**Plan de Proyecto**

**“UN KINE AMIGO”**

**Asignatura:** Capstone – Proyecto APT

**Institución:** Duoc UC – Sede San Joaquín

**Carrera:** Ingeniería en Informática

**Sección:** 002D

**Integrantes:**Adonis Nuñez,Italo Carvajal,Juan Medina,Jose Vergara

**Tabla de contenido**

**Contenido**

[**Introducción 6**](#_heading=h.30j0zll)

[**Datos del documento 7**](#_heading=h.2et92p0)

[Histórico de Revisiones 7](#_heading=h.ol0hn9nk8qtl)

[Información del Proyecto 7](#_heading=h.1fob9te)

[Integrantes 7](#_heading=h.atq5hh4t2mx7)

[**Información del Proyecto 8**](#_heading=h.vx6bi89gy992)

[**Propósito del Plan de Proyecto 9**](#_heading=h.8ckdkiaiejjj)

[**Visión del Proyecto Scrum 10**](#_heading=h.n4244d6u4acw)

[**Alcance del Proyecto 11**](#_heading=h.dutnk7w7zonc)

[Chat en Tiempo Real 11](#_heading=h.ymwshegt69b8)

[Panel Profesional 11](#_heading=h.xezxrh5nu282)

[**Avances a futuro 12**](#_heading=h.13hebkk2pozl)

[**Metodología de Desarrollo 13**](#_heading=h.fh04gk1hfoeh)

[Estructura del Sprint 13](#_heading=h.k6ijrfks54tp)

[Ceremonias Scrum 13](#_heading=h.d0mtft4m3s7b)

[1. Sprint Planning 13](#_heading=h.rm6bdzs4wj4o)

[2. Daily Meetings 13](#_heading=h.b974oao0ogx5)

[3. Sprint Review 14](#_heading=h.6lkb03noam85)

[4. Sprint Retrospective 14](#_heading=h.ccarlxtoaaf6)

[Incrementos entregados por Sprint 14](#_heading=h.ubt0e42mjsfh)

[Herramientas Utilizadas 14](#_heading=h.wni9644j3iod)

[Trello – Gestión del Trabajo 14](#_heading=h.6s80vbabktr7)

[GitHub – Control de Versiones 15](#_heading=h.v9vcfe9x636m)

[Google Drive – Respaldo y Coordinación 15](#_heading=h.5zso38r6t02u)

[**Definición de Roles y Responsabilidades 16**](#_heading=h.edehrija7d96)

[Product Owner — Adonis Núñez 16](#_heading=h.5s3791twewzv)

[Scrum Master — Juan Medina 16](#_heading=h.p369urq4lhjp)

[Scrum Team — José Vergara e Italo Carvajal 16](#_heading=h.tk5wnlux7c9c)

[**Costos 17**](#_heading=h.hjip1hkaru0l)

[**Resumen de Riesgos 17**](#_heading=h.rvgziw23p8t9)

[**Épicas del Proyecto 18**](#_heading=h.du4s3lxzn5m9)

[**Priorización de Épicas 19**](#_heading=h.xv9zgffo8ktk)

[**Definición de Historias de Usuario 20**](#_heading=h.fjmpnhwmswvf)

[**Product Backlog del Proyecto 22**](#_heading=h.rmixv74ybat4)

[**Estimación de Puntos de Historia 23**](#_heading=h.lkoal0j0oirs)

[**Sprint Planning 24**](#_heading=h.786bzj9nbfbc)

[**Definición y estimación de tareas 28**](#_heading=h.g6lqbi5lryph)

[**Sprint Backlog 28**](#_heading=h.7cd9u6pmr9tw)

[**BurnDown 28**](#_heading=h.f66wpb1mzek9)

[**Retrospectiva de Cada Sprint 29**](#_heading=h.tsnpl9is78v2)

[**Grafico del avance del sprint 1 29**](#_heading=h.m138n2p1lr56)

[Pila del producto actualizada del primer sprint 30](#_heading=h.pvh6ikpopm1l)

[Retrospectiva del primer sprint 31](#_heading=h.xp24e6h43o0c)

[**Grafico del avance del sprint 2 32**](#_heading=h.51dtc36i8ei4)

[Pila del producto actualizada del primer sprint 32](#_heading=h.sw7dxs7s4n09)

[Retrospectiva del primer sprint 33](#_heading=h.qpiud2vnkldw)

[**Grafico del avance del sprint 3 34**](#_heading=h.x9i3ee9fovqu)

[Pila del producto actualizada del primer sprint 34](#_heading=h.yh0b0vyi4ze8)

[Retrospectiva del primer sprint 35](#_heading=h.kwoy556ucvib)

[**Grafico del avance del sprint 4 37**](#_heading=h.uqlpnvq4xnso)

[Pila del producto actualizada del primer sprint 37](#_heading=h.5r4gdqejxi4a)

[Retrospectiva del primer sprint 39](#_heading=h.c3xppexlb6yu)

[**Grafico del avance del sprint 5 41**](#_heading=h.gurun3hhbq31)

[Pila del producto actualizada del primer sprint 41](#_heading=h.hr766spg12gp)

[Retrospectiva del primer sprint 42](#_heading=h.qbvtlbqeu9iy)

[**Retrospectiva del Proyecto 44**](#_heading=h.i7nzs74p0inw)

[**Plan de Pruebas 44**](#_heading=h.kpvdzsiv1nng)

[**Control de Riesgos 45**](#_heading=h.c2o6gv6qabwt)

[**Desarrollo del Proyecto 46**](#_heading=h.ja41l2osnq1k)

[Arquitectura Interna del Proyecto (Flutter) 46](#_heading=h.oz3nfrxsfdmm)

[Integraciones Técnicas con Firebase 47](#_heading=h.mngik6dp2lh4)

[Firebase Authentication 47](#_heading=h.1t7ion9r5k8)

[Cloud Firestore (Base de Datos Principal) 47](#_heading=h.60u43qnt0vxn)

[Firebase Cloud Messaging (FCM) 48](#_heading=h.pemcq9h08dll)

[Cloud Functions v2 (Node.js) 48](#_heading=h.d26vzp8703sc)

[Firestore Security Rules 48](#_heading=h.uaabhgsjzpq6)

[Integración con Supabase Storage 49](#_heading=h.hwkpjhj9uskf)

[Implementación Técnica de las Funcionalidades del Sistema 49](#_heading=h.727w2hx4cv97)

[Autenticación y Gestión de Usuarios 49](#_heading=h.wu773f46pgv4)

[Módulo de Ejercicios 49](#_heading=h.1j414b2tj8qo)

[Agenda y Citas 49](#_heading=h.ct3ok6ckexnb)

[Chat en Tiempo Real 50](#_heading=h.vgwz7p9hhmzd)

[Progreso y Métricas 50](#_heading=h.dg7dc9b6tgm4)

[Panel Profesional / Administrativo 50](#_heading=h.q3xagh9v4t2)

[Procesos de Desarrollo y Construcción 50](#_heading=h.x08slb9ki87t)

[**Plan de Capacitación 51**](#_heading=h.ar0xseoa18nw)

[**Manuales de Usuario 51**](#_heading=h.2siodmofr8ea)

[**Arquitectura del Sistema 52**](#_heading=h.1mv728tj9r1o)

[**La arquitectura del sistema Un Kine Amigo se diseñó con un enfoque modular, escalable y serverless, utilizando tecnologías modernas de desarrollo móvil (Flutter) y backend administrado en la nube (Firebase y Supabase). Esta arquitectura permite entregar un producto seguro, eficiente y mantenible, evitando la necesidad de servidores tradicionales y reduciendo costos operativos. 52**](#_heading=h.v51o6jzgd9c7)

[Arquitectura General del Sistema 52](#_heading=h.x8ui83eph7w)

[La aplicación está compuesta por: 52](#_heading=h.pf1ny3od7z51)

[Esta arquitectura permite: 52](#_heading=h.q84koo3gmufq)

[**Arquitectura Lógica Componentes de Software 53**](#_heading=h.uulpxb9qiz6h)

[Módulo de Autenticación 53](#_heading=h.addrntr42ncv)

[Módulo de Gestión de Usuarios 53](#_heading=h.n7zr27yiin7d)

[Módulo de Ejercicios y Rutinas 53](#_heading=h.jgevr828jl1h)

[Módulo de Citas / Agenda 53](#_heading=h.calbsur950wq)

[Módulo de Chat en Tiempo Real 54](#_heading=h.ewpuswkkdk7o)

[Módulo de Progreso y Métricas 54](#_heading=h.7tfl0kqca6ot)

[Módulo Administrativo 54](#_heading=h.vx17wtql30w2)

[**Arquitectura Física 55**](#_heading=h.t6ub6ji7xh7b)

[Capa Cliente (Frontend Flutter) 55](#_heading=h.wazk0y85rcqm)

[Capa Backend Serverless 55](#_heading=h.xf5k47yjfwtl)

[Firebase Authentication 55](#_heading=h.uce3i1cnjcyk)

[Firestore Database 55](#_heading=h.84sz2p92sc5y)

[Cloud Functions 56](#_heading=h.6zxr3lfp14xm)

[Firebase Cloud Messaging (FCM) 56](#_heading=h.pw8ymo7diswk)

[Capa de Almacenamiento Externo (Supabase Storage) 56](#_heading=h.qufq97yqap2s)

[**Flujos Principales del Sistema 57**](#_heading=h.b9ddwr8w7bay)

[Flujo de Login 57](#_heading=h.p3y2wud7rzrh)

[Flujo de Chat 57](#_heading=h.jtyelfkzbgom)

[Flujo de Agenda 57](#_heading=h.mxxesyianwj)

[Flujo de Verificación de Kinesiólogo 57](#_heading=h.fx6jqul5kde5)

[**Seguridad del Sistema 58**](#_heading=h.p9ywnuxsp4ah)

[Reglas de Firestore 58](#_heading=h.xxl99mxa8eq5)

[Seguridad en Supabase 58](#_heading=h.m33gd9gy80i)

[Validación de Roles 58](#_heading=h.7nv2nuey1u7i)

[Cloud Functions 58](#_heading=h.txsx1qv9gdu8)

[**Beneficios de la Arquitectura 59**](#_heading=h.n9lq2w59acvk)

[Alta escalabilidad 59](#_heading=h.hgkgxgwk4ta)

[Bajo costo operativo 59](#_heading=h.b0wm2634kkc2)

[Alta disponibilidad y resiliencia 59](#_heading=h.qutf5vtki95g)

[Velocidad de desarrollo y mantenimiento 59](#_heading=h.ohanunm5af58)

[Actualizaciones rápidas y sincronización en tiempo real 59](#_heading=h.vnq9n4cy1cs8)

[Seguridad robusta y control de acceso avanzado 60](#_heading=h.2rghtdrmpwm1)

[**Modelo de Base de Datos UML 61**](#_heading=h.nockdd8xfw3j)

[**Puntos de mejora 62**](#_heading=h.3hha6t9xdww2)

[**Lecciones aprendidas 63**](#_heading=h.1jc7mafwzhpl)

[**Mockups 64**](#_heading=h.rqk55ndzu3ur)

[**Condiciones de aceptación para el cierre del proyecto. 65**](#_heading=h.yesi9ogdniy2)

[1. Cumplimiento Funcional y de Alcance 65](#_heading=h.7s6kbiwvcmj1)

[2. Estabilidad Técnica y Rendimiento 65](#_heading=h.qxdgp0pujo8a)

[3. Usabilidad (UI/UX) y Experiencia del Usuario 65](#_heading=h.6cgpz48fqmx8)

[**Conclusiones 66**](#_heading=h.k9vv4y3wjsdd)

# Introducción

El presente informe final documenta el desarrollo del proyecto **“Un Kine Amigo”**, una aplicación móvil orientada a apoyar procesos de rehabilitación kinesiológica, conectando pacientes con kinesiólogos mediante herramientas digitales de monitoreo, comunicación y seguimiento de tratamientos.

Este informe consolida todos los artefactos generados en la Fase 2 bajo metodología ágil Scrum, incluyendo:

* planificación,
* épicas e historias,
* product backlog completo,
* sprint backlog,
* burn down por sprint,
* retrospectivas,
* plan de capacitación,
* manual de usuario,
* evaluación global del proyecto,
* y secciones pendientes como el DAS, el plan de pruebas y el modelo UML final.

De esta forma, se presenta un documento único, coherente y ordenado que refleja el trabajo realizado durante el ciclo de desarrollo.

# Datos del documento

## Histórico de Revisiones

| Versión | Fecha | Descripción/cambio | autor |
| --- | --- | --- | --- |
| 1.0 | 10-10-2025 | Primera versión del informe | Adonis Núñez |
|  |  |  |  |
|  |  |  |  |

## Información del Proyecto

| Organización | Asignatura Capstone / Institución Educativa (Implícito |
| --- | --- |
| Sección | 002D |
| Proyecto (Nombre) | UN KINE AMIGO |
| Fecha de Inicio |  |
| Fecha de Término |  |
| Patrocinador principal | Proyecto de Título |
| Docente | Jorge Alberto Castro Silvestre |

## Integrantes

| Sección | Nombre | Correo |
| --- | --- | --- |
| **002D** | **Adonis Núñez** | [**ado.nunez@duocuc.cl**](mailto:ado.nunez@duocuc.cl) |
| **002D** | **Italo carvajal** | [**it.carvajal@duocuc.cl**](mailto:it.carvajal@duocuc.cl) |
| **002D** | **Jose Vergara** | [**ji.vergara@duocuc.cl**](mailto:ji.vergara@duocuc.cl) |
| **002D** | **Juan Medina** | [**jua.medinap@duocuc.cl**](mailto:jua.medinap@duocuc.cl) |

# Información del Proyecto

Este proyecto propone el desarrollo de una aplicación móvil enfocada en el monitoreo de sesión de kinesiología con respaldo y ayuda de un profesional o de manera individual, con información confiable obtenida de profesionales de la salud. La plataforma contará con un sistema que planifica sesiones de kinesiología, ejercicios rápidos (estiramiento, reposo, etc), un sistema de seguimiento de los avances en los planes, un sistema donde uno puede buscar un profesional de la salud, y una sección donde un profesional puede ofrecer sus servicios.

