Capítulo 4 – Diseño

En este capítulo se describe el diseño general del sistema propuesto, abordando planificación metodológica y aspectos técnicos relevantes, con el fin de proporcionar una visión clara y organizada de cómo se construye el proyecto.

Se describe la metodología utilizada, los requerimientos software y hardware que se necesitan para el desarrollo y la arquitectura del sistema. Además, se incluyen diagramas de casos de uso, de flujo y de secuencia para representar de manera estructurada los procesos y las interacciones entre componentes y actores.

4.1 Metodología

Para un desarrollo como el que se plantea en este trabajo, es importante tener clara cierta estructura metodológica que sea de ayuda a la hora de organizar las tareas y mejorar la eficiencia del proceso, así como facilitar el análisis y la documentación. Contar con una planificación estructurada es recomendable para entender el proyecto, especialmente cuando se trata de integrar varias tecnologías como son el desarrollo móvil, web y servicios en la nube.

Para este proyecto, se plantea un enfoque general basado en una metodología en cascada, que se caracteriza por la división del trabajo en fases secuenciales y definidas, que en este caso serán el análisis de requisitos, diseño, implementación, pruebas y mantenimiento. Cabe destacar que el mantenimiento se presenta como una etapa algo diferente a las demás, al empezar cuando termina el proyecto y ser un proceso continuo y de duración indeterminada.

Aunque en la práctica estas etapas se puedan solapar puntualmente, seguir esta guía es una buena forma de construir el proyecto de manera ordenada.

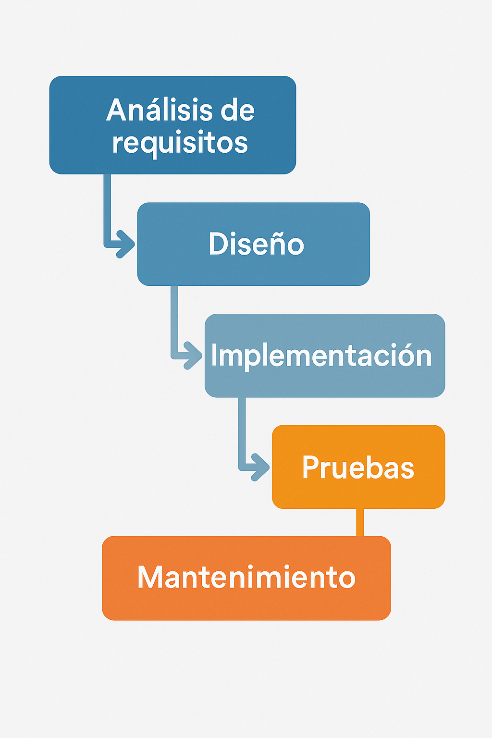


Figura 1 - Metodología en cascada

4.2 Requerimientos Software

En esta sección se desarrollan cuáles son los requerimientos software que el sistema necesita para su funcionamiento, tanto para el desarrollo como para su uso una vez haya una versión final.

4.2.1 Entorno de desarrollo móvil

En cuanto al desarrollo en Android, el IDE que proporciona la comunicación nativa con los dispositivos es Android Studio, además de ser el IDE de referencia y el más utilizado a día de hoy, por lo que será el utilizado en este proyecto. Android Studio ofrece una completa integración con los SDK y buenas herramientas de depuración, además de ofrecer emuladores Android, que en etapas tempranas del trabajo son una herramienta muy útil.

Este entorno permite utilizar dos lenguajes: Java y Kotlin, y se ha decidido usar el primero dada la experiencia previa con este lenguaje, que agiliza el proceso y reduce la curva de aprendizaje.

4.2.1 Entorno de desarrollo web

En cuando al entorno web elegido para el desarrollo web del proyecto, será Visual Studio Code (VSC). Este entorno es el más popular entre los desarrolladores web, entre otras cosas, por su ligereza, ya que no es un IDE, sino un editor de texto. Esto hace que consuma menos recursos y, en términos generales, sea un desarrollo más fluido. Por otra parte, VSC cuenta con una gran cantidad de extensiones, que hacen este entorno una herramienta muy versátil y adaptable en muchos casos al tipo de proyecto que se vaya a llevar a cabo.

Es ampliamente conocido su uso con Angular, al permitir trabajar de forma ágil con las tecnologías que se necesitarán como TypeScript, HTML y CSS

4.2.2 Control de versiones

Hoy en día el control de versiones en desarrollos que implican escritura de código es una práctica muy recomendable. Desde el lanzamiento de Git en 2005 por Linus Torlvars, Git se ha convertido en la herramienta de este tipo más utilizada en el mundo.

