

Trabajo Fin de Grado

Título del trabajo:

Desarrollo de un sistema inteligente de gestión de
alimentos basado en RFID

English title:

Development of an intelligent food management
system based on RFID

Autora:

Berta Olano Guillén

Directores:

Diego Gutiérrez Pérez
Ana Serrano Pacheu

Titulación de la autora:

Ingeniería Informática

ESCUELA DE INGENIERÍA Y ARQUITECTURA

Junio de 2025

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que han formado parte de esta etapa de mi vida y que, de una u otra forma, han contribuido a que este trabajo haya sido posible.

Estos cinco años de carrera han sido una experiencia intensa, llena de aprendizajes, desafíos y transformaciones. No ha sido un camino fácil. Ha habido momentos de duda, de cansancio, de querer rendirme. Pero también ha habido descubrimientos, logros y muchas razones para seguir adelante. Este trabajo no es solo el final de una etapa académica: es el reflejo de todo ese recorrido, del esfuerzo acumulado, y de todas las personas que me han acompañado.

En primer lugar, a mi familia, por estar presente en cada paso del camino, su ánimo y su fe en mí. Gracias por recordarme quién soy, incluso en los días en los que me costaba verlo. En especial, gracias a mi madre, por su amor constante, por su ejemplo de entrega y por recordarme cada día el valor de la dedicación y la perseverancia.

A mis amigos, por compartir risas, desahogos y silencios, por su compañía, su alegría y por ofrecerme siempre un refugio en los días difíciles.

Y, por último, a mí misma, por resistir, por crecer, y por no rendirme cuando más difícil parecía.

Este trabajo ha sido parcialmente financiado por el Programa de Becas y Ayudas del Instituto de Investigación en Ingeniería de Aragón (I3A) y por la Cátedra URBASER e INETUM.

Resumen

Este Trabajo de Fin de Grado explora el uso de tecnología RFID para optimizar la gestión de alimentos en el hogar, con el objetivo de reducir el desperdicio y mejorar el almacenamiento. A través del desarrollo de un prototipo, se estudiará cómo las etiquetas RFID pueden facilitar el registro automático de productos y la visualización del inventario en tiempo real.

El sistema propuesto incluirá funcionalidades como el aviso de fechas de caducidad y visualización de recetas, comprobando los ingredientes disponibles, contribuyendo así a una gestión más eficiente y sostenible de los alimentos.

Para el desarrollo, se integrará un lector RFID Proxmark3 con una aplicación móvil en Java, permitiendo el uso de los datos contenidos en las etiquetas RFID. Se adoptará una metodología iterativa, explorando el uso de bases de datos para almacenar y sincronizar la información.

El proyecto se estructura en varias fases: primero, la familiarización con la tecnología RFID y la configuración del entorno de trabajo; posteriormente, el desarrollo de una arquitectura de conexión entre los distintos componentes del sistema; y finalmente, la implementación de una interfaz en Java que permita una experiencia de usuario intuitiva para la gestión del inventario y la trazabilidad de los productos. Así mismo, se comprobó el correcto funcionamiento del sistema mediante un conjunto de pruebas, tanto con usuarios reales como mediante pruebas automáticas.

Este trabajo se enmarca dentro de los Objetivos de Desarrollo Sostenible (ODS) de la ONU, contribuyendo a los objetivos 12.3 (reducción del desperdicio de alimentos), 9.4 (tecnología e innovación para la sostenibilidad) y 8.4 (eficiencia en el uso de recursos).

Summary

This Bachelor's Thesis explores the use of RFID technology to optimize food management at home, with the goal of reducing waste and improving storage. Through the development of a prototype, the study will examine how RFID tags can facilitate the automatic registration of products and the real-time visualization of inventory.

The proposed system will include features such as expiration date alerts and recipes with available ingredients, thus contributing to more efficient and sustainable food management.

For the development, a Proxmark3 RFID reader will be integrated with a mobile application developed in Java, allowing the use of data stored in RFID tags. An iterative methodology will be adopted, exploring the use of databases to store and synchronize information.

The project is structured in several phases: first, familiarization with RFID technology and the configuration of the working environment; next, the development of a connection architecture between the different components of the system; and finally, the implementation of a Java interface that offers an intuitive user experience for inventory management and product traceability. The correct functioning of the system was verified through a set of tests, both with real users and through automated testing.

This work aligns with the United Nations Sustainable Development Goals (SDGs), contributing to goals 12.3 (food waste reduction), 9.4 (technology and innovation for sustainability), and 8.4 (efficient use of resources).

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Metodología de desarrollo	2
2. Estado del arte	3
2.1. Fundamentos y aplicaciones de la tecnología RFID	3
2.2. Lector Proxmark3: características y funcionamiento	4
2.2.1. Software Proxspace	4
2.3. Soluciones existentes para la gestión de inventarios y alimentos	5
3. Diseño y planificación del sistema	7
3.1. Arquitectura general del sistema	7
3.2. Selección de tecnologías y herramientas	8
3.2.1. Hardware	8
3.2.2. Software	8
3.3. Requisitos del sistema	9
4. Implementación del sistema	11
4.1. Configuración y pruebas del lector RFID	11
4.2. Automatización del registro de productos mediante etiquetas RFID	12
4.2.1. Codificación de las etiquetas	12
4.2.2. Gestión de la base de datos	13
4.2.3. Lectura e interpretación de datos	13
4.3. Desarrollo de la aplicación móvil	14
4.3.1. Esquema general	14
4.3.2. Diseño de la interfaz de usuario y estructura de la aplicación	15
4.3.3. Comunicación con el lector e integración con bases de datos	16
4.3.4. Persistencia y datos	16
4.4. Consideraciones sobre seguridad	17
5. Evaluación y Pruebas	19
5.1. Fase 1: Pruebas iniciales de desarrollo y pruebas informales con usuarios	19
5.2. Fase 2: Pruebas formales de usuario	20
5.3. Pruebas automáticas	22
5.3.1. Pruebas básicas de inserción, recuperación y actualización	23
5.3.2. Pruebas de rendimiento	23
5.3.3. Pruebas de acceso concurrente	23
5.4. Conclusiones	23

6. Conclusiones	25
6.1. Limitaciones	25
6.2. Líneas futuras	26
6.3. Conclusión final	26
Bibliografía	27

Capítulo 1

Introducción

1.1. Motivación

La idea de una cocina del futuro plantea un entorno donde la tecnología no solo simplifica tareas, sino que mejora la forma en la que gestionamos los alimentos. Partiendo de esta visión, este trabajo explora cómo integrar la innovación en el ámbito doméstico para hacer más cómoda y eficiente la experiencia en la cocina.

Uno de los principales problemas detectados en los hogares es el desperdicio de alimentos. Según las Naciones Unidas [8]: “Cada día, los hogares desperdician más de 1000 millones de comidas, el equivalente a 1,3 comidas diarias para cada persona hambrienta en el mundo”. Este dato evidencia no solo un problema logístico sino también ético y ambiental, y expone la necesidad de desarrollar soluciones tecnológicas que ayuden a las personas a gestionar mejor lo que compran, almacenan y consumen.

En los últimos años han surgido diferentes propuestas para resolver este problema, desde sistemas que usan visión artificial mediante cámaras en neveras hasta aplicaciones móviles que requieren registro manual del inventario. Sin embargo, estas soluciones presentan limitaciones: las cámaras pueden tener errores de detección o problemas de privacidad, y el registro manual es poco práctico, propenso al olvido y sensible a los errores humanos.

Frente al panorama actual, surge la oportunidad de explorar el uso de tecnologías como la RFID, ya ampliamente utilizadas en sectores como la logística y el retail, pero aún poco implantadas en el entorno doméstico. Este trabajo busca cubrir ese hueco: diseñar un sistema que automatice la gestión del inventario alimentario en casa, aprovechando las ventajas que nos ofrece el uso de RFID como son la precisión, automatización y el bajo coste a pequeña escala, para mejorar la planificación del consumo y reducir el desperdicio.

1.2. Objetivos

Aunque existen soluciones similares, este trabajo pretende aportar una propuesta funcional basada en RFID adaptada específicamente al entorno doméstico. No se trata solo de aplicar una tecnología existente, sino de demostrar su viabilidad, evaluar su usabilidad y proponer una arquitectura que podría integrarse en futuras cocinas inteligentes.

En este contexto, este trabajo tiene como objetivo principal el desarrollo de un prototipo de un sistema funcional de gestión de inventario de alimentos basado en tecnología RFID, demostrando cómo puede adaptarse a entornos cotidianos, mejorando la eficiencia y sostenibilidad en el manejo de productos alimenticios en el hogar.

De forma más concreta, los objetivos son:

- Automatizar el seguimiento de los productos alimenticios en los hogares, registrando su entrada, salida y caducidad sin intervención manual constante mediante el uso de etiquetas y un lector RFID.

- Diseñar una aplicación móvil que integre el sistema con una interfaz que le permita conocer qué productos tiene, en qué cantidad y cuándo caducan.
- Utilizar esta información para otras funcionalidades como una lista de la compra más eficaz o búsqueda de recetas con los alimentos registrados.
- Reducir el desperdicio de alimentos fomentando un consumo más planificado y, por tanto, más consciente.
- Explorar el potencial de integración futura con entornos de cocina inteligente.

Este sistema está concebido como un primer paso a una cocina completamente automatizada, donde la tecnología asista al usuario a tomar sus decisiones diarias de una forma no solo más cómoda sino más comprometida.

1.3. Metodología de desarrollo

En este trabajo se ha seguido una metodología iterativa, estructurada en fases que permitieron avanzar progresivamente y realizar mejoras constantes sobre el prototipo. Este proceso se ha dividido en distintas fases:

1. Investigación inicial: estudio del estado del arte de la tecnología RFID: cómo funciona, qué tipos hay y cuál se adecúa más a este propósito y casos de uso similares en el ámbito doméstico e industrial.
2. Diseño del sistema: definición de la arquitectura, diseño del flujo de datos y selección de herramientas y tecnologías según la compatibilidad y facilidad de integración.
3. Automatización de lectura: desarrollo de un script en Python que ejecuta periódicamente la lectura de etiquetas actualizando la base de datos.
4. Desarrollo de la aplicación móvil: implementación de una interfaz funcional que permite consultar y gestionar el inventario a partir de los datos obtenidos por el lector.
5. Pruebas y validación: evaluación con usuarios reales del prototipo, identificando problemas y soluciones en la usabilidad.

Esta metodología permitió corregir errores, ajustar decisiones técnicas y adaptar el sistema de la mejor manera a las necesidades detectadas durante el desarrollo.

