

Memoria del proyecto

Desarrollo de una cesta de compra con autodetección de productos orientado al sector retail

Trabajo de Fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA



**VNiVERSiDAD
D SALAMANCA**

Julio de 2023

Autor

Pablo Santos Blázquez

Tutores

Sergio García González

Gabriel Villarrubia González

Certificado de los tutores del TFG

D. Gabriel Villarrubia González y Sergio García González, profesores del Departamento de Informática y Automática de la Universidad de Salamanca,

HACEN CONSTAR:

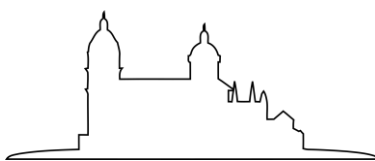
Que el Trabajo de Fin de Grado titulado “Desarrollo de una cesta de compra con autodetección de productos orientado al sector retail”, que se presenta, ha sido realizado por Pablo Santos Blázquez, con DNI 70907042C y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en Ingeniería Informática en esta Universidad.

Salamanca, 3 de julio de 2023.

Fdo.: Sergio García González

Fdo.: Gabriel Villarrubia González

A mis padres



*“Si tienes que buscar, anda y busca,
siempre aquí tendrás tu casa.”*

Resumen

En el mundo del sector retail, la necesidad de agilizar el proceso de cobro de productos se ha convertido en uno de los principales desafíos. Las largas colas en las cajas de pago generan inconvenientes tanto para los clientes como para los establecimientos, provocando la necesidad de contar con grandes grupos de trabajadores en las cajas para mantener la fluidez y reducir los tiempos de espera.

Diversas investigaciones han identificado este punto como un cuello de botella que afecta la eficiencia y la satisfacción del cliente y que provoca anualmente pérdidas en ventas de más de 37.000 millones de dólares^[1]. Según un informe de la National Retail Federation (NRF)^[2], las ventas en el sector retail experimentarán un incremento entre el 4% y el 6% durante 2023, por lo que la mejora de este sector se debe llevar a cabo de forma casi inmediata.

En este contexto se me planteó un reto para solucionar este problema: el desarrollo de un carro de la compra inteligente que permitiese a los clientes gestionar todo el proceso de compra de manera autónoma. Para ello he diseñado un dispositivo equipado con un lector de código de barras y un sensor de peso, capaz de identificar los productos que los clientes seleccionan y de realizar el pago automáticamente al finalizar la compra, eliminando así la necesidad de pasar por una caja de cobro tradicional.

Mi propuesta busca abordar el desafío de la congestión en las áreas de caja de pago, ofreciendo una solución que agiliza el proceso de compra y mejora la experiencia del cliente. Con el respaldo de estudios que han identificado el crecimiento continuo de las ventas en establecimientos y la problemática de las áreas de caja de pago, este enfoque innovador tiene el potencial de transformar la dinámica de las tiendas del sector retail, brindando mayor eficiencia y satisfacción tanto para los clientes como para los establecimientos comerciales.

Palabras clave: sector retail, satisfacción del cliente, carro de la compra inteligente, pago automático.

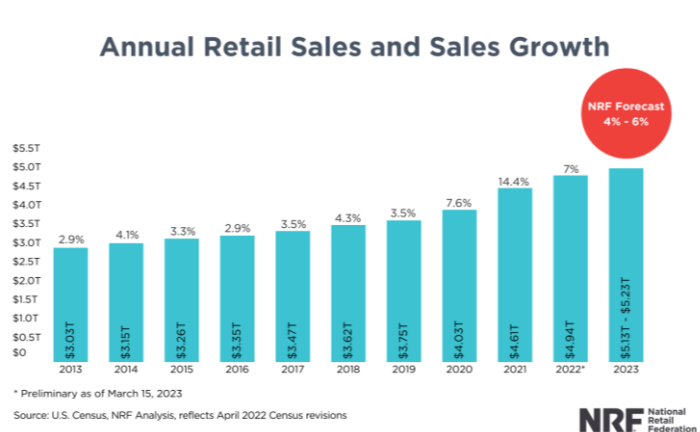


Figura 1. Crecimiento anual de ventas y del sector retail según la NRF (2023)

Summary

In the retail sector, the need to speed up the checkout process has become one of the main challenges. Long lines at checkout counters generate inconveniences for both customers and establishments, causing the need to have large groups of workers at checkout counters to maintain fluidity and reduce waiting times.

Several studies have identified this as a bottleneck that affects efficiency and customer satisfaction and causes annual sales losses of more than \$37 billion^[1]. According to a report by the National Retail Federation (NRF)^[2], sales in the retail sector will increase by 4% to 6% during 2023, so improvements in this sector must be made almost immediately.

In this context, I was challenged to resolve this problem: the development of an intelligent shopping cart that would allow customers to manage the entire shopping process autonomously. To this end, I designed a device equipped with a barcode reader and a weight sensor, capable of identifying the products that customers are selecting and automatically making the payment at the end of the purchase, eliminating the need to go through a traditional checkout.

My proposal aims to solve the challenge of congestion at checkout areas by providing a solution that speeds up the checkout process and improves the customer experience. Supported by studies that have identified the continued growth of in-store sales and checkout area issues, this innovative approach has the potential to transform the dynamics of retail stores, providing greater efficiency and satisfaction for both customers and stores.

Keywords: retail sector, customer satisfaction, smart shopping cart, automatic payment.

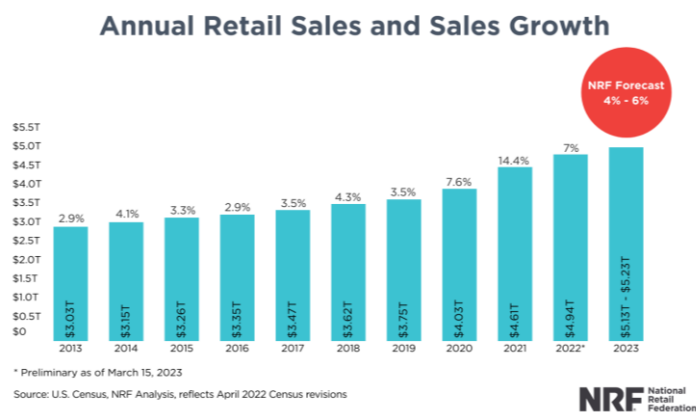


Figura 1 (bis). Annual retail and sales growth by NRF (2023)

Tabla de contenidos

1) Introducción	1
2) Objetivos	3
2.1) Objetivos del sistema.....	3
2.2) Objetivos personales	4
3) Conceptos teóricos	5
3.1) Lectura de códigos de barras.....	5
3.2) Lectura de cambios de peso.	5
3.2.1) Células de carga de punto de apoyo único:	6
3.2.2) Células de carga de punto de apoyo múltiple.....	6
3.3) Reconocimiento de la posición final del dispositivo.....	8
4) Técnicas y herramientas	9
4.1) Técnicas y herramientas utilizadas en el desarrollo software.....	9
4.2) Técnicas y herramientas utilizadas en el desarrollo hardware	11
4.3) Técnicas y herramientas CASE	11
4.4) Técnicas y herramientas utilizadas en el control de versiones	12
5) Estado del arte	13
5.1) Amazon Go	13
5.2) Caper Cart.....	13
5.3) Veeve Connected Screen	14
6) Aspectos relevantes del desarrollo	15
6.1) Planificación y gestión del proyecto	15
6.1.1) Estimación de la duración del proyecto.....	15
6.1.2) Planificación temporal	17
6.2) Elicitación de requisitos.....	19
6.2.1) Organización y autores.....	19
6.2.2) Objetivos del sistema	19
6.2.3) Requisitos de información.....	20
6.2.4) Requisitos funcionales	21
6.2.5) Requisitos no funcionales.....	24
6.3) Análisis de requisitos	26
6.3.1) Diagramas de secuencia.....	26
6.3.2) Arquitectura del sistema	27
6.4) Diseño del sistema	29
6.4.1) Patrón arquitectónico.....	29
6.4.2) Subsistemas de diseño	29

6.4.3) Vista arquitectónica.....	31
6.4.4) Realización de casos de uso	32
6.4.5) Diseño de la base de datos	32
6.4.6) Modelo de despliegue	33
6.5) Implementación del sistema	34
6.5.1) Implementación software	35
6.5.2) Construcción del dispositivo.....	46
7) Limitaciones y/o problemas identificados.....	52
7.1) Realización del pago.....	52
7.2) Detección de productos de poco peso.....	52
7.3) Diseño e implementación del dispositivo.....	52
8) Conclusiones y líneas de trabajo futuras	53
8.1) Conclusiones	53
8.2) Líneas de trabajo futuras.....	53
9) Bibliografía	55

Índice de figuras

Figura 1. Crecimiento anual de ventas y del sector retail según la NRF (2023)	6
Figura 1 (bis). Annual retail and sales growth by NRF (2023)	7
Figura 2. Lector de código de barras omnidireccional Lenvii E688.....	5
Figura 3. Estructura de las células de carga de punto de apoyo único	6
Figura 4. Arquitectura de punto de apoyo único mediante el uso de células de carga de apoyo múltiple.....	6
Figura 5. Arquitectura de punto de apoyo doble mediante el uso de células de carga de apoyo múltiple.....	7
Figura 6. Arquitectura de punto de apoyo cuádruple mediante el uso de células de carga de apoyo múltiple.....	7
Figura 7. Esquema de conexiones del dispositivo receptor RFID	8
Figura 8. Tienda Amazon Go.....	13
Figura 9. Caper Cart 3	13
Figura 10. Veeve Connected Screen	14
Figura 11. Iconografía del Proceso Unificado	15
Figura 12. Fases del Proceso Unificado	15
Figura 13. Introducción de factores en EZEstimate	16
Figura 14. Estimación temporal de EZEstimate	16
Figura 15. Diagrama de Gantt de la vista global de la planificación temporal	17
Figura 16. Vista global de la planificación temporal	18
Figura 17. Diagrama de paquetes.....	21
Figura 18. Actores del sistema.....	21
Figura 19. Diagrama de casos de uso	22
Figura 20. Ejemplo de diagrama de casos de uso distribuidos por subsistemas ...	23
Figura 21. Modelo del dominio.....	26
Figura 22. Ejemplo de diagrama de secuencia: UC-01: Identificar usuario	27
Figura 23. Ejemplo de clase de análisis: Vista.....	27
Figura 24. Propuesta de arquitectura	28
Figura 25. Arquitectura MVVM en proyectos .NET	29
Figura 26. Diagrama de subsistemas de diseño.....	30
Figura 27. Ejemplo de clases de diseño: Vista	31
Figura 28. Vista arquitectónica	31
Figura 29. Ejemplo de realización de caso de uso: UC-01: Identificar usuario.....	32
Figura 30. Modelo de diseño de la base de datos.....	33
Figura 31. Diagrama de despliegue	33

Figura 32. Estructura general del proyecto	34
Figura 33. Iconografía del dispositivo final: USuAL Shopping Cart.....	35
Figura 34. Colores institucionales de la Universidad de Salamanca.....	35
Figura 35. Pantalla de standby	36
Figura 36. Pantalla de petición de QR	37
Figura 37. Pantalla de introducción del código pin para usuarios	37
Figura 38. Pantalla de introducción del código pin para administradores.....	38
Figura 39. Pantalla de introducción del código pin tras superar el máximo de intentos	38
Figura 40. Pantalla de compra en su estado inicial.....	39
Figura 41. Pantalla de compra con un producto detectado	40
Figura 42. Pantalla de compra con un producto añadido.....	40
Figura 43. Pantalla de compra durante la modificación de la cantidad de un producto añadido.....	41
Figura 44. Pantalla de compra tras la modificación de la cantidad de un producto añadido.....	41
Figura 45. Pantalla de compra con varios productos añadidos	42
Figura 46. Pantalla de compra tras un cambio de peso no esperado.....	42
Figura 47. Pantalla de pago mientras se este se realiza	43
Figura 48. Pantalla de imposibilidad de pago	44
Figura 49. Pantalla de estadísticas generales.....	45
Figura 50. Pantalla de estadísticas de ventas.....	45
Figura 51. Pantalla de estadísticas de venta de productos.....	46
Figura 52. Carro de compra Marsanz 100 ATH	47
Figura 53. Modelo 3D del soporte principal	47
Figura 54. Alzado del modelo 3D del soporte principal	48
Figura 55. Soporte principal incorporado al carro	48
Figura 56. Modelo 3D de la base del sensor de peso.....	49
Figura 57. Modelo 3D de los soportes de las células de carga.	49
Figura 58. Modelo 3D de la caja contenedora del lector RFID	50
Figura 59. Lector RFID incorporado al carro.....	50
Figura 60. Resultado final del dispositivo	51
Figura 61. Ejemplo de estado del dispositivo: vacío	51

Índice de tablas

Tabla 1. Ejemplo de objetivo: OBJ-01: Identificación de usuarios	19
Tabla 2. Ejemplo de requisito de información: IRQ- 01: Información sobre usuarios	20
Tabla 3. Ejemplo de actor: ACT- 01: Usuario no identificado.....	21
Tabla 4. Ejemplo de realización de caso de uso: UC-01: Identificar usuario	24
Tabla 5. Ejemplo de requisito no funcional: NFR-01: Precisión.....	25

1) Introducción

El objetivo principal de este proyecto es desarrollar un dispositivo que colabore en la adaptación del sector retail al mundo tecnológico, aprovechando las ventajas que este ofrece.

Para lograr dicho objetivo se debe partir de una relación de colaboración entre cliente y dispositivo, ya que la principal función de este es la de facilitar y mejorar la experiencia de todo usuario. Añadir medidas de seguridad adicionales para los diferentes establecimientos del sector retail es un objetivo de carácter secundario.

Como mejora en la gerencia del establecimiento, se pone a disposición de los responsables y/o administradores una herramienta de seguimiento y control de ventas y productos realizados mediante el dispositivo desarrollado.

Para obtener una visión global más clara de los principales aspectos del proyecto se ha dividido este documento en los siguientes apartados^[3]:

- **Objetivos:** se especifican los objetivos que se persiguen con la realización del proyecto.
- **Conceptos teóricos:** se especifican los aspectos más relevantes para la correcta comprensión de los contenidos más avanzados.
- **Técnicas y herramientas:** se presentan las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto.
- **Estado del arte:** se exponen las últimas actualizaciones e investigaciones en torno a la temática del proyecto.
- **Aspectos relevantes del desarrollo:** se recogen los aspectos más importantes, o a destacar, del desarrollo del sistema software dentro del proyecto.
- **Limitaciones y/o problemas identificados:** se exponen los problemas y desafíos identificados durante la elaboración del proyecto.
- **Conclusiones y líneas de trabajo futuras:** se exponen las conclusiones extraídas y las posibles mejoras a realizar en futuras versiones.

De forma complementaria a este documento se adjuntan los siguientes anexos:

- **Anexo I – Plan del proyecto software:** se describen las principales tareas, la estimación de su duración y la planificación de estas a lo largo del tiempo.
- **Anexo II – Especificación de requisitos software:** se elicitan y especifican los diferentes tipos de requisitos del sistema.
- **Anexo III – Análisis de requisitos software:** se analizan con mayor detalle los requisitos del sistema.

- **Anexo IV - Diseño del sistema software:** se especifican las diferentes fases del diseño del sistema.
- **Anexo V - Documentación técnica:** se explica la funcionalidad del sistema para lograr un entendimiento más sencillo de la estructura de la documentación y el código aportado.
- **Anexo VI - Manual de usuario:** explica detalladamente el correcto uso de la aplicación y su puesta en marcha.
- **Anexo VII - Diseño y construcción del dispositivo:** describe tanto el diseño como el montaje del prototipo.

2) Objetivos

En este apartado se exponen los objetivos perseguidos con la realización del proyecto, distinguiendo entre los objetivos marcados por los requisitos del software (objetivos del sistema) y los objetivos personales planteados para afrontar la elaboración del mismo.

2.1) Objetivos del sistema

El principal objetivo es el diseño y desarrollo de un carro de la compra inteligente capaz de gestionar todo el proceso de compra basado en la interacción y colaboración con los clientes para conseguir una mejora en su experiencia durante la fase de compra y una descongestión de las principales zonas de pago.

Para lograr estos objetivos, este dispositivo cuenta con un lector de código de barras y un sensor de peso capaz de identificar y comprobar los artículos que se introducen dentro del mismo. Al finalizar la compra, el cliente solo tiene que dirigirse a la salida. Si su carro contiene todos los artículos correctamente identificados, podrá avanzar hacia la salida sin tener que realizar ningún trámite más, agilizando así el proceso de pago. Si por el contrario se detecta alguna discrepancia, el cliente tendrá que avanzar hacia una caja de cobro.

A continuación, se describen los objetivos de manera más específica:

- **Identificación de usuarios:** el sistema deberá ser capaz de identificar los usuarios que hagan uso de él, distinguiendo entre clientes (usuarios estándar) y administradores. Todos los usuarios deberán verificar su identidad mediante la introducción de una clave personal.
- **Seguimiento de productos:** el sistema deberá ser capaz de identificar los productos escaneados por el usuario a la vez que comprobará mediante el uso del peso del producto si el producto escaneado se podría corresponder con el realmente introducido. El sistema también deberá ser capaz de adaptarse en todo momento a las modificaciones que los usuarios puedan realizar dentro del dispositivo.
- **Monitorización e información del estado del dispositivo:** el sistema deberá ser capaz de comprobar y comunicar su estado a los usuarios de forma simple. Todo tipo de usuario debe ser capaz de entender qué acciones se deben realizar para conseguir una correcta interacción y, por lo tanto, un correcto uso del dispositivo.
- **Recopilación de estadísticas:** el sistema deberá ser capaz de recoger estadísticas sobre las ventas realizadas a través del uso de los dispositivos para procesarlas y así ser capaz de mostrar la información más relevante a los administradores de forma gráfica y visual.
- **Simplificación del pago:** el sistema deberá ser capaz de aligerar la confluencia en las cajas de pago mediante la realización de pagos automáticos. Estos pagos se llevarán a cabo cuando el usuario decida

abandonar el recinto y solo se realizarán si el sistema detecta que su estado es el correcto.

2.2) Objetivos personales

Cuando se me propuso desarrollar este proyecto, mi principal motivación fue superarme a mí mismo, ya que sabía que este proyecto no solo pondría a prueba mis conocimientos como programador, sino también mi determinación, mi capacidad resolutiva y mi capacidad de adaptación a nuevos ámbitos como el diseño 3D o el desarrollo hardware.

La idea de este proyecto surgió durante el desarrollo de mis prácticas externas, realizadas con el grupo de investigación de la Universidad de Salamanca "ESALAB". Esta idea de proyecto me planteó grandes desafíos, pero a la vez consiguió que afianzara conocimientos y fuera capaz de dar más perspectiva a mis capacidades.

Anteriormente había desarrollado pequeños proyectos con Arduino o diseñado piezas 3D simples, pero nunca me había enfrentado a un proyecto de hardware/diseño de estas dimensiones.

Para el desarrollo del sistema y del dispositivo mis principales objetivos fueron la profundización en el conocimiento de las herramientas que ofrece la tecnología .NET, desarrollar la capacidad de diseñar piezas 3D complejas para su posterior implantación física y aprender a ensamblar diferentes tipos de sensores y actuadores para crear sistemas complejos.

