La clase de servicio abstracta AbstractService es una clase que crea un repositorio de un tipo genérico R, el cual Java infiere para crear los métodos de los servicios de manera genérica y será un repositorio del tipo genérico E. Esta clase extiende a otra clase de repositorio abstracta, AbstractRepository, la cual es un puente entre nuestra clase de servicio abstracta y JpaRepository.

La cabecera del servicio queda tal que así:
public abstract class AbstractService<R extends AbstractRepository<E>, E extends DomainEntity>

AbstractService tiene 4 atributos protegidos: *repository*, de tipo R, *validator*, de tipo Validator, *repositoryClass*, de tipo Class<R> y *domainClass*, de tipo E.

El primero de estos atributos se usa como repositorio de la clase R, la cual es un repositorio de la clase del dominio E. Sirve para instanciar los métodos save(), delete(), findOne() y findAll() de manera genérica.

El segundo se utiliza para validar los campos en los métodos reconstruct() de las clases que extiendan a esta clase.

El tercero y el cuarto se usan dentro de un método auxiliar instanceClass(), el cual recibe un parámetro de tipo clase del tipo genérico C (Class<C>) y asigna unos valores genéricos a los atributos de las clases de las entidades de dominio utilizando la reflexión del lenguaje Java. De esta manera conseguimos instanciar el método create() de manera genérica salvo pequeñas excepciones en valores concretos deseados en ciertos campos tal y como puedan ser valores de tipos enumerados o de entidades del dominio, por ejemplo, en cuyo caso bastaría con llamar al método create() genérico de la clase padre en el método create() de la clase hija y asignar sólo esos valores.

La clase AbstractRepository está vacía. Su única utilidad es servir de puente entre AbstractRepository y JpaRepository como ya se explicó antes, aunque, ya que hemos tenido que hacer uso de ella, hemos instanciado el tipo Integer del índice en JpaRepository de manera automática, lo cual previene posibles descuidos.

Su cabecera es la siguiente:
public interface AbstractRepository<T extends DomainEntity> extends JpaRepository<T, Integer>

Estas 2 herramientas son muy poderosas y de gran utilidad, pues reduce en gran medida el tiempo destinado a la creación los métodos de una clase de servicio por cada clase del dominio, unos métodos que son exactamente iguales excepto en el tipo del repositorio que usan y en el tipo que devuelven. Además, previenen de errores como puede ser el olvidarse del atributo *validator*.