



NTT Data

# Curso Vacacional Automatización de Pruebas

21 Diciembre 2022 – Clase 2

---

**FUTURE  
AT HEART**

# Contenido

¿Que esperan de este curso?



## Modulo I – Introducción la calidad de software

- Calidad de software
- Tipos de pruebas
- Automatización de pruebas

## Modulo II – Introducción a Java

- Que es Java
- Conceptos de programación
- Instalación Ambiente (IDE, MVN)
- Creación de primer proyecto
- Descarga de dependencias (Selenium , Junit)
- Ejecutar primer script

## Modulo III – Automatización en Java con Selenium

- Componentes del FW
- BDD (Gherkin, Steps, Métodos)
- Identificadores o Selectores
- Inspeccionar objetos
- Tipos de Objetos
- Xpath (Tipos)
- Introducción a testing Continuo

En capítulos anteriores...



# PREREQUISITOS

## Conocimientos requeridos:

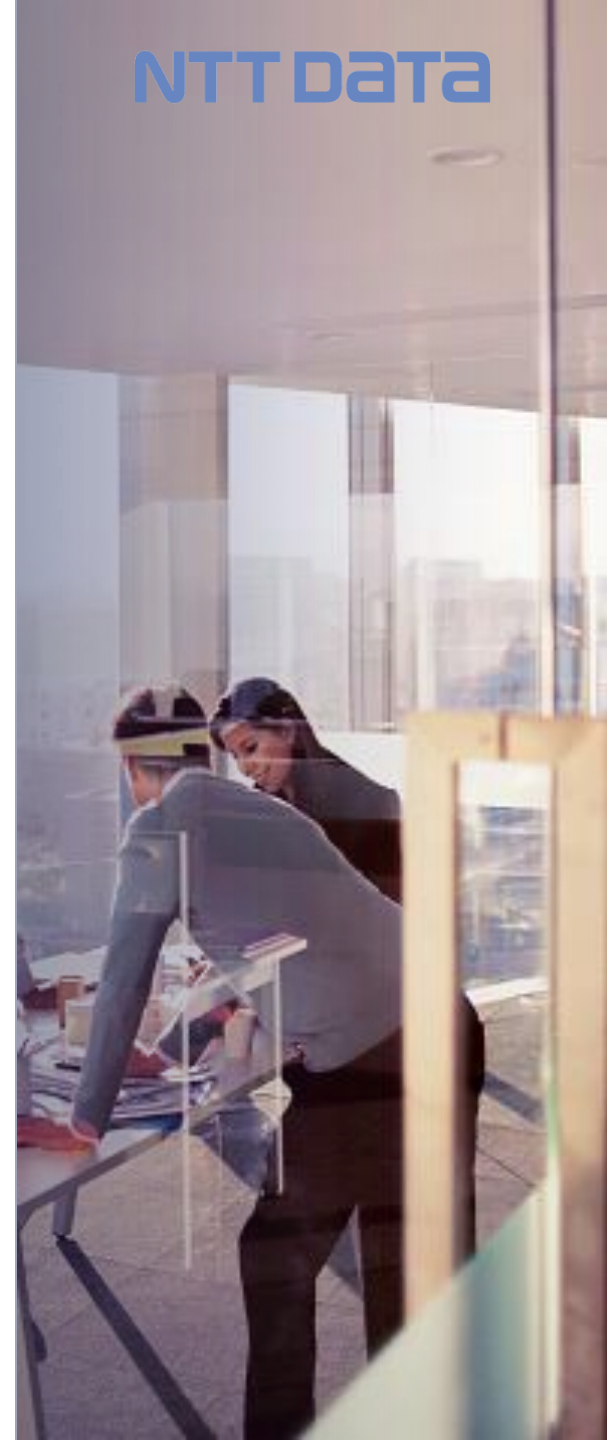


Es importante que quienes participen de este curso de formación conozcan de:

- Ninguna (Mas o Menos)
- Programación orientada a objetos
- Experiencia de desarrollo básico en Java

