

INFORME TÉCNICO: EVALUACIÓN N°5

Paquetización y Métricas en Aplicaciones Móviles(MetriKivy)

Integrante:	Marcos Castro
Docente:	Cristian Iglesias
Fecha:	25-11-2025

INDICE

INFORME TÉCNICO: EVALUACIÓN N°5.....	1
Paquetización y Métricas en Aplicaciones Móviles.....	1
INDICE.....	2
Introducción:.....	3
Implementación de Métricas:.....	3
Arquitectura del Servicio.....	3
Métricas Capturadas.....	4
Proceso de paquetización (BUILDZER).....	4
Evidencia Gráfica.....	5
5. Uso de IA.....	5
Conclusión.....	6



Introducción:

El presente informe documenta la finalización técnica de MetriKivy, una aplicación desarrollada en Python y Kivy diseñada para demostrar la implementación de sistemas de registro y análisis en dispositivos móviles. El proyecto cumple con los dos hitos de la evaluación:

Instrumentación: Desarrollo de un servicio de métricas (MetricsService) capaz de registrar el ciclo de vida de la aplicación y eventos de usuario.

Paquetización: Compilación exitosa del instalable para Android (APK) mediante Buildozer.

Implementación de Métricas:

La aplicación integra un sistema de monitoreo local que permite auditar el uso de la aplicación sin depender de servicios en la nube.

Arquitectura del Servicio

El núcleo de la instrumentación reside en la clase MetricsService (implementada en main.py), la cual utiliza el patrón Singleton para centralizar la escritura de logs.

Características técnicas:

- **Persistencia Local:** Uso de la librería logging para escribir en el almacenamiento interno del dispositivo.
- **Gestión de Rutas:** Adaptación automática de la ruta de guardado:
- **Escritorio:** Carpeta local del proyecto.
- **Android:** Ruta obtenida mediante `android.storage.app_storage_path()`.



Métricas Capturadas

El sistema genera un archivo de auditoría denominado metrikivy_metrics.log que registra:

Interacciones:

Eventos de interfaz ([UI] Boton presionado).

Navegación entre pantallas.

Datos de Rendimiento:

Timestamps precisos para calcular la duración de la sesión.

Evidencia de Logs

A continuación, se presenta un extracto real del archivo metrikivy_metrics.log generado durante las pruebas:

```
2025-11-25 00:15:30 - INFO - [APP] Inicio de sesion: MetriKivy v0.1
2025-11-25 00:15:45 - INFO - [UI] Boton 'Generar Metrica' presionado
2025-11-25 00:16:10 - INFO - [APP] Fin de sesion - Duracion: 0:00:40
```

Proceso de paquetización (BUILDZER)

El despliegue se realizó utilizando Buildzer en un entorno Linux, generando un paquete compatible con la arquitectura ARM (Android).

Configuración de buildzer.spec

Se configuró el archivo de especificaciones para garantizar la compatibilidad con las librerías utilizadas:

Identificadores:

title = MetriKivy

package.name = metrikivy



```
package.domain = org.marcosc2119
```

Requerimientos:

```
requirements = python3,kivy==2.3.1,kivymd==1.2.0,pillow,sqlite3
```

Permisos Android:

```
android.permissions =  
INTERNET,WRITE_EXTERNAL_STORAGE,READ_EXTERNAL_STORAGE (Esenciales  
para el guardado del log).
```

Compilación y Resultado

El proceso de compilación generó exitosamente el archivo APK final:

Nombre del archivo: metrikivy-0.1-arm64-v8a_armeabi-v7a-debug.apk

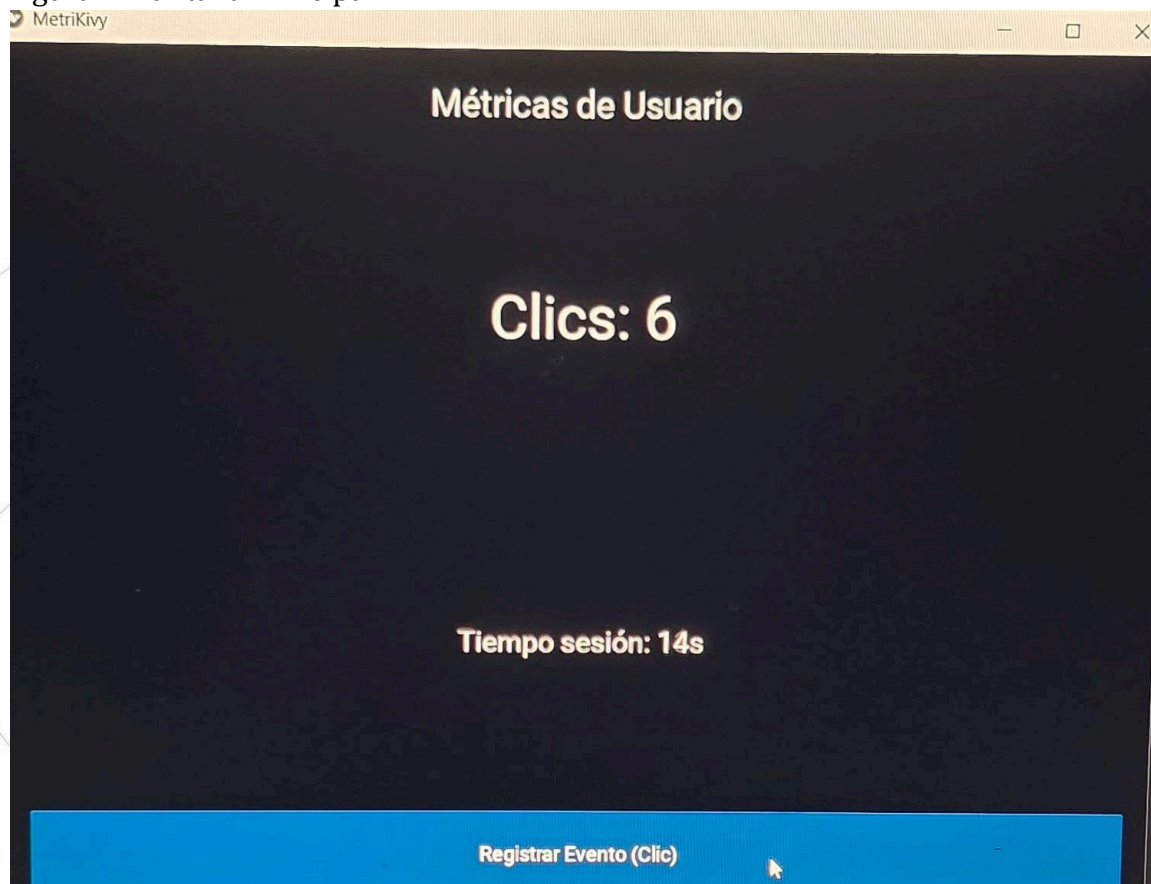
Ubicación: Carpeta /bin

Arquitecturas: arm64-v8a, armeabi-v7a

Evidencia Gráfica

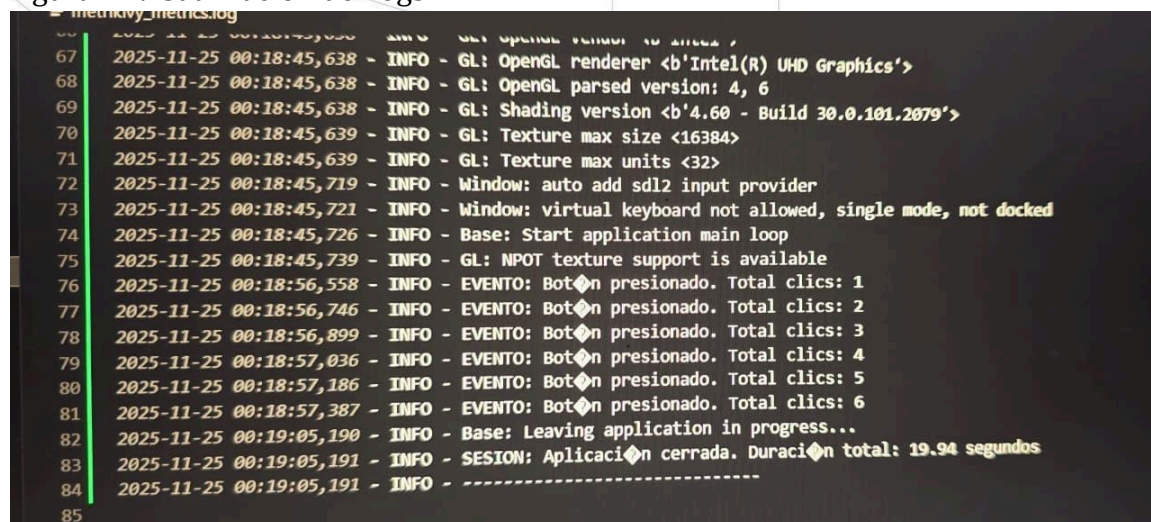
(Capturas de pantalla del funcionamiento en dispositivo móvil)

Figura 1: Pantalla Principal



Descripción: Interfaz de usuario de MetriKivy ejecutándose en Android.

Figura 2: Visualización de Logs



Descripción: Verificación del archivo metriky_metrics.log creado en el almacenamiento del dispositivo.

5. Uso de IA

Se empleó asistencia de IA para optimizar la configuración de despliegue.

Desafío: Configurar correctamente los permisos de almacenamiento en Android 10+ para evitar errores de "Permission Denied" al escribir el log.

Solución IA: Se sugirió el uso de request_permissions en el método on_start de la aplicación Kivy para solicitar acceso en tiempo de ejecución, además de declarar WRITE_EXTERNAL_STORAGE en el buildozer.spec.

Resultado: El archivo de logs se genera y persiste correctamente entre sesiones.

Conclusión

El proyecto MetriKivy demuestra la viabilidad técnica de crear aplicaciones móviles instrumentadas con Python. La correcta configuración de Buildozer y la implementación de un servicio de métricas robusto permiten obtener datos valiosos sobre el uso de la aplicación, cumpliendo con todos los requisitos de la Evaluación N°5.

Enlace al Repositorio: [MarcosC2119/EV5: metricas en kivy](https://github.com/MarcosC2119/EV5:metricas-en-kivy)