



Integrantes:

Barajas Diego
Cruz Brandonn
González Juan Sebastián
Miranda Nicolás
Orozco Juan David

Healthy Routine

Pontificia Universidad Javeriana
Ingeniería de Sistemas
Ingeniería de Software



Docente: Anabel Montero

Versión 2.0

1. Historial de cambios

Fecha	Versión	Detalles	Encargado
05/03/2019	1.0	Creación del documento y primera revisión de la plantilla del SRS	Todos los miembros
06/03/2019	1.2	Portada	Juan González
08/03/2019	1.3	Características del usuario	Juan González
08/03/2019	1.4	Suposiciones y dependencias	Diego Barajas
09/03/2019	1.5	Requisitos no funcionales	Juan Orozco
09/03/2019	1.6	Perspectiva del producto	Nicolás Miranda
09/03/2019	1.7	Suposiciones y dependencias	Diego Barajas
09/03/2019	1.8	Funciones del producto	Brandonn Cruz
13/3/2019	2.0	Introducción	Juan González, Diego Barajas y Brandonn Cruz
13/03/2019	2.1	Resumen	Juan Orozco
13/03/2019	2.2	Modelo de dominio	Diego Barajas
15/03/2019	2.3	Requisitos específicos	Juan González
15/03/2019	2.4	Proceso de verificación y validación	Juan Orozco
15/03/2019	2.5	Características del producto Software	Juan González
16/03/2019	2.6	Requisitos específicos	Juan González
16/03/2019	2.7	Proceso de ingeniería de requisitos	Juan Orozco
16/03/2019	2.8	Descripción de casos de uso	Brandonn Cruz
18/03/2019	2.9	Requisitos de desempeño	Nicolás Miranda
19/03/2019	3.0	Lista de tablas y lista de figuras	Diego Barajas

Tabla 1 Historial de cambios

Fecha	Versión	Detalles	Encargado
27/03/2019	3.1	Modificaciones de los casos de uso	Brandonn Cruz
27/03/2019	3.2	Correcciones SPMP	Nicolás miranda
28/03/2019	3.3	Modificaciones de los casos de uso	Brandonn Cruz
30/03/2019	3.4	Justificación según priorización	Juan Orozco
30/03/2019	3.5	Atributos del sistema Software	Nicolás Miranda
31/03/2019	3.6	Avance y funcionalidad	Juan Orozco
31/03/2019	3.7	Reporte Gerencial	Nicolás Miranda y Brandonn Cruz
31/03/2019	3.8	Correcciones SPMP	Brandonn Cruz
01/04/2019	3.9	Atributos del sistema software, requisitos de la base de datos	Nicolás Miranda
01/04/2019	4.0	Reporte gerencial	Nicolás Miranda y Brandonn Cruz
01/04/2019	4.1	Revisión completa del documento	Todos los miembros
03/04/2019	4.2	Correcciones de: suposiciones y dependencias, requisitos no funcionales	Juan González, Nicolás Miranda, Diego Barajas y Brandonn Cruz
04/04/2019	4.3	Correcciones de: proceso de verificación y validación	Juan Orozco
06/04/2019	4.4	Reporte gerencial	Brandonn Cruz

Tabla 2 Historial de cambios

2. Resumen

El presente documento tiene como objetivo dar a conocer la especificación de requisitos y todo el proceso necesario para la respectiva documentación de la aplicación *Healthy Routine*.

En el desarrollo del escrito se presenta inicialmente una introducción, donde se expone el alcance del proyecto y la descripción del producto. Continúa exponiendo el modelo del dominio de la aplicación, basándose en la abstracción global del sistema. Posteriormente se documentan los requisitos de la aplicación y los factores generales que afectan el producto, haciendo énfasis en su especificación, los atributos del sistema de software y los requisitos de la base de datos.

En otro orden de ideas se da a conocer el proceso de ingeniería de requisitos, los métodos, procesos y métricas utilizadas para el manejo de los requisitos del sistema. Por último, el documento consta con la descripción del proceso de verificación y validación de los requisitos del sistema. El proyecto está dirigido principalmente a todos aquellos interesados en conocer los requisitos de la plataforma y el diseño del software de *Healthy Routine*. De esta manera se espera que comprendan la especificación de aquellas ideas imprescindibles para el correcto desarrollo la aplicación móvil.

3. Tabla de contenidos

5.	Introducción.....	7
6.	Descripción global.....	8
6.1	Modelo de dominio.....	8
6.2	Perspectiva de producto	8
6.2.1	Interfaces con el sistema	9
6.2.2	Operaciones.....	9
6.2.2.1	Modos de operación del usuario.....	9
6.2.2.2	Periodos de actividad e inactividad.....	10
6.2.2.3	Procesos de recuperación	10
6.2.2.4	Funciones de soporte a procesamiento de datos	10
6.3	Funciones del producto.....	10
6.4	Características del usuario.....	11
6.5	Requisitos no funcionales	13
6.6	Suposiciones y dependencias.....	15
7.	Requisitos específicos.....	16
7.1	Características del producto software	16
7.2	Requisitos de desempeño.....	16
7.3	Atributos del sistema software	16
7.4	Requisitos de la base de datos	18
8.	Proceso de ingeniería de requisitos	21
8.1	Planeación	21
8.2	Levantamiento	22
8.3	Administración de Requisitos.....	24
9.	Proceso de verificación y validación	25
10.	Anexos.....	26
11.	Referencias	27