## Requisitos técnicos:

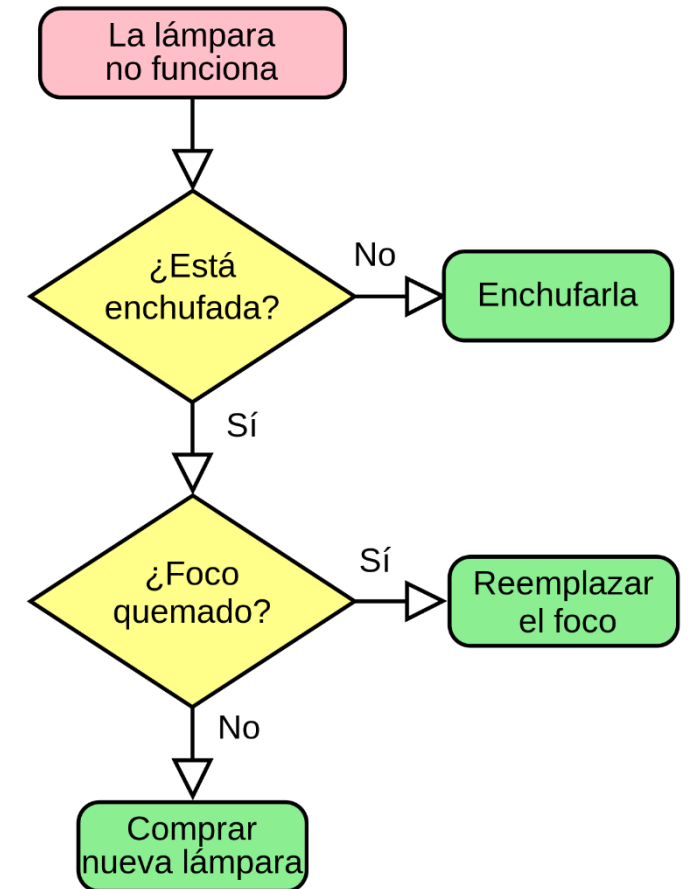
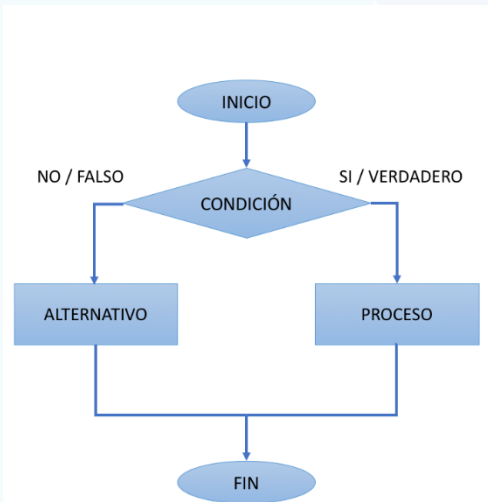
- JDK Java 11
- Sistema operativo  
Windows
- IntelliJ Idea Community  
Edition
- Git
- Maven





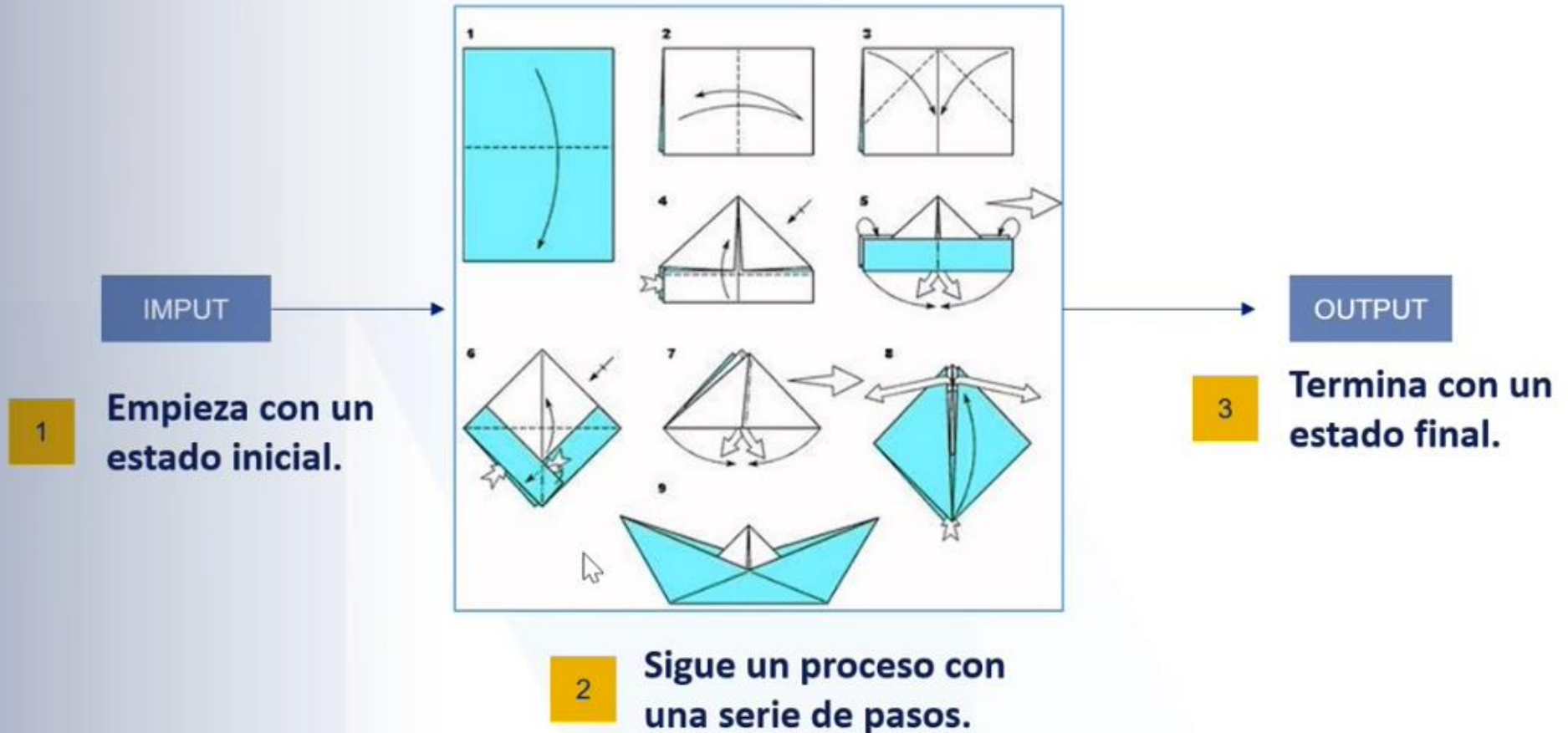
# Algoritmo

En matemáticas, lógica, ciencias de la computación y disciplinas relacionadas, un algoritmo es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades



# Algoritmo

ES UNA SERIE DE SECUENCIA DE INSTRUCCIONES



# Algoritmo



¿Cómo recargar una tarjeta del metropolitano?



