



PRÁCTICA UNIDAD 1

Programación avanzada

Ricardo Xochitiotzi Flores

10/02/2024

M.C. Jaime Alejandro Romero Sierra



Programa 1.- Sistema de Reservas para un Cine

El sistema de reservas es esencial para gestionar de manera eficiente las funciones, salas, reservas y promociones en un cine. La organización de clases y herencias permite modelar adecuadamente las relaciones entre usuarios, empleados, salas y películas, haciendo el sistema más flexible y escalable.

- **Clases Persona, Usuario y Empleado:** Definir una clase base Persona con clases derivadas como Usuario y Empleado facilita la organización de roles, funcionalidades y restricciones que dependen de si son clientes o personal del cine. Esto optimiza la gestión de permisos, reservas y roles específicos (por ejemplo, que un taquillero pueda gestionar entradas o un administrador pueda añadir nuevas películas).
- **Clases Espacio, Sala y ZonaComida:** Facilitan la gestión de los distintos tipos de salas y zonas dentro del cine. Heredar de Espacio permite que se mantenga la flexibilidad para añadir otros tipos de espacios, sin necesidad de reestructurar todo el sistema.
- **Gestión de reservas y funciones:** Las clases Reserva y Funcion permiten asociar a un usuario con una película en un horario y sala específicos, manteniendo un control sobre la disponibilidad de asientos y las promociones aplicadas.

```

usuario1 = Usuario("Carlos Pérez", "carlos@email.com")
empleado1 = Empleado("Ana Gómez", "ana@email.com", "Administrador")

pelicula1 = Pelicula("Interstellar", 169, "PG-13", "Ciencia Ficción")
pelicula2 = Pelicula("Inception", 148, "PG-13", "Ciencia Ficción")

sala1 = Sala("Sala 1", "IMAX", 100)
sala2 = Sala("Sala 2", "3D", 80)

funcion1 = Funcion(pelicula1, sala1, "20:00")
funcion2 = Funcion(pelicula2, sala2, "18:00")

promocion1 = Promocion("Descuento 20%", 0.20)
promocion2 = Promocion("Descuento 10%", 0.10)

reserva1 = Reserva(usuario1, funcion1, 2, promocion1)
reserva2 = Reserva(usuario1, funcion2, 3)

zona_comida = ZonaComida("Snack Bar")
zona_comida.agregar_producto("Palomitas", 5.0)
zona_comida.agregar_producto("Refresco", 3.0)

print("\n===== SISTEMA DE RESERVAS DE CINE FINALIZADO =====\n")

```

✓ 0.0s

Python



```
-- Persona creada --
Nombre: Carlos Pérez | Email: carlos@email.com
-----
[ Usuario registrado ] Nombre: Carlos Pérez

-- Persona creada --
Nombre: Ana Gómez | Email: ana@email.com
-----
[ Empleado creado ] Nombre: Ana Gómez | Rol: Administrador
-- Película creada --
Título: Interstellar | Duración: 169 min
-----
-- Película creada --
Título: Inception | Duración: 148 min
-----
-- Espacio creado --
Nombre: Sala 1
-----
[ Sala creada ] Nombre: Sala 1 | Tipo: IMAX | Capacidad: 100
-- Espacio creado --
Nombre: Sala 2
-----
[ Sala creada ] Nombre: Sala 2 | Tipo: 3D | Capacidad: 80
-- Función creada --
Película: Interstellar | Sala: Sala 1 | Horario: 20:00
-----
-- Función creada --
Película: Inception | Sala: Sala 2 | Horario: 18:00
-----
```

```
-----
[ Promoción creada ] Descuento 20% - Descuento: 20.0%
[ Promoción creada ] Descuento 10% - Descuento: 10.0%
[ Reserva realizada ] Usuario: Carlos Pérez | Película: Interstellar | Horario: 20:00
[ Reserva realizada ] Usuario: Carlos Pérez | Película: Inception | Horario: 18:00
-- Espacio creado --
Nombre: Snack Bar
-----
[ Producto agregado ] Palomitas - Precio: $5.0
[ Producto agregado ] Refresco - Precio: $3.0
```

```
=====
SISTEMA DE RESERVAS DE CINE FINALIZADO
=====
```



Programa 2.- Gestión de Pedidos en una Cafetería

La gestión de pedidos en una cafetería requiere un sistema organizado que permita controlar el inventario, personalizar pedidos y aplicar descuentos. Usar programación orientada a objetos permite estructurar el sistema de forma eficiente y fácil de mantener.

