
**TASKMAKER, UNA APLICACIÓN WEB
PARA REALIZAR GESTIONES Y TAREAS
COTIDIANAS**
**TASKMAKER, A WEB APPLICATION FOR
PERFORMING DAILY TASKS AND TASKS**



**Trabajo de Fin de Grado
Curso 2024–2025**

Nota Final: 9,5

Autor

Javier Sanz Muñoz

Director

Ramón González del Campo Rodríguez Barbero

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

TASKMAKER, UNA APLICACIÓN
WEB PARA REALIZAR GESTIONES Y
TAREAS COTIDIANAS
TASKMAKER, A WEB APPLICATION
FOR PERFORMING DAILY TASKS
AND TASKS

Trabajo de Fin de Grado en Ingeniería Informática

Autor

Javier Sanz Muñoz

Director

Ramón González del Campo Rodríguez Barbero

Convocatoria: *Junio 2025*

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

20 de mayo de 2025

Dedicatoria

A mis padres, por su amor incondicional, su apoyo constante y por enseñarme con su ejemplo el valor del esfuerzo y la perseverancia.

A mis hermanos, por estar siempre a mi lado, animándome en cada paso del camino y compartiendo conmigo cada logro.

Y a mi novia, por su comprensión, paciencia y por ser un pilar fundamental en los momentos más exigentes de este camino.

Agradecimientos

Quiero agradecer a todas las personas que, de una forma u otra, han hecho posible la realización de este Trabajo de Fin de Grado.

En especial, a mis profesores y tutores por su dedicación, por compartir su conocimiento y por acompañarme a lo largo de esta etapa. Su apoyo ha sido fundamental para mi crecimiento académico y personal.

También quiero agradecer a mis compañeros de clase, con quienes he compartido no solo el aprendizaje, sino también grandes momentos durante esta etapa universitaria.

También, agradezco a mis compañeros, con quienes he compartido no solo el estudio, sino también experiencias, aprendizajes y buenos momentos que recordaré siempre.

Por último, gracias a todos lo que, directa o indirectamente, han contribuido con ideas, ánimos o consejos. Vuestra ayuda ha sido clave en los momentos importantes.

Resumen

TASKMAKER, UNA APLICACIÓN WEB PARA REALIZAR GESTIONES Y TAREAS COTIDIANAS

Este Trabajo de Fin de Grado tiene como objetivo el diseño y desarrollo de Taskmaker, una plataforma web orientada a la gestión y contratación de servicios digitales de manera personalizada. La aplicación permite a los usuarios registrarse, navegar por un catálogo de servicios, añadir productos a un carrito, realizar pagos, marcar servicios como favoritos y consultar el historial de pedidos. Todo ello se realiza a través de una interfaz amigable y adaptativa, diseñada para ofrecer una experiencia de usuario fluida y eficiente. El sistema ha sido desarrollado empleando tecnologías como HTML, CSS, JavaScript y PHP, junto con una base de datos MySQL gestionada mediante phpMyAdmin. La arquitectura del proyecto sigue un enfoque cliente-servidor modular, lo que facilita su escalabilidad y mantenimiento. Además, se han implementado funcionalidades clave como la verificación de cuenta por correo electrónico, la recuperación de contraseña y la personalización de preferencias del usuario. El proyecto se ha desplegado en entorno local utilizando XAMPP y ha sido validado mediante pruebas funcionales y de usuario, demostrando su eficacia y su potencial aplicación en contextos reales de digitalización de servicios.

Palabras clave

Taskmaker, Aplicación web, Servicios digitales, Arquitectura cliente-servidor, HTML, PHP, JavaScript, MySQL, Interfaz de usuario, Verificación por correo.

Abstract

TASKMAKER, A WEB APPLICATION FOR PERFORMING DAILY TASKS AND TASKS

The objective of this Final Degree Project is the design and development of Taskmaker, a web platform aimed at the personalized management and contracting of digital services. The application allows users to register, browse a service catalog, add items to a shopping cart, make payments, mark service as favorites, and review their order history. All of this is carried out through a user-friendly and responsive interface designed to provide a smooth and efficient user experience. The system has been developed using technologies such as HTML, CSS, JavaScript, and PHP, alongside a MySQL database managed through phpMyAdmin. The project architecture follows a modular client-server model that supports scalability and maintainability. Additionally, key features such as email account verification, password recovery, and user preference customization have been implemented. The platform was deployed in a local environment using XAMPP and validated through functional and user testing, demonstrating its effectiveness and potential applicability in real-world digital service environments.

Keywords

Taskmaker, Web application, Digital services, Client-server architecture, HTML, PHP, JavaScript, MySQL, User interface, Email verification.

Índice

1. Introducción	1
1.1. Antecedentes	1
1.2. Objetivos	3
1.3. Plan de trabajo	4
1.4. Tecnologías Empleadas	5
1.4.1. HTML (HyperText Markup Language)	5
1.4.2. CSS (Cascading Style Sheets)	5
1.4.3. JavaScript	6
1.4.4. PHP (Hypertext Preprocessor)	7
1.4.5. phpMyAdmin y XAMPP	7
1.5. Conclusión	8
2. Estado de la Cuestión	9
3. Descripción del Trabajo	11
3.1. Registro de Usuario (Register)	11
3.1.1. Pantalla de Bienvenida (Splash Screen)	11
3.1.2. Pantalla de Inicio de Sesión (Login)	11
3.1.3. Formulario de Registro	11
3.1.4. Verificación por Correo Electrónico	13

3.1.5.	Confirmación del Código de Verificación	14
3.1.6.	Reenvío del Código de Verificación	15
3.1.7.	Inicio de Sesión	16
3.2.	Inicio de Sesión (Login)	17
3.2.1.	Pantalla de Inicio de Sesión	17
3.2.2.	Recuperación de Contraseña	17
3.3.	Página Principal (Index)	19
3.3.1.	Header (Encabezado)	19
3.3.2.	Main (Cuerpo Principal)	20
3.3.3.	Footer (Pie de Página)	21
3.3.4.	Monedero (Wallet)	22
3.3.5.	Sobre Nosotros (About Us)	23
3.4.	Favoritos (Favorites)	24
3.4.1.	Interacción con la Estrella de Favoritos	24
3.4.2.	Eliminar de Favoritos	25
3.5.	Pasos para Reservar el Servicio (Steps)	26
3.5.1.	Información del Servicio	26
3.5.2.	Botones de Acción	26
3.6.	Selección de Fecha y Hora (Date)	28
3.6.1.	Título	28
3.6.2.	Selección de Fecha	28
3.6.3.	Selección de Hora	28
3.6.4.	Servicios Seleccionados	28
3.6.5.	Mensaje de Error	29
3.6.6.	Botones de Acción	29
3.7.	Carrito de Compras (Shopping_cart)	31
3.7.1.	Título	31

3.7.2.	Resumen del Pedido	31
3.7.3.	Botones de Acción	31
3.8.	Perfil de Usuario (Profile)	33
3.8.1.	Información del Usuario	33
3.8.2.	Botones de Acción	33
3.9.	Pasarela de Pago (Payment)	36
3.9.1.	Total a Pagar	36
3.9.2.	Selección de Tarjeta de Pago	36
3.9.3.	Botones de Acción	36
4.	Casos de Uso	39
4.1.	CU1: Registrarse	39
4.2.	CU2: Iniciar sesión	40
4.3.	CU3: Recuperar contraseña	41
4.4.	CU4: Visualizar servicios	42
4.5.	CU5: Añadir servicio a favoritos	42
4.6.	CU6: Eliminar servicio de favoritos	43
4.7.	CU7: Consultar pasos del servicio	43
4.8.	CU8: Seleccionar fecha y hora	44
4.9.	CU9: Adjuntar autorización	45
4.10.	CU10: Añadir servicio al carrito	45
4.11.	CU11: Añadir tarjeta	46
4.12.	CU12: Realizar pago	46
4.13.	CU13: Consultar historial de pedidos	47
4.14.	CU14: Modificar datos personales	47
5.	Base de Datos	49
6.	Conclusiones y Trabajo Futuro	53

Introduction	57
6.1. Background	57
6.2. Objectives	59
6.3. Work Plan	60
6.4. Technologies Used	61
6.4.1. HTML (HyperText Markup Language)	61
6.4.2. CSS (Cascading Style Sheets)	61
6.4.3. JavaScript	62
6.4.4. PHP (Hypertext Preprocessor)	63
6.4.5. phpMyAdmin and XAMPP	63
6.5. Conclusion	64
Conclusions and Future Work	65
Bibliografía	69
A. Guía de Uso	71
B. Repositorio código Taskmaker	73

Índice de figuras

1.1. Logo HTML	5
1.2. Logo CSS	6
1.3. Logo JS	6
1.4. Logo PHP	7
1.5. Logo phpMyAdmin	8
1.6. Logo XAMPP	8
3.1. Pantalla Splash Screen + Register	12
3.2. Pantalla Envío código de verificación	13
3.3. Pantalla Verificar cuenta	14
3.4. Pantalla Reenvío código de verificación	15
3.5. Pantalla Slpash Screen + Inicio Sesion	16
3.6. Pantalla Recuperación de Contraseña	18
3.7. Pantalla Página principal + Menu	20
3.8. Pantallas Tarjetas	22
3.9. Pantalla Añadir Tarjetas	23
3.10. Pantallas Sobre Nosotros	23
3.11. Pantallas Favoritos Añadidos	24
3.12. Pantallas Favoritos Eliminados	25
3.13. Pantallas Pasos a seguir	27

3.14. Pantalla Fecha y Hora	29
3.15. Pantallas Calendario + Reloj	30
3.16. Pantalla Carrito Vacío	32
3.17. Pantalla Carrito	32
3.18. Pantallas Perfil de Usuario + Cambiar Contraseña + Editar Datos Personales	35
3.19. Pantalla Pasarela de Pago	37
5.1. Tablas Base de Datos Taskmaker	51
6.1. HTML Logo	61
6.2. CSS Logo	62
6.3. JavaScript Logo	62
6.4. PHP Logo	63
6.5. phpMyAdmin Logo	64
6.6. XAMPP Logo	64

Índice de tablas

2.1. Comparación de servicios entre Glovo, Uber y Taskmaker	9
4.1. Caso de uso: Registrarse	39
4.2. Caso de uso: Iniciar sesión	40
4.3. Caso de uso: Recuperar contraseña	41
4.4. Caso de uso: Visualizar servicios	42
4.5. Caso de uso: Añadir servicio a favoritos	42
4.6. Caso de uso: Eliminar servicio de favoritos	43
4.7. Caso de uso: Consultar pasos del servicio	43
4.8. Caso de uso: Seleccionar fecha y hora	44
4.9. Caso de uso: Adjuntar autorización	45
4.10. Caso de uso: Añadir servicio al carrito	45
4.11. Caso de uso: Añadir tarjeta	46
4.12. Caso de uso: Realizar pago	46
4.13. Caso de uso: Consultar historial de pedidos	47
4.14. Caso de uso: Modificar datos personales	47

Introducción

“El éxito es la suma de pequeños esfuerzos repetidos día tras día”

— Robert Collier

En esta sección se presenta el desarrollo de Taskmaker, una aplicación web pensada para facilitar la gestión de distintos servicios personales de forma sencilla y eficaz. Se explican el contexto y la motivación del proyecto, los objetivos que se persiguen, la planificación llevada a cabo, así como las asignaturas del grado que han resultado especialmente útiles durante su desarrollo.

1.1. Antecedentes

La transformación digital ha dado lugar a un entorno en el que la gestión y contratación de servicios a través de plataformas web se ha convertido en una práctica habitual tanto para empresas como para usuarios particulares. La comodidad de poder acceder a un catálogo de servicios, contratar productos, realizar pagos y gestionar interacciones desde un mismo entorno digital, ha redefinido los modelos de consumo y atención. Este auge ha impulsado el desarrollo de aplicaciones web cada vez más completas, accesibles y centradas en el usuario.

No obstante, en este contexto de crecimiento acelerado, muchas plataformas presentan aún carencias notables: falta de personalización, interfaces poco intuitivas, escasa integración entre módulos o una experiencia de usuario fragmentada. Además, para nuevos emprendedores o pequeñas empresas que deseen digitalizar su catálogo de servicios, las soluciones suelen ser complejas o poco adaptables a sus necesidades específicas.

