**Prueba Sofka - Automatización de Pruebas**

Este repositorio contiene una suite de pruebas automatizadas desarrolladas para la evaluación técnica de Sofka Technologies, utilizando Playwright para testing end-to-end.

**## 🚀 Descripción**

Suite de automatización que incluye pruebas de e-commerce, validación de formularios y flujos completos de usuario utilizando tecnologías modernas de testing.

**## 🛠️ Tecnologías Utilizadas**

- **\*\*[Playwright](**https://playwright.dev/**)\*\*** - Framework de automatización web

- **\*\*JavaScript/Node.js\*\*** - Lenguaje de programación

- **\*\*GitHub Actions\*\*** - CI/CD (opcional)

**## 📋 Prerequisitos**

Antes de ejecutar las pruebas, asegúrate de tener instalado:

- [Node.js](https://nodejs.org/) (versión 14 o superior)

- [Git](https://git-scm.com/)

**## 🔧 Instalación**

1. **\*\*Clonar el repositorio:\*\***

   ```bash

   git clone https://github.com/ycuestah/Prueba-Sofka.git

   cd Prueba-Sofka

   ```

2. **\*\*Instalar dependencias:\*\***

   ```bash

   npm install

   ```

3. **\*\*Instalar navegadores de Playwright:\*\***

   ```bash

   npx playwright install

   ```

**## 🎮 Ejecución de Pruebas**

**### Ejecutar todas las pruebas**

```bash

npx playwright test

```

**### Ejecutar una prueba específica**

```bash

npx playwright test ecommerce-test.spec.js

```

**### Ejecutar pruebas en modo debug**

```bash

npx playwright test --debug

```

**### Ejecutar pruebas con interfaz gráfica**

```bash

npx playwright test --ui

```

**## 📁 Estructura del Proyecto**

Prueba-Sofka/

├── tests/

│   ├── ecommerce-test.spec.js    # Pruebas de e-commerce

│   └── ...                       # Otras pruebas

├── playwright.config.js          # Configuración de Playwright

├── package.json                  # Dependencias del proyecto

└── README.md                     # Documentación

**🧪 Casos de Prueba Incluidos**

**### E-commerce Test Suite**

- ✅ Navegación por catálogo de productos

- ✅ Añadir productos al carrito

- ✅ Proceso completo de checkout

- ✅ Validación de formularios

- ✅ Confirmación de compra

**\*\*Sitio de prueba:\*\*** [DemoBlaze](https://www.demoblaze.com/)

**\*\*Flujo de prueba:\*\***

1. Selección de dos productos diferentes

2. Adición al carrito con manejo de diálogos

3. Proceso de checkout con datos de prueba

4. Finalización exitosa de la compra

**## 📊 Reportes**

Los reportes de ejecución se generan automáticamente en:

- `test-results/` - Screenshots y videos de fallos

- `playwright-report/` - Reporte HTML detallado

Para visualizar el reporte:

```bash

npx playwright show-report

**🔍 Datos de Prueba**

Los tests utilizan los siguientes datos de ejemplo:

| Campo | Valor |

|-------|-------|

| Nombre | Yas |

| País | Uruguay |

| Tarjeta de Crédito | 1234567898966355 |

| Mes | 12 |

| Año | 2025 |

**## 📈 Configuración Avanzada**

**### Configurar diferentes navegadores**

```javascript

// playwright.config.js

use: {

  browserName: 'chromium', // 'firefox', 'webkit'

  headless: false,         // true para modo headless

  viewport: { width: 1280, height: 720 },

}

```

**### Timeouts personalizados**

```javascript

test.setTimeout(60000); // 60 segundos

```

**## 🐛 Troubleshooting**

**### Problemas comunes:**

1. **\*\*Error: Cannot find module '@playwright/test'\*\***

   ```bash

   npm install @playwright/test

   ```

2. **\*\*Navegadores no instalados\*\***

   ```bash

   npx playwright install

   3. **\*\*Timeouts en elementos\*\***

   - Verificar selectores

   - Aumentar timeout si es necesario

   - Usar `waitForSelector()` para elementos dinámicos

**## 📋 Checklist de Pruebas**

- [ ] Todas las pruebas pasan localmente

- [ ] Código sigue las convenciones establecidas

- [ ] Documentación actualizada

- [ ] Screenshots/videos de evidencia incluidos

**## 👤 Autor**

Yadriel Cuesta

- GitHub: [@ycuestah](https://github.com/ycuestah)

- LinkedIn: <https://www.linkedin.com/in/yadrielcuesta/>

---