**Plan de Trabajo Detallado - AudioMinutes MVP**

**Día 1: Setup del Proyecto y Backend Base**

**Mañana (3-4 horas)**

**1.1 Inicialización del proyecto**

* Crear repo en GitHub
* Estructura de carpetas según arquitectura
* Setup de entorno virtual Python
* requirements.txt inicial (FastAPI, SQLAlchemy, psycopg2, etc.)

**1.2 Configuración base FastAPI**

* main.py con app básica y health endpoint
* Configuración de variables de entorno (.env)
* Setup de CORS para frontend
* Docker setup básico para development

**1.3 Database setup**

* Models SQLAlchemy (models.py)
* Configuración PostgreSQL connection
* Migrations setup (Alembic)
* Tabla usage\_logs inicial

**Tarde (3-4 horas)**

**1.4 Servicios base**

* services/assembly.py - integración AssemblyAI básica
* services/openai\_service.py - client OpenAI
* services/email\_service.py - Resend integration
* utils.py - helpers (validación email, duración audio, IP geolocation)

**1.5 Testing inicial**

* Test de conexión con APIs externas
* Test de database connection
* Endpoint /health funcionando

**Día 2: Core Backend Logic**

**Mañana (4 horas)**

**2.1 Upload endpoint**

* /upload endpoint con validación de archivos
* Validación de duración de audio
* Sistema de whitelist (variables de entorno)
* Validación de email format

**2.2 Background processing**

* Background tasks setup en FastAPI
* Pipeline completo: Audio → Assembly → OpenAI → Email
* Error handling robusto
* Logging de cada paso del proceso

**Tarde (4 horas)**

**2.3 Cost tracking y database**

* Cálculo de costos AssemblyAI + OpenAI
* Guardado en database con todos los campos
* Geolocation por IP
* Status endpoint para tracking de jobs

**2.4 Email service completar**

* Template de email con acta
* Manejo de errores de envío
* Testing con emails reales

**Día 3: Frontend y Docker**

**Mañana (3-4 horas)**

**3.1 Frontend básico**

* index.html con form de upload
* CSS responsive y clean
* JavaScript para upload con progress
* Validación frontend (email, tipo archivo, tamaño)

**3.2 UX/UI básico**

* Loading states
* Error handling visual
* Success states
* Responsive design mobile

**Tarde (3-4 horas)**

**3.3 Docker y EasyPanel prep**

* Dockerfile para FastAPI
* Dockerfile para Nginx frontend
* docker-compose.yml para desarrollo
* Variables de entorno production-ready

**3.4 Integration testing**

* Test completo del pipeline
* Test con archivos de diferentes duraciones
* Test de whitelist functionality

**Día 4: EasyPanel Deploy y Testing**

**Mañana (3-4 horas)**

**4.1 Deploy en EasyPanel**

* Setup PostgreSQL service
* Deploy FastAPI backend
* Deploy Nginx frontend
* Configurar networking entre services

**4.2 Environment setup**

* Variables de entorno en EasyPanel
* API keys configuradas
* Database migrations en production
* Domain/subdomain setup

**Tarde (3-4 horas)**

**4.3 Testing en producción**

* Test completo end-to-end
* Test con whitelist emails
* Test de límites y validaciones
* Performance testing básico

**4.4 Analytics setup**

* Google Analytics implementado
* Contador interno de archivos procesados
* Dashboard básico para métricas
* Error monitoring setup

**Día 5: Polish y Launch**

**Mañana (3 horas)**

**5.1 Bug fixes y improvements**

* Fix issues encontrados en testing
* Mejoras de UX basadas en pruebas
* Optimización de performance
* Security review básico

**5.2 Documentation**

* README con setup instructions
* API documentation (FastAPI auto-docs)
* Notas de deployment para EasyPanel

**Tarde (2-3 horas)**

**5.3 Soft launch**

* Testing con usuarios reales (tu equipo)
* Monitoring de errores y costos
* Feedback collection setup
* Launch checklist final

**Checklist de Preparación**

**Antes de empezar:**

* Cuentas creadas: AssemblyAI, OpenAI, Resend
* API keys obtenidas y documentadas
* Acceso a Vultr VPS confirmado
* EasyPanel funcionando
* Google Analytics account setup

**Variables de entorno necesarias:**

env

ASSEMBLYAI\_API\_KEY=

OPENAI\_API\_KEY=

RESEND\_API\_KEY=

DATABASE\_URL=

WHITELIST\_EMAILS=email1@domain.com,email2@domain.com

GA\_TRACKING\_ID=

**Archivos de audio para testing:**

* Archivo de 5 minutos (caso normal)
* Archivo de 15 minutos (límite normal)
* Archivo de 30 minutos (caso whitelist)
* Archivo con ruido (edge case)

**Contingencias**

**Si hay delays:**

* **Día 3 → 4:** Frontend simplificado (sin progress bar)
* **Día 4 → 5:** Deploy en localhost, production después
* **APIs lentas:** Mocks para development, real APIs solo para testing

**Criterio de "MVP listo":**

* ✅ Usuario puede subir audio + email
* ✅ Recibe acta por email
* ✅ Tracking de costos funciona
* ✅ Deployed y accesible públicamente