

- 1.– **Inicio.** Pantalla de inicio de la aplicación.
- 2.– **Administración.** Desde esta pantalla el usuario tendrá acceso a las preferencias, la gestión de usuarios y contraseñas y la generación de los códigos QR para los clientes y empleados.
- 3.– **Inventario.** En esta pantalla el administrador tendrá acceso a las pantallas de configuración de la carta, el inventario y los proveedores.
- 4.– **Estadísticas.** Desde esta pantalla el usuario tendrá acceso a diferentes gráficos que muestran estadísticas sobre las valoraciones de los clientes, el consumo de productos, los tiempos del servicio, etc.



**Figura 6.8:** Diagrama de navegación entre pantallas *The Waiter TPV*.

# PRUEBAS

---

## 7.1. Alcance de las pruebas

Las pruebas se han ido desarrollando a medida que se cerraban las tareas o se estimaba necesario tras las revisiones. El motivo por el que no se ha dedicado una fase aislada del desarrollo para las pruebas se basa en la escasez de recursos humanos. Así mismo, al no conocer previamente las herramientas fundamentales que se han utilizado en la implementación ha resultado complicado planificar unas pruebas desde el inicio.

En este apartado se listarán las pruebas que se han utilizado para validar y verificar el correcto funcionamiento de la aplicación, las cuales se han centrado en la comprobación de la funcionalidad, compatibilidad, accesibilidad y usabilidad.

### 7.1.1. Funcionalidad

Las pruebas sobre la funcionalidad tienen como objetivo la comprobación del correcto cumplimiento de los requisitos funcionales planteados para el proyecto.

Para ello se han utilizado como guía los requisitos de los tres aplicativos y se ha comprobado el correcto funcionamiento de cada uno individualmente y en contexto con otras funcionalidades relacionadas. Estas pruebas se han realizado en varias etapas del desarrollo, a medida que se implementaban nuevas funcionalidades o se integraban nuevos elementos. Los resultados obtenidos en las últimas pruebas garantizan el correcto funcionamiento de aproximadamente un 90% de los requisitos planteados. Esto se debe a que algunos de los requisitos no pueden ser validados completamente, como se verá a lo largo del capítulo.

Para el desarrollo de las pruebas han colaborado 33 usuarios con conocimientos técnicos y de la aplicación, reportando un total de 86 errores que han podido ser solventados. Algunos de estos errores fueron:

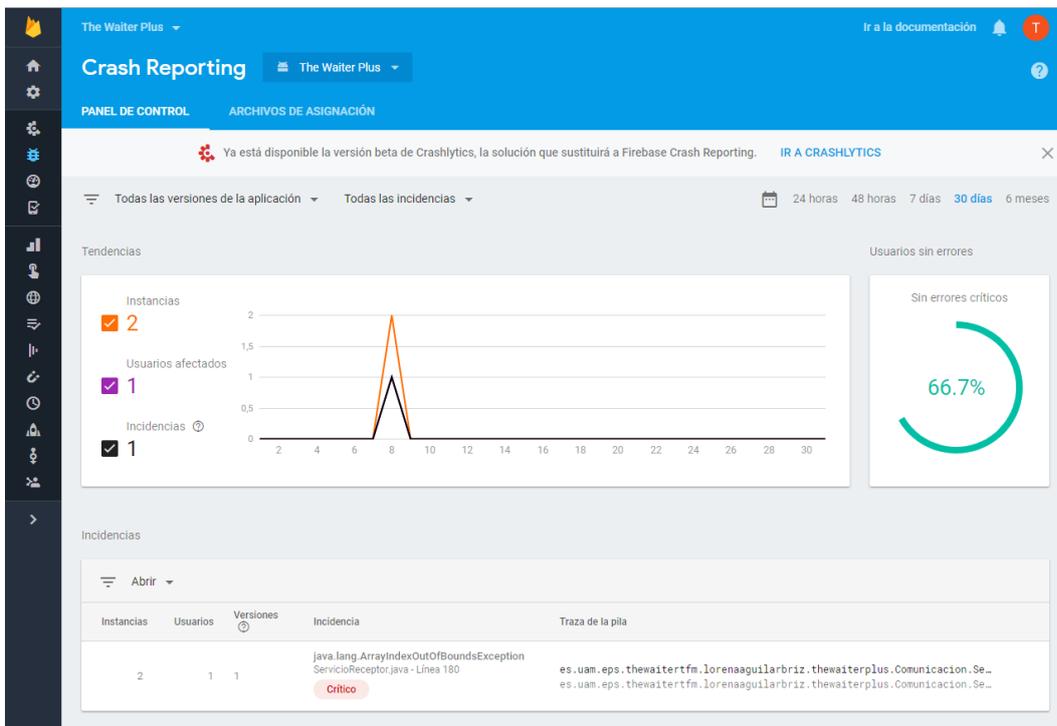
- Duplicidad en las notificaciones que reciben los camareros, lo que se ha controlado con la gestión de prioridades.
- Fallos de conexión en la red. Éste es uno de los errores que no ha podido solventarse, puesto que depende más de la propia red del establecimiento y sus buenas condiciones que del sistema en sí.
- Unicidad de los usuarios. Al no estar registrados en una base de datos es difícil distinguirlos y por ello se ha asociado un UUID a cada usuario.
- Introducción de datos erróneos en los formularios. Los usuarios han introducido todo tipo de valores inesperados en aquellos campos que les permitían hacerlo, lo que ha dado lugar a gran cantidad de errores. Por ello se han controlado en estos campos todos los rangos de valores válidos, avisando en caso de error en lugar de transmitirle a los aplicativos una información errónea.

- Acciones no disponibles sin el informe correspondiente para el usuario. Como se ha mencionado, el acceso a la funcionalidad completa de algunos de los aplicativos solo está disponible tras la conexión a la red de un establecimiento y la lectura de un código QR válido. En los casos en los que los usuarios no podían acceder a determinadas funciones, no eran correctamente informados del motivo.

Para estas pruebas se ha utilizado la funcionalidad de Firebase Crash Reporting (figuras 7.1 y 7.2), la cual ha permitido que los usuarios realizaran pruebas a la vez, o cuando les fuera posible, recibiendo un informe de todos los fallos que sufrían los aplicativos. También ha permitido que realicen las pruebas sin supervisión de los desarrolladores, que en otro caso deberían estar pendientes al momento en el que se produzca cualquier fallo.



**Figura 7.1:** En esta imagen se muestra el crash reporting de los últimos 6 meses.



**Figura 7.2:** En esta imagen se muestra el crash reporting de los últimos 30 días.

Los reportes recibidos tienen todo tipo de información, a parte del log, que es la información más útil para solventar el problema, otra información que ha resultado realmente útil es la versión de Android del dispositivo en el que se ha producido el fallo, como veremos en las pruebas de compatibilidad.

En el anexo G están disponibles algunos de estos informes.

## 7.1.2. Compatibilidad

La compatibilidad en Android se centra en las versiones. Es difícil asegurar la compatibilidad dado que, desde que comenzó el proyecto en 2015, se han publicado 3 versiones, con sus correspondientes parches y APIs. Con ello, se han quedado obsoletas muchas funciones de Android, y requiere un constante esfuerzo mantener la retrocompatibilidad adaptándose a las actualizaciones de las nuevas versiones.

Las aplicaciones han sido desarrolladas en base a la API 19 y desde ahí hasta la 26, la cual es la más reciente a fecha de impresión de este documento. Como se ha mencionado, el rango de versiones que se ha escogido es la opción que más compatibilidad proporcionaba.

También hay que tener en cuenta que desarrollando para la última versión que se conoce, se ha adquirido el riesgo de que muchos dispositivos, que no puedan ser actualizados, no tengan acceso a las aplicaciones, pero la decisión que se ha tomado aseguraba la mayor compatibilidad para el mayor número de usuarios posible.

Pese a esto, como se ha dicho, la compatibilidad es muy difícil de asegurar, y deja mucho trabajo a realizar en la fase de mantenimiento.

Las pruebas sobre la compatibilidad se han realizado utilizando 21 dispositivos de 7 versiones diferentes de Android. El resultado ha sido peor de lo esperado, pues los fallos de compatibilidad son frecuentes al compilar la aplicación para más de 5 versiones diferentes y, en especial, se han encontrado problemas con la última versión de Android, Oreo (versión 8.0, API 26).

## 7.1.3. Usabilidad

Se realizan pruebas sobre la usabilidad puesto que, a través de ellas, se obtiene una información directa e irremplazable de cómo los usuarios reales utilizan el sistema.

Estas pruebas se enfocan en medir la capacidad de un producto para cumplir el propósito por el cual fue diseñado.

### Métricas

Las métricas sobre la usabilidad que se han utilizado son las siguientes:

- **Exactitud.** El número de errores cometidos por los usuarios al utilizar el sistema fue muy reducido y en todos los casos fueron reproducibles.
- **Tiempo.** El tiempo medio requerido para concluir las actividades de la aplicación es de 5 segundos.
- **Recuerdo.** Los usuarios tras un periodo medio de dos meses sin utilizar la aplicación recuerdan las funcionalidades más accedidas en media, es decir, el QR se escanea únicamente una vez pero las peticiones al camarero pueden ser muchas en cada visita.
- **Respuesta emocional.** Los usuarios han reportado un grado elevado de satisfacción al realizar cualquiera de las tareas posibles de la aplicación. También han mostrado entusiasmo ante la idea de implantar el sistema en un negocio real.

## Características del sistema

**The Waiter** se basa en los criterios citados anteriormente, consiguiendo así una herramienta fácil de aprender y usar, flexible, consistente, que proporciona un buen tiempo de respuesta, se adecúa a las tareas y disminuye la carga cognitiva.

### 7.1.4. Acceso

Se realizan pruebas de este tipo para garantizar el correcto acceso de todos usuarios a aquellas funcionalidades que estén a su disposición en función de los roles y los permisos de cada uno.

Estas pruebas se han basado en comprobar que el usuario accede a todas las funcionalidades de la aplicación de forma intuitiva y que únicamente accede a aquellas cuyo rol le permite. Para ello se han utilizado 9 usuarios con las instrucciones de usar la aplicación e informar de todas las funcionalidades que encontrasen en ella. El resultado ha sido favorable, no se han detectado errores en los accesos dependientes de los permisos del tipo de usuario y de media han encontrado el 94,3 % de las funcionalidades de la aplicación.

## 7.2. Estrategia y desarrollo

En este apartado se detalla la estrategia seguida para elaborar una batería de pruebas que resulte eficaz para corroborar el correcto funcionamiento del sistema. También se exponen algunas de las pruebas que se han realizado con este fin.

### 7.2.1. Estrategia

Se han realizado pruebas unitarias sobre todos los elementos de la aplicación. El objetivo de estas pruebas ha sido conformar una estrategia de integración de dichos elementos en el proyecto. Una vez incorporados los elementos a la aplicación, se han realizado las pruebas correspondientes de integración que, finalmente, han dado lugar a las pruebas generales del sistema. La estrategia de integración que se ha utilizado consiste en dar preferencia a aquellos elementos con mayor número de funcionalidades básicas y que, por lo tanto, serían necesarias para elementos incorporados posteriormente. Cuando este criterio no ha resultado suficiente para seguir un orden en la incorporación de nuevos elementos, se ha tenido en cuenta si formaban parte de una funcionalidad principal o secundaria, es decir, si influían en el objetivo del proyecto o eran aportaciones de apoyo al objetivo principal.

### 7.2.2. Tipo de pruebas realizadas

Pese a la distinta naturaleza de los elementos que componen el sistema, el tipo de pruebas realizadas es el mismo. Esto ha aportado simplicidad en la etapa de pruebas destinando, en la medida de lo posible, el mayor número de horas a la etapa de desarrollo.

Las pruebas que hemos utilizado son las conocidas como pruebas de caja gris (caja blanca + caja negra):

**Caja blanca.** Estas pruebas se han realizado teniendo en cuenta la estructura interna de la aplicación. Su principal objetivo es explorar todas las posibilidades del flujo de la aplicación que se pueden dar en el código.

**Caja negra.** En estas pruebas no se tiene en cuenta la estructura interna de la aplicación por lo que nos centramos únicamente en las salidas que produce la aplicación ante determinadas entradas.

En este proyecto no se han realizado pruebas de caja blanca y caja negra como tal, sino que por el escaso número de desarrolladores se han realizado pruebas de caja gris. Estas pruebas consisten en una base de pruebas de caja negra centradas en puntos conflictivos que son conocidos por las personas que conocen bien la estructura interna del programa. Estas pruebas pueden resultar muy efectivas en el sentido de que se conocen bien aquellas partes de la aplicación que pueden resultar más problemáticas.

El sistema se ha probado exhaustivamente por los desarrolladores para comprobar el correcto funcionamiento de cada uno de los requisitos. Puesto que el equipo de desarrollo cuenta con una única persona, las pruebas también han sido realizadas por usuarios con conocimientos de informática y sin ellos. Para aquellos usuarios que no tenían conocimientos de informática no se han proporcionado instrucciones, puesto que una de las pruebas más importantes consistía en medir la facilidad e intuitividad que se quería para la aplicación.

Las pruebas más destacables que se han realizado sobre el sistema son las siguientes:

- Control de accesos. Comprobación de los accesos a determinadas funcionalidades en función del rol que ostente el usuario. Los accesos también se han limitado según los datos conocidos por la aplicación, es decir, sin la información del QR o de la base de datos no se ha permitido el acceso a funcionalidades que requieran de esta información.
- Interfaz de usuario. En estas pruebas se ha perseguido la comodidad de los usuarios haciéndoles en todo momento accesibles las funcionalidades necesarias de las aplicaciones. Básicamente se ha perseguido una navegación sencilla para el usuario por las pantallas de las mismas.
- Conexión. Uno de los puntos más problemáticos del sistema es el protocolo de comunicación, ya que al tratarse de un protocolo UDP, puede existir pérdida de mensajes. Para evitar esto se requiere de una recepción de señal intensa de la red para los usuarios y es algo que no se ha controlado internamente desde la aplicación. Otra solución que se tratará en el trabajo futuro, es un acercamiento al protocolo Transmission Control Protocol (TCP).

El sistema no se ha podido someter a pruebas en un negocio real pero sí se ha intentado, en la medida de lo posible, simular circunstancias parecidas. Las pruebas de integración se han realizado en el entorno de desarrollo.

## 7.3. Resultados y conclusiones

Los resultados obtenidos en estas pruebas validan aproximadamente un 90% de los requisitos funcionales planteados para este proyecto, del mismo modo, se validan también alrededor de un 80% de usuarios sin fallos (figura 7.3). Los fallos que no permiten completar el correcto funcionamiento de todos los requisitos han sido detectados pero no solventados, algunos de estos problemas se detallan a continuación:

Los fallos de compatibilidad requieren una inspección constante de las funcionalidades de Android, tanto las que hayan sido declaradas como obsoletas como las que hayan surgido en solución a esto. Las versiones tienen una media de 2 APIs y cambian, en media, 2 veces al año, por lo que no ha sido posible dar soporte a más de 5 versiones consecutivas aunque inicialmente se plantearon 8.

La red local sufre constantes pérdidas de paquetes debido al hardware utilizado. Esto se ha solventado mejorando las prestaciones del hardware utilizado para la red local. A nivel de aplicación no ha podido solventarse puesto que son elementos externos a la misma.



