

# GROCERLIST

Alumno: Noel Prieto Pardo

Fecha: 13/12/2024

# Índice

<b>1. FUNDAMENTACIÓN.....</b>	<b>1</b>
<b>2. DESTINATARIOS.....</b>	<b>1</b>
<b>3. OBJETIVOS.....</b>	<b>1</b>
<b>4. METODOLOGÍA.....</b>	<b>2</b>
4.1. HERRAMIENTAS.....	2
4.2. DIFICULTADES.....	3
<b>5. TEMPORALIZACIÓN.....</b>	<b>6</b>
<b>6. RECURSOS.....</b>	<b>6</b>
<b>7. CONCLUSIONES.....</b>	<b>6</b>
<b>8. BIBLIOGRAFÍA.....</b>	<b>7</b>
<b>I. APÉNDICE I: ESPECIFICACIÓN DE REQUISITOS.....</b>	<b>9</b>
<b>II. APÉNDICE II: MODELOS BASE DE DATOS.....</b>	<b>18</b>
<b>III. APÉNDICE III: DETALLE DE TEMPORALIZACIÓN.....</b>	<b>19</b>
<b>IV. APÉNDICE IV: GLOSARIO DE TÉRMINOS.....</b>	<b>22</b>

## 1. FUNDAMENTACIÓN

GROCERLIST es una aplicación de listas de la compra que pretende facilitar su creación y su uso. Aunque existe alguna aplicación similar (como Google Keep o Bring), GROCERLIST apuesta por añadir funcionalidades ausentes en otras iniciativas que no permiten una creación intuitiva de listas, un buscador de productos y la opción de compartirlas, así como la posibilidad de comparar los precios.

La idea surge a raíz de no haber encontrado una aplicación que cumpla todas mis expectativas. Actualmente, conviven aplicaciones de listas de uso general, pero no específicas para realizar listas de la compra. Aunque escasas aplicaciones permiten comparar precios, no he encontrado ninguna que conjunte los dos conceptos a través de una interfaz intuitiva y cómoda de usar para el usuario.

## 2. DESTINATARIOS

GROCERLIST está pensado para usarse en cualquier situación, lugar y por personas de diversas edades siempre que tengan acceso a internet. Sin embargo, su diseño se enfoca también hacia núcleos familiares (véase Apéndice IV) y, por ejemplo, personas en pisos compartidos, por la sencillez, rapidez y accesos generales que incorpora en su configuración. Además, ya que su interfaz se asemeja a otras aplicaciones de listas comunes, es particularmente eficiente en personas jóvenes.

## 3. OBJETIVOS

El objetivo general es desarrollar una aplicación para la creación, gestión y uso de listas de la compra, que puedan ser compartidas con otros usuarios, así como, realizar comparativas de precios entre distintos supermercados.

A partir de esta finalidad, surgen diferentes objetivos específicos. En primer lugar, la aplicación cuenta con un diseño intuitivo y cómodo de usar para el público general. También se ha dotado a GROCERLIST, en segundo lugar, la funcionalidad de creación y gestión de usuarios que surge de los requisitos de este proyecto. Y, por último, se ha implementado un buscador de productos que permite añadir productos existentes en los datos de la aplicación en las listas.

## 4. METODOLOGÍA

Para facilitar la explicación, he dividido las herramientas aplicadas en 6 apartados: bases de datos, frontend, backend, diseño, documentación y pruebas. Inmediatamente después, he enumerado una serie de dificultades surgidas durante el proceso de desarrollo.

### 4.1. HERRAMIENTAS

- Bases de datos
  - MySQL: un sistema de gestión de bases de datos relacionales.
- Frontend
  - Angular 17: un framework de código abierto basado en TypeScript que permite crear aplicaciones de alto nivel.
  - Visual Studio Code: un entorno de desarrollo de Microsoft.
  - Angular Material: un módulo construido por y para angular que permite introducir componentes angular siguiendo el diseño de Material Design (Apéndice IV).
- Backend
  - Spring: un framework de código abierto basado en Java. En este caso, se ha utilizado para realizar la Rest API del proyecto.
  - Spring Tools 4 para Eclipse: un entorno de desarrollo que contiene todas las herramientas necesarias para facilitar la creación de una aplicación Spring con la misma interfaz que el IDE Eclipse.
  - Spring Security: un framework basado en Java que aporta autenticación, autorización y otras medidas de seguridad.
  - Lombok: es una herramienta que permite simplificar el código a través de anotaciones dando lugar a código limpio, ordenado y más legible.
- Diseño
  - Figma: una herramienta de desarrollo de interfaz para realizar el prototipado.
  - SCSS: lenguaje de hoja de estilos en cascada para dar el estilo adecuado a la aplicación.
  - Power Designer: una herramienta de modelado empresarial colaborativo, utilizada para diseñar la base de datos.

