

SmartCoach: Aplicación Móvil para el Entrenamiento Físico Personalizado

Especificación de Requerimientos de Software (SRS)

Autores

Johanna Lisette Bolívar Calderón

Ingeniería Sistemas

bolivarjo@javeriana.edu.co

Maria Kamila Obregón Ortega

Ingeniería Sistemas

m-obregon@javeriana.edu.co

Director

Ing. Victor Hugo Vela Pinzón, MBIT, MBA.

v.vela@javeriana.edu.co



Pontificia Universidad
JAVERIANA
Colombia

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS

BOGOTÁ, D.C. 2023

Tabla de contenido

Lista de Tablas	4
Lista de Imágenes	5
Introducción	6
Propósito.....	6
Alcance.....	6
Definiciones, Acrónimos, Abreviaciones.....	9
Apreciación Global	11
Descripción Global	12
Perspectiva del Producto	12
Suposiciones.....	17
Dependencias	18
Restricciones	18
Funciones del producto y Operaciones	19
Modelo del Dominio.....	20
Características del Usuario.....	22
Características del Producto de Software.....	25
Plan de manejo de requerimientos	26
Levantamiento de Requerimientos.....	26
Verificación y Validación de requerimientos.....	27
Tipos de Requerimientos.....	28

Atributos de los Requerimientos	29
Mecanismo de Control de Estado de los Requerimientos.....	30
Mecanismos de Priorización de los Requerimientos.....	31
Requerimientos Específicos.....	33
Requerimientos Funcionales	33
Requerimientos No Funcionales	33
Trazabilidad	33
Requerimientos Funcionales	33
Requerimientos No Funcionales	34
Referencias.....	34
Anexos	36

Lista de Tablas

Tabla 1 Acrónimos y nombres	11
Tabla 2 Descripción restricciones	19
Tabla 3 Características del usuario cliente.....	23
Tabla 4 Características usuario administrador	25
Tabla 5 Tipo de requerimientos	29
Tabla 6 Estados de los Requerimientos funcionales.....	31
Tabla 7 Relación entre Requerimientos funcionales, Casos de Uso y Casos de Prueba	33
Tabla 8 Relación entre Requerimientos no funcionales, Elementos de diseño asociados y Escenarios de Calidad	34

Lista de Imágenes

Imagen 1 Modelo de dominio	21
Imagen 2 Detalles de las funcionalidades	25
Imagen 3 Detalle funcionalidad “Ver mapa”	25
Imagen 4 Detalle funcionalidad "Crear rutina"	26
Imagen 5 Detalle funcionalidad "Ubicar equipo o elemento en mapa"	26
Imagen 6 Escala de priorización de los requisitos funcionales.....	32

Introducción

El presente documento tiene como objetivo principal detallar los requisitos esenciales del proyecto *SmartCoach*, enfocándose en definir su propósito, alcance y visión global, a través de la definición clara y coherente de los requisitos funcionales y no funcionales, este documento proporcionará una base sólida para el diseño, desarrollo y evaluación del sistema *SmartCoach*. Este proyecto surge como una respuesta innovadora a la creciente integración de la tecnología en la vida diaria de las personas y la importancia de la salud y el entrenamiento físico.

Propósito

El propósito central de este documento es brindar al lector una descripción detallada de los requisitos específicos de software que juegan un papel fundamental al proporcionar una visión integral del comportamiento proyectado de la aplicación *SmartCoach*.

Esta aplicación móvil de entrenamiento físico personalizado está diseñada para satisfacer las necesidades de individuos entre 18 y 30 años que buscan adoptar hábitos saludables de ejercicio físico. Estos usuarios se benefician de una guía personalizada en sus rutinas de entrenamiento teniendo en cuenta variables como sus limitaciones físicas, metas individuales e implementos con los que cuenta para realizar su entrenamiento.

Además, la aplicación ofrece un valor añadido con servicios dirigidos a los administradores de gimnasios, proporcionándoles una herramienta para facilitar la interacción de los nuevos y antiguos clientes con los elementos disponibles en las instalaciones, mediante la recomendación de rutinas de ejercicio que promuevan un uso integral y óptimo de los recursos deportivos presentes en el gimnasio. Este enfoque no solo enriquece la experiencia de los usuarios, sino que también maximiza los beneficios que los clientes pueden obtener con su membresía.

En el transcurso de este documento se proporcionará un análisis detallado de los requisitos funcionales y no funcionales de la aplicación móvil *SmartCoach*, proporcionando una visión clara y completa de como la aplicación cumplirá con las expectativas y necesidades de sus usuarios finales. A través de esta especificación de requisitos, se establecerá la base esencial para la planificación, desarrollo y evaluación exitosa del proyecto.

Alcance

SmartCoach es una aplicación móvil disponible para dispositivos Android, creada con el propósito de potenciar la salud de los usuarios a través del ejercicio físico. Aunque está orientada

principalmente a aquellos que desean iniciar una rutina de ejercicio, su alcance no se limita solo a principiantes. La aplicación actúa como un entrenador personal, optimizando las rutinas de entrenamiento y asistiendo en la consecución de objetivos de acondicionamiento físico. *SmartCoach* ha sido diseñada para ser amigable e intuitiva para el usuario. A continuación, se detallarán las principales funcionalidades que ofrecerá:

- La aplicación diseñará rutinas personalizadas para los usuarios, tomando en consideración factores como sus metas específicas, los grupos musculares que desean enfocar, eventuales limitaciones físicas, el tiempo que tienen disponible y el equipo con el que cuentan. En función de los objetivos marcados, se determinarán los ejercicios más apropiados, estableciendo las repeticiones, series y periodos de descanso recomendados.
- Con cada sesión de entrenamiento, las rutinas se adaptarán y perfeccionarán. La aplicación rastreará meticulosamente el avance del usuario en cada ejercicio, asegurando una evolución continua. Se mantendrá un registro detallado de las modificaciones en los pesos o resistencias empleadas. Adicionalmente, se incorporará una encuesta concisa para refinar la rutina de acuerdo con el progreso del usuario. Por ejemplo, si el objetivo principal es la ganancia de masa muscular, la aplicación decidirá el momento óptimo para que el usuario eleve el peso, incremente las repeticiones o añada series.
- La aplicación proporcionará una guía completa sobre la ejecución de los ejercicios. Esto incluirá imágenes instructivas que demuestren cada movimiento, imágenes que destaquen la postura correcta, especificaciones sobre el equipo necesario, y una identificación clara de los músculos que se trabajan en cada ejercicio. Además, se incluirán descripciones detalladas que aborden la técnica adecuada y las recomendaciones para una respiración correcta.
- *SmartCoach* está diseñada para adaptarse al equipo disponible del usuario. Si el usuario dispone de un gimnasio en casa o de determinados equipos de ejercicio, podrá registrarlos en la plataforma. A partir de esta información, las rutinas se personalizarán para aprovechar al máximo dicho equipo. Además, si un usuario se asocia a un gimnasio, podrá acceder a rutinas específicas basadas en el equipo que el administrador del gimnasio haya registrado en la aplicación.
- Los usuarios tienen la opción de suscribirse a un gimnasio y, al hacerlo, acceder a un mapa detallado configurado por el administrador. Este mapa no solo destaca el equipo de ejercicio, sino que también señala áreas adicionales como baños, oficinas y una sala médica. Esta característica permite a los usuarios localizar fácilmente el equipo necesario para su rutina. Además, les brinda la oportunidad de explorar el gimnasio en su totalidad y descubrir una variedad de ejercicios que pueden realizar con el equipo disponible.

- El sistema está diseñado para ofrecer un seguimiento completo de las actividades realizadas en el gimnasio. Los usuarios podrán rastrear su avance en los ejercicios, definir metas y registrar su progreso hacia ellas, documentar información sobre su composición física y monitorear su evolución a lo largo del tiempo. Además, la plataforma brinda la posibilidad de actualizar y modificar cualquier limitación física, garantizando así una experiencia adaptada y flexible para cada usuario.

