



# TEST AUTOMATION ENGINEER – FORMACIÓN INTEGRAL

{desafío}  
*latam\_*

**CURSO:**

**TEST AUTOMATION**  
**ENGINEER – FORMACIÓN**  
**INTEGRAL**

- Módulo 1: FUNDAMENTOS DEL TESTING DE SOFTWARE
- Módulo 2: CONTROL DE VERSIONES Y ENTORNOS DE DESARROLLO
- Módulo 3: FUNDAMENTOS DE PROGRAMACIÓN APLICADOS AL TESTING (JAVASCRIPT)
- Módulo 4: CARACTERISTICAS DE CYPRESS – AUTOMATIZACIÓN WEB I
- Módulo 5: CARACTERISTICAS DE CYPRESS – AUTOMATIZACIÓN WEB II
- Módulo 6: PRUEBAS DE APIS CON POSTMAN Y SUPERTEST

# CURSO:

## TEST AUTOMATION

## ENGINEER – FORMACIÓN

## INTEGRAL

- Módulo 7: HERRAMIENTAS DE PLAYWRIGHT Y PRUEBAS CON MÚLTIPLES NAVEGADORES
- Módulo 8: DESARROLLO GUIADO EN EL COMPORTAMIENTO (BDD) CON CUCUMBER.JS
- Módulo 9: HERRAMIENTAS DE AUTOMATIZACIÓN MÓVIL CON APPIUM
- Módulo 10: HERRAMIENTAS DE INTEGRACIÓN DE PRUEBAS EN CI/CD
- Módulo 11: HERRAMIENTAS DE DOCKER, ENTORNOS VIRTUALIZADOS Y PRUEBAS EN LA NUBE
- HERRAMIENTAS DE AUTOMATIZACIÓN DE UN FLUJO COMPLETO WEB + API + CI/CD



Te encuentras aquí

# Módulo 8: Desarrollo guiado en el comportamiento (bdd) con cucumber.Js.



# OBJETIVO ESPECÍFICO DEL MÓDULO

- EXPLICAR EL DESARROLLO GUIADO EN EL COMPORTAMIENTO (BDD) CON CUCUMBER.JS, DE ACUERDO A LAS APLICACIONES WEB, MÓVILES Y APIs.



**¿De qué manera cree que escribir pruebas en lenguaje natural puede influir en la colaboración entre desarrolladores, testers y otros actores no técnicos de un proyecto?**



# ¿QUÉ ES BDD?

- BDD (Behavior Driven Development) es una metodología ágil que prioriza la comprensión del comportamiento esperado de una funcionalidad antes de desarrollarla o testearla.
  - Se basa en la colaboración entre áreas técnicas y no técnicas.
  - Fomenta escribir escenarios de prueba desde el punto de vista del usuario.
  - Ayuda a prevenir malentendidos en los requisitos.
- 
- **Componentes clave:**
  - Reglas de negocio
  - Ejemplos reales escritos como “features”
  - Automatización de estos escenarios



# SINTAXIS GHERKIN PARA DESCRIBIR COMPORTAMIENTOS

- Gherkin es el lenguaje estructurado que se usa en BDD.
- Permite escribir pruebas como si fueran historias, usando palabras clave:

Feature: Login de usuario

Scenario: Acceso exitoso

Given el usuario está en la página de login

When ingresa credenciales válidas

Then accede al panel de control

- **Palabras clave principales:**
- **Feature:** nombre de la funcionalidad
- **Scenario:** caso específico a validar
- **Given:** contexto inicial
- **When:** acción del usuario
- **Then:** resultado esperado
- El formato es claro, entendible y facilita automatizar las pruebas sin ambigüedad.



# INTEGRACIÓN BDD CON FRAMEWORKS DE AUTOMATIZACIÓN

- Cypress + Cucumber:
- Instalar con:

```
npm install --save-dev cypress-cucumber-preproc
```

- Soporta archivos .feature + funciones de paso (step definitions)
- Ejecuta pruebas escritas en Gherkin como cualquier test Cypress

- **Playwright + Cucumber:**
- Usualmente requiere una configuración personalizada con @cucumber/cucumber y bindings manuales.
- Ideal para proyectos multibrowser que requieren expresividad y colaboración.
- BDD puede aplicarse sobre la lógica ya automatizada, aumentando legibilidad sin sacrificar potencia.



# PRUEBAS AUTOMATIZADAS COMPRENSIBLES PARA TODOS

- El gran valor de BDD es la escritura de pruebas en lenguaje natural, lo que facilita la colaboración.
- Ejemplo práctico:

```
Scenario: Usuario olvida su contraseña
  Given el usuario accede a "Olvidé mi contraseña"
  When ingresa su email registrado
  Then debería recibir un correo con instrucciones
```

- Traducción a código automatizado:

```
Given('el usuario accede a {string}', (opcion)  
    cy.contains(opcion).click();  
});
```

- Las pruebas pueden ser revisadas por cualquier miembro del equipo, sin necesidad de saber programación.

# VENTAJAS DEL ENFOQUE BDD EN EQUIPOS ÁGILES

- **Entendimiento compartido:**
- Todas las áreas comprenden el comportamiento esperado de la aplicación
  
- **Menos retrabajo:**
- Se reduce el riesgo de malinterpretar los requisitos
  
- **Facilita validaciones tempranas:**
- Los escenarios pueden discutirse antes del desarrollo



- **Documentación viva:**
- Las pruebas escritas en Gherkin sirven como documentación ejecutable
  
- **Alineamiento entre QA, devs y negocio:**
- Mejora la colaboración y la entrega de valor continuo





No olvide desarrollar los ejercicios que  
contiene el Módulo...

**¿Cómo puede el uso de BDD, junto con herramientas como Cypress o Playwright, mejorar la trazabilidad de requisitos, la validación funcional y la colaboración en equipos multidisciplinarios?**



**Éxito en la evaluación parcial y  
en la Prueba Final...**

*{desafío}*  
**latam\_**

*Academia de  
talentos digitales*

