



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación web móvil
para la gestión de comandas en
restaurantes

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: José Espasa Saval

Tutor: Manuela Albert Albiol

Tutor: María Victoria Torres Bosch

Curso 2020/2021

Resumen

El Trabajo Final de Grado que se presenta a continuación consiste en desarrollar una aplicación web diseñada especialmente para dispositivos móviles en la que se permita pedir comida de un restaurante, así como reservar mesa, hacer y/o programar pedidos y consultar una opinión sobre este. Esta aplicación está pensada no solo para los consumidores sino también como una carta de presentación del restaurante.

La aplicación también permite a los usuarios elegir el método de pago entre varias alternativas. Entre ellas se podrá elegir un pago único con tarjeta de crédito como convencionalmente se conoce y también permitirá el pago fraccionado entre varios usuarios mediante el pago por bizum.

Palabras clave: restaurante, comida, aplicación web móvil, web, Spring, bizum.

Abstract

The Final Degree Project presented below consists of developing a web application designed especially for mobile devices in which it is possible to order food from a restaurant, as well as reserve a table, place and / or schedule orders and consult an opinion about it. This application is intended not only for consumers but also as a restaurant cover letter.

The application also allows users to choose the payment method among several alternatives. Among them, you can choose a single payment with a credit card as it is conventionally known and it will also allow fractional payment between several users by paying by bizum.

Keywords: restaurant, food, mobile web app, web, Spring, bizum.

Tabla de contenidos

1.	Introducción	9
1.1	Motivación	9
1.2	Objetivos	10
1.3	Estructura.....	10
2.	Diseño centrado en el usuario.....	13
2.1	Fases del proceso DCU.....	13
2.2	Aplicación del enfoque DCU en el proyecto.....	14
3.	Estado del arte.....	17
3.1	Apps de restaurantes.....	18
3.1.1	Alma Libre.....	18
3.1.2	La Fábrica de Hielo	19
3.2	Servicios a restaurantes	20
3.2.1	Pidopago.....	20
3.2.2	PortalRest	21
3.3	Limitaciones de las aplicaciones	22
4.	Análisis de Necesidades de Usuario.....	23
4.1	Investigación cualitativa.....	23
4.2	Definición de Persona.....	25
4.3	Escenarios	26
5.	Diseño de la interfaz	29
5.1	Prototipado de pantallas	31
5.2	Evaluación de la usabilidad	37
6.	Contexto tecnológico	39
6.1	Tecnologías.....	39
6.1.1	Spring Boot.....	39
6.1.2	Java	39
6.1.3	JavaScript.....	39
6.1.4	HTML y CSS.....	39
6.1.5	Thymeleaf	40
6.1.6	Bootstrap	40
6.1.7	MySQL.....	40

6.2	Herramientas	40
6.2.1	MySQL Workbench	40
6.2.2	SpringToolSuite4	41
6.2.3	Photoshop.....	41
7.	Implementación	43
7.1	Arquitectura software.....	43
7.2	Modelo	44
7.3	Vista	45
7.4	Controlador	47
7.5	Base de datos	48
8.	Aplicación final.....	51
8.1	Cambios a partir de la evaluación de usabilidad	51
8.2	Funcionalidad.....	53
9.	Conclusiones	57
9.1	Trabajo futuro.....	58
10.	Bibliografía	59
	ANEXO A.....	61
	ANEXO B.....	66



1. Introducción

Vivimos en una época en la que las tecnologías de la información están a la orden del día en la mayor parte del planeta, fundamentalmente gracias a Internet. Internet, así como las diferentes aplicaciones que surgen alrededor de esta red de redes, nos proporciona una inmensa cantidad de información y posibilidades.

Ante la llegada de la pandemia de COVID-19, estas tecnologías han adquirido aún más relevancia y se han convertido en ya casi imprescindibles para muchas personas. No solo por el hecho de entretener o por el poder de comunicación que tienen, sino también por hacer la vida del ciudadano más cómoda y segura.

Con la COVID-19 presente, todos los gobiernos han tomado medidas de prevención sanitarias basadas mayoritariamente en la reducción del contacto social. Un ejemplo de estas medidas sanitarias se puede ver reflejado en los restaurantes, donde, el formato físico de las cartas ha quedado prácticamente inexistente, dejando paso a cartas digitales disponibles a través de la web. Aprovechando esta circunstancia, nace la idea de desarrollar una aplicación web para ofrecer otras funcionalidades además de consultar la carta, como la realización de pedidos, la reserva de mesas, pago con el móvil, etc.

1.1 Motivación

En la actualidad se dispone de una cantidad innumerable de aplicaciones móviles orientadas a ser utilizadas por los usuarios en su vida diaria. Es bastante común que los usuarios de dispositivos móviles tengan instaladas aplicaciones que han utilizado una vez y ya no utilizan. Esto ha ocasionado que los usuarios sean cada vez más reticentes a instalarse aplicaciones de un solo uso en sus dispositivos móviles.

Las aplicaciones de pedidos de comida no son una excepción. Existen aplicaciones móviles creadas ad-hoc para un restaurante que los usuarios se tienen que instalar para realizar pedidos o reservar mesa. Sin embargo, estas aplicaciones no tienen un gran éxito entre usuarios. Aprovechando la situación de la digitalización de la hostelería, este trabajo final de grado trata de dar una alternativa a las aplicaciones móviles existentes, desarrollando una aplicación web para la gestión de pedidos de comida y reservas de mesa en restaurantes.

1.2 Objetivos

El objetivo principal de este proyecto es desarrollar una aplicación web para la gestión de pedidos de comidas y reservas de mesa en restaurantes. Además de este objetivo principal, se plantean los siguientes objetivos secundarios:

- Ofrecer variedad de métodos de pagos, considerando las plataformas más modernas de pago como bizum.
- Ofrecer la posibilidad de realizar el pago compartido de forma fácil agilizando la fracción de la cuenta entre comensales.
- Realizar un diseño *responsive*, que se adapte a cualquier tamaño de dispositivo.
- Realizar un diseño sencillo y de contenido claro, para que los usuarios tengan una navegación fácil y fluida.

1.3 Estructura

La memoria de este TFG se divide en 10 capítulos. Dentro de estos capítulos se encuentra un desglose por apartados sobre el tema a tratar. A continuación, se enumeran y se explica de forma breve su contenido:

1. **Introducción.** Este capítulo se centra en introducir el tema del proyecto, así como la motivación y los objetivos que se persiguen.
2. **Desarrollo centrado en el usuario (DCU).** En este capítulo se describe el enfoque DCU, así como sus fases y como se ha aplicado este enfoque en el proyecto.
3. **Estado del arte.** En este capítulo se analizan varias aplicaciones y servicios que se asemejan a la aplicación presentada.
4. **Análisis de necesidades de usuario.** Este capítulo describe el estudio realizado y se reúnen los datos obtenidos con el fin de identificar las necesidades de los usuarios.
5. **Diseño de la interfaz.** Este capítulo contiene los prototipos de la aplicación.
6. **Contexto tecnológico.** En este capítulo se presenta la descripción de las tecnologías y las distintas herramientas que se han utilizado para el desarrollo de este proyecto.
7. **Implementación.** En este capítulo se describe la implementación realizada siguiendo la arquitectura MVC, así como la base de datos diseñada e implementada en el desarrollo.
8. **Aplicación final.** En este capítulo se presenta la aplicación web final desarrollada mostrando los cambios realizados a partir de la evaluación de

usabilidad realizada a partir de los prototipos implementados, explicando además las diferentes funcionalidades que posee.

- 9. Conclusión.** Este último capítulo presenta las conclusiones del proyecto y también se incluye una serie de mejoras o adaptaciones como trabajos futuros para versiones futuras.

2. Diseño centrado en el usuario

Para el desarrollo de la interfaz gráfica de usuario que se plantea en este proyecto se ha seguido el enfoque de Diseño Centrado en el Usuario (DCU). Este enfoque tiene como principal objetivo situar al usuario como foco principal de las decisiones de diseño durante todo el proceso de desarrollo. Para ello, en este enfoque se precisa de la realización de estudios previos al desarrollo de la interfaz que nos permitan conocer las necesidades de los usuarios respecto a la aplicación.

2.1 Fases del proceso DCU

El proceso propuesto por este enfoque se divide en cuatro etapas: análisis, prototipado, implementación y evaluación.

- **Análisis:** Es la primera etapa donde se obtienen y se analizan las necesidades, requerimientos y objetivos de los usuarios finales. En esta etapa inicial se realiza una investigación cualitativa mediante entrevistas, cuestionarios, estudio de documentación o simplemente a través de la observación. Una vez recabada la información a través de estas técnicas se procede a la realización del modelado del usuario. Existen diversas técnicas para el modelado de usuario, una de ellas es la técnica *Personas*. Esta técnica consiste en la definición de un arquetipo de usuario en base a atributos comunes, creando así un personaje ficticio o arquetipo para el que se realiza el diseño del producto. En esta etapa también se describen las necesidades de los usuarios. Una de las técnicas para describir estas necesidades son los *Escenarios*. Esta técnica propone la construcción de pequeños relatos que describen una historia en la que se utiliza el sistema a desarrollar para satisfacer los objetivos de un usuario. Es decir, estos relatos cuentan como la persona alcanza sus objetivos utilizando la aplicación.
- **Prototipado:** Tras la recolecta de información de la primera etapa, en esta segunda etapa se pasa a crear diseños iniciales de la interfaz gráfica de la aplicación utilizando como base para la toma de decisiones la información obtenida en la fase anterior. Para llevar a cabo la realización de estos diseños iniciales se pueden utilizar diferentes técnicas de prototipado. Estas técnicas son una herramienta útil para permitir que el usuario participe activamente durante el desarrollo y evaluar así con el usuario el producto con el fin de que el resultado sea el esperado (evaluación temprana de la usabilidad). Existen diversas técnicas de prototipado que se usan en función de las características del prototipo que necesitamos. Las técnicas más destacables son:



bocetos, mockups, storyboards, vídeos. En esta etapa se tienen también en cuenta principios generales de usabilidad, como por ejemplo los 10 principios de usabilidad de Nielsen [1].

- **Implementación:** Esta etapa consiste en la implementación de la interfaz gráfica utilizando lenguajes de programación específicos para su implementación. De nuevo, en esta etapa, para llevar a cabo la toma de decisiones se utiliza la información que tenemos del usuario, así como los principios generales de usabilidad anteriormente mencionadas.
- **Evaluación:** La evaluación de la usabilidad se lleva a cabo después de las fases de prototipado e implementación realizando test de usuarios para evaluar el diseño de la aplicación. En esta fase se recolectan datos acerca de la satisfacción de los usuarios respecto a la aplicación. Esto permite verificar si la interfaz construida es usable. Este proceso se realiza mediante técnicas de evaluación de usabilidad tales como las técnicas de inspección (como la evaluación heurística), indagación (como la observación) o tests (como la medida de prestaciones).

2.2 Aplicación del enfoque DCU en el proyecto

Actualmente hay un gran número de aplicaciones tanto de escritorio como web para la realización de pedidos de comida o reservas de mesa en restaurantes. Esto lleva a que exista una gran competencia en este mercado de las aplicaciones. Como resultado, los usuarios cada vez son más exigentes con las aplicaciones, y ya no buscan una aplicación solo por su utilidad, sino que deciden también en función de criterios de usabilidad. Es por esto por lo que la usabilidad se convierte en un factor clave que marca la diferencia entre aplicaciones en este dominio.

En este proyecto se ha decidido dar un enfoque centrado en el usuario para crear un diseño de interfaz adaptado a las necesidades del usuario final, que ofrezca una buena experiencia de usuario. De esta manera, se persigue obtener una aplicación que pueda hacer frente a la gran competencia que existe en el mercado. Para ello, durante el desarrollo del proyecto, llevaremos a cabo los siguientes pasos:

1. **Análisis:** En primer lugar, se identifican las necesidades de los usuarios potenciales de este tipo de aplicaciones. Para ello, se ha realizado un cuestionario a un grupo de usuarios potenciales, con el objetivo de recoger información acerca de sus hábitos relacionados con las reservas de mesa, pedidos y pago en restaurantes. Con los datos obtenidos se construyó un arquetipo de usuario utilizando la técnica *Persona* y siguiendo con la descripción de los *Escenarios*. En este punto, también se ha realizado un análisis de mercado de aplicaciones similares para identificar convenciones.

2. **Prototipado:** En esta segunda etapa y con los datos obtenidos en el análisis se han realizado prototipos de interfaz para que estos sean validados de forma temprana con usuarios potenciales y permitan mejorar y ajustar la interfaz a las necesidades de estos.
3. **Evaluación temprana:** En este caso se ha decidido realizar una evaluación temprana de usabilidad sobre los prototipos creados en la segunda etapa. De esta manera, podemos ajustar más el diseño a las necesidades del usuario antes de empezar con la implementación.
4. **Implementación:** En esta cuarta etapa se ha desarrollado la codificación de la aplicación, trabajando con las tres capas, modelo, vista y controlador.