Las funcionalidades propuestas están orientadas a potenciar la experiencia del usuario en el gimnasio, minimizar el riesgo de accidentes y alcanzar resultados más significativos en salud y acondicionamiento físico. A partir de estos aspectos, es posible subrayar los siguientes beneficios:

- Personalización de rutinas: Dado que cada individuo posee características únicas en términos de composición física, nivel de actividad, objetivos y recursos, nuestra aplicación se caracteriza por su profunda personalización. De esta manera, garantizamos que cada usuario reciba un plan de entrenamiento meticulosamente adaptado a sus necesidades y condiciones particulares.
- Orientación visual y educativa: La esencia de nuestra aplicación radica en ofrecer a los usuarios rutinas efectivas acompañadas de instrucciones claras y precisas para la ejecución de los ejercicios. Esta orientación no solo busca potenciar la eficacia del entrenamiento, sino también minimizar cualquier riesgo de lesiones. Mediante la integración de recursos variados, como videos e imágenes ilustrativas, nos aseguramos de que los usuarios lleven a cabo los ejercicios de una manera correcta y segura.
- Seguimiento del progreso: Al llevar un registro detallado de la evolución en los ejercicios y de los cambios en la composición física, los usuarios pueden visualizar su avance a lo largo del tiempo. Esta perspectiva no solo sirve como fuente de motivación, sino que también incentiva a mantener una práctica consistente.
- Administración de objetivos: La capacidad de establecer y supervisar metas permite a los usuarios mantener un enfoque claro y dirigir sus esfuerzos hacia objetivos específicos. Esta orientación facilita la consecución de sus aspiraciones de manera más eficiente.
- Flexibilidad y adaptabilidad: La aplicación está diseñada para moldearse según las necesidades del usuario. Ya sea que surja una limitación física temporal, un cambio en el equipo disponible o una modificación en los objetivos, la plataforma se ajustará adecuadamente. Esto garantiza que la aplicación continúe siendo pertinente y valiosa, incluso ante cambios en las circunstancias o habilidades del usuario.

Es importante destacar algunas características que nuestra aplicación no incluirá en su versión actual:

- Integración con dispositivos wearables: Aunque algunas aplicaciones de gimnasio se sincronizan con relojes inteligentes o rastreadores de fitness para obtener datos en tiempo real de la actividad del usuario, nuestra aplicación no contará con esta funcionalidad en esta versión. Sin embargo, es una posibilidad que consideraremos para futuras actualizaciones.
- Espacio comunitario: A diferencia de otras aplicaciones que ofrecen un espacio para que los usuarios interactúen entre sí, la aplicación en su versión presentada no contará con esta característica.
- Entrenador personal en vivo: No ofreceremos un entrenador real que guíe a los usuarios a través de sus rutinas.
- Modelo freemium: A pesar de que muchas aplicaciones adoptan un modelo freemium, ofreciendo características básicas de forma gratuita y opciones avanzadas mediante pago, nuestra aplicación será completamente gratuita.

Definiciones, Acrónimos, Abreviaciones

SRS: Software Requirements Specification, documento cuyo objetivo es proporcionarle al lector una descripción completa sobre el producto de software desarrollado, abarcando el propósito, las características y los elementos fundamentales para su rendimiento y correcto funcionamiento. [1]

Android: Sistema operativo basado en el kernel de Linux, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes y proporciona una plataforma flexible y personalizable para el desarrollo y ejecución de aplicaciones. [2], [3]

Monolitos: Modelo tradicional de un programa de software, en la que todas las funciones y componentes están estrechamente relacionados y funcionan como una entidad única. [4] Las aplicaciones monolíticas generalmente se componen de tres elementos principales: una interfaz de usuario para el cliente, una base de datos y una aplicación en el servidor. [5]

Aplicación-App: Programa de software diseñado para realizar tareas específicas en dispositivos electrónicos diseñada para realizar tareas específicas o proporcionar ciertas funciones, ya sea para productividad, entretenimiento, comunicación, educación u otros propósitos.

Bases de datos: Colección organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Las bases de datos se utilizan para almacenar datos de manera que puedan ser fácilmente gestionados y accedidos por sistemas de software, aplicaciones y usuarios autorizados. [6]

Rutina de ejercicio: Plan organizado y estructurado de actividades físicas diseñadas para mejorar la condición física, la fuerza, la resistencia y la salud en general creadas con el propósito de establecer un programa regular de ejercicio que se ajuste a los objetivos personales de acondicionamiento físico de cada individuo. [7]

CRUD: Acrónimo de Create (Crear), Read (Leer), Update (Actualizar) y Delete (Borrar). Describe las cuatro operaciones básicas que pueden realizarse en la mayoría de las bases de datos y sistemas de gestión de información. [8]

Dispositivos Wearables: Dispositivo electrónico diseñado para ser portado en el cuerpo. Estos dispositivos tienen como objetivo de establecer una interacción continua con el usuario por ejemplo para recolectar información y procesar datos en tiempo real. [9]

Modelo Freemium: Modelo de negocio que ofrece un servicio y/o producto basándose en dos componentes clave “free” (gratis) y “premium” (algo de mejor calidad o más exclusivo). Bajo este enfoque la empresa ofrece un servicio básico de forma gratuita y también ofrece servicios con versiones mejoradas a las que se puede acceder mediante un pago. [10]

JWT: Es un estándar ampliamente utilizado para la transmisión segura de información en Internet. Este sistema emplea archivos con formato JSON. Su función principal consiste en verificar la identidad de un usuario que accede al sitio después de haber iniciado sesión previamente. [12]

PostgreSQL: Es un RDBMS de código abierto, desempeña un papel fundamental en el almacenamiento, administración y escalabilidad de datos sin importar su magnitud. Sus características notables incluyen confiabilidad, escalabilidad y la habilidad de manejar datos de gran amplitud y complejidad de manera eficaz. [13],[14]

Bases de datos relacionales: Este modelo de bases de datos organiza los datos en tablas compuestas por filas y columnas, utilizan un sistema de relaciones y restricciones para garantizar la integridad y la coherencia de los datos. Son principalmente usadas en aplicaciones empresariales y en sistemas de información, debido a su capacidad para gestionar datos estructurados de manera eficiente y mantener la integridad de los datos. [15]

SymmetricDS: Es un software de código abierto para la sincronización de datos entre múltiples bases de datos en tiempo real, tiene como objetivo garantizar la consistencia y disponibilidad de los datos en los entornos definidos. [16]

Spring Boot: Extensión de código abierto dentro del ecosistema de Spring Framework que facilita el desarrollo de aplicaciones Java, esta herramienta se centra en simplificar este proceso al proporcionar una configuración automática inteligente, servidores web integrados y una variedad de bibliotecas preconfiguradas. [17], [18]

SQLite: Herramienta de software libre, permite almacenar información en dispositivos de forma sencilla, eficaz y rápida. SQLite se integra directamente en la aplicación y opera en un solo archivo de base de datos. Esto lo hace especialmente adecuado para aplicaciones móviles en donde se necesita una base de datos local eficiente y fácil de administrar. [23]

SRS	Software Requirement Specification
CRUD	Create, Read, Update, Delete
JWT	JSON Web Token
RDBMS	Sistema de gestión de bases de datos relacionales

Tabla 1 Acrónimos y nombres

Apreciación Global

Este documento sirve como guía detallada para el proyecto *SmartCoach*, detallando cada aspecto relevante del desarrollo del sistema. Está estructurado de manera que facilite la comprensión y navegación del lector a través de sus distintas secciones:

- **Introducción:** Proporciona el propósito, alcance, definiciones clave, referencias y una visión general del contenido del documento.
- **Descripción Global:** Aborda la perspectiva del producto, incluyendo interfaces con el sistema, usuario, hardware, software, comunicación, restricciones de memoria, operaciones y adaptaciones. También se detallan las funciones del producto, características del usuario, restricciones, suposiciones y dependencias.

- Plan de Manejo de Requerimientos: Se centra en el levantamiento, verificación, validación, tipos, atributos y control de estado de los requerimientos. También se abordan mecanismos de priorización, trazabilidad.
- Requerimientos Específicos: Detalla los requerimientos funcionales y no funcionales del sistema.
- Proceso de Ingeniería de Requerimientos: Esta sección aborda el proceso específico de ingeniería de requerimientos.
- Análisis del Sistema – Modelo de Dominio: Presenta el modelo de dominio.

Con esta estructura, el lector podrá acceder de manera organizada a la información detallada de cada aspecto del proyecto, facilitando su comprensión y análisis.

Descripción Global

Esta sección proporciona una visión detallada de las diversas herramientas empleadas en la creación de la aplicación, las cuales contribuyen a la definición de requisitos, limitaciones del sistema, funciones, presentaciones y más.

Perspectiva del Producto

Este es un producto nuevo, compuesto por una plataforma móvil y un servidor. *SmartCoach* actuará como una guía en el gimnasio, generando rutinas personalizadas que toman en cuenta los objetivos, limitaciones y equipo disponible para los usuarios, entre otras funcionalidades. Además, la aplicación contará con un apartado adicional dedicado a la visualización e interacción con el mapa del gimnasio al que el usuario pertenece. En esta sección, se detallarán las diversas interfaces del sistema, incluyendo las de usuario, hardware, software y comunicación. También se abordarán las restricciones de memoria, operaciones y requisitos de adaptación.