- Neuromorphism: una herramienta para generar figuras que siguen la tendencia *neumorphic*.
- Hamburguers: una colección de botones tipo *hamburger*.
- Documentación:
  - Google Docs: herramienta para editar documentos.
  - Google Slides: herramienta para realizar presentaciones.
  - GitHub: una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git.
  - Jibble: explicado en el apartado de TEMPORALIZACIÓN.
- Pruebas
  - XAMPP: es un software para gestionar bases de datos MySQL.
  - Postman: plataforma API para probar la API del proyecto.

## 4.2. DIFICULTADES

### 4.2.1. Problemas de CORS

A grandes rasgos, CORS (Cross-Origin Resource Sharing) es un mecanismo que permite la solicitud de datos restringidos en una página web desde un dominio distinto al que sirvió el recurso en primer lugar. En mi caso, el error de CORS surgió al tener en local (127.0.0.0) el frontend y el backend desplegados en puertos distintos. Al tener introducida la seguridad de mi rest API y no permitir el acceso a los datos, salvo los de iniciar sesión y registro, la página web entendía que esos servicios restringidos no pertenecían ni al puerto donde despliega el front ni a los servicios que no requieren autenticación, desplegados en back. En concreto, recupera sin iniciar sesión y luego el back solicita los datos de acceso, un problema que se soluciona con un simple refresh de la página. La manera más inmediata para resolverlo se consigue con las anotaciones de Spring (@CrossOrigin) y colocando como dominio compartido el puerto donde despliega el front, pero esto no parece funcionar, ya que sigue intentando iniciar sesión cuando antes no era necesario.

### 4.2.2. Problemas con Angular Material

Como ya he comentado anteriormente, Angular Material es un módulo para aplicar el diseño de Material Design, un tipo de diseño de Google basado en cuadrículas, animaciones y transiciones receptivas. La problemática surge cuando se quiere modificar un solo elemento sin la necesidad de cambiar todo un tema completo. En la

documentación de Angular Material se muestran componentes con ejemplos de uso, muy útiles a nivel funcional, que no cuentan con la información necesaria para la edición de su estilo, y si lo hace, solamente se encuentra en el apartado de los ejemplos, y en ningún otro lugar se indican las partes o contenedores de los componentes. Por este motivo, la manera de manejar los estilos de los componentes Material se reduce a incomprensiva. Para cambiar el color de una opción (`mat-option`) no tienes que acceder a ese componente, sino al panel contenedor de las opciones del select (`mat-select`), pero no con su etiqueta, sino con la etiqueta de la clase de su panel, para luego acceder a la clase de estilo que se encuentra dentro del panel. En definitiva, algo poco intuitivo comparado con `html` y `css` nativos, y demasiado complejo para cambiar el color y el tamaño de la letra.

#### 4.2.3. Documentación de Angular

La documentación de este framework es excelente, con una sencilla búsqueda puedes acceder a toda la información relativa a los componentes, incluyéndose también en la página de detalle guías relacionadas. Aún así, estas están desactualizadas para utilizar componentes standalone, que, como he podido observar en el mundo empresarial y en distintos tutoriales no oficiales, son los que más se utilizan según la tendencia actual.

#### 4.2.4. Problemas de Lombok

Al utilizar Lombok para facilitar la lectura de código he descubierto un error que también sucede en algunos de los proyectos en los que he trabajado en mis prácticas laborales. Este ocurre al recuperar una lista recursiva, es decir, al haber una relación M:N se recupera un objeto que tiene una lista del tipo del objeto que produjo la llamada. En mi caso, `Lista` y `Usuario` pertenecen a una relación Many To Many que se resuelve en una tabla de relación `Tiene`. Siguiendo el método que propone Spring, se incluye una lista con usuarios dentro de `Lista` y una lista de listas en `Usuario` y, al utilizar la anotación `@Data` de Lombok y emplear la opción `ToString` da como resultado una búsqueda recursiva: si se busca una lista se recuperan los usuarios que a su vez recuperan las listas y así sucesivamente. Para solucionar este problema he recurrido a tres soluciones distintas, ya que ha surgido en múltiples fases y momentos del desarrollo.

La primera solución ha sido añadir en la arquitectura dos paquetes para manejar los tipos de respuestas y de peticiones, lo que ayuda a recuperar los datos necesarios y no el objeto entero en todas las ocasiones. Este me ha resultado el método más eficaz y útil, ya que, con pocas líneas de código, se puede plantear la idea de un servicio al completo, por ejemplo, un filtro de productos.

Para la segunda, he optado por utilizar `@JsonIgnore` cuando era indispensable recuperar el objeto como tal y la lista que provoca ese error de recursión.

Como última solución he completado la anotación de Lombok excluyendo la lista del `ToString` para facilitar y mejorar la legibilidad de los resultados de los servicios.

