



Gracias a

[openwebinars.net/cert/eC9J](https://openwebinars.net/cert/eC9J)



OpenWebinars certifica que

**Adrián Real Palacios**

Ha superado con éxito

**Desarrollador Java**

Duración de la ruta

**90 horas**

Fecha de expedición

**15 mayo 2023**

A handwritten signature in black ink, appearing to read 'Manuel Agudo', positioned over a large, light gray watermark of the OpenWebinars logo.

CEO de OpenWebinars

**Manuel Agudo**

# Contenido

## Desarrollador Java

### 1. Curso de Java 8 desde cero

#### 1. Comienza con Java

Presentación del profesor y curso

Instalación de Java

Posibles entornos de desarrollo

Hola Mundo

Algunos elementos de sintaxis

#### 2. Tipos de datos, operadores y estructuras de control

Variables

Manipulación de números, caracteres y otros valores I

Manipulación de números, caracteres y otros valores II

Manipulación de números, caracteres y otros valores III

Manipulación de números, caracteres y otros valores IV

Estructura de decisión

Estructura de repetición

Práctica: Creando una calculadora

#### 3. Clases y objetos

Orientación de objetos



Los constructores

Getters y setters

Ciclo de vida de un objeto

Uso de clases envoltorio

#### **4. Manipulación y tratamiento de datos**

Conoce Java API Docs

Uso de la clase String

Manejo de Arrays

Práctica: Creando el juego de hundir la flota

#### **5. Revisitando clases y objetos**

Argumento de un método y tipos de retorno

Paso de argumento por valor y referencia

Modificadores de acceso

Métodos estáticos y variables estáticas

Sobrecarga de métodos y constructores

#### **6. Uso de herencia**

Qué es la herencia y los beneficios que aporta

Polimorfismo

Uso de super para acceder a un objeto y sus constructores

Clases abstractas e interfaces I

Clases abstractas e interfaces II

#### **7. Manejo de excepciones**

Excepciones y errores

Tratamiento de excepciones

Excepciones más comunes

Lanzamiento y propagación de excepciones

#### **8. Algunas clases del API de Java**

Manipulación de datos con StringBuilder

Manejo de fechas

Colecciones ArrayList



Introducción a las expresiones lambda

Práctica: Creando un gestor de aparcamiento de coches

## **2. Java 8 para programadores Java**

### **1. Introducción al diseño de clases en Java**

Presentación del profesor y curso

Implementación de la encapsulación

Implementación de herencia con modificadores de acceso y composición

Polimorfismo

Sobreescribir los métodos hashCode, equals y toString de la clase Object

Uso del ámbito static para variables, métodos y clases

Clases singleton y clases de inmutables

### **2. Diseño avanzado de clases**

Clases y métodos abstractos

Código que usa final

Clases internas, locales y anónimas

Uso de enumeraciones

### **3. Clases genéricas y colecciones**

Creación de una clase genérica

Creación y uso de List, Set y Map

Interfaces Comparable y Comparator

### **4. Expresiones lambda e interfaces funcionales**

Interfaces funcionales

Predicate, Consumer, Function y Supplier

### **5. Java Stream API**

Introducción al API Stream

Métodos de búsqueda de datos

Métodos de datos, cálculo y ordenación

Uso de Map yflatMap

Uso de la clase Collector

Uso de streams y filtros

Referencias a métodos con stream

Práctica: Todos los elementos del API stream trabajando conjuntamente

## **6. Java I/O y NIO.2**

Uso de flujos de salida

Uso de flujos de entrada

Trabajar con la clase file

Trabajar con la interfaz path

Utilizar métodos de la clase files

Uso de API Stream con NIO.2

Práctica de uso de Java NIO.2

## **7. Concurrencia en Java**

Programación concurrente

Programación concurrente

Runnable, Callable, y ExecutorService

Código sincronizado y uso de valores atómicos

Colecciones concurrentes

## **8. Bases de datos y Java**

Conexión con una base de datos

Lanzar consultas y procesar resultados

Uso de RowSet

## **3. Nuevas funcionalidades después de Java 8**

### **1. Taller**

Nuevas funcionalidades después de Java 8

## **4. ¿Cuál es el mejor IDE para programar en Java?**

### **1. Taller**

¿Cuál es el mejor IDE para programar en Java?