En este marco nace Taskmaker, un proyecto que responde a la necesidad de contar con una plataforma sencilla, modular y eficiente que permita gestionar servicios digitales de forma personalizada. El desarrollo de este sistema surge como iniciativa

dentro del Trabajo de Fin de Grado en Ingeniería Informática, con el propósito de aplicar de manera práctica los conocimientos adquiridos en diseño web, programación full-stack, bases de datos, seguridad y experiencia de usuario.

Este proyecto busca, no solo construir una solución funcional, sino también afrontar los retos que supone el diseño de un sistema completo en términos de organización, escalabilidad, validación y usabilidad, aspectos que resultan clave para el desarrollo profesional en el ámbito de la ingeniería del software.

1.2. Objetivos

El objetivo general de este Trabajo de Fin de Grado es diseñar y desarrollar una aplicación web funcional denominada Taskmaker, que permita a los usuarios gestionar la contratación de servicios digitales de forma integral, sencilla y eficiente.

Los objetivos específicos definidos para alcanzar este fin son los siguientes:

- Analizar las necesidades funcionales y no funcionales del sistema.
- Diseñar una arquitectura modular basada en el modelo cliente-servidor.
- Desarrollar una interfaz de usuario responsive, clara e intuitiva.
- Implementar las funcionalidades básicas de registro, login, recuperación de cuenta y verificación por correo electrónico.
- Integrar un sistema de carrito de servicios, simulación de pagos y favoritos personalizados.
- Permitir la consulta de un historial de pedidos y la personalización de la experiencia del usuario.
- Diseñar y mantener una base de datos relacional en MySQL para almacenar de forma eficiente la información del sistema.
- Validar la aplicación mediante pruebas funcionales, pruebas de usuario y análisis del comportamiento esperado.

El cumplimiento de estos objetivos permitirá entregar una solución robusta, funcional y con capacidad de evolución, demostrando la madurez técnica alcanzada al finalizar el grado.

1.3. Plan de trabajo

Para garantizar una correcta evolución del proyecto, se ha estructurado el trabajo en distintas fases secuenciales que permiten dividir el proceso en bloques manejables y ordenados:

1. Fase - Análisis de requisitos: Se recopilaron y definieron los requisitos funcionales y no funcionales de la aplicación, estableciendo el alcance del sistema y las funcionalidades prioritarias.
2. Fase - Diseño del sistema: Se elaboró la arquitectura técnica del proyecto, incluyendo la estructura de la base de datos (modelo entidad-relación), los diagramas UML (casos de uso, clases y secuencia) y los esquemas de navegación de la interfaz.
3. Fase - Desarrollo e implementación: Se llevó a cabo la programación de los distintos módulos, comenzando por el backend en PHP y la base de datos en MySQL, e integrando posteriormente el frontend en HTML, CSS y JavaScript. Durante esta fase se incluyeron funcionalidades clave como autenticación, carrito de servicios, favoritos, historial, verificación y recuperación de cuenta por correo electrónico.
4. Fase - Integración y validación: Se realizaron pruebas funcionales para verificar el comportamiento del sistema, así como pruebas de usuario orientadas a evaluar la experiencia y detectar posibles mejoras.
5. Fase - Documentación: Se elaboró la memoria del proyecto incluyendo la descripción del desarrollo técnico, las pruebas realizadas y un análisis crítico de los resultados. También se preparó la documentación técnica necesaria para su posible despliegue o evolución futura.

Este plan ha sido diseñado con el objetivo de reflejar un ciclo de desarrollo realista, que permita aplicar buenas prácticas de ingeniería del software y gestionar de forma ordenada los recursos y tiempos disponibles.

1.4. Tecnologías Empleadas

El desarrollo de la aplicación **Taskmaker** ha requerido el uso de diversas tecnologías para crear una experiencia de usuario interactiva y dinámica, así como una gestión eficaz de los datos. A continuación, se describen las principales tecnologías utilizadas en la creación de la aplicación.

1.4.1. HTML (HyperText Markup Language)

HTML es el lenguaje estándar para la creación de páginas web. Se ha utilizado para estructurar el contenido y la disposición de los elementos en cada una de las páginas de la aplicación. HTML proporciona la base para todos los elementos visibles en la interfaz de usuario, como los formularios de registro, los botones de acción, las tablas y otros componentes esenciales. Además, permite crear enlaces entre las diferentes secciones de la aplicación, facilitando la navegación del usuario. MDN Web Docs (2025b)



Figura 1.1: Logo HTML

1.4.2. CSS (Cascading Style Sheets)

CSS se ha utilizado para dar estilo y diseño a las páginas web, permitiendo personalizar la apariencia de los elementos HTML. Gracias a CSS, se ha logrado una interfaz atractiva, con colores, fuentes, márgenes y espaciados que mejoran la experiencia del usuario. También se ha empleado para hacer la aplicación **responsive**, adaptando el diseño para que sea accesible en dispositivos móviles y pantallas de diferentes tamaños. El uso de media queries y otras propiedades avanzadas de CSS ha permitido ajustar el diseño de forma dinámica. MDN Web Docs (2025a)



Figura 1.2: Logo CSS

1.4.3. JavaScript

JavaScript es un lenguaje de programación utilizado para hacer la aplicación más interactiva y dinámica. En **Taskmaker**, JavaScript se utiliza para mejorar la experiencia de usuario en diversas áreas, como:

- Validación de formularios en tiempo real (por ejemplo, verificación de campos vacíos o correos electrónicos en el formulario de registro).
- Interacción con el usuario mediante eventos, como los clics sobre botones y tarjetas de servicios.
- Manipulación del DOM (Document Object Model) para actualizar el contenido de la página sin necesidad de recargarla (uso de AJAX).
- Implementación de funcionalidades interactivas como el calendario y el reloj desplegable en la página `date.html`.
- Gestión del carrito de compras y los servicios seleccionados, lo que permite una experiencia fluida al añadir o eliminar servicios. MDN Web Docs (2025c)



Figura 1.3: Logo JS

JavaScript también facilita la integración con APIs y servicios externos, mejorando la funcionalidad de la aplicación.

1.4.4. PHP (Hypertext Preprocessor)

PHP es un lenguaje de programación del lado del servidor que se ha utilizado para gestionar la lógica de negocio y la interacción con la base de datos. En **Taskmaker**, PHP se emplea para:

- Procesar los formularios de registro, inicio de sesión y recuperación de contraseñas, verificando las credenciales y enviando los datos a la base de datos.
- Gestionar las operaciones de pago y la creación de pedidos.
- Ejecutar consultas SQL para interactuar con la base de datos y recuperar la información necesaria, como los datos de los usuarios, servicios, pedidos, etc.
- Realizar validaciones del lado del servidor para garantizar la seguridad y la integridad de los datos enviados por el usuario. The PHP Group (2025)



Figura 1.4: Logo PHP

PHP facilita la conexión entre el frontend de la aplicación y la base de datos, permitiendo que la aplicación sea dinámica y funcione de manera eficiente.

1.4.5. phpMyAdmin y XAMPP

phpMyAdmin es una herramienta de administración de bases de datos MySQL basada en la web. En **Taskmaker**, phpMyAdmin se ha utilizado para crear y gestionar la base de datos de la aplicación, que contiene la información del usuario, los servicios, los pedidos, entre otros.

- La base de datos almacena los datos de los usuarios, incluidos los nombres, direcciones, contraseñas (encriptadas), y otros datos relacionados con la cuenta.
- Se gestionan las transacciones de los pedidos, incluyendo la información del servicio solicitado, la fecha, la hora, y el estado del pedido.

- phpMyAdmin facilita la creación de tablas, la ejecución de consultas SQL y la administración de las relaciones entre las diferentes tablas. phpMyAdmin Team (2025)



Figura 1.5: Logo phpMyAdmin

XAMPP es una solución de software libre que proporciona un entorno de servidor local. XAMPP se ha utilizado para configurar y ejecutar el servidor web en el desarrollo de la aplicación **Taskmaker**, permitiendo que el servidor local ejecute PHP y se conecte a la base de datos MySQL a través de phpMyAdmin.

- XAMPP facilita la ejecución de la aplicación de manera local, emulando el entorno de producción sin necesidad de tener un servidor web externo.
- Permite realizar pruebas y depurar la aplicación en un entorno controlado antes de implementarla en un servidor real. Apache Friends (2025)



Figura 1.6: Logo XAMPP

1.5. Conclusión

El uso combinado de HTML, CSS, JavaScript, PHP y herramientas como phpMyAdmin y XAMPP ha permitido desarrollar una aplicación web interactiva y funcional, capaz de gestionar servicios y pagos de forma eficiente, al mismo tiempo que proporciona una experiencia de usuario atractiva y fluida. Estas tecnologías han sido fundamentales para lograr la integridad y seguridad de los datos, así como la optimización de la interfaz de usuario.

Estado de la Cuestión

Taskmaker es una aplicación diseñada para facilitar la gestión de tareas cotidianas que requieren desplazamientos o servicios personalizados. A diferencia de otras plataformas como Glovo y Uber, que se centran principalmente en la entrega de productos o el transporte de personas, Taskmaker ofrece un enfoque más completo y flexible, abarcando una amplia gama de servicios que buscan optimizar el tiempo y mejorar la eficiencia de los usuarios en su vida cotidiana.

En la siguiente tabla, se presenta una comparación entre los servicios que ofrece Taskmaker y aquellos que ofrecen plataformas como Glovo y Uber.

<i>Servicio</i>	Glovo	Uber	Taskmaker
Pasar la ITV al coche	NO	NO	SI
Realizar gestiones bancarias	NO	NO	SI
Pasear a tus mascotas	NO	NO	SI
Recogida y envío de artículos de farmacia	SI	SI	SI
Entrega/Recogida del servicio de tintorería	NO	NO	SI
Entrega de pedidos a domicilio	SI	SI	SI
Realización de colas de espera para eventos	NO	NO	SI
Entrega de compras de Mini Markets	SI	SI	SI
Recogida de los niños del colegio	NO	NO	SI

Tabla 2.1: Comparación de servicios entre Glovo, Uber y Taskmaker

Diferenciación de Taskmaker

La principal distinción de Taskmaker respecto a Glovo y Uber es su enfoque en servicios especializados y personalizados, que van más allá de las entregas de productos o el transporte. Mientras que las plataformas competidoras se centran en la entrega de bienes y la movilidad, Taskmaker aborda necesidades cotidianas que no están relacionadas directamente con estos ámbitos, como gestionar trámites bancarios, pasar la ITV al coche o recoger a los niños del colegio.

- **Servicios personalizados:** a diferencia de Glovo y Uber, Taskmaker permite personalizar las tareas de acuerdo con las necesidades de cada usuario. Esto incluye actividades como realizar colas de espera en eventos. GlovoApp (2025) Uber Technologies Inc. (2025)
- **Variedad de servicios:** Taskmaker se especializa en una gama de servicios más amplia, desde entregas rápidas personalizadas hasta gestiones administrativas, ofreciendo una propuesta integral que optimiza el tiempo y facilita las tareas cotidianas de los usuarios.

Valor añadido de Taskmaker

El valor principal de Taskmaker reside en su capacidad para ofrecer una solución integral a las necesidades diarias de los usuarios. Al combinar servicios de entrega con tareas más especializadas, como el cuidado de mascotas o la gestión de tareas administrativas, Taskmaker permite delegar una amplia variedad de actividades. Esta flexibilidad y personalización son clave para diferenciarse del resto de plataformas, que se limitan a tareas básicas.

Capítulo 3

Descripción del Trabajo

3.1. Registro de Usuario (Register)

El proceso de registro de usuario en **Taskmaker** está diseñado para ser sencillo y seguro, asegurando que solo los usuarios verificados puedan acceder a los servicios de la aplicación. El flujo de trabajo del registro se compone de varias fases:

3.1.1. Pantalla de Bienvenida (Splash Screen)

Al acceder a la aplicación, el usuario se encuentra con una **pantalla de bienvenida (splash screen)** que se muestra durante unos segundos. Esta pantalla sirve como introducción a la aplicación y desaparece automáticamente, redirigiendo al usuario a la pantalla de **inicio de sesión (login)**.

3.1.2. Pantalla de Inicio de Sesión (Login)

Desde la pantalla de inicio de sesión, el usuario tiene la opción de acceder a la página de **registro** si no tiene una cuenta. Para ello, debe hacer clic en el enlace “*¿No tienes cuenta? Regístrate*”, lo que lo llevará automáticamente al formulario de registro.

