

APLICACIÓN MOVIL SMART PANTRY

DESARROLLO DE APLICACIONES
MULTIPLATAFORMA 2º

AUTOR: UNAI SUÁREZ CARBALLO



CPIFP LOS ENLACES
TUTOR: CELIA GARCIA

9 DE ENERO DE 2024



Contenido

Contenido	2
1. Documento de propuesta de proyecto	5
1.1. Título del proyecto	5
1.2. Título corto del proyecto.....	5
1.3. Contexto breve	5
1.4. Objetivo breve	5
2. Documento descripción del proyecto	5
2.1. Contexto del proyecto.....	5
2.1.1. Ámbito y entorno	5
2.1.2. Destinatarios	5
2.2. Objetivo del proyecto	5
3 Documento de Acuerdo del proyecto.....	6
3.1 . Requisitos funcionales y no funcionales / Historias de usuario	6
3.2 Tareas.....	14
3.3 Metodología a seguir para la realización del proyecto	15
3.4 Presupuesto (gastos, ingresos, beneficios).....	16
3.5 Prevención de riesgos	17
4 Documento de análisis y diseño	18
4.1 Modelo de datos. Análisis y diseño de base de datos.....	18
4.1.1. Diagrama E/R.....	18
4.1.2. Esquema relacional estándar	19
4.1.3. Normalización	20
4.1.4. Dominios de atributos.....	20
4.2 Análisis y diseño del sistema funcional (Diagramas que procedan como el de casos de uso, de clases, de secuencia, de actividad, de estados, de flujo.....)	21
4.2.1. Lista de actores.....	21
4.2.2. Caso de uso Usuarios de la aplicación	23
4.2.3. Caso de uso Registro de usuario	24
4.2.4. Caso de uso Crear almacén	25
4.2.5. Caso de uso Editar almacén	26
4.2.6. Caso de uso para agregar alimento.....	27
4.2.7. Caso de uso para Editar alimento	28
4.2.8. Caso de uso para Acceder al carrito de la compra	30
4.2.9. Caso de uso para Gestión de alimentos para comprar	31
4.2.10. Caso de uso para Cambio de contraseña	32
4.2.11. Caso de uso para Eliminar cuenta	33
4.2.12. Caso de uso para Cerrar sesión	34
4.2.13. Diagrama de despliegue Login Usuarios	35
4.2.14. Diagrama de despliegue Registro de usuarios	36
4.3 Análisis y diseño de la interfaz de usuarios. Mockups.....	36
4.3.1. Mockup de la pantalla de carga.....	37
4.3.2. Mockup Login de usuario	38
4.3.3. Mockup registro de usuario	39
4.3.4. Inicio de la aplicación.....	40
4.3.5. Mockup Almacén	41
4.3.6. Mockup Carrito	42

4.3.7.	Mockup Ajustes de usuario.....	43
4.3.8.	Mockup profesionales	44
4.4	Diseño de la arquitectura de la aplicación	48
4.4.1.	Diagrama interno de la estructura de mi app.	49
4.4.2.	Tecnologías y herramientas usadas y descripción de las mismas	51
4.4.3.	Arquitectura de componentes de la aplicación	53
5	Documento de análisis y diseño	54
5.1	Pruebas.....	54
5.1.1.	Pruebas funcionales	54
5.2	Instalación y configuración.....	56
5.3	Manual de usuario	57
5.3.1.	Página de inicio	57
5.3.2.	Página de registro	58
5.3.3.	Página de inicio	59
5.3.4.	Página de Almacenes.....	60
5.3.5.	Página de carrito.....	62
5.3.6.	Página de Ajustes	64
6	Documento de cierre	65
6.1	Diario de bitácora	66
7	Webgrafía.....	67



INTRODUCCIÓN

Mi proyecto se basa en una aplicación que permite gestionar tu comida en casa de manera inteligente y sencilla. Con ella puedes:

- Crear almacenes virtuales para cada lugar donde guardas comida (nevera, despensa, congelador, etc.) y registrar lo que tienes, cuánto y dónde.
- Generar una cesta de compra que se actualiza automáticamente según lo que se te va acabando o lo que necesitas para tus recetas o dietas.
- Añadir recetas o dietas y ver si tienes los ingredientes necesarios, usando un código de colores: verde (sí), amarillo (poco) y rojo (no). También puedes añadir los ingredientes que te falten a la cesta de compra con un solo clic.
- Acceder a otras funciones útiles, como:
 - Crear tu cuenta de usuario y acceder desde cualquier dispositivo.
 - Recibir recordatorios de los alimentos que se van a caducar pronto para evitar tirar comida.
 - Consultar el historial de consumo de los productos y saber cuánto te duran.
 - Integrar tus listas de compra anteriores para no repetir productos y optimizar tu tiempo.
 - Ver estadísticas de tu consumo alimentario y planificar mejor tus compras.
 - Recibir notificaciones cuando se te esté acabando algún alimento y debas añadirlo a la cesta de compra.
 - Compartir tus recetas con tus amigos y con otras personas.



1. Documento de propuesta de proyecto

1.1. Título del proyecto

Smart Pantry tu almacén virtual

1.2. Título corto del proyecto

Smart Pantry

1.3. Contexto breve

Smart Pantry es una aplicación móvil que pretende facilitar algo tan cotidiano como la lista de la compra, permitiendo ahorrar puesto que te permite ver de manera sencilla los productos de los que ya dispones, permitiendo así evitar un gasto adicional al comprar duplicados.

1.4. Objetivo breve

Llegar a desarrollar las funciones mínimas para que la aplicación móvil sea válida.

2. Documento descripción del proyecto

2.1. Contexto del proyecto

Este proyecto nace de problemas que hasta ahora yo he tenido y he querido solucionar, al ver como gente que me rodea también lo llega a sufrir se me ocurrió una manera de facilitar y ayudar al resto de personas, creando así una aplicación sencilla que permite ayudar de gran manera al usuario.

2.1.1. Ámbito y entorno

La finalidad de este proyecto es conseguir una meta propia, incluso una vez entregado mi intención es seguir desarrollando este proyecto, puesto que se que puede llegar a ser de gran utilidad. Quiero realizar una aplicación sencilla pero útil, con la cual cualquier usuario que tenga la más mínima iniciativa sea capaz de usarla de manera intuitiva.

2.1.2. Destinatarios

Como ya he mencionado anteriormente, esta aplicación va dirigida a todo aquel usuario que tenga la iniciativa de ahorrar ya se en productos alimenticios o de manera económica.

2.2. Objetivo del proyecto

El objetivo de este proyecto es desarrollar una aplicación móvil que permita ayudar a todo aquel usuario que lo necesite, ya sea porque quiere ahorrar, comer mejor, facilitarse la lista de la compra o simplemente porque quiere ver recetas.

Mi intención final es que cada usuario independientemente de la razón por la que la use se sienta familiarizado y seguro con mi aplicación de manera que le sea fácil y útil usarla en su vida cotidiana.



3 Documento de Acuerdo del proyecto

3.1 . Requisitos funcionales y no funcionales / Historias de usuario

En este punto detallaré los requisitos funcionales y no funcionales que tendrá mi aplicación móvil “Smart Pantry”.

Para el desarrollo de este proyecto voy a clasificar los requisitos de la siguiente manera:

- 3.1.1. Los **requisitos funcionales** serán aquellos que determinen el comportamiento de la aplicación.
- 3.1.2. Los **requisitos no funcionales** serán aquellos que definan las características de la aplicación, ya se la usabilidad, interfaz, etc.

Usaré el siguiente formato de tabla.

Requisito: RF-X “Nombre”			
Descripción	“breve descripción”		
Tipo	“tipo de requisito”	Prioridad	“prioridad: alta, media o baja”
Versión Alta	10/04/2022	Fecha Alta	10/05/2022

3.1.1. Requisitos funcionales

Requisito: RF-1 Registro de usuarios			
Descripción	Permitir a los usuarios registrarse de manera sencilla Permite registrarse usando una cuenta de Google lo que facilita las cosas, en caso de no disponer de una o no querer usarla se puede usar cualquier otra cuenta de correo. Para registrarse solo se necesita un correo y una contraseña, esta última se deberá introducir dos veces para mayor seguridad		
Tipo	Funcional	Prioridad	alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023



Requisito: RF-2 Iniciar sesión

Descripción	Permitir iniciar sesión, en caso de ya haber registrado un usuario este podrá iniciar sesión mediante las credenciales introducidas en el registro. también se puede iniciar sesión con Google		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-3. Cerrar la sesión

Descripción	Una vez dentro de la aplicación el usuario podrá cerrar sesión mediante un botón		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-4 Eliminar cuenta

Descripción	El usuario podrá eliminar su cuenta, eliminando así todos sus datos.		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RFN-5 Cabecera

Descripción	El color predominante es el naranja pues este estimula los sentidos, por ello la cabecera será de este color.		
Tipo	No Funcional	Prioridad	Baja
Versión Alta	18/12/2023	Fecha Alta	18/12/2023



Requisito: RF-6 Menu inferior

Descripción	Este menú es el más común en las apps, desde aquí se podrá acceder a las principales pantallas, en este caso son tres, pero en un futuro agregare mas. <ul style="list-style-type: none">• Inicio• Menú usuario• Cerrar sesión		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-7 Añadir almacén

Descripción	En la pantalla principal consta de 2 botones. El principal que permite añadir nuevos almacenes, este tiene una pequeña animación justo debajo la cual en caso de que no haya ningún almacén en esa cuenta, esta se activa, dando así una pequeña señalización de lo que debe hacer el usuario. Al darle al botón este debe mostrar un popup en el cual se pida el nombre del almacén, al escribir este y darle a aceptar se añadirá un nuevo almacén.		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023



Requisito: RF-8 Listar almacenes

Descripción	<p>Al acceder a la pantalla principal, se deben listar todos los almacenes de los que dispone ese usuario, estos han debido ser creados previamente.</p> <p>En caso de que el usuario no tenga ningún almacén se activará la animación mencionada anteriormente.</p> <p>Al hacer click en cualquier almacén, se deberá abrir una segunda pantalla donde aparecerá los siguiente</p> <ul style="list-style-type: none">• Nombre del almacen• Un icono en forma de lápiz con los ajustes del almacén (estos se mencionan posteriormente)• Un buscador• Un botón para añadir los alimentos escritos en el buscador• La lista de los alimentos que dispone el almacén		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-9 Ajustes almacén

Descripción	<p>Anteriormente mencioné un icono dentro del almacén con forma de lápiz, Al darle click a este se debe abrir un menú despegable el cual permita elegir entre dos opciones.</p> <ul style="list-style-type: none">• Eliminar, esto mostrará un despegable para asegurar que queremos eliminar el almacén, al aceptar este será eliminado.• Editar, esto permite cambiar el nombre del almacén		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023



Requisito: RF-10 Buscador de alimentos

Descripción	Dentro de cada almacén y del carrito hay un buscador que funciona de la misma manera. Este tiene cargado de la base de datos todos los alimentos disponibles, al empezar a escribir este irá mostrando las opciones disponibles según lo que escribamos, aun así este permitir escribir cualquier cosa, permitiendo así añadir alimentos que no estén en la base de datos		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-11 Añadir alimento

Descripción	Una vez escrito el alimento que se requiera en el buscador, al darle al botón aparecerá un despegable para introducir la cantidad de cada alimento. A la hora de añadir el alimento, en caso de existir en la base de datos este se guardará junto a su categoría, es decir, si el usuario guarda "Pan" este se añadirá con el categoría "Cereal". En caso de que el alimento no este en la base de datos el categoría será "Personalizado"		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023



Requisito: RF-12 Ajustes alimento

Descripción	Una vez añadido el alimento, este tiene 3 opciones distintas <ul style="list-style-type: none">• Eliminar, permite quitar el alimento del almacén• Editar, permite cambiar la cantidad de ese alimento• Añadir al carrito, permite añadir el alimento con la cantidad indicada al carrito de la compra		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-13 Carrito de la compra

Descripción	En el mismo menú principal donde está ubicado el botón para crear los almacenes hay otro para mostrar la lista de la compra.		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-14 Lista de la compra

Descripción	Por ahora cada usuario solo dispone de una única lista de la compra, esta se crea automáticamente. En ella se añadirán los alimentos que anteriormente se hayan elegido desde cada almacén, aun así cada usuario también puede buscarlos y añadirlos manualmente <ul style="list-style-type: none">• La primera que es la sección en donde aparecen los artículos para comprar• La segunda, en donde salen los artículos ya comprados		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023



Requisito: RF-15 Ajustes de usuario

Descripción	Dentro de los ajustes de usuario he creado diferentes tipos, en la primera versión funcional solo funcionan tres, estos serían <ul style="list-style-type: none">• Eliminar cuenta• Cambiar contraseña, que envía un correo para cambiarla• Cerrar sesión		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-16 Comprar

Descripción	Dentro del carrito, al hacer click en cualquier artículo que este en la primera sección, pasará a estar comprado, en caso de darle a uno que ya este comprado, volverá a estar en la lista para comprar		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023

Requisito: RF-17 Base de datos

Descripción	Todo lo que el usuario hace, se guarda en una base de datos relacionada a su correo.		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023



Requisito: RFN 18 Requisitos LOPD

Descripción	La aplicación ha de ser intuitiva y familiar para que le sea fácil de usar al usuario.		
Tipo	Funcional	Prioridad	Alta
Versión Alta	18/12/2023	Fecha Alta	18/12/2023



3.2 Tareas

Ahora que se han identificado todos los requisitos se va a definir las tareas de desarrollo del proyecto

Código tarea	Descripción	Duración (horas)
T1	Captura de requisitos	2
T1.1	Identificación de requisitos de los Usuarios.	2
T2	Realización de casos de uso	3
T3	Diagrama de secuencia iniciar sesión	1
T3.1	Diagrama de secuencia enviar mensaje	1
T3.2	Diagrama de secuencia añadir amigo	1
T3.3	Diagrama de secuencia registrase	1
T4	Diseño de la base de datos	3
T4.1	Diagrama Entidad Interrelación	1
T4.2	Esquema relacional estándar	1
T4.3	Normalización	2
T4.4	Insertar datos en las tablas	2
T6	Estudio de las tecnologías a utilizar.	0.50
T7	Creación de las interfaces gráficas	3
T8	Realizar pruebas y corregir errores.	4
T8.1	Realizar pruebas unitarias.	3
T8.2	Realizar pruebas del sistema completo.	4
T8.3	Realizar informe de pruebas y resultados.	4
T9	Realizar documentación. Informe de Instalación.	5
Total horas		43.50

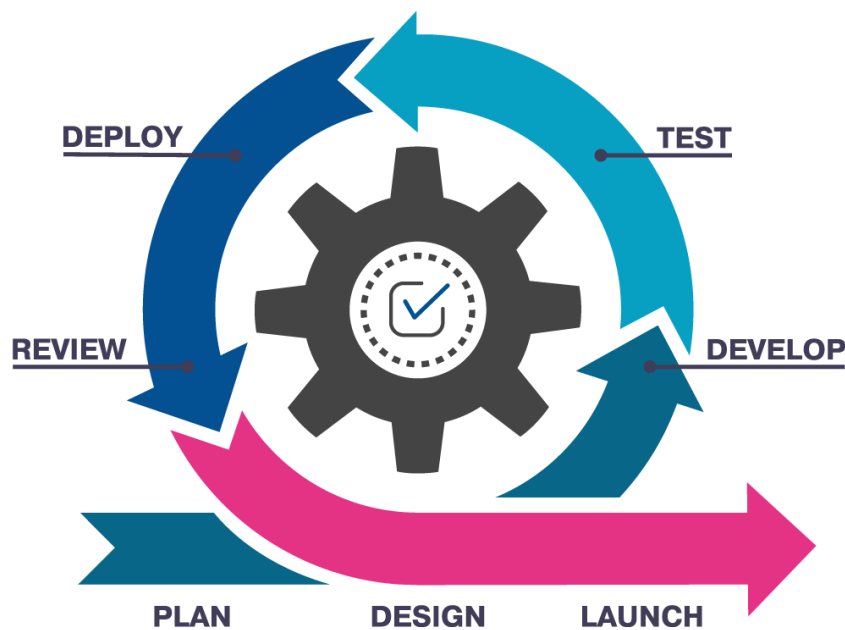


3.3 Metodología a seguir para la realización del proyecto

Hasta ahora en todos los proyectos que he desarrollado he usado la misma metodología.

