

Tarea Unidad 5

Diseño orientado a objetos. Elaboración de diagramas estructurales

Entornos de Desarrollo

Curso DAM 2024/25

Calificación obtenida: 10/10

Índice

Detalles de la tarea	3
1. Creación de la Tabla de Sustantivos y Selección de Clases	5
2. Obtención de las Relaciones entre Clases	10
Tabla resumen de las relaciones entre Clases	10
Relaciones de herencia:	11
Relaciones de Asociación	12
Relaciones de Composición	15
3. Identificación de Atributos	16
4. Identificación de Métodos	20
5. Creación del Diagrama de Clases UML	24
6. Implementación de Getters, Setters y Constructores	31
7. Refinamiento del Diagrama UML	34
8. Documentación del Proyecto	36
9. Integración del Proyecto VP-UML en NetBeans	37

Detalles de la tarea

Desde hace mucho tiempo, Ada no compra sus zapatos, botas, bolsos y otros artículos de piel en otro sitio que no sea **"El Tacón de Oro"**. Conoce a los dueños de esta antigua zapatería desde hace tiempo y sabe que la nueva generación está muy interesada en expandir el negocio.

Después de abrir varias sucursales en diferentes puntos de la ciudad y otras localidades, han decidido que el siguiente paso es la venta por Internet. Su idea es clara: **poner los artículos de temporada (disponibles en almacén) a la venta en forma de campañas**. Mientras dure una campaña, los artículos estarán visibles en la web y podrán pedirse siempre y cuando el usuario sea **socio registrado**.

El **administrativo** será el encargado de abrir y cerrar las campañas, y un **empleado de almacén** gestionará la entrega de los Artículos.

Descripción exacta del problema: "El Tacón de Oro"

Sistema de ventas en línea

Los usuarios podrán navegar por la web para ver los artículos que se venden en la tienda:

- Zapatos, bolsos y complementos.
- De cada artículo nos interesa: nombre, descripción, material, color, precio y stock.
- En el caso de los zapatos: talla y tipo.
- Para los bolsos: categoría (mochila, bandolera, fiesta, etc.).
- Para los complementos (cinturones y guantes): talla.

Organización y venta

- Los artículos se organizan en campañas de temporada (primavera/verano y otoño/invierno).
- Aunque la tienda fabrica sus propios Artículos, también puede vender artículos de otras firmas.
- Los artículos de firma se venden a través de proveedores.
- De cada firma se debe almacenar: nombre, CIF y domicilio fiscal.
- De cada proveedor se necesita: nombre, CIF y domicilio fiscal.

Gestión de usuarios y pedidos

- Los usuarios pueden registrarse para convertirse en socios.
- Un socio debe proporcionar: nombre completo, correo electrónico y dirección.
- Los socios pueden hacer pedidos de los artículos disponibles.
- Un pedido está formado por un conjunto de detalles de pedido (artículo + cantidad).
- Cada pedido debe almacenar su fecha de realización y el total a pagar.
- El pago se realiza mediante tarjeta bancaria, cuya validez es verificada por la entidad financiera.

- Se debe almacenar el número de la tarjeta utilizada.

Gestión de envíos

- Un empleado de almacén revisará y preparará los pedidos.
- Los Artículos empaquetados pasarán a un almacén hasta su reparto.
- Una empresa de transporte se encargará de la entrega.
- Cada empresa de transporte tiene varias rutas preestablecidas, asignadas según la dirección del socio.
- Se almacenará la fecha de reparto del pedido.
- Si ocurre alguna incidencia, se debe registrar la fecha y descripción del problema.

Opciones para los socios

- Los socios pueden visualizar sus pedidos y cancelarlos antes de que sean enviados.
- También pueden modificar sus datos personales.

1. Creación de la Tabla de Sustantivos y Selección de Clases

Enunciado: Extrae los sustantivos del enunciado y clasifícalos en clases, atributos o métodos.

Justificación: Explica por qué has elegido cada sustantivo como clase, atributo o método.

Pista:

- No todos los sustantivos deben ser clases. Pregunta: ¿Es una entidad independiente o solo una característica de otra?
- Organiza la información en una tabla con tres columnas: Sustantivo – Categoría – Explicación.

A continuación se muestra la tabla de Sustantivos con la clasificación en Clases, o Atributos, junto con la justificación de cada elección.

La tabla responde a la pregunta: ¿Es una entidad independiente o solo una característica de otra?

- Entidades independientes (clases) son aquellas que pueden existir por sí mismas en el sistema.
- Características de otra entidad (atributos) son propiedades que dependen de una clase.

Sustantivo	Categoría	Explicación
Tienda Online	Clase	Es una entidad independiente que gestiona Artículos, usuarios y ventas.
Usuario	Clase	Es una entidad independiente que representa a cualquier persona que interactúa con el sistema.
Artículo	Clase	Es una entidad independiente que representa los Artículos en venta.
Zapato	Clase	Es una entidad independiente , pero un subtipo de artículo.
Bolso	Clase	Es una entidad independiente , pero un subtipo de artículo.

Complemento	Clase	Es una entidad independiente , pero un subtipo de artículo.
Nombre (artículo)	Atributo	Es una característica de los artículos
Descripción (artículo)	Atributo	Es una característica de los artículos
Material (artículo)	Atributo	Es una característica de los artículos
Color (artículo)	Atributo	Es una característica de los artículos
Precio (artículo)	Atributo	Es una característica de los artículos
Stock (artículo)	Atributo	Es una característica de los artículos
Talla (zapato)	Atributo	Es una característica de zapatos
Tipo (zapato)	Atributo	Es una característica de zapatos
Categoría/Tipo (bolso)	Atributo	Es una característica de bolsos.
Tipo (complemento)	Atributo	Es una característica de complementos (cinturón o guante)
Talla (complemento)	Atributo	Es una característica de complementos.
Campaña	Clase	Es una entidad independiente que organiza la venta por temporadas.
Administrativo	Clase	Es una entidad independiente encargado de abrir y cerrar las campañas
Temporada	Atributo	Es una característica de campaña.
Proveedor	Clase	Es una entidad independiente que suministra Artículos de otras firmas a la tienda.
Nombre (proveedor)	Atributo	Es una característica de proveedor.
CIF (proveedor)	Atributo	Es una característica de proveedor.
Domicilio fiscal (proveedor)	Atributo	Es una característica de proveedor.
Firma	Clase	Es una entidad independiente que representa a las marcas externas cuyos Artículos se venden en la tienda.
Nombre (firma)	Atributo	Es una característica de Firma

CIF (firma)	Atributo	Es una característica de Firma
Domicilio fiscal (firma)	Atributo	Es una característica de Firma
Socio	Clase	Es una entidad independiente , pero un tipo especial de usuario.
Nombre (socio)	Atributo	Es una característica de los socios.
Correo electrónico (socio)	Atributo	Es una característica de los socios.
Dirección (socio)	Atributo	Es una característica de socios
Pedido	Clase	Es una entidad independiente que almacena información sobre las compras.
Fecha de realización	Atributo	Es una característica de los pedidos.
Total a pagar	Atributo	Es una característica de los pedidos.
Detalle de Pedido	Clase	Es una entidad independiente que contiene los artículos y la cantidad de artículos que hay en un Pedido
Artículo (Detalle pedido)	Atributo	Es una característica de Detalle de pedido.
Cantidad (Detalle pedido)	Atributo	Es una característica de Detalle de pedido.
Tarjeta bancaria	Clase	Es una entidad independiente que define cómo se realizó el pago del pedido
Número de tarjeta	Atributo	Es una característica de tarjeta bancaria.
Entidad financiera	Clase	Es una entidad independiente que se encarga de verificar y procesar los pagos realizados en el sistema
Almacén	Clase	Es una entidad independiente que representa el lugar físico donde se guardan los Pedidos antes de ser enviados a los socios.
Empleado de Almacén	Clase	Es una entidad independiente que representa a la persona encargada de gestionar y supervisar el proceso de almacenamiento y preparación de los Artículos para su envío.

Empresa de transporte	Clase	Es una entidad independiente que representa a la empresa encargada del envío de los pedidos.
Ruta	Clase	Es una entidad independiente que representa el trayecto realizado por la empresa de transporte para llevar el pedido desde su origen (almacén) hasta el destino (dirección del socio).
Fecha de reparto	Atributo	Es una característica de Pedido
Incidencia	Clase	Es una entidad independiente que representa un problema no previsto durante el proceso de envío del Pedido.
Descripción de incidencia	Atributo	Es una característica de Incidencia

Ahora que ya tenemos la lista completa estudiamos cada clase potencial para ver si, finalmente, es incluida en el diagrama. Para ayudarnos a decidir, utilizamos los siguientes criterios:

1. La información de la clase es necesaria para que el sistema funcione.
2. La clase posee un conjunto de atributos que podemos encontrar en cualquier ocurrencia de sus objetos. Si sólo aparece un atributo normalmente se rechazará y será añadido como atributo de otra clase.
3. La clase tiene un conjunto de operaciones identificables que pueden cambiar el valor de sus atributos y son comunes en cualquiera de sus objetos.
4. Es una entidad externa que consume o produce información esencial para la producción de cualquier solución en el sistema.

