



---

Ingeniería Web para Telecomunicaciones  
**Semestre: 2021-2**

### **Laboratorio 3: Herencia y Polimorfismo**

---

- Duración: 1 hora y 40 minutos.
  - Evaluación: **Individual**
  - Descargar : **Algunas\_Clases.zip** (Main.java)
  - Subir a Paideia:
    - El nombre del proyecto debe tener el siguiente formato: **LAB3\_<código PUCP 8 dígitos>**.
    - Comprimir la carpeta del proyecto y subirlo a Paideia.
- 

#### **Recomendación: “Leer todo el documento antes de empezar”**

Se requiere implementar una aplicación por consola que permita registrar las ventas de periféricos a personas pudiendo ser estas: estudiantes PUCP, trabajadores PUCP o público general. Entre los periféricos que pueden ser comprados son: monitor, teclado o mouse. Solo si la persona es un alumno PUCP tendrá un descuento del total de la compra del 50% y si es trabajador PUCP un descuento del 25%. La información a registrarse es la siguiente:

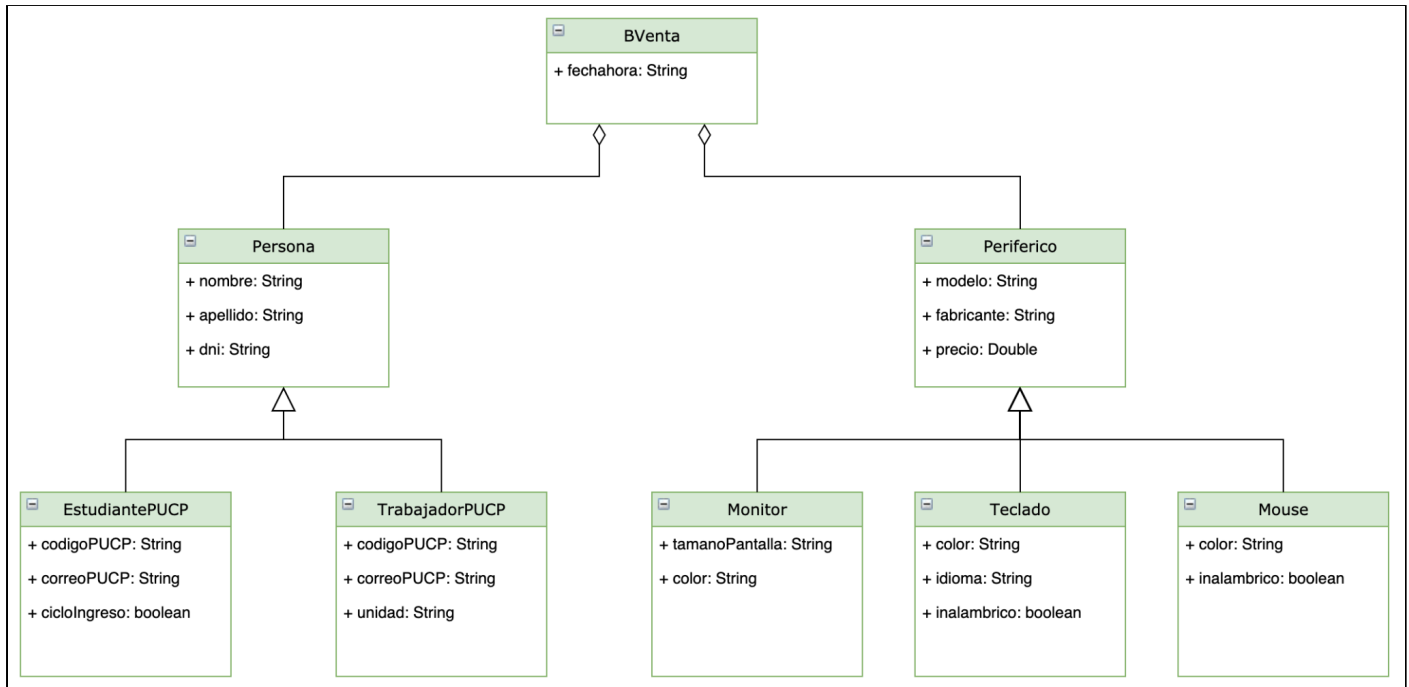
- registrar la información de la persona y los periféricos a ser comprados,
- mostrar un listado de todas las ventas,
- una opción que permita saber la compra de una determinada persona vía el DNI.

