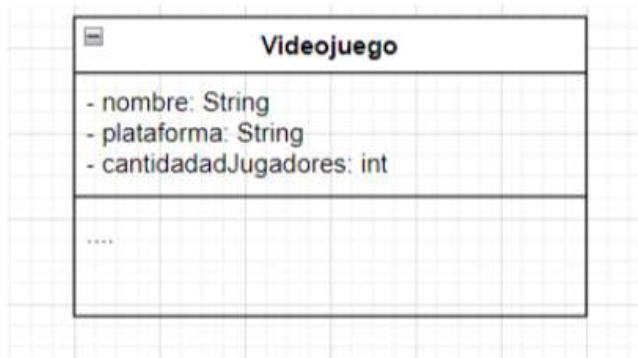


POLIMORFISMO

1. Sea la clase Videojuego:



- Instanciar al menos 2 videojuegos
- Sobrecargar el constructor 2 veces
- Implementar un método `mostrar()`
- Sobrecargar el método `agregarJugadores()` donde en el primero se agregue solo 1 jugador y en otro se ingrese una cantidad de jugadores a aumentar.

```
class Videojuego:
    def __init__(self, nombre="Desconocido", plataforma="No especificada", cantidad_jugadores=0):
        self.nombre = nombre
        self.plataforma = plataforma
        self.cantidad_jugadores = cantidad_jugadores

    def mostrar(self):
        print(f"Nombre del juego: {self.nombre}")
        print(f"Plataforma: {self.plataforma}")
        print(f"Cantidad de jugadores: {self.cantidad_jugadores}")

    def agregar_jugadores(self, cantidad=1):
        self.cantidad_jugadores += cantidad

def main():
    v1 = Videojuego("Minecraft")
    v2 = Videojuego("FIFA", "Deportes", 4)

    v1.agregar_jugadores()
    v2.agregar_jugadores(3)

    v1.mostrar()
    v2.mostrar()

if __name__ == "__main__":
    main()
```

Nombre del juego: Minecraft

Plataforma: No especificada

Cantidad de jugadores: 1

Nombre del juego: FIFA

Plataforma: Deportes

Cantidad de jugadores: 7

PS C:\Users\Angela\Desktop\Cristhian\PRACTICA INF-121\POLIIMORFISMO>

CODIGO EN JAVA:

```

package Ejercicio_1;

public class Videojuego {

    private String nombre;
    private String plataforma;
    private int cantidadJugadores;

    public Videojuego() {
        this.nombre = "Desconocido";
        this.plataforma = "No especificada";
        this.cantidadJugadores = 0;
    }

    public Videojuego(String nombre, String plataforma) {
        this.nombre = nombre;
        this.plataforma = plataforma;
        this.cantidadJugadores = 0;
    }

    public Videojuego(String nombre, String plataforma, int cantidadJugadores) {
        this.nombre = nombre;
        this.plataforma = plataforma;
        this.cantidadJugadores = cantidadJugadores;
    }

    public void mostrar() {
        System.out.println("Nombre del juego: " + nombre);
        System.out.println("Plataforma: " + plataforma);
        System.out.println("Cantidad de jugadores: " + cantidadJugadores);
    }

    public void agregarJugadores() {
        this.cantidadJugadores += 1;
    }

    public void agregarJugadores(int cantidad) {
        this.cantidadJugadores += cantidad;
    }
}

```

```

package Ejercicio_1;

public class Principal {

    public static void main(String[] args) {

        Videojuego juego1 = new Videojuego("The Legend of Zelda", "Nintendo Switch");
        Videojuego juego2 = new Videojuego("Cyberpunk 2077", "PC", 1000000);

        System.out.println("--- Información inicial ---");
        juego1.mostrar();
        System.out.println();
        juego2.mostrar();

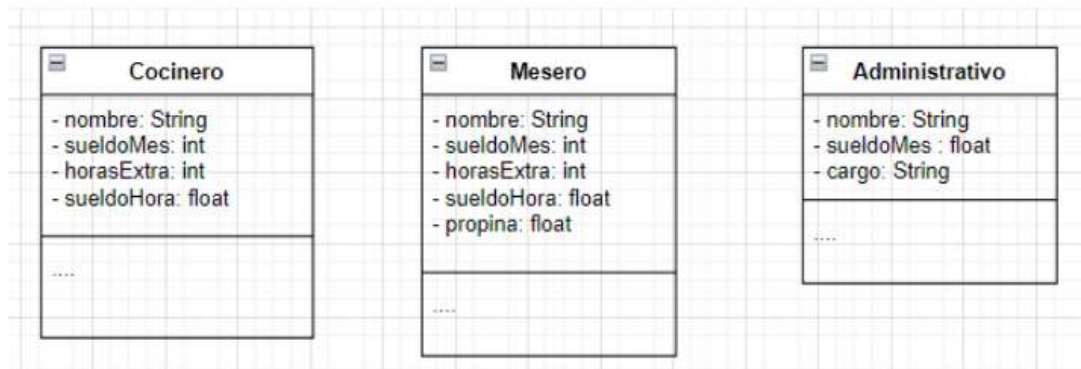
        juego1.agregarJugadores();
        juego1.agregarJugadores(5);

        juego2.agregarJugadores(10000);

        System.out.println("\n--- Después de agregar jugadores ---");
        juego1.mostrar();
        System.out.println();
        juego2.mostrar();
    }
}

