Comandos de consola para ejecutar Playwright

|  | |
| --- | --- |
| **Detalle** | **Comando** |
| Instalar Playwright | npm init playwright@latest Navegadores: npx playwright install |
| Ejecutar todos los test en la carpeta por defecto. | npx playwright test --headed |
| Ejecutar un test en específico | npx playwright test landing-page.spec.js |
| Ejecutar los scripts en formato gráfico | npx playwright test --ui |
| Ejecutar el generador de código | npx playwright codegen [www.ejemplo.com](http://www.ejemplo.com/) |
| Ejecutar un script con el navegador visible | npx playwright test miprueba.spec.js --headed |
| Ejecutar los test en un navegador en específico | npx playwright test miprueba.spec.js --project=firefox |

1- Instalación de Node.js

<https://nodejs.org/en>

2- Editor VSC

<https://code.visualstudio.com/>

3- Crear una carpeta

4- Instalar Extensión en español de VSC

5- Crear un proyecto de node.js . Desde la terminal de VSC

npm init -y

6- Instalar PlayWright

npm init playwright@latest

—------------------------------------------------------------ 29/01/2025

import { test } from '@playwright/test'; // indicar playwright

test('Ingresar en Mercado Libre', async ({ page }) => {

await page.goto('https://www.mercadolibre.com.ar') // abrí la página de Mercado Libre

await page.locator('#cb1-edit').fill('Linterna') // Escribo Linterna en el cuadro de búsqueda

await page.locator('#cb1-edit').press('Enter') // simular que el usuario digitó Enter

await page.waitForTimeout(6000) // Espera de 6 segundos

});

import { test } from '@playwright/test'; // indicar playwright

test('Ingresar en Mercado Libre', async ({ page }) => {

await page.goto('https://www.mercadolibre.com.ar') // abrí la página de Mercado Libre

await page.locator('#cb1-edit').fill('Linterna') // Escribo

await page.locator('#cb1-edit').press('Enter') // simular que el usuario digitó Enter

//await page.waitForTimeout(6000) // Espera de 6 segundos

});

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 03.02.2025

import { test } from '@playwright/test'

test('Formulario de Usuario y Clave', async ({ page }) => {

await page.goto('https://institutoweb.com.ar/test/login.html')

// escribo el usuario

await page.locator('#tuusuario').fill('Juan Cruz')

// escribo la clave

await page.locator('#tuclave').fill('ClaveSecreta1000')

// escribo un email

await page.locator('#tumail').fill('juanc@gmail.com')

// click sobre el boton 'ingresar'

await page.waitForTimeout(4000)

await page.getByRole('button', { name: 'Ingresar' }).click()

//await page.locator('body > form > button:nth-child(8)').click()

await page.waitForTimeout(4000)

await page.close()

})

video: 'on',

screenshot:'on',

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 05/02/2025

Desafío:

Como se obtienen con playwright los selectores de los botones repetidos:

<https://institutoweb.com.ar/test/Login2.html>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import { test } from '@playwright/test'

test('Formulario de Usuario y Clave', async ({ page }) => {

await page.goto('https://institutoweb.com.ar/test/login.html')

// escribo el usuario

await page.locator('#tuusuario').fill('Juan Cruz')

// escribo la clave

await page.locator('#tuclave').fill('ClaveSecreta1000')

// escribo un email

await page.locator('#tumail').fill('juanc@gmail.com')

// click sobre el boton 'ingresar'

await page.waitForTimeout(5000)

//await page.locator('body > form > button:nth-child(8)').click()

await page.getByRole('button', { name: 'Ingresar' }).click()

// paso a la siguiente pantalla

await page.waitForTimeout(2000)

await page.locator('#volver').click()

await page.waitForTimeout(2000)

await page.close()

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import { test,expect } from '@playwright/test'

test('Formulario de Usuario y Clave', async ({ page }) => {

await page.goto('https://institutoweb.com.ar/test/login.html')

// escribo el usuario

await page.locator('#tuusuario').fill('Juan Cruz')

// escribo la clave

await page.locator('#tuclave').fill('ClaveSecreta1000')

// escribo un email

await page.locator('#tumail').fill('juanc@gmail.com')

// click sobre el boton 'ingresar'

await page.waitForTimeout(5000)

//await page.locator('body > form > button:nth-child(8)').click()

// click en el boton ingresar

await page.getByRole('button', { name: 'Ingresar' }).click()

// paso a la siguiente pantalla

await page.waitForTimeout(2000)

//obtener el texto del encabezado

var texto\_obtenido = await page.locator('body > h3').textContent()

// mostrar el valor de la variable en la terminal

console.log(texto\_obtenido)

// realizar la comparación entre lo recibido y lo esperado

// Aserción de la respuesta CON control de Mayúsculas y minúsculas:

expect(texto\_obtenido).toEqual('Acceso correcto!')

