| DUOC UC - ESCUELA DE INFORMATICA Y TELECOMUNICACIONES |
| --- |
| Especificación de Requisitos de Software |
| *Proyecto:* |
|  |
| **Revisión*: [99.99]*** |
| **[Seleccionar fecha]** |

| Especificación de Requisitos según estándar de IEEE 830. |
| --- |

**Contenido**

[**Ficha del documento 3**](#_heading=h.49x2ik5)

[**1. Introducción 4**](#_heading=h.30j0zll)

[1.1.](#_heading=h.1fob9te) Propósito 4

[1.2.](#_heading=h.3znysh7) Ámbito del Sistema 4

[1.3.](#_heading=h.2et92p0) Definiciones, Acrónimos y Abreviaturas 4

[1.4.](#_heading=h.tyjcwt) Referencias 4

[1.5.](#_heading=h.3dy6vkm) Visión General del Documento 4

[**2.**](#_heading=h.1t3h5sf) **Descripción General 5**

[2.1.](#_heading=h.4d34og8) Perspectiva del Producto 5

[2.2.](#_heading=h.2s8eyo1) Funciones del Producto 5

[2.3.](#_heading=h.17dp8vu) Características de los Usuarios 5

[2.4.](#_heading=h.3rdcrjn) Restricciones 5

[2.5.](#_heading=h.26in1rg) Suposiciones y Dependencias 6

[2.6.](#_heading=h.lnxbz9) Requisitos Futuros 6

[**3.**](#_heading=h.35nkun2) **Requisitos Específicos 7**

[3.1](#_heading=h.1ksv4uv) Requisitos comunes de las interfaces 8

[*3.1.1*](#_heading=h.44sinio) *Interfaces de usuario 8*

[*3.1.2*](#_heading=h.2jxsxqh) *Interfaces de hardware 8*

[*3.1.3*](#_heading=h.z337ya) *Interfaces de software 8*

[*3.1.4*](#_heading=h.3j2qqm3) *Interfaces de comunicación 8*

[3.2](#_heading=h.1y810tw) Requisitos funcionales 9

[3.3](#_heading=h.4i7ojhp) Requisitos no funcionales 9

[*3.3.1*](#_heading=h.2xcytpi) *Requisitos de rendimiento 9*

[*3.3.2*](#_heading=h.1ci93xb) *Seguridad 10*

[*3.3.3*](#_heading=h.3whwml4) *Fiabilidad 10*

[*3.3.4*](#_heading=h.2bn6wsx) *Disponibilidad 10*

[*3.3.5*](#_heading=h.qsh70q) *Mantenibilidad 10*

[*3.3.6*](#_heading=h.3as4poj) *Portabilidad 10*

[3.4](#_heading=h.1pxezwc) Otros Requisitos 10

# 1. Introducción

En esta sección se proporcionará una introducción a todo el documento de Especificación de Requisitos Software (ERS). Consta de varias subsecciones: propósito, ámbito del sistema, definiciones, referencias y visión general del documento.

## 1.1. Propósito

El propósito de este documento es entregar un resumen claro y conciso a nuestro cliente, en donde se detalla los diferentes requerimientos y funcionalidades del sistema a entregar ya sean características estéticas de la interfaz del sistema como también detalles del rendimiento de este junto con los detalles y características de los usuarios que interactuaran finalmente.

## 1.2. Ámbito del Sistema

En esta subsección:

El sistema se llamará HappyLife el propósito de este sistema es principalmente dar una asesoría constante a aquellas personas que quieran cambiar su estilo de vida a uno más saludable, dándole la posibilidad de elegir según su capacidad física ejercicios que le ayudarán a mejorar día a día su condición, también HappyLife tiene la capacidad de contactar a los usuarios con profesionales de la salud tales como nutricionistas y kinesiólogos.

El principal beneficio de nuestro sistema es aquel que va dirigido hacia las propias personas que quieran utilizar nuestro sistema, ya que este le proporcionará seguimiento al progreso de cada usuario demostrando finalmente los resultados y cuánto ha avanzado.

Y como meta esperamos la masificación de este sistema ya que al masificarse no sólo estaremos contribuyendo a la mejoría de la salud de las personas sino que también esperamos un beneficio monetario gracias a la suscripción premium que tienen la opción de contratar los usuarios.

