

# Pruebas de Reconocimiento | Los Estudiantes

- **Autor:** Héctor Franco
- **Código:** 202412495
- **Curso:** MISW4103 - Pruebas Automatizadas de Software

## 1. Descripción de la prueba

El [Ejercito de los 12 Monos](#) ha decidido atacar el sitio web de [Los Estudiantes](#). Para ello lo ha contratado a usted, un hábil tester que hará todo lo posible por poner a prueba la aplicación en cuestión. para ello se va a realizar una prueba de reconocimiento basada en Random Testing o Monkey Testing.

A continuación de muestran los pasos a seguir para ejecutar el ataque con éxito y hacer que Los Estudiantes puedan ver lo que es capaz de hacer el Ejercito de los 12 Monos.

## 2. Ambiente de ejecución de la prueba

La prueba construida por el Ejercito se hizo con las siguientes especificaciones que debe seguir al pie de la letra para poder **replicar** la prueba en su máquina loca.

### 2.1 Especificaciones técnicas del ambiente de pruebas usado:

- Máquina: MacBook Pro M1 (2021)
- SO: MacOS Sonoma v14.1.1
- Node Version: v20.12.0
- NPM Version: v10.5.0
- Cypress Version: v13.7.3
- Chrome Browser Version: v124

### 2.2 Instalar Node JS o NVM (Node Version Manager )

Para poder replicar bien este set de pruebas es requerido instalar en su máquina local la versión de [Node JS](#) descrita en la sección 2.1, o mejor aun, instalar [NVM](#), para poder alternar entre las diferentes versiones de Node disponibles.

Una vez instalado, se puede comprobar con el siguiente comando en la terminal o linea de comando de windows: Node:

```
> node --version
v20.12.0
```

npm:

```
> npm --version
10.5.0
```

### 2.3 Instalar Cypress v13.7.3

Una vez instalado Node en la máquina local procedemos a instalar cypress de forma global con el siguiente comando

```
> npm -g install cypress@13.7.3
```

### 2.4 Instalar Google Chrome v124

Cypress viene con un navegador web por defecto que sirve para visualizar las pruebas, pero en esta ocasión se usó [Google Chrome](#). Por favor asegurarse que su versión de Chrome es la 124 o posterior.

### 2.5 Descomprimir la carpeta del proyecto

El archivo .zip del proyecto se llama **Monkey\_LosEstudiantes**, procedemos a descomprimir el .zip y dejarlo en nuestra ruta de preferencia, puede ser en /Documents, /Desktop o la que prefieran.

### 2.6 Instalar las librerías del proyecto

El proyecto usa una librería llamada **faker**, la cual sirve para generar datos aleatorios que serán usados en el código fuente de los Monkeys, cómo nombres y palabras predefinidas. A su vez permite generar una **semilla** sobre la cual se generarán los datos aleatorios para que se pueda mantener la **replicabilidad** de la prueba sin importar en qué máquina se ejecute.

Para instalar la librería del proyecto se deben hacer dos pasos unicamente:

1. Abrir una terminal o consola de comandos (CMD) sobre la carpeta raíz del proyecto, llamada **Monkey\_LosEstudiantes**
2. Ejecutar el siguiente comando de Node:

```
> npm install
```

Y listo..., Ya podemos proceder a la mejor parte del trabajo.

### 3. Cómo ejecutar las pruebas

Una vez nos aseguramos que todo el ambiente de pruebas está listo, con Node, NPM, Cypress, el navegador Chrome y las librerías de proyecto instaladas, procedemos a ejecutar las pruebas de reconocimiento con los siguientes pasos:

#### 3.1 Correr instancia de Cypress

En una terminal o consola de comandos abierta corremos el siguiente comando:

```
> cypress open
```

#### 3.2 Seleccionar la carpeta del proyecto Cypress:

Presionamos botón **Add project** de la vista principal de Cypress y seleccionamos la carpeta raíz del proyecto llamada **Monkey\_LosEstudiantes**.