// Aserción de la respuesta SIN control de Mayúsculas y minúsculas:

//expect(texto\_obtenido).toMatch(/Acceso CORRECTO!/i)

await page.locator('#volver').click()

// captura de imagen de pantalla

await page.screenshot({path : 'imagen.png'})

await page.waitForTimeout(2000)

await page.close()

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 10/02/2025

const fs = require('fs') //// fs = file system ->> archivos del sistema

function escribe\_log(escritura){

//obtengo la fecha y hora de la compu actual

**var fecha\_hora = new Date()**

fs.appendFileSync('testing.log',**fecha\_hora** + " - " + escritura+'\r\n')

// Windows: \r\n salto de línea en un archivo de texto

// Linux o Mac: \n

}

import{ expect, test} from '@playwright/test'

test('Grabo datos en un archivo', async ({ page }) =>{

escribe\_log('Inició el test')

await page.goto('https://www.institutoweb.com.ar/test/login.html')

escribe\_log('Cargó la página')

// obtengo el titulo de la página

var titulo = await page.title()

// reviso el título de la web

expect(titulo).toBe('Acceso al sistema')

// incorporo un dato en un archivo

escribe\_log('El título es correcto')

escribe\_log('Terminó el test')

escribe\_log('---------------------------------------------------------')

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 12/02/2025

npx playwright codegen <https://www.institutoweb.com.ar/test/login.html>

import { test, expect } from '@playwright/test';

test('test', async ({ page }) => {

await page.goto('https://www.institutoweb.com.ar/test/login.html');

await page.locator('#tuusuario').fill('MiUsuario');

await page.locator('#tuclave').fill('Micalvesecreta');

await page.locator('#tumail').fill('daniel@yahoo.com.ar');

await page.getByRole('button', { name: 'Ingresar' }).click();

await page.getByRole('link', { name: 'Cerrar Sesión' }).click();

});

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Hooks de playwright

**test.describe**('test', async ({ page }) => {

**test.beforeAll(**){

antes de todos los test

}

**test.beforeEac**h(){

antes del siguiente test (Test Principal)

}

**test**('test de prueba principal', async ({ page }) => {

test.step()

test.step()

}

**test.afterEach**(){

despues del siguiente test (Test Principal)

}

**test.afterAll**(){

page.close()

}

}

});

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import { test, expect } from '@playwright/test';

test.describe('Test de Usuarios', ()=>{

test.beforeEach('Conexion a la web', async ({ page }) => {

await page.goto('https://www.institutoweb.com.ar/test/login.html');

})

test('Test Principal', async ({ page }) => {

await page.locator('#tuusuario').fill('MiUsuario');

await page.locator('#tuclave').fill('Micalvesecreta');

await page.locator('#tumail').fill('daniel@yahoo.com.ar');

await page.getByRole('button', { name: 'Ingresar' }).click();

await page.getByRole('link', { name: 'Cerrar Sesión' }).click();

});

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[

{

"usuario": "shows0",

"clave": "fT2/BB2\"Ttd",

"email": "hdawks0@multiply.com"

},

{

"usuario": "jmaskall1",

"clave": "tX4|\_9vVj|7nL",

"email": "jwiddup1@buzzfeed.com"

},

{

"usuario": "icafferty2",

"clave": "dW1~AqLPVw$nt",

"email": "dglasheen2@marriott.com"

},

{

"usuario": "omohun3",

"clave": "mO4>%`4%rxiFB",

"email": "jwilman3@cornell.edu"

},

{

"usuario": "crottgers4",

"clave": "qO4=kQK{1/sd71",

"email": "lspooner4@sakura.ne.jp"

