

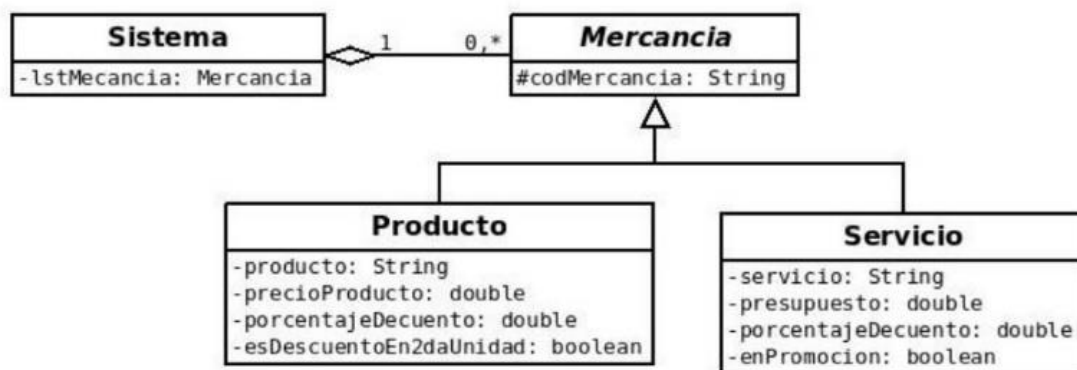
**IDE:** Eclipse

**Proyecto:** Crear una carpeta /tuApellidoNombre/tuNroDni (el nombre del proyecto es tu DNI)

\* **Entrega:** Aula Virtual subir la carpeta comprimida con tuApellidoNombre

### Sistema de Mercancia

**Nota:** El examen acredita puntos por la resolución del modelo y test de cada CU.



#### Capa modelo:

##### Clase Mercancia:

Superclase abstracta

##### Atributo:

String codMercancia //Es el identificador único que para que sea válido debe tener una longitud igual a 7, implementar excepción

##### Metodos:

~ calcularPrecioFinal() : double //método abstracto

##### Clase Producto:

Subclase de Mercancia

**Atributos:** String producto, double precioProducto, double porcentajeDescuento y boolean esDescuentoEn2daUnidad

##### Metodos:

+calcularPrecioFinal() : double //Si tiene descuento en la 2° unidad, para calcular el precio final aplica la mitad del porcentajeDescuento ya que tiene que comprar 2 artículos.

##### Clase Servicio:

Subclase de Mercancia

**Atributos:** String servicio, double presupuesto, double porcentajeDescuento, boolean enPromocion

##### Metodos:

+calcularPrecioFinal() : double //Si está en promoción, el precio final aplica porcentajeDescuento

##### Clase Sistema

**Atributo:** List<Mercancia> lstMercancia

##### Metodos:

- + traerMercancia (String codMercancia) : Mercancia
- + traerMercancia(boolean enOferta) : Mercancia //retorna una lista del tipo Mercancia al comparar el parámetro con si está en promoción si es un servicio o si tiene descuento en la 2°unidad si es un producto.

- + agregarProducto(String codMercancia, String producto, double precioProducto, double porcentajeDescuento, boolean esDescuentoEn2daUnidad) : boolean //implementar excepción si existe el objeto
- + agregarServicio(String codMercancia, String servicio, double presupuesto, double porcentajeDescuento, boolean enPromocion) : boolean //implementar excepción si existe el objeto

### **Capa test:**

**Nota:** Al comenzar cada test indicar el número a resolver ej: `System.out.println("1-3");` y luego la implementación del mismo.

### **Test.java**

1-1) Intentar crear el objeto Servicio: codMercancia=718392, servicio=Mantenimiento hidrolavadora, presupuesto=2700.0, porcentajeDescuento= 10.0, enPromocion=true

1-2) Crear e imprimir el objeto Servicio: codMercancia=7183921, servicio=Mantenimiento hidrolavadora, presupuesto=2700.0, porcentajeDescuento= 10.0, enPromocion=true

1-3) Intentar crear el objeto Producto: codMercancia=782951, producto=Diyuntor, precioProducto=4500.0, porcentajeDescuento= 25.0, esDescuentoEn2daUnidad=true

1-4) Crear e imprimir el objeto Producto: codMercancia=7829512, producto=Diyuntor, precioProducto=4500.0, porcentajeDescuento= 25.0, esDescuentoEn2daUnidad=true

2-1) Calcular e imprimir precio final del servicio:

Servicio [codMercancia=7183921, servicio=Mantenimiento hidrolavadora, presupuesto=2700.0, enPromocion=true]

2-2) Calcular e imprimir precio final del servicio:

Producto [codMercancia=7829512, producto=Diyuntor, precioProducto=4500.0, porcentajeDecuento=25.0, esDescuentoEn2daUnidad=true]

3-1) Agregar los siguientes objetos

Producto [codMercancia=3780173, producto=Lampara led 9W, precioProducto=330.0, porcentajeDecuento=10.0, esDescuentoEn2daUnidad=true],

Servicio [codMercancia=1973851, servicio=Mantenimiento de martillo neumático, presupuesto=3500.0, enPromocion=true],

Producto [codMercancia=4897183, producto=Lampara led 11W, precioProducto=450.0, porcentajeDecuento=10.0, esDescuentoEn2daUnidad=false],

Servicio [codMercancia=2891728, servicio=Mantenimiento aire acondicionado, presupuesto=2250.0, enPromocion=false]

3-2) Reintentar agregar el siguiente objeto:

Servicio [codMercancia=2891728, servicio=Mantenimiento aire acondicionado, presupuesto=2250.0, enPromocion=false]

3-3) Reintentar agregar el siguiente objeto:

Producto [codMercancia=3780173, producto=Lampara led 9W, precioProducto=330.0, porcentajeDecuento=10.0, esDescuentoEn2daUnidad=true]

4) Traer todos los objetos Mercancia en oferta enOferta=true