3.1.3. Formulario de Registro

El formulario de registro solicita la siguiente información, que es **obligatoria** para completar el proceso:

- **Nombre:** Para personalizar la experiencia dentro de la aplicación.
- **Apellidos:** Para completar los datos del usuario.
- **Correo electrónico:** Esencial para la creación de la cuenta y la verificación. Este campo debe ser único. Además, se valida que el correo tenga un formato correcto. Si el correo tiene un formato incorrecto, se mostrará el mensaje de error *“Introduce un correo electrónico válido”*.
- **DNI:** Para identificación del usuario, asegurando que los datos sean únicos.
- **Dirección:** Para completar la información de contacto y servicios, necesario.
- **Nombre de usuario:** Para la identificación dentro de la aplicación. Este será único.
- **Contraseña:** El usuario debe crear una contraseña segura que cumpla con los siguientes requisitos:
 - Al menos 6 caracteres.
 - Una letra mayúscula.
 - Un número.

Además, junto al campo de la contraseña aparece un icono de un ojo, que permite al usuario ver o ocultar la contraseña que está introduciendo. Al hacer clic en el icono, la contraseña se muestra en texto plano y, al hacer clic nuevamente, se oculta.

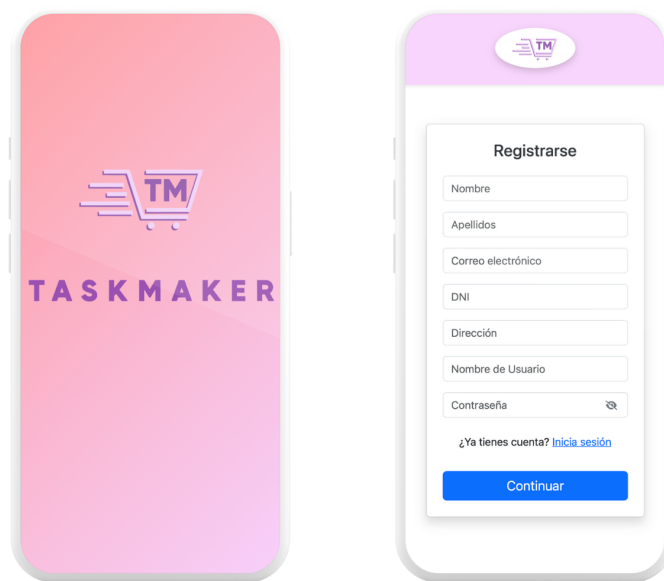


Figura 3.1: Pantalla Splash Screen + Register

Una vez que el usuario ha completado el formulario, debe hacer clic en “*Continuar*” para enviar los datos al backend para su validación y almacenamiento. Si algún campo no ha sido completado correctamente, se mostrará el mensaje de error “*Todos los campos son obligatorios*”. En caso de que todo sea correcto, el sistema mostrará el mensaje “*Registro exitoso. Revisa tu correo*” indicando que el registro se ha completado con éxito.

3.1.4. Verificación por Correo Electrónico

Tras completar el registro, **Taskmaker** envía un **código de verificación de 6 dígitos** al correo electrónico proporcionado por el usuario. Este paso es esencial para confirmar que la cuenta es legítima y que el correo electrónico introducido pertenece al usuario.

- **Código de verificación:** El código enviado es único, generado al azar y permite verificar la cuenta. El correo electrónico contiene el código de verificación, pero **sin un enlace directo** a la aplicación. El usuario deberá copiar el código o memorizarlo y luego ir manualmente a la aplicación para ingresarlo.



Figura 3.2: Pantalla Envío código de verificación

3.1.5. Confirmación del Código de Verificación

El usuario debe ingresar el **código de verificación de 6 dígitos** en el formulario correspondiente dentro de la aplicación. Si el código es correcto, la cuenta se activa y se muestra el mensaje *“Cuenta verificada correctamente”*. Si el código es incorrecto o ya ha sido utilizado, se mostrará el mensaje de error *“Código incorrecto o ya usado”*.

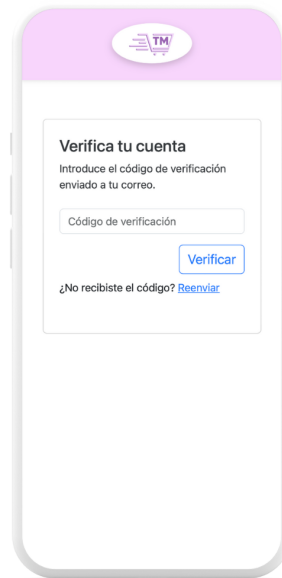


Figura 3.3: Pantalla Verificar cuenta

3.1.6. Reenvío del Código de Verificación

Si el usuario no recibe el correo con el código de verificación, tiene la opción de hacer clic en “*¿No recibiste el código? Reenviar*” para que el sistema envíe nuevamente el código de activación al correo electrónico proporcionado.

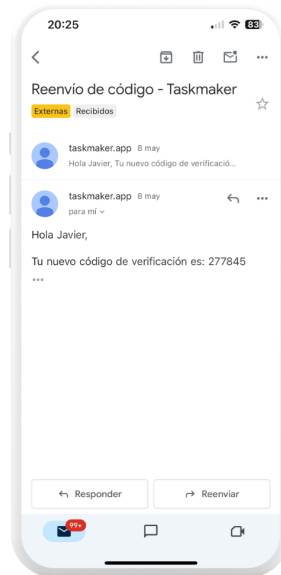


Figura 3.4: Pantalla Reenvío código de verificación

3.1.7. Inicio de Sesión

Una vez que el código de verificación ha sido introducido correctamente y la cuenta ha sido activada, el usuario es redirigido a la pantalla de **inicio de sesión** para ingresar su **Nombre de Usuario** y **Contraseña** recién creados. Esto asegura que el usuario pueda acceder de inmediato a su cuenta y empezar a utilizar los servicios de **Taskmaker**.

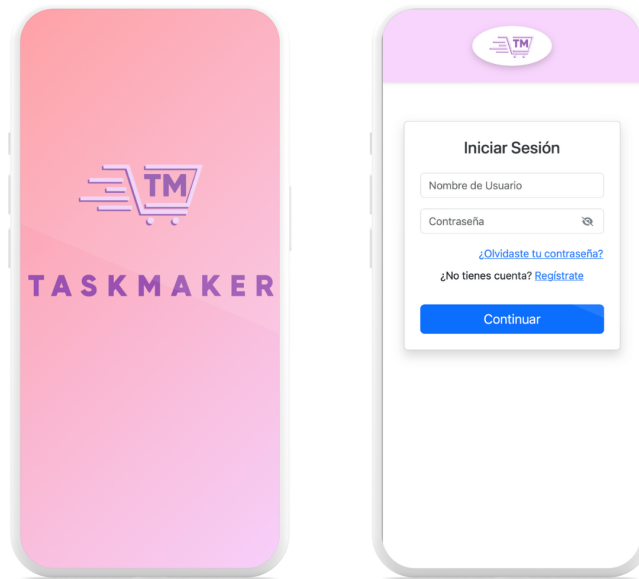


Figura 3.5: Pantalla Slpash Screen + Inicio Sesion

3.2. Inicio de Sesión (Login)

Una vez que el usuario tiene su cuenta verificada, puede acceder a la aplicación a través del **inicio de sesión (login)**. El flujo de trabajo de inicio de sesión se compone de varias fases:

3.2.1. Pantalla de Inicio de Sesión

En la pantalla de **inicio de sesión**, se le solicita al usuario que ingrese su **nombre de usuario** y **contraseña**. Después de ingresar las credenciales, el usuario debe hacer clic en “*Continuar*” para proceder con la autenticación.

- Si las credenciales son correctas, el sistema valida la autenticación y el usuario es redirigido a la **página principal** de la aplicación, donde podrá acceder a todos los servicios disponibles. Además, se muestra el mensaje “*Inicio de sesión exitoso*”.
- Si las credenciales son incorrectas, el sistema mostrará un mensaje de error indicando que las credenciales no son válidas.

Además, en la pantalla de inicio de sesión, se encuentra la opción para *registrarse* en caso de que el usuario no tenga una cuenta. Esto se logra haciendo clic en el enlace “*¿No tienes cuenta? Regístrate*”, que redirige al usuario al formulario de registro.

3.2.2. Recuperación de Contraseña

Si el usuario ha olvidado su contraseña, existe la opción de recuperarla. Para ello, debe hacer clic en “*¿Olvidaste tu contraseña?*”, lo que abrirá un pequeño formulario titulado “*Recuperar Contraseña*”.

- **Formulario de Recuperación de Contraseña:** el formulario solicita al usuario que introduzca el **DNI** y el **correo electrónico** con el que se registró en la aplicación. Luego, debe ingresar una nueva **contraseña** y confirmarla en el campo correspondiente.
- **Contraseña nueva:** la contraseña debe cumplir con los mismos requisitos que al momento del registro:
 - Al menos 6 caracteres.
 - Una letra mayúscula.
 - Un número.

Además, junto al campo de la nueva contraseña aparece un icono de un ojo, que permite al usuario ver o ocultar la contraseña que está introduciendo. Al hacer clic en el icono, la contraseña se muestra en texto plano y, al hacer clic nuevamente, se oculta.

- El usuario debe hacer clic en *“Guardar”* para guardar la nueva contraseña. Si en algún momento desea cancelar la acción, puede hacer clic en el botón *“Cancelar”*.
- Si la nueva contraseña se guarda correctamente, se muestra el mensaje *“Contraseña actualizada con éxito”*.

Una vez que el usuario haya completado el formulario de recuperación y guardado la nueva contraseña, podrá usarla para iniciar sesión en la aplicación.



Figura 3.6: Pantalla Recuperación de Contraseña

3.3. Página Principal (Index)

La página principal (`index.html`) es la página de inicio de la aplicación **Taskmaker**. En ella, el usuario podrá navegar por los servicios disponibles, acceder a información sobre cada servicio y gestionar sus servicios favoritos. A continuación se detalla su estructura:

3.3.1. Header (Encabezado)

1. Menú (Hamburguesa): En la parte superior izquierda se encuentra el menú de navegación, representado por un ícono de hamburguesa. Al hacer clic sobre él, el menú lateral se desplegará, permitiendo al usuario acceder a diferentes secciones de la aplicación. En este menú, se encuentran las siguientes opciones:

- **Inicio (index):** Redirige al usuario a la página principal (`index.html`).
- **Sobre Nosotros (about us):** Proporciona información sobre la empresa o los servicios que ofrece **Taskmaker**.
- **Monedero (wallet):** Muestra las **tarjetas** añadidas por el usuario, con la opción de **agregar más tarjetas** para facilitar pagos o transacciones dentro de la app.
- **Perfil (profile):** Dirige a la página del perfil del usuario, donde puede ver y editar su información personal y preferencias.
- **Cerrar sesión (log out):** Permite al usuario cerrar sesión y salir de la aplicación. Esto lo redirige a la página de inicio de sesión.

2. Logo de Taskmaker: En el centro del encabezado está el logo de **Taskmaker**, que actúa como un enlace hacia la página principal. Esto permite a los usuarios regresar rápidamente a la pantalla de inicio desde cualquier otra página dentro de la app.

3. Buscador y Carrito de Compras: A la derecha del encabezado, se encuentra el **buscador**, que permite a los usuarios buscar servicios específicos. Justo al lado del buscador se encuentra el **carrito de compras** (ícono del carrito), denominado *shopping_cart*, que muestra los servicios que el usuario ha añadido a su carrito. Al hacer clic en este ícono, el usuario puede ver y gestionar los artículos en su carrito.

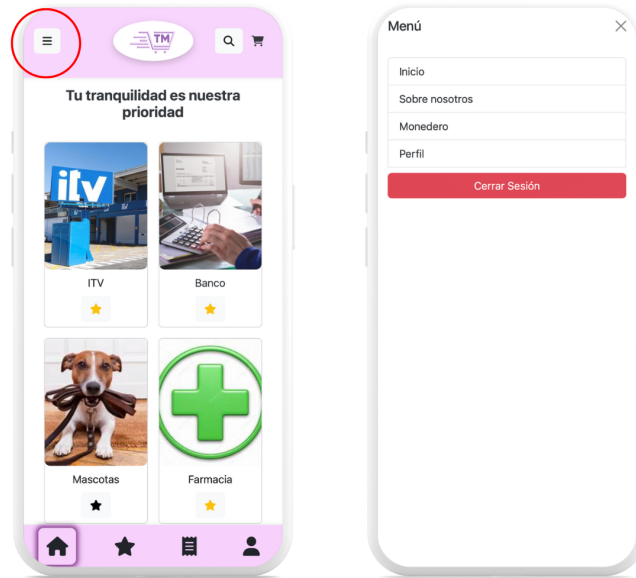


Figura 3.7: Pantalla Página principal + Menu

3.3.2. Main (Cuerpo Principal)

1. **Eslogan:** Justo debajo del encabezado, aparece el eslogan de la aplicación: “Tu tranquilidad es nuestra prioridad”, para reforzar el mensaje de confianza y seguridad que la app busca transmitir a sus usuarios.

