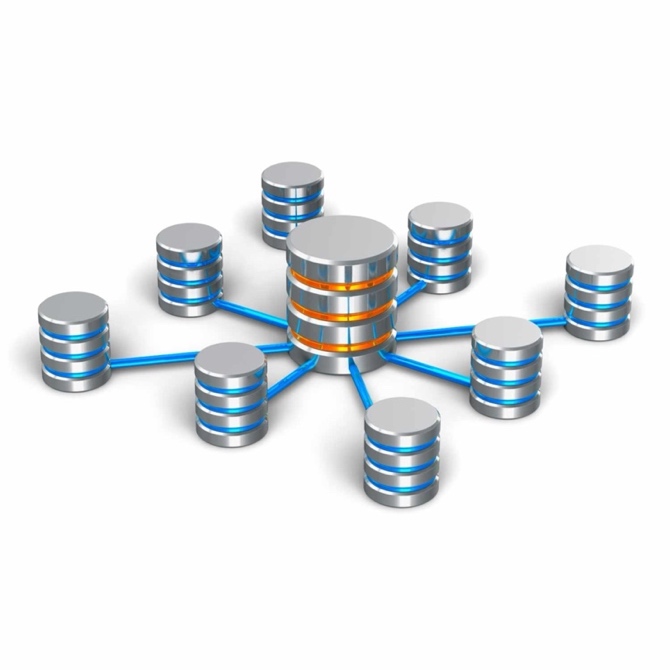
**ACCESO A DATOS**

**Spring:**

**@Component, @Repository y @Controller**

2º DESARROLLO DE APLICACIONES MULTIPLATAFORMA

****

****

*Por Agustín Rodríguez Márquez IES Rafael Alberti*

**¿Qué diferencias existen entre las anotaciones @Component, @Repository, y @Controller? ¿Cómo se aplican en el código del proyecto?**

Las anotaciones @Component, @Repository y @Controller no dejan de ser sencillas anotaciones de Spring que se utilizan para indicar el papel que una clase, como en mi caso son User y Post, en la estructura de una aplicación.

@Component: Es la anotación principal que indica que una clase es un componente o Bean de Spring. Se puede usar en cualquier clase que Spring vaya a manejar.

@Repository: Es una especialización de @Component que se utiliza en las clases que acceden a la base de datos. Proporciona traducción de excepciones, convirtiendo las excepciones específicas de la base de datos en excepciones de Spring sin comprobaciones. En mi código, esta anotación las he usado para las interfaces UserRepository y PostRepository.

@Controller: Controller no deja de ser otra especialización de @Component que se utiliza en las clases que manejan las solicitudes HTTP. Estas clases definen los puntos finales de la API. En mi código, uso esta anotación en el MainController.

@Service: Esta anotación es también una especialización de la anotación @Component. Se utiliza en las clases que implementan la lógica de negocio o las operaciones de servicio. Es una buena idea usar @Service en las clases de la capa de servicio porque especifica mejor la intención para las que la voy a usar. Yo uso @Service en la clase UserService. Esta clase se encarga de gestionar las operaciones de negocio más importantes que tengo, e intento aglutinar llamadas a varios repositorios de forma simultánea (las relaciones @OneToMany entre User y Posts). De esta forma, realizo operaciones de base de datos relacionadas con los usuarios y sus posts.

Aunque técnicamente se pueden usar las 3 anotaciones de manera intercambiable, cada anotación tiene un propósito específico y en caso de usar la anotación incorrecta, la respuesta a la API se puede romper, como ha ocurrido en alguna ocasión mientras realizaba el código.

**Bibliografía:**

Stack overflow (2018) - [*¿Diferencia entre anotaciones @Component, @Repository y @Service en Spring?*](https://es.stackoverflow.com/questions/174661/diferencia-entre-anotaciones-component-repository-y-service-en-spring)– Última consulta 23/01/2024.

Un poco de Java (2012) por Luismi Gracia – [*Anotaciones @Component, @Repository, @Service y @Controller en Spring*](https://unpocodejava.com/2012/09/19/anotaciones-component-repository-service-y-controller-en-spring/)– Última consulta 23/01/2024.