Eder Santiago Eraso Ortega - 22501966 Javier Esteban Bedoya Ruiz - 22501361 Ana Sofia Ledesma Garcia - 2251352 Johan David Vivas Esquivel - 22501603

Enlaces de github:

https://github.com/javier85bedoya-crypto/PROGRAMING

https://github.com/JohanVivas/project_cine/issues/1

https://github.com/santiagoerasso/Programming

https://github.com/sofialedesma011-hash/PROGRAMMING-PROYECT-1

Análisis:

- 1. Entradas (interacción del usuario)
- Menú principal:
 - o Opción seleccionada: entradas, clientes, cartelera, administración o salir.
- Registro o búsqueda de clientes:
 - Nombre completo del cliente.
 - ID único del cliente.
 - Número telefónico.
 - Activación opcional de tarjeta de fidelización (Sí/No).
 - El sistema genera automáticamente una tarjeta y define su nivel (Clásica, Premium o Élite) según el número de compras.
- Compra de entradas:
 - o Selección de película (por número o título en cartelera).
 - Elección de horario o sesión.
 - Selección de una o varias sillas (ejemplo: A1, B3, C5).
 - Selección de combos (Individual, Pareja o Familiar) y su cantidad.
 - o Identificación del cliente (para aplicar descuentos si tiene tarjeta activa).
 - Confirmación final de compra (Sí/No).
- Módulo de administración:
 - Generar reporte de ventas.
 - Visualizar estadísticas de ocupación y películas más vendidas.

- o Regresar al menú principal.
- Operaciones de mantenimiento:
 - Liberar sillas ocupadas o reservadas.
 - Reservar sillas temporalmente.
 - o Eliminar clientes del sistema.
 - 2. Salidas (resultados del sistema)
- Elementos mostrados al usuario:
 - Menús interactivos con colores (usando colorama) y emojis para una mejor experiencia.
 - Mapa visual de las sillas:
 - disponible
 - reservada
 - ocupada
 - ★ preferencial
 - Mensajes claros de confirmación, error o advertencia (al registrar, comprar o eliminar datos).
 - Información de cliente con su nivel de tarjeta y cantidad de compras.
 - Total de la compra, con el descuento aplicado según el nivel de tarjeta:
 - Clásica → 5%
 - Premium → 10%
 - Élite → 15%
- Informes y estadísticas administrativas:
 - Total de boletos vendidos.
 - Total de combos vendidos.
 - Película más vendida.
 - Sala más ocupada.
 - o Total recaudado por ventas.
 - 3. Restricciones del sistema

- Coordenadas de sillas válidas:
 - Filas entre A–J.
 - o Columnas válidas según tipo de sala:
 - $2D \rightarrow 15$ columnas.
 - 3D → 20 columnas.
- Identificación única de cliente:
 - o El ID no puede repetirse.
 - Solo se pueden realizar compras si el cliente existe o se acaba de registrar.
- Control de estados de sillas:
 - No se puede ocupar una silla ya reservada u ocupada.
 - o Si la compra se cancela o falla, las sillas deben liberarse automáticamente.
- Tarjeta de fidelización:
 - Activación voluntaria durante el registro.
 - Niveles automáticos según compras:
 - 1–3 → Clásica
 - $4-6 \rightarrow \text{Premium}$
 - 7 o más → Élite
 - o Descuento solo si la tarjeta está activa.
 - o Cada tarjeta pertenece a un solo cliente.
- Persistencia de datos:
 - Se realiza mediante archivos binarios .pkl usando pickle.
 - o Si los archivos no existen, el sistema genera datos por defecto.
 - 4. Descomposición funcional (por módulos y clases)

main.py

- Rol: punto de entrada del sistema.
- Funciones principales:

- o main():
 - Carga datos iniciales (clientes, películas, sesiones).
 - Crea el objeto principal SistemaCine.
 - Llama a la interfaz de usuario (mostrar_menu_principal).

