

# Bienvenidos!!



Selenium WebDriver



## Emiliano Gnocchi

Ingeniero en Sistemas

Desarrollador de Software | Fundador de Selenium Academy

[seleniumcurso@gmail.com](mailto:seleniumcurso@gmail.com)

+59894822457



## Sobre el curso

16 clases

2 horas semanales

Teóricos + Prácticos + Tips

Buenas prácticas de programación

Certificado de Asistencia ó de Aprobación



## Módulos del curso

El curso se compone de 3 grandes modulos:

- 1) Conceptos de programación estructural
- 2) Conceptos de programación orientada a objetos
- 3) Automatización con Selenium



Selenium WebDriver



# Conceptos de Programación Estructural

Se realizarán pequeños programas utilizando las estructuras de control y repetición de Java, así también como los operadores básicos del lenguaje como sumas, restas, multiplicaciones, etc.

Duración: 4 clases



selenium  academy

## Cronograma del Módulo 1

### Clase 1:

- Pseudocódigo
- Variables en Java
- Estructuras de los programas

### Clase 2:

- Estructuras de repetición for y while
- Sentencias en Java



selenium  academy

# Cronograma del Módulo 1

## Clase 3:

- Variables booleanas
- Métodos
- Variables finales

## Clase 4:

- Git
- Sourcetree
- Manejo de ramas, gitflow



# Conceptos de Programación Orientada a Objetos

Los objetivos de este módulo son conocer los conceptos relacionados a la programación orientada a objetos, para ello, se trabajarán con clases, objetos, métodos, herencias, listas, arreglos y excepciones.

Duración: 4 clases



# Cronograma del Módulo 2

## Tema 5:

- Programación orientada a objetos: Clases y objetos

## Tema 6:

- Excepciones, try/catch, Listas y Arreglos

## Tema 7:

- Herencia y polimorfismo
- Relaciones entre clases
- Privacidad (Public, private, protected)



# Módulo 3: Selenium

Los objetivos de este módulo son conocer los conceptos a Selenium Webdriver.

Conocer las herramientas para automatizar y automatizar sitios web activos validando los resultados obtenidos.

Duración: 8 clases



Selenium WebDriver



# Semana 1

## Introducción a Selenium

- Presentación e Introducción a la automatización
- Armado del proyecto de automation.
- Maven: armado del POM.xml
- Instalando webdrivers
- Identificando elementos. Html - CSS
- Primeras pruebas usando Selenium



Selenium WebDriver



# Semana 2

## Usando las herramientas de Selenium

- Esperas. Problemas de sincronismo
- Implicit waits y Explicit waits
- Links
- Interactuando con el webdriver
- Interactuando con un Select (combos)
- Nociones de Asserts.



Selenium WebDriver



## Semana 3

### XPath y Css selectors

- XPath
- Css selectors
- Introduccion a Testng
- Orden de ejecución (prioridades)
- Before y After (test, class, etc.)
- Ejemplos agregando testng
- Priority



Selenium WebDriver



## Semana 4

### TestNG

- Annotations
- Expected Exception
- @Parameters
- @DataProvider
- @Factory
- @DataProvider
- Test Groups



Selenium WebDriver



## Semana 5

### TestNG

- Disable/Ignore Tests
- Parallel Tests
- Dependent Tests
- Timeout Tests
- Runners (Test suits)
- Dropdowns
- Alert Popups



Selenium WebDriver



## Semana 6

- Clase WebElementHelper
- Ordenando el código
- Refactoring de clases usando utilities
- Fakers



Selenium WebDriver





## Semana 7

- Patron Page Object.
- Base page usando Herencia
- Page Factory



Selenium WebDriver



## Semana 8

- Ejercicios de integración



Selenium WebDriver



## Sitios a utilizar



## Metodología de Trabajo

- Teóricos
- Prácticos
- Tareas



# Introducción a Java



## Qué es un programa?

Serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento físico y/o lógico de una computadora

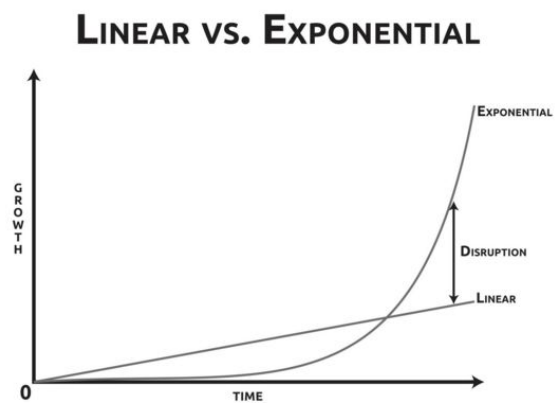


# Qué es Java?

Es un lenguaje!! Tiene los mismos desafíos que aprender un nuevo idioma



Practicar, Practicar y Practicar....



