

Solucionarios



Semana 6

28.07.2025 - 03.08.2025



Caso General



Gestión de un Cine

Un cine necesita gestionar su cartelera, los horarios de las funciones, la venta de entradas y la disponibilidad de asientos. Un sistema informático es esencial para que los clientes puedan consultar información, comprar entradas online y para que la administración pueda analizar qué películas son más exitosas.

Nivel Fácil: Consultar Horarios de una Película



```
def consultar_horarios(cartelera, titulo_pelicula):
    """
    Busca una película en la cartelera y devuelve su lista de horarios.
    """
    # 1. Recorremos cada película (que es un diccionario) en la lista de la cartelera.
    for pelicula in cartelera:
        # 2. Comparamos el título de la película actual con el que estamos buscando.
        if pelicula['titulo'] == titulo_pelicula:
            # 3. Si encontramos la película, devolvemos su lista de horarios y la función termina.
            return pelicula['horarios']

    # 4. Si el bucle termina sin haber encontrado la película, devolvemos una lista vacía.
    return []

# --- Pruebas ---
def probar_consultar_horarios():
    cartelera = [
        {'titulo': 'Inception', 'horarios': ['18:00', '21:00']},
        {'titulo': 'The Matrix', 'horarios': ['17:30', '20:30']},
        {'titulo': 'Parasite', 'horarios': ['19:00', '22:00']}
    ]

    horarios_matrix = consultar_horarios(cartelera, 'The Matrix')
    print(f"Prueba 1: {horarios_matrix == ['17:30', '20:30']}")

    horarios_interstellar = consultar_horarios(cartelera, 'Interstellar')
    print(f"Prueba 2: {horarios_interstellar == []}")

    horarios_vacio = consultar_horarios([], 'Inception')
    print(f"Prueba 3: {horarios_vacio == []}")

    probar_consultar_horarios()
```

★ [Click para acceder al código](#) ★

Nivel Medio: Vender Entradas para una Sesión



```
def vender_entrada(sesiones, titulo_pelicula, horario, cantidad_entradas):  
    """  
    Vende una cantidad de entradas para una sesión específica si hay asientos disponibles.  
    """  
    # 1. Verificamos si la película existe en nuestro diccionario de sesiones.  
    if titulo_pelicula in sesiones:  
        sesiones_de_la_pelicula = sesiones[titulo_pelicula]  
  
        # 2. Si la película existe, verificamos si el horario específico está disponible.  
        if horario in sesiones_de_la_pelicula:  
            asientos_disponibles = sesiones_de_la_pelicula[horario]  
  
            # 3. Si el horario existe, verificamos si hay suficientes asientos.  
            if asientos_disponibles >= cantidad_entradas:  
                # 4. Si todas las comprobaciones son exitosas, restamos las entradas vendidas.  
                sesiones_de_la_pelicula[horario] -= cantidad_entradas  
                return True # La venta fue un éxito.  
  
            # 5. Si cualquiera de las comprobaciones anteriores falla, la función llega a este punto.  
            # No se modifica nada y se devuelve False para indicar que la venta no se pudo realizar.  
            return False  
  
# --- Pruebas ---  
def probar_vender_entrada():  
    sesiones_base = {  
        'Inception': {'18:00': 10, '21:00': 5},  
        'The Matrix': {'17:30': 25, '20:30': 1}  
    }  
  
    sesiones_p1 = {k: v.copy() for k, v in sesiones_base.items()}  
    resultado1 = vender_entrada(sesiones_p1, 'Inception', '18:00', 3)  
    print(f"Prueba 1 (Venta exitosa): {resultado1 is True and sesiones_p1['Inception']['18:00'] ==  
7}")  
  
    sesiones_p2 = {k: v.copy() for k, v in sesiones_base.items()}  
    resultado2 = vender_entrada(sesiones_p2, 'The Matrix', '20:30', 2)  
    print(f"Prueba 2 (Falta de asientos): {resultado2 is False and sesiones_p2['The Matrix']['20:30']  
== 1}")  
  
    sesiones_p3 = {k: v.copy() for k, v in sesiones_base.items()}  
    resultado3 = vender_entrada(sesiones_p3, 'Inception', '19:00', 1)  
    print(f"Prueba 3 (Horario no existe): {resultado3 is False}")  
  
probar_vender_entrada()
```

★ [Click para acceder al código](#) ★

Nivel Avanzado: Generar Reporte de Taquilla



```
def generar_reporte_taquilla(cartelera_completa):
    """
    Calcula los ingresos totales por película y devuelve un ranking ordenado.
    """
    # 1. Creamos una lista para almacenar los resultados (título e ingresos).
    lista_ingresos = []

    # 2. Recorremos cada película en la cartelera.
    for pelicula in cartelera_completa:
        titulo = pelicula['titulo']
        precio_entrada = pelicula['precio_entrada']
        ingresos_pelicula = 0.0

        # 3. Por cada película, recorremos sus diferentes sesiones.
        # Usamos .values() porque solo nos interesan los datos de la sesión (asientos), no la hora (clave).
        for sesion in pelicula['sesiones'].values():
            entradas_vendidas = sesion['vendidas']
            # 4. Calculamos los ingresos de esta sesión y los sumamos al total de la película.
            ingresos_pelicula += entradas_vendidas * precio_entrada

        # 5. Añadimos una tupla con el título y el total de ingresos a nuestra lista.
        lista_ingresos.append((titulo, ingresos_pelicula))

    # 6. Ordenamos la lista de ingresos.
    # - key=lambda item: item[1] le dice que ordene usando el segundo elemento de la tupla (los ingresos).
    # - reverse=True ordena de mayor a menor.
    reporte_ordenado = sorted(lista_ingresos, key=lambda item: item[1], reverse=True)

    # 7. Devolvemos el reporte final ordenado.
    return reporte_ordenado

# --- Pruebas ---
def probar_generar_reporte_taquilla():
    cartelera = [
        {'titulo': 'Oppenheimer', 'precio_entrada': 12.50, 'sesiones': {'19:00': {'vendidas': 80}, '22:00': {'vendidas': 90}}},
        {'titulo': 'Barbie', 'precio_entrada': 10.00, 'sesiones': {'18:30': {'vendidas': 110}, '21:30': {'vendidas': 115}}},
        {'titulo': 'Gran Turismo', 'precio_entrada': 11.00, 'sesiones': {'20:00': {'vendidas': 40}}}
    ]

    reporte = generar_reporte_taquilla(cartelera)
    esperado1 = [
        ('Barbie', 2250.0),
        ('Oppenheimer', 2125.0),
        ('Gran Turismo', 440.0)
    ]
    print(f"Prueba 1: {reporte == esperado1}")

    reporte2 = generar_reporte_taquilla([])
    print(f"Prueba 2: {reporte2 == []}")

    probar_generar_reporte_taquilla()
```

★ [Click para acceder al código](#) ★



Éxitos en los ejercicios

Un buen diseño agrega valor más rápido de lo que agrega costo

Thomas C. Gale

Aprende mucho