PROYECTO FINAL DAM - SUBSIFY

Julián Ojea López

Índice

1. Introducción

Este proyecto es la adaptación de la aplicación web “Subsify” en la cual he participado en su desarrollo. De esta manera, se ha utilizado el backend implementado para, sobre él, construir tanto la versión de escritorio como la móvil. Por no entrar en el ámbito del ciclo en esta memoria no consta la definición de parte de las funcionalidades que contiene la web pero sí que se detalla su estructura de datos y su api rest así como los requisitos de usuario de escritorio y móvil que he desarrollado.

Subsify es un gestor de suscripciones online que permite al usuario realizar un seguimiento de todos sus compromisos con servicios digitales en un solo lugar.

Hoy en día vivimos en una era digital donde la oferta de servicios es amplia y variada, desde plataformas de entretenimiento y formación en línea hasta herramientas de productividad y software especializado. La mayoría de estos servicios adoptan un modelo de suscripción, en general, con precios muy asequibles. Sin embargo, esto puede derivar en que los usuarios acaben suscritos a servicios que apenas utilizan y por tanto realicen gastos innecesarios.

Subsify surge como respuesta a esta realidad, tratando de satisfacer la creciente necesidad de los usuarios de tener un mayor control sobre sus suscripciones digitales. Subsify ofrece una visión clara del gasto mensual del usuario, lo que facilita la toma de decisiones sobre qué servicios mantener y/o cancelar.

El desarrollo se Subsify tiene una serie de objetivos fundamentales que buscan mejorar la experiencia digital de sus usuarios:

Centralización de suscripciones: la aplicación actúa como un repositorio centralizado de todas las suscripciones digitales de un usuario. De esta forma se elimina la necesidad de consultar múltiples servicios para conocer importes y/o fechas de renovación
Recordatorios: Subsify informa a sus usuarios de las fechas en las que se producirán las renovaciones de los servicios que tiene contratados, de forma que pueda planificar y tomar decisiones sobre la continuidad de sus suscripciones. Además, recuerda al usuario que debe dar de baja aquellas suscripciones que haya decidido cancelar
Gestión de cancelaciones: Subsify trata de simplificar el proceso ofreciendo enlaces directos a las diferentes plataformas para proceder a la baja del servicio
Facilidad de uso: la aplicación ofrece una interfaz intuitiva y fácil de usar, garantizando que los usuarios puedan sacar el máximo provecho de las funcionalidades desarrolladas

2. Desarrollo

2.1. Tecnologías utilizadas:

Backend
Ha sido desarrollado en Java con ayuda del framework Spring para la inyección de dependencias y el acceso a datos. Además, se ha utilizado Ontimize Boot, framework basado en Spring Boot, que simplifica el desarrollo de una API REST.
Como herramienta de compilación y gestión de dependencias se ha empleado Maven.

Frontend
Se ha empleado Windows Forms y Android para realizar las interfaces de escritorio y móvil respectivamente.

Base de datos
El SGBD elegido es PostgreSQL, por ser de código abierto y ofrecer un gran rendimiento.
Otras herramientas
IDEs: Visual Studio 2022 y Android Studio(frontend), IntelliJ (backend)
Administrador de BBDD: DBeaver
Prueba de API: Postman
Control de versiones: Git y Github como escritorio remoto

2.2. Análisis:

La aplicación ha sido diseñada para atender a dos tipos de usuarios distintos: el Usuario Registrado y el Administrador:

- Usuario registrado(cliente), el que utilizará la aplicación para hacer seguimiento de sus servicios mediante la selección de plataformas y sus planes asociados. Este usuario podrá añadir nuevas suscripciones, listarlas y actualizarlas.
- Administrador, cuya función principal es la de mantener y dar de alta las plataformas y los planes a los que los usuarios se suscriben. Para ello podrá listar y añadir tanto plataformas como planes.

A continuación se definirán las funcionalidades de cada uno de los roles mediante el uso de diagramas de casos de uso:

Diagrama de usuario:

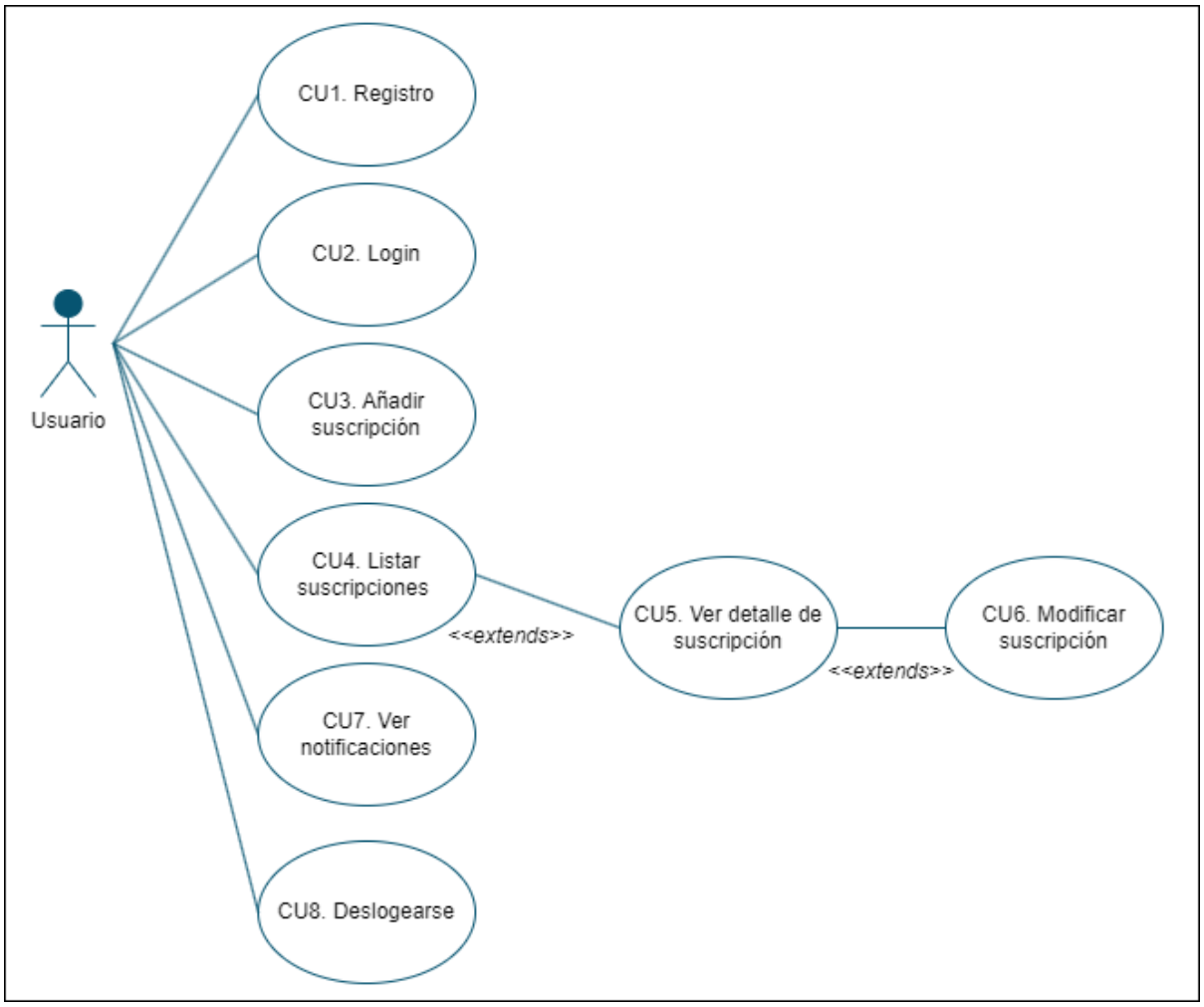

Diagrama1_casos_de_uso_de_usuario

Descripción de los casos de uso de usuario:

CU1. Registro. El usuario podrá registrarse en la plataforma. Tendrá que especificar los siguientes campos:

- Nombre de usuario
- Email
- Contraseña
- Una vez que los campos se han introducido correctamente el usuario quedará almacenado en la aplicación y se podrá logear.
- Validaciones:
- Si el nombre de usuario ya existe en la aplicación el usuario no se podrá registrar.
- Para introducir la contraseña existirán dos campos uno: “contraseña” y “repetir contraseña” así se validará que el usuario la ha introducido de manera correcta.

En el caso que las contraseñas sean diferentes se mostrará un mensaje de error y el usuario no se podrá registrar.
- El campo email también estará validado para que se introduzca siguiendo un formato correcto.

CU3. Añadir suscripción. El usuario podrá añadir una nueva suscripción introduciendo los siguientes datos:
- Nombre de plataforma, el cual será seleccionado de una lista desplegable con las plataformas existentes en la aplicación.
- Nombre de plan, será seleccionado también de una lista desplegable con los planes de la plataforma seleccionada en el campo anterior.
- Precio, se sugerirá automáticamente al seleccionar el nombre del plan.
- Fecha de inicio, el usuario seleccionará la fecha de inicio en la que su suscripción da comienzo.

Una vez añadida la suscripción ésta aparecerá en la lista de suscripciones y también en la pantalla principal de la aplicación. La fecha de fin se calculará con respecto a la frecuencia que tiene el plan seleccionado, es decir, si el plan tiene una frecuencia mensual y el usuario ha especificado una fecha de inicio del 1 de diciembre de 2023, la fecha de fin será 1 de enero de 2024.
Validaciones: Todos los campos tendrán que ser rellenados para poder añadir la suscripción.

CU4. Listar suscripciones. El usuario podrá ver un listado con todas las suscripciones activas, los datos que se mostrarán en cada fila serán los siguientes:
- Nombre de la plataforma.
- Nombre plan de suscripción.
- Precio mensual: es el precio que cuesta la suscripción mensualmente, en el caso de que la suscripción tenga una frecuencia mayor a la mensual el precio se calculará automáticamente mostrando el que valdría este mes.

Además se mostrará un sumatorio de todos los precios mensuales que indicará el gasto total mensual.
En este listado sólo se muestran las suscripciones que están activas, si una suscripción ha finalizado y no se quiere renovar ya no se generará una nueva suscripción y no se mostrará en este listado. Las suscripciones pasadas tampoco se muestran en este listado.

CU5. Ver detalle de suscripción: al seleccionar un elemento de la lista de suscripciones (CU4) se accederá al detalle de la suscripción. El cual muestra la siguiente información:
- Frecuencia de suscripción.
- Inicio de la suscripción: fecha de inicio de la suscripción.
- Próxima renovación: fecha de la próxima renovación.
- Renovar: checkbox que indica si se quiere renovar la suscripción. En el caso de que esté marcado, la suscripción se renovará, en caso contrario no se producirá la renovación.

CU6. **Modificar suscripción**: Una vez en el detalle de suscripción (CU8) se podrá presionar sobre el checkbox de renovación para que la suscripción no se renueve o se renueve en el caso de que no esté seleccionado. En ambos escenarios una vez modificado el checkbox aparecerá un botón de guardado para hacer los cambios efectivos. Cuando el botón de guardado se presiona se informará mediante un modal de que la suscripción se ha actualizado.

CU7. Ver notificaciones: las notificaciones se mostrarán en la pantalla principal de la aplicación a modo de botón el cual, al presionarlo redirige a la plataforma asociada para poder darse de baja. Este botón solo se muestra en el caso de que el usuario quiera no quiera seguir suscrito a un servicio. En esta pantalla principal se existirá una tarjeta por suscripción que se vaya a renovar con los siguientes datos:

- Icono de la plataforma.
- Nombre de la plataforma.
- Fecha en la que finaliza la suscripción.
- Número de días que faltan para que finalice.
- Precio con el que la suscripción se renueva.

En el caso de que no haya suscripciones activas se mostrará un mensaje informando de que no existen suscripciones.

Diagrama de administrador:

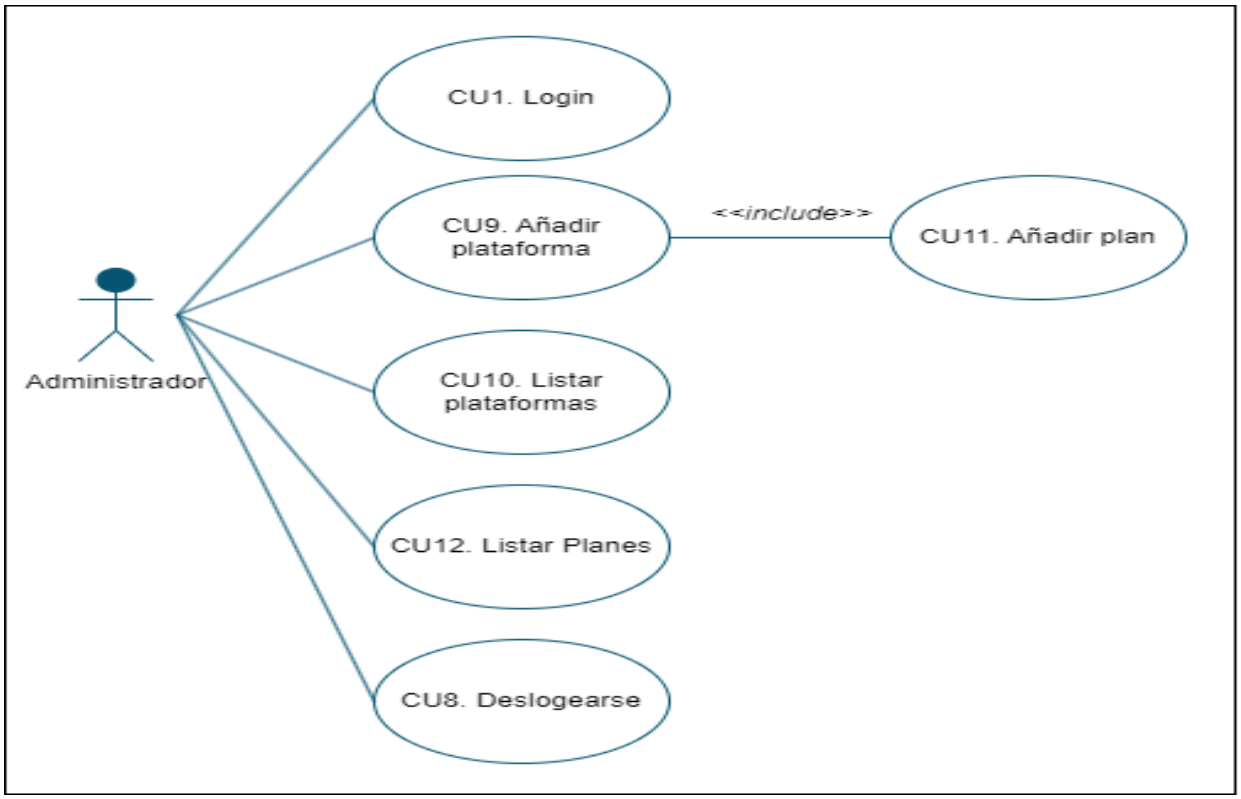

Diagrama2_casos_de_uso_de_administrador

Casos de uso de administrador:

CU9. Añadir plataforma: El administrador podrá añadir una nueva plataforma introduciendo los siguientes campos:

- Nombre de plataforma.
- Categoría de plataforma: combobox con las categorías existentes
- Enlace de plataforma: este es el link al que se redirigirá el usuario cuando quiera desuscribirse de su servicio.
- Imagen: este campo se seleccionará presionando un botón que abrirá un menú contextual que permitirá al administrador seleccionar un icono desde su sistema de archivos. Una vez seleccionado se previsualizará en el mismo formulario de añadir plataforma.

Una vez el administrador confirma los campos y estos están correctamente introducidos la plataforma se añadirá al listado de plataformas y ésta ya podrá ser seleccionada por el usuario cuando quiera añadir una suscripción.
Validaciones: todos los campos tendrán que rellenarse para poder añadir la plataforma.

- Imagen: la imagen deberá estar en formato png.

CU10. Listar plataforma: El administrador podrá visualizar todas las plataformas existentes en un listado, se mostrarán los siguientes campos en cada fila:

- Nombre de plataforma.
- Categoría.

CU11. Añadir plan: El administrador podrá añadir un nuevo plan introduciendo los siguientes campos:

- Nombre de plataforma: combobox con las plataformas existentes en la aplicación.
- Nombre del plan.
- Precio del plan.
- Fecha de inicio del plan.
- Frecuencia del plan: combobox con las frecuencias existentes en la aplicación.

Una vez que el administrador confirma los campos y estos están correctamente introducidos el plan se añadirá al listado de plataformas y éste podrá ser seleccionado por el usuario cuando quiera añadir una suscripción.
Validaciones: todos los campos tendrán que rellenarse para poder añadir la plataforma.

- Precio del plan. Tendrá que ser un valor entero o decimal.
- Fecha de inicio. La fecha tendrá que ser anterior a la actual.

CU12. Listar planes. El administrador podrá visualizar todos los planes existentes en un listado, se mostrarán los siguientes campos en cada fila:

- Nombre del plan.
- Nombre de la plataforma:
- Frecuencia del plan.
- Precio del plan.

Casos de uso compartidos:

CU2. Login. El usuario podrá acceder a la aplicación introduciendo los siguientes datos:

- Nombre de usuario
- Contraseña
- Cuando sean escritos correctamente el usuario podrá acceder a la aplicación.
- Validaciones:
- Si el nombre de usuario no existe en la base de datos no podrá haber login.
- Si la contraseña se ha introducido incorrectamente tampoco se podrá logear.

CU8. Desloguearse. El usuario podrá salir de la aplicación presionando el botón de salir lo que le redirigirá a la pantalla de login.

2.3. Diseño

Una vez definidos los requisitos del sistema se expondrá su estructura de base de datos:

Diagramas Entidad Relación:

(Para mejor visualización se recomienda consultar los diagramas correspondientes en la carpeta Esquemas)