**Figura 7.3:** En este gráfico se muestra la estabilidad del sistema en el último año.

Pese a esto, los resultados en general son buenos. Se ha conseguido que los usuarios utilicen la herramienta de forma intuitiva, los requisitos planteados se cumplen al 97 % utilizando un router que proporcione una alta intensidad de señal en la red local, frente al 90 % que se corrobora en el estado actual del sistema. El acceso de los diferentes tipos de usuarios a diferentes funcionalidades se ha validado correctamente y se han corregido todos los errores reportados por los usuarios, obteniendo finalmente opiniones muy positivas sobre **The Waiter**.

# EVALUACIÓN

---

## 8.1. Evaluación de los usuarios

**The Waiter** se ha sometido a diversos procesos de evaluación. La evaluación realizada por el equipo de desarrollo ha sido exhaustiva pero no suficiente, debido a que tener un amplio conocimiento de la aplicación impide descubrir, en muchos casos, carencias o incluso fallos de la misma. Es por esto que **The Waiter** ha sido evaluado por un grupo de 57 usuarios externos al equipo de desarrollo, 33 con conocimientos técnicos y sobre la aplicación, y 24 sin ellos. Estos usuarios tenían conocimientos muy dispares sobre el lenguaje de desarrollo, las tecnologías y herramientas utilizadas e incluso sobre la utilidad del proyecto.

Uno de los principales objetivos, como se ha comentado en capítulos anteriores, es el desarrollo de una interfaz intuitiva y amigable, de modo que los usuarios no requieran un manual o información previa sobre la aplicación. En cambio, los negocios que quieran utilizar la aplicación si necesitarán entrar en contacto con los desarrolladores, pero únicamente para dar los primeros pasos tales como el registro y la configuración inicial.

En general, los usuarios han reportado multitud de errores y sugerencias que han enriquecido la funcionalidad de la aplicación. La mayoría de los errores eran prácticamente inadvertibles por el equipo de desarrollo, por su aparición en casos poco realistas, como, por ejemplo, la posibilidad de introducir cantidades negativas. A raíz de un fallo en la mayoría de las ocasiones se han detectado otros de la misma naturaleza. Pese a esto, fallos en situaciones como las citadas previamente, se salen del flujo de la aplicación y sólo podrían ser enmendados en su totalidad con la colaboración de un usuario de pruebas experto.

En cuanto a la interfaz y la usabilidad, los usuarios no han reportado apenas errores, sino todo lo contrario, han destacado de manera muy positiva el diseño y la usabilidad de la misma.

La herramienta Firebase citada anteriormente ha desarrollado un papel fundamental en la evaluación del proyecto. Los errores reportados por los usuarios nos han dado pistas sobre dónde podrían producirse determinados fallos, pero la funcionalidad de Crash reporting nos ha permitido acumular estos errores con sus correspondientes trazas de ejecución. Esto ha dado lugar a una detección y corrección mucho más rápida y eficiente de los errores.

A parte de realizar una evaluación sobre la aplicación, era fundamental conocer el impacto que **The Waiter** podría causar en el mercado actual, incluso si las carencias de los negocios por las que comenzó este proyecto eran compartidas por otros usuarios. Para ello, se ha elaborado el estudio de mercado (apartado 2.4).

Con los datos obtenidos en esta encuesta, se confirma que los usuarios aceptarían de buen grado una aplicación de estas características, y que además, mejorar los tiempos de servicio es un campo susceptible de mejora, por lo que se vería justificada la necesidad de este proyecto.

## 8.2. Beneficios de la aplicación

Los beneficios que **The Waiter** puede proporcionar a los negocios destinados a la restauración y a los usuarios de éstos son muchos, la mayoría de los cuales se han ido comentando a lo largo del documento.

Incluir esta herramienta en los locales y los dispositivos de los clientes supondrá un método de agilización de las peticiones que los clientes realizan a los camareros. También servirá como plataforma para establecer una comunicación entre todos los usuarios de la aplicación que se encuentren conectados a la red local del establecimiento.

Uno de los beneficios más importantes, y una de las razones por las que se decidió crear este sistema, es que los clientes de un negocio puedan procesar sus solicitudes cuando así lo deseen, sin necesidad de que un camarero les mire al pasar, atienda su llamada o acuda a su mesa. Esto potencia en los clientes cierta sensación de autonomía y aumenta su satisfacción sobre la estancia en el establecimiento, sin olvidar que esto supone una gran mejora para los camareros, ya que podrán recibir todas las peticiones en un dispositivo móvil sin necesidad de ir mesa por mesa primero para conocer la naturaleza de la petición, y después para atenderla.

## 8.3. Modularidad

Como se ha comentando en secciones anteriores del documento, uno de los objetivos del desarrollo era poder extraer funcionalidades para construir así aplicaciones completamente distintas a las que se presentan.

En este sentido, se han planteado algunas de las posibles opciones que podrían existir al desligar las funcionalidades que conforman **The Waiter**:

- **Chat.** Una de las funcionalidades que presenta el proyecto es un chat abierto para todos los usuarios de la red local. Las ventajas que puede presentar con respecto a otros servicios existentes de mensajería instantánea es que carece de servidores y la información que se intercambia no es almacenada. Es más, ni siquiera las conversaciones de un usuario serán almacenadas una vez las abandone.
- **Generador de códigos QR .** Este elemento, al igual que el chat, no formaba parte explícita de los objetivos de la aplicación, pero si ha sido necesario para cumplirlos.  
Esto permite que las aplicaciones obtengan información acerca del establecimiento en el que se encuentran un cliente o un empleado. Existen multitud de generadores de códigos QR y éste, sin modificaciones, simplemente sería uno más, por lo que no sería competitivo en el mercado. Sin embargo, introduciendo algunas mejoras no contempladas en el alcance de este proyecto, sí podría serlo.
- **Test de alcoholemia.** En la aplicación **The Waiter Plus** se presenta un test de alcoholemia que tiene como objetivo garantizar, en la medida de lo posible, la seguridad de aquellos clientes que consuman bebidas alcohólicas en el establecimiento. Los cálculos que se aplican para medir la tasa son fiables pero el resultado no lo es tanto, ya que el mismo tiene que ver con el nivel de veracidad de los datos que introduzca el usuario. Aun siendo todos los datos correctos, el resultado que ofrece esta funcionalidad debe tomarse como algo meramente orientativo. Basándonos en las pruebas realizadas con usuarios, se ha puesto en duda que se utilice para fines de seguridad, en su lugar, se cree que se utilizará más como divertimento.
- **Protocolo de comunicación.** El protocolo de comunicación que se ha diseñado puede ser ex-

traído para usarlo en cualquier otra herramienta que quiera incorporar un sistema de mensajería y comparta las características de la red del proyecto, trabajando sobre un protocolo multicast sobre UDP .

Con esto se consigue un servicio de mensajería sin el uso de servidores, un sistema en el que no existe el almacenamiento de la información y los usuarios lo pueden utilizar de forma anónima; pero con un impedimento: sólo se puede establecer este servicio de mensajería entre personas que se encuentren conectadas a una misma red.

Como se ha dicho previamente, estas funcionalidades son algunas que se han planteado, pero pueden existir multitud de usos más, incluso utilizando combinaciones de las funcionalidades que puedan desligarse.



# CONCLUSIONES Y LÍNEAS FUTURAS

---

## 9.1. Conclusiones

En este trabajo se ha desarrollado y presentado un sistema compuesto por aplicaciones para dispositivos móviles, tabletas y smartwatches que permite reducir el tiempo de estancia por esperas “innecesarias”, así como una gestión más ágil de la carta e inventario por parte del administrador, en los establecimientos dedicados a ofrecer servicios de hostelería, más en concreto, servicios de restauración. Durante este proceso ha sido fundamental el trabajo de detección de las necesidades de los clientes y los establecimientos para cumplir con el objetivo principal, elaborar una herramienta “útil”, que solucione necesidades de los usuarios objetivo del sistema.

Una de las conclusiones derivadas de la experiencia con los usuarios es que sería una solución de gran utilidad por los bajos costes que supone implantar el sistema en un establecimiento y la comodidad que proporciona a los usuarios por ambas partes, tanto a los clientes como al negocio. Pese a la gran acogida que ha recibido por parte de los usuarios que lo han conocido y probado, se han recibido multitud de sugerencias que permitirían hacer más completas las opciones que ofrece la solución. Pese a esto, por la limitación temporal, y en algunos casos, de recursos, estas mejoras se han propuesto como líneas de trabajo futuro.

Las conclusiones finales del presente Trabajo de Fin de Máster son las siguientes:

- Se ha desarrollado un sistema que facilita y agiliza la gestión en los negocios de restauración.
- Se ha desarrollado una solución eficaz de comunicación entre los clientes de un establecimiento y sus trabajadores.
- Las carencias que se han observado en los resultados del estudio de mercado se han podido satisfacer prácticamente en su totalidad con el sistema desarrollado.
- La herramienta proporciona una comunicación privada sin el uso de servidores entre los usuarios del establecimiento a través de un servicio de chat, así como un servicio de mensajería instantánea entre dos usuarios.
- La interfaz que se ha diseñado se ha orientado, en todo momento, a la facilidad de manejo y entendimiento por parte de los usuarios. Se ha conseguido una interfaz sencilla e intuitiva, que cumple los estándares de Android y se ajusta a un gran número de criterios de usabilidad.
- La solución que se presenta duplica la funcionalidad que estaba prevista en un primer momento, sobre todo en el caso del aplicativo TPV, en el que el tamaño de la base de datos ha crecido considerablemente en respuesta a las propuestas de los usuarios encuestados. Se han añadido funcionalidades para graficar multitud de información, opciones de configuración, mejoras para la gestión de los productos y los proveedores, etc.
- Se ha priorizado que los elementos que componen el sistema sean reutilizables y ampliables de

una forma sencilla.

- Se ha desarrollado para ser soportada por la gran mayoría de los dispositivos Android que puedan estar actualmente en uso, aquellos cuya versión esté entre la 4.4 y la 8.0, puesto que estas versiones son las más distribuidas a fecha de impresión del documento, como se puede observar en el gráfico del anexo A.

## 9.2. Trabajo futuro

### 9.2.1. Extensibilidad

La extensibilidad fue una de las metas a alcanzar desde la primera versión de este proyecto. Gracias a los muchos esfuerzos que se pusieron en el desarrollo de la aplicación móvil **The Waiter** para el TFG, hoy es posible presentar un sistema completo en el que se encapsula esta aplicación con muchas de las mejoras que fueron propuestas en su día, dando lugar hoy a la aplicación **The Waiter Plus**.

De nuevo, en esta etapa del proyecto, la extensibilidad ha sido una de las prioridades durante el desarrollo, debido a que sigue planteado como un producto software abierto a cualquier tipo de mejora. Esto significa que, debido en gran parte a la modularidad del mismo, el sistema podría seguir incrementando su tamaño, carga y arquitectura sin necesidad de manipular total o parcialmente la mayoría de componentes implementados, y así poder extender la funcionalidad añadiendo una capa más al software de base.

### 9.2.2. Líneas futuras

Las líneas de trabajo futuro propuestas para el sistema son las siguientes:

- Incluir pasarelas de pago para completar la opción de pago electrónico de los clientes.
- Gestión de socios. Incorporar un código QR impreso en la cuenta que permita premiar a los usuarios por su fidelidad con el negocio, acumulando, por ejemplo, descuentos.
- Aumentar la comunicación entre el reloj y el teléfono para que puedan compartir información de interés como es, por ejemplo, el número de mesa, de este modo el usuario no tendrá que introducirlo a mano desde el reloj.
- Actualmente la mensajería instantánea entre dos usuarios no permite acumular mensajes anteriores, lo que evita que exista un contexto de la conversación. Puesto que se ha trabajado sin servidores y es un aspecto que no pretende cambiarse. Para almacenar el contexto de las conversaciones se utilizarán ficheros temporales en el dispositivo, o bien se hará uso de la base de datos SQLite que ya está incorporada en los dispositivos Android.
- Elaborar una página web para dar a conocer el sistema, incluyendo algunos tutoriales para los administradores de los establecimientos, además de un sistema de contacto para que los usuarios reporten sus dudas, comentarios, quejas, sugerencias, etc.
- Puesto que se trabaja sobre UDP, la comunicación puede sufrir pérdida de mensajes. Una de las opciones es implementar un servicio de control de entregas, y otra trabajar sobre un protocolo que ya lo garantice, como por ejemplo, TCP.
- Implementación de otras opciones para los usuarios. Estas ideas son fruto de las sugerencias que

se han recibido durante la evaluación de la aplicación.

- Peticiones rápidas: salero, salsas, cubiertos, servilletas.
- Incluir un buscador de productos.
- Recepción de notificaciones: mensajes, pedido recibido, etc.
- Control de pedidos. Se pretenden incorporar una serie de informes que permitan a un administrador gestionar los gastos de los pedidos realizados, tener constancia de lo que pidió, cuándo lo pidió, a que proveedor, etc.

Como se ha explicado, el proyecto se puede ampliar de muchas formas que permitan hacer cada vez más completa y útil la solución presentada, aunque el trabajo presentado cumple, con creces, los objetivos planteados al inicio del mismo.



# BIBLIOGRAFÍA

---

- [1] “Trabajo final de grado,” <https://repositorio.uam.es/handle/10486/673603>, accedido: 04/02/2018.
- [2] “Como hacer un estudio de mercado,” <https://es.wikihow.com/hacer-un-estudio-de-mercado>, accedido: 06/02/2018.
- [3] “Custom adapter implementations,” [http://www.vogella.com/tutorials/AndroidListView/article.html#adapterown\\_example](http://www.vogella.com/tutorials/AndroidListView/article.html#adapterown_example), accedido: 27/10/2017.
- [4] “Android: Elementos de ListView con varios botones cliqueables,” <https://datafull.co/p/android-elementos-de-listview-con-varios-botones-cliqueables>, accedido: 07/02/2018.
- [5] “Android: duplicar los elementos en ListVew. Quizás getView() llama demasiadas veces?” <http://es.androids.help/q24687>, accedido: 04/02/2018.
- [6] “Programación Android: Interfaz gráfica – Adapters I,” [https://elbauldelprogramador.com/programacion-android-interfaz-grafica\\_28/](https://elbauldelprogramador.com/programacion-android-interfaz-grafica_28/), accedido: 21/03/2017.
- [7] “Build an Android Instant Messaging App Using Sinch and Parse,” <https://www.sinch.com/tutorials/android-messaging-tutorial-using-sinch-and-parse/>, accedido: 12/01/2018.
- [8] “TEST de ALCOHOLEMIA: cómo calcular la cantidad de alcohol,” <http://guardiasresis.blogspot.com.es/2014/07/test-de-alcoholemia-como-calcular-la.html>, accedido: 17/01/2018.
- [9] “Tasa de alcoholemia,” <https://investigayaprende.wordpress.com/fisica/tasa-de-alcoholemia/>, accedido: 25/01/2018.
- [10] D. M. M. Sánchez, D. B. C. Martín, P. J. R. H. González *et al.*, *Metodología de desarrollo de sistemas de información*.
- [11] J. A. Gutiérrez de Mesa and C. Pagés Arévalo, *Planificación y gestión de proyectos informáticos*. Textos universitarios, Abril 2015.
- [12] “Entorno de desarrollo Android (Android Studio),” <http://www.sgoliver.net/blog/entorno-de-desarrollo-android-android-studio/>, accedido: 15/03/2017.
- [13] “Estructura de un proyecto Android (Android Studio),” <http://www.sgoliver.net/blog/estructura-de-un-proyecto-android-android-studio/>, accedido: 19/03/2017.
- [14] “Componentes de una aplicación Android,” <http://www.sgoliver.net/blog/componentes-de-una-aplicacion-android/>, accedido: 25/03/2017.
- [15] “Desarrollando una aplicación Android sencilla (Android Studio),” <http://www.sgoliver.net/blog/desarrollando-una-aplicacion-android-sencilla-android-studio/>, accedido: 27/03/2017.
- [16] A. L. Ayala, *Android Studio Curso Basico: Aprenda paso a paso*, 1st ed.
- [17] N. Smyth, *Android Studio 2 Development Essentials*.
- [18] A. Gerber and C. Craig, *Learn Android Studio: Build Android Apps Quickly and Effectively*. Apress.
- [19] “Genymotion,” <https://www.genymotion.com/>, accedido: 17/05/2017.
- [20] S. Basulto, “Tutorial de Git en Español,” [blog.santiagobasulto.com.ar/programacion/2011/11/27/tutorial-de-git-en-espanol.html](http://blog.santiagobasulto.com.ar/programacion/2011/11/27/tutorial-de-git-en-espanol.html), accedido: 12/06/2017.
- [21] “Facebook,” <https://es.wikipedia.org/wiki/Facebook>, accedido: 05/02/2018.
- [22] “Preferencias en Android I: Shared Preferences,” <http://www.sgoliver.net/blog/preferencias-en-android-i-shared-preferences/>, accedido: 15/05/2017.