Para la implementación seguir los siguientes pasos:

#### **1. Implementar las Clases (4 ptos)**

Implementar las Clases:

- **Persona**,
  - **Estudiante PUCP** hereda de Persona ,
  - **Trabajador PUCP** hereda de Persona,
- **Periferico**,
  - **Monitor** hereda de Periférico,
  - **Teclado** hereda de Periférico,
  - **Mouse** hereda de Periférico
- Entre los atributos a tomar en cuenta para el registro se tienen:



- **Periferico**
  - modelo
  - fabricante
  - precio
- **Monitor**
  - tamañoPantalla
  - color
- **Teclado**
  - color
  - idioma
  - inalambrico
- **Mouse**
  - color
  - inalambrico
- **Persona**
  - nombre
  - apellido
  - dni
- **EstudiantePUCP**
  - codigoPUCP
  - correoPUCP
  - cicloIngreso
- **TrabajadorPUCP**
  - codigoPUCP
  - correoPUCP
  - unidad



Se tiene una clase adicional que es **BVenta** la cual tiene los siguientes atributos:

- **FechaHora** cuyo valor es la fecha y hora del registro de venta
- **persona** objeto de tipo Persona
- **ListaPerifericos** que es un arreglo (ArrayList) de objetos de tipo Periferico

\*No olvidar encapsular todos los atributos (**get** y **set**).

\*No olvidar a los constructores aprovechando la herencia correspondiente.

## 2. Implementar Registro de Ventas (5 ptos)

- En el registro de ventas se registra la siguiente información:
  - Datos de la persona
    - Nombre
    - Apellido
    - DNI
  - Se consulta por datos adicionales de la persona:
    - Si es un estudiante PUCP
      - Código PUCP
      - Correo PUCP
      - Ciclo de Ingreso
    - Si es trabajador PUCP
      - Código PUCP
      - Correo PUCP
      - Unidad
  - Y, se registra la compra de periféricos con información general y particular de estos.
- Implementar la clase **DVentas** con un atributo de la lista de ventas llamado **listaVentas** (ArrayList de BVenta)
- Implementar en la clase **DVentas** el método **registroVentas** que tiene como parámetro un objeto BVenta. Se encarga de registrar las ventas en el atributo lista **listaVentas** y retorna el valor de este.
- Implementar la clase **CVenta** que tiene como atributo un objeto **DVentas** e implementar su constructor.
- Implementar en la clase **CVenta** el método **registrarCVenta**, no recibe parámetros, su función es capturar los datos ingresados por consola sobre la venta y utilizar el método **DVentas.registroVentas** implementado anteriormente. El método **registrarCVenta** no retorna un valor.

En la **Figura 1** se observa un ejemplo del registro de una Venta.



```
+++++
Ingrese opción : 1

Registrar venta
-----

Ingrese Nombre : Enrique
Ingrese Apellido : Larios
Ingrese DNI : 12345678
¿Es estudiante PUCP? (Si/No): Si
Ingrese Código PUCP : 19001234
Ingrese Correo PUCP : enrique.larios@pucp.edu.pe
Ingrese Ciclo de Ingreso : 1900-1
¿Desea comprar un Monitor? (Si/No): Si
Modelo : MT1000
Fabricante : Apple
Precio (S/.): 5000
Tamaño : 27"
Color : Blue
¿Desea comprar un Teclado? (Si/No): Si
Modelo : T1000
Fabricante : Apple
Precio (S/.): 1000
Idioma : Español
Color : Red
¿Es inalámbrico? (Si/No): Si
¿Desea comprar un Mouse? (Si/No): Si
Modelo : MS1000
Fabricante : Apple
Precio (S/.): 2000
Color : Green
¿Es inalámbrico? (Si/No): Si
```

Figura 1

\*En la clase **Main** se tiene la invocación a `cVentas.registrarCVenta()`;  
Utilizar herencia y polimorfismo.



### 3. Implementar listar Ventas (5 pts)

- Implementar en la clase **DVentas** el método **listarDVentas** que no tiene parámetros y retorna las ventas en un arreglo de cadenas (ArrayList de String).

En la cadena retorna:

- El **nombre y apellido** (primer elemento del arreglo)
- Fecha y hora de la venta** (segundo elemento del arreglo) \*\*\*\* ir a **OBSERVACIONES**
- La lista de periféricos vendidos (los siguientes elementos del arreglo):

- Tipo de periférico

- Si es Monitor

*Periférico : Monitor | Nombre : MT1000 | Fabricante : Apple | Tamaño : 27" | Color: Blue | Precio: 5000.0*

- Si es Teclado

*Periférico : Teclado | Nombre : T1000 | Fabricante : Apple | Inalámbrico : Si | Idioma: Español | Precio: 1000.0*

