



ULPGC
Universidad de
Las Palmas de
Gran Canaria

eii

ESCUELA DE
INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

Aplicación multiplataforma dedicada a la
comunicación entre un negocio y sus
clientes

Grado en Ingeniería Informática

AUTOR: Alejandro Ramírez Pulido

TUTORIZADO POR:

Francisco Alexis Quesada Arencibia

07/2024

Agradecimientos

A mi familia, por su apoyo durante toda mi vida.

*A mi tutor Francisco Alexis Quesada Arencibia
por ayudarme en este proyecto.*

A mis amigos, por su ayuda y compañía.

*Y sobre todo a mi pareja por ayudarme,
aguantarme y estar siempre ahí, tanto
en lo malo como en lo bueno.*

Resumen

En la actualidad, son numerosas las empresas que tienen dificultades de comunicación con los clientes por no estar actualizadas. Por ello, las compañías en ocasiones se enfrentan a obstáculos debido a la falta de un software especializado que les ayude a destacar en el mercado.

Este TFT plantea el desarrollo de una aplicación multiplataforma para dispositivos móviles que facilite la comunicación entre un negocio y sus clientes. En ella se implementa un programa de fidelización que recompense de forma monetaria a sus clientes, con el objetivo de atraer nuevos consumidores. Además, la herramienta permitirá al negocio enviar comunicaciones comerciales segmentadas, así como una funcionalidad a los clientes para solicitar un presupuesto y la correspondiente funcionalidad a los negocios para responder.

Abstract

Nowadays, many companies struggle to communicate effectively with their customers because they are not up to date. As a result, businesses sometimes face challenges due to the lack of specialized software that could help them stand out in the market.

This project proposes the development of a cross-platform mobile application to facilitate communication between a business and its clients. It includes a loyalty program that offers monetary rewards to customers, aiming to attract new clients. Additionally, the app will enable the business to send targeted commercial communications and provide a feature for customers to request a quote, as well as a corresponding feature for businesses to respond.

Índice general

Capítulo 1. Introducción	1
1.1. Estructura del documento	2
Capítulo 2. Estado actual y objetivos iniciales	4
2.1. Estado actual	4
2.1.1. El Paso 2000	4
2.1.2. Mi Energía DISA	6
2.1.3. Norauto	8
2.2. Objetivos iniciales	10
Capítulo 3. Competencias específicas y aportaciones del trabajo	12
3.1. Competencias específicas cubiertas	12
3.1.1. IS01	12
3.1.2. IS04	12
3.1.3. TI06	12
3.2. Aportaciones	13
3.2.1. Aportaciones al entorno socioeconómico	13
Capítulo 4. Normativa y legislación	15
4.1 Licencias del software utilizado	15
4.1.1. GNU GPL	15
4.1.2. Apache License	15
4.1.3. Microsoft Office 365 Education	16
4.1.4. MIT License	16
4.1.5. Microsoft Visual Studio Code	17
4.2. Seguridad de los datos	17
Capítulo 5. Metodología y planificación	19
5.1. Metodología	19
5.1.2. Ventajas del desarrollo en cascada	21
5.1.3. Desventajas del desarrollo en cascada	21
5.2. Planificación del proyecto	21

Capítulo 6. Herramientas y tecnologías utilizadas.....	23
6.1. Herramientas	23
6.1.1. Microsoft Visual Studio Code	23
6.1.2. GitHub	23
6.1.3. StarUML	23
6.1.4. Figma	24
6.1.5. Postman	24
6.1.6. Google Chrome	24
6.1.7. Android Studio	24
6.1.8. Microsoft Word	25
6.1.9. Firebase	25
6.2. Tecnologías	25
6.2.1. Git.....	25
6.2.2. JavaScript	26
6.2.3. NodeJs	26
6.2.4. MongoDB	26
6.2.5. React Native.....	26
6.2.6. Expo	27
6.2.7. Redux.....	27
Capítulo 7. Análisis.....	28
7.1. Actores.....	28
7.1.1. Cliente	28
7.1.2. Profesional	28
7.2. Casos de uso	28
7.3. Requisitos funcionales	33
7.4. Requisitos no funcionales	34
Capítulo 8. Diseño.....	35
8.1. Arquitectura del sistema	35
8.2. Base de datos.....	36

8.2.1 Estructura de la base de datos.....	37
8.3. Interfaz gráfica	40
Capítulo 9. Desarrollo.....	44
9.1. Implementación del Backend.....	44
9.1.1 Estructura del directorio raíz	44
9.1.2. Modelos, controladores y rutas	45
9.1.3. Utils	51
9.1.4. Middlewares	52
9.2. Implementación del Frontend	53
9.2.1. Estructura del directorio	53
9.2.2. Principales pantallas y componentes desarrollados.....	54
9.3. Pruebas	61
Capítulo 10. Conclusiones y trabajo futuro.....	63
10.1. Conclusiones	63
10.2. Trabajos futuros.....	64
Capítulo 11. Bibliografía	65
Capítulo 12. Anexos	68
Anexo 1. Manual de Usuario.....	68
A1.1. Iniciar sesión	68
A1.2. Registrarse	69
A1.3. Restablecer contraseña	69
A1.4. Vista de Tab Inicio	70
A1.5. Añadir oferta	72
A1.6. Vista de la oferta	72
A1.7. Clonar oferta	74
A1.8. Buzón de notificaciones	74
A1.9. Vista de Tab Presupuesto.....	75
A1.10. Solicitar presupuesto	76
A1.11. Vista de presupuesto	78

A1.12. Responder solicitud de presupuesto.....	80
A1.13. Vista Tab Facturas.....	80
A1.14. Añadir factura	81
A1.15. Vista de factura - Profesional.....	82
A1.16. Vista Tab Preferencias.....	83
A1.17. Perfil del usuario	84
A1.18. Editar perfil.....	84
A1.19. Vista Invitaciones.....	86
A1.20. Vista Facturas – Clientes	86
A1.21. Vista de factura – Cliente.....	87
A1.22. Vista Descuentos	88
A1.23. Añadir descuento.....	88
A1.24. Vista de descuento	89

Índice de figuras

Ilustración 2.1. Sección de ofertas y promociones en aplicación web EIPaso2000	5
Ilustración 2.2. Sección de contacto en aplicación web EIPaso2000	5
Ilustración 2.3. Formulario de solicitud de presupuesto a medida en aplicación web EIPaso2000	6
Ilustración 2.4. Sección de promociones en Mi Energía DISA.	6
Ilustración 2.5. Ejemplo de oferta en Mi Energía DISA.	7
Ilustración 2.6. Programa de fidelización por puntos en Mi Energía DISA.	7
Ilustración 2.7. Catálogo de canje de puntos en Mi Energía DISA.	7
Ilustración 2.8. Tarjeta amiga en Mi Energía DISA.....	7
Ilustración 2.9. Sección de ofertas en la aplicación web Norauto.....	8
Ilustración 2.10. Programa de fidelización por puntos en la aplicación web Norauto.	9
Ilustración 2.11. Ventajas de pertenecer al programa de fidelización en la aplicación web Norauto.	9
Ilustración 5.1. Metodología en cascada de cinco fases [12].	20
Ilustración 7.1. Diagrama de casos de uso – Actores.	29
Ilustración 7.2. Diagrama de casos de uso - Usuario registrado.	30
Ilustración 7.3. Diagrama de casos de uso – Cliente.	31
Ilustración 7.4. Diagrama de casos de uso – Profesional.....	31
Ilustración 8.1. Patrón de arquitectura de software MVC [36].	35
Ilustración 8.2. Diagrama entidad-relación.	37
Ilustración 8.3. Mockup pantalla Inicio.....	41
Ilustración 8.4. Mockup pantalla de registro.	43
Ilustración 9.1. Estructura del directorio del Backend.....	44
Ilustración 9.2. Archivo userRoutes.....	46
Ilustración 9.3. Función signInUser.	46
Ilustración 9.4. Archivo invoiceRoutes.....	47
Ilustración 9.5. Función getInvoiceById.....	47
Ilustración 9.6. Archivo discountRoutes.	48
Ilustración 9.7. Función disableDiscount.	48
Ilustración 9.8. Archivo notificationRoutes.....	49

Ilustración 9.9. Función updateReadNotification.	49
Ilustración 9.10. Archivo offerRoutes.	50
Ilustración 9.11. Función getAllByCategory.	50
Ilustración 9.12. Archivo budgetRoutes.	51
Ilustración 9.13. Función getUserBudgets.	51
Ilustración 9.14. Archivo nodemailer.	52
Ilustración 9.15. Middleware authMiddleware.	53
Ilustración 9.16. Estructura del directorio del Frontend.	54
Ilustración 9.17. Pantalla de inicio de sesión.	55
Ilustración 9.18. Esquema de validación del formulario de inicio de sesión.	55
Ilustración 9.19. Tab Nav de cliente.	56
Ilustración 9.20. Tab Nav de profesional.	56
Ilustración 9.21. Render de las ofertas activas.	56
Ilustración 9.22. Pantalla de inicio del profesional.	56
Ilustración 9.23. Componente BudgetItemComponent.	57
Ilustración 9.24. Pantalla presupuestos del cliente.	57
Ilustración 9.25. Pantalla del perfil del usuario.	58
Ilustración 9.26. Pantalla del formulario de nuevo descuento.	58
Ilustración 9.27. Componente LogoutButton.	59
Ilustración 9.28. Pantalla de Invitaciones.	59
Ilustración 9.29. Componente ShareComponent.	60
Ilustración 9.30. Componente DropdownComponent.	61
Ilustración 9.31. Formulario para añadir factura.	61
Ilustración 12.1 Iniciar sesión.	68
Ilustración 12.2. Formulario de registro.	69
Ilustración 12.3. Restablecer contraseña.	70
Ilustración 12.4. Vista Inicio – Profesional.	71
Ilustración 12.5. Vista Inicio – Cliente.	71
Ilustración 12.6. Añadir oferta.	72
Ilustración 12.7. Vista de una oferta activa - Profesional.	73
Ilustración 12.8. Vista de una oferta inactiva - Profesional.	73
Ilustración 12.9. Vista de una oferta – Cliente.	73
Ilustración 12.10. Clonar oferta.	74
Ilustración 12.11. Buzón de notificaciones - Profesional.	75

Ilustración 12.12. Buzón de notificaciones - Cliente.....	75
Ilustración 12.13. Vista de presupuesto - Profesional.	76
Ilustración 12.14. Vista de presupuesto - Cliente.	76
Ilustración 12.15. Solicitar presupuesto.....	77
Ilustración 12.16. Seleccionar archivos en solicitud de presupuesto.....	77
Ilustración 12.17. Límite de archivos alcanzado.....	77
Ilustración 12.18. Presupuesto sin responder.	78
Ilustración 12.19. Presupuesto respondido.	78
Ilustración 12.20. Ver foto de un presupuesto.	79
Ilustración 12.21. Archivo PDF descargado correctamente.....	79
Ilustración 12.22. Eliminar presupuesto.	79
Ilustración 12.23. Responder presupuesto.....	80
Ilustración 12.24. Vista Tab Facturas.	81
Ilustración 12.25. Añadir factura.....	82
Ilustración 12.26. Vista de factura - Profesional.	82
Ilustración 12.27. Eliminar factura - Profesional.....	82
Ilustración 12.28. Vista Tab Preferencias - Profesional.....	83
Ilustración 12.29. Vista Tab Preferencias - Cliente.....	83
Ilustración 12.30. Perfil de usuario – Profesional.....	84
Ilustración 12.31. Perfil de usuario - Cliente.....	84
Ilustración 12.32. Editar perfil - Profesional.....	85
Ilustración 12.33. Editar perfil - Cliente.....	85
Ilustración 12.34. Vista Invitaciones.	86
Ilustración 12.35. Alerta código amigo copiado.	86
Ilustración 12.36. Vista Facturas - Cliente.....	87
Ilustración 12.37. Vista de factura - Cliente.....	87
Ilustración 12.38. Eliminar factura - Cliente.....	87
Ilustración 12.39. Vista Descuentos.	88
Ilustración 12.40. Añadir descuento.	89
Ilustración 12.41. Vista de descuento.....	90

Índice de tablas

Tabla 2.1. Tabla comparativa de las funcionalidades de las aplicaciones.	10
Tabla 5.2. Planificación inicial.	22
Tabla 7.1. Resumen de los casos de uso.	33

Capítulo 1 . Introducción

Hoy en día, las pequeñas y medianas empresas desafían numerosos retos cada día para sobrevivir en un mercado altamente competitivo. Uno de los mayores desafíos a los que se enfrentan los negocios son las necesidades y expectativas de sus clientes.

Las tecnologías, las redes sociales e internet han tenido una evolución veloz que ha hecho que las PYMES tengan que adaptarse rápidamente ante estos cambios. Por lo que cada empresa deberá desarrollar y mejorar los canales de comunicación, la atención al cliente y crear una experiencia de compra y servicio más personalizada para cumplir las altas expectativas de los clientes.

Una de las herramientas más esenciales para nuestras vidas ha sido el teléfono móvil, y por ello ha convertido las aplicaciones móviles como algo primordial. Por lo que cada vez más PYMES utilizan como estrategia esta plataforma, que es accesible, versátil y eficiente para conseguir sus metas y poder mantener su competitividad. En cuanto a la implementación de esta herramienta en las pequeñas y medianas empresas tiene sus complicaciones, tal como la seguridad de los datos, los recursos técnicos y financieros, etc. No obstante, es un recurso con mucho potencial que podría hacer destacar a un negocio en el entorno empresarial.

Este proyecto se sitúa en el sector de los servicios, que afectan directa o indirectamente a la gestión de los complementos a dicho servicio, por ejemplo, en el caso de un taller de mecánica, hablamos de la gestión de los complementos al servicio, es decir, facturas, presupuestos, información adicional, comunicación directa, etc. No obstante, se podría extrapolar a cualquier otro negocio.

Por todo lo anteriormente mencionado, se ha propuesto una aplicación multiplataforma que permita tener un mejor servicio cliente-empresa y con ello conseguir un puesto destacable en el mercado. Es un proyecto con el que se pretende proporcionar un mejor servicio de comunicación, marketing y programa de fidelización que el público reclama.

1.1. Estructura del documento

Este Trabajo de Fin de Título se encuentra estructurado en los siguientes capítulos:

Capítulo 1. Introducción.

En este capítulo se presenta una breve introducción del proyecto realizado.

Capítulo 2. Estado actual y objetivos iniciales.

En este capítulo se analiza el estado del arte en el momento de la realización del trabajo, y se detallan los objetivos iniciales del proyecto.

Capítulo 3. Competencias específicas y aportaciones del trabajo.

En este capítulo se enumeran las competencias específicas del grado cubiertas durante la realización del TFT y se especifica las aportaciones de éste en el ámbito socioeconómico

Capítulo 4. Normativa y legislación.

En este capítulo se especifican las licencias software de los programas y las tecnologías usadas durante el desarrollo del prototipo, así como la normativa y legislación bajo la que se encuentra el proyecto.

Capítulo 5. Metodología y planificación.

En este capítulo se especifica la metodología que se ha utilizado y la planificación inicial del proyecto.

Capítulo 6. Herramientas y tecnologías utilizadas

En este capítulo se listan las herramientas y tecnologías utilizadas a lo largo de la implementación del trabajo.

Capítulo 7. Análisis.

En este capítulo se analiza los requisitos funcionales y no funcionales de la aplicación, así como los actores para los que está diseñada y sus correspondientes diagramas de casos de uso.

Capítulo 8. Diseño.

En este capítulo se detalla la arquitectura de software que se ha aplicado durante el desarrollo del sistema, así como el modelo de la base de datos implementada y los principios de diseño gráfico que se han tenido en cuenta para la interfaz gráfica de la aplicación.

Capítulo 9. Desarrollo

En este capítulo se detalla la estructuración de los directorios del código fuente del proyecto, se explica la implementación del proyecto y se describen las pruebas realizadas.

Capítulo 10 Conclusiones y trabajo futuro.

En este capítulo se detallan los resultados y conclusiones obtenidos tras el desarrollo de la aplicación. Además, se presentan una serie de mejoras y nuevas funcionalidades que se pueden añadir al sistema.

Capítulo 11. Bibliografía

En ese capítulo se lista todas las fuentes bibliográficas que se han consultado durante la realización de esta memoria.

Anexo 1: Manual de usuario.

En este anexo se expone de forma sencilla y detallada las distintas funcionalidades de la aplicación y como usarla correctamente.

Capítulo 2 . Estado actual y objetivos iniciales

2.1. Estado actual

Previo al desarrollo de un proyecto es imprescindible realizar un análisis de las aplicaciones o plataformas similares que ya se encuentran en el mercado, y así poder determinar cuál es el estado actual al que se enfrentará el proyecto una vez desarrollado. En este estudio se intenta identificar cuáles son las funcionalidades que cubre cada plataforma, determinando sus puntos fuertes y débiles, para definir posteriormente las estrategias y funciones a implementar, con el objetivo de que el proyecto sea útil y rentable, y así destacar ante los demás.

Para este proyecto, se ha realizado una búsqueda de aplicaciones móviles o servicios web con funcionalidades similares a las planteadas en el inicio del proyecto. Esta búsqueda se focaliza en las tres funcionalidades que el proyecto cubre:

- Poseer un programa de fidelización que recompense a sus clientes.
- Permitir a sus usuarios solicitar un presupuesto y la correspondiente capacidad de responder dicho presupuesto.
- La implementación de un servicio de comunicaciones comerciales que le permita a la empresa mostrarles a los clientes las distintas promociones activas.

A continuación, se exponen tres plataformas que cubren al menos una de las funcionales en las se ha focalizado la búsqueda realizada.

2.1.1. El Paso 2000

Aplicación web de la cadena de centros de automoción El Paso 2000, actualmente con 15 talleres repartidos por la Comunidad Autónoma de Canarias y especializados en mecánica rápida, servicio posventa y mantenimiento de neumáticos [1]. En las próximas ilustraciones se observa el diseño y funcionalidades que posee la aplicación. En ellas se puede apreciar que dispone

de una sección de solicitud de presupuesto y de un servicio de comunicaciones comerciales.

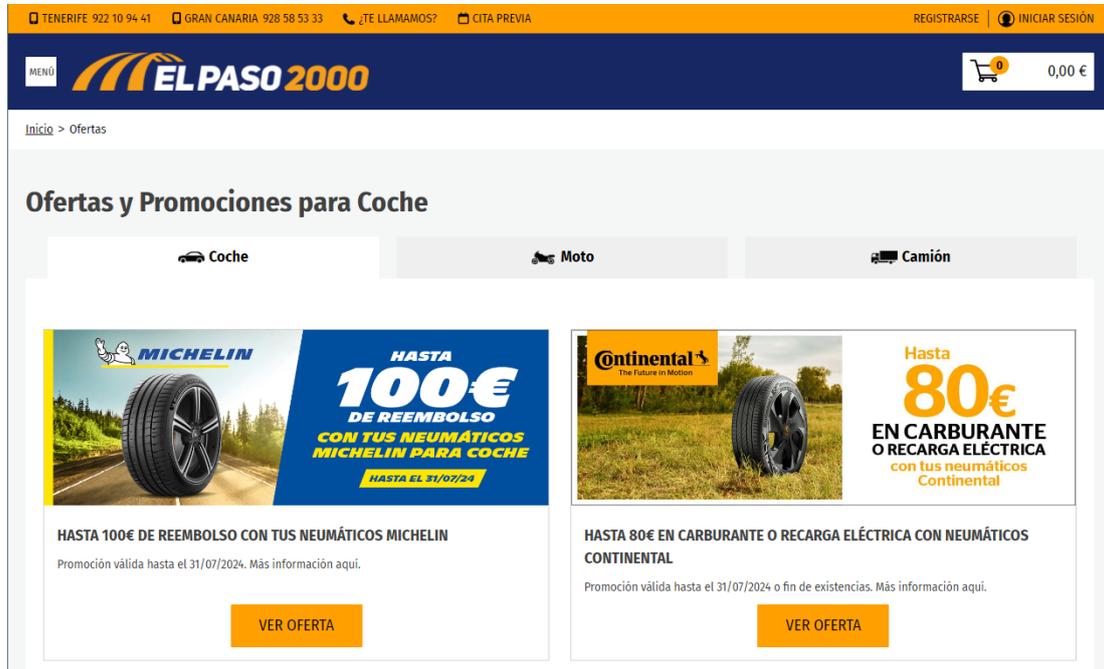


Ilustración 2.1. Sección de ofertas y promociones en aplicación web EIPaso2000

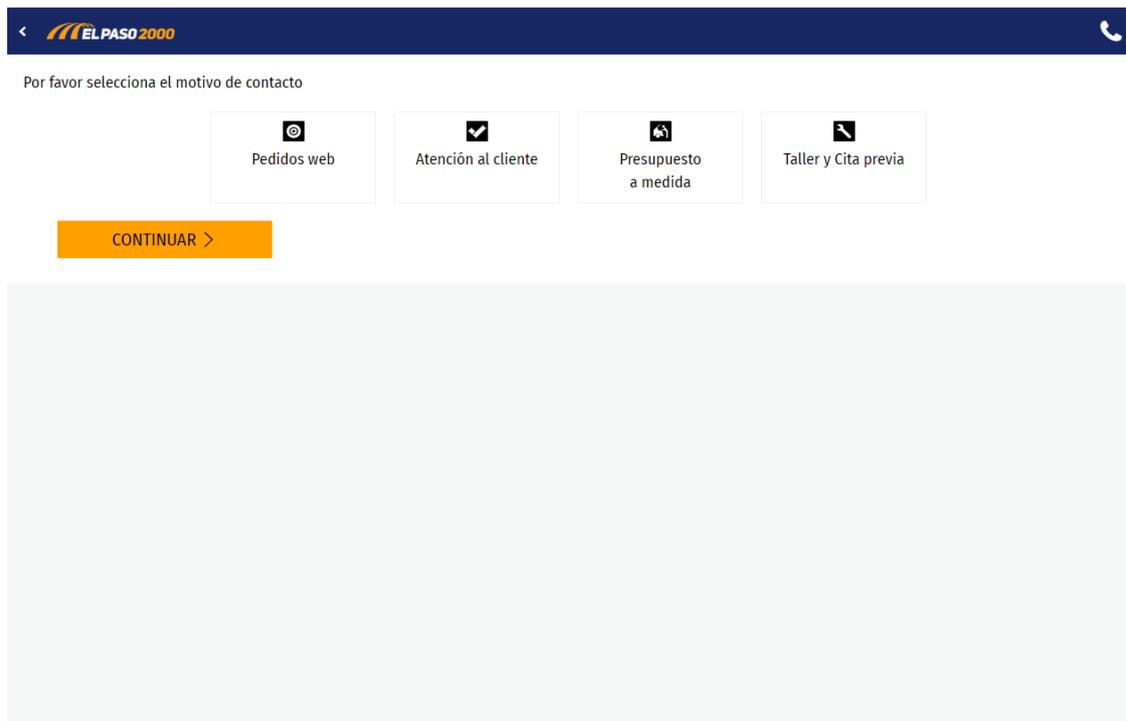


Ilustración 2.2. Sección de contacto en aplicación web EIPaso2000

Indica los siguientes datos (todos son obligatorios):

Nombre

Email

Teléfono

Indícanos los datos de tu vehículo y el servicio sobre el que te interesa recibir un presupuesto a medida.

