

# API Testing

## ¿Qué es BDD?

**BDD (Behavior-Driven Development)**, o Desarrollo Guiado por el Comportamiento, es una metodología de desarrollo de software que surge como evolución del Desarrollo Guiado por Pruebas (TDD). Esta metodología se distingue por considerar el desarrollo de software como una combinación de aspectos técnicos y comerciales, proporcionando un enfoque más integral para crear productos que cumplan con las expectativas del negocio.

El principal objetivo de BDD es expresar el comportamiento esperado del software de manera clara y comprensible, facilitando la colaboración entre los equipos técnicos y no técnicos. A través de un lenguaje accesible, BDD ayuda a captar y entender no solo los aspectos técnicos, sino también los requisitos y expectativas del negocio, promoviendo una mejor alineación entre los objetivos del proyecto.

Características clave de BDD en el contexto de la automatización:

- **Lenguaje natural:** Utiliza un lenguaje descriptivo y sencillo que facilita la comunicación entre los equipos técnicos y los stakeholders del negocio.
- **Especificaciones ejecutables:** Las descripciones de comportamiento se transforman en especificaciones que pueden ser ejecutadas como pruebas automatizadas, asegurando que el software se comporta según lo esperado.
- **Herramientas populares:** Herramientas como **Cucumber** permiten escribir pruebas en un formato legible para humanos y luego ejecutarlas como pruebas automatizadas contra el código de la aplicación.
- **Colaboración:** Fomenta la colaboración entre los diversos roles dentro del equipo de desarrollo (desarrolladores, testers, analistas de negocio) al centrarse en el comportamiento esperado del sistema.

- **Retroalimentación rápida:** Gracias a la automatización de pruebas, BDD permite obtener retroalimentación rápida sobre si el software cumple con las expectativas del negocio.

## Implementación en metodologías ágiles

En metodologías ágiles como Scrum o Kanban, los requisitos del software generalmente se expresan mediante **Historias de Usuario**, que son descripciones breves y orientadas al usuario de funcionalidades específicas. Cada Historia de Usuario incluye **criterios de aceptación**, que definen las condiciones claras y específicas que deben cumplirse para que la historia se considere completa.

Una vez definidos los criterios de aceptación, estos sirven para crear pruebas de aceptación automatizadas, que verifican que el software cumple con los requisitos. Para escribir estas pruebas, a menudo se utiliza **Gherkin**, un lenguaje comprensible para los usuarios no técnicos que permite describir los comportamientos esperados del software mediante una sintaxis simple, basada en las palabras clave "**Dado**", "**Cuando**", "**Entonces**".

### **Estructura de un escenario en Gherkin:**

- **Given (Dado):** Define el estado inicial antes de que ocurra la acción, las condiciones previas necesarias.
- **When (Cuando):** Especifica la acción o evento que ocurre en el escenario.
- **Then (Entonces):** Describe los resultados o cambios esperados después de que se ejecute la acción.

### **Ejemplo de escenario en Gherkin:**

**Escenario:** Escenario: Iniciar sesión con credenciales válidas

- **Dado** que estoy en la página de inicio de sesión
- **Cuando** ingreso mi nombre de usuario y contraseña válidos
- **Entonces** debería ser redirigido a la página de inicio

Este tipo de escenarios puede ser ejecutado directamente como pruebas automatizadas mediante herramientas como **Cucumber** o **SpecFlow**, facilitando así la validación del comportamiento del sistema.

## BDD en acción:

Vamos a comparar cómo una historia de usuario tradicional puede transformarse en un escenario BDD con Gherkin:

**Historia de Usuario tradicional:** Como usuario registrado en un sitio web de comercio electrónico, quiero poder agregar productos a mi carrito de compras para poder comprarlos más tarde.

### **Criterios de aceptación:**

- Como usuario, puedo ver los productos disponibles.
- Como usuario, puedo agregar productos al carrito de compras.
- Como usuario, puedo ver los productos en mi carrito.
- Como usuario, puedo eliminar productos de mi carrito.

### **Escenarios en Gherkin:**

```
Funcionalidad: Agregar productos al carrito de compras

Escenario: Agregar un producto al carrito de compras
Dado que estoy en la página de productos disponibles
Cuando selecciono el producto "Camiseta azul" y hago clic en "Agregar al carrito"
Entonces debería ver un mensaje confirmando que el producto fue agregado

Escenario: Verificar el contenido del carrito de compras
Dado que tengo productos en mi carrito
Cuando veo el contenido de mi carrito
Entonces debería ver una lista con los productos agregados

Escenario: Eliminar un producto del carrito de compras
Dado que tengo productos en mi carrito
Cuando elimino el producto "Camiseta azul" de mi carrito
Entonces debería ver un mensaje confirmando la eliminación

Escenario: Vaciar el carrito de compras
Dado que tengo productos en mi carrito
Cuando vacío mi carrito
Entonces debería ver un mensaje confirmando que el carrito está vacío
```

### **Frameworks BDD populares:**

- **JBehave:** Framework que permite el desarrollo y ejecución de historias de usuario escritas en el formato dado-cuando-entonces, integrable con JUnit y TestNG.
- **Cucumber:** Una de las herramientas más utilizadas en BDD, que permite escribir pruebas en Gherkin y ejecutarlas en lenguajes como Java, Groovy, Scala, entre otros.