4. Lista de tablas

Tabla 1 Historial de cambios	2
Tabla 2 Historial de cambios	3
Tabla 3 Formato entidades del modelo de dominio.....	8
Tabla 4 Interfaces con el sistema.....	9
Tabla 5 Modos de operación de usuarios	10
Tabla 6 Periodos de actividad e inactividad	10
Tabla 7 Características del usuario beneficiario.....	12
Tabla 8 Características del usuario experto	13
Tabla 9 Requerimientos no funcionales	14
Tabla 10 Reglas de negocio Healthy Routine	14
Tabla 11 Confiabilidad.....	17
Tabla 12 Disponibilidad	17
Tabla 13 Seguridad	17
Tabla 14 Mantenibilidad	18
Tabla 15 Portabilidad	18
Tabla 16 Base de datos REQ/21	19
Tabla 17 Base de datos REQ/26.....	19
Tabla 18 Base de datos REQ/28.....	19
Tabla 19 Base de datos REQ/28.....	20
Tabla 20 Tipos de datos	20

5. Introducción

El siguiente documento tiene como objetivo principal la especificación de requisitos de software para el desarrollo de la aplicación *Healthy Routine*. A lo largo del escrito se encontrarán puntos relevantes para definir las funcionalidades entre el sistema y el usuario, que serán imprescindibles para la implementación del software.

En primera instancia se abordará la descripción global, donde se abstrae la información relevante para permitir la identificación de los requisitos (funcionales y no funcionales), los usuarios que accederán a la plataforma junto con sus características y el reconocimiento de las funciones con las cuales contará el producto. Por otra parte, se encontrará la definición de las suposiciones y dependencias del sistema para tener conocimiento de las limitaciones del aplicativo para su óptimo funcionamiento.

El documento hace énfasis en la especificación de los requisitos del software con el objetivo de definir las características del producto, los requisitos de desempeño, los atributos del sistema y todo el proceso que conlleva la planeación, el levantamiento y la administración de éstos.

El producto es una aplicación llamada *Healthy Routine*, que tiene como propósito brindar a los beneficiarios un servicio de manejo de planes de ejercicios de gimnasio y recomendaciones de dietas. Principalmente se hará recomendaciones de dietas, basándose en información de profesionales, sitios web y artículos de investigación en el área de las ciencias de la salud y nutrición. Las dietas que ofrece la aplicación se muestran como recomendaciones para el beneficiario, presentando el objetivo de cada una y los platos que la conforman.

Para cumplir el propósito de la aplicación, ésta ofrece distintos planes de ejercicios clasificados por nivel de dificultad, de los cuales el beneficiario podrá seleccionar uno para seguir. Cuando un plan es seguido por un beneficiario, la aplicación muestra el trabajo realizado y por hacer por medio de un calendario, además ofrece una guía de la rutina a realizar en el día, mostrando el ejercicio y la duración o número de repeticiones en que se debe realizar.

Por último, la aplicación ofrece todas las funcionalidades de soporte necesarias para que el equipo experto puede mantenerla en funcionamiento. Estas funcionalidades incluyen: poder agregar, editar o eliminar planes de ejercicios y dietas, además de mostrar los reportes de las opiniones de los usuarios sobre los dos ítems mencionados anteriormente.

6. Descripción global

6.1 Modelo de dominio

En esta sección se presentará el modelo de dominio que describirá el sistema. En éste se muestra una abstracción que permitirá la implementación de la aplicación. [Ver anexo \(Modelo de dominio\).](#)

Basados en este modelo se procedió a documentar cada una de las entidades presentadas en el mismo, donde se establecen los propósitos de cada una y los atributos que abarcan con el siguiente formato.

ID		Elemento del dominio	
Descripción			
Atributos			
Nombre		Descripción	

Tabla 3 Formato entidades del modelo de dominio

Esta documentación se encuentra anexada. [Ver anexo \(Entidades modelo de dominio\).](#)

6.2 Perspectiva de producto

Con el boom del *fitness*, más personas día a día empiezan a realizar actividades físicas tales como ir al gimnasio o salir a trotar; también tienen más conciencia de la importancia de la alimentación saludable para su vida; sin embargo, este impulso motivador en la mayoría de los casos dura muy poco tiempo y se vuelve una rutina de hábitos poco saludables. Por ende, *Healthy Routine* busca ayudar y mejorar la calidad de vida de jóvenes y adultos para que logren adquirir hábitos de vida saludables mediante una aplicación móvil.

Lo anterior se logrará mediante dos módulos principales que va a tener la aplicación; el primero, será el de ejercicios y rutinas físicas, el cual consistirá en una serie de ejercicios que la aplicación propone al usuario de acuerdo con los objetivos que él haya establecido al momento de registrarse en la aplicación. Y el segundo módulo será el de nutrición, en el cual, el usuario podrá encontrar diferentes descripciones y consejos acerca de la alimentación saludable y la suplementación.

6.2.1 Interfaces con el sistema

En primer lugar, es importante mencionar que *Healthy Routine* no tendrá ninguna interacción con sistemas externos y tampoco ninguna interacción con entidades o instituciones. En la tabla 3, se describen las tres principales interfaces del sistema las cuales son: interfaz de usuario, de software y de hardware.