¿Cómo preparar mi plato favorito?



¿Cómo pedir un taxi por aplicación?

¿QUÉ TIENEN EN COMÚN LOS 3 CASOS?



# Que es la programación

**ES LA CIENCIA QUE SE ENCARGA DE INDICAR AL ORDENADOR QUE HACER, MEDIANTE UNA SERIE DE INSTRUCCIONES CONOCIDAS COMO ALGORITMOS**

```
1  Proceso sin titulo
2      totalfactura<-0;
3      Escribir "Ingrese la fecha actual";
4      Leer fecha;
5      Escribir "Ingrese el nombre del cliente";
6      Leer cliente;
7      Escribir "Ingrese la direccion del cliente";
8      Leer direccion;
9      Escribir "Ingrese la cantidad de detalles";
10     Leer detalles;
11     Para i<-1 Hasta detalles Con Paso 1 Hacer
12         Escribir "Ingrese la cantidad";
13         Leer cantidad;
14         Escribir "Ingrese la descripción";
15         Leer descripcion;
16         Escribir "Ingrese el precio";
17         Leer precio;
18         total<-cantidad*precio;
19         totalfactura<-totalfactura+total;
20     FinPara
21     Escribir totalfactura;
22 FinProceso
```

Ejemplo de pseudocódigo

```
public class repetitivo3 {
    public static void main(String[] args) {
        // TODO code application logic here
        int num, cont, pares, impares;
        pares = 0;
        impares = 0;
        Scanner ingreso=new Scanner(System.in);
        for(cont = 1; cont < 11; cont++){
            System.out.print("Ingrese número " + cont + " : ");
            num = Integer.parseInt(ingreso.next());
            if((num - ((num / 2)*2)) == 0){
                pares++;
            }else{
                impares++;
            }
        }
        System.out.println("PARES INGRESADOS : " + pares);
        System.out.println("IMPARES INGRESADOS : " + impares);
    }
}
```

Ejemplo de Código Fuente



# Que son los paradigmas de la programación

## ¿QUÉ SON LOS PARADIGMAS DE PROGRAMACIÓN?

Los paradigmas son los diferentes estilo de usar la programación para resolver un problema.



### PROGRAMACIÓN ESTRUCTURADA

Programación secuencial con la que todos aprendemos a programar. Usa ciclos y condicionales.



### PROGRAMACIÓN REACTIVA

Observa flujos de datos asíncronos y reacciona frente a sus cambios.



### PROGRAMACIÓN ORIENTADA A OBJETOS

Divide los componentes del programa en objetos que tienen datos y comportamiento y se comunican entre sí.

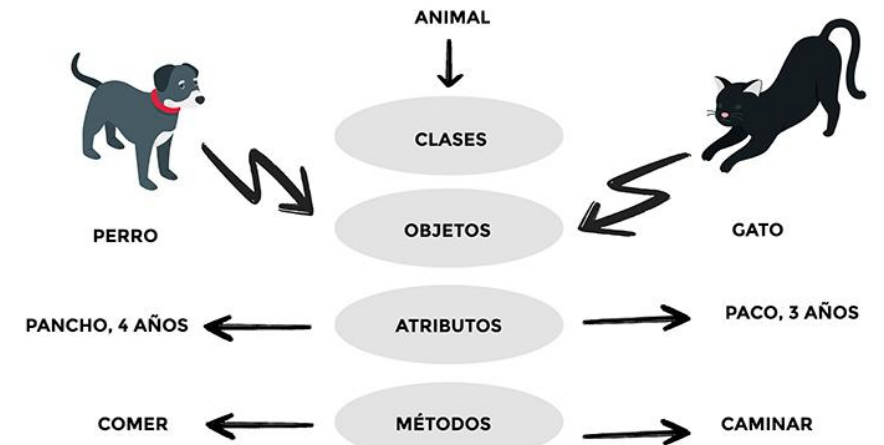


### PROGRAMACIÓN FUNCIONAL

Divide el programa en tareas pequeñas que son ejecutadas por funciones.



Ejemplo de POO:



# Que es Java

**JAVA ES UN TIPO DE LENGUAJE DE PROGRAMACIÓN Y UNA PLATAFORMA INFORMÁTICA, CREADA Y COMERCIALIZADA POR SUN MICROSYSTEMS EN EL AÑO 1995.**

## CARACTERÍSTICAS

### SIMPLE

Java ofrece la funcionalidad de un lenguaje potente, derivado de C y C++, pero sin las características menos usadas y más confusas de estos, haciéndolo más sencillo.

### ORIENTADO A OBJETOS

El enfoque orientado a objetos (OO) es uno de los estilos de programación más populares. Permite diseñar el software de forma que los distintos tipos de datos que se usen estén unidos a sus operaciones.

### MULTIHILO

Java logra llevar a cabo varias tareas simultáneamente dentro del mismo programa. Esto permite mejorar el rendimiento y la velocidad de ejecución.

### DISTRIBUIDO

Java proporciona una gran biblioteca estándar y herramientas para que los programas puedan ser distribuidos.

