



Alumno: Renata Vidal

Materia: Trabajo de diploma

Docente: Julián Martín Rodríguez Escobedo

Universidad: UAI

Localización: Castelar

Comisión: 3A

Turno: Noche

# Índice

<b>Negocio</b>	<b>30</b>
Análisis general del sistema	30
Requisitos	32
Requisitos Funcionales:	32
Requisitos No Funcionales:	34
Estándares Aplicables:	35
Requisitos de Sistema:	35
Requisitos de Desempeño:	35
Requisitos de Entorno:	35
Descripción de personas participantes	36
Diagrama de casos de uso	37
Especificaciones de casos de uso	37
Diagrama de actividad por carriles	48
Diagrama de secuencia	49
Diagrama de clases	50
Modelo de datos	51
Modelo Conceptual	51
Diagramas de secuencia	52
Crear Balneario	52
Var Balnearios	53
Borrar Balneario	53
Planificación del proyecto	54

# T01 Arquitectura del Sistema

¿Qué es?

La arquitectura del sistema indica la estructura, funcionamiento e interacción entre las partes del software.

La arquitectura en capas es un enfoque de diseño de software que separa la aplicación en capas lógicas y funcionales. Cada capa tiene un conjunto de responsabilidades y tareas específicas por cumplir, lo que permite poder realizar una mejor organización del código y una mayor facilidad de mantenimiento.

¿Cómo está conformado nuestro sistema?

Nuestro sistema se divide en capas. En otras palabras, está compuesto por subsistemas. Este sistema está compuesto por 7 capas, las cuales son UI, BE, BLL, MPP, DAL, Servicios y Abstracción.

Además de que nuestro sistema está conectado a una base de datos que nos permite almacenar, recuperar y gestionar información. Trayéndonos ventajas como son la eficiencia, seguridad, accesibilidad y escalabilidad.

Esta se conecta al sistema a través de la DAL ya que esta cumple un rol de intermediaria. Proporcionándonos un sistema consistente y unificado capaz de acceder a los datos almacenados en la base de datos.

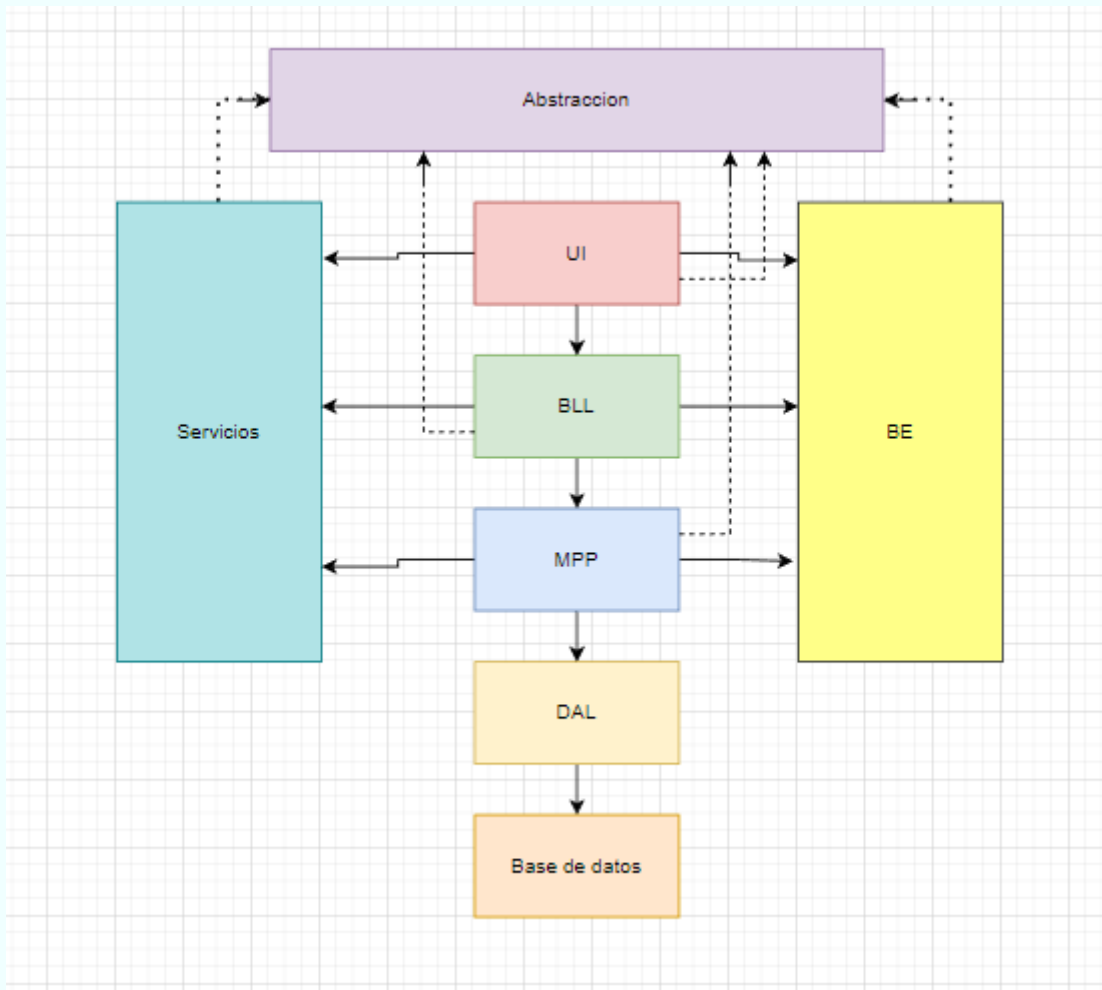


Diagrama de la arquitectura del sistema

Estas capas se conectan a través de interfaces y servicios, lo que permite a cada capa comunicarse con las capas cercanas de manera segura y eficiente.

La capa de usuario(UI): Esta capa se encarga de presentar la interfaz gráfica de usuario y procesar los datos de entrada de usuario. Esta capa se comunica con la BE, la BLL y la capa de servicios.

La capa de entidades de negocio(BE): esta capa se encarga de representar las entidades y objetos de negocio del sistema. Esta define cómo se modelan estas entidades y cómo se relacionan entre sí. Está conectada con la UI, la BLL, el MPP y la capa de Abstracción

La capa de lógica de negocios(BLL): esta capa contiene la lógica del negocio y las reglas con las que definen cómo se deben procesar los datos. Esta capa se comunica con la BE, el MPP, la capa de abstracción, la de servicios y la UI.

La capa de procesamiento de Datos(MPP): esta capa se utiliza para procesar los datos antes de que se envíen a la DAL o después de recibirlos de la misma. El MPP se comunica con la DAL, la BLL, la BE, la capa de abstracción y la capa de servicios

La capa de acceso a datos (DAL): esta capa se utiliza para acceder a la base de datos y realizar las operaciones de escritura y de lectura. La DAL se comunica con el MPP y la base de datos.

La capa de servicios se encarga de proveer todo tipo de servicios al sistema, un ejemplo de estos servicios es la seguridad que es utilizada por ejemplo para el encriptado de la contraseña.

La capa de Abstracción es una interfaz en donde declaramos una propiedad que es "id" que se utiliza para definir las característica básica de una entidad genérica. Esta capa está conectada con la BE.

La Base de datos se compone de tablas, que a su vez está compuesto por columnas y filas. Los usuarios pueden realizar consultas en la base de datos para buscar y recuperar información específica. Es fundamental ya que nos permite almacenar, recuperar y gestionar información del sistema.

#### Justificación de elección de arquitectura:

Elegimos realizar el sistema con este tipo de arquitectura porque separa claramente las responsabilidades del sistema en capas separadas, lo que en un futuro hace fácil el mantenimiento y la actualización del sistema, logrando un sistema mucho más organizado y más sencillo a la comprensión, logrando ser muy atractivo a la vista a comparación de otras arquitecturas con menos cantidad de capas. Además de que esta arquitectura es escalable a largo plazo.

También esta arquitectura es muy útil para que en el futuro cuando planteemos la propuesta de proyecto nos permita cumplir con los requisitos del sistema y del negocio.

A la arquitectura del sistema le agregamos las capas de Mapper y de abstracción debido a que permite integrar diferentes sistemas, sin afectar la lógica del negocio. Nos resultan bastante útiles ya que nos ayuda a separar la lógica de negocio de los detalles de implementación, oculta la complejidad, mejora la reutilización de componentes y facilita la integración.

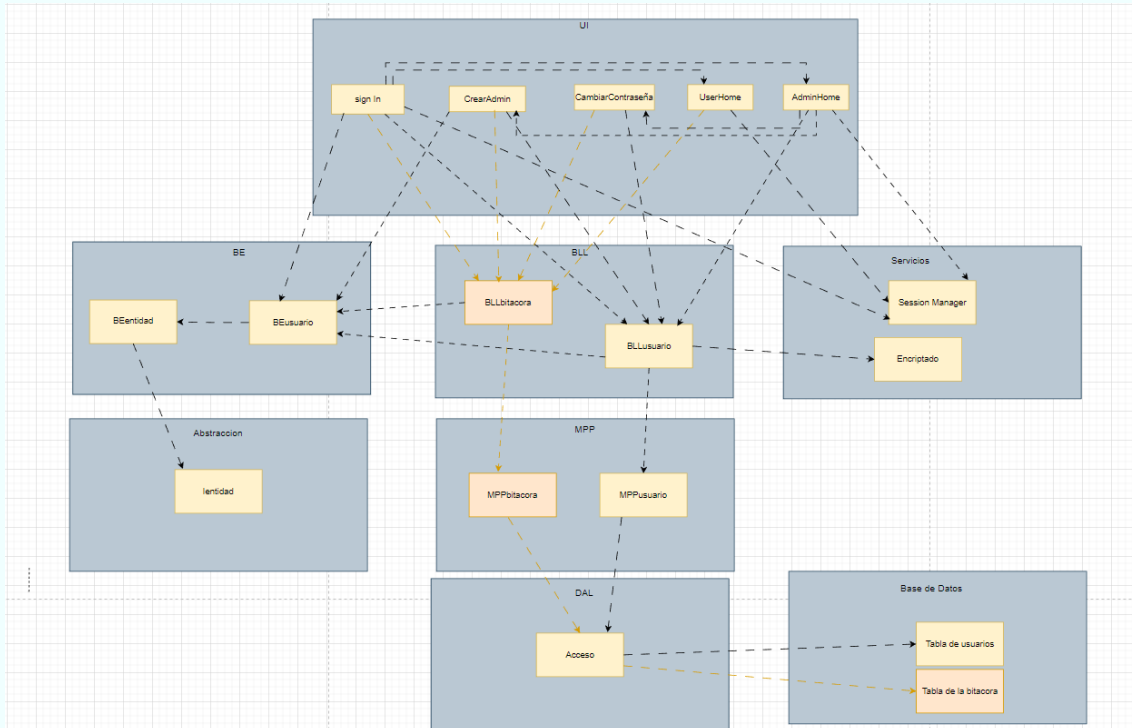


Diagrama de componentes de sign in

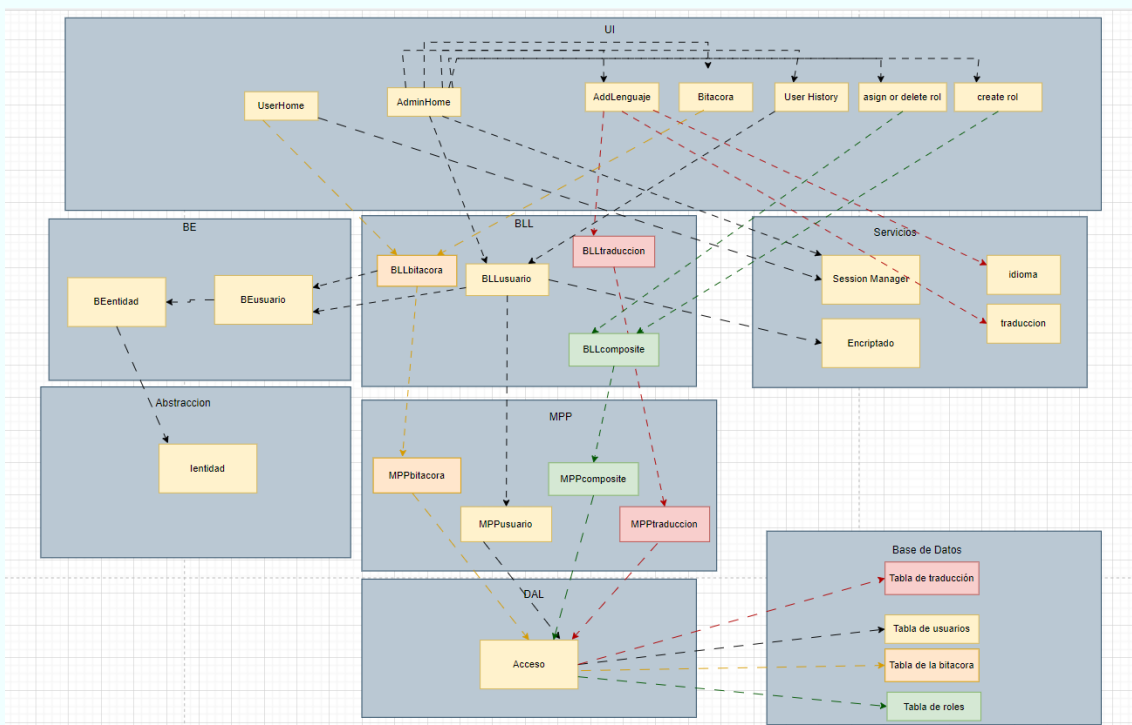
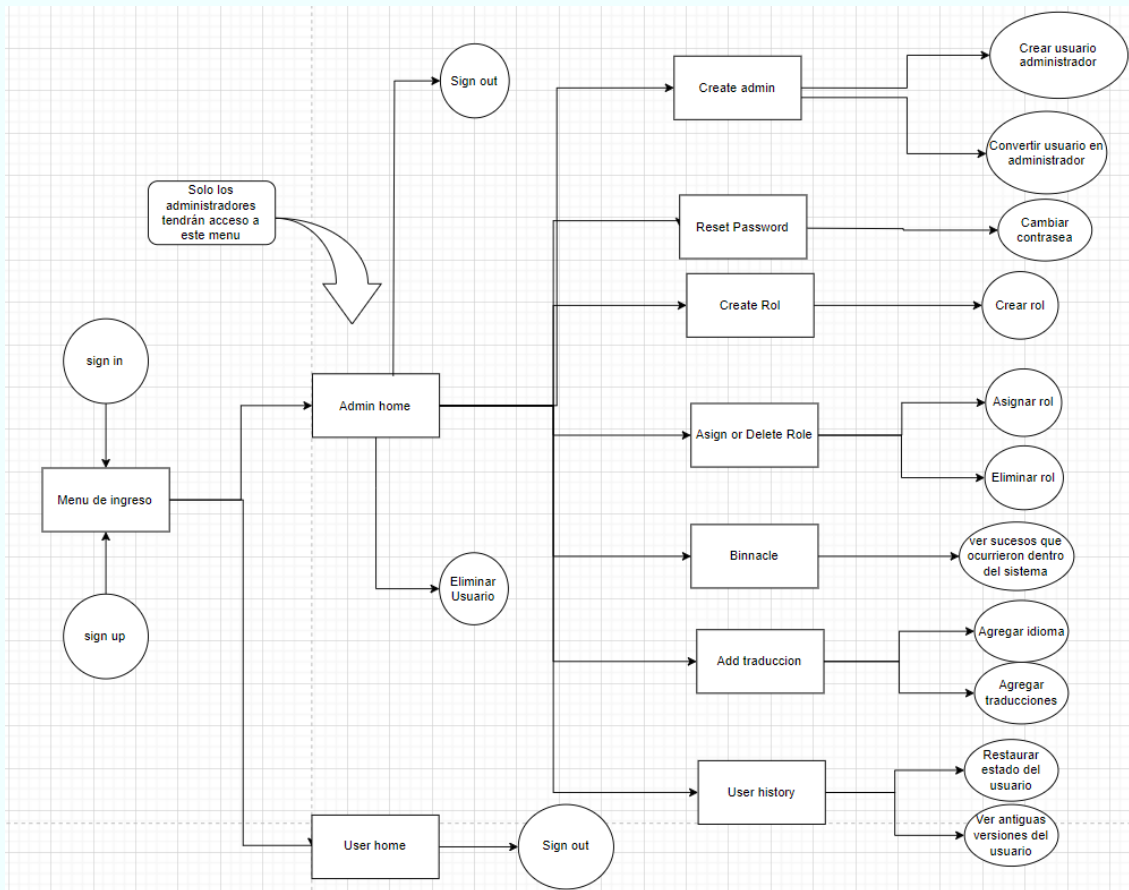


Diagrama de componentes dentro de Admin home



Mapa tentativo de navegación de los menús del sistema

## T02 Gestión de Log in/Log out del Sistema

### ¿Qué es?

El log in es un proceso que se utiliza en el sistema para autenticar al usuario y permitirles acceder tanto a los recursos como a los servicios que se encuentran protegidos por contraseña. Es una medida de seguridad que protege la privacidad de los datos y evita el acceso no autorizado al sistema.

El log out por otra parte es un proceso que permite a un usuario finalizar su sesión actual del sistema. Cuando un usuario inicia sesión en el sistema, se le asigna un conjunto de credenciales que le permiten acceder a ciertas funciones y recursos. Al cerrar sesión o hacer Log out, el usuario finaliza su sesión activa y se desconecta del sistema.

### ¿Cómo funciona?

Cuando el usuario quiera acceder al sistema se le pedirá que ingrese su credencial de usuario y contraseña a través de un formulario. Luego el sistema autentica la identidad del usuario comparando sus credenciales con las que están almacenadas en la base de datos. Además, que en el proceso de autenticación ocurrirá un proceso de encriptación de la contraseña para aumentar la seguridad del sistema. Esto ocurre gracias a los servicios del sistema, que es una de las capas de la arquitectura o subsistema del sistema.

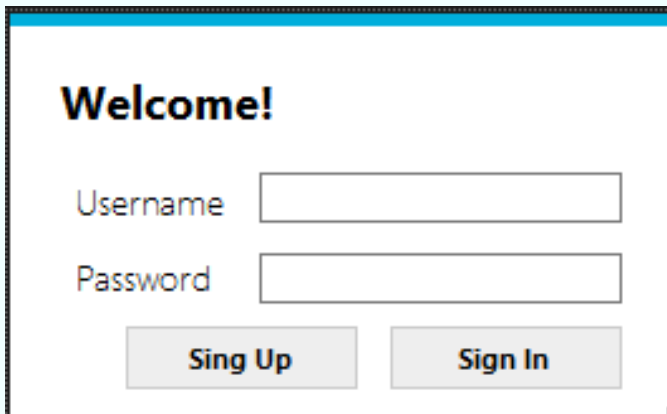
The image shows a login form with a light pink background and a blue border. At the top, it says "Welcome!" in bold black text. Below that, there are two input fields: "Username" and "Password". The "Username" field is a simple text box, while the "Password" field is a text box with a small eye icon on the right side. Below the input fields, there are two buttons: "Sing Up" and "Sign In". The "Sing Up" button is on the left and the "Sign In" button is on the right. Both buttons have a light gray background and black text.

Imagen del formulario de log in

#### Explicación del código:

Para realizar la gestión de log in y log out , utilizamos el patrón de diseño singleton, tiene como objetivo asegurar de que solo haya una instancia de una clase en todo el programa.

Este patrón se encuentra dentro de la capa de servicios. Esta clase tiene dos métodos estáticos, "log in" y "log out".

El método "log in" se utiliza para iniciar la sesión. Si "session" es nulo, se creará una nueva instancia de la clase y se almacenará en "session". Además, se establecerá el usuario y la hora de inicio de sesión. Si "sesión" no es nulo, se lanzará una excepción indicando que la sesión ya se ha iniciado.

El método "log out" se utiliza para cerrar la sesión. Si "session" no es nulo, se establecerá en nulo. Si "session" es nulo, se lanzará una excepción indicando que la sesión no se ha iniciado.

Para validar al usuario creamos una clase dentro del mapper, esta tiene dos métodos.



El primer método se llama "validar" y recibe un objeto de tipo BEUsuario, perteneciente a la capa BE. Este método se encarga de verificar si el usuario y contraseña que proporciona el objeto coinciden con algún usuario almacenado en una tabla de la base de datos. Para lograr eso, el método utiliza una instancia de la clase "Acceso" que representa una conexión a la base de datos. Este "Acceso" se encuentra dentro de la capa DAL.

El método crea una instancia de la clase DataTable para poder almacenar los resultados de la consulta que se le hizo a la base de datos. Después utiliza la instancia de "Acceso" para leer los datos de la tabla de usuarios llamando al método "leer" con el nombre de la consulta "s\_Usuario\_listar". Lo que devuelve esta consulta se guarda en el DataTable que se creó anteriormente.

El método utiliza un Foreach para recorrer todas las filas del DataTable. Se compara los valores del usuario y contraseña con los valores de la fila en la tabla. En caso de que haya una coincidencia el método retorna un True, en caso contrario retorna un False.

Dentro del UI, creamos un formulario con los controles necesarios para que el usuario sea capaz de ingresar el usuario y la contraseña.

