



BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA

ING. EN CIENCIAS DE LA COMPUTACION

LUIS ALBERTO SANCHEZ GONZALEZ

PROGRAMACION AVANZADA

PRACTICA UNIDAD 1

Sistema de Reserva para un Cine

En la actualidad, los cines buscan optimizar sus procesos de gestión y brindar una mejor experiencia a sus clientes. Un sistema de reservas eficiente no solo mejora la organización interna del cine, sino que también facilita la interacción con los usuarios, permitiendo una mayor satisfacción y fidelidad.

El proceso manual de reserva de boletos puede resultar en errores humanos, tiempos de espera prolongados y una experiencia frustrante para los clientes. Además, la falta de un sistema automatizado puede llevar a la sobreventa de boletos o la incapacidad de gestionar adecuadamente las reservas y cancelaciones.

El sistema de reserva para un cine automatiza el proceso de reserva de boletos, permitiendo a los usuarios seleccionar asientos y horarios de manera sencilla y eficiente. Este sistema reduce la carga de trabajo del personal del cine y minimiza los errores humanos, ofreciendo una experiencia de usuario fluida y satisfactoria.

CLASES

Clase Persona

La clase Persona representa a cualquier persona asociada con el sistema, como usuarios o empleados. Tiene una lista estática que almacena todas las instancias de Persona registradas. Los atributos incluyen nombre y correo, que se inicializan al crear una instancia de la clase. Los métodos incluyen Registrar para agregar la instancia a la lista de personas registradas, ActualizarDatos para actualizar la información de la persona y personas_registradas para imprimir todas las personas registradas.

Clase Usuario

La clase Usuario hereda de Persona y representa a un usuario que puede reservar asientos en funciones. Además de los atributos heredados, tiene una lista numreserva que almacena las reservas del usuario. Los métodos incluyen reserva para reservar un asiento, cancelar_reserva para cancelar una reserva y acceder_promo para acceder a una promoción.

Clase Empleado

La clase Empleado también hereda de Persona y representa a un empleado del cine. Tiene un atributo adicional rol que define el rol del empleado (por ejemplo, taquillero, administrador). Los métodos incluyen realizar_tarea para realizar tareas según el rol del empleado y agregar_funcion para agregar funciones a la lista de funciones si el empleado tiene permisos.

Clase Espacio

La clase Espacio representa un espacio físico en el cine, como una sala o una zona de comida. Los atributos incluyen tamaño e identificador. El método descripcion imprime la descripción del espacio.

Clase Sala

La clase Sala hereda de Espacio y representa una sala de cine. Tiene un atributo adicional tipo, que puede ser 2D, 3D o IMAX, y un atributo disponibilidad que indica si la sala está disponible. El método consultar_disponibilidad verifica la disponibilidad de la sala.

Clase Zona_comida

La clase Zona_comida hereda de Espacio y representa una zona de comida en el cine. Tiene un atributo menu, que es un diccionario que contiene los productos y sus precios. Los métodos incluyen agregar_productos para agregar productos al menú y mostrar_menu para mostrar el menú de comida.

Clase Pelicula

La clase Pelicula representa una película que se proyecta en el cine. Los atributos incluyen titulo, duracion, clasificacion y genero. El método Detalles imprime los detalles de la película.

