

Pau Piñol Moreno

**DESARROLLO DE UNA APLICACIÓN MÓVIL: CATÁLOGO DIGITAL
INTELIGENTE CON FLUTTER**

TRABAJO DE FIN DE GRADO

dirigido por Marc Sanchez Artigas

Grado de Ingeniería Informática



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

Resumen.

Este Trabajo de Fin de Grado consiste en el desarrollo de una aplicación que sirva como un catálogo de productos con funcionalidades avanzadas para diferentes tipos de usuarios. El desarrollo de esta aplicación se enmarca dentro de mis prácticas curriculares en la empresa TecnoL.

TecnoL ha identificado la necesidad de mejorar la comunicación con sus clientes y facilitar la gestión de los comerciales. Para abordar esta necesidad se ha desarrollado una aplicación móvil para mantener a los usuarios informados sobre los productos y noticias, facilitar el acceso a estos, y simplificar a los comerciales el seguimiento de pedidos y la distribución de información técnica a los clientes.

La metodología de desarrollo ha consistido en el análisis de requisitos, diseño, implementación y evaluación. El front end del proyecto, consistente en una aplicación móvil para Android y iOS, ha sido realizado con Flutter y el back end con Laravel.

La realización de este trabajo me ha permitido aplicar y afianzar los conocimientos obtenidos en el grado a la vez que me ha preparado para trabajar en un entorno empresarial.

Resum.

Aquest Treball de Fi de Grau consisteix en el desenvolupament d'una aplicació que serveixi com a catàleg de productes amb funcionalitats avançades per a diferents tipus d'usuaris. El desenvolupament d'aquesta aplicació s'emmarca dins les meves pràctiques curriculars a l'empresa TecnoL.

TecnoL ha identificat la necessitat de millorar la comunicació amb els seus clients i facilitar la gestió dels comercials. Per abordar aquesta necessitat s'ha desenvolupat una aplicació mòbil per mantenir els usuaris informats sobre els productes i notícies, facilitar-ne l'accés, i simplificar als comercials el seguiment de comandes i la distribució d'informació tècnica als clients.

La metodologia de desenvolupament ha consistit en l'anàlisi de requisits, disseny, implementació i avaluació. El front end del projecte, consistent en una aplicació mòbil per a Android i iOS, ha estat realitzat amb Flutter i el back end amb Laravel.

La realització d'aquest treball m'ha permès aplicar i consolidar els coneixements obtinguts al grau alhora que m'ha preparat per treballar en un entorn empresarial.

Abstract.

This Bachelor's Degree Final Project consists of the development of an application that serves as a catalogue of products with advanced functionalities for different types of users. The development of this application is part of my curricular internship in the company TecnoL.

Tecnol has identified the need to improve communication with its customers and facilitate the management to salespersons. To address this need, a mobile application has been developed to keep users informed about products and news, to facilitate access to them, and to simplify the tracking of orders and the distribution of technical information to customers.

The development methodology consisted of requirements analysis, design, implementation and evaluation. The front end of the project, consisting of a mobile application for Android and iOS, has been done with Flutter and the back end with Laravel.

The completion of this work has allowed me to apply and consolidate the knowledge obtained in the degree while it has prepared me to work in a business environment.

Índice

1	INTRODUCCIÓN.....	7
1.1	CONTEXTO.....	7
1.2	OBJETIVOS.....	7
2	DESCRIPCIÓN GENERAL.....	9
2.1	IDENTIFICACIÓN DE NECESIDADES.....	9
2.2	PREVISIONES DE USO.....	9
3	REQUISITOS.....	11
3.1	GUIONES.....	11
3.1.1	<i>Guion del usuario invitado</i>	11
3.1.2	<i>Guion del usuario externo</i>	11
3.1.3	<i>Guion del usuario comercial</i>	11
3.1.4	<i>Guion del usuario administrador</i>	12
3.2	REQUISITOS FUNCIONALES.....	12
3.2.1	<i>Listado de requisitos funcionales</i>	12
3.2.2	<i>Diagrama de casos de uso</i>	14
3.2.3	<i>Especificación textual de los casos de uso</i>	14
3.3	REQUISITOS NO FUNCIONALES.....	25
3.3.1	<i>RNF1. Usabilidad</i>	25
3.3.2	<i>RNF2. Rendimiento</i>	26
3.3.3	<i>RNF3. Seguridad</i>	26
3.3.4	<i>RNF4. Disponibilidad y Fiabilidad</i>	26
3.3.5	<i>RNF5. Mantenibilidad</i>	27
4	ANÁLISIS DE LOS REQUISITOS FUNCIONALES.....	28
4.1	DIAGRAMA DE CLASES.....	28
4.2	DIAGRAMAS DE SECUENCIAS DE LOS CASOS DE USO.....	29
5	DISEÑO.....	43
5.1	TECNOLOGÍAS A UTILIZAR.....	43
5.1.1	<i>Front end</i>	43
5.1.2	<i>Back end</i>	43
5.2	ARQUITECTURA DE LA APLICACIÓN.....	44
5.3	DISEÑO DE LA PERSISTENCIA DE DATOS.....	46
5.4	DISEÑO DE API.....	47
5.5	PROTOTIPOS DE LA INTERFAZ GRÁFICA.....	53
5.6	DISEÑO FINAL DE LA INTERFAZ GRÁFICA.....	61
5.6.1	<i>Pantalla de bienvenida</i>	61
5.6.2	<i>Pantalla de inicio</i>	61
5.6.3	<i>Barra de búsqueda de productos</i>	62
5.6.4	<i>Pantalla de categorías</i>	64
5.6.5	<i>Pantalla de detalles del producto</i>	66
5.6.6	<i>Pantalla de favoritos</i>	68
5.6.7	<i>Pantalla de notificaciones</i>	69
5.6.8	<i>Pantalla de cuenta</i>	70
5.6.9	<i>Pantalla de pedidos</i>	75
5.6.10	<i>Pantalla de listas</i>	75
5.6.11	<i>Pantallas de inicio de sesión y registro</i>	80
6	IMPLEMENTACIÓN.....	81
6.1	DETALLES DE LAS TECNOLOGÍAS UTILIZADAS.....	81
6.1.1	<i>Front end</i>	81
6.1.2	<i>Back end</i>	82
6.2	ESTRUCTURA DE DIRECTORIOS Y FICHEROS DE LA APLICACIÓN.....	82

6.2.1	<i>Directorio assets</i>	82
6.2.2	<i>Directorio lib</i>	83
6.3	ALGORITMOS ESPECÍFICOS MÁS DESTACADOS	84
6.3.1	<i>Gestión de favoritos</i>	85
7	AVALUACIÓN	90
8	CONCLUSIONES	96
9	REFERENCIAS	97

Índice de tablas

TABLA 1. PRUEBAS REALIZADAS	95
-----------------------------------	----

Índice de figuras

FIGURA 1. DIAGRAMA DE CASOS DE USO	14
FIGURA 2. DIAGRAMA DE CLASES	28
FIGURA 3. SECUENCIA DEL CASO DE USO 00. SIGNUP	29
FIGURA 4. SECUENCIA DEL CASO DE USO 01. LOGIN	29
FIGURA 5. SECUENCIA DEL CASO DE USO 02. LOGOUT	30
FIGURA 6. SECUENCIA DEL CASO DE USO 03. MOSTRAR CATEGORIAS.....	31
FIGURA 7. SECUENCIA DEL CASO DE USO 04. MOSTRAR LISTA PRODUCTOS.....	32
FIGURA 8. SECUENCIA DEL CASO DE USO 05. MOSTRAR GESTIONAR PRODUCTO.....	33
FIGURA 9. SECUENCIA DEL CASO DE USO 06. BUSCAR PRODUCTO.....	34
FIGURA 10. SECUENCIA DEL CASO DE USO 07. ESCANEAR PRODUCTO.....	35
FIGURA 11. SECUENCIA DEL CASO DE USO 08. MOSTRAR FAVORITOS	35
FIGURA 12. SECUENCIA DEL CASO DE USO 09. MOSTRAR NOTIFICACIONES	36
FIGURA 13. SECUENCIA DEL CASO DE USO 10. CONFIGURAR NOTIFICACIONES.....	37
FIGURA 14. SECUENCIA DEL CASO DE USO 11. MOSTRAR CONTACTO	37
FIGURA 15. SECUENCIA DEL CASO DE USO 12. ENVIAR FORMULARIO CONTACTO	38
FIGURA 16. SECUENCIA DEL CASO DE USO 13. MOSTRAR NOTICIAS	38
FIGURA 17. SECUENCIA DEL CASO DE USO 14. MOSTRAR ENLACES INTERES.....	39
FIGURA 18. SECUENCIA DEL CASO DE USO 15. MOSTRAR PEDIDOS.....	39
FIGURA 19. SECUENCIA DEL CASO DE USO 16. GESTIONAR LISTAS COMERCIAL	40
FIGURA 20. SECUENCIA DEL CASO DE USO 17. GESTIONAR LISTA COMERCIAL PARTICULAR	41
FIGURA 21. SECUENCIA DEL CASO DE USO 18. EDITAR PERFIL.....	42
FIGURA 22. SECUENCIA DEL CASO DE USO 19. CAMBIAR ROL.....	42
FIGURA 23. DIAGRAMA ARQUITECTURA MVC.....	45
FIGURA 24. DIAGRAMA ENTIDAD RELACIÓN DE LA BASE DE DATOS.....	46
FIGURA 25. PROTOTIPO DE PANTALLA DE BIENVENIDA.....	53
FIGURA 26. PROTOTIPOS DE PANTALLAS DE INICIO DE SESIÓN Y REGISTRO	54
FIGURA 27. PROTOTIPOS DE PANTALLAS DE INICIO Y PRODUCTOS ALEATORIOS	54
FIGURA 28. PROTOTIPOS DE PANTALLAS DE DIVISIONES Y SUBCATEGORÍAS.....	55
FIGURA 29. PROTOTIPOS DE PANTALLAS DE LISTA DE PRODUCTOS Y DETALLES DE PRODUCTO.....	55
FIGURA 30. PROTOTIPOS DE PANTALLAS DE BUSCAR PRODUCTO Y RESULTADOS DE BÚSQUEDA	56
FIGURA 31. PROTOTIPO DE PANTALLA DE FAVORITOS.....	56
FIGURA 32. PROTOTIPOS DE PANTALLAS DE NOTIFICACIONES CONTRAÍDAS, EXPANDIDAS Y CONFIGURACIÓN.....	57
FIGURA 33. PROTOTIPOS DE PANTALLAS DE CUENTA DE USUARIO INVITADO, EXTERNO, COMERCIAL Y ADMINISTRADOR.....	57
FIGURA 34. PROTOTIPOS DE PANTALLAS DE CONTACTO, FORMULARIO DE CONTACTO Y SUSCRIBIRSE A NEWSLETTER	58
FIGURA 35. PROTOTIPOS DE PANTALLAS DE DIVISIONES DE NOTICIAS, LISTA DE NOTICIAS Y WEB DE UNA NOTICIA.....	58
FIGURA 36. PROTOTIPOS DE PANTALLAS DE ENLACES DE INTERÉS GENERAL Y DEL USUARIO COMERCIAL.....	59
FIGURA 37. PROTOTIPOS DE PANTALLAS DE LISTAS, LISTA DE PRODUCTOS Y DE CLIENTES DEL COMERCIAL.....	59
FIGURA 38. PROTOTIPOS DE PANTALLAS DE PEDIDOS CONTRAÍDOS Y EXPANDIDOS	60
FIGURA 39. PROTOTIPO DE PANTALLA DE ACTUALIZAR PERFIL.....	60
FIGURA 40. PANTALLA DE BIENVENIDA	61
FIGURA 41. PANTALLA DE INICIO	62
FIGURA 42. PANTALLA DE PRODUCTOS ALEATORIOS.....	62
FIGURA 43. HISTORIAL Y RESULTADOS DE BÚSQUEDA	63
FIGURA 44. PANTALLA CON LISTA DE PRODUCTOS DE UNA BÚSQUEDA	63
FIGURA 45. PANTALLA DE ESCÁNER DE CÓDIGOS	64
FIGURA 46. PANTALLA DE DIVISIONES.....	64
FIGURA 47. PANTALLA DE SUBCATEGORÍAS.....	65
FIGURA 48. PANTALLA DE LISTA DE PRODUCTOS Y SU DESPLEGABLE	65
FIGURA 49. PANTALLA DE DETALLES DEL PRODUCTO DEL INVITADO/EXTERNO Y COMERCIAL	66
FIGURA 50. FICHA TÉCNICA.....	67
FIGURA 51. PANTALLA PARA AÑADIR PRODUCTO A LISTA	67
FIGURA 52. PANTALLA PARA SUBIR IMAGEN DE PRODUCTO	68
FIGURA 53. PANTALLA DE FAVORITOS.....	68
FIGURA 54. PANTALLA DE NOTIFICACIONES	69
FIGURA 55. MENSAJE DE NOTIFICACIÓN ELIMINADA	69
FIGURA 56. PANTALLA DE CONFIGURACIÓN DE NOTIFICACIONES DE USUARIO EXTERNO Y COMERCIAL	70
FIGURA 57. PANTALLA DE CUENTA DE INVITADO, EXTERNO, COMERCIAL Y ADMINISTRADOR.....	70

FIGURA 58. PANTALLA DE EDITAR PERFIL.....	71
FIGURA 59. PANTALLAS DE ENLACES DE INTERÉS Y WEB	72
FIGURA 60. PANTALLA DE ENLACES DE INTERÉS DEL COMERCIAL.....	72
FIGURA 61. PANTALLA DE CONTACTO.....	73
FIGURA 62. PANTALLAS DE FORMULARIO DE CONTACTO Y SUSCRIBIRSE A LA NEWSLETTER.....	73
FIGURA 63. ERROR CAMPOS INCOMPLETOS EN FORMULARIO DE CONTACTO Y NEWSLETTER.....	74
FIGURA 64. PANTALLA DE NOTICIAS Y BLOGS, LISTA DE NOTICIAS Y WEB DE UNA NOTICIA.....	74
FIGURA 65. PANTALLA DE PEDIDOS.....	75
FIGURA 66. PANTALLA DE LISTAS DE PRODUCTOS	76
FIGURA 67. DIÁLOGOS PARA CREAR, EDITAR Y ELIMINAR UNA LISTA	76
FIGURA 68. PRODUCTOS EN UNA LISTA	77
FIGURA 69. LISTAS DE CLIENTES	77
FIGURA 70. PANTALLA DE CLIENTES DE UNA LISTA	78
FIGURA 71. DIÁLOGO PARA BUSCAR Y AÑADIR UN CLIENTE	78
FIGURA 72. SELECCIONAR APP DE CORREO.....	79
FIGURA 73. NUEVO CORREO.....	79
FIGURA 74. PANTALLA DE INICIO DE SESIÓN	80
FIGURA 75. PANTALLA DE REGISTRO	80

Índice de códigos

CÓDIGO 1. CONSTRUCTOR DE PRODUCTPROVIDER.....	85
CÓDIGO 2. MÉTODO PARA OBTENER LOS PRODUCTOS FAVORITOS EN PRODUCTPROVIDER.....	85
CÓDIGO 3. MÉTODO PARA HACER LA PETICIÓN DE OBTENER LOS PRODUCTOS FAVORITOS EN PRODUCTSERVICE	86
CÓDIGO 4. MÉTODO PARA AÑADIR O QUITAR UN PRODUCTO DE FAVORITOS EN PRODUCTPROVIDER	87
CÓDIGO 5. MÉTODO PARA HACER LA PETICIÓN DE AÑADIR UN PRODUCTO A FAVORITOS EN PRODUCTSERVICE	87
CÓDIGO 6. MÉTODO PARA HACER LA PETICIÓN DE QUITAR UN PRODUCTO A FAVORITOS EN PRODUCTSERVICE.....	88
CÓDIGO 7. MÉTODO PARA CONSULTAR SI UN PRODUCTO ES FAVORITO EN PRODUCTPROVIDER.....	88
CÓDIGO 8. CONSUMER DE AUTHPROVIDER Y PRODUCTPROVIDER EN LA PANTALLA DE FAVORITOS.....	88
CÓDIGO 9. GESTIÓN DEL BOTÓN DE FAVORITO	89

1 Introducción

Este Trabajo de Fin de Grado consiste en el diseño e implementación de una aplicación móvil multiplataforma que sirva como un catálogo de productos con funcionalidades avanzadas para diferentes tipos de usuarios. El desarrollo de esta aplicación se enmarca dentro de mis prácticas curriculares en la empresa TecnoI, los cuales me propusieron este proyecto.

El desarrollo del proyecto constará, por una parte, del front end consistente en una aplicación móvil para Android y iOS¹ realizada con Flutter, un *framework* de código abierto desarrollado por Google que permite la creación de aplicaciones en múltiples plataformas a partir de una única base de código.

Por otra parte, el back end del proyecto será realizado con Laravel, un *framework* de desarrollo de aplicaciones y servicios web de código abierto basado en PHP².

1.1 Contexto

TecnoI es una empresa fundada en 1997 en Reus. Actualmente, tiene presencia en los sectores: sanitario, equipamientos urbanos, construcción y adhesivos cianocrilatos. Consta de cuatro divisiones principales [1]:

- TecnoI Techincs: se dedica a la fabricación de materiales y productos técnicos profesionales para la construcción y la industria.
- TecnoI Urban: ofrece soluciones de equipamientos urbanos eficientes y sostenibles como iluminación solar y material urbano reciclado.
- TecnoI Health: fabrica consumibles en el sector sanitario.
- TecnoI Fix: se dedica a la producción y envasado de adhesivos de cianocrilato en distintos formatos para la industria, el bricolaje y el *canal retail*.

En un entorno empresarial cada vez más digitalizado, TecnoI ha identificado la necesidad de mejorar la comunicación con sus clientes y facilitar la gestión de los comerciales. Para abordar esta necesidad se ha decidido desarrollar una aplicación móvil para mantener a los usuarios informados sobre los productos y noticias, facilitar el acceso a estos, y simplificar a los comerciales el seguimiento de pedidos y la distribución de información técnica a los clientes, todo ello proporcionando una experiencia de usuario intuitiva.

1.2 Objetivos

Desde el punto de vista formativo, los objetivos de este proyecto son los siguientes.

¹ Sistema operativo móvil desarrollado por Apple

² *Hypertext Preprocessor*, lenguaje de programación interpretado especialmente adecuado para el desarrollo web

- En primer lugar, aplicar los conocimientos prácticos y teóricos adquiridos durante el Grado de Ingeniería Informática en áreas como el desarrollo de aplicaciones móviles, programación back end, bases de datos, y gestión de proyectos.
- En segundo lugar, complementar los conocimientos del grado con nuevas técnicas que se adquirirán durante el desarrollo del proyecto. Estas nuevas habilidades técnicas permitirán la creación de una aplicación móvil desde cero, proporcionando una comprensión profunda del ciclo de desarrollo de software.
- En tercer lugar, se espera adquirir experiencia y entender las dinámicas de un entorno empresarial real con tal de hacer la transición del ámbito académico al profesional. Además, trabajar en colaboración con los equipos de la empresa permitirá mejorar las habilidades de comunicación y trabajo en equipo.

Los objetivos principales del proyecto se resumen a continuación.

- Se desarrollará una aplicación móvil para la empresa Tecnol que servirá como catálogo de productos a la vez que ofrecerá distintas funcionalidades en función del usuario.
- Todos los usuarios, es decir, invitados, externos, comerciales y administradores, podrán acceder a la información detallada de los productos, buscar por palabras clave o escaneando su código de barras, y acceder a noticias de la empresa e información de contacto.
- Los usuarios registrados, es decir, externos, comerciales y administradores, podrán guardar productos en favoritos y recibir notificaciones de nuevas noticias.
- Los comerciales podrán visualizar información resumida de sus pedidos y recibir notificaciones cuando cambien de estado.
- Finalmente, los administradores podrán cambiar de rol de usuario para acceder a las distintas funciones.

2 Descripción General

2.1 Identificación de Necesidades

En este apartado se busca comprender y definir las necesidades que deben resolverse con la realización del proyecto.

- Comunicación y notificaciones. Mantener a los usuarios informados de los productos y noticias relevantes de la empresa es esencial.
- Gestión de pedidos. Es crucial para los comerciales facilitarles el seguimiento y la gestión de los pedidos, incluidas las actualizaciones de estado.
- Eficiencia en la distribución de información a clientes. Es necesario ofrecer herramientas para que los comerciales organicen y compartan fichas técnicas con clientes de manera eficiente.
- Mejorar la experiencia de usuario. Es importante proporcionar una aplicación móvil con una interfaz intuitiva y fácil de usar para todos los tipos de usuarios, la cual facilite el acceso y navegación a productos, categorías, noticias, pedidos, notificaciones y enlaces de interés de la empresa.

2.2 Previsiones de Uso

En este apartado, se analizarán los usuarios a los que va dirigida la aplicación y los principales casos de uso a los que tienen acceso.

- Invitado: es una persona interesada en los productos de la empresa y sus categorías. Accede a la aplicación sin iniciar sesión y, por tanto, tiene acceso limitado a ciertas funciones y características. Podrá visualizar la información detallada de un producto, ver sus categorías, buscarlos por palabras clave o escaneando su código de barras, visualizar las noticias de la empresa y acceder a información de contacto y enlaces de interés.
- Externo: persona que no trabaja en la empresa, pero que se ha registrado para tener acceso a funciones que el invitado no tiene. Además de las características del invitado, podrá guardar productos en favoritos y recibir notificaciones de noticias de la empresa.
- Comercial: es aquella persona que trabaja en la empresa como comercial y ha iniciado sesión. Aparte de las funciones del usuario registrado externo, podrá añadir fichas técnicas de productos a un apartado de la aplicación que le permitirá gestionarlas para enviarlas a los clientes que desee. También podrá visualizar sus pedidos y recibir notificaciones cuando cambien de estado.

- Administrador: persona que trabaja en la empresa, ha iniciado sesión, y ha de tener acceso con privilegios a la aplicación con el propósito principal de supervisar, gestionar y controlar diversos aspectos. Podrá cambiar de rol de usuario para mostrar la aplicación y acceder a las funciones de esta según el rol seleccionado.

3 Requisitos

En este apartado se documentarán los requisitos funcionales y no funcionales.

3.1 Guiones

3.1.1 Guion del usuario invitado

El usuario invitado podrá visualizar la siguiente información y utilizar las funciones descritas a continuación.

En la pestaña de inicio podrá acceder a los productos destacados, más vendidos, nuevos y en promoción. También tendrá acceso a una pantalla con productos aleatorios para poder explorarlos sin tener que buscar por categorías concretas. En la parte superior podrá acceder al buscador de productos y al escáner de códigos de barras.

En la pestaña de categorías visualizará las divisiones de la empresa y podrá acceder a sus subcategorías hasta llegar a la lista de productos. Si accede a un producto en concreto podrá visualizar su información detallada, mostrar la ficha técnica y abrir la app de Tecnol. En caso de que le dé al botón de favoritos se mostrará la pantalla de inicio de sesión.

En la pestaña de favoritos le aparecerá un botón para navegar a la pantalla de inicio de sesión.

En la pestaña de notificaciones le aparecerá un botón para navegar a la pantalla de inicio de sesión.

En la pestaña de cuenta le aparecerá un botón para navegar a la pantalla de inicio de sesión. Podrá acceder a un apartado de enlaces de interés donde se le mostrará un conjunto de enlaces de interés a diversas webs y apps de utilidad para el usuario. También tendrá acceso a un apartado de contacto con diferentes opciones de contacto, un formulario y otra opción para suscribirse a la newsletter. Finalmente, habrá un apartado de noticias y blogs para poder visualizar las noticias de Tecnol o los blogs de Technics, Urban, Health y Fix.

3.1.2 Guion del usuario externo

El usuario externo aparte de poder acceder a la información y funcionalidades del invitado podrá hacer lo siguiente.

En la pestaña de favoritos podrá visualizar y gestionar sus productos favoritos. Cuando entre en la pantalla de detalles de un producto podrá añadirlo a favoritos.

En la pestaña de notificaciones podrá consultar y gestionar las notificaciones de noticias.

En la pestaña de cuenta podrá cerrar sesión y editar su perfil.

3.1.3 Guion del usuario comercial

El usuario comercial aparte de tener acceso a la información y funcionalidades del externo podrá hacer lo descrito a continuación.

En la pantalla de detalles del producto también tendrá las opciones de añadir el producto a una lista de comercial y subir una imagen del producto.

En la pestaña de notificaciones podrá consultar y gestionar las notificaciones de pedidos.

En la pestaña de cuenta podrá acceder a un apartado para visualizar sus pedidos. También habrá una sección para enviar fichas técnicas de productos seleccionados a múltiples clientes a través del correo.

3.1.4 Guion del usuario administrador

El usuario administrador podrá cambiar de rol de usuario para visualizar la aplicación y acceder a las funciones de esta según el rol seleccionado.

3.2 Requisitos Funcionales

3.2.1 Listado de requisitos funcionales

- RF1. El sistema debe permitir el registro de nuevos usuarios externos solicitando un correo electrónico y una contraseña.
- RF2. El sistema debe permitir a los usuarios externos, comerciales y administradores iniciar sesión utilizando su correo electrónico y contraseña.
- RF3. El sistema debe permitir a los usuarios registrados (externos, comerciales y administradores) cerrar sesión.
- RF4. El sistema debe mostrar las categorías y subcategorías de productos.
- RF5. El sistema debe mostrar una lista de productos basada en la categoría seleccionada por el usuario.
- RF6. El sistema debe permitir a los usuarios ver los detalles de un producto, incluyendo nombre, descripción y otras especificaciones.
- RF7. El sistema debe permitir a los usuarios buscar productos mediante palabras clave o escaneando un código de barras.
- RF8. El sistema debe mostrar productos destacados, nuevos, en promoción, top ventas de la empresa y productos aleatorios con imágenes reales en la pantalla de inicio.
- RF9. El sistema debe permitir a los usuarios registrados (externos y comerciales) añadir y gestionar productos en una lista de favoritos.
- RF10. El sistema debe mostrar notificaciones a los usuarios registrados (externos y comerciales) sobre noticias y entradas en los blogs de Tecnol.
- RF11. El sistema debe mostrar notificaciones a los usuarios comerciales sobre el cambio de estado de sus pedidos.
- RF12. El sistema debe permitir a los usuarios comerciales visualizar el historial de sus pedidos y la información resumida de cada pedido.
- RF13. El sistema debe mostrar diferentes opciones de contacto, incluyendo teléfono, WhatsApp, correo electrónico y dirección de oficinas, entre otros.
- RF14. El sistema debe permitir a los usuarios enviar comentarios o consultas a través de un formulario de contacto.

- RF15. El sistema debe mostrar una lista de noticias de Tecnol y entradas de los blogs de Technics, Urban, Health y Fix.
- RF16. El sistema debe mostrar una lista de enlaces a webs, redes sociales y aplicaciones de interés relacionadas con Tecnol.
- RF17. El sistema debe permitir a los usuarios externos editar el nombre y apellidos de su perfil.
- RF18. El sistema debe permitir a los administradores cambiar de rol de usuario para mostrar la aplicación y acceder a las funciones según el rol seleccionado.
- RF19. El sistema debe permitir a los usuarios comerciales gestionar listas de productos y clientes, incluyendo la creación, edición y eliminación de listas.
- RF20. El sistema debe permitir a los usuarios comerciales añadir productos a una lista y enviar fichas técnicas de productos a múltiples clientes a través del correo.
- RF21. El sistema debe permitir a los usuarios registrados configurar sus preferencias de notificación, incluyendo noticias, blogs y estados de pedidos (para comerciales).