La clase se considera si cumple todos (o casi todos) los criterios.

Sustantivo	Condiciones que cumple	Incluida/Rechazada
Tienda Online	1,4	Rechazada
Usuario	1,4	Rechazada
Artículo	1,2,3	Incluida
Zapato	1,2,3	Incluida
Bolso	1,2,3	Incluida

Complemento	1,2,3	Incluida
Campaña	1,2,3	Incluida
Administrativo	1,4	Incluida
Proveedor	1,2,3,4	Incluida
Firma	1,2,3,4	Incluida
Socio	1,2,3,4	Incluida
Pedido	1,2,3	Incluida
Detalle de Pedido	1,2,3	Incluida
Pago	1,2,3	Incluida
Entidad financiera	1,2,3,4	Incluida
Almacén	1,2,3	Incluida
Empleado de Almacén	1,2,3,4	Incluida
Envío	1,2,3	Incluida
Empresa de transporte	1,2,3,4	Incluida
Ruta	1,2,3	Incluida
Incidencia	1,2,3	Incluida

2. Obtención de las Relaciones entre Clases

Enunciado: Establece las relaciones entre las clases, incluyendo herencia, asociación, composición y agregación.

Justificación:

- Explica por qué cada relación se ha definido de esa manera.
- Justifica la multiplicidad (1, 0..*, 1..*).

A partir de la tabla del apartado anterior obtenemos las siguientes Clases:

Sustantivos (Clases)		
Artículo	Firma	Empleado de Almacén
Zapato	Socio	Empresa de transporte
Bolso	Pedido	Ruta
Complemento	Detalle de Pedido	Incidencia
Campaña	Tarjeta bancaria	
Administrativo	Entidad financiera	
Proveedor	Almacén	

A continuación se muestran las relaciones entre las clases:

Tabla resumen de las relaciones entre Clases

Tipo de relación	clase A	clase B	Multiplicidad
Herencia	Artículo	Zapato	No aplica
Herencia	Artículo	Bolso	No aplica
Herencia	Artículo	Complemento	No aplica
Asociación	Artículo	Detalle Pedido	[Artículo] (1..*) — (0..*) [Detalle de Pedido]
Asociación	Socio	Pedido	[Socio] (1) — (0..*) [Pedido]
Asociación	Entidad financiera	Tarjeta bancaria	[Entidad Financiera] (1) — (0..*) [Tarjeta Bancaria]

Asociación	Empleado almacén	Pedido	[Empleado de Almacén] (1) — (0..*) [Pedido]
Asociación	Pedido	Almacén	[Pedido] (0..*) — (1) [Almacén]
Asociación	Ruta	Pedido	[Ruta] (1) — (0..*) [Pedido]
Asociación	Empresa Transporte	Ruta	[Empresa de Transporte] (1) — (0..*) [Ruta]
Asociación	Campaña	Artículo	[Campaña] (1) — (1..*) [Artículo]
Asociación	Administrativo	Campaña	[Administrativo] (1) — (0..*) [Campaña]
Asociación	Firma	Artículo	[Firma] (0..1) — (0..*) [Artículo]
Asociación	Proveedor	Artículo	[Proveedor] (1) — (0..*) [Artículo]
Composición	Pedido	Detalle Pedido	[Pedido] (1) ◆ — (1..*) [Detalle de Pedido]
Composición	Pedido	Tarjeta bancaria	[Pedido] (1) ◆ — (1) [Tarjeta Bancaria]
Composición	Pedido	Incidencia	[Pedido] (1) ◆ — (0..*) [Incidencia]

A continuación se muestra la justificación de las relaciones según el tipo de relación:

Relaciones de herencia:

La herencia se utiliza cuando una clase hija hereda atributos y métodos de una clase padre. En las relaciones de herencia la multiplicidad no aplica, ya que cada instancia de la clase hija es una instancia de la clase padre. En UML la relación de herencia se indica con una flecha vacía (←) apuntando desde la subclase hacia la superclase.

Clase Padre	Símbolo	Clase Hija	Justificación	Multiplicidad
Artículo	←	Zapato	Un Zapato es un tipo especializado de Artículo, heredando sus atributos y métodos generales.	No aplica
Artículo	←	Bolso	Un Bolso es un tipo especializado de Artículo, que comparte características comunes y define atributos específicos.	No aplica

Artículo	←	Complemento	Un Complemento es un tipo especializado de Artículo, heredando la estructura base y agregando particularidades.	No aplica
Usuario	←	Socio	Un Socio es una especialización de Usuario, con atributos y comportamientos adicionales orientados a la fidelización.	No aplica

Relaciones de Asociación

La asociación se utiliza cuando dos clases están relacionadas, pero pueden existir de manera independiente. Una clase está conectada o interactúa de alguna manera con otra. Esta relación puede ser simple o contener información adicional sobre cómo interactúan las clases, pero no implica una fuerte dependencia entre ellas. En UML la relación de asociación se representa con una línea que conecta las clases (—).

Clase A	Clase B	Justificación	Multiplicidad
Artículo	Detalle Pedido	<p>Un Artículo puede existir sin estar asociado a un Detalle de Pedido. Es decir, puede estar disponible en la tienda sin haber sido vendido o estar incluido en un pedido..</p> <p>Un Detalle de Pedido, sin embargo, no tiene sentido por sí solo sin un Artículo. Un Detalle Pedido representa un Artículo dentro de un pedido, por lo que siempre estará asociado con al menos un Artículo.</p>	<p>[Artículo] (1..*) — (0..*) [Detalle de Pedido]</p> <p>Un Artículo puede aparecer en ninguno o muchos Detalles de Pedido (0..*).</p> <p>Un Detalle Pedido siempre tiene al menos uno o más artículos (1..*).</p>
Socio	Pedido	La relación entre Socio y Pedido es de asociación, ya que un Socio puede realizar múltiples Pedidos, y cada Pedido está vinculado a un	<p>[Socio] (1) — (0..*) [Pedido]</p> <p>Un Socio puede realizar cero o más Pedidos (0..*).</p>

		Socio que lo realiza. Un Pedido no tiene sentido sin un Socio que lo haya realizado, pero un Socio puede no tener ningún pedido realizado en un momento dado (por ejemplo, si no ha realizado ninguna compra).	Cada Pedido está vinculado a un único Socio (1)
Entidad financiera	Tarjeta bancaria	Las Tarjetas bancarias pueden existir sin una Entidad financiera (aunque no serían válidas sin la entidad que las emite), y las Entidades financieras pueden seguir existiendo sin una tarjeta en particular.	<p>[Entidad Financiera] (1) — (0..*) [Tarjeta Bancaria]</p> <p>Una Entidad financiera puede emitir muchas Tarjetas bancarias (0..*).</p> <p>Cada Tarjeta bancaria está asociada a una única Entidad financiera (1).</p>
Empleado almacén	Pedido	Son entidades independientes. Un Empleado de Almacén puede estar trabajando en el sistema sin estar asociado a un pedido en ese momento, y un Pedido puede existir antes de ser gestionado por un Empleado de Almacén.	<p>[Empleado de Almacén] (1) — (0..*) [Pedido]</p> <p>Un Empleado de Almacén puede gestionar 0 o más Pedidos (0..*)</p> <p>Un Pedido debe ser gestionado por un Empleado de Almacén (1)</p>
Pedido	Almacén	Son entidades independientes. Un Pedido puede existir sin estar asociado a un Almacén en ciertos momentos (por ejemplo, antes de la preparación o si el pedido ya ha sido enviado). Del mismo modo, un Almacén puede existir sin tener pedidos asociados en su almacenamiento en un momento dado (por ejemplo, si el almacén está vacío o no tiene pedidos en proceso).	<p>[Pedido] (0..*) — (1) [Almacén]</p> <p>Un Pedido se almacena en un solo Almacén (1)</p> <p>Un Almacén puede almacenar múltiples Pedidos.(0..*)</p>

Ruta	Pedido	Ambas entidades son independientes, pero se vinculan cuando un pedido es asignado a una ruta para su entrega.	[Ruta] (1) — (0..*) [Pedido] Cada Ruta puede tener muchos Pedidos(0..*) Un pedido sólo puede tener una ruta asignada para su entrega(1)
Empresa Transporte	Ruta	Son entidades independientes. Si una Empresa de Transporte es eliminada, sus Rutas no necesariamente se eliminarían ya que pueden ser redefinidas o gestionadas por otra empresa si es necesario.	[Empresa de Transporte] (1) — (0..*) [Ruta] Una empresa de transporte tiene múltiples rutas (0..*) Cada ruta pertenece a una sola empresa (1).
Campaña	Artículo	Ambas clases existen de forma independiente. Una Campaña organiza y agrupa artículos durante un periodo determinado, pero los artículos pueden existir fuera de una campaña (en el almacén, por ejemplo). Cuando una Campaña termina, los artículos no desaparecen, sino que pueden asignarse a otra campaña en el futuro.	[Campaña] (1) — (1..*) [Artículo] Cada campaña tiene al menos un artículo (1..*). Cada artículo pertenece a una única campaña activa (1).
Administrativo	Campaña	Ambas entidades existen de manera independiente. La relación se establece sólo cuando un Administrativo gestiona una Campaña activa	[Administrativo] (1) — (0..*) [Campaña] Un Administrativo está asociado con una Campaña activa en un momento dado pero podría gestionar múltiples campañas a lo largo del tiempo, por lo tanto, puede gestionar cero o más campañas (0..*) Una Campaña está gestionada por un Administrativo (1)

