

Plataforma Web de Rehabilitación a Distancia

Entrega Modelado Draft 1



Organización basada en componentes

Integrantes:

Cuevas Rodríguez, Marta
de Pablo, Diego
Silva Rodríguez, Alejandro
Soriano Muñoz, Juan Ignacio

Ingeniería web

Universidad de Málaga

Octubre 2024

Índice

1. Introducción	2
2. Modelado	2
2.1. Diagrama IFML	2
2.2. Diagrama de Casos de Uso	2
2.3. Explicación de Casos de Uso	3
2.4. Diagrama de Secuencia	6
2.4.1. Diagrama de Secuencias: Envío de mensaje	6
2.5. Diagrama de requisitos en SysML	8
2.5.1. Tipos de relaciones	10
2.5.2. Requisitos Principales y sus Relaciones	10
2.6. Diagrama de Clases	12

1. Introducción

La rehabilitación es una fase crítica en el proceso de recuperación de pacientes que han sufrido lesiones, intervenciones quirúrgicas o padecen enfermedades crónicas. Tradicionalmente, la rehabilitación se realiza de manera presencial, lo que puede generar barreras logísticas, económicas y geográficas tanto para los pacientes como para los profesionales de la salud. Ante esta realidad, surge la necesidad de una Plataforma Web de Rehabilitación a Distancia, cuyo objetivo es facilitar el acceso a programas de rehabilitación personalizados, ofrecer seguimiento remoto y mejorar la calidad de vida de los pacientes sin la necesidad de visitas constantes a centros de rehabilitación.

Este proyecto plantea el desarrollo de una plataforma web integral que permita a los pacientes recibir tratamientos de rehabilitación de manera remota, mientras que los profesionales de la salud pueden monitorizar el progreso y ajustar las terapias en tiempo real. Los principales stakeholders involucrados en este proyecto incluyen a pacientes, profesionales de la salud (fisioterapeutas, médicos rehabilitadores) y desarrolladores de software. Los pacientes se beneficiarán de un acceso más flexible a sus tratamientos, mientras que los profesionales podrán optimizar el seguimiento clínico y ajustar terapias de manera eficiente.

Entre los posibles casos de uso se encuentran situaciones como la rehabilitación de un paciente con una lesión muscular, que puede realizar sus ejercicios desde casa bajo la supervisión de un fisioterapeuta a través de videollamadas, o un paciente crónico que, mediante dispositivos de telemetría y un registro de ejercicios, permite que su progreso sea monitorizado de forma continua.