*NOTA: Si a la primera vez que intentan abrir el proyecto desde Cypress este queda en una pantalla de loading que no avanza, por favor cerrar la venta del programa y volver a ejecutar y seleccionar la carpeta del proyecto de nuevo hasta que cargue completamente.*

#### 3.3 Seleccionar la prueba E2E

Las pruebas de reconocimiento que se harán son del tipo E2E (Extremo a Extremo), Por ende procedemos a escoger el cuadro de texto que dice **E2E Testing**

#### 3.4 Iniciar la prubeba E2E

Cypress nos abrirá una ventana donde selecciona por defecto el Navegador Chrome, con un botón en color verde, el cual debemos precionar y que dice **Start E2E Testing in Chrome**,

#### 3.5 Ejecutar la prueba

Una vez presionado el botón de la sección 3.4, se abrirá una ventana de Google Chrome en donde aparece el proyecto mostrando el árbol de archivos de la carpeta e2e, debe lucir así:

```
cyress/e2e
  monkey_testing.cy.js
```

damos click al archivo **monkey\_testing.cy.js** y la prueba ejecutará automaticamente

### 4. Descripción del código fuente de la prueba E2E

#### 4.1 Función randomEvent()

Esta es la función maestro que llama de forma aleatoria cada uno de los eventos sobre el cual el mono va a operar en la página de Los Estudianes.

#### 4.2 Tipos de eventos

En total hay 10 tipos de eventos que serán usados por el Ejercito de monos los cuales se listan en la siguiente tabla:

Evento	Descripción	Función
Clicks en enlaces	Este evento se encarga de hacer clicks aleatorios sobre enlaces <code>&lt;a&gt;</code> de la página web	randomClicksOverLinks()
Llenar campos de texto	Aquí se buscan todos los elementos <code>&lt;input&gt;</code> de la página y son llenados con datos aleatorios de nombres, emails, telefonos y contraseñas	randomInputFills()
Seleccionar elementos	Aquí se buscan los elementos <code>&lt;select&gt;</code> de la página y se intenta seleccionar campos al azar	randomSelectsOverComboBox()
Clicks en botones	En este evento se busca hacer clicks sobre elementos <code>&lt;button&gt;</code> de la página web	randomClicksOverButtons()
Scroll sobre la ventana	En este evento se hace scroll sobre los distintos puntos de la página de forma <code>vertical</code> , <code>top</code> , <code>bottom</code> , <code>right</code> , etc	randomWindowScroll()
Doble Clicks en botones	En este evento se busca hacer dobles clicks sobre elementos <code>&lt;button&gt;</code> de la página web	randomDoubleClicksOverButtons

ir a un sitio	Este evento redirecciona la página a sus páginas internas dentro de Los Estudiantes	randomGoVisit()
Crear cookies	Aquí se pretende crear cookies aleatorias dentro de la aplicación web	randomSetCookies()
Ajustar el viewport	Este evento hace que se redimensione el viewport de la página web a distintos tamaños	randomSetViewport()
Capturas de pantalla	Este evento hace capturas de pantallas aleatorias de la vista en donde se encuentre.	randomTakeScreenshots()

### 4.3 Cantidad de intentos

La prueba se ha parametrizado con una cantidad fija de intentos adicionales en caso de que al primer intento fallen debido a que no se encuentra un elemento dentro de la vista actual. El código en cuestión que realiza los intentos adicionales se encuentra en la línea 5 del archivo **monkey\_testing.cy.js**, y luce de la siguiente manera

```
{ retries: 20 }
```

## 5. ¿Será que Los Estudiantes sobreviven a este ataque?

Eso es todo, una vez empieza a correr la prueba la página de Los Estudiantes será puesta a prueba con un ejercito de monos aleatorios que jugarán con el sistema para ver su comportamiento bajo pruebas, mientras usted solo se sienta y disfruta de ver cómo la página es atacada.

