

Prueba - Implementación de un pipeline de integración continua

En la siguiente evaluación pondrás a prueba tus conocimientos sobre los fundamentos y prácticas de Integración Continua (CI) a partir de un caso práctico diseñado para simular un entorno real de desarrollo. Para facilitar tu desarrollo, descarga el material llamado “apoyo prueba - Implementación de un pipeline de integración continua”.

Descripción

"TechFlow" es una empresa dedicada al desarrollo de soluciones digitales para pequeñas y medianas empresas. Actualmente, el equipo de desarrollo está construyendo una API para la gestión de usuarios que se integrará en un sistema más amplio de administración de recursos empresariales (ERP).

Uno de los principales problemas que enfrentan es la falta de un proceso automatizado para garantizar la calidad del código y realizar despliegues rápidos y confiables. Hasta ahora, todas las pruebas se realizan manualmente y no hay un sistema estructurado para detectar errores antes de que el código llegue a producción.

Tu rol como desarrollador backend será implementar un pipeline de integración continua que permita:

- Asegurar que el código pase por pruebas unitarias antes de ser integrado.
- Generar retroalimentación inmediata para el equipo en caso de errores.
- Mantener una API funcional utilizando una base de datos simulada (db.json).

Requerimientos

1. Configuración inicial del repositorio (2 puntos)

- Inicializa un repositorio Git local y realiza un commit inicial con el archivo db.json y el código proporcionado.
- Crea un repositorio remoto en GitHub y configura el enlace remoto en tu máquina local.

2. Configuración básica de la API (2 puntos)

- Verifica que la API funciona correctamente utilizando los datos en db.json.

- Ejecuta la API con json-server o incluye código para servir el archivo como base de datos local.

3. Automatización básica con Jenkins (3 puntos)

- Configura un pipeline en Jenkins utilizando un Jenkinsfile que:
 - Clone el código desde el repositorio remoto.
 - Ejecute npm install para instalar las dependencias.
 - Inicie el servidor para validar que la API responde en el puerto configurado.

4. Ejecución de pruebas automatizadas (2 puntos)

- Modifica el pipeline para ejecutar las pruebas preescritas (npm test) automáticamente.
- Asegúrate de que el pipeline falle si alguna prueba no pasa.

5. Reporte y retroalimentación (1 punto)

- Configura un paso en el pipeline que genere un reporte con los resultados de las pruebas.
- Incluye un archivo REPORT.md explicando los pasos realizados y los problemas encontrados.

¡Mucho éxito!



Consideraciones y recomendaciones

- En la carpeta apoyo prueba, encontrarás:
 - **db.json**: Base de datos simulada.
 - **app.js**: API básica para gestionar usuarios.
 - **tests/app.test.js**: Pruebas unitarias preescritas.
 - **package.json**: Archivo para inicializar el proyecto con las dependencias necesarias.
- Puedes realizar modificaciones en el código proporcionado si lo consideras necesario, pero cualquier cambio debe ser documentado en el archivo REPORT.md.
- Se valorará la claridad en la configuración del pipeline y la capacidad de integrar correctamente todos los elementos.