




12 DE FEBRERO DE 2025

SISTEMAS EN PYTHON USANDO POO

PROGRAMACION AVANZADA

CRISTIAN EMILIO MUÑOZ VAZQUEZ
BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA
CAMPUS- CIUDAD UNIVERSITARIA 2



PRACTICA UNIDAD 1- SISTEMAS USANDO POO EN PYTHON

Sistema 1: Cine

Motivo de desarrollo del sistema:

Reservas de cine: Permite simular un ambiente de un sistema para administrar un cine, permitiendo que usuarios comunes y empleados hagan tareas como reservar asientos para una función o acceder a promociones, los empleados tienen mas herramientas como agregar funciones o películas nuevas.

Gestión de Espacios y Servicios: El sistema también gestiona diferentes espacios como las salas de cine y zonas de comida, con funcionalidades específicas como la disponibilidad de asientos y la reserva de estos asientos.

Facilitar trabajo: Por medio del sistema al ser el usuario el que puede reservar, agiliza el proceso para entrar a una función lo que hace mas dinámica la funcionalidad del cine

Clases a ocupar (Métodos y atributos usados)

Clase Persona

Atributos:

nombre: Nombre de la persona.

correo: Correo electrónico de la persona.

Métodos:

__init__(nombre, correo): Constructor de la clase.

Clase Usuario (hereda de Persona)

Atributos:

reservas: Lista de reservas realizadas por el usuario.

Métodos:

hacer_reserva(reserva): Añade una reserva a la lista de reservas.

Clase Empleado (hereda de Persona)

Atributos:

rol: Rol del empleado (Ej. Administrador).

Métodos:

agregar_funcion(funciones, pelicula, sala, horario): Añade una nueva función de cine.

agregar_pelicula(peliculas, titulo, duracion, clasificacion, genero): Añade una nueva película.

agregar_promocion(promociones, descripcion, descuento): Añade una nueva promoción.

Clase Espacio

Atributos:

nombre: Nombre del espacio.

Métodos:

__init__(nombre): Constructor de la clase.

Clase Sala (hereda de Espacio)

Atributos:

tipo: Tipo de sala (2D, 3D).

capacidad: Número de asientos disponibles.

disponibilidad: Matriz que indica la disponibilidad de los asientos.

Métodos:

verificar_disponibilidad(asientos): Verifica si los asientos solicitados están disponibles.

reservar_asientos(asientos): Reserva los asientos si están disponibles.

Clase ZonaComida (hereda de Espacio)

Atributos:

productos: Diccionario de productos y sus precios.

Métodos:

agregar_producto(producto, precio): Añade un producto al menú de la zona de comida.

Clase Pelicula

Atributos:

titulo: Título de la película.

duracion: Duración de la película.

clasificacion: Clasificación de la película (Ej. PG-13).

genero: Género de la película.

Métodos:

__init__(titulo, duracion, clasificacion, genero): Constructor de la clase.

Clase Promocion

Atributos:

descripcion: Descripción de la promoción.

descuento: Porcentaje de descuento que ofrece la promoción.

Métodos:

__init__(descripcion, descuento): Constructor de la clase.

Clase Funcion

Atributos:

pelicula: Película que se está proyectando en la función.

sala: Sala en la que se proyecta la película.

horario: Horario de la función.

Métodos:

__init__(pelicula, sala, horario): Constructor de la clase.

Clase Reserva

Atributos:

usuario: Usuario que realiza la reserva.

funcion: Función de cine reservada.

asientos: Lista de asientos reservados.

promocion: Promoción aplicada, si existe.

total: Total a pagar por la reserva.

Métodos:

calcular_precio(): Calcula el precio de la reserva.

confirmar_reserva(): Confirma si la reserva puede realizarse.

detalle_reserva(): Muestra los detalles de la reserva.

Salida de datos

```
Reserva para Chris en la sala Sala 3D
Función: Sonic 3 a las 18:00
Asientos reservados: [(0, 1), (0, 2), (1, 3)]
Subtotal: $300
Descuento: $30.0
Total a pagar: $270.0
```

Sistema 2: Cafeteria

Motivo de desarrollo del sistema

Clientes: Permitir que los clientes realicen pedidos cómodamente y, al mismo tiempo, acumulen descuentos por su fidelidad, creando una experiencia más atractiva para ellos.

Empleados: Ayudar a los empleados a gestionar los pedidos de manera más eficiente, desde su recepción hasta su entrega, y facilitar el control de productos e inventarios.

Productos: Ofrecer una variedad de productos (como bebidas o postres) y asegurarse de que estén disponibles en todo momento.

Control de Inventario: Saber en tiempo real qué ingredientes están disponibles y cuáles están por agotarse. Esto garantiza que nunca falten productos para la preparación de los pedidos.

Descuentos Personalizados: Ofrecer descuentos de manera automática a los clientes que han demostrado ser fieles, basados en la cantidad de pedidos que han realizado. Esto no solo recompensa a los clientes frecuentes, sino que también motiva a que sigan comprando en el negocio.

Clases a ocupar:

Clase Persona

Atributos:

nombre: Nombre de la persona.

