

# Índice de Documentación - Sistema Prácticas 1 a 1

**Proyecto:** SpeaklyPlan

**Fecha:** 17 de octubre de 2025

**Tema:** Análisis, mejoras y plan de implementación del sistema 1 a 1

---

## Documentos Generados

### 1. Resumen Ejecutivo

**Archivo:** RESUMEN\_EJECUTIVO\_1A1.md | RESUMEN\_EJECUTIVO\_1A1.pdf

**Para:** Tomadores de decisión, Product Managers

**Duración lectura:** 5 minutos

**Contenido:** - Situación actual del sistema - Problema principal identificado - Solución propuesta en 4 fases - Inversión y ROI esperado - Comparativa con competencia - Métricas de éxito - Mockups visuales - Recomendación final

**Cuándo leer:** - Antes de aprobar el proyecto - Para entender el panorama completo en 5 minutos - Para presentar a stakeholders

---

### 2. Análisis Completo y Plan de Mejoras

**Archivo:** ANALISIS\_Y\_MEJORAS\_1A1.md | ANALISIS\_Y\_MEJORAS\_1A1.pdf

**Para:** Product Managers, Tech Leads, Desarrolladores

**Duración lectura:** 20 minutos

**Contenido:** - Estado actual del sistema (backend vs frontend) - Análisis de funcionalidades existentes - Problemas identificados (5 críticos) - Plan de mejoras detallado (4 fases) - Agendamiento inteligente con Google Calendar - Sala de práctica integrada (Daily.co) - Priorización y roadmap - Wireframes y mockups - Métricas de éxito por fase

**Cuándo leer:** - Antes de comenzar la implementación - Para entender arquitectura actual - Para planificar sprints

---

### 3. Guía de Implementación - Google Calendar

**Archivo:** GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.md | GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.pdf

**Para:** Desarrolladores, DevOps

**Duración lectura:** 30 minutos + implementación

**Contenido:** - Configuración inicial paso a paso - Código completo de backend (TypeScript) - Código completo de frontend (React) - API routes para OAuth - Servicio de Calendar - Componentes UI listos para copiar/pegar - Testing y troubleshooting - Modelo Prisma actualizado - Checklist de implementación

**Cuándo usar:** - Durante la implementación de Fase 2 - Como referencia técnica - Para copiar código específico

---

### 4. Comparativa de Herramientas

**Archivo:** COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.md | COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.pdf

**Para:** Tech Leads, Arquitectos, Product Managers

**Duración lectura:** 15 minutos

**Contenido:** - Evaluación de 6 herramientas: - Google Calendar API - Calendly - Cal.com - Daily.co - Jitsi Meet - Zoom API - Pros/contras de cada una - Análisis de costos detallado - Matriz de decisión - Recomendaciones por caso de uso - Proyección de costos Año 1 y 2

**Cuándo leer:** - Antes de decidir stack tecnológico - Para justificar decisiones técnicas - Para comparar alternativas

---

## 5. Plan MVP Original (Referencia)

**Archivo:** MVP\_PRACTICAS\_1A1\_PLAN.md | MVP\_PRACTICAS\_1A1\_PLAN.pdf

**Para:** Referencia histórica

**Duración lectura:** 30 minutos

**Contenido:** - Plan original del MVP - Arquitectura de datos (Prisma schemas) - API endpoints implementados - Estructura frontend (planeada pero no implementada) - Flujos de usuario - Sistema de notificaciones - Gamificación

**Cuándo leer:** - Para entender el diseño original - Para ver modelos de datos completos - Como referencia de arquitectura

---

## Guía de Lectura por Rol

### Para CEO / Founder

**Leer en orden:** 1. RESUMEN\_EJECUTIVO\_1A1.pdf (5 min) 2. COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.pdf (Sección “Proyección de Costos”)