Clase Funcion

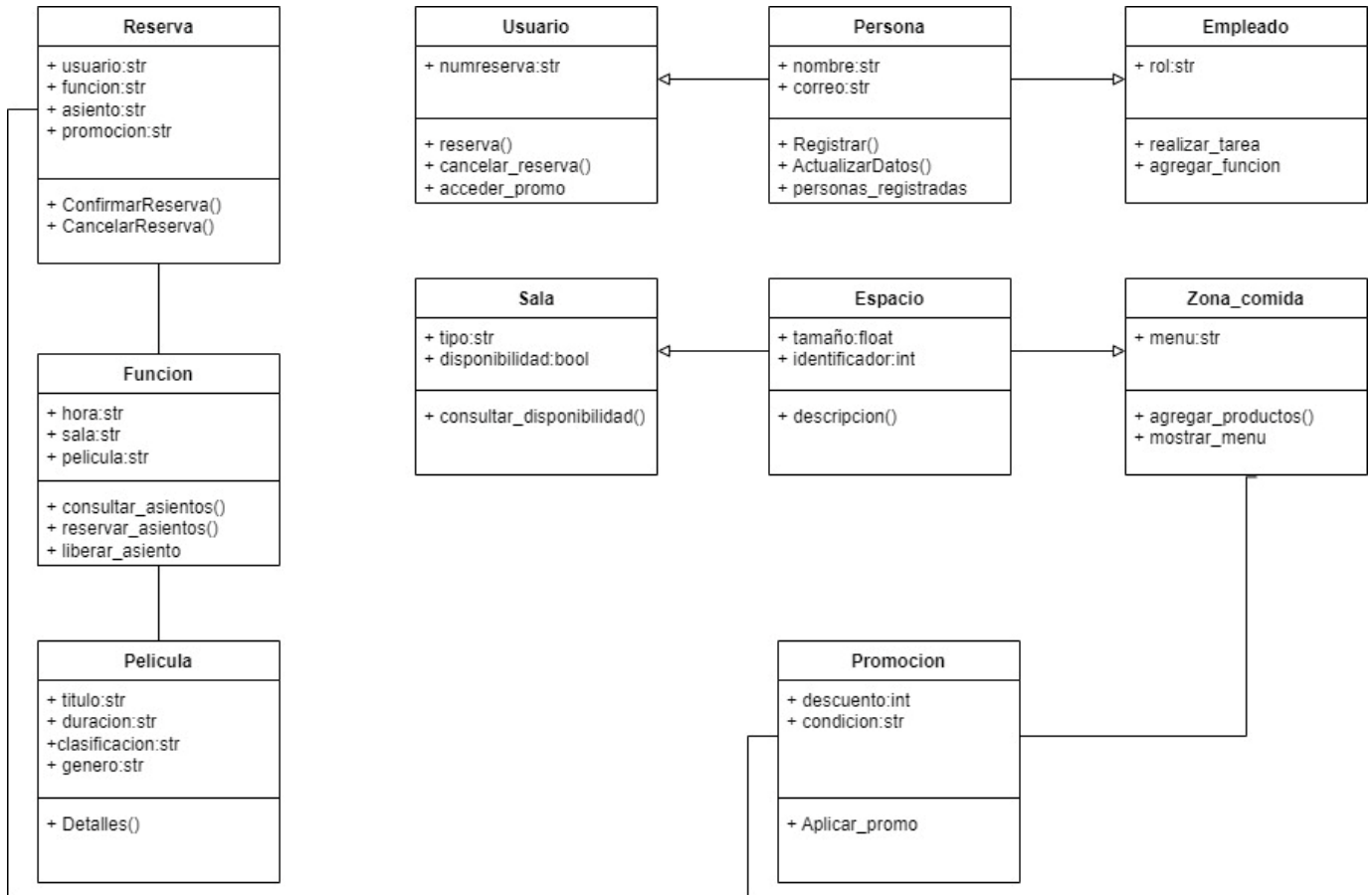
La clase Funcion representa una función de cine. Los atributos incluyen hora, sala, pelicula y una lista asientos_disponibles. Los métodos incluyen consultar_asientos para consultar los asientos disponibles, reservar_asientos para reservar un asiento y liberar_asiento para liberar un asiento reservado.

Clase Promocion

La clase Promocion representa una promoción que ofrece descuentos a los usuarios. Los atributos incluyen descuento y condicion, que define la condición para aplicar la promoción. El método Aplicar_promo aplica la promoción al usuario si cumple la condición.

Clase Reserva

La clase Reserva representa una reserva de un asiento en una función. Los atributos incluyen usuario, funcion, asiento y una promocion opcional. Los métodos incluyen ConfirmarReserva para confirmar la reserva de un asiento y CancelarReserva para cancelar la reserva de un asiento.