interfaz_cli.py

- Rol: interfaz de línea de comandos que conecta al usuario con las operaciones internas.
- Funciones principales:
 - o Menús:
 - mostrar menu principal(sistema): muestra las opciones iniciales.
 - menu_entradas(sistema): proceso de compra de entradas.
 - menu_clientes(): gestión de clientes (registrar, buscar, eliminar, listar).
 - menu_admin(sistema): acceso a reportes y estadísticas.
 - Gestión de compras:
 - comprar_boletos(sesion, sistema): selecciona película, sillas, combos y calcula total.
 - registrar_cliente(): solicita datos y crea el cliente.
 - reservar_sillas(sesion), liberar_sillas(sesion): control de ocupación.
 - Validaciones:
 - validar_coordenada(coord, filas, columnas): asegura que la silla exista y tenga formato correcto.

models.py

- Rol: define todas las clases que representan las entidades del sistema.
- Clases principales:
 - Silla: representa cada asiento (estado y tipo).
 - Sala (abstracta) y subclases Sala2D, Sala3D: definen el mapa de asientos y precios.
 - o Pelicula: contiene la información y cartelera.

- Sesion: maneja horarios, ocupación y cálculo de totales.
- SistemaSesiones: administra todas las sesiones.
- TarjetaFidelizacion: controla descuentos y niveles.
- Cliente: almacena información personal y registra compras.
- Combo, Entrada, Compra: modelan los componentes de una transacción.

sistema_cine.py

- Rol: núcleo del sistema, encargado de la lógica de negocio y persistencia.
- Funciones principales:
 - __init__(): carga los datos o crea por defecto.
 - o registrar_compra(): crea una venta y la guarda.
 - guardar_ventas() / cargar_ventas(): persistencia de transacciones.
 - generar_estadisticas(): calcula totales de boletos, combos, películas y recaudación.
 - generar_reporte_ventas(): muestra informe general.
 - obtener_salas(): devuelve las salas activas.

PSEUDOCÓDIGO

INICIO PROGRAMA

IMPRIMIR "Bienvenido al Sistema de Gestión de Cine"

LLAMAR Cliente.cargar clientes()

LLAMAR Pelicula.cargar_cartelera()

LLAMAR SistemaSesiones.cargar sesiones()

CREAR sistema ← nuevo SistemaCine()

LLAMAR InterfazCLI.mostrar_menu_principal(sistema)

