



# Práctica 1

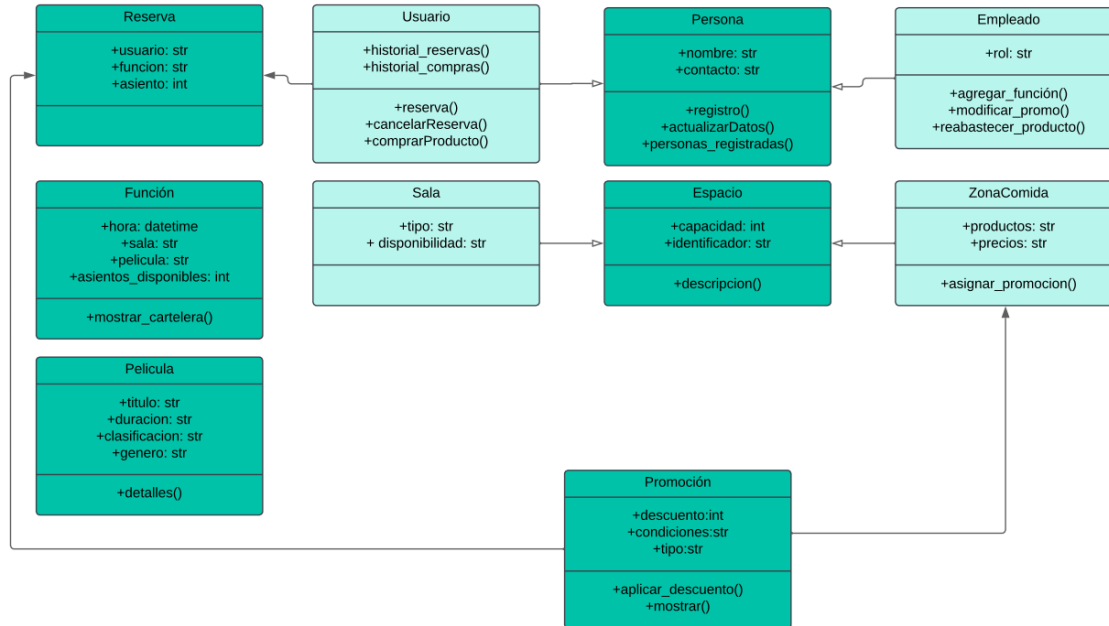
# Programación Avanzada

Alumna: Lidia Gizem Sánchez Montiel

Profesor: Jaime Alejandro Romero Sierra

12/02/2025

# Diagrama UML del programa: Sistema de reserva de un cine



## Justificación

Un programa que funcione como sistema de reservación de un cine puede ser una gran herramienta de gestión que automatice todas las acciones. Puede dar solución a problemas como el tiempo de reserva, ofrecer el servicio de manera online y minimizar errores de horarios.

# Clases, atributos y métodos

## Persona

### Métodos:

*registro()* -> Servirá para almacenar los datos de los usuarios.  
*actualizar\_datos()* -> Modificará los datos de los usuarios.  
*personas\_registradas()* -> Mostrará la información de cada usuario registrado.

### Atributos:

*nombre* -> Almacena el nombre del usuario.  
*correo* -> Almacena el contacto del usuario.

## Usuario (hereda de Persona)

### Métodos:

*reserva()* -> Le permite a los usuarios reservar una función.  
*cancelar\_reserva()* -> Eliminará la reserva realizada.  
*comprar\_producto()* -> Permite al usuario comprar un alimento.

### Atributos:

*historial\_compra* -> Almacena los datos de compra del usuario.  
*historial\_reserva* -> Almacena los datos de reserva del usuario.

## Empleado (hereda de Persona)

### Métodos:

*agregar\_función()* -> Servirá para asignar una función al usuario.  
*modificar\_promo()* -> Modificará la promoción para la función reabastecer\_producto() -> Permite agregar más productos

### Atributos:

*rol* -> Guarda el rol del empleado

## Espacio

### Métodos:

*descripción()* -> Mostrará los datos del espacio.

### Atributos:

*capacidad* -> guardará la capacidad del espacio.  
*identificador* -> Almacena el nombre del espacio.

## Sala (hereda de Espacio)

### Métodos:

*Los que hereda de Espacio.*

### Atributos:

*tipo* -> Indica el tipo de sala que es.  
*disponibilidad* -> Indica si la sala está libre u ocupada

## ZonaComida (hereda de Espacio)

### Métodos:

*asignar\_promoción()* -> Asigna una promoción al usuario.

### Atributos:

*productos* -> Almacena los productos de la zona.  
*precios* -> Guarda los precios de los productos.

