# Guía de instalación y configuración del entorno (Windows + CMD) – Appium + Python para Vita Wallet

Documento técnico detallado y reproducible para instalar, configurar y dejar listo el entorno de automatización móvil con Appium (Android) y Python. Todos los pasos usan CMD (NO PowerShell) y las rutas/valores corresponden al entorno real utilizado por la usuaria (vaned).

## 0. Requisitos y convención

• Sistema operativo: Windows 10/11 64‑bits.

• Consola: CMD (Símbolo del sistema). Evitar PowerShell para estos comandos.

• Usuario Windows: vaned.

• Carpeta de trabajo del proyecto: C:\Users\vaned\Downloads\VitaAppium

• APK a probar: C:\Users\vaned\Downloads\VitaAppium\VitaQA.apk

• Emulador objetivo: Pixel 6 – Android 14 (API 34).

• Appium Server: http://127.0.0.1:4723 (mantener la ventana abierta durante las pruebas).

## 1. Instalar Node.js LTS y Appium 2

1.1 Verificar si Node.js ya está instalado:

node -v

npm -v

1.2 Si no está instalado:

winget install OpenJS.NodeJS.LTS -h

1.3 Cerrar y reabrir CMD, verificar versiones:

node -v

npm -v

1.4 Instalar Appium 2 globalmente:

npm i -g appium

1.5 Verificar Appium:

appium -v (ej. v3.0.2)

1.6 Instalar driver Android UIAutomator2:

appium driver install uiautomator2

1.7 Verificar drivers:

appium driver list (debe verse uiautomator2@5.0.3 [installed])

## 2. Instalar Android Studio + SDK + Emulador (API 34)

2.1 Instalar Android Studio (con Winget o manual):

winget install Google.AndroidStudio -h

2.2 Abrir Android Studio → More Actions → SDK Manager.

2.3 SDK Platforms: marcar Android 14 (UpsideDownCake) – API Level 34. Desmarcar previews (ej. Android 16).

2.4 SDK Tools: marcar (Show Package Details si corresponde):

• Android SDK Platform-Tools

• Android Emulator

• Android Emulator Hypervisor Driver (si aparece)

(Si NO usarás dispositivo físico, NO es necesario Google USB Driver).

2.5 Aplicar (Apply) y finalizar la instalación de componentes.

2.6 Crear emulador: More Actions → Device Manager → Create Device → Pixel 6 → System Image API 34 (Google APIs x86\_64) → Finish.

2.7 Iniciar el emulador (Play ▶). Esperar la pantalla de inicio de Android.

2.8 Verificar ADB ve el dispositivo:

adb devices (debe aparecer emulator-5554 device)

Ruta SDK esperada: %LOCALAPPDATA%\Android\Sdk (C:\Users\vaned\AppData\Local\Android\Sdk)

## 3. Configurar variables de entorno (ANDROID\_HOME y PATH)

3.1 ANDROID\_HOME:

setx ANDROID\_HOME "%LOCALAPPDATA%\Android\Sdk"

(Cerrar y reabrir CMD para que surta efecto)

3.2 Agregar al PATH (usuario) las herramientas del SDK (cuidado si el PATH es muy largo):

setx PATH "%PATH%;%ANDROID\_HOME%\platform-tools;%ANDROID\_HOME%\emulator"

(Cerrar y reabrir CMD).

3.3 Verificar ADB:

adb version

## 4. Crear carpeta VitaAppium y ubicar la APK

4.1 Crear carpeta del proyecto en Descargas:

mkdir %HOMEPATH%\Downloads\VitaAppium

4.2 Mover la APK a la carpeta del proyecto (si estaba en Descargas):

move %HOMEPATH%\Downloads\VitaQA.apk %HOMEPATH%\Downloads\VitaAppium\VitaQA.apk

Ruta final esperada: C:\Users\vaned\Downloads\VitaAppium\VitaQA.apk

## 5. Instalar Python y crear entorno virtual + dependencias

5.1 Verificar Python:

python --version

5.2 Si no está instalado:

winget install Python.Python.3.12 -h

5.3 Crear y activar entorno virtual en la carpeta del proyecto:

cd %HOMEPATH%\Downloads\VitaAppium

python -m venv .venv

.\.venv\Scripts\activate

5.4 Actualizar pip e instalar dependencias:

pip install --upgrade pip

pip install Appium-Python-Client selenium pytest

(Si aparece error con UiAutomator2Options, usar AppiumOptions: lo documentamos en la sección 8).