FIN PROGRAMA

FUNCIÓN mostrar_menu_principal(sistema)
REPETIR
IMPRIMIR opciones del menú principal
LEER opcion

```
SEGÚN opcion HACER
      CASO "1": menu_entradas(sistema)
      CASO "2": menu_clientes()
      CASO "3": Pelicula.mostrar_cartelera()
      CASO "4": menu admin(sistema)
      CASO "5":
         LLAMAR guardar_todos_los_datos(sistema)
         IMPRIMIR "Datos guardados correctamente."
         SALIR
      CASO "0":
         LIBERAR sillas temporales
         IMPRIMIR "Saliendo sin guardar."
         SALIR
      OTRO:
         IMPRIMIR "Opción no válida."
    FIN SEGÚN
  HASTA opcion = "0" O "5"
FIN FUNCIÓN
FUNCIÓN menu entradas(sistema)
  MOSTRAR cartelera numerada
  LEER indice_pelicula
  SI índice inválido → SALIR
  MOSTRAR horarios disponibles
  LEER horario_elegido
  SI horario inválido → SALIR
  sesion ← SistemaSesiones.buscar_sesion(pelicula, horario)
  SI NO EXISTE sesion:
    sesion ← nueva Sesion(pelicula, horario, sala_aleatoria)
    SistemaSesiones.agregar_sesion(sesion)
  MOSTRAR info de película y sala
  LLAMAR comprar_boletos(sesion, sistema)
FIN FUNCIÓN
FUNCIÓN comprar_boletos(sesion, sistema)
  MOSTRAR mapa de sillas disponibles
  lista_sillas ← []
  REPETIR
    LEER coordenada_silla (ejemplo A5)
    SI vacío → SALIR
```

```
SI silla DISPONIBLE:
       MARCAR silla como ocupada
       AGREGAR silla a lista sillas
    SINO:
       IMPRIMIR "Silla no disponible."
  HASTA que usuario deje de ingresar
  SI lista sillas VACÍA → CANCELAR compra
  lista combos ← []
  MOSTRAR combos disponibles
  REPETIR
    LEER tipo_combo, cantidad
    SI válido → AGREGAR combo a lista_combos
  HASTA finalizar selección
  IMPRIMIR "Ingrese ID del cliente:"
  LEER id cliente
  cliente ← Cliente.buscar_cliente_por_id(id_cliente)
  SI cliente NO EXISTE:
    PREGUNTAR "¿Desea registrarse? (s/n)"
    SI "s": cliente ← registrar_cliente()
    SINO: CANCELAR compra
  total ← sesion.calcular_total(lista_sillas, lista_combos, cliente)
  SI cliente.tiene tarjeta:
    descuento ← cliente.tarjeta.calcular_descuento(total)
    total ← total - descuento
    IMPRIMIR "Descuento aplicado por nivel", cliente.tarjeta.nivel_actual, ":", descuento,
"$"
  SINO:
    IMPRIMIR "Sin descuento aplicado."
  MOSTRAR resumen de compra (película, sillas, combos, total)
  PREGUNTAR "¿Confirmar compra? (s/n)"
  SI confirmado:
    sistema.registrar_compra(cliente, sesion, lista_sillas, lista_combos, total)
                               ← actualiza número de compras
    cliente.registrar_compra()
    SI cliente.tiene_tarjeta:
       cliente.tarjeta.registrar compra()
    IMPRIMIR "Compra registrada exitosamente."
  SINO:
    LIBERAR sillas seleccionadas
    IMPRIMIR "Compra cancelada."
FIN FUNCIÓN
```

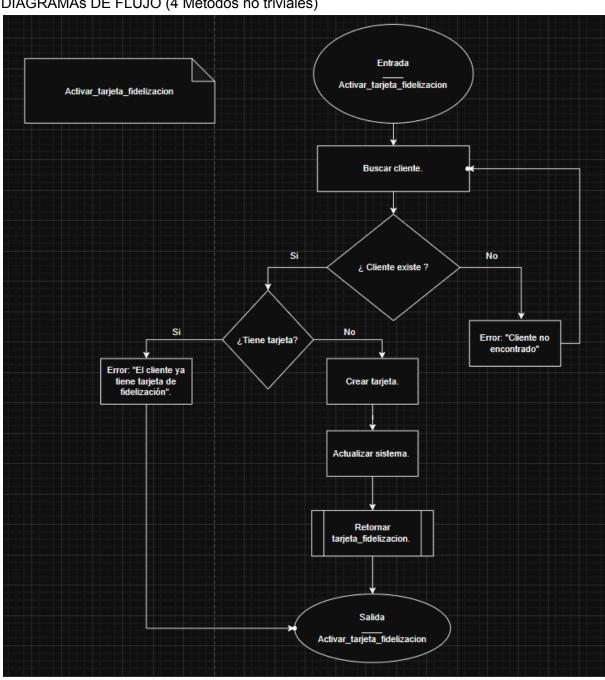
```
FUNCIÓN registrar cliente()
  LEER nombre, id_cliente, telefono
  SI Cliente.buscar cliente por id(id cliente) EXISTE → CANCELAR
  PREGUNTAR "¿Activar Tarjeta Fidelix?"
  LEER respuesta
  tiene tarjeta ← (respuesta == "s")
  nuevo ← Cliente(nombre, id_cliente, telefono, tiene_tarjeta)
  Cliente.registrar_cliente(nuevo)
  IMPRIMIR "Cliente registrado correctamente"
  RETORNAR nuevo
FIN FUNCIÓN
FUNCIÓN menu_clientes()
  REPETIR
    MOSTRAR opciones
    LEER opcion
    SEGÚN opcion HACER
       CASO 1: registrar_cliente()
       CASO 2: Cliente.ver clientes()
       CASO 3: LEER nombre → Cliente.buscar_por_nombre(nombre)
       CASO 4: LEER id → Cliente.buscar_cliente_por_id(id)
       CASO 5: LEER tel → Cliente.buscar por telefono(tel)
       CASO 6: LEER nivel → Cliente.buscar_por_nivel(nivel)
       CASO 7:
         LEER id → confirmar → eliminar si "s"
       CASO 0: SALIR
    FIN SEGÚN
  HASTA opcion = 0
FIN FUNCIÓN
Clase Cliente:
  ATRIBUTOS: nombre, id_cliente, telefono, tiene_tarjeta, tarjeta, compras
  registrar_compra():
    compras += 1
    SI tiene tarjeta → tarjeta.registrar compra()
  calcular_descuento(total):
    SI tiene tarjeta → retornar tarjeta.calcular descuento(total)
```

