

### Simulacro Primer Parcial IPOO

Una empresa dedicada a la venta de materiales para la construcción de la región, desea implementar una aplicación que le permita gestionar la información de sus Clientes, de los productos y de las ventas realizadas. Para ello se almacena información de todos sus clientes, de cada uno de los productos disponibles para la venta y de todas las ventas realizadas.

Implementar las siguientes clases: **Productos, Venta, Cliente y Empresa**

En la clase **Cliente**:

1. Se registra la siguiente información: nombre, apellido, si esta o no dado de baja, el tipo y el número de documento. Si un cliente esta dado de baja, no puede registrar compras desde el momento de su baja.
2. Método constructor que recibe como parámetros los valores iniciales para los atributos.
3. Los métodos de acceso de cada uno de los atributos de la clase.
4. Redefinir el método **\_toString** para que retorne la información de los atributos de la clase.

En la clase **Producto**:

1. Se registra la siguiente información: código, costo, año de fabricación, descripción, porcentaje incremento anual, activo (atributo que va a contener un valor true, si el producto esta disponible para la venta y false en caso contrario).
2. Método constructor que recibe como parámetros los valores iniciales para los atributos definidos en la clase.
3. Los métodos de acceso de cada uno de los atributos de la clase.
4. Redefinir el método **toString** para que retorne la información de los atributos de la clase.
5. Implementar el método **darPrecioVenta** el cual calcula el valor por el cual puede ser vendido el producto. Si el producto no se encuentra disponible para la venta retorna un valor < 0. Si el producto esta disponible para la venta, el método realiza el siguiente cálculo:

$$\$_{venta} = \$_{compra} + \$_{compra} * (anio * por\_inc\_anual)$$

donde  $\$_{compra}$ : es el costo del producto.

$anio$ : cantidad de años transcurridos desde que se fabrico el producto.

$por\_inc\_anual$ : porcentaje incremento anual del producto.

En la clase **Venta**:

1. Se registra la siguiente información: número, fecha, referencia al cliente, referencia a una colección de productos y el precio final.
2. Método constructor que recibe como parámetros cada uno de los valores a ser asignados a cada atributo de la clase.
3. Los métodos de acceso de cada uno de los atributos de la clase.
4. Redefinir el método **\_toString** para que retorne la información de los atributos de la clase.
5. Implementar el método **incorporarProducto(objProducto)** que recibe por parámetro un objeto producto y lo incorpora, si es posible la venta, a la colección de productos de la venta. El método cada vez que incorpora un producto a la venta, debe actualizar la variable instancia *precio final* de la venta. Utilizar el método que calcula el precio de venta de un producto donde crea necesario.

En la clase **Empresa**:

1. Se registra la siguiente información: denominación, dirección, la colección de clientes, colección de productos y la colección de ventas realizadas.
2. Método constructor que recibe como parámetros los valores iniciales para los atributos de la clase.
3. Los métodos de acceso para cada una de las variables instancias de la clase.
4. Redefinir el método **\_toString** para que retorne la información de los atributos de la clase.
5. Implementar el método **retornarVehículo(\$codigoProducto)** que recorre la colección de productos de la Empresa y retorna la referencia al objeto producto cuyo código coincide con el recibido por parámetro.
6. Implementar el método **registrarVenta(\$colCodigosProductos, \$objCliente)** método que recibe por parámetro una colección de códigos de productos, la cual es recorrida, se busca el objeto producto co-

respondiente al código y se incorpora a la colección de productos de la instancia Venta que debe ser creada.

Recordar que no todos los clientes ni todas los productos, están disponibles para registrar una venta en un momento determinado.

El método debe setear los variables instancias de venta que corresponda y retornar el importe final de la venta.

7. Implementar el método **retornarVentasXCliente(\$tipo,\$numDoc)** que recibe por parámetro el tipo y número de documento de un Cliente y retorna una colección con las ventas realizadas al cliente.

Implementar un script **TestEmpresa** en la cual:

1. Cree 2 instancias de la clase Cliente: \$objCliente1, \$objCliente2.
2. Se crea 3 objetos productos con la información visualizada en la tabla: código, costo, año fabricación, descripción, porcentaje incremento anual, activo.

código	costo	año_fabricacion	Descripcion	porc_increment	activo
11	50000	2018	Cemento loma Negra	70%	true
12	10000	2019	Hierro del 12	60%	true
13	10000	2020	Cal Santa Clara	50%	false

3. Se crea un objeto Empresa con la siguiente información: **denominación = "Cosmos"**, **dirección = "Av Argentina 123"**, **colección de productos = [\$objProducto1, \$objProducto1, \$objProducto1]**, **colección de clientes = [objCliente1, objCliente2]**, **la colección de ventas realizadas=[]**.
4. Invocar al método **registrarVenta(\$colCodigosProducto, \$objCliente)** de la Clase Empresa donde el \$objCliente es una referencia a la clase Cliente almacenada en la variable \$objCliente2 (creada en el punto 1) y la colección de códigos de productos es la siguiente [11,12,13]. Visualizar el resultado obtenido.
5. Invocar al método **registrarVenta(\$colCodigosProductos, \$objCliente)** de la Clase Empresa donde el \$objCliente es una referencia a la clase Cliente almacenada en la variable \$objCliente2 (creada en el punto 1) y la colección de códigos de productos es la siguiente [0]. Visualizar el resultado obtenido.
6. Invocar al método **registrarVenta(\$colCodigosProductos, \$objCliente)** de la Clase Empresa donde el \$objCliente es una referencia a la clase Cliente almacenada en la variable \$objCliente2 (creada en el punto 1) y la colección de códigos de productos es la siguiente [2]. Visualizar el resultado obtenido.
7. Invocar al método **retornarVentasXCliente(\$tipo,\$numDoc)** donde el tipo y número de documento se corresponden el tipo y número de documento del \$objCliente1.
8. Invocar al método **retornarVentasXCliente(\$tipo,\$numDoc)** donde el tipo y número de documento se corresponden el tipo y número de documento del \$objCliente2.
9. Realizar un echo de la variable Empresa creada en 2.