



Universidad de Valladolid

**ESCUELA DE INGENIERÍA INFORMÁTICA
DE SEGOVIA**

**Grado en Ingeniería Informática
de Servicios y Aplicaciones**

**TPVa: Aplicación Web de Terminal de
Punto de Venta para gasolineras**

Alumno: Carlos Bernaldo de Quirós Sanz

**Tutores: Juan José Álvarez Sánchez
José Miguel Moreno López**

El éxito no es definitivo.

El fracaso no es fatal.

Es el coraje de continuar lo que cuenta.

Agradecimientos

A mi familia por enseñarme a no rendirme nunca y apoyarme siempre.

A Carlota, por estar conmigo siempre, pase lo que pase.

A José Miguel, por animarme a terminar y compartir su extenso conocimiento y tan buenos ratos conmigo.

A mi tutor, Juanjo, por todo lo que nos ha enseñado y por ser siempre tan cercano y amable con todos sus alumnos.

Y a mis compañeros y ahora amigos, en especial a mi “pareja de baile” en todos los trabajos durante la carrera.

A todos, muchas gracias de corazón.

Resumen

Los Terminales Punto de Venta (TPV) se han convertido en un componente esencial de cualquier comercio, ya sea minorista o mayorista. En la actualidad, existen multitud de TPVs comerciales y gratuitas para tiendas generalistas. Sin embargo, los negocios más específicos requieren de soluciones a medida adaptadas a sus necesidades particulares.

El objetivo de este proyecto es el desarrollo de una aplicación web de venta que permita la gestión de estaciones de servicio, de sus operaciones, productos y empleados, sin la necesidad de la instalación de hardware específico. Para su realización, se ha empleado Angular como framework de desarrollo para la interfaz gráfica, combinado un conjunto de tecnologías en la nube, tales como Firebase y Cloudflare. Este stack tecnológico ha permitido la creación de una solución completamente *serverless* y altamente escalable.

Palabras clave: Terminal Punto de Venta (TPV), estaciones de servicio, Angular, serverless

Abstract

Point of Sale Terminals (POS) have become an essential component of any business, whether retail or wholesale. Currently, there are a wide range of commercial and free POS terminals for general stores. However, more specific businesses require customized solutions adapted to their particular needs.

The goal of this project is the development of a web-based sales application that allows the management of service stations, their operations, products and employees, without the need for the installation of specific hardware. For its realization, Angular has been used as a development framework for the user interface, combined with a set of cloud technologies, such as Firebase and Cloudflare. This technological stack has allowed the creation of a completely serverless and highly scalable solution.

Keywords: Point of Sale (POS), service stations, Angular, serverless

Índice de contenidos

1. Introducción	11
1.1. Motivación	11
1.2. Estado del arte	12
1.3. Objetivos y alcance	16
2. Gestión del proyecto	19
2.1. Metodología de trabajo	19
2.2. Planificación	23
2.3. Presupuesto	34
3. Análisis	37
3.1. Características del sistema	37
3.2. Actores	39
3.3. Reglas de negocio	39
3.4. Requisitos de usuario	40
3.5. Requisitos de información	72
3.6. Requisitos no funcionales	76
4. Diseño	77
4.1. Arquitectura lógica	77
4.2. Arquitectura física	79
4.3. Diagramas de secuencia	81
4.4. Diseño de la interfaz	85
5. Implementación	95
5.1. Estructura del proyecto	95
5.2. Detalles de implementación	99
6. Pruebas	107
6.1. Registro e inicio de sesión	107
6.2. Editar productos	108
6.3. Administrar estación	110
6.4. Ventas	113
6.5. Consulta de operaciones y stock	115
7. Documentación	117
7.1. Manual de usuario	117
7.2. Manual de despliegue	142
8. Conclusiones	145
8.1. Futuras mejoras	145
Referencias	147
Índice de tablas	149
Índice de figuras	153

1. Introducción

La administración eficiente de los datos se erige como un pilar fundamental para la viabilidad y progreso de cualquier comercio en la actualidad. En este contexto, la búsqueda constante de herramientas tecnológicas que posibiliten una gestión ágil y efectiva, sin desembolsos desorbitados y con un mantenimiento eficiente, se ha convertido en una prioridad para la gran mayoría de los negocios.

En el universo de las estaciones de servicio, todo esto no es una excepción. Las corporaciones líderes en el mercado destinan considerables recursos financieros al desarrollo de aplicaciones especializadas que supervisen y gestionen las complejas operaciones llevadas a cabo en cada una de sus estaciones. No obstante, las estaciones de servicio independientes, con recursos financieros más limitados, también enfrentan la imperiosa necesidad de contar con herramientas de gestión de datos.

El propósito central de este proyecto radica en la creación de una aplicación web diseñada específicamente para estaciones de servicio independientes y con presupuestos más reducidos. La finalidad es proporcionar a estos negocios una plataforma que les permita gestionar sus operaciones y artículos de manera sencilla, prescindiendo de la necesidad de realizar instalaciones complejas de hardware y software en las propias instalaciones. En este escenario, se busca democratizar el acceso a soluciones tecnológicas, brindando a las estaciones de servicio independientes una herramienta que potencie su eficiencia operativa sin comprometer sus limitados recursos.

1.1. Motivación

La creación de este proyecto encuentra su origen en las conversaciones entre un grupo de compañeros de trabajo, compuesto por desarrolladores e ingenieros de software especializados en la creación y mantenimiento de la aplicación de medios de pago para las estaciones de servicio propias y afiliadas a una de las principales compañías petrolíferas que operan en España. Dicha aplicación se caracteriza “únicamente” por realizar la comunicación efectiva de toda la información relativa a los procesos de pago. Para la gestión de las estaciones se integra dentro de otro tipo de aplicaciones, es decir, con un Terminal de Punto de Venta (TPV) específico para los comercios de la mencionada compañía.

En el transcurso de una de estas conversaciones laborales, surgió una interrogante que despertó mi interés: ¿cómo abordan las estaciones de servicio independientes la gestión de sus operaciones, inventario y logística comercial?, teniendo en cuenta que la gran parte de

estas estaciones son pequeños comercios que no tienen una gran capacidad económica. Esta incógnita, lejos de ser simplemente una pregunta retórica, representó la mecha que inició la creación de este proyecto.

Motivado por el deseo de encontrar soluciones innovadoras y accesibles, me propuse desarrollar una respuesta a esta pregunta. Este proyecto, entonces, nace como una propuesta ágil, económica y eficiente para las estaciones de servicio independientes que buscan optimizar su gestión operativa sin incurrir en costos prohibitivos. Al aspirar a ofrecer una herramienta que se adapte a las necesidades específicas de este sector, el objetivo es impulsar la capacidad de estas estaciones para competir en un entorno cada vez más digitalizado, proporcionándoles una plataforma que simplifique y potencie sus operaciones cotidianas.

1.2. Estado del arte

Las principales soluciones de aplicaciones de Terminal de Punto de Venta (TPV) que podemos encontrar en el mercado según la compañía Merchant Maverick son VeriFone Commander [15], LS Retail [10], NCR VOYIX POS [11], CPista [13] y TPV Garum [1]. A continuación se detallan las características principales de estas soluciones.

1.2.1. VeriFone Commander

- Requiere la instalación de la solución con su hardware específico.
- Integración completa para la gestión conjunta de tienda y combustible.
- Informes detallados. Proporciona informes detallados de ventas e inventario para análisis y toma de decisiones.
- Comunicación en tiempo real entre los surtidores de la estación y el TPV (requiere la instalación de hardware y controladores específicos de la solución).
- Funciones de fidelización. Puede incluir programas de fidelización y promociones.
- Gestión de inventario dinámico. Control de inventario y seguimiento de productos en la tienda.
- Seguridad. Incorpora medidas de seguridad para proteger las transacciones.



Figura 1. Captura de pantalla del C18, perteneciente a la solución VeriFone Commander.

1.2.2. LS Retail

- Requiere la instalación de la solución con su hardware específico.
- Integración completa para la gestión conjunta de tienda y combustible.
- Personalización. Puede adaptarse a diversos tipos de estaciones de servicio.
- Análisis en tiempo real. Ofrece análisis en tiempo real de las operaciones y stock para la toma de decisiones.
- Gestión de fidelización. Integrado con distintas soluciones que ofrecen programas de fidelización y promociones.

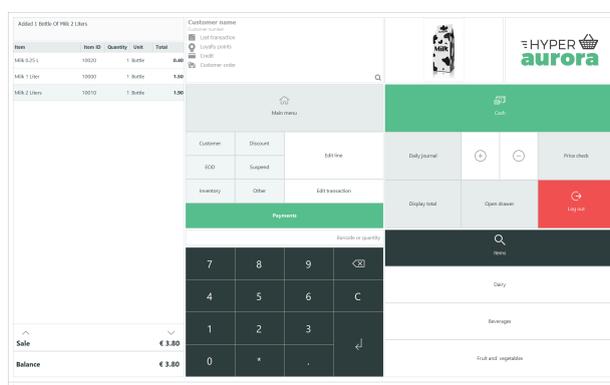


Figura 2. Captura de pantalla de LS Retail.

1.2.3. NCR VOYIX POS (antiguo Radiant)

- Requiere la instalación de la solución con su hardware específico.
- Integración completa para la gestión conjunta de tienda y combustible.
- Integración con distintas soluciones de medios de pago y descuentos.
- Gestión de inventario dinámico. Permite el alta, baja y modificación de los distintos productos y su eficiente gestión desde la propia aplicación.
- Comunicación en tiempo real entre los surtidores de la estación y el TPV (requiere la instalación de hardware y controladores específicos de la solución).

- Informes detallados. Proporciona informes detallados de ventas e inventario para análisis y toma de decisiones.
- Personalizable, ya que puede adaptarse a las necesidades específicas de la estación de servicio, aunque no se hace de forma dinámica desde el TPV.
- Seguridad. Incluye medidas de seguridad para proteger las transacciones y los datos del cliente.

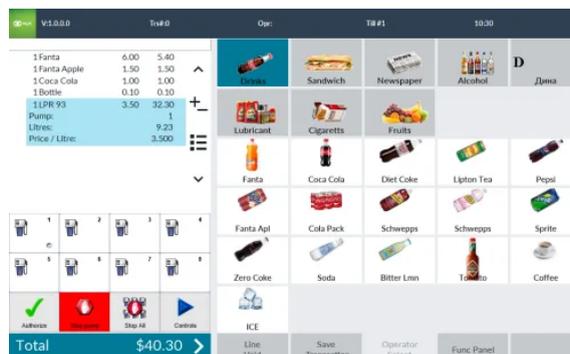


Figura 3. Captura de pantalla de NCR VOYIX POS.

1.2.4. CPista

- Requiere la instalación de la solución con su hardware específico.
- Integración completa para la gestión conjunta de tienda y combustible.
- Integración con distintas soluciones de medios de pago y descuentos (SOLRED, MEROIL, VALCARCE).
- Comunicación en tiempo real entre los surtidores de la estación y el TPV (requiere la instalación de hardware y controladores específicos de la solución).



Figura 4. Captura de pantalla de CPista.

1.2.5. TPV Garum

- Requiere la instalación de la solución con su hardware específico.
- Integración completa para la gestión conjunta de tienda y combustible.
- Integración con la solución de medios de pago y descuentos de SOLRED.

- Cuenta con una solución para su dispositivo portátil que permite la gestión del punto de venta desde cualquier punto de la estación.
- Personalización dinámica de los tickets de venta de impresión.



Figura 5. Captura de pantalla de TPV Garum.

1.2.6. Comparativa

Teniendo en cuenta estos datos sobre cada uno de ellos, en la Tabla 1 se muestra una comparativa de los distintos productos software mencionados en el estado del arte.

Tabla 1. Comparativa entre las distintas soluciones de TPV más comunes en el mercado.

	VeriFone Commander	LS Retail	NCR VOYIX POS	CPista	TPV Garum
Gestión y visualización de operaciones de ventas	Sí	Sí	Sí	Sí	Sí
Integración de gestión de tienda y combustibles	Sí	Sí	Sí	Sí	Sí
Requiere hardware específico	Sí	Sí	Sí	Sí	Sí
Integración con solución de medios de pago	Sí	Sí	Sí	Sí	Sí
Comunicación con surtidores en tiempo real	Sí	Sí	Sí	Sí, se integra con distintos controladores	No
Configuración dinámica de los datos de la estación	No	Sí	No	No	No
Informes de operaciones detallados	Sí	Sí	Sí	No	No
Gestión de inventario dinámica	Sí	No	Sí	No	No
Versatilidad para ejecutarse en distintos dispositivos	No*	No	No	No	No*

Respecto al último punto de la Tabla 1 (marcado con el símbolo *), las soluciones “VeriFoneCommander” y “TPV Garum” —aunque únicamente se puedan ejecutar en el

dispositivo para el que están diseñadas—, cuentan con otras soluciones adicionales que permiten ejecutarse en otros dispositivos portátiles, aunque con algunas de sus características limitadas.

En este contexto, la singularidad y fortaleza de la aplicación que se presenta en este proyecto radica en su capacidad para proporcionar una solución de gestión completamente automatizada para estaciones de servicio, prescindiendo por completo de la necesidad de realizar actuaciones de instalación en el establecimiento. Esto se logra mediante la traslación de todos los procesos de gestión al *cloud* o nube, permitiendo a los usuarios acceder y operar la aplicación desde cualquier lugar con conexión a Internet, incluso utilizando un dispositivo tan común como un teléfono móvil.

La ventaja estratégica de esta aplicación reside en su enfoque innovador, al liberar a las estaciones de servicio de las restricciones tradicionales asociadas con la instalación de hardware específico. La eliminación de este requisito no solo reduce significativamente los costos iniciales de implementación, sino que también minimiza los gastos recurrentes de mantenimiento, aliviando la carga financiera que tradicionalmente acompaña a la adopción de soluciones tecnológicas avanzadas.

Además, al posibilitar la gestión remota mediante dispositivos comunes como teléfonos móviles, la aplicación proporciona flexibilidad y movilidad a los gestores de estaciones de servicio, permitiéndoles supervisar y gestionar las operaciones desde cualquier lugar con acceso a internet. Esta innovadora perspectiva representa un cambio significativo en la forma en que las estaciones de servicio pueden abordar la gestión de sus datos, al tiempo que potencia la eficiencia operativa y la adaptabilidad en un entorno empresarial dinámico.

1.3. Objetivos y alcance

Para definir el alcance de este proyecto hay que entender el entorno en el que nos movemos. En la actualidad, según los datos recopilados en la Memoria Anual de la Asociación de Operadores Petrolíferos (AOP) en España, se identifican un total de 12.084 Estaciones de Servicio, cada una caracterizada por su propio modelo de gestión [3]. Estas estaciones se clasifican en distintos tipos, y resulta fundamental comprender las particularidades de cada una para abordar de manera específica las necesidades de gestión de datos en este contexto diversificado.

1.3.1. Tipos de estaciones

Estaciones libres: Las estaciones de servicio de este tipo se distinguen por la autonomía en la gestión de suministro, imagen y explotación. La libertad de suministro permite a los

empresarios la adquisición de carburantes de diversos proveedores, generalmente a través de CLH, en función de las condiciones y requisitos de cada momento. Este enfoque proporciona flexibilidad y opciones estratégicas para los dueños de estas estaciones.

CO-CO (Company Owned – Company Operated): En este escenario, las estaciones son propiedad directa de la petrolera. La empresa gestiona todos los aspectos, desde la propiedad del suelo hasta la operación diaria. La definición de márgenes y precios está bajo control total de la petrolera propietaria.

DO-CO (Dealer Owned – Company Operated): La propiedad de la instalación recae en un particular o sociedad, pero la gestión operativa se delega a la petrolera mediante un contrato de arrendamiento de la instalación. Este modelo fusiona la propiedad privada con la operación dirigida por la petrolera.

DO-DO (Dealer Owned – Dealer Operated): En este caso, tanto la propiedad como la operación de la instalación están en manos de una sociedad independiente. Mantienen un contrato de suministro exclusivo con la petrolera, regulando los márgenes, comisiones y gastos asociados.

CO-DO (Company Owned – Dealer Operated): La petrolera es propietaria de la instalación, pero la gestión operativa se confía a un tercero mediante un contrato que establece los términos de imagen y suministro exclusivo. Este modelo permite a terceros operar bajo la marca de la petrolera.

1.3.2. Definición de objetivos

En el marco de este trabajo, nos enfocaremos específicamente en las **Estaciones libres**. Este enfoque se justifica por el hecho de que, para los otros tipos de estaciones, la empresa petrolífera, ya sea como propietaria o proveedora de imagen comercial, es responsable de implementar su propia aplicación de Terminal de Punto de Venta. Estas estaciones son las que necesitan buscar en el mercado aquellas herramientas o aplicaciones que le permitan gestionar correctamente los datos de su comercio.

En síntesis, el objetivo principal de este proyecto es proporcionar a las estaciones de servicio libres una aplicación de punto de venta (TPV) que sea sencilla, eficiente y económicamente accesible. La singularidad de esta propuesta radica en su capacidad para prescindir de instalaciones previas, permitiendo su uso en diversos dispositivos.

Si bien, el alcance inicial se centra en el desarrollo de una aplicación versátil que aborde las operaciones fundamentales de una estación de servicio, como son la gestión de combustibles y productos de tienda y contabilidad de las transacciones, en un futuro el proyecto podría adaptarse a soluciones específicas para grandes compañías petrolíferas. La flexibilidad de la

aplicación permitirá una integración efectiva con los sistemas existentes y una respuesta ágil a las particularidades de cada proveedor, especialmente en lo que respecta a los terminales de medios de pago que utilicen cada uno de ellos.

En resumen, este proyecto aspira a ofrecer a las estaciones de servicio una solución de TPV avanzada, económica y fácil de implementar, con la visión de evolucionar y adaptarse a las demandas específicas de la industria petrolífera en un futuro próximo. La propuesta no solo busca satisfacer las necesidades operativas actuales, sino también posicionarse como una solución integral y flexible en el panorama de los sistemas de punto de venta.

2. Gestión del proyecto

En este apartado detallaremos la metodología que se ha seguido para el desarrollo de la aplicación, la estimación tanto del tamaño total del producto como del tiempo invertido para la completa realización del proyecto, así como la planificación que se ha seguido para poder completarse con los medios disponibles y en el tiempo más óptimo.

2.1. Metodología de trabajo

La realización del proyecto se basará en una metodología iterativa e incremental. Esto quiere decir que se creará en primer lugar un prototipo inicial con las características más básicas de un TPV y a partir de este prototipo se irán añadiendo las nuevas funcionalidades con cada incremento. Al mismo tiempo, se mejoran y se van adaptando aquellas funcionalidades ya existentes en base a los nuevos requerimientos añadidos (véase Figura 6).

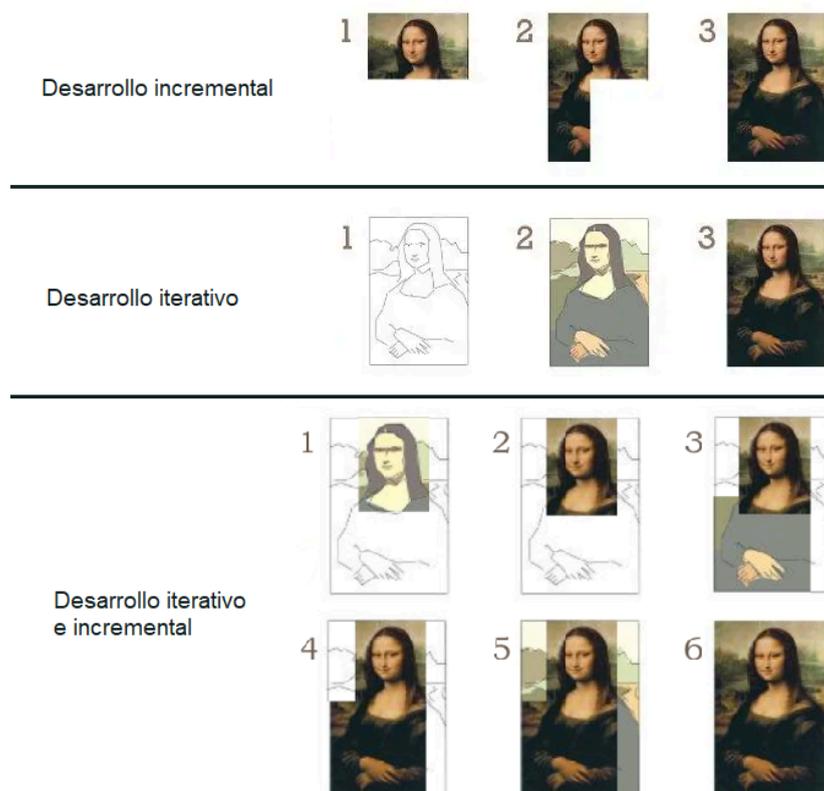


Figura 6. Comparativa de metodologías de desarrollo.

Como se ha comentado, el desarrollo se va a dividir en distintas fases que incluyen nuevas funcionalidades (incremento) y adaptaciones y mejoras de las funcionalidades ya existentes

(iteración), aunque a cada una de las evoluciones de la solución la llamaremos simplemente “incremento”.

La solución la vamos a dividir en 7 fases, una primera de creación del prototipo inicial con las características básicas, 5 incrementos que irán añadiendo funcionalidad a la aplicación y mejorando y adaptando las ya existentes y una última parte que será la documentación del proyecto, la cual se llevará a cabo durante todo el desarrollo de codificación del software. A continuación se detalla cada una de las tareas a realizar:

Creación del prototipo: Esta es la parte más importante y con más peso dentro del desarrollo de la solución. En ella crearemos la estructura básica de nuestra pantalla principal, así como el flujo de venta básico con datos insertados manualmente y directamente en el código. Además del propio desarrollo, en esta parte llevará implícito el aprendizaje de herramientas como Firebase y Cloudflare, así como de los servicios que estas ofrecen y su configuración para el desarrollo y el despliegue automatizado mediante Cloudflare Pages [4] y Cloudflare Workers [5]. Dentro de estos servicios, se creará la estructura básica y conexión de la base de datos de Firebase para poder acceder en siguientes incrementos.

Primer incremento: Este incremento priorizará la gestión de usuarios. Aquí se crearán las pantallas de acceso y registro de usuarios, así como la pantalla para recuperar la contraseña. Se creará y configurará la conexión con Firebase Authentication [8] para poder manejar el control de sesiones de forma óptima además de añadir el guardado de la información de usuarios en nuestra base de datos de Firebase (no almacenaremos los datos sensibles, se ello se encarga de forma segura Firebase Authentication).

Segundo incremento: Para este segundo incremento nos centraremos en la gestión propia de los productos. Se añadirán pantallas y formularios para dar de alta, modificar y eliminar productos, tanto de tienda como combustibles. Se creará la estructura básica para mostrar y seleccionar todos los productos y añadirlos al ticket de venta en el flujo previamente creado.

Tercer incremento: Al igual que en el incremento anterior, en este se añadirán las pantallas para crear, modificar y eliminar surtidores. Además, se añadirá la funcionalidad para el manejo y navegación dentro de la región visual de surtidores. También, se añadirá la selección de surtidor y combustible al flujo de venta creado con anterioridad.

Cuarto incremento: En este incremento se añadirán las funcionalidades para gestionar los datos de la estación. Para ello también se deberán crear las pantallas para mostrar y modificar dichos datos. Además, dentro de este incremento tendrá lugar la parte más compleja técnicamente, ya que se añadirá la jerarquía de usuarios. Este desarrollo lleva implícito el control de acceso a distintas partes de la aplicación, así como la creación de pantallas para el alta de usuarios no propietarios, la modificación de sus datos y la eliminación de estos datos.

Quinto incremento: Este será el último incremento, y con él se dará por finalizado (por el momento) el desarrollo de todas las funcionalidades de la aplicación. En este se implementará la consulta de stock y la consulta de operaciones, con sus respectivas pantallas. Dentro de estas se añadirá la funcionalidad para exportar los datos a Excel. Por último, se añadirá el flujo de devolución de operaciones, que se iniciará desde la pantalla de consulta de operaciones.

Documentación del proyecto: Siguiendo buenas prácticas de gestión de proyecto y desarrollo, la documentación se realizará durante todos los incrementos y en ella se incluyen tanto los comentarios dentro del código, como la guía y manejo de la aplicación.

2.1.1. Herramientas y tecnologías de desarrollo

Las herramientas para desarrollar y desplegar la solución final que se han usado durante la realización del proyecto han sido las siguientes:

- **VS Code:** Editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma (Windows, GNU/Linux, macOS y web). Cuenta con una grandísima cantidad de extensiones que facilitan su uso y permite escribir y ejecutar cualquier lenguaje de programación. Además cuenta con una muy buena integración con Git y cuenta con programas de refactorización para TypeScript.
- **GitHub:** Servicio propiedad de Microsoft basado en la nube que permite el alojamiento de proyectos software mediante el software de control de versiones Git. Cuenta con la aplicación gratuita de código abierto GitHub Desktop, que básicamente es una interfaz que permite manejar de manera sencilla los repositorios de GitHub.
- **GitHub Actions:** Es una plataforma desde la que se puede realizar la integración y despliegue continuo, lo cual permite la reducción y la automatización de las tareas necesarias para la ejecución del código mediante un flujo de trabajo previamente definido y configurado por el usuario. Esta herramienta permite la compilación, ejecución de test unitarios y despliegue del código, en respuesta a uno o varios eventos que se definen con anterioridad.
- **Cloudflare Pages:** Es un servicio de alojamiento web que permite el despliegue de aplicaciones frontend estáticas o que sigan el patrón Jamstack [12], otorgando a la aplicación de una gran escalabilidad y seguridad. Cuenta con integración con repositorios en GitHub, lo cual permite la puesta en producción automática una vez se ha actualizado el código fuente en el repositorio. En su combinación con Cloudflare Workers otorga la posibilidad de crear aplicaciones *serverless*, lo que libera al usuario de la configuración del aprovisionamiento y administración de los recursos del servidor.

- **Cloudflare Workers:** Es una herramienta de Cloudflare que permite el desarrollo de aplicaciones *serverless* mediante el despliegue de funciones JavaScript que han sido desarrolladas por el usuario y que se ejecutan en la nube a través del motor V8 de Google Chrome cada vez que se recibe el tipo de solicitud especificado para cada una de ellas.

Las tecnologías escogidas para ser empleadas en el stack de desarrollo del producto final han sido las siguientes:

- **JavaScript:** Es un lenguaje de programación interpretado, de secuencia de comandos, basado en prototipos, orientado a objetos y débilmente tipado, desarrollado a partir del estándar ECMAScript. Se puede implementar tanto en el cliente para agregar dinamismo a las páginas web como en el servidor.
- **TypeScript:** Lenguaje libre de código abierto desarrollado por Microsoft. Es una extensión del lenguaje de programación JavaScript que amplía la sintaxis de tipos, lo cual otorga más claridad en el código. Una vez compilado, es traducido todo a JavaScript, por lo que tiene la misma flexibilidad de uso que este.
- **Angular:** *Framework* de desarrollo de código abierto mantenido por Google y construido sobre TypeScript. Cuenta con una biblioteca bien integrada que cubre una gran variedad de servicios, además de un conjunto de herramientas para desarrollar, compilar, probar y actualizar el código. Está basado en componentes, lo que permite una buena escalabilidad al código y facilita la creación de aplicaciones web de página única.
- **SCSS:** Lenguaje de hojas de estilo que extiende la sintaxis de CSS, que permite, por ejemplo, el uso de variables, anidación e importación de archivos, otorgando al usuario más facilidad y claridad en el desarrollo y reduciendo las líneas necesarias para dar estilo a la página.
- **Bootstrap:** *Framework* de *frontend* gratuito y de código abierto (FOSS) que contiene un conjunto de herramientas y plantillas que facilitan la construcción de vistas de todo tipo de aplicaciones web.
- **Firebase:** Plataforma en la nube desarrollada por Google que facilita el desarrollo de aplicaciones web y móviles. Dispone de un gran número de servicios para el desarrollo de forma gratuita entre los que se encuentran la autenticación de usuarios y el realtime database, los cuales facilitan la gestión de usuarios y sesiones y el almacenamiento en bases de datos NoSQL de acceso rápido. Además, también cuenta con servicios de monetización para añadir anuncios y de análisis.

2.2. Planificación

La planificación del proyecto se ha realizado en dos partes. Primero se ha hecho una estimación del proyecto, tanto del tamaño del mismo como del tiempo de codificación y de finalización del proyecto, incluida la documentación. Después se ha desglosado el proyecto en tareas, teniendo en cuenta los incrementos señalados en la metodología y a cada una de ellas se le ha otorgado un tiempo de desarrollo estimado, teniendo en cuenta la estimación del proyecto previa, para obtener el diagrama de Gantt.

2.2.1. Estimación

Para calcular de una manera fiable el tamaño total del proyecto se ha realizado la estimación por Puntos de Función (PF). En este caso se ha utilizado el método Albrecht, ya que permite medir la funcionalidad entregada al usuario final para la ejecución del software independientemente de la tecnología utilizada para la construcción.

2.2.1.1. Identificación de componentes y su complejidad

- Entradas externas: para el cálculo de su complejidad nos basaremos en la Tabla 2.
 - Registrar usuario administrador. (Media)
 - Recuperar contraseña de usuario administrador/empleador. (Baja)
 - Autenticar usuario administrador/empleador. (Baja)
 - Modificar datos de estación. (Baja)
 - Añadir usuario empleado. (Media)
 - Modificar usuario administrador/empleador. (Media)
 - Eliminar usuario empleado. (Baja)
 - Añadir producto de tienda. (Baja)
 - Modificar producto de tienda. (Baja)
 - Eliminar producto de tienda. (Baja)
 - Añadir combustible. (Baja)
 - Modificar combustible. (Baja)
 - Eliminar combustible. (Baja)
 - Añadir surtidor. (Baja)
 - Modificar surtidor. (Media)
 - Eliminar surtidor. (Baja)
 - Añadir productos al ticket de venta. (Media)
 - Modificar productos en el ticket de venta. (Baja)
 - Eliminar productos del ticket de venta. (Baja)
 - Seleccionar surtidor. (Baja)
 - Seleccionar combustible por surtidor. (Media)

- Seleccionar medio de pago de la venta. (Baja)
- Confirmar pago de la venta. (Baja)

Tabla 2. Criterios de evaluación de entradas externas.

Ficheros referenciados	Datos elementales		
	1-4	5-15	16 o más
0-1	Baja	Baja	Media
2	Baja	Media	Alta
3 o más	Media	Alta	Alta

- Salidas externas: para el cálculo de su complejidad nos basaremos en la Tabla 3.
 - Generar ticket de venta. (Alta)
 - Generar excel de ventas. (Baja)
 - Generar excel de stock de productos. (Baja)
 - Menú de surtidores. (Media)
 - Listado de productos de tienda. (Baja)

Tabla 3. Criterios de evaluación de salidas externas.

Ficheros referenciados	Datos elementales		
	1-5	6-19	20 o más
0-1	Baja	Baja	Media
2-3	Baja	Media	Alta
4 o más	Media	Alta	Alta

- Consultas: para el cálculo de su complejidad nos basaremos en la Tabla 4.
 - Listado de combustibles por surtidor. (Baja)
 - Listado de combustibles. (Baja)
 - Detalles de producto de tienda/combustible. (Baja)
 - Listado de administración de surtidores. (Baja)
 - Detalle de surtidores. (Baja)
 - Listado de usuarios. (Baja)
 - Detalles de usuario. (Baja)
 - Listado de ventas. (Baja)
 - Listado de stock de productos. (Baja)
 - Detalle de datos de estación. (Baja)

Tabla 4. Criterios de evaluación de consultas.

Ficheros referenciados	Datos elementales		
	1-5	6-19	20 o más
0-1	Baja	Baja	Media
2-3	Baja	Media	Alta
4 o más	Media	Alta	Alta

- Ficheros lógicos internos: para el cálculo de su complejidad nos basaremos en la Tabla 5.
 - Base de datos de la aplicación. (Media)

Tabla 5. Criterios de evaluación de ficheros lógicos internos.

Registros elementales	Datos elementales		
	1-19	20-50	51 o más
0-1	Baja	Baja	Media
2-5	Baja	Media	Alta
6 o más	Media	Alta	Alta

2.2.1.2. Tabla de Puntos de Función sin ajustar

Una vez hemos identificado los componentes y su complejidad, podemos sacar el total de Puntos de Función sin ajustar, tal y como se puede observar en la Tabla 6.

Tabla 6. Puntos de función sin ajustar.

Componentes	Complejidad	Cantidad	Valor	Resultado
Entradas externas	Alta	0	x6	0
	Media	6	x4	24
	Baja	17	x3	51
Salidas Externas	Alta	1	x7	7
	Media	1	x5	5
	Baja	3	x4	12
Consultas	Alta	0	x7	0
	Media	0	x5	0
	Baja	10	x4	40
Ficheros lógicos internos	Alta	0	x15	0
	Media	1	x10	10
	Baja	0	x7	0
PUNTOS DE FUNCIÓN NO AJUSTADOS				144

2.2.1.3. Evaluación de los Factores de Ajuste

Para poder obtener los Puntos de Función ajustados debemos primero evaluar la influencia de los distintos Factores de Ajuste dentro de la aplicación. Estos factores y su influencia vienen especificados en la Tabla 7.

Tabla 7. Evaluación de Factores de Ajuste.