## **5. Cómo elegir el mejor JDK**

### **1. Taller**



¿Cómo elegir el mejor JDK?

## **6. Documentación del código con Javadoc**

### **1. Taller**

Documentación del código con Javadoc

## **7. Empaqueta tu aplicación Java en un fichero JAR**

### **1. Taller**

Empaqueta tu aplicación Java en un fichero JAR

## **8. Java 18: Colecciones y Genéricos**

### **1. Taller**

Java 18: Colecciones y Genéricos

## **9. Java 18: Estructuras de datos**

### **1. Taller**

Java 18: Estructuras de datos

## **10. Programación funcional en Java 18: Métodos de referencia**

### **1. Taller**

Programación funcional en Java 18: Métodos de referencia

## **11. Programación funcional en Java 18: API Stream**

### **1. Taller**

Programación funcional en Java 18: API Stream

## **12. Programación funcional en Java 18: Interfaces funcionales y lambdas**

### **1. Taller**

Programación funcional en Java 18: Interfaces funcionales y lambdas

## **13. Programación funcional en Java 18: API Optional**

### **1. Taller**

Programación funcional en Java 18: API Optional

## **14. Java 18: API java.time**

### **1. Taller**

Java 18: API java.time



## **15. Java 18: Entrada y salida IO/NIO**

### **1. Taller**

Java 18: Entrada y salida IO / NIO

## **16. Curso de Java y MySQL JDBC: Operaciones básicas**

### **1. Introducción**

Presentación

Bases de datos relacionales

API JDBC

Instalación con Docker

Instalación con Xampp

Creación de una BBDD en MySQL

Test autoevaluación

### **2. Establecimiento de la conexión**

Creación del proyecto

Carga del driver

Establecimiento de la conexión

Test autoevaluación

### **3. Sentencias SQL**

Tipos de sentencias SQL

Creación de un statement

Test autoevaluación

### **4. Inserción de registros**

Sentencia SQL INSERT INTO

Test autoevaluación

### **5. Consulta de registros y cierre de conexión**

Sentencia SQL SELECT FROM

Cierre de la conexión

Test autoevaluación

### **6. Conclusiones**



Conclusiones

## **17. Curso de Java y MySQL JDBC: Elaboración de un CRUD**

### **1. Introducción**

Presentación

Creación de una base de datos en MySQL

Patrón DAO

Sentencias precompiladas con PreparedStatement

Test autoevaluación

### **2. Establecimiento de la conexión**

Creación del proyecto

Establecimiento de la conexión con la base de datos

Test autoevaluación

### **3. Desarrollo del DAO**

Creación del objeto y método de inserción

Método de consulta: Introducción

Método de consulta: Conclusión

Método de actualización

Método de borrado

Test autoevaluación

### **4. Desarrollo del resto de la aplicación**

Desarrollo del menú

Método de inserción

Método de consulta de todos los registros

Método de consulta por la clave primaria

Método de actualización

Método de borrado

Test autoevaluación

### **5. Conclusiones**

Conclusiones



## **18. Curso de Java y MySQL JDBC: Procedimientos almacenados y transacciones ACID**

### **1. Introducción**

Presentación

Creación de la BD en MySQL

Test autoevaluación

### **2. Procedimientos almacenados**

¿Qué es un procedimiento almacenado?

Creación de un procedimiento almacenado

CallableStatement

Ejecución del procedimiento almacenado

Test autoevaluación

### **3. Transacciones ACID**

¿Qué es una transacción?