## Reserva

### Atributos:

*usuario* -> Relaciona el usuario para la reserva.  
*funcion* -> Agrega la función para el usuario.  
*asiento* -> Hace relación con un asiento para el usuario.

## Función

### Métodos:

*mostrar\_cartelera()* -> Muestra las funciones.

### Atributos:

*hora* -> Guarda la hora de la función reservada.  
*sala* -> Guarda la sala de la función reservada.  
*película* -> Guarda la película de la función reservada.  
*asientos\_disponibles* -> Obtiene el número de asientos libres.

## Película

### Métodos:

*detalles()* -> Describe la película.

### Atributos:

*titulo, genero, duración y clasificación.*

## Promoción

### Métodos:

*aplicar\_escuento()* -> hace un descuento al usuario.

*mostrar()* -> muestra la promoción.

### Atributos:

**condiciones** -> guarda las condiciones de descuento.

**descuento** -> Guarda el descuento a aplicar.

**Tipo** -> Guarda el tipo de promoción.

# Código ejecutado

Cartelera del Cine:

Película: Intensamente 2 | Sala: Sala 1 | Hora: 15:00 | Asientos disponibles: 50

Película: El Conjuro | Sala: Sala 2 | Hora: 18:30 | Asientos disponibles: 30

Reserva realizada para 'Intensamente 2' en la sala Sala 1.

Reserva realizada para 'El Conjuro' en la sala Sala 2.

Reserva cancelada para 'Intensamente 2'.

La función de 'Iron Man' ya está en la cartelera y no se agregará nuevamente.

Cartelera del Cine:

Película: Intensamente 2 | Sala: Sala 1 | Hora: 15:00 | Asientos disponibles: 50

Película: El Conjuro | Sala: Sala 2 | Hora: 18:30 | Asientos disponibles: 25

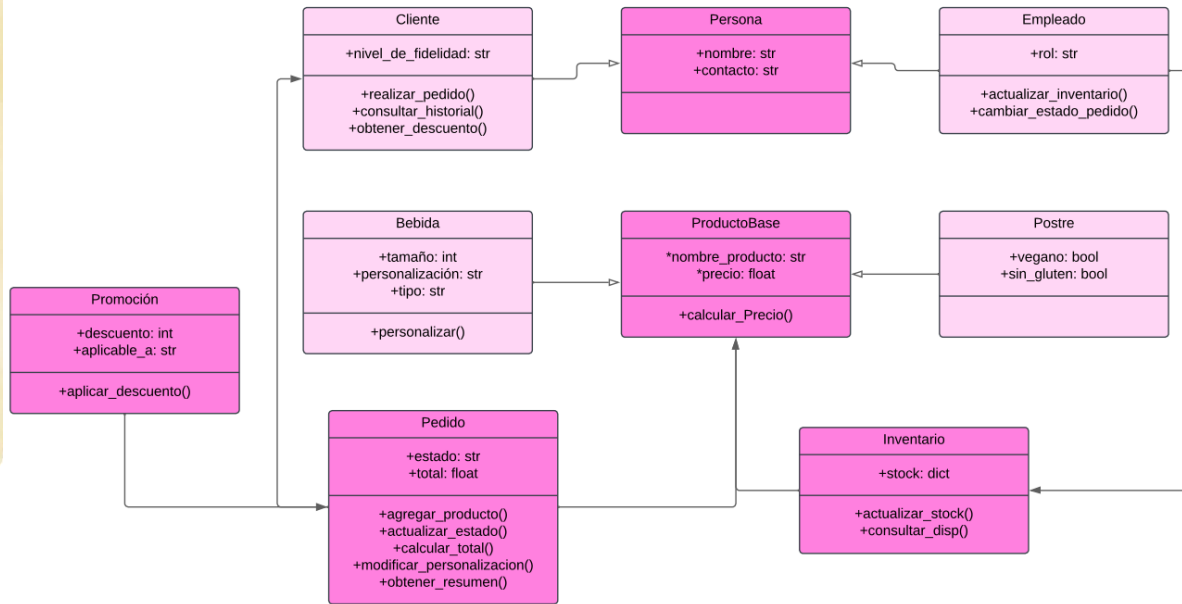
Película: Iron Man | Sala: Sala 1 | Hora: 22:00 | Asientos disponibles: 50

Promoción: 20% de descuento. Condiciones: Descuento para funciones después de las 22:00

Promoción: 30% de descuento. Condiciones: Descuento para funciones IMAX

Perla Morales compró 2 Palomitas(s) por un total de \$100.00

# Diagrama UML del programa: Gestión de una cafetería



## Justificación

Este programa permitirá tener un mejor control de los pedidos y del inventario que tiene la cafetería. Además, la aplicación de descuentos ayudaría a ganar más clientes. Se permitirá gestionar las personalizaciones de los pedidos para mejorar la atención al cliente.