Capítulo 2

Estado del arte

2.1. Fundamentos y aplicaciones de la tecnología RFID

La tecnología RFID (Radiofrequency Identification o Identificación por Radio Frecuencia) [1] permite la identificación y seguimiento de objetos mediante el uso de ondas de radio. Un sistema RFID normalmente está compuesto por tres elementos principales: una etiqueta (tag), un lector (reader) y un sistema de gestión y procesamiento de datos (middleware). La etiqueta es un dispositivo que puede cobrar distintas formas: pegatinas, tarjetas, etiquetas... Esta contiene un chip con información que puede ser escrita y leída por el lector que captura la señal cuando está dentro del rango.

Una etiqueta está formada por una antena, un transductor radio y el microchip. El funcionamiento es: la antena permite al microchip transmitir la información contenida en la etiqueta. El chip contiene una memoria con una capacidad que depende del modelo. Además, existe un material (sustrato) que mantiene juntos todos los componentes y que depende de la forma que tenga la etiqueta. Por ejemplo, en el caso de las etiquetas adherentes se trata de un film de plástico.

Existen tres tipos principales de RFID según la banda de frecuencia utilizada [3]:

- Baja Frecuencia (LF): tiene un rango de lectura corto (10 cm) y la velocidad de lectura es más lenta. Se suele utilizar en control de accesos y animales.
- Alta Frecuencia (HF): tiene un rango de lectura desde 10 cm hasta 1 m y la velocidad es rápida. Se suele utilizar para ticketing, tarjetas y sistemas de pago, entre otros.
- Ultra Alta Frecuencia (UHF): tiene el rango de lectura más grande (hasta 12 m) y es el más rápido, sin embargo, es más sensible a interferencias. Se usa, sobre todo, en inventarios en tiendas y almacenes y otros procesos logísticos.

Aunque también existen otros como la frecuencia media (MF) y súper alta (SHF), algo menos utilizadas.

Este tipo de tecnología, debido a su bajo coste y reducido tamaño, ha hecho que pueda ser utilizada en numerosos productos. Su precedente más importante es el código de barras, tecnología que, actualmente, se usa para el registro de alimentos y está presente en todos los productos comercializados. Sin embargo, la tecnología RFID ofrece como ventaja que no necesita visibilidad directa para funcionar y que identifica de forma única cada unidad del producto (no del tipo de producto). Además, este tipo de etiquetado permite un seguimiento del movimiento de los productos, localizando inequívocamente cada unidad dentro del rango de lectura y se pueden localizar varias etiquetas simultáneamente, permitiendo gestionar una elevada cantidad de información en tiempo real.

En el contexto doméstico, la aplicación de esta tecnología es todavía limitada, pero tiene un alto potencial. Automatizar el control de alimentos usando esta tecnología puede contribuir a una mejora de la planificación en el consumo de alimentos.

2.2. Lector Proxmark3: características y funcionamiento

Para este trabajo se ha utilizado un lector Proxmark3 Easy, un dispositivo versátil capaz de trabajar tanto en baja como en alta frecuencia, lo que lo hace compatible con una amplia gama de etiquetas RFID. Este lector es conocido por su facilidad de uso, bajo coste y por contar con herramientas de software libre que facilitan el desarrollo de prototipos y pruebas. La versión Proxmark3 Easy es una versión simplificada del lector Proxmark3 original, especialmente diseñado para aquellos que no tienen experiencia previa con este tipo de tecnología.

Este lector se conecta al ordenador mediante USB y se controla a través de comandos en la terminal. Para este proyecto se ha utilizado el entorno de software libre Proxspace [10] que proporciona una interfaz cómoda para interactuar con el lector, leer, escribir, analizar y clonar etiquetas.

Aunque existen otros lectores RFID en el mercado (como lectores fijos o industriales, lectores integrados en móviles...), esta es una opción adecuada para el desarrollo de un prototipo en el contexto de un Trabajo de Fin de Grado por su flexibilidad y soporte técnico. No obstante, para una posible implementación futura orientada al usuario doméstico, sería necesario evaluar otro tipo de lectores para la integración en el entorno de una cocina. En concreto, se debería encontrar un lector capaz de capturar simultáneamente numerosas etiquetas y en un rango mayor. Además, para partes de la cocina como la nevera se debería tener en cuenta la temperatura, ya que puede influir en el funcionamiento del lector.

Algunas alternativas comerciales viables para este tipo de entorno son los lectores fijos, como el *Impinj Speedway R420* [2], que permite la lectura masiva de etiquetas UHF en tiempo real, o el *ThingMagic M6e Nano* [6], un módulo compacto ideal para integrarse en electrodomésticos. También existen soluciones como los lectores *Feig ID ISC.MRU102* [4] que están optimizados para entornos industriales con condiciones adversas, y que podrían adaptarse a situaciones como la humedad o temperaturas más bajas en frigoríficos. La elección del lector dependerá de factores como el rango de lectura necesario, el entorno físico, el tipo de etiquetas utilizadas y el coste de implementación.

2.2.1. Software Proxspace

El entorno Proxspace es una herramienta de software libre disponible en GitHub [10] que permite compilar, flashear y ejecutar comandos del lector Proxmark3 de forma sencilla para los usuarios de Windows. En concreto, es una colección de herramientas que utiliza MSYS2, un distribuidor de software y plataforma de desarrollo para Windows.

Esta plataforma proporciona un cliente que incluye todas las dependencias necesarias (como gcc, arm, python, pkg-config, cmake, entre otras), reduciendo significativamente la complejidad del proceso de instalación y uso. Proporciona una terminal de bash donde se pueden utilizar comandos de lectura, escritura y configuración de etiquetas detalladas en el Anexo 1 (*Detalles técnicos del uso de Proxspace para Proxmark3*). Proporciona también herramientas automáticas, sistemas de control de versiones y similares para crear aplicaciones nativas de Windows, mediante cadenas de herramientas MinGW-w64. ProxSpace utiliza la cadena de herramientas integrada GNU Arm para compilar el firmware del Proxmark 3.

2.3. Soluciones existentes para la gestión de inventarios y alimentos

El uso de RFID para la gestión de inventarios está ampliamente extendido en entornos industriales y logísticos. Numerosas empresas utilizan esta tecnología para mejorar el control de stock, reducir errores de inventario y optimizar la trazabilidad de productos.

En el ámbito del consumidor, algunas cadenas de supermercados, como Amazon Fresh, han incorporado etiquetas RFID en sus productos para automatizar el proceso de compra: el cliente entra, recoge los productos y sale sin pasar por caja, ya que las etiquetas pueden identificar automáticamente los artículos adquiridos.

Este tipo de soluciones apunta a un futuro donde los alimentos podrían llegar a casa ya etiquetados. En el marco de ese escenario ideal, sistemas como el desarrollado en este trabajo serían mucho más viables y eficientes, ya que el usuario dispondría de un inventario completamente automatizado y siempre disponible en su teléfono móvil.

Actualmente, existen varias aplicaciones orientadas a permitir al consumidor llevar un control de los alimentos almacenados en la nevera o despensa. Aunque ninguna de ellas utiliza la tecnología RFID, ofrecen funcionalidades similares en cuanto a planificación y reducción del desperdicio. Algunas de ellas son:

- NoWaste [9]: que permite almacenar una lista manualmente de los alimentos que contiene tu nevera, qué alimentos debes usar primero, planear tus comidas, crear una lista de la compra, reducir el desperdicio de alimentos...

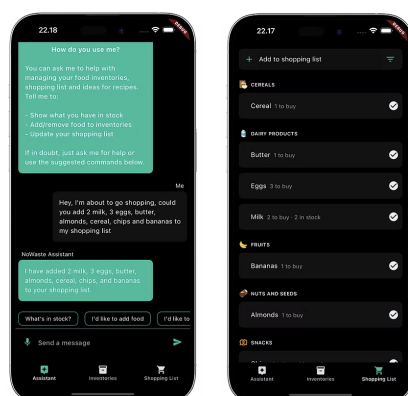


Figura 2.1: Interfaz proporcionada por la app NoWaste.

- SmartThings de Samsung [11]: permite tomar fotos de cada alimento registrando su fecha de caducidad y te permite fijar un recordatorio para cuando caduque.

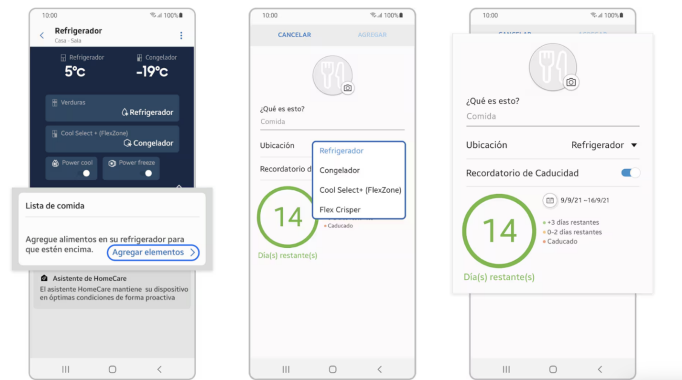


Figura 2.2: Interfaz proporcionada por la app SmartThings de Samsung.

- Smarter FridgeCam [12]: es una herramienta que añade artículos de tu inventario a la lista de la compra cuando se te acaban. Está integrado con AmazonFresh para hacer la compra solo con un clic. También permite rastrear las fechas de caducidad. Funciona con una cámara integrada en la nevera y un escáner para códigos de barras utilizando la cámara del móvil.

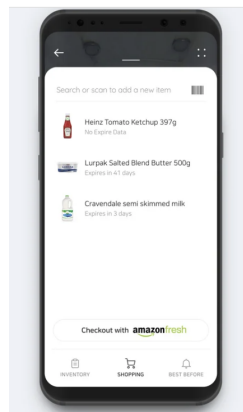


Figura 2.3: Interfaz proporcionada por la app Smarter.

Aunque estas aplicaciones tienen funcionalidades parecidas a las mostradas en este prototipo, todas requieren una entrada manual constante por parte del usuario. Esto puede suponer una limitación para su uso a largo plazo. Por tanto, con esta propuesta de automatización del registro se mejora significativamente la experiencia del usuario. Además, el sistema está diseñado de forma abierta y modular, con lo que el prototipo podría integrarse fácilmente en otros entornos (como una cocina inteligente) o combinarse con funcionalidades presentes en aplicaciones ya existentes, como la cámara o la integración con un supermercado online.

Capítulo 3

Diseño y planificación del sistema

Este capítulo describe la arquitectura del sistema desarrollado y las tecnologías empleadas para cada componente.