En el código del formulario se crea una instancia de la clase "BLLUsuario" y una instancia de la clase "BEUsuario" para interactuar con los datos del usuario y llegar hasta el MPP, debido a que la BLL está conectada con el mismo.

El evento "metroButton1\_Click" se desencadena cuando se presiona el botón iniciar sesión dentro del formulario. Aquí es donde se verifican si se han ingresado los datos válidos en los campos de usuario y contraseña.

El evento comprueba si los campos de usuario y contraseña están vacíos, si esto ocurre se lanza una excepción y se muestra un mensaje de error.

En el caso de que los campos no estén vacíos, el evento crea un objeto de la clase BE usuario.

Aquí es donde el objeto toma los valores de los Textbox y a su vez son encriptados gracias al método "Encriptar" de la clase de Servicios.

Posteriormente se llama al método validar de la instancia "oLog" de la clase BLLUsuario para verificar si las credenciales son válidas. En caso de que las credenciales no sean válidas, se muestra un mensaje de error.

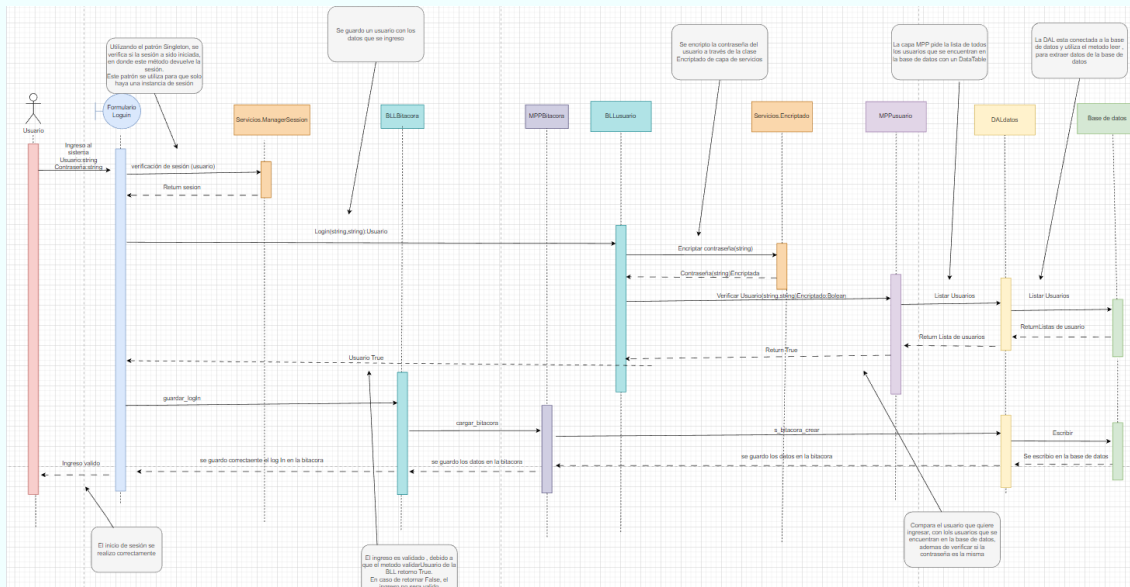


Diagrama de secuencia del Log in

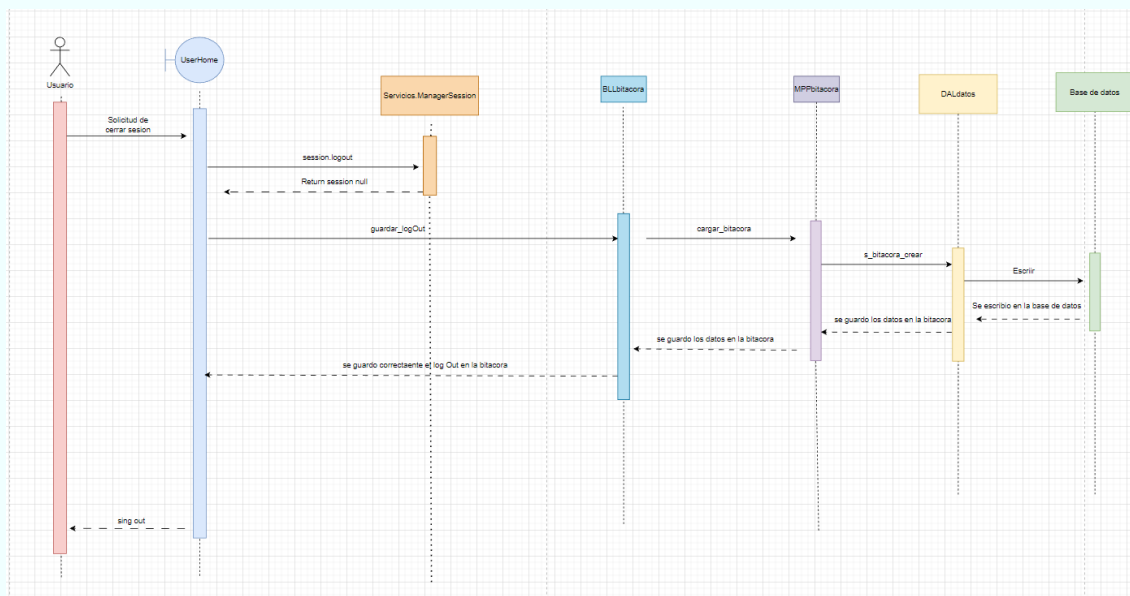


Diagrama de secuencia de Log out

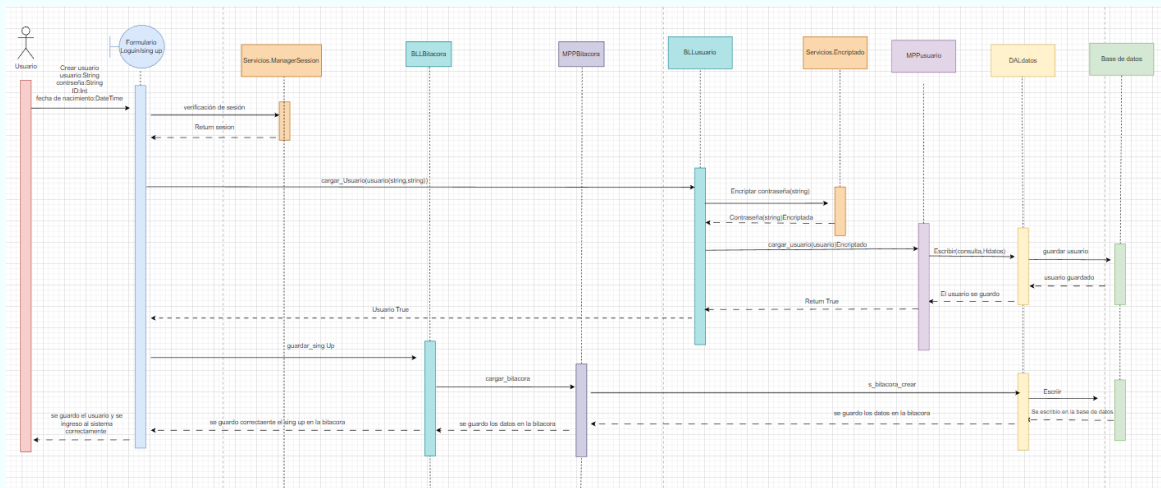


Diagrama de secuencia de Sign Up

## T03 Gestión de encriptado

### ¿Qué es?

El encriptado es un proceso de seguridad que utilizamos para proteger la información confidencial. El objetivo de este encriptado es transformar el texto plano en un formato ilegible y protegido.

La utilización de este genera ventajas al sistema como son la seguridad, la privacidad y confidencialidad.

Ya que su utilización garantiza la protección de información, además de que la vuelve confidencial y privada.

### ¿Cómo funciona?

Este tipo de encriptado es TripleDES este utiliza una clave secreta compartida y un HASH de mensaje.

TripleDES es un algoritmo de cifrado simétrico que utiliza una clave secreta y compartida, para descifrar y cifrar los datos. Esta clave compartida tiene una cierta longitud de bits y se utiliza para cifrar los datos utilizando tres rondas de cifrado DES de forma consecutiva.

### Explicación de código:

Nuestra encriptación tiene dos funciones, ambas retornan un string , nuestra primera función va a ser "static string encriptar" que recibe como parámetro un string que es lo que queremos encriptar.

Este código se encuentra en la capa de Servicios. Hay dos métodos de encriptación de datos.

El primer método se llama "GenerarSHA", este se utiliza para generar un hash SHA-1 a partir de una cadena de texto dada. Este utiliza una instancia de la clase "SHA1CryptoServiceProvider" para calcular un hash SHA-1 de la cadena de texto dada. En donde el resultado retorna como una cadena en un formato base de 64 bytes.

El segundo método se llama "encriptar" y utiliza una combinación de MD5 y tripleDES para encriptar una cadena de texto dada. Este utiliza una clave llamada "qualityinfosolutions" que es transformada en una matriz de bytes utilizando la clase "MD5CryptoServiceProvider".

Posteriormente utiliza la clase "TripleDESCryptoServiceProvider" para crear un objeto de cifrado y encriptar la cadena de texto dada.

La cadena de texto encriptada se devuelve como un objeto Hashtable que contiene los valores encriptados de usuario y contraseña, que son agregados a través del método "Add" de la clase Hashtable.

## T04 Gestión de perfiles de usuario

¿Qué es?

La gestión de roles es la capacidad del sistema para asignar y administrar diferentes roles y patentes. De esta manera se administran los permisos que los usuarios poseen a las diferentes funcionalidades del sistema.

¿Qué ventajas trae aplicarlo?

La gestión de roles por medio del composite permite que los usuarios puedan tratar a un rol o patente de la misma manera, es decir, es transparente para ellos. De este modo pueden crear, eliminar, asignar o desasignar roles o patentes si tienen el permiso necesario para ello.

¿cómo lo realizamos?

Para gestionar roles utilizamos el patrón composite, en el cual existen roles (familias) y patentes (permisos). Esto nos permite hacer esa diferencia transparente al usuario, y agrupar roles y permisos de una manera flexible y escalable.

Para agregar un nuevo rol un usuario deberá seleccionar un rol padre y tantos roles o permisos hijos como desee, siempre y cuando no genere un loop infinito, ni resulte repetido.

Se introdujo también la funcionalidad de asignar o desasignar roles a un usuario. Esto solo puede llevarse a cabo por otro usuario con permisos de administrador. Para esto se utiliza una interfaz intuitiva que permite seleccionar roles y asignarlos o quitarlos.

Por último, es posible eliminar roles (no así permisos) mediante una interfaz sencilla, se prevé siempre que esos roles no estén en uso por ningún usuario primero.

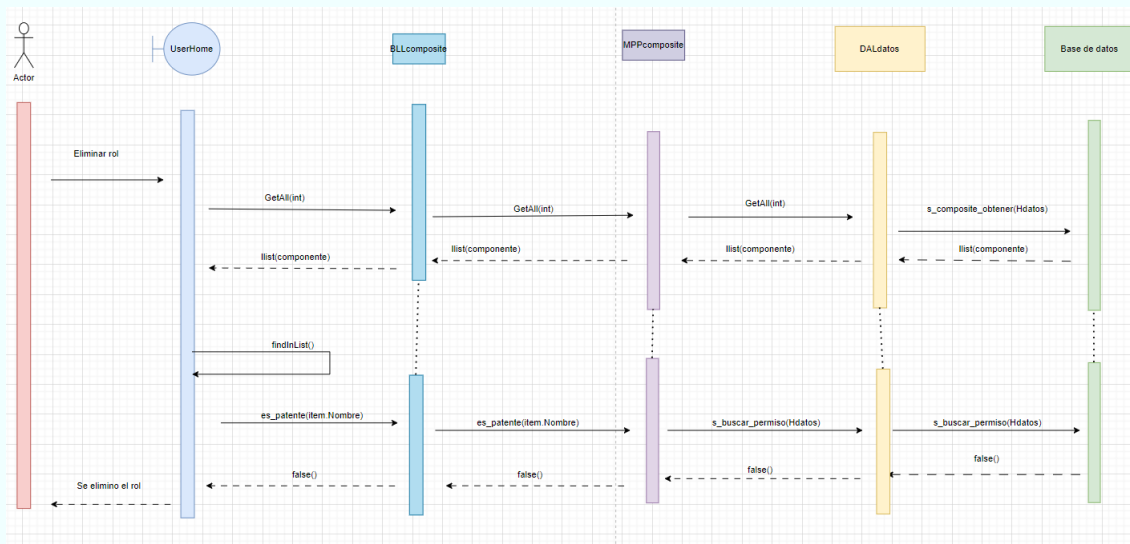


Diagrama de secuencia de eliminación de rol

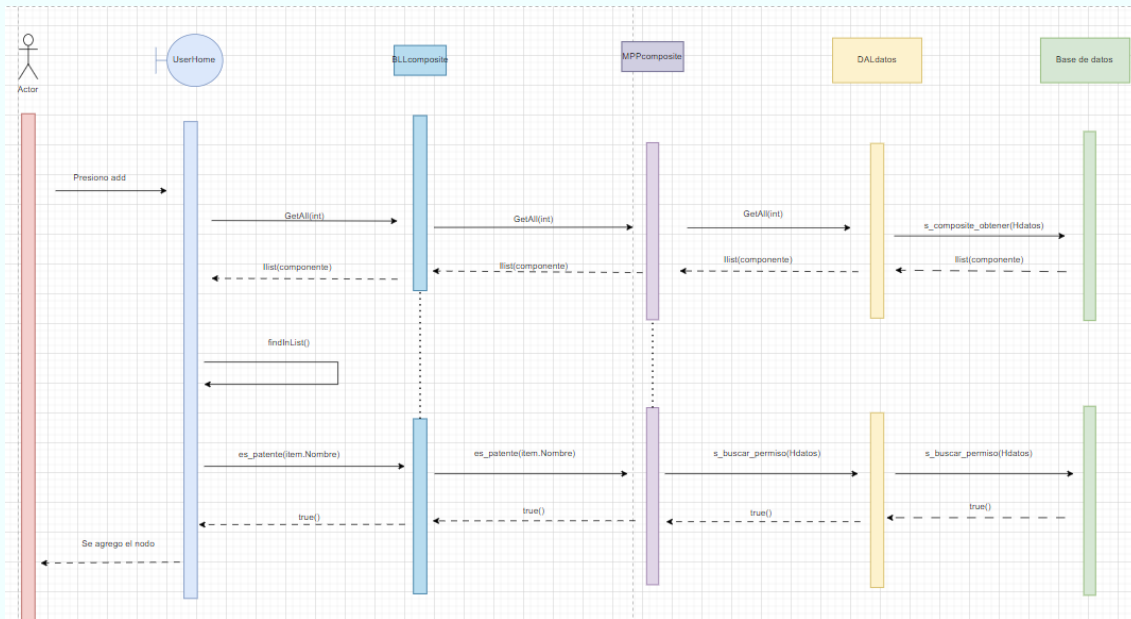


Diagrama de secuencia de creación de rol

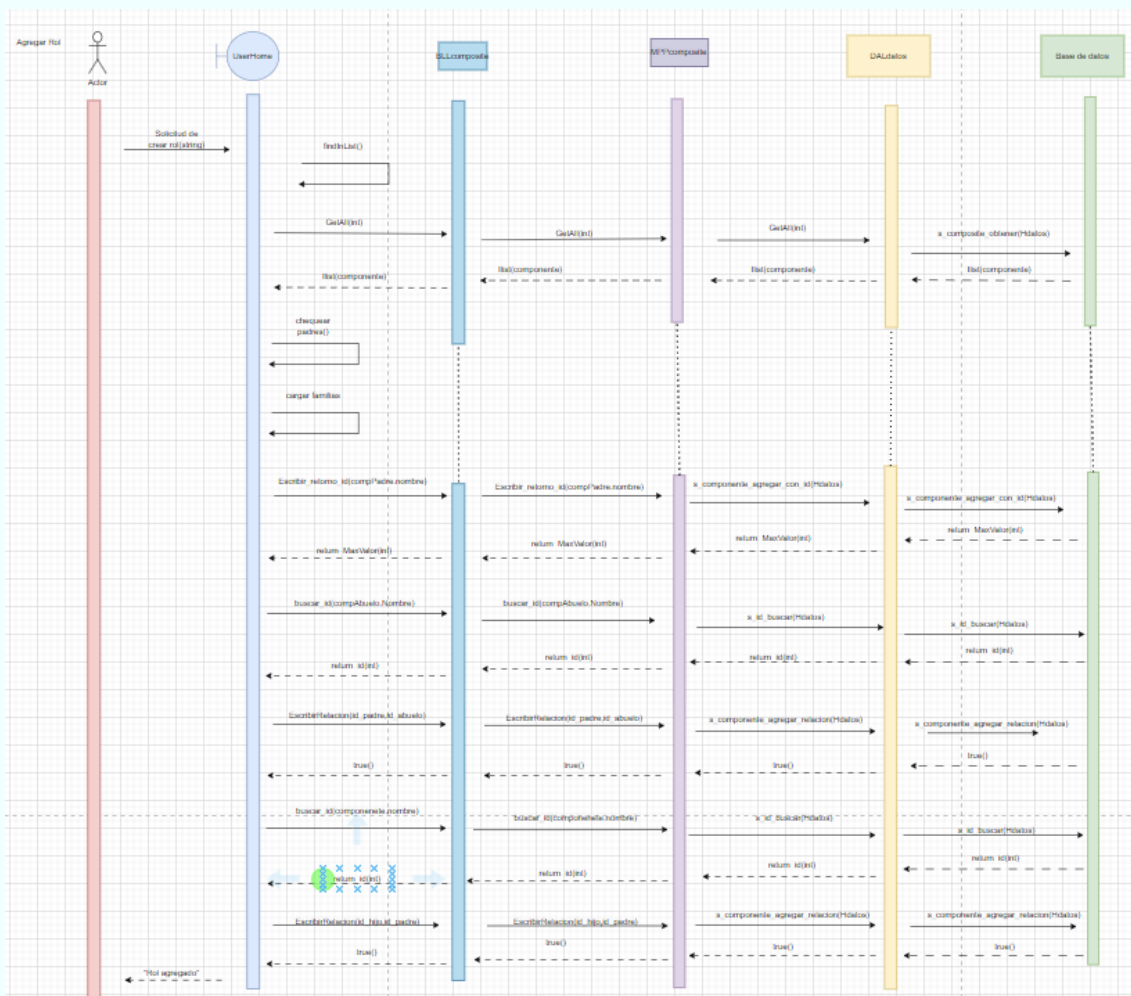
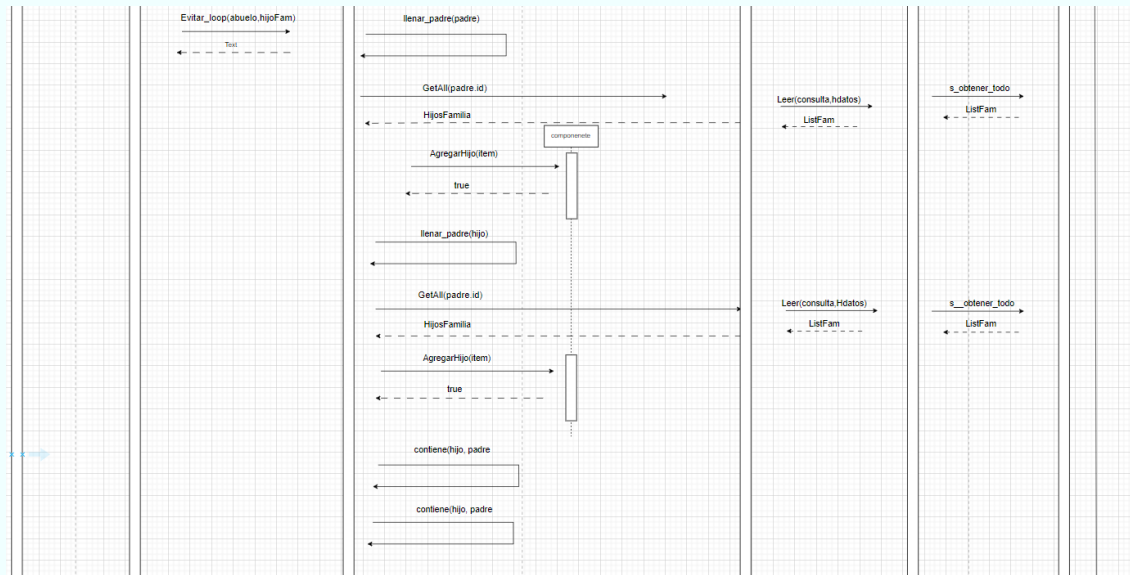
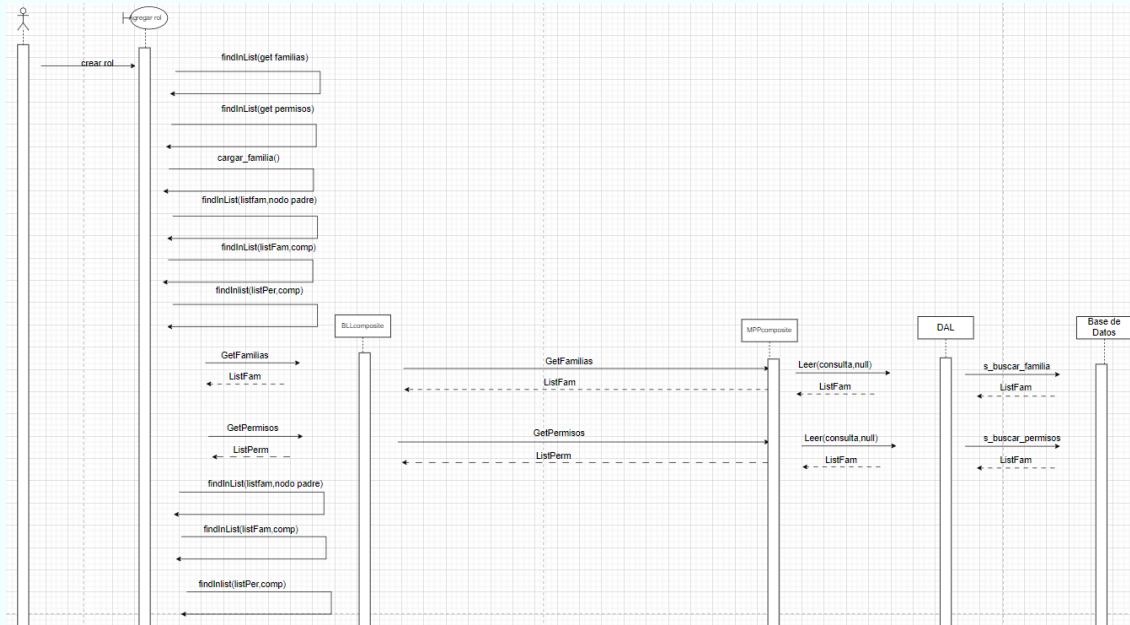


