


Hardware Store Inventory

MANUAL TÉCNICO DEL SISTEMA

Versión 1.0

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

Integrantes

Juan David Gallego López

Diseñador UI, Desarrollador Backend y Frontend

David Ricardo Graffe Rodríguez

Diseñador UX. Desarrollador Backend y Frontend

Joaquín Humberto Jiménez Rosas

Diseñador Base De Datos, Desarrollador Backend y Frontend



	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

Tabla de contenido

1	RESUMEN	1
2	INTRODUCCIÓN	2
3	ASPECTOS TÉCNICOS	3
3.1	HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO.....	3
3.1.1	Visual Studio Code	3
3.1.2	IntelliJ IDEA	4
3.1.3	Spring Boot.....	5
3.1.4	React	6
3.1.5	Node.js	6
3.1.6	Vite	7
3.1.7	Git.....	8
3.1.8	GitHub	9
3.1.9	MySQL.....	10
3.1.10	MySQL Server	10
3.1.11	MySQL Workbench	10
3.2	PRERREQUISITOS DE INSTALACIÓN	11
3.3	FRAMEWORKS Y ESTÁNDARES UTILIZADOS.....	12
3.3.1	Lenguajes de programación:.....	12
3.3.2	Frameworks y tecnologías:	12
3.3.3	Estándares y patrones de diseño:.....	12
4	DIAGRAMAS DE CASO DE USO	13
4.1	Caso de Uso No. 1	13
4.2	Caso de Uso No. 2	15
4.3	Caso de Uso No. 3	17
4.4	Caso de Uso No. 4	19
4.5	Caso de Uso No. 5	21
4.6	Caso de Uso No. 6	23
4.7	Caso de Uso No. 7	25
4.8	Caso de Uso No. 8	27
4.9	Caso de Uso No. 9	29
4.10	Caso de Uso No. 10	31

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

5	MODELO RELACIONAL DE LA BASE DE DATOS	33
6	DICCIONARIO DE DATOS	34
7	SCRIPTS DE INSTALACIÓN	39
7.1	Backend – Spring Boot con JPA	39
7.2	Creación manual de la base de datos.....	40
7.3	Configuración de conexión (application.properties)	41
7.4	Ejecución del backend.....	42
7.5	Frontend – React + Vite.....	43
7.6	Instalación de dependencias	44
7.7	Ejecución del proyecto.....	45
7.8	Dependencias utilizadas	46
7.9	Nota aclaratoria sobre Vite	47
8	DIAGRAMA DE COMPONENTES	48
8.1	Diagrama de Componente General	48
8.2	Nivel 1: Vista global (sistemas o capas).....	48
8.3	Nivel 2: Componentes internos del backend Dentro del backend, puedes tener dos componentes grandes:	49
8.4	Diagrama de componente modulo usuarios.....	50
8.5	Diagrama de componente módulo inventario.....	50
9	BIBLIOGRAFIA.....	51


	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

Tabla 1: Prerrequisitos de instalación	11
Tabla 2: Dependencias utilizadas	11
Tabla 3: Caso de uso Administrar Usuarios, Fuente: JuDaJo System Soft (autores)	13
Tabla 4: Caso de uso Crear Usuarios, Fuente: JuDaJo System Soft (autores).....	15
Tabla 5: Caso de uso Consultar Usuarios, Fuente: JuDaJo System Soft (autores)	17
Tabla 6: Caso de uso Actualizar Usuarios, Fuente: JuDaJo System Soft (autores)	19
Tabla 7: Caso de uso Eliminar Usuarios, Fuente: JuDaJo System Soft (autores)	21
Tabla 8: Caso de uso Administrar Inventario, Fuente: JuDaJo System Soft (autores)	23
Tabla 9: Caso de uso Registrar Inventario, Fuente: JuDaJo System Soft (autores).....	25
Tabla 10: Caso de uso Consultar Inventario, Fuente: JuDaJo System Soft (autores).....	27
Tabla 11: Caso de uso Actualizar Inventario, Fuente: JuDaJo System Soft (autores)	29
Tabla 12: Caso de uso Eliminar Inventario, Fuente: JuDaJo System Soft (autores).....	31
Tabla 13: Diccionario de datos entidad categoría, Fuente: JuDaJo System Soft (autores)	35
Tabla 14: Diccionario de datos entidad empleado, Fuente: JuDaJo System Soft (autores)	35
Tabla 15: Diccionario de datos entidad movimiento_inventario, Fuente: JuDaJo System Soft (autores)...	36
Tabla 16: Diccionario de datos entidad perfil, Fuente: JuDaJo System Soft (autores)	36
Tabla 17: Diccionario de datos entidad permiso, Fuente: JuDaJo System Soft (autores)	36
Tabla 18: Diccionario de datos entidad producto, Fuente: JuDaJo System Soft (autores).....	37
Tabla 19: Diccionario de datos entidad producto_proveedor, Fuente: JuDaJo System Soft (autores)	37
Tabla 20: Diccionario de datos entidad proveedor, Fuente: JuDaJo System Soft (autores)	37
Tabla 21: Diccionario de datos entidad rol, Fuente: JuDaJo System Soft (autores)	38
Tabla 22: Diccionario de datos entidad rol_permiso, Fuente: JuDaJo System Soft (autores)	38
Tabla 23: Diccionario de datos entidad tipo_documento, Fuente: JuDaJo System Soft (autores)	38
Tabla 24: Diccionario de datos entidad usuario, Fuente: JuDaJo System Soft (autores)	38




	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

Imagen 1: Diagrama UML caso de uso Administrar Usuarios, Fuente: JuDaJo System Soft (autores)	14
Imagen 2: Diagrama UML caso de uso Crear Usuarios, Fuente: JuDaJo System Soft (autores).....	16
Imagen 3: Diagrama UML caso de uso Consultar Usuarios, Fuente: JuDaJo System Soft (autores).....	18
Imagen 4: Diagrama UML caso de uso Actualizar Usuarios, Fuente: JuDaJo System Soft (autores)	20
Imagen 5: Diagrama UML caso de uso Eliminar Usuarios, Fuente: JuDaJo System Soft (autores)	22
Imagen 6: Diagrama UML caso de uso Administrar Inventario, Fuente: JuDaJo System Soft (autores)	24
Imagen 7: Diagrama UML caso de uso Registrar Inventario, Fuente: JuDaJo System Soft (autores).....	26
Imagen 8: :Diagrama UML caso de uso Consultar Inventario, Fuente: JuDaJo System Soft (autores).....	28
Imagen 9: Diagrama UML caso de uso Actualizar Inventario, Fuente: JuDaJo System Soft (autores)	30
Imagen 10: Diagrama UML caso de uso Eliminar Inventario, Fuente: JuDaJo System Soft (autores).....	32
Imagen 11: Diagrama MER proyecto Hardware Store Inventory, Fuente: JuDaJo System Soft (autores)...	33
Imagen 12: Script para conocer la versión de Java y salida por consola.....	39
Imagen 13: Configuración Spring boot Initializr.....	39
Imagen 14: Script SQL para la creación de la base de datos en Workbench	40
Imagen 15: Configuración para la conexión BD en la API Rest.	41
Imagen 16: Creación de las entidades mapeadas por JPA desde la API Backend.....	42
Imagen 17: Script para ejecutar la API Backend desde consola.....	42
Imagen 18: Scripts para conocer la versión instalada de Node.js y npm, salida en consola.	43
Imagen 19: Script para instalar las dependencias y librerías necesarias para el API Frontend.	44
Imagen 20: Script para ejecutar el servidor local del API Frontend.....	45
Imagen 21: Dependencias instaladas y usadas en el API Frontend.	46
Imagen 22: Diagrama UML componente general del sistema Hardware Store Inventory, Fuente: JuDaJo System Soft (autores)	48
Imagen 23: Diagrama UML componente módulo usuario proyecto Hardware Store Inventory, Fuente: JuDaJo System Soft (autores)	50
Imagen 24: : Diagrama UML componente módulo inventario proyecto Hardware Store Inventory, Fuente: JuDaJo System Soft (autores)	50

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/08	1.0	DT-MTS-01

1 RESUMEN

El presente manual describe los aspectos técnicos e informáticos del software Hardware Store Inventory, con el objetivo de proporcionar al personal encargado una comprensión detallada de su estructura y funcionamiento. A través de esta documentación, los administradores podrán conocer las herramientas utilizadas en su desarrollo, así como su operatividad, permitiéndoles obtener una visión integral del sistema desde sus dimensiones lógica, funcional y física.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01


2 INTRODUCCIÓN

Este manual técnico documenta el desarrollo y funcionamiento del sistema de inventario y gestión de usuarios **Hardware Store Inventory FFIG**, un proyecto formativo para el SENA. Su propósito es proporcionar a los usuarios una guía detallada sobre la instalación, configuración y operación del sistema, asegurando una implementación eficiente y alineada con los estándares tecnológicos actuales.

El sistema está dirigido a **ferreterías pequeñas y medianas**, que buscan modernizar sus procesos de gestión de inventarios y administración de empleados. Actualmente, muchas de estas empresas aún manejan sus registros de manera manual, lo que puede generar errores, pérdida de información y dificultades en la consulta de datos en tiempo real.

Hardware Store Inventory FFIG resuelve este problema al ofrecer una plataforma digital que permite un **control automatizado del inventario y una gestión eficiente de los datos de los empleados**, facilitando la consulta rápida y segura de la información. Gracias a esta sistematización, los negocios pueden optimizar su tiempo, mejorar la toma de decisiones y aumentar la productividad.

El manual abarca aspectos clave como los **prerrequisitos de instalación**, los **frameworks y estándares utilizados**, el **diagrama de casos de uso**, el **modelo relacional de la base de datos**, el **diccionario de datos**, los **scripts de instalación** y el **diagrama de componentes**. Cada uno de estos puntos está diseñado para ofrecer una comprensión clara de la arquitectura y funcionamiento del sistema, facilitando su adopción y mantenimiento.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/10		1.0	DT-MTS-01

3 ASPECTOS TÉCNICOS

El aplicativo web está diseñado para optimizar el proceso de gestión de inventario en la Ferretería Ferro Industrias Ger, así como en cualquier otra ferretería que lo implemente. Además, facilita la administración del personal que opera dentro de la empresa y utiliza la plataforma.

Se recomienda que este manual sea consultado exclusivamente por el administrador del sistema o por la persona encargada de gestionar el software. Es importante que el responsable posea conocimientos de nivel intermedio-avanzado en sistemas, ya que deberá garantizar la seguridad y protección de los datos almacenados en la base de datos.

3.1 HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO


En esta sección se procede a explicar las herramientas informáticas empleadas para el desarrollo del aplicativo web.

3.1.1 Visual Studio Code

(VS Code) es un **editor de código fuente** desarrollado por Microsoft. Es ligero, potente y **gratuito**, diseñado para facilitar la programación en diversos lenguajes como Python, JavaScript, C++, entre muchos otros.

¿Por qué es tan popular?

- **Multiplataforma:** Funciona en Windows, macOS y Linux.
- **Extensiones:** Cuenta con una gran cantidad de complementos para ampliar sus funcionalidades.
- **Integración con Git:** Permite administrar versiones de código de manera eficiente.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/10		1.0	DT-MTS-01

- **Depuración integrada:** Facilita la detección y solución de errores dentro del código.
- **Editor inteligente:** Ofrece autocompletado, resaltado de sintaxis y sugerencias basadas en inteligencia artificial.


Es ideal tanto para desarrolladores principiantes como avanzados, ya que combina simplicidad con herramientas robustas

3.1.2 IntelliJ IDEA

Es un **entorno de desarrollo integrado (IDE)** creado por JetBrains, especializado en programación con Java, aunque también admite otros lenguajes como Kotlin, Python y JavaScript. Es ampliamente utilizado por desarrolladores gracias a sus herramientas avanzadas para la escritura, depuración y optimización de código.

¿Por qué es tan potente?

- **Soporte para múltiples lenguajes:** Ideal para aplicaciones Java y otros lenguajes modernos.
- **Análisis de código inteligente:** Ofrece sugerencias automáticas para mejorar la eficiencia y corregir errores.
- **Integración con Git:** Facilita el control de versiones y la colaboración en proyectos.
- **Depuración avanzada:** Incluye herramientas para inspección de código, pruebas y corrección de bugs.
- **Extensiones y plugins:** Permite personalizar el entorno de trabajo según las necesidades del desarrollador.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

Es una opción muy potente para desarrolladores que buscan productividad y una experiencia fluida al programar.


3.1.3 Spring Boot

Es un **framework basado en Java** que facilita el desarrollo de aplicaciones robustas y escalables sin necesidad de configuraciones complejas. Es parte del ecosistema **Spring**, diseñado para agilizar la creación de aplicaciones empresariales con una estructura modular y eficiente.

¿Por qué es tan útil?

- **Configuración automática:** Reduce el tiempo de desarrollo al evitar configuraciones manuales extensas.
- **Arquitectura modular:** Permite integrar componentes de Spring fácilmente.
- **Gestión de dependencias con Maven y Gradle:** Facilita la administración de librerías externas.
- **API REST y seguridad integrada:** Ideal para desarrollar servicios web con autenticación y autorización avanzadas.
- **Alta escalabilidad:** Se adapta bien a aplicaciones de cualquier tamaño, desde pequeñas herramientas hasta sistemas empresariales.

Spring Boot es una excelente opción para desarrolladores que buscan rapidez y simplicidad sin sacrificar potencia.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/10		1.0	DT-MTS-01

3.1.4 React

Es una **biblioteca de JavaScript** creada por Meta (antes Facebook) para desarrollar interfaces de usuario dinámicas y eficientes. Se basa en un enfoque **componente por componente**, lo que permite construir aplicaciones interactivas de manera modular y reutilizable.


¿Por qué es tan popular?

- **Arquitectura basada en componentes:** Facilita la organización y mantenimiento del código.
- **Virtual DOM:** Mejora el rendimiento al actualizar solo los elementos necesarios en la interfaz.
- **Desarrollo declarativo:** Permite escribir código más predecible y fácil de depurar.
- **Gran ecosistema:** Compatible con herramientas como Redux, Next.js y React Native para aplicaciones móviles.
- **Amplia comunidad:** Cuenta con documentación detallada y soporte de desarrolladores en todo el mundo.

React es una opción poderosa para la creación de aplicaciones web modernas, desde pequeños proyectos hasta plataformas complejas.

3.1.5 Node.js

es un **entorno de ejecución de JavaScript** basado en el motor V8 de Chrome, diseñado para ejecutar código JavaScript en el servidor. Su enfoque asíncronico y basado en eventos lo hace ideal para aplicaciones escalables y de alto rendimiento.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

¿Por qué es tan útil?

- **JavaScript en el backend:** Permite a los desarrolladores usar el mismo lenguaje tanto en el cliente como en el servidor.
- **Modelo asíncrono y no bloqueante:** Optimiza el rendimiento al manejar múltiples solicitudes simultáneamente.
- **Gran ecosistema de paquetes:** Usa npm (Node Package Manager) para acceder a miles de bibliotecas y módulos.
- **Ideal para APIs y microservicios:** Facilita la creación de servicios web eficientes y modulares.
- **Soporte activo y comunidad amplia:** Cuenta con documentación extensa y una gran comunidad de desarrolladores.