2. **Servicios Disponibles:** El contenido principal está compuesto por una serie de **tarjetas de servicios** que se muestran en dos columnas. Cada tarjeta representa un servicio disponible. Los servicios disponibles en **Taskmaker** incluyen:

- Pasar la ITV al coche
- Realizar gestiones con el banco
- Pasear a tus mascotas
- Recogida y envío de artículos de farmacia a domicilio
- Entrega/recogida del servicio de tintorería
- Entrega de pedidos a domicilio
- Realizar colas de espera para eventos
- Entregar la compra realizada en un Mini Market o supermercado que no tiene servicio a casa
- Entrega/recogida del servicio de zapatería

- **Recogida de los niños del colegio**

Los servicios se presentan en una cuadrícula con desplazamiento hacia abajo, permitiendo que el usuario vea más servicios a medida que hace scroll.

3. **Acceso a los Pasos del Servicio:** Si el usuario hace clic sobre una de las tarjetas de servicio, será redirigido a la página **steps.html**, donde se le explicarán los pasos a seguir para realizar dicho servicio.

4. **Favoritos:** Cada tarjeta de servicio incluye una **estrella** de favoritos. Si el usuario pulsa sobre la estrella, el servicio se añade a sus **favoritos**. Si la estrella está activada (amarilla) y se vuelve a pulsar, el servicio se elimina de la lista de favoritos.

3.3.3. Footer (Pie de Página)

En la parte inferior de la pantalla se encuentra el **footer**, que tiene los siguientes íconos para navegar a diferentes secciones de la app. Además, el usuario siempre sabe en qué parte de la app se encuentra, ya que el botón de la sección activa estará resaltado mediante un recuadro.

- **Casita (Home):** Lleva al usuario de vuelta a la página principal (index.html).
- **Estrella (Favoritos):** Esta opción lleva al usuario a una nueva página donde puede ver todos los servicios que ha marcado como favoritos. Aquí podrá gestionar y acceder rápidamente a los servicios que más le interesen. Si el usuario se encuentra en la página de favoritos, el botón de la estrella estará destacado con un recuadro.
- **Pedidos (Orders):** En esta sección, el usuario puede ver los **pedidos en curso** (servicios que ha solicitado y están siendo gestionados) y los **pedidos pasados** (servicios que ya ha completado). Esto permite al usuario llevar un control de sus solicitudes. Si el usuario está en la sección de pedidos, el botón de pedidos tendrá un recuadro.
- **Perfil:** Dirige a la página de perfil del usuario, donde puede gestionar sus datos personales y configuraciones de cuenta. Si el usuario está en la página de perfil, el botón de perfil será el que esté rodeado por el recuadro.

3.3.4. Monedero (Wallet)

Dentro del menú de navegación, al hacer clic en la opción **Monedero (wallet)**, el usuario puede ver las **tarjetas de pago** que ha añadido para gestionar sus transacciones dentro de la aplicación. Esta sección incluye:

- **Tarjetas Guardadas:** El usuario podrá ver una lista de las tarjetas que ha agregado previamente. Los primeros dígitos de la tarjeta estarán ocultos con asteriscos (*), y solo los **últimos 4 dígitos** serán visibles. Además, se muestra la **fecha de vencimiento** de cada tarjeta.
- **Eliminar:** Junto a cada tarjeta guardada, hay un botón de **Eliminar** que permite al usuario eliminar la tarjeta de su monedero si lo desea.
- **Agregar Tarjetas:** En la parte inferior de la página, el usuario tiene la opción de **Agregar Tarjetas**, lo que abrirá un formulario donde podrá ingresar los detalles de una nueva tarjeta de pago. Al hacer clic en el botón “*Agregar Tarjeta*”, la nueva tarjeta se añadirá a su monedero.

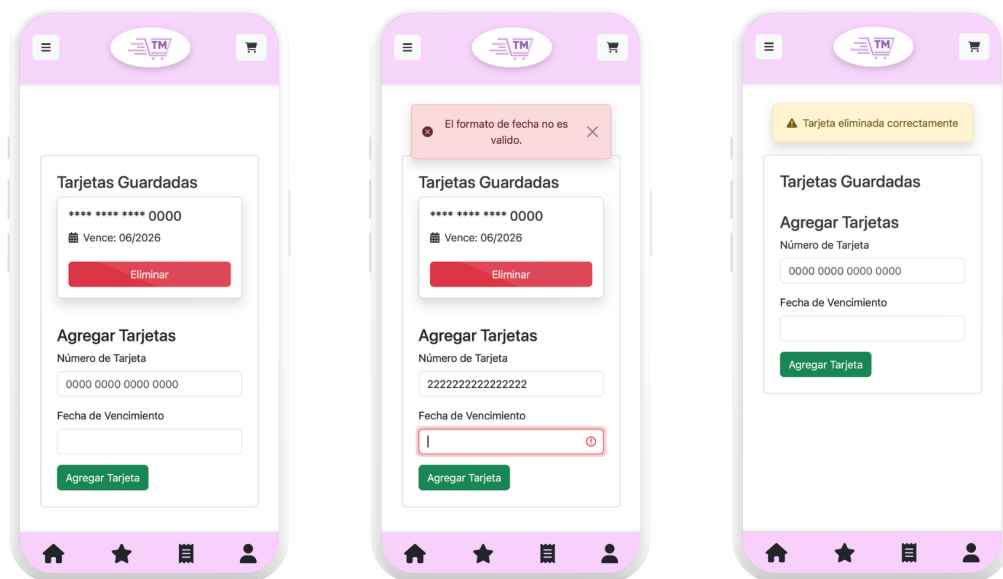


Figura 3.8: Pantallas Tarjetas

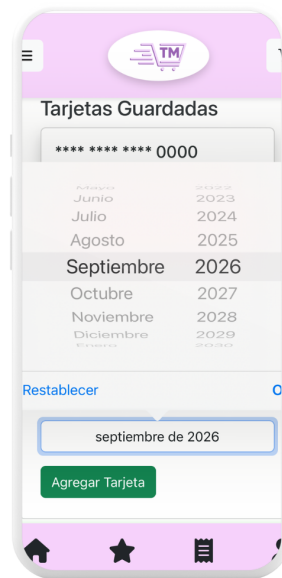


Figura 3.9: Pantalla Añadir Tarjetas

3.3.5. Sobre Nosotros (About Us)

En la sección de **Sobre Nosotros (about us)**, el usuario podrá obtener información acerca de la aplicación, sus objetivos y los servicios que ofrece. Al final de esta página, se encuentra un botón titulado “Volver” que redirige al usuario a la **página principal (index.html)**.

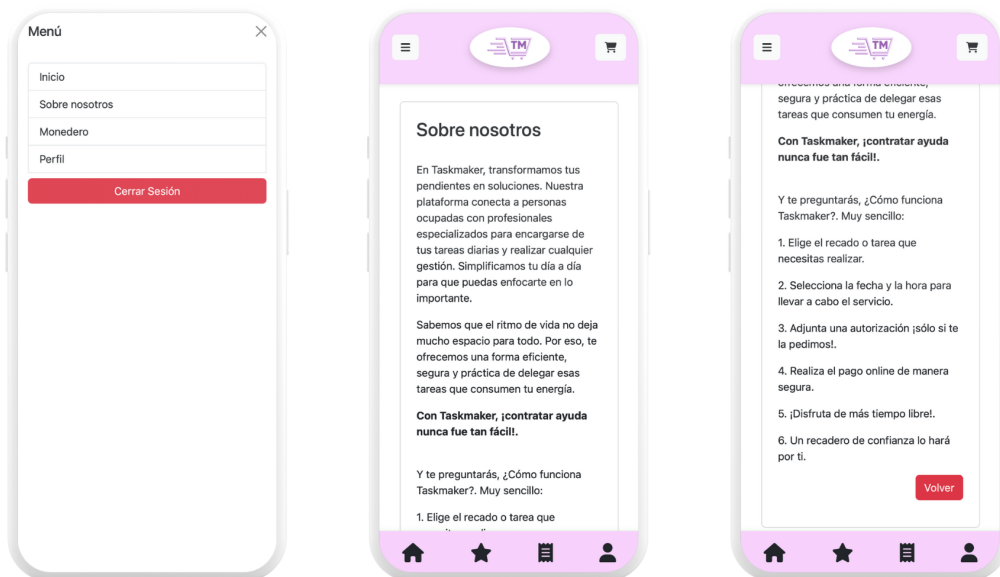


Figura 3.10: Pantallas Sobre Nosotros

3.4. Favoritos (Favorites)

La página de **Favoritos (favorites.html)** permite al usuario ver todos los servicios que ha añadido a su lista de favoritos. Esta sección es accesible a través del ícono de estrella en el **footer**, que redirige al usuario a una página donde podrá gestionar y acceder rápidamente a sus servicios preferidos. A continuación se detalla su funcionamiento:

3.4.1. Interacción con la Estrella de Favoritos

1. Añadir a Favoritos: Cuando el usuario se encuentra en la página principal (*index.html*) y pulsa sobre la estrella de un servicio, la estrella cambia a color amarillo, indicando que el servicio ha sido añadido a los favoritos del usuario. Al mismo tiempo, se muestra un mensaje que dice *"Servicio añadido a favoritos"*.

2. Ver Favoritos: Si el usuario hace clic en el botón de estrella en el **footer**, se le redirige a la página **Favoritos**, donde podrá ver todos los servicios que ha añadido a favoritos. Si no ha añadido ningún servicio a favoritos, se mostrará el mensaje *"No tienes servicios añadidos a favoritos"*.

3. Número de Servicios en Favoritos: El usuario puede tener de 1 a todos los servicios destacados en esta página. Los servicios se muestran de forma accesible, permitiendo al usuario gestionarlos fácilmente.

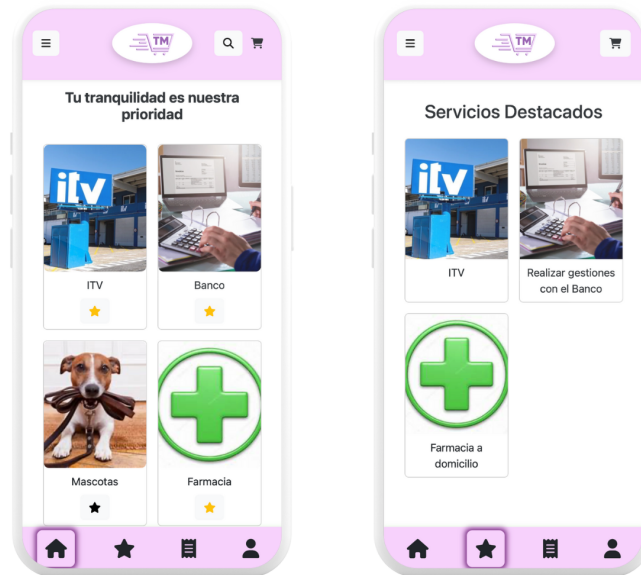


Figura 3.11: Pantallas Favoritos Añadidos

3.4.2. Eliminar de Favoritos

1. Eliminar un Servicio de Favoritos: Desde la página principal (*index.html*), el usuario puede eliminar un servicio de sus favoritos al pulsar nuevamente sobre la estrella amarilla. En ese caso, la estrella volverá a su color por defecto y se mostrará un mensaje de “*Servicio eliminado de favorito*”.

2. Reflejo en la Página de Favoritos: Si el usuario elimina un servicio de favoritos, al volver a la página **Favoritos**, comprobará que el servicio ya no aparece en su lista de favoritos.

