



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación móvil para la gestión integral de movimientos de almacén con el ERP Odo

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Samuel Calvo Ballesteros

Tutor: Juan Sánchez Díaz

Tutor externo: Ignacio Hermoso de Mendoza García

Curso 2018-2019

Resumen

Este proyecto se centra en el desarrollo de una aplicación Android integrada con el ERP Odoo que administra toda gestión de un almacén.

La aplicación permite al usuario controlar todos los movimientos que se producen desde que un producto llega al almacén hasta que es enviado. Proporcionando funciones concretas de la gestión de almacén como el control en tiempo real del stock, la creación y seguimiento de inventarios y transferencias tanto internas como externas.

El sistema contempla la posibilidad de la existencia de varios almacenes y permite realizar movimientos entre los mismos y controlar el stock de cada producto en cada uno de ellos.

Para la implementación de la aplicación se ha usado el lenguaje de programación Kotlin, así como XML para las interfaces de usuario.

Este proyecto ha surgido en base a una propuesta de la empresa donde se han realizado las prácticas del grado ya que no existía ninguna aplicación Android que conectara de esta manera con Odoo.

Palabras clave: Android, Kotlin, ERP, Odoo, SGA

Resum

Aquest projecte se centra en el desenvolupament d'una aplicació Android integrada amb l'ERP Odoo que administra tota la gestió d'un magatzem.

L'aplicació permet a l'usuari controlar tots els moviments que es produïxen des que un producte arriba al magatzem fins que és enviat. Proporcionant funcions concretes de la gestió de magatzem com el control en temps real de l'estoc, la creació i seguiment d'inventaris i transferències tant internes com externes.

El sistema també contempla la possibilitat de l'existència de diversos magatzems i permet realitzar moviments entre els mateixos i controlar l'estoc de cada producte en cada un d'ells.

Per a la implementació de l'aplicació s'ha usat el llenguatge de programació Kotlin, així com XML per a les interfícies d'usuari.

Aquest projecte ha sorgit basant-se en una proposta de l'empresa on s'han realitzat les pràctiques del grau ja que no existia cap aplicació Android que connectara d'aquesta manera amb Odoo.

Paraules clau: Android, Kotlin, ERP, Odoo, SGA



Abstract

This project focuses on the development of an Android application integrated with the Odoo ERP that manages all management of a warehouse.

The application allows the user to control all the movements that occur from when a product arrives in the warehouse until it is sent. Providing specific functions of warehouse management as well as real-time control of stock, the creation of inventories and monitoring of internal and external transfers.

The system contemplates the possibility of the existence of several warehouses and allows you to make movements between them and control the stock of each product in each of them.

The Kotlin programming language as well as XML for user interfaces has been used for application implementation.

This project has arisen based on a proposal of the company where the practices of the degree have been carried out since there was no Android application that connected in this way with Odoo.

Keywords: Android, Kotlin, ERP, Odoo, SGA

- **ERP:** *Enterprise Resource Planning*. Software que se encarga de gestionar las operaciones internas de una empresa.
- **SGA:** Software de gestión de almacén. Herramienta que se utiliza para controlar todos los procesos propios de un almacén.
- **Picking:** En el ámbito de la logística, se refiere a la preparación de pedidos, seleccionando distintos productos de uno o más almacenes.
- **Packing:** En el ámbito de la logística, se refiere al embalaje de un pedido.
- **Albarán:** Documento que se genera al realizar cualquier movimiento de almacén.
- **CRM:** *Customer Relationship Management*. Software que permite gestionar todas las interacciones con los clientes.
- **Python:** Lenguaje de programación de scripting orientado a objetos. Es un lenguaje interpretado, es decir, no es necesario compilar el código para ejecutarlo.
- **Patrón observador-suscriptor:** Este patrón de comportamiento permite observar los cambios producidos por un objeto. Cada vez que se produce un cambio en el estado del objeto, se notifica a todos los observadores.
- **REST:** *Representational State Transfer*. Arquitectura utilizada para obtener o enviar datos mediante el protocolo HTTP.
- **POJO:** *Plain Old Java Object*. Instancia de una clase que no extiende ni implementa nada.
- **JSON:** *JavaScript Object Notation*. Formato de intercambio de datos sencillo de leer y escribir.
- **IDE:** *Integrated Development Environment*. Software que ofrece servicios para desarrolladores.
- **APK:** *Android Application Package*. Resultado de compilar el código de una aplicación Android. Mediante este archivo se puede instalar la aplicación en un terminal.
- **AOT:** *Ahead-Of-Time compiler*. Consiste en compilar el código de alto nivel a código máquina nativo para ejecutarlo.
- **JIT:** *Just-In-Time compiler*. El código se traduce mientras se ejecuta la aplicación.
- **API:** *Application Programming Interface*. Interfaz que da acceso a la funcionalidad de un software.

Tabla de contenido

1. Introducción.....	9
1.1 Objetivos.....	9
1.2 Estructura	10
1.3 Relación con los estudios cursados	10
2. Contexto tecnológico	11
2.1 My Odo	12
3. Especificación de requisitos	13
3.1 Diagrama de casos de uso.....	13
3.2 Descripción de casos de uso	14
3.3 Diagrama de clases	20
3.4 Requisitos futuros	20
4. Diseño	21
4.1 Arquitectura	21
4.2 Diseño de la Base de Datos.....	23
4.3 Diseño de la Interfaz de Usuario	23
4.3.1 Material Design.....	23
4.3.2 Inicio de sesión.....	24
4.3.3 Menú lateral.....	25
4.3.4 Listas.....	26
4.3.5 Pestañas	27
4.3.6 Formulario de edición	28
5. Implementación	29
5.1 Herramientas para el desarrollo	29
5.1.1 Android Studio	29
5.1.2 GitLab.....	30
5.2 Kotlin y XML.....	30
5.3 Android	31
5.3.1 Características	31
5.3.2 Arquitectura	31
5.4 Estructura interna del proyecto	33
5.5 PostgreSQL	33

5.6 InkScape.....	34
5.7 Flujos de la aplicación.....	34
6. Manual de usuario.....	39
6.1 Introducción.....	39
6.2 Inicio de sesión.....	39
6.3 Tablero del inventario.....	40
6.4 Menú lateral.....	41
6.5 Catálogo de productos.....	43
6.6 Detalles de producto.....	44
6.7 Editar producto.....	45
6.8 Ajustes de inventario.....	46
6.9 Detalle de inventario.....	47
6.10 Crear inventario y añadir productos.....	48
6.11 Transferencias.....	49
6.12 Detalle y edición de albarán.....	50
7. Conclusiones.....	53
8. Referencias.....	54



Tabla de figuras

Figura 3.1.- Diagrama de casos de uso	13
Figura 3.2.- Diagrama de clases.....	20
Figura 4.1.- Modelo cliente-servidor	21
Figura 4.2.- Diagrama entidad-relación.....	23
Figura 4.3.- Diseño de inicio de sesión	24
Figura 4.4.- Diseño del menú lateral.....	25
Figura 4.5.- Diseño de lista (I)	26
Figura 4.6.- Diseño de lista (II)	26
Figura 4.7.- Diseño de pestañas.....	27
Figura 4.8.- Diseño de formulario de edición	28
Figura 5.1.- Arquitectura de Android.....	32
Figura 5.2.- Diagrama de secuencia del flujo de inicio de sesión.....	35
Figura 5.3.- Diagrama de secuencia del flujo de edición de producto	35
Figura 5.4.- Diagrama de secuencia del flujo de edición y validación de albarán..	36
Figura 5.5.- Diagrama de secuencia del flujo de creación de inventario	37
Figura 6.1.- Vista de inicio de sesión	39
Figura 6.2.- Vista de tablero de inventario	40
Figura 6.3.- Menú lateral (I)	41
Figura 6.4.- Menú lateral (II)	41
Figura 6.5.- Menú lateral (III)	42
Figura 6.6.- Vista de catálogo.....	43
Figura 6.7.- Vista de detalle de producto (I).....	44
Figura 6.8.- Vista de detalle de producto (II).....	44
Figura 6.9.- Vista de edición de producto	45
Figura 6.10.- Selección de tipo de producto	45
Figura 6.11.- Lista de ajustes de inventario	46
Figura 6.12.- Vista de detalle de inventario.....	47
Figura 6.13.- Formulario para añadir productos a un inventario.....	48
Figura 6.14.- Lista de transferencias	49
Figura 6.15.- Vista de detalle de albarán	50
Figura 6.16.- Formulario de edición de albarán	51

1. Introducción

A partir de la necesidad de la empresa donde se han realizado las prácticas del grado, se ha desarrollado un conector para integrar Android con el ERP Odoó llamado *Nucleus*. Esto permite a las empresas que utilizan este servicio, poder gestionar las distintas posibilidades que ofrece desde un dispositivo Android.

En concreto, este TFG se centra en la implementación de una aplicación Android que implemente la funcionalidad del SGA de Odoó usando el conector antes mencionado.

Esta aplicación proporciona las herramientas necesarias para cubrir todos los procesos propios de la gestión de almacenes de una empresa, desde el momento en que el producto llega al almacén, pasando por todos los movimientos que se producen dentro de este, hasta que es enviado a su destino.

También permite controlar en todo momento el stock de productos, así como la ubicación en la que se encuentra cada uno de ellos, incluso realizar movimientos entre los distintos almacenes de la empresa, en caso de existir varios.

Ofrece la posibilidad de realizar un seguimiento de todos los albaranes generados y ver en que estado se encuentran estos. Además, es posible editar estos albaranes para realizar las operaciones de *picking* y *packing*.

1.1 Objetivos

La finalidad de este TFG es proporcionar una herramienta que facilite y agilice el trabajo de la gestión de almacenes en grandes empresas, tanto el de los operarios del almacén como la organización y gestión de productos.

- Facilitar la creación y gestión de inventarios.
- Facilitar la realización de operaciones de *picking* y *packing*.
- Controlar en todo momento el stock de productos.
- Controlar en todo momento la ubicación de cada producto.
- Llevar un seguimiento de los albaranes generados.

1.2 Estructura

Este documento se compone de siete puntos. En el primero, se explica brevemente el problema a resolver y se definen los objetivos que debe cumplir la aplicación. También se hace una pequeña reflexión acerca de la relación que tiene el proyecto con el grado cursado.

En el segundo punto, se introducen las tecnologías que se han utilizado y cuál es su estado actual. También se habla de una aplicación similar a la desarrollada que ha servido como base.

En el tercer punto se definen los requisitos funcionales mediante el uso de distintos diagramas.

El cuarto punto se centra en el diseño, tanto de la interfaz de usuario como del modelo de datos y la arquitectura usadas.

En el quinto punto se explica cómo se ha desarrollado la aplicación, haciendo énfasis en las herramientas que se han utilizado y explicando algunos de los flujos.