3. Estado del arte

La hostelería, sobre todo en el ámbito gastronómico, es un sector bastante tradicional en cuanto a la forma de operar se refiere. Actualmente en muchos locales de comida siguen tomando las comandas a través de papel y boli y posteriormente pinchando la hoja de la comanda en un corcho. Sin embargo, esta técnica manual está quedando atrás con la aparición de PDA's y tablets en estos locales, empezando así su digitalización. Siguiendo con esa digitalización y la modernización de los dispositivos móviles se empezó a ver un nuevo uso de estas tecnologías que consiste en poder pedir comida a través de una aplicación. Tras las nuevas medidas sanitarias que se han adoptado con la COVID-19 la hostelería ha seguido su proceso de digitalización, pasando de las cartas en papel a cartas digitales.

Un seguimiento del mercado nos ha permitido observar que algunos restaurantes ya están empezando a aplicar esta digitalización. Y es que tal y como se redacta en el informe del 'Anuario de la restauración organizada en España' realizado por KPMG [2], red global de firmas de servicios profesionales que ofrece servicios de auditoría y asesoramiento en 156 países y es una de las Big4 [3] (término con el que se refiere a las 4 firmas más importantes del mundo en este sector):

[...] “la restauración no es ajena al proceso de transformación digital que están acometiendo las empresas en muchos sectores de actividad. Así, la digitalización es la prioridad estratégica para el 9% de las empresas, al igual que la mejora y transformación de procesos internos (11%), aspecto crucial para adaptarse a un entorno cada vez más competitivo y con nuevas herramientas tecnológicas que están posibilitando el desarrollo de nuevos modelos de negocio”. [4]

En Valencia capital algunos restaurantes disponen ya de aplicaciones con objetivos similares a los que se persiguen en este proyecto. Algunos de ellos son: *Alma Libre*¹ o *La Fabrica de Hielo*². Las aplicaciones que ofrecen estos restaurantes consiguen algunos de los objetivos que en este proyecto se persiguen, como son los pedidos en mesa a través de la aplicación, la lectura de la carta y a la par la selección de productos a consumir. Sin embargo, adolecen de algunas limitaciones tal y como se explica a continuación (ver sección 3.1)

Además, existen empresas que se dedican expresamente a desarrollar este tipo de soluciones demandadas por los restaurantes y que se están ofreciendo a través de un

¹ <https://www.almalibreacaihouse.com/>

² <https://www.lafabricadehielo.net/carta/>

servicio REST. En este contexto encontramos *Pidopago*³ o *Portalrest*⁴ los cuales se detallan más adelante (ver sección 3.2).

3.1 Apps de restaurantes

A continuación, se presentan dos aplicaciones móviles ofertadas por dos restaurantes para gestionar los pedidos de sus clientes, así como también ofrecer la posibilidad de programar pedidos, reservar mesa entre otras funcionalidades que se describen a continuación.

Es interesante hacer un estudio sobre las aplicaciones que están en funcionamiento en el mercado actualmente ya que esto puede dar una visión global sobre que es lo que los usuarios necesitan y cómo lo quieren.

3.1.1 Alma Libre

Alma libre es una franquicia de comida vegana la cual actualmente dispone de 3 establecimientos abiertos en España, específicamente en Valencia, Barcelona y Madrid. Esta franquicia se distingue por el uso de açai, una baya pequeña muy rica en nutrientes y antioxidantes procedente de la Amazonia. Esta franquicia dispone de un servicio de comandas a través de una aplicación web móvil. En la Figura 1 se observa una sección de la carta de la aplicación que ha sido desarrollada para este restaurante. En esta imagen podemos ver que las distintas secciones de la carta están disponibles en la parte superior (Açaí, Desayuno/merienda, Para picar...). Estas secciones de la aplicación se disponen en forma de scroll horizontal, aunque no es muy intuitivo ya que no aparece ningún elemento que indique que este menú sea móvil horizontalmente. Según un estudio realizado Katie Sherwin del grupo Nielsen Norman⁵, determinó que en los dispositivos móviles el desplazamiento horizontal es común. Sin embargo, este desplazamiento horizontal no es común en aplicaciones de escritorio por lo que los usuarios de dispositivos móviles a menudo no saben que existe más contenido desplazando horizontalmente. En la parte central del interfaz se encuentra una lista de productos los cuales se pueden añadir al carrito con el botón '+'. Sin embargo, no existe la posibilidad de eliminar un producto del carrito sin entrar a este. Así mismo cada producto posee una ficha con más detalle al que se accede pulsando sobre el propio producto.

³ <https://www.pidopago.com>

⁴ <https://web.portalrest.com/>

⁵ <https://www.nngroup.com/articles/horizontal-scrolling/>

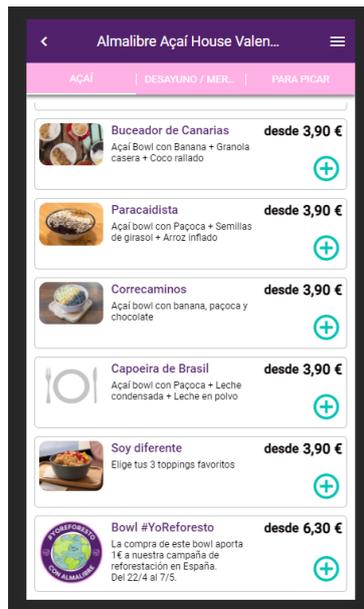


Figura 1. Carta almalibre.com

3.1.2 La Fábrica de Hielo

La Fábrica de Hielo es un espacio creativo independiente. Este local se caracteriza por su eclecticismo, aunando diferentes áreas y disciplinas artísticas. La Figura 2 muestra una imagen de su aplicación web. La aplicación completa solo está disponible al escanear un código QR que se da en el mismo restaurante. La versión disponible desde la página web solo tiene disponible la lectura de la carta. En la versión completa de esta aplicación web la carta se encuentra dividida en dos secciones importantes, la bebida y la comida. Dentro de cada sección se encuentra una lista de subsecciones con distintos productos. La aplicación permite añadir y eliminar productos al carrito desde la propia carta lo cual hace más cómodo e intuitivo realizar un pedido.



Figura 2. Carta lafabricadehielo.net

3.2 Servicios a restaurantes

En la web se pueden encontrar multitud de empresas que ofrecen estos servicios a otras empresas, ya sean en el ámbito de la restauración, en comercio o en otro tipo de establecimientos. Es interesante hacer un estudio sobre los servicios que ofrecen las distintas empresas en este sector. Con ello, se puede obtener una idea globalizada de los servicios que los clientes necesitan y buscar algunas carencias que estos servicios todavía no ofrecen.

A continuación, se presentan dos servicios ofertados a restaurantes para la gestión del restaurante y ofreciendo distintas funcionalidades a quienes contratan su servicio.

3.2.1 Pidopago

Pidopago es un servicio que ofrece la adaptación del sitio web del restaurante que lo solicite ofreciéndole funcionalidades como carta online, pedidos sin contacto y entrega o delivery.

Aunque el servicio que ofrece Pidopago en la sección de pago sí ofrece el pago mediante pago alternativos como Paypal, no ofrece la opción de fraccionar el pago que es uno de los objetivos de este desarrollo.

En la Figura 3 se encuentra una captura de la demo que ofrece Pidopago. En ella se puede ver que esta aplicación posee un buscador donde se puede escribir el nombre de

los productos y filtrarlos. La demo ofrece una carta dividida por categorías. En cada categoría se encuentra un listado de productos que pertenecen a ésta. Del mismo modo que se ha comentado anteriormente en el apartado 3.1.1. esta demo ofrece la posibilidad de añadir un producto al carrito, pero no ofrece la posibilidad de eliminar sin acceder al carrito. En esta aplicación se puede observar que el total del pedido está siempre visible en la parte inferior de la pantalla con un enlace que muestra el pedido que se está realizando.

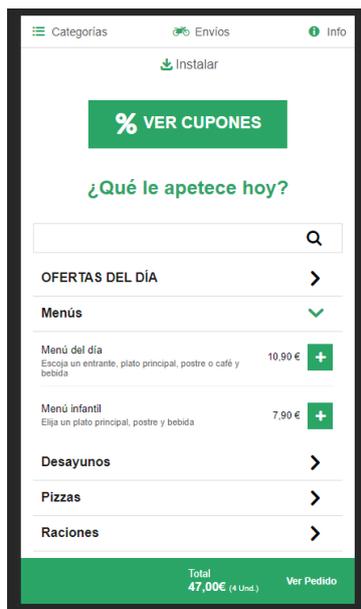


Figura 3. Demo Pidopago

3.2.2 PortalRest

PortalRest es un aplicativo de la empresa ICG Software cuya plataforma web se ocupa de mejorar la gestión del comercio, creando una web móvil para el encargo de pedidos online, reservas de mesa, etc. Este aplicativo ofrece los siguientes productos a sus clientes:

- *Servicio de pedidos online.* Con este servicio ofrecen la posibilidad de integrar sus portales web con las redes sociales habilitando los pedidos a través de estas.
- *Un servicio de reservas de mesa.* En este servicio PortalRest ofrece a sus clientes la posibilidad de añadir la opción de realizar un pago de un importe cuando se realiza una reserva, para evitar reservas perdidas. En este servicio de reservas de mesa se permite que el usuario que realice la reserva tenga la opción de escribir preferencias de ubicación o alguna nota como alguna intolerancia o alergia. Este servicio también cuenta con la confirmación de la reserva vía email con lo que se trata de evitar los *no-shows*, es decir, que los clientes no se presenten finalmente en el restaurante en la fecha y hora

Desarrollo de una aplicación web móvil para la gestión de comandas en restaurantes

reservada. Finalmente, el servicio ofrece la posibilidad de crear relaciones entre el cliente, reservas, ventas y con ello identificar platos favoritos del usuario, gasto medio del usuario o la frecuencia con la que éste acude al restaurante.

En particular, PortalRest cuya web se muestra en la Figura 4, ha sido el encargado de llevar a cabo el proyecto de Alma Libre.



Figura 4. Web web.portalrest.com

3.3 Limitaciones de las aplicaciones

Para finalizar esta sección se analizan los distintos objetivos que no alcanzan estas aplicaciones o las limitaciones que estas presentan respecto a los objetivos que se plantean en la idea de este proyecto.

Las aplicaciones de los restaurantes que se han analizado anteriormente presentan algunos objetivos que también se persiguen en este proyecto, como son la lectura de la carta a través de la web, la realización de pedidos a la mesa a través de la aplicación web móvil, entre otras. Sin embargo, las dos aplicaciones analizadas carecen de algunos elementos importantes como son:

- Solo es posible el pago mediante tarjeta de crédito y no ofrecen alternativas de pago a la forma tradicional.
- En la aplicación de *La Fábrica de Hielo* no se puede añadir en el carrito dos o más elementos de un mismo producto con un solo clic, lo que puede dificultar la experiencia de usuario.
- En la aplicación de *Alma Libre* no se puede eliminar un elemento ya añadido a no ser que se acceda a la información del producto o al carrito. Esta limitación puede entorpecer la experiencia del usuario
- Ninguna de las aplicaciones revisadas permite dividir la cuenta por comensal.

4. Análisis de Necesidades de Usuario

En este apartado se describe la fase de análisis de necesidades de usuario que hemos llevado a cabo en el proyecto. Como se ha comentado anteriormente, esta es la primera fase del enfoque DCU. En esta fase se recopilan datos que permiten identificar las necesidades de los usuarios. A partir de estas necesidades, se realiza un diseño de la interfaz que las satisfaga.

Esta fase se divide en varios pasos:

- 1) *Investigación cualitativa*, donde a través de distintas técnicas se recolectan datos sobre el comportamiento y necesidades de los usuarios.
- 2) *Definición de Personas*, en este paso se aplicará la técnica Persona para obtener la descripción de un modelo o arquetipo de usuario.
- 3) *Descripción de Escenarios*, en este paso se recrean historias dónde la persona creada alcanza los objetivos identificados con la aplicación a desarrollar.

A continuación, se describe cada uno de estos pasos.

4.1 Investigación cualitativa

La investigación cualitativa es la primera etapa en el análisis de necesidades. Esta investigación consiste en la recolección de datos acerca de hábitos de los usuarios que nos ayuden a explicar y comprender las razones del comportamiento de nuestros usuarios respecto a tareas relacionadas con las de la aplicación a desarrollar.