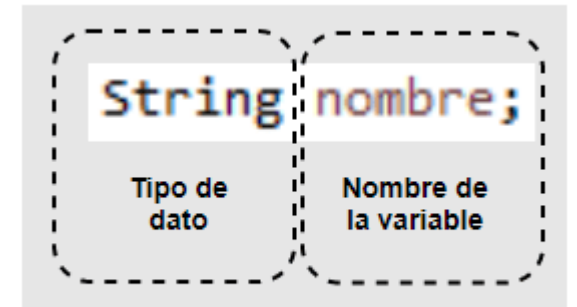
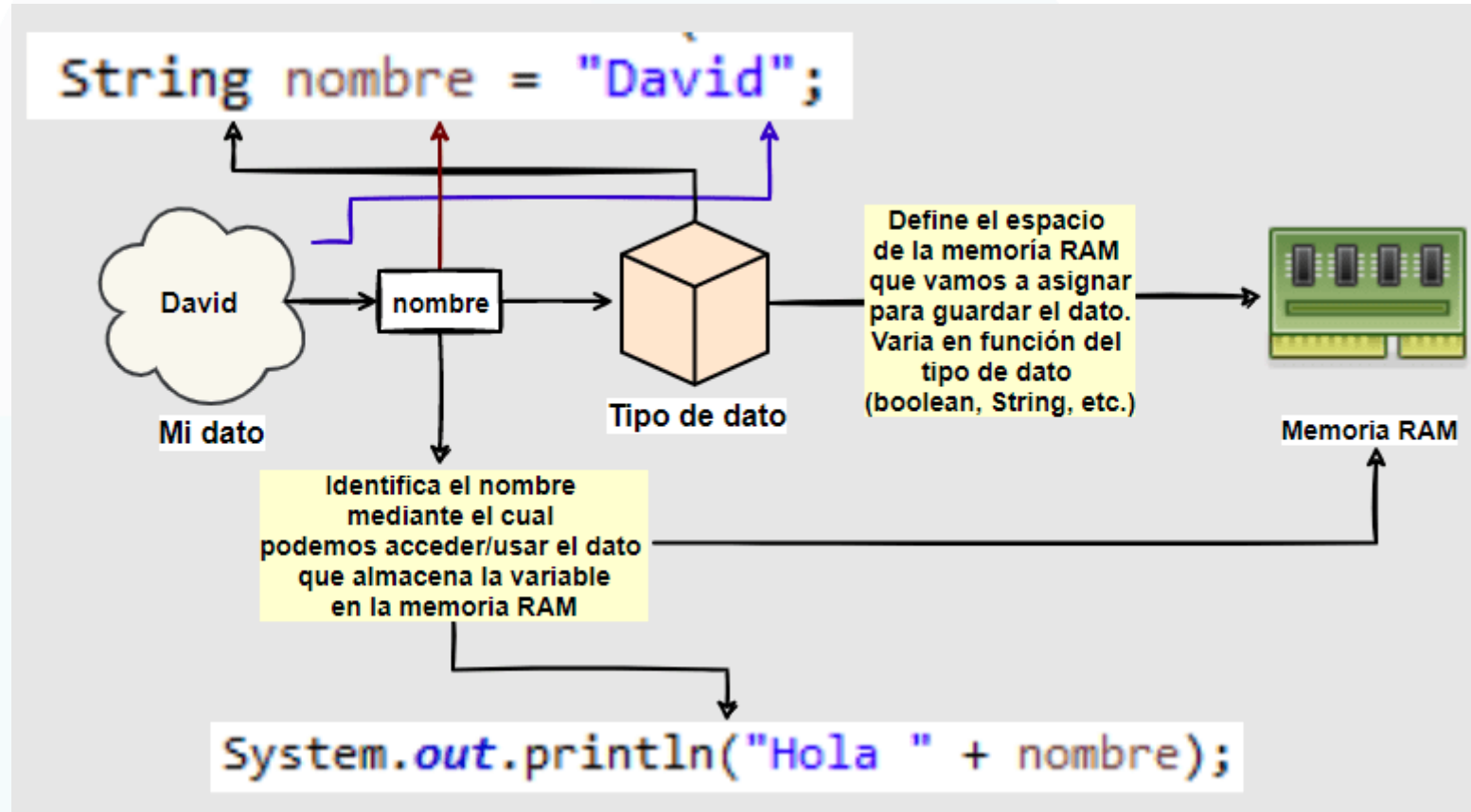
### INDEPENDIENTE DE PLATAFORMA

Esto significa que programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware, lo que lo hace portable.

