

**Subrayar los sustantivos (sin repetición):**

Los usuarios del sistema navegan por la web para ver los artículos, zapatos, bolsos y complementos que se venden en la tienda. De los artículos nos interesa su nombre, descripción, material, color, precio y stock. De los zapatos nos interesa su número y el tipo. De los bolsos nos interesa su tipo (bandolera, mochila, fiesta). De los complementos (cinturones y guantes) su talla.

Los artículos se organizan por campañas para cada temporada (primavera/verano y otoño/invierno) de cada año.

Los artículos son de fabricación propia, pero, opcionalmente, pueden venderse artículos de otras firmas. De las firmas nos interesa saber su nombre, CIF y domicilio fiscal. La venta de artículos de firma se realiza a través de proveedores, de forma que un proveedor puede llevar varios artículos de diferentes firmas, y una firma puede ser suministrada por más de un proveedor. Los artículos pertenecen a una firma solamente. De los proveedores debemos conocer su nombre, CIF, y domicilio fiscal.

Los usuarios pueden registrarse en el sitio web para hacerse socios. Cuando un usuario se hace socio debe proporcionar los siguiente datos: nombre completo, correo electrónico y dirección.

Los socios pueden hacer pedidos de los artículos. Un pedido está formado por un conjunto de detalles de pedido que son parejas formadas por artículo y la cantidad. De los pedidos interesa saber la fecha en la que se realizó y cuanto debe pagar el socio en total. El pago se hace a través tarjeta bancaria, cuando se va a pagar una entidad bancaria comprueba la validez de la tarjeta. De la tarjeta interesa conocer el número.

Las campañas son gestionadas por el administrativo de la tienda que se encargará de dar de baja la campaña anterior y dar de alta la nueva siempre que no haya ningún pedido pendiente de cumplimentar.

Existe un empleado de almacén que revisa los pedidos a diario y los cumplimenta. Esto consiste en recopilar los artículos que aparecen en el pedido y empaquetarlos. Cuando el paquete está listo se pasa al almacén a la espera de ser repartido. Del reparto se encarga una empresa de transportes que tiene varias rutras preestablecidas. Según el destino del paquete (la dirección del socio) se asigna a una u otra ruta. De la empresa de transportes se debe conocer su nombre, CIF y domicilio fiscal. Las rutas tienen un área de influencia que determina los destinos, y unos días de reparto asignados. Se debe conocer la fecha en la que se reparte el pedido. Si se produce alguna incidencia durante el reparto de algún pedido se almacena la fecha en la que se ha producido y una descripción.

Los socios pueden visualizar sus pedidos y cancelarlos siempre y cuando no hayan sido cumplimentados por el empleado de almacén. Así mismo puede modificar sus datos personales.

Tabla de sustantivos	
Clase/Objeto potencial	Categoría
Usuario	Entidad externa o rol
artículos	Entidad externa
zapatos	Entidad externa
bolsos	Entidad externa
complementos	Entidad externa
tienda	Unidad organizacional
nombre	atributo
descripción	atributo
material	atributo
color	atributo
precio	atributo
stock	atributo
numero	atributo
tipo	atributo
talla	atributo
campana	suceso
temporada	atributo
firma	cosa
cif	atributo
domicilioFiscal	atributo
venta	suceso
proveedor	Entidad externa o rol
socio	Entidad externa o rol
correoElectronico	atributo
direccion	atributo
pedido	suceso
detalle	estructura
articulo	atributo
cantidad	atributo
fecha	atributo
total	atributo
pago	suceso
tarjetaBancaria	cosa
entidad bancaria	Unidad organizacional
numero	atributo
administrativo	Entidad externa o rol
empleadoAlmacén	Entidad externa o rol
paquete	cosa
almacén	Unidad organizacional
reparto	suceso
empresaTransporte	Entidad externa
ruta	cosa
areaInfluencia	atributo
destino	atributo
diasDeReparto	atributo
fecha	atributo
incidencia	suceso

**PAUTAS A SEGUIR que se desprende del texto (refinamiento).**

Desglosando el enunciado descubrimos que tanto los zapatos, bolsos y complementos son artículos y que tendrán los atributos de éste más los suyos propios. En los artículos deberíamos almacenar la temporada a la que pertenecen.