El diseño de la plataforma considerará atributos de funcionalidad, usabilidad y confiabilidad, garantizando así una herramienta efectiva y accesible.

La relevancia de este proyecto radica en la forma en que se aborda un problema de salud global a través de una solución tecnológica, accesible y escalable. En lugar de ser solo una aplicación de ejercicios, nuestra propuesta ofrece una herramienta integral que conecta a pacientes y profesionales, democratizando el acceso a la atención médica.

# Propósito del Plan de Proyecto

El propósito del Plan de Proyecto es establecer de manera formal la visión, alcance, estructura de trabajo y lineamientos operativos para el desarrollo de la aplicación **“Un Kine Amigo”**. Este documento define cómo el equipo organizará, gestionará y ejecutará cada fase del proyecto, garantizando coherencia técnica, claridad en los roles, alineación con el docente guía y un uso eficiente del tiempo disponible.

El plan proporciona una guía centralizada para:

* **La gestión del producto**, asegurando que la solución responda a las necesidades reales de kinesiólogos y pacientes en rehabilitación.
* **La priorización y refinamiento del Product Backlog**, permitiendo desarrollar primero las funcionalidades de mayor valor.
* **La coordinación interna del equipo**, definiendo responsabilidades, dependencias y formas de trabajo colaborativo.
* **La construcción del MVP en un plazo de 4 meses**, mediante entregas iterativas e incrementales bajo la metodología Scrum.

# Visión del Proyecto Scrum

Nuestro propósito es democratizar y optimizar la rehabilitación física y la prevención de lesiones, Buscamos ser el puente digital que resuelva la baja adherencia a las terapias y la comunicación limitada, entregando una plataforma eficiente tanto para pacientes como para kinesiólogos. Se creará una aplicación móvil que conectará a pacientes con profesionales de la kinesiología, proporcionando herramientas de monitoreo, planes de ejercicio, sistema de seguimiento, etc. Con esto buscamos proporcionar una mejora medible en la calidad de vida del paciente al reducir el abandono de la terapia y aumentar la seguridad de los ejercicios. Para los kinesiólogos, se permitirá un contacto en tiempo real con los pacientes y una gestión más fácil.

El grupo objetivo de nuestro proyecto serán pacientes de rehabilitación y los que busquen prevenir problemas físicos. También, kinesiólogos y terapeutas físicos que buscan promocionarse y ofrecer sus servicios de manera más fácil y en un espacio especializado.

Con este proyecto satisfacemos las necesidades de aquellas personas que no pueden pagar un kinesiólogo, o por problemas de traslado no pueden ir a un centro de rehabilitación. Uno de los mayores problemas que se tienen son los altos costos de las consultas de kinesiología, se estima que va entre los $14.000 hasta los $40.000 pesos chilenos, y se deben desarrollar como mínimo 6 sesiones para tener un trabajo adecuado. Esto causa la baja adherencia de las personas a realizar kinesiología, sea preventiva o por alguna lesión. Con este proyecto se tendría una mejor conexión paciente-profesional ya que se pueden realizar consultas con profesionales certificados. Al poder realizar tus rutinas en los momentos oportunos, y sin necesidad de trasladarse y reservar un tiempo para realizar esto, estamos aliviando el tiempo que las personas utilizan para su salud.

"Un Kine Amigo" es una aplicación de salud digital que funciona como asistente de rehabilitación. Sus características clave incluyen: 1. Rutinas de kinesiología con seguimiento de sesiones. 2. Módulo de agenda/servicios con profesionales aprobados. 3. Chat directo con profesionales certificados. 4. Historial de progreso del usuario.

Mejora medible en la calidad de vida del paciente al reducir el abandono de la terapia y aumentar la seguridad de los ejercicios. Para los kinesiólogos, permite un contacto en tiempo real y una gestión más fácil. Para el equipo, un posicionamiento como innovadores en eHealth y generación de ingresos por suscripciones *premium*.

# Alcance del Proyecto

El proyecto abarca el desarrollo de la aplicación móvil "Un Kine Amigo" para Android, la cual integra rutinas guiadas, monitoreo de progreso y agendamiento con profesionales. Su impacto radica en potenciar la adherencia a los tratamientos y mejorar la calidad de vida de los pacientes, asegurando la inclusión digital de los adultos mayores mediante una interfaz accesible y simplificada.

| **Alcances principales del sistema** | **Descripcion** |
| --- | --- |
| **Gestión de Usuarios** | * Registro y autenticación (correo y Google). * Perfiles para pacientes y kinesiólogos. |
| **Planificación de Rutinas y Ejercicios Guiados** | * Catálogo de ejercicios segmentado por tipo de lesión, parte del cuerpo y nivel de dificultad. * Visualización de ejercicios con video explicativo. |
| **Sistema de Citas** | * Reservas de horas con kinesiólogos. * Vista de disponibilidad. * Gestión básica de citas. |
| Chat en Tiempo Real | * Comunicación entre paciente y kinesiólogo. * Notificaciones internas y contador de mensajes no leídos. |
| **Progreso y Métricas** | * Registro de rutinas completadas. * Panel básico de progreso. |
| Panel Profesional | * Vista de pacientes. * Revisión de métricas y ejercicios realizados. |

# 

# Avances a futuro

El proyecto tiene proyección para seguir creciendo con nuevas funcionalidades y mejoras que aumenten su valor para pacientes y kinesiólogos. Entre los avances futuros se considera ampliar los módulos actuales, integrar análisis más avanzados del progreso del paciente, mejorar la personalización de los ejercicios, optimizar la experiencia de usuario y sumar herramientas administrativas adicionales. También se proyecta escalar la infraestructura y reforzar la seguridad para soportar un mayor número de usuarios.

| **Avances a futuro** | **Descripcion** |
| --- | --- |
| **Implementación de chatbot** | Se implementará un chatbot para poder responder consultas en tiempo real sin necesidad de acudir al soporte |
| **Subir ficha medica para analitica con IA** | Se podra subir una ficha médica para que la IA la pueda analizar y le pueda recomendar ejercicios |
| **Validación de kinesiólogo Automatizada con IA** | Se implementará la validación automática de certificados profesional con IA para Kinesiologos. |
| **Enfoque en Accesibilidad Total (Control por Voz)** | Se realizará control por voz para personas de la 3ra edad para que puedan manejar mejor la app |

# Metodología de Desarrollo

El proyecto *Un Kine Amigo* se desarrolló utilizando el marco de trabajo Scrum, adaptado a la realidad académica y disponibilidad del equipo. Se implementó una estructura iterativa e incremental que permitió entregar valor continuo en cada etapa del desarrollo.

## Estructura del Sprint

* **Duración total:** 5 sprints.
* **Duración por sprint:** 2 semanas.
* Cada sprint incluyó planificación, ejecución, revisión con el equipo y cierre formal mediante una retrospectiva.

## Ceremonias Scrum

### 1. Sprint Planning

* Se definieron los objetivos del sprint basados en el Product Backlog priorizado.
* Se seleccionaron Historias de Usuario según capacidad del equipo.
* Se desglosaron HU en tareas con estimaciones en horas.
* Se creó un Sprint Backlog claro y visible en Trello.

Link del trello usado: [Trello](https://trello.com/invite/b/68d33fdd57857e16aeff7028/ATTI7a652c71c2b8a5f7b5201fc382b93eb70CE06C28/tablero-un-kine-amigo)

### 

### 2. Daily Meetings

* Reuniones breves (5–10 minutos).
* Adaptadas a la disponibilidad del equipo por horarios académicos y laborales.
* Cada integrante respondió:  
  + ¿Qué hice ayer?
  + ¿Qué haré hoy?
  + ¿Hay algún bloqueo?
* Permitieron detectar problemas temprano y redistribuir tareas cuando fue necesario.

### 3. Sprint Review

* Se presentaron los incrementos funcionales desarrollados.
* Se revisó el avance respecto al objetivo del sprint.
* Cada módulo completado (ej: Auth, Chat, Planes, Citas, Métricas, Notificaciones) fue mostrado y validado.
* Se recibieron comentarios y se ajustó el Product Backlog según las mejoras identificadas.

### 4. Sprint Retrospective

* Evaluación interna del desempeño del equipo.
* Se identificaron puntos fuertes, oportunidades de mejora y acuerdos para el siguiente sprint.
* Se analizaron aspectos como comunicación, cumplimiento de tiempos y uso de herramientas.

## Incrementos entregados por Sprint

Cada sprint generó un **incremento funcional** listo para ser probado o integrado.  
 Ejemplos:

* **Sprint 1:** Módulo de Auth (Login, Registro, integración con Google/Facebook).
* **Sprint 2:** Chats en tiempo real y sincronización con Firebase.
* **Sprint 3:** Citas y disponibilidad del profesional.
* **Sprint 4:** Planes de ejercicios, métricas, dashboards de progreso.
* **Sprint 5:** Notificaciones push, pruebas finales, pulido UI/UX y entrega del producto final.

## Herramientas Utilizadas

### Trello – Gestión del Trabajo

* Tableros organizados por Sprints.
* Columnas: *Por hacer*, *En progreso*, *En revisión*, *Completado*.
* Registro de tareas, responsables, fechas y etiquetas.
* Adjuntos de evidencia (capturas, commits, prototipos).

### GitHub – Control de Versiones

* Uso de repositorio centralizado.
* Ramas para cada módulo o funcionalidad.
* Commits frecuentes con mensajes descriptivos.
* Pull Requests para revisar código y evitar errores.
* Releases al finalizar cada sprint.

### Google Drive – Respaldo y Coordinación

* Carpeta central del proyecto: documentación, evidencia y entregables.
* Backups de reportes, diagramas, imágenes y presentaciones.
* Canal para compartir archivos entre integrantes y docentes.

# Definición de Roles y Responsabilidades

## Product Owner — Adonis Núñez

Es el responsable de maximizar el valor del producto y representar la voz del usuario. Entre sus funciones destacan:

* Definir y comunicar la visión del producto.
* Priorizar el Product Backlog según valor, riesgo y dependencias.
* Asegurar que las historias de usuario cuenten con criterios de aceptación claros.
* Validar los incrementos desarrollados en cada Sprint.

## Scrum Master — Juan Medina

Encargado de facilitar el marco de trabajo Scrum y apoyar al equipo en la eliminación de obstáculos. Sus responsabilidades incluyen:

* Garantizar el uso correcto de prácticas y ceremonias Scrum.
* Facilitar reuniones de planificación, revisión, daily y retrospectiva.  
  Identificar y remover impedimentos del equipo.
* Proteger al equipo de interrupciones externas y asegurar un flujo de trabajo estable.

## Scrum Team — José Vergara e Italo Carvajal

Equipo multidisciplinario encargado de construir el incremento del producto en cada Sprint. Sus tareas principales son:

* Desarrollar las funcionalidades definidas en el Product Backlog.
* Diseñar interfaces y experiencias de usuario (UI/UX) coherentes y accesibles.
* Integrar y probar cada módulo de la aplicación.
* Elaborar documentación técnica y funcional según avance del proyecto.

# 

# Costos

La sección de costos consolida todas las estimaciones económicas asociadas al desarrollo del proyecto “Un Kine Amigo”. Incluye costos directos (horas de trabajo del equipo), costos indirectos (servicios, herramientas y gastos operativos) y gastos fijos asociados al desarrollo.

Actualmente, la estimación final se encuentra en revisión y pendiente de consolidación, ya que depende de la validación de horas, recursos efectivamente utilizados y ajustes finales del plan de trabajo.

**Excel de costo :** [**Costo**](https://docs.google.com/spreadsheets/d/107Y7xfvwSaZeeaUGYAfCwhtY3yjP8RBN/edit?usp=sharing&ouid=115770852576023431618&rtpof=true&sd=true)

# Resumen de Riesgos

Durante la planificación del proyecto **Un Kine Amigo**, se realizó un análisis detallado de los riesgos que podrían afectar el desarrollo, la calidad y los plazos del producto. Dado que el proyecto involucra tecnologías móviles, servicios en la nube, datos personales de usuarios y un equipo con experiencia limitada en algunas áreas, fue necesario identificar riesgos técnicos, operacionales, legales y de gestión que pudieran comprometer la continuidad del proyecto.

Entre los riesgos detectados se encuentran la dependencia de profesionales externos para validar contenido clínico, la posibilidad de filtración de datos sensibles, problemas derivados de la comunicación del equipo, subestimación de tiempos, fallas en plataformas externas como Firebase o Supabase, y desafíos propios del desarrollo móvil como compatibilidad entre dispositivos y sistemas operativos. También se identificaron oportunidades tecnológicas relacionadas con inteligencia artificial, que pueden potenciar el valor futuro de la aplicación.

