

## Introducción

Cypress es una herramienta de pruebas de interfaz de usuario de última generación diseñada para la web moderna. Con Cypress, podemos crear fácilmente pruebas para nuestras aplicaciones web modernas, depurarlas visualmente y ejecutarlas automáticamente en nuestras compilaciones de integración continua. Esta herramienta aborda los principales problemas que enfrentan los desarrolladores e ingenieros de control de calidad al probar aplicaciones modernas.

Cypress simplifica la configuración de pruebas, permitiendo a los usuarios comenzar a escribir pruebas rápidamente. Ofrece una API amigable y potente para escribir pruebas, lo que facilita la creación de casos de prueba robustos y comprensibles. Las pruebas pueden ser ejecutadas automáticamente, tanto en modo interactivo como en integración continua. Además, proporciona herramientas visuales para depurar pruebas, incluyendo la capacidad de ver el estado de la aplicación en tiempo real.

Cypress se compara con Selenium con mucha frecuencia, pero Cypress es diferente tanto en lo fundamental como en la arquitectura. A diferencia de Selenium, Cypress no está sujeto a las mismas restricciones, lo que permite escribir pruebas más rápidas, fiables y confiables.

En este Taller, lo guiaremos a través de todo lo necesario para comenzar a utilizar Cypress eficazmente, desde la configuración inicial hasta la escritura y depuración de pruebas complejas.

## Objetivos del Taller

---

Este Taller enseña a crear un framework de automatización desde cero, abordando desde conceptos básicos hasta ejecución en paralelo y generación de reportes con Allure. Se enfoca en la resolución de problemas comunes en la automatización de pruebas, combinando teoría y práctica con ejemplos reales. Incluye instalación, selectores, assertions, ciclo de vida de pruebas, comandos personalizados y ejecución en entornos remotos. Además, cubre integración con herramientas externas, debugging y un proyecto final para aplicar los conocimientos adquiridos.

2

## Contenidos

---

### Introducción general

- El testin y su necesidad en la actualidad
- Características de Cypress
- Documentación

### Introducción a Cypress

- Visión general de Cypress
- Instalación y configuración inicial y primer caso de pruebas
- Configuración del entorno
- Primer caso de prueba
- Selectores de Cypress
- Tipos de selectores
- Buenas prácticas en el uso de selectores

### Assertions en Cypress

- Tipos de assertions
- Uso de assertions en pruebas
- Interacción con elementos
- Click, type y check
- Manejo de formularios y eventos

### Comandos personalizados (Custom commands)

- Creación y uso de comandos personalizados
- Ejemplos prácticos
- Logs
- Configuración y uso de logs en Cypress
- Mejores prácticas para el registro de pruebas
- Clases, métodos estáticos y getters
- Uso de clases en Cypress
- Métodos estáticos y getters para mejorar la estructura de pruebas

## Automatización de pruebas: paso a paso con ejemplos

- Ejemplos prácticos de automatización
- Implementación de pruebas end-to-end (E2E)
- Automatización con código usando Cypress
- Integración de Cypress con frameworks de desarrollo
- Ejemplos avanzados de automatización

## Ejemplo de prueba en Cypress

- Desglose y explicación de un ejemplo completo
- Pruebas E2E (End-to-End)
- Conceptos y estrategias para pruebas E2E
- Implementación de pruebas integrales

## Organización de pruebas con Hooks y Debugging

- Uso de hooks (before, after, beforeEach, afterEach)
- Técnicas de debugging y resolución de problemas
- Generación de evidencias en Cypress
- Captura de pantallas y videos
- Reportes de pruebas con Allure y otras herramientas

## Proyecto Final

- Configuración del proyecto
- Casos de pruebas manuales
- Generador de selectores
- Page Object Model (POM)
- Logger
- Automatizando pruebas
- Análisis y corrección de fallos
- Mejoras
- Ejecutando una prueba a la vez
- Ejecución en paralelo
- Integración continua

## Cypress Hooks

- Detalle de hooks en Cypress
- Aplicaciones avanzadas de hooks

## Inteligencia Artificial

- Acompañamiento con IA para la resolución específica de temas del Taller

## Metodología

---

- Taller práctico (80% Práctico 20% Teórico).
- Este Taller está diseñado para ofrecer una formación integral en Cypress, combinando teoría y práctica de manera equilibrada, y promoviendo la participación y el aprendizaje colaborativo entre los participantes.

## Requisitos

---

Conocimientos en:

- HTML5
- JavaScript ES6 (ECMAScript6)
- JSON (JavaScript Object Notation)
- Node.js
- NPM (Node Package Manager)
- API RESTful (consumo)

## Dirigido a:

---

Desarrolladores de Aplicaciones

## Generalidades

---

- Se aplicará un cuestionario previo al inicio del taller a cada participante para evaluar su nivel de conocimiento. Esto permitirá identificar la necesidad de nivelación y/o realizar ajustes en el temario si es necesario.
- Duración 20 horas cronológicas.
- Taller cerrado.