

Práctica 1

1. Crear un proyecto Spring Boot, siguiendo los pasos del [repositorio público de prácticas](#)
2. Asegurarse de configurar la [conexión a la BD H2](#)
 - a. [Ésta configuración](#) puede servir de referencia
3. Vamos a crear un servicio web muy sencillo que simplemente va a realizar algunas operaciones CRUD sobre una entidad **Persona**
4. Crear la entidad del modelo de datos, **Persona** con los siguientes atributos:
 - a. ID
 - b. Apellidos y nombres
 - c. Fecha de nacimiento
 - d. Crear getters/setters para la entidad. Pueden usar [Lombok](#)
5. Crear una clase **DTO** para la entidad Persona, que contenga los siguientes atributos:
 - a. ID
 - b. Nombre completo: surge de concatenar el apellido y nombre
 - c. Edad: calculada a partir de la fecha de nacimiento
6. Definir un repositorio de JPA para las personas, Se debe crear una interfaz que extienda de alguna de las que provee el framework.
 - a. Por ejemplo, **JpaRepository<T, ID>**
 - b. Hay varios tipos de repositorios más en la jerarquía, por ejemplo **PagingAndSortingRepository**, cómo su nombre indica, permite implementar queries a la BD considerando paginación y ordenamiento
7. Para la capa de servicios de la aplicación, vamos a definir dos:
 - a. Uno va a ser el encargado de transformar o *mapear* una entidad de la base de datos, en el DTO que definimos anteriormente
 - b. Otro servicio va a tener las operaciones que va a exportar el servicio web, que van a ser:
 - i. Dar de alta una nueva Persona en el sistema
 - ii. Obtener el listado de todas las Personas
8. Crear las clases que implementen estas dos interfaces. Anotarlas con **@Service** para registrarlas en Spring
 - a. Para el servicio transformador, se puede utilizar la librería [MapStruct](#). También es válido hacerlo *a mano*
 - b. Recordar usar **@Autowired** en estos servicios para inyectar las dependencias necesarias. Por ejemplo, el servicio que va a implementar las operaciones CRUD, va a necesitar inyectar el repositorio definido en el punto 5
9. Crear un Controller en donde se definan los endpoints para las dos operaciones que exportará el servicio. Considerar criterios REST (rutas, verbos HTTP, etc)
10. Crear un inicializador que cargue algunos datos de prueba. Esto se logra definiendo un servicio que implemente la interfaz **ApplicationRunner**. Allí, se inyecta el repositorio y se lo usa para dar de alta Personas en el sistema

Docker

En el repositorio del inciso 1, hay un [Dockerfile](#) que contiene las definiciones para *dockerizar* el proyecto. El Dockerfile [se compone de dos etapas](#), una para construcción del proyecto (compilación y generación de un **jar**), y otra para ejecución

En el README del repositorio se encuentran las instrucciones de uso del contenedor Docker. Deben copiar el Dockerfile a su proyecto, al directorio raíz, para poder utilizarlo.

Gitlab

Vamos a usar Gitlab para subir el código fuente, y también la imagen de Docker a la [Registry que nos provee](#).

1. Registrarse en [Gitlab](#)
2. Crear un proyecto y agregar a los integrantes del grupo
 - a. Para invitar: **Project information > Members**, botón "Invite members"
3. En la sección **Packages and registries > Container Registry** se accede al registro de imágenes Docker del proyecto. El mismo contiene las instrucciones para logearse y subir una imagen de Docker

Entregables

1. Proyecto java completo en Gitlab: código fuente e imagen Docker
2. Proyecto postman para ejecutar los escenarios

Referencias:

- <https://projects.spring.io/spring-boot/>
- <https://www.baeldung.com/spring-boot-h2-database>
- <https://docs.spring.io/spring-boot/docs/2.7.9/reference/htmlsingle/#web>
- <https://docs.spring.io/spring-framework/docs/5.3.25/reference/html/web.html#mvc-controller>
- <https://spring.io/guides/tutorials/rest/>
- <https://www.baeldung.com/jpa-entities>
- <https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa>
- <https://www.baeldung.com/spring-data-repositories>
- <https://www.baeldung.com/spring-component-repository-service#service>
- <https://www.baeldung.com/spring-autowire>
- <https://mapstruct.org/documentation/stable/reference/html/#basic-mappings>
- <https://www.baeldung.com/jackson-annotations>
- <https://www.baeldung.com/running-setup-logic-on-startup-in-spring#7-spring-boot-applicationrunner>