Interfaz	Descripción
Usuario	El usuario podrá interactuar con el sistema, una vez descargue la aplicación móvil y desde este momento, encontrará diferentes ventanas o pantallas, en las cuales se presentan las funcionalidades de la aplicación, Para evidenciar lo anterior, Ver anexo (Mock-ups) .
Software	Se utilizará las <i>APIs</i> que ofrece <i>Firebase</i> para las llamadas a la base de datos y el sistema de autenticación de la plataforma. Del mismo modo, se usarán las <i>APIs</i> disponibles en <i>Ionic</i> tales como <i>Apache Cordova</i> , <i>Angular</i> y <i>NativeScript</i> para los temas de <i>frontend</i> .
Hardware	No se requiere ninguna interfaz de hardware particular además del dispositivo móvil donde se va a desplegar la aplicación, y tener conexión a internet.

Tabla 4 Interfaces con el sistema

6.2.2 Operaciones

6.2.2.1 Modos de operación del usuario

En la tabla 4, se describirá los modos de usuario que tendrá el sistema y las funciones a groso modo que realizará cada uno de ellos.

Tipo de usuario	Funciones
Beneficiario	<ul style="list-style-type: none">• Registrarse en la aplicación con la información necesaria, como el nombre, apellido, edad, objetivos.• Iniciar sesión con el correo y contraseña establecidos a la hora de registrarse• Seleccionar un plan de ejercicios de acuerdo con sus objetivos• Seleccionar una dieta de acuerdo con sus objetivos• Revisar el calendario con sus principales actividades realizadas.• Comenzar rutina de ejercicios diaria.• Cambiar su información personal.

Experto	<ul style="list-style-type: none"> • Gestión de planes de ejercicios (crear, eliminar y modificar). • Gestión de dietas (crear, eliminar y modificar). • Publicar planes de ejercicios y dietas. • Generar reporte de los planes y dietas.

Tabla 5 Modos de operación de usuarios

6.2.2.2 Periodos de actividad e inactividad

Estado	Proceso
Activo	<ul style="list-style-type: none"> • La aplicación siempre estará activa, siempre y cuando el dispositivo lo soporte.
Inactivo	<ul style="list-style-type: none"> • El sistema estará inactivo cuando el dispositivo se encuentre apagado o cuando el usuario no tenga la sesión activa en la aplicación. • El sistema estará inactivo cuando los expertos realicen mantenimiento, para realizar actualizaciones o parches a la aplicación.

Tabla 6 Periodos de actividad e inactividad

6.2.2.3 Procesos de recuperación

El protocolo que se va a seguir para recuperar la información, en caso de algún fallo, es el de *backup* diario a las 11:59pm. Para cumplir con lo anterior, se utilizará *Firebase Realtime Database* que permite hacer copias de seguridad diarias de los datos almacenados en la base de datos, los cuales se exportarán en formato *JSON* y se almacenarán en *Google Cloud Storage* [1] para ser accedidos por el experto cuando éste lo requiera.

6.2.2.4 Funciones de soporte a procesamiento de datos

La aplicación debe validar el tipo de usuario que está intentando ingresar al sistema, para determinar qué interfaz debe mostrar, si la interfaz de usuario o la interfaz de experto. El procesamiento de datos se realiza en la misma aplicación, sin embargo, los datos se encuentran alojados en la base de datos externa *Firebase*.

6.3 Funciones del producto

Las funciones del producto se dividen en tres grupos, el primero son las funciones que brindan servicios al beneficiario y el segundo las que brindan servicios al experto. El tercer grupo corresponde a las funciones generales de la aplicación, como registrar, autenticar y cerrar sesión.

- **Funciones para el beneficiario:** el beneficiario podrá ver información detallada de las dietas y planes de ejercicios publicados por los expertos. También podrá inscribir y eliminar tanto dietas como planes de ejercicios. Por último, podrá consultar un calendario que tendrá el historial de las actividades que hasta el momento ha realizado.
- **Funciones para el experto:** el experto podrá consultar las dietas y planes de ejercicios que han sido subidos a la aplicación, publicados o no. También podrá crear, editar, eliminar o publicar esas dietas y planes. Por último, se les ofrece a los expertos el poder generar reportes que muestren las dietas y planes más seguidos hasta el momento.

La descripción detallada de cada uno de estos planes puede ser consultada en la descripción de los casos de uso [Ver anexo \(Mock-ups\)](#). Para ver perspectiva completa de las funciones de la aplicación consulte el diagrama de casos de uso [Ver anexo \(Mock-ups\)](#).

6.4 Características del usuario

6.4.1 Beneficiario

Características del usuario	Todo usuario que acceda al sistema de <i>Healthy Routine</i> con el fin de informarse de dietas nutricionales y ejercicios físicos.
Nivel de seguridad o de privilegios	El beneficiario tendrá acceso a toda la información del sistema referente a rutinas de alimentación saludable, haciendo énfasis en los alimentos que contiene dicho plato y comunicando al usuario respecto a la información nutricional que contiene. Por otra parte, el beneficiario podrá conocer las rutinas de ejercicios físicos dentro de la plataforma, especialmente <i>Stretching</i> y <i>Crossfit</i> . También tendrá la posibilidad de agregar y modificar su información (información básica de contacto: nombre, apellidos, edad, peso, contraseña, género, objetivo y estado físico (de 1 a 10)).
Roles	El beneficiario tendrá la posibilidad de acceder a las funcionalidades de recomendaciones alimenticias y físicas.

Nivel de estudios o experiencia técnica	Conocimiento básico con tecnologías móviles/web y conocimiento básico respecto a términos nutricionales.
Frecuencia de uso	Cada vez que desee conocer de planes de alimentación o cada vez que dese informarse de ejercicios corporales. (Se recomienda periódicamente: diariamente, semanalmente).