#### 4.2.5. APIs de supermercados

El API `losprecios.co`, al que he accedido utilizando una contraseña prestada, ha servido para poder nutrir la base de datos con datos reales de supermercados, pero en la utilización de la misma han surgido algunos desafíos.

En primer lugar, encontrar y solicitar el acceso a una API para simplemente buscar datos de productos ha sido prácticamente imposible. He intentado contactar con Carrefour, Dia y otros supermercados españoles, pero la respuesta ha sido inexistente o negativa. Además, APIs que están públicas, como la de Mercadona, no tienen los servicios necesarios para nutrir una base de datos o servir como un buscador de productos. En el caso de Mercadona, no hay ningún servicio que recupere una lista de productos ni por categoría ni por nombre, y tampoco una documentación a la que poder consultar. Por lo tanto, ha resultado imposible utilizar una API de algún supermercado conocido. Después de investigar y plantear un `scraper web`, lo cual habría llevado mucho tiempo y no era relevante, decidí probar con supermercados de habla hispana para nutrir mi base de datos; lo que me llevó a poder contactar con el creador de la API `losprecios.co`.

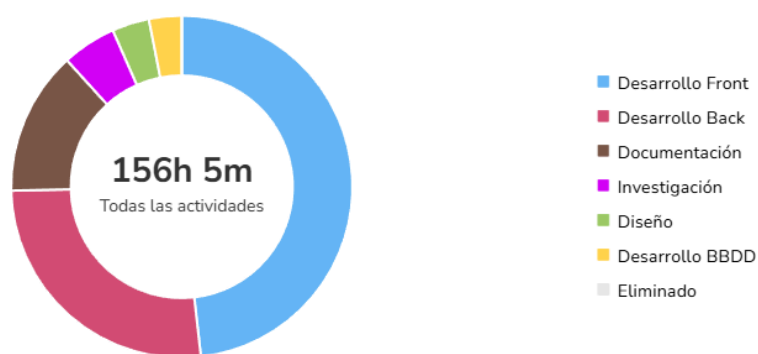
En segundo lugar, la utilización de la API `losprecios.co` no está optimizada. Hay pocos servicios y sirven más para recuperar un producto real (el objeto con código de barras, etc.), que el producto simbólico (nombre, categoría, etc.). Tampoco tiene los servicios necesarios para poder recuperar las categorías o filtrar los productos, aunque en la documentación se indique que sí.

En tercer lugar, en cuanto al uso en GROCERLIST, la API anteriormente mencionada no permite llamar a sus servicios a menos que haya pasado un segundo, que es suficiente, pero a veces ese segundo no es real y pasaba más tiempo. Esto deriva en errores de funcionamiento al utilizarlo en la aplicación.

Para conseguir superar estos desafíos opté por construir un script que se lanza desde el front para recuperar los productos e introducirlos en mi base de datos, lo cual otorga velocidad, comprensión y la capacidad de amoldar esos datos para las necesidades de GROCERLIST.

## 5. TEMPORALIZACIÓN

En cuanto a la temporalización, he utilizado [libble](#), una aplicación de gestión de trabajo que permite contar las horas de trabajo de manera ágil. A continuación, adjunto capturas con la cantidad de horas y el desglose del tiempo dedicado a cada actividad del proyecto:



## 6. RECURSOS

Un ordenador con estas especificaciones:

- Sistema operativo: Windows 10
- Procesador: Intel(R) Core(TM) i7-10750H
- Memoria RAM: 16 GB

Licencias:

- Power designer 16
- Jibble

## 7. CONCLUSIONES

Pese a las dificultades expuestas en apartados anteriores, GROCERLIST se ha desarrollado satisfactoriamente siguiendo la especificación de requisitos.

Respecto a líneas futuras, dadas las características y el tiempo estipulado para un proyecto de estas características, no se ha implementado en la aplicación una sección



de recomendaciones; una funcionalidad que definitivamente se podría añadir posteriormente. Asimismo, para adaptar la aplicación a distintas discapacidades o dificultades tecnológicas, sería interesante profundizar más en el apartado social y de destinatarios. La expansión de GROCERLIST tendría como base fundamental desarrollar un sistema de seguidores que nos permitiría visualizar las listas de otros usuarios (en caso de hacerlas públicas) y copiarlas. Y unido a los dos puntos anteriores, añadir GROCERLIST a otras plataformas, por ejemplo: android o iOs.