Vehículo

Servicios

Matrícula

Comentario

Acepto recibir Descuentos y ofertas de ElPaso2000

He leído y acepto la [política de privacidad](#)

ENVIAR >

Ilustración 2.3. Formulario de solicitud de presupuesto a medida en aplicación web EIPaso2000

2.1.2. Mi Energía DISA

Aplicación móvil de la compañía grupo DISA. Esta empresa, proveedora de productos y servicios energéticos en todo el territorio nacional, se ha convertido en la única sociedad con ocupación en la logística, el almacenamiento y el transporte terrestre y marítimo, y con la única cadena de estaciones de servicios presente en todas las islas del Archipiélago Canario [2]. A continuación, se muestra el diseño y funcionalidades que posee la aplicación. En este caso se contempla un programa de fidelización y un apartado de promociones comerciales.



Ilustración 2.4. Sección de promociones en Mi Energía DISA.



Juega con DISA y entra en el sorteo de más de 2.000€ en premios de carburante

Rellena nuestras Europorra DISA y si escoges la selección ganadora de cada partido, entrarás en el sorteo de 1 premio de 100€ de carburante. ¡¡¡Puedes ganar tantos premios como sorteos hay!!!

Pero esto no acaba aquí. Recibirás un vale de 1€ en carburante por cada partido que aciertes.

¿Entonces, quieres marcar el gol del año con DISA?

Si es así, antes que nada, debes tener tu email correctamente informado en la app Mi Energía DISA, así como haber aceptado los consentimientos para el envío de comunicaciones comerciales y la opción de envío de correos electrónicos. Podrás comprobarlo en la sección Perfil, en el lateral superior derecho de la pantalla de inicio de la App, en las pestañas de Consentimientos y Notificaciones.

Si ya cumples estos requisitos, solo tienes que responder al email que te hemos enviado invitándote a participar en esta

Ilustración 2.5. Ejemplo de oferta en Mi Energía DISA.



Ilustración 2.6. Programa de fidelización por puntos en Mi Energía DISA.

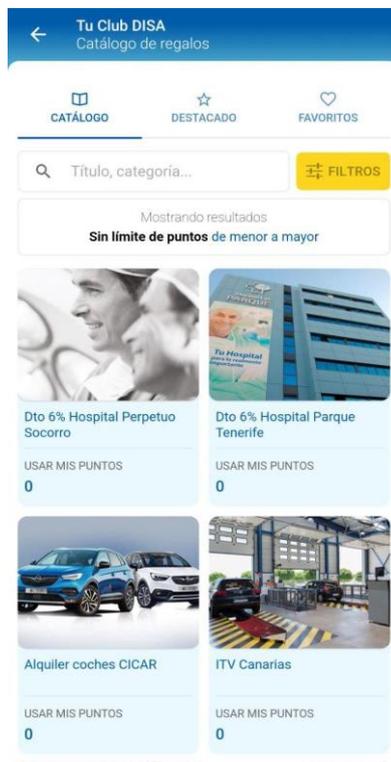


Ilustración 2.7. Catálogo de canje de puntos en Mi Energía DISA.



Ilustración 2.8. Tarjeta amiga en Mi Energía DISA.

2.1.3. Norauto

Aplicación web de la cadena de centros de automoción Norauto. Empresa líder en Europa en mantenimiento de automóviles con más de 50 años de experiencia con más de 90 centros en territorio español [3]. En las siguientes ilustraciones se observa el diseño y funcionalidades que tiene la aplicación. Se aprecia que posee un programa de fidelización por puntos y un servicio de comunicaciones comerciales.

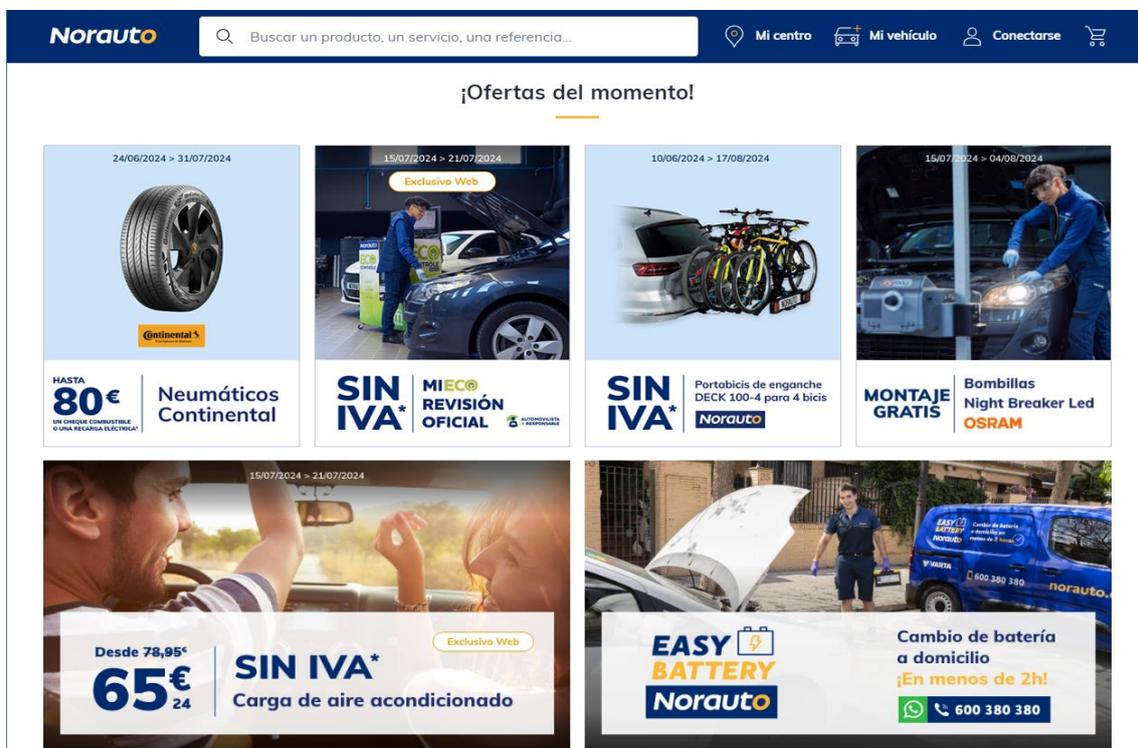


Ilustración 2.9. Sección de ofertas en la aplicación web Norauto.

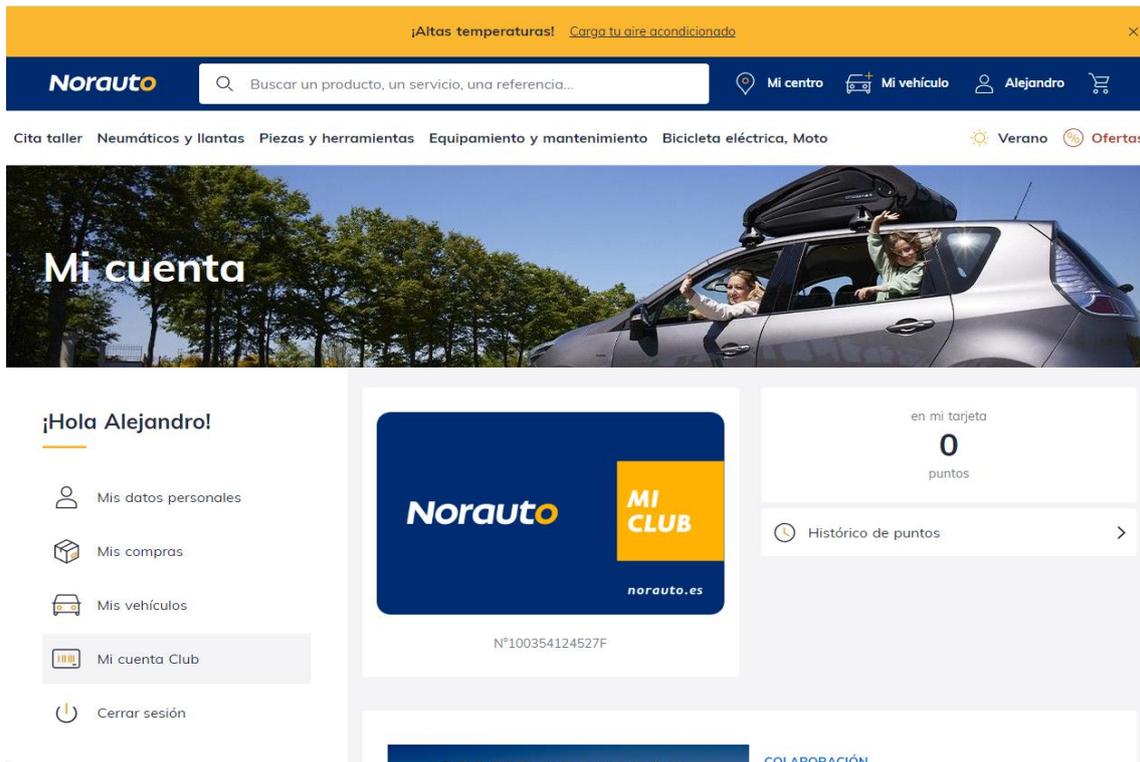


Ilustración 2.10. Programa de fidelización por puntos en la aplicación web Norauto.



Ilustración 2.11. Ventajas de pertenecer al programa de fidelización en la aplicación web Norauto.

Para finalizar con el apartado del Estado Actual, se muestra una tabla comparativa con las características destacables de las tres plataformas y la aplicación desarrollada.

Características destacables	EIPaso2000	Mi Energía DISA	Norauto	TFT
Funcionalidades básicas sin inicio de sesión	Sí	No	Sí	No
Programa de fidelización	No	Sí	Sí	Sí
Solicitud de presupuesto personalizado	Sí	No	No	Sí
Sección de comunicaciones comerciales	Sí	Sí	Sí	Sí
Sistema de registro de facturas	Sí	Sí	Sí	Sí
Sistema de notificaciones	No	Sí	No	Sí

Tabla 2.1. Tabla comparativa de las funcionalidades de las aplicaciones.

2.2. Objetivos iniciales

El objetivo de este trabajo es el diseño y desarrollo de un prototipo de aplicación multiplataforma con la que facilitar la comunicación entre un negocio y sus clientes. Siendo los objetivos iniciales del proyecto los siguientes:

- La implementación de un programa de fidelización para sus clientes.
- La implementación de solicitudes y envío de presupuestos.
- La implementación de un servicio de envío de comunicaciones comerciales que permita la segmentación de clientes.

Para poder cumplir correctamente con estos objetivos, se han marcado unas funcionalidades más específicas, según el rol del usuario que use la aplicación, los cuales se detallan a continuación:

- Desde el punto de vista del empresario:
 - Publicar y desactivar las promociones comerciales de su negocio.
 - Definir los descuentos a aplicar en el programa de fidelización y el objetivo necesario para su aplicación.
 - Añadir las facturas de los servicios prestados a sus clientes.

- Responder a las solicitudes de presupuestos presentadas por los clientes.
- Desde el punto de vista del cliente:
 - Solicitar un presupuesto personalizado para su problema.
 - Ver las distintas promociones activas del negocio.
 - Compartir la aplicación entre sus conocidos.
 - Ver las facturas de los servicios que ha consumido en el negocio
 - Observar qué objetivo del programa de fidelización ha cumplido y su descuento aplicable.

Cabe destacar que independientemente del rol que tengan, todos los usuarios podrán administrar su perfil y recibir notificaciones.

Capítulo 3 . Competencias específicas y aportaciones del trabajo

3.1. Competencias específicas cubiertas

3.1.1. IS01

“Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la ingeniería del software”.

Durante el desarrollo del proyecto se ha realizado un estudio de los requisitos necesarios para implementar un sistema fiable y eficiente que cubra los requisitos de usuarios fijados, determinando los actores presentes durante el uso de la aplicación y sus necesidades.

3.1.2. IS04

“Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales”.

Previo al inicio del desarrollo del proyecto se ha hecho un estudio de las aplicaciones similares existentes en el mercado. Tras esto, se comenzó con el diseño y desarrollo de una aplicación multiplataforma que cubre los distintos objetivos del proyecto.

3.1.3. TI06

“Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil”.

Este proyecto hace uso de servicios y tecnologías web que han sido estudiadas previamente para poder entender su funcionamiento y poder realizar una correcta implementación de ellas.

3.2. Aportaciones

3.2.1. Aportaciones al entorno socioeconómico

Cada año que transcurre, se aprecia cómo las empresas grandes se “comen” a las pequeñas o negocios que abren y al poco cierran. Hoy en día es muy difícil que se mantenga a “flote” una ferretería, un taller o una tienda de pueblo, ya que la ciudadanía prefiere ir a grandes almacenes. De modo que, para “sobrevivir” entre tanta competencia se tiene que ofrecer servicios o productos de calidad a los clientes, es decir, es importante destacar en el mercado.

En el supuesto de que un negocio sufra dificultades de gestión de los servicios, captación y retención de clientes, deberá anticiparse y realizar pequeñas acciones que generen grandes cambios.

Para comenzar el cambio, la empresa debe conocer su competencia y cómo distinguirse de estos. Por lo que es importante conocer los productos y servicios ofrecidos, el marketing, uso de plataformas o sitios webs, colaboraciones, etc.

La realización de este proyecto pretende que un determinado negocio consiga solventar estas adversidades haciendo uso de la aplicación multiplataforma desarrollada. Esta app proporcionará un mejor servicio de comunicación, marketing y un programa de fidelización al negocio.

En primer lugar, brinda una vía personalizada a la hora de solicitar presupuestos con capacidad de subir imágenes y videos, en el que el cliente pueda detallar de forma gráfica cuál es el problema que presenta.

En segundo lugar, ofrece un canal de comunicación comercial segmentada en función de las necesidades de cada cliente, mostrándole a cada cliente las ofertas que el negocio tiene activas.

Por último, proporciona la posibilidad al negocio de definir una serie de descuentos a sus usuarios, que logren atraer a nuevos clientes y que estos contraten los servicios ofrecidos por el negocio. Este descuento irá en aumento según el número de clientes atraídos.

En definitiva, la aplicación contribuye de forma positiva en la empresa y crea un valor único que le hace diferente a sus competidores.

Capítulo 4 . Normativa y legislación

4.1 Licencias del software utilizado

4.1.1. GNU GPL

La Licencia Pública General de GNU, proveniente del inglés GNU General Public License, es una licencia de software libre creada por Free Software Foundation en 1989. Esta licencia asegura que los usuarios finales tengan la libertad de estudiar, usar, modificar y compartir el software. Esta licencia ha tenido varias versiones desde su creación, siendo la más reciente la GNU GPL V3. Tiene dos propósitos principales, declarar que todo proyecto cubierto por ella sea de libre acceso, y proteger, mediante una práctica legal llamada copyleft, que todo software derivado de un software con GNU GPL también se distribuya bajo esta licencia [4].

Durante el desarrollo del proyecto se han usado las herramientas Git, StarUML y MongoDB que están cubiertas por esta licencia.

4.1.2. Apache License

La licencia Apache, creada y mantenida por Apache Software Foundation (ASF), se caracteriza por ser una licencia de software libre permisiva, ya que los usuarios deben cumplir una serie de condiciones esenciales al usar software cubierto por esta licencia. Estas condiciones incluyen la preservación de aviso de derechos de autor, la disposición de cualquier modificación realizada a un software con esta licencia y la incorporación de una copia de la licencia en cualquier modificación del software. Hay que destacar que la Apache License no es una licencia de tipo copyleft. Esta licencia ha tenido varias versiones, siendo la más reciente y utilizado la licencia Apache 2.0 [5].

Durante el desarrollo del proyecto se han usado las herramientas Firebase Storage y Android Studio que están cubiertas por esta licencia.

4.1.3. Microsoft Office 365 Education

Microsoft Office 365 Education facilita el acceso al paquete ofimático a estudiantes y educadores de las instituciones elegibles, entre las que se incluye la Universidad de Las Palmas de Gran Canaria (ULPGC), pudiendo su alumnado y profesorado, registrarse gratuitamente en Office 365 Education, utilizando su dirección de correo electrónico educativa, y así pudiendo acceder a una amplia gama de aplicaciones como Word, Excel, PowerPoint, OneNote y Microsoft Teams, entre otros servicios de la compañía. Este acceso gratuito permite a las instituciones educativas ahorrar en costos de licencias, ya que los estudiantes y educadores pueden utilizar estas aplicaciones en sus dispositivos personales, facilitando el acceso y la disponibilidad en cualquier momento y lugar. Tanto profesores como alumnos pueden instalar el paquete completo de Office en hasta 5 dispositivos, ya sean PC, tabletas o smartphones [6].

Durante el desarrollo del proyecto se ha usado la herramienta Microsoft Word que está cubierta por esta licencia.

4.1.4. MIT License

La licencia MIT, también conocida como Licencia X11, debido a que es el software gráfico que muestra la información X Window System, fue creada por el Instituto Tecnológico de Massachusetts. Esta licencia es un tipo de licencia de software libre y de código abierto que permite a los usuarios usar, copiar, modificar, publicar, distribuir y sublicenciar el software licenciado con muy pocas restricciones, siempre que se incluya el aviso de derechos de autor y el texto de la licencia MIT [7].

A diferencia de las licencias copyleft como la GNU GPL, la licencia MIT permite que el software derivado pueda ser tanto software libre como software propietario. Esto la hace muy compatible con otras licencias, incluyendo las licencias copyleft. La licencia MIT no ofrece garantías ni responsabilidad por los daños resultantes del uso del software, protegiendo así a los creadores y distribuidores de responsabilidades legales [7].

Durante el desarrollo del proyecto se han usado las herramientas React Native, React, React Redux, React Redux Toolkit, Expo, IonicIcons y NodeJs que están cubierta por esta licencia.

4.1.5. Microsoft Visual Studio Code

Este editor de código fuente desarrollado por Microsoft Corporation, lanzado en 2015 bajo la MIT License [8], concediendo a sus usuarios usar cualquier número de copias del software para desarrollar y probar sus aplicaciones. A pesar de esto, Microsoft solo otorga a sus usuarios algunos derechos de uso del software, reservándose el resto de los derechos [9].

Durante el desarrollo del proyecto se ha usado la herramienta Microsoft Visual Studio Code que está cubierta por esta licencia.

4.2. Seguridad de los datos

Los datos de los usuarios recogidos por la aplicación desarrollada son tratados de acuerdo con la Ley Orgánica de Datos Personales y Garantía de Derechos Digitales, que entró en vigor el 7 de diciembre de 2018, sustituyendo a la antigua Ley Orgánica 15/1999 de Protección Datos de Carácter Personal. En ella se constituyen los requisitos y responsabilidades en materia de protección de datos en compañías sobre cómo usar la información personal, de la misma manera que los derechos que tienen los usuarios y consumidores. La LODPGDD tiene como objetivo adecuar la normativa española al Reglamento General de Protección de Datos (RGPD).

El Reglamento (UE) 2016/679 o Reglamento General de Protección de Datos entró en vigor el 24 de mayo de 2016 y empezó a aplicarse a partir del 25 de mayo de 25 de mayo de 2018. El principal objetivo de este reglamento es la protección de las personas físicas, ya sean residentes o no de la Unión Europea, en cuanto al manejo de sus datos personales y a la libre circulación de estos datos dentro del territorio de la Unión Europea y el Espacio Económico Europeo [10].

Para cumplir con este reglamento, el tratamiento y almacenamiento de los datos se restringe a lo imprescindible. Siendo los datos recogidos (nombre, apellidos, correo electrónico y número del documento de identidad) de carácter general de tipo identificativo según el reglamento mencionado.

Capítulo 5 . Metodología y planificación

5.1. Metodología

Para la realización de este proyecto se ha escogido la metodología de desarrollo en cascada (en inglés, Waterfall Model), un método secuencial que permite representar un proyecto en base a una serie de fases que se suceden entre sí.

El desarrollo en cascada es un procedimiento secuencial caracterizado por dividir los proyectos en fases. Estas fases comienzan una vez se ha terminado la anterior, siendo los resultados de esta, la hipótesis de partida de la siguiente. A la inversa de otros modelos de desarrollo, como el modelo iterativo, las etapas en este modelo se ejecutan una sola vez [11].

La versión original de esta metodología apareció por primera vez en el ensayo *“Managing the Development of Large Software Systems”* del teórico Winston W. Royce, en el que propone un modelo iterativo incremental en el que cada etapa se basa en la anterior y verifica sus resultados [12]. Este modelo está compuesto por siete fases, y se ejecuta en varias iteraciones. Estas fases son:

1. Requisitos de sistema.
2. Requisitos de software.
3. Análisis.
4. Diseño.
5. Implementación.
6. Prueba.
7. Servicio.

La propuesta de Royce fue revisada por Barry Boehm en 1980 y por Ian Sommerville en 1985 [13].