En el sexto punto, se explica paso a paso como usar todas las funciones que ofrece la aplicación. Este manual está dirigido a usuarios que tengan su primer contacto con ella.

Para finalizar, se incluye un punto de conclusiones donde se hace una reflexión acerca de los problemas que han surgido y como se han solventado. También se encuentra una bibliografía con referencias a las fuentes consultadas para la obtención de información.

1.3 Relación con los estudios cursados

En este apartado se va a hablar de la aplicación de los conocimientos adquiridos durante el grado para la realización de este proyecto. Ha sido necesario combinar aspectos de muchas asignaturas, a continuación se van a destacar las más significativas.

Por un lado, la más útil ha sido la optativa de cuarto “Desarrollo de aplicaciones para dispositivos móviles”, ya que ha servido para asentar las bases de la programación en Android.

Por otro lado, las asignaturas “Tecnologías de bases de datos” y “Bases de datos y sistemas de la información”, ambas cursadas en tercero, también han aportado conocimientos que han facilitado el desarrollo para obtener una conexión estable de la aplicación con la base de datos de Odoo y para el diseño del modelo de datos.

Por último, para la gestión de las comunicaciones en red con el ERP han sido de gran utilidad asignaturas como “Redes de computadores” y “Redes corporativas”, cursadas en segundo y cuarto, respectivamente.

2. Contexto tecnológico

Odoo es un ERP integrado, de código abierto, enfocado a cubrir todos los aspectos de gestión de una empresa, sea cual sea el sector de esta. Este servicio funciona mediante módulos, los cuales se pueden instalar o desinstalar dependiendo de las necesidades particulares de la empresa en cuestión. Cada uno de estos módulos se centra en una función concreta (gestión de almacén, ventas, compras, CRM...).

Odoo está desarrollado en el lenguaje de programación Python y permite modificar los módulos existentes e incluso crear otros nuevos para así personalizar al máximo el servicio para cada cliente. [1]

Con *Nucleus* se pretende implementar toda esta funcionalidad en un dispositivo Android, usando una aplicación distinta para cada módulo que se podrá instalar o desinstalar al igual que estos.

Para desarrollar este conector y las distintas aplicaciones que lo componen, se ha usado el lenguaje Kotlin.

Kotlin es un lenguaje de programación que ofrece muchas posibilidades en la programación de aplicaciones Android. Es 100% interoperable con Java y permite realizar proyectos mixtos con código Kotlin y Java.

La interfaz de usuario se ha diseñado con lenguaje XML, que es un lenguaje de marcado que también da facilidades para gestionar la información que maneja la aplicación, junto con el editor de interfaces integrado en Android Studio. Todas las interfaces se han diseñado siguiendo las reglas de diseño ofrecidas por Material Design, ya que actualmente se está utilizando en la mayoría de las aplicaciones Android debido a que ofrece una interfaz muy limpia e intuitiva.

2.1 My Odoo

My Odoo es una aplicación Android similar a la desarrollada en este proyecto, ya que también trabaja con Odoo.

En un principio, se usó como base para el desarrollo, tanto a nivel de funcionalidad como de interfaz, pero al profundizar en su funcionalidad, los flujos no se correspondían con el problema que pretendía solucionar *Nucleus* y la interfaz no era intuitiva para un usuario que no estuviera familiarizado con tecnologías móviles.

Ha servido como apoyo inicial para desarrollar *Nucleus*, pero al final se han hecho muchos cambios y mejoras con respecto a *My Odoo*.

3. Especificación de requisitos

En este apartado se van a explicar las funciones que debe cumplir la aplicación para cubrir las necesidades del cliente.

3.1 Diagrama de casos de uso

Con el diagrama de casos de uso se pretende definir que funciones va a proporcionar la aplicación desde el punto de vista de distintos actores.

Se han definido dos roles de usuario más uno genérico que podrá realizar las funciones comunes a ambos. Por un lado, el usuario empleado equivale a la persona que está físicamente en el almacén, puede realizar las operaciones básicas de consulta además de editar cierta información que es necesaria en el día a día de un almacén. Por otro lado, el usuario encargado puede realizar también las operaciones básicas de consulta, pero también tendrá permitido hacer operaciones más orientadas a la gestión, como validaciones o creaciones de inventarios que gestionen el almacén desde fuera.

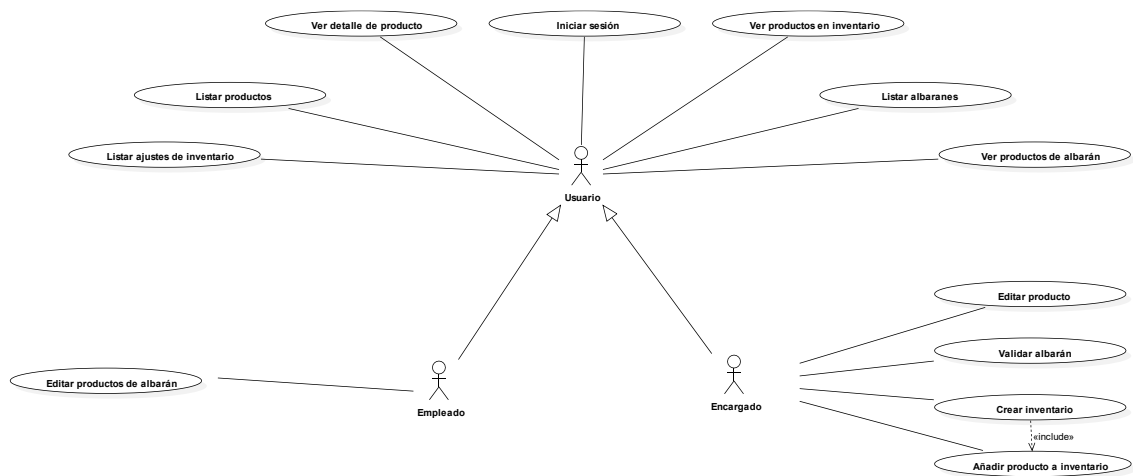


Figura 3.1.- Diagrama de casos de uso

3.2 Descripción de casos de uso

A continuación, se van a explicar con más detalle los casos de uso vistos en la figura 3.1. También se van a mostrar algunos flujos de la aplicación y se va a explicar cómo se comporta en caso de errores.

Caso de uso	Iniciar sesión
Actores	Empleado / Encargado
Propósito	Entrar en la aplicación.
Resumen	El usuario se identifica con su nombre de usuario y contraseña.
Precondiciones	-
Postcondiciones	El usuario queda autenticado.

Intenciones de usuario	Obligaciones del sistema
1. El usuario abre la aplicación.	2. El sistema muestra la pantalla de autenticación y pide el servidor.
3. El usuario introduce el servidor al que conectarse.	4. El sistema comprueba que el servidor existe y se conecta. Pide las credenciales de inicio de sesión.
5. El usuario introduce su nombre de usuario y contraseña y elige la base de datos.	6. El sistema comprueba que los datos son correctos.
	7. El sistema muestra la pantalla principal.

Extensiones síncronas
#1. Si se introduce un servidor inexistente, aparecerá un mensaje de error. (3)
#2. Si se introducen datos erróneos, se mostrará un mensaje de error. (5)

En los siguientes tres casos de uso, la interacción del usuario con el sistema es muy similar así que solo se va a explicar un flujo, concretamente el de “**Visualizar catálogo**”.

Caso de uso	Listar albaranes
Actores	Empleado / Encargado
Propósito	Ver el listado de albaranes.
Resumen	El usuario visualiza una lista de todos los albaranes.
Precondiciones	El usuario ha iniciado sesión.
Postcondiciones	-

Caso de uso	Listar ajustes de inventario
Actores	Empleado / Encargado
Propósito	Ver un listado de los ajustes de inventario.
Resumen	El usuario visualiza una lista de todos los ajustes de inventario.
Precondiciones	El usuario ha iniciado sesión.
Postcondiciones	-

Caso de uso	Visualizar catálogo
Actores	Empleado / Encargado
Propósito	Ver el listado de productos.
Resumen	El usuario visualiza la lista de los productos disponibles para obtener información sobre estos.
Precondiciones	El usuario ha iniciado sesión.
Postcondiciones	-

Interacciones de usuario	Obligaciones del sistema
1. El usuario selecciona en el menú la opción “Productos”.	2. El sistema consulta en la base de datos todos los productos.
	3. El sistema muestra la lista de productos.

En esta interacción no existe ningún posible caso de error ya que la operación que se realiza es simplemente una consulta a la base de datos.

En los siguientes tres casos de uso, la interacción del usuario con la aplicación también es muy similar, así que solo se explicará el flujo de “**Ver productos de albarán**”.

Caso de uso	Ver detalles de producto
Actores	Empleado / Encargado
Propósito	Ver información detallada de un producto.
Resumen	El usuario selecciona un producto de la lista para obtener más información de un producto en concreto.
Precondiciones	El usuario ha iniciado sesión. El producto existe.
Postcondiciones	-

Caso de uso	Ver productos en inventario
Actores	Empleado / Encargado
Propósito	Ver información detallada sobre los productos de un inventario.
Resumen	El usuario selecciona un ajuste de inventario para ver el listado de productos que contiene.
Precondiciones	El usuario ha iniciado sesión. El ajuste de inventario existe.
Postcondiciones	-

Caso de uso	Ver productos de albarán
Actores	Empleado / Encargado
Propósito	Ver información detallada de un albarán.
Resumen	El usuario selecciona un albarán para obtener más información sobre el mismo.
Precondiciones	El usuario ha iniciado sesión. El albarán existe.
Postcondiciones	-

Interacciones de usuario	Obligaciones del sistema
1. El usuario selecciona un albarán.	2. El sistema hace una búsqueda en la base de datos de los productos que contiene dicho albarán.
	3. El sistema muestra una lista con los productos pedidos.

En esta interacción no existe ningún posible caso de error ya que la operación que se realiza es simplemente una consulta a la base de datos.

En los siguientes dos casos de uso, la interacción del usuario con la aplicación también es muy similar, así que solo se explicará el flujo de “**Editar productos de albarán**”.

Caso de uso	Editar producto
Actores	Encargado
Propósito	Modificar atributos del producto.
Resumen	El usuario selecciona un producto y modifica los atributos que desee.
Precondiciones	El usuario ha iniciado sesión. El producto existe.
Postcondiciones	Los datos del producto quedan modificados.

Caso de uso	Editar productos de albarán
Actores	Empleado
Propósito	Modificar los valores de un producto de un albarán.
Resumen	El usuario compara las cantidades reales del producto con las teóricas y cambia dichos valores si es necesario.
Precondiciones	El usuario ha iniciado sesión. El albarán existe. El producto existe dentro del albarán seleccionado.
Postcondiciones	Los valores del producto quedan actualizados.