Firma	Artículo	<p>Tanto Firma como Artículo son entidades que pueden existir de manera independiente.</p> <p>Un artículo puede o no estar asociado a una firma, dependiendo de si es un artículo externo o uno fabricado por la propia tienda (controlado por el atributo "origen" en la clase Artículo).</p>	<p>[Firma] (0..1) — (0..*) [Artículo]</p> <p>Una firma puede fabricar 0 o más artículos (0..*)</p> <p>Un artículo puede no tener firma (si es interno) o estar asociado a una firma (si es externo). (0..1).</p>
Proveedor	Artículo	<p>Tanto Proveedor como Artículo son entidades independientes; la existencia de un artículo no depende del proveedor, y viceversa.</p>	<p>[Proveedor] (1) — (0..*) [Artículo]</p> <p>Un Proveedor puede suministrar varios Artículos de una firma (0..*).</p> <p>Cada Artículo (cuando es de una firma) está asociado a un único Proveedor(1).</p>

Relaciones de Composición

La composición indica una relación fuerte donde la existencia de una clase depende totalmente de otra. Una clase compuesta no puede existir sin la clase contenedora. Si la clase contenedora deja de existir, la clase compuesta también desaparece. En UML el símbolo ◆ representa la composición.

Clase Contenedora	Clase Compuesta	Justificación	Multiplicidad
Pedido	Detalle Pedido	<p>Un Detalle de Pedido sólo tiene sentido dentro de un Pedido. Si el Pedido se elimina, sus detalles también desaparecen.</p>	<p>[Pedido] (1) ◆ — (1..*) [Detalle de Pedido]</p> <p>Un Pedido debe tener uno o más Detalles de Pedido (1...*).</p> <p>Un Detalle Pedido siempre está asociado a exactamente un Pedido (1).</p>

Pedido	Tarjeta bancaria	Pedido y Tarjeta bancaria tienen una fuerte dependencia. La Tarjeta Bancaria no tiene sentido sin un Pedido asociado. Si el Pedido es eliminado, la Tarjeta Bancaria también debería eliminarse, ya que solo tiene relevancia dentro del contexto del pedido.	Pedido (1) ◆—— (1) Tarjeta Bancaria Cada Pedido debe tener una Tarjeta bancaria asociada (1), ya que todos los pedidos requieren una tarjeta para el pago. La Tarjeta bancaria sólo tiene sentido en el contexto de un Pedido (1).
Pedido	Incidencia	La Incidencia depende completamente de la existencia del Pedido, y su existencia no tiene sentido sin estar asociada a un pedido específico.	[Pedido] (1) ◆—— (0..*) [Incidencia] Un pedido puede tener cero o más incidencias asociadas (0..*) Una Incidencia está asociada a un único Pedido (1)

3. Identificación de Atributos

Enunciado: Define los atributos esenciales de cada clase.

Justificació: Explica por qué cada atributo es necesario y dónde se utilizará.

Tabla resumen de los atributos

Clase	Atributos
Artículo	Nombre, Descripción, Material, Color, Precio, Stock
Zapato	Talla, Tipo
Bolso	Categoría
Complemento	Talla, Tipo
Campaña	Temporada
Administrativo	-
Proveedor	Nombre, CIF, Domicilio Fiscal
Firma	Nombre, CIF, Domicilio Fiscal
Socio	Nombre completo, Correo electrónico, Dirección

Pedido	Fecha de realización, Total a pagar, Fecha de reparto
Detalle de Pedido	Cantidad, Artículo
Tarjeta bancaria	Número de tarjeta
Entidad financiera	-
Almacén	-
Empleado de Almacén	-
Empresa de transporte	-
Ruta	-
Incidencia	Fecha de incidencia, Descripción de incidencia

Justificación de por qué el atributo es necesario, especificando dónde se utilizará el atributo.

Clase	Atributo	¿Porqué es necesario?	Dónde se utilizará
Artículo	Nombre	Es el nombre del Artículo, esencial para su identificación.	En la visualización de Artículos en la tienda y en los pedidos.
	Descripción	Proporciona información adicional sobre el Artículo.	En la descripción detallada de los Artículos en la tienda online.
	Material	Especifica el material del Artículo, útil para clasificar o filtrar artículos.	En la búsqueda o filtrado de Artículos por material.
	Color	El color del Artículo, importante para Artículos como ropa, zapatos, bolsos, etc.	En la visualización y filtrado de Artículos por color.
	Precio	Indica el precio del Artículo.	En la visualización de precios de los Artículos y en la facturación.

	Stock	Define la cantidad disponible del artículo en inventario.	En el control de inventarios y la gestión de disponibilidad de artículos.
Zapato	Talla	Especifica la talla del zapato, necesario para clasificar este tipo de Artículos.	En la visualización y la compra de zapatos para seleccionar la talla.
	Tipo	Define el tipo de zapato, por ejemplo, casual, deportivo, formal, etc.	En el filtrado de Artículos por tipo de zapato.
Bolso	Categoría	Categoriza el bolso, útil para filtrar y organizar diferentes tipos de bolsos (mochila, bandolera, fiesta,...).	En la visualización y filtrado de bolsos por categoría.
Complemento	Tipo	Define el tipo de complemento, como cinturón y guante	En el filtrado de Artículos por tipo de complemento.
	Talla	Especifica la talla de los complementos, como guantes o cinturones.	En la visualización y compra de complementos, especialmente para guantes o cinturones.
Campaña	Temporada	Define la temporada de la campaña (primavera/verano y otoño/invierno).	En la promoción y filtrado de Artículos por temporada durante las campañas.
Administrativo	-		
Proveedor	Nombre	Nombre del proveedor, utilizado para identificar a la entidad proveedora.	En la gestión de proveedores y en la asignación de Artículos a proveedores.
	CIF	Código de identificación fiscal del proveedor.	En la facturación, pagos y registro de proveedores.
	Domicilio fiscal	Dirección fiscal del proveedor, necesaria para contacto y formalidades legales.	En la gestión administrativa de proveedores y en los trámites fiscales.
Firma	Nombre	Nombre de la firma, utilizado para identificar la marca de Artículo externos.	En la visualización de marcas y en la gestión de Artículos de diferentes firmas.

	CIF	Código de identificación fiscal de la firma, similar al de los proveedores.	En la facturación y los trámites fiscales relacionados con firmas externas.
	Domicilio fiscal	Dirección fiscal de la firma, importante para contactar y realizar gestiones administrativas.	En los registros y comunicaciones oficiales de la firma.
Socio	Nombre completo	Es fundamental para identificar a cada socio de manera única.	En la gestión de socios, para identificación y personalización de los pedidos y comunicaciones.
	Correo electrónico	Es el correo electrónico del socio, importante para la comunicación y validación de cuentas.	En la gestión de cuentas, comunicaciones y marketing dirigido a socios.
	Dirección	Dirección del socio, necesaria para el envío de Artículos y la facturación.	En el proceso de envío de Artículos y la facturación del socio.
Pedido	Fecha de realización	Indica la fecha en la que se realizó el pedido, importante para el seguimiento y la gestión de plazos.	En el sistema de gestión de pedidos y en la programación de envíos.
	Total a pagar	Total que debe pagar el socio por el pedido realizado.	En el cálculo del pago y la facturación del pedido.
	Fecha de reparto	Define la fecha en la que se realiza el reparto del pedido.	En la programación y seguimiento de los envíos a los socios.
Detalle de Pedido	Cantidad	Especifica la cantidad de Artículos de cada tipo dentro de un pedido.	En el sistema de gestión de pedidos y en el cálculo total del pedido.
	Artículo	Cada detalle se asocia con un artículo específico (Artículo) que está siendo pedido.	En la gestión de artículos dentro de un pedido, para su visualización y stock.
Tarjeta bancaria	Número de tarjeta	Número de tarjeta utilizado para realizar el pago, necesario para procesar la transacción.	En el procesamiento de pagos y en la validación de la transacción.

Entidad financiera	-		
Almacén	-		
Empleado de almacén	-		
Empresa de transporte	-		
Ruta	-		
Incidencia	Fecha de la incidencia	Fecha en la que se produjo la incidencia	En la gestión de incidencias relacionadas con el proceso de envío.
	Descripción de incidencia	Descripción del problema que ha ocurrido durante el envío, importante para la resolución de incidencias.	En la gestión de incidencias relacionadas con el proceso de envío.

4. Identificación de Métodos

Enunciado: Define los métodos que necesita cada clase.

Justificación: Explica qué hace cada método y en qué parte del sistema es necesario.