En la práctica, existen diferentes versiones de esta metodología, siendo los más habituales los modelos compuestos por cinco fases, en los que las etapas 1, 2 y 3 del modelo de Royce se unen en una única fase de análisis. Resultando un modelo compuesto por las siguientes fases:

1. Análisis.
2. Diseño.
3. Implementación.
4. Verificación
5. Mantenimiento.



Ilustración 5.1. Metodología en cascada de cinco fases [12].

En la fase de análisis, se realiza un estudio de viabilidad y se definen los requisitos del sistema.

En la etapa de diseño, se diseña la arquitectura completa del proyecto y un plan de diseño detallado del mismo.

A continuación, en la fase de implementación, se aplica lo estudiado y definido en las fases anteriores.

Posteriormente, en la cuarta etapa, se prueba el prototipo implementado para verificar su correcto funcionamiento y determinar si el software cumple con los requisitos definidos. Tras esta etapa, el producto es entregado al cliente.

Finalmente, en la fase de mantenimiento, el producto se pone en marcha tras la aceptación por parte del cliente. Esta fase incluye el mantenimiento y mejora del producto.

5.1.2. Ventajas del desarrollo en cascada

- Posee una estructura sencilla.
- Está enfocado a documentos.
- La carga de trabajo y los costes totales pueden estimarse desde el comienzo del proyecto.
- Permite llevar un orden y una organización del trabajo más clara.

5.1.3. Desventajas del desarrollo en cascada

- En la práctica, es raro que un proyecto siga una secuencia lineal, creando una mala implementación del modelo.
- Un error de diseño detectado en la fase de verificación implica el rediseño y nueva programación del código.
- El cliente no se integra en el proceso de producción hasta que no acaba la etapa de implementación.
- Cualquier fase no puede empezarse hasta que la/s anterior/es no hayan terminado.

5.2. Planificación del proyecto

Al inicio de este trabajo se diseñó una planificación del proyecto para el documento “Propuesta de trabajo de fin de título”, en la que se detalla las distintas fases, con sus correspondientes tareas, y la duración estimada.

En la tabla 5.2 se muestra esta planificación, además de la duración real que se ha estado en cada fase:

Fases	Duración estimada (horas)	Duración real (horas)	Tareas
Estudio previo / Análisis	60	50	Tarea 1.1: Análisis del problema y extracción y formulación de los requisitos.
			Tarea 1.2: Aprendizaje sobre las tecnologías a utilizar y estudio de la documentación del proyecto.

			Tarea 1.3: Selección de las herramientas para la implementación del proyecto. Tarea 1.4: Definición de los roles – Definición de los diferentes tipos de usuarios.
Diseño / Desarrollo / Implementación	150	180	Tarea 2.1: Diseño de mockups de las distintas interfases de la que dispone la aplicación. Tarea 2.2: Diseño de la arquitectura de la aplicación Tarea 2.3: Diseño de la arquitectura de la Base de Datos. Tarea 2.4: Implementación de la aplicación
Evaluación / Validación / Prueba	40	30	Tarea 3.1: Comprobación del funcionamiento de la aplicación. Tarea 3.2: Refactorización y corrección de tests que hayan podido fallar. Tarea 3.3: Comprobación final y evaluación de la aplicación.
Documentación / Presentación	50	60	Tarea 4.1: Elaboración de la memoria y documentación. Tarea 4.2: Revisión de los posibles fallos en la memoria y documentación adicional.

Tabla 5.1. Planificación inicial.

Como se puede ver en la tabla 5.2, la duración real de cada fase difiere con la duración estimada, siendo las fases de análisis y de pruebas menor, mientras que, en las fases de desarrollo y de documentación ha sido superior. En el caso de la etapa de implementación, este desajuste se debe a que en un principio se estimó de forma optimista ciertas funcionalidades de la aplicación, que finalmente han resultado más complicadas de implementar.

Por otro lado, en la fase de documentación, se consideró que la elaboración de la memoria no resultaría lo compleja que realmente ha sido, por ello, se ha estado en esta fase más horas de las estimadas.

Así mismo, las fases de análisis y evaluación se consideraron más complicadas de lo que realmente han sido, resultando las horas finales empleadas en cada fase menores que las estimadas.

Capítulo 6 . Herramientas y tecnologías utilizadas

6.1. Herramientas

6.1.1. Microsoft Visual Studio Code

Visual Studio Code es un editor de código fuente creado y lanzado por Microsoft Corporation en 2015. Entre sus ventajas, destaca que es un software de código abierto, gratuito y multiplataforma. Esta herramienta permite a sus usuarios la edición y desarrollo de código fuente, la depuración de aplicaciones y scripts, la gestión de proyectos y carpetas de trabajo, la integración con control de versiones, la automatización de tareas y flujos de trabajos, y programar en diversos lenguajes de programación [14].

6.1.2. GitHub

GitHub es una plataforma de desarrollo colaborativo diseñada para almacenar y gestionar el código fuente de programas diseñados por los usuarios utilizando el sistema de versiones Git [15]. Al almacenar su código en repositorios de esta plataforma, los usuarios pueden presentarlo o compartirlo, realizar un seguimiento y administrar los cambios realizados, permitir que otros usuarios lo revisen y sugieran mejoras, y colaboren en un proyecto compartido [16].

6.1.3. StarUML

StarUML es un software, creado por MKLab, que permite a sus usuarios realizar modelos basados en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture) [17]. Se basa en la versión 1.4 de UML, pero acepta notación UML 2.0 [18].

6.1.4. Figma

Figma es una herramienta web de diseño que permite a sus usuarios crear, colaborar y prototipar interfaces para sus aplicaciones. Las principales ventajas de esta plataforma son: su accesibilidad multiplataforma, que ofrece un sistema de colaboración en tiempo real, que permite el uso total de sus herramientas enfocada a las respuestas para un diseño óptimo y que trabaja con gráficos vectoriales [19].

6.1.5. Postman

Postman es una aplicación que permite a sus usuarios realizar peticiones de una forma sencilla para probar interfaces de programación de aplicaciones (del inglés Application Programming Interface, API) propias o de terceros. Esta herramienta comenzó siendo una extensión, pero con el paso del tiempo ha evolucionado y hoy en día es una aplicación multiplataforma [20].

6.1.6. Google Chrome

Google Chrome es un navegador web creado y lanzado por Google Inc. en 2008. Diseñado con una interfaz sencilla que les permita a sus usuarios navegar por internet de forma simple y rápida. Su código fuente se fundamenta en el proyecto libre Chromium [21].

6.1.7. Android Studio

Android Studio es el entorno de desarrollo integrado (del inglés Integrated Development Environment, IDE) oficial para la plataforma Android. Fue publicado en 2014 por Google, bajo la Licencia Apache 2.0, y está basado en el IDE IntelliJ IDEA Community Edition de JetBrains. Este IDE ofrece un editor de código para crear aplicaciones, soporte para construcción basada en Gradle, una refactorización específica de Android y arreglos rápidos, renderizado en tiempo real, un dispositivo virtual de Android para ejecutar y probar aplicaciones, y una consola de desarrollador [22].

6.1.8. Microsoft Word

Microsoft Word es un programa para el procesamiento de textos creado por Microsoft Corporation en 1983. Este software se encuentra incluido dentro del paquete ofimático Microsoft Office. Entre sus principales funciones se encuentran el tratamiento, redacción y corrección de textos, siendo el formato docx su estándar [23].

6.1.9. Firebase

Firebase es una plataforma, ubicada en la nube, para el desarrollo de programas web y aplicaciones móviles, que usa un cúmulo de herramientas para la creación y sincronización de proyectos. Fue lanzada en 2011 y comprada posteriormente por Google en 2014. Esta plataforma permite a sus usuarios sincronizar fácilmente los datos de sus sistemas sin necesidad de administrar conexiones o escribir una lógica de sincronización compleja, usa la infraestructura de Google, y permite a los desarrolladores crear proyectos sin necesidad de un servidor [24].

6.2. Tecnologías

6.2.1. Git

Git es un sistema de control de versiones distribuido creado por Linus Torvalds en 2005. Su principal objetivo es llevar el historial de los cambios realizados en los archivos de los proyectos, permitiendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio. Este software con el paso del tiempo ha evolucionado y madurado logrando una mejor usabilidad para sus usuarios [25].

6.2.2. JavaScript

JavaScript es un lenguaje de programación interpretado, débilmente tipado, basado en prototipos, multiparadigma, dinámico y con soporte para programación orientada a objetos. Fue desarrollado inicialmente por Brendan Eich en 1995 con el objetivo hacer las páginas web más dinámicas y con una mejor interfaz de usuario, pero hoy en día este lenguaje no solo puede ejecutarse en navegadores, también en servidores o cualquier dispositivo que tenga un motor o interprete de JavaScript [26].

6.2.3. NodeJs

NodeJs es un entorno de tiempo de ejecución basado en JavaScript, de código abierto, creado por Ryan Dahl en 2009. Este entorno se ejecuta en el motor V8 de Google, es multiplataforma y asíncrono. Utiliza un sistema de entrada y salida sin bloqueos controlado por eventos que lo hace liviano y eficiente [27].

6.2.4. MongoDB

MongoDB es un sistema de gestión de base de datos NoSQL, de código abierto, que usa documentos con un esquema dinámico, permitiendo que el almacenamiento de los datos y su posterior consulta sea más simple [28]. Esto permite crear un entorno altamente escalable para aplicaciones y servicios multiplataforma. Además, se puede usar tanto en Windows, como en GNU/Linux, OS X o Solaris [29].

6.2.5. React Native

React Native es un framework de programación de aplicaciones multiplataforma basado en JavaScript y ReactJS, de código abierto creado por Meta Platforms, Inc. [30]. Este framework no utiliza WebViews, como si lo hacen soluciones como Ionic o Cordova, en este caso genera una aplicación que utiliza

JavaScript Core para ejecutar el código escrito, con un rendimiento similar al nativo [31].

6.2.6. Expo

Expo es un framework de React Native, de código abierto, diseñado para desarrollar aplicaciones nativas de Android, IOS y Web, utilizando un único código base. Su kit de desarrollo de software (del inglés Software Development Kit, SDK) permite que las funciones nativas del dispositivo, como la cámara, los sensores hardware o el micrófono, sean accesibles [32].

6.2.7. Redux

Redux es una pequeña librería JavaScript con una API sencilla y limitada, diseñada para el manejo del estado de los programas. Fue diseñada por Dan Abramov y Andrew Clark, inspirándose la librería Flux de Facebook, y lanzada públicamente en 2015 [33].

Capítulo 7 . Análisis

7.1. Actores

En el prototipo desarrollado para este proyecto, los diferentes actores que interaccionarán con él serán los clientes del negocio y el empresario, denominado profesional para un mejor entendimiento de su rol. Seguidamente, se detallan las características y funcionalidades de cada uno de los actores.

7.1.1. Cliente

El cliente es el usuario registrado que disfruta de los servicios que la empresa ofrece. Este actor puede:

- Solicitar un presupuesto personalizado.
- Compartir su código de afiliado.
- Compartir la aplicación.
- Ver las ofertas de la empresa que se encuentran activas.
- Ver su progreso en el programa de fidelización de la empresa.

7.1.2. Profesional

El profesional es el usuario registrado que ofrece los servicios de su empresa. Este actor cuenta con las siguientes funcionalidades:

- Ofrecer sus ofertas a los clientes.
- Desactivar las ofertas que ya no se ofrece.
- Clonar ofertas que han estado activas en algún momento.
- Responder a las solicitudes de presupuestos.
- Añadir las facturas de los servicios prestados a los clientes.

7.2. Casos de uso

En este subapartado se muestran los distintos diagramas de casos de uso que esclarecen las funcionalidades que poseen los usuarios de la aplicación.

Entre las distintas funcionalidades que los actores poseen, existen algunas funcionalidades que comparten. Por esto, se ha decidido crear un nuevo actor denominado *Usuario registrado*, del que ambos actores heredan estas funcionalidades compartidas.

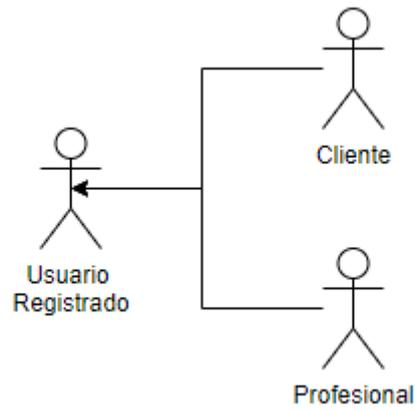


Ilustración 7.1. Diagrama de casos de uso – Actores.

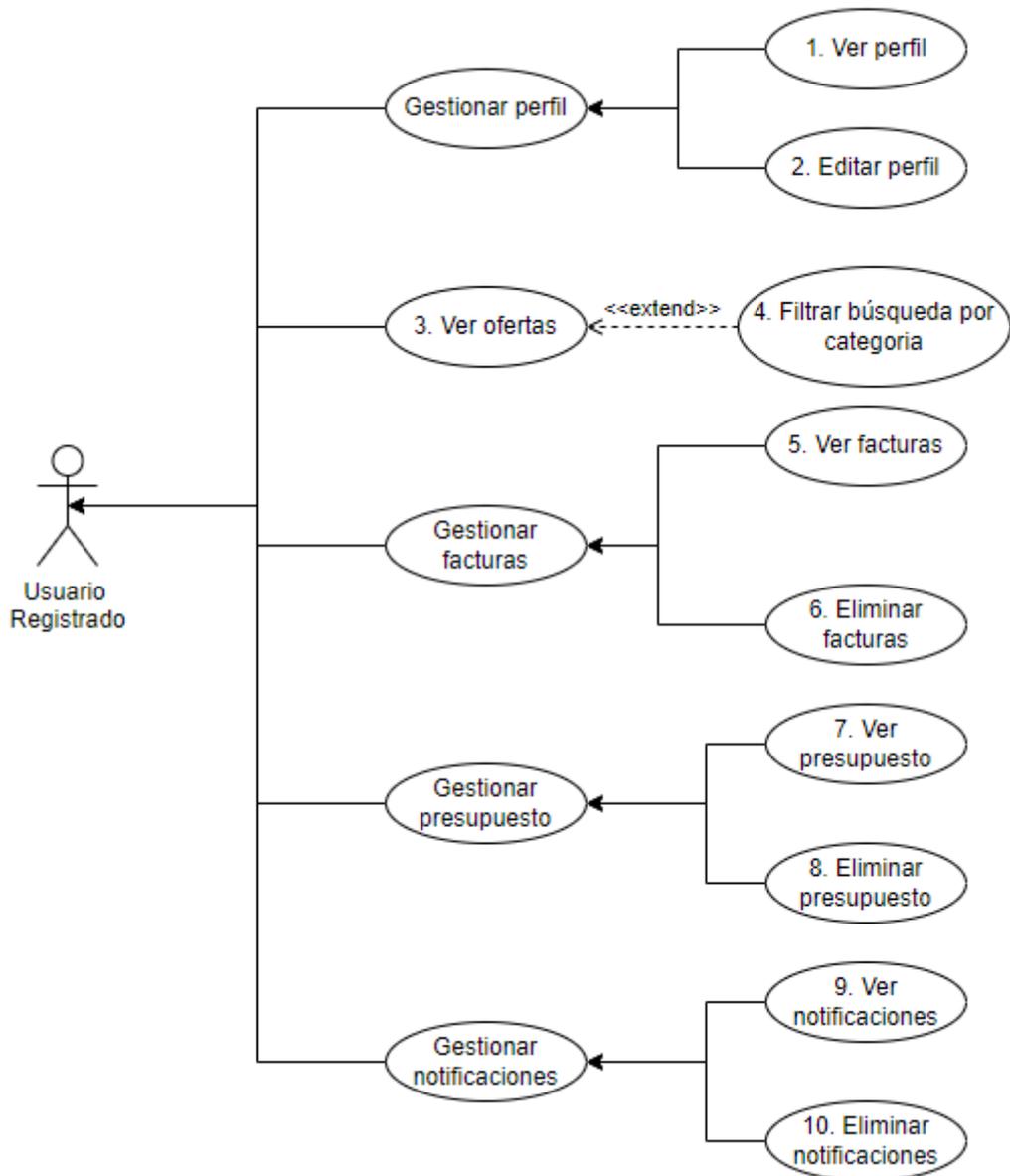


Ilustración 7.2. Diagrama de casos de uso - Usuario registrado.

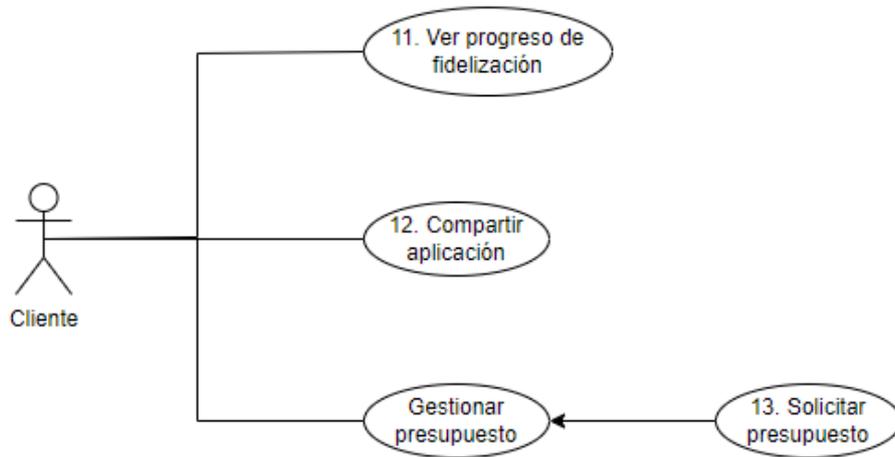


Ilustración 7.3. Diagrama de casos de uso – Cliente.

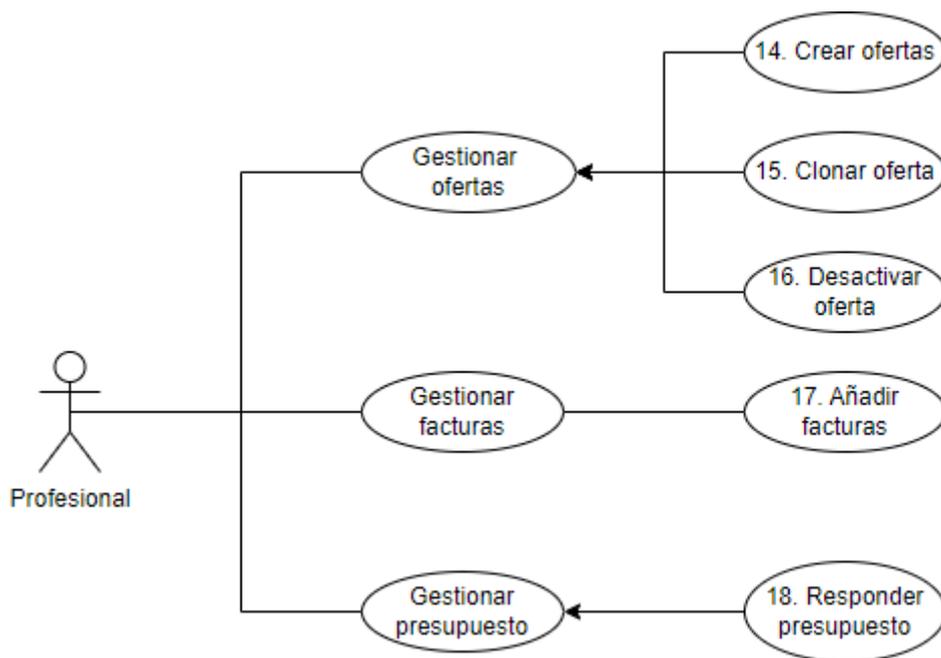


Ilustración 7.4. Diagrama de casos de uso – Profesional.

A continuación, en la tabla 2 se puede ver una descripción de los casos de usos presentados anteriormente.

ID	Caso de uso	Actor	Descripción
1	Ver perfil	Usuario Registrado	Se enseña al usuario su perfil.
2	Editar perfil	Usuario Registrado	Se modifica el perfil del usuario
3	Ver ofertas	Usuario Registrado	Se muestra al usuario las ofertas del negocio
4	Filtrar búsqueda por categoría	Usuario Registrado	Se muestra al usuario las ofertas según su categoría
5	Ver facturas	Usuario Registrado	Se enseñan al usuario sus facturas
6	Eliminar facturas	Usuario Registrado	El usuario elimina una factura
7	Ver presupuesto	Usuario Registrado	Se muestra al usuario un presupuesto
8	Eliminar presupuesto	Usuario Registrado	El usuario elimina un presupuesto
9	Ver notificaciones	Usuario Registrado	Se enseñan al usuario sus notificaciones
10	Eliminar notificaciones	Usuario Registrado	El usuario elimina una notificación
11	Ver progreso de fidelización	Cliente	Se muestra al usuario su progreso en el programa de fidelización
12	Compartir aplicación	Cliente	El cliente comparte la URL de descarga de la aplicación
13	Solicitar presupuesto	Cliente	El cliente solicita un presupuesto personalizado
14	Crear ofertas	Profesional	El profesional crea una oferta

15	Clonar ofertas	Profesional	El profesional clona una oferta desactivada
16	Desactivar ofertas	Profesional	El profesional desactiva una oferta
17	Añadir facturas	Profesional	El profesional añade una factura de un cliente
18	Responder presupuesto	Profesional	El profesional responde a la solicitud de un presupuesto del cliente

Tabla 7.1. Resumen de los casos de uso.