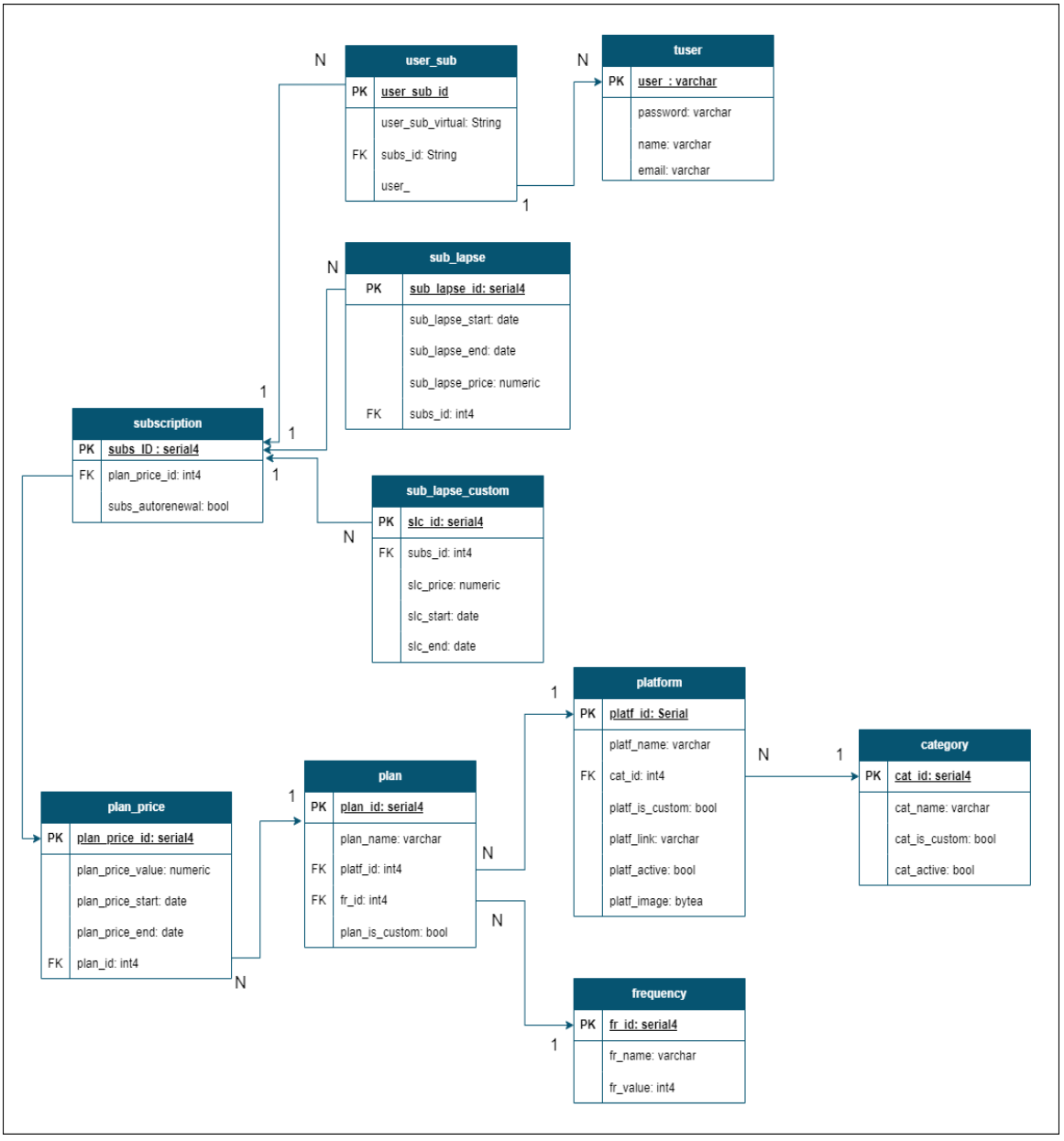

Diagrama3_ER1

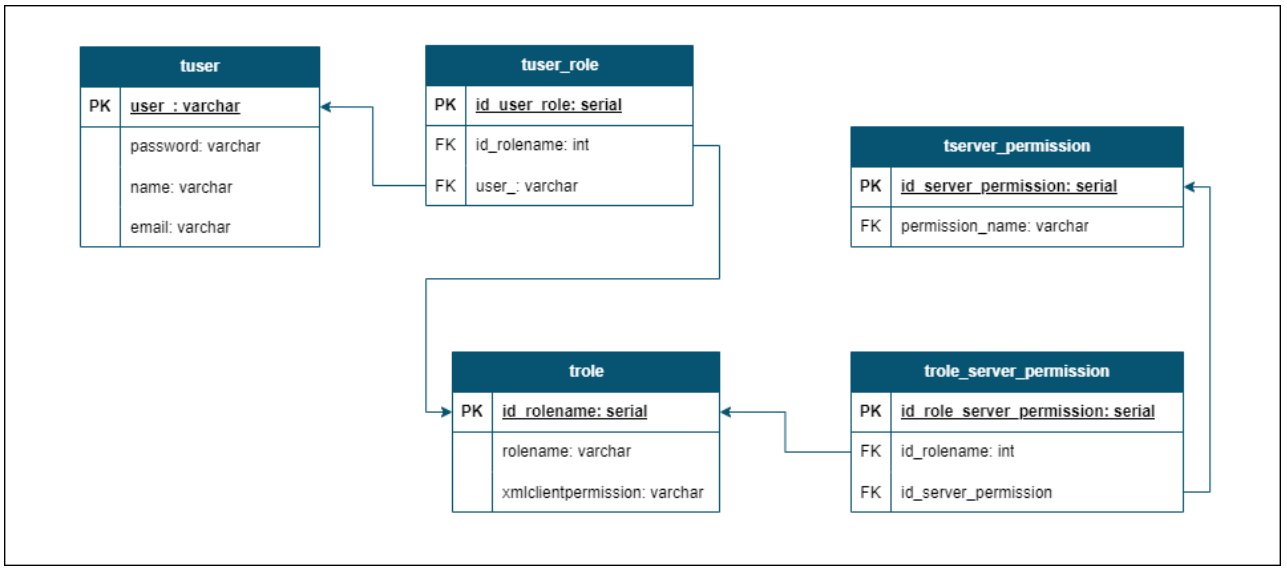

Diagrama4_ER2

Descripción de la estructura de datos:

El destino principal de este modelo de datos es el de representar una suscripción de usuario. Para ello se utiliza la entidad *suscripción*, cuyo campo *subs_autorenewal* indica si se quiere renovar dicha suscripción o no. Debido a que cada suscripción tiene un precio predefinido en la aplicación se ha utilizado el *plan_price_id* para hacer referencia a éste.
Como existía el problema de que cada suscripción tiene una fecha de inicio y de fin y que cada una de éstas dura una franja de tiempo definido por la frecuencia no era suficiente con utilizar la entidad suscripción en el caso de que se quisiera hacer un histórico de gastos en el que cada suscripción pudiera potencialmente cambiar de precio. Para ello existe *sub_lapse*, entidad que se ocupa principalmente de almacenar las fechas asociadas a una suscripción y su precio.
A su vez, se ha utilizado la tabla *user_sub* para contemplar el caso de que una suscripción fuera compartida por varios usuarios. Siendo la entidad *tuser* la responsable de modelar éstos últimos.
Por otra parte se ha modelado también lo relativo a los servicios, siendo la entidad *platform* la que se ocupa de representarlos. Como cada servicio puede llegar ofrecer diferentes planes ha sido necesario definir la tabla *plan* y como a su vez éstos pueden cambiar de precio también se ha creado *plan_price*.
También pertenecen al modelo las entidades *frequency* y *category.* La primera es utilizada para establecer frecuencias predefinidas a los planes, en este caso mensual, trimestral o anual. Por otra parte, *category* sirve para ubicar cada plataforma en su categoría correspondiente y poder llegar a filtrarlos más adelante.

Finalmente la figura *Diagrama3_ER2* define toda la estructura de roles y permisos que tienen los usuarios y sirve para dotar de acceso a cada uno de los roles que existen en la aplicación.

2.4. Implementación

Backend

Para la implementación del backend se ha hecho uso de spring, utilizando la siguiente estructura de funcionamiento:

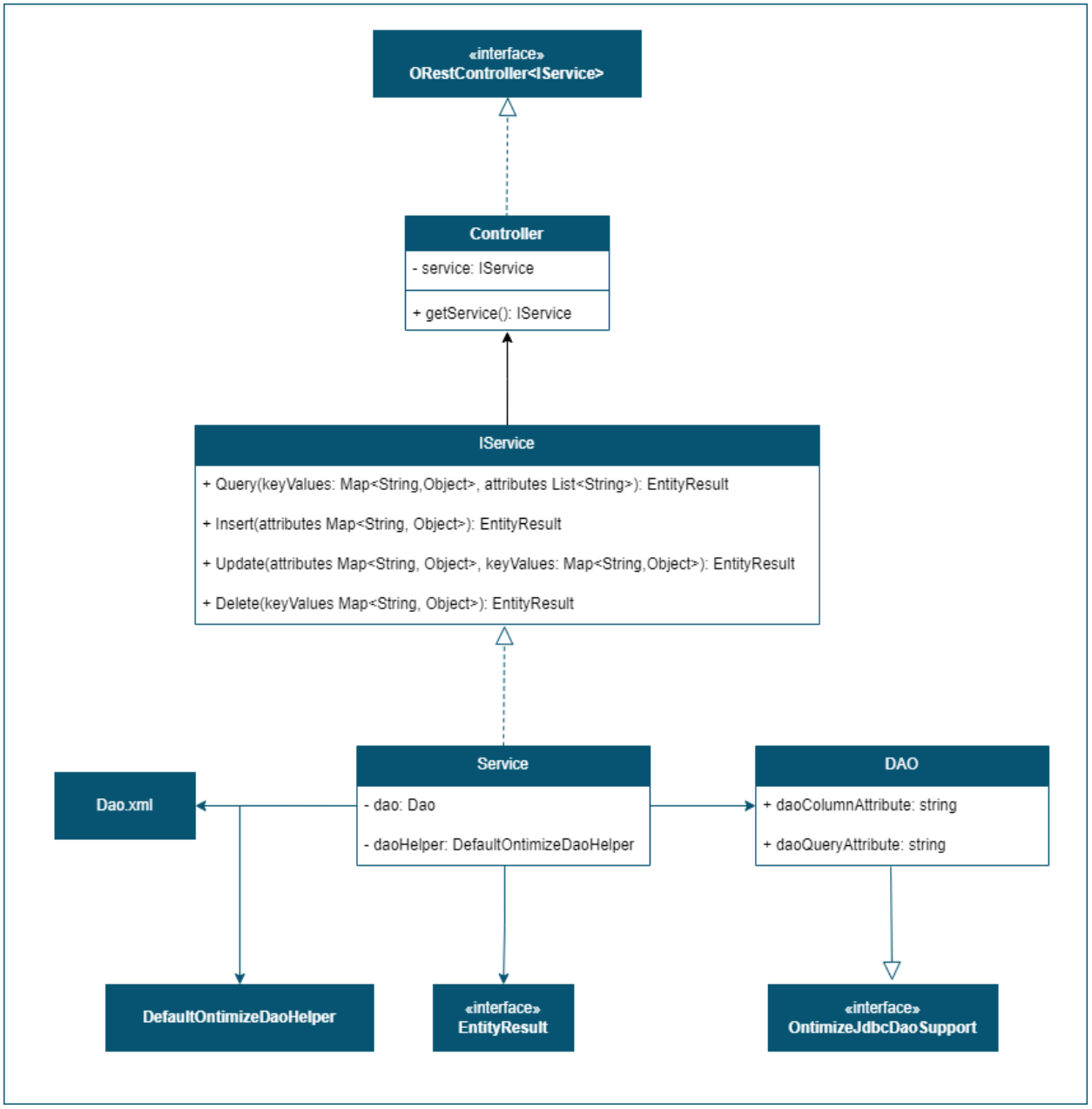

Diagrama5_UML_Class_Backend

Cada una de las entidades de la base de datos contiene su correspondiente definición de clases en el backend siguiendo la estructura de *Diagrama5_UML_Backend*. Es decir, la entidad category tendría su análogo CategoryService, ICategoryService, etc. Esta estructura es necesaria para definir el comportamiento de los métodos CRUD y muchas de las funcionalidades que le dan lógica a la aplicación. En este sentido, la clase encargada de la lógica principal de backend es *Service* mientras que *Dao.xml* se ocupa de definir las consultas realizadas a base de datos. El Controller es el que define el endpoint al que luego el front habrá de llamar para realizar las operaciones.

Cabe destacar que el backend contiene una tarea programada que se ocupa de actualizar las suscripciones del usuario en el caso de que éstas haya que renovarlas. La clase que se ocupa de realizar esto es *ScheduledTask* y contiene los siguientes métodos y atributos:

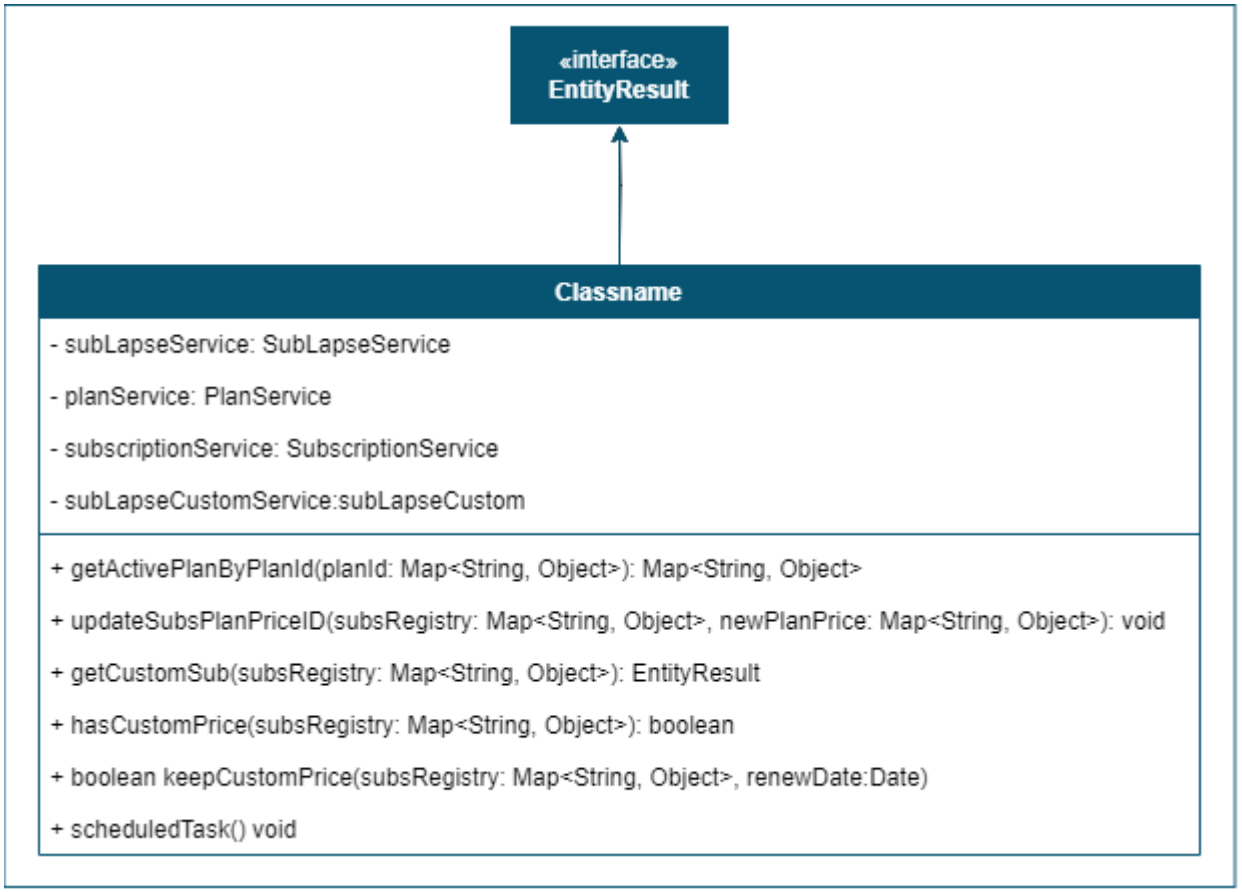

Diagrama6_UML_Class_ScheduledTask

El funcionamiento de este código se expone en el siguiente diagrama de secuencia:

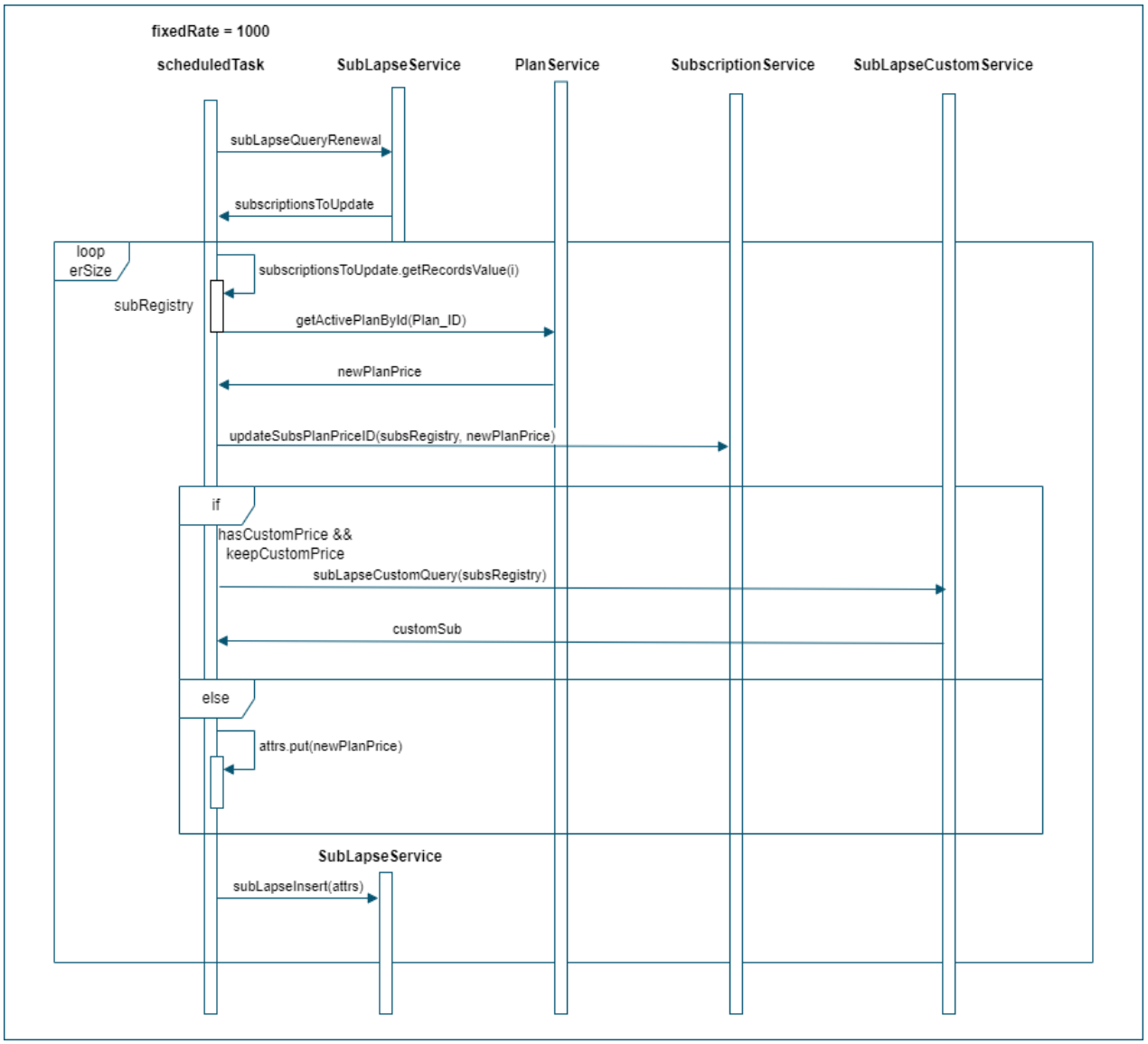

Diagrama7_UML_Secuence_ScheduledTask

Frontend

En el caso del frontend se ha utilizado la siguiente estructura para realizar las peticiones a los servicios ofrecidos en el backend y una vez recibida la respuesta en modo de Json deserializarla y convertirla en un objeto con el que poder trabajar en los formularios o actividades. La definición de las clases asociadas a la petición es la siguiente:

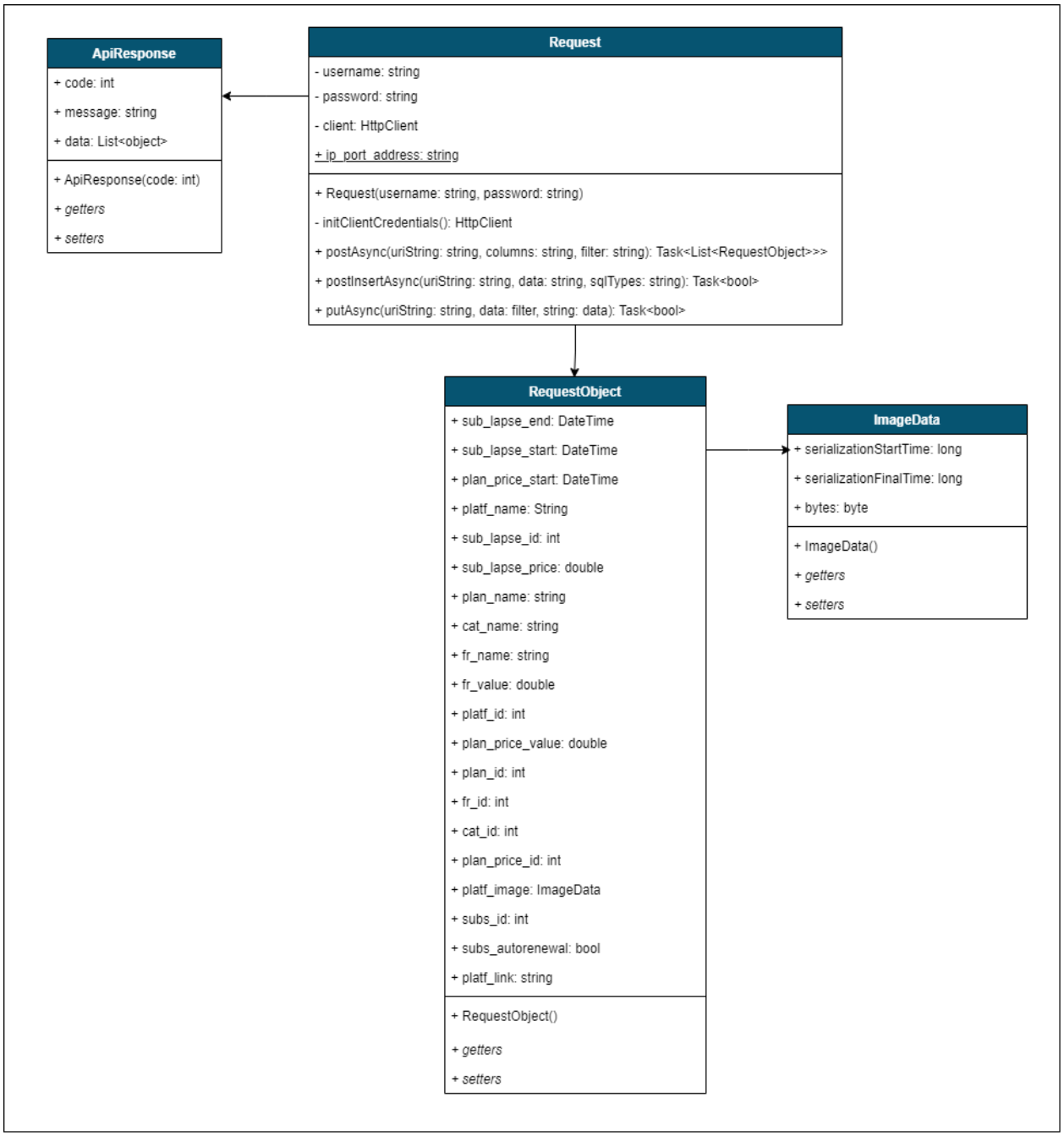

Diagrama8_UML_Class_Request

2.5. Pruebas

A continuación se elaborará una lista de todas las pruebas realizadas sobre la aplicación utilizando como referencia los casos de uso e intentando validar la funcionalidad core de la aplicación:

Escritorio - usuario:

CP1(caso de prueba 1) - **Registro exitoso y login**:

Descripción
Verificar que un usuario pueda registrarse correctamente, acceder al sistema mediante el login, y que la aplicación presente un estado inicial vacío.

Pasos
1. Hacer clic en el enlace de registro.
2. Completar los campos requeridos correctamente para el registro.
3. Verificar que el usuario se genere correctamente en la base de datos.
4. Iniciar sesión usando las credenciales recién registradas.
5. Confirmar que el sistema permite el acceso al usuario.
6. Verificar que la aplicación presente un estado inicial vacío después del inicio de sesión.

Resultados Esperados
Después de completar estos pasos, el usuario debería haberse registrado con éxito, haber iniciado sesión sin problemas, y la aplicación debería mostrar un estado inicial vacío.

CP2 - **Login Fallido**:

Descripción
Verificar que el sistema presente un formulario de error cuando un usuario introduce incorrectamente sus credenciales durante el intento de inicio de sesión.
Pasos
1. Ingresar al sistema mediante el formulario de login.
2. Introducir credenciales incorrectas (por ejemplo, nombre de usuario o contraseña incorrectos).
3. Observar la respuesta del sistema.

Resultados Esperados
Después de introducir las credenciales incorrectas, el sistema debería presentar un formulario de error indicando que el inicio de sesión no fue exitoso.

CP3 - **Añadir suscripción**:

Descripción
Verificar que al completar el formulario de añadir suscripción, el sistema genere correctamente una nueva suscripción para el usuario. Esto debería reflejarse en la base de datos y actualizar tanto el listado de suscripciones como la página de inicio.

Pasos
1. Acceder al formulario de añadir suscripción.
2. Seleccionar todos los campos requeridos en el formulario.
3. Completar los campos con información válida.
4. Enviar el formulario para añadir la suscripción.

Resultados Esperados
Después de seguir estos pasos, el sistema debería:
- Generar una nueva entrada de suscripción en la base de datos asociada al usuario.
- Actualizar el listado de suscripciones para incluir la nueva suscripción.
- Reflejar la nueva suscripción en la página de inicio.

CP4 - **Listar suscripciones**

Descripción
Verificar que el usuario pueda ver un listado actualizado de todas sus suscripciones. Además, se debe confirmar que existe un sumatorio que refleje el gasto mensual total.

Pasos
1. Acceder a la sección de listado de suscripciones.
2. Observar que todas las suscripciones existentes para el usuario estén correctamente listadas.
3. Verificar que el listado se actualiza automáticamente a medida que se añaden nuevas suscripciones.
4. Comprobar la presencia y exactitud del sumatorio del gasto mensual total.

Resultados Esperados
Después de seguir estos pasos, el sistema debería:
- Acceder a la sección de listado de suscripciones desde la interfaz de usuario.
- Observar que todas las suscripciones existentes para el usuario estén correctamente listadas.
- Verificar que el listado se actualiza automáticamente a medida que se añaden nuevas suscripciones.
- Comprobar la presencia y exactitud del sumatorio del gasto mensual total.

CP5 - **Ver detalle de suscripción:**

Descripción

Verificar que al seleccionar una fila en el listado de suscripciones, el usuario pueda ver la información detallada relacionada con esa suscripción.

Pasos

1. Acceder al listado de suscripciones.
2. Seleccionar una fila específica que represente una suscripción.
3. Observar que se presenta la información detallada de esa suscripción.

Resultados Esperados

Después de seguir estos pasos, el sistema debería:

- Permitir al usuario seleccionar una fila en el listado de suscripciones.

CP6 - **Modificar suscripción:**

Descripción

Verificar que el usuario pueda modificar el estado de renovación de una suscripción en la sección de detalles. Al realizar cambios, deberá aparecer un botón de guardar para confirmar la modificación.

Pasos

1. Seleccionar una suscripción específica desde el listado.
2. Acceder a la sección de detalles de esa suscripción.
3. Modificar el estado de renovación según sea necesario.
4. Verificar que aparezca un botón de guardar después de realizar cambios.
5. Hacer clic en el botón de guardar para confirmar la modificación.

Resultados Esperados

Después de seguir estos pasos, el sistema debería:

- Permitir al usuario modificar el estado de renovación de una suscripción.
- Mostrar un botón de guardar después de realizar cambios.
- Confirmar y guardar los cambios cuando el usuario hace clic en el botón correspondiente.

CP7 - **Ver notificaciones:**

Descripción

Verificar que cuando una suscripción no se quiere renovar, aparezca un botón en la tarjeta correspondiente en la pantalla de inicio. Al presionar este botón, deberá redirigir al usuario a la página relativa a esa suscripción para permitirle darse de baja.

Pasos

1. Acceder a la pantalla de inicio.
2. Identificar una suscripción que no se renovará y que esté reflejada en una tarjeta.
3. Verificar que aparezca un botón indicando la opción de darse de baja.
4. Presionar el botón para redirigirse a la página de detalles de esa suscripción.