Tabla 7 Características del usuario beneficiario

6.4.2 Experto en vida saludable

Características del usuario	Son limitados las personas que acceden al sistema de <i>Healthy Routine</i> como expertos (2). Éstos se encargarán de mantener en orden las dietas nutricionales y los ejercicios físicos. Es importante aclarar que los expertos deberán comunicarse para que cada uno sepa cuáles dietas y planes de ejercicio se deben crear, debido a que no se deberían repetir dentro del aplicativo. Los expertos son personas calificadas en el área de la salud, con conocimientos en nutrición y ejercicio físico.
Nivel de seguridad o de privilegios	El experto tendrá la posibilidad de crear, modificar, eliminar y controlar toda la información del sistema referente a rutinas de alimentación saludable, haciendo énfasis en los alimentos que contiene dicho plato y comunicando al usuario y de igual manera con los planes de ejercicio físico. Por otro lado, también tendrá el privilegio de acceder a la información para analizar los datos del tráfico que llega a los sitios web según la información proporcionada por los usuarios, centrándose en los tópicos comunes entre los usuarios para generar reportes de <i>analítica</i> .
Roles	El experto desempeñará funcionalidades como configuración y administración de recomendaciones alimenticias y físicas.

Nivel de estudios o experiencia técnica	Conocimiento básico con tecnologías móviles/web. Formación alta respecto a conocimiento nutricional y físico.
Frecuencia de uso	Semanalmente se encargará de administrar los planes de alimentación y los planes de ejercicios. Diariamente si desea conocer reportes generados tras el ingreso de información por parte de los beneficiarios.

Tabla 8 Características del usuario experto

6.5 Requisitos no funcionales

Los requisitos no funcionales según [2] son una propiedad o calidad que un producto debe tener, como la propiedad de apariencia, velocidad, seguridad o precisión. Pensando en esto, IngeSAP levantó los requisitos para *Healthy Routine*, con el ánimo de brindarle una mejor experiencia al usuario más allá de las funcionalidades presentadas, estos requisitos pueden ser encontrados en la tabla a continuación.

Requisito	Descripción	Clasificación FURPS+
RNF-01	La aplicación debe estar disponible en el 98.2% del tiempo.	Confiabilidad
RNF-02	La aplicación debe atender mínimo 500 usuarios concurrentes.	Rendimiento
RNF-03	La aplicación debe tener un diseño <i>responsive</i> .	Usabilidad
RNF-04	El sistema debe contar con un manual de usuario.	Usabilidad.
RNF-05	Se debe realizar un back-up de los datos del sistema de manera diaria.	Confiabilidad
RNF-06	El sistema debe tardar como máximo 2 segundos ante una petición por parte del usuario.	Rendimiento

RNF-07	Las contraseñas provistas por los usuarios deberán ser almacenadas bajo una función hash.	Confiabilidad.

Tabla 9 Requerimientos no funcionales

6.5.1 Restricciones de Software

Las restricciones generales que rigen a la aplicación *Healthy Routine* son las reglas de negocio definidas en la tabla a continuación., Por último, el software aceptará como mínimo 500 usuarios concurrentes.

Regla de negocio	Descripción
El usuario no debe contar con más de un plan de entrenamiento activo.	Los usuarios de <i>Healthy Routine</i> no puede contar con más de un plan de ejercicios activo en el sistema, esto porque no tiene sentido que un usuario esté registrado en más de un plan a la vez. Debido a que podría dificultar el proceso para llevar un seguimiento al usuario.
El usuario no debe contar con más de una dieta activa a la vez.	Los usuarios de <i>Healthy Routine</i> no puede contar con más de un plan de un plan de alimentación activo en el sistema, porque no tiene sentido que un usuario esté siguiendo más de una dieta a la vez.
La aplicación móvil estará disponible únicamente en el idioma español.	Para la primera versión el alcance está limitado para usuarios hispanohablantes.
Tolerancia a fallos.	En caso de que el servidor falle ver el proceso descrito en la sección 7.2.2.3 para recuperar la información.

Tabla 10 Reglas de negocio Healthy Routine

6.5.2 Restricciones de Software

Los lenguajes que se definieron para desarrollar la aplicación móvil *Healthy Routine* son, HTML, SCSS y TypeScript; esta se desarrollará con el framework Ionic y tanto el servidor como el gestor de bases de datos serán proveídos por Firebase. Ionic es de arquitectura MVC y del tipo Cliente[4]. Finalmente, la licencia de la de la aplicación es una licencia BSD 3-Clause "New" or "Revised" License y el para ingresar a la mayor parte de las funcionalidades, el usuario deberá contar con una conexión a internet, si no, tendrá acceso limitado a la aplicación.

6.5.3 Restricciones de Hardware

Para el servidor al Firebase estar en la nube, sólo es necesaria una conexión a internet para la comunicación con Firebase.

Para el cliente se necesita como mínimo un procesador Quad-Core a 1.3GHz, 1 GB de RAM. Tanto para la aplicación móvil en IOS y Android, el tamaño de la pantalla debe ser mínimo de 6 pulgadas que es el tamaño promedio de un dispositivo móvil de gama media y en la resolución basta con una de 1000 x 800.

6.6 Suposiciones y dependencias

Para la correcta funcionalidad del aplicativo se tomará la siguiente lista de suposiciones y dependencias como afirmaciones.