Factor de ajuste	Influencia
1. Comunicación de datos	0
2. Funciones distribuidas	0
3. Rendimiento	2
4. Configuración sobrecargada	0
5. Velocidad de las transacciones	2
6. Entrada de datos en línea	5
7. Diseño para la eficiencia del usuario final	2
8. Actualización de datos en línea	4
9. Complejidad del proceso lógico interno de la aplicación	3
10. Reusabilidad del código	1
11. Facilidad de instalación	0
12. Facilidad de operación	0
13. Usabilidad múltiple	2
14. Facilidad en el cambio	2

Con estos datos podemos obtener el Factor de Ajuste a través de la siguiente fórmula, donde $\sum FC$ es la suma de la influencia de cada uno de los Factores de Ajuste de la Tabla 7.

$$FA = (0,01 * \sum FC) + 0,65 = (0,01 * 23) + 0,65 = 0,88$$

2.2.1.4. Ajuste de los puntos de función y obtención de líneas de código

Obtenido el Factor de Ajuste ya podemos calcular los Puntos de Función Ajustados (PFA) a través de la siguiente fórmula.

$$PFA = PFNA * FA = 144 * 0,88 = 126,72$$

La aplicación estará diseñada aproximadamente en 70% TypeScript y un 30% JavaScript. Si consultamos las líneas de código por punto de función que corresponden para JavaScript son 55 LDC/PF. Para TypeScript no existe esta información, pero al ser un lenguaje basado en JavaScript y debido a su similitud asumimos las mismas líneas de código por punto de función, por lo que el cálculo de las líneas de código quedaría de la siguiente forma.

$$Total\ de\ LDC = PFA * 55 = 126,72 * 55 = 6969,6\ LDC$$

2.2.1.5. Cálculo del tiempo en días de esfuerzo

Para poder obtener el tiempo total de codificación del proyecto daremos una estimación de líneas de código escritas por hora. Teniendo en cuenta mi experiencia profesional de casi 5 años en la ingeniería de software y mi experiencia con los lenguajes TypeScript y JavaScript estimamos que la media aproximada será de 52 líneas de código por hora¹. Por lo que el tiempo total de codificación sería el siguiente:

$$\textit{Tiempo de codificación} = 6969,6 \textit{ LDC} / 52 \textit{ líneas por hora} = 132,75 \textit{ horas}$$

En cuanto al tiempo total estimado de duración del proyecto, tomaremos en cuenta que la codificación supondrá el 40% del esfuerzo total y el resto será tiempo dedicado al análisis y documentación. De este modo, el tiempo total del proyecto estimado quedaría así:

$$\textit{Tiempo total del proyecto} = (132,75 \textit{ horas} * 100\%) / 40\% = 331,88 \textit{ horas}$$

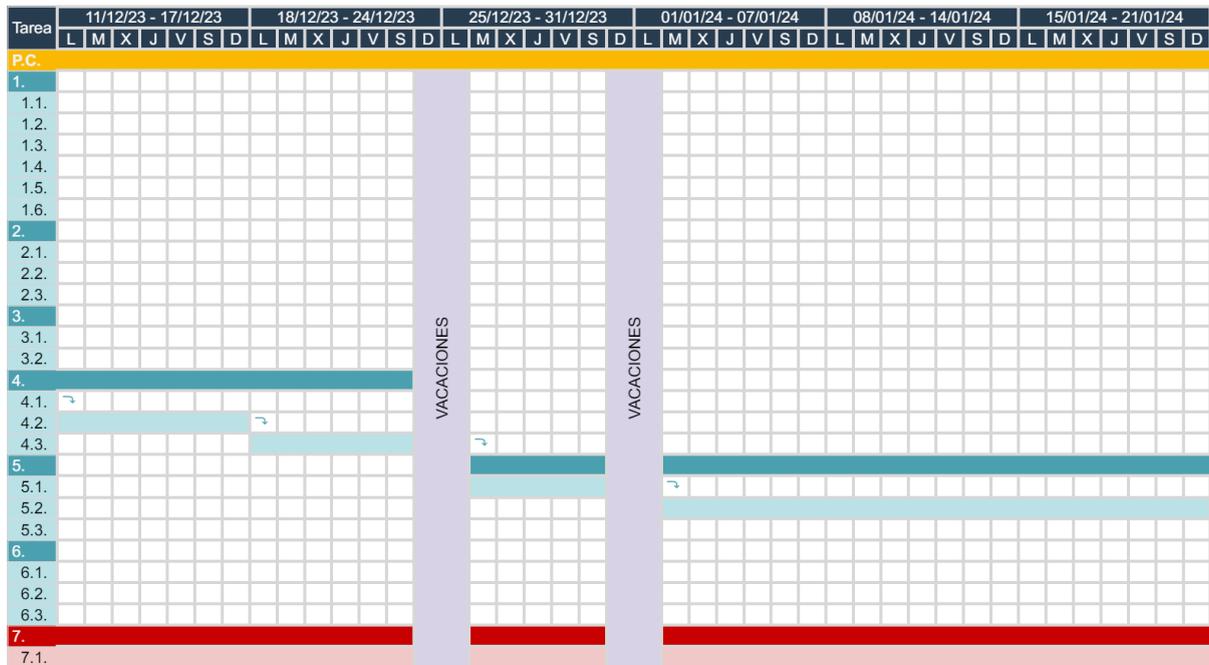
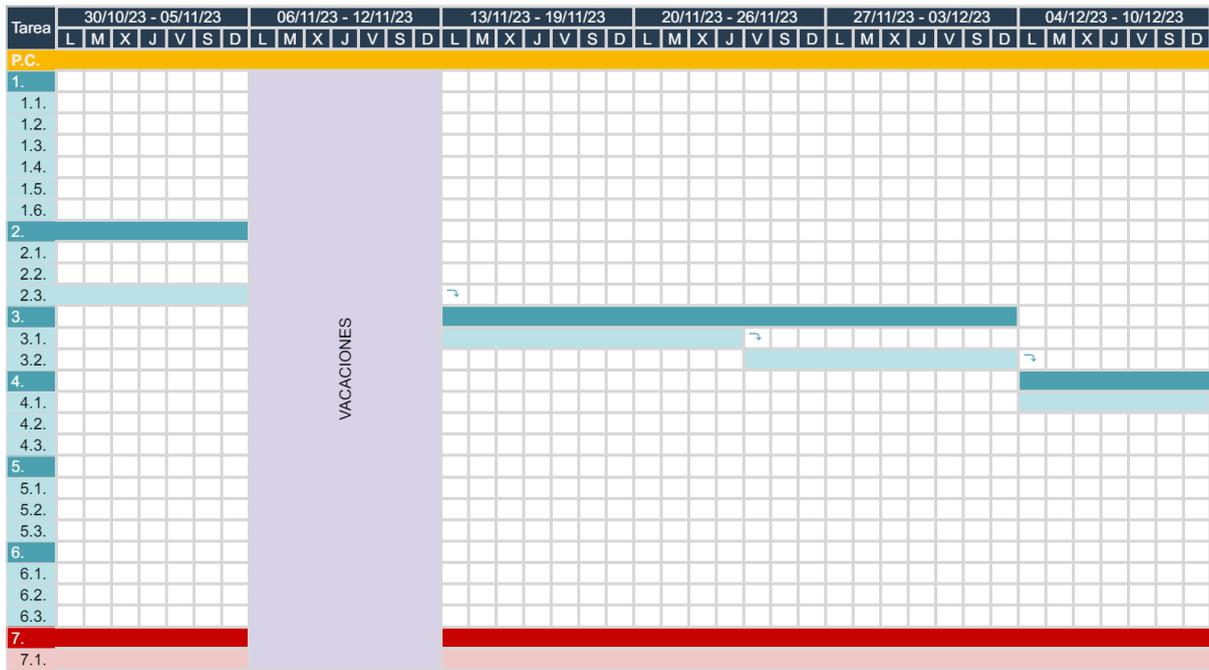
2.2.2. Desglose temporal

Teniendo en cuenta la estimación realizada en el apartado anterior se ha realizado una planificación inicial de realización del proyecto que va desde el 04/09/2023 al 25/02/2024. Esta fecha de finalización la obtenemos estimando una dedicación al proyecto de una media de 2 horas por día, o lo que es lo mismo 14 horas por semana.

Como se ha comentado en el apartado de metodología, se va a dividir el proyecto en 7 partes. Una primera que será la creación del prototipo inicial de la aplicación, 5 incrementos que otorgarán nuevas funcionalidades completas a dicha aplicación hasta completar la solución y un último apartado que será la documentación del proyecto, que se realizará durante todo el desarrollo del mismo. Cada una de estas partes está dividida en subtareas, que son pequeños desarrollos que otorgarán una nueva (o nuevas) funcionalidad a la solución. Será a estas subtareas a las que se les otorgará un tiempo estimado de desarrollo teniendo en cuenta el tiempo estimado para el proyecto.

En la Figura 7 se incluye el diagrama de Gantt que ha dado como resultado.

¹ Esta cifra se calcula en base a la media de las líneas de código programadas por hora en un experimento realizado durante una semana.



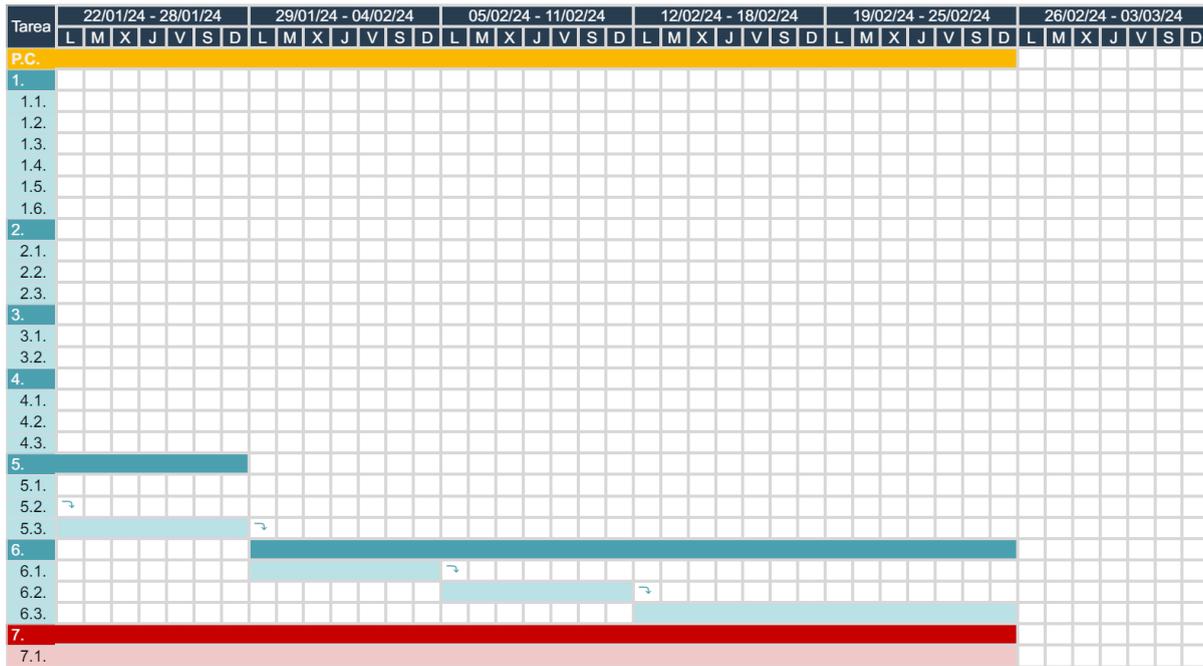
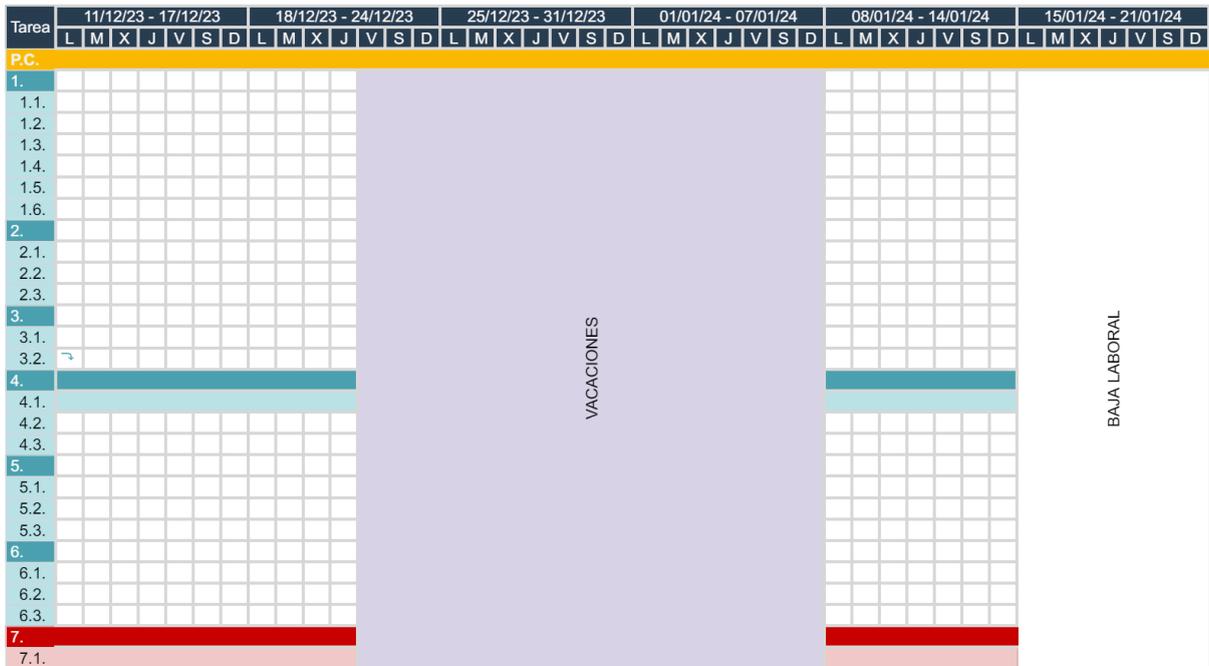
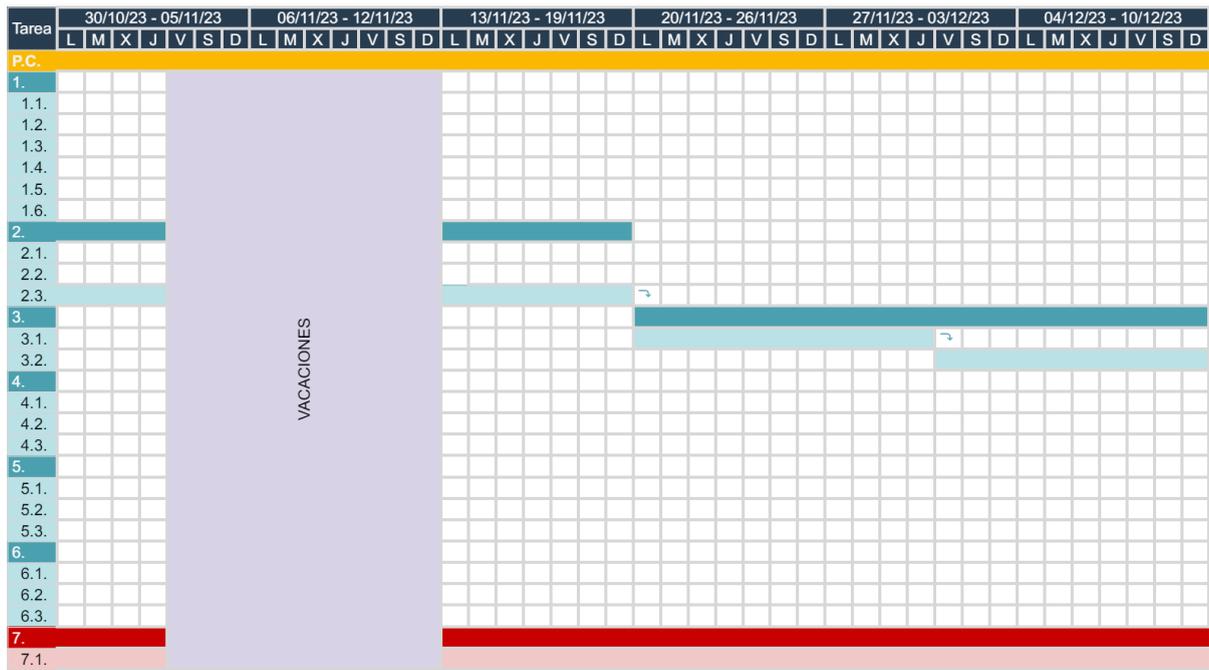


Figura 7. Diagrama de Gantt estimado del proyecto.

Una vez ejecutado el proyecto en su totalidad, se realiza un diagrama de Gantt con las horas reales invertidas y que ha llevado cada una de las tareas (Figura 8).



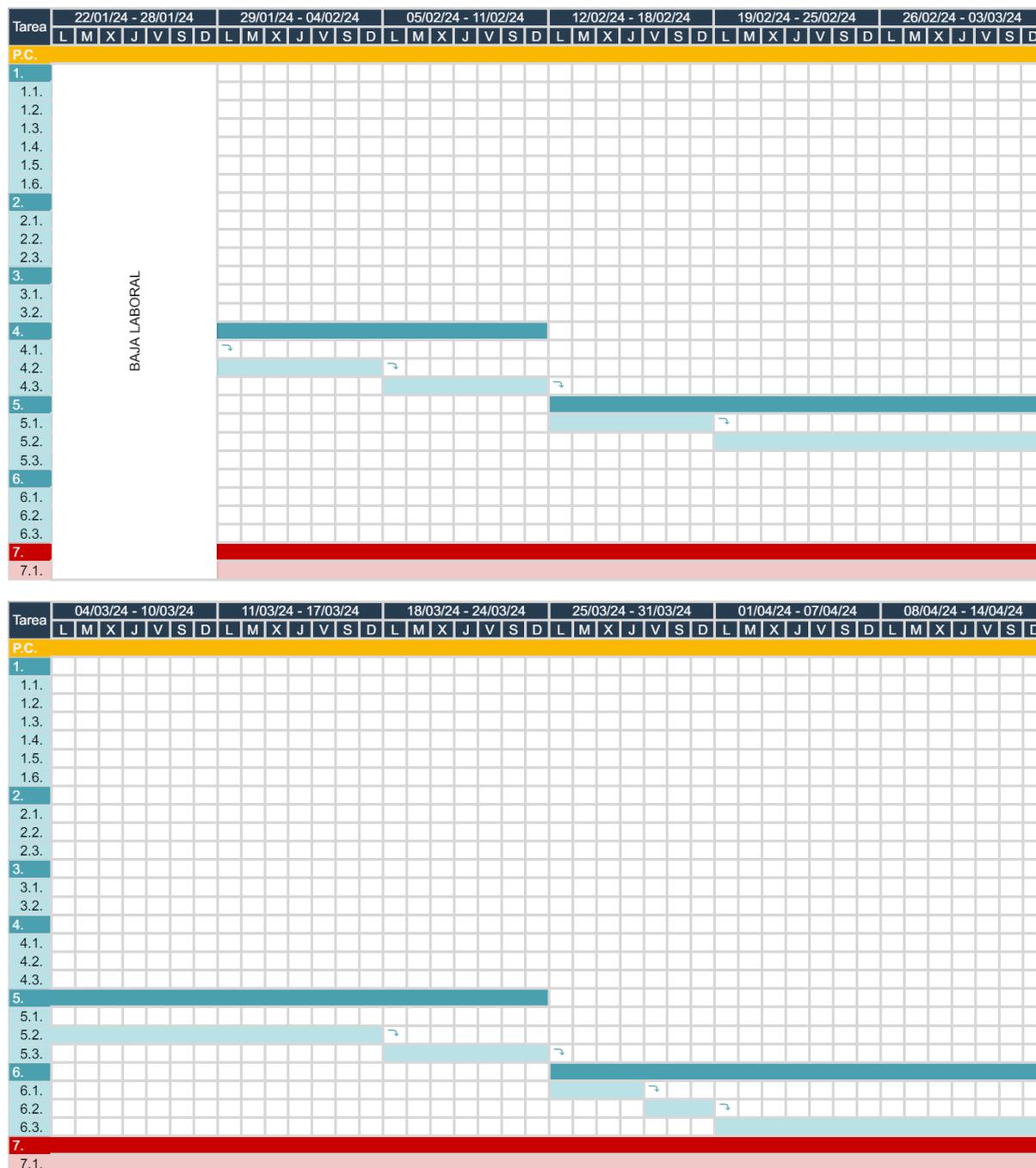


Figura 8. Diagrama de Gantt real de proyecto.

Como se puede observar haciendo una comparación entre el diagrama de Gantt planificado y el diagrama de Gantt real, la gran mayoría de tareas se hicieron en el tiempo planificado para las mismas, aunque debido a imprevistos como fueron la baja laboral entre los días 15/01/2024 y 28/01/2024, las vacaciones entre los días de Navidad y Año Nuevo y la no correcta planificación y extensión en el tiempo de tareas como “2.3. Pantalla y flujo para el acceso de usuarios y control de sesiones”, “4.1. Flujos de alta, modificación y eliminación de surtidores” y “5.2. Flujos de alta, modificación y eliminación de datos de usuarios no propietarios” han retrasado la fecha de finalización al día 14/04/2024. de más, o lo que es lo

mismo, dos semanas más realizando el proyecto, lo que nos permite concluir que la estimación del tiempo y su planificación, para un proyecto tan extenso en el tiempo, fue positiva.

2.3. Presupuesto

Debido a estos inconvenientes, las horas invertidas en el proyecto finalmente ascendieron a un total de 364 horas (sin contar los días de baja y de vacaciones de más), frente a las 331,88 horas estimadas. Esto hace un total de 32,12 horas.

Para obtener el presupuesto total del proyecto dividiremos el mismo en tres partes bien diferenciadas que son hardware, software y personal.

2.3.1. Presupuesto Hardware

En cuanto al hardware se va a utilizar un ordenador personal con una vida útil de 7 años y un precio total (incluidos periféricos) de 959 €. Considerando que de media este PC ha sido usado unas 5 horas diarias, su uso total ascendería a unas 12775 horas. Como hemos visto en el anterior apartado, el tiempo estimado del proyecto es de 331,88 horas, por lo que el uso total en el proyecto del PC en porcentaje será de 2,6 %.

También, debemos tener en consideración la conexión a internet. En este caso se cuenta con un contrato de 55,90 € al mes. Se ha estimado que el tiempo en meses de realización del proyecto es de 5,53 meses, por lo que deberán ser necesarios 6 meses de contratación, es decir, 335,40 € en total. El uso de dicha línea de internet es de un 20 %.

Teniendo en cuenta estos datos podemos calcular el presupuesto hardware como se especifica en la Tabla 8.

Tabla 8. Presupuesto Hardware.

Herramienta	Coste total	Porcentaje de uso	Coste real
Ordenador personal	959,00 €	2,60 %	24,94 €
Conexión a Internet	335,40 €	20,00 %	67,08 €
Presupuesto Hardware			92,02 €

2.3.2. Presupuesto Software

Dentro del presupuesto software, solo se va a utilizar la licencia de pago de Windows 10. Esta licencia tiene un coste total de 100 €. Teniendo en cuenta que su vida útil es la misma

que el ordenador personal (7 años) y que su uso diario también (5 horas diarias) nos queda un uso total de 12775 horas. Al igual que hemos hecho con el ordenador personal, su porcentaje de uso teniendo en cuenta que el proyecto me llevará 331,88 horas será de un 2,6 %.

De este modo el presupuesto software del proyecto lo podemos calcular tal y como viene detallado en la Tabla 9.

Tabla 9. Presupuesto Software.

Herramienta	Coste total	Porcentaje de uso	Coste real
Ordenador personal	100,00 €	2,60 %	2,60 €
Presupuesto Software			2,60 €

2.3.3. Presupuesto de personal

El proyecto será ejecutado por una sola persona, pero se van a desempeñar distintos roles dependiendo del trabajo a desarrollar. De este modo vamos a dividir el desarrollo en tres roles: Analista de Sistemas, Desarrollador Backend y Desarrollador Frontend.

Para poder obtener el presupuesto de cada uno de ellos, primero debemos obtener las horas de trabajo estimadas de cada rol. Dentro del análisis de sistemas hemos estimado en el apartado anterior que se realizarán unas 199,13 horas. Para el desarrollo obtuvimos 132,75 horas. Este desarrollo se va a dividir en parte Back y parte Front. Se estima teniendo en cuenta la solución a desarrollar que la parte Back del desarrollo será del 25% y la parte Front del 75%.

Una vez obtenidas las horas podemos calcular el presupuesto total como se puede ver en la Tabla 10 basándonos en el salario medio en España de dichos roles [14].

Tabla 10. Presupuesto de personal.

Rol	Sueldo por hora	Horas totales	Coste
Analista de Sistemas	16,67 €	199,13	3319,50 €
Desarrollador Backend	15,65 €	33,19	519,42 €
Desarrollador Frontend	17,95 €	99,56	1787,10 €
Presupuesto de personal			5626,02 €

2.3.4. Presupuesto total

Una vez obtenido el presupuesto desglosado podemos obtener el presupuesto total del proyecto como vemos en la Tabla 11.

Tabla 11. Presupuesto total del proyecto.

	Hardware	Software	Personal	Total
Porcentaje	1,60 %	0,05 %	98,35 %	100,00 %
Coste	92,02 €	2,60 €	5626,02 €	5720,64 €

3. Análisis

En este apartado se detallan los aspectos y características que se han tenido en cuenta para el desarrollo completo de la aplicación.

3.1. Características del sistema

Las características del sistema son el conjunto de funcionalidades más básicas que la aplicación debe ofrecer para llegar a completar la solución. A continuación se detalla cada una de ellas:

- **CAR-01 (realizar venta):** Incluye todas las opciones necesarias para llevar a cabo la venta, tanto la selección de cualquier tipo de producto como el método de pago.
 - **CAR-01.1 (seleccionar producto):** Permite seleccionar un producto para añadirlo al ticket de venta.
 - **CAR-01.2 (seleccionar surtidor):** Permite seleccionar el surtidor o surtidores que se han utilizado en la venta.
 - **CAR-01.3 (seleccionar carburante):** Permite seleccionar el carburante de los surtidores utilizados en la venta.
 - **CAR-01.4 (realizar pago):** Permite realizar el pago de la venta en curso, seleccionando en el proceso el método de pago elegido.
- **CAR-02 (editar productos):** Incluye gestionar todos los datos de los productos, tanto de tienda, combustible como las carpetas en las que se distribuyen los productos tienda.
 - **CAR-02.1 (gestionar productos combustible):** Permite añadir, modificar y eliminar los productos de tipo combustible.
 - **CAR-02.2 (gestionar productos tienda):** Permite añadir, modificar y eliminar los productos de tipo tienda.
 - **CAR-02.3 (editar carpetas de productos tienda):** Permite añadir, modificar y eliminar carpetas en las que se van a organizar los productos de tipo tienda.
- **CAR-03 (administrar estación):** Permite gestionar todos los datos propios de la estación, incluyendo los *usuarios dependientes* y los surtidores.
 - **CAR-03.1 (administrar usuarios):** Permite modificar los datos del usuario propietario actual, así como añadir, modificar y eliminar *usuarios dependientes*.
 - **CAR-03.2 (administrar surtidores):** Permite añadir y eliminar surtidores, además de editar los combustibles que va a tener cada uno de ellos.

- **CAR-03.3 (editar datos de estación):** Permite modificar los datos básicos de información de la estación.
- **CAR-04 (consulta de operaciones y stock):** Permite obtener toda la información de las operaciones y del stock de productos, además de realizar devoluciones.
 - **CAR-04.1 (consulta de operaciones):** Permite ver los datos básicos de las operaciones realizadas e importarlos a excel.
 - **CAR-04.2 (consulta de stock):** Permite ver los datos básicos y el stock de los productos dados de alta en la estación e importarlos a excel, así como actualizar el stock de productos.
 - **CAR-04.3 (devolución de operaciones):** Permite realizar la devolución de las operaciones realizadas en la estación y que hayan sido ya devueltas.
- **CAR-05 (gestión de usuarios):** Incluye toda la funcionalidad para permitir al usuario darse de alta, autenticarse (acceder) y recuperar la contraseña.
 - **CAR-05.1 (registrar usuario):** Permite a un usuario darse de alta en la aplicación con una estación por defecto a su nombre.
 - **CAR-05.2 (autenticar usuario):** Permite a un usuario ya registrado acceder a la pantalla principal de la aplicación.
 - **CAR-05.3 (recuperar la contraseña):** Permite al usuario ya registrado recuperar las credenciales en caso de que haya olvidado la contraseña.

3.1.1. Árbol de características

En base a las características definidas anteriormente, se muestra el árbol de características resultante en la Figura 9:

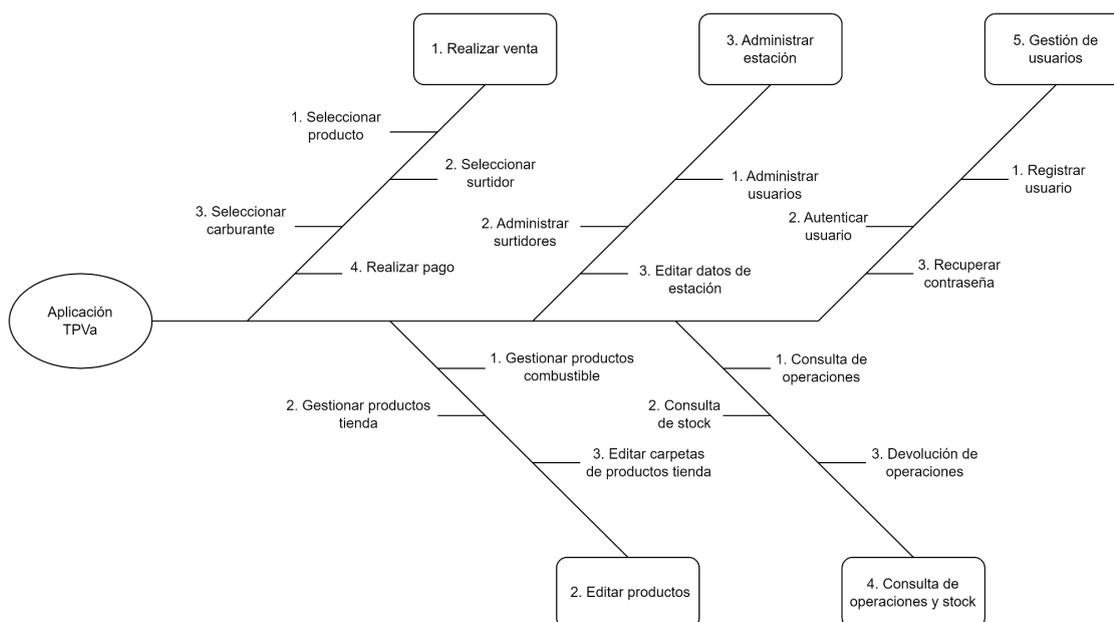


Figura 9. Árbol de características de la aplicación.

3.2. Actores

Incluye a todas las entidades externas que son capaces de interactuar con la aplicación. En nuestro caso estos actores van a ser siempre usuarios capaces de interactuar con alguna de las pantallas e interfaces de las que pone a disposición la aplicación. Se dividen en tres:

ACT-01 (usuario no registrado): Es el actor que tiene el acceso más restringido a la aplicación. Únicamente se le permitirá acceder a la funcionalidad de registro en el sistema.

ACT-02 (usuario dependiente): Este actor cuenta con las funcionalidades de acceso a la aplicación, recuperación de la contraseña, funcionalidad completa de ventas y consulta de operaciones y stock.

ACT-03 (usuario propietario): Este actor es capaz de acceder a la funcionalidad completa de la aplicación, incluido el alta de *usuarios dependientes*, gestión de la estación, consultas de operaciones y stock, realizar la funcionalidad completa de devolución, así como de dar de alta, modificar y eliminar productos.

3.3. Reglas de negocio

Estas son las reglas de negocio definidas previamente, las cuales restringen y crean una base para la definición posterior de requisitos y la implementación de cada una de las funcionalidades.

RN-01: La relación *usuario propietario* y estación será 1:1, es decir, cada estación solo tendrá un *usuario propietario* y cada *usuario propietario* únicamente podrá pertenecer a una estación.

RN-02: El *usuario propietario* es el único que puede dar de alta *usuarios dependientes* en su estación.

RN-03: Los *usuarios dependientes* únicamente podrán realizar las funciones de venta y consulta de operaciones y stock.

RN-04: Cada estación tendrá un máximo de 20 surtidores.

RN-05: No se permite devolver una venta ya devuelta.

RN-06: La devolución de una operación debe ser completa, no se admiten devoluciones parciales.

RN-07: Las devoluciones solamente se podrán realizar en el plazo de 30 días desde que se ha hecho la venta.

RN-08: Si se está en medio de una operación de venta no se permitirá editar productos, administrar la estación ni consultar operaciones.

RN-09: Si se está editando productos, administrando la estación o consultando las operaciones o el stock no se permitirá iniciar el flujo de venta.

El mapeo de las relaciones entre estas reglas de negocio, las características y los actores de la aplicación se recoge en la Tabla 12.

Tabla 12. Relación entre reglas de negocio, actores y características.

Regla de negocio	Características relacionadas	Actores relacionados
RN-01	CAR-05, CAR-05.1	ACT-03
RN-02	CAR-03, CAR-03.1	ACT-02, ACT-03
RN-03	CAR-01, CAR-04.1, CAR-04.2	ACT-02
RN-04	CAR-03, CAR-03.3	ACT-03
RN-05	CAR-04, CAR-04.3	ACT-03
RN-06	CAR-04, CAR-04.3	ACT-03
RN-07	CAR-04, CAR-04.3	ACT-03
RN-08	CAR-01	ACT-02, ACT-03
RN-09	CAR-02, CAR-03, CAR-04	ACT-02, ACT-03

3.4. Requisitos de usuario

Los requisitos de usuario detallan todas aquellas acciones y funcionalidades a las que el usuario puede acceder dentro de la aplicación. Estas se detallan a continuación discriminadas por tipo de usuario:

Usuario no registrado:

RU-01: El usuario no registrado podrá registrarse en la aplicación.

RU-02: El usuario no registrado podrá visualizar la información básica que describe la aplicación.

Usuario registrado (tanto *usuario dependiente* como *usuario propietario*):

RU-03: El usuario registrado podrá iniciar sesión en la aplicación.

RU-04: El usuario registrado podrá cerrar sesión en la aplicación.

RU-05: El usuario registrado podrá obtener una nueva contraseña en el caso de que no la recuerde.

RU-06: El usuario registrado podrá visualizar y navegar en el árbol de *productos tienda* de la estación a la que pertenece.

RU-07: El usuario registrado podrá visualizar y navegar en la lista de surtidores de la estación a la que pertenece.

RU-08: El usuario registrado podrá realizar ventas en la estación a la que pertenece.

RU-09: El usuario registrado podrá visualizar las opciones de “Consulta de operaciones y stock” relativas a su usuario.

RU-10: El usuario registrado podrá consultar las operaciones realizadas en la estación a la que pertenece.

RU-11: El usuario registrado podrá exportar a excel los datos de las operaciones realizadas en la estación a la que pertenece.

RU-12: El usuario registrado podrá consultar el stock de productos ya existentes en la estación a la que pertenece.

RU-13: El usuario registrado podrá actualizar el stock de productos ya existentes en la estación a la que pertenece.

RU-14: El usuario registrado podrá exportar a excel los datos del stock de productos de la estación a la que pertenece.

Usuario propietario:

RU-15: El *usuario propietario* podrá visualizar y seleccionar la lista de tipos de productos (combustible y tienda) a gestionar.

RU-16: El *usuario propietario* podrá visualizar el árbol de *productos tienda* y carpetas que puede gestionar de la estación a la que pertenece.

RU-17: El *usuario propietario* podrá crear *productos tienda* o carpetas de la estación a la que pertenece.

RU-18: El *usuario propietario* podrá modificar *productos tienda* y carpetas de la estación a la que pertenece.

RU-19: El *usuario propietario* podrá eliminar *productos tienda* y carpetas de la estación a la que pertenece.

RU-20: El *usuario propietario* podrá visualizar la lista de *productos combustible* que puede gestionar de la estación a la que pertenece.

RU-21: El *usuario propietario* podrá crear *productos combustible* en la estación a la que pertenece.

RU-22: El *usuario propietario* podrá modificar *productos combustible* en la estación a la que pertenece.

RU-23: El *usuario propietario* podrá eliminar *productos combustible* en la estación a la que pertenece.

RU-24: El *usuario propietario* podrá visualizar y seleccionar las opciones de “Administrar usuarios”, “Administrar surtidores” y “Editar datos de la estación”.

RU-25: El *usuario propietario* podrá visualizar los surtidores que puede gestionar para la estación a la que pertenece.

RU-26: El *usuario propietario* podrá añadir surtidores para la estación a la que pertenece.

RU-27: El *usuario propietario* podrá modificar surtidores para la estación a la que pertenece.

RU-28: El *usuario propietario* podrá eliminar surtidores para la estación a la que pertenece.

RU-29: El *usuario propietario* podrá visualizar la lista de *usuarios dependientes* asociados a la estación a la que pertenece.

RU-30: El *usuario propietario* podrá añadir *usuarios dependientes* asociados a la estación a la que pertenece.

RU-31: El *usuario propietario* podrá modificar los datos de *usuarios dependientes* asociados a la estación a la que pertenece (excepto contraseña) y los datos completos de su usuario.

RU-32: El *usuario propietario* podrá eliminar *usuarios dependientes* asociados a la estación a la que pertenece.

RU-33: El *usuario propietario* podrá visualizar los datos básicos de la estación (nombre y dirección).

RU-34: El *usuario propietario* podrá modificar los datos básicos de la estación a la que pertenece (nombre y dirección).

RU-35: El *usuario propietario* podrá visualizar la opción de “Realizar devoluciones” en la pantalla de “Consulta de operaciones y stock” además de las opciones de “Consultar operaciones” y “Consultar stock”.

RU-36: El *usuario propietario* podrá visualizar las ventas realizadas en la estación a la que pertenece que pueden ser devueltas.

RU-37: El *usuario propietario* podrá realizar la devolución de ventas realizadas en la estación a la que pertenece.