En definitiva, mi objetivo principal con este proyecto era dar un paso al frente y evolucionar como ingeniero y como informático para ser capaz de diseñar y desarrollar un gran sistema complejo, actuando así como un ingeniero informático.

3) Conceptos teóricos

En este apartado se sintetizan los principales conceptos teóricos necesarios para la sencilla comprensión del proyecto. Para ello se detallan los conceptos relevantes sobre los componentes físicos del sistema, así como su proceso de desarrollo y elección, justificando las diferentes decisiones tomadas.

3.1) Lectura de códigos de barras.

Uno de los principales objetivos a tratar es la correcta identificación de los productos. Para lograr dicho objetivo con la mayor precisión y rapidez posible se optó por el uso de los dispositivos más comunes y validados internacionalmente: los lectores de códigos de barras.

La incorporación de este dispositivo no solo cumple con las principales características necesarias, sino que aporta también cercanía al usuario, al estar estos familiarizados con su forma de uso.

Tras investigar sobre las principales tecnologías ofrecidas por estos dispositivos decidí que se debía utilizar un lector omnidireccional. Este tipo de lectores permiten que los usuarios no deban preocuparse por la orientación de los códigos de barras, favoreciendo una mejor experiencia de uso.

El dispositivo elegido para la incorporación en el sistema fue el modelo E688 desarrollado por la empresa Lenvii.

Dicho modelo es capaz de leer e identificar todo tipo de códigos, tanto unidimensionales (códigos de barras) como bidimensionales (QRs), incluso si estos están dañados, distorsionados o reflejan la luz directa.



Figura 2. Lector de código de barras omnidireccional Lenvii E688

3.2) Lectura de cambios de peso.

Para comprobar que los productos escaneados se corresponden con los introducidos en el carro decidí usar su peso como factor crítico. Fue por lo tanto de vital importancia incorporar un dispositivo capaz de obtener el peso de dichos productos con la mayor precisión posible, sin olvidar que los productos debían poder ser depositados en cualquier sector del fondo del carro sin dar lugar a modificaciones en la lectura del peso de los mismos.

Por estos motivos opté por diseñar y desarrollar un sensor de peso propio que se adaptase a la forma y requisitos del sistema.

Tras investigar sobre las diferentes posibilidades técnicas para el desarrollo de dicho sensor se optó por el uso de células de carga.

Estas células son esencialmente transductores que convierten la fuerza o el peso en una señal eléctrica mediante el uso de medidores de tensión. Cuando se encuentran bajo carga estas se deforman ligeramente, provocando un cambio físico que es detectado por los medidores de deformación que se distorsionan al unísono con el cuerpo, lo que resulta en un cambio de voltaje proporcional a la fuerza o peso inicial.

Para la incorporación al sistema se planteó el uso de dos tipos de células de carga:

- **3.2.1) Células de carga de punto de apoyo único:** células de carga con capacidad para acomodar la carga descentrada. Por lo general, la carga debe alinearse con la celda para mantener lecturas precisas, de otra forma se introducirán errores de lectura. Muy útiles en superficies pequeñas.

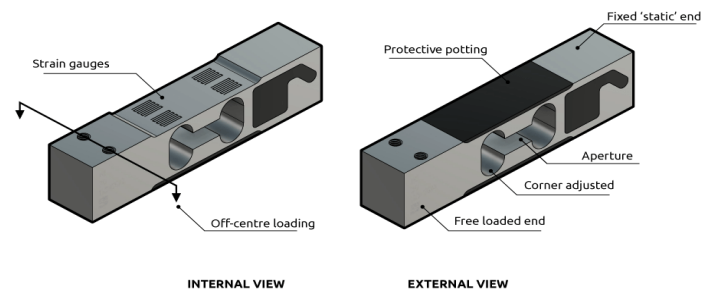


Figura 3. Estructura de las células de carga de punto de apoyo único

- **3.2.2) Células de carga de punto de apoyo múltiple:** células de carga versátiles capaces de adoptar diferentes tipos de funcionamiento dependiendo de su arquitectura:

- **Arquitectura de punto de apoyo único:** imita el comportamiento de una célula de carga de punto de apoyo único.

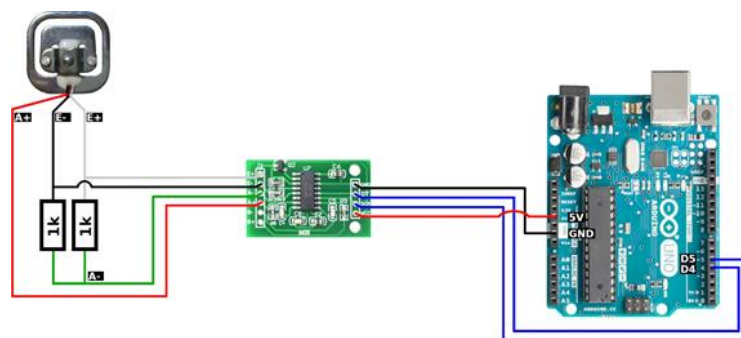


Figura 4. Arquitectura de punto de apoyo único mediante el uso de células de carga de apoyo múltiple

- **Arquitectura de doble punto de apoyo:** se utilizan dos células de carga para reducir los errores introducidos al colocar elementos de forma descentralizada en uno de los ejes de la superficie de contacto.

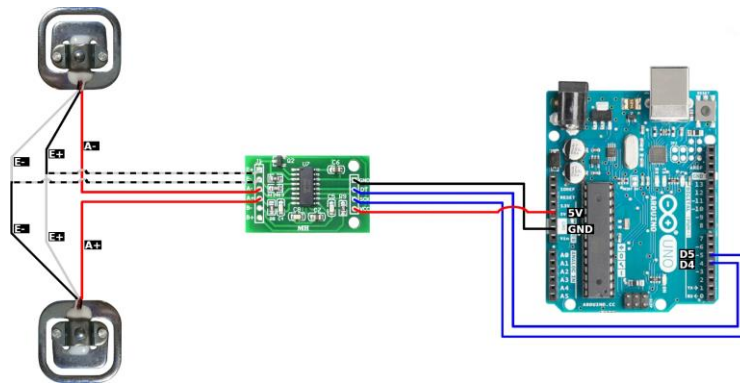


Figura 5. Arquitectura de punto de apoyo doble mediante el uso de células de carga de apoyo múltiple

- **Arquitectura de cuádruple punto de apoyo:** se utilizan cuatro células de carga para reducir los errores introducidos al colocar elementos en cualquier sector de la superficie de contacto.

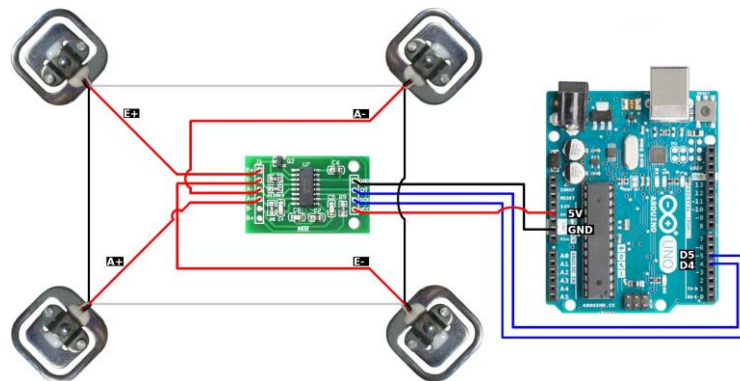


Figura 6. Arquitectura de punto de apoyo cuádruple mediante el uso de células de carga de apoyo múltiple

Finalmente opté por el uso de la *arquitectura de punto de apoyo cuádruple mediante el uso de células de carga de apoyo múltiple*, ya que dicho sistema ofrece el mejor comportamiento ante las distribuciones descentradas causadas por diferentes pesos. Como superficie de apoyo para dicha arquitectura utilicé un tablón de conglomerado de diez milímetros con la forma de la base del carro. Dicha base aportó la rigidez necesaria para conseguir mediciones precisas.

3.3) Reconocimiento de la posición final del dispositivo.

Otro de los principales objetivos es la realización del pago automáticamente al finalizar la compra.

Para detectar cuando un usuario ha finalizado una compra decidí detectar si el carro había llegado a un área de salida.

La principal dificultad de este objetivo era la detección de la posición del carro. Para abordar este problema investigué sobre diferentes métodos de rastreo y seguimiento de objetos, siendo el más adecuado para el sistema el uso de envíos por radiofrecuencia (RF).

Mediante el uso de esta tecnología se pretendían emitir mensajes dentro de las áreas de salida que serían detectados por los dispositivos de forma automática. El principal problema de este enfoque fue la emisión de dichos mensajes, ya que las ondas emitidas por los emisores se transmitían omnidireccionalmente, lo que hacía muy complicada y costosa su restricción a una cierta área.

La solución a este problema fue la identificación por radiofrecuencia (RFID).

Al acortar el rango de detección a prácticamente el contacto físico, tan sólo se debe indicar al usuario que colocara el receptor RFID del carro junto a una de las zonas especificadas en las áreas de salida. Dichas áreas contienen una tarjeta RFID que una vez detectadas dan paso a la tramitación del proceso de pago. Si durante este proceso se detectara alguna irregularidad se redirigiría al usuario hacia una de las cajas de pago convencionales, donde una vez realizado el pago el trabajador utilizaría su tarjeta RFID para reiniciar el estado del carro.

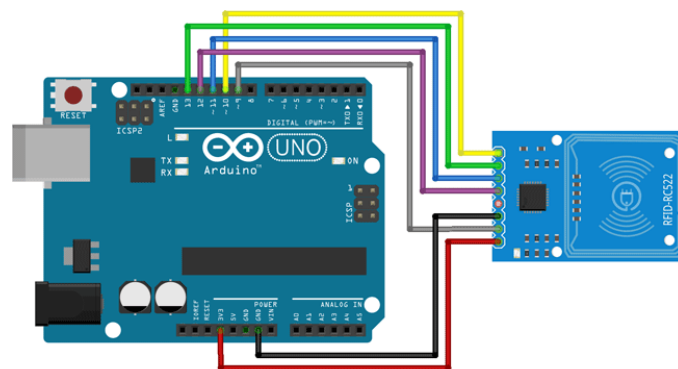


Figura 7. Esquema de conexiones del dispositivo receptor RFID

4) Técnicas y herramientas

En este apartado se presentan las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Para ello se va a distinguir entre las técnicas y herramientas utilizadas en cada uno de los distintos ámbitos del desarrollo del sistema.

4.1) Técnicas y herramientas utilizadas en el desarrollo software

Las técnicas y herramientas utilizadas en el desarrollo de software se refieren a los enfoques metodológicos y las aplicaciones utilizadas para diseñar, desarrollar, testear y mantener el software.

- **Visual Studio**

Entorno de Desarrollo Integrado (IDE) de software libre multiplataforma desarrollado por Microsoft que ofrece herramientas para la edición de código, depuración, gestión de proyectos y control de versiones entre otras herramientas. Es una solución completa para el desarrollo de software.

Se ha utilizado esta herramienta para el desarrollo del código fuente mediante la creación de un proyecto C# WPF (.NET).

- **C#**

Lenguaje de programación orientado a objetos de propósito general que combina elementos de C++ y Java desarrollado por Microsoft.

Algunas de sus características más importantes son:

- **Programación orientada a objetos:** C# permite utilizar conceptos de la programación orientada a objetos, como clases, objetos, herencia, polimorfismo y encapsulamiento, lo que facilita la organización y estructuración del código.

- **Framework de .NET:** C# se integra estrechamente con .NET, lo que proporciona una amplia biblioteca de clases y componentes reutilizables que aceleran el desarrollo de aplicaciones.

- **Windows Presentation Foundation (WPF)**

Framework de desarrollo de interfaces gráficas de usuario (IGUs) en Windows.

Algunas de sus características más importantes son:

- **Diseño basado en XAML:** WPF utiliza XAML (Extensible Application Markup Language) para definir la estructura y apariencia de la interfaz de usuario. XAML es un lenguaje declarativo que permite separar el diseño visual de la lógica subyacente, lo que facilita el desarrollo y la personalización de la interfaz.

- **Elementos visuales:** WPF proporciona una gran variedad de controles y elementos visuales personalizables para construir la interfaz de usuario. Se pueden aplicar estilos y plantillas para adaptar la apariencia de los controles a los requisitos del proyecto.

- **Enlazado de datos:** WPF permite enlazar los datos de la aplicación con los elementos visuales de la interfaz (binding), lo que provoca que los datos se reflejen automáticamente en los elementos visuales y viceversa.

- **Manejo de eventos:** los eventos son acciones que ocurren en WPF en respuesta a la interacción del usuario o cambios en el estado de los elementos visuales. Son muy útiles para responder y controlar múltiples acciones simultáneas.

- **Model-View-ViewModel (MVVM):** patrón arquitectónico que separa la lógica de presentación (ViewModel) de la interfaz de usuario (View) y los datos (Model), lo que facilita el mantenimiento del proyecto.

- **Arduino IDE**

Entorno de Desarrollo Integrado (IDE) utilizado para programar procesadores Arduino. Permite escribir, compilar y cargar código en las placas, haciendo uso de librerías predefinidas y proporcionando herramientas básicas de depuración y monitorización. Es una herramienta accesible y ampliamente utilizada para proyectos de hardware y software embebido.

Se ha utilizado esta herramienta para el desarrollo del código controlador de los diferentes dispositivos integrados.

- **Arduino**

Lenguaje de programación basado en C/C++ caracterizado por su versatilidad y accesibilidad. Cuenta con una amplia gama de librerías predefinidas que proporcionan funcionalidades adicionales que permiten la comunicación con otros dispositivos.

- **XAMPP**

Software que proporciona un entorno de servidor web local para el desarrollo de aplicaciones. Incluye Apache, MySQL y PHP, junto a otros intérpretes y herramientas.

Ha sido utilizado para la fácil instalación y eficiente gestión de la base de datos MySQL.

- **MySQL**

Sistema de gestión de bases de datos relacionales de código abierto gestionado por Oracle.

Algunas de sus características más importantes son:

- **Arquitectura cliente-servidor:** Cliente y servidor se comunican entre sí de manera diferencial para conseguir un mejor rendimiento.
- **Compatibilidad con SQL:** MySQL ofrece compatibilidad total con SQL.

- **phpMyAdmin**

Herramienta basada en web que proporciona una interfaz gráfica para administrar y gestionar bases de datos MySQL. Es una herramienta ampliamente utilizada en el desarrollo de aplicaciones web basadas en MySQL.

Su uso ha sido facilitado gracias al software XAMPP.

4.2) Técnicas y herramientas utilizadas en el desarrollo hardware

Las técnicas y herramientas utilizadas en el desarrollo hardware se refieren a los enfoques metodológicos y las aplicaciones utilizadas para diseñar, prototipar y fabricar componentes y sistemas electrónicos.

- **AutoCAD**

Herramienta de diseño y dibujo técnico utilizada para el desarrollo de proyectos hardware desarrollada por *Autodesk*. Permite la creación de diseños 2D y modelos 3D. Es una herramienta ampliamente utilizada en arquitectura, ingeniería y diseño industrial.

Se ha utilizado esta herramienta para el diseño de las piezas 3D necesarias para la incorporación de los diferentes dispositivos al sistema.

- **Repetier-Host**

Herramienta de software utilizada para controlar y gestionar impresoras 3D. Proporciona funciones de control de impresión, previsualización de modelos y monitorización en tiempo real, ofreciendo un servicio de slicer integrado.

Se ha utilizado esta herramienta para dividir en capas (slice) los diferentes modelos 3D creados para su posterior impresión.

4.3) Técnicas y herramientas CASE

Las herramientas CASE (Computer-Aided Software Engineering) son aplicaciones que ayudan en el proceso de ingeniería software a realizar tareas de análisis, diseño, implementación y mantenimiento de sistemas.

- **Visual Paradigm Community**

Herramienta CASE que proporciona un Entorno de Desarrollo Integrado para el modelado y la gestión de proyectos de desarrollo de software.

Permite el modelado UML, la generación de código, la gestión de requerimientos y la generación de documentación. Es utilizada para mejorar la calidad y la productividad en el desarrollo de software.

- **EZEstimate**

Herramienta CASE utilizada para realizar estimaciones de proyectos de desarrollo de software. Proporciona funcionalidades para estimar el esfuerzo, planificar recursos, realizar seguimiento y control del proyecto, generar informes y facilitar la integración con otras herramientas. Su objetivo es ayudar a los equipos de desarrollo a calcular de manera precisa los recursos necesarios y a realizar una gestión eficiente de los proyectos de software.

- **Microsoft Office Project**

Herramienta CASE utilizada para la gestión de proyectos de software. Proporciona funcionalidades para la planificación, asignación de recursos, programación, seguimiento y control del proyecto. Su objetivo es facilitar la gestión eficiente de los proyectos, asegurando que se cumplan los plazos y se optimicen los recursos disponibles.

- **KmCsharpHtmlDocGenerator**

Herramienta utilizada para la documentación del código fuente. Utiliza cabeceras con etiquetas XML para generar archivos HTML visualmente más entendibles. Su objetivo es facilitar la comprensión de la documentación generada durante el desarrollo de proyectos C#.

4.4) Técnicas y herramientas utilizadas en el control de versiones

- **GitHub Desktop**

Herramienta de control de versiones basada en Git, que facilita la colaboración y el seguimiento de cambios en proyectos de desarrollo de software. Proporciona funcionalidades para la gestión de ramas, como el historial de cambios o la comparación de versiones mediante una interfaz gráfica intuitiva. Su objetivo es mejorar la eficiencia y la colaboración en el desarrollo de software, garantizando el seguimiento preciso de los cambios.

- **Google Drive**

Herramienta de almacenamiento en la nube que también puede ser utilizada para el control de versiones de archivos ofimáticos.

Aunque no está diseñada específicamente para el control de versiones, ha sido utilizada para permitir el rápido acceso a las últimas versiones de los documentos que componen la memoria del proyecto desde varios dispositivos.

5) Estado del arte

En este apartado se exponen las últimas actualizaciones e investigaciones en torno a la temática del proyecto.

Respecto al desarrollo de carros de la compra inteligentes y automatizados, se han llevado a cabo varios proyectos a nivel mundial siendo, los siguientes proyectos expuestos, algunos de los más importantes.

5.1) Amazon Go

Uno de los proyectos más destacados es la implantación de tiendas sin cajeros realizada por Amazon, comenzando con la iniciativa en diciembre de 2016^[4].

Utiliza tecnologías como Machine Learning junto a sensores para permitir que los clientes tomen los productos de los estantes y se los lleven sin necesidad de pasar por una caja. Los sensores están ubicados en las diferentes estanterías y la tecnología de seguimiento de objetos registra los artículos seleccionados. Respecto al pago, se realiza automáticamente en la salida a través de la cuenta de Amazon del cliente.



Figura 8. Tienda Amazon Go

5.2) Caper Cart

Caper^[5] es una startup que ha desarrollado un carro de la compra inteligente capaz de interactuar con los clientes. Este muestra información sobre los productos y permite el pago directamente desde el carro. También ofrece recomendaciones personalizadas y promociones en tiempo real.