Interfaces con el Sistema

El sistema adopta una arquitectura cliente-servidor de 2-tier capas. Esta decisión se toma con el objetivo de mantener el sistema escalable, flexible y seguro. La aplicación móvil actúa como la "capa de presentación", siendo la interfaz principal para la interacción del usuario. El servidor, por otro lado, funciona como la "capa de lógica de negocio" y también como la "capa de acceso a datos". Este enfoque monolítico nos permite manejar la autenticación, la lógica de negocio y la integración de la base de datos en un solo lugar. Las interfaces que se manejarán con el sistema son:

- **Notificaciones:** Para las notificaciones, se opta por utilizar notificaciones locales generadas a través de Android Studio. Aunque es común utilizar servicios de terceros como Firebase Cloud Messaging, nuestras funcionalidades actuales no requieren tal complejidad.
- **Autenticación:** Para la autenticación, el sistema utiliza JWT (JSON Web Tokens). Este enfoque nos permite mantener un sistema de autenticación robusto y seguro sin depender de servicios de terceros.
- **Base de Datos e Interfaces de Base de Datos:** La aplicación funcionará tanto en modo offline como online. En el modo offline, se utilizará una base de datos local proporcionada por Android, datos Symmetric DS. En el modo online, todos los datos se sincronizarán con nuestra base de datos PostgreSQL.
- **API RESTful:** La comunicación entre la aplicación móvil y el servidor se facilitará mediante una API RESTful, desarrollada utilizando Spring Boot. Esta API manejará diversas operaciones, incluida la autenticación, la lógica de negocio, la interacción con la base de datos y el sincronizador de datos.
- **Permisos de Dispositivo:** Nuestra aplicación requerirá permisos específicos para acceder a ciertas funcionalidades del dispositivo, como la cámara y la galería de fotos, principalmente para la gestión de fotos de perfil de los usuarios.

Interfaces con el Usuario

La interfaz de usuario se segmenta según el tipo de usuario que utilice la aplicación móvil diseñada para el sistema operativo Android.

- **Usuarios Clientes:** Usuarios interesados en adoptar hábitos saludables de ejercicio diario encuentran en la aplicación interfaces gráficas interactivas. A través de estas interfaces, pueden registrarse, iniciar sesión y brindar los datos necesarios para que la aplicación genere rutinas personalizadas. Estas rutinas se ajustan a los ejercicios deseados por el usuario, tanto para ser realizados en casa con los equipos disponibles como en el gimnasio al que estén suscritos, siempre y cuando este último cuente con información en la aplicación. Además, se ofrece la opción de hacer uso del mapa para ubicar los elementos requeridos según la distribución del gimnasio.

Por otra parte, los usuarios también tienen la capacidad de consultar y actualizar sus perfiles, revisar los registros de sus sesiones de entrenamiento, incorporar especificaciones sobre limitaciones físicas y establecer metas personales. La aplicación brinda acceso a una lista de ejercicios generados automáticamente, y les permite personalizar sus rutinas según

sus preferencias. De igual manera, la plataforma ofrece la posibilidad de visualizar videos explicativos, descripciones detalladas de los ejercicios y particularidades específicas de las rutinas.

- **Usuarios Administradores:** Esta categoría engloba a los usuarios responsables de la gestión de gimnasios que buscan optimizar el valor de sus establecimientos mediante las capacidades de la aplicación. Estos usuarios tienen la opción de registrarse e iniciar sesión en la plataforma. Además, pueden solicitar la verificación para acceder a todas las funcionalidades que ofrece la aplicación. Dependiendo de la respuesta a esta solicitud, podrán acceder a las funciones que les permiten agregar un gimnasio y proporcionar información esencial, como fotos y dirección.

Asimismo, tienen la capacidad de añadir los elementos disponibles en su gimnasio, y a través de una función de mapeo, pueden crear un plano detallado que incluye zonas específicas para actividades como cardio, entrenamiento de fuerza, áreas de descanso, baños, enfermería, puertas, escaleras, y otros espacios relevantes en el gimnasio.

Interfaces con el Hardware

La **aplicación móvil** está diseñada para funcionar en dispositivos con pantallas táctiles, lo que facilita la interacción del usuario con la interfaz. Además, la aplicación hace uso de la cámara del dispositivo para funciones relacionadas con la toma de fotos. En cuanto a la **comunicación** entre la aplicación y el servidor, se utilizará el protocolo HTTP/HTTPS. Específicamente, se asignará el puerto 8043, que es el puerto estándar para comunicaciones HTTPS seguras. Dado que la aplicación está destinada para dispositivos móviles, la **conexión** con el servidor se establecerá a través de una conexión inalámbrica, ya sea mediante Wi-Fi o datos móviles. No se requiere hardware adicional para esta conexión, más allá del enrutador o punto de acceso necesario para proporcionar la conectividad Wi-Fi.

Interfaces con el Software

Las siguientes características de software son fundamentales para el correcto funcionamiento del sistema.

- **Sistema operativo Android compatible con Android 10 o versiones posteriores:** Para que la aplicación *SmartCoach* funcione correctamente se solicita que los usuarios cuenten con una versión de Android 10 o posteriores. Android se basa en el núcleo Linux y utiliza una interfaz de usuario basada en pantalla táctil, es un sistema operativo de código abierto y es uno de los sistemas operativos móviles más populares y ampliamente utilizados en todo el mundo. [3] La elección de este sistema operativo se basa la amplia adopción de

dispositivos móviles que utilizan versiones a partir de Android 10, lo que permite alcanzar a la mayor audiencia de usuarios posibles. Además, la aplicación *SmartCoach* ha sido diseñada para aprovechar las últimas características y mejoras que serán implementadas en futuras versiones de Android lo que asegura un soporte más sólido y una mayor adaptabilidad en caso de cambios en políticas o comportamientos. [12]

- **JWT Json Web Tokens:** Es un estándar establecido en el documento RFC 7519 y se utiliza como un mecanismo seguro para representar información entre dos partes. Este estándar se utiliza para servicios como autenticación y autorización, y también se utiliza para transmitir información específica y confiable entre diferentes componentes de una aplicación. [23] La elección de este estándar para el proyecto fue debido a su capacidad de ser autocontenidos es decir que toda la información necesaria para verificar la validez del token se encuentra dentro de este mismo lo que proporciona su escalabilidad e interoperabilidad. En el proyecto *SmartCoach* se implementará en el servidor para cuando un usuario inicie sesión en una aplicación, el servidor generará un JWT que contiene la información del usuario y su estado de autenticación.
- **PostgreSQL:** Es una plataforma de bases de datos utilizada para almacenar, gestionar y consultar datos estructurados. En el contexto del proyecto *SmartCoach*, se ha elegido implementar PostgreSQL en su versión 15 debido a sus beneficios significativos, que incluyen flexibilidad, confiabilidad, estabilidad y seguridad. [24]
En el proyecto, PostgreSQL desempeñará un papel fundamental al servir como la base de datos en línea que almacenará y gestionará toda la información relacionada con los usuarios, los equipos y las diversas funcionalidades de los mapas. Esto incluye datos críticos para el funcionamiento de la aplicación, como perfiles de usuarios, datos de seguimiento de equipos y configuraciones de mapas interactivos.
- **SQLite:** Es un sistema de gestión de bases de datos autocontenido, es decir que está dentro de un archivo y no necesita de componentes externos o servidores adicionales para que funcione correctamente, es utilizado para almacenar datos en aplicaciones y dispositivos. [23] Se optó por usar esta base de datos para el almacenamiento local de información como las rutinas, el registro del progreso, los objetivos entre otros, sin embargo, para la versión que se entregará no se implementará y solo quedarán diseñadas sus funcionalidades para obtener todos los detalles puede consultar el documento de definición de requerimientos que se encuentra en el anexo. [Ver anexo 2. Definición de requerimientos].
- **SymmetricDS:** Esta plataforma permite la sincronización de datos entre múltiples bases de datos. Se ha considerado la utilización de la versión 3.14.4 [16], la cual se incluirá en las versiones posteriores junto con las funcionalidades planificadas para el futuro, y se basó

debido a que es multiplataforma y es compatible con SQLite y PostgreSQL importante debido a que son las bases de datos usadas en el proyecto *SmartCoach* las cuales se comunicarán entre sí para almacenar y/o actualizar información del sistema.

