RESUMEN – TRABAJO DE GRADO

AUTORES	Cristian Alejandro Navarro Quintero
FACULTAD	**Ingenierías**
PLAN DE ESTUDIOS	**Ingeniería de Sistemas**
DIRECTOR	Brayan Fernando Sánchez Julio
TÍTULO DE LA TESIS	Desarrollo de aplicación para el sistema de inventario del área de soporte en la empresa World Pos Solutions S.A.S (WPOSS) en lenguaje Java
TITULO EN INGLES	Development of an application for the inventory system of the support area of World Pos Solutions S.A.S (WPOSS) in Java language.

RESUMEN
(70 palabras)

Se desarrolló una aplicación en JavaFX y FastAPI para la gestión de inventarios en WPOSS. La herramienta permite un seguimiento en tiempo real de los activos, mejorando la precisión y eficiencia del control. Se incluyeron funcionalidades CRUD, asignación de activos a personas, y generación de reportes en PDF. Esta solución optimiza la operación en el área de soporte, facilitando la gestión y control de los recursos disponibles

RESUMEN EN INGLES

A JavaFX and FastAPI application was developed for inventory management in WPOSS. The tool allows real-time tracking of assets, improving control accuracy and efficiency. CRUD functionalities were included, as well as the assignment of assets to people and the generation of PDF reports. This solution optimizes the operation in the support area, facilitating the management and control of available resources.

PALABRAS CLAVES	Inventarios, JavaFX, FastAPI, Asignación de Activos, Generación de Reportes
PALABRAS CLAVES EN INGLES	Inventory, JavaFX, FastAPI, Asset Allocation, Report Generation

CARACTERÍSTICAS

P□GINAS: 57	PLANOS:	ILUSTRACIONES:29	CD-ROM:

[redacted]

Desarrollo de Aplicación para el Sistema de Inventario del Área de Soporte en la Empresa World Pos Solutions S.A.S(WPOSS) en lenguaje Java

Cristian Alejandro Navarro Quintero

Facultad de Ingenierías, Universidad Francisco de Paula Santander Ocaña

Ingeniería de Sistemas

Ing. Brayan Fernando Sánchez Julio

25 julio del 2024

Índice

1 Desarrollo de Aplicación para el Sistema de Inventario del Área de Soporte en la Empresa World Pos Solutions S.A.S(WPOSS) en lenguaje Java 9

1.1 Descripción de la empresa 9

1.1.1 Misión..... 9

1.1.2 Visión. 10

1.1.3 Objetivos de la empresa. 10

1.1.4 Descripción de la dependencia y/o proyecto que fue asignado..... 11

1.1.5 Descripción de la estructura organizacional..... 11

2 Diagnóstico inicial de la dependencia asignada. 13

2.1 Planteamiento del problema..... 15

3 Objetivos de la pasantía 16

3.1 General..... 16

3.2 Específicos 16

3.3 Descripción de las actividades a desarrollar 17

4 Enfoques referenciales 21

4.1 Enfoque conceptual..... 21

4.1.1 Gestión de Inventarios..... 21

4.1.2 Teoría del Control de Inventarios..... 21

4.1.3 Tecnologías de la Información en la Gestión de Inventarios 22

4.1.4 Sistemas de Gestión de Inventarios (SGI)..... 22

4.1.5 Migración de Datos 22

4.1.6 ETL (Extracción, Transformación y Carga) 22

4.1.7 Desarrollo de API 23
4.1.8 Principios de Diseño de API REST 23
4.1.9 Desarrollo de Aplicaciones de Escritorio con JavaFX 23
4.1.10 Model-View-Controller (MVC) 23
4.1.11 Implementación de Reportes PDF con iTextPDF: 24
4.2 Enfoque Legal 24
4.2.1 Normativa sobre Gestión de Inventarios 24
4.2.2 Cumplimiento de Normas de Seguridad y Privacidad de Datos 24
4.2.3 Regulaciones Tecnológicas 25
4.2.4 Aspectos de Propiedad Intelectual 25
4.3 Cronograma de actividades 26
5 Informe de cumplimiento de trabajo 27
Presentación de resultados. 27
5.1 Creación y consumo de una API con FastAPI 27
5.1.1 Migración de datos y desarrollo de API 27
5.1.2 Creación de api en FastApi en lenguaje Python 31
5.1.3 Consumo de API en JAVAFX utilizando Jackson 35
5.2 Diseñar e implementar una interfaz gráfica en JavaFX 39
5.3 Implementar funcionalidades de CRUD 42
5.4 Relación entre personas y activos, y el seguimiento de esta relación 46
5.5 Generar un PDF de asignaciones 49
5.6 Crear un historial de asignaciones de activos 51
6 Diagnóstico Final 52

6.1.1 Desarrollo e Implementación de la Aplicación de Gestión de Inventario: 52

6.1.2 Creación y Consumo de API con FastAPI 52

6.1.3 Migración de Datos a PostgreSQL 52

6.1.4 Implementación del Patrón de Diseño MVC 53

6.1.5 Implementación de reportes PDF 53

7 Conclusiones 54

8 Recomendaciones 56

9 Referencias 58

Lista de Figuras

Figura 1 Estructura Organizacional de la empresa 11
Figura 2 Organigrama 12
Figura 3 Excel donde se manejaban el inventario en WPOSS 27
Figura 4 Excel convertido a CSV 28
Figura 5 Exportando el archivo CSV a PostgreSQL 28
Figura 6 Tabla con datos importados 29
Figura 7 Diagrama MERD sobre las tablas creadas 30
Figura 8 Endpoints creados 31
Figura 9 Creación de modelo el cual contiene la estructura de la base de datos 32
Figura 10 Schemas el cual define el tipo de datos 33
Figura 11 Documentación interactiva con el API 34
Figura 12 Clase Api Service consumo todos los endpoints 35
Figura 13 Consumo por número de activo 36
Figura 14 Implementación de JWT 36
Figura 15 Generación del token a partir del login 37
Figura 16 Clase donde se maneja el token de JWT en java 38
Figura 17 Vista Login 39
Figura 18 Vista Personas 40
Figura 19 Vista Activos 41
Figura 20 Vista Editar personas 42
Figura 21 Vista Agregar Personas 43
Figura 22 Vista Actualizar Activos 44
Figura 23 Vista Agregar Activos 45
Figura 24 Vista Asignación 46
Figura 25 Agregar Asignación 47
Figura 26 Eliminar Asignación 48
Figura 27 Vista Generar PDF Asignaciones 49
Figura 28 Reporte generado PDF Activos Asignados a personas 50
Figura 29 Vista Historial 51

Lista de Tablas

Tabla 1 Matriz DOFA 13
Tabla 2 Matriz de estrategias FO – DO – FA – DA 14
Tabla 3 Descripción de actividades 19
Tabla 4 Semanas por actividad 26

Resumen

Objetivos:

El objetivo principal de la pasantía fue desarrollar una aplicación efectiva y eficiente para la gestión de inventario en WPOSS, utilizando JavaFX para la interfaz gráfica y FastAPI para la creación y consumo de una API. Los objetivos específicos incluían diseñar una interfaz gráfica intuitiva, implementar funcionalidades de edición, y desarrollar una relación entre personas y activos para el seguimiento de asignaciones.

Metodología:

Para lograr estos objetivos, se llevaron a cabo diversas actividades, incluyendo el análisis de requisitos, la investigación sobre sistemas de inventario, el diseño de la arquitectura de la aplicación y la base de datos, la migración de datos de Excel a PostgreSQL, y el desarrollo de la API con FastAPI. El desarrollo de la aplicación se realizó en JavaFX, seguido de pruebas unitarias y de integración para asegurar la calidad del sistema.

La implementación de la aplicación de gestión de inventario resultó en una mejora significativa en la eficiencia operativa del área de soporte de WPOSS. La aplicación permite un seguimiento preciso y en tiempo real de los activos, reduce el tiempo y los recursos necesarios para la gestión del inventario, y aumenta la satisfacción del cliente. Esta experiencia de pasantía me permitió adquirir habilidades prácticas en desarrollo de software, así como una comprensión profunda de los desafíos y soluciones en la gestión de inventarios.

Introducción

Durante mi pasantía en World Pos Solutions S.A.S. (WPOSS), me enfoqué en el desarrollo de una aplicación para la gestión de inventario en el área de soporte. Este proyecto surge debido a los desafíos significativos en la gestión de inventario que enfrenta la empresa, afectando su eficiencia operativa. La falta de una solución de software adecuada dificulta el seguimiento preciso de activos y recursos, lo que ocasiona pérdidas de tiempo y recursos. La aplicación desarrollada busca mejorar estos aspectos, proporcionando una herramienta que permite realizar un seguimiento en tiempo real, optimizando los procesos internos y fortaleciendo la posición competitiva de WPOSS en el mercado

1 Desarrollo de Aplicación para el Sistema de Inventario del Área de Soporte en la Empresa World Pos Solutions S.A.S(WPOSS) en lenguaje Java

1.1 Descripción de la empresa

WPOSS World Pos Solutions Somos una compañía con más de 25 años de experiencia en el desarrollo de soluciones de software y venta de dispositivos electrónicos para medios de pago como impresoras, lectores de códigos de barras y terminales de punto de venta especializadas.

Somos líderes en el desarrollo de soluciones transaccionales, enfocados en proteger la inversión de nuestros clientes, garantizándoles equipos seguros y tecnología avanzada que se ajusten a sus necesidades, brindando seguimiento continuo desde el área comercial y tecnológica, apoyados en un excelente servicio, procesos de certificación, asesoría y gestión de conocimiento. (WPOSS, 2021)

1.1.1 Misión.

La misión de la empresa es "ser aliado estratégico de nuestros clientes en la innovación y evolución de modelos de negocios de soluciones integrales, que incluyen paquetes completos de productos tecnológicos y servicios asociados, cumpliendo con los estándares de seguridad, altos niveles de servicio y calidad" (WPOSS, 2021)

1.1.2 Visión.

Convertirnos en la primera opción como socio estratégico y tecnológico de nuestros clientes en el mercado de América Latina. (WPOSS, 2021)

1.1.3 Objetivos de la empresa.

- Obtener un ingreso de 15.000 millones de pesos en ventas.
- Mantener nuestra posición en Latinoamérica a través de la venta de 50.000 dispositivos de pago.
- Tener un centro de desarrollo con altos estándares que se concentre en ofrecer soluciones tecnológicas y de pago.
- Generar ingresos fusionando dos nuevos productos: comercio electrónico y fábrica de software.
- Implementar un sistema de gestión de calidad.
- Establecer una nueva sede aliada en Paraguay para ampliar la cobertura de nuestros servicios post-venta.
- Desarrollar Adaptaciones y personalizaciones en sistemas de información y aplicaciones
- Ampliar nuestras sedes de investigación y desarrollo en Colombia, mediante la gestión y conservación del conocimiento.
- Proveer las mejores soluciones especializadas en los sectores bancarios, como transaccionales y de medios de pago, con el objetivo de maximizar la rentabilidad de nuestros clientes.

1.1.4 Descripción de la dependencia y/o proyecto que fue asignado.

El área de desarrollo de World Pos Solutions S.A. se encarga de realizar los proyectos de software de la empresa, que serán uno de los productos de valor que ofrecerá a los clientes. Esta área está conformada por subáreas como QA (Quality Assurance), UX (Experiencia del usuario), POS (Dispositivos Móviles), PMO (Oficina de Gestión del Proyecto) y, por último, IP. El área de POS (dispositivos móviles) fue la que se fue asignada

1.1.5 Descripción de la estructura organizacional.

La estructura organizacional de WPOSS está dada de la siguiente manera:

Fuente: World Pos Solutions SAS, 2020

Figura 1

Estructura Organizacional de la empresa

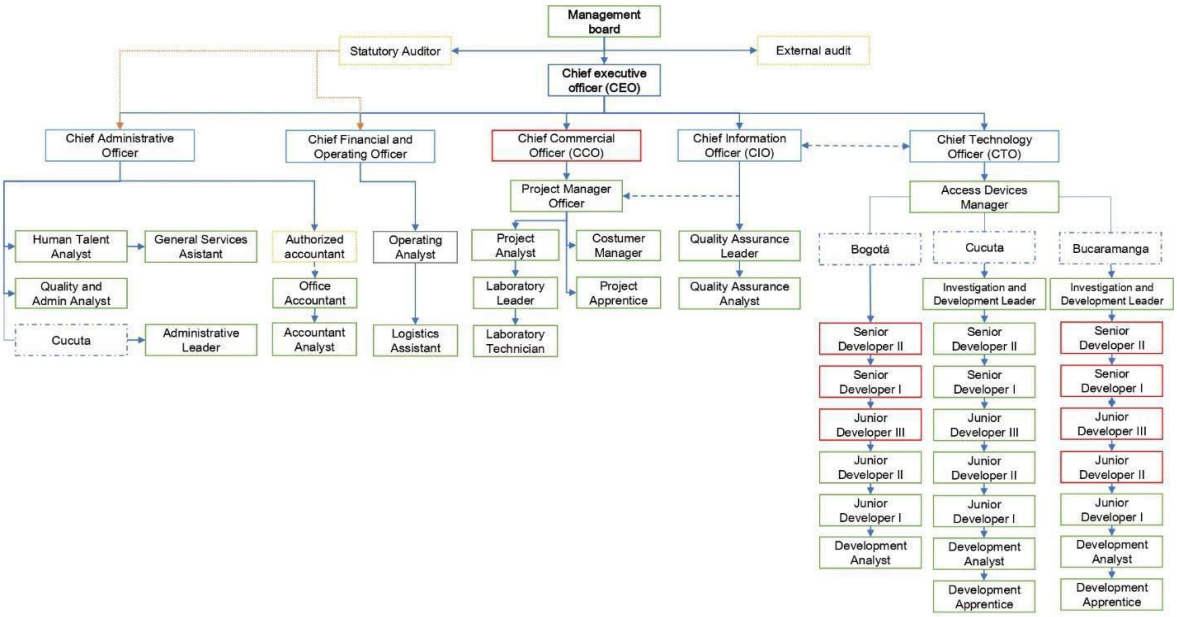
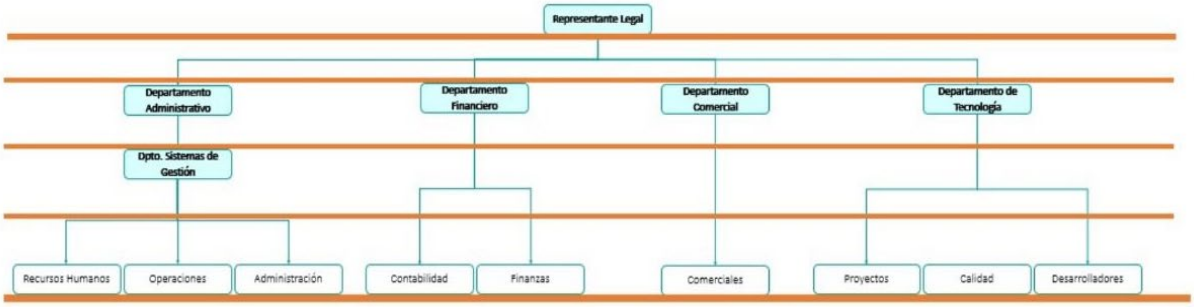


Figura 2

Organigrama



Nota: Organigrama WPOSS. (Organigrama WPOSS.)

