

## CERTIFICACIÓN PROGRAMADOR JAVA II 1Z0-819

***SOMOS Y FORMAMOS EXPERTOS EN T.I***



**Metodología**

**100% PRACTICO**



**Duración**

**25 HRS.**

# ACERCA DEL WORKSHOP

## OBJETIVOS PRINCIPALES

- Obtienes la preparación para la certificación OCPJP
- Obtienes acceso a más de 400 preguntas similares al examen oficial de certificación

## OBJETIVOS SECUNDARIOS

- Conoces las nuevas características del lenguaje con ejemplos prácticos

## PREREQUISITOS

- Haber cursado el primer módulo Certificación Programador Java I 1Z0-819
    - Experiencia de 1 año programando con Java o haber cursado nuestro [Diplomado Java Empresarial](#).
- [¿Cuál es mi nivel en programación? clic aquí](#)

# ¡NUNCA DEJES DE APRENDER!

## 1. Java Fundamentals

- 1.1 Create and use final classes
- 1.2 Create and use inner, nested and anonymous classes
- 1.3 Create and use enumerations

## 2. Exception Handling and Assertions

- 2.1 Use try-with-resources construct
- 2.2 Create and use custom exception classes
- 2.3 Test invariants by using assertions

## 3. Java Interfaces

- 3.1 Create and use interfaces with default methods
- 3.2 Create and use interfaces with private methods

## 4. Generics and Collections

- 4.1 Use wrapper classes, auto-boxing and auto-unboxing
- 4.2 Create and use generic classes, methods with diamond notation and wildcards
- 4.3 Describe Collections Framework and use key collection interfaces
- 4.4 Use Comparator and Comparable interfaces
- 4.5 Create and use convenience methods for collections

## 5. Functional Interface and Lambda Expressions

- 5.1 Define and write functional interfaces
- 5.2 Create and use lambda expressions including statement lambdas, local-variable for lambda parameters

## 6. Java Stream API

- 6.1 Describe the Stream interface and pipelines
- 6.2 Use lambda expressions and method references

## 7. Built-in Functional Interfaces

- 7.1 Use interfaces from java.util.function package
- 7.2 Use core functional interfaces including Predicate, Consumer, Function and Supplier
- 7.3 Use primitive and binary variations of base interfaces of java.util.function package

## 8. Lambda Operations on Streams

- 8.1 Extract stream data using map, peek and flatMap methods
- 8.2 Search stream data using search findFirst, findAny, anyMatch, allMatch and noneMatch methods
- 8.3 Use Optional class
- 8.4 Perform calculations using count, max, min, average and sum stream operations
- 8.5 Sort a collection using lambda expressions
- 8.6 Use Collectors with streams, including the groupingBy and partitioningBy operation



## 9. Migration to a Modular Application

9.1 Migrate the application developed using a Java version prior to SE 9 to SE 11 including top-down and bottom-up migration, splitting a Java SE 8 application into modules for migration

9.2 Use `jdeps` to determine dependencies and identify ways to address the cyclic dependencies

## 10. Concurrency

10.1 Create worker threads using `Runnable`, `Callable` and use an `ExecutorService` to concurrently execute tasks

10.2 Use `java.util.concurrent` collections and classes including `CyclicBarrier` and `CopyOnWriteArrayList`

10.3 Write thread-safe code

10.4 Identify threading problems such as deadlocks and livelocks

## 11. Services in a Modular Application

11.1 Describe the components of Services including directives

11.2 Design a service type, load services using `ServiceLoader`, check for dependencies of the services including consumer and provider modules

## 12. I/O (Fundamentals and NIO2)

12.1 Read data from and write console and file data using I/O Streams

12.2 Use I/O Streams to read and write files

12.3 Read and write objects by using serialization

12.4 Use `Path` interface to operate on file and directory paths

12.5 Use `Files` class to check, delete, copy or move a file or directory

12.6 Use Stream API with Files

## 13. Secure Coding in Java SE Application

13.1 Prevent Denial of Service in Java applications

13.2 Secure confidential information in Java application

13.3 Implement Data integrity guidelines- injections and inclusion and input validation

13.4 Prevent external attack of the code by limiting Accessibility and Extensibility, properly handling input validation, and mutability

13.5 Securely constructing sensitive objects

13.6 Secure Serialization and Deserialization

## 14. Database Applications with JDBC

14.1 Connect to databases using JDBC URLs and `DriverManager`

14.2 Use `PreparedStatement` to perform CRUD operations

14.3 Use `PreparedStatement` and `CallableStatement` APIs to perform database operations

## 15. Localization

15.1 Use `Locale` class

15.2 Use resource bundles

15.3 Format messages, dates, and numbers with Java

## 16. Annotations

16.1 Describe the purpose of annotations and typical usage patterns

16.2 Apply annotations to classes and methods

16.3 Describe commonly used annotations in the JDK

16.4 Declare custom annotations

## CERTIFICADO DIGITAL


Obtén una constancia que avala tu preparación, si cumples con la asistencia a tu capacitación y elaboras el proyecto final de cada curso, bootcamp o diplomado.

Registrado por la Secretaria del Trabajo y Previsión Social (México).



***¡Te esperamos!***

 55 5211 6931

 +52 55 6186 8835

 TecGurusNet