Con base en estos riesgos se elaboró una **Matriz de Riesgos**, donde cada evento fue evaluado según su probabilidad, impacto, categoría, prioridad y estrategia de respuesta. Este análisis permitió establecer medidas preventivas y correctivas para reducir retrasos, evitar fallas críticas y asegurar un desarrollo más controlado y eficiente.

**Matriz de riesgo :** [**Matriz de Riesgo**](https://docs.google.com/spreadsheets/d/1zWh4NyuYUXZg07AQ3YtoS5QeOpVvY8vF/edit?usp=sharing&ouid=115770852576023431618&rtpof=true&sd=true)

# Épicas del Proyecto

Las épicas son los grandes bloques funcionales del proyecto. Agrupan varias historias de usuario relacionadas y permiten organizar el desarrollo en áreas claras, como autenticación, comunicación, ejercicios, monitoreo y seguridad.

| ID | Epica |
| --- | --- |
| EP-01 | Gestión de Usuarios: Registro y Autenticación |
| EP-02 | Búsqueda y Recomendación de Ejercicios |
| EP-03 | Comunicación Usuario–Kinesiólogo (Chat, Notificaciones, Seguimiento) |
| EP-04 | Contenido Educativo y Preventivo |
| EP-05 | Usabilidad y Experiencia de Usuario |
| EP-06 | Monitoreo y Progreso del Paciente |
| EP-07 | Panel Administrativo |
| EP-08 | Seguridad y Privacidad |
| EP-09 | Soporte Técnico |

# Priorización de Épicas

La priorización define qué épicas se desarrollan primero según su impacto, urgencia y dependencias. Se priorizaron las funciones esenciales para que la app funcionará (registro, ejercicios, chat), seguidas de módulos complementarios como métricas, panel administrativo y soporte.

| ID | Epica | Prioridad | Comentario breve |
| --- | --- | --- | --- |
| EP-01 | Gestión de Usuarios | 1(Muy Alta) | Base del sistema, acceso a la app |
| EP-02 | Búsqueda y Recomendación | 2(Alta) | Clave para el paciente lesionado |
| EP-03 | Comunicacion Usuario-Kine | 2(Alta) | Interacción y asesoría personalizada |
| EP-04 | Ux y Experiencia de Usuario | 2(Alta) | Mejora retención y uso |
| EP-05 | Seguridad y Privacidad | 1(Muy Alta) | Protección de datos sensibles |
| EP-06 | Contenido Eductativo | 3(Media) | Valor agregado |
| EP-07 | Monitoreo y Progreso | 3(Media) | Depende de módulos previos |
| EP-08 | Panel Administrativo | 3(Media) | Gestión interna |
| EP-09 | Soporte Tecnico | 4(Baja) | Complementaria, futura iteración |

# Definición de Historias de Usuario

Las historias de usuario describen necesidades específicas desde el punto de vista del Usuario que permiten planificar tareas claras, enfocadas en entregar valor real en cada sprint.

**Historias de usuario Completa :**[**Historias de Usario**](https://docs.google.com/spreadsheets/d/1wz-yi3kkpJE8lL-vAbPFzYMjgM2i2xQb/edit?usp=sharing&ouid=115770852576023431618&rtpof=true&sd=true)

| ID)de la historia | Rol | Característica / Funcionalidad | Razón | N\* de escenario | Criterio de aceptación (Título) | Contexto | Evento | Resultado / Comportamiento esperado |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| H-01 | Como usuario | quiero buscar ejercicios específicos para mi tipo de lesión, para seguir un plan de recuperación adecuado. | Seguir un plan adecuado | 1 | Ejercicios disponibles | Usuario tiene una lesión | Busca ejercicios | Se muestran ejercicios asociados |
| 2 | Sin ejercicios | Usuario busca ejercicios | No existen ejercicios | “No hay ejercicios disponibles” |
| 3 | Ordenamiento | Usuario accede al listado | Se muestran ejercicios | Ordenados alfabéticamente |
| H-02 | Como Kinesiologo | quiero comunicarme directamente con los usuarios a través de un chat, para resolver dudas y dar seguimiento personalizado. | Resolver dudas | 1 | Envío exitoso | Tiene pacientes | Envía mensaje | Llega en tiempo real |
| 2 | Error de envío | Intenta enviar mensaje | No hay conexión | Muestra error |
| 3 | Historial disponible | Conversación previa | Abre chat | Historial visible |
| H-03 | Como Paciente | quiero comunicarme directamente con los kinesiologos a través de un chat, para resolver dudas, obtener ayuda, entre otros. | Obtener ayuda | 1 | Mensaje enviado | Tiene kine asignado | Envía mensaje | Llega al kinesiólogo |
| 2 | Notificación | Kine responde | Mensaje enviado | Paciente recibe notificación |
| 3 | Historial cargado | Hay historia previa | Abre chat | Ve historial completo |
| H-04 | Como nuevo usuario | quiero registrarme usando mi correo, cuenta Google o Facebook, para iniciar sesión rápida y fácilmente. | Acceso rápido | 1 | Registro exitoso | Usuario nuevo | Completa formulario | Cuenta creada |
| 2 | Error al registrarse | Usuario registra | Datos inválidos | Muestra error |
| 3 | Inicio rápido | Usuario existente | Usa Google/Facebook | Entra sin contraseña |

# Product Backlog del Proyecto

| Identificador (ID) de la Historia | Enunciado de la Historia | Alias | Estado | Dimensión / Esfuerzo | Iteración (Sprint) | Prioridad | Comentarios |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Historia de Usuario** | **Funcionalidad** | **Estado** | **Estimación (puntos)** | **Sprint** | **Prioridad** | **Justificación** |
| HU-01 | Como usuario con una lesión, quiero buscar ejercicios específicos para mi tipo de lesión, para seguir un plan de recuperación adecuado. | Buscar ejercicios | Pendiente | 5 |  | Alta | Núcleo del producto; aporta el valor principal al usuario. |
| HU-02 | Como kinesiólogo, quiero comunicarme directamente con los usuarios a través de un chat, para resolver dudas y dar seguimiento personalizado. | Chat directo | Listo | 8 |  | Alta | Funcionalidad crítica de interacción; requiere backend en tiempo real y notificaciones. |
| HU-03 | Como nuevo usuario, quiero registrarme usando mi correo o cuenta Google, para iniciar sesión rápida y fácilmente. | Registro rápido | Listo | 3 |  | Alta | Es la primera funcionalidad esencial; permite acceso y prueba inicial del sistema. |
| HU-04 | Como usuario, quiero consultar un historial de los ejercicios que he realizado, para hacer seguimiento de mi progreso. | Historial ejercicios | Pendiente | 5 |  | Media | Permite visualizar avances y fomenta la adherencia a la terapia. |

El Product Backlog se estructuró con: ID, historia, funcionalidad, estado, estimación en puntos, prioridad y justificación.

**Producto Backlog Completo:** [**Producto Backlog**](https://docs.google.com/spreadsheets/d/1YOzsb2W20N6Q83UIwwgOdPpe6tRgEADY/edit?usp=sharing&ouid=115770852576023431618&rtpof=true&sd=true)

# Estimación de Puntos de Historia

**Técnica: Planning Poker**

Descripción: El equipo de desarrollo asignó los puntos de historia usando la escala de Fibonacci (1, 2, 3, 5, 8, 13, 21), basándose en la complejidad, esfuerzo y tiempo requerido para implementar cada funcionalidad.

**Link a plantilla con todos los puntos de historia:** [**Puntos de historia**](https://docs.google.com/spreadsheets/d/1MzPD5kb_7PxrkS5c3CeHRmWFtlviJQkr/edit?gid=6680916#gid=6680916)

| **ID** | **Historia de Usuario** | **Estimación(puntos)** |
| --- | --- | --- |
| HU-01 | Como usuario lesionado, quiero buscar ejercicios según mi tipo de lesión para realizar una rutina adecuada a mi recuperación. | 5 |
| HU-02 | Como kinesiólogo, quiero acceder a un chat directo con los usuarios para brindar asesoría personalizada. | 13 |
| HU-03 | Como nuevo usuario, quiero registrarme con mi correo o cuenta Google para acceder rápidamente a la aplicación | 3 |
| HU-04 | Como usuario, quiero ver un historial de mis ejercicios realizados para monitorear mi progreso. | 5 |
| HU-05 | Como usuario, quiero recibir recomendaciones automáticas de ejercicios según mi lesión. | 8 |
| HU-06 | Como usuario, quiero poder recuperar mi contraseña si la olvido. | 3 |
| HU-07 | Como usuario, quiero guardar mis ejercicios favoritos para acceder a ellos fácilmente. | 3 |
| HU-08 | Como usuario, quiero recibir notificaciones cuando mi kinesiólogo me envía un mensaje para no perder información importante. | 8 |
| HU-09 | Como usuario, quiero ver videos explicativos de cada ejercicio, con la finalidad de realizar correctamente las rutinas y evitar lesiones. | 8 |
| HU-10 | Como usuario, quiero recibir recordatorios de mis sesiones o rutinas programadas, con la finalidad de mantener la constancia en mi recuperación. | 5 |
| HU-11 | Como usuario, quiero calificar y dejar comentarios sobre el kinesiólogo que me atiende, con la finalidad de valorar su atención y ayudar a otros usuarios. | 5 |
| HU-12 | Como usuario, quiero filtrar ejercicios por nivel de dificultad Básico, intermedio, avanzado, con la finalidad de realizar rutinas acordes a mi condición física actual. | 3 |

# Sprint Planning

El sprint planning es un evento de planificación del sprint que ayuda al equipo a decidir qué trabajo van a realizar durante el sprint, también llamado objetivo del sprint. En la planificación, el equipo elige elementos del backlog del producto, la lista detallada de características y tareas necesarias para completar el objetivo del producto o del proyecto y los añade a su backlog del sprint.

**Link del trello utilizado para el sprint Planning:** [**Trello**](https://trello.com/invite/b/68d33fdd57857e16aeff7028/ATTI7a652c71c2b8a5f7b5201fc382b93eb70CE06C28/tablero-un-kine-amigo)

| **Id** | **Historia de usuario** | **sprint** |
| --- | --- | --- |
| H-01 | Como usuario con una lesión, quiero buscar ejercicios específicos para mi tipo de lesión, para seguir un plan de recuperación adecuado. | **1** |
| H-02 | Como kinesiólogo, quiero comunicarme directamente con los usuarios a través de un chat, para resolver dudas y dar seguimiento personalizado. | **2** |
| H-03 | Como paciente, quiero comunicarme directamente con los kinesiologos a través de un chat, para resolver dudas, obtener ayuda, entre otros. | **2** |
| H-04 | Como nuevo usuario, quiero registrarme usando mi correo, cuenta Google o Facebook, para iniciar sesión rápida y fácilmente. | **3** |
| H-05 | Como usuario, quiero consultar un historial de los ejercicios que he realizado, para hacer seguimiento de mi progreso. | **3** |
| H-06 | Como usuario, quiero recuperar mi contraseña en caso de olvidarla, para mantener acceso seguro a mi cuenta. | **3** |
| H-07 | Como usuario, quiero recibir notificaciones de nuevos mensajes de mi kinesiólogo, para no perder información importante. | **2** |
| H-08 | Como usuario, quiero acceder a videos de ejemplo de cada ejercicio, para realizarlos correctamente y evitar lesiones. | **4** |
| H-09 | Como usuario, quiero recibir recordatorios de mis sesiones y rutinas, para mantener constancia en mi recuperación. | **4** |
| H-10 | Como usuario, quiero filtrar los ejercicios según su nivel de dificultad, para adaptar la rutina a mi capacidad física. | **1** |
| H-11 | Como usuario, quiero recibir notificaciones cuando interactúa con la aplicación, esto para saber si se ejecuto correctamente o no | **4** |
| H-12 | Como usuario, quiero utilizar una aplicacion sencilla y bonita de usar, con la finalidad de que sea entendible que se hace en la aplicacion. | **1** |
| H-13 | Como paciente, quiero poder ver un historial de todos los ejercicios que he completado y cuándo lo hice, para medir mi progreso, mantener la motivación y tener un registro de mi cumplimiento. | **5** |
| H-14 | Como kinesiólogo, quiero poder certificarme en la aplicacion, con la finalidad de poder utilizar las funciones de los kinesiologos | **5** |
| H-15 | Como usuario, quiero poder ver todos los ejercicios en una pantalla, con la finalidad de poder encontrar facilmente los ejercicios. | **1** |
| H-16 | Como usuario registrado, quiero poder iniciar sesión usando mi correo y contraseña, para acceder a mi área privada de la aplicación. | **3** |
| H-17 | Como usuario que olvidó su contraseña, quiero un proceso claro de restablecimiento para poder recuperar el acceso a mi cuenta de manera rápida y segura. | **1** |
| H-18 | Como paciente, quiero buscar profesionales con sus descripciones, para encontrar la opción que mejor se adapte a mis necesidades. | **5** |
| H-19 | Como kinesiologo no verificado, quiero poder cargar un documento con el cual poder verificarme en la aplicación" | **5** |
| H-20 | Como paciente, quiero poder ver el calendario de mi profesional y agendar una sesión en el horario disponible de mi preferencia, para poder tener una cita medica con el. | **5** |
| H-21 | Como kinesiólogo, quiero poder configurar de forma rápida y flexible mi disponibilidad horaria para que los pacientes solo puedan agendar sesiones en los momentos que me sean convenientes. | **5** |
| H-22 | Como administrador, quiero poder ingresar a una página donde pueda ver los usuarios y sus datos. | **5** |
| H-23 | Como administrador, quiero tener un panel de solicitudes para revisar, aprobar o rechazar los documentos de verificación de los kinesiólogos. | **5** |
| H-24 | Implementar el registro de rutinas completadas y el tiempo dedicado. | **5** |
| H-25 | Como kinesiólogo, quiero visualizar el progreso, métricas clave y la adherencia de mis pacientes, para evaluar la efectividad de la terapia. | **5** |
| H-26 | Como kinesiólogo, quiero poder suscribirme a un plan Premium para aumentar mi visibilidad, aparecer primero en las búsquedas y obtener más pacientes. | **5** |
| H-27 | Como usuario (paciente o kinesiólogo), quiero poder filtrar el catálogo de ejercicios por parte del cuerpo a trabajar. | **5** |

# Definición y estimación de tareas

# Sprint Backlog

El Sprint Backlog corresponde al conjunto de **historias de usuario y tareas seleccionadas** para desarrollarse durante cada sprint del proyecto *Un Kine Amigo*. Incluye funcionalidades clave como autenticación, chat, ejercicios, citas, planes y métricas, priorizadas según el valor que aportan a pacientes y kinesiólogos. En el siguiente Link (documento) se pueden ver las tareas y tiempo aproximado

**Sprint Backlog :** [**Sprint Backlog**](https://docs.google.com/spreadsheets/d/1zekNsCwYSs0m6BuwpxAUGrXvydyIxWMkUaZZEykySB4/edit?usp=sharing)

# BurnDown

El Burndown Chart es un gráfico que muestra el **avance diario del sprint**, comparando las horas o puntos planificados versus los que van quedando por completar. Permite visualizar si el equipo está avanzando al ritmo esperado y detectar atrasos o bloqueos a tiempo. En el proyecto *Un Kine Amigo*, este gráfico se utilizó para monitorear el progreso de cada sprint y asegurar el cumplimiento del objetivo dentro del plazo establecido.

