

INSTITUTO TECNOLÓGICO DE COSTA RICA
CURSO: ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE
PROFESOR: ANDRÉS VÍQUEZ VÍQUEZ
LABORATORIO: CYPRESS

Cypress es una plataforma de automatización de pruebas de extremo a extremo para aplicaciones web modernas que permite a los desarrolladores y equipos de prueba escribir, ejecutar y depurar pruebas de una manera más eficiente utilizando lenguajes de programación como JavaScript y TypeScript. Con herramientas de depuración avanzadas y una API intuitiva, Cypress ayuda a los desarrolladores a encontrar y corregir problemas en sus aplicaciones de manera más rápida y efectiva. En este taller vamos a desarrollar un ejemplo para aprender a utilizar la herramienta Cypress.

1. Verifique que cuenta con Java instalado en el equipo, para esto ejecute en una terminal el siguiente comando:

```
java -version
```

En caso de que el comando no sea reconocido, proceda a instalar una versión reciente de Java Development Kit (JDK)

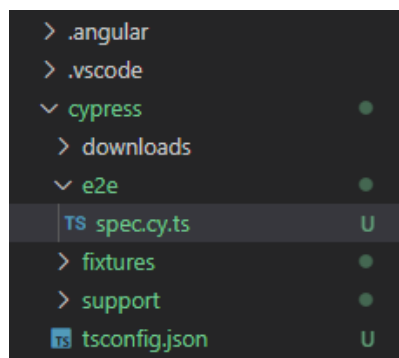
2. Cree un nuevo proyecto llamado **lab-qa3**:

```
ng new lab-qa3
```

3. Instale Cypress utilizando el comando:

```
ng add @cypress/schematic
```

Una vez instalado, podrá observar una nueva carpeta en el proyecto llamada **cypress**. Busque el archivo **cypress/e2e/spec.cy.ts**:

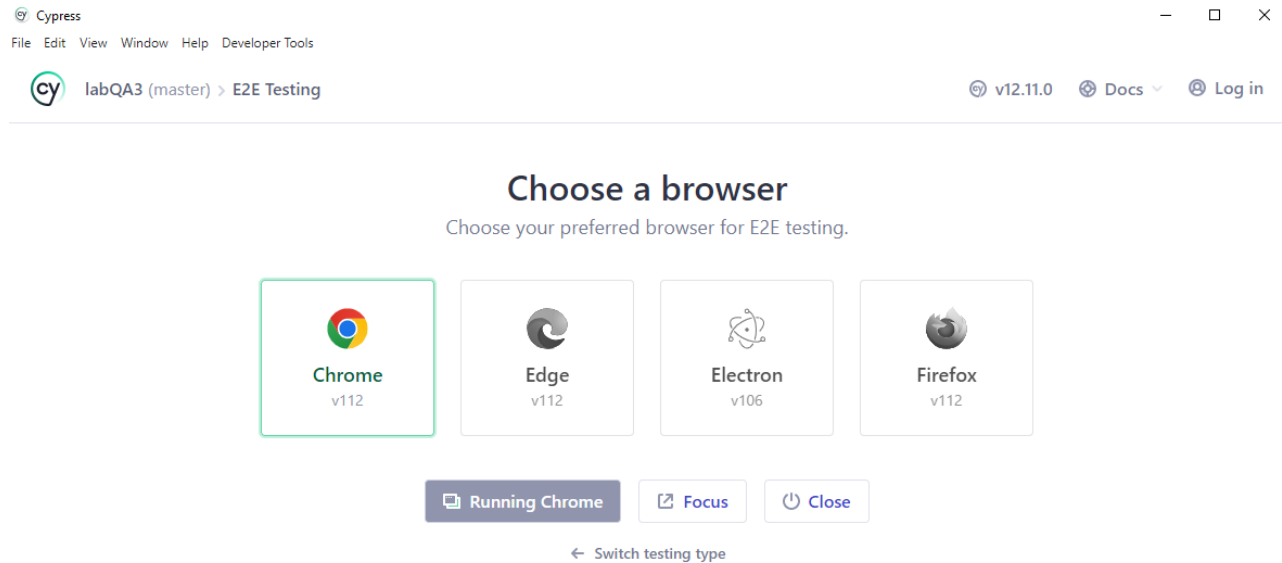


Este archivo tiene una prueba por defecto que valida que el título de la página inicial del proyecto contenga el texto “app is running!”.