En el presente proyecto, se ha utilizado la técnica de cuestionario para la recolección de estos datos. Esta técnica consiste en la realización de cuestionarios a un número alto de usuarios potenciales. En nuestro caso, se persigue obtener información sobre los usuarios acerca de:

- Su perfil demográfico.
- Sus habilidades tecnológicas.
- Sus hábitos en las salidas a comer fuera de casa.
- Sus hábitos en los pedidos de comida.

Desarrollo de una aplicación web móvil para la gestión de comandas en restaurantes

Para obtener esta información, se ha elaborado un cuestionario de 13 preguntas a través de la herramienta Google Forms y se ha recabado la respuesta de 60 voluntarios (ver ANEXO A).

Con los datos obtenidos, se ha realizado un análisis y se ha obtenido la información que se requería. Las conclusiones son las siguientes:

- Más del 50% de los encuestados tienen una edad entre los 21 y los 29 años, con lo que se puede entender que esta aplicación mayoritariamente estará enfocada a gente joven y con habilidad con las tecnologías.
- Los usuarios salen regularmente a comer fuera de casa, en particular un 35% sale 4 veces o más al mes y un 43,3% entre 2 y 3 veces al mes.
- Mayoritariamente los usuarios no realizan pedidos a domicilio, en particular un 18,3% nunca lo hace y un 30% casi nunca.
- Mayoritariamente los usuarios sí consultan opiniones, un 71,7% de los encuestados a través de Google y un 51,7% en Tripadvisor. Sin embargo, muy pocos usuarios reseñan los lugares que frecuentan obteniendo que un tercio de ellos nunca lo hace y un 45% casi nunca.
- Más del 75% no consulta la tabla de alérgenos. Así mismo, al 70% de los usuarios les resulta más fácil escoger mediante productos con texto e imagen.
- El 95% de los usuarios divide la cuenta total en partes iguales.

Las respuestas obtenidas en los cuestionarios se encuentran en el Anexo A de esta memoria.

Con los datos obtenidos se procede a la realización de una persona primaria y un conjunto de escenarios donde se describe cómo la persona utiliza la aplicación.

4.2 Definición de Persona

A partir de los datos obtenidos en la sección anterior, en esta sección definiremos la *Persona Primaria* que tomaremos como base para tomar todas las decisiones de diseño durante la construcción de la interfaz de la aplicación.



- Paula Sanchis Valor.
- 26 años.
- Los fines de semana suele salir a comer a algún restaurante.
- Para cerciorarse que no falla en su elección, Paula se lee muchas reseñas que han puesto otros usuarios en Google y en Tripadvisor.
- Tiene un perfecto manejo de los dispositivos electrónicos.
- Siempre está al tanto de lo que publica su perfil gastronómico preferido en redes sociales.
- Le gusta probar nuevos sitios para comer.

Objetivos de Paula respecto a la aplicación:

1. Programar el pedido de su comida.
2. Fraccionar el pago cuando sale a un restaurante con sus amigas.
3. Leer reseñas del local donde va a comer.
4. Utilizar bizum para pagar.
5. Ver imágenes en la carta de cómo son los platos.
6. Reservar mesa en el restaurante a través de la aplicación.

4.3 Escenarios

A partir de los datos recogidos en los cuestionarios, y una vez definida la *Persona Primaria*, se van a describir escenarios que detallen cómo se va a usar la aplicación para cubrir los 6 objetivos identificados en la *Persona Primaria*. Estos escenarios contienen el contexto de uso de la aplicación, lo cual se tendrá en cuenta a la hora de tomar decisiones durante la construcción de la interfaz.

A continuación, se describen 6 escenarios que permiten cubrir los 6 objetivos identificados:

Escenario 1. Programar pedido de comida

Paula tiene guardia el martes por la noche en el hospital donde trabaja. Como no se ha preparado nada decide programar un pedido para la hora de la cena. Paula entra en la web del restaurante con su teléfono móvil y pulsa sobre el botón 'Programar Pedido'. Aquí selecciona el día y la hora en la que pasará a recoger su pedido. Al pulsar 'Programar' le redirige a la carta y realiza el pedido de su cena.

Escenario 2. Consultar apartado de opiniones.

El otro día una amiga le comentó a Paula que había estado en un restaurante nuevo que le gustó mucho. Paula quiere proponer una salida con amigos para el sábado noche y está pensando que el restaurante que le han sugerido puede ser una buena opción. Así que entra en la aplicación del restaurante y pulsa en el apartado '*Opiniones*'. Allí ve una lista de opiniones de distintos usuarios y comprueba que la gran mayoría de las opiniones tienen buena puntuación por lo que decide ir a probar esta experiencia en persona.

Escenario 3. Fraccionar el pago cuando sale con amigas

Paula está en el restaurante con sus amigas, han quedado para comer. Son 6 y como siempre cuando acaba la comida van a pagar a partes iguales. Paula abre la aplicación del restaurante y pulsa sobre '*Pagar*'. Aquí se encuentra con varias opciones de pago y Paula selecciona el método de *pago fraccionado* e introduce el número de personas con el que quiere dividir la cuenta. En este punto la aplicación web muestra en la pantalla las instrucciones que debe seguir, como dónde y cómo realizar el pago fraccionado.

Escenario 4. Pagar con bizum

Paula está de comida con sus amigas en el restaurante y llega el momento de pagar. Paula se da cuenta que no ha cogido la cartera por lo que no puede pagar con tarjeta ni efectivo. Así que pulsa sobre '*Pagar*' y elige la opción de pago por 'bizum'. En este punto la aplicación web muestra en la pantalla las instrucciones que debe seguir para realizar el pago mediante este método (el número de teléfono para realizar el bizum y el concepto a indicar).

Escenario 5. Ver imágenes de los productos

Paula está en el restaurante y está ojeando la carta. Hay algunas cosas de ella que no sabe lo que son, pero gracias a la imagen de cada producto se hace una idea. Para observar el plato con más detalle pulsa sobre la imagen del producto con lo que se abre la ficha del producto con la imagen ampliada y los ingredientes que contiene ese plato.

Escenario 6. Realizar una reserva desde la app

Paula acaba de quedar con sus amigas para ir a cenar al restaurante el sábado 21 de agosto. Como es una fecha de temporada alta no se quieren arriesgar a no tener mesa para ese día. Por eso, Paula entra en la web del restaurante y pulsa sobre '*Reservar Mesa*'. En este punto le aparece un formulario a rellenar con la fecha, la hora y el número de personas que serán. Para poder finalizar la reserva Paula debe insertar también un número de teléfono de contacto y un email. Finalmente, pulsa sobre '*Reservar*' y ya tiene la reserva hecha.

La descripción de estos escenarios nos sirve como especificación de requisitos a partir de los cuales se inicia el diseño de la interfaz gráfica de la aplicación.

5. Diseño de la interfaz

En este apartado se presenta la fase de diseño que se ha llevado a cabo en el proyecto. En esta fase se construye un prototipo de la aplicación que ofrece una visión con detalle sobre la interfaz final de la aplicación. Con esta visión se obtiene una idea que refleja cómo será el diseño de la interfaz con el objetivo de facilitar la implementación de esta interfaz en la siguiente fase.

Para la realización de prototipos de una interfaz gráfica de usuario, se pueden utilizar diferentes técnicas de prototipado, entre las que se destacan los bocetos, mockups, storyboards, vídeos o prototipos funcionales. En este proyecto, se ha optado por la utilización de mockups. Para la creación de mockups existen muchas herramientas que ayudan a realizarlos, como Pencil Project, Photoshop, JustinMind, proto.io entre otras. En este caso se han realizado los mockups utilizando la aplicación Photoshop, ya que esta herramienta de edición de imagen nos permite adaptar nuestras ideas sin ningún tipo de restricción.

Este prototipo de aplicación web móvil se ha maquetado teniendo en cuenta las necesidades de los usuarios que se han identificado en la fase anterior, y que se han descrito a través de la Persona (ver sección 4.2) y los Escenarios (ver sección 4.3). Además, se han tenido en cuenta para ir tomando las decisiones de diseño los 10 principios de usabilidad de Nielsen que se describen a continuación:

- 1. Visibilidad del estado del sistema.** El sistema debe siempre mantener informado al usuario de lo que está ocurriendo. Este principio se ve resuelto en esta aplicación web ya que en todas las pantallas de la aplicación se informa al usuario mediante un título en qué sección se encuentra. También se encuentra este principio cuando por ejemplo el usuario realiza la acción de reservar una mesa, donde el sistema le informa de que su petición ha sido realizada.
- 2. Relación entre el mundo real y el sistema.** La aplicación web tiene que utilizar el lenguaje del usuario y palabras o conceptos que resulten familiares. Cuando el usuario desea añadir o quitar algún producto del carrito estos botones se encuentran con su respectivo símbolo '+' y '-'.
- 3. Control del usuario.** El usuario debe tener la posibilidad de corregir un error en caso de equivocación en cualquier momento. Por ello en todas las pantallas de la aplicación se encuentra un botón con una 'x' que permite al usuario cerrar la vista actual.

4. **Consistencia y estándares.** Es importante establecer y mantener la misma consistencia de los elementos, así como seguir los convenios y estándares establecidos para que el usuario no se confunda. En esta aplicación web hay diferentes botones, pero todos siguen el mismo estilo para no confundir al usuario.
5. **Prevención de errores.** Ayuda al usuario a que no caiga en un error. Por eso se validan los formularios antes de ser enviados. Por ejemplo, en el registro se pide dos veces el campo contraseña para prevenir errores cometidos por el usuario.
6. **Reconocer mejor que recordar.** Se debe hacer visibles acciones y opciones para que el usuario no tenga que recordar información entre secciones. En esta aplicación se encuentra por ejemplo en el icono de un carrito de compra, ya estandarizado como icono para acceder al carrito del sitio web.
7. **Flexibilidad y eficiencia de uso.** Los aceleradores o atajos de teclado pueden hacer más rápida la interacción para usuarios expertos. Esto se puede encontrar en el rellenado de distintos formularios de la aplicación ya que se muestran opciones de auto rellenado ya usadas por el usuario.
8. **Diseño estético y minimalista.** La página web debe ser lo más simple posible y no contener información innecesaria ya que esta información extra compite con la información relevante y disminuye la visibilidad. El diseño de esta aplicación web es simple y contiene los elementos necesarios para el uso de los usuarios.
9. **Ayudar a los usuarios a reconocer, diagnosticar y solucionar los errores.** Los mensajes de error se deben mostrar en un lenguaje claro y simple indicando el problema. Este principio se refleja en los formularios cuando un usuario introduce algún campo de forma errónea.
10. **Ayuda y documentación.** Algunas ocasiones se precisa proporcionar documentación de ayuda al usuario sobre la aplicación. En este proyecto este aspecto no ha sido contemplado ya que se trata de una aplicación simple que el usuario puede usar sin ayuda.

5.1 Prototipado de pantallas

Teniendo en cuenta las necesidades de los usuarios, así como las guías de usabilidad de Nielsen, a continuación, se describen las diferentes pantallas que se han prototipado para la aplicación.

En la Figura 5 se muestra el menú de inicio de la aplicación. En esta pantalla se muestran los enlaces, en modo de botones, a las distintas secciones de la aplicación. Inicialmente cuando se accede a la aplicación web todos los usuarios tienen acceso a cualquier elemento del menú inicial. Por lo que no es necesario un registro previo para utilizar esta aplicación web.

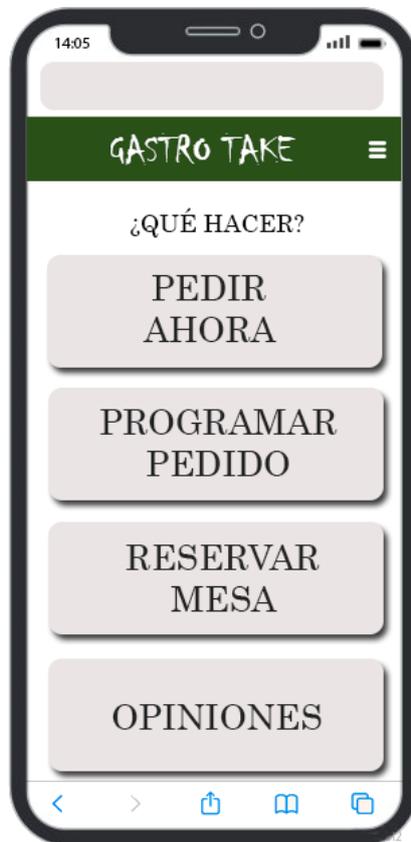


Figura 5. Prototipo página inicial

Desarrollo de una aplicación web móvil para la gestión de comandas en restaurantes

Al clicar sobre 'Pedir Ahora' el usuario puede empezar a realizar su pedido. Cuando se accede a este apartado se muestra la carta dividida por secciones como se observa en la Figura 6. Al desplegar estas secciones el usuario puede ver los productos que ofrece el establecimiento relacionados con el apartado que ha desplegado tal y como se muestra en la Figura 7.