},

{

"usuario": "wcatterall5",

"clave": "jH6$Oj'!E\\cC",

"email": "csmithin5@is.gd"

},

{

"usuario": "tjostan6",

"clave": "pB9)lTzU/IY",

"email": "lmcowis6@storify.com"

},

{

"usuario": "cpummery7",

"clave": "gF2{8J@B",

"email": "lmarke7@liveinternet.ru"

},

{

"usuario": "crushman8",

"clave": "sE1~Lj&c8@",

"email": "ihacun8@mapy.cz"

},

{

"usuario": "eearey9",

"clave": "nM9=eD1z",

"email": "kkleinzweig9@hexun.com"

}

]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import { test, expect } from '@playwright/test';

import fs from 'fs/promises'

test.describe('Test de Usuarios', ()=>{

var usuarios // creo la variable sin datos para todos los tests

test.beforeAll('Conexión con el archivo Json', async ()=>{

var muestras = await fs.readFile('datos.json') //relacion con archivo de datos

usuarios = JSON.parse(muestras) // los datos de las muestras vienen en formato json

})

test.beforeEach('Conexion a la web', async ({ page }) => {

await page.goto('https://www.institutoweb.com.ar/test/login.html');

})

test('Test Principal', async ({ page }) => {

for(var {usuario,clave,email} of usuarios ){ // separar los 3 datos de usuario, clave y email

await page.locator('#tuusuario').fill(usuario);

await page.locator('#tuclave').fill(clave);

await page.locator('#tumail').fill(email);

await page.waitForTimeout(2000)

await page.getByRole('button', { name: 'Ingresar' }).click();

// aserción de elemento visible en pantalla

await expect(page.getByRole('link', { name: 'Cerrar Sesión' })).toBeVisible()

await page.getByRole('link', { name: 'Cerrar Sesión' }).click();

}

});

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 17.02.2025

import { test, expect } from '@playwright/test';

import fs from 'fs/promises'

test.describe('Test de Usuarios', ()=>{

var usuarios // creo la variable sin datos para todos los tests

test.beforeAll('Conexión con el archivo Json', async ()=>{

var muestras = await fs.readFile('datos.csv') //relacion con archivo de datos

usuarios = muestras.split('\n') // los datos de las muestras vienen en formato csv

usuarios = usuarios.slice(1) // Omitir el primer renglon (encabezado)

})

test.beforeEach('Conexion a la web', async ({ page }) => {

await page.goto('https://www.institutoweb.com.ar/test/login.html');

})

test('Test Principal', async ({ page }) => {

for( var linea of usuarios ){ // separar los 3 datos de usuario, clave y email

var [usuario,clave,email] = linea.split(',')

await page.locator('#tuusuario').fill(usuario);

await page.locator('#tuclave').fill(clave);

await page.locator('#tumail').fill(email);

//await page.waitForTimeout(2000)

await page.getByRole('button', { name: 'Ingresar' }).click();

// aserción de elemento visible en pantalla

await expect(page.getByRole('link', { name: 'Cerrar Sesión' })).toBeVisible()

await page.getByRole('link', { name: 'Cerrar Sesión' }).click();

}

});

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* archivo csv

usuario,clave,email

wbonnell0,nF6=bb5.krD&&{wn,jcolbron0@timesonline.co.uk

rcroy1,mJ2|#F\*J,gaprahamian1@cornell.edu

rshoebridge2,oU4\f({Bq,clintill2@wiley.com

ahylton3,"vK9{wL@D'HX>N\i""",cbedingfield3@w3.org

thastwell4,gD7)t6h~c,ecreighton4@webeden.co.uk

mphifer5,rL6@omYLXIOu%9O,mapplegate5@github.io

jscoterbosh6,tQ1)FR'hINi,aweetch6@studiopress.com

dalchin7,eH7=1@U/,kfittes7@engadget.com

plowndesbrough8,eW1\*65yWQ&,clardier8@businessinsider.com

rharmes9,qX3/kMGbNu+,ggethings9@blogger.com

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import { test, expect } from '@playwright/test';

import fs from 'fs/promises'

test.describe('Test de Usuarios', ()=>{

var usuarios = [];// creo la variable ( para varios renglones ) sin datos para todos los tests

test.beforeAll('Conexión con el archivo CSV', async ()=>{

var muestras = await fs.readFile('datos.csv') //relacion con archivo de datos

usuarios = muestras.toString().split('\n') // los datos de las muestras (to string = lo pasa a texto) vienen en formato csv

usuarios = usuarios.slice(1) // Omitir el primer renglon (encabezado)

