

Recordatorios básicos Clases Servicios

@Service: Informo que ese componente, es un servicio. Hay que recordar que en esta clase debe iniciar y finalizar las “lógicas de negocio” . Es la clase que debe comunicarse con el Repositorio **SIEMPRE**

@Transactional: informo que ese método es una transacción.

@Autowired : Esta anotación nos permite crear una instancia para ser utilizada de un servicio / repositorio. En caso de que exista una en otra clase, la reutiliza.

Utilidad para importar:

`import org.springframework.transaction.annotation.Transactional;`

Spring soporta las siguientes propagaciones:

- **PROPAGATION_REQUIRED** – Es la que viene por defecto, así que no es necesaria especificarla. Si existe transacción la aprovecha y sino la crea
- **REQUIRES_NEW** – Abre una transacción nueva y pone en suspenso la anterior. Una vez el método marcado como REQUIRES_NEW termina se vuelve a la transacción anterior.
- **PROPAGATION_SUPPORTS** – Si existe transacción la aprovecha, sino no crea ninguna.
- **PROPAGATION_MANDATORY** – Si no existe una transacción abierta se lanza una excepción.
- **PROPAGATION_NEVER** – Si existe una transacción abierta se lanza una excepción.
- **PROPAGATION_NOT_SUPPORTED** – Si existe una transacción la pone en suspenso, la transacción se reactiva al salir del método.

Atributo para utilizar si solo son operaciones de consulta:

`@Transactional(readOnly = true)` → Recordar que optimiza el funcionamiento de mi aplicación.