RECORDATORIO ANOTACIONES SPRING

ANOTACION	USO	ADICIONAL
@Entity	Declara una clase como entidad (Futura tabla)	@Table(name=”loqquiera” {Si no lo uso, por defecto la tabla se va a llamar como la clase Entity} RECORDAR QUE ES DE JPA
@Id	Declara al atributo como Id	Toda clase del tipo Entity debe poseer un id.
@GeneratedValue	Declara que el atributo va a ser “autogenerado”	
@GenericGenerator	Para declarar estrategia de generar el Id	
@Enumerated	Declara que el atributo pertenece a una clase del tipo Enum.	Entre () debo escribir el criterio con el que va a usar el dato (EnumType.STRING) (EnumType.ORDINAL)
@ManyToOne **@OneToOne** **@ManyToMany** **@OneToMany**	Declara que hay relaciones con el atributo. El atributo es del tipo “entidad”(otra clase) Prestar atención a la cardinalidad de la relación.	En conveniente poner @JoinColumn(name =”id”) Luego de anunciar la relación, para evitar “tablas intermedias”
@Temporal	Todo dato que sea del tipo DATE debe utilizar esta anotación	Entre () debo escribir el criterio con el que va a usar el dato (TemporalType.TIME) (TemporalType.DATE) (TemporalType.TIMESTAMP)
@Autowired	El Sistema se ocupa de inicializar automáticamente el atributo.	
@Service	Declara una clase como servicio. Depende de un atributo de la clase reposotorio con las que se relaciona.	Recordar que los atributos del tipo RepositoryClass debo utilizar la anotación **@Autowired** para que los inicialice.
@Async	Le avisa que realice la ejecución de un método en paralelo.	(Ejemplo métodos para enviar notificaciones)
@Repository	Declara una clase como repositorio. Es una interface,no una clase	Interfax de la que depende: extends JpaRepository<Clase,TipodatoID>

Profe: Adri Bestilleiro

@Query	Utilizado antes de un método que realizara una consulta en una clase Repository ,incluye la consulta en la base de datos	@Query(" La consulta con formato MYSQL") La consulta debe estar entre ""
@Param	Utilizado en los métodos declarados previamente como QUERY	@Param(" nombrecolumna") Tipo de dato + nombre variable.
@Transactional	Antes de los métodos. Si el método se ejecuta sin largar excepciones, entonces se hace un comit a la base de datos	Si salta una excepción, el método vuelve atrás con la transacción y no guarda cambios en la base.
@Controller	Declara una clase como controlador. Permite el manejo de entradas y salidas a las diferentes vistas	Es lo que permite relacionar entrada y salida de datos HTML con JAVA.
@RequestMapping	Se utiliza tanto a nivel de clase como de método. Se utiliza para manejar "solicitudes"	Uso: @Controller @RequestMapping(value= "/xx") (entre comillas, el html que requiere) Debe estar en aquellos controladores que no son paginas de inicio, para indicar ruta de acceso
@RequestParam	Se utiliza para determinar la necesidad de recibir o no un parámetro de una vista	@RequestParam(required = false o true)
@Lob @Basic	Cuando utilizo carga de "archivos"	(fetch=FetchType.LAZY) Debe estar el parámetro de como hacer la carga del contenido
@Inheritance	Anotación a utilizar en clase padre abstracta si implemento herencia	@Entity @Inheritance(strategy = InheritanceType.TABLE_PER_CLASS) Debo dejar marcada la estrategia de comportamiento

- Las anotaciones se declaran antes del atributo, método o clase a la que va a afectar.
- Recordar investigar sobre los atributos específicos de cada anotación.