})

test.beforeEach('Conexion a la web', async ({ page }) => {

await page.goto('https://www.institutoweb.com.ar/test/login.html');

})

test('Test Principal', async ({ page }) => {

for( var linea of usuarios ){ // separar los 3 datos de usuario, clave y email

var [usuario,clave,email] = linea.split(',')

await page.locator('#tuusuario').fill(usuario);

await page.locator('#tuclave').fill(clave);

await page.locator('#tumail').fill(email);

//await page.waitForTimeout(2000)

await page.getByRole('button', { name: 'Ingresar' }).click(**{timeout:5000}**);

// aserción de elemento visible en pantalla

await expect(page.getByRole('link', { name: 'Cerrar Sesión' })).toBeVisible()

await page.getByRole('link', { name: 'Cerrar Sesión' }).click();

}

});

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

timeout:50000,

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Test de prueba try - catch

import { test, expect } from '@playwright/test';

test('Test Principal', async ({ page }) => {

await page.goto('https://example.com')

try{ // probá el siguiente código

//await page.locator('#boton-inexistente').click({timeout:1000});

await page.click('#boton-inexistente',{timeout:1000})

console.log('Click realizado correctamente')

}catch{ // código a ejecutar si hay un error

console.log('Existió un error')

}

console.log('Continúa el test')

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 19.02.2025

Práctica

1- Codegen

npx playwright codegen [www.saucedemo.com](http://www.saucedemo.com)

Test case 001 :

import { test, expect } from '@playwright/test';

test('Test de usuario válido', async ({ page }) => {

await page.goto('https://www.saucedemo.com/');

/////////////////////////// test de usuario válido

await page.locator('[data-test="username"]').fill('standard\_user');

await page.locator('[data-test="password"]').fill('secret\_sauce');

await page.locator('[data-test="login-button"]').click();

await page.getByRole('button', { name: 'Open Menu' }).click();

await page.locator('[data-test="logout-sidebar-link"]').click();

});

Test case 002:  
import { test, expect } from '@playwright/test';

test('Test de usuario inválido', async ({ page }) => {

await page.goto('https://www.saucedemo.com/');

/////////////////////////// test de usuario inválido

await page.locator('[data-test="username"]').fill('stan');

await page.locator('[data-test="password"]').fill('sakdjajdskj');

await page.locator('[data-test="login-button"]').click();

// mensaje de error de usuario no encontrado

var texto\_error = await page.locator('[data-test="error"]').textContent();

// revisión del texto obtenido

expect(texto\_error).toContain('Username and password do not match')

await page.locator('[data-test="error-button"]').click();

});

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

npx playwright test --headed --project=chromium

opción: –debug

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 24.02.2025

import { test, expect } from '@playwright/test';

test('Test Visual de Página Web', async ({ page }) => {

await page.goto('https://www.institutoweb.com.ar/test/login.html')

//await page.waitForTimeout(3000)

expect(await page.screenshot()).toMatchSnapshot('test-imagen.png')

});

// ejacutar el test por primera vez para crear las imágenes "maestras"

// **npx playwright test visualtest.spec.js --update-snapshots**

// para probar las imágenes y comparar las actuales con las "maestras"

// **npx playwright test visualtest.spec.js**

**/////////////////////////////////// Datos del servidor de base de datos de Saucedemo**

**url de acceso:** [**https://cursotesting.com.ar/phpmyadmin/**](https://cursotesting.com.ar/phpmyadmin/)

**usuario: testing**

**clave: cursoperformance**

**base de datos: productos**

**tabla: ecommerce**

**///////////////////////////////////////////////// conexion.js**

const mysql = require('mysql2')

/// creando la conexión con la base de datos

const connection = mysql.createConnection({

host:'cursotesting.com.ar',

user: 'testing',

password: 'cursoperformance',

database: 'productos'

})

// Probar la conexión

connection.connect(err => {

if(err){

console.log("Error al conectarse a la base de datos")

return

}else{

console.log('Conexión exitosa!')