Autocommit

Niveles de aislamiento

Creación y confirmación de una transacción

Deshacer las últimas operaciones

Test autoevaluación

### **4. Conclusiones**

Conclusiones

## **19. Java 18: Concurrencia**

### **1. Taller**

Java 18: Concurrencia

## **20. Java 18: Colecciones concurrentes**

### **1. Taller**

Java 18: Colecciones concurrentes

## **21. Modularización con Java Jigsaw**

### **1. Taller**

Modularización con Java Jigsaw



## **22. Java 18: Networking**

### **1. Taller**

Java 18: Networking

## **23. Crea aplicaciones de escritorio con JavaFX**

### **1. Taller**

Crea aplicaciones de escritorio con JavaFX

## **24. Java 18: Recursión**

### **1. Taller**

Java 18: Recursión

## **25. Java 18: Programación funcional avanzada**

### **1. Taller**

Java 18: Programación funcional avanzada

## **26. Java 18: Concurrencia avanzada**

### **1. Taller**

Java 18: Concurrencia avanzada

## **27. Java 18: Paralelización**

### **1. Taller**

Java 18: Paralelización

## **28. Java 18: Frameworks de logging**

### **1. Taller**

Java 18: Frameworks de logging

## **29. Curso de Introducción a pruebas automáticas con JUnit 5**

### **1. Introducción**

Presentación

¿Qué es el testing y por qué debemos hacerlo?

Pirámide del testing y JUnit

### **2. Primeros pasos**

Crear proyecto Maven con JUnit 5



Ejecutar primer test mediante IDE

Ejecutar primer test mediante terminal

### **3. Aserciones**

¿Qué son las aserciones?

Aserciones elementales

Aserción de iterables

Aserción Throws

Aserción Timeout

### **4. Agrupando aserciones y asunciones**

Agrupando aserciones

¿Qué son las asunciones?

### **5. Anotaciones**

¿Qué son las anotaciones @Test y @DisplayName?

Aplicando las anotaciones a Selenium

Anotaciones @BeforeAll y @AfterAll

Anotaciones @BeforeEach y @AfterEach

Repitiendo tests mediante @Repeated

Alimentando con datos mediante @ParameterizedTest

### **6. Filtrando y agrupando tests**

Filtrando tests mediante @Tag

Agrupando tests mediante @Nested

## **30. Testing avanzado con JUnit 5**

### **1. Taller**

Testing avanzado con JUnit 5

## **31. Curso de Clean Code**

### **1. Herramientas**

Presentación del curso y del profesor

GIT: Control de versiones

Entornos de desarrollo



Pruebas unitarias - junit

## **2. Clean Code**

Introducción

Comentarios

Nombres

Métodos

Clases

Arquitectura

## **3. Refactorización**

Introducción

Code Smells I

Code Smells II

Renombrar variables y métodos

Ocultar propiedades y métodos

Magic Numbers

Extraer método

Método Inline

Convertir parámetros en objetos

Replace Temp with Query

Introducir variable explicativa

Separar variable temporal

Borrar las asignaciones a parámetros

Reemplazar método con objeto de método

Descomponer condicional

Consolidar expresión condicional

Consolidar fragmentos de condicional duplicados

Eliminar bandera de control

Reemplazar condicionales anidadas con cláusulas guard

Reemplazar condicional con polimorfismo

Introducir objeto nulo



Separar consulta de modificación

Parametrizar método

Reemplazar parámetro con métodos explícito

Sustituir algoritmo

Extraer clase

Inline Class

Ocultar delegado

Introducir método externo

Introducir extensión local

Reemplazar un dato con un objeto

Encapsular colección

Pull Up

Push Down

Reemplazar array con objeto

#### **4. Patrones de Software**

Introducción

Creacionales singleton

Creacionales factory

Creacionales abstract factory

Creacionales builder

Creacionales Prototype

Estructurales adapter

Estructurales bridge

Estructurales composite

Estructurales decorator

Estructurales Façade

Estructurales flyweight

Estructurales proxy

Comportamiento Chain Of Responsibility

Comportamiento Command

Comportamiento interpreter

Comportamiento iterator

Comportamiento mediator

Comportamiento memento

Comportamiento observer

Comportamiento State

Comportamiento Strategy

Comportamiento Template Method

Comportamiento Visitor

## **5. Principios SOLID**

Introducción

SRP: Single Responsibility Principle

OCP: Open/Closed Principle

LSP: Liskov Substitution Principle

ISP: Interface Segregation Principle

DIP: Dependency Inversion Principle

Principios LEAN

## **32. Java 18: Principios SOLID**

### **1. Taller**

Java 18: Principios SOLID