ENLACE

<https://colab.research.google.com/github/Renm1090/Reservation-System-for-a-Cinema/blob/main/cine.ipynb>

```

p2.Registrar()
p3.Registrar()
em1.Registrar()
em2.Registrar()
em3.Registrar()

Persona.personas_registradas()

```

```

Persona registrada: Luis - luis@gamil.com
Persona registrada: Juan - juan@gmail.com
Persona registrada: Pedro - pedro@gmail.com
Persona registrada: Juju - juju@gmail.com
Persona registrada: Jaimy - Jaimy@gmail.com
Persona registrada: Isaac - isaac@gmail.com
Personas registradas
Nombre: Luis - Correo: luis@gamil.com
Nombre: Juan - Correo: juan@gmail.com
Nombre: Pedro - Correo: pedro@gmail.com
Nombre: Juju - Correo: juju@gmail.com
Nombre: Jaimy - Correo: Jaimy@gmail.com
Nombre: Isaac - Correo: isaac@gmail.com

```

```

men = Zona_comida()
men.agregar_productos("Palomitas", 70)
men.agregar_productos("Nachos", 45)
men.agregar_productos("Coca-Cola", 30)
men.agregar_productos("Agua", 20)

men.mostrar_menu()

```

```

Menú de Comida:
Palomitas: $70
Nachos: $45
Coca-Cola: $30
Agua: $20

```

```

peli1.Detalles()
peli2.Detalles()
peli3.Detalles()

```

```

Título: Your Name
Duración: 1:50
Clasificación: A
Género: Romance
-----
Título: Mi vecino Totoro
Duración: 1:26
Clasificación: AA
Género: Aventura/Fantasia
-----
Título: Aura
Duración: 1:30
Clasificación: A
Género: Comedia/Romance
-----

```

```

fun3 = Funcion("2:30", sal3, peli3)
fun3.consultar_asientos()
fun3.reservar_asientos("A7")
fun3.consultar_asientos()

```

```

Asientos disponibles para: Your Name
Horario: 1:15
Asientos Disponibles:  A1,A2,A3,A4,A5,A6,A7,A8,A9,A10

-----
Asientos disponibles para: Your Name
Horario: 1:15
Asientos Disponibles:  A2,A3,A4,A5,A6,A7,A8,A9,A10

-----
Asientos disponibles para: Mi vecino Totoro
Horario: 2:30
Asientos Disponibles:  A1,A2,A3,A4,A5,A6,A7,A8,A9,A10

-----
Asiento no disponible
Asientos disponibles para: Mi vecino Totoro
Horario: 2:30
Asientos Disponibles:  A1,A2,A3,A4,A6,A7,A8,A9,A10

-----

```

```

res2.ConfirmarReserva()

fun3.consultar_asientos()
res3 = Reserva(user3, fun3, "A5")
res3.ConfirmarReserva()

```

```

Asientos disponibles para: Your Name
Horario: 1:15
Asientos Disponibles:  A2,A3,A4,A5,A6,A7,A8,A9

```

```

Reserva cancelada para Luis en asiento A10.
Asientos disponibles para: Your Name
Horario: 1:15
Asientos Disponibles:  A2,A3,A4,A5,A6,A7,A8,A9,A10

```

```

Reserva confirmada para Juan en asiento A10 para la función Your Name.
Asientos disponibles para: Aura
Horario: 2:30
Asientos Disponibles:  A1,A2,A3,A4,A5,A6,A8,A9,A10

```

```

Reserva confirmada para Pedro en asiento A5 para la función Aura

```

Gestión de Pedidos de una Cafetería

Este programa está diseñado para gestionar una cafetería, incluyendo el manejo de clientes, empleados, productos, inventario y pedidos. La necesidad del programa radica en proporcionar una solución automatizada y eficiente para la administración diaria de la cafetería, lo que permite llevar un control preciso de las operaciones y mejorar la experiencia tanto de los clientes como del personal. Al integrar diferentes funcionalidades en una única aplicación, se facilita la gestión y se optimiza el tiempo y los recursos.

Clase Persona

Esta clase sirve como base para almacenar información común de individuos, como clientes y empleados. Los atributos de esta clase incluyen el nombre y la información de contacto de la persona. Los métodos de la clase permiten inicializar estos atributos al crear una nueva instancia, registrar la persona en una lista, actualizar sus datos y mostrar las personas registradas.

Clase Cliente

La clase Cliente, que hereda de Persona, extiende esta funcionalidad para incluir un historial de pedidos. Además de los atributos heredados, almacena una lista de pedidos realizados por el cliente. Sus métodos permiten inicializar un cliente, añadir pedidos al historial y consultar dicho historial.

Clase Empleado

La clase Empleado, también heredada de Persona, incluye información específica de los empleados, como su rol dentro de la cafetería (por ejemplo, Mesero, Barista, Gerente). Sus métodos permiten inicializar un empleado con su rol y mostrar las tareas correspondientes a dicho rol.