2 Diagnóstico inicial de la dependencia asignada.

Tabla 1

Matriz DOFA

Fortalezas.	**Debilidades.**
• El área de POS brinda el personal idóneo para desarrollar aplicaciones empleando el entorno de desarrollo adecuado • Se hace uso de la metodología SCRUM la cual se utiliza para la colaboración entre unidades de desarrollo • Se dispone de instalaciones adecuadas y amplias, así como los sistemas de computación de última generación.	Se dispone de poca o desactualizada documentación de software antiguamente desarrollado que aún necesita mantenimiento. El tiempo de desarrollo se ralentiza porque los empleados listos para el desarrollo deben rotar entre proyectos nuevos y antiguos
Oportunidades.	**Amenazas.**
• El deseo de los clientes de adquirir nuevos bienes y servicios de la empresa • La capacitación está disponible para los empleados contratados en una variedad de áreas de desarrollo.	Retraso en el proyecto. Insatisfacción del cliente. Costos adicionales. Cancelación del proyecto Cambios en el lenguaje a desarrollar

Nota: Elaboración propia

Tabla 2

Matriz de estrategias FO – DO – FA – DA

Factores Externos / **Factores Internos**	**Fortalezas**	**Debilidades**
Oportunidades	Desarrollar productos tecnológicos que mejoren la competitividad de la empresa Implementación exitosa de metodologías ágiles como SCRUM	Mejorar la gestión de recursos humanos para optimizar la asignación de empleados Actualizar y mejorar la documentación de software desarrollado en años anteriores para facilitar el mantenimiento.
Amenazas	Implementar una arquitectura de software basada en microservicios que permita la adaptación a cambios y la incorporación de nuevas funcionalidades, así como la implementación de nuevas tecnologías sin alterar las existentes Investigación y desarrollo de soluciones innovadoras en el área de POS, aprovechando la infraestructura tecnológica y la experiencia del equipo	Riesgo de que los proyectos nuevos no se desarrollen con la calidad y velocidad esperadas debido a la limitación de recursos causada por la dedicación a proyectos antiguos.

Nota: Elaboración propia.

2.1 Planteamiento del problema

En la empresa World Pos Solutions S.A.S (WPOSS), el área de soporte se enfrenta a desafíos significativos en la gestión de inventario, lo que afecta la eficiencia operativa y la satisfacción del cliente. Actualmente, la empresa carece de una solución de software que permita a los equipos de soporte realizar un seguimiento efectivo de los activos y recursos en tiempo real.

La falta de una aplicación dedicada a la gestión de inventario dificulta la identificación precisa de productos, el seguimiento de las existencias y la actualización oportuna de la información. Esto resulta en problemas como**:**

- Pérdida de tiempo y recursos debido a la gestión manual de inventario o sistemas obsoletos**.**
- Es difícil mantener un registro de productos disponibles actualizado, lo que puede causar errores en el inventario y pérdidas de dispositivos.

La creación de una aplicación personalizada para el área de soporte que facilite la gestión eficiente del inventario es necesaria en este momento. Esta aplicación no solo aumentará la eficiencia y la precisión del seguimiento de activos, sino que también mejorará los procesos internos y aumentará la ventaja de WPOSS en el mercado.

Johnson, A. (2022). "Mejora de la eficiencia operativa mediante la implementación de soluciones móviles en la gestión de inventario: Un estudio de caso en empresas del sector de tecnología." Revista de Gestión Empresarial, 15(2), 45-60

3 Objetivos de la pasantía

El objetivo de mi pasantía en World Pos Solutions S.A.S. (WPOSS) es adquirir experiencia práctica en el desarrollo de aplicaciones de gestión de inventario utilizando el entorno de desarrollo IntelliJ IDEA y metodologías ágiles como SCRUM. Busco mejorar mis habilidades en programación en Java y en la creación de interfaces de usuario intuitivas y atractivas a través de esta experiencia. Además, me propongo colaborar con el equipo de desarrollo para identificar y resolver problemas específicos de gestión de inventario, contribuyendo así al éxito y la innovación del mercado de WPOSS.

3.1 General

Desarrollar una aplicación para la gestión de inventario en el área de soporte de la empresa World Pos Solutions S.A.S. (WPOSS), utilizando JavaFX para la interfaz gráfica y FastAPI para la creación y consumo de una API, permitiendo una gestión eficiente.

3.2 Específicos

1. Diseñar e implementar una interfaz gráfica en JavaFX que permita a los usuarios visualizar y gestionar los datos de inventario en tablas.
2. Crear y consumir una API con FastAPI para realizar operaciones CRU sobre los datos de personas y activos, facilitando su actualización en tiempo real.
3. Implementar funcionalidades de CRUD para que los usuarios puedan modificar la información de personas y activos directamente desde la interfaz.
4. Desarrollar una relación entre personas y activos, permitiendo la asignación de activos a personas y el seguimiento de esta relación.
5. Crear un historial de asignaciones de activos, registrando cada movimiento y permitiendo la trazabilidad de los activos a lo largo del tiempo.

6. Generar un PDF el cual contenga el reporte de activos asignados a la persona.

3.3 Descripción de las actividades a desarrollar

Para lograr los objetivos planteados, se desarrollarán las siguientes actividades:

1. Análisis de requisitos: Identificación y documentación de los requisitos del sistema de inventario.
2. Diseño del sistema: Diseño de la arquitectura de la aplicación y la base de datos.
3. Migración de datos: Conversión de datos de Excel a CSV y posterior importación a PostgreSQL.
4. Creación de base de datos: Creación de las tablas según la información suministrada.
5. Investigación sobre Python: Estudio del lenguaje Python y su uso en la creación de la API.
6. Desarrollo de API: Implementación de la API con FastAPI para la gestión de inventario.
7. Desarrollo de la aplicación: Implementación de la aplicación de gestión de inventario en JavaFX.
8. Implementar funcionalidades de CRUD: Implementación de funcionalidades de CRUD para que los usuarios puedan modificar la información de personas y activos directamente desde la interfaz.
9. Asignación de activos a personas: Desarrollar una relación entre personas y activos, permitiendo la asignación de activos a personas y el seguimiento de esta relación.

10. Historial de asignaciones: Crear un historial de asignaciones de activos, registrando cada movimiento y permitiendo la trazabilidad de los activos a lo largo del tiempo.
11. Generar Reporte: Generar un PDF el cual contenga el reporte de activos asignados a la persona.
12. Generar Reporte: Implementar reportes en PDF para activos asignados a personas

Tabla 3

Descripción de actividades

Objetivo General	Objetivos Específicos	Actividades a desarrollar
Desarrollar una aplicación efectiva y eficiente para la gestión de inventario en WPOSS, que permita al equipo de soporte realizar un seguimiento preciso y en tiempo real de los activos y recursos disponibles	1. Diseñar e implementar una interfaz gráfica en JavaFX que permita a los usuarios visualizar y gestionar los datos de inventario en tablas. 2. Crear y consumir una API con FastAPI para realizar operaciones CRU sobre los datos de personas y activos, facilitando su actualización en tiempo real. 3. Implementar funcionalidades de edición para que los usuarios puedan modificar la información de personas y activos directamente desde la interfaz. 4. Desarrollar una relación entre personas y activos, permitiendo la asignación de activos a personas y el seguimiento de esta relación. 5. Crear un historial de asignaciones de activos, registrando cada movimiento y permitiendo la trazabilidad de los	1. Análisis de requisitos 2. Diseño del sistema 3. Migración de datos 4. Creación de base de datos 5. Investigación sobre Python 6. Desarrollo de api 7. Desarrollo de la aplicación 8. Implementar funcionalidades de CRUD 9. Asignación de activos a personas 10. Historial de asignaciones 11. Generar Reporte

	activos a lo largo del tiempo 6. Generar un PDF el cual contenga el reporte de activos asignados a la persona.

Nota: Elaboración propia

4 Enfoques referenciales

4.1 Enfoque conceptual

El enfoque conceptual de este proyecto se basa en la implementación de tecnologías modernas para la gestión eficiente de inventarios. A continuación, se describen los conceptos y teorías clave que fundamentan el desarrollo de la aplicación de gestión de inventarios para WPOSS.

4.1.1 Gestión de Inventarios

La gestión de inventarios es un proceso crucial en las operaciones de cualquier empresa que maneje bienes físicos. Este proceso implica la supervisión constante y el control de las cantidades de bienes que una empresa posee en sus almacenes, con el objetivo de mantener niveles óptimos de inventario que aseguren la disponibilidad de productos sin incurrir en excesos que aumenten los costos.

4.1.2 Teoría del Control de Inventarios

Esta teoría se centra en la optimización del inventario mediante modelos matemáticos y estadísticos que ayudan a determinar las cantidades adecuadas de productos que deben mantenerse en stock. Entre los modelos más utilizados se encuentran el Modelo de Cantidad Económica de Pedido (EOQ) y el Sistema de Revisión Periódica.

4.1.3 Tecnologías de la Información en la Gestión de Inventarios

La incorporación de tecnologías de la información (TI) en la gestión de inventarios ha revolucionado la manera en que las empresas manejan sus bienes. Las TI permiten la automatización de procesos, la mejora de la precisión en los registros y la posibilidad de acceder a información en tiempo real, lo que facilita la toma de decisiones informadas.

4.1.4 Sistemas de Gestión de Inventarios (SGI)

Los SGI son aplicaciones diseñadas para rastrear, gestionar y organizar los niveles de inventario, pedidos, ventas y entregas. Estos sistemas pueden ser integrados con otras herramientas empresariales como los sistemas ERP (Enterprise Resource Planning) para proporcionar una solución completa y cohesiva.

4.1.5 Migración de Datos

La migración de datos es el proceso de transferir datos de un sistema de almacenamiento a otro. En este proyecto, se realiza la migración de datos desde un archivo Excel a una base de datos PostgreSQL. Este proceso es fundamental para asegurar que la información de inventarios sea accesible y manejable mediante la nueva aplicación desarrollada.

4.1.6 ETL (Extracción, Transformación y Carga)

Este es un proceso de tres etapas que implica la extracción de datos de fuentes originales, la transformación de esos datos a un formato adecuado y la carga de los datos transformados en un sistema de destino. En este proyecto, se utiliza ETL para convertir los datos de Excel a formato CSV y luego importar estos datos a PostgreSQL

4.1.7 Desarrollo de API

Una API (Interfaz de Programación de Aplicaciones) permite la comunicación entre diferentes componentes de software. En este proyecto, se desarrolla una API utilizando FastAPI para manejar las operaciones CRUD (Crear, Leer, Actualizar y Borrar) sobre los datos de inventario.

4.1.8 Principios de Diseño de API REST

REST (Representational State Transfer) es un estilo arquitectónico para el diseño de servicios web. Las API RESTful son conocidas por su simplicidad y escalabilidad, y se basan en operaciones HTTP estándar. FastAPI es un marco de Python que facilita la creación de API RESTful de manera rápida y eficiente.

4.1.9 Desarrollo de Aplicaciones de Escritorio con JavaFX

JavaFX es una plataforma para la creación de aplicaciones de escritorio y web. En este proyecto, se utiliza JavaFX para desarrollar la interfaz gráfica de la aplicación de gestión de inventarios.

4.1.10 Model-View-Controller (MVC)

Este es un patrón de diseño que separa la lógica de la aplicación en tres componentes: el Modelo (datos y lógica de negocio), la Vista (interfaz de usuario) y el Controlador (gestión de la comunicación entre el Modelo y la Vista). La aplicación desarrollada sigue este patrón para mantener un código modular y fácil de mantener.

4.1.11 Implementación de Reportes PDF con iTextPDF:

Para la generación de reportes en formato PDF, se utilizó la biblioteca iTextPDF. Esta biblioteca proporciona una API robusta y flexible para crear y manipular documentos PDF. En el contexto de la aplicación de gestión de inventarios, se ha integrado iTextPDF para permitir a los usuarios exportar datos importantes a archivos PDF con un formato profesional. Los reportes generados incluyen información detallada.

4.2 Enfoque Legal

4.2.1 Normativa sobre Gestión de Inventarios

En Colombia, la gestión de inventarios está regulada por diversas normativas que buscan asegurar la transparencia y eficiencia en la administración de bienes y recursos. Entre estas normativas se encuentran la Ley 1314 de 2009, que establece principios y normas de contabilidad e información financiera y de aseguramiento de la información, y las regulaciones específicas del sector al que pertenece WPOSS.

