

Informe Evaluación N°5: Paquetización de Aplicación Kivy y Métricas

Camila Aceitón
Profesor: Cristian Iglesia

El proyecto "MetriKivy" consiste en una aplicación móvil nativa para Android desarrollada en Python utilizando el framework Kivy. La aplicación tiene como objetivo principal demostrar el ciclo completo de desarrollo, desde la codificación de la interfaz hasta la generación del instalable (APK).

Funcionalmente, la app permite al usuario interactuar con un botón contador y mide automáticamente el tiempo de uso de la sesión, registrando estos datos mediante un sistema de logs interno para su posterior análisis.

Proceso de Paquetización (Buildozer)

Para generar el archivo APK , se utilizó la herramienta Buildozer ejecutada sobre un entorno Linux (WSL con Ubuntu 24.04). Los pasos principales fueron:

- **Preparación del Entorno:** Se instalaron las dependencias del sistema (openjdk-17-jdk, git, zip, unzip, python3-pip).
- **Configuración:** Se inicializó el proyecto con buildozer init y se editó el archivo buildozer.spec para definir el nombre del paquete (org.camila.metrikivy) y los requisitos (python3,kivy).
- **Compilación:** Se ejecutó el comando buildozer -v android debug.

Solución de Conflictos: Durante el proceso, fue necesario realizar ajustes técnicos avanzados para solucionar errores de compatibilidad con versiones modernas de Ubuntu:

- Instalación de autoconf, automake y libtool para solucionar errores de autoreconf.
- Instalación de libffi-dev y libssl-dev para la compilación de _ctypes.
- Instalación específica de Cython < 3 para compatibilidad con pyjnius.

Se implementó un sistema de instrumentación basado en la librería estándar logging de Python.

Configuración: Los logs se guardan en el almacenamiento interno de la aplicación (app_storage_path en Android) o en la carpeta local (Windows).

- **Métrica 1 (Eventos):** Registro de interacción cada vez que el usuario presiona el botón "Registrar Evento".
- **Métrica 2 (Sesión):** Cálculo del tiempo total de uso (en segundos) capturado en el evento on_stop de la aplicación.

Evidencias y resultados

Imagen 1: Aplicación funcionando en Windows (Interfaz Gráfica)

