



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Práctica unidad 1 Programación Avanzada

Asignatura: PROGRAMACION AVANZADA

Carrera: INGENIERIA EN CIENCIA DE DATOS

Nombre del alumno: JOANA MONTSERRAT CARMONA NOGUEZ

Nombre del profesor: JAIME A. ROMERO SIERRA

Horario de la materia: LUNES, MIERCOLES Y VIERNES

No. De cuenta: 202466401

Fecha de entrega: 11 DE FEBRERO DE 2025

Sistema de Gestión para un Cine

- Justificacion

Este programa es necesario para organizar y gestionar los procesos de un cine, que incluyen la administración de películas, la gestión de asientos y la reserva de boletos. Este sistema beneficia tanto a los empleados del cine como a los clientes al permitir una búsqueda rápida de películas disponibles, facilitar la selección de asientos por parte de los clientes, automatizar la gestión de boletos y evitar errores al asignar asientos. La implementación de un sistema así optimiza la experiencia del cliente y mejora la operación interna del cine, asegurando un servicio eficiente.

- Clases a ocupar, describiendo atributos y métodos necesarios

1. Pelicula

- **Atributos:**

- titulo: Título de la película.
- genero: Género de la película
- duracion: Duración en minutos.
- clasificacion: Clasificación por edades

- **Métodos:**

- mostrar_detalle(): Muestra el título, género, duración y clasificación de la película.

2. Sala

- **Atributos:**

- numero: Número de la sala.
- asientos: Diccionario o matriz que representa el estado de cada asiento
- pelicula: Película proyectada en la sala.

- **Métodos:**

- mostrar_asientos(): Muestra el estado actual de los asientos.
- asignar_pelicula(pelicula): Asocia una película a la sala.
- reservar_asiento(fila, columna): Marca un asiento como ocupado.

3. Cliente

- **Atributos:**

- nombre: Nombre del cliente.
- correo: Correo del cliente.
- reservas: Lista de reservas realizadas.

- **Métodos:**

- reservar_boleto(sala, fila, columna): Permite reservar un asiento en una sala específica.
- mostrar_reservas(): Muestra las reservas realizadas.

4. Boleto

- **Atributos:**

- pelicula: Película asociada al boleto.
- sala: Sala donde se proyecta la película.
- asiento: Asiento reservado.
- precio: Precio del boleto.

- **Métodos:**

- mostrar_detalle(): Muestra la información del boleto.

5. Cine

- **Atributos:**

- nombre: Nombre del cine.

- salas: Lista de salas disponibles.