Metodología agile

Se pretende aplicar una metodología la cual los usuarios (en este caso amigos y yo mismo) siempre tengan una implicación en el proyecto, permitiendo así comprobar y validar los requisitos y el diseño antes de comenzar una nueva fase.



Lo primero y mas necesario será el diseño del registro de usuario, una vez completo, se podrá seguir con las siguiente funcionalidades, como agregar almacenes.



3.4 Presupuesto (gastos, ingresos, beneficios)

He creado un plan económico donde se detallan los gastos y beneficios esperados

Elemento	Precio sin IVA (€)	IVA (21%) (€)	Precio con IVA (€)
Diseño de la interfaz de usuario	2000	420	2420
Diseño de iconos y gráficos	500	105	605
Desarrollo de la aplicación para Android	4000	840	4840
Desarrollo de la aplicación para iOS	4000	840	4840
Pruebas de funcionalidad	1000	210	1210
Pruebas de rendimiento	1000	210	1210
Lanzamiento en Google Play Store	500	105	605
Lanzamiento en Apple App Store	500	105	605
Mantenimiento y actualizaciones durante un año	2000	420	2420
Silla de oficina	150	31.5	181.5
Ordenador	1000	210	1210
Teclado	50	10.5	60.5
Ratón	25	5.25	30.25
Mesa de oficina	200	42	242

Considerando que es todo en 12 meses los totales serian los siguientes

Total sin IVA: 33.925€

IVA (21%): 7.124,25€

Total con IVA: 41.049,25€



3.4.1. Presupuesto

Realizando una estimación el presupuesto final sería de 42.000€

Dentro de este entran distintas cosas:

- El plazo de entrega se estima en un plazo aproximado de 12 meses.
- Se aplicará el IVA vigente en el momento del pago.
- Se realizará un diseño que se adapte a las especificaciones de los usuarios.

3.4.2. Beneficios

He calculado los siguientes beneficios.

- Usuarios activos por día (DAU): 10.000
- Impresiones de anuncios por usuario por día: 5
- Tasa de clics (CTR): 1%
- Costo por clic (CPC): 0,10€

Entonces, los ingresos diarios por publicidad serían:

- Ingresos diarios=DAU×Impresiones por usuario×CTR×CPC

Sustituyendo los valores:

Ingresos diarios=10.000×5×0.01×0.10=50€

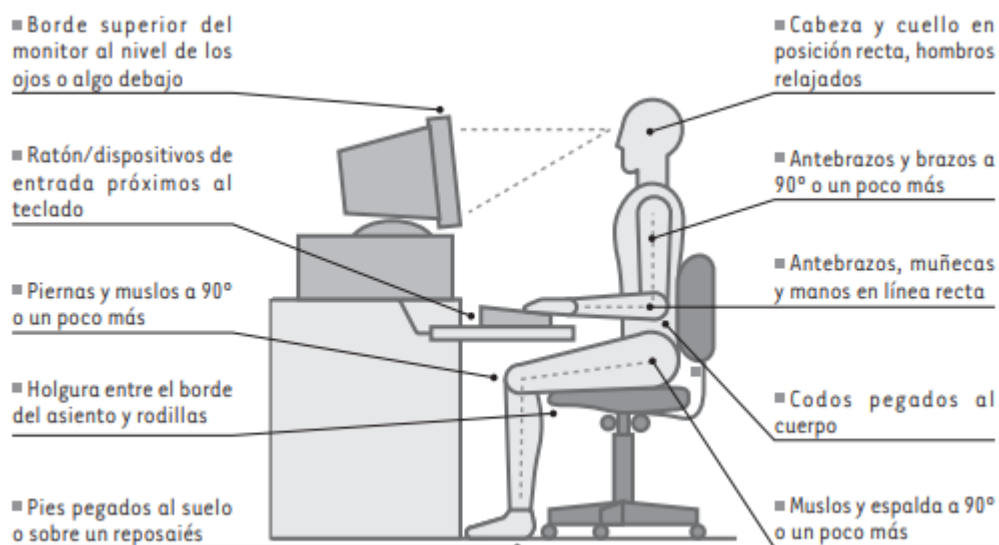
Por lo tanto, mis ingresos mensuales serían de aproximadamente 1500€ (considerando un mes de 30 días).

Con estos beneficios, considerando que se mantienen constantes, tardaría alrededor de 27 meses en recuperar mi inversión.

3.5 Prevención de riesgos

Puesto que la gran mayor parte que dedicaré a la realización de mi proyecto será desde casa, necesitare tener diferentes cuidados.

Adjunto una imagen que los enumera.





4 Documento de análisis y diseño

4.1 **Modelo de datos. Análisis y diseño de base de datos**

Para la implementación de mi aplicación, he optado por utilizar Firebase, una base de datos ofrecida por Google. He seleccionado Firebase por varias razones clave que la hacen ideal para este proyecto:

Escalabilidad: Firebase es una base de datos NoSQL, lo que significa que puede manejar grandes cantidades de datos con alta velocidad y eficiencia. Esto es especialmente útil para una aplicación como la mía, que puede tener un gran número de usuarios y una gran cantidad de datos.

Tiempo real: Firebase ofrece actualizaciones en tiempo real, lo que significa que los cambios en los datos se reflejan inmediatamente en todos los dispositivos conectados. Esto es crucial para mantener a todos los usuarios de la aplicación sincronizados con la información más reciente.

Autenticación: Firebase proporciona un sistema de autenticación integrado que soporta correo electrónico y contraseña, teléfono, y proveedores de identidad federados como Google, Facebook y Twitter. Esto facilita la gestión de cuentas de usuario y mejora la seguridad de la aplicación.

Almacenamiento en la nube: Firebase ofrece un servicio de almacenamiento en la nube, lo que permite almacenar y compartir imágenes, audio y video de manera segura y eficiente.

Análisis de uso: Con Firebase, puedo tener acceso a análisis detallados sobre cómo los usuarios interactúan con la aplicación. Esto me permite entender mejor a los usuarios y mejorar la aplicación en función de sus necesidades y comportamientos.

4.1.1. **Diagrama E/R**

El modelo Entidad/Relación es una herramienta conceptual que se utiliza para representar los datos en un sistema y cómo estos datos se relacionan entre sí. Este modelo fue propuesto por Peter Chen en la década de 1970 y sigue siendo ampliamente utilizado hoy en día debido a su simplicidad y eficacia.

En el modelo Entidad/Relación, los datos se representan como entidades, relaciones y atributos:

Entidades: Las entidades son objetos o conceptos sobre los que se recoge información. Se representan mediante un cuadrado en los diagramas entidad-relación. Hay dos tipos de entidades:

Entidades fuertes: Son entidades que existen por méritos propios. No dependen de ninguna otra entidad para su existencia.

Entidades débiles: Son entidades cuya existencia depende de otra entidad fuerte. Se representan con un cuadrado doble en los diagramas entidad-relación.

Relaciones: Las relaciones representan las asociaciones entre dos o más entidades. Cada relación tiene un nombre que expresa la finalidad de la relación. Se representan mediante rombos en los diagramas entidad-relación. Las relaciones se clasifican según su grado, que es el número de entidades que participan en la relación:

Relaciones binarias: Involucran a dos entidades.

Relaciones ternarias: Involucran a tres entidades.



Relaciones unarias o reflexivas: Involucran a una entidad consigo misma.

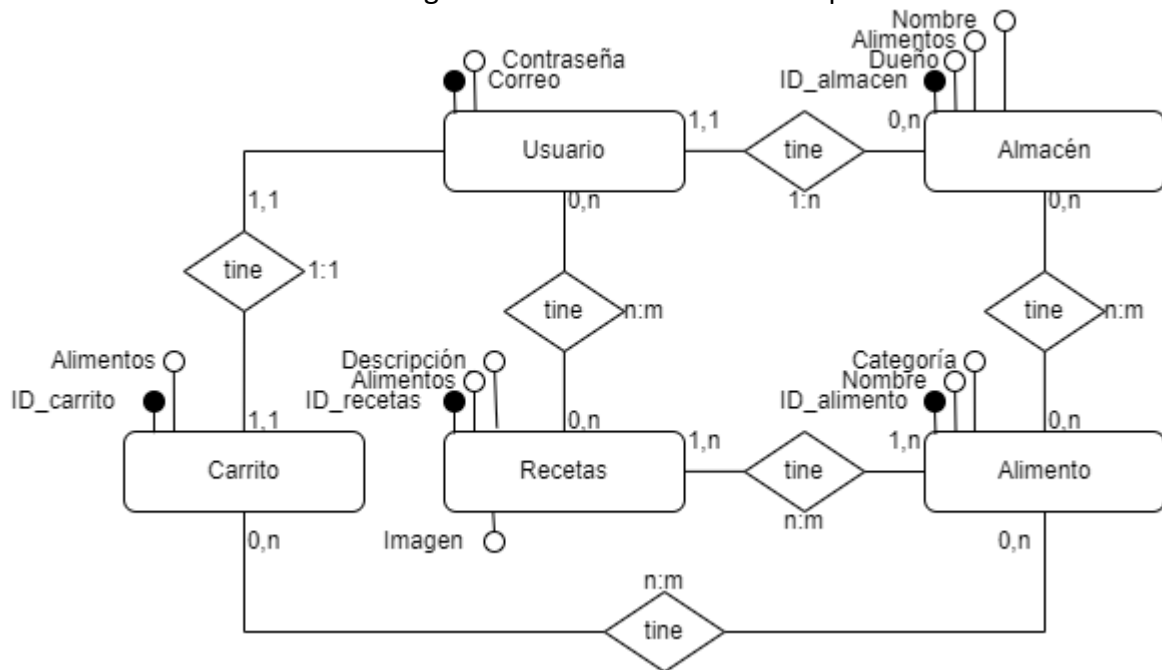
Relaciones n-arias: Involucran a n entidades.

Atributos: Los atributos son las características o propiedades que definen a una entidad o relación. Hay varios tipos de atributos:

Atributo clave: Un atributo clave es un atributo que identifica de manera única a cada instancia de una entidad o relación. En los diagramas entidad-relación, los atributos clave se subrayan.

Atributo de relación: Un atributo de relación es un atributo que es propio de una relación y que interviene en la relación de dos o más entidades.

A continuación muestro el diagrama entidad relación de mi aplicación web



4.1.2. Esquema relacional estándar

Usuarios (Correo, contraseña)

CP: correo

Almacén (ID_almacen, dueño, alimentos, nombre)

CP(ID_Almacen)

CF (Dueño) referencia usuario

Carrito (ID_carrito, alimetros)

CP: ID_carrito

CF (ID_carrito) referencia a usuario

Recetas (ID_recetas, alimentos, descripcion)

CP(ID_recetas)

CF (ID_recetas) referencia usuario



Alimento (ID_alimento, categoria, nombre)

CP(ID_alimento)

CF (ID_alimento) referencia usuario

4.1.3. Normalización

El proceso de creación de una base de datos es un procedimiento detallado que suele finalizar con la conversión del modelo entidad-relación a un modelo relacional. Sin embargo, es crucial recordar que al diseñar cualquier sistema, ya sea una base de datos o cualquier otra solución de TI, es vital evaluar su rendimiento. Si el sistema no satisface ciertos estándares de calidad, es necesario realizar mejoras iterativas en el diseño hasta lograr el nivel de calidad requerido.

Uno de los indicadores que se utilizan para evaluar la calidad de una base de datos es la forma normal en la que se encuentra su diseño. Este proceso de asegurar que los atributos de un diseño cumplan con ciertas formas normales se denomina normalización.

Primera Forma Normal (1FN) es el primer nivel de la normalización. En esta forma normal, se prohíbe que un atributo en una tabla pueda tener múltiples valores. Esta restricción es inherente al modelo relacional, ya que las tablas gestionadas por un Sistema de Gestión de Bases de Datos (SGBD) relacional están construidas de esta manera. Además, no se permiten duplicados en esta forma normal.

Segunda Forma Normal (2FN) es el siguiente nivel de normalización. Un diseño se encuentra en 2FN si ya está en 1FN y, además, cada atributo que no forma parte de la clave principal tiene una dependencia completa de la clave principal.

Tercera Forma Normal (3FN) es el tercer nivel de normalización. Un diseño se encuentra en 3FN si ya está en 2FN y, además, no hay ningún atributo no clave que dependa de forma transitiva de la clave principal.

4.1.4. Dominios de atributos

Tabla Usuarios				
Nombre Atributo	Tipo Dato	PRIMARY_KEY	FOREIGN_KEY	NULL
Email	String	*		
Contraseña	String			
fechaInicio	date			

Tabla Alimentos				
Nombre Atributo	Tipo Dato	PRIMARY_KEY	FOREIGN_KEY	NULL
ID_alimetno	String	*		
Nombre	String			
Categoria	String			
Cantidad	Int			?



Tabla Almacenes				
Nombre Atributo	Tipo Dato	PRIMARY_KEY	FOREIGN_KEY	NULL
ID_almacen	String	*		
Nombre	String			
Dueño	String		Usuario	
Alimentos	Array<Alimento>		Alimentos	¿

Tabla Carrito				
Nombre Atributo	Tipo Dato	PRIMARY_KEY	FOREIGN_KEY	NULL
ID_almacen	String	*	Usuarios	
Alimentos	Array<Alimento>		Alimentos	¿

Tabla Receta				
Nombre Atributo	Tipo Dato	PRIMARY_KEY	FOREIGN_KEY	NULL
ID_receta	String	*		
Nombre	String			
Dueño	String		Usuario	¿
Alimentos	Array<Alimento>		Alimentos	

4.2 Análisis y diseño del sistema funcional (Diagramas que procedan como el de casos de uso, de clases, de secuencia, de actividad, de estados, de flujo...)

En este punto el objetivo es obtener los elementos que han sido definidos en la etapa de análisis, para así obtener los diagramas de casos de uso, de secuencia, actividad

4.2.1. Lista de actores

Actor: ACT-001 Usuario	
Descripción	Usuario registrado en la aplicación



Ahora mostraré los casos de uso, diagramas de actividad y de secuencia realizados para el desarrollo de la aplicación.

Caso de Uso	Descripción
CU-001	Identificación de usuarios
CU-002	Registro de usuarios
CU-003	Añadir almacén
CU-004	Añadir alimento al almacén
CU-005	Editar almacén
CU-006	Editar alimento del almacén
CU-007	Eliminar alimento del almacén
CU-008	Añadir alimento del almacén al carrito
CU-009	Entrar al carrito
CU-010	Añadir alimento al carrito
CU-011	Eliminar alimento del carrito
CU-012	Comprar alimento
CU-013	Eliminar alimento
CU-014	Cambiar contraseña
CU-015	Eliminar cuenta
CU-016	Cerrar sesión

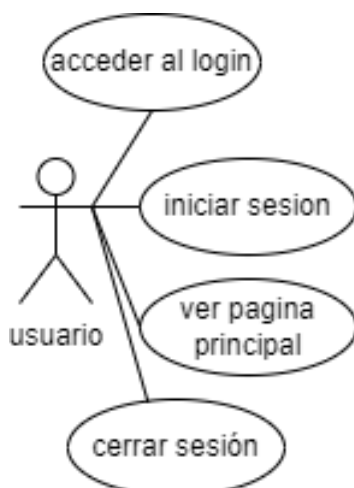


4.2.2. Caso de uso Usuarios de la aplicación

Caso de Uso 001 Identificación de usuarios	
Descripción	Un usuario registrado puede acceder a la página
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">El sistema muestra la pantalla de inicio con las opciones "Iniciar sesión" y "Registrarse".El usuario selecciona la opción "Iniciar sesión".El sistema muestra el formulario de inicio de sesión solicitando correo electrónico y contraseña.El usuario introduce su correo electrónico y contraseña.5. El usuario selecciona la opción "Iniciar sesión" para enviar el formulario.El sistema valida las credenciales proporcionadas por el usuario.Si las credenciales son válidas, el sistema inicia la sesión del usuario y lo redirige a la pantalla principal de la aplicación.Fin del caso de uso.
Secuencia(s) Alternativas(s)	El usuario no está registrado y el sistema muestra una alerta
Precondiciones	<ul style="list-style-type: none">El usuario ha descargado la aplicación y la ha abierto.El usuario ya tiene una cuenta registrada.
Postcondiciones	<ul style="list-style-type: none">El usuario ha iniciado sesión en la aplicación.El usuario puede acceder a las funcionalidades de la aplicación.

