

ANEXO IV: Diseño del sistema software

Plataforma de monitorización remota para la atención domiciliar

Trabajo de Fin de Grado

Ingeniería Informática



**VNiVERSiDAD
D SALAMANCA**

Julio de 2023

Autor

Germán Francés Tostado

Tutor/a

David Cruz García

Gabriel Villarrubia González

Tabla de contenidos

1. Introducción.....	4
2. Modelo de diseño.....	5
2.1 Patrones arquitectónicos.....	5
2.1.1 API Gateway.....	5
2.1.2 MVC.....	6
2.1.3 Middleware.....	6
2.1.4 Inheritance.....	7
2.2 Subsistemas de diseño.....	8
2.3 Clases de diseño.....	10
2.3.1 Frontend.....	10
2.3.2 Backend.....	11
2.3.3 Tablet.....	12
2.4 Vista arquitectónica.....	13
2.5 Realización de casos de uso.....	14
2.5.1 Paquete Gestión de Usuarios.....	14
2.5.2 Paquete Gestión de Pacientes.....	20
2.5.3 Paquete Gestión de Médicos.....	23
2.5.4 Paquete Gestión de Plataforma Web.....	26
2.5.5 Paquete Gestión de Tablet.....	29
3. Diseño de la base de datos.....	31
4. Modelo de despliegue.....	32
5. Referencias.....	33

Índice de figuras

Figura 1: Patrón API Gateway.....	5
Figura 2: Patrón MVC.....	6
Figura 3: Patrón Middleware.....	6
Figura 4: Patrón Inheritance.....	7
Figura 5: Subsistema de diseño.....	9
Figura 6: Subpaquete Páginas.....	10
Figura 7: Subpaquete Servicios.....	10
Figura 8: Subpaquete Datos médicos.....	11
Figura 9: Subpaquete Controlador API.....	12
Figura 10: Subpaquete Layout.....	12
Figura 11: Subpaquete ControladorAPITablet.....	13
Figura 12: Vista arquitectónica.....	14
Figura 13: Diagrama de secuencia Registro.....	14
Figura 14: Diagrama de secuencia Login.....	15
Figura 15: Diagrama de secuencia Recuperar Contraseña.....	16
Figura 16: Diagrama de secuencia Enviar token de recuperación.....	16
Figura 17: Diagrama de secuencia Cambiar Contraseña.....	17
Figura 18: Diagrama de secuencia Logout.....	17
Figura 19: Diagrama de secuencia Modificar datos perfil.....	18
Figura 20: Diagrama de secuencia Consultar info perfil.....	18
Figura 21: Diagrama de secuencia de Eliminar Cuenta.....	19
Figura 22: Diagrama de secuencia Ver pacientes.....	20
Figura 23: Diagrama de secuencia Buscar Pacientes.....	20
Figura 24: Diagrama de secuencia Ver historiales médicos.....	21
Figura 25: Diagrama de secuencia Descargar historiales médicos.....	21
Figura 26: Diagrama de secuencia Subir historiales médicos.....	22
Figura 27: Diagrama de secuencia Buscar historiales.....	22
Figura 28: Diagrama de secuencia Añadir Médico.....	23
Figura 29: Diagrama de secuencia Eliminar Médico.....	24
Figura 30: Diagrama de secuencia Ver médicos.....	25
Figura 31: Diagrama de secuencia Buscar médicos.....	25
Figura 32: Diagrama de secuencia Consultar estadísticas personales.....	26
Figura 33: Diagrama de secuencia Consultar estadísticas generales.....	26
Figura 34: Diagrama de secuencia Acceder a sala de consultas.....	27
Figura 35: Diagrama de secuencia Actualizar datos.....	27
Figura 36: Diagrama de secuencia Crear datos.....	28
Figura 37: Diagrama de secuencia de Leer sensores.....	29
Figura 38: Diagrama de secuencia Enviar datos a plataforma web.....	30
Figura 39: Diagrama de secuencia Acceder a sala de consultas desde tablet.....	30
Figura 40: Diagrama Entidad-Relación de la base de datos.....	31

Figura 41: Diagrama de despliegue.....32

1.Introducción

En los anexos anteriores, se ha estudiado el dominio del problema recogiendo todos los requisitos software del sistema a desarrollar, dejando reflejada su funcionalidad y posteriormente, tuvo lugar la fase de análisis de estos requisitos, basándose en el modelo del dominio y creando una vista arquitectónica.

Este anexo, se centrará en el dominio de la solución, aproximando más la aplicación a lo que sería una implementación real del sistema. Para ello, se recogerá la documentación pertinente al diseño del mismo, dividida en los siguientes apartados:

- **Modelo de diseño:** consistirá en:
 - ◆ **Patrones arquitectónicos:** se detallan los patrones arquitectónicos utilizados durante el diseño.
 - ◆ **Subsistemas de diseño:** se organizará el sistema en diferentes paquetes y su relación.
 - ◆ **Clases de diseño:** se mostrarán las clases de diseño explicando sus operaciones, atributos y relaciones en base a la fase de análisis.
 - ◆ **Vista arquitectónica:** vista por capas según el patrón de diseño.
 - ◆ **Realización de casos de uso:** se analizarán los diagramas de secuencia para casos de uso del sistema donde se verá la comunicación entre los objetos
- **Diseño de la base de datos:** diagrama entidad-relación para la información que debe almacenar el sistema.
- **Modelo de despliegue:** se exhibe la disposición del sistema completo, revelando los nodos y elementos participantes que están involucrados en la distribución.

2. Modelo de diseño

Durante la fase de diseño se ha decidido utilizar diferentes patrones que facilitarán el desarrollo del sistema, de manera que se vaya aproximando a una implementación real del mismo. Se usará el patrón Modelo-Vista-Controlador (MVC) para la arquitectura del sistema, tal y como se vió en el anexo anterior.

Para el desarrollo del modelo se hará uso de diferentes tecnologías:

- **Node.js:** entorno de desarrollo para los servidores backend y frontend, además de permitir conexiones WebRTC.
- **Express.js:** framework web de código abierto para el desarrollo de APIs en Node.js. Se empleará para el desarrollo de una API REST en el servidor backend.
- **React:** librería de JavaScript de código abierto diseñada para crear interfaces de usuario, facilitando el desarrollo.
- **Java:** siendo la tablet un dispositivo android, se creará una aplicación Java para ella mediante Android Studio.

2.1 Patrones arquitectónicos

2.1.1 API Gateway

El patrón API Gateway es el patrón recomendado para aplicaciones basadas en microservicios con múltiples aplicaciones cliente. Es similar al patrón Facade del diseño orientado a objetos, por lo que proporciona un único punto de entrada a las APIs, encapsulando la arquitectura subyacente del sistema.

El patrón redirigirá las peticiones a la ruta que corresponda, tal que proporciona un único punto final para las aplicaciones de los clientes que necesiten comunicarse con los microservicios.

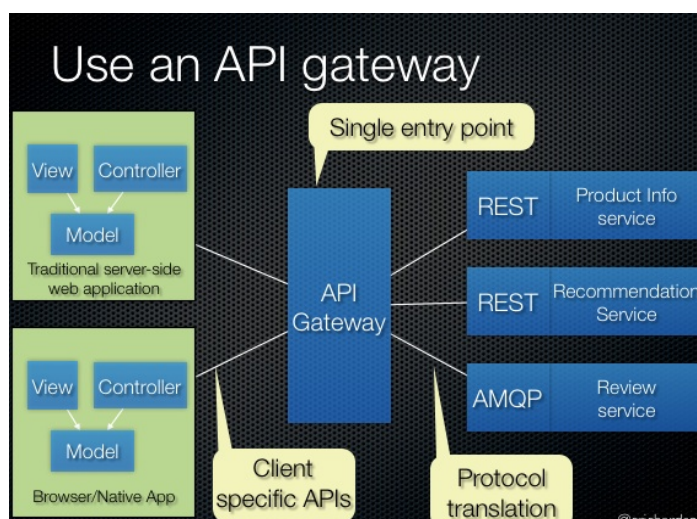


Figura 1: Patrón API Gateway

2.1.2 MVC

El patrón MVC (Modelo-Vista-Controlador) es un patrón de arquitectura software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control. Esta arquitectura propone la separación de los componentes en tres grupos.

- **Modelo:** será la representación de los datos del sistema y la lógica de negocio.
- **Vista:** será la representación visual de los datos contenidos en el modelo al usuario.
- **Controlador:** Es el encargado de comunicar la vista con el modelo, gestionando el flujo de información.

Esta arquitectura será la utilizada en los clientes, como se puede inferir en la figura 1.

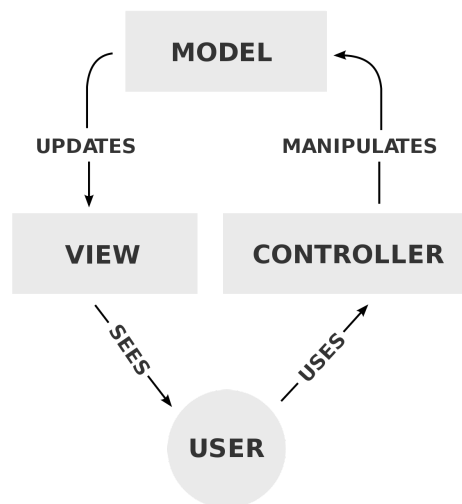


Figura 2: Patrón MVC

2.1.3 Middleware

El patrón Middleware (o Mediator) es el patrón que será utilizado para manejar la comunicación entre los componentes, especialmente las API del sistema. Hace posible que los componentes interactúen entre sí a través de un punto central: el mediador. En lugar de hablar directamente entre sí, el mediador recibe las peticiones y las envía.

En el caso del sistema, cuando llegue una petición, si necesita ser validada (por ejemplo comprobar que el usuario exista en un login), el mediador se encargará de redirigir a la función comprobadora, y si es correcta, continuar con la petición.

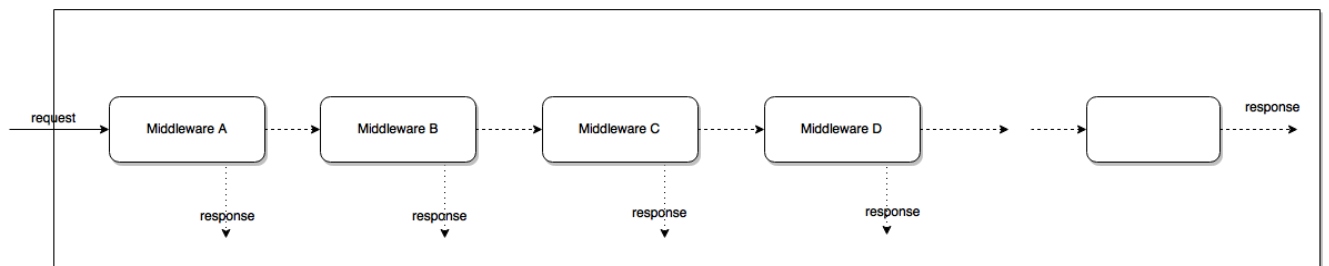


Figura 3: Patrón Middleware

2.1.4 Inheritance

El patrón de diseño Inheritance (herencia) permitirá reducir el número de endpoints del sistema, estableciendo los métodos de las rutas de los endpoint bajo un subtipo, de manera que queden ordenados bajo una jerarquía.

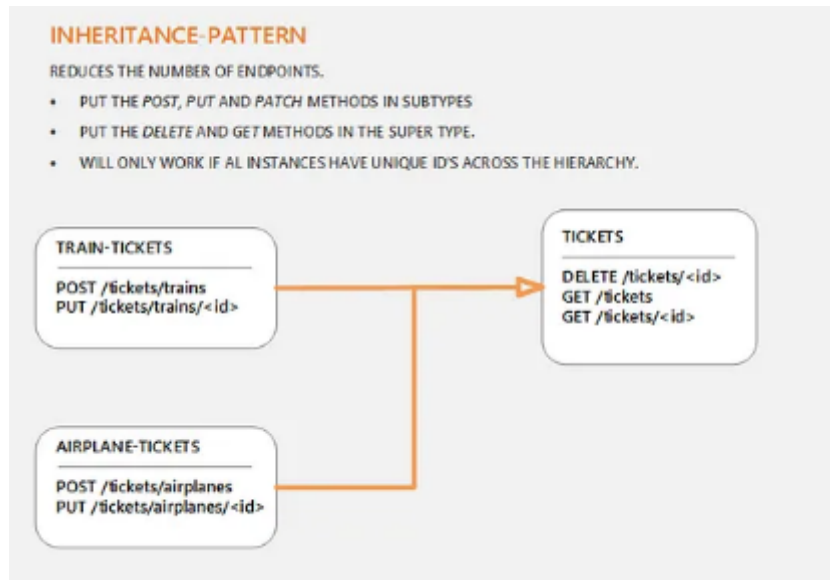


Figura 4: Patrón Inheritance

2.2 Subsistemas de diseño

Se realizará una división del proyecto en subpaquetes más pequeños y más específicos dentro de la estructura general de la aplicación:

- **Plataforma web:** Este subpaquete hará referencia a los servidores desarrollados para la aplicación, separando el frontend del backend en dos paquetes.
 - ◆ **Frontend:** Servidor que proporcionará la web con una interfaz gráfica a los usuarios, de manera que podrán interactuar con ella para identificarse, ver datos etc. Será la Vista dentro del patrón MVC de la aplicación. A su vez estará dividido en dos subpaquetes.
 - **Páginas:** Hará referencia a las diferentes interfaces web que verá el usuario.
 - **Servicios:** Hará referencia al objeto encargado de comunicarse con el backend mediante peticiones a la API, enviando los datos recibidos en las Páginas.
 - ◆ **Backend:** Servidor que proporcionará la API con la que interactuará el resto del programa. Procesará todas las peticiones recibidas y gestionará los datos del sistema. A su vez estará dividido en dos subpaquetes.
 - **DatosMedicos:** Guardará los datos del sistema, tanto la base de datos con lo necesario, como los historiales médicos y los últimos datos medidos de cada paciente. Será el Modelo dentro del patrón MVC de la aplicación.
 - **ControladorAPI:** Recogerá la lógica necesaria para ofrecer los microservicios del sistema, seguirá el patrón API Gateway para ofrecer un punto unificado, que estará dividido mediante el patrón Inheritance para que sean endpoints intuitivos. Al llegar a cada endpoint hará uso del patrón middleware para que las peticiones lleguen de manera correcta. Será el Controlador dentro del patrón MVC de la aplicación.
- **Tablet:** Este subpaquete hará referencia a la aplicación android desarrollada exclusivamente para la tablet, se comunicará con la plataforma web mediante peticiones a la API REST.
 - ◆ **Layout:** Será la interfaz que ve el usuario de la aplicación
 - ◆ **ControladorAPITablet:** Actualizará los datos de los sensores y los enviará a la Plataforma Web mediante peticiones a la API REST.