Tabla resumen de los métodos

Clase	Método
Administrativo	abrirCampaña() cerrarCampaña()
Proveedor	registrarProveedor() venderArticuloAFirma()
Firma	registrarFirma()
Socio	registrarSocio() modificarDatosPersonales() realizarPedido()

	verPedidos() cancelarPedido()
Pedido	agregarDetalle(DetallePedido detalle) calcularTotal() registrarFechaPedido() asignarTarjeta(TarjetaBancaria tarjeta) procesarPago(EntidadFinanciera entidad) → boolean registrarFechaReparto()
Detalle de Pedido	calcularSubtotal() → double
Tarjeta bancaria	esValida() → boolean
Entidad financiera	verificarTarjeta(TarjetaBancaria tarjeta, double monto) → boolean
Almacén	almacenarPedido()
Empleado de Almacén	prepararPedido()
Empresa de transporte	asignarRutaPedido()
Ruta	obtenerRutaParaPedido()
Incidencia	registrarIncidencia()

Justificación de los métodos:

Métodos			
Clase	Método	¿Qué hace?	¿En qué parte del sistema es necesario?
Artículo	-		
Zapato	-		
Bolso	-		
Complemento	-		
Campaña	-		

Administrativo	abrirCampaña()	Inicia una nueva campaña para la venta de artículos.	Gestión de campañas
	cerrarCampaña()	Finaliza una campaña activa.	Gestión de campañas
Proveedor	registrarProveedor()	Agrega un nuevo proveedor con sus datos.	Administración de proveedores
	venderArticuloDeFirma()	Permite que un proveedor venda artículos de una firma	Gestión de artículos externos
Firma	registrarFirma()	Registra una nueva firma con su información.	Administración de firmas
Socio	registrarSocio()	Permite que un usuario se registre como socio.	Panel de usuario / Registro
	modificarDatosPersonales()	Actualiza la información del socio (nombre, correo, etc.).	Panel de usuario / Configuración de cuenta
	realizarPedido()	Crea un nuevo pedido con artículos seleccionados.	Proceso de compra / Checkout
	verPedidos()	Muestra el historial de pedidos del socio.	Panel de usuario / Seguimiento de pedidos
	cancelarPedido()	Permite cancelar un pedido antes de ser enviado.	Panel de usuario / Gestión de pedidos
Pedido	agregarDetallePedido(DetallePedido detalle)	Agrega un detalle de pedido con artículo y cantidad.	Proceso de compra / Creación del pedido
	calcularTotal()	Calcula el total a pagar sumando los	Resumen y confirmación del pedido

		subtotales de los detalles.	
	registrarFechaPedido()	Almacena la fecha en que se realizó el pedido.	Registro de pedidos
	asignarTarjeta(TarjetaBancaria tarjeta)	Asocia una tarjeta bancaria al pedido para el pago.	Proceso de pago
	procesarPago(EntidadFinanciera entidad) → boolean	Verifica el pago con la entidad financiera.	Validación y cobro del pedido
	registrarFechaReparto()	Guarda la fecha en la que el pedido será entregado.	Seguimiento de pedidos
Detalle de Pedido	calcularSubtotal() → double	Calcula el subtotal (precio * cantidad) del artículo.	Resumen del pedido / Cálculo del total
Tarjeta bancaria	esValida() → boolean	Verifica si la tarjeta es válida (fecha de expiración, formato).	Proceso de pago / Validación antes de la compra
Entidad financiera	verificarTarjeta(TarjetaBancaria tarjeta, double monto) → boolean	Confirma si la tarjeta tiene fondos y autoriza el pago.	Validación del pago con la entidad financiera
Almacén	almacenarPedido()	Registra el pedido en el almacén antes de su reparto.	Gestión de almacén
Empleado de Almacén	prepararPedido()	Empaqueta los artículos del pedido.	Gestión de pedidos en almacén
Empresa de transporte	asignarRutaPedido()	Asigna una ruta a un pedido en función de la dirección del socio.	Gestión de envíos / Logística
Ruta	obtenerRutaParaPedido()	Devuelve la ruta correspondiente para la entrega del pedido.	Gestión de envíos / Logística

Incidencia	registrarIncidencia()	Registra una incidencia en la entrega del pedido.	Gestión de incidencias / Atención al cliente
------------	-----------------------	---	--

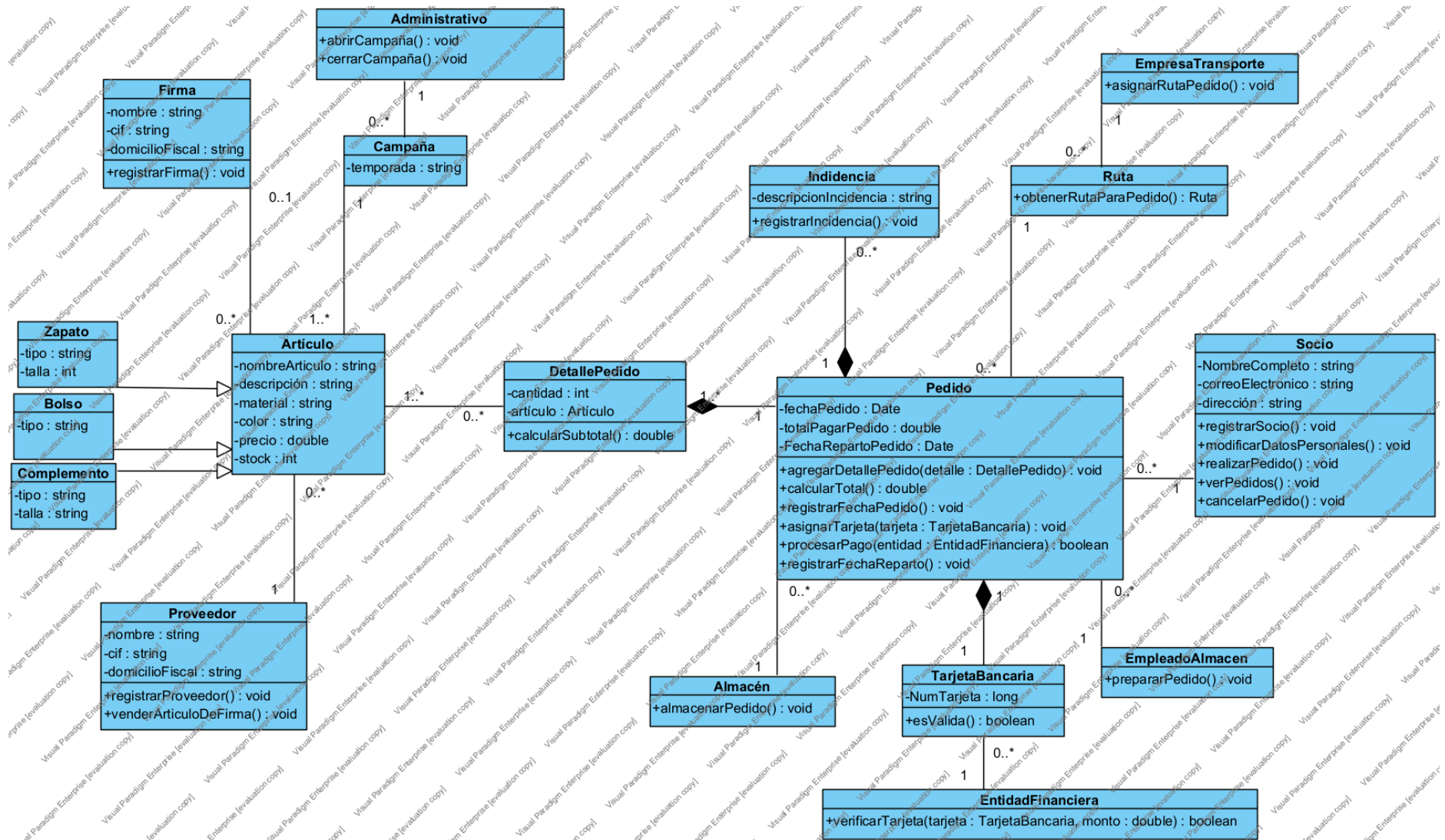
5. Creación del Diagrama de Clases UML

Enunciado: Dibuja el diagrama de clases UML en Visual Paradigm.

Justificación:

- Explica cómo has estructurado el diagrama y si hay relaciones que puedan optimizarse.
- Pantallazo.

