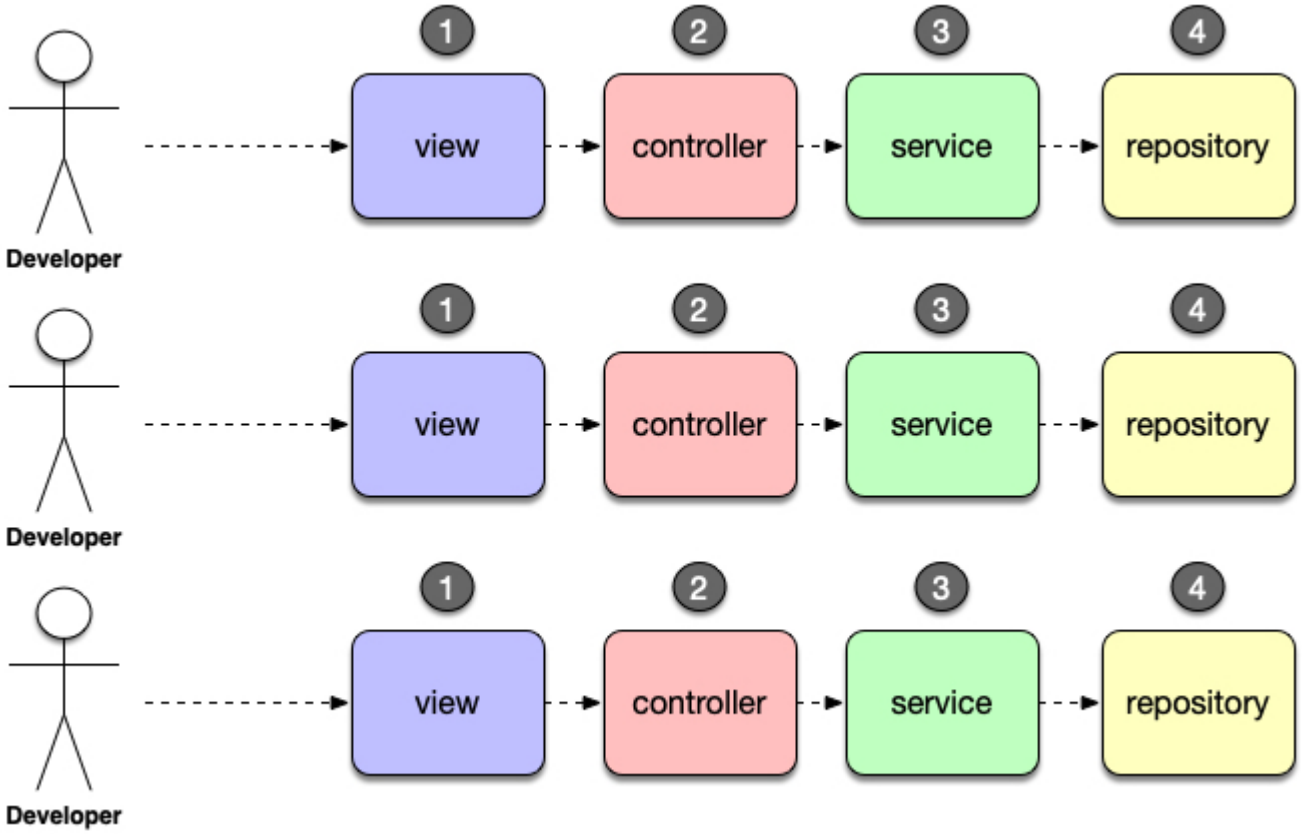
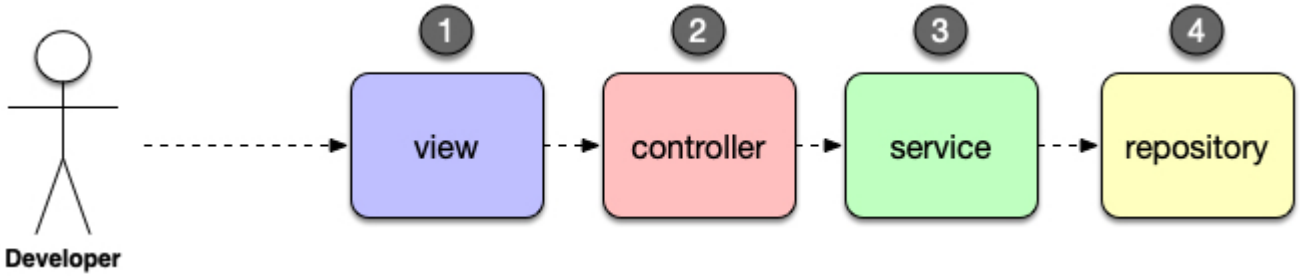
Los Frameworks nos ayudan mucho en el día a día de nuestros desarrollos . En unos casos usaremos Spring en otros Hibernate o Angular .Cada uno de ellos nos permite simplificar sobremanera la forma en la que trabajamos generando homogeneidad y buenas prácticas de forma prácticamente automática. Sin embargo aunque esto es muy positivo los frameworks tienen también aunque no lo parezca un lado oscuro . Eso no implica que no tengamos que utilizarlos , siempre son la opción de referencia pero tenemos que tener en cuenta que ese lado oscuro existe. ¿A qué me refiero con ello? . Los frameworks automatizan muchas cosas y convierten el desarrollo en una cadena de montaje.