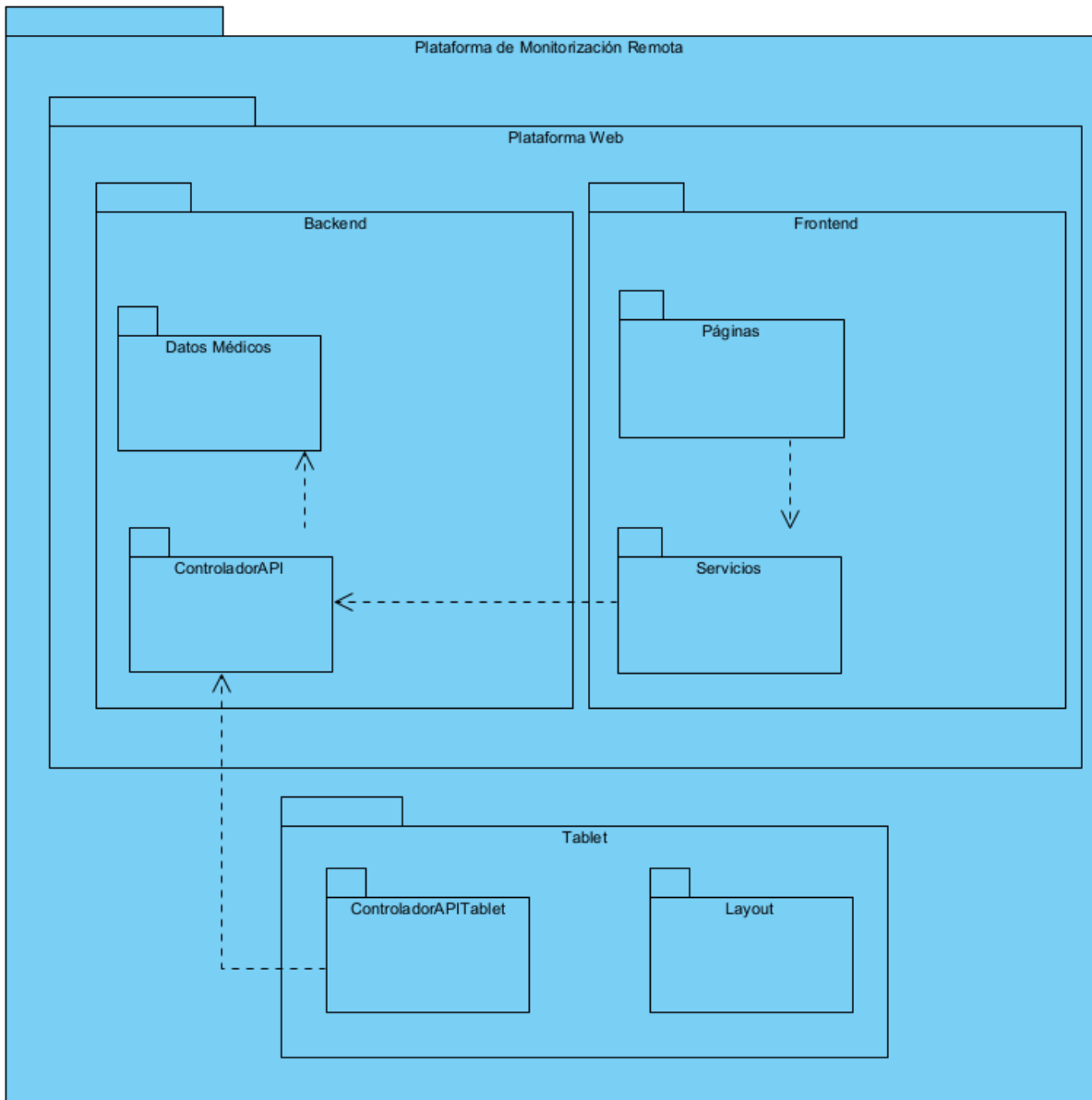


Figura 5: Subsistema de diseño

2.3 Clases de diseño

En este apartado se especificarán los contenidos del sistema, las clases de cada paquete y los métodos de cada clase.

2.3.1 Frontend

- **Páginas:** Se encargará de proporcionar las vistas web y las interfaces para los usuarios.

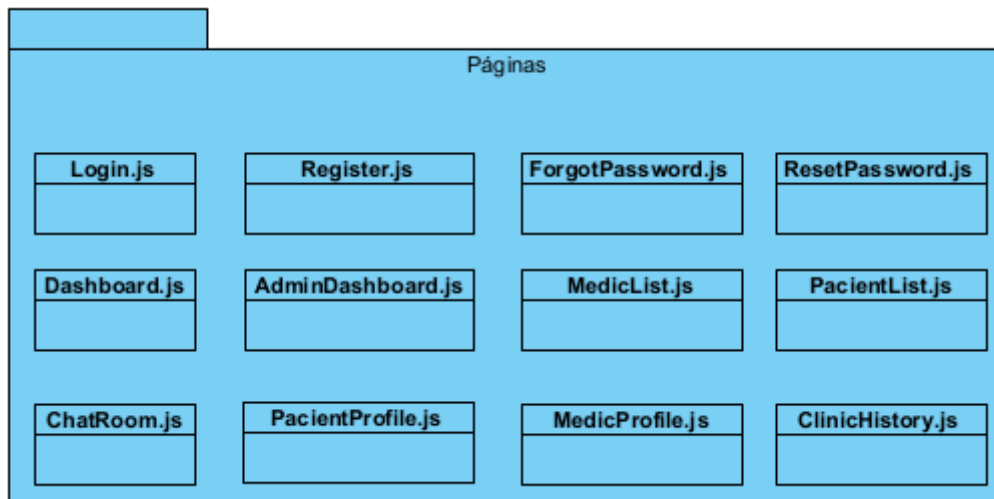


Figura 6: Subpaquete Páginas

- **Servicios:** Hará referencia al objeto encargado de comunicarse con el backend mediante peticiones a la API, enviando los datos recibidos en las Páginas.

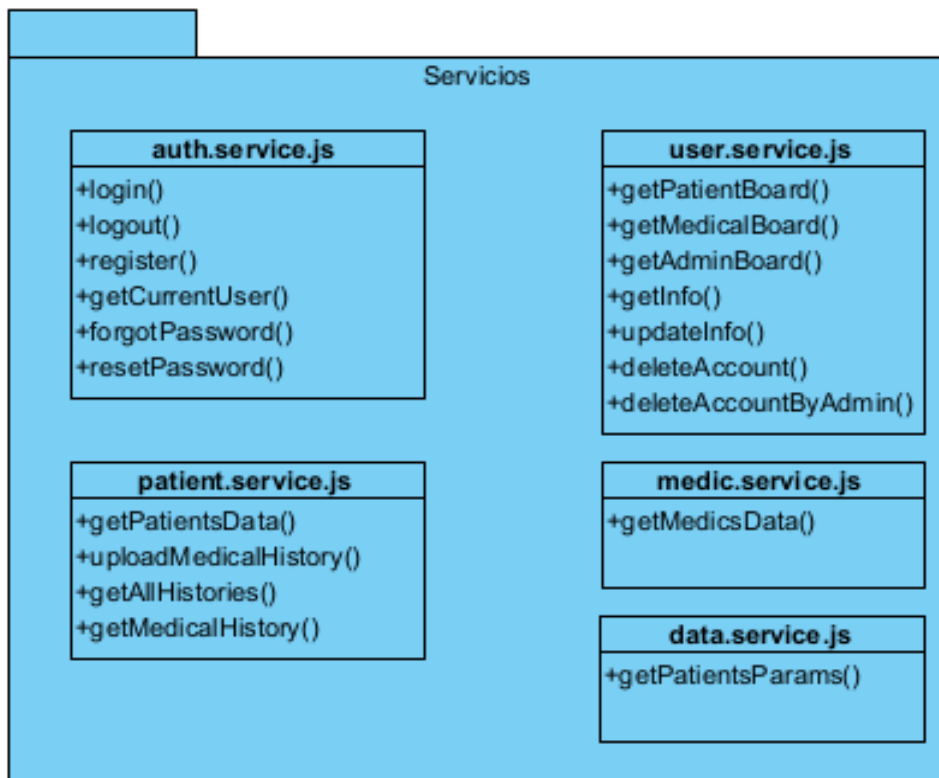


Figura 7: Subpaquete Servicios

2.3.2 Backend

- **Datos Médicos:** Guardará los datos del sistema, tanto la base de datos con lo necesario, como los historiales médicos y los últimos datos medidos de cada paciente.

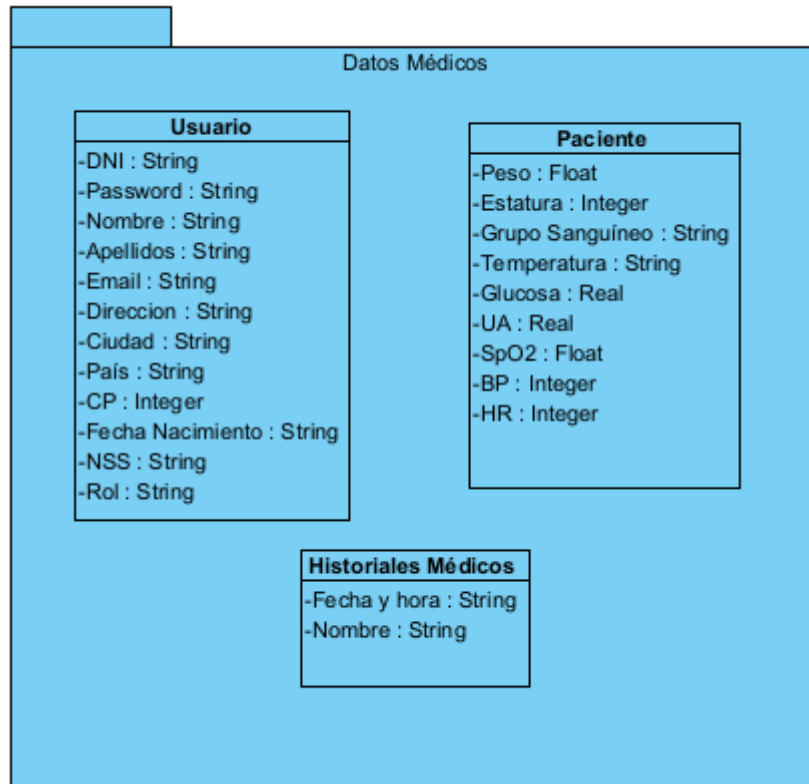


Figura 8: Subpaquete Datos médicos

- **ControladorAPI:** Controlador encargado de la API que recibe peticiones y realizar las acciones necesarias sobre los datos

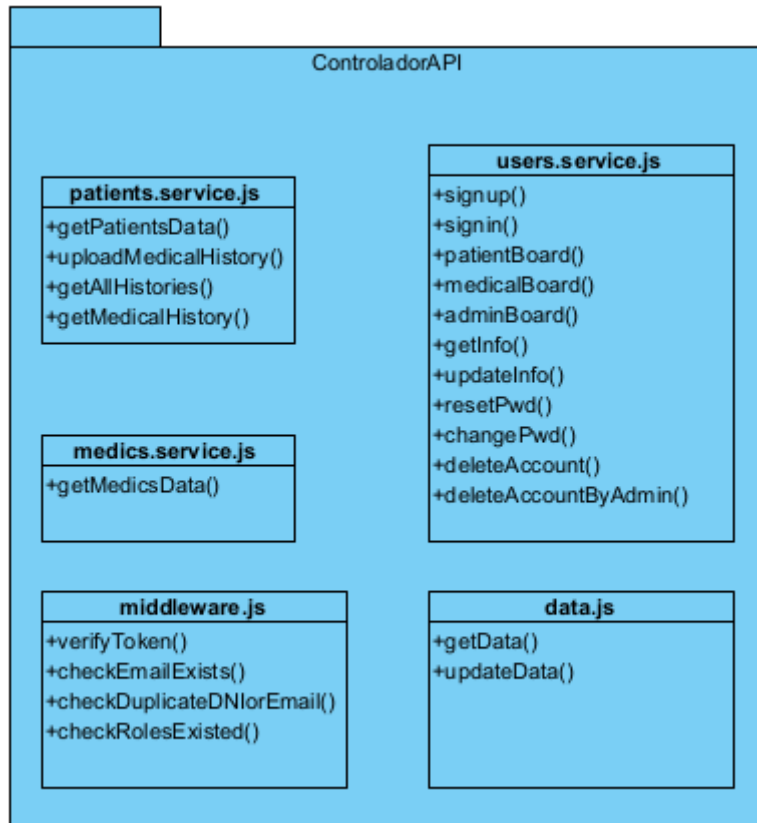


Figura 9: Subpaquete Controlador API

2.3.3 Tablet

- **Layout:** Será la interfaz que ve el usuario de la aplicación

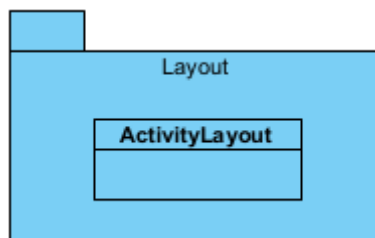


Figura 10: Subpaquete Layout

- **ControladorAPITablet:** Actualizará los datos de los sensores y los enviará a la Plataforma Web mediante peticiones a la API REST.

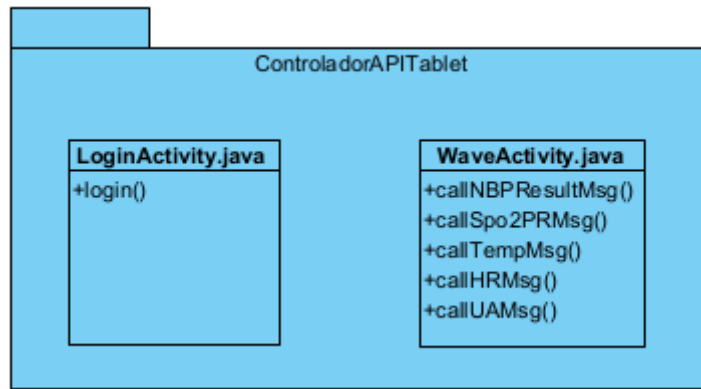


Figura 11: Subpaquete ControladorAPITablet

2.4 Vista arquitectónica

En el apartado, se mostrará la vista completa de la arquitectura inicial que se ha obtenido, de acuerdo al patrón arquitectónico Modelo-Vista-Controlador (MVC).

