

## **PRACTICA NRO. 1**

### **PROGRAMACION ORIENTADA A OBJETOS**

#### **A. REALIZAR LOS SIGUIENTES EJERCICIOS EN UML Y PSEUDOCODIGO**

1. Identifica y representa en diagramas UML, 4 clases y tres métodos de cada uno de los siguientes contextos (Represente cada clase en UML:
  - a. Crisis económica en Bolivia
  - b. Curso de invierno de la carrera de informática
  - c. Aplicación tiktok
2. Crear la clase cuadrado. Un cuadrado se define por:
  - LadoRealizar las siguientes operaciones:
  - Leer datos
  - Mostrar datos
  - Calcular el área
  - Calcular el perímetro
  - Modificar el lado y calcular nuevamente el área y el perímetro
  - Mostrar Datos
3. Modela la clase perro con atributos: nombre, raza y edad. Realizar las siguientes operaciones :
  - a. Crea un constructor par inicializar datos
  - b. Métodos getters y setters de cada atributo.
  - c. Muestra datos
  - d. Implementa los métodos: ladrar(), comer()
  - e. Verifica si la edad del perro es menor o igual a dos años y muestra un mensaje de que es cachorro
4. Modela la clase Celular:
  - a. Define atributos, 3 constructores diferentes.
  - b. Instanciar 3 objetos de forma diferente
  - c. Métodos getters y setters de cada atributo.
  - d. Calcula el costo de una llamada según minutos.
  - e. Compara si dos celulares son de la misma marca.
  - f. Muestra el celular que tenga el sistema operativo iOS.
5. La tienda “Compre rápido” que vende celulares quiere tener registro de cada producto que posee. Para ello, se requiere crear un programa donde se pueda almacenar cada producto con su información. Crear una entidad Movil con los atributos marca, precio, modelo, memoriaRam, almacenamiento (GB) y código (es un arreglo numérico de dimensión 7, donde cada subíndice alojará un número correspondiente al código).

A continuación, se implementarán los siguientes métodos:

  - Un constructor por defecto y un constructor parametrizado.
  - Métodos getters y setters de cada atributo.
  - Instanciar 2 objetos de forma diferente
  - Método ingresarCodigo(): este método permitirá ingresar el código completo de siete números de un celular.
  - Se desea instalar la aplicación “Banca movil” que ocupa 60 MB, sin no existe espacio se deberá emitir un mensaje, caso contrario se deberá actualizar el almacenamiento.

6. Sean las siguientes clases:

Inventario
-nombre -direccion -nroActivos -activos[50,3]
.....

Activo
-nombre -codigo -fecha de compra -precio -tiempo de vida
...

Dónde: activo[i][0] = nombre, activo [i][1] = codigo , activo [i][2] = fechaCompra, activo [i][3] = precio, activo [i][4] = tiempovida . Se pide:

- Instanciar un objeto Inventario y 3 objetos activo de diferente manera y mostrar datos
- Dada la fecha Z, verificar si un activo a cumplido si tiempo de vida, si es así, emitir un mensaje de que requiere ser revaluado.
- Mostrar cual es el activo más caro
- Adicionar un objeto activo al inventario, siempre y cuando el activo no exista

## POLIMORFISMO

### I. Clases abstractas

- Escribir las clases para la gestión de productos.

Los productos tienen los siguientes atributos:

- Nombre
- Precio

Se distinguen dos tipos de productos:

- Estándar: Incluye un atributo adicional llamado sección.
- Ofertado: Incluye un atributo llamado días, que es el número de días que faltan para que la oferta finalice.

Además de constructores, getters y setters, tendrán un método obtenerPrecioPedido() que recibe como parámetro las unidades pedidas del producto y devuelve el precio final.

Esté método funcionará de forma diferente si es un producto estándar u ofertado

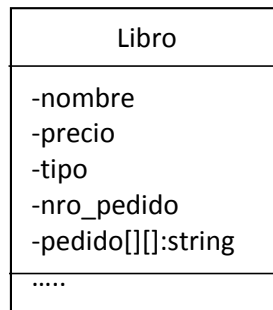
- Si es estándar, devolverá el número de unidades por el precio, aplicando un 15% de descuento en caso de que el número de unidades sea igual o superior a 5.

- Si es ofertado, independientemente del número de unidades:
  - Si le queda 1 día a la oferta aplicará un 20% de descuento.
  - Si le quedan 2 o 3 a la oferta, se aplicará un 15% de descuento.
  - Si le quedan más de 3 días se aplicará un 10% de descuento.

Después, se creará un programa principal y se crearán dos pedidos de productos estándar y 3 de productos ofertados con ofertas que caducan en 2 días, asignando un precio cualquiera a cada uno. Después mostrar el total del pedido teniendo en cuenta que se compran 5 unidades de cada uno.

## II. Sobrecarga de Métodos

2. Crea una clase Area que calcule el área de diferentes figuras:
  - a) calcularArea(int lado) → área del cuadrado
  - b) calcularArea(double radio) → área del círculo
  - c) calcularArea(int base, int altura) → área del rectángulo
3. Crea una clase base InstrumentoMusical con el método tocar().
  - a) Crea las subclases Guitarra, Piano, Tambor, cada una sobrescribiendo el método tocar() para mostrar un mensaje diferente.
4. Dado el siguiente diagrama UML:



Donde: tipo={estándar, ofertado},

pedido[i][0] = nroPedido, pedido[i][0] = cliente, pedido[i][2] = costoTotal. Se pide:

- a) Instanciar 4 productos con un constructor por defecto y 3 con constructores parametrizados.
- b) Sobrecargar el método mostrar, para:
  1. obtener los pedidos que realizo el cliente x
  2. obtener los productos de tipo estándar

## III. Sobrecarga de Operadores

5. Crea una clase Fecha con atributos día, mes, año. Sobrecarga el operador == para comparar si dos fechas son iguales.
6. Crea una clase Tiempo con atributos horas, minutos, segundos. Realiza las siguientes operaciones:
  - a) Sobrecarga el operador - para calcular la diferencia entre dos tiempos.
  - b) Sobrecarga el operador + para calcular la suma de dos objetos Tiempo, devuelve otro Tiempo.

- c) Sobrecarga el operador `==` para devolver un mensaje que muestre la hora si los dos objetos tienen el mismo tiempo.
7. Sobrecarga el operador `>` para comparar dos objetos tiempo y devolver un mensaje que muestre la hora del tiempo mayor.
8. Dada la siguiente clase:

Fraccion
-numerador -denominador
+Fraccion(n,d) +leer() +mostar()

- a) Sobrecargar el operador `+`, para realizar la suma de dos objetos fracción
- b) Sobrecargar el operador `-`, para realizar la resta de dos objetos fracción
- c) Sobrecargar el operador `*`, para realizar la multiplicación de dos objetos fracción
- d) Sobrecargar el operador `/`, para realizar la división de dos objetos fracción
- e) Sobrecargar el operador `!`, para obtener el mcd(máximo común divisor) de dos objetos fracción. Sugerencia: usa el algoritmo de Euclides

**B. La defensa de esta práctica será coordinada en clases**