| DUOC UC - Escuela de informática y telecomunicaciones |
| --- |
| Propuesta de Proyecto y Especificación de Requisitos de Software |
| *Proyecto: VitaLink* |
|  |
| **Revisión*: [01]*** |
| **[Seleccionar fecha]** |

| Planificación y Especificación de Requisitos según estándares; IEEE 830, ISO9000 y PMI. |
| --- |

**Contenido**

[*DUOC UC - Escuela de informática y telecomunicaciones 1*](#_heading=h.t18corti8gvz)

[**Ficha del documento 4**](#_heading=h.li41pqfl07vw)

[**1. Introducción 6**](#_heading=h.ixg7hktp8l76)

[1.1. Propósito 6](#_heading=h.qkd3eja47e3u)

[1.2. Ámbito del Sistema 7](#_heading=h.r9xo6iakuo7m)

[1.3. Definiciones, Acrónimos y Abreviaturas 8](#_heading=h.kwq519r2z22j)

[1.4. Referencias 8](#_heading=h.1mqd2dfcwvgn)

[1.5. Visión General del Documento 8](#_heading=h.70gb0v2dr1ki)

[**2. Descripción General 9**](#_heading=h.oup1hxvblvxr)

[2.1. Perspectiva del Producto 10](#_heading=h.aul7czcrvfuc)

[2.2. Funciones del producto 12](#_heading=h.xoqasrrxn4u9)

[2.3. Características de los Usuarios 13](#_heading=h.pne71h3uf987)

[2.4. Restricciones 14](#_heading=h.edjnd4abuxma)

[1. Políticas de la empresa 14](#_heading=h.aneql151vhay)

[2. Limitaciones de hardware 14](#_heading=h.90q2eqha5kl9)

[3. Interfaces con otras aplicaciones 15](#_heading=h.wfhutfaude42)

[4. Operaciones paralelas 15](#_heading=h.p8li53vmnn7w)

[5. Funciones de auditoría y control 15](#_heading=h.w2b52gk5fekm)

[6. Lenguajes de programación y frameworks 15](#_heading=h.71vq49npjkbn)

[7. Protocolos de comunicación 15](#_heading=h.l7a6llduo8sg)

[8. Requisitos de habilidad 16](#_heading=h.pdb974xrs3v)

[9. Criticidad de la aplicación 16](#_heading=h.ke7gw89820gw)

[10. Consideraciones de seguridad 16](#_heading=h.e23lqsqrh6vc)

[2.5. Suposiciones y Dependencias 16](#_heading=h.ecjfhfavu0)

[2.6. Requisitos Futuros 18](#_heading=h.bfpqnt9j2rt4)

[**3. Requisitos Específicos 20**](#_heading=h.2q5qi5yp80qc)

[3.1 Requisitos comunes de las interfaces 20](#_heading=h.u8l9xk925vbc)

[*3.1.1 Interfaces de usuario 20*](#_heading=h.tt4d9g6r2kno)

[*3.1.2 Interfaces de hardware 20*](#_heading=h.b0x6g629ejyg)

[*3.1.3 Interfaces de software 20*](#_heading=h.cwu0fvcxffru)

[3.2 Requisitos funcionales 22](#_heading=h.chvfk4kdhd9q)

[3.3 Requisitos no funcionales (Organización) y de calidad (Producto) 23](#_heading=h.mufmowoo7eaz)

[*3.3.1 Requisitos de Rendimiento o Eficiencia 23*](#_heading=h.yc84sfas86n9)

[*3.3.2 Requisitos de Seguridad 23*](#_heading=h.qfgi1urariu6)

[*3.3.3 Requisitos de Usabilidad 24*](#_heading=h.g6ps3mj8bmio)

[*3.3.4 Requisitos de Disponibilidad 24*](#_heading=h.5o9ws6xozyh1)

[*3.3.5 Requisitos de Portabilidad 24*](#_heading=h.udvv4kejwvk8)

[*3.3.6 Requisitos de Mantenibilidad 24*](#_heading=h.1k3qu27qafpl)

[*3.3.7 Requisitos de Funcionalidad 24*](#_heading=h.teqndnxwozar)

[3.4 Requisitos No funcionales Organizacionales 25](#_heading=h.fzr89dvfsck7)

[**4. Propuesta de Planificación 26**](#_heading=h.o46g7eu926vb)

[4.1 Descripción general acerca de la Planificación 26](#_heading=h.tjbw96ply7h2)

[*4.1.2 Definición del Equipo de Trabajo 27*](#_heading=h.75kkccj3tww)

[*4.1.3 Definición de Actividades principales del Proyecto 28*](#_heading=h.te36nzomuf2q)

[*4.1.4 Resumen Costos del Desarrollo del Proyecto 28*](#_heading=h.fzmw7wtmfl7i)

[*4.1.6 Carta Gantt 30*](#_heading=h.sohhqjfk9rw2)

[5. Anexos 31](#_heading=h.6a0ad71h8zhg)

[*5.1 Acta de Proyecto 31*](#_heading=h.ppvri518wcew)

[*5.2 Matriz Especificación de Requerimientos 31*](#_heading=h.yralcasoyh3t)

[*5.3 Prototipo de Software 31*](#_heading=h.kkuzz1s7dty8)

[*5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo 31*](#_heading=h.dgx9vmbmhs5r)

[*5.5 Planilla Carta Gantt 31*](#_heading=h.7hex9xglla3n)