4.2.2 Cumplimiento de Normas de Seguridad y Privacidad de Datos

El manejo de información de inventarios implica la recopilación, almacenamiento y procesamiento de datos, lo cual debe hacerse conforme a las leyes de protección de datos personales establecidas en la Ley 1581 de 2012 y sus decretos reglamentarios. Estas normativas aseguran que los datos sean manejados con la debida confidencialidad y seguridad, protegiendo la privacidad de la información.

4.2.3 Regulaciones Tecnológicas

El uso de tecnologías de la información en la gestión de inventarios debe cumplir con las regulaciones tecnológicas vigentes, que incluyen la Ley 527 de 1999 sobre comercio electrónico, firmas digitales y mensajes de datos, que establece el marco legal para la validación de transacciones electrónicas y la autenticidad de la información digital.

4.2.4 Aspectos de Propiedad Intelectual

El desarrollo de software, como la aplicación de gestión de inventarios, está protegido bajo la Ley 23 de 1982 sobre derechos de autor y sus modificaciones. Esta ley protege los derechos de autor sobre las creaciones intelectuales y garantiza el reconocimiento y la protección de los derechos morales y patrimoniales del autor del software.

4.3 Cronograma de actividades

El cronograma está estipulado para un total de 4 meses igual a 16 semanas con un horario laboral de lunes a sábados, excluyendo los días festivos.

Tabla 4

Semanas por actividad

Semana	Actividad
1	Análisis de requisitos
2	Investigación de manejo y usos de sistemas de inventario
3-4	Diseño del sistema
5-6	Migración de datos
7-8	Creación de base de datos
8	Investigación sobre Python
9-10	Desarrollo de API
11-12	Desarrollo de la aplicación
13-14	Implementar asignación de activos a personas
15	Historial de asignaciones
16	Generar Reporte
Total	**16 semanas**

Nota: Elaboración propia.

5 Informe de cumplimiento de trabajo

Presentación de resultados.

5.1 Creación y consumo de una API con FastAPI

5.1.1 Migración de datos y desarrollo de API

Para lograr una gestión de inventario eficiente y centralizada, se realizó la migración de datos desde un archivo Excel de activos a una base de datos PostgreSQL utilizando pgAdmin. El proceso incluyó los siguientes pasos:

Conversión a CSV: Los datos del archivo Excel fueron convertidos a un formato CSV para facilitar su importación a PostgreSQL

Figura 3

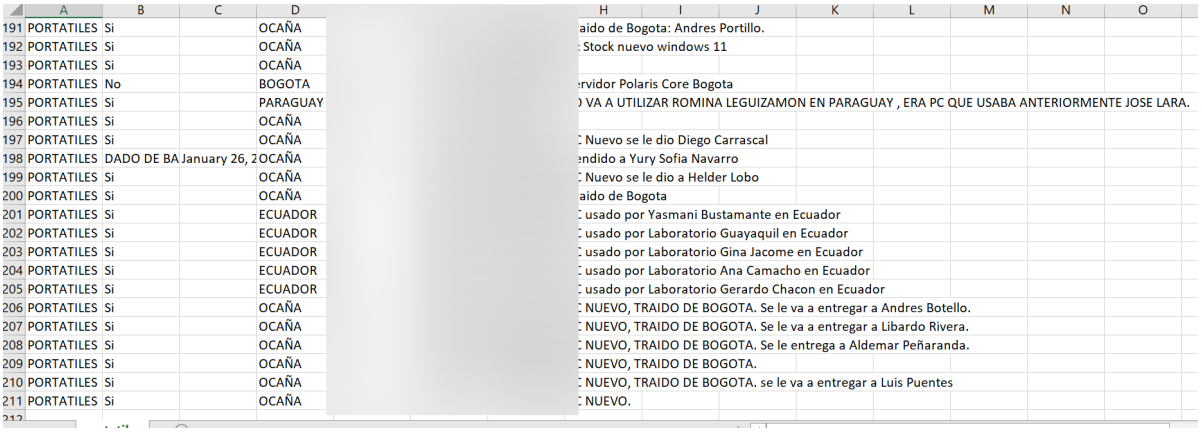
Excel donde se manejaban el inventario en WPOSS



	A	B	C	D	E	F	G	H
	NÚMERO DE ACTIVO	DISPONIBLE	FECHA DE BAJA	SEDE ASIGNADO	SEDE BASE	MARCA	SERIAL	MODELO
	WPOSS-152805.01-001	Si			OCAÑA	HP		
	WPOSS-152805.01-002	no			OCAÑA	HP		
	WPOSS-152805.01-003	Si			OCAÑA	HP		
	WPOSS-152805.01-004	Si			ECUADOR	HP		
	WPOSS-152805.01-005	DADO DE BAJA	August 30, 2023		B/MANGA	HP		
	WPOSS-152805.01-006	Si			OCAÑA	HP		
	WPOSS-152805.01-007	DADO DE BAJA	January 12, 2023		OCAÑA	HP		
	WPOSS-152805.01-008	Si			OCAÑA	MSI		
	WPOSS-152805.01-009	Si			OCAÑA	HP		
	WPOSS-152805.01-010	Si			OCAÑA	HP		
	WPOSS-152805.01-011	DADO DE BAJA	January 12, 2023		OCAÑA	HP		
	WPOSS-152805.01-012	Si			OCAÑA	HP		
	WPOSS-152805.01-013	Si			OCAÑA	HP		
	WPOSS-152805.01-014	Si			OCAÑA	HP		
	WPOSS-152805.01-015	Si			OCAÑA	HP		
	WPOSS-152805.01-016	DADO DE BAJA	February 6, 2023		OCAÑA	MSI		
	WPOSS-152805.01-017	Si			OCAÑA	HP		
	WPOSS-152805.01-018	Si			OCAÑA	HP		
	WPOSS-152805.01-019	Si			OCAÑA	MSI		
	WPOSS-152805.01-020	Si			OCAÑA	MSI		

Figura 4

Excel convertido a CSV



	A	B	C	D	H	I	J	K	L	M	N	O
191	PORTATILES	Si		OCAÑA	aido de Bogota: Andres Portillo.							
192	PORTATILES	Si		OCAÑA	: Stock nuevo windows 11							
193	PORTATILES	Si		OCAÑA								
194	PORTATILES	No		BOGOTA	rvidor Polaris Core Bogota							
195	PORTATILES	Si		PARAGUAY	) VA A UTILIZAR ROMINA LEGUIZAMON EN PARAGUAY , ERA PC QUE USABA ANTERIORMENTE JOSE LARA.							
196	PORTATILES	Si		OCAÑA								
197	PORTATILES	Si		OCAÑA	: Nuevo se le dio Diego Carrascal							
198	PORTATILES	DADO DE BA	January 26, 2	OCAÑA	ndido a Yury Sofia Navarro							
199	PORTATILES	Si		OCAÑA	: Nuevo se le dio a Helder Lobo							
200	PORTATILES	Si		OCAÑA	aido de Bogota							
201	PORTATILES	Si		ECUADOR	: usado por Yasmani Bustamante en Ecuador							
202	PORTATILES	Si		ECUADOR	: usado por Laboratorio Guayaquil en Ecuador							
203	PORTATILES	Si		ECUADOR	: usado por Laboratorio Gina Jacome en Ecuador							
204	PORTATILES	Si		ECUADOR	: usado por Laboratorio Ana Camacho en Ecuador							
205	PORTATILES	Si		ECUADOR	: usado por Laboratorio Gerardo Chacon en Ecuador							
206	PORTATILES	Si		OCAÑA	: NUEVO, TRAIDO DE BOGOTA. Se le va a entregar a Andres Botello.							
207	PORTATILES	Si		OCAÑA	: NUEVO, TRAIDO DE BOGOTA. Se le va a entregar a Libardo Rivera.							
208	PORTATILES	Si		OCAÑA	: NUEVO, TRAIDO DE BOGOTA. Se le entrega a Aldemar Peñaranda.							
209	PORTATILES	Si		OCAÑA	: NUEVO, TRAIDO DE BOGOTA.							
210	PORTATILES	Si		OCAÑA	: NUEVO, TRAIDO DE BOGOTA. se le va a entregar a Luis Puentes							
211	PORTATILES	Si		OCAÑA	: NUEVO.							

Se eliminaron las comillas y caracteres especiales, convirtiendo el archivo a UTF-8 para poder importarlo correctamente a la base de datos, y se ajustaron las tablas para que coincidan con las creadas en la base de datos.

Figura 5

Exportando el archivo CSV a PostgreSQL

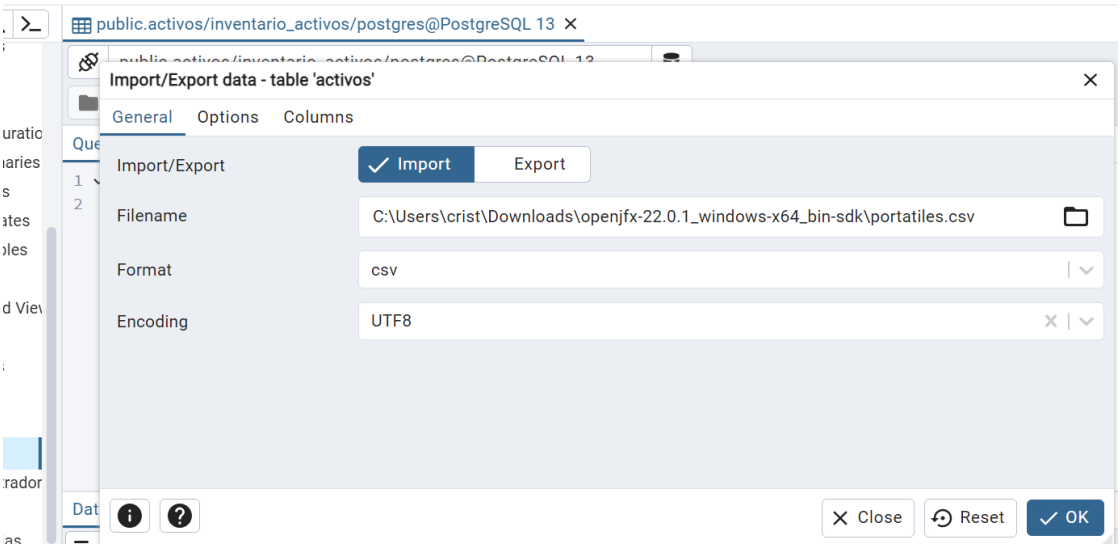
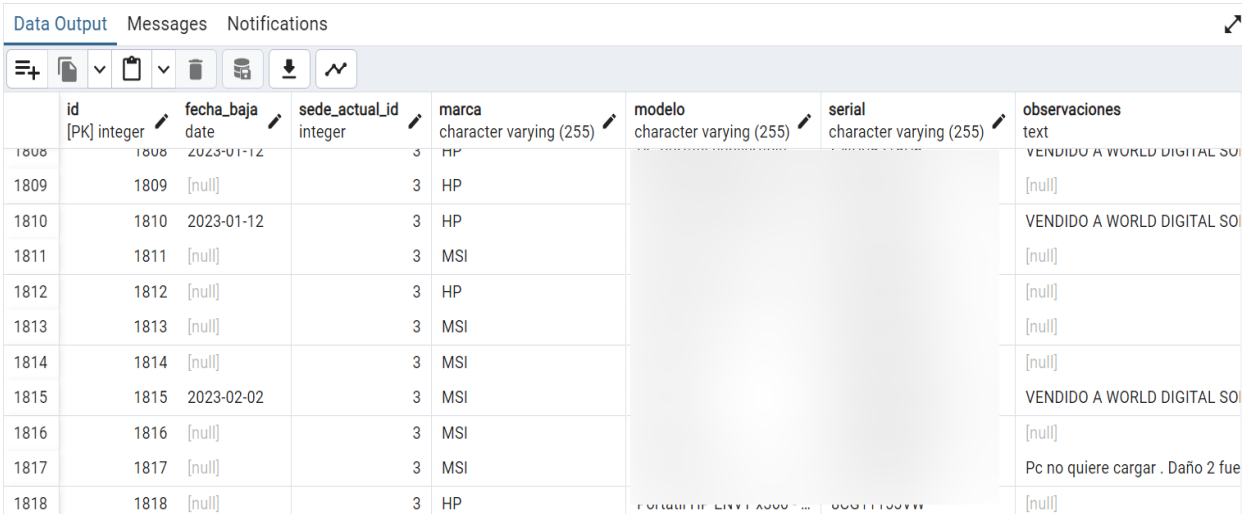


Figura 6

Tabla con datos importados



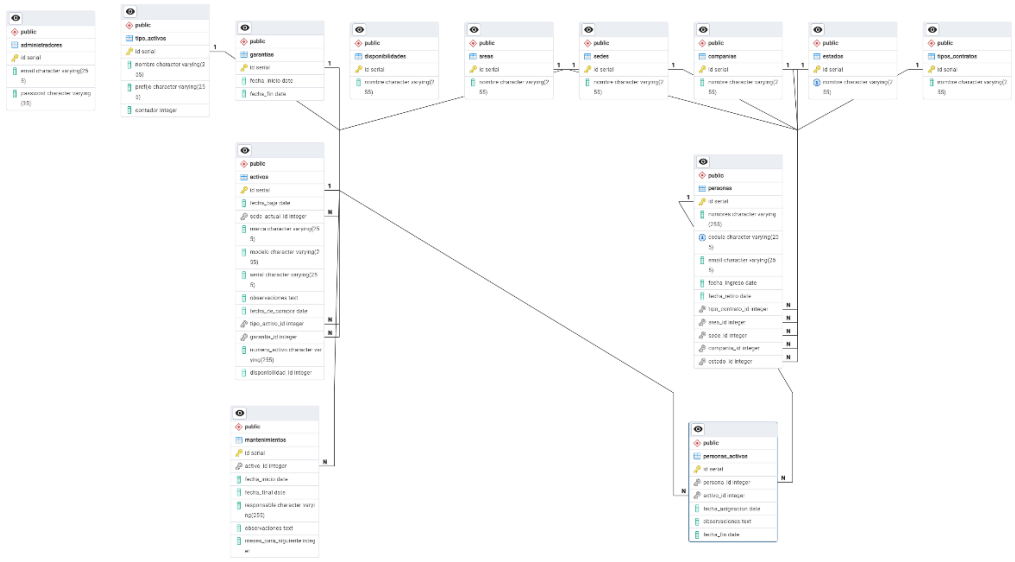
Data Output | Messages | Notifications

	id [PK] integer	fecha_baja date	sede_actual_id integer	marca character varying (255)	modelo character varying (255)	serial character varying (255)	observaciones text
1808	1808	2023-01-12	3	HP			VENDIDO A WORLD DIGITAL SO
1809	1809	[null]	3	HP			[null]
1810	1810	2023-01-12	3	HP			VENDIDO A WORLD DIGITAL SO
1811	1811	[null]	3	MSI			[null]
1812	1812	[null]	3	HP			[null]
1813	1813	[null]	3	MSI			[null]
1814	1814	[null]	3	MSI			[null]
1815	1815	2023-02-02	3	MSI			VENDIDO A WORLD DIGITAL SO
1816	1816	[null]	3	MSI			[null]
1817	1817	[null]	3	MSI			Pc no quiere cargar . Daño 2 fue
1818	1818	[null]	3	HP			[null]

Importación a PostgreSQL: Utilizando pgAdmin, los datos del archivo CSV fueron importados a una nueva base de datos creada específicamente para gestionar los activos de WPOSS.