En esta vista el usuario puede estar simplemente en modo lectura o puede ir realizando su pedido. Esta y las demás secciones se pueden cerrar volviendo al menú pulsando en la X de la parte superior izquierda.



Figura 6. Prototipo carta



Figura 7. Prototipo carta abierta

Para ver información con más detalle de los productos, el usuario puede pulsar sobre la imagen. Cuando pulsa sobre ella, se abre la vista mostrada en la Figura 8 donde aparece una imagen ampliada del producto junto con los ingredientes y los alérgenos y la posibilidad de añadir o quitar este producto al pedido.

Una vez el usuario tenga el pedido completo o desee revisar su pedido puede dirigirse al carrito pulsando sobre el icono del carrito situado en la parte superior derecha del menú. Una vez en esta vista, Figura 9, el usuario puede retroceder pulsando sobre la 'X' para seguir completando su pedido o realizar el pedido pulsando sobre pagar.

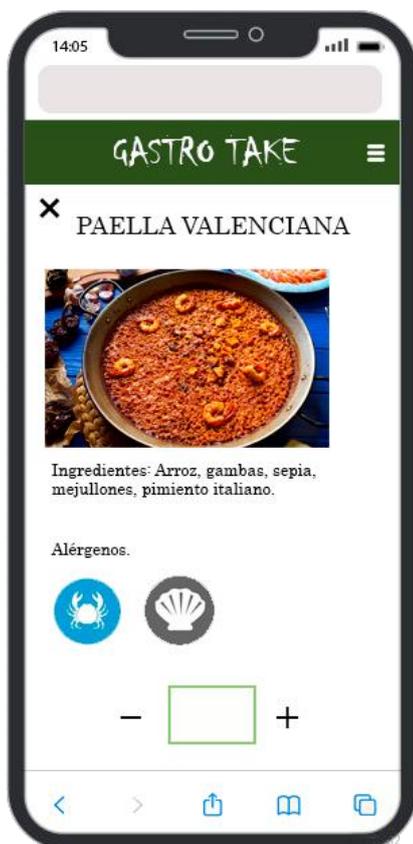


Figura 8. Prototipo de ficha de producto



Figura 9. Prototipo carrito

Cuando el usuario pulsa sobre el botón 'PAGAR' la aplicación le redirigirá a la ventana de pago, Figura 10, si anteriormente se había logueado, sino se le redirigirá al formulario de Login, Figura 14. Desde la ventana 'PAGAR' se ofrece al usuario la opción de realizar el pago de diferente forma. Si el usuario elige la forma de pago único, la aplicación mostrará el formulario de pago con tarjeta. En cambio, si elige la forma de pago fraccionada se mostrará un cuadro para insertar el número de personas con el que se dividirá la cuenta. Seguidamente se mostrarán las instrucciones para realizar el pago.

Por otra parte, si desde el menú principal el usuario accede a "Programar Pedido" la aplicación le redirigirá a un formulario donde rellenará los campos para realizar su programación tal y como se muestra en la Figura 11. Una vez seleccionada la programación de día y hora se redirige al usuario al apartado de la carta.

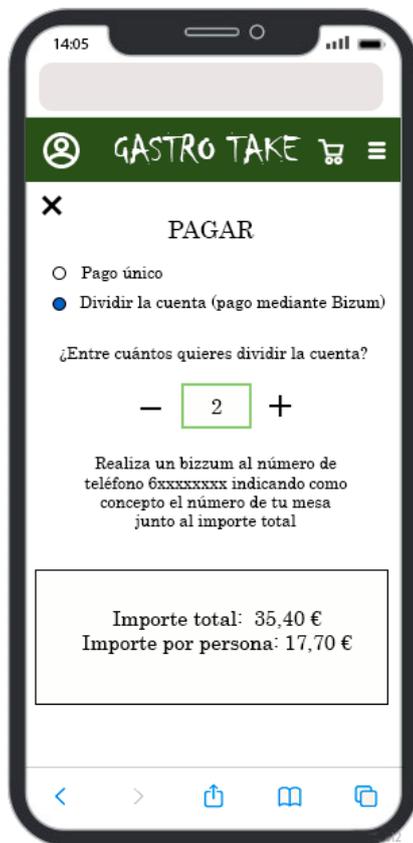


Figura 10. Prototipo pagar



Figura 11. Prototipo programar pedido

Siguiendo con los apartados del menú principal, el usuario se encuentra con la opción de reservar mesa, que al pulsar sobre este botón se redirige al usuario al formulario de reservas como se muestra en la Figura 12. En este formulario el usuario seleccionará el día que quiere hacer la reserva desde el calendario que se muestra. También deberá seleccionar la hora a la cual quiere hacer la reserva, así como indicar el número de personas. Para completar la reserva el usuario deberá rellenar los campos de teléfono y correo electrónico por si el restaurante tuviera que ponerse en contacto con el cliente o para realizar la confirmación.

En la última sección del menú principal se encuentra el botón de 'OPINIONES' desde el cual se mostrará al usuario las diferentes opiniones de otros usuarios tal y como se ve en el prototipo de la Figura 13.



Figura 12. Prototipo reservar mesa



Figura 13. Prototipo opciones

Desarrollo de una aplicación web móvil para la gestión de comandas en restaurantes

También se presentan los prototipos para el inicio de sesión y para el registro de un nuevo usuario, Figuras 14 y 15 respectivamente. En estas vistas se muestra un formulario donde tendrán que rellenar sus datos. Para logearse utilizarán el email y la contraseña que han utilizado en el registro. Para evitar que el usuario introduzca errores, para registrarse se requiere, a modo de comprobación repetir la contraseña.



14:05

GASTRO TAKE

×

LOGIN

Email

Contraseña

Recuérdame [Recuperar contraseña](#)

ENTRAR

Iniciar con Google

< > 📄 📖 📄

Figura 14. Prototipo formulario login



14:05

GASTRO TAKE

×

REGÍSTRATE

Nombre

Apellidos

Email

Teléfono

Contraseña

Repita la contraseña

Aceptar términos y condiciones

CREAR

< > 📄 📖 📄

Figura 15. Prototipo formulario registro

5.2 Evaluación de la usabilidad

El DCU aboga por la evaluación de la usabilidad temprana. Así pues, una vez realizados los prototipos de la aplicación, se ha realizado un test con usuarios que ha permitido recoger información sobre la percepción de uso de la aplicación. Se ha elegido a 4 personas que cumplen el perfil de usuarios potenciales. Estas personas son:

- Mujer de 23 años que va 4 veces al mes a restaurantes.
- Mujer de 24 años que va 3 veces al mes a restaurantes.
- Mujer de 23 años que pide 3 veces al mes la comida a través de una app.
- Hombre de 25 años que pide 1 vez al mes la comida a través de una app.

A las 4 personas se les citó para una sesión de 5 minutos en la que tenían que realizar las siguientes tareas:

1. Logearse/Registrarse en la aplicación web.
2. Reservar una mesa para el sábado 25 de junio a las 21:00
3. Añadir un producto a la comanda.
4. Realizar un pago fraccionado por Bizum.

Al acabar las tareas se les pidió que respondieran un cuestionario con preguntas acerca de cómo les había resultado el uso de la aplicación dónde los encuestados tenían que elegir en un rango de 'Nunca' a 'Siempre' (cuestionario disponible en el Anexo B). Los datos obtenidos a partir de los cuestionarios son los siguientes:

- Para los 4 usuarios el menú en forma de hamburguesa⁶ no les ha resultado útil, 2 de ellos estaban en desacuerdo y 2 en total desacuerdo.
- Para 1 de los usuarios el diseño no le ha parecido un diseño atractivo respondiendo a esta pregunta con "en desacuerdo". Por otro lado, los 3 restantes tampoco les ha convencido del todo el diseño y han marcado la respuesta como "Neutral".
- Para el 100% de los usuarios encuestado la aplicación no les ha parecido innecesariamente compleja.
- El 100% de los usuarios se han sentido seguros al utilizar la aplicación.
- En la tarea que se proponía en el cuestionario para logearse o registrarse 1 usuario ha puntuado 4 la dificultad para realizar esta acción y otro con un 5. En

⁶ Forma gráfica de representar el menú siguiendo una disposición horizontal de bloques paralelos.



cambio, los dos restantes han puntuado esta tarea como algo más compleja siendo sus calificaciones de 7 y 8 respectivamente.

- En la tarea que se proponía en el cuestionario para realizar una reserva todos los usuarios han puntuado esta tarea con una calificación de baja dificultad para realizarla, siendo 3 de ellos puntuación con 2 y uno de ellos con 4.
- En la tarea que se proponía en el cuestionario para añadir un producto a la comanda todos los usuarios han puntuado con una calificación de baja dificultad para realizarla, siendo 1 y 2 para tres de los usuarios y un usuario respectivamente.
- En la tarea que se proponía en el cuestionario para realizar un pago fraccionado por Bizum la puntuación de dificultad de los usuarios ha sido baja, siendo 1, 2, 3 y 5.
- Para 3 de los 4 usuarios les gustaría usar esta aplicación y uno de ellos tal vez.
- Finalmente, se ha dejado una pregunta de respuesta abierta para que los encuestados pudieran hacer algún tipo de comentario sobre la aplicación. Dos de ellos han comentado que el diseño de la página principal no es muy agradable. Por otra parte, uno de los encuestados ha recalado que el formato para la programación del pedido no es muy adecuado. Ya que de la forma que se había propuesto no había vuelta atrás para no programar el pedido. Entre los comentarios también se encuentra que algunos de los encuestados no están de acuerdo con la tipografía utilizada.

Con los datos obtenidos en el cuestionario se han realizado algunos cambios en el diseño, que se muestran en la aplicación final (ver sección 8).

6. Contexto tecnológico

En este apartado se describen las distintas tecnologías utilizadas para el desarrollo de esta aplicación, así como las herramientas utilizadas.

6.1 Tecnologías

6.1.1 Spring Boot

Spring Boot [5] es un sub-proyecto de Spring Framework que busca facilitarnos la creación de proyectos con Spring Framework eliminando la necesidad de crear largos archivos de configuración XML.

6.1.2 Java

Java [6] es un lenguaje de programación orientado a objetos. La programación orientada a objetos ofrece un gran control sobre el código y una mejor organización. Java es un lenguaje muy flexible ya que está preparado para la reutilización de código, es decir, facilita a un desarrollador actualizar un código antiguo sin muchas complicaciones. Otra gran característica de este lenguaje es que a diferencia de programas que requieren versiones específicas según el sistema operativo dónde se lance, las aplicaciones desarrolladas con Java funcionan en cualquier entorno.

6.1.3 JavaScript

JavaScript (JS) [7] es un lenguaje de programación interpretado. Se define como un lenguaje orientado a objetos, basado en prototipos, imperativo y dinámico.

Este lenguaje se utiliza mayoritariamente en la parte del cliente, permitiendo mejoras de interfaz de usuario y páginas dinámicas, pero también se utiliza en el servidor.

Los programas que son escritos con JavaScript no precisan de un compilador ya que los navegadores pueden interpretar el código sin necesidad de intermediarios.

6.1.4 HTML y CSS

HTML [8] son las siglas conocidas de su nombre en inglés HyperText Markup Language. HTML es un lenguaje de programación utilizado en la realización de páginas web y es un estándar de codificación y estructuración mantenidas por World Web Consortium (W3C), que es una organización que se dedica a los estándares de las tecnologías para el desarrollo web.



El código HTML opera en base a la diferenciación mediante etiquetas de los atributos que lo conforman y de la ubicación de los elementos que conforman la página web.

Para otorgar diseño y estilo a los elementos codificados en el HTML se emplean hojas de estilo en cascada, conocidas como CSS. Las posibilidades de diseño que nos provee CSS son casi infinitas. Existen frameworks como Bootstrap (ver sección 6.1.6) que nos proporcionan unos estilos predefinidos haciendo que la aplicación web sea adaptable/responsive ante cualquier dispositivo.

6.1.5 Thymeleaf

Thymeleaf [9] es un motor de plantillas para aplicaciones web desarrolladas con Java. Esta tecnología es bastante similar a la tecnología JSP. Thymeleaf generalmente se utiliza con Spring Boot para generar vistas con código HTML para aplicaciones web.

6.1.6 Bootstrap

Bootstrap [10] es un framework CSS que se utiliza para el desarrollo de aplicaciones web y móvil. Este framework combina CSS y JavaScript para dar formato a los elementos que se encuentran en una página HTML. El principal objetivo de este framework no es solo ofrecer un diseño agradable sino permitir la realización de sitios web adaptable/responsive para dispositivos móviles.

6.1.7 MySQL

MySQL [11] es un sistema de gestión de bases de datos de código abierto para bases de datos relacionales. Una base de datos relacional es una colección de datos almacenados y organizados en tablas relacionadas entre si.