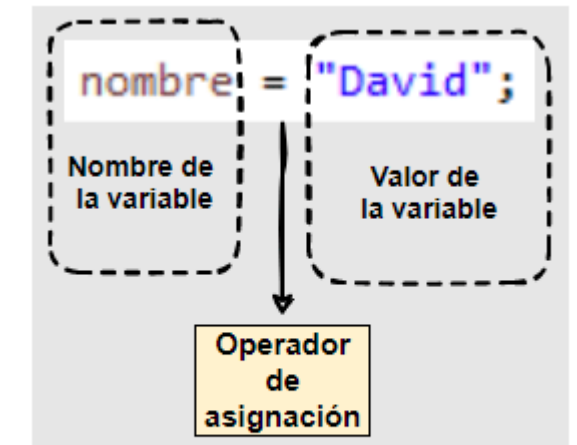




# Que es Java - Variables



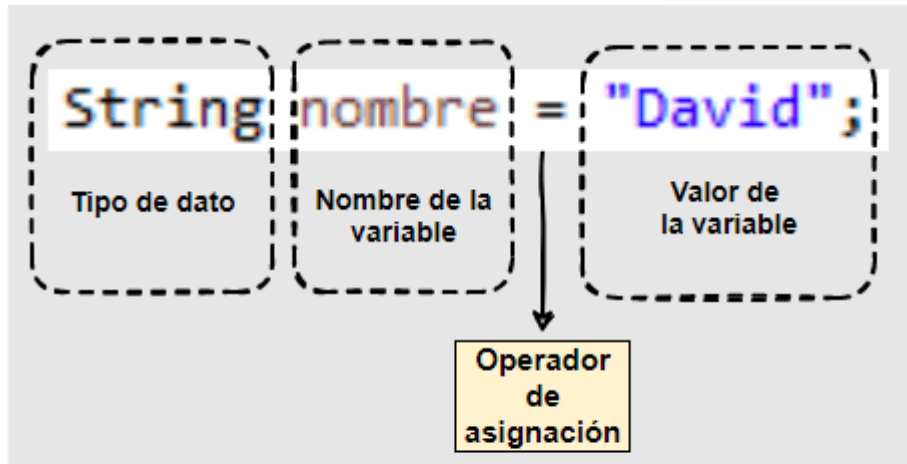
Declaración de una variable en Java.



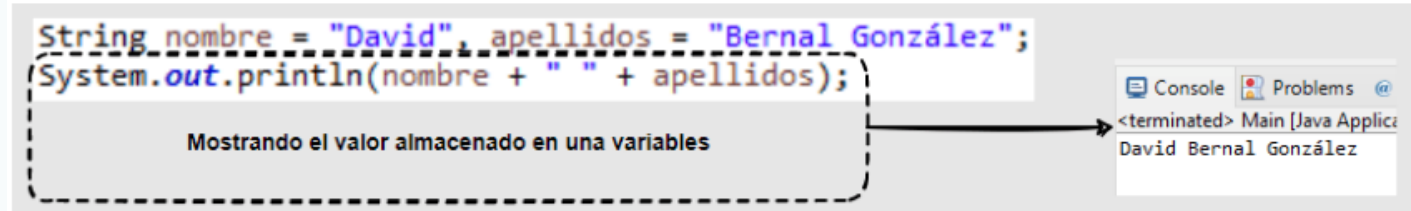
Asignación de un valor a una variable en Java.



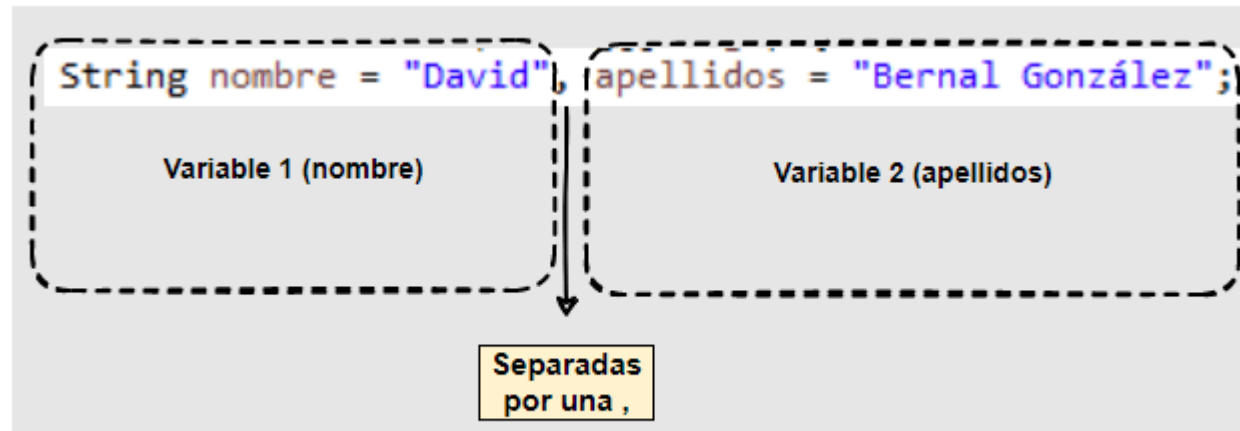
# Que es Java - Variables



Declaración de una variable en Java y asignación de un valor en una sola línea.