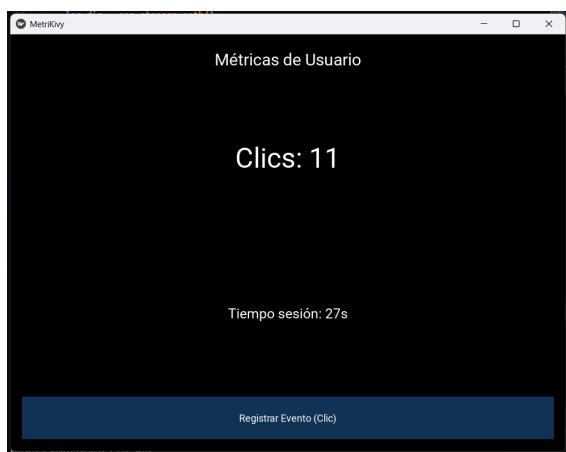
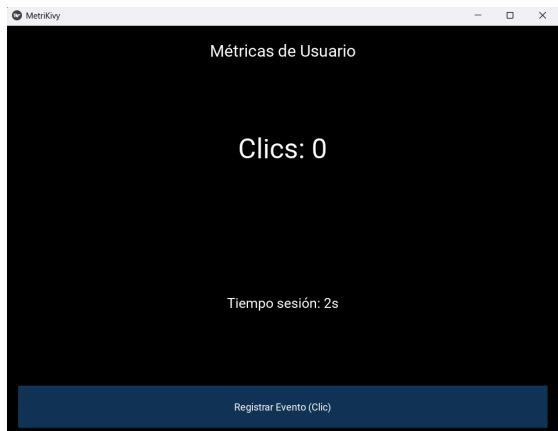
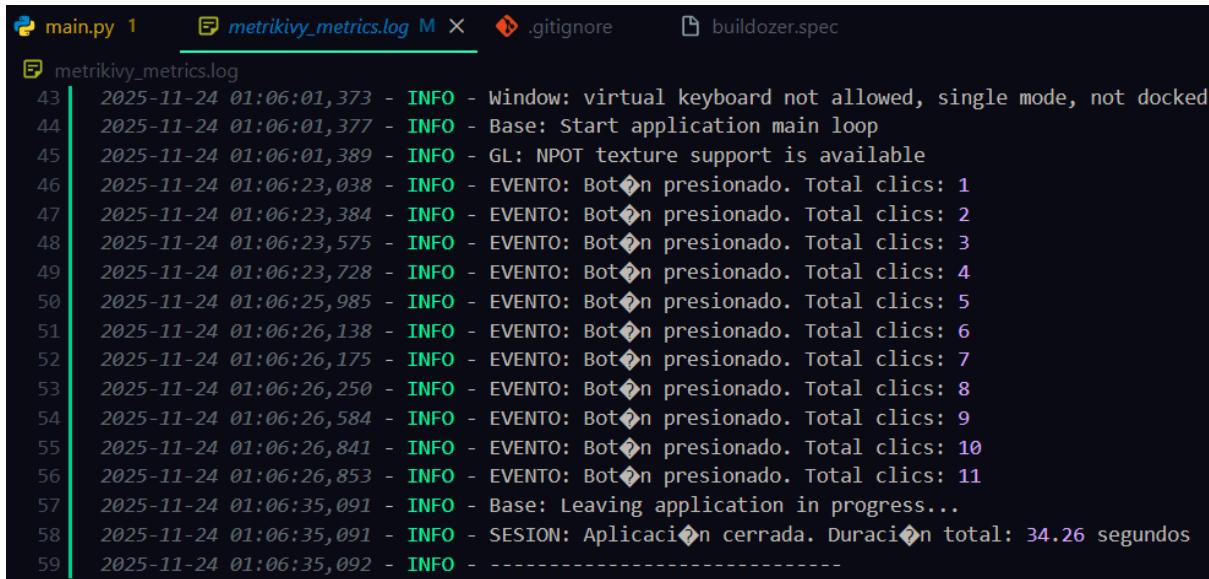


Imagen 2 : información de la terminal del proyecto en visual studio

```
PS C:\Users\camil\MetriKivy> & C:/Users/camil/AppData/Local/Programs/Python/Python313/python.exe c:/Users/camil/MetriKivy/main.py
[INFO] [EVENTO] Botón presionado. Total clics: 7
Métrica registrada: Clic #7
[INFO] [EVENTO] Botón presionado. Total clics: 8
Métrica registrada: Clic #8
[INFO] [EVENTO] Botón presionado. Total clics: 9
Métrica registrada: Clic #9
[INFO] [EVENTO] Botón presionado. Total clics: 10
Métrica registrada: Clic #10
[INFO] [EVENTO] Botón presionado. Total clics: 11
Métrica registrada: Clic #11
[INFO] [Base] Leaving application in progress...
[INFO] [SESION] Aplicación cerrada. Duración total: 34.26 segundos
[INFO]
```

