**PROYECTO FINAL**



**Christian Gregorio**

**Diego Novillo**

**Mario de Domingo**

**ÍNDICE**

**Portada…………………………………………………………………………….Pág. 1**

**Índice……………………………………………………………………………….Pág. 2**

**Requisitos………………………………………………………………………...Pág. 3**

**Funcionales……………………………………………………………..Pág. 3**

**No Funcionales…………………………………………………………Pág. 4**

**De Información………………………………………………………...Pág. 5**

**Casos de USO…………………………………………………………………....Pág. 8**

**Comprar entrada……………………………………………………….Pág. 8**

**Gestión de productos…………………………………………………Pág. 9**

**Editar………………………………………………………………Pág. 9**

**Añadir…………………………………………….……………..Pág. 10**

**Borrar........…………………………………………………….Pág. 11**

**Importar…………………………………………………………Pág. 12**

**Costes……………………………………………………………………………..Pág. 13**

**Informe…………………………………………………………………..Pág. 13**

**Coste Total………………………………………………………………Pág. 14**

**Diagramas………………………………………………………………………..Pág. 15**

**Casos de USO………………………………………………………….Pág. 15**

**Clases…………………………………………………………………….Pág. 16**

**Secuencia……………………………………………………………….Pág. 17**

**Comprar entrada…………………………………………….Pág. 17**

**CRUD…………………………………………………………….Pág. 18**

**Create………………………………………………….Pág. 18**

**Read…………………………………………………….Pág. 19**

**Update………………………………………………….Pág. 20**

**Delete…………………………………………………..Pág. 21**

**Interfaces y Navegabilidad.…………………………………………………Pág. 22**

**Especificación de Requisitos Funcionales:**

1. **Comprar Entrada:** Permite al usuario comprar una entrada para una o varias butacas, reservándola cos un usuario.
   * 1. El usuario selecciona una butaca disponible.
     2. El usuario selecciona los complementos deseados.
     3. Se genera la entrada (un archivo HTML) que contiene los detalles (butacas, socio, complementos y fecha de compra).
2. **Inicio de Sesión:** permite al usuario usar una cuenta propia con sus datos
   * 1. El usuario puede registrarse
     2. El usuario puede iniciar sesión con una cuenta previamente registrada
     3. El usuario puede ser de tipo admin y de tipo cliente
3. **Devolver Entrada:** Permite al usuario devolver una entrada previamente comprada, liberando los recursos asociados.
   * 1. El usuario ingresa el número de la entrada a devolver.
     2. Se libera la butaca y los complementos asociados.
4. **Estado del Cine:** Muestra el estado actual del cine, incluyendo la disponibilidad de las butacas y su estado (libre, reservada u ocupada).
   * 1. Se muestra una representación visual del cine con el estado de cada butaca.
5. **Obtener Recaudación:** Calcula la recaudación total del cine en una fecha específica.
   * 1. El usuario ingresa una fecha válida en formato AAAA/MM/DD.

Se calcula la suma total de las ventas realizadas en esa fecha.

1. **Importar Complementos:** Importa complementos desde un archivo CSV dado y los agrega a la base de datos.
   * 1. Se lee el archivo CSV.
     2. Se importan los complementos a la base de datos.
2. **Exportar Estado del Cine:** Exporta el estado actual del cine en un archivo JSON, para una fecha específica.
   * 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.
3. **Configurar Butacas:** Configura las butacas del cine utilizando un archivo CSV dado.
   * 1. Se lee el archivo CSV.
     2. Se configuran las butacas según la información proporcionada.
4. **Actualizar Butaca:** Permite cambiar la información de una butaca.
   * 1. El administrador selecciona la butaca a actualizar.
     2. Se actualiza la información de la butaca según los datos proporcionados.
5. **Gestión de Usuarios:** se pueden añadir borrar y editar usuarios
   * 1. un administrador debe poder:
        1. leer usuario
        2. borrar usuario
        3. crear usuario
        4. editar/actualizar usuario
     2. un usuario debe poder ver su cuenta y editar sus datos
6. **Gestión de Productos:** se pueden añadir borrar y editar productos
   * 1. un administrador debe poder:
        1. leer Producto
        2. borrar Producto
        3. crear Producto
        4. editar/actualizar Producto