El usuario puede acceder al login

El usuario puede iniciar sesión





4.2.3. Caso de uso Registro de usuario

Caso de Uso 002 Registro de usuario	
Descripción	Un nuevo usuario necesita registrarse para poder acceder
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">▪ El sistema muestra la pantalla de inicio con las opciones "Iniciar sesión" y "Registrarse".▪ El usuario selecciona la opción "Registrarse".▪ El sistema muestra el formulario de registro solicitando detalles como nombre de usuario, correo electrónico y contraseña.▪ El usuario introduce su nombre de usuario, correo electrónico y contraseña.▪ El usuario selecciona la opción "Registrarse" para enviar el formulario.▪ El sistema valida la información proporcionada por el usuario.▪ Si la información es válida, el sistema crea una nueva cuenta para el usuario y muestra un mensaje de éxito.▪ El sistema redirige al usuario a la pantalla de inicio de sesión.▪ Fin del caso de uso.
Secuencia(s) Alternativas(s)	Si la información proporcionada por el usuario no es válida (por ejemplo, el correo electrónico ya está en uso), el sistema muestra un mensaje de error y solicita al usuario que introduzca la información de nuevo.
Postcondiciones	<ul style="list-style-type: none">▪ Se ha creado una nueva cuenta para el usuario.▪ El usuario puede iniciar sesión en la aplicación con sus credenciales.
Precondiciones	<ul style="list-style-type: none">▪ El usuario ha descargado la aplicación y la ha abierto.▪ El usuario no tiene una cuenta existente.

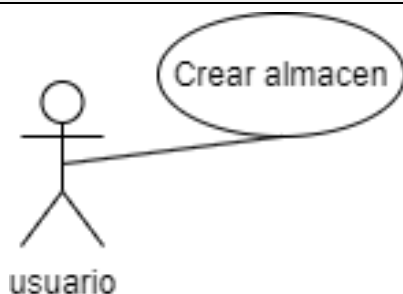


El usuario se puede registrar

En caso de no registrarse el usuario no puede acceder

**4.2.4. Caso de uso Crear almacén**

Caso de Uso 003 Crear almacenes	
Descripción	Cuando un usuario accede puede crear almacenes
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">▪ El sistema muestra la pantalla principal de la aplicación con varias opciones.▪ El usuario selecciona la opción "Crear almacenamiento".▪ El sistema muestra un formulario para crear un nuevo almacenamiento, solicitando detalles como el nombre del almacenamiento.▪ El usuario introduce el nombre del almacenamiento y selecciona la opción "Crear".▪ El sistema valida el nombre proporcionado por el usuario.▪ Si el nombre es válido y no está en uso, el sistema crea un nuevo almacenamiento con ese nombre y muestra un mensaje de éxito.▪ El sistema redirige al usuario a la pantalla del nuevo almacenamiento.▪ Fin del caso de uso.
Secuencia(s) Alternativas(s)	Salta un error
Precondiciones	El usuario debe estar registrado El usuario ha iniciado sesión
Postcondiciones	<ul style="list-style-type: none">▪ Se ha creado un nuevo almacenamiento con el nombre proporcionado por el usuario.▪ El usuario puede acceder y gestionar el nuevo almacenamiento.



El usuario puede crear tantos almacenes como quiera

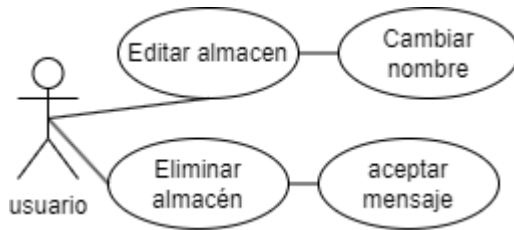


4.2.5. Caso de uso Editar almacén

Caso de Uso 004 Editar almacén	
Descripción	El usuario puede editar el nombre o eliminar el almacén accediendo a este
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">▪ El sistema muestra la pantalla principal de la aplicación con varias opciones.▪ El usuario selecciona el almacenamiento que desea editar.▪ El sistema muestra la pantalla del almacenamiento con las opciones "Cambiar nombre" y "Eliminar".▪ El usuario selecciona la opción "Cambiar nombre".▪ El sistema muestra un formulario para cambiar el nombre del almacenamiento.▪ El usuario introduce el nuevo nombre y selecciona la opción "Guardar".▪ El sistema valida el nuevo nombre proporcionado por el usuario.▪ Si el nuevo nombre es válido y no está en uso, el sistema cambia el nombre del almacenamiento y muestra un mensaje de éxito.▪ El usuario selecciona la opción "Eliminar".▪ El sistema muestra un mensaje de confirmación para asegurarse de que el usuario desea eliminar el almacenamiento.▪ El usuario confirma la eliminación.▪ El sistema elimina el almacenamiento y muestra un mensaje de éxito.▪ Fin del caso de uso.
Secuencia(s) Alternativas(s)	<ul style="list-style-type: none">▪ A. Si el nuevo nombre proporcionado por el usuario no es válido o ya está en uso, el sistema muestra un mensaje de error y solicita al usuario que introduzca un nombre diferente.▪ B. Si el usuario cancela la eliminación del almacenamiento, el sistema cierra el mensaje de confirmación y no realiza ninguna acción.
Precondiciones	<ul style="list-style-type: none">▪ - El usuario ha iniciado sesión en la aplicación.▪ - El usuario tiene los permisos necesarios para editar el almacenamiento.

**Requisitos relacionados**

- El nombre del almacenamiento ha sido cambiado.
- El almacenamiento ha sido eliminado.



En caso de elegir editar, este podrá cambiar el nombre

En caso de elegir eliminar, saltará una pestaña que deberá aceptar por seguridad.

4.2.6. Caso de uso para agregar alimento

Caso de Uso 005 Agregar alimento	
Descripción	Este sirve para agregar alimentos tanto dentro de cada almacén como en el carrito de la compra
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">▪ El sistema muestra la pantalla del almacenamiento seleccionado por el usuario.▪ El usuario selecciona la opción "Añadir alimento".▪ El sistema muestra un formulario para añadir un nuevo alimento, solicitando detalles como el nombre del alimento.▪ El usuario introduce el nombre del alimento y selecciona la opción "Añadir".▪ El sistema valida el nombre del alimento proporcionado por el usuario.▪ Si el nombre del alimento es válido, el sistema muestra un formulario para introducir la cantidad del alimento.▪ El usuario introduce la cantidad del alimento y selecciona la opción "Guardar".▪ El sistema valida la cantidad proporcionada por el usuario.▪ Si la cantidad es válida, el sistema añade el alimento con la cantidad especificada al almacenamiento y muestra un mensaje de éxito.▪ Fin del caso de uso.



Secuencia(s) Alternativas(s)	<ul style="list-style-type: none">▪ A. Si el nombre del alimento proporcionado por el usuario no es válido, el sistema muestra un mensaje de error y solicita al usuario que introduzca un nombre diferente.▪ B. Si la cantidad proporcionada por el usuario no es válida, el sistema muestra un mensaje de error y solicita al usuario que introduzca una cantidad válida.
Precondiciones	<ul style="list-style-type: none">▪ El usuario ha iniciado sesión en la aplicación.▪ El usuario tiene los permisos necesarios para añadir alimentos al almacenamiento.
Postcondiciones	<ul style="list-style-type: none">▪ Se ha añadido un nuevo alimento con la cantidad especificada al almacenamiento.▪ El usuario puede ver y gestionar el nuevo alimento en el almacenamiento.



El usuario dispone de la opción para agregar nuevos alimentos

4.2.7. Caso de uso para Editar alimento

Caso de Uso 006 Editar alimento	
Descripción	Este sirve para editar cada alimento por separado con tres diferentes opciones
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">▪ 1. El sistema muestra la pantalla del almacenamiento seleccionado por el usuario.▪ 2. El usuario selecciona el alimento que desea editar.▪ 3. El sistema muestra la pantalla del alimento con las opciones "Cambiar cantidad", "Eliminar" y "Añadir al carrito".▪ 4. El usuario selecciona la opción "Cambiar cantidad".▪ 5. El sistema muestra un formulario para cambiar la cantidad del alimento.▪ 6. El usuario introduce la nueva cantidad y selecciona la opción "Guardar".▪ 7. El sistema valida la nueva cantidad proporcionada por el usuario.▪ 8. Si la nueva cantidad es válida, el sistema cambia la cantidad del alimento en el almacenamiento y muestra un mensaje de éxito.



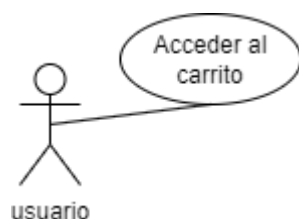
	<ul style="list-style-type: none"> 9. El usuario selecciona la opción "Eliminar". 10. El sistema muestra un mensaje de confirmación para asegurarse de que el usuario desea eliminar el alimento del almacenamiento. 11. El usuario confirma la eliminación. 12. El sistema elimina el alimento del almacenamiento y muestra un mensaje de éxito. 13. El usuario selecciona la opción "Añadir al carrito". 14. El sistema añade el alimento al carrito de la compra del usuario y muestra un mensaje de éxito. 15. Fin del caso de uso.
Secuencia(s) Alternativas(s)	<ul style="list-style-type: none"> A. Si la nueva cantidad proporcionada por el usuario no es válida, el sistema muestra un mensaje de error y solicita al usuario que introduzca una cantidad válida. B. Si el usuario cancela la eliminación del alimento, el sistema cierra el mensaje de confirmación y no realiza ninguna acción.
Precondiciones	<ul style="list-style-type: none"> - El usuario ha iniciado sesión en la aplicación. - El usuario tiene los permisos necesarios para editar alimentos en el almacenamiento.
Postcondiciones	<ul style="list-style-type: none"> - La cantidad del alimento en el almacenamiento ha sido cambiada. - El alimento ha sido eliminado del almacenamiento. - El alimento ha sido añadido al carrito de la compra del usuario.





4.2.8. Caso de uso para Acceder al carrito de la compra

Caso de Uso 007 Acceder al carrito de la compra	
Descripción	Este permite acceder al carrito personal del usuario
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">1. El sistema muestra la pantalla principal de la aplicación con varias opciones.2. El usuario selecciona la opción "Carrito de la compra".3. El sistema muestra la pantalla del carrito de la compra, que incluye una lista de los alimentos que contiene.4. El usuario puede ver los alimentos en el carrito, junto con sus cantidades y los precios correspondientes.5. El usuario puede seleccionar un alimento para ver más detalles o para eliminarlo del carrito.6. Fin del caso de uso.
Secuencia(s) Alternativas(s)	
Precondiciones	<ul style="list-style-type: none">- El usuario ha iniciado sesión en la aplicación.- El usuario tiene los permisos necesarios para acceder al carrito de la compra..
Postcondiciones	<ul style="list-style-type: none">- El usuario ha accedido al carrito de la compra.- El usuario puede ver y gestionar los alimentos en el carrito de la compra.



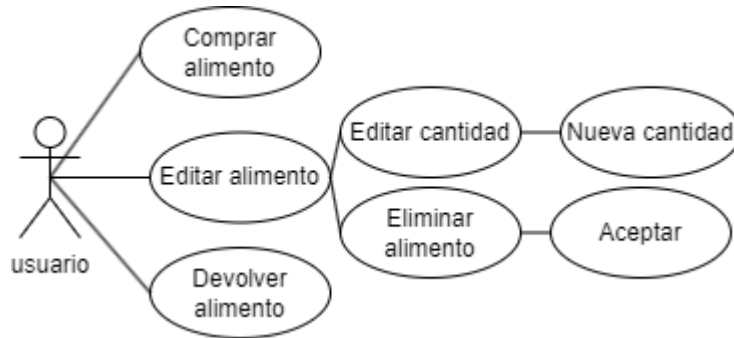


4.2.9. Caso de uso para Gestión de alimentos para comprar

Caso de Uso 008 Gestión de alimentos para comprar	
Descripción	Este sirve para editar cada alimento por separado con tres diferentes opciones
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none"> ▪ 1. El sistema muestra la pantalla del carrito de la compra, que incluye dos listas: "Alimentos para comprar" y "Alimentos comprados". ▪ 2. El usuario puede ver los alimentos en ambas listas, junto con sus cantidades y los precios correspondientes. ▪ 3. El usuario selecciona un alimento de la lista "Alimentos para comprar". ▪ 4. El sistema muestra las opciones "Comprar" y "Eliminar" para el alimento seleccionado. ▪ 5. El usuario selecciona la opción "Comprar". ▪ 6. El sistema mueve el alimento de la lista "Alimentos para comprar" a la lista "Alimentos comprados" y muestra un mensaje de éxito. ▪ 7. El usuario selecciona un alimento de la lista "Alimentos comprados". ▪ 8. El sistema muestra las opciones "Devolver a la lista de compra" y "Eliminar" para el alimento seleccionado. ▪ 9. El usuario selecciona la opción "Devolver a la lista de compra". ▪ 10. El sistema mueve el alimento de la lista "Alimentos comprados" a la lista "Alimentos para comprar" y muestra un mensaje de éxito. ▪ 11. Fin del caso de uso. ▪
Secuencia(s) Alternativas(s)	<ul style="list-style-type: none"> ▪ A. Si el usuario selecciona la opción "Eliminar", el sistema muestra un mensaje de confirmación para asegurarse de que el usuario desea eliminar el alimento de la lista. Si el usuario confirma la eliminación, el sistema elimina el alimento de la lista y muestra un mensaje de éxito. ▪
Precondiciones	<ul style="list-style-type: none"> ▪ - El usuario ha iniciado sesión en la aplicación. ▪ - El usuario tiene los permisos necesarios



	para gestionar el carrito de la compra.
Postcondiciones	<ul style="list-style-type: none"> Los alimentos se han movido correctamente entre las listas "Alimentos para comprar" y "Alimentos comprados". - El usuario puede ver y gestionar los alimentos en ambas listas.

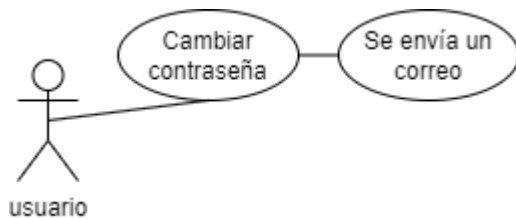


4.2.10. Caso de uso para Cambio de contraseña

Caso de Uso 009 Cambiar contraseña	
Descripción	Este permite el cambio de contraseña del usuario
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none"> 1. El sistema muestra la pantalla principal de la aplicación con varias opciones. 2. El usuario selecciona la opción "Ajustes de la cuenta". 3. El sistema muestra la pantalla de ajustes de la cuenta con la opción "Cambiar contraseña". 4. El usuario selecciona la opción "Cambiar contraseña". 5. El sistema solicita al usuario que introduzca su contraseña actual y la nueva contraseña. 6. El usuario introduce su contraseña actual y la nueva contraseña y selecciona la opción "Guardar". 7. El sistema valida las contraseñas proporcionadas por el usuario. 8. Si las contraseñas son válidas, el sistema cambia la contraseña del usuario, envía un correo electrónico de confirmación al usuario y muestra un mensaje de éxito. 9. Fin del caso de uso.
Secuencia(s) Alternativas(s)	<ul style="list-style-type: none"> A. Si las contraseñas proporcionadas por el usuario no son válidas, el sistema muestra un mensaje de error y solicita al usuario que

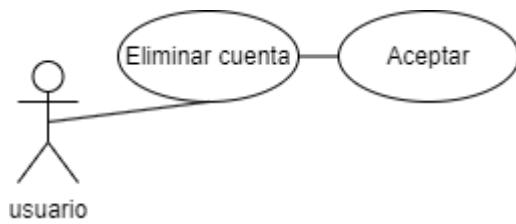


	introduzca las contraseñas de nuevo.
Precondiciones	<ul style="list-style-type: none">- El usuario ha iniciado sesión en la aplicación.- El usuario tiene los permisos necesarios para cambiar su contraseña.
Postcondiciones	<ul style="list-style-type: none">- La contraseña del usuario ha sido cambiada.



