



Elaboración de diagramas y plantillas para casos de uso del proyecto - Invextrack

Yeider Darío Gaona López

Brayan Palacios Guzmán

Programa: Análisis y Desarrollo de Software

Ficha: 3118491

Evidencia GA2-220501094-AA4-EV01

Versión: 1.1

17 de junio de 2025

Centro de Servicios Financieros (SENA)



1. Introducción

En el desarrollo de software, la correcta definición de los requisitos funcionales es esencial. El Lenguaje Unificado de Modelado (UML) permite representar de forma visual y estructurada los distintos aspectos de un sistema. Este documento presenta los diagramas de casos de uso, secuencia, actividad, estados y componentes, junto con sus plantillas asociadas al proyecto **InvetTrack**, un sistema de gestión de inventarios. Se incluye trazabilidad entre requisitos, diagramas y funcionalidades.

2. Requisitos del Software

Requisitos funcionales:

- RF-01: El sistema debe permitir el registro de usuarios.
- RF-02: El usuario debe poder iniciar sesión.
- RF-03: El administrador debe poder generar reportes.
- RF-04: El usuario debe recuperar su contraseña.

Requisitos no funcionales:

- RNF-01: La plataforma debe estar disponible 24/7.
- RNF-02: Las respuestas del sistema deben tardar menos de 3 segundos.
- RNF-03: Se debe cumplir con estándares de seguridad para autenticación.

3. Clasificación de Diagramas UML

Tabla 1

Tipos de Diagramas UML, Propósito y Justificación.

Diagrama UML	Propósito	Justificación
Casos de uso	Describir funcionalidades desde el punto de vista del usuario	Permite identificar interacciones claves
Actividad	Representar flujos de procesos	Ideal para modelar acciones como "Registrar usuario"
Secuencia	Mostrar interacción entre objetos	Fundamental para representar llamadas entre entidades
Estados	Definir comportamiento por estados	Aplicable al ciclo de vida de un producto
Clases	Mostrar estructura estática	Permite ver relaciones y atributos del sistema
Componentes	Describir arquitectura modular	Esencial para ver la división por capas (Frontend/backend)

4. Tabla de Trazabilidad

Tabla 2

Requisitos Funcionales y Diagramas Asociados.

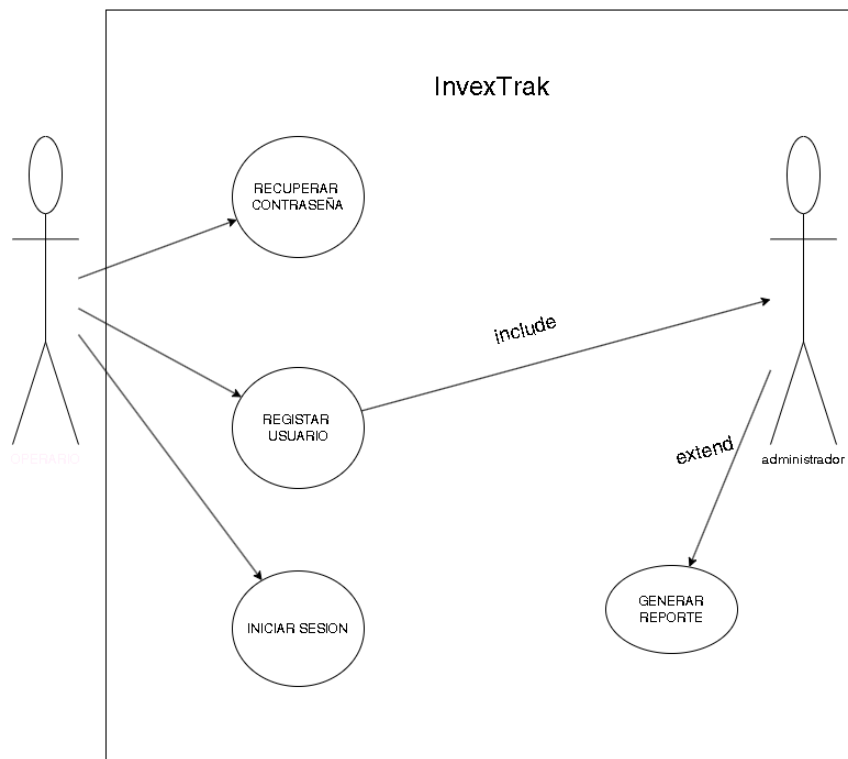
REQ-ID	Requisito	Diagramas Asociados	Casos de Uso
RF-01	Registro de usuario	CU-01, Actividad 01, Secuencia 01	CU-01

RF-02	Inicio de sesión	CU-02, Actividad 02, Secuencia 02	CU-02
RF-03	Generación de reportes	CU-03, Secuencia 03	CU-03
RF-04	Recuperar contraseña	CU-04, Actividad 04	CU-04

5. Diagrama de Casos de Uso Global

Figura 1

Diagrama de flujo de las funciones principales de InvexTrack.