- **Clases Persona, Cliente y Empleado:** Al igual que en el sistema del cine, la base común Persona ayuda a organizar roles y permisos, lo que permite asignar distintas funcionalidades a cada tipo de usuario (clientes realizando pedidos y empleados gestionando el servicio).
- **Clases ProductoBase, Bebida, y Postre:** Permiten manejar productos de manera general (como la clase base ProductoBase) y luego especializarse para las bebidas y postres con propiedades específicas. Esto permite agregar nuevas categorías de productos de manera sencilla.
- **Sistema de inventario y pedidos:** La clase Inventario asegura que el sistema gestione las existencias correctamente, y que no se puedan hacer pedidos con ingredientes agotados. Además, la clase Pedido maneja el estado de los pedidos y facilita la actualización de su estado durante el proceso de preparación.

```
cliente2 = Cliente("Juan Pérez", "juan@email.com")

empleado2 = Empleado("Sofía Ramírez", "sofia@email.com", "Mesera")

inventario_completo = Inventario()
inventario_completo.agregar_ingredientes("Café", 20)
inventario_completo.agregar_ingredientes("Leche", 20)
inventario_completo.agregar_ingredientes("Azúcar", 20)
inventario_completo.agregar_ingredientes("Chocolate", 20)
inventario_completo.agregar_ingredientes("Harina", 20)
inventario_completo.agregar_ingredientes("Mantequilla", 20)
inventario_completo.agregar_ingredientes("Huevo", 20)

bebida3 = Bebida("Capuchino", 4.0, "Grande", "Caliente", ["Café", "Leche", "Azúcar"])
bebida4 = Bebida("Té Verde", 3.0, "Mediano", "Caliente", ["Azúcar"])
postre2 = Postre("Croissant", 2.5, False, False)
postre3 = Postre("Galleta Vegana", 3.0, True, False)

pedido3 = Pedido(cliente2, [bebida3, postre2], inventario_completo)
pedido4 = Pedido(cliente2, [bebida4, postre3], inventario_completo)

promo2 = Promocion("Descuento del 10% en postres", 0.10)

print("\n===== PRUEBA COMPLETA DEL SISTEMA FINALIZADA\n=====")
```



```
-- Persona creada --
Nombre: Juan Pérez | Email: juan@email.com
-----
[ Cliente registrado ] Nombre: Juan Pérez

-- Persona creada --
Nombre: Sofía Ramírez | Email: sofia@email.com
-----
[ Empleado creado ] Nombre: Sofía Ramírez | Rol: Mesera
-- Inventario creado --
[ Ingrediente agregado ] Café: 20 unidades
[ Ingrediente agregado ] Leche: 20 unidades
[ Ingrediente agregado ] Azúcar: 20 unidades
[ Ingrediente agregado ] Chocolate: 20 unidades
[ Ingrediente agregado ] Harina: 20 unidades
[ Ingrediente agregado ] Mantequilla: 20 unidades
[ Ingrediente agregado ] Huevo: 20 unidades
-- Producto creado --
Nombre: Capuchino | Precio: $4.0
-----
[ Bebida creada ] Nombre: Capuchino | Tamaño: Grande | Tipo: Caliente | Opciones: Café, Leche, Azúcar
-- Producto creado --
Nombre: Té Verde | Precio: $3.0
-----
[ Bebida creada ] Nombre: Té Verde | Tamaño: Mediano | Tipo: Caliente | Opciones: Azúcar
-- Producto creado --
Nombre: Croissant | Precio: $2.5
-----
[ Postre creado ] Nombre: Croissant | Vegano: False | Sin Gluten: False
-- Producto creado --
```

```
-- Producto creado --
Nombre: Galleta Vegana | Precio: $3.0
-----
[ Postre creado ] Nombre: Galleta Vegana | Vegano: True | Sin Gluten: False
[ Ingrediente usado ] Leche - Restante: 19
[ Ingrediente usado ] Azúcar - Restante: 20
[ Ingrediente usado ] Harina - Restante: 19
[ Ingrediente usado ] Chocolate - Restante: 19
[ Pedido creado ] Cliente: Juan Pérez | Total: $6.5 | Estado: Pendiente
[ Ingrediente usado ] Leche - Restante: 18
[ Ingrediente usado ] Azúcar - Restante: 20
[ Ingrediente usado ] Harina - Restante: 18
[ Ingrediente usado ] Chocolate - Restante: 18
[ Pedido creado ] Cliente: Juan Pérez | Total: $6.0 | Estado: Pendiente
[ Promoción creada ] Descuento del 10% en postres - Descuento: 10.0%
```

```
=====
PRUEBA COMPLETA DEL SISTEMA FINALIZADA
=====
```