- **Spring Boot:** Es una herramienta que permite crear aplicaciones autónomas que se ejecutan por sí solas sin depender de un servidor externo. La versión implementada para el desarrollo del proyecto es la 2.5.4 y su función para la aplicación *SmartCoach* se basa en la creación del servidor debido a sus características para un rápido desarrollo, el soporte que tiene para diversas bases de datos es altamente escalable y puede manejar una gran cantidad de solicitudes de los usuarios móviles. [21]

Interfaces de Comunicación

La aplicación móvil establecerá una comunicación segura y eficiente con el servidor back-end mediante diversos protocolos y tecnologías. Los componentes clave de esta comunicación son:

- **Protocolos de red:** Se utilizará HTTPS para asegurar una comunicación cifrada y auténtica entre el cliente y el servidor. Cabe mencionar que este protocolo opera sobre TCP, lo que garantiza una entrega confiable y ordenada de los paquetes de datos. Además, HTTPS es una extensión segura del protocolo HTTP, que sigue el estándar definido en RFC 2818.
- **API RESTful:** La aplicación hará uso de este estilo arquitectónico que permite realizar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar), haciendo uso de los métodos de HTTP estándar como GET, POST, PUT y DELETE. Esta API es agnóstica respecto al tipo de cliente, lo que facilita la integración con diversos dispositivos y plataformas.
- **Formato de datos:** Al hacer uso de la API RESTful, tenemos varias opciones a elegir para el formato. Para este proyecto, se hará uso del formato JSON (JavaScript Object Notation). Se selecciona debido a que es ligero y fácil de interpretar, lo que contribuye a un rendimiento más rápido y menor utilización del ancho de banda, ajustándose así a las necesidades de nuestra aplicación y sus usuarios.
- **Autenticación y autorización:** Se utilizará TOKEN JWT de modo que solo los usuarios autenticados y autorizados puedan acceder a los recursos del servidor. Una vez que el usuario sea autenticado, se asignará un token JWT a las cabeceras de las solicitudes HTTP subsiguientes. De este modo, el servidor podrá autorizar o denegar el acceso a los recursos. Haciendo uso de estos componentes, se asegura que nuestro sistema tenga una comunicación robusta, segura y eficiente entre la aplicación móvil y el servidor, contribuyendo así a ofrecer una experiencia de usuario fluida y segura.

Restricciones de Memoria

En el desarrollo del proyecto *SmartCoach*, se han establecido requisitos mínimos de memoria para garantizar un rendimiento óptimo y el funcionamiento adecuado de la aplicación.

En lo que respecta a la memoria RAM, se sugiere disponer de un mínimo de 4 GB para proporcionar a los usuarios una experiencia más fluida al utilizar los diversos componentes de la aplicación. En cuanto al almacenamiento, se recomienda contar con al menos 1 GB para garantizar un rendimiento adecuado de las funciones gráficas y el correcto funcionamiento de la gestión de la información.

Suposiciones

- **Disponibilidad de Internet:** La aplicación tiene modos de funcionamiento online y offline, pero se requiere de una conexión a internet para crear las rutinas, actualizar la información del gimnasio, suscribirse a uno, entre otras cosas. Es necesario que los usuarios tengan acceso a una conexión a internet para utilizar todas las funcionalidades de la aplicación de forma eficiente y actualizada.
- **Dispositivos Móviles Compatibles:** Se asume que los usuarios del Sistema *SmartCoach* tendrán dispositivos móviles compatibles con la aplicación en términos de sistema operativo y hardware.
- **Terceros:** El sistema hace uso de terceros como PostgreSQL, Spring Boot, API RESTful; por lo que se supone que estos estarán disponibles y no tendrán cambios significativos que afecten la funcionalidad de la aplicación.
- **Privacidad:** Se hace la suposición de que los usuarios aceptarán el tratamiento de sus datos de salud, ya que estos son básicos para el funcionamiento de la aplicación.
- **Información del Gimnasio:** Se asume que los administradores de gimnasio registrarán y actualizarán la información de su gimnasio correctamente.
- **Conocimiento:** Se asume que los usuarios y administradores tendrán conocimiento básico para operar el dispositivo móvil, y los administradores tendrán conocimiento técnico sobre los equipos que hay en su gimnasio.

- **Zona Horaria:** Se asume que los usuarios son de Colombia, ya que la zona horaria a utilizar será de este país.
- **Entrenamiento:** Se asume que los usuarios buscarán rutinas generales como aumentar masa muscular, aumentar fuerza o aumentar la resistencia, más no rutinas específicas de recuperación, aumento de desempeño para deportistas o maratones, entre otras.

Dependencias

- **Velocidad de Red:** La aplicación dependerá de la velocidad de la red para conectar con el servidor, acceder a los servicios de este y a la base de datos, cargar la información y hacer la integración de datos entre la base de datos local y principal.
- **Arquitectura:** La aplicación depende de los servidores y las bases de datos para gestionar la información.
- **Sistema Operativo:** El funcionamiento del sistema dependerá del sistema operativo, su debida actualización y los dispositivos aceptados por este.

Restricciones

Las limitaciones que afectan directamente el proceso de implementación del proyecto se relacionan con varios aspectos de desarrollo. Estas limitaciones incluyen:

Tipo de Restricción	Restricciones
Requisitos de lenguaje	El sistema se desarrollará en el idioma español principalmente debido a que el sistema va a ser lanzado para Colombia en la ciudad de Bogotá.
Cliente	El producto debe contar con las funciones especificadas.
Interfaz de usuario	Se espera que la interfaz sea intuitiva y fácil de usar para todos los usuarios, tanto los usuarios comunes como los administradores. Además, se busca que la

	interfaz esté libre de cualquier tipo de errores.
Lenguaje de programación	El sistema <i>SmartCoach</i> será desarrollada en Android Studio a través del lenguaje de programación JAVA.
Legales	El software empleado para el desarrollo de este proyecto será en su mayoría de software libre. El contenido del proyecto será de propiedad de las estudiantes Johanna Bolívar y Maria Kamila Obregón.
Persistencia	Toda la información será persistida en el sistema de almacenamiento PostgreSQL.
Sistema operativo	El sistema operativo para el proyecto será Android 10.0 o posteriores.

Tabla 2 Descripción restricciones

Funciones del producto y Operaciones

A continuación, se mencionan las funciones principales de la aplicación *SmartCoach*. Para obtener todos los detalles puede consultar el documento de especificación de casos de uso que se encuentra en el anexo. [Ver Anexo 1. Definición CU]

Para la operación de la plataforma móvil, se debe tener en cuenta el tipo de perfil al que accede el usuario. Por lo tanto, se distinguen:

- En el modo administrador, puede gestionar mapas, equipos y perfiles mediante operaciones CRUD. Además, tiene la capacidad de ubicar equipos en el mapa, iniciar sesión y validar perfiles de usuario.
- En el modo usuario, se pueden gestionar perfiles, equipos y objetivos personales mediante operaciones CRUD. Los usuarios pueden modificar horarios de rutinas, registrar limitaciones físicas y datos de composición corporal. También tienen la opción de

suscribirse a un gimnasio, ver su ubicación y equipos disponibles, así como acceder a información detallada de ejercicios. Y adicional pueden iniciar rutinas, registrar progreso y eliminar ejercicios según lo necesiten. Se define que la aplicación móvil estará disponible 24/7 y para los períodos de mantenimiento, en caso de ser necesarios, se realizarán los domingos de 2 a 4 a.m. bajo la zona horaria de Colombia.

Por último, para los procesos de recuperación, se tiene planeado manejar dos bases de datos: la online, que contiene toda la información del sistema, y la local, que registra la información del usuario cuando este no tiene internet. Esta última mantiene información para que el usuario pueda hacer uso de la aplicación sin conexión a internet. Cuando el usuario o administrador entre a la aplicación y esta detecte conexión a internet y datos que no se han cargado a la base de datos principal, se le consultará si desea actualizarla. De ser así, la aplicación hará un respaldo y actualización de la información.

Modelo del Dominio

A continuación, se presenta el modelo del dominio de *SmartCoach*. Este modelo se creó como un paso fundamental en el análisis inicial del proyecto, permitiéndonos identificar y visualizar las principales entidades que debían estar presentes en el sistema, así como las relaciones entre ellas. Con esto, se busca obtener una comprensión más estructurada del sistema, facilitando las fases subsecuentes del desarrollo. Para una visualización más detallada y ampliada del modelo, puede referirse al anexo de diagramas. [Ver anexo 4. Carpeta de diagramas]

Es importante señalar que el diagrama se construyó basándose en los requerimientos del sistema. Se realizó un análisis detenido de cómo estos requerimientos podrían ser satisfechos de manera eficiente a través de las entidades y sus relaciones. Este modelo se enfocó no solo en resolver los problemas planteados, sino también en hacerlo de una manera eficiente y coherente, buscando reducir redundancias y pensando en el futuro del sistema. Para ver el diagrama con mejor resolución consulte el anexo 4. [Ver anexo 4. Carpeta de diagramas].