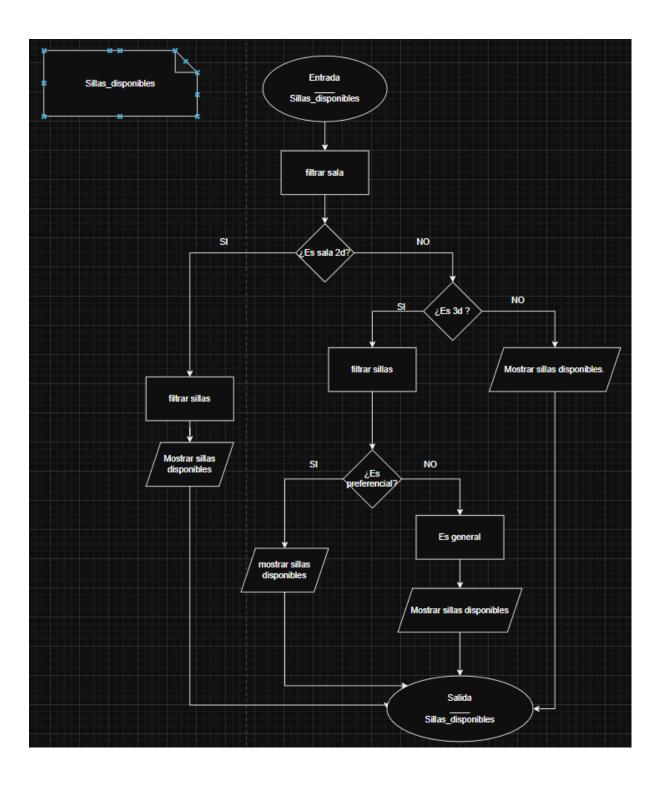
```
SINO → retornar total
  activar tarjeta():
    SI tarjeta NO EXISTE → tarjeta ← TarjetaFidelizacion()
Clase TarjetaFidelizacion:
  ATRIBUTOS: compras realizadas, nivel actual
  registrar_compra():
    compras_realizadas += 1
    nivel_actual ← obtener_nivel()
  obtener_nivel():
    SI compras ≤ 3 → "Clásica"
    SI 4 \le \text{compras} \le 6 \rightarrow \text{"Premium"}
    SI > 6 \rightarrow "Élite"
  calcular_descuento(total):
    SEGÚN nivel:
       CASO "Clásica" → 5%
       CASO "Premium" → 10%
       CASO "Élite" → 15%
    RETORNAR total - descuento
Clase SistemaCine:
  ATRIBUTOS:
    ventas[], sesiones[]
  CONSTRUCTOR:
    ventas ← cargar_ventas()
    sesiones ← SistemaSesiones.cargar sesiones()
    SI vacío → inicializar sesiones por defecto
  registrar_compra(cliente, sesion, sillas, combos, total):
    crear compra con todos los datos
    agregar a ventas
    guardar_ventas()
  guardar_ventas():
    serializar con pickle
  cargar_ventas():
    SI archivo existe → cargar datos
    SINO → retornar []
  generar estadisticas():
```