Figura 7

Diagrama MERD sobre las tablas creadas



5.1.2 Creación de api en FastApi en lenguaje Python

Para permitir el acceso y la manipulación de los datos desde la aplicación, se desarrolló una API utilizando FastAPI. Esta API proporciona endpoints para realizar operaciones CRU (Crear, Leer, Actualizar) sobre los datos de inventario.

Figura 8

Endpoints creados

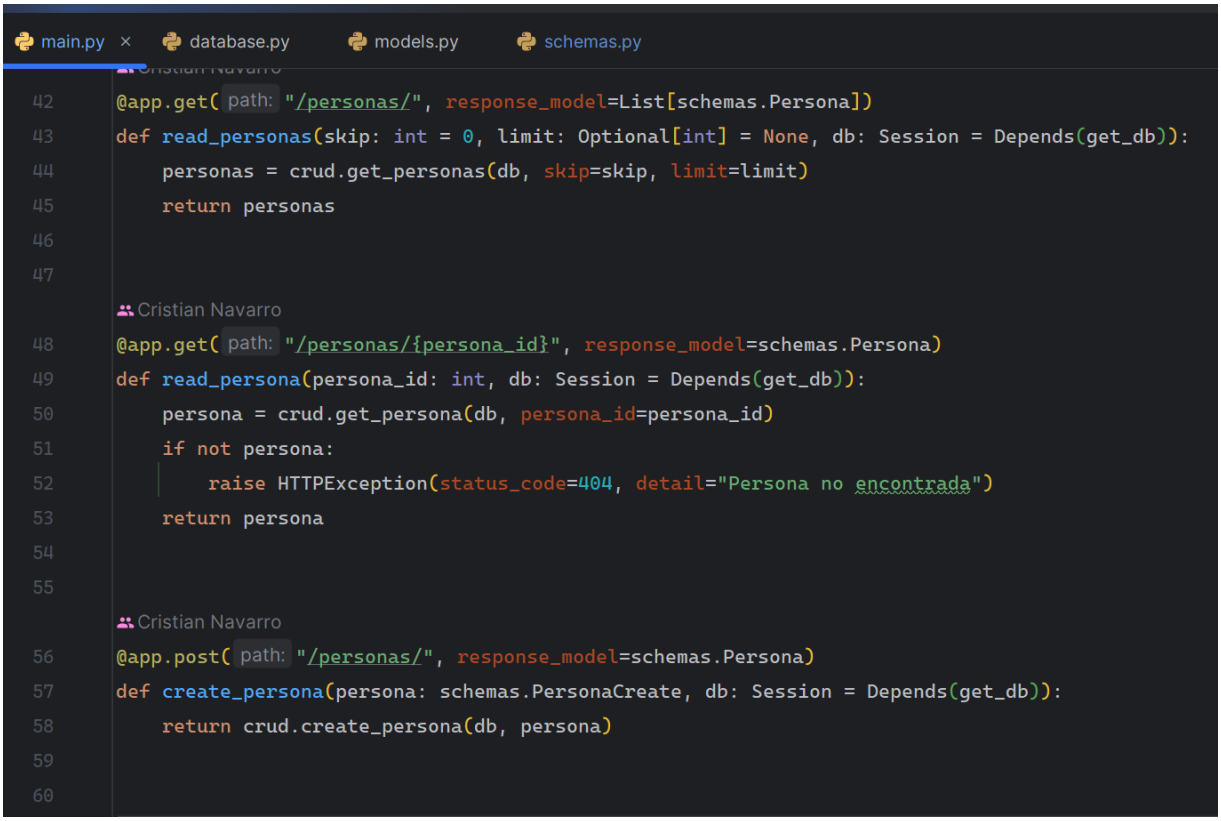


Figura 9

Creación de modelo el cual contiene la estructura de la base de datos

```python
class Persona(Base):
    __tablename__ = "personas"

    id = Column(Integer, primary_key=True, index=True)
    nombres = Column(String, index=True)
    cedula = Column(String, unique=True, index=True)
    email = Column(String, index=True)
    fecha_ingreso = Column(Date, nullable=True)
    fecha_retiro = Column(Date, nullable=True)
    tipo_contrato_id = Column(Integer, ForeignKey('tipos_contratos.id'))
    estado_id = Column(Integer, ForeignKey('estados.id'))
    area_id = Column(Integer, ForeignKey('areas.id'))
    sede_id = Column(Integer, ForeignKey('sedes.id'))
    compania_id = Column(Integer, ForeignKey('companias.id'))

    area = relationship( argument: "Area", back_populates="personas")
    tipo_contrato = relationship( argument: "TipoContrato", back_populates="personas")
    sede = relationship( argument: "Sede", back_populates="personas")
    compania = relationship( argument: "Compania", back_populates="personas")
    activos_asignados = relationship( argument: "PersonasActivos", back_populates="persona")
    estado = relationship( argument: "Estado", back_populates="personas")
    historial_asignaciones = relationship( argument: "HistorialAsignacion", back_populates="persona")
```

Figura 10

Schemas el cual define el tipo de datos

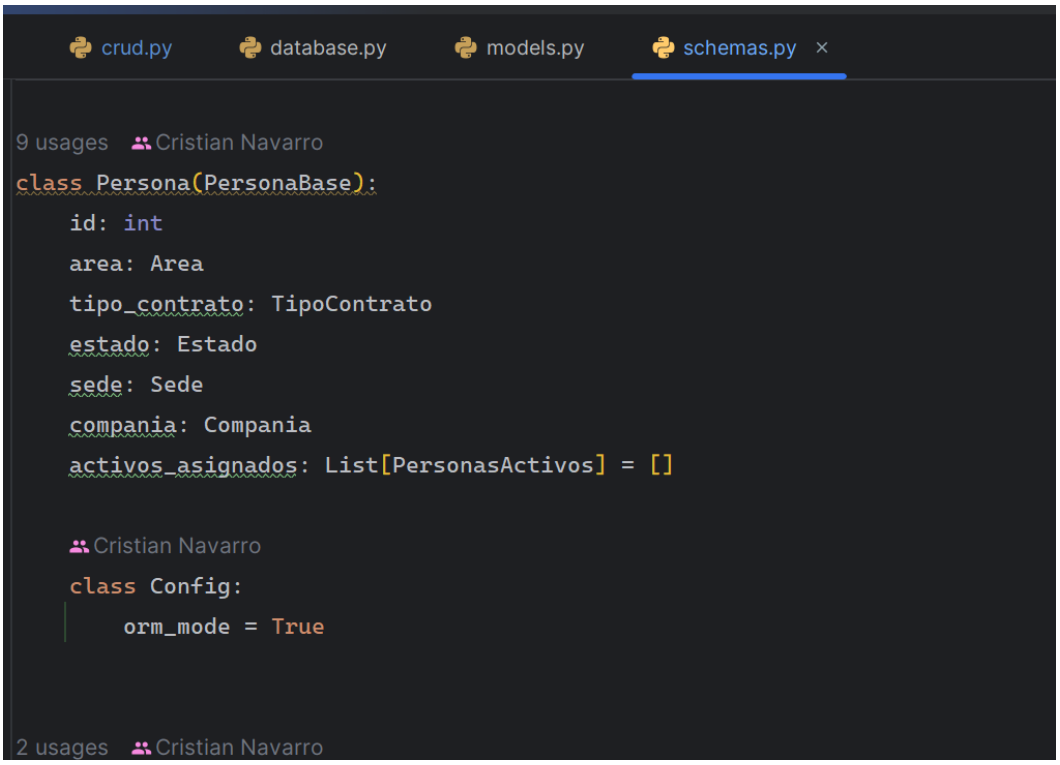
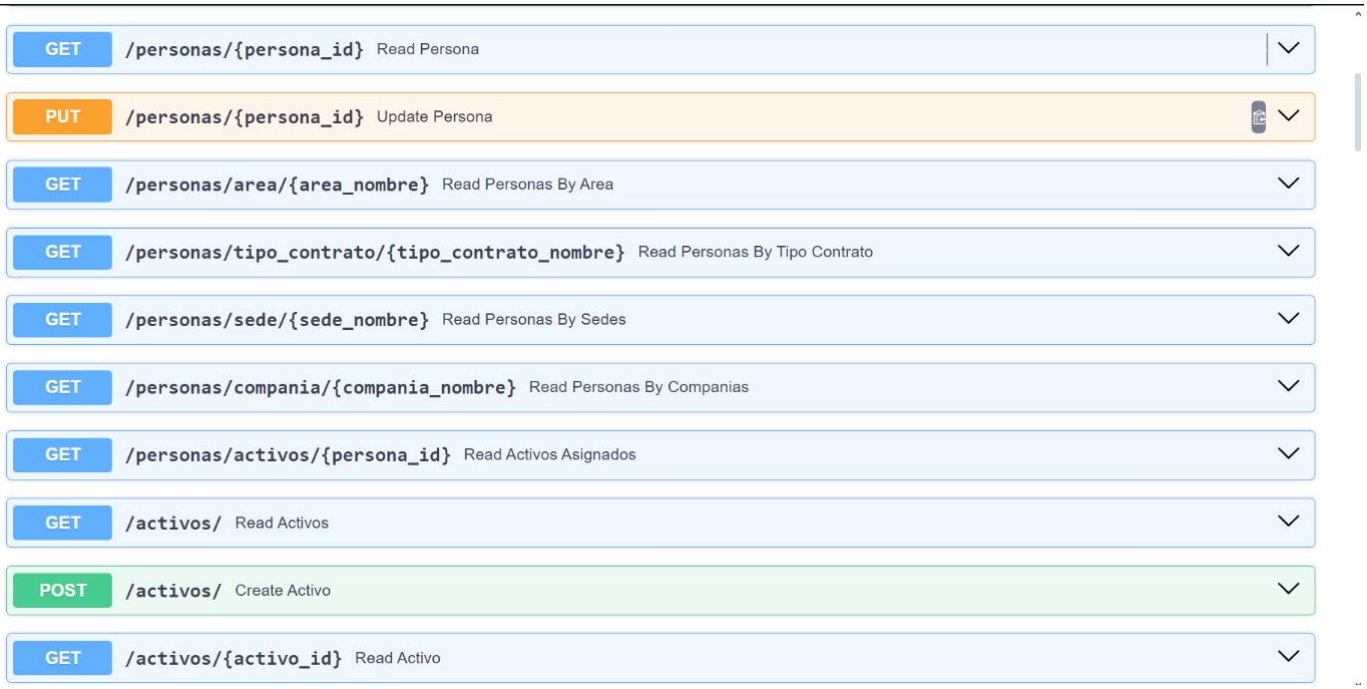


Figura 11

Documentación interactiva con el API



5.1.3 Consumo de API en JAVAFX utilizando Jackson

JavaFX es un framework de Java utilizado para desarrollar aplicaciones de escritorio. Jackson, por otro lado, es una biblioteca de procesamiento JSON en Java que proporciona varias formas de manejar datos JSON.