Programa 3.- Biblioteca Digital

El sistema de biblioteca digital es necesario para organizar y gestionar de manera eficiente los materiales disponibles en diferentes sucursales, los préstamos, y las penalizaciones por retrasos. La utilización de clases permite mantener una estructura clara y ordenada en el manejo de materiales y usuarios.

- **Clases Material, Libro, Revista y MaterialDigital:** Estas clases modelan los diferentes tipos de materiales que pueden estar disponibles en la biblioteca. Usar una clase base como Material facilita la adición de otros tipos de materiales en el futuro sin alterar la estructura básica del sistema.
- **Clases Persona, Usuario, y Bibliotecario:** Al igual que en los otros sistemas, la clase base Persona facilita la diferenciación entre los roles de usuario y bibliotecario, asegurando que los usuarios puedan consultar el catálogo mientras que los bibliotecarios puedan gestionarlo.
- **Clases Sucursal, Préstamo y Penalización:** Las clases Sucursal permiten gestionar varias ubicaciones de la biblioteca, y las clases de préstamo y penalización permiten controlar la relación entre los usuarios y los materiales prestados, así como las multas por retrasos.

```
from datetime import datetime, timedelta

libro1 = Libro("Python para Todos", "Juan Pérez", "Tecnología")
libro2 = Libro("La Magia de la Matemática", "Ana Gómez", "Ciencia")
revista1 = Revista("Revista de Ciencia", "Edición 10", "Mensual")
material_digital1 = MaterialDigital("Curso de POO", "PDF", "www.curso.com/poo")

sucursal1 = Sucursal("Sucursal A")
sucursal2 = Sucursal("Sucursal B")

sucursal1.agregar_material(libro1)
sucursal1.agregar_material(revista1)
sucursal2.agregar_material(libro2)
sucursal2.agregar_material(material_digital1)

catalogo = Catalogo()
catalogo.agregar_material(libro1)
catalogo.agregar_material(libro2)
catalogo.agregar_material(revista1)
catalogo.agregar_material(material_digital1)

usuario1 = Usuario("Carlos Ruiz", 25)
usuario2 = Usuario("Laura Sánchez", 30)
bibliotecario1 = Bibliotecario("Lucía Pérez", 35)

bibliotecario1.agregar_material(Material("Ciencia en el siglo XXI", "disponible"), sucursal1)
bibliotecario1.transferir_material(libro2, sucursal2, sucursal1)
```



```

print("\n=== Consulta de Catálogo por Usuario1 ===")
for material in usuario1.consultar_catalogo(catalogo):
    print(f"* {material}")

print("\n=== Solicitudes de Materiales ===")
usuario1.solicitar_material(libro1)
usuario2.solicitar_material(revista1)

print(f"\n=== Materiales Prestados por {usuario1.nombre} ===")
for material in usuario1.materiales_prestados:
    print(f" - {material}")

fecha_prestamo = datetime.now()
fecha_devolucion = fecha_prestamo + timedelta(days=7)
prestamo1 = Prestamo(usuario1, libro1, fecha_prestamo, fecha_devolucion)

usuario1.devolver_material(libro1)

penalizacion1 = Penalizacion(usuario2, 10, "Retraso en la devolución de revista")
penalizacion1.aplicar_penalizacion()

print("\n=== Materiales en Sucursal A después de la Transferencia ===")
for material in sucursal1.catalogo:
    print(f" - {material}")

print("\n=== Materiales en el Catálogo ===")
for material in catalogo.materiales:
    print(f" - {material}")
print("\n*****")

