



Sistema de Monitoreo para Android

Hola equipo,

Les escribo para proponer el proyecto que vamos a desarrollar para los cursos de **Sistemas Operativos** y **Base de Datos**. La idea es hacer una sola aplicación que sirva para ambos, así no nos partimos la cabeza con dos proyectos separados. Acá les explico de qué va todo, en español bien sencillo.



¿Qué Vamos a Hacer?

Vamos a crear un **Sistema de Monitoreo y Análisis Histórico para Dispositivos Android**. Básicamente, es un programa que se instala en la laptop y que:

1. **Se conecta al celular Android** (con cable USB) y **lee en tiempo real** todo lo que pasa adentro: cuánta CPU se está usando, cuánta memoria, la batería, qué aplicaciones están abiertas, etc. → Esto es puro **Sistemas Operativos**.
2. **Guarda toda esa información** en una **base de datos** en nuestra computadora. → Esto es pura **Base de Datos**.
3. **Muestra toda la info** en una **página web** bien fancy que podemos abrir en el navegador. Ahí veremos gráficos que se actualizan al momento y también podremos ver reportes del pasado (ej: "¿Qué app me chupó toda la batería ayer?").



¿Por qué Este Proyecto Sirve para los Dos Cursos?

- Para **Sistemas Operativos**: Porque estamos interactuando directamente con el Android (que funciona con Linux por dentro). Vamos a usar comandos para "preguntarle" al sistema operativo del celular qué está pasando con sus recursos internos (procesos, memoria, CPU). Es como tener el Administrador de Tareas de Windows pero para el celular y hecho por nosotros.
- Para **Base de Datos**: Porque toda la información que recolectemos la vamos a guardar de forma ordenada en tablas (SQL). Luego, vamos a hacer consultas super interesantes a esa base de datos para analizar el comportamiento del celular en el tiempo. Vamos a aplicar todo lo de diseño de bases de datos, relaciones y queries.



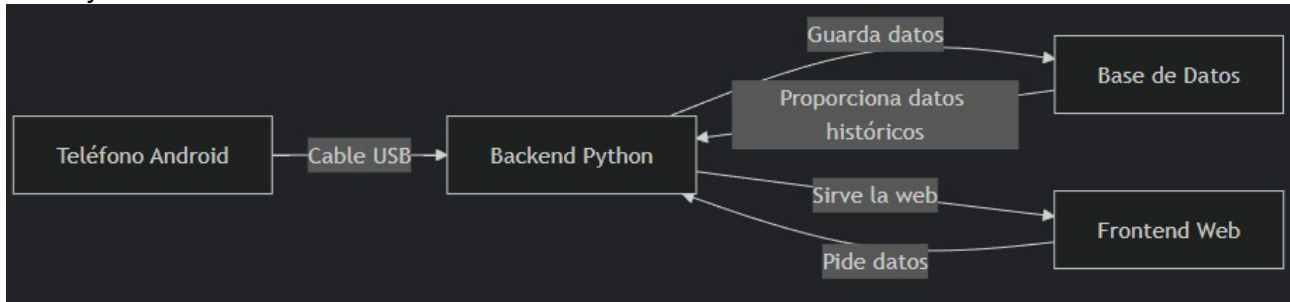
¿Cómo Lo Vamos a Armar? (Arquitectura)

Imagínense el proyecto dividido en 3 partes principales:

1. **El Backend (Python)**: Este es el cerebro. Es un programita en Python que se va a encargar de:
 - Hablar con el celular (usando una herramienta que se llama adb).
 - Pedirle los datos del CPU, memoria, etc.
 - Guardar esos datos en la base de datos.
 - Crear una mini aplicación web (con una librería llamada Flask) que nos sirva para mostrar los datos.
2. **La Base de Datos (SQLite)**: Acá es donde se guarda toda la historia. Vamos a crear tablas para guardar:
 - **Métricas**: Uso de CPU, memoria, batería, etc. Cada dato con su fecha y hora exacta.
 - **Procesos**: Lista de apps abiertas y cuántos recursos usa cada una.
3. **El Frontend (Página Web)**: La cara bonita del proyecto. Una página web (con HTML, CSS y JavaScript) que se vea moderna y que:
 - Muestre gráficos en tiempo real (usando una librería como Chart.js).