Figura 12

Clase Api Service consumo todos los endpoints

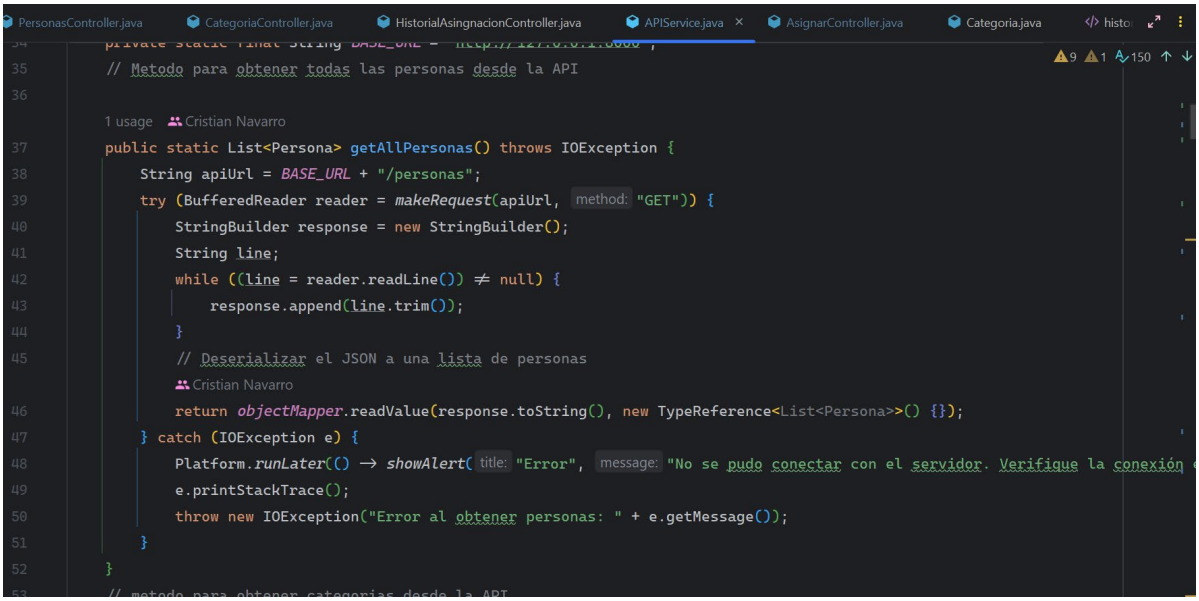


Figura 13

Consumo por número de activo

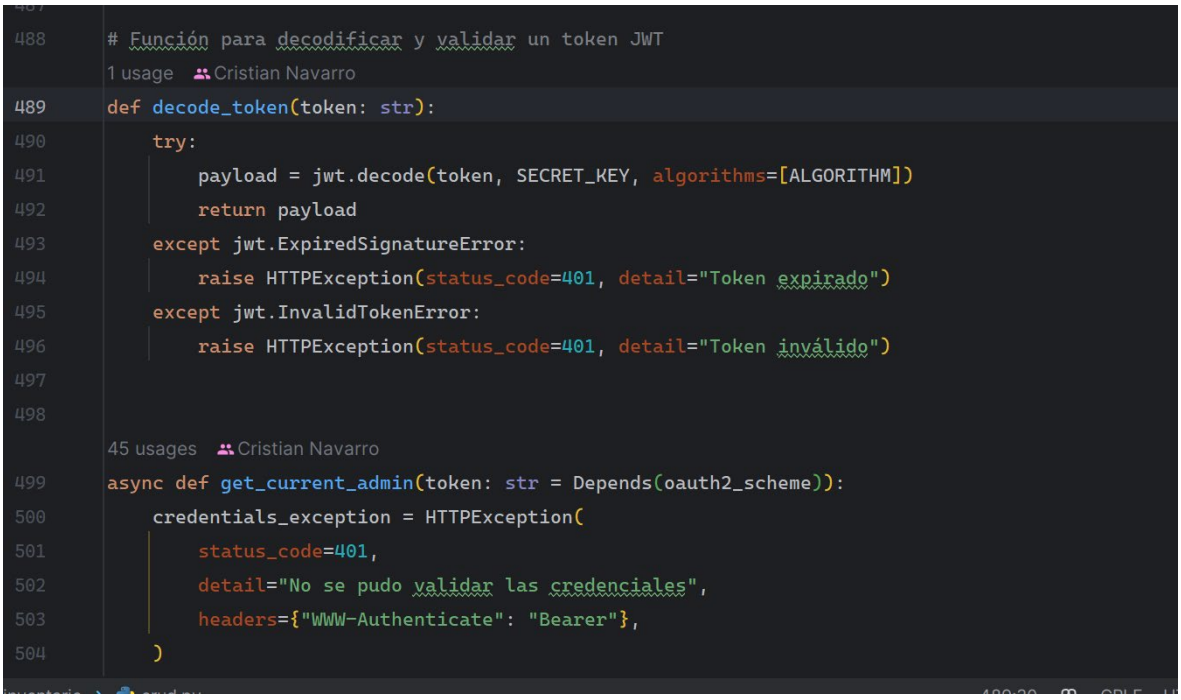


```
}
// metodo para obtener activos por numero de activo
    1 usage  Cristian Navarro
public static void getActivosNumero(String numeroActivo, Consumer<Categoria> callback) {
    new Thread(() -> {
        try {
            String apiUrl = BASE_URL + "/activos/" + "numero_activo/" + numeroActivo;
            URL url = new URL(apiUrl);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("GET");
            conn.setRequestProperty("Accept", "application/json");

            int responseCode = conn.getResponseCode();
            if (responseCode == HttpURLConnection.HTTP_OK) {
                try (BufferedReader reader = new BufferedReader(new InputStreamReader(conn.getInputStream(), StandardCharsets
                    StringBuilder response = new StringBuilder();
                    String line;
                    while ((line = reader.readLine()) != null) {
                        response.append(line.trim());
                    }
                    Categoria categoria = objectMapper.readValue(response.toString(), Categoria.class);
```

Figura 14

Implementación de JWT

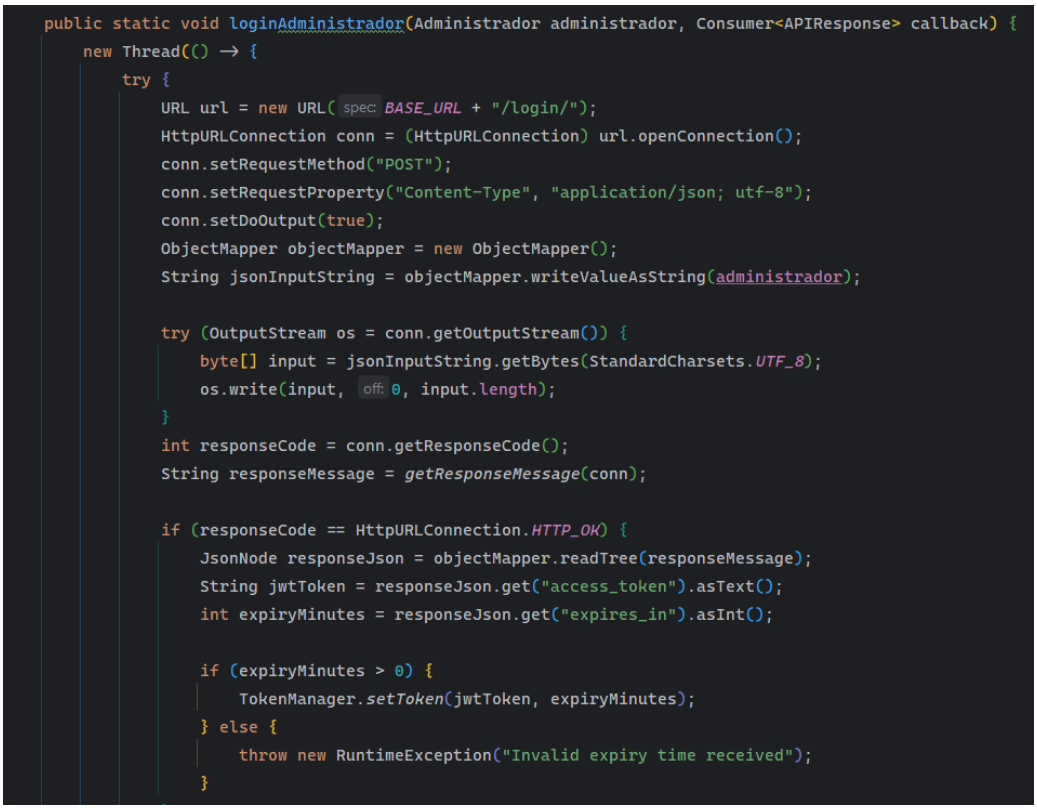


```
# Función para decodificar y validar un token JWT
1 usage  Cristian Navarro
def decode_token(token: str):
    try:
        payload = jwt.decode(token, SECRET_KEY, algorithms=[ALGORITHM])
        return payload
    except jwt.ExpiredSignatureError:
        raise HTTPException(status_code=401, detail="Token expirado")
    except jwt.InvalidTokenError:
        raise HTTPException(status_code=401, detail="Token inválido")


45 usages  Cristian Navarro
async def get_current_admin(token: str = Depends(oauth2_scheme)):
    credentials_exception = HTTPException(
        status_code=401,
        detail="No se pudo validar las credenciales",
        headers={"WWW-Authenticate": "Bearer"},
    )
```

Figura 15

Generación del token a partir del login



```
public static void loginAdministrador(Administrador administrador, Consumer<APIResponse> callback) {
    new Thread(() -> {
        try {
            URL url = new URL( spec: BASE_URL + "/login/");
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type", "application/json; utf-8");
            conn.setDoOutput(true);
            ObjectMapper objectMapper = new ObjectMapper();
            String jsonInputString = objectMapper.writeValueAsString(administrador);

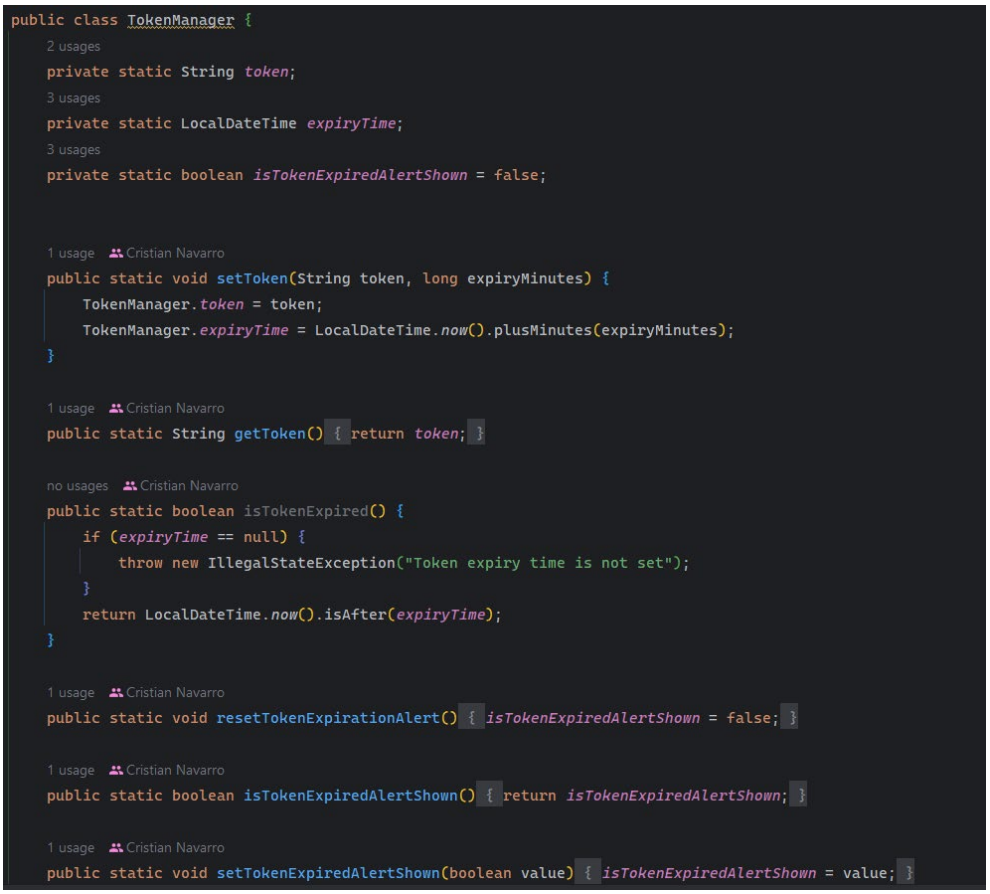
            try (OutputStream os = conn.getOutputStream()) {
                byte[] input = jsonInputString.getBytes(StandardCharsets.UTF_8);
                os.write(input,  off: 0, input.length);
            }
            int responseCode = conn.getResponseCode();
            String responseMessage = getResponseMessage(conn);

            if (responseCode == HttpURLConnection.HTTP_OK) {
                JsonNode responseJson = objectMapper.readTree(responseMessage);
                String jwtToken = responseJson.get("access_token").asText();
                int expiryMinutes = responseJson.get("expires_in").asInt();

                if (expiryMinutes > 0) {
                    TokenManager.setToken(jwtToken, expiryMinutes);
                } else {
                    throw new RuntimeException("Invalid expiry time received");
                }
```

Figura 16

Clase donde se maneja el token de JWT en java



```
public class TokenManager {
    2 usages
    private static String token;
    3 usages
    private static LocalDateTime expiryTime;
    3 usages
    private static boolean isTokenExpiredAlertShown = false;


    1 usage  Cristian Navarro
    public static void setToken(String token, long expiryMinutes) {
        TokenManager.token = token;
        TokenManager.expiryTime = LocalDateTime.now().plusMinutes(expiryMinutes);
    }

    1 usage  Cristian Navarro
    public static String getToken() { return token; }

    no usages  Cristian Navarro
    public static boolean isTokenExpired() {
        if (expiryTime == null) {
            throw new IllegalStateException("Token expiry time is not set");
        }
        return LocalDateTime.now().isAfter(expiryTime);
    }

    1 usage  Cristian Navarro
    public static void resetTokenExpirationAlert() { isTokenExpiredAlertShown = false; }

    1 usage  Cristian Navarro
    public static boolean isTokenExpiredAlertShown() { return isTokenExpiredAlertShown; }

    1 usage  Cristian Navarro
    public static void setTokenExpiredAlertShown(boolean value) { isTokenExpiredAlertShown = value; }
```