Cada cuadro en el modelo representa una entidad; en nuestro contexto, estas entidades son objetos o conceptos del mundo real, fundamentales para ayudar a definir y estructurar la información que manejará el sistema.

En nuestro modelo de dominio, identificamos cuatro tipos principales de relaciones: agregación, composición, herencia y asociación, las cuales se representan siguiendo el estándar UML. Las relaciones de asociación son, por defecto, bidireccionales; si no lo fueran, se añadiría una flecha para especificar la dirección.

Por último, es importante destacar que cada una de las entidades en el modelo de dominio ha sido detalladamente documentada en el diccionario de datos. En este documento, se ofrece una descripción de cada entidad, junto con las especificaciones de sus atributos, incluyendo descripciones y tipos de datos, entre otros detalles. Esto se realiza con el fin de que el lector tenga un entendimiento profundo y detallado de cada componente del sistema.

Características del Usuario

Como se mencionó previamente, la aplicación *SmartCoach* tendrá dos tipos de usuarios. A continuación, se describirán los aspectos importantes de cada uno.

Usuario

Características del usuario	Usuario Cliente
Descripción	Son hombres o mujeres en el rango de edad mayor a 18 años (no limitante) que desean iniciar o registrar su actividad física en el gimnasio. Deben contar con un dispositivo móvil inteligente con sistema operativo Android versión 10 o posterior.
Nivel de Seguridad o Privilegios	Estos usuarios tendrán registrados datos privados en nuestros sistemas, relacionados con su salud, por ende, estos datos deben ser resguardados. Para acceder a esta información, el usuario tendrá que autenticarse y solo podrá consultar los datos de salud de su propio perfil.
Actividades	Los usuarios pueden gestionar perfiles, equipos y objetivos personales mediante operaciones CRUD. También tienen la

	<p>opción de modificar horarios de rutinas, registrar limitaciones físicas y datos de composición corporal. Pueden suscribirse a un gimnasio, ver su ubicación y equipos disponibles de la instalación, así como acceder a información detallada de ejercicios. Además, pueden iniciar rutinas, registrar progreso y eliminar ejercicios según lo necesiten.</p>
Experiencia	<p>No se requiere ningún nivel de estudio específico; solo se debe contar con conocimiento básico del manejo de dispositivos móviles y acceso a la tienda de aplicaciones Play Store.</p>
Frecuencia de Uso	<p>La frecuencia de uso variará según la rutina diseñada para el usuario y su nivel de compromiso. Puede variar desde un uso diario durante períodos cortos hasta 3-6 días a la semana durante un período más largo de 2-4 horas por día, con una estimación de uso que abarca varios meses e incluso años. Por lo tanto, puede clasificarse como un uso frecuente.</p>
Características pertinentes	<p>Está enfocado al uso de usuarios mayores de 18 años, pero no está limitado a este tipo de usuarios. Deben contar con dispositivos móviles Android versión 10 o posterior y tener conexión a internet</p>

Tabla 3 Características del usuario cliente

Administradores

Características del usuario	Usuario de Gimnasio
------------------------------------	----------------------------

Descripción	Son hombres o mujeres mayores a 18 años (no limitante) que trabajen dentro de instalaciones deportivas o gimnasios. Deben contar con un dispositivo móvil inteligente con sistema operativo Android versión 10 o posterior.
Nivel de Seguridad o Privilegios	Este tipo de usuario compartirá con la aplicación información legal para autenticar su relación con el gimnasio al que quiere registrarse. Adicionalmente, guardará información sobre las instalaciones del gimnasio, por ende, su información debe ser resguardada. Para esto, se validará su identidad y solo podrá acceder a la información de su gimnasio.
Actividades	Este usuario puede gestionar mapas, equipos y perfiles mediante operaciones CRUD. Además, tiene la capacidad de ubicar equipos en el mapa, iniciar sesión y validar perfiles de usuario.
Experiencia	Este usuario no requiere ningún nivel de estudios o experiencia técnica en el área de tecnologías. Sin embargo, sí requerirá conocimiento técnico con respecto a la maquinaria que se quiere registrar. Por ejemplo, si el gimnasio tiene una prensa de piernas, deberá registrar la información necesaria como: nombre, marca, especificaciones, dimensiones, etc, además de contar con experiencia en el manejo de dispositivos móviles.
Frecuencia de Uso	Para este tipo de usuario, se espera que su frecuencia sea ocasional, ya que sus

	funciones están enfocadas en registrar elementos del gimnasio, por lo que no será necesario actualizar estos frecuentemente.
Características pertinentes	El usuario debe contar con un dispositivo móvil con sistema operativo Android versión 10 o posteriores. Así mismo y para versiones futuras se requerirá que el usuario deba ser verificado y aceptado como perfil de administrador para compartir la información de su gimnasio.

Tabla 4 Características usuario administrador

Características del Producto de Software

Con el objetivo de identificar las características del producto de software, se presentan en esta sección los requisitos funcionales, teniendo en cuenta las funcionalidades previamente establecidas. A continuación, se incluyen imágenes que organizan las características de las funcionalidades en columnas, proporcionando información como el número de requerimiento, el tipo de usuario asociado, el ID del caso de uso correspondiente, su prioridad, una descripción detallada y los métodos de verificación. Para tener más información sobre todas las funcionalidades y ver las imágenes con mejor resolución consulte el anexo de requerimientos. [Ver anexo 2. Especificación de requerimientos]

ID Req	Tipo Usuario	ID Caso de Uso	Nombre	Prioridad	Descripción	Verificación
--------	--------------	----------------	--------	-----------	-------------	--------------

Imagen 2 Detalles de las funcionalidades

- Funcionalidad: Ver mapa

RF - 40	Usuario Cliente	CU-11- Ver mapa	Ver mapa	Implementado	Alta	El usuario debe poder ver el mapa del gimnasio en pantalla al que está suscrito	El usuario puede seleccionar la opción de ver mapa en el gimnasio al que está suscrito y el sistema le mostrará en pantalla el mapa creado previamente por el administrador
RF - 41	Usuario Cliente	CU-11- Ver mapa	Ver información del equipo	Diseñado	Media	El usuario debe poder ver la información del equipo dando clic al icono que está ubicado dentro del mapa	El usuario puede seleccionar uno de los equipos que se muestran en pantalla ubicados dentro del mapa y el sistema le mostrará la información de este.
RF - 42	SmartCoach	CU-11- Ver mapa	Mensaje de error, el gimnasio no tiene mapa	Diseñado	Baja	El sistema emite un mensaje de error al usuario en caso de querer observar el mapa del gimnasio y este no tenga uno creado	El sistema muestra en pantalla un mensaje de error al no encontrar el mapa del gimnasio que desea ver debido a que no hay uno registrado dentro de la base de datos

Imagen 3 Detalle funcionalidad "Ver mapa"

- Funcionalidad: Crear rutina

RF - 150	SmartCoach	CU-25 - Crear rutina	Crear rutina de ejercicios para usuario	Implementado	Alta	El sistema debe poder crear una rutina de acuerdo a la información brindada por el usuario, como por ejemplo, sus objetivos, las limitaciones físicas, equipo personal o registrado en gimnasio que haya sido agregado al nivel de actividad	El sistema le mostrará en pantalla la rutina personalizada creada para el usuario dependiendo de la información brindada previamente
RF - 151	SmartCoach	CU-25 - Crear rutina	Mensaje de error por conexión no se puede crear la información de la rutina dentro la base de datos local del dispositivo del usuario	Diseñado	Baja	El sistema emite un mensaje de error al usuario en caso de que el usuario no cuente con conexión a internet y no se puedan actualizar los datos de la rutina creada en la base de datos local del dispositivo del usuario	El sistema muestra en pantalla un mensaje de error al no tener una conexión de internet para actualizar los datos de su nueva rutina y le recordará al usuario que sus datos se actualizarán cuando tenga una conexión a internet
RF - 152	SmartCoach	CU-25 - Crear rutina	Almacena los datos de la rutina en la base de datos local del dispositivo del usuario	Especificado	Media	El sistema almacena los datos de la rutina creada para el usuario previamente, en la base de datos local del dispositivo del usuario	El sistema realiza una consulta dentro de la base de datos local del dispositivo del usuario para verificar que la información de la rutina creada esté almacenada
RF - 153	SmartCoach	CU-25 - Crear rutina	Mensaje de error, no se puede encontrar un ejercicio para la rutina que cumpla con todas las necesidades del usuario	Implementado	Baja	El sistema emite un mensaje de error en caso de que al momento de crear la rutina no se pudiera encontrar un ejercicio que cumplan con los objetivos que se planteó el usuario, sus limitaciones físicas y su nivel de actividad	El sistema muestra en pantalla un mensaje de error al no encontrar ejercicios para crear que cumplan con las necesidades del usuario