## En que consiste un programa?

- En pedir que el usuario ingrese información
- Procesar los datos ingresados
- Mostrar los resultados

Ejemplos:

- Pedir el ingreso de un número y mostrar su doble
- Pedir el nombre de usuario
- Pedir la fecha de nacimiento

## Qué tiene un programa?

Inicio

X

X

Fin

# Operadores básicos

Operator	Name	Description	Example
+	Addition	Adds together two values	$x + y$
-	Subtraction	Subtracts one value from another	$x - y$
*	Multiplication	Multiplies two values	$x * y$
/	Division	Divides one value from another	$x / y$
%	Modulus	Returns the division remainder	$x \% y$
++	Increment	Increases the value of a variable by 1	$++x$
--	Decrement	Decreases the value of a variable by 1	$--x$

## Ejemplos de los operadores básicos

`Suma = suma + 4;`

`Edad = 2020 - año`

`Iva = sueldo * 0,22`

## Comparadores lógicos

Operator	Name	Example
==	Equal to	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

## Operadores lógicos

Necesitamos evaluar si ciertas sentencias son verdadera o falsas para saber si algo va a ocurrir o no...

Dos operadores:

“AND”

“OR”

## Operadores lógico: “Y”

“Y”  $\rightarrow$  Hoy es martes y es año 2020

- Esta oracion, es verdadera o falsa? De que depende?

## Operadores lógico: “Y”

“Y”  $\rightarrow$  Hoy es domingo y es año 2020

- Si una de ellas es falsa, podemos decir que todo es falso
- El operador Y es verdadero, si y sólo si, ambas partes de la condición son verdaderas



## Operadores lógico: “O”

“O” → Hoy es martes o es año 2020

- Esta oración, es verdadera o falsa? De que depende?

## Operadores lógico: “O”

El operador lógico “O”, es verdadero cuando una de las dos condiciones son verdaderas.

La única situación en la que una condición evaluada con “O” es falsa, es cuando ambos componentes son falsos.

Ejemplos??

## Operadores lógico: “O”

Hoy es lunes o sábado?.

⇒ La oración es verdadera

Hoy domingo o lunes

⇒ La oración es falsa

Hoy lunes o es enero

⇒ La oración es falsa

## Posibles ejercicios con “O” e “Y”

- 1) Mostrar los números múltiplos de 2 y mayores a 9
- 2) Mostrar los números múltiplos de 2 o mayores a 9
- 3) Mostrar los números no múltiplos de 2 y mayores a 9
- 4) Mostrar los números no múltiplos de 2 o mayores a 9

# Variables en Java

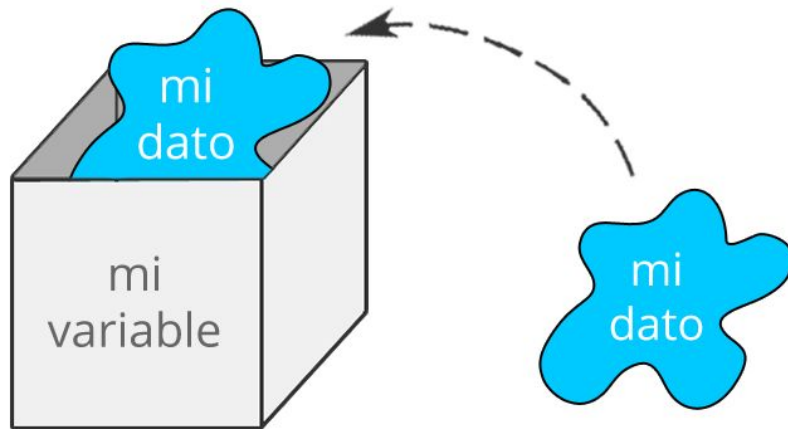
## Las variables

Guardan datos en memoria

Tienen un nombre, un tipo de dato y un valor

Su valor puede cambiar, de ahí el término “variable”

## Las variables



## Las variables

Java es un lenguaje tipado, es decir, debemos definir el “tipo de dato” que vamos a guardar en la variable

Es decir, si definimos que vamos a guardar un número, no podemos almacenar una letra.

```
tipo nombreVariable = valor;
```

## Definir variables

```
String nombre = "Juan";
```

```
int edad = 19;
```

## Sobre-escribir variables

```
int edad = 20;
```

```
edad = 26;
```

# Ejemplos

```
int myNum = 5;  
  
float myFloatNum = 5.99f;  
  
char myLetter = 'D';  
  
boolean myBool = true;  
  
String myText = "Hello";
```

## Tipos de datos simples (primitivos)

- short
- int
- long
- float
- boolean
- char

# Tipos primitivos

## short

Representa un tipo de dato de 16 bits. (-32,768 a 32,767)

## int

Es un tipo de dato de 32 bits (-2,147,483,648 a 2,147,483,647)



# Tipos primitivos

## long

Es un tipo de dato de 64 bits

## float

Es un tipo dato para almacenar números en coma con hasta 7 decimales.



# Tipos primitivos

## boolean

Sirve para definir tipos de datos booleanos. Es decir, aquellos que tienen un valor de true o false. Ocupa 1 bit de información.

## char

Es un tipo de datos que representa a un carácter (una letra o número)



# Sentencias a utilizar en Pseudocodigo

INICIO → inicio del programa

TECLADO → para reconocer el teclado

GUARDAR → guarda el valor en memoria

MOSTRAR → imprimir algo en pantalla

FIN → inicio del programa





# Ejercicios