## 1.3. Definiciones, Acrónimos y Abreviaturas

## 1.4. Referencias

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS.

## 

## 1.5. Visión General del Documento

Este documento consta de un área de definición del negocio, un área de especificación de requisitos, se proporciona el detalle de los requerimientos a través de formularios de caso de uso como anexos.

# 2. Descripción General

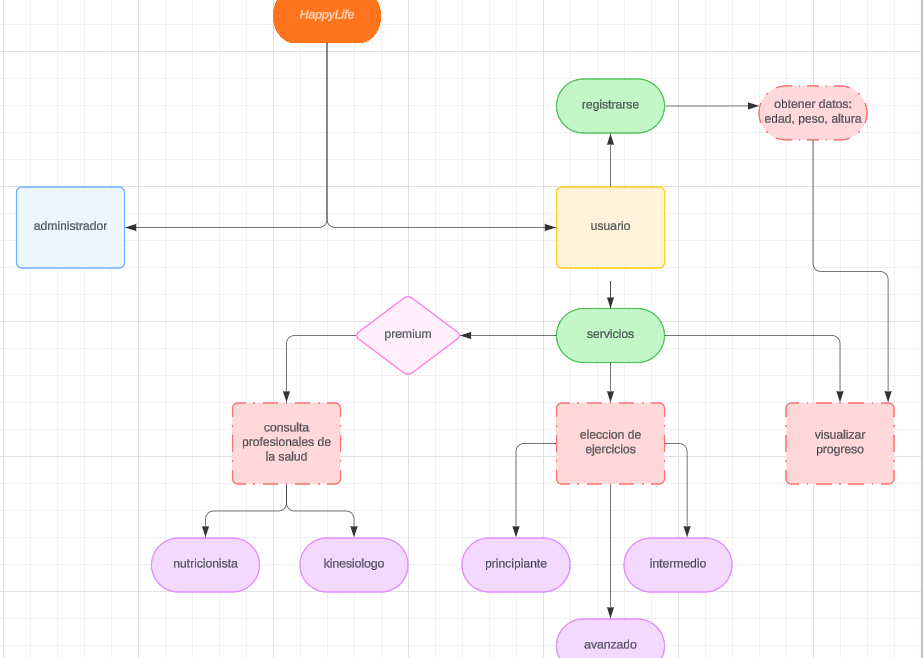
En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto**.** Esto permitirá definir con detalle los requisitos en la sección 3, haciendo que sean más fáciles de entender.

Normalmente, esta sección consta de las siguientes subsecciones: Perspectiva del producto, funciones del producto, características de los usuarios, restricciones, factores que se asumen y futuros requisitos.

## 2.1. Perspectiva del Producto

HappyLife es un sistema independiente que consta con diversas opciones que los usuarios pueden elegir dependiendo de su objetivo a cumplir, entre ellas se encuentran la selección de ejercicios físicos los cuales varían en dificultad y exigencia, los cuales son adaptables a las preferencias de los usuarios. Igualmente

HappyLife tiene también una sección en la cual los usuarios podrán hacer consultas y recibir recomendaciones de distintos profesionales de la salud para así poder tener una mejor supervisión, control y seguridad durante todo el proceso de cambio de estilo de vida, que como sabemos, en algunas personas puede ser un proceso bastante difícil



## 2.2. Funciones del Producto

**Funciones HappyLife**

**Gestión de usuario:**

* registro y autenticación de usuarios
* recuperación y gestión de contraseñas
* edición de perfil
* gestión de roles

**eleccion de servicio:**

* opción de adquirir la suscripción premium
* opción de seguir con suscripción gratuita

evaluación **inicial:**

* ingreso de datos del usuario: edad,estatura,peso
* plan de salud y ejercicios recomendados

**elección de ejercicios**

* tener un catálogo de ejercicios
* filtrado por exigencia
* video tutoriales

**seguimiento y progreso**

* seguimiento del progreso
* subida de fotos del progreso
* graficos de evolucion

**interacción con profesionales**

* los usuarios podrán enviar sus dudas o consultas que tengan a los diferentes profesionales

**notificaciones y recordatorios**:

* la app te recordará diariamente realizar los ejercicios
* la app te avisara si el profesional al cual el usuario le escribió ha contestado