- **Métodos:**

- agregar_sala(sala): Añade una nueva sala al cine.
- buscar_pelicula(titulo): Busca una película en las salas disponibles.
- mostrar_cartelera(): Muestra las películas proyectadas y sus horarios.

```
#Uso
pelicula1 = Pelicula("Percy Jackson", "Ciencia Ficción", 136)
pelicula2 = Pelicula("De todos los chicos de los que me enamore", "Drama/Romance", 195)

sala1 = Sala(100, "Sala 4", "4D")
sala2 = Sala(50, "Sala 5", "VIP")

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

usuario1 = Usuario("Montserrat Carmona", "joanamontse11@gmail.com")
empleado1 = Empleado("Rosa Gonazalez", "rosa.gonzalez@gmail.com", "Gerente")

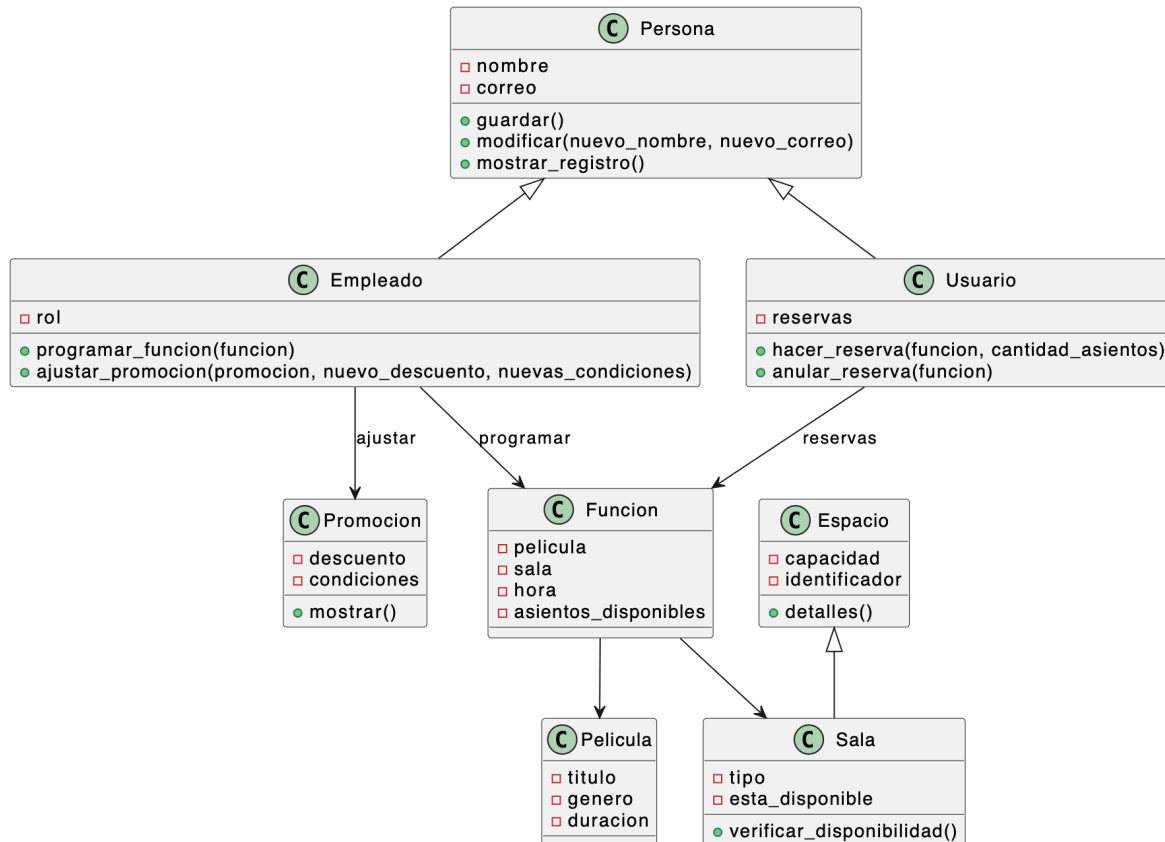
usuario1.guardar()
empleado1.guardar()

usuario1.hacer_reserva(funcion1, 3)
usuario1.anular_reserva(funcion1)

promocion1 = Promocion(20, "Válido de lunes a sabado.")
promocion1.mostrar()
empleado1.ajustar_promocion(promocion1, 30, "Válido antes de las 4 PM.")

Persona.mostrar_registro()
```

```
===== RESTART: /Users/mc/Documents/Cine_JMCN.py =====
Montserrat Carmona ha sido añadida al sistema con el correo joanamontse11@gmail.com.
Rosa Gonazalez ha sido añadida al sistema con el correo rosa.gonzalez@gmail.com.
Reserva realizada para 'Percy Jackson' en la sala Sala 4.
Reserva para 'Percy Jackson' cancelada.
Promoción: 20% de descuento. Válido de lunes a sabado.
Promoción actualizada: 30% de descuento. Válido antes de las 4 PM.
Listado de personas registradas:
Nombre: Montserrat Carmona, Correo: joanamontse11@gmail.com
Nombre: Rosa Gonazalez, Correo: rosa.gonzalez@gmail.com
|
```



Gestión de Pedidos en una Cafetería

- Justificación

Este programa es necesario para facilitar el proceso de toma de pedidos y manejo del inventario en una cafetería. Automatiza tareas como personalización de pedidos por parte de los clientes, control del inventario, evitando desabastos o excesos, registro de promociones aplicadas a productos específicos.

La digitalización de estos procesos mejora la eficiencia operativa, reduce errores humanos y aumenta la satisfacción del cliente.

- Clases a ocupar, describiendo atributos y métodos necesarios

1. **Persona**

- **Atributos:**

- nombre: Nombre del cliente o empleado.
- correo: Dirección de correo del cliente o empleado.

- **Métodos:**

- mostrar_informacion(): Muestra el nombre y correo de la persona.

2. **Cliente** (hereda de Persona)

- **Atributos:**

- pedido: Lista de productos seleccionados.

- **Métodos:**

- agregar_al_pedido(producto): Añade un producto al pedido.
- consultar_cuenta(): Muestra el costo total del pedido.

3. **Empleado** (hereda de Persona)

- **Atributos:**

- puesto: Puesto del empleado (barista, cajero, etc.).

- **Métodos:**

- registrar_pedido(cliente): Procesa el pedido del cliente.
- actualizar_inventario(): Descuenta del inventario los productos vendidos.

4. **ProductoBase**

- **Atributos:**

- nombre: Nombre del producto.

- precio: Precio del producto.

- **Métodos:**

- mostrar_detalle(): Muestra el nombre y precio del producto.

5. **Bebida** (hereda de ProductoBase)

- **Atributos:**

- tamaño: Tamaño de la bebida (chico, mediano, grande).

