

Lab 02

Herramienta utilizada:

Para esta ocasión, se utilizo Playwright además de usar Visual Studio code configurado con Maven.

El apartado designado fue “Time”, así que para estas pruebas se crearon los siguientes escenarios:

1. Escenario de prueba 1: Prueba inicial y básica

- Además de hacer el login, comprobar que el botón ‘Time’ lleve a la página correspondiente.
- Corroborar que exista listado
- Probar que el botón View funcione
- Contar botones de forma que se cuenta la creación de los elementos
- Probar botón ‘Edit’
- Probar botón ‘Cancel’

Código implementado para las pruebas:

```
1 //primera prueba boton view del primer listado funcionando
2 // contar botones
3 Locator viewButtons = page.locator("button:has-text('View')");
4 System.out.println("Total botones View encontrados: " + viewButtons.count());
5 page.locator("button:has-text('View')").nth(1).click();
6 page.waitForTimeout(1000); // espera 3 segundos (1000 milisegundos)
7
8
9 //prueba 2 boton edit Funciona
10 page.waitForSelector("button:has-text('Edit')");
11 page.locator("button:has-text('Edit')").click();
12 page.waitForTimeout(1000); // espera 3 segundos (1000 milisegundos)
13
14 // prueba 3 boton cancel en el edit
15 page.waitForSelector("button:has-text('Cancel')");
16 page.locator("button:has-text('Cancel')").click();
17 page.waitForTimeout(1000); // espera 3 segundos (1000 milisegundos)
```

2. Escenario de prueba 2: Prueba de 'Project'

- Comprobar funcionamiento de botón 'Project info'
- Corroborar que se renderice el contenido
- Probar botón de editar del icono de lápiz
- Probar botón 'Add'
- Corroborar que se genere el modal al momento de darle click a dicho botón
- Comprobar y probar el input y su escritura
- Probar a guardar con el botón 'Save'
- Luego de abre creado un Add, intentar eliminarlo al seleccionar
- Seleccionar los botones
- Seleccionar 'Delete'

Código implementado para la prueba:

```
1 // escenario 2
2 // Paso 1. Click en "Project Info"
3 page.locator("li:has-text('Project Info')").click();
4
5 // Paso 2. Esperar y hacer clic en "Projects"
6 page.waitForSelector("a:has-text('Projects')");
7 page.locator("a:has-text('Projects')").click();
8
9 // Paso 3. Esperar a que cargue la URL de proyectos
10 page.waitForURL("**/time/viewProjects");
11
12 // Paso 4. Espera adicional para renderizado
13 page.waitForTimeout(3000);
14
15 // Paso 5. Clic en boton de editar (icono lápiz)
16 page.locator("button:has(i.oxd-icon.bi-pencil-fill)").first().click();
17
18 // Paso 6. Clic en boton "Add"
19 page.waitForSelector("button:has(i.oxd-icon.bi-plus)");
20 page.locator("button:has-text('Add')").nth(2).click();
21
22 // Paso 7. Esperar que el modal este visible
23 Locator modal = page.locator("div.orangehrm-dialog-modal");
24 modal.waitFor(new Locator.WaitForOptions().setState(WaitForSelectorState.VISIBLE));
25
26 // Input dentro del modal
27 Locator inputNuevo = modal.locator("input.oxd-input--active").first();
28 inputNuevo.waitFor(new Locator.WaitForOptions().setState(WaitForSelectorState.VISIBLE));
29
30 // Hacer doble click para asegurar foco y seleccion
31 inputNuevo.dblclick();
32 page.waitForTimeout(2000);
33 page.locator("input.oxd-input--focus").first().fill("prueba 2");
34 page.waitForTimeout(3000);
35
36 // Paso 9. Click en boton "Save" dentro del modal
37 modal.locator("button:has-text('Save')").click();
38
39 // Paso 10. Click en checkbox correspondiente
40 page.locator("span.oxd-checkbox-input").first().click();
41
42 // Paso 13. Esperar 3 segundos
43 page.waitForTimeout(3000);
44 // Paso 11. Click en botón "Delete Selected"
45 page.locator("button:has-text('Delete Selected')").click();
46
47 // Paso 13. Esperar 3 segundos
48 page.waitForTimeout(3000);
49 // Paso 12. Click en boton de confirmacion "Yes, Delete"
50 page.locator("button:has-text('Yes, Delete')").click();
51
52 // Paso 13. Esperar 3 segundos
53 page.waitForTimeout(3000);
```