7.3. Requisitos funcionales

Los objetivos funcionales que se han contemplado para el desarrollo de la aplicación son los siguientes:

- El sistema permitirá a los usuarios no registrados crear una cuenta, proporcionando su correo electrónico, nombre, número de documento de identidad y contraseña.
- El sistema validará los datos introducidos por el usuario al crear su cuenta.
- El sistema permitirá a los usuarios restablecer sus contraseñas.
- El sistema permitirá a los usuarios, previamente registrados, iniciar sesión.
- El sistema permitirá cerrar sesión al usuario.
- El sistema permitirá al profesional crear una oferta.
- El sistema enviará una notificación a los clientes cuando se cree una oferta de su interés.
- El sistema permitirá al profesional desactivar una oferta.
- El sistema borrará las notificaciones de una oferta desactivada.
- El sistema permitirá a los usuarios ver las ofertas activas.
- El sistema permitirá al cliente solicitar un presupuesto personalizado.
- El sistema notificará al profesional cuando reciba la solicitud de un presupuesto.
- El sistema permitirá al profesional responder a una solicitud de presupuesto.

- El sistema notificara al cliente cuando el profesional responda a su solicitud de presupuesto.
- El sistema permitirá a los usuarios ver sus solicitudes de presupuesto.
- El sistema permitirá a los usuarios eliminar un presupuesto.
- El sistema permitirá al profesional añadir la factura de los servicios prestados a un cliente
- El sistema notificará al cliente cuando se añada una factura relacionada con él.
- El sistema permitirá a los usuarios eliminar las facturas relacionadas con ellos.
- El sistema permitirá a los clientes ver cuántos usuarios ha atraído al negocio.
- El sistema permitirá a los clientes ver el descuento que ha obtenido del programa de fidelización.
- El sistema permitirá a los clientes compartir su código de amigo y la URL de descarga de la aplicación.

7.4. Requisitos no funcionales

Los requisitos no funcionales que se han tenido en cuenta para la realización de esta aplicación son los siguientes:

- La aplicación será simple y de fácil uso para los usuarios.
- La aplicación estará disponible para los sistemas móviles Android e IOS.
- El sistema mostrará mensajes informativos a los usuarios en caso de haya un error.
- El sistema cifrará las contraseñas de los usuarios.
- El sistema tendrá un manual de uso para los usuarios.
- El sistema otorgará un identificador único a cada usuario.
- El sistema deberá mantener los datos almacenados seguros y protegidos.
- El sistema deberá proporcionar tiempos de respuesta rápidos.

Capítulo 8 . Diseño

8.1. Arquitectura del sistema

Para el desarrollo del proyecto, se ha decidido que la aplicación tenga un diseño arquitectónico basado en el patrón MVC. Este patrón arquitectónico separa los datos y la lógica de negocio de una aplicación de su visualización. Para llevar a cabo esto, el plantea la construcción de tres componentes distintos que son el modelo, el controlador y la vista [34]. Estos componentes se pueden definir de la siguiente manera:

- El modelo representa los datos de la aplicación. Es responsable de gestionar la información, realizar operaciones sobre ella y enviar al controlador la información solicitada para que pueda comunicárselos a la vista.
- El controlador se encarga de gestionar las peticiones que realiza el usuario. Este componente actúa como intermediario entre el Modelo y la Vista, solicitando los datos necesarios al Modelo, para manipularlos posteriormente hasta conseguir los resultados deseados y entregárselos a la Vista para que pueda mostrárselos al usuario [35].
- La vista es el componente que muestra los datos de la aplicación en la interfaz gráfica de usuario.

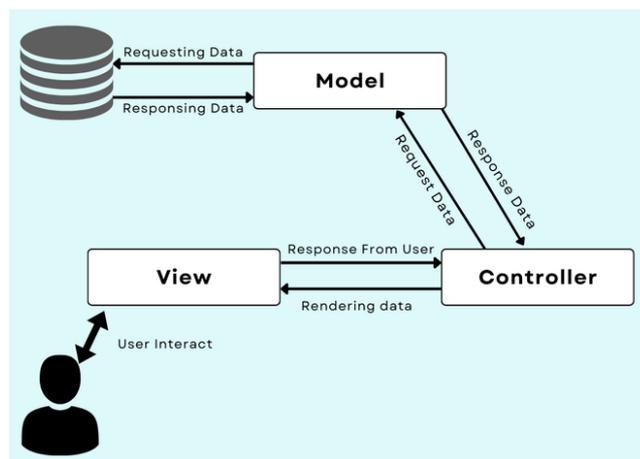


Ilustración 8.1. Patrón de arquitectura de software MVC [36].

Como se puede observar en la ilustración 8.1, el flujo de control que normalmente sigue este patrón sería el siguiente:

1. El usuario interactúa con la vista y realiza una acción.
2. La vista detecta la acción y realiza una petición al controlador.
3. El controlador recibe la petición y solicita o envía los datos al modelo.
4. El modelo realiza las operaciones convenientes en la base de datos y devuelve la información pertinente al controlador.
5. El controlador recibe la información y se la envía a la vista.
6. La vista le muestra al usuario la información.

8.2. Base de datos

Este proyecto hace uso de MongoDB como sistema de Base de Datos para almacenar los datos de la aplicación. MongoDB es una base de datos NoSQL orientada a documentos que se utiliza para almacenar y gestionar datos de una manera flexible y escalable. A diferencia de las bases de datos relacionales tradicionales que utilizan tablas y filas, MongoDB utiliza un modelo de datos basado en documentos BSON lo que permite una estructura de datos más dinámica.

Se ha optado por utilizar este sistema ya que está diseñado para escalar horizontalmente a través del particionamiento de datos, permitiendo distribuir grandes volúmenes de información y cargas de trabajo en múltiples servidores, mejorando el rendimiento y la capacidad de manejo de datos, así como, garantiza una alta disponibilidad y redundancia. Además, ofrece potentes capacidades de consulta y agregación, permitiendo realizar búsquedas ad-hoc, filtrado, ordenación y transformación de datos de manera eficiente.

Aunque la base de datos elegida es NoSQL, a continuación, en la figura 8.2, se puede observar un modelo entidad-relacional creado para mostrar cómo sería de forma aproximada en una base de datos SQL, la estructura de datos del proyecto.

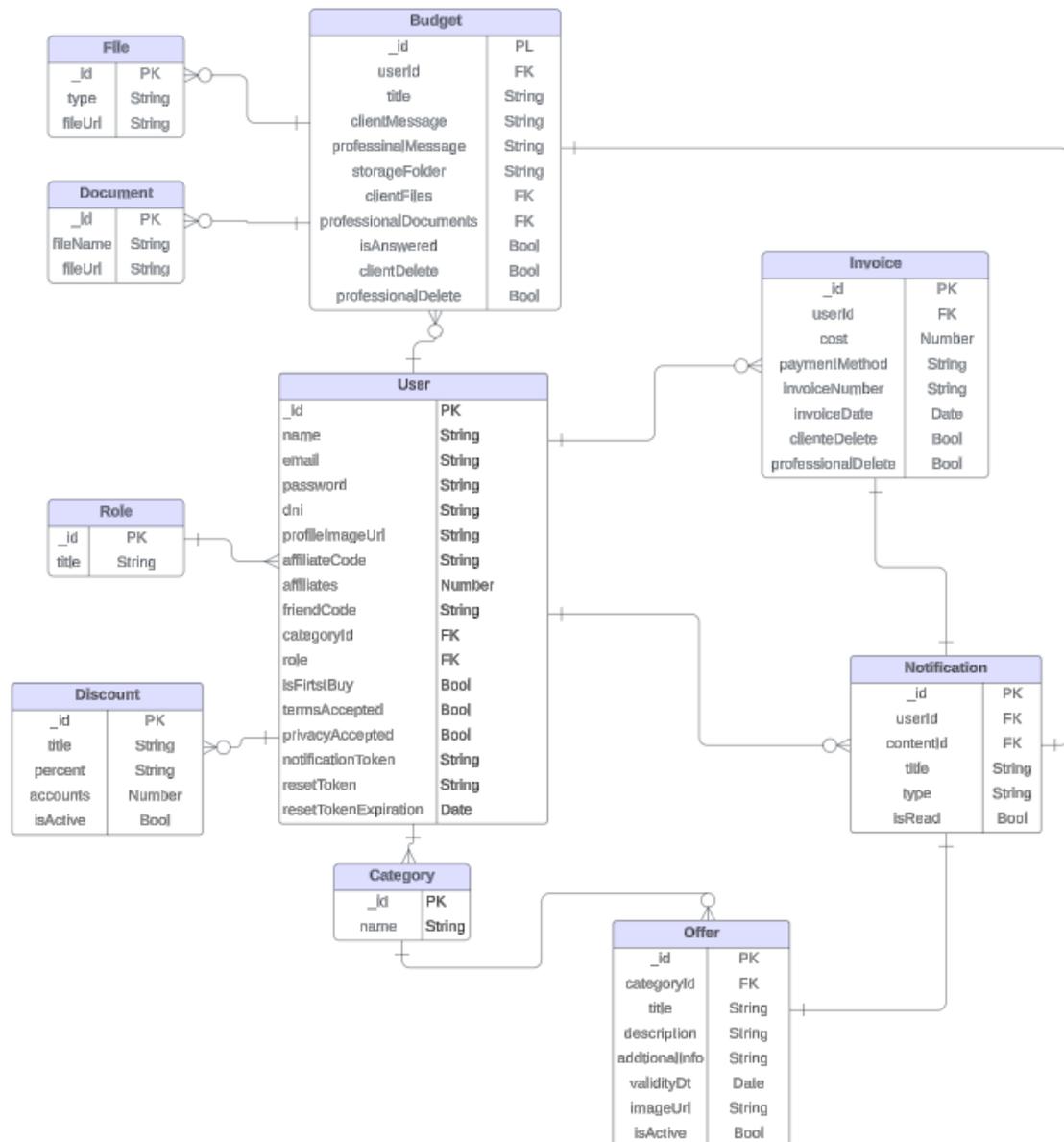


Ilustración 8.2. Diagrama entidad-relación.

8.2.1 Estructura de la base de datos

La base de datos creada para este proyecto cuenta con seis colecciones de datos principales. Estas son:

- **Users**: Esta colección guarda los datos de los usuarios. Entre estos, se encuentran:
 - _id: clave primaria del documento.
 - email: el correo electrónico del usuario.
 - name: nombre completo del usuario.

- password: contraseña del usuario.
- dni: número del documento de identidad del usuario.
- profileImageURL: URL en la que se aloja la imagen de perfil.
- category: las distintas categorías de interés del usuario.
- affiliateCode: código de afiliado del usuario.
- affiliates: número de personas atraídas por el usuario que han realizado una compra.
- isProfessional: propiedad con la que se identifica si el rol del usuario es cliente o profesional.
- isFirstBuy: propiedad que indica si el usuario ha realizado una compra en el negocio con anterioridad.
- termsAccepted: propiedad que indica si el usuario ha aceptado los términos y condiciones.
- privacyAccepted: indica si el usuario a aceptado la política de privacidad.
- resetToken: token para restablecer la contraseña.
- resetTokenExpiration: fecha en la que el resetToken deja de ser válido.
- Invoices: Este modelo almacena los datos de las facturas. Cada documento tiene los siguientes campos:
 - _id: clave primaria del documento.
 - userId: id del usuario al que pertenece la factura.
 - cost: Valor del subtotal de la factura sin impuestos.
 - invoiceDate: fecha en la fue realizada la factura.
 - paymentMethod: método de pago de la factura.
 - invoiceNumber: número de factura.
 - clientDelete: esta propiedad registra si el cliente ha eliminado la factura o no.
 - professionalDelete: este campo indica si el profesional ha eliminado la factura o no.
- Notifications: Colección que almacena las notificaciones locales de los usuarios. Sus propiedades son las siguientes:
 - _id: clave primaria del documento.
 - userId: id del usuario al que pertenece la notificación.

- contentId: id del contenido al que refiere la notificación.
- title: título de la notificación.
- type: propiedad que especifica el tipo de la notificación. Pudiendo ser 'message', 'offer' o 'invoice'.
- isRead: propiedad que indica si la notificación ha sido leída o no.
- Offers: Modelo que almacena los datos de las ofertas del negocio, cuyos campos son:
 - _id: clave primaria del documento.
 - title: título de la oferta.
 - description: descripción de la oferta.
 - additionalInfo: información adicional de la oferta.
 - category: categorías a las que pertenece la oferta.
 - validityDt: fecha hasta la que la oferta es válida.
 - imageUrl: URL en la que se almacena la imagen de la oferta.
 - isActive: campo que indica si la oferta está activa o no.
- Discounts: Donde se almacena la información de los descuentos activos del programa de fidelización. Los documentos de esta colección poseen las siguientes propiedades:
 - _id: clave primaria del documento
 - title: título del descuento.
 - percent: porcentaje de descuento que se aplica.
 - accounts: número de clientes atraídos necesarios para que se aplique el descuento.
 - isActive: propiedad que indica si el descuento se encuentra activo o no.
- Budgets: En esta colección se almacena los datos de los presupuestos solicitados. Siendo estos los siguientes:
 - _id: clave primaria.
 - userId: id del cliente que ha solicitado el presupuesto.
 - title: título del presupuesto.
 - clientMessage: descripción del problema que tiene el cliente.
 - professionalMessage: respuesta del profesional.

- storageFolder: carpeta en la que se almacena los archivos relacionados con el presupuesto.
- clientFiles: sub-colección que almacena los datos de cada archivo subido por el cliente, siendo estos: el tipo del archivo (video o imagen) y la URL de donde se almacena el archivo.
- professionalFiles: sub-colección que almacena los datos de cada documento subido por el profesional, siendo estos: el nombre del documento y la URL de donde se guarda.
- isAnswered: campo que indica si el profesional ha respondido la solicitud o no.
- clienteDelete: propiedad que indica si el cliente ha eliminado el presupuesto o no.
- professionalDelete: campo que indica si el profesional ha eliminado el presupuesto o no.

8.3. Interfaz gráfica

Para el diseño de la interfaz de usuario se ha llevado a cabo un enfoque centrado en la usabilidad, donde el principal objetivo es presentar al usuario un entorno limpio e intuitivo, orientado a las necesidades del usuario.

Con esto en mente, se ha planteado una aplicación que evita introducir elementos gráficos innecesarios que puedan suponer una distracción para el usuario o, en el peor de los casos, su desconcierto. Para evitar esto, se ha tomado como guía una combinación de diferentes sistemas de diseño como pueden ser Material Design (Google), Spectrum (Adobe), o Carbon Design (IBM), donde se establecen las bases para llevar a cabo un diseño no solo elegante y visualmente agradable sino también coherente y, en consecuencia, intuitivo y satisfactorio para el usuario.

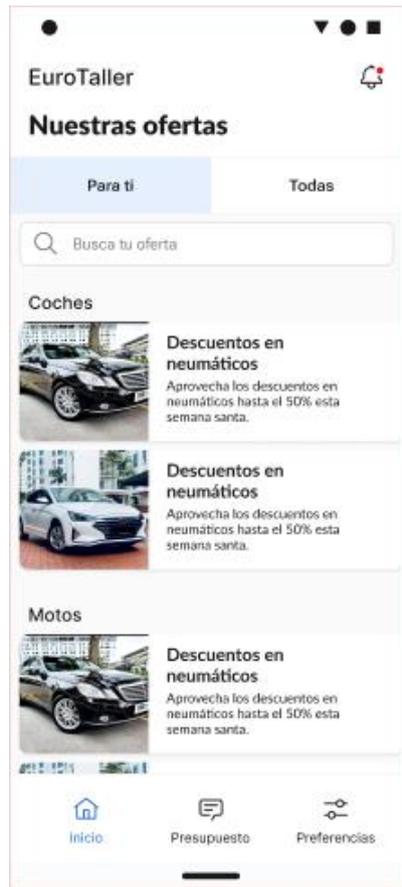


Ilustración 8.3. Mockup pantalla Inicio.

Para ello, se ha establecido una paleta de colores minimalista compuesta de:

- Un color primario, azul #3979FF. Utilizado únicamente cuando es necesario, siendo el principal encargado de ayudar al usuario a identificar determinados elementos y su función. Así, se utiliza en botones, en el tab activo en la barra de navegación y en elementos no accionables pero que presentan información importante para el usuario como es el caso del spinner que se muestra cuando la aplicación está cargando la página.



Ilustración 8.4. Mockup pantalla Invitaciones.

- Colores neutrales: blanco #FFFFFF, gris #F2F2F2 y negro #1A1A1A. Estos colores, son usados principalmente para los colores de fondo y de texto, a excepción del gris que se utiliza para definir el estado de ciertos componentes (como en los formularios cuando no son editables) o como fondo de botones secundarios que no pretenden ser una llamada a la acción, sino que solamente se presentan como una opción para el usuario (como es el caso del botón para cerrar sesión, no queremos incitar al usuario a cerrar sesión). Se ha optado por evitar el uso de la combinación del blanco puro con el negro puro para reducir la fatiga visual de los usuarios.
- Colores semánticos: rojo #EF1414. Se utilizan para ayudar al usuario a identificar estados, como es el caso de los estados de error en los formularios cuando la información o datos introducidos no son válidos.

EuroTaller

Regístrate

Introduce tus datos para crear tu cuenta

Nombre completo

Email

Documento de identidad

Contraseña

La contraseña debe incluir:

- 8 caracteres o más
- Una mayúscula
- Una minúscula
- Un número.

Confirmar contraseña

Las contraseñas no coinciden

Registrarse

¿Ya tienes una cuenta? [Regístrate](#)

Ilustración 8.4. Mockup pantalla de registro.

Partiendo de estos principios se plantearon los Mockups de la aplicación a través de la herramienta de diseño Figma, que permitió la rápida conceptualización e iteración de cada una de las pantallas de la aplicación.

Capítulo 9 . Desarrollo

Para el desarrollo de este trabajo se ha optado por utilizar NodeJS junto con Express para la implementación del Backend, mientras que, para la implementación de la interfaz de usuario, se ha usado la librería React Native y el framework Expo, que está basado en ésta.

9.1. Implementación del Backend

9.1.1 Estructura del directorio raíz

La estructura resultante del directorio raíz del Backend del proyecto es la mostrada en la ilustración 9.1.

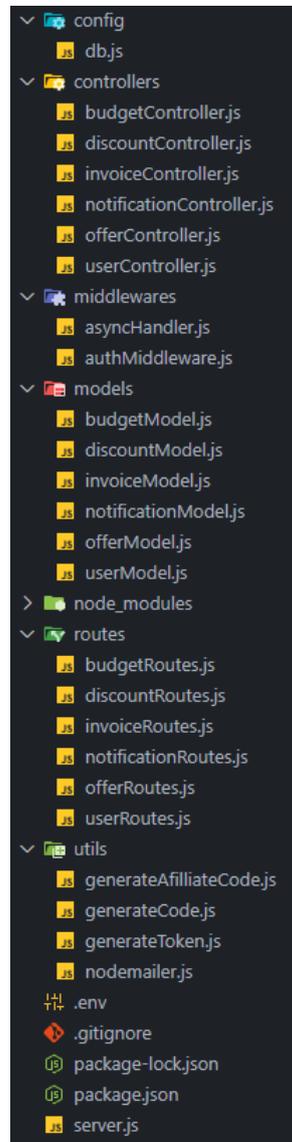


Ilustración 9.1. Estructura del directorio del Backend.

Como se puede observar en este directorio se encuentran los siguientes subdirectorios y archivos:

- `/config`: en este directorio se ubican los archivos de configuración.
- `/controllers`: dentro de esta carpeta están los controladores correspondientes para cada modelo de la base de datos.
- `/middlewares`: carpeta en la que se encuentran los middlewares creados para un mejor funcionamiento del sistema.
- `/models`: contiene todos los esquemas de los modelos de la base de datos.
- `/routes`: en esta carpeta se encuentran definidas las rutas de las peticiones HTTP que recibe el servidor para cada uno de los modelos.
- `/utils`: este directorio contiene funciones necesarias para cubrir ciertas funcionalidades, como el envío de emails o la generación del código de afiliados de los clientes.
- `/node_modules`: esta carpeta contiene todos los módulos instalados para que funcione correctamente el servidor.
- `server.js`: archivo en el que se crea el servidor y se configura Express.

9.1.2. Modelos, controladores y rutas

En este proyecto, para cada uno de los seis modelos de datos definidos, se ha creado su esquema dentro del directorio `/models`, su correspondiente controlador en la carpeta `/controllers`, que contiene toda la lógica necesaria para ese modelo, y el enrutado de las peticiones HTTP concernientes a ese modelo en el directorio `/routes`.

Para el modelo `User`, se ha creado el controlador `UserController` que contiene todas las funciones necesarias para el manejo de la información de los usuarios. Entre estas funciones se encuentran: `signInUser`, `logoutUser` y `registerUser`, para el inicio y cierre de sesión y el registro de nuevos usuarios; `forgotPassword` y `resetPassword`, para el restablecimiento de la contraseña del usuario; `updateUser`, para editar la información del perfil del usuario; `getUsersList`, para devolver una lista de todos los clientes mostrando su

nombre y el dni, `getProfile`, para responder con la información del perfil del usuario; y `getUserInvitation`, para obtener la información del cliente necesaria para el programa de fidelización.

```
import express from 'express';
import {
  signInUser,
  registerUser,
  logoutUser,
  getProfile,
  getUsersList,
  updateUser,
  getUserInvitations,
  forgotPassword,
  resetPassword,
} from '../controllers/userController.js';

import { protect, professionalProtect } from '../middlewares/authMiddleware.js';

const router = express.Router();

router.route('/').post(registerUser).get(protect, professionalProtect, getUsersList);
router.post('/signin', signInUser);
router.post('/logout', logoutUser);
router.post('/forgotpassword', forgotPassword);
router.post('/resetpassword', resetPassword);

router.route('/profile').get(protect, getProfile).put(protect, updateUser);

router.route('/invitation').get(protect, getUserInvitations);

export default router;
```

Ilustración 9.2. Archivo userRoutes.

```
// @desc Auth user & get token
// @route POST /api/users/signin
// @access Public
const signInUser = asyncHandler(async (req, res) => {
  const { email, password } = req.body;

  const formattedEmail = email.trim().toLowerCase();

  try {
    const user = await Users.findOne({ email: formattedEmail });

    if (user && (await user.comparePassword(password))) {
      generateToken(res, user._id);

      res.status(200).json({ user });
    } else {
      res.status(401).json({ message: 'Credenciales erroneas' });
      throw new Error('Invalid email or password');
    }
  } catch (error) {
    console.log(error);
  }
});
```

Ilustración 9.3. Función signInUser.