5.2 Diseñar e implementar una interfaz gráfica en JavaFX

Este diseño fue hecho JAVAFX utilizando fxml e implementación de JWT

Figura 17

Vista Login

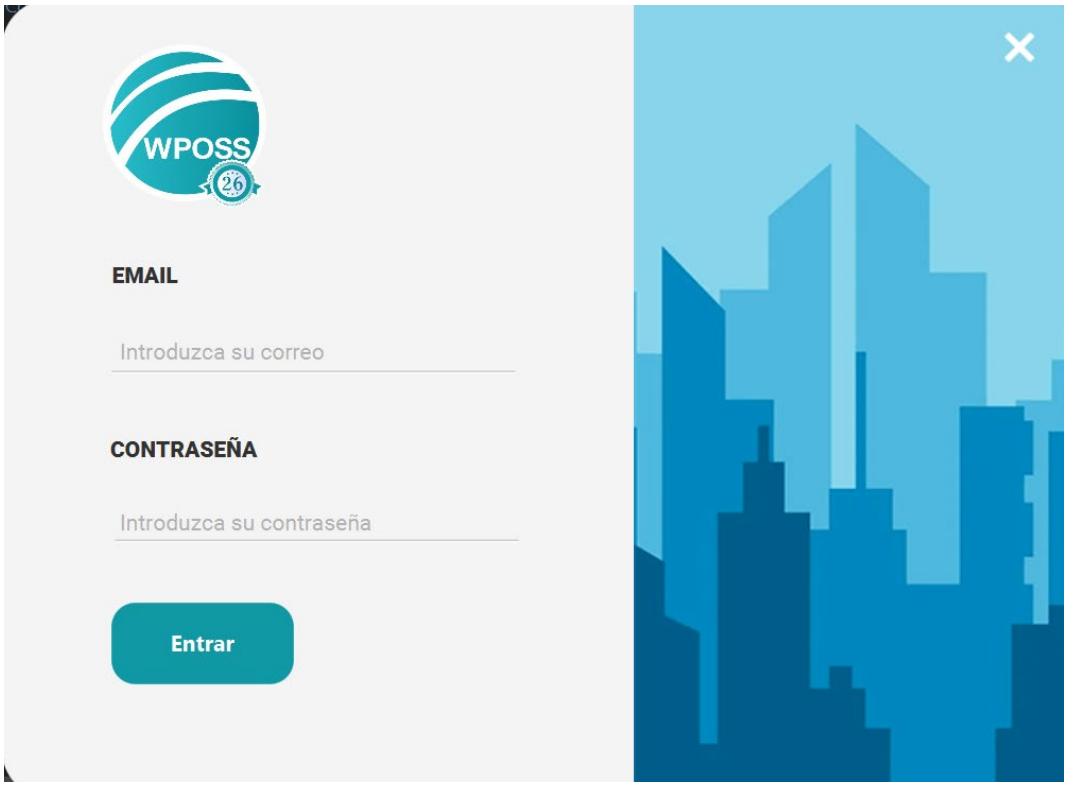


Figura 18

Vista Personas

En esta parte se maneja el consumo del endpoint de personas, trayendo los datos creados del API con la petición GET.

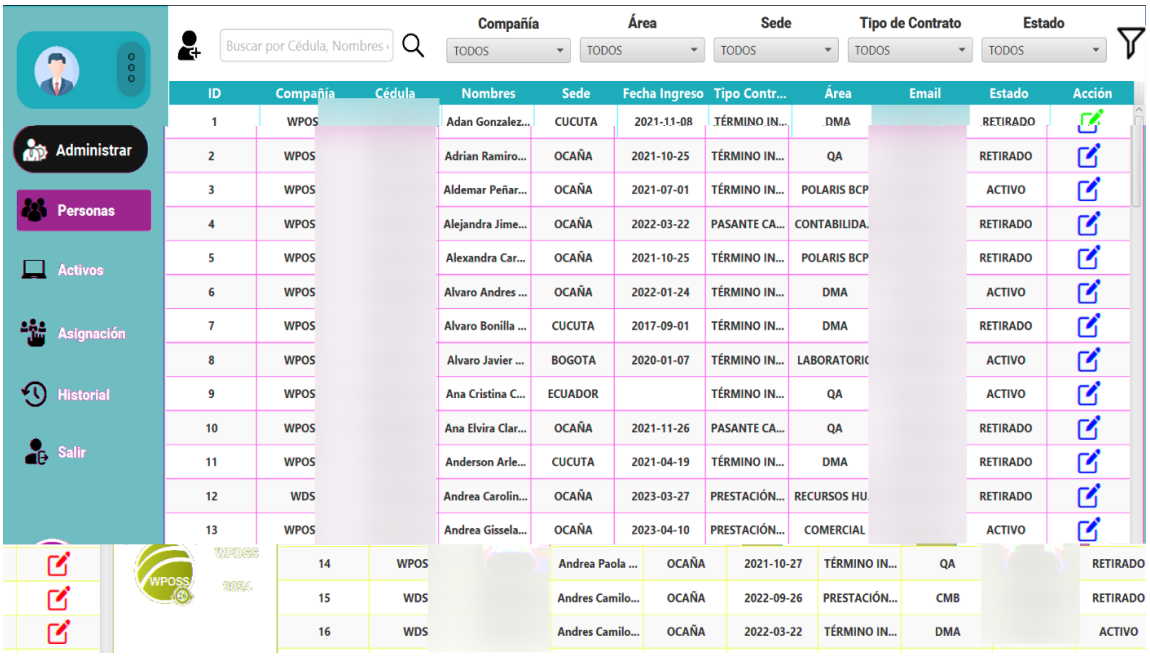
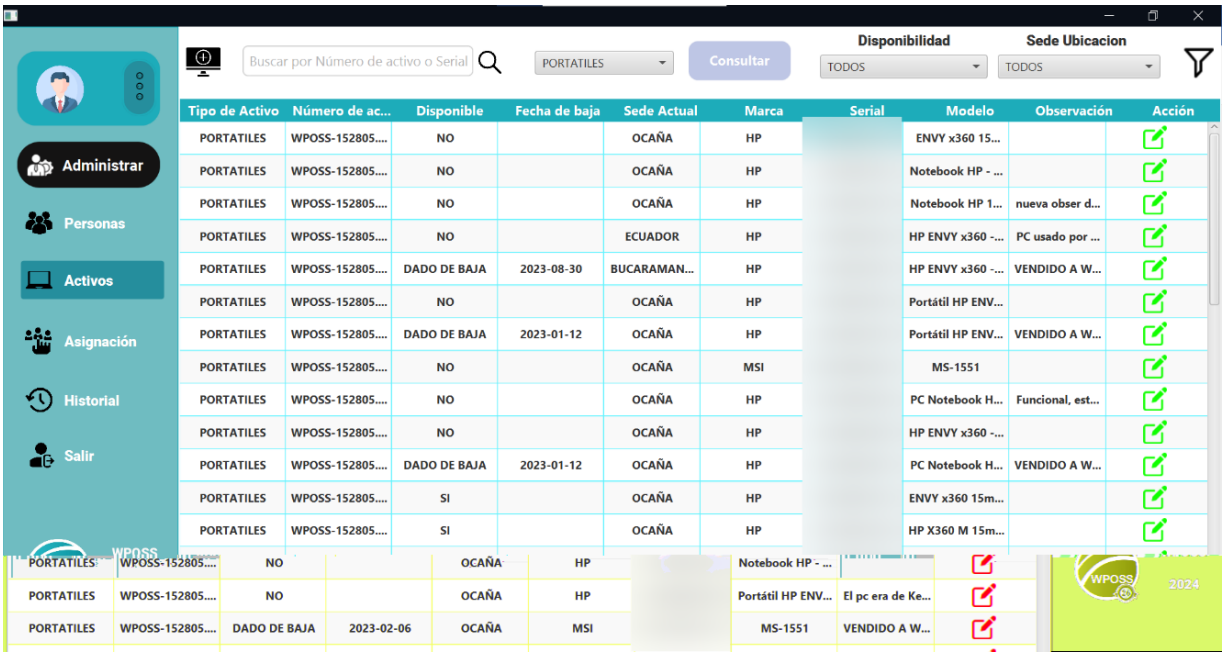


Figura 19

Vista Activos



5.3 Implementar funcionalidades de CRUD

Se traen datos de la persona que se desea editar y junto a los ComboBox que consumen los endpoints de la API, se seleccionan las opciones y se realiza la gestión de personas con la petición PUT.

Figura 20

Vista Editar personas

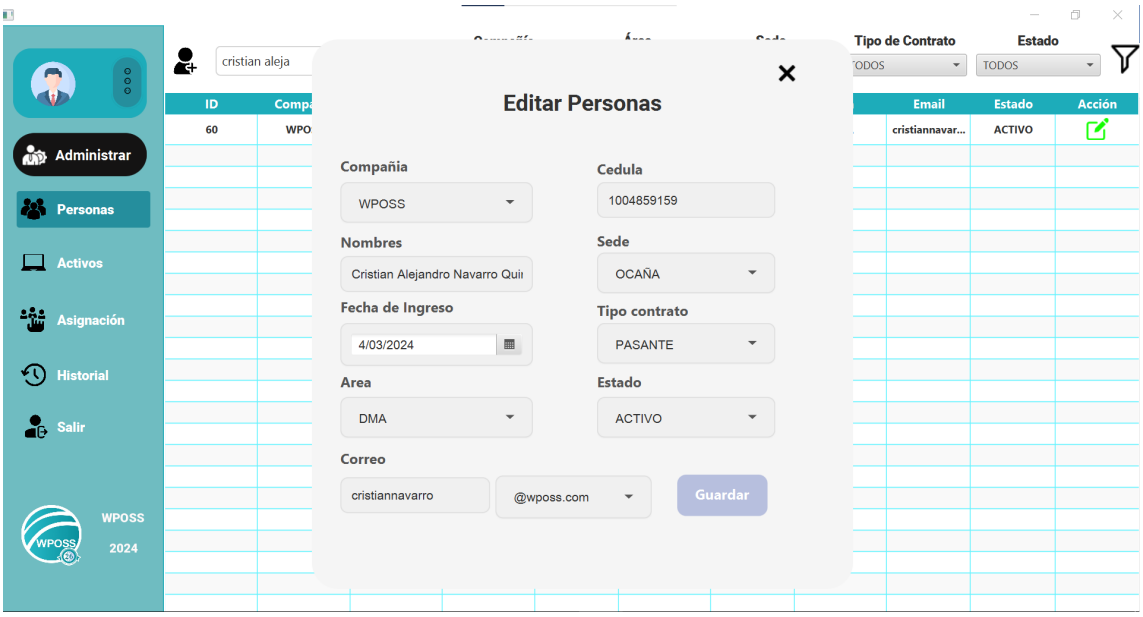


Figura 21

Vista Agregar Personas

Creación de personas mediante una ventana en la que se toman los datos a ingresar a hacemos una petición POST