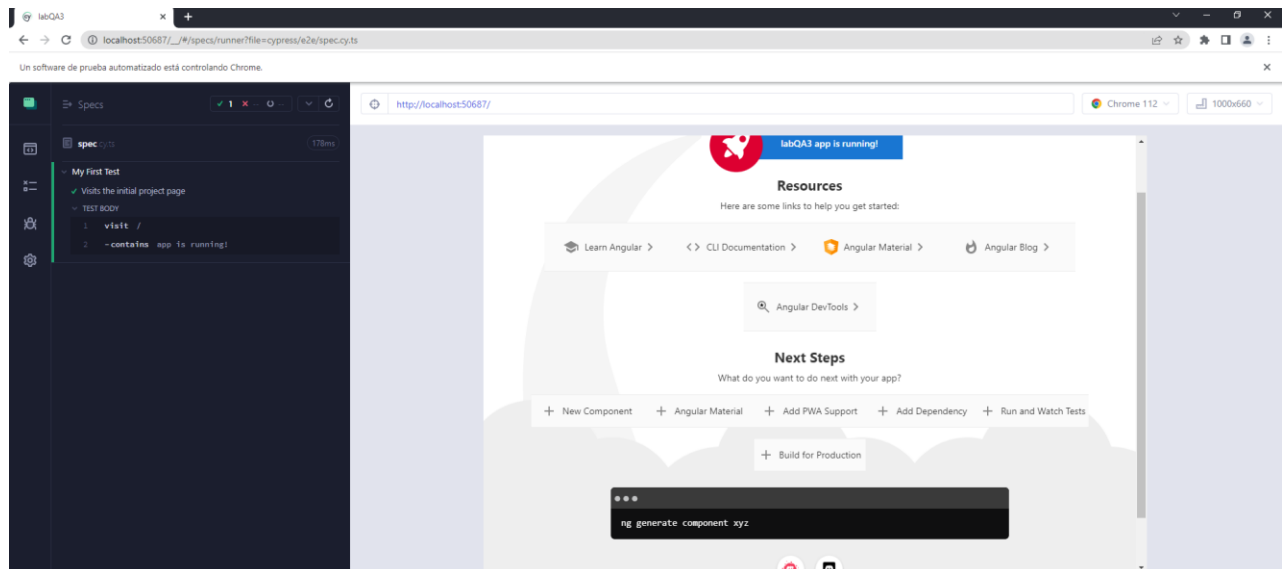
4. Ejecute Cypress utilizando el comando:

```
ng e2e
```

Aparecerá una nueva pantalla que le permitirá seleccionar el navegador para correr la prueba.



Seleccione un navegador para que inicie la ejecución de las pruebas.



5. Instale el plugin Xpath:

```
npm install -D cypress-xpath
```

Este es un complemento que agrega soporte para la selección de elementos basados en rutas XPath en las pruebas escritas con Cypress.

6. Instale los paquetes de TypeScript:

```
npm install -D @types/jest @types/cypress
```

Los archivos de definición de tipos son necesarios para que el compilador de TypeScript pueda verificar que el código escrito es correcto y también para proporcionar una mejor experiencia de programación en un editor de texto compatible con TypeScript.

7. Vamos a realizar nuestra primera prueba en Cypress. En la carpeta **cypress/e2e**, damos clic derecho para que aparezca un nuevo menú emergente y luego clic en **New File**. Colocamos como nombre al archivo **test.cy.ts**.

8. Agregue la siguiente línea de código al encabezado del archivo que acabamos de crear:

```
import '../..//node_modules/cypress-xpath'
```

9. Seguidamente procedemos a definir los métodos **before** y **after** para nuestra prueba. El método **before** nos llevará a la página que realizaremos la prueba, en este caso el sitio web del Edus, mientras que el método **after** se encargará de recargar la página.