Para el modelo Invoice, se definió el controlador `invoiceController`, que contiene todas las funciones para implementar la lógica necesaria para los datos de facturas. Estas funciones son: `createInvoice` y `deleteInvoice`, para la

creación y eliminación de las facturas; `getAllInvoices`, que devuelve un listado con todas las facturas definidas; `getInvoicesByUserId`, que devuelve un listado con todas las facturas asociadas a un cliente; y `getInvoiceById`, que devuelve los datos de una factura determinada.

```
import express from 'express';
import {
  createInvoice,
  getAllInvoices,
  getInvoicesByUserId,
  getInvoiceById,
  deleteInvoice,
} from '../controllers/invoiceController.js'

import {protect, professionalProtect} from '../middlewares/authMiddleware.js';

const router = express.Router();

router.route('/')
  .post(protect, professionalProtect, createInvoice)
  .get(protect, professionalProtect, getAllInvoices);

router.route('/userinvoices')
  .get(protect, getInvoicesByUserId);

router.route('/:id')
  .get(protect, getInvoiceById)
  .put(protect, deleteInvoice);

export default router;
```

Ilustración 9.4. Archivo invoiceRoutes.

```
// @desc Get an invoice
// @route GET /api/invoices/:id
// @access Private
const getInvoiceById = asyncHandler( async(req, res) => {
  const dateOptions = {
    year: 'numeric',
    month: 'long',
    day: 'numeric',
  }

  try{
    const invoice = await Invoices.findById(req.params.id, '-createdAt -updatedAt').lean();

    if( invoice ){
      const user = await Users.findById(invoice.userId);

      invoice.userName = user.name

      const formattedDate = invoice.invoiceDate.toLocaleDateString('es-ES', dateOptions);
      invoice.invoiceDate = formattedDate;

      res.status(200).send(invoice);
    } else {
      res.status(404).json({ message: 'Factura no encontrada'})
      throw new Error('Invoice not found');
    }
  } catch( error ) {
    console.log(error)
  }
});
```

Ilustración 9.5. Función getInvoiceById.

El modelo Discount cuenta con el controlador `discountController`, en él se encuentran las funciones necesarias para tratar sus datos. Estas funciones

son: `createDiscount`, para la creación de nuevos descuentos; `disableDiscount`, para desactivar un descuento; `getAll`, para obtener el listado de todos los descuentos activos; y `getDiscountById`, que devuelve la información de un descuento en particular.

```
import express from 'express';
import {
  createDiscount,
  getAll,
  getDiscountById,
  disableDiscount
} from '../controllers/discountController.js'

import {protect, professionalProtect} from '../middlewares/authMiddleware.js';

const router = express.Router();

router.route('/')
  .post(protect, professionalProtect, createDiscount)
  .get(protect, professionalProtect, getAll);

router.route('/:id')
  .get(protect, professionalProtect, getDiscountById)
  .delete(protect, professionalProtect, disableDiscount);

export default router;
```

Ilustración 9.6. Archivo `discountRoutes`.

```
// @desc   Disable a discount
// @route  Delete /api/discounts/:id
// @access Private/Professional
const disableDiscount = asyncHandler( async(req, res) => {
  try{
    const discount = await Discounts.findById(req.params.id, '-createdAt -updatedAt');

    if( discount ){
      discount.isActive = false;
      discount.save();

      res.status(200).json({message: 'Descuento eliminado.'});
    } else {
      res.status(404).json({ message: 'Descuento no encontrado'})
      throw new Error('Discount not found');
    }
  } catch( error ) {
    console.log(error)
  }
});
```

Ilustración 9.7. Función `disableDiscount`.

El modelo `Notification` tiene el controlador `notificationController`, que cuenta con las funciones para el manejo de sus datos. Estas funciones son: `getNotificationByUserId`, que devuelve un listado de las notificaciones de cada usuario; `updateReadNotification`, que establece si una notificación ha sido leída por el usuario; y `deleteNotification`, que elimina una notificación.

```

import express from 'express';
const router = express.Router();

import {
  getNotificationsByUserId,
  updateReadNotification,
  deleteNotification
} from '../controllers/notificationController.js';

import {protect} from '../middlewares/authMiddleware.js';

router.route('/').get(protect, getNotificationsByUserId);

router.route('/:id')
  .put(protect, updateReadNotification)
  .delete(protect, deleteNotification);

export default router;

```

Ilustración 9.8. Archivo notificationRoutes.

```

// @desc Update read Notification
// @route PUT /api/notifications/:id
// @access Private
const updateReadNotification = asyncHandler( async(req, res) => {

  try{
    const notification = await Notifications
      .findById(req.params.id)
      .select("-createdAt -updatedAt");

    if( notification ){
      notification.isRead = true;

      await notification.save();

      res.status(200).send(notification);
    } else {
      res.status(404).json({ message: 'Notificacion no encontrada'})
      throw new Error('Notification not found');
    }
  } catch( error ) {
    console.log(error)
  }
});

```

Ilustración 9.9. Función updateReadNotification.

Para el modelo Offer, se ha creado el controlador offerController, que contiene todas las funciones para el tratamiento de los datos de las ofertas. Estas funciones son: createOffer, para crear una oferta nueva; getAllByCategory, que devuelve un listado de todas las ofertas activas desglosadas según la categoría a la que pertenezcan, getAllUserOfferByCategory, que devuelve una lista de las ofertas activas según la/s categoría/s que le interesen al cliente; getAllInactiveOffers, que retorna una lista con todas las ofertas inactivas;

getOfferById, que devuelve la información de una oferta en particular; y deactivateOffer, que desactiva una oferta.

```
import express from 'express';
import {
  createOffer,
  getAllByCategory,
  getOfferById,
  getAllInactiveOffers,
  getAllUserOfferByCategory,
  deactivateOffer
} from '../controllers/offerController.js'

import {protect, professionalProtect} from '../middlewares/authMiddleware.js';

const router = express.Router();

router.route('/')
  .post(protect, professionalProtect, createOffer)
  .get(getAllByCategory);

router.route('/inactive')
  .get(protect, professionalProtect, getAllInactiveOffers);

router.route('/useroffers')
  .get(protect, getAllUserOfferByCategory);

router.route('/:id')
  .get(protect, getOfferById)
  .put(protect, professionalProtect, createOffer);

router.route('/:id/inactive')
  .put(protect, professionalProtect, deactivateOffer);

export default router;
```

Ilustración 9.10. Archivo offerRoutes.

```
// @desc Get all offers filter by category
// @route GET /api/offers
// @access Private
const getAllByCategory = asyncHandler( async(req, res) => {
  try{
    const offers = await Offers.find({isActive: true})
      .select('-validityDt -isActive -createdAt -updatedAt -additionalInfo');

    if( offers ){
      const cocheCategory = offers.filter(offer => offer.category.includes('Coche'));
      const motoCategory = offers.filter(offer => offer.category.includes('Moto'));
      const camionCategory = offers.filter(offer => offer.category.includes('Camion'));
      const offersByCategories = [
        {
          "name": "Ofertas para coches",
          "offers": [...cocheCategory]
        },
        {
          "name": "Ofertas para motos",
          "offers": [...motoCategory]},
        {
          "name": "Ofertas para camiones",
          "offers": [...camionCategory]
        }
      ]
      res.status(200).send(offersByCategories);
    } else {
      res.status(404).json({ message: 'Ofertas no encontradas'})
      throw new Error('Offers not found');
    }
  } catch( error ) {
    console.log(error)
  }
});
```

Ilustración 9.11. Función getAllByCategory.

Por último, para el modelo Budget, se ha definido el controlador budgetController, cuyas funciones son: createBudget y deleteBudget, para crear y eliminar un presupuesto; getAllBudgets, para retornar un listado con todos los presupuestos; getUserBudgets, que devuelve una lista con los presupuestos asociados al usuario; getBudgetById, que devuelve la

información de un presupuesto; y `updateProfessionalAnswer`, que agrega al presupuesto la respuesta del profesional.

```
import express from 'express';
import {
  createBudget,
  deleteBudget,
  getAllBudgets,
  getBudgetById,
  getUserBudgets,
  updateProfessionalAnswer
} from '../controllers/budgetController.js'

import {protect, professionalProtect} from '../middlewares/authMiddleware.js';

const router = express.Router();

router.route('/')
  .post(protect, createBudget)
  .get(protect, professionalProtect, getAllBudgets);

router.route('/userbudgets')
  .get(protect, getUserBudgets);

router.route('/:id')
  .get(protect, getBudgetById)
  .put(protect, professionalProtect, updateProfessionalAnswer);

router.route('/:id/deletebudget').post(protect, deleteBudget)

export default router;
```

Ilustración 9.12. Archivo `budgetRoutes`.

```
// @desc Get budgets of an user
// @route GET /api/budgets/userbudgets
// @access Private
const getUserBudgets = asyncHandler( async(req, res) => {
  try{
    const budgets = await Budgets.find({userId: req.user._id, clientDelete: false})
      .select('-createdAt -updatedAt');

    if( budgets ){
      res.status(200).send(budgets);
    } else {
      res.status(404).json({ message: 'Presupuestos de usuario no encontrada'})
      throw new Error('User budgets not found');
    }
  } catch( error ) {
    console.log(error)
  }
});
```

Ilustración 9.13. Función `getUserBudgets`.

9.1.3. Utils

En este directorio se encuentran varios archivos que cumplen una determinada función. Estos son: `generateAffiliateCode`, que genera un código aleatorio que sirva como código de afiliación para los clientes; `generateCode`, que genera un el código de verificación para que los usuarios puedan restablecer su contraseña; `generateToken`, que genera el JWT para autenticar a los usuarios; y `nodemailer`, que permite el envío de un correo electrónico al usuario que

quiere restablecer su contraseña, en el que se le proporciona su código de verificación.

```
import nodemailer from 'nodemailer';

const sendMail = async (userEmail, message) => {
  const transport = nodemailer.createTransport({
    host: "smtp.gmail.com",
    port: 465,
    secure: true,
    auth: {
      user: process.env.EMAIL_USER,
      pass: process.env.EMAIL_PASSWORD
    }
  });

  const mailOptions = {
    from: 'eurotalleremail@gmail.com',
    to: userEmail,
    subject: 'Restablecer contraseña EuroTaller',
    text: message
  };

  transport.sendMail(mailOptions, (error, info) => {
    if(error){
      console.log('Error sendMail: ', error);
    } else {
      console.log('Email sent: ', info.response);
    }
  })
};

export default sendMail;
```

Ilustración 9.14. Archivo nodemailer.

9.1.4. Middlewares

Este directorio contiene los middlewares creados y necesarios para un mejor funcionamiento del sistema. Estos son: `asyncHandler`, que permite un mejor control de las promesas; y `authMiddleware`, en el que comprueba si los usuarios están autenticados y si son profesionales o no.

```

import jwt from 'jsonwebtoken';
import asyncHandler from './asyncHandler.js';
import Users from '../models/userModel.js';

//Protect routes
const protect = asyncHandler(async (req, res, next) => {

  //Read the JWT from the cookie
  let token = req.cookies.jwt;

  if( token ){
    try{
      const decoded = jwt.verify(token, process.env.JWT_SECRET);
      req.user = await Users.findById(decoded.userId)
        .select('-name -password -dni -category');
      next();
    } catch (error){
      res.status(401).json({ message: 'No autorizado'});
      throw new Error('No authorized, token failed');
    }
  } else {
    res.status(401).json({ message: 'No autorizado'});
    throw new Error('No authorized, no token');
  }
});

const professionalProtect = (req, res, next) => {
  if(req.user && req.user.isProfessional) {
    next();
  } else {
    res.status(401).json({ message: 'No autorizado'});
    throw new Error('No authorized as profesional');
  }
}

export { professionalProtect, protect };

```

Ilustración 9.15. Middleware authMiddleware.

9.2. Implementación del Frontend

9.2.1. Estructura del directorio

La estructura resultante del directorio del código fuente del Frontend del proyecto es la mostrada en la ilustración 9.16.

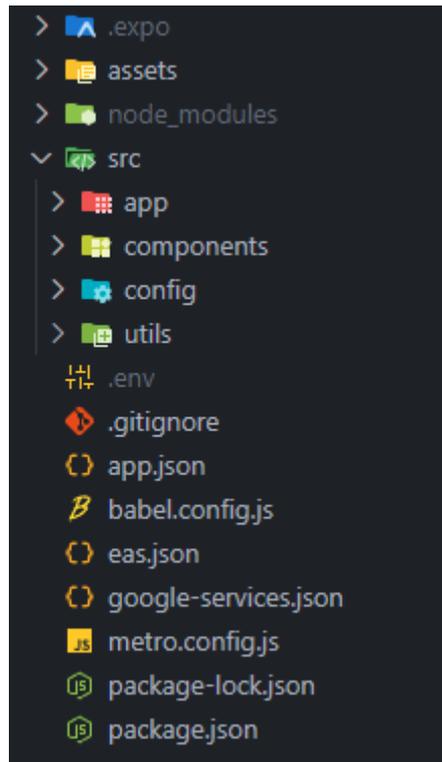


Ilustración 9.16. Estructura del directorio del Frontend.

Como se puede observar en este directorio se encuentra el directorio /src, en el que se ubica todos los archivos desarrollados para la aplicación. Estos son:

- /app: directorio en el que se encuentra las pantallas que conforman la aplicación.
- /components: carpeta en la que se ubican los distintos componentes desarrollados.
- /config: en esta carpeta están ubicados los archivos de configuración.
- /utils: este directorio contiene los archivos relacionados con el manejo de la sesión del usuario, para lo cual se hace uso de Redux.

9.2.2. Principales pantallas y componentes desarrollados

En primer lugar, mencionar las pantallas de `signIn`, `signUp` y `resetPassword`, que son las que el usuario se encuentra cuando no posee una cuenta o no ha iniciado sesión. En ellas se implementan las funcionalidades de inicio de sesión, registro y restablecer la contraseña. Para ello, cada una cuenta con un formulario para que el usuario introduzca la información necesaria para cada funcionalidad, haciendo uso del paquete `Formik` para implementar el

formulario, y el paquete Yup, para la validación de los campos introducidos por el usuario. Cabe destacar que estos dos paquetes mencionados, se usan en el resto de los formularios que posee la aplicación.



Ilustración 9.17. Pantalla de inicio de sesión.

```
const signInSchema = Yup.object().shape({
  email: Yup.string()
    .email('El email no tiene un formato correcto.')
    .required('Este campo no puede estar vacío'),
  password: Yup.string().required('Este campo no puede estar vacío'),
});
```

Ilustración 9.18. Esquema de validación del formulario de inicio de sesión.

Una vez el usuario haya iniciado sesión, es llevado a la pantalla de inicio de la aplicación, en la que puede si es un cliente puede ver las ofertas de las categorías que le interesa o todas, y si es el profesional, puede ver todas las ofertas activas o todas las inactivas. Además, en esta pantalla el profesional puede navegar hasta la pantalla de añadir oferta pulsando el botón “Añadir”. Cabe destacar que, tras iniciar sesión, todas las pantallas poseen un Tab Nav en la parte inferior, el cual posee cuatro Tabs para el profesional y tres Tabs para los clientes. Por último, ambos roles de usuario tienen la posibilidad de navegar hasta el buzón de notificaciones si hacen clic en el botón con forma de campana que se encuentra en la parte superior derecha de la pantalla.



Ilustración 9.19. Tab Nav de cliente.

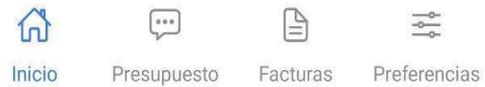


Ilustración 9.20. Tab Nav de profesional.

```
const renderItem = ({ item: category }) => {
  return (
    <View>
      {
        (category.offers.length !== 0) && (
          <>
            <Text style={styles.categoryTitle}>{category.name}</Text>
            <FlatList
              data={category.offers}
              style={{marginBottom: 20}}
              keyExtractor={({ offer } => offer._id}
              renderItem={({ item: offer }) => (
                <OfferCardComponent
                  title={offer.title}
                  id={offer._id}
                  description={offer.description}
                  imageUrl={offer.imageUrl}
                />
              )}
            ></FlatList>
          </>
        )
      }
    </View>
  );
};

return (
  <>
    {
      loading ? (
        <View style={styles.loadingContainer}>
          <ActivityIndicator
            size='large'
            color={'#3979FF'}
          />
        </View>
      ) : (
        <>
          {
            <View style={styles.container}>
              <SearchInput
                searchQuery={searchQuery}
                handleSearch={handleSearch}
              />
              <FlatList
                style={styles.flatlistContainer}
                data={offers}
                renderItem={renderItem}
              ></FlatList>
            </View>
          }
        </>
      )
    }
  </>
);
```

Ilustración 9.21. Render de las ofertas activas

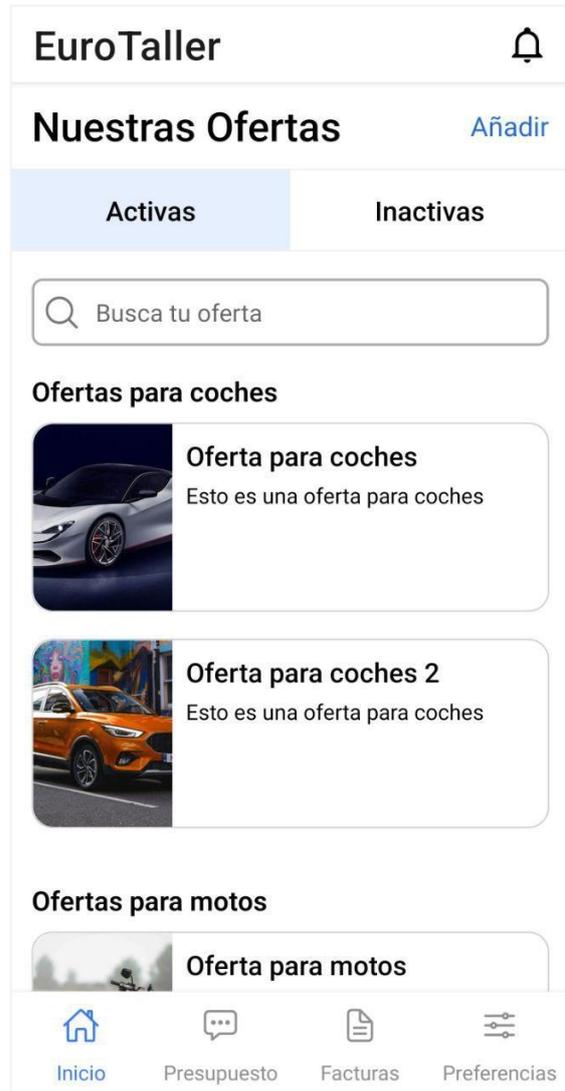


Ilustración 9.22. Pantalla de inicio del profesional.

En la pantalla de presupuesto los usuarios pueden ver un listado de los presupuestos. En el caso del profesional, puede ver todos los presupuestos, mientras que los clientes solo pueden ver sus presupuestos. Además, los clientes tienen la posibilidad de navegar hasta la pantalla de solicitud de

presupuesto, al hacer clic en el botón “Solicitar presupuesto” que se encuentra en la parte inferior de la pantalla, encima del Tab nav.

```
export default BudgetItemComponent = ({ title, clientMessage, id }) => {  
  
  const router = useRouter()  
  
  const onPress = () => {  
    router.push({  
      pathname: `/budgets/${id}`,  
    })  
  };  
  
  return (  
    <View style={styles.container}>  
      <TouchableOpacity style={styles.cardContainer} onPress={onPress}>  
        <View style={styles.textContainer}>  
          <Text style={styles.title} numberOfLines={1}>{title}</Text>  
          <Text style={styles.clientMessage} numberOfLines={2}>{clientMessage}</Text>  
        </View>  
        <View style={styles.buttonContainer}>  
          <AntDesign name='rightcircleo' size={24} color='#2C2A2A' />  
        </View>  
      </TouchableOpacity>  
    </View>  
  );  
};
```

Ilustración 9.23. Componente BudgetItemComponent.

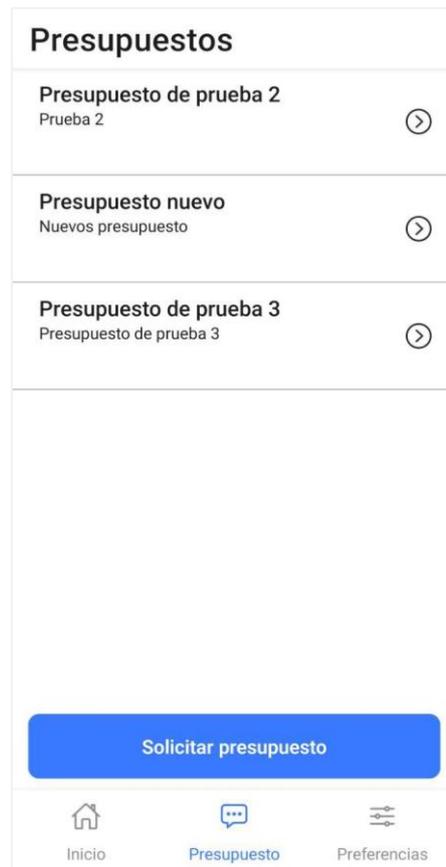


Ilustración 9.24. Pantalla presupuestos del cliente.