A continuación se muestra la versión 1 del Diagrama de clases:



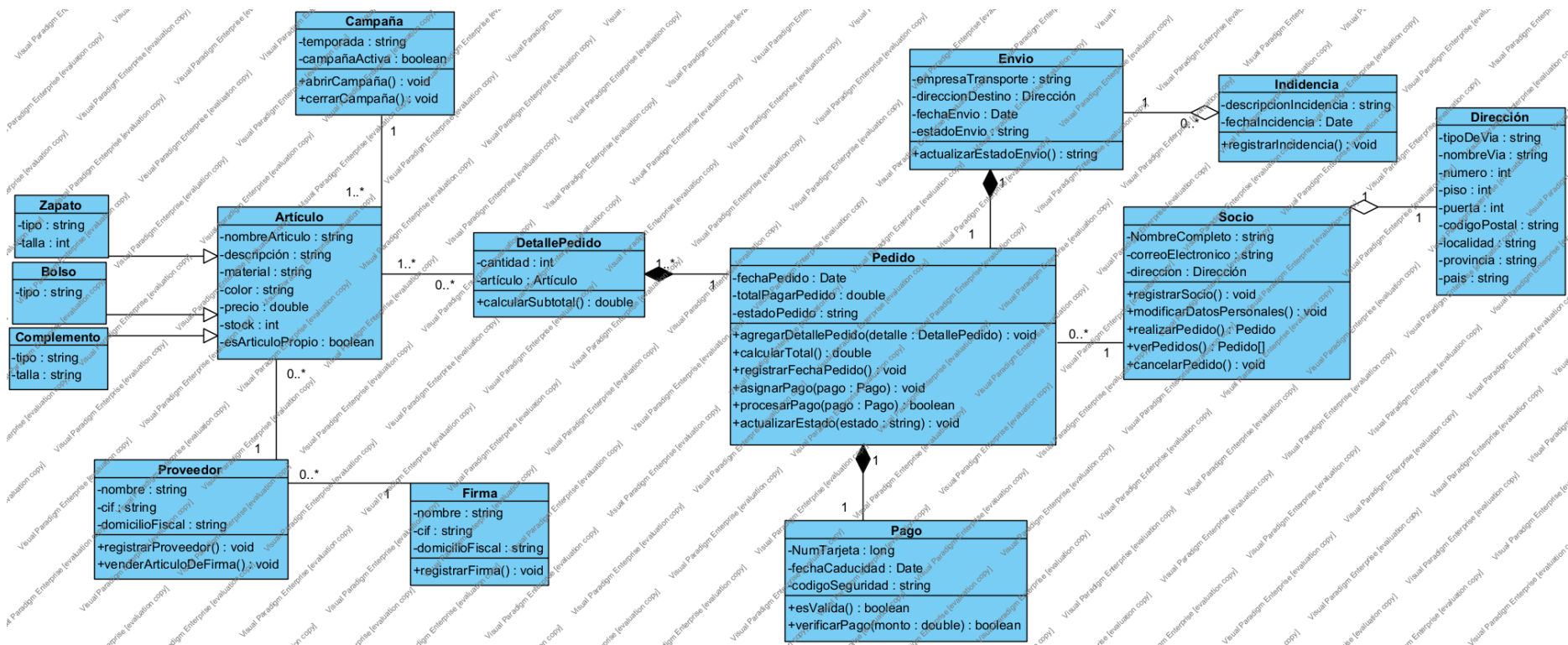
Una vez hecha la primera aproximación al diagrama, lo he evaluado y he hecho diversas mejoras:

Clase	Relaciones	Revisión/Optimización
Tienda Online	No tiene, ya que la funcionalidad de la tienda se distribuye en otras clases como Socio, Pedido.	Eliminada, porque no es necesaria como clase independiente. Sus responsabilidades ya están cubiertas por otras clases.
Usuario	No tiene, ya que la clase Socio cubre sus funcionalidades.	Eliminada, ya que Socio ya gestiona los datos de los usuarios y permite realizar compras.
Artículo	Herencia con Zapato, Bolso, Complemento. Asociación con DetallePedido, Campaña, Proveedor, Firma.	Revisando los atributos decido añadir un atributo que me diga si es un artículo propio o un artículo de una firma: <ul style="list-style-type: none"> • Añado el atributo esArticuloPropio como un boolean el cual Indica si el artículo es propio de la tienda o de una firma (vendido a través de un proveedor)
Campaña	Asociación con Artículo, Administrativo.	Decido añadir un atributo que indique si la campaña está activa o no. <ul style="list-style-type: none"> • Añado el atributo campañaActiva: boolean
Administrativo	Asociación con Campaña.	Elimino la clase Administrativo y paso sus métodos a Campaña
Proveedor	Asociación con Artículo.	Bien definida
Firma	Asociación con Artículo.	Cambio la relación Firma-Artículo por la relación de asociación Firma-Proveedor, debido a que he añadido el atributo esArticuloPropio a Artículo. La multiplicidad en Firma es 1 (porque una firma puede ser vendida por solo un Proveedor) y en Proveedor es 0..* (ya que un proveedor puede vender artículos de una o varias firmas).
Socio	Asociación con Pedido.	Revisados los atributos realizo el siguiente cambio: <ul style="list-style-type: none"> • Creo una clase Dirección • Creo un relación de agregación entre Socio(1)◊——(1)Dirección

		<p>Revisados los métodos realizo los siguientes cambios:</p> <ul style="list-style-type: none"> • Actualizo el retorno de realizarPedido() a Pedido en lugar de void. • Actualizo el retorno de verPedidos() a Pedido[] en lugar de void.
Pedido	Asociación con Socio, DetallePedido, TarjetaBancaria, Incidencia. Composición con DetallePedido, TarjetaBancaria, Incidencia.	<p>Elimino la clase EntidadBancaria y modifiko los métodos de Pedido</p> <p>Elimino la clase EmpleadoAlmacen y la clase Almacen.</p> <p>Añado un atributo a Pedido que indica el estadoPedido</p> <p>Añado el método actualizarEstado(estado:string):void para actualizar el estado del Pedido</p> <p>Elimino la clase EmpresaTransporte y la clase Ruta y creo una nueva clase llamada Envío, encargada de gestionar el envio del Pedido</p>
Detalle de Pedido	Asociación con Artículo. Composición con Pedido.	Adecuada.
Tarjeta Bancaria	Asociación con Pedido.	<p>Cambio el nombre de la clase Tarjeta Bancaria a Pago.</p> <p>Añado los traibutos fechaCaducidad y codigoSeguridad</p> <p>Añado el método verificarPago que estaba en EntidadFinanciera</p>
Entidad Financiera	Asociación con TarjetaBancaria.	Elimino la clase EntidadFinanciera del diagrama.
Almacén	Asociación con Pedido.	Elimino la clase Almacén
Empleado de Almacén	Asociación con Pedido.	Elimino la clase EmpleadoAlmacen
Empresa de Transporte	Asociación con Ruta.	Elimino la clase EmpresaTransporte y creo la clase Envio

Ruta	Asociación con Pedido.	Elimino la clase Ruta y creo la clase Envio
Incidencia	Asociación con Pedido.	Al crear la clase Envio, cambio la relación de Incidencia-Pedido por la relación Indicencia-Envio

A continuación se muestra el diagrama con todos los cambios:



Hechos todos los cambios la estructura del diagrama es la siguiente:

Artículos y sus tipos (Zapatos, Bolsos, Complementos)

La clase Artículo tiene los siguientes atributos:

- nombreArticulo: string
- descripción: string
- material: string
- color: string
- precio: double
- stock: int
- esArticuloPropio: boolean (para distinguir si es un artículo de la tienda o de una firma externa)

Subclases especializadas (Zapato, Bolso, Complemento)

- Zapato tiene talla y tipo
- Bolso tiene tipo
- Complemento tiene talla y tipo

Organización y venta (Campañas, Proveedores y Firmas)

Clase Campaña: Relacionada con Artículo:

- temporada: string (Primavera/Verano, Otoño/Invierno)
- campañaActiva: boolean
- Métodos: abrirCampaña(), cerrarCampaña()
- Relación 1..* con Artículo (una campaña debe tener al menos un artículo)

Clases Proveedor y Firma:

- nombre: string
- cif: string
- domicilioFiscal: string
- Métodos: registrarProveedor(), registrarFirma()

Gestión de Usuarios y Pedidos (Socio y Pedido)

Clase Socio:

- nombreCompleto: string
- correoElectronico: string
- dirección: Direccion
- Métodos: registrarSocio(), modificarDatosPersonales(), realizarPedido(), verPedidos(), cancelarPedido()

Clase Pedido:

- fechaPedido: Date
- totalPagarPedido: double
- estadoPedido: string
- Métodos: agregarDetallePedido(), calcularTotal(), registrarFechaPedido(), asignarPago(), procesarPago(), actualizarEstado()

Clase DetallePedido:

- cantidad: int
- artículo: Artículo
- Método: calcularSubtotal()
- Relación con Pedido: (1 Pedido puede tener múltiples DetallesPedido)

Gestión de Envíos

Clase Envío:

- empresaTransporte: string
- direccionDestino: Direccion
- fechaEnvio: Date
- estadoEnvio: string
- Método: actualizarEstadoEnvio()

Clase Dirección:

- tipoDeVia, nombreVia, número, piso, puerta, códigoPostal, localidad, provincia, país

Clase Incidencia:

- descripcionIncidencia: string
- fechaIncidencia: Date
- registrarIncidencia()
- Relación con Envio como agregación (0..* incidencias por envío)

6. Implementación de Getters, Setters y Constructores

Enunciado: Añade constructores, getters y setters en todas las clases donde sea necesario.

Justificación:

- Explica cómo estos métodos permiten la encapsulación y acceso seguro a los datos.
- Pantallazo

Justificación: Encapsulación y Acceso Seguro a los Datos

En la implementación del sistema de ventas en línea para *El Tacón de Oro*, he añadido **constructores, getters y setters** en las clases necesarias. Esto nos permite aplicar el principio de **encapsulación**, que es fundamental en la programación orientada a objetos (POO). Estos métodos contribuyen a la seguridad y organización de los datos por las siguientes razones:

- **Encapsulación: Protección y Control de los Datos**

La encapsulación se basa en **restringir el acceso directo a los atributos** de una clase y proporcionar métodos controlados para su manipulación.