## 8. BIBLIOGRAFÍA

Documentación oficial de Spring: <https://spring.io/guides>

Documentación oficial de Angular: <https://angular.dev/>

Documentación oficial de Angular Material:

<https://material.angular.io/>

Documentación oficial Figma: <https://help.figma.com/hc/en-us>

Neuromorphism: <https://neuromorphism.io/>

Hamburguers:

<https://jonsuh.com/hamburgers/>

LeonardoAI: <https://app.leonardo.ai/>

ColorHunt: <https://colorhunt.co/palette/5f6f52a9b388fefae0b99470>

## I. APÉNDICE I: ESPECIFICACIÓN DE REQUISITOS

grocerlist	<b>Grocerlist</b> <b>Especificación de Requisitos</b>	<b>Noel Prieto Pardo</b> <b>Gregorio Fernández</b>
------------	--	---

### HOJA DE CONTROL

<b>Organismo</b>			
<b>Proyecto</b>	Grocerlist		
<b>Entregable</b>	Especificación de Requisitos		
<b>Autor</b>	Noel Prieto Pardo		
<b>Versión/Edición</b>	0101	<b>Fecha Versión</b>	25/11/2024
<b>Aprobado por</b>		<b>Fecha Aprobación</b>	DD/MM/AAAA
		<b>Nº Total de Páginas</b>	10

### REGISTRO DE CAMBIOS

<b>Versión</b>	<b>Causa del Cambio</b>	<b>Responsable del Cambio</b>	<b>Fecha del Cambio</b>
0100	Versión inicial	Noel Prieto Pardo	16/10/2024
0101	Versión 1.0.1	Noel Prieto Pardo	25/11/2024

### CONTROL DE DISTRIBUCIÓN

<b>Nombre y Apellidos</b>
Noel Prieto Pardo

grocerlist	Grocerlist Especificación de Requisitos	Noel Prieto Pardo Gregorio Fernández
------------	--	---

ÍNDICE

1 INTRODUCCIÓN.....	4
1.1 Alcance.....	4
1.2 Objetivos.....	4
2 INFORMACIÓN DEL DOMINIO DEL PROBLEMA.....	5
2.1 Introducción al Dominio del Problema.....	5
2.2 Glosario de Términos.....	5
3 CATÁLOGO DE REQUISITOS DEL SISTEMA A DESARROLLAR.....	6
3.1 Requisitos Funcionales del Sistema.....	6
3.1.1 Requisitos de Información del Sistema.....	6
3.1.2 Requisitos de Reglas de Negocio del Sistema.....	8
3.2 Requisitos No Funcionales del Sistema.....	9
3.2.1 Requisitos de Usabilidad.....	9
3.2.2 Requisitos de Seguridad.....	9
3.3 Restricciones Técnicas del Sistema.....	9
3.4 Requisitos de Integración del Sistema.....	10

grocerlist	<b>Grocerlist</b> <b>Especificación de Requisitos</b>	<b>Noel Prieto Pardo</b> <b>Gregorio Fernández</b>
------------	--	---

## 1 INTRODUCCIÓN

Se quiere realizar una aplicación para ayudar a la gestión de las compras mediante la creación de listas de la misma. Se ha observado que no existe una aplicación en la que se aúnen conceptos como la creación de listas, facilidad para compartirlas y la recomendación de precios. Grocerlist busca ser una aplicación importante para cualquier persona que busque hacer la compra de una manera más sencilla.

### 1.1 Alcance

El alcance del proyecto está pensado en 3 fases:

La primera fase consiste en la creación de las listas de la compra y este es el pilar fundamental. En esta fase se incluirán las operaciones CRUD sobre las listas así como añadir, eliminar y editar productos en las mismas.

En la segunda fase se contemplaría el apartado social de la aplicación. En esta fase se centraría en las funciones de compartir las listas entre los dos o más usuarios.

La tercera fase se centraría en la recomendación de precios. Con ayuda de APIs de supermercados se compararía el precio en productos seleccionados de primera necesidad.

De estas fases en la versión 1.0.0 se incluirá la primera y se intentará tener, si no todas, algunas de las funcionalidades de la segunda fase. Estas funcionalidades y requisitos se especifican más detalladamente en el apartado 3.

### 1.2 Objetivos

- Una aplicación fácil de entender para poder realizar listas de la compra.
- Una interfaz visual y agradable.
- Poder gestionar tu usuario.
- Poder compartir listas de la compra a tus amigos o familiares.

grocerlist	<b>Grocerlist</b> <b>Especificación de Requisitos</b>	<b>Noel Prieto Pardo</b> <b>Gregorio Fernández</b>
------------	--	---

## 2 INFORMACIÓN DEL DOMINIO DEL PROBLEMA

### 2.1 Introducción al Dominio del Problema

El campo sobre el que se va a trabajar la aplicación utiliza un lenguaje natural al que ya estamos acostumbrados. Aún así cabe recalcar varias cosas:

- Las listas de la compra contienen productos y de cada producto se puede seleccionar la cantidad.
- Los usuarios pueden tener infinidad de listas de la compra, por eso no se indica como requisito una cantidad máxima, al menos en esta versión.
- Para compartir listas de la compra no hace falta tener ningún tipo de conexión extra en la aplicación. Al contrario de como se hace en otras aplicaciones no hace falta ser seguidor para acceder a las listas, simplemente una invitación directa.

