

# Informe de Automatización FE

## Estrategia de Automatización:

El proyecto utiliza Playwright como framework principal para pruebas automatizadas de frontend, implementado en TypeScript. Se aplica el patrón Page Object Model (POM) para organizar la lógica de interacción con las páginas, facilitando la reutilización y el mantenimiento del código. Los escenarios de prueba están escritos en lenguaje Gherkin, permitiendo una definición clara y legible de los casos de prueba orientados a negocio. La ejecución de pruebas se realiza mediante scripts de npm, con soporte para ejecución por tags y entornos personalizados. El workflow de CI/CD está integrado en GitHub Actions, asegurando la ejecución automática de pruebas en cada push y permitiendo la ejecución manual desde la interfaz de GitHub.

## Patrones Utilizados:

- **Page Object Model (POM):** Organización de clases de páginas para encapsular la lógica de interacción y mejorar la mantenibilidad.
- **Gherkin:** Definición de escenarios de prueba en formato legible por humanos, facilitando la colaboración entre QA y negocio.
- **Data Driven Testing:** Uso de archivos JSON para parametrizar escenarios y ampliar la cobertura sin duplicar código.
- **Separación de Escenarios:** Los escenarios de prueba están organizados por funcionalidades y resultados esperados.
- **Reporte Automatizado:** El workflow sube automáticamente los reportes de ejecución generados por Playwright como artefactos.
- **Configuración por Entorno:** Variables de entorno gestionadas en archivos dedicados, permitiendo la ejecución flexible en diferentes ambientes.

## Ventajas:

- Escalabilidad y fácil mantenimiento gracias a la estructura modular y el uso de POM.
- Definición clara de pruebas con Gherkin, facilitando la comunicación con el negocio.
- Ejecución automatizada y continua mediante GitHub Actions.
- Trazabilidad y generación de reportes accesibles.
- Flexibilidad para incorporar nuevos casos de prueba y adaptar la automatización a distintos entornos.

## Conclusión:

Este enfoque garantiza calidad, rapidez en la validación de funcionalidades frontend y facilidad para incorporar nuevos escenarios de prueba, contribuyendo a la mejora continua del producto y la eficiencia del equipo de QA.