

Actividad 3: Selección de Herramientas y Configuración del Entorno de Desarrollo

La verdad, cuando empecé con este MVP, tenía claro que necesitaba herramientas que me facilitaran mucho el trabajo y me permitieran avanzar rapido sin romper nada. Para programar, me decanté por IntelliJ IDEA porque, además de ser super potente, me ofrece un entorno bastante amigable. Me encanta que se integre casi de forma mágica con otros sistemas, y que la depuración y el refactorizado sean mucho menos pesados de lo que estoy acostumbrado.

En cuanto al control de versiones, uso Git con GitHub. Con Git, llevo un registro super detallado de cada cambio, y GitHub me pone todo más organizado para revisar el código e incluso colaborar en equipo (aunque ahora esté yo solo, nunca se sabe si se unirá alguien mas). Así me aseguro de no romper nada y siempre puedo volver atrás si hago algún desastre.

Para la integración continua, me gusta Jenkins porque automatiza casi todo. Cada vez que meto un commit, se encarga de compilar, pasar pruebas y, si todo sale bien, dejar listo el despliegue. Me ahorra un monton de tiempo y dolores de cabeza, porque así no me encuentro con sorpresas de última hora.

Por otro lado, Docker es otro de mis grandes aliados. Me permite meter la aplicación en un contenedor, con todas sus dependencias bien definidas, y eso hace que se ejecute igual en mi máquina, en el entorno de pruebas o en el de producción. Adios a la típica frase de “a mí me funciona y a tí no”, algo que todos los que programamos hemos sufrido alguna vez.

En resumen, IntelliJ IDEA me da un salto cualitativo muy grande en productividad, GitHub mantiene todos mis cambios bajo control, Jenkins automatiza todo el proceso de pruebas y despliegue, y Docker me asegura la misma configuración en cualquier sitio. Con este combo, siento que el desarrollo del MVP va mucho más fluido y no me paso el día atorado con problemas de entorno o versiones.