## 6. Estructura inicial del proyecto

6.1 Crear carpetas (en CMD, preferiblemente):

mkdir tests pages utils

\*Si usas PowerShell, crea una por vez (mkdir tests / mkdir pages / mkdir utils).

6.2 Crear archivos para que Python trate las carpetas como paquete:

type NUL > tests\\_\_init\_\_.py

type NUL > pages\\_\_init\_\_.py

6.3 Crear archivo tests\conftest.py con la configuración del driver Appium (fixture).

6.4 Verificar estructura:

VitaAppium\  
├─ pages\  
│ └─ \_\_init\_\_.py  
├─ tests\  
│ └─ \_\_init\_\_.py  
├─ utils\  
└─ .venv\

## 7. Appium Server y Appium Inspector

7.1 Levantar Appium Server (ventana dedicada de CMD, mantener abierta):

appium

Salida esperada: 'Welcome to Appium v3.0.2' y 'Appium REST http interface listener started on http://0.0.0.0:4723'

7.2 Instalar Appium Inspector (descarga .exe):

Descargar 'Appium-Inspector-<versión>-win-x64.exe' desde releases de GitHub y ejecutar instalador.

7.3 Abrir Appium Inspector y configurar conexión:

Remote Host: 127.0.0.1

Remote Port: 4723

Desired Capabilities (JSON):

{

"platformName": "Android",

"automationName": "UiAutomator2",

"app": "C:\\Users\\vaned\\Downloads\\VitaAppium\\VitaQA.apk",

"newCommandTimeout": 180,

"autoGrantPermissions": true,

"dontStopAppOnReset": false,

"fullReset": false,

"noReset": true

}

Start Session → debe visualizarse la app en el emulador y el árbol de elementos.

## 8. Resolución de errores frecuentes (observados en este setup real)

8.1 'pytest' no se reconoce como comando → activar el entorno virtual:

.\.venv\Scripts\activate

python -m pytest -q (alternativa si 'pytest' sigue sin resolverse)

8.2 ModuleNotFoundError: No module named 'pages' → crear \_\_init\_\_.py:

type NUL > pages\\_\_init\_\_.py

type NUL > tests\\_\_init\_\_.py

8.3 ImportError UiAutomator2Options → usar AppiumOptions genérico:

from appium.options.common import AppiumOptions

# y pasar options=options a webdriver.Remote(...) en lugar de desired\_capabilities

8.4 Appium Inspector vía npm 404 → instalar versión .exe desde GitHub (no via npm).

8.5 Tap vs Send Keys en Inspector → para click usar el botón 'Tap' (manito), no 'Send Keys'.

8.6 El 'TextView' no es clickable → seleccionar el padre clickable en App Source o usar UiSelector por texto:

AppiumBy.ANDROID\_UIAUTOMATOR, 'new UiSelector().text("Iniciar sesión")'

8.7 Emulador preview API 36 → crear AVD estable con API 34 (UpsideDownCake).

8.8 PATH demasiado largo → si setx PATH falla o trunca, editar PATH de usuario desde Panel de control → Sistema → Configuración avanzada → Variables de entorno.

## 9. Checklist final – entorno listo para pruebas

□ Appium 2 instalado y ejecutando en http://127.0.0.1:4723 (ventana con logs abierta).

□ Driver uiautomator2 instalado (listado en 'appium driver list').

□ Android Studio con SDK API 34 instalado.

□ Emulador Pixel 6 – API 34 iniciado (adb devices muestra 'device').

□ Carpeta de proyecto creada: C:\Users\vaned\Downloads\VitaAppium con VitaQA.apk dentro.

□ Entorno virtual Python activo (.venv) y dependencias instaladas (Appium-Python-Client, selenium, pytest).

□ Appium Inspector instalado (.exe) y capaz de iniciar sesión con las capabilities indicadas.