**Curso Java**

**INSTRUCTOR**

**LUCAS tomas GOICOECHEA FORTE**

**OBJETIVOS**

* Aprender el lenguaje Java y su entorno para el desarrollo de aplicaciones a través del aprendizaje de los conceptos básicos, características, uso de objetos y clases, flujo de control, Strings, Arrays, paquetes y módulos, manejo de excepciones, interfaces, etc.
* Aprender del IDE Eclipse, su entorno, características, funciones y cómo utilizarlo junto con Java para el desarrollo de las aplicaciones.
* Conocer nuevas características y funciones del java que permiten optimizar y agilizar el desarrollo de las funciones y su integración con otras herramientas para un fácil desarrollo de aplicaciones.

**DURACIÓN**

65 HORAS

**TEMAS**

JAVA -> jvm - jdk - jre

<https://www.oracle.com/in/java/technologies/javase/jdk11-archive-downloads.html>

COMANDO PARA CHEQUEAR VERSIÓN DE JAVA java -version

IDE ECLIPSE (STS) -> <https://www.eclipse.org/downloads/packages/release/2022-09/r/eclipse-ide-enterprise-java-and-web-developers>

GIT -> <https://github.com/git-for-windows/git/releases/download/v2.38.0.windows.1/Git-2.38.0-64-bit.exe>

<https://docs.github.com/es/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

REPOSITORIO GIT

git clone git@github.com:lucasgoicoechea/JavaIndra.git

- ORM -SPRING FRAMEWORK - MAVEN - TESTING - INTEGRIDAD CONTINUA

<https://github.com/>

**TEMARIO**

Introducción

Una clase Java sencilla

Nuestro “Hello World” en Java

El Shell de Java (REPL)

Conceptos básicos de Java

Características del lenguaje y de la plataforma

El ciclo de Release de Java

Ciclo de vida de un programa

El Java SE Development Kit (JDK)

Conceptos básicos de clases y objetos

Modelo de objetos y OOP

Clases, referencias e instanciación

Añadir datos a la definición de una clase

Añadir métodos (comportamiento)

Conceptos avanzados de clases y objetos

Acceder a datos, la variable “this”

Encapsulación y control de acceso público y privado

Constructores e inicialización

Miembros static de una clase

Inferencia de Tipos (Java 10+)

Scopes, Blocks, Referencias a Objetos

Type-safe Enums

Flujo de Control

Branching: if, if-else, switch

Iteración: while, do-while, for, break, continue

Strings, Arrays y Dates/Times

String, StringBuffer, StringBuilder

Arrays, Primitive Arrays, Arrays de tipos por referencia

varargs

LocalDate/LocalTime (Java 8+)

Paquetes y Módulos

Visión general de los Paquetes – Utilizar paquetes para organizar el código

Sentencias import

Crear paquetes, estructura de directorios

Visión general de los módulos de Java 11

Definir módulos, requires y exportaciones

Módulo Path y Classpath – Diferencias y coexistencia

Composición y herencia

Utilizar composición

Composición/HAS-A, delegación

Utilizar herencia y polimorfismo

IS-A, extends, características de herencia, sobrecargar métodos

Clase Object

Clases abstractas

Interfaces

Utilizar interfaces para definir Tipos

Interfaces y Clases Abstractas

Métodos predeterminados y estáticos

Excepciones

Excepciones y jerarquía de excepciones

try y catch

Manejar excepciones

Flujo del programa y excepciones

finally

Colecciones y genéricos Java

El Framework Collections y su API

Colecciones y genéricos en Java

Collection, Set, List, Map, Iterator

Autoboxing

Colecciones de objetos (no genéricos)

Utilizar ArrayList, HashSet y HashMap

Bucle for-each

Procesar elementos con un Iterator

Más sobre genéricos

Características adicionales

Anotaciones

Expresiones Lambda y referencias de métodos

Características adicionales

Streams E/S

Readers y Writers

Filter Streams

Byte Streams

Salida formateada

La clase File

Nuevas API I/O (NIO)

IDE ECLIPSE

Inicio con Eclispse

Introducción

Instalación

Características

Crear un proyecto

Configurar las preferencias

Paquetes, Clases

Ejecutar un programa

Operaciones avanzadas

Prueba con JUnit

Importar un archivo JAR

Borrar/remover un proyecto

Importar y configurar un proyecto

El editor y los accesos directos

Archivos y Edición de Menús

Fuentes

Navegación y Búsqueda

Construir y Ejecutar proyectos

GIT

Que es un sistema de control de versiones

Git como sistema de control de versiones

Ventajas respecto a otros sistemas

Accesibilidad

Escalabilidad

Repositorios

Local

Remoto

Comandos de GIT

Clone

Checkout

Add

Commit

Pull

Push

Fetch

Branches

Concepto de Branches

Creación de Branches

Tags

Concepto de Tag

Creación de Tags

Pull Request

Fusión de Branches

Merge

Maven

Que es Maven

Configuración de Maven

Artefact\_id

Group\_id

Version

Scopes

El pom.xml

Herencia en Maven (Parent)

Repositorios

Locales

Remotos

Comandos de maven

Clean

Compile

Install

Build

Deploy

Test

Artefactos

Que son Artefactos

Para que sirven los Artefactos

Arquetipos

Que son Arquetipos

Construcción de Arquetipos

Plugins

Que son Plugins

Creación de Plugins

Ejecución de plugins

Spring

Spring core

Concepto de Bean

Configuración de Spring

Con XML

Con Annotations

Con clases

Spring AOP

Interceptores (Aspect)

PointCuts

Advices

Before

After

After Returning

After Throwing

Around

Spring Boot

Starters

Spring Boot MVC

Controladores

Vistas

Flujo de datos

Vista -> controlador

Controlador -> vistas

Taglibs en spring MVC

Spring boot JPA

Concepto de ORM

EntityManagerFactory

EntityManager

Entitys

Mapeo relacional

ManyToOne

ManyToMany

OneToMany

OneToOne

Transactions

Commit

Flush

Query

Nativas

JPAQL

Concurrencia

Spring Boot Rest

Controladores Rest

EndPoints

Json

Methods

GET

POST

PUT

PATCH

DELETE

Hateoas

Arquitectura de una api rest

Lombok

Manejo de propiedades

Manejo de constructores

Testing

Junit

Test

Assert

Before

After

BeforeClass

AfterClass

Suites

Ejecución en IDE

Ejecución con maven

Mockito

¿Qué es Mockito?

¿Para qué necesitamos Mockito?

Métodos de Mockito para simular Servicios o clases externas

Crear y testear

Métodos de Mockito para simular Servicios (métodos) que terminan en error

Test unitario con Mockito

Jenkins

Instalación

Plugins

Proyectos

Configuración

Ejecución

Directa

Temporizada

Conexión con Sonar

GIT + Maven + Jenkins (Integración continua)