**BurnDown completo:**[**BurnDown**](https://docs.google.com/spreadsheets/d/16a1w9iIWbvOsX_DtE1SNivr6i8m9YvcT/edit?usp=sharing&ouid=115770852576023431618&rtpof=true&sd=true)

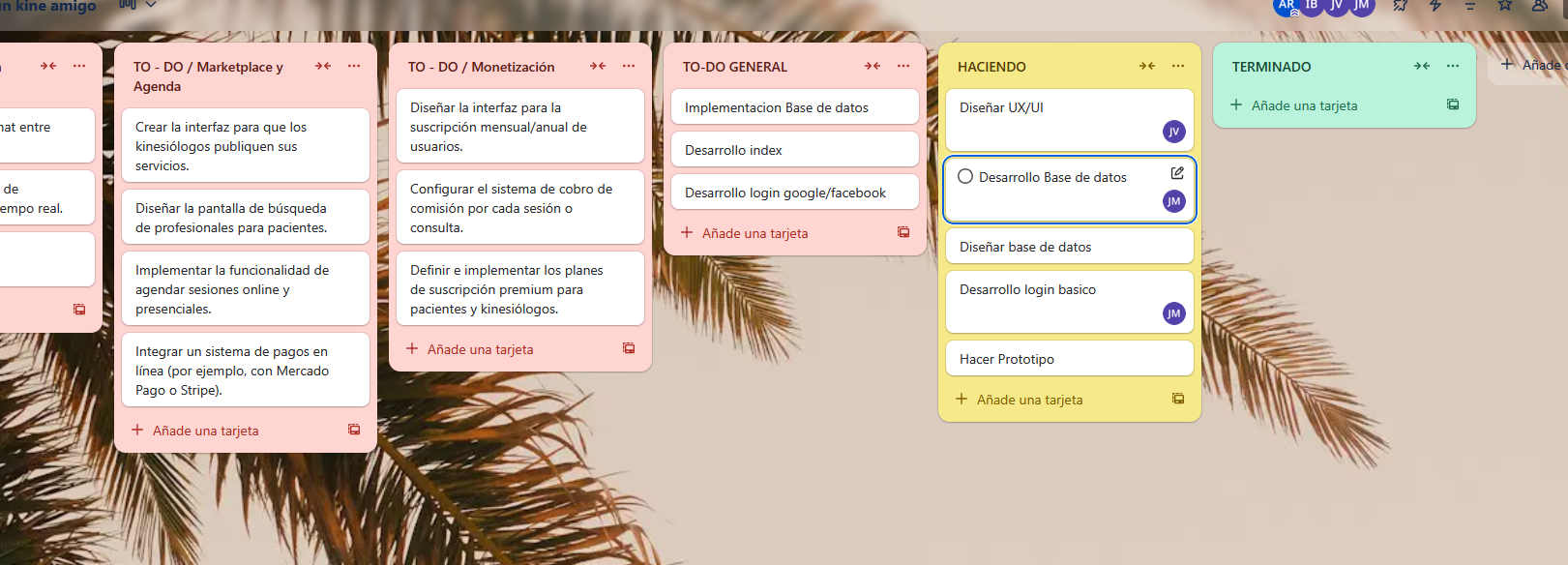
# Retrospectiva de Cada Sprint

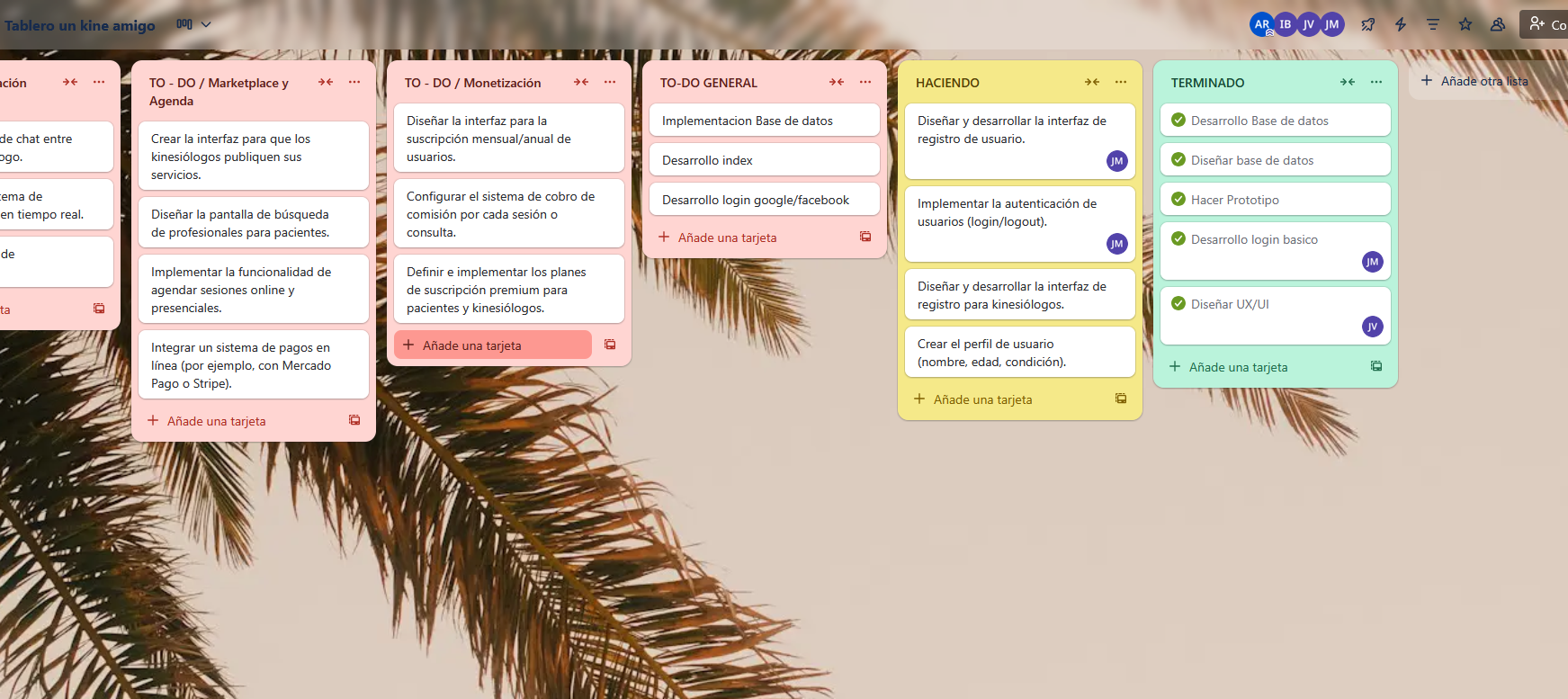
La Retrospectiva es la instancia donde el equipo analiza **cómo trabajó durante el sprint**, identificando aspectos positivos, problemas y oportunidades de mejora. En *Un Kine Amigo*, estas reuniones permitieron ajustar la comunicación, mejorar la organización de tareas, optimizar estimaciones y corregir bloqueos detectados. Gracias a estas retrospectivas, cada sprint se ejecutó con mayor eficiencia y el equipo pudo mejorar continuamente su forma de trabajo.

**Retrospectiva de los sprint completo :** [**Restrospectiva**](https://docs.google.com/document/d/1ENx9lbmziOO_h-VNufzEDt5Hf4ENY1v5os1w5nV9FiM/edit?usp=sharing)

# 

# Grafico del avance del sprint 1





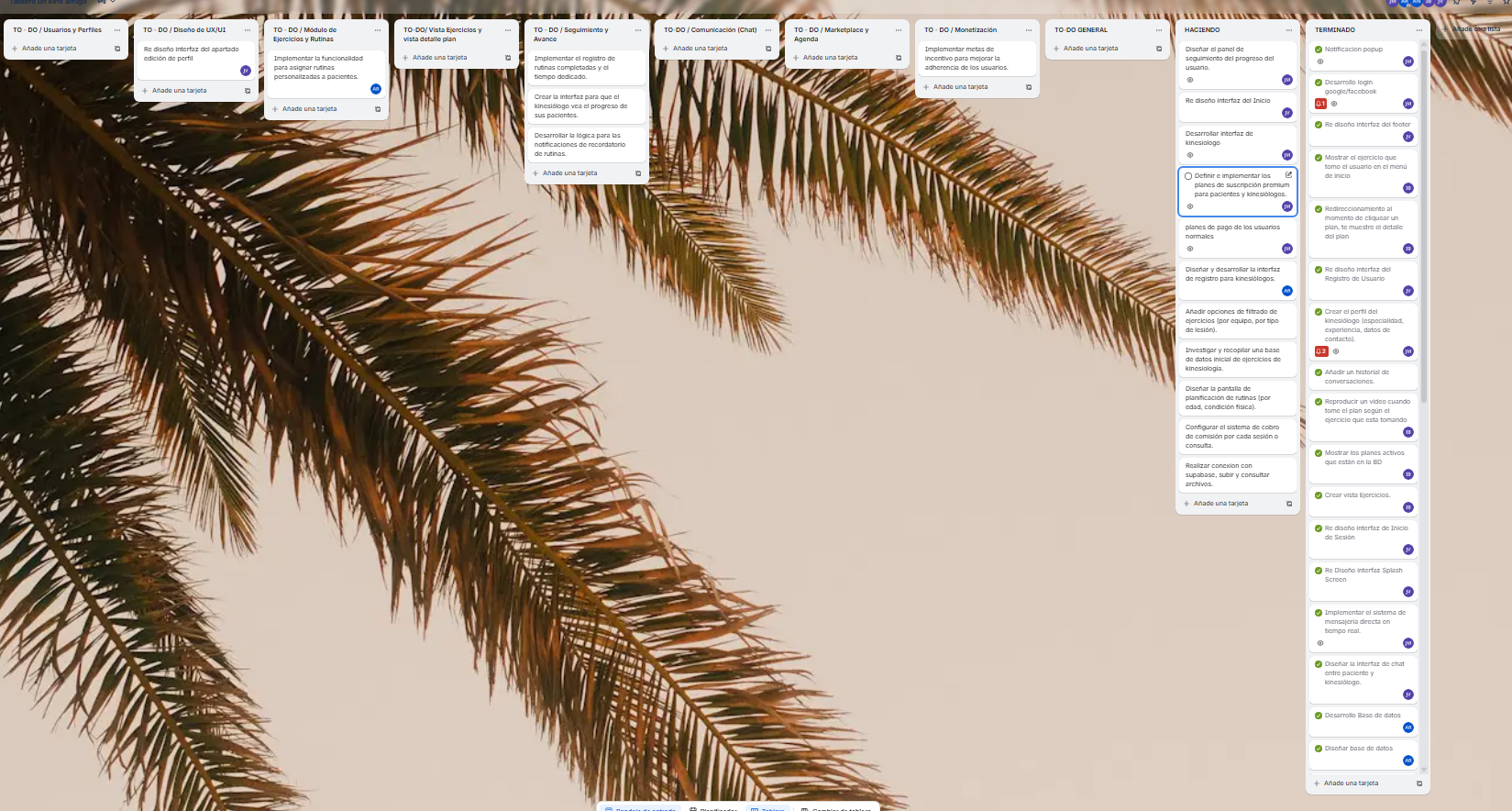
## Pila del producto actualizada del primer sprint

| Id | Historia de usuario | Estimacion | Prioridad |
| --- | --- | --- | --- |
| H-04 | Como nuevo usuario, quiero registrarme usando mi correo, cuenta Google o Facebook, para iniciar sesión rápida y fácilmente. | 3 | Alta |
| H-06 | Como usuario, quiero recuperar mi contraseña en caso de olvidarla, para mantener acceso seguro a mi cuenta. | 3 | Alta |
| H-16 | Como usuario registrado, quiero poder iniciar sesión usando mi correo y contraseña, para acceder a mi área privada de la aplicación. | 3 | Alta |
| H-17 | Como usuario que olvidó su contraseña, quiero un proceso claro de restablecimiento para poder recuperar el acceso a mi cuenta de manera rápida y segura. | 5 | Alta |