3.4.1. Diagramas de casos de uso

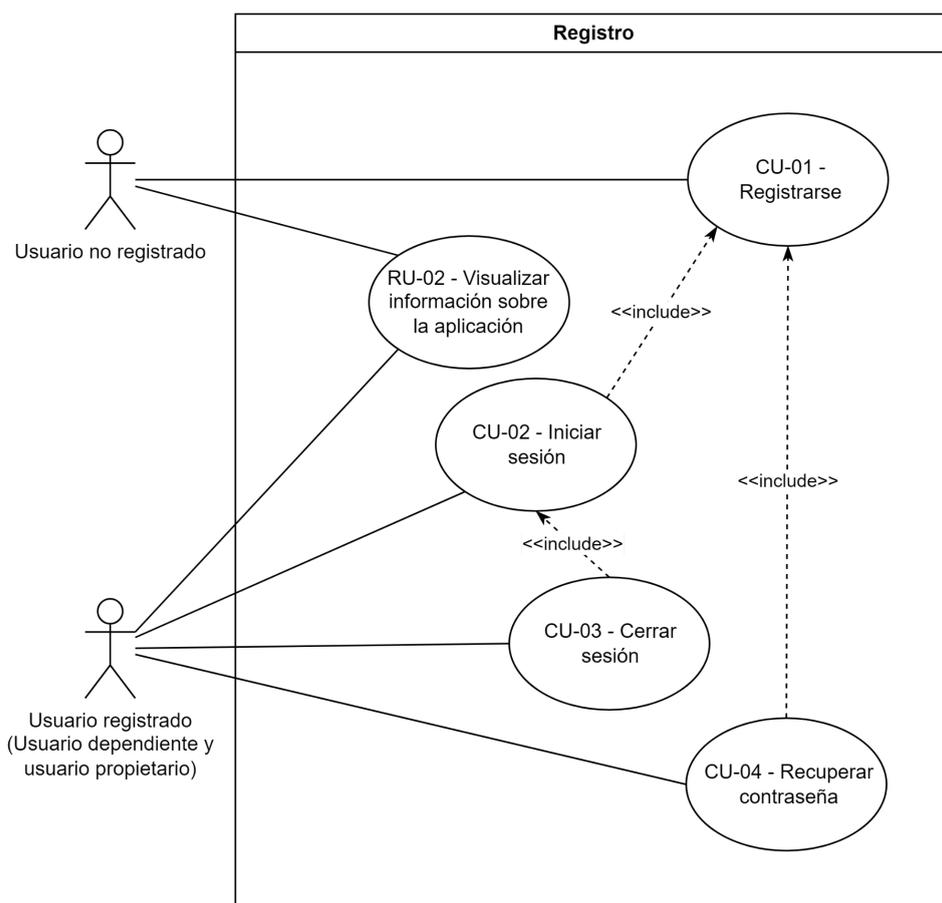


Figura 10. Diagrama de casos de uso "Registro".

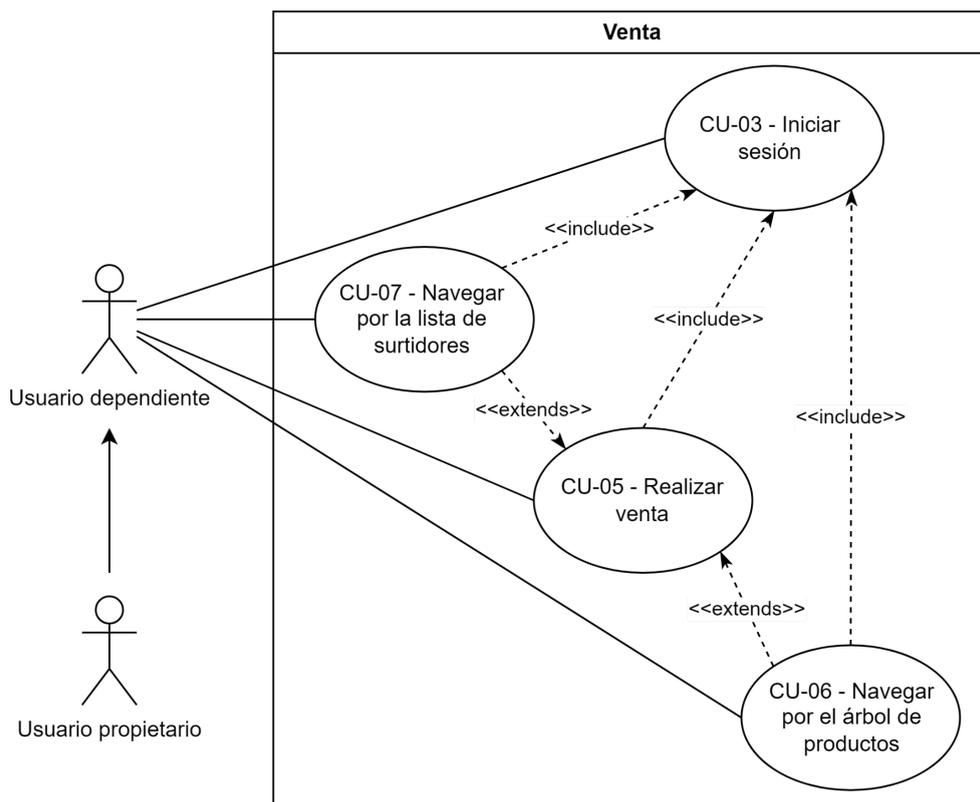


Figura 11. Diagrama de casos de uso “Venta”.

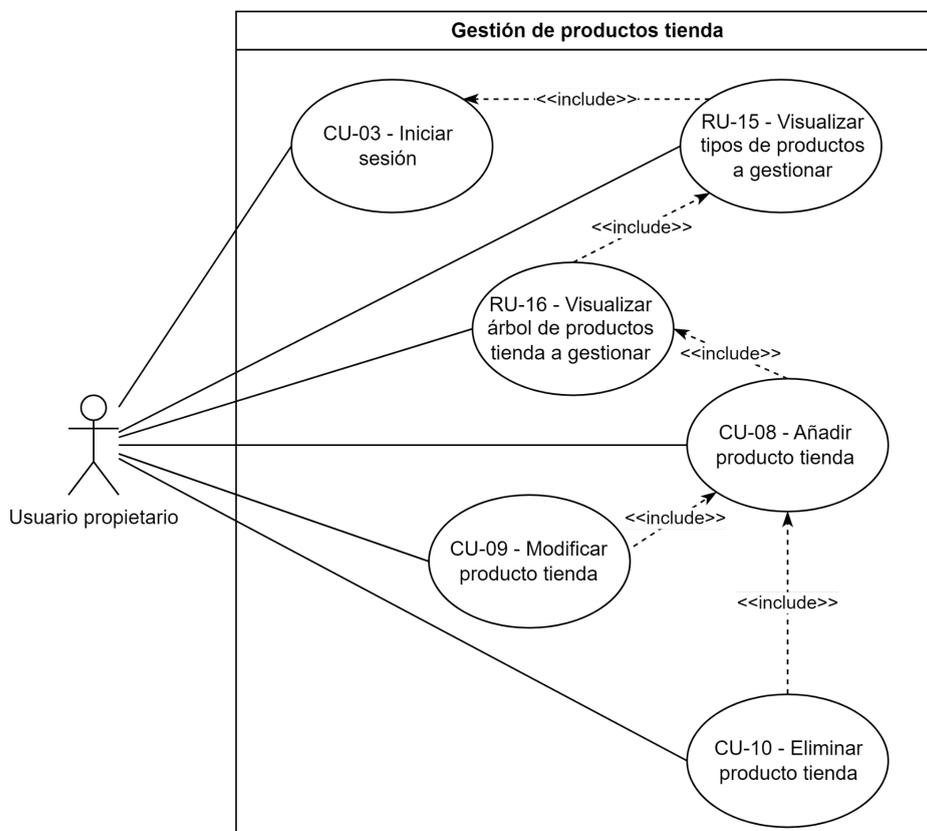


Figura 12. Diagrama de casos de uso “Gestión de productos tienda”.

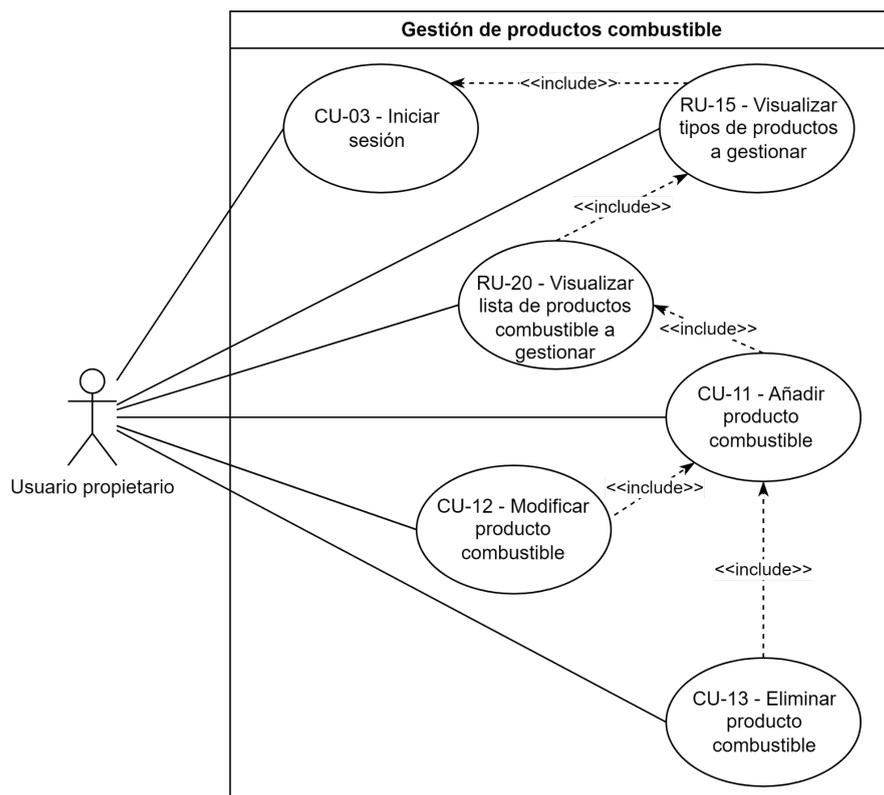


Figura 13. Diagrama de casos de uso "Gestión de productos combustible".

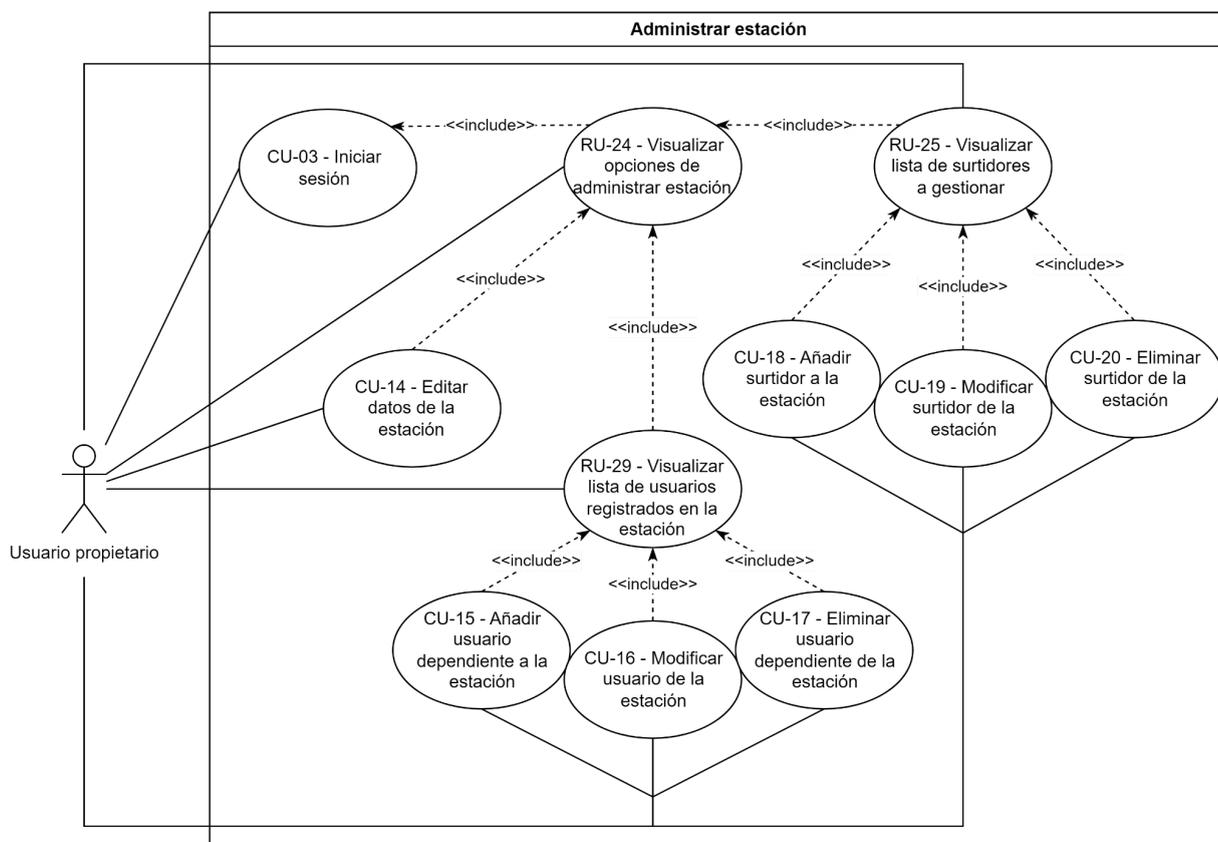


Figura 14. Diagrama de casos de uso "Administrar estación".

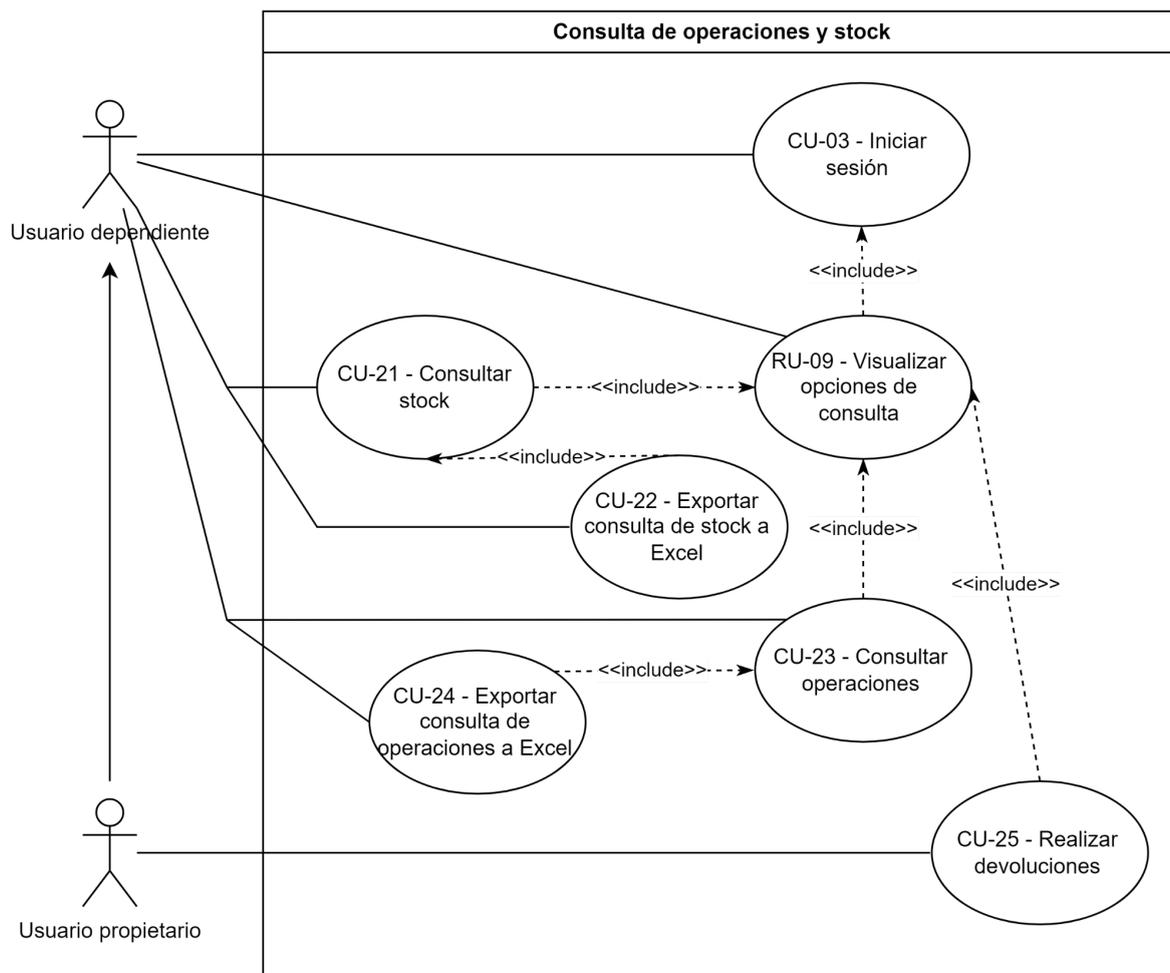


Figura 15. Diagrama de casos de uso “Consulta de operaciones y stock”.

3.4.2. Especificación de los casos de uso

Tabla 13. Caso de uso “Registrarse”.

CU-01 Registrarse	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-01
Actor	ACT-01 (usuario no registrado)
Descripción	El usuario que no esté previamente registrado en la aplicación podrá introducir sus datos y credenciales para darse de alta en la aplicación.
Precondiciones	El usuario no debe estar registrado en la aplicación.

CU-01 Registrarse	
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El usuario selecciona la opción “Crear nueva cuenta”. 3. El sistema muestra el formulario para darse de alta. 4. El usuario introduce sus datos (nombre, correo electrónico y contraseña). 5. El usuario selecciona en la opción “Darse de alta”. 6. El sistema muestra al usuario la pantalla principal de la aplicación, con los datos de productos y surtidores vacíos.
Postcondiciones	N/A
Excepciones	Si el usuario ya está dado de alta se mostrará un mensaje por pantalla indicando el problema al usuario.
Frecuencia	Baja: los usuarios mayoritariamente solo se darán de alta la primera vez que accedan a la aplicación.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 14. Caso de uso “Iniciar sesión”.

CU-02 Iniciar sesión	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-03
Actor	ACT-02 (<i>usuario dependiente</i>) ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que esté previamente registrado en la aplicación podrá introducir sus datos y credenciales para iniciar sesión en la aplicación y ver sus datos de estación.
Precondiciones	El usuario debe estar registrado en la aplicación.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El sistema muestra el formulario para el inicio de sesión. 3. El usuario introduce sus credenciales (correo electrónico y contraseña). 4. El usuario selecciona en la opción “Iniciar sesión”. 5. El sistema muestra al usuario la pantalla principal de la aplicación, con los datos de la estación y del usuario que se han creado previamente.

CU-02 Iniciar sesión	
Postcondiciones	N/A
Excepciones	Si el usuario no está registrado en la aplicación se mostrará un mensaje informativo al usuario en el momento que seleccione la opción “Iniciar sesión”. Si las credenciales introducidas en el formulario no son correctas se mostrará un mensaje informativo al usuario en el momento que seleccione la opción “Iniciar sesión”.
Frecuencia	Alta: los usuarios iniciarán sesión cada vez que quieran acceder a sus datos dentro de la aplicación.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 15. Caso de uso “Cerrar sesión”.

CU-03 Cerrar sesión	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-04
Actor	ACT-02 (<i>usuario dependiente</i>) ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que esté con la sesión iniciada podrá cerrar la sesión.
Precondiciones	El usuario debe estar registrado en la aplicación y con la sesión iniciada.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la aplicación con los datos de estación. 2. El sistema muestra la información de la estación. 3. El usuario selecciona la opción “Cerrar sesión”. 4. El sistema cierra la sesión del usuario y muestra la pantalla de inicio de sesión.
Postcondiciones	El usuario no debe poder visualizar la pantalla principal de la aplicación con la información de la estación a la que pertenece hasta que vuelva a iniciar sesión (CU-02).
Excepciones	N/A
Frecuencia	Media: los usuarios accederán a esta cada vez que finalicen un turno, pero solo una única vez por turno.

CU-03 Cerrar sesión	
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 16. Caso de uso “Recuperar contraseña”.

CU-04 Recuperar contraseña	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-05
Actor	ACT-02 (<i>usuario dependiente</i>) ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que esté previamente registrado en la aplicación podrá recibir un correo con un enlace para introducir la nueva contraseña que desee.
Precondiciones	El usuario debe estar registrado en la aplicación.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página de inicio de la aplicación. 2. El sistema muestra el formulario para el inicio de sesión. 3. El usuario introduce sus credenciales (email y contraseña). 4. El usuario selecciona en la opción “Iniciar sesión”. 5. El sistema indica al usuario que las credenciales no son correctas. 6. El usuario selecciona la opción “Olvidé mi contraseña”. 7. El sistema muestra la pantalla para introducir el correo electrónico asociado a su cuenta. 8. El usuario introduce su correo electrónico asociado a su cuenta. 9. El sistema muestra la pantalla de inicio de sesión y envía un enlace al correo electrónico indicado. 10. El usuario abre el enlace. 11. El sistema muestra la pantalla para introducir la nueva contraseña. 12. El usuario introduce la nueva contraseña que desee. 13. El sistema informa al usuario de que ya puede iniciar sesión con la nueva contraseña. 14. El usuario inicia sesión con la nueva contraseña (CU-02).
Postcondiciones	N/A

CU-04 Recuperar contraseña	
Excepciones	Si el usuario no está registrado en la aplicación se mostrará un mensaje informativo al usuario en el momento que seleccione la opción “Enviar correo”.
Frecuencia	Baja: los usuarios únicamente accederán a esta funcionalidad en el caso de que no recuerden su contraseña.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 17. Caso de uso “Realizar venta”.

CU-05 Realizar venta	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-08
Actor	ACT-02 (<i>usuario dependiente</i>) ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que haya iniciado sesión podrá realizar ventas seleccionando los productos, la cantidad o importe y el método de pago requerido.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y tiene que tener datos de alta productos tienda o productos combustible y surtidores habilitados.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la estación. 2. El sistema muestra los datos de surtidores y productos de tienda. 3. El usuario selecciona un producto de tienda (CU-06). 4. El sistema muestra el producto de tienda seleccionado en el ticket de compra y el precio total de la futura venta. 5. El usuario indica la cantidad del producto de tienda. 6. El sistema calcula de nuevo y muestra el precio total de la futura venta. 7. El usuario selecciona el surtidor (CU-07). 8. El sistema muestra el surtidor seleccionado en el ticket de compra junto con un desplegable para seleccionar el combustible. 9. El usuario selecciona el combustible. 10. El usuario indica la cantidad o el importe del combustible seleccionado. 11. El sistema calcula de nuevo y muestra el precio total

CU-05 Realizar venta	
	<p>de la futura venta.</p> <p>12. El usuario selecciona la opción “Pagar”.</p> <p>13. El sistema muestra las opciones de pago “Efectivo” y “Tarjeta”.</p> <p>14. El usuario selecciona el método de pago.</p> <p>15. El sistema muestra las opciones de confirmar venta.</p> <p>16. El usuario confirma la venta.</p> <p>17. El sistema muestra al usuario que la venta se ha realizado correctamente.</p>
Postcondiciones	La operación debe aparecer en la lista de operaciones realizadas.
Excepciones	<p>Si alguno de los productos no tiene stock suficiente para la realización de la venta se le indicará mediante un mensaje informativo al usuario.</p> <p>Si el usuario se ha equivocado a la hora de introducir algún producto de venta puede seleccionar la opción “Limpiar” para dejar el ticket de venta sin datos y volver a iniciar la venta.</p> <p>Si hubiese algún problema a la hora de guardar la información de la venta en el sistema se le comunicaría al usuario mediante un mensaje informativo.</p>
Frecuencia	Alta: es la funcionalidad que más se espera que realicen los usuarios en la aplicación.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 18. Caso de uso “Navegar por el árbol de productos tienda”.

CU-06 Navegar por el árbol de productos tienda	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-06
Actor	ACT-02 (<i>usuario dependiente</i>) ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que haya iniciado sesión podrá navegar por el árbol de productos, seleccionando las carpetas y subcarpetas para visualizar todos los productos habilitados.
Precondiciones	El usuario debe estar registrado en la aplicación, debe

CU-06 Navegar por el árbol de productos tienda	
Flujo normal	<p>haber iniciado sesión y tiene que tener datos de alta productos tienda en la estación a la que está asociado.</p> <ol style="list-style-type: none"> 1. El usuario accede a la página principal de la estación. 2. El sistema muestra los datos de surtidores y productos de tienda. 3. El usuario seleccionará la carpeta de productos que requiera. 4. El sistema muestra la lista de productos y subcarpetas habilitadas que contiene la carpeta seleccionada. 5. El usuario selecciona una subcarpeta dentro de la carpeta previamente seleccionada. 6. El sistema muestra la lista de productos y subcarpetas habilitadas que contiene la subcarpeta seleccionada. 7. El usuario selecciona la opción “Volver”. 8. El sistema muestra la lista de productos y subcarpetas habilitadas que contiene la carpeta a la que pertenece la subcarpeta en la que se encontraba el usuario. 9. El usuario selecciona la opción “Volver”. 10. El sistema muestra la lista de productos y subcarpetas habilitadas que contiene la raíz del árbol de productos de la estación. 11. El usuario selecciona un producto para iniciar una venta (CU-05).
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de obtener la información de los productos tienda y carpetas de productos guardados en el sistema para dicha estación se le comunicaría al usuario mediante un mensaje informativo.
Frecuencia	Alta: cada vez que se haga una venta de productos tienda será necesario navegar por el árbol de productos previamente.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 19. Caso de uso “Navegar por la lista de surtidores”.

CU-07 Navegar por la lista de surtidores	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-07
Actor	ACT-02 (usuario dependiente) ACT-03 (<i>usuario</i>)

CU-07 Navegar por la lista de surtidores	
Descripción	<p><i>propietario</i>)</p> <p>El <i>usuario propietario</i> que haya iniciado sesión podrá visualizar la lista de surtidores habilitados y navegar por dicha lista en el caso de que todos ellos no entren en la pantalla.</p>
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y tiene que tener datos de alta surtidores con combustibles habilitados en ellos en la estación a la que está asociado.
Flujo normal	<ol style="list-style-type: none"> 1. El usuario accede a la página principal de la estación. 2. El sistema muestra los datos de surtidores y productos de tienda. 3. El usuario selecciona la opción “>” para visualizar el siguiente surtidor que está a la derecha. 4. El sistema muestra el siguiente surtidor a la derecha y deja de mostrar el primer surtidor que había a la izquierda. 5. El usuario selecciona la opción “<” para visualizar el siguiente surtidor que está a la izquierda. 6. El sistema muestra el siguiente surtidor a la izquierda y deja de mostrar el primer surtidor que había a la derecha. 7. El usuario selecciona un surtidor para iniciar una venta (CU-05).
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de obtener la información de los surtidores guardados en el sistema para dicha estación se le comunicaría al usuario mediante un mensaje informativo.
Frecuencia	Alta: cada vez que se haga una venta de combustible será necesario navegar por la lista de surtidores previamente.
Importancia	Alta
Prioridad	Alta
Observaciones	En el que caso de que la lista de surtidores no excediera la pantalla, no sería necesario navegar por la lista de surtidores para visualizar y seleccionar todos ellos.

Tabla 20. Caso de uso “Añadir *producto tienda*”.

CU-08 Añadir <i>producto tienda</i>	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz

CU-08 Añadir <i>producto tienda</i>	
Requisitos asociados	RU-17
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá dar de alta productos tienda o carpetas de productos en la estación a la que esté asociado.
Precondiciones	El usuario debe estar registrado en la aplicación y debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Editar productos”. 3. El sistema muestra la lista de los tipos de productos de la estación. 4. El usuario selecciona la opción “Tienda”. 5. El sistema muestra la lista de <i>productos tienda</i> y carpetas, tanto habilitadas como no habilitadas, que están dados de alta en la estación. 6. El usuario selecciona la opción “Añadir producto”. 7. El sistema muestra el formulario para añadir los datos de <i>producto tienda</i>. 8. El usuario introduce los datos del producto. 9. El usuario selecciona la opción “Guardar”. 10. El sistema da de alta y guarda en base de datos la información del nuevo producto. 11. El sistema muestra la pantalla del árbol de <i>productos tienda</i> a gestionar con los nuevos datos.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de guardar los datos del nuevo producto se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla del árbol de productos tienda a gestionar.
Frecuencia	Media: cada vez que el <i>usuario propietario</i> requiera dar de alta un nuevo producto.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 21. Caso de uso “Modificar *producto tienda*”.

CU-09 Modificar <i>producto tienda</i>	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz

CU-09 Modificar <i>producto tienda</i>	
Requisitos asociados	RU-18
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá modificar productos tienda o carpetas de productos en la estación a la que esté asociado.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y haber dado de alta productos previamente en la estación.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Editar productos”. 3. El sistema muestra la lista de los tipos de productos de la estación. 4. El usuario selecciona la opción “Tienda”. 5. El sistema muestra la lista de <i>productos tienda</i> y carpetas, tanto habilitadas como no habilitadas, que están dados de alta en la estación. 6. El usuario selecciona el producto que quiere modificar dentro del árbol de productos. 7. El sistema muestra el formulario con los datos del producto para modificar. 8. El usuario introduce los datos del producto que desee modificar. 9. El usuario selecciona la opción “Guardar”. 10. El sistema modifica y guarda en base de datos la información actualizada del producto. 11. El sistema muestra la pantalla del árbol de <i>productos tienda</i> a gestionar con los nuevos datos.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de guardar los nuevos datos del producto se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla del árbol de productos tienda a gestionar.
Frecuencia	Media: cada vez que el <i>usuario propietario</i> requiera modificar un producto ya existente en la estación.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 22. Caso de uso “Eliminar *producto tienda*”.

CU-10 Eliminar <i>producto tienda</i>	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-19
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá eliminar productos tienda o carpetas de productos en la estación a la que esté asociado.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y haber dado de alta productos previamente en la estación.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Editar productos”. 3. El sistema muestra la lista de los tipos de productos de la estación. 4. El usuario selecciona la opción “Tienda”. 5. El sistema muestra la lista de <i>productos tienda</i> y carpetas, tanto habilitadas como no habilitadas, que están dados de alta en la estación. 6. El sistema muestra el árbol de productos a gestionar, tanto habilitados como no. 7. El usuario selecciona la carpeta donde se encuentra el producto o carpeta que desea eliminar. 8. El sistema muestra el árbol de productos dentro de la carpeta o subcarpeta seleccionada. 9. El usuario selecciona la opción “Eliminar” del producto que desea eliminar. 10. El sistema elimina y borra en base de datos la información del producto. 11. El sistema muestra la pantalla del árbol de <i>productos tienda</i> a gestionar con los datos actualizados.
Postcondiciones	N/A
Excepciones	<p>Si hubiese algún problema a la hora de eliminar los datos del producto se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla del árbol de productos tienda a gestionar.</p> <p>En el caso de que se requiera eliminar una carpeta de productos primero se debe eliminar todos los productos y subcarpetas que hay dentro de esta, de lo contrario este caso de uso no estará permitido.</p>
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> requiera eliminar un producto.

CU-10 Eliminar *producto tienda*

Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 23. Caso de uso “Añadir *producto combustible*”.**CU-11 Añadir *producto combustible***

Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-21
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá añadir <i>productos combustible</i> en la estación a la que está asociado.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Editar productos”. 3. El sistema muestra la lista de los tipos de productos de la estación. 4. El usuario selecciona la opción “Combustible”. 5. El sistema muestra la lista de <i>productos combustibles</i> que están dados de alta en la estación. 6. El usuario selecciona la opción “Añadir nuevo producto”. 7. El sistema muestra el formulario para introducir los datos del combustible. 8. El usuario introduce los datos del nuevo combustible. 9. El usuario selecciona la opción “Guardar”. 10. El sistema da de alta el nuevo producto en base de datos y muestra la pantalla con la lista de productos combustible a gestionar con el nuevo producto dado de alta.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de dar de alta el nuevo combustible se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de lista de productos combustible.
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> requiera añadir un nuevo combustible en la estación.

CU-11 Añadir <i>producto combustible</i>	
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 24. Caso de uso “Modificar *producto combustible*”.

CU-12 Modificar <i>producto combustible</i>	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-22
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá modificar productos combustible en la estación a la que está asociado.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y haber dado de alta algún producto combustible en la estación.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Editar productos”. 3. El sistema muestra la lista de los tipos de productos de la estación. 4. El usuario selecciona la opción “Combustible”. 5. El sistema muestra la lista de <i>productos combustibles</i> que están dados de alta en la estación. 6. El usuario selecciona el combustible que desea modificar. 7. El sistema muestra el formulario para introducir los datos del combustible con los datos del combustible seleccionado. 8. El usuario introduce los nuevos datos del combustible. 9. El usuario selecciona la opción “Guardar”. 10. El sistema actualiza los nuevos datos del producto en base de datos y muestra la pantalla con la lista de <i>productos combustible</i> a gestionar con los datos de productos actualizados.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de actualizar los nuevos datos de combustible se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de lista de <i>productos combustible</i> .

CU-12 Modificar *producto combustible*

Frecuencia	Alta: cada vez que el <i>usuario propietario</i> requiera modificar los datos de combustibles en la estación.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 25. Caso de uso “Eliminar *producto combustible*”.**CU-13 Eliminar *producto combustible***

Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-23
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá eliminar productos combustible en la estación a la que está asociado.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y haber dado de alta algún producto combustible en la estación.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Editar productos”. 3. El sistema muestra la lista de los tipos de productos de la estación. 4. El usuario selecciona la opción “Combustible”. 5. El sistema muestra la lista de <i>productos combustibles</i> que están dados de alta en la estación. 6. El usuario selecciona la opción “Eliminar” del combustible que desea eliminar de la estación. 7. El sistema elimina los datos del producto en base de datos y muestra la pantalla con la lista de <i>productos combustible</i> a gestionar con los datos de productos actualizados.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de actualizar los nuevos datos de combustible se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de lista de <i>productos combustible</i> .
Frecuencia	Alta: cada vez que el <i>usuario propietario</i> requiera modificar los datos de combustibles en la estación.

CU-13 Eliminar <i>producto combustible</i>	
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 26. Caso de uso “Editar datos de la estación”.

CU-14 Editar datos de la estación	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-33 y RU-34
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá modificar la información básica de la estación.
Precondiciones	El usuario debe estar registrado en la aplicación y debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Administrar estación”. 3. El sistema muestra las opciones de administración de la estación. 4. El usuario selecciona la opción “Editar datos de la estación”. 5. El sistema muestra el formulario con los datos actuales de la estación de nombre y dirección. 6. El usuario cambia los datos de la estación que desea modificar. 7. El usuario selecciona la opción “Guardar”. 8. El sistema actualiza los datos de la estación y muestra la pantalla de administración de la estación.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de mostrar o actualizar los datos de estación se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de administración de estación.
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> requiera modificar los datos de la estación.
Importancia	Media
Prioridad	Baja

CU-14 Editar datos de la estación

Observaciones	Estos datos de estación se visualizarán en los tickets de venta y devolución.
----------------------	---

Tabla 27. Caso de uso “Añadir *usuario dependiente* a la estación”.**CU-15 Añadir *usuario dependiente* a la estación**

Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-29 y RU-30
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá añadir nuevos <i>usuarios dependientes</i> a la estación.
Precondiciones	El <i>usuario propietario</i> debe estar registrado en la aplicación y debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Administrar estación”. 3. El sistema muestra las opciones de administración de la estación. 4. El usuario selecciona la opción “Administrar usuarios”. 5. El sistema muestra la lista de todos los usuarios de la estación. 6. El usuario selecciona la opción “Añadir nuevo usuario”. 7. El sistema muestra el formulario para introducir los datos del nuevo usuario. 8. El usuario introduce los datos del <i>usuario dependiente</i> que desea dar de alta en la estación. 9. El usuario selecciona la opción “Guardar”. 10. El sistema da de alta el nuevo usuario en la estación y muestra la pantalla de lista de usuarios con los nuevos datos actualizados.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de mostrar o dar de alta los datos del nuevo usuario se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de administración de estación.
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> quiera dar de alta nuevos usuarios en la estación.
Importancia	Media

CU-15 Añadir <i>usuario dependiente</i> a la estación	
Prioridad	Media
Observaciones	Todos los usuarios que se den de alta mediante este caso de uso serán <i>usuarios dependientes</i> .

