



## **Elaboración de los diagramas del Modelo de dominio del proyecto - Invextrack**

Yeider Darío Gaona López

Brayan Palacios Guzmán

Programa: Análisis y Desarrollo de Software

Ficha: 3118491

Evidencia GA2-220501093-AA2-EV01

Versión: 1.1

26 de junio de 2025

Centro de Servicios Financieros (SENA)



## **1. Introducción**

El presente documento detalla la elaboración del modelo de dominio para el sistema de gestión de inventarios InvexTrack, desarrollado bajo el paradigma orientado a objetos.

InvexTrack es un software diseñado para optimizar el registro, consulta y control de productos, categorías, proveedores, movimientos de inventario y usuarios, proporcionando una solución eficiente para la gestión de inventarios en pequeñas y medianas empresas. A través del diagrama de clases UML, se representan las entidades principales del sistema, sus atributos y las relaciones entre ellas, sentando las bases para un desarrollo estructurado y escalable. Este documento tiene como propósito proporcionar una visión clara y estructurada del modelo conceptual, facilitando la comprensión de los requerimientos funcionales y su implementación técnica.

## **2. Alcance**

El alcance de este documento se centra en la definición del modelo de dominio para el sistema InvexTrack mediante el diseño de un diagrama de clases UML. Se identifican las clases principales, sus atributos y las relaciones que permiten gestionar el ciclo completo del inventario, desde el registro de productos hasta el control de entradas y salidas. El documento no incluye la implementación del software, el diseño de la interfaz de usuario ni la especificación técnica de la base de datos, los cuales se abordarán en fases posteriores del proyecto.

### 3. Historial de Versiones

**Tabla 1**

*Historial de versiones*

Versión	Fecha	Autor	Descripción
1.0	19/05/2025	Yeider Gaona, Brayan Palacios	Creación inicial del documento con el modelo de dominio y diagrama de clases.
1.1	26/06/2025	Yeider Gaona, Brayan Palacios	Se realizaron mejoras al modelo de dominio y se ajustó el diagrama de clases.

### 4. Objetivos

- Diseñar el diagrama de clases del sistema de gestión de inventarios InvexTrack.
- Identificar las clases principales, sus atributos y relaciones.
- Establecer una base conceptual sólida para el desarrollo orientado a objetos del proyecto.

### 5. Modelo de Clases

A continuación, se presenta el modelo de dominio del sistema InvexTrack, representado mediante un diagrama de clases UML. Este modelo refleja las entidades fundamentales y sus interrelaciones, basadas en los requerimientos funcionales identificados.

#### 5.1. Clases Principales

- Producto: Representa los ítems del inventario.



– Atributos: idProducto, nombre, descripción, cantidad, precioUnitario, idCategoria, idProveedor

- Categoría: Agrupa los productos según su tipo.
- Atributos: idCategoria, nombre, descripción
- Proveedor: Entidad que suministra los productos.
- Atributos: idProveedor, nombre, contacto, dirección
- MovimientoInventario: Registra las entradas y salidas de productos.
- Atributos: idMovimiento, tipo (entrada/salida), fecha, cantidad, idProducto
- Usuario: Gestiona el acceso al sistema.
- Atributos: idUsuario, nombre, rol, contraseña