A raíz de Git surgen diferentes plataformas como Bitbucket, Gitlab o GitHub, en este proyecto será este último el programa de control de versiones utilizado. Github cuenta con una interfaz web intuitiva que gestiona los repositorios Git, así como funcionalidades como el control de ramas, pull request o documentación integrada con los README.

Resulta de interés en este proyecto tanto para la colaboración y revisión de los tutores del trabajo, como también para mantener el control de las versiones con los cambios guardados, pudiendo deshacerlos o volviendo a un commit anterior por alguna necesidad durante el desarrollo.

Además, GitHub cuenta con una nube donde se puede guardar el código y sus cambios, pudiendo acceder a él de manera segura desde diferentes dispositivos.

4.3 Requerimientos Hardware

En esta sección se describen los componentes hardware necesarios para el correcto funcionamiento del sistema, en base a los requisitos funcionales detallados en el capítulo anterior.

4.3.1 Smartphone

Para el registro de pacientes, la realización de las pruebas y la comunicación con los sensores será necesario un teléfono inteligente con conectividad a Internet y Bluetooth. No se requiere un dispositivo de alta gama, ya que el procesamiento y los cálculos que se realizan no exigen un alto rendimiento. Sin embargo, es importante que el dispositivo sea compatible con la versión mínima de Android definida en el desarrollo (Android 8.0 o superior) y cuente con los recursos básicos necesarios para ejecutar la aplicación sin problemas.

En la *tabla 1* se detallan los requisitos mínimos recomendados para el uso del sistema por parte del personal facultativo.

|  |  |
| --- | --- |
| **Requisito** | **Valor mínimo** |
| Sistema operativo | Android 8.0 (Oreo) o superior |
| Conectividad | Bluetooth 4.0 o superior, WiFi o 4G/5G |
| Memoria RAM | 2 GB |
| Almacenamiento interno | 16 GB disponibles |
| Resolución de pantalla | 720p o superior |
| Acceso a cámara | No requerido |

Tabla 1 - Requisitos mínimos smartphone

Cualquier teléfono que cumpla o supere estas especificaciones podrá ser utilizado por el facultativo encargado de realizar las pruebas.

Durante el desarrollo del proyecto se ha utilizado el dispositivo Xiaomi Redmi Note 13 Pro+ 5G, cuyas características más relevantes se detallan en la *tabla 2.*

|  |  |
| --- | --- |
| **Característica** | **Especificación** |
| Sistema operativo | Android 13 |
| Conectividad | Bluetooth 5.3, 5G, WiFi 6 |
| Memoria RAM | 8 GB |
| Almacenamiento interno | 256 GB |
| Procesador | MediaTek Dimensity 7200-Ultra |
| Pantalla | 6,67” AMOLED, 120 Hz |

Tabla 2 - Smartphone utilizado

Este dispositivo ha sido utilizado tanto para el desarrollo como para la validación de la aplicación móvil en condiciones reales.

4.3.2 Computador

Será necesario el uso de un PC tanto para todo el proceso de desarrollo como para la visualización de la web. En este caso se ha utilizado uno cuyas características se exponen en la *tabla 3.*

|  |  |
| --- | --- |
| **Característica** | **Especificación** |
| Sistema operativo | Windows 11 |
| Tarjeta gráfica | NVIDIA GeForce RTX 3050 |
| Memoria RAM | 32 GB |
| Disco duro | SSD 500 GB |
| Procesador | AMD Ryzen 9 5900X 12-Core |

Tabla 3 - PC utilizado

4.3.3 Otros dispositivos

Además del dispositivo móvil y el PC, el sistema se apoya en tres sensores específicos: una plantilla de presión, un giróscopo y un sensor de distancia. Estos dispositivos transmiten los datos en tiempo real a través de Bluetooth, y aunque no forman parte del hardware desarrollado en este proyecto, son imprescindibles para la ejecución de las pruebas SPPB. Su compatibilidad con Android mediante un protocolo estándar ha sido fundamental para su integración.

4.4 Arquitectura del sistema

En cuanto a la arquitectura del sistema, se plantea una estructura en cuatro capas, con lo que se busca separar responsabilidades, organizar el código y una mejor escalabilidad y mantenimiento posterior.

Con esta estructura, se diferencia entre la capa de presentación, procesamiento, comunicación y datos.

El sistema está compuesto por dos clientes (aplicación web desarrollada en Angular y aplicación móvil en Android) que tienen comunicación con plataforma en la nube (Firebase)

Por otro lado, el sistema también tiene comunicación con los tres sensores mencionados anteriormente, que recogen los datos y que los transmiten en tiempo real al dispositivo móvil vía Bluetooth.