## Retrospectiva del primer sprint

| ¿Qué salió bien en la iteración? (aciertos) | * Logramos completar las historias principales de autenticación del sprint: registro de usuarios, inicio de sesión y flujo básico de recuperación/restablecimiento de contraseña. * Pudimos definir y aplicar una arquitectura clara para esta parte del sistema (frontend, API y base de datos), lo que nos dejó una base sólida para las siguientes iteraciones. * El equipo se organizó bien: repartimos las tareas por componentes (pantallas, API, BD) y mantuvimos una comunicación constante para integrar los avances. |
| --- | --- |
| ¿Qué no salió bien en la iteración? (errores) | * Subestimamos el tiempo necesario para integrar correctamente el login con proveedores externos (Google/Facebook), lo que generó más trabajo técnico del que habíamos considerado al principio. * Dedicamos más tiempo del esperado a configuración inicial del entorno (conexión a la base de datos, variables de entorno, manejo de hashes de contraseña), lo que retrasó el desarrollo de algunas tareas. * No alcanzamos el nivel de pruebas y casos de error que teníamos planificado (por ejemplo, mensajes más detallados para errores específicos en el login o recuperación de cuenta). |
| ¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua) | * Refinar la estimación de esfuerzo, partiendo las historias más grandes en tareas más pequeñas y manejables, para tener un seguimiento más claro del avance. * Definir mejor la “Definition of Done” incluyendo siempre pruebas básicas (validaciones, mensajes de error) y una pequeña revisión entre pares antes de dar una HU por terminada. |

# Grafico del avance del sprint 2



## Pila del producto actualizada del primer sprint

| Id | Historia de usuario | Estimacion | Prioridad |
| --- | --- | --- | --- |
| H-11 | Como usuario, quiero recibir notificaciones cuando interactuo con la aplicacion, esto para saber si se ejecuto correctamente o no | 3 | Media |
| H-12 | Como usuario, quiero utilizar una aplicacion sencilla y bonita de usar, con la finalidad de que sea entendible que se hace en la aplicacion. | 3 | Media |

## Retrospectiva del primer sprint

| ¿Qué salió bien en la iteración? (aciertos) | * Logramos implementar las notificaciones internas de la aplicación (pop-ups/mensajes de feedback) para informar al usuario cuando una acción se ejecuta correctamente o cuando ocurre un error. * Mejoramos la apariencia general de la interfaz, aplicando un diseño más limpio y consistente: colores, tipografías y componentes más ordenados y fáciles de entender. |
| --- | --- |
| ¿Qué no salió bien en la iteración? (errores) | * Al principio no definimos bien todos los casos donde debían aparecer notificaciones, y tuvimos que volver a algunas pantallas para agregar mensajes de éxito/error que faltaban. * Subestimamos el tiempo que toma probar el diseño en distintos flujos (por ejemplo, versiones de pantallas móviles, distintos estados vacíos, errores del servidor, etc.). * Hubo pequeños desacuerdos en el equipo sobre detalles visuales (colores, tamaños, textos), lo que generó iteraciones extra y algunos retrabajos. |
| ¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua) | * Antes de desarrollar, vamos a listar todos los escenarios donde se necesitan notificaciones, para evitar tener que “parchar” después las pantallas. * Definiremos un pequeño “manual de diseño interno” (colores, estilos de botones, textos estándar de error/éxito) para que todo el equipo use los mismos criterios y se reduzcan los retrabajos. * Vamos a dedicar tiempo específico dentro del sprint a pruebas de usabilidad y revisión visual, no solo a que “la funcionalidad funcione”. |

# Grafico del avance del sprint 3

## Pila del producto actualizada del primer sprint

| Id | Historia de usuario | Estimacion | Prioridad |
| --- | --- | --- | --- |
| H-01 | Como usuario con una lesión, quiero buscar ejercicios específicos para mi tipo de lesión, para seguir un plan de recuperación adecuado. | 5 | Alta |
| H-08 | Como usuario, quiero acceder a videos de ejemplo de cada ejercicio, para realizarlos correctamente y evitar lesiones. | 8 | Alta |
| H-10 | Como usuario, quiero filtrar los ejercicios según su nivel de dificultad, para adaptar la rutina a mi capacidad física. | 3 | Media |
| H-13 | Como paciente, quiero poder ver un historial de todos los ejercicios que he completado y cuándo lo hice, para medir mi progreso, mantener la motivación y tener un registro de mi cumplimiento. | 3 | Media |
| H-14 | Como kinesiólogo, quiero poder certificarme en la aplicacion, con la finalidad de poder utilizar las funciones de los kinesiologos | 5 | Alta |
| H-15 | Como usuario, quiero poder ver todos los ejercicios en una pantalla, con la finalidad de poder encontrar facilmente los ejercicios. | 5 | Alta |
| H-20 | Como paciente, quiero poder ver el calendario de mi profesional y agendar una sesión en el horario disponible de mi preferencia, para poder tener una cita medica con el. | 8 | Alta |

## Retrospectiva del primer sprint

| ¿Qué salió bien en la iteración? (aciertos) | * Logramos construir el núcleo funcional del módulo de ejercicios: búsqueda por tipo de lesión, catálogo de ejercicios en una sola pantalla y filtros por nivel de dificultad, lo que deja una base sólida para la experiencia del paciente. * Implementamos la reproducción de videos de ejemplo asociada a los ejercicios, lo que mejora mucho la guía al usuario y reduce el riesgo de que realice mal los movimientos. * Pudimos levantar un historial de ejercicios completados, permitiendo registrar qué hizo el paciente y cuándo, reforzando el seguimiento y la motivación. |
| --- | --- |
| ¿Qué no salió bien en la iteración? (errores) | * La cantidad de historias de usuario de este sprint fue muy alta y muy ambiciosa, lo que hizo que algunas funcionalidades quedaran justas de tiempo o con detalles por pulir (por ejemplo, estados visuales o mensajes de error). * La integración entre historial de ejercicios y calendario de sesiones no quedó tan alineada como queríamos al principio; hubo que ajustar varias veces la estructura de datos para que todo conversara bien. * Subestimamos la complejidad de combinar filtros (lesión, dificultad, catálogo completo) y a ratos el rendimiento o la claridad de la interfaz se vieron afectados. |
| ¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua) | * Ser más realistas al seleccionar el alcance del sprint, dividiendo las historias grandes (como agenda + ejercicios + historial) en incrementos más pequeños y entregables. * Diseñar y acordar modelos de datos y flujos complejos antes de programar (por ejemplo, relación entre ejercicios, historial, calendario y certificación), para reducir cambios de estructura a mitad de sprint. * Definir mejor la prioridad interna dentro del sprint: primero dejar completamente cerrado lo crítico (búsqueda, videos, agenda básica) y luego avanzar en mejoras (filtros combinados avanzados, vistas más detalladas del historial). |

# Grafico del avance del sprint 4

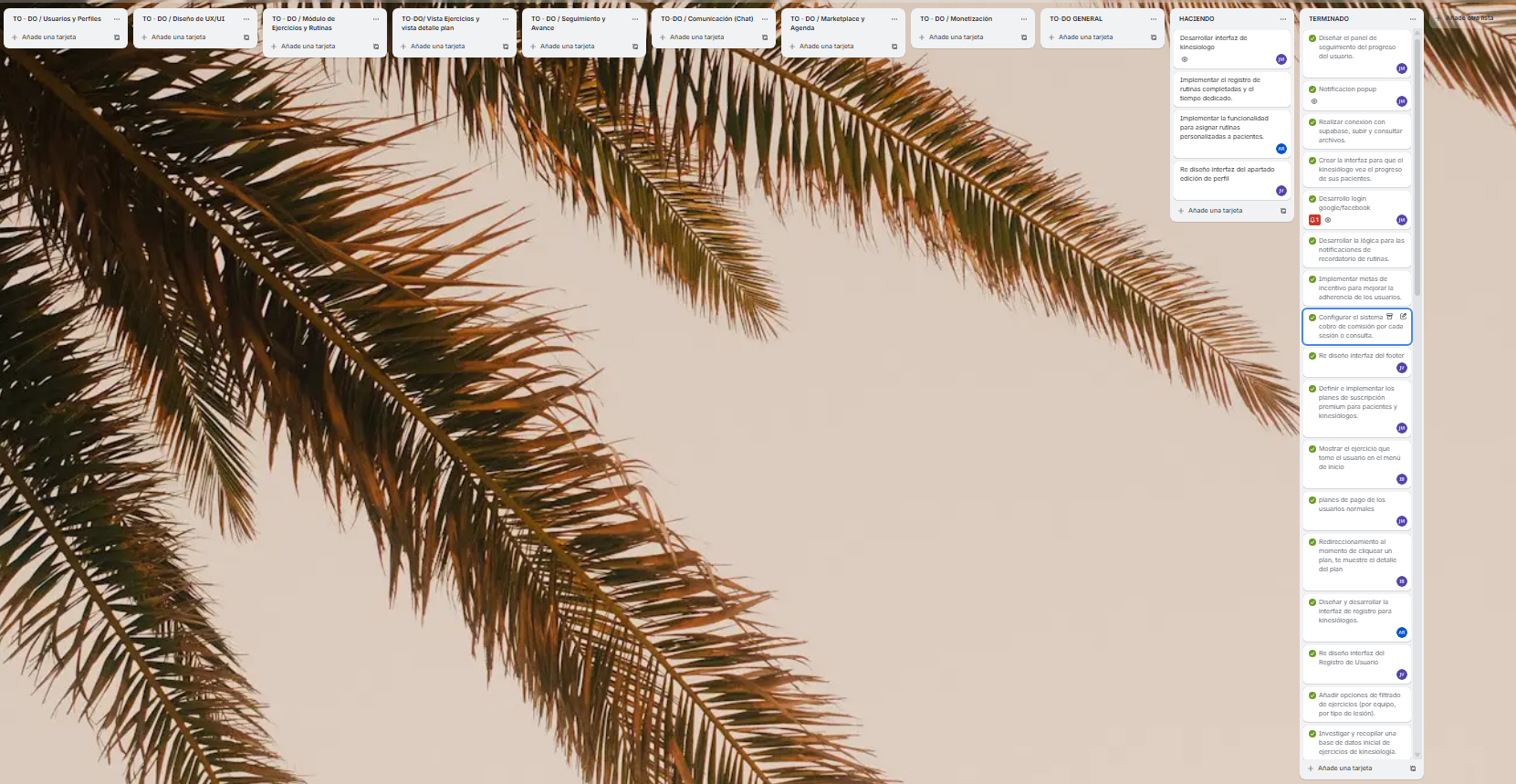
## Pila del producto actualizada del primer sprint

| Id | Historia de usuario | Estimacion | Prioridad |
| --- | --- | --- | --- |
| H-02 | Como kinesiólogo, quiero comunicarme directamente con los usuarios a través de un chat, para resolver dudas y dar seguimiento personalizado. | 8 | Alta |
| H-03 | Como paciente, quiero comunicarme directamente con los kinesiologos a través de un chat, para resolver dudas, obtener ayuda, entre otros. | 8 | Alta |
| H-05 | Como usuario, quiero consultar un historial de los ejercicios que he realizado, para hacer seguimiento de mi progreso. | 5 | Media |
| H-07 | Como usuario, quiero recibir notificaciones de nuevos mensajes de mi kinesiólogo, para no perder información importante. | 8 | Alta |
| H-18 | Como paciente, quiero buscar profesionales con sus descripciones, para encontrar la opción que mejor se adapte a mis necesidades. | 5 | Media |
| H-21 | Como kinesiólogo, quiero poder configurar de forma rápida y flexible mi disponibilidad horaria para que los pacientes solo puedan agendar sesiones en los momentos que me sean convenientes. | 5 | Alta |
| H-24 | Implementar el registro de rutinas completadas y el tiempo dedicado. | 5 | Alta |
| H-25 | Como kinesiólogo, quiero visualizar el progreso, métricas clave y la adherencia de mis pacientes, para evaluar la efectividad de la terapia. | 5 | Alta |
| H-26 | Como kinesiólogo, quiero poder suscribirme a un plan Premium para aumentar mi visibilidad, aparecer primero en las búsquedas y obtener más pacientes. | 5 | Alta |