3.2.2 Diagrama de casos de uso

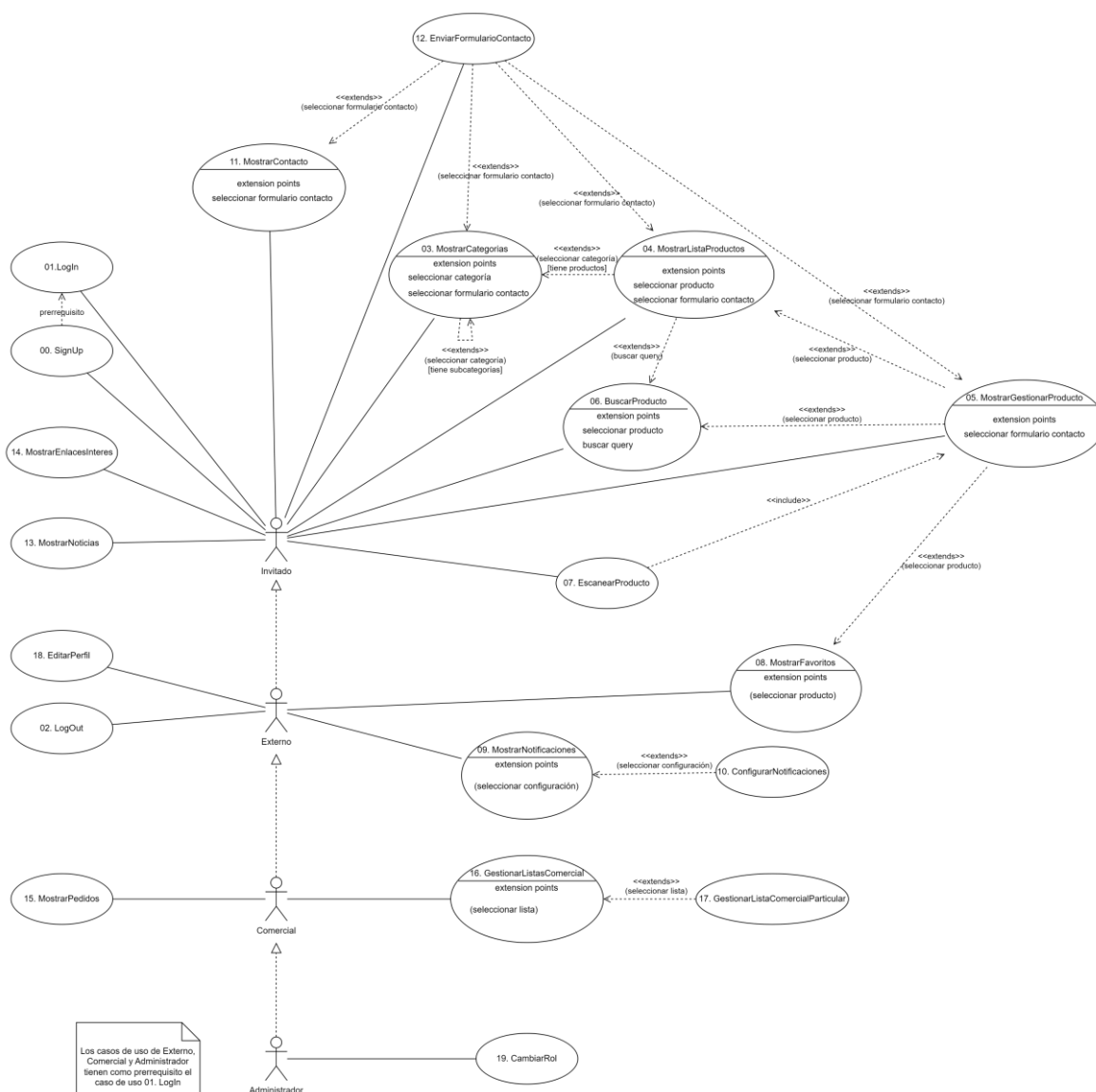


Figura 1. Diagrama de casos de uso

3.2.3 Especificación textual de los casos de uso

00. SignUp

Resumen de la funcionalidad: registrar un nuevo usuario Externo y dar acceso a sus funcionalidades.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado.

Precondición: ninguna.

Postcondición: se ha registrado un nuevo usuario Externo y tiene acceso a las funciones pertinentes.

Proceso normal principal:

1. El sistema pide el correo y la contraseña al usuario.
2. El Invitado introduce los datos.
3. El sistema registra el nuevo usuario en el back end y da acceso a las funciones permitidas al usuario Externo.

Alternativas de proceso y excepciones: ninguna.

01. LogIn

Resumen de la funcionalidad: identificar al usuario Externo, Comercial o Administrador y dar acceso a las funciones que le corresponden.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado.

Precondición: ninguna.

Postcondición: se ha identificado el usuario y se da acceso a las funciones que le corresponden según su rol.

Proceso normal principal:

1. El sistema pide el correo y la contraseña del usuario.
2. El usuario introduce los datos.
3. El sistema consulta al back end para que compruebe las credenciales.
4. El sistema da acceso a las funciones permitidas del usuario.

Alternativas de proceso y excepciones:

- 4a. El usuario introduce un correo o contraseña incorrectos:
 - 4a1. El sistema detecta que los datos son incorrectos.
 - 4a2. El sistema muestra al usuario un mensaje de error.
 - 4a3. El sistema vuelve al paso 1.

02. LogOut

Resumen de la funcionalidad: finalizar la sesión del usuario y mostrar la aplicación como Invitado.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Externo, Comercial o Administrador.

Precondición: el usuario ha iniciado sesión.

Postcondición: se ha cerrado la sesión del usuario.

Proceso normal principal:

1. El usuario selecciona cerrar sesión.
2. El sistema finaliza la sesión del usuario indicándoselo al back end.
3. El sistema muestra la aplicación como Invitado.

Alternativas de proceso y excepciones: ninguna.

03. MostrarCategorías

Resumen de la funcionalidad: mostrar las categorías/subcategorías de productos.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema lee del back end las categorías/subcategorías.
2. El sistema muestra las categorías/subcategorías.
3. El usuario selecciona una categoría.
4. El sistema consulta al back end si esa categoría tiene subcategorías o productos.
5. En caso de que tenga subcategorías el sistema ejecuta el caso de uso 03. MostrarCategorías.

Alternativas de proceso y excepciones:

3a. El usuario selecciona la opción de enviar formulario de contacto asociado a la categoría:

3a1. El sistema ejecuta el caso de uso 12. EnviarFormularioContacto.

5a. El sistema detecta que la categoría no tiene subcategorías, tiene productos:

5a1. El sistema ejecuta el caso de uso 04. MostrarListaProductos.

04. MostrarListaProductos

Resumen de la funcionalidad: mostrar una lista de productos.

Parámetros de entrada: productos a mostrar.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema muestra una lista con los productos.
2. El usuario selecciona un producto.
3. El sistema ejecuta el caso de uso 05. MostrarGestionarProducto.

Alternativas de proceso y excepciones:

- 2a. El usuario selecciona una opción para ordenar/filtrar la lista:
 - 2a1. El sistema ordena/filtra la lista y vuelve al paso 1.
- 2b. El usuario selecciona la opción de enviar formulario de contacto asociado a la categoría de los productos mostrados:
 - 2b1. El sistema ejecuta el caso de uso 12. EnviarFormularioContacto.

05. MostrarGestionarProducto

Resumen de la funcionalidad: mostrar la información de un producto y dar acceso a sus opciones.

Parámetros de entrada: producto a mostrar.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema presenta la información al usuario.
2. El usuario selecciona la opción de abrir la app de compras de TecnoL.
3. El sistema abre la aplicación de compras de TecnoL.

Alternativas de proceso y excepciones:

- 2a. El usuario selecciona la opción de añadir producto a favoritos:
 - 2a1. En caso de que el usuario haya iniciado sesión, el sistema guarda en la base de datos el producto en los favoritos del usuario y vuelve al paso 1.
 - 2a2. En caso de que el usuario no haya iniciado sesión, el sistema ejecuta el caso de uso 01. LogIn y vuelve al paso 2a1.
- 2b. El Comercial selecciona la opción de subir imagen del producto:
 - 2b1. El sistema muestra una pantalla para hacer una foto o seleccionarla de la galería.
 - 2b2. El comercial sube la/s imagen/es.
 - 2b3. El sistema guarda la/s imagen/es en la base de datos.
 - 2b4. El sistema vuelve al paso 1.
- 2c. El Comercial selecciona la opción de añadir producto a lista de comercial:
 - 2c1. El sistema muestra las listas del Comercial.
 - 2c2. El usuario selecciona una lista.
 - 2c3. El sistema guarda en el back end la petición de añadir producto a lista del Comercial seleccionada.

- 2c4. El sistema vuelve al paso 1.
- 2d. El usuario selecciona la opción de enviar formulario de contacto asociado al producto:
 - 2d1. El sistema ejecuta el caso de uso 12. EnviarFormularioContacto.
- 2e. El usuario selecciona la opción abrir ficha técnica:
 - 2e1. El sistema descarga la ficha técnica y la muestra al usuario.
- 3a. La app de compras de TecnoL no está instalada en el dispositivo:
 - 3a1. El sistema abre la tienda de aplicaciones del dispositivo con la página de la app.

06. BuscarProducto

Resumen de la funcionalidad: buscar un producto a partir de la query introducida por el usuario.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema muestra el teclado y el historial de búsquedas.
2. El usuario introduce la/s palabra/s relacionada/s con el producto a buscar.
3. El sistema consulta el back end y muestra los productos que más se ajustan al texto de consulta.
4. El usuario selecciona un producto.
5. El sistema ejecuta el caso de uso 05. MostrarGestionarProducto.

Alternativas de proceso y excepciones:

- 2a. El usuario selecciona una query del historial:
 - 2a1. El sistema ejecuta el caso de uso 04. MostrarListaProductos.
- 4a. El usuario selecciona buscar query:
 - 4a1. El sistema guarda la query en el historial.
 - 4a2. El sistema ejecuta el caso de uso 04. MostrarListaProductos.

07. EscanearProducto

Resumen de la funcionalidad: buscar un producto a partir de su código de barras.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema muestra un escáner de códigos.
2. El usuario escanea el código.
3. El sistema detecta el código de barras y consulta el back end para encontrar el producto.
4. El sistema ejecuta el caso de uso 05. MostrarGestionarProducto.

Alternativas de proceso y excepciones: ninguna.

08. MostrarFavoritos

Resumen de la funcionalidad: consultar y gestionar los productos añadidos a favoritos por un usuario registrado.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema lee del back end los productos añadidos a favoritos del usuario que haya iniciado sesión.
2. El sistema presenta la lista de productos añadidos a favoritos.
3. El usuario selecciona un producto.
4. El sistema ejecuta el caso de uso 05. MostrarGestionarProducto.

Alternativas de proceso y excepciones:

- 1a. En caso de que el usuario no haya iniciado sesión, el sistema ejecuta el caso de uso 01. LogIn y vuelve al paso 1.
- 3a. El usuario selecciona la opción de eliminar un producto de favoritos:
 - 3a1. El sistema guarda en el back end que el producto se ha eliminado de favoritos.
 - 3a2. El sistema vuelve al paso 1.

09. MostrarNotificaciones

Resumen de la funcionalidad: consultar y gestionar las notificaciones de un usuario registrado.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema lee del back end las notificaciones del usuario que haya iniciado sesión.
2. El sistema presenta la lista de notificaciones.
3. El usuario expande una notificación.
4. El sistema muestra la notificación expandida.
5. El usuario selecciona la opción de abrir.
6. El sistema abre el navegador con la web del pedido o noticia.

Alternativas de proceso y excepciones:

- 1a. En caso de que el usuario no haya iniciado sesión, el sistema ejecuta el caso de uso 01. LogIn y vuelve al paso 1.
- 3a. El usuario selecciona la opción de eliminar una notificación:
 - 3a1. El sistema guarda en el back end que la notificación se ha eliminado.
 - 3a2. El sistema vuelve al paso 2.
- 3b. El usuario selecciona la opción de configurar notificaciones:
 - 3b1. El sistema ejecuta el caso de uso 10. ConfigurarNotificaciones.

10. ConfigurarNotificaciones

Resumen de la funcionalidad: configurar que notificaciones recibe el usuario.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Externo o Comercial

Precondición: el usuario ha iniciado sesión.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema muestra las opciones de activar/desactivar las notificaciones de noticias de Tecnol y entradas en los blogs de Technics, Urban, Health y Fix.
2. El usuario selecciona la opción de activar/desactivar en alguna de las posibles notificaciones.
3. El sistema guarda las preferencias del usuario.

Alternativas de proceso y excepciones:

- 1a. El usuario que ha iniciado sesión es un Comercial:

- 1a1. El sistema además muestra las opciones de activar/desactivar las notificaciones de los estados de los pedidos.

1a2. El sistema vuelve al paso 1.

11. MostrarContacto

Resumen de la funcionalidad: mostrar las diferentes opciones de contacto.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema muestra todas las opciones de contacto.
2. El usuario selecciona una opción
3. El sistema abre la app del dispositivo correspondiente (correo, Google Maps, teléfono, WhatsApp, etc.)

Alternativas de proceso y excepciones:

- 2a. El usuario selecciona la opción de formulario de contacto:
 - 2a1. El sistema ejecuta el caso de uso 12. EnviarFormularioContacto.
 - 2a2. El sistema vuelve al paso 1.
- 2b. El usuario selecciona la opción de suscribirse a la newsletter:
 - 2b1. El sistema pide el nombre y dirección de correo electrónico al usuario.
 - 2b2. El usuario introduce los datos.
 - 2b3. El sistema registra la suscripción en el back end y vuelve al paso 1.

12. EnviarFormularioContacto

Resumen de la funcionalidad: enviar un comentario por parte del usuario.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema pide el nombre, dirección de correo electrónico y mensaje al usuario.
2. El usuario introduce los datos y selecciona la opción de enviar.
3. El sistema registra el formulario en el back end.

Alternativas de proceso y excepciones: ninguna.

13. MostrarNoticias

Resumen de la funcionalidad: mostrar la lista de noticias de Tecnol o los blogs de Technics, Urban, Health y Fix

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema muestra las diferentes divisiones de noticias.
2. El usuario selecciona una división.
3. El sistema lee de la base de datos las noticias de esa división.
4. El sistema presenta la lista de noticias.
5. El usuario selecciona una noticia.
6. El sistema muestra la web de la noticia.

Alternativas de proceso y excepciones: ninguna.

14. MostrarEnlacesInteres

Resumen de la funcionalidad: mostrar un conjunto de enlaces de interés a diversas webs y apps de utilidad para el usuario

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Invitado, Externo o Comercial

Precondición: ninguna.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema consulta al back end los enlaces de interés del Comercial.
2. El sistema muestra una lista de enlaces de interés que será diferente en función de si el usuario es Comercial o Invitado/Externo.
3. El usuario selecciona un enlace.
4. El sistema abre la web o app del enlace.

Alternativas de proceso y excepciones: ninguna.

15. MostrarPedidos

Resumen de la funcionalidad: mostrar el historial de pedidos del Comercial que ha iniciado sesión.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Comercial

Precondición: el usuario Comercial ha iniciado sesión.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema lee del back end los pedidos del Comercial.
2. El sistema presenta al usuario la lista de pedidos.
3. El Comercial selecciona la opción de expandir un pedido.
4. El sistema presenta el pedido expandido.
5. El Comercial selecciona abrir
6. El sistema abre la web del pedido

Alternativas de proceso y excepciones: ninguna.

16. GestionarListasComercial

Resumen de la funcionalidad: gestionar las listas de productos/clientes del Comercial que ha iniciado sesión.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Comercial

Precondición: el usuario Comercial ha iniciado sesión.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema lee de la base de datos las listas de productos/clientes del Comercial.
2. El sistema presenta al usuario las listas.
3. El usuario selecciona una lista.
4. El sistema ejecuta el caso de uso 17. GestionarListaComercialParticular

Alternativas de proceso y excepciones:

3a. El usuario selecciona la opción de editar una lista:

3a1. El sistema muestra el teclado y el nombre de la lista para editar.

3a2. El usuario introduce el nuevo nombre para la lista.

3a3. El sistema guarda en el back end el nuevo nombre de la lista.

3b. El usuario selecciona la opción de eliminar una lista:

3b1. El sistema realiza la petición de eliminar la lista del back end.

3c. El usuario selecciona la opción de crear una lista:

3c1. El sistema muestra el teclado y pide el nombre de la nueva lista.

3c2. El usuario introduce el nombre para la lista.

3c3. El sistema hace la petición de crear la lista en el back end.

17. . GestionarListaComercialParticular

Resumen de la funcionalidad: gestionar los productos/clientes que tiene en una lista particular el Comercial que ha iniciado sesión.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Comercial

Precondición: el usuario Comercial ha iniciado sesión.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema lee de la base de datos los productos/clientes que tiene guardadas el Comercial en la lista.
2. El sistema presenta una lista de productos/clientes.
3. El Comercial selecciona los productos/clientes que quiere.

Alternativas de proceso y excepciones:

3a. El Comercial deselecta los productos/clientes que quiere.

3b. El Comercial elimina de la lista los productos/clientes que quiere.

3c. El Comercial selecciona la opción de añadir un cliente a la lista:

3c1. El sistema pide el nombre del cliente.

3c2. El Comercial introduce el nombre del cliente.

3c3. El sistema consulta en el back end los clientes con ese nombre y los muestra.

3c4. El Comercial selecciona añadir cliente.

3c5. El sistema realiza en el back end la petición de añadir cliente a la lista.

3d. El Comercial selecciona la opción de enviar productos a clientes:

3d1. El sistema pide que app de correo quiere usar.

3d2. El Comercial selecciona una app de correo.

3d3. El sistema abre la app de correo con los correos de los clientes seleccionados en copia oculta y las fichas técnicas de los productos seleccionados adjuntas.

18. EditarPerfil

Resumen de la funcionalidad: editar la información del perfil del usuario que ha iniciado sesión.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Externo, Comercial o Administrador

Precondición: el usuario ha iniciado sesión.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema muestra un campo para editar el nombre.
2. El usuario introduce el nuevo nombre.
3. El sistema guarda el nuevo nombre en el back end.

Alternativas de proceso y excepciones: ninguna.

19. CambiarRol

Resumen de la funcionalidad: permitir cambiar de rol de usuario al Administrador para mostrar la aplicación y dar acceso a las funciones de esta según el rol seleccionado.

Parámetros de entrada: ninguno.

Parámetros de salida: ninguno.

Usuarios: Administrador

Precondición: el usuario Administrador ha iniciado sesión.

Postcondición: ninguna.

Proceso normal principal:

1. El sistema muestra la lista de roles de usuario (Invitado, Externo y Comercial).
2. El Administrador selecciona un rol.
3. El sistema muestra la aplicación y da acceso a las funciones de esta según el rol seleccionado.

Alternativas de proceso y excepciones: ninguna.

3.3 Requisitos no Funcionales

3.3.1 RNF1. Usabilidad

El requisito de usabilidad para la aplicación se debe centrar en asegurar que tanto los nuevos usuarios como los experimentados puedan interactuar de manera intuitiva y eficiente con todas las funcionalidades. La interfaz debe diseñarse de manera que sea fácil de aprender y utilizar, minimizando la curva de aprendizaje.

Otra parte esencial del requisito de usabilidad es mantener la consistencia en toda la aplicación y una navegación lógica. Esto implica utilizar patrones de diseño, navegación y estilos visuales uniformes en todas las pantallas y componentes.

También, la aplicación debe proporcionar una retroalimentación clara e inmediata a las acciones del usuario. Esto incluye indicadores visuales de carga, confirmaciones de acción y mensajes de error informativos. La retroalimentación adecuada ayuda a mantener a los usuarios informados sobre el progreso de sus acciones y les permite corregir rápidamente en caso de errores.

3.3.2 RNF2. Rendimiento

El requisito de rendimiento para la aplicación móvil es garantizar que la aplicación sea rápida, eficiente y receptiva en todas las condiciones de uso. Esto implica optimizar el tiempo de carga de la aplicación y minimizar los retrasos en la respuesta a las interacciones del usuario.

La aplicación debe ser capaz de manejar múltiples usuarios simultáneamente sin degradación significativa en el rendimiento, asegurando una experiencia fluida incluso durante picos de uso. Por tanto, tanto el back end como la aplicación deben ser escalables.

Además, se debe garantizar un consumo eficiente de recursos, como la memoria y la capacidad de procesamiento del dispositivo. La aplicación debe estar diseñada para utilizar recursos de manera óptima y evitar consumir excesivamente la batería del dispositivo, lo que contribuye a una mejor experiencia de usuario y ahorro de energía.

3.3.3 RNF3. Seguridad

La seguridad para la aplicación es fundamental y tiene que implementar prácticas robustas en todos los niveles. De esta forma se debe usar un protocolo seguro de transferencia de datos como HTTPS³ para que las comunicaciones entre la aplicación y el back end sean encriptadas, asegurando la integridad y confidencialidad de los datos transmitidos y protegiendo contra posibles ataques de intermediarios.

Además, la aplicación tiene que implementar un sistema de autenticación segura de usuarios para poder gestionar y restringir correctamente el acceso a funcionalidades e información a las que cada usuario tiene derecho según su rol. En la misma línea, hay que gestionar de manera efectiva las sesiones activas mediante API⁴ tokens para asegurar el acceso adecuado y prevenir el acceso no autorizado.

Por otra parte, se deben establecer las medidas necesarias para garantizar la privacidad y seguridad de la información del usuario cumpliendo con Reglamento General de Protección de Datos (GDPR⁵).

Finalmente, es esencial usar las mejores prácticas y estándares de seguridad con tal de evitar posibles vulnerabilidades.

3.3.4 RNF4. Disponibilidad y Fiabilidad

La aplicación móvil debe garantizar que esté disponible y funcione de manera confiable durante períodos extendidos de tiempo. Tiene que ser capaz de manejar cargas de usuarios variables y picos de tráfico sin comprometer su disponibilidad. Es decir, tiene que

³ Protocolo de transferencia de hipertexto seguro

⁴ *Application Programming Interface*, pieza de código que permite a diferentes aplicaciones comunicarse entre sí y compartir información y funcionalidades

⁵ El Reglamento General de Protección de Datos es una ley europea relativa a la protección de las personas físicas en lo que respecta al tratamiento de sus datos personales

responder de manera fluida y sin cortes inesperados con tal de cumplir también con los requisitos de usabilidad y rendimiento.

Se tiene que diseñar la aplicación con un alto grado de estabilidad y robustez, minimizando la posibilidad de errores o bloqueos inesperados. Se deben implementar estrategias de manejo de errores y recuperación para garantizar que la aplicación pueda manejar situaciones inesperadas de manera fluida y amigable para el usuario.

3.3.5 RNF5. Mantenibilidad

El requisito de mantenibilidad implica diseñarla de manera que sea fácil de mantener y mejorar a lo largo del tiempo. Esto incluye escribir un código limpio y bien estructurado, utilizando buenas prácticas de programación y siguiendo estándares de diseño que faciliten la comprensión y modificación del software por parte de los desarrolladores. Para conseguir esto también se debe documentar adecuadamente la implementación con tal que otros desarrolladores puedan realizar cambios de manera eficiente.