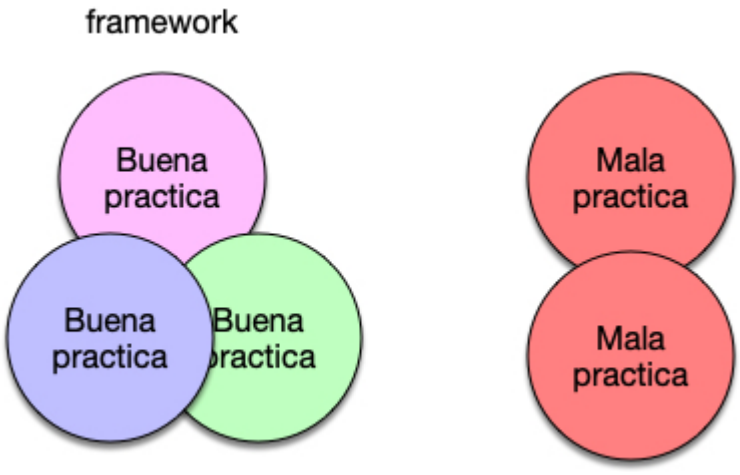
Frameworks y su lado oscuro

¿Qué implica esto ? Cuando construimos a nivel de desarrollo una cadena de montaje en muchas ocasiones podemos acabar convirtiendo a los desarrolladores en meros robots que realizan tareas repetitivas o muy cerca de ellas. Es decir oye , construyeme una nueva

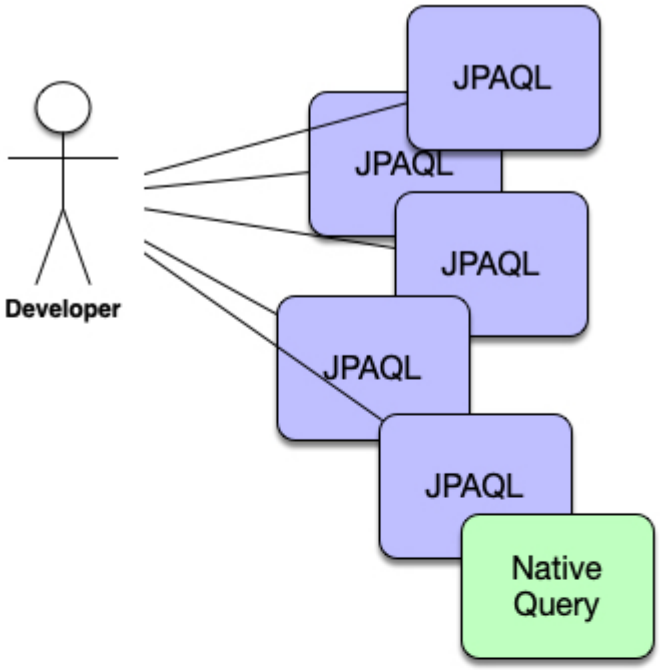
pantalla , un nuevo repositorio ,un nuevo servicio etc.