- **Métodos:**

- mostrar_detalle(): Incluye el tamaño en los detalles.

6. **Postre** (hereda de ProductoBase)

- **Atributos:**

- sabor: Sabor del postre (chocolate, vainilla, etc.).

- **Métodos:**

- mostrar_detalle(): Incluye el sabor en los detalles.

#Uso

```
almacen = Inventario()
almacen.actualizar("Café", 15)
almacen.actualizar("Leche", 10)

cliente = Cliente("Montserrat Carmona", "joanamontsell@gmail.com")
cliente.agregar_registro()

empleado = Empleado("Gloria Ortega", "ortega.gloria@gmail.com", "Barista")
empleado.agregar_registro()

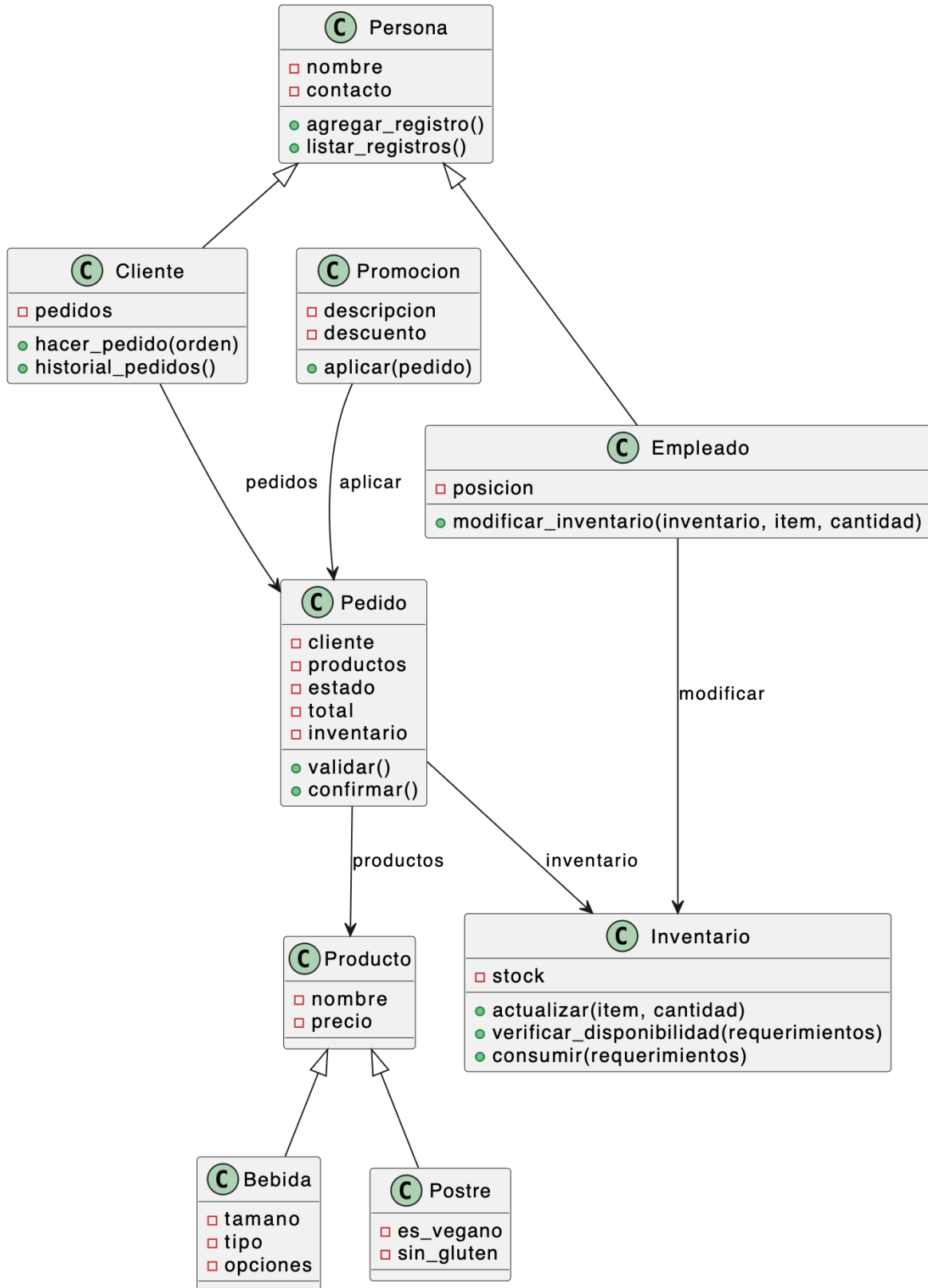
cafe = Bebida("Café Americano", 2.5, "Mediano", "Caliente", ["Sin azúcar"])
pastel = Postre("Cheesecake", 3.5, sin_gluten=True)

orden = Pedido(cliente, [cafe, pastel], almacen)
cliente.hacer_pedido(orden)

promo = Promocion("Descuento del 15%", 15)
promo.aplicar(orden)

cliente.historial_pedidos()
Persona.listar_registros()
```

```
===== RESTART: /Users/mc/Documents/Cafeteria_JMCN.py =====
Inventario actualizado: Café -> 15 unidades.
Inventario actualizado: Leche -> 10 unidades.
Registro exitoso: MMontserrat Carmona.
Registro exitoso: Gloria Ortega.
Pedido confirmado. Total: $6.0.
Promoción aplicada: Descuento del 15%. Total final: $5.1.
Historial de MMontserrat Carmona:
- Productos: [<__main__.Bebida object at 0x105747230>, <__main__.Postre object at 0x105747380>], Estado: En preparación, Total: $5.1
Listado de personas registradas:
- MMontserrat Carmona, Contacto: joanamontsell@gmail.com
- Gloria Ortega, Contacto: ortega.gloria@gmail.com
```

