

* Dependency Injection :

```
public class PersonaRest {
```

@Autowired

Anotación

```
private PersonaService personaService; } Propiedad
```

→ Hacemos eso para evitar ejecutar el servicio, ejemplo:

```
public class PersonaRest {
```

```
private final PersonaService personaService;
```

solo para constructor

```
public class PersonaRest {
```

```
personaService = new PersonaService();
```

```
}
```

→ forma sucia de hacerlo (con pocos usuarios sirve pero con muchos no, porque se tiene que instanciar muchas veces)

→ Con el @Autowired se instancia 1 sola vez

* Anotaciones: empiezan con @, indican alguna funcionalidad a lo que viene (clase, funciones, etc)

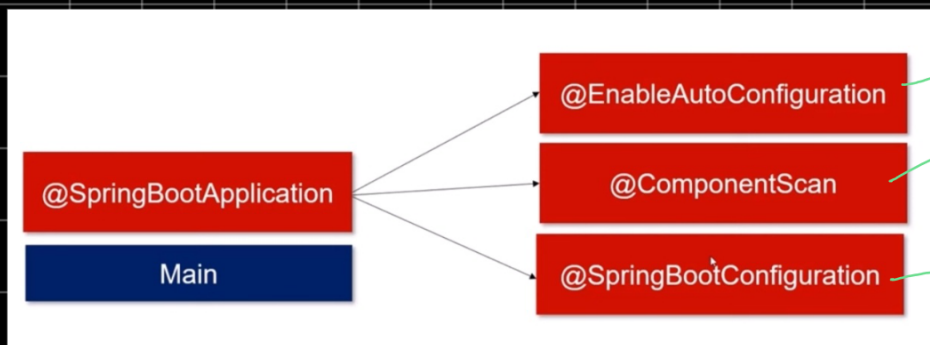
* en carpetas/documentos publicos
static (imagenes, pgs. web, etc)
index.html

@SpringBootApplication

```
public class CursoApplication {
```

```
public static void main(String[] args) { SpringApplication.run(CursoApplication.class, args); }
```

(ruta)
(proyecto)

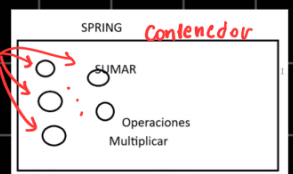


Se encarga de buscar en el classpath todas las clases con @Entity

Revisa los paquetes actuales que tengan las anotaciones de estereotipo que serían las demás capas (servicio, Repositorio, Controller)

Anotación que define el fichero de configuración de Spring
define que la clase anotada es una fuente de definiciones de beans

beans



base de datos





entidad: Alumno

MVC (model View Controller)

* Anotaciones :

@Entity: define entidad

@Controller:

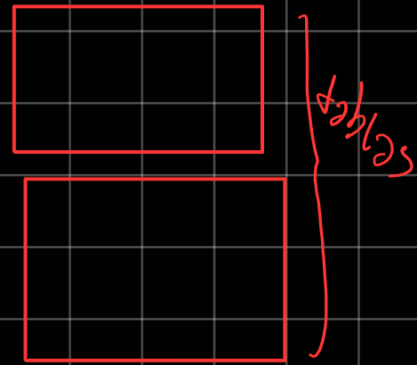
@Service:

@Repository:



cliente

Mostrar listado alumnos



<