2. Modelado

2.1. Diagrama IFML

Insertar Diagrama Ale

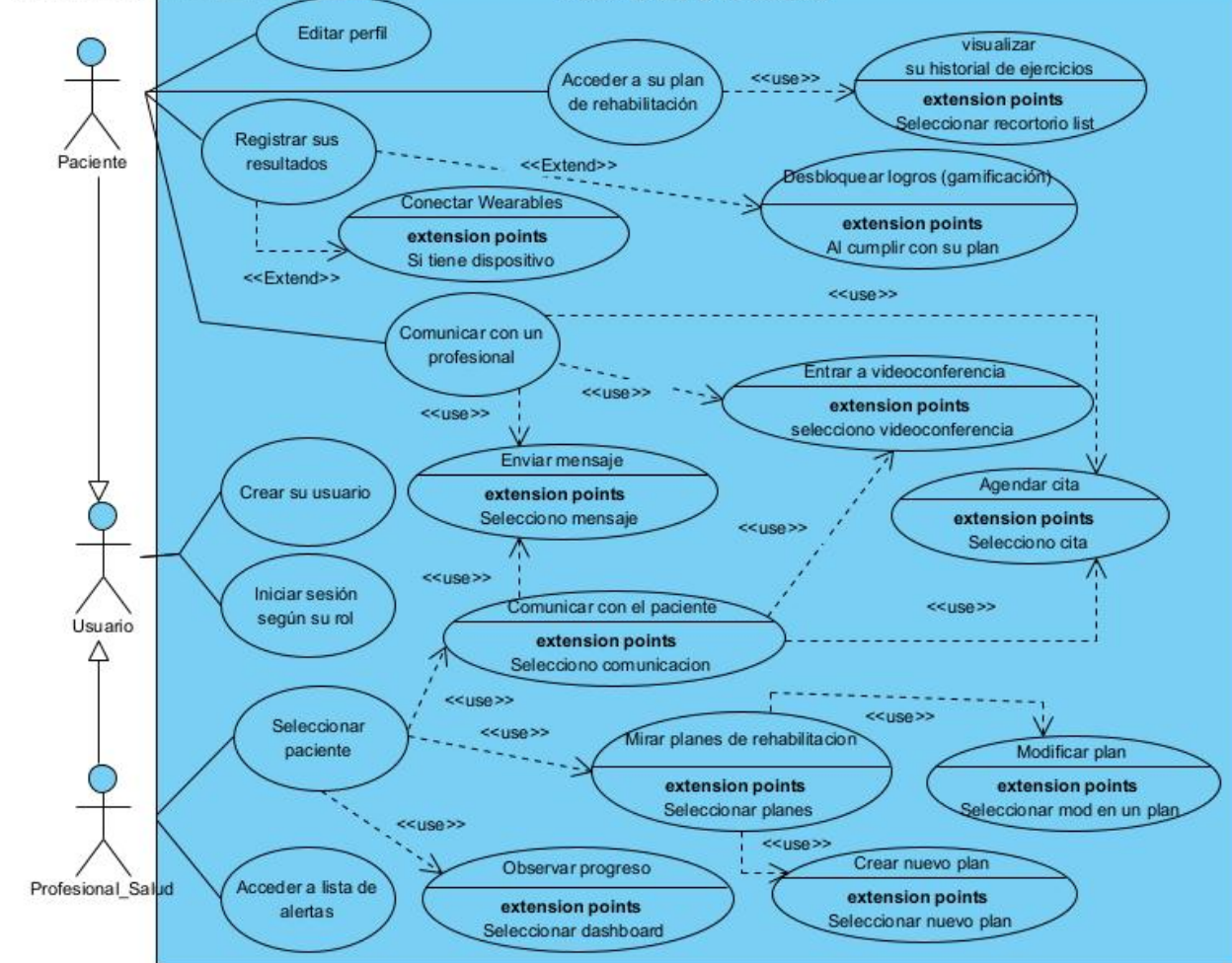
2.2. Diagrama de Casos de Uso

Un Diagrama de Casos de Uso muestra las operaciones que se esperan de un sistema y cómo interactúan los usuarios u otros sistemas con él. Es una herramienta esencial para la planificación y control de proyectos interactivos, permitiendo una visión clara de los requisitos que debe cumplir el sistema.

Cada caso de uso se representa mediante una elipse que denota una operación completa desarrollada entre el sistema y sus actores. El conjunto de casos de uso refleja la totalidad de las operaciones realizadas por el sistema.

Otros elementos que serán visto en el diagrama de casos de uso (ver figura 1)

- **Actor:** Es un usuario del sistema que interactúa con uno o más casos de uso. Un actor puede representar tanto a personas como a sistemas externos que necesitan acceder a información o servicios del sistema. Un actor puede tener múltiples roles, y un caso de uso puede tener varios actores.
- **Generalización de Actor:** Permite agrupar actores con comportamientos similares. Por ejemplo, un *Usuario* puede ser una generalización de *Paciente* y *Profesional de la salud*, dado que ambos comparten ciertas acciones comunes, pero también tienen roles específicos.
- **Relaciones Especiales:**
 - **Uses:** Denota la inclusión del comportamiento de un caso de uso en otro. Se utiliza cuando el comportamiento es compartido entre varios casos de uso.
 - **Extends:** Indica una especialización de un caso de uso base. Se utiliza cuando un comportamiento adicional o alternativo se activa bajo ciertas condiciones.



2.3. Explicación de Casos de Uso

CASO DE USO 1: Paciente registrando sus resultados

- 3

Precondiciones:

- El paciente está registrado en la aplicación.
- El paciente cuenta con un plan de rehabilitación activo.
- El paciente puede disponer de un dispositivo Wearable y conexión a internet adecuada.

Escenario principal:

1. El paciente abre la página web.
2. El paciente inicia sesión correctamente.
3. La página web verifica las credenciales y confirma el inicio de sesión.
4. El paciente hace clic en el botón *Progreso*.
5. La página web carga la pestaña de progreso y logros.
6. El paciente comienza a introducir sus avances en la plataforma.
 - Si el paciente tiene un dispositivo Wearable, este carga automáticamente los datos de actividad en su progreso.
7. La aplicación guarda los datos introducidos.
8. La aplicación otorga un logro al paciente por cumplir su cuota de actividad.

Escenario alternativo:

- En el caso de que el paciente no haya logrado su cuota de actividad, la aplicación no otorgará un logro.
- El paciente recibirá un mensaje de motivación para fomentar una mayor actividad física.

CASO DE USO 2: Pactar una cita con un paciente

- **Nombre:** Pactar una cita con un paciente
- **Objetivo:** El profesional de la salud programa una cita de seguimiento con un paciente.
- **Autor:** Profesional de la salud
- **Descripción:** El profesional de la salud, después de iniciar sesión correctamente, selecciona un paciente de su lista de pacientes. El profesional puede revisar la disponibilidad del paciente y programar una cita de seguimiento, ya sea en persona o por videollamada, para continuar con el plan de rehabilitación. La cita se agenda en el calendario del paciente, y ambos reciben una notificación automática.

Precondiciones:

- El profesional de la salud está registrado en la plataforma y ha iniciado sesión.
- El paciente está registrado en la plataforma y tiene un plan de rehabilitación activo.
- El paciente tiene habilitado el acceso a su calendario para recibir notificaciones.

Escenario principal:

1. El profesional de la salud abre la página web.
2. El profesional inicia sesión correctamente.

3. La página web verifica las credenciales y confirma el inicio de sesión.
4. El profesional selecciona un paciente de la lista.
5. El profesional hace clic en la opción *Pactar cita*.
6. La plataforma muestra el calendario del paciente con las franjas horarias disponibles.
7. El profesional selecciona una fecha y hora para la cita.
8. La plataforma confirma la cita y la agenda en el calendario del paciente.
9. Se envía una notificación automática al paciente y al profesional.

Escenario alternativo:

- Si no hay disponibilidad en el calendario del paciente, la plataforma notifica al profesional y sugiere nuevas franjas horarias.
- Si el paciente cancela la cita, la plataforma envía una notificación al profesional para reprogramar.