Diagrama de secuencia de agregar rol



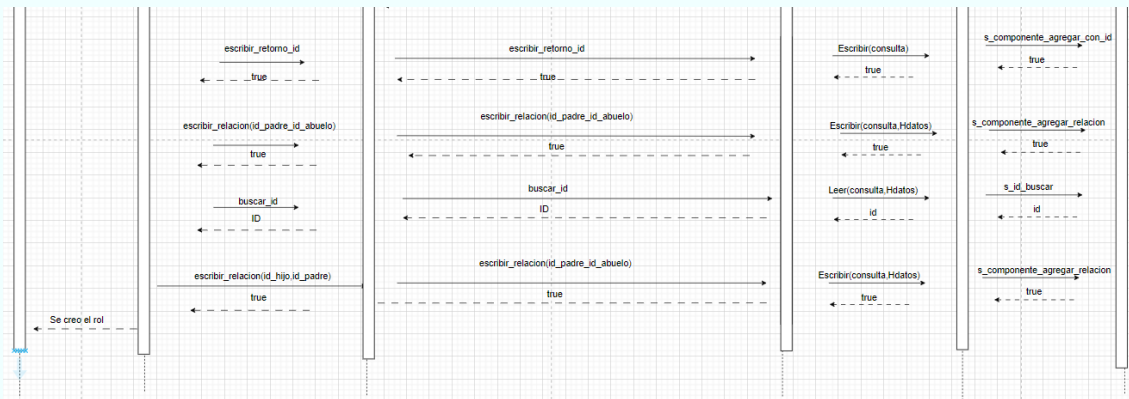
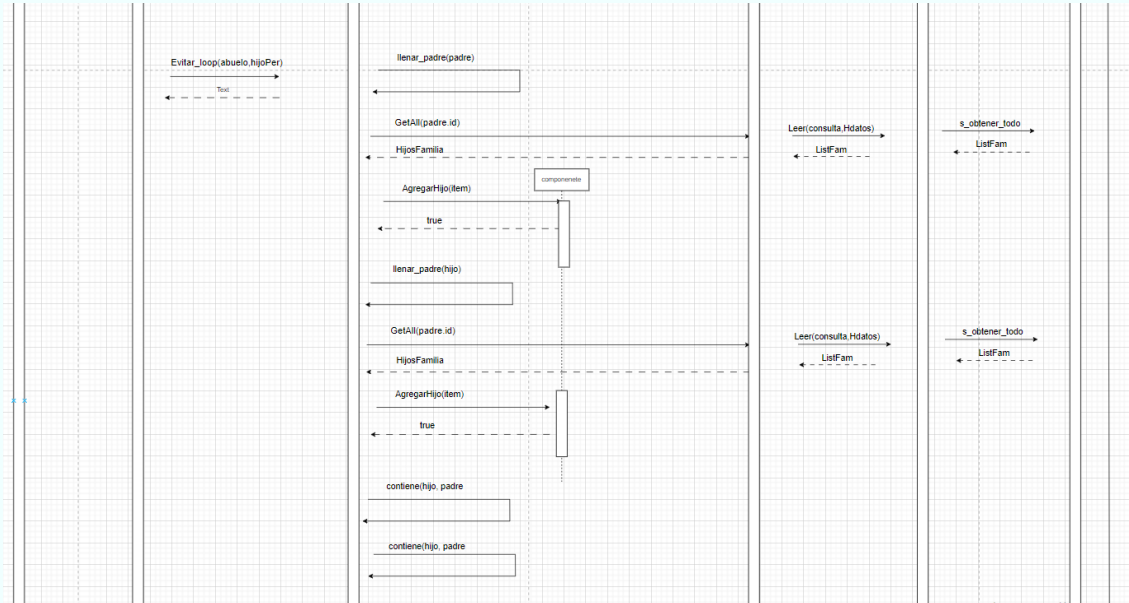


Diagrama de secuencia de crear rol

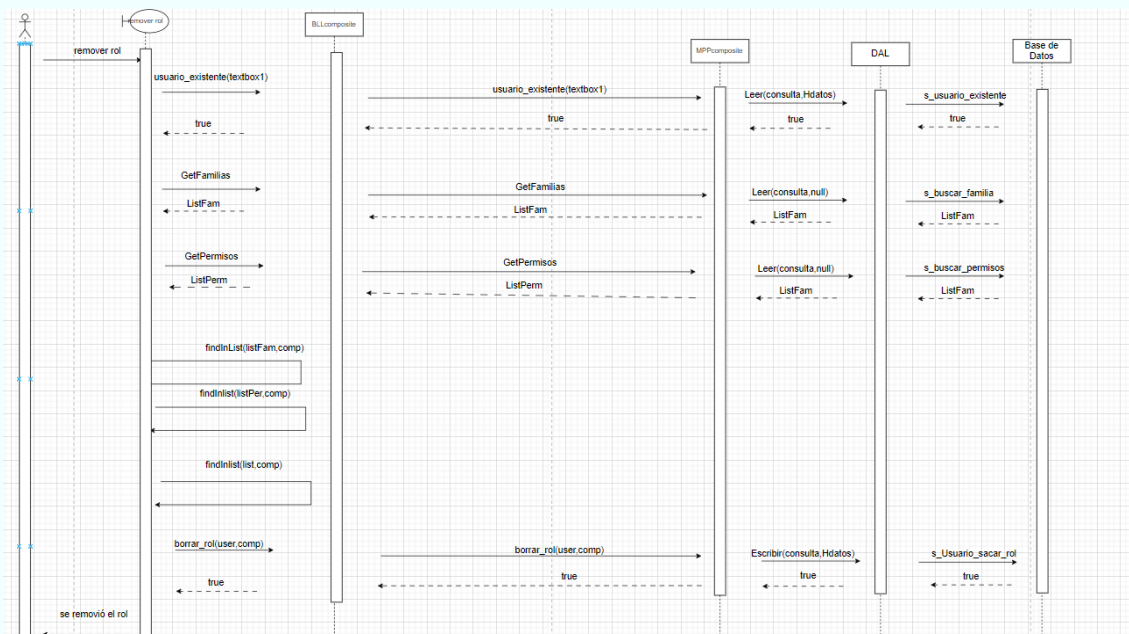


Diagrama de secuencia de remover rol a un usuario



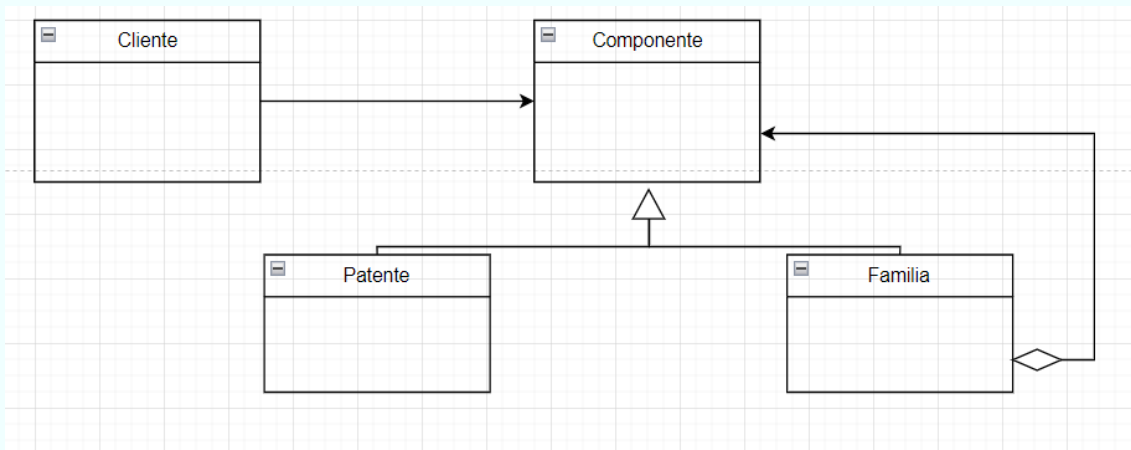


Diagrama de clases de la gestión de roles

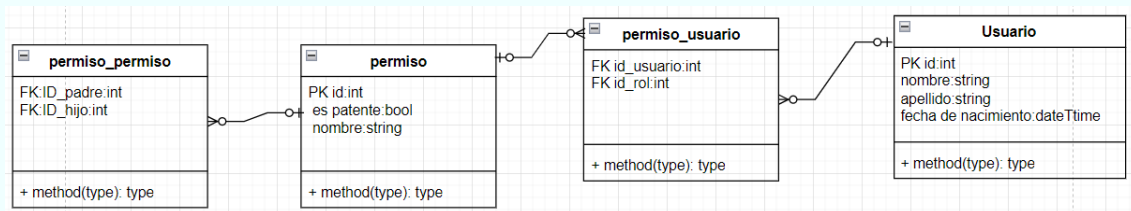


Diagrama de modelado de datos de la gestión de roles

### Diagramas de persistencias:

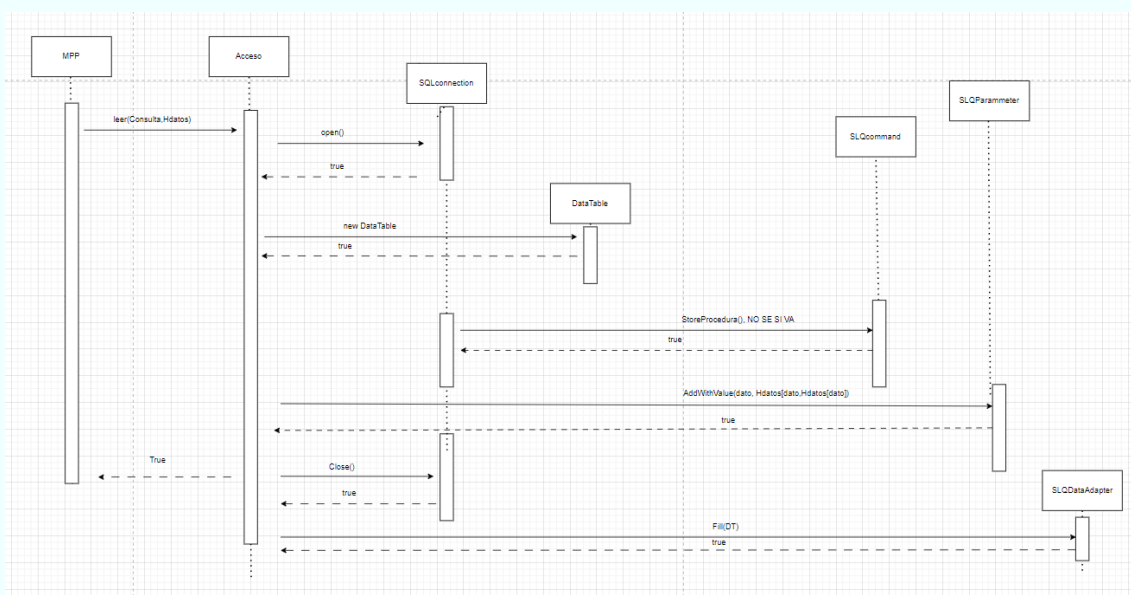


Diagrama de secuencia de lectura de datos

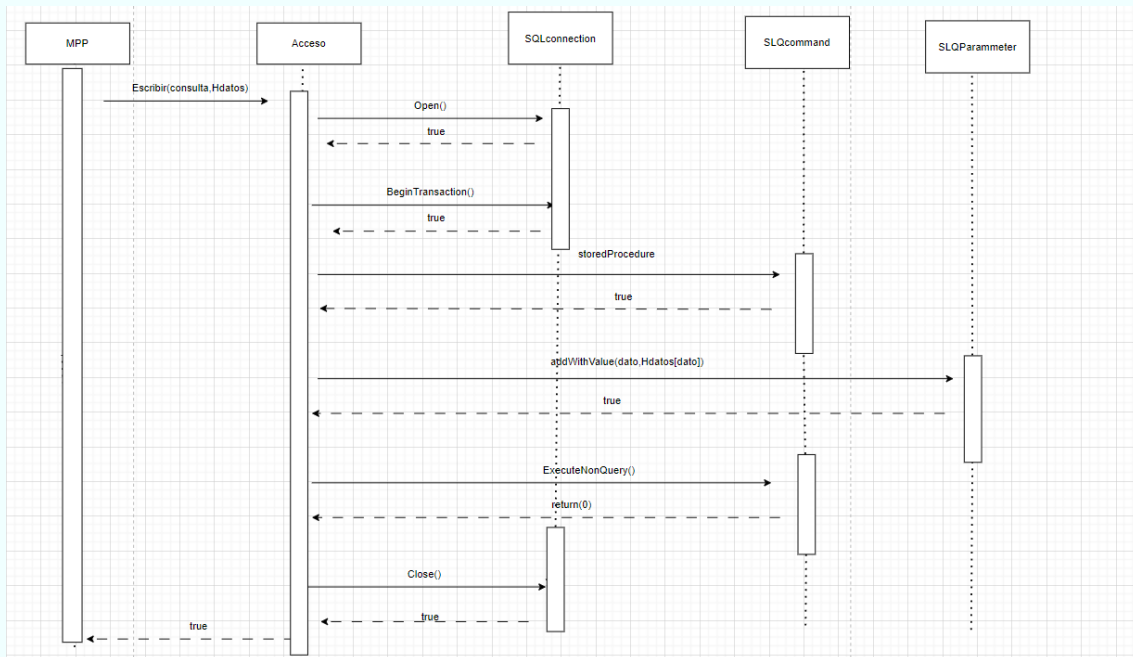


Diagrama de secuencia de escritura de datos

## Create Rol

select the permission you want to add to the role o seleccione desde aqui

[-] Main

[+] admin

[+] gestor

[+] traductor

permissions

roles

select as father node

add

x

confirm

Formulario para crear roles

19

The screenshot shows a web interface titled "Assing role" (note the typo). At the top, there is a "user id" field containing "42933252" and a "search" button. Below this, the instruction "select the role" is displayed. The interface is divided into two main panels: "all roles" on the left and "user roles" on the right. Both panels show a tree structure with a "Main" category containing sub-items "admin", "gestor", and "traductor". The "user roles" panel currently only shows "admin" and "traductor". Below these panels, there is a section "or select from here" with two dropdown menus labeled "permissions" and "roles", and an "add" button. A "remove role" button is located between the two role panels. At the bottom right, there is a small dropdown menu with a downward arrow.

Formulario para crear y eliminar roles

## T05 Gestión de múltiples idiomas

¿Qué es?

La gestión de múltiples idiomas es la capacidad de un sistema para adaptarse y ofrecer contenido en diferentes idiomas, de acuerdo con las preferencias del usuario.

Explicación:

Esto lo podemos desarrollar a través del patrón observer.

Este es un patrón de diseño que se utiliza para establecer una relación de dependencia de uno a muchos entre objetos, de modo que cuando un objeto cambia su estado, todos los objetos dependientes son notificados y actualizados automáticamente.

Este patrón está compuesto por 3 métodos que son agregar observador, eliminar observador y notificar a los observadores.

En este caso los observadores son los formularios, dentro de la clase observer se instanciara una lista de observadores, cuando se abre un formulario , este será agregado a la lista, en caso de que el formulario se cierre, este se removerá de la lista.

Cuando se quiera cambiar el idioma, este se le notificará a todos los observadores que estén en la lista.

