

**11<sup>a</sup>**  
Emisión

# DIPLOMADO Desarrollo de Sistemas con Tecnología Java

## Módulo 5

### Introducción de Aplicaciones Empresariales con Spring Framework.

*Mtro. ISC Miguel Ángel Sánchez Hernández*



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

Dirección General de Cómputo y de Tecnologías de información y Comunicación

Dirección de Docencia en TIC



Educación  
Continua  
1971 - 2021

# Objetivo

Crear pruebas unitarias con JUnit y Mockito, para un desarrollo confiable

The JUnit logo, featuring the word "JUnit" in a serif font. The "J" is green, and "Unit" is red.The Mockito logo, featuring the word "mockito" in a lowercase, sans-serif font. The "mock" part is green, and "ito" is black. To the right of the text is an illustration of a glass with a straw and a sprig of mint.The Spring logo, featuring a green leaf icon to the left of the word "spring" in a lowercase, sans-serif font.

# Lo que veremos

- Configurar Spring con JUnit 5 para pruebas unitarias
- Configurar Spring con Mockito para pruebas unitarias



# Configuración JUnit con Spring

Las pruebas unitarias nos permiten probar nuestro código, JUnit es un framework que nos permitirá hacer varias pruebas de funcionalidad de nuestro código, siguiendo un esquema relativamente sencillo.



# Beneficios de JUnit

JUnit nos permitirá hacer lo siguiente:

- Pruebas unitarias: verifica piezas del código que cumplen ciertas reglas de negocio, con esto podemos certificar un resultado que esperamos.
- Pruebas continuas: Si hay un cambio se puede volver hacer una prueba general para ver si no hay afectaciones en otras clases.
- Escribir pruebas: Nos permite crear nuestras propias pruebas de acuerdo a nuestras necesidades.



# Configurar JUnit con Spring

Los objetivos de la practica son:

- Configuración del proyecto para ocupar Junit 5 con Spring
- assertEquals.
- @BeforeEach,@AfterEach, @BeforeAll,@AfterAll
- @assertArrayEquals
- @assertNotNull,@assertNull
- @assertSame,@assertNotSame
- @assertTrue,@assertFalse
- @assertAll,@assertThrows,@assertIterableEquals
- @assertTimeout



## Práctica 13: JUnit 5



# Mockito

Con el Framework de Mockito nos sirve para hacer pruebas unitarias de nuestro código, sin preocuparnos por código de tercera partes. Esto se logra creando objetos simulados(mock) para tener un entorno determinador y controlado.

Ahora nuestro desarrollo de pruebas será un desarrollo impulsado por el comportamiento BDD(Behavior Driven Development).



# Mockito

Nosotros ahora deberemos pensar de la siguiente manera para hacer las pruebas unitarias:

- Dado que: Preparar el contexto, dependencias y parámetros de entrada.
- Cuando: Invocamos el método a validar con los parámetros de entrada.
- Entonces: Validamos si pasa la prueba o no.



## Práctica 14: Mockito