A continuación, se detallan las capas que forman el sistema.

4.4.1 Capa de presentación

Esta se puede definir como la capa que el usuario ve. Se compone de las interfaces de usuario y es lo que permite la interacción con el sistema. Por un lado, Android permite registrar pacientes, conectar con los sensores, recibir sus datos en crudo y gestionar las pruebas. Por otro lado, la aplicación web, se orienta principalmente a la visualización, permitiendo consultar datos de pacientes, históricos, pruebas, etc.

4.4.2 Capa de procesamiento

En esta capa ocurre toda la lógica de negocio. Esta se encuentre distribuida entre la web y la app Android. Esto incluye la interpretación de las señales recibidas por los dispositivos, el procesamiento de los datos y cálculos o asignación de puntuaciones en la parte de la aplicación Android. En cuanto a la web, incluye lógicas como validaciones, filtrados, o controles de acceso por rol.

Esta capa, además, coordina la comunicación con Firebase y se encarga de que los datos cumplan con la estructura prevista.

4.4.3 Capa de datos

Es la capa donde se encuentran los datos, en este caso está centralizada en Firebase. Ofrece los servicios de almacenamiento en la nube y sincronización en tiempo real entre las dos aplicaciones.

Firebase permite operaciones de escritura a la aplicación Android, que serán el registro de usuarios y las pruebas subidas. Para el caso de la aplicación Angular, se permitirán tanto operaciones de escritura (eliminar y editar paciente) como operaciones de lectura, ya que necesitará los datos de pacientes y pruebas para su representación.

4.4.4 Capa de transporte

En cuanto a la comunicación entre los dispositivos, se utilizan redes seguras sobre Internet, utilizando conexiones móviles (4G/5G) o WiFi. Esta capa, aunque se gestiona de forma implícita por el resto, es importante para garantizar una comunicación continua entre los dispositivos, que es fundamental en este sistema para su correcto funcionamiento.

Por otro lado, la comunicación entre los sensores y la aplicación Android se realizará por Bluetooth, estableciéndose una conexión directa y en tiempo real.

4.5 Modelado casos de uso

Los diagramas de casos de uso son una herramienta esencial en el contexto del desarrollo de software, permitiendo dar una perspectiva clara e intuitiva de las funcionalidades principales del sistema. Muestran las interacciones que hay entre los actores y el sistema, pero evitando los detalles técnicos.

Son especialmente útiles para definir los requisitos funcionales y visualizar de manera más clara la lógica de negocio.

En esta sección se representan los diagramas de casos de uso de las dos partes principales del proyecto: la aplicación Android y la web.

4.5.1 Diagrama casos de uso de aplicación Android

En la *figura 2* se muestra como para esta parte del sistema solo hay un actor: el facultativo. Este será el único que tenga acceso a la aplicación y el único que podrá hacer las acciones principales:

* Registrar paciente
* Buscar paciente

En la primera, se pide introducir unos datos del paciente, y al completar el registro de la opción de hacer pruebas al paciente, lo que se representa en el diagrama con una línea discontinua (extend) que refleja que esta secuencia es opcional.

Por otro lado, al ‘buscar paciente’ se puede elegir entre las tres pruebas SPPB, redirigiendo al usuario a las pantallas en las que se realizan las pruebas.

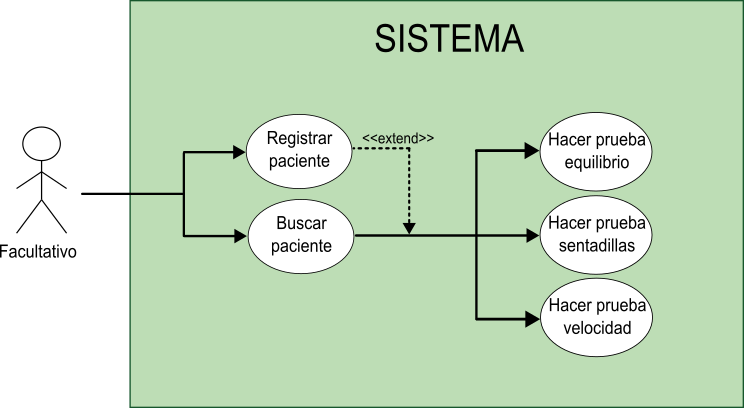


Figura 2 - Diagrama de casos de uso de la aplicación Android

4.5.2 Diagramas casos de uso de la web

En este sistema se contemplan tres tipos de actores con acceso a la aplicación web:

* Paciente *(figura 3)*
* Facultativo *(figura 4)*
* Administrador*(figura 5)*

Cada uno de ellos dispone de un conjunto de funcionalidades específico, adaptado a su rol y nivel de permisos dentro del sistema.