- [23] “Tareas en segundo plano en Android (I): Thread y AsyncTask,” <http://www.sgoliver.net/blog/tareas-en-segundo-plano-en-android-i-thread-y-async-task/>, accedido: 12/10/2017.
- [24] Copernic, “Android Generate QR code and Barcode using Zxing,” <http://stackoverflow.com/questions/22371626/android-generate-qr-code-and-barcode-using-zxing>, accedido: 28/03/2018.
- [25] “Integrar Barcode Scanner en nuestra aplicación Android,” <https://www.adictosaltrabajo.com/tutoriales/qr-reader-android/#03>, accedido: 17/03/2017.
- [26] “Estilos Android,” <https://developer.android.com/guide/topics/ui/themes.html?hl=es-419>, accedido: 25/01/2018.
- [27] “Dashboards,” <https://developer.android.com/about/dashboards/index.html?hl=es>, accedido: 13/04/2017.
- [28] “Campañas 2007- Alcohol y menores. El alcohol te destroza por partida doble,” <http://www.msssi.gob.es/campañas/campanas07/alcoholmenores10.htm>, accedido: 17/06/2017.

# DEFINICIONES

---

**crash reporting** En informática, un crash es la condición en la cual una aplicación informática, ya sea un programa o parte o la totalidad del sistema operativo dejan de funcionar de la forma esperada y dejan de responder a otras partes del sistema. Crash report se traduce por informe de accidente, de fallo.

**framework** Es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos concretos de software, que puede servir de base para la organización y desarrollo de software..

**multicast** Consiste en el envío de la información en múltiples redes a múltiples destinos simultáneamente..

**offline** Anglicismo que se traduce literalmente como fuera de línea o fuera de red, sin conexión.

**protocolo de comunicación** Es un término utilizado en el mundo de la informática, para dar nombre a una serie de normas y criterios, los cuales, son utilizados para mantener una comunicación entre los dispositivos que forman parte de una red informática, es decir, entre los dispositivos que se encuentran conectados entre si por cualquier sistema de comunicación, sea alámbrico o inalámbrico..

**smartwatch** Un smartwatch es un reloj inteligente. La palabra, como tal, es un vocablo inglés compuesto con las palabras smart, que significa 'inteligente', y watch, que traduce 'reloj'.



# ACRÓNIMOS

---

**QR** Quick Response.

**SQLite** Structured Query Language.

**TCP** Transmission Control Protocol.

**TFG** Trabajo Final de Grado.

**TFM** Trabajo Final de Máster.

**TPV** Terminal Punto de Venta.

**UDP** User Datagram Protocol.



I

# ANEXOS



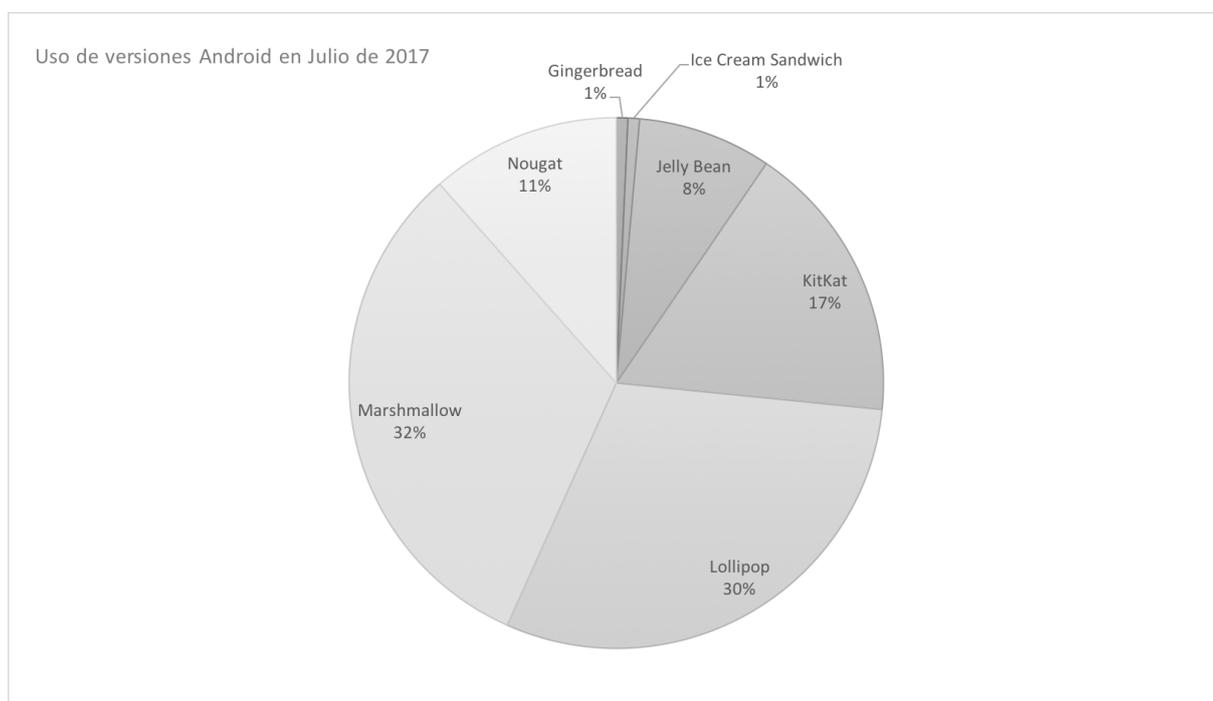
# ANDROID

---

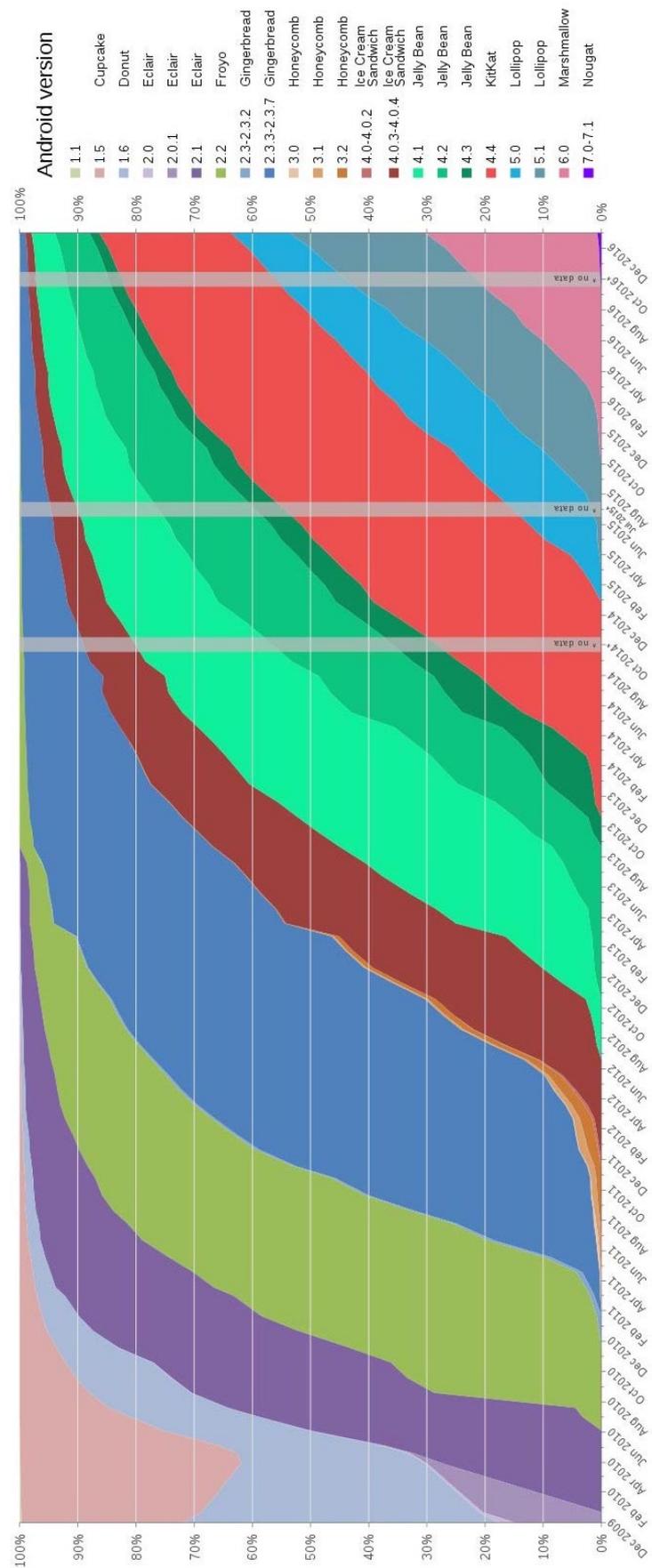
Android, a fecha de impresión de este documento, ha comercializado 16 versiones cuyo detalle se muestra en la tabla A.1. El uso que han tenido estas versiones se detalla en el gráfico de la figura A.1 [27].

Versión	Fecha de publicación	Nivel API	Nombre
Beta	05/11/2007		
1.0	23/09/2008	1	Astro
1.1	09/02/2009	2	Bender
1.5	30/04/2009	3	Cupcake
1.6	15/09/2009	4	Donut
2.0 / 2.1	26/10/2009	5(2.0) 6(2.0.1) 7(2.1.x)	Eclair
2.2	20/05/2009	8	Froyo
2.3	06/12/2010	9(2.3 - 2.3.2) 10(2.3.3 - 2.3.7)	Gingerbread
3.0 / 3.1 / 3.2	22/02/2011	11(3.0.x) 12(3.1) 13(3.2)	Honeycomb
4.0	19/10/2011	14(4.0 - 4.0.2) 15(4.0.3 - 4.0.4)	Ice Cream Sandwich
4.1	09/07/2012	16(4.1.1)	Jelly Bean
4.2	13/11/2012	17(4.2)	Jelly Bean
4.3	24/07/2013	18	Jelly Bean
4.4	31/11/2013	19	KitKat
5.0	12/11/2014	21	Lollipop
6.0	05/10/2015	23	Marshmallow
7.0 - 7.1.2	15/06/2016	24 - 25	Nougat
8.0	21/08/2017	26	Oreo

**Tabla A.1:** Versiones de Android, fechas de publicación y nombres.



**Figura A.1:** Uso de versiones Android en Julio de 2017.



**Figura A.2:** Distribución de las distintas versiones Android en Enero de 2017.



# RESULTADOS DEL ESTUDIO DE MERCADO

## B.0.1. Configuración del formulario

### Configuración

GENERAL   PRESENTACIÓN   TEST DE AUTOEVALUACIÓN

Recopilar las direcciones de correo electrónico

Acuse de recibo de respuestas ?

**Requiere iniciar sesión:**

Limitar a 1 respuesta

Los encuestados deberán iniciar sesión en Google.

---

**Los encuestados pueden:**

Editar después de enviar

Ver gráficos de resumen y respuestas de texto

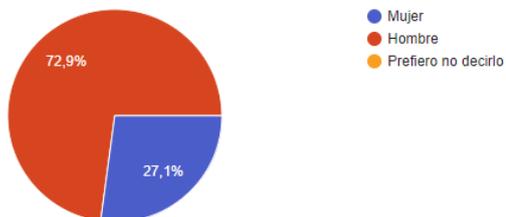
CANCELAR   GUARDAR

Figura B.1: Configuración del formulario.

## B.0.2. Resumen de resultados

### Seleccione su sexo

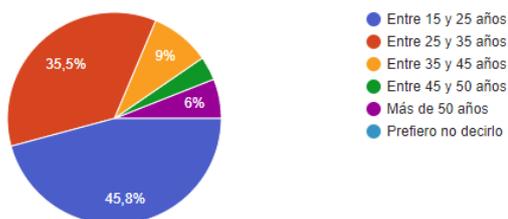
166 respuestas



**Figura B.2:** Gráfico 1 de resultados del formulario.

### Seleccione su edad

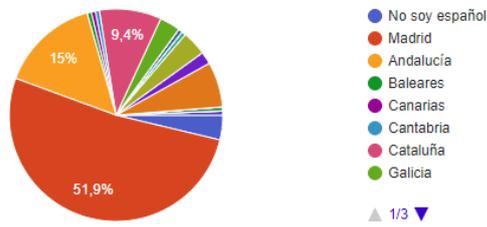
166 respuestas



**Figura B.3:** Gráfico 2 de resultados del formulario.

### Seleccione la Comunidad Autónoma en la que reside

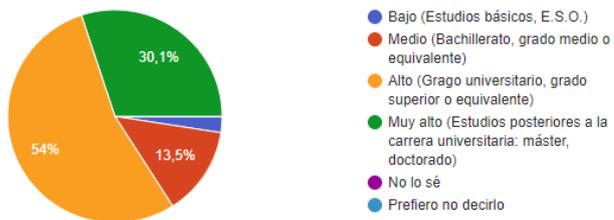
160 respuestas



**Figura B.4:** Gráfico 3 de resultados del formulario.

### Seleccione su nivel de formación

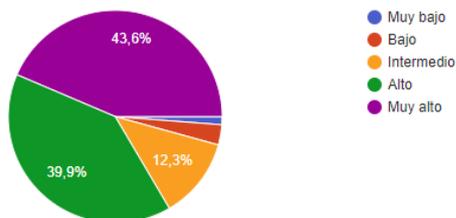
163 respuestas



**Figura B.5:** Gráfico 4 de resultados del formulario.

### Seleccione su grado de comodidad con la tecnología

163 respuestas

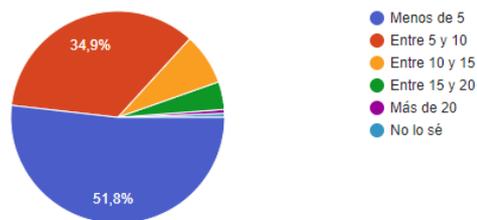


**Figura B.6:** Gráfico 5 de resultados del formulario.

Indique el número de visitas que realizó a negocios de restauración en los 7 días anteriores a la realización de esta encuesta



166 respuestas



**Figura B.7:** Gráfico 6 de resultados del formulario.

Indique el tiempo medio que tardó el servicio del establecimiento en dejar su mesa o su zona limpia de consumiciones anteriores o desperdicios



166 respuestas



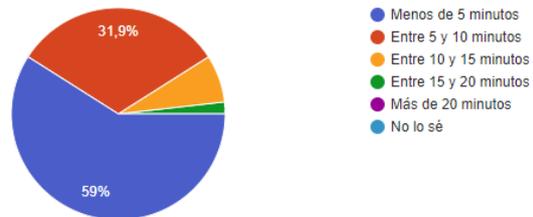
**Figura B.8:** Gráfico 7 de resultados del formulario.

