**Especificación de Requisitos Funcionales:**

1. **Comprar Entrada:**
   * Descripción: Permite al usuario comprar una entrada para una butaca específica, reservándola con un número de socio. Se pueden agregar hasta 3 complementos.
   * Flujo:
     1. El usuario selecciona una butaca disponible.
     2. El usuario ingresa su número de socio.
     3. El usuario selecciona los complementos deseados.
     4. Se genera un archivo HTML que contiene los detalles de la entrada (butaca, socio, complementos y fecha de compra).
2. **Devolver Entrada:**
   * Descripción: Permite al usuario devolver una entrada previamente comprada, liberando los recursos asociados.
   * Flujo:
     1. El usuario ingresa el número de la entrada a devolver.
     2. Se libera la butaca y los complementos asociados.
3. **Estado del Cine:**
   * Descripción: Muestra el estado actual del cine, incluyendo la disponibilidad de las butacas y su estado (libre, reservada u ocupada).
   * Flujo:
     1. Se muestra una representación visual del cine con el estado de cada butaca.
4. **Obtener Recaudación:**
   * Descripción: Calcula la recaudación total del cine en una fecha específica.
   * Flujo:
     1. El usuario ingresa una fecha válida en formato AAAA/MM/DD.
     2. Se calcula la suma total de las ventas realizadas en esa fecha.
5. **Importar Complementos:**
   * Descripción: Importa complementos desde un archivo CSV dado y los agrega a la base de datos.
   * Flujo:
     1. Se lee el archivo CSV.
     2. Se importan los complementos a la base de datos.
6. **Exportar Estado del Cine:**
   * Descripción: Exporta el estado actual del cine en un archivo JSON, para una fecha específica.
   * Flujo:
     1. El usuario ingresa una fecha válida en formato AAAA/MM/DD.
     2. Se genera un archivo JSON con el estado del cine en esa fecha.
7. **Configurar Butacas:**
   * Descripción: Configura las butacas del cine utilizando un archivo CSV dado.
   * Flujo:
     1. Se lee el archivo CSV.
     2. Se configuran las butacas según la información proporcionada.
8. **Actualizar Butaca:**
   * Descripción: Permite cambiar la información de una butaca dada su identificador.
   * Flujo:
     1. El usuario ingresa el identificador de la butaca a actualizar.
     2. Se actualiza la información de la butaca según los datos proporcionados.

**Especificación de Requisitos No Funcionales:**

1. **Base de Datos SQLite:**
   * La aplicación usa SQLite para gestionar los datos.
2. **Git y GitHub:**
   * Para este proyecto colaborativo usamos un repositorio en el que vamos subiendo en distintas ramas la implementación de la app.
3. **Mecanismos de Inyección de Dependencias:**
   * Usamos mecanismos automatizados de inyección de dependencias para mejorar la modularidad y la testabilidad del código.
4. **Manejo de Excepciones Adaptadas al Dominio:**
   * Implementación de excepciones adaptadas al dominio para manejar errores de manera eficiente y clara.

**Especificación de Requisitos de Información:**

1. **Tablas de la Base de Datos:**
   * **Butacas**: Para almacenar información sobre las butacas.
     + ID: FC (fila, columna)
     + Tipo: normal o VIP
     + Estado: Activa, Ocupada o en Mantenimiento
     + Ocupamiento: Libre u Ocupada
     + IsDeleted: Booleano
     + CreatedAt: fecha de creación
     + UpdatedAt: fecha de actualización
   * **Complementos**: Para almacenar información sobre los complementos/productos disponibles.
     + ID: Identificador único del complemento
     + Nombre: Nombre del complemento o producto
     + Precio: Precio del complemento
     + Stock: Cantidad disponible en stock
     + Tipo: Tipo de complemento
     + IsDeleted: Booleano
     + CreatedAt: fecha de creación
     + UpdatedAt: fecha de actualización
   * **Ventas**: Para almacenar información sobre las entradas vendidas, incluyendo los detalles de la butaca, el socio y la fecha de compra.
     + ID: Identificador único de la venta
     + ID\_Butaca: ID de la butaca asociada a la venta
     + ID\_Socio: ID del socio asociado a la venta
     + IsDeleted: Booleano
     + CreatedAt: fecha de creación
     + UpdatedAt: fecha de actualización
   * **Cuentas:** Con la información de usuarios que entran para poder comprar la entrada
     + ID: Identificador único de la cuenta
     + IsDeleted: Booleano
     + CreatedAt: fecha de creación
     + UpdatedAt: fecha de actualización
   * **LineaVenta**: para relacionar el Usuario con su venta tanto de butaca como de complementos
     + ID: Identificador único de la línea de venta
     + ID\_Venta: ID de la venta asociada a la línea
     + ID\_Complemento: ID del complemento asociado a la línea
     + Cantidad: Cantidad del complemento vendido
     + Precio: Precio unitario del complemento
     + IsDeleted: Booleano
     + CreatedAt: fecha de creación
     + UpdatedAt: fecha de actualización
2. **Integridad de Claves Primarias y Referenciales:**
   * Se debe garantizar la integridad de las claves primarias y referencias entre las tablas para mantener la consistencia de los datos.
3. **Formato de Archivos Generados:**
   * Los archivos generados (HTML, JSON) deben seguir un formato específico que contenga la información relevante de manera clara y estructurada.