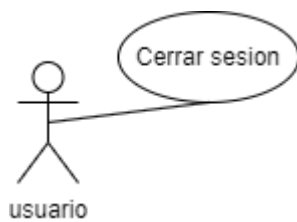
4.2.11. Caso de uso para Eliminar cuenta

Caso de Uso 010 Eliminar cuenta	
Descripción	Este permite la eliminación de la cuenta así como todos sus datos
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">1. El sistema muestra la pantalla principal de la aplicación con varias opciones.2. El usuario selecciona la opción "Ajustes de la cuenta".3. El sistema muestra la pantalla de ajustes de la cuenta con la opción "Eliminar cuenta".4. El usuario selecciona la opción "Eliminar cuenta".5. El sistema muestra un mensaje de confirmación para asegurarse de que el usuario desea eliminar su cuenta.6. El usuario confirma la eliminación.7. El sistema elimina la cuenta del usuario y muestra un mensaje de éxito.8. Fin del caso de uso.
Secuencia(s) Alternativas(s)	<ul style="list-style-type: none">A. Si el usuario cancela la eliminación de su cuenta, el sistema cierra el mensaje de confirmación y no realiza ninguna acción.
Precondiciones	<ul style="list-style-type: none">- El usuario ha iniciado sesión en la aplicación.- El usuario tiene los permisos necesarios para eliminar su cuenta.
Postcondiciones	<ul style="list-style-type: none">- La cuenta del usuario ha sido eliminada.



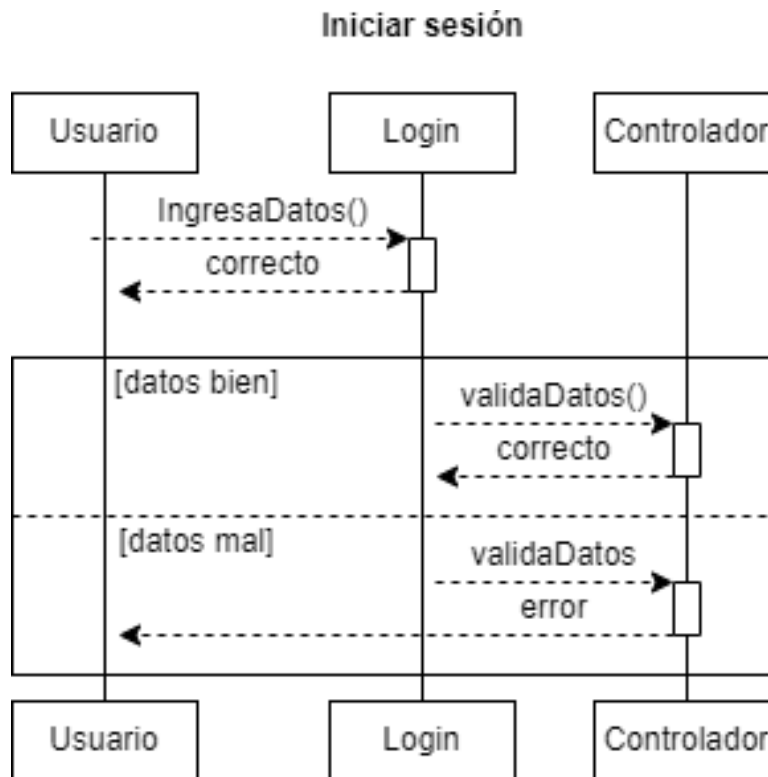
4.2.12. Caso de uso para Cerrar sesión

Caso de Uso 011 Cerrar sesión	
Descripción	El usuario puede cerrar su sesión en cualquier momento dentro de la aplicación
Actores	ACT-001 Usuario
Secuencia Normal	<ul style="list-style-type: none">1. El sistema muestra la pantalla principal de la aplicación con varias opciones.2. El usuario selecciona la opción "Cerrar sesión".3. El sistema cierra la sesión del usuario y muestra un mensaje de éxito.4. El sistema redirige al usuario a la pantalla de inicio de sesión.5. Fin del caso de uso.
Secuencia(s) Alternativas(s)	
Precondiciones	<ul style="list-style-type: none">El usuario ha iniciado sesión en la aplicación.
Postcondiciones	<ul style="list-style-type: none">La sesión del usuario ha sido cerrada.El usuario ha sido redirigido a la pantalla de inicio de sesión.





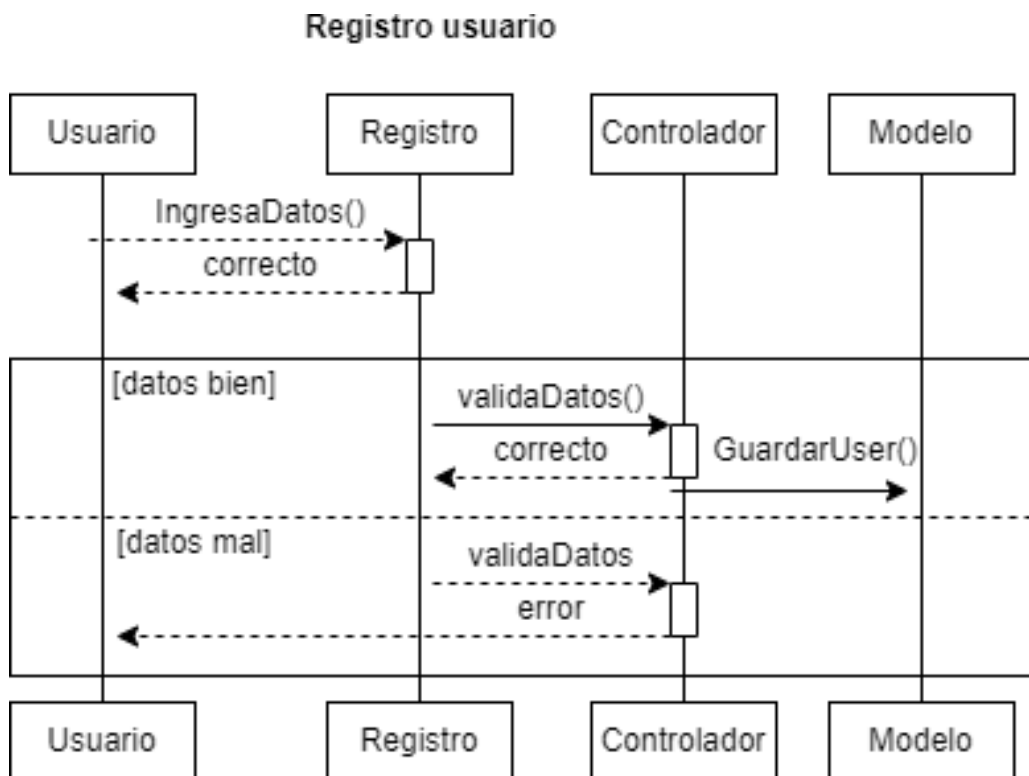
4.2.13. Diagrama de despliegue Login Usuarios



Aquí se muestra la interacción del usuario con la app al a hora de realizar el login, cuando el usuario introduce sus credenciales estas se validan, en caso de que sean correctas este podrá acceder, y en caso de que no lo sean mostrará un error



4.2.14. Diagrama de despliegue Registro de usuarios



Un nuevo usuario puede registrarse, cuando este introduce todas sus credenciales el programa se encarga de verificar que este correcto.

En caso de estar todo bien el usuario es guardado en la BD y se mostrará la pantalla principal.

En caso de que ocurra algún error este será mostrado.

4.3 Análisis y diseño de la interfaz de usuarios. Mockups

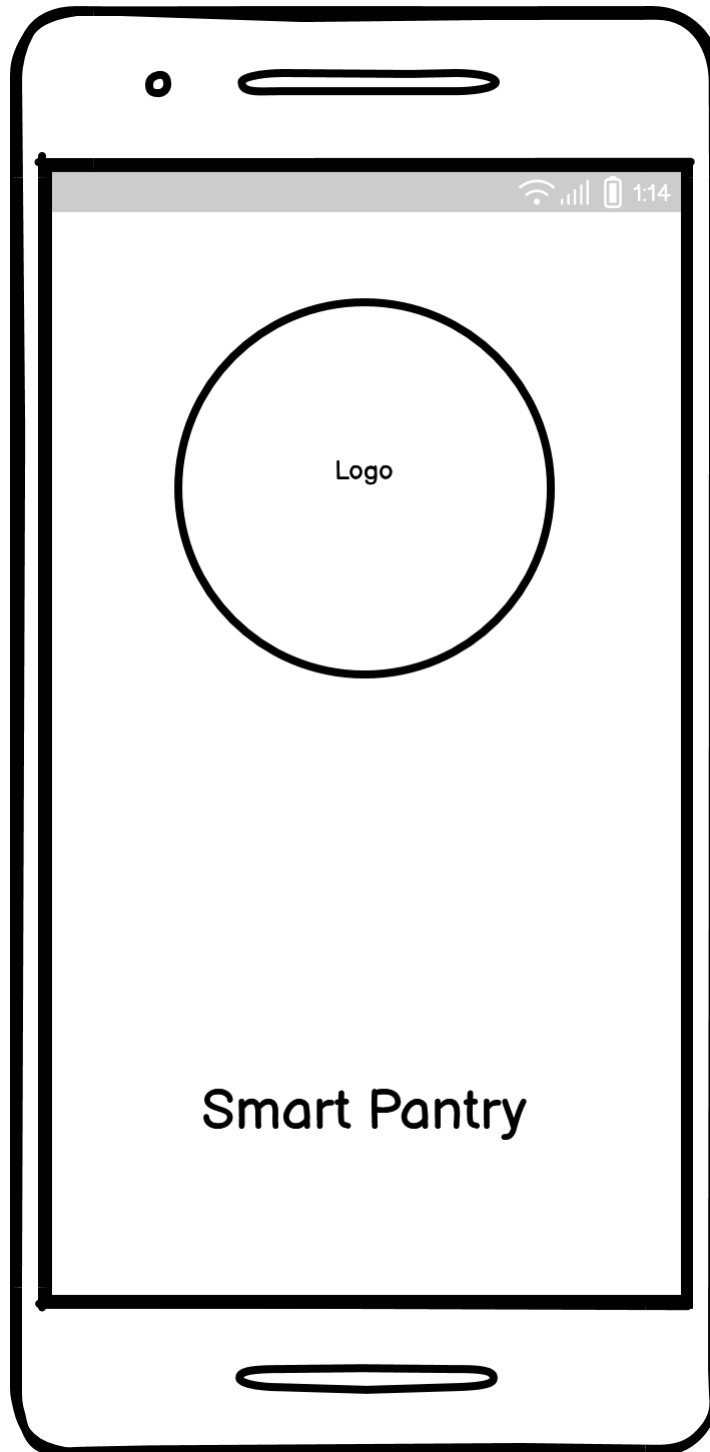
En este punto consta de dos partes.

En la primera muestro los primeros Mockups de los cuales sale el principal diseño de mi app

La segunda son unos Mockups mas detallados los cuales me servirán como inspiración para un futuro pues, como ya dije anteriormente, este proyecto se seguirá desarrollando a futuro.



4.3.1. Mockup de la pantalla de carga

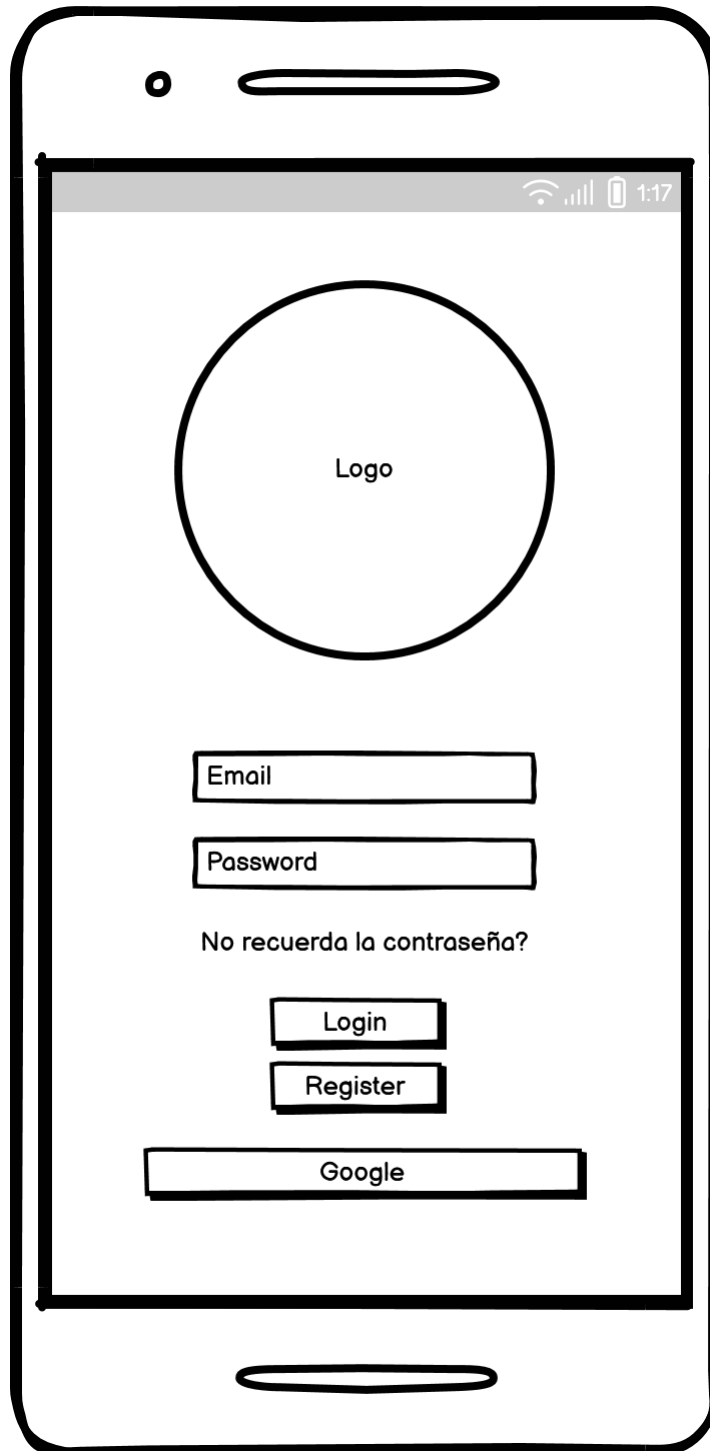


Esta pantalla es la primera en mostrarse, esta se quita automáticamente después de 4 segundos.

En ella solo debe aparecer un pequeño texto de bienvenida y el logo de la app



4.3.2. Mockup Login de usuario



En esta, también aparece principalmente el logo de la app.

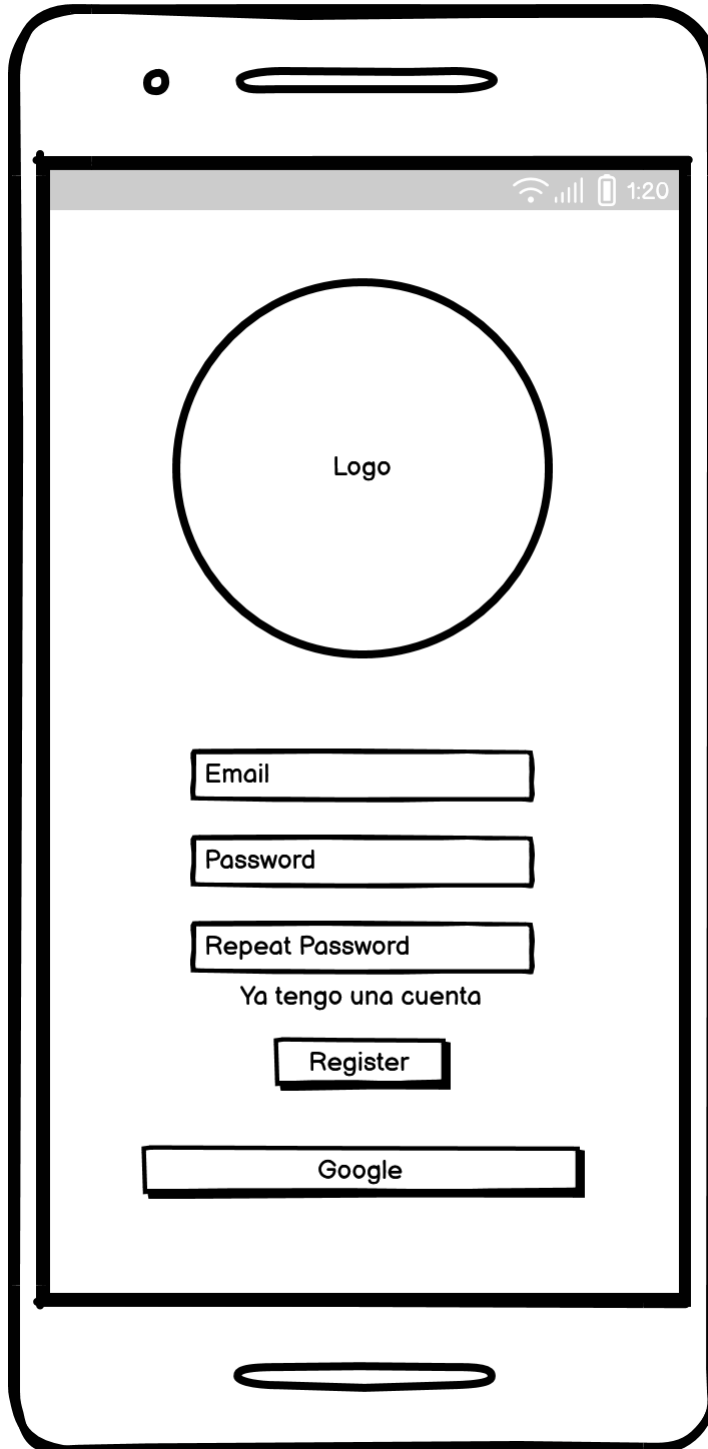
Permite introducir los datos para iniciar sesión.