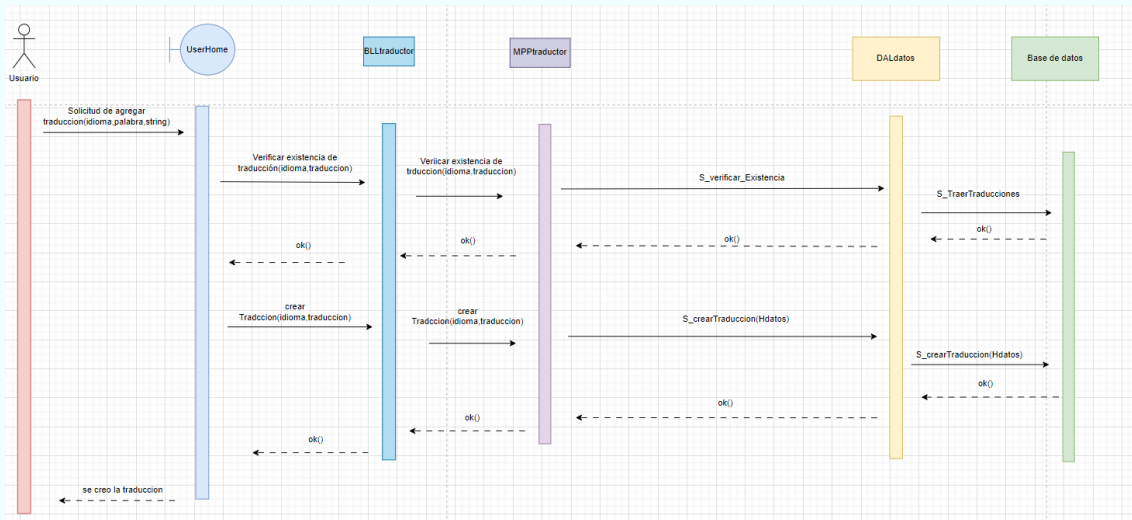


Diagrama de secuencia de agregar traducción

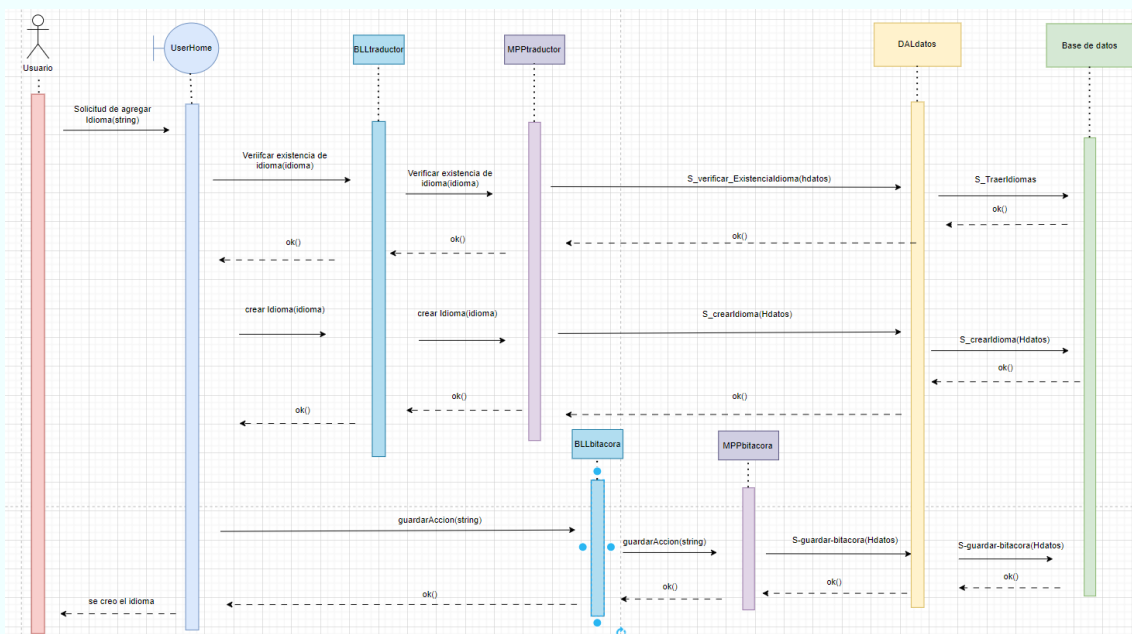


Diagrama de secuencia de agregar idioma

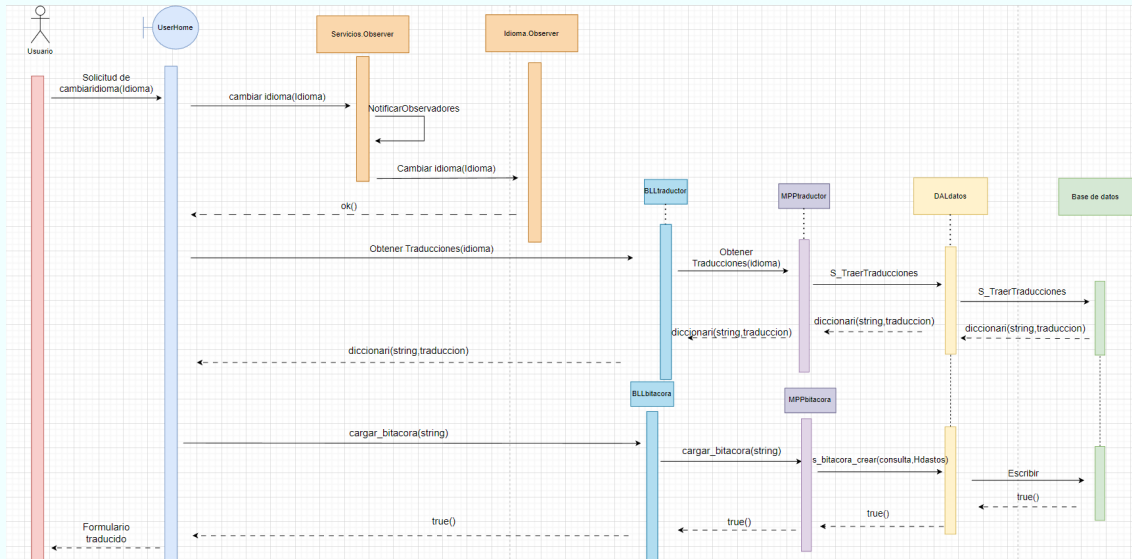


Diagrama de secuencia cambio de idioma

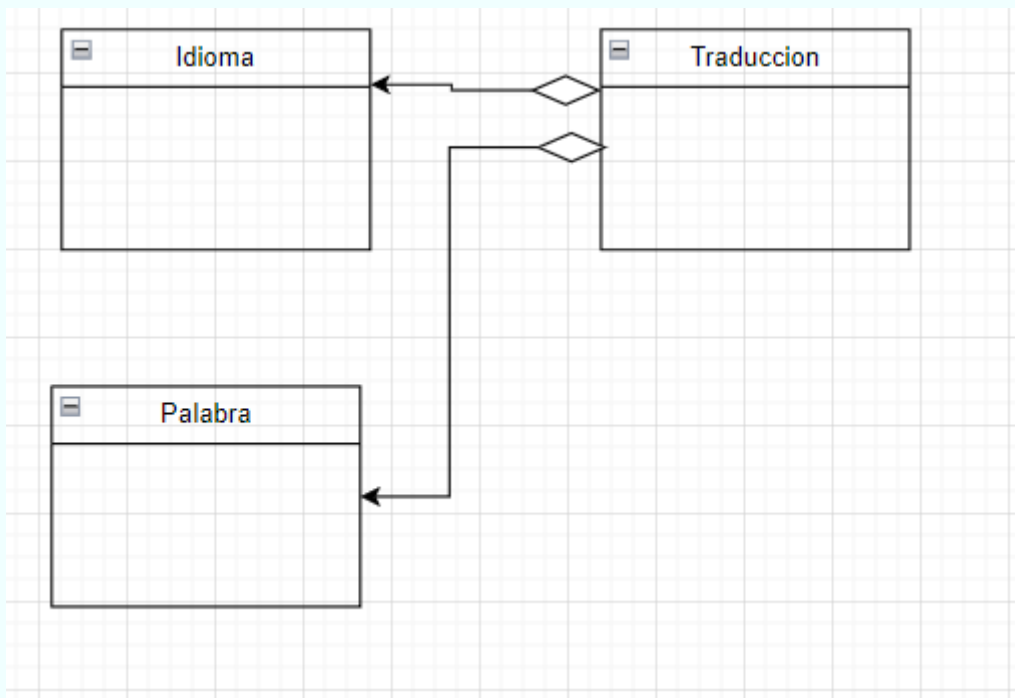
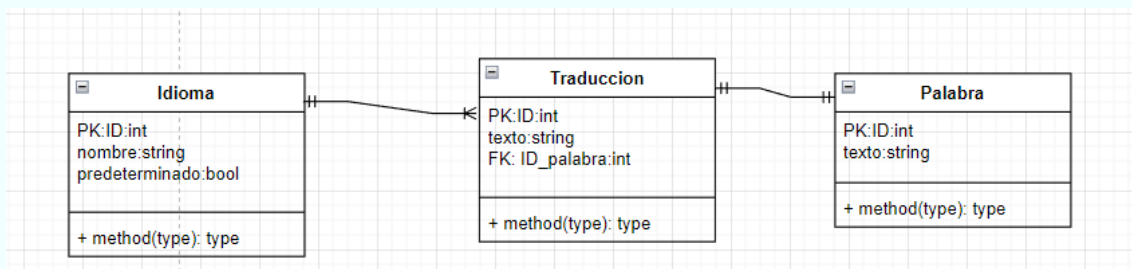


Diagrama de clases



## diagrama de modelo de datos

Traduccion	Palabra
casa do administrador	admin home
id	id
Bemvindo	welcome!
senha	password
linguagem	lenguaje
*	

*Formulario para crear un lenguaje*

## T06 Gestión de bitácora y control de cambios

### ¿Qué es la bitácora?

La bitácora es el registro detallado y cronológico de los eventos, actividades y cambios que ocurren dentro del sistema. Esta es fundamental para el seguimiento y la resolución de los problemas, ya que nos puede proporcionar información histórica para encontrar la secuencia de eventos que llevaron a una situación en particular.

La bitácora incluye información como la fecha y hora de una actividad, el usuario que la realizó y cualquier tipo de resultado o error asociado a esa actividad. Volviéndose una herramienta esencial para el monitoreo y el mantenimiento del sistema, volviéndose de gran utilidad.

### Explicación de código de la bitácora:

Para poder realizar una gestión de bitácora creamos dos clases, una se encuentra dentro de la capa BLL llamada "BLLbitácora" y otra dentro de la capa MPP llamada "MPPbitacora".

Dentro de la clase BLLbitacora se encuentran tres métodos, estos son: "guardar\_accion", "guardar\_logIn" y "guardar\_loggOut".

El primer método "guardar\_accion" recibe un parámetro "acción" que es del tipo string, esta almacena información de la acción realizada por el usuario. Este método crea una instancia del objeto "MPPbitacora" de la capa MPP, carga el usuario actual y la fecha actual gracias a la sessionmanager, para después llamar al método "cargar\_bitacora" de la instancia creada anteriormente para almacenar la información en la base de datos.

Los otros dos métodos "guardar\_logIn" y "guardar\_LogOut" funcionan de la misma manera que el método explicado anteriormente solo que estos no reciben un parámetro, sino que la acción ya está asignado dentro del método. Por ejemplo en el de "guardar\_logOut" la acción ya almacenada dentro del método será "logged out".

Una vez que estos parámetros lleguen a MPPbitacora a través de alguno de estos 3 métodos, el método "cargar\_bitacora" asigna los valores del parámetro a las claves de Hastable. Después utiliza la clase "Acceso" para llamar al método "escribir" que ejecutará la consulta que está almacenada que es "S\_bitacora\_crear" con los parámetros correspondientes. Si todo se realiza de manera exitosa el método retorna un true sabiendo que los datos se han almacenado correctamente dentro de la base de datos, de caso contrario retorna un false.

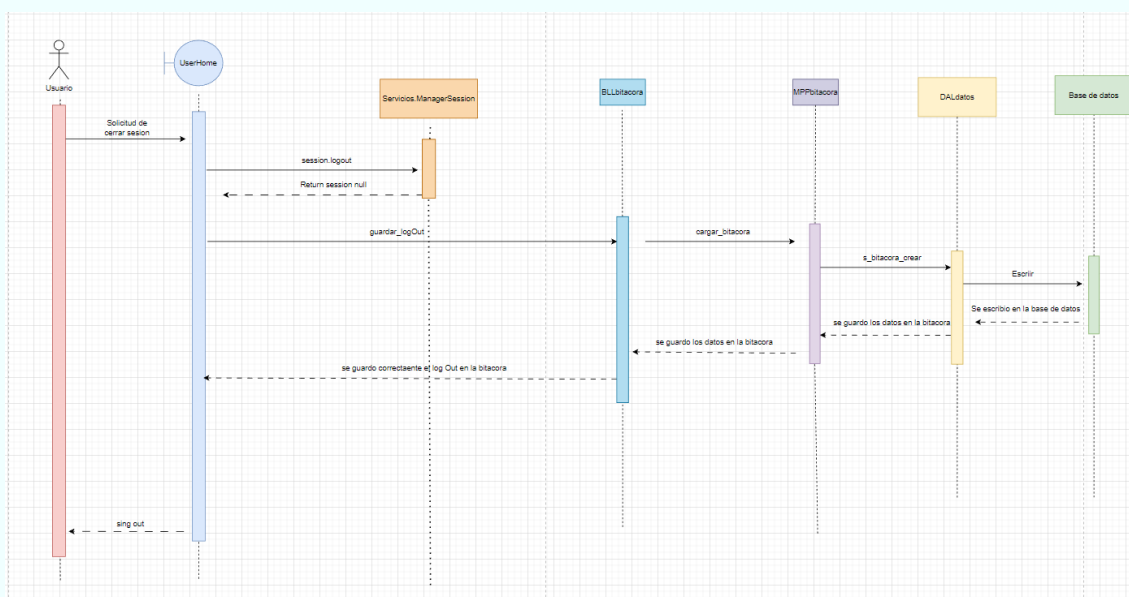




Diagrama de secuencia de Log out que también incluye el funcionamiento de la Bitácora

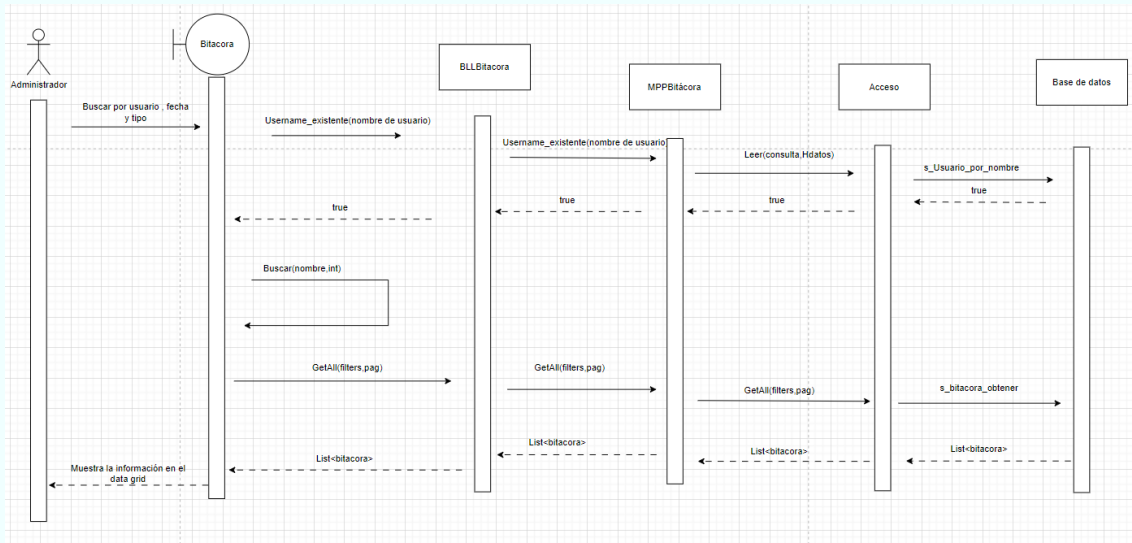


diagrama de secuencia de como funciona la búsqueda dentro de la bitácora

¿Qué es la gestión de cambios?

Es el seguimiento de las modificaciones realizadas en un objeto, en este caso, esta gestión de cambios se los hicimos a los usuarios. Esto implica registrar y controlar los cambios efectuados a sus datos, así como administrar las versiones del objeto a lo largo del tiempo.

¿Cómo funciona la gestión de cambios?

Dentro de la base de datos se guardan las versiones que va teniendo los datos del usuario a lo largo del tiempo, esto nos permite poder hacer una observación a las distintas versiones del mismo.

Dentro del programa hay un formulario que a través de un textbox nos permite ingresar el nombre de un usuario y poder observar sus cambios.

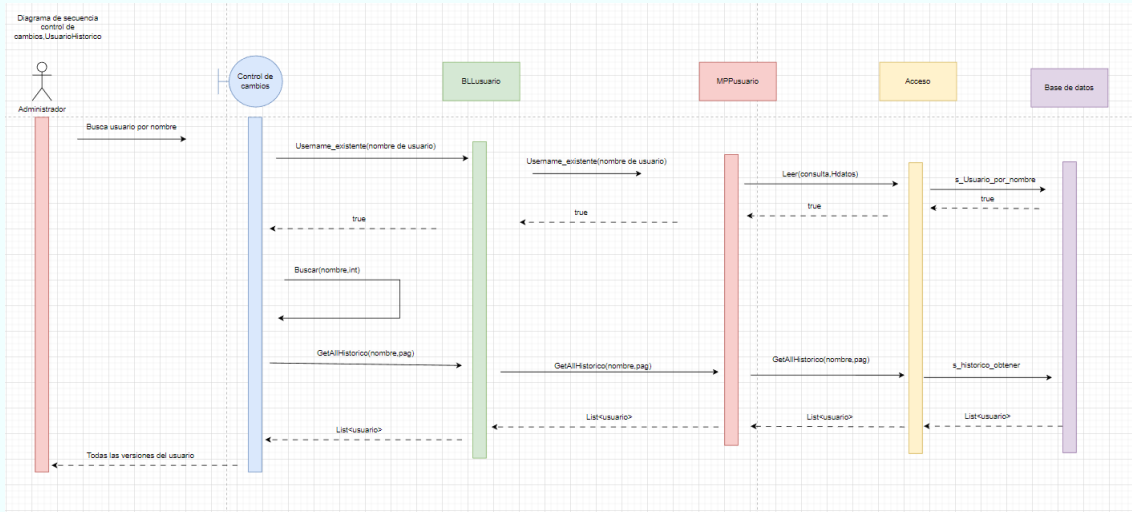


diagrama de secuencia de como se busca dentro del sistema todas las versiones históricas del usuario

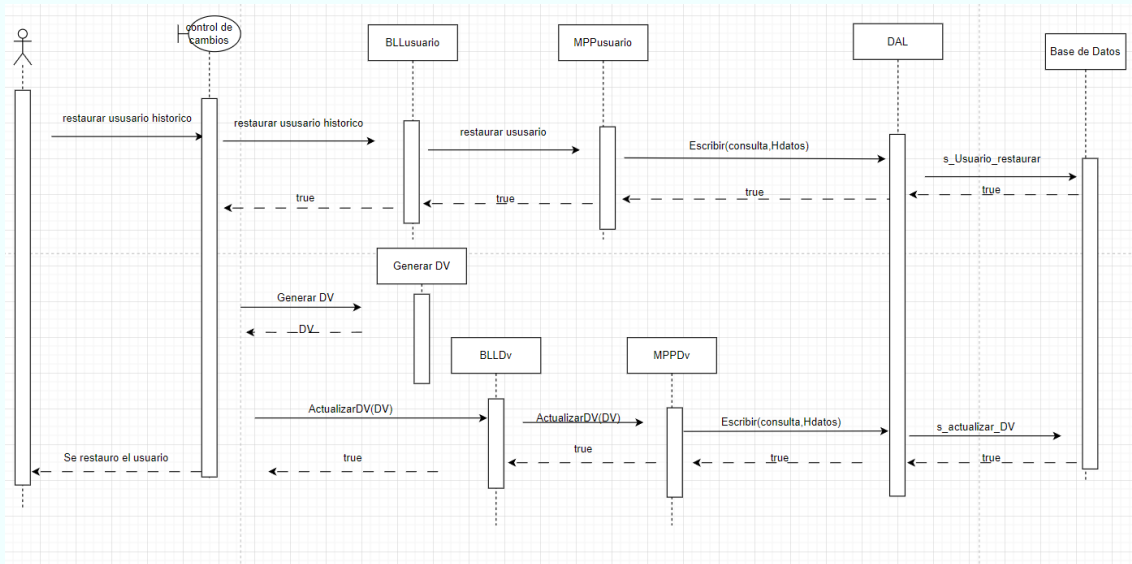


Diagrama de secuecnia de la restaurio del usuario

Binnacle

username  x type  x date from viernes, 14 de julio d▼ to viernes, 14 de julio d▼ apply delete

	IdBitacora	Username	Date	Message	Type
▶	1657	gaston	14/7/2023 09:46	logged in	2
	1658	gaston	14/7/2023 09:46	logged out	2
	1659	gaston	14/7/2023 09:46	logged in	2
	1660	gaston	14/7/2023 09:46	logged out	2
	1661	renavidal	14/7/2023 09:46	logged in	2
	1662	renavidal	14/7/2023 09:46	logged out	2
	1663	renavidal	14/7/2023 09:47	logged in	2
	1664	renavidal	14/7/2023 09:48	logged in	2
	1665	renavidal	14/7/2023 09:48	dio privilegios de ...	2
	1666	renavidal	14/7/2023 09:50	dio al usuario 449...	2

< >

Formulario para ver la bitácora del sistema

## T07 Gestión de dígito verificador

¿Qué es?

El dígito verificador es un dígito que tiene todo los usuarios, este se realiza a través de un HASH , en donde se encripta todos los datos del usuario formando un string, este es almacenado dentro de la base de datos, a su vez el sistema en general tendrá un dígito verificador , este se realiza también a través de un HASH , en donde se encripta todos los dígitos verificadores de los usuarios.

¿Qué ventajas trae aplicarlo?

Este Dígito nos permite mejorar la seguridad del sistema, debido a que cuando un usuario quiera entrar al mismo , se verificará que el Dígito verificador sea correcto, de no ser así, nos daremos cuenta de que hubo una alteración de datos dentro de la base de datos.

Gracias a la gestión de control de cambios , podremos volver los datos del usuario a la última vez que se haya guardado dentro de la gestión de cambios, modificando estos datos que fueron alterados al estado que tenían anteriormente.

¿cómo lo realizamos?

Para poder realizar el dígito verificador de la tabla usuario , primero reuno los datos que voy a utilizar del usuario y a través de un HASH lo

transformó todo a una cadena de string. Esta cadena es almacenada dentro de la base de datos dentro del usuario.

Posteriormente se obtienen todos los dígitos verificadores de los usuarios y los encripto a través de un HASH obteniendo una cadena string, esta es almacenada en la base de datos dentro de la base de datos.

Cada vez que se quiera ingresar al sistema se creará una un nuevo dígito verificador con los datos de los usuarios y se comparará con el dígito verificador de la base de datos que se almacenó anteriormente, es caso de que estos dos sean distintos , nos daremos cuenta que hubo una alteración dentro de la base de datos.

