

## GUION DEL CURSO

**DENOMINACIÓN DE LA ESPECIALIDAD:** DESARROLLO DE APLICACIONES CON SPRING E HIBERNATE

**CÓDIGO DE LA ESPECIALIDAD:** IFCD0014

**CÓDIGO DEL CURSO:** 24/1501

**FECHA DE INICIO:** 24/04/2025

**FECHA DE FINAL:** 23/05/2025

**HORARIO:** 16:00 -21:00 horas

**CALENDARIO (días no lectivos):** 1 y 2 de mayo.

**DURACIÓN (horas totales):** 100 horas

---

### OBJETIVO GENERAL DEL CURSO:

Proporcionar el conocimiento necesario para la creación de proyectos de desarrollo de aplicaciones con tecnologías Spring e Hibernate y el empleo de tecnologías complementarias como Maven, Jenkins, Docker y Kubernetes.

### RELACIÓN DE MÓDULOS (objetivo, contenido y duración):

#### **APACHE MAVEN - 20 HORAS**

##### **OBJETIVO**

Conocer la forma de crear y la configuración de proyectos empleando Maven, enriqueciendo las operaciones con el empleo de plugins para interactuar con sistemas de ejecución de pruebas, de control de versiones, de publicación en repositorios, de despliegues y de automatizaciones de tareas comunes.

##### **CONTENIDO**

- ▶ Visión global de la arquitectura de Maven.
- ▶ Descripción de la funcionalidad que proporciona Maven.
- ▶ Conceptos: proyecto, arquetipo, repositorio local/remoto.
- ▶ Configuración y puesta en marcha.
- ▶ Creación de proyectos con Maven.
- ▶ Fases del ciclo de vida de un proyecto.
- ▶ Configuración de plugins.
- ▶ Estructura de un proyecto Maven: directorios + pom.xml.
- ▶ Gestión de la configuración de un proyecto en diferentes entornos/condiciones mediante perfiles.
- ▶ Gestión de las dependencias de un proyecto.
- ▶ Empleo de los plugins más habituales.
- ▶ Configuración de repositorios propios para publicar proyectos y para descargar dependencias.
- ▶ Empleo de la agregación de proyectos.
- ▶ Empleo de la herencia entre proyectos- Empleo de seguridad en la configuración Maven.
- ▶ Desarrollo de plugins propios.

#### **DESARROLLO DE APLICACIONES CON SPRING E HIBERNATE – 25 HORAS**

##### **OBJETIVO**

Conocer las características de Spring e Hibernate, priorizando los aspectos prácticos y arquitectónicos en la construcción de aplicaciones con un alto grado de flexibilidad y fácil mantenimiento.

##### **CONTENIDO**

###### **Hibernate**

- ▶ ¿Qué es Hibernate?
- ▶ Arquitectura
- ▶ Configuración básica
- ▶ Definición de entidades

- ▶ Anotaciones
- ▶ Hibernate nativo VS Hibernate JPA
- ▶ Definición de relaciones entre entidades
- ▶ Estrategias de fetching
- ▶ HQL
- ▶ Sesiones y transacciones
- ▶ Arquitectura de capa de persistencia
- ▶ Data Access Objects
- ▶ Spring Framework
- ▶ ¿Qué es Spring?
- ▶ Novedades de Spring
- ▶ Arquitectura
- ▶ Patrón de diseño: Inversión de control
- ▶ Patrón de diseño: Factory Pattern
- ▶ Patrón de diseño: Service Locator
- ▶ Patrón de diseño: Inyección de dependencias (CDI)
- ▶ Inyección de dependencias en Spring
- ▶ ¿Qué es un Bean?
- ▶ Cómo crear un proyecto basado en Spring
- ▶ Scopes
- ▶ Bean Callbacks
- ▶ Herencia
- ▶ Autowiring
- ▶ Manejo de eventos
- ▶ Programación orientada a aspectos (AOP)
- ▶ Integración con Spring
- ▶ Spring + Hibernate
- ▶ JSF + Spring VS Spring Web

## INTEGRACIÓN CONTINUA CON JENKINS- 25 HORAS

---

### OBJETIVO

---

Conocer y utilizar la herramienta Jenkins en las diferentes tareas dentro de un proceso de integración y despliegue continuo de los proyectos de software en un entorno ágil.