---

Indique el tiempo medio en el que el servicio del establecimiento inició el primer contacto con usted



166 respuestas

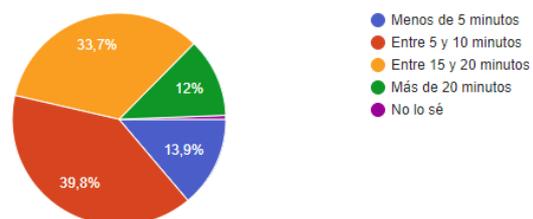


**Figura B.9:** Gráfico 8 de resultados del formulario.

Indique el tiempo medio hasta que el servicio le entregó su pedido completo



166 respuestas

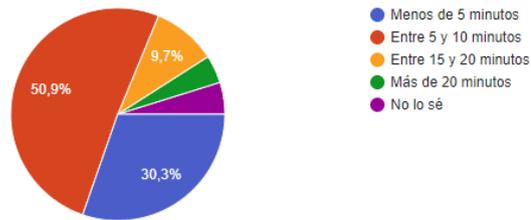


**Figura B.10:** Gráfico 9 de resultados del formulario.

Indique el tiempo medio que esperó para pedir la cuenta más el tiempo que tardó, una vez pedida, en realizar el pago y recibir su cambio si fuera necesario



165 respuestas

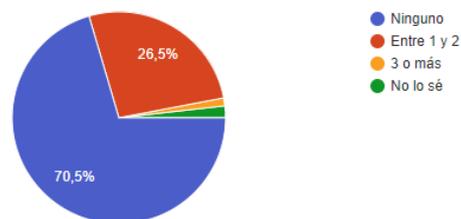


**Figura B.11:** Gráfico 10 de resultados del formulario.

Indique el número medio de errores que cometió el servicio con su pedido



166 respuestas



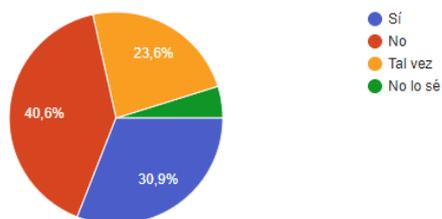
**Figura B.12:** Gráfico 11 de resultados del formulario.

---

¿Cree que unas fechas en las que estos negocios se encuentren especialmente concurridos (saturados) de clientes han podido afectar a las respuestas de su encuesta?



165 respuestas

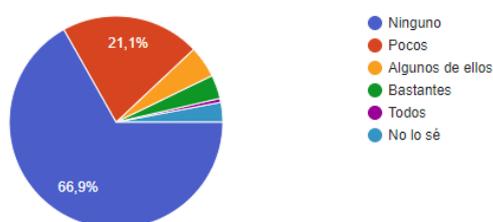


**Figura B.13:** Gráfico 12 de resultados del formulario.

¿Los sitios que frecuenta permiten que realice las peticiones desde su dispositivo móvil?



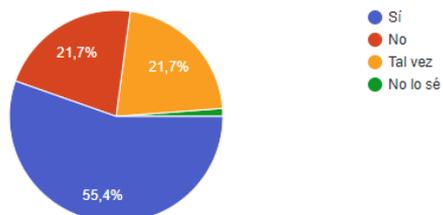
166 respuestas



**Figura B.14:** Gráfico 13 de resultados del formulario.

Con respecto a la pregunta anterior, ¿le gustaría poder hacerlo?

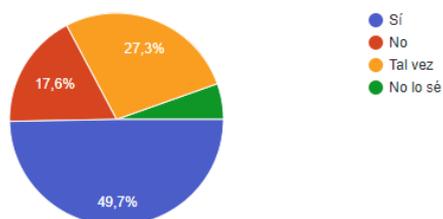
166 respuestas



**Figura B.15:** Gráfico 14 de resultados del formulario.

¿Considera que utilizar el teléfono móvil para las funciones descritas anteriormente agilizaría el servicio de los establecimientos y mejoraría su experiencia como cliente?

165 respuestas



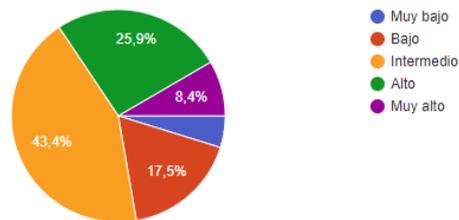
**Figura B.16:** Gráfico 15 de resultados del formulario.

---

Indique el grado de molestia que le supone esperar a que el servicio del establecimiento lo atienda o tarde en realizar su petición



166 respuestas

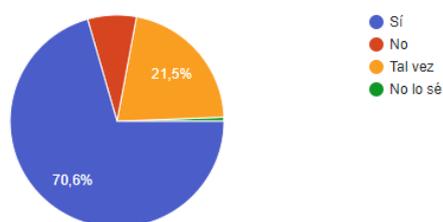


**Figura B.17:** Gráfico 16 de resultados del formulario.

¿Le gustaría poder enviar al establecimiento sus valoraciones y comentarios sobre el servicio recibido, los productos consumidos, etc.?



163 respuestas

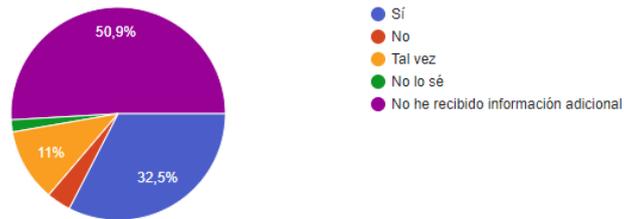


**Figura B.18:** Gráfico 17 de resultados del formulario.

Si ha recibido información sobre el sistema que se ha implementado para este proyecto, ¿cree que supondría una mejora implantarlo en los negocios?



163 respuestas

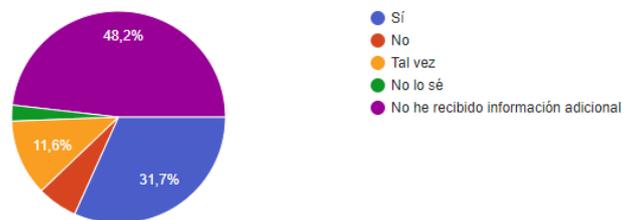


**Figura B.19:** Gráfico 18 de resultados del formulario.

De nuevo, si ha recibido información, en su caso personal, ¿haría uso del sistema?



164 respuestas

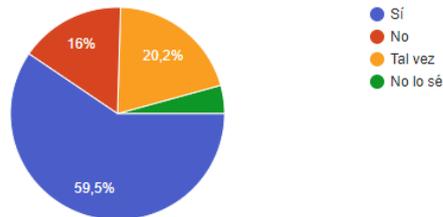


**Figura B.20:** Gráfico 19 de resultados del formulario.

Tanto si ha recibido información adicional sobre el sistema como si no, en términos generales, ¿descargaría una aplicación en su teléfono que le permitiese realizar en un establecimiento de restauración las peticiones oportunas?



163 respuestas



**Figura B.21:** Gráfico 20 de resultados del formulario.

La siguiente parte solo deben responderla aquellas personas que han trabajado o trabajan actualmente en negocios de restauración (si no cumple esto, pulse siguiente, verá otro formulario, sin rellenarlo baje hasta el final y envíelo)

### Encuesta sobre la eficiencia en la gestión de los negocios de restauración

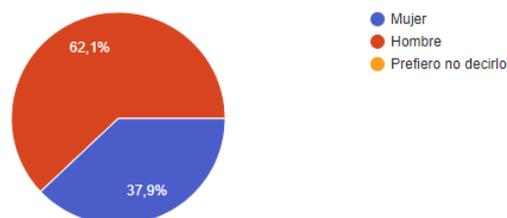
(RESPONDER SOLO SI HA TRABAJADO EN ESTOS NEGOCIOS O LO HACE EN LA ACTUALIDAD)

**Figura B.22:** Información sobre la segunda parte del formulario.

Seleccione su sexo



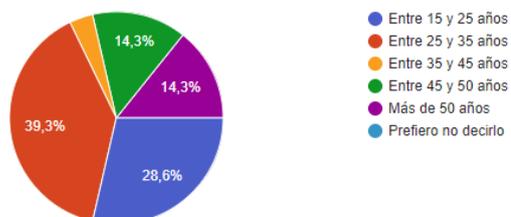
29 respuestas



**Figura B.23:** Gráfico 21 de resultados del formulario.

### Seleccione su edad

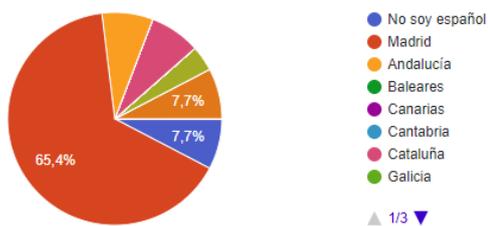
28 respuestas



**Figura B.24:** Gráfico 22 de resultados del formulario.

### Seleccione la Comunidad Autónoma en la que reside

26 respuestas



**Figura B.25:** Gráfico 23 de resultados del formulario.

Seleccione la Comunidad Autónoma en la que trabaja o ha trabajado en este tipo de negocios

24 respuestas

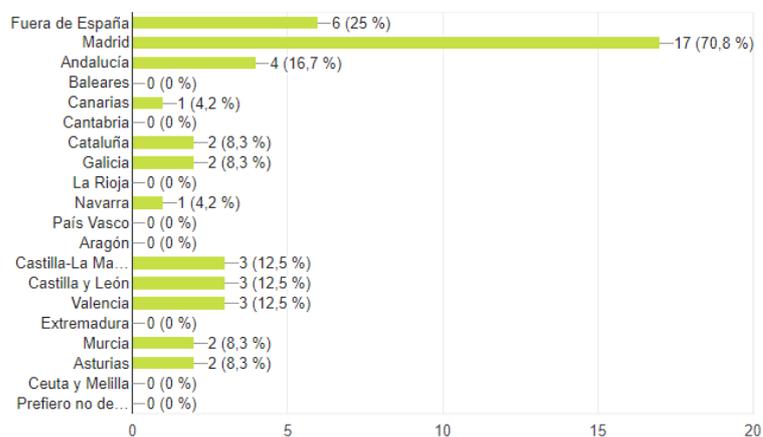


Figura B.26: Gráfico 24 de resultados del formulario.

Seleccione su nivel de formación

24 respuestas

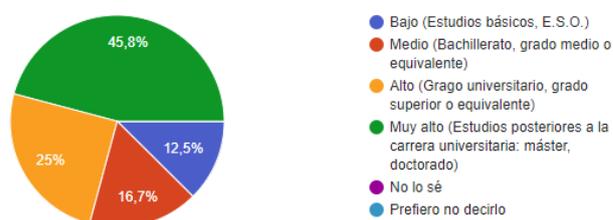
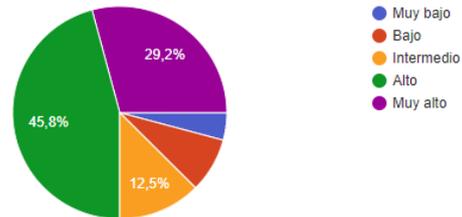


Figura B.27: Gráfico 25 de resultados del formulario.

Seleccione su grado de comodidad con la tecnología

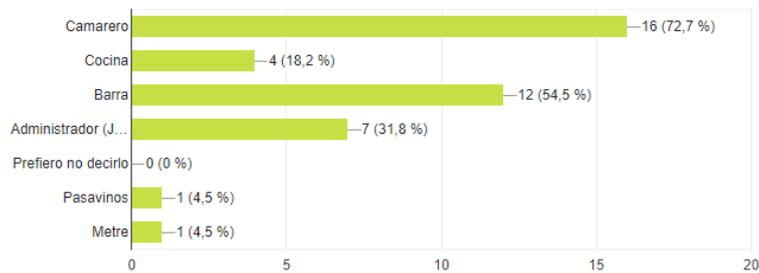
24 respuestas



**Figura B.28:** Gráfico 26 de resultados del formulario.

¿Que puesto o puestos ha desempeñado al trabajar para estos negocios?

22 respuestas

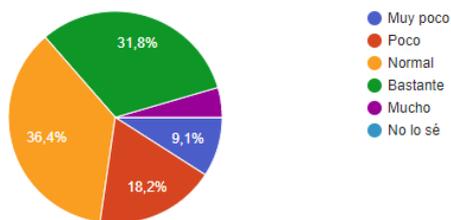


**Figura B.29:** Gráfico 27 de resultados del formulario.

¿Considera que el servicio proporcionado por el establecimiento en el que trabaja es rápido?



22 respuestas

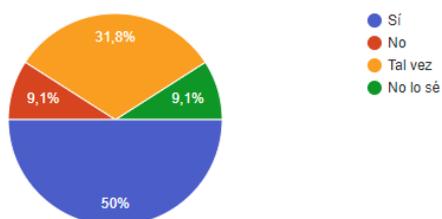


**Figura B.30:** Gráfico 28 de resultados del formulario.

¿Considera que sería más rápido si pudiera recibir todas las peticiones de los clientes en su teléfono móvil?



22 respuestas

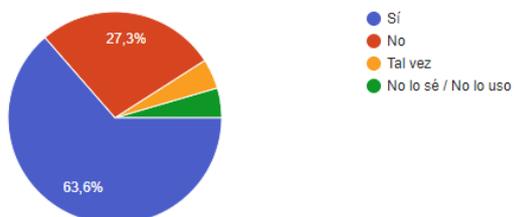


**Figura B.31:** Gráfico 29 de resultados del formulario.

¿Considera que el dispositivo principal que utilizan (caja registradora, TPV...) es sencillo de usar?



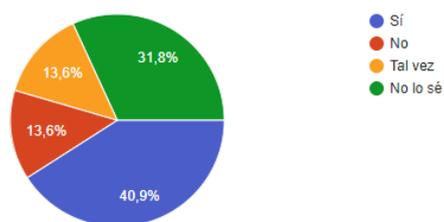
22 respuestas



**Figura B.32:** Gráfico 30 de resultados del formulario.

¿Considera que los precios del dispositivo utilizado y su mantenimiento son elevados?

22 respuestas

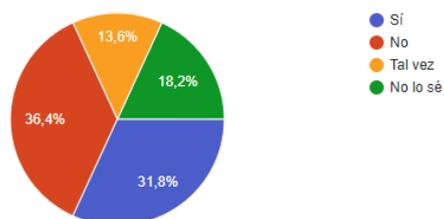


**Figura B.33:** Gráfico 31 de resultados del formulario.

¿Considera que el dispositivo citado anteriormente ofrece toda la funcionalidad deseada?



22 respuestas

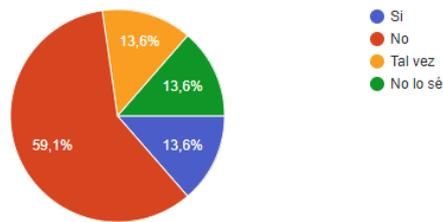


**Figura B.34:** Gráfico 32 de resultados del formulario.

El dispositivo, ¿permite llevar una gestión detallada del inventario del establecimiento?