```

✓ 0.0s

Python

```

=== Consulta de Catálogo por Usuario1 ===
*
    ***** Libro *****
    Título: Python para Todos
    Autor: Juan Pérez
    Género: Tecnología
    Estado: disponible
    *****

*
    ***** Libro *****
    Título: La Magia de la Matemática
    Autor: Ana Gómez
    Género: Ciencia
    Estado: disponible
    *****

*
    ##### Revista #####
    Título: Revista de Ciencia
    Edición: Edición 10
    Periodicidad: Mensual
    Estado: disponible
    #####

*
    ----- Material Digital -----
    Título: Curso de POO
    Tipo de Archivo: PDF

```



```

=== Solicitudes de Materiales ===

=== Materiales Prestados por Carlos Ruiz ===
-
    ***** Libro *****
    Título: Python para Todos
    Autor: Juan Pérez
    Género: Tecnología
    Estado: prestado
    *****

** Penalización **: $10 aplicada a Laura Sánchez por: Retraso en la devolución de revista **

```

```

=== Materiales en Sucursal A después de la Transferencia ===

```

```

-
    ***** Libro *****
    Título: Python para Todos
    Autor: Juan Pérez
    Género: Tecnología
    Estado: disponible
    *****

-
    ##### Revista #####
    Título: Revista de Ciencia
    Edición: Edición 10
    Periodicidad: Mensual
    Estado: prestado
    #####

- *-* Ciencia en el siglo XXI (disponible) *-*
-
    ***** Libro *****
    Título: La Magia de la Matemática
    Autor: Ana Gómez
    Género: Ciencia
    Estado: disponible
    *****

```

```

=== Materiales en el Catálogo ===

```

```

-
    ***** Libro *****
    Título: Python para Todos
    Autor: Juan Pérez
    Género: Tecnología
    Estado: disponible
    *****

-
    ***** Libro *****
    Título: La Magia de la Matemática
    Autor: Ana Gómez
    Género: Ciencia
    Estado: disponible
    *****

-
    ##### Revista #####
    Título: Revista de Ciencia
    Edición: Edición 10
    Periodicidad: Mensual
    Estado: prestado
    #####

```



----- Material Digital -----

Título: Curso de POO

Tipo de Archivo: PDF

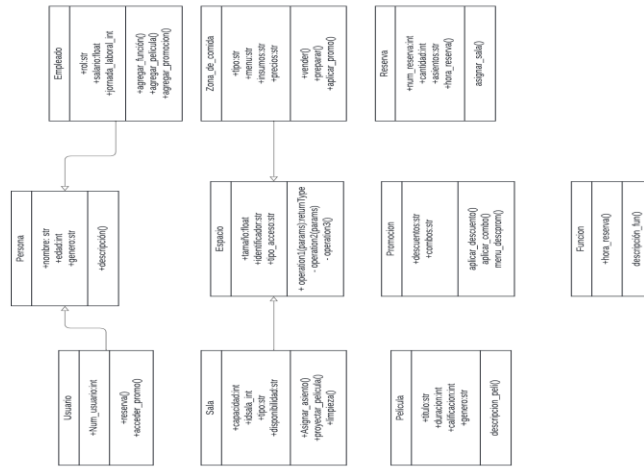
Enlace de Descarga: www.curso.com/poo

Estado: disponible

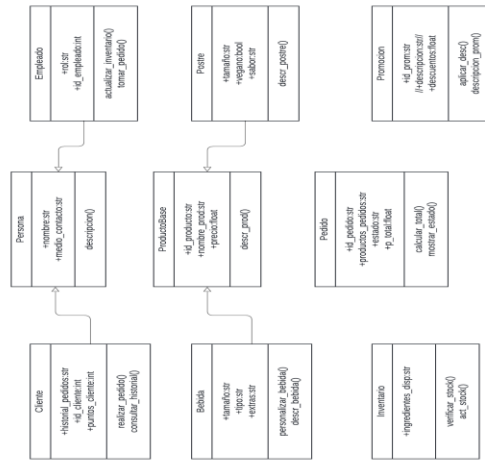


Diagramas UML

Sistema de reservas para un cine



Gestión de pedidos de una cafetería



Gestión de una biblioteca digital