Las primeras versiones incluían un lector de código de barras, mientras que los últimos prototipos utilizan reconocimiento de imágenes en tiempo real para detectar los productos introducidos automáticamente.



Figura 9. Caper Cart 3

6) Aspectos relevantes del desarrollo

En este apartado se recogen los aspectos más importantes a destacar del desarrollo del sistema software y el dispositivo dentro del proyecto.

6.1) Planificación y gestión del proyecto

La primera tarea que se debe llevar a cabo en la gestión de un proyecto es definir el marco de trabajo dentro del cual se lleva a cabo el desarrollo del proyecto.

Para este proyecto he decidido trabajar siguiendo el marco de desarrollo software implantado por el Proceso Unificado^[7], el cual permite un desarrollo iterativo e incremental, y por lo tanto, otorga gran flexibilidad al proceso de desarrollo.



Figura 11. Iconografía del Proceso Unificado

El Proceso Unificado se conforma de varias fases de desarrollo solapadas que van evolucionando junto al proyecto. Estas fases son: modelado de negocio, elicitación de requisitos, análisis de requisitos, diseño del sistema, implementación y pruebas.

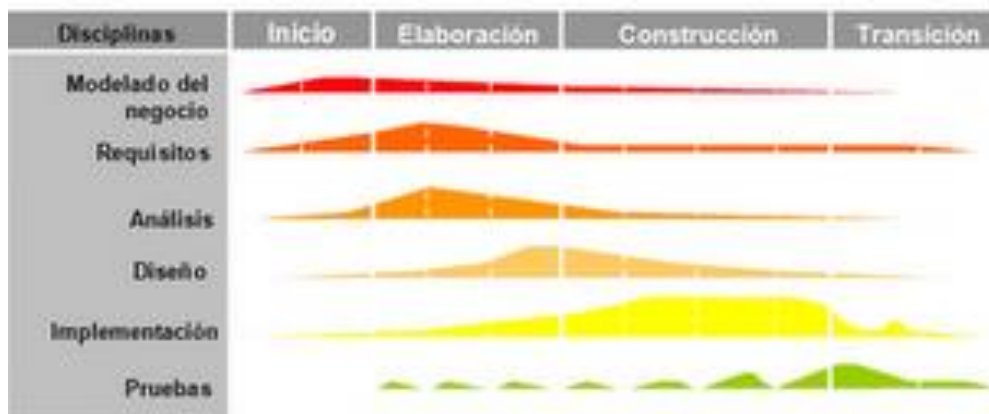


Figura 12. Fases del Proceso Unificado

6.1.1) Estimación de la duración del proyecto

La realización de la estimación del tiempo necesario para el desarrollo del proyecto ha sido llevada a cabo mediante el uso de la herramienta *EZEstimate*.

Para obtener dicha estimación hay que introducir en dicha aplicación los factores influyentes en el desarrollo (complejidad técnica y entorno) y la valoración de la complejidad de los actores y requisitos funcionales.

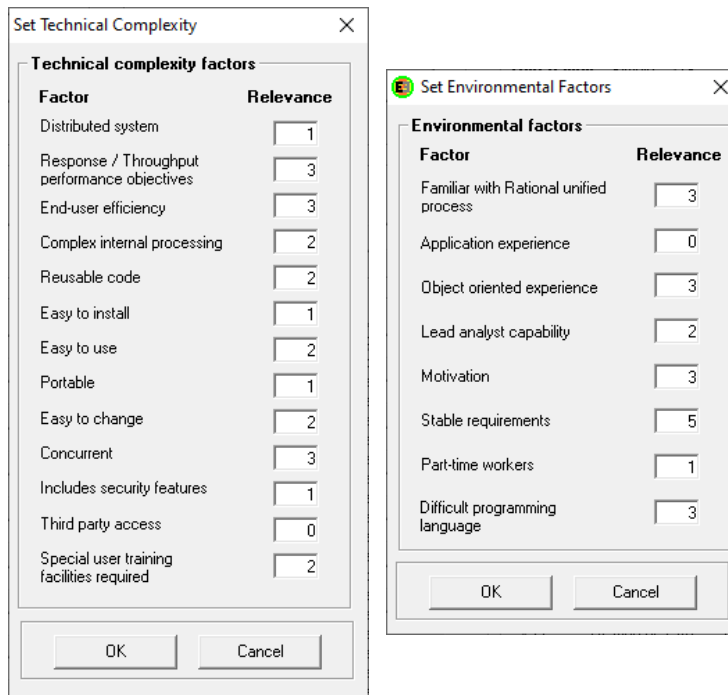


Figura 13. Introducción de factores en EZEstimate

Tras la introducción de la información acerca de los actores y los requisitos funcionales se obtiene la estimación temporal.

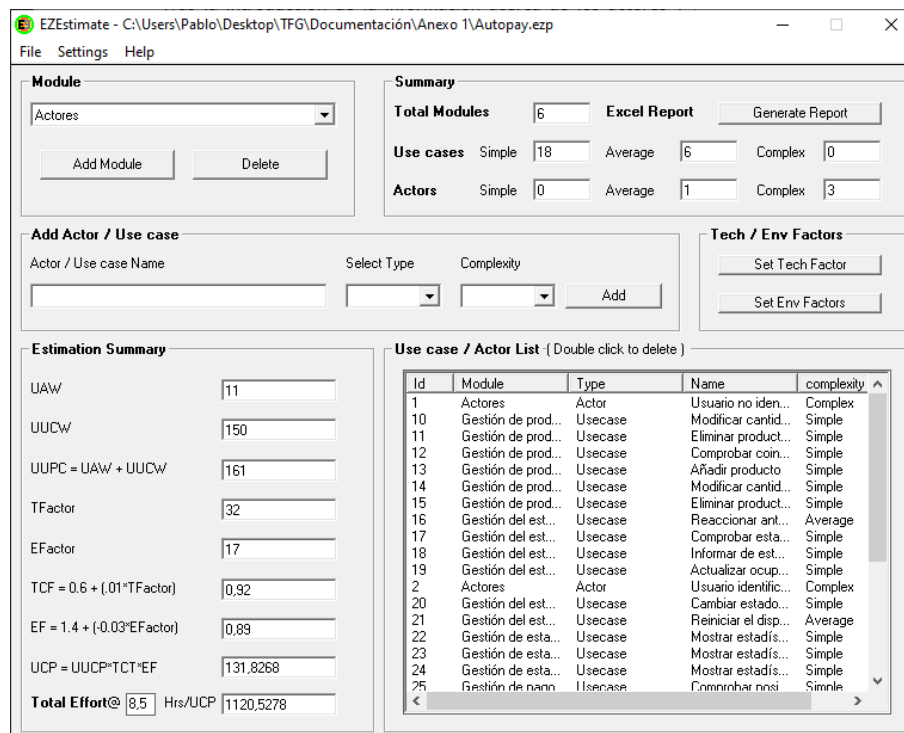


Figura 14. Estimación temporal de EZEstimate

La estimación temporal obtenida es de aproximadamente 1120 horas de trabajo. Si se trabajan 8 horas diarias el proyecto tardaría en realizarse 140 días, lo que equivale a aproximadamente 4 meses y 20 días laborables, equivalentes a 5 meses y 10 días naturales.

6.1.2) Planificación temporal

El desarrollo del sistema se ha dividido en las diferentes fases especificadas por el marco de trabajo del Proceso Unificado expuestas en la “Figura 12. Fases del Proceso Unificado”.

Al ser un proceso incremental e iterativo cada fase se compone de un número de iteraciones a realizar. Para el desarrollo de este proyecto se han asignado a cada fase el siguiente número de iteraciones:

- 1) Inicio: 2 iteraciones
- 2) Elaboración: 5 iteraciones
- 3) Construcción: 4 iteraciones
- 4) Transición: 2 iteraciones

A continuación, se muestra la vista global de la planificación temporal diseñada para el desarrollo del proyecto:

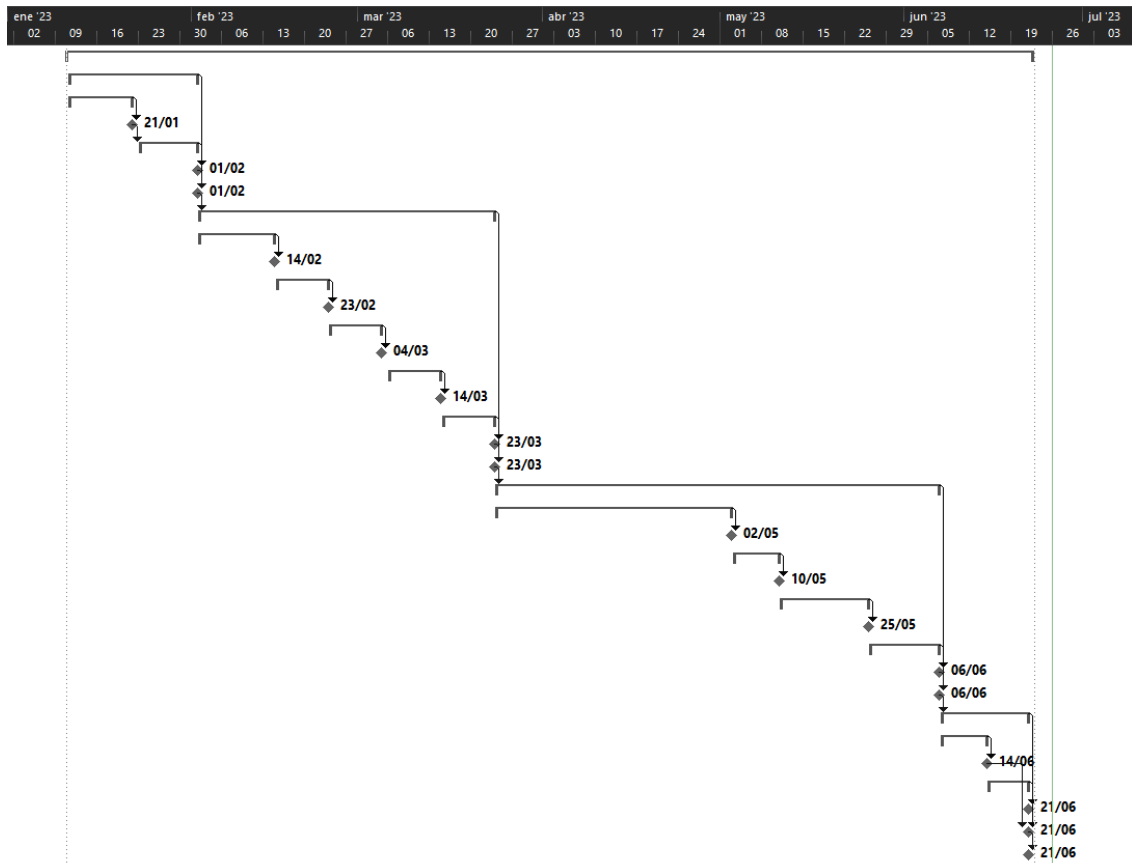


Figura 15. Diagrama de Gantt de la vista global de la planificación temporal

Planificación temporal	140 días	mié 11/01/23	jue 22/06/23	
Inicio	19 días?	mié 11/01/23	mié 01/02/23	
Iteración I	10 días?	mié 11/01/23	sáb 21/01/23	
Fin iteración I	0 días	sáb 21/01/23	sáb 21/01/23	3;16
Iteración II	9 días?	lun 23/01/23	mié 01/02/23	17
Fin iteración II	0 días	mié 01/02/23	mié 01/02/23	30;18
Fin fase de Inicio	0 días	mié 01/02/23	mié 01/02/23	31;2
Elaboración	43 días?	jue 02/02/23	jue 23/03/23	32
Iteración I	11 días?	jue 02/02/23	mar 14/02/23	
Fin iteración I	0 días	mar 14/02/23	mar 14/02/23	47;34
Iteración II	8 días?	mié 15/02/23	jue 23/02/23	
Fin iteración II	0 días	jue 23/02/23	jue 23/02/23	61;49
Iteración III	8 días?	vie 24/02/23	sáb 04/03/23	
Fin iteración III	0 días	sáb 04/03/23	sáb 04/03/23	75;63
Iteración IV	8 días?	lun 06/03/23	mar 14/03/23	
Fin iteración IV	0 días	mar 14/03/23	mar 14/03/23	77;89
Iteración V	8 días?	mié 15/03/23	jue 23/03/23	
Fin iteración V	0 días	jue 23/03/23	jue 23/03/23	103;91
Fin fase de elaboración	0 días	jue 23/03/23	jue 23/03/23	104;33
Construcción	64 días?	vie 24/03/23	mar 06/06/23	105
Iteración I	34 días?	vie 24/03/23	mar 02/05/23	
Fin iteración I	0 días	mar 02/05/23	mar 02/05/23	107;122
Iteración II	7 días?	mié 03/05/23	mié 10/05/23	
Fin iteración II	0 días	mié 10/05/23	mié 10/05/23	124;130
Iteración III	13 días?	jue 11/05/23	jue 25/05/23	
Fin iteración III	0 días	jue 25/05/23	jue 25/05/23	132;144
Iteración IV	10 días?	vie 26/05/23	mar 06/06/23	
Fin iteración IV	0 días	mar 06/06/23	mar 06/06/23	146;154
Fin de fase de construcción	0 días	mar 06/06/23	mar 06/06/23	106;155
Transición	13 días?	mié 07/06/23	mié 21/06/23	156
Iteración I	7 días?	mié 07/06/23	mié 14/06/23	
Fin de Iteración I	0 días	mié 14/06/23	mié 14/06/23	158;168
Iteración II	6 días	jue 15/06/23	mié 21/06/23	
Fin de Iteración II	0 días	mié 21/06/23	mié 21/06/23	174;170
Fin fase de transición	0 días	mié 21/06/23	mié 21/06/23	169;157
Fin planificación temporal	0 días	mié 21/06/23	mié 21/06/23	176

Figura 16. Vista global de la planificación temporal

Para una mayor profundización en la planificación y gestión del proyecto consúltese el documento “Anexo I: Plan del proyecto software”.

6.2) Elicitación de requisitos

Para realizar de forma metódica el proceso de ingeniería software se ha seguido el modelo establecido por el Proceso Unificado, empleando la metodología Durán y Bernárdez^[8] para la elicitación de requisitos del sistema.

6.2.1) Organización y autores

Este proyecto ha sido desarrollado por el autor:

Pablo Santos Blázquez

y supervisado por los tutores:

Sergio García González

Gabriel Villarrubia González

Todos los participantes del mismo pertenecen a la Universidad de Salamanca (USAL).

6.2.2) Objetivos del sistema

Los objetivos descritos anteriormente en el apartado “2.1) *Objetivos del sistema*” han sido estudiados y definidos con mayor precisión mediante el uso de tablas como la mostrada a continuación:

OBJ - 01	Identificación de usuarios
Versión	1.0
Autores	Santos Blázquez, Pablo
Fuentes	García González, Sergio Villarrubia González, Gabriel
Descripción	El sistema deberá ser capaz de identificar los usuarios que hagan uso de él, distinguiendo entre clientes (usuarios estándar) y administradores. Todos los usuarios deberán de verificar su identidad mediante el uso de una clave privada.
Subobjetivos	-
Importancia	Vital
Urgencia	Inmediatamente
Estado	Validado
Estabilidad	Alta

Tabla 1. Ejemplo de objetivo: OBJ-01: Identificación de usuarios

6.2.3) Requisitos de información

A continuación, se exponen los requisitos de información necesarios para el sistema. Estos muestran los datos que deben ser almacenados en el sistema.

- IRQ-01: Información sobre usuarios.
- IRQ-02: Información sobre carros.
- IRQ-03: Información sobre productos.
- IRQ-04: Información sobre productos vendidos.
- IRQ-05: Información sobre ventas.

Para describir dichos requisitos de información se han elaborado tablas como la mostrada a continuación:

IRQ - 01	Información sobre usuarios	
Versión	1.0	
Autores	Santos Blázquez, Pablo	
Fuentes	García González, Sergio Villarrubia González, Gabriel	
Objetivos asociados	• OBJ-01: Identificación de usuarios	
Descripción	El sistema deberá ser capaz de almacenar información sobre los usuarios registrados en el sistema. En concreto:	
Datos específicos	<ul style="list-style-type: none"> • Identificador • Pin de acceso • Nombre • Apellidos • Forma de pago 	
Tiempo de vida	Medio	Máximo
	-	-
Ocurrencias simultáneas	Medio	Máximo
	-	-
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	
Comentarios	-	

Tabla 2. Ejemplo de requisito de información: IRQ- 01: Información sobre usuarios

6.2.4) Requisitos funcionales

Los requisitos funcionales describen los servicios ofrecidos por el sistema y el comportamiento que debe adoptar el mismo para cumplir los objetivos.

Para mostrar la división del sistema en elementos más pequeños (subsistemas) durante la fase de dominio del problema se ha elaborado un diagrama de paquetes.

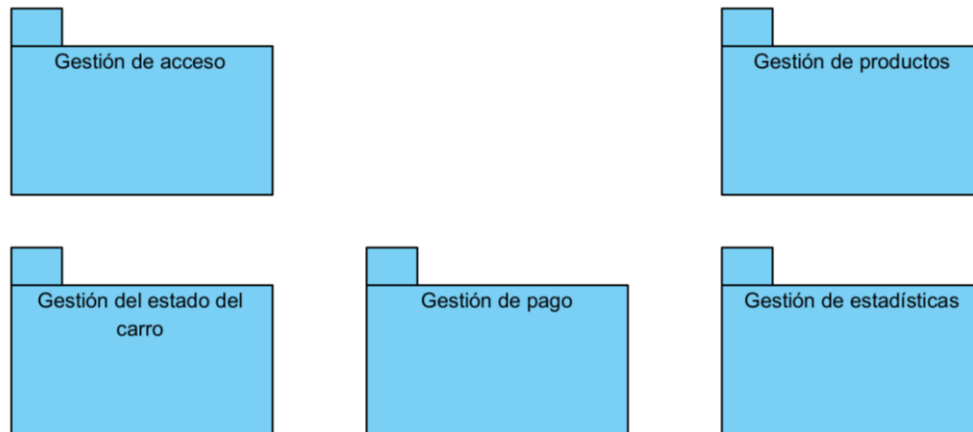


Figura 17. Diagrama de paquetes

Una vez obtenida una visión más definida del sistema, se han descrito los distintos tipos de usuarios que interactúan con el mismo:



Figura 18. Actores del sistema

Para describir de forma más detallada a los distintos tipos de actores se han elaborado tablas como la mostrada a continuación:

ACT - 01	Usuario no identificado
Versión	1.0
Autores	Santos Blázquez, Pablo
Fuentes	García González, Sergio Villarrubia González, Gabriel
Descripción	Usuario que interacciona con el sistema sin haber sido identificado.
Comentarios	-

Tabla 3. Ejemplo de actor: ACT- 01: Usuario no identificado

A continuación, se exponen las funcionalidades que componen el sistema, especificando las interacciones llevadas a cabo por los distintos tipos de usuarios.

Para obtener una visión global del sistema se ha realizado un diagrama de casos de uso, el cual agrupa y expone todas las funcionalidades del sistema de forma visual.