Beneficio: Se evita que el código externo altere los datos de manera inesperada, reduciendo errores y manteniendo la integridad de los objetos.

Para lograrlo:

1. **Los atributos son private:** No pueden ser modificados directamente desde fuera de la clase.

2. **Se crean métodos getter y setter:** Permiten leer y modificar los atributos de manera controlada.

- **Constructores: Creación de Objetos de Forma Segura**

Los constructores permiten inicializar objetos de manera correcta.

Beneficio: Se evita la creación de objetos en estados inválidos o incompletos.

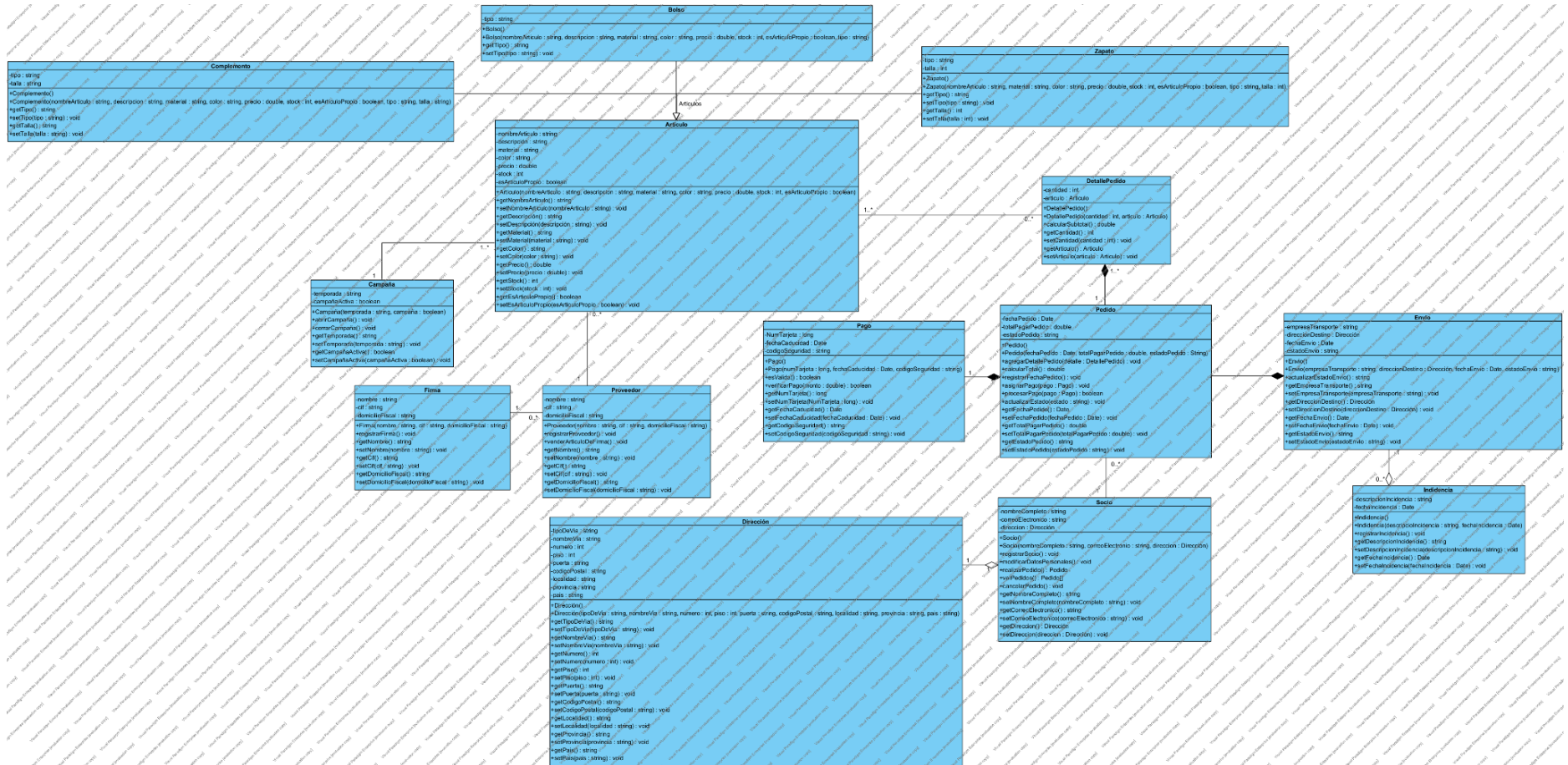
- **Constructor vacío:** Permite crear objetos sin valores iniciales.
- **Constructor con parámetros:** Garantiza que los objetos tengan datos desde su creación.

- **Acceso Seguro a los Datos: Validación y Control**

Los setters permiten agregar **validaciones antes de modificar un atributo**.

Beneficio: Se evita asignar valores incorrectos o incoherentes.

A continuación se muestra el Diagrama de Clases con sus constructores, getters y setters:



7. Refinamiento del Diagrama UML

Enunciado: Revisa el diagrama UML y mejora su estructura si es necesario.

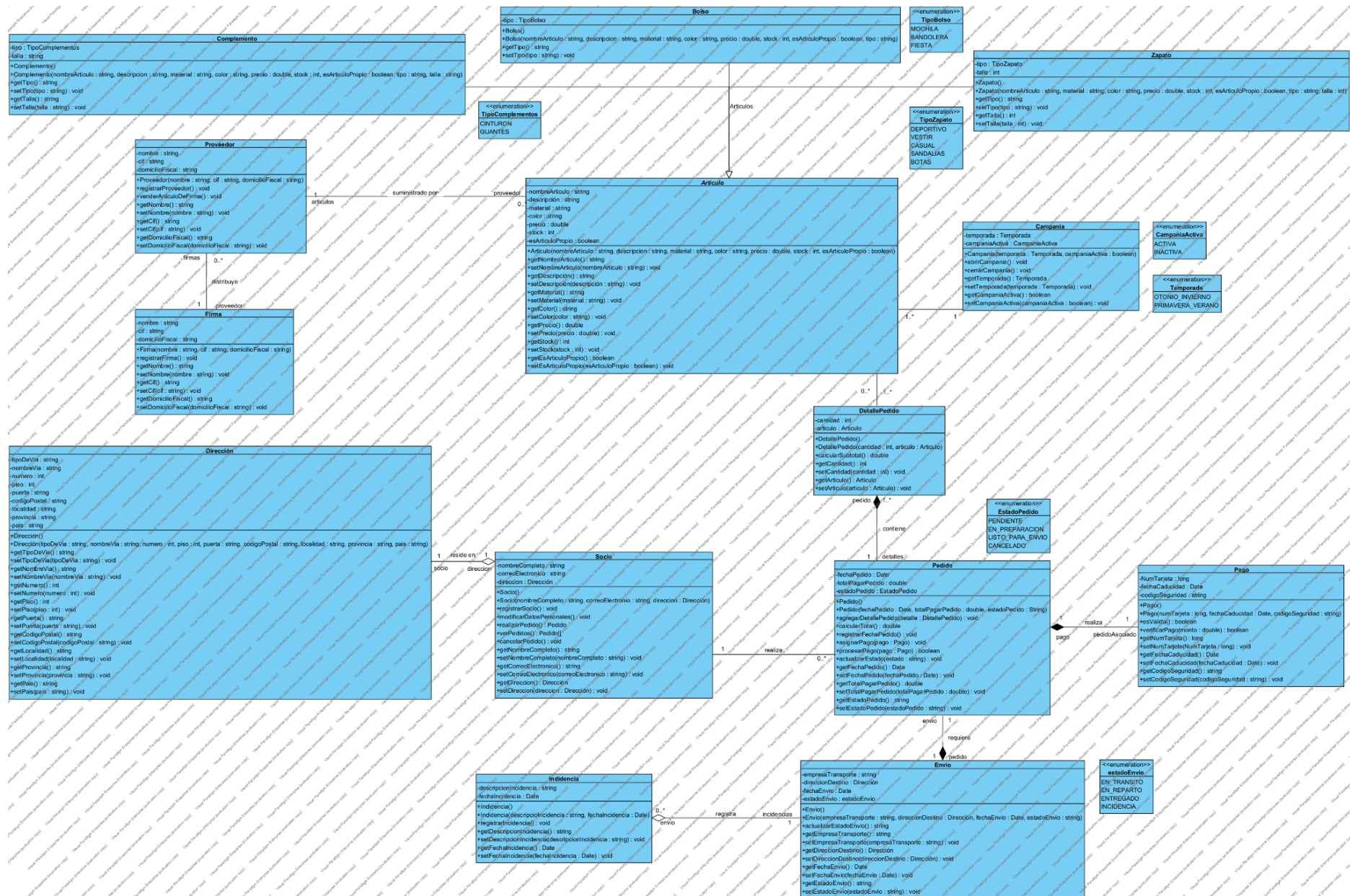
Justificación:

- Explica qué cambios has hecho y por qué mejoran el diseño.
- Pantallazo.