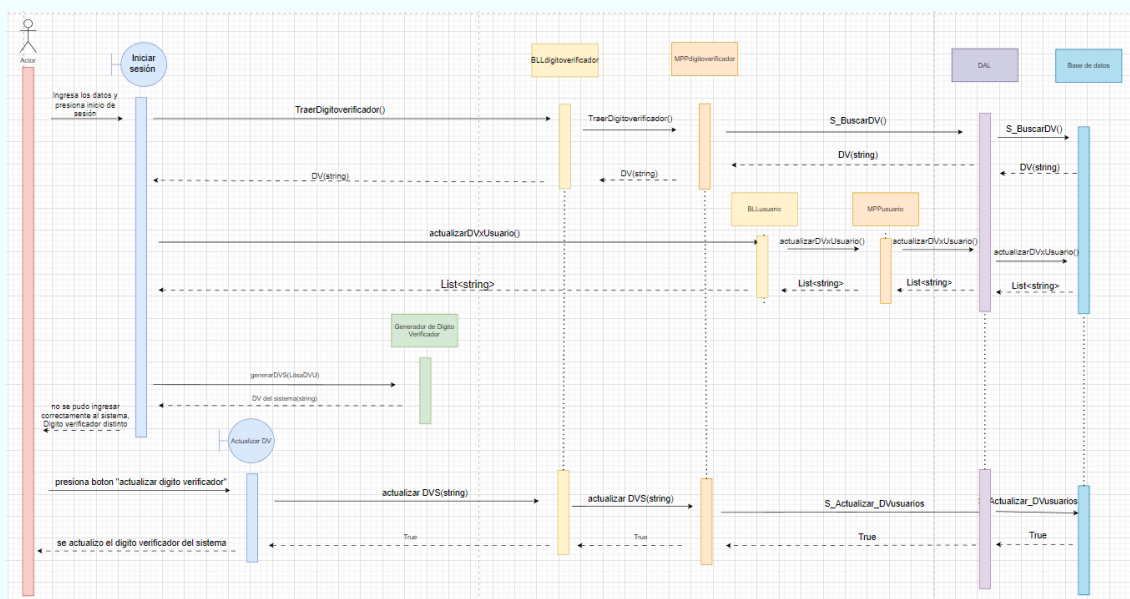


Diagrama de secuencia de error de dígito verificador y la actualización del mismo

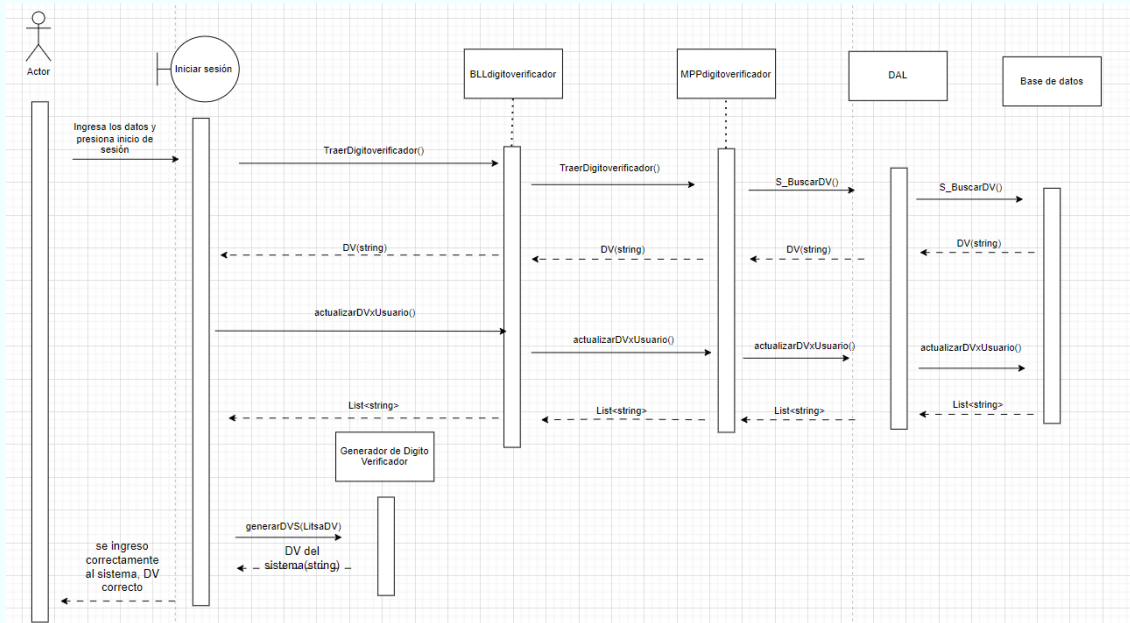


Diagrama de secuencia de verificación del dígito verificador

# Negocio

## Análisis general del sistema

Nuestra solución integral de gestión de alquiler de carpas en balnearios costeros representa una plataforma tecnológica avanzada y altamente eficiente diseñada para satisfacer las necesidades tanto de los usuarios que buscan la mejor experiencia playera como de los dueños de balnearios que desean optimizar la gestión de sus servicios. Con un enfoque profesional y técnico, nuestro programa ofrece una gama completa de funcionalidades para brindar a ambas partes una experiencia fluida y enriquecedora.

### Características Clave:

*Para Usuarios:*

**Filtrado Personalizado:** Nuestra plataforma permite a los usuarios filtrar las opciones de alquiler de carpas según sus preferencias específicas, servicios adicionales, políticas sobre niños, y mascotas.

**Reservas y Cancelaciones:** Los usuarios pueden realizar reservas de carpas en tiempo real y cancelarlas con facilidad, lo que garantiza una experiencia de usuario flexible y sin complicaciones.

**Opiniones y Valoraciones:** Fomentamos la participación de los usuarios al permitirles dejar opiniones y valoraciones sobre las experiencias de alquiler, contribuyendo así a una comunidad de viajeros informada.

*Para Dueños de Balnearios:*

**Interfaz Interactiva de Creación de Balnearios:** Nuestra plataforma ofrece a los dueños de balnearios una interfaz intuitiva y completa para crear y personalizar sus perfiles de balneario. Pueden posicionar carpas, incluir imágenes de sus instalaciones, definir políticas sobre niños, informar sobre servicios adicionales (como juegos, bares, piscinas, etc.) y ajustar los precios de alquiler.

**Gestión de Perfiles de Balneario:** Los propietarios pueden acceder y administrar fácilmente sus perfiles de balneario en cualquier momento, lo que les permite realizar cambios, actualizaciones y ajustes según sea necesario.

**Estadísticas y Seguimiento:** Nuestra plataforma proporciona herramientas de análisis y estadísticas para ayudar a los dueños de balnearios a comprender mejor el rendimiento de sus carpas y tomar decisiones informadas para maximizar sus ingresos.

**Gestión de Alquileres:** La plataforma simplifica la gestión de alquileres, permitiendo a los propietarios ver y administrar todas las reservas de sus carpas, lo que facilita el seguimiento y la organización.

Nuestra solución se basa en tecnología de vanguardia, garantizando la seguridad de las transacciones y la confidencialidad de los datos. Nos esforzamos por proporcionar una experiencia de usuario de primera clase tanto para los usuarios como para los dueños de balnearios, lo que contribuye al crecimiento y éxito de sus negocios en el mercado de alquiler de carpas en balnearios costeros.

### **Motivación**

Solucionar una necesidad actual de agilizar la reserva de carpas en los balnearios que actualmente no poseen un sistema digital.

### **Objetivo general**

Los balnearios argentinos actualmente no poseen un sistema que facilite la experiencia de los clientes, teniendo estos que realizar filas bajo el sol o llegar antes de lo que desean para poder conseguir una carpa y/o sombrilla.

### **Objetivos específicos**

**Eficiencia en la Administración de Balnearios:** Permitir a los administradores configurar la disposición de las carpas y las características del balneario de manera intuitiva, agilizando la gestión y minimizando los errores en la asignación.

**Mejora de la Experiencia del Cliente:** Proporcionar a los clientes información detallada y precisa sobre las opciones de carpas y servicios en los balnearios, permitiéndoles tomar decisiones informadas de acuerdo a sus necesidades y preferencias.

**Generación de Información Accesible:** Proveer informes y análisis que sean comprensibles y útiles para los administradores, ayudándoles a identificar patrones, tendencias y oportunidades de mejora en la operación de los balnearios.

### **Descripción funcional y alcance**

El sistema de gestión de alquiler de carpas es una aplicación de escritorio diseñada para facilitar y optimizar el proceso de alquiler de carpas en balnearios. El sistema está dirigido tanto a administradores de balnearios como a clientes que deseen alquilar carpas en dichos establecimientos.

### **Procesos de negocio involucrados**

**Creación y Gestión de Balnearios:** Este proceso permite a los administradores crear nuevos balnearios, definir sus características y establecer la disposición de las carpas.

**Configuración de Carpas:** Los administradores determinarán la cantidad de carpas, su disposición y las restricciones asociadas, como la admisión de mascotas o niños.

**Búsqueda y Selección de Carpas:** Los clientes buscarán balnearios disponibles y seleccionarán una carpa en función de sus preferencias y necesidades.

**Reserva y QR:** Los clientes reservarán una carpa y recibirán el código QR correspondiente de manera segura a través del sistema.

**Opiniones de usuarios:** Los clientes podrán dejar sus opiniones acerca de las reservas que han hecho en el pasado, ayudando así a futuros usuarios a decidir y a los propios balnearios a lograr mejoras.

**Generación de Informes y Análisis:** Los informes generados por el sistema ayudarán a los administradores a tomar decisiones informadas sobre la operación de los balnearios y la satisfacción del cliente.

## Información

**Histórico de Reservas:** Registro de todas las reservas realizadas, incluyendo fechas, balnearios, y servicios adicionales.

**Ingresos y Facturación:** Información sobre los ingresos generados por cada balneario y en conjunto, ayudando en la gestión financiera.

**Preferencias de los Clientes:** Datos sobre las preferencias más comunes de los clientes, como la presencia de mascotas o niños, y servicios solicitados.

**Ocupación y disponibilidad:** Información sobre la ocupación promedio de las carpas.

**Rendimiento de Servicios Adicionales:** Datos sobre la demanda de servicios como restaurantes, piletas y otros entre los clientes.

## Requisitos

### Requisitos Funcionales:

#### Registro de Usuarios:

Los usuarios pueden registrarse en el sistema proporcionando información personal como nombre, dirección, dni y contraseña.

#### Iniciar Sesión (Log In) y Cerrar Sesión (Log Out):

Los usuarios y dueños de balnearios pueden iniciar sesión en sus cuentas utilizando su nombre de usuario y contraseña.

Deben poder cerrar sesión de manera segura cuando lo deseen.



#### **Gestión de Perfil:**

Los administradores deben poder cambiar contraseñas de usuarios y eliminarlos.

#### **Filtrado de Carpas:**

Los usuarios pueden filtrar las opciones de alquiler de carpas según sus preferencias, incluyendo servicios adicionales y políticas sobre niños y mascotas.

#### **Alquiler de Carpas:**

Los usuarios pueden realizar reservas de carpas en tiempo real, proporcionando fechas de inicio y finalización.

Deben recibir confirmaciones de reserva por código QR.

#### **Cancelación de Reservas:**

Los usuarios pueden cancelar sus reservas antes de la fecha de inicio, siguiendo una política de cancelación definida.

#### **Opiniones y Valoraciones:**

Los usuarios pueden dejar opiniones y valoraciones sobre las experiencias de alquiler de carpas.

#### **Creación de Perfiles de Balneario:**

Los dueños de balnearios pueden crear perfiles de balnearios con información detallada, incluyendo nombre , imágenes, políticas sobre niños y servicios adicionales.

#### **Posicionamiento de Carpas:**

Los dueños de balnearios pueden utilizar una interfaz interactiva para posicionar las carpas en su área de playa asignada.

#### **Información de Servicios Adicionales:**

Los dueños de balnearios pueden informar sobre servicios adicionales disponibles, como juegos, bares, piscinas, etc.

#### **Políticas sobre Niños y mascotas:**

Los dueños de balnearios pueden definir políticas sobre si permiten o no la presencia de niños y/o mascotas en su balneario.

#### **Gestión de Imágenes:**

Los dueños de balnearios pueden agregar una foto de sus instalaciones y carpas.

#### **Visualización de Estadísticas:**

Los dueños de balnearios pueden ver estadísticas relacionadas con las reservas y el rendimiento de su balneario.

#### **Gestión de Alquileres:**

Los dueños de balnearios pueden ver y administrar todas las reservas de sus carpas.

**Eliminación de Perfiles de Balneario:**

Los dueños de balnearios pueden eliminar sus perfiles de balneario cuando lo deseen.

**Cambio de Idioma:**

Los usuarios pueden cambiar el idioma de la interfaz según sus preferencias.

**Requisitos No Funcionales:**

**Seguridad:**

El sistema debe garantizar la seguridad de los datos personales y financieros de los usuarios.

La contraseña de los usuarios debe estar encriptada y almacenada de forma segura.

**Rendimiento:**

El sistema debe ser eficiente y capaz de manejar múltiples reservas y usuarios concurrentes.

**Disponibilidad:**

El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, para que los usuarios puedan realizar reservas en cualquier momento.

**Escalabilidad:**

El sistema debe ser escalable para permitir la adición de nuevos balnearios y usuarios a medida que crece.

**Integridad de los Datos:**

Los datos del sistema deben mantenerse íntegros y no deben corromperse debido a fallos del sistema. Para esto debe usarse un Dígito verificador.

**Interfaz de Usuario Amigable:**

La interfaz de usuario debe ser intuitiva y fácil de usar tanto para usuarios como para dueños de balnearios.

**Bitácora (Logging):**

El sistema debe llevar un registro de actividades importantes, como inicio de sesión, reservas y errores.

**Gestión de Cambios:**

El sistema debe permitir actualizaciones y cambios de manera controlada, con pruebas adecuadas antes de implementar nuevas características.

**Multilingüismo:**

El sistema debe admitir múltiples idiomas y permitir la creación y modificación de idiomas según sea necesario.

**Gestión de Roles:**

Debe haber roles de usuario definidos, como usuarios normales y dueños de balnearios, con permisos específicos para cada tipo de usuario.

### **Estándares Aplicables:**

**Seguridad de Datos:** Cumplir con estándares de seguridad de datos para proteger la información personal de los usuarios.

**Normativas Locales:** Cumplir con las regulaciones locales y normativas gubernamentales relacionadas con el turismo y los negocios en balnearios costeros.

**Estándares de Experiencia del Usuario:** Implementar mejores prácticas de diseño de experiencia del usuario (UX) para garantizar una interfaz de usuario intuitiva y amigable.

### **Requisitos de Sistema:**

**Base de Datos:** Utilizar una base de datos segura y escalable para almacenar la información de usuarios, reservas y perfiles de balnearios.

**Encriptación:** Implementar encriptación de datos para garantizar la seguridad de la información confidencial, como contraseñas y direcciones.

**Disponibilidad:** El sistema debe estar disponible las 24 horas del día, los 7 días de la semana, con tiempos de inactividad mínimos para mantenimiento.

**Escalabilidad:** Diseñar el sistema para ser escalable y capaz de manejar un aumento en el número de usuarios y balnearios registrados.

### **Requisitos de Desempeño:**

**Velocidad de Respuesta:** El sistema debe responder de manera rápida y eficiente a las solicitudes de los usuarios, especialmente durante el proceso de reserva.

**Tiempo de Carga de Página:** Las páginas del sistema deben cargar en un tiempo razonable, incluso en conexiones de Internet más lentas.

**Capacidad de Procesamiento:** El sistema debe ser capaz de procesar múltiples reservas simultáneamente sin demoras significativas.

### **Requisitos de Entorno:**

**Infraestructura de Servidores:** Disponer de servidores confiables y bien administrados para alojar la aplicación y la base de datos.

**Acceso a Internet:** Asegurar una conexión a Internet de alta velocidad y confiable para garantizar la disponibilidad del sistema.

**Soporte Técnico:** Contar con un equipo de soporte técnico disponible para abordar problemas de usuarios y resolver cualquier problema técnico.

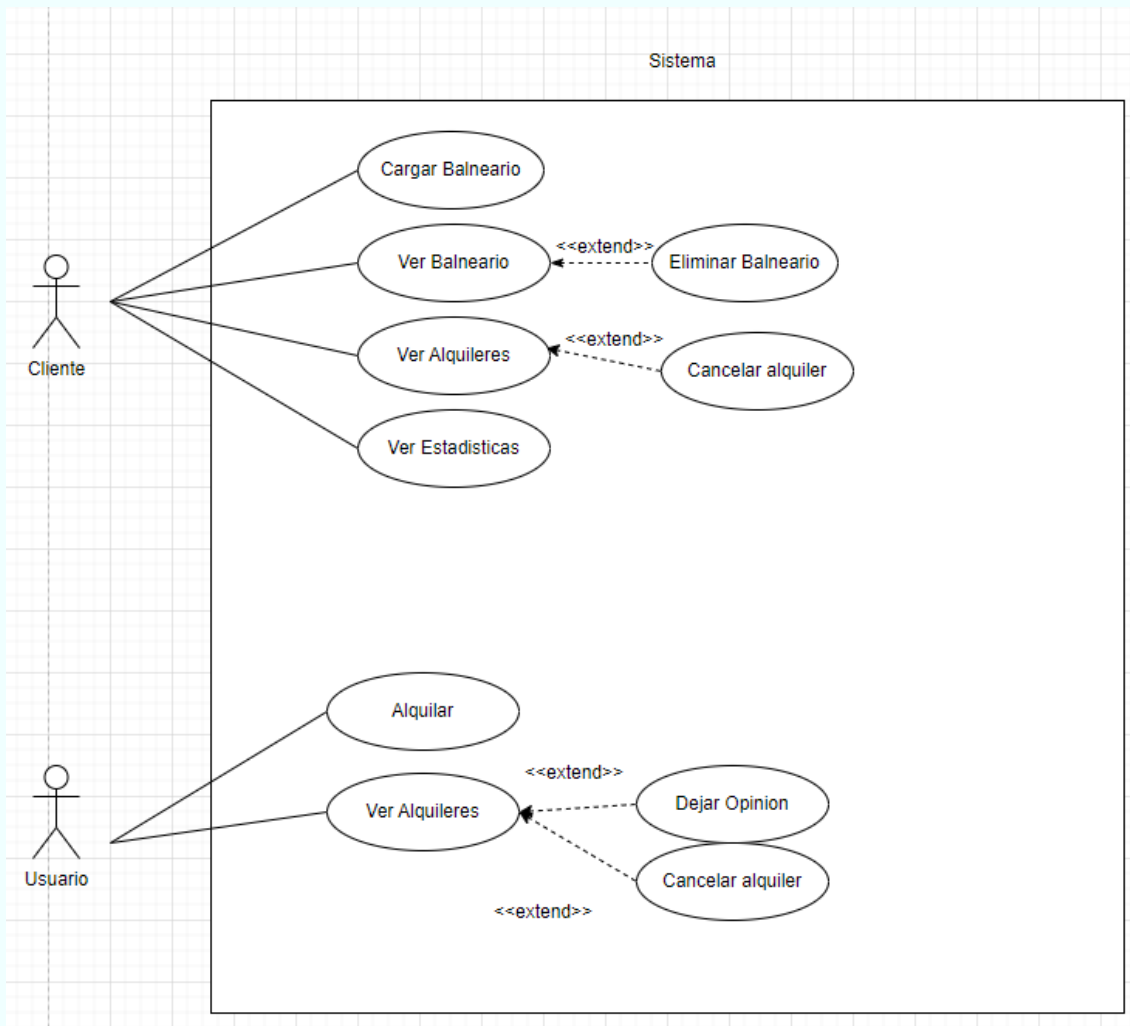
**Actualización y Mantenimiento:** Planificar y realizar actualizaciones regulares del sistema y mantenimiento de la infraestructura para garantizar un rendimiento óptimo.

**Seguridad Física:** Garantizar la seguridad física de los servidores y la infraestructura de TI donde se aloja el sistema.

### Descripción de personas participantes

Nombre	Rol	Funciones
Lucas Ramírez	Ventas y gestión de clientes	<ul style="list-style-type: none"><li>. Visitar clientes</li><li>. Cargar clientes al sistema</li><li>. Manejar contraseñas</li><li>. Realizar cobros</li></ul>
Carolina Silva	Inspección de seguridad y QA	<ul style="list-style-type: none"><li>. Testear el sistema</li><li>. Rigurosas inspecciones de seguridad de encriptado y vulnerabilidades de la base de datos.</li></ul>
Diego González	Legales	<ul style="list-style-type: none"><li>. Redactar contratos con clientes.</li><li>. Regular el cumplimiento de las normas de balnearios.</li><li>. Regular el cumplimiento de las normas de atención al cliente.</li></ul>
Renata Vidal	Desarrolladora y diseñadora	<ul style="list-style-type: none"><li>. Diseño ux/UI.</li><li>. Realizar la documentación del sistema.</li><li>. Crear y mantener la base de datos.</li><li>. Desarrollo del sistema.</li></ul>

## Diagrama de casos de uso



## Especificaciones de casos de uso

<b>ID y Nombre:</b> CU001 Cargar Balneario
<b>Estado:</b> En proceso
<b>Descripción:</b> El cliente ingresa los datos del balneario que desea guardar
<b>Actor Principal:</b> Cliente
<b>Actor Secundario:</b> -

<b>Precondiciones:</b> -
<b>Puntos de Extensión:-</b>
<b>Condición:</b> El cliente ingresa a la aplicación y selecciona el botón "Create Resort"
<b>Escenario Principal:</b>  <ol style="list-style-type: none"><li>1. El cliente ingresa a la aplicación y selecciona el botón "Create Resort"</li><li>2. El sistema muestra la interfaz para cargar el nombre, la foto, los extras, las restricciones y el precio y la distribución de carpas del balneario.</li><li>3. El cliente ingresa al menos una carpa, una foto, un nombre y un precio para sus carpas.</li><li>4. El sistema verifica que los datos sean correctos y lo guarda en la base de datos.</li><li>5. El sistema informa al cliente que el balneario fue guardado y puede visualizarse en la sección "My resorts".</li><li>6. El sistema vacía el formulario y lo deja listo para volver a cargar la data.</li></ol>
<b>Flujos Alternativos:</b>  4.a El sistema determina que hay datos erróneos o faltantes, vuelve al punto 3.  5.b El sistema informa que hubo un error al guardar la información, vuelve al punto 2.
<b>Postcondiciones:</b> El usuario tiene un balneario asociado persistido en la base de datos.

<b>ID y Nombre:</b> CU002 Ver Balneario
<b>Estado:</b> En proceso
<b>Descripción:</b> El cliente puede ver sus balnearios.
<b>Actor Principal:</b> Cliente

<b>Actor Secundario: -</b>
<b>Precondiciones:</b> Debe haber balnearios cargados en el sistema asociados a dicho cliente.
<b>Puntos de Extensión:</b> CU003 Eliminar Balneario
<b>Condición:</b> El cliente selecciona el botón “My resorts” del menú principal.
<b>Escenario Principal:</b>  <ol style="list-style-type: none"><li>1. El cliente selecciona el botón “My resorts” del menú principal.</li><li>2. El sistema busca todos los balnearios asociados a dicho cliente y los imprime en pantalla.</li><li>3. El cliente puede visualizar y navegar por las páginas de balnearios existentes.</li></ol>
<b>Flujos Alternativos:</b>  2.a El sistema no encuentra ningún balneario asociado, no imprime nada en pantalla. Finaliza el caso de uso.
<b>Postcondiciones:</b> El cliente tiene una vista clara de todos los balnearios existentes a su nombre.