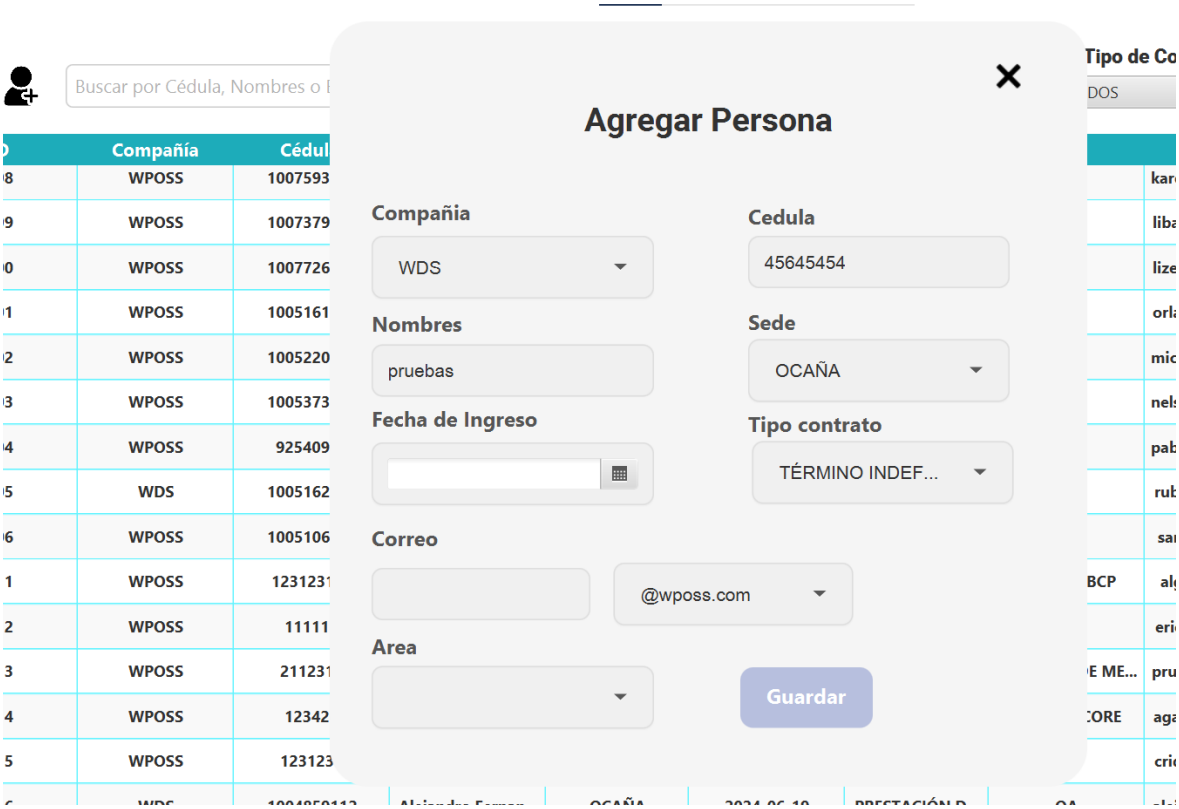


Figura 22

Vista Actualizar Activos

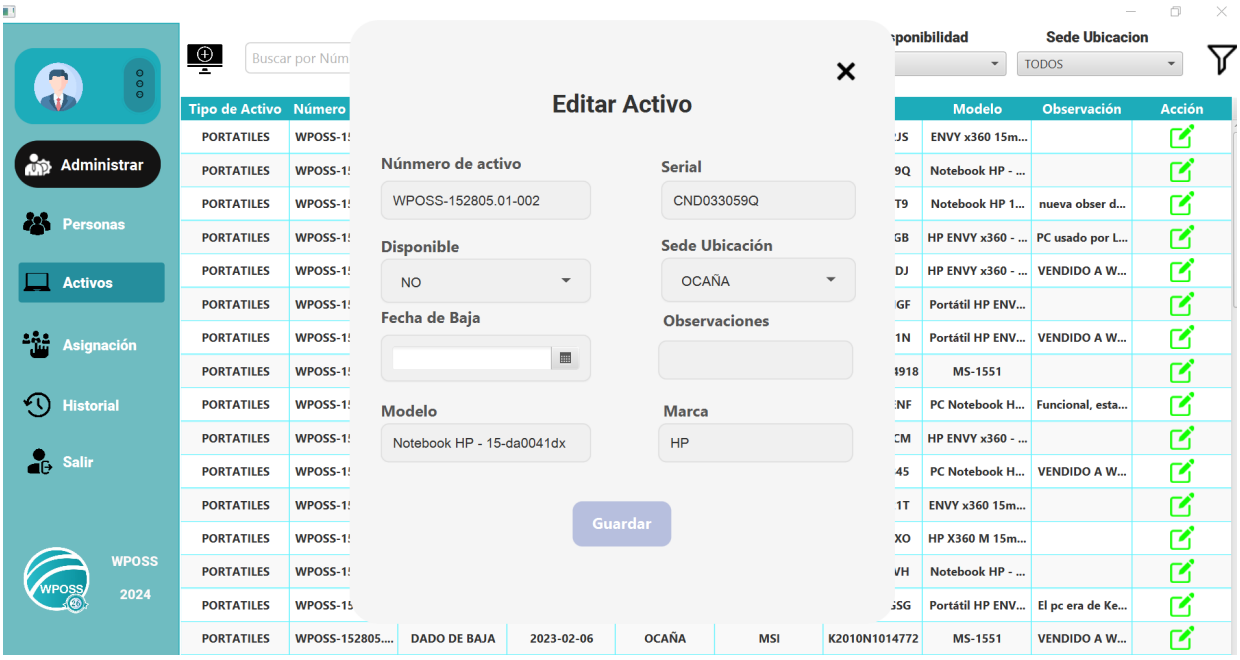
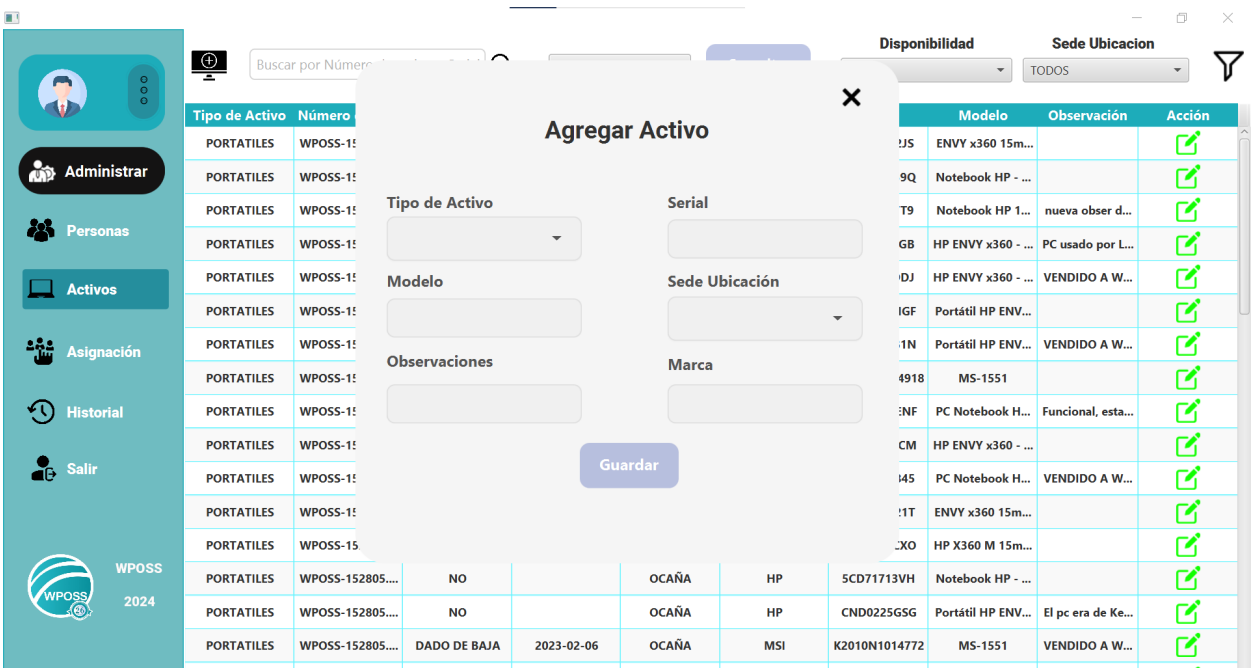


Figura 23

Vista Agregar Activos



5.4 Relación entre personas y activos, y el seguimiento de esta relación

Figura 24

Vista Asignación

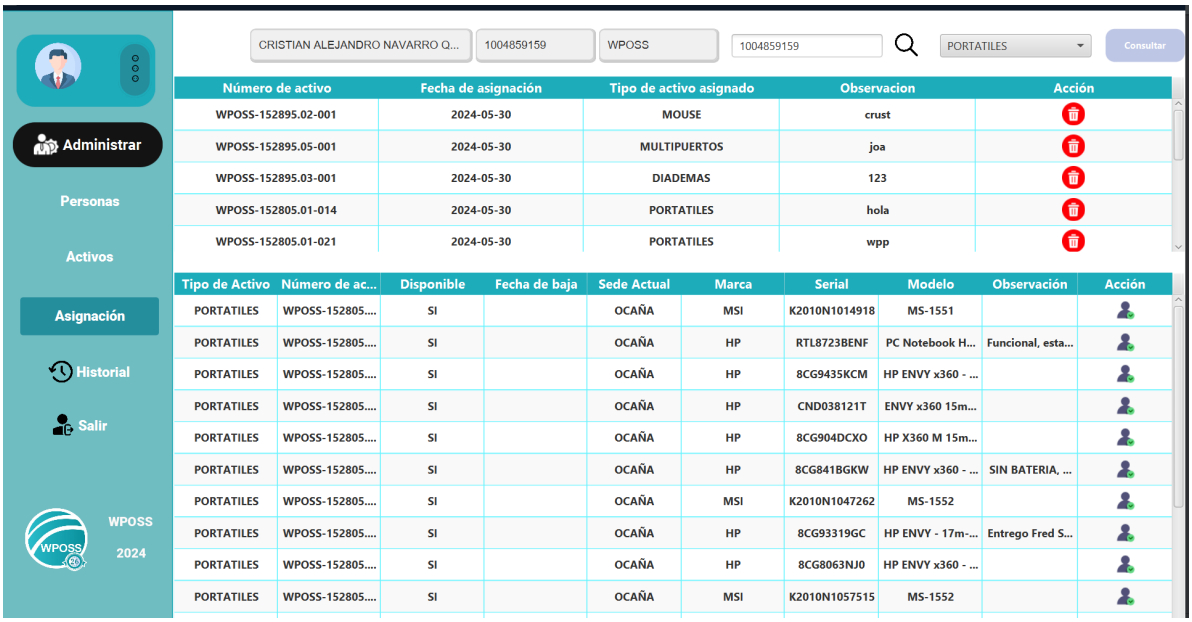


Figura 25

Agregar Asignación

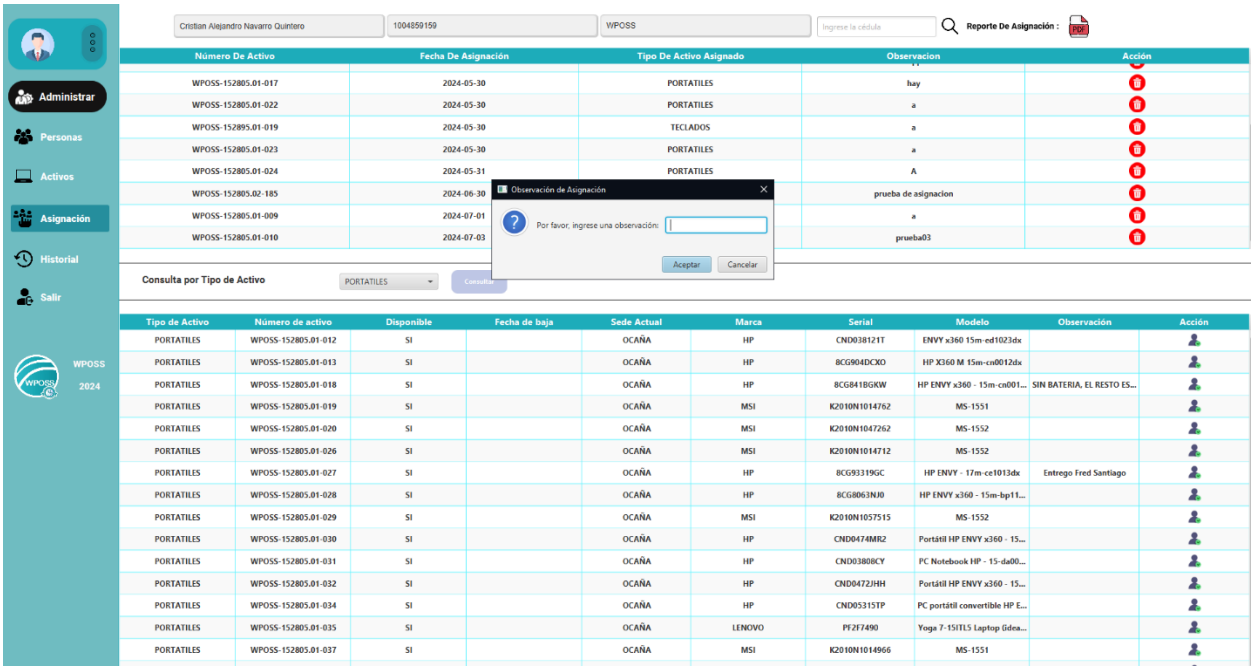
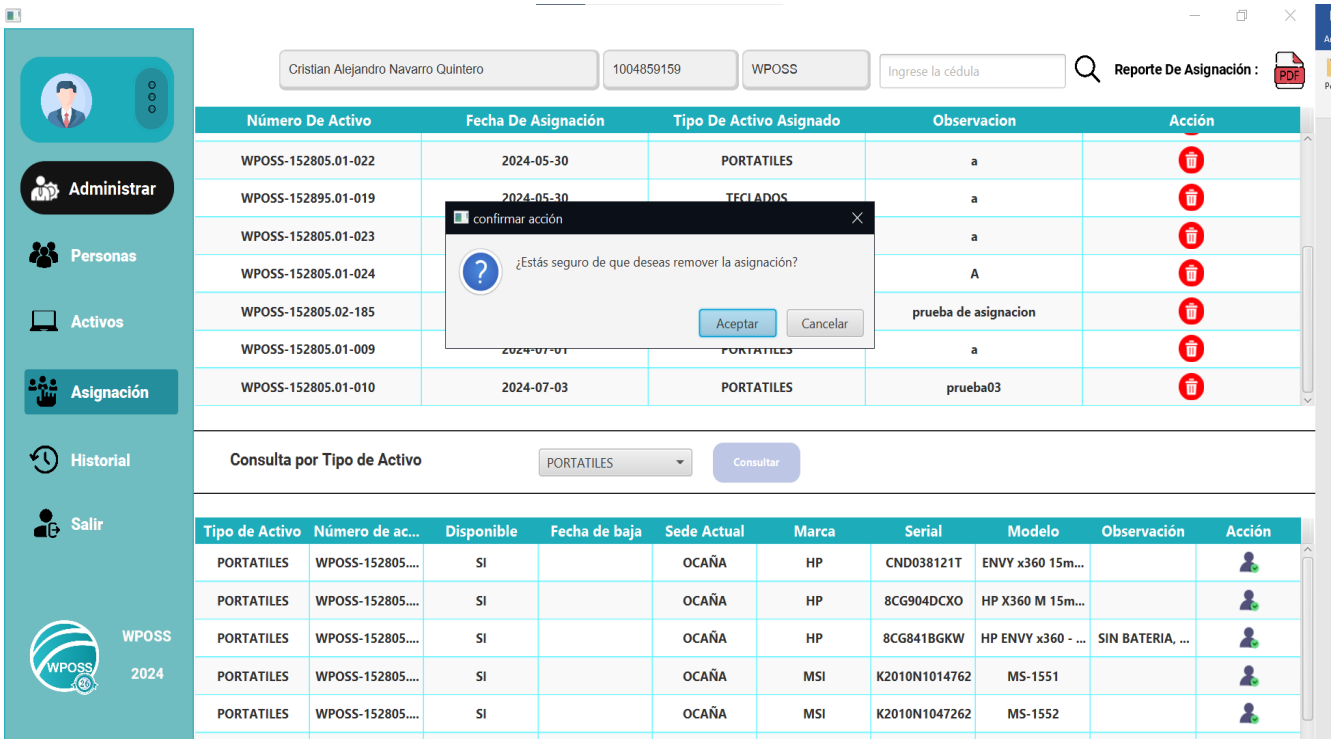