4 Análisis de los Requisitos Funcionales

4.1 Diagrama de Clases

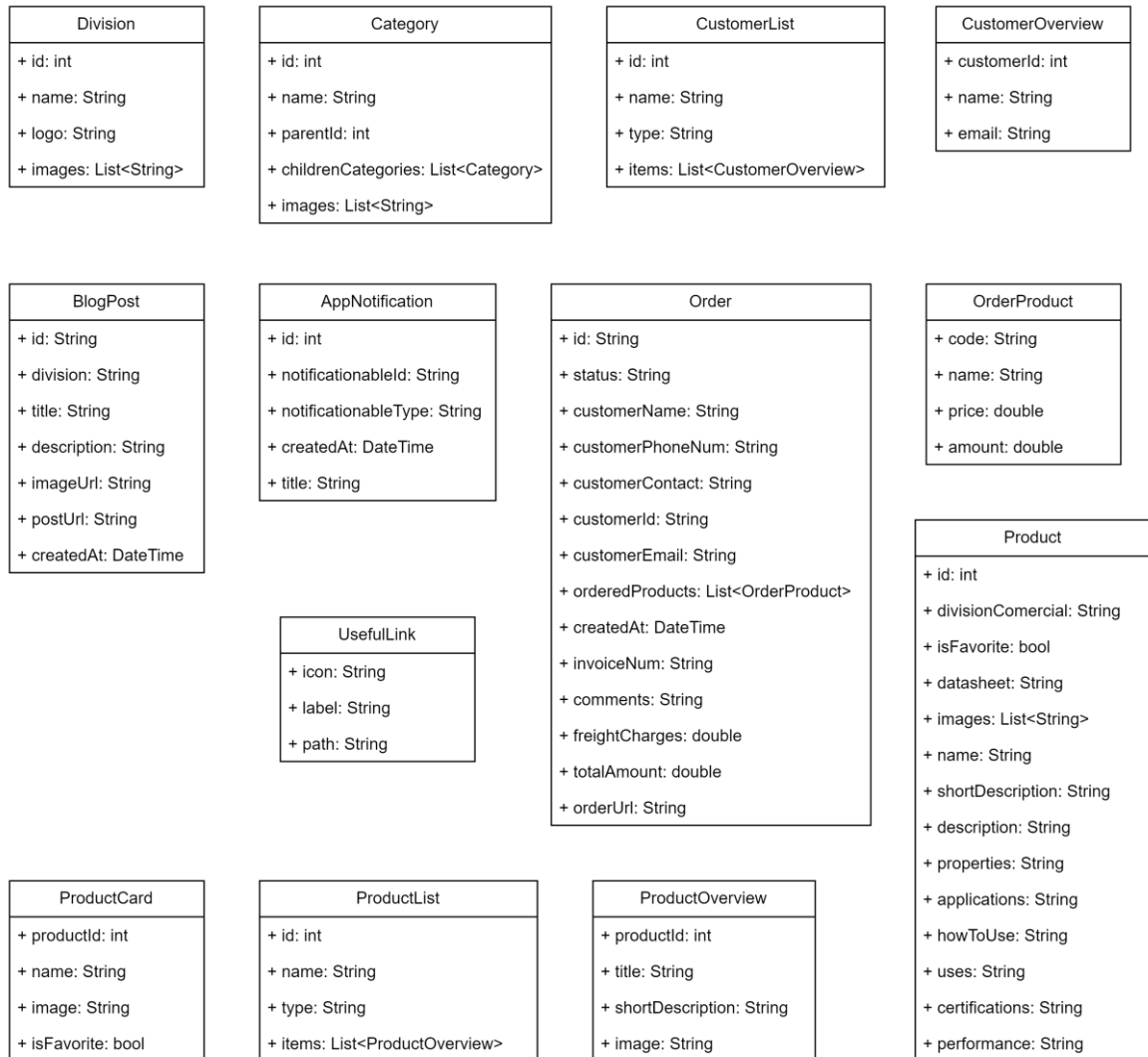


Figura 2. Diagrama de clases

4.2 Diagramas de Secuencias de los Casos de Uso

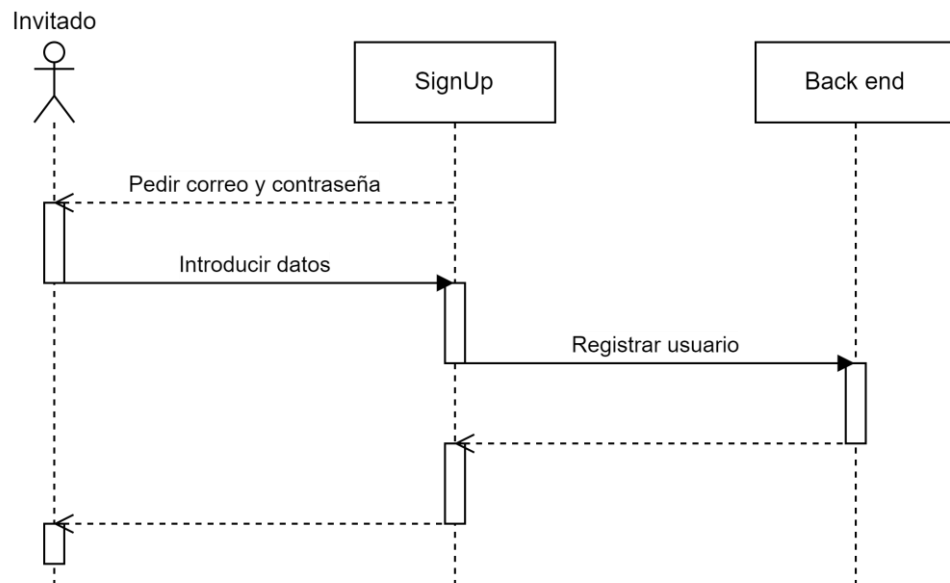


Figura 3. Secuencia del caso de uso 00. SignUp

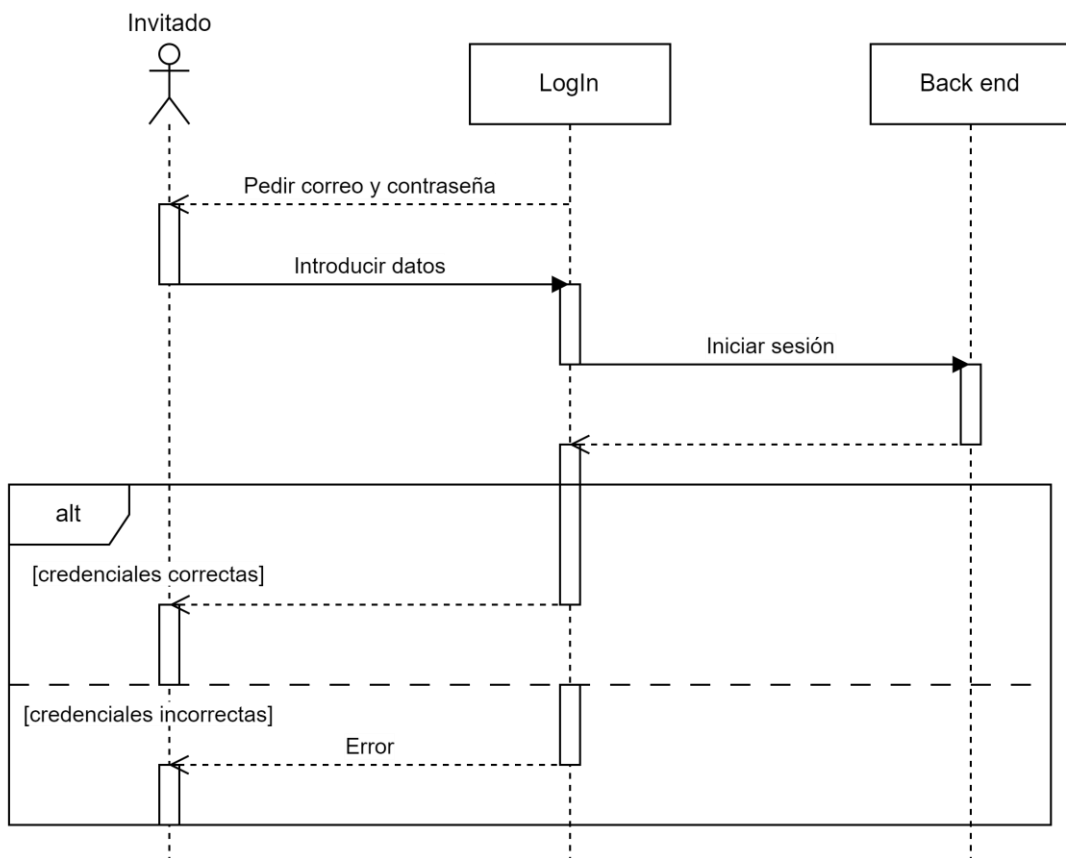


Figura 4. Secuencia del caso de uso 01. LogIn

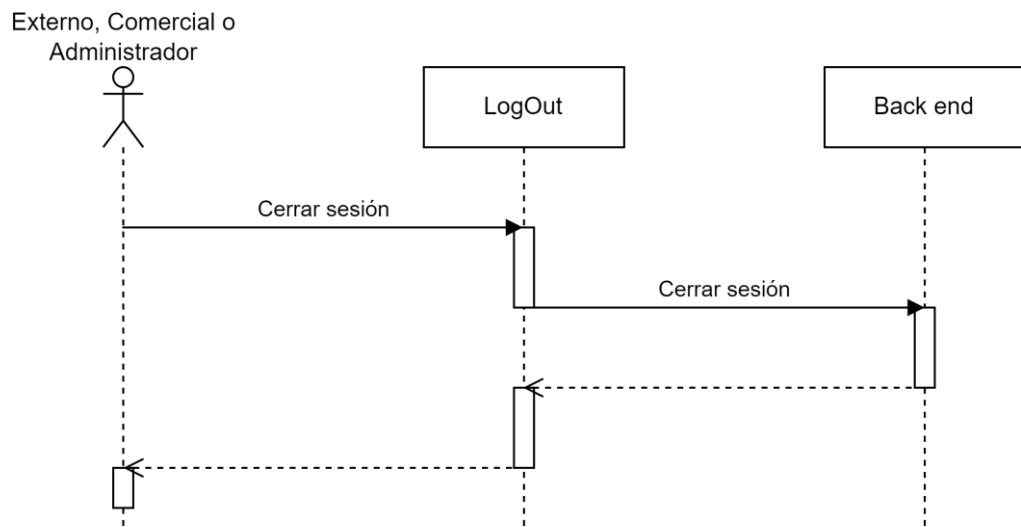


Figura 5. Secuencia del caso de uso 02. LogOut

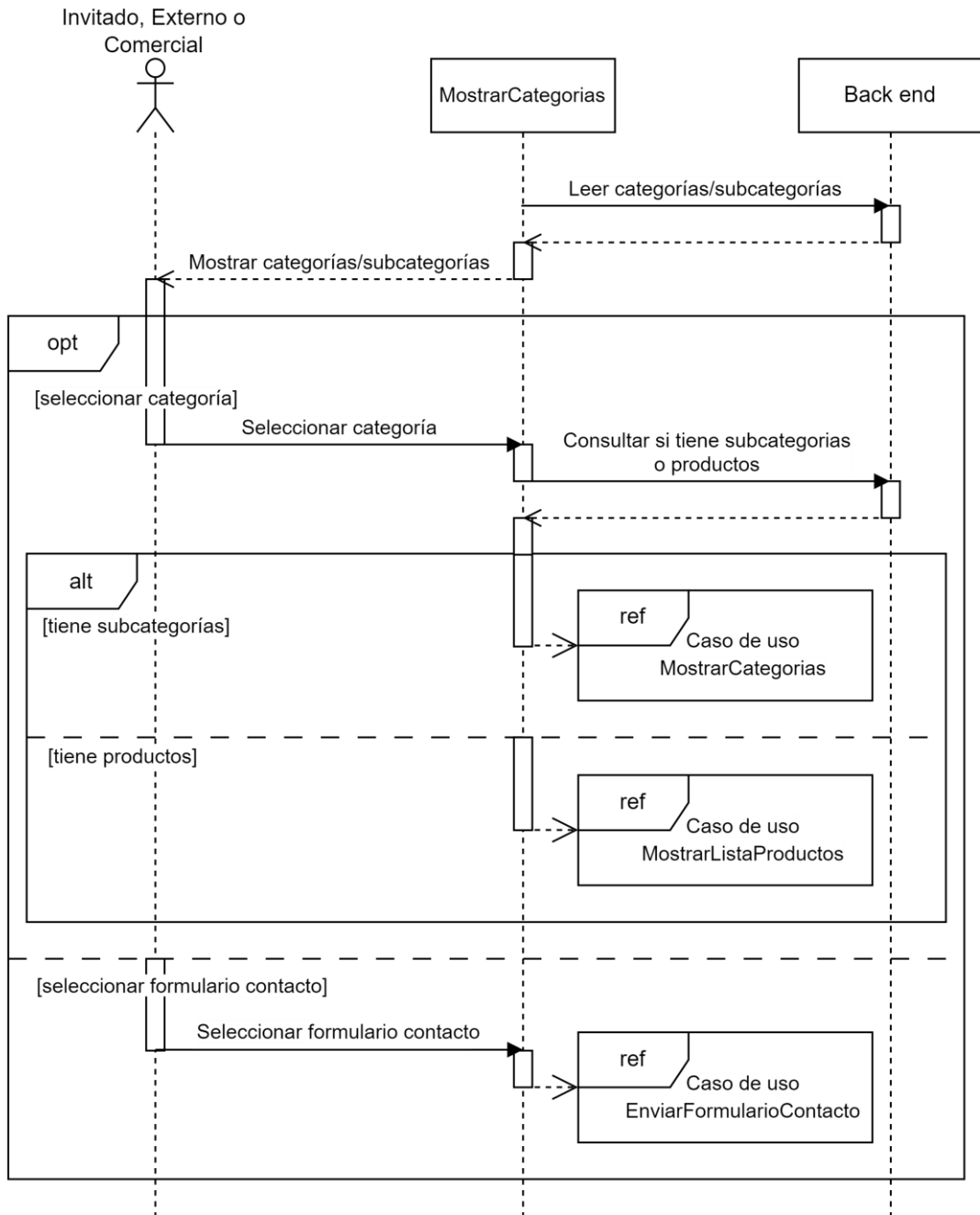


Figura 6. Secuencia del caso de uso 03. MostrarCategorías

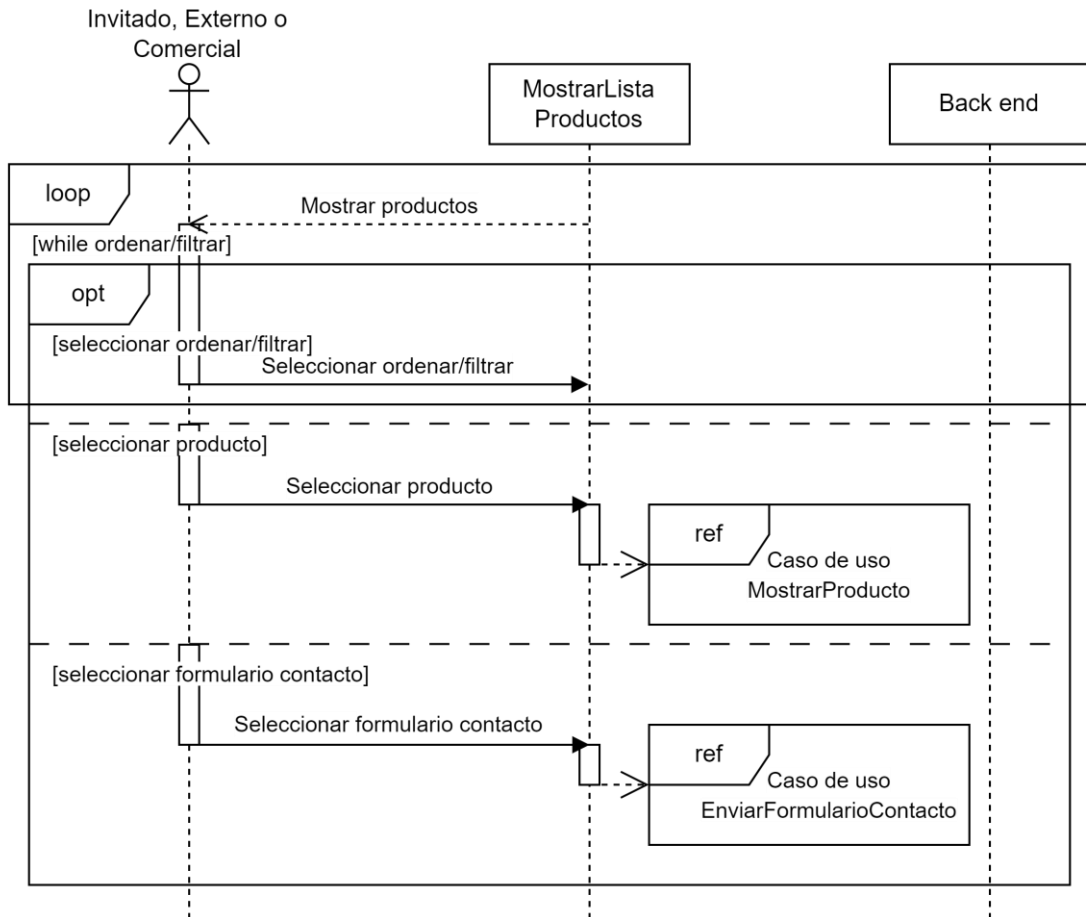


Figura 7. Secuencia del caso de uso 04. MostrarListaProductos

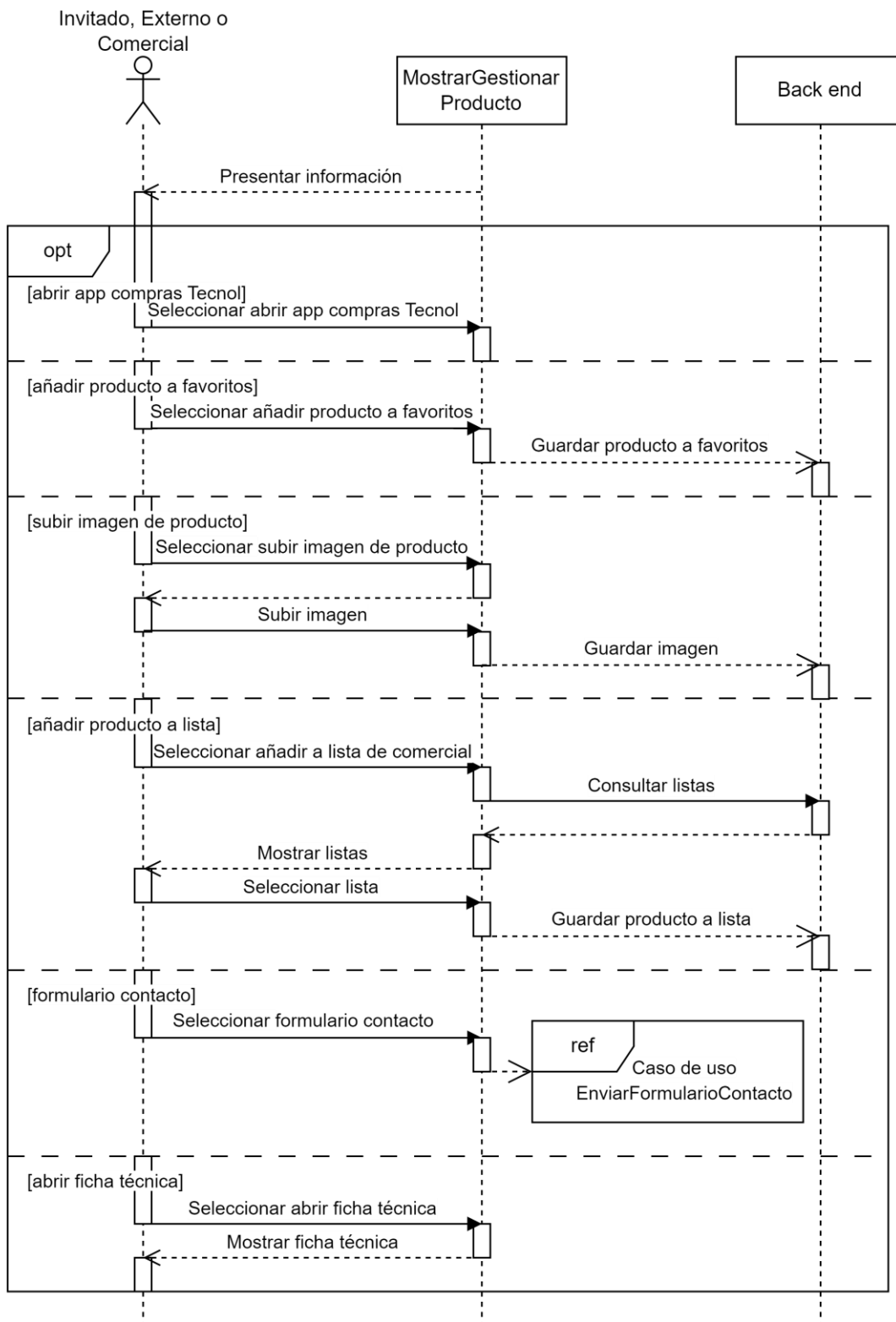


Figura 8. Secuencia del caso de uso 05. MostrarGestionarProducto

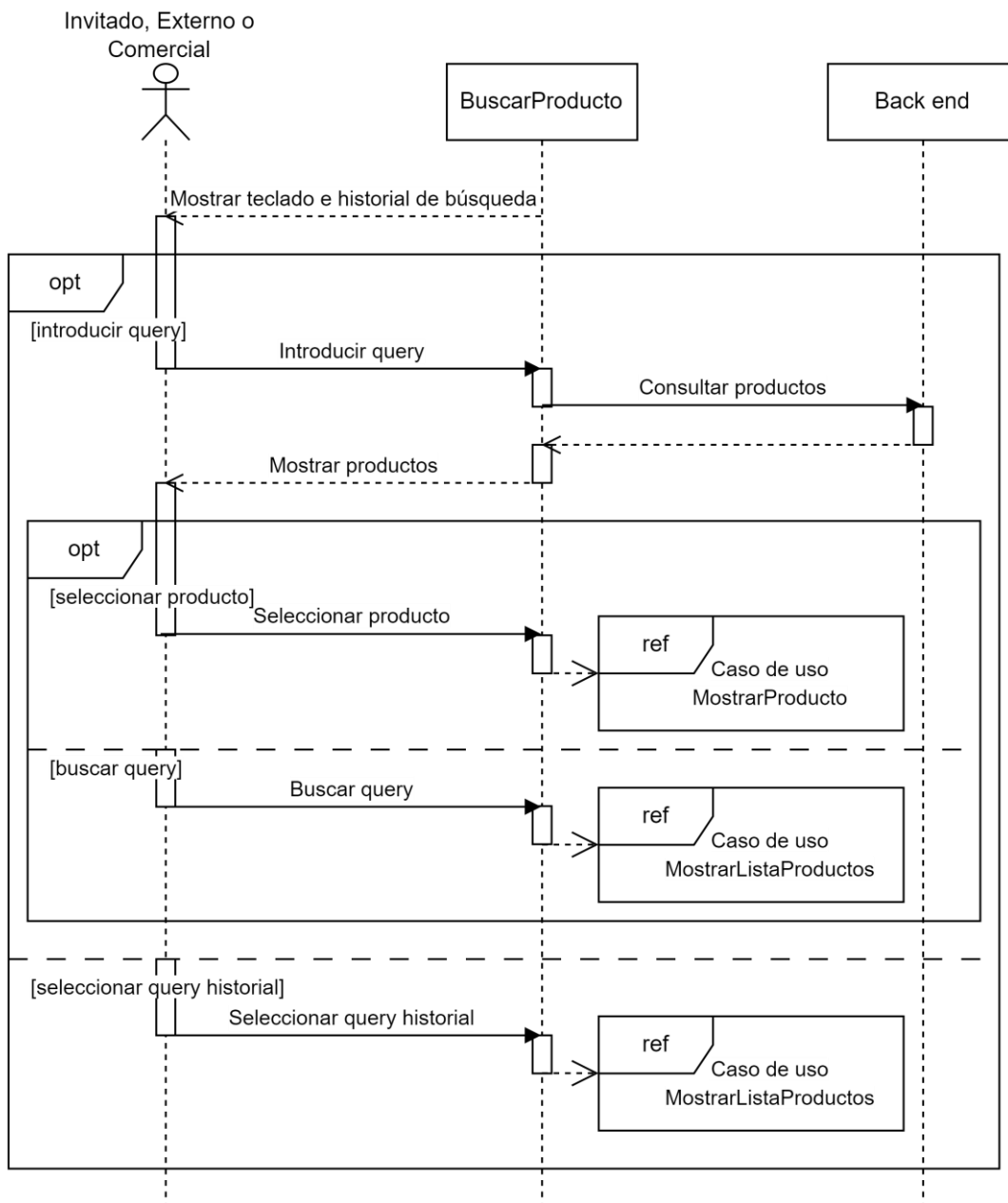


Figura 9. Secuencia del caso de uso 06. BuscarProducto

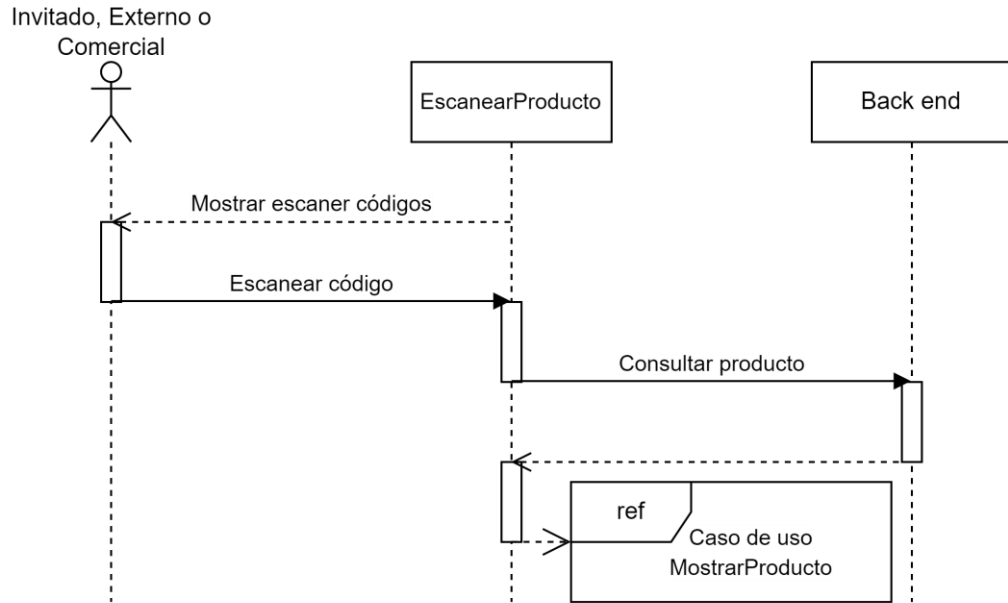


Figura 10. Secuencia del caso de uso 07. EscanearProducto

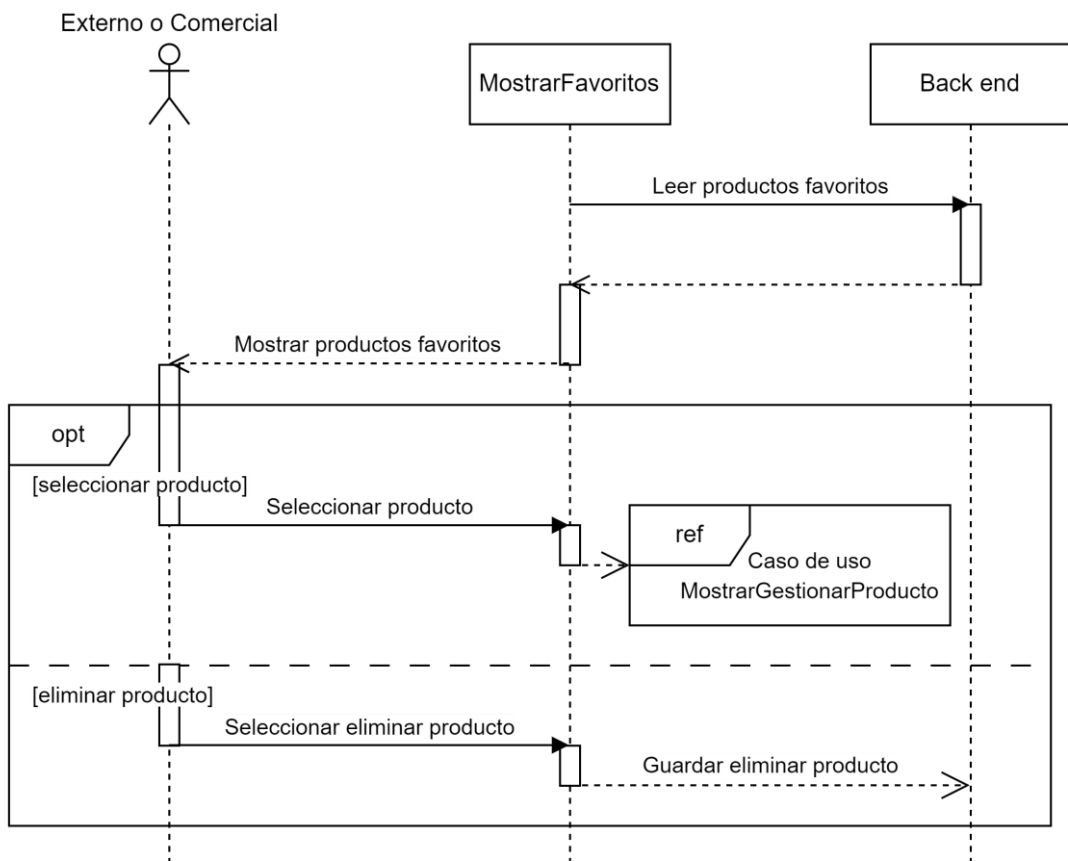


Figura 11. Secuencia del caso de uso 08. MostrarFavoritos

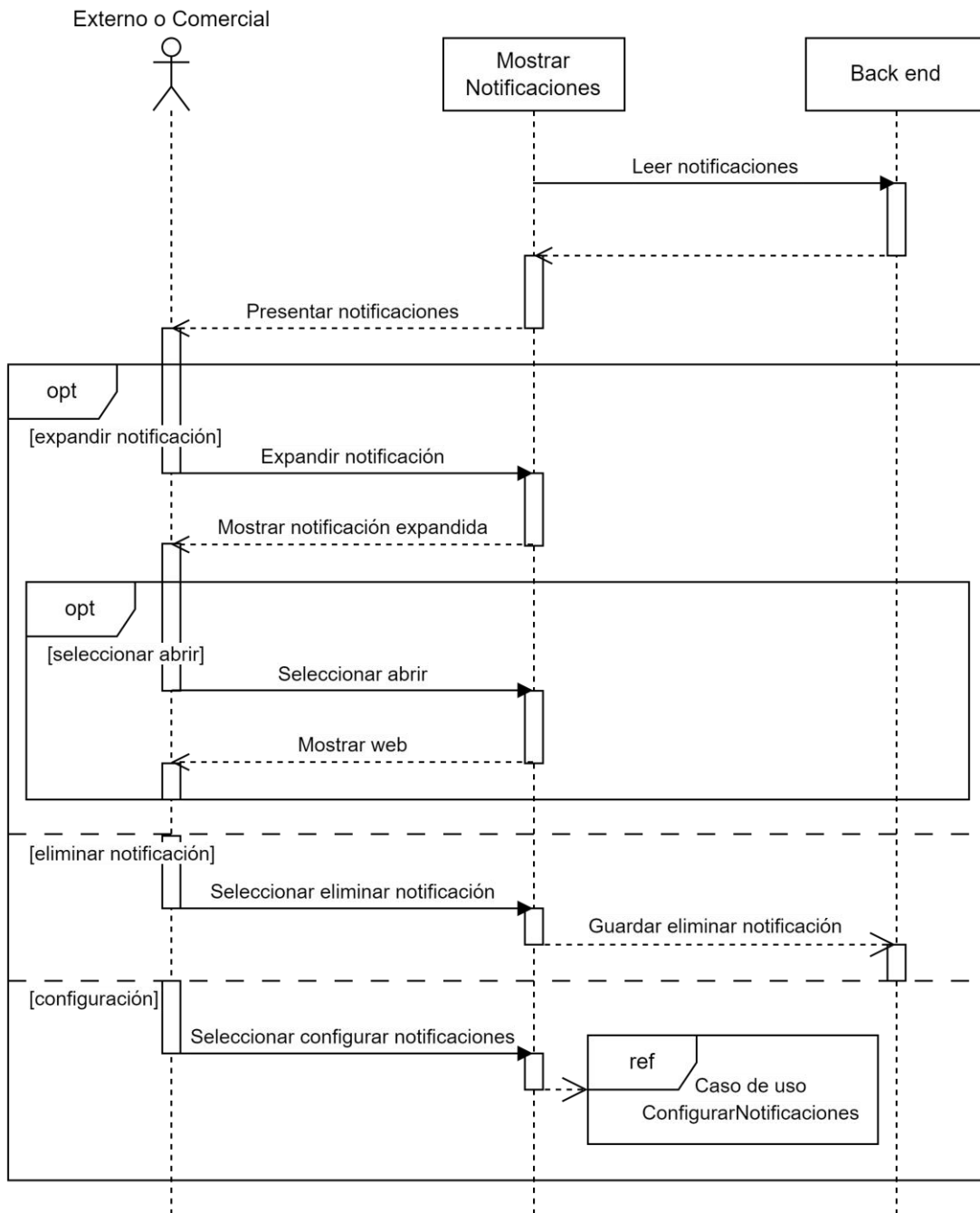


Figura 12. Secuencia del caso de uso 09. MostrarNotificaciones

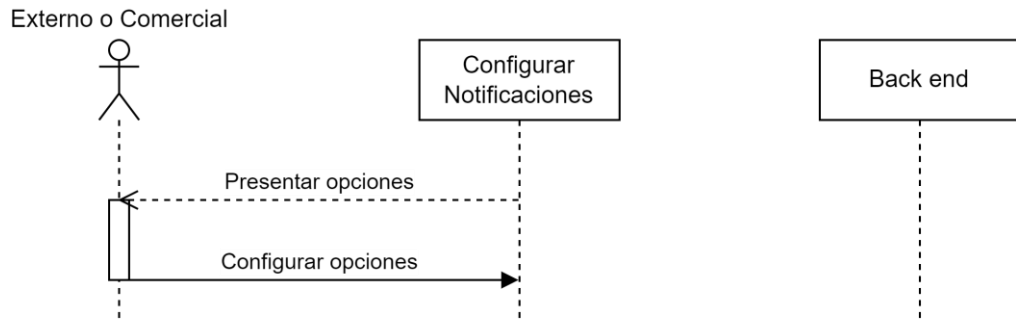


Figura 13. Secuencia del caso de uso 10. ConfigurarNotificaciones

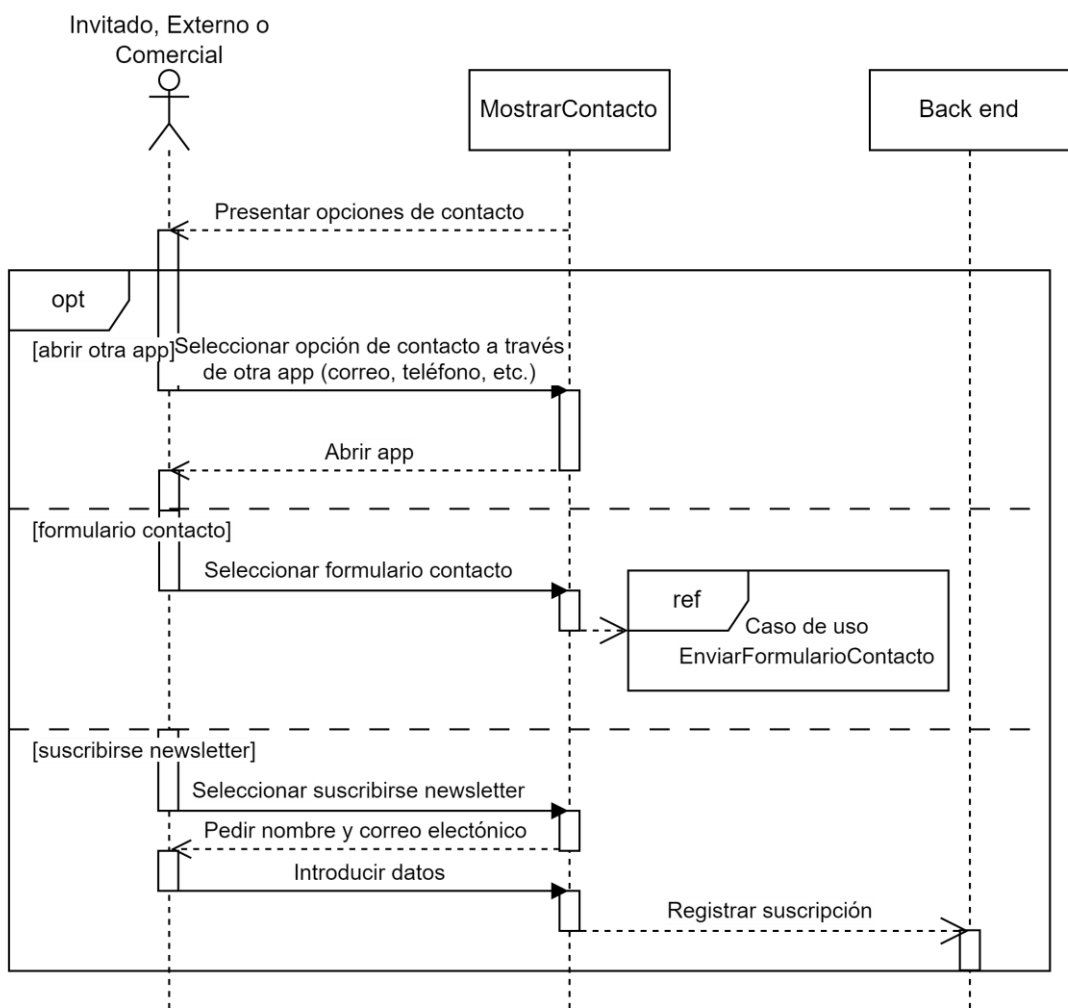


Figura 14. Secuencia del caso de uso 11. MostrarContacto

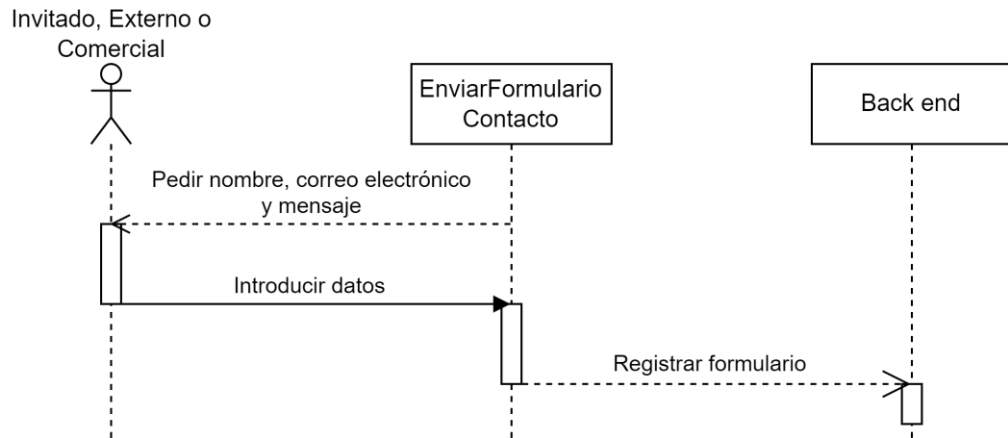


Figura 15. Secuencia del caso de uso 12. EnviarFormularioContacto

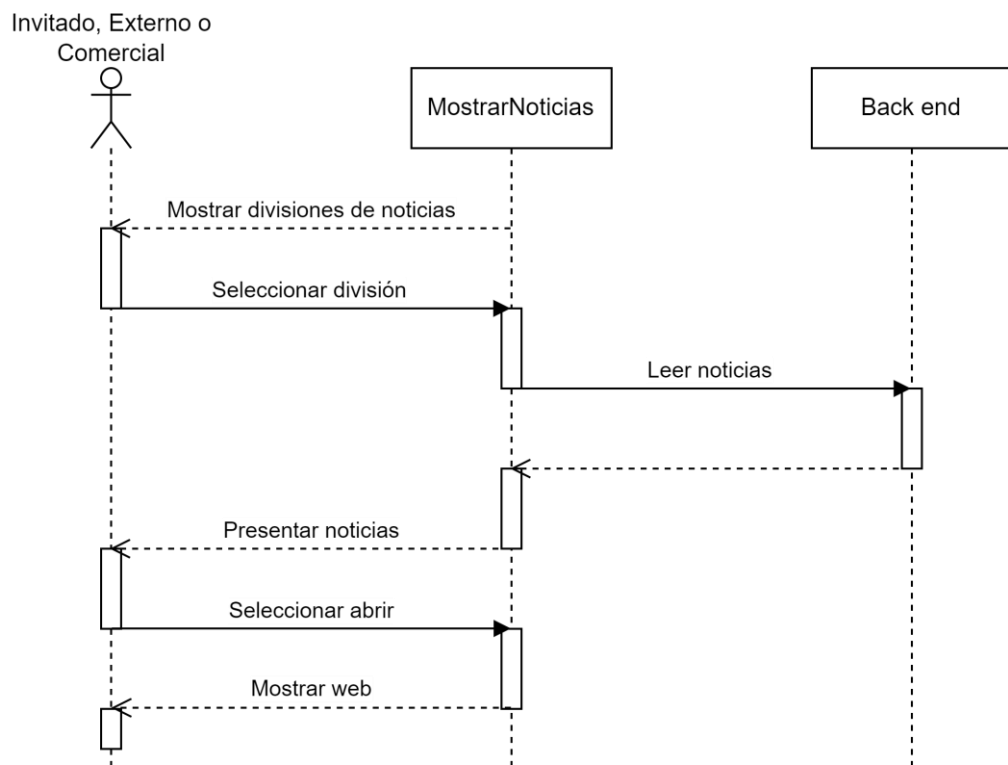


Figura 16. Secuencia del caso de uso 13. MostrarNoticias

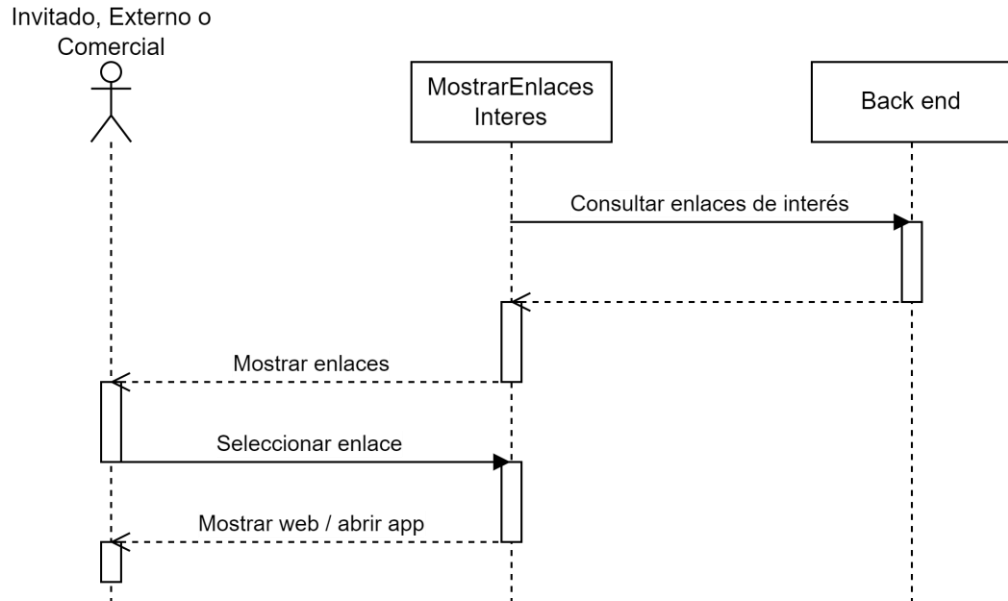


Figura 17. Secuencia del caso de uso 14. MostrarEnlacesInteres

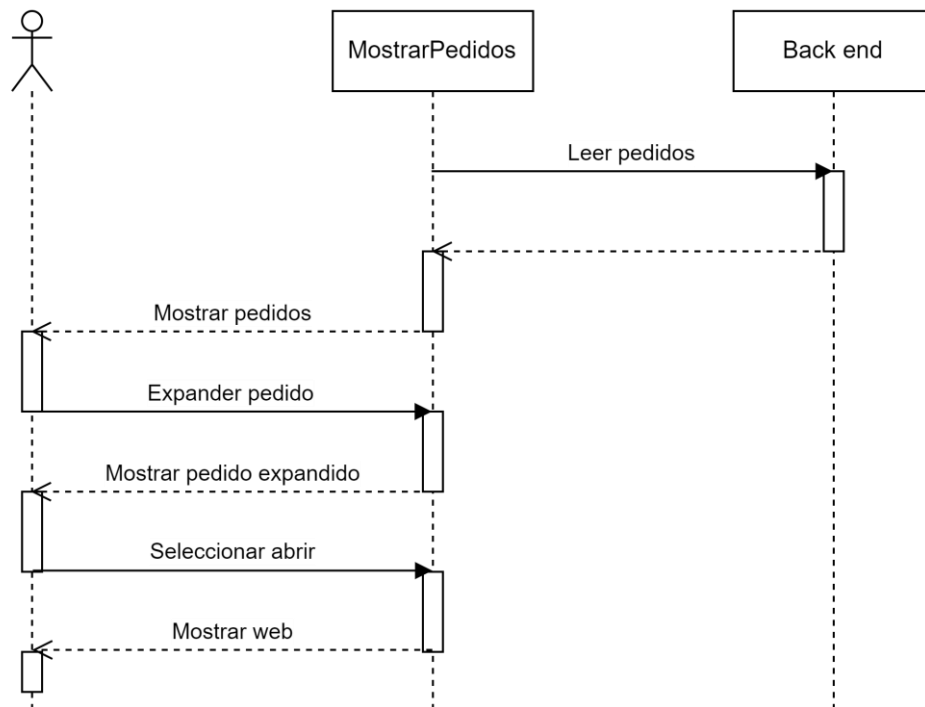


Figura 18. Secuencia del caso de uso 15. MostrarPedidos

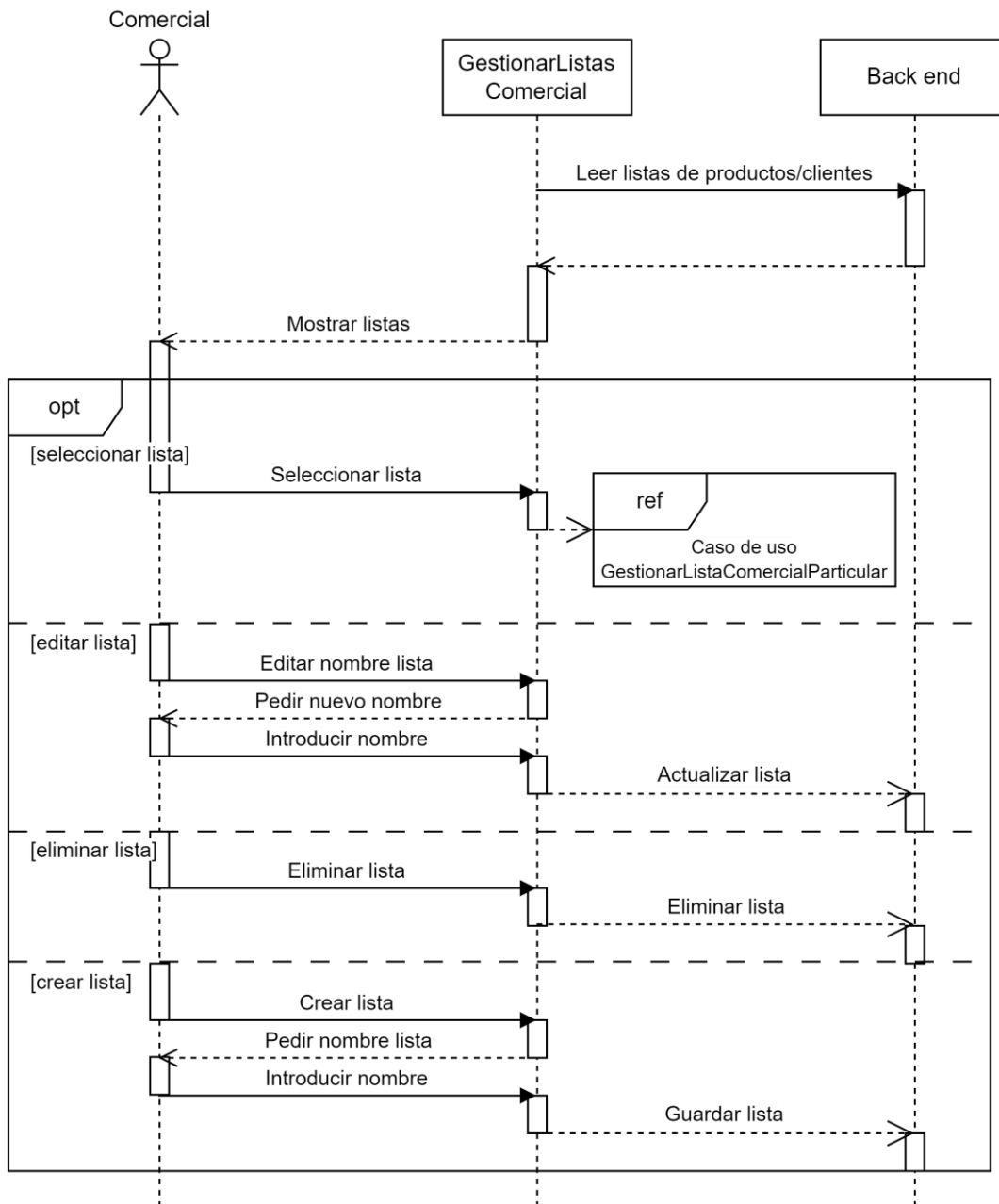


Figura 19. Secuencia del caso de uso 16. GestionarListasComercial

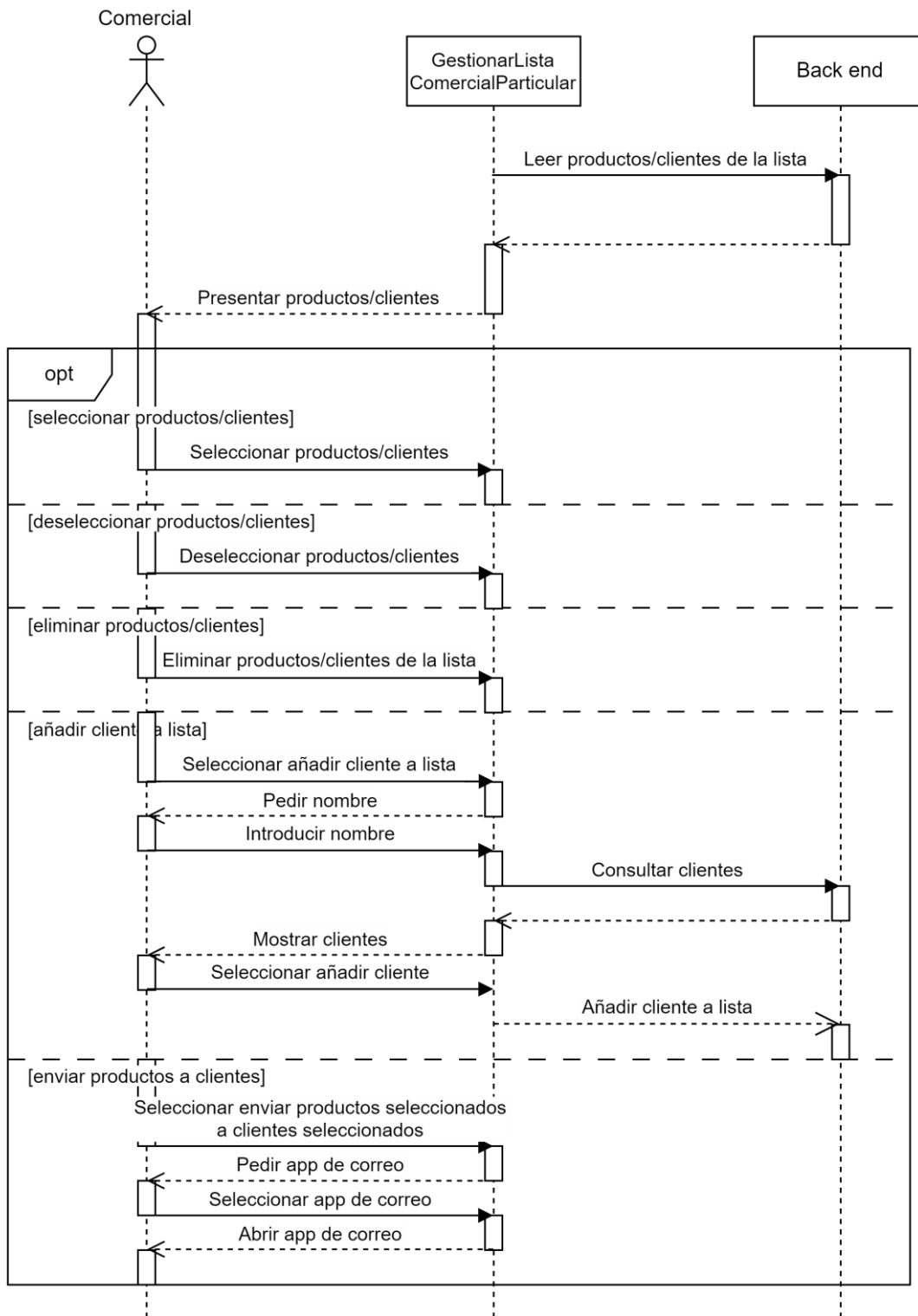


Figura 20. Secuencia del caso de uso 17. GestionarListaComercialParticular

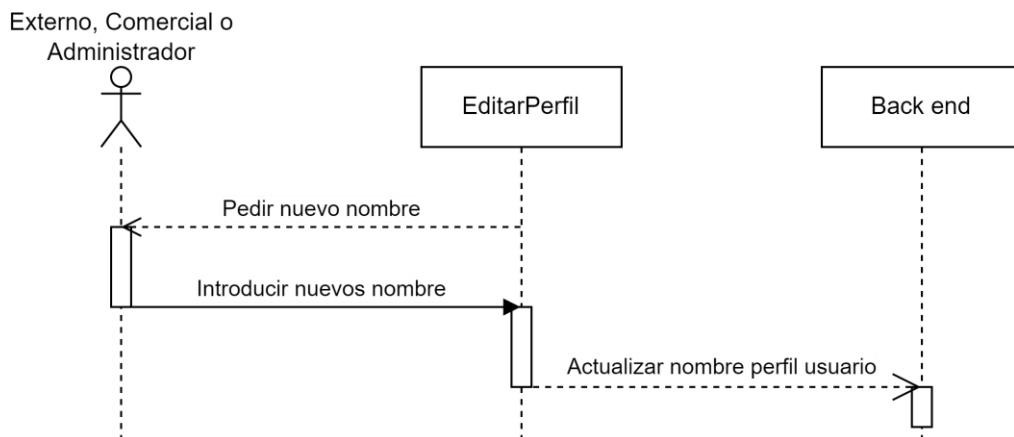


Figura 21. Secuencia del caso de uso 18. EditarPerfil

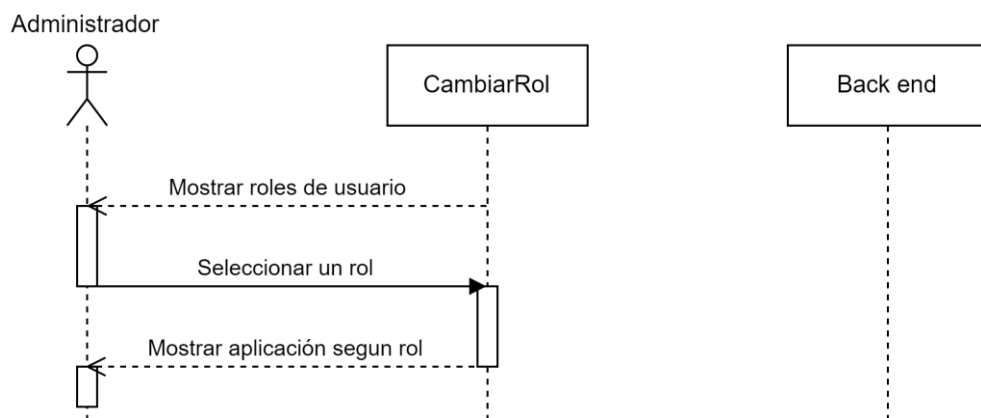


Figura 22. Secuencia del caso de uso 19. CambiarRol

5 Diseño

5.1 Tecnologías a Utilizar

5.1.1 *Front end*

Para desarrollar la aplicación del proyecto se ha utilizado el *framework* Flutter.

Flutter [2] es un *framework* de código abierto desarrollado por Google que permite la creación de aplicaciones en las plataformas iOS, Android, web, Windows, MacOS⁶ y Linux a partir de una única base de código. Su objetivo principal es permitir a los desarrolladores construir aplicaciones nativas de alta calidad para múltiples plataformas de manera eficiente y rápida.

Flutter utiliza el motor gráfico de código abierto Skia de Google, lo que permite que las aplicaciones rendericen gráficos a alta velocidad y de forma consistente, proporcionando un rendimiento cercano al de las aplicaciones nativas.

Dart es el lenguaje de programación que utiliza Flutter. Este es un lenguaje optimizado para el desarrollo rápido de interfaces de usuario en cualquier plataforma. Se compila en código máquina, lo que garantiza un rendimiento rápido y eficaz. Además, ofrece una sintaxis simple y fácil de aprender.

Los componentes básicos de un programa de Flutter son los widgets. Estos describen la lógica, interacción y diseño de los elementos de una UI⁷. Flutter proporciona una amplia gama de widgets predefinidos que pueden ser personalizados para crear interfaces de usuario atractivas y coherentes.

Una de las otras ventajas de Flutter es el *hot reload*. Esta herramienta permite a los desarrolladores previsualizar el aspecto de los cambios de código en tiempo real sin necesidad de recompilar toda la aplicación. Esto permite acelerar el proceso de desarrollo y prueba.

5.1.2 *Back end*

Para desarrollar el back end del proyecto se ha utilizado el *framework* Laravel.