# Ficha del documento

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
|  |  |  |  |
| *07-09-2025* | *v1.1* | *Juan Inostroza* | *Redacción profesional y estructuración de la Introducción del documento.* |
| *07-09-2025* | *v1.2* | *Juan Inostroza* | *Mejora de la redacción y coherencia del apartado 1.1 Propósito.* |
| *07-09-2025* | *v1.3* | *Juan Inostroza* | *Reestructuración completa de 1.2 Ámbito del sistema con lenguaje técnico y mayor detalle funcional.* |
| *07-09-2025* | *v1.4* | *Juan Inostroza* | *Expansión y enriquecimiento de la 2.1 Perspectiva del producto, incluyendo arquitectura, interoperabilidad y tecnologías base.* |
| *07-09-2025* | *v1.5* | *Juan Inostroza* | *Redacción profesional de la sección 2.3 Características de los usuarios, definiendo perfiles, niveles educacionales, experiencia técnica y responsabilidades.* |
| *07-09-2025* | *v1.6* | *Juan Inostroza* | *Creación de la tabla resumen de funciones del producto, detallando nombre, descripción, usuario responsable y módulo asociado para mejorar la claridad y trazabilidad de los requisitos funcionales.* |

Documento validado por las partes en fecha:

**Integrantes:**

| **Nombre Integrante del Equipo** | **Rol Definido** |
| --- | --- |
| *Diego Kent Lahana* | *Jefe de Proyecto* |
| *Bastián Nahuel* | *Analista Programador, Diseñador, Ingeniero de Sistemas.* |
| *Juan Inostroza* | *Analista Programador, Calidad y Documentación. Tester QA,responsable de documentación técnica y Gestor de calidad, DBA.* |
| *Martin Salinas* | *Analista de Control de Calidad (QA) - Tester QA,Responsable del diseño e implementación del plan de pruebas y casos de prueba.* |

# 1. Introducción

El presente documento de **Especificación de Requisitos de Software (ERS)** describe de manera detallada los objetivos, alcance, funcionalidades y restricciones del sistema **VitaLink**. Este sistema integral está diseñado para **modernizar la gestión de consultorios médicos** mediante tres componentes principales:

* **Aplicación móvil para pacientes**
* **Plataforma web para profesionales de la salud**

VitaLink centraliza y optimiza el **registro y seguimiento de crisis de pánico**, mejorando la **continuidad de los tratamientos de salud mental** a través del **análisis de datos en tiempo real** y una **comunicación eficiente entre pacientes y médicos**. Además, busca **reducir el abandono terapéutico** y **proporcionar herramientas digitales accesibles** que permitan una atención más **personalizada, efectiva y segura**.

Este documento sirve como referencia para todas las partes interesadas, estableciendo una visión clara y compartida del sistema a desarrollar, así como los criterios necesarios para su correcta implementación y validación.

## 1.1. Propósito

Este documento define formalmente los requisitos funcionales y no funcionales del sistema VitaLink, estableciendo las bases para su diseño, desarrollo y validación. Está dirigido al equipo técnico (desarrolladores, diseñadores, testers), stakeholders clave y usuarios finales, con el objetivo de alinear expectativas, guiar la implementación y servir como referencia contractual durante todas las fases del proyecto. Su elaboración garantiza que el producto final cumpla con los objetivos de negocio y las necesidades reales de los usuarios del consultorio.

## 

## 1.2. Ámbito del Sistema

**Nombre del sistema:** VitaLink

**Funcionalidades principales:**

* **Gestión de usuarios:** registro, modificación y administración de perfiles de pacientes
* **Consulta de métricas de salud:** acceso a estadísticas, tendencias y reportes personalizados.
* **Recursos de ayuda inmediata:** provisión de material de contención y asistencia rápida ante crisis de pánico.
* **Gestión clínica para médicos:** registro, actualización, eliminación y visualización de historiales clínicos, métricas y perfiles de pacientes.
* **Notificaciones en tiempo real:** envío de alertas y recordatorios tanto para pacientes como para profesionales.

**Exclusiones del sistema:**

* **No reemplaza los sistemas hospitalarios existentes** de gestión integral, sino que **se integra con ellos mediante APIs**.
* **No realiza diagnósticos médicos automáticos**, limitándose a proporcionar datos e información para apoyar la toma de decisiones clínicas.
* **No reemplaza al profesional,** sirviendo como herramienta bidireccional para mejorar la comunicación paciente-profesional.

**Beneficios esperados:**

* Mejora la **continuidad de los tratamientos** de salud mental.
* Proporciona una **base de datos eficiente y centralizada**.
* Ofrece una **interfaz intuitiva** para pacientes, médicos y administradores.
* Facilita el **acceso a información en tiempo real**, permitiendo **decisiones médicas más precisas**.
* **Fideliza** al paciente, reduciendo la posibilidad de que el mismo abandone su terapia antes de terminar su tratamiento correspondiente.

## 1.3. Definiciones, Acrónimos y Abreviaturas

* **ERS:** Especificación de Requisitos de Software
* **App:** Aplicación
* **UX:** User Experience (Experiencia de Usuario)
* **UI:** User Interface (Interfaz de Usuario)
* **Supabase:** Plataforma de base de datos basada en PostgreSQL
* **React Native:** Framework para desarrollo de aplicaciones móviles
* **Django:** Framework para desarrollo web en Python
* **JWT:** JSON Web Token para autenticación
* **SW:** Servicio Web

## 

## 1.4. Referencias

**Tecnologías**

* [REF-101] Supabase Docs. Auth, Postgres, RLS.
* [REF-102] PostgreSQL Docs v15.
* [REF-103] React Native Docs y Django Docs
* [REF-104] Git/GitFlow. Convenciones de ramas y versionado semántico.