3.1. Arquitectura general del sistema

El sistema propuesto sigue una arquitectura modular y local, que permite una integración entre el hardware de lectura, la capa de automatización, la base de datos y la aplicación móvil. Todos los componentes funcionan de forma local y están conectados mediante un flujo de datos sencillo, como se muestra en la figura 3.1. Esta arquitectura fue pensada para ser funcional, reproducible en un entorno doméstico y adaptable a futuras expansiones.

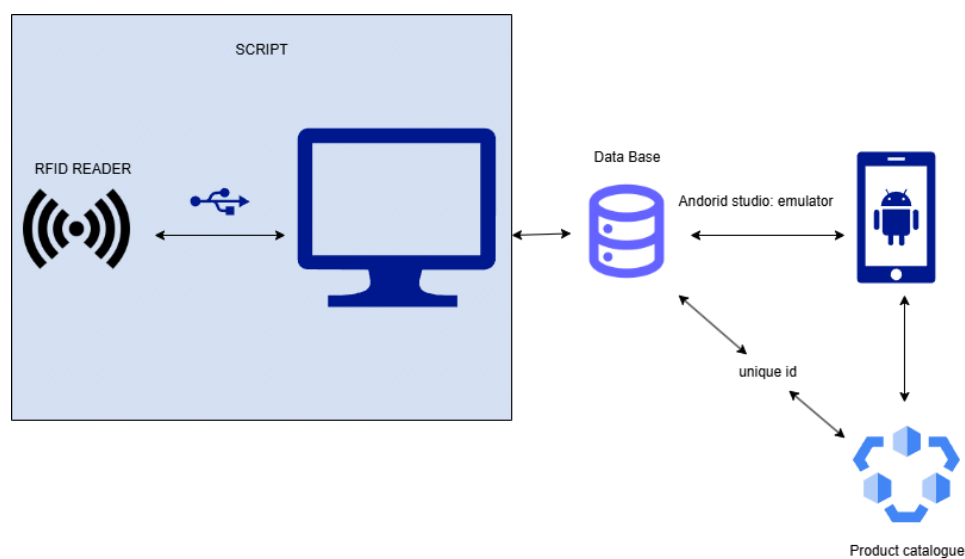


Figura 3.1: Arquitectura general del sistema.

En concreto, los principales componentes son:

- Lector RFID Proxmark3 Easy: conectado al ordenador a través de un puerto USB (COM3 o COM4), que permite la lectura de etiquetas RFID de alta y baja frecuencia. Actúa como punto de entrada para los datos.
- Script de automatización en Python: ejecuta el cliente de Proxmark usando el software de Proxspace y periódicamente un comando de lectura de etiquetas. Cada vez que detecta una lectura válida, guarda la información correspondiente (ID, tipo de producto, fecha de

caducidad y cantidad del producto) en una base de datos local. Es, por tanto, la parte encargada de procesar los datos.

- Base de datos SQLite: un archivo .db ubicado en la carpeta */assets* de la aplicación móvil. Esta base de datos contiene el inventario activo y la información asociada a los productos registrados en tiempo real. Es decir, almacena los datos.
- Aplicación móvil: desarrollada en Java, accede a la base de datos para mostrar al usuario una vista actualizada del inventario. Esta parte utiliza y presenta los datos al usuario. Además, esta app tiene acceso a dos catálogos diferentes (archivos .json en */assets*) que contienen una lista de los productos y recetas respectivamente. Los productos representan el catálogo de alimentos que puede comprar el usuario y su identificador corresponde con el contenido por la tarjetas, de forma que podemos identificar de forma única cada alimento.

Este diseño responde a un enfoque secuencial en local: el lector captura, el script procesa, la base de datos almacena y la app muestra la información.

3.2. Selección de tecnologías y herramientas

La elección de las herramientas se ha guiado por criterios de compatibilidad, sencillez de integración y conocimiento de las herramientas.

3.2.1. Hardware

Se eligió el lector Proxmark3 Easy tras un detallado estudio de los diferentes tipos de lectores y sus principales usos actuales en el mercado. Este lector se eligió por diversas razones:

- Su versatilidad, ya que puede ser usado tanto en alta como baja frecuencia y en diferentes tipos de etiquetas.
- Su amplia documentación por parte de usuarios.
- La existencia de un software libre para su uso, que es además intuitivo y permite todas las funcionalidades necesarias para el desarrollo de este sistema.
- Su bajo coste en comparación con otros lectores del mercado. Aunque algunos de estos otros lectores son más parecidos a la clase de lector que formaría parte del sistema final.

3.2.2. Software

En cuanto a tecnologías y herramientas de Software se eligió:

- Proxspace: entorno para Windows que permite compilar y ejecutarse el cliente de Proxmark3 con todos los binarios preinstalados.
- Visual Studio Code: utilizado para el desarrollo del Script en Python por su soporte y comodidad de uso.
- Python 3: elegido por su rapidez de desarrollo y especialmente por sus librerías nativas para manejo de procesos y subprocesos y SQLite, y facilidad de integración.
- Android Studio: entorno completo para el desarrollo de aplicaciones móviles en Android. Facilita el diseño visual de interfaces y emulación de dispositivos Android.
- Java: lenguaje principal para la app, elegido por su compatibilidad con Android y abundante documentación. Además, el conocimiento previo de este lenguaje facilitó su uso.

- SQLite: base de datos ligera, fácil de integrar tanto desde Python como desde la app Android.
- Git y GitHub: utilizados para el control de versiones, backup del código y organización del repositorio del proyecto.

3.3. Requisitos del sistema

Esta sección recoge las funcionalidades que se han definido para el sistema, desde el punto de vista del usuario final.

Las funcionalidades fueron seleccionadas en base a los objetivos planteados anteriormente, en particular la necesidad de automatizar la gestión de productos alimenticios, facilitar la planificación del consumo y reducir el desperdicio de alimentos. Se ha buscado cubrir todo el flujo de uso: desde la entrada del producto en el hogar, su registro automático, la visualización del inventario, hasta el uso activo de esa información para planificar compras y cocinar.

A continuación, se detallan los requisitos funcionales definidos:

Código	Descripción
RF1	Registrar automáticamente productos al detectar una etiqueta RFID.
RF2	Almacenar los datos del producto en una base de datos local.
RF3	Visualizar el inventario de productos en la aplicación móvil.
RF4	Mostrar detalles como nombre, fecha de caducidad y cantidad.
RF5	Permitir la compra de los productos contenidos en el catálogo.
RF6	Evitar duplicados cuando se detecta una etiqueta ya registrada.
RF7	Mostrar una lista de recetas.
RF8	Para cada receta mostrar si los ingredientes están disponibles o no en el inventario.

Cuadro 3.1: Requisitos funcionales del sistema.

Estos requisitos funcionales permiten automatizar tareas clave del día a día en la cocina. Por ejemplo, al registrar los productos automáticamente con RFID (RF1), se elimina la entrada manual. La base de datos local (RF2) permite independencia de la nube, alineada con el objetivo de un sistema autónomo. La visualización del inventario (RF3, RF4) y la integración con recetas (RF7, RF8) promueven un consumo más planificado, mientras que evitar duplicados (RF6) mejora la fiabilidad del sistema.

Por su parte, los requisitos no funcionales garantizan que el sistema sea viable en el entorno donde se va a probar:

Código	Descripción
RNF1	El sistema debe funcionar completamente en local.
RNF2	La app debe ejecutarse correctamente en emuladores Android.
RNF3	El script debe operar de forma autónoma sin intervención continua.
RNF4	El tiempo de procesamiento por producto debe ser menor a 5 segundos.

Cuadro 3.2: Requisitos no funcionales del sistema.

Estos requisitos aseguran que el prototipo pueda probarse sin necesidad de infraestructuras complejas, permitiendo centrarse en validar la experiencia de usuario y la funcionalidad básica. Además, marcan unos mínimos de rendimiento que buscan garantizar una experiencia fluida y sin esperas prolongadas.

En conjunto, estos requisitos cubren tanto la experiencia del usuario como las condiciones técnicas necesarias para validar el enfoque propuesto.

Capítulo 4

Implementación del sistema

En este capítulo se detallan los pasos seguidos para la implementación del sistema RFID, incluyendo tanto el desarrollo de los componentes como su integración en un prototipo funcional. El objetivo es mostrar con rigor técnico cómo se ha construido, cuáles han sido los desafíos encontrados y qué soluciones se han adoptado.

4.1. Configuración y pruebas del lector RFID

La primera fase del desarrollo consistió en la configuración del lector Proxmark3 Easy, la preparación del entorno de trabajo y la realización de pruebas con distintas etiquetas RFID. El objetivo era validar su funcionamiento, alcance y compatibilidad con el sistema propuesto.

Para poder utilizar el lector, fue necesario configurar el entorno *Proxspace*, que incluye el cliente Proxmark3 y las herramientas necesarias para compilar firmware y ejecutar comandos sobre el lector. El entorno se instaló en un sistema operativo Windows, y el lector se conectó correctamente a través del puerto USB, generalmente asignado como COM3 o COM4. Una vez verificada la conexión, se llevaron a cabo pruebas básicas de lectura con etiquetas RFID de distintas frecuencias:

- **Etiquetas LF (125 kHz):** Se utilizó el comando `lf search` para detectar la presencia de etiquetas de baja frecuencia. El lector identificó correctamente el tipo de protocolo utilizado y mostró el UID de la tarjeta.
- **Etiquetas HF (13.56 MHz):** Se usó el comando `hf search`, que permitió detectar tarjetas compatibles con esta frecuencia. De la misma manera, se mostraba la información sobre la lectura.

Se comprobó la estabilidad de la lectura, el tiempo de respuesta y el alcance efectivo. Este último resultó ser limitado, lo cual obliga a colocar la etiqueta prácticamente encima de la zona activa del lector correspondiente a la frecuencia usada.

Además, la principal limitación de este lector es que no permite la lectura simultánea de múltiples etiquetas. Al tratarse de un prototipo, se ha considerado que es suficiente para probar la usabilidad de este tipo de tecnologías en el marco de la gestión de inventario de alimentos. Sin embargo, en el caso de realizar un producto para el usuario final, el lector más adecuado sería un lector que permitiese detectar un número grande de etiquetas (para leer todos los alimentos al mismo tiempo) y que alcanzase a leer en el rango de la parte de la cocina donde se desee mantener el registro de alimentos (nevera, despensa, armario de cocina...).