Laravel [3] es un *framework* de desarrollo de aplicaciones y servicios web de código abierto basado en PHP. Es uno de los más populares y ampliamente utilizados en el ecosistema PHP debido a su expresiva y elegante sintaxis, su potente conjunto de herramientas y su enfoque en las mejores prácticas del desarrollo web.

Laravel sigue el patrón de diseño MVC⁸, que ayuda a separar la lógica de negocio, la presentación y la manipulación de los datos. Esta separación ayuda a mantener el código organizado y modular, facilitando el mantenimiento y la escalabilidad del código.

⁶ Sistema operativo que utilizan los ordenadores Macintosh de Apple

⁷ *User Interface*, interfaz de usuario en español

⁸ Modelo-Vista-Controlador

Eloquent es un ORM⁹ incluido en Laravel que proporciona una manera sencilla y elegante de interactuar con la base de datos mediante modelos, permitiendo realizar consultas complejas con una sintaxis intuitiva.

Las migraciones en Laravel permiten versionar y gestionar la estructura de la base de datos de manera controlada. Esto facilita la colaboración entre desarrolladores y asegura que la base de datos se mantenga en un estado consistente. Este sistema hace más sencillo el despliegue de cambios en la base de datos en diferentes entornos (desarrollo, producción), reduciendo el riesgo de errores.

Laravel proporciona un sistema robusto de autenticación y autorización integrado que permite gestionar usuarios y permisos de manera segura y eficiente. Proporciona funcionalidades como el registro de usuarios, inicio de sesión, restablecimiento de contraseñas y control de acceso basado en roles.

Otra característica importante es que facilita la creación de APIs RESTful¹⁰ con enrutamiento potente y flexible y controladores específicos para manejar solicitudes. También incluye herramientas para serializar datos y respuestas JSON¹¹ de manera eficiente, proporcionando una comunicación rápida y clara con el front end. Además, el enrutamiento soporta middleware permitiendo filtrar y manipular solicitudes HTTP¹² antes de que lleguen a los controladores, facilitando la implementación de funciones como autenticación, registro de actividad y protección contra CSRF¹³.

5.2 Arquitectura de la Aplicación

La arquitectura de una aplicación determina su diseño y organización general. Decidir e implementar correctamente la arquitectura a usar permite la escalabilidad y mantenibilidad del proyecto y facilita la colaboración en equipo. Aunque Flutter no impone una estructura de proyecto específica, mantener una separación clara de responsabilidades es clave. Por tanto, después de haber considerado varias arquitecturas se ha decidido que la que mejor se adapta a nuestros requerimientos es el diseño MVC [4] [5].

La arquitectura MVC está compuesta por tres componentes distintos:

- **Modelo:** Representa los datos y la lógica de negocio de la aplicación. Encapsula la información de la aplicación y proporciona métodos para manipularla. En Flutter, los modelos se implementan normalmente utilizando clases Dart que definen la estructura de los datos y contienen métodos para su manipulación, acceso y validación.

⁹ *Object-Relational Mapping*

¹⁰ Implementación de la arquitectura de software *Representational State Transfer*, que permite realizar una comunicación cliente-servidor

¹¹ JavaScript Object Notation, formato de texto para intercambio de datos

¹² Protocolo de Transferencia de Hipertexto

¹³ Cross-site request forgery, tipo de exploit malicioso de un sitio web en el que comandos no autorizados son transmitidos por un usuario en el cual el sitio web confía

- **Vista:** Se encarga de renderizar la interfaz de usuario y mostrar los datos al usuario. En Flutter, la vista se construye utilizando widgets, que son elementos reutilizables y configurables que definen la interfaz de usuario.
- **Controlador:** Maneja la comunicación entre el modelo y la vista. Este permite gestionar la entrada del usuario y actualizar el modelo y la vista en consecuencia.