Tabla 28. Caso de uso “Modificar usuario de la estación”.

CU-16 Modificar usuario de la estación	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-29 y RU-31
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá modificar la información básica de los <i>usuarios dependientes</i> (excepto contraseña) de la estación a la que pertenece y modificar todos los datos de su usuario.
Precondiciones	El <i>usuario propietario</i> debe estar registrado en la aplicación y debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Administrar estación”. 3. El sistema muestra las opciones de administración de la estación. 4. El usuario selecciona la opción “Administrar usuarios”. 5. El sistema muestra la lista de todos los usuarios de la estación. 6. El usuario selecciona el usuario del que desea modificar su información. 7. El sistema muestra el formulario de usuario con los datos del usuario seleccionado. 8. El usuario modifica los datos que desea del usuario seleccionado. 9. El usuario selecciona la opción “Guardar”. 10. El sistema actualiza la información del usuario en la estación y muestra la pantalla de lista de usuarios con los nuevos datos actualizados.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de mostrar o actualizar los datos del usuario se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de administración de estación.

CU-16 Modificar usuario de la estación

Frecuencia	Baja: cada vez que el <i>usuario propietario</i> quiera modificar la información de los usuarios.
Importancia	Baja
Prioridad	Baja
Observaciones	Todos los usuarios que se den de alta mediante este caso de uso serán <i>usuarios dependientes</i> .

Tabla 29. Caso de uso “Eliminar *usuario dependiente* de la estación”.

CU-17 Eliminar *usuario dependiente* de la estación

Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-29 y RU-30
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá eliminar <i>usuarios dependientes</i> de la estación.
Precondiciones	El <i>usuario propietario</i> debe estar registrado en la aplicación y debe haber iniciado sesión.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Administrar estación”. 3. El sistema muestra las opciones de administración de la estación. 4. El usuario selecciona la opción “Administrar usuarios”. 5. El sistema muestra la lista de todos los usuarios de la estación. 6. El usuario selecciona la opción “Eliminar” del usuario que desea eliminar de la estación. 7. El sistema elimina los datos del usuario de la estación y muestra la pantalla de lista de usuarios con los nuevos datos actualizados.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de eliminar el <i>usuario dependiente</i> , se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de administración de estación.
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> quiera eliminar un <i>usuario dependiente</i> de la estación.

CU-17 Eliminar *usuario dependiente* de la estación

Importancia	Media
Prioridad	Baja
Observaciones	El <i>usuario dependiente</i> que se elimine de la estación ya no podrá iniciar sesión en la aplicación.

Tabla 30. Caso de uso “Añadir surtidor a la estación”.**CU-18 Añadir surtidor a la estación**

Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-25 y RU-26
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá añadir surtidores a la estación.
Precondiciones	El <i>usuario propietario</i> debe estar registrado en la aplicación, debe haber iniciado sesión y debe haber dado de alta <i>productos combustible</i> en la estación.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Administrar estación”. 3. El sistema muestra las opciones de administración de la estación. 4. El usuario selecciona la opción “Administrar surtidores”. 5. El sistema muestra la lista de todos los surtidores dados de alta en la estación, estén habilitados o no. 6. El usuario selecciona la opción “Añadir nuevo surtidor”. 7. El sistema muestra el formulario para añadir mangueras en el surtidor. 8. El usuario añade la información de mangueras en el surtidor seleccionando los combustibles deseados. 9. El usuario selecciona la opción “Guardar”. 10. El sistema guarda los datos del nuevo surtidor en la estación y muestra la pantalla de lista de surtidores a gestionar con los nuevos datos actualizados y actualiza la lista principal de surtidores.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de mostrar o dar de alta los datos del nuevo surtidor se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla

CU-18 Añadir surtidor a la estación

	de administración de estación.
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> quiera añadir surtidores nuevos a la estación.
Importancia	Alta
Prioridad	Alta
Observaciones	N/A

Tabla 31. Caso de uso “Modificar surtidor a la estación”.**CU-19 Modificar surtidor a la estación**

Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-25 y RU-27
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá modificar <i>usuarios dependientes</i> de la estación.
Precondiciones	El <i>usuario propietario</i> debe estar registrado en la aplicación, debe haber iniciado sesión y debe haber dado de alta previamente algún surtidor.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Administrar estación”. 3. El sistema muestra las opciones de administración de la estación. 4. El usuario selecciona la opción “Administrar surtidores”. 5. El sistema muestra la lista de todos los surtidores dados de alta en la estación, estén habilitados o no. 6. El usuario selecciona el surtidor que desea modificar. 7. El sistema muestra el formulario para añadir mangueras en el surtidor, con la información de combustibles del surtidor seleccionado. 8. El usuario modifica la información de mangueras en el surtidor seleccionando los combustibles deseados. 9. El usuario selecciona la opción “Guardar”. 10. El sistema actualiza los datos del surtidor en la estación y muestra la pantalla de lista de surtidores a gestionar con los nuevos datos actualizados y actualiza la lista principal de surtidores.
Postcondiciones	N/A

CU-19 Modificar surtidor a la estación	
Excepciones	Si hubiese algún problema a la hora de mostrar o actualizar los datos del surtidor se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de administración de estación.
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> quiera modificar los datos de los surtidores.
Importancia	Media
Prioridad	Media
Observaciones	N/A

Tabla 32. Caso de uso “Eliminar surtidor a la estación”.

CU-20 Eliminar surtidor a la estación	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-25 y RU-28
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá modificar <i>usuarios dependientes</i> de la estación.
Precondiciones	El <i>usuario propietario</i> debe estar registrado en la aplicación, debe haber iniciado sesión y debe haber dado de alta previamente algún surtidor.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Administrar estación”. 3. El sistema muestra las opciones de administración de la estación. 4. El usuario selecciona la opción “Administrar surtidores”. 5. El sistema muestra la lista de todos los surtidores dados de alta en la estación, estén habilitados o no. 6. El usuario selecciona la opción “Eliminar” del surtidor que desea eliminar. 7. El sistema actualiza los datos del surtidor en la estación y muestra la pantalla de lista de surtidores a gestionar con los nuevos datos actualizados y actualiza la lista principal de surtidores.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de mostrar o dar de

CU-20 Eliminar surtidor a la estación	
	alta los datos del nuevo usuario se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de administración de estación.
	En el caso de que el surtidor tenga mangueras de combustibles asociadas la opción “Eliminar” no estará disponible para dicho surtidor.
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> quiera eliminar un surtidor de la estación.
Importancia	Media
Prioridad	Media
Observaciones	N/A.

Tabla 33. Caso de uso “Consultar y actualizar stock”.

CU-21 Consultar y actualizar stock	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-12 y RU-13
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que haya iniciado sesión podrá modificar y consultar el stock de productos en la estación.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y debe haber dado de alta previamente algún producto.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Consulta de operaciones y stock”. 3. El sistema muestra las opciones de consulta de operaciones y stock del usuario. 4. El usuario selecciona la opción “Consulta de stock”. 5. El sistema muestra la lista de todos los productos dados de alta en la estación, estén habilitados o no, con su información básica y su stock en la estación. 6. El usuario selecciona la opción “Actualizar” sobre el producto del que desea actualizar su stock. 7. El sistema muestra los botones de “Cancelar” y “Guardar” y permite introducir el nuevo stock para el producto. 8. El usuario introduce el nuevo stock para el producto seleccionado.

CU-21 Consultar y actualizar stock	
	<p>9. El usuario selecciona la opción “Guardar”.</p> <p>10. El sistema actualiza el stock del producto y muestra la lista de productos y su stock actualizada.</p>
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de mostrar o actualizar el stock de productos se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de “Consulta de operaciones y stock”.
Frecuencia	Media: cada vez que el usuario quiera consultar y actualizar el stock.
Importancia	Media
Prioridad	Media
Observaciones	N/A.

Tabla 34. Caso de uso “Exportar la consulta de stock a Excel”.

CU-22 Exportar la consulta de stock a Excel	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-12 y RU-14
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que haya iniciado sesión podrá modificar consultar el stock de productos en la estación.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y debe haber dado de alta previamente algún producto.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Consulta de operaciones y stock”. 3. El sistema muestra las opciones de consulta de operaciones y stock del usuario. 4. El usuario selecciona la opción “Consulta de stock”. 5. El sistema muestra la lista de todos los productos dados de alta en la estación, estén habilitados o no, con su información básica y su stock en la estación. 6. El usuario selecciona la opción “Exportar a Excel”. 7. El sistema exporta los datos de productos y su stock a Excel y descarga el archivo en el dispositivo.
Postcondiciones	N/A

CU-22 Exportar la consulta de stock a Excel	
Excepciones	Si hubiese algún problema a la hora de exportar la información de productos a Excel se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de “Consulta de operaciones y stock”.
Frecuencia	Baja: cada vez que el usuario quiera exportar la consulta de stock a Excel.
Importancia	Baja
Prioridad	Baja
Observaciones	N/A.

Tabla 35. Caso de uso “Consultar operaciones”.

CU-23 Consultar operaciones	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-10
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que haya iniciado sesión podrá consultar las operaciones realizadas en la estación y su información básica.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y debe haber realizado alguna operación previamente en la estación.
Flujo normal	El sistema muestra la pantalla principal con los datos de la estación. El usuario selecciona la opción “Consulta de operaciones y stock”. El sistema muestra las opciones de consulta de operaciones y stock del usuario. El usuario selecciona la opción “Consulta de operaciones”. El sistema muestra la lista de todas las operaciones realizadas en la estación con su información básica.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de mostrar los datos de operaciones se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de “Consulta de operaciones y stock”.
Frecuencia	Media: cada vez que el usuario quiera consultar las operaciones realizadas en la estación.

CU-23 Consultar operaciones	
Importancia	Media
Prioridad	Media
Observaciones	N/A.

Tabla 36. Caso de uso “Exportar la consulta de operaciones a Excel”.

CU-24 Exportar la consulta de operaciones a Excel	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-12 y RU-14
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El usuario que haya iniciado sesión podrá modificar consultar el stock de productos en la estación.
Precondiciones	El usuario debe estar registrado en la aplicación, debe haber iniciado sesión y debe haber realizado previamente alguna operación en la estación.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Consulta de operaciones y stock”. 3. El sistema muestra las opciones de consulta de operaciones y stock del usuario. 4. El usuario selecciona la opción “Consulta de operaciones”. 5. El sistema muestra la lista de todas las operaciones realizadas en la estación con su información básica. 6. El usuario selecciona la opción “Exportar a Excel”. 7. El sistema exporta los datos de operaciones a Excel y descarga el archivo en el dispositivo.
Postcondiciones	N/A
Excepciones	Si hubiese algún problema a la hora de mostrar o dar de alta los datos del nuevo usuario se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de administración de estación.
Frecuencia	Baja: cada vez que el usuario quiera exportar la consulta de operaciones a Excel.
Importancia	Baja
Prioridad	Baja
Observaciones	N/A.

Tabla 37. Caso de uso “Realizar devoluciones”.

CU-25 Realizar devoluciones	
Versión	1.0
Autor	Carlos Bernaldo de Quirós Sanz
Requisitos asociados	RU-35, RU-36 y RU-37
Actor	ACT-03 (<i>usuario propietario</i>)
Descripción	El <i>usuario propietario</i> que haya iniciado sesión podrá realizar la devolución de operaciones realizadas en la estación que no hayan sido ya devueltas.
Precondiciones	El <i>usuario propietario</i> debe estar registrado en la aplicación, debe haber iniciado sesión y debe haber realizado previamente alguna venta en la estación.
Flujo normal	<ol style="list-style-type: none"> 1. El sistema muestra la pantalla principal con los datos de la estación. 2. El usuario selecciona la opción “Consulta de operaciones y stock”. 3. El sistema muestra las opciones de consulta de operaciones y stock del <i>usuario propietario</i>. 4. El usuario selecciona la opción “Devoluciones”. 5. El sistema muestra la lista de operaciones que no han sido devueltas. 6. El usuario selecciona la opción “Devolver” sobre la operación que desea devolver. 7. El sistema muestra las opciones “Cancelar” y “Confirmar devolución”. 8. El usuario selecciona la opción “Confirmar devolución”. 9. El sistema devuelve la operación y muestra la lista de operaciones no devueltas con los datos actualizados.
Postcondiciones	La operación devuelta ya no aparecerá en la lista de operaciones a devolver y no podrá ser devuelta.
Excepciones	Si hubiese algún problema a la hora de mostrar o de realizar la devolución de la operación seleccionada se le indicaría al usuario mediante un mensaje informativo y regresaría a la pantalla de “Devoluciones”, mostrando la información sin alterar de operaciones que no se han devuelto.
Frecuencia	Baja: cada vez que el <i>usuario propietario</i> quiera realizar la devolución de alguna de las operaciones realizadas.
Importancia	Alta
Prioridad	Alta

CU-25 Realizar devoluciones

Observaciones

Para los *usuarios dependientes* está opción no estará visible dentro del submenú “Consulta de operaciones y stock”.

3.5. Requisitos de información

En este apartado detallaremos los requisitos de información, que son aquellas especificaciones de datos que son necesarias para que la aplicación pueda funcionar correctamente. En nuestro caso estos son los requisitos de información:

RI-01: El sistema almacenará los datos de los usuarios (nombre y correo electrónico) cuando este se registre en el sistema.

RI-02: Las credenciales de usuario se almacenarán a través de Firebase Authentication. No se guardará la contraseña directamente en nuestra base de datos.

RI-03: En el registro de los usuarios propietarios se crearán los datos básicos de estación relacionados con el usuario, entre los que se incluye el nombre de la estación. La dirección de la estación se guardará como una cadena de texto vacía y no tendrá ningún producto ni surtidor por defecto.

RI-04: El sistema guardará los datos de los usuarios dependientes asociando dicho usuario a la estación ya existente a la que pertenece el usuario propietario que lo da de alta.

RI-05: El sistema guardará un registro de las operaciones de venta y devolución realizadas para cada una de las estaciones.

RI-06: El sistema almacenará los datos de los productos, tanto de tienda como de combustibles asociados a la estación. Para los productos de tienda incluirá la carpeta o subcarpeta en la que el producto se debe mostrar.

RI-07: El sistema almacenará los datos de los surtidores asociados a la estación, incluyendo los combustibles de los que dispone.

3.5.1. Diagrama entidad-relación

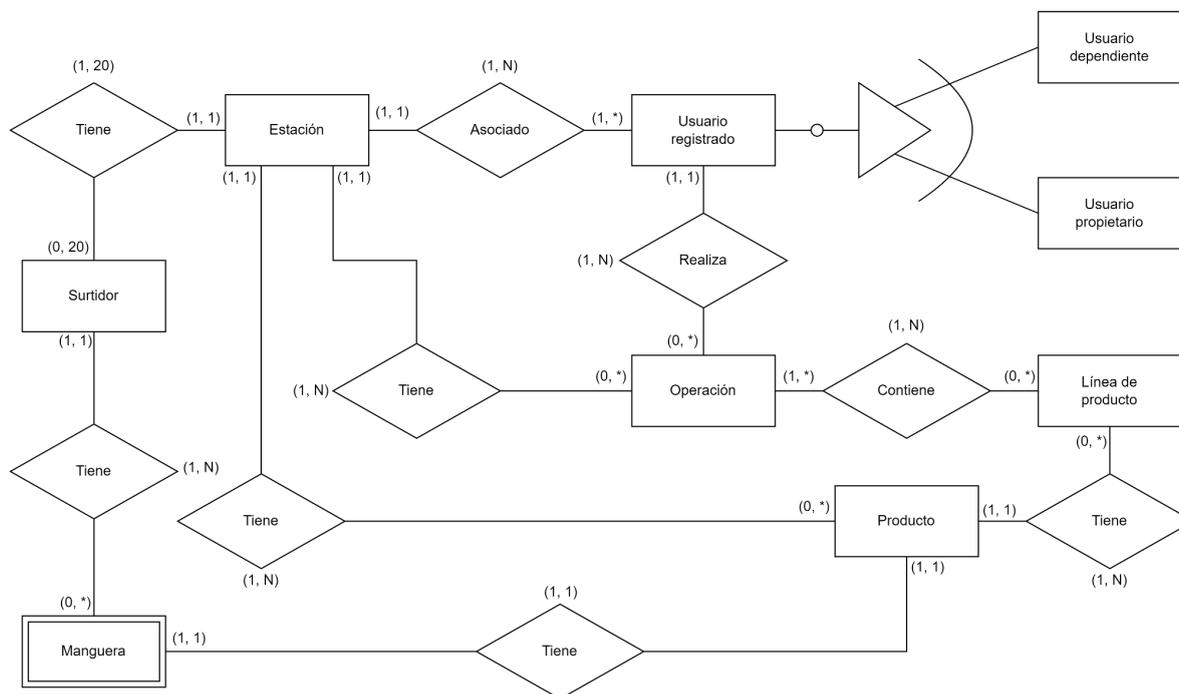


Figura 16. Diagrama de Entidad-Relación.

El diagrama de entidad-relación nos otorga una imagen conceptual de la relación existente en la aplicación entre los distintos datos que componen la misma.

Como se puede apreciar en la Figura 16, el elemento central tanto de este diagrama como de la aplicación es la entidad “Estación”, que es a partir de la cual se generan el resto de datos de la misma. Todas las entidades, excepto la entidad “Manguera”, que es la única entidad débil del diagrama, y la entidad “Línea de producto”, que es una entidad que surge de la relación entre “Operación” y “Producto”, se relacionan directamente con una “Estación”.

Aunque en el diagrama se distinguen dos entidades para los dos tipos de usuarios, en la práctica usaremos la misma tabla para almacenar ambos y serán discriminados mediante un campo que indique el tipo de usuario. En el apartado siguiente se explica esto en detalle, así como la implementación real de la base de datos.

3.5.2. Diccionario de datos

En este apartado se detalla la información de cada una de las entidades que componen la base de datos relacional de la aplicación.

En la práctica únicamente utilizaremos una base de datos no relacional para almacenar y manejar la información de manera más eficiente, que nos permita acceder de forma más rápida a la información.

Esta base de datos no relacional que utilizaremos será Firestore Database [9] que proporciona Firebase. Esta base de datos se basa en una serie de colecciones en las que se almacenan una serie de información relacionada, con una estructura de datos *JSON*, como un dato único en la colección.

Aunque se utilice dicha base de datos no relacional, cada estructura de datos *JSON* que se genera y se guarda en las colecciones de Firestore Database se basa en la estructura lógica de la base de datos relacional lógica en la que aquí se detalla cada una de sus tablas.

Tabla 38. Diccionario de datos de “Users”.

Atributo	Tipo	Único	Nulo	Clave primaria	Clave foránea
<u>id</u>	ObjectID ²	Sí	No	<u>Sí</u>	No
name	String	No	No	No	No
email	String	Sí	No	No	No
role	Number	No	No	No	No
shop_id	ObjectID	No	No	No	<i>Sí</i>

Tabla 39. Diccionario de datos de “Shops”.

Atributo	Tipo	Único	Nulo	Clave primaria	Clave foránea
<u>id</u>	ObjectID	Sí	No	<u>Sí</u>	No
name	String	No	No	No	No
address	String	No	Sí	No	No
owner_id	ObjectID	Sí	No	No	<i>Sí</i>

Tabla 40. Diccionario de datos de “Products”.

Atributo	Tipo	Único	Nulo	Clave primaria	Clave foránea
<u>id</u>	ObjectID	Sí	No	<u>Sí</u>	No
name	String	No	No	No	No
price	Number	No	No	No	No
stock	Number	No	No	No	No
active	Boolean	No	No	No	No
is_folder	Boolean	No	No	No	No
is_fuel	Boolean	No	No	No	No
folder_id	ObjectID	No	No	No	<i>Sí</i>
shop_id	ObjectID	No	No	No	<i>Sí</i>

² Un ObjectID es un ID alfanumérico interno generado automáticamente por Firestore. En bases de datos relacionales, este tipo es equivalente a un UUID o, en su defecto, a un String.

Tabla 41. Diccionario de datos de “Pumps”.

Atributo	Tipo	Único	Nulo	Clave primaria	Clave foránea
<u>id</u>	String	Sí	No	<u>Sí</u>	No
pump_id	Number	No	No	No	No
pipes_count	Number	No	No	No	No
shop_id	ObjectID	No	No	No	Sí

Tabla 42. Diccionario de datos de “Pipes”.

Atributo	Tipo	Único	Nulo	Clave primaria	Clave foránea
<u>id</u>	ObjectID	Sí	No	<u>Sí</u>	No
pipe_id	Number	No	No	No	No
<i>fuel_id</i>	ObjectID	No	No	No	Sí
<i>pump_id</i>	ObjectID	No	No	No	Sí

Tabla 43. Diccionario de datos de “Operations”.

Atributo	Tipo	Único	Nulo	Clave primaria	Clave foránea
<u>id</u>	ObjectID	Sí	No	<u>Sí</u>	No
<i>shop_id</i>	ObjectID	No	No	No	Sí
<i>user_id</i>	ObjectID	No	No	No	Sí
date	Date	No	No	No	No
total_amount	Number	No	No	No	No
ticket	String	No	No	No	No
payment_method	Number	No	No	No	No
is_refund	Boolean	No	No	No	No

Tabla 44. Diccionario de datos de “Operation_products”.

Atributo	Tipo	Único	Nulo	Clave primaria	Clave foránea
<u>id</u>	ObjectID	Sí	No	<u>Sí</u>	No
<i>product_id</i>	ObjectID	No	No	No	Sí
quantity	Number	No	No	No	No
unit_price	Number	No	No	No	No
total_amount	Number	No	No	No	No
<i>operation_id</i>	ObjectID	No	No	No	Sí

3.6. Requisitos no funcionales

Los requisitos no funcionales hacen referencia a aquellas características de la aplicación que no afectan directamente a su desarrollo o comportamiento, es decir, que no afectan a su funcionalidad. Para este proyecto, los requisitos no funcionales son los siguientes:

RNF-01: La aplicación deberá ser fácil de entender y de usar.

RNF-02: La aplicación deberá cumplir con los principios de *responsive design* para poder ser utilizada en cualquier dispositivo en un navegador con acceso a Internet.

RNF-03: La aplicación deberá informar al usuario de errores de forma clara y sin datos innecesarios.

RNF-04: La aplicación tiene que permitir ser ejecutada en los navegadores más utilizados del mercado.

RNF-05: Las credenciales se almacenarán en Firebase Authentication.

RNF-06: La comunicación con Firebase se realizará de forma segura.

RNF-07: El sistema deberá ser escalable y soportar la concurrencia de distintos usuarios.

RNF-08: El sistema será desplegado en Cloudflare Pages para facilitar su escalabilidad.

RNF-09: El sistema almacenará la información en Firestore Database de forma que sea capaz de obtener y guardar dicha información de forma rápida, eficiente y segura.

RNF-10: El sistema deberá ser capaz de tener una disponibilidad completa.

4. Diseño

En este apartado vamos a definir el diseño que se ha seguido para la realización de la aplicación, tanto en su arquitectura, centrándonos en la parte lógica, debido a que al ser una aplicación *serverless* no controlamos la arquitectura física de la solución, como en el proceso de diseño de la interfaz gráfica.

4.1. Arquitectura lógica

La arquitectura lógica de la aplicación se divide principalmente en dos partes: el cliente y el servidor.

Al ser esta una aplicación web, el cliente es el navegador web, que es la parte lógica que demanda los ficheros de ejecución estáticos al servidor. Se encarga de interpretar y ejecutar el código definido en estos ficheros, además de realizar una serie de consultas y operaciones al servidor, que viene definido en el propio código que provee el servidor.

El servidor, está definido por los servicios en la nube de Cloudflare y Firebase. Dentro de Cloudflare podemos encontrar dos productos:

- Cloudflare Pages, que es donde está almacenado todo el código estático que se envía al cliente cuando llega una petición para que este lo interprete y ejecute. Únicamente se encarga de proveer al navegador (cliente) de estos ficheros que necesita para poder ejecutar la aplicación.
- Cloudflare Workers, que es donde se encuentra el API de la aplicación para realizar la comunicación específica con los servicios de base de datos y autenticación de Firebase que no pueden ejecutarse en la parte back por limitaciones técnicas y de seguridad.

En cuanto a Firebase, es la parte de back donde almacenamos toda la información y donde se configura toda la restricción de usuarios para garantizar la seguridad de los datos. Dentro de Firebase también se divide en dos partes:

- Firestore Database, que es la base de datos no relacional donde se almacena y se relaciona toda la información de la aplicación, excepto la parte de autenticación.
- Firebase Authentication es tanto la base de datos segura de autenticación como el servicio que otorga la gestión segura de usuarios y sesiones a la aplicación.

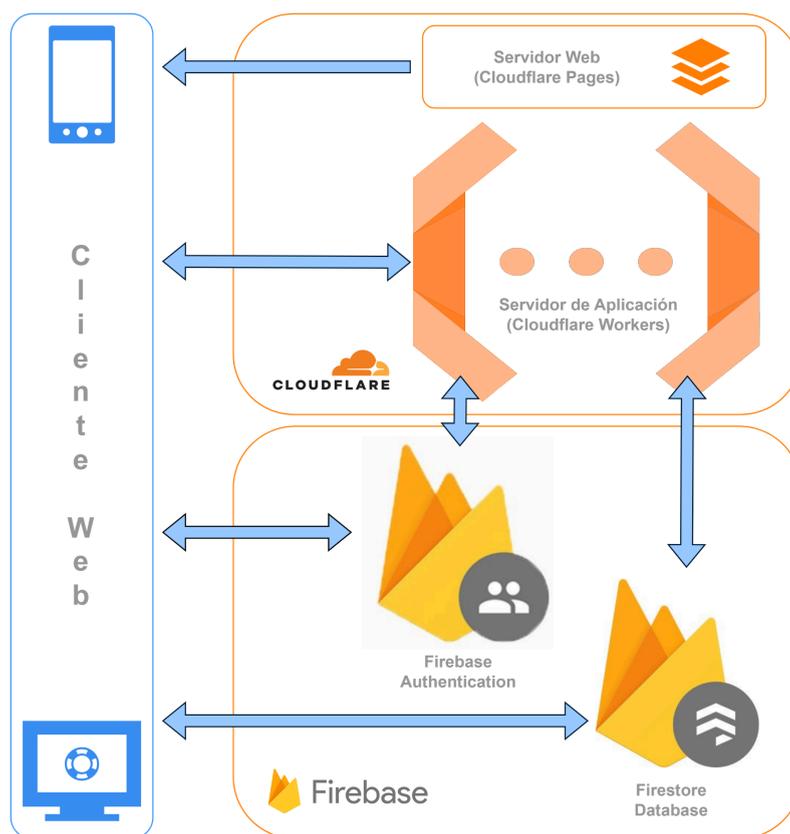


Figura 17. Diagrama de arquitectura lógica.

En cuanto al patrón de diseño de la aplicación, al estar desarrollada la solución usando el *framework* de AngularJS, se ha utilizado MVVM (*Model - View - ViewModel*). Este patrón de arquitectura divide la aplicación en tres partes bien diferenciadas, como indica su nombre: Modelo, Vista y Vista-Modelo.

- **Modelo.** El modelo representa la información almacenada en la aplicación y su lógica de negocio. Se encarga de recuperar y almacenar datos hacia el *backend* de la solución, bien sea la base de datos o bien las *functions* del servidor de aplicación.
- **Vista.** Representa los componentes de la interfaz de usuario como *HTML* y *SCSS*. En el patrón *MVVC*, la *Vista* es responsable de mostrar los datos que se reciben de la *Vista-Modelo*. Esta vista se encarga esencialmente de transformar los *Modelos* en la interfaz de usuario.
- **Vista-Modelo.** Sirve como intermediario entre la vista y el modelo. Contiene la lógica de presentación de la aplicación, es decir, todas las funciones y comandos, para poder vincular la *Vista* al *Modelo* y manejar los eventos que surjan en la *Vista*, así como.

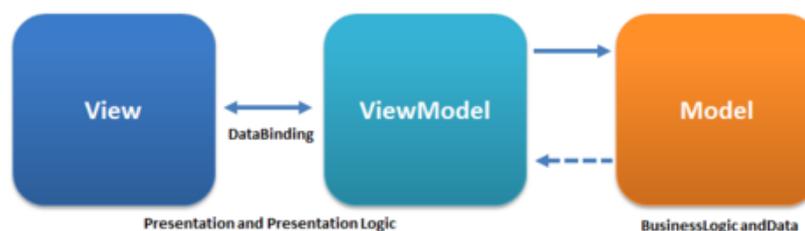


Figura 18. Patrón de arquitectura MVVM.

La diferencia entre MVVM y el MVC clásico radica en la comunicación entre la *Vista* y el *Modelo*, ya que en el MVVM la *Vista-Modelo* se encarga de crear el *DataBinding*, es decir, de relacionar directamente la *Vista* con los datos del *Modelo* [7]. Es debido a esto que si el usuario cambia algún dato en la *Vista*, este se cambia al momento en el *Modelo* y viceversa, si hay algún cambio en el *Modelo* de datos, la *Vista* se actualiza con el nuevo dato existente. A este paradigma se le conoce como programación *reactiva*, del cual reciben su nombre frameworks como React. En el patrón de arquitectura MVC sería el *Controlador* el que debería actualizar los datos en el *Modelo* una vez que haya un evento en la vista que se maneje desde el propio *Controlador*.

4.2. Arquitectura física

Esta solución es una aplicación completamente *serverless*, es decir, sin servidor gestionado por el desarrollador. Evidentemente, sí que hay servidores por detrás que ofrecen los recursos necesarios para ejecutar las acciones requeridas por el usuario y el cliente, pero el aprovisionamiento y administración de los mismos lo realiza el proveedor del *cloud*, que en nuestro caso serán tanto Cloudflare como Firebase.

Teniendo en cuenta esto, la arquitectura física de la aplicación se muestra en la Figura 19, donde los servidores físicos tanto de Firebase como Cloudflare actúan como una caja negra. Estos proveedores se encargan de administrar los servidores (sistema operativo, software necesario, memoria y demás recursos), replicar la información, balancear la carga de peticiones y añadir o quitar recursos en función de la cantidad de peticiones que lleguen a los distintos servicios con los que cuenta la aplicación, siempre con una alta disponibilidad garantizada. Esto permite centrarnos plenamente en el desarrollo de la aplicación.

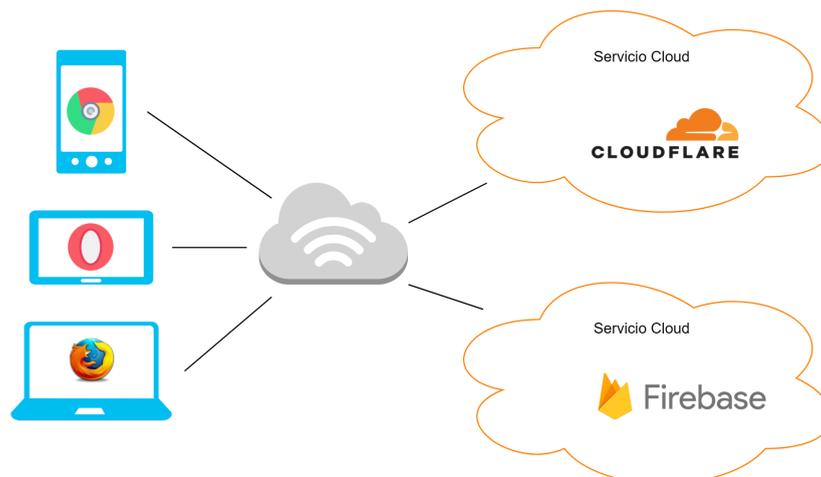


Figura 19. Diagrama de arquitectura física.

Las ventajas principales de adoptar la arquitectura serverless son las siguientes:

- **Eliminación de la gestión de servidores.** No es necesario ocuparnos de la provisión y administración de servidores, ni de mantener los sistemas operativos o instalar y actualizar su software. Todas estas tareas relacionadas con la infraestructura son responsabilidad de los proveedores del *cloud*.
- **Escalabilidad automática.** Con la arquitectura *serverless*, nuestra aplicación puede escalar automáticamente según la demanda. Esto significa que se ajusta dinámicamente a las necesidades de los usuarios, manteniendo un equilibrio en el uso de recursos. Este punto está íntimamente ligado con el siguiente.
- **Ahorro de costes.** Al igual que con otros servicios en la nube, en la arquitectura *serverless* solo pagamos por lo que usamos. En nuestro caso, estamos utilizando los servicios gratuitos de ambas plataformas *cloud*, que ofrecen una disponibilidad de recursos limitada, pero en el caso de que optemos por cambiar a un modelo de pago, por ejemplo, en Cloudflare Workers, únicamente pagaremos por la cantidad de peticiones que se realicen a las distintas *functions* allí desplegadas. Esto conduce a un ahorro significativo de costes, ya que los recursos se utilizan de manera óptima. Es necesario mencionar que, ante aplicaciones con un gran número de usuarios o peticiones, el coste de plataformas *serverless* suele ser mayor que el de gestionar nuestra propia infraestructura [2]. Sin embargo, para nuestro caso de uso es ideal.
- **Alta disponibilidad y tolerancia a fallos.** Los servicios *serverless* ofrecen alta disponibilidad y tolerancia a fallos de forma transparente para el desarrollador, sin necesidad de implementar procesos adicionales para replicar datos en diferentes zonas de disponibilidad. Esto garantiza que nuestra aplicación esté siempre (o casi siempre) disponible y pueda tolerar fallos de manera efectiva, propiedad conocida como resiliencia.

4.3. Diagramas de secuencia

En este apartado se detallan los diagramas de secuencia de la aplicación, donde se puede observar como es la comunicación entre los distintos servicios que componen la aplicación en cada uno de los flujos de esta.

Se ha optado por la realización de los diagramas de secuencia de los flujos más representativos, ya que el resto de flujos tendrán una secuencia de eventos y comunicación similar a estos.

4.3.1. Diagrama de secuencia - Iniciar sesión y Venta

Debemos indicar que la secuencia de eventos para la completa realización de devoluciones es prácticamente igual a la de la venta. Lo único que se modifica es la comunicación entre el usuario y el cliente web (interfaz).