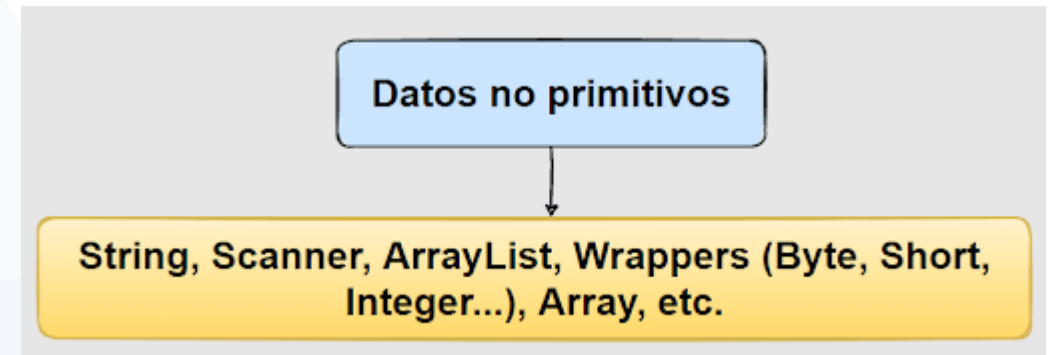
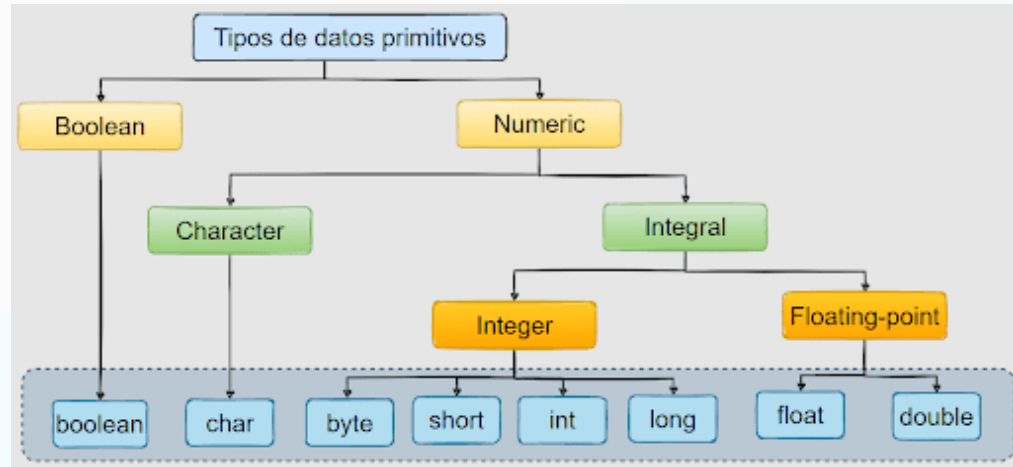


Ejemplo de uso de una variable en Java.




Declaración de varias variables en Java y asignación de sus valores.

# Que es Java - Variables




# Selenium IDE


**Welcome to Selenium IDE!**  
Version 3.17.2

What would you like to do?

- [Record a new test in a new project](#)
- [Open an existing project](#)
- [Create a new project](#)
- [Close Selenium IDE](#)

To learn more on Selenium IDE and how to use it visit the [the Selenium IDE project page](#).

 chrome web store

 oacarrillo@gmail.com ▾

✕

selenium ide

selenium ide **plugin**

selenium ide **dev**

selenium **identity**

selenium **ideas**


selenium idea


**a studio**

seleniu... **reservation**

Extensiones ⓘ

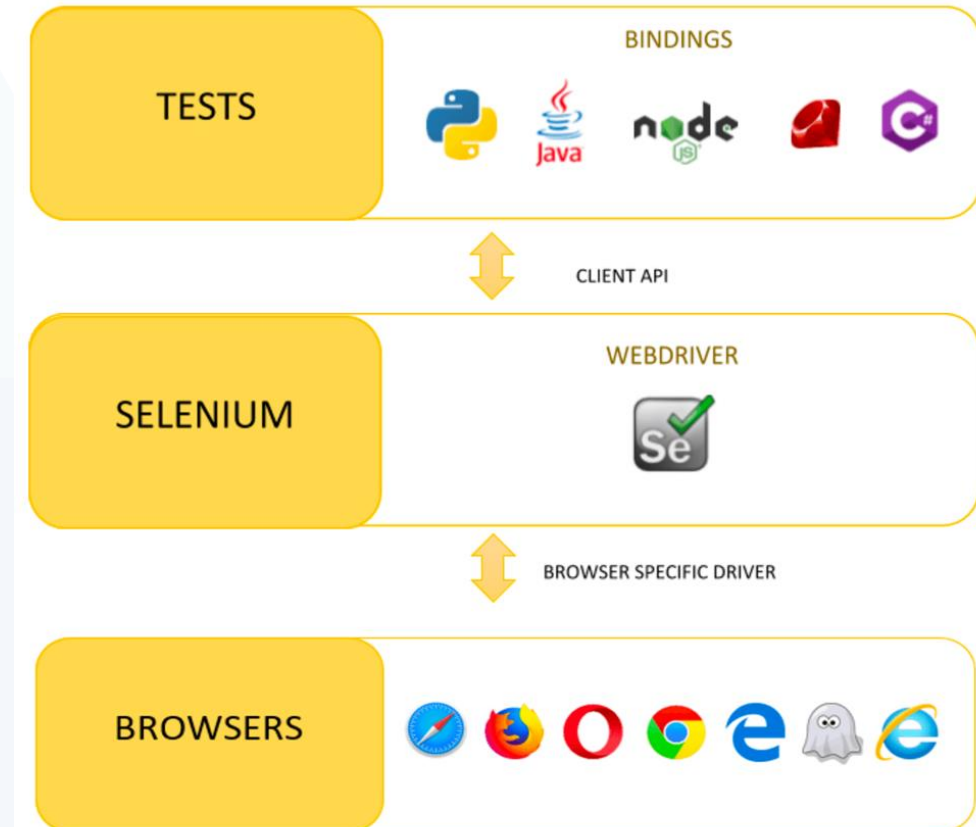
Más extensiones



**Selenium IDE**  
[seleniumhq.org](#)  **Destacados**  
Selenium Record and Playback tool for ease of getting acquainted with Selenium WebDriver  
★★★★★ 236 Herramientas para desarrolladores

