

# Práctica Unidad 1 Programación Avanzada

## Sandro Farid Díaz Rodríguez

### **Justificación e Importancia del Sistema de Reservas para un Cine**

#### **Justificación**

En la actualidad, la automatización de procesos es esencial para mejorar la eficiencia y experiencia del cliente en distintos sectores, incluyendo el entretenimiento. Un sistema de reservas para un cine optimiza la gestión de funciones, boletos y promociones, garantizando una administración efectiva de los recursos disponibles.

La implementación de este sistema responde a la necesidad de ofrecer una solución integral que permita a los usuarios adquirir sus boletos de manera rápida y segura, evitando largas filas y reduciendo la posibilidad de errores humanos. Además, facilita la labor del personal al gestionar inventarios de boletos, promociones y programaciones de funciones sin inconvenientes.

Desde una perspectiva empresarial, este sistema permite un control preciso de la ocupación de las salas, mejorando la planificación de funciones y optimizando el uso del espacio. Además, la integración de promociones ayuda a incrementar la asistencia y fidelización de los clientes, generando mayores ingresos para el cine.

#### **Importancia**

El desarrollo de este sistema de reservas es crucial por diversas razones:

1. **Mejora la experiencia del usuario:** Permite que los clientes realicen reservas de manera sencilla y puedan seleccionar asientos con anticipación, asegurando su comodidad y preferencia.

2. **Eficiencia operativa:** Optimiza la gestión del cine al automatizar el control de funciones, disponibilidad de salas y asignación de asientos, reduciendo la carga de trabajo del personal.
3. **Administración efectiva del inventario:** Evita la sobreventa o la asignación errónea de boletos, asegurando que cada asiento sea reservado de manera adecuada.
4. **Promoción de ventas:** Facilita la aplicación de descuentos y promociones, incentivando la compra de boletos y productos en la zona de comida.
5. **Flexibilidad en la gestión del cine:** Los empleados pueden agregar nuevas funciones, gestionar promociones y modificar la programación de manera dinámica según la demanda.
6. **Seguridad y reducción de errores:** Minimiza la posibilidad de errores humanos en la asignación de asientos y reservas, asegurando una operación confiable.

En conclusión, la creación de un sistema de reservas para un cine representa un avance significativo en la modernización de la industria del entretenimiento, brindando beneficios tanto para los clientes como para la administración del cine. Con esta solución, se logra una operación más eficiente, una mejor experiencia para los usuarios y un aumento en la rentabilidad del negocio.

Ahora mencionaré las clases, atributos y métodos llevados a cabo en este programa:

## **Clases y sus Descripciones**

### **1. Clase Persona**

Clase base para representar a las personas en el sistema, ya sean empleados o usuarios.

- **Atributos:**
  - nombre: Nombre de la persona.
  - contacto: Información de contacto.

- lista\_personas: Lista donde se almacenan las personas registradas.
- **Métodos:**
  - registro(per1): Agrega una persona a la lista de personas registradas.
  - act\_nombre(nombre\_nuevo): Actualiza el nombre de la persona.
  - act\_contacto(contacto\_nuevo): Actualiza el contacto de la persona.

## **2. Clase Empleado**

Hereda de Persona y representa a los empleados del cine, quienes pueden gestionar funciones, películas y promociones.

- **Atributos:**
  - rol: Tipo de empleado (mesero, barista, gerente, etc.).
- **Métodos:**
  - agregar\_func(hora, sala, pelicula): Crea y devuelve una nueva función de cine.
  - agregar\_pelicula(titulo, genero, duracion, clasificacion): Crea y devuelve una nueva película.
  - agregar\_promocion(descuento, condicion): Crea y devuelve una nueva promoción.

## **3. Clase Usuario**

Hereda de Persona y representa a los clientes que pueden realizar reservas y acceder a promociones.

- **Atributos:**
  - reservas\_hechas: Lista de reservas realizadas por el usuario.
- **Métodos:**
  - hacer\_reserva(funcion, asientos): Realiza una reserva si hay asientos disponibles.
  - cancelar\_reserva(reserva): Cancela una reserva y libera los asientos.
  - acceder\_promo(promocion): Valida y aplica promociones si el usuario cumple con la condición.

## **4. Clase Espacio**

Clase base para representar espacios en el cine, como salas y zonas de comida.

- **Atributos:**
  - nombre: Nombre del espacio.
  - capacidad: Capacidad del espacio en personas.
- **Métodos:**
  - desc(): Devuelve una descripción del espacio y su capacidad.