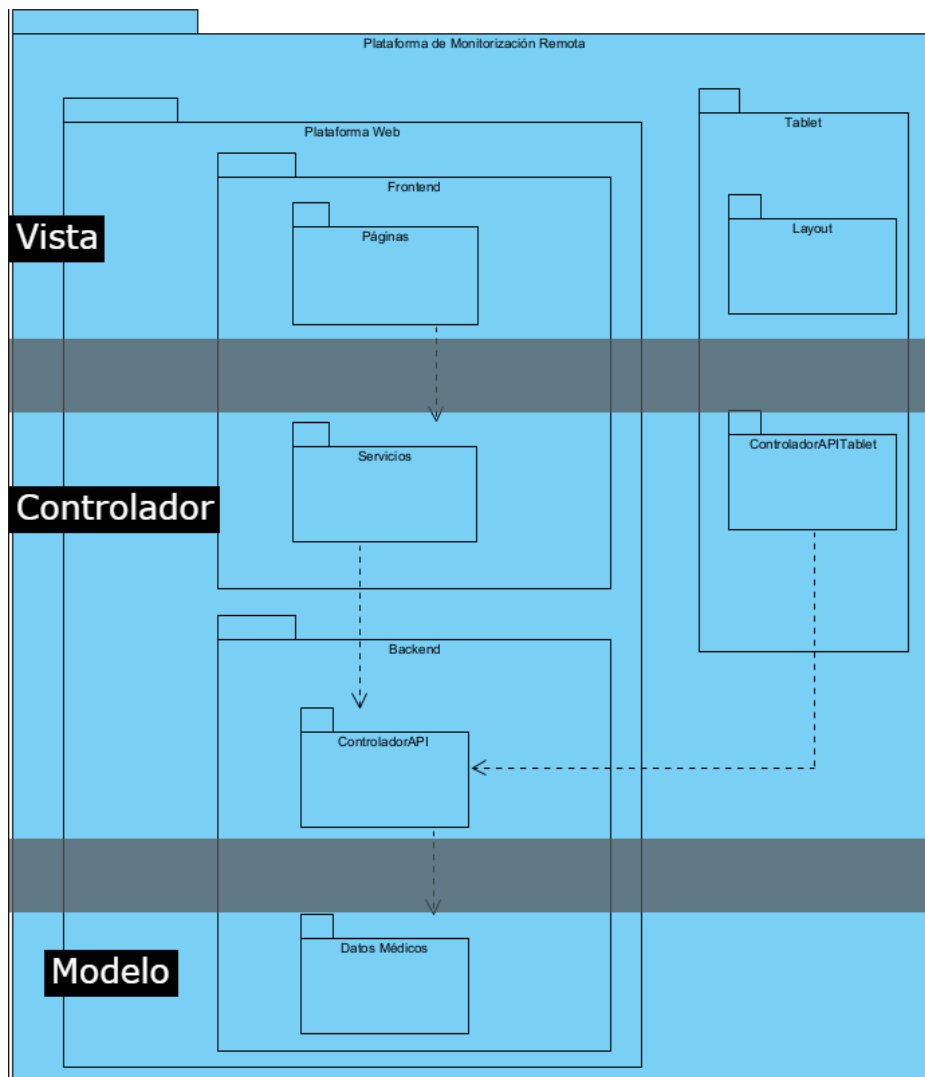


Figura 12: Vista arquitectónica

2.5 Realización de casos de uso

2.5.1 Paquete Gestión de Usuarios

- UC-001: Registro

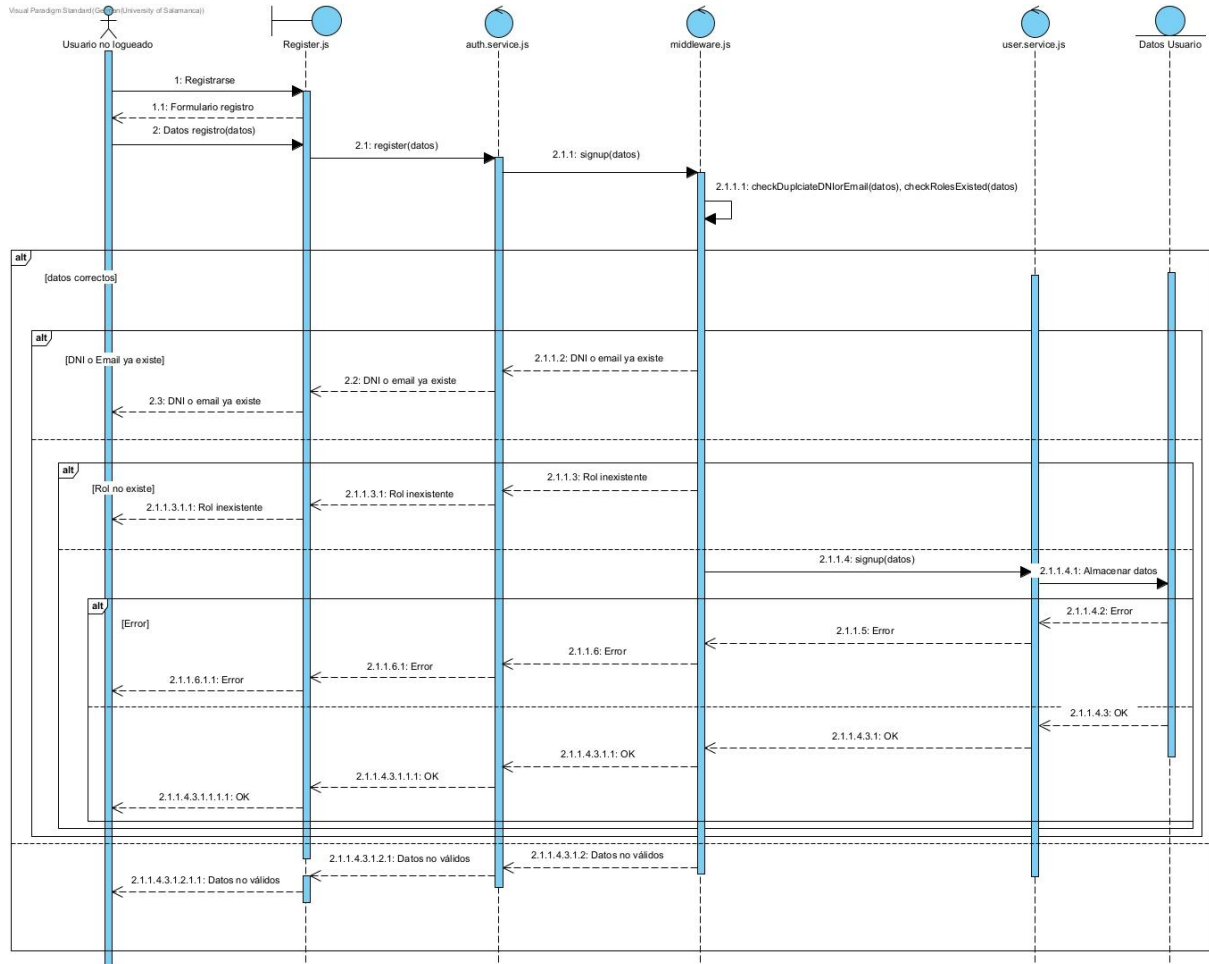


Figura 13: Diagrama de secuencia Registro

- UC-002 Login

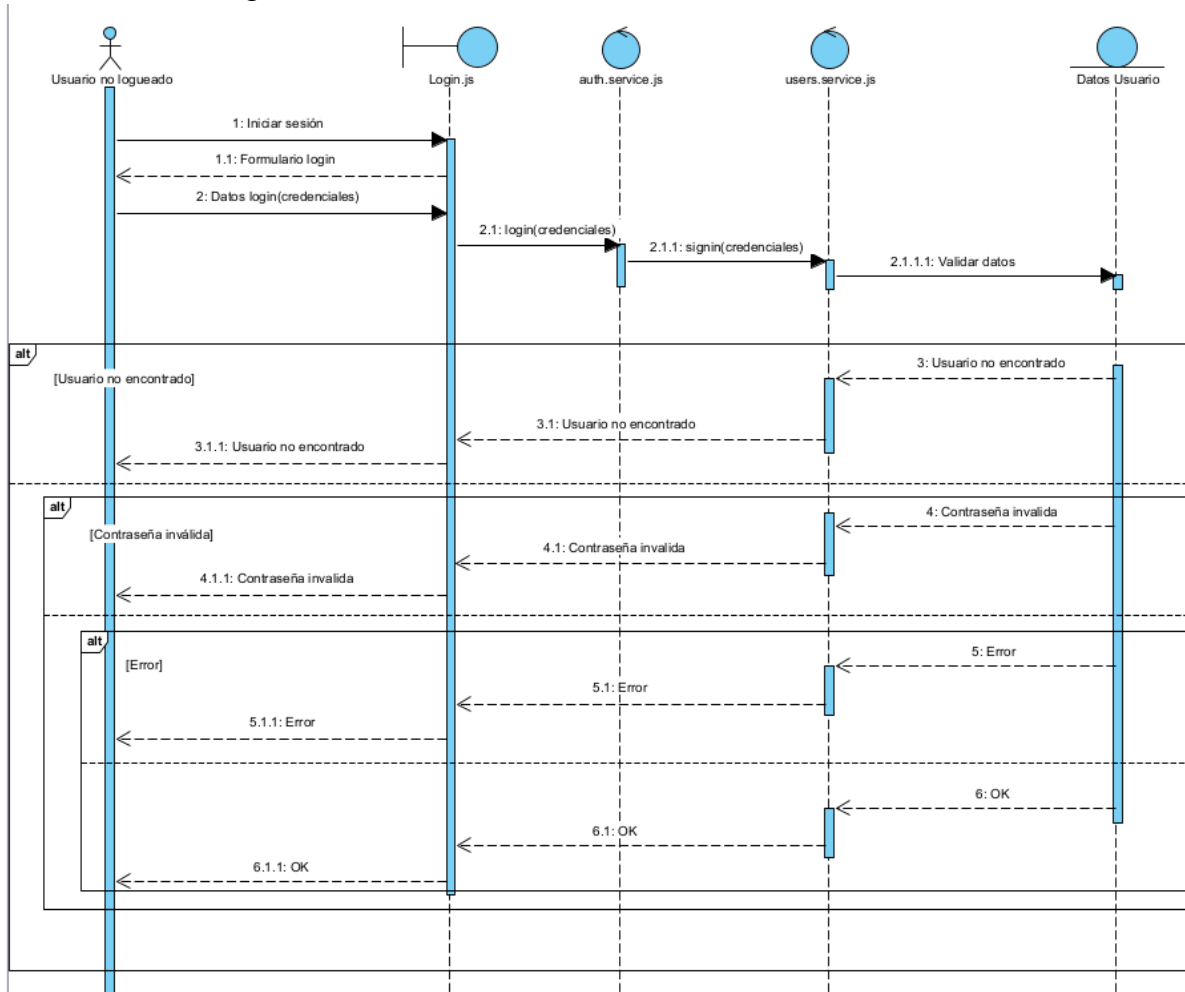


Figura 14: Diagrama de secuencia Login

- UC-003 Recuperar contraseña

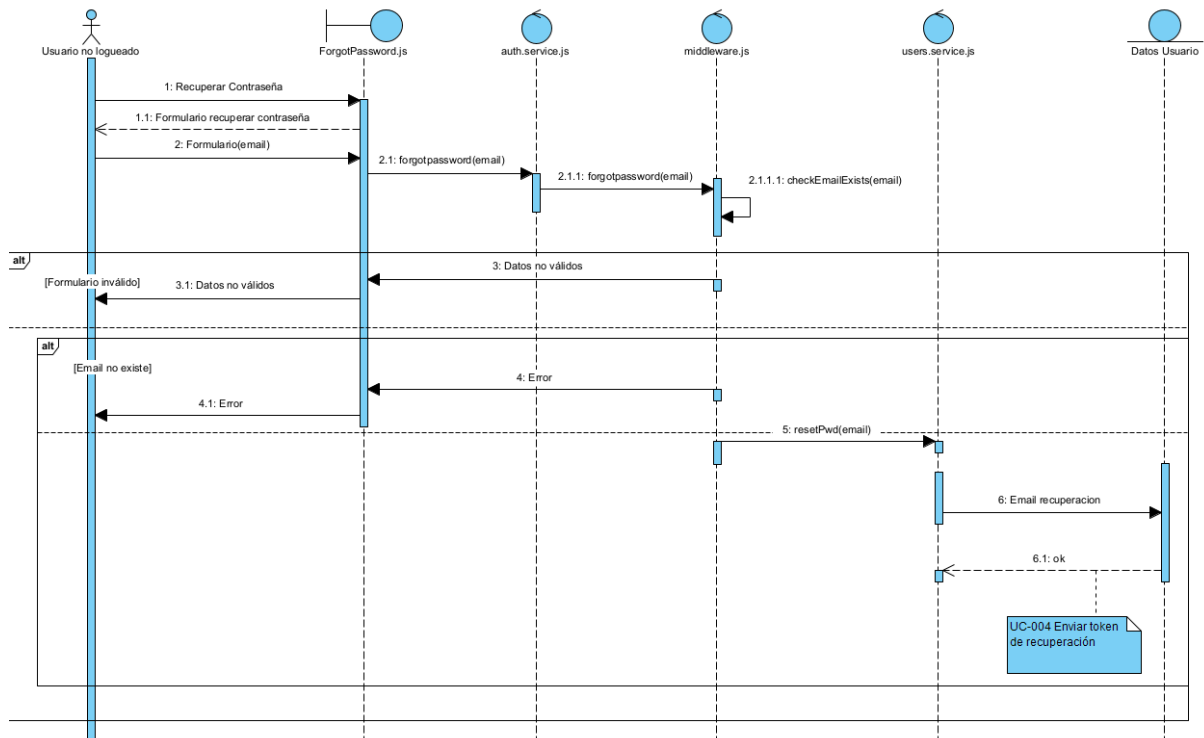


Figura 15: Diagrama de secuencia Recuperar Contraseña

- UC-004 Enviar token de recuperación

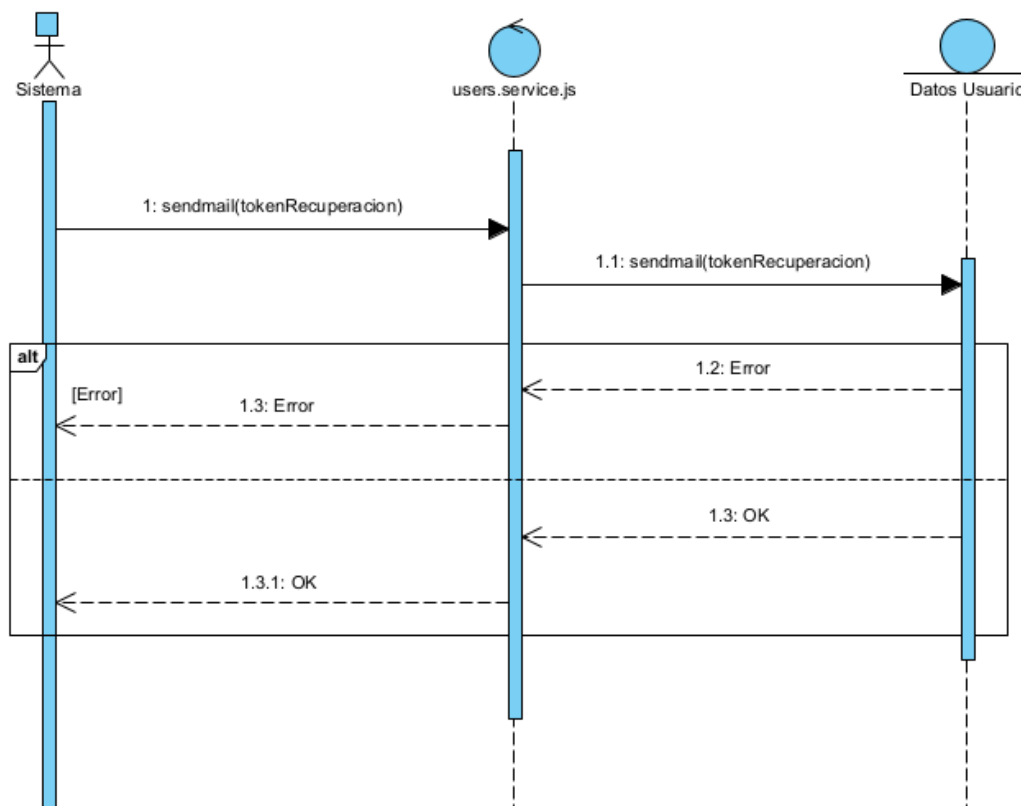


Figura 16: Diagrama de secuencia Enviar token de recuperación

- UC-005 Cambiar Contraseña

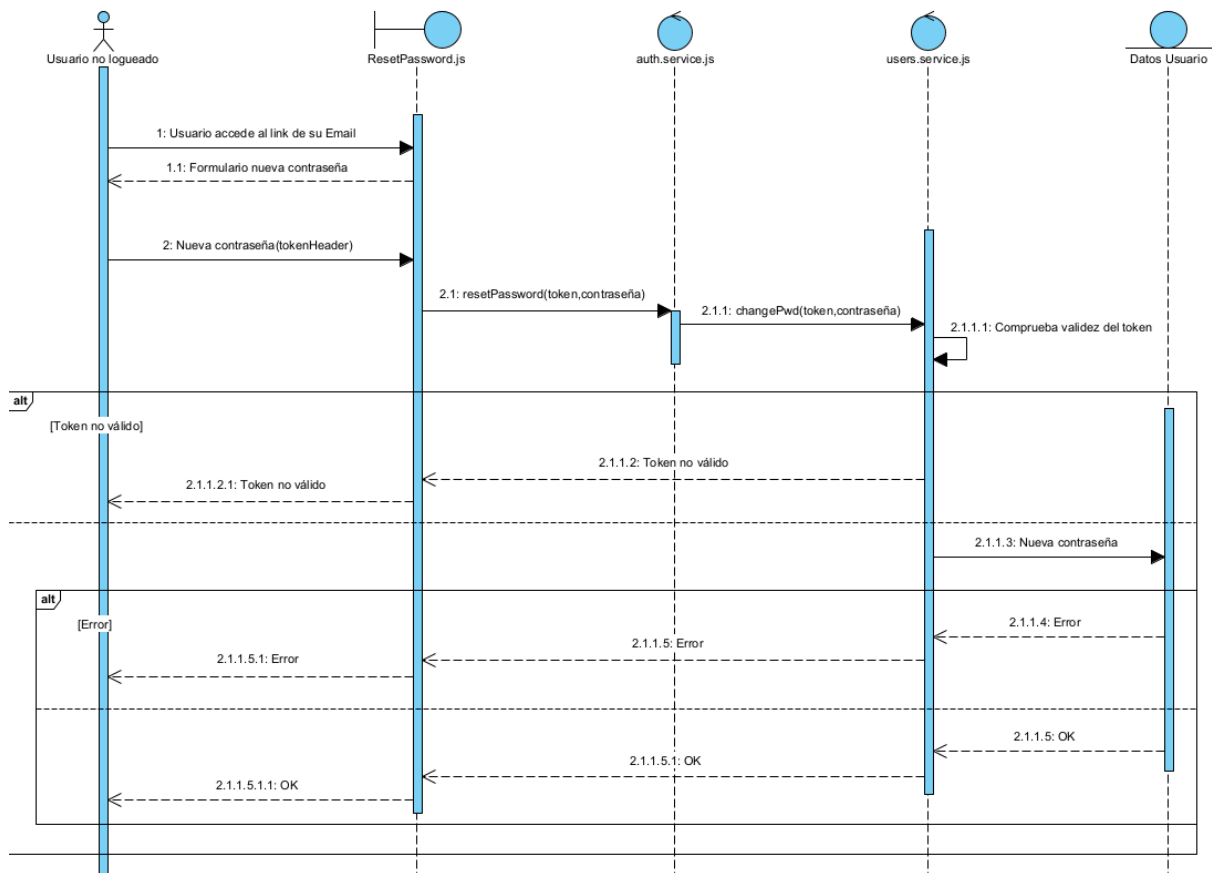


Figura 17: Diagrama de secuencia Cambiar Contraseña

- UC-006 Logout

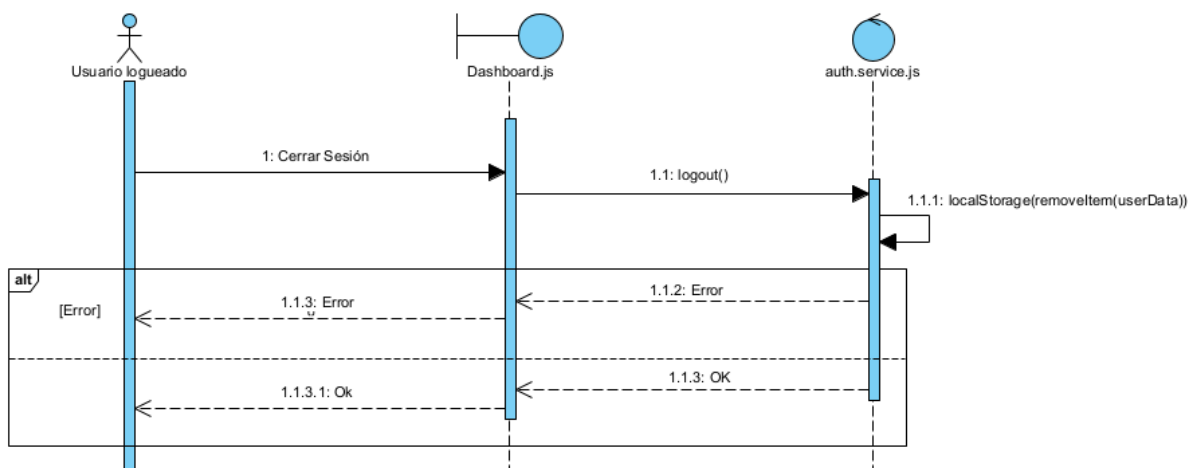


Figura 18: Diagrama de secuencia Logout

- UC-007 Modificar datos perfil

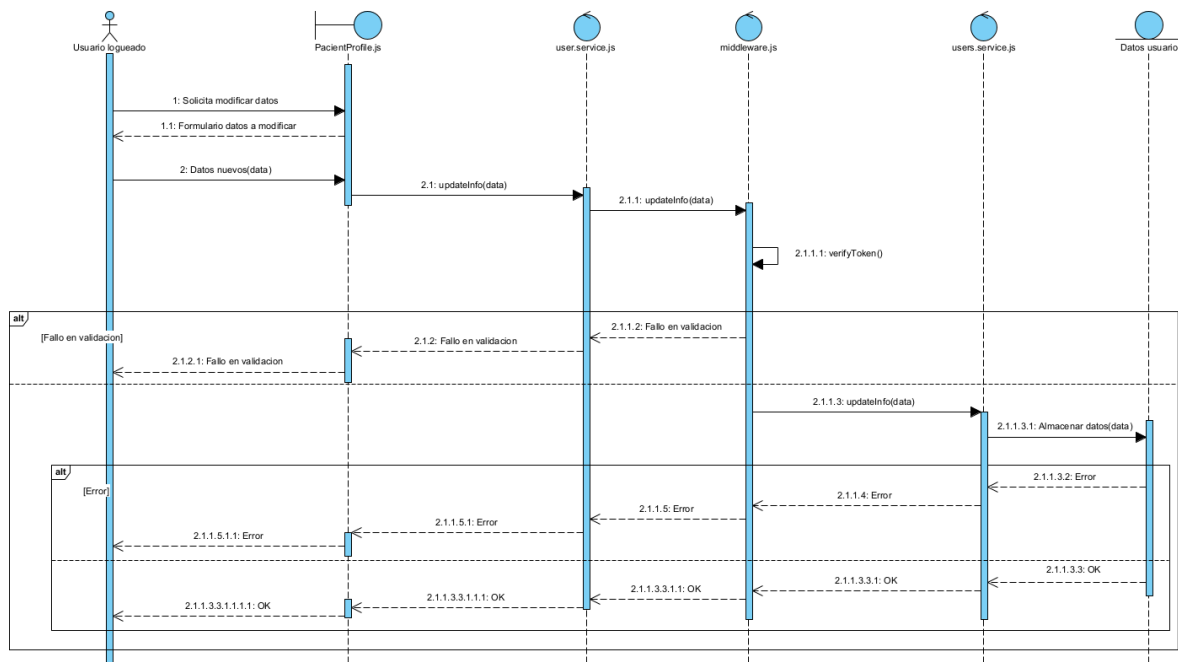


Figura 19: Diagrama de secuencia Modificar datos perfil

- UC-008 Consultar info perfil

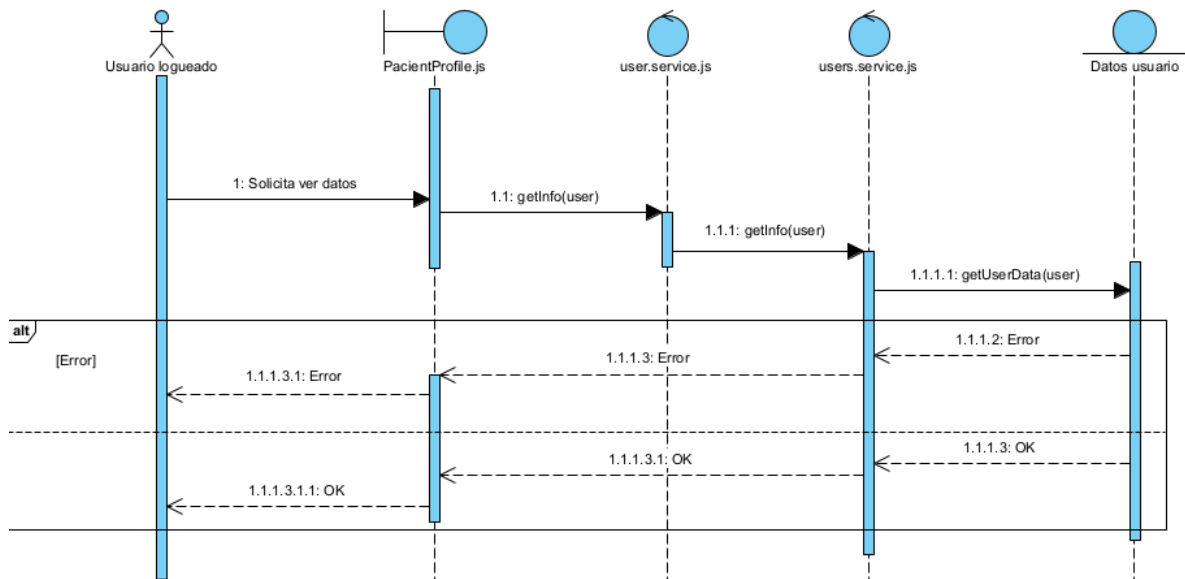


