

AUTOCONFIGURACIÓN Y USO DE @SPRINGBOOTAPPLICATION



CONFIGURACIÓN DE UNA APLICACIÓN **SPRING**

- ▶ Múltiples elementos de configuración.
- ▶ Clases anotadas con @Configuration.
- ▶ Uso de properties.

AUTOCONFIGURACIÓN CON SPRING BOOT

- ▶ Se obtiene gracias a `@SpringBootApplication` (o `@EnableAutoconfiguration`).
- ▶ Configura nuestra aplicación en función de las dependencias jar que encuentre en el *classpath*.
 - ▷ Si añadimos la dependencia de H2 sin configuración, autoconfigura una base de datos en memoria.

CÓMO VER LA CONFIGURACIÓN POR DEFECTO DE UN PROYECTO

- ▶ `mvn --debug spring-boot:run`

CÓMO DESHABILITAR UN ELEMENTO ESPECÍFICO DE LA AUTOCONFIGURACIÓN

- ▶ Si nos interesa que no se aplique un elemento específico de autoconfiguración, lo podemos deshabilitar con `@EnableAutoConfiguration(exclude=...)`

```
import org.springframework.boot.autoconfigure.*;
import org.springframework.boot.autoconfigure.jdbc.*;
import org.springframework.context.annotation.*;

@Configuration
@EnableAutoConfiguration(exclude={DataSourceAutoConfiguration.class})
public class MyConfiguration {
}
```

CÓMO DESHABILITAR UN ELEMENTO ESPECÍFICO DE LA AUTOCONFIGURACIÓN

- ▶ También lo podemos hacer mediante la propiedad *spring.autoconfigure.exclude*, que acepta una lista de clases (preferiblemente, nombres cualificados: *paquete.nombre*).