Imagen 3: Evidencia de Logs Generados



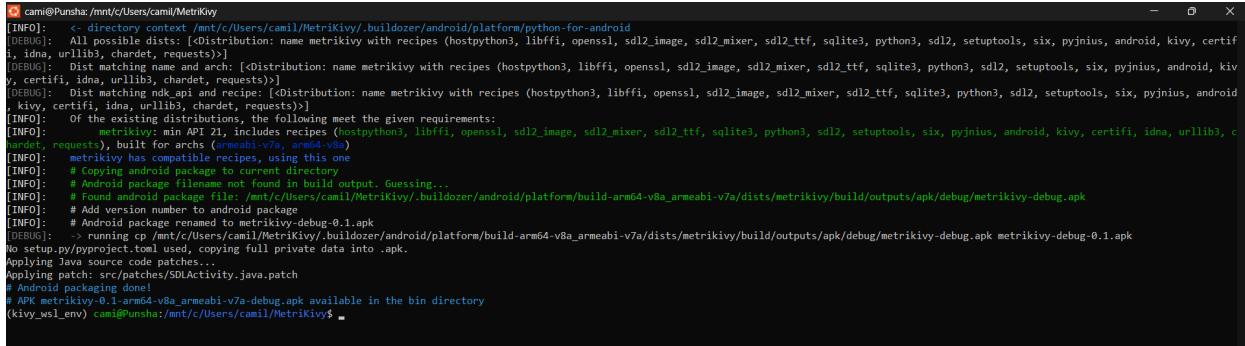
```

main.py 1 metrikiivy_metrics.log M .gitignore buildozer.spec

metrikiivy_metrics.log
43 2025-11-24 01:06:01,373 - INFO - Window: virtual keyboard not allowed, single mode, not docked
44 2025-11-24 01:06:01,377 - INFO - Base: Start application main loop
45 2025-11-24 01:06:01,389 - INFO - GL: NPOT texture support is available
46 2025-11-24 01:06:23,038 - INFO - EVENTO: Botón presionado. Total clics: 1
47 2025-11-24 01:06:23,384 - INFO - EVENTO: Botón presionado. Total clics: 2
48 2025-11-24 01:06:23,575 - INFO - EVENTO: Botón presionado. Total clics: 3
49 2025-11-24 01:06:23,728 - INFO - EVENTO: Botón presionado. Total clics: 4
50 2025-11-24 01:06:25,985 - INFO - EVENTO: Botón presionado. Total clics: 5
51 2025-11-24 01:06:26,138 - INFO - EVENTO: Botón presionado. Total clics: 6
52 2025-11-24 01:06:26,175 - INFO - EVENTO: Botón presionado. Total clics: 7
53 2025-11-24 01:06:26,250 - INFO - EVENTO: Botón presionado. Total clics: 8
54 2025-11-24 01:06:26,584 - INFO - EVENTO: Botón presionado. Total clics: 9
55 2025-11-24 01:06:26,841 - INFO - EVENTO: Botón presionado. Total clics: 10
56 2025-11-24 01:06:26,853 - INFO - EVENTO: Botón presionado. Total clics: 11
57 2025-11-24 01:06:35,091 - INFO - Base: Leaving application in progress...
58 2025-11-24 01:06:35,091 - INFO - SESIÓN: Aplicación cerrada. Duración total: 34.26 segundos
59 2025-11-24 01:06:35,092 - INFO -