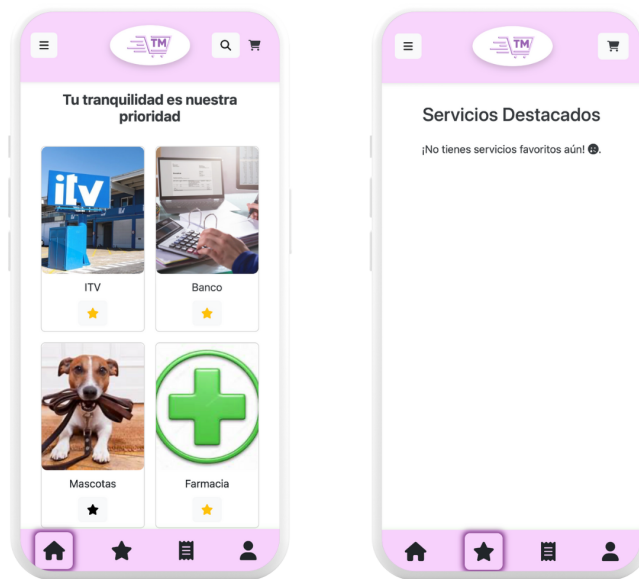


Figura 3.12: Pantallas Favoritos Eliminados

3.5. Pasos para Reservar el Servicio (Steps)

Cuando el usuario hace clic sobre cualquier servicio en la página **index.html**, es redirigido a la página **steps.html**, donde se le muestran los pasos que debe seguir para reservar dicho servicio, así como el **precio** correspondiente al servicio seleccionado. Esta página tiene la siguiente estructura:

3.5.1. Información del Servicio

Al cargar la página, se muestra la siguiente información relacionada con el servicio seleccionado:

- **Pasos a seguir:** Se detallan los pasos específicos que el usuario debe seguir para poder realizar la reserva de este servicio. Estos pasos guían al usuario a lo largo del proceso de contratación del servicio.
- **Precio del servicio:** Se muestra el precio total que el usuario deberá pagar por el servicio. Esta información es clara y transparente para el usuario.

3.5.2. Botones de Acción

En la parte inferior de la página, se encuentran dos botones:

- **Cancelar:** Al hacer clic en el botón “*Cancelar*”, el usuario será redirigido a la página principal **index.html**.
- **Continuar:** Si el usuario decide continuar con el proceso, debe hacer clic en el botón “*Continuar*”, que lo redirigirá a la siguiente página del proceso de reserva, es decir, **date.html**.

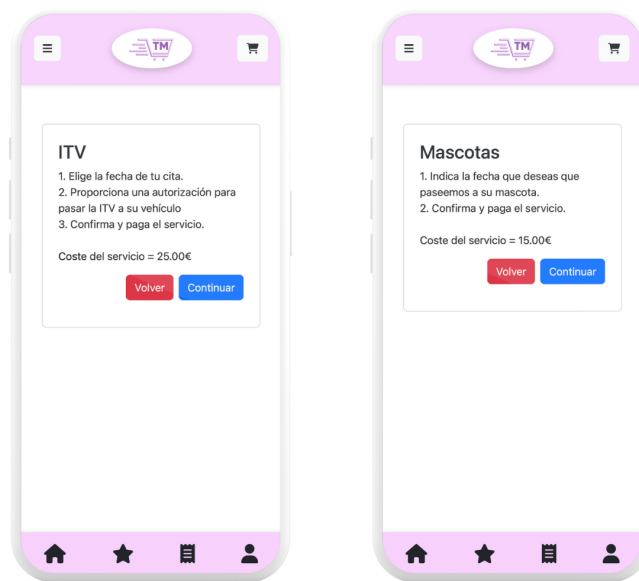


Figura 3.13: Pantallas Pasos a seguir

3.6. Selección de Fecha y Hora (Date)

La página **date.html** permite al usuario seleccionar la fecha y la hora en la que desea que se realice el servicio. Esta página tiene la siguiente estructura:

3.6.1. Título

En la parte superior de la página, se muestra el título: “*Selecciona fecha y hora*”, que indica al usuario lo que debe hacer en esta sección.

3.6.2. Selección de Fecha

Se despliega un **calendario interactivo** en el que el usuario puede elegir la fecha en la que desea que se realice el servicio. El calendario permite seleccionar cualquier día disponible, facilitando la planificación del servicio.

3.6.3. Selección de Hora

Para seleccionar la hora, el usuario tiene dos opciones:

- El usuario puede escribir la hora directamente en un campo de texto.
- También se muestra un **reloj desplegable** desde el cual el usuario puede elegir la hora en un formato de 24 horas. Al hacer clic en el reloj, se abre una interfaz donde se puede seleccionar fácilmente la hora y los minutos.

3.6.4. Servicios Seleccionados

Debajo de la selección de hora, aparece un recuadro titulado **Servicios seleccionados**. En esta sección, el usuario puede ver los servicios que ha seleccionado y proceder a añadir una **autorización** para que el servicio se pueda realizar en su nombre.

- **Adjuntar Autorización:** En este recuadro, el usuario tiene la opción de adjuntar un archivo en formato **PDF** o **imagen**. Esta autorización es necesaria para servicios como:
 - Pasar la ITV al coche.
 - Realizar gestiones administrativas.

- Recoger a los niños del colegio.
- Realizar colas de espera para eventos.

3.6.5. Mensaje de Error

Si el usuario no selecciona ni la fecha ni la hora, se muestra un mensaje de error en pantalla: *“Por favor, selecciona una fecha y una hora antes de continuar.”* Este mensaje asegura que el usuario no pueda continuar con la reserva sin haber completado ambos campos.

3.6.6. Botones de Acción

En la parte inferior de la página, el usuario encuentra dos botones:

- **Cancelar:** Si el usuario hace clic en el botón *“Cancelar”*, será redirigido a la **página principal (index.html)**.
- **Continuar:** Si el usuario decide proceder con la reserva, debe hacer clic en el botón *“Continuar”*, que lo redirigirá al **shopping_cart** para revisar el servicio y confirmar su compra.

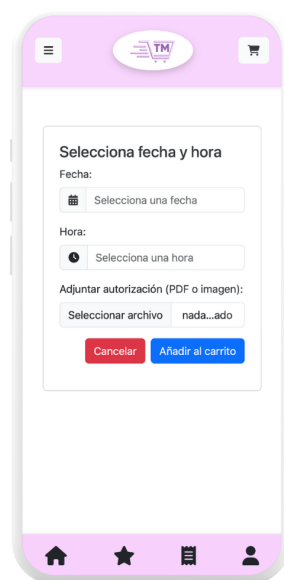


Figura 3.14: Pantalla Fecha y Hora

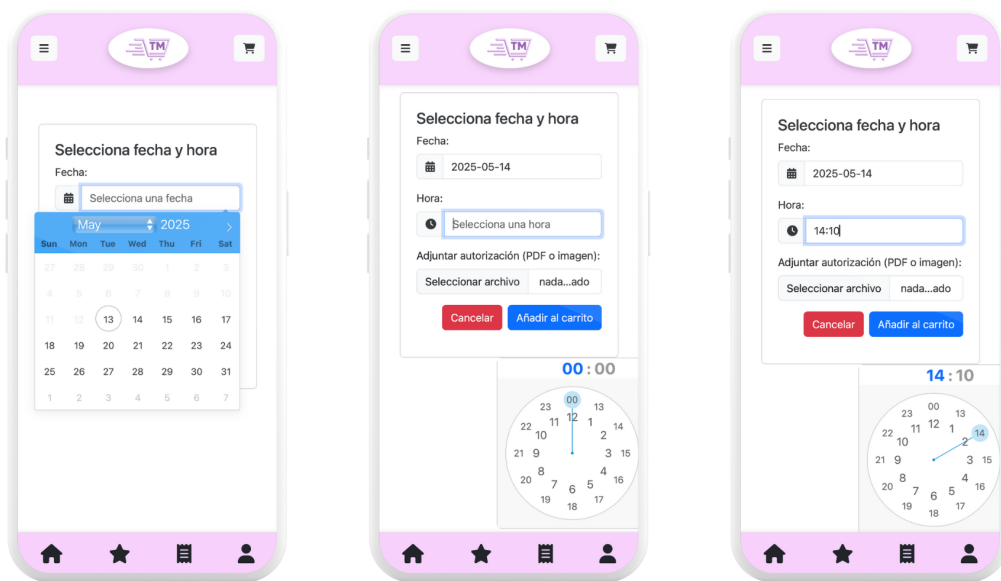


Figura 3.15: Pantallas Calendario + Reloj

3.7. Carrito de Compras (Shopping_cart)

La página `shopping_cart.html` permite al usuario ver un resumen de los servicios que ha añadido a su carrito de compras. En esta sección, el usuario podrá revisar los servicios seleccionados antes de proceder al pago. La estructura de esta página es la siguiente:

3.7.1. Título

En la parte superior de la página, se muestra el título: *“Resumen del pedido”*, que indica al usuario que está viendo un resumen de los servicios añadidos al carrito.

3.7.2. Resumen del Pedido

A continuación, se muestran todos los servicios que el usuario ha añadido al carrito y que aún no han sido pagados. Para cada servicio en el carrito, se presenta la siguiente información:

- **Imagen del servicio:** Se muestra una imagen representativa del servicio seleccionado.
- **Nombre del servicio:** El nombre del servicio añadido al carrito.
- **Fecha y hora:** La fecha y hora seleccionada para la realización del servicio.

Si el usuario no ha añadido ningún servicio al carrito, se mostrará el mensaje: *“¡No tienes servicios añadidos al carrito!”*.

3.7.3. Botones de Acción

En la parte inferior de la página, se encuentran tres botones que permiten al usuario gestionar su pedido:

- **Volver:** Si el usuario hace clic en el botón *“Volver”*, será redirigido a la **página principal (index.html)**.
- **Tramitar pedido:** Si el usuario desea continuar con la compra, debe hacer clic en el botón *“Tramitar pedido”*, que lo llevará a la **Pasarela de pago** para completar la transacción.

- **Añadir otro servicio:** Si el usuario desea añadir más servicios al carrito antes de proceder al pago, puede hacer clic en el botón “*Añadir otro servicio*”, que lo redirigirá a la **página principal (index.html)** para seleccionar nuevos servicios.

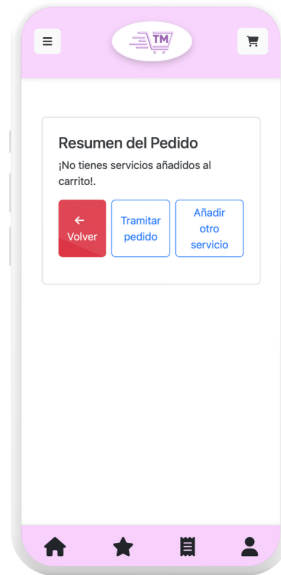


Figura 3.16: Pantalla Carrito Vacío



Figura 3.17: Pantalla Carrito

3.8. Perfil de Usuario (Profile)

La página `profile.html` permite al usuario ver y editar su información personal. En esta página, el usuario puede modificar su contraseña, editar sus datos personales y cerrar sesión. La estructura de esta página es la siguiente:

3.8.1. Información del Usuario

En la parte superior de la página, se muestra el título “*Perfil de Usuario*”. A continuación, se presenta toda la información del usuario, que incluye:

- **Nombre:** El nombre completo del usuario.
- **Email:** La dirección de correo electrónico asociada a la cuenta.
- **DNI:** El número de documento de identidad del usuario.
- **Dirección:** La dirección de residencia del usuario.
- **Nombre de usuario:** El nombre de usuario único dentro de la aplicación. Este campo no es editable y aparece en un recuadro gris para indicar que no se puede modificar.

3.8.2. Botones de Acción

En la parte inferior de la página, el usuario encuentra tres botones que le permiten realizar diversas acciones:

- **Modificar contraseña:** Al hacer clic en el botón “*Modificar contraseña*”, se abre un cuadro con el título “*Cambiar Contraseña*”. Este formulario solicita al usuario que ingrese:
 - **Contraseña actual:** El usuario debe introducir su contraseña actual.
 - **Nueva contraseña:** El usuario debe ingresar su nueva contraseña, que debe cumplir con los requisitos de seguridad establecidos.
 - **Confirmar nueva contraseña:** El usuario debe confirmar la nueva contraseña ingresada.

También se incluye un **icono de ojo** junto a los campos de contraseña, que permite al usuario mostrar u ocultar la contraseña mientras la escribe. Los siguientes mensajes de error o éxito pueden aparecer según el caso:

- Si la **contraseña actual** es incorrecta, se muestra el mensaje “*La contraseña actual no es correcta*”.