Mejoras realizadas:

- **He añadido enumerados:** La adición de enumerados mejora la estructura del diagrama al hacerlo más claro, consistente, mantenible y preciso.
- **He añadido anotaciones de roles:** ayuda a la comprensión de las relaciones entre las clases, haciendo que el diagrama sea más completo y fácil de entender.
- **He añadido nombres de relaciones:** ayuda a la comprensión de las relaciones entre las clases.

A continuación se muestra el Diagrama de clases con las mejoras realizadas:



8. Documentación del Proyecto

Enunciado: Redacta un documento explicando cómo diseñaste el sistema.

Justificación: Describe el proceso paso a paso y las decisiones tomadas.

Por último se debe rellenar la documentación de cada clase, atributo y método con una descripción de los mismos que será necesaria para la generación de informes posterior. A continuación se listan las clases con su documentación:

Clase/Objeto	Descripción
Socio	Representa a un usuario registrado en el sistema que puede realizar pedidos. Un socio debe proporcionar su nombre, correo electrónico y dirección. Puede modificar sus datos personales, ver sus pedidos y cancelarlos antes del envío.
Dirección	Contiene la información de la dirección de un socio, incluyendo calle, número, piso, código postal, localidad, provincia y país. Se usa tanto en el registro del socio como en los envíos.
Pedido	Representa un pedido realizado por un socio. Un pedido está compuesto por detalles de pedido, tiene una fecha de realización, un estado y un total a pagar. Se puede agregar artículos, asignar pagos y actualizar su estado.
DetallePedido	Define cada artículo dentro de un pedido, almacenando el artículo y la cantidad solicitada. Permite calcular el subtotal de cada detalle.
Pago	Representa el pago de un pedido realizado mediante tarjeta bancaria. Se almacena el número de tarjeta, fecha de caducidad y código de seguridad. Se valida y procesa el pago a través de métodos específicos.
Envío	Se encarga de la entrega de los pedidos a los socios. Almacena la empresa de transporte, la dirección de destino, la fecha de envío y su estado. Permite actualizar el estado del envío.
Incidencia	Registra cualquier problema ocurrido durante el envío. Contiene la descripción de la incidencia y la fecha en que ocurrió.
Artículo	Representa un producto disponible en la tienda. Contiene información como el nombre, descripción, material, color, precio y stock. Puede ser un artículo propio de la tienda o de una firma externa.
Zapato	Subclase de Artículo, representa un zapato. Contiene información adicional como el tipo de zapato y la talla.

Bolso	Subclase de Artículo, representa un bolso. Incluye un atributo adicional que define su tipo (mochila, bandolera, fiesta, etc.).
Complemento	Subclase de Artículo, representa accesorios como cinturones y guantes. Contiene información adicional sobre su tipo y talla.
Campaña	Representa las campañas de venta organizadas por la tienda. Cada campaña pertenece a una temporada específica y puede estar activa o inactiva. Solo los artículos en campañas activas pueden ser vendidos.
Proveedor	Representa un proveedor de artículos de firma. Un proveedor puede vender artículos a la tienda. Se almacena su nombre, CIF y domicilio fiscal.
Firma	Representa una marca de artículos vendidos en la tienda a través de un Proveedor. Almacena su nombre, CIF y domicilio fiscal.

9. Integración del Proyecto VP-UML en NetBeans

Enunciado: Importa el proyecto UML en NetBeans.

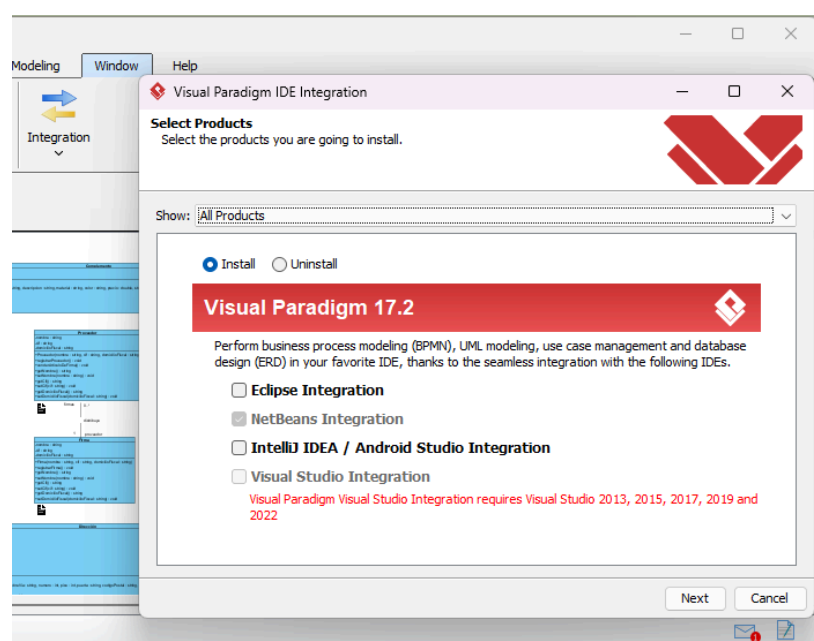
Justificación:

- Explica qué pasos seguiste para la integración y si encontraste dificultades.
- Pantallazo de la importación
-

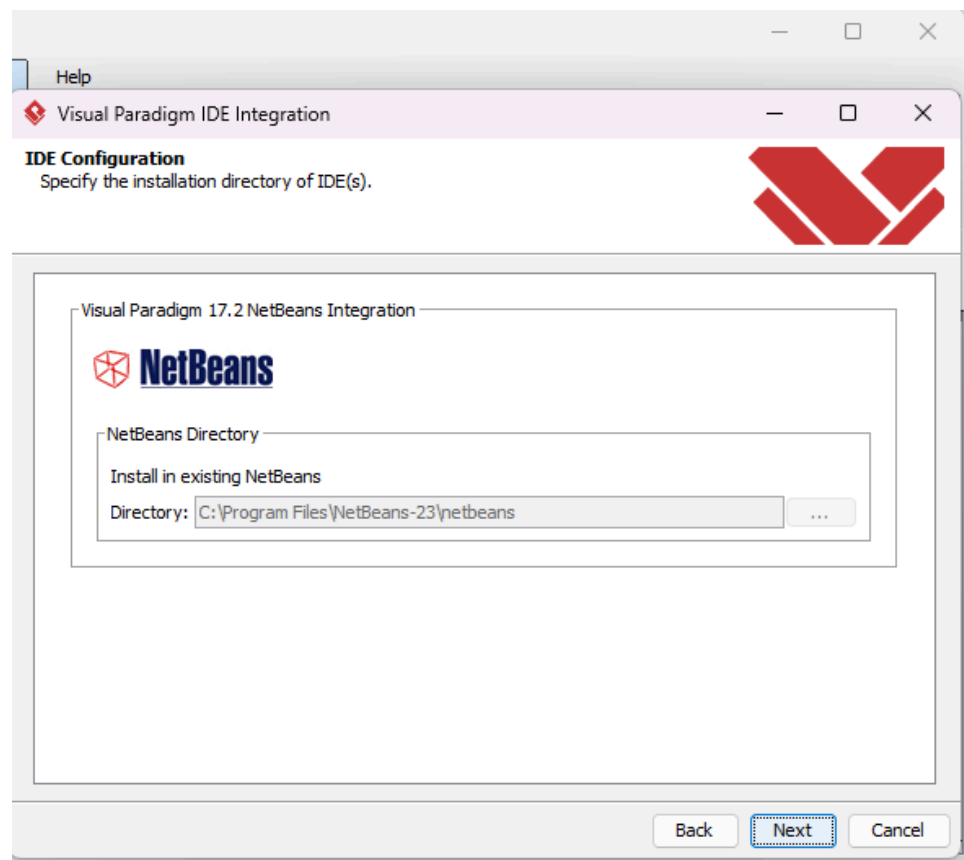
Integración del Proyecto VP-UML en NetBeans

Integrar VP a NetBeans

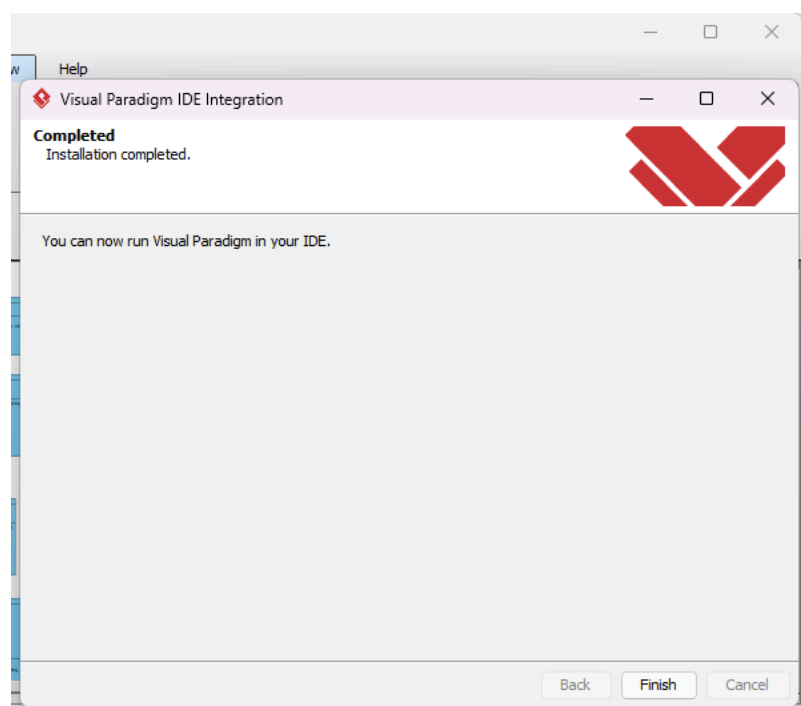
Seleccionar la pestaña Window / Integration y en la ventana que aparece seleccionar NetBeans