**Especificación de Requisitos No Funcionales:**

1. **Base de Datos:**
   * La aplicación usa SQLite para gestionar los datos.
   * Base de datos relacional
   * Uso de un manejador de base de datos y mapeo manual
2. **Uso de JavaFx e Interfaces**
   * Cada vista o pantalla consta de un archivo .fxml
   * En caso de excepción o error sale una ventana de aviso.
3. **Importación y exportación de datos CSV**
   * Al inicio se importar a un estado por defecto del cine, con el estado de las butacas, así como los productos y usuarios activos
4. **Git y GitHub:**
   * Para este proyecto colaborativo usamos un repositorio en el que vamos subiendo en distintas ramas la implementación de la app.
5. **Mecanismos de Inyección de Dependencias:**
   * Usamos mecanismos automatizados de inyección de dependencias para mejorar la modularidad y la testabilidad del código.
6. **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
     + Precio: Int,
     + Id\_venta: String (clave ajena)
     + 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
     + **Nombre:** String
     + **Correo:** String
     + **Contraseña:** String
     + **Tipo:** (administrador, cliente)
     + **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.
4. **Diseño:**
   * **Tres Resoluciones:**
     + estándar (1280\*800),
     + registro de usuario
     + gestión
     + inicio sesión, reset pasword:

**CU: Comprar Entrada**

**Requisitos Funcionales:**

1. Comprar entrada

2. Inicio de Sesión

4. Estado del cine

**Autores :**  Mario, Diego y Christian

**Version:** 1.0

**Actores que intervienen:**

* Cine (sistema)
* Usuario (comprador de entradas)

**Descripción:**

* El sistema deberá permitir al usuario adquirir una entrada para el cine, reservando las butacas elegidas disponibles y, opcionalmente, añadiendo complementos.

**Precondición:**

* Debe estar alguna butaca disponible.

**Secuencia Normal:**

1. El usuario selecciona la sala (en este caso solo una)
2. El usuario selecciona las butacas libres disponibles
3. El usuario escoge o no los complementos.
4. El usuario inicia sesión o se registra
   1. Si el usuario es correcto, el sistema reserva las butacas.
5. El usuario introduce los datos de pago.
   1. Se verifican los datos.
6. Se Genera la entrada con sus complementos.
7. El sistema actualiza la base de datos con la información de la venta.

**Secuencia Alternativa:**

2.1. Si no quedan butacas disponibles, el sistema muestra un mensaje de error y no se podría continuar.

* 1. Si el usuario no se registra correctamente, el sistema muestra un mensaje de error y el sistema, el sistema regresa al punto 4.
  2. Si los datos para formalizar la compra son incorrectos o están incompletos, el sistema muestra un mensaje de error y se solicita los datos restantes o modificación de los dados.

**Postcondición:**

* Las butacas están reservadas y el usuario tiene un fichero de entrada.
* La base de datos ha sido actualizada con la información de la venta.

**CU:** **Edición de productos**

**Requisitos Funcionales:**

2. Inicio de Sesión

11. Gestión de Productos

**Autores :**  Mario, Diego y Christian

**Version:** 1.0

**Actores que intervienen:**

* Cine (sistema)
* Usuario (Administrador)

**Descripción:**

* El Sistema debe permitir al administrador poder editar productos si así lo desea

**Precondición:**

* El sistema está activo y el usuario tiene credenciales de administrador.

**Secuencia normal:**

* 1. El usuario elige la opción de gestionar productos.
  2. El sistema muestra la lista de productos disponibles.
  3. El usuario selecciona el producto que desea editar.
  4. El sistema muestra la información del producto seleccionado.
  5. El usuario edita la información del producto.
  6. El usuario guarda los cambios realizados en el producto.
  7. El sistema valida los datos ingresados por el usuario.
     1. Si los datos son correctos, el sistema guarda la información actualizada del producto.
  8. El sistema muestra un mensaje de confirmación de que los cambios han sido guardados correctamente.

**Secuencia Alternativa:**

8.1. Si los datos ingresados por el usuario no son válidos, el sistema muestra un mensaje de error y permite al usuario corregir los datos.

**Postcondición:**

* Los cambios realizados en el producto son guardados correctamente en el sistema.

**Caso de Uso: Añadir Producto**

**Requisitos Funcionales:**

1. Inicio de Sesión
2. Gestión de Productos

**Autores:** Mario, Diego y Christian

**Actores:**

* Cine (sistema)
* Usuario (Administrador)

**Descripción:**

El Sistema permite al administrador agregar nuevos productos al sistema.

**Precondición:**

El sistema está activo y el usuario tiene credenciales de administrador.