**Decisión a tomar:** ¿Aprobamos las 4 semanas de desarrollo?

---

### Para Product Manager

**Leer en orden:** 1. RESUMEN\_EJECUTIVO\_1A1.pdf (5 min) 2. ANALISIS\_Y\_MEJORAS\_1A1.pdf (20 min) 3. COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.pdf (15 min)

**Acciones:** - Priorizar sprints - Definir métricas de éxito - Planificar testing con usuarios

---

### Para Tech Lead / Arquitecto

**Leer en orden:** 1. ANALISIS\_Y\_MEJORAS\_1A1.pdf (Sección “Arquitectura”) 2. COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.pdf (completo) 3. MVP\_PRACTICAS\_1A1\_PLAN.pdf (Sección “Arquitectura de Datos”)

**Acciones:** - Revisar modelos de Prisma - Elegir herramientas definitivas - Planificar infraestructura

---

### Para Desarrollador Frontend

**Leer en orden:** 1. ANALISIS\_Y\_MEJORAS\_1A1.pdf (Sección “Fase 1: UI Básica”) 2. GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.pdf (Sección “Código Frontend”)

**Acciones:** - Crear componentes UI - Implementar modales - Integrar con APIs existentes

---

## Para Desarrollador Backend

**Leer en orden:** 1. ANALISIS\_Y\_MEJORAS\_1A1.pdf (Sección “Estado Actual”) 2. GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.pdf (Sección “Código Backend”) 3. MVP\_PRACTICAS\_1A1\_PLAN.pdf (Sección “API Endpoints”)

**Acciones:** - Implementar servicio de Calendar - Crear API routes OAuth - Actualizar modelos Prisma

---

## Para QA / Tester

**Leer en orden:** 1. RESUMEN\_EJECUTIVO\_1A1.pdf (Sección “Flujos de Usuario”) 2. ANALISIS\_Y\_MEJORAS\_1A1.pdf (Sección “Flujos de Usuario”) 3. GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.pdf (Sección “Testing”)

**Acciones:** - Crear casos de prueba - Verificar flujos completos - Testear integraciones externas

---

## Roadmap Visual

Semana 1: MVP UI

ANALISIS\_Y\_MEJORAS\_1A1.pdf → Sección "Fase 1"  
Crear: /app/practice/page.tsx + componentes

Semana 2: Google Calendar

GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.pdf → Seguir paso a paso  
Implementar: OAuth + Calendar service + UI

Semana 3: Video Integrado

COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.pdf → Daily.co  
Implementar: Sala de práctica con video

Semana 4: Pulido

ANALISIS\_Y\_MEJORAS\_1A1.pdf → Sección "Fase 4"  
Implementar: Matching, stats, optimizaciones

---

## Quick Start

¿Nuevo en el proyecto? Empieza aquí:

**5 minutos:** 1. Lee RESUMEN\_EJECUTIVO\_1A1.pdf 2. Entiende el problema y la solución

**30 minutos:** 3. Lee ANALISIS\_Y\_MEJORAS\_1A1.pdf 4. Comprende arquitectura actual y plan

**1 hora:** 5. Lee GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.pdf 6. Prepárate para implementar

**2 horas:** 7. Lee COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.pdf 8. Entiende alternativas y decisiones

**3 horas:** 9. Revisa MVP\_PRACTICAS\_1A1\_PLAN.pdf 10. Conoce diseño original completo

---

## Conceptos Clave

Problema Principal

Backend completo (90%) pero sin UI = Sistema inutilizable

## Solución

4 semanas de desarrollo frontend + integraciones

## Stack Recomendado

Agendamiento: Google Calendar API (gratis)

Video: Daily.co (10K min/mes gratis)

Frontend: Next.js + shadcn/ui

Backend: Ya existe (Prisma + PostgreSQL)

## Inversión

Tiempo: 4 semanas

Costo: \$0 → \$99/mes (al escalar)