## Retrospectiva del primer sprint

| ¿Qué salió bien en la iteración? (aciertos) | * Pudimos habilitar la comunicación directa entre paciente y kinesiólogo a través del chat (H-02, H-03), dejando operativo un canal de mensajería que soporta el seguimiento personalizado y la resolución de dudas en tiempo real. * Implementamos las notificaciones de nuevos mensajes (H-07), lo que hace que el usuario no dependa de entrar manualmente al chat para saber si tiene respuestas pendientes. * Avanzamos en el historial y registro de actividad del usuario (H-05, H-24), guardando rutinas completadas y tiempo dedicado, lo que alimenta directamente el módulo de progreso y adherencia. * Logramos que el kinesiólogo pueda configurar su disponibilidad horaria (H-21), alineando la agenda con su realidad y permitiendo que los pacientes solo reserven en horarios válidos. * Incorporamos el listado/búsqueda de profesionales con sus descripciones (H-18), mejorando la etapa de elección del kinesiólogo por parte del paciente. |
| --- | --- |
| ¿Qué no salió bien en la iteración? (errores) | * La combinación de chat + notificaciones + disponibilidad + métricas en un mismo sprint resultó muy exigente, y algunas partes quedaron menos pulidas de lo que esperábamos (por ejemplo, ciertos edge cases del chat o de las notificaciones). * En el módulo de progreso (H-25) nos dimos cuenta tarde de que necesitábamos definir mejor las métricas (qué considerar como adherencia, qué período analizar, qué gráficos mostrar), lo que provocó cambios de diseño sobre la marcha. * El flujo de suscripción Premium (H-26) no alcanzó a quedar completamente integrado con todas las pantallas (por ejemplo, visibilidad prioritaria en el directorio y marcadores visuales), quedando en una versión más básica de lo planificado. * La configuración de disponibilidad horaria (H-21) generó algunos problemas de validación y solapamiento de horarios, lo que obligó a revisar la lógica varias veces y consumir más tiempo del estimado. |
| ¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua) | * Reducir la cantidad de historias complejas en un mismo sprint, priorizando terminar bien los flujos críticos (chat + notificaciones y progreso básico) antes de sumar funciones avanzadas como suscripciones o métricas más complejas. * Definir de antemano los indicadores y métricas clave (para H-24 y H-25) antes de desarrollar, de modo que el modelo de datos y las pantallas se diseñen alineados desde el inicio. * Mejorar las pruebas integrales de flujo: no solo probar cada pantalla aisladamente, sino recorrer el proceso completo paciente–kinesiólogo para detectar incoherencias entre módulos (agenda, chat, progreso, notificaciones). * Documentar mejor los estados de las funcionalidades en tiempo real (chat, notificaciones, disponibilidad), para que todo el equipo tenga claro qué debe ocurrir cuando hay cambios (nuevos mensajes, cambios de horario, nuevas rutinas registradas). |

# Grafico del avance del sprint 5



## Pila del producto actualizada del primer sprint

| Id | Historia de usuario | Estimacion | Prioridad |
| --- | --- | --- | --- |
| H-09 | Como usuario, quiero recibir recordatorios de mis sesiones y rutinas, para mantener constancia en mi recuperación. | 5 | Media |
| H-19 | Como kinesiologo no verificado, quiero poder cargar un documento con el cual poder verificarme en la aplicación" | 5 | Media |
| H-22 | Como administrador, quiero poder ingresar a una página donde pueda ver los usuarios y sus datos. | 8 | Alta |
| H-23 | Como administrador, quiero tener un panel de solicitudes para revisar, aprobar o rechazar los documentos de verificación de los kinesiólogos. | 5 | Alta |
| H-27 | Como usuario (paciente o kinesiólogo), quiero poder filtrar el catálogo de ejercicios por parte del cuerpo a trabajar. | 8 | Alta |

## Retrospectiva del primer sprint

| ¿Qué salió bien en la iteración? (aciertos) | * Conseguimos implementar los recordatorios de sesiones y rutinas (H-09), lo que ayuda a que el usuario mantenga constancia en su recuperación y no dependa solo de su memoria. * Habilitamos el flujo para que el kinesiólogo no verificado pueda cargar sus documentos de respaldo (H-19), dejando formalizado el proceso de verificación desde el lado del profesional. * Creamos la página de administrador para ver usuarios y sus datos (H-22), lo que da una vista global del sistema y facilita la gestión interna. * Desarrollamos un panel de solicitudes de verificación (H-23), donde el administrador puede revisar, aprobar o rechazar documentos de kinesiólogos, cerrando el circuito de verificación que se había definido en sprints anteriores. |
| --- | --- |
| ¿Qué no salió bien en la iteración? (errores) | * La lógica de recordatorios (H-09) fue más compleja de lo esperado, especialmente al definir cada cuánto tiempo recordar, qué canal usar y cómo evitar recordatorios repetidos o molestos. * El flujo de carga de documentos del kinesiólogo (H-19) inicialmente no contemplaba todos los formatos o tamaños de archivo, lo que generó errores y obligó a ajustar validaciones. * En el panel del administrador (H-22, H-23) al principio mezclamos demasiada información en una sola vista, y fue necesario reorganizar tablas y filtros para que fuera realmente útil. * El filtro por parte del cuerpo (H-27) hizo evidente que la clasificación de ejercicios no estaba completamente estandarizada, y hubo que corregir o completar datos en la base para que los filtros devolvieran resultados coherentes. |
| ¿Qué mejoras vamos a implementar en la próxima iteración? (recomendaciones de mejora continua) | * Definir de forma más clara y temprana las reglas de negocio de los recordatorios (frecuencia, condiciones para enviarlos, forma de desactivar), para reducir retrabajos en la lógica. * Establecer un conjunto de requisitos mínimos para los documentos de verificación (formatos aceptados, tamaño máximo, campos obligatorios) y reflejarlos tanto en la UI como en la validación del backend. * Refinar el diseño del panel de administración, separando vistas (usuarios / solicitudes) y agregando filtros y paginación desde el inicio para que la pantalla sea más escalable. * Mejorar la normalización de los datos de ejercicios (zonas del cuerpo, etiquetas consistentes), antes de aplicar nuevos filtros avanzados en la interfaz. |

# Retrospectiva del Proyecto

La retrospectiva del proyecto corresponde a una evaluación global del trabajo realizado durante todo el desarrollo de *Un Kine Amigo*. En esta instancia se analizó el desempeño del equipo, la efectividad de la metodología Scrum y la calidad de los resultados obtenidos. Se identificaron los principales logros, como la correcta integración de módulos clave (Auth, Chat, Ejercicios, Citas y Notificaciones), así como los desafíos enfrentados, entre ellos la coordinación de tiempos, la gestión de cargas académicas y la resolución de dependencias técnicas.  
 Esta reflexión final permitió reconocer las buenas prácticas aplicadas, y dejó definidas oportunidades de mejora para futuros proyectos, especialmente en planificación, comunicación interna y validación temprana de entregables.

**Retrospectiva del proyecto :** [***Retrospectiva***](https://docs.google.com/document/d/1CDTEJJObdA2-9-I30DQ5cE3FGgJ6xejX/edit?usp=sharing&ouid=115770852576023431618&rtpof=true&sd=true)

# Plan de Pruebas

El Plan de Pruebas establece la estrategia, actividades y criterios necesarios para asegurar la calidad funcional, técnica y de experiencia de usuario de la aplicación Un Kine Amigo. Su propósito es validar que cada módulo —autenticación, planes de ejercicios, métricas, citas, chat, notificaciones y pagos— funcione correctamente, mantenga la integridad de los datos y cumpla con los requerimientos definidos en el proyecto.

El plan define el alcance de las pruebas, los tipos de test a ejecutar, el entorno utilizado (Flutter + Firebase), los roles responsables, el cronograma de ejecución en los cuatro meses del proyecto y los criterios de aceptación necesarios para aprobar cada funcionalidad.

Este documento guía al equipo en la detección temprana de errores, mejora continua del producto y asegura que la versión final del sistema sea estable, segura y lista para su despliegue.

**Plan de pruebas :** [**Plan de pruebas**](https://docs.google.com/document/d/1FaAhipNVkYYmZYrI5A_vU1p5jSGDm5Wg/edit?usp=sharing&ouid=115770852576023431618&rtpof=true&sd=true)

**Registro de ejecución del plan de pruebas:** [**Registro ejecucion plan de pruebas**](https://docs.google.com/spreadsheets/d/1sTOyx15M46ybJU6mxXmXxV_zl0vStP2S/edit?gid=839539889#gid=839539889)

# Control de Riesgos

El control de riesgos del proyecto Un Kine Amigo tiene por objetivo anticipar, gestionar y reducir los eventos que puedan afectar el desarrollo, la calidad o el funcionamiento de la aplicación. Dado que el proyecto utiliza tecnologías móviles, servicios en la nube y datos personales de usuarios, se identificaron riesgos técnicos, operacionales y de seguridad que pueden impactar los plazos y la estabilidad del sistema.

Durante el desarrollo se monitorearon riesgos tales como problemas de comunicación dentro del equipo, problemas técnicos en Firebase o Supabase, subestimación de tiempos, filtración de datos, errores en compatibilidad móvil y retrasos por falta de validación profesional. Para cada riesgo se definieron acciones de mitigación y se revisaron continuamente en las retrospectivas de cada sprint, permitiendo ajustar el plan, redistribuir tareas y evitar retrasos mayores.

**Control de riesgo :** [**Control de riesgo**](https://docs.google.com/document/d/1IC0kE-nsp6JbvXUOwyLlZp5Ji94xAq55NNlBYAex7jc/edit?usp=sharing)

# Desarrollo del Proyecto

El desarrollo del proyecto Un Kine Amigo se construyó siguiendo una arquitectura modular en Flutter, con una capa de datos completamente serverless basada en Firebase y Supabase. El objetivo técnico fue implementar un MVP funcional, escalable y seguro, utilizando buenas prácticas de ingeniería, arquitectura limpia y separación de responsabilidades entre UI, lógica de negocio y persistencia.

## Arquitectura Interna del Proyecto (Flutter)

El código se estructuró bajo una arquitectura por módulos, compuesta por:

* /features/ → módulos funcionales (Auth, Chat, Appointments, Exercises, Progress, Admin, Storage).
* /services/ → clases para comunicación con Firebase/Supabase.
* /models/ → entidades del dominio (User, Appointment, Exercise, Message, Metric, etc.).
* /screens/ → UI con widgets desacoplados y reutilizables.
* /controllers o providers → lógica del estado (Provider, StreamBuilder o ChangeNotifier).

Se utilizaron patrones como:

* Repository Pattern para desacoplar la capa de datos.
* DTOs para transformar documentos Firestore ↔ modelos Dart.
* Asynchronous programming (async/await, streams, Futures).

## Integraciones Técnicas con Firebase

### Firebase Authentication

Se implementaron los siguientes flujos:

* Login y registro por correo/contraseña.
* Integración con Google Sign-In usando OAuth 2.0.
* Gestión de sesiones mediante **FirebaseAuth.instance.authStateChanges().**
* Validación de roles almacenados en Firestore:
  + *paciente*
  + *kinesiólogo*
  + *administrador*

### Cloud Firestore (Base de Datos Principal)

Firestore se utilizó para todas las entidades clave:

* /usuarios
* /citas
* /ejercicios
* /planes
* /chats
* /chats/{id}/mensajes
* /solicitudesKine/metricas
* /tokens\_fcm

Técnicamente se implementó:

* Consultas indexadas (incluso compuestas).
* Paginación de listas para ejercicios y profesionales.
* Snapshot listeners con sincronización en tiempo real.
* Subcolecciones para escalabilidad (por ejemplo, mensajes dentro de cada chat).
* Optimización de reads y writes para reducir costos.

### Firebase Cloud Messaging (FCM)

Para notificaciones:

* Obtención y persistencia del token FCM en Firestore.
* Envío de notificaciones push desde Cloud Functions.
* Notificaciones silenciosas para actualización interna.

### Cloud Functions v2 (Node.js)

Se desarrollaron funciones:

#### Triggers

* **onCreate** en mensajes → notificación push.
* **onCreate** en citas → notificación al kinesiólogo.
* **onUpdate** en citas → notificación de cambios.

#### Funciones programadas (cron)

* Recordatorios de citas.
* Limpieza de datos antiguos.

#### Validaciones automáticas

* Seguridad adicional sobre datos escritos.
* Normalización de campos críticos.

### Firestore Security Rules

Se implementaron reglas para:

* Acceso basado en UID.
* Lecturas restringidas por rol.
* Permisos específicos por colección.
* Validaciones de integridad al crear o actualizar documentos.
* Restricciones severas para documentos de verificación.

## Integración con Supabase Storage

Supabase se utilizó exclusivamente para carga de documentos profesionales, implementando:

* Subida de archivos con supabase.storage.from(bucket).upload().
* Generación de URLs firmadas temporales.
* Almacenamiento seguro de metadatos en Firestore.
* Validación administrativa desde el panel admin.

## Implementación Técnica de las Funcionalidades del Sistema

### Autenticación y Gestión de Usuarios

* Flujo completo con Firebase Auth.
* Asignación y actualización de roles.
* Manejo de tokens FCM.
* Persistencia de datos en Firestore.

### Módulo de Ejercicios

* Listas dinámicas cargadas desde Firestore.
* Filtros por:
  + tipo de lesión
  + parte del cuerpo
  + dificultad
* Reproductor de video integrado.
* Modelos optimizados para carga rápida.

### Agenda y Citas

* Gestión de disponibilidad del kinesiólogo.
* Reservas de citas y cambios de estado.
* Sincronización en tiempo real.
* Notificaciones automáticas por Cloud Functions.

### Chat en Tiempo Real

* Streams de Firestore para mensajes.
* Subcolecciones /mensajes.
* Indicadores:
  + mensajes no leídos
  + últimos mensajes
  + timestamps ordenados
* Notificaciones push automáticas.

### Progreso y Métricas

* Registro de ejercicios completados.
* Cálculo de adherencia.
* Panel profesional con métricas por paciente.

### Panel Profesional / Administrativo

* Gestión de pacientes.
* Validación de kinesiólogos.
* Visualización de datos críticos (consultas optimizadas).

### Procesos de Desarrollo y Construcción

* Control de versiones mediante GitHub (branches feature → develop).
* Code Review en PRs antes de merge.
* Testing básico con usuarios internos.
* Uso del Firebase Emulator Suite para pruebas locales.
* Optimización final del proyecto para build release.

