# UNIVERSIDAD NACIONAL DE INGENIER´IA

**FACULTAD DE CIENCIAS ESCUELA PROFESIONAL DE CIENCIA DE LA**

**COMPUTACIO´N**

**Curso: Sistemas Operativos**

# Proyecto Formativo

Sistema de Monitoreo y An´alisis Histo´rico para Dispositivos Android

**Co´digo del curso: CC222**

**Alumno: Huamani Bonifacio, Marco Antonio Alumno: Paul FLores, Enrique Alejandro Alumno: Navarro Ruiz, Angel**

**Alumno: Alumno: Alumno:**

**Octubre, 2024**

**´Indice**

1. [Resumen](#_bookmark0) 2
2. [Introduccio´n](#_bookmark1) 2
   1. [Motivaci´on](#_bookmark2) 2
   2. [Objetivos](#_bookmark3) 2
      1. [Descripcio´n del Problema](#_bookmark4) 2
   3. [Estructura del Proyecto](#_bookmark5) 2
3. [Estado del Arte](#_bookmark6) 2
   1. [Trabajos Relacionados](#_bookmark7) 2
   2. [Conceptos Te´oricos](#_bookmark8) 2
      1. [Herramientas Utilizadas](#_bookmark9) 3
   3. [Conclusiones](#_bookmark10) 3
4. [Desarrollo](#_bookmark11) 3
   1. [Descripcio´n del Caso de Estudio](#_bookmark12) 3
      1. [Problem´atica](#_bookmark13) 3
      2. [Objetivo del Caso de Estudio](#_bookmark14) 3
      3. [Evaluacio´n de la Soluci´on](#_bookmark15) 3
   2. [Metodolog´ıa de Desarrollo de la Solucio´n](#_bookmark16) 3
      1. [Especificacio´n de Requerimientos](#_bookmark17) 3
      2. [Herramientas](#_bookmark18) 4
   3. [Disen˜o de la Arquitectura de Desarrollo](#_bookmark19) 4
      1. [Integraci´on e Interfaces](#_bookmark20) 4
   4. [Mo´dulos](#_bookmark21) 4
      1. [Disen˜o de Mo´dulos](#_bookmark22) 4
5. [Resultados](#_bookmark23) 5
   1. [Pruebas y Ana´lisis](#_bookmark24) 5
      1. [Logros Alcanzados](#_bookmark25) 5
6. [Conclusiones y Trabajo Futuro](#_bookmark26) 6
   1. [Conclusiones](#_bookmark27) 6
   2. [Trabajo Futuro](#_bookmark28) 6
7. [Bibliograf´ıa](#_bookmark29) 6

## Resumen

El presente proyecto consiste en el disen˜o e implementaci´on de un **Sistema de Moni- toreo para Dispositivos Android**, que permite la recoleccio´n y visualizaci´on de m´etri- cas de rendimiento del sistema en tiempo real. La solucio´n actual integra un backend en Python y un frontend web interactivo, con planificacio´n para incorporar almacenamiento histo´rico en futuras iteraciones.

## Introduccio´n

### Motivacio´n

Los dispositivos m´oviles se han convertido en herramientas esenciales en la vida coti- diana, y comprender su comportamiento interno es clave para optimizar su rendimiento. Este proyecto nace de la necesidad de contar con una solucio´n accesible y personalizada para monitorear el uso de CPU, memoria, bater´ıa y procesos en tiempo real.

### Objetivos

 Desarrollar un sistema que capture m´etricas de rendimiento de dispositivos Android mediante el uso de ADB.

Implementar una interfaz web que visualice los datos en tiempo real.

 Planificar la integraci´on futura de base de datos para almacenamiento histo´rico.

### Descripci´on del Problema

Actualmente, los usuarios y desarrolladores carecen de herramientas gratuitas y acce- sibles que permitan un monitoreo detallado en tiempo real de dispositivos Android. Este proyecto busca llenar ese vac´ıo mediante una soluci´on integral y educativa.

### Estructura del Proyecto

El informe se organiza de la siguiente manera: Estado del Arte, Desarrollo, Resultados, Conclusiones y Trabajo Futuro, Bibliograf´ıa y Anexos.

## Estado del Arte

### Trabajos Relacionados

Existen herramientas como *Android Studio Profiler* y aplicaciones de terceros como *CPU Monitor*, pero muchas requieren configuracio´n compleja o no ofrecen una interfweb personalizada.

### Conceptos Te´oricos

 **Android Debug Bridge (ADB):** Herramienta de l´ınea de comandos que permite la comunicacio´n con un dispositivo Android.

**Sistemas Operativos Linux:** Android esta´ basado en el kernel de Linux, lo que permite el acceso a m´etricas del sistema mediante comandos shell.

 **Arquitectura Cliente-Servidor:** Implementada con Flask para el backend y un frontend web.

### Herramientas Utilizadas

 Flask: Framework web para Python.

Chart.js: Librer´ıa JavaScript para gra´ficos interactivos.

 ADB (Android Debug Bridge): Para comunicacio´n con dispositivos Android.

### Conclusiones

El an´alisis del estado del arte confirma la viabilidad del proyecto para ofrecer una solucio´n de monitoreo en tiempo real utilizando tecnolog´ıas accesibles.