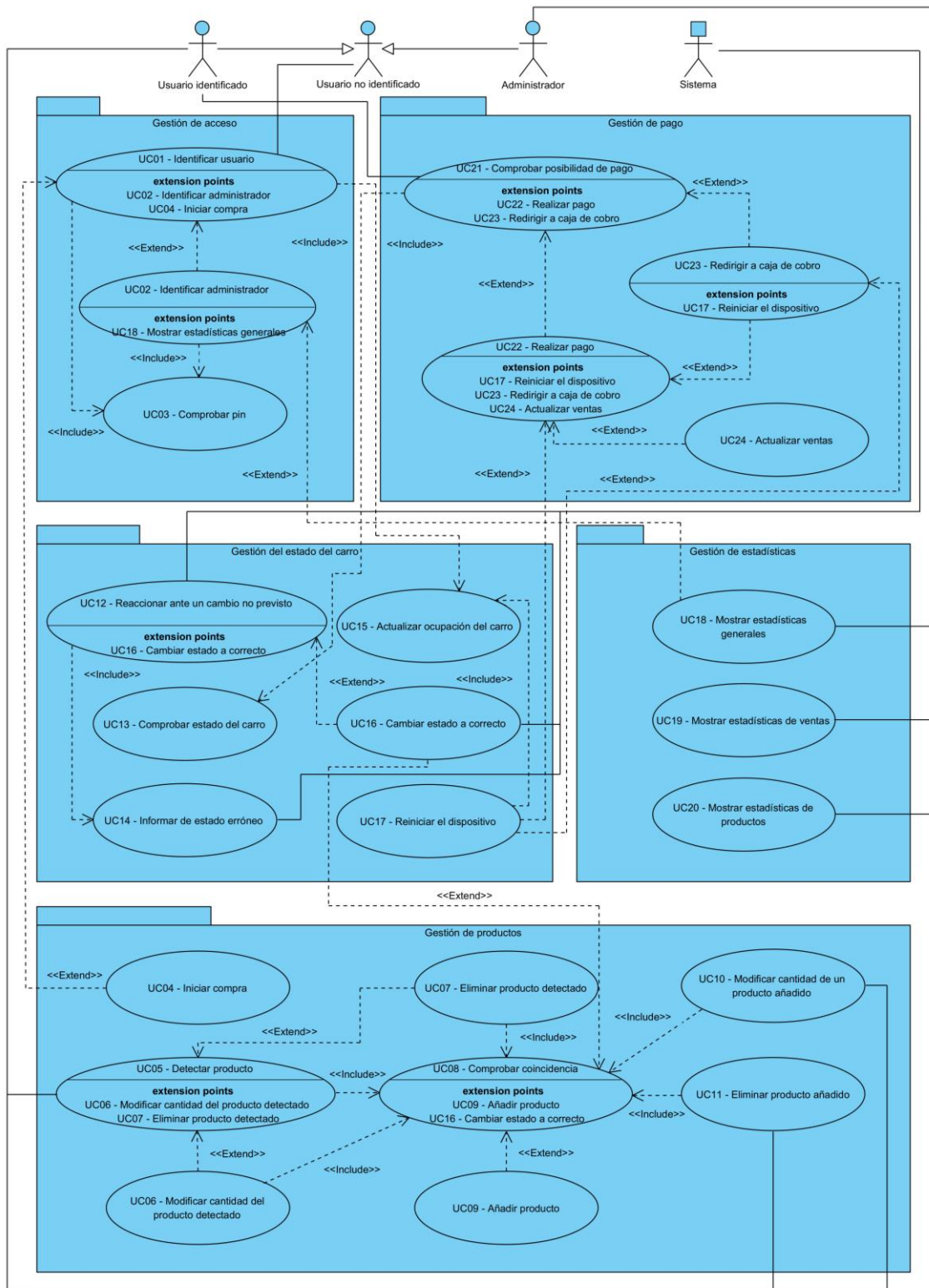


Figura 19. Diagrama de casos de uso

Para exponer de forma más concreta cada uno de los casos de uso, se ha realizado un diagrama centrado en cada uno de los distintos subsistemas:

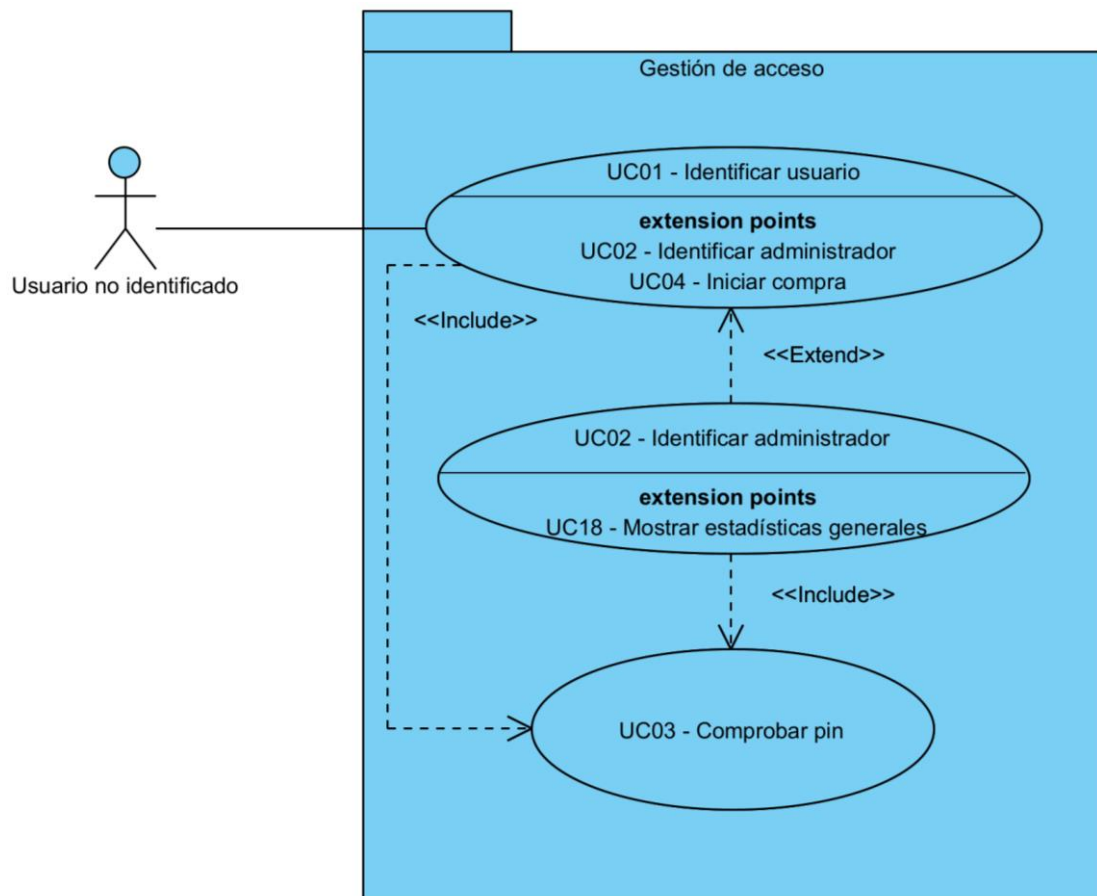


Figura 20. Ejemplo de diagrama de casos de uso distribuidos por subsistemas

Se ha especificado la realización de cada uno de los casos de uso mediante el desarrollo de tablas como la mostrada a continuación:

UC - 01	Identificar usuario
Versión	1.0
Autores	Santos Blázquez, Pablo
Fuentes	García González, Sergio Villarrubia González, Gabriel
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Identificación de usuarios • OBJ-02: Seguimiento de productos • OBJ-03: Monitorización del estado del dispositivo
Requisitos asociados	<ul style="list-style-type: none"> • UC-02: Identificar administrador • UC-03: Comprobar pin • UC-04: Iniciar compra • UC-15: Actualizar ocupación del carro
Descripción	El sistema debe ser capaz de realizar el inicio de sesión de los usuarios
Precondición	El usuario no está identificado

Secuencia normal	Paso	Acción
	1	El usuario indica al sistema que quiere comenzar el proceso de inicio de sesión y se realiza el caso de uso UC-15: Actualizar ocupación del carro
	2	El sistema informa al usuario de que debe escanear su código QR
	3	El usuario escanea su código QR
	4	El sistema realiza una petición a la base de datos para obtener la información sobre el usuario
	5	El sistema realiza el caso de uso UC-03: Comprobar pin
	6	El sistema realiza el caso de uso UC-04: Iniciar compra
Postcondición	El usuario está identificado	
Excepciones	Paso	Acción
	4	Si el QR no corresponde a ningún usuario existente la ejecución vuelve al paso 2.
	4	Si el QR se corresponde con el de un administrador, se realizará el caso de uso UC-02: Identificar administrador, abortando el actual
	5	Si al realizar el caso de uso UC-03: Comprobar pin este genera un resultado negativo, se informa al usuario y se aborta la ejecución del caso de uso
Frecuencia esperada	750 veces/día	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	Validado	
Estabilidad	Alta	
Comentarios	-	

Tabla 4. Ejemplo de realización de caso de uso: UC-01: Identificar usuario

6.2.5) Requisitos no funcionales

A continuación, se exponen los requisitos no funcionales que imponen las restricciones para el diseño e implementación del sistema y los estándares de calidad.

- NFR-01: Precisión.
- NFR-02: Adaptabilidad.
- NFR-03: Usabilidad.
- NFR-04: Manejabilidad.

Para describir dichos requisitos no funcionales se han elaborado tablas como la mostrada a continuación:

NFR - 01	Precisión
Versión	1.0
Autores	Santos Blázquez, Pablo
Fuentes	García González, Sergio Villarrubia González, Gabriel
Objetivos asociados	· OBJ-02: Seguimiento de productos
Descripción	El sistema debe ser capaz de medir los cambios realizados en su entorno con gran precisión
Importancia	Vital
Urgencia	Inmediata
Estado	Validado
Estabilidad	Alta
Comentarios	Se admite para el sensor de peso un margen de error en la medida de hasta ± 5 gramos

Tabla 5. Ejemplo de requisito no funcional: NFR-01: Precisión

Para una mayor profundización en la elicitación de requisitos consúltese el documento “Anexo II: Especificación de requisitos software”.

6.3) Análisis de requisitos

En este apartado se describe el modelo del dominio del sistema software.

El modelo del dominio es una representación conceptual de los principales conceptos, entidades y relaciones de un sistema. Ayuda a comprender los requisitos y características del software y recoge las necesidades de almacenamiento e información.

Para representar el modelo del dominio del sistema se ha desarrollado un diagrama de clases:

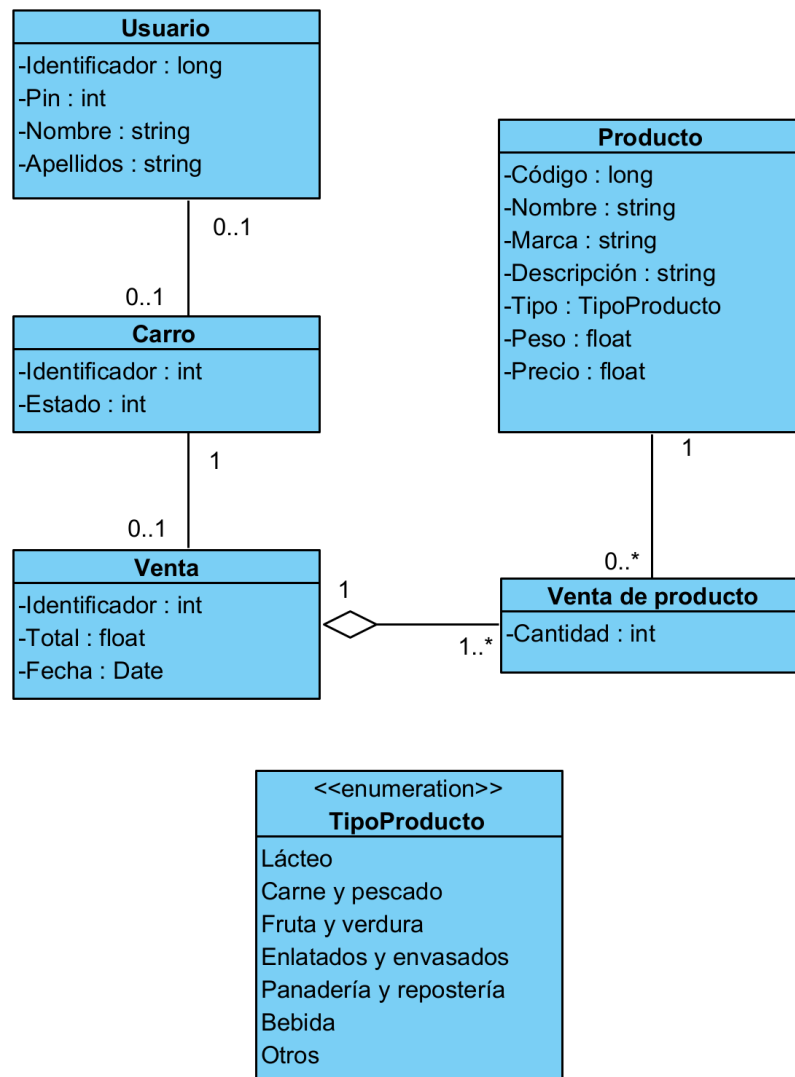


Figura 21. Modelo del dominio

6.3.1) Diagramas de secuencia

Para la comprensión visual de las interacciones realizadas entre los distintos actores y el sistema se ha realizado un diagrama de secuencia para cada uno de los diferentes casos de uso desarrollados

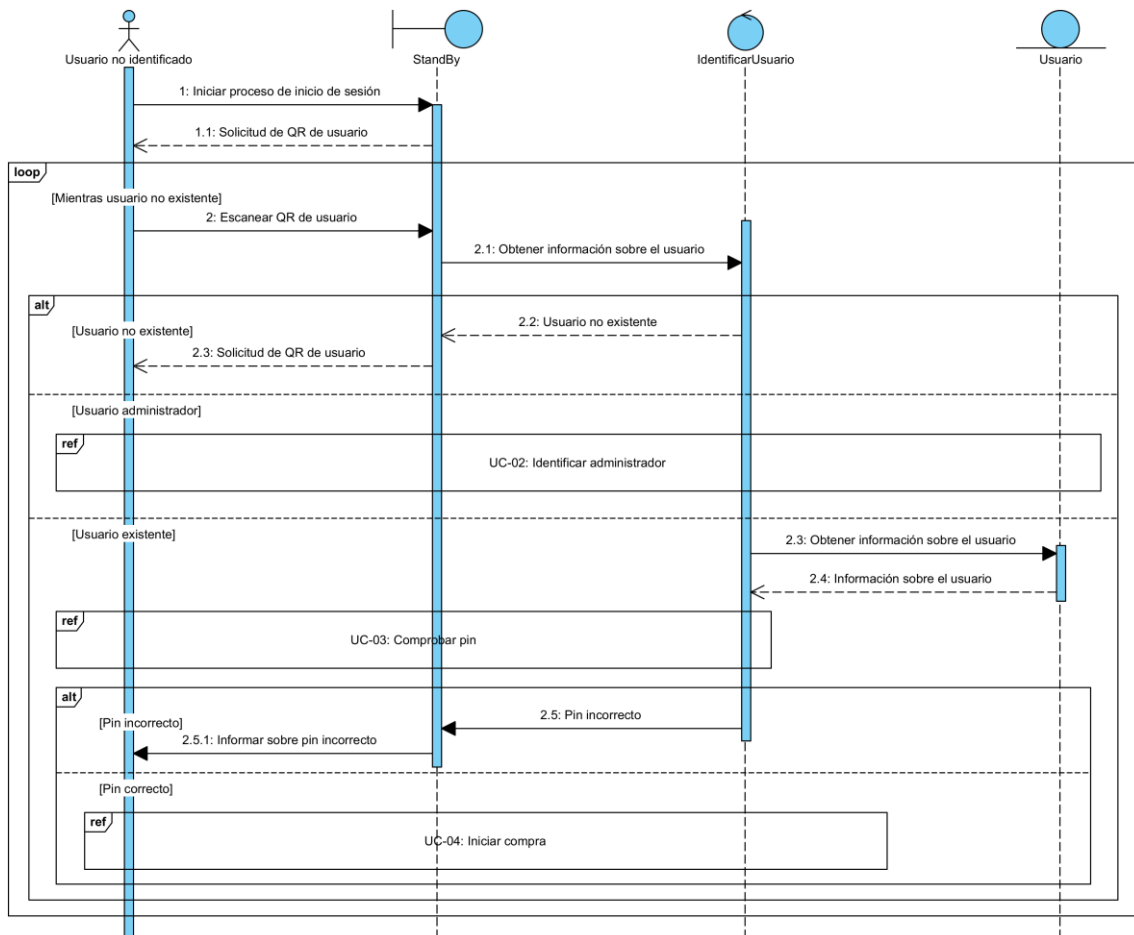


Figura 22. Ejemplo de diagrama de secuencia: UC-01: Identificar usuario

6.3.2) Arquitectura del sistema

Para mostrar de forma visual las interacciones llevadas a cabo entre las diferentes clases de análisis, se ha desarrollado una propuesta de arquitectura del sistema. Con el desarrollo de este diagrama se pretende distribuir las diferentes clases del análisis en sus correspondientes capas.

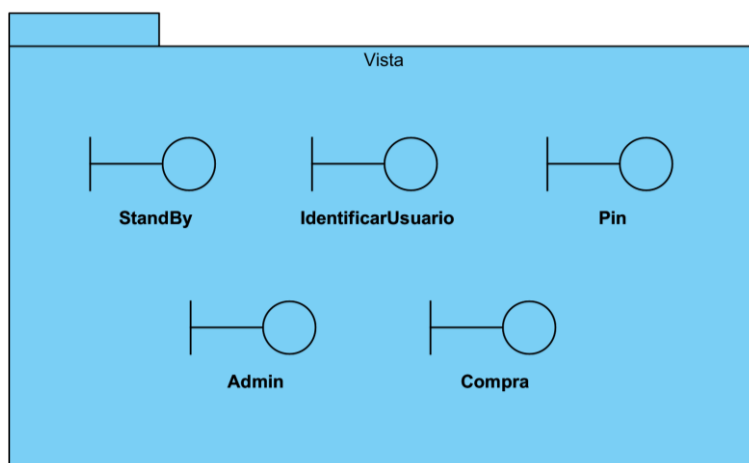


Figura 23. Ejemplo de clase de análisis: Vista

A continuación, se expone la propuesta de arquitectura desarrollada para el sistema.