Interacciones de usuario	Obligaciones del sistema
1. El usuario pulsa el botón editar de uno de los productos que hay en el albarán.	2. El sistema muestra la vista de formulario de edición.
3. El usuario edita la cantidad real.	4. El sistema actualiza la cantidad mostrada.
5. El usuario confirma los cambios.	6. El sistema graba los cambios en la base de datos.

Extensiones síncronas
#1. Si el valor introducido para la cantidad no es correcto, se pondrá a 0.0 por defecto. (3)

Caso de uso	Validar albarán
Actores	Encargado
Propósito	Confirmar un albarán.
Resumen	El usuario comprueba que los datos del albarán son correctos y lo confirma al cliente.
Precondiciones	El usuario ha iniciado sesión. El albarán existe.
Postcondiciones	El albarán queda confirmado.

Interacciones de usuario	Obligaciones del sistema
1. El usuario selecciona un albarán de la lista.	2. El sistema muestra el contenido del albarán.
3. El usuario comprueba que los datos del albarán son correctos.	
4. El usuario pulsa el botón validar.	5. El sistema pasa el albarán a estado "Hecho" .

En este caso no se contemplan casos de error ya que básicamente es un cambio de estado.

Caso de uso	Crear ajuste de inventario
Actores	Encargado
Propósito	Agrupar productos.
Resumen	El usuario crea un inventario para agrupar ciertos productos.
Precondiciones	El usuario ha iniciado sesión.
Postcondiciones	El inventario se crea.
Incluye	Añadir producto a inventario.

Interacciones de usuario	Obligaciones del sistema
1. El usuario pulsa el botón de crear inventario.	2. El sistema muestra el formulario para añadir productos.
3. El usuario introduce los productos y las cantidades de cada uno.	4. El sistema añade cada producto a una lista que actúa de vista previa.
5. El usuario pulsa el botón aceptar.	6. El sistema pide introducir el nombre del inventario.
7. El usuario introduce el nombre y pulsa el botón aceptar.	8. El sistema crea el inventario.

Extensiones síncronas
#1. Si no se introduce ningún producto, se mostrará un mensaje de error. (5)
#2. Si no se introduce un nombre para el inventario, se mostrará un mensaje de error. (7)

Caso de uso	Añadir producto a inventario
Actores	Encargado
Propósito	Añadir un producto a un inventario.
Resumen	El usuario selecciona un producto, la ubicación e introduce la cantidad para añadir una nueva línea al inventario.
Precondiciones	El usuario ha iniciado sesión. Existe el inventario. Existe el producto. Existe la ubicación.
Postcondiciones	Se añade el producto al inventario.

Interacciones de usuario	Obligaciones del sistema
1. El usuario selecciona el inventario al que desea añadir un producto.	2. El sistema muestra el contenido de ese inventario.
3. El usuario pulsa el botón de añadir producto.	4. El sistema muestra el formulario correspondiente.
5. El usuario introduce el nombre y la cantidad del producto.	6. El sistema añade cada producto a una lista que actúa de vista previa.
7. El usuario pulsa el botón aceptar.	8. El sistema añade el producto al inventario.

Extensiones síncronas
#1. Si el inventario se encuentra en estado validado no aparecerá el botón para añadir productos. (3)
#2. Si no se han introducido todos los datos para el producto, aparecerá un mensaje de error. (7)

3.3 Diagrama de clases

En este apartado se van a explicar las entidades que se han usado para conectar con cada modelo de Odoo. Para ello, se va a usar un diagrama de clases.

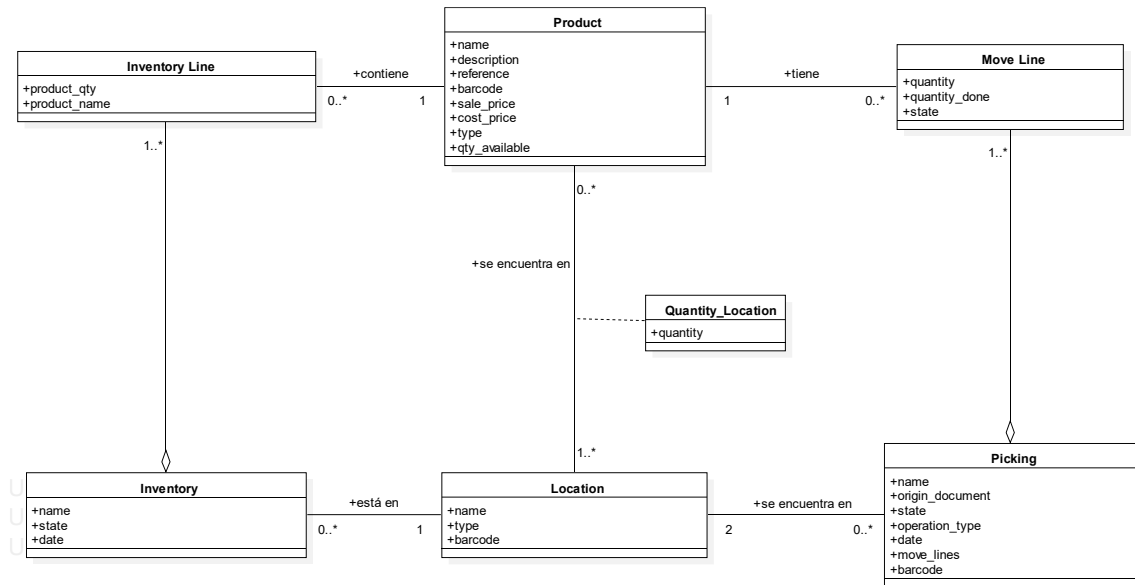


Figura 3.2.- Diagrama de clases

En la figura 3.2, se puede observar como un inventario, se encuentra en una ubicación física y está compuesto por líneas de inventario que, a su vez, contienen un producto cada uno. Lo mismo pasa en el caso de las transferencias (*Picking*), aunque en este caso contiene dos ubicaciones, ya que es necesario especificar el origen y el destino.

Por otro lado, una ubicación puede tener varios inventarios y transferencias, así como productos, los cuales pueden estar divididos en distintas ubicaciones. Para representar esta relación se ha usado una clase de asociación que indica la cantidad de cada producto en cada ubicación en la que se encuentra.

3.4 Requisitos futuros

En un futuro sería interesante implementar la siguiente funcionalidad:

- Poder registrar nuevos usuarios y editarlos desde la aplicación.
- Permitir cambiar de base de datos sin necesidad de cerrar sesión.
- Añadir más opciones de configuración desde dentro de la aplicación.
- Crear nuevos productos desde la aplicación.

4. Diseño

En este apartado se introduce la arquitectura usada por Odoon, así como la seguida para el desarrollo de la aplicación y como se ha logrado conectar Android con el ERP. También se explica el modelo de datos mediante un diagrama entidad-relación.

Por último, se explica, con el uso de capturas, el diseño de la interfaz de usuario y los patrones de diseño que se han seguido.

4.1 Arquitectura

La arquitectura usada por Odoon es el modelo cliente-servidor, el cual consiste en la repartición de tareas. El procesamiento de la información se distribuye de forma cooperativa, donde los clientes solicitan recursos a un servidor que se encarga de proporcionárselos.

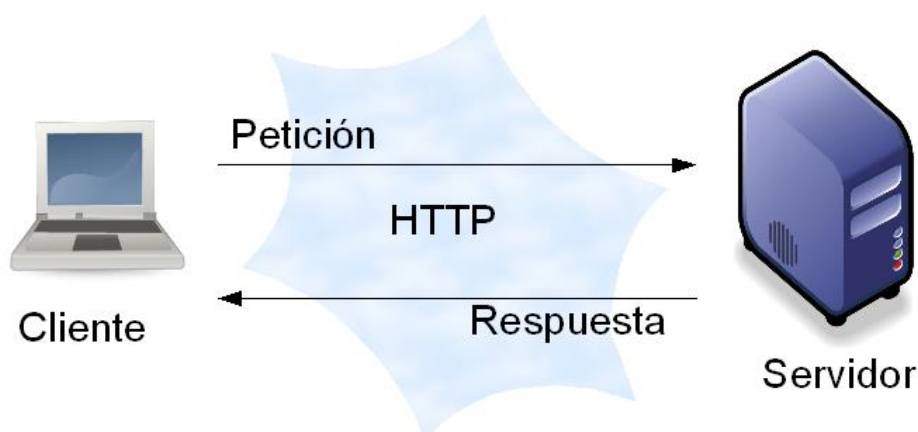


Figura 4.1.- Modelo cliente-servidor

Para gestionar la comunicación con Odoon desde la aplicación, se han utilizado diferentes librerías:

- RxJava

Implementación de ReactiveX en Java, es una API que usa el patrón observador-suscriptor. [2]

El observador (clase Observable) emite datos y el suscriptor o suscriptores (clase Subscriber) interactúa con estos datos.

- Retrofit
Cliente REST que permite hacer peticiones HTTP, gestionar los parámetros y convertir la respuesta a un objeto POJO.
- OkHttp
Cliente HTTP para Android y Java. Facilita sobre todo la descarga de archivos multimedia ya que aporta algunas características como agrupación de conexiones, compresión de archivos o cacheado de respuestas para evitar cargas en peticiones repetidas entre otras.
- Glide
Junto con OkHttp, se ha usado esta librería para administrar los archivos multimedia. Glide permite capturar, decodificar y visualizar imágenes, videos y GIF's. Se ha obtenido un resultado muy eficiente en la gestión de multimedia con el uso estas dos librerías.
- Gson
Librería desarrollada por Google para tratar archivos JSON. Proporciona métodos útiles y sencillos de usar para convertir cualquier tipo de objeto a JSON y viceversa.
- Data Binding
Librería de Android para vinculación de datos. Permite vincular los elementos de la interfaz gráfica a los datos de la aplicación de una forma más sencilla y sin tener que usar otros métodos para hacer referencia a los componentes de la interfaz. [3]

Para acceder a los métodos específicos de Odoo, se ha utilizado un objeto *singleton* que los implementa.

El patrón de diseño *singleton* consiste en tener una sola instancia de una clase que permita acceder a ella. Para implementar esto, hay que restringir la creación de instancias de la clase, imponiendo un constructor privado y un método estático para obtener la instancia. [4]

4.2 Diseño de la Base de Datos

Odoo utiliza una base de datos PostgreSQL. La estructura de las tablas y las relaciones entre ellas se muestra en el siguiente diagrama.