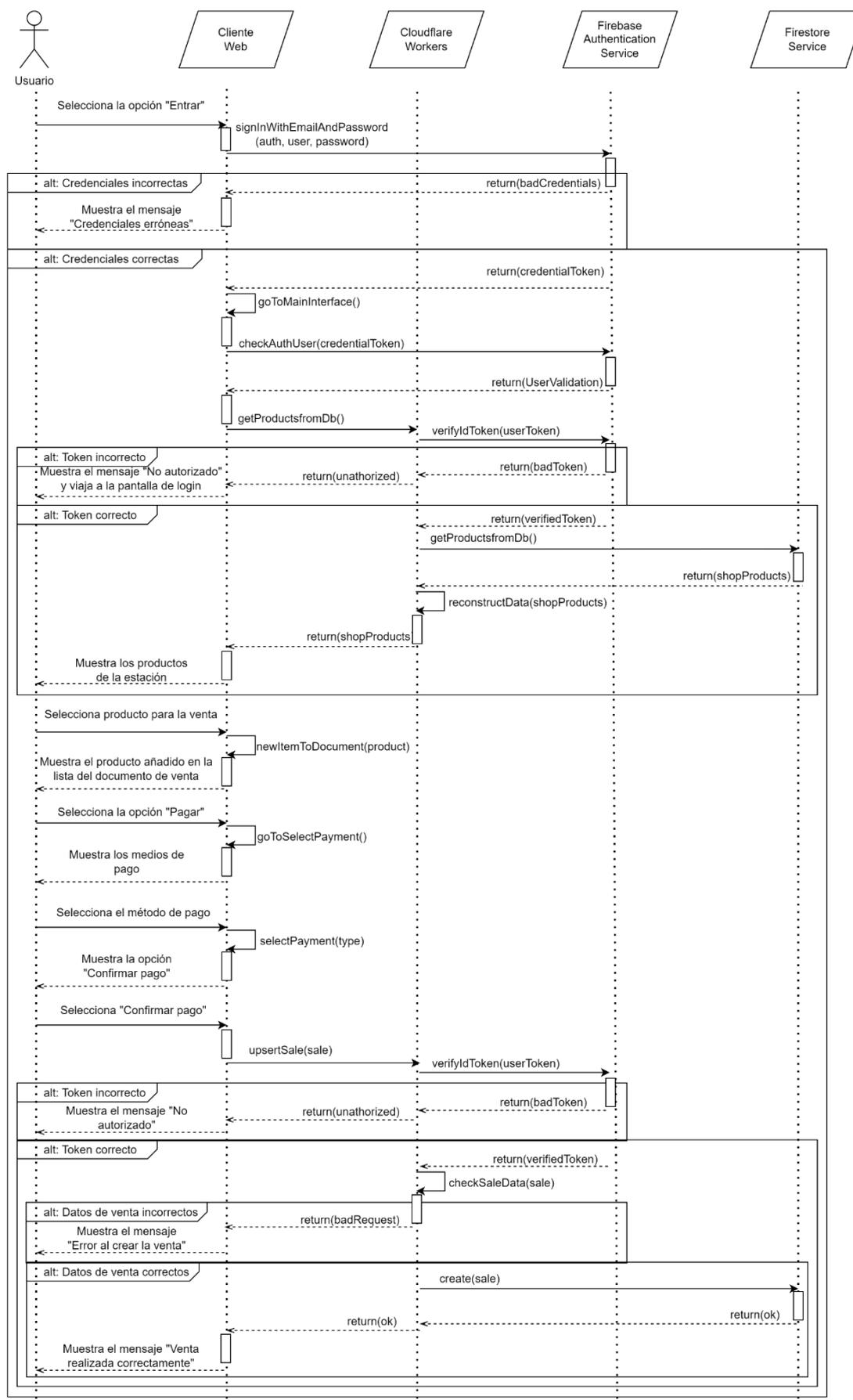


Figura 20. Diagrama de secuencia - Iniciar sesión y Venta.

4.3.2. Diagrama de secuencia - Registrar y editar datos de estación

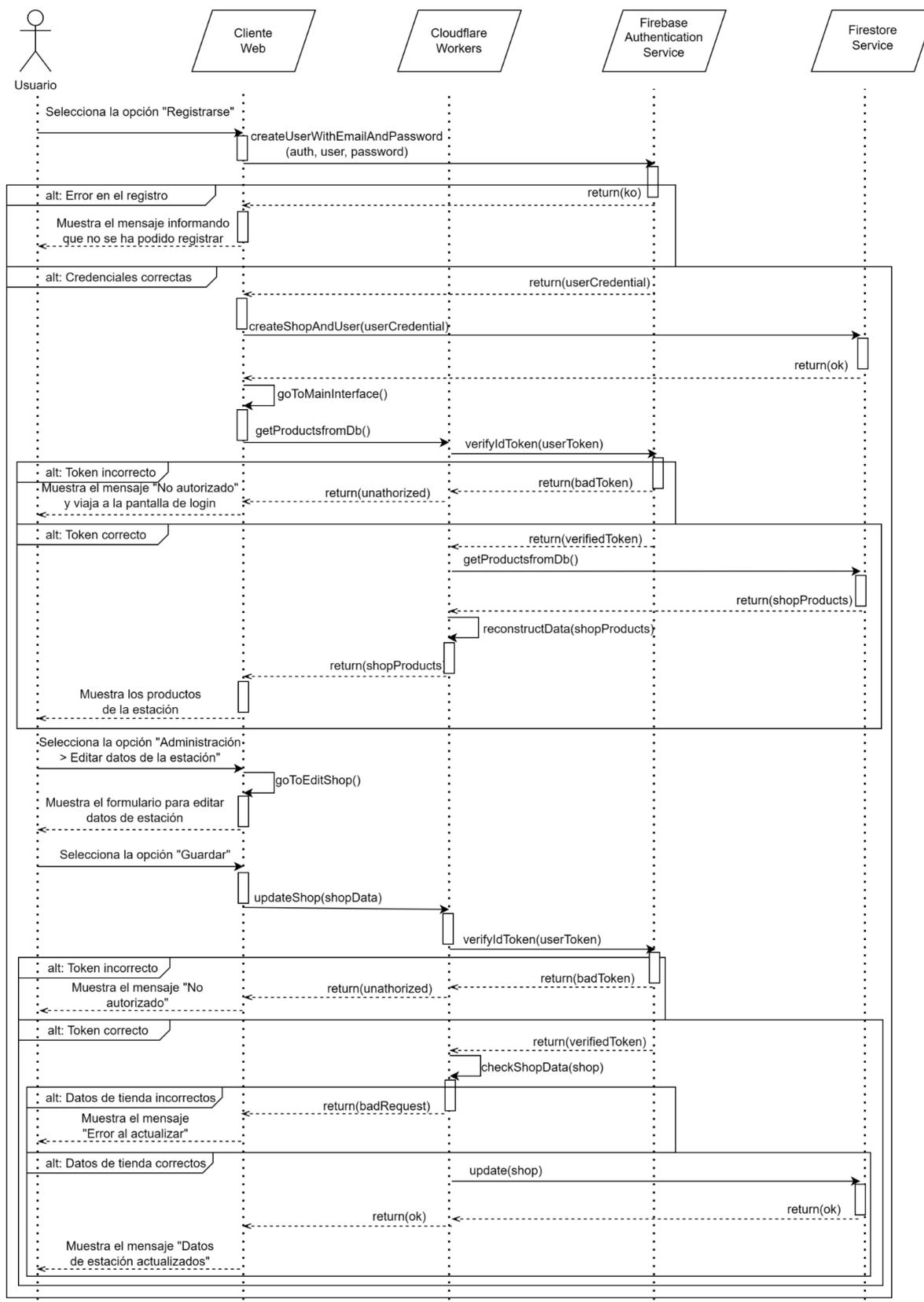


Figura 21. Diagrama de secuencia - Registrar y editar datos de estación.

4.3.3. Diagrama de secuencia - Crear usuario empleado y Eliminar usuario empleado

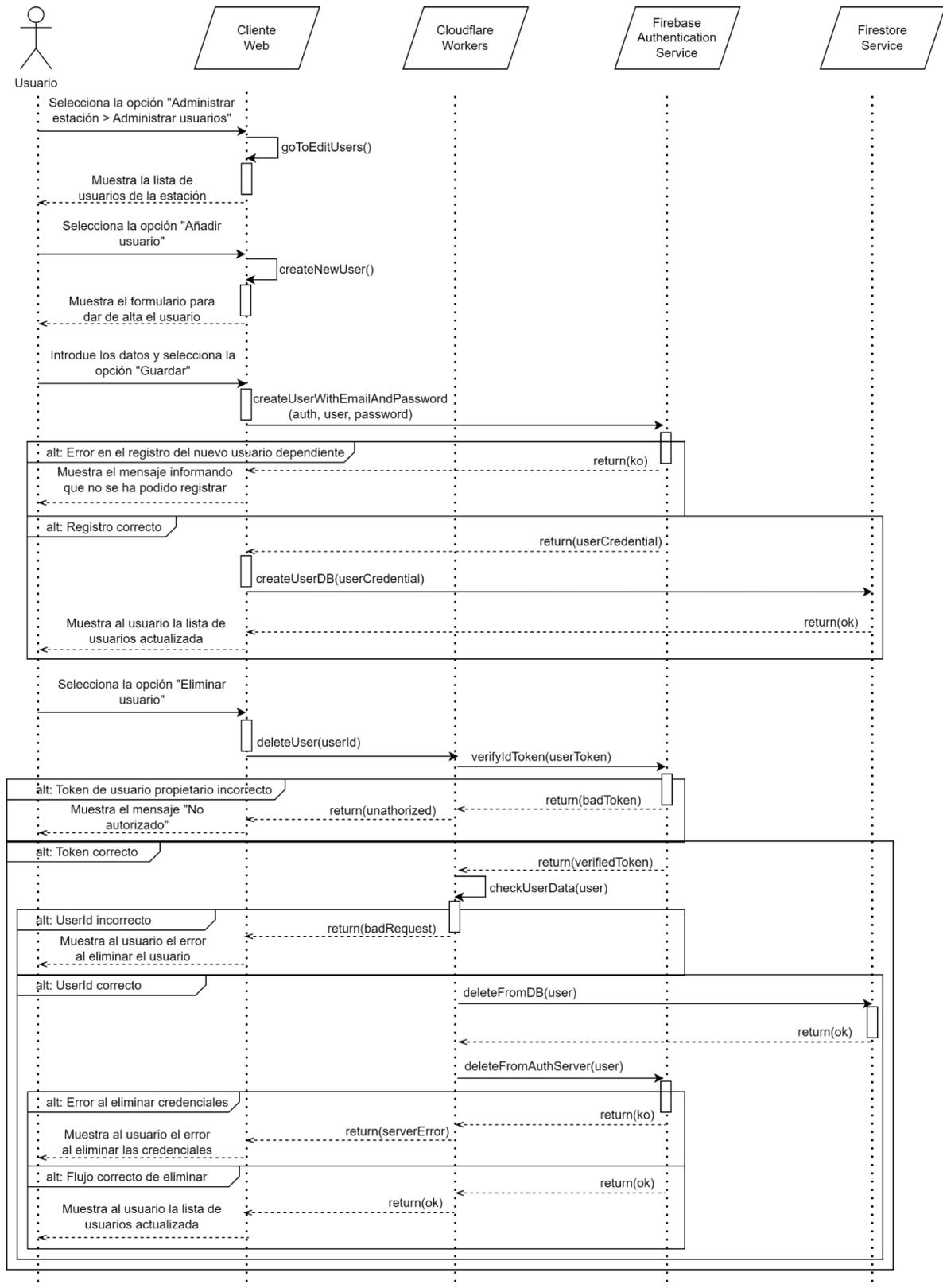


Figura 22. Diagrama de secuencia - Crear y Eliminar usuario dependiente.

4.4. Diseño de la interfaz

En este apartado se detallan las distintas interfaces de usuario con las que cuenta la aplicación. Para el diseño de estas interfaces se ha utilizado la herramienta de Google, Draw.io. Aunque las interfaces tienen un tema en tonos azulados, para el diseño y su mejor visualización en este documento se ha optado por el blanco y negro, con colores azul y rojo para los distintos botones en función de la acción que realicen.

Debemos indicar también, que se ha usado el diseño de una sola página (este es uno de los principales motivos por los que se ha optado por utilizar el *framework* de Angular), es decir, que simplemente se va a mostrar una única interfaz principal y sobre ella se van a crear los distintos componentes que irán alternándose en función de las distintas opciones que se vayan seleccionando, pero siempre con la misma base.

4.4.1. Inicio de sesión

Esta interfaz será la primera pantalla que vea el usuario cuando entre a la aplicación y no esté autenticado. Contiene un formulario para introducir las credenciales y 3 opciones para iniciar sesión y viajar a la pantalla principal, ir a la pantalla de registro e ir a la pantalla de recuperar contraseña. Además, cuenta con un apartado donde se puede ver la información de la aplicación.



<p>Usuario</p> <input type="text"/>	<h2>TPVa</h2> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
-------------------------------------	---

Figura 23. Interfaz de usuario - Inicio de sesión.

4.4.2. Registro

Esta interfaz es similar a la anterior, pero el formulario y el texto informativo de la aplicación están ocupando los espacios opuestos, de manera que de la sensación que es el apartado informativo el que se desplaza cuando se pasa de una interfaz a otra para descubrir el formulario de inicio de sesión o el de registro. Esta pantalla cuenta con las opciones de

registro, que crea el usuario y lo lleva a la pantalla principal, y la opción para volver a la pantalla de inicio de sesión.

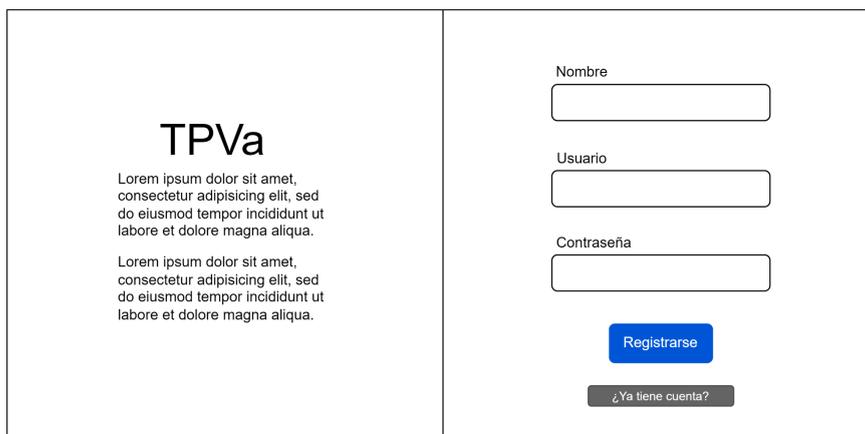


Figura 24. Interfaz de usuario - Registro.

4.4.3. Recuperar contraseña

Esta es la interfaz más simple de la aplicación. Se llega únicamente desde la pantalla de inicio de sesión y cuenta con un formulario para introducir el correo y las opciones de enviar correo para recuperar contraseña y la opción de ir a la pantalla de registro.



Figura 25. Interfaz de usuario - Recuperar contraseña.

4.4.4. Pantalla principal de estación

Esta es la pantalla principal de la aplicación. Todas las interfaces, excepto las tres que se han descrito anteriormente, se visualizarán dentro de esta interfaz. Está dividida en tres partes, el componente superior, que es donde se muestran los surtidores, el componente de la izquierda, que es donde se muestra el documento de venta con la lista de productos y las opciones de pago, y el componente de la derecha.

Este último componente es el más importante, ya que no solo contiene las distintas opciones para moverse por la aplicación, si no que es donde se mostrarán el resto de interfaces de la

aplicación que se describen en los siguientes apartados. Su interfaz principal muestra el árbol de productos seleccionables para la venta.

El componente de surtidores y el del documento de venta se mostrarán en todo momento en la aplicación (siempre que el usuario esté autenticado), pero estarán deshabilitadas hasta que se vuelva a la pantalla de productos.

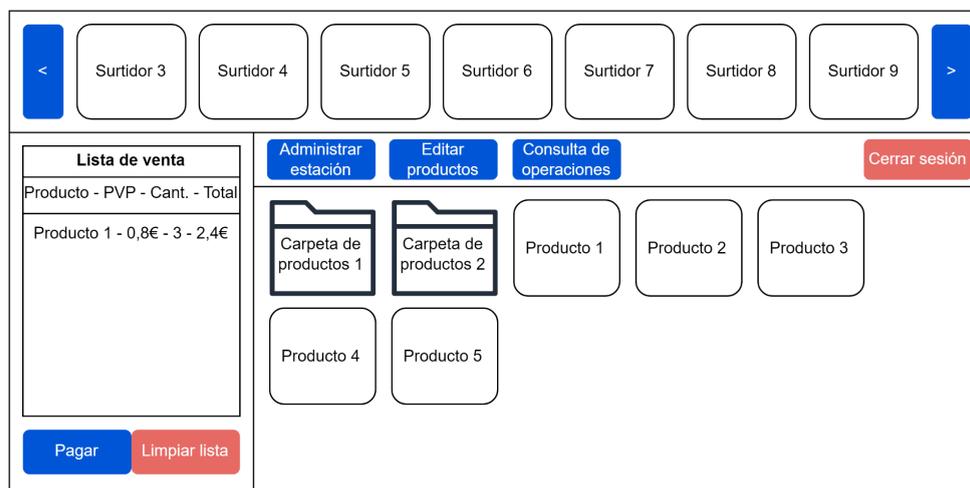


Figura 26. Interfaz de usuario - Pantalla principal de estación.

4.4.5. Administrar estación

Es la interfaz que aparecerá al pulsar sobre la opción “Administrar estación”. En ella se muestran las opciones para administrar los surtidores, los usuarios de la estación y editar los datos de la estación. También tiene la opción de regresar a la pantalla principal del árbol de productos.

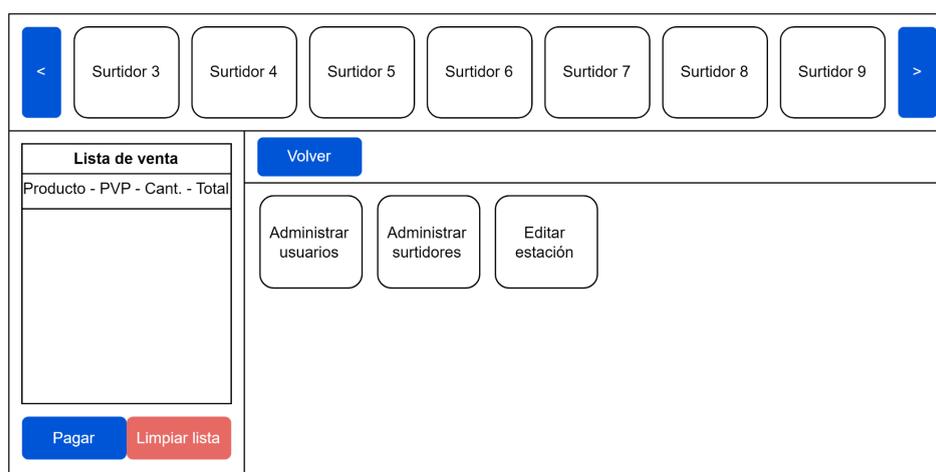


Figura 27. Interfaz de usuario - Administrar estación.

4.4.6. Administrar usuarios

Es la interfaz que aparecerá al pulsar sobre la opción “Administrar usuarios” sobre la pantalla anterior. En ella se muestra la lista de usuarios dados de alta en la estación. Las opciones en esta interfaz son seleccionar el usuario para editar o eliminar o añadir uno nuevo. Las opciones de editar y crear muestran el formulario de usuarios. También tiene la opción de volver a la pantalla anterior de “Administrar estación”.

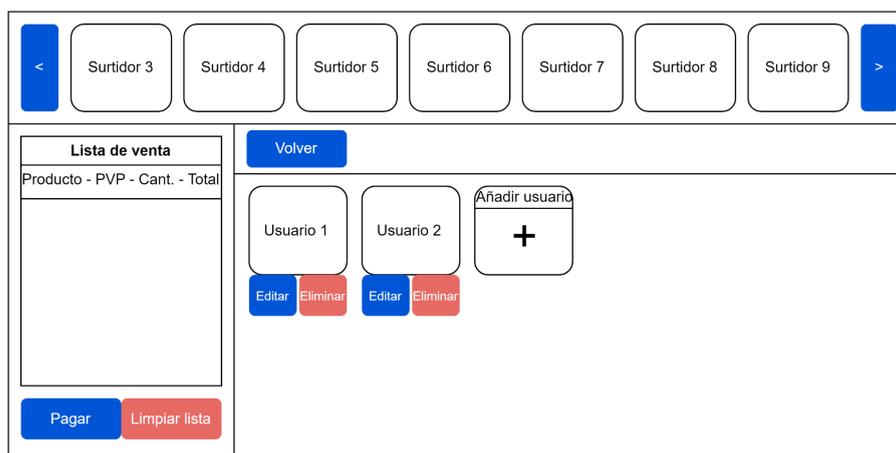


Figura 28. Interfaz de usuario - Administrar usuarios.

4.4.7. Formulario de usuarios

Esta interfaz muestra el formulario con la información de los usuarios. Tiene las opciones de guardar la información del usuario y darle de alta en la estación, y cancelar para volver a la pantalla anterior.

Figura 29. Interfaz de usuario - Formulario de usuarios.

4.4.8. Administrar surtidores

Es la interfaz que aparecerá al pulsar sobre la opción “Administrar surtidores” sobre la pantalla de “Administrar estación”. En ella se muestra la lista de surtidores dados de alta en la estación. Las opciones en esta interfaz son seleccionar el surtidor para editar o eliminar o añadir uno nuevo. Las opciones de editar y crear muestran el formulario de surtidores. También tiene la opción de volver a la pantalla anterior de “Administrar estación”.

Figura 30. Interfaz de usuario - Administrar surtidores.

4.4.9. Formulario de surtidores

Esta interfaz muestra el formulario con la lista de las mangueras de cada surtidor. Tiene las opciones de guardar la información del surtidor y cancelar para volver a la pantalla anterior.

The screenshot shows a web interface for configuring gas station nozzles. At the top, there is a navigation bar with buttons for nozzles 3 through 9, and left and right arrow buttons. Below this, the main content area is divided into two sections. On the left is a 'Lista de venta' (Sales List) table with columns for 'Producto', 'PVP', 'Cant.', and 'Total', and a 'Limpiar lista' (Clear list) button. On the right is a form for configuring nozzles. It includes four 'Manguera' (Nozzle) sections, each with a 'Combustible' (Fuel) input field. There is also an 'Añadir manguera' (Add nozzle) button with a plus sign. At the bottom of the form are 'Guardar' (Save) and 'Cancelar' (Cancel) buttons. At the bottom left of the main content area are 'Pagar' (Pay) and 'Limpiar lista' (Clear list) buttons.

Figura 31. Interfaz de usuario - Formulario de surtidores.

4.4.10. Editar datos de estación

Esta interfaz muestra el formulario con los datos de la estación. Tiene las opciones de guardar la información y cancelar para volver a la pantalla de “Administrar estación”.

The screenshot shows a web interface for editing station data. At the top, there is a navigation bar with buttons for nozzles 3 through 9, and left and right arrow buttons. Below this, the main content area is divided into two sections. On the left is a 'Lista de venta' (Sales List) table with columns for 'Producto', 'PVP', 'Cant.', and 'Total', and a 'Limpiar lista' (Clear list) button. On the right is a form for editing station data. It includes two input fields: 'Nombre' (Name) and 'Dirección' (Address). At the bottom of the form are 'Guardar' (Save) and 'Cancelar' (Cancel) buttons. At the bottom left of the main content area are 'Pagar' (Pay) and 'Limpiar lista' (Clear list) buttons.

Figura 32. Interfaz de usuario - Editar datos de estación.

4.4.11. Consulta de operaciones y stock

Es la interfaz que aparecerá al pulsar sobre la opción “Consulta de operaciones y stock”. En ella se muestran las opciones para consultar las operaciones realizadas, consultar el stock de productos y realizar devoluciones. También tiene la opción de regresar a la pantalla principal del árbol de productos.

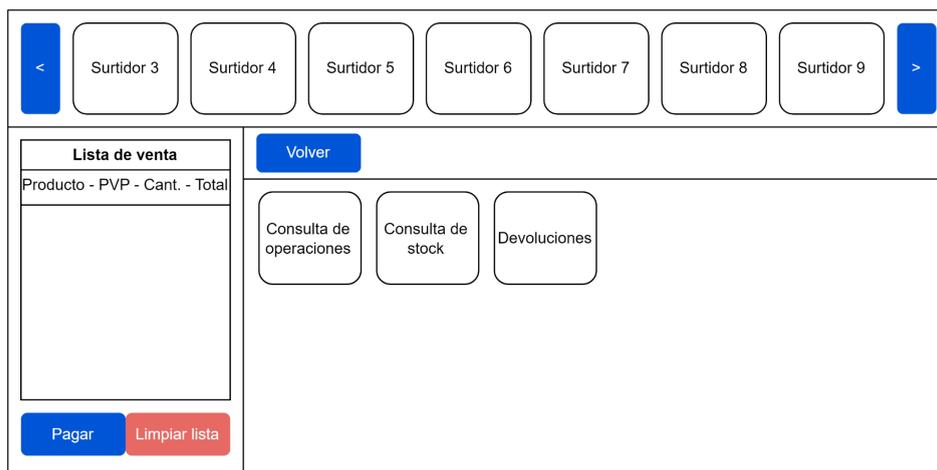


Figura 33. Interfaz de usuario - Consulta de operaciones y stock.

4.4.12. Consulta de operaciones

Esta interfaz muestra la lista de las operaciones realizadas en la estación. Tiene las opciones de exportar a excel la información de operaciones y volver a la pantalla de “Consulta de operaciones y stock”.

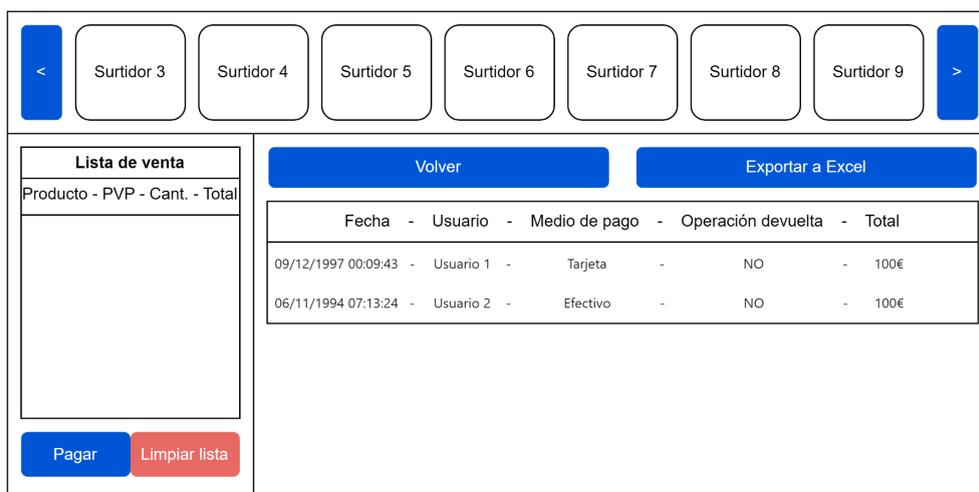


Figura 34. Interfaz de usuario - Consulta de operaciones.

4.4.13. Consulta de stock

Esta interfaz muestra la lista de los productos dados de alta en la estación. Tiene las opciones de exportar a excel la información de operaciones y volver a la pantalla de “Consulta de operaciones y stock”. También la opción de actualizar el stock de cada producto.

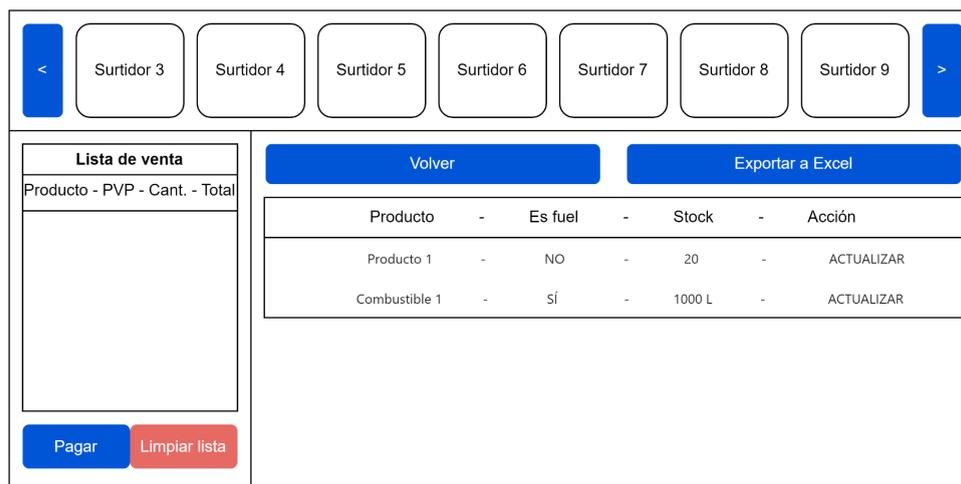


Figura 35. Interfaz de usuario - Consulta de stock.

4.4.14. Devoluciones

Esta interfaz muestra la lista de las operaciones realizadas en la estación que no han sido devueltas. Tiene las opciones de devolver cada una de las operaciones y volver a la pantalla de “Consulta de operaciones y stock”.

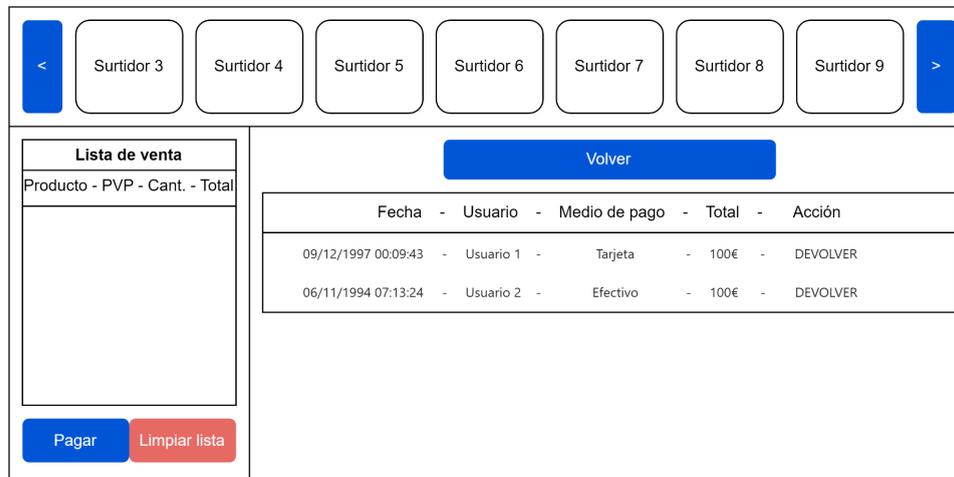


Figura 36. Interfaz de usuario - Devoluciones.

4.4.15. Editar productos

Es la interfaz que aparecerá al pulsar sobre la opción “Editar productos”. En ella se muestra la lista seleccionable de tipos de productos existentes para cada estación. Al seleccionar una de ellas nos mostrará la lista de productos combustible o el árbol de productos tienda. También tiene la opción de regresar a la pantalla principal del árbol de productos.

Figura 37. Interfaz de usuario - Editar productos.

4.4.16. Formulario de productos tienda

Esta interfaz muestra el formulario con la información de los productos de tienda. Tiene las opciones de guardar la información del producto y darle de alta en la estación, y cancelar para volver a la pantalla anterior.

Figura 38. Interfaz de usuario - Formulario de productos tienda.

4.4.17. Formulario de productos combustible

Esta interfaz muestra el formulario con la información de los productos combustible. Tiene las opciones de guardar la información del producto y darle de alta en la estación, y cancelar para volver a la pantalla de “Editar productos”.

The image shows a web application interface for a fuel station. At the top, there is a navigation bar with buttons for different fuel dispensers: Surtidor 3, Surtidor 4, Surtidor 5, Surtidor 6, Surtidor 7, Surtidor 8, and Surtidor 9. Below this is a main form area. On the left, there is a table titled "Lista de venta" with columns for "Producto", "PVP", "Cant.", and "Total". Below the table are two buttons: "Pagar" and "Limpiar lista". On the right, there is a form with three input fields labeled "Nombre", "Precio", and "Stock". Below these fields are two buttons: "Guardar" and "Cancelar".

Lista de venta			
Producto	PVP	Cant.	Total

Nombre: Precio: Stock:

Guardar Cancelar

Pagar Limpiar lista

Figura 39. Interfaz de usuario - Formulario de productos combustible.

5. Implementación

En este apartado se detalla la estructura y la ejecución práctica del proyecto. Exploramos cómo está organizado el proyecto en términos generales y nos adentramos en los detalles importantes del código que hacen posible la solución propuesta y ,de ese modo, entender cómo se conectan las diferentes partes del proyecto y qué aspectos técnicos han sido clave para su desarrollo y funcionamiento.

5.1. Estructura del proyecto

En este apartado se describe la estructura de la solución, así como la funcionalidad de cada uno de los módulos que componen la aplicación y la relación entre los mismos. La estructura completa se puede observar en la Figura 40.

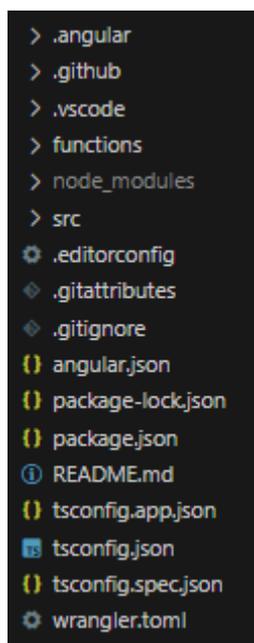


Figura 40. Estructura completa de la aplicación.

La lógica propia de la aplicación se encuentra únicamente en las carpetas “src” y “function/api/” que detallaremos más adelante. Los archivos destacados en esta estructura son los siguientes:

.github/workflows/deploy.yml: Los archivos *.yml* son archivos de configuración, con una sintaxis propia, pero equivalente a un archivo *.json*. En este caso, el fichero “deploy.yml” indica la configuración del despliegue, es decir, los pasos a seguir por la plataforma GitHub Actions una vez se haya integrado el código en el repositorio de GitHub. Este fichero contiene una serie de pasos para el despliegue, los cuales son: descargar el código del

repositorio, instalar la versión de Node.js y todas sus dependencias en el servidor de despliegue, hacer un *build* del código y, por último, desplegar la solución en Cloudflare Pages. Github Actions se encargará de ejecutar la configuración de despliegue de cada uno de los archivos *.yaml* que se encuentren en la carpeta “.github/workflows/”.

functions/api/: Contiene todas las funciones existentes en el *backend* de la aplicación. Las funciones son las que aparecen en la Figura 41. Cada función representa un *endpoint* del API y cada *endpoint* ejecuta las comprobaciones de seguridad y las llamadas a la base de datos específicas para llevar a cabo una tarea concreta. Cuando se despliega el código en Cloudflare Pages, a través de GitHub Actions como hemos visto anteriormente, Cloudflare se encarga de buscar los archivos dentro de la carpeta “functions/api/” para crear un *Worker* (este es el nombre que reciben las funciones *serverless* en Cloudflare) por cada archivo *.js* en base a unas especificaciones que veremos en el siguiente apartado.

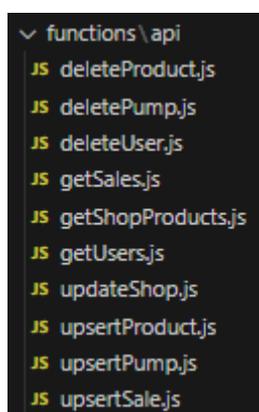


Figura 41. Carpeta de funciones del API del servidor de la aplicación.

src: En esta carpeta es donde se encuentra toda la lógica de la parte *frontend* que se ejecuta en el cliente. En ella se encuentran todos los ficheros estáticos que se van a desplegar en Cloudflare Pages. Dentro de esta carpeta vemos por un lado los ficheros “index.html”, “main.ts” y “style.scss” que son la base a partir de la cual se van a construir y mostrar todos los componentes de los ficheros estáticos de la aplicación. Por otro lado existen las carpetas “src/app”, “src/assets” y “src/utills”, las cuales se describen más adelante.

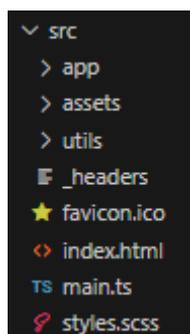


Figura 42. Carpeta “src” del proyecto.

src/app: En esta carpeta se encuentran todos los componentes del proyecto Angular creados para la solución. El componente principal es “app.component”, el cual se monta en el fichero principal “index.html” y se encarga de redirigir al componente pertinente en función de la ruta existente en la petición. Para ello utilizamos el fichero “app-routing.module.ts”.

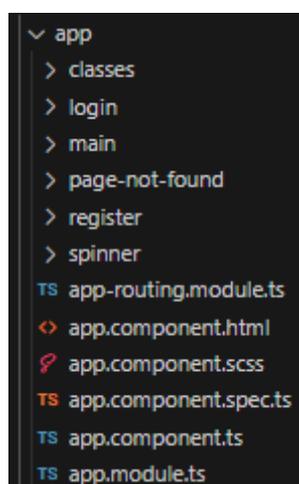


Figura 43. Carpeta “src/app” del proyecto.