Imagen 4 Detalle funcionalidad "Crear rutina"

● Funcionalidad: Ubicar equipo o elemento en mapa

RF - 68	Administrador	CU-17 - Ubicar equipo o elemento en mapa	Ubicar equipo	Implementado	Alta	El administrador debe tener la capacidad de ubicar un icono que represente los equipos en el mapa de su gimnasio	El administrador puede arrastrar un icono que represente los equipos al mapa creado previamente y ubicarlo como desee (no puede estar sobrepuesto a otro equipo o elemento agregado)
RF - 69	Administrador	CU-17 - Ubicar equipo o elemento en mapa	Ubicar otros elementos	Implementado	Alta	El administrador debe tener la capacidad de ubicar un icono que represente otros elementos en el mapa de su gimnasio	El administrador puede arrastrar un icono que represente los elementos al mapa creado previamente y ubicarlo como desee (no puede estar sobrepuesto a otro equipo o elemento agregado)
RF - 70	Administrador	CU-17 - Ubicar equipo o elemento en mapa	Definir equipo	Implementado	Alta	El administrador debe tener la capacidad de asignar un equipo al icono de equipo ubicado en su gimnasio	El administrador puede asociar equipos previamente creados en su perfil
RF - 71	Administrador	CU-17 - Ubicar equipo o elemento en mapa	Buscar equipos	Diseñado	Baja	El administrador debe poder buscar un equipo por su nombre o referencia, a fin de asignarlo a un icono de equipo en el mapa correspondiente	El administrador puede buscar un equipo por su nombre o referencia para asociarlo a los iconos de equipo ubicados en el mapa
RF - 72	Administrador	CU-17 - Ubicar equipo o elemento en mapa	Eliminar equipo	Implementado	Alta	El administrador debe tener la capacidad de eliminar iconos de equipo ubicados en su mapa	El administrador puede seleccionar el icono de basura para eliminar iconos de equipos previamente ubicados del mapa
RF - 73	Administrador	CU-17 - Ubicar equipo o elemento en mapa	Eliminar otros elementos	Implementado	Alta	El administrador debe tener la capacidad de eliminar el icono de otros elementos ubicados en su mapa	El administrador puede seleccionar el icono de basura para eliminar iconos de elementos previamente ubicados del mapa
RF - 74	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Mensaje de error, no se puede actualizar la información de los elementos ubicados en el mapa en la base de datos de la aplicación	Diseñado	Baja	El sistema emite un mensaje de error al administrador en caso de que el administrador no cuente con conexión a internet y no se puedan actualizar los datos de los elementos agregados en el mapa en la base de datos de la aplicación	El sistema muestra en pantalla un mensaje de error al no tener una conexión de internet para actualizar los datos de los elementos agregados en el mapa y le recordará al administrador que sus datos se actualizarán cuando tenga una conexión a internet
RF - 75	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Almacena los datos de los elementos ubicados en el mapa en la base de datos de la aplicación	Especificado	Media	El sistema almacena los datos de los elementos ubicados en el mapa creados por el administrador previamente en la base de datos de la aplicación	El sistema realiza una consulta dentro de la base de datos de la aplicación para verificar que la información de los elementos ubicados agregada esté almacenada
RF - 76	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Eliminar los datos de los elementos ubicados en el mapa en la base de datos de la aplicación	Especificado	Media	El sistema elimina los datos del elemento ubicado en el mapa seleccionado por el administrador de la base de datos de la aplicación	El sistema realiza una consulta dentro de la base de datos de la aplicación para verificar que la información de los elementos ubicados en el mapa ya no esté almacenada
RF - 77	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Actualizar los datos de los elementos ubicados en el mapa en la base de datos de la aplicación	Implementado	Media	El sistema actualiza la información de los datos de los elementos ubicados en el mapa que hayan sido modificados por el administrador dentro de la base de datos de la aplicación	El sistema verificará y actualizará los datos modificados de los elementos ubicados en el mapa en la base de datos de la aplicación
RF - 78	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Mensaje de error, no se puede actualizar la información de los equipos ubicados en el mapa en la base de datos de la aplicación	Diseñado	Baja	El sistema emite un mensaje de error al administrador en caso de que el administrador no cuente con conexión a internet y no se puedan actualizar los datos de los equipos agregados en el mapa en la base de datos de la aplicación	El sistema muestra en pantalla un mensaje de error al no tener una conexión de internet para actualizar los datos de los equipos agregados en el mapa y le recordará al administrador que sus datos se actualizarán cuando tenga una conexión a internet
RF - 79	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Almacena los datos de los equipos ubicados en el mapa en la base de datos de la aplicación	Especificado	Media	El sistema almacena los datos de los equipos ubicados en el mapa creados por el administrador previamente en la base de datos de la aplicación	El sistema realiza una consulta dentro de la base de datos de la aplicación para verificar que la información de los equipos ubicados agregada esté almacenada
RF - 80	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Eliminar los datos de los equipos ubicados en el mapa en la base de datos de la aplicación	Especificado	Media	El sistema elimina los datos del equipo ubicado en el mapa seleccionado por el administrador de la base de datos de la aplicación	El sistema realiza una consulta dentro de la base de datos de los equipos ubicados en el mapa ya no esté almacenada
RF - 81	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Actualizar los datos de los equipos ubicados en el mapa en la base de datos local de la aplicación	Especificado	Media	El sistema actualiza la información de los datos de los equipos ubicados en el mapa que hayan sido modificados por el administrador dentro de la base de datos de la aplicación	El sistema verificará y actualizará los datos modificados de los equipos ubicados en el mapa en la base de datos de la aplicación
RF - 82	SmartCoach	CU-17 - Ubicar equipo o elemento en mapa	Notificación de cambios en el mapa	Diseñado	Baja	El sistema debe enviar una notificación al usuario en caso de que el administrador del gimnasio de que está suscrito haya realizado algún cambio en el mapa, como la modificación de la ubicación o eliminación de algún equipo o elemento.	El usuario recibirá una notificación cuando se detecten cambios realizados por el administrador en el mapa del gimnasio al que el usuario se suscribió previamente

Imagen 5 Detalle funcionalidad "Ubicar equipo o elemento en mapa"

Plan de manejo de requerimientos

En este apartado del plan de manejo de requerimientos, se detallan los procesos llevados a cabo para la identificación, clasificación, registro, priorización y rastreo de los requerimientos del sistema, tanto funcionales como no funcionales.

Levantamiento de Requerimientos

En las siguientes secciones de esta propuesta, se detallará el proceso de levantamiento de requerimientos, enfocándonos en las necesidades esenciales identificadas y en cómo podrían abordarse de manera efectiva.

Requerimientos Funcionales

Para la identificación de los requerimientos funcionales, se adoptó un enfoque colaborativo y consultivo. Se llevaron a cabo múltiples sesiones de trabajo con el objetivo de discutir, analizar y

definir dichos requerimientos. En cada sesión, se evaluaron meticulosamente los requerimientos, ponderando su relevancia, factibilidad e impacto en el usuario final.

Adicionalmente, incorporamos nuestras propias perspectivas como usuarios finales del sistema. Dado que el proyecto está diseñado para personas en niveles de experiencia similares a los nuestros (principiantes e intermedios), esta autoevaluación nos permitió comprender de manera más profunda las necesidades de los usuarios finales y adaptar los requerimientos en consecuencia.

Se empleó también un enfoque participativo, involucrando a terceros en el proceso de manera informal. Se consultó con individuos que podrían ser usuarios finales o expertos en el dominio para enriquecer nuestro análisis con perspectivas externas. Esta retroalimentación adicional nos facilitó la tarea de reevaluar y afinar los requerimientos previamente definidos.

Este proceso iterativo y autoevaluado culminó en la formulación de un conjunto sólido y bien definido de requerimientos funcionales. En resumen, se utilizó un enfoque orientado a la problemática, identificando las dificultades o desafíos que enfrenta nuestro usuario objetivo, y a partir de este análisis se planteó cómo abordar y resolver dichas cuestiones.

Requerimientos No Funcionales

Para la identificación de los requisitos no funcionales, al igual que para los requisitos funcionales, se adoptó una aproximación colaborativa y consultiva. Durante las reuniones pertinentes, se llevaron a cabo observaciones detalladas, y se tuvieron en cuenta múltiples factores en relación con las funcionalidades requeridas para garantizar la confiabilidad y disponibilidad del sistema. Además, se consideraron otros criterios no funcionales cruciales para el desarrollo de la aplicación.