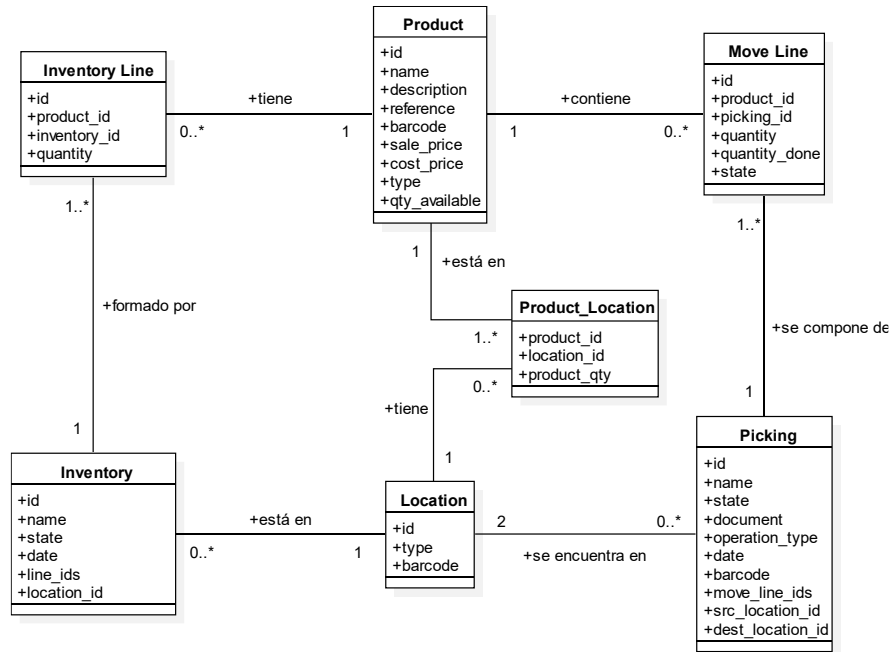


Figura 4.2.- Diagrama entidad-relación

Como se puede ver en el diagrama, se ha definido como clave primaria un identificador único para cada tabla. Y como claves ajenas todos los atributos con el sufijo *_id*. Por ejemplo, en la tabla "Inventory Line", el atributo "inventory_id" hace referencia al inventario al que pertenece esta línea.

4.3 Diseño de la Interfaz de Usuario

4.3.1 Material Design

Material Design es un conjunto de normas y patrones de diseño creado por Google con el objetivo de unificar el aspecto de las interfaces. Las principales características que ofrece son un diseño limpio, con efectos de sombras y el uso de animaciones, entre otras técnicas. [5]

El objetivo principal de Material Design es conseguir que el usuario capte fácilmente la información que aparece en la pantalla y que las interfaces sean intuitivas e interactivas.

Estas normas de diseño empezaron a usarse en la versión 5.0 de Android.

4.3.2 Inicio de sesión

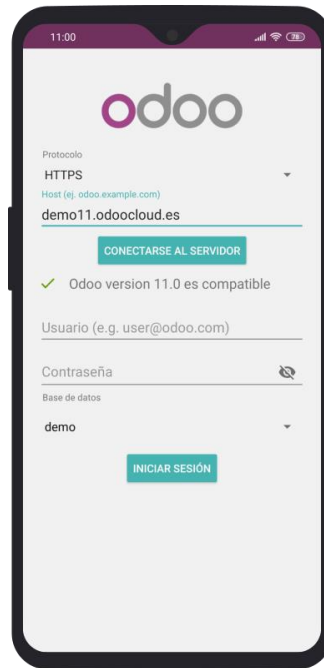


Figura 4.3.- Diseño de inicio de sesión

Al iniciar la aplicación, si no se ha iniciado sesión antes, aparece un formulario para elegir el servidor al que conectarse e introducir el usuario y la contraseña.

También permite seleccionar la base de datos que se desee, de entre las disponibles en el servidor escogido, mediante una lista desplegable que muestra las opciones.

4.3.3 Menú lateral

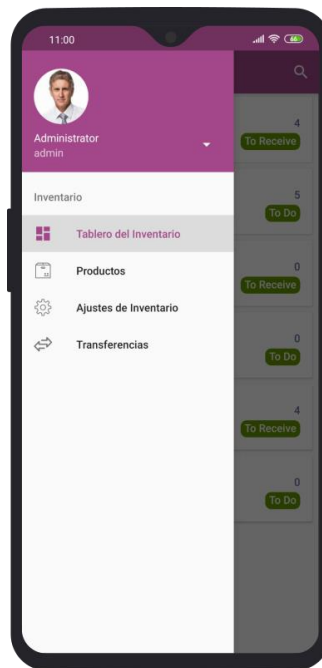


Figura 4.4.- Diseño del menú lateral

Para navegar entre las distintas pantallas de la aplicación, se ha optado por usar un menú desplegable lateral en el que se ha introducido una lista con las distintas opciones principales.

En la parte superior se puede observar información general del usuario conectado actualmente. También se encuentra un botón que cambiará la parte inferior del menú y mostrará otras opciones, como los ajustes de la aplicación o el perfil de usuario con más datos sobre este.

4.3.4 Listas

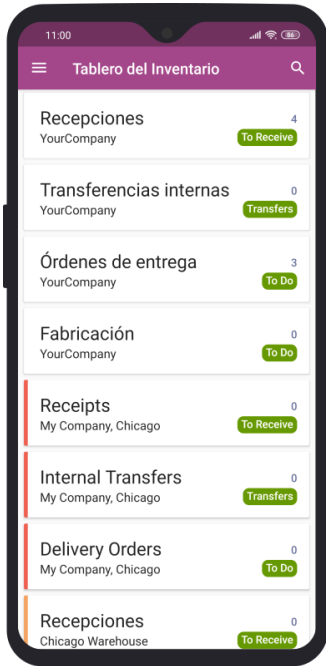


Figura 4.5.- Diseño de lista (I)

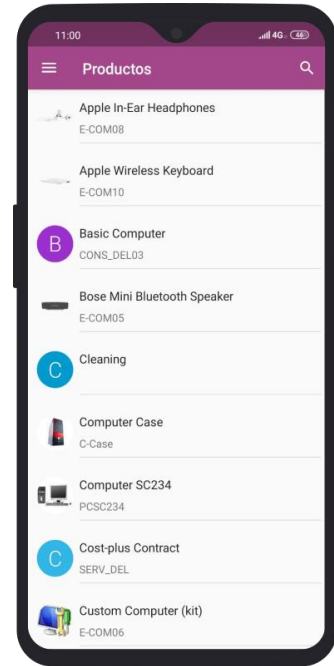


Figura 4.6.- Diseño de lista (II)

En la aplicación se encuentran dos diseños de lista distintos, por un lado, una lista formada por “cartas”, un elemento muy característico de *Material Design* y que ha servido para dar más visibilidad a cada elemento. Además, se ha añadido un código de colores para facilitar aún más la búsqueda al usuario. Este tipo de lista solo se ha usado en la vista de tablero, mostrada en la figura 4.5.

Por otro lado, una lista convencional, con o sin imágenes en los elementos y con separadores entre cada uno de ellos. Este tipo de lista se puede ver en la figura 4.6, que muestra el listado de productos, aunque este tipo de lista también se ha usado en las listas de ajustes de inventario y transferencias.

4.3.5 Pestañas



Figura 4.7.- Diseño de pestañas

En la figura 4.7, se puede ver la pantalla de detalle de producto, en este caso se ha optado por dividir la información en dos pestañas para facilitar al usuario la visualización de la información. Así se evita mezclar datos que no tienen relación.

4.3.6 Formulario de edición

The image shows a smartphone screen displaying a mobile application interface for editing product details. The screen is titled 'Detalles de producto' and shows the following information:

Producto	
[LAP-E5] Laptop E5023	
Demanda inicial	
3.0	
Hecho	Reservado
0.0	3.0
Ubicación de origen	Reservado
WH/Stock/Shelf 2	3.0
Ubicación de destino	Hecho
Partner Locations/Customers	0.0

Figura 4.8.- Diseño de formulario de edición

En la figura 4.8, se observa el diseño de un formulario de edición, en este caso es para un producto contenido en un albarán, pero se ha usado el mismo formato para editar los datos de un producto.

Se compone de distintos campos que pueden ser rellenados automáticamente o que permiten seleccionar distintas opciones desde un diálogo. En todos estos formularios se incluyen dos botones en la barra superior para confirmar o descartar los cambios.

5. Implementación

5.1 Herramientas para el desarrollo

En este apartado se va a explicar el software usado para el desarrollo de la aplicación, así como las razones por las que se han decidido usar dichos programas.

5.1.1 Android Studio

Android Studio es un IDE basado en IntelliJ IDEA para el desarrollo de aplicaciones Android. Se ha escogido este entorno de desarrollo por ofrecer herramientas muy potentes además de ser el editor oficial para el desarrollo en Android.

Este editor ofrece la posibilidad de programar tanto en Java como en Kotlin, pudiendo crear clases en ambos lenguajes dentro de la misma aplicación. Además, ofrece la opción de traducir código Kotlin a código Java y viceversa.

Otra de las opciones que ofrece es la posibilidad de desarrollar aplicaciones para cualquier tipo de dispositivo Android, desde móviles o *tabletas* hasta coches con este SO, pasando por televisores y relojes.

Android Studio ofrece un emulador integrado en el que instalar las APK y poder probar a la vez que se va desarrollando. Permite interactuar con el igual que si fuera un terminal físico. Además, ofrece muchas opciones de configuración hardware y software para adaptarlo a las necesidades del usuario.

El sistema de compilación de Android Studio consiste en compilar el código y los recursos de la aplicación en un APK instalable. Para esto utiliza *Gradle*, un paquete de herramientas de compilación avanzadas, para automatizar y administrar este proceso. Se ejecuta independientemente de Android Studio, por lo que se puede compilar desde el mismo editor o, por ejemplo, desde la línea de comandos. [6]

Por último, el entorno ofrece un editor visual muy potente y completo, dando la opción de diseñar las interfaces de usuario tanto usando directamente código XML como desde su editor gráfico en el que se puede ver una visión global del estado actual de la interfaz mediante una vista de árbol de los componentes que se encuentran actualmente en la interfaz. Además, da facilidad a la hora de añadir nuevos componentes permitiendo arrastrarlos al lugar deseado y editar sus propiedades desde el mismo editor.

5.1.2 GitLab

Para el control de versiones se ha optado por usar GitLab, un servicio basado en Git y de software libre, lo cual permite instalarlo en un servidor privado. En este caso, se ha usado el servidor de la empresa donde se han realizado las prácticas del grado.

Este software permite gestionar fácilmente repositorios y, además, ofrece la posibilidad de crear wikis.

5.2 Kotlin y XML

Kotlin es un lenguaje de programación estáticamente tipado, es decir, las variables no tienen un tipo definido por el desarrollador, sino que es la máquina virtual quien se encarga de ello.