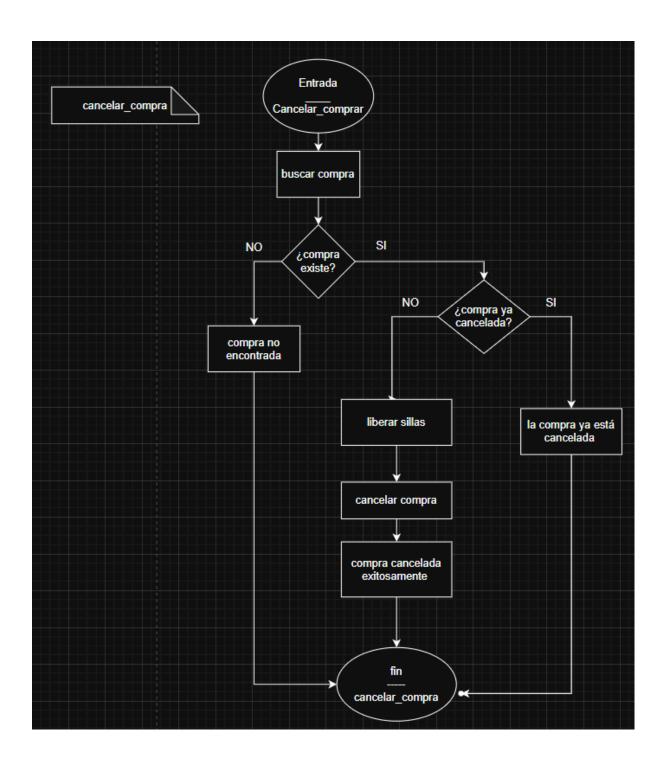
calcular total_boletos, total_combos, total_recaudo obtener película más vendida y ocupación promedio retornar resumen

generar_reporte_ventas(): stats ← generar_estadisticas() IMPRIMIR resumen con totales y porcentajes

DIAGRAMAs DE FLUJO (4 Métodos no triviales)







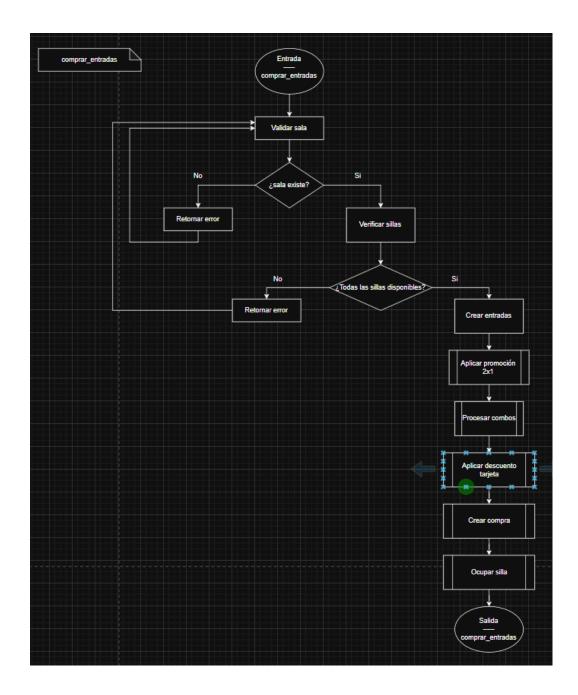
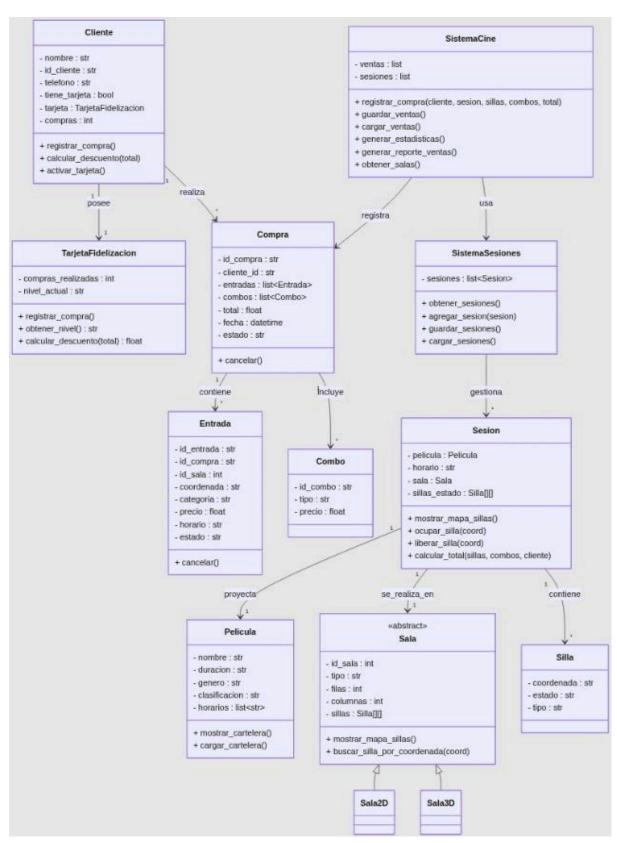
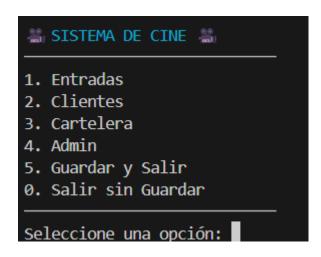