Biblioteca Digital

- Justificacion

Un sistema de biblioteca digital es imprescindible para la gestión moderna de bibliotecas, especialmente aquellas con múltiples sucursales. Este programa permite gestionar préstamos y devoluciones de forma eficiente, organizar un catálogo con diferentes tipos de materiales (libros, revistas y materiales digitales), administrar penalizaciones para fomentar el cumplimiento de plazos, facilitar la búsqueda y transferencia de materiales entre sucursales.

Con esto, se optimiza el uso de los recursos y se brinda un mejor servicio a los usuarios.

- Clases a ocupar, describiendo atributos y métodos necesarios

Biblioteca Digital

1. Material

- **Atributos:**

- titulo: Título del material.
- estado: Estado del material (disponible/prestado).

- **Métodos:**

- prestar(): Cambia el estado del material a “prestado”.
- devolver(): Cambia el estado del material a “disponible”.

2. Libro (hereda de Material)

- **Atributos:**

- autor: Nombre del autor.

- genero: Género literario del libro.

- **Métodos:**

- mostrar_detalle(): Muestra los detalles del libro.

3. **Revista** (hereda de Material)

- **Atributos:**

- edicion: Edición de la revista.
- periodicidad: Frecuencia de publicación.

- **Métodos:**

- mostrar_detalle(): Muestra los detalles de la revista.

4. **Usuario** (hereda de Persona)

- **Atributos:**

- materiales_prestados: Lista de materiales prestados.
- penalizaciones: Multas acumuladas por el usuario.

- **Métodos:**

- consultar_catalogo(catalogo): Muestra los materiales del catálogo.
- solicitar_prestamo(material, bibliotecario): Solicita un préstamo de material.

5. **Bibliotecario** (hereda de Persona)

- **Atributos:**

- sucursal: Sucursal donde trabaja.

- **Métodos:**

- procesar_prestamo(usuario, material): Autoriza y registra un préstamo.
- procesar_devolucion(usuario, material): Registra la devolución del material.

6. **Sucursal**

- **Atributos:**

- nombre: Nombre de la sucursal.
- catalogo: Lista de materiales disponibles.

- **Métodos:**

- agregar_material(material): Añade un material al catálogo.
- transferir_material(material, otra_sucursal): Transfiere un material entre sucursales.

```
#Uso
biblioteca_central = Biblioteca("Central Universitaria")
biblioteca_computacion = Biblioteca("Facultad de Computacion")

encargado = Encargado("Rogelio Lopez", "ro.lopez@alm.bua.mx", biblioteca_central)
lector = Lector("Montserrat Carmona", "cn202466401@alm.buap.mx")

libro = Libro("Programacion General", "Resnick & Halliday", "Computacion")
revista = Revista("National Geographic", "Febrero 2025", "Mensual")
recurso_digital = RecursoDigital("Manual de Python", "PDF", "https://uni.edu/python.pdf")

encargado.agregar_recurso(libro)
encargado.agregar_recurso(revista)
lector.solicitar_recurso(libro, encargado)
lector.devolver_recurso(libro, encargado)

registro_multa = RegistroMultas(lector, 3)
registro_multa.mostrar_multa()

biblioteca_central.transferir_recurso(libro, biblioteca_computacion)

===== RESTART: /Users/mc/Documents/Bibli
Montserrat Carmona ha tomado prestado: Programacion General
Montserrat Carmona ha devuelto: Programacion General
Montserrat Carmona tiene una multa de $15 por 3 días de retraso.
'Programacion General' ha sido transferido a Facultad de Computacion.
,
```