4.2. Automatización del registro de productos mediante etiquetas RFID

Una vez validado el correcto funcionamiento del lector RFID, el siguiente paso fue automatizar el proceso de lectura y registro de productos. Para ello, se desarrolló un script en Python capaz de ejecutar comandos del entorno *proxspace*, interpretar los datos leídos y almacenarlos en una base de datos local.

Este flujo se puede entender como un proceso ETL (*Extract, Transform, Load*), muy común en el ámbito de la ingeniería de datos. En este caso:

- **Extract:** se extrae la información bruta de la memoria de la etiqueta RFID mediante el comando `hf mf autopwn`. En concreto, este comando proporciona una salida detallada sobre la lectura y, si la lectura ha sido exitosa, sobre el contenido de la tarjeta leída. Muestra los diferentes bloques contenidos e información sobre los archivos donde vuelca la información completa.
- **Transform:** se interpreta la salida textual del lector, extrayendo los campos relevantes (ID, tipo de producto, fecha de caducidad y cantidad) y validando su formato. Además, verifica que todos estos datos cumplen con la estructura esperada y limpia la información recibida, re-formateando la lectura para dejar los datos en la forma que los espera la base de datos destino.
- **Load:** se inserta la información estructurada en una base de datos SQLite utilizada por la aplicación móvil.

El objetivo principal de este proceso es minimizar la intervención del usuario: basta con acercar una etiqueta RFID al lector para que el sistema registre el producto, su cantidad y su fecha de caducidad de forma automática. Este enfoque no solo mejora la eficiencia del sistema, sino que sienta las bases para una futura ampliación a entornos más complejos con múltiples fuentes de datos.

A continuación, se detallan las partes clave de este flujo automatizado.

4.2.1. Codificación de las etiquetas

Las etiquetas utilizadas son de tipo MIFARE Classic, con memoria distribuida en 16 sectores, cada uno con 4 bloques de 16 bytes. Esta estructura permite dividir la información de forma ordenada para que luego pueda ser extraída y procesada. Además, cada una puede ser identificada mediante una cadena en hexadecimal única, lo que nos permite reconocer de manera inequívoca de qué producto concreto se trata. En concreto, para este prototipo se ha utilizado la siguiente información codificada en hexadecimal:

- Sector 0 / bloque 3 / parte B guarda el ID del tipo de producto (de qué alimento se trata) como un entero de 16 B.
- Sector 1 / Bloque 7 / parte A contiene la fecha de caducidad codificada en formato YYYYMMDD.
- Sector 2 / Bloque 11 / parte B representa la cantidad o peso en unidades enteras en 16 B.

Este diseño es escalable y pueden utilizarse cualquiera de los bloques para añadir más información sobre el producto (información nutricional, alérgenos...).

Por ejemplo, para una etiqueta con id 1, fecha de caducidad 25-06-2025 (07E90619 en hexadecimal en formato YYYYMMDD) y cantidad 500 (01F4 en hexadecimal) el resultado de la codificación es el siguiente. Notar que hay bloques que no han sido modificados y, por tanto, contienen información aleatoria.


```

"blocks": {
  "0": "6A672C84A50804006263646566676869",
  "1": "00000000000000000000000000000000",
  "2": "00000000000000000000000000000000",
  "3": "000000000000FF078069000000000001",
  "4": "00000000000000000000000000000000",
  "5": "00000000000000000000000000000000",
  "6": "00000000000000000000000000000000",
  "7": "07E906190000FF078069000000000000",
  "8": "00000000000000000000000000000000",
  "9": "00000000000000000000000000000000",
  "10": "00000000000000000000000000000000",
  "11": "000000000000FF0780690000000001F4",
  "12": "00000000000000000000000000000000",
  "13": "00000000000000000000000000000000",
}

```

Figura 4.1: Ejemplo de codificación en memoria de una etiqueta RFID.

4.2.2. Gestión de la base de datos

La información que recoge el lector se almacena en una base de datos SQLite ubicada en la carpeta `/assets` del proyecto Android. Esta base de datos contiene una única tabla (registros) que contiene los siguientes campos correspondientes a los productos:

- `cardId`: identificador único de la tarjeta (UID del producto).
- `productID`: identificador del tipo de producto al que pertenece esa etiqueta.
- `expDate`: fecha de caducidad del producto.
- `weight`: representa la cantidad del producto en ese momento.

Por tanto, la estructura de la tabla es:

```

1 CREATE TABLE registros (
2     cardId TEXT PRIMARY KEY,
3     productId INTEGER,
4     expDate TEXT,
5     weight INTEGER
6 )
7

```

Esta solución es sencilla y eficaz para un prototipo local, aunque no está optimizada para grandes volúmenes de datos ni para sincronización multiusuario.

4.2.3. Lectura e interpretación de datos

El script principal ejecuta periódicamente el comando `hf mf autopwn`, que permite leer el contenido en bruto de las etiquetas. Posteriormente, se analiza línea por línea la salida generada, identificando los bloques que contienen información útil.

Los datos se extraen mediante expresiones regulares y se convierten en un formato estructurado. En concreto, las expresiones buscan el patrón que identifican tanto el id de la tarjeta como los tres valores escritos en los bloques, tal y como se ha detallado previamente. Para ello, el script busca los sectores, bloques y partes donde sabemos que está guardada la información. Una vez encontradas, se extrae la cadena en hexadecimal y, posteriormente, se transforman los datos al formato adecuado.

En el caso del identificador del tipo de producto y la cantidad, se pasa a un número entero y en el caso de la fecha se separa en año, mes y día y, posteriormente, se cambia a números enteros. En el caso del identificador de la tarjeta se mantiene como una cadena en hexadecimal.

Además, el programa comprueba si se trata de un identificador no presente en la base de datos y se asegura de haber obtenido todos los datos necesarios correctamente. Solo en ese caso se insertará de la siguiente manera, utilizando una instrucción de SQLite:

```
1      INSERT INTO registros (cardId, productId, expDate, weight) VALUES (?, ?,  
2      ?, ?)
```

Como el lector no es capaz de detectar de manera automática la salida de las etiquetas del rango de lectura, tal y como se esperaría en un prototipo final, fue necesario buscar una forma alternativa de eliminar los alimentos ya consumidos. La solución propuesta es una segunda lectura del producto. Es decir, si se detecta un producto ya existente en la base de datos, se elimina como forma de simular su consumo. En concreto, se utiliza la siguiente instrucción:

```
1      DELETE FROM registros WHERE cardId = ?  
2
```

Este comportamiento refleja una gestión básica del inventario: los productos entran cuando se escanean, y se eliminan cuando se consumen o detectan duplicados. Notar que para este primer prototipo se considera que los alimentos son consumidos una sola vez. Aunque este no es el comportamiento real de un usuario, las limitaciones del lector impiden reflejar este comportamiento. Un sistema más completo podría usar sensores de peso para detectar cantidades restantes, pero esto excede el alcance del presente trabajo.

Además, el script controla errores comunes, como la lectura incompleta, formatos inválidos o conflictos en el acceso a la base de datos.

4.3. Desarrollo de la aplicación móvil

La aplicación móvil desarrollada para este proyecto cumple el rol de interfaz principal entre el sistema de automatización RFID y el usuario final. Está diseñada con el fin de mostrar de forma clara y accesible el inventario doméstico de alimentos, la compra de productos y la búsqueda de recetas con los alimentos disponibles. Además, en la app también se pueden visualizar productos próximos a caducar, facilitando así la gestión inteligente de alimentos en el ámbito del hogar.

La tecnología elegida para el desarrollo fue Java utilizando el entorno de desarrollo Android Studio, probada en el emulador proporcionado por este último, utilizando Android como sistema operativo.

4.3.1. Esquema general

La aplicación se estructura en torno a tres funcionalidades principales: gestión del inventario, compra de productos y visualización de recetas. Estas funcionalidades se distribuyen en varias pantallas conectadas entre sí mediante un menú en la parte inferior.

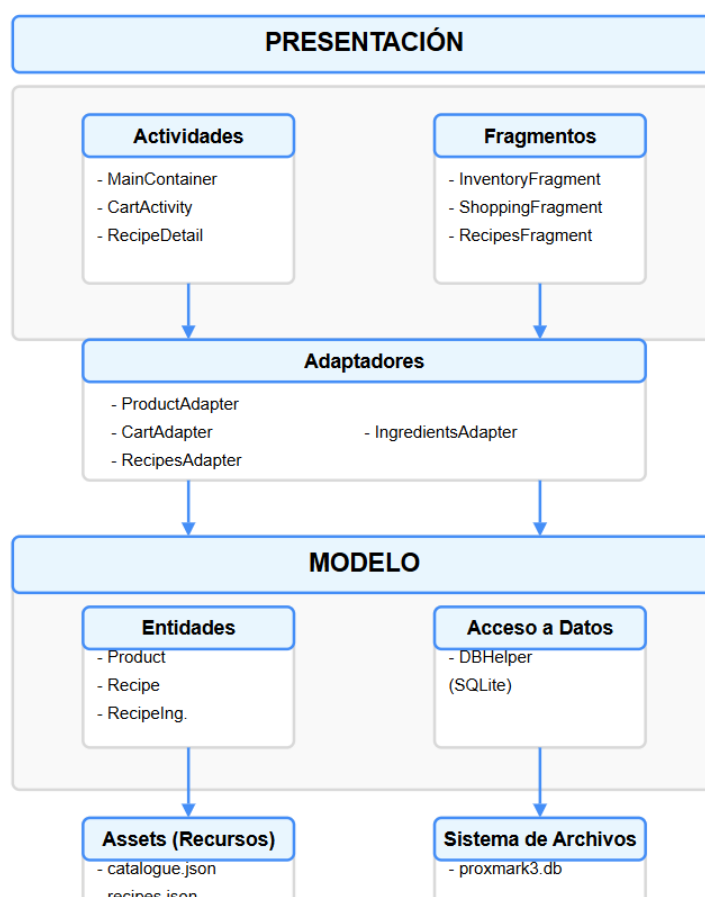


Figura 4.2: Arquitectura general del sistema.

4.3.2. Diseño de la interfaz de usuario y estructura de la aplicación

El diseño de la interfaz se centró en tres principios: simplicidad, claridad y funcionalidad. Buscando que el usuario encuentre de manera rápida y cómoda los productos que tiene disponibles y comprarlos directamente cuando se acaban. Además, el objetivo es que la navegación entre pantallas sea de forma fluida e intuitiva. Para ello, se han diseñado las pantallas definidas en el Anexo 2 (*Mapa de navegación de la aplicación*).

En cuanto a la arquitectura interna, se ha seguido una estructura basada en el modelo MVC (Modelo-Vista-Controlador), que se adapta de forma natural a las aplicaciones Android más sencillas. Esta arquitectura interna facilita el mantenimiento, la escalabilidad y la separación de responsabilidades dentro de la aplicación.