## 1.5. Visión General del Documento

El presente documento de **Especificación de Requisitos de Software (ERS)** está organizado en cinco secciones principales que permiten comprender de forma clara el alcance, las funcionalidades y la planificación del sistema **VitaLink**:

* **Sección 1 — Introducción:** Describe el propósito del documento, el alcance del sistema, las definiciones relevantes, las referencias utilizadas y esta visión general del contenido.
* **Sección 2 — Descripción General:** Expone el contexto en el que operará **VitaLink**, incluyendo su perspectiva dentro del ecosistema actual, las funciones principales, las características de los usuarios, las restricciones, los supuestos asumidos y los posibles requisitos futuros.
* **Sección 3 — Requisitos Específicos:** Detalla de forma precisa los **requisitos funcionales** y **no funcionales** del sistema, así como los casos de uso necesarios para garantizar su correcto desarrollo e implementación.
* **Sección 4 — Propuesta de Planificación:** Presenta la planificación del proyecto, incluyendo las fases de desarrollo, hitos, cronograma estimado y recursos necesarios para la implementación exitosa del sistema.
* **Sección 5 — Anexos:** Reúne información complementaria, diagramas, glosarios, referencias normativas y cualquier material de apoyo relevante para la comprensión del documento.

Esta estructura permite que el ERS funcione como una **guía integral** para desarrolladores, diseñadores, testers, stakeholders y usuarios clave, asegurando que todos los involucrados compartan una **visión común y alineada** del sistema.

# 2. Descripción General

En esta sección se presenta una visión global del sistema **VitaLink** y de todos los **factores que influyen en la definición de sus requisitos**. No se detallan aún los requisitos específicos, sino que se describe el **contexto funcional, técnico y operativo** en el que el sistema será desarrollado e implementado.

El objetivo de esta sección es **proporcionar el marco de referencia necesario** para que, en la **sección 3**, los requisitos puedan definirse de forma más clara, precisa y comprensible para todas las partes interesadas.

La **Descripción General** se organiza en varias subsecciones que abarcan distintos aspectos clave:

* **Perspectiva del producto (2.1):** Describe la relación de **VitaLink** con otros sistemas, su arquitectura general y su nivel de integración con plataformas externas.
* **Funciones del producto (2.2):** Resume las principales funcionalidades del sistema y cómo éstas satisfacen las necesidades de los distintos tipos de usuarios.
* **Características de los usuarios (2.3):** Define los perfiles de los usuarios finales, su nivel educacional, experiencia técnica y responsabilidades dentro del sistema.
* **Restricciones (2.4):** Establece las limitaciones técnicas, normativas y operativas que afectan el diseño y la implementación.
* **Suposiciones y dependencias (2.5):** Enumera los factores asumidos para el desarrollo del sistema y las dependencias externas que podrían impactar su funcionamiento.
* **Requisitos futuros (2.6):** Identifica las posibles funcionalidades o extensiones que podrían incorporarse en versiones posteriores.

Con esta estructura, se asegura que el diseño y desarrollo de **VitaLink** partan de una comprensión clara de su **contexto operativo**, **interacciones con otros sistemas**, **usuarios involucrados** y **condiciones que afectan su implementación**.

## 2.1. Perspectiva del Producto

**VitaLink** es un **producto software independiente** diseñado para **modernizar y optimizar la gestión de consultorios médicos**, con un enfoque principal en la **salud mental** y el **seguimiento de crisis de pánico**. Si bien funciona como un sistema autónomo, **está concebido para integrarse de forma transparente** con las **plataformas de gestión hospitalaria** y otros sistemas ya existentes en los consultorios mediante **APIs RESTful seguras**.

#### Arquitectura y módulos principales

VitaLink está compuesto por tres módulos que interactúan con una **base de datos centralizada** y comparten un ecosistema único:

1. **Aplicación móvil (React Native):** Dirigida a pacientes, permite registrar crisis de pánico, consultar métricas personales, acceder a recursos de ayuda inmediata y recibir notificaciones en tiempo real.
2. **Servicio web (Django + React.js):** Orientado a médicos y administradores, facilita la gestión de historiales clínicos, el seguimiento de métricas, la visualización de estadísticas y la administración de perfiles de pacientes.

#### 

#### Interoperabilidad con sistemas externos

Aunque VitaLink no reemplaza a los sistemas hospitalarios existentes, **los complementa** mediante **integraciones seguras** basadas en **APIs RESTful**. Estas interfaces permiten:

* **Importar y sincronizar datos** desde plataformas de gestión hospitalaria.
* **Compartir métricas e historiales clínicos** con otros sistemas autorizados.
* Mantener la **consistencia y trazabilidad** de la información sin duplicar registros.

#### Base tecnológica

El sistema utiliza **Supabase** como **plataforma de base de datos centralizada**, asegurando:

* **Alta disponibilidad** y **escalabilidad** para manejar grandes volúmenes de datos.
* **Autenticación robusta** y control de accesos.
* **Sincronización en tiempo real** entre todos los módulos.

#### Resumen de integración en el ecosistema

VitaLink es un **componente independiente**, pero diseñado para **coexistir e integrarse** dentro del ecosistema tecnológico del consultorio a modo de simbiosis para nuestros potenciales clientes. Esto permite que el sistema:

* Funcione de forma autónoma para tareas específicas de salud mental.
* Aproveche la **infraestructura hospitalaria existente** para evitar duplicidad de datos.
* Se comunique con otras plataformas mediante **interfaces estandarizadas**, garantizando **compatibilidad y escalabilidad futura**.

