

# Mini-Informe

Paquetización de app Kivy (.py) a .apk

Estudiante: Benjamin C. Dos Santos

Ramo: Desarrollo Móvil

Docente: Cristian Iglesias

## Introducción

En el siguiente documento, se hablará sobre como fue el proceso y desarrollo de convertir una aplicación desarrollada en **Python** utilizando el framework **Kivy** en un archivo **.APK** instalable en dispositivos Android.

La compilación se realizó utilizando **Buildozer** y **python-for-android** dentro de un entorno **WSL** (Subsistema de Windows para Linux), en este caso, **Ubuntu** en **Windows**, configurado especialmente para generar el **APK** desde cero.

## Pasos que se hizo durante el desarrollo

Durante el desarrollo se siguieron las siguientes etapas principales:

### 1. Configuración del entorno

Se hizo la instalación de dependencias necesarias en Ubuntu/WSL, tales como:

- Python 3.11 y sus herramientas (venv, dev)
- Java JDK 17
- Autotools (autoconf, automake, libtool)
- pkg-config

Luego se hizo la creación de un entorno virtual con **Python 3.11** para evitar fallos de compatibilidad.

Y de ahí, se procedió con la instalación de Buildozer, Kivy y una versión compatible de Cython (En este caso, 0.29.x).

### 2. Preparación del proyecto

Inicialización del archivo **buildozer.spec** mediante **buildozer init**, Luego la configuración de los requisitos del proyecto (Python3 y Kivy) y por último, la limpieza de builds antiguos y regeneración cuando fue necesario.

### 3. Compilación

Todo ya hecho, se procedió con la ejecución del comando de compilación:

**“buildozer -v android debug”**

Se uso archivo de logs (buildozer.log) para identificar y resolver errores, para hacer reconstrucciones del proyecto después de corregir los problemas detectados.

### 4. Corrección de errores durante la compilación

Para cada error surgido, se instalaron o ajustaron las herramientas correspondientes y se reintentó el proceso de compilación.

## Resultado obtenido

Finalmente, se logró:

- Compilar exitosamente la aplicación Python/Kivy.
- Generar un archivo **APK** funcional (**Hola.apk**).
- Resolver todos los errores relacionados con dependencias, compilación Cython, librerías nativas, y compatibilidades del entorno.
- Establecer un entorno estable para futuras compilaciones.

El objetivo inicial, el cual era obtener un **APK** desde un proyecto Python, fue **cumplido con éxito**.

## Problemas encontrados durante el desarrollo

A lo largo del proceso surgieron múltiples dificultades técnicas, entre ellas:

### 1. Incompatibilidad con Python 3.12

Buildozer y python-for-android aún dependen de **distutils**, que ya no existe en Python 3.12. Lo cual obligó a instalar y usar **Python 3.11**, una versión totalmente compatible.

### 2. Falta de herramientas esenciales

El proyecto falló inicialmente debido a la ausencia de: **Autoconf, Automake, Libtool y de pkg-config**. Estas herramientas eran necesarias para compilar **libffi**, una dependencia nativa.

### 3. Problemas con Cython 3.x

La librería **pyjnius** no es compatible con **Cython 3.x**.

Se generaba el error: “**undeclared name not builtin: long**”

Para solucionarlo, se instaló **Cython 0.29.36**, compatible con el código existente.

### 4. Builds parciales dañados

Después de cada error, era necesario borrar directorios de compilación fallida para permitir un nuevo build limpio.

## **Uso de IA (Inteligencia Artificial)**

La IA que utilice fue ChatGPT

El uso de ChatGPT durante el proceso de paquetización permitió:

### **1. Diagnosticar errores complejos:**

A lo largo del desarrollo surgieron múltiples fallos relacionados con compilación, dependencias, versiones de Python, herramientas del sistema, Cython y librerías nativas. ChatGPT analizó los logs que yo le proporcione y ayudó a identificar con precisión la causa de cada problema, incluso cuando los mensajes de error eran extensos o confusos.

### **2. Proporcionar soluciones inmediatas y contextualizadas:**

ChatGPT ofreció instrucciones específicas para resolver cada error detectado, incluyendo:

- Instalación de paquetes faltantes.
- Reemplazo de versiones incompatibles de Python y Cython.
- Limpieza selectiva de builds dañados.
- Configuración correcta del entorno virtual y de Buildozer.
- Ajustes al archivo buildozer.spec.

**Enlace REPO: [https://github.com/BenDDoSantos/demo\\_paquetizacion\\_kivy.git](https://github.com/BenDDoSantos/demo_paquetizacion_kivy.git)**