

Onboarding-Log DevEx Spring PetClinic

Rol: Staff Software Engineer Senior

Objetivo: Evaluar el proceso de onboarding y la experiencia de desarrollo (DevEx).

1. Puntos Clave

- **Tiempo estimado de onboarding:** 20–30 minutos (incluyendo descarga inicial de dependencias).
- **Entorno utilizado:** Windows + PowerShell + Java 22.
- **Principal fricción:** README orientado a comandos Unix sin aclaración explícita para Windows.
- Estructura del proyecto compleja sin explicación arquitectónica inicial.
- Primer arranque (~10.6 segundos) con logs extensos sin señal clara de sistema listo.
- **Oportunidad de mejora:** documentación técnica, validación de entorno y soporte con Docker.

2. Entorno de Desarrollo

Sistema Operativo: Windows

Terminal: PowerShell

Versión de Java: Java 22

IDE utilizado: IntelliJ IDEA Community Edition

Gestor de dependencias: Maven Wrapper (mvnw)

Base de datos: H2 embebida (en memoria)

Herramientas adicionales: Git

3. Pasos Realizados

3.1 Exploración del repositorio

Se revisó el repositorio oficial en GitHub para comprender la estructura del proyecto.

Se identificó el archivo README.md como punto principal de referencia para la ejecución.

3.2 Clonado del repositorio

```
git clone https://github.com/spring-projects/spring-petclinic.git
```

```
cd spring-petclinic
```

3.3 Compilación del proyecto

En Windows PowerShell se ejecutó:

```
.\mvnw clean install
```

Durante este proceso se descargaron dependencias, se compiló el código y se ejecutaron los tests automáticamente.

3.4 Ejecución de la aplicación

```
.\mvnw spring-boot:run
```

Alternativamente puede ejecutarse desde IntelliJ ejecutando la clase PetClinicApplication.java.

3.5 Configuración de base de datos

El sistema utiliza una base de datos H2 en memoria por defecto, por lo que no fue necesario configurar una base de datos externa ni variables de entorno adicionales.

3.6 Acceso a la aplicación

Una vez iniciado el servidor, se accedió desde el navegador a <http://localhost:8080>, verificando su correcto funcionamiento.

3.7 Ejecución de pruebas

Los tests se ejecutaron automáticamente durante el proceso de build.

Las pruebas cubren principalmente lógica de dominio básica. No se observa una estrategia avanzada de pruebas de integración o cobertura arquitectónica profunda.

4. Problemas Encontrados

- Comandos del README no adaptados explícitamente para Windows.
- Ausencia de documentación arquitectónica clara.
- Logs extensos durante el arranque sin señal clara de aplicación lista.
- No existe validación automática de versión de Java ni del entorno.
- Estrategia de testing limitada en profundidad e integración.

5. Calificación DevEx

Calificación otorgada: 8 / 10

El proceso de onboarding es funcional y relativamente sencillo. Sin embargo, existen oportunidades claras de mejora en documentación, automatización del entorno y fortalecimiento de la estrategia de pruebas para alcanzar estándares modernos de ingeniería de software.

6. Sugerencias de Mejora

1. Agregar una sección de arquitectura en el README con explicación de capas y flujo del sistema.
2. Incluir ejemplos de comandos específicos para Windows.
3. Crear scripts de validación de entorno (check-env.sh / check-env.ps1).
4. Proveer Dockerfile y docker-compose oficiales para facilitar el onboarding corporativo.
5. Mejorar la estrategia de testing incorporando pruebas de integración más robustas.
6. Añadir un mensaje claro en consola que indique cuando la aplicación está completamente lista.

7. Conclusión

Spring PetClinic es un sistema monolítico estable y bien estructurado, adecuado para fines demostrativos y educativos. No obstante, desde la perspectiva de un Staff Engineer encargado de estabilizar y modernizar el sistema, se identifican áreas claras de mejora en documentación, automatización y estrategia de pruebas que permitirían optimizar significativamente la experiencia de desarrollo.