Node.js es perfecto para construir aplicaciones web dinámicas, servidores en tiempo real y sistemas escalables.

3.1.6 Vite

Es una herramienta **de construcción rápida** para proyectos web modernos, diseñada para mejorar el rendimiento y la experiencia de desarrollo. Se enfoca en la rapidez y eficiencia al manejar aplicaciones basadas en JavaScript y frameworks como React, Vue y Svelte.

¿Por qué es tan potente?

- **Servidor de desarrollo ultrarrápido:** Usa ES Modules nativos para minimizar el tiempo de carga.
- **Compilación optimizada:** Genera archivos eficientes para producción con Rollup.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

- **Hot Module Replacement (HMR):** Permite actualizaciones instantáneas en el navegador sin recargar la página.
- **Configuración mínima:** Funciona de inmediato sin necesidad de ajustes complejos.
- **Gran compatibilidad:** Soporta múltiples frameworks y bibliotecas populares.


Vite es una excelente opción para desarrolladores que buscan velocidad y facilidad en la creación de aplicaciones web modernas.

3.1.7 Git

es un **sistema de control de versiones distribuido** diseñado para gestionar el historial de cambios en proyectos de software. Es fundamental para el desarrollo colaborativo, ya que permite a múltiples desarrolladores trabajar en el mismo código sin conflictos.

¿Por qué es tan poderoso?

- **Control de versiones eficiente:** Permite rastrear cada modificación en los archivos del proyecto.
- **Trabajo distribuido:** No depende de un servidor central; cada copia del repositorio es completa e independiente.
- **Ramas y fusiones:** Facilita la experimentación sin afectar el código principal y permite integrar cambios fácilmente.
- **Integración con GitHub, GitLab y Bitbucket:** Mejora la colaboración y almacenamiento en la nube.
- **Seguridad y velocidad:** Está optimizado para manejar grandes cantidades de código de forma rápida y segura.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

Git es esencial para equipos de desarrollo y proyectos individuales que requieren una gestión estructurada del código.


3.1.8 GitHub

es una **plataforma de alojamiento de código** basada en Git, que permite a los desarrolladores colaborar, compartir y gestionar proyectos de software de manera eficiente. Se ha convertido en una herramienta esencial para el desarrollo moderno, facilitando el trabajo en equipo y el control de versiones.

¿Por qué es tan útil?

- **Repositorios públicos y privados:** Permite almacenar y compartir código con equipos o la comunidad.
- **Control de versiones con Git:** Facilita la administración del historial de cambios en los proyectos.
- **Colaboración y revisiones:** Los desarrolladores pueden trabajar en conjunto, proponer cambios y hacer revisiones de código.
- **Automatización con GitHub Actions:** Permite configurar flujos de trabajo para pruebas y despliegues automáticos.
- **Integración con otras herramientas:** Compatible con IDEs, CI/CD y servicios como Docker y AWS.

GitHub es clave para equipos de desarrollo y proyectos personales que necesitan organización, seguridad y colaboración efectiva.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/10		1.0	DT-MTS-01

3.1.9 MySQL

es un **sistema de gestión de bases de datos relacional (RDBMS)** de código abierto. Se usa para almacenar, organizar y administrar datos de forma eficiente, siendo ampliamente utilizado en aplicaciones web, sistemas empresariales y desarrollo de software en general.


3.1.10 MySQL Server

Es el **motor de base de datos** que ejecuta MySQL. Se encarga de procesar consultas, gestionar usuarios y garantizar la integridad de los datos. MySQL Server puede operar en distintos entornos, ya sea en servidores locales o en la nube, optimizando el almacenamiento y recuperación de información.

3.1.11 MySQL Workbench

Es una **herramienta gráfica** para administrar bases de datos MySQL. Permite diseñar, modelar y ejecutar consultas de manera visual, facilitando la gestión de esquemas, migraciones y el rendimiento del sistema. Ideal para desarrolladores y administradores de bases de datos.

Estos tres elementos trabajan en conjunto para ofrecer una solución completa de administración de datos.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01


3.2 PRERREQUISITOS DE INSTALACIÓN

Tabla 1: Prerrequisitos de instalación

Componente	Requisito
Sistema Operativo	Windows 10 o Posterior.
Memoria RAM	Mínimo 8 GB de memoria RAM, Óptimo 16 GB o más de memoria RAM para un rendimiento fluido.
Procesador	Intel Core i5 / AMD Ryzen 5 o superior.
Almacenamiento	50 GB de espacio disponible en disco.
Red	Conexión a Internet estable para sincronización en tiempo real.
Navegador Web	Google Chrome, Brave o Firefox actualizados a su última versión disponible.

Tabla 2: Dependencias utilizadas

Dependencias	
Backend	Java JDK 17, springboot starter data jpa, spring boot starter web, - spring-boot-devtools, mysql-connector-j, spring-boot-starter-validation, - jjwt-api, - jjwt-impl, jjwt-jackson, spring-boot-starter-security
Frontend	node.js 20.14.0, react 18.3.1, react-dom 18.3.1, mui/icons-material 6.1.8, react-router-dom 6.28.0, xlsx 0.18.5, yup 1.4.0, axios 1.9.0, jspdf 3.0.1, jspdf-autotable 5.0.2, jwt-decode 4.0.0, vite 6.2.3

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

3.3 FRAMEWORKS Y ESTÁNDARES UTILIZADOS

Para el desarrollo del sistema **Hardware Store Inventory FFIG**, se han utilizado tecnologías clave y estándares reconocidos que garantizan la calidad, seguridad y escalabilidad del proyecto.


3.3.1 Lenguajes de programación:

- **Frontend:** JavaScript
- **Backend:** Java JDK 17

3.3.2 Frameworks y tecnologías:

- **Frontend:** React.js
- **Backend:** Spring Boot
- **Base de datos:** MySQL
- **ORM:** JPA / Hibernate
- **Estándares y patrones de diseño:**
Patrón de arquitectura: Modelo-Vista-Controlador (MVC)
- **Gestión de datos:** DTO (Data Transfer Object)
- **Seguridad:** Spring Security con autenticación basada en **JWT (JSON Web Token)**
- **Buenas prácticas de desarrollo:** SOLID, Clean Code

Estas herramientas y estándares permiten que el sistema sea robusto, escalable y seguro, facilitando su integración y mantenimiento.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01


4 DIAGRAMAS DE CASO DE USO

Son esenciales para entender los requisitos del sistema desde la perspectiva del usuario. Se emplean en la fase de análisis para describir cómo los actores externos interactúan con el software, facilitando la comunicación entre el equipo técnico y los clientes.

4.1 Caso de Uso No. 1

Tabla 3: Caso de uso Administrar Usuarios, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 1	
NOMBRE	Administrar Usuarios	
ACTOR	Administrador IT, Sistema, Usuario Funcional (Gerente, Administrador, Almacenista, Vendedor)	
OBJETIVOS ASOCIADOS	Gestionar el acceso de acuerdo con los roles de cada usuario	
REQUISITOS ASOCIADOS	Iniciar sesión como administrador IT	
DESCRIPCION	El usuario administrador puede gestionar las cuentas de usuario	
PRECONDICION	Tener el inicio de sesión activo	
VERSIÓN	1,1	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo
	2	El usuario ingresa nombre de usuario y contraseña
	3	El usuario inicia sesión
	4	El sistema redirige al usuario al módulo correspondiente según su rol: módulo de usuarios o módulo de inventario.
POSTCONDICION	Se valida el rol de acceso en el sistema	
	El sistema cerrará la sesión si el token de autenticación ha expirado o si el usuario permanece inactivo por más de 5 minutos.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	4	Se muestra una alerta indicando que no tiene los suficientes privilegios de ingreso al módulo de usuarios si los roles son: (Dueño, administrador, almacenista y vendedor).
NOTA	El término "Usuario Funcional" se utiliza para englobar los roles que interactúan con el sistema según sus permisos: Gerente, Administrador, Almacenista y Vendedor.	
RENDIMIENTO	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moqups.com/86ZfGX0riZ2VngoVgkfWzoMhCox99lbB/view/page/a21b7b685	

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/10		1.0	DT-MTS-01

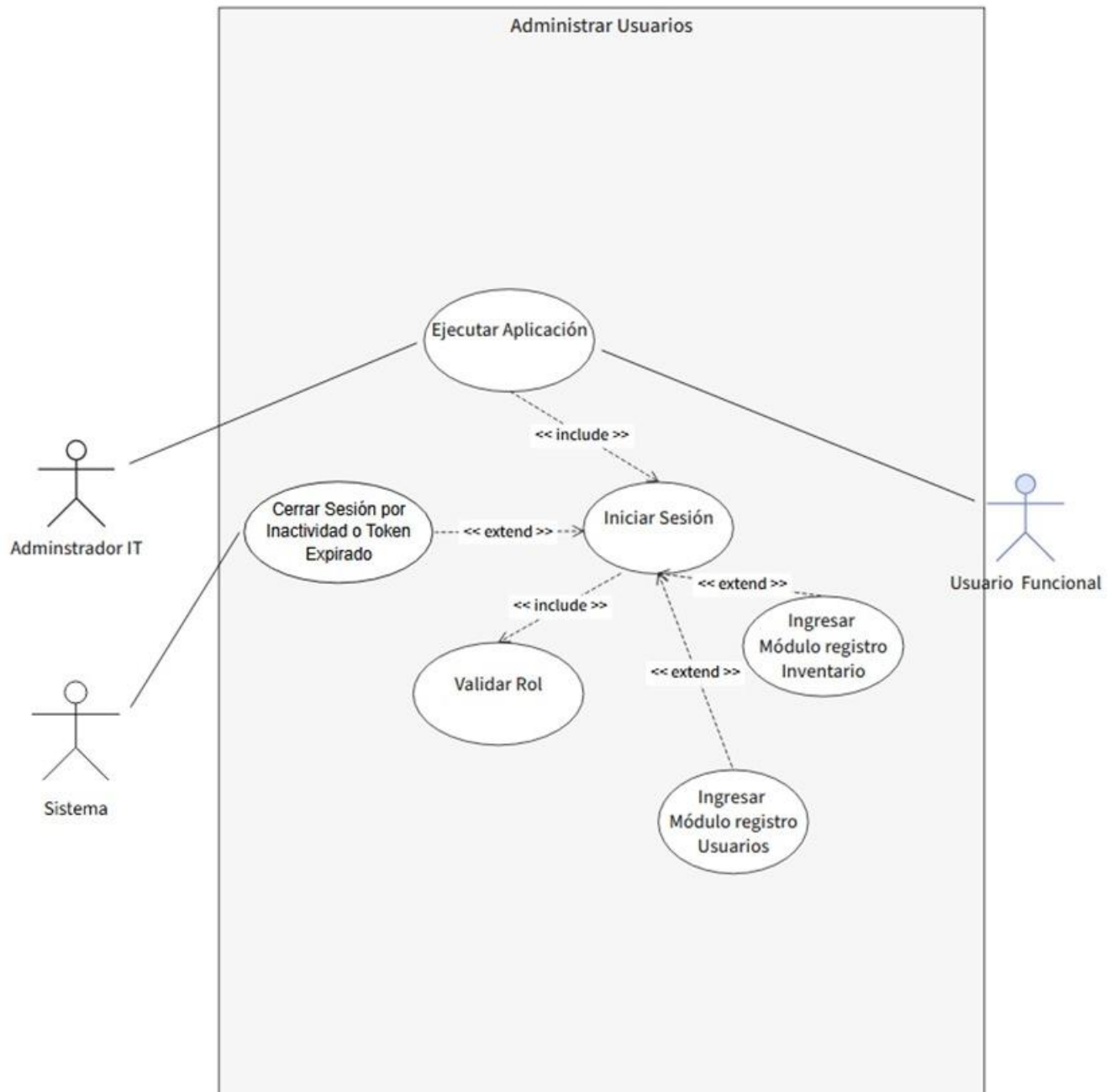




Imagen 1: Diagrama UML caso de uso Administrar Usuarios, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.2 Caso de Uso No. 2

Tabla 4: Caso de uso Crear Usuarios, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 2	
NOMBRE	Crear Usuarios	
ACTOR	Administrador IT, Gerente	
OBJETIVOS ASOCIADOS	Crear Usuarios Funcionales	
REQUISITOS ASOCIADOS	Iniciar sesión como administrador IT	
DESCRIPCION	El usuario administrador puede crear usuarios nuevos dentro de la aplicación	
PRECONDICION	Tener el inicio de sesión activo	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo.
	2	El usuario ingresa nombre de usuario y contraseña.
	3	El usuario inicia sesión.
	4	El usuario seleccionara el módulo registro de usuarios.
	5	El usuario selecciona la pestaña de registro de usuarios.
	6	El usuario ingresa los datos requeridos según sea el caso.
	7	El usuario guarda los datos.
	8	El sistema genera una ventana de confirmación.
	9	El sistema guarda y actualiza la base de datos.
POSTCONDICION	Se valida el rol de acceso en el sistema, Se guardan los datos creados en el sistema.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	6	Se muestra un error si los campos obligatorios no han sido diligenciados
NOTA	El Gerente tiene permisos para crear y consultar usuarios, pero no puede actualizarlos ni eliminarlos.	
RENDIMIENTO	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.mogups.com/86ZfGX0riZ2VngoVgkfWzoMhCox99IbB/view/page/a9139d0ca	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