En cuanto a los componentes, cada uno de los principales tiene una carpeta independiente, como pueden ser “login”, “register” o “main”. Estos componentes cuentan con el archivo *.html* que especifica la estructura del componente, el archivo *.scss* que especifica el estilo de la estructura del componente y el fichero *.ts* en el que se especifica la lógica del cliente relativa a este componente. Dentro de cada carpeta del componente, además de estos archivos, también se encuentran las carpetas con los subcomponentes que se utilizan en cada componente principal (por ejemplo, en la carpeta “main” se encuentra la carpeta “document” con cada archivo relativo al subcomponente “document”).

src/app/classes/dbModels.ts: Este archivo contiene todas las clases que representan los datos propios de la base de datos de la aplicación. Cuando se hace una llamada al *backend* para obtener los datos, se convierten siempre a una instancia nueva de una clase existente en este fichero. En la Figura 44 se puede observar un ejemplo de la clase *ShopsDbModel* que representa los datos de la tabla *Shops*.

```
export class ShopDbModel {
  id: string;
  address: string;
  name: string;
  products: string[];
  pumps: PumpDbModel[];

  constructor(id: string, address: string, name: string, products: string[], pumps: PumpDbModel[]) {
    this.id = id;
    this.address = address;
    this.name = name;
    this.products = products;
    this.pumps = pumps;
  }

  getJSPumps() {
    return JSON.parse(JSON.stringify(this.pumps));
  }
}
```

Figura 44. Ejemplo de clase existente en “src/app/classes/dbModels.ts”.

src/assets/images: En esta carpeta se encuentran todas las imágenes que se muestran en la aplicación.

src/utills: En esta carpeta se encuentran todos los servicios que agregan funcionalidad a los distintos componentes. Los ficheros “documentService.ts” y “pumpService.ts” se encargan de transmitir los eventos entre los distintos componentes de la solución. El fichero “firebase.ts” únicamente se encarga de crear la configuración para crear la comunicación con Firebase. En cuanto al fichero “databaseService.ts” se detalla más adelante.

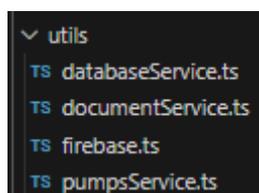


Figura 45. Carpeta “src/utills”.

src/app/utills/databaseService.ts: Este fichero contiene todas las llamadas al API de funciones de la aplicación para acceder a base de datos y convertir los datos que llegan de las distintas funciones en instancias de las clases existentes en el fichero “dbModels.ts”.

node_modules: Contiene todas las dependencias de Node.js de nuestra solución. No se *commitea* al repositorio, ya que se genera automáticamente con NPM, el gestor de dependencias del proyecto.

.editorconfig: Configuración del editor de código, en este caso Visual Studio Code.

.gitignore: Fichero en el que indicamos a *Git* los ficheros y carpetas que debe ignorar.

angular.json: Contiene la configuración del *framework* de Angular. Este archivo incluye las rutas, nombres, versiones y configuraciones de las dependencias de Angular.

package.json: Contiene la configuración de la aplicación, como son el nombre de la aplicación, la versión, y todas las dependencias necesarias para su correcta ejecución.

tsconfig.json: Este archivo contiene la configuración del lenguaje TypeScript.

wrangler.toml: Contiene la configuración específica de Wrangler. Wrangler es la herramienta de comandos de Cloudflare Workers y Pages. En nuestro caso únicamente utilizamos el fichero para indicar que no envía métricas del uso de la aplicación a Cloudflare.

5.2. Detalles de implementación

En este apartado se mencionan dos partes del código especialmente relevantes y de cierta complejidad técnica a nivel de implementación. Concretamente, se detalla el algoritmo para moverse y mostrar correctamente la lista de surtidores y la función *serverless* de eliminación de usuarios de una estación “deleteUser.js”.

Algoritmo de lista de surtidores: Este algoritmo se encuentra en el archivo “pump.component.ts” del componente “src/app/main/pump”. Los datos con los que trabaja se encuentran en el campo “pumps” que es una lista de la clase “Pump” que se detalla en el fichero “src/app/classes/pump.ts”.

```
import { ProductDbModel } from "../dbModels";

export class Pump {
  numberId: number;
  pipesCount: number;
  pipes: Pipe[];

  constructor(numberId: number, pipes: Pipe[]) {
    this.numberId = numberId;
    this.pipesCount = pipes.length;
    this.pipes = pipes;
  }
}

export class Pipe {
  fuel: ProductDbModel;

  constructor(fuel: ProductDbModel) {
    this.fuel = fuel;
  }
}
```

Figura 46. Fichero “pump.ts”.

Esta lista de surtidores de la clase “Pump” se obtiene de los datos de la clase “ShopDbModel”, los cuales a su vez se obtienen de la llamada a *backend* para obtenerlo de la base de datos. Para convertir los datos de la lista de instancias de la clase “PumpDbModel” a la lista de la clase “Pump” se utiliza el método “GetPumps()” que se muestra en la Figura 47.

```
private GetPumps(shop: ShopDbModel, products: ProductDbModel[]) {
  const pumpsDb = shop.pumps;

  const pumps: Pump[] = [];
  pumpsDb.forEach(pumpDb => {
    const pipes = this.GetPipes(pumpDb, products);
    const pump: Pump = new Pump(pumpDb.PumpId, pipes);
    pumps.push(pump);
  });
  this.pumps = pumps.filter(pump => pump.pipes.length > 0);
  this.totalPumps = this.pumps.length;
}
```

Figura 47. Método “GetPumps()” para convertir las listas de surtidores.

Una vez obtenida la lista de surtidores para la estación, se ejecuta el método “onResize()”. Este método se ejecutará siempre que cambie el tamaño de la ventana en la que se muestra la aplicación.

```
onResize(): void {
  if (this.pumps && this.pumps.length > 0) {
    const prevButton = document.getElementById('prevButton');
    const nextButton = document.getElementById('nextButton');
    if (prevButton) prevButton.hidden = true;
    if (nextButton) nextButton.hidden = true;
    for (let i = 1; i <= this.pumps.length; i++) {
      const pump = document.querySelector<HTMLElement>(`.pump-${i}`);
      if (pump) pump.hidden = false;
    }

    const container = document.getElementById('pumps');
    const pump = document.querySelector(`.pump-${this.pumps[0].numberId}`);
    var pumpWidth = 0;
    var containerWidth = 0;
    if (pump && container) {
      pumpWidth = pump?.clientWidth + 20;
      containerWidth = container?.clientWidth;
      this.maxScrollLeft = (pumpWidth * this.pumps.length) - containerWidth;
    }
    if (this.maxScrollLeft > 0) {
      const nextButton = document.getElementById('nextButton');
      if (nextButton) nextButton.hidden = false;

      this.firstShownPump = 0;
      this.lastShownPump = (containerWidth / pumpWidth | 0);

      for (let i = this.lastShownPump; i < this.pumps.length; i++) {
        const pump = document.querySelector<HTMLElement>(`.pump-${this.pumps[i].numberId}`);
        if (pump) pump.hidden = true;
      }
    }
  }
}
```

Figura 48. Método “onResize()”.

Lo primero que se comprueba en este método es la lista de surtidores. En el caso de que esté vacía no hacemos nada, ya que simplemente se mostrará al usuario el mensaje para informar al usuario de ello. En el caso de que no esté vacía, lo primero que hacemos es mostrar todos los surtidores y ocultar los botones para navegar por la lista de surtidores. Una vez hecho esto calculamos el ancho de cada surtidor y el ancho total del componente, ambos en píxeles. Con estos dos valores podemos obtener el campo “maxScrollLeft”, que indica el ancho que ocupan de más los surtidores respecto del total de la pantalla. Si este valor es igual o menor

de 0 no hacemos nada, ya que la lista de surtidores cabe en la pantalla sin ningún problema. Por el contrario, si el valor de “maxScrollLeft” es mayor de 0 mostramos el botón “nextButton”, que permite desplazarnos hacia la derecha en la lista. Además, guardamos el primer surtidor que se muestra y el último surtidor que cabe en la pantalla, y ocultamos los siguientes a este último.

Una vez hecho esto, ya hemos calculado los surtidores que debemos mostrar y si debemos mostrar el botón para desplazarnos por la pantalla o no es necesario. Para desplazarnos por la lista existen dos botones “nextButton” y “prevButton”. Si seleccionan estos botones se ejecutan sus respectivos eventos que están declarados en los métodos “addScrollRight()” y “addScrollLeft()” respectivamente.

```
nextButton.onclick = function () {
  const firstShowPumpId = context.pumps[context.firstShownPump].numberId;
  const firstNotShownPumpId = context.pumps[context.lastShownPump].numberId;
  const firstShownPump = document.querySelector<HTMLInputElement>(`.pump-${firstShowPumpId}`);
  const firstNotShownPump = document.querySelector<HTMLInputElement>(`.pump-${firstNotShownPumpId}`);
  if (firstShownPump) firstShownPump.hidden = true;
  if (firstNotShownPump) firstNotShownPump.hidden = false;

  context.firstShownPump++;
  context.lastShownPump++;

  if (context.totalPumps === context.lastShownPump) {
    nextButton.hidden = true;
  }
  if (prevButton) prevButton.hidden = false;
};
```

Figura 49. Evento “onclick” del botón “nextButton”.

Cuando se pulsa sobre el botón “nextButton” se ejecuta el evento que se muestra en la Figura 49. Lo primero que se hace es obtener el ID y el componente HTML del primer surtidor que se muestra y del primer surtidor que no se muestra por la derecha de la pantalla. Una vez obtenidos estos datos, ocultamos el primer surtidor que se mostraba y mostramos el primer surtidor que estaba oculto por la derecha. Si el surtidor que acabamos de mostrar es el último de la lista, dejamos de mostrar el botón “nextButton” ya que no existen más surtidores a la derecha en la lista. Por último, mostramos el botón “prevButton” para poder desplazarnos a la izquierda en la lista en caso de que no estuviese ya mostrándose.

```
prevButton.onclick = function () {
  const fistLeftNotShownPumpId = context.pumps[context.firstShownPump - 1].numberId;
  const lastShownPumpId = context.pumps[context.lastShownPump - 1].numberId;
  const fistLeftNotShownPump = document.querySelector<HTMLElement>(`.pump-${fistLeftNotShownPumpId}`);
  const lastShownPump = document.querySelector<HTMLElement>(`.pump-${lastShownPumpId}`);
  if (lastShownPump) lastShownPump.hidden = true;
  if (fistLeftNotShownPump) fistLeftNotShownPump.hidden = false;

  context.firstShownPump--;
  context.lastShownPump--;

  if (context.firstShownPump === 0) {
    prevButton.hidden = true;
  }

  if (nextButton) nextButton.hidden = false;
};
```

Figura 50. Evento “onclick” del botón “prevButton”.

Cuando se pulsa sobre el botón “prevButton” la lógica a ejecutar es prácticamente igual, pero al revés. Se ejecuta el evento que se muestra en la Figura 50. Lo primero que se hace es obtener el ID y el componente HTML del último surtidor que se muestra y del primer surtidor que no se muestra por la izquierda de la pantalla. Una vez obtenidos estos datos, ocultamos el último surtidor que se mostraba y mostramos el primer surtidor que estaba oculto por la izquierda. Si el surtidor que acabamos de mostrar es el primero de la lista, dejamos de mostrar el botón “prevButton” ya que no existen más surtidores a la izquierda en la lista. Por último, mostramos el botón “nextButton” para poder desplazarnos a la derecha en la lista en caso de que no estuviese ya mostrándose.

Función de eliminación de usuarios: Ponemos de ejemplo esta función debido a que es la más interesante a nivel de lógica, ya que es necesario hacer varias comprobaciones de seguridad, así como varias llamadas tanto Firebase Authentication como Firestore Database.

Esta función se encuentra en el archivo “functions/api/deleteUser.js”. Cuando desplegamos la solución Cloudflare Pages, automáticamente despliega como Workers cada uno de los archivos .js que existen en esta carpeta. Estos archivos deben contar con una función llamada “onRequest()” para poder desplegarse correctamente, de lo contrario se obviará y no se creará la función en Cloudflare Workers.

Una vez se haga la llamada a la función, lo primero que haremos, tanto para esta función como para el resto, será obtener los datos de Firebase a través de los *secrets* declarados en nuestro proyecto Cloudflare. Esto se explica más adelante en este documento en el Manual de Despliegue.

Como se puede ver en la Figura 51, con los datos de Firebase, comprobamos que el usuario que lanza la petición, esté autenticado en la aplicación y sea un *usuario propietario*. De lo contrario lanzaremos una excepción como respuesta.

```

export async function onRequest({env, request}) {
  const CREDENTIALS = JSON.parse(env.GOOGLE_CREDENTIALS);
  const FIRESTORE_URL = `https://firestore.googleapis.com/v1/projects/${CREDENTIALS.project_id}/databases/(default)/documents`;
  try {
    // Authenticate request
    const ownerIdToken = request.headers.get('authorization')?.replace(/^bearer /i, '').trim() || '';
    if (ownerIdToken === '') {
      throw new HttpError('Unauthenticated', 401);
    }
    const ownerId = await verifyIdToken({
      idToken: ownerIdToken,
      projectId: CREDENTIALS.project_id,
      credentials: CREDENTIALS,
    }).then(res => res.user_id);

    // Get admin access token
    const adminAccessToken = await getAccessToken({
      credentials: CREDENTIALS,
      scope: 'https://www.googleapis.com/auth/cloud-platform',
    });

    // Verify invoker is the owner of the shop
    const ownerDoc = await fetch(`${FIRESTORE_URL}/Users/${ownerId}`, {
      headers: {
        'Authorization': `Bearer ${adminAccessToken}`,
      },
    }).then(res => res.json());
    if (ownerDoc.fields.Role.stringValue !== 'owner') {
      throw new HttpError('User is not an owner', 401);
    }
  }
}

```

Figura 51. Inicio de función “deleteUser.js”.

Una vez hemos verificado el usuario, empezamos a verificar los datos de la petición, para ello, tal y como se ve en la Figura 52, primero obtendremos el id del usuario a eliminar de la petición, comprobaremos que este usuario no sea el propio *usuario propietario* de la estación y que el usuario a eliminar pertenezca a la misma estación que el *usuario propietario* que lanza la petición. En el caso de que no se verifique ninguno de estos casos, lanzaremos una respuesta con su mensaje de error y el código HTTP 400.

```

// Get userId from request
const userId = await request.json().then(res => `${res.userId || ''}`);
if (userId.length < 10) {
  throw new HttpError('Invalid user ID', 400);
}

// Verify user to delete is not the owner
if (ownerDoc.id === userId) {
  throw new HttpError('User can not delete itself', 400);
}

// Verify user belongs to owner shop
const userDoc = await fetch(`${FIRESTORE_URL}/Users/${userId}`, {
  headers: {
    'Authorization': `Bearer ${adminAccessToken}`,
  },
}).then(res => res.json());
if (ownerDoc.fields.Shop.referenceValue !== userDoc.fields.Shop.referenceValue) {
  throw new HttpError('User does not belongs to current shop', 400);
}

```

Figura 52. Comprobaciones de la petición en la función “deleteUser.js”.

Tras verificar la petición y el usuario, se procede a eliminar el usuario. Primero eliminamos el usuario de la base de datos de Firestore. Después lanzamos la petición de eliminar el usuario de Firebase Authentication. Para esto es necesario obtener el token de administrador a través

de las credenciales obtenidas al inicio de la función. Si todo ha ido correctamente respondemos con un código HTTP 200 indicando que se ha eliminado el usuario correctamente.

```
// Delete user entry in Firestore
const deleteFirestoreResponse = await fetch(`${FIRESTORE_URL}/Users/${userId}`, {
  method: 'delete',
  headers: {
    'Authorization': `Bearer ${adminAccessToken}`,
  },
});
if (deleteFirestoreResponse.status !== 200) {
  throw new HttpError('User not found', 404);
}

// Delete account in Identity Toolkit
const idToken = await getIdToken({
  uid: userId,
  apiKey: CREDENTIALS.api_key,
  projectId: CREDENTIALS.project_id,
  credentials: CREDENTIALS,
}).then(res => res.idToken);
const deleteIdentityUrl = `https://identitytoolkit.googleapis.com/v1/accounts:delete?key=${CREDENTIALS.api_key}`;
const deleteIdentityResponse = await fetch(deleteIdentityUrl, {
  method: 'post',
  headers: {
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({idToken}),
});
if (deleteIdentityResponse.status !== 200) {
  throw new HttpError('Failed to delete credentials', 500);
}

return getJsonResponse({success: true});
```

Figura 53. Peticiones para eliminar el usuario en “deleteUser.js”.

En el caso de que se haya lanzado alguna excepción porque haya ido mal alguna de las verificaciones o intentos de eliminar el usuario, se ejecuta la lógica que se puede ver en la Figura 54. Primero se comprueba si el error es “HttpError” que es propio de nuestra solución. En ese caso se responde con el código 401 y el mensaje de error pertinente.

```
} catch (e) {
  if (e instanceof HttpError) {
    return getJsonResponse({success: false, reason: e.message}, 401);
  }
  console.error('Unexpected error', e);
  return getJsonResponse({success: false, reason: 'Unexpected error'}, 400);
}
```

Figura 54. Peticiones para eliminar el usuario en “deleteUser.js”.

La clase “HttpError” se puede ver en la Figura 55 y simplemente es una clase que hereda de la ya existente en Node.js “Error” y añade el código de error. Esta clase está presente en todas las funciones de la aplicación.

```
class HttpError extends Error {  
  /**  
   * @param {string} message Error message  
   * @param {string} code HTTP status code  
   */  
  constructor(message, code) {  
    super(message);  
    this.code = code;  
  }  
}
```

Figura 55. Clase “HttpError”.

Del mismo modo, también existe para función del API, una función propia para obtener la respuesta del API. Esta función espera como parámetros los datos de respuesta para convertirlos en el cuerpo en formato *json* de la respuesta y el código HTTP de la misma.

```
function getJsonResponse(data, status = 200) {  
  const body = JSON.stringify(data);  
  return new Response(body, {  
    status,  
    headers: {  
      'Content-Type': 'application/json',  
    },  
  });  
}
```

Figura 56. Función “getJsonResponse()”.

6. Pruebas

Las pruebas de caja negra son fundamentales para validar las funcionalidades del código sin necesidad de conocer los detalles internos de su implementación. Este tipo de pruebas se centra en verificar las entradas y salidas del sistema, sin considerar los procesos internos que ocurren entre ellas. En las pruebas de caja negra, se especifican las entradas de datos para cada acción y se define el resultado esperado, lo que permite evaluar el correcto funcionamiento de la aplicación desde una perspectiva externa.

6.1. Registro e inicio de sesión

Tabla 45. PCN-01 - Registro de *usuario propietario*.

PCN-01	Registro de <i>usuario propietario</i>
Descripción	Registro de nuevos <i>usuarios propietarios</i> y crear una nueva estación.
Precondiciones	El usuario no debe estar registrado previamente en la aplicación.
Datos de entrada	Nombre: Pruebas Usuario: goriyed565@togito.com Contraseña: 12341235
Resultado esperado	El usuario se registra en la aplicación y se muestra la pantalla principal con el árbol y de productos y la lista de surtidores del comercio vacía.

Tabla 46. PCN-02 - Inicio de sesión.

PCN-02	Inicio de sesión
Descripción	Inicio de sesión del usuario en la aplicación.
Precondiciones	El usuario debe estar registrado previamente en la aplicación.
Datos de entrada	Usuario: goriyed565@togito.com Contraseña: 12341235
Resultado esperado	El usuario se identifica correctamente, inicia una nueva sesión y se muestra la pantalla principal de productos y la lista de surtidores del comercio al que pertenece el usuario.

Tabla 47. PCN-03 - Recuperar contraseña.

PCN-03	Recuperar contraseña
Descripción	Recuperar la contraseña olvidada del usuario.
Precondiciones	El usuario debe estar registrado previamente en la aplicación.
Datos de entrada	Correo electrónico: goriyed565@togito.com Contraseña: 12341235

PCN-03	Recuperar contraseña
Resultado esperado	Se envía un correo al correo electrónico introducido en el formulario con un enlace para poder introducir la nueva contraseña

6.2. Editar productos

Tabla 48. PCN-04 - Dar de alta un producto de tienda.

PCN-04	Dar de alta un producto de tienda
Descripción	Crear un nuevo producto de tienda en la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación.
Datos de entrada	Nombre del producto: Producto 1 Precio: 1 Stock: 100 Carpeta: Raíz Habilitado para la venta: Sí Carpeta de productos: No
Resultado esperado	El producto se da de alta correctamente en la aplicación y se muestra la pantalla principal con el nuevo producto.

Tabla 49. PCN-05 - Dar de alta una carpeta de productos.

PCN-05	Dar de alta una carpeta de productos
Descripción	Crear una nueva carpeta de productos en la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación.
Datos de entrada	Nombre del producto: Carpeta 1 Carpeta: Raíz Habilitado para la venta: Sí Carpeta de productos: Sí
Resultado esperado	La carpeta se da de alta correctamente en la aplicación y se muestra la pantalla principal con la nueva carpeta.

Tabla 50. PCN-06 - Dar de alta un combustible.

PCN-06	Dar de alta un producto combustible
Descripción	Crear un nuevo producto combustible en la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente.
Datos de entrada	Nombre del producto: Carburante 1 Precio: 1.5 Stock: 1000
Resultado esperado	El producto se da de alta correctamente en la aplicación y se muestra la pantalla principal. El producto aparecerá deshabilitado en la lista de carburantes.

Tabla 51. PCN-07 - Modificar un producto de tienda.

PCN-07	Modificar un producto de tienda
Descripción	Modificar un producto de tienda en la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir el producto en la aplicación para la estación.
Datos de entrada	Nombre del producto: Producto 1 Precio: 2 Stock: 200 Carpeta: Carpeta 1 Habilitado para la venta: Sí Carpeta de productos: No
Resultado esperado	El producto se modifica correctamente en la aplicación y se muestra la pantalla principal. El producto se muestra en el árbol de productos dentro de la carpeta "Carpeta 1".

Tabla 52. PCN-08 - Eliminar un producto de tienda.

PCN-08	Eliminar un producto de tienda
Descripción	Eliminar un producto de tienda en la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir el producto en la aplicación para la estación.
Datos de entrada	Seleccionar "Eliminar Producto 1" sobre la lista de productos.
Resultado esperado	El producto se elimina correctamente en la aplicación y se muestra la pantalla principal. En la carpeta "Carpeta 1" no se muestra el producto "Producto 1".

Tabla 53. PCN-09 - Modificar un producto combustible.

PCN-09	Modificar un producto combustible
Descripción	Modificar un producto de tienda en la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir el producto en la aplicación para la estación.
Datos de entrada	Nombre del producto: Carburante 1 Precio: 2.154 Stock: 2000
Resultado esperado	El producto se modifica correctamente en la aplicación y se muestra la pantalla principal. El producto aparecerá modificado en la lista de carburantes.

Tabla 54. PCN-10 - Eliminar un producto combustible.

PCN-10	Eliminar un producto combustible
Descripción	Eliminar un producto de tienda en la estación.

PCN-10	Eliminar un producto combustible
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir el producto en la aplicación para la estación.
Datos de entrada	Seleccionar “Eliminar Carburante 1” sobre la lista de combustibles.
Resultado esperado	El producto se elimina correctamente en la aplicación y se muestra la pantalla principal. En pantalla con la lista de carburantes no se muestra el producto “Carburante 1”.

6.3. Administrar estación

Tabla 55. PCN-11 - Crear nuevo *usuario dependiente*.

PCN-11	Crear nuevo <i>usuario dependiente</i>
Descripción	Crear un nuevo <i>usuario dependiente</i> para la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación.
Datos de entrada	Nombre: Usuario 2 Usuario: Contraseña: 12341234
Resultado esperado	El usuario se da de alta correctamente en la aplicación y se muestra la lista de usuarios de la estación, con el nuevo usuario dado de alta.

Tabla 56. PCN-12 - Modificar *usuario dependiente*.

PCN-12	Modificar <i>usuario dependiente</i>
Descripción	Modificar un <i>usuario dependiente</i> de la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir el producto en la aplicación para la estación.
Datos de entrada	Nombre: Usuario 3
Resultado esperado	El usuario se modifica correctamente en la aplicación y se muestra la lista de usuarios de la estación, con los datos del usuario modificados.

Tabla 57. PCN-13 - Eliminar *usuario dependiente*.

PCN-13	Eliminar <i>usuario dependiente</i>
Descripción	Eliminar <i>usuario dependiente</i> de la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir al menos un <i>usuario dependiente</i> en la estación.
Datos de entrada	Seleccionar “Eliminar Usuario 3” sobre la lista de usuarios.

PCN-13	Eliminar <i>usuario dependiente</i>
Resultado esperado	El usuario se elimina correctamente en la aplicación y se muestra la lista de usuarios de la estación, sin el usuario eliminado.

Tabla 58. PCN-14 - Modificar nombre y email de *usuario propietario*.

PCN-14	Modificar nombre y email de <i>usuario propietario</i>
Descripción	Modificar el nombre y el email del <i>usuario propietario</i> de la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación.
Datos de entrada	Nombre: Usuario 3 Usuario: goriyed565@togito.com
Resultado esperado	El usuario se modifica correctamente en la aplicación y se muestra la lista de usuarios de la estación, con los datos del usuario modificados.

Tabla 59. PCN-15 - Modificar contraseña de *usuario propietario*.

PCN-15	Modificar contraseña de <i>usuario propietario</i>
Descripción	Modificar la contraseña del <i>usuario propietario</i> de la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación.
Datos de entrada	Contraseña actual: 12341235 Nueva contraseña: 12341234
Resultado esperado	La contraseña del usuario se modifica correctamente en la aplicación y se muestra la lista de usuarios de la estación.

Tabla 60. PCN-16 - Crear nuevo surtidor sin combustibles.

PCN-16	Crear nuevo surtidor sin combustibles
Descripción	Crear un nuevo surtidor para la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la app.
Datos de entrada	Añadir nueva manguera: <<Sin seleccionar datos>>
Resultado esperado	El surtidor se da de alta correctamente en la aplicación y se muestra la lista de surtidores de la estación con el nuevo surtidor dado de alta. En la lista superior de surtidores no aparece el nuevo surtidor.

Tabla 61. PCN-17 - Crear nuevo surtidor con combustibles.

PCN-17	Crear nuevo surtidor con combustibles
Descripción	Crear un nuevo surtidor para la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir en la estación al menos un carburante dado de alta.
Datos de entrada	Manguera 1: Carburante 1
Resultado esperado	El surtidor se da de alta correctamente en la aplicación y se muestra la lista de surtidores de la estación con el nuevo surtidor dado de alta. En la lista superior de surtidores aparece el nuevo surtidor.

Tabla 62. PCN-18 - Modificar surtidor.

PCN-18	Modificar surtidor
Descripción	Crear un nuevo surtidor para la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir en la estación al menos un carburante y un surtidor dado de alta.
Datos de entrada	Manguera 1: Carburante 1 Manguera 2: Carburante 1
Resultado esperado	El surtidor se modifica correctamente en la aplicación y se muestra la lista de surtidores de la estación. En la lista superior de surtidores aparece el surtidor modificado.

Tabla 63. PCN-19 - Eliminar surtidor.

PCN-19	Eliminar surtidor
Descripción	Eliminar surtidor de la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir en la estación al menos un surtidor sin combustibles dado de alta.
Datos de entrada	Seleccionar la opción “Eliminar Surtidor 1”
Resultado esperado	El surtidor se elimina correctamente de la aplicación y se muestra la lista de surtidores de la estación sin el surtidor eliminado y con el número de surtidores actualizado correctamente. La lista superior de surtidores aparece sin modificaciones.

Tabla 64. PCN-20 - Modificar datos de la estación.

PCN-20	Modificar datos de la estación
Descripción	Editar nombre y dirección de la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación.
Datos de entrada	Nombre: Estación de Prueba Dirección: Calle de la Piruleta 33
Resultado esperado	Los datos de la estación se modifican correctamente en la aplicación y se muestra el menú de “Administrar estación”.

6.4. Ventas

Tabla 65. PCN-21 - Venta en efectivo de combustible.

PCN-21	Venta en efectivo de combustible
Descripción	Realizar una venta en efectivo de combustible.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Productos: Surtidor 1 - Carburante 1 - 5 litros - 10 € Medio de pago: Efectivo
Resultado esperado	La venta se realiza correctamente y se muestra la pantalla principal con la lista del ticket virtual vacía.

Tabla 66. PCN-22 - Venta en efectivo de producto tienda.

PCN-22	Venta en efectivo de producto tienda
Descripción	Realizar una venta en efectivo de producto de tienda.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Productos: Producto 1 - 2 unidades - 4 € Medio de pago: Efectivo
Resultado esperado	La venta se realiza correctamente y se muestra la pantalla principal con la lista del ticket virtual vacía.

Tabla 67. PCN-23 - Venta en efectivo de varios productos.

PCN-23	Venta en efectivo de varios productos
Descripción	Realizar una venta en efectivo de producto de tienda y combustible.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Productos: - Surtidor 1 - Carburante 1 - 5 litros - 10 € - Producto 1 - 2 unidades - 4 € Medio de pago: Efectivo

PCN-23	Venta en efectivo de varios productos
Resultado esperado	La venta se realiza correctamente y se muestra la pantalla principal con la lista del ticket virtual vacía.

Tabla 68. PCN-24 - Venta con tarjeta de combustible.

PCN-24	Venta con tarjeta de combustible
Descripción	Realizar una venta con tarjeta de combustible.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Productos: Surtidor 1 - Carburante 1 - 5 litros - 10 € Medio de pago: Tarjeta
Resultado esperado	La venta se realiza correctamente y se muestra la pantalla principal con la lista del ticket virtual vacía.

Tabla 69. PCN-25 - Venta con tarjeta de producto tienda.

PCN-25	Venta con tarjeta de producto tienda
Descripción	Realizar una venta con tarjeta de producto de tienda.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Productos: Producto 1 - 2 unidades - 4 € Medio de pago: Tarjeta
Resultado esperado	La venta se realiza correctamente y se muestra la pantalla principal con la lista del ticket virtual vacía.

Tabla 70. PCN-26 - Venta con tarjeta de varios productos.

PCN-26	Venta con tarjeta de varios productos
Descripción	Realizar una venta con tarjeta de producto de tienda y combustible.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Productos: - Surtidor 1 - Carburante 1 - 5 litros - 10 € - Producto 1 - 2 unidades - 4 € Medio de pago: Tarjeta
Resultado esperado	La venta se realiza correctamente y se muestra la pantalla principal con la lista del ticket virtual vacía.

6.5. Consulta de operaciones y stock

Tabla 71. PCN-27 - Consultar lista de ventas.

PCN-27	Consultar lista de ventas
Descripción	Consultar la lista de ventas realizadas en la estación.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Seleccionar “Consulta de operaciones”
Resultado esperado	Se muestra la lista de operaciones realizadas en la estación con todos sus datos correctamente.

Tabla 72. PCN-28 - Exportar ventas a Excel.

PCN-28	Exportar ventas a Excel
Descripción	Exportar a Excel la consulta de la lista de ventas realizadas en la estación.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Seleccionar “Exportar a Excel” en la consulta de operaciones
Resultado esperado	Se descarga en el dispositivo un archivo .csv con todos los datos de las ventas realizadas.

Tabla 73. PCN-29 - Consultar stock de productos.

PCN-29	Consultar stock de productos
Descripción	Consultar el stock de productos de la estación.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Seleccionar “Consulta de stock”
Resultado esperado	Se muestra la lista de productos dados de alta en la estación con sus datos de stock actuales.

Tabla 74. PCN-30 - Exportar consulta de stock a Excel.

PCN-30	Exportar consulta de stock a Excel
Descripción	Exportar a Excel la consulta de stock de productos de la estación.
Precondiciones	El usuario debe estar autenticado previamente en la aplicación.
Datos de entrada	Seleccionar “Exportar a Excel” en la consulta de stock
Resultado esperado	Se descarga en el dispositivo un archivo .csv con todos los datos de stock de los productos dados de alta en la estación.

Tabla 75. PCN-32 - Consultar ventas a devolver.

PCN-32	Consultar ventas a devolver
Descripción	Consultar las ventas que pueden ser devueltas en la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación.
Datos de entrada	Seleccionar “Devoluciones”
Resultado esperado	Se muestra la lista de ventas realizadas en la estación que aún no han sido devueltas.

Tabla 76. PCN-33 - Realizar una devolución.

PCN-33	Realizar una devolución
Descripción	Realizar la devolución de una venta realizada previamente en la estación.
Precondiciones	El usuario debe ser <i>usuario propietario</i> y estar autenticado previamente en la aplicación. Debe existir en la estación una venta que no haya sido ya devuelta.
Datos de entrada	Seleccionar “Devolver” en una operación de venta realizada previamente.
Resultado esperado	Se devuelve correctamente la operación y se actualiza la lista de operaciones a devolver.

7. Documentación

En esta sección se detallan los manuales de usuario y de despliegue. El primero detalla el funcionamiento de la aplicación desde la perspectiva del usuario, mientras que el segundo incluye los pasos técnicos para desplegar la aplicación.

7.1. Manual de usuario

En este apartado se detalla el manual de usuario, detallando, con una presentación clara, cada una de las opciones existentes para los usuarios dentro de la aplicación en cada una de sus pantallas. Este manual se va a dividir en secciones, en función de los flujos, tal y como está distribuido en el árbol de características que se presenta en este documento en el apartado 3.1.1.

7.1.1. Registro, inicio de sesión y recuperar contraseña

Registro: En primer lugar, se encontrará con la pantalla de inicio de sesión. Si ya tiene cuenta puede iniciar la sesión a través de esta pantalla, como se describe más adelante en este manual. Por el contrario, para registrarse en la aplicación deberá pulsar sobre la opción “Crear una nueva” que aparece en la parte de abajo del formulario de inicio de sesión.



Figura 57. Inicio de sesión - Ir a “Registro”.

Una vez hemos seleccionado esta opción nos aparecerá la pantalla de registro. En ella se detalla el formulario con los campos necesarios para dar de alta, y las opciones para darse de alta y volver a la pantalla de inicio de sesión.

Gestión de estaciones de servicio

TPVa es una aplicación que simplifica la gestión de estaciones de servicio y gasolineras. Permite controlar inventario, operaciones y empleados desde cualquier dispositivo con conexión a Internet, sin coste alguno.

Con una interfaz intuitiva y funcionalidades avanzadas, TPVa ofrece flexibilidad y eficiencia, liberando al usuario de la limitación geográfica y facilitando la toma de decisiones para impulsar el crecimiento del negocio.

TPVa

Nombre

Usuario

Contraseña

Tu contraseña debe tener entre 6 y 30 caracteres.

Registrarse

¿Ya tienes una cuenta creada?

Campos del formulario

Darse de alta en la aplicación

Volver a la pantalla de inicio

Figura 58. Registro de usuarios.

Los campos del formulario son los siguientes:

- Nombre: nombre de su usuario que aparecerá tanto en la pantalla principal como en la información de las ventas que realice.
- Usuario: correo electrónico del usuario asociado a la cuenta. En el caso de que se olvide la contraseña este será el correo donde se enviará la información para recuperarlo.
- Contraseña: contraseña con la que podrá acceder a la aplicación.

En cuanto a las dos opciones, la opción de “Registrarse” le dará de alta en la aplicación con los datos que haya rellenado en el formulario anterior. En el caso de que el “Usuario” ya esté dado de alta en la aplicación se le informará una vez seleccione esta opción. Una vez el alta del usuario se haya realizado correctamente, la aplicación le redirigirá a la pantalla principal de la aplicación, la cual se explica en los siguientes apartados de este manual.

Por último, si selecciona la opción “¿Ya tienes una cuenta creada?” le redirigirá a la pantalla de inicio de sesión, sin dar de alta al usuario.