- La información proporcionada por los usuarios de la aplicación será verdadera.
- Los usuarios del aplicativo deberán contar con sus propios dispositivos para utilizar la aplicación, dispositivos los cuales requieren una versión de Android 4.0(Android Ice Cream Sandwich) o superior, para usuarios IOS sus dispositivos deberán contar con el sistema operativo IOS 2.5 o superior, para el correcto funcionamiento de la aplicación.
- La base de datos que utiliza la aplicación cumplirá de manera satisfactoria su propósito, sin fallos de servicio.
- El servicio de host utilizado para el proyecto no presentara fallos de servicio.
- Los Backus creados para la aplicación tomaran como base la última versión de la base de datos utilizada por la aplicación.
- Los clientes contaran con conexión a internet para utilizar la aplicación.
- Los cambios a una versión del proyecto serán notificados a todos los miembros del equipo de desarrollo del proyecto.
- La comunicación entre el cliente y el Product Owner es clara y asertiva.

7. Requisitos específicos

7.1 Características del producto software

En la siguiente sección se encontrarán aquellas características que debe poseer el sistema para que desempeñe todas las funcionalidades correctamente para el usuario (beneficiario y experto). Es importante tener en cuenta la trazabilidad y la capacidad de modificar requisitos conociendo el impacto en todo el sistema, debido a que es muy probable que a lo largo del desarrollo del proyecto de software se presenten imprevistos o cambios dentro de los mismos. Por lo tanto, la mejor manera de desempeñar la actividad de análisis del producto de software se hace documentando las características funcionales del aplicativo.

En [Ver anexo \(Requisitos\)](#) se enumeran y se explican todas las características que debe poseer el sistema a implementar.

Por otro lado, también se encuentran todos los requisitos (cada uno tiene su identificador) con los casos de uso asociados con el fin de cumplir con la característica de trazabilidad y de esta forma en caso de modificación o de implementación facilitar la ubicación del requisito.

7.2 Requisitos de desempeño

Una vez se han establecido los requisitos funcionales en la sección anterior, es necesario incluir los requisitos de desempeño. En esta sección se hace la división en requisitos estáticos (cantidad de usuarios concurrentes o simultáneos que el sistema puede y debe soportar; las terminales o equipos utilizados y conectados al mismo tiempo, etc.) y en requisitos dinámicos (número de transacciones que debe soportar el sistema dependiendo de la funcionalidad, los tiempos de respuesta del sistema ante las peticiones hechas por los usuarios, etc.) que está presente en el siguiente [Ver anexo \(Requisitos de desempeño\)](#).

Es importante darle criterios de medición a estos requisitos de desempeño, de forma que en las fases de verificación y validación se pueda comprobar el desempeño del sistema basándose en los siguientes atributos de calidad: latencia, tasa de procesamiento, capacidad y disponibilidad.

7.3 Atributos del sistema software

Es de vital importancia que el sistema tenga un correcto funcionamiento y así poder garantizar la calidad de este, por consiguiente, es necesario tener en cuenta una serie de características que definen cómo se realizan los procesos y la funcionalidad del sistema. Desde la tabla 10 hasta la tabla 14, se describirá cómo se van a implementar cada una de las características para la aplicación *Healthy Routine*. Los requisitos se encuentran definidos en [Ver anexo \(Requisitos\)](#)

Característica	Confiabilidad	Requisitos Asociados	REQ/21
Descripción	<p>Para el manejo de la información almacenada dentro del sistema se va a almacenar la información en el servidor de <i>Firebase</i>, sin embargo, la información del usuario también se va a almacenar en la memoria de manera local, esto con el fin de que haya persistencia y consistencia. Si ocurre alguna eventualidad, no se perdería la información y ésta se subiría al servidor de <i>Firebase</i> tan pronto se pueda.</p> <p>Así mismo, para que la aplicación se mantenga en funcionamiento a lo largo del tiempo, se van a realizar respaldos de la base de datos semanalmente, con el fin de ayudar a la tolerancia a fallos que se pueden presentar en cualquier momento y poder recuperar la información que se ha eliminado.</p>		

Tabla 11 Confiabilidad

Característica	Disponibilidad	Requisitos Asociados	REQ/26, REQ/26
Descripción	<p>Debido a que la aplicación está enfocada en el sector fitness, donde día a día las personas registran tanto sus dietas y ejercicios, se espera que el sistema esté activo los 7 días a la semana sin que el servidor tenga necesidad de reiniciarse.</p> <p>Como es muy difícil lograr una disponibilidad del 100%, se espera que por lo menos, la aplicación funcione 24 horas los días entre semana y los fines de semana, si puede existir una menor tolerancia a fallos. En términos generales se espera que la aplicación, mínimo esté disponible 165 horas a la semana, lo que equivale al 98.22%</p>		

Tabla 12 Disponibilidad

Característica	Seguridad	Requisitos Asociados	REQ/01, REQ/02, REQ/03
Descripción	<p>El sistema va a tener información personal de los usuarios, por lo que es importante que la información sensible como la contraseña se encuentre cifrada en la base de datos. Además, como se cuenta con 2 roles principalmente, es importante validar las funciones, mediante un usuario y contraseña, con el fin de poder llevar un registro de las actividades que estos hacen.</p>		

Tabla 13 Seguridad

Característica	<i>Mantenibilidad</i>	Requisitos Asociados	REQ/27
Descripción	El sistema va a contar con una arquitectura MVC, esto facilita el tema de modificaciones en componentes específicos sin necesidad de modificar todo el sistema. Otro aspecto que se debe tener presente es la documentación tanto del código como de los documentos técnicos para que los desarrolladores trabajen con estándares, para que cuando alguien del equipo desea hacer un cambio, se pueda entender con facilidad lo que hizo.		