Figura 20: Diagrama de secuencia Consultar info perfil

- UC-009 Eliminar cuenta

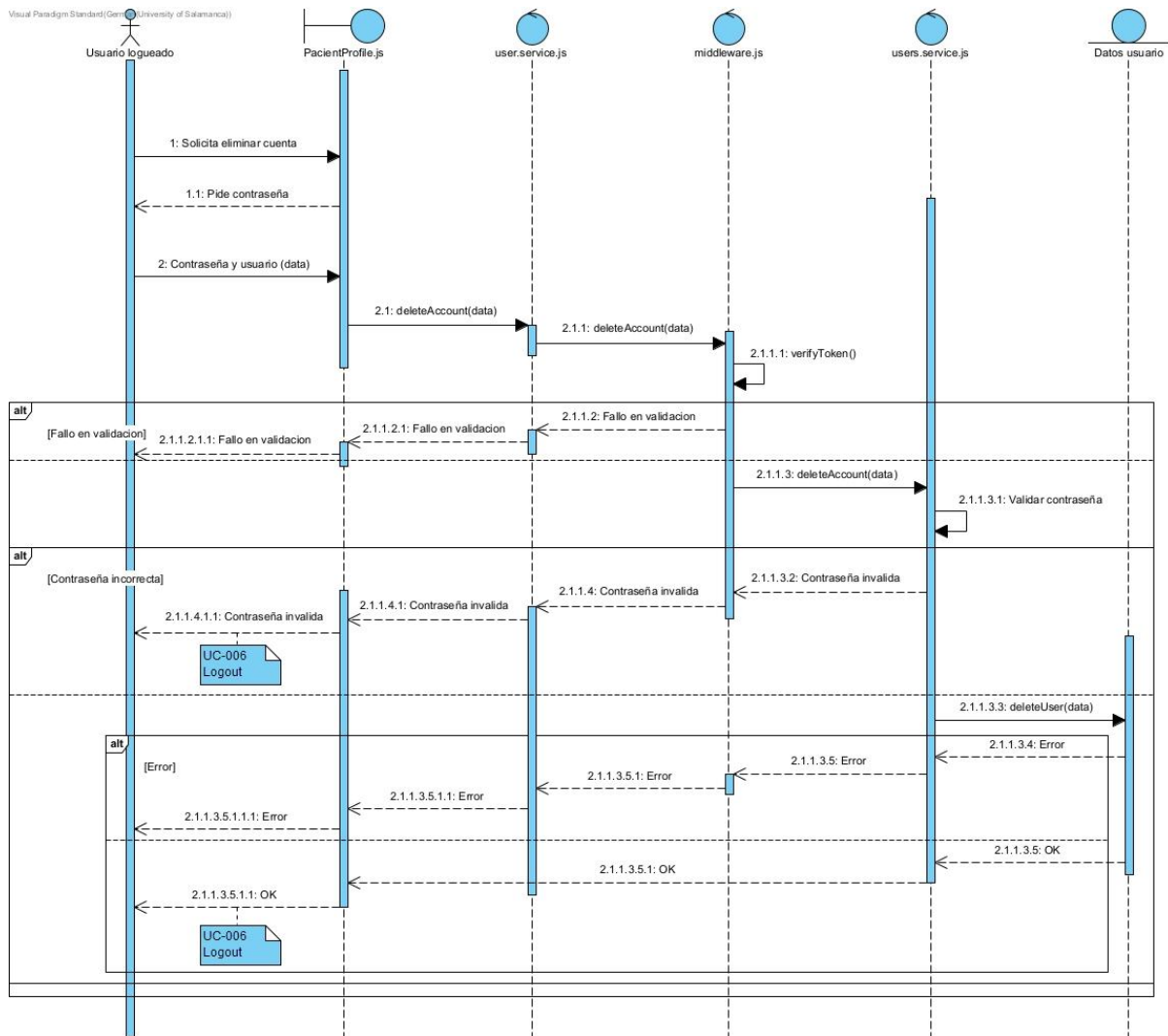


Figura 21: Diagrama de secuencia de Eliminar Cuenta

2.5.2 Paquete Gestión de Pacientes

- UC-010 Ver Pacientes

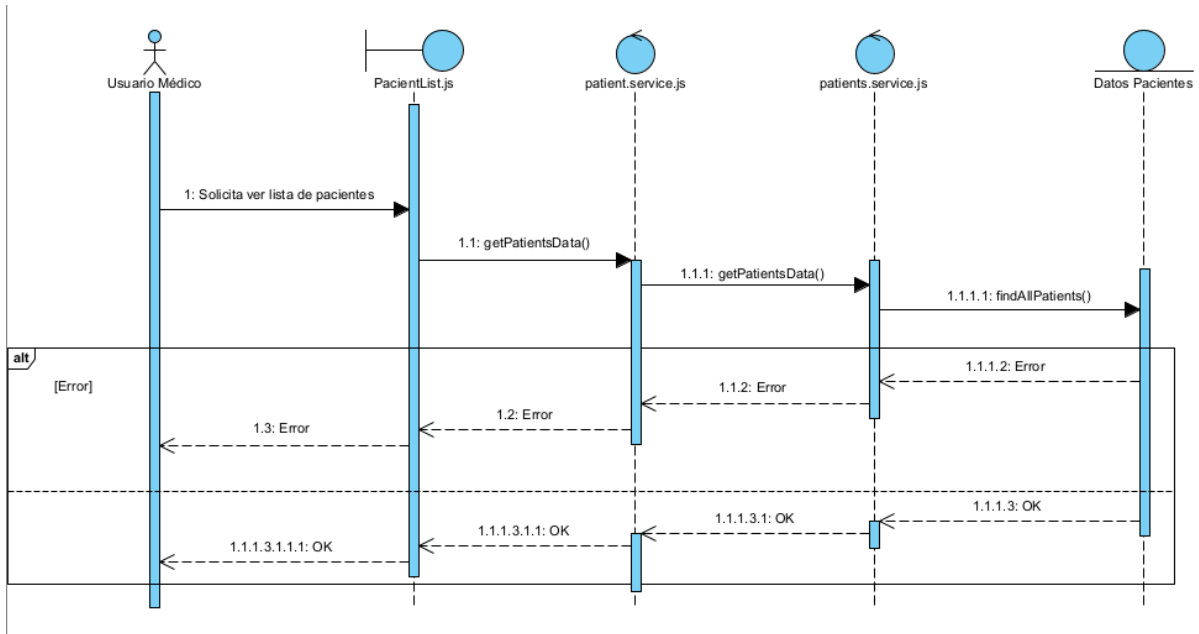


Figura 22: Diagrama de secuencia Ver pacientes

- UC-011 Buscar Pacientes

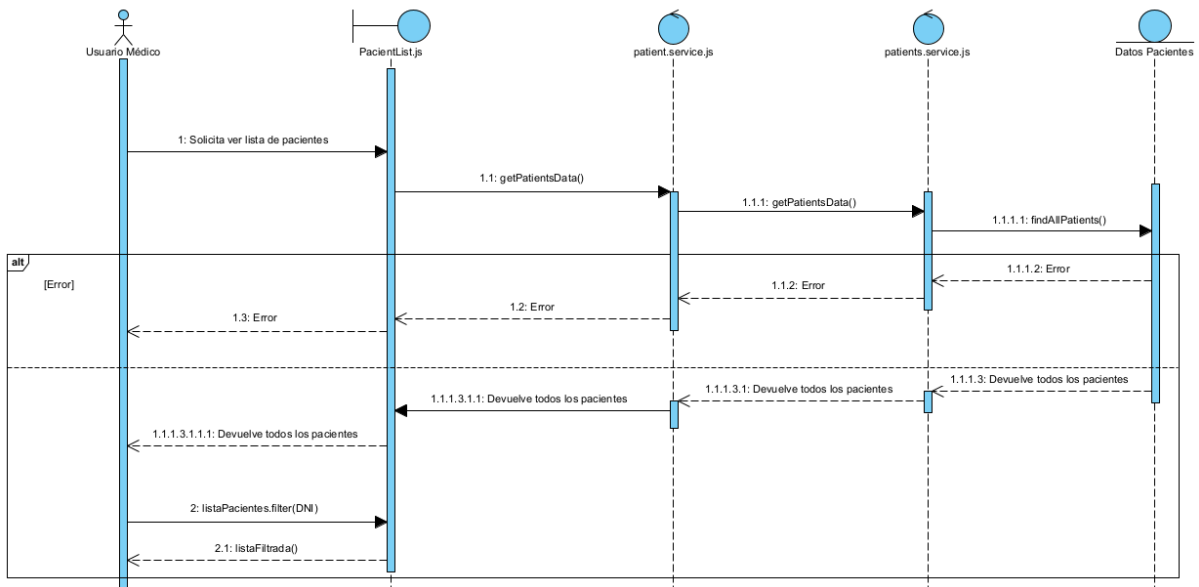


Figura 23: Diagrama de secuencia Buscar Pacientes

- UC-012 Ver historiales médicos

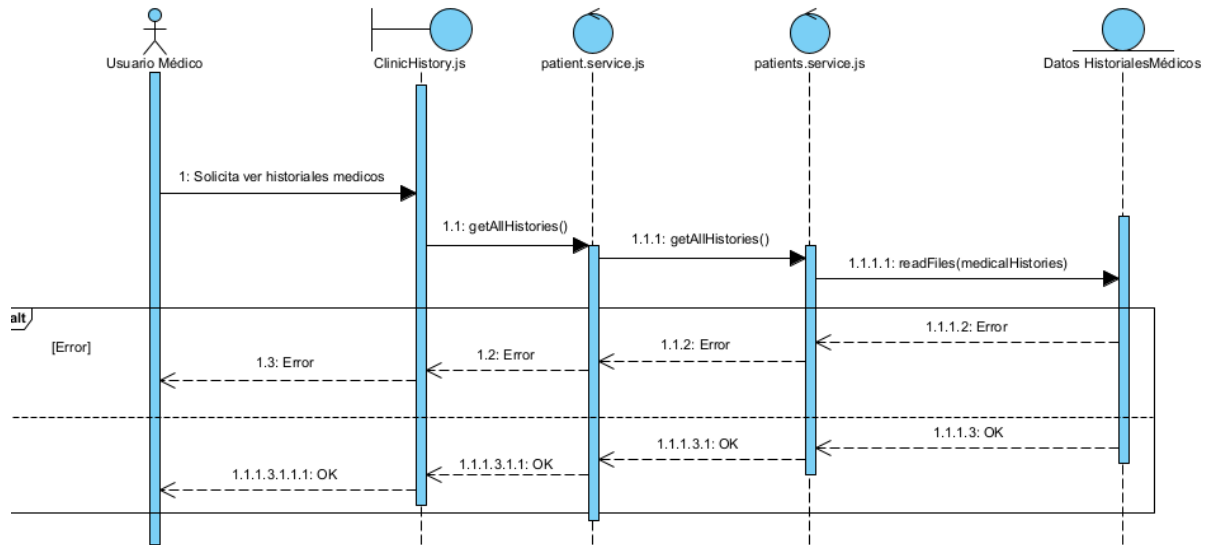


Figura 24: Diagrama de secuencia Ver historiales médicos

- UC-013 Descargar historiales médicos

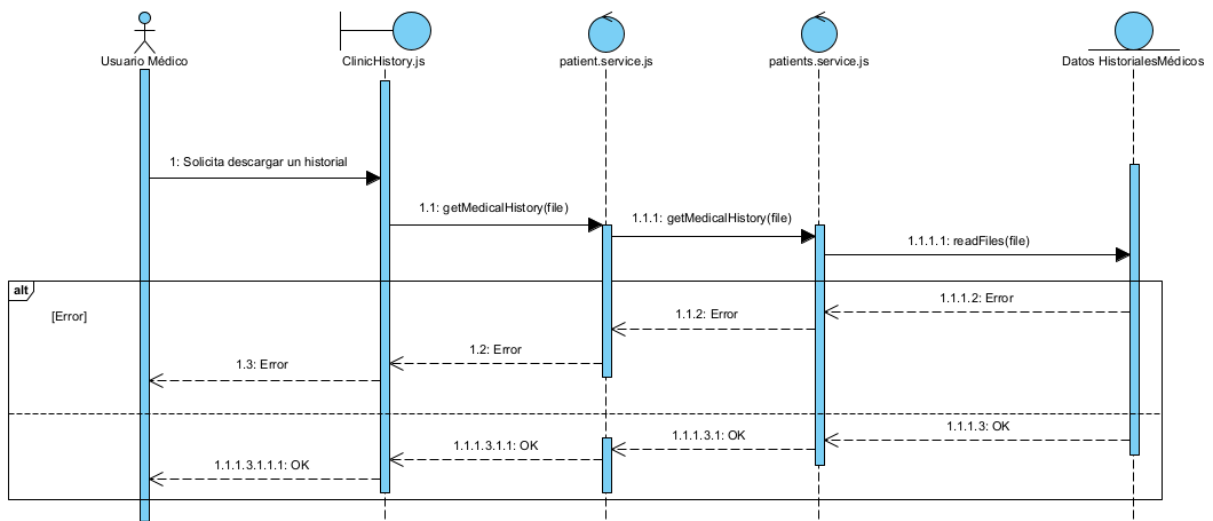


Figura 25: Diagrama de secuencia Descargar historiales médicos

- UC-014 Subir historiales médicos

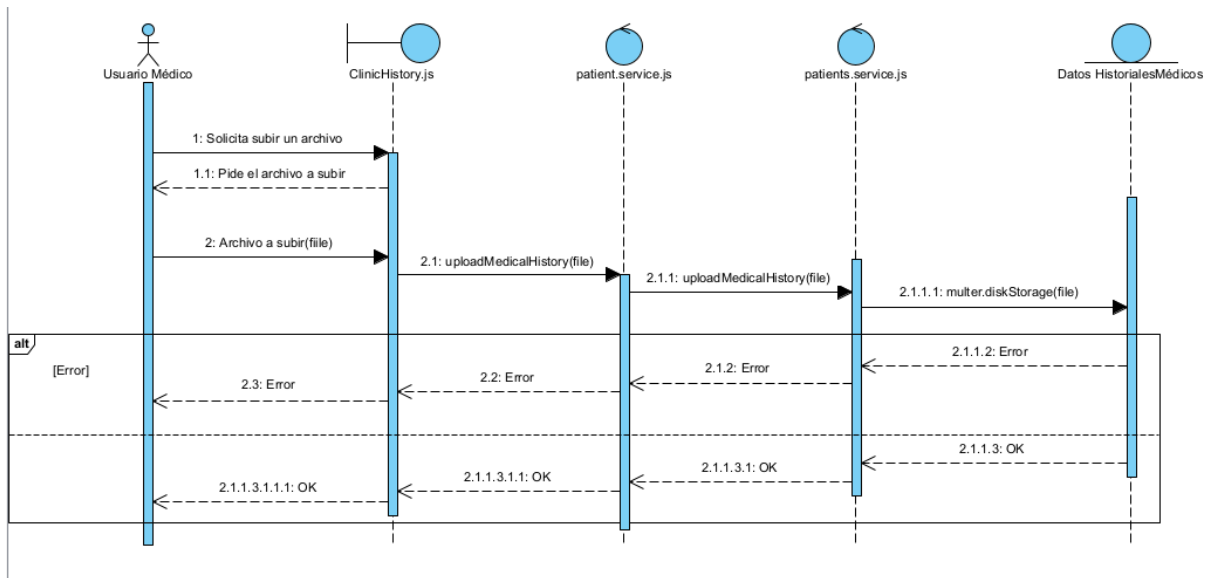


Figura 26: Diagrama de secuencia Subir historiales médicos

- UC-015 Buscar historiales

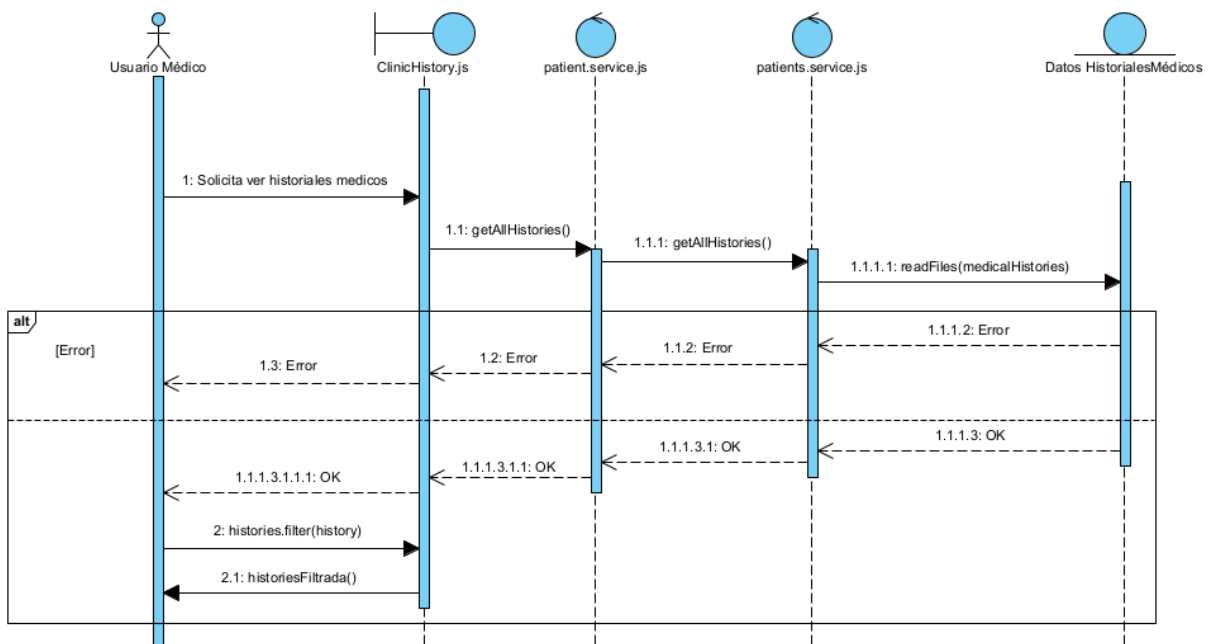


Figura 27: Diagrama de secuencia Buscar historiales

2.5.3 Paquete Gestión de Médicos

- UC-016 Añadir Médico