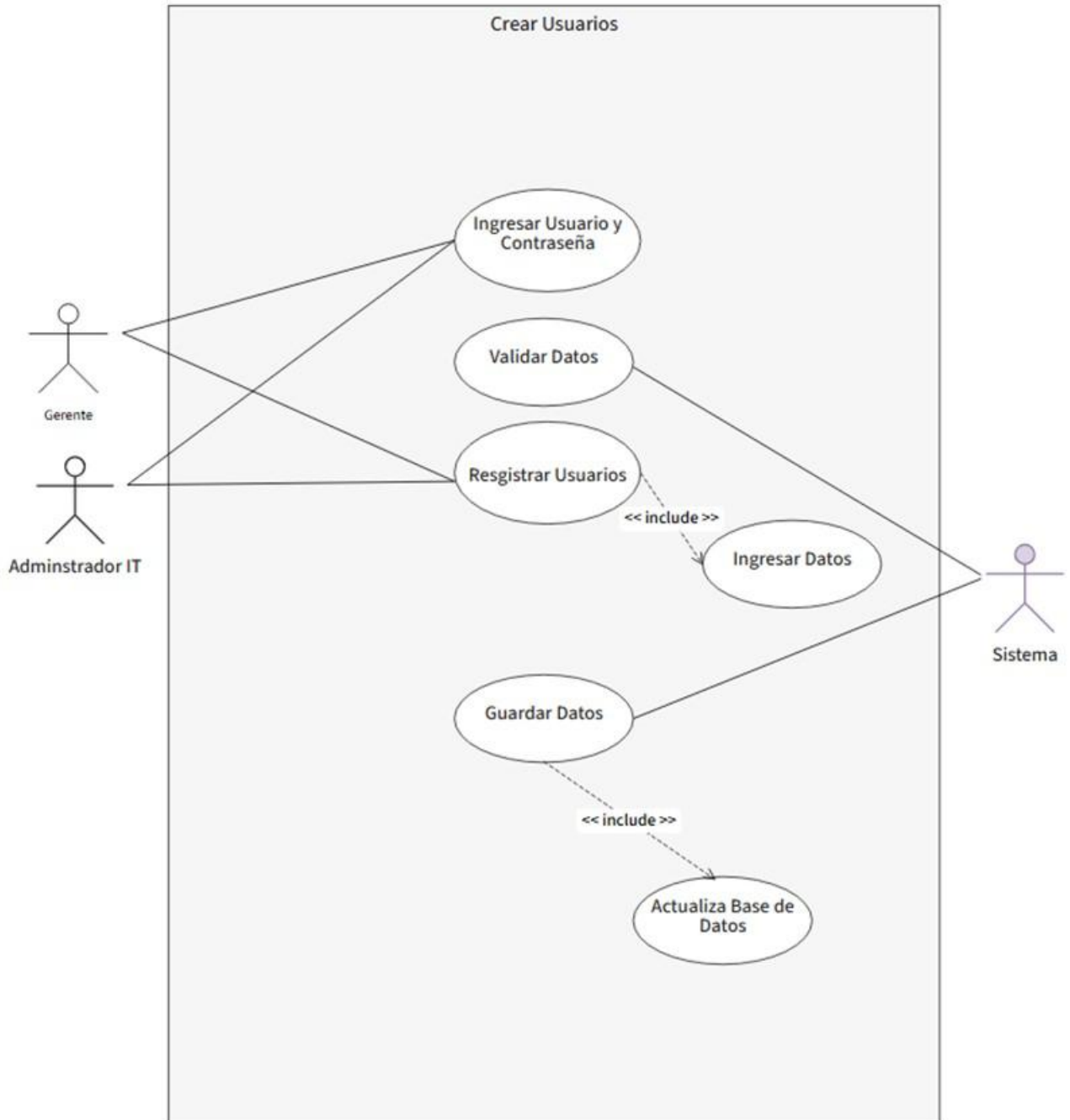




Imagen 2: Diagrama UML caso de uso Crear Usuarios, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.3 Caso de Uso No. 3

Tabla 5: Caso de uso Consultar Usuarios, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 3	
NOMBRE	Consultar Usuarios	
ACTOR	Administrador IT, Usuario Funcional (Gerente, Administrador)	
OBJETIVOS ASOCIADOS	Conocer los usuarios registrados en el sistema y las acciones realizadas	
REQUISITOS ASOCIADOS	Iniciar sesión como administrador IT, Gerente, Administrador	
DESCRIPCION	El Administrador IT, Gerente y Administrador pueden consultar los datos de los usuarios registrados, incluyendo roles y acciones realizadas.	
PRECONDICION	Tener el inicio de sesión activo y que exista usuarios en el sistema.	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo.
	2	El usuario ingresa nombre de usuario y contraseña.
	3	El usuario inicia sesión.
	4	El usuario seleccionara el módulo registro de usuarios.
	5	El usuario selecciona la pestaña de consulta de usuarios.
	6	El usuario ingresa los datos requeridos para la consulta.
	7	El sistema actualiza dinámicamente los resultados de la búsqueda conforme el usuario ingresa datos en el campo de consulta.
	8	El sistema muestra automáticamente los resultados que coinciden con los datos ingresados.
POSTCONDICION	El sistema muestra la información del usuario consultado si existe, o informa si no se encuentra.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	7	Se muestra un mensaje indicando que el usuario no existe en el sistema.
RENDIMIENTO	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moqups.com/86ZfGX0riZ2VngoVgkfWzoMhCox99IbB/view/page/a9058a81f	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

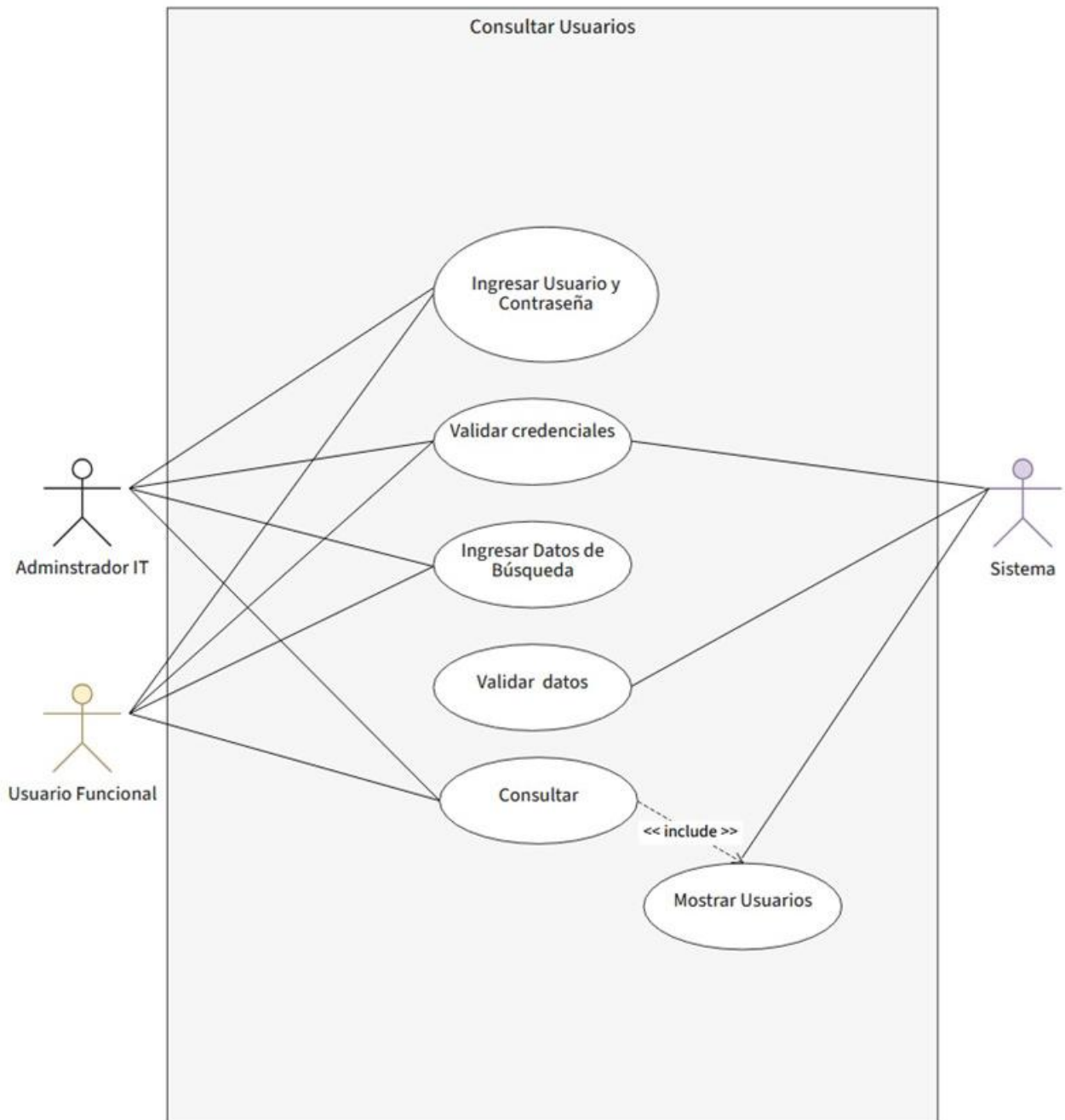




Imagen 3: Diagrama UML caso de uso Consultar Usuarios, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.4 Caso de Uso No. 4

Tabla 6: Caso de uso Actualizar Usuarios, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 4	
NOMBRE	Actualizar Usuarios	
ACTOR	Administrador IT	
OBJETIVOS ASOCIADOS	Modificar los datos de un usuario existente	
REQUISITOS ASOCIADOS	Iniciar sesión como administrador IT	
DESCRIPCION	El usuario administrador IT podrá modificar y gestionar la información asociada a un usuario, incluyendo el rol asignado.	
PRECONDICION	Tener el inicio de sesión activo y que exista una cuenta creada.	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo.
	2	El usuario ingresa nombre de usuario y contraseña.
	3	El usuario inicia sesión.
	4	El usuario selecciona el módulo registro de usuarios.
	5	El usuario selecciona la pestaña de Actualizar usuarios.
	6	El usuario ingresa los datos requeridos para la consulta.
	7	El usuario realiza la consulta mediante el botón consultar.
	8	El sistema devuelve reporte de la consulta realizada.
	9	El Usuario modifica los campos necesarios.
	10	El usuario guarda los cambios.
	11	El sistema actualiza la base de datos.
POSTCONDICION	Se actualiza la información del usuario en la base de datos y se confirma el cambio.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	6	Se muestra un error si los campos obligatorios no han sido diligenciados
RENDIMIENTO	7	Se muestra mensaje de información de usuario inexistente.
	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moqups.com/86ZfGX0riZ2VnqoVqkfWzoMhCox99lbB/view/page/af7a97e22	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

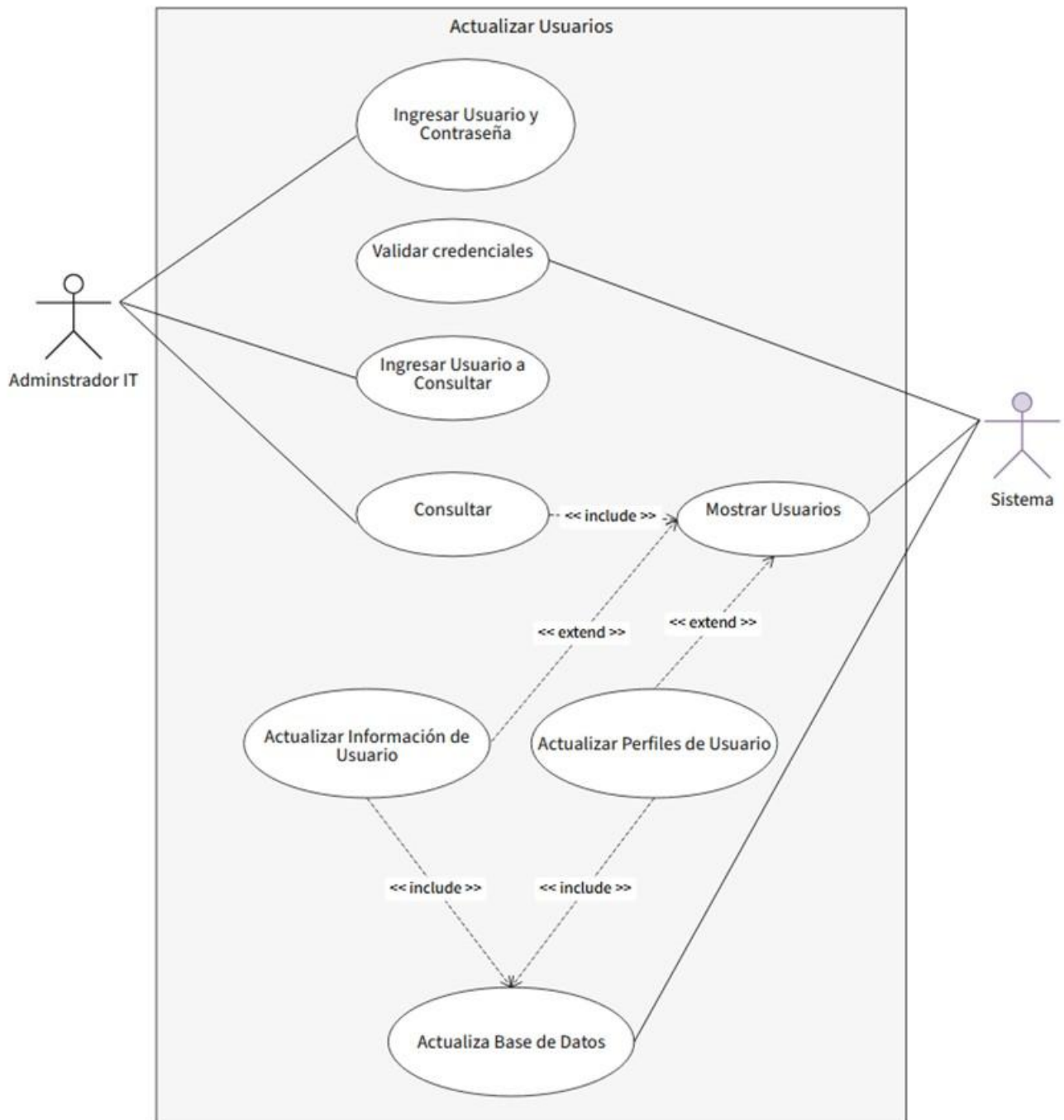




Imagen 4: Diagrama UML caso de uso Actualizar Usuarios, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.5 Caso de Uso No. 5

Tabla 7: Caso de uso Eliminar Usuarios, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 5	
NOMBRE	Eliminar Usuarios	
ACTOR	Administrador IT	
OBJETIVOS ASOCIADOS	Eliminar un usuario del sistema	
REQUISITOS ASOCIADOS	Iniciar sesión como administrador IT	
DESCRIPCION	El usuario administrador IT podrá eliminar un usuario existente según la necesidad	
PRECONDICION	Tener el inicio de sesión activo y que existan usuarios en el sistema.	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo.
	2	El usuario ingresa nombre de usuario y contraseña.
	3	El usuario inicia sesión.
	4	El usuario seleccionara el módulo registro de usuarios.
	5	El usuario selecciona la pestaña de Eliminar usuarios.
	6	El usuario ingresa los datos requeridos para la consulta.
	7	El sistema filtra dinámicamente los resultados a medida que el usuario escribe en el campo de búsqueda.
	8	El sistema muestra los resultados del usuario consultado.
	9	El Usuario Elimina el registro asociado a la consulta hecha.
	10	El usuario guarda los cambios hechos.
	11	El sistema actualiza la base de datos.
POSTCONDICION	Se guardan los cambios en el sistema y se confirma la eliminación del usuario.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	6	Se muestra un error si los campos obligatorios no han sido diligenciados
RENDIMIENTO	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moqups.com/86ZfGX0riZ2VngoVgkfWzoMhCox99lbB/view/page/af73c3ed1	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