Tabla 14 Mantenibilidad

Característica	<i>Portabilidad</i>	Requisitos Asociados	REQ/22, REQ/24
Descripción	La plataforma en la cual se va a desplegar la aplicación <i>Healthy Routine</i> es la plataforma móvil, tanto para dispositivos Android como para dispositivo iOS, donde se va a exportar un archivo con extensión <i>.apk</i> para que la aplicación se instale directamente en el celular con sistema operativo Android, lo cual permite una mayor portabilidad y versatilidad. Debido a que con <i>Ionic</i> se pueden crear aplicaciones móviles híbridas y también aplicaciones web, la aplicación también correrá en cualquier navegador web, principalmente <i>Firefox</i> y <i>Chrome</i> .		

Tabla 15 Portabilidad

7.4 Requisitos de la base de datos

Los requisitos de la base de datos son importantes porque definen cómo la información que hay dentro de la base de datos va a ser tratada y del mismo modo, cómo va a ser el funcionamiento de la base de datos en diferentes situaciones. Los requisitos se encuentran definidos en el siguiente Anexo [Ver anexo \(Requisitos\)](#)

# Requisito	REQ/21	Tipo de requisito	Soporte	Casos de uso asociados	-
Descripción	El sistema deberá almacenar y persistir la información en la base de datos en Firebase				
Razón	Es necesaria la vinculación de la aplicación con Firebase para mantener la información consistente				
Autor	Juan Sebastián González				
Criterio de medición	Conexión válida con la consola de Firebase				

Prioridad	Alta	Módulo asociado	Sistema
Versión	V1.1	Fecha	12/03/2019

Tabla 16 Base de datos REQ/21

# Requisito	REQ/26	Tipo de requisito	Soporte	Casos de uso asociados	-
Descripción	El sistema deberá estar en la capacidad de atender a por lo menos 50 usuarios.				
Razón	El sistema deberá soportar en simultáneo a 500 personas como mínimo				
Autor	Juan Sebastián González				
Criterio de medición	Escalabilidad del sistema si mantiene persistente los datos para cada usuario				
Prioridad	Alta	Módulo asociado	Sistema		
Versión	V2.0	Fecha	14/03/2019		

Tabla 17 Base de datos REQ/26

# Requisito	REQ/28	Tipo de requisito	Operación	Casos de uso asociados	-
Descripción	La base de datos debe tener como mínimo una capacidad de 1GB para almacenar la información de texto necesaria.				
Razón	La información de texto (No incluye imágenes, videos, contenido audiovisual) debe almacenarse en una base de datos no relacional a parte del contenido multimedia				
Autor	Juan Sebastián González				
Criterio de medición	Almacenamiento en Megabytes que posee el sistema a medida que se agrega información.				
Prioridad	Alta	Módulo asociado	Sistema		
Versión	V3.0	Fecha	16/03/2019		

Tabla 18 Base de datos REQ/28

# Requisito	REQ/29	Tipo de requisito	No funcional, Rendimiento	Casos de uso asociados	-
Descripción	El sistema deberá actualizar la base de datos a medida que se registra la información. La actualización deberá llevarse a cabo en menos de 2 segundos.				
Razón	Cualquier usuario que esté en el sistema deberá tener actualizada la información del sistema				
Autor	Juan Sebastián González				

Criterio de medición	Actualización válida en la consola de Firebase		
Prioridad	Alta	Módulo asociado	Sistema
Versión	V2.0	Fecha	14/03/2019

Tabla 19 Base de datos REQ/28

A continuación, se va a presentar los tipos de datos que tiene cada atributo en la base de datos *Firebase*:

Nombre atributo	Tipo de dato
apellidos	String
edad	Int
email	String
genero	String
nombres	String
password	String hash
rol	Int
descripcion	String
imagen	String Base64
objetivo	String

Tabla 20 Tipos de datos

Es importante mencionar que todos los datos que se alojan en la base de datos de *Firebase* en tiempo real, son almacenados como objetos JSON [3].

8. Proceso de ingeniería de requisitos

8.1 Planeación

En esta sección se muestra el plan que se desarrolló en la administración de requisitos para el desarrollo de la aplicación *Healthy Routine*. Se define el proceso que se realiza para manejar y controlar los cambios en los requisitos del sistema con el fin de mantener la calidad del software, tomando como referencia lo establecido y documentado inicialmente en el documento SPMP en la sección 9.1 Administración de requisitos.

8.1.1 Encargados

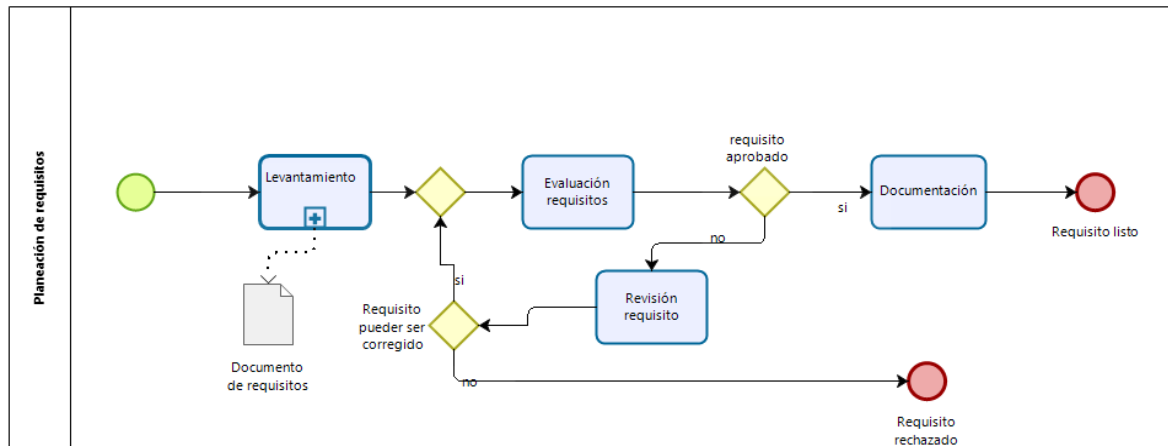
Los encargados de realizar el proceso de planeación de requisitos son: el director de desarrollo, el Scrum Master y el Product Owner. Aun así, los miembros restantes en el Scrum Team son libres de dar su opinión respecto a cómo se maneja la planificación de los requisitos del sistema para lograr requisitos de mayor calidad.