En este sentido, hemos prestado especial atención a lo que constituye una experiencia positiva en una aplicación, considerando nuestras propias preferencias, las preferencias de terceros y, en última instancia, lo que resulta más beneficioso para los usuarios finales. Este enfoque integral nos ha permitido establecer un conjunto completo de requisitos no funcionales que respaldan la efectividad y la satisfacción de los usuarios de la aplicación que estamos desarrollando como parte de este proyecto de investigación.

Verificación y Validación de requerimientos

Para la identificación inicial de los requerimientos funcionales y no funcionales, se adoptó un enfoque colaborativo y consultivo, tal como se describe en secciones anteriores de este documento. Este proceso implicó múltiples sesiones de trabajo e incorporó nuestras perspectivas como usuarias finales, así como la consulta con terceros para enriquecer nuestro análisis.

Posteriormente, se llevó a cabo un estudio para determinar las funcionalidades que debería tener la aplicación móvil, tomando en cuenta las necesidades del contexto y el alcance esperado del proyecto. Con esto y lo mencionado anteriormente, se elaboró la primera versión de los requerimientos funcionales y no funcionales de la aplicación. Estos fueron validados por el director de tesis, el Ing. Víctor Hugo Vela Pinzón. Tras esta primera verificación, se continuó con la debida documentación y definición de estos requerimientos, asignándoles un ID, un caso de uso, tipo de usuario, nombre, estado, prioridad, descripción y criterios de verificación. Después de esto, se realizó una segunda reunión para la validación y verificación de estos.

Este enfoque integral y riguroso asegura que los requerimientos finales no solo sean sólidos y bien definidos, sino también verificables y validados, contribuyendo así al éxito del desarrollo posterior del sistema.

Tipos de Requerimientos

A continuación, se presentan los diferentes tipos de requerimientos que están presentes en la aplicación. Para obtener información detallada y completa sobre estas funciones, consulte el documento de definición de requerimientos en la parte de anexos. [Ver anexo 2. Definición de requerimientos]

Requerimientos Funcionales	Iniciar Sesión
	CRUD Perfil
	Crear Rutina
	Ver Rutina
	Registrar progreso
	CRUD mapa

	CRUD equipo
Requerimientos No Funcionales	Disponibilidad
	Extensibilidad
	Fiabilidad
	Modificabilidad
	Portabilidad
	Rendimiento
	Seguridad
	Usabilidad

Tabla 5 Tipo de requerimientos

Atributos de los Requerimientos

Para la documentación de los requerimientos funcionales, se emplearon los siguientes atributos:

- **ID req:** Es el identificador único de los requerimientos funcionales, y se denotan como RF-#.
- **Tipo de usuario:** Se refiere al usuario que hará uso o validará este requerimiento; pueden ser usuarios finales o el mismo sistema.
- **ID Caso de uso:** Es el identificador del caso de uso asociado al requerimiento.
- **Nombre:** Consiste en un título corto que se utiliza para la identificación rápida del requerimiento.

- **Estado:** Indica la etapa en el ciclo de desarrollo en la que se encuentra el requerimiento. Los estados posibles son: especificado, diseñado, implementado y aprobado.
- **Prioridad:** Es un indicador de la importancia que el requerimiento tiene dentro del proyecto. Las categorías de prioridad son: alta, media y baja.
- **Descripción:** Proporciona una explicación detallada del requerimiento.
- **Verificación:** Describe el método mediante el cual se evaluará este requerimiento, estableciendo criterios para juzgar su correcto funcionamiento.

Para la documentación de los requerimientos no funcionales, se emplearon los siguientes atributos:

- **ID req:** Es el identificador único de los requerimientos no funcionales y se denotan como RNF-#
- **Tipo de requerimiento:** Se refiere a la categoría en la que se clasifica el requerimiento no funcional, es decir, el atributo de calidad que aborda (seguridad, rendimiento, portabilidad, modificabilidad, extensibilidad, usabilidad, disponibilidad y fiabilidad).
- **Nombre:** Consiste en un título corto que se utiliza para la identificación rápida del requerimiento no funcional.
- **Estado:** Indica la etapa en el ciclo de desarrollo en la que se encuentra el requerimiento. Los estados posibles son: especificado, diseñado, implementado y aprobado.
- **Prioridad:** Es un indicador de la importancia que el requerimiento tiene dentro del proyecto. Las categorías de prioridad son: alta, media y baja.
- **Descripción:** Proporciona una explicación detallada del requerimiento.

Mecanismo de Control de Estado de los Requerimientos

En el contexto del desarrollo del sistema *SmartCoach*, es esencial mantener un seguimiento estructurado y organizado de los requerimientos. Con este propósito, se implementó un Mecanismo de Control de Estado de los Requerimientos que establece cuatro estados principales. Estos estados sirven como indicadores fundamentales para señalar la fase en la que se encuentra cada requerimiento. Esto, a su vez, simplifica la tarea de supervisar y controlar el avance de cada aspecto del sistema de manera efectiva.

Estado	Descripción
Especificado	Requerimiento definido para los usuarios
Diseñado	Requerimiento ilustrado en mockups o diagramas
Implementado	Requerimiento implementado en código, sin haber realizado sus respectivas pruebas
Aprobado	Requerimiento implementado en código con sus pruebas respectivas aprobadas

Tabla 6 Estados de los Requerimientos funcionales

Mecanismos de Priorización de los Requerimientos

La priorización de los requerimientos funcionales y no funcionales del sistema se estableció durante las reuniones de levantamiento de requisitos y con los criterios de categorización de requerimientos definida por los estándares seguidos y las estudiantes Johanna Bolívar y Maria Kamila Obregón.

Los criterios considerados para determinar la prioridad de los requerimientos funcionales fueron los siguientes:

- **Valor para el usuario:** Se evaluaron criterios que agregan valor a la experiencia del usuario, priorizando aquellos que tienen un impacto significativo en la satisfacción y usabilidad de la aplicación.
- **Impacto en el negocio:** Durante el análisis, se examinaron cómo los requisitos afectan directamente a los objetivos del proyecto *SmartCoach* y su valor tanto en términos de generación de valor en la aplicación como la impresión que los usuarios tendrían de la misma.
- **Dependencias:** Identificamos requisitos que presentan dependencias con otros, lo que nos permite abordarlos de manera prioritaria, teniendo en cuenta la secuencia lógica de implementación.
- **Impacto en la Arquitectura:** Consideramos cuidadosamente cómo cada requisito influye en la arquitectura general del sistema, asegurándonos de mantener una estructura sólida y coherente.

- **Viabilidad Técnica:** Evaluar si la implementación de un requisito es técnicamente factible y si se pueden satisfacer los criterios de rendimiento y calidad.
- **Esfuerzo de implementación:** tomamos en cuenta la complejidad y el esfuerzo requerido para llevar a cabo cada requisito, lo que nos ayuda a asignar recursos de manera eficiente y efectiva en función de la prioridad de estos.

Los criterios considerados para determinar la prioridad de los requerimientos no funcionales fueron los siguientes:

- **Importancia Estratégica:** Evaluamos la relevancia estratégica de los requerimientos no funcionales, asegurándonos de que aporten un valor significativo en relación con los objetivos generales del proyecto.
- **Impacto en la Experiencia del Usuario:** Consideramos cómo cada requerimiento no funcional puede influir en la experiencia del usuario final, abarcando aspectos que incluyen la usabilidad, el rendimiento y la seguridad.
- **Impacto en el Rendimiento:** Se evaluó cómo un requerimiento no funcional afecta el rendimiento del sistema, como la velocidad de respuesta, la capacidad de carga y la escalabilidad.
- **Seguridad:** Damos prioridad a los requisitos de seguridad, que engloban la protección de datos y el acceso autorizado, debido a su importancia para preservar la integridad de la aplicación y la seguridad de los datos.
- **Satisfacción de Requisitos Funcionales:** Evaluamos si un requerimiento no funcional es necesario para garantizar el cumplimiento de los requisitos funcionales y si su omisión afectara la funcionalidad general de la aplicación.
- **Riesgo para la arquitectura:** Analizamos qué tipos de requerimientos tienen el potencial de influir directamente en la arquitectura general del sistema, considerando su impacto y necesidad de abordarlos de manera prioritaria.

Teniendo en cuenta estos criterios se clasificaron según su prioridad los requerimientos funcionales y no funcionales dentro de las siguientes categorías.

