

# Spring Certified Professional

**Nro. Horas: 40 horas**

**Inversión:**

1. Pago único de S/1,000.00 o USD 268.00

## **Acerca del Curso**

Preparación para la Certificación Spring Professional.

## **Objetivos del Curso**

Después de este curso, los participantes estarán familiarizados con:

- Fundamentos del contenedor
- Programación Orientada a Aspectos (AOP)
- Administración de Datos
- Testing
- Seguridad
- Spring Boot

## **Metodología**

- Clases con profesor en vivo en el horario definido por la academia.
- Material y ejercicios por cada clase.
- Simulacros para dar el [examen de certificación que cuesta US\\$ 250.00](#).
- Soporte vía WhatsApp del Instructor.

## **Para quien es este Curso**

- Para todos los profesionales interesados en la [certificación Spring Professional](#).

## **Pré-requisitos**

- Conocimientos de Java 8 o superior.
- Conocimientos básicos de Spring.

### **Formas de Pago:**

Transferencia Bancaria a JOSE AMADEO MARTIN DIAZ DIAZ:

- BCP 191-30759925-0-29 (Ahorro Soles)
- BBVA 0011-0814-0227145957 (Ahorro Soles)
- SCOTIABANK 880-2490022 (Ahorro Soles)
- INTERBANK 200-3116727850 (Ahorro Soles)

PayPal:

- Enviar el pago a [informes@joedayz.pe](mailto:informes@joedayz.pe)

Pago con Tarjeta de Débito y Crédito

- Solicitar el pago con tarjeta de débito y crédito al +51 939 965 148.

### **Consultas:**

Si tiene alguna consulta o comentario nos puede escribir al WhatsApp al +51939965148.

## Tópicos a Revisar

### Section 1 – Spring Core

- Objective 1.1 Introduction to Spring Framework
- Objective 1.2 Java Configuration
  - 1.2.1 Define Spring Beans using Java code
  - 1.2.2 Access Beans in the Application Context
  - 1.2.3 Handle multiple Configuration files
  - 1.2.4 Handle Dependencies between Beans
  - 1.2.5 Explain and define Bean Scopes
- Objective 1.3 Properties and Profiles
  - 1.3.1 Use External Properties to control Configuration
  - 1.3.2 Demonstrate the purpose of Profiles
  - 1.3.3 Use the Spring Expression Language (SpEL)
- Objective 1.4 Annotation-Based Configuration and Component Scanning
  - 1.4.1 Explain and use Annotation-based Configuration
  - 1.4.2 Discuss Best Practices for Configuration choices
  - 1.4.3 Use `@PostConstruct` and `@PreDestroy`
  - 1.4.4 Explain and use “Stereotype” Annotations
- Objective 1.5 Spring Bean Lifecycle
  - 1.5.1 Explain the Spring Bean Lifecycle
  - 1.5.2 Use a `BeanFactoryPostProcessor` and a `BeanPostProcessor`
  - 1.5.3 Explain how Spring proxies add behavior at runtime
  - 1.5.4 Describe how Spring determines bean creation order
  - 1.5.5 Avoid issues when Injecting beans by type
- Objective 1.6 Aspect Oriented Programming
  - 1.6.1 Explain the concepts behind AOP and the problems that it solves
  - 1.6.2 Implement and deploy Advices using Spring AOP
  - 1.6.3 Use AOP Pointcut Expressions
  - 1.6.4 Explain different types of Advice and when to use them

### Section 2 – Data Management

- Objective 2.1 Introduction to Spring JDBC
  - 2.1.1 Use and configure Spring’s `JdbcTemplate`
  - 2.1.2 Execute queries using callbacks to handle result sets
  - 2.1.3 Handle data access exceptions
- Objective 2.2 Transaction Management with Spring
  - 2.2.1 Describe and use Spring Transaction Management
  - 2.2.2 Configure Transaction Propagation
  - 2.2.3 Setup Rollback rules
  - 2.2.4 Use Transactions in Tests
- Objective 2.3 Spring Boot and Spring Data for Backing Stores
  - 2.3.1 Implement a Spring JPA application using Spring Boot
  - 2.3.2 Create Spring Data Repositories for JPA

### Section 3 – Spring MVC

- Objective 3.1 Web Applications with Spring Boot
  - 3.1.1 Explain how to create a Spring MVC application using Spring Boot
  - 3.1.2 Describe the basic request processing lifecycle for REST requests
  - 3.1.3 Create a simple RESTful controller to handle GET requests
  - 3.1.4 Configure for deployment
- Objective 3.2 REST Applications

- 3.2.1 Create controllers to support the REST endpoints for various verbs
- 3.2.2 Utilize RestTemplate to invoke RESTful services

#### Section 4 – Testing

##### Objective 4.1 Testing Spring Applications

- 4.1.1 Write tests using JUnit 5
- 4.1.2 Write Integration Tests using Spring
- 4.1.3 Configure Tests using Spring Profiles
- 4.1.4 Extend Spring Tests to work with Databases

##### Objective 4.2 Advanced Testing with Spring Boot and MockMVC

- 4.2.1 Enable Spring Boot testing
- 4.2.2 Perform integration testing
- 4.2.3 Perform MockMVC testing
- 4.2.4 Perform slice testing

#### Section 5 – Security

##### Objective 5.1 Explain basic security concepts

##### Objective 5.2 Use Spring Security to configure Authentication and Authorization

##### Objective 5.3 Define Method-level Security

#### Section 6 – Spring Boot

##### Objective 6.1 Spring Boot Feature Introduction

- 6.1.1 Explain and use Spring Boot features
- 6.1.2 Describe Spring Boot dependency management

##### Objective 6.2 Spring Boot Properties and Autoconfiguration

- 6.2.1 Describe options for defining and loading properties
- 6.2.2 Utilize auto-configuration
- 6.2.3 Override default configuration

##### Objective 6.3 Spring Boot Actuator

- 6.3.1 Configure Actuator endpoints
- 6.3.2 Secure Actuator HTTP endpoints
- 6.3.3 Define custom metrics
- 6.3.4 Define custom health indicators