}

/////////////// búsqueda del precio a desarrollar

var busqueda = "SELECT precio FROM ecommerce WHERE nombre = 'Sauce Labs Bolt T-Shirt'"

/////////////// Ejecución de la búsqueda con query

connection.query(busqueda, (err,results) => {

if(results.length >0 ){ // largo del resultado mayor que cero

var mi\_precio = results[0].precio

console.log(mi\_precio)

}

})

connection.end();

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 26.02.2025

import { test, expect } from '@playwright/test';

const fs = require('fs') //// fs = file system ->> archivos del sistema

test('test', async ({ page }) => {

// ir a la web de saucedemo

await page.goto('https://www.saucedemo.com/');

await page.locator('[data-test="username"]').fill('error\_user');

await page.locator('[data-test="password"]').fill('secret\_sauce');

await page.locator('[data-test="login-button"]').click();

//// ingresamos a ver los productos

// adiciono al carrito de compras el producto 1

await page.locator('[data-test="add-to-cart-sauce-labs-backpack"]').click();

// click sobre el ícono del carrito de compras

await page.locator('[data-test="shopping-cart-link"]').click();

// click en boton checkout

await page.locator('[data-test="checkout"]').click();

// ingreso datos personales

await page.locator('[data-test="firstName"]').fill('Daniel');

await page.locator('[data-test="lastName"]').fill('Balbi');

await page.locator('[data-test="postalCode"]').fill('1404');

await page.locator('[data-test="continue"]').click();

// obtengo el texto del nombre del producto

var nombre = await page.locator('[data-test="item-4-title-link"]').textContent()

// obtengo el importe a facturar por el ecommerce por el producto

var precio = await page.locator('[data-test="inventory-item-price"]').textContent()

console.log("El producto es:" + nombre)

console.log("El precio es:" + precio)

// Finalizar la compra

await page.locator('[data-test="finish"]').click();

// Hacer click en el boton volver a la página principal

try{

await page.locator('[data-test="back-to-products"]').click({timeout:2000});

}catch{

var fecha\_hora = new Date()

fs.appendFileSync('testing.log',fecha\_hora + " - " + "Falló el boton Back to Products"+'\r\n')

}

// click para abrir el menu

await page.getByRole('button', { name: 'Open Menu' }).click();

// click para hacer un logout

await page.locator('[data-test="logout-sidebar-link"]').click();

});

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 03.03.2025

**/// Ejemplo de búsqueda en BD sin contemplar el async**

**// TC-003**

import { test, expect } from '@playwright/test';

const fs = require('fs') //// fs = file system ->> archivos del sistema

const mysql = require('mysql2') /// conexion con la base de datos

// configuro la conexión

const connection = mysql.createConnection({

host: 'cursotesting.com.ar',

user: 'testing',

password:'cursoperformance',

database:'productos'

})

// ahora pruebo de conectarme

connection.connect(err => {

if(err){

console.log('Error en la conexión',err)

return

}

})

var precio\_comparativo

test('test', async ({ page }) => {

// ir a la web de saucedemo

await page.goto('https://www.saucedemo.com/');

await page.locator('[data-test="username"]').fill('standard\_user');

await page.locator('[data-test="password"]').fill('secret\_sauce');

await page.locator('[data-test="login-button"]').click();

//// ingresamos a ver los productos

// adiciono al carrito de compras el producto 1

await page.locator('[data-test="add-to-cart-sauce-labs-backpack"]').click();

// click sobre el ícono del carrito de compras

await page.locator('[data-test="shopping-cart-link"]').click();

// click en boton checkout

await page.locator('[data-test="checkout"]').click();

// ingreso datos personales

await page.locator('[data-test="firstName"]').fill('Daniel');

await page.locator('[data-test="lastName"]').fill('Balbi');

await page.locator('[data-test="postalCode"]').fill('1404');

await page.locator('[data-test="continue"]').click();

// obtengo el texto del nombre del producto

var nombre\_obtenido = await page.locator('[data-test="item-4-title-link"]').textContent()

// obtengo el importe a facturar por el ecommerce por el producto

var precio = await page.locator('[data-test="inventory-item-price"]').textContent()

const busqueda = "SELECT precio from ecommerce where nombre='" + nombre\_obtenido +"'"