## 5. Clase Sala

Hereda de Espacio y representa las salas de cine.

- **Atributos:**
  - tipo: Tipo de sala (2D, 3D, IMAX).
  - asientos: Matriz que representa la disponibilidad de asientos.
  - funciones: Lista de funciones programadas en la sala.
- **Métodos:**
  - tipo\_sala(): Devuelve el tipo de sala y la disponibilidad de asientos.
  - consultar\_disponibilidad(): Retorna la cantidad de asientos disponibles.

## 6. Clase ZonaComida

Hereda de Espacio y representa la zona de comida del cine.

- **Atributos:**
  - productos: Diccionario con productos y su disponibilidad.
- **Métodos:**
  - menu(): Devuelve el menú de productos disponibles.
  - vender\_producto(producto, cantidad): Verifica disponibilidad y realiza la venta de productos.

## 7. Clase Reserva

Representa una reserva hecha por un usuario para una función específica.

- **Atributos:**
  - usuario: Nombre del usuario que realizó la reserva.
  - funcion: Función reservada.
  - asientos: Lista de asientos reservados.
- **Métodos:**
  - confirmar\_reserva(): Devuelve los detalles de la reserva confirmada.
  - cancelar\_reserva(): Cancela la reserva y libera los asientos reservados.

## 8. Clase Funcion

Representa una función de cine programada en una sala.

- **Atributos:**

- hora: Hora de la función.
- sala: Sala donde se proyecta la película.
- pelicula: Película proyectada en la función.
- asientos\_ocupados: Conjunto de asientos ocupados.
- **Métodos:**
  - verificar\_disponibilidad(asientos): Verifica si los asientos están disponibles.
  - ocupar\_asientos(asientos): Marca los asientos como ocupados.
  - liberar\_asientos(asientos): Libera los asientos reservados.
  - consultar\_asientos(): Muestra la cantidad de asientos disponibles.

## **9. Clase Pelicula**

Representa una película en cartelera.

- **Atributos:**
  - titulo: Nombre de la película.
  - genero: Género de la película.
  - duracion: Duración en minutos.
  - clasificacion: Clasificación de edad.
- **Métodos:**
  - detalles(): Devuelve los detalles de la película.

## **10. Clase Promocion**

Permite aplicar descuentos bajo ciertas condiciones.

- **Atributos:**
  - descuento: Porcentaje de descuento aplicado.
  - condicion: Condición para aplicar la promoción.
- **Métodos:**
  - validar\_condicion(): Verifica si se cumple la condición para aplicar la promoción.
  - aplicar\_promo(): Aplica el descuento si la condición es válida.

Captura del código ejecutado:

```
[2] ✓ 0.0s

**
=== CARTELERA ===
Película: Avatar
Hora: 15:00
Sala: Sala 1 (2D)
Asientos disponibles: La sala Sala 1 tiene 100 asientos disponibles para la película: Avatar
-----
Película: Toy Story 4
Hora: 18:00
Sala: Sala 1 (2D)
Asientos disponibles: La sala Sala 1 tiene 100 asientos disponibles para la película: Toy Story 4
-----
Película: Avatar
Hora: 20:00
Sala: Sala 2 (3D)
Asientos disponibles: La sala Sala 2 tiene 80 asientos disponibles para la película: Avatar
-----
Película: Avengers
Hora: 17:00
Sala: Sala IMAX (IMAX)
Asientos disponibles: La sala Sala IMAX tiene 60 asientos disponibles para la película: Avengers
-----

=== DETALLES DE PELÍCULA ===
La película Avatar de género Ciencia Ficción, clasificación PG-13 tiene una duración de 180 minutos

=== DISPONIBILIDAD DE SALA ===
Sala IMAX tiene capacidad para 60 personas es una sala: IMAX y tiene 60 asientos disponibles

=== PROCESO DE RESERVA ===
Intentando reservar asientos [(1, 1), (1, 2)] para Avatar
Reserva exitosa!
Usuario: María, Función: Avatar, asientos: [(1, 1), (1, 2)]

=== PROMOCIONES DISPONIBLES ===
Descuento: 50% estudiantes
Descuento: 30% jubilados

=== RESERVAS DEL USUARIO ===
Película: Avatar
Hora: 15:00
Asientos: [(1, 1), (1, 2)]

=== ASIENTOS DISPONIBLES ACTUALIZADOS ===
La sala Sala 1 tiene 98 asientos disponibles para la película: Avatar
```

## Justificación e Importancia del Sistema de Gestión de Pedidos en una Cafetería

### Justificación

El desarrollo de un sistema de gestión de pedidos para una cafetería responde a la necesidad de optimizar la experiencia del cliente y la eficiencia operativa del negocio. En un entorno donde la rapidez y la personalización de los pedidos son clave, contar con una herramienta que permita administrar pedidos, inventario y promociones de manera automatizada es esencial.