<b>ID y Nombre:</b> CU003 Eliminar Balneario
<b>Estado:</b> En proceso
<b>Descripción:</b> El cliente eliminará un balneario a su nombre.
<b>Actor Principal:</b> Cliente.

<b>Actor Secundario: -</b>
<b>Precondiciones:</b> debe haber balnearios a su nombre y el cliente los visualiza desde “My resorts”.
<b>Puntos de Extensión:-</b>
<b>Condición:</b> El cliente, una vez en “My resorts”, presiona el botón eliminar sobre el balneario que desea.
<b>Escenario Principal:</b>  <ol style="list-style-type: none"><li>1. El cliente, una vez en “My resorts”, presiona el botón eliminar sobre el balneario que desea.</li><li>2. El sistema chequea que no hayan reservas para dicho balneario.</li><li>3. El sistema borra el balneario e informa al cliente que pudo borrarse.</li></ol>
<b>Flujos Alternativos:</b>  <p>2.a El balneario posee reservas, estas se eliminan, se envía un email con la devolución del dinero a los usuarios y el balneario procede a borrarse imprimiendo una lista de los usuarios afectados, y generando notas de crédito correspondientes al valor de su alquiler. Sigue el punto 3.</p>
<b>Postcondiciones:</b>

<b>ID y Nombre:</b> CU004 Ver Alquileres
<b>Estado:</b> En proceso



<b>Descripción:</b> El cliente puede ver sus alquileres pendientes.
<b>Actor Principal:</b> Cliente
<b>Actor Secundario:</b> -
<b>Precondiciones:</b> Debe haber alquileres cargados en el sistema asociados a dicho cliente.
<b>Puntos de Extensión:-</b>
<b>Condición:</b> El cliente selecciona el botón “Bookings” del menú principal.
<b>Escenario Principal:</b> <ol style="list-style-type: none"><li>1. El cliente selecciona el botón “Bookings” del menú principal.</li><li>2. El sistema busca todos los alquileres asociados a los balnearios de dicho cliente y los imprime en pantalla.</li><li>3. El cliente puede visualizar y navegar por las páginas de alquileres existentes.</li></ol>
<b>Flujos Alternativos:</b>  2.a El sistema no encuentra ningún alquiler asociado, no imprime nada en pantalla. Finaliza el caso de uso.
<b>Postcondiciones:</b> El cliente tiene una vista clara de todos los alquileres existentes asociados a sus balnearios.

**ID y Nombre:** CU005 Ver Estadísticas

<b>Estado:</b> En proceso
<b>Descripción:</b> El cliente puede ver las estadísticas de sus balnearios
<b>Actor Principal:</b> Cliente
<b>Actor Secundario:</b> -
<b>Precondiciones:</b> Debe haber balnearios cargados en el sistema asociados a dicho cliente.
<b>Puntos de Extensión:-</b>
<b>Condición:</b> El cliente selecciona el botón “Ratings” del menú principal.
<b>Escenario Principal:</b> <ol style="list-style-type: none"><li>1. El cliente selecciona el botón “Ratings” del menú principal.</li><li>2. El sistema busca todas las estadísticas, puntuaciones, cantidad de alquileres, porcentaje de ocupación, etc. asociadas a los balnearios de dicho cliente y los imprime en pantalla.</li><li>3. El cliente puede visualizar y navegar por las páginas de alquileres existentes.</li></ol>
<b>Flujos Alternativos:</b> <p>2.a El sistema no encuentra ninguna estadística asociada, no imprime nada en pantalla. Finaliza el caso de uso.</p>
<b>Postcondiciones:</b> El cliente tiene una vista clara de todas las estadísticas existentes asociadas a sus balnearios.

**ID y Nombre:** CU006 Alquilar

<b>Estado: En proceso</b>
<b>Descripción:</b> El usuario alquila una carpa por un periodo de tiempo en un balneario a elección.
<b>Actor Principal:</b> Usuario
<b>Actor Secundario:</b> -
<b>Precondiciones:</b> Debe haber balnearios cargados en el sistema.
<b>Puntos de Extensión:-</b>
<b>Condición:</b> El usuario selecciona el botón "Book" del menú principal.
<b>Escenario Principal:</b>  <ol style="list-style-type: none"><li>1. El usuario selecciona el botón "Book" del menú principal.</li><li>2. El sistema muestra todos los balnearios disponibles y ofrece filtros de búsqueda por extras o políticas de niños y/o mascotas.</li><li>3. El usuario puede visualizar y navegar por las páginas de balnearios existentes.</li><li>4. El usuario selecciona el balneario que desea.</li><li>5. El sistema muestra la disposición de carpas.</li><li>5. El usuario ingresa el rango de fechas en el cual desea alquilar.</li><li>6.El sistema muestra que carpas están disponibles y cuales están reservadas.</li><li>7. El usuario selecciona las carpas que desea.</li><li>8. El sistema almacena el alquiler y genera un código QR para realizar el pago.</li><li>10. El cliente realiza el pago.</li></ol>
<b>Flujos Alternativos:</b>  4.a El usuario no encuentra ningún balneario que se ajuste a sus requerimientos, finaliza el caso de uso. 6.a No hay carpas disponibles para la fecha seleccionada, se vuelve al punto 4.

**Postcondiciones:** El usuario tiene su alquiler en “My bookings” y este alquiler está persistiendo en la base de datos.

**ID y Nombre:** CU007 Ver Alquileres

**Estado:** En proceso

**Descripción:** El usuario puede ver sus alquileres, vigentes o vencidos.

**Actor Principal:** Usuario

**Actor Secundario:** -

**Precondiciones:** Debe haber alquileres cargados en el sistema asociados a dicho usuario.

**Puntos de Extensión:** CU008 Cancelar reserva, CU009 Dejar Opinion.

**Condición:** El usuario selecciona el botón “My Bookings” del menú principal.

**Escenario Principal:**

1. El usuario selecciona el botón “My Bookings” del menú principal.
2. El sistema busca todos los alquileres asociados al usuario y los imprime en pantalla separando en vigente y vencidos .
3. El usuario puede visualizar y navegar por las páginas de alquileres existentes.

**Flujos Alternativos:**

2.a El sistema no encuentra ningún alquiler asociado a dicho usuario, no imprime nada en pantalla. Finaliza el caso de uso.

3.a El cliente decide cancelar un alquiler vigente –CU008 Cancelar reserva.

3.b El cliente decide dejar una opinión sobre un alquiler pasado –CU009 Dejar Opinión.

**Postcondiciones:** El usuario tiene una vista clara de todas los alquileres existentes asociadas a sí mismo.

<b>ID y Nombre:</b> CU008 Cancelar Reserva
<b>Estado:</b> En proceso
<b>Descripción:</b> El usuario cancela una reserva.
<b>Actor Principal:</b> Usuario
<b>Actor Secundario:</b> -
<b>Precondiciones:</b> Debe haber una reserva con fecha límite hasta 48 horas antes.
<b>Puntos de Extensión:-</b>
<b>Condición:</b> El usuario selecciona el botón “Cancelar” de la reserva deseada.

**Escenario Principal:**

1. El usuario selecciona el botón “Cancelar” de la reserva deseada.
2. El sistema confirma que la fecha de la reserva es desde 48 horas mayor de la actual.
3. El sistema elimina la reserva de la base de datos.
4. El sistema genera una nota de crédito por el valor del alquiler y la envía por mail.
4. El usuario ve reflejado el cambio en “My bookings”.

**Flujos Alternativos:**

2.a La reserva se encuentra excedida de la fecha soportada, se informa al usuario y finaliza el caso de uso.

**Postcondiciones:** El usuario deja de observar dicha reserva en la sección “My bookings”.

**ID y Nombre:** CU009 Dejar opinión

**Estado:** En proceso

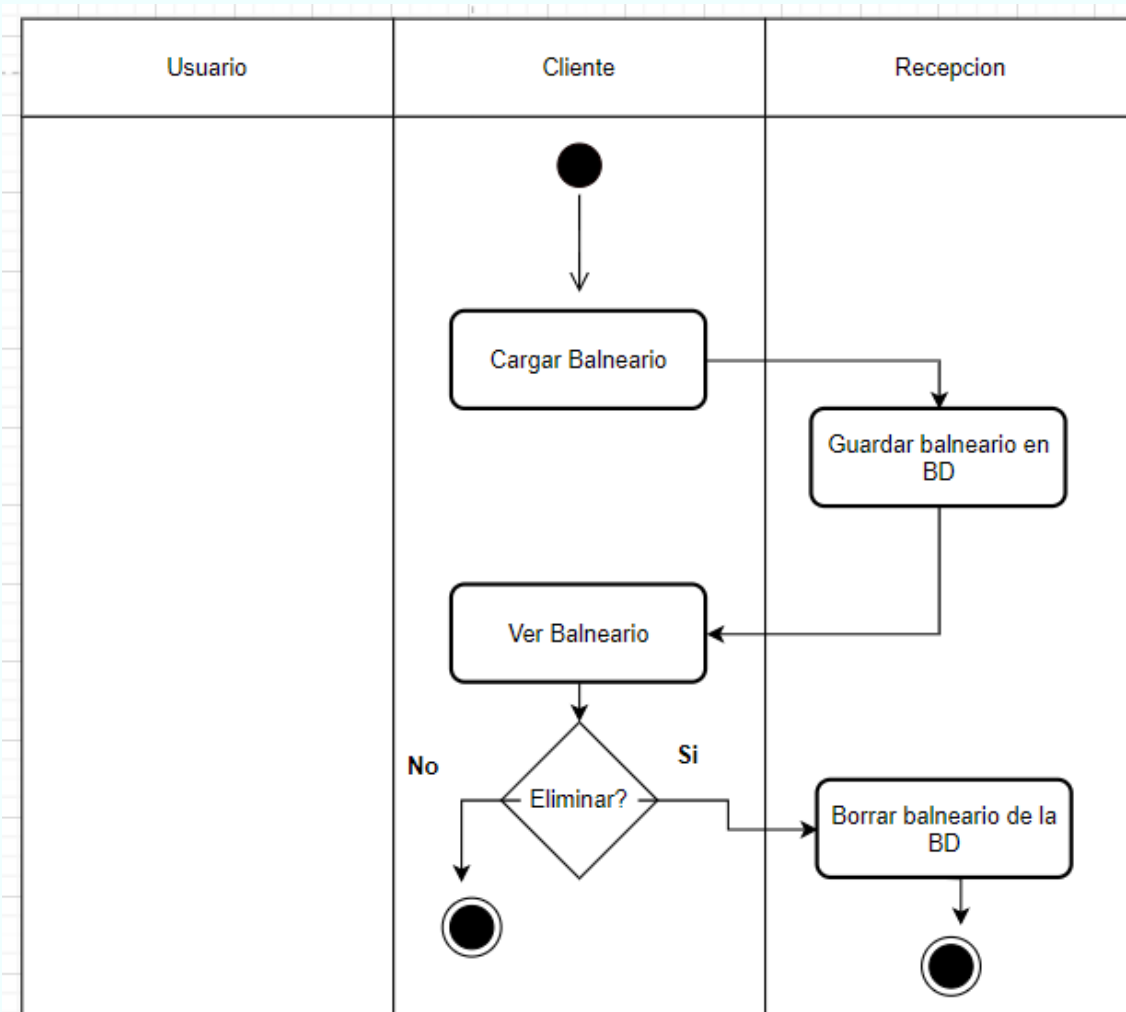
**Descripción:** El usuario da una reseña sobre una reserva pasada.

**Actor Principal:** Usuario

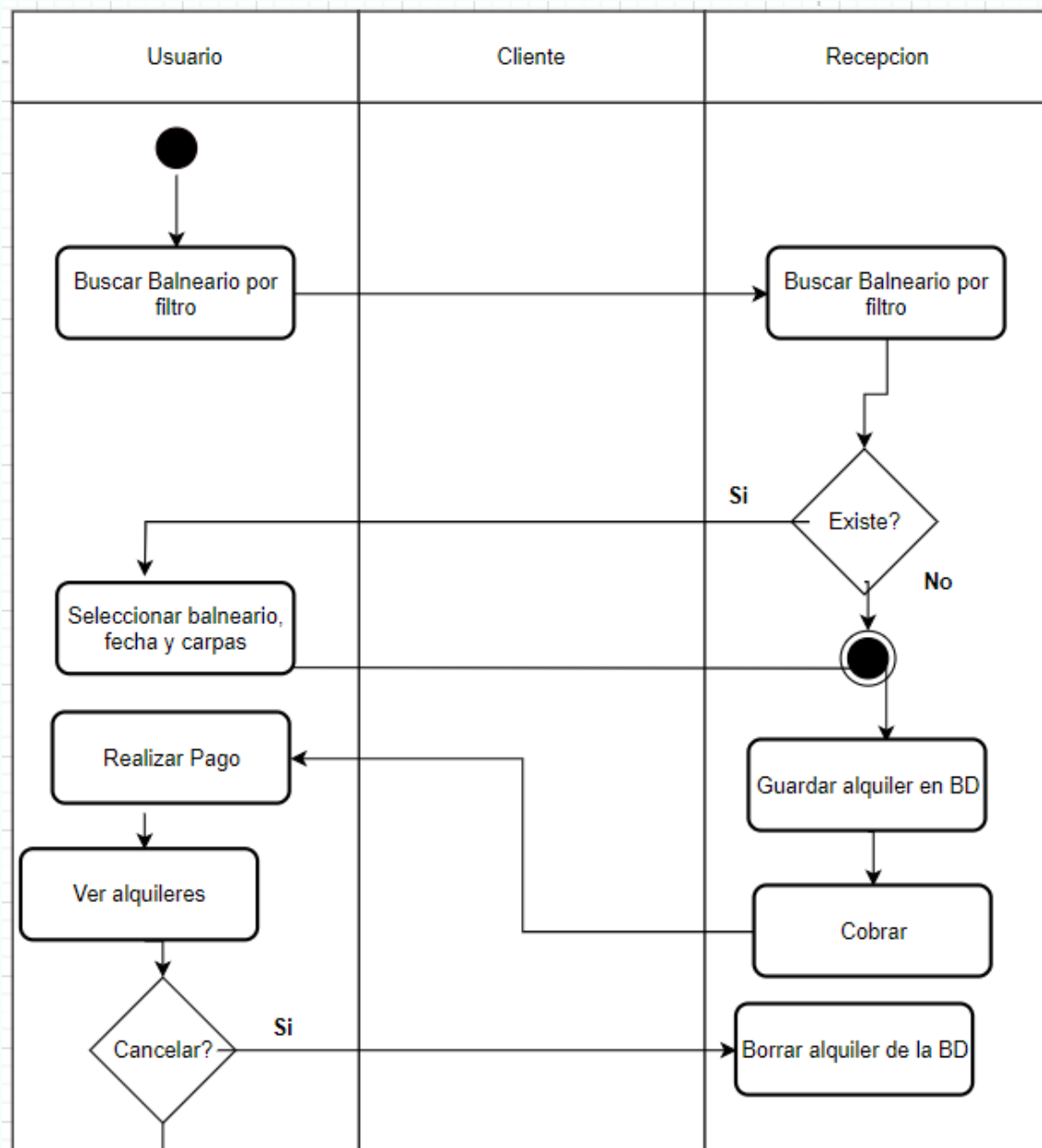
**Actor Secundario:** -

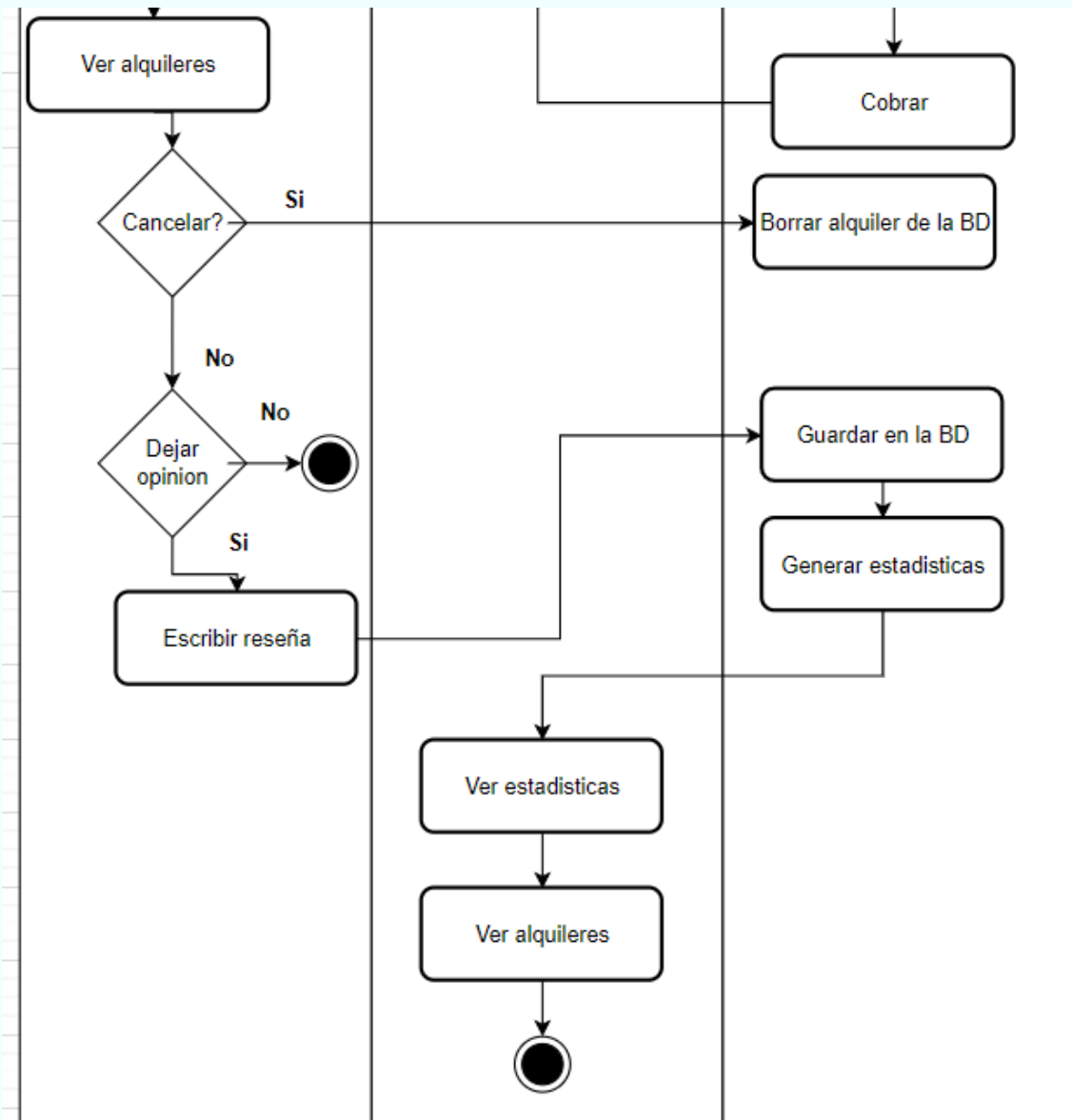
<b>Precondiciones:</b> Debe haber una reserva con fecha anterior a la actualidad.
<b>Puntos de Extensión:-</b>
<b>Condición:</b> El usuario selecciona el botón “Review” de la reserva deseada.
<b>Escenario Principal:</b>  <ol style="list-style-type: none"><li>1. El usuario selecciona el botón “Review” de la reserva deseada.</li><li>2. El sistema confirma que la fecha de la reserva es anterior a la actual.</li><li>3. El sistema muestra un formulario.</li><li>4. El usuario ingresa la cantidad de estrellas (1 a 5) y una opinión escrita sobre un balneario en particular.</li><li>5. El sistema almacena la información para ser visible por los balnearios y recalcula la cantidad de estrellas asociadas a dicho balenarrio.</li></ol>
<b>Flujos Alternativos:</b>  2.a La reserva se encuentra anterior a la fecha soportada, se informa al usuario y finaliza el caso de uso.
<b>Postcondiciones:</b> El balneario posee un nuevo feedback almacenado.

