

FORMAS DE CONFIGURACIÓN

- ▶ XML
- ▶ JavaConfig
- ▶ Anotaciones

XML es la forma que hemos venido usando hasta ahora.

JavaConfig

- ▶ La usaremos más con SpringBoot
- ▶ Nos permite realiza la configuración programáticamente

```
@Configuration
public class AppConfig {

    @Bean
    public MyService myService() {
        return new MyServiceImpl();
    }

}
```



Idéntico a definir <bean...> en XML

ANOTACIONES

- ▶ Será la que más usemos de ahora en adelante (junto con XML).

```
@Component  
public class Mundo {  
  
    @Value("Hola Mundo!")  
    private String saludo;  
  
    public Mundo() { }  
  
    //...  
}
```

Idéntico a definir <bean...> en XML



ESCANEO DE ANOTACIONES

- Para que Spring navegue por las clases y escanee las anotaciones

Namespaces

Configure Namespaces

Select XSD namespaces to use in the configuration file

☐ aop - <http://www.springframework.org/schema/aop>
☒ beans - <http://www.springframework.org/schema/beans>
☐ c - <http://www.springframework.org/schema/c>
☐ cache - <http://www.springframework.org/schema/cache>
☒ context - <http://www.springframework.org/schema/context>
☐ jee - <http://www.springframework.org/schema/jee>
☐ lang - <http://www.springframework.org/schema/lang>
☐ p - <http://www.springframework.org/schema/p>
☐ task - <http://www.springframework.org/schema/task>
☐ util - <http://www.springframework.org/schema/util>

```

<context:annotation-config />

<context:component-scan base-package="com.openwebinars.anotaciones.beans"></context:component-scan>

```

Source Namespaces Overview beans context Beans Graph

ANOTACIONES

- ▶ @Component y @Value

Definir un bean y darle valor a una propiedad

- ▶ @Required

Indicamos que hay que satisfacer una dependencia

- ▶ @Autowired y @Qualifier

Inyección automática de beans

ANOTACIONES vs XML