```

Imagen 4: Evidencia de Paquetización Exitosa

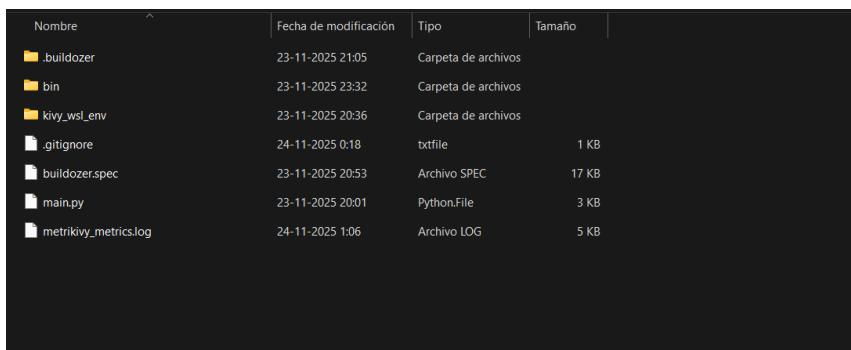


```

cam@Punsha:/mnt/c/Users/camil/MetriKivy
[INFO]: All possible dists: [<Distribution: name metrikiivy with recipes (hostpython3, libffi, openssl, sdl2_image, sdl2_mixer, sdl2_ttf, sqlite3, python3, sdl2, setuptools, six, pyjnius, android, kivy, certifi, idna, urllib3, chardet, requests)>]
[DEBUG]: Dist matching name and arch: [<Distribution: name metrikiivy with recipes (hostpython3, libffi, openssl, sdl2_image, sdl2_mixer, sdl2_ttf, sqlite3, python3, sdl2, setuptools, six, pyjnius, android, kivy, certifi, idna, urllib3, chardet, requests)>]
[DEBUG]: Dist matching ndk api and recipe: [<Distribution: name metrikiivy with recipes (hostpython3, libffi, openssl, sdl2_image, sdl2_mixer, sdl2_ttf, sqlite3, python3, sdl2, setuptools, six, pyjnius, android, kivy, certifi, idna, urllib3, chardet, requests)>]
[INFO]: Of the existing distributions, the following meet the given requirements:
[INFO]:   metrikiivy: min API 21, includes recipes (hostpython3, libffi, openssl, sdl2_image, sdl2_mixer, sdl2_ttf, sqlite3, python3, sdl2, setuptools, six, pyjnius, android, kivy, certifi, idna, urllib3, chardet, requests), built for archs (armeabi-v7a, arm64-v8a)
[INFO]:   metrikiivy has compatible recipes, using this one
[INFO]: # Copying android package to current directory
[INFO]: # Android package filename not found in build output. Guessing...
[INFO]: # Found android package file in /mnt/c/Users/camil/MetriKivy/.buildozer/android/platform/build-arm64-v8a_armeabi-v7a/dists/metrikiivy/build/outputs/apk/debug/metrikiivy-debug.apk
[INFO]: # Add version number to android package
[INFO]: # Version 0.1 added to android package
[INFO]: > running cp /mnt/c/Users/camil/MetriKivy/.buildozer/android/platform/build-arm64-v8a_armeabi-v7a/dists/metrikiivy/build/outputs/apk/debug/metrikiivy-debug.apk metrikiivy-debug-0.1.apk
[DEBUG]: > running cp /mnt/c/Users/camil/MetriKivy/.buildozer/android/platform/build-arm64-v8a_armeabi-v7a/dists/metrikiivy/build/outputs/apk/debug/metrikiivy-debug.apk metrikiivy-debug-0.1.apk
No setup.py/pyproject.toml used, copying full private data into .apk.
Applying Java source code patches...
Applying patch: src/patches/SDLActivity.java.patch
# Android packaging done!
# APK metrikiivy-0.1-arm64-v8a_armeabi-v7a-debug.apk available in the bin directory
(kivy_wsl_env) cam@Punsha:/mnt/c/Users/camil/MetriKivy$ 

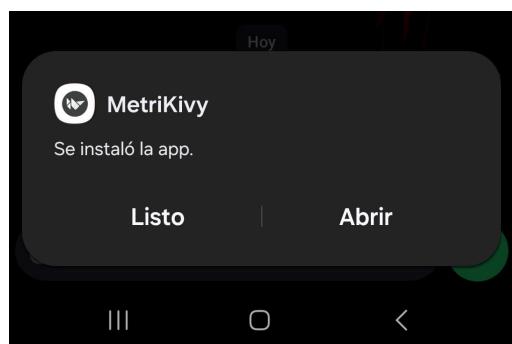
```

Imagen 5: Archivo APK Generado



Nombre	Fecha de modificación	Tipo	Tamaño
.buildozer	23-11-2025 21:05	Carpeta de archivos	
bin	23-11-2025 23:32	Carpeta de archivos	
kivy_wsl_env	23-11-2025 20:36	Carpeta de archivos	
.gitignore	24-11-2025 0:18	txtfile	1 KB
buildozer.spec	23-11-2025 20:53	Archivo SPEC	17 KB
main.py	23-11-2025 20:01	Python.File	3 KB
metrikiivy_metrics.log	24-11-2025 1:06	Archivo LOG	5 KB

Imagen 6 : apk funcionando en dispositivo android





Herramienta utilizada: Gemini (Modelo de Lenguaje Grande). Propósito: Asistencia en codificación, configuración de entorno Linux y resolución de errores de compilación cruzada.

A continuación, se detallan los prompts principales utilizados durante el desarrollo y los ajustes técnicos realizados tras la respuesta de la IA.