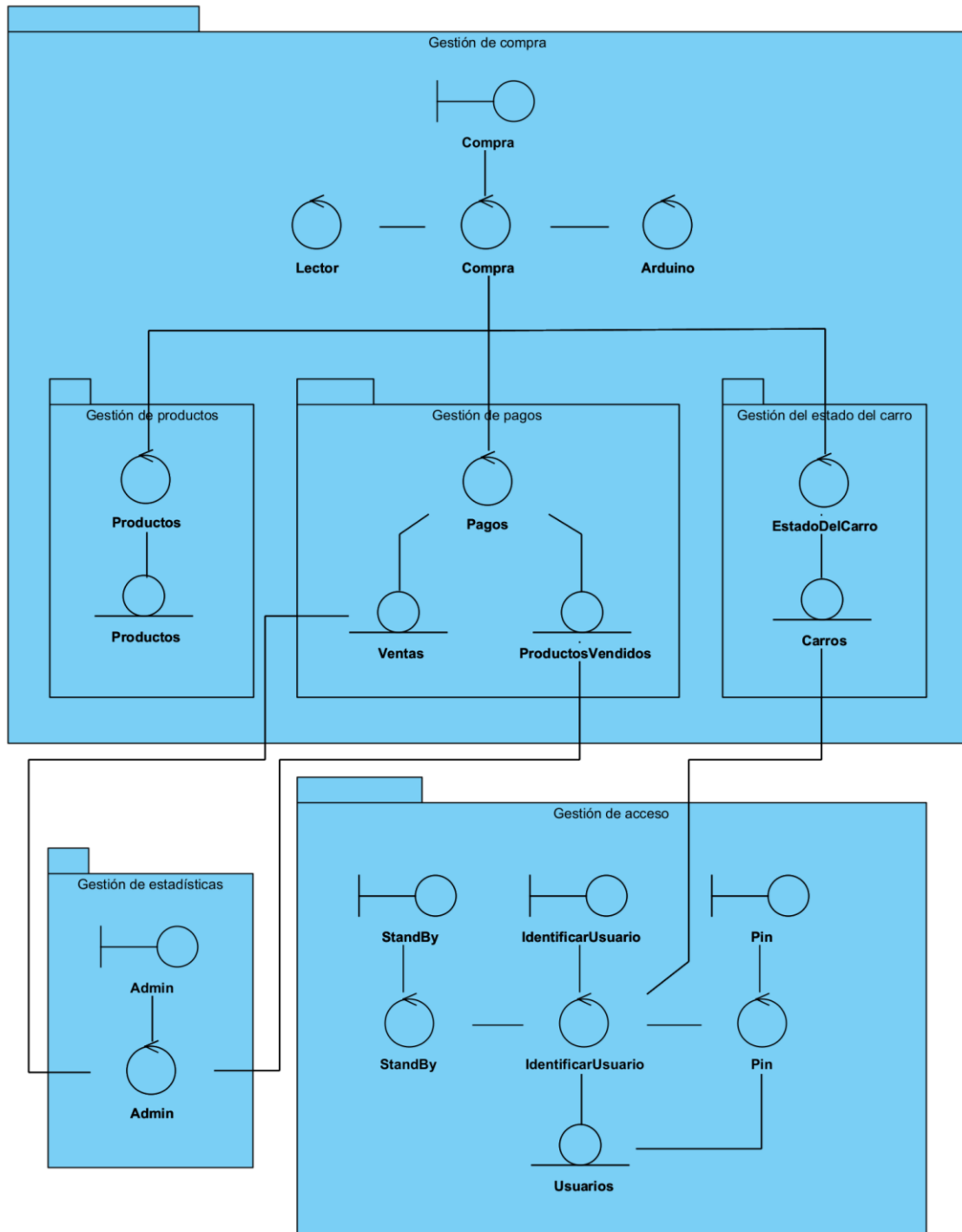


Figura 24. Propuesta de arquitectura

Para una mayor profundización en el análisis de requisitos consúltese el documento “Anexo III: Análisis de requisitos software”.

6.4) Diseño del sistema

Tras el análisis de requisitos se ha desarrollado el modelo de diseño, el cual describe la realización física de los casos de uso, teniendo en cuenta tanto los requisitos funcionales como los no funcionales y las posibles limitaciones que pueda tener la implementación del sistema.

6.4.1) Patrón arquitectónico

El patrón MVVM^[9] es el patrón por defecto recomendado por Microsoft para el desarrollo de aplicaciones WPF en la plataforma .NET. Esto se debe a que WPF está diseñado orientado al uso de este patrón, lo que facilita su implementación y aprovecha las funcionalidades proporcionadas por la tecnología .NET de manera más eficiente; lo que resulta en un desarrollo más rápido y consistente.

El patrón Modelo-Vista-Modelo de vista (MVVM) ayuda a separar limpiamente la lógica de presentación y de negocios de una aplicación de su interfaz gráfica., lo que ayuda a evitar numerosos problemas de desarrollo y facilita la prueba, el mantenimiento y la evolución del sistema.

Con el patrón MVVM, la interfaz de usuario de la aplicación y la lógica de presentación y negocios subyacente se separan en tres clases distintas:

- **La vista:** encapsula la lógica de la interfaz de usuario y la interfaz de usuario.
- **El modelo de la vista:** encapsula la lógica y el estado de la presentación
- **El modelo:** encapsula la lógica de negocios y los datos de la aplicación.

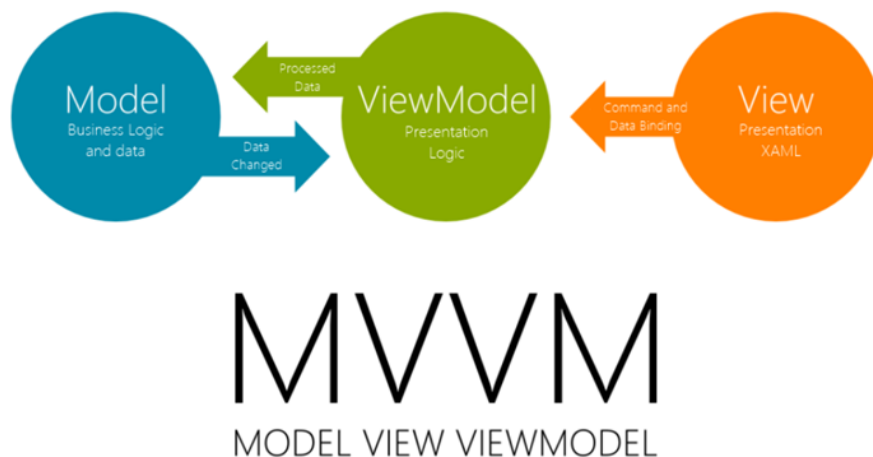


Figura 25. Arquitectura MVVM en proyectos .NET

6.4.2) Subsistemas de diseño

Para facilitar la comprensión del diseño se ha dividido el sistema en subsistemas más específicos denominados subsistemas de diseño.

La principal división se ha realizado entre la aplicación WPF, el lector de códigos de barras y el Arduino, el cual controla el resto de los dispositivos vinculados al sistema.

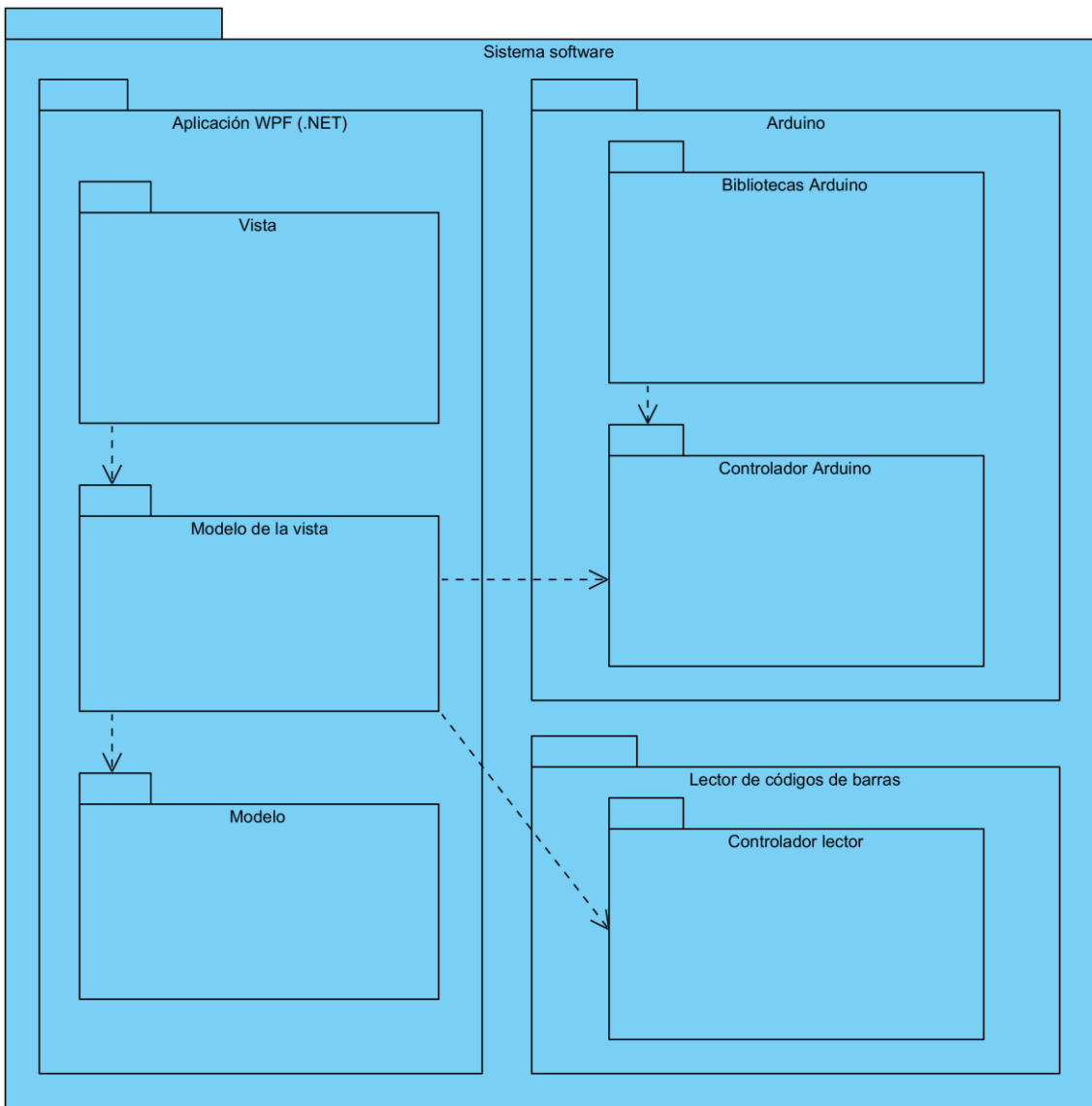


Figura 26. Diagrama de subsistemas de diseño

Tras la elaboración de este diagrama se han desarrollado las diferentes clases de diseño, las cuales se exponen dentro del contexto de su subsistema para facilitar su estructuración y comprensión.

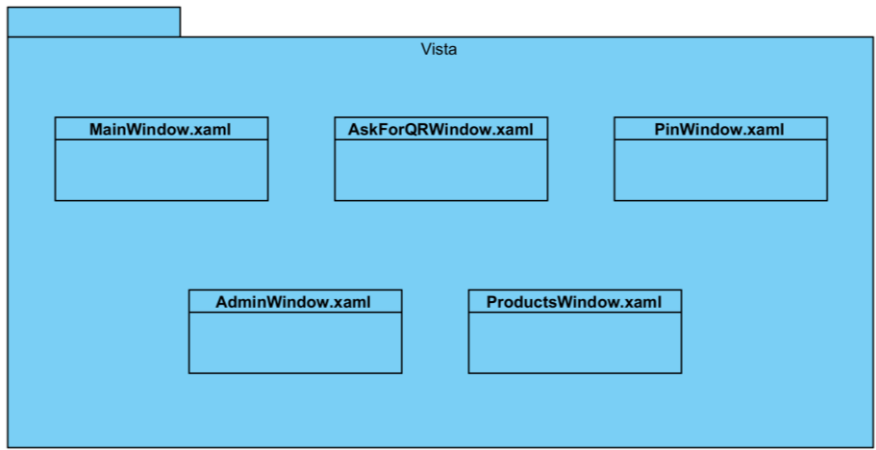


Figura 27. Ejemplo de clases de diseño: Vista

6.4.3) Vista arquitectónica

En ese apartado se expone como recapitulación un diagrama representativo de la vista arquitectónica, la cual permite la visión global de los subsistemas completos que conforman el sistema software.

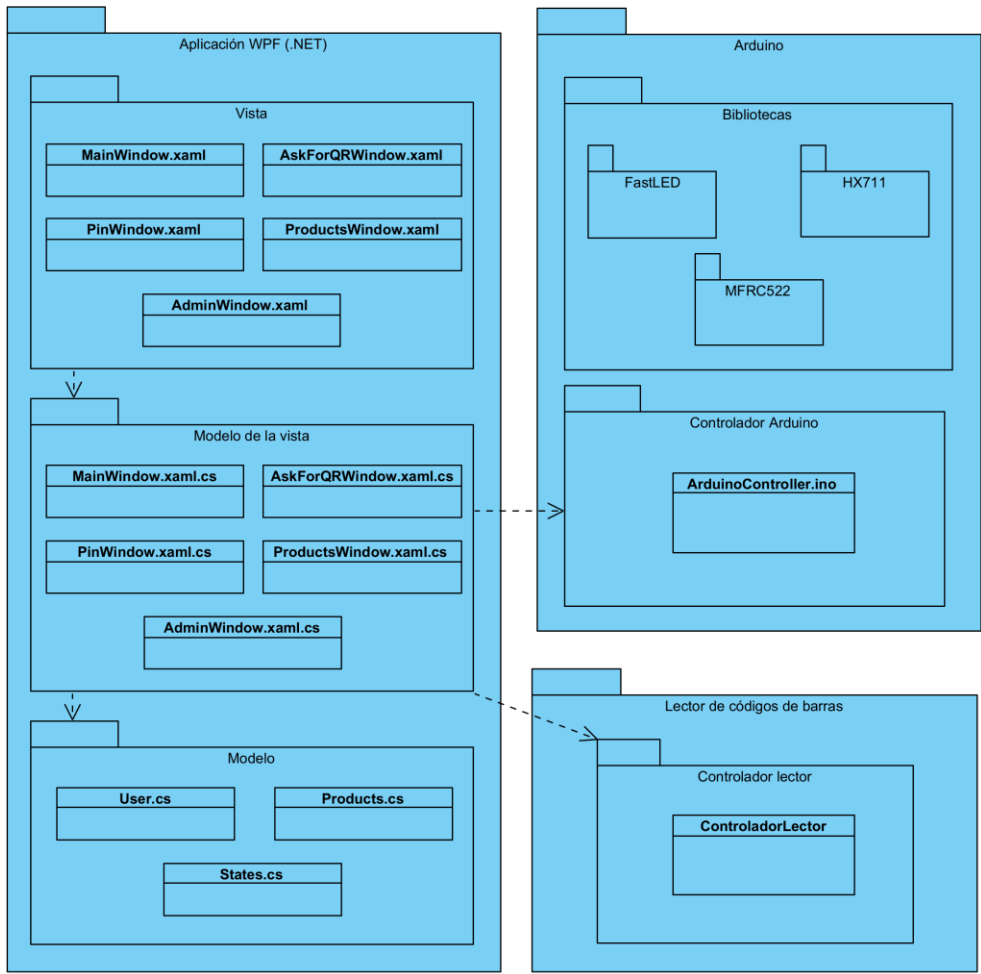


Figura 28. Vista arquitectónica

6.4.4) Realización de casos de uso

En este apartado se expone la realización física de los casos de uso mediante el desarrollo de diagramas de secuencia a partir de los realizados durante la fase de análisis de requisitos desarrollada en el documento “Anexo II: Especificación de requisitos software”.

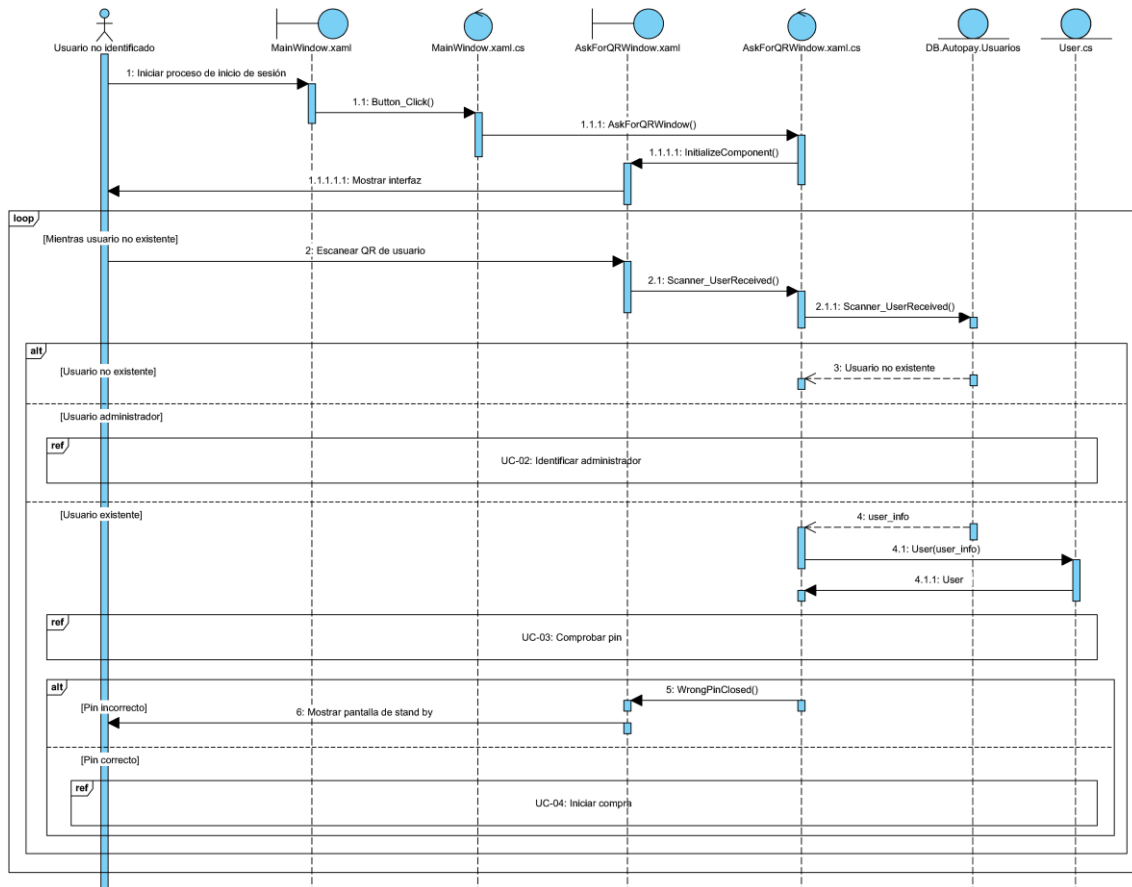


Figura 29. Ejemplo de realización de caso de uso: UC-01: Identificar usuario

6.4.5) Diseño de la base de datos

En este apartado se expone la estructura de la base de datos creada para el almacenamiento y gestión de los requisitos de información. Para la mayor seguridad, optimización y rapidez, se ha optado por la implementación de una base de datos relacional basada en MySQL.