22 respuestas

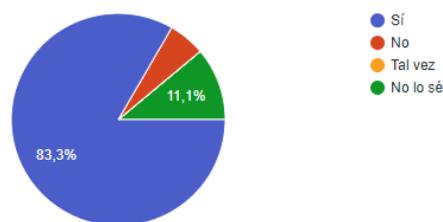


**Figura B.35:** Gráfico 33 de resultados del formulario.

En caso de responder negativamente la pregunta anterior, ¿le gustaría que el dispositivo tuviera dicha funcionalidad?



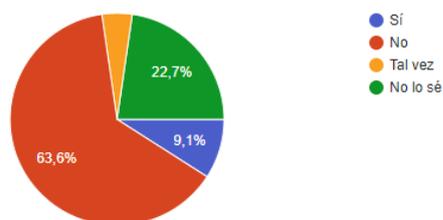
18 respuestas



**Figura B.36:** Gráfico 34 de resultados del formulario.

El dispositivo, ¿puede gestionar los proveedores de los productos del establecimiento?

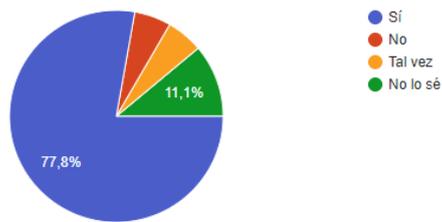
22 respuestas



**Figura B.37:** Gráfico 35 de resultados del formulario.

En caso de responder negativamente la pregunta anterior, ¿le gustaría que el dispositivo tuviera dicha funcionalidad?

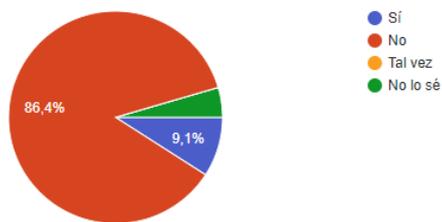
18 respuestas



**Figura B.38:** Gráfico 36 de resultados del formulario.

El dispositivo, ¿puede gestionar las valoraciones y comentarios de los clientes?

22 respuestas



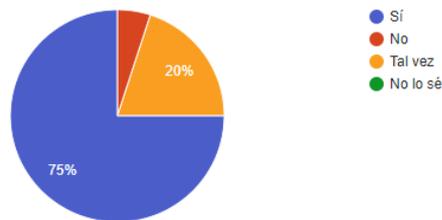
**Figura B.39:** Gráfico 37 de resultados del formulario.

---

En caso de responder negativamente la pregunta anterior, ¿le gustaría que el dispositivo tuviera dicha funcionalidad?



20 respuestas

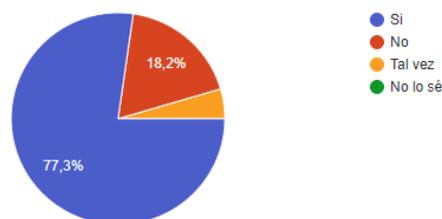


**Figura B.40:** Gráfico 38 de resultados del formulario.

El dispositivo, ¿es único en el establecimiento y utilizado por todos los empleados?



22 respuestas

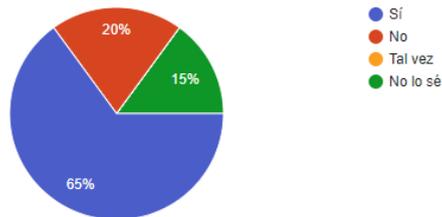


**Figura B.41:** Gráfico 39 de resultados del formulario.

En caso de responder afirmativamente la pregunta anterior, ¿cree que sería más eficaz descentralizar la funcionalidad de dicho dispositivo y permitir que cada empleado haga una gestión individual con un dispositivo propio?



20 respuestas

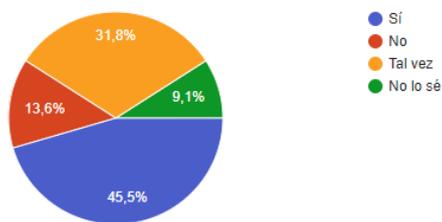


**Figura B.42:** Gráfico 40 de resultados del formulario.

¿Considera que incorporar tecnología en su establecimiento, o si ya ha integrado alguna tecnología, supone un gran gasto?



22 respuestas

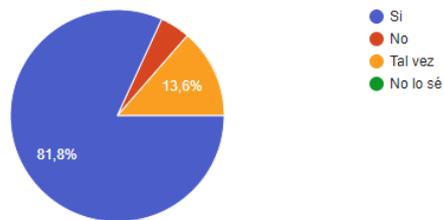


**Figura B.43:** Gráfico 41 de resultados del formulario.

---

¿Considera que incorporar tecnología en un negocio de este tipo es algo positivo para el mismo? 

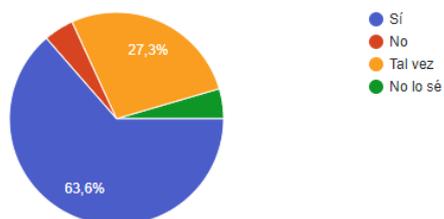
22 respuestas



**Figura B.44:** Gráfico 42 de resultados del formulario.

¿Considera que a los clientes les gustaría poder realizar algunas acciones de manera más autónoma? 

22 respuestas



**Figura B.45:** Gráfico 43 de resultados del formulario.

Si ha recibido información sobre el sistema que se ha implementado para este proyecto, ¿cree que supondría una mejora implantarlo en su negocio?



22 respuestas

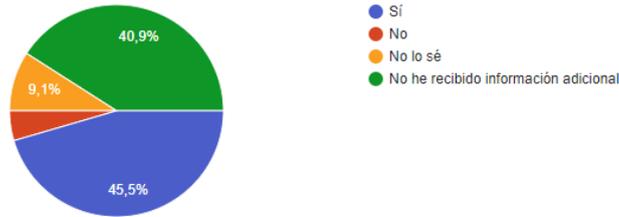


Figura B.46: Gráfico 44 de resultados del formulario.

De nuevo, si ha recibido información, en su caso personal, ¿haría uso del sistema?

22 respuestas

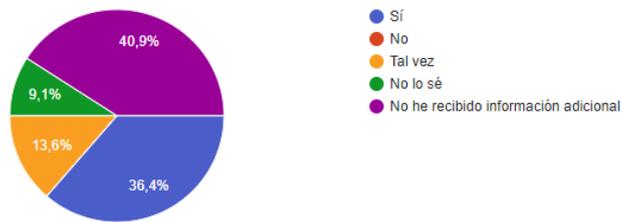


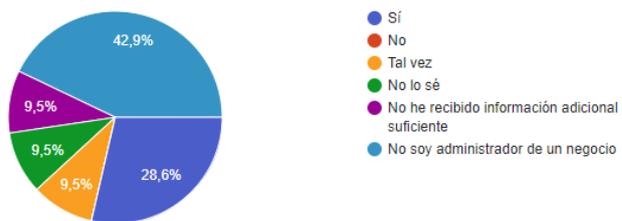
Figura B.47: Gráfico 45 de resultados del formulario.

---

En caso de ser el administrador de un negocio de este tipo y haber recibido la información suficiente, ¿integraría este sistema en su establecimiento?



21 respuestas



**Figura B.48:** Gráfico 46 de resultados del formulario.



# PLANIFICACIÓN

---

En la figura C.1 se puede observar el diagrama de Gantt completo del proyecto.

Esta planificación se ha dividido en 18 tareas principales, compuestas a su vez de subtareas, pero que por claridad no son mostradas en este diagrama:

- Definición.
- Planificación.
- Diseño de la interfaz.
- Diseño de la navegación entre pantallas.
- Diseño del protocolo de comunicación.
- Diseño de la base de datos.
- Primera revisión.
- Implementación de la base de datos.
- Implementación del protocolo de comunicación.
- Implementación de las funcionalidades de la aplicación para la gestión del negocio.
- Trabajo con la aplicación de empleados y clientes.
- Segunda revisión.
- Implementación de la aplicación para el reloj.
- Recogida de los datos de las encuestas.
- Elaboración de un estudio de mercado.
- Pruebas de implantación en un negocio real.
- Tercera revisión.
- Documentación.

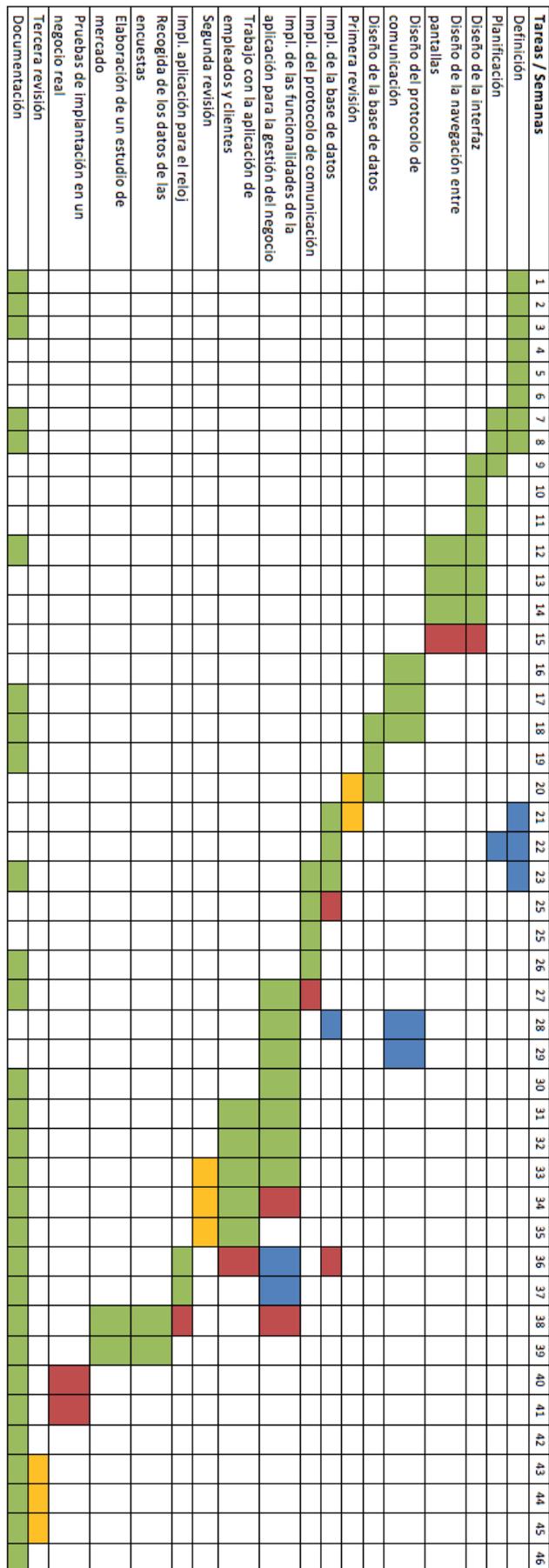


Figura C.1: Diagrama de Gantt con la planificación del proyecto.

# GENERACIÓN DE CÓDIGOS QR

---

```
private void generateQRCode_general(String data, ImageView img) throws WriterException
{

    com.google.zxing.MultiFormatWriter writer = new com.google.zxing.MultiFormatWriter();
    String finaldata = Uri.encode(data, "utf-8");

    BitMatrix bm = writer.encode(finaldata, BarcodeFormat.QR_CODE, 150, 150);
    Bitmap ImageBitmap = Bitmap.createBitmap(150, 150, Bitmap.Config.ARGB_8888);

    for (int i = 0; i < 150; i++) { // Width

        for (int j = 0; j < 150; j++) { // Height

            ImageBitmap.setPixel(i, j, bm.get(i, j) ? Color.BLACK: Color.WHITE);

        }
    }

    if (ImageBitmap != null) {

        img.setImageBitmap(ImageBitmap);

    } else {

        Toast.makeText(getApplicationContext(), "Error al generar el código QR",
            Toast.LENGTH_SHORT).show();

    }
}
```

**Código D.1:** Código utilizado para generar los códigos QR de la aplicación



# ENVÍO DE MENSAJES POR LA RED

---

```
package es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaiterplus.Comunicacion;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaiterplus.MainActivity;

public class SendMulticast {

    public static void sendMulticast(String dir, int port, String msgout) throws IOException {

        byte[] buf = msgout.getBytes();
        InetAddress address = InetAddress.getByName(dir);
        DatagramSocket sendSocket = new DatagramSocket();
        DatagramPacket msgPacket = new DatagramPacket(buf, buf.length, address, port);
        sendSocket.send(msgPacket);
        sendSocket.close();
    }

    public static void sendMulticast(int port, String msgout) throws IOException {

        byte[] buf = msgout.getBytes();
        InetAddress address = InetAddress.getByName(MainActivity.getMiRed().getIp());
        DatagramSocket sendSocket = new DatagramSocket();
        DatagramPacket msgPacket = new DatagramPacket(buf, buf.length, address, port);
        sendSocket.send(msgPacket);
        sendSocket.close();
    }

    public static void sendMulticast(String msgout) throws IOException {

        byte[] buf = msgout.getBytes();
        InetAddress address = InetAddress.getByName(MainActivity.getMiRed().getMulticast());
        DatagramSocket sendSocket = new DatagramSocket();
        DatagramPacket msgPacket = new DatagramPacket(buf, buf.length, address,
            MainActivity.getMiRed().getPuerto());
        sendSocket.send(msgPacket);
        sendSocket.close();
    }
}
```

**Código E.1:** Código utilizado para enviar los mensajes



# RECEPCIÓN DE MENSAJES DE LA RED

---

```
package es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaiterplus.Comunicacion;

import android.app.IntentService;
import android.content.Intent;
import android.os.StrictMode;
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.InetAddress;
import java.net.MulticastSocket;
import java.net.UnknownHostException;
import java.util.ArrayList;
import es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaiterplus.MainActivity;
import es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaiterplus.Notificacion;
import es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaiterplus.TipoPeticon;

public class ServicioReceptor extends IntentService {

    private MulticastSocket receiveSocket;
    private static final String DEFAULT_IP = "224.0.0.1";
    private static final Integer DEFAULT_PORT = 5355;
    private static final Integer DEFAULT_TIMEOUT = 0;

    private String ultimoMensaje = null;

    public ServicioReceptor() {

        super("ServicioReceptorCliente"); // Con esto recibe nombre el hilo de trabajo
    }

    @Override
    protected void onHandleIntent(Intent intent) {

        ControlLogs.logInfo("Servicio_receptor_--_ACTIVO");

        StrictMode.ThreadPolicy policy = new
            StrictMode.ThreadPolicy.Builder().permitAll().build();
        StrictMode.setThreadPolicy(policy);

        doInBackground();
    }
}
```