```

3. Un restaurante organiza a su personal mediante las siguientes clases:



- Instanciar 1 Cocinero, 2 objetos Mesero y 2 objetos Administrativo.
- Sobrecargar el método SueldoTotal para mostrar el sueldo total, sumándole las horas extra, considerando el sueldo por hora y la propina en caso de los meseros.
- Sobrecargar el método para mostrar a aquellos Empleados que tengan SueldoMes igual a X.

```
class Cocinero:
    def __init__(self, nombre, sueldoMes, horasExtra, sueldoHora):
        self.__nombre = nombre
        self.__sueldoMes = sueldoMes
        self.__horasExtra = horasExtra
        self.__sueldoHora = sueldoHora

    def sueldo_total(self):
        return self.__sueldoMes + (self.__horasExtra * self.__sueldoHora)

    def mostrar_si_sueldo_igual(self, x):
        if self.__sueldoMes == x:
            print(f"{self.__nombre} (Cocinero) tiene sueldoMes = {x}")

class Mesero:
    def __init__(self, nombre, sueldoMes, horasExtra, sueldoHora, propina):
        self.__nombre = nombre
        self.__sueldoMes = sueldoMes
        self.__horasExtra = horasExtra
        self.__sueldoHora = sueldoHora
        self.__propina = propina

    def sueldo_total(self):
        return self.__sueldoMes + (self.__horasExtra * self.__sueldoHora) + self.__propina

    def mostrar_si_sueldo_igual(self, x):
        if self.__sueldoMes == x:
            print(f"{self.__nombre} (Mesero) tiene sueldoMes = {x}")
```

```

class Administrativo:
    def __init__(self, nombre, sueldoMes, cargo):
        self.__nombre = nombre
        self.__sueldoMes = sueldoMes
        self.__cargo = cargo

    def sueldo_total(self):
        return self.__sueldoMes

    def mostrar_si_sueldo_igual(self, x):
        if self.__sueldoMes == x:
            print(f"{self.__nombre} (Administrativo) tiene sueldoMes = {x}")

cocinero = Cocinero("Luis", 3000, 10, 50)
mesero1 = Mesero("Ana", 2500, 5, 40, 200)
mesero2 = Mesero("Carlos", 2600, 8, 35, 150)
admin1 = Administrativo("Beatriz", 2800, "Contadora")
admin2 = Administrativo("Jorge", 3000, "Gerente")

print("=== Sueldos Totales ===")
print("Cocinero:", cocinero.sueldo_total())
print("Mesero 1:", mesero1.sueldo_total())
print("Mesero 2:", mesero2.sueldo_total())
print("Administrativo 1:", admin1.sueldo_total())
print("Administrativo 2:", admin2.sueldo_total())

x = 3000
print(f"\n=== Empleados con sueldoMes = {x} ===")
cocinero.mostrar_si_sueldo_igual(x)

```

```

x = 3000
print(f"\n=== Empleados con sueldoMes = {x} ===")
cocinero.mostrar_si_sueldo_igual(x)
mesero1.mostrar_si_sueldo_igual(x)
mesero2.mostrar_si_sueldo_igual(x)
admin1.mostrar_si_sueldo_igual(x)
admin2.mostrar_si_sueldo_igual(x)

```

```

=== Sueldos Totales ===
Cocinero: 3500
Mesero 1: 2900
Mesero 2: 3030
Administrativo 1: 2800
Administrativo 2: 3000

=== Empleados con sueldoMes = 3000 ===
Luis (Cocinero) tiene sueldoMes = 3000
Jorge (Administrativo) tiene sueldoMes = 3000
PS C:\Users\Angela\Desktop\Cristhian\PRACTICA INF-121\POLIIMORFISMO>