DIAGRAMA DE CLASES



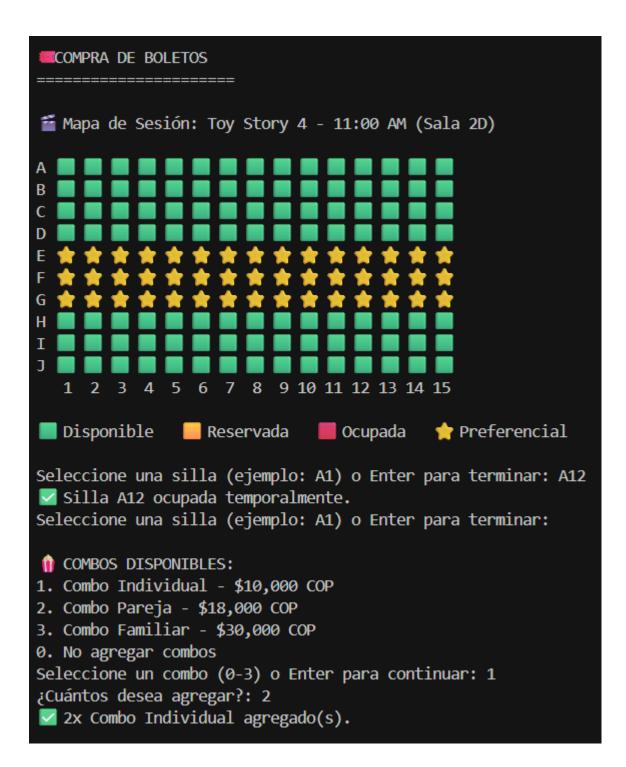
+

Funcionamiento Menú



Entradas

IIII SECCIÓN DE ENTRADAS IIII 1. Avengers: Endgame 2. Toy Story 4 3. Avatar: El camino del agua 4. Spider-Man: No Way Home 5. Encanto Seleccione la película (número): 2 Has seleccionado: Toy Story 4 Horarios disponibles: 1. 11:00 AM 2. 2:00 PM 3. 5:00 PM Seleccione el horario (número): 1 Película: Toy Story 4 (Horario: 11:00 AM masala: 2D