### 2.2 Glosario de Términos

**Usuario:** se refiere a la persona que se da de alta para utilizar el sistema.

**Listas:** se refiere a las listas de la compra, del tipo *todo*.

**Productos:** en general se refiere a todo lo que se pueda añadir a la lista de la compra, ya sean alimentos u otros elementos.

grocerlist	<b>Grocerlist</b> <b>Especificación de Requisitos</b>	<b>Noel Prieto Pardo</b> <b>Gregorio Fernández</b>
------------	--	---

### 3 CATÁLOGO DE REQUISITOS DEL SISTEMA A DESARROLLAR

#### 3.1 Requisitos Funcionales del Sistema

##### 3.1.1 Requisitos de Información del Sistema

<b>IS-001</b>	<i>Información sobre los usuarios</i>
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a los usuarios. En concreto: <ul style="list-style-type: none"><li>- Nombre de usuario</li><li>- Nombre</li><li>- Apellidos</li><li>- Correo electrónico</li><li>- Password</li></ul>
<b>Datos específicos</b>	Los apellidos se almacenarán separados como <i>apellido1</i> y <i>apellido2</i> .
<b>Comentarios</b>	El nombre de usuario servirá para iniciar sesión.

<b>IS-002</b>	<i>Información sobre las listas</i>
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a los usuarios. En concreto: <ul style="list-style-type: none"><li>- Identificador de la lista</li><li>- Nombre de la lista</li><li>- Usuario creador</li></ul>
<b>Datos específicos</b>	
<b>Comentarios</b>	

<b>IS-003</b>	<i>Información sobre los productos</i>
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente a los productos. En concreto: <ul style="list-style-type: none"><li>- Identificador del producto</li><li>- Nombre del producto</li><li>- Categoría</li></ul>

grocerlist	<b>Grocerlist</b> <b>Especificación de Requisitos</b>	<b>Noel Prieto Pardo</b> <b>Gregorio Fernández</b>
------------	--	---

Datos específicos	Categoría se almacena en una tabla con información relevante.
Comentarios	

IS-004	<i>Información sobre las categorías</i>
Descripción	El sistema deberá almacenar la información correspondiente a las categorías. En concreto: <ul style="list-style-type: none"><li>- Identificador de categoría</li><li>- Nombre de la categoría</li></ul>
Datos específicos	Categoría y tipo de almacenaje serán almacenados en dos tablas con información relevante.
Comentarios	

IS-005	<i>Información sobre los tipos de almacenaje</i>
Descripción	El sistema deberá almacenar la información correspondiente a los productos. En concreto: <ul style="list-style-type: none"><li>- Identificador del tipo</li><li>- Nombre del tipo</li><li>- Unidades</li></ul>
Datos específicos	Se incluirá un tipo de almacenaje que sea "Por definir" ya que el usuario es el que elige la cantidad que desea comprar y en las unidades que prefiera.
Comentarios	

IS-006	<i>Información sobre la relación entre productos y listas (incluye)</i>
Descripción	El sistema deberá almacenar la información correspondiente a la relación entre productos y listas. En concreto: <ul style="list-style-type: none"><li>- Identificador del producto</li><li>- Identificador de la lista</li><li>- Si están comprados o no</li><li>- Tipo de almacenaje</li><li>- Cantidad</li></ul>



grocerlist	<b>Grocerlist</b> <b>Especificación de Requisitos</b>	<b>Noel Prieto Pardo</b> <b>Gregorio Fernández</b>
------------	--	---

Datos específicos	Cantidad siempre en número.
Comentarios	<i>Se pondrá por defecto una cantidad de 0 y tipo de almacenaje "Por definir"</i>

IS-007	<i>Obtención de la información sobre productos</i>
Descripción	Se utilizará el api de <a href="https://losprecios.co">losprecios.co</a> .
Datos específicos	Se recogerán de esta API categorías y productos.
Comentarios	

### 3.1.2 Requisitos de Reglas de Negocio del Sistema

RN-001	<i>Modificación de la información sobre usuarios</i>
Descripción	Cada usuario podrá modificar su información personal.
Comentarios	

RN-002	<i>Modificación de la información sobre listas</i>
Descripción	Los usuarios con acceso a las listas podrán modificar la información sobre estas.
Comentarios	<i>Los datos a modificar son: título y productos (si están comprados o no y añadir y eliminar los mismos).</i>

