

SpringFramework

Java

Pruebas unitarias

Pruebas unitarias

SpringFramework

Java

Pruebas unitarias

Pruebas unitarias

SpringFramework

SpringFramework

Java

Pruebas unitarias

Java

Una prueba unitaria es la verificación y comprobación del correcto funcionamiento de un bloque o pequeño componente de software.

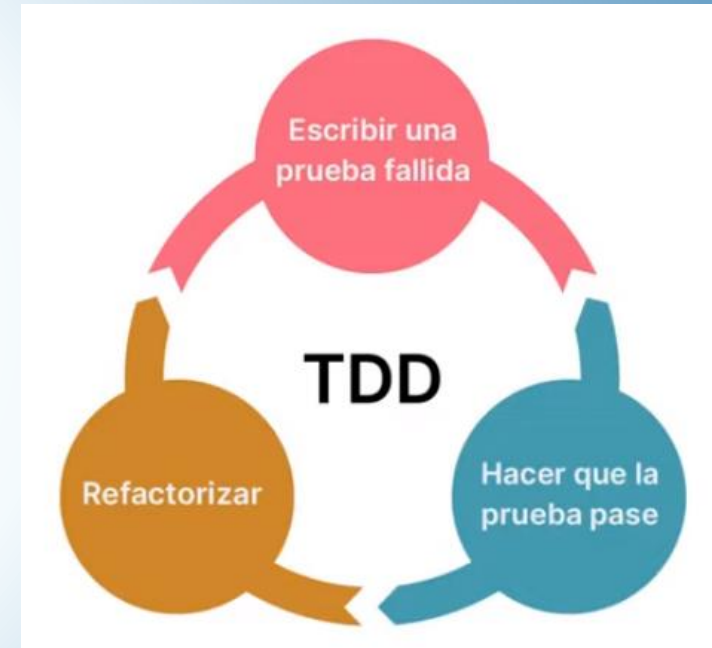


Debe ser de una ejecución rápida, reutilizable, independiente.

Finalmente se programa la prueba y se verifica que este se cumpla (Plan de pruebas). Las ejecuta el desarrollador.

Test-Driven Development es una técnica que se basa en construir las pruebas y luego ejecutarlas; basado en sus resultados se construye el código, se genera un ciclo hasta que los resultados de la prueba son exitosos.

Se deben crear antes del desarrollo



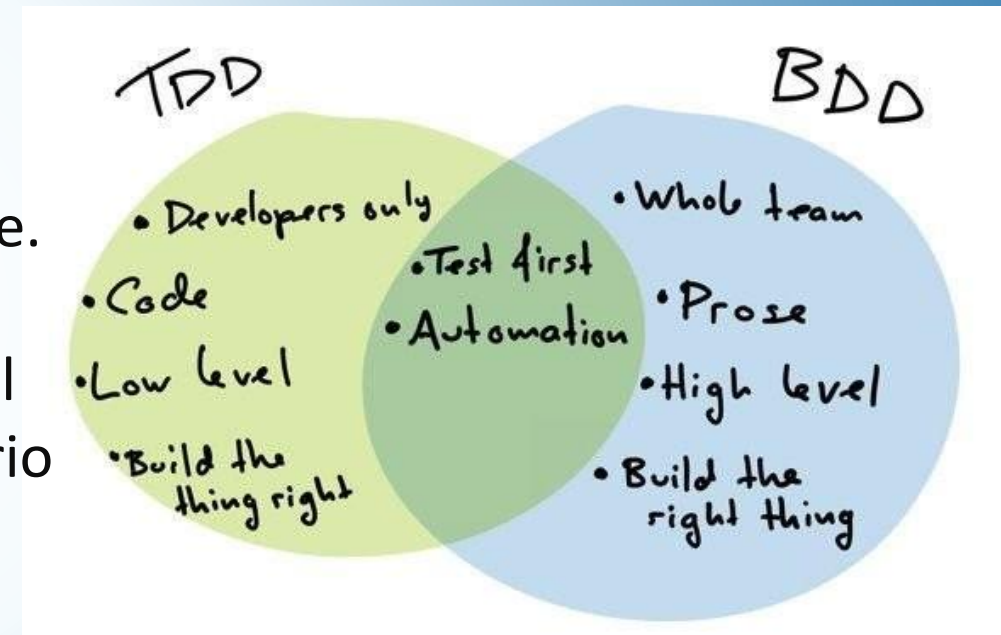
Se basa en TFT (Test First Development) y R (Refactoring)