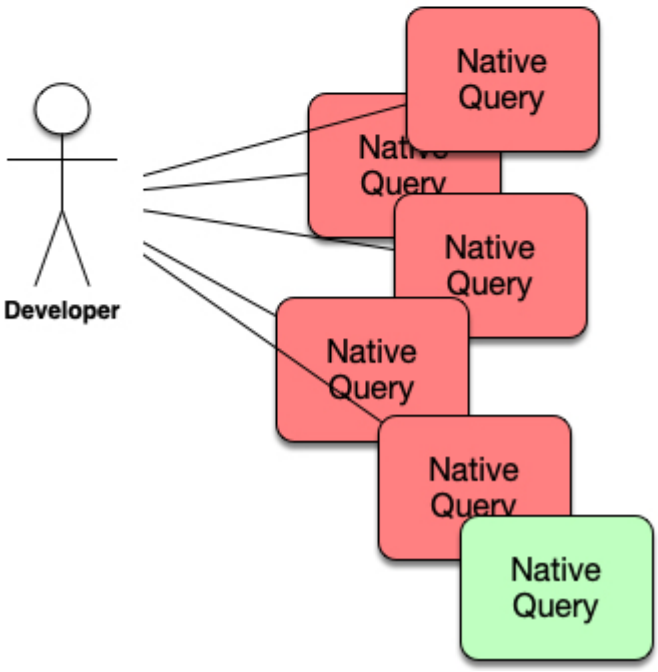


Eso en muchas ocasiones acaba en situaciones en las que nadie tiene muy claro porque se hacen unas cosas u otras . Simplemente se hacen porque así se hacían y el framework funciona de esta o esta otra manera. Esa es la realidad y abre las puertas a un problema que a veces es difícil de resolver y es el construir buenas aplicaciones con una buena arquitectura. Esto en principio puede resultar chocante ¿ Los frameworks no nos obligan a trabajar de una determinada forma y nos aportan buenas prácticas? .



La realidad es que esto es cierto , pero es cierto hasta cierto punto. Los frameworks “ayudan” pero lamentablemente no siempre “obligan” y podemos encontrarnos en situaciones que un desarrollador este utilizando los frameworks de una forma “incorrecta” . Por ejemplo en Hibernate existe la posibilidad de construir queries nativas para mejorar el rendimiento o abordar consultas que quizás JPA QL no soporte .





Es un buen añadido al framework y aporta. ¿ Puede o debe un desarrollador crear la aplicación únicamente utilizando queries nativas? . La realidad es que NO , lamentablemente yo me he encontrado con aplicaciones que estaban construidas así .

Parece imposible que se den estos cosas , pero suceden … suceden casuísticas de este estilo . Suceden porque los frameworks “automatizan el trabajo” y reducen las necesidades de que “pensemos” lo que estamos haciendo y si partimos de un ejemplo inicial incorrecto , podemos hacer la aplicación entera de forma incorrecta.

Frameworks y Expertise

Esto me hace volver a un tema clásico que es el concepto de experto. Hoy por hoy tener 3 o 6 años de experiencia utilizando un framework no te convierte necesariamente en un experto del area. ¿Por qué , pues porque puede ser que simplemente hayas hecho trabajo “robótico” y nada más. ¿Son por ejemplo buenas preguntas para contratar un experto en Spring framework las siguientes?:

- ¿Me puedes explicar qué es el patrón MVC?
- ¿Para qué sirve un Repositorio y con qué anotación se identifica?
- ¿Qué relación existe entre Repositorio y Servicio?
  Nos pueden parecer preguntas válidas pero solo nos ayudarán a identificar si la persona ha usado alguna vez el framework , aunque sea en modo robot. Sin embargo estas otras preguntas pueden ser mucho más adecuadas.

1. ¿Para qué sirve un PointCut en Spring?
2. ¿Cómo dividirías las funcionalidades de configuración del framework en ficheros?
3. ¿Para qué sirve un WebApplicationInitializer?
4. ¿Cómo darías del alta un contexto de Spring para manejar pruebas unitarias?
   Solo una persona que conozca realmente el framework y vaya más allá del concepto de linea de montaje contestará correctamente a varias de estas.

Conclusiones

Cuando utilicemos frameworks tenemos que estar seguros de que entendemos cuál es su funcionamiento real y tener la confianza suficiente para ver que las buenas prácticas que estamos aplicando son las correctas.

Otros artículos relacionados

1. El patrón de inyección de dependencia y su utilidad
2. OpenSessionInView AntiPattern y sus problemas
3. Framework vs Libreria dos conceptos importantes
4. Mis Libros
5. Frameworks Java