Clase ProductoBase

Esta clase sirve como base para los productos vendidos en la cafetería. Sus atributos incluyen el nombre y el precio del producto, y sus métodos permiten inicializar estos atributos.

Clase Bebida

La clase Bebida, que hereda de ProductoBase, incluye características específicas de las bebidas, como el tamaño, el tipo (caliente o fría) y las opciones de personalización. Los métodos de esta clase permiten inicializar una bebida y mostrar sus detalles en el menú.

Clase Postre

De manera similar, la clase Postre, también heredada de ProductoBase, incluye el tipo de postre (por ejemplo, pastel, galleta). Los métodos permiten inicializar un postre y mostrar sus detalles en el menú.

Clase Inventario

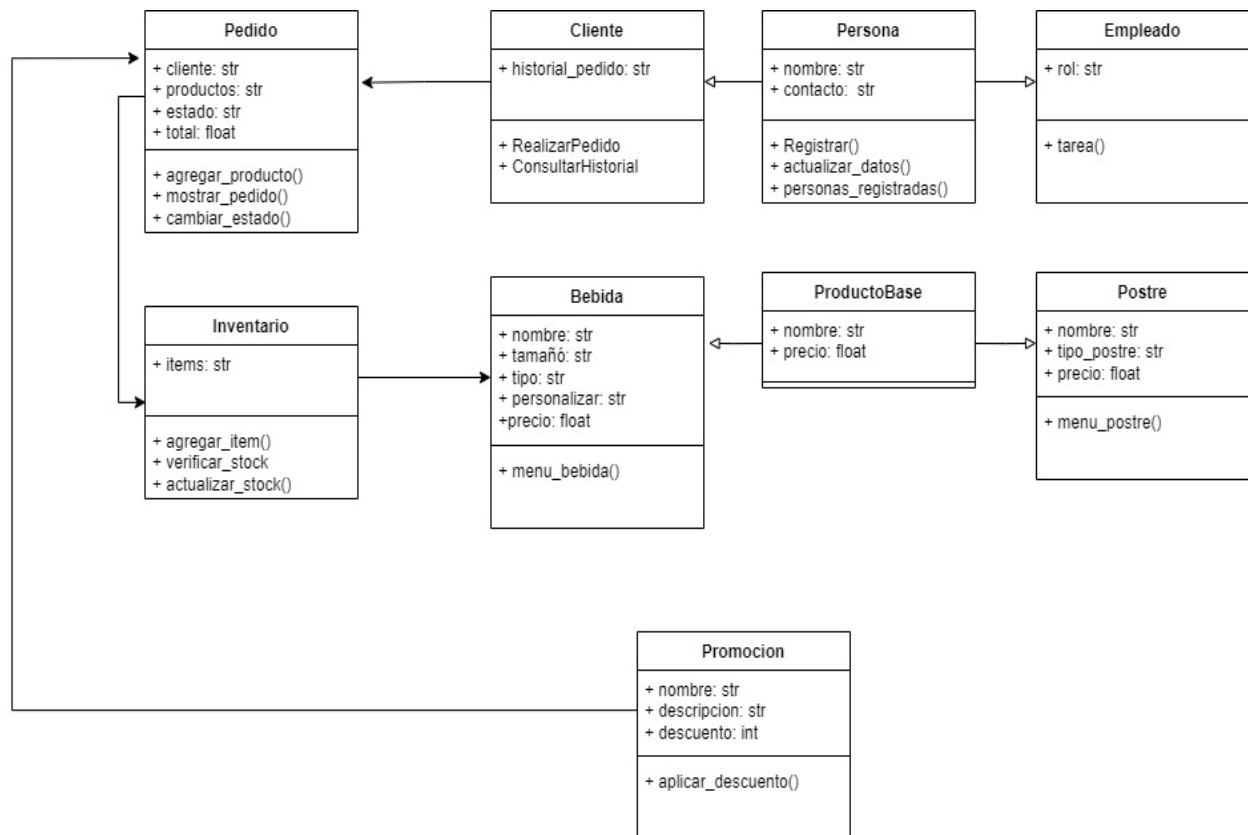
La clase Inventario gestiona los productos y sus cantidades disponibles. Utiliza un diccionario para almacenar la cantidad de cada producto. Sus métodos permiten inicializar el inventario, añadir cantidades de productos, verificar si hay suficiente stock y actualizar el stock después de su uso.