A continuación, se muestra el modelo de diseño de la base de datos desarrollada:

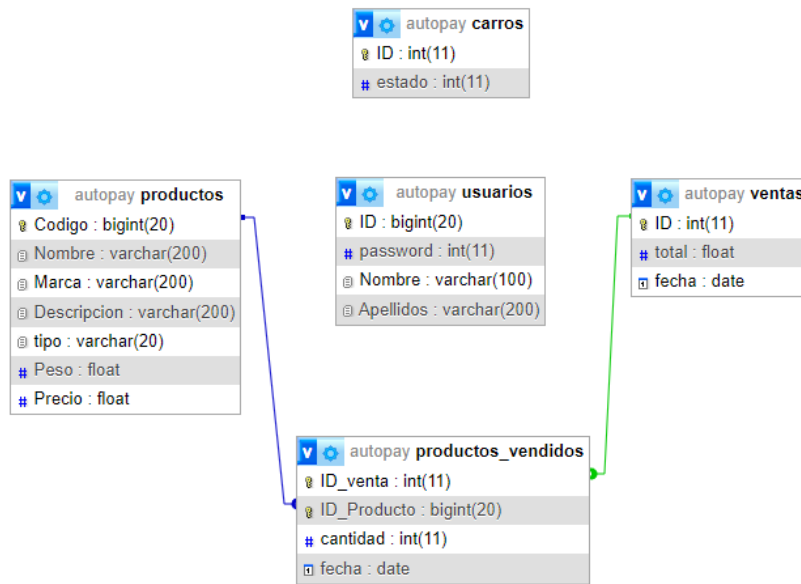


Figura 30. Modelo de diseño de la base de datos

6.4.6) Modelo de despliegue

Para concluir la etapa de diseño se ha desarrollado el diagrama de despliegue, el cual expone las relaciones finales entre los diferentes nodos distribuidos que componen el sistema.

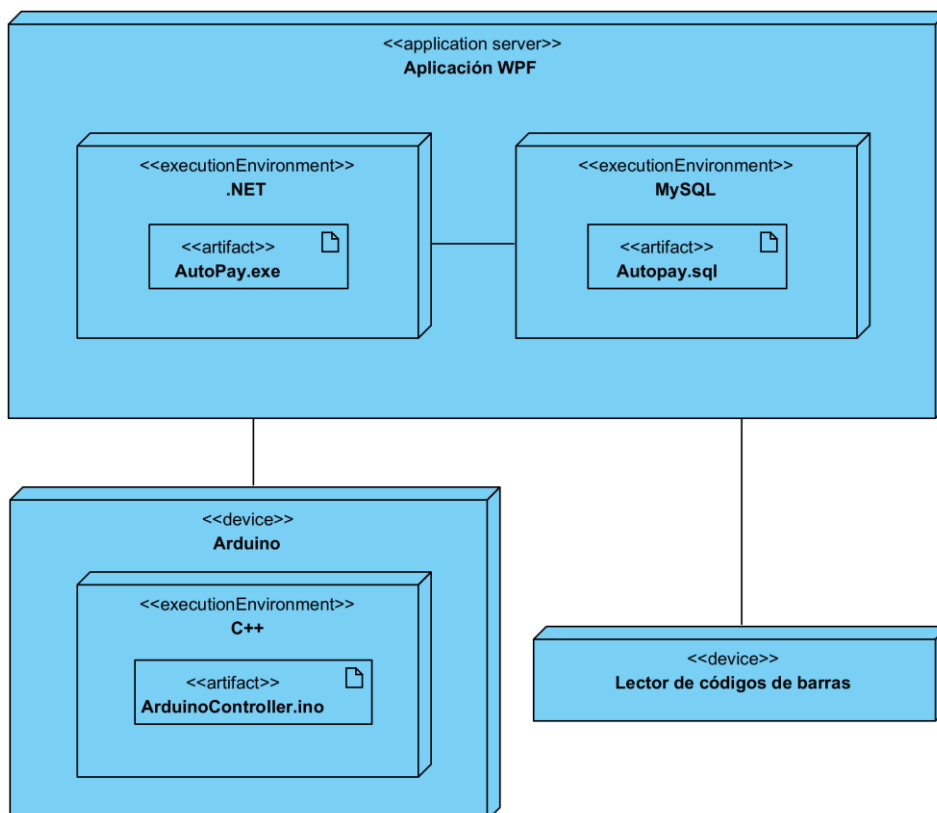


Figura 31. Diagrama de despliegue

Para una mayor profundización en el diseño del sistema consúltese el documento “Anexo IV: Diseño del sistema software”.

6.5) Implementación del sistema

Este apartado expone una de las etapas finales del desarrollo software, en la que se implementan de forma real los requisitos y objetivos especificados, analizados y diseñados. Esta fase abarca gran parte del desarrollo y ha puesto a prueba la capacidad práctica y la habilidad del uso de las herramientas de desarrollo anteriormente detalladas.

Durante esta etapa se ha implementado tanto el sistema software (aplicación WPF y controlador Arduino) como el dispositivo hardware.

La estructura general de los archivos entregados para el correcto funcionamiento del sistema es la siguiente:

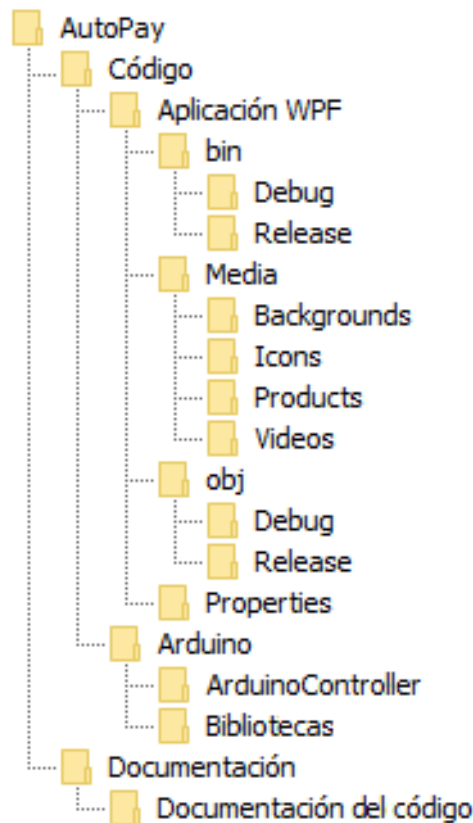


Figura 32. Estructura general del proyecto

Para una mayor profundización en el ámbito técnico del sistema consúltese el documento “Anexo V: Documentación técnica”.

6.5.1) Implementación software

Este apartado se centra en la exposición de los principales diseños y pantallas con las que interaccionan los usuarios para poder comprender el funcionamiento del sistema de forma sencilla.

6.5.1.1) Iconografía

El nombre elegido para la aplicación ha sido “AutoPay”, aunque el nombre del dispositivo final que incluye al sistema es “USuAL Shopping Cart”.



Figura 33. Iconografía del dispositivo final: USuAL Shopping Cart

Con este diseño se pretende alejarse de las diferentes marcas de supermercados o grandes empresas del sector retail en las que el dispositivo pueda llegar a ser incorporado, haciendo siempre referencia a su lugar de desarrollo, la Universidad de Salamanca.

Para mantener dicha conexión de forma visual se ha seguido el manual institucional de identidad corporativa^[10] de la Universidad de Salamanca. Este manual especifica que la tipografía utilizada debe ser la correspondiente a la familia tipográfica “Lato” y que los colores institucionales son el “Rojo Víctor”, “Gris Alumnio” y “Azul VIII Centenario”.

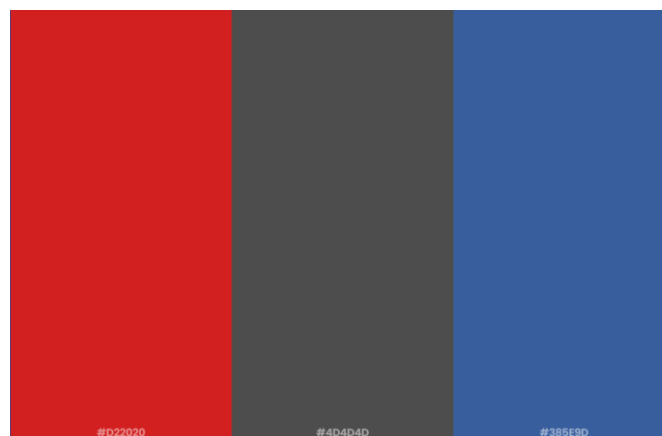


Figura 34. Colores institucionales de la Universidad de Salamanca

Estos colores han sido también utilizados en el diseño y desarrollo del dispositivo físico.

6.5.1.2) Ventanas

- Pantalla de standby

La pantalla de standby es mostrada por el sistema mientras este no es utilizado.

En ella se reproduce un video en bucle que muestra la iconografía diseñada para el sistema realizando una animación de llegada, frenada, arrancada y salida del carro que forma parte de dicho diseño.



Figura 35. Pantalla de standby

Para abandonar esta pantalla, un usuario debe indicar que quiere comenzar el proceso de identificación mediante una pulsación en cualquier parte de la misma. Tras la pulsación se mostrará la pantalla de identificación de usuarios.

- Pantalla de identificación de usuarios

La pantalla de identificación de usuarios se compone de varias subpantallas con distintas funcionalidades.

- **Petición de QR:**

En ella se pide al usuario que acerque su código QR de usuario al lector de códigos de barras.



Figura 36. Pantalla de petición de QR

Para abandonar esta pantalla, un usuario o administrador debe acercar un código QR válido.

Si el QR pertenece a un usuario normal del sistema se abrirá la pantalla para la introducción de su código pin.

- **Introducción del código pin:**

En ella se pide al usuario que introduzca su código pin para verificar su identidad. Se permitirá hasta un total de tres intentos por sesión.

Dependiendo de si el usuario es un usuario normal o un administrador se le muestra una pantalla u otra.

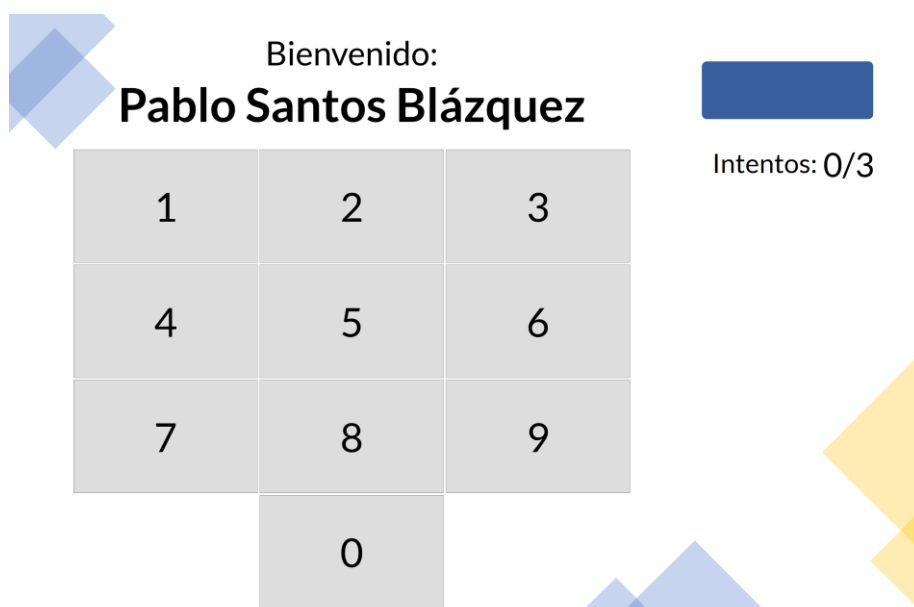


Figura 37. Pantalla de introducción del código pin para usuarios

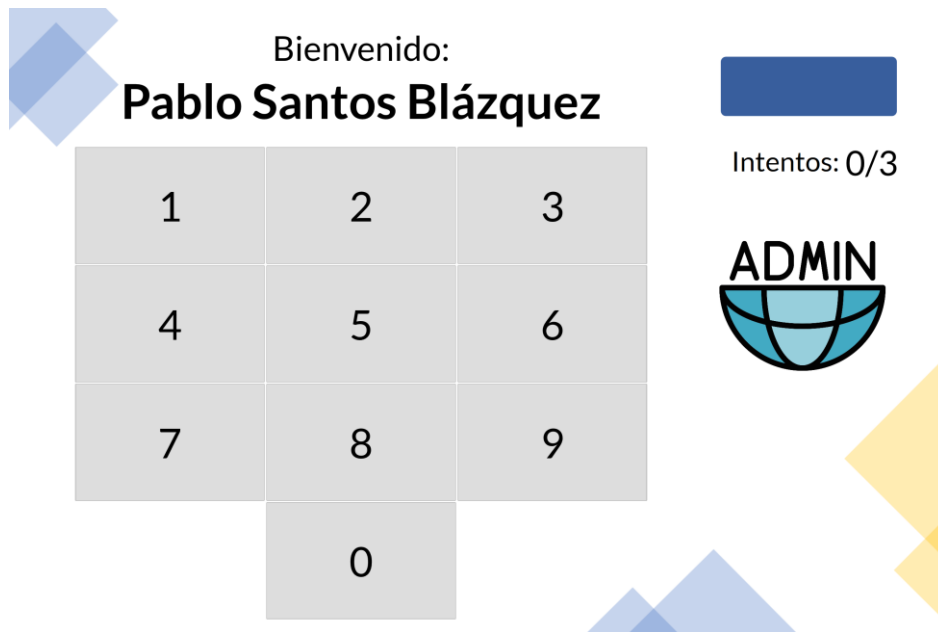


Figura 38. Pantalla de introducción del código pin para administradores

Con cada pulsación se introduce un número al pin y se muestra oculto en el recuadro azul mediante el carácter '*'.

Para abandonar esta pantalla, el usuario o administrador debe introducir su código pin de forma correcta. Una vez verificada la identidad del usuario, si es un usuario normal se abre la pantalla de compra, mientras que si se trata de un administrador se abre la pantalla de estadísticas.

Si tras el número máximo de intentos el usuario no consigue superar el proceso de verificación, se muestra un mensaje informando de la situación y de la redirección a la pantalla de standby, finalizando así su sesión.



Figura 39. Pantalla de introducción del código pin tras superar el máximo de intentos

- Pantalla de compra

La pantalla de compra es mostrada por el sistema cuando un usuario normal es correctamente identificado.

En ella se muestra un listado con los productos escaneados e introducidos en el dispositivo a la vez que el total de la compra.

Su estado inicial es el mostrado en la siguiente figura:

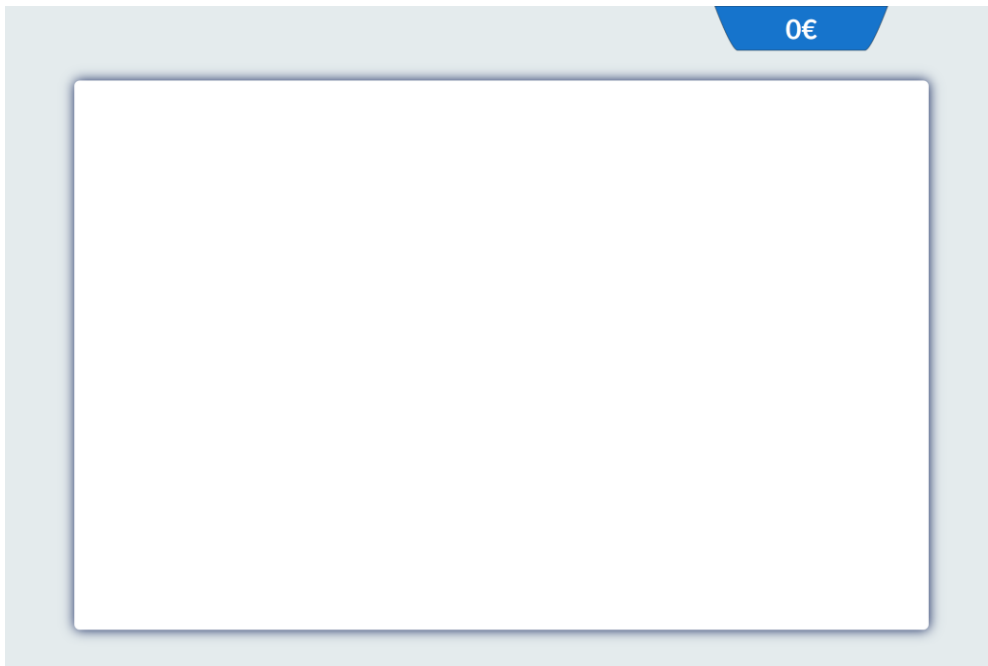


Figura 40. Pantalla de compra en su estado inicial

El color de fondo de la sección correspondiente al precio total de los productos añadidos se corresponde al mostrado por los leds del carro en todo momento, facilitando la comprensión del estado del carro para cualquier tipo de usuario.

Cuando el usuario acerca un producto al lector y este existe en la base de datos, sus datos son mostrados en la pantalla de detección de productos mostrada en la siguiente figura.

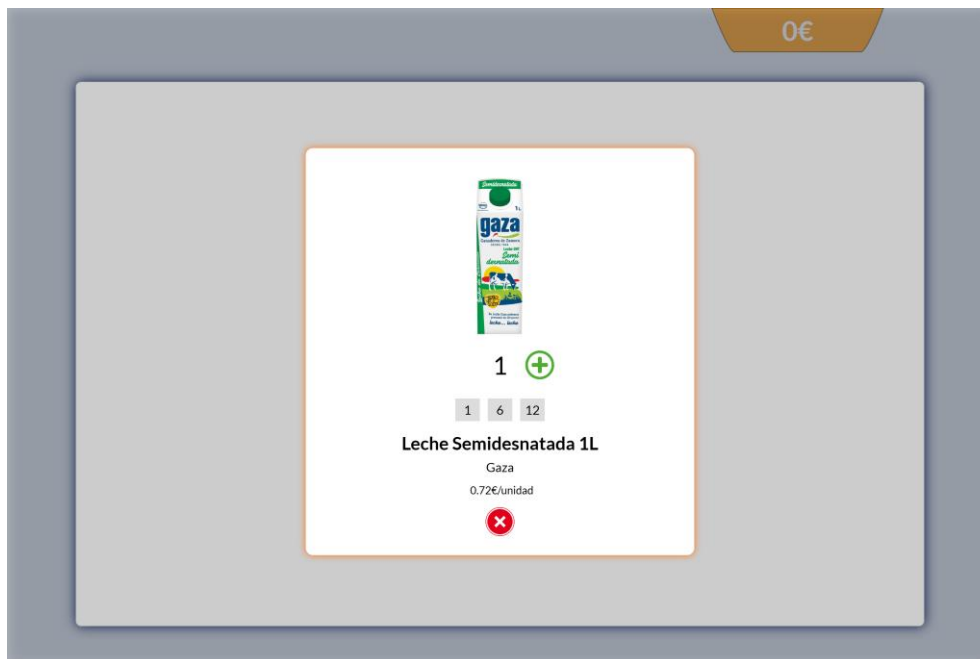


Figura 41. Pantalla de compra con un producto detectado

Desde esta pantalla se puede modificar la cantidad de unidades a introducir del producto, ya sea mediante los botones de añadir y sustraer o mediante los botones fijadores de cantidad de 1, 6 o 12 unidades. Desde la misma pantalla también se puede abortar el proceso de identificación del producto mediante el botón de cancelar, volviendo así a la pantalla de compra.

