

# Justificaciones del Proyecto

## 1. Caffeine para caché clave-valor

- Latencia ultra-baja y simplicidad de despliegue en JVM sin red.
- Operacionalmente ligero, sin proceso externo.
- Control local de políticas: `expireAfterWrite`, `maximumSize`, etc.

## 2. MongoDB para lista de símbolos favoritos

- En un entorno de prueba de concepto (POC) o challenge, permite iterar rápidamente: no requiere esquema rígido y facilita almacenar arrays de símbolos sin migraciones.
- Configuración mínima para levantar la base y empezar a probar funcionalidades.

## 3. H2 local para datos de usuario y autenticación

- Ligero y embebido, ideal para desarrollo y pruebas.
- ACID e integridad referencial (transacciones, claves únicas).
- Configuración cero con Spring Boot y consola `/h2-console`.
- Portabilidad a producción cambiando solo la URL JDBC.

## 4. Modelo de capas

- Separación de responsabilidades: controladores, servicios, repositorios, clientes externos.
- Reutilización y testabilidad: capas testeables aisladamente.
- Flexibilidad: cambiar fuente de datos sin tocar lógica de negocio.
- Claridad en flujos y errores: cada capa expone su API interna.

## 5. Secrets expuestos en `application.properties`

- Facilidad de configuración en desarrollo: credenciales listas para clonar y correr.
- Consciente de inconvenientes:
  - \* Riesgo de filtración si se sube a VCS.
  - \* No cumple 12-factor apps, conviene `env-vars` o gestores de secretos.
  - \* Dificultad para distintos entornos sin copias o cambios manuales.
- Plan de mejora: migrar a variables de entorno o gestor de secretos (Vault, AWS SM).