El paciente es el actor con menor nivel de acceso. Al iniciar sesión con su rol, la mayoría de las funcionalidades de la web permanecen bloqueadas. Únicamente puede consultar sus propios datos personales, acceder a la información de sus pruebas y visualizar el histórico, donde se presenta un resumen de todas las pruebas realizadas hasta la fecha. En cualquier momento, el paciente puede cerrar sesión.

El actor facultativo dispone de un conjunto más amplio de funcionalidades. Además de heredar todos los permisos del paciente, puede consultar un listado con todos los pacientes registrados en el sistema, accediendo a sus datos y pruebas individuales. También tiene habilitada la opción de descargar los datos en formato Excel para su tratamiento externo. A diferencia de los otros perfiles, el facultativo es el único actor que puede iniciar su propio registro en la web. Sin embargo, tras el registro, el sistema le asigna inicialmente el rol no-role, lo cual le impide iniciar sesión hasta que un administrador le conceda un rol válido (facultativo o administrador).

El administrador es el actor con el mayor nivel de privilegios dentro de la aplicación web. Dispone de acceso a todas las funcionalidades del sistema, incluyendo las disponibles para facultativos y pacientes. Además, puede administrar usuarios, accediendo a un listado completo de facultativos registrados. Desde ahí, puede asignar roles (por ejemplo, validar el acceso de un nuevo facultativo) y también eliminar usuarios, acción que afecta únicamente a los facultativos. Si un usuario es eliminado, deberá registrarse de nuevo para recuperar el acceso. Además, el administrador tiene la capacidad de editar los datos de los pacientes y eliminarlos desde el listado correspondiente.

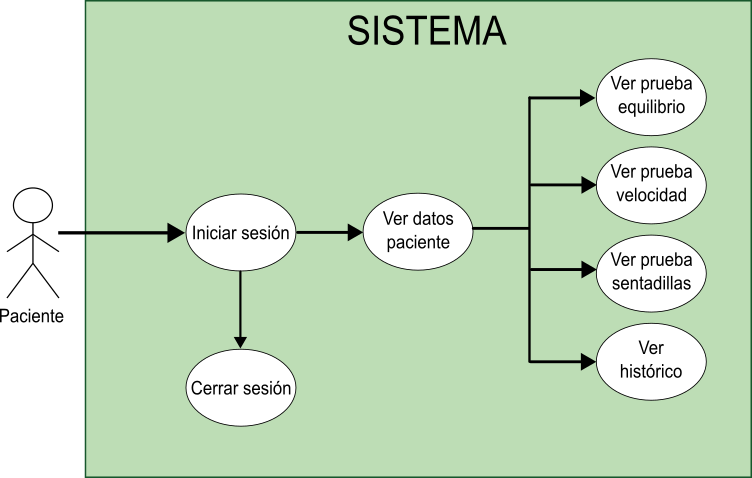


Figura 3 - Diagrama de casos de uso de la web (paciente)

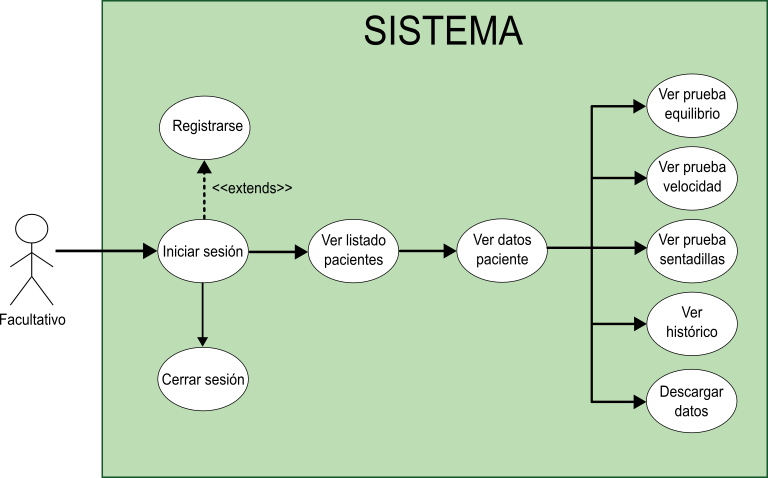


Figura 4 - Diagrama de casos de uso de la web (facultativo)