Resultados Esperados

Después de seguir estos pasos, el sistema debería:

- Permitir al usuario modificar el estado de renovación de una suscripción.
- Mostrar un botón de guardar después de realizar cambios.
- Confirmar y guardar los cambios cuando el usuario hace clic en el botón correspondiente.

CP8 - **Desloguearse:**

Descripción

Verificar que al presionar sobre el botón de cerrar sesión, el usuario salga de su sesión y sea redirigido a la pantalla de inicio de sesión.

Pasos

1. Hacer clic en el botón de cerrar sesión.
2. Confirmar que se cierra la sesión actual del usuario.
3. Verificar que el sistema redirige al usuario a la pantalla de inicio de sesión.

Resultados Esperados

Después de seguir estos pasos, el sistema debería:

- Cerrar la sesión actual del usuario.
- Redirigir al usuario a la pantalla de inicio de sesión.

Escritorio - administrador:

*Nota: Para loguearse como administrador es necesario utilizar las credenciales usuario: Admin y contraseña: demouser

CP9 - **Añadir plataforma y plan:**

Descripción

Verificar que el administrador pueda crear una plataforma y su plan asociado, y que el usuario pueda suscribirse correctamente a este plan.

Pasos

Pasos para el Administrador (Añadir Plataforma y Plan)

1. Acceder a la interfaz de administrador.
2. Abrir el formulario para crear una plataforma.
3. Completar la información requerida para la nueva plataforma.
4. Abrir el formulario para crear un plan y generar uno para la plataforma que se acaba de crear.
5. Confirmar que la plataforma y el plan se han creado correctamente.

Pasos para el Usuario (Suscribirse al Plan)

1. Acceder a la interfaz de usuario.
2. Buscar y seleccionar la plataforma y plan recién creados.
3. Iniciar el proceso de suscripción al plan.
4. Completar los pasos necesarios para la suscripción.
5. Verificar que la suscripción se realiza correctamente.

Resultados Esperados

Después de seguir estos pasos, el sistema debería:

- El administrador debería haber creado una nueva plataforma y su plan asociado.
- El usuario debería haberse suscrito correctamente a este plan.

Móvil:

Las pruebas que se han realizado sobre el entorno móvil son las análogas al CP1 (únicamente login) y al CP4.

3.Conclusiones

El diseño de la estructura de datos ha sido la parte más compleja de todo el desarrollo y ha sido gracias a una implementación iterativa que se ha conseguido facilitar su comprensión y su correcta definición. De esta manera, se ha pulido al máximo la parte básica de la aplicación que son sus entidades y por ello se ha podido desarrollar tanto el back como el front de una manera más sencilla. Pese a esto, al tener tantas tablas interrelacionadas, en ciertos puntos del código puede ser algo más complicado de entender las peticiones debido a la cantidad de columnas que intervienen en ellas. Ha sido crucial el manejo de sql a la hora de definir las consultas y de generar columnas calculadas.

A su vez, haber implementado un backend robusto que se encarga de la mayoría de la lógica de negocio de la aplicación ha encapsulado su comportamiento y facilitado que el frontend se limite a la petición y mostrado de datos. Tareas como la de creación de nuevas suscripciones delegan toda la responsabilidad en el servicio del backend simplificando el código en el front.

Ha sido este desarrollo intensivo del backend como de la estructura lo que ha supuesto que no se hayan implementado todas las funcionalidades que se hubieran deseado. Aunque

por otra parte, funcionalidades como la de la tarea programada han enriquecido el código y, por lo tanto, conllevado a que el backend esté más completo.
Por otra parte, se han desarrollado en escritorio la mayoría de funcionalidades posibles lo que ha conllevado a desequilibrar la cantidad de funciones que tiene en comparación con la versión de Android. De esta manera se ha preferido tener una versión de escritorio con una funcionalidad más elaborada en vez de limitar la misma y obtener una versión de android algo más desarrollada pero más igualada en complejidad con el escritorio.
Se ha priorizado la implementación de escritorio debido a la existencia de dos roles de usuario y no se ha visto tan lógico que el administrador utilice la aplicación en entorno móvil. Por esta razón, se ha decidido condensar toda la funcionalidad en el escritorio. Es cierto que la limitación de tiempo también ha llevado a una planificación poco realista sobre las horas dedicadas a cada plataforma lo que ha influido en este desbalanceo entre ambas versiones.
También se ha intentado proponer una interacción usuario máquina lo más intuitiva posible facilitando el uso de la aplicación y su entendimiento. La intención de las pantallas es la de ser lo más limpias posibles y dirigir la atención del usuario hacia la información que necesita en cada momento.
Finalmente, vuelvo a destacar la importancia del diseño de una estructura de datos para evitar la pérdida o la redundancia de los mismos, al igual que el desarrollo iterativo para fomentar el avance en el proyecto y evitar posibles bloqueos.

Por último agradecer a mis compañeros de equipo de la bootcamp en Imatia (en especial a Noa), por la ayuda y el apoyo en el desarrollo de la versión web. No habría sido posible utilizar un backend tan elaborado para este proyecto sin su ayuda y apoyo. Agradecer también a los tutores (especialmente a Álvaro) y mentores de Imatia por la guía en el uso de Ontimize y el seguimiento en todo el proyecto. También agradecer a los profesores y tutores del colegio Montecastelo por esta oportunidad de crecimiento y formación que tanto me ayudará en mi futuro profesional.

4. Líneas Futuras

Como comentado en el anterior apartado es urgente y necesario implementar las funcionalidades de usuario en la versión móvil.
Otras de las funcionalidades que no se han implementado por falta de tiempo:

- Histórico de gastos: el usuario podrá ver un histórico de sus suscripciones pasadas para saber cuánto ha gastado a lo largo del tiempo.
- Precios personalizados: esta funcionalidad permitiría al usuario agregar períodos de prueba gratuítos a las suscripciones así como promociones.
- Cambio de precio del plan: en el caso de que una plataforma suba de precio el administrador podrá agregar un nuevo precio al plan que entrará en vigor a partir de la fecha seleccionada.

Otras de las propuestas a futuro de la aplicación:

- Tips de ahorro: la aplicación propondrá al usuario diferentes precios que ofrece una plataforma y comparaciones con otras de la misma categoría y así dar oportunidad a reducir sus gastos.
- Número de horas por servicio: la aplicación tendrá datos sobre la cantidad de horas que se ha utilizado un servicio para saber si ha valido la pena contratarlo.
- Noticias en servicios: la aplicación ofrecerá noticias sobre las últimas novedades en cada plataforma.
- Internacionalización: la aplicación contendrá precios específicos por ubicación.
- Comunicación: la aplicación dará la opción de agregar amistades y comunicarse para compartir suscripciones.

5. Bibliografía

https://ontimizeweb.github.io/docs/v8/
https://developer.android.com/docs?hl=es-419
https://learn.microsoft.com/en-us/dotnet/desktop/winforms/?view=netdesktop-8.0

6. Anexo

Los esquemas utilizados en la memoria están contenidos en la carpeta del proyecto “esquemas”