**Código F.1:** Código utilizado para recibir los mensajes

```

// Este metodo aloja la funcionalidad a realizar por el servicio cuando este activo.
// En nuestro caso esta encapsula en un while infinito, que implica que estara activo siempre

protected void doInBackground() {

    String recMsg;

    while (true) {

        recMsg = receptorMulticast();
        synchronized (this) {
            try {
                wait(2*1000);
                ultimoMensaje = null;
            }
            catch (Exception e) {
            }
        }

        if (recMsg != null)
            parseoMensaje(recMsg);
    }
}

public String receptorMulticast(){

    String recMsg;
    InetAddress ia = null;
    byte[] buffer = new byte[1024];
    MulticastSocket ms = null;
    int port = 5355;

    try {

        ia = InetAddress.getByName(DEFAULT_IP);
        DatagramPacket dp = new DatagramPacket(buffer, buffer.length, ia, port);
        ms = new MulticastSocket(DEFAULT_PORT);
        ms.joinGroup(ia);
        ms.receive(dp);

        byte[] buf2 = new byte[dp.getLength()];

        for (int i = 0; i < dp.getLength(); ++i)
            buf2[i] = buffer[i];

        recMsg = new String(buf2);

        if (recMsg.equals(ultimoMensaje)) return null;

        ultimoMensaje = recMsg;
        return recMsg;

    } catch (UnknownHostException e){
        ControlLogs.logError(e.getMessage());
    } catch (IOException e) {
        ControlLogs.logError(e.getMessage());
    }
}

return null;
}

```

**Código F.2:** Código utilizado para recibir los mensajes

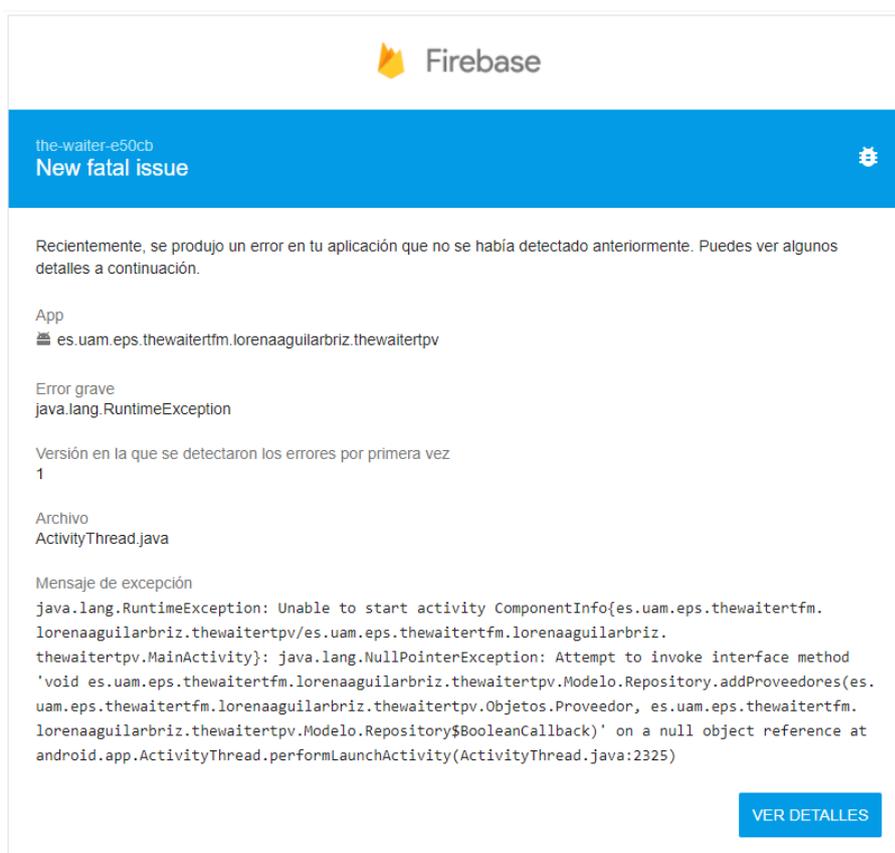
```

public void parseoMensaje(String... recMsgs){
    for(String s : recMsgs){
        String[] tokens = s.split("\\|");
        // Comprobacion de la clave de verificacion
        if(tokens[0].equals("25072013")){
            if (tokens[1].equals("CAMARERO")) {
                String tokens2[] = tokens[2].split("::");
                Notificacion n = new Notificacion(0, tokens2[0], Integer.parseInt(tokens2[2]),
                    TipoPeticon.valueOf(tokens2[1]));
                TratamientoNotificaciones.notificacionCamarero(n);
            } else if (tokens[1].equals("CLIENTE")) {
                TratamientoMensajes.mensajeRecibidoCliente(m);
            } else if (tokens[1].equals("ADMIN")){
                TratamientoMensajes.mensajeRecibidoAdmin(m);
            } else {
                return;
            }
        }
    }
}

```

**Código F.3:** Código utilizado para recibir los mensajes





The screenshot shows an email notification from Firebase. At the top, there is the Firebase logo and the text "New fatal issue". Below this, the email body contains the following information:

- App: `es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaitertpv`
- Error grave: `java.lang.RuntimeException`
- Versión en la que se detectaron los errores por primera vez: `1`
- Archivo: `ActivityThread.java`
- Mensaje de excepción: `java.lang.RuntimeException: Unable to start activity ComponentInfo{es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaitertpv/es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaitertpv.MainActivity}: java.lang.NullPointerException: Attempt to invoke interface method 'void es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaitertpv.Modelo.Repository.addProveedores(es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaitertpv.Objetos.Proveedor, es.uam.eps.thewaitertfm.lorenaaguilarbriz.thewaitertpv.Modelo.Repository$BooleanCallback)' on a null object reference at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2325)`

At the bottom right of the email body, there is a blue button labeled "VER DETALLES".

**Figura G.1:** Email recibido cuando se genera un informe de fallos.

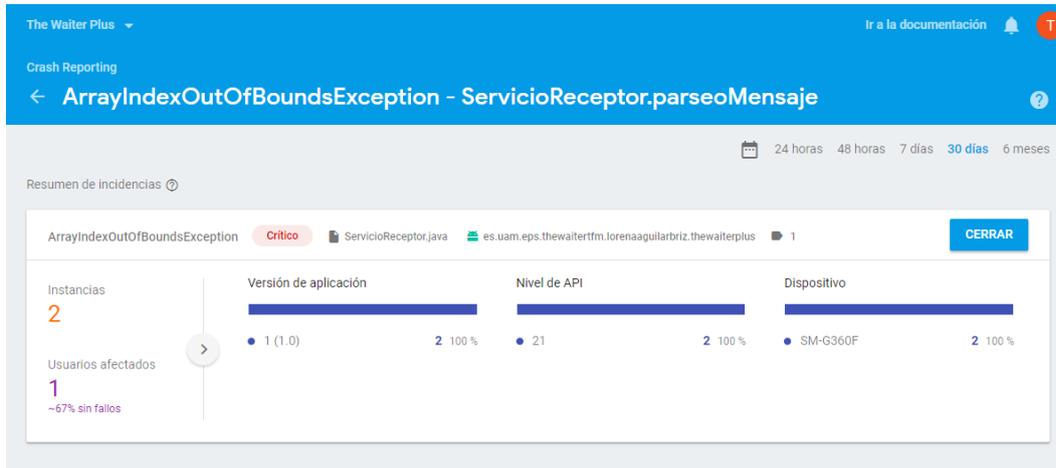


Figura G.2: Reporte.

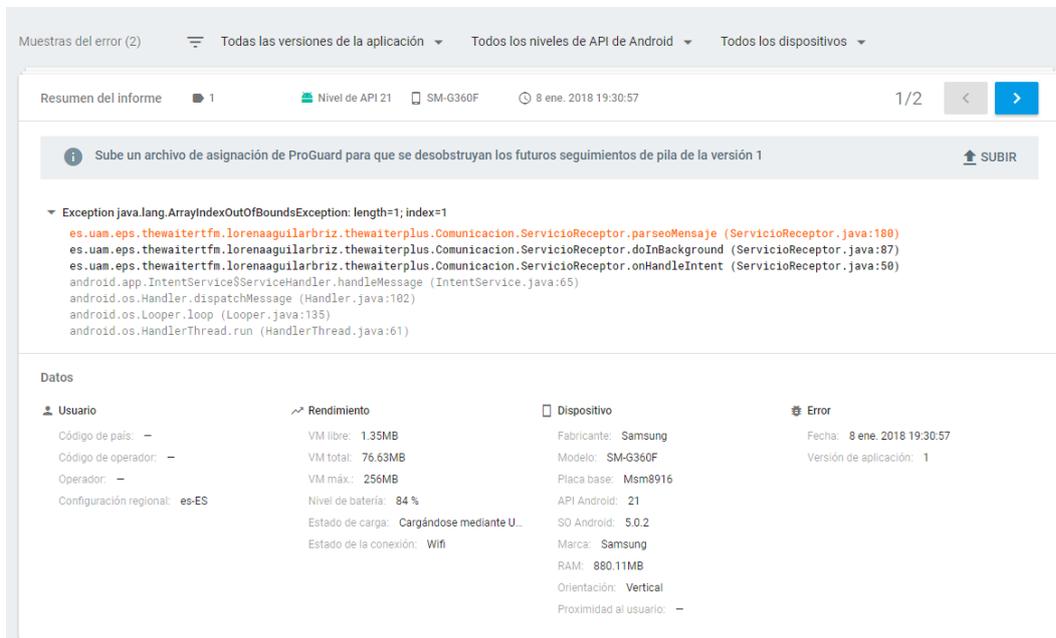
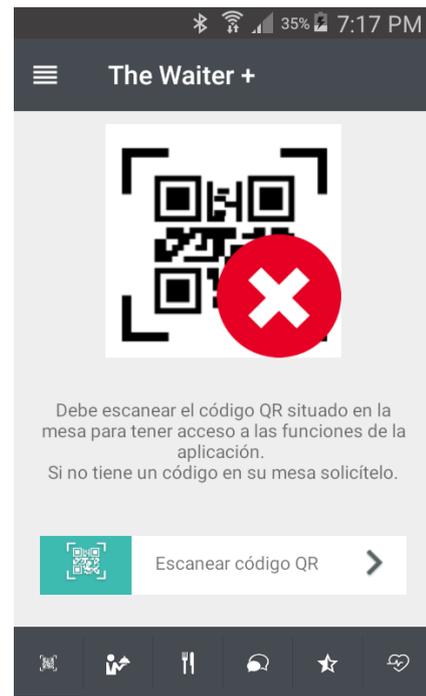


Figura G.3: Detalle del reporte.

# INTERFAZ *The Waiter Plus*

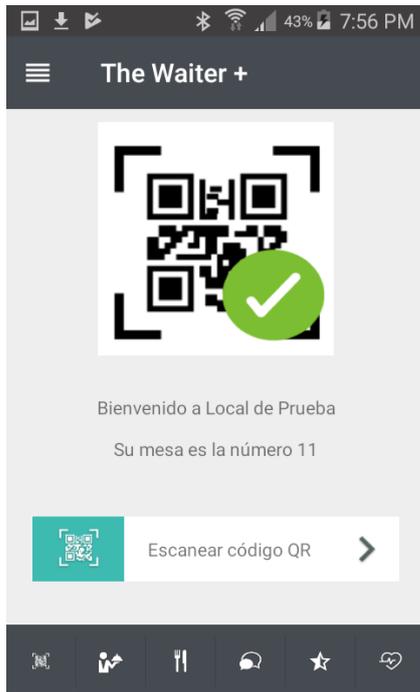


(a) Pantalla principal



(b) Pantalla de lectura del QR

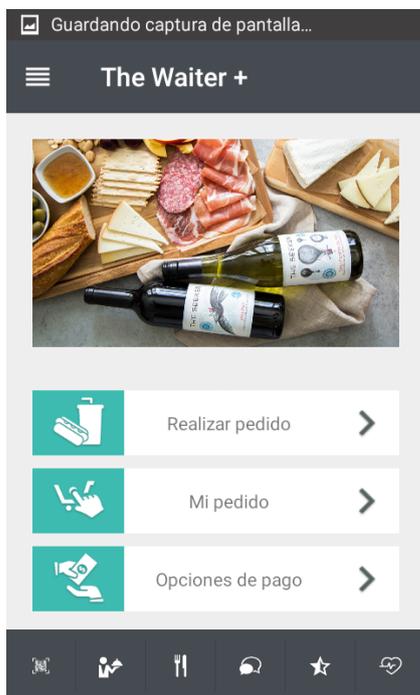
**Figura H.1:** Imágenes de la aplicación *The Waiter Plus*



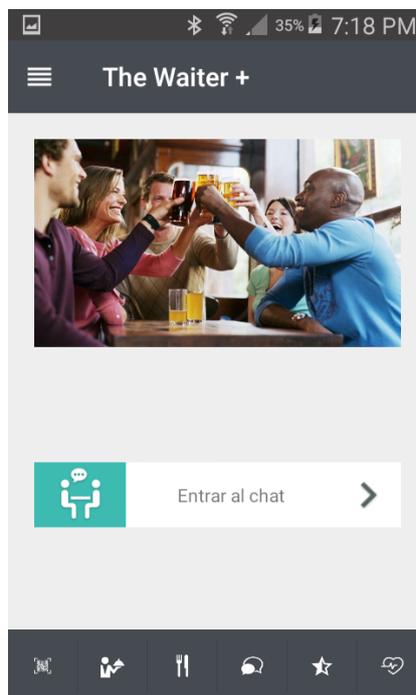
(a) Pantalla de lectura del QR



(b) Solicitudes al camarero

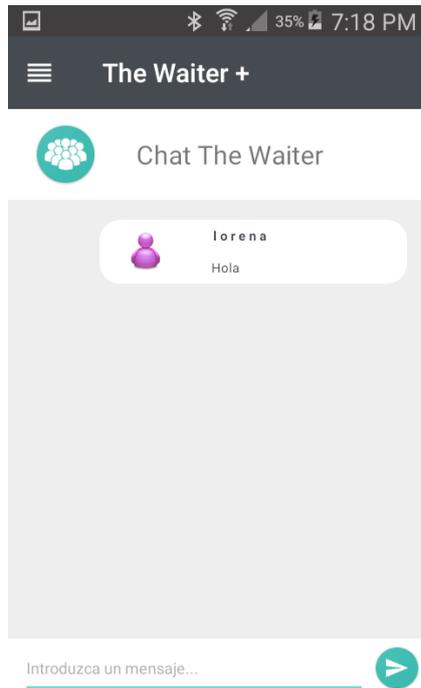


(c) Pedido y pago

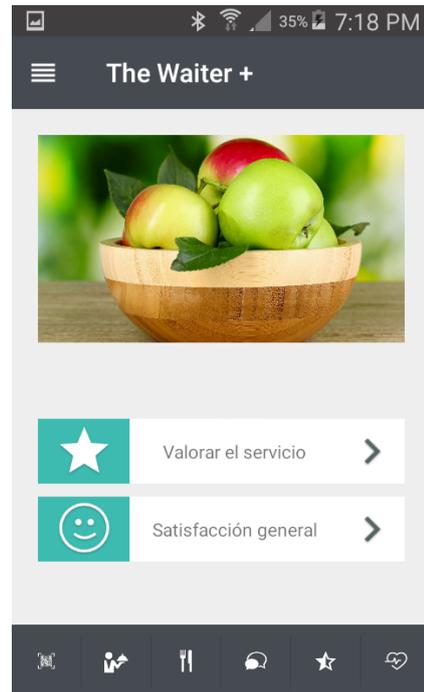


(d) Acceso al chat

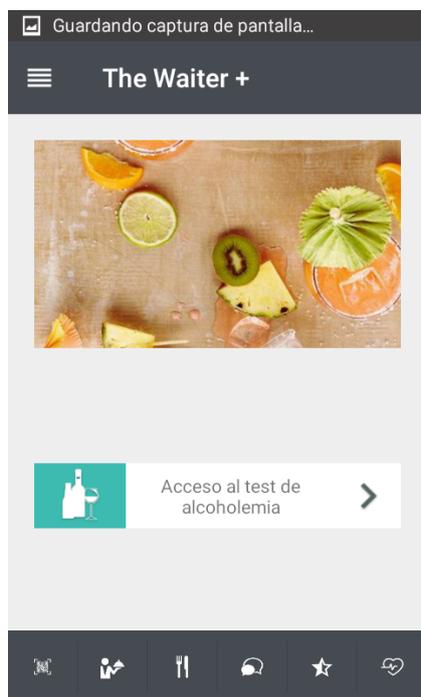
**Figura H.2:** Imágenes de la aplicación *The Waiter Plus*