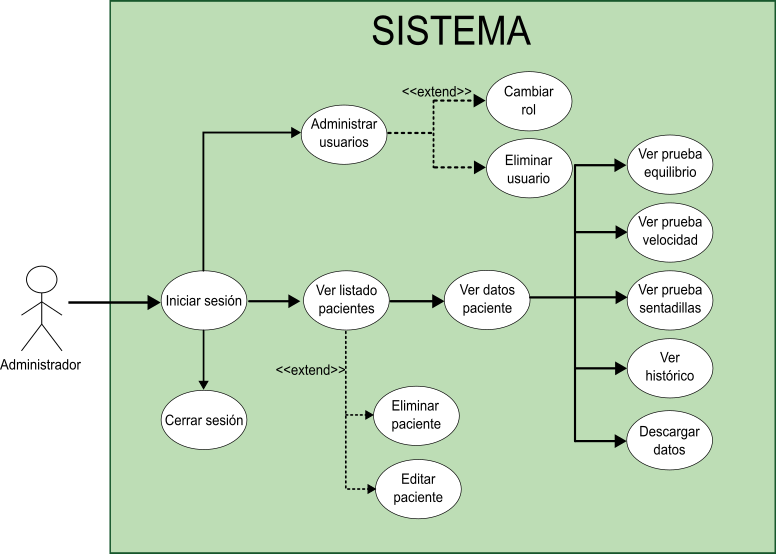


Figura 5 - Diagrama de casos de uso de la web (administrador)

4.6 Diagramas de flujo

En esta sección se representan los diagramas de flujo del sistema. Estos representan de forma secuencial los pasos que sigue la aplicación, ayudando a entender las decisiones y las operaciones que se realizan. De la misma forma que en el anterior apartado, se representa tanto el diagrama de la app móvil como de la web.

4.6.1 Diagrama de flujo de la aplicación Android

En la *figura 6* se representa en primer lugar cómo el usuario que inicia la aplicación tiene la opción de registrar a un paciente o buscarlo (introduciendo su ID) para hacer las pruebas. En el caso de que el registro haya sido exitoso, introduciendo datos válidos, estos se suben a la base de datos en la nube (representado con el logo de Firebase en la *figura 6*) y se dirige a la realización de pruebas de ese paciente que se acaba de registrar, de la misma manera que si es con la búsqueda de paciente.

En cuanto a la prueba de equilibrio, empieza con la parte de equilibrio en paralelo. Se realiza la prueba y, en caso de que se haya superado, se pasa a la siguiente. Si no se ha superado, se vuelve a empezar. La prueba de equilibrio semi-tándem funciona de la misma manera, en caso de no superar la prueba se repetiría esta última. La prueba de equilibrio tándem no es necesario haberla superado para poder subir los datos a Firebase.

Por otro lado, la prueba de sentadillas se realiza entera de una vez. En caso de no haber conseguido alcanzar las sentadillas previstas, se repite la prueba. Si se ha superado, se envían los datos de la prueba a Firebase.

Por último, la prueba de velocidad de la marcha. En esta prueba se realiza la ida y la vuelta. Si no se ha alcanzado la distancia mínima en la ida, se repite. Si se ha alcanzado, se realiza la vuelta. En este caso, si no se ha superado la vuelta, debido a limitaciones que se tratarán en otro apartado de este proyecto, el paciente debe volver a realizar la prueba desde el principio. Una vez superada la ida y la vuelta, los datos se suben a Firebase.

