



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

TEMA

Preparation of the environment for development and
integration continue.

PRESENTADO POR

Perez Bello Vanory Esperanza

GRUPO

10° B

MATERIA

Software Development Process

PROFESOR

Ray Brunett Parra Galaviz

Tijuana, Baja California, 7 Enero del 2025

Preparación del Entorno para el Desarrollo e Integración Continua

La preparación adecuada del entorno es esencial para implementar prácticas de desarrollo e integración continua (CI/CD) de manera efectiva. Estas prácticas permiten a los equipos de desarrollo integrar y entregar cambios de código de forma frecuente y confiable, mejorando la calidad del software y acelerando el tiempo de entrega.

Pasos para Preparar el Entorno:

1. Configuración del Repositorio de Código:

- Utilizar un sistema de control de versiones como Git para gestionar el código fuente.
- Establecer ramas principales y de desarrollo claras para facilitar la colaboración.

2. Automatización de la Construcción:

- Implementar herramientas que automaticen la compilación del código cada vez que se realicen cambios.
- Asegurar que el proceso de construcción sea reproducible y eficiente.

3. Integración de Pruebas Automatizadas:

- Desarrollar y mantener un conjunto de pruebas automatizadas que se ejecuten durante el proceso de integración.
- Incluir pruebas unitarias, de integración y de aceptación para garantizar la calidad del código.

4. Implementación de Herramientas de Integración Continua:

- Seleccionar y configurar herramientas de CI/CD que se integren con el repositorio de código y las herramientas de construcción.

- Estas herramientas deben gestionar el flujo de trabajo desde la integración hasta la entrega del software.

5. Establecimiento de Notificaciones y Monitoreo:

- Configurar notificaciones para alertar a los desarrolladores sobre el estado de las construcciones y las pruebas.
- Implementar sistemas de monitoreo para supervisar el rendimiento y la estabilidad del entorno de integración.

Beneficios de una Preparación Adecuada:

- Detección Temprana de Errores: La integración frecuente y las pruebas automatizadas permiten identificar y corregir errores rápidamente, reduciendo costos y tiempos de desarrollo.
- Mejora de la Calidad del Software: La automatización de pruebas y la integración continua contribuyen a mantener altos estándares de calidad en el código.
- Entrega Más Rápida: La preparación adecuada del entorno facilita una entrega continua y rápida de software, permitiendo a los equipos responder más ágilmente a los cambios en los requisitos y a las necesidades del mercado.
- Colaboración Mejorada: La integración continua fomenta la colaboración entre los desarrolladores, ya que todos trabajan en una versión actualizada y funcional del código base, evitando conflictos y duplicidades.

Herramientas Recomendadas:

- Jenkins: Una herramienta de automatización de código abierto que facilita la integración continua y la entrega continua.
- GitLab CI/CD: Parte de la plataforma GitLab, esta herramienta de CI/CD está completamente integrada con los repositorios de GitLab, permitiendo un flujo de trabajo sin interrupciones desde la codificación hasta la entrega.

- CircleCI: Otra herramienta de CI que se integra bien con GitHub y Bitbucket, ofreciendo pipelines de CI/CD configurables y soportando múltiples lenguajes de programación.

Una preparación adecuada del entorno para el desarrollo e integración continua es fundamental para lograr un proceso de desarrollo de software eficiente, de alta calidad y con tiempos de entrega reducidos.