### CONTENIDO

---

- Introducción a la integración continua
- Ejemplos de integración continua
- Introducción a Jenkins y características
- Despliegue de Jenkins
- Flujo de Integración continua con Jenkins
- Jenkins core
  - Agents
  - Jobs
  - Plugins
- Jenkins jobs
  - Types
    - o Freestyle
    - o Maven
    - o Folder
    - o Pipeline
    - o Multibranch
    - o Organization/Team/Namespace projects
    - o Others
    - o External job (process monitoring)
    - o Multi configuration (matrix freestyle projects)
  - Setup
    - o Freestyle
    - o Pipeline
    - o Multibranch

- o Organization/Team/Namespaces projects
- o filtering traits
- Jenkins Plugins
- Cloud providers
  - o mesos
  - o docker
  - o yet another docker
  - o ec2 (eucalyptus)
  - o openstack
  - o azure
  - o kubernetes
  - o nomad
- Cloud statistics
- Milestone
- Lockable resources
- Metrics
- Notifications
  - o Mail
  - o Slack
- SCM
  - o File system
  - o Git
  - o Mercurial
- Pipeline ecosystem
  - o declarative
- Blueocean
- Pipelines
- Development
  - o steps provided
  - o shared libraries
- Testing
  - o Niveles de testing
  - o Unit tests
  - o Replay
- Debugging
- Monitoring. Monitorización de rendimiento

## ORQUESTACIÓN DE CONTENEDORES DOCKER/KUBERNETES- 30 HORAS

---

### OBJETIVO

---

Construir aplicaciones de forma nativa en la nube utilizando herramientas y tecnologías de contenedores, y desplegar sus aplicaciones en cualquier nube pública, privada o híbrida.

### CONTENIDO

---

#### DOCKER

- Introducción
  - o Contenedores vs Virtualización
  - o Imágenes
  - o Componentes
- Primeros Pasos
  - o Ejecutar contenedores
  - o Inspeccionar contenedores
  - o Modo interactivo
  - o Arrancar/detener contenedores
  - o Exponer puertos
- Introducción a la integración continua
- Contenedores
  - o Crear contenedores sin ejecutarlos
  - o Acciones básicas sobre contenedores
  - o Copias de seguridad

- o Restaurar
- o Inspeccionando contenedores
- o Obteniendo estadísticas de uso
- Imágenes
  - o Introducción
  - o Listar imágenes
  - o Buscar imágenes en repositorio oficiales
  - o Copias de seguridad
  - o Histórico de Imágenes
  - o Eliminar/Restaurar Imágenes
  - o Convertir un contenedor en una imagen
  - o Etiquetado
  - o Publicar imágenes en repositorio Oficial
  - o Publicar imágenes en repositorio Local
  - o Buenas prácticas Dockerfile
- Redes
  - o Introducción
  - o Redes predefinidas
  - o Crear redes
  - o Rango autogenerados
  - o Rango específico
  - o Especificar redes al crear contenedores
  - o Inspeccionar redes
  - o Conectar y desconectar contenedor a una red
- Almacenamiento
  - o Introducción
  - o AULFS
  - o OverlayFS/Overlayfs2
  - o Device Mapper
  - o Brtfs
  - o Volúmenes
  - o Plugin para volúmenes
  - o Persistencia de Volúmenes

## KUBERNETES ORQUESTADOR DE CONTENEDORES

- Introducción a Kubernetes
  - o Arquitectura de Kubernetes
  - o Controladores en kubernetes
- Desplegar objetos en kubernetes
  - o Desplegar contenedores en kubernetes, pod
  - o RC
  - o Deployments/RS
  - o Services
  - o Gestión de Namespaces
  - o Despliegue declarativos en kubernetes
  - o Aplicaciones multicontenedor en kubernetes
  - o Rolling Update
  - o Autoscaling
  - o Secretos
- Almacenamiento con kubernetes
  - o Volúmenes en kubernetes
  - o Asignar un volumen persistente a un pod
  - o emptyDir
  - o hostPath
  - o NFS
- Advanced Topics
  - o Service Discovery
  - o Dns
  - o Ingress Controller
  - o Affinity / Anti-Affinity
  - o NodeSelector using Labels
  - o Healthchecks

- o Liveness and Readiness probe
- o ConfigMap
- o Horizontal Pod Autoscaler
- o Prometheus, exporters, Grafana

## PREVENCIÓN DE RIESGOS LABORALES (Transversal)

### OBJETIVOS

- ▶ Conocer e identificar los riesgos laborales que se consideran generales y el procedimiento para su prevención:
- ▶ Riesgos relacionados con las condiciones de seguridad.
- ▶ Riesgos relacionados con el medio ambiente de trabajo.
- ▶ La carga de trabajo, la fatiga y la insatisfacción laboral.
- ▶ Sistemas de control de riesgos.
- ▶ Planes de emergencia y evacuación.
- ▶ Principios generales de la acción preventiva.
- ▶ Organismos públicos relacionados con la seguridad y salud laboral.