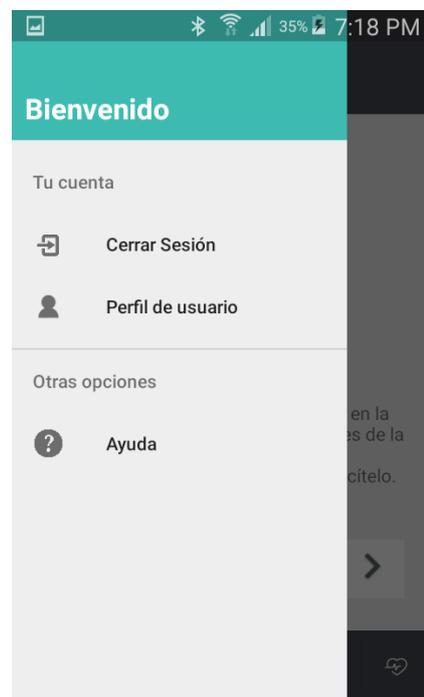
(a) Chat global



(b) Valoraciones y satisfacción

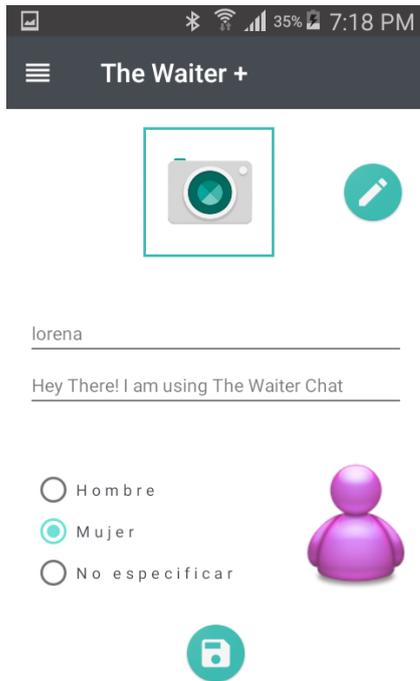


(c) Test de alcoholemia

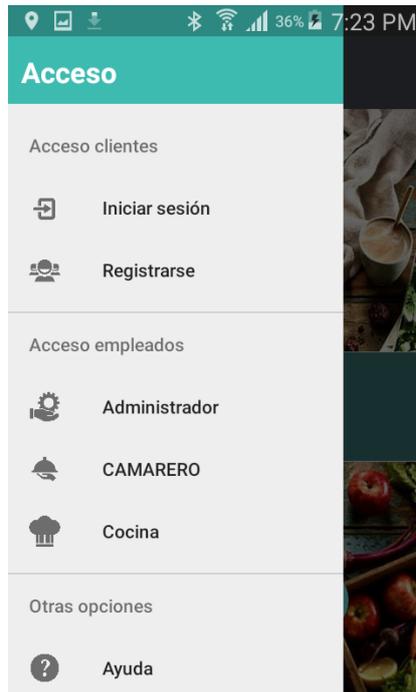


(d) Menú latera clientes

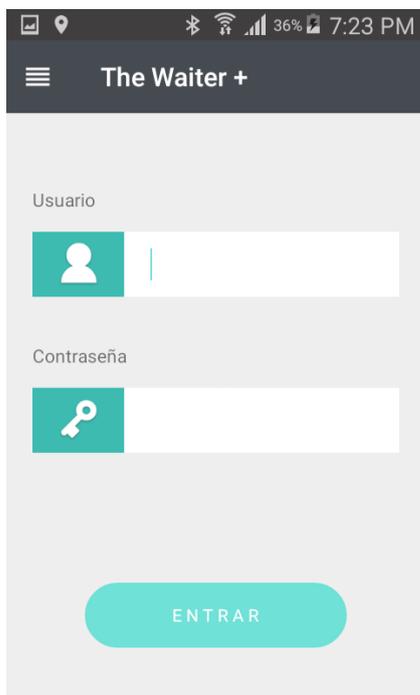
**Figura H.3:** Imágenes de la aplicación *The Waiter Plus*



(a) Perfil de usuario



(b) Menú lateral

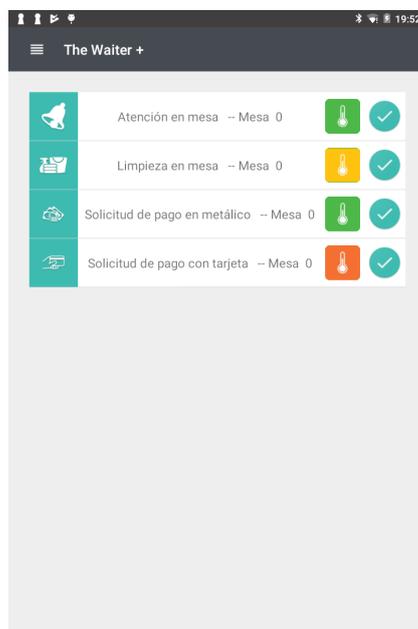


(c) Autenticación empleado



(d) Pantalla de lectura del QR para empleados

**Figura H.4:** Imágenes de la aplicación *The Waiter Plus*



(a) Listado de notificaciones de los camareros

**Figura H.5:** Imágenes de la aplicación *The Waiter Plus*



# INTERFAZ *The Waiter Wear*



(a) Pantalla principal

Seleccione su mesa

2
3
4

(b) Selección de mesa

Mesa



Llamar camarero



Solicitar limpieza

(c) Peticiones de atención y limpieza



Solicitar cuenta

Pago en efectivo



Pago con tarjeta



(d) Peticiones de pago

Satisfacción general



(e) Satisfacción general

Valoración del servicio

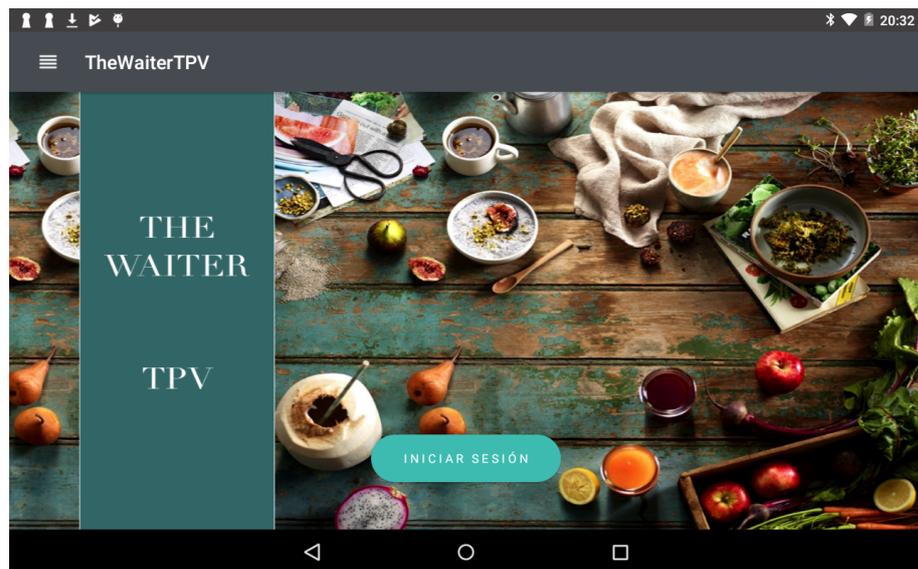


(f) Valoración

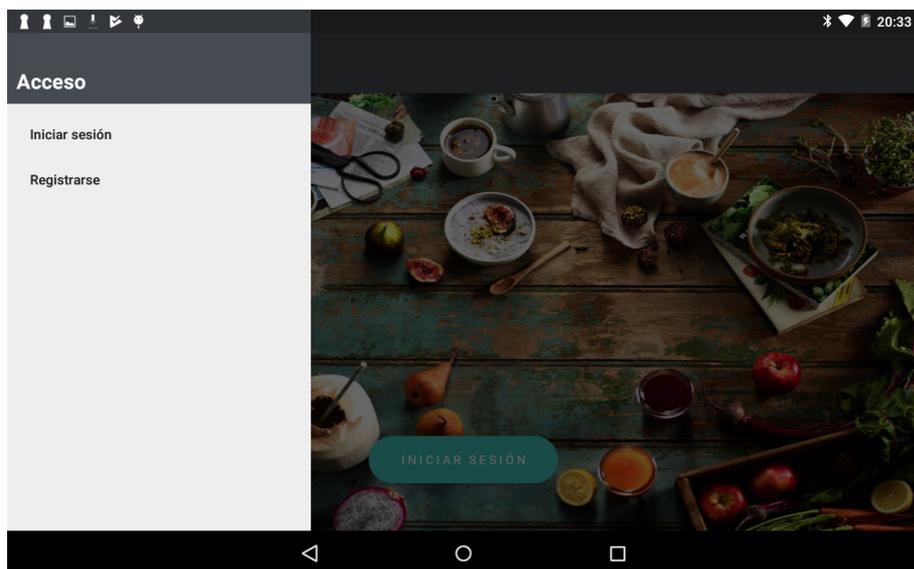
**Figura I.1:** Imágenes de la aplicación *The Waiter Wear*