Una opción que enviaría un correo en caso de que el usuario no se acuerde de las contraseña.



Además los botones correspondientes para acceder, registrarse o acceder con Google

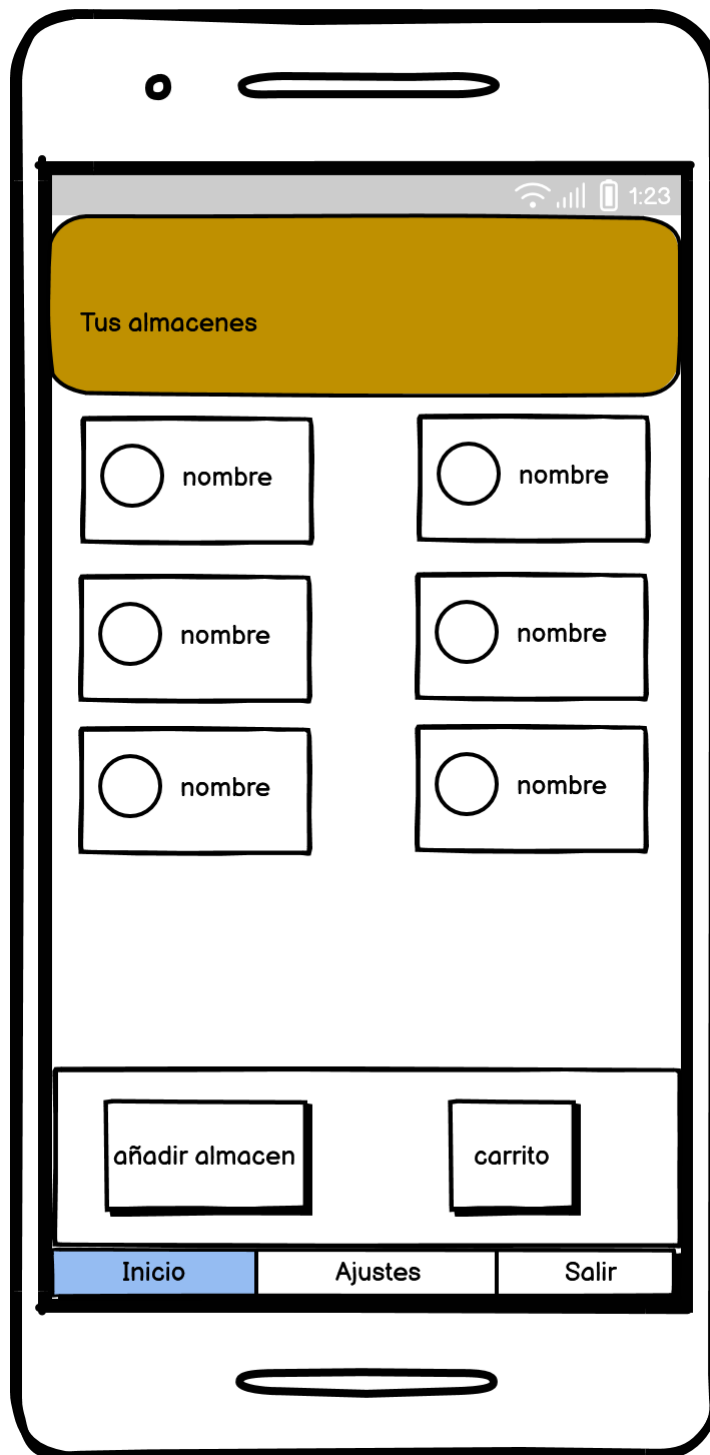
4.3.3. Mockup registro de usuario



Muy parecida a la anterior, con la peculiaridad de que en esta se necesita introducir la contraseña dos veces para poder registrarse.



4.3.4. Inicio de la aplicación

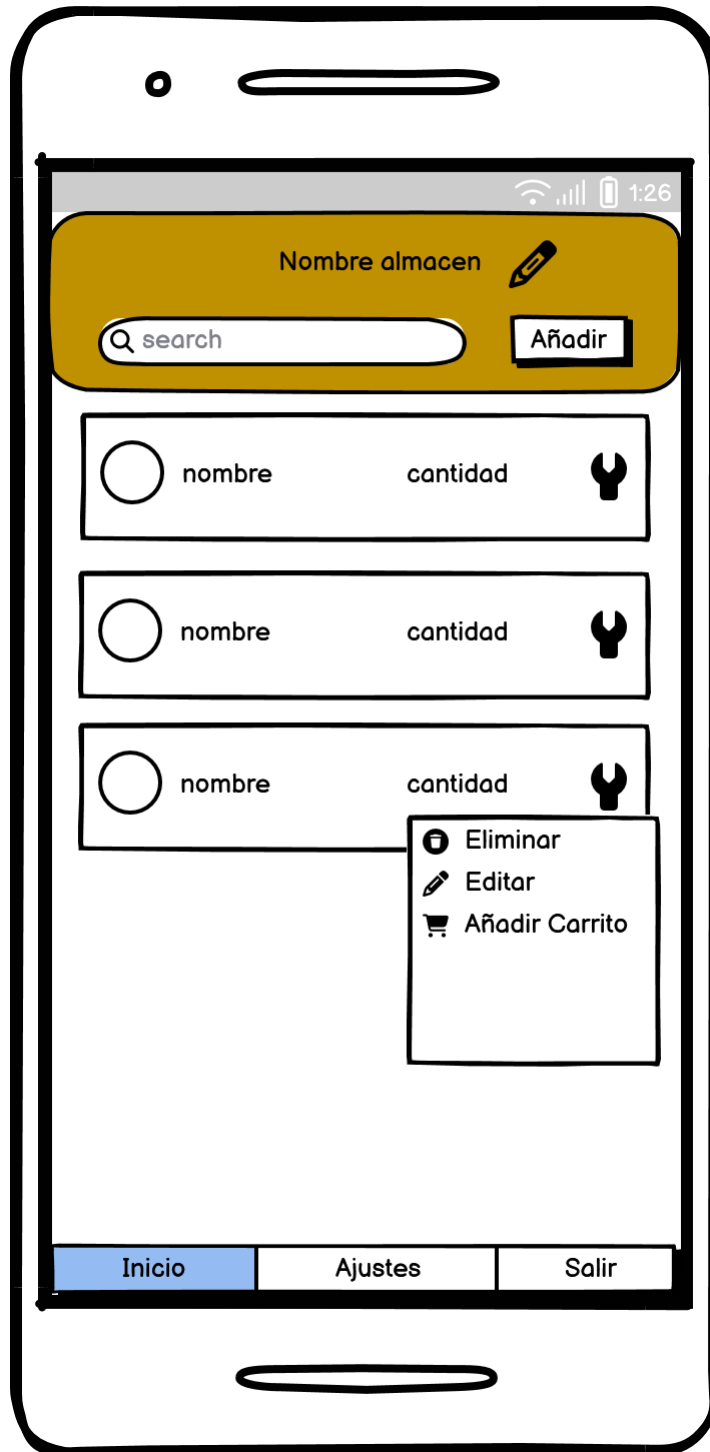


Nada más acceder a la app, debe aparecer la lista de los almacenes de los que dispone el usuario.

Debajo hay 2 botones, estos permiten como su nombre indica, añadir un nuevo almacén o acceder al carrito.



4.3.5. Mockup Almacén



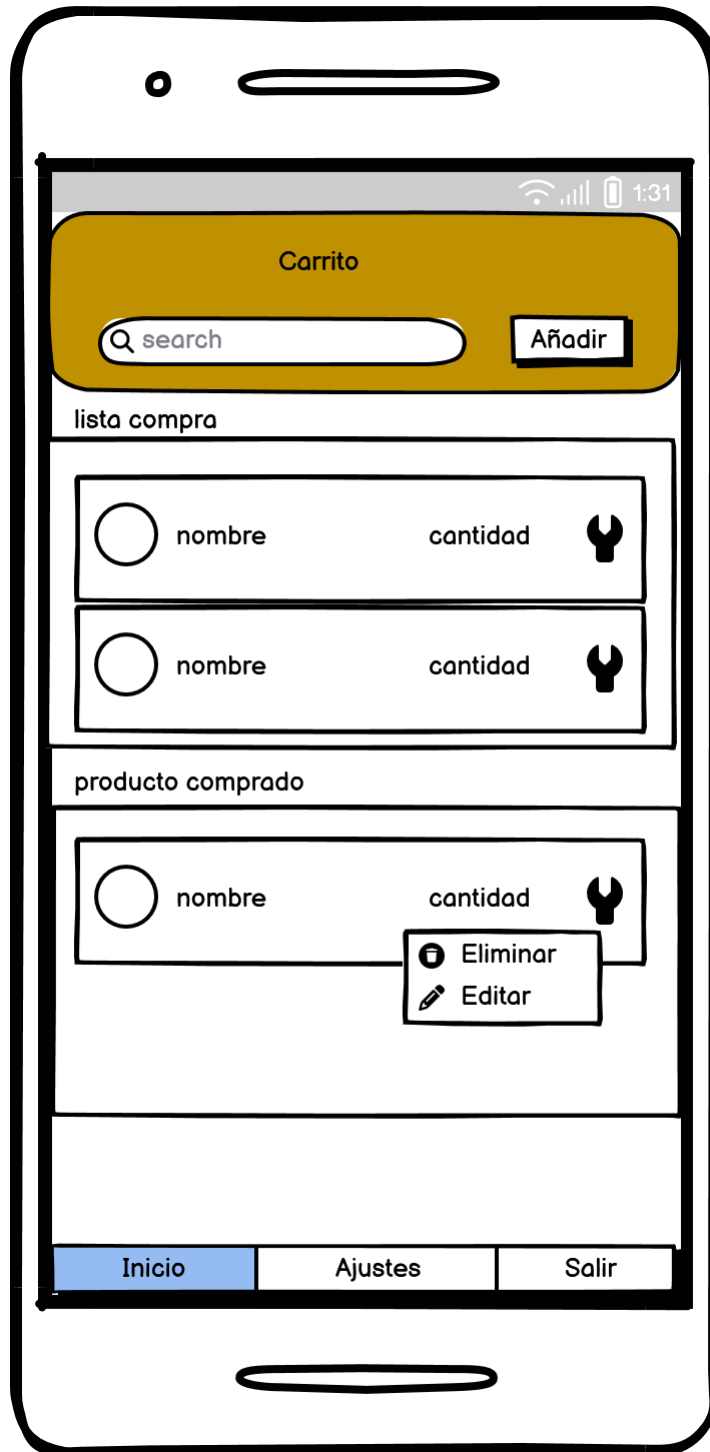
Dentro de cada almacén hay una lista con los alimentos que este contiene, así como la opción de buscar para añadir más.

El icono del lápiz será para acceder a las opciones del almacén

La llave inglesa se refiere a los ajustes de cada alimento que permite, eliminarlo, editar la cantidad o añadirlo al carrito



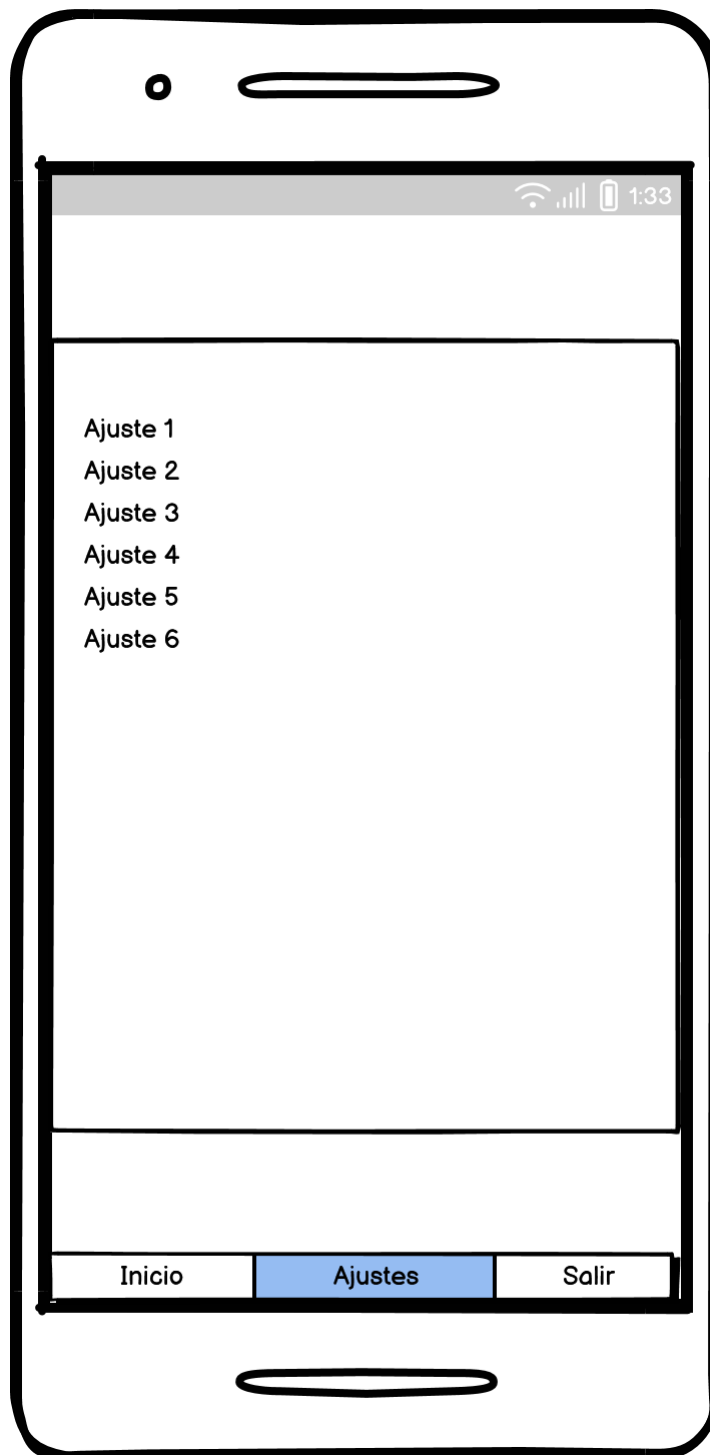
4.3.6. Mockup Carrito



Bastante parecido al anterior, con la peculiaridad de que este se divide en dos partes. En la primera aparecen los alimentos que aún se deben comprar. En la segunda justo debajo aparecen los alimentos ya comprados.