Las campañas las gestiona el administrativo de la tienda que dará de alta la nueva temporada siempre que pueda dar de baja la anterior (crearemos un método que compruebe si hay pedidos pendientes y otro para dar de alta la nueva temporada)

Los artículos se muestran en tienda y los pueden ver los usuarios (aunque no sean socios)

Las firmas son suministradas por proveedores y además éstos pueden vender artículos.

Cuando un usuario se hace socio puede modificar sus datos, hacer, visualizar o cancelar pedidos.

El pedido está formado por detalles (guardamos en pedido el socio que lo realiza, el total de la compra y la fecha de la misma, y en detalles cada artículo adquirido, la cantidad y el precio del mismo). Necesitaremos en pedido un método que nos calcule el total del par cantidad por precio.

El pago se hace a través de tarjeta bancaria que debemos comprobar por lo que crearemos una clase que denominaremos *Pago* y que recogerá la información del número de tarjeta, la fecha y el socio al que pertenece y dentro de ella pondremos un método para comprobar la validez de la tarjeta (se indica que es una entidad externa la que lo comprobará, pero como carecemos de más información, de momento lo solucionamos con este método).

El empleado de almacén es el encargado de empaquetar los pedidos y enviarlos al almacén.

En almacén grabaremos el número del paquete y el destino el cual averiguaremos mediante el método `asignarDestino` que recibirá el parámetro de la dirección del socio.

La ruta viene determinada por la dirección del cliente, y tienen un día de reparto dado, por lo que para esta clase tendremos que crear un método denominado `destino` que recibirá como parámetro la dirección del socio, y otro denominado `diaReparto` que recibirá como parámetro la fecha del paquete.

La empresa de transporte es la que distribuirá los paquetes por las distintas rutas según las condiciones comprobadas en la clase *Ruta*.

Si se produce alguna incidencia se almacenará la fecha y descripción de la misma.

---

Ahora vamos a crear la relación de criterios a aplicar según la clase u objeto potencial, basándonos en:

1. ¿La información es necesaria para que el sistema funcione?
2. ¿El objeto posee un conjunto de atributos?
3. ¿El objeto posee un conjunto de operaciones identificables?
4. ¿Es una entidad externa que consume o produce información esencial?

Veamos cómo quedaría la tabla resultante de la aplicación de las anteriores premisas:

Tabla de elección de sustantivos como objetos o clases del sistema	
Clase/objeto potencial	Criterios aplicables
Usuario	2,3,4
Articulo	1,2,3
Zapatos	1,2,3
Bolsos	1,2,3
Complementos	1,2,3
Tienda	1,3
Campaña	1,2,3
Firma	2,3,4
Venta	1
Proveedor	2,3,4
Socio	2,3,4
Pedido	1,2,3
Detalle	1,2,3
Pago	1,2,3
TarjetaBancaria	2,3
EntidadBancaria	
Administrativo	2,3,4
EmpleadoAlmacen	1,2,4
Paquete	1,2
Almacen	1,2,3
Reparto	1
EmpresaTransporte	1,2,4
Ruta	1,2,3
Incidencia	1,2

Según el enunciado las clases/objetos potenciales pueden poseer una serie de atributos que definan a dicho objeto en el contexto del sistema, por lo que identificamos a cada clase/objeto con los atributos (sustantivos de tipo atributo en la primera tabla).

Tabla de relación de las clases u objetos con sus atributos	
Clase/Objeto potencial	Atributos
Usuario	nombre,email
Articulo	nombre,descripción,material,color,precio,stock,tipo,temporada
Zapatos	numero,tipo
Bolsos	tipo
Complementos	talla
Tienda	
Campaña	temporada
Firma	nombre,cif,domicilioFiscal
Proveedor	nombre,cif,domicilioFiscal
Socio	nombre,correoElectronico,dirección
Pedido	fecha,total,tarjeta
Detalle	pedido,articulo,cantidad,fecha
Pago	tarjetaBancaria,fecha,socio
Administrativo	nombre
EmpleadoAlmacen	nombre
Almacén	paqueteNum,destino
EmpresaTransporte	nombre,cif,domicilioFiscal

<b>Ruta</b>	areaInfluencia,diasDeReparto,fecha
<b>Incidencia</b>	fecha,descripcion