// Conexión que no respeta el async de playwright

**connection.query(busqueda, (err,results)=>{**

**if(results.length > 0) {**

**precio\_comparativo = results[0].precio**

**}**

**})**

console.log(busqueda)

console.log("El precio de la base de datos es" + precio\_comparativo)

console.log("El producto es:" + nombre\_obtenido)

console.log("El precio es:" + precio)

// Finalizar la compra

await page.locator('[data-test="finish"]').click();

// Hacer click en el boton volver a la página principal

try{

await page.locator('[data-test="back-to-products"]').click({timeout:2000});

}catch{

var fecha\_hora = new Date()

fs.appendFileSync('testing.log',fecha\_hora + " - " + "Falló el boton Back to Products"+'\r\n')

}

// click para abrir el menu

await page.getByRole('button', { name: 'Open Menu' }).click();

// click para hacer un logout

await page.waitForTimeout(2000)

await page.locator('[data-test="logout-sidebar-link"]').click();

**console.log("El test del precio n° 1: " + precio\_comparativo)**

//saco el símbolo $ reemplazandolo por ""

var nuevo\_precio = precio.replace("$","")

console.log("Precio sin el signo $ :" +nuevo\_precio)

**expect(nuevo\_precio).toBe(precio\_comparativo)**

**});**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Código corregido de una busqueda en BD con acompañamiento del async**

**\*\*\*\*\* TC-004**

import { test, expect } from '@playwright/test';

**import fs from 'fs' //// fs = file system ->> archivos del sistema**

**import mysql from 'mysql2/promise' /// conexion con la base de datos**

// configuro la conexión

**const connectionConfig = ({**

**host: 'cursotesting.com.ar',**

**user: 'testing',**

**password:'cursoperformance',**

**database:'productos'**

**})**

var precio\_comparativo

test('test', **async** ({ page }) => {

**// conexión asincrónica para la BD**

**const connection = await mysql.createConnection(connectionConfig)**

// ir a la web de saucedemo

await page.goto('https://www.saucedemo.com/');

await page.locator('[data-test="username"]').fill('standard\_user');

await page.locator('[data-test="password"]').fill('secret\_sauce');

await page.locator('[data-test="login-button"]').click();

//// ingresamos a ver los productos

// adiciono al carrito de compras el producto 1

await page.locator('[data-test="add-to-cart-sauce-labs-backpack"]').click();

// click sobre el ícono del carrito de compras

await page.locator('[data-test="shopping-cart-link"]').click();

// click en boton checkout

await page.locator('[data-test="checkout"]').click();

// ingreso datos personales

await page.locator('[data-test="firstName"]').fill('Daniel');

await page.locator('[data-test="lastName"]').fill('Balbi');

await page.locator('[data-test="postalCode"]').fill('1404');

await page.locator('[data-test="continue"]').click();

// obtengo el texto del nombre del producto

var nombre\_obtenido = await page.locator('[data-test="item-4-title-link"]').textContent()

// obtengo el importe a facturar por el ecommerce por el producto

var precio = await page.locator('[data-test="inventory-item-price"]').textContent()

const busqueda = "SELECT precio from ecommerce where nombre='" + nombre\_obtenido +"'"

**const [results] = await connection.execute(busqueda)**

**precio\_comparativo = results[0].precio**

console.log("El test del precio n° 1: " + precio\_comparativo)

console.log(busqueda)

console.log("El precio de la base de datos es" + precio\_comparativo)

console.log("El producto es:" + nombre\_obtenido)

console.log("El precio es:" + precio)

//saco el símbolo $ reemplazandolo por ""

**var nuevo\_precio = precio.replace("$","")**

console.log("Precio sin el signo $ :" +nuevo\_precio)

////

**expect(nuevo\_precio).toBe(precio\_comparativo)**

// Finalizar la compra

await page.locator('[data-test="finish"]').click();

// Hacer click en el boton volver a la página principal

try{

await page.locator('[data-test="back-to-products"]').click({timeout:2000});

}catch{

var fecha\_hora = new Date()

fs.appendFileSync('testing.log',fecha\_hora + " - " + "Falló el boton Back to Products"+'\r\n')