MySQL es el sistema de base de datos más utilizado en desarrollo web ya que ofrece una gran simplicidad para realizar cambios. Este gestor de base de datos permite el almacenamiento y la gestión de grandes cantidades de datos de una forma fácil y rápida.

6.2 Herramientas

6.2.1 MySQL Workbench

MySQL Workbench [12] es una herramienta gráfica que integra desarrollo de software, administración y diseño de bases de datos, gestión y mantenimiento para trabajar con servidores y bases de datos MySQL.

Esta herramienta ofrece múltiples funcionalidades entre las que se destacan:

- Conexión y gestión de las instancias de base de datos.
- Exploración de esquema de objetos.
- Conjuntos de resultados múltiples, editables
- Colección de fragmentos SQL que ayudan al desarrollo de sentencias SQL más complejas
- Diagrama entidad relación
- Iniciar y detener instancias de base de datos
- Configurar instancias

6.2.2 SpringToolSuite4

SpringToolSuite [13] es un IDE (Entorno de Desarrollo Integrado) para crear aplicaciones con Spring. Este IDE está construido sobre el entorno de desarrollo de Eclipse añadiendo algunos complementos para el fácil desarrollo de aplicaciones con Spring.

6.2.3 Photoshop

Photoshop [14] es una potente herramienta de edición de imágenes. Esta herramienta de Adobe permite crear, editar, manipular, retocar cualquier imagen. Por este gran potencial de edición esta aplicación ha sido elegida para la creación de los mockups de este proyecto.



7. Implementación

En esta sección se describe cómo se ha implementado en la aplicación. La arquitectura de la aplicación sigue el patrón de diseño Spring MVC. Por lo que, en el primer apartado de esta sección, se describe este patrón, y en las siguientes secciones se presentan los componentes principales en los que se divide la aplicación siguiendo este patrón.

7.1 Arquitectura software

En esta sección se detallarán los diferentes componentes que conforman la aplicación desarrollada en este TFG, así como la relación que existe entre ellos. Para el desarrollo de la aplicación se ha utilizado el patrón Modelo, Vista, Controlador más conocido por sus siglas MVC [15]. MVC es un patrón arquitectónico software que separa la capa lógica de la aplicación, lo que se denomina modelo, de la representación, en este caso la vista. Para la interacción entre estas dos capas se precisa del controlador, el cual se encarga de las solicitudes que llegan desde la vista para que esta haga las operaciones correspondientes y posteriormente el controlador devuelve a la vista el resultado para que este lo represente al usuario.

En la capa *Modelo* se encuentra la representación de la información con la que el sistema opera. Esta capa gestiona los accesos a la información, consultas y actualizaciones. Aquí también se encuentra la lógica de negocio que se encarga del control de acceso.

La capa *Vista* es la encargada de representar las solicitudes de los usuarios y los datos del modelo de una forma adecuada para la interacción del usuario. En esta capa se encuentra la interfaz de usuario.

En la tercera capa se encuentra el *Controlador*. Este componente actúa como intermediario entre el sistema y el usuario. Este controlador se encarga de procesar y gestionar las peticiones del usuario en la capa *Vista* y transmitir las al *Modelo* y viceversa. La Figura 16 muestra una representación gráfica de las 3 capas que forman el patrón arquitectónico MVC y la comunicación entre estas capas.

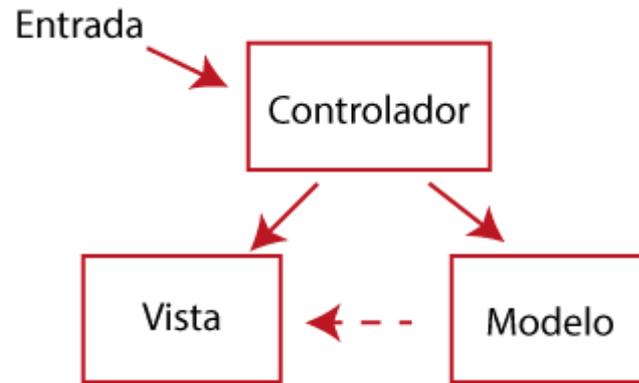


Figura 16. Modelo Vista Controlador

7.2 Modelo

El modelo representa los datos de la aplicación. Estas clases son clases Java que siguen la convención de JavaBeans. Estas clases se ocupan de acceder a los datos de la base de datos, así como modificarlos o añadir nuevos datos a la base de datos. En la Figura 17 se puede observar un ejemplo de una clase de modelo, en concreto este ejemplo es la clase modelo de los productos. Esta clase implementa la interfaz `IProductoService` doónde se encuentran las declaraciones de los métodos que en esta clase se implementan.

En esta clase también se encuentra la instancia a la clase `ProductoRepository` que es una interfaz que extiende de `JpaRepository` doónde se encuentran los métodos que acceden o modifican la base de datos como `findAll()` o `findById(id)` un ejemplo de estos métodos se encuentra en la Figura 17.

```

1 package com.proyecto.service.db;
2
3 import java.util.List;
4
5 @Service
6 @Primary
7 public class ProductoServiceJpa implements IProductoService {
8     @Autowired
9     private ProductoRepository productoRepo;
10
11     public List<Producto> obtenerProductos() {
12         return productoRepo.findAll();
13     }
14
15     public Producto obtenerPorId(Integer id) {
16         Optional<Producto> optional = productoRepo.findById(id);
17         if (optional.isPresent()) {
18             return optional.get();
19         }
20         return null;
21     }
22 }

```

Figura 17. Modelo Producto

En las clases de modelo de encuentran los métodos que son llamados desde el controlador y por eso a nivel de clase se marcan con la anotación `@Service`. También se utiliza la anotación `@Primary` a nivel de clase ya que en un proceso intermedio de la implementación se habían creado unas implementaciones de modelo estáticas y con esta anotación marcamos cual de las dos clases será la encargada de tratar los datos.

7.3 Vista

La vista es la interfaz gráfica que el usuario recibe en su navegador después de realizar una petición. Esto es código HTML que es renderizado de forma gráfica por el navegador. Para que el código HTML sea renderizado con los datos que recibe desde el modelo se ha usado el framework Thymeleaf. Para usar este framework basta con añadir en la etiqueta html de los documentos la extensión

`<html xmlns:th="http://www.thymeleaf.org">` como se muestra en la línea 2 del código de la Figura 19.

Para obtener los datos que llegan desde el controlador se utiliza el prefijo `th:` seguido de la etiqueta correspondiente, por ejemplo, `text`, para escribir dentro de una etiqueta que acepte texto, `each` para recorrer una lista de objetos que se recibe desde el controlador, entre otras; para obtener el objeto o el valor del atributo del objeto se

escribe en estas etiquetas el símbolo \$ seguido del nombre del objeto o objeto punto atributo entre llaves como por ejemplo se puede ver en la línea 35 y 38 de la Figura 18.

```

34<  <ul class="list-group list-group-flush">
35<  <li class="list-group-item" th:each="cat : ${ListaCategorias}">
36<    <div>
37<      <div class="borde">
38<        <h2 class="titulo-categoria" th:text="${cat.categoria}"></h2>
39<        <button class="btn btn-prim" type="button"
40<          data-bs-toggle="collapse"
41<          th:data-bs-target="'#'+cat.categoria" aria-expanded="false"
42<          th:aria-controls="${cat.categoria}">
43<          
44<        </button>
45<      </div>
46<      <div class="producto collapse" th:id="${cat.categoria}"
47<        th:each="productos : ${ListaProductos}"
48<        th:if="${cat.id} == ${productos.categoria}">
49<        <form th:object="${producto}" th:action="@{/anyadir}" method="post">
50<          <input type="hidden" th:value="*id" id="id" />

```

Figura 18. Vista Carta

Para hacer referencia a las distintas secciones del proyecto se utiliza el prefijo th: seguido de la etiqueta que se requiera utilizar y el link se escribe iniciando con el símbolo @ y seguido de la url a dónde apunta entre llaves como por ejemplo se encuentra en el link en la línea 5 y 6 de la Figura 19 que hace referencia a los archivos CSS que posee este proyecto.

```

1 <!DOCTYPE html>
2< <html xmlns:th="http://www.thymeleaf.org">
3< <head>
4 <meta charset="ISO-8859-1">
5 <link rel="stylesheet" th:href="@{/css/header.css}">
6 <link rel="stylesheet" th:href="@{/css/gastrotake.css}">
7 <link
8   href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
9   rel="stylesheet"
10  integrity="sha384-+0n0xVW2eSR50omGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCyb00L7+AMvyTG2x"
11  crossorigin="anonymous">
12< <script
13  src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js"
14  integrity="sha384-gtEjrD/SeCtmISkJKNUAaKMoLD0//ELJ19smozuHV6z3Iehds+3ULb9B9PLx0x4"
15  crossorigin="anonymous"></script>
16 <title>GastroTake</title>
17 </head>
18< <body class="body">
19   <header th:insert="fragments/header :: menu-cabecera"></header>
20<   <div class="container">
21<     <div class="row center" id="seccion">
22<       <div class="col-12">
23<         <h2 class="titulo col-sm-12 "></h2>
24<       </div>

```

Figura 19. Vista Home

7.4 Controlador

Los controladores son unas clases java especiales que se encargan de procesar todas las peticiones HTTP que realizan los usuarios y generar el modelo que será renderizado en la vista final.

Spring MVC está diseñado siguiendo el patrón de diseño de *Front Controller*. Esto significa que todas las peticiones de los usuarios serán recibidas por un servlet llamado *DispatcherServlet* que es el que actúa como *Front Controller*. Este servlet tiene varias funciones. Una de las principales funciones que posee este servlet es enviar las peticiones que recibe a su respectivo controlador para que sean procesadas. Otra función que desarrolla *Front Controller* es resolver los nombres de las vistas que serán enviadas a la Vista como se puede ver en la línea 53 de la Figura 20.

Las clases controlador se especifican con la anotación a nivel de clase con `@Controller`. También los métodos que se ejecutaran al recibir la petición deben estar etiquetados. Esta selección de métodos se hace a través de anotaciones como `@GetMapping` o `@PostMapping`. Estos métodos retornan un `String` que es el nombre del fichero HTML que se va a renderizar.

```
32 @Controller
33 public class HomeController {
34
35     @Autowired
36     private ICategoriasService serviceCategorias;
37     @Autowired
38     private IProductoService serviceProducto;
39     @Autowired
40     private IOpinionService serviceOpinion;
41     @Autowired
42     private IUsuariosService serviceUsuario;
43     @Autowired
44     private IReservasService serviceReservas;
45
46     @GetMapping("/")
47     public String mostrarHome(Model model) {
48         List<String> menu = new LinkedList<String>();
49         menu.add("CARTA");
50         menu.add("RESERVAR MESA");
51         menu.add("OPINIONES");
52         model.addAttribute("menu", menu);
53         return "home";
54     }
55
56     @GetMapping("/carta")
57     public String mostrarCarta(Model model) {
58
59         List<Categoria> listaC = serviceCategorias.obtenerCategorias();
60
61         System.out.println(listaC.toString());
62         model.addAttribute("listaCategorias", listaC);
63
64         List<Producto> lista = serviceProducto.obtenerProductos();
65         System.out.println(lista.toString());
66         model.addAttribute("listaProductos", lista);
67
68         return "carta";
69     }
70 }
```

Figura 20. Controlador

7.5 Base de datos

Para la implementación de la base de datos relacional se ha utilizado la herramienta MySQL Workbench. Aquí se ha definido un esquema relacional con un esquema llamado `gastrotakedb`. Aquí se encuentra la persistencia de los datos y para enlazar esta base de datos con la aplicación se han seguido los siguientes pasos:

1. En el fichero `pom.xml` de la aplicación se ha añadido la dependencia para tener disponible el conector java con MySQL como se muestra en la Figura 21.

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

Figura 21. Dependencia MySQL

2. El siguiente paso que se ha realizado es modificar el archivo `application.properties`. Este archivo viene dado, pero de forma vacía al crear el proyecto con SpringBoot. Aquí, tal y como se muestra en la Figura 22, se especifica la url de la base de datos, así como el usuario y la contraseña para acceder a ella.

```
1 # DATASOURCE (MySQL 8.0)
2 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
3 spring.datasource.url=jdbc:mysql://127.0.0.1:3306/gastrotakedb?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true
4 spring.datasource.username=root
5 spring.datasource.password=gastrotake
```

Figura 22. Conexión a base de datos

En la Figura 23 se presenta el modelo de base de datos que se ha utilizado con las diferentes tablas para guardar los datos de la aplicación web.