CASO DE USO 3: Modificar plan de rehabilitación de un paciente

- **Nombre:** Modificar plan de rehabilitación de un paciente
- **Objetivo:** El profesional de la salud modifica el plan de rehabilitación de un paciente.
- **Autor:** Profesional de la salud
- **Descripción:** El profesional de la salud selecciona un paciente de su lista y revisa su progreso. Basado en el desempeño y las necesidades actuales del paciente, el profesional decide modificar el plan de rehabilitación existente. Esto puede implicar ajustar la rutina de ejercicios, cambiar la frecuencia de sesiones o actualizar las metas de recuperación. Una vez modificado, el nuevo plan es guardado y notificado al paciente.

Precondiciones:

- El profesional de la salud está registrado en la plataforma y ha iniciado sesión.
- El paciente tiene un plan de rehabilitación activo.
- El profesional tiene acceso al historial de progreso del paciente.

Escenario principal:

1. El profesional de la salud abre la página web.
2. El profesional inicia sesión correctamente.
3. La página web verifica las credenciales y confirma el inicio de sesión.
4. El profesional selecciona un paciente de la lista.
5. El profesional hace clic en la opción *Modificar plan de rehabilitación*.
6. La plataforma muestra el plan de rehabilitación actual del paciente.
7. El profesional realiza modificaciones al plan (ej. ajustar ejercicios, cambiar frecuencia, actualizar metas).
8. El profesional guarda las modificaciones.

9. La plataforma envía una notificación al paciente informando los cambios en su plan.

Escenario alternativo:

- Si el profesional no está seguro de los cambios, puede optar por guardar un borrador del nuevo plan y continuar más tarde.
- Si el paciente no está de acuerdo con las modificaciones, puede enviar un mensaje al profesional para discutir los ajustes.

2.4. Diagrama de Secuencia

Un diagrama de secuencia muestra la interacción de un conjunto de objetos de una aplicación a lo largo del tiempo. Es una herramienta importante porque aporta detalles a los casos de uso, aclarando cómo se intercambian mensajes entre los objetos. Además, revela cómo las clases diseñadas interactúan en el contexto de una operación específica.

Este tipo de diagrama es esencial para comprender la dinámica del sistema en términos de secuencia temporal, permitiendo visualizar el orden en que los mensajes se envían entre objetos y cómo se desarrollan las operaciones en el sistema. Detalla la lógica de las interacciones y ayuda a verificar que el diseño satisface los requisitos funcionales descritos en los casos de uso.

Elementos principales en Visual Paradigm (ver Figura 2):

- **Línea de vida (lifeline):** Representa la existencia de un objeto a lo largo del tiempo, y se dibuja como una línea vertical punteada con un rectángulo de encabezado.
- **Activación:** Denotada por un rectángulo a lo largo de la línea de vida, representa el periodo en el cual un objeto está ejecutando una operación.
- **Mensaje:** Los mensajes entre objetos se representan mediante líneas sólidas con flechas, que indican el flujo de información desde el objeto que emite el mensaje hasta el objeto que lo recibe y ejecuta.

2.4.1. Diagrama de Secuencias: Envío de mensaje

En este apartado, se ha seleccionado un caso de uso particular para detallar la interacción entre las diferentes entidades involucradas en el proceso de **envío de un mensaje prioritario** por parte del paciente a su profesional de la salud. Este enfoque se eligió para profundizar en las interacciones sin caer en la redundancia con otros diagramas. El caso de uso describe cómo el paciente puede optar por enviar un mensaje marcado como *prioritario*, lo que genera una notificación adicional por correo al profesional, además de ser almacenado en el buzón de alertas.

Clases involucradas (lifelines/actores):

- **Paciente:** Inicia el flujo al redactar y enviar el mensaje a través de la aplicación.
- **Servidor de la aplicación:** Recibe la solicitud del paciente, almacena el mensaje y gestiona las notificaciones.
- **Base de datos:** Se encarga de almacenar los mensajes enviados y gestionar las alertas en el buzón.
- **Servidor de correo:** Es el encargado de enviar una notificación por correo electrónico al profesional si el mensaje es marcado como prioritario.
- **Profesional de la salud:** El destinatario del mensaje, quien accede a la plataforma para leer el mensaje.

Secuencia de Interacciones:

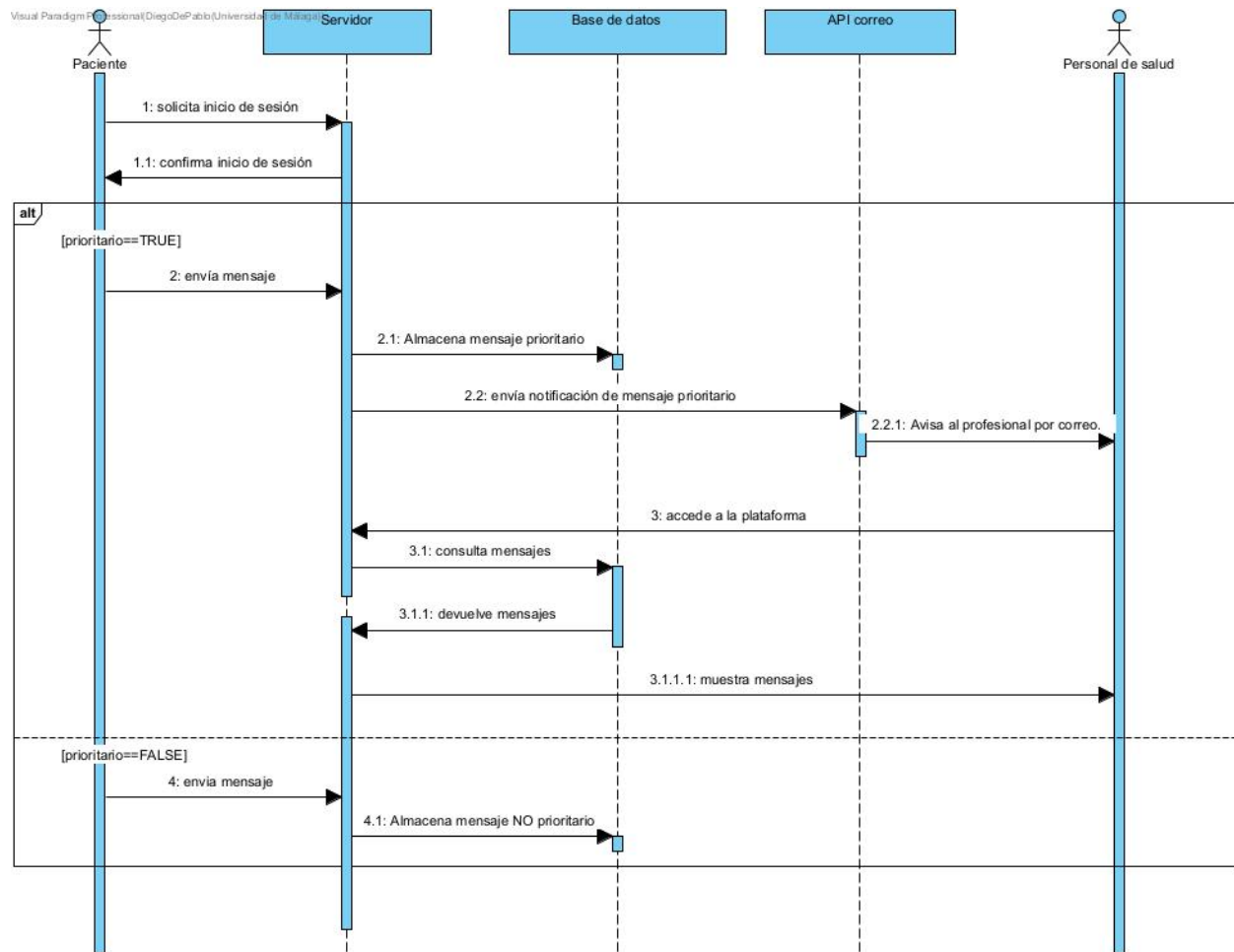


Figura 2: Diseño de Diagrama de Secuencia: Paciente envía un mensaje

- **Paciente → Iniciar sesión** (`IniciarSesion(String: usuario, String: contraseña)`): El paciente solicita iniciar sesión en la aplicación para acceder a sus funcionalidades, proporcionando sus credenciales (usuario y contraseña).
- **Servidor de la aplicación → Verificación de credenciales** (`Verificación(String: usuario, String: contraseña)`): El servidor de la aplicación valida las credenciales consultando la base de datos para confirmar la autenticidad de los datos.
- **Servidor de la aplicación → Login exitoso** (`Login(Boolean: correcto)`): El servidor responde con un valor booleano, confirmando si las credenciales son correctas (`true`) o incorrectas (`false`). En caso positivo, el paciente accede a la interfaz de usuario.

Redacción y envío del mensaje:

- **Paciente → Servidor de la aplicación** (`EnviarMensaje(String: mensaje, Boolean: prioritario)`): El paciente redacta el mensaje y lo envía al servidor, seleccionando si es prioritario o no. El mensaje se transmite junto con un parámetro booleano que indica la prioridad.

- **Servidor de la aplicación** → **Base de datos** (`AlmacenarMensaje(String: mensaje)`): El servidor almacena el mensaje en la base de datos, junto con la etiqueta de prioridad si corresponde.
- **Servidor de la aplicación** → **Profesional de la salud** (`NotificarMensaje()`): Si el mensaje no es prioritario, el servidor notifica al profesional a través de la plataforma, almacenando el mensaje en el buzón de alertas.
- **Servidor de la aplicación** → **Servidor de correo** (`EnviarCorreoNotificacion()`): En el caso de que el mensaje sea prioritario, el servidor también envía una notificación por correo electrónico al profesional a través del servidor de correo.

Lectura del mensaje por el profesional:

- **Profesional de la salud** → **Servidor de la aplicación** (`ConsultarMensajes()`): El profesional accede a la plataforma y solicita la lista de mensajes.
- **Servidor de la aplicación** → **Base de datos** (`RecuperarMensajes()`): El servidor consulta los mensajes almacenados en la base de datos para ese profesional.
- **Base de datos** → **Servidor de la aplicación** (`DevolverMensajes()`): La base de datos devuelve los mensajes solicitados.
- **Servidor de la aplicación** → **Profesional de la salud** (`MostrarMensajes()`): El servidor muestra los mensajes al profesional, quien puede proceder a leerlos y responder si es necesario.

Escenarios alternativos: El diagrama incluye un bloque condicional (*alt*) que refleja los dos escenarios posibles:

- Si el mensaje es **prioritario**, el profesional recibe una notificación adicional por correo.
- Si el mensaje no es prioritario, simplemente se almacena en el buzón de alertas del profesional.

2.5. Diagrama de requisitos en SysML

El Diagrama de Requisitos en SysML es una herramienta visual utilizada para documentar y mostrar las relaciones entre los distintos requisitos de un sistema y su trazabilidad. Su objetivo principal es facilitar la comprensión de cómo los requisitos están interconectados y asegurar que se gestionen de manera adecuada durante todo el ciclo de vida del sistema. A través de este diagrama, se pueden identificar relaciones como dependencias, derivaciones o refinamientos entre requisitos, lo que ayuda a los equipos a garantizar que todos los aspectos del sistema estén cubiertos y alineados con los objetivos y restricciones establecidos.