DATOS DEL CLIENTE Ingrese su ID: 1022536 No existe un cliente con ese ID. ¿Desea registrarse como nuevo cliente? (s/n): s REGISTRO DE NUEVO CLIENTE Ingrese el nombre completo: Sofia Ledesma Ingrese el ID del cliente: 1062292984 Ingrese el número de teléfono: 3219705363 ¿Desea activar tarjeta de fidelización? (s/n): s ✓ Cliente registrado exitosamente: 👤 Sofia Ledesma | ID: 1062292984 | Tel: 3219705363 | Tarjeta: Clásica 💟 🎁 Tarjeta creada para Sofia Ledesma. ¡Ahora eres miembro del programa de fidelización! Total a pagar: \$28,500.0 COP. ¿Confirmar compra? (s/n): s 🎉 ¡Compra realizada exitosamente para Sofia Ledesma! Sillas: A12 n Combos: 2x Combo Individual 💶 Total pagado: \$28,500.0 COP

Clientes

__ SISTEMA DE CINE __ 1. Entradas 2. Clientes 3. Cartelera 4. Admin 5. Guardar y Salir Salir sin Guardar Seleccione una opción: 2 MENÚ DE CLIENTES Registrar nuevo cliente 2. Ver clientes registrados 3. Buscar cliente por nombre 4. Buscar cliente por ID Buscar cliente por teléfono 6. Buscar cliente por nivel de tarjeta 7. Eliminar cliente Volver al menú principal Seleccione una opción: 2 CLIENTES REGISTRADOS: 👤 0 | ID: 0 | Tel: 0 | Sin tarjeta 🗶 👤 Sofia Ledesma | ID: 1062292984 | Tel: 3219705363 | Tarjeta: Clásica 🔽

```
Seleccione una opción: 3
n cartelera de hoy
1. Avengers: Endgame
   🕒 Duración: 3h 2m
   🦏 Género: Acción / Ciencia Ficción
   Clasificación: PG-13
   🧕 Horarios: 1:00 PM, 4:30 PM, 8:00 PM
2. Toy Story 4
   ( Duración: 1h 40m
   🧺 Género: Animación / Aventura
   (iii) Clasificación: ATP
   🧕 Horarios: 11:00 AM, 2:00 PM, 5:00 PM
3. 🖀 Avatar: El camino del agua
   Duración: 3h 12m
   🦏 Género: Fantasía / Aventura
   Clasificación: PG-13
   🧕 Horarios: 12:00 PM, 4:00 PM, 8:30 PM
4. 
Spider-Man: No Way Home
   🕒 Duración: 2h 28m
   🧺 Género: Acción / Superhéroes
   Clasificación: PG-13
   🧕 Horarios: 10:30 AM, 3:00 PM, 7:30 PM
5. Encanto
   🕒 Duración: 1h 42m
   🧺 Género: Animación / Musical
   Clasificación: ATP
   🧕 Horarios: 10:00 AM, 1:00 PM, 4:00 PM
```

admin

Seleccione una opción: 4 MENÚ DE ADMINISTRACIÓN 1. Generar reporte de ventas 0. Volver al menú principal Seleccione una opción: 1 REPORTE DE ESTADÍSTICAS Total boletos vendidos: 1 Total combos vendidos: 2 i Ingresos totales: \$28,500.0 COP Película más vendida: Toy Story 4 mSala con mayor ocupación: 2D

salir sin guardar

Seleccione una opción: 5

🖺 Datos guardados. ¡Gracias por visitar nuestro cine!

Estadísticas



■Total boletos vendidos: 1

↑ Total combos vendidos: 1

ingresos totales: \$19,000.0 COP

🟆 Película más vendida: Avengers: Endgame

mSala con mayor ocupación: 2D

Compra

Cliente registrado exitosamente:

👤 Javier | ID: 1 | Tel: 300 | Tarjeta: Clásica 🔽

🎁 Tarjeta creada para Javier. ¡Ahora eres miembro del programa de fidelización!

Total a pagar: \$19,000.0 COP. ¿Confirmar compra? (s/n): s

🎉 ¡Compra realizada exitosamente para Javier!

Sillas: A1

Combos: 1x Combo Individual
 Total pagado: \$19,000.0 COP