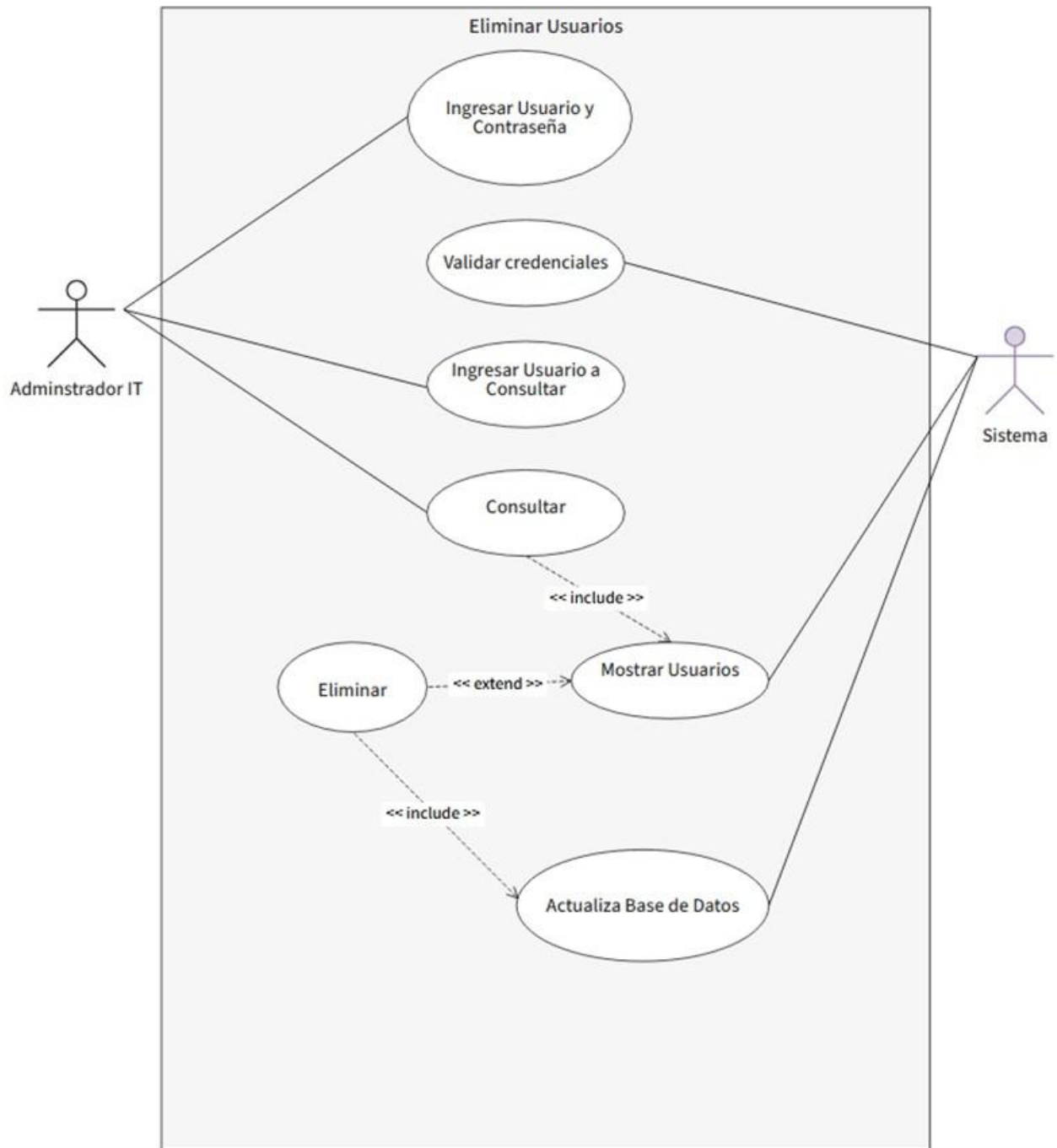




Imagen 5: Diagrama UML caso de uso Eliminar Usuarios, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.6 Caso de Uso No. 6

Tabla 8: Caso de uso Administrar Inventario, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 6	
NOMBRE	Administrar Inventario	
ACTOR	Almacenista	
OBJETIVOS ASOCIADOS	Gestionar el inventario	
REQUISITOS ASOCIADOS	Iniciar sesión como almacenista	
DESCRIPCION	El usuario almacenista puede gestionar el inventario	
PRECONDICION	Tener el inicio de sesion activo	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo
	2	El usuario ingresa nombre de usuario y contraseña
	3	El usuario inicia sesión
	4	El usuario seleccionara el módulo de registro de inventario en el aplicativo.
	5	El sistema valida las credenciales ingresadas.
	6	El sistema valida el rol del usuario para acceder al módulo de inventario.
POSTCONDICION	Se valida el rol del usuario y se habilita el acceso al módulo de inventario.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	6	Se muestra mensaje indicando que el usuario no tiene permisos para acceder al módulo de inventario.
RENDIMIENTO	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moqups.com/86ZfGX0riZ2VngoVgkfWzoMhCox99lbB/view/page/a0a99ab00	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

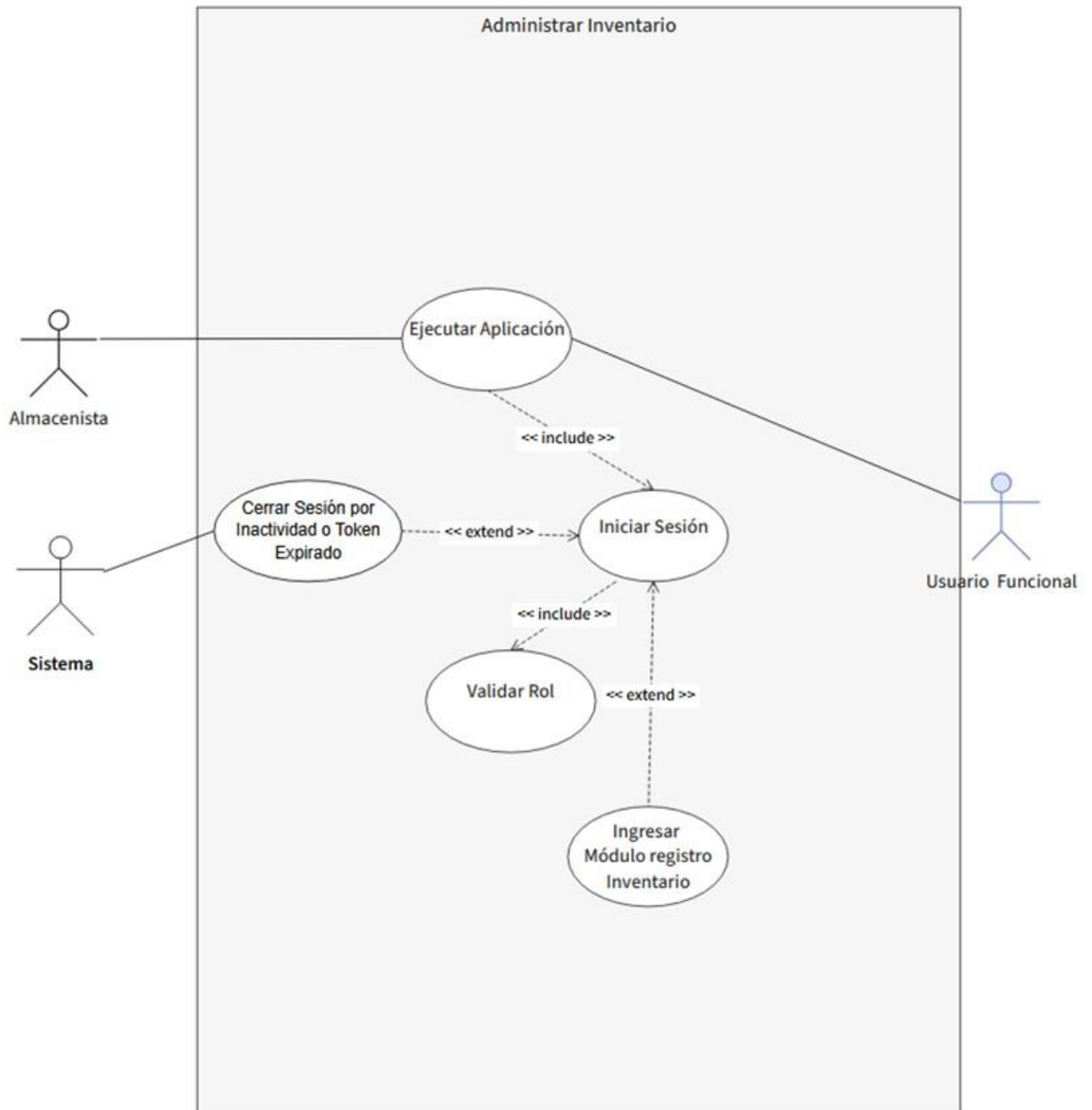




Imagen 6: Diagrama UML caso de uso Administrar Inventario, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.7 Caso de Uso No. 7

Tabla 9: Caso de uso Registrar Inventario, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 7	
NOMBRE	Registrar Inventario	
ACTOR	Almacenista	
OBJETIVOS ASOCIADOS	Registrar el nuevo inventario	
REQUISITOS ASOCIADOS	Iniciar sesión como almacenista	
DESCRIPCION	El usuario almacenista puede realizar el ingreso del nuevo inventario al sistema	
PRECONDICION	Tener el inicio de sesión activo	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo
	2	El usuario ingresa nombre de usuario y contraseña
	3	El usuario inicia sesión
	4	El usuario seleccionara el módulo registro de inventario en el aplicativo.
	5	El usuario ingresa a la pestaña de registrar inventario
	6	El usuario ingresa los datos requeridos según sea el caso.
	7	El usuario hace clic en el botón "Guardar".
	8	El sistema valida los datos e informa si el registro fue exitoso.
POSTCONDICION	9	El sistema guarda y actualiza la base de datos.
	Se valida el rol de acceso en el sistema y se almacena el nuevo inventario en la base de datos.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	6	Se muestra un error si los campos obligatorios no han sido diligenciados.
RENDIMIENTO	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moqups.com/86ZfGX0riZ2VngoVgkfWzoMhCox99lbB/view/page/ab7db5af1	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

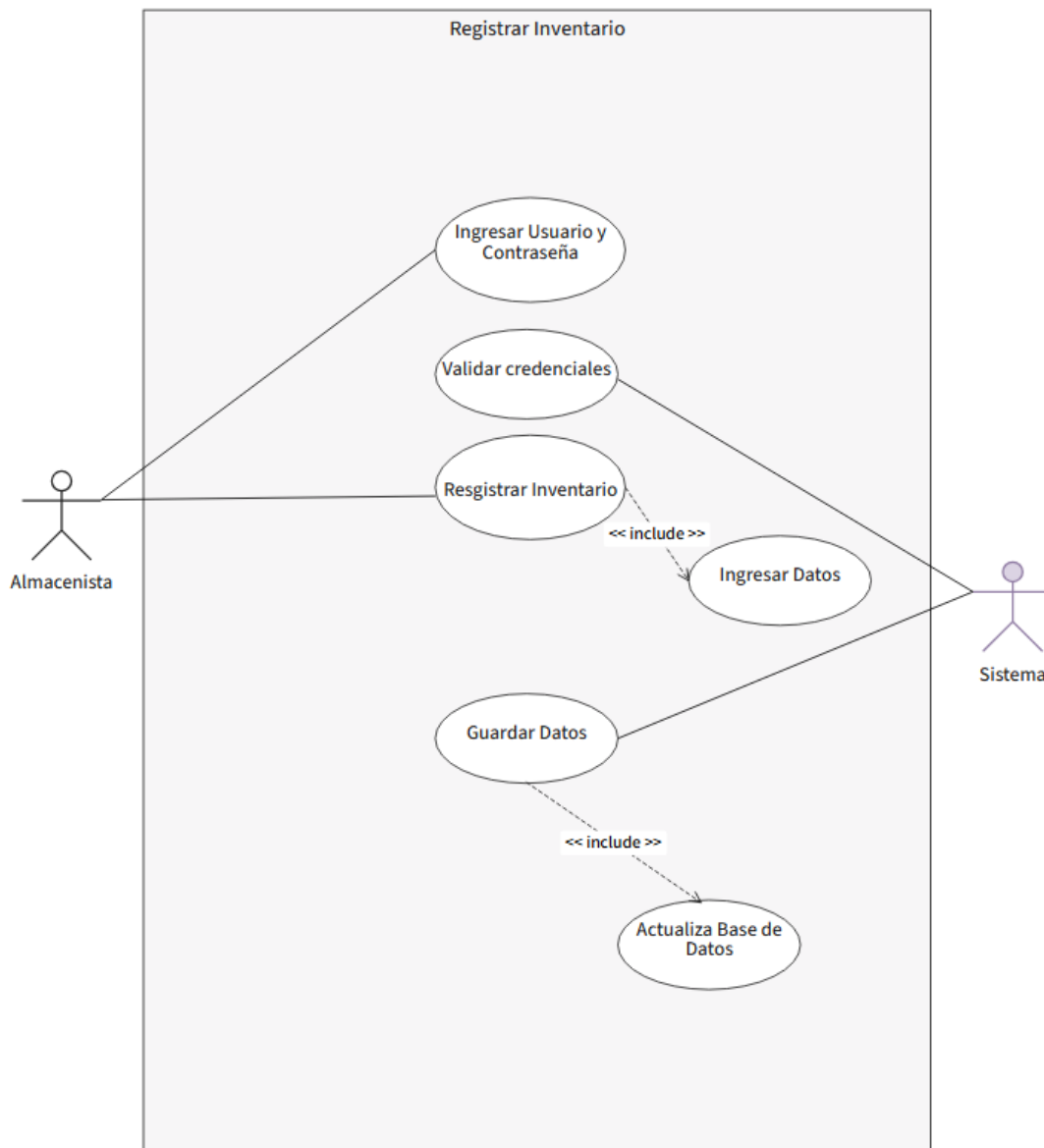




Imagen 7: Diagrama UML caso de uso Registrar Inventario, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.8 Caso de Uso No. 8

Tabla 10: Caso de uso Consultar Inventario, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 8	
NOMBRE	Consultar Inventario	
ACTOR	Almacenista, Usuario Funcional (Administrador IT, Gerente, Administrador, Vendedor)	
OBJETIVOS ASOCIADOS	Conocer el inventario registrado en el sistema	
REQUISITOS ASOCIADOS	Iniciar sesión como Administrador IT, Gerente, Administrador, Vendedor	
DESCRIPCION	El usuario almacenista, Administrador IT, Gerente, Administrador, Vendedor podrán realizar la consulta del inventario registrado en el sistema.	
PRECONDICION	Tener el inicio de sesión activo y que exista mercancía en el sistema.	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo.
	2	El usuario ingresa nombre de usuario y contraseña.
	3	El usuario inicia sesión.
	4	El usuario seleccionara el módulo registro de inventario.
	5	El usuario selecciona la pestaña de consultar inventario.
	6	El usuario ingresa los datos requeridos para la consulta.
	7	El sistema filtra dinámicamente los resultados a medida que el usuario escribe en el campo de búsqueda o interactúa con los filtros de la búsqueda avanzada.
	8	El sistema muestra automáticamente los productos que coinciden con los datos ingresados o seleccionados en los filtros.
POSTCONDICION	No hay postcondición.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	7	Se muestra mensaje de información de producto inexistente.
RENDIMIENTO	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moguups.com/86ZfGX0riZ2VngoVgkfWzoMhCox99lbB/view/page/a26e3893d	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