En la pantalla de preferencias, a todos los usuarios le aparece la opción de ir a su perfil, en el que puede ver su información, así como modificarla, y les

aparece en la parte inferior de la pantalla, encima del Tab Nav el botón de “logout”. Además, en el caso del profesional, este tiene la opción de Descuentos, en la que puede ver los distintos descuentos que ha creado, hacer clic en cada descuento para ver la información de este, y hacer clic en el botón de “Añadir”, ubicado en la parte superior derecha de la pantalla, para ir hasta la pantalla de creación de un descuento.

The screenshot shows a mobile application screen titled "Datos" with a back arrow on the left and an "Editar" button on the right. The screen displays a profile section with a "Foto de perfil" placeholder. Below the photo, the "Nombre Completo" is "Cliente de prueba". The "Correo electrónico" is "test@gmail.com". The "Documento de identidad" is "54665467F". Under "Categorías", there are two buttons: "Camion" and "Moto". At the bottom, there is a navigation bar with icons for "Inicio", "Presupuesto", and "Preferencias".

Ilustración 9.25. Pantalla del perfil del usuario.

The screenshot shows a mobile application screen titled "Nuevo Descuento" with a back arrow on the left. The screen contains three input fields: "Título", "Invitaciones para aplicar el descuento", and "Porcentaje de descuento aplicado". At the bottom of the form area is a large blue button labeled "Añadir Descuento". Below the form is a navigation bar with icons for "Inicio", "Presupuesto", "Facturas", and "Preferencias".

Ilustración 9.26. Pantalla del formulario de nuevo descuento.

```

const LogoutButton = () => {
  const router = useRouter();
  const [logoutAPICall] = useLogoutMutation();
  const dispatch = useDispatch();

  const onLogout = async () => {
    try {
      await appApi.post('/users/deleteexptoken');

      const res = await logoutAPICall().unwrap();

      dispatch(logout());

      router.replace('/');
    } catch (error) {
      console.log(error);
    }
  };

  return (
    <View style={styles.container}>
      <TouchableOpacity style={styles.button} onPress={onLogout}>
        <Text style={styles.label}>Cerrar sesión</Text>
      </TouchableOpacity>
    </View>
  );
};

```

Ilustración 9.27. Componente LogoutButton.

Por otro lado, los clientes tienen dos opciones más, Invitaciones y Facturas. En la primera puede ver su progreso en el programa de fidelización, así como copiar su código de afiliado y compartir la URL de descarga de la aplicación. En la segunda puede ver un listado de todas las facturas asociadas a él.



Ilustración 9.28. Pantalla de Invitaciones.

```

export default ShareComponent = ({affiliateCode}) => {

  const androidUrl = 'url de play Store'
  const iosUrl = 'url de app Store'
  const message = 'Unete a EuroTaller y usa mi codigo de amigo: ' + affiliateCode + '\n';
  let fullMessage = '';

  if(Platform.OS === 'android'){
    fullMessage = message + androidUrl;
  } else if(Platform.OS === 'ios') {
    fullMessage = message + iosUrl
  }
  const onShare = async () =>{
    try {
      const result = await Share.share({
        message: fullMessage
      })
    } catch (error) {
      console.log(error.message)
    }
  }

  return (
    <View style={styles.container}>
      <TouchableOpacity style={styles.buttonContainer} onPress={onShare}>
        <AntDesign name="sharealt" size={45} color="black" />
        <Text style={styles.shareText}>Compartir</Text>
      </TouchableOpacity>
    </View>
  )
}

```

Ilustración 9.29. Componente ShareComponent.

Por último, como se ha mencionado anteriormente, el Tab de Facturas solo le aparece al profesional, y en su pantalla principal, le aparece un listado con todas las facturas de los clientes que ha añadido. Además, en la esquina superior derecha le aparece el botón “Añadir” que le lleva hasta el formulario para añadir una nueva factura.

```

export default DropdownComponent = ({
  label,
  users,
  setFieldValue,
}) => {
  const [value, setValue] = useState(null);

  return (
    <View style={styles.container}>
      <View>
        <Text style={styles.label}>{label}</Text>
      </View>
      <Dropdown
        style={styles.dropdown}
        placeholderStyle={styles.placeholderStyle}
        selectedTextStyle={styles.selectedTextStyle}
        inputSearchStyle={styles.inputSearchStyle}
        data={users}
        search
        placeholder= ' '
        maxHeight={300}
        labelField= "user"
        valueField= "userId"
        searchPlaceholder= "Buscar cliente"
        value={value}
        onChange={item => {
          setFieldValue('userId', item.userId)
          setValue(item.userId);
        }}
      />
    </View>
  );
};

```

Ilustración 9.30. Componente DropdownComponent.

Ilustración 9.31. Formulario para añadir factura

9.3. Pruebas

Tras finalizar la fase de implementación, se inició la de verificación del prototipo desarrollado, para comprobar su correcto funcionamiento y si cumplía con los requisitos definidos.

Cabe destacar que, durante la fase de implementación, en todo momento las funcionalidades desarrolladas fueron comprobadas y probadas en varios dispositivos virtuales Android.

Para realizar las pruebas de usabilidad y funcionales, se organizó con un grupo de amigos y familiares una reunión en la que pudieron instalar la aplicación en sus dispositivos, crearse una cuenta y probar las distintas funcionalidades implementadas.

Tras esta reunión se les pidió su feedback de la aplicación y se enumeraron una serie de fallos que fueron apareciendo a lo largo del encuentro. Algunos de estos fallos son:

- Cuando el profesional no ha creado ningún descuento a aplicar en el programa de fidelización, a los clientes les seguía apareciendo en el progress bar de la pantalla de Invitaciones que habían completado los requisitos para un descuento.
- A los clientes no se les alertaba cuando habían seleccionado la cantidad máxima de archivos en el formulario de solicitud de presupuesto. Directamente no les dejaba añadir más archivos sin alertarlos del porqué.
- Ciertos componentes se veían con un tamaño inadecuado en dispositivos pequeños.

Estos errores fueron solventados y se organizó una nueva reunión para volver a probar el prototipo. En ella se comprobó que se arreglaron los fallos encontrados y se confirmó que la aplicación funcionaba correctamente.

Lamentablemente, al no contar con un dispositivo IOS durante la realización de las pruebas, no se pudo comprobar que todo funcione correctamente en este sistema operativo.

Capítulo 10 . Conclusiones y trabajo futuro

10.1. Conclusiones

Tras la finalización del proyecto, se ha obtenido una aplicación multiplataforma que cumple los objetivos inicialmente marcados. Se ha desarrollado una herramienta que ayude a una determinada empresa a mejorar la comunicación con sus clientes, y poder recompensarlos cuando promocionen el negocio.

Por un lado, la aplicación permite a los usuarios registrados, ya sean clientes o el profesional, las siguientes funcionalidades:

- Iniciar sesión haciendo uso del correo electrónico y la contraseña proporcionados en el registro.
- Restablecer la contraseña si se da el caso de que no la recuerden.
- Ver y modificar sus datos personales.
- Ver las ofertas que se encuentran activas.
- Ver y eliminar sus facturas registradas.
- Ver y eliminar sus presupuestos personalizados solicitados.
- Ver y eliminar sus notificaciones.

Por otro lado, el profesional tiene la capacidad de:

- Crear nuevas ofertas.
- Desactivas ofertas que se encuentren activas
- Ver las ofertas que se encuentran inactivas.
- Clonar una oferta que se encuentre inactiva y quiera activar de nuevo.
- Añadir facturas al sistema facturas de los servicios prestados a los clientes registrados.
- Responder a las solicitudes de presupuestos de los clientes.

Por último, en el caso de los clientes, estos pueden:

- Ver su progreso en el programa de fidelización y el descuento que ha obtenido de él.
- Compartir la URL de descarga de la aplicación.
- Solicitar un presupuesto personalizado para el problema que tenga.

Por ello, como se comenta anteriormente, el proyecto cuenta con todos los objetivos propuestos al inicio de este, otorgándoles a los usuarios una aplicación con una interfaz sencilla que les sea fácil de entender y usar. Pudiendo utilizarla en los dos sistemas operativos de dispositivos móviles más usados actualmente, Android e IOS.

En lo que al ámbito personal se refiere, el desarrollo de este trabajo me ha permitido aprender y entender el funcionamiento de tres grandes tecnologías que actualmente son bastantes usadas en el mercado laboral, como son React, React Native y NodeJS. Además, he podido aplicar conceptos aprendidos durante el grado y desarrollar una aplicación desde cero, pudiendo ver y comprender las distintas fases necesarias para ello.

10.2. Trabajos futuros

Como se ha afirmado en la introducción, la veloz evolución de las tecnologías, las redes sociales e internet, han causado que continuamente las personas necesitemos mejoras en lo que ya está inventado o implementado en el área de la programación.

Dicho de otro modo, cualquier proyecto de software constantemente tendrá nuevas actualizaciones y funcionalidades que se pueden agregar con el paso del tiempo. A continuación, se muestra las posibles mejoras y actualizaciones que se han tenido en mente para renovar y completar el proyecto:

- Crear una nueva sección en la que el profesional pueda mostrarle a los clientes el catálogo de los servicios que proporciona y los productos que vende.
- Implementar junto con la funcionalidad anterior un sistema ecommerce que permita a los clientes comprar y pagar directamente desde la aplicación en cualquier momento. Almacenándose la correspondiente factura directamente, sin necesidad de que el profesional tenga que introducirla al sistema.
- Implementar que los usuarios puedan abrir el documento PDF del presupuesto en otra aplicación.
- Implementar el registro e inicio de sesión a través de una cuenta de Google o Facebook.

Capítulo 11 . Bibliografía

- [1] «El Paso 2000,» [En línea]. Available: <https://www.elpaso2000.com/empresa/>.
- [2] DISA, «El Grupo DISA,» [En línea]. Available: <https://www.disagrupo.es/conoce/>.
- [3] «Norauto,» [En línea]. Available: <https://pro.norauto.es/es/quienes-somos/sobre-norauto>.
- [4] Wikipedia, «GNU General Public License,» [En línea]. Available: https://es.wikipedia.org/wiki/GNU_General_Public_License.
- [5] Wikipedia, «Apache License,» [En línea]. Available: https://es.wikipedia.org/wiki/Apache_License.
- [6] Microsoft, «Office 365 Education,» [En línea]. Available: <https://www.microsoft.com/en-us/education/products/office>.
- [7] Wikipedia, «Licencia MIT,» [En línea]. Available: https://es.wikipedia.org/wiki/Licencia_MIT.
- [8] Wikipedia, «Visual Studio Code,» [En línea]. Available: https://es.wikipedia.org/wiki/Visual_Studio_Code.
- [9] Microsoft, «MICROSOFT SOFTWARE LICENSE TERMS,» [En línea]. Available: <https://code.visualstudio.com/license>.
- [10] Wikipedia, «Reglamento General de Protección de Datos,» [En línea]. Available: https://es.wikipedia.org/wiki/Reglamento_General_de_Protecci%C3%B3n_de_Datos.
- [11] D. T. Agency, «Modelo Waterfall o en Cascada,» [En línea]. Available: https://www.dtagency.tech/cursos/metodologias_gestion_proyectos/tema_1-ModeloWaterfall.pdf.
- [12] IONOS, «El modelo en cascada: desarrollo secuencial de software,» [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>.
- [13] Wikipedia, «Desarrollo en cascada,» [En línea]. Available: https://es.wikipedia.org/wiki/Desarrollo_en_cascada.

- [14] arsys, «¿Qué es Visual Studio Code y cuáles son sus ventajas?,» [En línea]. Available: <https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>.
- [15] Wikipedia, «GitHub,» [En línea]. Available: <https://es.wikipedia.org/wiki/GitHub>.
- [16] GitHub, «Acerca de GitHub y Git,» [En línea]. Available: <https://docs.github.com/es/get-started/start-your-journey/about-github-and-git>.
- [17] Wikipedia, «StarUML,» [En línea]. Available: <https://es.wikipedia.org/wiki/StarUML>.
- [18] documentation.HELP, «What is StarUML,» [En línea]. Available: https://documentation.help/StarUML/what_is_staruml.htm.
- [19] admin.impulsoh, «Figma: ¿Qué es y para qué sirve?,» [En línea]. Available: <https://impulsoh.com/que-es-figma-y-para-que-sirve/>.
- [20] Y. Muradas, «Qué es Postman y primeros pasos,» 3 junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-postman/>.
- [21] Wikipedia, «Google Chrome,» [En línea]. Available: https://es.wikipedia.org/wiki/Google_Chrome.
- [22] Wikipedia, «Android Studio,» [En línea]. Available: https://es.wikipedia.org/wiki/Android_Studio.
- [23] Wikipedia, «Microsoft Word,» [En línea]. Available: https://es.wikipedia.org/wiki/Microsoft_Word.
- [24] Wikipedia, «Firebase,» [En línea]. Available: <https://es.wikipedia.org/wiki/Firebase>.
- [25] Wikipedia, «Git,» [En línea]. Available: <https://es.wikipedia.org/wiki/Git>.
- [26] Wikipedia, «JavaScript,» [En línea]. Available: <https://es.wikipedia.org/wiki/JavaScript>.
- [27] J. L. Flores, «Qué es NodeJS y para qué sirve,» 4 septiembre 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-nodejs/>.
- [28] IBM, «¿Qué es MongoDB?,» [En línea]. Available: <https://www.ibm.com/es-es/topics/mongodb>.

- [29] Wikipedia, «MongoDB,» [En línea]. Available: <https://es.wikipedia.org/wiki/MongoDB>.
- [30] Á. J. Martín, «React Native: ¿Qué es y para que sirve este framework de programación?,» 18 junio 2019. [En línea]. Available: <https://openwebinars.net/blog/react-native-que-es-para-que-sirve/>.
- [31] códigofacilito, «Qué es React Native,» [En línea]. Available: <https://codigofacilito.com/articulos/que-es-react-native>.
- [32] S. Panagia y G. Alonzi, «What is Expo and why it matters for app development,» 21 febrero 2022. [En línea]. Available: <https://www.mozestudio.com/journal/what-is-expo-and-why-it-matters-for-app-development/>.
- [33] Wikipedia, «Redux (JavaScript),» [En línea]. Available: [https://es.wikipedia.org/wiki/Redux_\(JavaScript\)](https://es.wikipedia.org/wiki/Redux_(JavaScript)).
- [34] Wikipedia, «Modelo–vista–controlador,» [En línea]. Available: <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>.
- [35] Admin, «Patrón de diseño MVC. ¿Qué es y cómo puedo utilizarlo?,» 2 septiembre 2020. [En línea]. Available: <https://www.easyappcode.com/patron-de-diseno-mvc-que-es-y-como-puedo-utilizarlo>.
- [36] jaydeepsathwara272, «Introduction to Laravel and MVC Framework,» 20 marzo 2024. [En línea]. Available: <https://www.geeksforgeeks.org/introduction-to-laravel-and-mvc-framework/>.

Capítulo 12 . Anexos

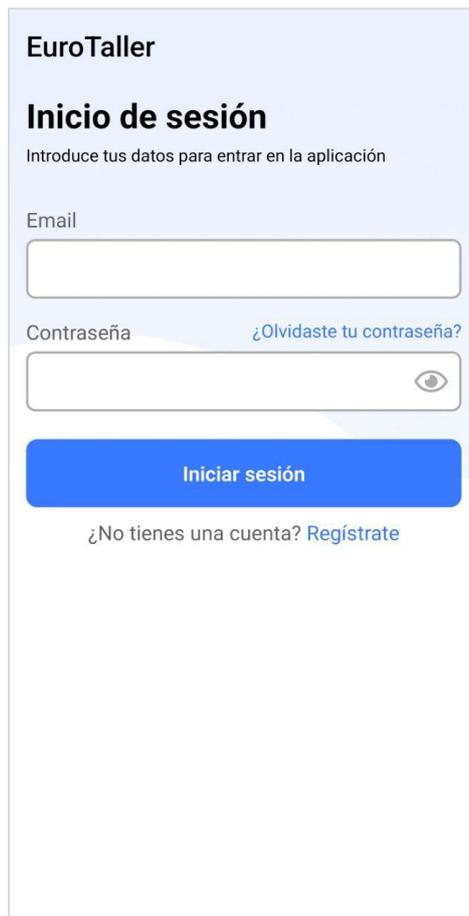
Anexo 1. Manual de Usuario

A1.1. Iniciar sesión

Para comenzar a utilizar la aplicación, es necesario iniciar sesión con una cuenta registrada. Para ello, se debe introducir el correo electrónico y la contraseña proporcionados.

En el caso de no disponer de una cuenta, se puede acceder al formulario de registro pulsando en “Regístrate”.

Por otro lado, en caso de no recordar la contraseña, se puede restablecer pulsando en “¿Olvidaste tu contraseña?” y siguiendo los pasos.



EuroTaller

Inicio de sesión

Introduce tus datos para entrar en la aplicación

Email

Contraseña [¿Olvidaste tu contraseña?](#)

Iniciar sesión

[¿No tienes una cuenta? Regístrate](#)

Ilustración 12.1 Iniciar sesión.

A1.2. Registrarse

Para crear una cuenta nueva, se debe rellenar todos los campos obligatorios del formulario y si el usuario lo desea, también puede completar los campos opcionales. Además, es necesario aceptar la política de privacidad y los términos y condiciones de la aplicación.

Por otro lado, desde esta vista, se puede regresar a iniciar sesión pulsando en “Inicia sesión”.

The image displays three sequential screenshots of the EuroTaller registration form. Each screenshot shows the 'EuroTaller' logo and the title 'Regístrate' with the instruction 'Introduce tus datos para crear tu cuenta'.
The first screenshot shows the initial form with the following fields: 'Foto de perfil (opcional)' with a placeholder image and a 'Seleccionar imagen' button; 'Nombre Completo' with a red error message 'Este campo no puede estar vacío.'; 'Email' with a red error message 'Este campo no puede estar vacío.'; and 'Documento de identidad'. A blue 'Registrarse' button and a link '¿Ya tienes una cuenta? Inicia Sesión' are at the bottom.
The second screenshot shows the password fields: 'Contraseña' and 'Confirmar contraseña', both with red error messages 'Este campo no puede estar vacío.'. Below the password fields, a list of requirements is shown: 'La contraseña debe incluir: ·8 caracteres o más ·Una mayúscula ·Una minúscula ·Un número.'. The 'Categorías de interés (opcional)' section has checkboxes for 'Moto', 'Coche', and 'Camion'. A blue 'Registrarse' button and a link '¿Ya tienes una cuenta? Inicia Sesión' are at the bottom.
The third screenshot shows the 'Confirmar contraseña' field with a red error message 'Este campo no puede estar vacío.'. The 'Categorías de interés (opcional)' section has checkboxes for 'Moto', 'Coche', and 'Camion'. The 'Código amigo (opcional)' field is empty. Below it, there are two checkboxes: 'Términos y condiciones' (with a red error message 'Debe aceptar los Términos y Condiciones.') and 'Política de privacidad' (with a red error message 'Debe aceptar la Política de privacidad.'). A blue 'Registrarse' button and a link '¿Ya tienes una cuenta? Inicia Sesión' are at the bottom.

Ilustración 12.2. Formulario de registro.

A1.3. Restablecer contraseña

En el caso de que el usuario no recuerde su contraseña, puede restablecerla introduciendo el correo electrónico con el que se registró. Si el correo electrónico introducido es correcto, le llegará un email en el que se le proporcionará un código de verificación con el que poder completar el restablecimiento de su contraseña. Hay que destacar que este código de verificación solo es válido durante una hora, pasado este tiempo, ya no podrá utilizarlo y habrá que solicitar uno nuevo.

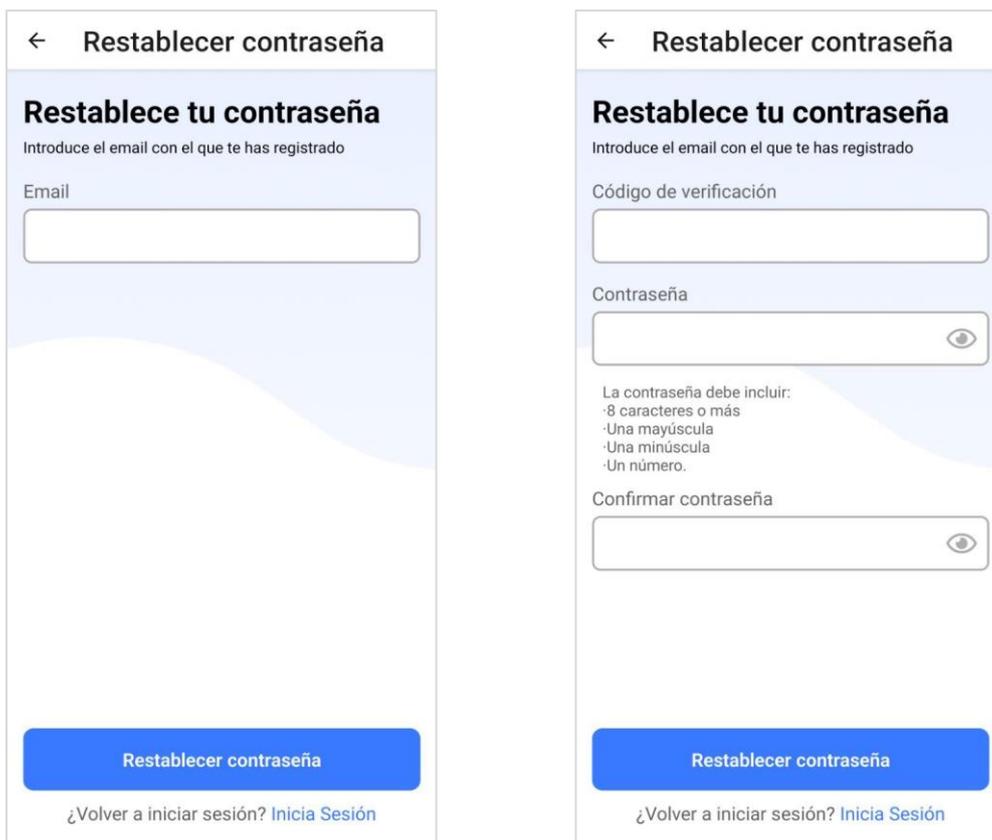


Ilustración 12.3. Restablecer contraseña.

