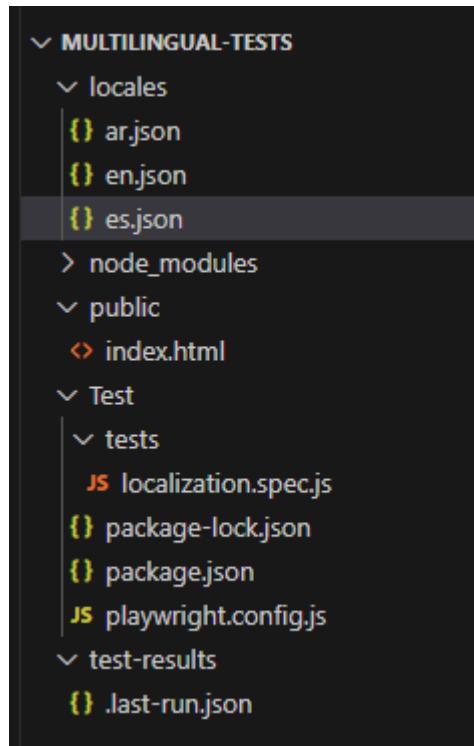


Alumno: Ricardo Bladimir Zuniga Cruz

1. Estructura:



2. Inicializar proyecto Node.js:

```
Microsoft Windows [Versión 10.0.26100.7171]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\zunig\multilingual-tests>npm init -y
Wrote to C:\Users\zunig\multilingual-tests\package.json:

{
  "name": "multilingual-tests",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \\\"Error: no test specified\\\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}
```

3. Instalar Playwright:

```
C:\Users\zunig\multilingual-tests>npm install --save-dev @playwright/test
added 3 packages, and audited 4 packages in 22s
found 0 vulnerabilities

C:\Users\zunig\multilingual-tests>npx playwright install
Downloading Chromium 143.0.7499.4 (playwright build v1200) from https://cdn.playwright.dev/dbazure/download/playwright/builds/chromium/1200/chromium-win64.zip
169.8 MiB [=====] 100% 0.0s
Chromium 143.0.7499.4 (playwright build v1200) downloaded to C:\Users\zunig\AppData\Local\ms-playwright\chromium-1200
Downloading Chromium Headless Shell 143.0.7499.4 (playwright build v1200) from https://cdn.playwright.dev/dbazure/download/playwright/builds/chromium/1200/chromium-headless-shell-win64.zip
107.2 MiB [=====] 100% 0.0s
Chromium Headless Shell 143.0.7499.4 (playwright build v1200) downloaded to C:\Users\zunig\AppData\Local\ms-playwright\chromium_headless_shell-1200
Downloading Firefox 144.0.2 (playwright build v1497) from https://cdn.playwright.dev/dbazure/download/playwright/builds/firefox/1497/firefox-win64.zip
107.1 MiB [=====] 100% 0.0s
Firefox 144.0.2 (playwright build v1497) downloaded to C:\Users\zunig\AppData\Local\ms-playwright\firefox-1497
Downloading Webkit 26.0 (playwright build v2227) from https://cdn.playwright.dev/dbazure/download/playwright/builds/webkit/2227/webkit-win64.zip
58.2 MiB [=====] 100% 0.0s
FFMPEG playwright build v1011 downloaded to C:\Users\zunig\AppData\Local\ms-playwright\ffmpeg-1011
Downloading Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
Downloaded Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
Downloading Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
Downloaded Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
Downloading Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
Downloaded Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
Downloading Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
Downloaded Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
0.1 MiB [=====] 100% 0.0s
Downloading Winldd playwright build v1007 from https://cdn.playwright.dev/dbazure/download/playwright/builds/winldd/1007/winldd-win64.zip
0.1 MiB [=====] 100% 0.0s
Winldd playwright build v1007 downloaded to C:\Users\zunig\AppData\Local\ms-playwright\winldd-1007

C:\Users\zunig\multilingual-tests>npm install express
added 65 packages, and audited 69 packages in 25s
19 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

4. Crear app web simple con i18n:

```
public > index.html > ...
1
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5    <meta charset="UTF-8">
6    <title>Multilingual App</title>
7  </head>
8  <body>
9    <h1 id="title">Hello</h1>
10   <select id="language">
11     <option value="en">English</option>
12     <option value="es">Español</option>
13     <option value="ar">العربية</option>
14   </select>
15
16  <script>
17    const translations = {
18      en: { title: "Hello" },
19      es: { title: "Hola" },
20      ar: { title: "مرحباً" }
21    };
22
23    const title = document.getElementById('title');
24    const select = document.getElementById('language');
25
26    select.addEventListener('change', () => {
27      const lang = select.value;
28      title.textContent = translations[lang].title;
29    });
30  </script>
31 </body>
32 </html>
33
```

5. Crear una página que cambie idioma dinámicamente. (public/index.html):

```
public > index.html > ...
1
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5    <meta charset="UTF-8">
6    <title>Multilingual App</title>
7  </head>
8  <body>
9    <h1 id="title">Hello</h1>
10   <select id="language">
11     <option value="en">English</option>
12     <option value="es">Español</option>
13     <option value="ar">العربية</option>
14   </select>
15
16  <script>
17    const translations = {
18      en: { title: "Hello" },
19      es: { title: "Hola" },
20      ar: { title: "مرحباً" }
21    };
22
23    const title = document.getElementById('title');
24    const select = document.getElementById('language');

25    select.addEventListener('change', () => {
26      const lang = select.value;
27      title.textContent = translations[lang].title;
28    });
29  </script>
30 </body>
31 </html>
32
33 
```

6. Crear archivos de traducción:

locales/es.json (Español)

```
locales > {} es.json > ...
1
2  {
3    "title": "Hola",
4    "welcome": "Bienvenido a nuestro sitio"
5  }
6  |
```

locales/en.json (Ingles)

```
locales > {} en.json > ...
1
2  {
3    "title": "Hello",
4    "welcome": "Welcome to our site"
5  }
6  |
```

locales/ar.json (Árabe)

```
locales > {} ar.json > ...
1
2  {
3    "title": "مرحباً",
4    "welcome": "مرحباً بكم في موقعنا"
5  }
6  |
```

7. Crear servidor simple (para Playwright):

```
Test > JS playwright.config.js > ...
1
2 import { defineConfig } from '@playwright/test';
3
4 export default defineConfig({
5   testDir: './tests', // Carpeta donde están las pruebas
6   timeout: 30000,     // Tiempo máximo por prueba (30s)
7   use: {
8     headless: true,   // Ejecutar sin abrir ventana del navegador
9     baseURL: 'http://localhost:3000', // URL del servidor
10    viewport: { width: 1280, height: 720 }, // Tamaño de pantalla
11  },
12 });
13
```

8. Escribir pruebas automatizadas con Playwright en tests/localization.spec.js:

```
Test > tests > JS localization.spec.js > ...
1
2 import { test, expect } from '@playwright/test';
3
4 test('Debe cambiar el idioma dinámicamente', async ({ page }) => {
5   // Abrir la página principal
6   await page.goto('/');
7
8   // Verificar que el texto inicial es "Hello"
9   await expect(page.locator('#title')).toHaveText('Hello');
10
11  // Cambiar a Español
12  await page.selectOption('#language', 'es');
13  await expect(page.locator('#title')).toHaveText('Hola');
14
15  // Cambiar a Árabe
16  await page.selectOption('#language', 'ar');
17  await expect(page.locator('#title')).toHaveText('مرحباً');
18 });
19
```

9.Configurar Playwright: playwright.config.js

