CONFIGURACIÓN USANDO JAVA

JAVACONFIG

- Spring soporta la configuración vía código Java.
- Nos permite prescindir por completo de XML.
- Podemos combinar el uso de JavaConfig con las anotaciones trabajadas en el bloque anterior.

ANOTACIONES CLAVE

- @Configuration
  - A nivel de clase
  - Indica que una clase va a definir uno o más @Bean
- @Bean.
  - A nivel de método
  - Equivalente a <bean … />

JAVACONFIG BÁSICO

```
@Configuration
public class AppConfig {

    @Bean
    public Saludator saludator() {
        return new Saludator();
    }

}
```

INSTANCIACIÓN DEL CONTENEDOR

- Ahora usamos `AnnotationConfigApplicationContext`
- Recibe como argumento la/s clase/s que tienen alguna configuración.

```
public static void main(String[] args) {
   ApplicationContext appContext = new
            AnnotationConfigApplicationContext(AppConfig.class);
   Saludator saludator = appContext.getBean(Saludator.class);
   //…
}
```

INSTANCIACIÓN DEL CONTENEDOR

- Podemos usar el constructor vacío y registrar las clases.

```
public static void main(String[] args) {
    AnnotationConfigApplicationContext ctx =
                        new AnnotationConfigApplicationContext();
    ctx.register(AppConfig.class, OtherConfig.class);
    ctx.register(AdditionalConfig.class);
    ctx.refresh();
    MyService myService = ctx.getBean(MyService.class);
    myService.doStuff();
}
```

ESCANEO DE COMPONENTES

- Idéntico comportamiento que en XML
- @ComponentScan(basePackages=...)
- También programáticamente

```
public static void main(String[] args) {
    AnnotationConfigApplicationContext ctx =
                        new AnnotationConfigApplicationContext();
    ctx.scan("com.acme");
    ctx.refresh();
    MyService myService = ctx.getBean(MyService.class);
}
```