Clase Pedido

La clase Pedido gestiona los pedidos realizados por los clientes. Sus atributos incluyen el cliente que realiza el pedido, una lista de productos incluidos, el estado del pedido (inicialmente "Pendiente") y el total del pedido. Los métodos permiten inicializar un pedido, añadir productos al pedido, mostrar los detalles del pedido y cambiar su estado.

Clase Promocion

La clase Promocion gestiona las promociones y descuentos aplicables a los pedidos. Sus atributos incluyen el nombre, la descripción y el monto del descuento de la promoción. Los métodos permiten inicializar una promoción y aplicar el descuento a un pedido, actualizando su total.



ENLACE

<https://colab.research.google.com/github/Renm1090/Cinema-cafe-and-digital-library/blob/main/Caffe.ipynb>


```
promo.aplicar_descuento(pedido1)
cliente1.RealizarPedido(pedido=pedido1)
cliente1.ConsultarHistorial()
```

✓ 0.0s

Café Latte agregado al pedido.
Galleta Vegana agregado al pedido.
Pedido de Ana - Estado: Pendiente
Café Latte - \$50
Galleta Vegana - \$25
Total: \$75
El pedido ahora está en estado: En preparación
El pedido ahora está en estado: Entregado
Descuento aplicado: \$10. Nuevo total: \$65

✓ Pedido realizado para Ana:
Pedido de Ana - Estado: Entregado
Café Latte - \$50
Galleta Vegana - \$25
Total: \$65

📅 Historial de pedidos de Ana:
📦 Pedido 1:
Pedido de Ana - Estado: Entregado
Café Latte - \$50
Galleta Vegana - \$25
Total: \$65

```
inventario = Inventario()
inventario.agregar_item("cafe", 50)
inventario.agregar_item("Té", 5)
inventario.agregar_item("Leche de almendra", 3)
inventario.agregar_item("Azúcar", 8)
```

✓ 0.0s

Agregado 50 de cafe al inventario.
Agregado 5 de Té al inventario.
Agregado 3 de Leche de almendra al inventario.
Agregado 8 de Azúcar al inventario.

```
empleado1.tarea()
empleado2.tarea()
empleado3.tarea()
```

Luis (Mesero) toma las órdenes de los clientes.
María (Barista) prepara bebidas y hace trucos.
Jaime (Gerente) dirige la cafetería.

Biblioteca Digital

Este código representa un sistema de gestión de una biblioteca, diseñado para organizar y controlar la circulación de materiales como libros, revistas y materiales digitales. En el contexto actual, las bibliotecas no solo administran libros físicos, sino también materiales digitales y revistas. Por lo tanto, la implementación de un sistema como este puede ayudar a mejorar la eficiencia y la accesibilidad de los recursos bibliotecarios, facilitando el proceso de préstamo y devolución, así como el registro y seguimiento de usuarios.

Clase Material

La clase Material es la base del sistema y representa cualquier tipo de recurso disponible en la biblioteca, como libros, revistas y materiales digitales. Sus atributos incluyen el título del material y su disponibilidad. Los métodos de esta clase permiten mostrar información sobre el material y actualizar su estado de disponibilidad.

Clase Libro

La clase Libro hereda de Material y añade atributos específicos para los libros, como el autor y el género. Hereda los métodos de la clase Material, permitiendo así mostrar información y actualizar el estado del libro.

Clase Revista

La clase Revista hereda de Material y añade atributos específicos para las revistas, como la edición y la periodicidad. Al igual que Libro, hereda los métodos de la clase Material.

Clase MaterialDigital

La clase MaterialDigital hereda de Material y representa materiales digitales. Añade atributos como el tipo de archivo y el enlace de acceso al material digital, y hereda los métodos de Material.

Clase Persona

La clase Persona actúa como una base para representar a cualquier individuo asociado con la biblioteca. Sus atributos son el nombre y el correo electrónico de la persona.