**seguridad y privacidad**

* control de acceso a sus datos personales
* privacidad de datos

## 2.3. Características de los Usuarios

aquí existen 3 tipos de perfiles de usuarios, está el usuario administrador, el usuario al que va dirigido la aplicación y los profesionales de la salud

el usuario administrador deberá poseer un conocimiento en el control y gestión de usuarios entre las funciones que este tiene que poder desarrollar esta poder eliminar o suspender cuentas, verificar cuentas de los profesionales de la salud, revisión de quejas y reportes y supervisar contenido

el usuario al que va dirigida la app es aquella persona que quiere cambiar su estilo de vida sedentario, dicho usuario solo debe conocer el uso básico de móvil nivel usuario

el usuario profesional de la salud deber tener y comprobar estudios superiores en el área de la salud específicamente solicitada, en este caso debe ser un kinesiólogo o nutricionista el cual debe atender dudas y asesorar a los usuarios que lo requieran

## 2.4. Restricciones

Esta subsección describe aquellas limitaciones que se imponen sobre los desarrolladores del producto:

• **Políticas de la empresa.**

* en cuanto a la política de la empresa, se solicita que las normas de desarrolla en cuanto a seguridad, privacidad y calidad sigan la norma ISO 25010, cumpliento con la confidencialidad de los datos, la responsabilidad, completitud funcional y todos aspectos necesarios para garantizar el correcto funcionamiento y seguridad de nuestros clientes en nuestra app

• **Limitaciones del hardware.**

* debe ser compatible con dispositivos móviles tanto con sistema operativo android e IOS

• **Interfaces con otras aplicaciones.**

* se puede implementar apis y servicios externos como por ejemplo los servicios de pagos como transbank, paypal u otros servicios de pago como también servicios de mensajería, los cuales se deben ajustar a las capacidades y costos acordados

• **Operaciones paralelas.**

* La app debe funcionar correctamente incluso cuando una cantidad de usuarios intentan iniciar sesión o realizar las mismas funciones o acciones dentro de la app al mismo tiempo teniendo cuidado de que no se sobrecargue el servidor.

• **Funciones de control.**

* debe existir en el sistema un sistema de permisos y roles, en la cual cada rol pueda realizar acciones determinadas y permitidas según su rol
* también debe existir un control de errores y validaciones

• **Lenguaje(s) de programación.**

* los desarrolladores pueden utilizar el lenguaje de programación que más se les acomode ya sea tanto en frontend como en backend, mientras se cumpla con las funcionalidades del sistema y la calidad exigida

• **Criticidad de la aplicación.**

* esta debe tener una alta disponibilidad, es decir, es necesario que la app esté disponible de manera continua requiriendo una escalabilidad de los servidores y la base de datos

• **Consideraciones acerca de la seguridad.**

* la seguridad debe ser muy importante ya que manejamos datos personales de nuestros clientes por lo que la comunicación entre el servidor y los clientes deben estar cifradas utilizando protocolos https y tls, también utilizar sistema de autenticación para asegurar que solo los usuarios autorizados según su rol puedan acceder a determinadas áreas del producto

## 2.5. Suposiciones y Dependencias

una suposición en torno a este sistema son los sistemas operativos en los que funcionara y se desplegará, los cuales son IOS y android

mientras que una dependencia seria el caso de que se decidiera expandirse a windows o alguna otra plataforma ya que se tendrían que cambiar ajustes de desarrollo, diseño y requisitos

otra suposición ahora en cuanto a la infraestructura de la base de datos es que este sistema desplegará su base de datos utilizando servicios en la nube, ya sea AWS o microsoft azure

y una dependencia sería que si la empresa decida cambiar el proveedor de servicios en la nube o decide tener una estructura de servidores locales cambiando y afectando los requisitos en cuanto a seguridad,escalabilidad,gestión de datos,etc

ahora una suposición en cuanto a los trabajadores profesionales de la salud que serían parte de nuestro equipo es que sean trabajadores independientes

y una dependencia en torno a esto seria que estén afiliados a alguna clínica u hospital, los cuales podrían ser integrados, logrando una asociación con estos cambiando así los flujos de trabajo, la jerarquía, permisos, creando nuevos roles dentro de la app