- Si la **contraseña actual** es correcta y las **nuevas contraseñas** coinciden, se muestra el mensaje “*¡Contraseña actualizada con éxito!*”.
- Si la **nueva contraseña** y la **confirmación de la nueva contraseña** no coinciden, se muestra el mensaje “*Las contraseñas nuevas no coinciden*”.

Después de completar el formulario, el usuario puede hacer clic en:

- **Cancelar:** Redirige al usuario al perfil sin realizar ningún cambio en la contraseña.
 - **Guardar:** Actualiza la contraseña y redirige al usuario al perfil con la nueva contraseña registrada.
- **Editar datos personales:** Al hacer clic en el botón “*Editar datos personales*”, se abre un cuadro con el título “*Editar Datos Personales*”. En este formulario, el usuario puede editar la siguiente información:
- **Nombre:** El usuario puede modificar su nombre.
 - **Apellidos:** El usuario puede modificar sus apellidos.
 - **Correo electrónico:** El usuario puede modificar su dirección de correo electrónico.
 - **Dirección:** El usuario puede modificar su dirección de residencia.

El campo **Nombre de usuario** aparece en un recuadro gris, indicando que este campo no se puede modificar. En la parte inferior del formulario de edición, el usuario encontrará dos botones:

- **Cancelar:** Redirige al usuario al perfil sin realizar ningún cambio en los datos personales.
 - **Guardar cambios:** Guarda los cambios realizados en los datos personales y redirige al usuario al perfil, donde podrá ver los datos actualizados. Si los datos se guardan correctamente, se muestra el mensaje “*Datos actualizados correctamente*”.
- **Cerrar sesión:** Al hacer clic en el botón “*Cerrar sesión*”, el usuario cierra su sesión en la aplicación. Esto lo redirige a la página de **login**, y se muestra primero la **splash screen**.

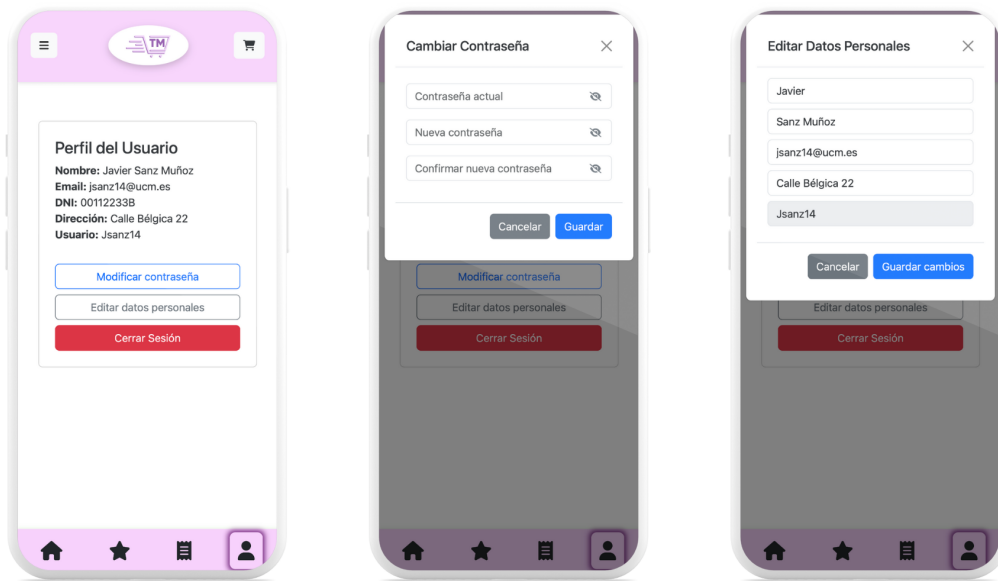


Figura 3.18: Pantallas Perfil de Usuario + Cambiar Contraseña + Editar Datos Personales

3.9. Pasarela de Pago (Payment)

La página **payment.html** permite al usuario completar el proceso de pago de los servicios seleccionados. En esta sección, el usuario podrá ver el total a pagar y seleccionar la tarjeta con la que desea realizar el pago. La estructura de esta página es la siguiente:

3.9.1. Total a Pagar

En la parte superior de la página, se muestra el total a pagar por los servicios seleccionados en el **carrito de compras**. Esta cantidad incluye todos los servicios que el usuario ha agregado y la fecha/hora seleccionada para su realización.

3.9.2. Selección de Tarjeta de Pago

El usuario debe seleccionar una tarjeta de las que ha añadido previamente en su **monedero**. Si el usuario tiene más de una tarjeta, podrá elegir la que desea usar para completar el pago. Si solo tiene una tarjeta añadida, la selección será obligatoria, aunque la tarjeta seleccionada sea la única disponible.

- Si el usuario no ha añadido ninguna tarjeta, se muestra un enlace al **Monedero** con el mensaje: *“No tienes tarjetas añadidas. Añade una tarjeta para realizar el pago.”*, que redirige al usuario a la sección de **Monedero** donde podrá agregar una tarjeta.

3.9.3. Botones de Acción

En la parte inferior de la página, se encuentran dos botones de acción:

- **Volver al carrito:** Si el usuario hace clic en el botón *“Volver al carrito”*, será redirigido al **carrito de compras** (*shopping_cart.html*), donde podrá revisar y modificar los servicios que ha agregado al carrito.
- **Pagar:** Si el usuario hace clic en el botón *“Pagar”*, se completa el proceso de pago y el usuario es redirigido a la página **orders.html**, donde podrá ver los detalles de su pedido en curso.

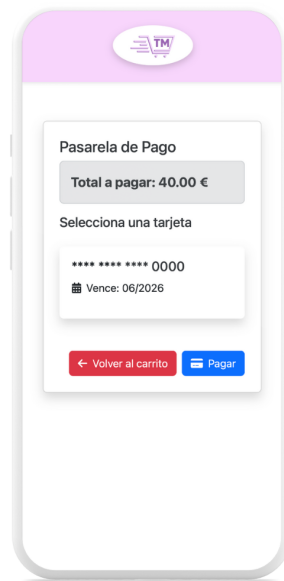


Figura 3.19: Pantalla Pasarela de Pago

Capítulo 4

Casos de Uso

4.1. CU1: Registrarse

Actor principal:	Usuario
Descripción:	Permite a un nuevo usuario registrarse en la plataforma introduciendo sus datos personales.
Flujo principal:	<ol style="list-style-type: none">1. El usuario accede al formulario de registro.2. Introduce su DNI, nombre, apellidos, email, dirección, nombre de usuario y contraseña.3. El sistema valida los datos y guarda la información.4. Se genera un código de verificación que se envía por correo electrónico.5. El usuario introduce el código para verificar su cuenta.
Precondición:	El usuario no debe estar registrado previamente.
Postcondición:	El usuario queda registrado y verificado en la plataforma.

Tabla 4.1: Caso de uso: Registrarse

4.2. CU2: Iniciar sesión

Actor principal:	Usuario
-------------------------	---------

Descripción:	El usuario accede a la aplicación introduciendo sus credenciales.
---------------------	---

Flujo principal:	<ol style="list-style-type: none">1. El usuario accede al formulario de inicio de sesión.2. Introduce nombre de usuario y contraseña.3. El sistema verifica las credenciales.4. Si son válidas, el usuario accede a la plataforma.
-------------------------	---

Precondición:	El usuario debe estar registrado y haber verificado su cuenta.
----------------------	--

Postcondición:	El usuario accede a su cuenta.
-----------------------	--------------------------------

Tabla 4.2: Caso de uso: Iniciar sesión

4.3. CU3: Recuperar contraseña

Actor principal: Usuario

Descripción: Permite al usuario recuperar el acceso a su cuenta mediante el restablecimiento de su contraseña.

Flujo principal:

1. El usuario accede al formulario de recuperación de contraseña.
2. Introduce su DNI y el correo electrónico asociado.
3. El sistema verifica que los datos coinciden con un usuario registrado.
4. Si los datos son correctos, se habilita el formulario de cambio de contraseña.
5. El usuario introduce una nueva contraseña y la confirma repitiéndola.
6. El sistema valida que ambas contraseñas coinciden.
7. La nueva contraseña se guarda en la base de datos y se informa al usuario del cambio exitoso.

Precondición: El usuario debe estar registrado en el sistema.

Postcondición: La contraseña se actualiza y el usuario puede iniciar sesión con la nueva.

Tabla 4.3: Caso de uso: Recuperar contraseña

4.4. CU4: Visualizar servicios

Actor principal:	Usuario
Descripción:	El usuario consulta el catálogo de servicios disponibles en la página principal. Se muestra una vista general de cada servicio y, si desea más información, puede acceder al detalle.
Flujo principal:	<ol style="list-style-type: none"> 1. El usuario accede a la página principal tras iniciar sesión. 2. El sistema muestra todos los servicios disponibles con su nombre e imagen representativa. 3. Si el usuario desea consultar más información (como la descripción o el precio), hace clic sobre el servicio. 4. El sistema redirige al usuario a la sección steps, donde se muestran los pasos a seguir para contratar el servicio, junto con la descripción detallada y el precio.
Precondición:	El usuario debe haber iniciado sesión correctamente.
Postcondición:	El usuario visualiza la información general o detallada de los servicios según su interacción.

Tabla 4.4: Caso de uso: Visualizar servicios

4.5. CU5: Añadir servicio a favoritos

Actor principal:	Usuario
Descripción:	Permite al usuario marcar servicios como favoritos.
Flujo principal:	<ol style="list-style-type: none"> 1. El usuario pulsa el icono de favorito junto a un servicio. 2. El sistema guarda la relación en la base de datos. 3. El servicio se muestra como favorito.
Precondición:	Usuario autenticado.
Postcondición:	El servicio queda almacenado como favorito.

Tabla 4.5: Caso de uso: Añadir servicio a favoritos

4.6. CU6: Eliminar servicio de favoritos

Actor principal:	Usuario
Descripción:	Permite al usuario quitar un servicio previamente marcado como favorito directamente desde la página principal.
Flujo principal:	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de servicios. 2. Identifica el servicio marcado como favorito (estrella amarilla). 3. Hace clic sobre la estrella para desmarcarlo. 4. El sistema elimina la relación entre el usuario y el servicio en la base de datos. 5. La estrella vuelve a su estado original (gris) y se muestra un mensaje indicando que el servicio ha sido eliminado de favoritos.
Precondición:	El usuario debe haber iniciado sesión y el servicio debe estar previamente marcado como favorito.
Postcondición:	El servicio ya no aparece en la lista de favoritos del usuario.

Tabla 4.6: Caso de uso: Eliminar servicio de favoritos

4.7. CU7: Consultar pasos del servicio

Actor principal:	Usuario
Descripción:	Muestra los pasos que debe seguir el usuario para completar el servicio seleccionado.
Flujo principal:	<ol style="list-style-type: none"> 1. El usuario hace clic en un servicio. 2. El sistema muestra los pasos detallados.
Precondición:	Servicio seleccionado.
Postcondición:	El usuario visualiza la guía del servicio.

Tabla 4.7: Caso de uso: Consultar pasos del servicio

4.8. CU8: Seleccionar fecha y hora

Actor principal:	Usuario
-------------------------	---------

Descripción:	Permite al usuario elegir la fecha y hora en la que desea que se realice el servicio seleccionado. Este paso se realiza tras consultar los pasos del servicio.
---------------------	--

Flujo principal:	<ol style="list-style-type: none">1. El usuario accede a la página principal y hace clic sobre un servicio.2. El sistema redirige a la sección steps, donde se muestran los pasos del servicio, junto con la descripción y el precio.3. El usuario pulsa el botón “Continuar” en steps.4. El sistema redirige al usuario a la sección date.5. En date, se muestra un calendario interactivo y un campo de selección de hora.6. El usuario elige la fecha y la hora deseadas para la realización del servicio.7. El sistema guarda temporalmente esa información para vincularla con el pedido.
-------------------------	--

Precondición:	El usuario debe haber iniciado sesión y haber seleccionado un servicio.
----------------------	---

Postcondición:	El sistema registra internamente la fecha y hora elegidas como parte de la configuración del pedido.
-----------------------	--

Tabla 4.8: Caso de uso: Seleccionar fecha y hora

4.9. CU9: Adjuntar autorización

Actor principal:	Usuario
Descripción:	El usuario adjunta una autorización necesaria para determinados servicios.
Flujo principal:	<ol style="list-style-type: none">1. El usuario sube un archivo PDF o imagen.2. El sistema lo almacena asociado al pedido.
Precondición:	Servicio que requiere autorización.
Postcondición:	La autorización queda adjunta al pedido.