}

// click para abrir el menu

await page.getByRole('button', { name: 'Open Menu' }).click();

// click para hacer un logout

await page.locator('[data-test="logout-sidebar-link"]').click();

});

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 10.03.2025

**/////////////////////////////////// Datos del servidor de base de datos de Api’s**

**url de acceso:** [**https://cursotesting.com.ar/phpmyadmin/**](https://cursotesting.com.ar/phpmyadmin/)

**usuario: testing**

**clave: cursoperformance**

**base de datos: testing**

**tabla: personas**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* consumo api token**

import { test, expect, request } from '@playwright/test'

test('Api Pido Token', async ({ page }) => {

// contexto de valores de la pagina ( cookie, valor guardado )

const contexto = page.context()

// solicitud a una api

const request = contexto.request

// los datos de envío de la clave para recibir el token

const pidotoken = {

username: "institutoweb",

password: "cursoperformance"

}

// Consumir la Api y recibir una respuesta

const response =

await request.post('https://cursotesting.com.ar:3000/token',

{ data: JSON.stringify(pidotoken),

headers : {'Content-Type':'application/json'}

}

)

const respuestaJson = await response.json()

const mitoken = respuestaJson.token

console.log("Token Recibido: ",mitoken)

expect(mitoken).not.toBe(undefined)

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 12.03.2025

Timeouts, ejemplos:

page.visit("www....",{timeout:2000})

page.locator("miboton").click({timeout:10000})

page.locator("miusuario").fill("Miclave",{timeout : 1500})

**page.waitForTimeout(15000)** → no es dinámico

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import { test, expect, request } from '@playwright/test'

// Definir los datos que contendra el pidotoken

interface Claves {

username : String;

password : String;

}

**interface MiToken {**

**token : String;**

**}**

test('Api Pido Token', async ({ page }) => {

// contexto de valores de la pagina ( cookie, valor guardado )

const contexto = page.context()

// solicitud a una api

const request = contexto.request

// los datos de envío de la clave para recibir el token

const pidotoken : Claves = {

username: "institutoweb",

password: "cursoperformance"

}

// Consumir la Api y recibir una respuesta

const response =

await request.post('https://cursotesting.com.ar:3000/token',

{ data: JSON.stringify(pidotoken),

headers : {'Content-Type':'application/json'}

}

)

const respuestaJson **: MiToken**= await response.json()

const mitoken: **String** = respuestaJson.token

console.log("Token Recibido: ",mitoken)

expect(mitoken).not.toBe(undefined)

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Alta de persona

import { test, expect, request, APIRequestContext } from '@playwright/test'

// alta de una persona en la base de datos

// Definir los datos que contendra el pidotoken

interface Persona {

nombre : String;

apellido : String;

edad: number;

pais: String;

}

test('Alta de una persona', async ({ }) => {

// contexto de trabajo independiente de un navegador

// check de tipo de datos Api Context.-

const apiContext: APIRequestContext = await request.newContext()

// Datos de la persona a dar de alta

const nuevaPersona: Persona= {

nombre : "Juan",

apellido : "Otero",

edad: 39,

pais: "Colombia",

}

// Enviar la petición POST

const response =

await apiContext.post('https://cursotesting.com.ar:3000/personas/nueva/', {

data: nuevaPersona

})

// aserción del código de respuesta

expect(response.status()).toBe(200)

const responseBody : string = await response.text()

console.log("Mensaje obtenido: " + responseBody)

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Configuración de un proyecto exclusivo de api en el config:

name: 'api',

use: {},

},

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 17.03.2025

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Manejo de pestañas adicionales

import { test, expect } from '@playwright/test';

test('Inicio apertura solapa', async ({ page }) => {

await page.goto('https://institutoweb.com.ar/test/pop/solapa.html');

// Abrir una nueva solapa / pestaña

// espera a que se abra una nueva solapa

const nuevaSolapa = page.context().waitForEvent('page')

// Click en el enlace que abre la nueva solapa

await page.click('#mi-enlace')

await page.waitForTimeout(5000)