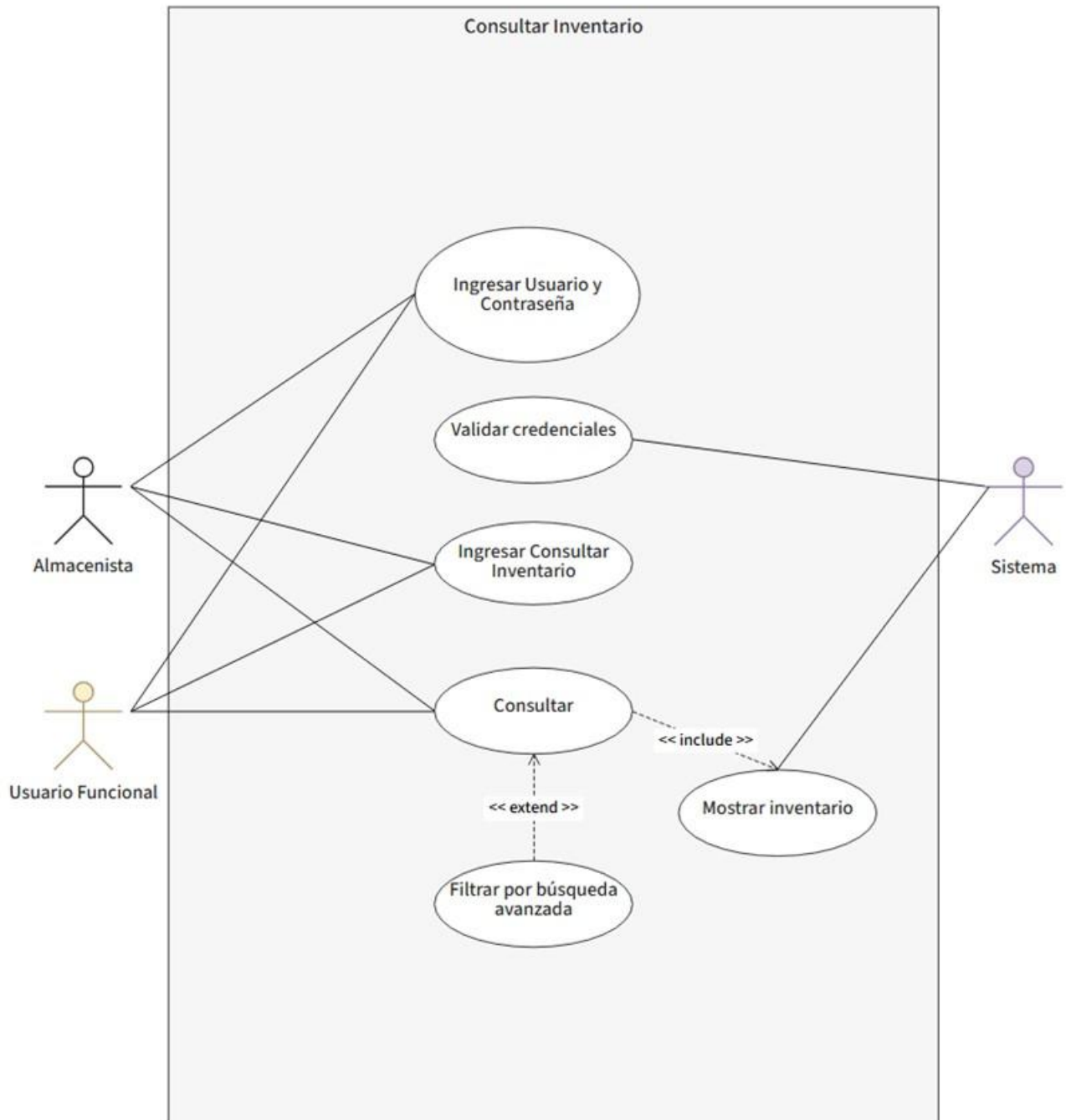




Imagen 8:: Diagrama UML caso de uso Consultar Inventario, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.9 Caso de Uso No. 9

Tabla 11: Caso de uso Actualizar Inventario, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 9	
NOMBRE	Actualizar Inventario	
ACTOR	Administrador	
OBJETIVOS ASOCIADOS	Modificar los datos de la mercancía existente	
REQUISITOS ASOCIADOS	Iniciar sesión como administrador	
DESCRIPCION	El usuario administrador podrá modificar y gestionar la información asociada al producto.	
PRECONDICION	Tener el inicio de sesion activo y que exista el producto credo.	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo.
	2	El usuario ingresa nombre de usuario y contraseña.
	3	El usuario inicia sesión.
	4	El usuario seleccionara el módulo registro de inventario.
	5	El usuario selecciona la pestaña de Actualizar inventario.
	6	El usuario ingresa los datos requeridos para la consulta.
	7	El usuario realiza la consulta mediante el boton consultar.
	8	El sistema devuelve reporte de la consulta realizada.
	9	El Usuario modifica los campos necesarios.
	10	El usuario gurma los cambio hechos.
	11	El sistema actualiza la base de datos y muestra mensaje de confirmación.
POSTCONDICION	Se guardan los datos actualizados en el sistema.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	6	Se muestra un error si los campos obligatorios no han sido diligenciados
RENDIMIENTO	7	Se muestra mensaje de información de producto inexistente.
	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moqups.com/86ZfGX0riZ2VngoVqkfWzoMhCox99lbb/view/page/a5addfeb0	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

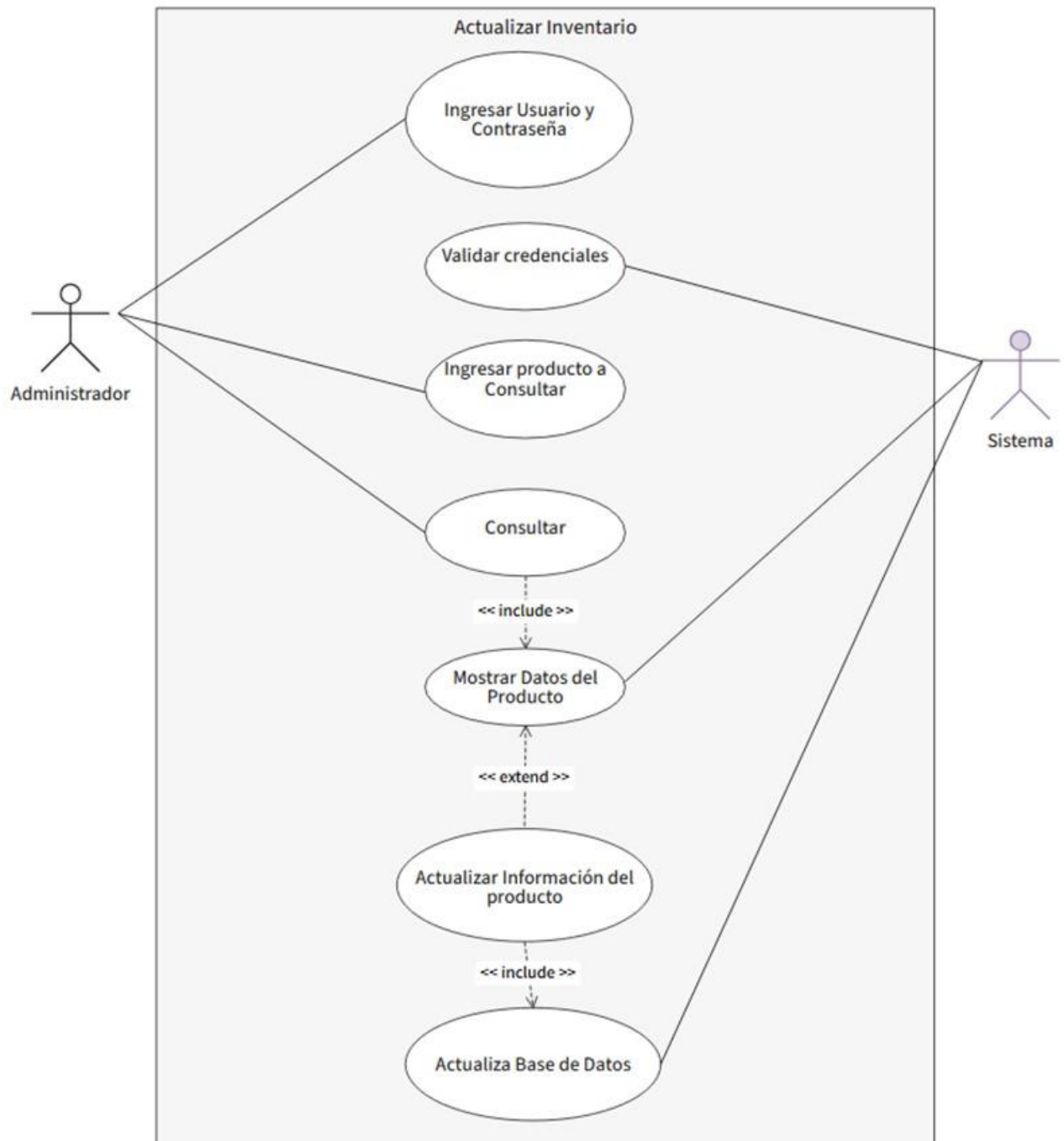




Imagen 9: Diagrama UML caso de uso Actualizar Inventario, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

4.10 Caso de Uso No. 10

Tabla 12: Caso de uso Eliminar Inventario, Fuente: JuDaJo System Soft (autores)

CASO DE USO		
CODIGO	No. 10	
NOMBRE	Eliminar Inventario	
ACTOR	Administrador IT y Gerente	
OBJETIVOS ASOCIADOS	Eliminar inventario y registros de la Base de Datos	
REQUISITOS ASOCIADOS	Iniciar sesión como administrador IT o Gerente	
DESCRIPCION	El usuario administrador IT o gerente podrán eliminar un producto existente según la necesidad.	
PRECONDICION	Tener el inicio de sesión activo y que exista el producto en el sistema.	
SECUENCIA-FLUJO NORMAL	PASO	ACCION
	1	El usuario ejecuta el aplicativo.
	2	El usuario ingresa nombre de usuario y contraseña.
	3	El usuario inicia sesión.
	4	El usuario seleccionara el módulo registro de inventario.
	5	El usuario selecciona la pestaña de Eliminar inventario.
	6	El usuario ingresa los datos requeridos para la consulta.
	7	El sistema filtra dinámicamente los resultados a medida que el usuario escribe en el campo de búsqueda.
	8	El sistema muestra automáticamente los productos que coinciden con los datos ingresados.
	9	El Usuario Elimina el registro asociado a la consulta hecha.
	10	El usuario guarda los cambios hechos.
	11	El sistema actualiza la base de datos.
POSTCONDICION	Se eliminan los datos de la mercancía seleccionada y se confirma la acción.	
EXCEPCIONES	PASO	ACCION
	2	Se muestra mensaje de error indicando que el nombre de usuario y / o contraseña es incorrecto.
	7	Se muestra mensaje de información de mercancía inexistente.
RENDIMIENTO	PASO	COTA DE TIEMPO
REFERENCIA DEL DIAGRAMA	https://app.moqups.com/86ZfGX0riZ2VngoVqkfWzoMhCox99lbB/view/page/a368852d0	

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

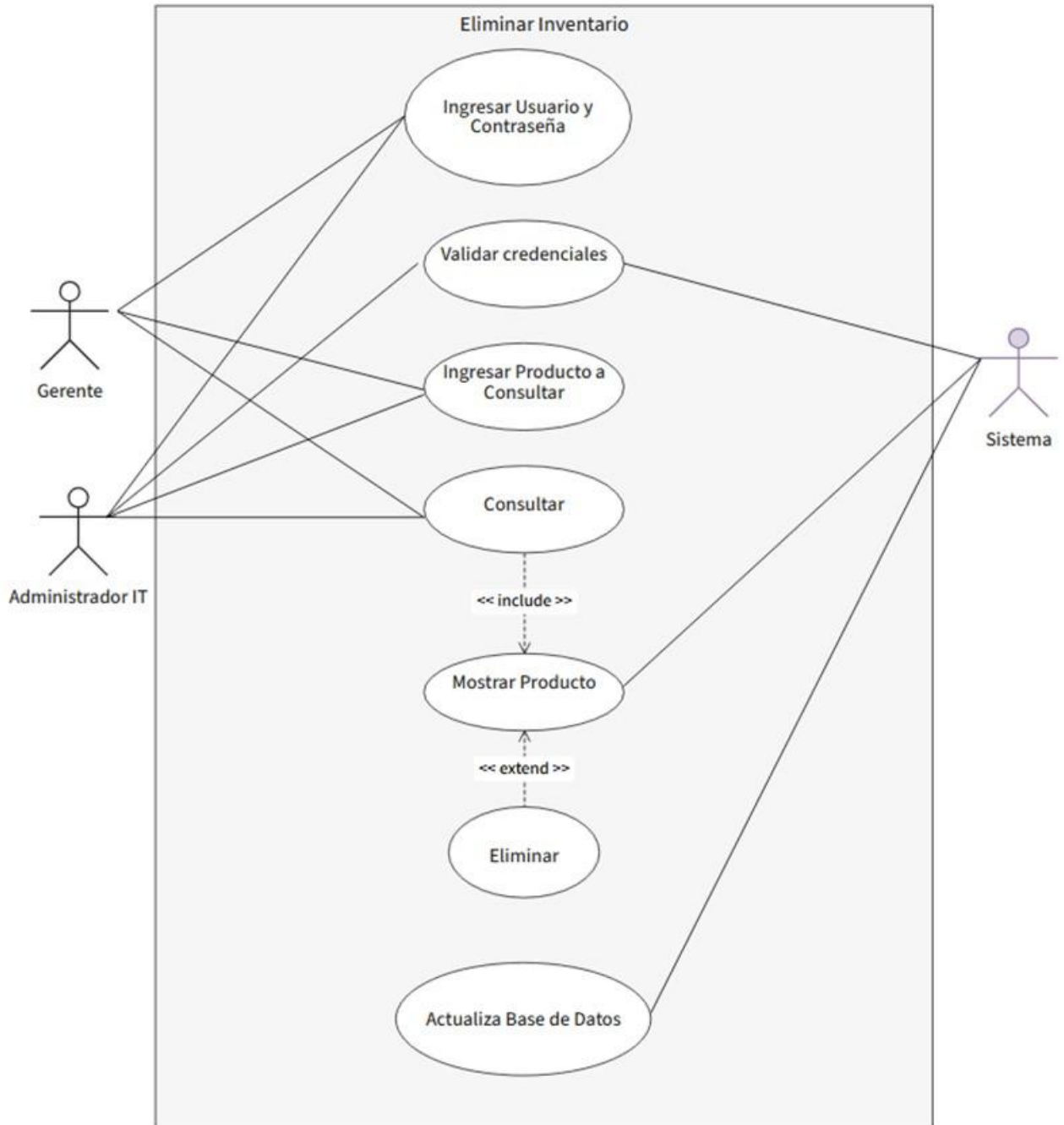



Imagen 10: Diagrama UML caso de uso Eliminar Inventario, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/10		1.0	DT-MTS-01