Este lenguaje corre sobre la máquina virtual de Java, por lo que el rendimiento de ambos lenguajes es muy similar. Además, son interoperables entre ellos, lo cual permite tener clases Java y Kotlin sin provocar conflictos. También ofrece compatibilidad con todas las librerías disponibles para Java. [7]

Se ha elegido Kotlin para desarrollar la aplicación por algunas de las ventajas que ofrece frente a Java. Como, por ejemplo, la simplicidad a la hora de escribir código, ya que Kotlin es un lenguaje mucho más conciso en ese sentido. También trata internamente los errores por valores nulos, ya que, para todos los tipos, existe una versión anulable, simplemente añadiendo un signo de interrogación después del tipo, esto permite que esa variable tome el valor nulo evitando así que se lancen excepciones por errores de este tipo. [8]

Otra de las ventajas que han sido muy útiles en la realización de este proyecto, han sido las clases de datos, que se han utilizado para crear todas las entidades necesarias en la aplicación. Estas clases realizan internamente todo lo necesario, como la creación de *getters* y *setters*, constructores y más métodos básicos de una clase Java.

Por otro lado, para el diseño de las interfaces de usuario, además del editor gráfico de Android Studio ya mencionado, se ha utilizado el lenguaje de marcado XML, ya que para el uso de algunos componentes es más útil modificarlos directamente desde el código.

5.3 Android

Android es un sistema operativo diseñado principalmente para dispositivos con pantalla táctil. Está basado en el Kernel de Linux y, por tanto, es un sistema operativo libre y gratuito.

5.3.1 Características

En este apartado se van a describir brevemente las principales características del sistema operativo Android: [9]

- **Compatibilidad con distintos tamaños de pantalla y resoluciones:** Android realiza escalamiento y modificación de tamaños para adaptar las interfaces de usuario a diferentes pantallas.
- **Almacenamiento con SQLite:** Android incluye una librería con la que poder almacenar los datos de la aplicación en una base de datos SQLite.
- **Soporte de Java:** El código Java no se ejecuta, sino que se compila a un lenguaje intermedio que más adelante es interpretado por el terminal.
- **Soporte de hardware:** Android soporta distintos tipos de hardware en los dispositivos, como GPS, cámaras, algunos sensores, etc.
- **Google Play:** Android incluye una tienda de aplicaciones, tanto gratuitas como de pago que se pueden instalar en el dispositivo.
- **Multitarea:** Android permite ejecutar varias aplicaciones al mismo tiempo, gestionando las que no están en primer plano mediante ciclos de reloj.

5.3.2 Arquitectura

En la figura 5.1 se pueden ver los distintos componentes del sistema operativo Android. A continuación, se van a explicar con más detalle: [10] [11]

- **Kernel de Linux:** La base de Android para servicios del sistema como la seguridad, la gestión de memoria y la gestión de procesos entre otros.
- **Capa de abstracción de hardware:** Gestiona las capacidades hardware del dispositivo. Consiste en varias bibliotecas que implementan interfaces específicas para cada componente hardware.
- **Tiempo de ejecución de Android (ART):** A partir de la versión 5.0 de Android, cada aplicación ejecuta sus propios procesos. El ART ejecuta varias máquinas virtuales en los dispositivos. Las principales funciones del ART son la compilación *ahead-of-time* (AOT) y *just-in-time* (JIT), la recolección de elementos no usados y la optimización de la depuración.
- **Bibliotecas C/C++ nativas:** Android proporciona la API del framework de Java para exponer la funcionalidad de estas bibliotecas nativas a las aplicaciones.

- **Framework de la API de Java:** En esta capa se encuentran todas las funciones que ofrece Android, escritas en lenguaje Java. Estas API son la base para desarrollar aplicaciones. Entre ellas se incluye:
 - Sistema de vista para compilar las interfaces de usuario.
 - Administrador de notificaciones que permite mostrar alertas personalizadas.
 - Administrador de actividad para controlar el ciclo de vida de las aplicaciones.
 - Proveedores de contenido que permiten que las aplicaciones se comuniquen entre si compartiendo datos.
- **Aplicaciones del sistema:** Aquí se encuentran las aplicaciones básicas del sistema, como el correo electrónico, aplicaciones de mensajería, calendario o navegador, entre muchas otras.

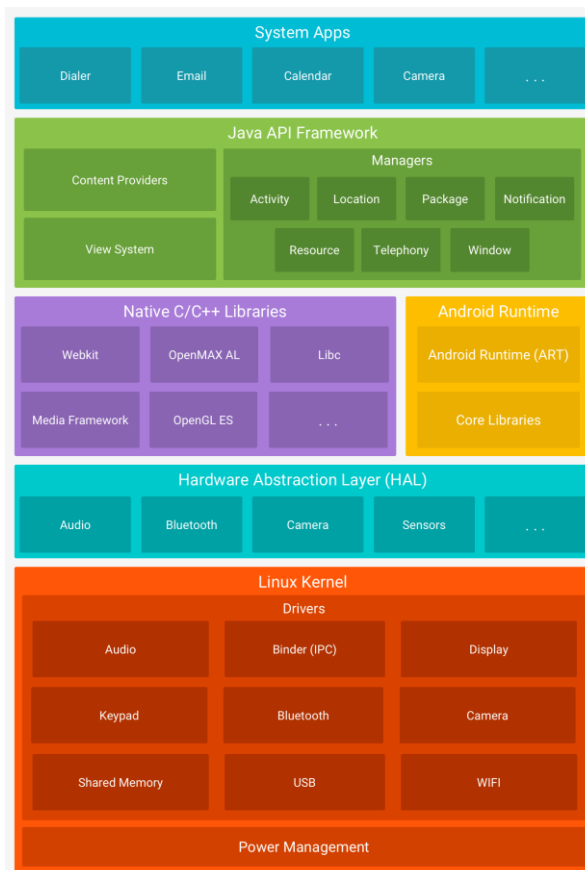


Figura 5.1.- Arquitectura de Android

5.4 Estructura interna del proyecto

En este apartado se va a explicar cómo se ha estructurado internamente la aplicación, pero para ello, primero se van a explicar algunos de los componentes básicos que ofrece Android.

- **Actividad:** Componente que contiene una pantalla con la que el usuario puede interactuar. Dentro de este componente se pueden introducir otros, como botones, listas, campos de texto, entre otros.
- **Fragmento:** Componente que representa una parte de la interfaz en una actividad. Se pueden combinar varios fragmentos en la misma actividad. Son independientes a la actividad, ya que tienen su propio ciclo de vida y se pueden añadir, eliminar o intercambiar sin necesidad de modificar la actividad en la que se encuentran.
- **Adaptador:** Enlace entre la interfaz de usuario y la funcionalidad implementada en el código.

La aplicación se compone de una actividad principal sobre la que se desarrollan todas las acciones. Cada una de las pantallas que componen la aplicación están diseñadas en distintos fragmentos que van reemplazándose sobre la misma actividad.

También se han usado actividades adicionales cuando, por ejemplo, ha sido necesario seleccionar un objeto para obtener algún dato de este. La interfaz que muestra la lista de estos objetos está implementada en una actividad a parte para facilitar la comunicación.

En cuanto a los adaptadores, se han usado principalmente para rellenar las listas que hay en la aplicación y también para implementar el menú desplegable lateral. Aunque Android proporciona adaptadores predefinidos, se han creado otros personalizados para poder implementar la funcionalidad deseada.

5.5 PostgreSQL

Servicio de base de datos relacional de código abierto y gratuito. Se ha optado por este servicio ya que el ERP sobre el que se basa la aplicación lo utiliza y la integración es mucho más sencilla.

Entre las principales características de PostgreSQL se encuentran: [12]

- **Alta concurrencia:** Permite distintos accesos a la misma tabla sin implantar bloqueos, dando una visión consistente a cada usuario.
- **Variedad de tipos:** Soporta muchos tipos en los campos de sus tablas. También ofrece la opción de crear tipos de datos personalizados.
- **Claves ajenas:** Relaciones entre las distintas tablas.
- **Disparadores:** Acciones que se ejecutan cuando se produce un evento determinado.



- **ACID:** Cumple las propiedades de atomicidad, consistencia, aislamiento y durabilidad.
- **Extensiones:** Ofrecidas por PostgreSQL o por terceros. Permiten ampliar la funcionalidad.

5.6 InkScape

Este software de edición de gráficos vectoriales de código abierto se ha utilizado para diseñar los distintos logos que se encuentran dentro de la aplicación y el icono de esta.

Se ha decidido usar esta herramienta dado que ya se tenía un conocimiento básico de uso. Además de porque ofrece herramientas muy potentes para la edición de logos e iconos.

5.7 Flujos de la aplicación

En este apartado se van a explicar con más detalle algunos de los flujos de la aplicación desde el punto de vista de las interacciones que se realizan mediante el uso de diagramas de secuencia.

Antes de pasar a los diagramas, se va a introducir la sintaxis usada para que resulte más fácil de comprender:

- **Usuario:** El usuario que está realizando la acción.
- **IU:** Interfaz de Usuario. La interacción por parte del usuario se realiza directamente con este elemento.
- **:OC:** Objeto de Control. Intermediario entre la IU y las bases de datos.
- **Base de Datos Odoo:** Contiene los servidores existentes entre otros datos, no es única en cada servidor.
- **Base de Datos Servidor:** Contiene los datos que están en el servidor. Es única en el servidor al que se conecta el usuario.

En primer lugar, se va a mostrar la interacción que se realiza cuando un usuario quiere iniciar sesión en la aplicación.

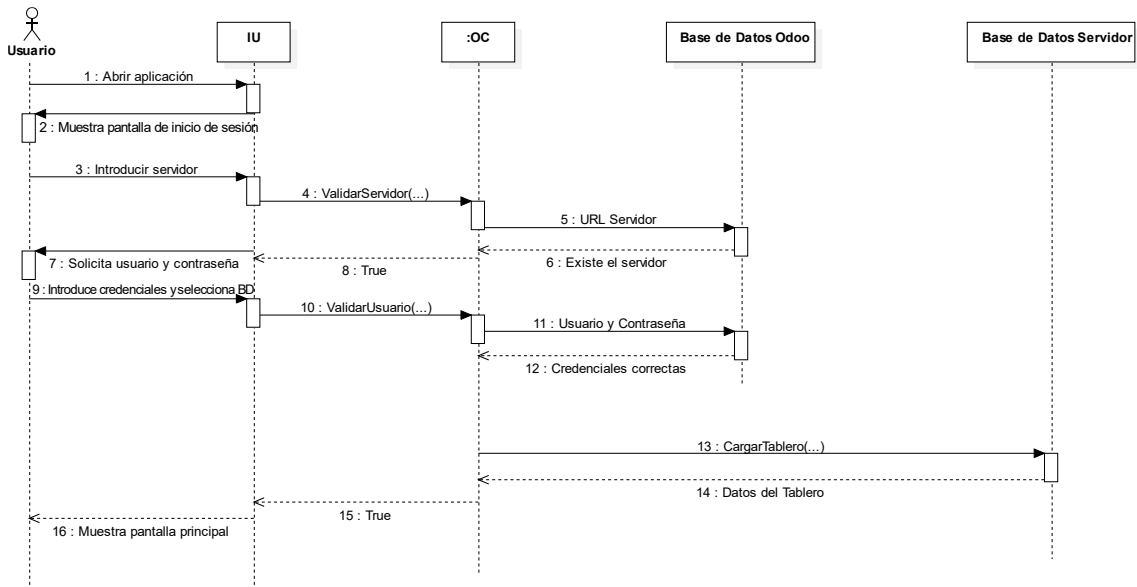


Figura 5.2.- Diagrama de secuencia del flujo de inicio de sesión

En la figura 5.2 se observa que primero el usuario debe introducir el servidor al que desea conectarse y, tras comprobar en la base de datos de Odoo que dicho servidor existe y es compatible, se introducen las credenciales de acceso y se selecciona la base de datos del servidor.