Este sistema está diseñado para mejorar la gestión de los productos y garantizar que los clientes puedan personalizar sus pedidos según sus preferencias. Además, permite a los empleados actualizar el inventario en tiempo real y prevenir la aceptación de pedidos cuando faltan ingredientes, lo que reduce errores y mejora la atención al cliente. Asimismo, la incorporación de promociones dirigidas fomenta la lealtad de los clientes frecuentes, incentivando el consumo y mejorando la rentabilidad del negocio.

## Importancia

La implementación de este sistema de gestión es crucial por diversas razones:

1. **Mejora de la experiencia del cliente:** Facilita la personalización de pedidos, permitiendo a los usuarios seleccionar opciones específicas como tamaño, temperatura y adiciones.
2. **Optimización del tiempo y recursos:** Automatiza la toma de pedidos y el control de inventario, reduciendo el tiempo de espera y evitando problemas derivados de la falta de ingredientes.
3. **Eficiencia operativa:** Permite a los empleados actualizar el inventario y gestionar pedidos de manera ágil, evitando confusiones o errores en la preparación.
4. **Fidelización de clientes:** La integración de un sistema de promociones incentiva la repetición de compras y premia a los clientes frecuentes.
5. **Reducción de desperdicios:** Al mantener un control preciso del inventario, se minimizan las pérdidas por sobrecompra o caducidad de ingredientes.
6. **Transparencia y control:** Permite registrar y consultar el historial de pedidos, brindando información valiosa para análisis y toma de decisiones.

En conclusión, el desarrollo de este sistema representa una solución integral para la administración eficiente de una cafetería, proporcionando una experiencia de usuario mejorada y optimizando los procesos internos del negocio. Con su implementación, se logra una operación más ordenada, rentable y adaptable a las necesidades del mercado actual.

Ahora haré mención de las clases, atributos y métodos utilizados:

## Clases y sus Descripciones

### 1. Persona (Clase Base)

*Atributos:*

- nombre: Nombre de la persona.
- contacto: Datos de contacto.

*Métodos:*

- actualizar\_datos(nombre, contacto): Permite actualizar el nombre y/o contacto de la persona.

## **2. Cliente (Hereda de Persona)**

*Atributos:*

- historial\_pedidos: Lista de pedidos realizados.
- puntos\_fidelidad: Puntos acumulados por pedidos.

*Métodos:*

- realizar\_pedido(productos, personalizaciones): Verifica stock y crea un nuevo pedido.
- consultar\_historial(): Retorna el historial de pedidos del cliente.

## **3. Empleado (Hereda de Persona)**

*Atributos:*

- rol: Define si el empleado es mesero, barista o gerente.

*Métodos:*

- actualizar\_inventario(producto, cantidad): Permite modificar el stock de un producto.
- preparar\_pedido(pedido): Cambia el estado de un pedido a "en preparación".
- entregar\_pedido(pedido): Cambia el estado de un pedido a "entregado".

## **4. ProductoBase (Clase Base para Productos)**

*Atributos:*

- nombre: Nombre del producto.
- precio: Precio del producto.
- stock\_minimo: Cantidad mínima en inventario.



## 5. Bebida (Hereda de ProductoBase)

*Atributos:*

- tamaño: Define el tamaño de la bebida (pequeño, mediano, grande).
- tipo: Tipo de bebida (caliente o fría).
- opciones\_personalizables: Diccionario con opciones de personalización como tipo de leche, azúcar y extras.

## 6. Postre (Hereda de ProductoBase)

*Atributos:*

- es\_vegano: Indica si el postre es apto para veganos.
- sin\_gluten: Indica si el postre es libre de gluten.

## 7. Inventario

Clase para la gestión del stock de productos.

*Atributos:*

- \_stock: Diccionario que almacena la cantidad disponible de productos.

*Métodos:*

- actualizar\_stock(producto, cantidad): Modifica la cantidad de un producto en el inventario.
- verificar\_stock(productos): Comprueba si hay stock suficiente para los productos solicitados.
- descontar\_stock(productos): Reduce la cantidad disponible de productos tras realizar un pedido.