A1.4. Vista de Tab Inicio

Una vez el usuario haya iniciado sesión, accederá a la pantalla de inicio de la aplicación. En esta vista, se le muestra al usuario las ofertas del negocio. En el caso de que el usuario sea profesional, podrá ver todas las ofertas activas y todas las ofertas inactivas, mientras que, si el usuario es cliente, podrá ver todas las ofertas activas o solo las que pertenecen a las categorías que a él le interesan.

Además, en esta pantalla los usuarios podrán acceder al buzón de notificaciones pulsando en la campana situada en la esquina superior derecha. Si tiene notificaciones sin leer, la campana se mostrará con un punto en la parte superior derecha, mientras que, si ha leído todas sus notificaciones, le aparece una campana sin punto.

Por otro lado, en esta pantalla el usuario con rol de profesional podrá además acceder al formulario de creación de una oferta pulsando en “Añadir”.

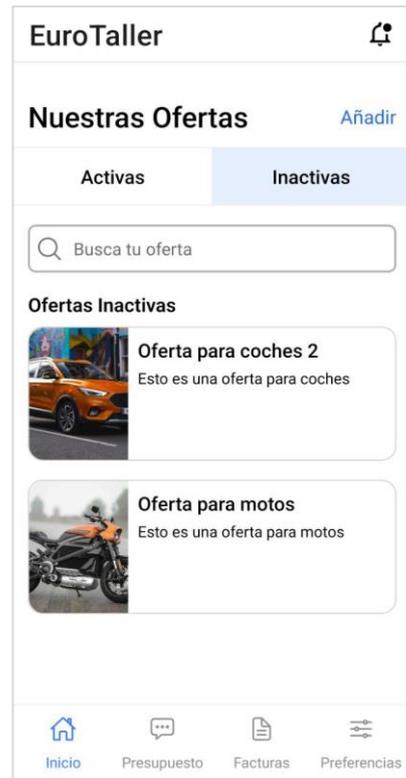
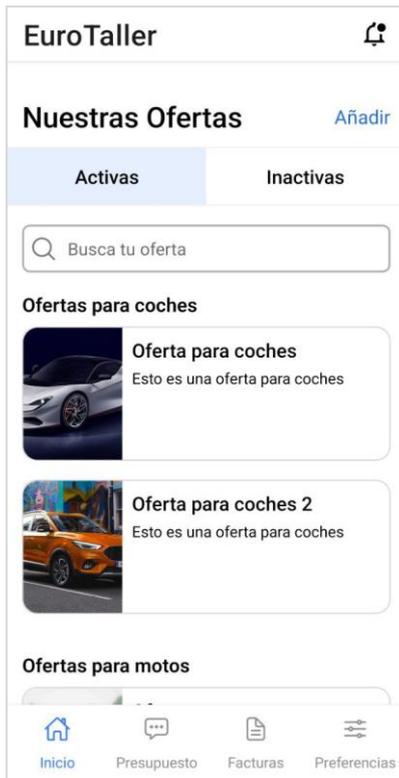


Ilustración 12.4. Vista Inicio – Profesional.

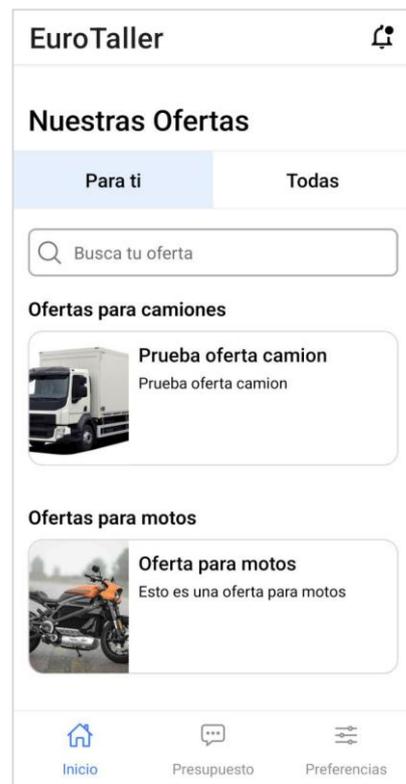
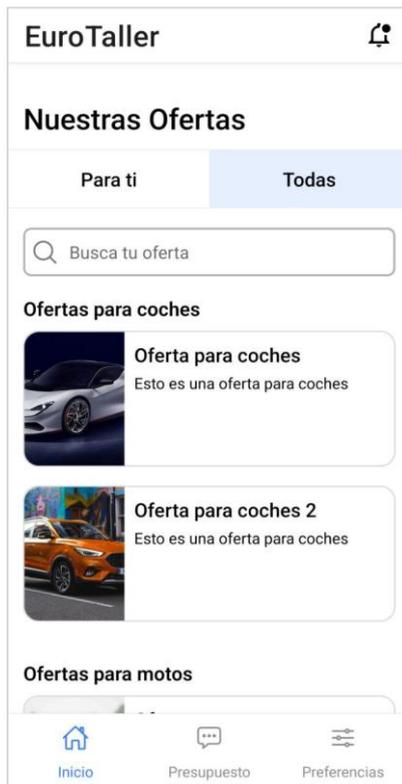


Ilustración 12.5. Vista Inicio – Cliente.

A1.5. Añadir oferta

Para añadir una nueva oferta el profesional deberá rellenar todos los campos del formulario, proporcionando el título, la descripción, la fecha de vencimiento, la imagen, las instrucciones para solicitar la oferta y la/s categoría/s a la/s que pertenece.

The image displays two sequential screenshots of a mobile application interface for adding a new offer. Both screens are titled 'Añadir oferta' and feature a bottom navigation bar with icons for 'Inicio', 'Presupuesto', 'Facturas', and 'Preferencias'. The left screenshot shows a form with the following fields: 'Título de la oferta', 'Descripción', 'Fecha de vencimiento' (with a calendar icon), 'Instrucciones para solicitar la oferta', and 'Imagen'. A blue button labeled 'Añadir Oferta' is positioned below the 'Imagen' field. The right screenshot shows the 'Imagen' field with a placeholder image and a button labeled 'Seleccionar imagen'. Below this, there is a section titled 'Categorías a aplicar la oferta' with three checkboxes: 'Moto', 'Coche', and 'Camion'. A blue button labeled 'Añadir Oferta' is also present at the bottom of this screen.

Ilustración 12.6. Añadir oferta.

A1.6. Vista de la oferta

En la pantalla de la oferta se puede observar la información correspondiente a esa oferta.

Por otro lado, si la oferta está activa al profesional le aparece en la parte inferior de la vista, encima del Nav Bar, un botón para desactivar la oferta, mientras que, si la oferta esta inactiva le aparece un botón para clonarla.

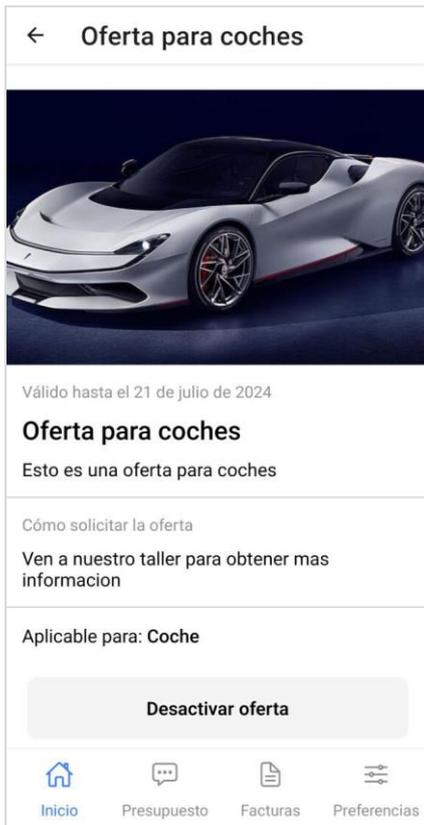


Ilustración 12.7. Vista de una oferta activa - Profesional.

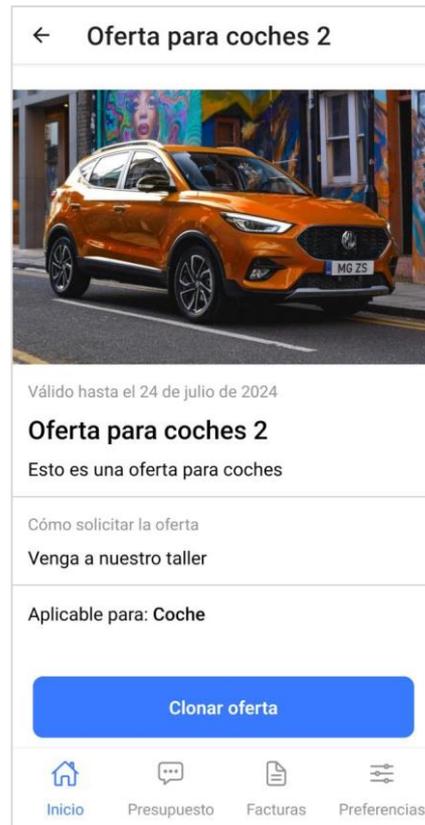


Ilustración 12.8. Vista de una oferta inactiva - Profesional.



Ilustración 12.9. Vista de una oferta – Cliente.

A1.7. Clonar oferta

Para clonar una oferta que se encuentre inactiva, el profesional debe pulsar el botón “Clonar oferta” que se muestra en la ilustración 12.9, que le redirige al formulario para clonar una oferta. En este formulario, el profesional solo tiene que introducir la fecha de vencimiento de la nueva oferta.

Ilustración 12.10 muestra dos pantallas de un formulario de clonación de ofertas. La pantalla de la izquierda muestra la imagen de un coche naranja MG ZS y un botón "Clonar Oferta". La pantalla de la derecha muestra el formulario con campos para el título de la oferta, descripción, fecha de vencimiento, instrucciones para solicitar la oferta, y una imagen de referencia. Ambas pantallas tienen un menú de navegación inferior con opciones: Inicio, Presupuesto, Facturas y Preferencias.

Pantalla 1 (Izquierda):

- Título: Clonar oferta
- Imagen: Imagen de un coche naranja MG ZS.
- Botón: Seleccionar imagen
- Categorías a aplicar la oferta:
 - Moto
 - Coche
 - Camion
- Botón: Clonar Oferta
- Menú de navegación: Inicio, Presupuesto, Facturas, Preferencias

Pantalla 2 (Derecha):

- Título: Clonar oferta
- Título de la oferta: Oferta para coches 2
- Descripción: Esto es una oferta para coches
- Fecha de vencimiento: [Campo vacío]
- Instrucciones para solicitar la oferta: Venga a nuestro taller
- Imagen: Imagen de referencia del coche.
- Botón: Clonar Oferta
- Menú de navegación: Inicio, Presupuesto, Facturas, Preferencias

Ilustración 12.10. Clonar oferta

A1.8. Buzón de notificaciones

En el buzón de notificaciones los usuarios pueden ver las notificaciones que han recibido. Las notificaciones que ya haya leído el usuario tendrán un fondo blanco, mientras que, las notificaciones no leídas lo tendrán azul. Las notificaciones para los clientes pueden ser porque se haya añadido una oferta de una categoría que le interese, porque el profesional haya registrado una factura o porque el profesional le haya respondido a una solicitud de presupuesto. En el caso del profesional, solo recibe notificación si un cliente le ha solicitado un nuevo presupuesto.

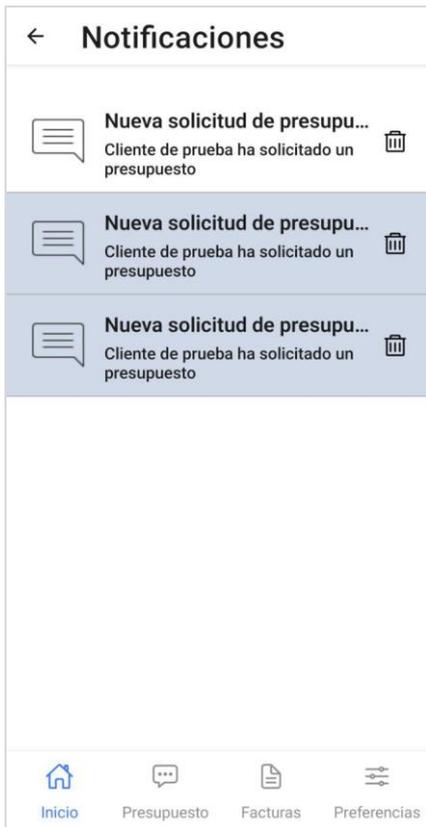


Ilustración 12.11. Buzón de notificaciones - Profesional.

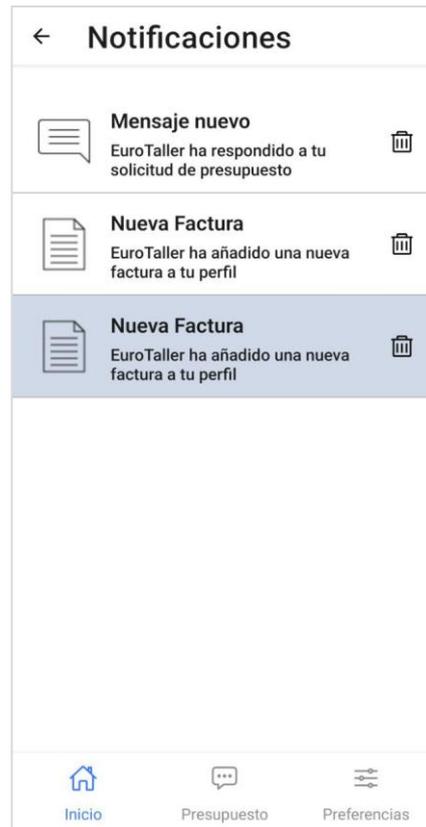


Ilustración 12.12. Buzón de notificaciones - Cliente.

A1.9. Vista de Tab Presupuesto

En esta vista a los usuarios les aparecerá una lista de los presupuestos. En el caso del profesional, le aparecerá todos los presupuestos que todos los clientes hayan solicitado, mientras que a los clientes les aparecerá solo los presupuestos que ellos hayan solicitado.

Además, a los clientes, en la parte inferior de la vista, encima del Nav Bar, les aparece un botón para solicitar un nuevo presupuesto.

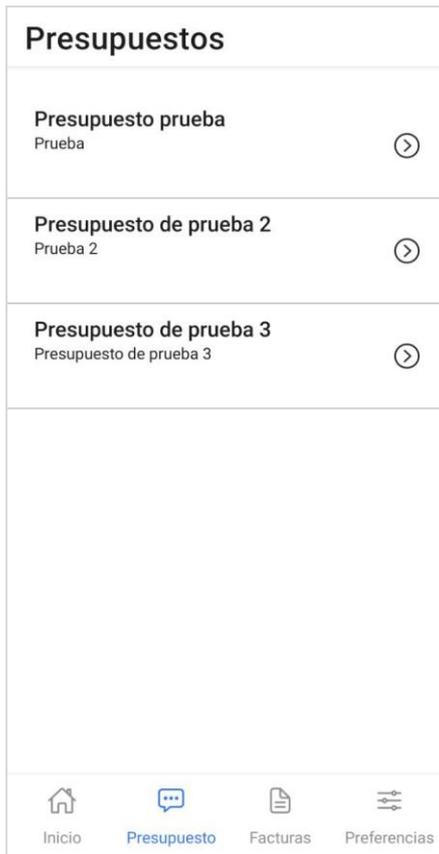


Ilustración 12.13. Vista de presupuesto - Profesional.

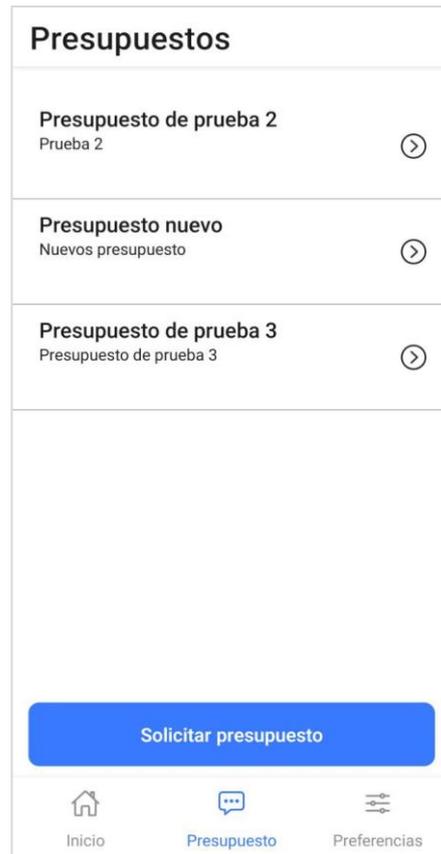


Ilustración 12.14. Vista de presupuesto - Cliente.

A1.10. Solicitar presupuesto

En el formulario para solicitar un nuevo presupuesto, el cliente debe proporcionar un título y una descripción del problema que tiene. Adicionalmente, puede adjuntar hasta seis archivos de foto o videos que tengan un tamaño máximo de 150MB, si se supera estos límites, se le muestra al cliente un mensaje de alerta informándolo de que lo ha superado. Para adjuntar los archivos, debe pulsar el botón “Seleccionar archivos”, que le abrirá un modal con tres botones:

- “Hacer foto o video”: que le abrirá la cámara con la que podrá sacar una foto o un video.
- “Seleccionar de la galería”: que le abrirá su galería de imágenes y videos, en la que podrá seleccionar las fotos o videos.
- “Cancelar”: que cancelará la selección de archivos.

Una vez el cliente haya seleccionado los archivos, estos aparecen listados justo encima del botón “Seleccionar archivos” y tendrán una cruz (X) junto al nombre del archivo (en lado derecho) que, al pulsar en ella, se elimina ese archivo de la selección.



Ilustración 12.15. Solicitar presupuesto.

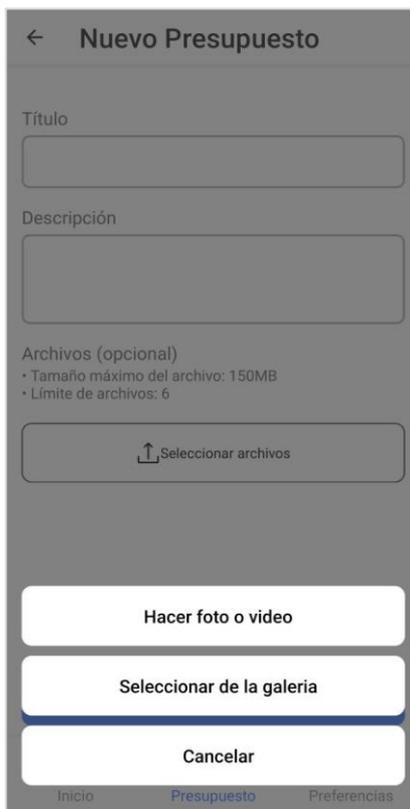


Ilustración 12.16. Seleccionar archivos en solicitud de presupuesto.

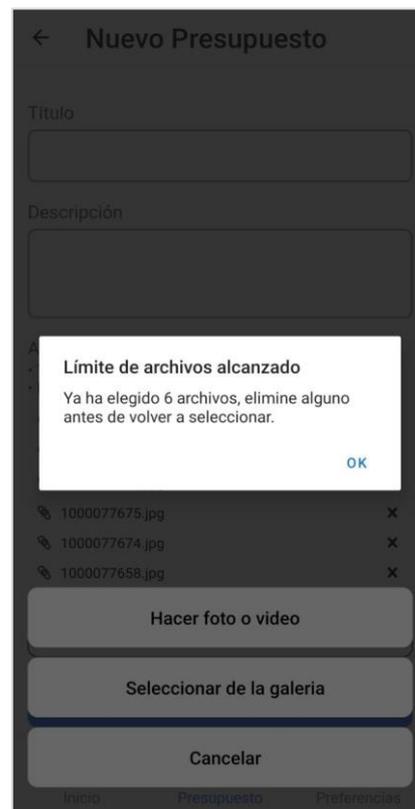


Ilustración 12.17. Límite de archivos alcanzado.

A1.11. Vista de presupuesto

En esta vista, se les mostrará a los usuarios los detalles de cada presupuesto. Una vez el presupuesto haya sido contestado por el profesional, a los usuarios les aparecerá en la esquina superior derecha un botón formado por tres puntos verticales, al pulsar en él, se abre un modal que le permitirá a los usuarios eliminar el presupuesto o cancelar la operación.

Por otro lado, el profesional puede adjuntar archivos PDF en su respuesta. Al pulsar sobre ellos, a los usuarios se les descargará el archivo en cuestión, solicitándoles primero en qué ubicación quieren descargarlo. Una vez descargado se les notifica que se ha descargado correctamente, en el caso de que ocurra un error durante la descarga, también se les notifica.

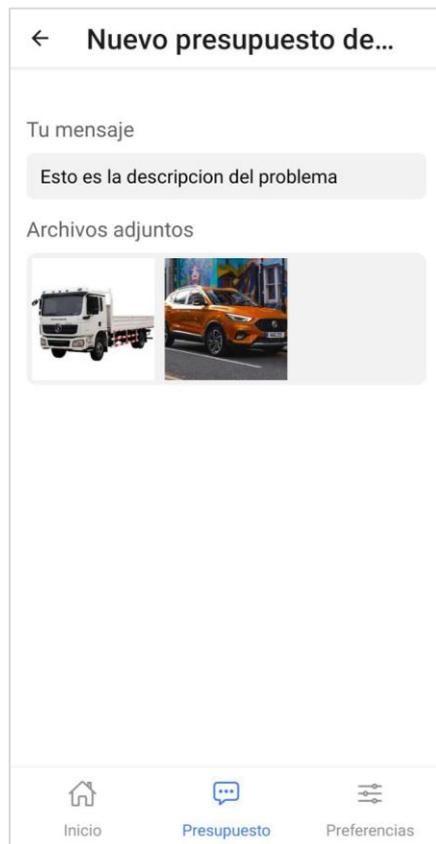


Ilustración 12.18. Presupuesto sin responder.