En la siguiente imagen, se va a mostrar el flujo que se produce cuando el usuario desea editar algún campo de un producto.

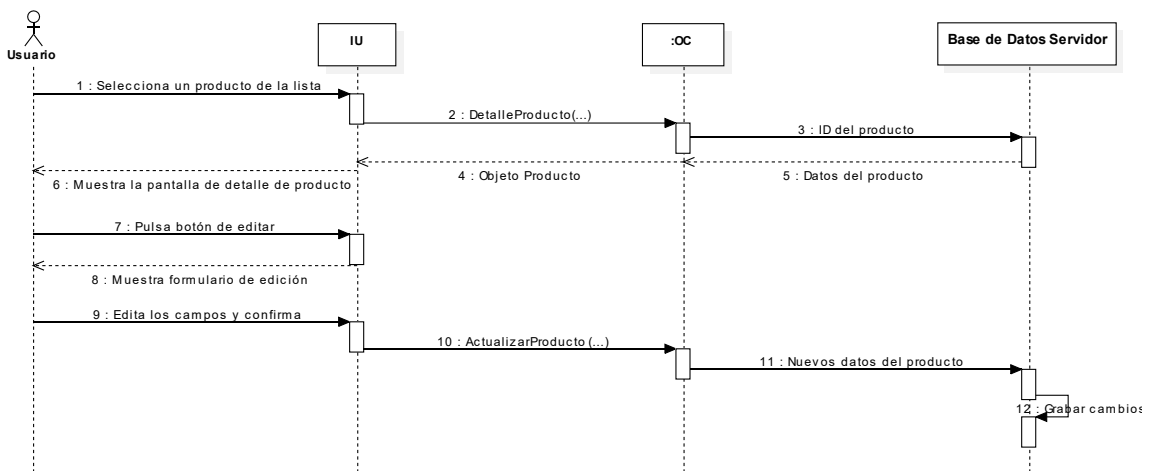


Figura 5.3.- Diagrama de secuencia del flujo de edición de producto



En la figura 5.3 se observa que el usuario debe acceder al producto, el sistema devuelve los datos de este y, después de esto, el usuario modifica los campos que desee. Al guardar los cambios, estos se actualizan en la base de datos.

Para conseguir los datos del producto que se desea modificar, se hace uso del identificador que posee, siendo así más sencillo encontrarlo en la base de datos y mostrarlo al usuario.

En el siguiente diagrama se muestra una interacción del usuario en la que este desea editar un producto de un albarán y después validarlo.

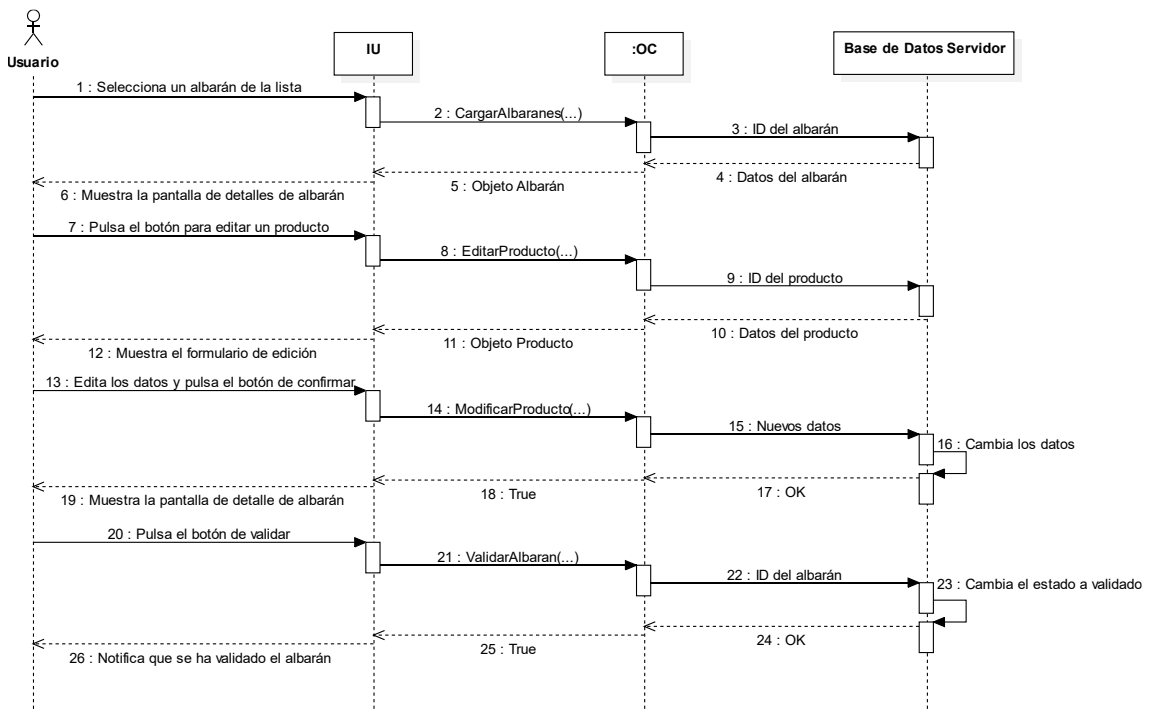


Figura 5.4.- Diagrama de secuencia del flujo de edición y validación de albarán

Como se puede ver en el diagrama, el usuario debe seleccionar el albarán que desea modificar, para después ver los datos de este y editar el producto que desea. Para eso hace uso del formulario de edición correspondiente.

Para validar el albarán, el usuario simplemente debe pulsar el botón de validar y el sistema se encarga de cambiar el estado de dicho albarán a “Validado”.

Para hacer referencia al albarán seleccionado se hace uso del identificador único de este, así en la base de datos se puede encontrar fácilmente y agilizar la visualización de los datos en pantalla.

Por último, se va a mostrar el flujo de creación de inventarios y de cómo añadir productos a este.

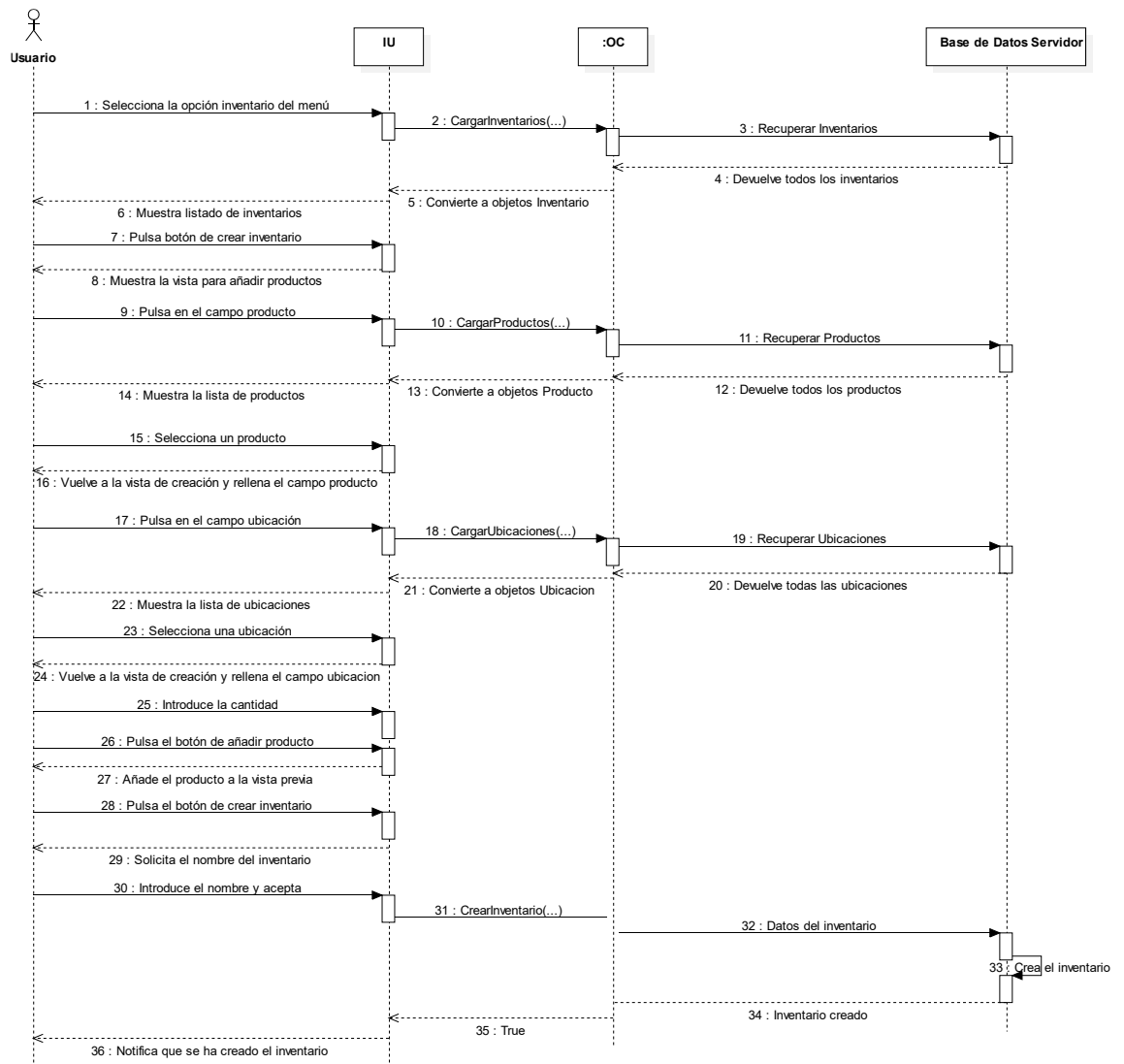


Figura 5.5.- Diagrama de secuencia del flujo de creación de inventario

Como se observa en el diagrama de secuencia, tras seleccionar el botón de crear inventario, el usuario debe introducir el producto que desee añadir al inventario y la ubicación seleccionándolos en una lista. La cantidad se introduce manualmente.