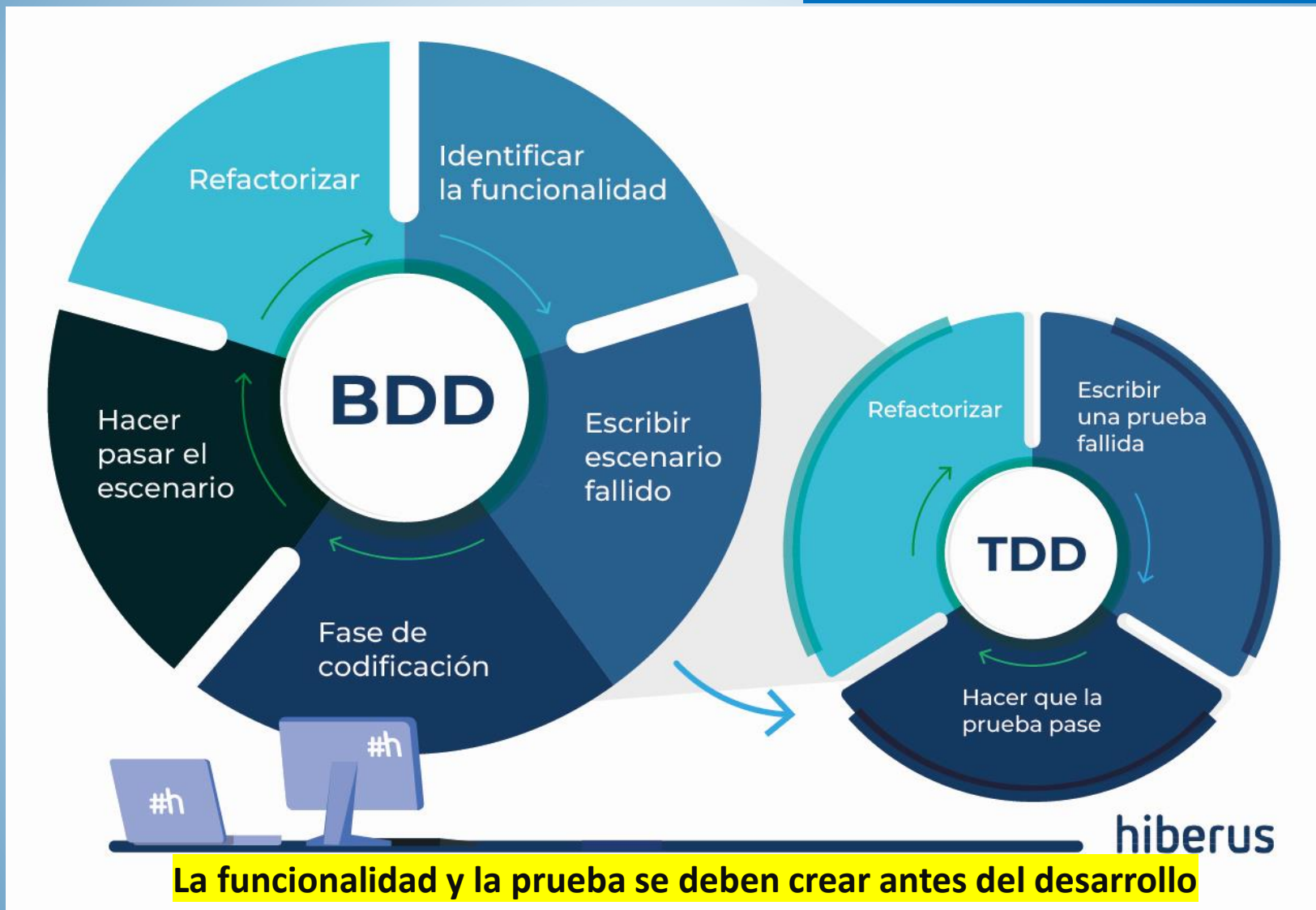
Behavior Driven Development (BDD), o desarrollo orientado al comportamiento, es una metodología para realizar testing.

El BDD permite a todos los implicados entender el proceso desarrollado y el contenido del código fuente.

Hay relación directa entre BDD y TDD, la diferencia del TDD, el BDD define las pruebas centradas en el usuario y el comportamiento del sistema y no en las funcionalidades de este.

En otras palabras, el BDD describe las pruebas en un lenguaje natural que entienden todos los equipos de un proyecto, y no únicamente los programadores.







JUnit



Ambiente de desarrollo y Librerías para pruebas



Esta es un conjunto de librerías creadas por Erich Gamma y Kent Beck, estas se usan para construir pruebas unitarias en aplicaciones con lenguaje java.

Version más reciente a la fecha es:



Esta es una librería Java usada para simular el comportamiento de una clase dinámicamente, usado para verificar servicios y/o controladores usados en la estructura de desarrollo de SpringFramework y Spring-boot

Más información:
<https://site.mockito.org/>



Para implementar en java la librería se requiere la dependencia Maven:

```
<dependency>  
  <groupId>org.mockito</groupId>  
  <artifactId>mockito-core</artifactId>  
  <version>5.11.0</version>  
  <scope>test</scope>  
</dependency>
```