## 

## 2.2. Funciones del producto

| **Función** | **Descripción** | **Usuario responsable** | **Módulo asociado** |
| --- | --- | --- | --- |
| Registro de crisis | Permite a los pacientes registrar episodios de pánico indicando **fecha, hora, dosis, ubicación, causas, descripción y duración.** | Paciente | Aplicación móvil |
| Consulta de crisis | Visualiza el historial de crisis registradas, con **filtros avanzados** y acceso a **detalles completos** de cada episodio. | Paciente, Médico | App móvil / Web |
| Métricas de salud | Muestra **gráficos, estadísticas y reportes dinámicos** para analizar la evolución del tratamiento. | Paciente, Médico | App móvil / Web |
| Análisis clínico | Los médicos comparan métricas actuales e históricas para **evaluar la efectividad del tratamiento.** | Médico | Plataforma web |
| Gestión de historial clínico | Permite a los médicos **visualizar, actualizar y analizar** historiales clínicos de los pacientes. | Médico | Plataforma web |
| Modificación y eliminación de registros | Posibilidad de **editar o eliminar** registros clínicos y métricas, respetando **roles y permisos** definidos. | Médico | Plataforma web |
| Gestión de perfiles | Administración de **datos personales, roles y permisos** de pacientes y médicos. | Administrador, Médico | Plataforma web |
| Notificaciones en tiempo real | Envío de **alertas push y web** sobre **nuevos registros, actualizaciones clínicas, recordatorios y eventos críticos.** | Paciente, Médico | App móvil / Web |

## 

## 2.3. Características de los Usuarios

El sistema **VitaLink** está diseñado para ser utilizado por **tres tipos principales de usuarios**, cada uno con distintos **perfiles**, **niveles de conocimiento** y **responsabilidades**. La interfaz y funcionalidades de la plataforma se adaptan a las necesidades y competencias de cada rol, garantizando una experiencia eficiente y segura.

#### A. Administrador del sistema

* **Perfil general:** Personal encargado de la gestión técnica y administrativa de la plataforma.
* **Nivel educacional:** Formación técnica o profesional en informática, ingeniería o administración de sistemas.
* **Experiencia técnica:** Conocimientos avanzados en **gestión de bases de datos**, **seguridad informática**, **auditorías de acceso** y **mantenimiento de software**.
* **Responsabilidades en VitaLink:**
  + Administrar usuarios y roles dentro del sistema.
  + Generar reportes de uso, métricas globales y estadísticas administrativas.
  + Configurar parámetros de seguridad y gestionar la integración con sistemas externos.

#### B. Médico / Psicólogo (Profesional de la salud)

* **Perfil general:** Profesionales especializados en la atención clínica y el seguimiento de pacientes con condiciones de salud mental.  
  **Nivel educacional:** Título universitario en medicina o psicología; formación de postgrado opcional en salud mental.
* **Experiencia técnica:** Competencia básica en el uso de **plataformas digitales de gestión clínica** y herramientas informáticas comunes.
* **Responsabilidades en VitaLink:**
  + Registrar, actualizar y consultar historiales clínicos.
  + Analizar métricas de salud de los pacientes.
  + Emitir recetas, generar reportes y dar seguimiento a la evolución terapéutica.
  + Recibir alertas y notificaciones relevantes en tiempo real.

#### C. Paciente

* **Perfil general:** Personas que reciben atención médica y hacen uso de la aplicación para el registro y seguimiento de su salud mental.
* **Nivel educacional:** Variable; puede ir desde educación básica hasta formación universitaria.
* **Experiencia técnica:** Manejo general de **dispositivos móviles** y aplicaciones de uso cotidiano.
* **Responsabilidades en VitaLink:**
  + Registrar información personal y de salud relevante.
  + Acceder a métricas, reportes y recursos de ayuda inmediata.
  + Recibir notificaciones, recordatorios y alertas de seguimiento.
  + Comunicarse con su médico o psicólogo a través de la plataforma.

## 2.4. Restricciones

El desarrollo del sistema **VitaLink** está sujeto a un conjunto de restricciones técnicas, operativas y organizacionales que condicionan su diseño, implementación y despliegue. A continuación, se detallan las principales limitaciones identificadas:

### 1. Políticas de la empresa

* Cumplimiento estricto de las políticas internas del consultorio respecto al manejo y confidencialidad de datos médicos.
* Alineación con protocolos internos de gestión de pacientes y tratamiento de datos sensibles.

### 2. Limitaciones de hardware

* **Aplicación móvil:** Compatibilidad con dispositivos Android (**9.0 o superior**) e iOS (**14 o superior**).
* **Servicio web:** Optimizado para equipos estándar utilizados en consultorios, con un mínimo recomendado de **4 GB de RAM** y navegadores actualizados.

### 3. Interfaces con otras aplicaciones

* Integración exclusiva mediante **APIs RESTful** con sistemas hospitalarios externos.
* No se contempla conexión con hardware médico especializado ni sistemas locales de escritorio.

### 4. Operaciones paralelas

* Soporte para **hasta 500 usuarios concurrentes** entre pacientes, médicos y administradores, garantizando la estabilidad del servicio.

### 5. Funciones de auditoría y control

* Registro obligatorio de operaciones críticas (registro, modificación y eliminación de datos) en un **log de auditoría** accesible solo para administradores autorizados.
* Gestión centralizada de roles y permisos para garantizar control de acceso seguro.

### 6. Lenguajes de programación y frameworks

* **Frontend móvil:** React Native.
* **Frontend web:** React.js.
* **Backend:** Django + Django REST Framework.
* **Base de datos centralizada:** Supabase (PostgreSQL).

### 7. Protocolos de comunicación

* Todas las comunicaciones entre cliente y servidor se realizan mediante **HTTPS** con soporte para **TLS 1.3**.

### 8. Requisitos de habilidad

* El equipo de desarrollo debe contar con experiencia en **desarrollo web, móvil y APIs RESTful**.
* Los médicos y administradores deben poseer conocimientos básicos para gestionar perfiles y consultar reportes.

### 9. Criticidad de la aplicación

* Por manejar información médica sensible, **VitaLink** es considerado un sistema **de alta criticidad**. Se exige alta disponibilidad, integridad y seguridad de los datos.