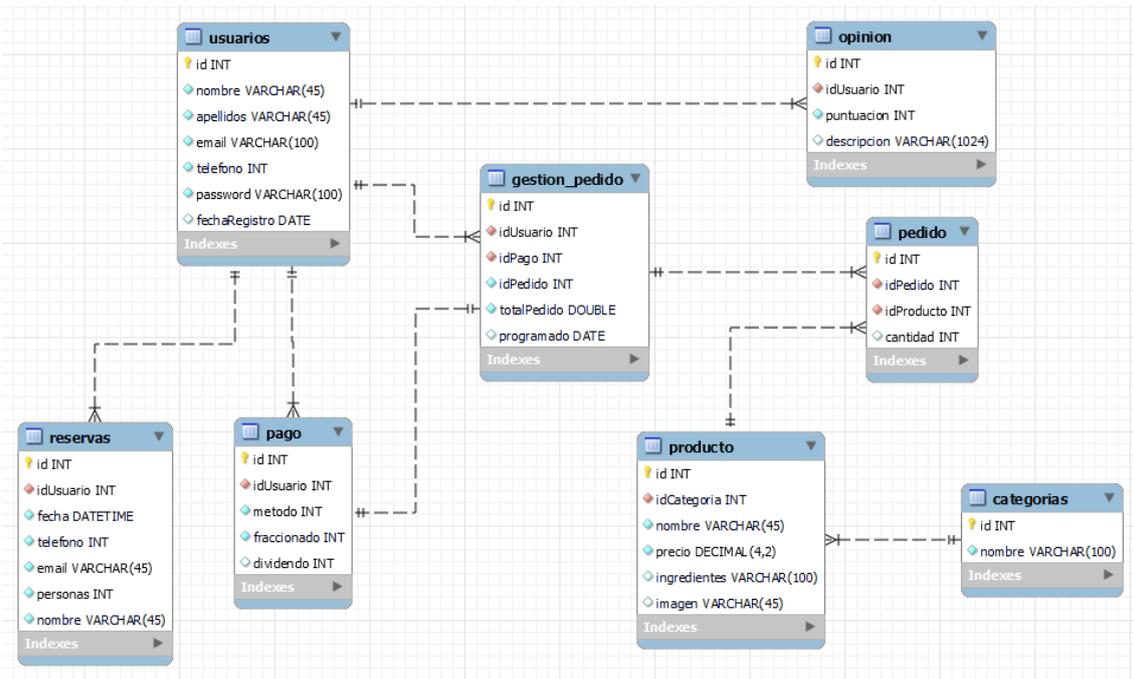


Figura 23. Esquema de la base de datos

A continuación, se especifican las diferentes tablas empleadas para almacenar la información de la aplicación web:

1. **usuarios**: En esta tabla se almacenan los datos de los usuarios registrados.
 - id: identificador de cada usuario.
 - nombre: nombre del usuario.
 - apellidos: apellidos del usuario.
 - email: email de registro de usuario.
 - telefono: teléfono del usuario.
 - password: contraseña del usuario.
 - fechaRegistro: fecha de registro del usuario.
2. **categorias**: Esta tabla contiene cada categoría de la carta, así como un identificador de cada una.
 - id: identificador de la categoría.
 - categoria: nombre para cada categoría.
3. **producto**: Esta tabla contiene toda la información relacionada de cada producto que se muestra en la carta. La tabla esta compuesta por los siguientes campos:
 - id: identificador de cada producto.
 - idCategoria: clave foránea de categoría.
 - nombre: nombre del producto.



- precio: precio del producto.
 - ingredientes: ingredientes del producto.
 - imagen: url de la imagen.
4. **reservas:** En esta tabla se guardan cada reserva que se realiza.
- id: identificador de la reserva.
 - idUsuario: clave foránea usuario.
 - fecha: fecha de la reserva.
 - telefono: número de teléfono del usuario que realiza la reserva.
 - email: dirección de correo del usuario de la reserva.
 - personas: número de personas para la reserva.
 - nombre: nombre de la reserva.
5. **opinion:** En esta tabla se encuentran las distintas opiniones de los usuarios.
- id: identificador de cada opinión.
 - idUsuario: clave foránea del usuario.
 - puntuacion: puntuación que se ha dado en la opinión.
 - descripcion: comentario que ha realizado el usuario.
6. **pedido:** En esta tabla se almacenan todos los pedidos que se realizan.
- id: identificador de cada pedido.
 - idPedido: clave foránea del pedido.
 - idProducto: clave foránea del producto.
 - programado: fecha de la entrega del pedido si ha sido programado.
 - cantidad: cantidad del producto seleccionado en la carta.
7. **pago:** En esta tabla se guardan los campos necesarios para la realización del pago. Estos campos son los siguientes:
- id: identificador, clave primaria.
 - idUsuario: clave foránea del usuario.
 - metodo: método de pago seleccionado.
 - fraccionado: si se ha seleccionado el pago fraccionado o no.
 - dividendo: número por el que se divide el total.
8. **gestion-pedido:** En esta tabla se guardan los pedidos.
- id: identificador del pedido.
 - idUsuario: clave foránea del usuario.
 - idPago: clave foránea del pago.
 - idPedido: clave foránea del pedido.
 - totalPedido: total del pedido.
 - programado: fecha de la entrega del pedido si ha sido programado.

8. Aplicación final

En esta sección se describe la aplicación final desarrollada. Por un lado, se encuentran los cambios que se han realizado a partir de la evaluación de usabilidad realizada con los usuarios. Por otra parte, se encuentra la descripción del funcionamiento de la aplicación final. Estas secciones están acompañadas de algunas capturas de la aplicación final para apreciar los cambios o entender el funcionamiento de esta aplicación.

8.1 Cambios a partir de la evaluación de usabilidad

A partir del cuestionario realizado en la evaluación de usabilidad se ha podido observar algunos fallos de diseño en el prototipado que han sido modificados y corregidos. Las distintas modificaciones realizadas se enumeran a continuación:

1. En la página inicial se ha modificado los siguientes puntos (que se pueden ver reflejados en la Figura 24):
 - 1.1. El diseño de los botones que acceden a las distintas partes de la aplicación.
 - 1.2. Se ha eliminado el botón de 'Programar Pedido' y esta funcionalidad se ha integrado a la hora de realizar el pago.
 - 1.3. Se ha eliminado el menú en forma de hamburguesa añadiendo las funcionalidades que tenía este menú al icono de perfil situado en la parte superior izquierda.
 - 1.4. Se ha cambiado el tipo de fuente a una más agradable y legible para el usuario. Para ello se ha elegido una tipografía que poseía las letras minúsculas más altas, ya que según algunos blogs esto favorece a la legibilidad sobre todo en tamaños reducidos como puede ser un smartphone. [16]
2. En el apartado de carta se han realizado las siguientes modificaciones que también se pueden observar en la Figuras 26 y 27:
 - 2.1. Se ha cambiado el aspecto de esta vista. En primer lugar, se ha quitado el borde que cubría a cada producto.
 - 2.2. También se ha modificado el diseño de la imagen pasando a ser con forma circular. La tipografía del título del producto, así como del precio se ha cambiado.
 - 2.3. La ubicación del precio, ha pasado a situarse a la derecha del título del producto dejando más espacio para el elemento del contador. De esta forma el contador se ha podido hacer más grande para facilitar la tarea al usuario.

- 2.4. Inicialmente, debajo de la imagen del producto aparecía el botón '+info' el cual redirigía al usuario a una nueva página con los detalles del producto. En esta vista se encontraba el nombre del producto, la imagen de éste junto a los ingredientes y los alérgenos junto a un contador para añadir o quitar el producto del carrito. Esta página ha sido sustituida por un pop up que muestra de forma similar la vista anterior sin la necesidad de redirigir innecesariamente al usuario a otra vista. En esta vista se ha omitido el contador de producto ya que al no cambiar de vista el usuario tiene acceso al contador de la carta. En este punto también se ha añadido la opción de que no solo al pulsar sobre el botón '+info' aparezca el pop up con el detalle, sino que también se muestra al pulsar sobre la imagen ya que es bastante usual que los usuarios presionen sobre la imagen para ampliarla.
- 2.5. El nuevo pop up aparece sobre la vista de la carta, oscureciendo el fondo para resaltar el detalle que aparece. Este se puede cerrar de dos formas. Pulsando sobre el botón en forma de 'x' que aparece junto al nombre o pulsando en cualquier lugar fuera del pop up.



Figura 24. Pantalla Inicio

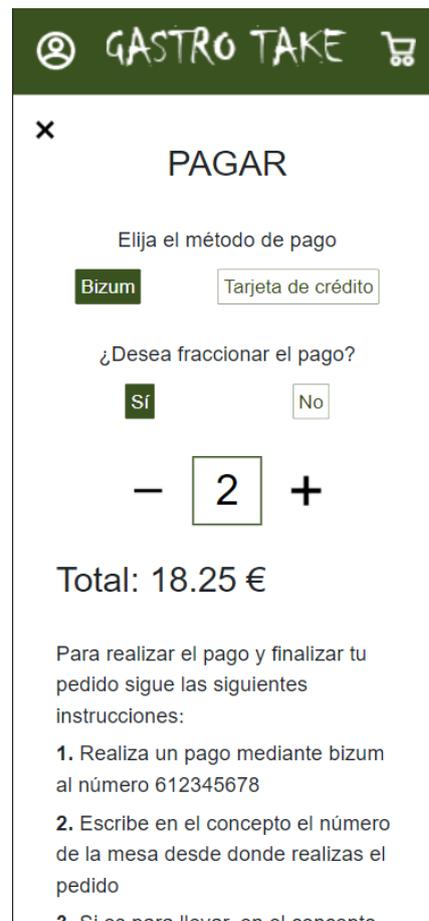


Figura 25. Pantalla Pagar



Figura 26. Pantalla Carta

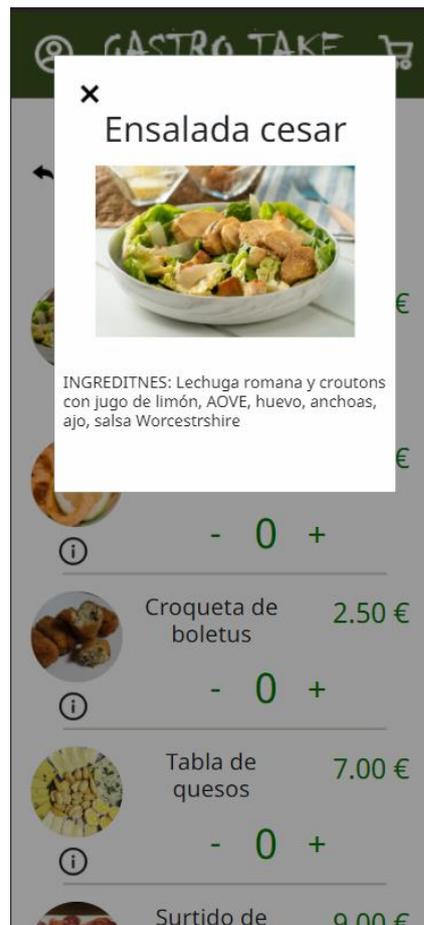


Figura 27. Popup Carta

8.2 Funcionalidad

En este apartado se presenta la funcionalidad de la aplicación. Aunque se trata de una aplicación sencilla que no precisa de manual, esta sección puede tomarse a modo de manual de la aplicación.

Cuando el usuario accede desde su teléfono móvil a la url desde donde se carga la aplicación en primer lugar se encontrará la pantalla de inicio donde se encuentra el menú principal de la aplicación. Aquí se encuentran los tres grandes apartados en los que se divide la aplicación: Carta, reservar mesa y opiniones.

Si el usuario clicca sobre opiniones, se le redirigirá a <https://www.url.es/opiniones> donde se encuentra la vista donde aparecen todas las opiniones que otros usuarios han escrito. Estas opiniones van acompañadas de una valoración entre 0 y 5 estrellas. Estas valoraciones se pueden ver a modo de estadística en la parte inicial de la vista. Para

volver al menú inicial el usuario debe clicar en la 'x' que se encuentra en la parte superior izquierda. Esta funcionalidad se encuentra en todas las vistas.

Si el usuario pulsa sobre 'Reservar mesa' se le redirigirá a <https://www.url.es/reservas> donde se presenta la vista donde tendrá que rellenar un formulario, seleccionando: el día en el que quiere hacer la reserva, la hora, las personas para las que hace la reserva, un nombre, su teléfono y email para recibir actualizaciones de su reserva. Para finalizar la reserva pulsará sobre el botón 'RESERVAR' que aparece en la parte final del formulario. En caso de que algún campo esté vacío o esté incorrecto se le mostrará al usuario y no le permitirá enviar la reserva. En caso de que todos los campos sean correctos se realizará la reserva y se redigirá al usuario a la pantalla inicial y se le mostrará un mensaje 'Reserva realizada'.