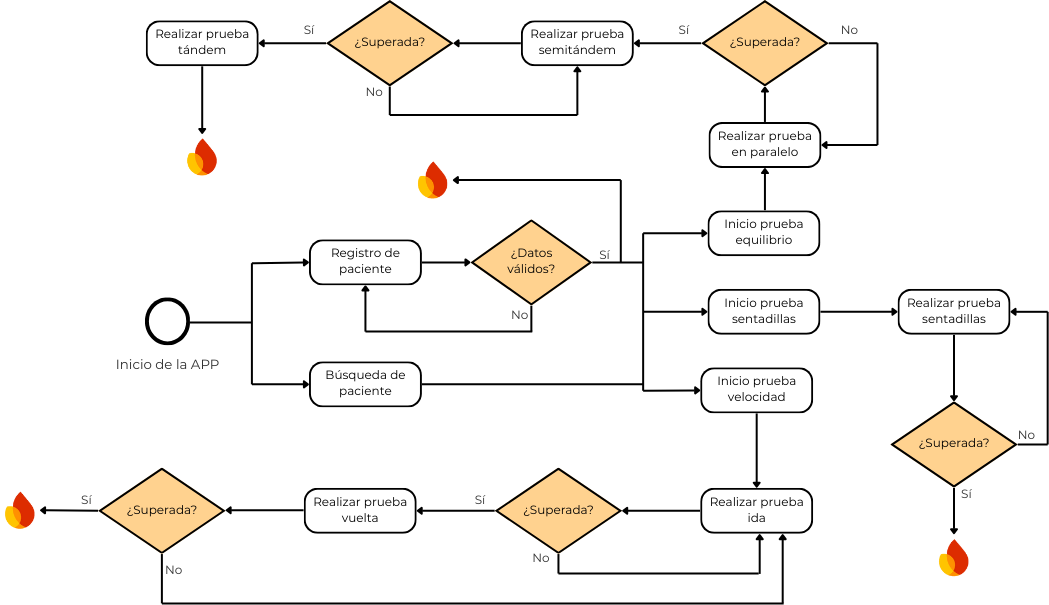


Figura 6 - Diagrama de flujo app móvil

4.6.2 Diagrama de flujo de la web

En la *figura 7* se muestra, en primer lugar, cómo para acceder a cualquier parte de la web es necesario pasar por un inicio de sesión.

Si el usuario no es un paciente (es facultativo), en caso de que sea la primera vez debe pasar por un registro. Si no es la primera vez (ya se ha registrado y el administrador le ha dado permiso modificando su rol) puede iniciar sesión e ir al perfil de usuario. Desde ahí, si es el administrador del sistema podrá acceder a administrar usuarios, donde podrá cambiar el rol a los usuarios que se han registrado en la web. Tanto si es facultativo como si es el administrador, pueden acceder desde el perfil de usuario al listado de los pacientes. En esta parte, solo el administrador tendrá acceso a editar los datos del paciente.

Desde el listado de pacientes se puede acceder al perfil del paciente, donde estarán sus datos y sus pruebas realizadas. Este es el punto donde accede el paciente cuando inicia sesión por su vía. Desde el perfil de paciente se pueden ver sus pruebas individuales y el histórico con todos los datos de las pruebas realizadas.

Solo el facultativo y el administrador tienen acceso a descargar los datos del paciente.

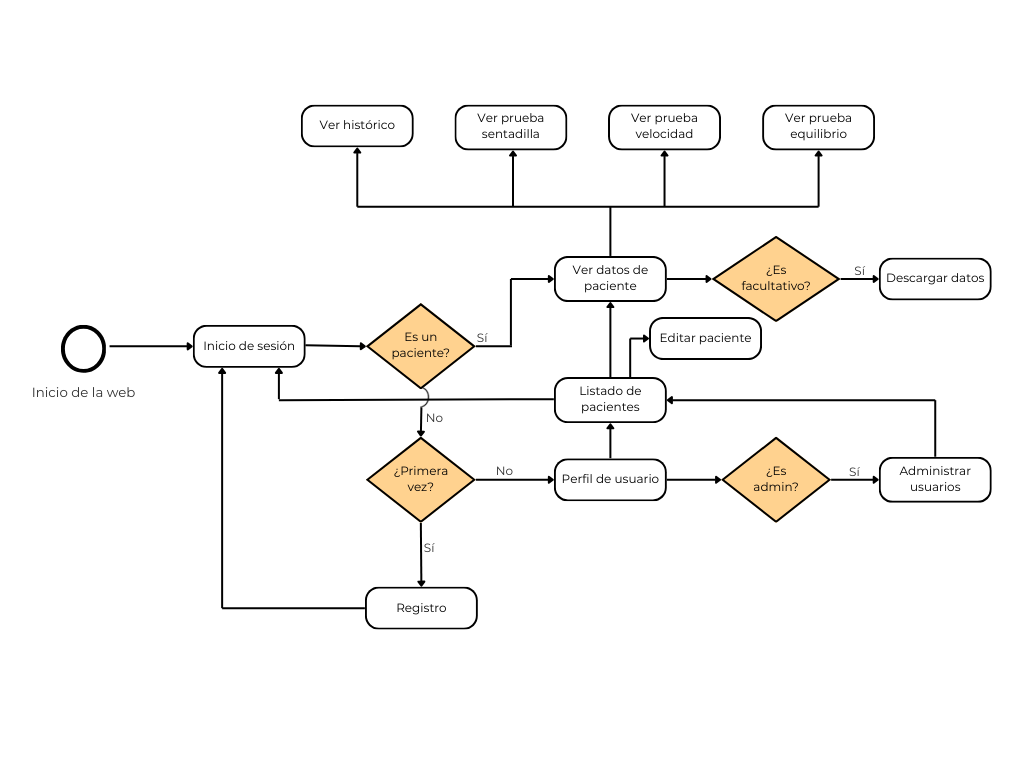


Figura 7 - diagrama de flujo de la web

4.7 Diagramas de secuencia

Para complementar los diagramas vistos, en esta sección se presentan los diagramas de secuencia del sistema. Estos diagramas muestran cómo interactúan los componentes y los actores del proyecto a lo largo del tiempo. Representan el orden de las llamadas entre los objetos y son útiles para visualizar un flujo de control y comunicación de escenarios concretos, por lo que se hace uno por cada uno de los escenarios principales.