Tabla 4.9: Caso de uso: Adjuntar autorización

4.10. CU10: Añadir servicio al carrito

Actor principal:	Usuario
Descripción:	Añade un servicio ya configurado al carrito de compras.
Flujo principal:	<ol style="list-style-type: none">1. El usuario confirma la selección del servicio.2. El sistema lo añade al carrito.
Precondición:	El servicio debe estar completo (fecha, hora, autorización si aplica).
Postcondición:	El servicio se encuentra en espera de pago.

Tabla 4.10: Caso de uso: Añadir servicio al carrito

4.11. CU11: Añadir tarjeta

Actor principal:	Usuario
Descripción:	Permite almacenar una tarjeta de crédito o débito.
Flujo principal:	<ol style="list-style-type: none">1. El usuario accede al monedero.2. Introduce el número y la fecha de caducidad.3. El sistema valida y guarda la tarjeta.
Precondición:	Usuario autenticado.
Postcondición:	La tarjeta queda disponible para futuros pagos.

Tabla 4.11: Caso de uso: Añadir tarjeta

4.12. CU12: Realizar pago

Actor principal:	Usuario
Descripción:	Completa el pedido pagando los servicios del carrito.
Flujo principal:	<ol style="list-style-type: none">1. El usuario accede al carrito.2. Revisa el total y selecciona una tarjeta.3. Confirma el pago.4. El sistema registra el pedido.
Precondición:	El usuario debe tener una tarjeta añadida.
Postcondición:	El pedido queda confirmado y registrado.

Tabla 4.12: Caso de uso: Realizar pago

4.13. CU13: Consultar historial de pedidos

Actor principal: Usuario

Descripción: Permite ver el historial de servicios solicitados, tanto activos como pasados.

Flujo principal:

1. El usuario accede a su historial.
2. El sistema muestra los pedidos agrupados por estado.

Precondición: Usuario autenticado.

Postcondición: El usuario visualiza todos sus pedidos.

Tabla 4.13: Caso de uso: Consultar historial de pedidos

4.14. CU14: Modificar datos personales

Actor principal: Usuario

Descripción: Permite modificar datos personales como nombre, dirección, email o contraseña.

Flujo principal:

1. El usuario accede a su perfil.
2. Edita los campos deseados.
3. El sistema guarda los cambios.

Precondición: Usuario autenticado.

Postcondición: La información personal se actualiza correctamente.

Tabla 4.14: Caso de uso: Modificar datos personales

Base de Datos

La base de datos de la aplicación ha sido implementada mediante **MySQL**, utilizando el entorno de gestión **phpMyAdmin** incluido en el paquete **XAMPP**. Este entorno ha facilitado el diseño, administración y prueba de las tablas necesarias para el correcto funcionamiento de la plataforma.

El modelo de datos se ha estructurado en torno a **cuatro tablas principales**, interrelacionadas mediante claves primarias y foráneas, lo que asegura la integridad referencial y la eficiencia en las consultas. A continuación, se detallan las tablas que conforman la base de datos:

- **users**: contiene la información principal de los usuarios registrados. Incluye los atributos:
 - **dni** (*clave primaria*): identificador único del usuario.
 - **name**, **surname**, **address**: datos personales del usuario.
 - **email** y **username** (*claves foráneas*): utilizados para asociar registros con otras tablas.
 - **password**: contraseña encriptada y hasheada del usuario.
 - **verification_code** y **verified**: empleados en el proceso de verificación de cuenta por correo electrónico.
- **services**: almacena los servicios ofrecidos por la plataforma. Sus campos son:
 - **id** (*clave primaria*): identificador del servicio.
 - **service_name**: nombre del servicio.
 - **price**: precio asociado.
 - **steps**: pasos a seguir para completar el servicio (almacenados como texto estructurado).

- **cards:** guarda la información de las tarjetas asociadas a cada usuario para la gestión de pagos. Incluye:
 - `card_id` (*clave primaria*): identificador de la tarjeta.
 - `card_number`: número de tarjeta.
 - `expiration_date`: fecha de caducidad.
 - `dni_user` (*clave foránea*): enlaza con el usuario propietario de la tarjeta.
- **orders:** recoge la información relativa a los pedidos realizados. Está formada por:
 - `order_id` (*clave primaria*): identificador del pedido.
 - `dni` (*clave foránea*): usuario que ha realizado el pedido.
 - `service`: nombre del servicio solicitado.
 - `date` y `time`: fecha y hora seleccionadas para la prestación del servicio.
 - `created_in`: momento exacto en que se creó el pedido.

Este modelo de datos relacional permite una gestión clara y coherente de la información, facilitando la escalabilidad del sistema y el control de accesos, pedidos y servicios contratados por los usuarios. La estructura ha sido diseñada teniendo en cuenta los requisitos funcionales de la aplicación y la seguridad en la gestión de datos personales y financieros.

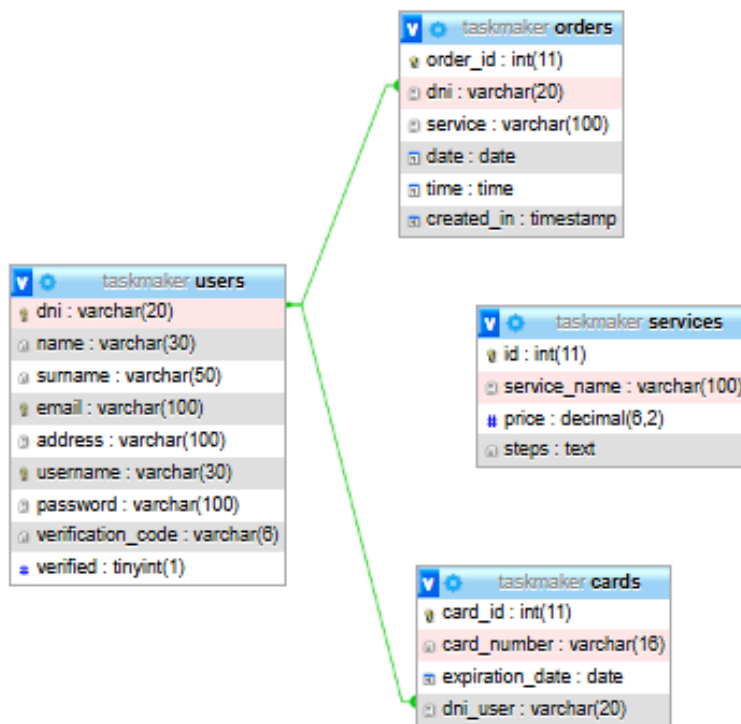


Figura 5.1: Tablas Base de Datos Taskmaker

Conclusiones y Trabajo Futuro

Conclusiones

El desarrollo de la aplicación web **Taskmaker** ha permitido abordar de forma práctica y transversal un conjunto amplio de competencias adquiridas a lo largo del grado en Ingeniería Informática, integrando conocimientos clave en diseño de interfaces, desarrollo frontend y backend, arquitectura de bases de datos relacionales, seguridad y experiencia de usuario. El objetivo principal del proyecto ha sido construir una plataforma funcional y accesible que permita a los usuarios contratar servicios del día a día —como pasar la ITV, realizar trámites administrativos, recoger a los niños del colegio o gestionar recados— de manera cómoda, rápida y centralizada.

Una de las principales fortalezas del proyecto ha sido la implementación de una aplicación web completamente **responsive**, diseñada con un enfoque *mobile-first*. Esto garantiza una experiencia de usuario óptima desde cualquier dispositivo, especialmente smartphones, lo que se alinea con las tendencias actuales de consumo digital. Gracias a esta estrategia de diseño, Taskmaker no requiere instalación adicional y puede utilizarse directamente desde el navegador, lo que facilita su adopción por parte de los usuarios.

Desde el punto de vista técnico, la plataforma se ha construido empleando un stack de tecnologías ampliamente utilizadas y contrastadas por los usuarios: HTML, CSS, JavaScript, PHP y MySQL. El entorno de desarrollo local ha sido gestionado mediante XAMPP y phpMyAdmin, permitiendo un control eficiente del servidor y la base de datos durante todo el ciclo de desarrollo. La arquitectura del sistema, basada en la separación entre cliente y servidor, facilita la mantenibilidad del código y su futura evolución.

La base de datos ha sido diseñada siguiendo criterios de integridad, eficiencia y escalabilidad, y está compuesta por cuatro tablas principales: “users”, “services”,

“cards” y “orders”. Estas entidades permiten almacenar y relacionar de forma estructurada toda la información necesaria para el funcionamiento de la plataforma, incluyendo usuarios, servicios disponibles, métodos de pago y pedidos realizados. Además, se han incorporado funcionalidades como la verificación de cuentas por correo electrónico, la recuperación de contraseñas, la gestión de favoritos, la posibilidad de adjuntar autorizaciones, el sistema de carrito y el historial de pedidos, todo ello orientado a una experiencia de uso sencilla y eficiente.

A nivel funcional y de interfaz, el proyecto ha priorizado la usabilidad, con flujos intuitivos, mensajes de validación claros y un diseño visual consistente. Esto ha permitido construir una aplicación sólida, alineada con los principios del diseño web moderno, capaz de guiar al usuario a través de todas las etapas necesarias para contratar un servicio sin fricciones innecesarias.

Líneas de trabajo futuro

Aunque Taskmaker ya presenta un sistema funcional y operativo con todas las funcionalidades básicas implementadas, existen diversas líneas de mejora y ampliación que permitirían reforzar la plataforma y prepararla para un posible despliegue real, sin que ello implique deficiencias en el trabajo realizado, sino como una evolución natural del producto:

- **Empaquetado como aplicación móvil híbrida:** dado que la aplicación es completamente responsive, se podría adaptar fácilmente para su publicación en tiendas de aplicaciones mediante frameworks como Capacitor o React Native WebView, lo que permitiría incluir funcionalidades como notificaciones push o acceso a recursos del dispositivo.
- **Integración de una pasarela de pagos real:** aunque actualmente se gestiona la información de tarjetas, integrar servicios como Stripe o PayPal permitiría procesar pagos reales de forma segura y profesionalizar la plataforma.
- **Panel de administración:** desarrollar un área interna para la gestión de usuarios, pedidos, servicios y estadísticas facilitaría el control de la plataforma y permitiría su operación en un contexto empresarial.
- **Sistema de valoraciones y reputación:** permitir a los usuarios puntuar los servicios recibidos fomentaría la transparencia y mejoraría la calidad del servicio a través de feedback directo.
- **Geolocalización:** la integración de funciones de localización permitiría optimizar la asignación de tareas por proximidad, estimar tiempos de llegada y ofrecer una experiencia más personalizada y eficiente.
- **Despliegue en servidores cloud:** subir la plataforma a servicios como AWS, Vercel o Heroku permitiría su acceso público y su validación en un entorno real, con escalabilidad automática y monitorización del rendimiento.

- **Cumplimiento normativo y privacidad:** para garantizar la protección de los datos personales, sería necesario implementar políticas de privacidad y mecanismos que aseguren el cumplimiento del Reglamento General de Protección de Datos (RGPD).

En definitiva, **Taskmaker** constituye una base sólida para una plataforma digital de servicios bajo demanda. El proyecto desarrollado no solo cumple con los objetivos establecidos al inicio del trabajo, sino que ha superado las expectativas iniciales al convertirse en una aplicación accesible desde cualquier dispositivo móvil, con un flujo de usuario completo, seguro y escalable. Este trabajo representa una experiencia enriquecedora que ha permitido combinar habilidades técnicas, organizativas y analíticas, sentando las bases para futuros desarrollos profesionales en el ámbito de la transformación digital y la innovación en servicios.

Introduction

“Success is the sum of small efforts repeated day in and day out”

— Robert Collier

This section presents the development of **Taskmaker**, a web application designed to facilitate the management of various personal services in a simple and efficient way. It explains the context and motivation behind the project, its main goals, the planned workflow, and the undergraduate subjects that proved especially useful during its development.

6.1. Background

Digital transformation has led to an environment where managing and contracting services through web platforms has become a common practice for both businesses and individual users. The convenience of accessing a service catalog, placing orders, making payments, and managing interactions all within a single digital environment has redefined consumption and service models. This trend has driven the development of increasingly comprehensive, accessible, and user-centered web applications.

However, in this context of rapid growth, many platforms still show significant shortcomings: lack of personalization, unintuitive interfaces, poor integration between modules, or a fragmented user experience. Moreover, for new entrepreneurs or small businesses wishing to digitize their service offerings, existing solutions are often complex or poorly suited to their specific needs.