Según los verbos del enunciado, y las actividades en general, que describan el comportamiento de los objetos o que modifiquen su estado, podemos obtener la siguiente tabla:

<b>Tabla de clases u objetos del sistema con sus posibles métodos</b>	
<b>Clase/objeto potencial</b>	<b>Métodos</b>
Usuario	verArticulos(Articulo:articulo):void
Tienda	mostrarComplementos():void mostrarZapatos():void mostrarBolsos():void
Administrativo	compruebaPedidoPendiente(fecha:date):boolean altaNuevaCampaña(temporada:string):boolean
Pedido	hallarTotal(cantida:float,precio:float):float
Socio	verPedido(fecha:date):void cancelarPedido(fecha:date):boolean
Ruta	destino(direSocio:string):string diaReparto(fecha:date):int
Pago	comprobarTarjeta(numero:long):void
Almacen	asignarDestino(direSocio:string):string

Ahora vamos a ver, según el enunciado, qué tipo de relaciones pueden existir entre las diferentes clases u objetos, para lo que hemos de fijarnos en la acción que se realiza entre dichos objetos/clases.

<b>Tabla de relaciones</b>		
<b>Clase/Objeto</b>	<b>Relación</b>	<b>Clase/Objeto</b>
<b>Usuario</b>	visualiza productos	<b>Tienda</b>
<b>Usuario</b>	se registra	<b>Socio</b>
<b>Socio</b>	modifica	<b>Socio</b>
<b>Socio</b>	hace	<b>Pedido</b>
<b>Administrativo</b>	gestiona	<b>Campaña</b>
<b>Campaña</b>	organizada por	<b>Tienda</b>
<b>Tienda</b>	vende	<b>Firma</b>
<b>Proveedor</b>	suministra	<b>Firma</b>
<b>Proveedor</b>	vende	<b>Pedido</b>
<b>Pedido</b>	formado por	<b>Detalle</b>
<b>Detalle</b>	se abona	<b>Pago</b>
<b>Pago</b>	empaqueta	<b>EmpleadoAlmacen</b>
<b>EmpleadoAlmacen</b>	envía	<b>Almacen</b>
<b>Almacen</b>	asigna	<b>Ruta</b>
<b>Ruta</b>	distribuye	<b>EmpresaTransporte</b>
<b>EmpresaTransporte</b>	se produce	<b>Incidencia</b>

Si nuestras clases/objetos poseen una serie de atributos que se puedan/necesiten inicializar, ver o crear al inicio, tendremos que definir sus correspondientes getters, setters y constructores respectivamente.

Tabla de Getters/Setters y constructores	
Clase/Objeto	Constructores
<b>Articulos</b>	Articulos() Articulos(nombre:string,descripcion:string,material:string,color:string,precio:float,stock:int,temporada:string)
<b>Complementos</b>	Complementos(articulo:Articulo,talla:float)
<b>Zapatos</b>	Zapatos(articulo:Articulo,numero:int,tipo:string)
<b>Bolsos</b>	Bolsos(articulo:Articulo,tipo:string)
<b>Usuario</b>	Usuario() Usuario(nombre:string,email:string)
<b>Firma</b>	Firma(nombre:string,cif:string,domicilioFiscal:string)
<b>Administrativo</b>	Administrativo(nombre:string)
<b>Proveedor</b>	Proveedor(nombre:string,cif:string,domicilioFiscal:string)
<b>Socio</b>	Socio() Socio(nombre:string,correoElectronico:string,dirección:string)
<b>Pedido</b>	Pedido() Pedido(fecha:date,total:float,socio:Socio)
<b>Pago</b>	Pago() Pago(tarjeta:long,fecha:date,socio:Socio)
<b>EmpleadoAlmacen</b>	EmpleadoAlmacen(nombre:string)
<b>Almacen</b>	Almacen()
<b>Ruta</b>	Ruta()
<b>EmpresaTransporte</b>	EmpresaTransporte()
<b>Incidencia</b>	Incidencia()