```


CODIGO EN JAVA

```
package Ejercicio_3;

public class Administrativo {
    private String nombre;
    private float sueldoMes;
    private String cargo;

    public Administrativo(String nombre, float sueldoMes, String cargo) {
        this.nombre = nombre;
        this.sueldoMes = sueldoMes;
        this.cargo = cargo;
    }

    public float sueldoTotal() {
        return sueldoMes;
    }

    public void mostrarSiSueldoIgual(float x) {
        if (sueldoMes == x) {
            System.out.println(nombre + " (Administrativo) tiene sueldoMes = " + x);
        }
    }
}
```

```
package Ejercicio_3;

public class Cocinero {
    private String nombre;
    private int sueldoMes;
    private int horasExtra;
    private float sueldoHora;

    public Cocinero(String nombre, int sueldoMes, int horasExtra, float sueldoHora) {
        this.nombre = nombre;
        this.sueldoMes = sueldoMes;
        this.horasExtra = horasExtra;
        this.sueldoHora = sueldoHora;
    }

    public float sueldoTotal() {
        return sueldoMes + (horasExtra * sueldoHora);
    }

    public void mostrarSiSueldoIgual(float x) {
        if (sueldoMes == x) {
            System.out.println(nombre + " (Cocinero) tiene sueldoMes = " + x);
        }
    }
}
```

```
package Ejercicio_3;

public class Mesero {
    private String nombre;
    private int sueldoMes;
    private int horasExtra;
    private float sueldoHora;
    private float propina;

    public Mesero(String nombre, int sueldoMes, int horasExtra, float sueldoHora, float propina) {
        this.nombre = nombre;
        this.sueldoMes = sueldoMes;
        this.horasExtra = horasExtra;
        this.sueldoHora = sueldoHora;
        this.propina = propina;
    }

    public float sueldoTotal() {
        return sueldoMes + (horasExtra * sueldoHora) + propina;
    }

    public void mostrarSiSueldoIgual(float x) {
        if (sueldoMes == x) {
            System.out.println(nombre + " (Mesero) tiene sueldoMes = " + x);
        }
    }
}
```

```

package Ejercicio_3;

public class Principal {
    public static void main(String[] args) {
        Cocinero cocinero = new Cocinero("Luis", 3000, 10, 50);
        Mesero mesero1 = new Mesero("Ana", 2500, 5, 40, 200);
        Mesero mesero2 = new Mesero("Carlos", 2600, 8, 35, 150);
        Administrativo admin1 = new Administrativo("Beatriz", 2800, "Contadora");
        Administrativo admin2 = new Administrativo("Jorge", 3000, "Gerente");

        System.out.println("=== Sueldos Totales ===");
        System.out.println("Cocinero: " + cocinero.sueldoTotal());
        System.out.println("Mesero 1: " + mesero1.sueldoTotal());
        System.out.println("Mesero 2: " + mesero2.sueldoTotal());
        System.out.println("Administrativo 1: " + admin1.sueldoTotal());
        System.out.println("Administrativo 2: " + admin2.sueldoTotal());

        float x = 3000;
        System.out.println("\n=== Empleados con sueldoMes = " + x + " ===");
        cocinero.mostrarSiSueldoIgual(x);
        mesero1.mostrarSiSueldoIgual(x);
        mesero2.mostrarSiSueldoIgual(x);
        admin1.mostrarSiSueldoIgual(x);
        admin2.mostrarSiSueldoIgual(x);
    }
}

```

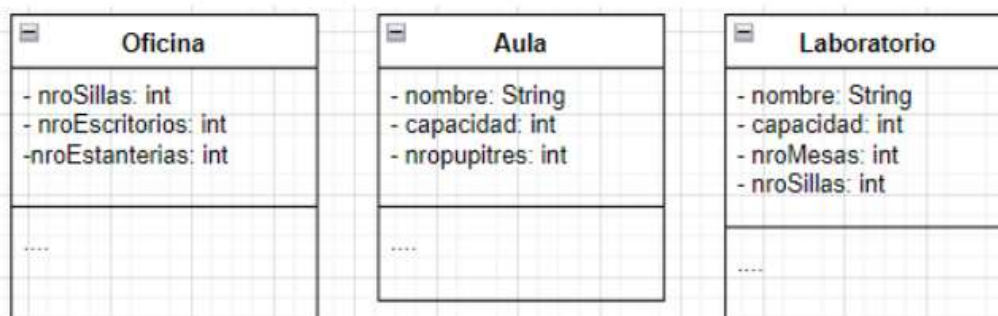
```

=== Sueldos Totales ===
Cocinero: 3500.0
Mesero 1: 2900.0
Mesero 2: 3030.0
Administrativo 1: 2800.0
Administrativo 2: 3000.0

=== Empleados con sueldoMes = 3000.0 ===
Luis (Cocinero) tiene sueldoMes = 3000.0
Jorge (Administrativo) tiene sueldoMes = 3000.0