- Si es Mouse

*Periférico : Mouse | Nombre : MS1000 | Fabricante : Apple | Inalámbrico : Si | Precio: 2000.0*

- Implementar en la clase **CVenta** el método **listarCVenta**, no recibe parámetros, su función es listar todas las ventas para ello hace uso del método **DVentas.listarDVentas**, ver Figura 2

```
+++++
Ingrese opción : 2

Listar ventas
-----

----- Enrique Larios -----
Fecha : 2021-09-19T10:47:57.203294
Periférico : Monitor | Nombre : MT1000 | Fabricante : Apple | Tamaño : 27" | Color: Blue | Precio: 5000.0
Periférico : Teclado | Nombre : T1000 | Fabricante : Apple | Inalámbrico : Si | Idioma: Español | Precio: 1000.0
Periférico : Mouse | Nombre : MS1000 | Fabricante : Apple | Inalámbrico : Si | Precio: 2000.0
----- Juan Barturen -----
Fecha : 2021-09-19T11:49:52.329335
Periférico : Mouse | Nombre : M1001 | Fabricante : Apple | Inalámbrico : Si | Precio: 2000.0
```

Figura 2

- Si no hay ventas registradas se muestra el mensaje:

```
---> No hay ventas registradas <---
```

\*En la clase **Main** se tiene la invocación a **cVentas.listarCVenta()**;

Utilizar herencia y polimorfismo.



#### 4. Implementar buscar ventas por DNI (5 ptos)

- Implementar en la clase **DVentas** el método **listarDVentasxPersona** que recibe como parámetro el Dni como una cadena de String y retorna cada una de las ventas en un arreglo de cadenas (ArrayList de String).
- Implementar en la clase **CVenta** el método **listarCVentasxDNI** que no recibe parámetros, su función recibir por consola el DNI de una persona y listar todas las ventas realizadas por este, para ello hace uso del método DVentas.**listarDVentasxPersona**.

```
Ingrese opción : 3

Buscar ventas por persona (DNI)
-----

Ingrese el DNI : 12345678
[{Persona : Enrique Larios, DNI : 12345678}, Periférico : Monitor | Nombre : MT1000| Fabricante : Apple| Tamaño : 27"| Color: Blue| Precio: 5000.0, Periférico : Teclado | Nombre : T1000| Fabricante : Apple| Inalámbrico : Si| Idioma: Español| Precio: 1000.0, Periférico : Mouse | Nombre : MS1000| Fabricante : Apple| Inalámbrico : Si| Precio: 2000.0, Total con desc. 50% (S/.) = 4000.0]
```

**Se debe observar que al final se muestra el total con el descuento que le corresponde según el tipo de persona.**

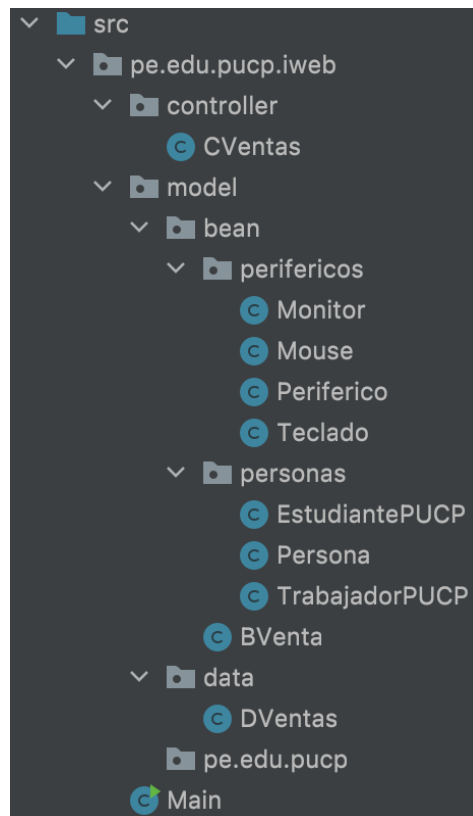
\*En la clase **Main** se tiene la invocación a **cVentas.listarCVentasxDNI**

Utilizar herencia y polimorfismo.



## OBSERVACIONES

1. La estructura del directorio **src** del proyecto debe ser como se muestra, **Figura 3**:



**Figura 3**

2. \*\*\*\* Para obtener la hora actual utilizar la clase **java.time.LocalDateTime** y la invocación:  
`LocalDateTime.now().toString()`
3. La clase **Main.java** ya se tiene implementada:  
NO SE DEBEN MODIFICAR LOS CONTENIDOS DE LOS ARCHIVOS **Main.java** caso contrario se descontarán 10 puntos.

Lima, 20 de setiembre 2021