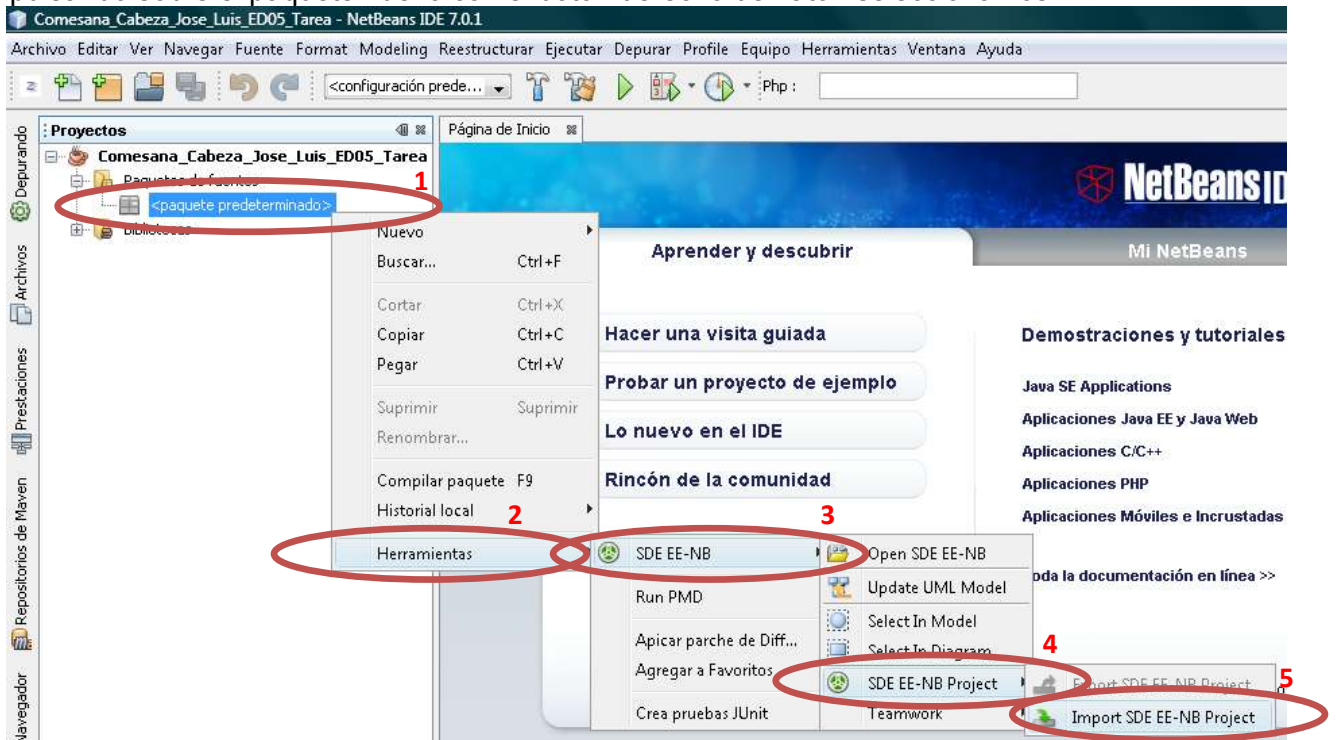
Clase/Objeto	Setters/Getters
<b>Articulos</b>	setNombre(nombre:string):void setDescripcion(descripción:string):void setMaterial(material:string):void setColor(color:string):void setPrecio(precio:float):void setStock(stock:int):void setTemporada(temporada:string):void
<b>Complementos</b>	getTalla():float getArticulo():Articulos setTalla(talla:float):void setArticulo(articulo:Articulos):void
<b>Zapatos</b>	getNumero():int getTipo():string getArticulo():Articulos setNumero(numero:int):void setTipo(tipo:string):void setArticulo(articulo:Articulos):void
<b>Bolsos</b>	getTipo():string getArticulo():Articulos setTipo(tipo:string):void setArticulo(articulo:Articulos):void
<b>Usuario</b>	getNombre():string getEmail():string setNombre(nombre:string):void setEmail(email:string):void
<b>Firma</b>	getNombre():string getCif():string