En el caso de que el usuario pulse sobre 'Carta', se le redirigirá a <https://www.url.es/carta> donde se muestra la carta. Esta vista se presenta con las distintas secciones en las que se divide la carta como se muestra en la Figura 6. El usuario deberá pulsar sobre la sección o las secciones que desee abrir. En este punto se despliegan los diferentes productos de cada sección. En esta subsección cada producto posee una imagen y un botón de información donde al pulsar sobre ellos aparece un pop up con el nombre del producto, su imagen a tamaño completo y los ingredientes de éste. El usuario puede cerrar este popup pulsando sobre la 'x' o pulsando sobre cualquier punto de su pantalla fuere del popup. Siguiendo con el uso de la interfaz de los productos, cada uno de ellos también aparece un contador, en este si el usuario no solo está leyendo la carta, sino que desea realizar un pedido, añadirá a su carrito cuantas unidades quiere de cada producto. En todos los productos aparece el precio de cada uno de ellos.

Si el usuario ha añadido productos al carrito y desea seguir con el pedido, pulsará sobre el icono de la cesta que se encuentra en la parte superior derecha de la aplicación. Esto redirigirá al usuario a <https://www.url.es/carrito> donde se muestra al usuario los productos seleccionados, la cantidad y el precio de cada uno, en la parte final se muestra el total del pedido. En el caso de que el usuario quiera programar su pedido deberá marcar la casilla y se le mostrará un formulario para que seleccione cuando quiere su pedido. Para seguir con el pago el usuario debe pulsar el botón 'PAGAR'. Si el usuario no está logeado se le redirigirá a la página de login donde podrá elegir si logearse o registrarse y posteriormente seguirá con su pago.

Una vez el usuario está logeado se redirige a la parte final. En esta vista se muestra un formulario dinámico en el que se le pide al usuario que elija su método de empleo entre: bizum y tarjeta de crédito. Si elije el método de tarjeta de crédito se le mostrará un

formulario convencional para pagos con tarjeta. En caso de que el usuario elija la opción de pago mediante bizum en el formulario se mostrará otras dos opciones donde debe elegir si fraccionar el pago o no. Si pulsa sobre 'No' aparece el total de su pedido y las instrucciones para realizar el pago. En caso de que el usuario deseara fraccionar el pago, el formulario seguiría con un contador donde el usuario debe introducir el número de personas entre las que quiere fraccionar el pago del pedido. La aplicación calcula y muestra el resultado de la operación y muestra los pasos a seguir para realizar el pago. Al pulsar sobre 'FINALIZAR' en ambos métodos de pago se redirige al usuario a la página principal y se le muestra el mensaje "Su pedido esta siendo tramitado".

9. Conclusiones

La finalidad de este TFG consistía en desarrollar una aplicación web móvil que ofreciera una buena experiencia de usuario. La aplicación debía ofrecer la posibilidad de no solo leer las cartas de los restaurantes, sino que, ya que el usuario hace el esfuerzo por visualizarla que ésta tenga más funcionalidades como poder crear un pedido en el restaurante, realizar el pago de este mediante nuevos métodos de pago y/o fraccionar el pago entre un número finito de personas.

El patrón MCV que se ha utilizado para implementar la aplicación ha facilitado la estructura de la misma, organizando por separado los componentes en interfaz gráfica, datos y conector.

Para la realización de esta aplicación se ha optado por Spring Boot, ya que esta tecnología de Spring permite simplificar el despliegue de la aplicación en un servidor porque Spring Boot posee uno internamente.

La herramienta Workbench de MySQL ha permitido crear y gestionar de una manera fácil los datos de la aplicación.

Con el framework de Thymeleaf se ha podido crear contenido HTML de forma dinámica, esto ha sido relativamente sencillo, aunque si que han surgido algunas complicaciones, pero gracias a la documentación encontrada en Internet y a la comunidad que utiliza estas tecnologías se han podido solventar las complicaciones.

Aunque con Spring Boot para la gestión del backend y Thymeleaf para el diseño del código HTML dinámico se ha podido completar los objetivos marcados, el framework Thymeleaf ha generado algunos obstáculos. El mayor de ellos ha sido la implementación del carrito ya que los objetos procesados por este framework no se podían capturar la implementación utilizada para el carrito teniendo que cambiar la implementación para conseguir este objetivo. Tras finalizar este proyecto y a modo de valoración personal, no elegiría otra vez el framework de Thymeleaf para un proyecto con características similares. En su defecto con la experiencia obtenida en este proyecto y fuera de él creo que otro framework como por ejemplo Vue.js hubiese facilitado bastante la realización de éste trabajo de final de grado.

9.1 Trabajo futuro

Los objetivos propuestos para el desarrollo de la aplicación web han sido cumplidos, pero durante el proceso de desarrollo de la aplicación han surgido algunas ideas nuevas que se plantean como trabajo futuro:

- Añadir la opción para que un usuario pueda realizar una crítica y que se muestre en el apartado de 'Opiniones'.
- Crear un sistema de fidelización con descuentos exclusivos para usuarios registrados.
- Añadir opción para realizar un pedido a domicilio.
- Añadir funcionalidad de eliminar una reserva.

10. Bibliografía

- [1] *Los 10 principios de usabilidad de Jakob Nielsen*. [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://profile.es/blog/los-10-principios-de-usabilidad-web-de-jakob-nielsen/>
- [2] *KPMG*. [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://es.wikipedia.org/wiki/KPMG>
- [3] *BIG4* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: [https://es.wikipedia.org/wiki/Big_Four_\(consultor%C3%ADa_y_auditor%C3%ADa\)](https://es.wikipedia.org/wiki/Big_Four_(consultor%C3%ADa_y_auditor%C3%ADa))
- [4] Anuario de la reestauración organizada en España. [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://assets.kpmg/content/dam/kpmg/es/pdf/2018/09/anuario-restauracion-organizada-espana.pdf>
- [5] *Spring Boot* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en:
- [6] *Java* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://definicion.de/java/>
- [7] *Java Script* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://es.wikipedia.org/wiki/JavaScript>
- [8] *HTML y CSS* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://concepto.de/html/>
https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- [9] *Thymeleaf* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://www.thymeleaf.org>
- [10] *Bootstrap* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://getbootstrap.com/>
- [11] *MySQL* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en:
- [12] *Workbench* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en: <https://dev.mysql.com/doc/workbench/en/wb-intro.html>



[13] *SpringToolSuite*. [En línea]. [Consultado el 20 de junio de 2021]. Disponible en:
<https://spring.io/tools>

[14] *Photoshop*. [En línea]. [Consultado el 20 de junio de 2021]. Disponible en:
https://es.wikipedia.org/wiki/Adobe_Photoshop

[15] *Modelo-Vista-Controlador*. [En línea]. [Consultado el 20 de junio de 2021].
Disponible en:

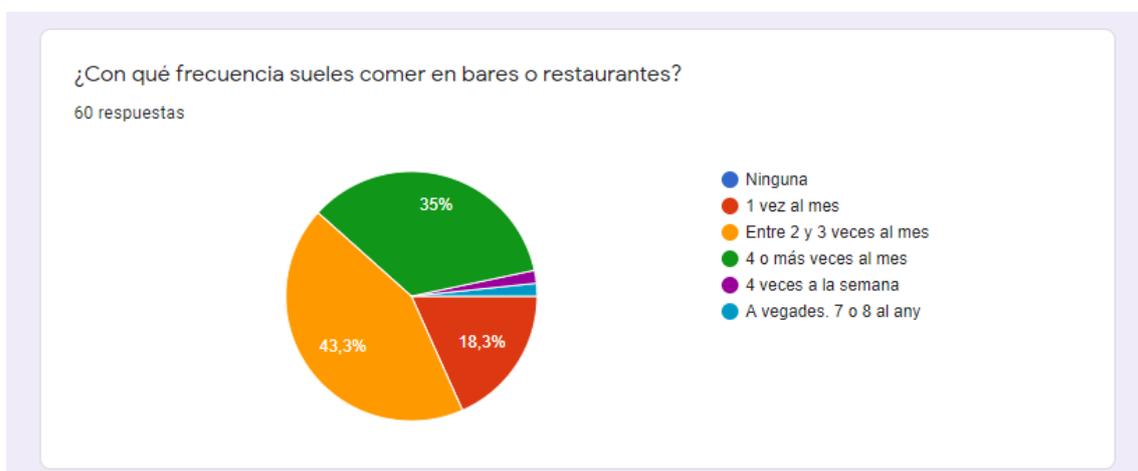
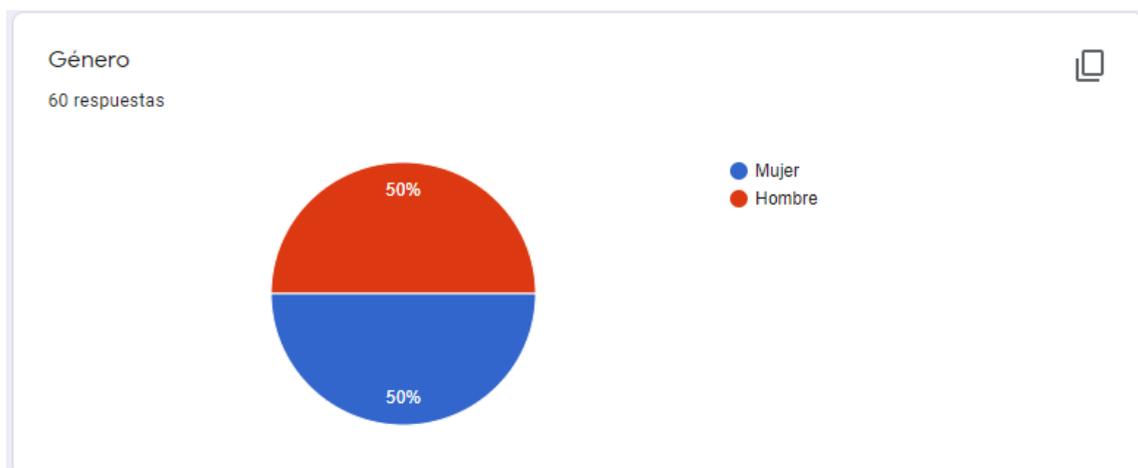
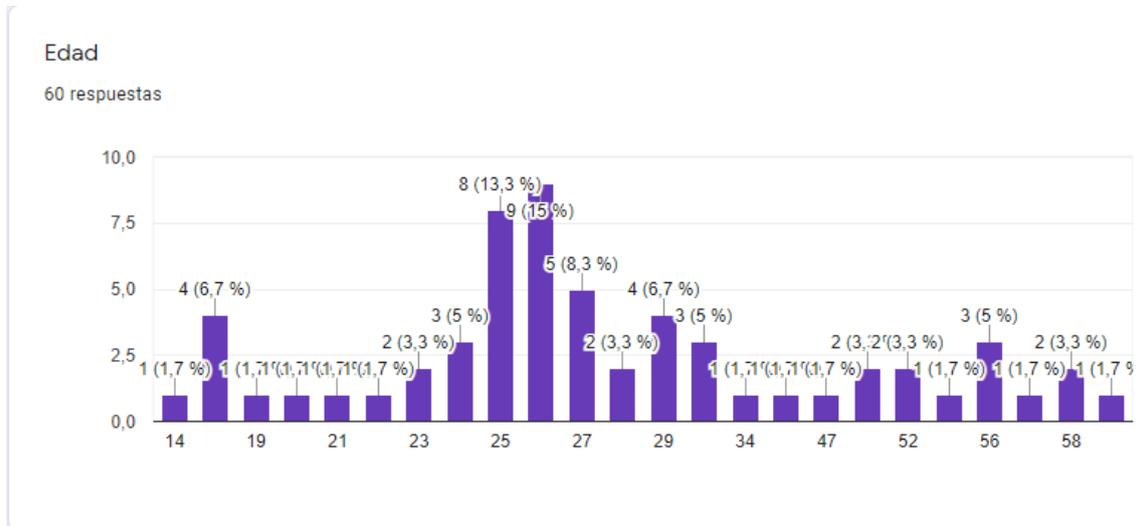
<https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>

[16] *Apps y fuentes: ¿cómo elegir la fuente adecuada?* [En línea]. [Consultado el 20 de junio de 2021]. Disponible en:

<https://www.pixartprinting.es/blog/elegir-fuente-aplicacion/>

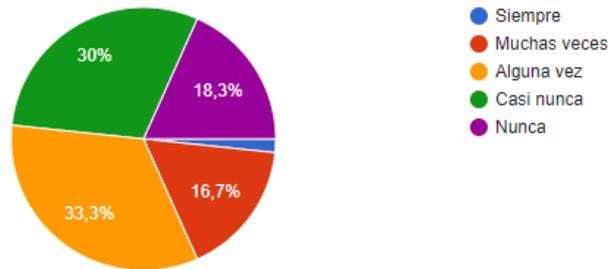
ANEXO A

En este anexo se presentan los resultados obtenidos en los cuestionarios de la primera etapa, análisis, del proceso DCU.