## 5.2.Relaciones

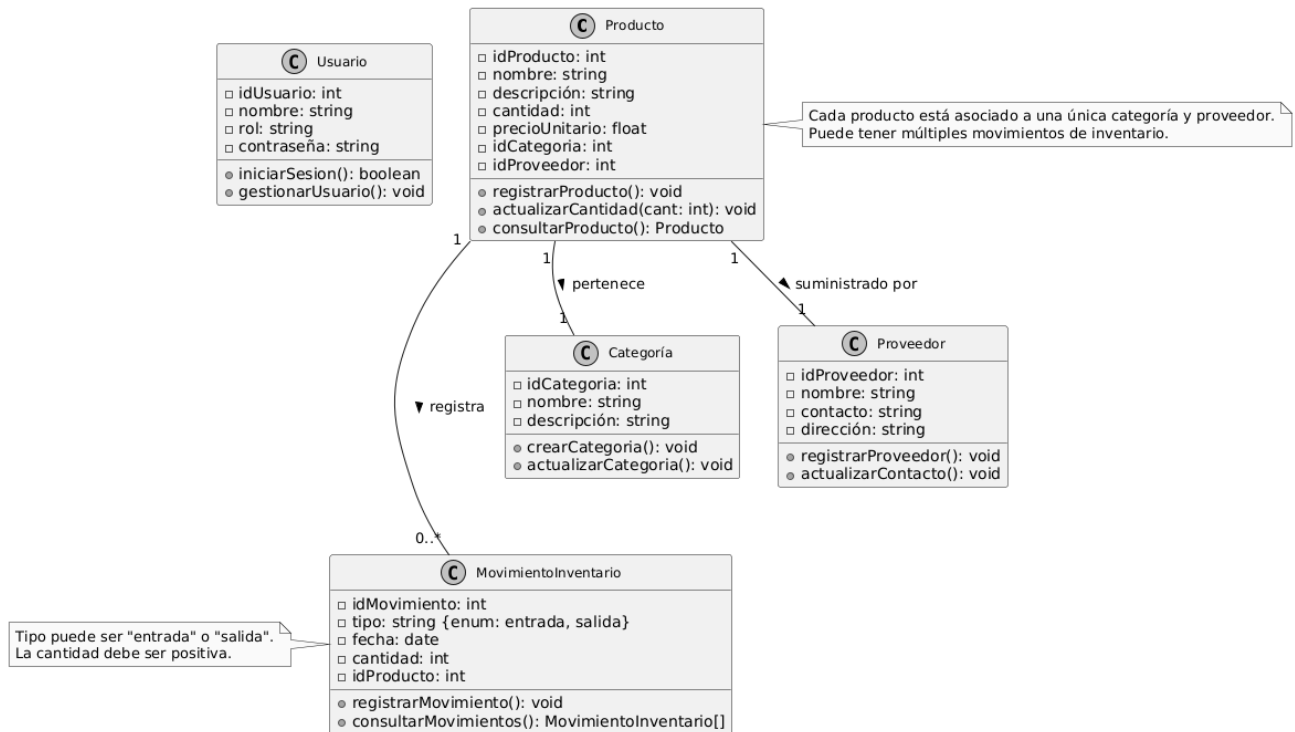
Las relaciones entre las clases permiten modelar el ciclo completo de gestión de inventario:

- Producto - Categoría: Un producto pertenece a una categoría (relación 1:N).
- Producto - Proveedor: Un producto es suministrado por un proveedor (relación 1:N).
- Producto - MovimientoInventario: Un producto puede tener múltiples movimientos de inventario (relación 1:N).
- Usuario: Administra las operaciones del sistema, pero no tiene relaciones directas con otras clases en este modelo.