### 10. Consideraciones de seguridad

* **Autenticación JWT** para sesiones seguras.
* **Cifrado de datos sensibles** en tránsito y en reposo.
* Cumplimiento con la **Ley 19.628 sobre Protección de la Vida Privada** en Chile y normativas equivalentes.

## 2.5. Suposiciones y Dependencias

#### Suposiciones

1. **Disponibilidad de infraestructura técnica:**
   * Se asume que el consultorio contará con servidores y equipos compatibles para alojar la plataforma web.
   * Los dispositivos móviles de los pacientes y médicos cumplirán con los requisitos mínimos de hardware y sistema operativo definidos.
2. **Acceso estable a internet:**
   * Se presupone que los usuarios (pacientes, médicos y administradores) tendrán acceso estable a **conexión Wi-Fi** o **datos móviles** para garantizar el uso correcto de las funciones en tiempo real, como notificaciones push y sincronización de datos.
3. **Colaboración activa del personal médico y administrativo:**
   * Se asume que médicos, psicólogos y administradores colaborarán en la validación de requisitos, pruebas y ajustes funcionales, aportando retroalimentación para garantizar que el sistema satisfaga las necesidades reales del consultorio.
4. **Disponibilidad de APIs de integración:**
   * Se asume que los sistemas externos de gestión hospitalaria permitirán la integración mediante **APIs RESTful**. En caso contrario, será necesario rediseñar parte de la arquitectura del sistema.
5. **Protección de datos y cumplimiento normativo:**
   * Se asume que el consultorio implementará políticas alineadas con normativas de protección de datos, como la **Ley 19.628 sobre Protección de la Vida Privada** en Chile, para respaldar las medidas de seguridad del sistema.

#### Dependencias

1. **Servicios externos y base de datos:**
   * El sistema depende del servicio de **Supabase** para la gestión centralizada de la base de datos y el almacenamiento seguro de la información.
   * Cualquier cambio en la política, costos o disponibilidad de Supabase podría requerir ajustes importantes en la infraestructura del sistema.
2. **Plataformas y frameworks utilizados:**
   * **VitaLink** depende de la estabilidad y soporte continuo de las tecnologías seleccionadas: **React Native**, **React.js**, **Django REST Framework** y **Supabase**.
   * Actualizaciones mayores o cambios en la compatibilidad podrían requerir modificaciones significativas en el desarrollo.
3. **Sistemas operativos y navegadores:**
   * El correcto funcionamiento de la aplicación móvil depende de la compatibilidad con **Android 9+** y **iOS 14+**.
   * La plataforma web debe ser accesible desde navegadores modernos y actualizados (**Chrome, Firefox, Edge, Safari**).
4. **Seguridad y comunicaciones cifradas:**
   * La aplicación depende del uso de **protocolos HTTPS** y **TLS 1.3** para garantizar la confidencialidad e integridad de los datos en tránsito

## 2.6. Requisitos Futuros

El sistema **VitaLink** ha sido diseñado con una arquitectura modular y escalable, lo que permitirá incorporar nuevas funcionalidades y mejoras en futuras versiones sin comprometer la estabilidad del sistema actual. A continuación, se presentan posibles requisitos y líneas de evolución a considerar:

#### 1. Integración con dispositivos IoT de salud

* Soporte para la conexión con **wearables** y equipos biomédicos (pulseras, oxímetros, monitores de ritmo cardíaco, etc.).
* Sincronización automática de métricas de salud, como **frecuencia cardíaca**, **niveles de oxígeno** y **patrones de sueño**, para complementar el seguimiento clínico de pacientes con crisis de pánico.

#### 2. Conexión con redes de farmacias

* Integración con **APIs de farmacias** para consultar la disponibilidad de medicamentos recetados.
* Implementación de una **trazabilidad automatizada** de recetas médicas, permitiendo al paciente localizar farmacias cercanas con stock disponible.
* Posibilidad de generar **recordatorios automáticos** para la compra y reposición de medicamentos.

#### 3. Sistema avanzado de analítica predictiva

* Desarrollo de módulos basados en **inteligencia artificial** y **machine learning** para predecir posibles episodios de crisis de pánico en función de patrones históricos y métricas biomédicas.
* Generación de **alertas preventivas** para pacientes y médicos, facilitando intervenciones tempranas.

#### 4. Telemedicina y asistencia virtual

* Inclusión de un módulo de **videollamadas seguras** para consultas médicas a distancia.
* Implementación de un **chatbot clínico** con IA para ofrecer orientación inicial y derivar casos complejos a los especialistas correspondientes.

#### 5. Integración con sistemas nacionales de salud

* Posible conexión con plataformas gubernamentales de **historial clínico electrónico**, permitiendo el acceso centralizado a la información médica del paciente.
* Cumplimiento de estándares como **FHIR** y **HL7** para garantizar la interoperabilidad con otros sistemas hospitalarios.

#### 6. Gamificación y adherencia terapéutica

* Creación de un módulo de **motivación interactiva** para pacientes, incluyendo logros, recordatorios dinámicos y seguimiento de objetivos de tratamiento.
* Implementación de **notificaciones inteligentes** que fomenten la continuidad de la terapia y reduzcan el abandono del tratamiento.

# 3. Requisitos Específicos

• 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.

## 3.1 Requisitos comunes de las interfaces

### 3.1.1 Interfaces de usuario

* Las pantallas son de color café moro
* Letra tamaño 10
* Los botones deben ser en negrita
* La interfaz debe seguir principios de diseño UX/UI.
* Colores corporativos: Fondo negro y letras en blanco / Botones en amarillo
* Tipografía: Arial o sans-serif, tamaño 12px para textos principales, 14px para títulos.
* Botones claros, visibles y con feedback táctil/visual.
* Diseño responsive para adaptarse a diferentes tamaños de pantalla.
* Navegación simple con menús accesibles.