## 2.6. Requisitos Futuros

una posible mejora en nuestro sistema es en cuanto a la atención y experiencia de nuestros usuarios ya que planeamos futuramente implementar videollamadas con nuestros profesionales asociados para que asi puedan dar una mejor y más precisa atención

otra mejora sería implementar según los objetivos de nuestros usuarios, poder recomendarles una dieta que les ayude a lograr sus objetivos más fácilmente y facilitarle el camino en este proceso, obviamente dieta armada y revisada por nuestro profesionales

dependiendo del impacto de nuestra app y la cantidad de usuarios nuevos que se nos presenten seria ideal mejorar y ampliar nuestros servidores de base de datos

de acuerdo a los comentarios y reseñas de nuestros usuarios ir implementando mejoras en la interfaz para que sea más amigable e intuitiva para los usuarios de mayor edad quienes no tengan tanta familiaridad con la tecnología

sección de logros en la cual el usuario a ir progresando en sus ejercicios, peso ideal, cantidad de calorías o algún factor que comprueba que ha ido cumpliendo su objetivo se le dará una series de logros para que este se pueda seguir motivando y llegar a su meta

implementación de inteligencia artificial la cual según los datos ingresados por el usuario pueda recomendarle un plan ideal de ejercicios y alimentación

# 3. Requisitos Específicos

Esta sección contiene los requisitos a un nivel de detalle suficiente como para permitir a los diseñadores diseñar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos. Todo requisito aquí especificado describirá comportamientos externos del sistema, perceptibles por parte de los usuarios, operadores y otros sistemas. Esta es la sección más larga e importante de la ERS. Deberán aplicarse los siguientes principios:

• El documento debería ser perfectamente legible por personas de muy distintas formaciones e intereses.

• Deberán referenciarse aquellos documentos relevantes que poseen alguna influencia sobre los requisitos.

• Todo requisito deberá ser unívocamente identificable mediante algún código o sistema de numeración adecuado.

• Lo ideal, aunque en la práctica no siempre realizable, es que los requisitos posean las siguientes características:

* **Corrección**: La ERS es correcta si y sólo si todo requisito que figura aquí (y que será implementado en el sistema) refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
* **No ambiguos**: Cada requisito tiene una sola interpretación. Para eliminar la ambigüedad inherente a los requisitos expresados en lenguaje natural, se deberán utilizar gráficos o notaciones formales. En el caso de utilizar términos que, habitualmente, poseen más de una interpretación, se definirán con precisión en el glosario.
* **Completos**: Todos los requisitos relevantes han sido incluidos en la ERS. Conviene incluir todas las posibles respuestas del sistema a los datos de entrada, tanto válidos como no válidos.
* **Consistentes**: Los requisitos no pueden ser contradictorios. Un conjunto de requisitos contradictorio no es implementable.
* **Clasificados**: Normalmente, no todos los requisitos son igual de importantes. Los requisitos pueden clasificarse por importancia (esenciales, condicionales u opcionales) o por estabilidad (cambios que se espera que afecten al requisito). Esto sirve, ante todo, para no emplear excesivos recursos en implementar requisitos no esenciales.
* **Verificables**: La ERS es verificable si y sólo si todos sus requisitos son verificables. Un requisito es verificable (testable) si existe un proceso finito y no costoso para demostrar que el sistema cumple con el requisito. Un requisito ambiguo no es, en general, verificable. Expresiones como a veces, bien, adecuado, etc. Introducen ambigüedad en los requisitos. Requisitos como “en caso de accidente la nube tóxica no se extenderá más allá de 25 Km" no es verificable por el alto costo que conlleva.
* **Modificables**: La ERS es modificable si y sólo si se encuentra estructurada de forma que los cambios a los requisitos pueden realizarse de forma fácil, completa y consistente. La utilización de herramientas automáticas de gestión de requisitos facilitan enormemente esta tarea.
* **Trazables**: La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica que componentes del sistema son los que realizan el requisito R.

## 3.1 Requisitos comunes de las interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

MÓDULO DE REGISTRO Y VALIDACIÓN:

ENTRADAS:

* Datos de registro del usuario: nombre, email, contraseña, fecha de nacimiento, género, altura, peso inicial.
* Credenciales de acceso: email y contraseña.
* Preferencias de privacidad (por ejemplo, aceptar términos y condiciones).