Tras introducir los datos, se añade el producto a la vista previa de la parte inferior, por lo que este proceso se puede repetir tantas veces como desee el usuario. Cuando todos los productos hayan sido añadidos, se introduce el nombre del inventario y este es creado en la base de datos.



6. Manual de usuario

6.1 Introducción

Este manual está orientado a usuarios que tengan su primer contacto con la aplicación. Se van a explicar detalladamente y con apoyo de capturas reales todas las funciones que ofrece y como hacer uso de ellas. Además, también se explican los flujos de trabajo y como llevarlos a cabo de manera correcta.

6.2 Inicio de sesión

Al iniciar la aplicación, lo primero que hay que hacer es seleccionar el servidor deseado, introducir el nombre de usuario y la contraseña correspondientes y seleccionar la base de datos del servidor.

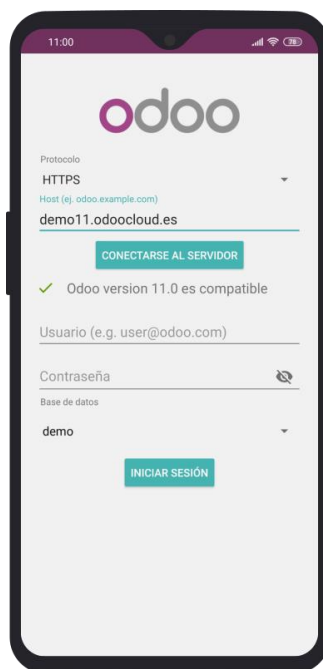


Figura 6.1.- Vista de inicio de sesión

Como se puede observar, en la pantalla aparecen diversos campos de texto. Aunque inicialmente solo aparece el desplegable de selección de protocolo y el campo para introducir la dirección del servidor. Aunque no es recomendable usar protocolo HTTP, se ha contemplado esta opción ya que algunos usuarios tienen servidores de prueba que funcionan con él. Al introducir los datos del servidor y pulsar el botón “**Conectarse al servidor**” se comprueba que este exista y sea compatible con la versión 11.0 de Odoo.

Si las comprobaciones son correctas, aparecen los campos de usuario y contraseña, además del desplegable para seleccionar la base de datos. Tras rellenarlos y pulsar el botón **“Iniciar sesión”**, si los datos son correctos, se establece la conexión con la base de datos elegida.

6.3 Tablero del inventario

Después de identificarse, por defecto se muestra la vista de tablero de inventario, que es una vista general que agrupa todos los movimientos por categorías.

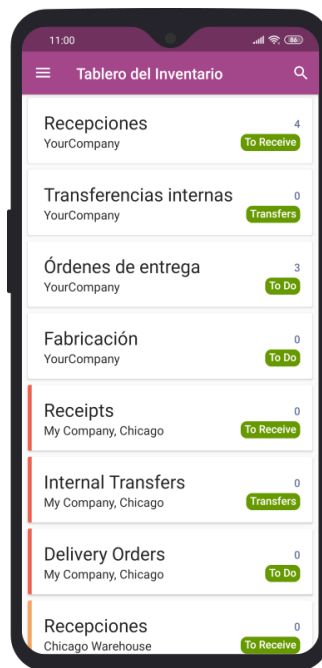


Figura 6.2.- Vista de tablero de inventario

Como se puede ver, esta pantalla se compone de una lista con los tipos de movimiento, mostrando para cada uno de ellos el nombre, la cantidad de albaranes que contiene entre otros datos. También permite añadir un color a cada elemento para poder distinguirlos fácilmente, aunque esto último se debe hacer desde la versión web.

En la barra superior se dispone de un botón de búsqueda, mediante el cual se puede introducir el nombre de la categoría que se desee, evitando así deslizar en la lista.

Al seleccionar cualquier elemento de la lista, se mostrará otro listado con los albaranes que contiene dicha categoría.

6.4 Menú lateral

En las pantallas principales, es decir, a las que no se accede desde otra pantalla, se dispone de un botón en la barra superior para abrir el menú desplegable del lateral izquierdo. Desde aquí, se puede acceder a las distintas vistas de la aplicación.

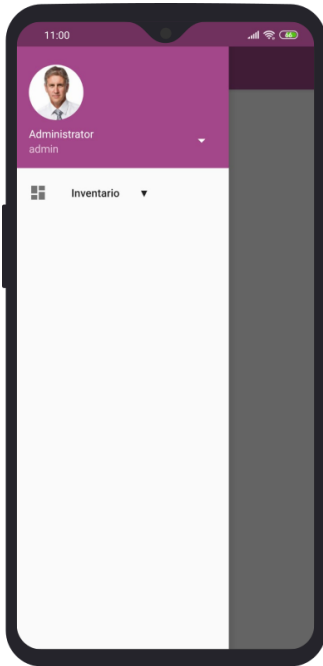


Figura 6.3.- Menú lateral (I)

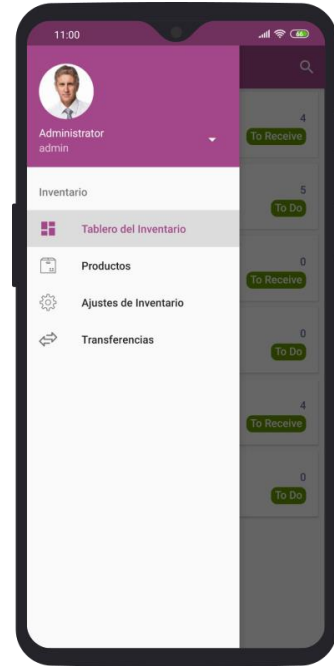


Figura 6.4.- Menú lateral (II)

Inicialmente, el menú lateral aparece con una sola opción, pero al seleccionarla se muestran todas las opciones de la aplicación de inventario, como se observa en la figura 6.4. Esto es así ya que dentro de *Nucleus* se pueden implementar más módulos de Odoo y de esta manera cada uno de ellos está dentro de un desplegable independiente, lo cual da como resultado un menú mucho más organizado.

Al desplegar el menú, se muestran cuatro opciones, ahora se va a explicar brevemente lo que ofrece cada una de ellas:

- **Tablero del Inventario:** Muestra una lista de movimientos organizada por categorías.
- **Productos:** Catálogo de productos.
- **Ajustes de Inventario:** Desde esta opción se pueden realizar todas las operaciones relacionadas con los inventarios.
- **Transferencias:** Esta opción permite ver un listado de albaranes y editarlos.

En la parte superior del menú también se ofrece información sobre el usuario conectado actualmente. Al lado derecho se encuentra un botón que lleva al menú secundario, como se puede ver en la siguiente imagen.

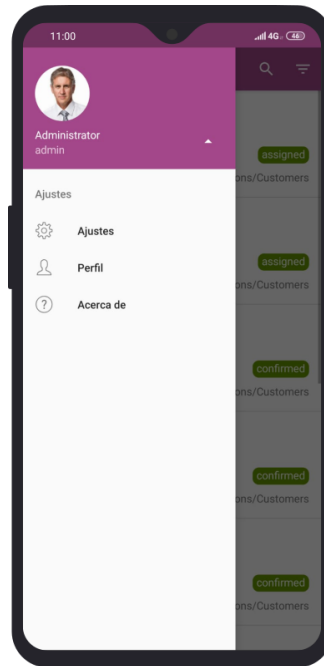


Figura 6.5.- Menú lateral (III)

En este menú se muestran tres opciones, desde “**Ajustes**” se puede cambiar el idioma de la aplicación. En “**Perfil**” se muestra más información sobre el usuario actual. Y en “**Acerca de**” hay información sobre los desarrolladores y la versión de la aplicación.

6.5 Catálogo de productos

Al usar el botón “**Productos**” del menú lateral, se accede al listado de todos los productos que se encuentran actualmente en la base de datos.

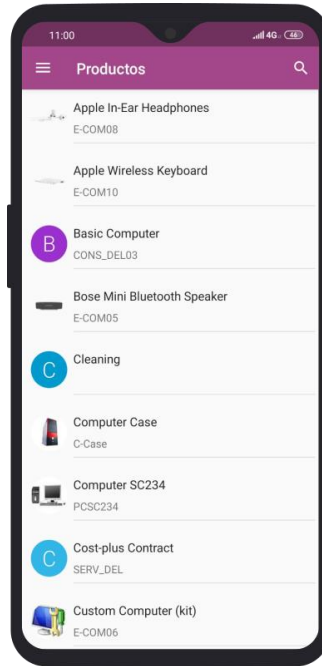


Figura 6.6.- Vista de catálogo

En la imagen se puede ver la pantalla de catálogo. Formada por una lista que contiene todos los productos y que muestra información básica sobre estos.

Al igual que en la vista de tablero, aquí también existe una opción de búsqueda mediante el botón de la barra superior, esta se puede realizar por nombre del producto o por referencia.

Si se selecciona cualquier producto, se accede a información más detallada sobre el mismo.

6.6 Detalles de producto

En esta pantalla se encuentra más información sobre el producto. Esta información está dividida en dos pestañas. Se van a analizar por separado para explicar las diferencias.

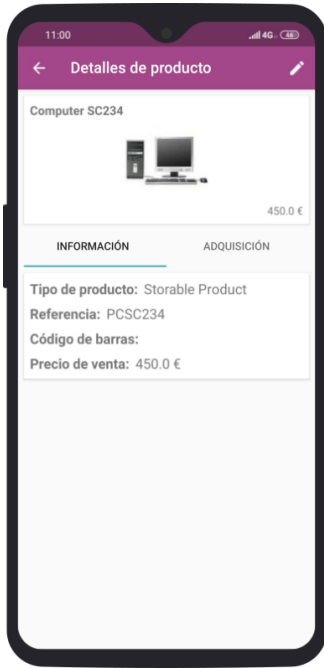


Figura 6.7.- Vista de detalle de producto (I)



Figura 6.8.- Vista de detalle de producto (II)

Por un lado, en la pestaña “**Información**”, es dónde se encuentra la información más general, como el tipo de producto, la referencia, el número del código de barras y el precio de venta.

Por otro lado, la pestaña “**Adquisición**” contiene información más orientada al vendedor o al encargado de almacén, como el precio de coste o el stock actual.

En la barra superior, se puede observar un botón el cual sirve para editar los datos del producto.

