

- ¿Qué aprendiste sobre el ciclo de vida de Maven?

Aprendí que el ciclo de vida de Maven está compuesto por fases secuenciales como `validate`, `compile`, `test`, `package`, `install` y `deploy`, las cuales automatizan todo el proceso de construcción del software. Esto permite compilar, probar y empaquetar una aplicación de forma consistente y ordenada, reduciendo errores humanos y ganando eficiencia.

- ¿Cómo facilita Maven el trabajo en equipo y la reproducibilidad?

Maven estandariza la estructura del proyecto y centraliza la configuración en el archivo `pom.xml`, lo cual permite que cualquier miembro del equipo pueda clonar el repositorio, ejecutar los mismos comandos y obtener los mismos resultados. Esto garantiza que todos trabajen con las mismas dependencias y configuraciones, lo que mejora la colaboración y la reproducibilidad del proyecto.

- ¿Cuál fue el mayor reto al trabajar con dependencias?

El mayor reto fue entender cómo funciona la resolución de versiones y los diferentes *scopes* (alcances) de las dependencias. También fue desafiante cuando se presentaban conflictos entre versiones o cuando una dependencia no se encontraba en el repositorio. Aprender a gestionar correctamente el `pom.xml` fue clave para superar estos problemas.

- ¿Por qué crees que Maven es tan usado en entornos empresariales?

Porque permite automatizar y estandarizar completamente el proceso de construcción del software, lo que reduce tiempo y errores. Además, facilita la integración con herramientas de CI/CD, el manejo de versiones, la incorporación de pruebas automatizadas y la gestión eficiente de dependencias. Todo esto lo convierte en una herramienta muy robusta y confiable para proyectos profesionales.

- ¿Qué harías diferente si tuvieras que automatizar otro proyecto?

Utilizaría perfiles desde el inicio para separar configuraciones de desarrollo y producción, y documentaría mejor el `pom.xml`. También incluiría herramientas de análisis estático de código y cobertura de pruebas, para mejorar la calidad del software. Además, integraría Maven desde el principio con GitHub Actions para tener una pipeline de CI/CD desde la primera versión del proyecto.