Prioridad	Descripción
Alta	Esencial para uno o varios interesados, este requisito debe incorporarse en las etapas iniciales del producto para evitar interferencias en el progreso global del prototipo.
Media	Aunque no es crucial para el avance global del proyecto, aporta características significativas y puede influir en la satisfacción de ciertos interesados.
Baja	Su desarrollo no es esencial, no tiene un impacto considerable en la evolución del prototipo y no repercute en los interesados.

Imagen 6 Escala de priorización de los requisitos funcionales

Requerimientos Específicos

Con el objetivo de obtener una visión detallada de los requerimientos que se han especificado, diseñado, implementado y aprobado, se ha elaborado un documento que incluye información sobre cada requerimiento, su impacto en los diferentes tipos de usuarios y el método de verificación correspondiente.

Requerimientos Funcionales

La especificación detallada de estos requerimientos se encuentra disponible en el documento de Definición de Requerimientos, que está adjunto en los anexos. [Ver anexo 2. Definición de requerimientos].

Requerimientos No Funcionales

La especificación detallada de estos requerimientos se encuentra disponible en el documento de Definición de Requerimientos, que está adjunto en los anexos. [Ver anexo 2. Definición de requerimientos].

Trazabilidad

Para garantizar el seguimiento continuo de la trazabilidad a lo largo de todas las fases del proyecto (análisis, diseño, implementación y pruebas), se ha decidido implementar un Excel donde se establecen distintas categorías que permiten la identificación y seguimiento detallado de cada componente involucrado. Para ver la información detallada de la tabla de trazabilidad por favor dirigirse al anexo [Ver anexo 3. Trazabilidad]

. A continuación, se explica en detalle la plantilla usada por tipo de requerimiento.

Requerimientos Funcionales

ID Requerimiento Funcional	ID Caso de Uso Asociado	ID Caso de Prueba Asociado

Tabla 7 Relación entre Requerimientos funcionales, Casos de Uso y Casos de Prueba

- ID de Requerimientos: Cada requerimiento funcional está asignado con un identificador único (ID de requerimientos) que facilita su rastreo en todas las fases del desarrollo del proyecto.

- ID de Caso de Uso: A cada caso de uso se le asigna un identificador único (ID de caso de uso) que está ligado a los requerimientos funcionales, permitiendo así una correlación directa y facilitando el seguimiento de cómo cada requerimiento se traduce en casos de uso específicos.
- ID de Caso de Prueba: Para asegurar que cada caso de uso se prueba adecuadamente, se asigna un identificador único (ID de caso de prueba) a cada caso de prueba, estableciendo una relación directa con el caso de uso correspondiente y, por ende, con el requerimiento funcional asociado.

Requerimientos No Funcionales

ID Requerimiento No Funcional	Elementos de Diseño Asociados	Escenarios de Calidad

Tabla 8 Relación entre Requerimientos no funcionales, Elementos de diseño asociados y Escenarios de Calidad

- ID de Requerimientos: Similar a los requerimientos funcionales, cada requerimiento no funcional se identifica mediante un ID único, facilitando su rastreo y gestión durante todo el ciclo de vida del proyecto.
- Elementos de Diseño Asociados: En esta columna se detallan los elementos de diseño que están relacionados con cada requerimiento no funcional. Estos pueden ser diagramas, especificaciones técnicas, o cualquier recurso de diseño.
- Escenarios de Calidad: Para abordar los aspectos no funcionales del sistema, hemos introducido los "escenarios de calidad", que describen las condiciones bajo las cuales el sistema debe operar y las expectativas de rendimiento, seguridad, fiabilidad, etc. Cada escenario está vinculado a uno o más ID de requerimientos no funcionales, proporcionando una hoja de ruta clara para las pruebas y validaciones necesarias.

Referencias

- [1] *Especificación de requisitos de software (SRS): consejos y plantillas* - Visure Solutions. (s.f.). Visure Solutions. <https://visuresolutions.com/es/software-requirement-specification-srs-tips-template/>
- [2] *¿Qué es el Sistema Operativo Android?* - Base de Conocimientos - ICTEA. (s.f.). Administración - ICTEA. <https://www.ictea.com/cs/index.php?rp=/knowledgebase/8974/iQue-es-el-Sistema-Operativo-Android.html>

- [3] *Qué es Android / Android.* (s.f.). Android. https://www.android.com/intl/es_us/what-is-android/
- [4] *Microservices vs. monolithic architecture / Atlassian.* (s.f.). Atlassian. <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>
- [5] *Monolítico frente a microservicios: diferencia entre arquitecturas de desarrollo de software - AWS.* (s.f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/compare/the-difference-between-monolithic-and-microservices-architecture/>
- [6] *¿Qué es una base de datos?* (s.f.). Oracle | Cloud Applications and Cloud Platform. <https://www.oracle.com/co/database/what-is-database/>
- [7] *¿Cómo armar una rutina de ejercicios? / Aprende Institute.* (s.f.). Aprende Institute. <https://aprende.com/blog/bienestar/entrenador-personal/como-armar-rutina-de-ejercicios/#:~:text=Una%20rutina%20de%20entrenamiento%20es,a%205%20series%20por%20ejercicio>
- [8] *CRUD - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web / MDN.* (s.f.). MDN Web Docs. <https://developer.mozilla.org/es/docs/Glossary/CRUD>
- [9] Arimetrics. (2023). Wearable. <https://www.arimetrics.com/glosario-digital/wearable>
- [10] *Freemium: conoce cómo funciona, sus ventajas y 3 ejemplos prácticos.* (s.f.). Rock Content - ES. <https://rockcontent.com/es/blog/freemium/#:~:text=El%20freemium%20es%20un%20modelo,para%20tener%20algunas%20funcionalidades%20extra.&text=11%20min%20read-.El%20freemium%20es%20un%20modelo%20de%20negocio%20basado%20en%20la,un%20producto%20o%20servicio%20gratuito>
- [11] M, Á. G. (2022, 23 de marzo). *Estas son las 10 novedades de Android 13 que más ganas tenemos de probar.* Xataka Android - Sistema operativo móviles Google, Play store, Apps. <https://www.xatakandroid.com/sistema-operativo/estas-10-novedades-android-13-que-ganas-tenemos-probar>
- [12] *¿Qué es JSON Web Token? / KeepCoding Bootcamps.* (s.f.). KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-json-web-token/>
- [13] *¿Qué es PostgreSQL? / KeepCoding Bootcamps.* (s.f.). KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-postgresql/>
- [14] *¿Qué es PostgreSQL?* (s.f.). Kinsta®. <https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/>
- [15] *¿Qué es una base de datos relacional?* (s.f.). Oracle | Cloud Applications and Cloud Platform. <https://www.oracle.com/co/database/what-is-a-relational-database/>

- [16] *SymmetricDS: Software de replicación de base de datos.* (s.f.). Un poco de Java. <https://unpocodejava.com/2013/01/14/symmetricds-software-de-replicacion-de-base-de-datos/>
- [17] *¿Qué es Java Spring Boot? | IBM.* (s.f.). IBM in Deutschland, Österreich und der Schweiz | IBM. <https://www.ibm.com/mx-es/topics/java-spring-boot>
- [18] Microsoft. (2023). *¿Qué es Java Spring Boot? Introducción a Spring Boot | Microsoft Azure.* (s.f.). <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-java-spring-boot>
- [19] Magaña, L. M. L. (2020, 17 de enero). *Qué es Json Web Token y cómo funciona.* OpenWebinars.net. <https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/>
- [20] *PostgreSQL: PostgreSQL 15 Press Kit.* (s.f.). PostgreSQL: The world's most advanced open source database. <https://www.postgresql.org/about/press/presskit15/es/>
- [21] *¿Qué es SQLite? | KeepCoding Bootcamps.* (s.f.). KeepCoding Bootcamps. <https://keepcoding.io/blog/que-es-sqlite/>
- [22] *SymmetricDS | Open Source Database Replication.* (s.f.). SymmetricDS | Open Source Database Replication. <https://www.symmetricds.org>
- [23] *SQLite: La Base de Datos Embebida.* (s.f.). SG Buzz. <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>
- [24] Vela, V. (2014) *Software requirements specification 201430 SRS V 1.0.*

Anexos

En este apartado de anexos, se presenta una recopilación de información adicional que complementa y respalda el contenido del presente documento de Especificación de Requisitos de Software (SRS). Los anexos incluyen datos específicos, tablas, diagramas y otros materiales que mejora la comprensión de los requisitos del software y proporcionan un contexto más completo para la implementación del proyecto de investigación. Cada anexo se encuentra numerado y se hace referencia a lo largo del SRS cuando sea pertinente para facilitar la consulta y la correlación de la información.

Anexo 1. Definición CU.

Anexo 2. Definición de requerimientos.

Anexo 3. Trazabilidad.

Anexo 4. Carpeta Diagramas