### 3.1.2 Interfaces de hardware

Especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

* App móvil: Android 8 o superior o iOS 12 o superior
* Pagina web:
  + Conexión Ethernet, WI-FI
  + 4GB Ram
  + Navegador Chrome 80+, Firefox 75+, Safari 13+

### 3.1.3 Interfaces de software

Indicar si hay que integrar el producto con otros productos de software.

* Para cada producto de software debe especificarse lo siguiente:
* Descripción del producto software utilizado
* Propósito del interfaz
* Definición del interfaz: contiendo y formato

| **Sistema utilizado** | **Descripción** | **Propósito** | **Interfaz (Contenido y formato)** |
| --- | --- | --- | --- |
| Supabase | Plataforma de código abierto que proporciona servicios de backend, incluyendo una base de datos PostgreSQL en tiempo real, autenticación, APIs REST. | Servir como la capa principal de persistencia de datos, gestionar la autenticación y autorización de usuarios, y proporcionar capacidades de sincronización en tiempo real. | **Contenido**: Datos de usuarios, registros de crisis, métricas de salud, historiales clínicos.  **Formato:** Comunicación mediante APIs RESTful y SDK de Supabase, Datos en JSON y Conexiones cifradas |
| Sistema de Gestión del Consultorio (Existente) | Sistema legacy interno del consultorio utilizado para la gestión administrativa, agenda de citas y historial clínico electrónico (HCE) básico. | Sincronizar información básica de pacientes (nombre, RUT, contacto) y datos de citas para evitar duplicidad y mantener consistencia. | **Contenido:** Datos de pacientes y agendas.  **Formato:** Integración mediante archivos planos (CSV/JSON) vía API REST. Comunicación segura mediante HTTPS. |
| Servicio de Notificaciones Push | Servicio externo para el envío eficiente y confiable de notificaciones push a dispositivos móviles (Android/iOS). | Permitir el envío de notificaciones en tiempo real a pacientes (recordatorios) y médicos (alertas de nuevas crisis). | **Contenido:** Tokens de dispositivos y mensajes de notificación.  **Formato:** Integración mediante la API HTTP.  Mensajes en formato JSON, siguiendo la estructura definida. |

## 3.2 Requisitos funcionales

Los requisitos funcionales deben estar enumerados (diagrama caso uso) y pueden ser divididos en sub-secciones.

| RF-001 | Registrar crisis de pánico |
| --- | --- |
| RF-002 | Consultar métricas de salud |
| RF-003 | Visualizar historial del paciente |
| RF-004 | Autenticación y sesiones |
| RF-005 | Gestión de pacientes |
| RF-006 | Gestión de tratamientos |
| RF-007 | Notificaciones al paciente |
| RF-008 | Exportar informes |
| RF-009 | Roles y permisos |
| RF-010 | Auditoría y trazabilidad |
| RF-011 | Tablero de métricas para médico |
| RF-012 | Contacto de emergencia |
| RF-013 | Cerrar Sesión |
| RF-014 | Iniciar Sesión |
| RF-015 | Notificaciones PUSH |
| RF-016 | Funcionalidad offline |

## 

## 

## 

## 

## 3.3 Requisitos no funcionales (Organización) y de calidad (Producto)

| RNF-001 | Seguridad en tránsito y acceso |
| --- | --- |
| RNF-002 | Privacidad de datos |
| RNF-003 | Usabilidad |
| RNF-004 | Rendimiento |
| RNF-005 | Disponibilidad y respaldo |
| RNF-006 | Observabilidad |
| RNF-007 | Interoperabilidad |
| RNF-008 | Sistema de solicitudes |
| RNF-009 | Compatibilidad |
| RNF-010 | Accesibilidad |
| RNF-011 | Capacidad de respuesta |
| RNF-012 | Eficiencia energetica |

### 3.3.1 Requisitos de Rendimiento o Eficiencia

Concurrencia esperada: 100 pacientes, 10 médicos, 3 administradores simultáneos.  
Throughput sostenido: ≥ 50 req/s promedio; picos de 150 req/s por 1 min.  
Latencia lectura métricas: p95 ≤ 1,5 s; p99 ≤ 2,5 s.  
Latencia escritura evento: p95 ≤ 2,0 s; p99 ≤ 3,0 s.  
Tamaño de dato por evento: ≤ 4 KB promedio.  
Volumen mensual: hasta 10.000 eventos.  
Límite de tasa por usuario: 30 req/min con backoff exponencial.  
Aceptación: pruebas de carga confirman los umbrales en 95% de escenarios.

### 3.3.2 Requisitos de Seguridad

Autenticación con Supabase Auth (JWT). Expiración de sesión: 60 min. Refresh tokens habilitados.  
Autorización por roles: Paciente, Médico, Admin. RLS activa en Postgres.  
Cifrado en tránsito: TLS 1.2+; en reposo: cifrado gestionado por proveedor.  
Gestión de secretos vía variables de entorno. Rotación semestral.  
Políticas de contraseñas: ≥ 12 caracteres, complejidad y bloqueo tras 5 intentos.  
Auditoría: registro inmutable de acceso y cambios a datos sensibles por 5 años.  
Backups diarios. RPO ≤ 24 h. RTO ≤ 4 h.  
Cumplimiento: protección de datos personales conforme normativa local vigente.  
Aceptación: pentest sin hallazgos críticos y checklist OWASP ASVS nivel 1 superado.