5 MODELO RELACIONAL DE LA BASE DE DATOS

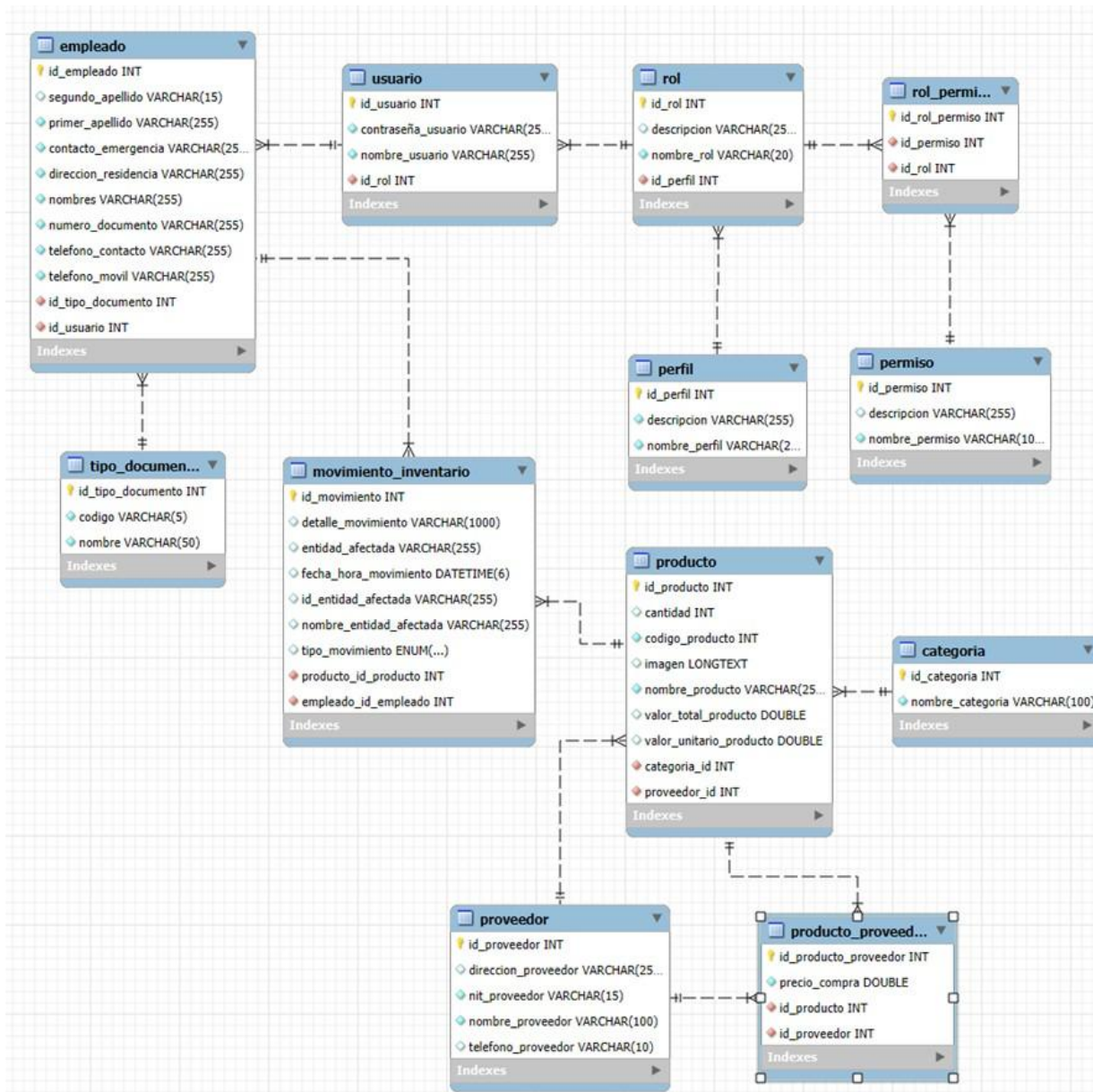



Imagen 11: Diagrama MER proyecto Hardware Store Inventory, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/10	1.0	DT-MTS-01

6 DICCIONARIO DE DATOS


El presente diccionario de datos fue creado por el plugin de workbench **datadict_grt.py** como archivo **.HTML**, se guardó como **PDF** y luego fue convertido a **Word** para ajustar las tablas y agregar los comentarios (descripción).

hardware_store_inventory_api Data Dictionary

2025-06-09

Alphabetic Index

- [categoria](#)
- [empleado](#)
- [movimiento inventario](#)
- [perfil](#)
- [permiso](#)
- [producto](#)
- [producto proveedor](#)
- [proveedor](#)
- [rol](#)
- [rol permiso](#)
- [tipo documento](#)
- [usuario](#)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/08	1.0	DT-MTS-01

categoria


Tabla 13: Diccionario de datos entidad categoría, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_categoria	INT	✓	✓					✓		Identificador único de la categoría (clave primaria).
nombre_categoria	VARCHAR(100)		✓							Nombre descriptivo de la categoría del producto

empleado

Tabla 14: Diccionario de datos entidad empleado, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_empleado	INT	✓	✓					✓		Identificador único del empleado (clave primaria).
segundo_apellido	VARCHAR(15)								NULL	Segundo apellido del empleado (opcional).
primer_apellido	VARCHAR(255)		✓							Primer apellido del empleado.
contacto_emergencia	VARCHAR(255)		✓							Nombre del contacto en caso de emergencia.
direccion_residencia	VARCHAR(255)		✓							Dirección de residencia actual del empleado.
nombres	VARCHAR(255)		✓							Nombres completos del empleado.
numero_documento	VARCHAR(255)		✓							Número del documento de identidad del empleado.
telefono_contacto	VARCHAR(255)		✓							Teléfono de contacto alternativo (fijo o adicional).
telefono_movil	VARCHAR(255)		✓							Número de teléfono móvil personal
id_tipo_documento	INT		✓							Identificador del tipo de documento asociado (clave foránea a tipo_documento).
id_usuario	INT		✓							Identificador del usuario en el sistema (clave foránea a usuario).

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

movimiento_inventario

Tabla 15: Diccionario de datos entidad movimiento_inventario, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_movimiento	INT	✓	✓					✓		Identificador único del movimiento de inventario (clave primaria).
detalle_movimiento	VARCHAR(1000)								NULL	Descripción del movimiento realizado (por ejemplo: "Producto actualizado").
entidad_afectada	VARCHAR(255)								NULL	Tipo de entidad afectada por el movimiento (ejemplo: producto,
fecha_hora_movimiento	DATETIME(6)								NULL	Fecha y hora exacta en la que se registró el movimiento.
id_entidad_afectada	VARCHAR(255)								NULL	Identificador específico de la entidad afectada (por ejemplo: ID del producto).
nombre_entidad_afectada	VARCHAR(255)								NULL	Nombre legible de la entidad afectada.
tipo_movimiento	ENUM('ACTUALIZAR', 'CONSULTAR', 'CREAR', 'ELIMINAR')								NULL	Tipo de operación realizada: ACTUALIZAR, CONSULTAR, CREAR o ELIMINAR.
id_empleado_responsable	INT		✓							ID del empleado que ejecutó la acción (clave foránea a empleado).

perfil


Tabla 16: Diccionario de datos entidad perfil, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_perfil	INT	✓	✓					✓		Identificador único del perfil de usuario (clave primaria).
descripcion	VARCHAR(255)		✓							Descripción detallada del perfil y su propósito dentro del sistema.
nombre_perfil	VARCHAR(20)		✓							Nombre corto o etiqueta asignada al perfil (por ejemplo: "Administrador", "Almacenista").

permiso

Tabla 17: Diccionario de datos entidad permiso, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_permiso	INT	✓	✓					✓		Identificador único del permiso (clave primaria).
descripcion	VARCHAR(255)								NULL	Descripción detallada de lo que permite esta acción dentro del sistema.
nombre_permiso	VARCHAR(100)		✓							Nombre identificador del permiso (ej. "crear_usuario", "ver_inventario").

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

producto

Tabla 18: Diccionario de datos entidad producto, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_producto	INT	✓	✓					✓		Identificador único del producto (clave primaria).
cantidad	INT								NULL	Número de unidades disponibles del producto en inventario.
codigo_producto	INT		✓							Código único que identifica el producto (no es clave primaria, pero es requerido).
Imagen	LONGTEXT								NULL	Imagen del producto codificada en base64.
nombre_producto	VARCHAR(255)		✓							Nombre descriptivo del producto.
valor_total_producto	DOUBLE								NULL	Valor total calculado del producto (cantidad × valor unitario).
valor_unitario_producto	DOUBLE								NULL	Precio por unidad del producto.
categoria_id	INT		✓							Clave foránea que indica la categoría a la que pertenece el producto.
proveedor_id	INT		✓							Clave foránea que indica el proveedor asociado al producto.

producto_proveedor


Tabla 19: Diccionario de datos entidad producto_proveedor, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_producto_proveedor	INT	✓	✓					✓		Identificador único de la relación entre producto y proveedor.
precio_compra	DOUBLE		✓							Precio de compra del producto ofrecido por el proveedor.
id_producto	INT		✓							Clave foránea que referencia el producto asociado.
id_proveedor	INT		✓							Clave foránea que referencia al proveedor que suministra el producto.

proveedor

Tabla 20: Diccionario de datos entidad proveedor, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_proveedor	INT	✓	✓					✓		Identificador único del proveedor.
direccion_proveedor	VARCHAR(255)								NULL	Dirección física del proveedor.
nit_proveedor	VARCHAR(15)		✓							Número de Identificación Tributaria del proveedor (incluye dígito de verificación).
nombre_proveedor	VARCHAR(100)		✓							Nombre o razón social del proveedor.
telefono_proveedor	VARCHAR(10)								NULL	Número de contacto del proveedor.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

rol

Tabla 21: Diccionario de datos entidad rol, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_rol	INT	✓	✓					✓		Identificador único del rol dentro del sistema.
descripcion	VARCHAR(255)								NULL	Breve explicación de las funciones o permisos asociados al rol.
nombre_rol	VARCHAR(20)		✓							Nombre representativo del rol (Ej: Administrador, Vendedor, etc.).
id_perfil	INT		✓							Llave foránea que relaciona el rol con un perfil específico.

rol_permiso

Tabla 22: Diccionario de datos entidad rol_permiso, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_rol_permiso	INT	✓	✓					✓		Identificador único de la relación entre un rol y un permiso.
id_permiso	INT		✓							Llave foránea que hace referencia al permiso asignado.
id_rol	INT		✓							Llave foránea que indica el rol al cual se le asigna el permiso correspondiente.

tipo_documento


Tabla 23: Diccionario de datos entidad tipo_documento, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_tipo_documento	INT	✓	✓					✓		Identificador único del tipo de documento.
Código	VARCHAR(5)		✓							Código corto representativo del tipo de documento (ej. CC, TI, CE).
Nombre	VARCHAR(50)		✓							Nombre completo del tipo de documento.

Usuario

Tabla 24: Diccionario de datos entidad usuario, Fuente: JuDaJo System Soft (autores)

Column name	DataType	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
id_usuario	INT	✓	✓					✓		Identificador único del usuario del sistema.
contraseña_usuario	VARCHAR(255)		✓							Contraseña encriptada utilizada para autenticación.
nombre_usuario	VARCHAR(255)		✓							Nombre de usuario utilizado para iniciar sesión.
id_rol	INT		✓							Llave foránea que define el rol asignado al usuario.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

7 SCRIPTS DE INSTALACIÓN

7.1 Backend – Spring Boot con JPA

Requisitos previos

- Java JDK 17 o superior



Imagen 12: Script para conocer la versión de Java y salida por consola.

- Spring Boot (generado con Spring Initializr)

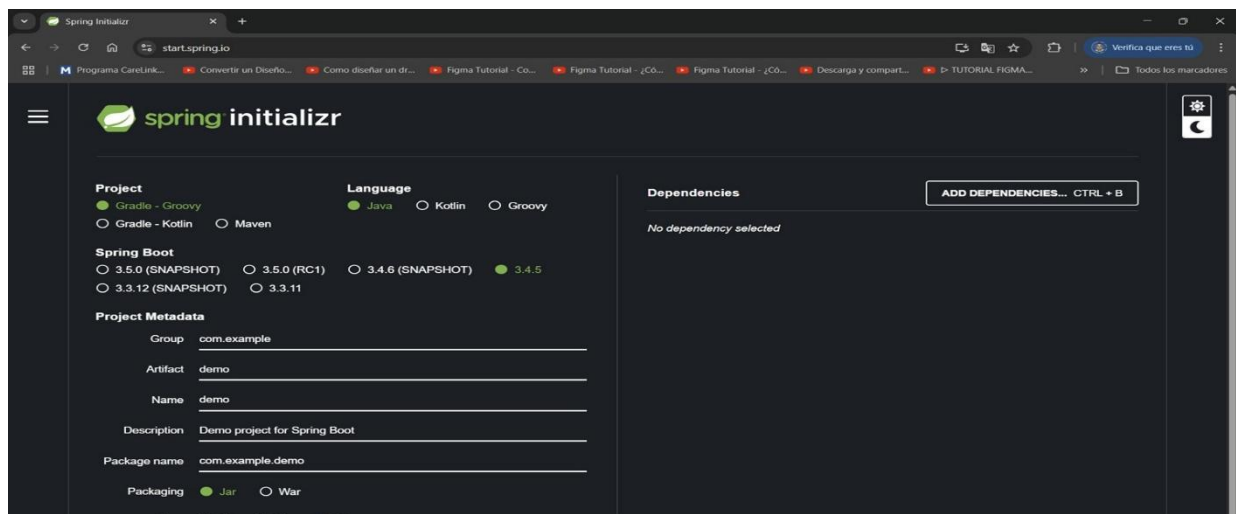



Imagen 13: Configuración Spring boot Initializr

	<i>Curso/ Grupo</i>		<i>Proyecto</i>	
	ADSO 2879723		Hardware Inventory Store FFIG	
	<i>Fecha</i>		<i>Versión</i>	<i>Código</i>
	2025/06/08		1.0	DT-MTS-01

- **Maven**
- **MySQL Server** (preferiblemente gestionado desde XAMPP o Workbench)
- **MySQL Workbench** o phpMyAdmin para crear la base de datos inicial

7.2 Creación manual de la base de datos

La base de datos debe crearse manualmente antes de ejecutar el proyecto.

