VALIDACIÓN Y VERIFICACIÓN DE SOFTWARE (INF 732)
GUÍA DE LABORATORIO 11
Introducción a Cypress con Aplicación Web de Ejemplo

Objetivo

Familiarizarse con Cypress mediante la instalación, configuración y ejecución de pruebas automatizadas en una aplicación web básica (una lista de tareas).

Preparación

- Node.js y npm instalados
- Editor de código (Visual Studio Code recomendado)
- Conexión a Internet

PRIMERA PARTE
CREACIÓN DEL PROYECTO

Paso 1: Crear el proyecto

1.1. Crear una carpeta para el proyecto y ábrelo con VSCode

Abre tu terminal o línea de comandos y ejecuta:

```
mkdir cypress-lab
cd cypress-lab
code .
```

Paso 2: Inicializar el proyecto y crear el servidor

Vamos a usar Node.js y Express para servir nuestra aplicación web localmente.

2.1. Inicializa un proyecto Node.js

```
npm init -y
```

Esto creará un archivo package.json básico.

2.2. Instala Express

```
npm install express
```

Paso 3: Crear los archivos del proyecto

3.1. Crea la estructura de carpetas

En la raíz del proyecto:

```
mkdir public
```

Tu estructura será:

```
cypress-lab/
├── public/
└── (otros archivos)
```

3.2. Dentro de public/, crea index.html

Y añade el siguiente contenido:

```
<!-- public/index.html -->
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Lista de Tareas</title>
  <style>
    body { font-family: sans-serif; margin: 2em; }
    input, button { padding: 0.5em; margin: 0.5em 0; }
    li { margin: 0.3em 0; }
  </style>
</head>
<body>
  <h1>Lista de Tareas</h1>
  <input id="nueva-tarea" placeholder="Escribe una tarea" />
  <button onclick="agregarTarea()">Agregar</button>
  <ul id="lista"></ul>

  <script>
    function agregarTarea() {
      const texto = document.getElementById('nueva-
tarea').value.trim()
      if (texto) {
        const li = document.createElement('li')
        li.textContent = texto
        document.getElementById('lista').appendChild(li)
        document.getElementById('nueva-tarea').value = ''
      }
    }
  </script>
</body>
</html>
```

Esta aplicación permite:

- Escribir una tarea en un input.
- Presionar un botón para agregarla a una lista (ul).

- Las tareas aparecen dinámicamente como elementos li.

3.3. Crear el archivo del servidor

Crea el siguiente archivo en la raíz del proyecto:

```
server.js
```

Agrega este contenido:

```
// server.js
const express = require('express')
const app = express()
const port = 3000

app.use(express.static('public'))

app.listen(port, () => {
  console.log(`Servidor corriendo en http://localhost:${port}`)
})
```

Paso 4: Ejecutar la aplicación

Desde la terminal, en la carpeta del proyecto, ejecuta:

```
node server.js
```

Deberías ver algo como:

```
Servidor corriendo en http://localhost:3000
```

Abre tu navegador y visita:
http://localhost:3000

Verás tu aplicación en funcionamiento.

SEGUNDA PARTE
PRUEBAS CON CYPRESS

Excelente, ahora que tienes tu aplicación web en funcionamiento, vamos a instalar y comenzar a usar **Cypress** para escribir pruebas automatizadas sobre ella.

Paso 5: Instalar Cypress

5.1. Instala Cypress como dependencia de desarrollo

Asegúrate de estar en la raíz del proyecto (cypress-lab) y ejecuta:

```
npm install cypress --save-dev
```

Esto descargará Cypress y lo agregará en tu archivo package.json como dependencia de desarrollo.

Paso 6: Abrir Cypress por primera vez

Ejecuta:

```
npx cypress open
```

Esto hará lo siguiente:

- Creará una estructura básica con carpetas como cypress/, cypress/e2e/, etc.
- Abrirá la **interfaz gráfica de Cypress** (Test Runner), donde podrás ver y ejecutar pruebas.

La estructura será algo como:

```
cypress-lab/
├── cypress/
│   ├── e2e/
│   └── support/
├── node_modules/
├── public/
├── server.js
├── package.json
```

Si te pregunta qué tipo de pruebas quieres crear, elige "E2E Testing" y selecciona el navegador que tengas disponible (Chrome o Electron).

Paso 7: Crear una prueba básica

7.1. Crear un archivo de prueba

En la carpeta cypress/e2e/, crea el archivo:

```
cypress/e2e/lista_tareas.cy.js
```

Agrega el siguiente contenido:

```
// cypress/e2e/lista_tareas.cy.js

describe('Lista de Tareas', () => {
  beforeEach(() => {
    cy.visit('http://localhost:3000')
  })

  it('Carga correctamente la página', () => {
    cy.contains('Lista de Tareas')
  })

  it('Agrega una tarea a la lista', () => {
    cy.get('#nueva-tarea').type('Aprender Cypress')
    cy.contains('Agregar').click()
    cy.get('#lista li').should('contain.text', 'Aprender Cypress')
  })

  it('No agrega tarea vacía', () => {
    cy.contains('Agregar').click()
    cy.get('#lista li').should('have.length', 0)
  })
})
```

Paso 8: Ejecutar las pruebas

1. Asegúrate de que tu servidor esté corriendo:

   node server.js

2. En otro terminal, ejecuta:

   npx cypress open

3. En la interfaz, haz clic sobre lista_tareas.cy.js.

Verás cómo Cypress abre un navegador **automatizado**, visita tu app, interactúa con ella y verifica los resultados.