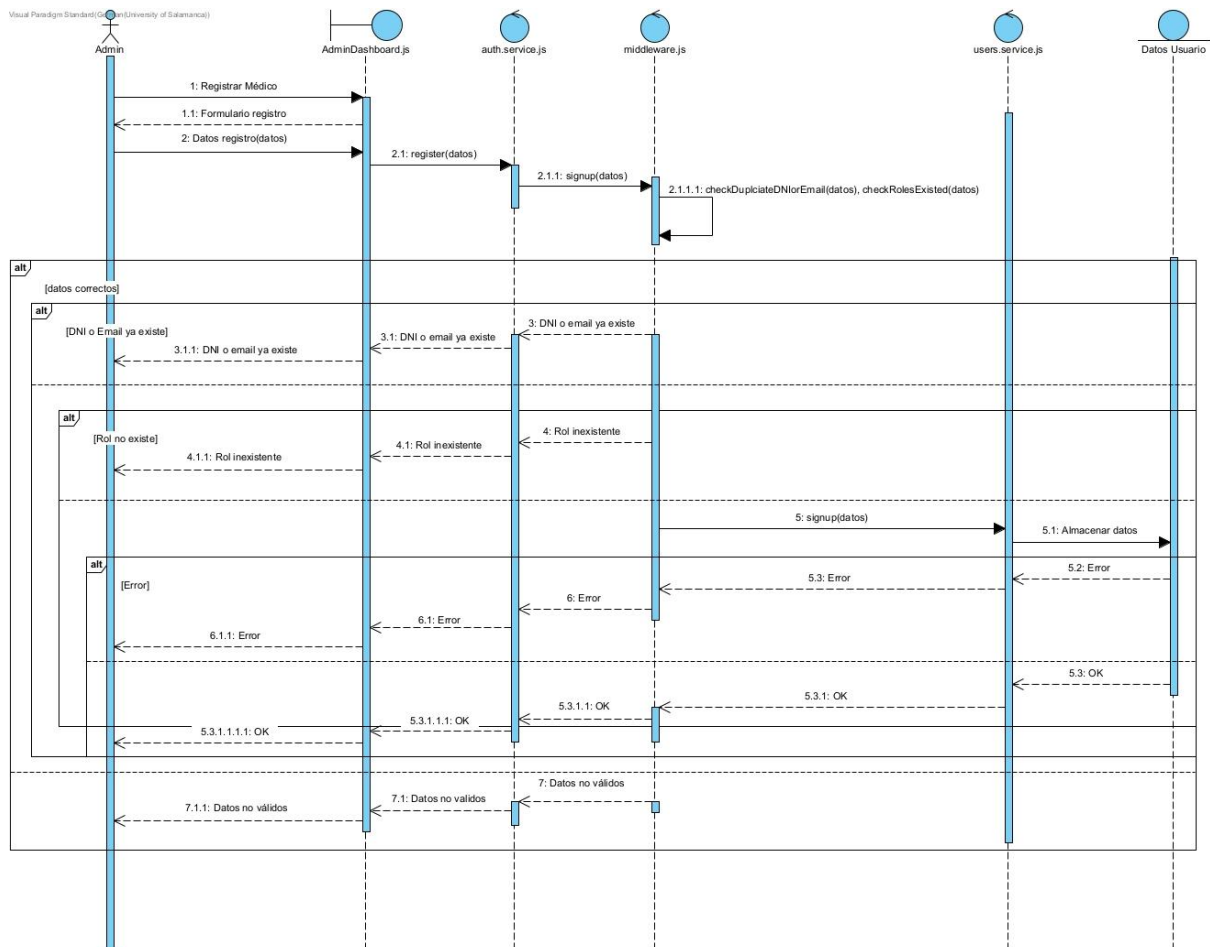


Figura 28: Diagrama de secuencia Añadir Médico

- UC-017 Eliminar Médico

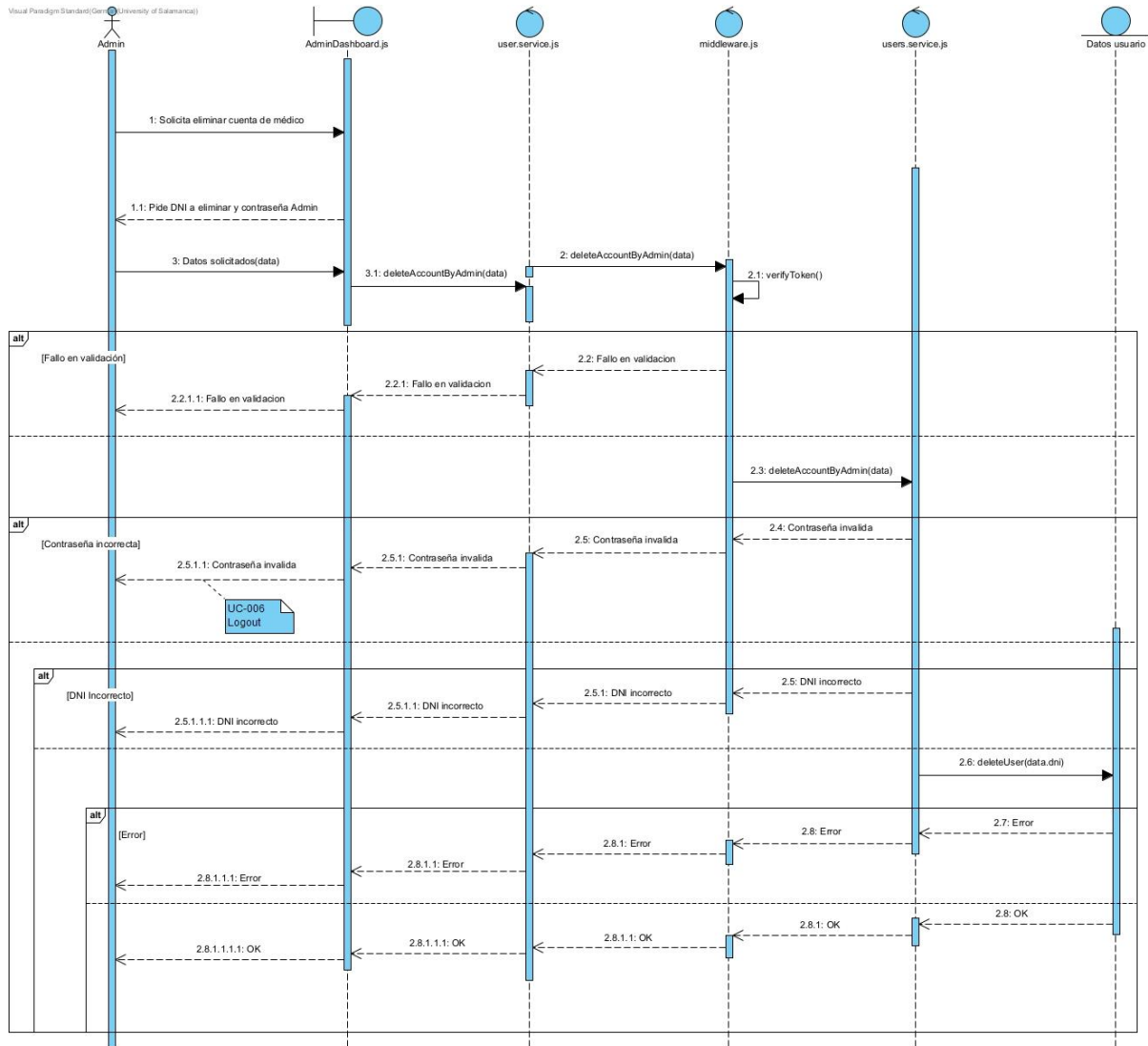


Figura 29: Diagrama de secuencia Eliminar Médico

- UC-018 Ver médicos

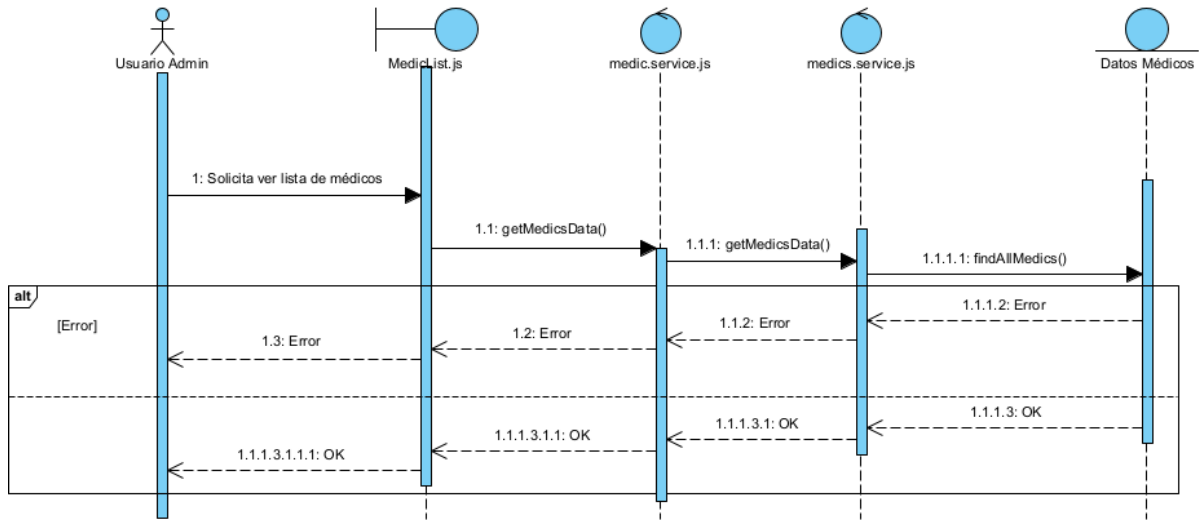


Figura 30: Diagrama de secuencia Ver médicos

- UC-019 Buscar médicos

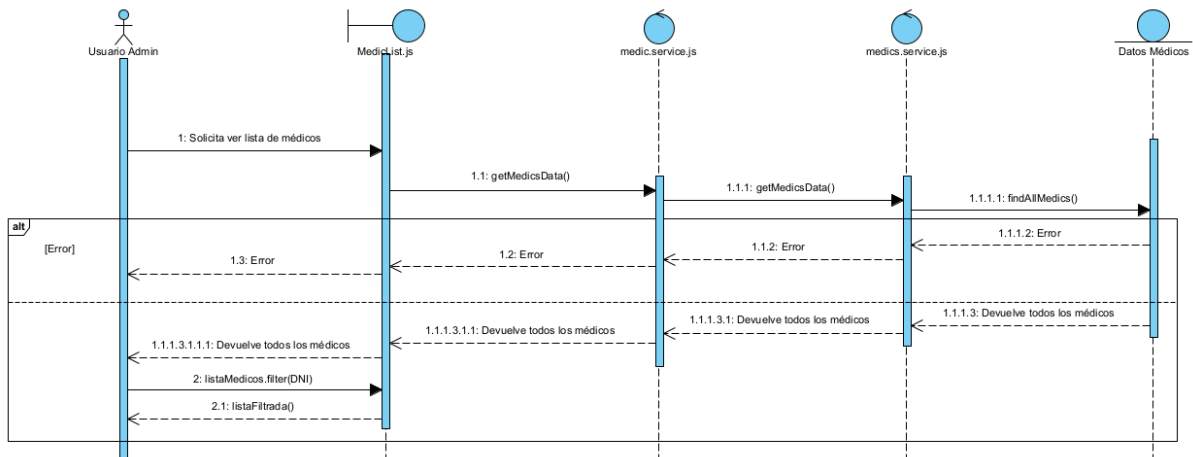


Figura 31: Diagrama de secuencia Buscar médicos

2.5.4 Paquete Gestión de Plataforma Web

- UC -020 Consultar estadísticas personales

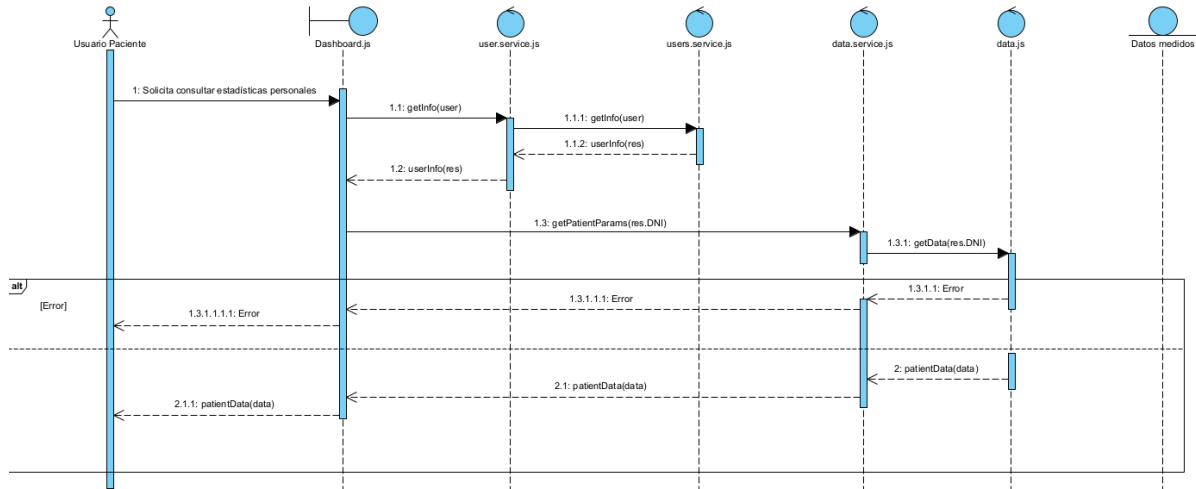


Figura 32: Diagrama de secuencia Consultar estadísticas personales

- UC-021 Consultar estadísticas generales

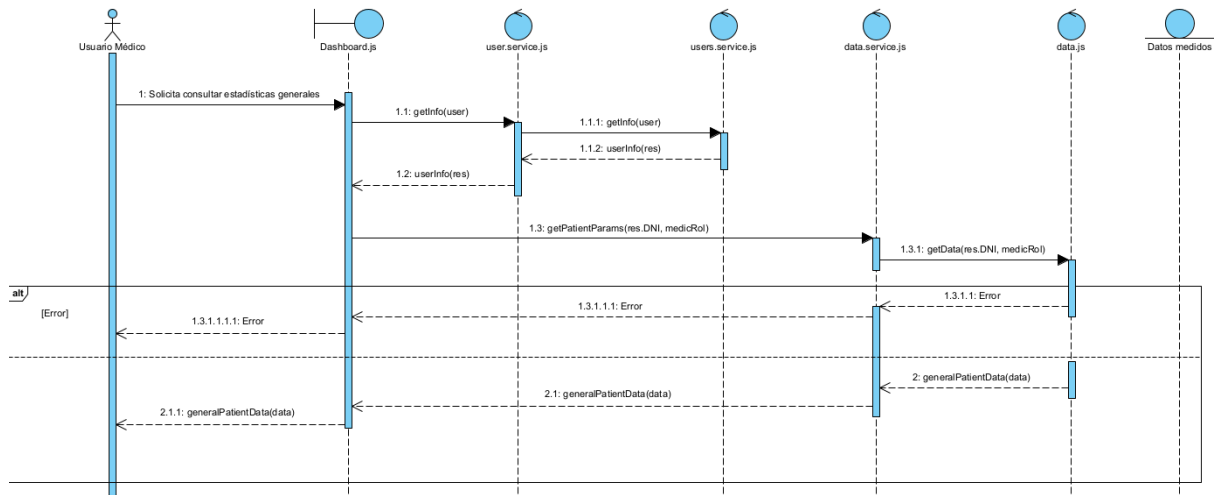


Figura 33: Diagrama de secuencia Consultar estadísticas generales

- UC-022 Acceder a sala de consultas

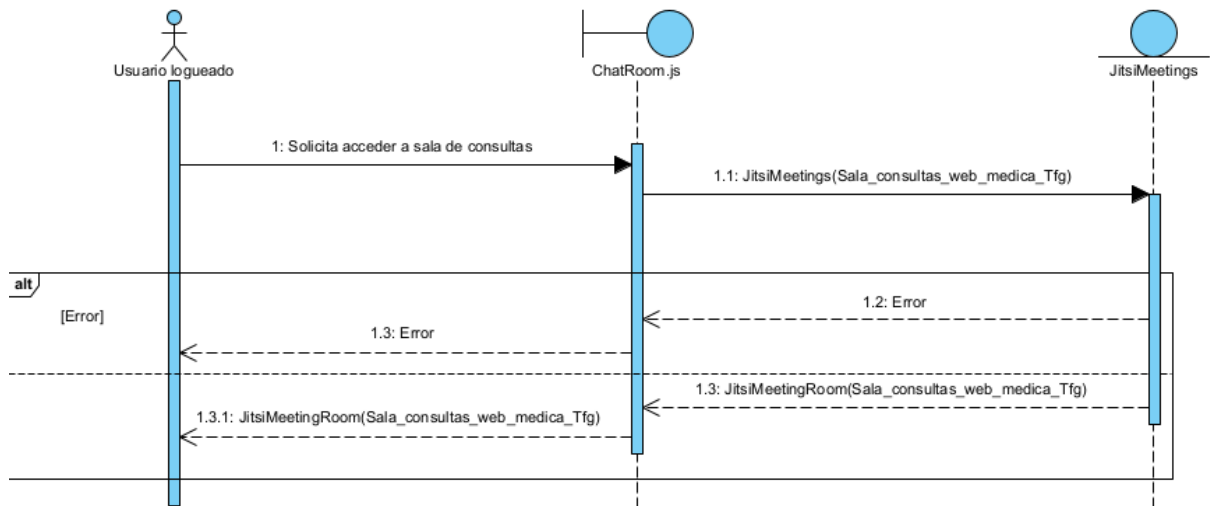


Figura 34: Diagrama de secuencia Acceder a sala de consultas

- UC-023 Actualizar datos

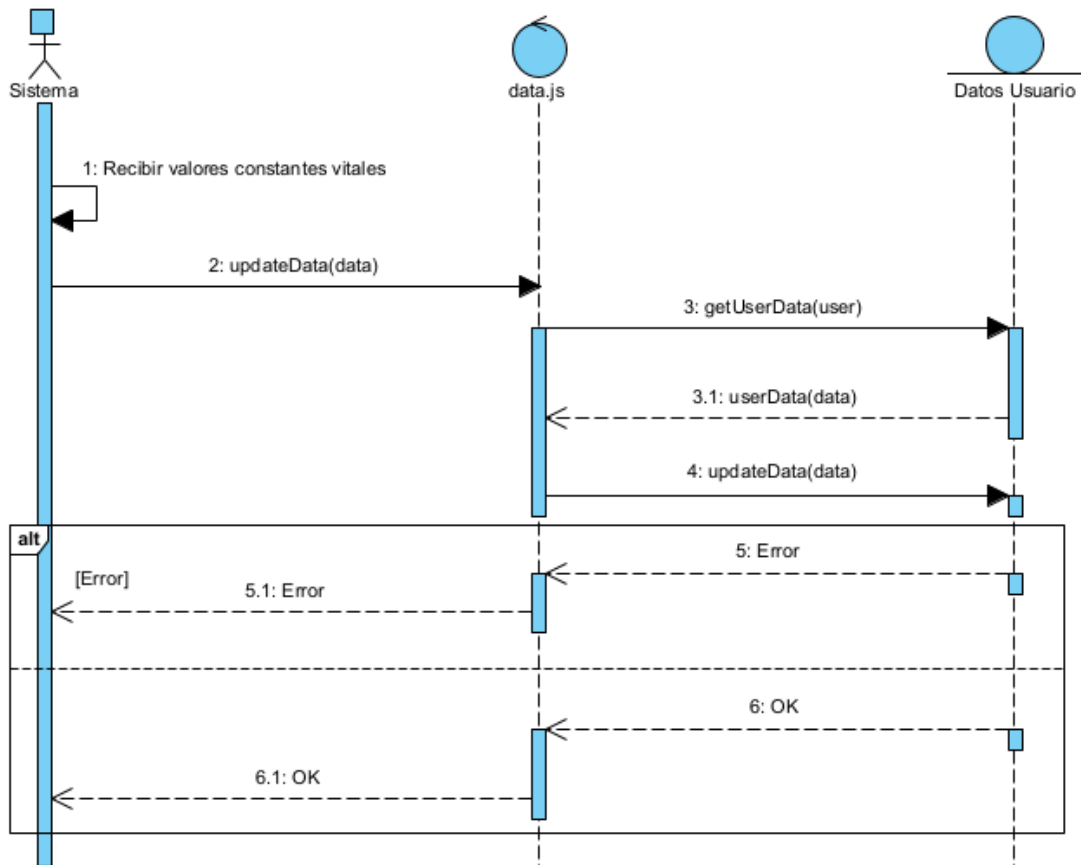


Figura 35: Diagrama de secuencia Actualizar datos