### 3.3.3 Requisitos de Usabilidad

Tiempo para registrar una crisis: ≤ 30 s por usuario nuevo; ≤ 15 s usuario recurrente.  
Tasa de finalización de tareas clave (registrar, consultar métricas, ver historial): ≥ 95%.  
Error de entrada por formulario: ≤ 2% por sesión; mensajes de validación claros.  
Accesibilidad: WCAG 2.1 AA en web. Tamaño táctil ≥ 44 px en móvil.  
Curva de aprendizaje: onboarding de 3 pantallas. Primer uso efectivo ≤ 5 min.  
Satisfacción SUS: ≥ 75 en prueba con ≥ 10 usuarios.  
Aceptación: pruebas de usabilidad cumplen metas y hallazgos críticos resueltos.

### 3.3.4 Requisitos de Disponibilidad

Disponibilidad mensual del servicio: ≥ 99,5%.  
Ventanas de mantención planificada: ≤ 2 h/mes en horario valle.  
Monitoreo y alertas p95 latencia, errores 5xx, y consumo de cuotas.  
Failover lógico a réplicas del proveedor cuando aplique.  
Aceptación: reporte mensual de SLO ≥ objetivo.

### 3.3.5 Requisitos de Portabilidad

Web: navegadores evergreen (últimas 2 versiones de Chrome, Edge, Firefox, Safari).  
Móvil: Android ≥ 10; iOS ≥ 15.  
Backend: SQL estándar y funciones portables; evitar extensiones propietarias.  
Empaquetado: contenedores reproducibles para módulos web cuando aplique.  
Aceptación: smoke tests pasan en entornos soportados.

### 3.3.6 Requisitos de Mantenibilidad

Arquitectura por capas y módulos desacoplados.  
Estilo y linting automáticos. Complejidad ciclomática media < 10.  
Cobertura de pruebas: ≥ 70% backend crítico; ≥ 60% frontend crítico.  
CI/CD con build, test y despliegue automatizados a entornos Dev/QA/Prod.  
Documentación: README por módulo, diagramas actualizados, runbooks de operación.  
Tiempos: corrección de bug crítico ≤ 24 h; alta ≤ 72 h.  
Aceptación: checklist de mantenibilidad y métricas de calidad superadas.

### 3.3.7 Requisitos de Funcionalidad

Adecuación: todos los CU críticos del MVP implementados y trazados a RF.  
Exactitud de métricas: error ≤ 1% respecto a consultas SQL de referencia.  
Interoperabilidad: API REST con JSON; exportación CSV/PDF consistente.  
Consistencia entre canales: el evento registrado en móvil es visible en web en ≤ 5 s.  
Conformidad: mensajes y validaciones alineados a reglas de negocio documentadas.  
Aceptación: pruebas funcionales pasan 100% en CU críticos y 95% en no críticos.

## 3.4 Requisitos No funcionales Organizacionales

**Identidad corporativa**

* Incluir logo oficial en login, header y PDFs.
* Tipografías corporativas: **Poppins** y **Open Sans**.
* Favicon y nombre de producto

**Gobernanza y procesos**

* Repositorio en **GitLab/GitHub Enterprise** de la empresa. Prohibido repos personales.
* Flujo de ramas **GitFlow** con convenciones: feature/, release/, hotfix/.

**Documentación y comunicación**

* Entregables en **DOCX** y **PDF** con plantilla corporativa.
* Idioma: español neutro e Inglés. Formato de fecha **dd-mm-aaaa**. Zona horaria **America/Santiago**.

**Entornos y nomenclatura**

* Entornos **DEV / QA / PROD**. Convenciones de nombres estandarizadas.
* Prefijo de BD y esquemas: corp\_ (ej.: corp\_eventos, corp\_pacientes).
* Variables de entorno gestionadas por **vault** corporativo.

**Soporte y capacitación**

* Horario de soporte: **lun–vie 09:00–18:00 CLT**.
* Dos sesiones de **90 min** de capacitación inicial a usuarios clave.
* Material de onboarding con branding corporativo.

**Hitos y reportes**

* Comité de seguimiento el **día 25 de cada mes**.
* Informe ejecutivo mensual de avance en formato corporativo.

# 4. Propuesta de Planificación

## 4.1 Descripción general acerca de la Planificación

El desarrollo del proyecto **VitaLink** se abordará mediante una metodología ágil, organizando el trabajo en Sprints de dos semanas de duración. El tiempo total estimado para la finalización del proyecto es de 5 meses (20 de agosto - 20 de diciembre de 2025), lo que equivale a aproximadamente 10 Sprints de desarrollo activo.

El equipo de trabajo está compuesto por tres integrantes con roles multifuncionales, lo que permite una distribución flexible de las tareas y una mayor resiliencia ante imprevistos. La ejecución estará a cargo de:

**Diego Kent Lahana:** Jefe de Proyecto. Desarrollador Full-Stack, Encargado del desarrollo de la aplicación móvil (React Native), toma de decisiones técnicas críticas.

**Martin Salinas y Juan Inostroza:** Analista Programador. documentación técnica de requisitos y testing de funcionalidades.

**Bastián Nahuel:** Analista Programador. Responsable del desarrollo del módulo web (Django), coordinación de la integración con Supabase y soporte en el diseño de interfaces.

Para garantizar el buen término del proyecto, se implementarán las siguientes buenas prácticas y condiciones:

* **Control de versiones:** Uso de Git con un repositorio centralizado (GitHub/GitLab), empleando un modelo de ramas (Git Flow) que facilite el desarrollo paralelo de features y la integración continua.
* **Gestión de proyecto:** Seguimiento de tareas mediante una herramienta de gestión (Trello, Jira o Azure DevOps), manteniendo un Product Backlog priorizado y un Sprint Backlog visible para todo el equipo.
* **Entornos de desarrollo:** Se mantendrán tres entornos claramente diferenciados:
  + **Desarrollo:** Para integración continua y testing inicial de features.
  + **Producción:** Entorno final para los usuarios.
* **Garantía de calidad:** Implementación de pruebas unitarias y de integración. Se dedicará la última semana de cada Sprint exclusivamente a testing y estabilización, asegurando que el incremento de software esté listo para desplegarse.