8.1.2 Frecuencia

El equipo de IngeSAP determinó realizar la planeación de los requisitos para cada entrega en esencia porque se determinó que el tiempo era prudente para hacer seguimiento a la planeación y documentación de los requisitos de la aplicación *Healthy Routine*. Es necesario corroborar todos los requisitos y determinar si hay alguno que necesite modificaciones.

9.1.3 Desarrollo del Proceso

Para el desarrollo del proceso, se definió inicialmente, empezar con la evaluación de los requisitos, el encargado de esta tarea tendrá que verificar que los requisitos cumplan con los criterios establecidos del sistema, si el encargado de la tarea encuentra algún comentario respecto a alguno de los requisitos este para a un estado de revisión, donde tendrá que ser revisado para que, de acuerdo al comentario planteado se puedan realizar cambios al requisito si es necesario o rechazar el requisito del sistema, una vez el requisito esté aprobado podrá pasar a la fase de documentación del mismo. El desarrollo se describe en el diagrama a continuación.



Powered by
bizagi
Modeler

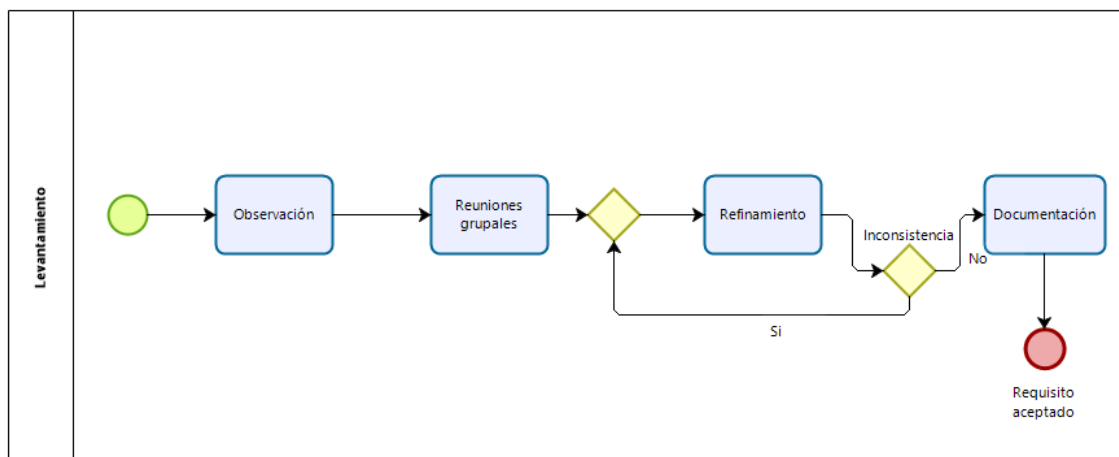
Ilustración 1 Proceso de ingeniería de requisitos

Para llevar un manejo claro de los requisitos se decidió realizar el manejo de versiones sobre el documento que incluye los requisitos del sistema, donde:

- El primer número, asciende cuando se realiza la modificación dentro del proceso de planificación. Principalmente porque el requisito tendrá un cambio muy grande.
- El segundo número, que está después del punto, asciende cuando un requisito es modificado en la revisión de este.

8.2 Levantamiento

El levantamiento es el punto de inicio para el proceso de desarrollo de un software. En éste se detallan todos los pasos a seguir para el desarrollo de las funcionalidades con las cuales contará el sistema, como las características adicionales con las que contará el software. A continuación, se presenta el proceso de levantamiento de requisitos para la aplicación móvil *Healthy Routine*.



Powered by
bizagi
Modeler

Ilustración 2 Proceso de levantamiento de requisitos

8.2.1 Observación

En esta fase, el equipo de IngeSAP estudió los problemas que presentan los usuarios del gimnasio a la hora de lograr sus metas de entrenamiento, y cómo muchas de las aplicaciones disponibles no presentan soluciones definitivas a los usuarios respecto a dietas y planes de entrenamiento.

A partir de la investigación por parte de los miembros del equipo de IngeSAP sobre este tema, se plantearon los primeros requisitos con los que debía cumplir una aplicación móvil, con el fin de brindarle la mejor solución a los usuarios de este mercado.

8.2.2 Reuniones grupales

El siguiente paso para el levantamiento de requisitos consistió en las reuniones semanales del equipo de IngeSAP, en éstas se realizaron lluvias de ideas para determinar aquellos requisitos con los que debía cumplir la aplicación a desarrollar. Este proceso fue realizado para la primera entrega del proyecto, donde aún se estaba llegando a un acercamiento de las especificaciones del sistema.

