



1. (JHE-070) Herramientas de Desarrollo

Duración: 30 horas

Perfil: Experto

Dirigido a

Esta acción formativa va dirigida a Programadores, Analistas Java, Jefes de proyecto y Arquitectos.

Requisitos previos

Los alumnos necesitarán tener experiencia con el lenguaje de programación Java, así como de Servlets, JSP y el patrón MVC previos para poder comprender la materia del curso con éxito.

Objetivos

Conocer y aplicar los diferentes tipos de herramientas durante el ciclo de vida de desarrollo de una aplicación. Saber aplicar conceptos de programación extrema en sus proyectos o utilizar metodologías ágiles y sus herramientas para sus desarrollos.

Contenido

1. Introducción y principios Agiles
 - 1.1. Las metodologías ágiles
 - 1.2. Prácticas esenciales en XP
 - 1.3. Test Driven Development
2. Gestión de Proyectos con Maven
 - 2.1. Estructura de un proyecto
 - 2.2. Arquetipos (plantillas de proyectos)
 - 2.3. Tipos de proyecto: Java, Web
 - 2.4. Compilado, test y empaquetado
 - 2.5. El fichero POM (Project Object Model)
 - 2.6. El ciclo de vida del proyecto
 - 2.7. Repositorios de Maven
 - 2.8. Gestión de dependencias
 - 2.9. Jerarquía entre proyectos
 - 2.10. Generando informes y documentación
 - 2.11. Gestión de repositorios: Artifactory/Archiva/Nexus
3. Gradle
 - 3.1. Instalación
 - 3.2. Bases de los scripts de construcción
 - 3.3. Tipos de proyectos
 - 3.4. Artefactos y repositorios
 - 3.5. Línea de comandos vs GUI
 - 3.6. Tareas complejas
 - 3.7. Logging
 - 3.8. Plugins
 - 3.9. Gestión de dependencias
 - 3.10. El ciclo de vida de construcción
4. Repositorios Nexus
 - 4.1. Primeros pasos
 - 4.1.1. Puesta en marcha para la exploración inicial
 - 4.1.2. Familiarizarse con la interfaz de usuario
 - 4.1.3. Utilizando Nexus en desarrollo
 - 4.2. Gestión del repositorio
 - 4.2.1. Comprender los artefactos, las coordenadas del repositorio y gestión de repositorio
 - 4.2.2. El rol del Repositorio Central

- 4.2.3. Configuración de los ajustes del servidor
- 4.2.4. La creación de repositorios y configurar sus opciones
- 4.2.5. Administración de Nexus
- 4.2.6. La diferencia entre proxy y hosting de artefactos
- 5. Uso de GIT para el control de versiones
 - 5.1. Uso de control de versiones y cambios
 - 5.2. Checkout inicial de un repositorio
 - 5.3. update->change->review->commit
 - 5.4. Revisiones e historia de documentos
 - 5.5. Tagging/Branching/Merge
 - 5.6. Desarrollo paralelo
- 6. Integración Continua con Jenkins
 - 6.1. Administración de Jenkins
 - 6.2. Construcción de un proyecto SW
 - 6.3. Construcción de un proyecto maven2
 - 6.4. Construcción de una de matriz proyectos
 - 6.5. Trabajos de supervisión
 - 6.6. Construcción Distribuida
 - 6.7. Usando los plugins de Jenkins
 - 6.8. Interfaz de línea de comandos
 - 6.9. API de acceso remoto
 - 6.10. División de una tarea en trabajos
 - 6.11. Proyectos no Java
 - 6.12. Pipelines
 - 6.12.1. Stashing code
 - 6.12.2. Nodes, steps, stages
 - 6.12.3. Declarative syntax y gestión de errores
 - 6.12.4. Jenkins File
 - 6.12.5. Parallel stages
 - 6.12.6. Parametrización de tareas
 - 6.12.7. Blue Ocean: mejorando la experiencia de usuario Jenkins
- 7. Métricas de proyectos son Sonar
 - 7.1. Configurar un proyecto
 - 7.2. Reglas y estándares de codificación
 - 7.3. Herramientas de Análisis de código: Checkstyle, PMD, Findbugs, Clover, etc
 - 7.4. Métricas de la ingeniería del software
 - 7.5. Control de tests
 - 7.6. Evolución temporal de un proyecto
 - 7.7. Visión del código

Documentación Recomendada

Java Power Tools

John Ferguson Smart
Abril 2008 | 918 páginas
ISBN: 978-0596527938