**Secuencia Normal:**

1. El usuario elige administrar productos.
2. El usuario elige la opción de añadir un nuevo producto.
3. El sistema muestra un formulario vacío para ingresar los detalles del nuevo producto.
4. El usuario completa los datos del formulario.
5. El usuario confirma la creación del producto.
6. El sistema valida los datos.
   1. Si los datos son correctos, el sistema guarda la información del nuevo producto.
7. El sistema muestra un mensaje de confirmación de que el producto ha sido añadido correctamente.

**Secuencia Alternativa:**

6.1. Si los datos ingresados por el usuario no son válidos, el sistema muestra un mensaje de error y permite al usuario corregir los datos, volver al paso3.

**Postcondición:**

* El nuevo producto es añadido al sistema correctamente.

**CU: Borrar Producto**

**Requisitos Funcionales:**

1. Inicio de Sesión
2. Gestión de Productos

**Autores:** Mario, Diego y Christian

**Actores:**

* Cine (sistema)
* Usuario (Administrador)

**Descripción:**

El Sistema permite al administrador eliminar productos existentes del sistema.

**Precondición:**

* El sistema está activo y el usuario tiene credenciales de administrador.

**Secuencia Normal:**

1. El usuario elige administrar productos.
2. El usuario selecciona un producto de la lista.
3. El usuario elige la opción de eliminar el producto.
4. El sistema solicita confirmación al usuario para eliminar el producto.
5. El usuario confirma la eliminación del producto.
6. El sistema elimina el producto seleccionado de la base de datos.
7. El sistema muestra un mensaje de confirmación de que el producto ha sido eliminado correctamente.

**Secuencia Alternativa:**

2.1. En el caso de que no haya productos, no se podrá eliminar ninguno por tanto el caso de uso finaliza.

**Postcondición:**

* El producto seleccionado es eliminado del sistema correctamente.

**CU Importar productos desde un CSV**

**Requisitos Funcionales:**

1. Inicio de Sesión
2. Gestión de Productos

6. Importar Complementos

**Autores:** Mario, Diego y Christian

**Actores:**

* Cine (sistema)
* Usuario (Administrador)

**Descripción:**

El Sistema permite al administrador importar productos desde un archivo CSV al sistema.

**Precondición:**

* El sistema está activo.
* El usuario tiene credenciales de administrador.

**Secuencia Normal:**

1. El usuario elige administrar productos.
2. El usuario elige la opción de importar productos.
3. El sistema solicita al usuario que seleccione el archivo CSV a importar.
4. El usuario selecciona el archivo CSV.
5. El sistema procesa el archivo CSV.
6. El usuario confirma la importación de los productos.
7. El sistema verifica los datos de los productos a importar.
   1. Si los datos son correctos, el sistema añade los productos al sistema.
8. El sistema muestra un mensaje de confirmación de que los productos han sido importados correctamente.

**Secuencia Alternativa:**

7.2. Si hay errores en los datos de los productos en el archivo CSV, el sistema muestra un mensaje de error y permite al usuario corregir los datos o coger otro archivo, volver al paso 3.

**Postcondición:**

* Los productos del archivo CSV son importados al sistema correctamente.

**Informe de desglose de costes asociados al desarrollo del proyecto**

**Salario de los Desarrolladores:**  
3 desarrolladores × 13.36€/hora × 8 horas/día × 5 días/semana × 6 semanas = **9,619.2€  
  
  
Coste Herramientas de Desarrollo:  
-**IntelliJ IDEA Ultimate. Suscripción 599€ al año por usuario. Coste para 6 semanas (aproximadamente 1/8.67 de un año):  
3 desarrolladores × 599€/año × 1/8.67= 207.21€

-Balsamiq wireframes. Suscripción 90€/Año. Coste para 6 semanas (aproximadamente 1/8.67 de un año):  
3 desarrolladores × 90€/año × 1/8.67 = 31.14€  
  
Total coste herramientas desarrollo para 3 desarrolladores: **238.35 €**  
  
  
**Coste amortización herramienta equipos portátiles:**Con una vida aproximada de 3 años (36 meses). Calculo el coste de amortización mensual de los dos equipos portátiles valorados cada uno en 1500€, dando como resultado:

3 equipos portátiles × 1500€/equipo / 36 meses = 125€ /mes  
  
Para un proyecto de 6 semanas (aproximadamente 1.5 meses), el costo de amortización resultante es: 125€/mes × 1.5 meses = **187.5€  
  
  
Costo de Pruebas:**Porcentaje de tiempo dedicado a test para garantizar eficiencia, estabilidad y seguridad de la aplicación: 20% de salario desarrolladores.