8.2.3 Refinamiento

Este proceso se desarrolló a lo largo de la fase de desarrollo de la aplicación *Healthy Routine*. Consistió en el estudio de cada uno de los requisitos del sistema entre todos los miembros del equipo de trabajo y determinar que requisitos se podían pulir un poco más, esto con el fin de mejorar la calidad de todos los requisitos y así mejorar la calidad de software del producto a entregar.

8.2.4 Documentación

En esta última fase del proceso de levantamiento de requisitos, se realiza una documentación detallada de cada uno de los requisitos levantados para el sistema, esta documentación está centrada en la especificación del requisito y su prioridad en el sistema.

Esta documentación es revisada por el Scrum Master quien da el visto bueno a los requisitos especificados para ser verificados y validados a lo largo del desarrollo del proyecto. Si el Scrum Master detecta alguna inconsistencia en alguno de los requisitos planteados, este será regresado para su reevaluación y corrección.

8.3 Administración de Requisitos

Administrar los requisitos del sistema, es decir, llevar la trazabilidad y manejar los cambios y el versionamiento de éstos, facilita la verificación de los requisitos definidos para que estén de acuerdo con las especificaciones del sistema y de lo acordado con el cliente.

A partir del documento en [Ver anexo \(Requisitos\)](#) se procedió a realizar una matriz de trazabilidad en la cual se incluyeron los siguientes campos:

- **ID:** este ID es el identificador único cada requisito. Cada ID tendrá la siguiente estructura:
 - a. REQ/ (número de requisito), donde primero se pone “REQ” de requisito, a continuación, se pone el ID correspondiente.
- **Especificación Requisito:** Se enuncia el requisito.
- **Estado:** Se selecciona el estado actual para cada uno de los requisitos, este define el avance que tiene cierto requisito en el marco del desarrollo del proyecto, para medir el estado de un requisito el equipo de IngeSAP determino los siguientes estados:
 - a. **Validado:** Si el requisito ha sido aprobado exitosamente.
 - b. **Verificado:** Si el requisito tiene diagramas de diseño asociados a él.
 - c. **En implementación:** Si el caso de uso asociado al requisito se encuentra en proceso de implementación.
 - d. **Implementado:** Si el caso de uso asociado al requisito ha sido implementado, cuenta con pruebas unitarias, pero aún no se han realizado pruebas de aseguramiento de calidad.
 - e. **Probado:** Si caso de uso asociado el requisito ha superado la fase de pruebas en QA, pero aún no ha sido presentado al cliente.
 - f. **En producción:** Si el caso de uso asociado al requisito esta implementado, probado y en producción para uso del cliente.
- **Versión:** Indica cual es la versión actual del requisito.
- **Caso de uso:** Se indica a que caso de uso se encuentra asociado un requisito en particular.
- **Autor:** Autor asociado al requisito.
- **Comentarios:** Texto con comentarios respecto al requisito.

9. Proceso de verificación y validación

En esta sección se podrán encontrar los procesos a llevar a cabo para verificar y validar los requisitos del sistema de *Healthy Routine*. A continuación, se especifica cada uno de estos procesos a seguir para la verificación y validación de requisitos.

9.1 Proceso de validación

El proceso de validación de un requisito se basa en la revisión de la estructura y la documentación asociada al requisito. Este proceso consiste en dos actividades principales: la verificación de la estructura del requisito y la verificación de la estructura de la documentación.

9.1.1 Validación de la estructura del requisito

En esta actividad se hace la verificación sintáctica del requisito, es decir, cómo está escrito. El requisito debe cumplir con los criterios establecidos en [Ver anexo \(Verificación requisitos\)](#). Para el requisito sea aprobado debe tener la aprobación para cada uno de los criterios definidos, una vez el requisito cumpla con estos criterios, el requisito será incluido en los requisitos del sistema.

9.1.2 Validación de la estructura de la documentación

En esta actividad se realiza la validación del contexto de la documentación del requisito, es decir, si la documentación propuesta para el requisito es acorde a las especificaciones del sistema, a las reglas de negocio y se encuentra escrita de manera correcta y adecuada. Principalmente se verifican las columnas de especificación y supuestos [Ver anexo \(Requisitos\)](#). Esta actividad se desarrolla en paralelo a la actividad anterior, a la validación de la estructura del requisito.

9.2 Proceso de verificación

Con el fin de verificar que los requisitos establecidos para la aplicación *Healthy Routine* cumplan con lo acordado en la documentación del requisito, se realizará el diseño asociado a cada uno de estos requisitos, junto con la implementación de estos en la plataforma, con esto se comprobará que el requisito realiza justamente la funcionalidad por la que fue incluido.

Una parte importante de este proceso es la matriz de trazabilidad explicada en la sección 9.3 Administración de Requisitos. En este se consulta el estado actual de un requisito y los comentarios asociados al mismo.

10. Anexos

- a) Casos de uso
 - 1. [Diagrama](#)
 - 2. [Especificaciones](#)
 - 3. [Excel](#)
- b) Modelos de dominio
 - 1. [Diagrama](#)
 - 2. [Entidades](#)
- c) Reporte gerencial
 - 1. [Word](#)
 - 2. [Excel](#)
 - 3. [Diagrama Gantt](#)
- d) Requisitos
 - 1. [Verificación](#)
 - 2. [Excel](#)
 - 3. [Requisitos de desempeño](#)

11. Referencias

- [1] Firebase, "Copias de seguridad automáticas," 2018.
- [2] J. Robertson, S., & Robertson, *Mastering the requirements process: Getting requirements right*. 2012.