SALIDAS:

* Confirmación del registro exitoso.
* Mensajes de error (credenciales incorrectas, email ya registrado, etc.).

MODULO EDITAR PERFIL DE USUARIO:

ENTRADAS:

* Datos personales adicionales: objetivo de salud (perder peso, ganar masa muscular), nivel de actividad física, restricciones alimenticias.
* Imágenes o fotos de perfil.

SALIDAS

* Visualización del perfil actualizado.
* Mensajes de error o advertencias (por ejemplo, si no se completaron todos los campos obligatorios).

MODULO SELECCIÓN DE EJERCICIOS

ENTRADAS:

* Selección del nivel de dificultad (principiante, intermedio, avanzado).
* Frecuencia semanal y duración de los ejercicios.

SALIDAS:

* Instrucciones detalladas de los ejercicios seleccionados.
* Video o imágenes que muestren cómo realizar correctamente los ejercicios.
* Notificaciones sobre el inicio o fin de una sesión de entrenamiento.
* Registro de progreso diario/semanal.
* Alertas de recordatorio para entrenamientos futuros.

MÓDULO COMUNICACIÓN PROFESIONALES DE LA SALUD

ENTRADAS:

* Solicitud de consulta (usuario selecciona el tipo de profesional: nutricionista, kinesiólogo, entrenador).
* Consulta o preguntas por chat

SALIDAS:

* Confirmación de la solicitud de consulta.
* Respuestas y recomendaciones de los profesionales.
* Recetas, dietas o ejercicios enviados por el profesional.

MODULO PROGRESO USUARIO

ENTRADAS:

* Fotos de progreso del usuario (opcional).
* Peso y medidas corporales actualizadas.
* Diario personal de ejercicios y alimentación

SALIDAS:

* Gráficos y estadísticas que muestren el progreso
* Recomendaciones de ajustes en el plan de entrenamiento o nutrición.
* Notificaciones de progreso significativo o alertas si no se alcanzan objetivos establecidos.

MÓDULO DE NOTIFICACIONES:

ENTRADAS:

* Preferencias del usuario sobre el tipo y frecuencia de notificaciones.

SALIDAS:

* Recordatorios de los próximos entrenamientos.
* Actualizaciones sobre el progreso o logros obtenidos.

MÓDULO PAGOS Y SUSCRIPCIONES

ENTRADAS:

* Datos de tarjeta de crédito o cuenta de pago
* Selección del tipo de suscripción
* Confirmación de pago.

SALIDAS:

* Confirmación de suscripción exitosa.
* Factura electrónica o recibo de pago.

### 3.1.1 Interfaces de usuario

el cliente quiere que la interfaz de usuario sea simple pero a la vez intuitiva y que los elementos estén bien distribuidos y organizados, la pantalla principal que se mostrará será el login el cual solicitará el email de usuario previamente registrado y su contraseña junto con el botón de ingresar debajo de este formulario como tambien un boton de registrarse si es que no se tiene alguna cuenta

el la pantalla de registrarse, al usuario se le solicitara su email, su nombre, apellido, su fecha de nacimiento, su contraseña y nuevamente se le solicitará repetir su contraseña, completando dichos campos se le llevará de nuevo a la pantalla de login en donde inicia sesión

si inicia sesión correctamente se le solicitarán los datos de perfil, los cuales son su nombre, edad,peso,estatura, a que peso quiere llegar, si quiere perder grasa, ganar musculo o mantenerse, si tiene problemas médicos, si tiene alguna limitante en su alimentación y completando esto se le entregará un tiempo recomendado en la cual lograr su objetivo junto con los ejercicios recomendados

ya en la pantalla de selección de ejercicios se despliega un menú superior en el cual podrá revisar su perfil,logros,ejercicios favoritos, consultar con un profesional y gráficos

en el cuerpo de la página se desplegará las rutinas de ejercicios disponible según su dificultad que haya elegido el usuario en el filtro de ejercicios

ya dentro de la rutina, por cada ejercicio se desplegará un video tutorial de como hacerlo junto con un cronómetro que tomará el tiempo.