Figura 26

Eliminar Asignación



5.5 Generar un PDF de asignaciones

Figura 27

Vista Generar PDF Asignaciones

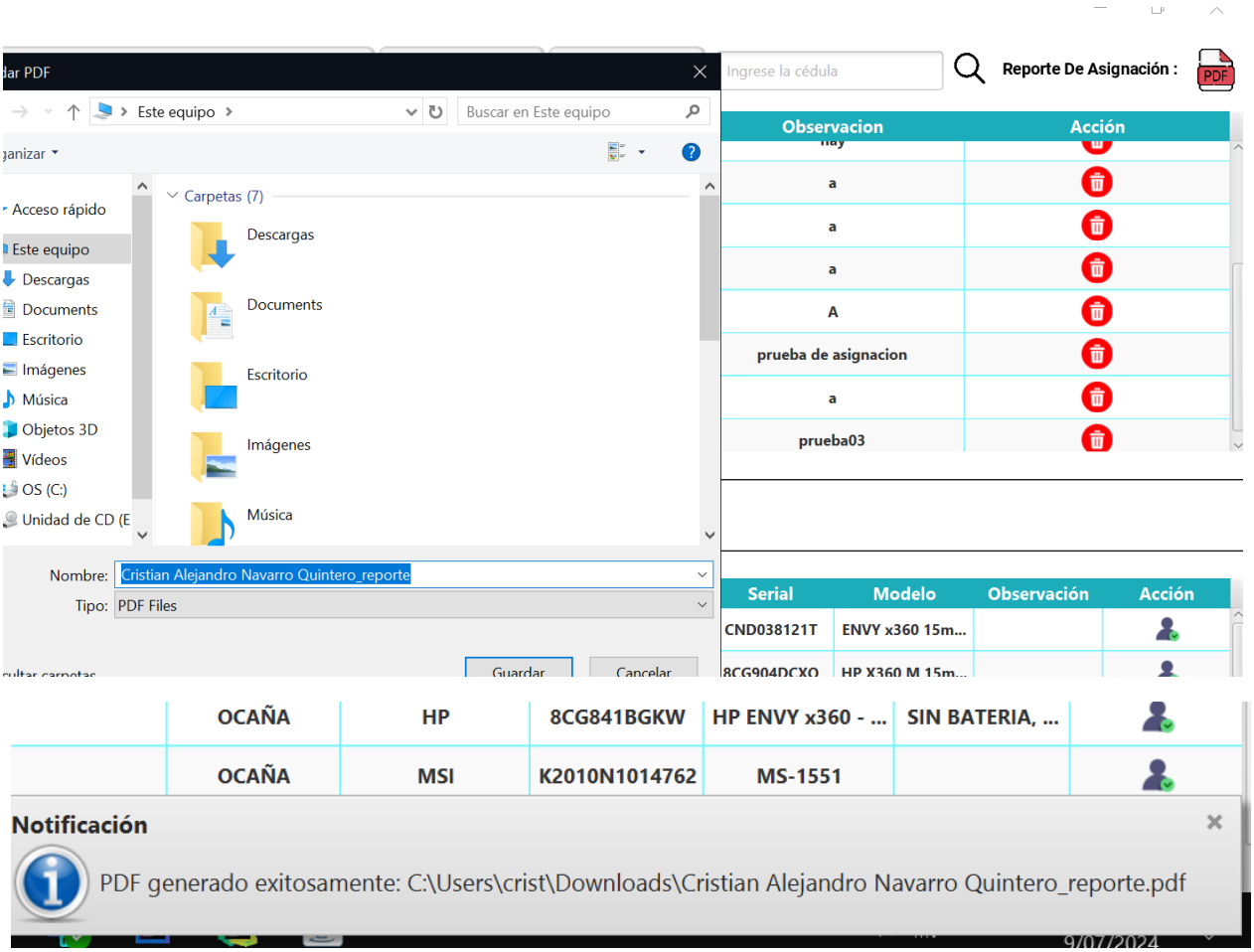


Figura 28

Reporte generado PDF Activos Asignados a personas

WORLD POS SOLUTIONS	ACTA DE ENTREGA DE ELEMENTOS DE TRABAJO	**Código:** ST-FO-001 **Versión:** 01 **Fecha:** 05/04/2022

NOMBRE COMPLETO: Cristian Alejandro Navarro Quintero	**TIPO DE VINCULACIÓN LABORAL:** PASANTE
CARGO: DMA **CELULAR:**	**DOCUMENTO DE IDENTIDAD:** 1004859159 **SEDE:** OCAÑA

A través de la presente hago constar la entrega de el o los siguientes elementos de propiedad de WORLD POS SOLUTIONS S.A.S., para ser usados como herramienta(s) de trabajo durante la duración de su contrato laboral.

TIPO DE ACTIVO	CÓDIGO DE ACTIVO	OBSERVACIÓN	CANTIDAD
MOUSE	WPOSS-152895.02-001	crust	1
MULTIPUERTOS	WPOSS-152895.05-001	joa	1
DIADEMAS	WPOSS-152895.03-001	123	1
PORTATILES	WPOSS-152805.01-021	wpp	1
PORTATILES	WPOSS-152805.01-017	hay	1
PORTATILES	WPOSS-152805.01-022	a	1
TECLADOS	WPOSS-152895.01-019	a	1
PORTATILES	WPOSS-152805.01-023	a	1
PORTATILES	WPOSS-152805.01-024	A	1
PANTALLAS	WPOSS-152805.02-185	prueba de asignacion	1
PORTATILES	WPOSS-152805.01-009	a	1
PORTATILES	WPOSS-152805.01-010	prueba03	1

OBSERVACIONES:

Entrega: Recibe:

Fecha de entrega de los elementos de trabajo: 2024-07-09

¡IMPORTANTE!

El trabajador es responsable del buen uso y el cuidado de los elementos de trabajo asignados de propiedad de la

5.6 Crear un historial de asignaciones de activos

Figura 29

Vista Historial



6 Diagnóstico Final

Durante mi pasantía en World Pos Solutions S.A.S. (WPOSS), he realizado varios aportes significativos y he implementado cambios que han mejorado los procesos y la eficiencia del área de soporte en la gestión de inventarios. A continuación, se describen los principales logros y modificaciones realizadas durante este periodo.

6.1.1 Desarrollo e Implementación de la Aplicación de Gestión de Inventario:

He desarrollado una aplicación de escritorio utilizando JavaFX, que permite una gestión más eficiente y precisa del inventario. Esta herramienta facilita la visualización en tiempo real de los niveles de inventario, optimizando el proceso de control y reduciendo el riesgo de errores humanos.

6.1.2 Creación y Consumo de API con FastAPI

Una de las contribuciones más importantes fue el desarrollo de una API utilizando FastAPI para manejar las operaciones CRU sobre los datos de inventario. Esta API permite una integración eficiente entre la base de datos y la aplicación de escritorio, proporcionando una plataforma robusta para la creación y actualización y gestión de inventarios en tiempo real. Este cambio ha permitido una mayor agilidad y precisión en la administración de los datos de inventario.

6.1.3 Migración de Datos a PostgreSQL

Se llevó a cabo la migración de datos desde un sistema basado en archivos Excel a una base de datos PostgreSQL. Este cambio ha permitido una gestión más segura y estructurada de

los datos de inventario, mejorando la accesibilidad y la integridad de la información. La migración incluyó la utilización de procesos ETL (Extracción, Transformación y Carga), asegurando que los datos se trasladaran de manera eficiente y precisa.

6.1.4 Implementación del Patrón de Diseño MVC

Para asegurar la modularidad y facilidad de mantenimiento del código, se implementó el patrón de diseño Model-View-Controller (MVC) en el desarrollo de la aplicación. Esta estructura ha facilitado la separación de la lógica de negocio, la interfaz de usuario y la gestión de datos, permitiendo un desarrollo más ordenado y una futura escalabilidad del sistema.

6.1.5 Implementación de reportes PDF

Se incorporó la funcionalidad de generación de reportes en formato PDF utilizando la biblioteca iTextPDF. Esta característica permite a los usuarios exportar los activos asignado a las personas, Los reportes incluyen detalles como el nombre completo, tipo de vinculación laboral, área, celular, y documento de identidad y activos asignados. Además, se aseguraron de que los PDFs generados sean compatibles con diferentes visores de PDF, manteniendo un diseño consistente y profesional.

7 Conclusiones

El desarrollo e implementación de la aplicación para la gestión de inventario en el área de soporte de WPOSS ha permitido una mejora significativa en la precisión y eficiencia del control de activos. La posibilidad de visualizar en tiempo real los niveles de inventario ha reducido los tiempos y la carga de trabajo asociados a la administración manual de inventarios.

La creación y consumo de una API con FastAPI ha proporcionado una solución robusta para la comunicación entre la base de datos PostgreSQL y la aplicación de escritorio desarrollada en JavaFX. Esta integración ha permitido la actualización en tiempo real de los datos de inventario, mejorando la exactitud y fiabilidad de la información manejada por WPOSS.

La implementación de funcionalidades CRUD ha facilitado a los usuarios modificar la información de personas y activos directamente desde la interfaz gráfica, mejorando la precisión y velocidad de las actualizaciones de datos. Esto ha asegurado una gestión más estructurada y segura de los datos, facilitando el acceso y la administración de la información de inventarios.

El desarrollo de una relación entre personas y activos ha permitido la asignación efectiva de activos a personas y el seguimiento de esta relación. La creación de un historial detallado de las asignaciones de activos ha registrado cada movimiento, permitiendo una trazabilidad completa de los activos a lo largo del tiempo.

Finalmente, la generación de PDFs que contienen reportes de activos asignados a las personas ha mejorado la capacidad de WPOSS para documentar y compartir información relevante sobre el estado de sus activos.

En resumen, todos los objetivos específicos y el objetivo general del proyecto han sido alcanzados con éxito. La aplicación desarrollada cumple con los requisitos de WPOSS, mejorando la eficiencia y precisión en la gestión de inventarios en el área de soporte.

8 Recomendaciones

A partir de los resultados obtenidos y del análisis realizado durante la pasantía, se presentan las siguientes recomendaciones para continuar mejorando la gestión de inventarios en WPOSS:

Se recomienda mejorar las funcionalidades de la aplicación de gestión de inventarios para incluir características adicionales como la integración con sistemas de venta, permitiendo una sincronización automática de las entradas y salidas de inventarios.

Es crucial mantener un programa de capacitación continua para el personal encargado de la gestión de inventarios. Esto asegurará que los empleados estén actualizados con las nuevas funcionalidades de la aplicación y las mejores prácticas en la gestión de inventarios.

Establecer un sistema de monitoreo y evaluación constante de los procesos de gestión de inventarios para identificar áreas de mejora y asegurar el correcto funcionamiento de la aplicación. Esto puede incluir la implementación de auditorías regulares y revisiones periódicas de los procedimientos.

Aunque la migración de datos actual ha sido exitosa, se recomienda desarrollar un protocolo de migración de datos más robusto que incluya pruebas exhaustivas y validaciones adicionales para garantizar la integridad y precisión de los datos migrados.

Reforzar las medidas de seguridad en la gestión de datos de inventarios, incluyendo la encriptación de datos y la implementación de un sistema de control de accesos que restrinja el acceso a los datos según el rol y las responsabilidades de los usuarios.

Mantenerse al tanto de las nuevas tecnologías y tendencias en la gestión de inventarios, como el uso de inteligencia artificial para predicciones de inventario y blockchain para la trazabilidad de productos. Evaluar la posibilidad de integrar estas tecnologías en el futuro para mejorar aún más la eficiencia y precisión del sistema.

Estas recomendaciones están diseñadas para asegurar que WPOSS continúe mejorando su sistema de gestión de inventarios, optimizando procesos y mejorando tiempo de respuesta.

9 Referencias

¿Qué es Android? ((s.f.)). Obtenido de https://www.android.com/intl/es-419_mx/what-is-android/

3.12.4 documentation Python. ((s.f.)). Obtenido de Python.org: https://docs.python.org/3/

Castro Zuluaga, C. U. (2014). *Marco de referencia para el desarrollo de un sistema de apoyo para la toma de decisiones para la gestión de inventarios.* Corporación Universidad de la Costa. Obtenido de http://hdl.handle.net/11323/1836

Chopra, S. &. (2016). *Supply Chain Management: Strategy, Planning, and Operation.* Editorial: Pearson. Obtenido de https://www.pearson.com/us/higher-education/program/Chopra-Supply-Chain-Management-Strategy-Planning-and-Operation-6th-Edition/PGM216866.html

Connolly, T. &. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management.* Pearson. Obtenido de https://www.pearson.com/store/p/database-systems-a-practical-approach-to-design-implementation-and-management/P100000368463/9780132943260

Corvo, H. S. (31 de agosto de 2023). *Sistema de inventarios*. Obtenido de https://www.lifeder.com/sistema-de-inventarios/

Enrique, V. C. (1 de diciembre de 2022). *Repositorio Institucional Universidad Cooperativa de Colombia*. Obtenido de https://repository.ucc.edu.co/entities/publication/0853592c-012e-413a-984b-677a76505787

FAST API. ((s.f.)). Obtenido de Devdocs.Io: https://devdocs.io/fastapi/

Getting started with JavaFX. ((s.f.)). Obtenido de Openjfx.Io: https://openjfx.io/openjfx-docs/

Jackson-core 2.17.1 javadoc (com.Fasterxml.Jackson.Core). ((n.d.)). Obtenido de https://javadoc.io/doc/com.fasterxml.jackson.core/jackson-core/latest/index.html

Meet Android Studio. ((s.f.)). Obtenido de https://developer.android.com/studio/intro

Oracle Help Center. (31 de enero de 2023). Obtenido de Java Documentation - Get started: https://docs.oracle.com/en/java/

Waller, M. A. (2013). *Data science, predictive analytics, and big data: A revolution that will transform supply chain design and management.* Journal of Business Logistics, 34(2), 77-84. Obtenido de https://onlinelibrary.wiley.com/doi/abs/10.1111/jbl.12010

WPOSS. (2021). Obtenido de https://www.wposs.com/quienes-somos/