Una vez el sistema detecta que el producto introducido en el dispositivo se corresponde con el escaneado se añade al producto a la lista de la pantalla de compra:

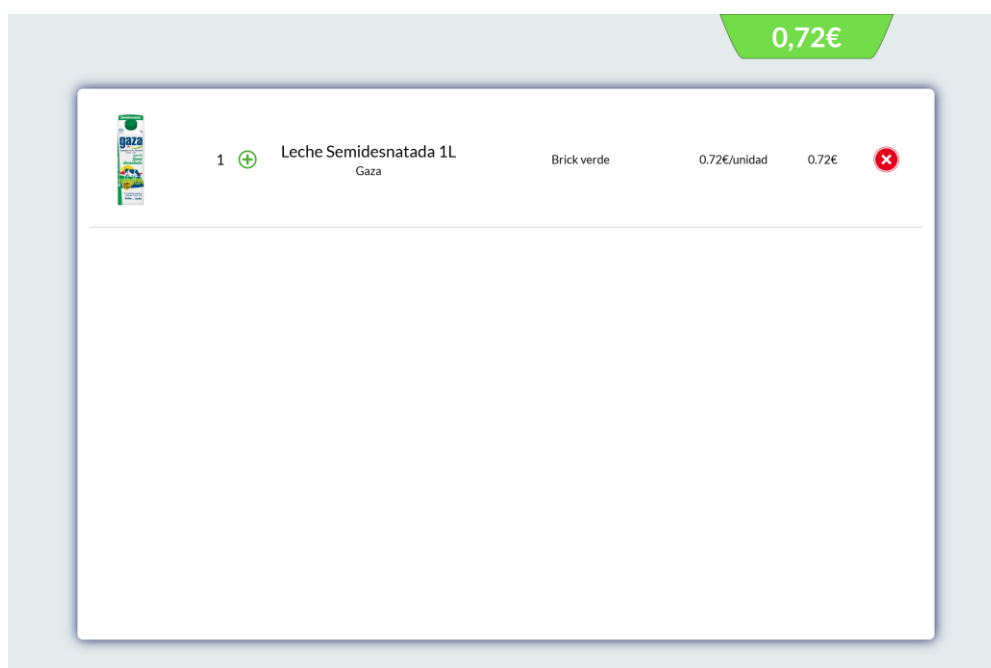


Figura 42. Pantalla de compra con un producto añadido

Desde la entrada añadida para cada producto también se puede modificar la cantidad introducida del mismo mediante el uso de los botones de añadir y sustraer. También se permite la opción de eliminar el producto cuando se pretende sacarlo del dispositivo.

Cuando se modifica la cantidad de un producto añadido, se informa al usuario de que debe asegurarse de que los productos dentro del dispositivo deben coincidir con los indicados en el listado.

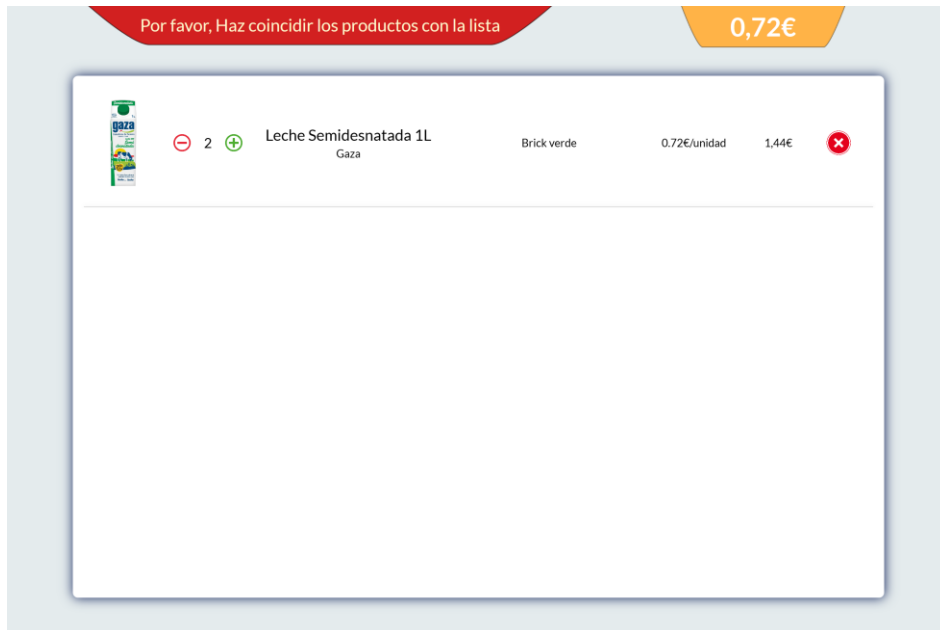


Figura 43. Pantalla de compra durante la modificación de la cantidad de un producto añadido

Una vez el sistema detecte que los productos se corresponden con la modificación especificada el dispositivo vuelve al estado correcto.

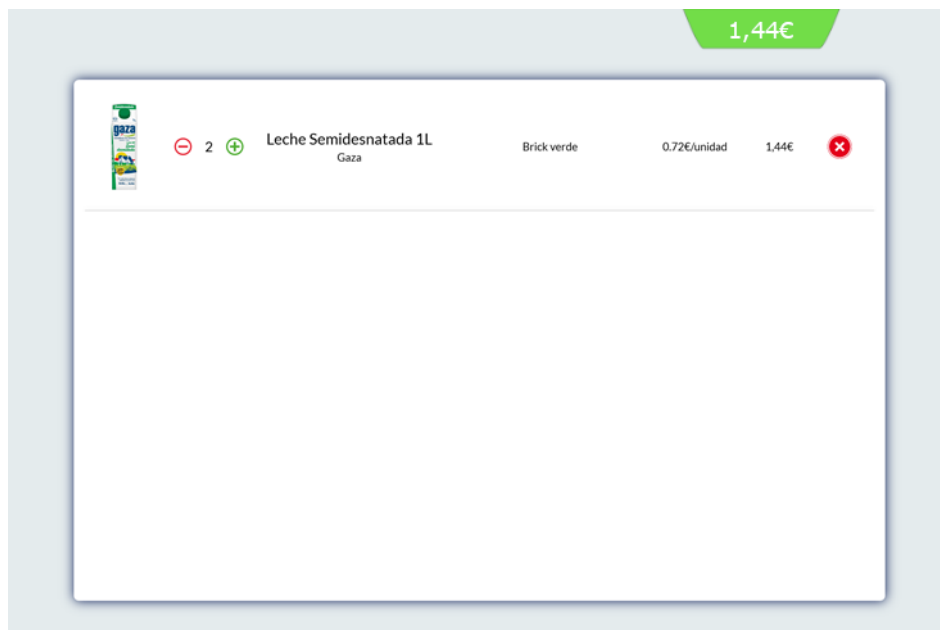


Figura 44. Pantalla de compra tras la modificación de la cantidad de un producto añadido

La lista se incrementa con cada producto escaneado que es detectado como añadido de forma correcta mediante la adición de una nueva entrada en la parte más alta de la misma.

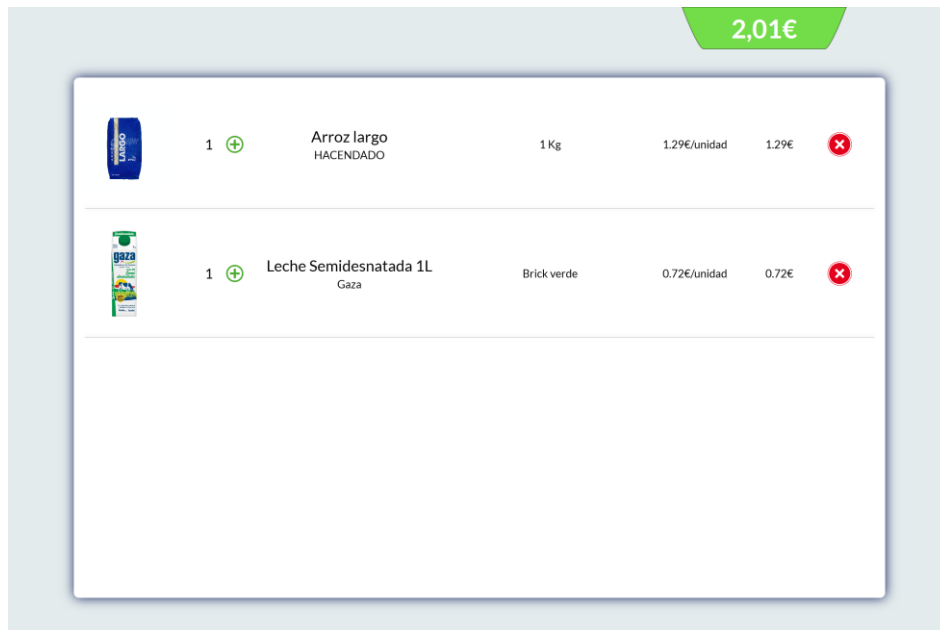


Figura 45. Pantalla de compra con varios productos añadidos

Si durante el proceso de compra se detecta algún cambio en el peso no esperado, se informa al usuario del problema y se muestra el carro en su estado erróneo.

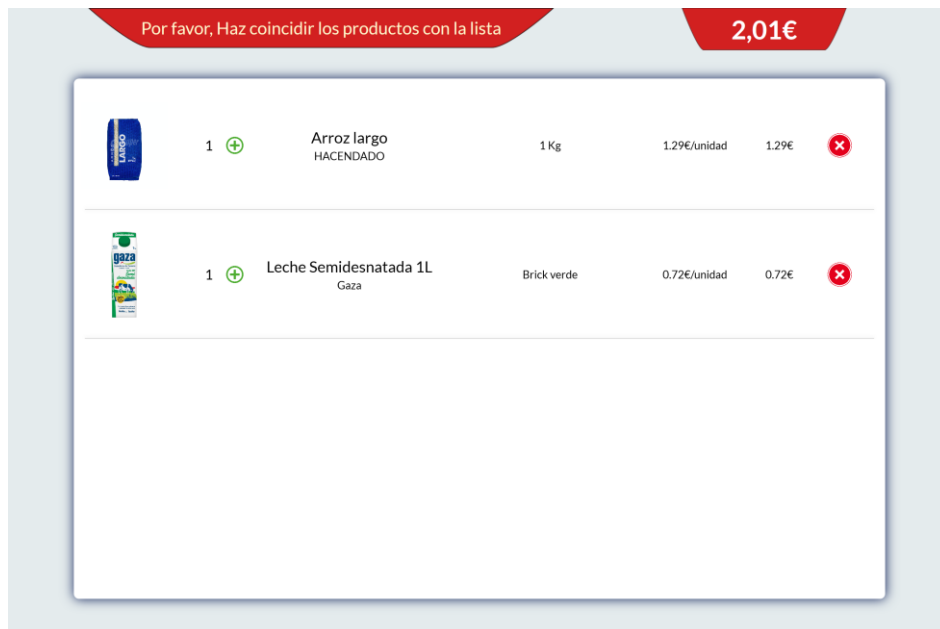


Figura 46. Pantalla de compra tras un cambio de peso no esperado

Para abandonar esta pantalla, un usuario debe indicar que quiere comenzar el proceso de pago acercando la caja del lector RFID (caja amarilla) a los puestos indicados en la zona de salida del establecimiento.

- Pantalla de pago

La pantalla de pago es mostrada cuando se desea comenzar el proceso de pago.

En el momento en el que el usuario acerca la caja del lector RFID (caja amarilla) a los puestos indicados en la zona de salida del establecimiento, se comprueba de forma inmediata la posibilidad de pago.

- **Posibilidad de pago**

En ella se informa al usuario de que el pago está siendo procesado a la vez que los LEDS lucen de color morado:

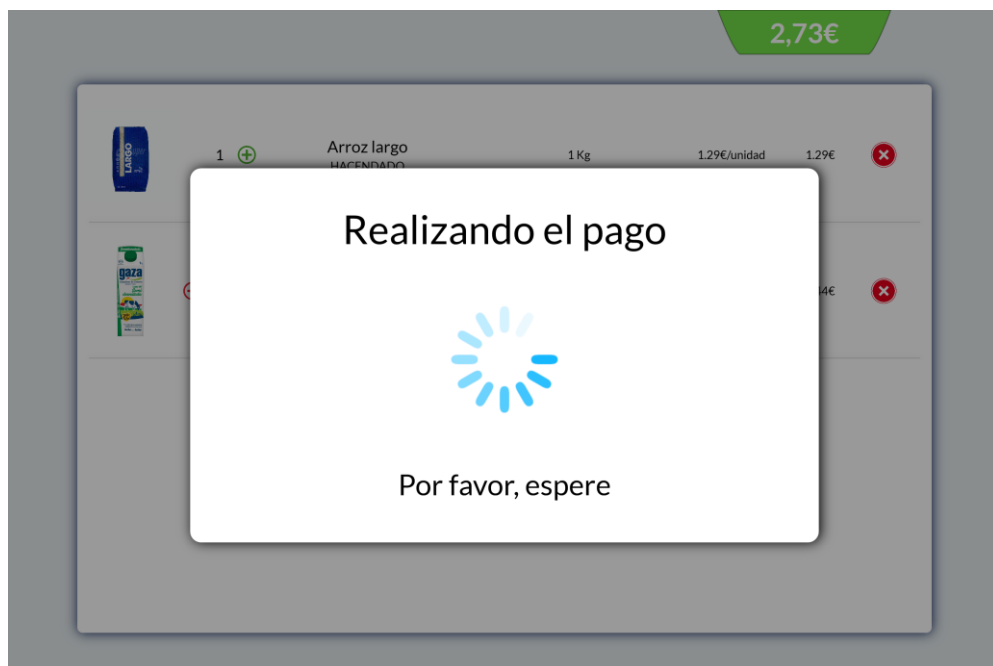


Figura 47. Pantalla de pago mientras se este se realiza

Tras la realización del pago se informa al usuario de que la operación ha sido satisfactoria y se procede al reinicio del dispositivo hasta la pantalla de standby.

- **Imposibilidad de pago**

Si el dispositivo detecta que el estado no está en estado correcto no permita la emisión de pago y muestra la pantalla correspondiente. En ella se informa al usuario de que no es posible realizar el pago a la vez que los LEDS lucen de color rojo:

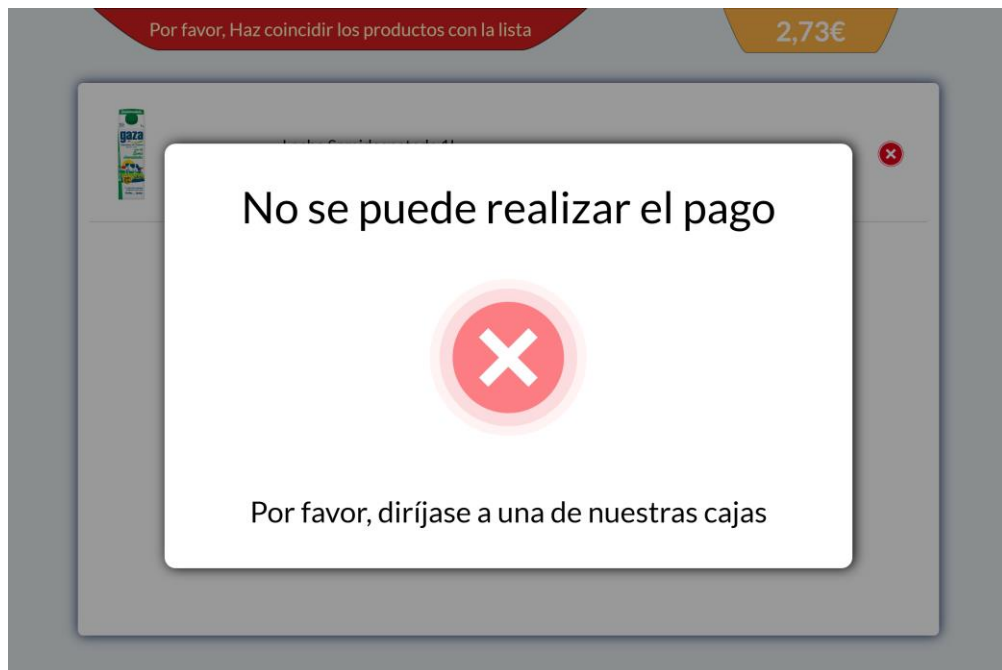


Figura 48. Pantalla de imposibilidad de pago

Mientras esta pantalla es mostrada el dispositivo es bloqueado, por lo que no admite ningún tipo de modificación.

Tras realizar el pago en una de las cajas de cobro cotidianas, el empleado reiniciará el dispositivo hasta la pantalla de standby mediante el uso de una tarjeta RFID.

- Pantalla de estadísticas

La pantalla de estadísticas es mostrada tras la correcta identificación de un administrador.

Esta pantalla se compone de tres subpantallas sobre las que se permite completa libertad de navegación. Para abandonar cualquiera de estas pantallas se utiliza el botón de cierre situado en la esquina superior derecha.

- **Estadísticas generales**

En ella se muestran las principales estadísticas del sistema, como las ventas diarias, el porcentaje de ocupación de carros, la evolución de las ventas en la última semana y la distribución de productos vendidos en la última semana por categorías.



Figura 49. Pantalla de estadísticas generales

- **Estadísticas de ventas**

En ella se muestran las principales estadísticas sobre las ventas, como las ventas diarias, semanales y mensuales, a la vez que la evolución de las ventas en el último mes.



Figura 50. Pantalla de estadísticas de ventas

- **Estadísticas de ventas de productos**

En ella se muestran los tres productos más vendidos en la última semana de cada categoría agrupados por la misma y su cantidad.

Si no se han vendido tres productos en una categoría se respetan los espacios vacíos que corresponderían a los mismos.



Figura 51. Pantalla de estadísticas de venta de productos

Para una mayor profundización en la implementación y puesta en marcha del sistema consúltese el documento “Anexo VI: Manual de usuario”.

6.5.2) Construcción del dispositivo

Este apartado se centra en el proceso de diseño, implementación y construcción del dispositivo desarrollado.

La estructura sobre la que se ha incorporado el dispositivo es el carro para la realización de compras Marsanz 100 ATH^[11], pero todos los componentes han sido diseñados para poder adaptarse a cualquier otro tipo de carro utilizado por el sector retail.



Figura 52. Carro de compra Marsanz 100 ATH

El principal objetivo durante el diseño de este dispositivo ha sido incorporar de la manera menos invasiva posible la pantalla de interacción, el lector de códigos de barras, el sensor de peso, el lector RFID, el Arduino, la tira LED y su correspondientes cableados a la estructura del carro.

Para cumplir dicho objetivo se desarrollaron tres elementos: un soporte para el lector de códigos de barras y la pantalla, una base con el sensor de peso incorporado y una caja para proteger el lector RFID.

6.5.2.1) Soporte principal

El soporte principal se ha diseñado para que pueda ser incorporado a cualquier carro dentro del mango de agarre. El objetivo de esta pieza es la incorporación del lector de códigos de barras, la pantalla y el Arduino en un único elemento.