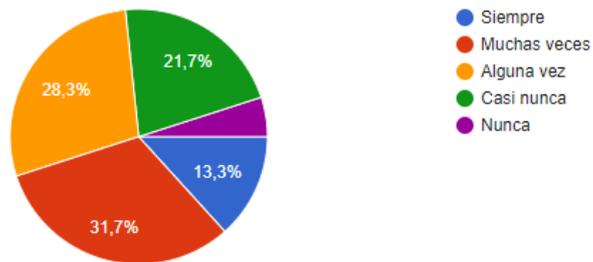
Con la aparición de la pandemia, ¿sueles hacer pedidos en bares/restaurantes a través del móvil?

60 respuestas



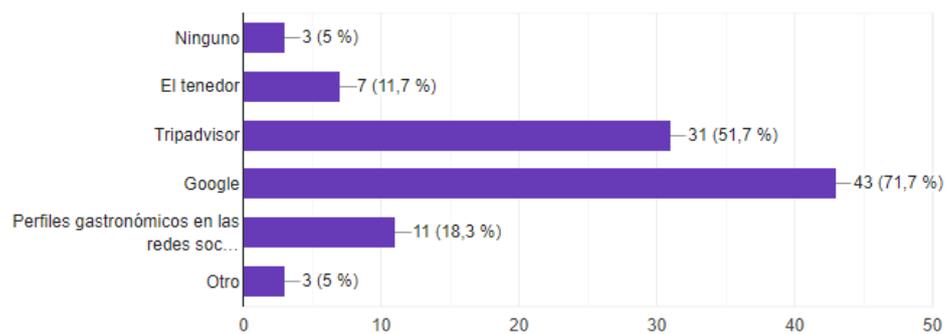
¿Consultas opiniones sobre los locales que visitas?

60 respuestas



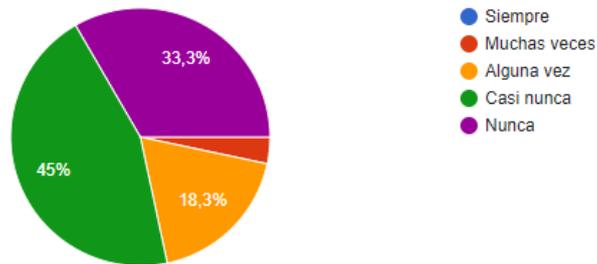
¿Dónde consultas las opiniones?

60 respuestas



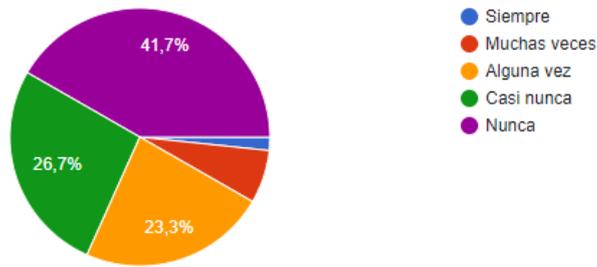
¿Dejas reseñas de los sitios que visitas?

60 respuestas



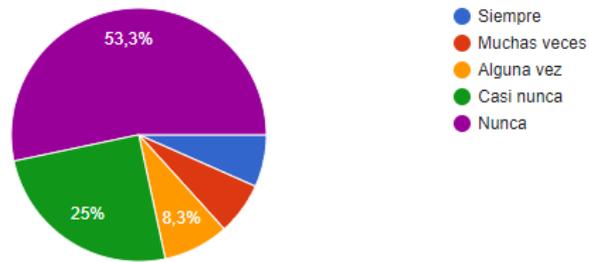
¿Te interesas por opciones vegetarianas y/o veganas?

60 respuestas



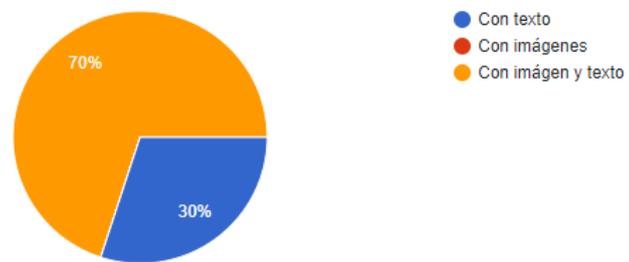
¿Consultas la tabla de alérgenos de los platos que vas a consumir?

60 respuestas



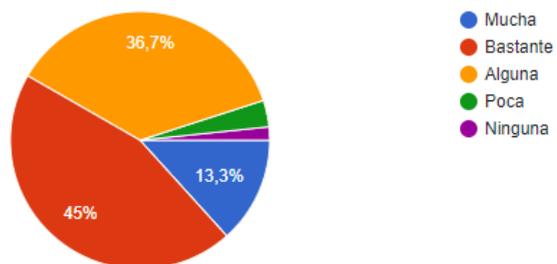
¿Cómo te resulta más fácil saber que plato elegir de la carta?

60 respuestas



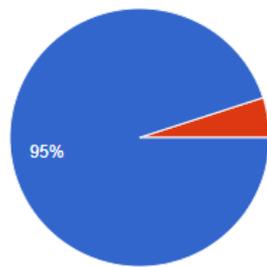
¿Qué importancia le das al precio de los productos?

60 respuestas



¿Cómo organizas el pago de la cuenta con el resto de comensales?

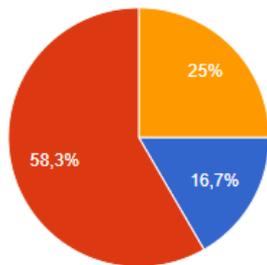
60 respuestas



- La cuenta se divide entre todos por igual
- Cada comensal solo paga aquello que ha comido o consumido

¿Cómo efectúas el pago de la cuenta?

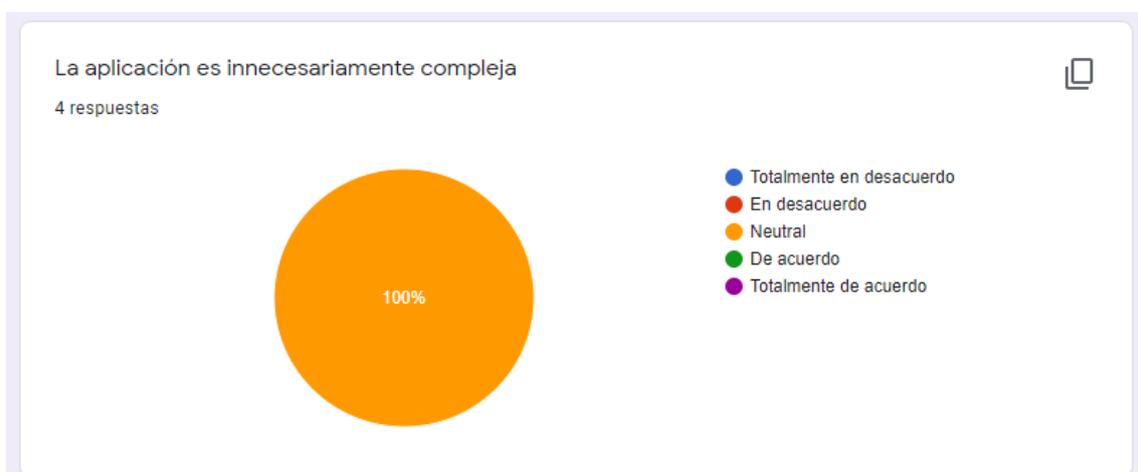
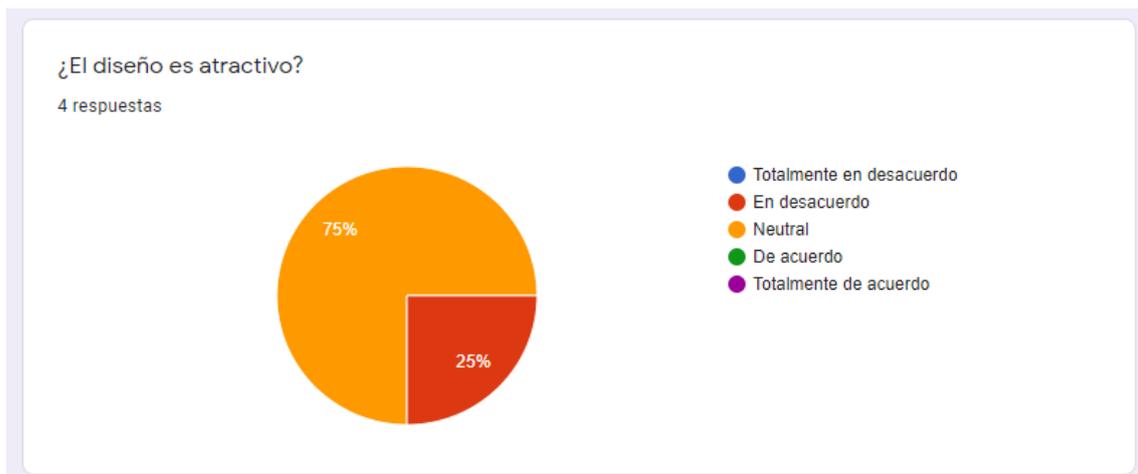
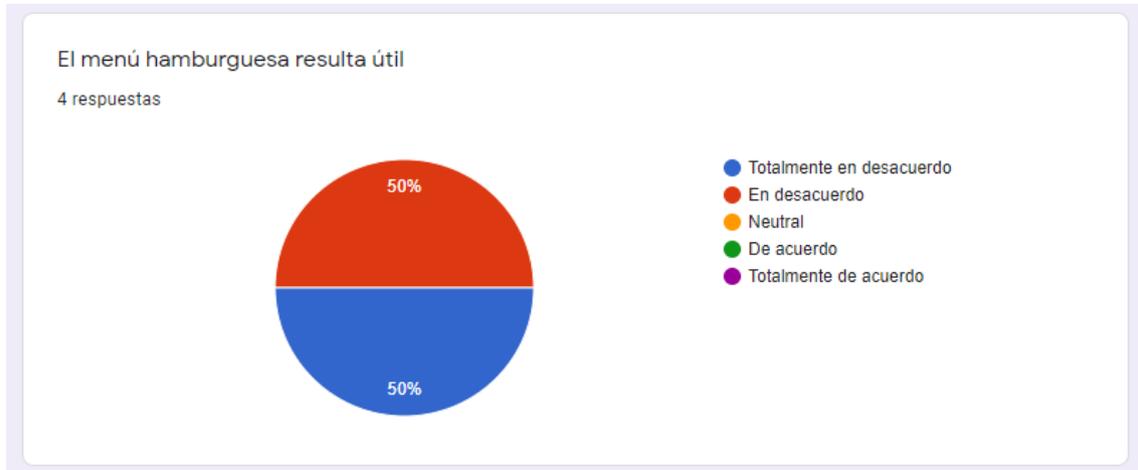
60 respuestas



- En metálico
- Con tarjeta de crédito
- A través del móvil

ANEXO B

En este anexo se presentan los resultados obtenidos en los cuestionarios de la etapa de evaluación temprana de los prototipos.



¿se siente seguro utilizando esta aplicación?

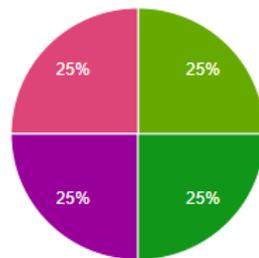
4 respuestas



- Sí
- No
- Tal vez

Del 1 al 10, ¿qué nivel de dificultad ha encontrado para logearse/registrarse?

4 respuestas

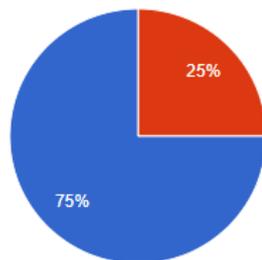


- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

▲ 1/2 ▼

Del 1 al 10, ¿qué nivel de dificultad ha encontrado para añadir un producto a la comanda?

4 respuestas



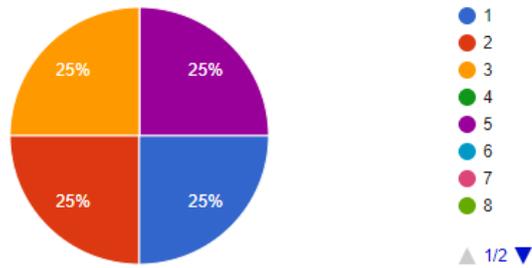
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

▲ 1/2 ▼

Desarrollo de una aplicación web móvil para la gestión de comandas en restaurantes

Del 1 al 10, ¿qué nivel ha encontrado para realizar un pago fraccionado por Bizum?

4 respuestas



¿Le gustaría usar esta aplicación?

4 respuestas

