

Paso a paso para correr correctamente una solución Playwright con automatización móvil Android y Navegador Chrome

Automatizar una prueba de inicio de sesión en Gmail y Chrome usando **Playwright**, ejecutándola en un **dispositivo Android real** con **Google Chrome** y en el navegador **Google Chrome en PC**

REQUISITOS

En tu computador:

1. **Node.js** (v16 o superior)
👉 Descarga: <https://nodejs.org/>

Playwright

bash

```
npm install -D @playwright/test  
npx playwright install
```

2. **ADB (Android Debug Bridge)** instalado y funcionando
Generalmente viene con Android Studio o puede instalarse desde:
<https://developer.android.com/studio/releases/platform-tools>
3. **Dispositivo Android físico** con:
 - **Modo desarrollador activado**
 - **Depuración USB habilitada**
 - **Google Chrome instalado**
 - Conectado por cable USB al computador

PRUEBA DE CONEXIÓN CON ADB

bash

```
adb devices
```

Deberías ver tu dispositivo listado como `device`.

INSTALACIÓN Y CONFIGURACIÓN DE PLAYWRIGHT

Paso 1: Inicializar proyecto

bash

```
npm init playwright@latest
```

Responde las preguntas (puedes marcar que no usarás TypeScript si lo prefieres).

Paso 2: Instalar navegadores

bash

```
npx playwright install
```

Instala Chromium, Firefox y WebKit.

CONFIGURAR PLAYWRIGHT PARA ANDROID

1. Importar el módulo Android:

ts

```
import { test, _android as android } from '@playwright/test';
```

2. Conectarse al dispositivo:

ts

```
const [device] = await android.devices();
```

3. Lanzar navegador Chrome en Android:

ts

```
await device.shell('am force-stop com.android.chrome'); // Detiene Chrome
const context = await device.launchBrowser(); // Lanza Chrome
const page = await context.newPage(); // Nueva pestaña
await page.goto('https://accounts.google.com');
```

TEST COMPLETO DE EJEMPLO

ts

```
import { test, _android as android } from '@playwright/test';

test('Login Gmail en Android', async ({}, testInfo) => {
  test.setTimeout(120_000); // Aumenta timeout a 2 min

  const [device] = await android.devices();
  console.log(`Modelo: ${device.model()}`);
  console.log(`Serial: ${device.serial()}`);

  await device.shell('am force-stop com.android.chrome');
  const context = await device.launchBrowser();
  const page = await context.newPage();

  await page.goto('https://accounts.google.com/');
  await testInfo.attach('Pantalla Gmail', {
    body: await page.screenshot(),
    contentType: 'image/png',
  });

  // Aquí deberías agregar tus métodos de login:
  // await page.fill('//input[@type="email"]', 'tu_email');
  // await page.click('//button[@type="submit"]');
});
```

EJECUTAR EL TEST

bash

```
npx playwright test
```

O ejecuta un archivo específico:

bash

```
npx playwright test tests/loginGmail.spec.ts
```

ESTRUCTURA DEL PROYECTO (recomendada)

lua

```
/project-root
├─ tests/
│   └─ loginGmail.spec.ts
├─ pageObjects/
│   └─ PO_LoginGmail.ts
├─ selectors/
│   └─ QS_LoginGmail.ts
├─ playwright.config.ts
└─ package.json
```

CONSEJOS

- Siempre aumenta el timeout para pruebas móviles (como viste con `test.setTimeout()`).
 - No uses **page** como parámetro si estás trabajando con Android, ya que **page** debe obtenerse desde `device.launchBrowser()`.
-

OPCIONAL: CONFIGURACIÓN GLOBAL EN `playwright.config.ts`

ts

```
import { defineConfig } from '@playwright/test';

export default defineConfig({
  timeout: 120_000,
  use: {
    screenshot: 'only-on-failure',
    trace: 'on-first-retry',
  },
});
```

¿Y si no funciona?

Verifica:

¿Chrome está instalado en el celular? Usa:

bash

```
adb shell pm list packages | grep chrome
```

- ¿Tu dispositivo aparece en `adb devices`?
- ¿Autorizaste la depuración USB?

Para generar y ver el reporte HTML

Una vez ejecutes tus pruebas:

bash

```
npx playwright test
```

Puedes abrir el reporte con:

bash

```
npx playwright show-report
```