Clase Cliente (Hereda de Persona)

Atributos:

historial_pedidos: Lista de pedidos realizados por el cliente.

Métodos:

realizar_pedido(pedido): Agrega un pedido al historial.

obtener_descuento_fidelidad(): Calcula el descuento basado en la cantidad de pedidos.

Clase Empleado (Hereda de Persona)

Atributos:

rol: Rol del empleado (ej. barista, cajero).

Clase ProductoBase

Atributos:

nombre: Nombre del producto.

precio: Precio del producto.

Clase Bebida (Hereda de ProductoBase)

Atributos:

tamano: Tamaño de la bebida.

tipo: Tipo de bebida (ej. caliente o fría).

opciones_personalizadas: Ingredientes personalizados de la bebida.

Clase Postre (Hereda de ProductoBase)

Atributos:

es_vegano: Indica si el postre es vegano.

sin_gluten: Indica si el postre es libre de gluten.

Clase Inventario

Atributos:

ingredientes: Diccionario con los ingredientes y su cantidad disponible.

Métodos:

agregar_ingrediente(nombre, cantidad): Agrega ingredientes al inventario.

verificar_disponibilidad(producto): Verifica si hay suficientes ingredientes.

descontar_ingredientes(producto): Resta los ingredientes utilizados en una bebida.

Clase Pedido

Atributos:

productos: Lista de productos en el pedido.

estado: Estado del pedido (ej. "pendiente").

Métodos:

agregar_producto(producto): Agrega un producto al pedido.

calcular_total(): Calcula el costo total del pedido.

Clase Promocion

Atributos:

descuento: Porcentaje de descuento aplicado.

criterio: Función que determina si el descuento aplica.

Métodos:

aplicar_descuento(pedido): Aplica el descuento si se cumplen las condiciones.

Función procesar_pedido(cliente, empleado, *productos)

Descripción:

Crea un pedido y verifica la disponibilidad de los productos.

Aplica descuentos de fidelidad si corresponden.

Registra el pedido en el historial del cliente.

Muestra el resumen de la compra.

Salida de datos

Descripción del pedido:

Cliente: Chris

Productos: ['Café Americano']

Descuento por fidelidad: 0.0%

Empleado que atendió: Juan

Total: \$3.00

Descripción del pedido:

Cliente: Edi

Productos: ['Té Chai', 'Brownie']

Descuento por fidelidad: 0.0%

Empleado que atendió: Axel

Total: \$6.50

No hay suficientes ingredientes para preparar Latte.

No se pudo completar el pedido de Jochi.

Sistema 3: Biblioteca

Motivo De desarrollo del sistema

Usuarios: El sistema permite que los usuarios (clientes) puedan solicitar préstamos de libros, revistas y materiales digitales de manera cómoda y eficiente. Además, premia la fidelidad de los usuarios frecuentes con beneficios como extensión de plazos de préstamo o reducción de multas, incentivando el uso constante de la biblioteca.

Bibliotecarios: Facilita la gestión de préstamos, devoluciones y transferencias de materiales entre sucursales. También les permite llevar un control preciso del catálogo, asegurando la disponibilidad de libros y otros recursos para los usuarios.

Materiales: El sistema organiza y administra una variedad de materiales, incluyendo libros, revistas y recursos digitales, garantizando su acceso en todo momento.

Control de Inventario: Permite conocer en tiempo real qué materiales están disponibles, cuáles están prestados y cuáles requieren reposición. Esto ayuda a optimizar la gestión del catálogo y evita la falta de recursos.

Clases a ocupar

Clases utilizadas y su función

Material (Clase Base)

Representa cualquier tipo de material disponible en la biblioteca (libros, revistas, materiales digitales).

Atributos: titulo, estado (disponible/prestado).

Libro (Subclase de Material)

Específico para libros, con atributos adicionales como autor y género.

Revista (Subclase de Material)

Representa revistas, con atributos como edición y periodicidad.

MaterialDigital (Subclase de Material)

Material en formato digital con tipo_archivo y enlace_descarga.

Persona (Clase Base)

Clase general para representar personas en el sistema.

Usuario (Subclase de Persona)

Representa a los usuarios que pueden pedir materiales en préstamo.

Atributos: materiales_prestados (lista de materiales que tiene en préstamo) y multa (monto acumulado por retrasos).

Métodos: aplicar_multa().

Bibliotecario (Subclase de Persona)

Puede agregar material a una sucursal, gestionar préstamos y transferir materiales entre sucursales.

Sucursal

Representa una biblioteca física con un catálogo de materiales.

Métodos: buscar_material(), listar_materiales().

Préstamo

Gestiona los préstamos de los materiales a los usuarios.

Métodos: devolver_material() (gestiona devoluciones y aplica multas si hay retraso).

Salida de datos

```
Materiales disponibles: ['Muy Interesante', 'El libro de la selva']
Préstamos activos:
Christian Muñoz ha prestado 'El principe' por 14 días
Multa de Christian Muñoz: 0
```

LINK DE GITHUB: https://github.com/MvzChriz/Unidad_1