RN-003	<i>Eliminar listas</i>
Descripción	Los usuarios creadores podrán eliminar sus listas, al contrario de un usuario con una lista compartida que no podrá borrarla ni compartirla.
Comentarios	

grocerlist	<b>Grocerlist</b> <b>Especificación de Requisitos</b>	<b>Noel Prieto Pardo</b> <b>Gregorio Fernández</b>
------------	--	---

<b>RN-004</b>	<i>Compartir listas</i>
<b>Descripción</b>	Los usuarios podrán compartir listas siempre que las dos partes acepten que se hayan compartido las listas.
<b>Comentarios</b>	

## 3.2 Requisitos No Funcionales del Sistema

### 3.2.1 Requisitos de Usabilidad

<b>NF-001</b>	<i>Acceso sencillo a la creación de listas y añadir productos</i>
<b>Descripción</b>	El sistema deberá tener botones intuitivos que permitan crear listas de manera sencilla y añadir productos a las mismas.
<b>Comentarios</b>	

<b>NF-002</b>	<i>Acceso sencillo a los productos</i>
<b>Descripción</b>	El sistema deberá tener un sistema de búsqueda de productos por nombre y categoría que faciliten la inserción de productos.
<b>Comentarios</b>	

### 3.2.2 Requisitos de Seguridad

<b>NF-003</b>	<i>Acceso con credenciales</i>
<b>Descripción</b>	El sistema deberá permitir el acceso a través de credenciales.
<b>Comentarios</b>	

grocerlist	<b>Grocerlist</b> <b>Especificación de Requisitos</b>	<b>Noel Prieto Pardo</b> <b>Gregorio Fernández</b>
------------	--	---

### 3.3 Restricciones Técnicas del Sistema

RT-001	<i>Compatibilidad con navegadores</i>
Descripción	El sistema deberá ser compatible con los navegadores Microsoft Edge, Mozilla Firefox y Google Chrome en su última versión.
Comentarios	

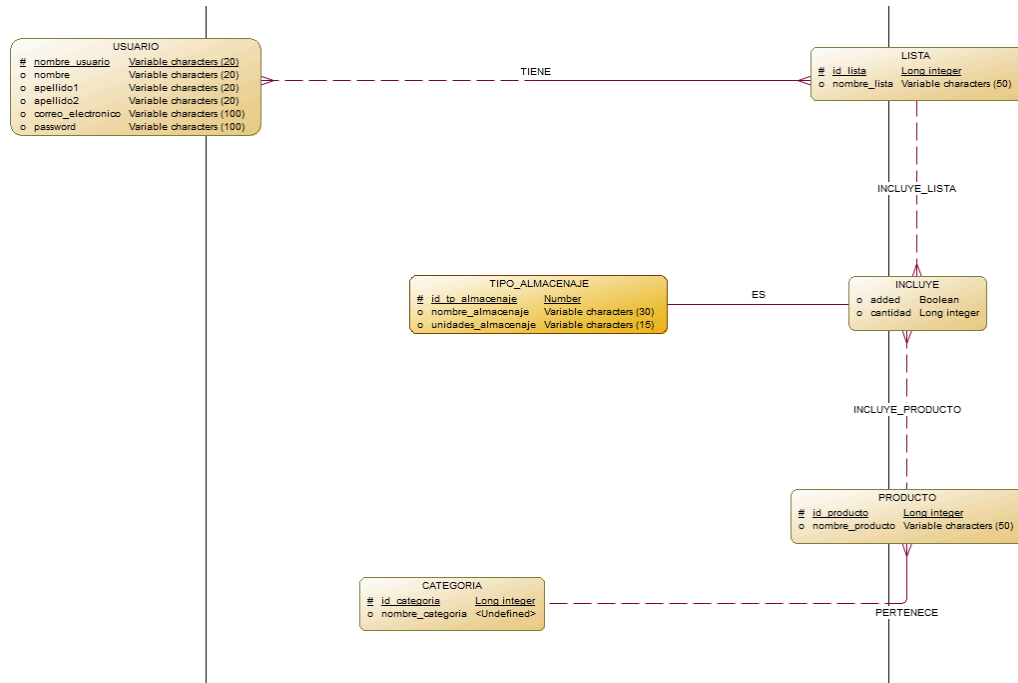
RT-002	<i>Tecnologías de desarrollo</i>
Descripción	El sistema deberá desarrollarse en base a las siguientes tecnologías: <ul style="list-style-type: none"><li>- Sistema gestor de base de datos MySQL</li><li>- IDEs: Spring Tools Suite y Visual Studio Code</li><li>- Angular 17 para front-end</li><li>- Java 17 para back-end</li></ul>
Comentarios	