- **Modelo:** el modelo se implementa a través de clases de identidad como Product.java, Recipe.java o RecipeIngredient.java, que representan el dominio de negocio de la aplicación. La capa de persistencia está gestionada por la clase DBHelper.java que proporciona una interfaz para interactuar con la base de datos usando SQLite. Esto permite optimizar consultas, tener un control preciso sobre la carga de datos y evitar dependencias de gestión de los datos.
- **Vista:** las vistas se implementan mediante layouts desarrollados en XML y se complementan con Fragments como InventoryFragment.java, ShoppingFragment.java y RecipesFragment.java. Esta aproximación ofrece una separación de los elementos de UI y la programación de los mismos. Además, se hace uso de elementos como RecyclerView, que facilitan el diseño de listas o Fragments que permiten implementar una navegación de tipo *tab* que mantiene el estado entre los cambios de pantalla.

- Controlador: esta capa incluye actividades como `CartActivity.java` y `MainContainerActivity.java`, junto con adaptadores como `ProductAdapter.java` y `RecipesAdapter.java`. Su misión es conectar el modelo y la vista, procesando las interacciones del usuario y solicitando los datos apropiados al modelo. Estos componentes, además, coordinan el flujo de la aplicación y manejan la lógica de visualización.

Esta arquitectura ha sido elegida debido a que:

- Permite un flujo de datos sencillo, simplificando la detección y corrección de errores.
- Permite las modificaciones de manera ágil, haciendo que un cambio en los requisitos del sistema sea fácil de implementar.
- Se alinea con los recursos de este sistema, adaptándose a las necesidades específicas de la tecnología RFID como la gestión de los datos locales.
- Presenta un equilibrio entre la eficiencia y la complejidad del sistema.

4.3.3. Comunicación con el lector e integración con bases de datos

Aunque la aplicación móvil no se comunica directamente con el lector RFID, sí depende de los datos que este genera mediante el script automatizado. Por tanto, se diseñó una estrategia de integración simple y funcional: compartir una base de datos local en formato SQLite entre el script y la app Android.

La base de datos (`proxmark3.db`) se encuentra ubicada en la carpeta `/assets` de la aplicación. Esto permite que esté incluida en el paquete de instalación (`.apk`) y sea accesible en tiempo de ejecución desde el dispositivo o el emulador Android.

Para acceder a la base de datos desde la app, se implementó una clase `DBHelper` que utiliza la librería `SQLiteOpenHelper` que se encarga de:

- Abrir la base de datos desde los assets.
- Realizar consultas (`SELECT`) para obtener la información necesaria de los productos presentes en la base de datos.

Aunque esta solución funciona correctamente para un prototipo local, para una versión final de sistema debería hacerse una sincronización de los datos mediante una arquitectura cliente-servidor o mediante servicios de sincronización utilizando separación de acceso a datos mediante una API REST.

4.3.4. Persistencia y datos

La aplicación usa tres fuentes de datos:

1. `Catalogue.json`: contiene productos estándar con toda su información relevante (identificador del alimento, nombre, cantidad por unidad, medida, foto y link).
2. `Recipe.json`: estructura las recetas, ingredientes e instrucciones de forma detallada.
3. `Proxmark3.db`: almacena la información persistente del usuario (productos, cantidades y fechas).

Los ID entre los productos del catálogo, inventario y recetas están sincronizados para facilitar el cruce de datos entre módulos. El siguiente esquema refleja los datos contenidos por la aplicación:



Figura 4.3: Datos que guarda la aplicación.

4.4. Consideraciones sobre seguridad

Aunque el sistema desarrollado es funcional para un entorno controlado, en su uso real sería necesario tener en cuenta varios aspectos de seguridad. En particular, las etiquetas utilizadas (MIFARE Classic) presentan algunas vulnerabilidades. En numerosos estudios [7] [5], se ha probado que estas etiquetas son vulnerables a ataques de clonación, acceso no autorizado y recuperación de claves mediante técnicas de fuerza bruta.

En concreto, estas tarjetas utilizan un sistema de cifrado CRYPTO1 con claves generadas aleatoriamente. Sin embargo, este tipo de cifrado no se considera seguro, ya que puede ser fácilmente vulnerado. Para una versión final del sistema, sería más seguro utilizar otro tipo de etiquetas con protocolos de seguridad más avanzados que ofrezcan autenticación basada en estándares más modernos, comunicación cifrada y mayor resistencia a clonación. Un ejemplo de este tipo es MIFARE Plus, que utiliza AES o MIFARE DESFire EV2/EV3 que utiliza AES y EAL5+ como sistemas de cifrado.

Además, en una versión real del sistema con múltiples accesos concurrentes (desde distintos dispositivos), sería necesario incorporar mecanismos de autenticación y autorización, para verificar la identidad y definir los permisos de cada perfil, respectivamente. Esto aseguraría que no se producen accesos no autorizados, así como proteger la integridad de los datos.

Capítulo 5

Evaluación y Pruebas

La evaluación y las pruebas de la aplicación se llevaron a cabo en varias fases con el objetivo de asegurar tanto la funcionalidad como la usabilidad del sistema. A continuación, se detallan las distintas etapas del proceso de prueba, así como los ajustes realizados a partir de los resultados obtenidos.

5.1. Fase 1: Pruebas iniciales de desarrollo y pruebas informales con usuarios

El primer paso del proceso de evaluación fue la prueba de la aplicación que realicé, como desarrolladora, para asegurarme de que las funcionalidades básicas del sistema estaban operativas. Durante esta fase, se verificó que las interacciones principales de la aplicación funcionaban correctamente, como la navegación entre pantallas, la consulta del inventario, la visualización de recetas y la actualización de datos.

Posteriormente, se llevó a cabo una primera fase de pruebas informales con un grupo reducido de usuarios, los cuales no tenían experiencia previa con la aplicación. Estos usuarios interactuaron con la app en un entorno controlado y proporcionaron comentarios espontáneos sobre la navegación y la estructura general de la interfaz. A raíz de estas pruebas, se tomaron algunas decisiones clave sobre el diseño de la aplicación:

- Se decidió modificar la distribución inicial de la pantalla. Originalmente, la aplicación contaba con una pantalla inicial que incluía tres botones para acceder al inventario, recetas y compras. Sin embargo, tras recibir retroalimentación de los usuarios, se optó por reestructurar la navegación utilizando un menú inferior, lo que facilitó una experiencia de usuario más fluida y accesible.

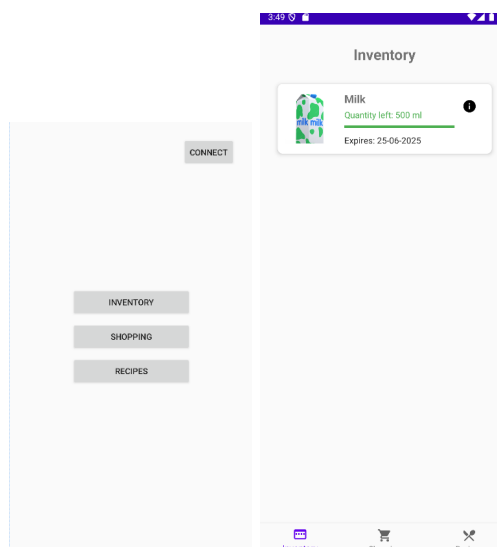


Figura 5.1: Modificación de la paleta de colores de la aplicación.

- Además, se implementó una mejora en la gestión de los productos: antes, los productos se identificaban solo mediante un único ID. En esta fase, se determinó que era necesario dividir el ID de la tarjeta RFID del ID del producto. Este cambio permitiría que múltiples productos del mismo tipo pudieran estar presentes en el inventario simultáneamente, mejorando así la flexibilidad del sistema.

5.2. Fase 2: Pruebas formales de usuario

La segunda fase del estudio consistió en una serie de pruebas de usabilidad realizadas con un grupo reducido de usuarios externos, con el objetivo de obtener datos más estructurados sobre el uso real de la aplicación. En total participaron seis personas, seleccionadas para representar dos perfiles diferenciados: tres jóvenes con experiencia previa en el uso de tecnologías móviles y tres adultos con conocimientos tecnológicos limitados. Este equilibrio permitió obtener una visión más amplia sobre la accesibilidad y usabilidad de la aplicación en distintos contextos.

Las sesiones se realizaron de forma individual, utilizando un simulador que emulaba el comportamiento de la aplicación en un dispositivo móvil. A cada participante se le entregó un documento con una serie de tareas representativas del uso previsto de la aplicación (ver productos disponibles, añadir o eliminar artículos del carrito, consultar recetas y comprobar disponibilidad de ingredientes, entre otras). Además, se les indicó que verbalizaran en voz alta sus razonamientos mientras interactuaban con la interfaz, siguiendo la técnica de *think aloud*. Esta metodología, aunque tiende a aumentar ligeramente el tiempo de ejecución de las tareas, permitió identificar con mayor claridad los puntos de confusión o duda.

Durante las sesiones se recogieron métricas cuantitativas tales como el tiempo requerido para completar cada tarea, la tasa de éxito (entendida como la finalización correcta sin ayuda), así como el número de errores cometidos. Los resultados detallados de estas métricas se incluyen a continuación. La observación directa permitió además anotar comentarios espontáneos de los usuarios, reacciones gestuales, y patrones de navegación que no siempre se traducen en errores pero sí en indicios de mejora potencial.

Tanto el documento entregado a los participantes como las anotaciones recogidas durante las pruebas están recogidas en el Anexo 3 (*Test de Usabilidad - Aplicación para la gestión de alimentos en casa*).

En concreto, la siguiente tabla que resume algunas medidas cuantitativas que recogen los resultados obtenidos durante los tests de usabilidad.

Tarea	Tiempo medio (s)	Errores medios	Tasa de éxito (%)
Tarea 1: Productos disponibles	77.5	0.5	100
Tarea 2: Editar la compra	101.67	1.5	83.3
Tarea 3: Detalles del producto	58.33	0	100
Tarea 4: Ver receta y comprobar	47.5	0	100

Cuadro 5.1: Resumen de métricas por tarea (media entre los 6 participantes).

Los resultados obtenidos reflejan un rendimiento general bastante positivo en el uso de la aplicación. Todas las tareas fueron mayormente completadas con éxito. La tarea relacionada con la compra fue la que presentó mayor dificultad para los usuarios, lo que sugiere que esta funcionalidad necesitaba alguna mejora y mayor claridad en sus elementos. A pesar de ello, y teniendo en cuenta que se utilizó un método que ralentiza el proceso, los tiempos medios de ejecución fueron razonables.

