# Guía de instalación y ejecución de pruebas con Appium + Python

Este documento describe paso a paso la configuración del entorno y la ejecución de pruebas automatizadas en la aplicación Vita Wallet, utilizando Appium, Python y Android Studio. Los comandos incluidos son los utilizados en el proceso real, de manera que cualquier persona pueda reproducir la instalación.

## 1. Preparar entorno base

Instalar Node.js desde la página oficial.

Instalar Appium 2 globalmente:

npm install -g appium

Instalar el driver de Android UiAutomator2:

appium driver install uiautomator2

Verificar que Appium arranca correctamente:

appium

## 2. Android Studio + SDK + Emulador

Instalar Android Studio y configurar los SDKs necesarios.

Verificar que el SDK está en la ruta: C:\Users\vaned\AppData\Local\Android\Sdk

Crear un emulador (ejemplo: Pixel 7 con Android 14).

Ejecutar el emulador para comprobar que inicia correctamente.

## 3. Python + entorno virtual + dependencias

Ubicarse en la carpeta del proyecto VitaAppium:

cd C:\Users\vaned\Downloads\VitaAppium

Crear un entorno virtual:

python -m venv .venv

Activar el entorno virtual:

.\.venv\Scripts\activate

Instalar dependencias:

pip install Appium-Python-Client selenium pytest

## 4. Estructura del proyecto

La estructura del proyecto se organizó en carpetas:

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

Ejemplo de import en test\_login.py:

from pages.login\_page import LoginPage

## 5. Appium Inspector

Descargar Appium Inspector desde GitHub (versión .exe para Windows).

Ejecutar Appium Inspector y configurar la sesión:

Host: 127.0.0.1

Port: 4723

APK: C:\Users\vaned\Downloads\VitaAppium\VitaQA.apk

## 6. Locators obtenidos

Correo electrónico:

//android.widget.EditText[1]

Contraseña:

//android.widget.EditText[2]

Botón Ingresar:

//android.view.ViewGroup[@content-desc="Ingresar"]

## 7. Ejecución del test

Desde la raíz del proyecto, con el entorno virtual activado:

.\.venv\Scripts\activate

pytest -q -k test\_login

Resultado esperado:

1 passed