### 3.4 Requisitos de Integración del Sistema

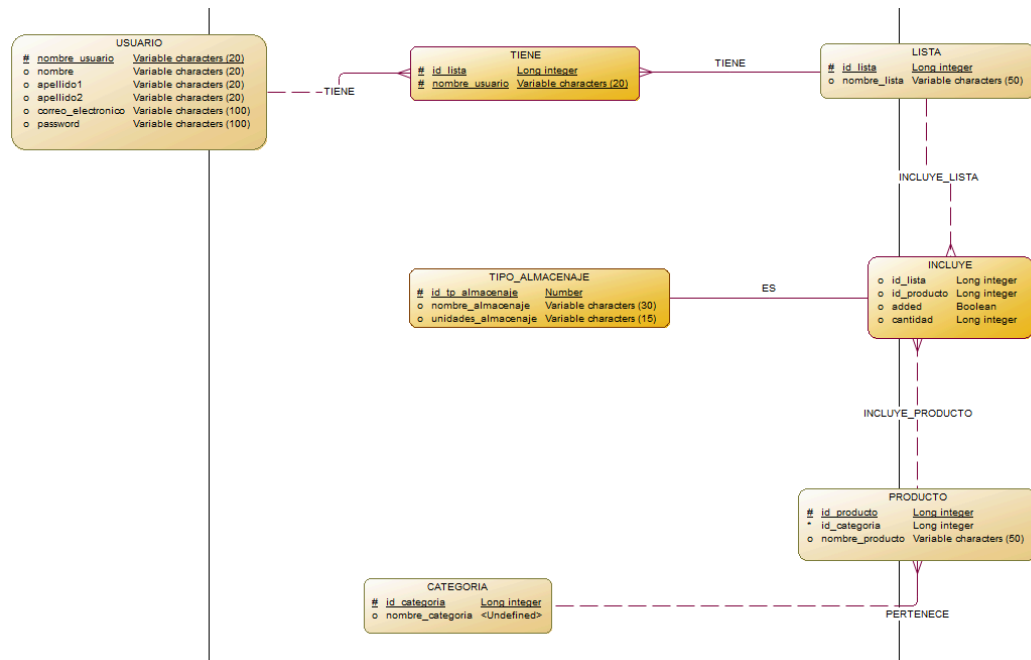
RIS-001	<i>Obtención de la información sobre productos</i>
Descripción	Se utilizará el api de <a href="https://losprecios.co">losprecios.co</a> .
Datos específicos	Se recogerán de esta API categorías y productos.
Comentarios	

## II. APÉNDICE II: MODELOS BASE DE DATOS

### Modelo Entidad Relación:



### Modelo Lógico:



## III. APÉNDICE III: DETALLE DE TEMPORALIZACIÓN

## Informe de tiempo registrado

Intervalo de fechas 04 diciembre 2023 - 03 diciembre 2024  
# Miembros 1

ACTIVITY	CÓDIGO ACT.	FECHA	DÍA	HRS REGISTRADAS
Desarrollo Back	ACT-215	09/10/2024	Miércoles	4h 00m
Desarrollo Back	ACT-215	10/10/2024	Jueves	2h 00m
Desarrollo Back	ACT-215	14/10/2024	Lunes	2h 00m
Desarrollo Back	ACT-215	30/10/2024	Miércoles	1h 30m
Desarrollo Back	ACT-215	31/10/2024	Jueves	1h 30m
Desarrollo Back	ACT-215	01/11/2024	Viernes	1h 00m
Desarrollo Back	ACT-215	04/11/2024	Lunes	1h 00m
Desarrollo Back	ACT-215	06/11/2024	Miércoles	0h 15m
Desarrollo Back	ACT-215	08/11/2024	Viernes	1h 00m
Desarrollo Back	ACT-215	13/11/2024	Miércoles	2h 00m
Desarrollo Back	ACT-215	17/11/2024	Domingo	3h 00m
Desarrollo Back	ACT-215	18/11/2024	Lunes	2h 00m
Desarrollo Back	ACT-215	18/11/2024	Lunes	2h 00m
Desarrollo Back	ACT-215	19/11/2024	Martes	6h 00m
Desarrollo Back	ACT-215	20/11/2024	Miércoles	1h 00m
Desarrollo Back	ACT-215	25/11/2024	Lunes	6h 00m
Desarrollo Back	ACT-215	26/11/2024	Martes	4h 00m
Desarrollo Back	ACT-215	28/11/2024	Jueves	1h 00m
<b>Actividad</b>				<b>41h 15m</b>
BBDD	ACT-773	16/04/2024	Martes	0h 50m
BBDD	ACT-773	15/09/2024	Domingo	1h 00m
BBDD	ACT-773	30/09/2024	Lunes	1h 00m
BBDD	ACT-773	16/10/2024	Miércoles	0h 45m
BBDD	ACT-773	06/11/2024	Miércoles	0h 15m
BBDD	ACT-773	19/11/2024	Martes	1h 00m
<b>Actividad</b>				<b>4h 50m</b>
Desarrollo Front	ACT-836	16/10/2024	Miércoles	1h 15m
Desarrollo Front	ACT-836	19/10/2024	Sábado	1h 00m