20% de salario desarrolladores. 9,619.2€ × 20% = **1,923.84€  
  
  
Costo de Mantenimiento:**  
15% del costo total del proyecto por año. Costo de mantenimiento para el primer año de servicio de la aplicación sería:   
  
(9,619.2€ + 238.35€ + 1,923.84€) × 15% = **1,746.24€  
  
  
Coste del servidor:**Estimado en 50€ al mes para un servicio del primer año:

50€/mes × 12 = 600€

Por lo tanto, el costo total estimado del proyecto sería:   
  
9,619.2€ (salario de los desarrolladores) +   
238.35€ (costo de las herramientas de desarrollo) +   
1,923.84€ (costo de pruebas) +   
1,746.24€ (costo de mantenimiento) +   
187.5€ (Amortización equipos) +   
600€ (servidor) = 14,315.13€

**Margen de beneficio: 25%.**14,315.13€ x 0.25 = 3578.8€ de beneficio.

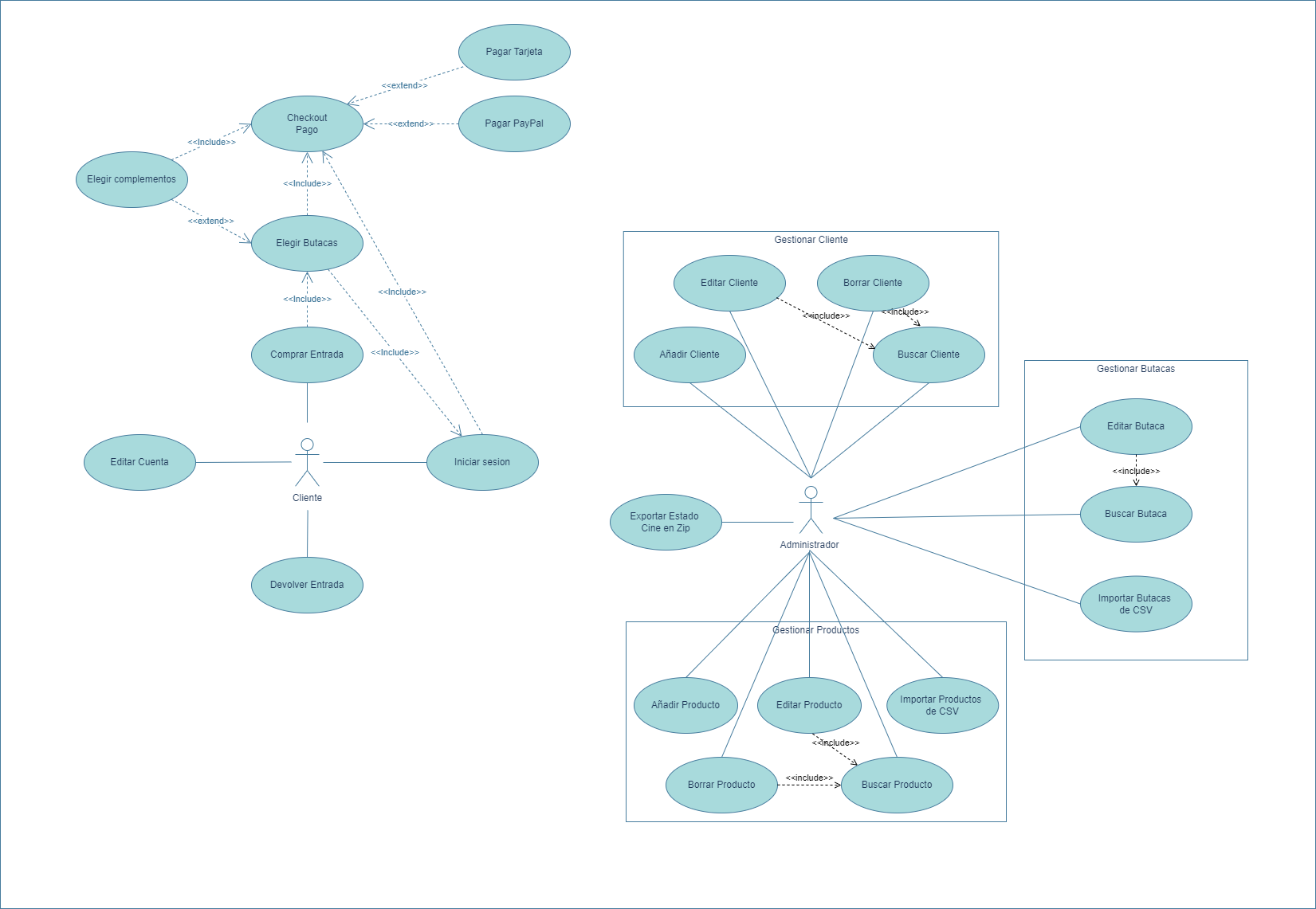
**Coste final para cliente**

**Precio total cobrado a cliente:**  
  
14,315.13€ (costo total del proyecto) + 14,315.13€ × 25% (beneficio) = 17,893.91€