### CONTENIDOS

- ▶ Riesgos relacionados con las condiciones de seguridad.
- ▶ Riesgos relacionados con el medio ambiente de trabajo.
- ▶ La carga de trabajo, la fatiga y la insatisfacción laboral.
- ▶ Sistemas de control de riesgos.
- ▶ Planes de emergencia y evacuación.
- ▶ Principios generales de la acción preventiva.
- ▶ Organismos públicos relacionados con la seguridad y salud laboral.

### METODOLOGÍA DIDÁCTICA (máquinas virtuales, conexiones remotas, versiones software, dominios, suscripciones,...):

Nuestra Formación está basada en métodos docentes que facultan al profesor y orientan al alumno con el fin de conseguir un proceso evolutivo favorable de aprendizaje.

1. **Proceso formativo.** En esta fase se ejecutan distintas metodologías según el momento. Ninguna es excluyente de la otra, sino que se combinan y desarrollan según la necesidad. En todo momento se alternan las explicaciones teóricas con la elaboración de ejercicios prácticos, siempre relacionados con casos reales, combinando métodos cuantitativos y cualitativos.

Según se observa la evolución de los alumnos en la asimilación de los contenidos, se incrementa paulatinamente los niveles de complejidad, asegurándonos de eliminar cualquier laguna que pueda perjudicar la asimilación de conceptos de mayor dificultad. Esta metodología evolutiva se observa claramente en la consecución de logros a través de la selección de ejercicios.

2. **Análisis de resultados:** El plan de formación será evaluado de forma continua, pudiendo cerrar cada módulo con la certeza que los conocimientos han sido comprendidos y aprendidos. En la etapa final se analizará el proyecto en su conjunto, valorando la calidad obtenida y si los efectos conseguidos son los esperados.

Se facilitará el siguiente software en las últimas versiones actualizadas correspondiente al requerido para esta especialidad según pliego:

- PyCharm
- Jupyter Notebook.
- Oracle Database Express Edition
- SQL Server Express
- Mysql Community Server

### MATERIAL DIDÁCTICO DEL ALUMNO (fungible):

Título: Desarrollo de aplicaciones mediante el Framework de Spring|ISBN-  
EAN: 9788499645568  
ISBN: 978-84-9964-556-8  
Autores: Eugenia Pérez Martínez  
Idioma: español  
Formato: Papel  
Editorial: Ra-ma



Dirección General de Formación  
CONSEJERÍA DE ECONOMÍA,  
HACIENDA Y EMPLEO



Título: Continuous Delivery with Docker and Jenkins - Second Edition: secure applications by building complete CI/CD pipelines

ISBN-10: 1838552189

ISBN-13: 978-1838552183

Autores: Rafal Leszko

Idioma: español

Formato: Papel

Editorial Packt Publishing

### **TALLER SOFTSKILL:**

Taller práctico de 10 horas de duración adicionales a las horas establecidas en la especialidad, “**Redes freelance para autoempleo**”, impartido los días 26 y 27 de mayo de 2025.

### **SISTEMA DE EVALUACIÓN (instrumentos, criterios y momentos):**

Al finalizar el último módulo, se realizará una prueba teórico-práctica que contemplará todos los temas impartidos durante el curso.

Dicha prueba teórica contará con 30 ítems y se puntuará sobre un total de 10 puntos, representando cada ítem 0,33 puntos.

Para obtener la calificación final de cada alumno, al finalizar el curso, dicha prueba debe superar los 5 puntos.

Una vez realizada la prueba, a aquellos alumnos que no superen los 5 puntos, se les realizará una prueba final, que contemple todos los contenidos impartidos. Dicha prueba se puntuará sobre un total de 10 puntos, siendo 5 la nota mínima para aprobar el curso.

La no realización de la prueba de evaluación supondrá un 0 en la calificación de ésta.