Desarrollo Front	ACT-836	21/10/2024	Lunes	3h 00m
Desarrollo Front	ACT-836	24/10/2024	Jueves	4h 00m
Desarrollo Front	ACT-836	25/10/2024	Viernes	3h 00m
Desarrollo Front	ACT-836	27/10/2024	Domingo	1h 00m
Desarrollo Front	ACT-836	28/10/2024	Lunes	2h 00m
Desarrollo Front	ACT-836	30/10/2024	Miércoles	0h 30m
Desarrollo Front	ACT-836	31/10/2024	Jueves	2h 00m
Desarrollo Front	ACT-836	05/11/2024	Martes	3h 00m
Desarrollo Front	ACT-836	06/11/2024	Miércoles	0h 30m
Desarrollo Front	ACT-836	08/11/2024	Viernes	1h 00m
Desarrollo Front	ACT-836	17/11/2024	Domingo	5h 00m
Desarrollo Front	ACT-836	18/11/2024	Lunes	2h 00m
Desarrollo Front	ACT-836	18/11/2024	Lunes	3h 00m
Desarrollo Front	ACT-836	19/11/2024	Martes	3h 00m
Desarrollo Front	ACT-836	20/11/2024	Miércoles	2h 00m
Desarrollo Front	ACT-836	25/11/2024	Lunes	6h 00m
Desarrollo Front	ACT-836	26/11/2024	Martes	6h 00m
Desarrollo Front	ACT-836	27/11/2024	Miércoles	11h 00m
Desarrollo Front	ACT-836	28/11/2024	Jueves	11h 00m
Desarrollo Front	ACT-836	02/12/2024	Lunes	2h 00m
<b>Actividad</b>				<b>73h 15m</b>
Diseño	ACT-134	16/12/2023	Sábado	2h 00m
Diseño	ACT-134	22/12/2023	Viernes	0h 30m
Diseño	ACT-134	13/01/2024	Sábado	1h 00m
Diseño	ACT-134	13/06/2024	Jueves	2h 00m
<b>Actividad</b>				<b>5h 30m</b>
Documentación	ACT-27	16/12/2023	Sábado	0h 15m
Documentación	ACT-27	22/07/2024	Lunes	2h 00m
Documentación	ACT-27	24/09/2024	Martes	2h 00m
Documentación	ACT-27	30/09/2024	Lunes	1h 00m
Documentación	ACT-27	16/10/2024	Miércoles	1h 00m
Documentación	ACT-27	19/10/2024	Sábado	2h 00m
Documentación	ACT-27	28/11/2024	Jueves	1h 30m
Documentación	ACT-27	30/11/2024	Sábado	5h 00m
Documentación	ACT-27	03/12/2024	Martes	3h 00m
<b>Actividad</b>				<b>17h 45m</b>
Investigación	ACT-39	18/04/2024	Jueves	2h 00m
Investigación	ACT-39	24/04/2024	Miércoles	1h 15m
Investigación	ACT-39	09/10/2024	Miércoles	1h 00m
Investigación	ACT-39	16/10/2024	Miércoles	0h 15m
Investigación	ACT-39	10/11/2024	Domingo	2h 00m
Investigación	ACT-39	19/11/2024	Martes	1h 00m
Investigación	ACT-39	20/11/2024	Miércoles	0h 30m
<b>Actividad</b>				<b>8h 00m</b>
<b>Horas</b>				<b>150h 35m</b>

# Informe de tiempo registrado

Intervalo de fechas	04 diciembre 2024 - 13 diciembre 2024
# Miembros	1

ACTIVITY	CÓDIGO ACT.	FECHA	DÍA	HRS REGISTRADAS
Desarrollo Front	ACT-836	11/12/2024	Miércoles	2h 00m
<b>Actividad</b>				<b>2h 00m</b>
Documentación	ACT-27	04/12/2024	Miércoles	1h 00m
Documentación	ACT-27	05/12/2024	Jueves	1h 00m
Documentación	ACT-27	10/12/2024	Martes	0h 30m
Documentación	ACT-27	11/12/2024	Miércoles	1h 00m
<b>Actividad</b>				<b>3h 30m</b>
<b>Horas</b>				<b>5h 30m</b>

#### IV. APÉNDICE IV: GLOSARIO DE TÉRMINOS

**Núcleo familiar:** matrimonios o parejas que conviven juntos, con o sin hijos. También personas solteras que conviven.

**Material Design:** lenguaje de diseño basado en objetos materiales colocados en espacios y tiempos determinados. Se desarrollan cuadrículas, animaciones, transiciones receptivas, rellenos, profundidades y otros tipos de efectos espaciales y de iluminación.