	getDomicilioFiscal():string setNombre(nombre:string):void setCif(cif:string):void setDomicilioFiscal(domicilioFiscal:string):void
<b>Campaña</b>	getTemporada():string setTemporada(temporada:string):void
<b>Administrativo</b>	getNombre():string setNombre(nombre:string):void
<b>Proveedor</b>	getNombre():string getCif():string getDomicilioFiscal():string setNombre(nombre:string):void setCif(cif:string):void setDomicilioFiscal(domicilioFiscal:string):void
<b>Socio</b>	getNombre():void getCorreoElectronico():string getDireccion():string setNombre(nombre:string):void setCorreoElectronico(correoElectronico:string):void setDireccion(dirección:string):void
<b>Pedido</b>	getFecha():date getTotal():float getSocio():Socio setFecha(fecha:date):void setTotal(total:float):void
<b>Detalle</b>	getArticulo():articulo getCantidad():float getPrecio():float setArticulo(articulo:Articulos):void setCantidad(cantidad:float):void setPrecio(precio:float):void
<b>Pago</b>	getTarjeta():long setTarjeta(tarjeta:long):void
<b>EmpleadoAlmacen</b>	getNombre():string setNombre(nombre:string):void
<b>Ruta</b>	getArealInfluencia():string getDiasReparto():string getFecha():date setArealInfluencia(arealInfluencia:string):void setDiasReparto(diasReparto:string):void setFecha(fecha:date):void
<b>EmpresaTransporte</b>	getNombre():string getCif():string getDomicilioFiscal():string setNombre(nombre:string):void setCif(cif:string):void setDomicilioFiscal(domicilioFiscal:string):void
<b>Incidencia</b>	getFecha():date getDescripcion():string setFecha(fecha:date):void setDescripcion(descripción:string):void

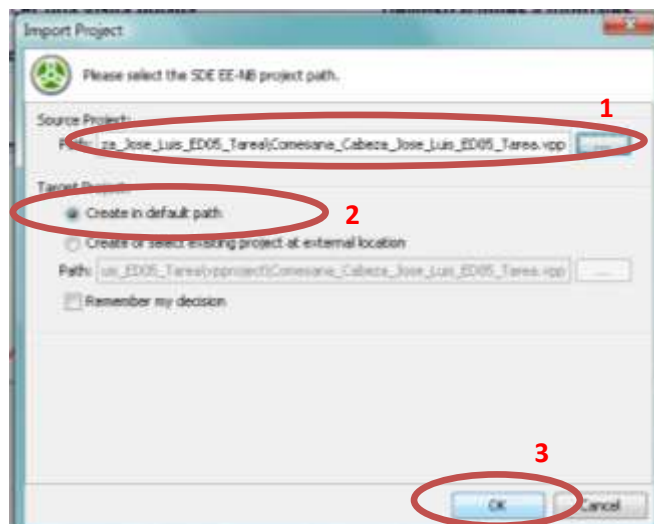
Plasmando todos los requisitos obtenidos en los puntos anteriores, tendremos el siguiente diagrama (que se incluye en un documento anexo denominado Comesana\_Cabeza\_Jose\_Luis\_ED05\_Tarea.vpp):



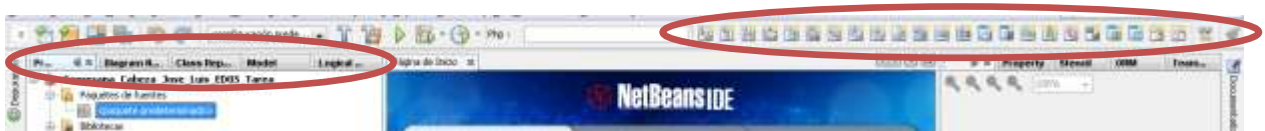
Ha llegado el momento de crear el proyecto en Java, para lo que abriremos NetBeans y crearemos un proyecto nuevo con el nombre Comesana\_Cabeza\_Jose\_Luis\_ED05\_Tarea y pulsando sobre el paquete nuevo con el botón derecho del ratón seleccionamos:



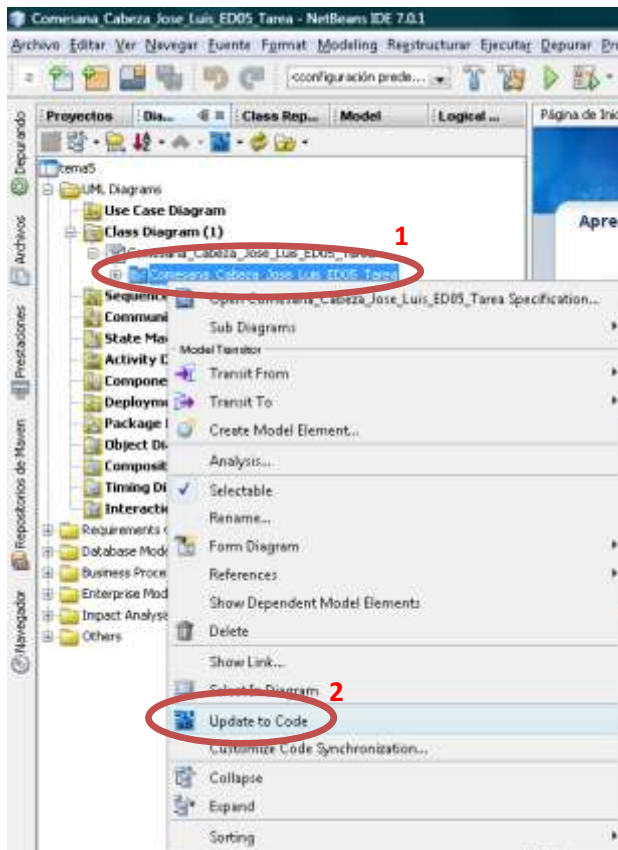
Indicamos dónde se encuentra el proyecto que hemos creado con Visual Paradigm dejando que utilice como destino la carpeta por defecto:



Tras ello nos aparece una serie de iconos de generación de diagramas en la parte superior derecha y un menú superior nuevo donde antes sólo estaba "Proyectos":



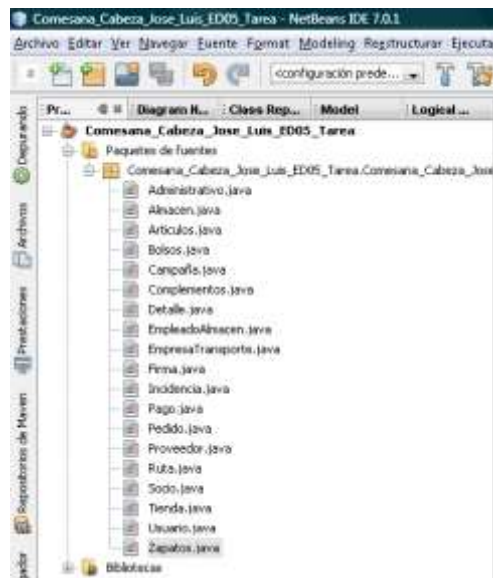
Seleccionamos “Navegador de diagramas” (segunda opción del nuevo menú superior). Abrimos los diagramas de clase y nos debe aparecer el que hemos creado con Visual Paradigm y dentro de éste, el paquete creado.



Pulsamos con el botón derecho sobre dicho paquete<sup>1</sup> y seleccionamos “Actualizar a código”<sup>2</sup> con lo que a partir de este momento tenemos nuestro proyecto Java creado, aunque plagado de errores porque la codificación se ha hecho en UTF8 y los tipos de datos no son del todo correctos, etc



Depuraremos el código hasta tenerlo libre de errores, quedando:



Cuando cerramos NetBeans nos preguntará si queremos grabar el diagrama, por lo que contestaremos que sí para que queden reflejados todos los cambios que hemos realizado.

### Explicación del desarrollo de la tarea

Para la realización de la presente tarea he usado **Visual Paradigm UML 8.3** con la que cree infinidad de diagramas de clase hasta que me decidí por el que presento, el cual viene en el directorio principal del ejercicio en su versión anterior a la generación del código con NetBeans, y en la carpeta **Comesana\_Cabeza\_Jose\_Luis\_ED05\_Tarea\vpproject** en su versión depurada tras la generación del código Java. Así mismo, en ese directorio se encuentra el informe final del proyecto generado en Word desde el mismo Visual Paradigm, con todas las clases comentadas y el proyecto generado con NetBeans en la carpeta **Comesana\_Cabeza\_Jose\_Luis\_ED05\_Tarea**.

Considero que VP es una muy buena herramienta CASE para el diseño de diagramas y generación automatizada de código, pero que necesita de muchas más horas que las que le podemos dedicar. Me ha costado bastante basándome en el temario y sobre todo en información obtenida por internet y libros descargados.

---