- **CREATE DATABASE** hardware_store_inventory_api;

Esta base debe coincidir con el nombre configurado en el archivo **application.properties**

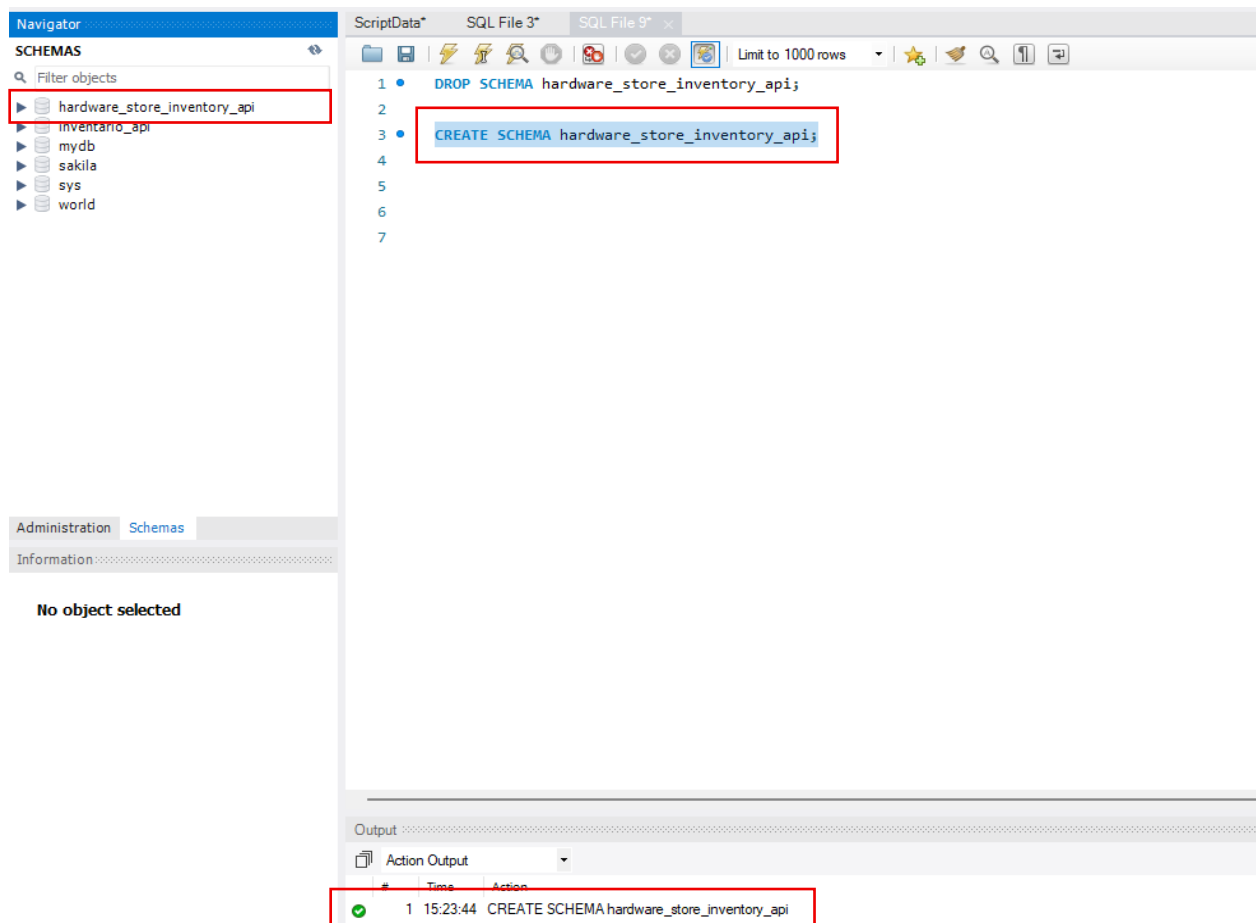



Imagen 14: Script SQL para la creación de la base de datos en Workbench

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

7.3 Configuración de conexión (application.properties)

spring.datasource.url=jdbc:mysql://localhost:3306/hardware_store_inventory_api

spring.datasource.username=root

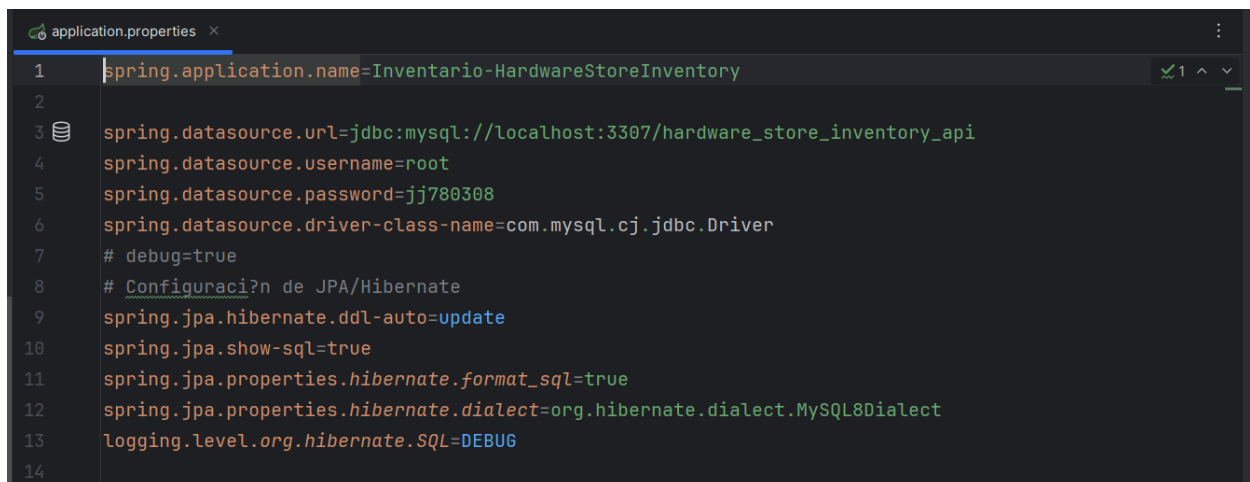
spring.datasource.password=

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

Esta configuración permite que **JPA cree automáticamente todas las tablas** a partir de las entidades del proyecto.




```

1  spring.application.name=Inventario-HardwareStoreInventory
2
3  spring.datasource.url=jdbc:mysql://localhost:3307/hardware_store_inventory_api
4  spring.datasource.username=root
5  spring.datasource.password=jj780308
6  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
7  # debug=true
8  # Configuración de JPA/Hibernate
9  spring.jpa.hibernate.ddl-auto=update
10 spring.jpa.show-sql=true
11 spring.jpa.properties.hibernate.format_sql=true
12 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
13 logging.level.org.hibernate.SQL=DEBUG
14

```

Imagen 15: Configuración para la conexión BD en la API Rest.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

7.4 Ejecución del backend

Con **Maven**:


➤ `.\mvnw.cmd spring-boot:run`

```
Terminal Local x + -
PS C:\Users\JJime\OneDrive\Escritorio\Integracion\Backend_ProyectoFinal_2025> .\mvnw.cmd spring-boot:run
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.JuDaJo.SENA.api:Inventario-HardwareStoreInventory >-----
[INFO] Building Inventario-HardwareStoreInventory 0.0.1-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> spring-boot:3.4.3:run (default-cli) > test-compile @ Inventario-HardwareStoreInventory >>>
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ Inventario-HardwareStoreInventory ---
[INFO] Copying 1 resource from src/main/resources to target/classes
[INFO] Copying 2 resources from src/main/resources to target/classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ Inventario-HardwareStoreInventory ---
[INFO] Recompiling the module because of changed source code.
[INFO] Compiling 52 source files with javac [debug parameters release 17] to target/classes
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ Inventario-HardwareStoreInventory ---
[INFO] skip non existing resourceDirectory C:\Users\JJime\OneDrive\Escritorio\Integracion\Backend_ProyectoFinal_2025\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ Inventario-HardwareStoreInventory ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 1 source file with javac [debug parameters release 17] to target\test-classes
[INFO]
[INFO] <<< spring-boot:3.4.3:run (default-cli) < test-compile @ Inventario-HardwareStoreInventory <<<
[INFO]
[INFO]
```

Imagen 17: Script para ejecutar la API Backend desde consola.

```
Database JDBC URL [connecting through datasource: hikari:hikariDataSource (hikari:pool-1)]
Database driver: undefined/unknown
Database version: 8.0
Autocommit mode: undefined/unknown
Isolation level: undefined/unknown
Minimum pool size: undefined/unknown
Maximum pool size: undefined/unknown
2025-06-10T15:30:12.067-05:00 INFO 19248 --- [Inventario-HardwareStoreInventory] [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH00004
89: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2025-06-10T15:30:12.123-05:00 DEBUG 19248 --- [Inventario-HardwareStoreInventory] [ restartedMain] org.hibernate.SQL :
create table categoria (
  id_categoria integer not null auto_increment,
  nombre_categoria varchar(100) not null,
  primary key (id_categoria)
) engine=InnoDB
Hibernate:
create table categoria (
  id_categoria integer not null auto_increment,
  nombre_categoria varchar(100) not null,
  primary key (id_categoria)
) engine=InnoDB
2025-06-10T15:30:12.156-05:00 DEBUG 19248 --- [Inventario-HardwareStoreInventory] [ restartedMain] org.hibernate.SQL :
create table empleado (
  id_empleado integer not null auto_increment,
  segundo_apellido varchar(15),
  primer_apellido varchar(255) not null,
```

Imagen 16: Creación de las entidades mapeadas por JPA desde la API Backend.

	<i>Curso/ Grupo</i>		<i>Proyecto</i>	
	ADSO 2879723		Hardware Inventory Store FFIG	
	<i>Fecha</i>		<i>Versión</i>	<i>Código</i>
	2025/06/08		1.0	DT-MTS-01

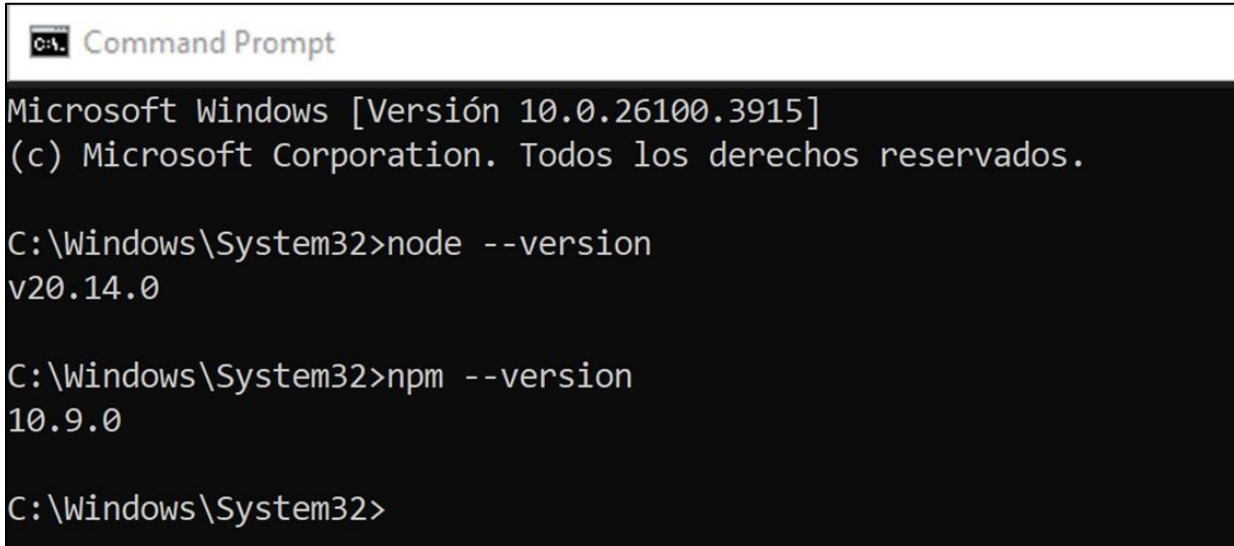
7.5 Frontend – React + Vite

Requisitos previos

- **Node.js** versión 18 o superior
- **NPM** instalado

```
npm --version
```

```
Node --version
```



Command Prompt


Microsoft Windows [Versión 10.0.26100.3915]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\System32>node --version
v20.14.0

C:\Windows\System32>npm --version
10.9.0

C:\Windows\System32>

Imagen 18: Scripts para conocer la versión instalada de Node.js y npm, salida en consola.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

7.6 Instalación de dependencias

- npm install

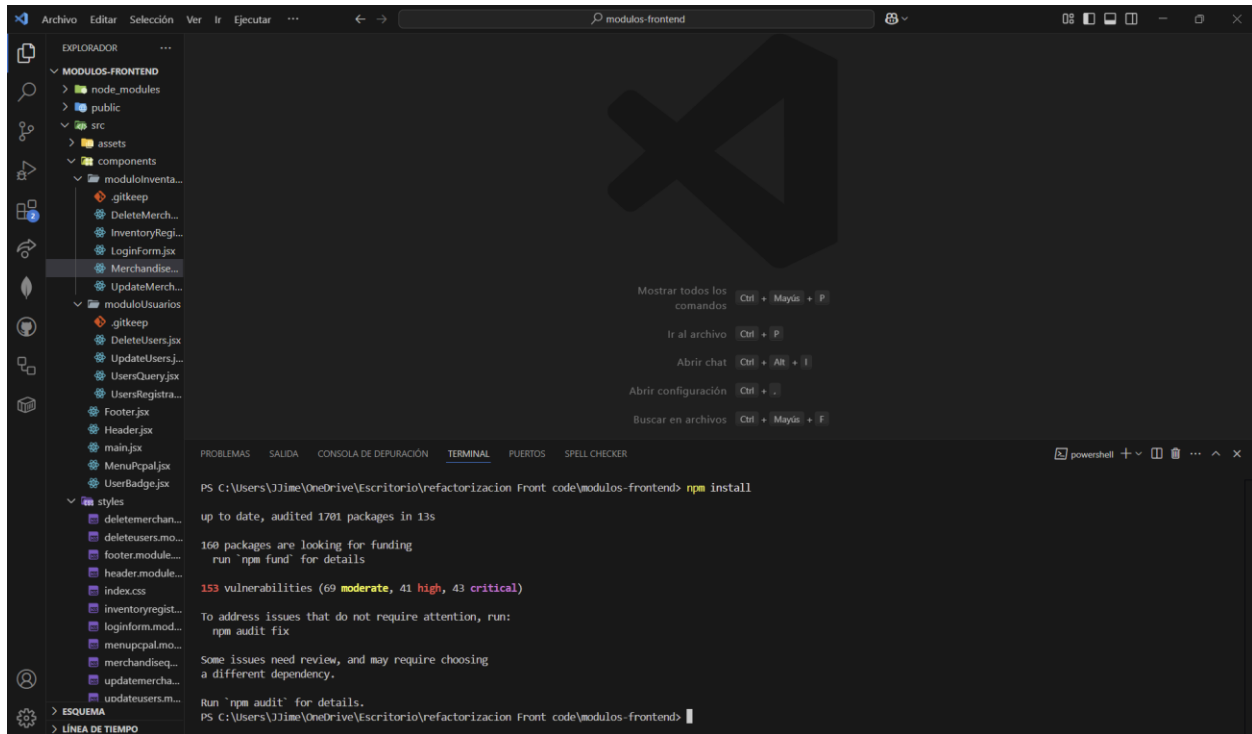



Imagen 19: Script para instalar las dependencias y librerías necesarias para el API Frontend.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