4.3.7. Mockup Ajustes de usuario

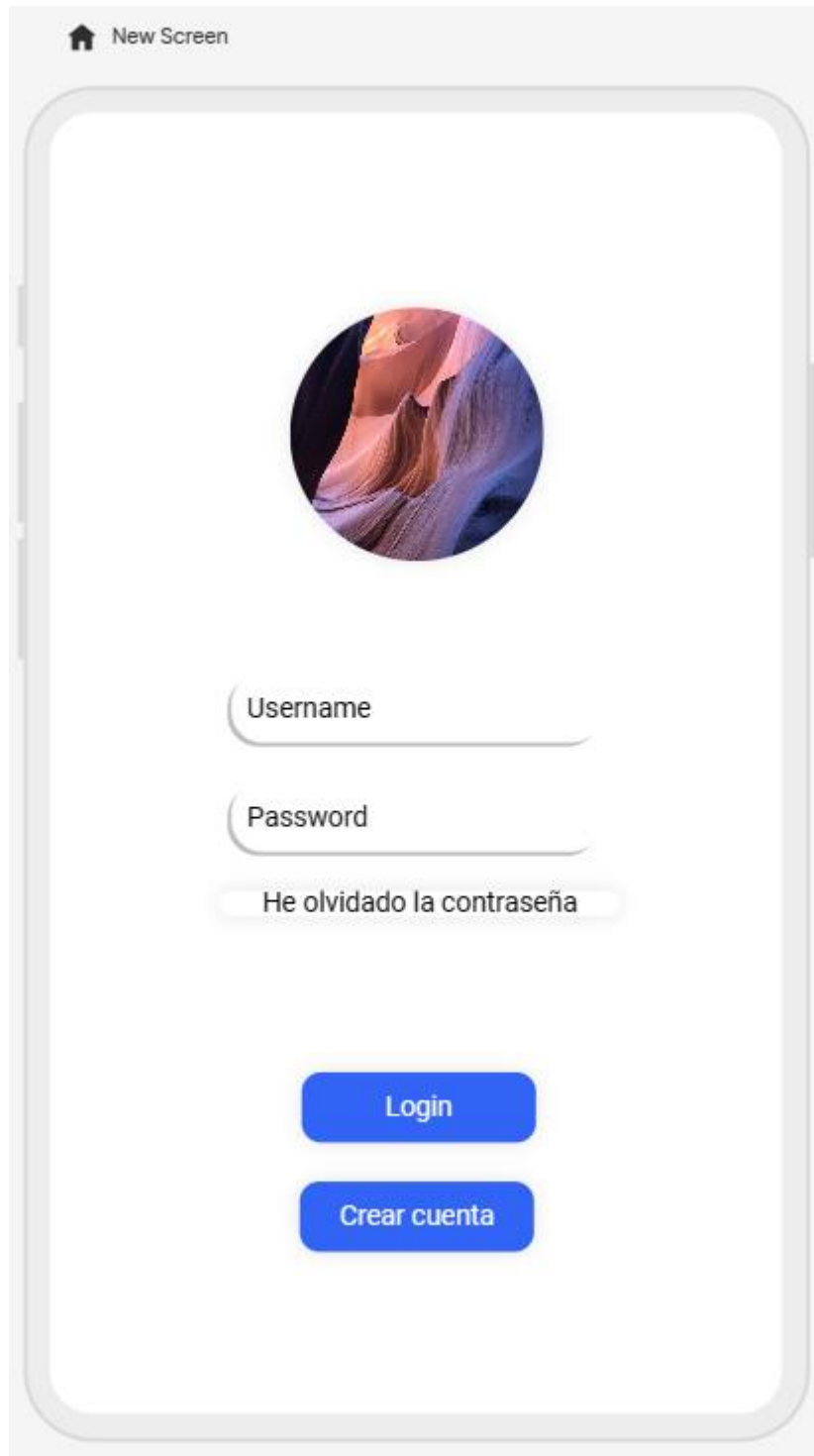


En este caso solo aparecen como ajustes pues aun no se cuales necesitaré.

Aquí se mostrará una lista de ajustes que el usuario podrá acceder, como pueden ser desde cambiar la contraseña, eliminar la cuenta, cambiar el mínimo de alimentos, etc.



4.3.8. Mockup profesionales





Home

Bienvenido, user



Organiza tus comidas

planifica tus comidas
y haz listas de la compra

Create

Your virtual pantries



Despensa



Despensa



despensa



despensa

New features



Crea un nuevo almacen



Carrito de la compra



Inicio



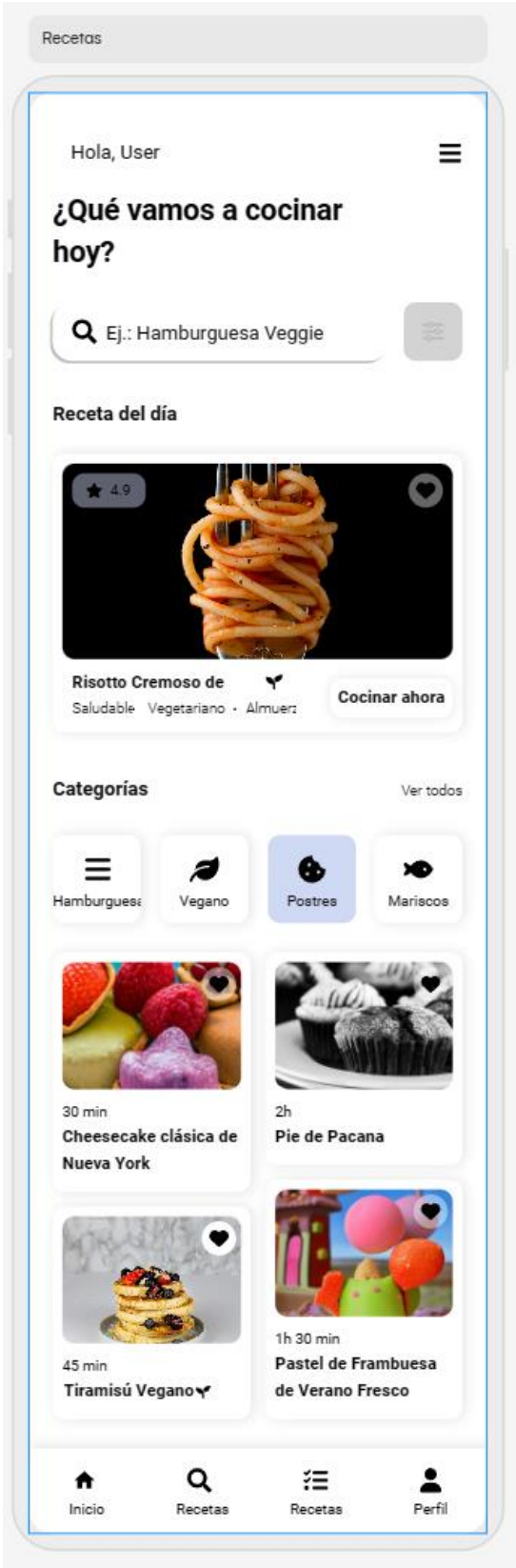
Buscar

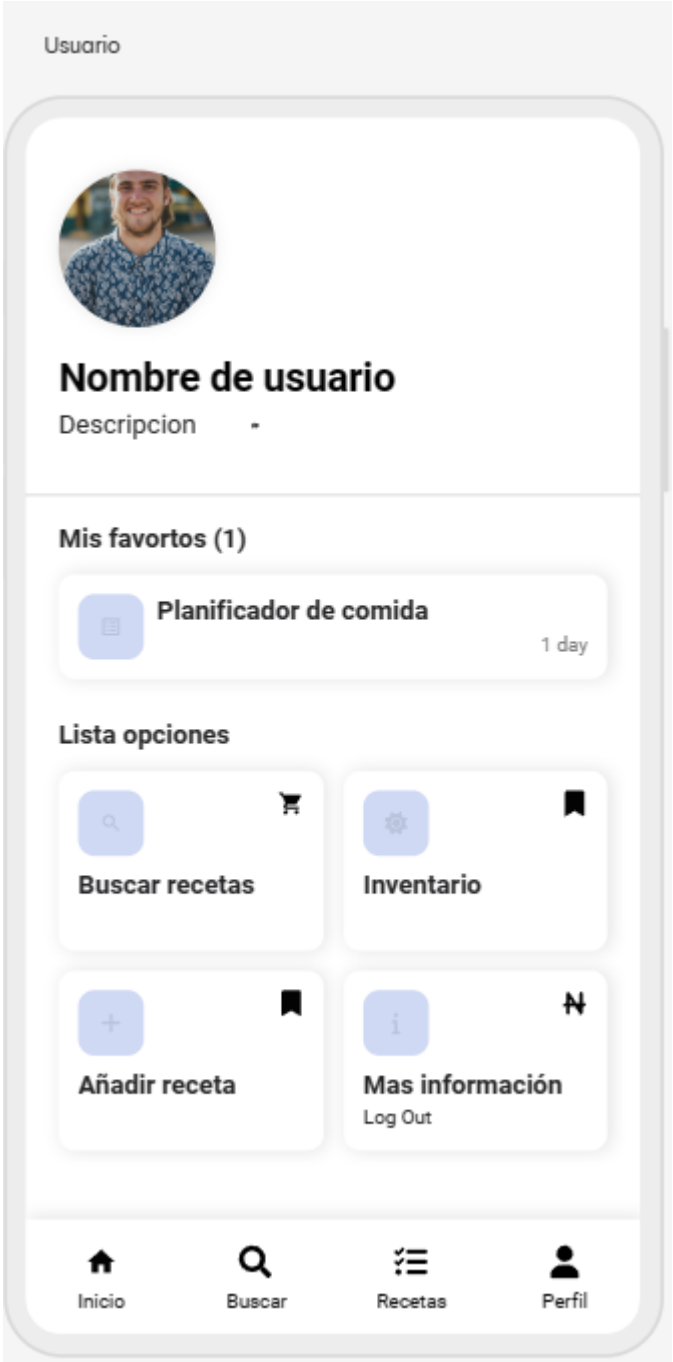


Recetas



Perfil







4.4 Diseño de la arquitectura de la aplicación

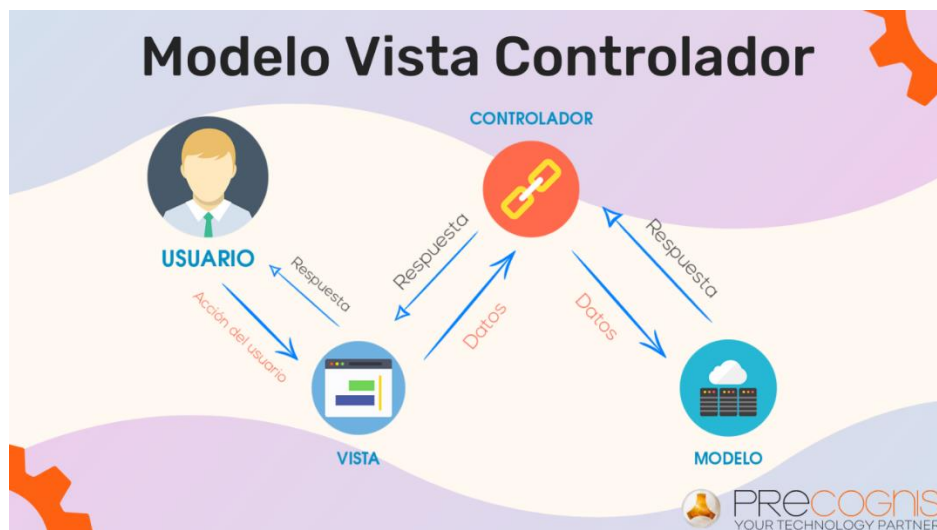
En el proceso de creación de mi aplicación, he optado por la arquitectura cliente/servidor. Este es un diseño de software que se divide en tres niveles, cada uno con funciones específicas. La idea principal de este enfoque es distinguir entre la capa de datos y la lógica de negocio de la capa de presentación, lo que facilita la administración y el mantenimiento del sistema.

Capa de datos: Este nivel más bajo se encarga de administrar la base de datos de la aplicación. Aquí se llevan a cabo todas las operaciones relacionadas con los datos, como la inserción, actualización, eliminación y consulta de datos. Esta capa interactúa directamente con la base de datos y procesa todas las solicitudes de datos.

Capa de negocio: También conocida como capa de lógica de negocio, esta capa actúa como un puente entre la capa de datos y la capa de presentación. El servidor de la aplicación reside en esta capa y controla las solicitudes y respuestas al usuario. Aquí se implementan las reglas de negocio, se procesan las solicitudes de los clientes, se interactúa con la capa de datos y se preparan las respuestas adecuadas.

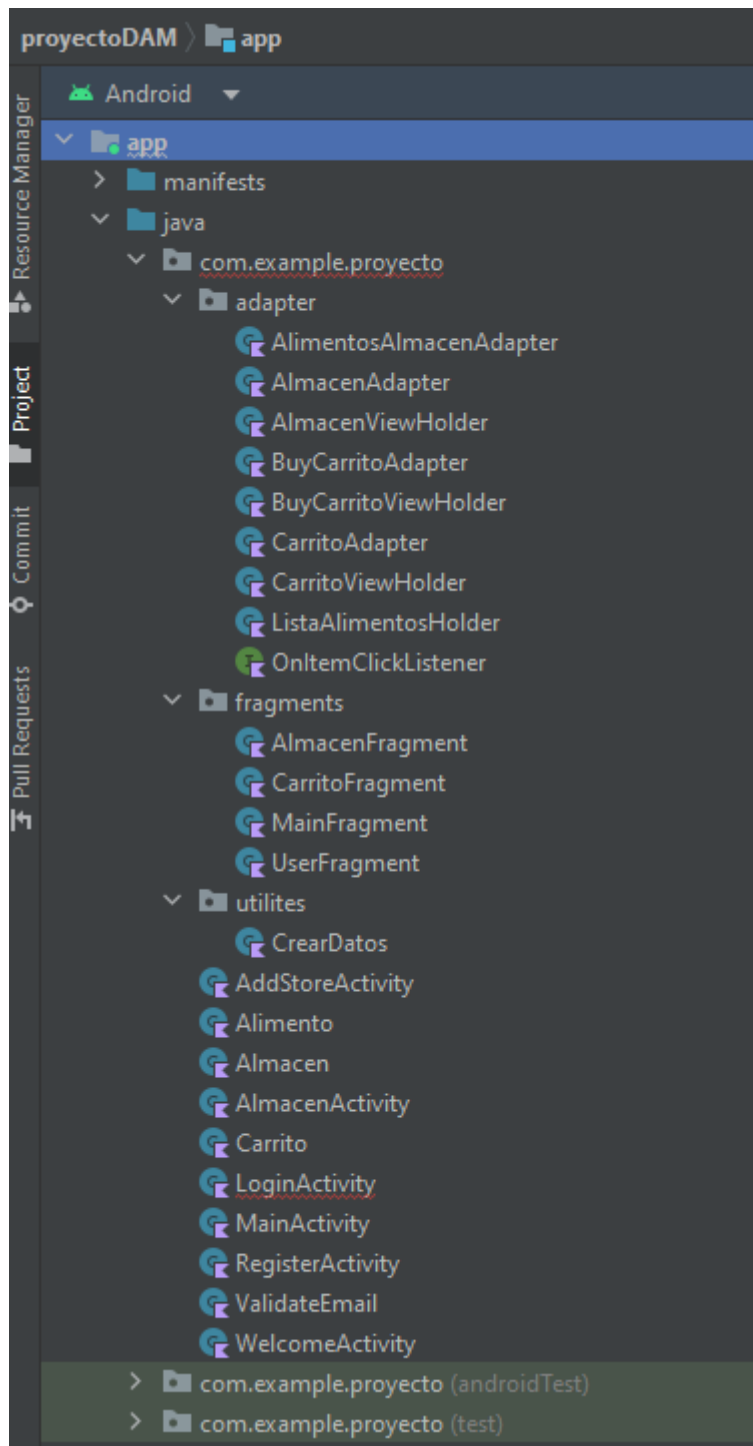
Capa de presentación: Este es el nivel que interactúa directamente con el usuario. Básicamente, es la interfaz de usuario y se encarga de presentar los datos al usuario de una manera comprensible. Esta capa recoge las entradas del usuario y las pasa a la capa de negocio para su procesamiento.

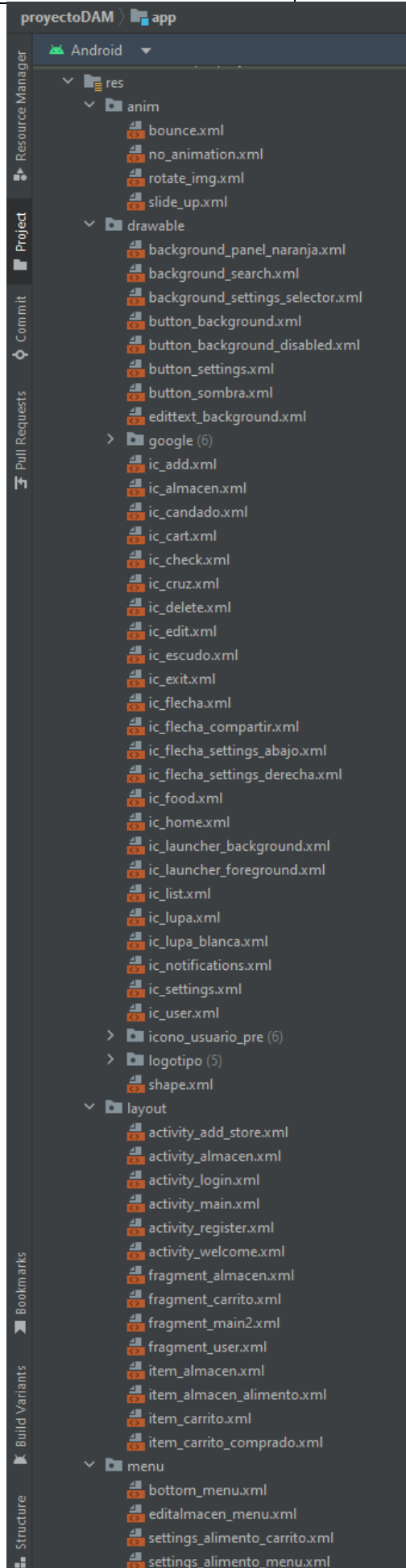
Además, voy a utilizar un patrón de diseño llamado MVC (Modelo-Vista-Controlador). Este es un patrón de arquitectura de software que separa la aplicación en tres componentes interconectados. El 'Modelo' corresponde a la capa de datos, la 'Vista' a la capa de presentación y el 'Controlador' actúa como intermediario, manejando la entrada del usuario y pasándola al modelo o a la vista según corresponda. Este patrón facilita la organización del código y hace que la aplicación sea más escalable y mantenible.