Clase Usuario

La clase Usuario hereda de Persona y representa a un usuario de la biblioteca. Sus atributos incluyen una identificación del usuario, una lista de materiales prestados y las

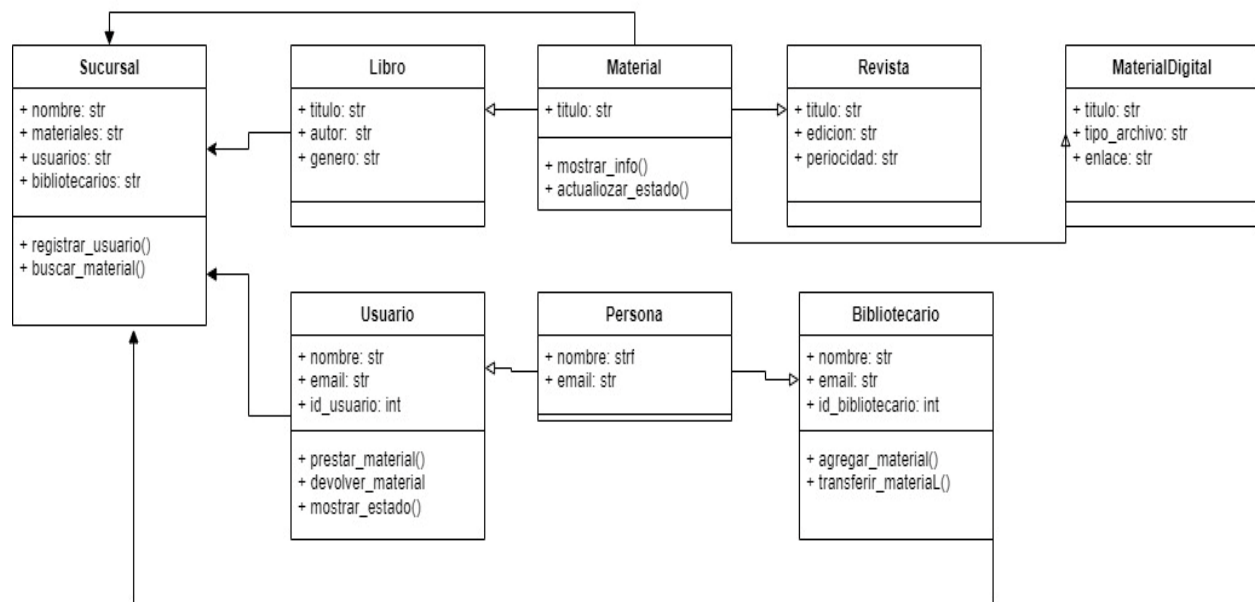
muchas acumuladas. Los métodos de esta clase permiten prestar materiales, devolver materiales y mostrar el estado del usuario.

Clase Bibliotecario

La clase Bibliotecario hereda de Persona y representa a un bibliotecario de la biblioteca. Su atributo adicional es la identificación del bibliotecario. Los métodos de esta clase permiten agregar materiales al inventario de una sucursal y transferir materiales entre sucursales.

Clase Sucursal

La clase Sucursal representa una biblioteca o una rama de la misma. Sus atributos incluyen el nombre de la sucursal, las listas de materiales, usuarios y bibliotecarios asociados. Los métodos de esta clase permiten registrar nuevos usuarios y buscar materiales específicos por su título.



ENLACE

<https://colab.research.google.com/github/Renm1090/Cinema-cafe-and-digital-library/blob/main/Biblioteca.ipynb>

```
● user1.prestar_material(libro3)
  fecha_devolucion = datetime.now() - timedelta(days=2)
  fecha_devolucion = datetime.now() - timedelta(days=2)
  user1.devolver_material(libro1, fecha_devolucion)

  user1.devolver_material(libro3, fecha_devolucion)

  user1.mostrar_estado()
```

✓ 0.0s

Crónica de una muerte anunciada prestado a Juan
Este material no pertenece a este usuario.
Crónica de una muerte anunciada devuelto por Juan
Usuario: Juan, Multas acumuladas: \$0
Materiales prestados:

```
biblio1.agregar_material(sucursal1, mdigital)
biblio1.agregar_material(sucursal1, mdigita2)
```

✓ 0.0s

Material 1984 añadido al sistema
Material Cien Años de Soledad añadido al sistema
Material Crónica de una muerte anunciada añadido al sistema
Material National Geographic añadido al sistema
Material Forbes añadido al sistema
Material Popular Science añadido al sistema
Material Curso de Python Avanzado añadido al sistema
Material Enciclopedia de Ciencia Moderna añadido al sistema