Temario Full Stack

Fase: Java 8 APX

Total Hours (Módulos 1 y 2): 160 h

MÓDULO 1

- JUnit
- Mockito
- Spring Core
- Spring MVC
- Spring Rest
- Spring Data JPA

MÓDULO 2

- Spring Batch
- Git (Command Line and GUI Tool)
- Maven
- Relational Database
- MongoDB
- Understanding HTTP/HTTPS protocols
- HTTP verbs (POST, PUT, GET, DELETE, PATCH)
- Response codes
- REST vs SOAP: differences and advantages
- Creating RESTful services with Java

MÓDULO 3*

- Java class structure
- Package declarations and imports
- Creating objects
- Primitive types vs. reference types
- Variable declaration, initialization, and scope
- Garbage collection
- Java operators (arithmetic, unary, assignment, relational, logical)
- Control flow statements (if-then, if-then-else, switch)
- Iteration statements (while, do-while, for)
- Advanced flow control (break, continue, labels)
- String manipulation

**De estudio independiente por parte del participante*

MÓDULO 4*

- StringBuilder and StringBuffer
- Java arrays and ArrayList
- Wrapper classes and autoboxing
- Working with dates and times
- Method design and overloading
- Access modifiers
- Static methods and fields
- Passing data among methods
- Constructors
- Encapsulation and immutability
- Inheritance
- Abstract classes
- Interfaces

**De estudio independiente por parte del participante*

MÓDULO 5*

- Polymorphism
- Casting objects
- Exception types
- Try-catch-finally blocks
- Throwing and handling exceptions
- Creating nested classes (inner classes, anonymous classes, static nested classes)
- Coding equals(), hashCode(), and toString()
- Working with Enums
- Using instanceof
- Designing interfaces and functional interfaces
- Understanding design principles (JavaBeans, is-a and has-a relationships)
- Working with design patterns (Singleton, Immutable Objects, Builder, Factory)
- Dependency Injection

**De estudio independiente por parte del participante*

MÓDULO 6*

- Advanced use of collections
- Working with generics (classes, interfaces, methods)
- Using Lists, Sets, Maps, and Queues
- Comparing collection types
- Comparator vs. Comparable
- Advanced use of lambdas
- Working with built-in functional interfaces
- Using Optional
- Working with Streams (creating, terminal operations, intermediate operations)
- Working with primitive streams
- Advanced stream pipeline concepts
- Using multi-catch and try-with-resources
- Creating custom exceptions
- Working with assertions
- Introducing threads and concurrency
- Creating and managing threads
- Using ExecutorService

**De estudio independiente por parte del participante*