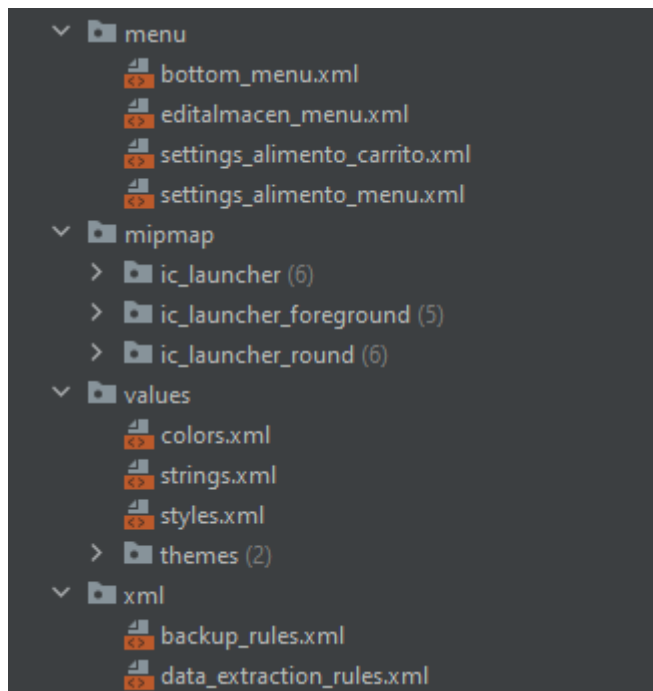




4.4.1. Diagrama interno de la estructura de mi app.







4.4.2. Tecnologías y herramientas usadas y descripción de las mismas

a) Hardware

a. **Ordenador:** en mi caso he usado un equipo propio con las siguientes características:

- i. **Gráfica:** RTX 4070
- ii. **CPU:** I5-13600KF
- iii. **RAM:** 32 GB

b) Software:

- a. **Gestor de base de datos:** En términos de administración de bases de datos, Firebase proporciona dos opciones: Firestore y Realtime Database. Ambas son bases de datos NoSQL que permiten guardar y sincronizar datos entre los usuarios en tiempo real. Firestore es una opción más reciente y opera con una estructura de documentos y colecciones, mientras que Realtime Database guarda los datos en un gran árbol JSON. Aunque no son bases de datos relacionales como SQLite 3, ofrecen flexibilidad y escalabilidad, lo que las hace perfectas para aplicaciones en tiempo real y de gran escala. Además, Firebase se encarga automáticamente de todas las operaciones de la base de datos, lo que minimiza la necesidad de administración del servidor.
- b. **GitHub:** GitHub es una plataforma de colaboración para desarrolladores que permite alojar proyectos y trabajar en ellos utilizando Git, un sistema de control de versiones. Aunque su uso principal es alojar código fuente de software, GitHub ofrece mucho más que eso.
- GitHub proporciona una interfaz de usuario intuitiva además de las funcionalidades de línea de comandos de Git. Los desarrolladores pueden seguir



proyectos y recibir notificaciones de actualizaciones, descargar versiones de proyectos para trabajar en su entorno local, o contribuir a proyectos a través de solicitudes de extracción.

Además de ser un repositorio de código, GitHub también ofrece funcionalidades como seguimiento de problemas, gestión de proyectos y la integración continua/despliegue continuo (CI/CD). También permite la creación de páginas web estáticas para proyectos, conocidas como GitHub Pages.

En resumen, GitHub es más que una herramienta esencial para cualquier desarrollador, ya que facilita la colaboración en proyectos de código abierto, mejora el flujo de trabajo de desarrollo de software y aumenta la visibilidad del proyecto en la comunidad de desarrolladores. Es más que un simple repositorio de código, proporcionando un ecosistema completo para el desarrollo de software.

- c. **Tecnologías y lenguajes:** Como lenguaje principal he usado Kotlin, para los diseños he usado XML y para la base de datos NoSQL

i. **Kotlin:** Kotlin es un lenguaje moderno y de tipado estático que prioriza la seguridad y la claridad del código. Se utiliza en una amplia gama de aplicaciones, incluyendo aplicaciones Android, aplicaciones de servidor y aplicaciones nativas. Empresas notables como Google, JetBrains y Pinterest lo utilizan en sus proyectos. Kotlin es un lenguaje multiparadigma, ya que soporta completamente la programación orientada a objetos y también incorpora características de la programación funcional. A diferencia de otros lenguajes, Kotlin es un lenguaje compilado, lo que puede resultar en un mejor rendimiento. Además, Kotlin puede interoperar con otros lenguajes que se ejecutan en la Máquina Virtual de Java (JVM), lo que lo hace extremadamente versátil y compatible con una amplia gama de plataformas.

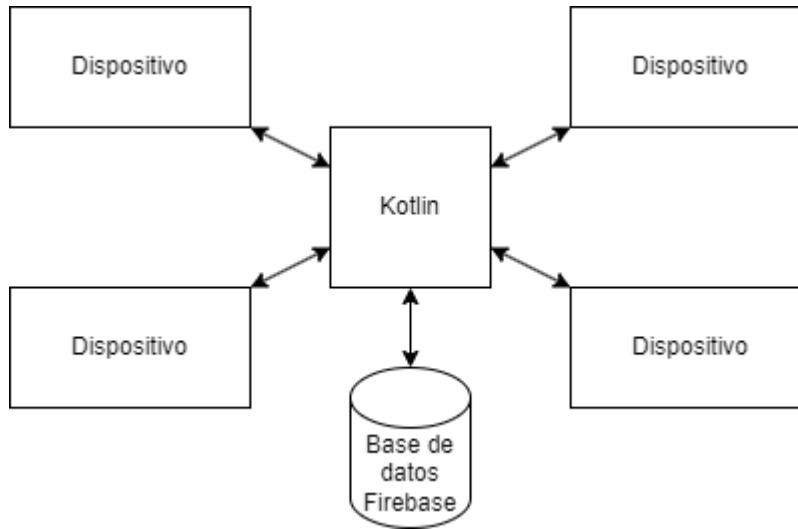
ii. **Sistema operativo:** Android es un sistema operativo para dispositivos móviles que se basa en el kernel de Linux y otros software de código abierto. Fue creado para dispositivos con pantalla táctil como smartphones, tabletas, relojes inteligentes, coches y televisores. Android Inc. fue la empresa que inicialmente desarrolló Android, pero Google LLC la adquirió en 2005.

Android se lanzó en 2007 junto con la creación de la Open Handset Alliance, un consorcio de empresas de hardware, software y telecomunicaciones dedicadas a promover los estándares abiertos para dispositivos móviles. El código fuente principal de Android se conoce como Android Open Source Project (AOSP), y se licencia principalmente bajo la Licencia Apache.

Android es el sistema operativo móvil más popular del mundo, con una cuota de mercado de más del 90%. Su gran flexibilidad y capacidad de personalización han contribuido a su popularidad entre los desarrolladores y usuarios.



4.4.3. Arquitectura de componentes de la aplicación



Los elementos fundamentales de la aplicación son:

Dispositivos: Los dispositivos se refieren a los aparatos móviles que los usuarios tienen y utilizan para interactuar con la aplicación. Estos pueden ser smartphones, tabletas o relojes inteligentes, todos capaces de instalar y ejecutar la aplicación. Los dispositivos son la principal interfaz entre el usuario y la aplicación, y es a través de estos que los usuarios pueden acceder a las funcionalidades de la aplicación.

Kotlin: Kotlin es el lenguaje de programación que se utiliza para construir la aplicación. Es un lenguaje moderno, conciso y seguro que se utiliza para escribir el código que hace funcionar la aplicación. Kotlin también se utiliza para manejar la conexión entre el usuario y la base de datos, asegurando que los datos se transmitan de manera eficiente y segura entre la base de datos y el dispositivo del usuario.

Firebase: Firebase es la base de datos que se utiliza para almacenar y gestionar los datos de la aplicación. Es una base de datos NoSQL que ofrece una variedad de servicios, incluyendo almacenamiento en la nube, autenticación de usuarios, análisis en tiempo real y mucho más. Firebase permite a la aplicación almacenar datos de manera segura y eficiente, y proporciona herramientas para gestionar y analizar estos datos. Con Firebase, la aplicación puede escalar para manejar un gran número de usuarios sin comprometer el rendimiento o la seguridad.



5 Documento de análisis y diseño

En este apartado se realizan las pruebas funcionales para el buen y correcto funcionamiento de mi aplicación

5.1 Pruebas

5.1.1. Pruebas funcionales

Usaré el siguiente diseño para mostrar las pruebas realizadas

CP_00	
Descripción	
Precondición	
Datos de entrada	
Datos de salida	

CP_001 Pantalla de carga	
Descripción	- Al iniciar la aplicación se muestra la pantalla de carga durante 4 segundos
Precondición	El usuario ha de iniciar la aplicación
Datos de entrada	
Datos de salida	Muestra la pantalla OK

CP_002 Usuarios registrados	
Descripción	Los usuarios registrados pueden acceder mediante el correo y la contraseña introducidos cuando se registraron.
Precondición	El usuario debe haberse registrado previamente
Datos de entrada	Se inserta el correo y la contraseña en el login
Datos de salida	a) Si los datos son correctos se redirige a la página principal. OK b) En caso de ser incorrecto muestra el error. OK

CP_003 Registro de usuarios	
Descripción	Los usuarios se pueden registrar rellenando unos campos
Precondición	El usuario debe haberle dado al botón de registro en la pantalla de login
Datos de entrada	Se inserta n los datos requeridos en la página de registro
Datos de salida	c) Si los datos son correctos se redirige a la página principal. OK d) En caso de ser incorrecto muestra el error. OK



CP_004 identificación	
Descripción	Si el usuario registrado ha iniciado sesión previamente, el inicio de sesión tiene que ser automático incluso y la app fue cerrada.
Precondición	El usuario debe haberse registrado previamente
Datos de entrada	Se accede a la aplicación.
Datos de salida	El usuario accede automáticamente. OK En caso de error aparece la pagina de login OK

CP_005 Cerrar sesión	
Descripción	Los usuarios que han iniciado sesión pueden cerrarla mediante un botón. El usuario será redirigido a la página de inicio.
Precondición	El usuario debe haberse registrado previamente
Datos de entrada	Esta opción aparece abajado a la derecha.
Datos de salida	El usuario cierra la sesión y es redirigido a la página login. OK En caso de error muestra un mensaje OK

CP_006 Carga de almacenes	
Descripción	Al acceder al inicio los almacenes que pertenecen al usuario deben estar cargados y listados
Precondición	El usuario debe haberse registrado previamente
Datos de entrada	
Datos de salida	Muestra los almacenes. OK En caso de error muestra un mensaje OK

CP_007 Carga de alimentos en almacen	
Descripción	Al acceder a un almacén los alimentos que pertenecen a este deben estar cargados y listados
Precondición	El usuario debe haberse registrado previamente El usuario ha de haber elegido un almacén
Datos de entrada	
Datos de salida	Muestra los alimentos. OK En caso de error muestra un mensaje OK

CP_008 Añadir alimento almacen	
Descripción	Dentro de cada almacén el usuario podrá añadir alimentos
Precondición	El usuario debe haberse registrado previamente El usuario ha de haber elegido un almacén El usuario debe haber escrito algo en el buscador
Datos de entrada	Campo introducido en el buscador
Datos de salida	Añade el nuevo alimento. OK Muestra la lista actualizada. OK En caso de error muestra un mensaje OK



CP_009 Añadir al carrito	
Descripción	Dentro de cada almacén el usuario podrá seleccionar un alimento y añadirlo al carrito
Precondición	El usuario debe haberse registrado previamente El usuario ha de haber elegido un almacén El usuario debe haber elegido la opción “añadir al carrito”
Datos de entrada	Opción escogida
Datos de salida	Añade el nuevo alimento. OK Muestra la lista actualizada. OK En caso de error muestra un mensaje OK

CP_010 Comprar alimentos	
Descripción	Dentro del carrito, en la primera lista “lista compra” el usuario toca encima de cualquier alimento este es considerado como comprado y debe pasar a la lista de abajo
Precondición	El usuario debe haberse registrado previamente El usuario debe haber elegido un alimento
Datos de entrada	Campo introducido en el buscador
Datos de salida	Añade el nuevo alimento. OK Muestra la lista actualizada. OK Elimina de la anterior lista. OK En caso de error muestra un mensaje OK

5.2 Instalación y configuración

En un principio mi idea era subir la apk a Google Play, pero después de una serie de problemas he decidido dejarlo para más adelante.

a) Requisitos

- Al ser una apk para móvil, con tener un dispositivo de este tipo y una versión minia de Android 17 sería suficiente

b) Instalación

- La apk esta disponible en mi repositorio de GitHub, lo único que has de hacer es descargarla e introducirla en su dispositivo móvil.
- Una vez con la apk en el móvil ha de acceder a los archivos de este, ahí la encontrará
- Lo siguiente es tan fácil como pulsar encima y darle instalar.

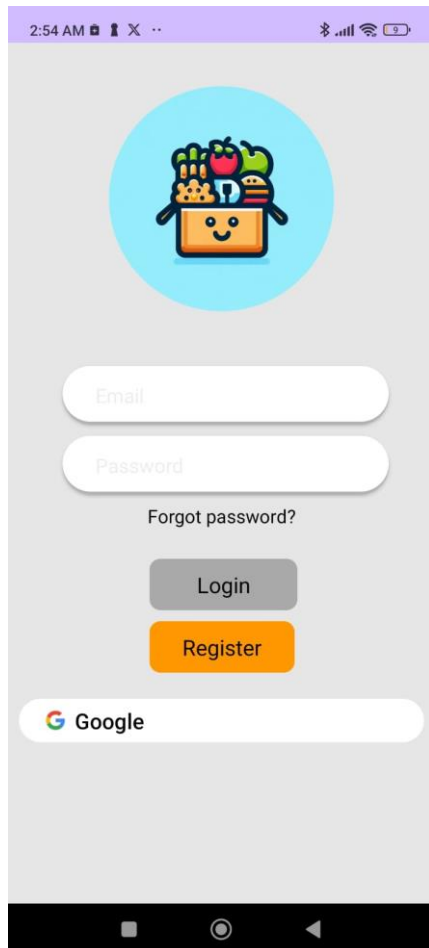


5.3 Manual de usuario

5.3.1. Página de inicio

En esta página el usuario podrá acceder con sus credenciales

En caso de no disponer, puede darle a la opción de registro



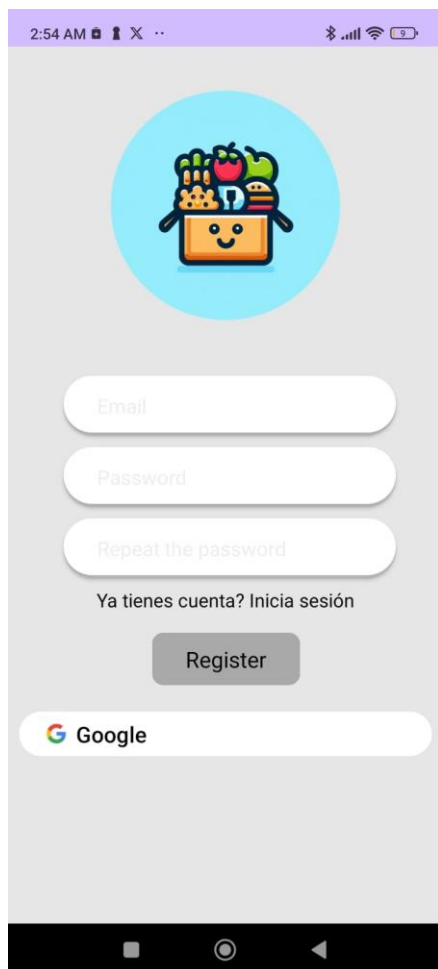


5.3.2. Página de registro

Muy parecida a la anterior en diseño.