1. Generación de Código y Lógica de Negocio

Prompt Utilizado:

"Actúa como un experto en desarrollo móvil con Python. Necesito crear una aplicación Kivy desde cero llamada 'MetriKivy' para una evaluación académica. Requisitos técnicos:

Interfaz simple con un contador y un temporizador.

Sistema de métricas: Debe registrar en un archivo .log cada vez que se presiona un botón y calcular la duración total de la sesión al cerrar la app.

El código debe manejar las rutas de archivos de manera dinámica para funcionar tanto en Windows (local) como en Android (almacenamiento interno). Proporcioname el código main.py listo para usar."

Respuesta y Ajuste: La IA generó el código base correctamente. Sin embargo, se detectó que Kivy interceptaba los mensajes de log y no creaba el archivo físico en disco.

Ajuste realizado: Se modificó la configuración del logging para añadir manualmente un FileHandler al logger raíz, forzando la escritura del archivo metrikivy_metrics.log.

2. Configuración del Entorno de Compilación (WSL/Ubuntu)

Prompt Utilizado:

"Estoy configurando el entorno para Buildozer en Ubuntu 24.04 (WSL) siguiendo una guía antigua. Al intentar instalar las dependencias, obtengo el error E: Unable to locate package libncurses5. Además, al ejecutar buildozer init, necesito saber qué campos específicos del buildozer.spec son críticos para evitar errores de empaquetado en un proyecto básico."

Respuesta y Ajuste: La IA identificó que la versión de la librería en Ubuntu "Noble" (24.04) ha cambiado.

Ajuste realizado: Se reemplazó la instalación de libncurses5 por libncurses6 y libncurses-dev. Se configuraron los campos package.domain y requirements en el archivo .spec.

3. Resolución de Errores de Compilación (Depuración)

Prompt Utilizado:

"Al intentar compilar el APK con buildozer -v android debug, el proceso falla con múltiples errores en cadena. Primero aparece exec: autoreconf: not found. Tras solucionarlo, falla el módulo _ctypes de Python. Finalmente, Cython arroja un error de sintaxis: undeclared name not builtin: long en el módulo jnius. Necesito una solución para reparar el entorno de compilación."

Respuesta y Ajuste: La IA diagnosticó tres problemas distintos de dependencias y versiones.

Ajuste 1: Instalación de herramientas de sistema faltantes: sudo apt install autoconf automake libtool.

Ajuste 2: Instalación de cabeceras de desarrollo para Python: sudo apt install libffi-dev libssl-dev.

Ajuste 3 (Crítico): Se identificó una incompatibilidad entre Kivy y Cython 3.0. Se forzó un downgrade mediante el comando pip install "cython<3".

4. Gestión de Versiones y Repositorio

Prompt Utilizado:

"Necesito subir el proyecto a GitHub, pero recibo errores de que hay 'demasiados cambios activos' o que los archivos son muy pesados. El directorio contiene las carpetas .buildozer y kivy_wsl_env. ¿Cómo puedo configurar Git para ignorar estos archivos temporales de compilación y subir únicamente el código fuente, la documentación y el archivo APK final ubicado en la carpeta bin?"

Respuesta y Ajuste:

Ajuste realizado: Se creó un archivo .gitignore específico excluyendo los directorios de construcción. Se utilizó el comando git add -f bin/ para forzar la inclusión del instalable APK, asegurando una entrega completa conforme a la rúbrica.

Para esta evaluación, la Inteligencia Artificial (Gemini) fue una ayuda primordial. Dado que decidí realizar el proyecto desde cero, utilicé la IA para guiarme paso a paso en la creación completa de la aplicación, desde la escritura del código hasta la solución de errores complejos en la compilación.

No solo la usé para corregir fallos, sino como un tutor que me fue indicando exactamente qué escribir y qué comandos ejecutar en cada etapa. Sin esta guía detallada, configurar el entorno en Linux y lograr que el APK funcionara habría sido mucho más difícil, ya que muchas instrucciones de las guías antiguas no servían para mi versión de Ubuntu.

<https://github.com/punshaa/MetriKivy-Evaluacion.git>