## 8. Pedido

*Atributos:*

- cliente: Cliente que realiza el pedido.
- productos: Lista de productos en el pedido.
- personalizaciones: Diccionario con personalizaciones aplicadas.

- estado: Estado actual del pedido (pendiente, en preparación, entregado).
- total: Costo total del pedido.

*Métodos:*

- calcular\_total(): Calcula el precio total del pedido aplicando descuentos si existen.
- agregar\_producto(producto, personalizacion): Añade un producto al pedido si hay stock disponible.

## **9. Promocion**

Clase para gestionar descuentos en los pedidos.

*Atributos:*

- nombre: Nombre de la promoción.
- tipo: Define si la promoción se aplica a productos individuales o al pedido completo.
- descuento: Porcentaje de descuento.
- condicion: Función que evalúa si la promoción se aplica.

*Métodos:*

- aplicar\_descuento(total): Aplica el descuento al total de un pedido.
- obtener\_promociones(pedido): Devuelve las promociones aplicables a un pedido.

Captura del código ejecutado:

[4] ✓ 0.0s

```
...
=== Creando Productos ===
Productos creados:
- Café Americano: $25
- Café Latte: $35
- Frappé: $45
- Croissant: $30
- Muffin de Plátano: $28

=== Actualizando Inventario ===
Stock actual:
- Café Americano: 20 unidades
- Café Latte: 15 unidades
- Frappé: 10 unidades
- Croissant: 12 unidades
- Muffin de Plátano: 8 unidades

=== Creando Personal ===
Personal de la cafetería:
- Juan: barista
- María: mesero
- Carlos: gerente

=== Registrando Clientes ===

=== Configurando Promociones ===
Promoción creada: Cliente Frecuente - 15.0% descuento

=== Procesando Pedidos ===

Pedido de Ana:
Productos:
- Café Latte
- Croissant
Personalización del café:
  tipo_leche: almendra
  azucar: sin azúcar
  extras: ['canela']
Total: $65

=== Preparación y Entrega ===
Estado inicial: pendiente
Después de preparación: en_preparacion
Después de entrega: entregado

=== Programa de Fidelidad ===
Puntos de fidelidad de Ana: 2

=== Historial de Pedidos ===
Pedidos de Ana:
- Pedido con 2 productos - Estado: entregado
```

## Justificación e Importancia del Sistema de Biblioteca Digital

### Justificación

El desarrollo de un sistema de biblioteca digital es esencial para mejorar la gestión y acceso a materiales bibliográficos en distintas sucursales. La digitalización de catálogos, la automatización de préstamos y la aplicación de penalizaciones por retrasos son aspectos clave que optimizan la administración de recursos y mejoran la experiencia de los usuarios.

Este sistema permite que los usuarios puedan consultar, reservar y devolver materiales de manera eficiente, garantizando la disponibilidad equitativa de los recursos. Además, los bibliotecarios pueden gestionar el inventario, registrar préstamos y aplicar sanciones automáticas por devoluciones tardías. La posibilidad de transferir materiales entre sucursales facilita el acceso a recursos específicos sin importar la ubicación física del usuario.

### Importancia

Implementar este sistema de biblioteca digital conlleva una serie de beneficios significativos:

1. **Accesibilidad y disponibilidad:** Permite a los usuarios consultar materiales desde cualquier ubicación y realizar préstamos de manera organizada.
2. **Optimización de la gestión bibliotecaria:** Facilita la administración de catálogos, materiales y préstamos en múltiples sucursales.
3. **Control automatizado de préstamos y devoluciones:** Reduce la carga administrativa al gestionar fechas límite y aplicar penalizaciones automáticamente.
4. **Prevención de pérdida de materiales:** Registra los préstamos y devoluciones, evitando la pérdida de recursos bibliográficos.
5. **Sistema de penalizaciones eficiente:** Implementa sanciones automáticas para incentivar la devolución puntual de materiales.
6. **Facilidad para localizar materiales:** A través del catálogo digital, los usuarios pueden buscar materiales por autor, género o tipo de contenido.
7. **Intercambio de materiales entre sucursales:** Garantiza que los usuarios puedan acceder a materiales específicos sin importar su ubicación.

En conclusión, este sistema de biblioteca digital no solo moderniza la administración de recursos bibliográficos, sino que también mejora la eficiencia operativa de las bibliotecas y la experiencia del usuario, promoviendo un acceso equitativo y organizado al conocimiento.

Ahora mencionaré las clases, atributos y métodos empleados:

## 1. Material (Clase base)

Representa cualquier material disponible en la biblioteca.

