# Guía de instalación y configuración del entorno de pruebas con Appium y Python

Este documento describe de manera detallada, paso a paso y desde cero, cómo instalar, configurar y preparar el entorno necesario para la ejecución de pruebas automatizadas sobre la aplicación Vita Wallet, utilizando Appium, Android Studio y Python. Está pensado para personas que no tienen experiencia previa con estas herramientas.

## 1. Instalación de Node.js y Appium

1.1 Verificar si Node.js ya está instalado en el sistema:

node -v

npm -v

1.2 Si no está instalado, ejecutar en CMD:

winget install OpenJS.NodeJS.LTS -h

1.3 Verificar nuevamente la instalación:

node -v

npm -v

1.4 Instalar Appium 2 globalmente:

npm install -g appium

1.5 Verificar la instalación de Appium:

appium -v

1.6 Instalar el driver Android UiAutomator2:

appium driver install uiautomator2

1.7 Verificar que el driver aparece como instalado:

appium driver list

## 2. Instalación de Android Studio, SDK y Emulador

2.1 Instalar Android Studio (puede hacerse desde CMD con winget o desde la página oficial).

winget install Google.AndroidStudio -h

2.2 Abrir Android Studio e ir a 'More Actions' → 'SDK Manager'.

2.3 En la pestaña SDK Platforms, seleccionar Android 14 (UpsideDownCake) – API Level 34.

2.4 En la pestaña SDK Tools, marcar:

- Android SDK Platform-Tools

- Android Emulator

- Android Emulator Hypervisor Driver

2.5 Aplicar cambios y esperar la descarga.

2.6 Crear un emulador en Android Studio desde 'Device Manager':

- Seleccionar un dispositivo (ejemplo: Pixel 6).

- Descargar e instalar la imagen de sistema con API 34.

- Finalizar la creación.

2.7 Iniciar el emulador desde Android Studio y verificar que arranca correctamente.

2.8 Verificar que el emulador está conectado con ADB desde CMD:

adb devices

## 3. Configuración de variables de entorno

3.1 Configurar la variable ANDROID\_HOME:

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

3.2 Agregar al PATH los siguientes directorios:

%ANDROID\_HOME%\platform-tools

%ANDROID\_HOME%\emulator

3.3 Verificar que ADB funciona:

adb version

## 4. Instalación de Python y creación de entorno virtual

4.1 Verificar si Python está instalado:

python --version

4.2 Si no está instalado, ejecutar:

winget install Python.Python.3.12 -h

4.3 Crear carpeta del proyecto VitaAppium:

mkdir %HOMEPATH%\Downloads\VitaAppium

4.4 Entrar en la carpeta:

cd %HOMEPATH%\Downloads\VitaAppium

4.5 Crear un entorno virtual:

python -m venv .venv

4.6 Activar el entorno virtual:

.\.venv\Scripts\activate

4.7 Instalar dependencias necesarias:

pip install --upgrade pip

pip install Appium-Python-Client selenium pytest

## 5. Instalación de Appium Inspector

5.1 Descargar el instalador desde GitHub (archivo Appium-Inspector-win-x64.exe).

5.2 Ejecutar el instalador y completar la instalación.

5.3 Abrir Appium Inspector desde el menú Inicio.

5.4 Configurar la sesión con los siguientes parámetros:

Remote Host: 127.0.0.1

Remote Port: 4723

Desired Capabilities:

{

"platformName": "Android",

"automationName": "UiAutomator2",

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

"newCommandTimeout": 180,

"autoGrantPermissions": true,

"dontStopAppOnReset": false,

"fullReset": false,

"noReset": true

}

## 6. Creación de carpetas y archivos del proyecto

6.1 Dentro de VitaAppium, crear las carpetas necesarias:

mkdir tests pages utils

6.2 Crear el archivo conftest.py dentro de la carpeta tests con la configuración del driver Appium.

6.3 Crear los archivos base para las páginas (ejemplo: login\_page.py) y para los tests (ejemplo: test\_login.py).

6.4 Verificar que la estructura del proyecto queda así:

VitaAppium/  
├── pages/  
│ └── login\_page.py  
├── tests/  
│ └── conftest.py  
│ └── test\_login.py  
├── utils/  
└── .venv/

## Conclusión

Con estos pasos se completa la instalación y configuración del entorno para pruebas móviles con Appium, dejando listo el emulador, el servidor Appium, el cliente Python y Appium Inspector. El siguiente documento describirá la implementación de pruebas automatizadas.