// nueva page en base al resultado del click en el enlace

const nuevaPage = await nuevaSolapa

// Controlar cuando se termino la carga de la nuevaPage

await nuevaPage.waitForLoadState('domcontentloaded')

const la\_url = nuevaPage.url()

expect(la\_url).toBe('https://institutoweb.com.ar/test/pop/popover.html')

console.log(la\_url)

// click en el boton de la nueva pestaña creada por el click('#mi-enlace') que abre el popover

await nuevaPage.click('#mi-boton')

await page.waitForTimeout(5000)

await nuevaPage.click('#close-popover')

await page.waitForTimeout(5000)

await nuevaPage.close()

await page.waitForTimeout(5000)

});

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import { test, expect } from '@playwright/test';

test('Inicio apertura alerta', async ({ page }) => {

await page.goto('https://institutoweb.com.ar/test/pop/alerta.html');

////////////////////////////Espera la alerta////////////////////////////////////////////

// espera que la alerta aparezca

const alertaPromesa = new Promise((resolve) => {

// capturar acciones paralelas del navegador

page.on('dialog', async (dialog)=>{

// manejar los datos y controles de la alerta

console.log(dialog.message())

await page.waitForTimeout(5000)

await dialog.accept()

})

}

)

//////////////////////////////////////////////////////////////////////////

// Hago click sobre el botón que convoca a la alerta

await page.locator('#mi-boton').click()

await page.waitForTimeout(5000)

});

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* POM

import { test, expect } from '@playwright/test';

import {multiplica, suma } from './matematica.spec'

test('Calculadora', async ({ page }) => {

const a:number = 10;

const b:number = 25;

const c:number = 25;

console.log(suma(a,b))

console.log(multiplica(c,b))

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* matematica.spec.ts

export function suma(param1:number,param2:number):number{

return param1 + param2

}

export function multiplica(param1:number,param2:number):number{

return param1 \* param2

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 19.03.2025

import { test } from '@playwright/test'

test('Formulario de Usuario y Clave', async ({ page }) => {

await page.goto('https://institutoweb.com.ar/test/login.html')

// escribo el usuario

await page.locator('#tuusuario').fill('Juan Cruz')

// escribo la clave

await page.locator('#tuclave').fill('ClaveSecreta1000')

// escribo un email

await page.locator('#tumail').fill('juanc@gmail.com')

// click sobre el boton 'ingresar'

await page.waitForTimeout(4000)

await page.getByRole('button', { name: 'Ingresar' }).click()

//await page.locator('body > form > button:nth-child(8)').click()

await page.waitForTimeout(4000)

await page.close()

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Archivo principal en carpeta test

\*\*\* TestLogin.spec.ts

import { test } from '@playwright/test'

import { selector } from '../selectores/mis\_selectores'

import { datos } from '../datos/mis\_datos'

test('Formulario de Usuario y Clave', async ({ page }) => {

// Base Url ubicada en el archivo de configuración

await page.goto('/test/login.html')

// escribo el usuario

// Los selectores se encuentran en el archivo selectores.ts en la carpeta selectores

// login refiere a los selectores de la pantalla login.html

await page.locator(selector.login\_usuario).fill(datos.datos\_usuario)

// escribo la clave

await page.locator(selector.login\_clave).fill(datos.datos\_clave)

// escribo un email

await page.locator(selector.login\_email).fill(datos.datos\_email)

// click sobre el boton 'ingresar'

await page.waitForTimeout(4000)

await page.getByRole('button', { name: selector.login\_btn\_ingresar }).click()

//await page.locator('body > form > button:nth-child(8)').click()

await page.waitForTimeout(4000)

await page.close()

})

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* en carpeta selectores

\*\*\*\* selectores.ts

export interface SelectoresLogin{

login\_usuario : string;

login\_clave : string;

login\_email :string;

login\_btn\_ingresar : string;

}

export const selector : SelectoresLogin = {

login\_usuario : '#tuusuario',

login\_clave : '#tuclave',

login\_email :'#tumail',

login\_btn\_ingresar : 'Ingresar'

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* en carpeta datos

\*\*\*\* datos.ts

export interface DatosLogin{

datos\_usuario : string;

datos\_clave : string;

datos\_email :string;

}

export const datos : DatosLogin = {

datos\_usuario : 'Juan Cruz',

datos\_clave : 'ClaveSecreta1000',

datos\_email :'juanc@gmail.com'

}