En la figura 3 se muestran las diferentes relaciones entre los requisitos funcionales y no funcionales de nuestro sistema.

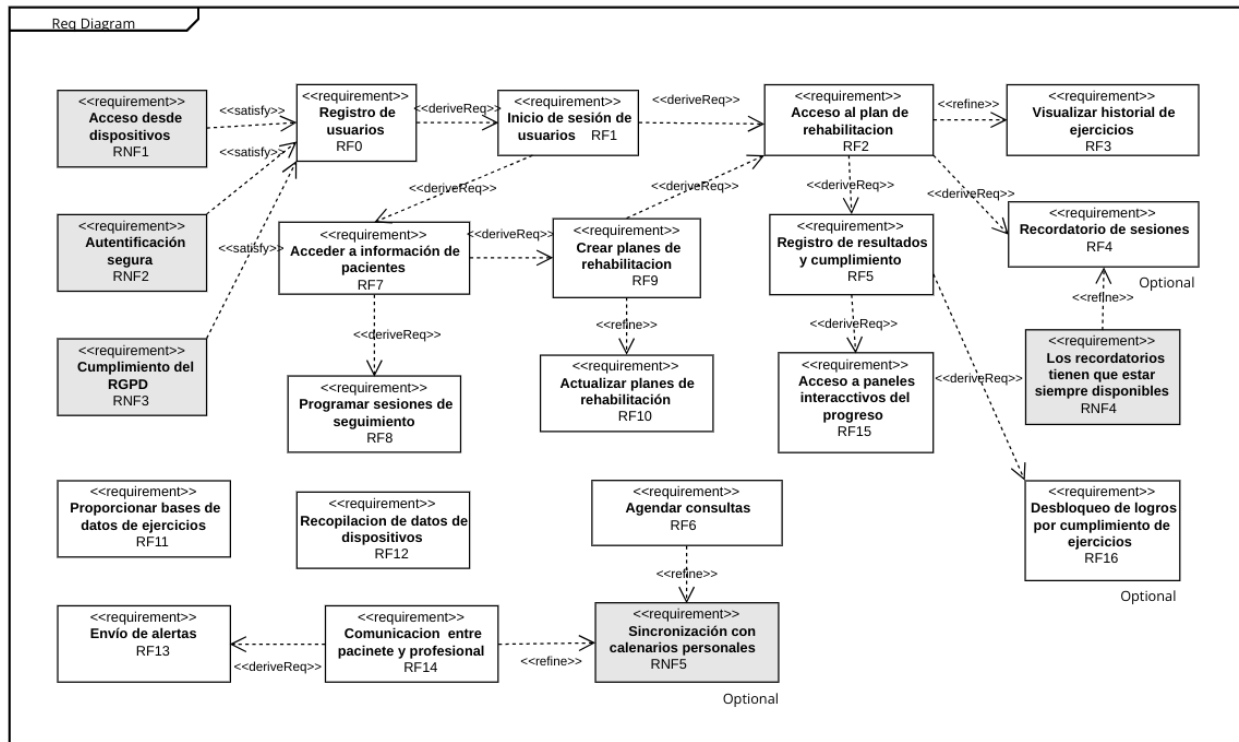


Figura 3: Diseño de Diagrama de requisitos en SysML

2.5.1. Tipos de relaciones

En el esquema se describen tres tipos de relaciones diferentes.

- **Derivación ("deriveReq")**: Un requisito se deriva de otro, es decir, el segundo requisito es una consecuencia directa del primero.
- **Satisfacción ("satisfy")**: Un requisito no funcional satisface o cumple con un requisito funcional, indicando que ese aspecto técnico o de comportamiento es necesario para su implementación.
- **Refinamiento (refine")**: Un requisito refina otro, añadiendo más detalles o especificaciones para clarificar o profundizar en cómo se debe cumplir el requisito original.

2.5.2. Requisitos Principales y sus Relaciones

1. RNF1 - Acceso desde dispositivos *satisface* a RF0 - Registro de usuarios.

- El requisito no funcional RNF1 indica que el sistema debe ser accesible desde distintos dispositivos, lo que satisface el requisito **Registro de usuarios**, ya que los usuarios deben poder registrarse sin importar el dispositivo.

2. RNF2 - Autenticación segura *satisface* a RF0 - Registro de usuarios.

- El requisito de autenticación segura satisface el requisito de **Registro de usuarios**, ya que el proceso de registro debe ser seguro para proteger la información de los usuarios.

3. RNF2 - Autenticación segura *satisface* a RF7 - Acceder a información de pacientes.

- La autenticación segura satisface el acceso a la información de los pacientes, asegurando que solo los usuarios autorizados puedan acceder a dicha información.

4. RF0 - Registro de usuarios *derivaReq* a RF1 - Inicio de sesión de usuarios.

- El inicio de sesión se deriva del **Registro de usuarios**, ya que solo un usuario registrado podrá iniciar sesión en el sistema.

5. RF1 - Inicio de sesión de usuarios *derivaReq* a RF2 - Acceso al plan de rehabilitación.

- El acceso a los planes de rehabilitación se deriva del **Inicio de sesión de usuarios**, ya que solo un usuario autenticado podrá acceder a su plan.