Seleccionar la ruta donde está NetBeans en el equipo y seleccionar Next

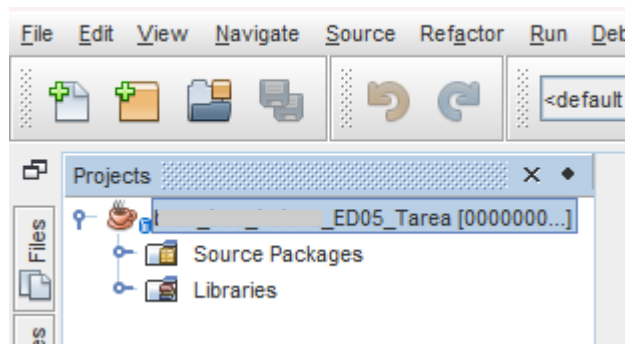


Instalación completada

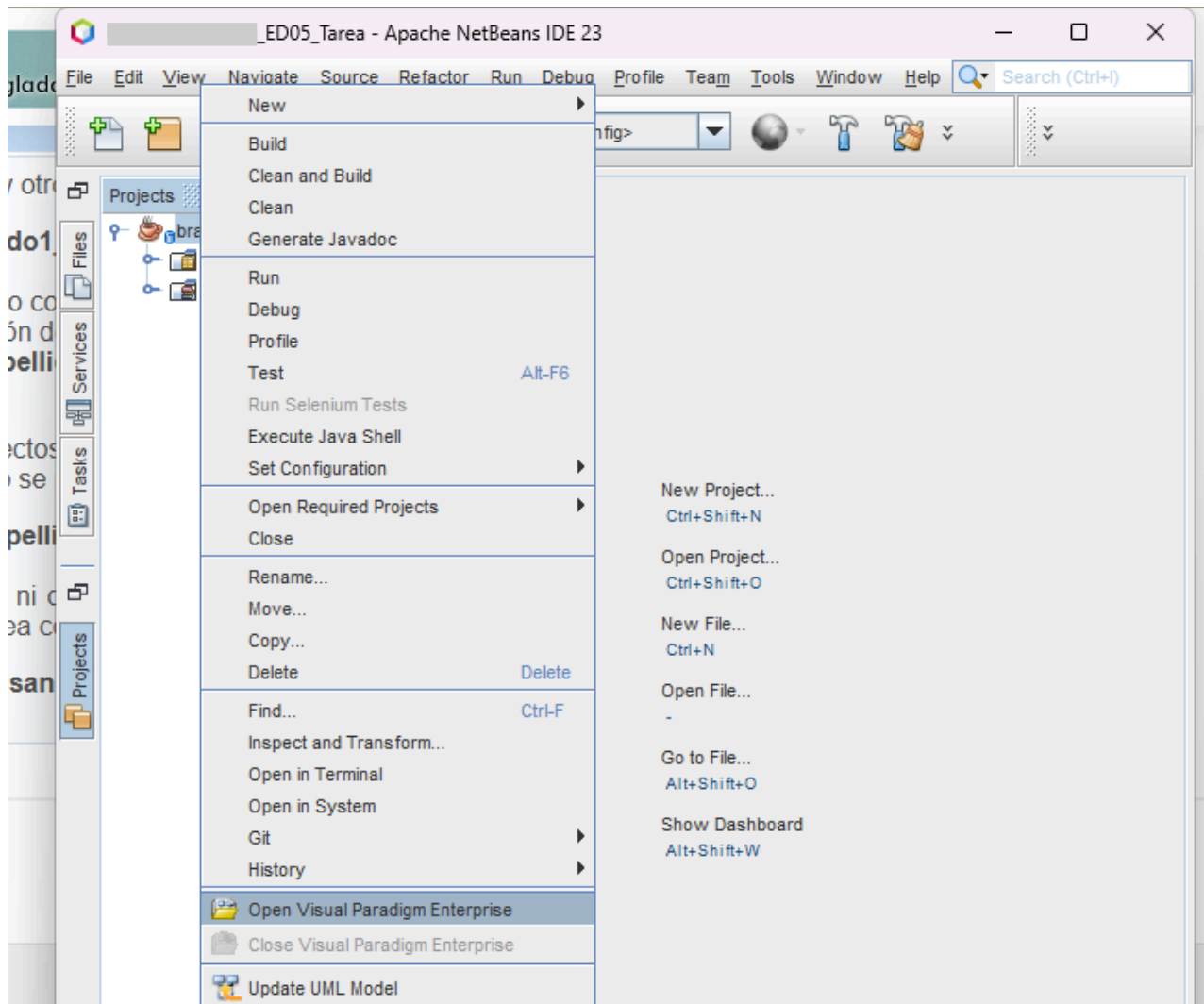


Abrir el Diagrama de clases en NetBeans

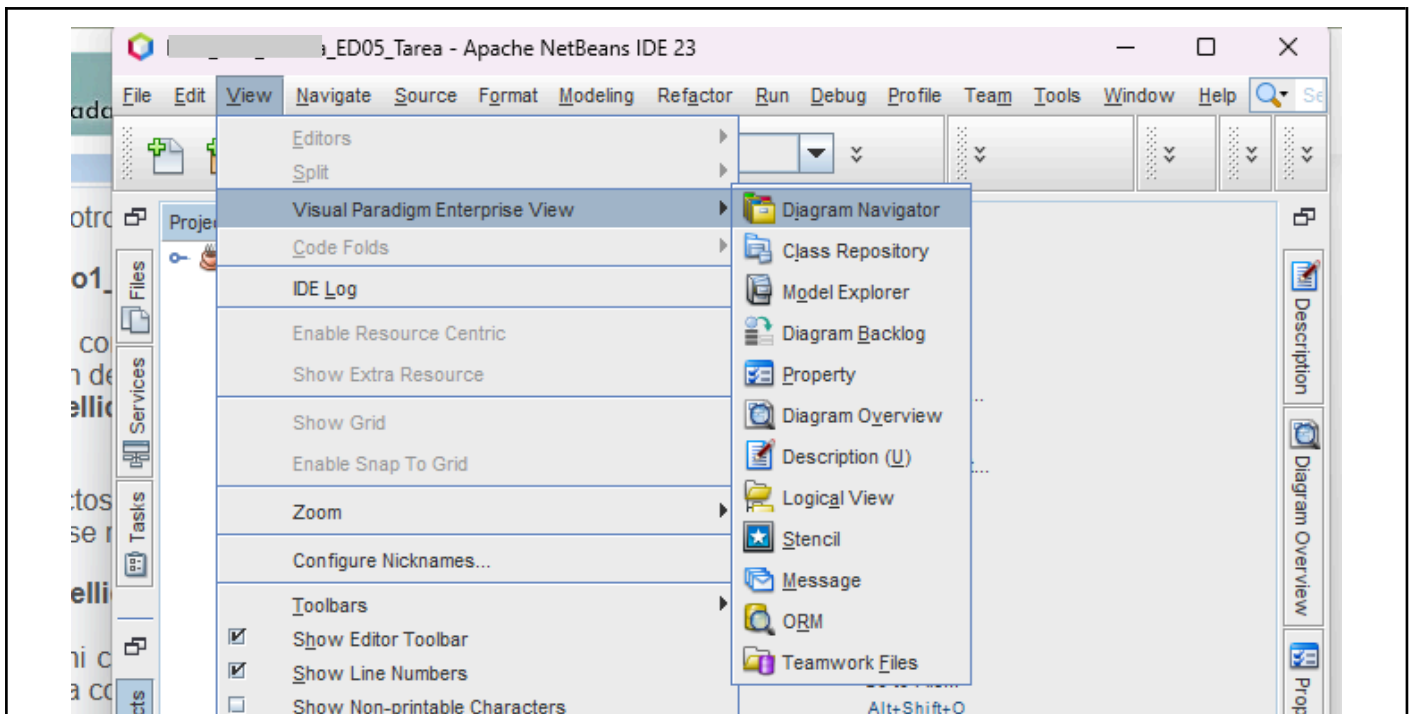
Abrir NetBeans y crear un nuevo proyecto



Clic derecho encima del proyecto y seleccionar Open Visual ParadigmEnterprise



Seleccionar View / Visual Paradigm Enterprise View / Diagram Navigator



En el panel de diagramas, dentro de UML Diagrams / Class Diagram aparece el Diagrama de clases