- UC-024 Crear datos

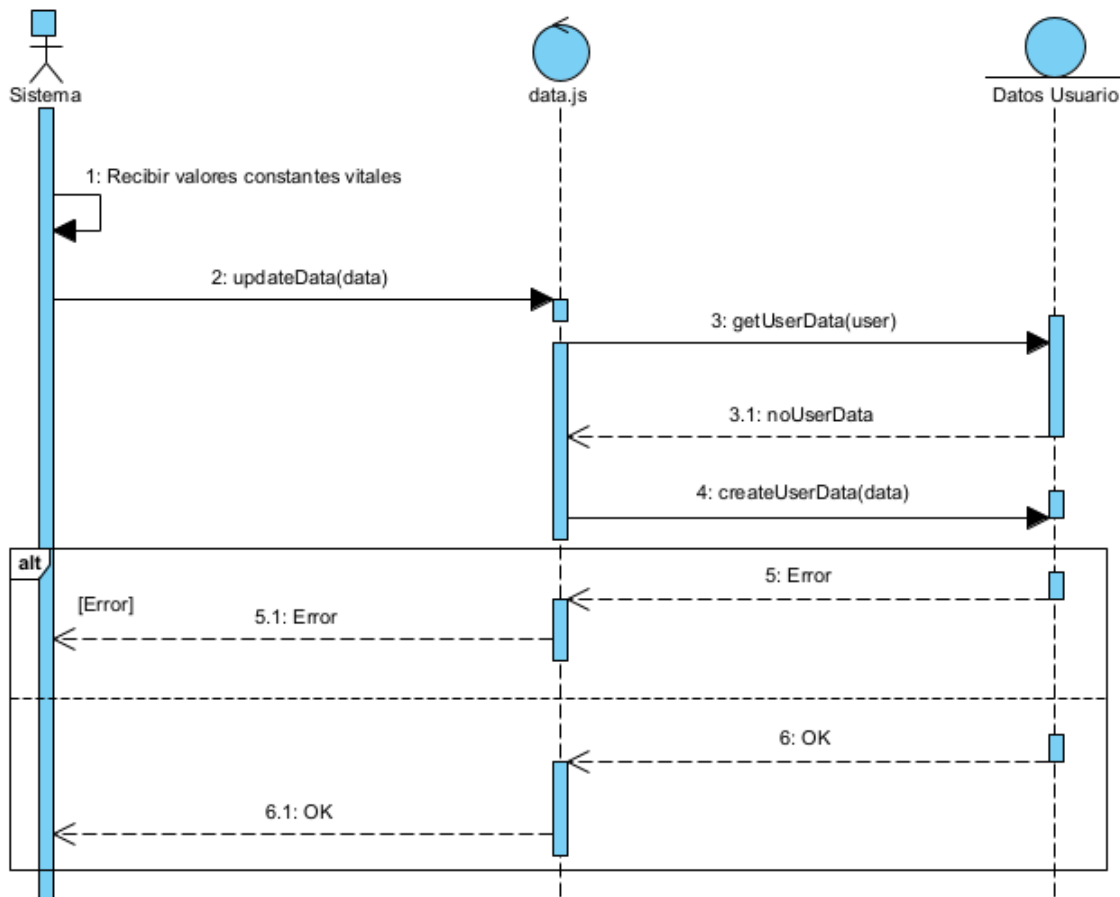


Figura 36: Diagrama de secuencia Crear datos

2.5.5 Paquete Gestión de Tablet

- UC-025 Leer sensores

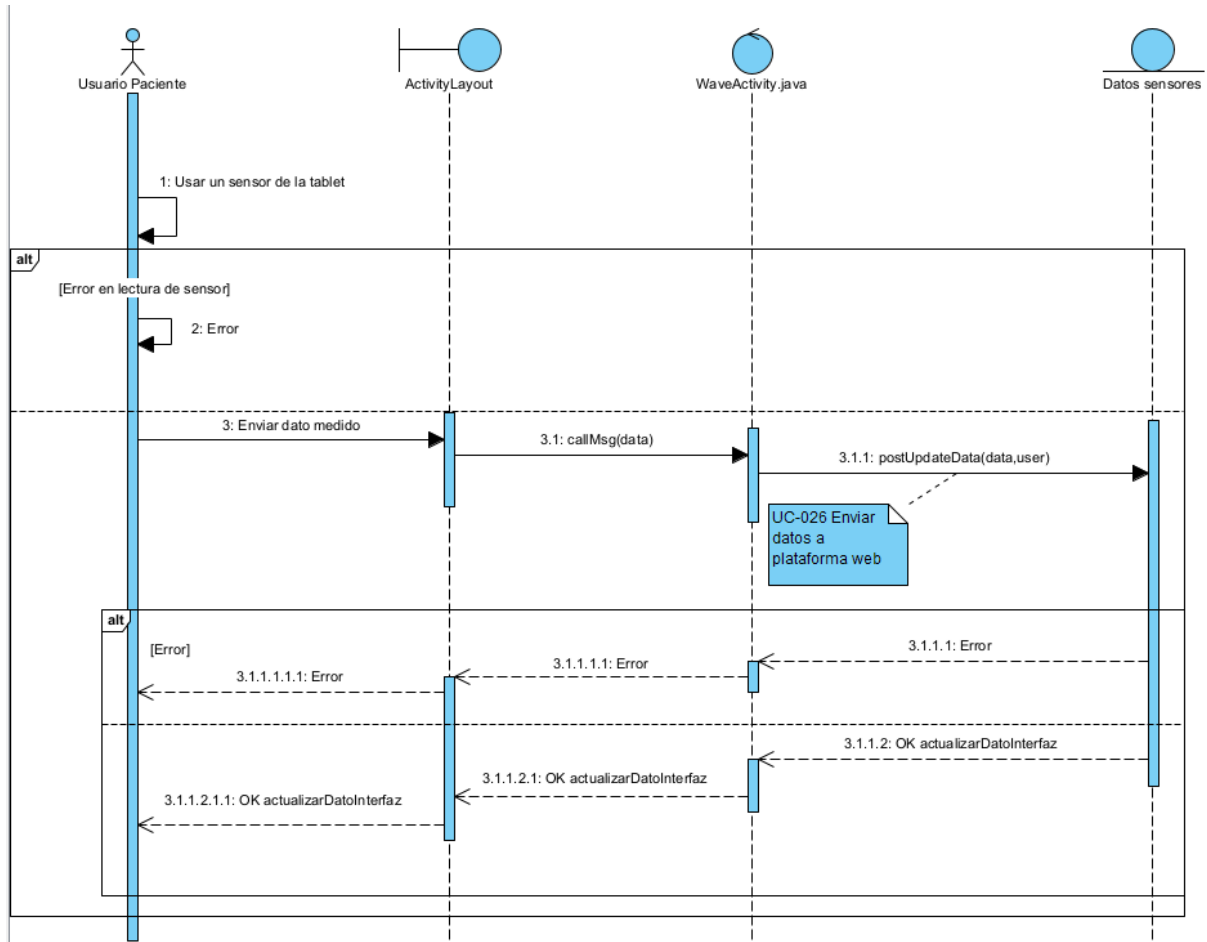


Figura 37: Diagrama de secuencia de Leer sensores

- UC-026 Enviar datos a plataforma web

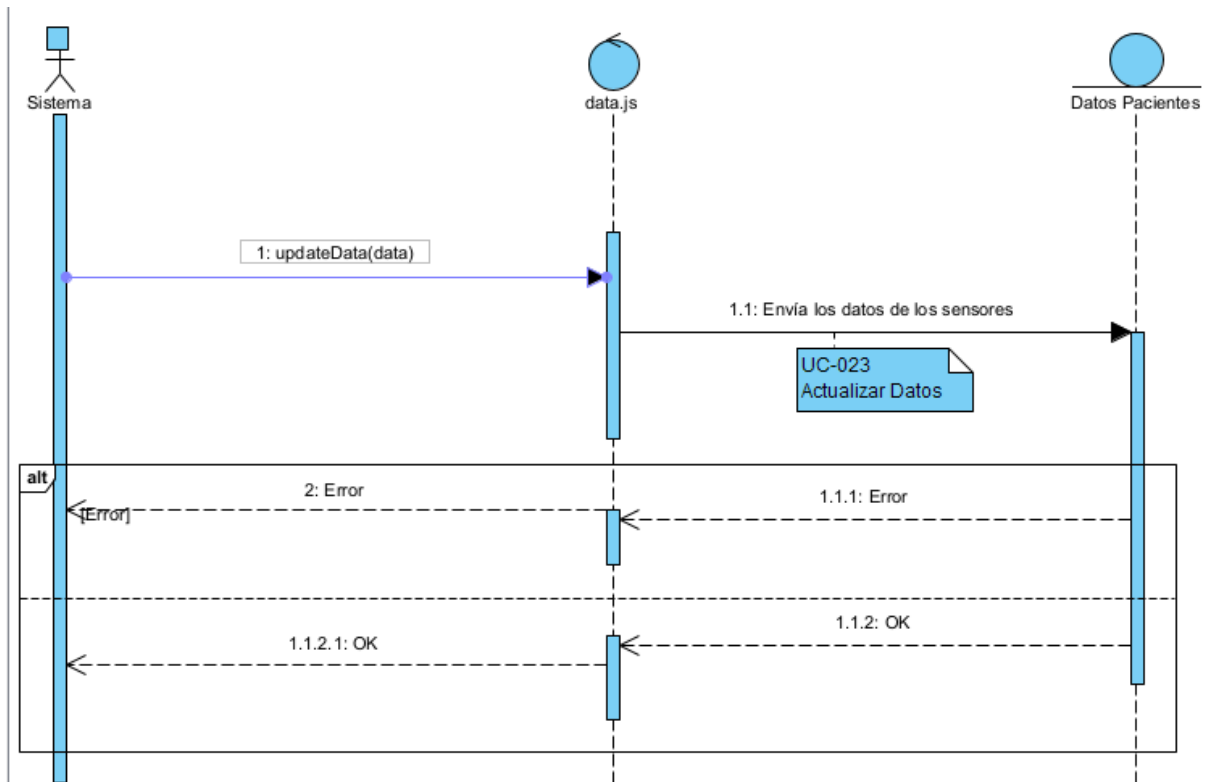


Figura 38: Diagrama de secuencia Enviar datos a plataforma web

- UC-027 Acceder a sala de consultas

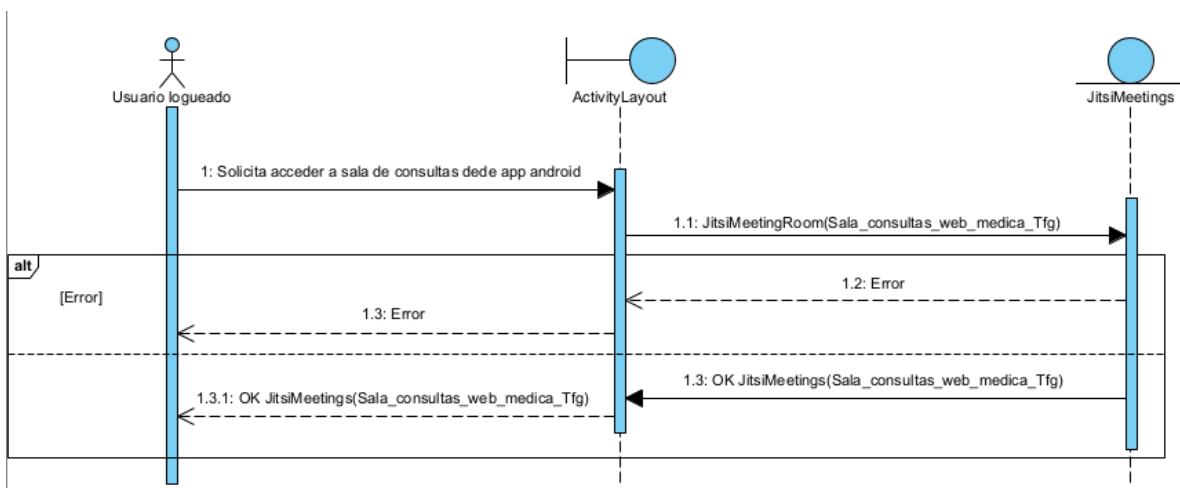


Figura 39: Diagrama de secuencia Acceder a sala de consultas desde tablet

3. Diseño de la base de datos

Para el correcto almacenamiento y gestión de los datos e información necesaria, se usará una base de datos MySQL con el siguiente diagrama Entidad-Relación.

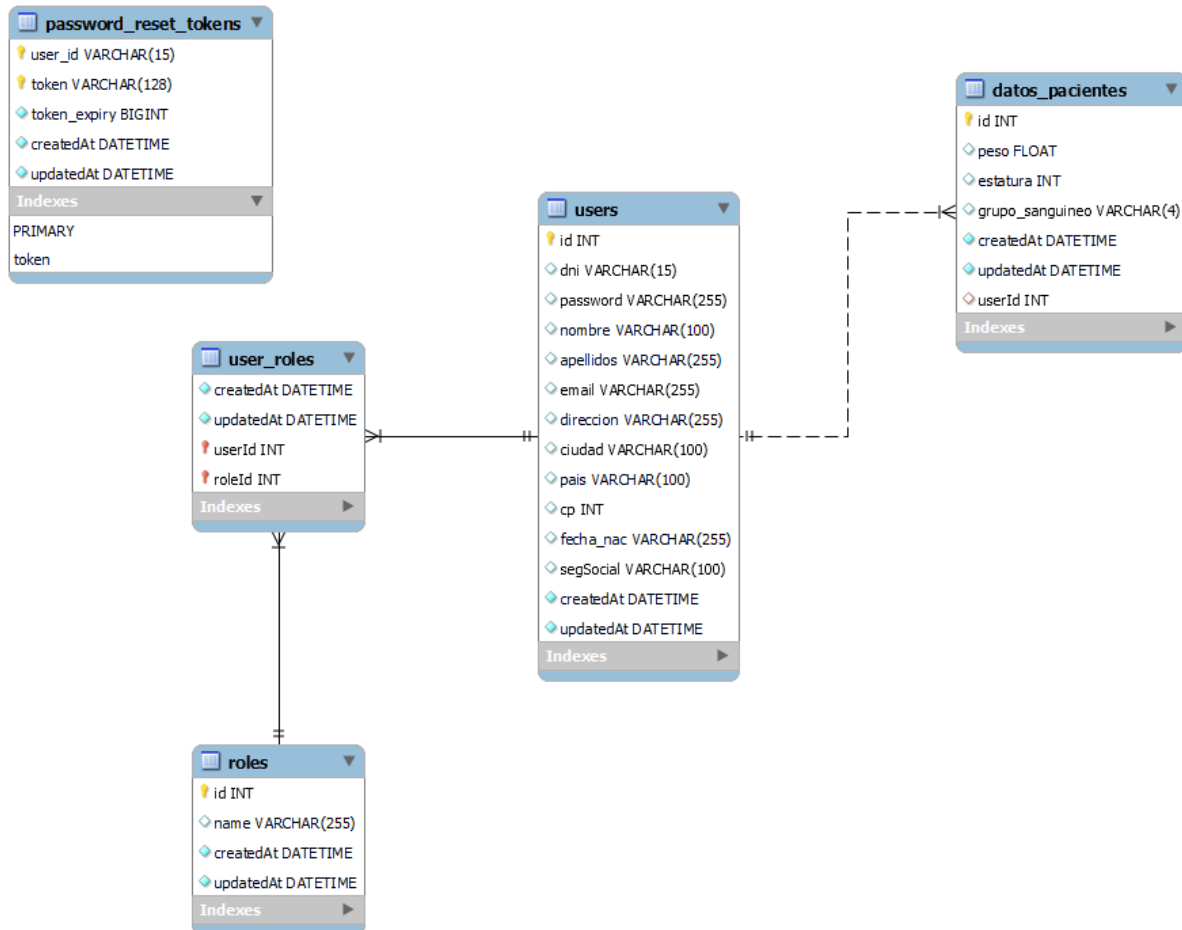


Figura 40: Diagrama Entidad-Relación de la base de datos

Tendremos tres tablas principales:

- **Users:** Tabla principal con todos los datos de los usuarios.
- **Roles:** Tabla con los diferentes tipos de roles que los usuarios podrán tomar.