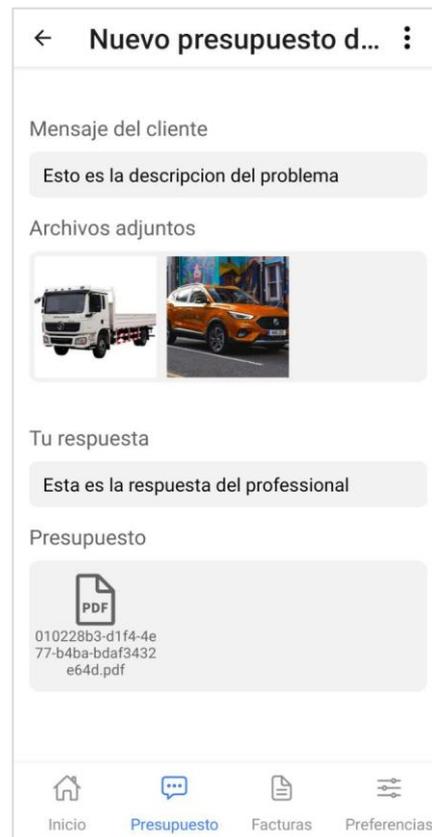


Ilustración 12.19. Presupuesto respondido.

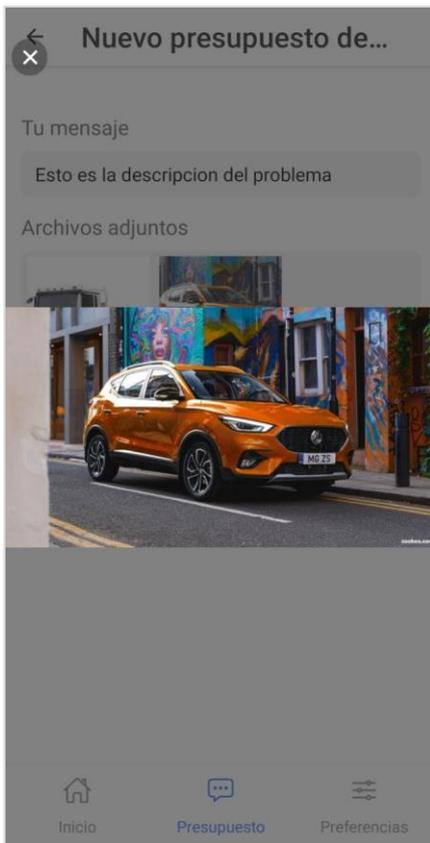


Ilustración 12.20. Ver foto de un presupuesto.

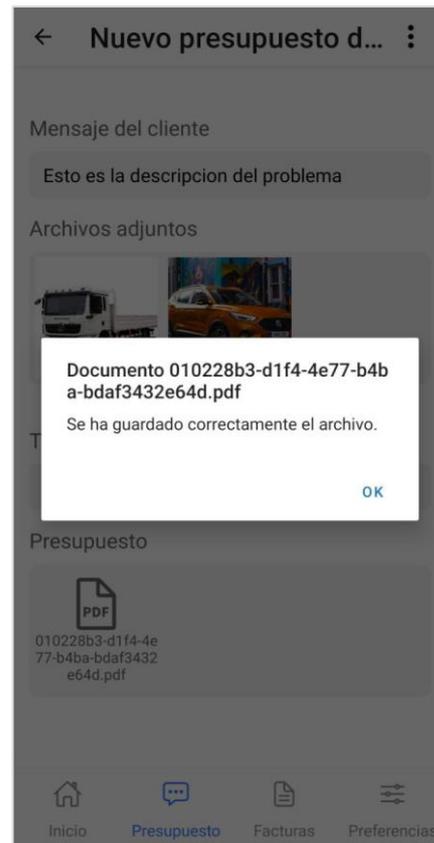


Ilustración 12.21. Archivo PDF descargado correctamente.

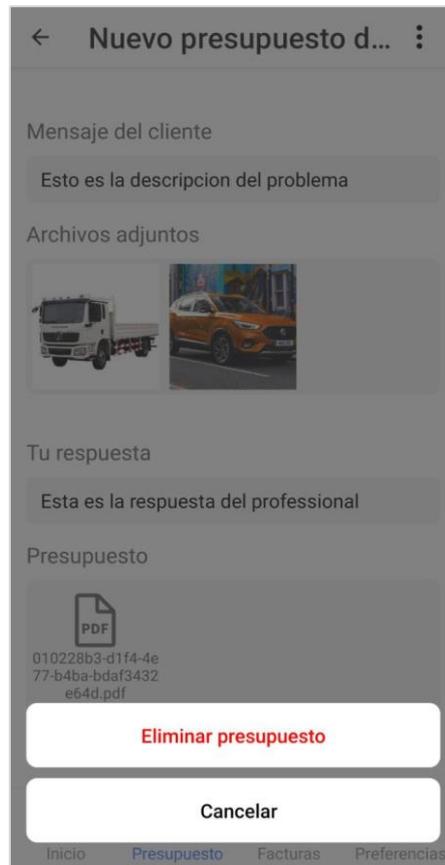


Ilustración 12.22. Eliminar presupuesto.

A1.12. Responder solicitud de presupuesto

El profesional puede responder a las nuevas solicitudes de presupuesto añadiendo su respuesta en el campo “Respuesta a la solicitud”. De forma adicional, al pulsar el botón “Seleccionar archivo”, el profesional puede seleccionar hasta un máximo de dos archivos PDF que tengan un tamaño máximo de 40MB.

← Presupuesto de prueba 3

Mensaje del cliente

Presupuesto de prueba 3

Respuesta a la solicitud

Esto es la respuesta del profesional

Archivos (opcional)

- Tamaño máximo del archivo: 40MB
- Límite de archivos: 2

010228b3-d1f4-4e77-b4ba-bdaf3432e64d.pdf ×

Seleccionar archivo

Enviar respuesta

Inicio Presupuesto Facturas Preferencias

Ilustración 12.23. Responder presupuesto.

A1.13. Vista Tab Facturas

En esta vista exclusiva del profesional, se le muestra una lista con todas las facturas que ha añadido al sistema. Además, tiene la opción de añadir una nueva factura al pulsar sobre “Añadir” en la esquina superior derecha.



Ilustración 12.24. Vista Tab Facturas.

A1.14. Añadir factura

Cuando el profesional pulse el botón “Añadir” que se aprecia en la ilustración 12.24, se le redirige al formulario para añadir una nueva factura. En él deberá introducir el número de la factura, el cliente al que pertenece, el importe subtotal sin impuestos, el método de pago y la fecha en la que se expidió la factura. Una vez haya introducido la información, debe pulsar el botón “Añadir factura”.

← Nueva factura

Número de factura

Cliente

Subtotal sin impuestos

Método de pago

Fecha de la factura

Añadir factura

Inicio Presupuesto Facturas Preferencias

Ilustración 12.25. Añadir factura.

A1.15. Vista de factura - Profesional

En esta vista el profesional puede ver la información correspondiente de cada factura añadida. Además, en la esquina superior derecha tiene un botón con tres puntos verticales que al pulsarlo puede eliminar la factura o cancelar la operación.

← Factura #1111-1111

Fecha de factura: 18 de julio de 2024

Nombre del cliente

Cliente de prueba

Método de pago

Efectivo

Subtotal sin IVA

10,00 €

IGIC 7%

0,70 €

Total

10,70 €

Inicio Presupuesto Facturas Preferencias

Ilustración 12.26. Vista de factura - Profesional.

← Factura #1111-1111

Fecha de factura: 18 de julio de 2024

Nombre del cliente

Cliente de prueba

Método de pago

Efectivo

Subtotal sin IVA

10,00 €

IGIC 7%

0,70 €

Total

10,70 €

Eliminar factura

Cancelar

Inicio Presupuesto Facturas Preferencias

Ilustración 12.27. Eliminar factura - Profesional

A1.16. Vista Tab Preferencias

En esta vista se les muestra a los usuarios varias opciones, dependiendo del rol que tengan. Todos los usuarios tienen la opción de entrar a su perfil, y en él poder acceder al formulario para modificar sus datos. Además, a todos los usuarios les aparecerá en la parte inferior de la vista, encima del Nav Bar, el botón “Cerrar sesión”, con el que podrán cerrar su sesión e inmediatamente se les redirige a la vista de iniciar sesión.

Por otro lado, los clientes tienen también las opciones de “Invitaciones”, en la que podrán ver su progreso en el programa de fidelización del negocio, y “Facturas”, en la que podrán ver sus facturas.

A su vez, el profesional tiene también la opción de “Descuentos”, en la que podrá ver los descuentos que ha añadido al programa de fidelización, así como, añadir nuevos descuentos.

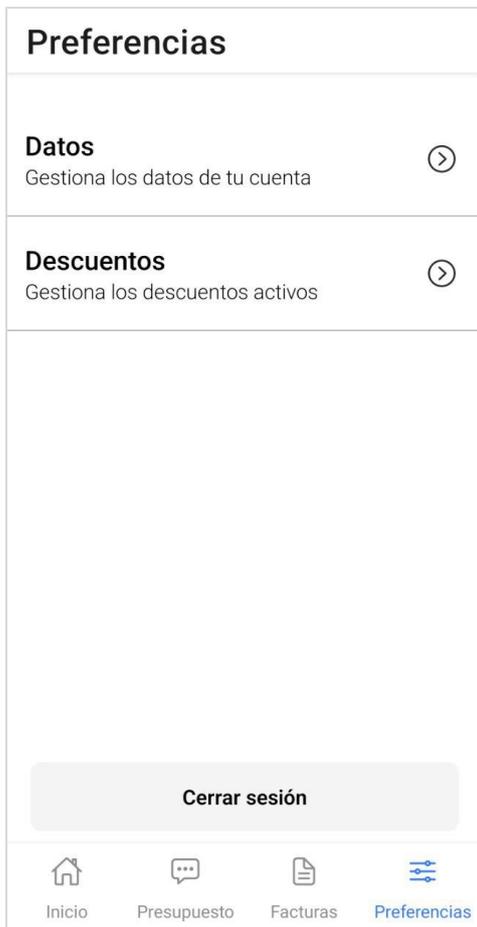


Ilustración 12.28. Vista Tab Preferencias - Profesional.



Ilustración 12.29. Vista Tab Preferencias - Cliente.

A1.17. Perfil del usuario

En esta vista se les muestra a los usuarios sus datos. Además, en la esquina superior derecha les aparece el botón “Editar”, que abrirá una nueva vista con el formulario para editar su información.

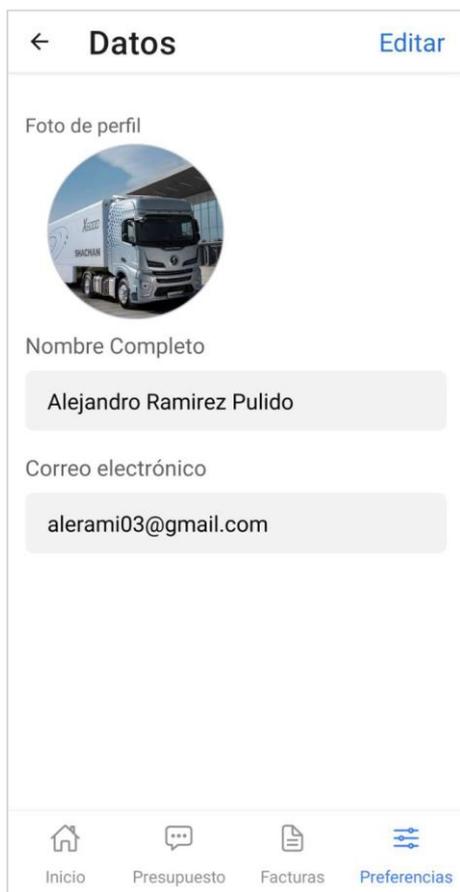


Ilustración 12.30. Perfil de usuario – Profesional

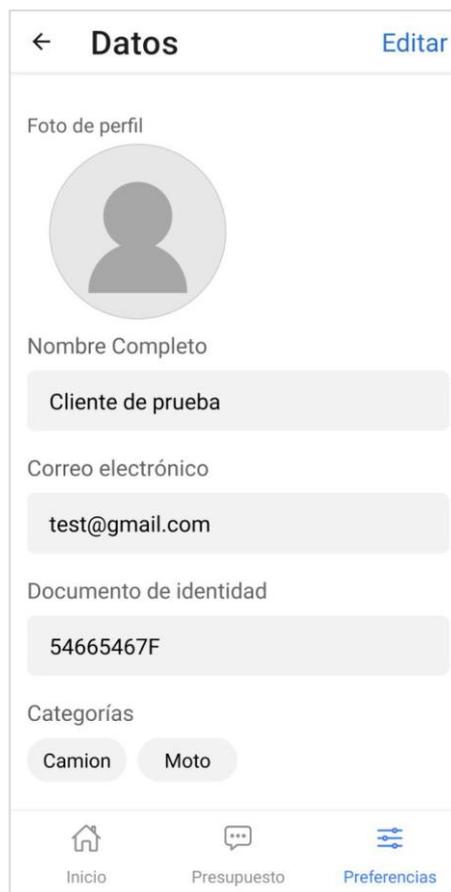


Ilustración 12.31. Perfil de usuario - Cliente.

A1.18. Editar perfil

En esta vista les aparece a los usuarios un formulario con el que pueden modificar sus datos proporcionados. A todos los usuarios les aparecerá los campos de:

- Foto de perfil.
- Nombre completo.
- Contraseña.
- Confirmar contraseña.

No obstante, a los clientes les aparecerá un campo adicional en el que puedan modificar las categorías que le interesan.

Datos

Foto de perfil



↑ Seleccionar imagen

Nombre Completo

Alejandro Ramirez Pulido

Contraseña

La contraseña debe incluir:
 · 8 caracteres o más
 · Una mayúscula
 · Una minúscula
 · Un número.

Confirmar contraseña

Inicio Presupuesto Facturas **Preferencias**

Datos



↑ Seleccionar imagen

Nombre Completo

Alejandro Ramirez Pulido

Contraseña

La contraseña debe incluir:
 · 8 caracteres o más
 · Una mayúscula
 · Una minúscula
 · Un número.

Confirmar contraseña

Guardar cambios

Inicio Presupuesto Facturas **Preferencias**

Ilustración 12.32. Editar perfil - Profesional.

Datos

Foto de perfil



↑ Seleccionar imagen

Nombre Completo

Cliente de prueba

Contraseña

La contraseña debe incluir:
 · 8 caracteres o más
 · Una mayúscula
 · Una minúscula
 · Un número.

Confirmar contraseña

Inicio Presupuesto **Preferencias**

Datos

Nombre Completo

Cliente de prueba

Contraseña

La contraseña debe incluir:
 · 8 caracteres o más
 · Una mayúscula
 · Una minúscula
 · Un número.

Confirmar contraseña

Categorías de interés

- Moto
- Coche
- Camion

Guardar cambios

Inicio Presupuesto **Preferencias**

Ilustración 12.33. Editar perfil - Cliente.

A1.19. Vista Invitaciones

En esta vista exclusiva para los clientes, se les muestra su progreso en el programa de fidelización. Además, pueden ver y copiar su “código amigo” para compartir con otras personas, así como, compartir la URL de descarga de la aplicación.



Ilustración 12.34. Vista Invitaciones.

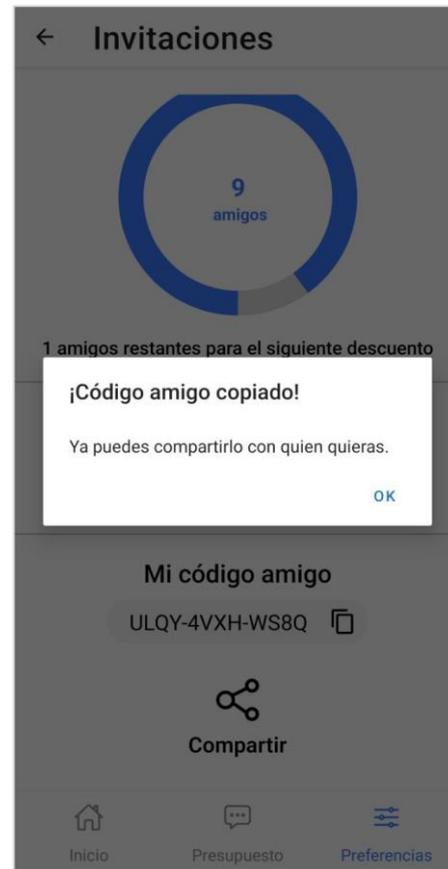


Ilustración 12.35. Alerta código amigo copiado.

A1.20. Vista Facturas – Clientes

En esta vista se les muestra a los clientes un listado con sus facturas añadidas por el profesional.



Ilustración 12.36. Vista Facturas - Cliente.

A1.21. Vista de factura – Cliente

En esta pantalla se les muestra a los clientes los detalles de cada factura asociadas a ellos. Además, en la esquina superior derecha tiene un botón con tres puntos verticales, que al pulsarlo puede eliminar la factura o cancelar la operación.



Ilustración 12.37. Vista de factura - Cliente.

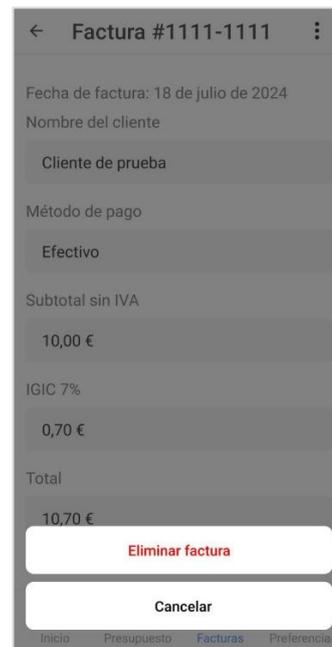


Ilustración 12.38. Eliminar factura - Cliente.

A1.22. Vista Descuentos

En esta vista exclusiva para el profesional se muestra los descuentos para el programa de fidelización que el profesional ha añadido. Además, tiene la opción de añadir un nuevo descuento al pulsar sobre “Añadir” en la esquina superior derecha.

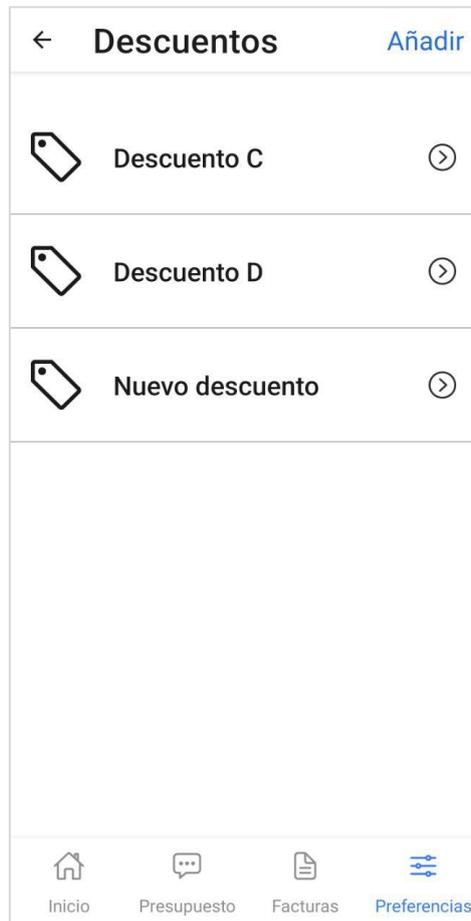


Ilustración 12.39. Vista Descuentos.

A1.23. Añadir descuento

En esta vista se le proporciona un formulario al profesional para que rellene los distintos campos necesarios para añadir un descuento. Una vez ha rellenado el formulario, debe pulsar en el botón “Añadir descuento” que se encuentra en la parte inferior de la pantalla, encima del Nav Bar.

← Nuevo Descuento

Título

Invitaciones para aplicar el descuento

Porcentaje de descuento aplicado

Añadir Descuento

Inicio Presupuesto Facturas Preferencias

Ilustración 12.40. Añadir descuento.

A1.24. Vista de descuento

En esta pantalla se muestra la información correspondiente a cada descuento creado. Así mismo, en la parte inferior de la pantalla se le proporciona al profesional el botón “Eliminar descuento” con el que podrá eliminar el descuento en cuestión al pulsarlo.

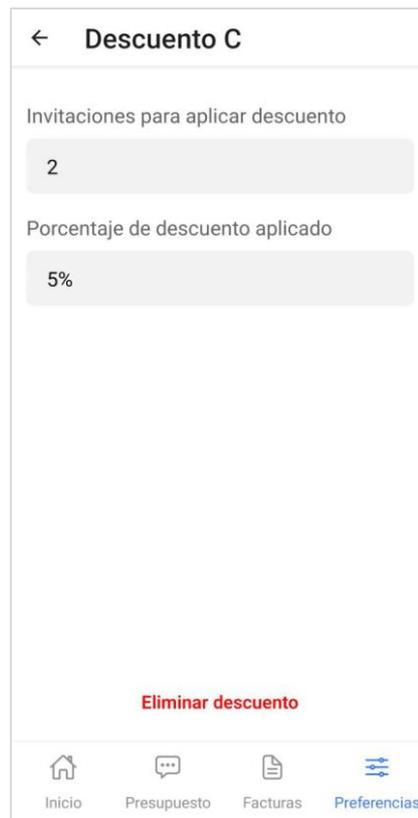


Ilustración 12.41. Vista de descuento.