si elige la opción de comunicarse con un profesional habrá dos cards con las opciones de kinesiólogo o nutricionista, posteriormente de haber elegido se le abrirá un chat con el cual puede enviar dudas y consultas al profesional y cuando este responda le llegar una notificación al usuario

si elige la opcion de logros se le mostrara todos los logros con iconos los cuales se mostraran en color los que ha cumplido el usuario y en blanco y negro los que todavia no ha logrado

y si selecion los gráfico, igualmente tendrá que menú superior como en todas las visas de usuarios y se desplegaran los gráficos de progreso en cuanto a su peso, nivel de ejercicio, grupo musculares trabajados,etc

### 3.1.2 Interfaces de hardware

interfaz de dispositivos móviles:

* compatibilidad con pantallas táctiles
* diseño responsive y adaptable
* acceso al teclado virtual
* gestor táctil
* necesita tener vibraciones para avisar de los eventos dentro de la app

### 3.1.3 Interfaces de software

* por ahora no tenemos planeado integrar ningún software externo

### 3.1.4 Interfaces de comunicación

Describir los requisitos de interfaces de comunicación si hay comunicaciones con otros sistemas y cuales son los protocolos de comunicación.

COMUNICACIÓN CON EL SISTEMA DE ALMACENAMIENTO DE LA NUBE

requisitos:

* La aplicación debe poder almacenar datos de usuarios (como su información personal, fotos del progreso, peso, etc) en un servicio de almacenamiento en la nube.

protocolos de comunicación:

* Amazon S3, Google Cloud Storage o Azure Blob Storage: Utilizan HTTPS para la transmisión segura de datos y la autenticación mediante tokens de acceso, OAuth 2.0 o claves API.
* El sistema debe soportar la transferencia de archivos a través de HTTP PUT/POST y el acceso a los mismos mediante GET, con políticas de seguridad como cifrado en tránsito (SSL/TLS) y en reposo.

COMUNICACIÓN CON SERVICIOS DE ALETAS Y COMUNICACIONES

requisitos:

* La aplicación debe enviar notificaciones push a los usuarios para alertarlos sobre nuevos mensajes, recordatorios de ejercicios o consultas programadas.
* Las notificaciones deben ser recibidas incluso cuando la aplicación no esté activa.

protocolo de comunicación

* Firebase Cloud Messaging (FCM) para dispositivos Android: Utiliza el protocolo HTTP/HTTPS para el envío de notificaciones y el protocolo XMPP (Extensible Messaging and Presence Protocol) para comunicaciones persistentes en tiempo real.
* Apple Push Notification Service (APNS) para dispositivos iOS: Utiliza HTTP/2 para el envío de notificaciones, lo que asegura que los mensajes se envíen de forma rápida y segura con baja latencia.

## 3.2 Requisitos funcionales

3.2.1 Requisito funcional 1

Requerimiento funcional 1: Recomendaciones de Ejercicio Personalizadas

Actores: usuario

Descripción:El sistema debe ofrecer recomendaciones de ejercicios adaptadas al perfil del usuario..

3.2.2 Requisito funcional 2

Requerimiento funcional 1: Registro y Monitoreo del Progreso de Salud

Actores: usuario

Descripción:La aplicación permite registrar y monitorear indicadores de salud como peso, presión arterial y glucosa.

3.2.3 Requisito funcional 3

Requerimiento funcional 1: Acceso a Expertos

Actores: usuario

Descripción:El usuario puede acceder a entrenadores o médicos para recibir consejos personalizados.

3.2.4 Requisito funcional 4

Requerimiento funcional 1: Notificaciones Inteligentes

Actores: usuario

Descripción:El sistema enviará notificaciones basadas en el progreso del usuario, recordatorios y sugerencias.

3.2.5 Requisito funcional 5

Requerimiento funcional 1: Gestión de Fotos de Usuario

Actores: usuario

Descripción:El usuario podrá tomar y gestionar fotos mensuales para registrar su progreso físico.

3.2.6 Requisito funcional 6

Requerimiento funcional 1: Registro de Clientes y Pago

Actores: usuario

Descripción:El sistema permitirá registrar nuevos clientes y procesar pagos de suscripciones o servicios adicionales.