7.7 Ejecución del proyecto

- npm run dev

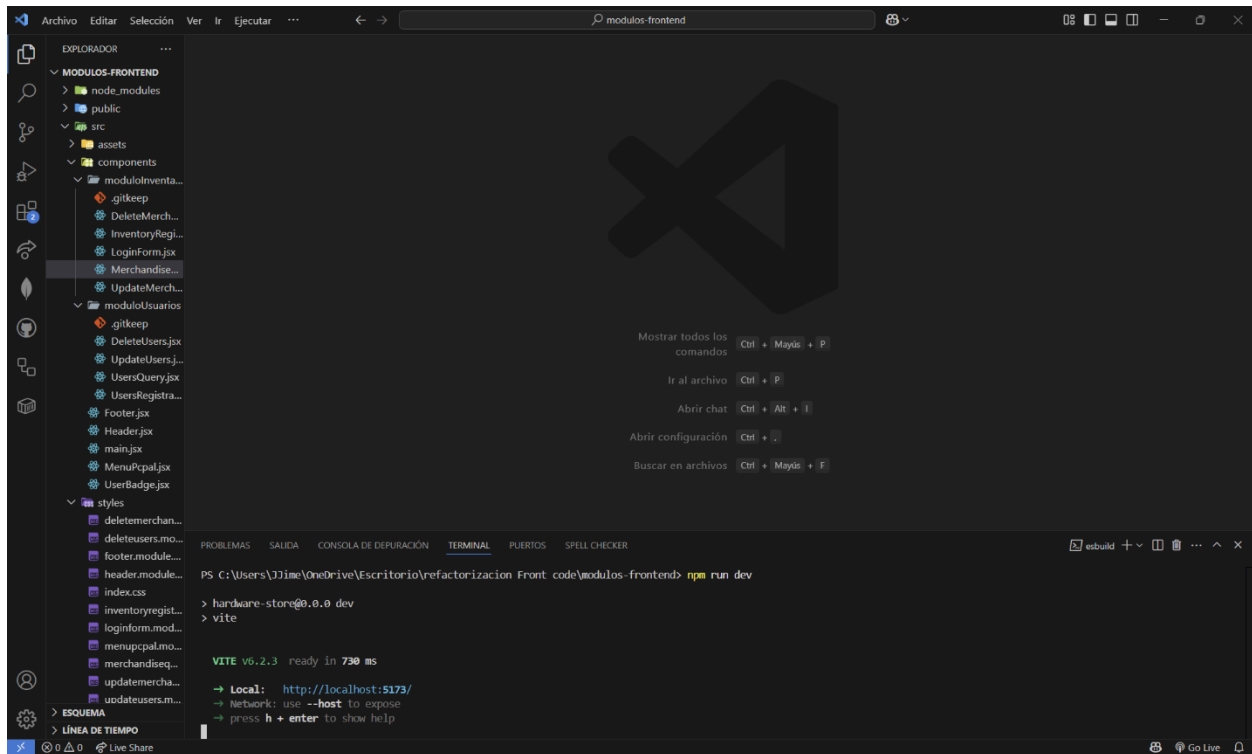



Imagen 20: Script para ejecutar el servidor local del API Frontend.

	Curso/ Grupo		Proyecto	
	ADSO 2879723		Hardware Inventory Store FFIG	
	Fecha		Versión	Código
	2025/06/08		1.0	DT-MTS-01

7.8 Dependencias utilizadas

- react
- vite
- axios
- react-router-dom
- @mui/material

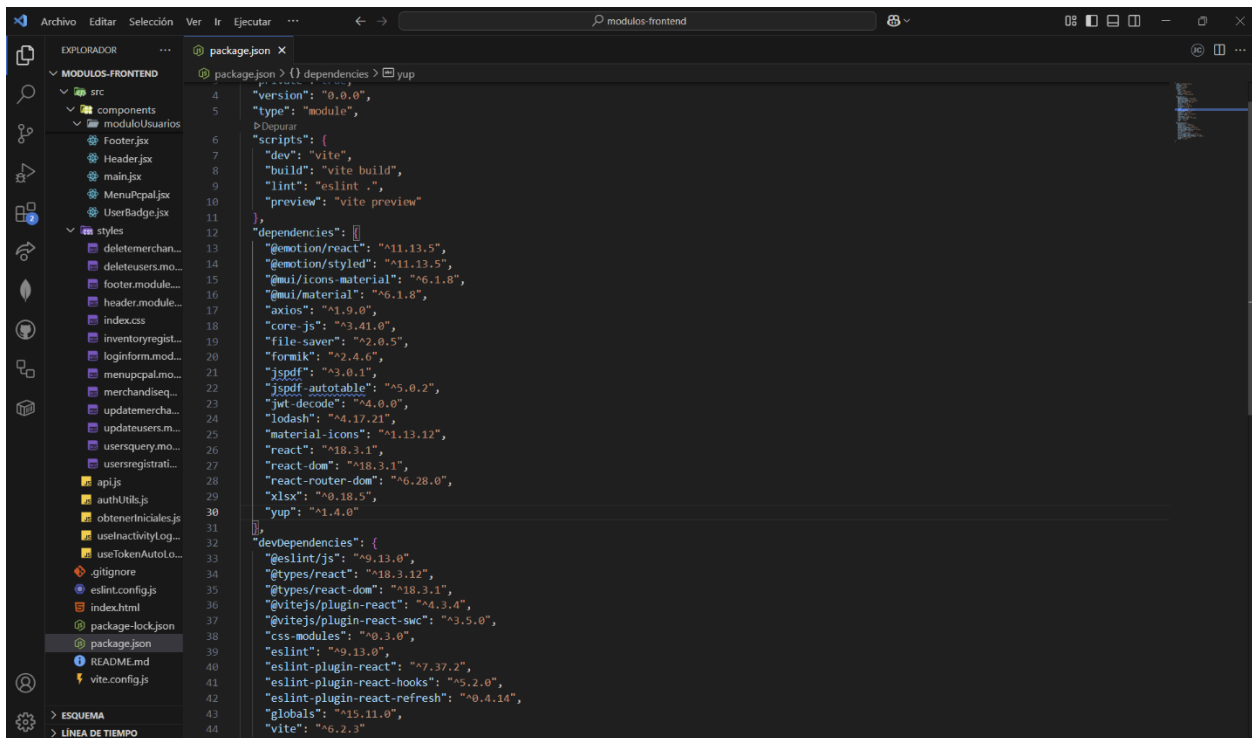



Imagen 21: Dependencias instaladas y usadas en el API Frontend.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/08	1.0	DT-MTS-01

7.9 Nota aclaratoria sobre Vite

El proyecto frontend fue creado previamente utilizando Vite con la plantilla oficial de React (vite@latest), por lo tanto, **no es necesario ejecutar nuevamente el comando npm create vite@latest.**


Todo el código fuente ya se encuentra preconfigurado y funcional.

Para ejecutar correctamente el entorno de desarrollo, basta con instalar las dependencias y ejecutar el servidor local usando los siguientes comandos:

- npm install
- npm run dev

En caso de requerir la creación de un nuevo proyecto desde cero, se recomienda el siguiente comando:

- npm create vite@latest nombre-del-proyecto -- --template react

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/08	1.0	DT-MTS-01

8 DIAGRAMA DE COMPONENTES

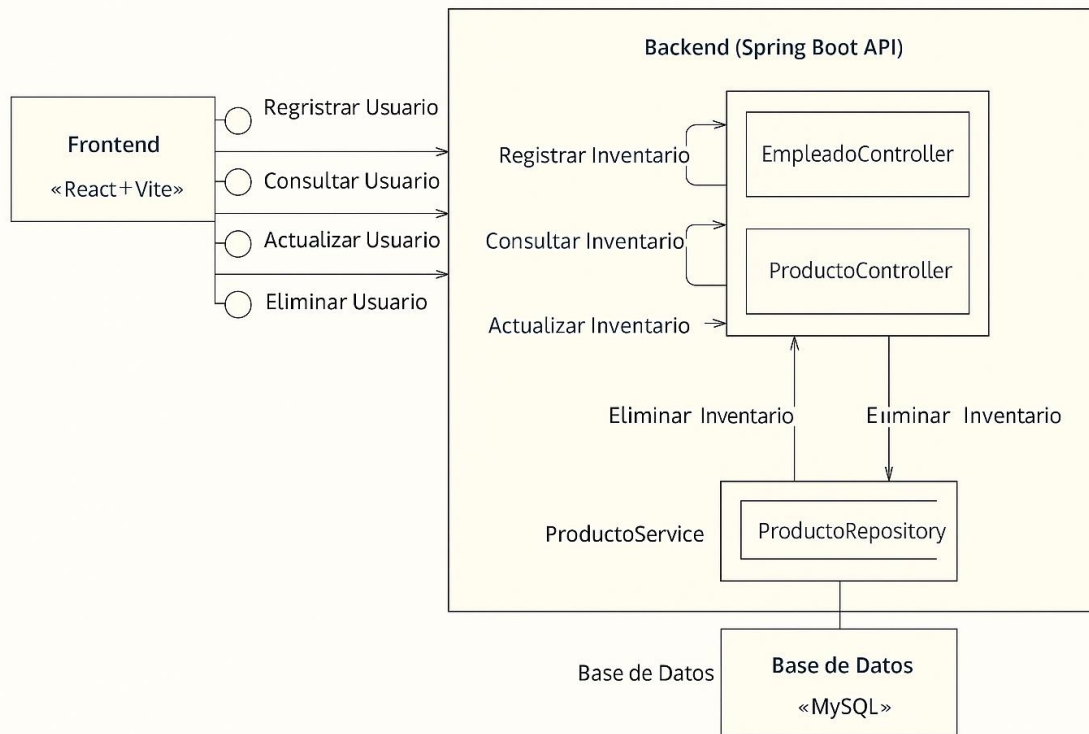


Imagen 22: Diagrama UML componente general del sistema Hardware Store Inventory, Fuente: JuDaJo System Soft (autores)


8.1 Diagrama de Componente General

Cada **componente** representa **una unidad funcional o lógica del sistema**, no necesariamente una carpeta literal o un archivo de código.

8.2 Nivel 1: Vista global (sistemas o capas)

- **Frontend (React + Vite)**

Es un *cliente* que se comunica por HTTP (usualmente con Axios) hacia los servicios REST.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/08	1.0	DT-MTS-01

- **Backend (Spring Boot)**

Aquí están los módulos *usuarios* e *inventario*, que se exponen como **endpoints REST** separados (aunque pertenezcan al mismo proyecto).

- **Base de datos (MySQL)**


Es la capa persistente a la que accede tu backend usando JPA.

8.3 Nivel 2: Componentes internos del backend

Dentro del backend, puedes tener dos componentes grandes:

- UsuariosService (o UsuariosController, EmpleadoService, etc.)
- InventarioService (y sus controladores, DTOs, etc.)

Ambos **se comunican con la base de datos**.

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/08	1.0	DT-MTS-01

8.4 Diagrama de componente modulo usuarios

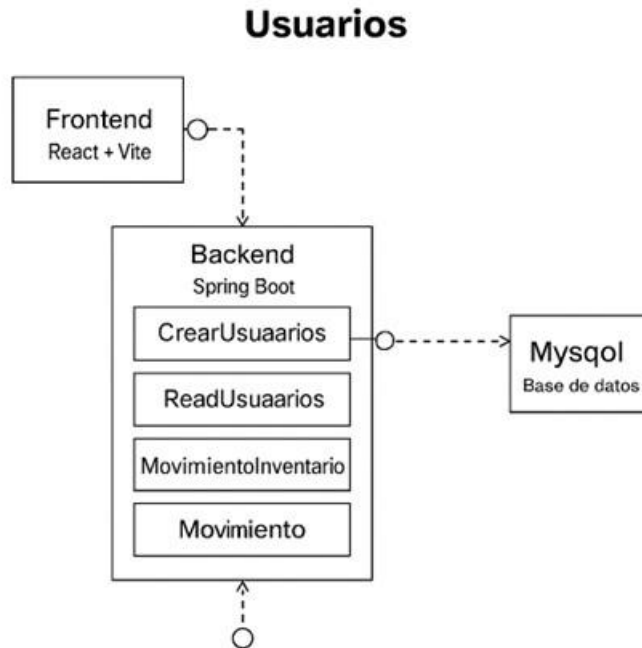


Imagen 23: Diagrama UML componente módulo usuario proyecto Hardware Store Inventory, Fuente: JuDaJo System Soft (autores)

8.5 Diagrama de componente módulo inventario

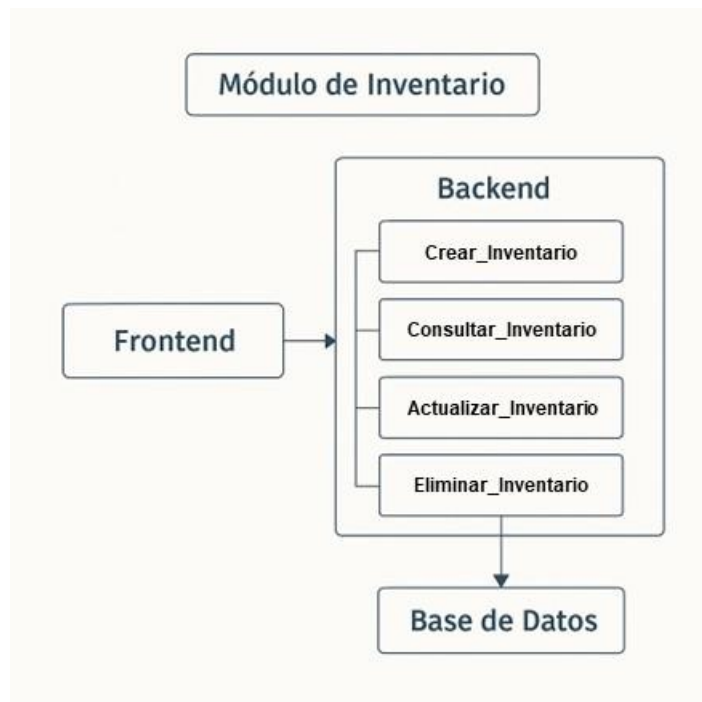



Imagen 24: : Diagrama UML componente módulo inventario proyecto Hardware Store Inventory, Fuente: JuDaJo System Soft (autores)

	Curso/ Grupo	Proyecto	
	ADSO 2879723	Hardware Inventory Store FFIG	
	Fecha	Versión	Código
	2025/06/08	1.0	DT-MTS-01

9 BIBLIOGRAFIA

1. Larman, C. (2005). *UML y Patrones: Una introducción al análisis y diseño orientado a objetos y al proceso unificado* (3.ª ed.). Pearson Educación.
2. Booch, G., Rumbaugh, J., & Jacobson, I. (2007). *El Lenguaje Unificado de Modelado: Manual de referencia* (2.ª ed.). Addison-Wesley.
3. Pressman, R. S., & Maxim, B. R. (2014). *Ingeniería de software: Un enfoque práctico* (8.ª ed.). McGraw-Hill.
4. IEEE Computer Society. (1998). *IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications*. <https://ieeexplore.ieee.org/document/720574>
5. Universidad Nacional Abierta y a Distancia (UNAD). (2020). *Guía para la elaboración del manual técnico del software*. <https://repository.unad.edu.co/handle/10596/30091>