- **Password_reset_tokens:** Almacenará un token hashado para la recuperación de contraseña de un usuario, junto con el tiempo de expiración. Se permitirán varios tokens por usuario ya que al almacenar el hash, no podremos enviarle dos veces el mismo.
- **Datos_pacientes:** Tabla auxiliar con datos adicionales de los pacientes.

Para relacionar la tabla de usuarios con la tabla de roles, hay una tabla intermedia:

- **User_roles:** Indicará qué rol pertenece a cada usuario.

Las relaciones de cardinalidad entre las tablas se pueden ver en el diagrama.

4. Modelo de despliegue

Se representará un modelo de despliegue utilizando un diagrama que muestre la arquitectura de ejecución del sistema incluyendo los sistemas hardware, y software que intervienen en el mismo.

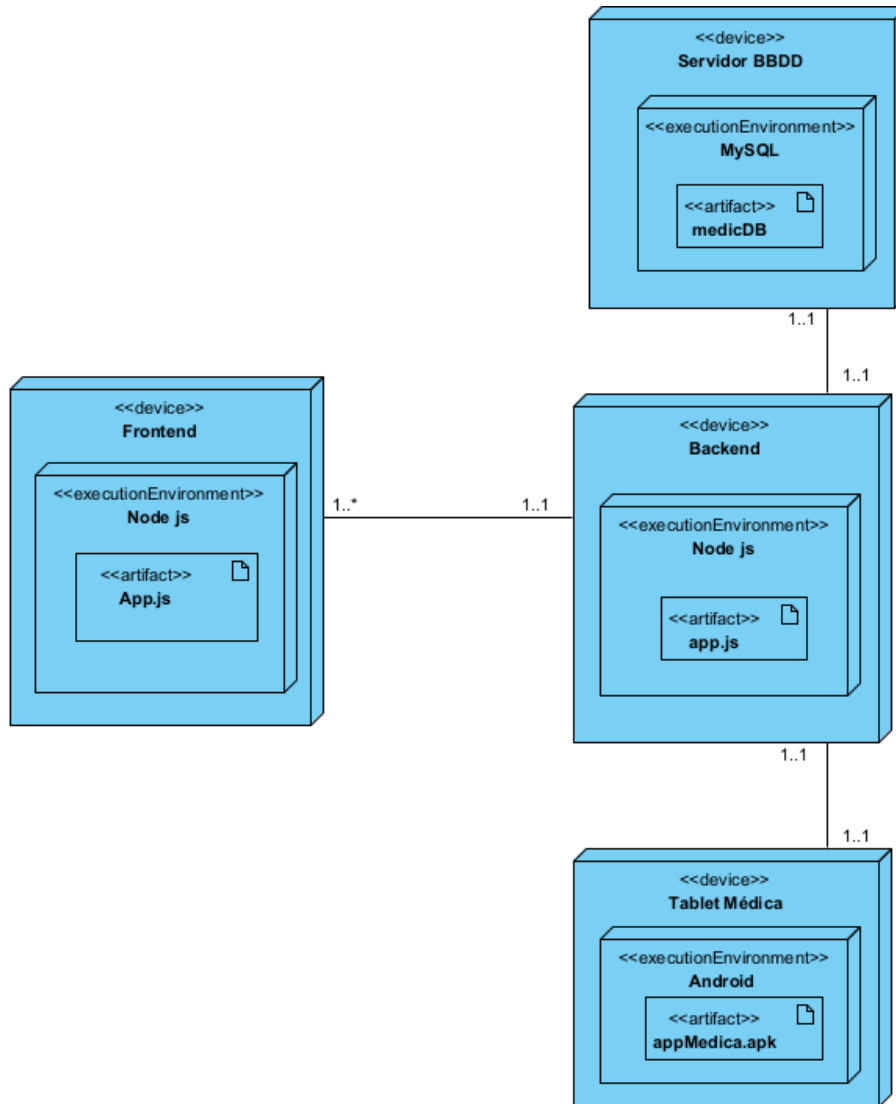


Figura 41: Diagrama de despliegue

Como se puede observar, el sistema se compondrá de tres nodos:

- **Frontend:** Servidor que hará de vista al usuario, donde podrá interactuar de manera sencilla. Se trata de un entorno Node.js
- **Backend:** Servidor que proporcionará todos los microservicios al resto de partes a través de APIs, llevará toda la lógica del sistema y se tratará de un entorno Node.js
- **Tablet Médica:** Dispositivo android con capacidad de conexión a sensores que se comunicarán con el Backend mediante APIs.
- **Servidor BBDD:** Servidor MySQL que almacenará información en una base de datos.

5. Referencias

- Moreno García, María N. TEMA 2 UML. Unified Modeling Language INGENIERÍA DEL SOFTWARE II (Accedido en 22-05-2023)
- Visual Paradigm Manual: (Accedido en 20-05-2023)
 - ◆ <https://www.visual-paradigm.com/support/documents/vpuserguide.jsp>
- Referencias sobre Patrones (Accedido en 01-06-2023)
 - ◆ <https://medium.com/@patricksavalle/rest-api-design-as-a-craft-not-an-art-a3fd97ed3ef4>
 - ◆ <https://dzone.com/articles/understanding-middleware-pattern-in-expressjs>
 - ◆ <https://www.patterns.dev/posts/mediator-pattern>
 - ◆ <https://medium.com/design-microservices-architecture-with-patterns/api-gateway-pattern-8ed0ddfce9df>
 - ◆ <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>