## Diagrama de actividad por carriles



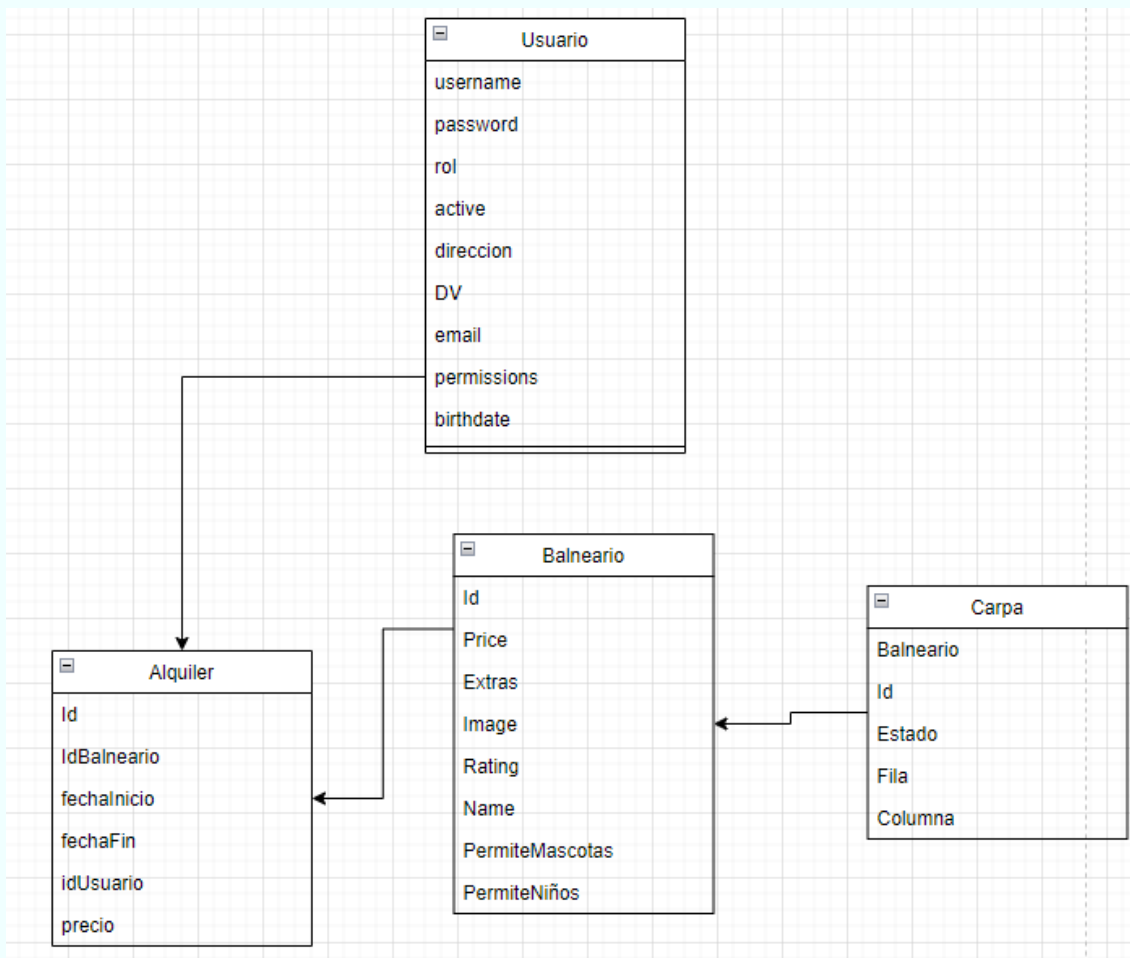




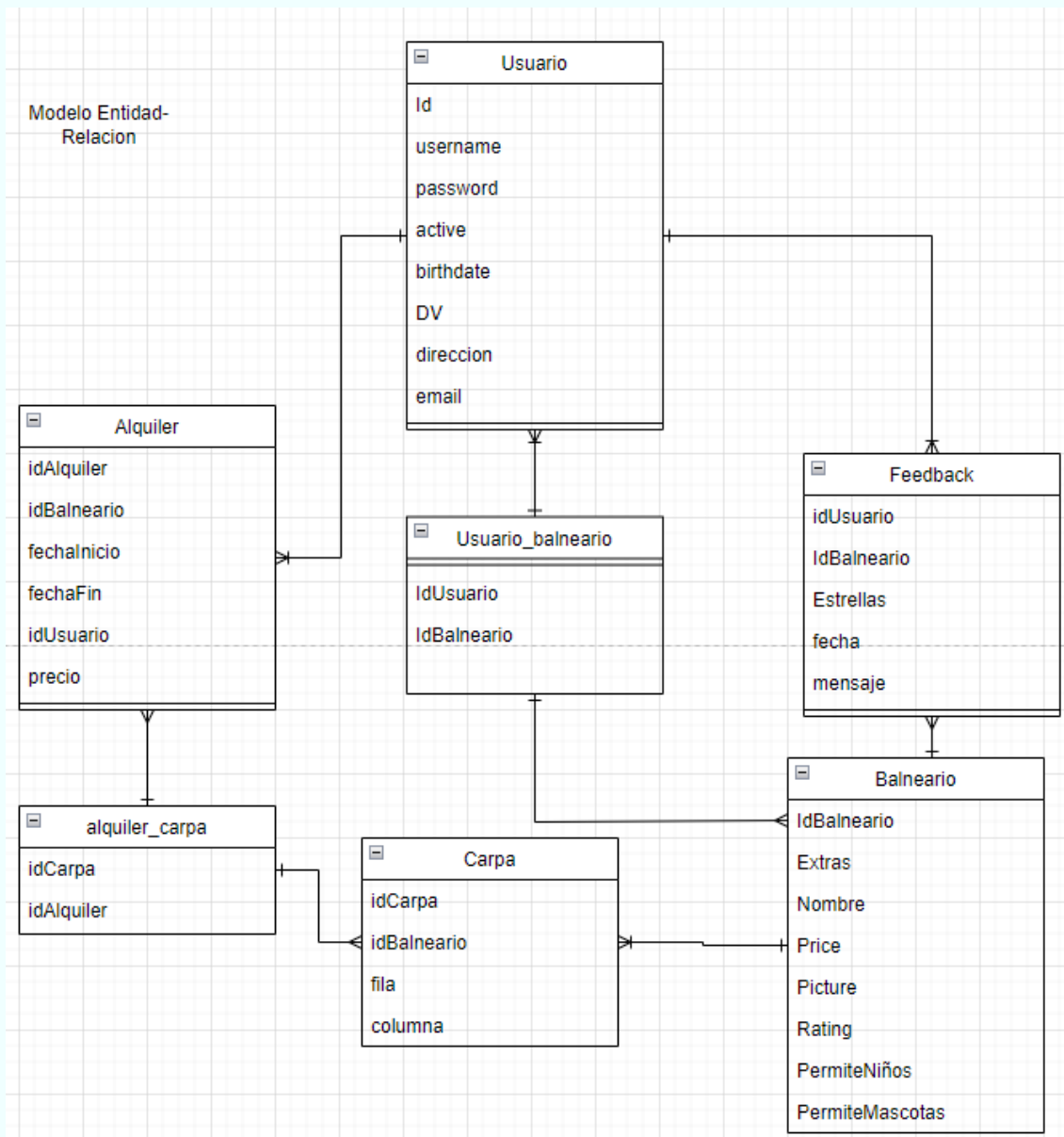


## Diagrama de secuencia

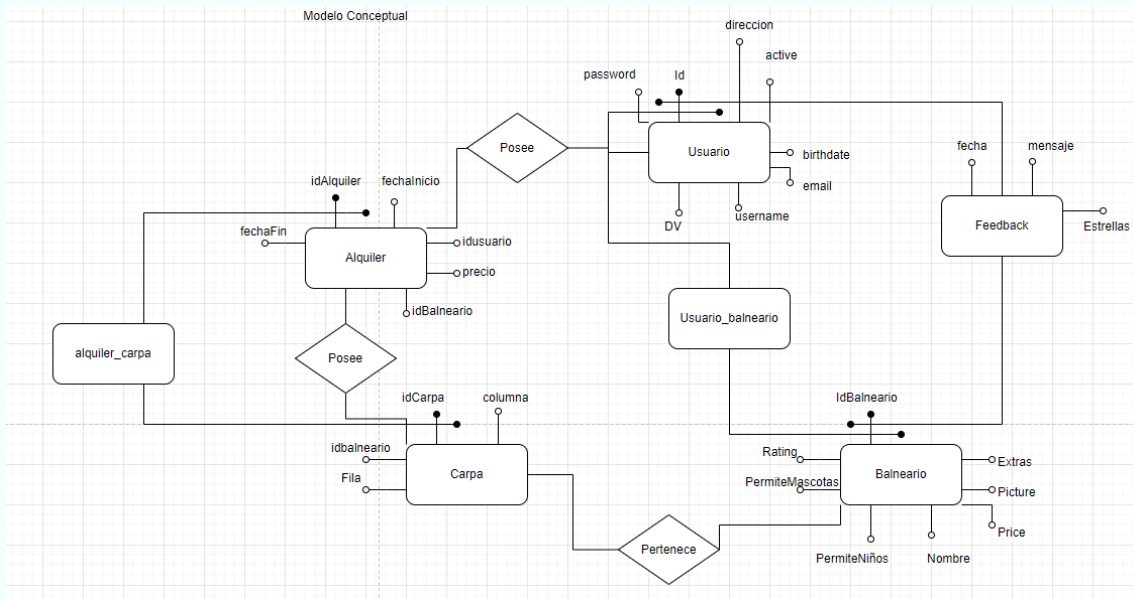
## Diagrama de clases



## Modelo de datos

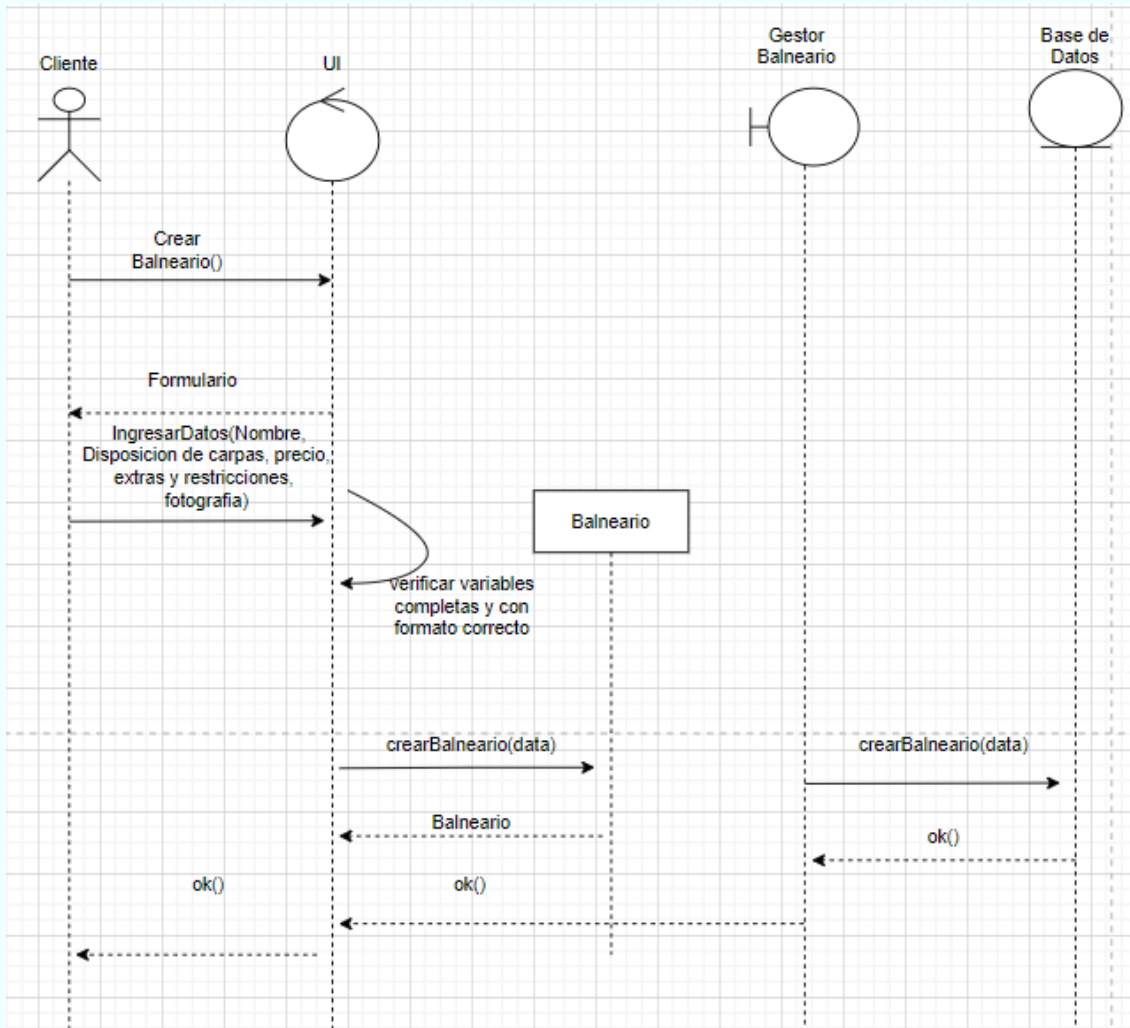


## Modelo Conceptual

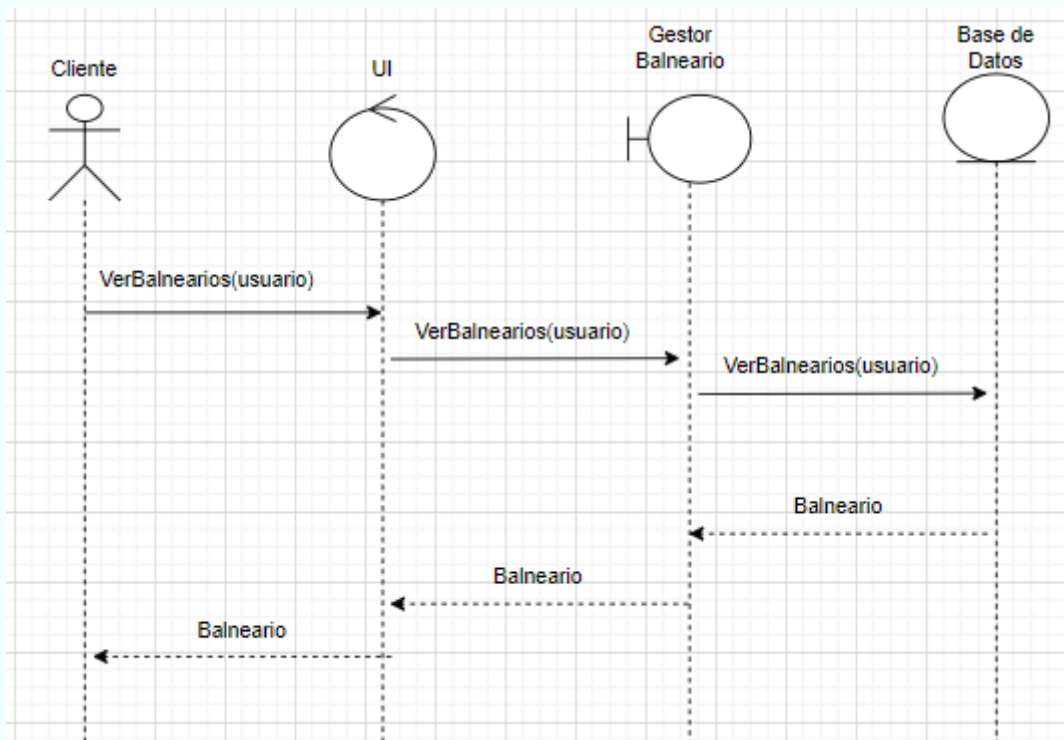


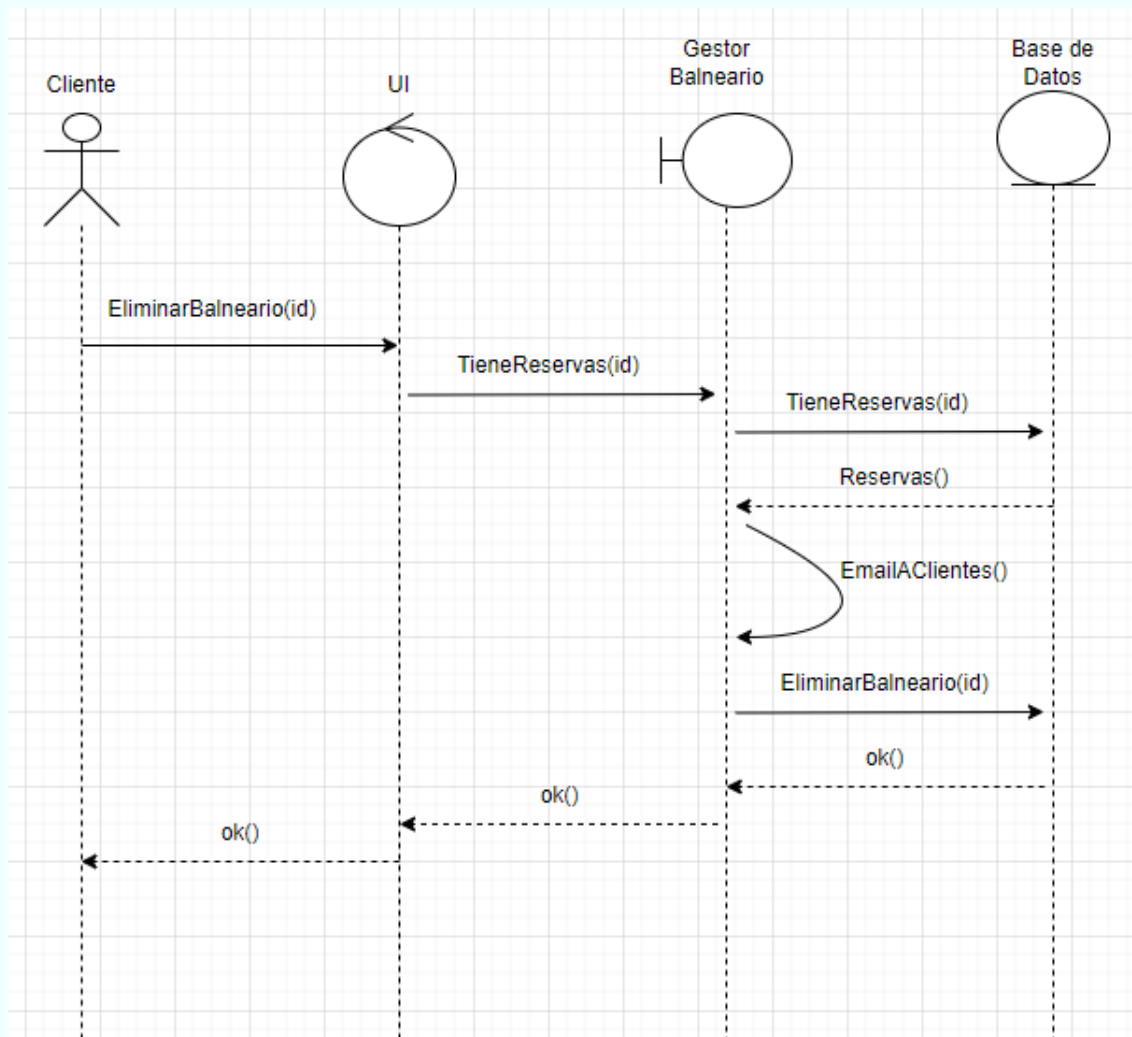
## Diagramas de secuencia

### Crear Balneario



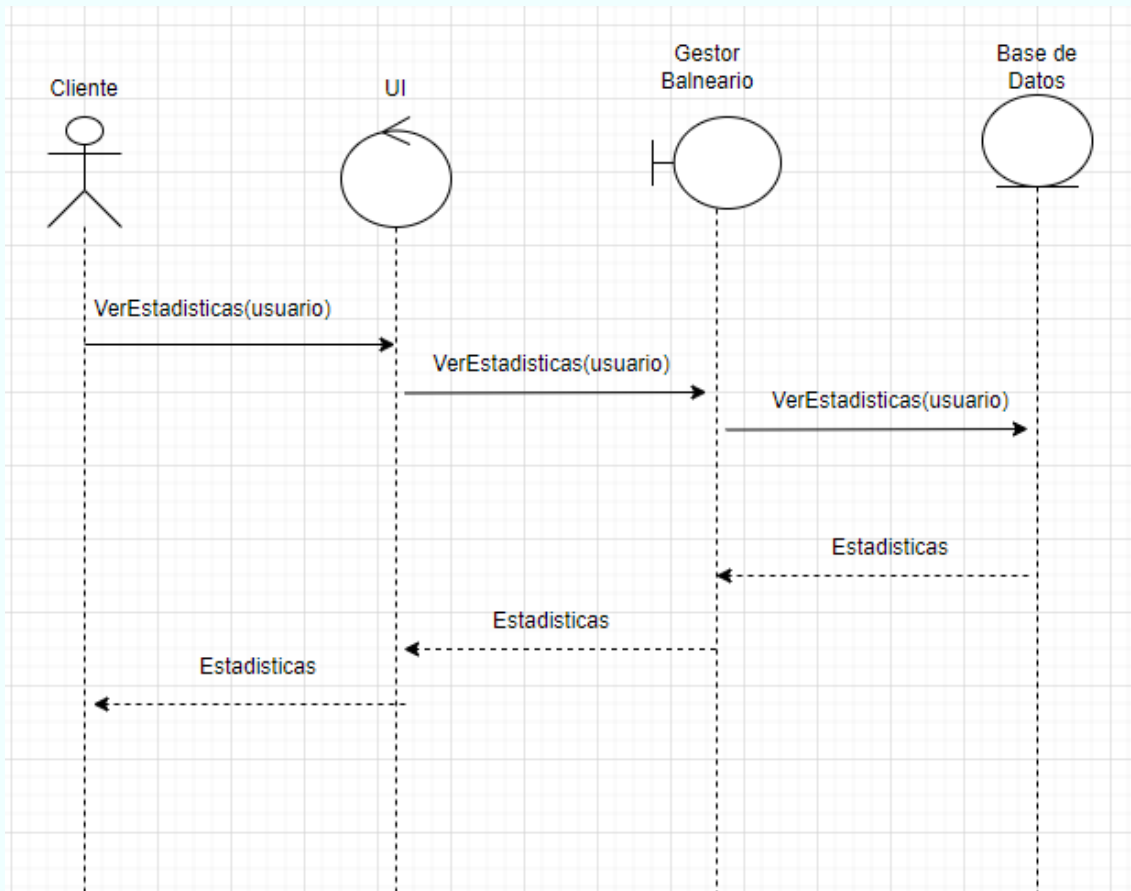
## Var Balnearios

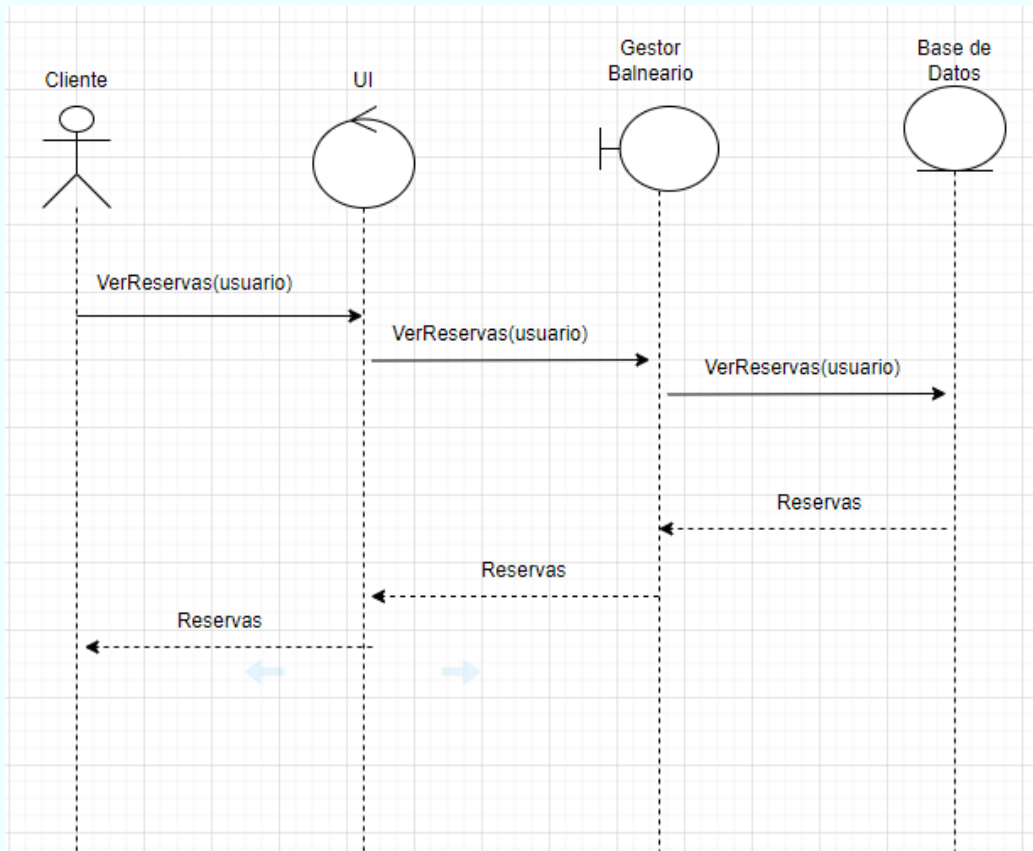
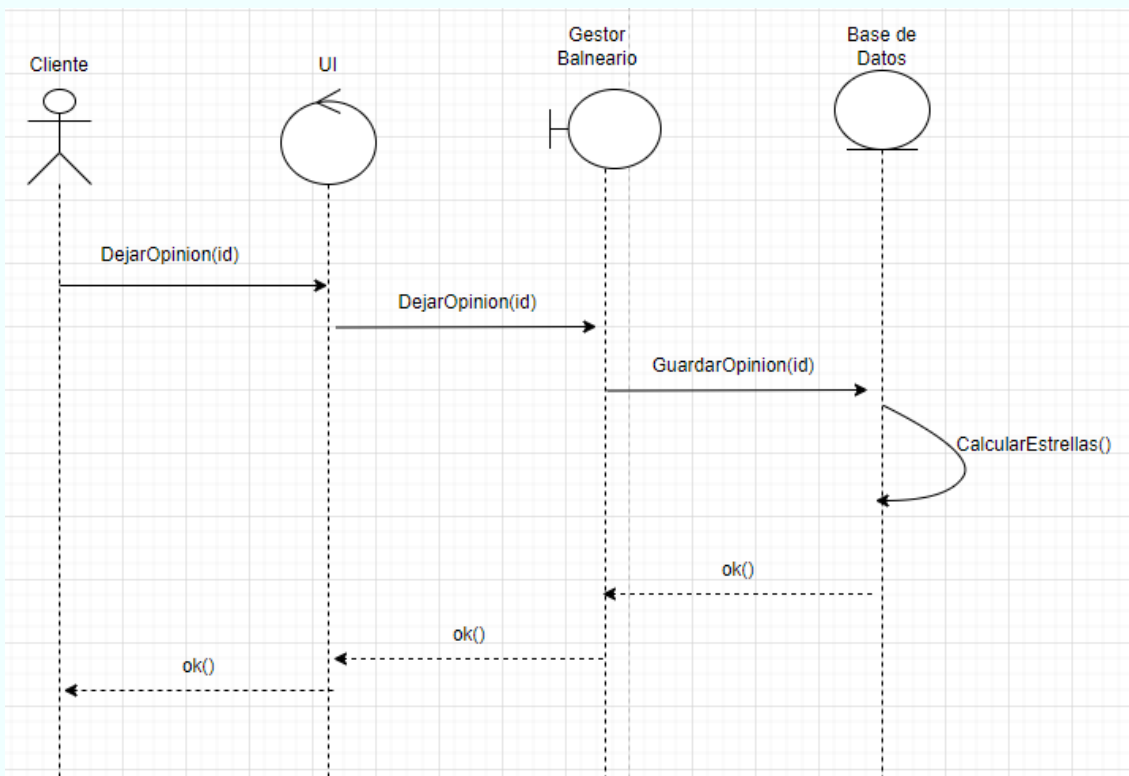


**Borrar Balneario**

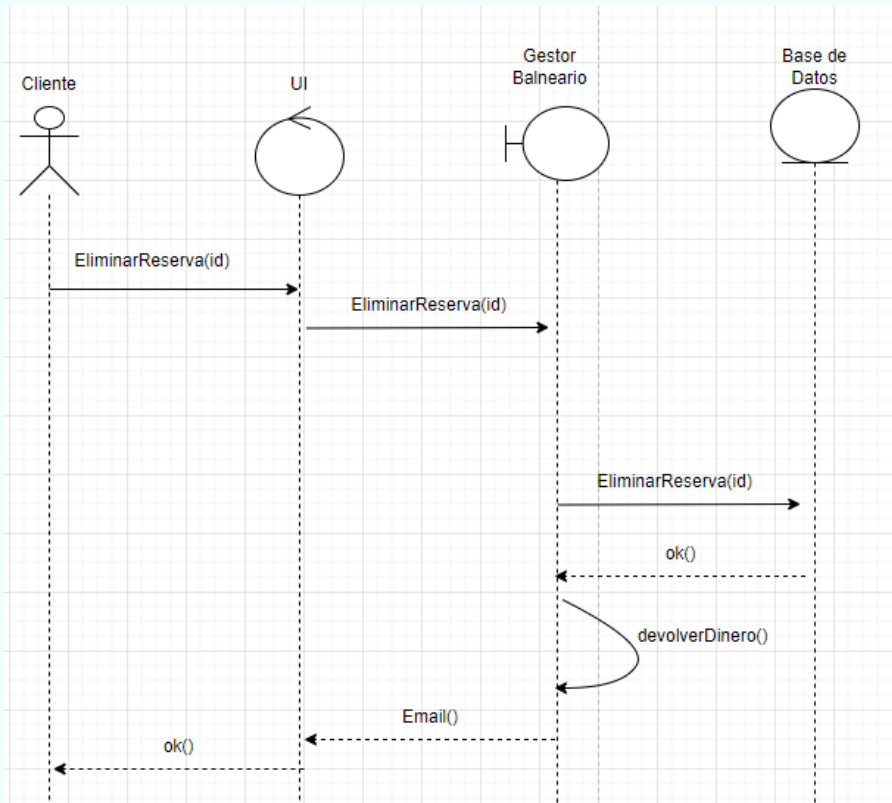


## Ver Estadísticas

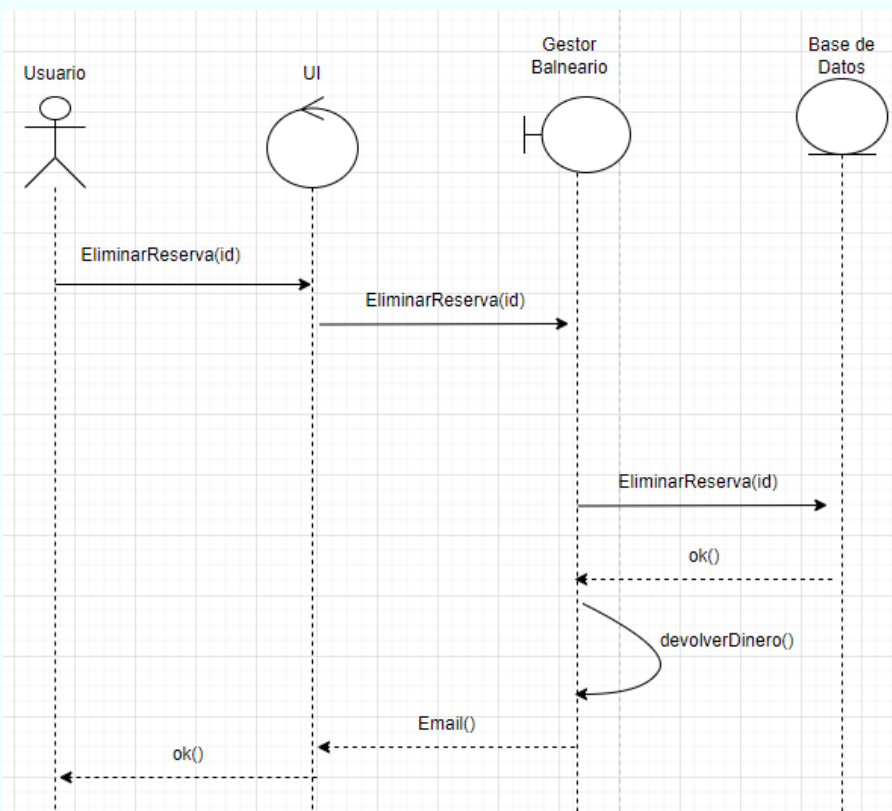


**Ver reservas****Dejar Opinión**

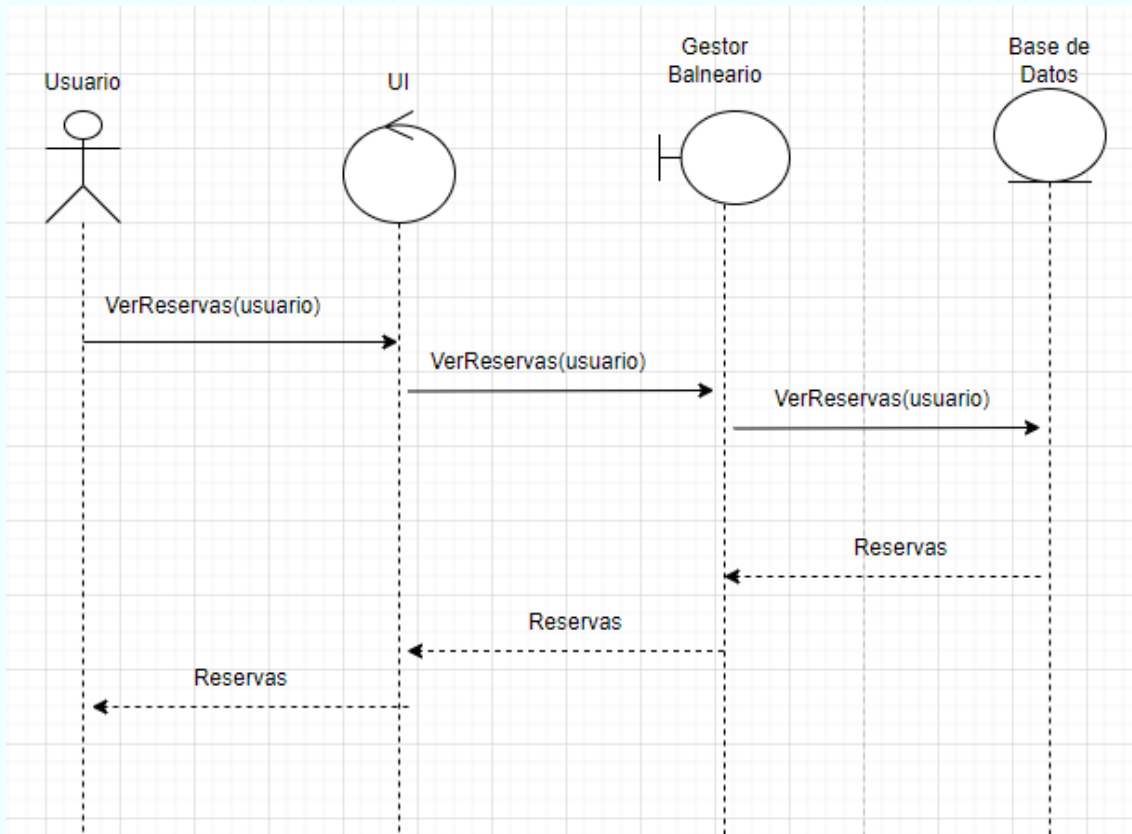
### Eliminar reserva (cliente)

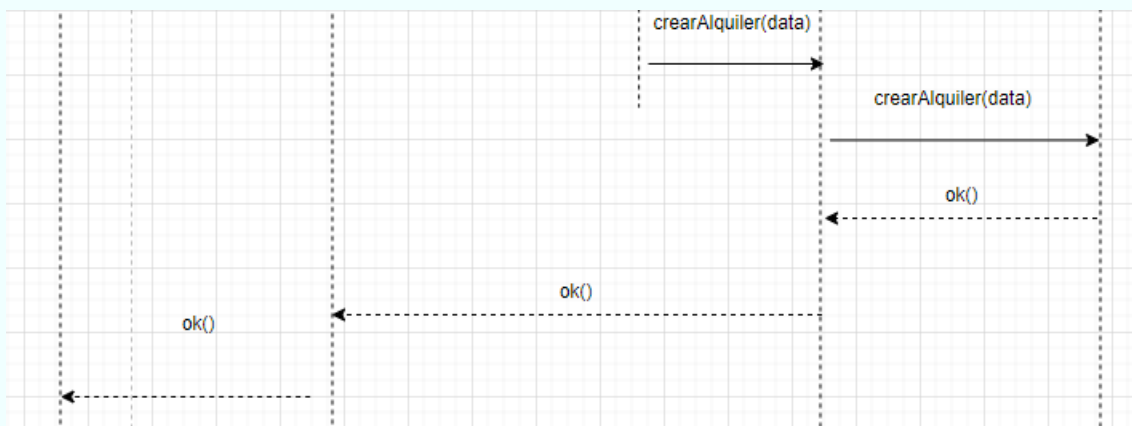
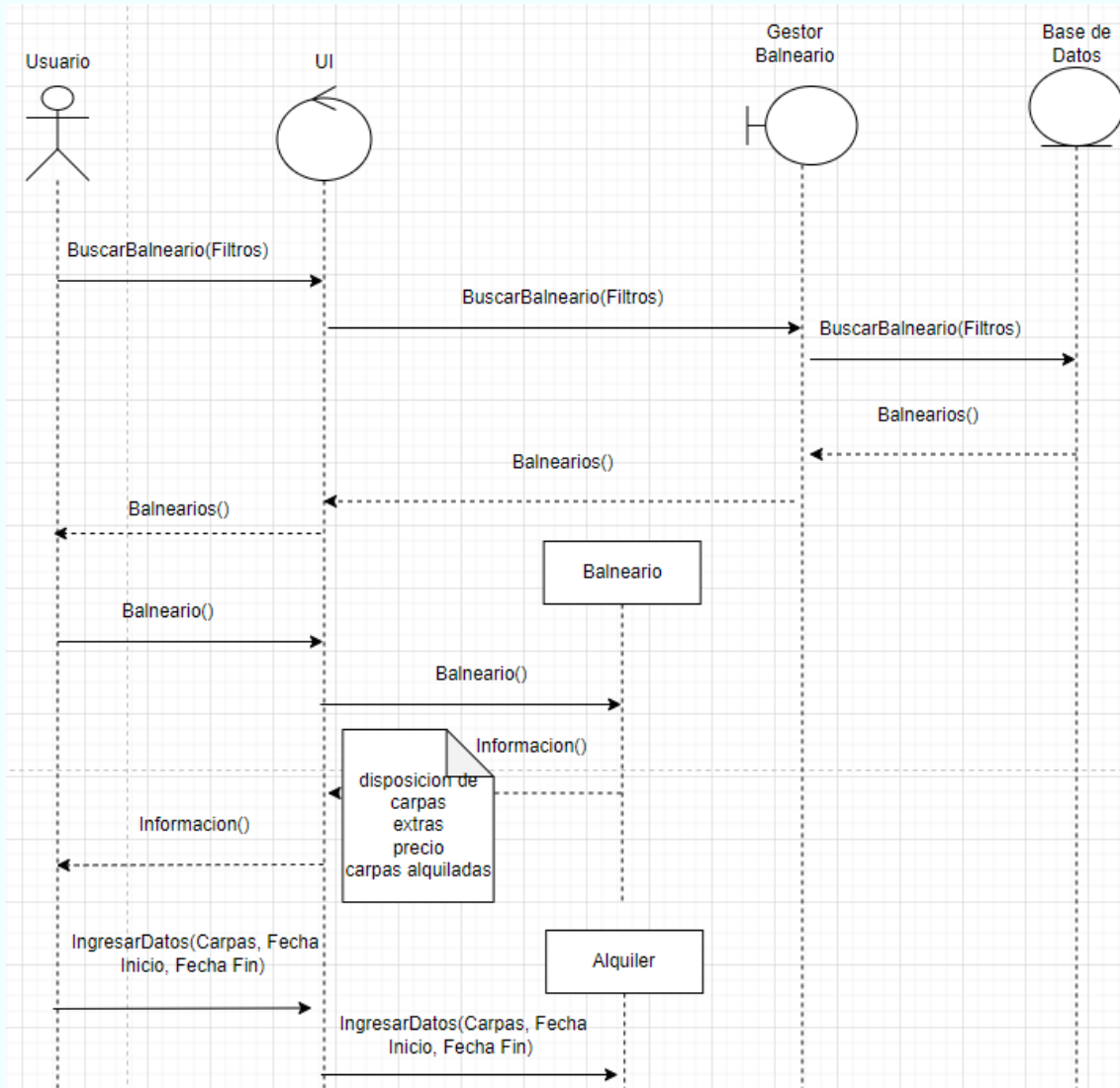


### Eliminar Reserva (usuario)



### Ver Reservas (usuario)



**Alquilar**

## Planificación del proyecto

### Mes 1: Planificación y Diseño Inicial

#### Semana 1-2 (Hito 1 - Planificación):

Definir requisitos funcionales y no funcionales.  
Crear documento de especificaciones del proyecto.  
Establecer equipo de desarrollo.