6. RF2 - Acceso al plan de rehabilitación *refine* a RF3 - Visualizar historial de ejercicios.

- El acceso al historial de ejercicios refina el requisito de **Acceso al plan de rehabilitación**, proporcionando detalles adicionales sobre qué información específica se puede ver en los planes.

7. RF2 - Acceso al plan de rehabilitación *derivaReq* a RF5 - Registro de resultados y cumplimiento.

- El registro de resultados y cumplimiento deriva del **Acceso al plan de rehabilitación**, ya que los pacientes deben poder registrar su progreso relacionado con el plan.

8. RF3 - Visualizar historial de ejercicios *refine* a RF4 - Recordatorio de sesiones.

- El recordatorio de sesiones refina el **Historial de ejercicios**, ya que los recordatorios son una función adicional del sistema que ayuda a los pacientes a seguir sus planes de ejercicios.

9. RF4 - Recordatorio de sesiones *refine* a RNF4 - Los recordatorios tienen que estar siempre disponibles.

- Los recordatorios tienen que estar siempre disponibles refina el **Recordatorio de sesiones** al indicar que los recordatorios deben estar disponibles 24/7, asegurando que los pacientes reciban esta información en cualquier momento.
- 10. **RF5 - Registro de resultados y cumplimiento** *derivaReq* a **RF16 - Desbloqueo de logros por cumplimiento de ejercicios**.
 - El desbloqueo de logros se deriva del **Registro de resultados y cumplimiento**, ya que los logros solo se pueden desbloquear una vez que los resultados están registrados y se ha demostrado el cumplimiento del plan.
- 11. **RF5 - Registro de resultados y cumplimiento** *derivaReq* a **RF15 - Acceso a paneles interactivos del progreso**.
 - El acceso a paneles interactivos del progreso se deriva del **Registro de resultados**, ya que los datos registrados deben estar disponibles para su visualización en los paneles interactivos.
- 12. **RF7 - Acceder a información de pacientes** *derivaReq* a **RF9 - Crear planes de rehabilitación**.
 - La creación de planes de rehabilitación se deriva del **Acceso a la información de los pacientes**, ya que los profesionales necesitan estos datos para diseñar planes personalizados.
- 13. **RF9 - Crear planes de rehabilitación** *refine* a **RF10 - Actualizar planes de rehabilitación**.
 - La actualización de los planes refina la **Creación de planes**, especificando que los planes pueden y deben ser actualizados a lo largo del tiempo.
- 14. **RF6 - Agendar consultas** *refine* a **RNF5 - Sincronización con calendarios personales**.
 - **Sincronización con calendarios personales** refina la **Función de agendar consultas**, proporcionando más detalles sobre cómo deben integrarse estas consultas con los calendarios de los usuarios.
- 15. **RF13 - Envío de alertas** *derivaReq* a **RF14 - Comunicación entre paciente y profesional**.
 - La **Comunicación entre paciente y profesional** se deriva del **Envío de alertas**, ya que estas alertas pueden generar interacciones directas entre el paciente y el profesional.
- 16. **Proporcionar bases de datos de ejercicios y Recopilación de datos de dispositivos**
 - Estos requisitos están aislados y no tienen conexiones directas con otros requisitos, lo cual indica que son funcionalidades independientes dentro del sistema.

2.6. Diagrama de Clases

En el desarrollo de software orientado a objetos, el diagrama de clases es una herramienta esencial para representar la estructura del sistema. Este diagrama nos permite modelar las clases principales que intervienen en la aplicación, sus atributos y métodos, así como las relaciones entre ellas. El objetivo es tener una visión clara de cómo se organizan los datos y cómo interactúan los diferentes componentes en el sistema.