Esta permite al usuario registrarse.

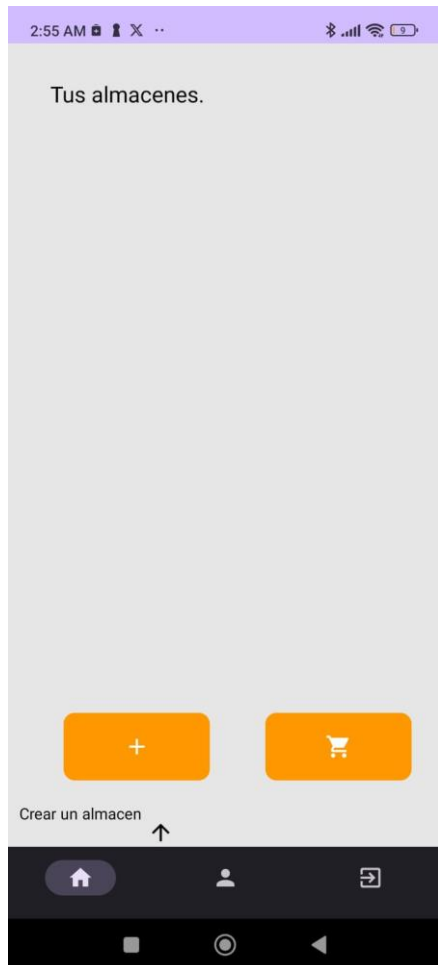
El botón no estará disponible hasta que el usuario introduzca los datos correctamente.

A screenshot of a mobile application's registration screen. At the top, there's a status bar showing the time as 2:54 AM and various icons. Below that is a large circular logo featuring a shopping basket with food items and a smiling face. Under the logo are three white input fields with rounded corners, labeled 'Email', 'Password', and 'Repeat the password'. Below these fields is a link that says 'Ya tienes cuenta? Inicia sesión'. Underneath the link is a grey 'Register' button. At the bottom of the form area is a white button with the Google logo and the text 'Google'. The entire screen has a light grey background.

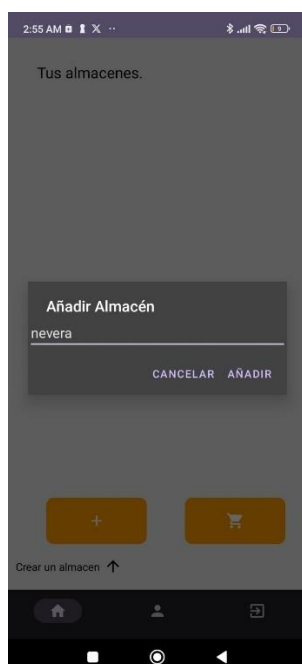
**5.3.3. Página de inicio**

Al acceder por primera vez la lista estará en blanco.

Justo debajo del botón para añadir almacenes hay un pequeño mensaje que lo avisa para los usuarios que accedan por primera vez



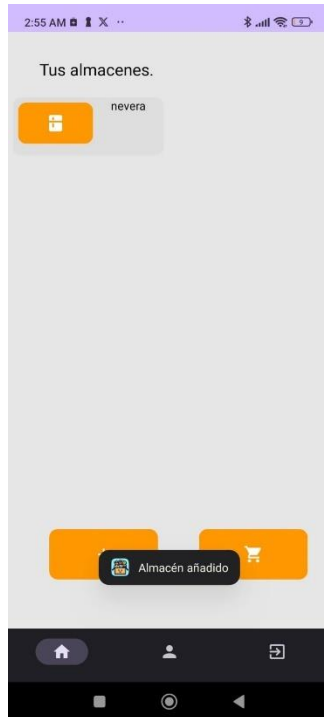
Al darle a ese botón saldrá un menú que permite añadir el nombre que queremos para el almacén





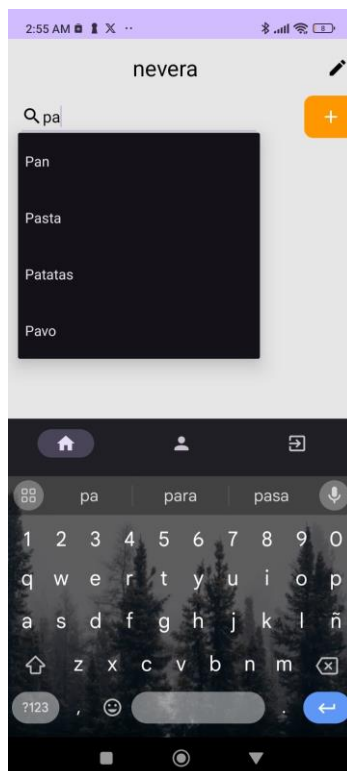
Y justo después nos aparecerá en la lista.

Al tener ya un almacén el mensajito desaparece



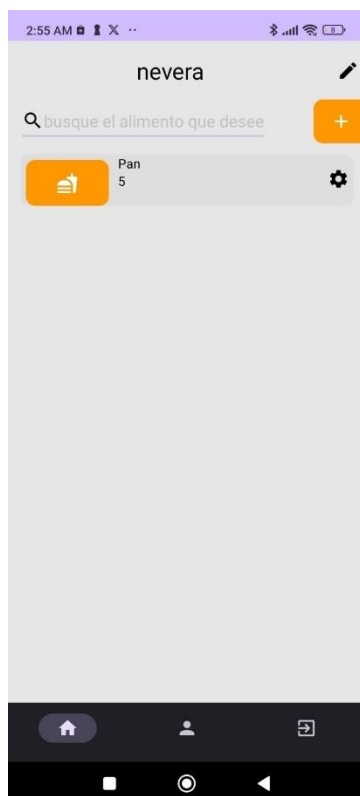
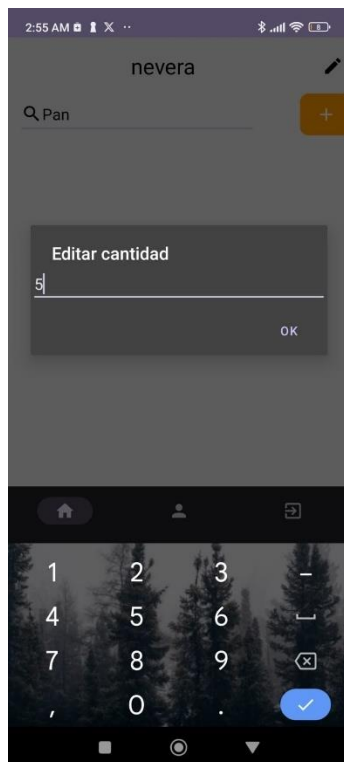
5.3.4. Página de Almacenes

Dentro del almacén podemos realizar una búsqueda de los alimentos que queremos agregar a este



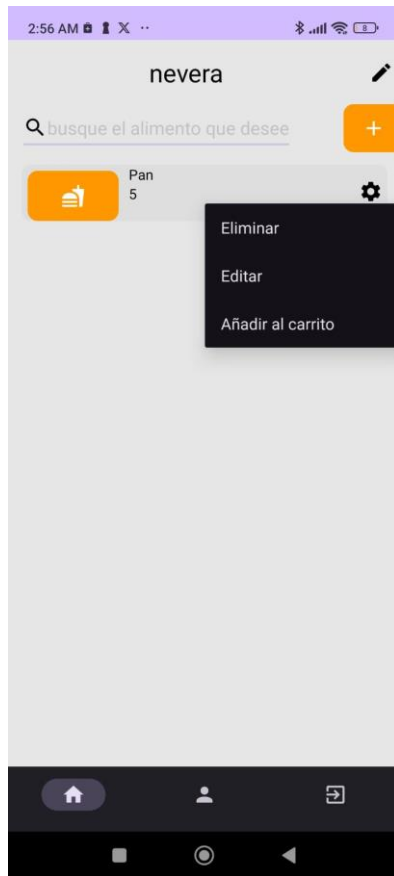


Podemos elegir la cantidad que queramos.





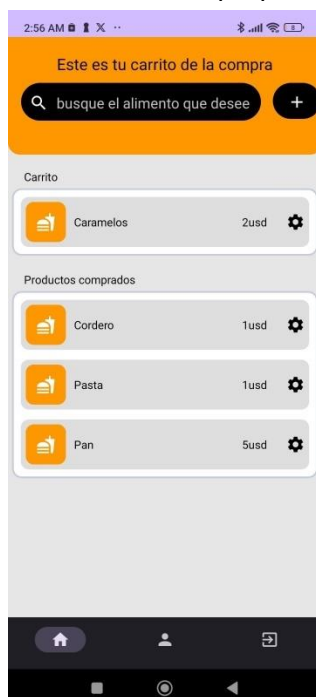
Si hacemos click en los ajustes de cada alimento se muestra un desplegable



La mas interesante es “Añadir al carrito” esta añade automáticamente el alimento al carrito de la compra

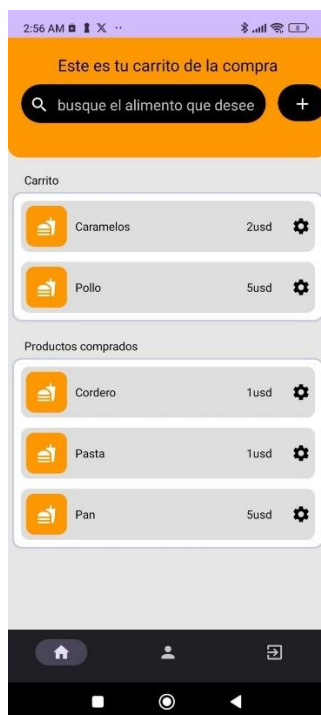
5.3.5. **Página de carrito**

Se muestra una pequeña lista, con alimentos en el carrito y ya comprados

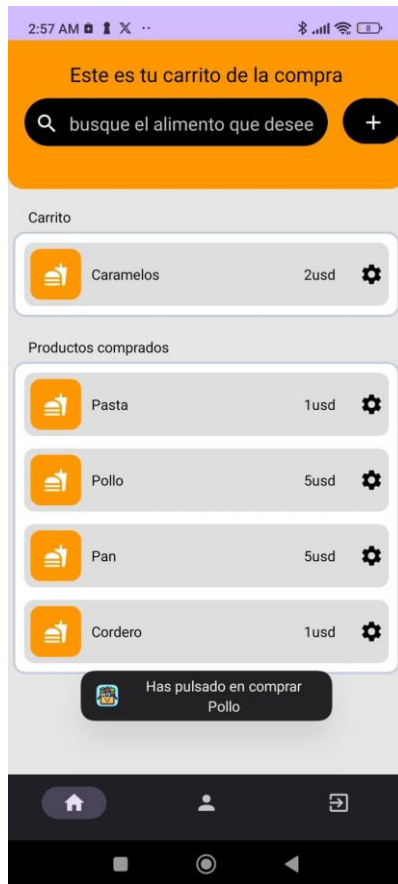




Podemos añadir manualmente alimentos

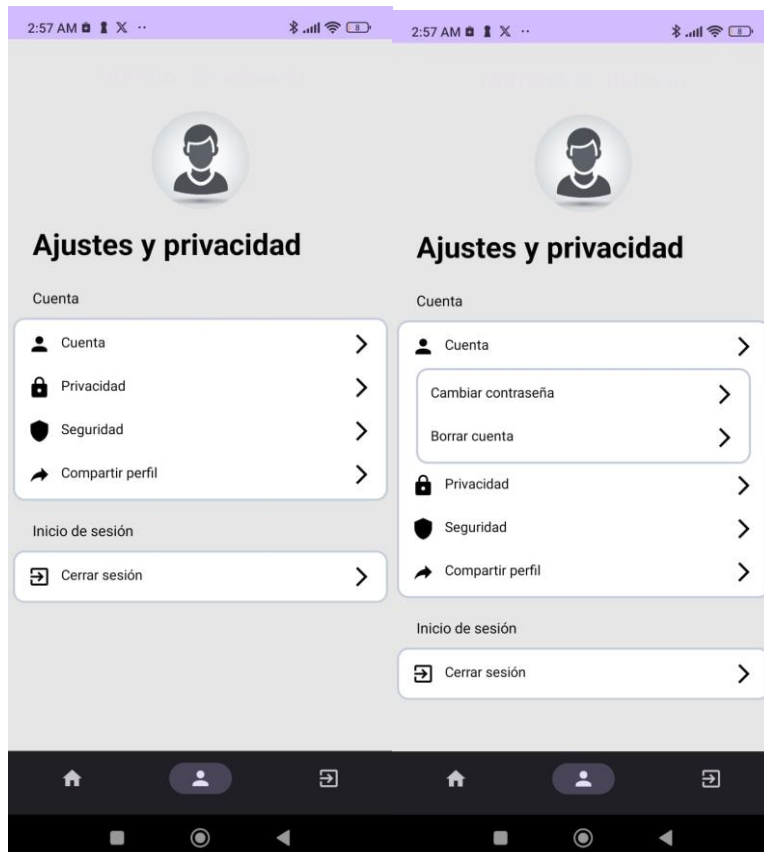


Y si hacemos click en este mismo, pasará a estar comprado



5.3.6. Página de Ajustes

En esta pagina estarán los futuros ajustes para cuando siga actualizando la app, de momento, al darle a “cuenta” se muestran 2 opciones





Por seguridad del usuario, si pulsa la opción de eliminar, dará una segunda oportunidad mostrando un mensaje.



6 Documento de cierre

Con la culminación de este Trabajo de Fin de Grado (TFG), pongo de manifiesto las competencias y habilidades que he adquirido a lo largo de este curso, así como durante mi formación en el Grado de Desarrollo de Aplicaciones Web. Este proyecto es un testimonio tangible de mi crecimiento y aprendizaje en este campo.

Este proyecto, aunque significativo, no está completo. Mi visión es continuar su desarrollo, añadiendo nuevas funcionalidades y mejorando las existentes, con el objetivo de que se convierta en una aplicación robusta y útil para el usuario promedio en el futuro.

Este viaje de desarrollo me ha brindado la oportunidad de aprender y explorar muchas cosas nuevas. Un ejemplo destacado es el lenguaje de programación Kotlin, que ha capturado mi interés y fascinación. Planeo continuar utilizando Kotlin para el desarrollo futuro de mi proyecto.

Uno de los logros más notables de este proyecto es el desarrollo de la característica principal de mi aplicación: la creación de almacenes virtuales para guardar alimentos. Esta funcionalidad tiene el potencial de ayudar a los usuarios a ahorrar al evitar el desperdicio de alimentos. Aunque esta característica está en sus etapas iniciales, estoy emocionado por su potencial y planeo continuar mejorándola en el futuro.



6.1 Diario de bitácora

Actividad	Fecha de ejecución
Captura de requisitos	1 al 2 de diciembre
Identificación de requisitos de los Usuarios.	1 al 2 de diciembre
Realización de casos de uso	2 al 5 de diciembre
Diagrama de secuencia iniciar sesión	5 al 5 de diciembre
Diagrama de secuencia registrase	8 de diciembre
Diseño de la base de datos	9 al 12 de diciembre
Diagrama Entidad Interrelación	12 al 14 de diciembre
Esquema relacional estándar	14 al 15 de diciembre
Grafo relacional	15 de diciembre
Normalización	16 al 17 de diciembre
Insertar datos en las tablas	22 al 22 de diciembre
Definir entorno de hardware y software.	1 al 3 de diciembre
Estudio de las tecnologías a utilizar.	4 de diciembre
Creación de las interfaces gráficas	27 al 30 de diciembre
Realizar pruebas y corregir errores.	03 al 05 de enero
Realizar pruebas unitarias.	04 al 05 de enero
Realizar pruebas del sistema completo.	05 al 05 de enero
Realizar informe de pruebas y resultados.	06 de enero
Realizar documentación. Informe de Instalación.	06 de enero
Captura de requisitos	06 de enero



Repositorio:

<https://github.com/UnaiSuarez/proyectoDAM>

7 Webgrafía

[CURSO DE KOTLIN](#)

GITHUB – CONTROL DE VERSIONES

<https://github.com/>