#### Semana 3-4 (Hito 2 - Diseño Inicial):

Diseñar la estructura de la base de datos.  
Crear prototipos de interfaz de usuario.

### Mes 2: Desarrollo y Pruebas

#### Semana 5-8 (Hito 3 - Desarrollo):

Desarrollar funciones de gestión de balnearios y carpas.  
Implementar lógica de alquiler y cancelación.  
Configurar autenticación de usuarios y roles.  
Desarrollar interfaz de usuario para usuarios y dueños de balnearios.

#### Semana 9-10 (Hito 4 - Frontend):

Agregar funcionalidad de búsqueda y filtrado de balnearios.  
Comenzar guía de instalador.

#### Semana 11-12 (Hito 5 - Pruebas):

Realizar pruebas de integración y rendimiento.  
Realizar instalador.

### Mes 3: Despliegue, Capacitación y Lanzamiento

#### Semana 13-14 (Hito 6 - Despliegue):

Realizar pruebas finales en entorno de producción (como aplicación de escritorio).

#### Semana 15-16 (Hito 7 - Capacitación):

Capacitar a los dueños de balnearios en el uso del sistema.  
Proporcionar documentación de usuario.

#### Semana 17-18 (Hito 8 - Lanzamiento):

Lanzar oficialmente el sistema de alquiler de carpas.

Monitorear rendimiento y estabilidad.

**Semana 19 (Hito 9 - Evaluación Final):**

Realizar evaluación general del proyecto.