Arquitectura Multicapas

Arquitectura Multicapas

- La **Arquitectura Multicapa** plantea que un proyecto desarrollado en Spring tiene una determinada **división en capas** según la función o tarea que cada una de ellas cumple.
- Generalmente esta división consta de 5 capas pero puede variar según el proyecto:

1. **Repository (conocida también como DAO)**
2. **Entity (conocida también como Model)**
3. **DTO**
4. **Service**
5. **Controller**

- **Repository (o DAO):** Es la capa de persistencia de datos. Facilitan el trabajo con diferentes tecnologías de acceso a datos como JDBC, Hibernate o JPA de manera consistente. Cada clase de repositorio debe estar marcada con la anotación @Repository.
- **Entity (o Model):** Una entidad representa una tabla almacenada en una base de datos. Cada instancia de una entidad representa una fila en la tabla. Cada clase de entidad debe estar marcada con la anotación @Entity.
- **DTO (Data Transfer Object):** Los objetos de transferencia de datos se utilizan para desacoplar la representación de datos (la vista) de los objetos del modelo.

- **Service:** En pocas palabras se encarga de contener la lógica de negocio de la aplicación. Cada clase de servicio debe marcarse con la anotación @Service.
- **Controller:** Encargada de atender una solicitud desde el momento en que es interceptada hasta la generación de la respuesta y su transmisión. Cada clase de controlador debe estar marcada con la anotación @RestController.

Estructura de un Proyecto:

- Cuando se crea un proyecto, la estructura del mismo debe representar la arquitectura multicapa que haya sido elegida.
- En particular, cada capa debe incluirse dentro de un paquete específico y cada uno de ellos, debe tener el mismo nombre que la propia capa.

Annotations

Estructura de un Proyecto:

- En la mayoría de las aplicaciones existen diferentes capas. En cada capa pueden existir varios beans. Para detectar estos beans automáticamente, Spring usa anotaciones de escaneo de classpath y luego registra cada bean en el ApplicationContext.
- Algunas de estas anotaciones son:
    - **@Component** es un estereotipo genérico para cualquier componente administrado por Spring.
    - **@Service** anota clases en la capa de servicio.
    - **@Repository** anota clases en la capa de persistencia.
    - **@Controller o @RestController** anota al controlador y sus diferentes end-points

@Repository:

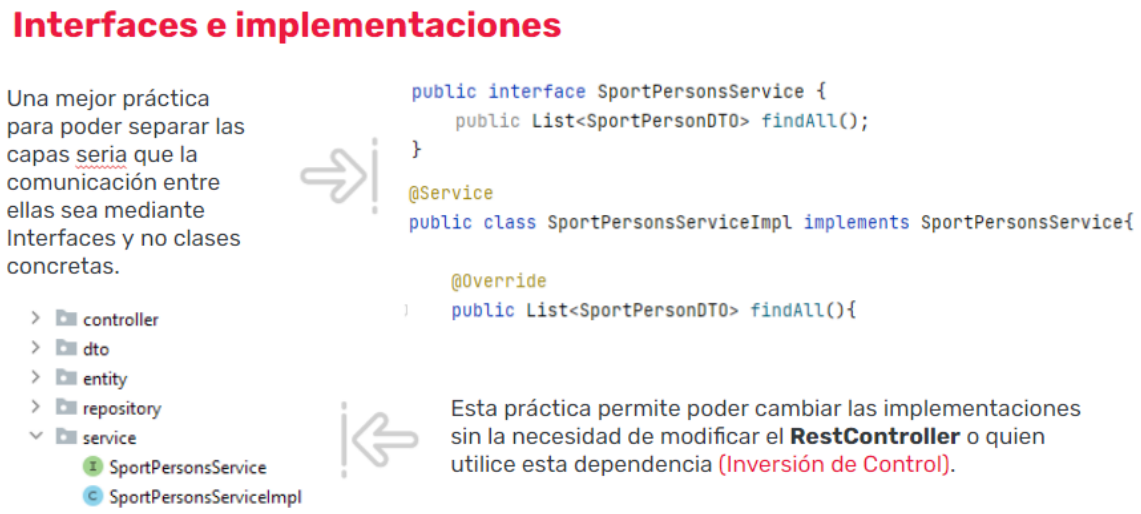
- Es una anotación de Spring que indica que la clase anotada es un repositorio.

- Un repositorio es un mecanismo para encapsular el comportamiento de almacenamiento, recuperación y búsqueda que emula una colección de objetos.
- Es una especialización de la anotación **@Component** que permite que la implementación de clases se detecten automáticamente a través del escaneo de classpath.

Una ventaja de usar esta anotación es que tiene habilitada la traducción automática de excepciones de persistencia. Cuando se usa un marco de persistencia como Hibernate, las excepciones nativas lanzadas dentro de las clases anotadas con ***@Repository*** *se traducirán automáticamente en subclases de DataAccessException de Spring.*

@Service:

- La lógica de negocio de una aplicación generalmente reside dentro de la capa de servicio, en una capa diferente, separada de la clase **@RestController.**
- Usaremos la anotación **@Service** para indicar que una clase pertenece a esa capa.



Resumen de Annotations