```

5. Se hace referencia a algunos de los diferentes ambientes de la Universidad mediante las siguientes clases:



- Instanciar 2 objetos Oficina, 2 Aulas y 1 Laboratorio
- Crear un método mostrar() para mostrar los datos de cada objeto
- Sobrecargar el método cantidadMuebles(), para conocer el número total de muebles dentro de cada ambiente

```

class Habitacion:
    def __init__(self, nombre):
        self.nombre = nombre

    def calcular_costo(self, noches):
        return 0

    def reservar(self, noches, personas=None):
        if personas:
            print(f"{self.nombre} reservada por {noches} noche(s) para {personas} persona(s).")
        else:
            print(f"{self.nombre} reservada por {noches} noche(s).")

class HabitacionSimple(Habitacion):
    def __init__(self):
        super().__init__("Habitación Simple")

    def calcular_costo(self, noches):
        return noches * 50 # $50 por noche

class HabitacionDoble(Habitacion):
    def __init__(self):
        super().__init__("Habitación Doble")

    def calcular_costo(self, noches):
        return noches * 80 # $80 por noche

class Suite(Habitacion):
    def __init__(self):
        super().__init__("Suite de Lujo")

```

```

    def calcular_costo(self, noches):
        return noches * 150 # $150 por noche

def main():
    simple = HabitacionSimple()
    doble = HabitacionDoble()
    suite = Suite()

    habitaciones = [simple, doble, suite]

    for hab in habitaciones:
        hab.reservar(2)
        hab.reservar(3, 2)
        print(f"Costo por {hab.nombre}: ${hab.calcular_costo(2)}\n")

if __name__ == "__main__":
    main()

```

Habitación Simple reservada por 2 noche(s).
Habitación Simple reservada por 3 noche(s) para 2 persona(s).
Costo por Habitación Simple: \$100

Habitación Doble reservada por 2 noche(s).
Habitación Doble reservada por 3 noche(s) para 2 persona(s).
Costo por Habitación Doble: \$160

Suite de Lujo reservada por 2 noche(s).
Suite de Lujo reservada por 3 noche(s) para 2 persona(s).
Costo por Suite de Lujo: \$300

7. Se hace referencia a algunos animales mediante las siguientes clases:

- a) Instanciar 1 Perro, 1 Gato y 1 Pájaro.
- b) Sobrecargar el método hacerSonido() para que cada animal emita su sonido característico.
- c) Implementar un método moverse() que indique cómo se mueve cada animal (correr, saltar, volar, etc.).


```
class Animal:
    def hacer_sonido(self):
        print("Sonido genérico de animal.")

    def moverse(self):
        print("Movimiento genérico de animal.")

class Perro(Animal):
    def hacer_sonido(self):
        print("El perro dice: ¡Guau guau!")

    def moverse(self):
        print("El perro corre alegremente.")

class Gato(Animal):
    def hacer_sonido(self):
        print("El gato dice: ¡Miau!")

    def moverse(self):
        print("El gato salta y camina con elegancia.")

class Pajaro(Animal):
    def hacer_sonido(self):
        print("El pájaro dice: ¡Pío pío!")

    def moverse(self):
        print("El pájaro vuela por el cielo.")

def main():
    perro = Perro()
```

```
    perro = Perro()
    gato = Gato()
    pajaro = Pajaro()

    animales = [perro, gato, pajaro]

    for animal in animales:
        animal.hacer_sonido()
        animal.moverse()
        print("-" * 40)

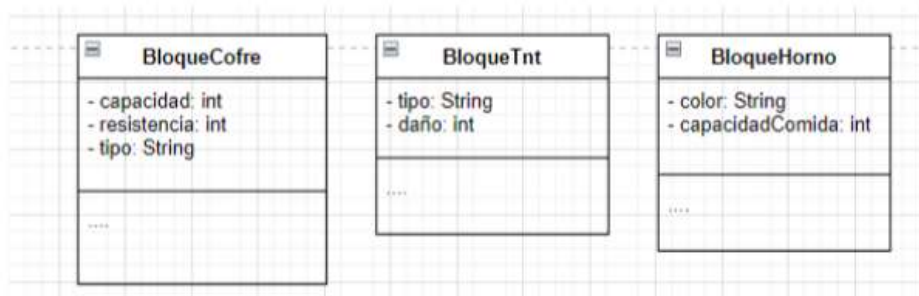
if __name__ == "__main__":
    main()
```

```
El perro dice: ¡Guau guau!  
El perro corre alegremente.
```

```
-----  
El gato dice: ¡Miau!  
El gato salta y camina con elegancia.
```

```
-----  
El pájaro dice: ¡Pío pío!  
El pájaro vuela por el cielo.
```