* **Registro de paciente**: La *figura 8* representa cómo se registra un paciente en el sistema desde la aplicación Android.

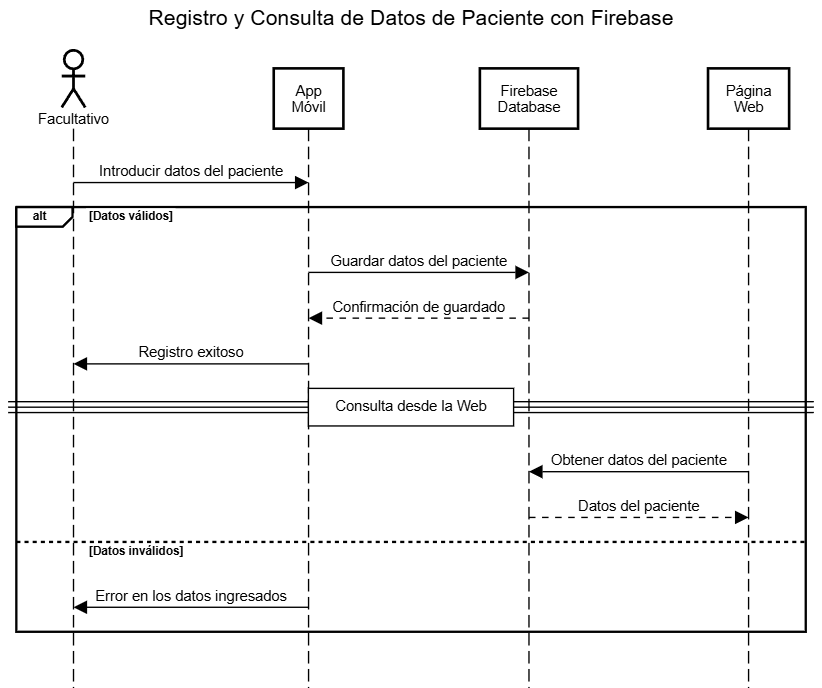


Figura 8 - Diagrama de secuencia registro de paciente

* **Realización de pruebas (genérico):** La *figura 9* representa el modo en el que interactúa el móvil, los dispositivos y Firebase cuando se realizan las pruebas SPPB.

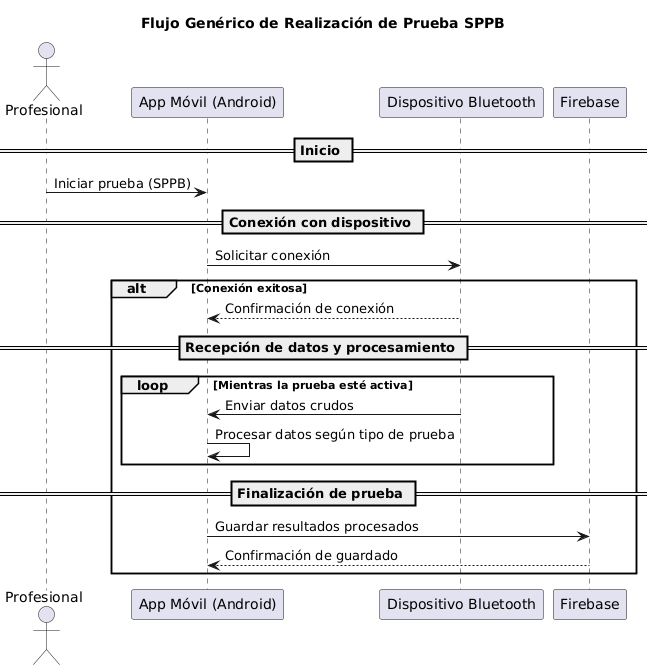


Figura 9 - Diagrama de secuencia realización de pruebas (genérico)

* **Consulta y descarga de pruebas**: La *figura 10* representa cómo interactúan los tres sistemas desde que se realiza una prueba hasta que se descargan los datos, pasando por cómo es la consulta de los datos por parte del paciente y del facultativo.

