

# Integración continua, mejores prácticas de aplicación

Curso práctico de 3 días - 21h

Ref.: DIN - Precio 2024: 1 590€ sin IVA

La integración continua es uno de los aspectos esenciales de los métodos ágiles. Su objetivo es garantizar la calidad y fluidez del ciclo de desarrollo, así como el despliegue de nuevas versiones. Este curso presenta las mejores prácticas y técnicas asociadas para implementarla.

## OBJETIVOS PEDAGÓGICOS

Al término de la formación, el alumno podrá:

Comprender los componentes y principios de la integración continua

Manejo de un gestor de control de versiones

Comprender los mecanismos que intervienen en la creación de software y las herramientas de creación asociadas.

Configuración de un proyecto en un servidor de integración continua

Descifrar las principales métricas de las herramientas de análisis de código

Comprender el papel de los repositorios de artefactos y la gestión de la configuración

## PROGRAMA

última actualización: 04/2024

### 1) Introducción

- Los distintos tipos de entorno: desarrollo, aceptación, producción.
- Gestión de la configuración: sistema y aplicación.
- El papel de los contenedores de aplicaciones : Docker.
- Centralización de configuraciones: puppet, Ansible.
- Industrialización de los despliegues.
- Creación de una plataforma de integración continua.

### 2) Gestión de versiones

- Las características.
- Los distintos gestores de fuentes: centralizados o distribuidos.
- Los problemas de la integración del cambio.
- El papel de las ramas y las etiquetas.

*Trabajo práctico : Manejo de un repositorio Git.*

### 3) Automatización de edificios

- ¿Qué es la construcción de software?
- Construcciones, ciclo de vida del proyecto.
- Herramientas de compilación con Maven.
- Plug-ins, perfiles y módulos.

*Trabajo práctico : Configuración de la construcción de un proyecto sencillo (Maven).*

### 4) Gestión de depósitos

- Funciones de los jefes de depósito.
- Tipos de repositorio: público, interno.

#### PARTICIPANTES

Jefes de proyecto, desarrolladores, arquitectos.

#### REQUISITOS PREVIOS

Conocimientos básicos de desarrollo de software.

#### COMPETENCIAS DEL FORMADOR

Los expertos que imparten la formación son especialistas en las materias tratadas. Han sido validados por nuestros equipos pedagógicos, tanto en el plano de los conocimientos profesionales como en el de la pedagogía, para cada curso que imparten. Cuentan al menos con entre cinco y diez años de experiencia en su área y ocupan o han ocupado puestos de responsabilidad en empresas.

#### MODALIDADES DE EVALUACIÓN

El formador evalúa los progresos pedagógicos del participante a lo largo de toda la formación mediante preguntas de opción múltiple, escenificaciones de situaciones, trabajos prácticos, etc. El participante también completará una prueba de posicionamiento previo y posterior para validar las competencias adquiridas.

#### MEDIOS PEDAGÓGICOS Y TÉCNICOS

- Los medios pedagógicos y los métodos de enseñanza utilizados son principalmente: ayudas audiovisuales, documentación y soporte de cursos, ejercicios prácticos de aplicación y ejercicios corregidos para los cursillos prácticos, estudios de casos o presentación de casos reales para los seminarios de formación.
- Al final de cada cursillo o seminario, ORSYS facilita a los participantes un cuestionario de evaluación del curso que analizarán luego nuestros equipos pedagógicos.
- Al final de la formación se entrega una hoja de presencia por cada media jornada de presencia, así como un certificado de fin de formación si el alumno ha asistido a la totalidad de la sesión.

#### MODALIDADES Y PLAZOS DE ACCESO

La inscripción debe estar finalizada 24 horas antes del inicio de la formación.

#### ACCESIBILIDAD DE LAS PERSONAS CON DISCAPACIDAD

¿Tiene alguna necesidad específica de accesibilidad? Póngase en contacto con la Sra. FOSSE, interlocutora sobre discapacidad, en la siguiente dirección psh-accueil@orsys.fr para estudiar de la mejor forma posible su solicitud y su viabilidad.

- Gestión de entregables (artifacts): releases, snapshots, deployment.
- Organización de depósitos: desarrollo, pruebas, producción.
- Integración de gestores de repositorios con herramientas de compilación y gestión de tareas.
- Ejemplos de soluciones: Nexus, Artifactory...

*Trabajo práctico* : *Publicación de artefactos en Nexus.*

#### 5) Automatización de pruebas

- ¿Qué pruebas y con qué fin?
- Entornos de prueba.
- Presentación de las herramientas de prueba.
- Implantación de la automatización de pruebas.

*Trabajo práctico* : *Configuración de pruebas automatizadas con JUnit.*

#### 6) Servidor de integración continua

- El papel del servidor de integración continua.
- Características principales.
- Gestión del trabajo.
- Dependencia de tareas.
- Servidores: Hudson/Jenkins, CruiseControl, Bamboo...

*Trabajo práctico* : *Configurar el proyecto en un servidor de integración continua Jenkins.*

#### 7) Establecimiento de métricas

- Generación de informes.
- Herramientas de análisis e informes (PMD, Findbugs, Cobertura, Emma, Checkstyle, etc.).
- Publicación de los resultados.

*Trabajo práctico* : *Configuración de herramientas de medición.*

#### 8) Gestión de la información

- Gestionar un ciclo completo.
- Trazabilidad de los cambios.

*Trabajo práctico* : *Usando Sonar.*

## FECHAS

---

Contacto