3.2.6 Requisito funcional 7

Requerimiento funcional 1: Manejos de informes de Administrador

Actores: administrador

Descripción:Los administradores podrán generar y visualizar informes de rendimiento y uso de la plataforma.

3.2.6 Requisito funcional 7

Requerimiento funcional 1: Manejos de contratos Entrenadores

Actores: administrador

Descripción: El sistema permitirá gestionar los contratos de los entrenadores con los usuarios y la plataforma.

3.3 Requisitos no funcionales

3.3.1 requisito no funcional 1

Requerimiento no funcional 1: Seguridad y Privacidad

Actores: usuario

Descripción: La aplicación debe cumplir con los estándares más estrictos de seguridad y privacidad de datos, incluyendo cifrado y protección de datos personales.

3.3.2 requisito no funcional 2

Requerimiento no funcional 2: Compatibilidad

Actores: usuario

Descripción: La aplicación debe ser compatible con los sistemas operativos móviles y navegadores web más recientes.

3.3.3 requisito no funcional 3

Requerimiento no funcional 3: Documentación y Soporte

Actores: usuario,administrador

Descripción: Debe proporcionar documentación clara y completa, además de soporte técnico y asistencia al cliente para resolver problemas y consultas.

3.3.4 requisito no funcional 4

Requerimiento no funcional 4: Rendimiento y Escalabilidad

Actores: usuario,administrador

Descripción: La aplicación debe manejar un gran número de usuarios simultáneamente sin pérdida de rendimiento, con tiempos de respuesta rápidos.

3.3.5 requisito no funcional 5

Requerimiento no funcional 5: Usabilidad

Actores: usuario, administrador

Descripción: La aplicación debe ser intuitiva, cómoda y facil de usar para que el usuario y Administrador puedan usar la app

3.3.6 requisito no funcional 6

Requerimiento no funcional 6: Respaldo y Recuperación de Datos

Actores: usuario, administrador

Descripción: La app debe tener respaldo automático de datos y permitir recuperación en caso de fallos.

3.3.7 requisito no funcional 7

Requerimiento no funcional 7: Actualización Automática

Actores: usuario, administrador

Descripción: La app debe actualizarse automáticamente sin interrumpir al usuario.

3.3.8 requisito no funcional 8

Requerimiento no funcional 8: Fiabilidad del Sistema

Actores: usuario, administrador

Descripción:El sistema debe garantizar la correcta operación y procesamiento de datos sin errores.

3.3.9 requisito no funcional 9

Requerimiento no funcional 9: Disponibilidad del Sistema

Actores: usuario, administrador

Descripción:La aplicación debe estar disponible al menos el 99.9% del tiempo.

3.3.10 requisito no funcional 10

Requerimiento no funcional 10: Mantenibilidad del Sistema

Actores: usuario, administrador

Descripción:Debe permitir actualizaciones y mantenimiento sin interrumpir el servicio..

### 3.3.1 Requisitos de rendimiento

#### 3.3.1.1 Requisito de Rendimiento 1: Usuarios simultáneos

* **Descripción**: El sistema debe soportar una alta cantidad de usuarios conectados simultáneamente sin afectar la funcionalidad y la velocidad de la aplicación.
* **Medida**:
  + Debe poder gestionar al menos **10,000 usuarios simultáneos** sin caída de rendimiento.
  + En condiciones pico, debe ser capaz de soportar hasta **15,000 usuarios simultáneos** con un tiempo de respuesta dentro de los límites aceptables (ver especificación de tiempo de respuesta más adelante).

#### 

#### 

#### 3.3.1.2 Requisito de Rendimiento 2: Tiempo de respuesta

* **Descripción**: El sistema debe ofrecer tiempos de respuesta rápidos y consistentes para las acciones realizadas por el usuario.
* **Medida**:
  + El **95%** de las transacciones realizadas por los usuarios deben completarse en **menos de 1 segundo**.
  + El **99%** de las transacciones deben completarse en **menos de 2 segundos**.

#### 3.3.1.3 Requisito de Rendimiento 3: Número de transacciones por segundo

* **Descripción**: El sistema debe ser capaz de procesar múltiples transacciones por segundo para mantener la eficiencia operativa.
* **Medida**:
  + El sistema debe ser capaz de manejar al menos 500 transacciones por segundo sin errores ni degradación del rendimiento.
  + En horas de alta demanda, debe poder soportar hasta 1000 transacciones por segundo de manera eficiente.