# Selenium

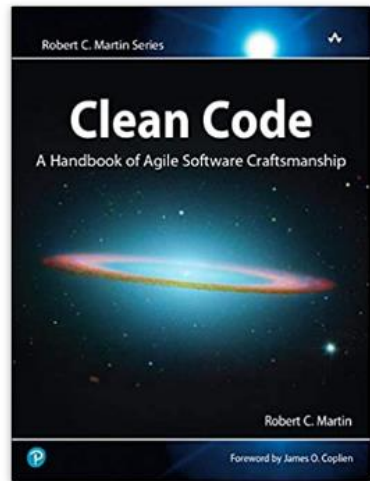
Selenium es un entorno de pruebas de software para aplicaciones basadas en la web. Selenium provee una herramienta de grabar/reproducir para crear pruebas sin usar un lenguaje de scripting para pruebas (Selenium IDE). Incluye también un lenguaje específico de dominio para pruebas (Selenese) para escribir pruebas en un amplio número de lenguajes de programación populares incluyendo Java, C#, Ruby, Groovy, Perl, PHP y Python. Las pruebas pueden ejecutarse entonces usando la mayoría de los navegadores web modernos en diferentes sistemas operativos como Windows, Linux y OSX.







Libros > Computadoras y Tecnología > Programación



Escuchar



Ver esta imagen

Sigue a los autores



Robert C. Martin

Seguir

## Clean Code: A Handbook of Agile Software Craftsmanship 1st Edición

de Robert C. Martin (Author)

★★★★☆ 4,703 calificaciones 4.4 en Goodreads 19,245 calificaciones

Parte de: Robert C. Martin (14 libros)

Ver todos los formatos y ediciones

Kindle  
US\$28.99

Leer con nuestra **Aplicación gratuita**

Pasta blanda  
**US\$42.10 - US\$44.99**

7 Usado de US\$39.31

2 Nuevo de US\$44.99

Even bad code can function. But if code isn't clean, it can bring a development organization to its knees. Every year, countless hours and significant resources are lost because of poorly written code. But it doesn't have to be that way.

Noted software expert Robert C. Martin, presents a revolutionary paradigm with *Clean Code: A Handbook of Agile Software Craftsmanship*. Martin, who has helped bring agile principles

▼ Leer más

Reportar información de producto incorrecta

ISBN-10



9780132350884

ISBN-13



978-0132350884

Edición



1er

Editorial



Pearson



2008 - 2022

14 Años

# Estructura de las pruebas unitarias

Por norma general, los unit test deberían seguir una estructura AAA, Este patrón fue creado por BILL WAKE (2001) y esta compuesto por:

- **Arrange** (Organizar): En esta parte de la prueba, debes establecer las condiciones iniciales para poder realizarla, así como el resultado que esperas obtener. Y esto significa por ejemplo, declarar las variables y crear las instancias de los objetos.
- **Act** (Accionar): Es la parte de ejecución, tanto del fragmento de código de la prueba, como del fragmento de código a testear.
- **Assert** (Comprobar o certificar): Por último, se realiza la comprobación para verificar que el resultado obtenido, coincide con el esperado.

NTT DATA

AAA  
INVEST



```
[TestMethod]
public void TestMethod()
{
    //Arrange test
    testClass objtest = new testClass();
    Boolean result;

    //Act test
    result = objtest.testFunction();

    //Assert test
    Assert.AreEqual(true, result);
}
```

# Estructura de las pruebas unitarias

```
public class Suma {  
    public int sumar(int numeroUno, int numeroDos){  
        return numeroUno + numeroDos;  
    }  
}
```

@Test

```
public void sumarDosNumerosPositivos(){
```

//Arrange

```
Suma suma = new Suma();
```

```
int numeroUno = 2;
```

```
int numeroDos = 3;
```

```
int resultadoObtenido;
```

//Act

```
resultadoObtenido = suma.sumar(numeroUno,numeroDos);
```

//Assert

```
Assert.assertEquals( expected: 5, resultadoObtenido);
```

1

2

3

Beneficios:

- Estándar (Mantenimiento)
- Se elimina la duplicidad
- Separación clara

1[

## Ingredientes

- ☐ 250 gramos de harina
- ☐ 150 gramos de mantequilla
- ☐ 150 gramos de azúcar
- ☐ 5 huevos enteros
- ☐ 1 cucharada copeteada de polvo para hornear
- ☐ 1 cucharada de vainilla

## Preparación

### ✓ Paso 1

Precalienta el horno a 175° centígrados y engrasa y enharina un molde para hornear.

### ✓ Paso 2

Bate la mantequilla con el azúcar hasta que acrete y, sin dejar de batir, agrega

### ✓ Paso 3

Cierne la harina con el polvo para hornear en incorpóralo a la mezcla anterior y añade la vainilla.

## Resultado esperado



2[

3[

Arrange

Act

Assert



**F**ast (rápido)

**I**ndependent (independiente)

**R**epeatable (repetible)

**S**elf-validating (auto evaluable)

**T**imely (oportuno)

# Principios FIRST - F

## **F**ast (rápido)

Los test unitarios deben completar su ejecución lo más rápido posible. Al ser pruebas que se realizan (o deberían realizarse) sobre fragmentos pequeños de código, deben finalizar su ejecución lo antes posible.