Inicio de sesión: En cuanto al inicio de sesión, como ya hemos comentado anteriormente, será la pantalla que aparezca al iniciar la aplicación. En ella se encuentra el formulario para introducir los datos de usuario, la opción de iniciar sesión, recuperar contraseña y registrarse, que ya se ha explicado en el apartado anterior de este manual.



Figura 59. Inicio de sesión.

Los campos del formulario son los siguientes:

- Usuario: correo electrónico del usuario asociado a la cuenta que haya introducido en el formulario de registro. En el caso de que se olvide la contraseña este será el correo donde se enviará la información para recuperarlo.
- Contraseña: contraseña con la que podrá acceder a la aplicación y que haya introducido en el formulario de registro.

En cuanto a las dos opciones, la opción de “Entrar” iniciará la sesión de su usuario en la aplicación y le llevará a la pantalla principal, mostrando los datos de su usuario y su estación. En el caso de que los datos de usuario no sean correctos, se le mostrará un mensaje informativo al usuario informando de ello y no se le redirigirá a la pantalla principal.

En cuanto a la opción “¿Olvidaste tu contraseña?” redirigirá al usuario a la pantalla para recuperar la contraseña que se describe en el siguiente apartado, sin hacer el intento de inicio de sesión del usuario.

Recuperar contraseña: La pantalla para recuperar la contraseña del usuario es la más simple de la aplicación. Se encuentra el formulario con el campo para introducir el correo electrónico, y las opción para enviar el correo para recuperar contraseña y la opción para ir a la pantalla de registro de usuarios.

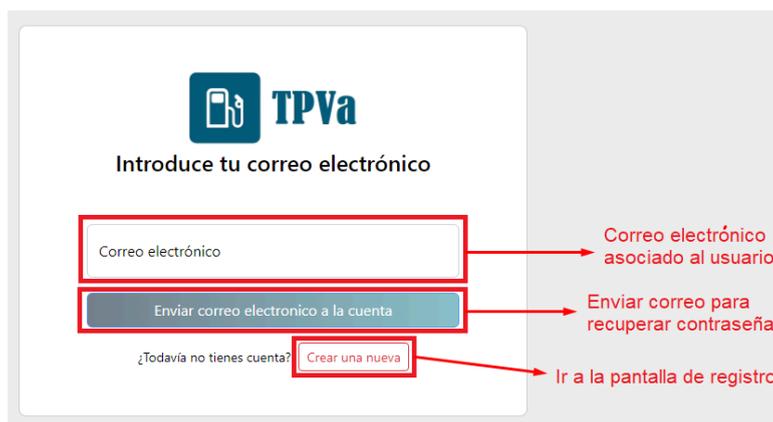


Figura 60. Recuperar contraseña.

El campo del formulario únicamente es el siguiente:

- Correo electrónico: correo electrónico del usuario asociado a la cuenta de la que se quiera recuperar la contraseña. Será el campo que haya introducido en el formulario de registro.

En cuanto a las dos opciones, la opción de “Enviar correo electrónico a la cuenta” enviará un correo electrónico a la cuenta indicada en el formulario. En dicho correo se le proporcionará un enlace que le mostrará un formulario para introducir la nueva contraseña con la que acceder a la aplicación. Una vez se envíe el correo, se le informará que el envío del correo se ha realizado correctamente y se le redirigirá de nuevo a la pantalla de inicio de sesión.

En cuanto a la opción “Crear una nueva” le redirigirá a la pantalla de registro de la aplicación, que ya se ha comentado en este manual.

7.1.2. Crear y editar productos

Pantalla principal: Una vez se ha registrado en la aplicación, se le mostrará la pantalla principal de la misma. Como puede ver en la Figura 61, dicha pantalla está dividida en tres partes. Por un lado, en la parte superior se encuentra la lista de surtidores, en la parte izquierda se encuentra el documento de productos seleccionados en el flujo de venta y por último, en la parte inferior derecha se encuentra el árbol de productos de tienda de la estación, junto con las distintas opciones que puede seleccionar el *usuario propietario* para la estación. En este apartado del documento nos centraremos sobre esta última parte de la pantalla principal.

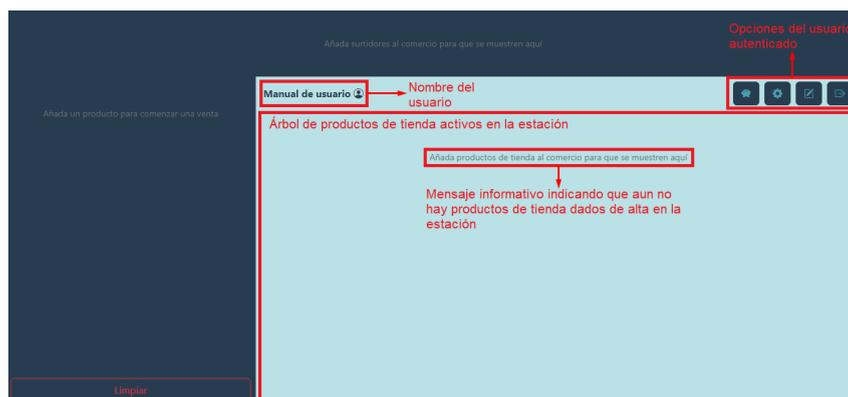


Figura 61. Pantalla principal sin datos de estación.

En esta subpantalla de la pantalla principal puede ver en la parte superior izquierda el nombre del usuario que ha introducido en el formulario de registro. También, en la parte superior derecha de la subpantalla puede ver las distintas opciones existentes en la aplicación para el *usuario propietario*, que son, de izquierda a derecha, las siguientes:

- Consulta de operaciones y stock: muestra las distintas opciones de consultas y devoluciones existentes en la aplicación. Esta opción está habilitada tanto para los *usuarios propietarios* de la estación como para los *usuarios dependientes*. Esta opción se detalla más adelante en el documento.
- Administrar estación: muestra las distintas opciones para administrar los datos de estación. Esta opción se detalla más adelante en el documento y solo estará habilitada para el *usuario propietario* de la estación.
- Editar productos: muestra la lista de tipos de productos posibles para la estación, es decir, “Tienda” y “Combustible”. Esta opción se detalla más adelante en el documento y solo estará habilitada para el *usuario propietario* de la estación.
- Cerrar sesión: termina la sesión activa del usuario en la aplicación y redirige a la pantalla de inicio de sesión. Esta opción estará habilitada para todos los usuarios.

Editar productos: En este apartado se describe la opción de “Editar productos”. Como se podrá ver en la Figura 62, en la parte superior izquierda de la subpantalla, aparece junto al nombre del usuario la acción que está realizando. Esto será similar para todas las opciones disponibles en la pantalla principal de la aplicación.

En cuanto a la opciones disponibles en la opción “Editar productos” están las opciones para listar los productos tienda y los carburantes ya dados de alta en la aplicación. Además, existe la opción para regresar a la pantalla principal de la aplicación.

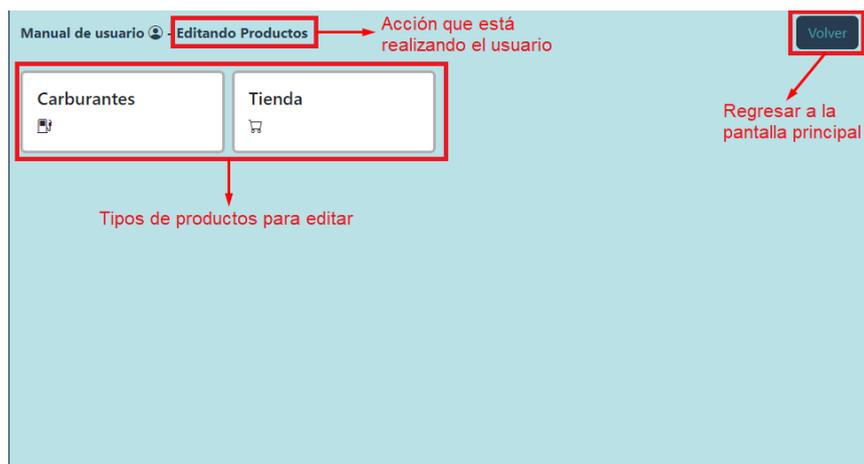


Figura 62. Subpantalla - Editar productos.

Editar carburantes: Si selecciona la opción de “Carburantes” se le mostrará simplemente la lista con los carburantes dados de alta en la aplicación. En este caso, tal y como se ve en la Figura 63, si se acaba de dar de alta en la aplicación no se muestra ningún carburante y simplemente se muestra la opción “Añadir nuevo producto”.

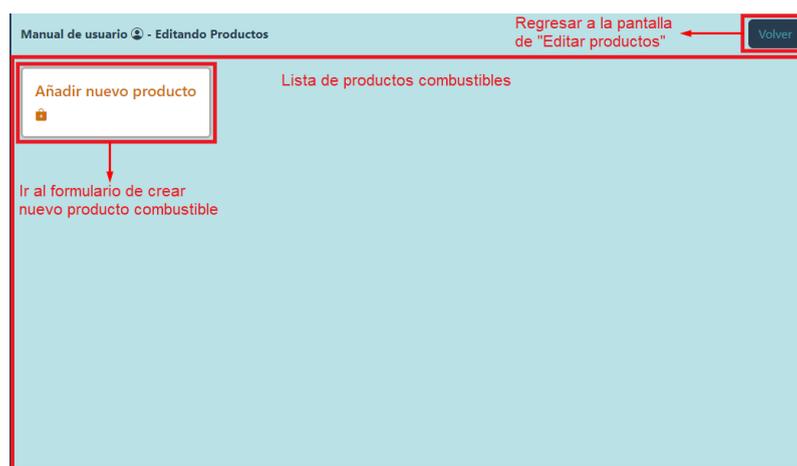


Figura 63. Subpantalla - Editar carburantes.

La opción “Añadir nuevo producto” le lleva a la subpantalla con el formulario para dar de alta y editar productos carburantes. En este caso, como ha seleccionado “Añadir nuevo producto” el formulario le aparece con todos los campos vacíos.

Manual de usuario - Editando Productos

Campos del formulario

Nombre de Producto

Precio Unitario
0

Stock
0

Cancelar

Guardar

Regresar a la lista de carburantes sin guardar

Guardar los nuevos datos del producto carburante

Figura 64. Subpantalla - Formulario de carburantes.

Los campos del formulario son los siguientes:

- Nombre del producto: nombre con el que aparecerá el producto tanto en la lista como en el ticket de venta y en la consulta de ventas.
- Precio unitario: precio asignado al producto.
- Stock: cantidad (en litros) con las que cuenta la estación inicialmente de este producto.

En cuanto a las opciones, si selecciona la opción “Cancelar” volverá a la lista de carburantes, sin guardar los cambios del producto. Si selecciona en “Guardar” la aplicación almacena los datos del producto y le redirige a la pantalla principal de la aplicación.

En este manual, para poder continuar con las pruebas, daremos de alta un carburante con nombre “Prueba carburante”, precio unitario “1” y stock “10000”. La subpantalla de lista de carburantes con el nuevo producto se muestra de la siguiente forma:

Manual de usuario - Editando Productos

Volver

Prueba carburante
1€

Añadir nuevo producto

✎

🗑️

Figura 65. Subpantalla - Editar carburantes.

Debajo de cada producto le aparecen dos opciones que son las siguientes (de izquierda a derecha): editar el producto y eliminar el producto.

- Editar el producto: muestra el formulario de productos carburantes con los datos del producto seleccionado, para que sean modificados. Si pulsa sobre el nombre del producto también le llevará a esta pantalla.
- Eliminar el producto: elimina el producto de tienda de la estación completamente. Esta opción sólo aparecerá si el carburante no está asociado a ningún surtidor. Más adelante se detalla cómo asociar un carburante a un surtidor.

Editar productos de tienda: Si selecciona la opción de “Tienda” en la pantalla de “Editar productos” se le muestra en vez de la lista únicamente de productos, el árbol de productos de tienda, debido a que dentro de este apartado se pueden crear carpetas en las que se pueden dar de alta productos, por lo que no es una lista en sí. En este caso, tal y como se ve en la Figura 66, si se acaba de dar de alta en la aplicación no se muestra ningún producto de tienda y simplemente se muestra la opción “Añadir nuevo producto”.



Figura 66. Subpantalla - Editar productos de tienda.

La opción “Añadir nuevo producto” le lleva a la subpantalla con el formulario para dar de alta y editar productos de tienda. En este caso, como ha seleccionado “Añadir nuevo producto” el formulario le aparece con todos los campos vacíos.



Figura 67. Subpantalla - Formulario de productos de tienda.

Los campos del formulario son los siguientes:

- Nombre del producto: nombre con el que aparecerá el producto tanto en el árbol de productos como en el ticket de venta y en la consulta de ventas.
- Precio unitario: precio asignado al producto.
- Stock: cantidad (en litros) con las que cuenta la estación inicialmente de este producto.
- Carpeta en la que se mostrará el artículo: subcarpeta en la que se mostrará el producto de la tienda dentro del árbol de productos. Si hemos dado de alta una carpeta anteriormente esta nos aparecerá en el desplegable de la siguiente forma:

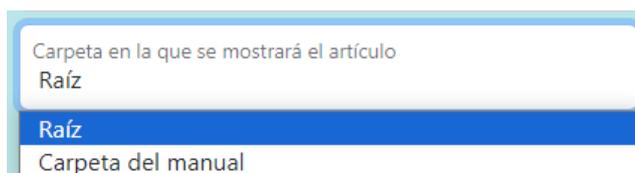


Figura 68. Desplegable de “Carpeta en la que se mostrará el producto”.

- Habilitado para la venta: indica si el producto estará habilitado para la venta o no.
- Carpeta de productos: indica si es una carpeta de productos o no. En el caso de que este este campo seleccionado no se le permitirá introducir ni precio ni stock.

En cuanto a las opciones, si selecciona la opción “Cancelar” volverá a la subpantalla con el árbol de productos tienda, sin guardar los cambios del producto. Si selecciona en “Guardar” la aplicación almacena los datos del producto y le redirigirá a la pantalla principal de la aplicación, donde se mostrarán los datos del producto tienda actualizados

En este manual, para poder continuar con las pruebas, daremos de alta un producto de tienda con nombre “Prueba tienda”, precio unitario “1”, stock “10000”, habilitado para la venta y en la carpeta raíz. La subpantalla para editar el árbol de productos tienda con el nuevo producto se muestra de la siguiente forma:



Figura 69. Subpantalla - Editar árbol de tienda.

Debajo de cada producto le aparecen tres opciones que son las siguientes (de izquierda a derecha): editar el producto, deshabilitar el producto y eliminar el producto.

- Editar el producto: muestra el formulario de productos de tienda con los datos del producto seleccionado, para que sean modificados. Si pulsa sobre el nombre del producto también le llevará a esta pantalla.
- Deshabilitar el producto: el producto quedará deshabilitado para la venta y ya no aparecerá en la pantalla principal dentro del árbol de productos, pero seguirá almacenado en los datos de estación y se mostrará en el árbol de productos de tienda de la opción “Editar productos”. Si el producto ya está deshabilitado en vez de aparecer esta opción, aparecerá la opción “Habilitar el producto”.
- Habilitar el producto: el producto quedará habilitado para la venta y ya aparecerá en la pantalla principal dentro del árbol de productos. Si el producto ya está habilitado en vez de aparecer esta opción, aparecerá la opción “Deshabilitar el producto”.
- Eliminar el producto: elimina el producto de tienda de la estación completamente.

7.1.3. Administrar estación

Para acceder a la pantalla de administración, seleccione opción con el engranaje (segunda opción empezando por la izquierda) en la lista de opciones que aparece en la parte superior derecha en la subpantalla de productos. Esta opción únicamente estará disponible para el *usuario propietario* de la estación.



Figura 70. Subpantalla - Seleccionar “Administrar estación”.

Una vez ha seleccionado la opción de “Administrar estación” se listarán en la subpantalla de productos las tres opciones disponibles para la administración de los datos propios de estación, tal y como se puede apreciar en la Figura 71.

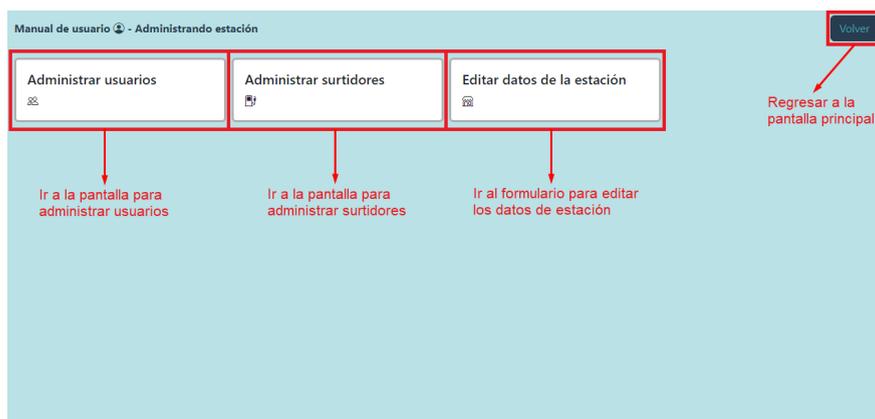


Figura 71. Subpantalla - Administrar estación.

Estas opciones que aparecen en dicha subpantalla son las siguientes:

- Administrar usuarios: muestra la lista de usuarios dados de alta en la estación, incluido el *usuario propietario*. Desde esta opción se pueden crear, editar y eliminar usuarios.
- Administrar surtidores: muestra la lista de surtidores dados de alta en la estación. Desde esta opción se pueden crear, editar y eliminar surtidores en la estación.
- Editar datos de la estación: muestra el formulario con los datos actuales del comercio listos para ser modificados.

Además de estas opciones, en esta subpantalla, al igual que en la de “Editar productos” se cuenta con la opción “Volver” que le permite regresar a la pantalla principal con el árbol de productos.

Administrar usuarios: Si se selecciona la opción “Administrar usuarios” sobre la subpantalla de “Administrar estación” se muestra la lista de usuarios de la estación. Si es la primera vez que se accede a esta opción únicamente se muestra el *usuario propietario*, junto a la opción para crear un nuevo usuario.

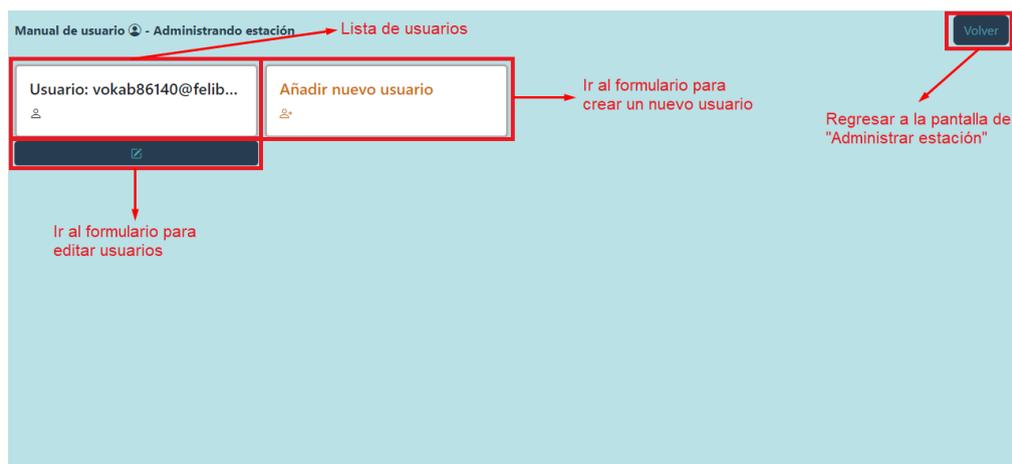


Figura 72. Subpantalla - Administrar usuarios.

Si se selecciona la opción de “Añadir nuevo usuario” se redirigirá a la subpantalla con el formulario vacío para dar de alta nuevos usuarios.



Figura 73. Subpantalla - Formulario “Añadir nuevo usuario”.

Este formulario es similar al de la pantalla de “Registro” y cuenta con los mismos campos, que son los siguientes:

- Nombre: nombre del nuevo *usuario dependiente* que aparecerá tanto en la pantalla principal como en la información de las ventas que realice.
- Usuario: correo electrónico del *usuario dependiente* asociado a la cuenta. En el caso de que se olvide la contraseña este será el correo donde se enviará la información para recuperarlo.
- Contraseña: contraseña con la que el *usuario dependiente* podrá acceder a la aplicación.

En cuanto a las opciones existentes en el formulario son, la opción de “Cancelar” que redirige a la pantalla con la lista de usuarios sin dar de alta un nuevo usuario, y la opción de “Guardar” que dará de alta el nuevo usuario en la estación y redirigirá a la subpantalla con la lista de usuarios con los datos actualizados.

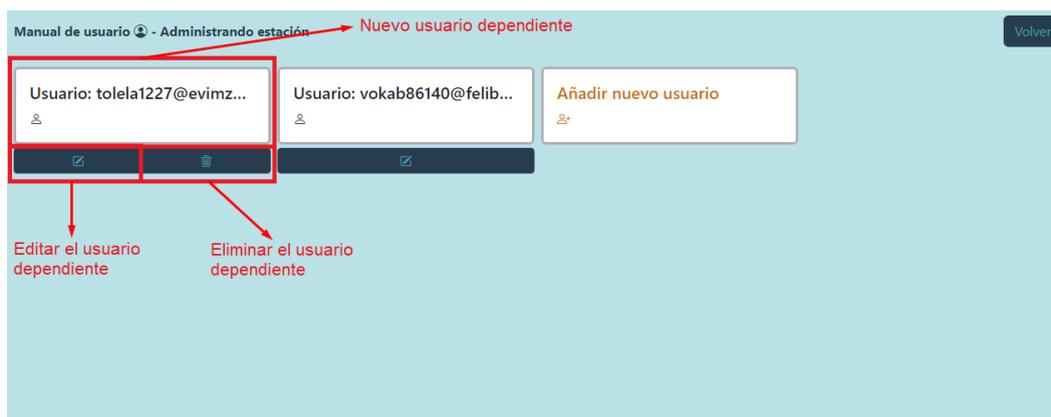


Figura 74. Subpantalla - Administrar usuarios.

Una vez se haya dado de alta un nuevo usuario, aparecerá en la lista junto con dos opciones nuevas. La opción de editar el usuario y la opción de eliminar el usuario.

Si se selecciona la opción de eliminar el usuario este se eliminará de la aplicación y dejará de aparecer en la lista de usuarios. Además, el *usuario dependiente* ya no tendrá acceso a la aplicación. Esta opción únicamente estará disponible para ejecutarla sobre los *usuarios dependientes*.

En cambio, si se selecciona sobre el *usuario dependiente* o sobre la opción para editar del usuario dependiente, se mostrará el formulario para editar el usuario tal y como se puede ver en la Figura 75.

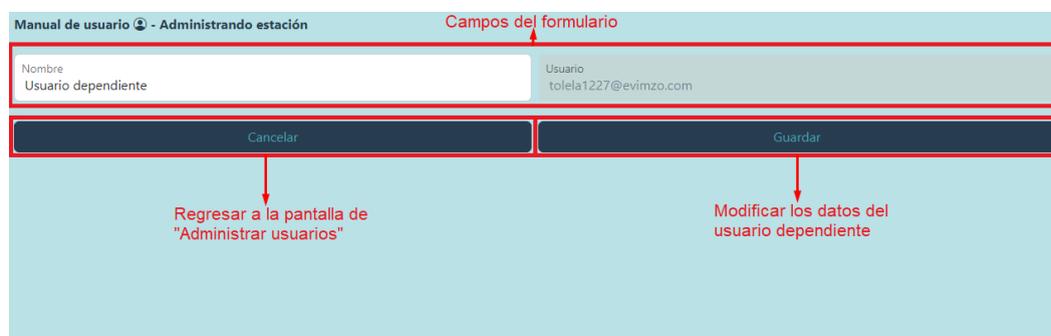


Figura 75. Subpantalla - Formulario “Modificar *usuario dependiente*”.

Para este formulario existen dos campos, “Nombre” y “Usuario” pero únicamente el campo “Nombre” es modificable. Si selecciona la opción “Cancelar” regresará a la lista de usuarios sin modificar ningún dato. En cambio, si selecciona la opción “Guardar” se actualizarán los datos y se le redirigirá a la lista de usuarios mostrando los nuevos datos.

Para modificar los datos del *usuario propietario* debe seleccionarse la opción editar del *usuario propietario* o pulsar sobre el nombre de este usuario. Una vez se seleccione esta opción se mostrará el siguiente formulario:

Figura 76. Subpantalla - Formulario “Modificar *usuario propietario*”.

Este formulario cuenta con tres campos, que son “Nombre”, “Usuario” y “Contraseña”. En un primer momento solo se permite modificar el nombre. Si se selecciona la opción “Modificar correo” se permitirá cambiar el usuario una vez introduzca la contraseña actual del usuario.

Para actualizar los datos, se deberá seleccionar la opción “Guardar” que además redirigirá a la pantalla con la lista de usuarios actualizada. Si se selecciona la opción “Cancelar” llevará a la pantalla con la lista de usuarios sin actualizar ninguno de los datos.

Para modificar la contraseña se debe seleccionar la opción “Cambiar contraseña” la cual mostrará el siguiente formulario:

Figura 77. Subpantalla - Formulario “Modificar contraseña”.

Este formulario cuenta con dos campos, “Contraseña actual” en la que se debe introducir la contraseña actual del usuario, y “Nueva contraseña” en la que se debe introducir la nueva contraseña que se desea para el usuario.

En cuanto a las opciones, la opción “Cancelar” es similar al formulario anterior. Las opciones “Modificar correo” y “Cancelar cambiar contraseña” redirigirán al formulario anterior, pero la opción “Modificar correo”, además, permitirá modificar el correo directamente.

En cuanto a la opción “Guardar”, actualizará la contraseña del usuario y llevará a la subpantalla inicial para actualizar los datos de usuario.

Administrar surtidores: Si se selecciona la opción “Administrar surtidores” sobre la subpantalla de “Administrar estación” se muestra la lista de surtidores de la estación. Si es la

primera vez que se accede a esta opción únicamente se muestra la opción para añadir un nuevo surtidor.



Figura 78. Subpantalla - Administrar surtidores.

Si se selecciona la opción de “Añadir nuevo surtidor” se redirigirá a la subpantalla con el formulario vacío para dar de alta un nuevo surtidor.

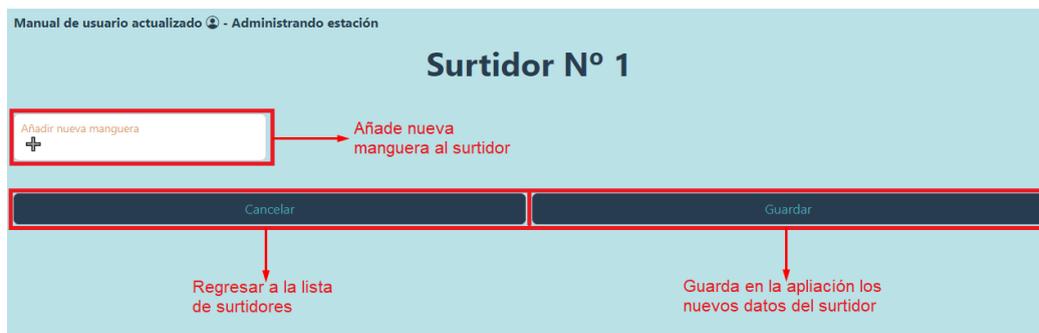


Figura 79. Subpantalla - Formulario “Añadir nuevo surtidor”.

Este formulario cuenta simplemente con la opción “Añadir nueva manguera”. Esta opción es un desplegable, el cual permite seleccionar el carburante de la manguera. Se permite añadir nuevas mangueras hasta un máximo de 20 por surtidor. Si se desea eliminar la manguera, seleccione la opción “Ninguno” dentro del desplegable.

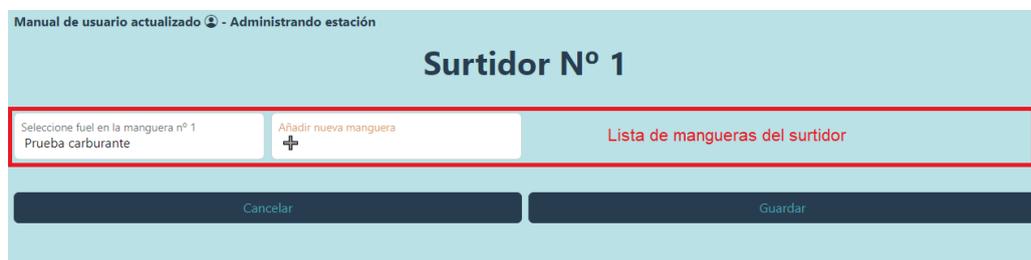


Figura 80. Subpantalla - Formulario “Añadir nuevo surtidor” con manguera.

El número del surtidor es automático y se asigna en función de los surtidores que haya ya creados en la estación con anterioridad, es decir, si ya hubiese dos surtidores cuando vaya a crear uno nuevo le aparecerá el nombre “Surtidor Nº 3”.

En cuanto a las opciones existentes en el formulario son, la opción de “Cancelar” que le redirigirá a la pantalla con la lista de surtidores sin dar de alta un nuevo surtidor, y la opción de “Guardar” que dará de alta el nuevo surtidor en la estación y le redirigirá a la subpantalla con la lista de surtidores con los datos actualizados.

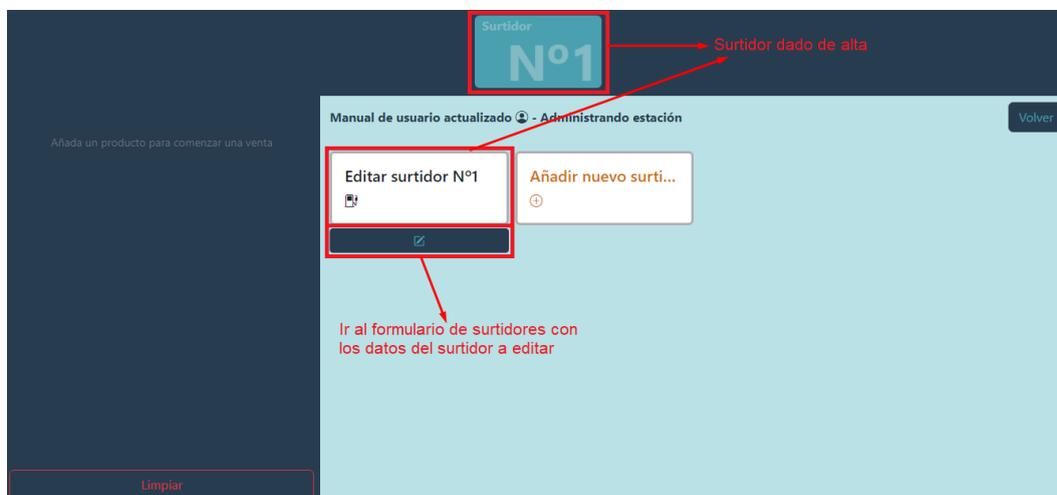


Figura 81. Subpantalla - Administrar surtidores.

Una vez dado de alta un nuevo surtidor, éste aparecerá tanto en la lista de surtidores de de la subpantalla “Administrar surtidor” como en la subpantalla superior de surtidores, indicando el número de surtidor que es.

En el formulario de alta de surtidores, es posible crear un surtidor sin mangueras. En este caso, el surtidor no aparecerá en la lista superior. En la lista de surtidores de la opción “Administrar surtidores”, debajo de ese surtidor, le aparecerá la opción de “Eliminar” además de la opción de “Editar” que aparece también en los surtidores que sí tienen mangueras. Si selecciona la opción “Eliminar”, el surtidor se eliminará de la estación.

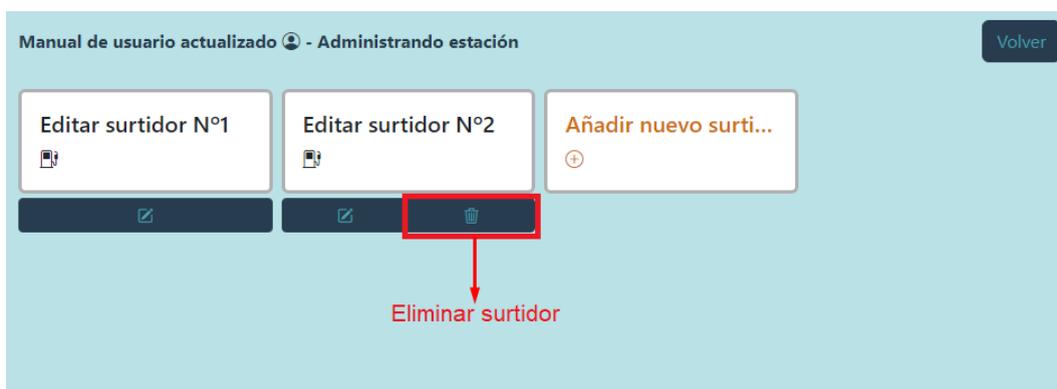


Figura 82. Subpantalla - Administrar surtidores.

En el caso de que seleccione la opción “Editar” o seleccione el nombre del surtidor, se le mostrará el formulario de alta con las mangueras existentes para el surtidor seleccionado y la opción de modificarlas.

Editar estación: Si selecciona la opción “Editar estación” sobre la subpantalla de “Administrar estación” se le muestra el formulario de datos de estación con los datos actuales de la estación.

Manual de usuario actualizado - Administrando estación

Campos del formulario

Nombre
Estación de Manual de usuario

Dirección
Calle de la Piruleta 1

Cancelar

Guardar

Regresar a la pantalla de "Administrar estación"

Actualizar los datos de estación

Figura 83. Subpantalla - Formulario estación.

Por defecto, cuando se crea la estación, se le asignará como nombre “Estación de <<nombre del *usuario propietario*>>” y como dirección “Calle de la Piruleta 1”. Estos son los dos campos modificables de la estación, no pueden estar vacíos y se pueden modificar en cualquier momento (siempre a través del *usuario propietario*).

En cuanto a las opciones existentes en el formulario son, la opción de “Cancelar” que le redirigirá a la pantalla de “Administrar estación” sin actualizar los datos, y la opción de “Guardar” que actualizará los datos de la estación en la aplicación. De momento estos datos son simplemente internos para la estación, pero en futuras mejoras serán los datos que aparecerán en el ticket de compra.

7.1.4. Realizar ventas

Para mostrar el flujo de venta, a parte de usar los productos y surtidores creados en el manual en los apartados anteriores, se va a crear un producto dentro de la carpeta “Carpeta del manual” para poder navegar por el árbol de productos. Del mismo modo, se van a crear surtidores nuevos hasta que sobrepasen el tamaño de la pantalla y mostrar cómo navegar por la lista de surtidores. De este modo la pantalla principal para la estación nos quedaría de la siguiente forma:

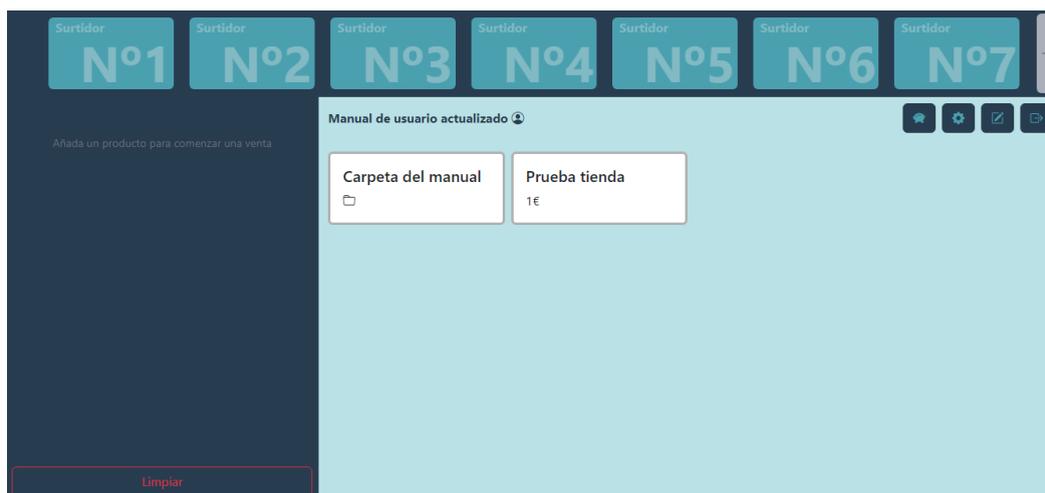


Figura 84. Pantalla principal.