Figura 23. Diagrama arquitectura MVC

5.3 Diseño de la Persistencia de Datos

El diagrama entidad relación de la base de datos es el siguiente.

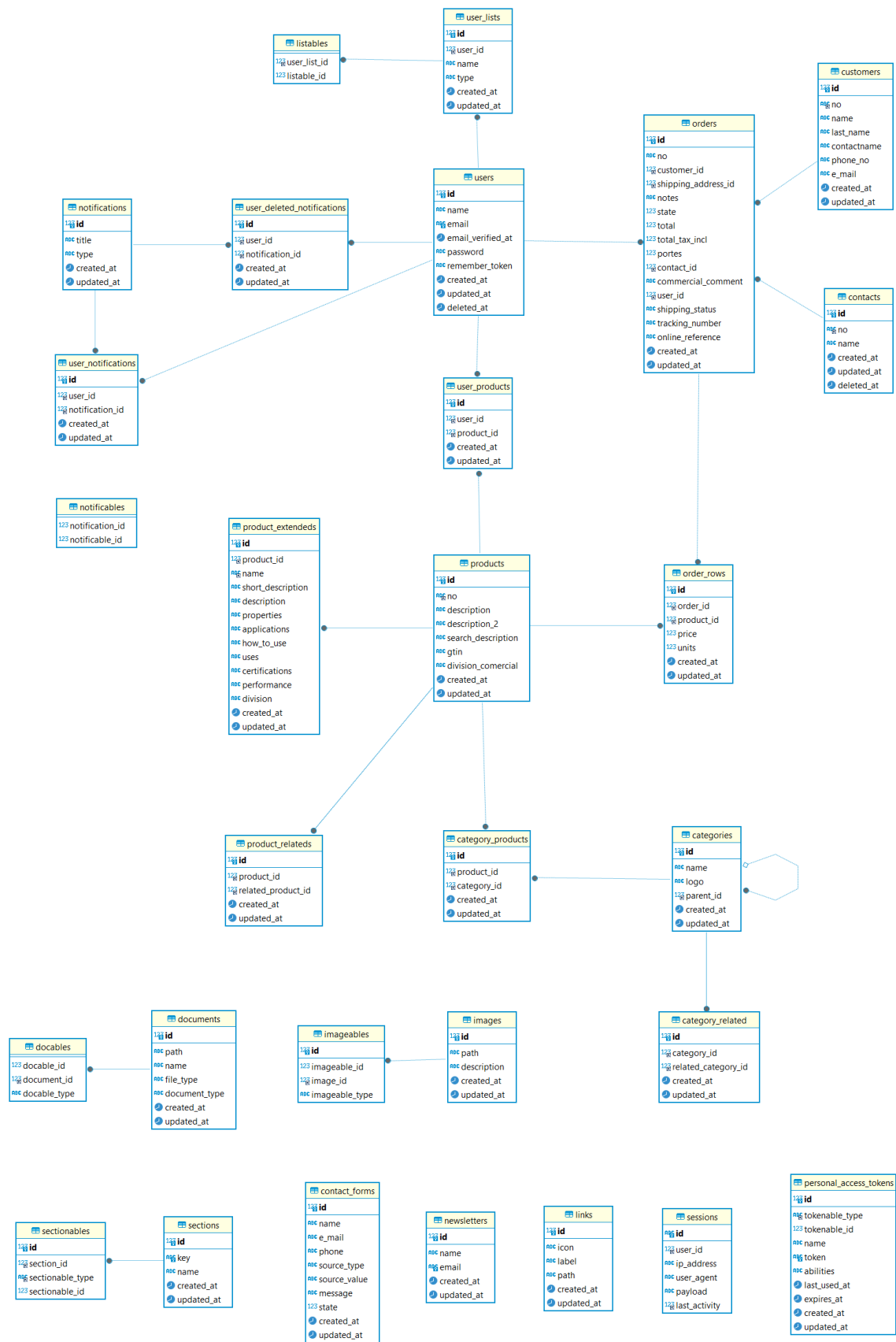


Figura 24. Diagrama entidad relación de la base de datos

5.4 Diseño de API

A continuación, se describirán los endpoints implementados y su propósito.

1. *GET /categories/home/divisions*

Devuelve la información de las divisiones.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body: no.

2. *GET /categories/{categoryId}/tree*

Devuelve el árbol de categorías desde una categoría raíz específica.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {categoryId}: Id de la categoría raíz.

Parámetros query: no.

Body: no.

3. *GET /categories/{categoryId}/related_products*

Devuelve los productos de una categoría específica.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {categoryId}: Id de la categoría.

Parámetros query: no.

Body: no.

4. *GET /products/{productId}*

Devuelve un producto específico.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {productId}: Id del producto.

Parámetros query: no.

Body: no.

5. *POST /products/documents*

Devuelve los enlaces de las fichas técnicas de los productos indicados.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body:

```
{  
  "product_ids": [1,2]  
}
```

6. *GET /user/favorite_products*

Devuelve los productos favoritos del usuario.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body: no.

7. *POST /user/favorite_products*

Registra un producto específico como favorito del usuario.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body:

```
{  
  "product_id": 1  
}
```

8. *DELETE /user/favorite_products/{productId}*

Elimina un producto específico de los favoritos del usuario.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {productId}: Id del producto.

Parámetros query: no.

Body: no.

9. *GET /search?search={search}&type={type}*

Devuelve el/los producto/s o cliente/s que coinciden con el parámetro de búsqueda.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query:

- search: query de los productos a buscar, número de código de barras o query del cliente a buscar.
- type: tipo de búsqueda. Puede valer product, barcode o customer.

Body: no.

10. *POST /contact-form*

Registra un nuevo formulario de contacto.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body:

```
{  
  "name": "Ejemplo Nombre",  
  "e_mail": "ejemplo@email.com",  
  "phone": "123456789",  
  "message": "Mensaje de ejemplo",  
  "source_type": "product",  
  "source_value": "1"  
}
```

11. *POST /newsletter*

Registra una nueva suscripción a la newsletter.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body:

```
{  
  "name": "Ejemplo Nombre",  
  "email": "ejemplo@email.com"  
}
```

12. GET /links

Devuelve los enlaces de interés del comercial.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body: no.

13. GET /user/lists?list_type={listType}

Devuelve las listas y sus productos/clientes según el tipo que se indique.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query:

- listType: tipo de lista, puede valer product o customer.

Body: no.

14. POST /user/lists

Crea una nueva lista.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body:

```
{  
  "list_type": "product",  
  "name": "Ejemplo nombre lista"  
}
```

15. DELETE /user/lists/{listId}/remove

Elimina la lista indicada.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {listId}: id de la lista.

Parámetros query: no.

Body: no.

16. PATCH /user/lists/{listId}

Actualiza el nombre de la lista indicada.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {listId}: id de la lista.

Parámetros query: no.

Body:

```
{
  "name": "Ejemplo nuevo nombre lista"
}
```

17. POST /user/lists/{listId}

Añadir un producto/cliente a la lista indicada.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {listId}: id de la lista.

Parámetros query: no.

Body:

```
{
  "listable_id": "1"
}
```

18. DELETE /user/lists/{listId}/remove/{listableId}

Quitar un producto/cliente de la lista indicada.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {listId}: id de la lista.
- {listableId}: id del producto/cliente.

Parámetros query: no.

Body: no.

19. GET /home

Devuelve las secciones y sus productos de la pantalla de inicio.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body: no.

20. *PATCH /user*

Actualiza el nombre del usuario.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body:

```
{  
  "name": "Nuevo Nombre"  
}
```

21. *GET /orders*

Devuelve los pedidos.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body: no.

22. *GET /notifications*

Devuelve las notificaciones.

Parámetros header: Bearer personal_access_token.

Parámetros path: no.

Parámetros query: no.

Body: no.

23. *GET /posts/{divisionId}*

Devuelve las noticias de la división indicada.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {divisionId}: id de la división.

Parámetros query: no.

Body: no.

24. *POST /products/{productId}/upload_image*

Permite subir una imagen para un producto específico.

Parámetros header: Bearer personal_access_token.

Parámetros path:

- {productId}: id del producto.

Parámetros query: no.

Body: debe ser multipart/form-data con un campo 'image' que contenga el archivo de la imagen.

5.5 Prototipos de la Interfaz Gráfica

En este apartado se mostrarán los prototipos de la interfaz de usuario creados para la aplicación. Estos fueron elaborados utilizando Figma [6], una aplicación web colaborativa para el diseño de interfaces, que me ha permitido visualizar la disposición de elementos usuario antes de proceder al desarrollo del código.



Figura 25. Prototipo de pantalla de bienvenida

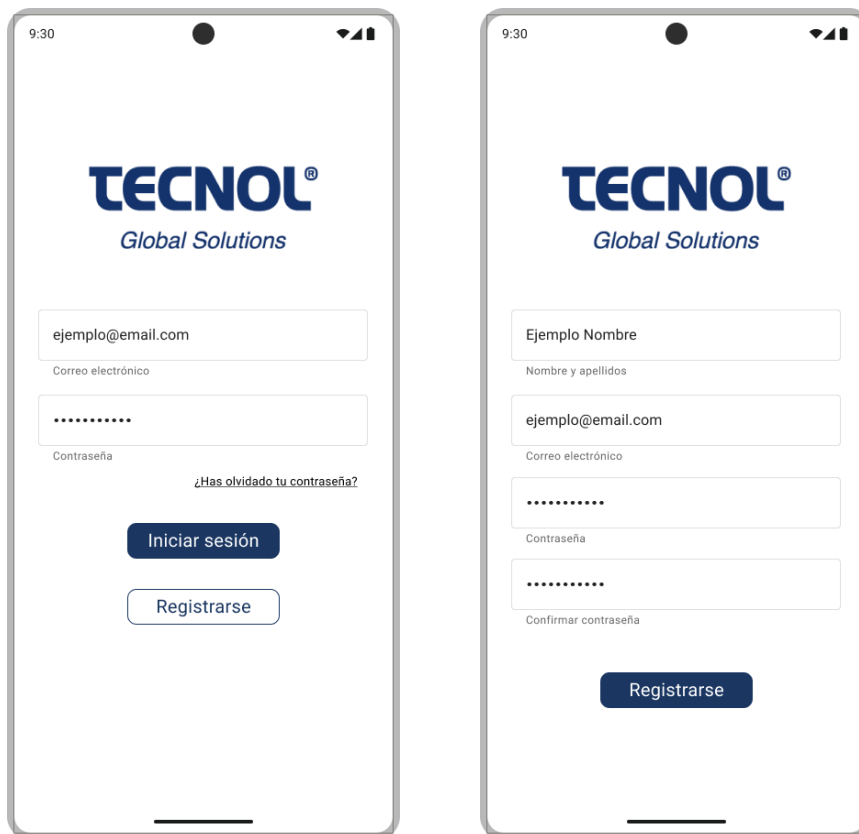


Figura 26. Prototipos de pantallas de inicio de sesión y registro

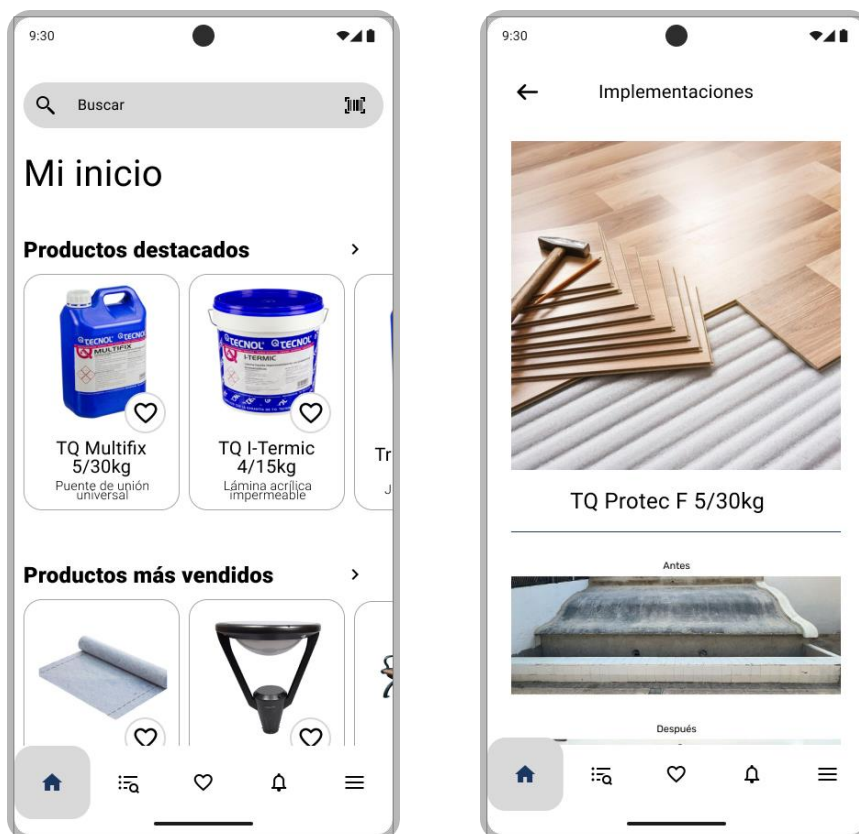


Figura 27. Prototipos de pantallas de inicio y productos aleatorios



Figura 28. Prototipos de pantallas de divisiones y subcategorías

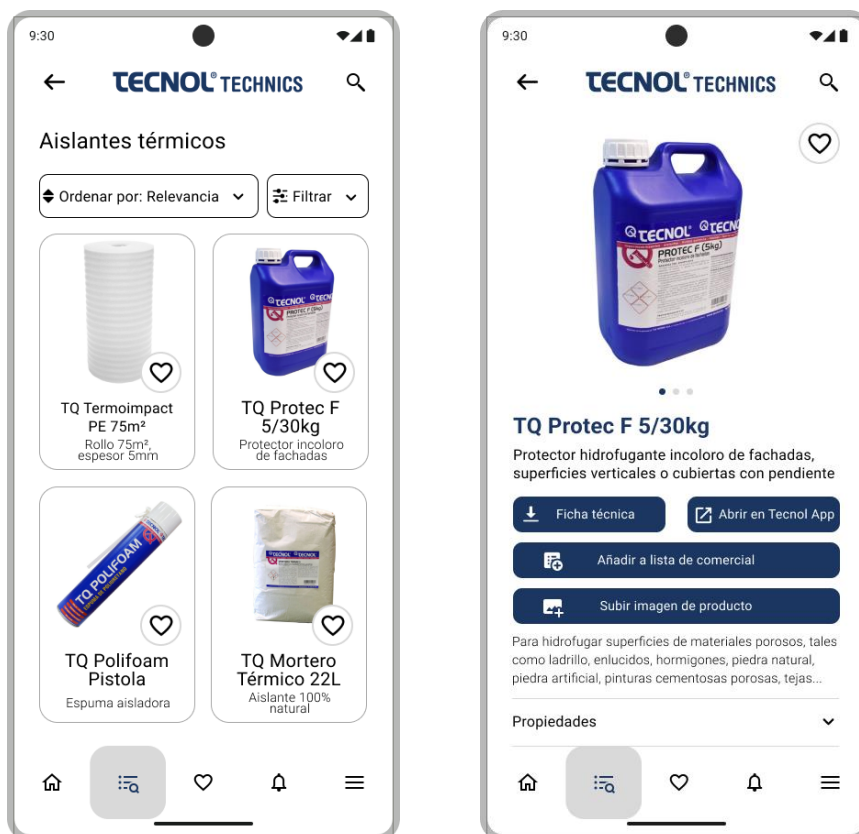


Figura 29. Prototipos de pantallas de lista de productos y detalles de producto

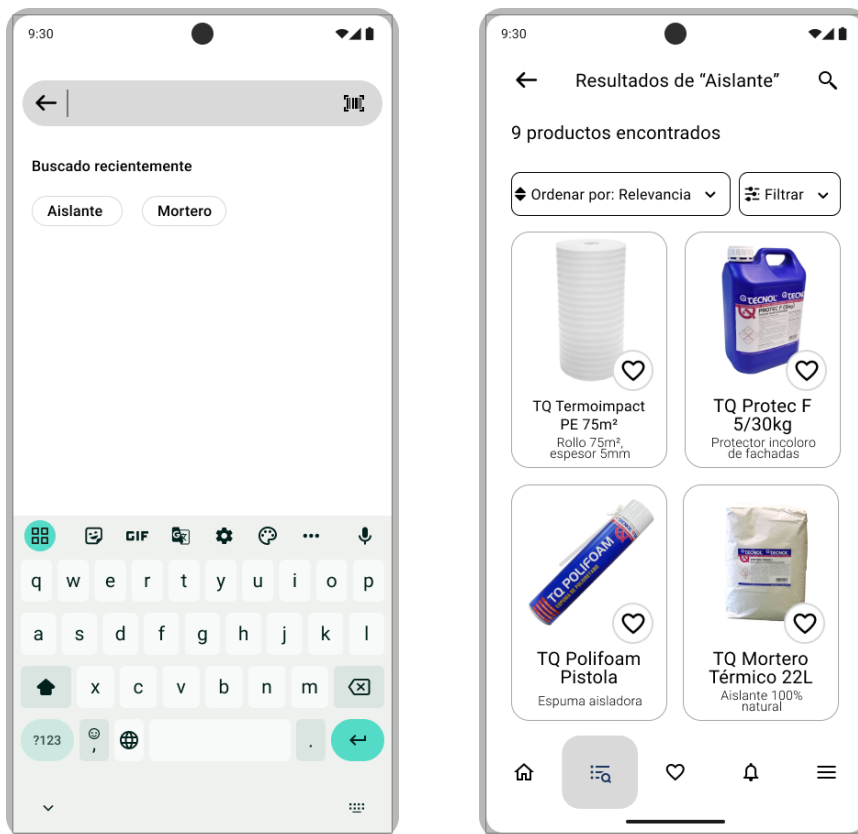


Figura 30. Prototipos de pantallas de buscar producto y resultados de búsqueda

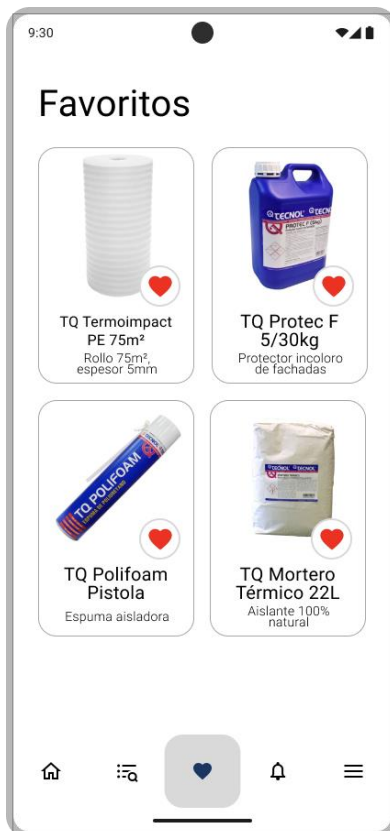


Figura 31. Prototipo de pantalla de favoritos

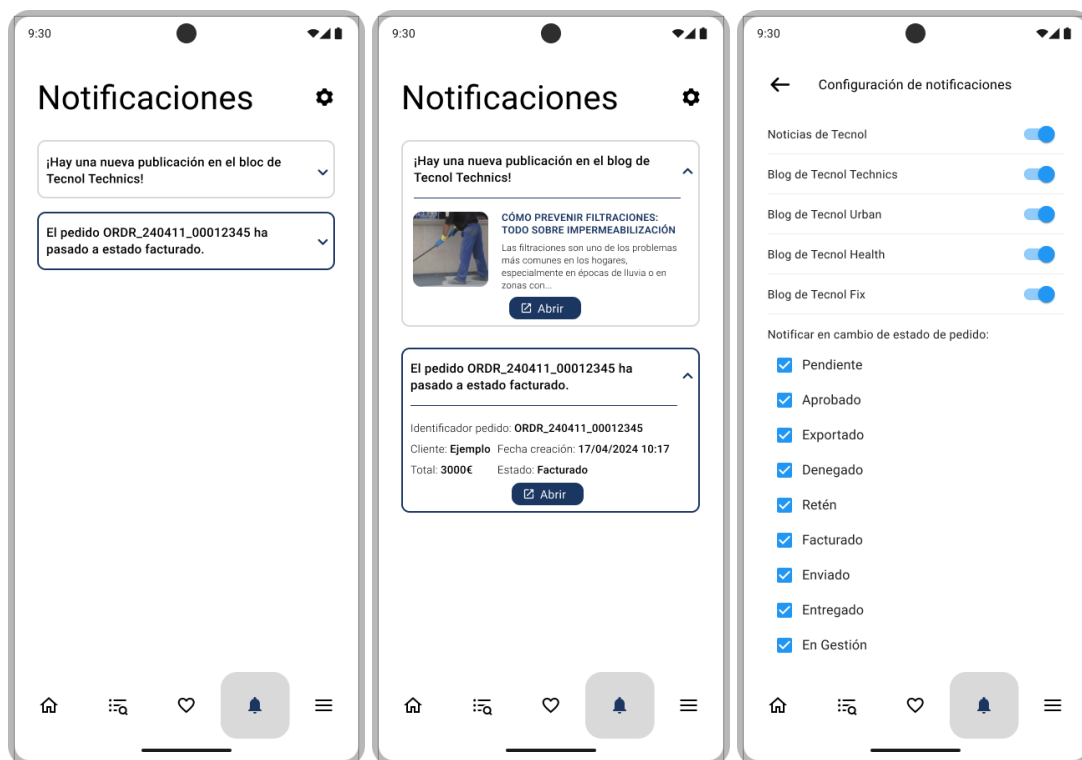


Figura 32. Prototipos de pantallas de notificaciones contraídas, expandidas y configuración

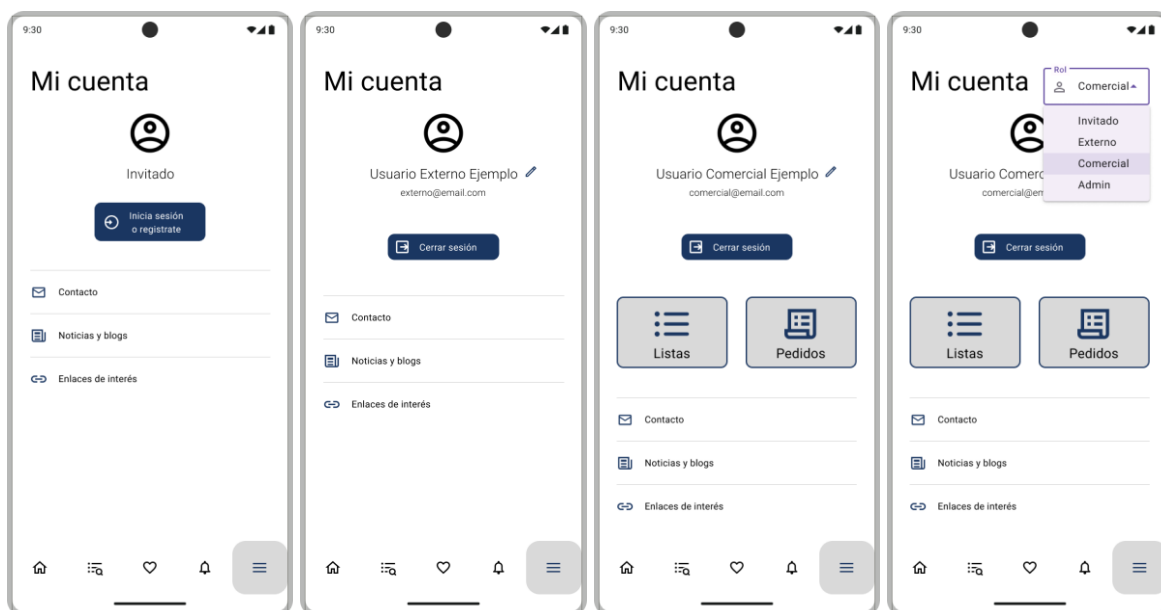


Figura 33. Prototipos de pantallas de cuenta de usuario invitado, externo, comercial y administrador

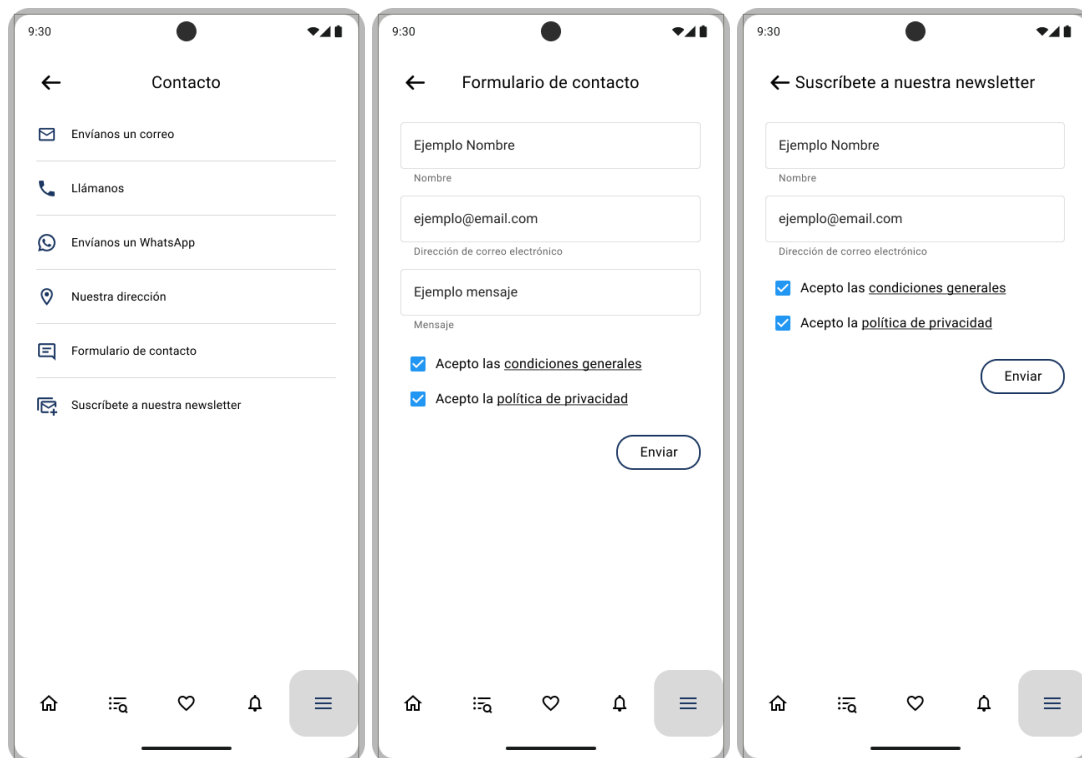


Figura 34. Prototipos de pantallas de contacto, formulario de contacto y suscribirse a newsletter