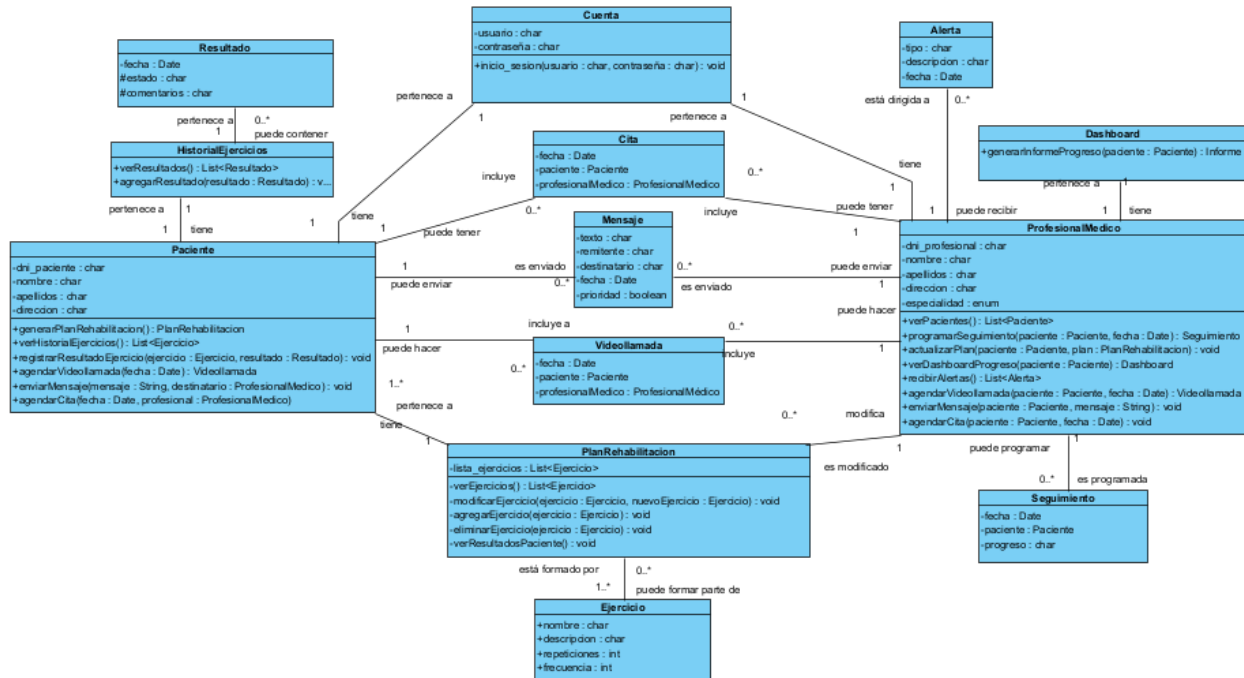


Figura 4: Diseño de Diagrama de clases

Clases y Métodos de la Aplicación de Rehabilitación

Paciente

Atributos: *dni_paciente*, *nombre*, *apellidos*, *direccion*.

Esta entidad mostrará las funciones a las que puede acceder desde la aplicación iniciando sesión como paciente.

- **generarPlanRehabilitacion() : PlanRehabilitacion** Genera un plan de rehabilitación personalizado para el paciente, que incluye los ejercicios y rutinas recomendadas para su recuperación.
- **verHistorialEjercicios() : List<Ejercicio>** Permite al paciente ver el historial de ejercicios que ha realizado o que se le han asignado, mostrando información como repeticiones, frecuencia, etc.
- **registrarResultadoEjercicio(ejercicio: Ejercicio, resultado: Resultado)** El paciente registra los resultados de un ejercicio específico, incluyendo si completó las repeticiones y cualquier otro dato relevante.
- **agendarVideollamada(fecha: Date) : Videollamada** Permite al paciente agendar una videollamada con un profesional médico en una fecha específica.
- **enviarMensaje(mensaje: String, destinatario: ProfesionalMedico)** El paciente puede enviar un mensaje al profesional médico dentro de la plataforma para resolver dudas o hacer consultas.
- **agendarCita(fecha: Date, profesional: ProfesionalMedico)** Permite al paciente agendar una cita presencial o virtual con un profesional médico en una fecha específica.

ProfesionalMedico

Atributos: *dni_profesional, nombre, apellidos, direccion, especialidad.*

Esta entidad mostrará las funciones a las que puede acceder desde la aplicación iniciando sesión como profesional médico.

- **verPacientes() : List<Paciente>** Muestra una lista de los pacientes registrados en la aplicación, con acceso a sus planes de rehabilitación y progreso.
- **programarSeguimiento(paciente: Paciente, fecha: Date) : Seguimiento** Permite al profesional médico programar una sesión de seguimiento con un paciente para revisar su progreso.
- **actualizarPlan(paciente: Paciente, plan: PlanRehabilitacion)** Actualiza el plan de rehabilitación de un paciente, permitiendo modificar los ejercicios o las rutinas según sea necesario.
- **verDashboardProgreso(paciente: Paciente) : Dashboard** Accede a un tablero de control con el progreso del paciente, como resultados de ejercicios, cumplimiento de metas y otros indicadores.
- **recibirAlertas() : List<Alerta>** Permite al profesional recibir alertas relacionadas con sus pacientes, como recordatorios o notificaciones sobre problemas en el progreso.
- **agendarVideollamada(paciente: Paciente, fecha: Date) : Videollamada** Permite al profesional médico agendar una videollamada con un paciente en una fecha específica.
- **enviarMensaje(paciente: Paciente, mensaje: String)** El profesional médico puede enviar mensajes a sus pacientes para dar seguimiento o brindar instrucciones adicionales.
- **agendarCita(paciente: Paciente, fecha: Date)** El profesional médico puede agendar citas para una consulta presencial o virtual con el paciente en una fecha específica.

PlanRehabilitacion

Atributos: *lista_ejercicios.*

- **verEjercicios() : List<Ejercicio>** Muestra una lista de todos los ejercicios que forman parte del plan de rehabilitación.
- **modificarEjercicio(ejercicio: Ejercicio, nuevoEjercicio: Ejercicio)** Permite modificar un ejercicio en el plan de rehabilitación, cambiando sus características como repeticiones, frecuencia, etc.
- **agregarEjercicio(ejercicio: Ejercicio)** Añade un nuevo ejercicio al plan de rehabilitación del paciente.
- **eliminarEjercicio(ejercicio: Ejercicio)** Elimina un ejercicio del plan de rehabilitación del paciente.

Cuenta

Atributos: *lista_ejercicios.* Representa la entidad encargada de gestionar la autenticación de los usuarios, tanto pacientes como profesionales médicos, en la aplicación.

- **inicio_sesion(usuario : char, contraseña : char) : void** Este método verifica si las credenciales proporcionadas (usuario y contraseña) son correctas y permiten el acceso al sistema. Si las credenciales coinciden con las almacenadas, se concede el acceso al usuario. De lo contrario, se rechaza la autenticación y se notificará al usuario que las credenciales son incorrectas.

Ejercicio

Atributos: *nombre, descripcion, repeticiones, frecuencia.* Los ejercicios contienen información relevante sobre las repeticiones, frecuencia y una descripción detallada del movimiento o acción a realizar.

