

# Documento de Requerimientos del Producto (PRD)

- **Proyecto:** 'Chapelotas'
- **Versión:** 1.1 (Definición de MVP Completa - Actualizada)
- **Fecha:** 28 de junio de 2025

## 1. Visión del Producto

Una aplicación nativa para Android que funciona como una secretaria personal inteligente, cuyo nombre deriva de "Hincha Pelotas" por su insistencia servicial. Utiliza IA para gestionar de forma proactiva la agenda y las tareas del usuario, centralizando información y minimizando la carga administrativa diaria.

## 2. Objetivos Principales

- **Anticipación:** La IA debe anticiparse a las necesidades del usuario, ofreciendo información y acciones relevantes antes de que se soliciten.
- **Centralización:** Unificar la gestión de la agenda del calendario en un único lugar.
- **Automatización:** Reducir la entrada manual de datos mediante la interpretación de lenguaje natural.
- **Flexibilidad:** Construir sobre una arquitectura modular que permita añadir nuevas funcionalidades e integraciones a futuro.

## 3. Público Objetivo

- **Profesionales Ocupados:** Individuos con agendas complejas que necesitan optimizar su tiempo y preparación.
- **Estudiantes y Académicos:** Usuarios que gestionan múltiples materias, plazos de entrega y horarios de estudio.
- **Autónomos y Emprendedores:** Personas que deben gestionar tanto su tiempo personal como el de su negocio de manera eficiente.

## 4. Personalidad y Tono de la IA

- **Nombre del Asistente:** 'Chapelotas'
- **Rasgo Principal: Insistencia Servicial.** La IA debe ser persistente en sus recordatorios para asegurar que el usuario no olvide sus compromisos, pero siempre desde un tono de ayuda.
- **Tono por Defecto (Modo Sarcástico APAGADO): Cercano y Amigable.** Utilizará un lenguaje coloquial y frases que denoten persistencia con amabilidad.
  - *Ejemplo:* "Perdón por insistir, pero recordá que en 20 minutos empieza tu reunión de equipo."
- **Tono Alternativo (Modo Sarcástico ENCENDIDO):** Mantiene la insistencia pero con un humor ácido e irónico.
  - *Ejemplo:* "Mirá, yo te aviso por tercera vez. Si llegás tarde a la reunión, después no digas que no tenías un asistente increíble que intentó salvarte."

## 5. MVP v1.0 - Funcionalidades Prioritarias

### [MVP-CAL-01] Lector de Calendario Robusto (Read-Only)

- La aplicación leerá todos los eventos de los calendarios nativos del dispositivo Android.
- No se permitirá la edición ni creación de eventos en el calendario desde la app en esta versión.

### [MVP-NOTIF-01] Sistema de Notificaciones Persistente

- Las notificaciones deben tener una apariencia similar a las de WhatsApp.
- **Fiabilidad Máxima:** El sistema debe ser persistente, sobrevivir reinicios y cierres de la app, probablemente usando un `Foreground Service` y solicitando permisos para ignorar las optimizaciones de batería.

### [MVP-CORE-01] Resumen de Agenda (Hoy y Mañana)

- **Contenido y Tono:** El resumen debe simular el mensaje de una secretaria personal eficiente, con comentarios inteligentes sobre los eventos y preguntas proactivas.
- **Disparadores (Triggers):**
  1. **Resumen de Hoy:** Automáticamente cada mañana (hora configurable).
  2. **Resumen de Mañana:** Automáticamente cada tarde/noche (hora configurable).
  3. **Bajo Demanda:** Botones en la app para solicitar el resumen de "Hoy" o "Mañana" en cualquier momento.

### [MVP-CORE-02] Alertas Críticas Inteligentes (Llamada Simulada)

- Un recurso de alta prioridad que ocupa toda la pantalla para eventos que no pueden ser olvidados.
- **Lógica de Criticidad (v1.0):** La definición de criticidad será dada por el usuario mediante un intercambio directo con la aplicación. Tras recibir el resumen del día o de mañana, el usuario podrá marcar manualmente uno o más eventos como "meritorios de llamado", activando así la alerta crítica.

### [MVP-CORE-03] Modo Sarcástico

- Un interruptor (switch) en la interfaz para activar el tono sarcástico en los mensajes generados por la IA.

### [UX-01] Flujo de Configuración Inicial

- Al terminar la configuración por primera vez, la aplicación debe dirigir al usuario directamente a la página principal de input para una interacción inmediata.

### [UX-02] Pantalla de Consentimiento de Privacidad

- En el primer inicio, se mostrará una pantalla obligatoria con el mensaje: "Esta app necesita enviar información de tu calendario a servidores externos de IA para poder funcionar." El usuario debe aceptar para continuar; si no, la app se cierra.

## 6. Lógica de Interacción App-IA (El Planificador)

1. **Recopilación:** La app recopila los eventos del día del calendario.
2. **Consulta de Planificación:** La app envía los eventos a la IA para obtener un "plan de comunicación" estructurado (en JSON) que defina qué notificar, cuándo y con qué insistencia.
3. **Certificación del Plan:** La app recibe y valida que el plan de la IA sea lógico y ejecutable.
4. **Ejecución y Scheduling:** La app utiliza sus mecanismos internos ( `AlarmManager` , `WorkManager` ) para programar las notificaciones de forma fiable.
5. **Generación del Mensaje Final:** En el momento de notificar, la app hace una segunda llamada a la IA para generar el texto final, aplicando el "Modo Sarcástico" si está activo.

## 7. Manejo de Errores y Casos Borde

- **Fallo de Conexión con la IA:**

1. **Mensaje de Alerta:** Se notifica al usuario del problema, usando el tono correspondiente (normal o sarcástico).
2. **Plan de Contingencia:** Se muestra un resumen básico pre-formateado y sin IA extraído del calendario.

## 8. Explícitamente Fuera de Alcance para el MVP v1.0

- Módulo de To-Do.
- Creación o edición de eventos en el calendario.
- Interacción por voz.
- Integraciones con terceras apps (Spotify, etc.).
- Sugerencia de eventos críticos por parte de la IA (planificado para v1.1).

## 9. Monetización

- **MVP (v1.0):** No aplica.
- **Visión a Futuro:** Flexibilidad para futura monetización (ej. Freemium).

## 10. Arquitectura y Desarrollo

- **Arquitectura Limpia (Clean Architecture)** para modularidad y escalabilidad.
- Uso de scripts de automatización ( `.bat` , Gradle) durante el desarrollo.