9. Para los bloques del juego Minecraft se usará los siguientes objetos:



- Crear e instanciar al menos 2 bloques de cada tipo
- Sobrescribe accion() en BloqueCofre, BloqueTnt y BloqueHorno, mostrando distintos mensajes según el tipo de bloque.
- Sobrecarga colocar() para permitir colocar un bloque en diferentes orientaciones (por ejemplo, en el suelo o en la pared).
- Sobrescribe romper() en BloqueCofre, BloqueTnt y BloqueHorno, mostrando distintos mensajes según el tipo de bloque y que puede suceder al romperlos.

```

class Bloque:
    def __init__(self, nombre):
        self.nombre = nombre

    def accion(self):
        print(f"{self.nombre}: Realizando acción genérica del bloque.")

    def colocar(self, orientacion="suelo"):
        print(f"{self.nombre} colocado en el {orientacion}.")

    def romper(self):
        print(f"{self.nombre} fue roto sin efectos especiales.")

class BloqueCofre(Bloque):
    def __init__(self):
        super().__init__("Bloque Cofre")

    def accion(self):
        print("Abriendo el cofre... ¡Hay objetos dentro!")

    def romper(self):
        print("El cofre fue roto. ¡Los objetos se dispersaron!")

class BloqueTnt(Bloque):
    def __init__(self):
        super().__init__("Bloque TNT")

    def accion(self):
        print("Encendiendo la TNT... ¡Explosión inminente!")

```

```

    def romper(self):
        print("¡Boom! Rompiste la TNT y explotó.")

class BloqueHorno(Bloque):
    def __init__(self):
        super().__init__("Bloque Horno")

    def accion(self):
        print("Usando el horno... Cocinando ítems.")

    def romper(self):
        print("El horno fue roto. Se perdió el contenido cocinando.")

def main():
    bloques = [
        BloqueCofre(), BloqueCofre(),
        BloqueTnt(), BloqueTnt(),
        BloqueHorno(), BloqueHorno()
    ]

    for bloque in bloques:
        bloque.colocar()
        bloque.colocar("pared")
        bloque.accion()
        bloque.romper()

```

```
print("-" * 40)

if __name__ == "__main__":
    main()
```

```
Bloque Cofre colocado en el suelo.
Bloque Cofre colocado en el pared.
Abriendo el cofre... ¡Hay objetos dentro!
El cofre fue roto. ¡Los objetos se dispersaron!
```

```
Bloque Cofre colocado en el suelo.
Bloque Cofre colocado en el pared.
Abriendo el cofre... ¡Hay objetos dentro!
El cofre fue roto. ¡Los objetos se dispersaron!
```

```
Bloque TNT colocado en el suelo.
Bloque TNT colocado en el pared.
Encendiendo la TNT... ¡Explosión inminente!
¡Boom! Rompiste la TNT y explotó.
```

```
Bloque TNT colocado en el suelo.
Bloque TNT colocado en el pared.
Encendiendo la TNT... ¡Explosión inminente!
¡Boom! Rompiste la TNT y explotó.
```

```
Bloque Horno colocado en el suelo.
Bloque Horno colocado en el pared.
Usando el horno... Cocinando ítems.
El horno fue roto. Se perdió el contenido cocinando.
```

```
Bloque Horno colocado en el suelo.
Bloque Horno colocado en el pared.
Usando el horno... Cocinando ítems.
El horno fue roto. Se perdió el contenido cocinando.
```