# Plan de Capacitación

El Plan de Capacitación del proyecto Un Kine Amigo tiene como objetivo asegurar que tanto el equipo de desarrollo como los usuarios finales (pacientes y kinesiólogos) dispongan de las competencias necesarias para utilizar, mantener y escalar la aplicación. Se divide en dos grandes componentes: capacitación interna (equipo del proyecto) y capacitación externa (usuarios finales).

**Plan de Capacitacion:**[**Plan de Capacitacion**](https://docs.google.com/document/d/14z-RnProS5vSz2YW-YlR8HFEC5nOGzrB/edit?usp=sharing&ouid=115770852576023431618&rtpof=true&sd=true)

# Manuales de Usuario

Los manuales de usuario describen de forma clara y guiada cómo utilizar cada una de las funcionalidades de la aplicación “Un Kine Amigo”. Estos documentos están dirigidos tanto a pacientes como a kinesiólogos, y explican el proceso de registro, navegación, uso de planes de ejercicios, agendamiento de citas, comunicación por chat y visualización de métricas de progreso.

Incluyen capturas de pantalla, pasos detallados y buenas prácticas para garantizar una experiencia simple, segura y eficiente. Los manuales sirven como apoyo durante la puesta en marcha y reducen la curva de aprendizaje de los usuarios finales.

**Manual de usuario :** [**Manual de usuario**](https://docs.google.com/document/d/1SXC1UWt2uCg4hdWSZXLsEotM4PVpQ3_7tLJ5ujH-yGQ/edit?usp=sharing)

# Arquitectura del Sistema

# La arquitectura del sistema Un Kine Amigo se diseñó con un enfoque modular, escalable y serverless, utilizando tecnologías modernas de desarrollo móvil (Flutter) y backend administrado en la nube (Firebase y Supabase). Esta arquitectura permite entregar un producto seguro, eficiente y mantenible, evitando la necesidad de servidores tradicionales y reduciendo costos operativos.

## Arquitectura General del Sistema

La arquitectura del sistema Un Kine Amigo se diseñó bajo un enfoque cliente–servidor serverless, priorizando la escalabilidad, el bajo costo operativo y la simplicidad de mantenimiento. El sistema está compuesto principalmente por una aplicación móvil desarrollada en Flutter, que consume servicios en la nube gestionados por Firebase y Supabase.

### La aplicación está compuesta por:

* Frontend móvil: desarrollado en Flutter, desplegado en dispositivos Android.
* Backend serverless:compuesto por Firebase Auth, Firestore, Cloud Functions, FCM y Supabase Storage.
* Servicios complementarios: manejo de notificaciones, validación de kinesiólogos, almacenamiento de documentos y registro de métricas.

### Esta arquitectura permite:

* Sincronización en tiempo real (chat, citas, progreso).
* Escalabilidad automática de la carga.
* Seguridad gestionada mediante Reglas de Firestore y tokens de autenticación.
* Bajo costo operativo al no requerir servidores dedicados.

**Documento Das :**[**Das**](https://docs.google.com/document/d/12iGzIA0RoW_HP2vXvAPNpUTpNbwT7MEr/edit)

# Arquitectura Lógica Componentes de Software

A nivel lógico, el sistema se organiza en módulos de software bien definidos, cada uno con responsabilidades específicas. Esta modularidad facilita el mantenimiento, la reutilización de código y la incorporación de nuevas funcionalidades en el futuro.

## Módulo de Autenticación

* Inicio de sesión con correo y contraseña.
* Login con Google.
* Registro y validación.
* Generación y almacenamiento de tokens FCM.

Tecnologías: Firebase Auth + Firestore.

## Módulo de Gestión de Usuarios

* Perfil de paciente.
* Perfil de kinesiólogo.
* Flujo de activación profesional.
* Validación de documentos.

Tecnologías: Firestore + Supabase Storage.

## Módulo de Ejercicios y Rutinas

* Catálogo de ejercicios por:
  + tipo de lesión,
  + parte del cuerpo,
  + nivel de dificultad.
* Visualización de rutinas con video.
* Registro de ejercicios completados.

Tecnologías: Firestore + Firebase Hosting (videos embebidos desde YouTube).

## Módulo de Citas / Agenda

* Reserva de horas.
* Calendario dinámico.
* Disponibilidad del kinesiólogo.
* Cancelación / modificación de sesiones.

Tecnologías: Firestore (colección citas) + Cloud Functions (notificaciones).

## Módulo de Chat en Tiempo Real

* Mensajería bidireccional paciente → kinesiólogo.
* Contador de mensajes no leídos.
* Notificaciones internas por nuevos mensajes.

Tecnologías: Firestore (subcolección mensajes) + FCM + Cloud Functions.

## Módulo de Progreso y Métricas

* Registro de ejecuciones de rutinas.
* Panel para kinesiólogo con información por paciente.
* Estadísticas simples de cumplimiento.

Tecnologías: Firestore.

## Módulo Administrativo

* Revisión de documentos enviados por kinesiólogos.
* Aprobación / rechazo de solicitudes.
* Gestión del listado de usuarios.

Tecnologías: Firestore + Supabase Storage.

# Arquitectura Física

La arquitectura física describe cómo se despliegan los componentes del sistema en la infraestructura tecnológica

## Capa Cliente (Frontend Flutter)

Ejecutada en dispositivos Android.  
 Incluye:

* Pantallas de login, registro, perfil, chat, citas, ejercicios.
* Controladores y servicios para acceso a Firebase.
* Manejo de estado (Provider / Riverpod / setState según implementación).

## Capa Backend Serverless

### Firebase Authentication

* Maneja usuarios, sesiones, tokens y permisos.
* Controla accesos a recursos protegidos.

### Firestore Database

Base NoSQL donde residen todas las entidades:

* usuarios
* tipo\_usuario
* solicitudes\_kine
* citas
* planes
* ejercicios
* chats
* mensajes
* métricas
* tokens\_fcm

Organizada por colecciones y subcolecciones para optimizar consultas.

### Cloud Functions

Funcionalidades automatizadas

* Notificación de nuevos mensajes.
* Recordatorios de citas.
* Actualización de estados.
* Ejecución de tareas programadas (cron).

### Firebase Cloud Messaging (FCM)

Envía alertas al dispositivo por

* nuevos mensajes,
* cambios en citas,
* actualizaciones importantes.

## Capa de Almacenamiento Externo (Supabase Storage)

Utilizado para:

* Documentos de verificación enviados por kinesiólogos.
* Archivos sensibles o PDF.

Permite:

* Subida segura
* Lectura mediante URLs firmadas
* Control de acceso por reglas

# Flujos Principales del Sistema

## Flujo de Login

* Usuario ingresa correo/Google.
* Firebase Auth verifica identidad.
* Se obtiene UID.
* Se carga perfil asociado desde Firestore.
* Se almacena token FCM.

## Flujo de Chat

* Usuario escribe un mensaje.
* Mensaje se guarda en subcolección **/chats/{chatID}/mensajes**.
* Firestore emite evento en tiempo real al receptor.
* Cloud Function detecta mensaje → envía notificación vía FCM.

## Flujo de Agenda

* Paciente selecciona kinesiólogo.
* Consulta disponibilidad almacenada en Firestore.
* Reserva cita → se crea documento en **/citas.**
* El kinesiólogo recibe notificación interna.

## Flujo de Verificación de Kinesiólogo

* Profesional sube documento a Supabase.
* Firestore crea solicitud con metadatos.
* Admin revisa y aprueba / rechaza.  
  Se actualiza rol en Firestore.

# Seguridad del Sistema

La arquitectura integra múltiples medidas de seguridad

## Reglas de Firestore

* Usuarios solo pueden leer su propio perfil.
* Kinesiólogos solo acceden a pacientes asignados.
* Chats solo accesibles a sus dos participantes.
* Citas solo visibles para paciente y kinesiólogo involucrados.

## Seguridad en Supabase

* Documentos accesibles solo con URLs firmadas.
* Eliminación automática cuando expira el tiempo.

## Validación de Roles

* Tipo de usuario (paciente / kine / admin) definido en Firestore.
* Accesos controlados por nivel.

## Cloud Functions

* Sanitizan datos.
* Evitan accesos indebidos.
* Controlan flujos críticos (citas, mensajes).

# Beneficios de la Arquitectura

La arquitectura seleccionada para **Un Kine Amigo**, basada en Flutter como framework de desarrollo y en servicios cloud serverless como Firebase y Supabase, ofrece múltiples ventajas técnicas, operativas y económicas que fortalecen la estabilidad y evolución del sistema.

## Alta escalabilidad

Firebase y Supabase son plataformas con capacidad de escalar automáticamente según la carga de usuarios, permitiendo que la aplicación soporte desde decenas hasta miles de conexiones simultáneas sin necesidad de configurar infraestructura adicional. Esto asegura un crecimiento sostenido del proyecto sin rediseños complejos.

## Bajo costo operativo

Al estar construida sobre un backend serverless, la solución no requiere servidores propios ni mantenimiento de infraestructura física. Los costos se basan en consumo real (lecturas, escrituras, almacenamiento y transferencias), lo que resulta especialmente favorable para un MVP en etapa académica y para un proyecto que busca reducir gastos iniciales.

## Alta disponibilidad y resiliencia

Los servicios de Firebase y Supabase están desplegados globalmente y cuentan con mecanismos internos de replicación, redundancia y tolerancia a fallos. Esto permite ofrecer una aplicación accesible 24/7 con una probabilidad muy baja de caídas prolongadas, incluso ante picos de uso o problemas regionales.

## Velocidad de desarrollo y mantenimiento

Implementar Flutter como base del frontend permite reutilizar más del 90% del código para Android y iOS, reduciendo significativamente los tiempos de desarrollo y facilitando la incorporación de nuevas funcionalidades. Al contar con widgets reutilizables y un flujo de compilación rápido (hot reload), se agiliza la iteración durante los sprints.

## Actualizaciones rápidas y sincronización en tiempo real

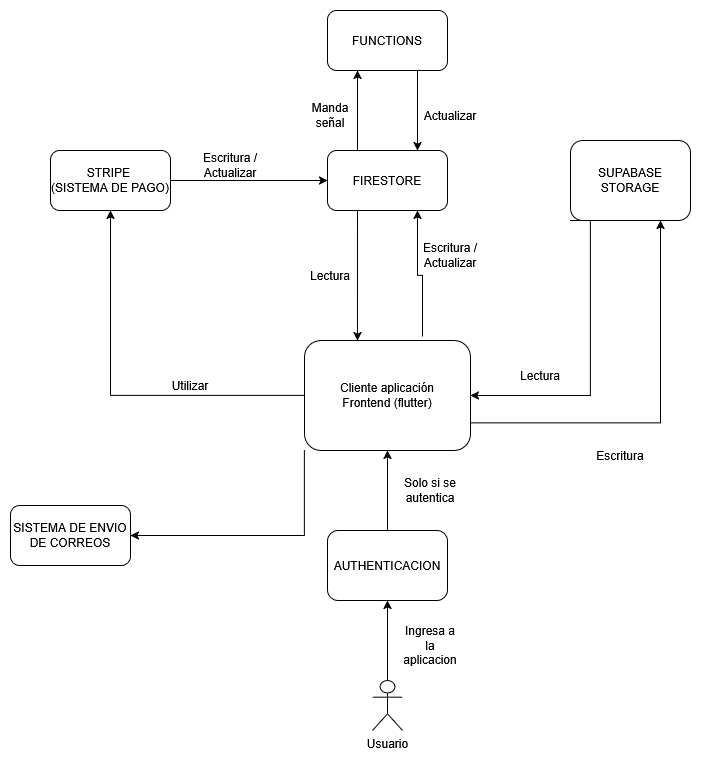
Cloud Firestore incorpora listeners reactivos que permiten actualizar datos instantáneamente en todos los dispositivos conectados, sin necesidad de recargar la aplicación. Esto es clave para módulos como chat, agenda, métricas y estados de verificación, asegurando una experiencia fluida y moderna.