En conjunto, estos resultados validan que la aplicación es funcional y usable por usuarios con distintos conocimientos tecnológicos.

A partir del análisis de la información cualitativa ofrecida por los usuarios, se identificaron varias oportunidades de mejora en el diseño de la interfaz, que condujeron a los siguientes ajustes funcionales y estéticos:

- Modificar el código de colores utilizado para las fechas de caducidad. En una primera versión, se utilizaba el color rojo para indicar que el alimento estaba a punto de caducarse, sin embargo, varios usuarios entendían que este color indicaba que ya estaba caducado. Por lo tanto, en la nueva versión se utiliza el color rojo para indicar que ya está caducado y amarillo/naranja para indicar la proximidad de la fecha.

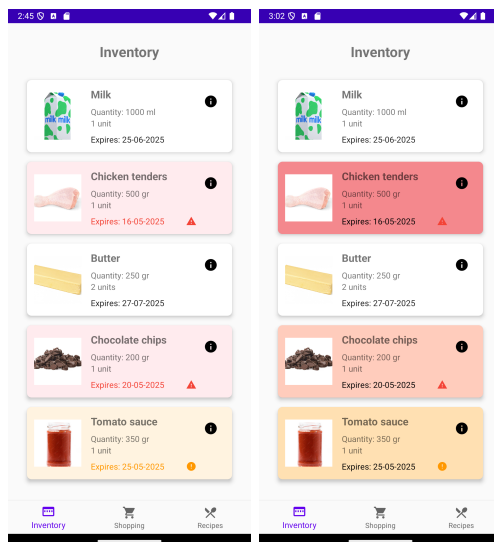


Figura 5.2: Modificación de la paleta de colores para indicar la caducidad de los productos.

- Modificar la paleta de colores de la aplicación para que fuera coherente con el logo de la aplicación. En concreto, se seleccionó el color verde como color principal ya que este es, a menudo, utilizado para representar productos frescos.

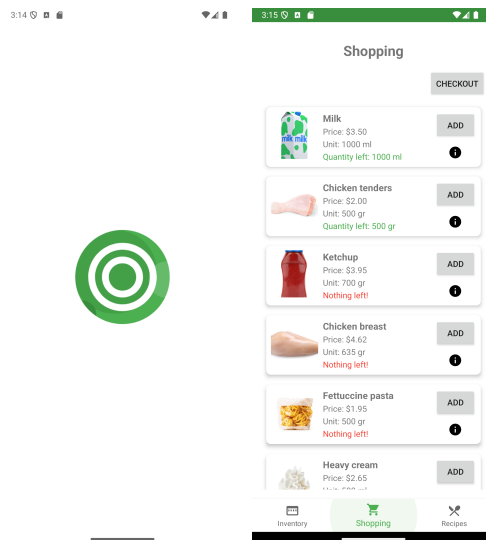


Figura 5.3: Modificación de la paleta de colores de la aplicación.

- Indicar de manera más clara cómo realizar la revisión de la compra. Varios usuarios tuvieron dificultad para encontrar esta funcionalidad del sistema. Como solución se propuso cambiar el color del botón para que destacase más y fuese diferente al resto de botones presentes en la misma pantalla.

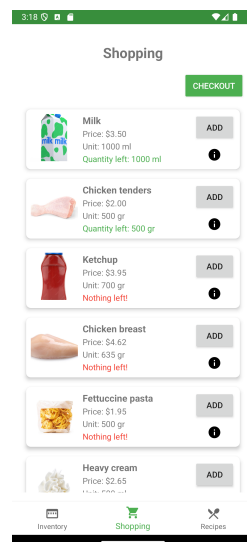


Figura 5.4: Modificación del botón para revisar la compra.

- Se solucionaron algunos problemas menores encontrados durante el test. Por ejemplo, se encontró algún link que no correspondía adecuadamente al producto.
- Algunos usuarios propusieron mejoras para la aplicación como por ejemplo: la posibilidad de buscar recetas por producto o tipo de receta, poder añadir al carrito productos directamente desde la lista de ingredientes de una receta...

5.3. Pruebas automáticas

Con el objetivo de garantizar la correcta funcionalidad de la aplicación y probar su robustez, se desarrolló una batería de pruebas automatizadas usando JUnit. Estas pruebas se centran

en la gestión de la base de datos, probando tanto los principales casos de uso previstos como situaciones más extremas, incluyendo también pruebas sobre condiciones de carga y concurrencia. A continuación, se describen en detalle las distintas pruebas implementadas.

5.3.1. Pruebas básicas de inserción, recuperación y actualización

Se verificó que:

- Los registros se insertan correctamente.
- La recuperación de atributos (peso y fecha de caducidad) por identificador de producto devuelve los valores esperados. En el caso de múltiples registros con el mismo tipo, se comprobó que acumula correctamente el peso total y devuelve la fecha de caducidad más cercana.
- El sistema responde con valores adecuados en el caso de un identificador inexistente (peso 0 y fecha null).
- Se sobrescriben correctamente la fecha y el peso tras una operación de actualización.
- Se eliminan correctamente los productos.

5.3.2. Pruebas de rendimiento

Para evaluar el rendimiento y la escalabilidad del sistema, se realizó una prueba de inserción de 500 registros únicos. Además, se midió el tiempo necesario y se comprobó que era razonable, en concreto, menos de 20 segundos para toda la carga.

5.3.3. Pruebas de acceso concurrente

El sistema fue sometido a condiciones de acceso simultáneo. Para ello, se ejecutaron tres hilos concurrentes, evaluando:

- La robustez del acceso a la base de datos sin corrupción de datos.
- El porcentaje de operaciones exitosas es superior al 70 %.

5.4. Conclusiones

A partir de las pruebas realizadas (tanto con usuarios como las pruebas automáticas), se llegó a una serie de conclusiones. Estos resultados fueron fundamentales para el proceso de refinamiento de la aplicación y contribuyeron a mejorar tanto su funcionalidad como su experiencia de usuario. Asimismo, se abordaron aspectos de rendimiento y robustez.

En cuanto a la evaluación con usuarios, se observó una clara diferencia en la facilidad de uso entre los dos perfiles. Los participantes más jóvenes, con mayor familiaridad con la tecnología, completaron las tareas de forma más fluida y rápida. En cambio, los usuarios adultos con menos experiencia tecnológica requirieron más tiempo y cometieron más errores, especialmente en tareas que implicaban navegación entre pantallas. Esto puso de manifiesto la necesidad de reforzar elementos como la consistencia visual, el uso de etiquetas claras y la accesibilidad general del diseño.

Además, los tiempos registrados por los participantes deben ser interpretados con cautela, ya que el protocolo incluía la técnica de pensar en voz alta. Esto ralentizó intencionadamente la ejecución de las tareas, pero proporcionó información valiosa sobre la toma de decisiones de los usuarios, sus dudas y los elementos de la interfaz que les generaban confusión.

Las principales mejoras aplicadas tras la primera fase incluyeron la simplificación del menú principal, el rediseño de algunos iconos y la mejora del contraste visual. Estas modificaciones tuvieron un efecto positivo en la segunda ronda de pruebas, donde se observaron menos errores y una navegación más fluida.

El número reducido de participantes (6 en total) limita la generalización de los resultados. Además, al tratarse de un entorno simulado y no del uso real en un contexto doméstico, algunas interacciones pueden diferir respecto al uso habitual. Finalmente, el hecho de que un miembro del equipo estuviera presente durante la prueba podría haber influido en el comportamiento de los usuarios (efecto observador).

Aun así, los resultados obtenidos permiten identificar puntos críticos de mejora y confirmar que, en líneas generales, la aplicación resulta comprensible y útil para los perfiles a los que va dirigida.

En paralelo, se llevaron a cabo las pruebas automáticas centradas en el módulo de acceso a la base de datos. Estas pruebas incluyeron validaciones unitarias de inserción, recuperación, actualización y pruebas de carga y acceso concurrente. Los resultados confirmaron la correcta lógica de los datos, así como su estabilidad y escalabilidad.

Capítulo 6

Conclusiones

Este Trabajo ha demostrado la viabilidad tanto técnica como funcional de la aplicación de la tecnología RFID para la gestión de alimentos en el entorno doméstico. En concreto, se ha desarrollado un prototipo completo capaz de automatizar el registro de productos, almacenar la información en una base de datos local y presentarla al usuario a través de una aplicación móvil.

Específicamente, este sistema permite registrar los productos acercando su etiqueta al lector, visualizarlos en tiempo real a través de un inventario digital, identificar aquellos que tienen una fecha de caducidad próxima y consultar recetas, comprobando si los ingredientes están disponibles. Todo ello se integra bajo una arquitectura basada en un flujo de tipo ETL que separa las responsabilidades de lectura, procesamiento y almacenamiento de datos. El usuario accede a esos datos mediante una aplicación que ha sido desarrollada pensando en la usabilidad para distintos usuarios. Para comprobarlo, se ha llevado a cabo una fase de evaluación mediante pruebas con usuarios reales. Estas han permitido detectar puntos de mejora y validar su uso.

A diferencia de otras aplicaciones existentes, esta propuesta automatiza por completo el registro y control de productos, eliminando una de las principales barreras de uso: la introducción manual de los datos. Esto no solo mejora la experiencia del usuario en la cocina, sino que contribuye de forma directa a reducir el desperdicio alimentario.

6.1. Limitaciones

Entre las principales limitaciones del prototipo presentado, destaca el uso de un lector RFID pensado para realizar una prueba de concepto. Es decir, la elección del lector Proxmark3 impide la lectura múltiple de etiquetas y su alcance es limitado. Sin embargo, esto no impide demostrar el potencial de la solución desarrollada.

Además, este sistema parte de la situación ideal futura donde los propios supermercados incluyen este tipo de etiquetas en sus productos. Esto ocurre ya en algunas tiendas alimenticias, pero su uso aún no está generalizado.

Otra limitación importante es que el sistema, por el momento, está concebido para un solo usuario o dispositivo, es decir, un único acceso a la base de datos en local. No contempla un uso multiusuario, como podría ocurrir en hogares donde varias personas quisieran ver el inventario desde varios teléfonos. Para mejorar este aspecto, se debería implementar una arquitectura cliente-servidor con una base de datos centralizada utilizando un sistema de autenticación para el acceso de usuarios.