Taskmaker was created to address this gap — a project aimed at delivering a simple, modular, and efficient platform for personalized digital service management. This system was developed as part of the Final Degree Project in Computer Engineering, with the goal of putting into practice the knowledge acquired in web design, full-stack development, databases, security, and user experience.

The purpose of this project is not only to build a functional solution but also to face the challenges of designing a complete system in terms of organization, scalability, validation, and usability — all of which are key aspects for professional growth in software engineering.

6.2. Objectives

The general objective of this Final Degree Project is to design and develop a functional web application called **Taskmaker**, allowing users to comprehensively, simply, and efficiently manage the contracting of digital services.

The specific objectives defined to achieve this goal are:

- Analyze the system's functional and non-functional requirements.
- Design a modular architecture based on the client-server model.
- Develop a responsive, clear, and intuitive user interface.
- Implement basic functionalities such as registration, login, account recovery, and email verification.
- Integrate a service cart, payment simulation, and personalized favorites system.
- Allow access to a history of orders and the customization of the user experience.
- Design and maintain a relational MySQL database to efficiently store system data.
- Validate the application through functional tests, user testing, and behavior analysis.

Meeting these objectives will result in a robust, functional, and scalable solution that demonstrates the technical maturity achieved by the end of the degree.

6.3. Work Plan

To ensure the successful progression of the project, the work has been structured into sequential phases, allowing the process to be divided into manageable and organized blocks:

1. **Requirements Analysis Phase:** Functional and non-functional requirements were collected and defined, establishing the system's scope and prioritizing functionalities.
2. **System Design Phase:** The technical architecture of the project was developed, including the database structure (entity-relationship model), UML diagrams (use cases, class and sequence diagrams), and navigation wireframes for the interface.
3. **Development and Implementation Phase:** The different modules were programmed, starting with the backend in PHP and the MySQL database, and later integrating the frontend using HTML, CSS, and JavaScript. Key functionalities such as authentication, service cart, favorites, order history, email verification, and account recovery were implemented during this phase.
4. **Integration and Validation Phase:** Functional tests were conducted to verify system behavior, along with user testing aimed at evaluating the experience and identifying potential improvements.
5. **Documentation Phase:** The final report was drafted, including a description of the technical development, the tests performed, and a critical analysis of the results. Technical documentation was also prepared for possible deployment or future evolution of the system.

This plan was designed to reflect a realistic development cycle, allowing the application of good software engineering practices and efficient management of available resources and time.

6.4. Technologies Used

The development of the **Taskmaker** application required the use of several technologies to create an interactive and dynamic user experience, as well as effective data management. Below are the main technologies used in the creation of the application.

6.4.1. HTML (HyperText Markup Language)

HTML is the standard language for creating web pages. It was used to structure the content and layout of elements on each page of the application. HTML forms the foundation for all visible elements in the user interface, such as registration forms, action buttons, tables, and other essential components. It also enables links between different sections of the application, improving navigation. MDN Web Docs (2025b)



Figure 6.1: HTML Logo

6.4.2. CSS (Cascading Style Sheets)

CSS was used to style and design the web pages, allowing customization of the appearance of HTML elements. CSS helped achieve an attractive interface, with colors, fonts, margins, and spacing that enhance user experience. It was also used to make the application **responsive**, adapting the layout to be accessible on mobile devices and various screen sizes. Media queries and other advanced CSS features were used to adjust the layout dynamically. MDN Web Docs (2025a)



Figure 6.2: CSS Logo

6.4.3. JavaScript

JavaScript is a programming language used to make the application more interactive and dynamic. In **Taskmaker**, JavaScript improves user experience in several areas, including:

- Real-time form validation (e.g., checking empty fields or email format in the registration form).
- User interaction via events, such as button clicks and service card selections.
- DOM manipulation to update page content without reloading (via AJAX).
- Implementation of interactive features like the calendar and time picker on the `date.html` page.
- Management of the shopping cart and selected services, enabling a smooth user experience when adding or removing services. MDN Web Docs (2025c)



Figure 6.3: JavaScript Logo

JavaScript also facilitates integration with APIs and external services, enhancing the application's functionality.

6.4.4. PHP (Hypertext Preprocessor)

PHP is a server-side programming language used to manage business logic and database interaction. In **Taskmaker**, PHP is used for:

- Processing registration, login, and password recovery forms, verifying credentials, and sending data to the database.
- Handling payment operations and order creation.
- Running SQL queries to interact with the database and retrieve necessary information.
- Performing server-side validations to ensure data security and integrity. The PHP Group (2025)



Figure 6.4: PHP Logo

PHP connects the frontend of the application to the database, allowing it to operate dynamically and efficiently.

6.4.5. phpMyAdmin and XAMPP

phpMyAdmin is a web-based MySQL database management tool. In **Taskmaker**, it was used to create and manage the application's database.

- The database stores user data, including names, addresses, encrypted passwords, and other account-related details.
- It manages order transactions, including service information, date, time, and order status.
- phpMyAdmin enables table creation, SQL query execution, and table relationship management. phpMyAdmin Team (2025)



Figure 6.5: phpMyAdmin Logo

XAMPP is a free software package that provides a local server environment.

- XAMPP enables local execution of the application, simulating the production environment without the need for an external server.
- It allows testing and debugging in a controlled environment before deployment. Apache Friends (2025)



Figure 6.6: XAMPP Logo

6.5. Conclusion

The combined use of HTML, CSS, JavaScript, PHP, and tools like phpMyAdmin and XAMPP enabled the development of a functional and interactive web application, capable of efficiently managing services and payments while delivering a smooth and engaging user experience. These technologies were key to ensuring data integrity and security, as well as optimizing the interface for end users.

Conclusions and Future Work

Conclusions

The development of the **Taskmaker** web application has provided a practical and interdisciplinary opportunity to apply a broad range of competencies acquired throughout the Computer Engineering degree. It has successfully integrated key knowledge areas such as user interface design, frontend and backend development, relational database architecture, cybersecurity, and user experience design. The primary objective of this project was to create a functional and accessible platform that enables users to request everyday services—such as vehicle inspections (ITV), administrative errands, school pick-ups, or general task handling—in a convenient, fast, and centralized manner.

One of the main strengths of the project is the implementation of a fully responsive web application, designed with a mobile-first approach. This ensures an optimal user experience across all devices, particularly smartphones, in line with current digital consumption trends. Thanks to this design strategy, Taskmaker requires no additional installation and can be accessed directly through the browser, which facilitates user adoption.

From a technical perspective, the platform has been built using a reliable and widely adopted technology stack: HTML, CSS, JavaScript, PHP, and MySQL. The local development environment was managed with XAMPP and phpMyAdmin, enabling efficient server and database control throughout the development cycle. The system architecture, based on the separation of frontend and backend components, promotes maintainability and supports future scalability.

The database was designed with criteria of integrity, efficiency, and scalability in mind, and consists of four main tables: *users*, *services*, *cards*, and *orders*. These entities allow structured storage and relationships for all necessary information, including user data, available services, payment methods, and service orders.

In addition, features such as email account verification, password recovery, favorites management, file attachment for authorizations, a service cart, and order

history have been implemented—all aimed at delivering a streamlined and efficient user experience.

From a usability and interface standpoint, the project emphasizes user-friendly design, with intuitive flows, clear validation messages, and a consistent visual structure. These elements contribute to a robust application aligned with modern web development principles, capable of guiding users through every stage of the service booking process without unnecessary friction.

Future Work

Although Taskmaker already offers a fully functional system with all core features implemented, several enhancements and extensions could strengthen the platform and prepare it for a potential real-world deployment. These suggestions represent a natural evolution of the product rather than addressing shortcomings:

- **Packaging as a hybrid mobile application:** Given its responsive design, the application could easily be adapted and published on app stores using frameworks such as Capacitor or React Native WebView. This would also enable features like push notifications and access to device resources.
- **Integration of a real payment gateway:** While card information is currently managed internally, integrating solutions like Stripe or PayPal would enable secure and professional real-time payment processing.
- **Administrative panel:** Developing an internal dashboard for managing users, services, orders, and usage statistics would improve operational oversight and allow platform management in a business context.
- **User ratings and reputation system:** Allowing users to rate the services received would encourage transparency and promote continuous improvement based on direct feedback.
- **Geolocation features:** Integrating location services would optimize task assignments based on proximity, estimate arrival times, and offer a more personalized and efficient experience.
- **Deployment on cloud servers:** Hosting the platform on cloud providers such as AWS, Vercel, or Heroku would allow public access and testing in a production environment, with benefits such as automatic scalability and performance monitoring.
- **Regulatory compliance and privacy:** To ensure the protection of personal data, it would be necessary to implement privacy policies and mechanisms aligned with the General Data Protection Regulation (GDPR).

In conclusion, **Taskmaker** provides a solid foundation for an on-demand digital service platform. The project not only meets the objectives initially defined but exceeds expectations by delivering a mobile-accessible application with a complete, secure, and scalable user experience. This work has offered a highly enriching experience, combining technical, organizational, and analytical skills, and sets the stage for future professional development in digital transformation and service innovation.

Bibliografía

APACHE FRIENDS. XAMPP - Cross-Platform Apache MySQL PHP Perl. <https://www.apachefriends.org/>, 2025. Último acceso: mayo de 2025.

GLOVOAPP. Glovo - Tu ciudad al instante. <https://www.glovoapp.com/>, 2025. Último acceso: mayo de 2025.

MDN WEB DOCS. CSS - Cascading Style Sheets. <https://developer.mozilla.org/en-US/docs/Web/CSS>, 2025a. Último acceso: mayo de 2025.

MDN WEB DOCS. HTML - HyperText Markup Language. <https://developer.mozilla.org/en-US/docs/Web/HTML>, 2025b. Último acceso: mayo de 2025.

MDN WEB DOCS. JavaScript. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>, 2025c. Último acceso: mayo de 2025.

PHPMYADMIN TEAM. phpMyAdmin. <https://www.phpmyadmin.net/>, 2025. Último acceso: mayo de 2025.

THE PHP GROUP. Php - hypertext preprocessor. <https://www.php.net/>, 2025. Último acceso: mayo de 2025.

UBER TECHNOLOGIES INC. Uber eats - comida a domicilio. <https://www.ubereats.com/>, 2025. Último acceso: mayo de 2025.

Guía de Uso

Instrucciones para probar la aplicación

Para poder probar la aplicación, deberán seguir los siguientes pasos:

1. Descargar e instalar **XAMPP** en su ordenador.
2. Activar los módulos de **Apache** y **MySQL** desde el panel de control de XAMPP.
3. Importar el fichero correspondiente a la base de datos que se encuentra en el **Apéndice A**.
4. Obtener la dirección **IP local** de su ordenador. Para ello, pueden utilizar el comando `ipconfig` (Windows) o `ifconfig` (Mac/Linux) en la terminal.
5. Introducir en el navegador de su dispositivo móvil la siguiente dirección, sustituyendo por su IP local:

`xxx.xxx.x.xx/Taskmaker/register.html`

6. Una vez cargada la aplicación, pueden:
 - Acceder directamente con el usuario de prueba:
 - **Nombre de usuario:** Test2025
 - **Contraseña:** Test2025
 - O bien registrarse como un nuevo usuario, completando el formulario de registro y verificando su cuenta a través del correo electrónico.

El día de la presentación, se facilitará la **IP de mi ordenador personal** para que puedan acceder a la aplicación directamente desde sus móviles.

Apéndice **B**

Repositorio código Taskmaker

El código desarrollado en el trabajo, tanto del *frontend* como del *backend*, se encuentra disponible en el siguiente repositorio de GitHub:

<https://github.com/Javisanz14/TFG-TASKMAKER>

Dado que el repositorio es de acceso privado, será necesario disponer de una cuenta de GitHub y solicitarme permiso para poder visualizar su contenido.

—*Sancho amigo, ¿qué piensas de nuestras aventuras?*

—*Señor, pienso que más que aventuras, son desventuras.*

—*No digas eso, Sancho. Cada lance es una oportunidad para mostrar nuestro valor.*

—*Pues yo preferiría mostrar mi valor en una buena posada, con vino y pan.*

—*Eres un alma sencilla, Sancho, pero noble. Sigamos nuestro camino, que la gloria nos espera.*

Segunda parte del Ingenioso Caballero

Don Quijote de la Mancha

Miguel de Cervantes