6.7 Editar producto

Esta vista es muy similar a la del detalle de los productos, solo que los campos de texto se convierten en editables para poder modificarlos.



Figura 6.9.- Vista de edición de producto

Todos los campos son de texto libre, es decir, se puede escribir a mano en ellos. Excepto el de tipo de producto, que al pulsarlo aparecerá un diálogo donde seleccionar entre las tres opciones que proporciona Odoo, como se observa en la figura 6.10. Se pueden editar todos los datos contenidos en ambas pestañas.



Figura 6.10.- Selección de tipo de producto

En la barra superior hay dos botones, uno para confirmar los cambios realizados y otro para descartarlos.

6.8 Ajustes de inventario

Al pulsar “**Ajustes de inventario**” en el menú lateral, aparece la lista de todos los ajustes de inventario.

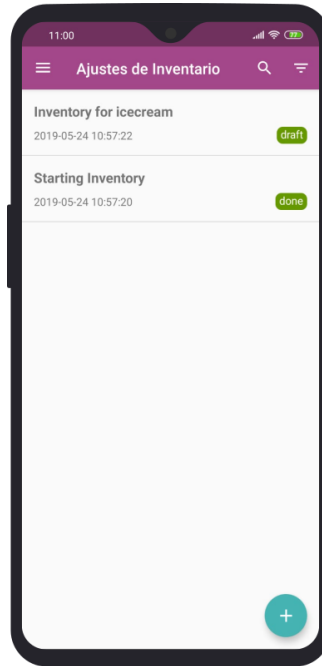


Figura 6.11.- Lista de ajustes de inventario

En cada elemento se puede ver el nombre del inventario, la fecha en que se realizó y el estado actual del mismo.

Al seleccionar cualquier elemento de la lista aparecerá otra lista con los productos que se encuentran en dicho inventario.

Usando el botón flotante que se encuentra en la parte inferior derecha, se puede crear un nuevo ajuste de inventario.

En la barra superior se ha incluido un botón para buscar por nombre y unos filtros por estado del inventario que permiten acotar la lista.

6.9 Detalle de inventario

Al seleccionar un inventario de la lista, se muestran los productos que contiene en formato de lista

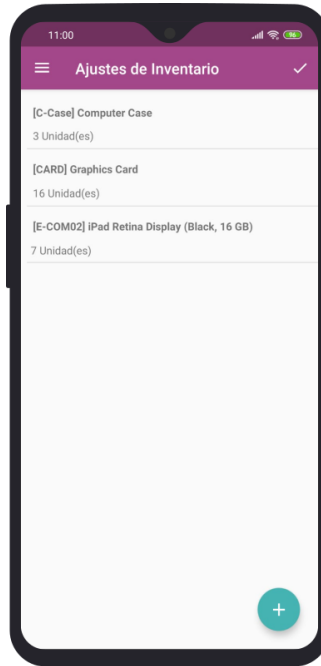


Figura 6.12.- Vista de detalle de inventario

Como se puede ver en la imagen, cada elemento de la lista se corresponde con uno de los productos que contiene el inventario. Mostrando el nombre y la cantidad.

Con el botón de la parte inferior derecha es posible añadir nuevos productos al inventario. Este botón no aparecerá si el inventario se encuentra en estado confirmado, ya que no es posible añadir nuevos productos. En la barra superior se encuentra otro botón que permite confirmar el inventario actual.

6.10 Crear inventario y añadir productos

El formulario para crear nuevos inventarios y para añadir productos a inventarios ya existentes es muy similar.

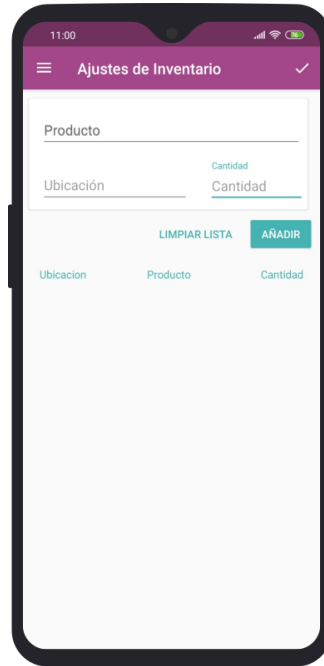


Figura 6.13.- Formulario para añadir productos a un inventario

Para crear un nuevo inventario, primero hay que introducir los productos mediante un formulario en el que hay que seleccionar el producto haciendo uso de un listado, la ubicación y la cantidad. Al pulsar el botón añadir, se mostrará en la lista de la parte inferior. Cuando se pulsa el botón “**crear inventario**”, hay que introducir el nombre de este y ya estará creado en la base de datos.

El proceso para añadir productos a un inventario existente es el mismo, pero sin necesidad de introducir el nombre de inventario al confirmar.

6.11 Transferencias

Al pulsar el botón “**Transferencias**” del menú lateral, se muestra un listado con todos los albaranes.

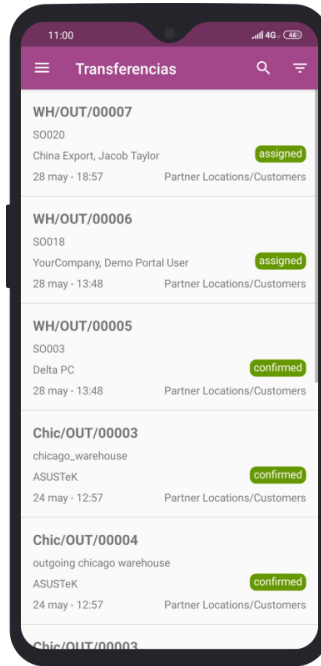


Figura 6.14.- Lista de transferencias

Como se puede observar en la figura 6.14, cada elemento de la lista representa un albarán y muestra datos como el nombre, el estado en que se encuentra, la fecha, entre otros. Al seleccionar cualquiera de ellos se accede a la lista de productos que contiene.

En la barra superior existen las mismas opciones que en la vista de ajustes de inventario, búsqueda y filtrado por estado.

6.12 Detalle y edición de albarán

Al seleccionar cualquier albarán, se muestra un listado con los productos que contiene.

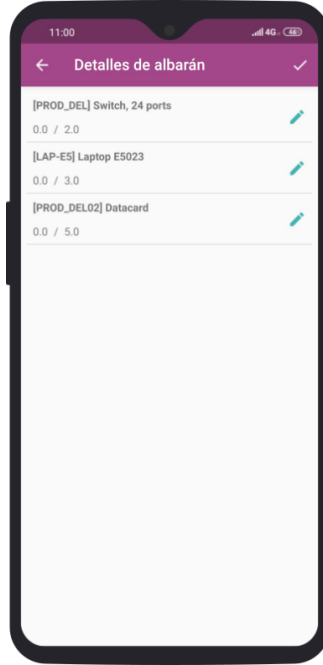


Figura 6.15.- Vista de detalle de albarán

En la lista se muestra el nombre de cada producto junto con la cantidad real y la cantidad teórica.

En la barra superior se encuentra la opción de “**validar albarán**”, que está disponible solo para usuarios autorizados y que permite confirmar dicho albarán.

Si el albarán no está validado, aparece a la derecha de cada producto un botón para editarlo.

Al pulsar el botón editar, se muestra un formulario con los datos de dicho producto, donde es posible modificar la cantidad real, como se observa en la siguiente imagen.

The image shows a smartphone screen with a purple header bar containing the text "Detalles de producto" and navigation icons. The main content area is white and contains the following information:

Producto	
[LAP-ES] Laptop E5023	
Demanda inicial	
3.0	
Hecho	Reservado
0.0	3.0
Ubicación de origen	
WH/Stock/Shelf 2	Reservado
	3.0
Ubicación de destino	
Partner Locations/Customers	Hecho
	0.0

Figura 6.16.- Formulario de edición de albarán

Este formulario es algo distinto al de edición de productos, ya que se divide en dos secciones. Por un lado, en la parte superior aparecen los datos del producto, como el nombre y la cantidad pedida por el cliente, entre otros. Por otro lado, en la parte inferior se muestra información sobre el origen y destino de la transferencia.

7. Conclusiones

El objetivo principal de este proyecto era diseñar e implementar una aplicación que cubriera por completo todas las tareas relacionadas con la gestión de almacén y facilitara el trabajo de los usuarios. Esto ha derivado, como se indica en el punto 1.1, en objetivos secundarios. Después de finalizar el desarrollo, se puede afirmar que se han cubierto todos los objetivos.

Por otro lado, en cuanto a la implementación, al inicio del proyecto fue mas costoso ya que nunca se había trabajado con muchas de las tecnologías usadas, y eso supuso un proceso de aprendizaje y adaptación ya que, por ejemplo, fue necesario aprender cómo funcionaba el ERP para conseguir conectar correctamente la aplicación. Tras esto, todo el desarrollo fue más fluido.

Para finalizar, se va a hablar sobre el aprendizaje adquirido con la realización de este proyecto. Profesionalmente, se han adquirido habilidades de trabajo en equipo y de organización para el desarrollo de proyectos de más envergadura que los realizados en la universidad y por supuesto, se han ampliado considerablemente los conocimientos sobre programación entre otras disciplinas informáticas. Por otro lado, en el ámbito personal, se ha aprendido a lidiar con la presión que supone enfrentarse a un entorno real, donde es necesario ajustarse a los tiempos marcados por el cliente.



8. Referencias

- [1] G. Moss, *Working with Odo 11*, Packt, 2018.
- [2] «RxJava: Reactive Extensions for the JVM,» [En línea]. Available: <https://github.com/ReactiveX/RxJava>.
- [3] «Android Developers,» [En línea]. Available: <https://developer.android.com/topic/libraries/data-binding>.
- [4] «Patrones de Diseño (VI): Patrones de Creación - Singleton».
- [5] Agencia MDW, «Agencia MDW,» [En línea]. Available: <https://www.agenciamdw.es/material-design-google-guia-en-espanol/>.
- [6] «Android Developers,» [En línea]. Available: <https://developer.android.com/studio/build>.
- [7] F. L. Jurado, «Qué es Kotlin,» 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-kotlin/>.
- [8] J. Thornsby, «Kotlin vs Java: key differences between Android's officially-supported languages,» 2018. [En línea]. Available: <https://www.androidauthority.com/kotlin-vs-java-783187/>.
- [9] «Android OS,» [En línea]. Available: <https://androidos.readthedocs.io/en/latest/data/caracteristicas/>.
- [10] «Android Developers,» 2018. [En línea]. Available: <https://developer.android.com/guide/platform>.
- [11] J. T. Gironés, «Arquitectura de Android,» de *El gran libro de Android*, Marcombo.
- [12] J. G. Gil, «8 Características más importantes de PostgreSQL,» [En línea]. Available: <https://openwebinars.net/blog/caracteristicas-importantes-de-postgresql>.