#### 3.3.1.4 Requisito de Rendimiento 4: Rendimiento de las notificaciones

* **Descripción**: Las notificaciones inteligentes deben enviarse de manera oportuna en función del comportamiento y progreso del usuario.
* **Medida**:
  + Las notificaciones deben enviarse en **menos de 2 segundos** desde que el sistema detecta que una notificación debe ser enviada.
  + El sistema debe ser capaz de manejar **al menos 10,000 notificaciones simultáneas** sin afectar el rendimiento global de la aplicación.

#### 3.3.1.5 Requisito de Rendimiento 5: Uso de memoria y CPU

* **Descripción**: El sistema debe optimizar el uso de recursos (memoria y CPU) para no afectar la experiencia del usuario.
* **Medida**:
  + El uso promedio de CPU no debe superar el **50%** de su capacidad total en condiciones normales de uso, con picos de hasta el **80%** en situaciones de alta demanda.
  + El uso de memoria no debe exceder el **75%** de la capacidad disponible en condiciones normales, con un margen máximo de **90%** en momentos de carga máxima.

#### 3.3.1.6 Requisito de Rendimiento 6: Capacidad de almacenamiento

* **Descripción**: La aplicación debe ser capaz de almacenar y gestionar grandes volúmenes de datos de los usuarios (fotos de progreso, datos de salud, etc.).
* **Medida**:
  + El sistema debe poder almacenar y gestionar al menos 10TB de datos en su fase inicial.
  + Debe ser escalable para soportar hasta 100TB de almacenamiento sin interrupciones ni caídas de rendimiento.

#### 3.3.1.7 Requisito de Rendimiento 7: Copia de seguridad y recuperación

* **Descripción**: El sistema debe realizar copias de seguridad automáticas de los datos y permitir una recuperación rápida en caso de fallo.
* **Medida**:
  + El sistema debe realizar copias de seguridad cada 24 horas de manera automática.
  + La recuperación de datos debe completarse en menos de 10 minutos en el 95% de los casos de pérdida de datos o fallos.

### 3.3.2 Seguridad

Respaldo y Recuperación de Datos

La app debe tener respaldo automático de datos y permitir recuperación en caso de fallos.

### 3.3.3 Fiabilidad

La fiabilidad de la aplicación debe garantizar la operación continua y correcta del sistema con un mínimo de interrupciones. A continuación se detallan los factores de fiabilidad que debe cumplir el sistema:

en el caso de nuestra sistema, El sistema debe garantizar la correcta operación y procesamiento de datos sin errores.

### 3.3.4 Disponibilidad

El sistema debe estar disponible para los usuarios la mayor parte del tiempo, minimizando el tiempo de inactividad planificado y no planificado.

La aplicación debe estar disponible al menos el 99.9% del tiempo.

### 3.3.5 Mantenibilidad

en este caso debemos realizar dos tipos de mantenimiento, uno correctivo y uno preventivo

el mantenimiento correctivo se realiza para corregir fallos o errores detectados en el sistema, los responsables de llevarlo a cabo son los desarrolladores o el equipo técnico, realizando cada vez que se detecte algún problema o fallo en el sistema

El mantenimiento preventivo se realiza para evitar la ocurrencia de fallos mediante la actualización de software y la optimización del sistema, los encargados de llevarlo a cabo igualmente son los desarrolladores y debe realizarse al menos una vez cada 3 meses para actualizar bibliotecas, verificar vulnerabilidades y optimizar el rendimiento.

### 3.3.6 Portabilidad

obviamente el sistema debe estar construido sobre una base responsiva ya que de no ser asi, no asegura un correcto funcionamiento en las distintas plataformas

la arquitectura del sistema debe ser construida basada en componentes modulares El sistema debe estar diseñado de manera modular, lo que permite que cada componente sea independiente y fácilmente modificable o reemplazable facilitando la migración de este

También minimizar la cantidad de código dependiente del servidor específico (backend) para que sea adaptable a otros entornos.

en cuanto al lenguaje de programación,compiladores y programas de desarrollo igualmente tienen que ser portables