## 5.3.Diagrama de Clases UML

**Figura 1**

*Diagrama de clases del sistema InvexTrack.*

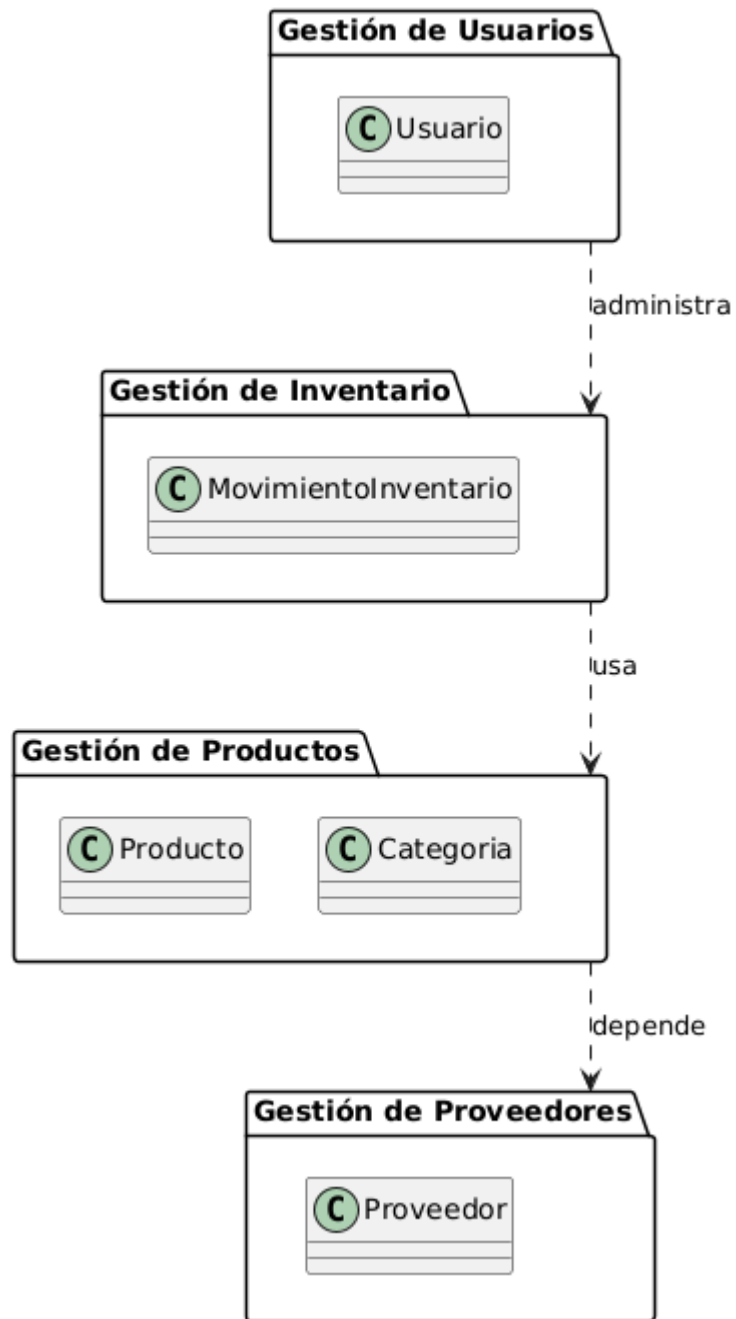


## 6. Diagrama de Paquetes

El siguiente diagrama presenta la organización modular del sistema InvexTrack en paquetes funcionales, agrupando las clases según su responsabilidad lógica.

**Figura 2**

*Diagrama de paquetes del sistema InvexTrack.*



## 7. Casos de Uso

A continuación, se documentan dos casos de uso relevantes para el sistema:

**Tabla 2***Caso de Uso 1: Registrar Producto*

Campo	Descripción
Nombre	Registrar producto
ID	CU-01
Actor principal	Operario
Precondiciones	El usuario debe estar autenticado.
Postcondiciones	El producto queda registrado en el sistema con toda la información válida.
Flujo principal	<ol style="list-style-type: none"><li>1. El operario accede al módulo de productos.</li><li>2. Ingresa los datos requeridos.</li><li>3. El sistema valida los campos.</li><li>4. El sistema guarda el producto.</li><li>5. El sistema confirma el registro.</li></ol>
Flujos alternativos	Si el producto ya existe, el sistema notifica el conflicto.
Reglas de negocio	El código del producto debe ser único.

