

**Justifica las decisiones tomadas a la hora de crear tu aplicación: ¿Qué tecnologías has empleado? ¿Por qué?**

En primer lugar, he utilizado JHipster. Se trata de un generador de Yeoman que usa Spring Boot y Angular para crear aplicaciones web haciendo el setup del proyecto y del entorno de forma rápida y sencilla. El código que genera JHipster es fácil de comprender y las actualizaciones de código son no intrusivas, útil para hacer modificaciones. Además, tiene un proceso de login implementado, hace uso de las tecnologías más recientes y la posibilidad de añadir plugins.

Las tecnologías que utiliza en el back-end son:

- **Spring Boot:** es una herramienta que nos permite crear un aplicación Spring de manera rápida y desplegarla rápidamente evitando tener que realizar manualmente ciertas configuraciones.
- **Maven o Gradle:** en el caso de nuestra aplicación he utilizado Maven. Este permite realizar el build, correr los test y ejecutar la aplicación tanto en desarrollo como en producción.
- **Spring Security:** es un framework que permite proporcionar autenticación y autorización a las aplicaciones.
- **Spring Data JPA:** tiene como objetivo mejorar significativamente la implementación de capas de acceso a datos al reducir el esfuerzo a la cantidad que realmente se necesita. Como desarrollador, escribe las interfaces de su repositorio, incluidos los métodos de búsqueda personalizados, y Spring proporciona la implementación automáticamente.
- **Liquibase:** es un control de versiones de la BBDD. Gracias a él podremos gestionar el estado de la BBDD, hacer modificaciones sobre esta ejecutando comandos de manera parecida a como lo haríamos con git. Para ejecutarlo, utilizaremos el plugin de maven para Liquibase.
- **Elasticsearch:** es una BBDD no relacional. Cada vez que guardamos información, va a crear una serie de índices para que luego podamos acceder a ella de una manera rápida. Es distribuible y escalable. Se puede acceder a través de una API Rest.

En cuanto al front-end, no he tenido que modificar ni configurar nada para este ejercicio pero las tecnologías que utiliza JHipster para crearlo son: Angular, Bootstrap, HTML5

Boilerplate, Gulp.js o Webpack, entre otros. Además, permite soporte para internacionalización aunque no lo he configurado en este proyecto.

Al configurar la opción de paginación, hemos hecho uso del componente de Bootstrap de paginación.

Para poder hacer uso de JHipster, he tenido que instalar Node Package Manager, sistema de gestión de paquetes que nos permite obtener cualquier librería con pocas líneas de código, lo que permite agregar dependencias de forma simple, distribuir paquetes y administrar tanto el módulo como el proyecto.

**Justifica las decisiones tomadas a la hora de crear tu entidad: ¿Por qué has elegido ese tipo de campo? ¿Has seleccionado validación? ¿Has seleccionado paginación? ¿Por qué?**

- **Título:** tipo de campo string con tres validaciones: longitud máxima y mínima y que sea requerido. He añadido que sea requerido para que no se añadan registros sin ningún tipo de información.
- **Descripción:** tipo de campo string con dos validaciones, longitud máxima y mínima. En este caso, la longitud máxima la he puesto más amplia ya que se presupone que al ser la descripción de una experiencia será más extensa.
- **Localización:** también la he configurado de tipo string y con una longitud máxima más amplia que el título para poder incluir la dirección entera.
- **Fecha:** es un campo de tipo Instant sin validación, ya que las opciones de validación que tenemos en este tipo de campo son requerido y único y al no tener el contexto del proyecto no lo he creído necesario.

En la mayoría de los campos es recomendable añadir validaciones para poder restringir lo que el usuario va a introducir en dicho campo. De esta manera reducimos las posibilidades de que el usuario introduzca información diferente a la que le solicitamos.

Por ejemplo, si el usuario pone un string demasiado largo y yo no he gestionado ese error, el servidor puede devolvernos un 500.

Por último, sí he seleccionado paginación ya que se presupone que experiencia es una entidad que puede tener muchas instancias. Si no aplicamos paginación y acabamos teniendo muchas instancias, tendremos problemas con el servidor.