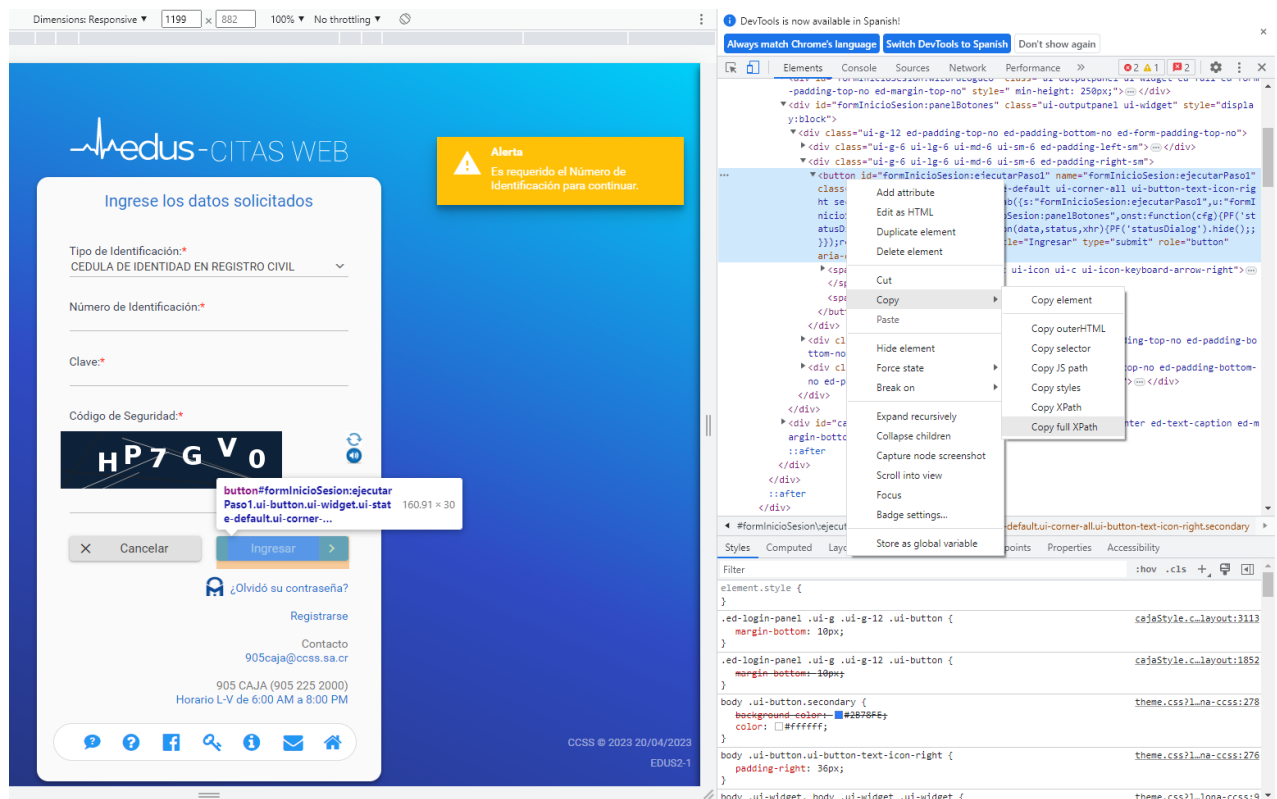
```
describe('Login1', function() {  
  beforeEach(function() {  
    cy.visit('https://edus.ccss.sa.cr/eduscitasweb/');  
  });  
  
  afterEach(function() {  
    cy.reload();  
  });  
});
```

10. Vamos a automatizar un caso de prueba para verificar el comportamiento de la aplicación cuando el usuario deja los datos en blanco en el formulario de autenticación.

```
it('Numero de identificacion requerida', () => {  
  cy.xpath('/html/body/form[1]/div[1]/div[2]/div/div[2]/div/div[2]/button').click()  
  }).then(function() {  
    cy.contains('Es requerido el Número de Identificación para continuar.');  });  
});
```

El caso de prueba utiliza el método **cy.xpath** que permite buscar un elemento directamente en un formulario a través de su ruta Xpath. Este método es sumamente útil cuando necesitamos referirnos a algún elemento de un formulario que no tiene un identificador (valor id) definido, como en este caso es el botón Ingresar.