Por último, otro aspecto con margen de mejora es la concurrencia. El sistema actual utiliza una base de datos que se guarda en la carpeta `/assets` del proyecto Android. Esto hace que cuando el script modifica los datos no se reflejan automáticamente en la aplicación. Esto ocurre porque los archivos contenidos en esa carpeta se copian al sistema de archivos del emulador durante la instalación de la APK. Por tanto, cualquier modificación posterior en el archivo original de la base de datos no se ve reflejada hasta que se vuelve a compilar y reinstalar la aplicación en el

emulador.

6.2. Líneas futuras

De cara a futuras versiones, el sistema podría evolucionar en varias direcciones:

- Reemplazar el lector actual por uno con mayor alcance y preparado para el uso simultáneo de varias etiquetas y usuarios.
- Implementar una sincronización de datos y acceso remoto.
- Ampliar la funcionalidad de la aplicación: distintos perfiles para usuarios, creación de nuevas recetas a partir de ingredientes disponibles, seguimiento de la compra, introducir Inteligencia Artificial para sugerir compras automáticas o recetas según las necesidades alimentarias del usuario...

6.3. Conclusión final

En conjunto, este proyecto ha permitido unir hardware, software y experiencia en una solución funcional y realista. A pesar de las existentes barreras técnicas, este proyecto demuestra que la integración de RFID en el hogar no solo es viable, sino que supone un cambio significativo en la forma en la que gestionamos nuestra alimentación diaria. Esto tiene un impacto directo en la sostenibilidad y bienestar del usuario.

En resumen, este proyecto representa un primer paso hacia una cocina más inteligente, más eficiente y más alineada con las presentes necesidades en este ámbito.

Bibliografía

- [1] Eva Gotor Carrasco. *Estado del Arte en Tecnologías RFID*. Inf. téc. Trabajo académico. Madrid: Escuela Universitaria de Informática, Universidad Politécnica de Madrid, jun. de 2009.
- [2] Dipole RFID. *Lector RFID UHF Impinj Speedway Revolution R420*. Accedido el 17 de abril de 2025. 2025. URL: <https://www.dipolerfid.es/producto/Lector-RFID-UHF-Impinj-Speedway-Revolution>.
- [3] Dipole RFID. *Tipos de Sistemas RFID*. Consultado el 17 de abril de 2025. s.f. URL: <https://www.dipolerfid.es/blog-rfid/Tipos-Sistemas-RFID>.
- [4] FEIG ELECTRONIC GmbH. *ID MRU102: Short Range UHF RFID Universal Reader*. Accedido el 17 de abril de 2025. 2025. URL: <https://www.feig.de/en/rfid-and-barcode-systems/id-mru102/>.
- [5] Flavio D. Garcia et al. «Dismantling MIFARE Classic». En: *Computer Security – ESORICS 2008*. Ed. por Sushil Jajodia y Javier Lopez. Vol. 5283. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2008, págs. 97-114. DOI: [10.1007/978-3-540-88313-5_7](https://doi.org/10.1007/978-3-540-88313-5_7).
- [6] JADAK, a Novanta Company. *ThingMagic M6e Nano UHF RAIN RFID Module*. Accedido el 17 de abril de 2025. 2025. URL: <https://www.jadaktech.com/product/thingmagic-m6e-nano-uhf-rain-rfid/>.
- [7] Gerhard de Koning Gans, Jaap-Henk Hoepman y Flavio D. Garcia. «A Practical Attack on the MIFARE Classic». En: *Smart Card Research and Advanced Applications (CARDIS 2008)*. Ed. por Gérard Grimaud y François-Xavier Standaert. Vol. 5189. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2008, págs. 267-282. DOI: [10.1007/978-3-540-85893-5_20](https://doi.org/10.1007/978-3-540-85893-5_20).
- [8] Naciones Unidas. *Día Internacional de Concienciación sobre la Pérdida y el Desperdicio de Alimentos*. Accedido el 17 de abril de 2025. 2024. URL: <https://www.un.org/es/observances/end-food-waste-day>.
- [9] NoWaste App. *NoWaste: Organiza y gestiona fácilmente los alimentos de tu hogar*. Accedido el 17 de abril de 2025. 2025. URL: <https://www.nowasteapp.com/>.
- [10] RfidResearchGroup. *Proxmark3 Releases*. Accedido el 17 de abril de 2025. 2025. URL: <https://github.com/RfidResearchGroup/proxmark3/releases>.
- [11] Samsung Electronics. *Cómo usar mi refrigerador con SmartThings*. Última actualización: 23 de octubre de 2024. Accedido el 17 de abril de 2025. 2024. URL: <https://www.samsung.com/latin/support/home-appliances/how-to-use-my-refrigerator-with-smartthings/>.
- [12] Smarter. *FridgeCam – Haz tu nevera inteligente*. Accedido el 17 de abril de 2025. 2025. URL: <https://smarter.am/es/products/smarter-fridgecam>.

Anexo 1

Detalles técnicos del uso de Proxspace para Proxmark3

Este anexo recoge los comandos utilizados durante el desarrollo del prototipo para la lectura, escritura y análisis de etiquetas RFID mediante el lector Proxmark3 utilizando el entorno proxspace.

Preparación del entorno

Utilizando la terminal de comandos desde la carpeta con el software descargado (clonando el repositorio de GitHub) ejecutamos el siguiente comando para iniciar el software.

```
1 ./runme64.bat
```

Accedemos al entorno del Proxmark3

```
1 cd proxmark3
```

Compilamos e indicamos el tipo de plataforma con el que vamos a utilizar, en caso del lector utilizado para este proyecto es PM3:

```
1 make clean && make PLATFORM=PM3GENERIC
```

Flasheamos el cliente mediante el siguiente comando:

```
1 ./pm3-flash-all
```

Nos aparecerán una serie de características de compilación, si tenemos el dispositivo correctamente conectado nos indicará si está siendo ejecutado con COM3 o COM4 y debemos guardar esa información para ejecutar el cliente en el puerto correspondiente:

```
1 cd client
2 ./proxmark3 com3
```

Comandos generales

- hw tune: sirve para verificar las antenas disponibles.

Comandos de lectura

Existen varios comandos que nos permiten leer tarjetas:

- auto: verifica tanto alta como baja frecuencia y por la salida estándar podemos ver información sobre la tarjeta leída (si la hay).
- lf search: igual que auto pero solo para baja frecuencia.

- `hf searc`: igual que `auto` pero solo para alta frecuencia.
- `hf mf autopwn`: revisa el tipo de tarjeta leída (de alta frecuencia) y crea cuatro archivos donde se guarda la información de la lectura. El más importante de estos es un archivo `.json` que contiene el uid de la tarjeta, el contenido por bloques y las claves de lectura y escritura de cada uno de los bloques, es indispensable para la escritura.
- `lf mf autopwn`: igual que el anterior pero para baja frecuencia.
- `hf mf rdbl -blk <n_bloque>-k <clave>`: sirve para leer un bloque utilizando la clave (disponible en el `.json` previamente mencionado).
- `lf mf rdbl -blk <n_bloque>-k <clave>`: igual que el anterior para baja frecuencia.

Comandos de escritura

Para la escritura existe de manera similar varios comandos:

- `hf mf wrbl -blk <n_bloque>-k <clave>-d <datos>`: escribe los 16B de un bloque utilizando la clave de escritura del bloque, es importante que la tarjeta tenga permiso de escritura en este bloque.
- `lf mf wrbl -blk <n_nloque>-k <clave>-d <datos>`: igual que el anterior pero bajas frecuencias.

Anexo 2

Mapa de navegación de la aplicación

A continuación, se muestra un mapa de navegación con la versión final de las pantallas principales de la aplicación desarrollada.

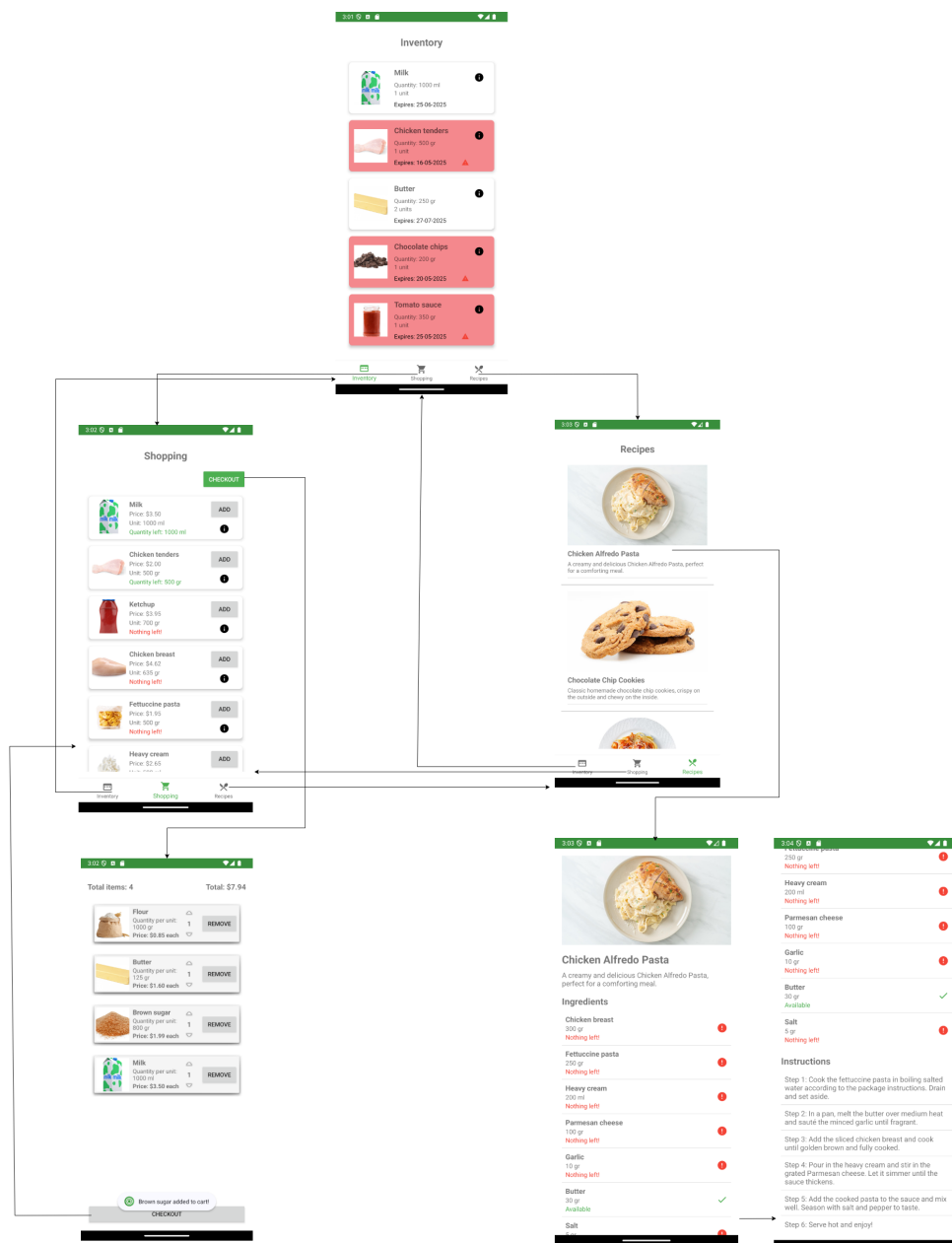


Figura 6.1: Mapa de navegación de la aplicación.