- titulo: Nombre del material.
- estado: Indica si el material está **disponible** o **prestado**.
- sucursal\_actual: Sucursal donde está ubicado.

*Métodos:*

- cambiar\_estado(nuevo\_estado): Modifica el estado del material.
- asignar\_sucursal(sucursal): Asigna el material a una sucursal específica.

## 2. Libro (Hereda de Material)

- autor: Nombre del autor.
- genero: Género del libro.

## 3. Revista (Hereda de Material)

- edicion: Número de edición.
- periodicidad: Frecuencia de publicación (mensual, semanal, etc.).

## 4. MaterialDigital (Hereda de Material)

- tipo\_archivo: Tipo de archivo digital (PDF, EPUB, etc.).
- enlace\_descarga: URL para descarga.

## 5. Persona (Clase base)

Representa cualquier persona dentro del sistema.

- nombre: Nombre de la persona.
- id: Identificación única.

## 6. Usuario (Hereda de Persona)

Representa a los clientes de la biblioteca.

- materiales\_prestados: Lista de materiales actualmente prestados.
- multas: Lista de multas pendientes.

*Métodos:*

- consultar\_catalogo(catalogo, \*\*criterios): Busca materiales en el catálogo.
- solicitar\_préstamo(material, préstamo): Intenta realizar un préstamo.
- devolver\_material(material, préstamo): Devuelve un material prestado.
- tiene\_multas\_pendientes(): Verifica si el usuario tiene multas sin pagar.

## **7. Bibliotecario (Hereda de Persona)**

Representa al personal de la biblioteca.

- sucursal: Sucursal donde trabaja.

*Métodos:*

- agregar\_material(material): Añade un material a la biblioteca.
- gestionar\_préstamo(usuario, material): Procesa un préstamo.
- transferir\_material(material, sucursal\_destino): Mueve un material entre sucursales.

## **8. Sucursal**

Representa una sucursal de la biblioteca.

- nombre: Nombre de la sucursal.
- ubicación: Dirección.
- catalogo: Catálogo de materiales disponibles.
- materiales: Lista de materiales en la sucursal.

*Métodos:*

- agregar\_material(material): Añade un material a la sucursal.
- remover\_material(material): Elimina un material del inventario.

## **9. Préstamo**

Gestiona el préstamo de materiales.

- fecha\_prestamo: Fecha en que se realizó el préstamo.
- fecha\_devolución: Fecha límite para devolver el material.

- usuario: Persona que realizó el préstamo.
- material: Material prestado.

*Métodos:*

- crear\_préstamo(usuario, material): Registra un nuevo préstamo.
- finalizar\_préstamo(usuario, material): Finaliza el préstamo y verifica retrasos.
- calcular\_dias\_retrazo(): Calcula cuántos días de retraso tiene la devolución.

## **10. Penalización**

Gestiona multas por retraso en devoluciones.

- MULTA\_POR\_DIA: Define el costo por día de retraso (\$1 por día).

*Métodos:*

- crear\_multa(usuario, dias\_retrazo): Calcula y asigna una multa.

## **11. Catálogo**

Almacena y gestiona la búsqueda de materiales en una sucursal.

- materiales: Lista de materiales disponibles.

*Métodos:*

- agregar\_material(material): Añade un material al catálogo.
- remover\_material(material): Elimina un material del catálogo.
- buscar(\*\*criterios): Permite filtrar materiales por autor, género o tipo.

## **12. SistemaBibliotecario**

Administra todas las sucursales.

- sucursales: Lista de sucursales registradas.

*Métodos:*

- agregar\_sucursal(sucursal): Añade una nueva sucursal al sistema.
- buscar\_en\_todas\_sucursales(\*\*criterios): Busca materiales en todas las sucursales.

Captura del código ejecutado:

```
[3] ✓ 0.0s

...   === Prueba de búsqueda en todas las sucursales ===
      Libros encontrados en todas las sucursales:
      - Cien años de soledad en Biblioteca Central
      - El código Da Vinci en Biblioteca Central
      - Don Quijote en Biblioteca Norte

      === Prueba de préstamo y penalización ===
      Préstamo exitoso del libro: Cien años de soledad
      Libro devuelto con retraso
      Multas del usuario: 1
      Monto de la multa: $5.0

      === Prueba de transferencia entre sucursales ===
      Ubicación inicial de El código Da Vinci: Biblioteca Central
      Libro transferido exitosamente a: Biblioteca Norte
```