## Desarrollo

### Descripci´on del Caso de Estudio

### Problem´atica

Falta de una herramienta unificada para el monitoreo en tiempo real de dispositivos Android con interfaz web accesible.

### Objetivo del Caso de Estudio

Implementar un sistema que permita visualizar en tiempo real el rendimiento de un dispositivo Android mediante una interfaz web.

### Evaluacio´n de la Solucio´n

La solucio´n sera´ evaluada mediante pruebas de funcionalidad, precisi´on en la recolec- cio´n de datos y usabilidad de la interfaz web.

### Metodolog´ıa de Desarrollo de la Solucio´n

* + 1. **Especificaci´on de Requerimientos Requerimientos Funcionales (Implementados):**

RF01: El sistema debe conectarse a un dispositivo Android mediante ADB.

RF02: Debe recolectar m´etricas de CPU, memoria, bater´ıa y procesos en tiempo real.

 RF03: Debe mostrar los datos en una interfaz web con gr´aficos actualizados en tiempo real.

### Requerimientos No Funcionales:

RNF01: El sistema debe ser responsive y accesible desde cualquier navegador mo- derno.

 RNF02: Debe mantener estabilidad en la conexio´n ADB durante el monitoreo.

### Requerimientos Futuros:

RFF01: Almacenamiento de datos histo´ricos en base de datos.

 RFF02: Generaci´on de reportes histo´ricos y ana´lisis de tendencias.

### Herramientas

 Python 3.11

Flask para el backend

ADB (Android Debug Bridge)

HTML, CSS, JavaScript para el frontend

 Chart.js para visualizaciones

### Disen˜o de la Arquitectura de Desarrollo

La arquitectura actual se compone de dos componentes principales:

1. **Backend (Python + Flask):** Gestiona la comunicacio´n con el dispositivo Android mediante ADB y expone APIs REST para el frontend.
2. **Frontend (HTML + CSS + JS):** Muestra los datos en forma de gr´aficos y tablas en tiempo actual.

### Integracio´n e Interfaces

 El backend se comunica con el dispositivo Android mediante comandos ADB.

Los datos se env´ıan al frontend mediante APIs JSON en tiempo real.

 Los datos se mantienen en memoria durante la sesio´n activa.

### M´odulos

### Disen˜o de Mo´dulos

 **M´odulo 1: Recolector de Datos**

Script en Python que ejecuta comandos ADB y parsea la salida en tiempo real.

### M´odulo 2: Servidor Web (Flask)

Expone endpoints REST para proporcionar datos en tiempo actual al frontend.

### M´odulo 3: Interfaz de Usuario

Pa´ginas web con gra´ficos en tiempo real y visualizacio´n de m´etricas actuales.

arquitectura\_actual.png

Figura 1: Arquitectura actual del sistema (sin base de datos)

## Resultados

### Pruebas y An´alisis

Se realizaron pruebas exitosas de:

 Conectividad ADB con dispositivos Android

 Precisio´n en la recolecci´on de datos del sistema  Actualizacio´n en tiempo real de la interfaz web  Estabilidad del servidor Flask

Los resultados demuestran que el sistema es capaz de recolectar y visualizar datos en

tiempo real de manera estable y precisa.

### Logros Alcanzados

 Comunicacio´n estable con dispositivos Android via ADB web responsive y funcional  Gra´ficos en tiempo actual usando Chart.js

 Arquitectura escalable para futuras mejoras

monitoreo\_tiempo\_real.png

Figura 2: Pantalla de monitoreo en tiempo real funcionando

## Conclusiones y Trabajo Futuro

### Conclusiones

El proyecto demuestra que es posible desarrollar una herramienta de monitoreo en tiempo real para Android utilizando tecnolog´ıas accesibles. La integracio´n de ADB con Flask y una interfaz web moderna result´o efectiva para cumplir con los objetivos de monitoreo en tiempo actual.

### Trabajo Futuro

 **Trabajo 1:** Implementaci´on de base de datos PostgreSQL para almacenamiento histo´rico

**Trabajo 2:** Disen˜o de esquema de base de datos para m´etricas y procesos

**Trabajo 3:** Desarrollo de funcionalidades de an´alisis histo´rico

 **Trabajo 4:** Implementacio´n de consultas SQL para reportes avanzados

## Bibliograf´ıa

1. Documentaci´on oficial de Android Debug Bridge (ADB)
2. Flask Web Development Documentation
3. Chart.js Documentation
4. Tanenbaum, A. S. (2015). *Sistemas Operativos Modernos*

metricas\_actuales.png

Figura 3: Ejemplo de visualizacio´n de m´etricas actuales

## Anexos

### Estructura de Datos Actual

 **Datos en Tiempo Real:** CPU, memoria, bater´ıa, procesos activos

**Formato:** JSON via API REST

 **Almacenamiento:** Memoria temporal durante sesio´n

### Plan de Base de Datos Futuro

 **Tabla Dispositivos:** id, nombre, modelo, fecha registro

**Tabla M´etricas:** id, dispositivo id, cpu, memoria, bater´ıa, timestamp

**Tabla Procesos:** id, dispositivo id, nombre, pid, uso cpu, timestamp