6. Plantilla de Caso de Uso (CU-01)

Caso de uso: Registrar usuario

ID: CU-01

Actor: Usuario

Descripción: Permite a un nuevo usuario registrarse en el sistema proporcionando datos básicos.



Precondiciones: El usuario no debe estar registrado previamente.

Postcondiciones: El usuario queda registrado y puede iniciar sesión.

Flujo principal:

1. El usuario accede al formulario de registro.
2. Ingresa nombre, correo y contraseña.
3. El sistema valida los datos.
4. El sistema almacena la información y muestra confirmación.

Flujo alternativo:

- Si el correo ya existe, el sistema muestra un mensaje de error.

Excepciones:

- Error en la conexión a la base de datos.

Reglas de negocio:

- Los correos deben ser únicos.
- La contraseña debe tener al menos 8 caracteres.

Requisitos asociados: RF-01, RNF-03

Trazabilidad: CU-01 <-> RF-01

7. Diagramas Complementarios

Figura 2

Diagrama de Actividad - Registrar usuario.

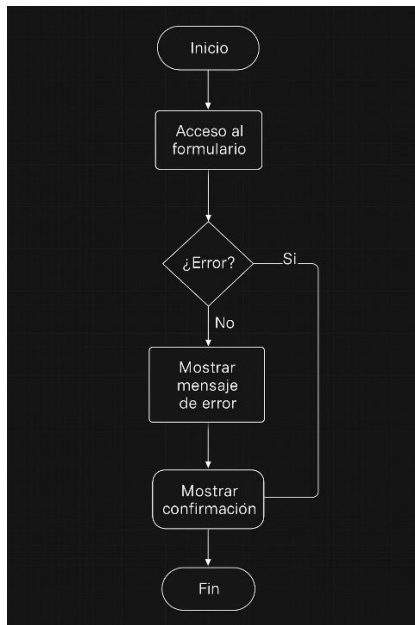


Figura 3

Diagrama de Secuencia - Registrar usuario

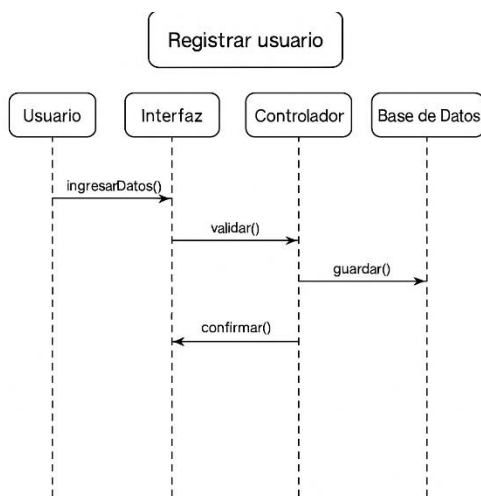
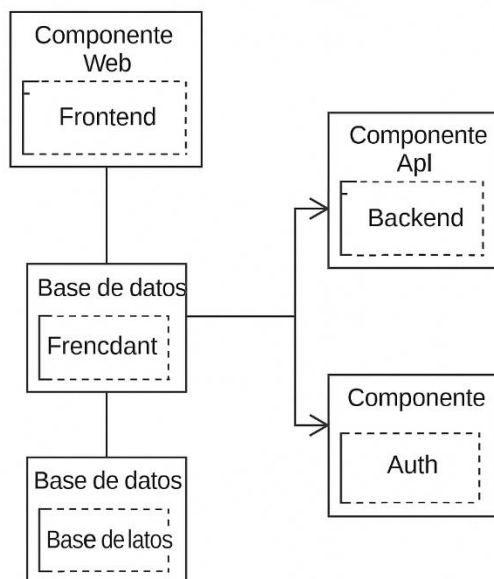


Figura 4

Diagrama de Estados – Producto

**Figura 5**

Diagrama de Componentes del Sistema





8. Historias de Usuario

- Como usuario, quiero registrarme en el sistema, para acceder a los servicios.
- Como administrador, quiero consultar reportes, para gestionar la plataforma de forma efectiva.

9. Control de Versiones

Repositorio GitHub: <https://github.com/yeider04/InvexTrack.git>

Carpeta: /documentación/uml

Archivos fuente subidos: .png, .pdf

10. Conclusión

El uso adecuado de diagramas UML junto con una plantilla estructurada de casos de uso, tabla de trazabilidad y control de versiones garantiza claridad, coherencia y trazabilidad en el desarrollo del sistema InvexTrack. Estas buenas prácticas mejoran la comunicación entre desarrolladores y usuarios, fortaleciendo la calidad del producto final.



Referencias

- Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *El lenguaje unificado de modelado (UML)* (2ª ed.). Addison-Wesley.
- Larman, C. (2004). *Aplicación de UML y patrones: Un enfoque práctico al análisis y diseño orientado a objetos* (3ª ed.). Prentice Hall.
- Pressman, R. S. (2014). *Ingeniería del software: Un enfoque práctico* (8ª ed.). McGraw-Hill.