## Independent (independiente)

En las pruebas unitarias, el objeto de prueba no debe depender de otra unidad de código. Esto no significa que no requiera de datos para poder ejecutar ese código, pero esto se realizará a través de **mocks o stubs**, que vienen a ser «objetos falsos» creados específicamente para realizar las pruebas. Pero, en ningún caso, utilizaremos otras partes del código que no forme parte del unit test.

**Dummy** : Objetos ficticios que se pasan al método pero no se usan. Por lo general, solo se usan para rellenar listas de parámetros.

**Fake** : Objetos falsos que toman un atajo para cumplir suprir un valor de entrada o componente. Ej: Base de datos en memoria.

**Stubs** : Respuestas enlatadas para un solo contexto de prueba. Ej: Un parámetro constante (Cedula, Tipo)

**Spies** : Proporcionan información específica reaccionando de acuerdo a como fueron llamados. Ej: Ingreso una cedula y me retorna # de líneas

**Mocks** : Objetos pre programados complejos (Subrutinas).

Cada prueba unitaria debe ser independiente de otra

- Se pueden ejecutar las pruebas en diferente orden
- El resultado no se ve alterado por el orden de ejecución de la prueba
- Su nombre lo dice prueba unitaria (No de integración)

## Repeatable (repetible)

Las pruebas unitarias deben poder repetirse. De hecho, lo ideal es repetirla muchas veces. Por ejemplo, se deberían ejecutar las pruebas cada vez que suba un nuevo código al repositorio.

- En mi computadora si funciona
- Si pero no le vamos a dar tu computadora al cliente



Repetible independiente del

- Servidor
- Computador
- Entorno
- No depende de configuración de usuario
- No depende de configuración de las herramientas
- Me deben dar el mismo resultado en cualquier entorno



### **Self-validating (auto evaluable)**

Los unit test deben mostrar de forma clara, y sin que sea necesario tu intervención, el resultado de la prueba. Gracias a la parte de «arrange», donde estableces los resultados a obtener, la parte de «assert» puede responder si la prueba ha sido satisfactoria o no.



## **Timely (oportuno)**

Los test unitarios deben realizarse lo antes posible, el código no debe llegar a producción sin haber escrito y superado las pruebas unitarias. Incluso, si es posible, deberían desarrollarse antes que el código (Test Driven Development)



FAST SERVICE

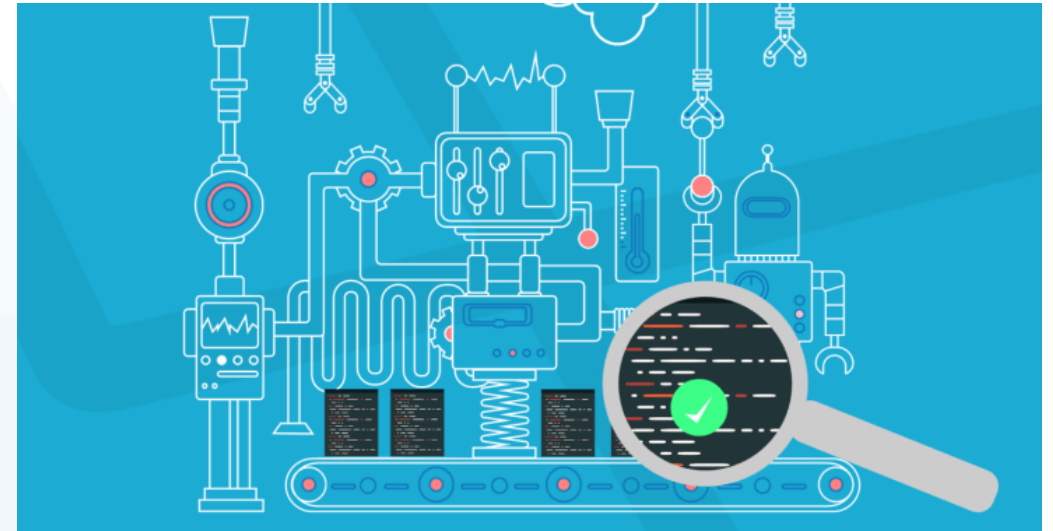
# Cuando hacer o no pruebas unitarias

Se hacen para:

- Menos errores en producción
- Resolución mas rápida de errores
- Comprensión del código
- Sencillez de integración
- Mejoras en la estructura del código
- Localización de errores
- Reducción de costes

Cuando no es posible:

- El código cambia mucho
- El periodo de vida del producto es muy corto
- Componentes de menor importancia dentro del código con tiempos de entrega ajustados





**JUnit** es un conjunto de bibliotecas creadas por Erich Gamma y Kent Beck que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

JUnit es un conjunto de clases (*framework*) que permite realizar la ejecución de clases Java de manera controlada, para poder evaluar si el funcionamiento de cada uno de los métodos de la clase se comporta como se espera. Es decir, en función de algún valor de entrada se evalúa el valor de retorno esperado; si la clase cumple con la especificación, entonces JUnit devolverá que el método de la clase pasó exitosamente la prueba; en caso de que el valor esperado sea diferente al que regresó el método durante la ejecución, JUnit devolverá un fallo en el método correspondiente.

<https://junit.org/junit5/docs/current/user-guide/>



**NTT DATA**

**GRACIAS**

**FUTURE  
AT HEART**