HistorialEjercicios

- **verResultados()** : **List<Resultado>** Muestra los resultados obtenidos por el paciente en los ejercicios realizados, con detalle del estado y comentarios.
- **agregarResultado(resultado: Resultado)** Agrega un nuevo resultado al historial de ejercicios, almacenando detalles sobre la fecha y el estado del ejercicio.

Resultado

Atributos: *fecha, estado, comentarios*. Guarda información sobre el desempeño de un paciente en un ejercicio específico, incluyendo la fecha en que se realizó, el estado del ejercicio (completado o incompleto) y cualquier comentario adicional.

Videollamada

Atributos: *fecha, paciente, profesionalMedico*. Almacena la información de una videollamada programada, como la fecha y los participantes (paciente y profesional médico).

Mensaje

Atributos: *texto, remitente, destinatario, fecha, prioridad*. Un mensaje enviado entre un paciente y un profesional médico, con detalles sobre el contenido, remitente, destinatario, la fecha de envío y si es prioritario o no.

Cita

Atributos: *fecha, paciente, profesionalMedico*. Guarda la información de una cita programada, ya sea presencial o virtual, entre un paciente y un profesional médico.

Seguimiento

Atributos: *fecha, paciente, progreso*. Registra los detalles de una sesión de seguimiento, incluyendo la fecha, el paciente y el progreso discutido o observado durante la sesión.

Dashboard

- **generarInformeProgreso(paciente: Paciente)** : **Informe** Genera un informe detallado del progreso del paciente, basado en sus resultados, ejercicios realizados y cumplimiento de objetivos.

Alerta

Atributos: *tipo, descripcion, fecha*. Contiene información sobre una alerta que puede estar relacionada con citas próximas, tareas incompletas o actualizaciones en el plan de rehabilitación.

Clases y Relaciones con Multiplicidades

- **Paciente**
 - **PlanRehabilitacion** (1) — Un paciente tiene un plan de rehabilitación.
 - **HistorialEjercicios** (1) — Un paciente tiene un historial de ejercicios.
 - **Cuenta** (1) — Un paciente tiene una cuenta.
 - **Videollamada** (0..*) — Un paciente puede tener cero o muchas videollamadas.
 - **Mensaje** (0..*) — Un paciente puede enviar o recibir cero o muchos mensajes.

- **Cita** (0..*) — Un paciente puede tener cero o muchas citas agendadas con un profesional médico.
- **Cuenta**
 - **Paciente** (1) — Una cuenta pertenece a un paciente.
 - **ProfesionalMedico** (1) — Una cuenta pertenece a un profesional médico.
- **ProfesionalMedico**
 - **Paciente** (1) — Un profesional médico puede atender a cero o muchos pacientes.
 - **Cuenta** (1) — Un profesional médico tiene una cuenta.
 - **PlanRehabilitacion** (0..*) — Un profesional médico puede modificar los planes de rehabilitación de cero o muchos pacientes.
 - **Videollamada** (0..*) — Un profesional médico puede tener cero o muchas videollamadas con pacientes.
 - **Mensaje** (0..*) — Un profesional médico puede enviar o recibir cero o muchos mensajes.
 - **Cita** (0..*) — Un profesional médico puede tener cero o muchas citas agendadas con pacientes.
 - **Seguimiento** (0..*) — Un profesional médico puede programar cero o muchas sesiones de seguimiento para pacientes.
 - **Dashboard** (1) — Un profesional médico tiene un dashboard de progreso para cada paciente que sigue.
 - **Alerta** (0..*) — Un profesional médico puede recibir cero o muchas alertas relacionadas con los pacientes.
- **PlanRehabilitacion**
 - **Ejercicio** (1..*) — Un plan de rehabilitación debe tener al menos un ejercicio.
 - **Paciente** (1) — Un plan de rehabilitación pertenece a exactamente un paciente.
- **Ejercicio**
 - **PlanRehabilitacion** (1) — Cada ejercicio pertenece a exactamente un plan de rehabilitación.
- **HistorialEjercicios**
 - **Resultado** (0..*) — Un historial de ejercicios puede contener cero o muchos resultados de los ejercicios realizados.
 - **Paciente** (1) — Un historial de ejercicios pertenece a exactamente un paciente.
- **Resultado**
 - **Ejercicio** (1) — Un resultado pertenece a exactamente un ejercicio.
 - **HistorialEjercicios** (1) — Un resultado pertenece a exactamente un historial de ejercicios.
- **Videollamada**
 - **Paciente** (1) — Cada videollamada incluye a exactamente un paciente.
 - **ProfesionalMedico** (1) — Cada videollamada incluye a exactamente un profesional médico.
- **Mensaje**
 - **Paciente** (1) — Cada mensaje es enviado o recibido por exactamente un paciente.

- **ProfesionalMedico** (1) — Cada mensaje es enviado o recibido por exactamente un profesional médico.
- **Cita**
 - **Paciente** (1) — Cada cita incluye a exactamente un paciente.
 - **ProfesionalMedico** (1) — Cada cita incluye a exactamente un profesional médico.
- **Seguimiento**
 - **Paciente** (1) — Cada seguimiento está asociado a exactamente un paciente.
 - **ProfesionalMedico** (1) — Cada seguimiento está asociado a exactamente un profesional médico.
- **Dashboard**
 - **Paciente** (1) — Un dashboard de progreso está asociado a exactamente un paciente.
 - **ProfesionalMedico** (1) — Un dashboard de progreso pertenece a exactamente un profesional médico.
- **Alerta**
 - **ProfesionalMedico** (1) — Cada alerta está dirigida a exactamente un profesional médico.