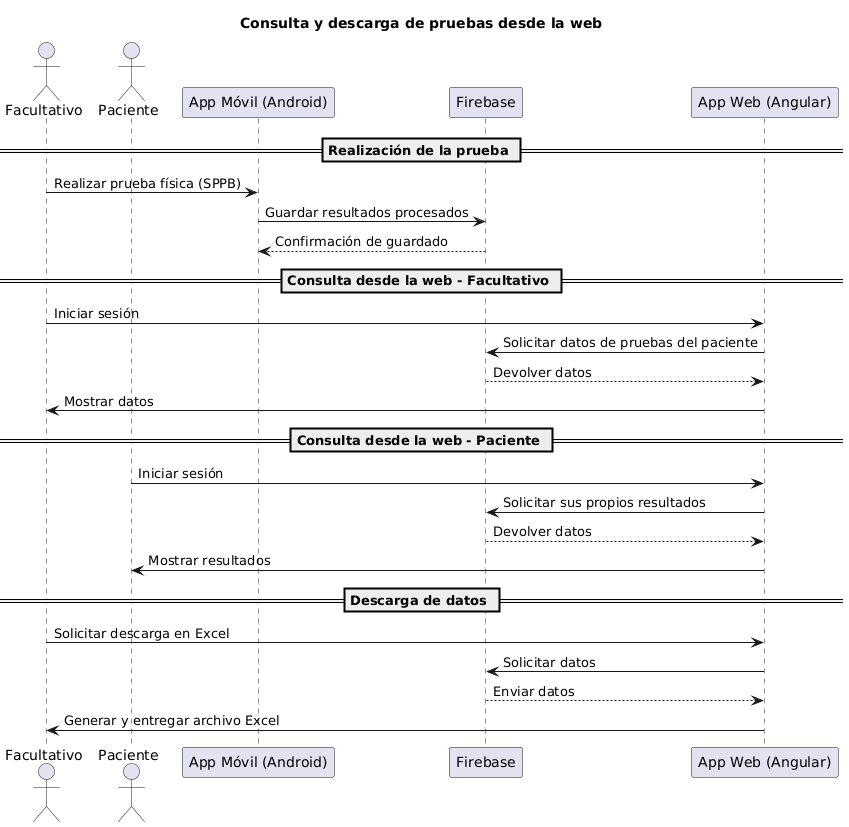


Figura 10 - Diagrama de secuencia consulta y descarga de pruebas

* **Gestión de los roles de usuario**: La *figura 11* representa cómo se gestionan los roles de los facultativos por parte del administrador del sistema y cómo este puede iniciar sesión.

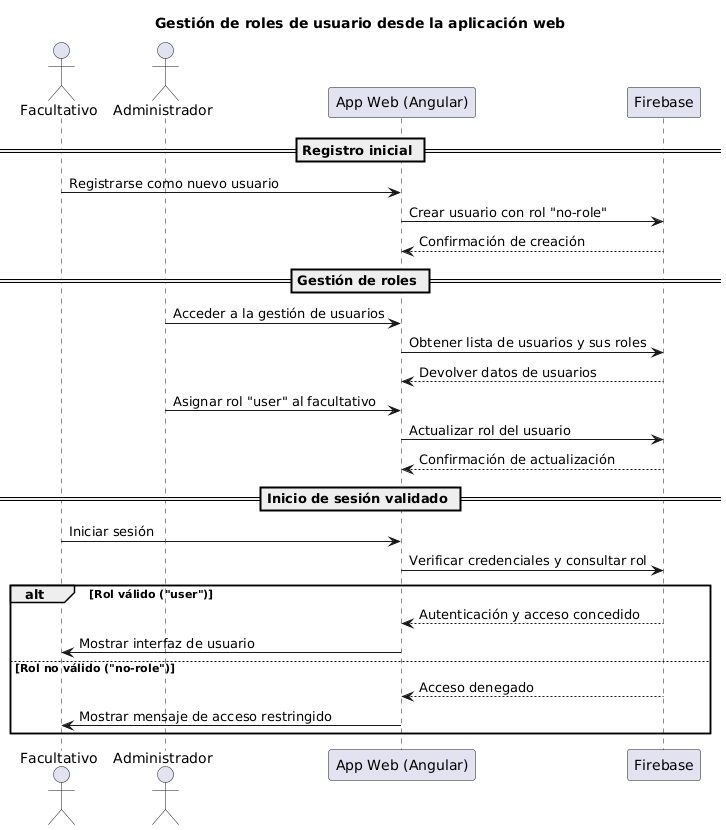


Figura 11 - Diagrama de secuencia gestión de los roles de usuario