3. Escenario 3: Agregar a la lista y verificar

- Comprobar el funcionamiento del botón Attendance
- Prueba de botón 'Employee Records' para ver la lista
- Prueba de botón 'View' en la lista
- Prueba de botón 'Add'
- Prueba de escribir texto en el input
- Al momento de crear, dependiendo del estado de la lista de muestra un botón 'In' o 'Out', así que para esto se hizo un caso en el que se verifique cual de los dos esta presente y se haga click en ambos para probar ambos
- Luego de esperar, verificar la creación del elemento en la lista

Código implementado para la prueba:

```
1 // escenario 3
2 // --- Paso 1: Click en Attendance (abre dropdown) ---
3 page.waitForSelector("span.oxd-topbar-body-nav-tab-item:has-text('Attendance')").click();
4
5 // --- Paso 2: Click en "Employee Records" ---
6 page.waitForSelector("a.oxd-topbar-body-nav-tab-link:has-text('Employee Records')").click();
7
8 // --- Paso 3: Esperar a que cargue Employee Records ---
9 page.waitForURL("*/attendance/viewAttendanceRecord");
10
11 // --- Paso 4: Click en boton "View" (primero de la lista) ---
12 page.locator("button:has-text('View')").nth(1).click();
13
14 // --- Paso 5: Esperar que cargue la nueva pagina ---
15 page.waitForLoadState();
16
17 // --- Paso 6: Click en boton "Add" ---
18 page.waitForSelector("button:has-text('Add')").click();
19
20 // --- Paso 7: Esperar que cargue el formulario ---
21 page.waitForLoadState();
22
23 // --- Paso 8: Escribir texto en textarea ---
24 Locator textarea = page.locator("textarea[placeholder='Type here']");
25 textarea.fill("Este es un escenario de prueba");
26
27 // --- Paso 9: Click en boton "In" ---
28 if (page.locator("button:has-text('In')").isVisible()) {
29     page.locator("button:has-text('In')").click();
30     System.out.println("Clic en 'In'");
31 } else if (page.locator("button:has-text('Out')").isVisible()) {
32     page.locator("button:has-text('Out')").click();
33     System.out.println("Clic en 'Out'");
34 } else {
35     System.out.println("No se encontro boton 'In' ni 'Out'");
36 }
37
38 // --- Paso 10: Esperar hasta que se muestre la fila con el texto ---
39 try {
40     // Esperar explicitamente hasta que aparezca la fila deseada (timeout: 10s)
41     page.waitForSelector(
42         "div.oxd-table-row.oxd-table-row--with-border[role='row']:has-text('Este es un escenario de prueba')",
43         new Page.WaitForSelectorOptions().setTimeout(10000)
44     );
45     System.out.println("El registro esperado existe en la tabla.");
46 } catch (PlaywrightException e) {
47     System.out.println("No se encontro el registro.");
48 }
49
50 }
51 }
52 }
53 }
```

Y al momento de pasar las pruebas automáticas, todas nos muestran que pasaron al momento de ejecutarlas.

```
PS C:\Users\jramo\OneDrive\Escritorio\U\posgrado\testing\lab2> mvn exec:java
WARNING: A terminally deprecated method in sun.misc.Unsafe has been called
WARNING: sun.misc.Unsafe::staticFieldBase has been called by com.google.inject.internal.aop.HiddenClassDefiner (file:/t
3.9.10/lib/guice-5.1.0-classes.jar)
WARNING: Please consider reporting this to the maintainers of class com.google.inject.internal.aop.HiddenClassDefiner
WARNING: sun.misc.Unsafe::staticFieldBase will be removed in a future release
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.ejemplo:playwright-java-demo >-----
[INFO] Building playwright-java-demo 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- exec:3.1.0:java (default-cli) @ playwright-java-demo ---
Página después de login: Dashboard
Página cargada: https://opensource-demo.orangehrmlive.com/web/index.php/time/viewEmployeeTimesheet
Total botones View encontrados: 0
Clic en 'In'
? El registro esperado existe en la tabla.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:52 min
[INFO] Finished at: 2025-06-17T00:14:30-06:00
[INFO] -----
```

Código adjunto al archivo en

src>main>java>program>com>ejemplo>PruebaLogin.java

Se ejecuta como “mvn exec:java”