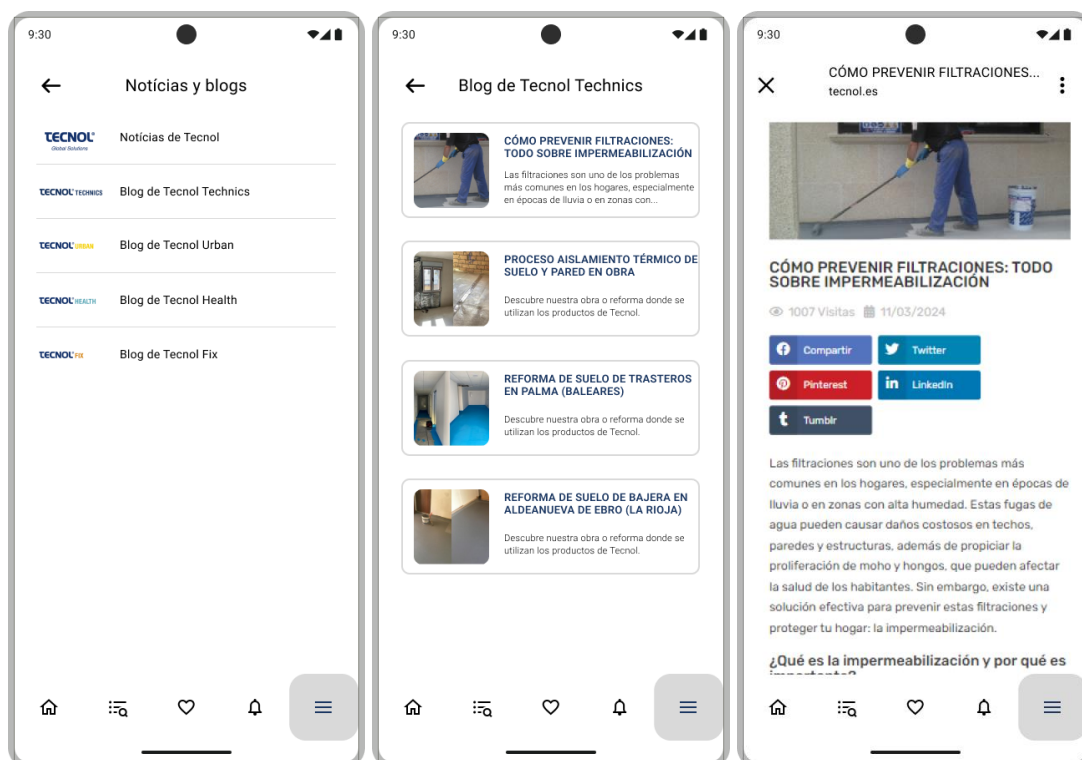


Figura 35. Prototipos de pantallas de divisiones de noticias, lista de noticias y web de una noticia

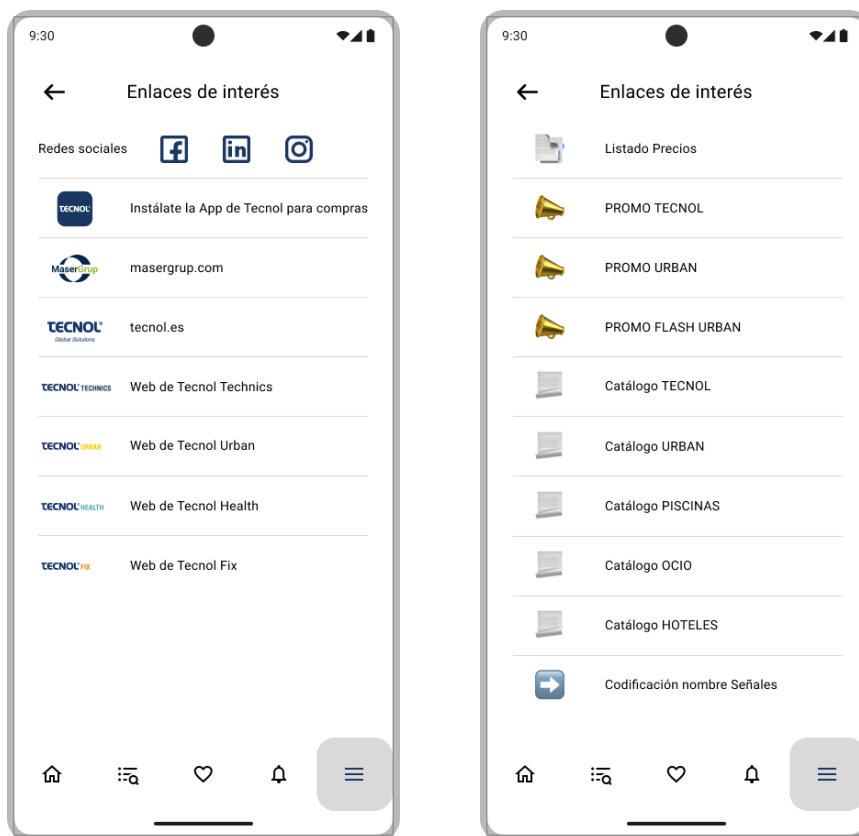


Figura 36. Prototipos de pantallas de enlaces de interés general y del usuario comercial

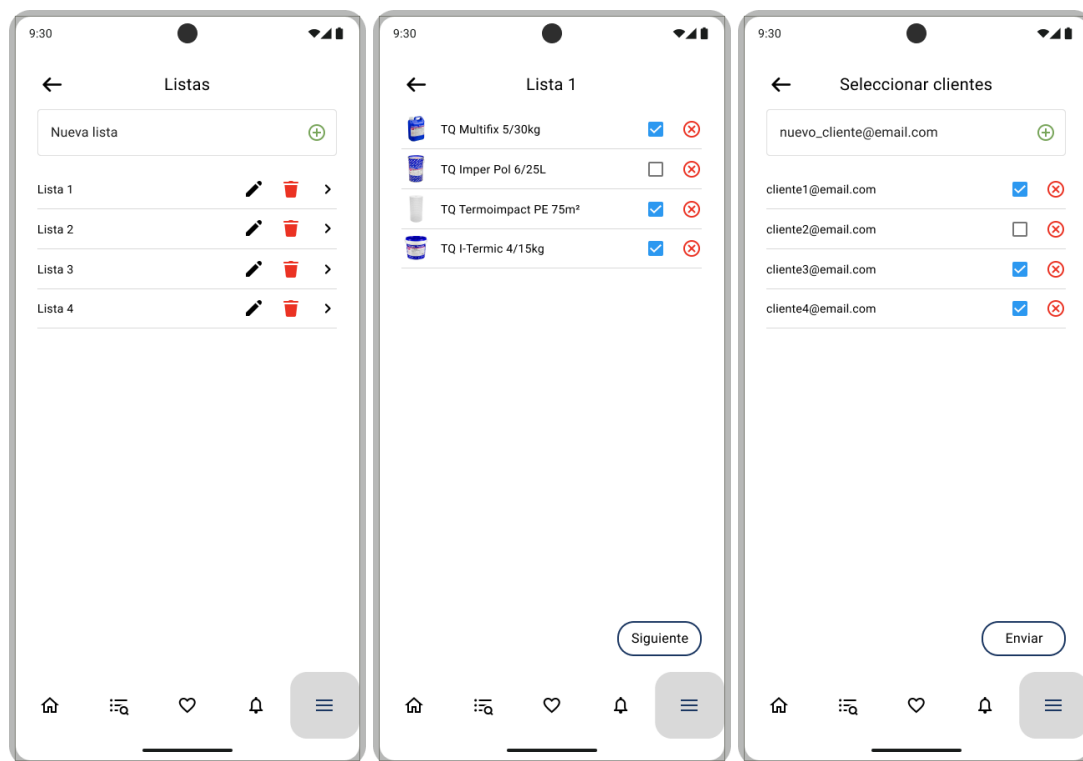


Figura 37. Prototipos de pantallas de listas, lista de productos y de clientes del comercial

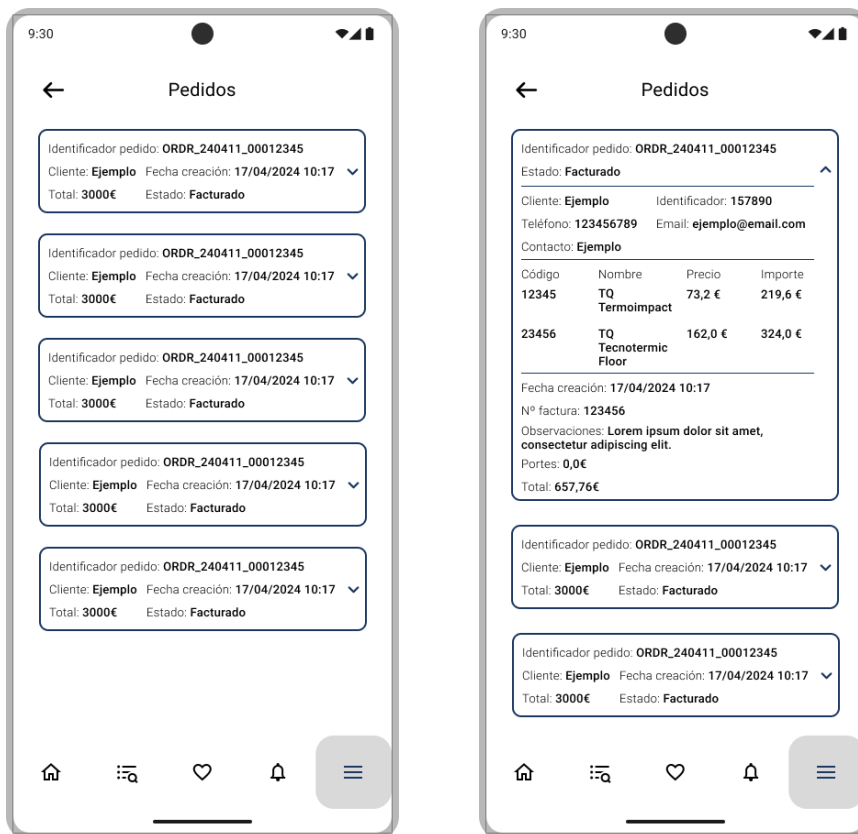


Figura 38. Prototipos de pantallas de pedidos contraídos y expandidos

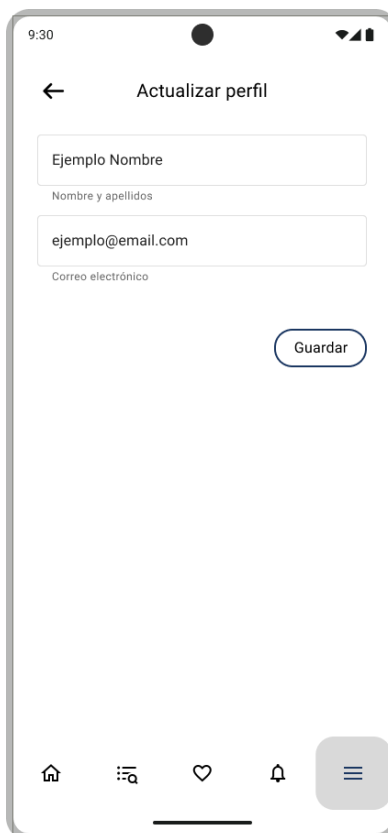


Figura 39. Prototipo de pantalla de actualizar perfil

5.6 Diseño Final de la Interfaz Gráfica

En esta sección se explicará el flujo de uso de la aplicación entre las diferentes pantallas y se mostrarán capturas con su diseño final.

5.6.1 Pantalla de bienvenida

Cuando se inicia la aplicación de muestra la pantalla de bienvenida, que presenta el icono de la aplicación durante 1 segundo.



Figura 40. Pantalla de bienvenida

5.6.2 Pantalla de inicio

Una vez desaparece la pantalla de bienvenida se muestra la pantalla de inicio. Esta pantalla permite acceder a los productos destacados, más vendidos, nuevos y en promoción.



Figura 41. Pantalla de inicio

También, deslizando hacia abajo, se tiene acceso a una pantalla con productos aleatorios para poder explorarlos sin tener que buscar por categorías concretas.



Figura 42. Pantalla de productos aleatorios

5.6.3 Barra de búsqueda de productos

Cuando se accede a la barra de búsqueda superior se muestra la siguiente pantalla donde se visualiza el historial de búsquedas y si se introduce un texto de consulta se exponen los nombres de productos que coinciden.

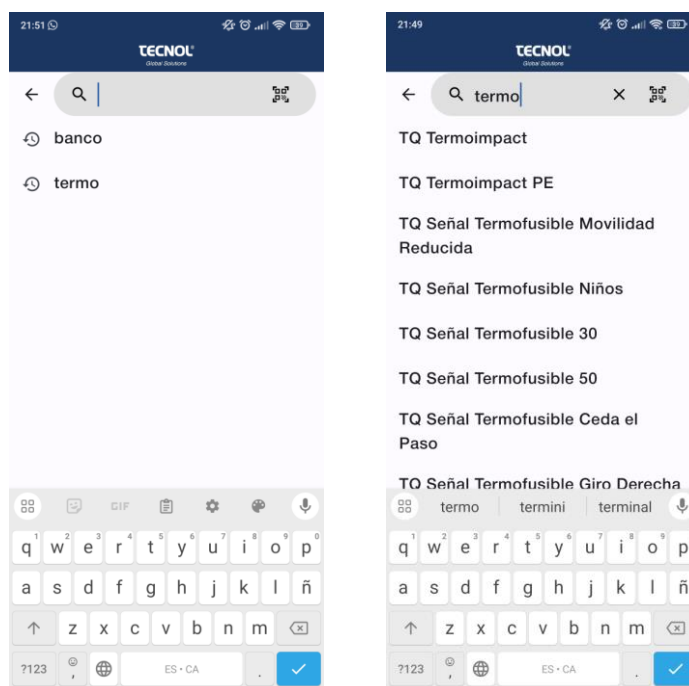


Figura 43. Historial y resultados de búsqueda

Si le damos a buscar (icono ✓ en el teclado) la consulta introducida, aparece la pantalla de lista de productos con todos aquellos que coinciden.

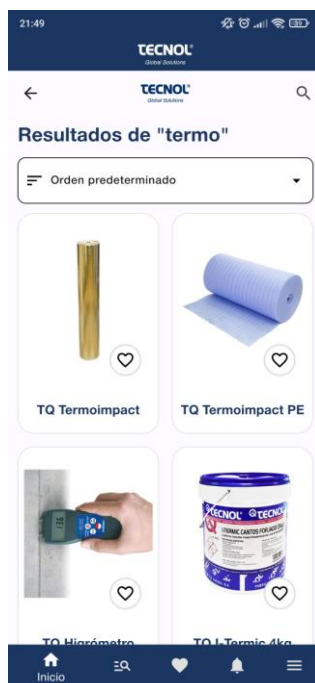


Figura 44. Pantalla con lista de productos de una búsqueda

Por otra parte, si se accede al icono de un escáner de códigos aparece la siguiente pantalla que inicia la cámara para detectar un código de barras y mostrar la pantalla de detalles de producto una vez detectado.

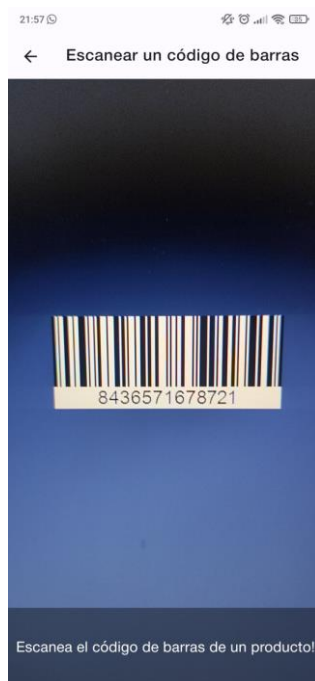


Figura 45. Pantalla de escáner de códigos

5.6.4 Pantalla de categorías

Cuando se navega a la pestaña de categorías en primer lugar aparece la pantalla con las divisiones de la empresa.



Figura 46. Pantalla de divisiones

Si se accede a una división aparecen sus subcategorías. Debajo de ellas hay un botón para ir al formulario de contacto asociado a la categoría.

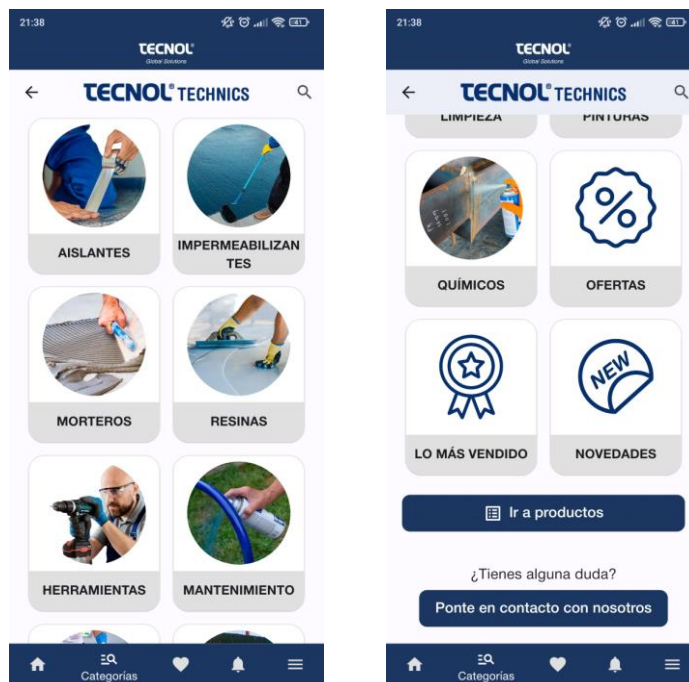


Figura 47. Pantalla de subcategorías

Cuando se accede a una subcategoría se muestran recursivamente sus subcategorías hasta que tenga productos. Entonces se muestra la pantalla de lista de productos, la cual permite ordenar los productos con un desplegable.

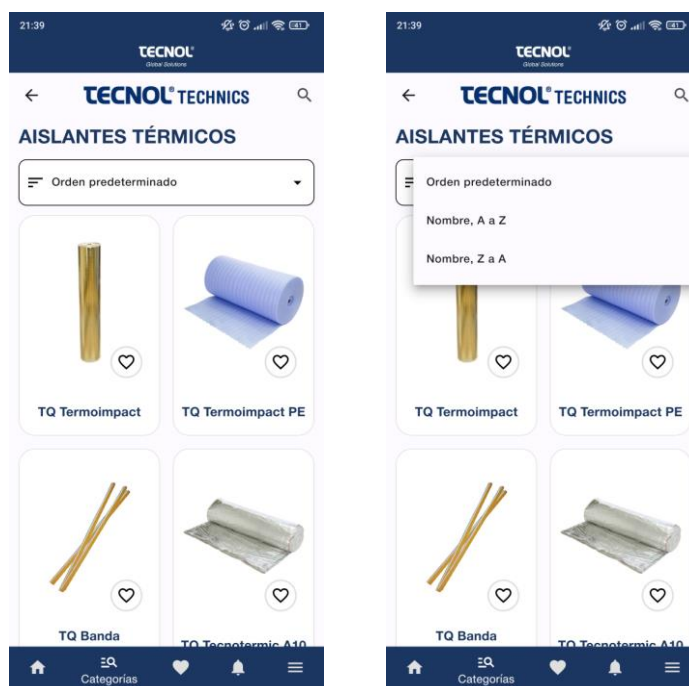


Figura 48. Pantalla de lista de productos y su desplegable

5.6.5 Pantalla de detalles del producto

Si se le da a un producto se accede a la pantalla con sus detalles y opciones en función de si el usuario es invitado, externo o comercial.

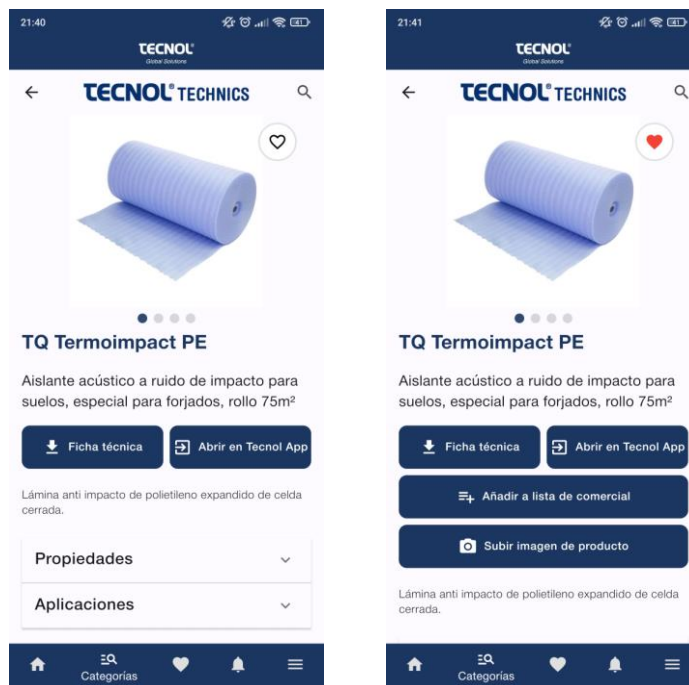


Figura 49. Pantalla de detalles del producto del invitado/externo y comercial

A nivel visual para el invitado y el externo son la misma pantalla, pero el externo puede añadir el producto a favoritos.

Cuando se le da al botón de ficha técnica esta se muestra con la app nativa del dispositivo.



Figura 50. Ficha técnica

El botón de “Abrir en TecnoL App” abre la aplicación de compras de TecnoL

Por otra parte, cuando se accede a la opción de añadir a lista de comercial, aparecen las listas creadas. Si se le da a una de ellas se añade el producto.

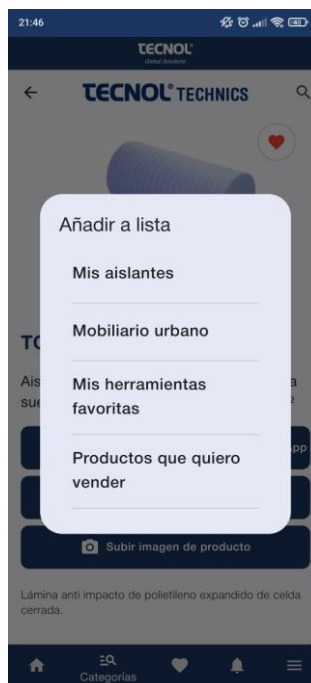


Figura 51. Pantalla para añadir producto a lista

La opción para subir imagen de producto permite tomar una foto o seleccionar de galería.



Figura 52. Pantalla para subir imagen de producto

5.6.6 Pantalla de favoritos

Cuando se navega a la pestaña de favoritos se muestra una lista de productos favoritos si se ha iniciado sesión.

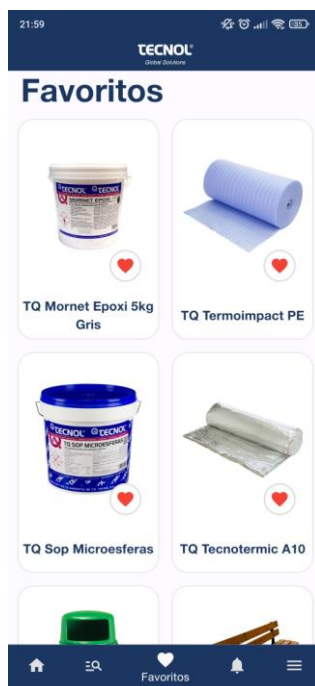


Figura 53. Pantalla de favoritos

5.6.7 Pantalla de notificaciones

A continuación, se muestra la pantalla de notificaciones, siempre que el usuario haya iniciado sesión. Si el usuario es externo solo podrá tener notificaciones de noticias, si el usuario es comercial también podrá tener de cambios de estado de productos. Estas se pueden expandir, contraer y eliminar deslizándolas a un lado.

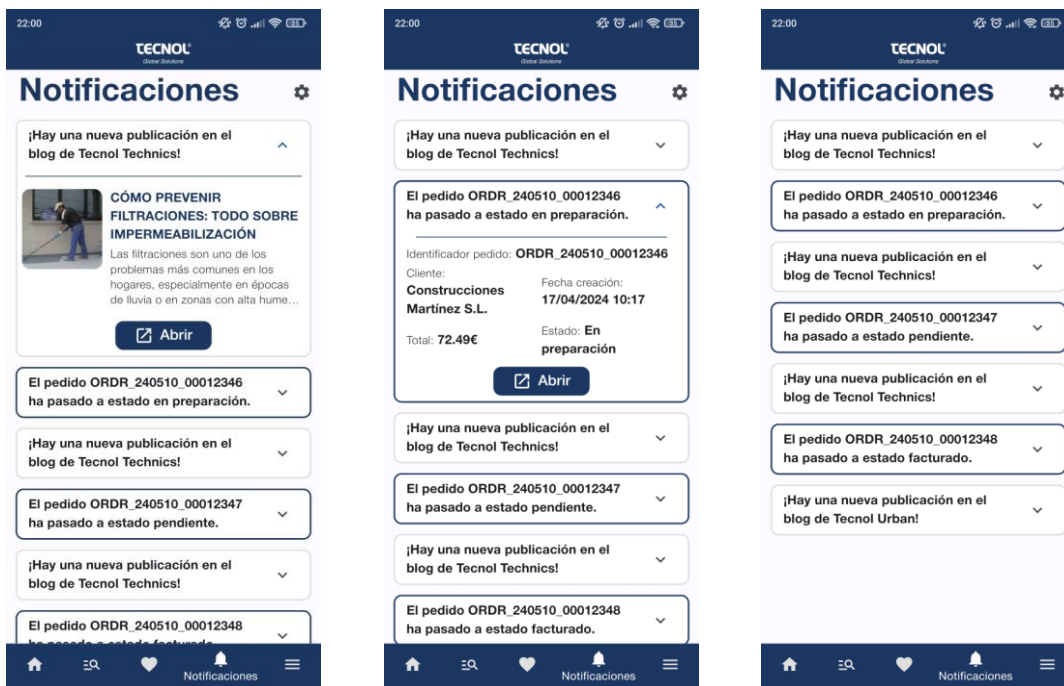


Figura 54. Pantalla de notificaciones

Cuando se elimina una notificación se muestra el siguiente mensaje.

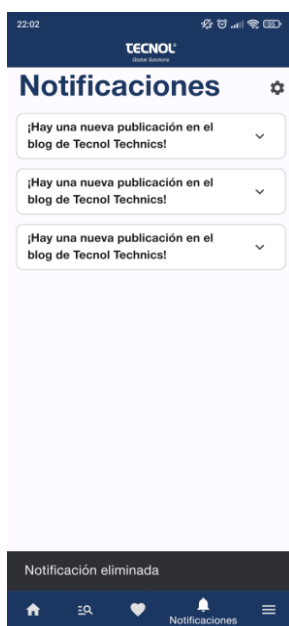


Figura 55. Mensaje de notificación eliminada

Si se accede al icono de configuración se muestra la siguiente pantalla en función de si el usuario es externo o comercial.

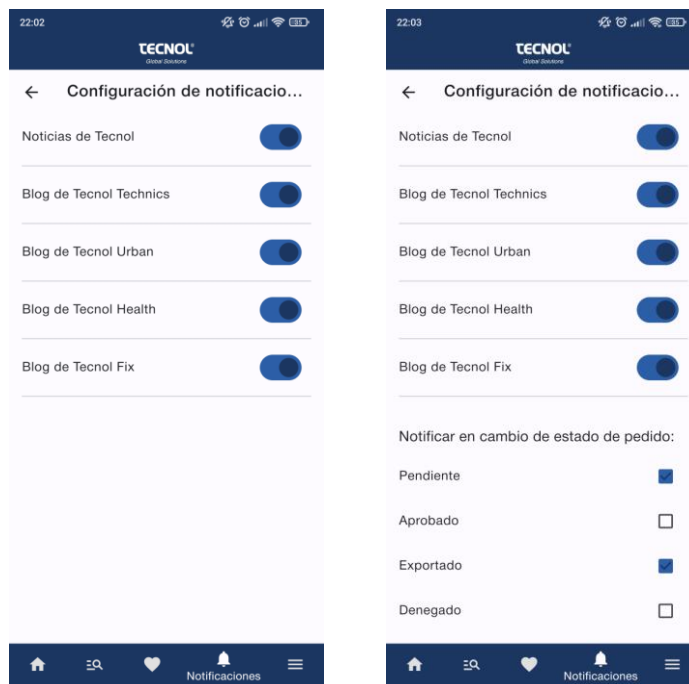


Figura 56. Pantalla de configuración de notificaciones de usuario externo y comercial

5.6.8 Pantalla de cuenta

La pestaña de cuenta es diferente para los usuarios invitado, externo, comercial y administrador.