## Seguridad robusta y control de acceso avanzado

La arquitectura integra múltiples capas de seguridad:

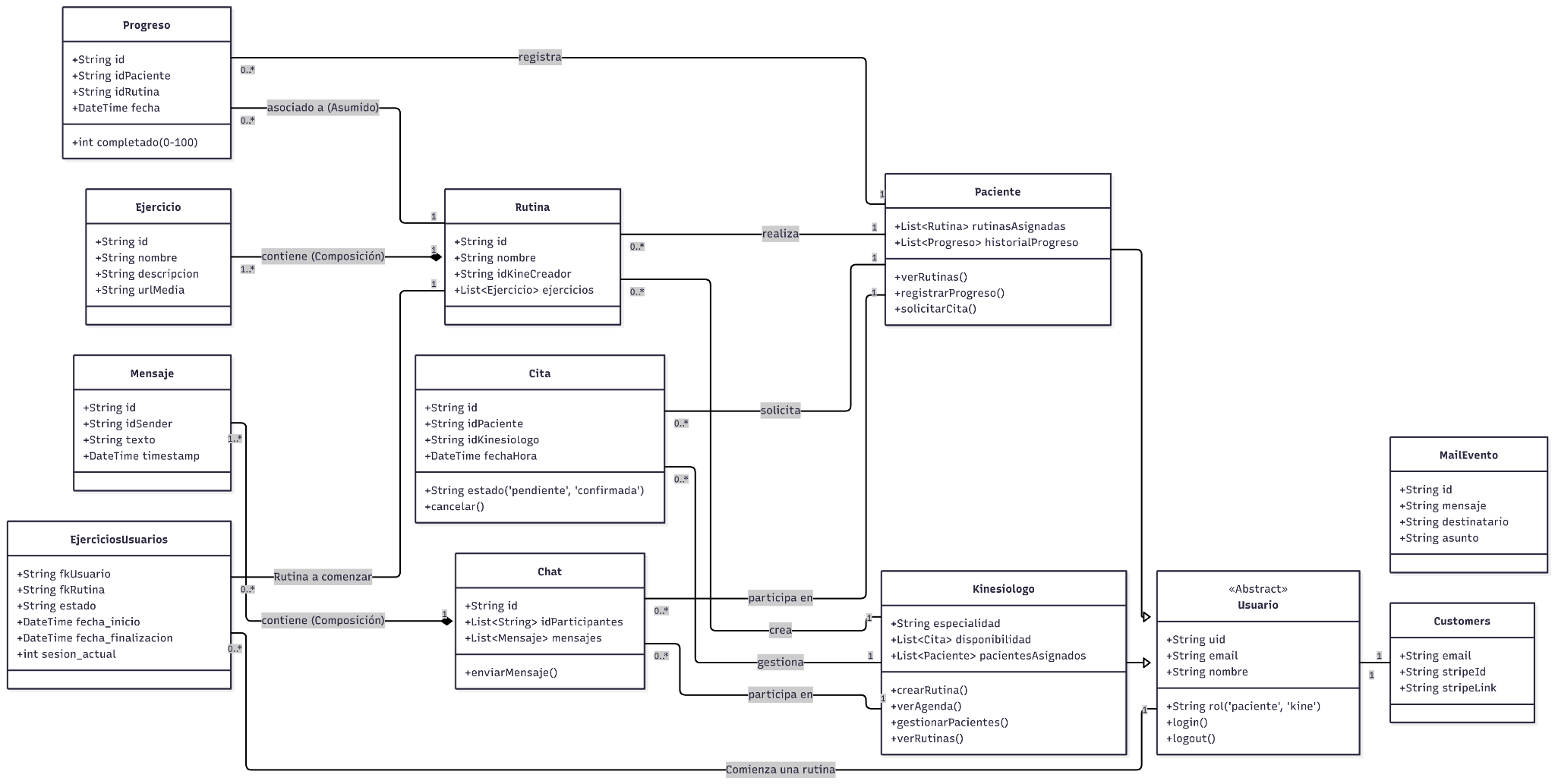
* Reglas de acceso en Firestore basadas en roles y UID.
* Autenticación segura mediante Firebase Auth.
* Cifrado de datos en tránsito (HTTPS/TLS) y en reposo.
* Control de documentos sensibles mediante URLs firmadas de Supabase.

Esto garantiza protección de datos personales y confidenciales, cumpliendo buenas prácticas del manejo seguro de información.



# Modelo de Base de Datos UML

El modelo de base de datos del proyecto Un Kine Amigo se diseñó bajo una arquitectura NoSQL orientada a documentos, utilizando Cloud Firestore como base principal y Supabase Storage como repositorio externo para archivos sensibles.  
 El modelo sigue un esquema modular, altamente escalable, organizado por colecciones independientes y subcolecciones que permiten un rendimiento óptimo y un acceso seguro basado en roles.



# Puntos de mejora

**1. Mejoras Técnicas y de Calidad (QA)**

* **Automatización de Pruebas (Testing)**: Actualmente, el proyecto cuenta con pruebas básicas manuales y con usuarios internos. Un punto de mejora crítico es la implementación de un pipeline de CI/CD que incluya pruebas unitarias y de integración automatizadas, garantizando que nuevos cambios no rompan funcionalidades existentes.
* **Optimización del Modo Offline**: Si bien Firestore ofrece persistencia básica, se debe mejorar la experiencia "Offline-First". Esto implica gestionar el cómo los usuarios pueden utilizar la aplicación sin estar conectados a internet, por ejemplo: puedan ver sus rutinas y videos descargados sin conexión a internet.

**2. Mejoras Metodológicas y de Gestión**

* Precisión en la Estimación de Tiempos: Durante el desarrollo se identificó la subestimación de tiempos como un riesgo y un problema recurrente . Como mejora, se propone utilizar datos históricos de velocidad del equipo para calibrar mejor los Story Points en futuros Sprints, considerando márgenes de holgura para la curva de aprendizaje de nuevas tecnologías.
* Validación Temprana con Usuarios Reales: Aunque hubo validación interna, el proyecto se beneficiaría de pruebas de usabilidad con pacientes reales y kinesiólogos externos en etapas más tempranas del desarrollo (fase de prototipo), para ajustar flujos de navegación antes de escribir código definitivo.

**3. Mejoras Funcionales y de Producto**

* Automatización de Procesos Administrativos: Actualmente, la validación de kinesiólogos requiere intervención manual de un administrador para revisar documentos en Supabase . Una mejora sustancial sería integrar servicios de reconocimiento de texto (OCR) o IA para realizar una pre-validación automática de los certificados profesionales, agilizando el onboarding .
* Accesibilidad Universal (WCAG): Dado que el público objetivo incluye adultos mayores o personas en rehabilitación , es imperativo mejorar la accesibilidad de la UI. Esto incluye soporte completo para lectores de pantalla (TalkBack), modos de alto contraste y la implementación futura de control por voz mencionada en la visión del proyecto .

# Lecciones aprendidas

Tras finalizar el ciclo de desarrollo de "Un Kine Amigo", el equipo ha identificado aprendizajes clave que fortalecen nuestra formación profesional y la calidad de futuros proyectos. Estas lecciones se dividen en gestión, técnica y trabajo en equipo:m

1. **Gestión del Tiempo y Metodología Ágil**

• Importancia de la Estimación Realista: Aprendimos que subestimar la complejidad de ciertas tareas puede impactar el flujo del Sprint . La utilización del Planning Poker fue útil, pero aprendimos que se debe considerar un margen de holgura para imprevistos y cargas académicas externas que afectan la disponibilidad del equipo.

• El valor de las ceremonias Scrum: Las Daily Meetings y Retrospectivas no fueron solo trámites, sino herramientas vitales. Nos permitieron detectar bloqueos a tiempo y, lo más importante, ajustar la estrategia de trabajo sprint a sprint, mejorando nuestra velocidad y organización progresivamente .

1. **Aprendizaje Técnico y Arquitectura**

• Eficacia del Modelo Serverless: Comprobamos que para un MVP (Producto Mínimo Viable) con plazos acotados, una arquitectura serverless es muy rápida para el proyecto realizado. Aprendimos que esto reduce drásticamente los costos operativos y tiempos de configuración, permitiendo centrarnos en la lógica de negocio y la experiencia del usuario .

• Modularidad en Flutter: Aprendimos que mantener una arquitectura limpia y modular en el frontend es crucial. Separar la lógica (Providers/Controllers) de la interfaz (UI).

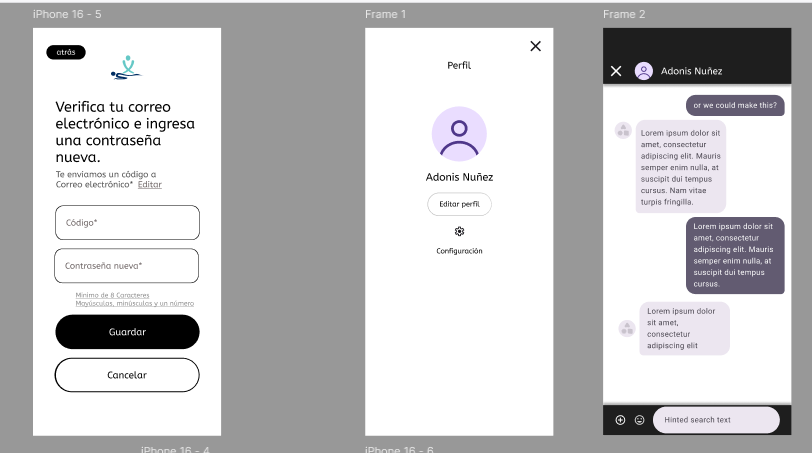
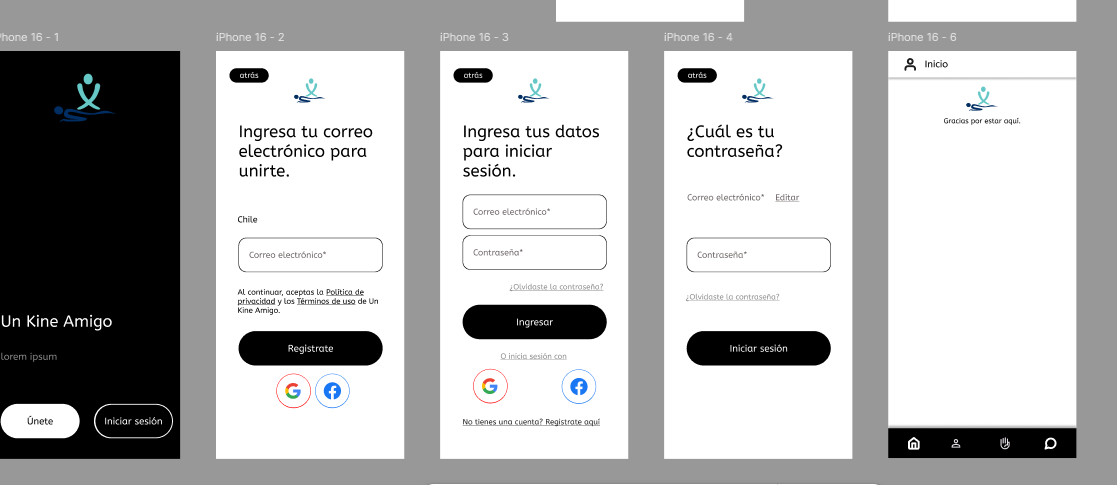
1. **Trabajo en Equipo y Comunicación**

• **Comunicación Asíncrona y Documentación:** Debido a los distintos horarios del equipo, aprendimos que no basta con hablar; es necesario dejar evidencia clara en las herramientas de gestión (Trello y GitHub)

• **Gestión de la Incertidumbre:** Enfrentamos el desafío de trabajar con requisitos que evolucionaron (como la validación de kinesiólogos). Aprendimos a ser flexibles y adaptar el Product Backlog priorizando siempre lo que aportaba mayor valor inmediato al usuario final, en lugar de aferrarnos a un plan rígido inicial .

# Mockups

Los mockups fueron hechos con figma



# Condiciones de aceptación para el cierre del proyecto.

El presente apartado establece los criterios y condiciones que deben cumplirse para que el proyecto de la aplicación móvil se considere cerrado. Estos criterios garantizan la calidad del producto, el cumplimiento del alcance y la madurez del proceso de desarrollo.

### 1. Cumplimiento Funcional y de Alcance

La aceptación se basará en el principio de que la funcionalidad principal debe estar operativa y estable.

* **Alcance Mínimo:** Se establece como condición esencial que al menos el **80%** de las funcionalidades totales especificadas en el *Documento de Requisitos* o las *User Stories* estén completamente implementadas y sean demostrables.
* **Funciones principales:** El 100% de las funcionalidades catalogadas como "Principales" (como la autenticación de usuarios, la navegación principal y el flujo de datos esencial) deben operar sin fallos, *crashes* o errores de bloqueo.
* **Flujo Completo Demostrable:** El equipo debe ser capaz de ejecutar un flujo de usuario clave y complet**o** (ej: desde el registro hasta la finalización de la tarea principal de la *app*) sin interrupciones técnicas.

### 2. Estabilidad Técnica y Rendimiento

Para asegurar que la aplicación es un producto usable y bien construido, se exigen las siguientes condiciones técnicas:

* **Estabilidad de la Aplicación:** La aplicación no debe presentar cierres inesperados (*crashes*) durante un período continuo de 10 minutos de uso activo y pruebas funcionales.
* **Tiempo de Respuesta:** Los tiempos de carga de pantallas y las respuestas a las interacciones del usuario (ej: al presionar un botón de envío) no deben superar los 3 segundos bajo condiciones normales de *testing*.

### 3. Usabilidad (UI/UX) y Experiencia del Usuario

La interacción del usuario debe ser intuitiva y profesional, lo cual se medirá mediante:

* **Coherencia de la Interfaz:** La interfaz de usuario debe mantener una **coherencia visual alta** (utilizando la misma tipografía, colores y estilos de botones) en al menos el 90% de sus componentes para facilitar la experiencia del usuario.

# Conclusiones

El proyecto **“Un Kine Amigo”** logró desarrollar un MVP funcional y sólido que responde a problemas reales de la rehabilitación física, especialmente la baja adherencia, la falta de seguimiento y la limitada comunicación entre pacientes y kinesiólogos. Durante cuatro meses de trabajo bajo metodología Scrum, el equipo construyó una aplicación moderna, escalable y segura utilizando Flutter, Firebase y Supabase, aplicando buenas prácticas de arquitectura, modularidad y control de versiones.

Se implementaron los módulos esenciales: autenticación, chat en tiempo real, agenda de citas, catálogo de ejercicios, métricas de progreso y validación de profesionales. Las herramientas de gestión y pruebas permitieron mantener un desarrollo ordenado, detectar errores tempranos y asegurar la calidad final del producto.

El resultado es una plataforma estable y técnicamente robusta, que aporta valor al área de salud digital y deja una base preparada para futuras mejoras, como integración de IA, automatización de procesos y mayor accesibilidad. En síntesis, el proyecto cumplió sus objetivos, entregando una solución viable, escalable y alineada a las necesidades reales de usuarios y profesionales.