17,893.91€ (precio total antes del IVA) \* 21% (IVA) = 3,757.72€

17,893.91€ + 3,757.72€ = **TOTAL IVA INCLUIDO 21,651.63€**

**Diagramas de Casos de USO**

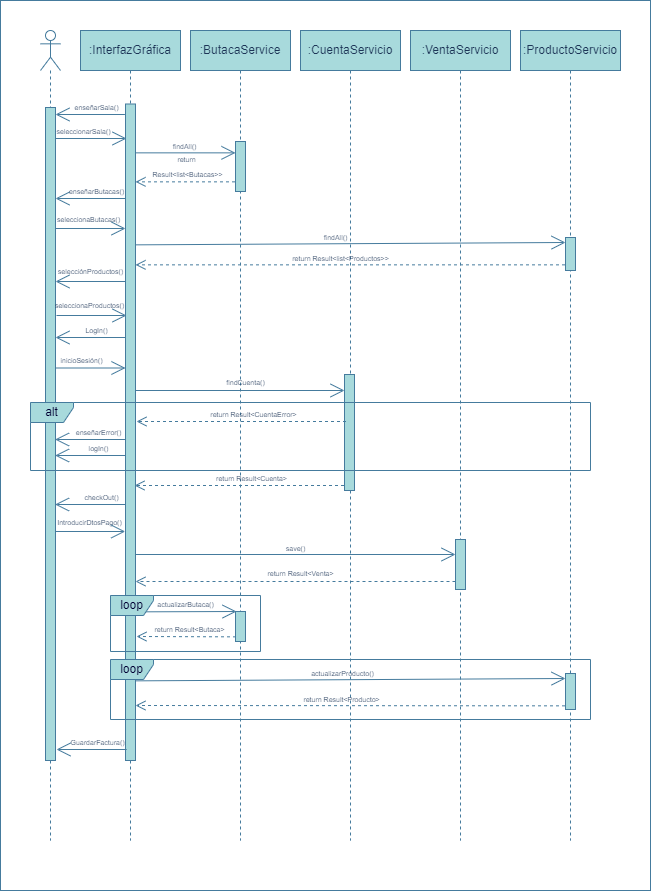


**Diagrama de Clases**

Diagrama

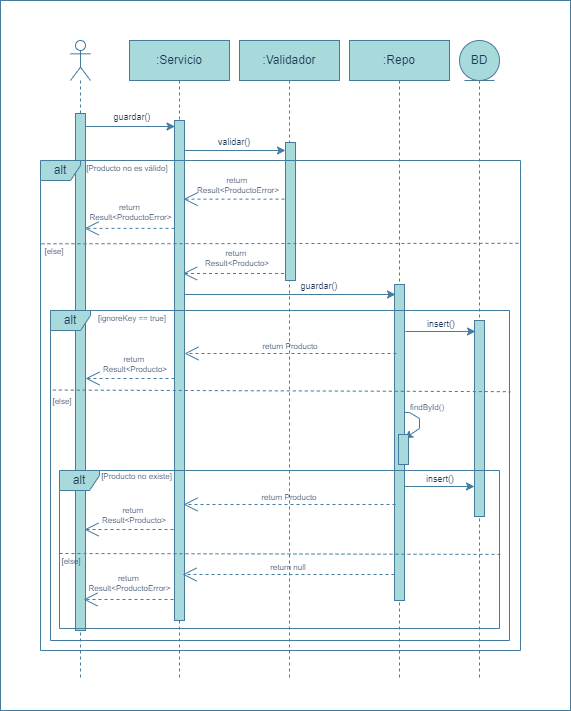
Descripción generada automáticamente

**Diagrama de Secuencia Comprar Entrada**



**Diagramas de Secuencia CRUD**

**Create**



**Read**

Diagrama

Descripción generada automáticamente

**Update**

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

**Delete**

Diagrama

Descripción generada automáticamente

**Interfaces y Navegabilidad**

Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

[Enlace al proyecto en Balsamiq](https://balsamiq.cloud/sg9wqbg/pn57gjx)