ROI: +40% engagement

---

## Estructura de Archivos

speaklyplan/

RESUMEN\_EJECUTIVO\_1A1.md/pdf

Para: Decisión ejecutiva

ANALISIS\_Y\_MEJORAS\_1A1.md/pdf

Para: Planning y arquitectura

GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.md/pdf

Para: Implementación técnica

COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.md/pdf

Para: Decisiones de stack

MVP\_PRACTICAS\_1A1\_PLAN.md/pdf

Para: Referencia original

INDICE\_DOCUMENTACION\_1A1.md

Este archivo (navegación)

---

## Checklist de Lectura

### Antes de Empezar Implementación

- ☐ Leí resumen ejecutivo
- ☐ Entiendo el problema actual
- ☐ Revisé el plan de 4 semanas
- ☐ Conozco las herramientas recomendadas
- ☐ Sé cuánto costará

### Antes de Sprint 1 (UI)

- ☐ Revisé wireframes en ANALISIS\_Y\_MEJORAS
- ☐ Conozco componentes a crear
- ☐ Entiendo flujos de usuario

- ☐ Tengo diseños de referencia

### Antes de Sprint 2 (Calendar)

- ☐ Leí guía de Google Calendar completa
- ☐ Tengo cuenta en Google Cloud Console
- ☐ Entiendo flujo OAuth
- ☐ Sé cómo generar Meet links

### Antes de Sprint 3 (Video)

- ☐ Comparé opciones en COMPARATIVA
- ☐ Decidí usar Daily.co
- ☐ Creé cuenta en Daily.co
- ☐ Entiendo API de Daily

### Antes de Sprint 4 (Pulido)

- ☐ Revisé métricas de éxito
- ☐ Planeé testing con usuarios
- ☐ Definí KPIs a medir
- ☐ Preparé feedback loops

---

## Glosario

**MVP:** Minimum Viable Product (producto mínimo viable)

**OAuth:** Protocolo de autorización para acceder a APIs externas

**WebRTC:** Tecnología para video/audio en tiempo real en navegador

**Meet link:** URL para videollamada (Google Meet, Zoom, etc.)

**Prisma:** ORM (Object-Relational Mapping) para base de datos

**API Route:** Endpoint de backend en Next.js

**shadcn/ui:** Librería de componentes UI usada en el proyecto

---

## Soporte

¿Tienes dudas sobre...?

**Decisiones de negocio:** → Lee RESUMEN\_EJECUTIVO\_1A1.pdf

**Arquitectura técnica:** → Lee ANALISIS\_Y\_MEJORAS\_1A1.pdf

**Implementación específica:** → Lee GUIA\_IMPLEMENTACION\_GOOGLE\_CALENDAR.pdf

**Alternativas de herramientas:** → Lee COMPARATIVA\_HERRAMIENTAS\_AGENDAMIENTO.pdf

**Diseño original:** → Lee MVP\_PRACTICAS\_1A1\_PLAN.pdf

---

## Estado Actual del Proyecto

**Última actualización:** 17 de octubre de 2025

**Fase actual:** Análisis completado

**Próxima fase:** Decisión de implementación

**Bloqueadores:** Ninguno (todo listo para empezar)

**Backend:** 90% completo - Modelos de datos - API endpoints - Servicios de lógica - Notificaciones

**Frontend:** 0% completo - Página principal - Componentes UI - Modales - Integraciones

**Integraciones:** 0% completo - Google Calendar - Daily.co - OAuth

**Documentación:** 100% completa - Análisis - Plan de implementación - Guías técnicas - Comparativas

---

## Próximos Pasos

1. **Decisión:** Aprobar o rechazar plan de 4 semanas
  2. **Si aprobado:**
    - Asignar desarrollador(es)
    - Crear proyecto en Google Cloud Console
    - Comenzar Sprint 1 (UI)
  3. **Si rechazado:**
    - Feedback sobre plan
    - Ajustar propuesta
    - Re-evaluar
- 

**Todo listo para empezar**

**¿Preguntas?** Consulta el documento correspondiente según la tabla arriba.

---

**Última revisión:** 17 de octubre de 2025

**Versión:** 1.0

**Estado:** Completo y listo para implementación