# Clases, atributos y métodos

## Persona

### Atributos:

*nombre -> Almacena el nombre del cliente.  
contacto -> Almacena el contacto del cliente.*

## Cliente (hereda de Persona)

### Métodos:

*realizar\_pedido() -> El cliente puede ordenar su pedido.  
consultar:historial() -> Muestra los pedidos del cliente.  
obtener\_descuento() -> Permite al cliente acceder a la promoción.*

### Atributos:

*nivel\_de\_fidelidad -> Almacena el estatus del cliente.*

## Empleado (hereda de Persona)

### Métodos:

*actualizar\_inventario() -> Permite abastecer de productos.  
cambiar\_estado\_pedido() -> Modificará el estado del pedido.*

### Atributos:

*rol -> Guarda el rol del empleado*

## ProductoBase

### Métodos:

*calcular\_precio() -> Calcula el precio del pedido.*

### Atributos:

*nombre\_producto -> guarda el nombre del producto.  
precio -> Almacena el precio del producto.*

## Bebida (hereda de ProductoBase)

### Métodos:

*personalizar() -> Permite agregar especificaciones.*

### Atributos:

*tamaño -> Indica el tamaño del producto.  
personalización -> Indica las especificaciones de la bebida.  
tipo -> Indica el tipo de bebida.*

## Postre (hereda de ProductoBase)

### Atributos:

*vegano -> Indica si es vegano el postre.  
sin\_gluten -> Indica si el postre es sin gluten.*

## Promoción

### Métodos:

*aplicar\_descuento() -> Aplica el descuento al pedido.*

### Atributos:

*descuento -> Indica el descuento de la promoción.  
aplicable\_a -> Indica a qué puede aplicarse el descuento..*

## Pedido

### Métodos:

*agregar\_producto() -> Agrega un producto al pedido.  
actualizar\_estado() -> Modifica el estado del pedido.  
calcular\_total() -> Indica el total a pagar.  
modificar\_personalización() -> Puede modificar los detalles.  
obtener\_resumen() -> Muestra los pedidos.*

### Atributos:

*estado -> indica el estado del pedido.  
total -> Indica el total a pagar.*

## Inventario

### Métodos:

*actualizar\_stock() ->  
Modifica la cantidad de  
productos..*

*Consultar\_disp() -> Indica si  
el producto está disponible.*

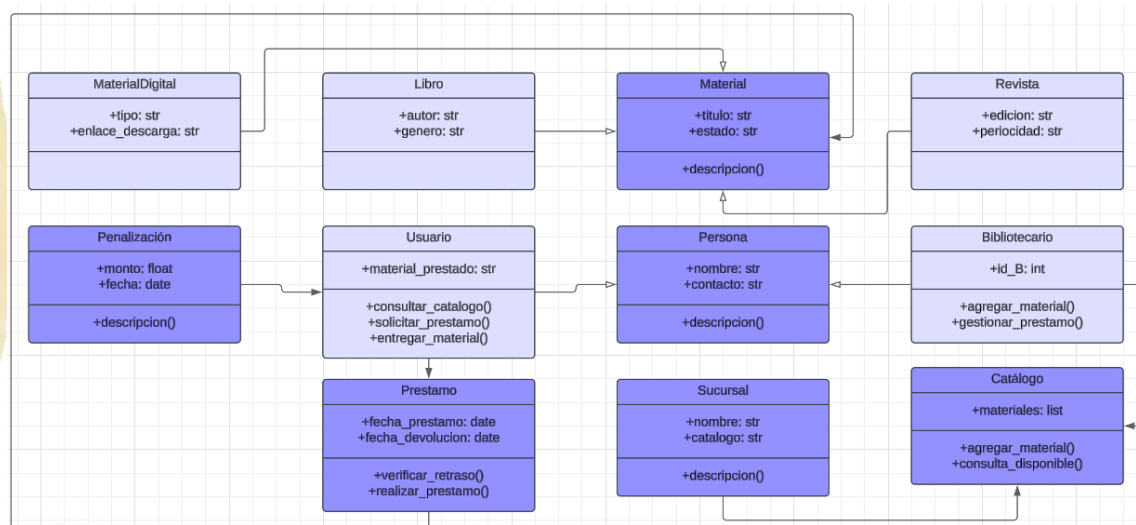
### Atributos:

**stock**-> indica los  
productos del inventario..

# Código ejecutado

```
Inventario inicializado: {'Café Americano': 5, 'Latte': 3, 'Brownie': 4}
Bebida Latte personalizada: leche = almendra
Bebida Latte personalizada: azúcar = sin azúcar
Pedido creado para Lidia Montiel
Café Americano: Disponible
Stock Café Americano: 4
Café Americano agregado al Pedido
Brownie: Disponible
Stock Brownie: 3
Brownie agregado al Pedido
Pedido creado para Daniel Sánchez
Latte: Disponible
Stock Latte: 2
Latte agregado al Pedido
Brownie: Disponible
Stock Brownie: 2
Brownie agregado al Pedido
Descuento aplicado (Cliente Leal): 10.0%
Descuento aplicado (Café Adicto): 15.0%
Promoción creada: 10% en pedido
Total del Pedido: $70.00
Descuento aplicado: Nuevo total $63.00
Lidia Montiel no tiene pedidos.
Daniel Sánchez no tiene pedidos.
Estado del Pedido cambiado a: en preparación
...
Estado del Pedido cambiado a: entregado
Adrián Morales cambió el estado del Pedido a entregado
Bebida Latte personalizada: extra espresso = doble
Personalización modificada: Latte (extra espresso = doble)
```

# Diagrama UML del programa: Gestión de una Biblioteca



## Justificación

Este programa se ajusta a las necesidades de una biblioteca. Puede hacer préstamos a los usuarios, hacer un seguimiento para evitar retrasos y mostrar el catálogo de los materiales disponibles en la biblioteca.



# Clases, atributos y métodos

## Material

### Métodos:

*descripción()* -> Muestra los datos del material.

### Atributos:

*título* -> Guarda el nombre del material.

*estado* -> Guarda el estado del material.

## Libro (hereda de Material)

### Atributos:

*autor* -> Almacena el nombre del libro.

*genero* -> Indica el género del libro.

## MaterialDigital (hereda de Material)

### Atributos:

*tipo* -> Guarda el tipo de material digital

*enlace\_descarga* -> Guarda el enlace del material.

## Revista (hereda de Material)

### Atributos:

*edición* -> guarda el nombre de la edición.

*periodicidad* -> Almacena la periodicidad de la revista.

## Persona

### Métodos:

*descripción()* -> Muestra los datos de la persona.

### Atributos:

*nombre* -> Guarda el nombre de la persona.

*contacto* -> Guarda el contacto de la persona.

## Usuario (hereda de Persona)

### Métodos:

*consultar\_catálogo()* -> Muestra los materiales.

*solicitar\_préstamo()* -> Permite solicitar un préstamo de material.

*entregar\_material()* -> Permite entregar el material de vuelta.

### Atributos:

*material\_prestado* -> Indica qué material solicitó.

## Bibliotecario (hereda de Persona)

### Métodos:

*agregar\_material()* -> Agrega un material disponible.

*gestionar\_préstamo()* -> Puede hacer préstamos.

### Atributos:

*id\_B* -> Indica el identificador del bibliotecario.

## Penalización

### Métodos:

*descripción()* -> Muestra los detalles de la penalización.

### Atributos:

*monto* -> Indica la cantidad de penalización.

*fecha* -> Fecha límite del préstamo.

## Préstamo

### Métodos:

*verificar\_retraso()* -> Verifica si está retrasado el usuario.

*realizar\_préstamo()* -> Permite facilitar un préstamo.

### Atributos:

*fecha\_préstamo* -> Fecha donde inicia el préstamo.

*fecha\_devolución* -> Indica la fecha de devolución.

## Sucursal

### Métodos:

*descripción()* -> Describe los detalles de la sucursal.

### Atributos:

**Nombre** -> indica el nombre de la sucursal.

**catálogo** -> Indica qué materiales tiene.

## Catálogo

### Métodos:

*agregar\_material()* -> Agrega un material al catálogo.

*Consulta\_disponible()* -> Verifica si el material está disponible.

### Atributos:

**Materiales** -> Indica qué materiales hay.

# Código ejecutado

Material disponible en catálogo:

```
['Título: Veinte Mil Leguas de Viaje Submarino, Estado: disponible', 'Título: National Geographic, Estado: disponible']
```

Después del préstamo:

```
['Título: National Geographic, Estado: disponible']
```

Estado del préstamo:

No hay retraso :)

Entregando material:

El usuario: Lidia ha entregado el material con éxito.

Después de la entrega:

```
['Título: Veinte Mil Leguas de Viaje Submarino, Estado: disponible', 'Título: National Geographic, Estado: disponible']
```