```
Test > JS playwright.config.js > ...
1
2 import { defineConfig } from '@playwright/test';
3
4 export default defineConfig({
5   testDir: './tests', // Carpeta donde están las pruebas
6   timeout: 30000,     // Tiempo máximo por prueba (30s)
7   use: {
8     headless: true,   // Ejecutar sin abrir ventana del navegador
9     baseURL: 'http://localhost:3000', // URL del servidor
10    viewport: { width: 1280, height: 720 }, // Tamaño de pantalla
11  },
12 });
13
```

10.Ejecutar pruebas

- Terminal 1: Iniciar servidor

NOTA:

Acá se me presento un error en la ejecución sobre la terminal #1

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Microsoft Windows [Versión 10.0.26100.7171]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\zunig\multilingual-tests>node server.js
node:internal/modules/cjs/loader:1422
    throw err;
    ^
Error: Cannot find module 'C:\Users\zunig\multilingual-tests\server.js'
    at Module._resolveFilename (node:internal/modules/cjs/loader:1419:15)
    at defaultResolveImpl (node:internal/modules/cjs/loader:1057:19)
    at resolveForCJSWithHooks (node:internal/modules/cjs/loader:1062:22)
    at Module._load (node:internal/modules/cjs/loader:1225:37)
    at TracingChannel.traceSync (node:diagnostics_channel:328:14)
    at wrapModuleLoad (node:internal/modules/cjs/loader:244:24)
    at Module.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:154:5)
    at node:internal/main/run_main_module:33:47 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}
Node.js v25.0.0
```

■ Terminal 2: Ejecutar pruebas

En la segunda terminal se ejecuto

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

C:\Users\zunig\multilingual-tests>npx playwright test

Running 1 test using 1 worker

X 1 ...localization.spec.js:4:5 > Debe cambiar el idioma dinámicamente (9.0s)
  1) Test\tests\localization.spec.js:4:5 > Debe cambiar el idioma dinámicamente

  Error: page.goto: Protocol error (Page.navigate): Cannot navigate to invalid URL
  Call log:
    - navigating to "/", waiting until "load"

  4 | test('Debe cambiar el idioma dinámicamente', async ({ page }) => {
  5 |   // Abrir la página principal
  6 |   await page.goto('/');
lization.spec.js:6:14
lization.spec.js:6:14
lization.spec.js:6:14
lization.spec.js:6:14
lization.spec.js:6:14
lization.spec.js:6:14
```

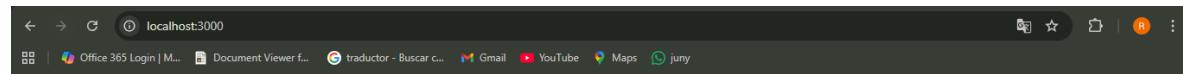
A continuación, los resultados que se tiene en la pagina web:

Ingles



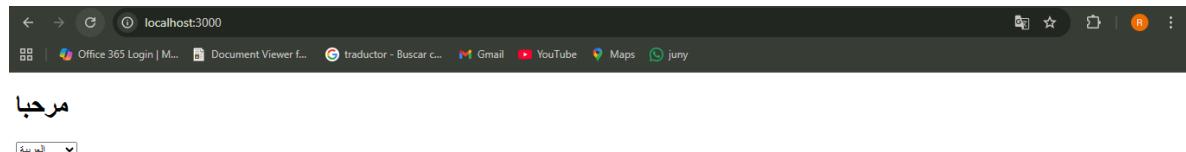
Hello

Español



Hola

Árabe



Configuración del server

A screenshot of a code editor showing a file named 'server.js'. The code is a simple Express.js application that serves files from the 'public' directory and logs a message when it starts on port 3000.

```
node_modules > @playwright > test > server.js > ...
1
2  const express = require('express');
3  const app = express();
4  const PORT = 3000;
5
6  // Servir la carpeta 'public'
7  app.use(express.static('public'));
8
9  // Iniciar el servidor
10 app.listen(PORT, () => {
11   console.log(`Servidor corriendo en http://localhost:${PORT}`);
12 });
13
```

The code editor interface includes tabs for 'index.html', 'server.js', 'en.json', and 'es.json'. Below the code editor is a terminal window showing the command 'node server.js' being run and the output 'Servidor corriendo en http://localhost:3000'.