# INTERFAZ *The Waiter TPV*

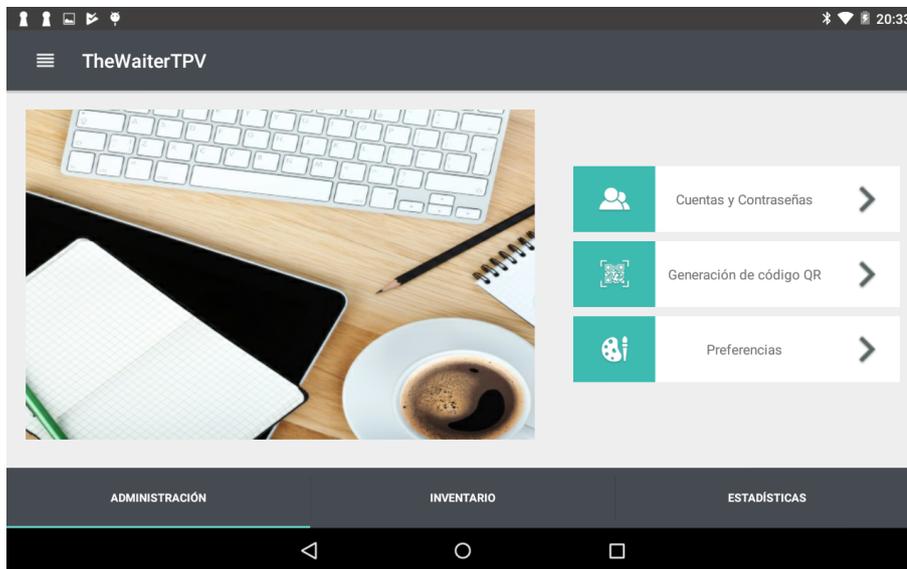


(a) Pantalla principal

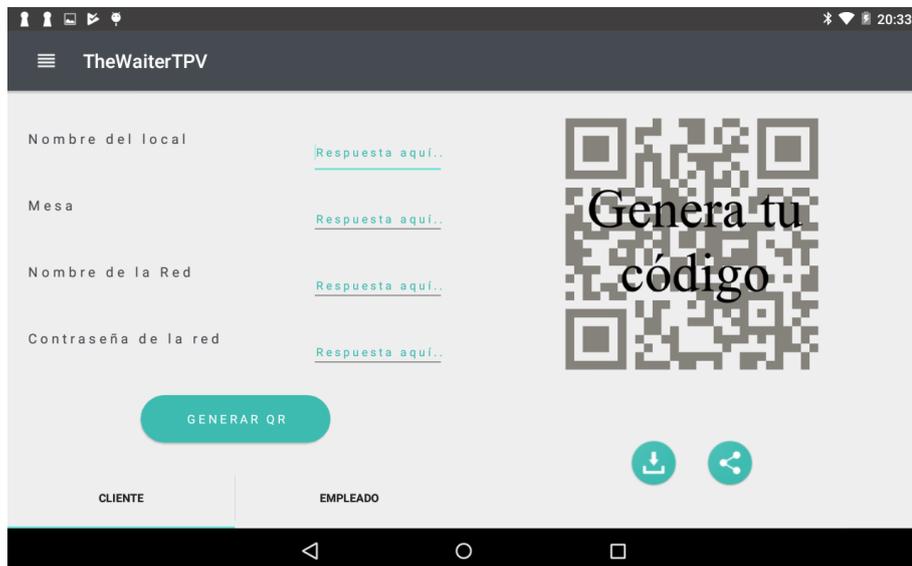


(b) Menú lateral

**Figura J.1:** Imágenes de la aplicación *The Waiter TPV*

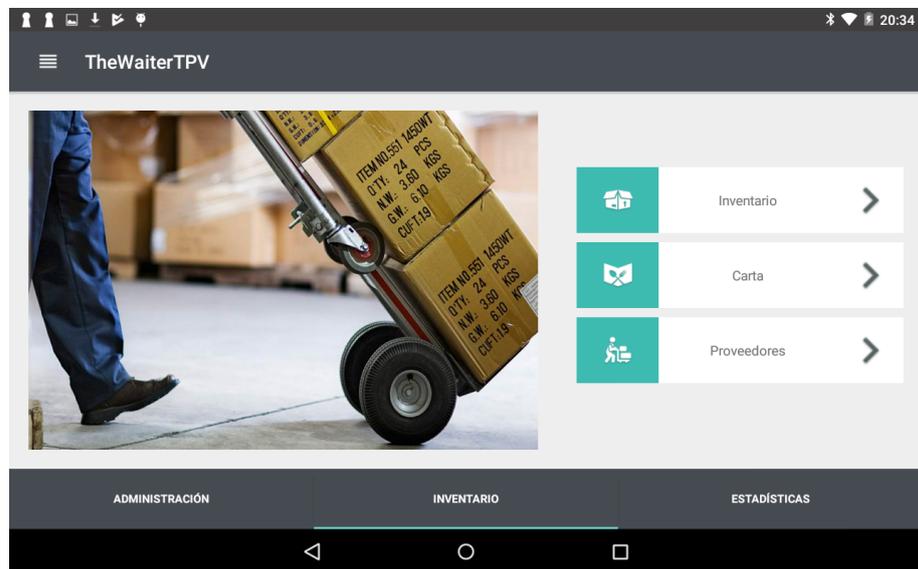


(a) Pestaña de administración

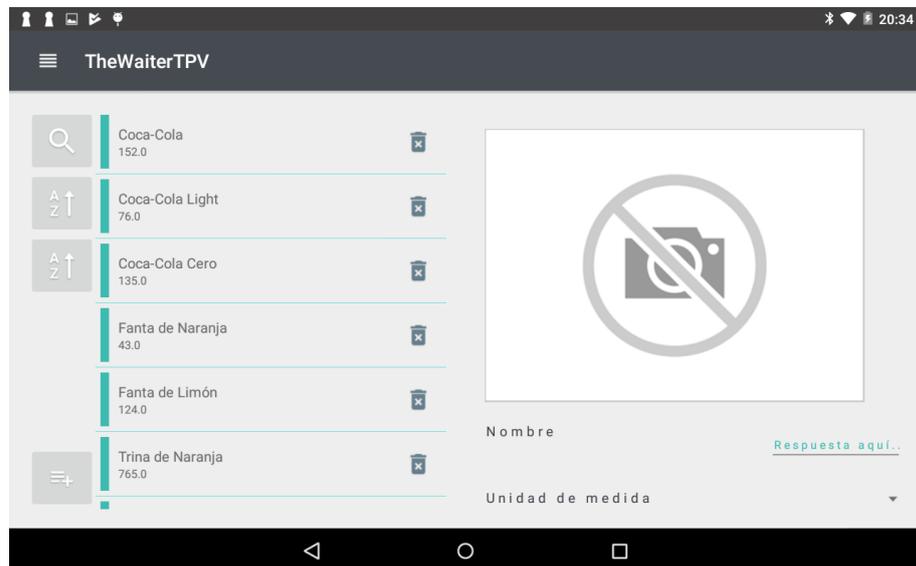


(b) Generación de códigos QR

**Figura J.2:** Imágenes de la aplicación *The Waiter TPV*



(a) Pestaña de inventario

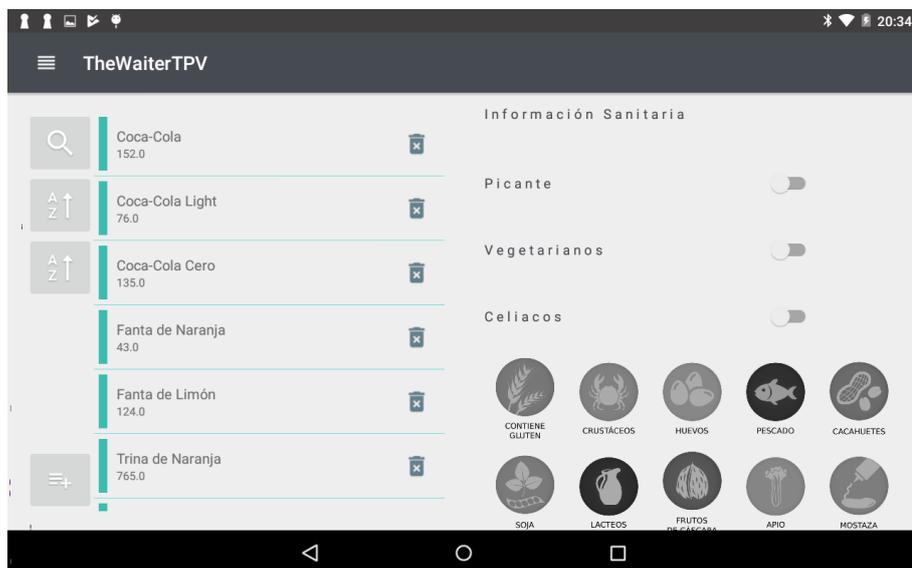


(b) Detalle del inventario

**Figura J.3:** Imágenes de la aplicación *The Waiter TPV*

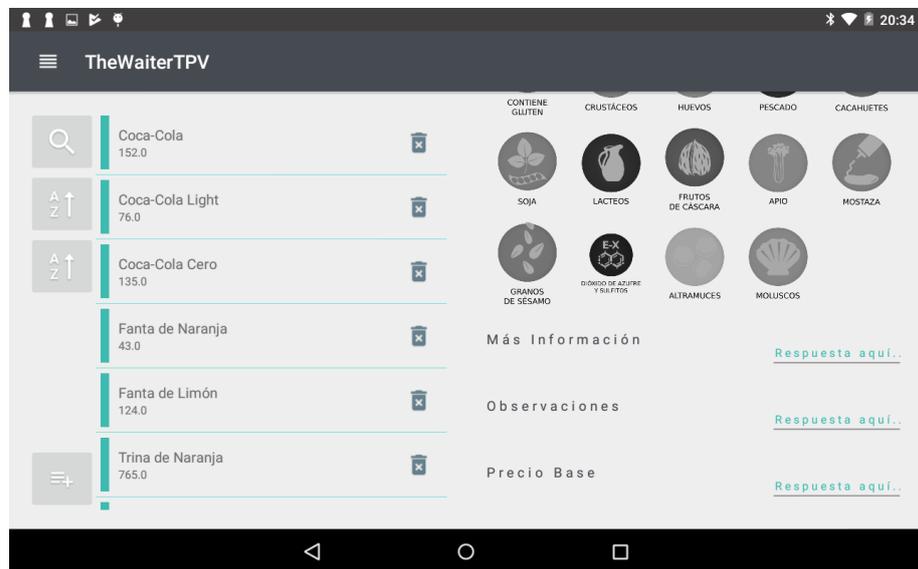


(a) Detalle del inventario

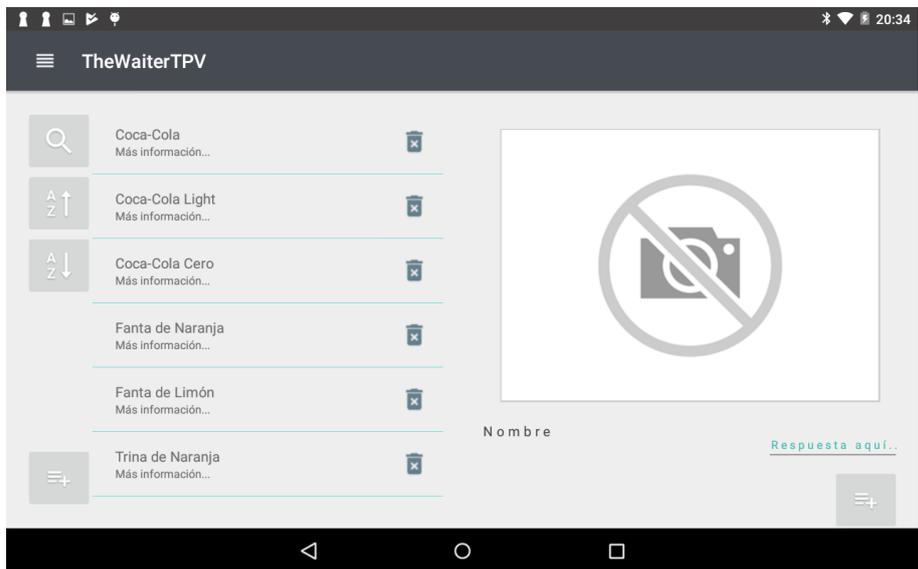


(b) Detalle del inventario

**Figura J.4:** Imágenes de la aplicación *The Waiter TPV*

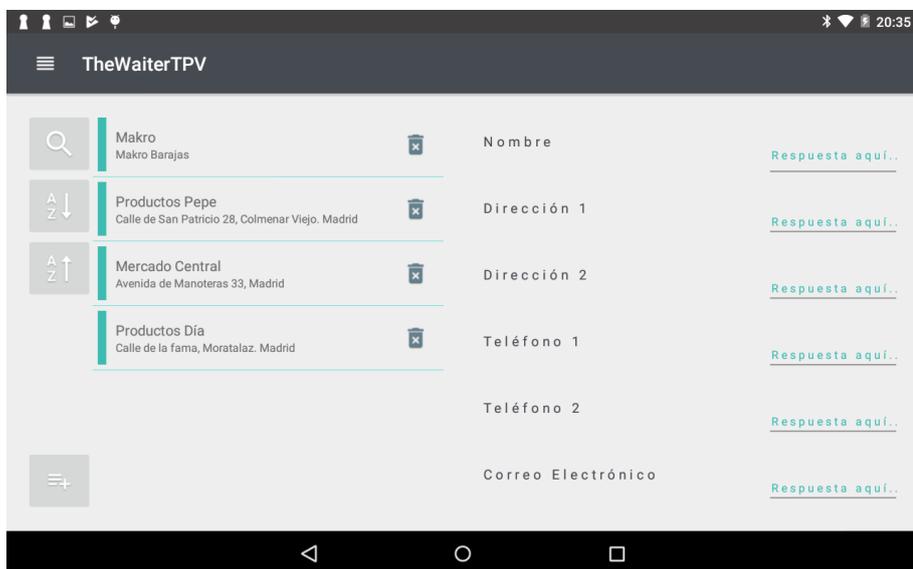


(a) Detalle del inventario

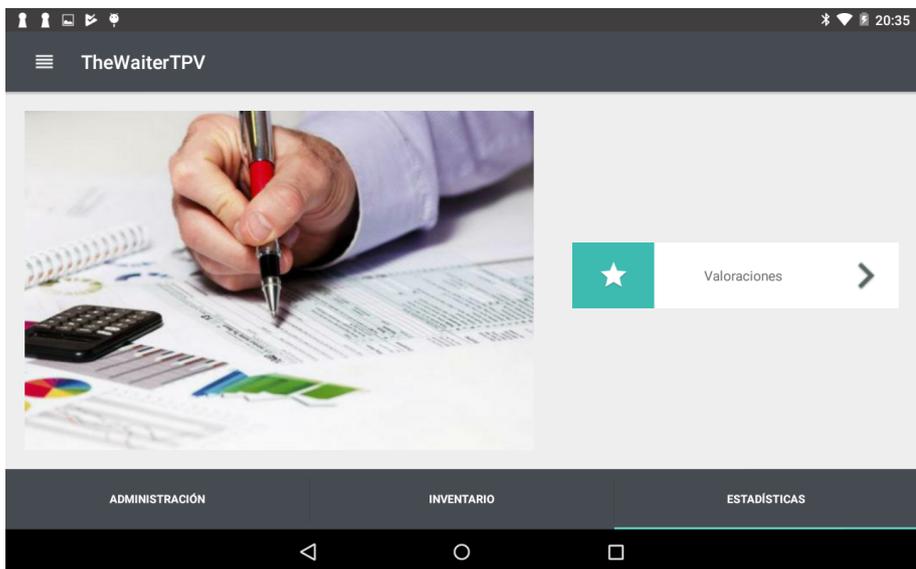


(b) Detalle de la carta

**Figura J.5:** Imágenes de la aplicación *The Waiter TPV*

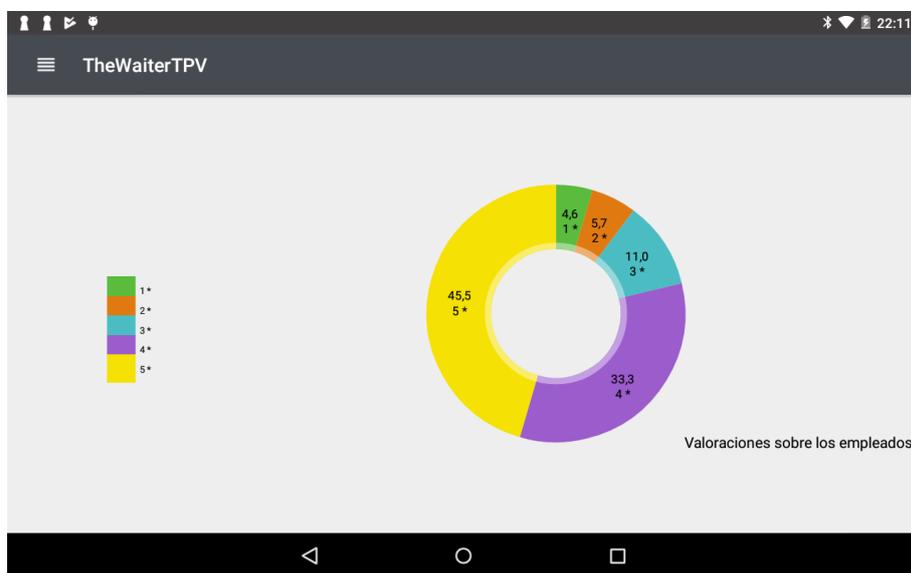


(a) Detalle de los proveedores



(b) Pestaña de estadísticas

**Figura J.6:** Imágenes de la aplicación *The Waiter TPV*



(a) Gráfico de valoraciones sobre los empleados

**Figura J.7:** Imágenes de la aplicación *The Waiter TPV*





# TEST DE ALCOHOLEMIA

---

## K.0.1. Introducción

El test de alcoholemia es uno de los elementos del proyecto. En su origen se implementó para garantizar, en la medida de lo posible, la seguridad de los usuarios que ingieran bebidas alcohólicas en los establecimientos de hostelería. Tras las pruebas realizadas se ha detectado que es utilizada por los usuarios con fines recreativos. Aunque no se puede estimar la tasa de alcoholemia de forma exacta, como lo harían unos análisis de sangre, las fórmulas utilizadas para este cálculo son muy efectivas, siempre que los datos se acerquen lo más posible a la realidad. Las fuentes han sido obtenidas de la página web oficial del Ministerio de Sanidad, Servicios Sociales e Igualdad [28].

## K.0.2. Datos

El cálculo de la tasa de alcoholemia de los usuarios se ha realizado basándonos en los datos que se muestran a continuación:

Número de UBE	Alcoholemia Grs./l	Efectos físicos y psicológicos
1	0,2 - 0,3	Sin efectos evidentes. Ligera elevación del estado de ánimo.
2	0,5 - 0,6	Relajación, calor, disminución del tiempo de reacción y de la coordinación fina.
5	1,4 - 1,5	Alteración mayor del control físico y mental: habla y visión difíciles.
7	2	Pérdida del control motor (requieren de ayuda) confusión mental.
10	3	Intoxicación severa; control consciente mínimo.
14	4	Inconsciencia; umbral del estado de coma.
17	5	Coma profundo.
20	6	Muerte por depresión respiratoria.

**Tabla K.1:** Detalle de los datos utilizados para el cálculo de la tasa de alcoholemia.

## K.0.3. Cómo calcular la tasa de alcoholemia

Como las bebidas tienen distintas concentraciones de alcohol, las consumiciones se traducen a Unidades de Bebida Estándar (UBE): Una UBE equivale a 10 Grs. de alcohol.

**Calculo de los gramos de alcohol:**

Gramos de alcohol = Volumen (expresado en c.c.) x Graduación x densidad / 100

- graduación = Graduación de etanol en la bebida
- densidad = Densidad de alcohol (0,8)

Una vez realizada esta operación aun no hemos obtenido la tasa de alcoholemia de un individuo, debemos aplicar factores referentes al peso y al sexo de dicho individuo de la siguiente forma:

El peso lo multiplicaremos por un factor de la siguiente forma:

- x 0,7 en el caso de los hombres
- x 0,6 en el caso de las mujeres

Para continuar se divide el resultado de la primera operación entre el peso del individuo por el factor correspondiente a su sexo.

Finalmente, teniendo en cuenta que el alcohol no se almacena de forma indefinida en el cuerpo, se debe tener en cuenta el factor de eliminación. Este factor es 0,0025 que tendrá que ser multiplicado por el número de minutos transcurridos desde la primera consumición.

El dato final se obtiene restándole al dato calculado previamente en la operación anterior de la tasa de eliminación (minutos transcurridos desde la primera consumición x 0,0025).

Ahora se tiene la tasa en Gramos/Litro y se aplica la tabla de medidas que se proporcionó al principio del apartado para estimar los resultados de la tasa de alcoholemia sobre un determinado individuo.

Es importante recordar que la tasa de alcoholemia permitida en sangre para conductores está por debajo de 0,5 g/L en sangre (o lo que es lo mismo, 0,25 mg/L en aire espirado), y para noveles y profesionales por debajo de 0,3 g/L.