- Tenga botones y filtros para ver datos pasados.
- Se comunique con el backend para pedirle la información.

Así fluye la info:



🔧 ¿Qué Necesitamos Instalar?

- **Lenguaje:** Python 3 en adelante puede ser 3.11, etc. (el corazón de todo).
- **Librerías de Python:** Flask (para el servidor web), psutil (para recursos de la laptop opcional), sqlite3 (para la base de datos).
- **Herramienta ADB:** Las **Android SDK Platform-Tools** (es gratis, de Google). Esto es lo que nos deja hablar con el celular.
- **Base de Datos:** PostgreSQL (Porque ya lo tenemos).
- **Frontend:** HTML, CSS, JavaScript puro. Y la librería Chart.js para los gráficos.
- **Control de Versiones:** Git (obvio, para no pisarnos el código).



Plan de Juego y Entregas

Tenemos dos fechas clave:

1. Hito del 50% (Para mediados de ciclo – 8 de Octubre): EL MONITOR EN TIEMPO REAL

- **Objetivo:** Tener la parte que habla con el celular y muestra los datos EN VIVO en una página web.
- **Tareas:**
 - Configurar ADB en las laptops y conectar un celular Android (activar "Opciones de Desarrollador y el depurador").
 - Hacer un script en Python que ejecute comandos ADB (ej: adb shell top -n 1) para sacar datos.
 - Usar Flask para crear una API que diga "en este momento, la CPU está al 45%".
 - Hacer una página web simple que pida esos datos a la API y los muestre en números grandes.

2. Hito del 100% (Para finales de ciclo – 3 de Diciembre): EL ANÁLISIS HISTÓRICO

- **Objetivo:** Agregar la base de datos y los reportes del pasado.
- **Tareas:**
 - Diseñar las tablas en la base de datos (Dispositivos, Métricas, Procesos).
 - Modificar el backend para que guarde TODO lo que recolecta en la base de datos.
 - Crear nuevas páginas en la web para ver gráficos de líneas de los últimos días.
 - Escribir consultas SQL potentes para hacer reportes. Ej: "Top 5 apps que más batería consumen".



¿Quién Hace Qué?

Podemos dividirnos el trabajo así, dependiendo de en qué sean mejores:

- **Especialista en Backend/Python (1-2 personas):** Encargados de la lógica de comunicación con ADB, la API con Flask y insertar datos en la base de datos.
- **Especialista en Base de Datos (1-2 persona):** Diseña las tablas, escribe las consultas SQL más complejas para los reportes y asegura que los datos se guarden bien.

- **Especialista en Frontend (1-2 personas):** Se encarga de que la página web se vea profesional, que los gráficos estén chéveres y que sea fácil de usar.



¿Por qué Este Proyecto Está Buenísimo?

- **Es real:** No es un ejercicio de libro. Es una herramienta de verdad.
- **Aprendemos un montón:** Le metemos mano a Python, bases de datos, APIs, frontend, y hasta cómo funciona Android por dentro.
- **Podemos mostrarlo en el CV:** Queda mucho mejor decir "Hice un sistema de monitoreo".
- **Le podemos agregar cosas:** Si nos da el tiempo, luego le podemos agregar notificaciones, una app para el celular, etc.

Bueno gente, ese es el plan. Estoy seguro de que podemos hacerlo y sacar una nota alta en el curso.

Ahora **confirмен** que les parece bien y **qué parte les gustaría desarrollar**. Armemos un grupo de Discord para coordinar.

GAAAAAAA

Marco HB