Anexo 3

Test de Usabilidad - Aplicación para la gestión de alimentos en casa

Consentimiento Informado

Yo, _____, doy mi consentimiento para ser grabado en **voz** **video** **ambos** durante la realización del test de usabilidad de esta aplicación en desarrollo. Entiendo que estas grabaciones se usarán únicamente para analizar y mejorar el diseño de la aplicación.

Firma: _____

Fecha: _____

Contexto de la Aplicación

Vas a participar en un test de usabilidad sobre una aplicación que estamos desarrollando. Esta aplicación te muestra qué productos tienes en casa (como comida o bebidas), sus cantidades, fechas de caducidad, etc. También permite hacer la compra online de forma sencilla y consultar recetas para ver si puedes prepararlas con lo que tienes. Todo esto funciona gracias a sensores y tecnología de radio frecuencia, cuyos detalles no es necesario conocer para la realización de este test de usabilidad.

Explicación del Test

Vas a usar un ordenador con un simulador que imita el uso de un móvil. En la pantalla verás unos botones que sirven para volver atrás o salir. Son importantes para poder moverte por la aplicación. En la imagen siguiente puedes ver dónde están.

También recibirás este documento, que contiene una serie de tareas. Léelas con calma y realiza cada una por tu cuenta.

Habrà una persona presente únicamente para ayudarte si surge algún problema técnico. No resolverá dudas sobre cómo funciona la aplicación. Aquí no hay respuestas buenas o malas: lo que buscamos es detectar puntos de confusión y mejorar el diseño.

Te pedimos que pienses en voz alta mientras haces las tareas. Si te bloqueas, intenta reflexionar: “¿Qué haría ahora? ¿Qué me está diciendo la aplicación? ¿Qué creo que está pasando?”

Tareas del Usuario

Objetivo General

Estamos diseñando una app para ayudarte a llevar un control de los alimentos que tienes en casa.

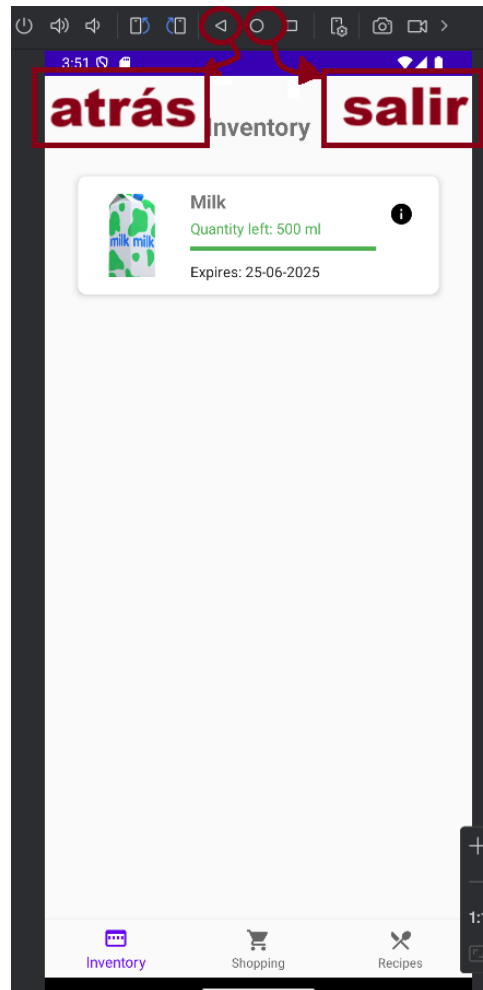


Figura 6.2: Ejemplo de pantalla con los botones de navegación que vas a utilizar

Tarea 1: Ver qué productos tienes en casa

Situación: Vives solo/a y quieres evitar tirar comida. Quieres revisar qué productos tienes, cuánto queda de cada uno y cuándo caducan, para planificar las comidas de esta semana.

Inicio: Empiezas en la pantalla principal de la aplicación.

Tarea: Mira qué productos tienes ahora mismo, cuánta cantidad queda y qué fecha de caducidad tienen. ¿Hay alguno caducado?

Fin: Cuando termines, vuelve a la pantalla principal y avisa de que has acabado.

Tarea 2: Añadir productos a tu compra

Situación: Estás haciendo la compra semanal. Ya tienes una lista de lo que necesitas. Quieres añadir leche y harina a la compra, pero luego te das cuenta de que ya tienes leche, así que decides quitarla.

Inicio: Empiezas en la pantalla principal.

Tarea: Añade dos productos al carrito y luego elimina uno de ellos.

Fin: Cuando termines, vuelve a la pantalla principal.

Tarea 3: Ver más detalles sobre un producto

Situación: Estás dudando si comprar alitas de pollo o pechugas de pollo. Quieres saber más sobre uno de los dos (de dónde viene, qué cantidad incluye, etc.).

Inicio: Empiezas en la pantalla donde puedes ver los productos disponibles.

Tarea: Busca más información sobre uno de los productos.

Fin: Cuando termines, vuelve a la pantalla principal.

Tarea 4: Ver una receta y comprobar si puedes cocinarla

Situación: Es sábado por la mañana y te apetece preparar un plato de pasta. Quieres ver una receta y comprobar si tienes todo lo necesario para cocinarla.

Inicio: Estás en la pantalla principal.

Tarea: Mira la lista de recetas, elige una de pasta y revisa qué ingredientes te faltan o tienes en poca cantidad.

Fin: Cuando termines, vuelve a la pantalla principal.

Resultados Tests de Usuario

Resumen por Usuario

Usuario 1 JAVIER

Descripción del usuario: 20 años
estudiante

Tarea	Completada	Observaciones o problemas encontrados
Tarea 1	Si 00:50	fecha de caducidad en rojo ¿caducado? ¿pantalla principal?
Tarea 2	Si 00:20	¿Bajar para ver más productos?
Tarea 3	Si 01:00	Problemas de conexión
Tarea 4	Si 00:45	Todo OK

Observaciones generales:

Respuesta algo lenta de la aplicación

Usuario 2 *SDRA*

Descripción del usuario: *28 años
abogada*

Tarea	Completada	Observaciones o problemas encontrados
Tarea 1	<i>Si</i> <i>03:00</i>	<i>todo ok</i>
Tarea 2	<i>Si</i> <i>03:30</i>	<i>difficultad para encontrar el comito de compra</i> <i>confusión entre productos</i>
Tarea 3	<i>Si</i> <i>00:20</i>	<i>todo ok</i> <i>(Problemas con el link)</i>
Tarea 4	<i>Si</i> <i>00:35</i>	<i>todo ok</i>

Observaciones generales:

Ordenar productos por fecha de caducidad
Recetas por secciones

Usuario 3 **MARIA JOSÉ**

Descripción del usuario: **58 años**
administrativa

Tarea	Completada	Observaciones o problemas encontrados
Tarea 1	SC 00:30	Dudas con caducidad (color rojo)
Tarea 2	03:00 SC (con dificultad)	Dificultad para encontrar tienda Dificultad para encontrar cómo y dónde eliminar? Los productos comprados no aparecen
Tarea 3	SC 01:00	todo ok
Tarea 4	SC 01:00	todo ok

Observaciones generales:

Cambiar el color al botón de checkout
Mejorar productos en camino

Usuario 4 CARLOS

Descripción del usuario: 22 años
estudiante

Tarea	Completada	Observaciones o problemas encontrados
Tarea 1	Si 00:45	todo ok
Tarea 2	Si 01:30	eliminar desde el catálogo?
Tarea 3	Si 01:30	problemas técnicos con el botón de info volver a la app?
Tarea 4	Si 00:40	todo ok

Observaciones generales:

Usuario 5 JUAN

Descripción del usuario: 57 años
funcionario

Tarea	Completada	Observaciones o problemas encontrados
Tarea 1	Si 01:40	confusión con colores (rojo → caducado)
Tarea 2	Si 01:30 (Más o menos)	dificultad para encontrar henda no finalizada la compra
Tarea 3	Si 01:00	link erróneo?
Tarea 4	Si 01:00	todo ok

Observaciones generales:

comprar desde la lista de ingredientes de la receta
Propiedades nutricionales de ingredientes

Usuario 6 **ANDBEL**

Descripción del usuario: **56 años**
funcionaria

Tarea	Completada	Observaciones o problemas encontrados
Tarea 1	SC 03:00	¿seleccionar productos? confusión con tarea
Tarea 2	SC 02:20 (con dificultad)	confusión con botón de información para borrar productos la cantidad es 1 no deja quitar con flechas
Tarea 3	SC 01:00	todo ok
Tarea 4	SC 00:45	todo ok

Observaciones generales:

Anexo 4

Control de esfuerzos

La siguiente tabla recoge un resumen aproximado del tiempo dedicado a cada parte del trabajo:

Fase	Horas	Observaciones
Investigación inicial y documentación técnica	45	Estudio de la tecnología RFID, análisis de soluciones existentes, y planteamiento general.
Diseño del sistema y arquitectura	20	Planificación de estructura, codificación de etiquetas, base de datos, y navegación.
Estudio del lector	25	Búsqueda de lectores, investigación sobre el Proxmark3, descarga y estudio del software asociado.
Desarrollo del script RFID	55	Lectura con Proxmark3, parseo, almacenamiento y control de errores.
Desarrollo de la aplicación móvil	82	Interfaz con fragments, integración con base de datos, funcionalidades de recetas, carrito e inventario.
Creación y gestión de datos	16	Estructuración de archivos JSON con productos y recetas.
Pruebas funcionales y test de usabilidad	47	Ejecución de pruebas, recolección de métricas, y análisis de resultados.
Redacción de la memoria y anexos	55	Elaboración del documento en LaTeX, gráficos, esquemas, referencias y anexos técnicos.
Planificación, revisión y entrega final	27	Organización del trabajo, control de versiones, revisión general...
Total	372	

Cuadro 6.1: Estimación de tiempo dedicado por fase del proyecto