### 4.1.2 Definición del Equipo de Trabajo

| **Rol** | **Funciones clave** | **Dedicación estimada** | **Responsable** |
| --- | --- | --- | --- |
| Sponsor / Product Owner | Prioriza alcance, valida entregables | 10% | Cliente |
| Líder de Proyecto / Desarrollador Full-Stack | Plan, riesgos, arquitectura, trazabilidad | 30% | Diego Kent |
| Dev Backend/BaaS (Supabase) | Esquema, RLS, endpoints, métricas | 40% | Diego Kent |
| Dev Frontend Móvil | App paciente, autenticación, registro | 40% | Diego Kent |
| Dev Frontend Web | Desarrollo del módulo web (Django), testing de funcionalidades, coordinación de la integración con Supabase y soporte en diseño de interfaces. | 40% | Bastian Nahuel |
| QA | Plan de pruebas, casos, evidencias | 30% | Martin Salinas / Juan Inostroza |
| UX/UI | Wireframes, UI kit, accesibilidad | 20% | Bastian Nahuel |
| DevOps | CI/CD, monitoreo, despliegues | 20% | Juan Inostroza |

### 4.1.3 Definición de Actividades principales del Proyecto

Alineadas a EDT y carta Gantt.

1. Inicio y alcance: kickoff, riesgos, plan (EDT 1.0).
2. Arquitectura y diseño: modelo de datos, CU y flujos (EDT 2.0–2.2).
3. Backend/BaaS: auth, esquema, servicios y métricas (EDT 3.0–3.3).
4. App móvil paciente: registro, historial, sesión (EDT 4.0–4.3).
5. Web médico/admin: pacientes, tratamientos, permisos (EDT 5.0–5.3).
6. Analítica y tablero: KPIs, horas/lugares críticos (EDT 6.0–6.2).
7. Seguridad y cumplimiento: RLS, auditoría (EDT 7.0–7.2).
8. Pruebas y QA: funcionales, rendimiento, UAT (EDT 8.0–8.2).
9. Despliegue y operación: ambientes, monitoreo (EDT 9.0–9.2).
10. Documentación y cierre: manuales y traspaso (EDT 10.0–10.2).

### 4.1.4 Resumen Costos del Desarrollo del Proyecto

* **Tarifas de referencia** PM: **CLP 17.000/h** · Dev: **CLP 15.000/h** · QA: **CLP 12.000/h** · UX: **CLP 14.000/h** · DevOps: **CLP 18.000/h**.
* **Costos por FASE** (estimación basada en 726 HH totales)

| COSTO POR FASE | |
| --- | --- |
| Fase de Planificación | $ 363.000 |
| Fase de Análisis y Diseño | $ 980.000 |
| Fase de Desarrollo | $ 5.694.000 |
| Fase de QA | $ 385.000 |
| Fase de Implementación y Cierre | $ 452.000 |
| **TOTAL HH FASES** | **$ 7.874.000** |

* **Costos por Actor o Rol**

| COSTO HH POR ROL | |
| --- | --- |
| Jefe de Proyecto | $ 1.360.000 |
| Analista Programador | $ 4.050.000 |
| Administrador BD | $ 828.000 |
| Calidad Y Testing | $ 768.000 |
| Diseñador | $ 868.000 |
| **TOTAL HH** | **$ 7.874.000** |

### 

### 4.1.6 Carta Gantt

**07 agosto, 2025:** Inicio del proyecto junto a la creación del *acta de constitución* del mismo.

**15 agosto, 2025:** Revisión y aprobación del *acta de constitución* del proyecto.

**16 agosto, 2025:** Organización del equipo de trabajo, definición de roles y definición de requisitos generales.

**21 agosto, 2025:** Captura de requerimientos específicos y documentación de arquitectura servicio web.

**27 agosto, 2025:** Prototipos y propuesta ERS.

**01 septiembre, 2025:** Inicio ambiente de desarrollo.

**02 septiembre, 2025:** Inicio creación BBDD.

**08 septiembre, 2025:** Creación de procesos almacenados y consultas PL/SQL.

## 5. Anexos

### 5.1 Acta de Proyecto

[Acta de constitución.docx](https://docs.google.com/document/d/1itX2gx2wADy6xh84RnWfk9mpFpjkXGqv/edit?usp=drive_link&ouid=104380150534883371392&rtpof=true&sd=true)

### 5.2 Matriz Especificación de Requerimientos

[Planilla de Requerimientos.xlsx](https://docs.google.com/spreadsheets/d/1-6Dg1Yw8sv5qJCxhObYT3X4hKZkKr6L4/edit?usp=drive_link&ouid=104380150534883371392&rtpof=true&sd=true)

### 5.3 Prototipo de Software

[Documento Mockups.docx](https://docs.google.com/document/d/1rxE0Jo9qM7c-sAX9ssqmGYqCcTJoJZFo/edit?usp=drive_link&ouid=104380150534883371392&rtpof=true&sd=true)

### 5.4 Matriz EDT. Planilla Detallada Cálculo de Esfuerzo

[Matriz EDT.xlsx](https://docs.google.com/spreadsheets/d/1iZWaJh5ZzMTJ2ngiREtRJx8HzKBuiSsY/edit?usp=drive_link&ouid=104380150534883371392&rtpof=true&sd=true)

### 5.5 Planilla Carta Gantt

[Carta Gantt.xlsx](https://docs.google.com/spreadsheets/d/1ppVG4Tta5KrPbOyirKPSmEea-o59Gnfg/edit?usp=drive_link&ouid=104380150534883371392&rtpof=true&sd=true)