**Tabla 3***Caso de Uso 2: Registrar Movimiento de Inventario*

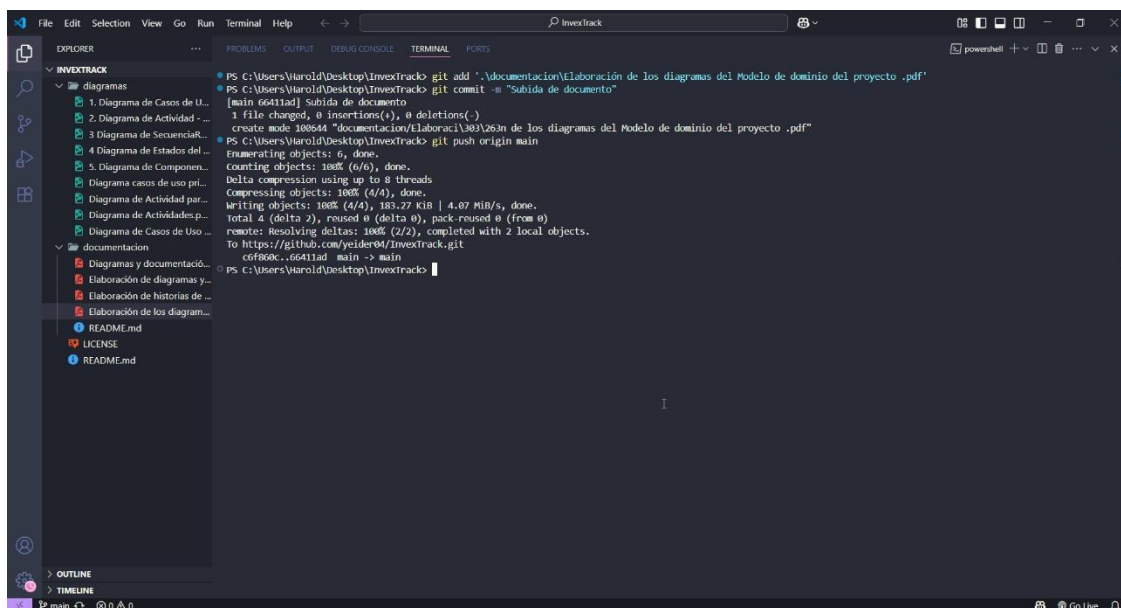
Campo	Descripción
Nombre	Registrar movimiento de inventario

ID	CU-02
Actor principal	Operario
Precondiciones	El producto debe existir en el sistema.
Postcondiciones	El movimiento queda registrado y afecta el stock. 1. El operario accede al módulo de movimientos.2.
Flujo principal	Selecciona tipo de movimiento.3. Ingresa cantidad y producto.4. El sistema actualiza el stock.5. El sistema confirma la operación.
Flujos alternativos	Si la cantidad es mayor al stock disponible (para salidas), se notifica error.
Reglas de negocio	Los movimientos deben estar fechados y registrados por usuarios autorizados.

## 8. Historial de Commits

### Figura 3

*Captura de pantalla historial de commits*



```

PS C:\Users\Haro\Desktop\InvexTrack> git add ".\documentacion\Elaboración de los diagramas del Modelo de dominio del proyecto .pdf"
PS C:\Users\Haro\Desktop\InvexTrack> git commit -m "Subida de documento"
[main 66411ad] subida de documento
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "documentacion\Elaboraci393263in de los diagramas del Modelo de dominio del proyecto .pdf"
PS C:\Users\Haro\Desktop\InvexTrack> git push origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 183.27 KiB | 4.07 MiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/yeider94/InvexTrack.git
c6f86dc..66411ad main -> main
PS C:\Users\Haro\Desktop\InvexTrack>

```





## 9. Conclusión

La elaboración del modelo de dominio y la representación estructurada mediante diagramas UML han permitido establecer una base conceptual sólida para el desarrollo del sistema de gestión de inventarios *InvexTrack*. A través del diagrama de clases, se identificaron claramente las entidades principales, sus atributos y relaciones, facilitando la comprensión del comportamiento del sistema desde una perspectiva orientada a objetos. La inclusión del diagrama de paquetes permitió visualizar la modularización funcional del sistema, promoviendo un diseño organizado y escalable.

Asimismo, la documentación de los casos de uso proporciona una visión detallada de las interacciones del usuario con el sistema, asegurando que se cubran los escenarios funcionales clave desde etapas tempranas del desarrollo. Estas representaciones gráficas y textuales no solo mejoran la trazabilidad y comunicación del equipo de trabajo, sino que también sirven como insumo fundamental para la siguiente fase de implementación técnica.