Para copiar la ruta Xpath de algún componente, podemos hacerlo fácilmente a través de las herramientas de inspección que nos ofrezca el navegador web que estemos utilizando. Por ejemplo, en Google Chrome podemos realizarlo utilizando las herramientas para desarrolladores, seleccionamos el elemento que nos interesa obtener su Xpath, para posteriormente dar clic derecho en el código correspondiente para abrir un submenú. Seguidamente damos clic en Copiar, Copiar Full Xpath, tal como se aprecia en la imagen a continuación:



11. El siguiente caso de prueba que vamos a realizar es para cuando el usuario no introduce la contraseña.

```
it('Contraseña requerida', () => {

    cy.xpath('/html/body/form[1]/div[1]/div[2]/div/div[1]/div/div/div[2]/div[2]/input').type('123456789');

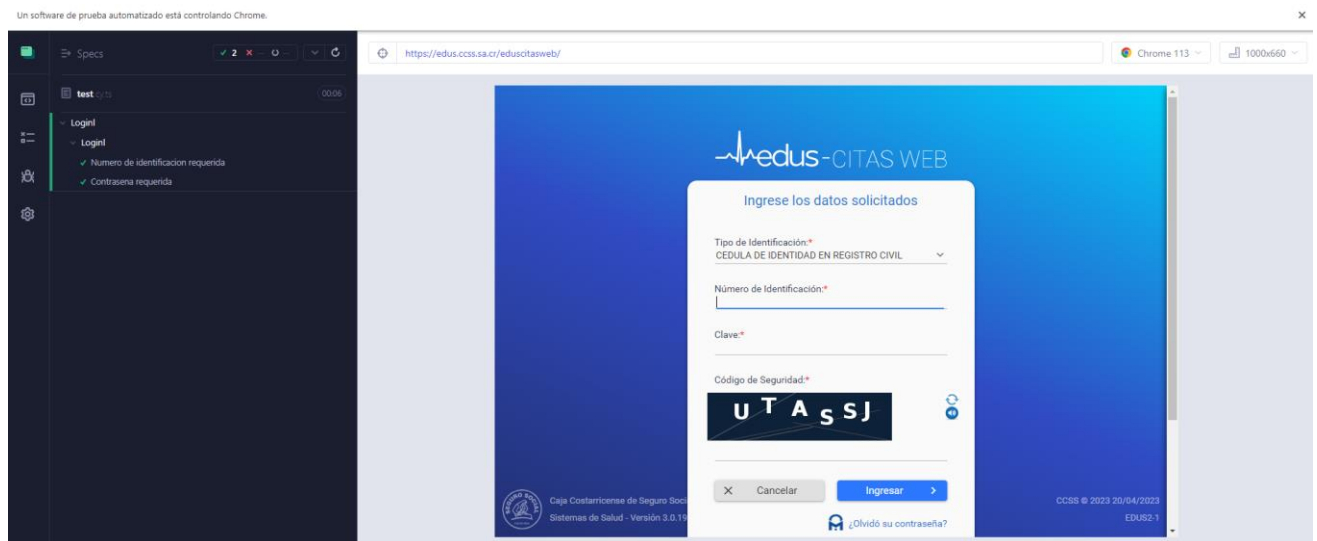
    cy.xpath('/html/body/form[1]/div[1]/div[2]/div/div[2]/div/div[2]/button').click();

    cy.contains('Es requerida la Clave para continuar');

});
```

12. Ejecute el archivo y analice los resultados.

```
TS test.cy.ts U X
labQA3 > cypress > e2e > TS test.cy.ts > describe('Login') callback > describe('Login') callback
1  import '../node_modules/cypress-xpath'
2
3  describe('Login', function() {
4    describe('Login', function() {}
5      beforeEach(function() {
6        cy.visit('https://edus.ccss.sa.cr/eduscitasweb/');
7      });
8
9      afterEach(function() {
10        cy.reload();
11      });
12
13      it('Numero de identificacion requerida', () => {
14        cy.xpath('/html/body/form[1]/div[1]/div[2]/div/div[2]/div/div[2]/button').click().then(function() {
15          cy.contains('Es requerido el Número de Identificación para continuar.');
```



Referencias de apoyo: <https://docs.cypress.io/api/table-of-contents>