

MICROSERVICIOS Y SPRING BOOT



Spring Boot

- Simplifica el proceso de configuración de aplicaciones y gestión de dependencias
- Permite integrar entorno de ejecución en la aplicación
- Aplicaciones Java estándar (.jar)
- Ideal para creación de microservicios



Dependencias Maven


➤ La inclusión de dependencias en una aplicación Spring boot se simplifica mediante los starters

➤ Un starter incluye un conjunto de dependencias básicas maven para desarrollar un tipo de aplicación

➤ Ejemplo:

Incluye Spring core, context, web, web-mvc y jackson

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

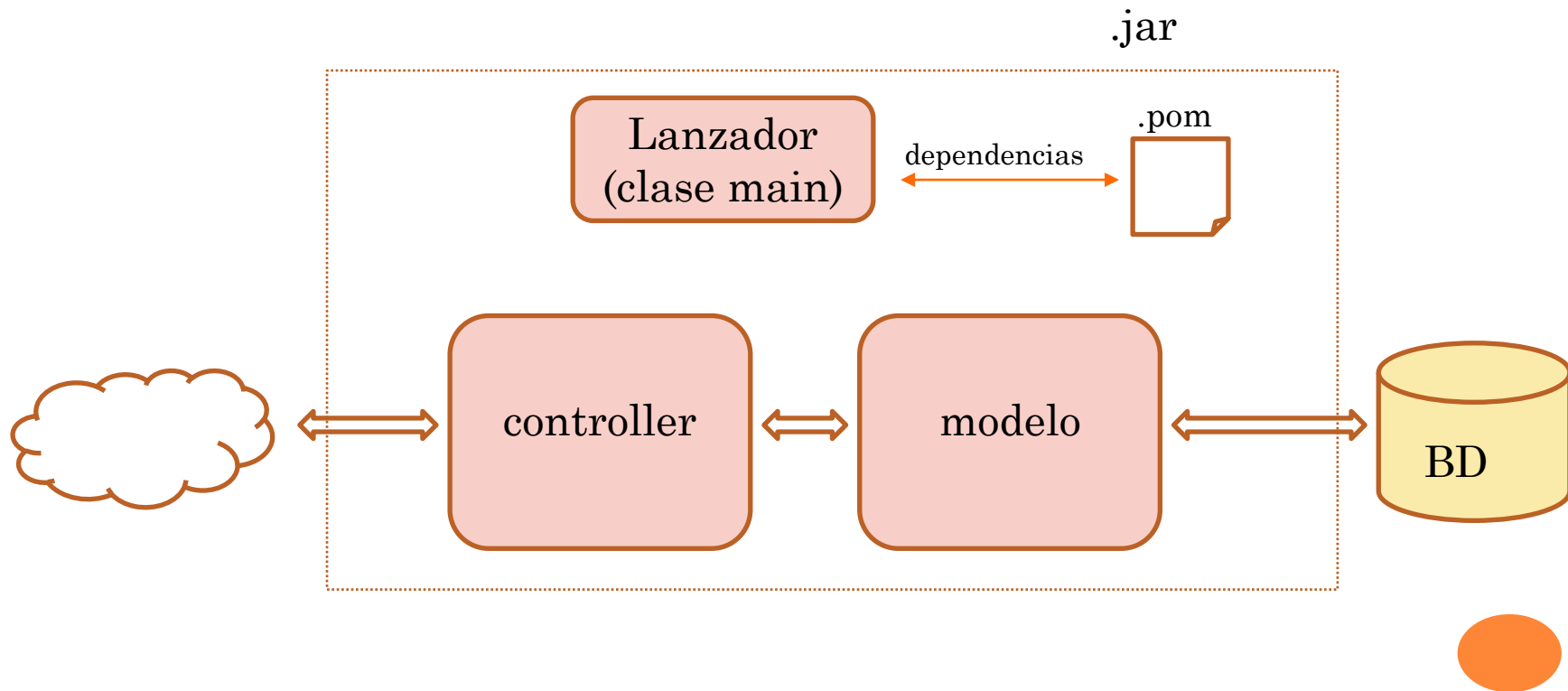


Configuración de aplicaciones

- Se eliminan los archivos de configuración .xml
- Se asumen una serie de configuraciones por defecto
- Para indicar parámetros de configuración específicos:
 - application.properties
 - application.yml



Estructura microservicio boot



La classe main

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

➤ @SpringBootApplication:

- @Configuration
- @EnableAutoConfiguration
- @ComponentScan



Controlador Rest

➤ **Clase POJO que define los métodos para atender las peticiones al servicio**

url asociada al
recurso

Tipo de devolución
respuesta

```
@RestController
public class Controller {
    @GetMapping(value="saludo", produces="MediaType.TEXT_PLAIN_VALUE")
    public String metodo(){
        return "Bienvenido a mi servicio";
    }
}
```