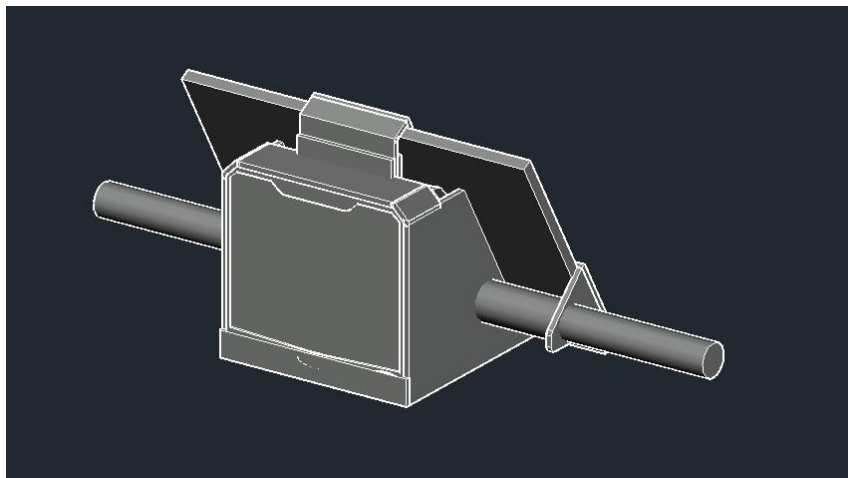


Figura 53. Modelo 3D del soporte principal

Como se puede observar la pieza ha sido diseñada para que soporte la pantalla con una inclinación que permita al usuario interactuar con ella de una forma cómoda visualmente.

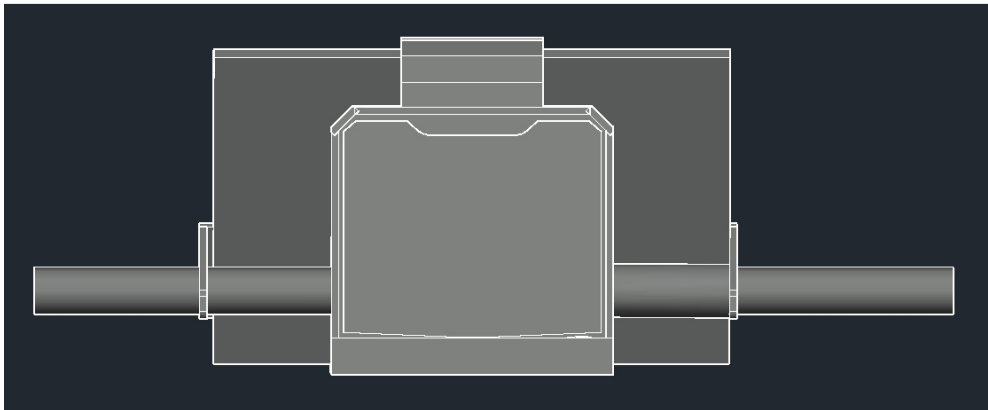


Figura 54. Alzado del modelo 3D del soporte principal



Figura 55. Soporte principal incorporado al carro

6.5.2.2) Sensor de peso

El sensor de peso se ha realizado utilizando cuatro células de carga de punto de apoyo múltiple de 50Kg conforme a la arquitectura de cuádruple punto de apoyo, la cual reduce los errores introducidos al colocar elementos en cualquier sector de la superficie de contacto. Estas células de carga han sido acopladas a un tablero aglomerado de 10 mm de ancho cortado con la forma de la base del carro.

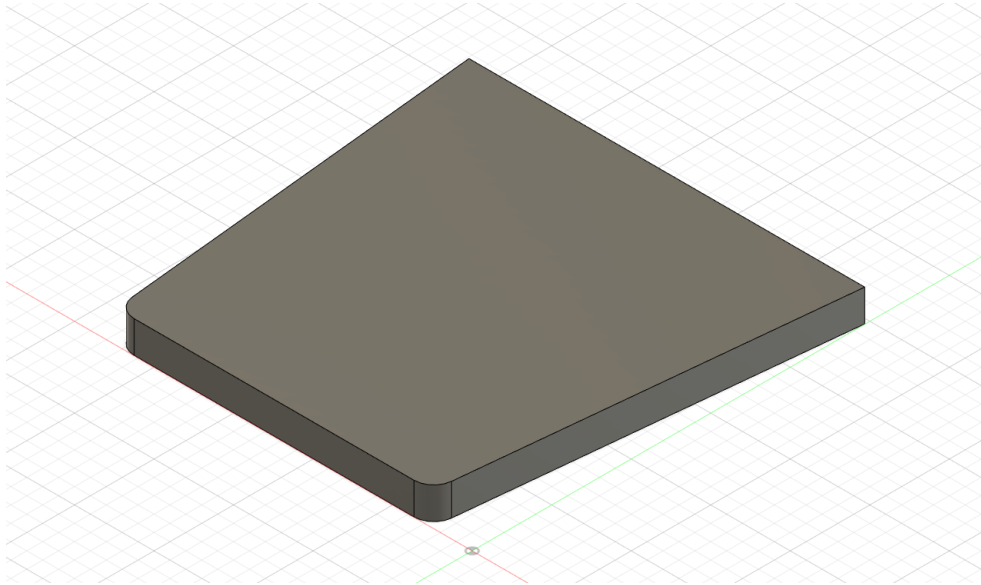


Figura 56. Modelo 3D de la base del sensor de peso.

Para el correcto funcionamiento de las células de carga, estas han sido introducidas en unas estructuras^[12] que permiten el correcto acoplamiento a la base y posibilitan las deformaciones físicas necesarias.

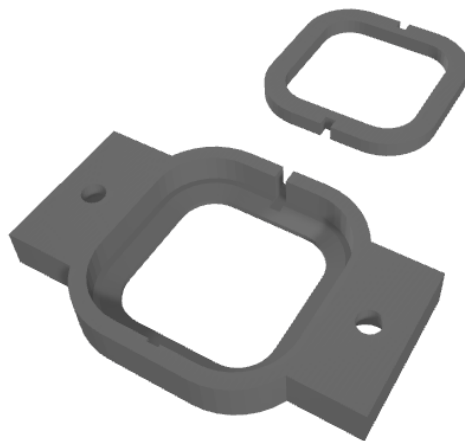


Figura 57. Modelo 3D de los soportes de las células de carga.

6.5.2.3) Lector RFID

Para proteger el lector RFID se ha impreso una de caja^[13] que ha sido acoplada en el lado izquierdo del carro. Esta caja ha sido impresa con PLA amarillo fluorescente para que sea fácil de identificar por los usuarios.

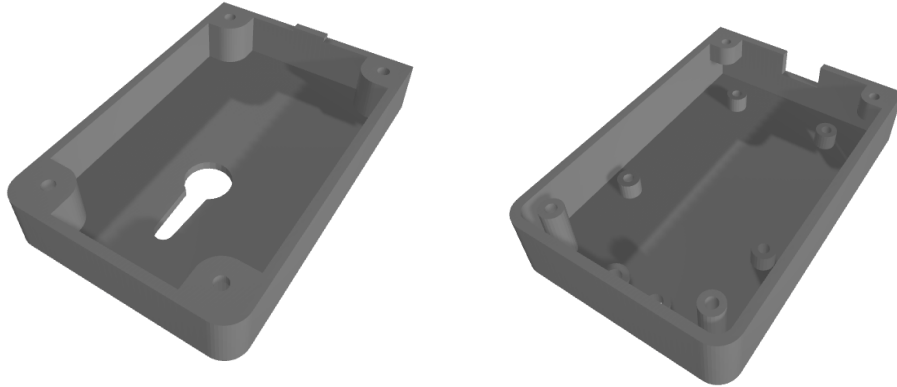


Figura 58. Modelo 3D de la caja contenedora del lector RFID

Para realizar el pago el usuario tan solo debe acercar dicha caja a la zona de pago especificada en la salida del establecimiento.



Figura 59. Lector RFID incorporado al carro

6.5.2.3) Resultado final

Tras la integración de todos los dispositivos el carro tiene el siguiente aspecto:

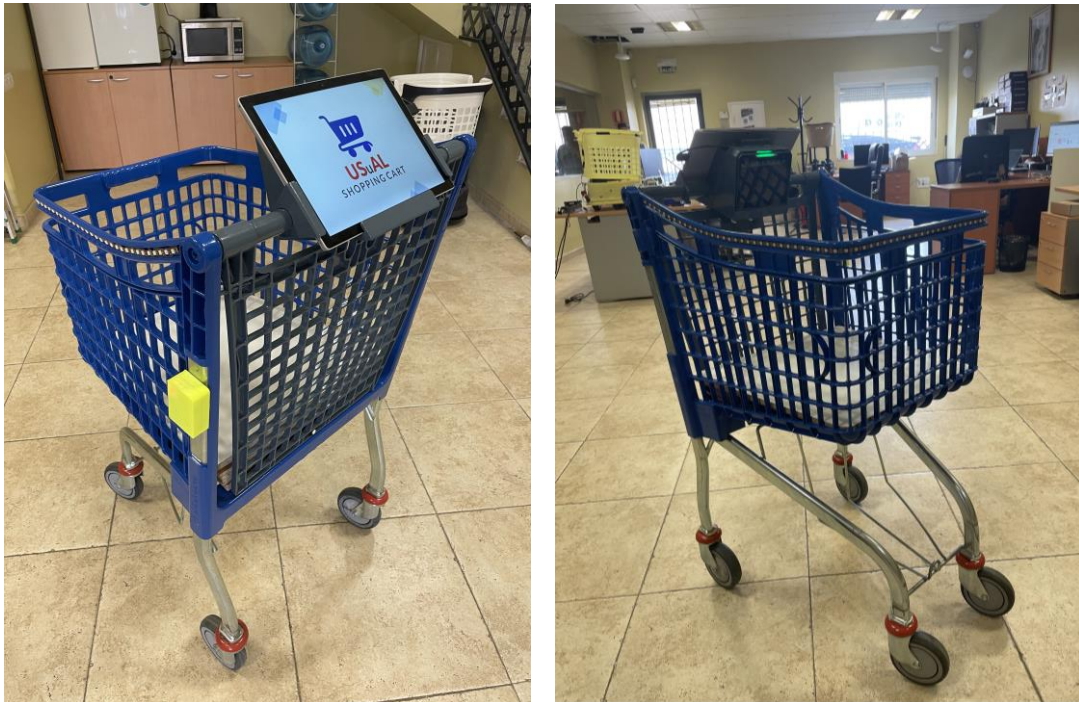


Figura 60. Resultado final del dispositivo

A continuación, se muestra uno de los distintos estados que pueden ser alcanzados por el dispositivo:

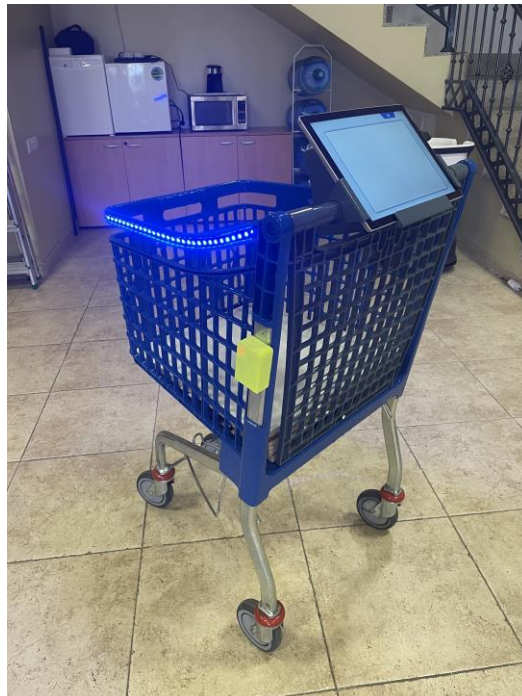


Figura 61. Ejemplo de estado del dispositivo: vacío

Para una mayor profundización en el diseño y creación del dispositivo o en el resultado final del mismo consúltese el documento "Anexo VII: Diseño y construcción del dispositivo".

7) Limitaciones y/o problemas identificados

En este apartado se exponen las limitaciones del sistema y del prototipo construido y los problemas identificados durante el desarrollo del proyecto.

7.1) Realización del pago

No se ha podido implementar un sistema de pago real.

Los requisitos impuestos por la Unión Europea (UE) en el ámbito de las transacciones financieras hace que se necesite obtener licencias específicas y cumplir con regulaciones rigurosas establecidas por la Autoridad Bancaria Europea (EBA) y la Directiva de Servicios de Pago (PSD2) de la UE^[14].

Para obtener una licencia como proveedor de servicios de pago, es necesario cumplir con requisitos como la protección de datos personales, la autenticación fuerte del cliente y la prevención del fraude. Su proceso de obtención es complejo y costoso, ya que implica la presentación de documentación detallada, demostración de capacidad financiera y técnica, y una evaluación exhaustiva por parte de las autoridades competentes. A demás, el incumplimiento de estas regulaciones puede acarrear sanciones legales y financieras significativas.

7.2) Detección de productos de poco peso

Los productos de un peso menor a 20 gramos no pueden ser detectados con fiabilidad.

El sensor de peso construido para la incorporación en el dispositivo tiene un margen de precisión de ± 5 gramos, lo cual implica que solo puede detectar cambios de peso mayores a ese rango. A partir del doble de este margen (± 10 gramos) se pueden empezar a considerar las mediciones como fiables.

Para lograr una mayor precisión en la detección de productos ligeros, se requerirían células de carga mucho más precisas y, por lo tanto, mucho más costosas.

7.3) Diseño e implementación del dispositivo

El diseño del dispositivo ha supuesto un esfuerzo mucho mayor al esperado, ya que no solo he necesitado aprender a manejar programas y herramientas utilizadas por ingenieros industriales, sino que ha sido necesario investigar sobre los materiales que debían ser utilizados, las fuerzas que estos debían resistir o los ángulos correctos que debían tener las distintas piezas diseñadas para permitir una cómoda visualización de la pantalla entre otras cosas.

La implementación del dispositivo también fue muy compleja, ya que debía adaptarse un carro cotidiano a las necesidades estructurales y de cableado impuestas por las distintas conexiones de los dispositivos.

8) Conclusiones y líneas de trabajo futuras

En este último apartado se exponen las conclusiones obtenidas tras el desarrollo del sistema software y el dispositivo y se indican algunas de las posibles líneas de trabajo futuras para la evolución del proyecto.

8.1) Conclusiones

Tras la finalización del proyecto, puedo afirmar que todos los objetivos del sistema han sido cumplidos:

- **Identificación de usuarios:** el sistema es capaz de identificar correctamente tanto a usuarios como a administradores.
- **Seguimiento de productos:** el sistema es capaz de identificar y comprobar los productos escaneados por el usuario. El sistema es capaz de detectar e informar de las interacciones realizadas de forma errónea.
- **Monitorización e información del estado del dispositivo:** el sistema es capaz de comprobar y comunicar su estado a los usuarios de forma simple mediante la actualización de la interfaz gráfica.
- **Recopilación de estadísticas:** el sistema es capaz de generar y mostrar estadísticas sobre las ventas realizadas mediante el uso del dispositivo.
- **Simplificación del pago:** el sistema es capaz de aligerar la confluencia en las cajas de pago mediante la realización de pagos automáticos.

Se puede decir que el desarrollo del sistema ha sido un éxito

También se puede afirmar que mis objetivos personales también se han cumplido, ya que siento que he evolucionado como desarrollador al ser capaz de implantar un sistema de alta complejidad.

Entre los logros conseguidos cabe destacar la construcción de un prototipo funcional del dispositivo, los conocimientos adquiridos como desarrollador software y también los adquiridos como diseñador 3D.

8.2) Líneas de trabajo futuras

Tras el desarrollo de una primera versión del dispositivo quedan abiertas varias líneas de desarrollo prometedoras:

- **Detección de productos a granel:** reconocer productos a granel, como frutas o verduras con el uso del propio dispositivo, sin necesidad de ningún sensor de peso externo.
- **Incorporación de Machine Learning:** sustituir el sistema de identificación mediante códigos por uno basado en reconocimiento de imágenes en tiempo real.

- **Integración con aplicaciones móviles:** sustituir la pantalla por el propio teléfono móvil del usuario.
- **Análisis y recomendaciones personalizadas:** utilizar algoritmos de aprendizaje automático y análisis de datos para recopilar información sobre las preferencias de compra de los clientes y ofrecer recomendaciones personalizadas de productos o promociones.

Seguir cualquiera de estas líneas de trabajo futuras aportaría funcionalidad al dispositivo, pero habría que estudiar cómo y cuando implantar las nuevas funcionalidades, estudiando anteriormente la forma de interacción realizada por los usuarios con la primera versión del dispositivo.

9) Bibliografía

[1] *Long lines cost US retailers \$37.7 billion.* (2018, 1 mayo). Adyen. Recuperado 12 de junio de 2023, de <https://www.adyen.com/blog/new-study-long-lines-cost-us-retailers-377-billion>

[2] National Retail Federation. (s. f.). *State of Retail.* NRF. Recuperado 12 de junio de 2023, de <https://nrf.com/research-insights/state-retail>

[3] (2000). *Proyecto De Final De Carrera En La Ingeniería Técnica En Informática: Guía De Realización Y Documentación: Vol.1.52.* <https://diaweb.usal.es/diaweb/comun/descargar.jsp?f=10011523pfc.pdf>

[4] Colaboradores de Wikipedia. (2023, enero 7). *Amazon Go.* Wikipedia. Recuperado 20 de junio de 2023, de https://es.wikipedia.org/wiki/Amazon_Go

[5] *Caper Cart.* (s.f.). Caper IA. Recuperado 20 de junio de 2023, de <https://www.caper.ai/cart>

[6] Veeve. (2023, 9 junio). Recuperado 20 de junio de 2023, de <https://veeve.io/>

[7] Martínez, A., & Martínez, R. (s.f.). *Guía a Rational Unified Process.* Escuela Politécnica Superior de Albacete - Universidad de Castilla la Mancha. <https://anaylenlopez.files.wordpress.com/2011/03/trabajo-guia20rup.pdf>

[8] Durán Toro, A., & Bernárdez Jiménez, B. (2000). *Metodología para la Elicitación de Requisitos de Sistemas Software.* Universidad de Sevilla, Informe Técnico LSI-2000-10. Recuperado 21 de junio de 2023, de <http://www.lsi.us.es/docs/informes/lsi-2000-10.pdf>

[9] Michaelstonis, Erjain, IEvangelist. (2023, 8 junio). *Modelo-Vista-Modelo de vista.* Microsoft Learn. Recuperado 24 de junio de 2023, de <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>

[10] Universidad de Salamanca. *Manual Institucional de Identidad Corporativa.* Recuperado 5 de abril de 2023, de <https://identidadcorporativa.usal.es/>

[11] *100 ATH | Marsanz.* (2022, 25 marzo). Marsanz. Recuperado 29 de junio de 2023, de <https://marsanz.es/products/100-ath-multigrip/>

[12] Thingiverse.com. (s. f.). *50kg Loadcell Bracket versionF by patrick3345.* Thingiverse. Recuperado 8 de febrero de 2023, de <https://www.thingiverse.com/thing:2624188>

[13] Thingiverse.com. (s. f.-b). *Desktop RFID reader 13,56MHz + Arduino Nano by dgoma.* Thingiverse. Recuperado 22 de febrero de 2023, de <https://www.thingiverse.com/thing:1413827>

[14] *Normas revisadas sobre servicios de pago en la Unión Europea.* (s. f.). EUR-Lex. Recuperado 22 de junio de 2023, de <https://eur-lex.europa.eu/ES/legal-content/summary/revised-rules-for-payment-services-in-the-eu.html>