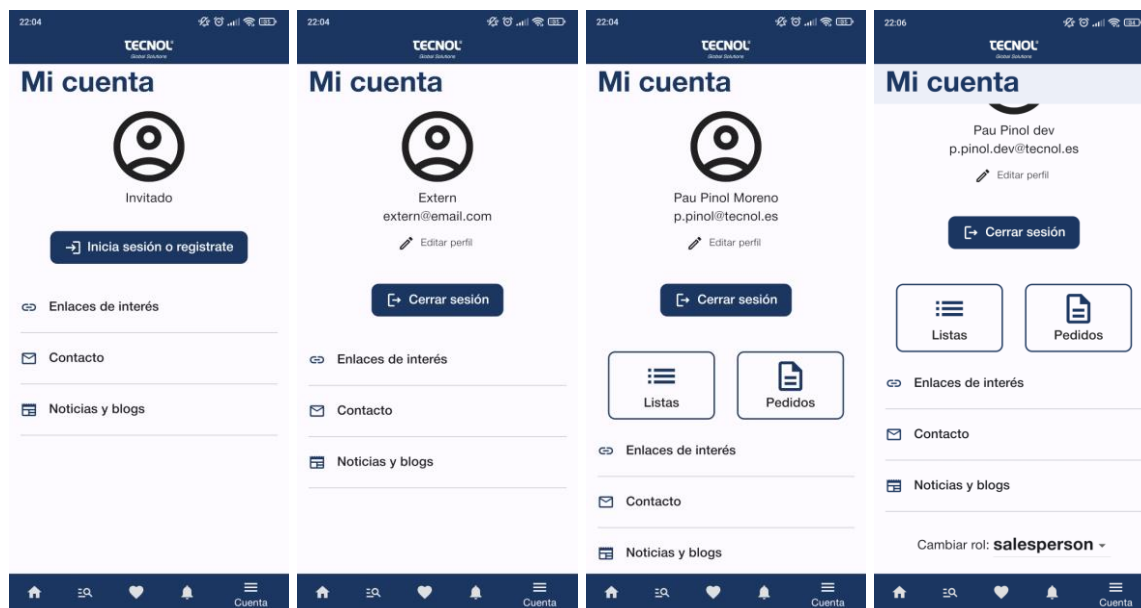


Figura 57. Pantalla de cuenta de invitado, externo, comercial y administrador

La pantalla de cuenta de invitado permite iniciar sesión o registrarse mientras que las de los usuarios loggeados ofrece la opción de cerrar sesión y editar perfil.

El administrador además tiene un selector para cambiar de rol de usuario que le permitirá visualizar la aplicación y acceder a las funciones de esta según el rol seleccionado.

La pantalla de editar perfil es la siguiente.



Figura 58. Pantalla de editar perfil

Cuando se accede a enlaces de interés se muestra la siguiente pantalla. Si se accede a un enlace se muestra su web.

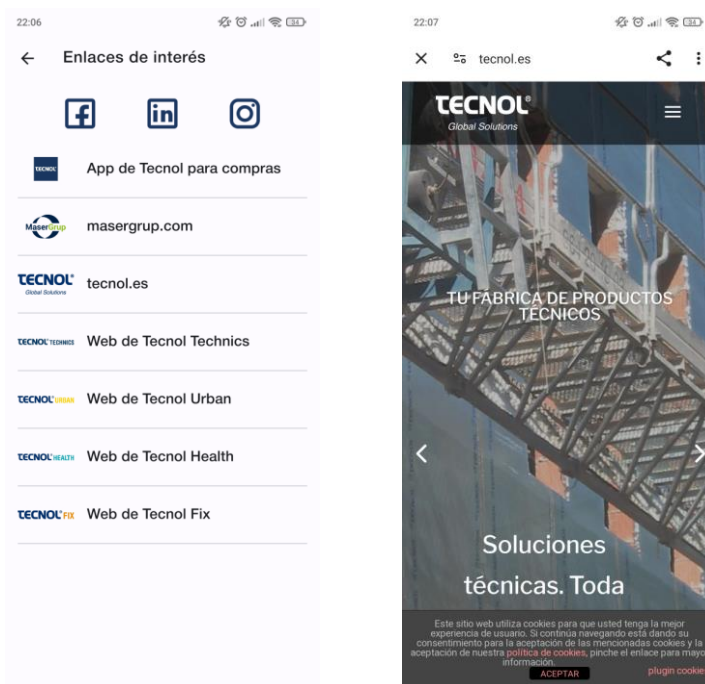


Figura 59. Pantallas de enlaces de interés y web

Los enlaces de interés de un comercial son diferentes.

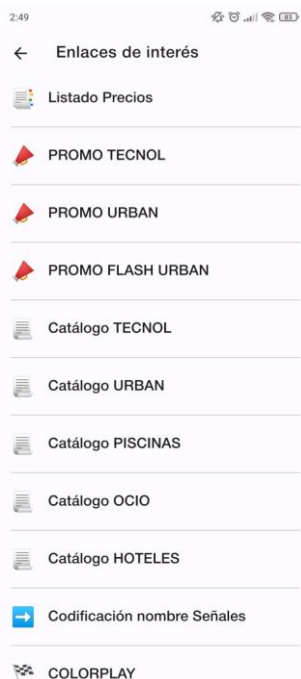


Figura 60. Pantalla de enlaces de interés del comercial

La pantalla del apartado de contacto es la siguiente.

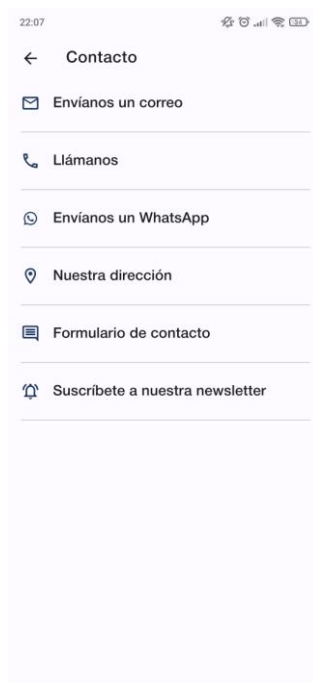


Figura 61. Pantalla de contacto

El formulario de contacto y la pantalla para suscribirse a la newsletter se muestran de la siguiente manera.

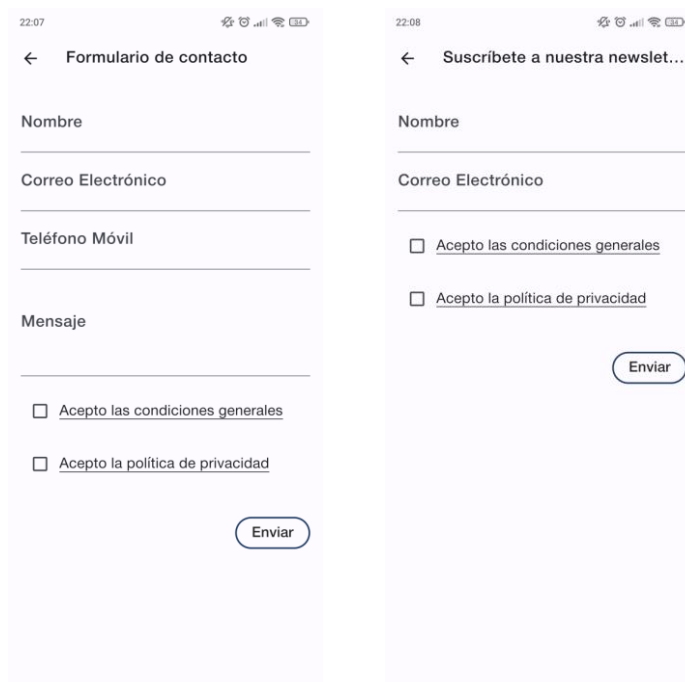


Figura 62. Pantallas de formulario de contacto y suscribirse a la newsletter

Si no se llenan los campos y se le da a enviar, muestran el siguiente error.

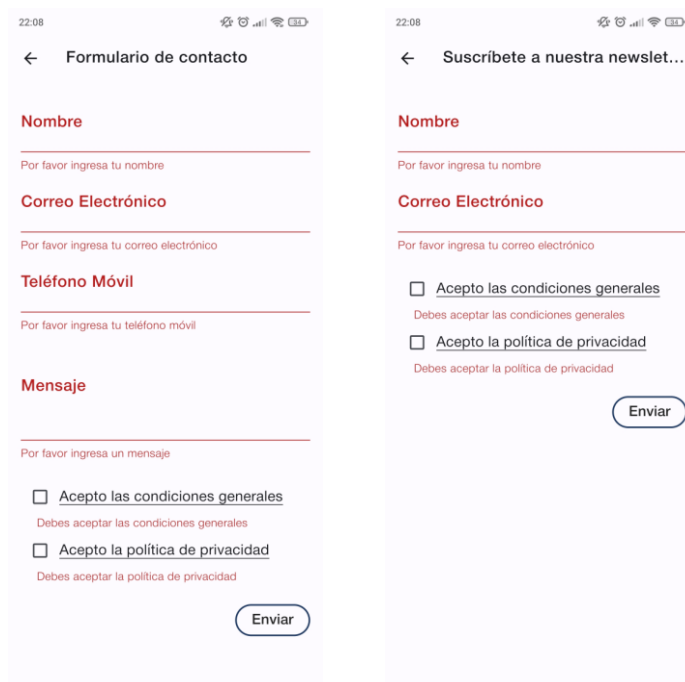


Figura 63. Error campos incompletos en formulario de contacto y newsletter

Por otro lado, el apartado de noticias y blogs, sus noticias y la web de una noticia se muestran de la siguiente manera.

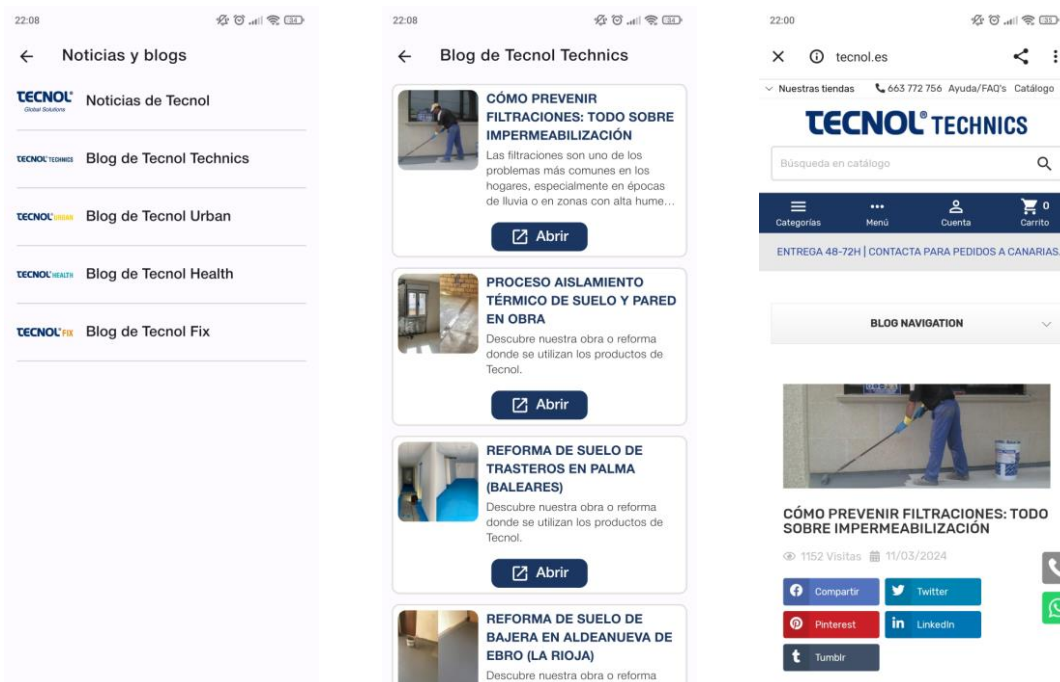


Figura 64. Pantalla de noticias y blogs, lista de noticias y web de una noticia

5.6.9 Pantalla de pedidos

La sección de pedidos en la pestaña de cuenta de un comercial accede a la siguiente pantalla con una lista de pedidos que se pueden expandir y contraer.

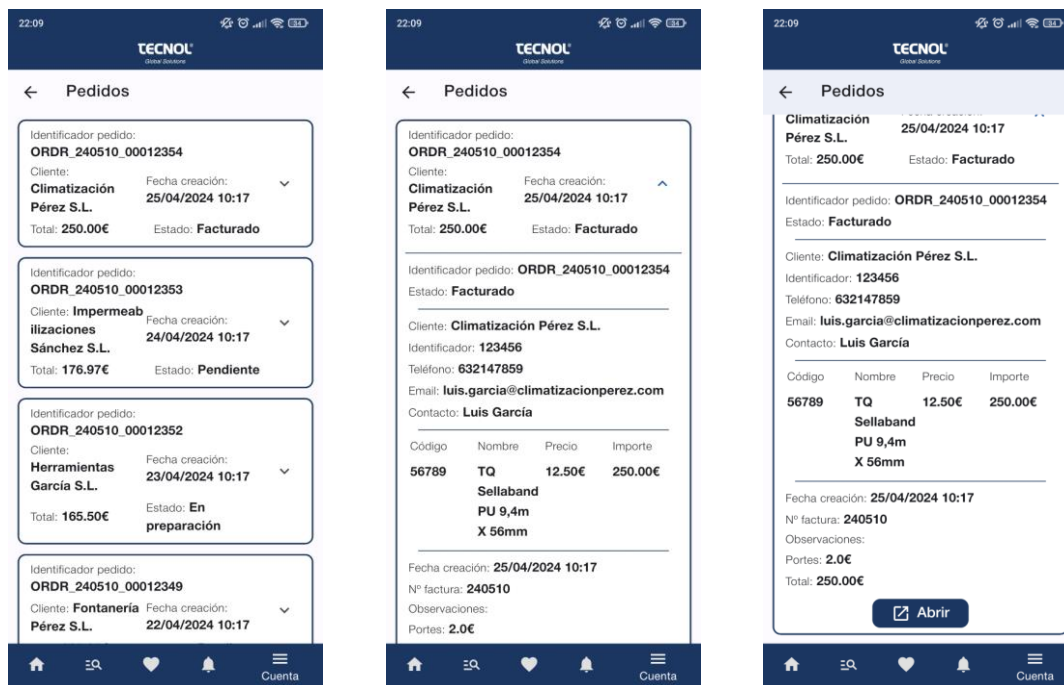


Figura 65. Pantalla de pedidos

5.6.10 Pantalla de listas

Cuando se accede a la sección de listas de la cuenta de un comercial se muestran en primer lugar las listas de productos.

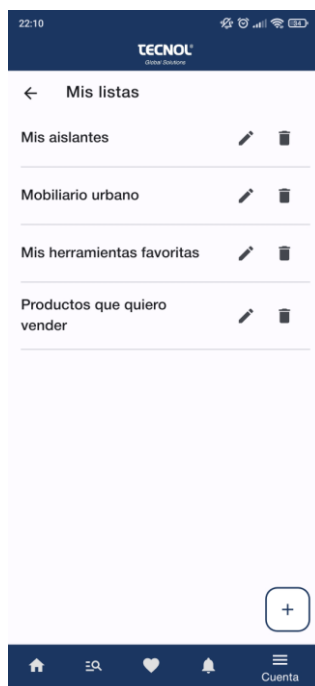


Figura 66. Pantalla de listas de productos

Estas listas se pueden crear (icono más), editar (icono lápiz) y eliminar (icono cubo de basura) a través de los siguientes diálogos.

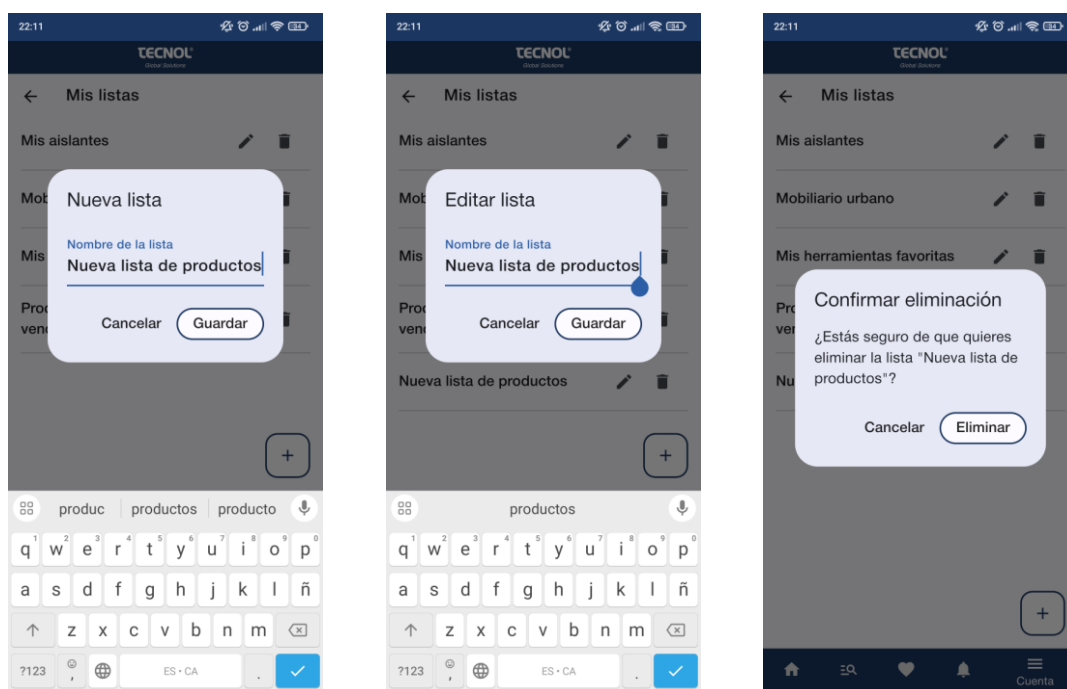


Figura 67. Diálogos para crear, editar y eliminar una lista

Dentro de una lista aparecen sus productos, los cuales se pueden seleccionar y quitar de la lista.



Figura 68. Productos en una lista

Cuando se le da al botón con la flecha se pasa a la pantalla con las listas de clientes. Las listas de clientes también se pueden crear, editar y eliminar como las de productos.

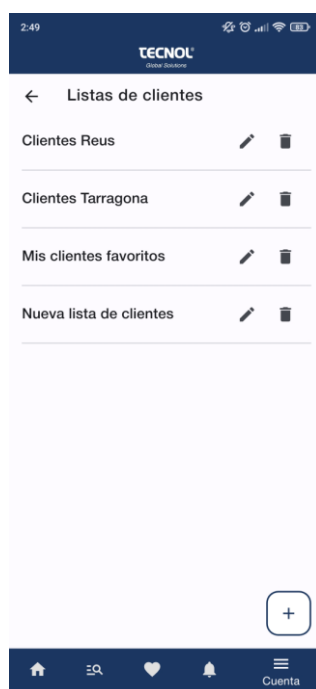


Figura 69. Listas de clientes

Dentro de una lista hay sus respectivos clientes, los cuales se pueden seleccionar. También es posible buscar y añadirlos accediendo al diálogo del icono más.

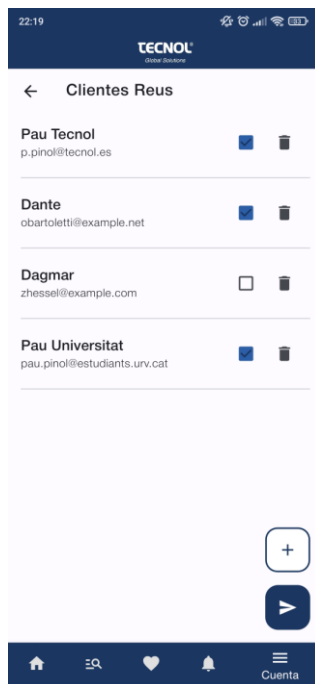


Figura 70. Pantalla de clientes de una lista

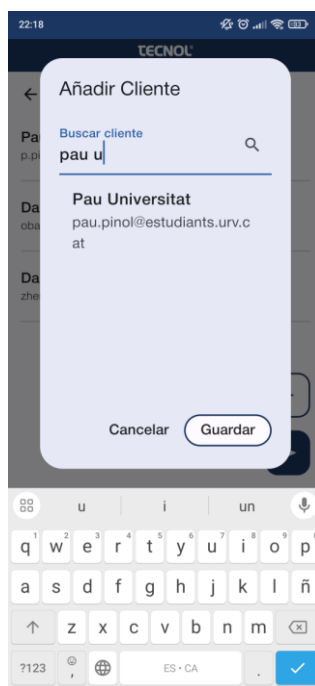


Figura 71. Diálogo para buscar y añadir un cliente

Finalmente, si le damos al icono del avión nos pedirá con qué app de correo queremos enviar los productos seleccionados a los correos de los clientes indicados.

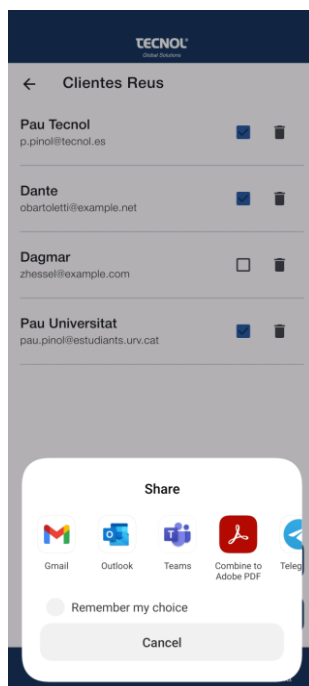


Figura 72. Seleccionar app de correo

Una vez seleccionada la app, se mostrará un nuevo correo con las fichas técnicas adjuntas de los productos seleccionados y los correos de los clientes en copia oculta.



Figura 73. Nuevo correo

5.6.11 Pantallas de inicio de sesión y registro

La pantalla de inicio de sesión se muestra de la siguiente manera. Si se deja un campo vacío, muestra un error.

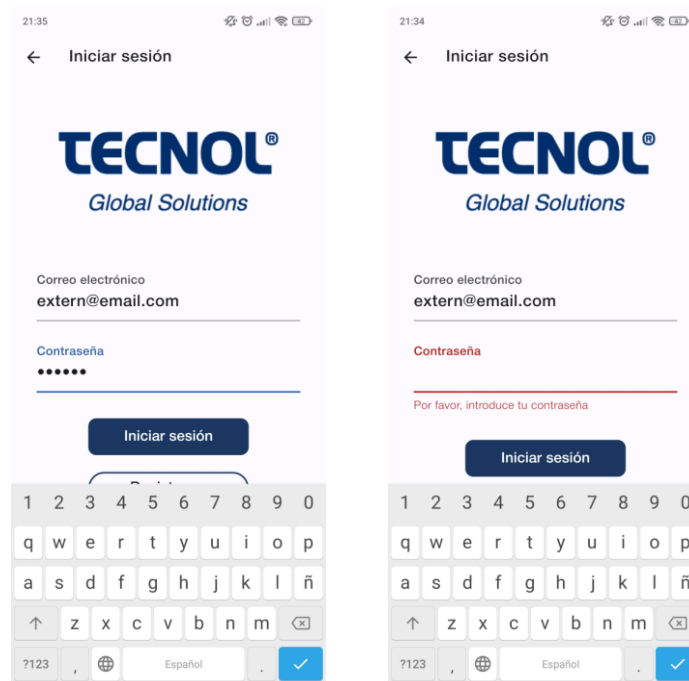


Figura 74. Pantalla de inicio de sesión

Por otro lado, la pantalla de registro es la siguiente.

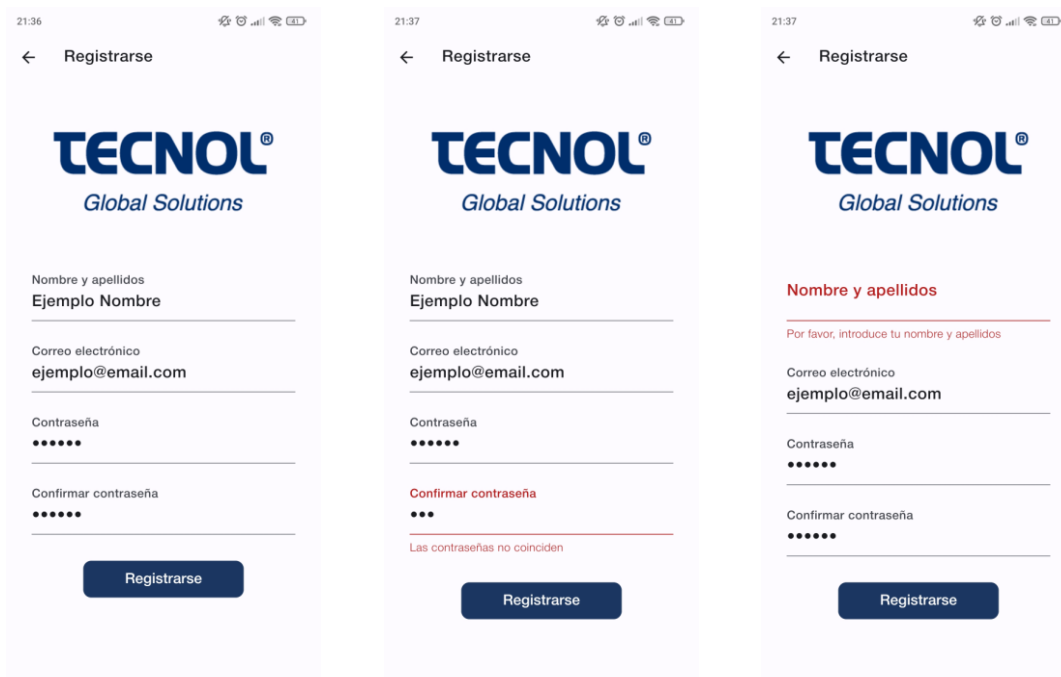


Figura 75. Pantalla de registro

6 Implementación

6.1 Detalles de las Tecnologías Utilizadas

6.1.1 *Front end*

Seguidamente se detallarán los paquetes de Flutter que se han usado y donde se han implementado.

- `carousel_slider` [7]: permite crear un widget de carrusel deslizable con soporte para deslizamiento infinito y widgets personalizados como contenido. Se ha usado para implementar el carousel de la pantalla de detalles del producto.
- `photo_view` [8]: proporciona un widget zoomable sensible a gestos, ideal para mostrar imágenes interactivas. Se ha utilizado para ampliar una imagen de un producto.
- `http` [9]: facilita la realización de solicitudes HTTP. Se ha usado para realizar las peticiones a la API en los archivos del directorio `services`.
- `path_provider` [10]: permite obtener ubicaciones comúnmente usadas en los sistemas de archivos de la plataforma anfitriona, como los directorios de datos de la aplicación y temporales. Se ha empleado para obtener el directorio donde descargar los PDFs¹⁴ de las fichas técnicas de los productos.
- `open_filex` [11]: permite a las aplicaciones Flutter abrir archivos utilizando aplicaciones nativas. Se ha implementado para abrir el PDF de la ficha técnica de un producto con la app nativa del dispositivo
- `url_launcher` [12]: plugin para abrir una URL. Se ha utilizado para abrir webs en la misma aplicación y abrir otras apps externas.
- `image_picker` [13]: permite seleccionar imágenes de la biblioteca de imágenes y tomar nuevas fotos con la cámara. Se ha empleado para subir una imagen de un producto.
- `shared_preferences` [14]: complemento de Flutter para leer y escribir pares clave-valor simples. Se ha usado para almacenar en local el nombre y email del usuario y la configuración de las notificaciones.
- `mobile_scanner` [15]: escáner universal de códigos de barras y códigos QR para Flutter basado en MLKit. Se ha implementado para detectar el código de barras de un producto.
- `intl` [16]: proporciona facilidades de internacionalización y localización, incluyendo traducción de mensajes, plurales y géneros, formateo y análisis de fechas/números, y texto bidireccional. Se ha usado para mostrar la fecha de los pedidos formateada.
- `provider` [17]: es una solución para la gestión del estado en Flutter, que facilita la inyección de dependencias y la notificación de cambios en el estado de la aplicación. Se ha empleado para manejar el estado de los widgets.
- `flutter_email_sender` [18]: permite enviar correos electrónicos desde Flutter utilizando la funcionalidad nativa de la plataforma. Se ha implementado para enviar fichas técnicas a clientes.

¹⁴ *Portable Document Format*, formato de fichero

- `flutter_secure_storage` [19]: proporciona una API para almacenar datos en un almacenamiento seguro. Se ha utilizado para almacenar de forma segura en local el personal access token y el rol del usuario.
- `flutter_widget_from_html` [20]: permite renderizar HTML como widgets en Flutter, soportando enlaces, imágenes, audio, video, iframes y muchas otras etiquetas. Se ha empleado para mostrar las características de los detalles de un producto.
- `flutter_native_splash` [21]: permite generar automáticamente una splash screen personalizada. Se ha usado para generar la splash screen de la aplicación.

6.1.2 Back end

Para gestionar la autenticación y protección de rutas, se ha optado por Laravel Sanctum [22], una herramienta sencilla y flexible que proporciona una autenticación ligera para aplicaciones SPA¹⁵, aplicaciones móviles y APIs basadas en tokens.