Como ya se ha mostrado anteriormente en este manual, los productos se dividen en dos tipos, carburantes y productos de tienda. Para realizar la venta se muestra en primer lugar como navegar y seleccionar los productos de tienda.

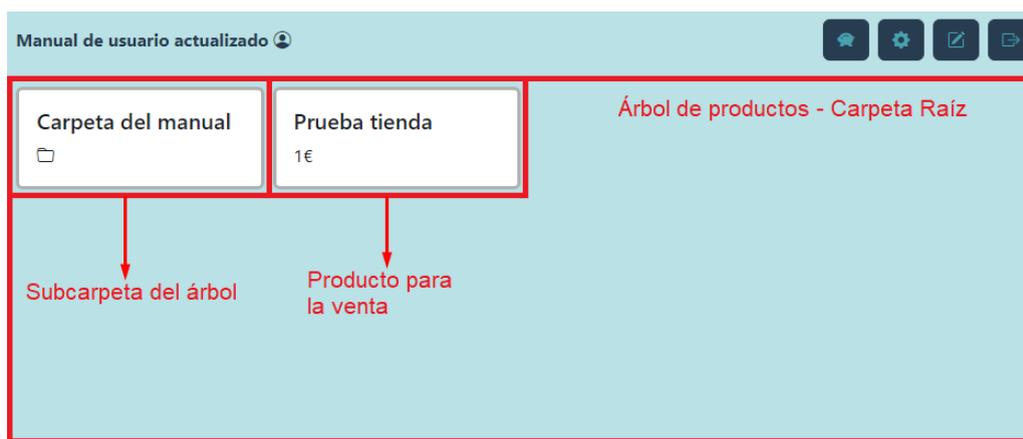


Figura 85. Pantalla principal.

En la pantalla principal se muestra el árbol de productos dados de alta para la estación. En nuestro caso aparece una carpeta llamada “Carpeta del manual” y un productos de tienda vendible llamado “Producto tienda”. Ambos están en la carpeta llamada “Raíz” por lo que aparecerán en la primera pantalla del árbol de productos. Si quiere añadir el producto “Producto tienda” al documento de compra, debe seleccionar dicho producto. Una vez seleccionado aparecerá en la lista de productos para vender, en la subpantalla izquierda de la aplicación.

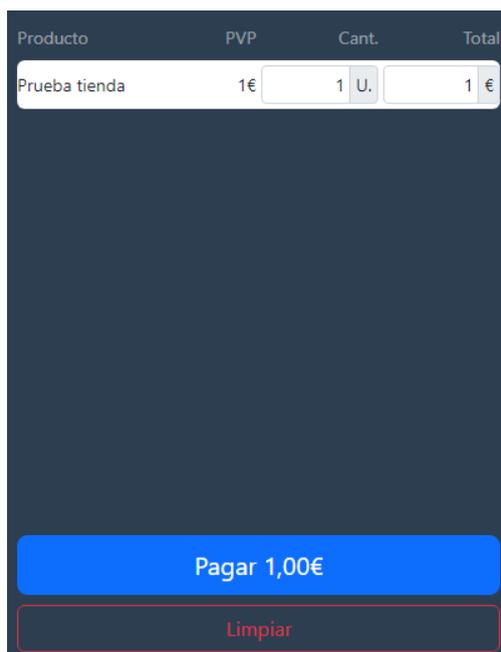


Figura 86. Subpantalla - Documento de venta.

Como se puede ver, se añade una nueva fila con los datos del producto dados de alta (de izquierda a derecha: nombre, precio unitario del producto, cantidad y precio total). Dentro de esta línea de producto, se puede modificar a mano la cantidad, únicamente con números enteros. También puede añadir cantidad de uno en uno volviendo a seleccionar el producto en el árbol de productos.

En cuanto a las opciones que se muestran son “Pagar” y “Limpiar”. En cuanto a la opción de “Pagar” se explicará más adelante en este apartado del manual. La opción “Limpiar” eliminará todos los productos de la lista de productos a vender.

Para añadir un nuevo producto a la lista de productos a vender, si este se encuentra en una subcarpeta debemos navegar por el árbol de productos vendibles. En este caso, para seleccionar el producto “Producto en subcarpeta” debemos seleccionar la carpeta “Carpeta del manual”. En este caso esta es la pantalla que se muestra:



Figura 87. Subpantalla - Subcarpeta del árbol de productos vendibles.

Como se ve en la Figura 87, cuando se selecciona una subcarpeta del árbol se puede ver la lista de productos vendibles que están dados de alta en esta subcarpeta. En este caso, únicamente se muestra el producto “Producto en subcarpeta”. También se muestra la opción “Volver”, que permite regresar a la carpeta superior del árbol de productos, en este caso, a la carpeta raíz.

Para añadir el producto “Producto en subcarpeta” a la lista de productos para vender, al igual que con el producto anterior, se selecciona el producto.

Producto	PVP	Cant.	Total
Prueba tienda	1€	5 U.	5 €
Producto en subcarpeta	2,50€	1 U.	2,5 €

Pagar 7,50€

Limpia

Figura 88. Subpantalla - Documento de venta.

Una vez seleccionado, se añade una nueva fila con los datos del producto, al igual que el producto anterior, y se sumará el precio del producto al precio a pagar. También es posible modificar la cantidad del producto similar a como se ha mostrado anteriormente en el manual.

Para añadir carburantes a la venta, primero hay que seleccionar el surtidor. Para ello es necesario navegar por la lista de surtidores que aparece en la subpantalla superior de la pantalla principal de la aplicación.



Figura 89. Subpantalla - Lista de surtidores.

Como se puede ver, cada surtidor muestra el número que tiene asignado. Además, en la lista se muestra la opción con la flecha hacia la derecha. Esta opción desplaza la lista hacia la izquierda, mostrando el siguiente surtidor (surtidor 8) y ocultando el primer surtidor visible de la lista (surtidor 1). Para seleccionar el surtidor 10 debemos seleccionar tres veces la opción con la flecha hacia la derecha.



Figura 90. Subpantalla - Lista de surtidores seleccionados.

Como se puede ver en la Figura 90, una vez se ha llegado al final de la lista de surtidores la opción con la flecha hacia la derecha desaparece, puesto que ya no hay más surtidores a la derecha en la lista. Del mismo modo, en cuanto hay un surtidor a la izquierda, se muestra la opción con la flecha hacia la izquierda, la cual desplaza la lista de surtidores hacia la derecha.

Una vez seleccionado el surtidor, este se muestra en la lista de productos a vender.

Producto	PVP	Cant.	Total
Prueba tienda	1€	5 U.	5 €
Producto en subcarpeta	2,50€	1 U.	2,5 €

Surtidor Nº10	PVP	Cant.	Total
Carburante			

[Limpiar](#)

Figura 91. Subpantalla - Documento de venta.

Cuando se añade el surtidor, la opción de pagar desaparece, ya que aún no se ha seleccionado el carburante. Para seleccionar el carburante existe el desplegable dentro de la fila que se ha añadido del surtidor. En este caso, solo existe un carburante y el documento quedaría de la siguiente forma una vez se haya introducido la cantidad o el total a pagar (en los carburantes sí se puede modificar el total a pagar. En este caso se recalcula la cantidad):

Producto	PVP	Cant.	Total
Prueba tienda	1€	5 U.	5 €
Producto en subcarpeta	2,50€	1 U.	2,5 €
Surtidor Nº10			
Carburante ▼	5€	5 L.	25 €

Pagar 32,50€

Limpiar

Figura 92. Subpantalla - Documento de venta.

Si se selecciona en este caso la opción “Pagar”, se muestran dos opciones nuevas “Efectivo” y “Tarjeta”, que le permite seleccionar el medio de pago de la venta.

Efectivo  Tarjeta 

Limpiar

Figura 93. Subpantalla - Opciones de pago.

Una vez se seleccione el medio de pago, aparecerá la opción para confirmar el pago y finalizar la venta:

Confirmar pago 32,50€ en efectivo

Cancelar

Figura 94. Subpantalla - Confirmar el pago.

Una vez se confirme el pago, la venta habrá finalizado correctamente, se guardarán los datos de la misma en la aplicación y se mostrará la siguiente pantalla indicándolo.

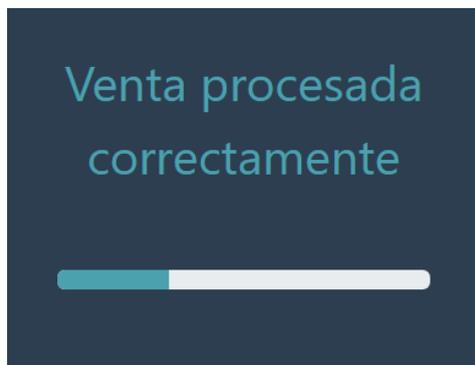


Figura 95. Subpantalla - Venta procesada.

7.1.5. Consulta de operaciones y stock

Para acceder a la pantalla de “Consulta de operaciones y stock”, seleccione opción con la hucha (primera opción empezando por la izquierda) en la lista de opciones que aparece en la parte superior derecha en la subpantalla de productos.

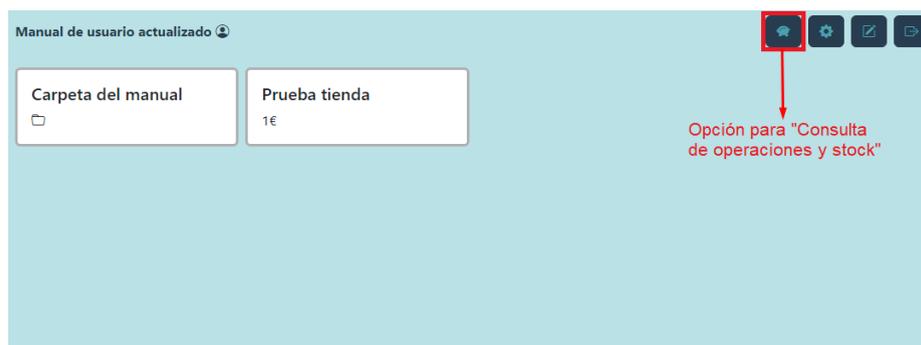


Figura 96. Subpantalla - Seleccionar “Consulta de operaciones y stock”.

Una vez ha seleccionado la opción de “Consulta de operaciones y stock” se listarán en la subpantalla de productos las tres opciones disponibles para la consulta de los datos propios de estación, tal y como se puede apreciar en la Figura 97.



Figura 97. Subpantalla - Consulta de operaciones y stock.

Estas opciones que aparecen en dicha subpantalla son las siguientes:

- Consulta de operaciones: muestra la lista de operaciones realizadas en la estación.
- Consulta de stock: muestra la lista de productos dados de alta en la estación, permitiendo actualizar su stock.
- Devoluciones: muestra la lista de ventas realizadas en la estación que aún no han sido devueltas. Esta opción únicamente estará disponible para el *usuario propietario*.

Además de estas opciones, en esta subpantalla, al igual que en la de “Editar productos” y “Administrar estación” se cuenta con la opción “Volver” que le permite regresar a la pantalla principal con el árbol de productos.

Consulta de operaciones: Si se selecciona la opción “Consulta de operaciones” sobre la subpantalla de “Consulta de operaciones y stock” se muestra la lista de operaciones realizadas. Si se accede a esta opción antes de realizar ninguna operación, esta lista aparecerá vacía.

Fecha	Usuario	Medio de pago	Devuelta	Total
14/04/2024 19:25:17	Manual de usuario actualizado	Efectivo	No	32.5€

Figura 98. Subpantalla - Consulta de operaciones.

Dentro de esta subpantalla, existen dos opciones, “Volver” que redirige a la pantalla de “Consulta de operaciones y stock” y la opción “Exportar a Excel” que exporta a .csv toda la información que se muestra relativa a las operaciones.

Consulta de stock: Si se selecciona la opción “Consulta de stock” sobre la subpantalla de “Consulta de operaciones y stock” se muestra la lista de productos dados de alta en la estación. Si se accede a esta opción antes de dar de alta ningún producto, esta lista aparecerá vacía.



Figura 99. Subpantalla - Consulta de stock.

Las opciones superiores son exactamente las mismas que en la pantalla anterior. En cuanto a la opción que aparece para cada producto es “Actualizar”, que permite modificar el stock del producto seleccionado.



Figura 100. Subpantalla - Actualizar stock.

Una vez se seleccione la opción “Actualizar”, se mostrarán las opciones “Confirmar” para actualizar y guardar el cambio de stock, y “Cancelar” para cancelar el cambio de stock del producto.

Devoluciones: Si se selecciona la opción “Devoluciones” sobre la subpantalla de “Consulta de operaciones y stock” se muestra la lista de ventas realizadas que aún no han sido devueltas. Si se accede a esta opción antes de realizar ninguna operación, esta lista aparecerá vacía.



Figura 101. Subpantalla - Devoluciones.

La opción superior de “Volver” es exactamente igual a la existente con el mismo nombre en las dos pantallas anteriores. En cuanto a la opción que aparece para cada operación es “Devolver”. Esta opción le permite devolver la operación seleccionada.

Fecha	Usuario	Medio de pago	Total	Acción
14/04/2024 19:25:17	Manual de usuario actualizado	Efectivo	32.5€	<input type="button" value="Cancelar"/> <input type="button" value="Confirmar devolución"/>

Figura 102. Subpantalla - Confirmar devolución.

Una vez seleccionada la opción, se mostrarán dos opciones nuevas, “Cancelar” que cancela la opción de devolver la operación, y “Confirmar devolución” que devuelve la operación y actualiza los datos en la estación.

Fecha	Usuario	Medio de pago	Total	Acción
-------	---------	---------------	-------	--------

Figura 103. Subpantalla - Devoluciones.

Una vez devuelta, la operación desaparece de la lista de la pantalla de “Devoluciones” y en la pantalla de “Consulta de operaciones” aparecerá como “Devuelta”.

Fecha	Usuario	Medio de pago	Devuelta	Total
14/04/2024 19:25:17	Manual de usuario actualizado	Efectivo	Sí	32.5€

Figura 104. Subpantalla - Consulta de operaciones devueltas.

7.2. Manual de despliegue

El despliegue de la aplicación, al ser esencialmente *serverless*, no requiere de la instalación de infraestructura software en máquinas *on-premise* de forma manual. En su lugar, se ha optado por utilizar GitHub Actions y Cloudflare Pages & Workers para automatizar este proceso, facilitando el despliegue continuo propio de las metodologías ágiles.

Para configurar el despliegue, se deben declarar una serie de variables de entorno tanto en el repositorio de GitHub como en el panel de administración de Cloudflare. Para el primer caso, necesitamos proveer a la instancia de GitHub Actions con las credenciales de acceso de Cloudflare para que pueda comunicarse con este servicio y desplegar la nueva versión de la aplicación, tanto en Cloudflare Pages (contenido estático) como en Cloudflare Workers

(funciones *serverless*). La Figura 105 muestra el extracto de código del *workflow* de GitHub Actions que se utiliza para el despliegue.

```
# Deploy to Cloudflare Pages
- name: Deploy to Cloudflare Pages
  if: ${{ success() && github.ref == 'refs/heads/master' }}
  uses: cloudflare/pages-action@v1
  with:
    apiToken: ${{ secrets.CLOUDFLARE_API_TOKEN }}
    accountId: ${{ secrets.CLOUDFLARE_ACCOUNT_ID }}
    projectName: ${{ secrets.CLOUDFLARE_PROJECT_NAME }}
    branch: main
    directory: ./dist
    githubToken: ${{ secrets.GITHUB_TOKEN }}
    wranglerVersion: '3'
```

Figura 105. Extracto de la implementación de GitHub Actions.

Para obtener las variables necesarias desde Cloudflare, se deben seguir los pasos mencionados en el manual del Action de despliegue oficial [6]:

1. Iniciar sesión en el panel de control de Cloudflare.
2. Seleccionar “My Profile” en el menú desplegable en la parte superior derecha.
3. Seleccionar “API tokens” y “Create token”.
4. En “Custom Token”, seleccionar “Get started”.
5. Asignar un nombre al API token en el campo “Name”.
6. En “Permissions”, seleccionar “Account”, “Cloudflare Pages” y “Edit”:
7. Seleccionar “Continue to summary” y “Create Token”.

Una vez obtenidas estas variables, se deben añadir al repositorio de GitHub (ver Figura 106):

1. Abrir la URL del repositorio de GitHub.
2. Pulsa en la pestaña “Settings” del proyecto.
3. Seleccionar “Secrets and Variables” y “Actions”.
4. En “Repository Secret”, pulsar en “New repository secret” para crear tres nuevas variables con los nombres CLOUDFLARE_API_TOKEN, CLOUDFLARE_ACCOUNT_ID y CLOUDFLARE_PROJECT_NAME.

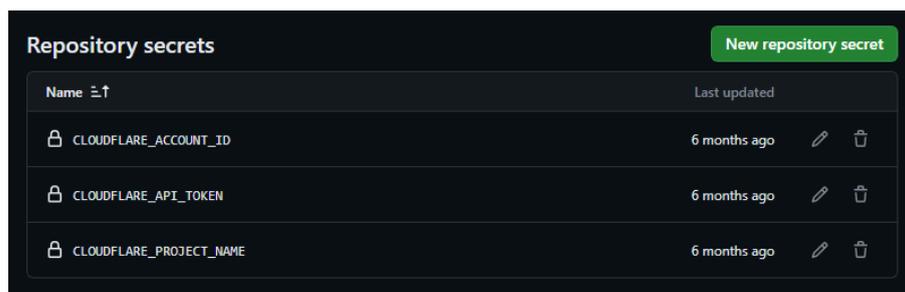


Figura 106. Captura de pantalla del menú “Repository Secrets” de GitHub Actions.

Por último, debemos configurar las variables de entorno necesarias para la autenticación con Firebase en los scripts de Cloudflare Workers. Este valor lo podemos obtener desde la consola de Firebase:

1. Abrir el proyecto de Firebase.
2. Seleccionar la pestaña “Service accounts” y “Firebase Admin SDK”.
3. Pulsar en “Generate new private key” y guardar el fichero JSON con el valor.

Por último, copiamos este valor al proyecto de Cloudflare:

1. Iniciar sesión en el panel de control de Cloudflare.
2. Desde el menú lateral, seleccionar “Workers & Pages” y el nombre del proyecto.
3. Pulsar en la pestaña “Settings” y “Environment variables”.
4. En el botón “Edit variables” de “Production”, crear una nueva variable con el nombre de `GOOGLE_CREDENTIALS` y el valor previamente obtenido en Firebase.

Tras *pushear* un nuevo commit al repositorio de GitHub, el despliegue de la aplicación se realizará de forma automática.

8. Conclusiones

El objetivo principal de este proyecto era desarrollar una aplicación plenamente funcional, lista para su implementación en una estación de servicio real, mientras se abordaban las complejidades tecnológicas inherentes al desarrollo *cloud* de aplicaciones. Pese a estas complejidades, se han alcanzado todos los objetivos del proyecto.

El proceso de desarrollo, tanto de la solución completa como del Trabajo de Fin de Grado, representó un desafío personal importante. Aunque utilicé principalmente tecnologías que ya conocía, como el framework Angular, la integración de plataformas en la nube nuevas para mí, como Firebase y Cloudflare, requirió una adaptación y aprendizaje amplio y continuo en el tiempo.

Además, aprender a equilibrar mi jornada laboral con el desarrollo de la aplicación fue crucial. Esto me llevó a enfrentar jornadas laborales interminables de trabajo, pero al final, han dado sus frutos. Este proyecto no solo me permitió ampliar mis habilidades técnicas, sino que también me enseñó la importancia de ser perseverante, constante y gestionar eficientemente mi tiempo. Todo ello me ha permitido crecer tanto a nivel técnico como personal.

8.1. Futuras mejoras

Para la implementación de la seguridad en la solución actual, se han utilizado funciones *serverless* que autentican y verifican al usuario con cada petición. A futuro, sería interesante sustituir algunas de estas funciones (aquellas en las que sea posible) con peticiones directas del cliente a Firestore Database. Esto sería posible gracias a las reglas de seguridad de este producto,³ que implementan todos los checks de seguridad automáticamente en base a una serie de parámetros que son establecidos en la configuración del proyecto de Firebase.

En cuanto a las funcionalidades, se prevén varias mejoras a medio y largo plazo. La primera es añadir notificaciones al usuario cuando el *stock* de los productos sea inferior a una cantidad que fije el usuario para cada uno de los productos. De este modo, el mantenimiento de productos y sus pedidos se podrían hacer de forma más óptima. Por otro lado, se prevé añadir los tickets de venta y devolución para entregar al usuario, con datos detallados de cada una de las operaciones e indicando también los datos de la aplicación. Y por último, como mejora principal, si la aplicación tiene éxito, se realizará integración de la aplicación web con

³ Ver <https://firebase.google.com/docs/firestore/security/get-started>

una o varias aplicaciones de medios de pago, que se comuniquen directamente con el lector de tarjetas y el centro autorizador de las mismas y de este modo tener una información del pago más extensa y real.

Por último, en un futuro se añadirá la internacionalización de la aplicación. Actualmente, únicamente está disponible en español, pero se prevé añadir la opción en inglés en función del país en el que se ejecute nuestra aplicación web.

Referencias

- [1] 4GL S.L., “TPV Para Gasolineras Y Estaciones De Servicio - 4GL,” 2023. [Online] <https://4glsp.com/productos/tpv/>
- [2] Amazon.com, Inc., “Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%,” 2023. [Online] <https://www.primevideotech.com/video-streaming/scaling-up-the-prime-video-audio-video-monitoring-service-and-reducing-costs-by-90>
- [3] Asociación Española de Operadores de Productos Petrolíferos, “Memoria 2022,” 2022. [Online] <https://www.aop.es/wp-content/uploads/2023/07/aop-memoria-2022.pdf>
- [4] Cloudflare, Inc., “Cloudflare Pages,” 2024. [Online] <https://pages.cloudflare.com/>
- [5] Cloudflare, Inc., “Cloudflare Workers®,” 2024. [Online] <https://workers.cloudflare.com/>
- [6] Cloudflare, Inc., “Cloudflare Pages GitHub Action,” 2024. [Online] <https://github.com/cloudflare/pages-action>
- [7] D. Jose, “MVVM Architecture in Angular: Guide with Code Examples,” 2023. [Online] <https://deepakjosecodes.com/mvvm-architecture-in-angular-guide-with-code-examples/>
- [8] Google, LLC., “Firebase Authentication,” 2024. [Online] <https://firebase.google.com/docs/auth>
- [9] Google, LLC., “Cloud Firestore,” 2024. [Online] <https://firebase.google.com/docs/firestore>
- [10] LS Retail ehf., “Gas station software system & convenience store pos: Forecourt software,” 2023. [Online] <https://www.l retail.com/industries/forecourt-c-store>
- [11] NCR Payment Solutions, LLC, “Convenience and Fuel Retail Solutions | NCR,” 2023. [Online] <https://www.ncr.com/retail/convenience-fuel-retail>
- [12] Netlify, Inc., “For fast and secure sites | Jamstack,” 2024. [Online] <https://jamstack.org>
- [13] Proconsi, S.L., “Software TPV para Gasolineras | cPista,” 2023. [Online] <https://www.proconsi.com/tpv-gasolinera-software-punto-de-venta-para-la-pista-cpista>
- [14] Talent.com, “Salario en España 2023,” [Online] <https://es.talent.com/salary>
- [15] VeriFone, Inc., “Site Controllers | Verifone,” 2023. [Online] <https://www.verifone.com/en/us/site-controllers>

Índice de tablas

Tabla 1. Comparativa entre las distintas soluciones de TPV más comunes en el mercado.	15
Tabla 2. Criterios de evaluación de entradas externas.	24
Tabla 3. Criterios de evaluación de salidas externas.	24
Tabla 4. Criterios de evaluación de consultas.	25
Tabla 5. Criterios de evaluación de ficheros lógicos internos.	25
Tabla 6. Puntos de función sin ajustar.	25
Tabla 7. Evaluación de Factores de Ajuste.	26
Tabla 8. Presupuesto Hardware.	34
Tabla 9. Presupuesto Software.	35
Tabla 10. Presupuesto de personal.	35
Tabla 11. Presupuesto total del proyecto.	36
Tabla 12. Relación entre reglas de negocio, actores y características.	40
Tabla 13. Caso de uso “Registrarse”.	46
Tabla 14. Caso de uso “Iniciar sesión”.	47
Tabla 15. Caso de uso “Cerrar sesión”.	48
Tabla 16. Caso de uso “Recuperar contraseña”.	49
Tabla 17. Caso de uso “Realizar venta”.	50
Tabla 18. Caso de uso “Navegar por el árbol de productos tienda”.	51
Tabla 19. Caso de uso “Navegar por la lista de surtidores”.	52
Tabla 20. Caso de uso “Añadir producto tienda”.	53
Tabla 21. Caso de uso “Modificar producto tienda”.	54
Tabla 22. Caso de uso “Eliminar producto tienda”.	56
Tabla 23. Caso de uso “Añadir producto combustible”.	57
Tabla 24. Caso de uso “Modificar producto combustible”.	58
Tabla 25. Caso de uso “Eliminar producto combustible”.	59
Tabla 26. Caso de uso “Editar datos de la estación”.	60
Tabla 27. Caso de uso “Añadir usuario dependiente a la estación”.	61
Tabla 28. Caso de uso “Modificar usuario de la estación”.	62
Tabla 29. Caso de uso “Eliminar usuario dependiente de la estación”.	63
Tabla 30. Caso de uso “Añadir surtidor a la estación”.	64
Tabla 31. Caso de uso “Modificar surtidor a la estación”.	65

Tabla 32. Caso de uso “Eliminar surtidor a la estación”.	66
Tabla 33. Caso de uso “Consultar y actualizar stock”.	67
Tabla 34. Caso de uso “Exportar la consulta de stock a Excel”.	68
Tabla 35. Caso de uso “Consultar operaciones”.	69
Tabla 36. Caso de uso “Exportar la consulta de operaciones a Excel”.	70
Tabla 37. Caso de uso “Realizar devoluciones”.	71
Tabla 38. Diccionario de datos de “Users”.	74
Tabla 39. Diccionario de datos de “Shops”.	74
Tabla 40. Diccionario de datos de “Products”.	74
Tabla 41. Diccionario de datos de “Pumps”.	75
Tabla 42. Diccionario de datos de “Pipes”.	75
Tabla 43. Diccionario de datos de “Operations”.	75
Tabla 44. Diccionario de datos de “Operation_products”.	75
Tabla 45. PCN-01 - Registro de usuario propietario.	107
Tabla 46. PCN-02 - Inicio de sesión.	107
Tabla 47. PCN-03 - Recuperar contraseña.	107
Tabla 48. PCN-04 - Dar de alta un producto de tienda.	108
Tabla 49. PCN-05 - Dar de alta una carpeta de productos.	108
Tabla 50. PCN-06 - Dar de alta un combustible.	108
Tabla 51. PCN-07 - Modificar un producto de tienda.	109
Tabla 52. PCN-08 - Eliminar un producto de tienda.	109
Tabla 53. PCN-09 - Modificar un producto combustible.	109
Tabla 54. PCN-10 - Eliminar un producto combustible.	109
Tabla 55. PCN-11 - Crear nuevo usuario dependiente.	110
Tabla 56. PCN-12 - Modificar usuario dependiente.	110
Tabla 57. PCN-13 - Eliminar usuario dependiente.	110
Tabla 58. PCN-14 - Modificar nombre y email de usuario propietario.	111
Tabla 59. PCN-15 - Modificar contraseña de usuario propietario.	111
Tabla 60. PCN-16 - Crear nuevo surtidor sin combustibles.	111
Tabla 61. PCN-17 - Crear nuevo surtidor con combustibles.	112
Tabla 62. PCN-18 - Modificar surtidor.	112
Tabla 63. PCN-19 - Eliminar surtidor.	112
Tabla 64. PCN-20 - Modificar datos de la estación.	113
Tabla 65. PCN-21 - Venta en efectivo de combustible.	113

Tabla 66. PCN-22 - Venta en efectivo de producto tienda.	113
Tabla 67. PCN-23 - Venta en efectivo de varios productos.	113
Tabla 68. PCN-24 - Venta con tarjeta de combustible.	114
Tabla 69. PCN-25 - Venta con tarjeta de producto tienda.	114
Tabla 70. PCN-26 - Venta con tarjeta de varios productos.	114
Tabla 71. PCN-27 - Consultar lista de ventas.	115
Tabla 72. PCN-28 - Exportar ventas a Excel.	115
Tabla 73. PCN-29 - Consultar stock de productos.	115
Tabla 74. PCN-30 - Exportar consulta de stock a Excel.	115
Tabla 75. PCN-32 - Consultar ventas a devolver.	116
Tabla 76. PCN-33 - Realizar una devolución.	116

Índice de figuras

Figura 1. Captura de pantalla del C18, perteneciente a la solución VeriFone Commander.	13
Figura 2. Captura de pantalla de LS Retail.	13
Figura 3. Captura de pantalla de NCR VOYIX POS.	14
Figura 4. Captura de pantalla de CPista.	14
Figura 5. Captura de pantalla de TPV Garum.	15
Figura 6. Comparativa de metodologías de desarrollo.	19
Figura 7. Diagrama de Gantt estimado del proyecto.	30
Figura 8. Diagrama de Gantt real de proyecto.	33
Figura 9. Árbol de características de la aplicación.	38
Figura 10. Diagrama de casos de uso “Registro”.	43
Figura 11. Diagrama de casos de uso “Venta”.	44
Figura 12. Diagrama de casos de uso “Gestión de productos tienda”.	44
Figura 13. Diagrama de casos de uso “Gestión de productos combustible”.	45
Figura 14. Diagrama de casos de uso “Administrar estación”.	45
Figura 15. Diagrama de casos de uso “Consulta de operaciones y stock”.	46
Figura 16. Diagrama de Entidad-Relación.	73
Figura 17. Diagrama de arquitectura lógica.	78
Figura 18. Patrón de arquitectura MVVM.	79
Figura 19. Diagrama de arquitectura física.	80
Figura 20. Diagrama de secuencia - Iniciar sesión y Venta.	82
Figura 21. Diagrama de secuencia - Registrar y editar datos de estación.	83
Figura 22. Diagrama de secuencia - Crear y Eliminar usuario dependiente.	84
Figura 23. Interfaz de usuario - Inicio de sesión.	85
Figura 24. Interfaz de usuario - Registro.	86
Figura 25. Interfaz de usuario - Recuperar contraseña.	86
Figura 26. Interfaz de usuario - Pantalla principal de estación.	87
Figura 27. Interfaz de usuario - Administrar estación.	87
Figura 28. Interfaz de usuario - Administrar usuarios.	88
Figura 29. Interfaz de usuario - Formulario de usuarios.	89
Figura 30. Interfaz de usuario - Administrar surtidores.	89
Figura 31. Interfaz de usuario - Formulario de surtidores.	90

Figura 32. Interfaz de usuario - Editar datos de estación.	90
Figura 33. Interfaz de usuario - Consulta de operaciones y stock.	91
Figura 34. Interfaz de usuario - Consulta de operaciones.	91
Figura 35. Interfaz de usuario - Consulta de stock.	92
Figura 36. Interfaz de usuario - Devoluciones.	92
Figura 37. Interfaz de usuario - Editar productos.	93
Figura 38. Interfaz de usuario - Formulario de productos tienda.	93
Figura 39. Interfaz de usuario - Formulario de productos combustible.	94
Figura 40. Estructura completa de la aplicación.	95
Figura 41. Carpeta de funciones del API del servidor de la aplicación.	96
Figura 42. Carpeta “src” del proyecto.	97
Figura 43. Carpeta “src/app” del proyecto.	97
Figura 44. Ejemplo de clase existente en “src/app/classes/dbModels.ts”.	98
Figura 45. Carpeta “src/utills”.	98
Figura 46. Fichero “pump.ts”.	99
Figura 47. Método “GetPumps()” para convertir las listas de surtidores.	100
Figura 48. Método “onResize()”.	100
Figura 49. Evento “onclick” del botón “nextButton”.	101
Figura 50. Evento “onclick” del botón “prevButton”.	102
Figura 51. Inicio de función “deleteUser.js”.	103
Figura 52. Comprobaciones de la petición en la función “deleteUser.js”.	103
Figura 53. Peticiones para eliminar el usuario en “deleteUser.js”.	104
Figura 54. Peticiones para eliminar el usuario en “deleteUser.js”.	104
Figura 55. Clase “HttpError”.	105
Figura 56. Función “getJSONResponse()”.	105
Figura 57. Inicio de sesión - Ir a “Registro”.	117
Figura 58. Registro de usuarios.	118
Figura 59. Inicio de sesión.	119
Figura 60. Recuperar contraseña.	120
Figura 61. Pantalla principal sin datos de estación.	121
Figura 62. Subpantalla - Editar productos.	122
Figura 63. Subpantalla - Editar carburantes.	122
Figura 64. Subpantalla - Formulario de carburantes.	123
Figura 65. Subpantalla - Editar carburantes.	123

Figura 66. Subpantalla - Editar productos de tienda.	124
Figura 67. Subpantalla - Formulario de productos de tienda.	124
Figura 68. Desplegable de “Carpeta en la que se mostrará el producto”.	125
Figura 69. Subpantalla - Editar árbol de tienda.	125
Figura 70. Subpantalla - Seleccionar “Administrar estación”.	126
Figura 71. Subpantalla - Administrar estación.	127
Figura 72. Subpantalla - Administrar usuarios.	128
Figura 73. Subpantalla - Formulario “Añadir nuevo usuario”.	128
Figura 74. Subpantalla - Administrar usuarios.	129
Figura 75. Subpantalla - Formulario “Modificar usuario dependiente”.	129
Figura 76. Subpantalla - Formulario “Modificar usuario propietario”.	130
Figura 77. Subpantalla - Formulario “Modificar contraseña”.	130
Figura 78. Subpantalla - Administrar surtidores.	131
Figura 79. Subpantalla - Formulario “Añadir nuevo surtidor”.	131
Figura 80. Subpantalla - Formulario “Añadir nuevo surtidor” con manguera.	131
Figura 81. Subpantalla - Administrar surtidores.	132
Figura 82. Subpantalla - Administrar surtidores.	132
Figura 83. Subpantalla - Formulario estación.	133
Figura 84. Pantalla principal.	134
Figura 85. Pantalla principal.	134
Figura 86. Subpantalla - Documento de venta.	135
Figura 87. Subpantalla - Subcarpeta del árbol de productos vendibles.	136
Figura 88. Subpantalla - Documento de venta.	136
Figura 89. Subpantalla - Lista de surtidores.	137
Figura 90. Subpantalla - Lista de surtidores seleccionados.	137
Figura 91. Subpantalla - Documento de venta.	137
Figura 92. Subpantalla - Documento de venta.	138
Figura 93. Subpantalla - Opciones de pago.	138
Figura 94. Subpantalla - Confirmar el pago.	138
Figura 95. Subpantalla - Venta procesada.	139
Figura 96. Subpantalla - Seleccionar “Consulta de operaciones y stock”.	139
Figura 97. Subpantalla - Consulta de operaciones y stock.	139
Figura 98. Subpantalla - Consulta de operaciones.	140
Figura 99. Subpantalla - Consulta de stock.	141

Figura 100. Subpantalla - Actualizar stock.	141
Figura 101. Subpantalla - Devoluciones.	141
Figura 102. Subpantalla - Confirmar devolución.	142
Figura 103. Subpantalla - Devoluciones.	142
Figura 104. Subpantalla - Consulta de operaciones devueltas.	142
Figura 105. Extracto de la implementación de GitHub Actions.	143
Figura 106. Captura de pantalla del menú “Repository Secrets” de GitHub Actions.	144