Laravel Sanctum es un paquete de Laravel que permite gestionar la autenticación de API mediante tokens de una manera sencilla y eficiente. A diferencia de Laravel Passport, otro paquete de autenticación de Laravel, este implementa el protocolo OAuth2 y es más complejo [23]. Por tanto, se ha decidido emplear Sanctum porque permite una autenticación basada en tokens y la aplicación desarrollada no requiere la complejidad de OAuth2.

En la autenticación de tokens API de Sanctum, se pueden generar tokens de acceso personal para los usuarios. Estos tokens se almacenan en la base de datos y se utilizan para autenticar solicitudes API mediante el encabezado Authorization con el formato Bearer <token>. Para realizar una solicitud autenticada, el cliente debe incluir el token en el encabezado de la solicitud, Sanctum verifica este token y autentica la solicitud si el token es válido. Además, los tokens pueden ser revocados en cualquier momento desde la base de datos, lo que invalida el acceso inmediato del token revocado.

También se pueden asignar habilidades específicas a los tokens generados. Esto permite verificar que una solicitud entrante está autenticada con un token al que se le ha concedido una habilidad determinada. De esta forma se pueden proteger rutas por autenticación, dejando acceso solo a aquellas peticiones que incluyen un token con la habilidad permitida en la ruta. En nuestro caso las habilidades definidas serán los roles.

6.2 Estructura de Directorios y Ficheros de la Aplicación

6.2.1 Directorio assets

Este directorio contiene los *assets* de la aplicación en los subdirectorios *fonts* e *images*.

6.2.1.1 Directorio images

Este directorio contiene imágenes de la aplicación como logos e iconos de la empresa.

¹⁵ Single Page Application

6.2.1.2 Directorio fonts

Almacena las fuentes que se usan en la aplicación, que son las corporativas.

6.2.2 *Directorio lib*

Este directorio contiene el código fuente de la aplicación. Está estructurado en diferentes subdirectorios que lo organizan.

6.2.2.1 Fichero main.dart

Este archivo es el punto de entrada principal de la aplicación. Se encarga de configurar y ejecutar de la app. Aquí se realiza la inicialización de los providers, en concreto se inicializa el AuthProvider en primer lugar y posteriormente los restantes a partir de este para que se sincronicen cuando se actualice [24].

6.2.2.2 Fichero constants.dart

Este fichero contiene las constantes necesarias que se usan en la aplicación.

6.2.2.3 Directorio models

El directorio models es utilizado para definir las clases que representan los datos de la aplicación. Estos modelos son fundamentales para estructurar y manipular la información que la aplicación utilizará y mostrará. Además, las clases proporcionan métodos para convertir datos desde el formato JSON para facilitar la comunicación con la API.

6.2.2.4 Directorio services

Este directorio contiene todos los ficheros con las funciones que permiten la comunicación con los *endpoints* de la API.

6.2.2.4.1 *Fichero auth_service.dart*

Maneja todas las interacciones relacionadas con la autenticación de usuarios. Contiene las funciones para inicializar un usuario invitado, iniciar sesión, registrarse y cerrar sesión.

6.2.2.4.2 *Fichero category_service.dart*

Define los métodos para obtener las divisiones y subcategorías.

6.2.2.4.3 *Fichero notification_service.dart*

Proporciona las funciones para obtener las notificaciones, obtener su información y eliminarlas.

6.2.2.4.4 *Fichero product_service.dart*

Ofrece los procedimientos necesarios para obtener los detalles de un producto, sus fichas técnicas, los productos de una categoría, los productos favoritos y para añadir o quitarlos.

6.2.2.4.5 Fichero *search_service.dart*

Define las funciones para recibir los resultados de búsqueda de productos, clientes o un código de barras.

6.2.2.4.6 Fichero *contact_service.dart*

Proporciona los métodos necesarios para realizar las peticiones de las secciones de contacto.

6.2.2.4.7 Fichero *list_service.dart*

Ofrece los procedimientos para manejar con el back end las listas del comercial.

6.2.2.5 Directorio providers

El directorio providers está destinado a contener todas las clases que gestionan el estado de la aplicación y proveer datos a los widgets. Estas clases utilizan *ChangeNotifier* para notificar a los widgets interesados sobre los cambios en el estado, permitiendo una actualización reactiva.

El principal es el *AuthProvider*, el cual se encarga de manejar el estado relacionado con la autenticación de usuarios. El resto de providers dependen de él para hacer peticiones a la API con el personal access token del usuario a través de los services y para gestionar los datos en función del usuario.

6.2.2.6 Directorio screens

En este directorio se organizan y almacenan las diferentes pantallas de la aplicación. Cada pantalla representa una interfaz de usuario completa y, en conjunto, definen la navegación para al usuario moverse a través de la aplicación.

6.2.2.7 Directorio widgets

El directorio widgets contiene los diferentes componentes reutilizables de la interfaz de usuario. Los widgets facilitan la actualización y mantenimiento de la interfaz de usuario, ya que los cambios en uno se reflejan en todas las partes de la aplicación donde se utiliza ese widget. Además, ofrecen una apariencia y comportamiento coherente en toda la aplicación. Algunos de los widgets definidos son el *filled button* con un icono y una etiqueta pasados por parámetro, el *appbar* con el logo de la empresa y la lupa o la *card* que representa un producto de una lista.

6.3 Algoritmos Específicos más Destacados

En este apartado se explicarán en detalle la implementación de algunas de las características más complejas de la aplicación.

6.3.1 Gestión de favoritos

Analizar la administración de los productos marcados como favoritos nos proporcionará una visión detallada de cómo es la gestión del estado y la obtención de datos del back end en la aplicación.

En primer lugar, cuando se crea el *ProductProvider* se hace la petición para obtener los productos favoritos, siempre que el usuario haya iniciado sesión. El constructor de *ProductProvider* recibe por parámetro el *AuthProvider*, de forma que cuando este último se actualiza porque el usuario ha iniciado o cerrado sesión, también se actualiza el *ProductProvider*.

```
ProductProvider(this._authProvider) {
    if (_authProvider.isUserLoggedIn) fetchFavoriteProducts();
}
```

Código 1. Constructor de ProductProvider

El método para obtener los productos favoritos hace una petición a través del *ProductService* para obtener la información de los productos, los cuales se mostrarán en la pantalla de favoritos. Esta lista de productos favoritos se guarda en el *ProductProvider* para ser consultada. Adicionalmente, se guarda una lista de los ids de productos favoritos para poder gestionarlos localmente de forma más eficaz sin tener que depender de la respuesta del back end cada vez que se añade o quita uno.

```
Future<void> fetchFavoriteProducts() async {
    try {
        _favoriteProducts = await _productService
            .getFavoriteProducts(_authProvider.personalToken!);

        _favoriteProductsIds =
            _favoriteProducts.map((prod) => prod.productId).toList();

        notifyListeners();
    } catch (e) {
        debugPrint('$e');
    }
}
```

Código 2. Método para obtener los productos favoritos en ProductProvider

```
Future<List<ProductCard>> getFavoriteProducts(String
personalToken) async {
    final response = await http.get(
        Uri.parse('$baseUrl/user/favorite_products'),
```

```

        headers: <String, String>{
            'Authorization': 'Bearer $personalToken',
        },
    );

    if (response.statusCode == 200) {
        final data = jsonDecode(response.body);
        final List<dynamic> products = data['data'];
        return products.map((json) =>
ProductCard.fromJson(json)).toList();
    } else {
        throw jsonDecode(response.body)['message'];
    }
}
}

```

Código 3. Método para hacer la petición de obtener los productos favoritos en ProductService

Para añadir o quitar un producto de favoritos se ha definido la función descrita a continuación. En primer lugar, se comprueba si el usuario ha iniciado sesión. En caso afirmativo, si el producto es favorito, este se elimina de la lista de ids de favoritos y se notifica a los *listeners* para que actualicen su estado. De esta forma aquellos widgets que contienen el producto en cuestión volverán a construirse, mostrando que ya no son favoritos. A continuación, se hace la petición a través de *ProductService* para eliminar el producto de favoritos en el back end. El objetivo de separar la gestión de la lista de favoritos local y en el back end, pero manteniendo la consistencia, es conseguir una respuesta prácticamente instantánea al añadir o quitar un producto a favoritos y que se actualicen el estado de los widgets sin tener que depender de la respuesta del back end, que empeoraría la fluidez. En caso de que el producto no sea favorito, se hace el proceso análogo añadiéndolo a la lista de ids de favoritos y haciendo la petición al back end. Finalmente, se llama al método para obtener los productos favoritos para obtener la información de los productos en la pantalla de favoritos.

```

Future<void> toggleFavorite(int productId) async {
    try {
        if (_authProvider.isUserLoggedIn) {
            if (isFavorite(productId)) {
                _favoriteProductsIds.removeWhere((id) => id == productId);
                notifyListeners();

                await _productService.removeFavoriteProduct(
                    _authProvider.personalToken!, productId);
            } else {
                _favoriteProductsIds.add(productId);
            }
        }
    }
}

```

```

        notifyListeners();

        await _productService.addFavoriteProduct(
            _authProvider.personalToken!, productId);
    }
    await fetchFavoriteProducts();
}
} catch (e) {
    debugPrint('$e');
}
}

```

Código 4. Método para añadir o quitar un producto de favoritos en ProductProvider

```

Future<void> addFavoriteProduct(String personalToken, int
productId) async {
    final response = await http.post(
        Uri.parse('$baseUrl/user/favorite_products'),
        headers: <String, String>{
            'Authorization': 'Bearer $personalToken',
        },
        body: jsonEncode(<String, int>{
            'product_id': productId,
        })),
    );

    if (response.statusCode != 200) {
        throw jsonDecode(response.body)['message'];
    }
}

```

Código 5. Método para hacer la petición de añadir un producto a favoritos en ProductService

```

Future<void> removeFavoriteProduct(
    String personalToken, int productId) async {
    final response = await http.delete(
        Uri.parse('$baseUrl/user/favorite_products/$productId'),
        headers: <String, String>{
            'Authorization': 'Bearer $personalToken',
        },
    );
}

```

```

    if (response.statusCode != 200) {
        throw jsonDecode(response.body) ['message'];
    }
}

```

Código 6. Método para hacer la petición de quitar un producto a favoritos en ProductService

```

bool isFavorite(int productId) {
    if (_authProvider.isUserLoggedIn) {
        return _favoriteProductsIds.any((id) => id == productId);
    } else {
        return false;
    }
}

```

Código 7. Método para consultar si un producto es favorito en ProductProvider

Respecto a la pantalla de favoritos, su body es *Consumer* de *AuthProvider* y *ProductProvider* para que su estado se actualice cuando lo notifican estos providers. Esto permite que cuando se inicia sesión se tenga acceso a la información de esta pantalla y que cuando se añade un producto a favoritos se muestren actualizados.

```

...
body: Consumer2<AuthProvider, ProductProvider>(
    builder: (context, authProvider, productProvider, child) {
        if (authProvider.isUserLoggedIn) {
            final favoriteProducts = productProvider.favoriteProducts;
        }
    }
)
...

```

Código 8. Consumer de AuthProvider y ProductProvider en la pantalla de favoritos

Por otra parte, los widgets de los productos también son *Consumer* de *AuthProvider* y *ProductProvider*, lo que permite actualizar su estado para habilitar el botón de favorito si el usuario ha iniciado sesión y mostrar el corazón en rojo o negro en función de si ese producto es favorito o no.

```

...
onPressed: () async {
    try {
        if (authProvider.isUserLoggedIn) {

```

```
        await productProvider
            .toggleFavorite(widget.product.id);
    } else {
        Navigator.of(context, rootNavigator: true).push(
            MaterialPageRoute(
                builder: (context) => const LoginScreen(),
            ),
        );
    }
} catch (error) {
    debugPrint(error.toString());
}
},
child: productProvider.isFavorite(widget.product.id)
    ? const Icon(
        Icons.favorite,
        color: Colors.red,
    )
    : const Icon(
        Icons.favorite_border,
        color: Colors.black,
    ),
...

```

Código 9. Gestión del botón de favorito

7 Avaluación

En este apartado, se detalla la metodología de evaluación llevada a cabo para asegurar que el proyecto cumple con todos los requisitos establecidos. Se han diseñado casos de prueba que abarcan todas las funcionalidades del sistema, realizándose pruebas unitarias en los diferentes módulos y pruebas relacionadas con los requisitos funcionales.

En la siguiente tabla se muestra por cada prueba su descripción, el requisito funcional asociado, el resultado y si este es correcto o no.

Prueba	Descripción	Requisito	Resultado	Correcto
1	Registrarse en el sistema.	00. SignUp	Se registra un nuevo usuario externo y se da acceso a sus funcionalidades.	✓
2	Intentar registrarse sin tener conexión.	00. SignUp	Mostrar mensaje de error.	✓
3	Intentar registrarse con un correo ya registrado.	00. SignUp	Mostrar mensaje de error.	✓
4	Intentar registrarse con un campo del formulario vacío.	00. SignUp	Mostrar mensaje de error.	✓
5	Intentar registrarse sin introducir la misma contraseña en el campo de confirmar contraseña.	00. SignUp	Mostrar mensaje de error.	✓
6	Iniciar sesión en el sistema.	01. LogIn	Se inicia la sesión del usuario en la aplicación y se da acceso a sus funcionalidades según su rol.	✓
7	Intentar iniciar sesión sin tener conexión.	01. LogIn	Mostrar mensaje de error.	✓
8	Intentar iniciar sesión con un campo del formulario vacío.	01. LogIn	Mostrar mensaje de error.	✓
9	Intentar iniciar sesión con un correo o contraseña incorrectos.	01. LogIn	Mostrar mensaje de error.	✓
10	Cerrar la sesión del usuario.	02. LogOut	Se finaliza la sesión del usuario y se muestra la aplicación como invitado.	✓

11	Acceder a las categorías principales (divisiones).	03. Mostrar Categorías	Mostrar las divisiones.	✓
12	Seleccionar una división.	03. Mostrar Categorías	Mostrar las subcategorías.	✓
13	Seleccionar una subcategoría que tiene subcategorías.	03. Mostrar Categorías	Mostrar las subcategorías.	✓
14	Seleccionar una subcategoría que no tiene subcategorías, tiene productos.	03. Mostrar Categorías	Navegar a la pantalla de lista de productos.	✓
15	Seleccionar opción de enviar formulario de contacto asociado a la categoría.	03. Mostrar Categorías	Navegar a la pantalla de formulario de contacto.	✓
16	Seleccionar un producto.	04. Mostrar Lista Productos	Navegar a la pantalla de detalles de producto.	✓
17	Seleccionar la opción de ordenar la lista por orden alfabético creciente.	04. Mostrar Lista Productos	Mostrar la lista de productos ordenada por orden alfabético creciente.	✓
18	Seleccionar la opción de ordenar la lista por orden alfabético decreciente.	04. Mostrar Lista Productos	Mostrar la lista de productos ordenada por orden alfabético decreciente.	✓
19	Seleccionar opción de enviar formulario de contacto asociado a la categoría.	04. Mostrar Lista Productos	Navegar a la pantalla de formulario de contacto.	✓
20	Seleccionar la opción de añadir producto a favoritos.	05. Mostrar Gestionar Producto	Añadir el producto a favoritos.	✓
21	Intentar añadir producto a favoritos sin la sesión iniciada.	05. Mostrar Gestionar Producto	Navegar a la pantalla de inicio de sesión.	✓
22	Seleccionar la opción de abrir ficha técnica.	05. Mostrar Gestionar Producto	Abrir el PDF de la ficha técnica del producto con la app nativa del dispositivo.	✓
23	Seleccionar la opción de abrir la app de compras de Tecnol.	05. Mostrar Gestionar Producto	Abrir la app de compras de Tecnol.	✓

24	Intentar abrir la app de compras de Tecnol sin que esté instalada en el dispositivo	05. Mostrar Gestionar Producto	Abrir la tienda de aplicaciones del dispositivo con la página de la app	✓
25	Selecciona la opción de añadir producto a lista de comercial y seleccionar una lista	05. Mostrar Gestionar Producto	Añadir el producto a la lista de comercial seleccionada.	✓
26	Selecciona la opción de subir imagen del producto y escoger la opción de tomar una foto.	05. Mostrar Gestionar Producto	Subir la imagen al back end.	✓
27	Selecciona la opción de subir imagen del producto y escoger una foto de la galería.	05. Mostrar Gestionar Producto	Subir la imagen al back end.	✓
28	Introducir un texto de consulta.	06. Buscar Producto	Mostrar el nombre de los productos que se ajustan al texto de consulta.	✓
29	Seleccionar un producto de los que se ajustan al texto de consulta.	06. Buscar Producto	Navegar a la pantalla de detalles del producto.	✓
30	Seleccionar la opción de buscar un texto de consulta.	06. Buscar Producto	Navegar a la pantalla de lista de productos y guardar la consulta al historial.	✓
31	Seleccionar una consulta del historial y escoger la opción de buscar.	06. Buscar Producto	Navegar a la pantalla de lista de productos.	✓
32	Escanear el código de barras de un producto.	07. Escanear Producto	Navegar a la pantalla de detalles del producto.	✓
33	Intentar escanear el código de barras de un producto que no existe.	07. Escanear Producto	La pantalla con el escáner permanece activa esperando leer el código de barras de un producto que exista.	✓
34	Acceder a la pantalla de favoritos.	08. Mostrar Favoritos	Mostrar la lista de productos favoritos.	✓
35	Intentar acceder a la pantalla de favoritos sin haber iniciado sesión.	08. Mostrar Favoritos	Informar que hay que iniciar sesión y mostrar un botón para navegar a	✓

			la pantalla de inicio de sesión.	
36	Seleccionar la opción de eliminar un producto de favoritos.	08. Mostrar Favoritos	Quitar el producto de favoritos.	✓
37	Seleccionar un producto.	08. Mostrar Favoritos	Navegar a la pantalla de detalles del producto.	✓
38	Acceder a la pantalla de notificaciones.	09. Mostrar Notificaciones	Mostrar la lista de notificaciones.	✓
39	Intentar acceder a la pantalla de notificaciones sin haber iniciado sesión.	09. Mostrar Notificaciones	Informar que hay que iniciar sesión y mostrar un botón para navegar a la pantalla de inicio de sesión.	✓
40	Expandir una notificación.	09. Mostrar Notificaciones	Mostrar la notificación expandida.	✓
41	Seleccionar la opción de abrir de una notificación expandida.	09. Mostrar Notificaciones	Abrir la web del pedido o noticia.	✓
42	Deslizar una notificación para eliminarla.	09. Mostrar Notificaciones	Eliminar la notificación	✓
43	Selecciona la opción de configurar notificaciones.	09. Mostrar Notificaciones	Navegar a la pantalla de configuración de notificaciones.	✓
44	Activar las notificaciones de noticias de una de las divisiones.	10. Configurar Notificaciones	Guardar las preferencias.	✓
45	Desactivar las notificaciones de noticias de una de las divisiones.	10. Configurar Notificaciones	Guardar las preferencias.	✓
46	Activar las notificaciones en el cambio a un estado de los pedidos.	10. Configurar Notificaciones	Guardar las preferencias.	✓
47	Desactivar las notificaciones en el cambio a un estado de los pedidos.	10. Configurar Notificaciones	Guardar las preferencias.	✓
48	Seleccionar una opción de contacto.	11. Mostrar Contacto	Abrir la app del dispositivo correspondiente.	✓

49	Seleccionar la opción de formulario de contacto.	11. Mostrar Contacto	Navegar a la pantalla de formulario de contacto.	✓
50	Selecciona la opción de suscribirse a la newsletter e introducir los datos.	11. Mostrar Contacto	Registrar la suscripción.	✓
51	Introducir los datos y seleccionar la opción de enviar.	12. Enviar Formulario Contacto	Registrar el formulario en el back end.	✓
52	Intentar enviar el formulario con un campo vacío.	12. Enviar Formulario Contacto	Mostrar mensaje de error.	✓
53	Seleccionar una división.	13. Mostrar Noticias	Mostrar las noticias de esa división.	✓
54	Seleccionar la opción de abrir de una noticia.	13. Mostrar Noticias	Abrir la web de la noticia.	✓
55	Seleccionar un enlace.	14. Mostrar Enlaces Interes	Abrir la web o app del enlace.	✓
56	Expandir un pedido.	15. Mostrar Pedidos	Mostrar el pedido expandido.	✓
57	Seleccionar la opción de abrir.	15. Mostrar Pedidos	Abrir la web del pedido.	✓
58	Seleccionar una lista.	16. Gestionar Listas Comercial	Navegar a la pantalla de gestionar una lista.	✓
59	Seleccionar la opción de eliminar una lista.	16. Gestionar Listas Comercial	Eliminar la lista.	✓
60	Seleccionar la opción de editar una lista.	16. Gestionar Listas Comercial	Mostrar el teclado y el nombre de la lista para editar.	✓
61	Introducir un nuevo nombre para una lista en la opción de editar.	16. Gestionar Listas Comercial	Guardar el nuevo nombre de la lista.	✓
62	Seleccionar la opción de crear una nueva lista e introducir el nombre.	16. Gestionar Listas Comercial	Crear la lista.	✓
63	Seleccionar la opción de añadir un cliente a la lista e introducir el nombre de un cliente.	17. Gestionar Lista Comercial Particular	Mostrar los clientes con ese nombre.	✓

64	Escoger un cliente de los mostrados al seleccionar la opción de añadir un cliente.	17. Gestionar Lista Comercial Particular	Añadir el cliente a la lista.	✓
65	Seleccionar y/o deseleccionar varios productos/clientes.	17. Gestionar Lista Comercial Particular	Quedan seleccionados los productos/clientes elegidos.	✓
66	Seleccionar la opción de enviar productos a clientes.	17. Gestionar Lista Comercial Particular	Abrir la app de correo con los correos de los clientes seleccionados en copia oculta y las fichas técnicas de los productos seleccionados adjuntas.	✓
67	Introducir el nuevo nombre y seleccionar guardar.	18. Editar Perfil	Guardar el nuevo nombre del usuario.	✓
68	Intentar guardar con el campo del nombre vacío.	18. Editar Perfil	Mostrar mensaje de error.	✓
69	Seleccionar un rol.	19. Cambiar Rol	Mostrar la aplicación y dar acceso a las funciones de esta según el rol seleccionado.	✓

Tabla 1. Pruebas realizadas

8 Conclusiones

El desarrollo de este proyecto ha sido una experiencia enriquecedora y desafiante que me ha permitido aplicar y afianzar los conocimientos obtenidos en el Grado de Ingeniería Informática. Además, realizar esta aplicación en Tecnol me ha preparado para trabajar en un entorno real y me ha enseñado a enfrentar desafíos prácticos, gestionar un proyecto de manera efectiva y colaborar con diferentes personas en un contexto empresarial.

Este Trabajo de Fin de Grado ha sido un proyecto integral que ha abarcado múltiples aspectos del desarrollo de software, desde la identificación de necesidades y diseño hasta la implementación y pruebas. A través de este proyecto, he aplicado y ampliado mis habilidades técnicas utilizando Flutter para el desarrollo del front end y Laravel para el back end. La implementación de funcionalidades avanzadas y la integración de estas tecnologías me han permitido enfrentar y superar diversos retos técnicos y de diseño.

La aplicación implementada permite acceder de manera sencilla e intuitiva a los productos, las categorías y otra información de la empresa como noticias y contacto. Además, ofrece funcionalidades como guardar productos en favoritos y mostrar notificaciones. También permite a los comerciales visualizar la información resumida de sus pedidos y compartir fichas técnicas a clientes de manera eficiente. Por tanto, este proyecto satisface las necesidades de la empresa y cumple con los objetivos planteados.

Respecto a posibles mejoras futuras, se podrían considerar las siguientes implementaciones adicionales. En primer lugar, integrar el back end realizado en este proyecto con los datos reales en producción de la empresa con el objetivo de que pueda ser usada en el día a día por usuarios reales. En segundo lugar, aparte de mostrar las notificaciones dentro de la app, implementar notificaciones push que se pudieran activar o desactivar con la pantalla de configuración ya integrada sería de gran utilidad. Finalmente, sería importante publicar la aplicación en la tienda de aplicaciones de Android y iOS para que pueda ser instalada y usada por cualquier usuario de esas plataformas.

9 Referencias

- [1] Tecnol, [En línea]. Available: <https://tecnol.es>. [Último acceso: 18 Mayo 2024].
- [2] Flutter, [En línea]. Available: <https://flutter.dev>. [Último acceso: 2024 Mayo 21].
- [3] Laravel, [En línea]. Available: <https://laravel.com/>. [Último acceso: 22 Mayo 2024].
- [4] F. Rhm, «Understanding MVC Architecture in Flutter: A Comprehensive Guide with Examples,» Medium, 3 Julio 2023. [En línea]. Available: https://medium.com/@Faiz_Rhm/understanding-mvc-architecture-in-flutter-a-comprehensive-guide-with-examples-5d1a372c7eaf. [Último acceso: 22 Mayo 2024].
- [5] A. Kumar, «Understanding MVC Pattern in Flutter: A Comprehensive Guide,» LinkedIn, 7 Abril 2024. [En línea]. Available: <https://www.linkedin.com/pulse/understanding-mvc-pattern-flutter-comprehensive-guide-ashutosh-kumar-r1hnc/>. [Último acceso: 22 Mayo 2024].
- [6] «Figma,» [En línea]. Available: <https://www.figma.com>. [Último acceso: 31 Mayo 2024].
- [7] serenader.me, «carousel_slider,» [En línea]. Available: https://pub.dev/packages/carousel_slider. [Último acceso: 31 Mayo 2024].
- [8] blue-fire.xyz, «photo_view,» [En línea]. Available: https://pub.dev/packages/photo_view. [Último acceso: 31 Mayo 2024].
- [9] dart.dev, «http,» [En línea]. Available: <https://pub.dev/packages/http>. [Último acceso: 31 Mayo 2024].
- [10] flutter.dev, «path_provider,» [En línea]. Available: https://pub.dev/packages/path_provider. [Último acceso: 31 Mayo 2024].
- [11] buildtoapp.com, «open_filex,» [En línea]. Available: https://pub.dev/packages/open_filex. [Último acceso: 31 Mayo 2024].
- [12] flutter.dev, «url_launcher,» [En línea]. Available: https://pub.dev/packages/url_launcher. [Último acceso: 31 Mayo 2024].
- [13] flutter.dev, «image_picker,» [En línea]. Available: https://pub.dev/packages/image_picker. [Último acceso: 31 Mayo 2024].
- [14] flutter.dev, «shared_preferences,» [En línea]. Available: https://pub.dev/packages/shared_preferences. [Último acceso: 31 Mayo 2024].
- [15] steenbakker.dev, «mobile_scanner,» [En línea]. Available: https://pub.dev/packages/mobile_scanner. [Último acceso: 31 Mayo 2024].
- [16] dart.dev, «intl,» [En línea]. Available: <https://pub.dev/packages/intl>. [Último acceso: 31 Mayo 2024].
- [17] dash-overflow.net, «provider,» [En línea]. Available: <https://pub.dev/packages/provider>. [Último acceso: 31 Mayo 2024].
- [18] «flutter_email_sender,» [En línea]. Available: https://pub.dev/packages/flutter_email_sender. [Último acceso: 31 Mayo 2024].
- [19] steenbakker.dev, «flutter_secure_storage,» [En línea]. Available: https://pub.dev/packages/flutter_secure_storage. [Último acceso: 31 Mayo 2024].
- [20] daohoangson.com, «flutter_widget_from_html,» [En línea]. Available: https://pub.dev/packages/flutter_widget_from_html. [Último acceso: 31 Mayo 2024].

- [21] jonhanson.net, «flutter_native_splash,» [En línea]. Available: https://pub.dev/packages/flutter_native_splash. [Último acceso: 6 Junio 2024].
- [22] «Laravel Sanctum,» Laravel, [En línea]. Available: <https://laravel.com/docs/11.x/sanctum>. [Último acceso: 1 Junio 2024].
- [23] «Authentication,» Laravel, [En línea]. Available: <https://laravel.com/docs/11.x/authentication>. [Último acceso: 1 Junio 2024].
- [24] S. Lee, «Injecting a Provider in Provider: ChangeNotifierProxyProvider,» Medium, [En línea]. Available: <https://medium.com/@soojlee0701/injecting-a-provider-in-provider-changenotifierproxyprovider-5707e72be8f8>. [Último acceso: 1 Junio 2024].