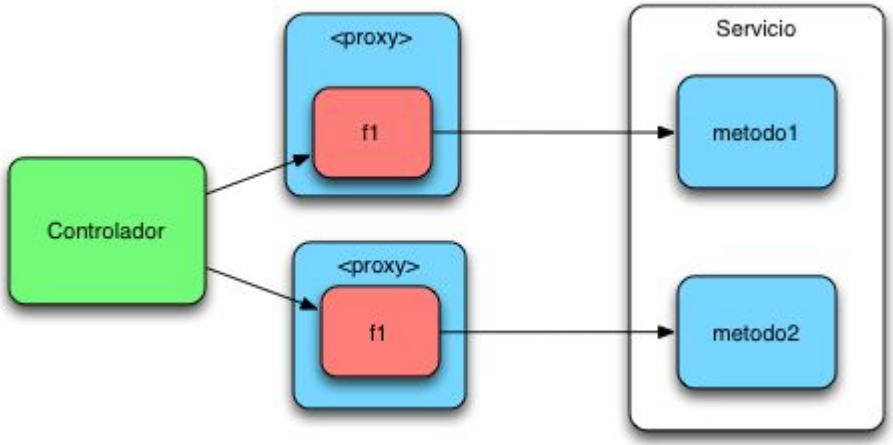
Programación orientada a Aspectos

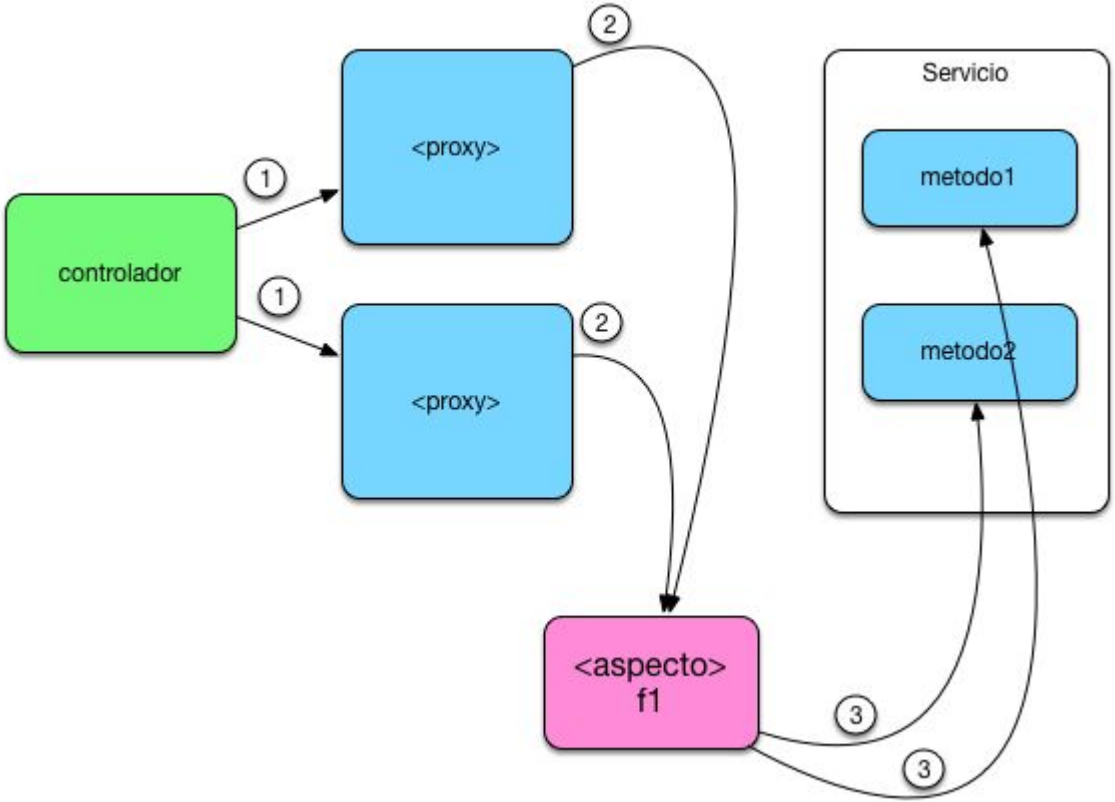
¿Qué es?

La programación orientada a aspectos se basa en añadir funcionalidad adicional a los métodos ya construidos sin tener que modificar el código de cada método.

Por ejemplo si tenemos el siguiente código

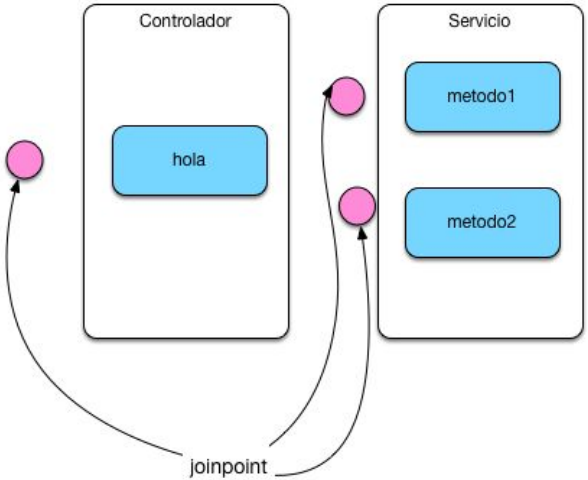


Podemos agregarle un aspecto



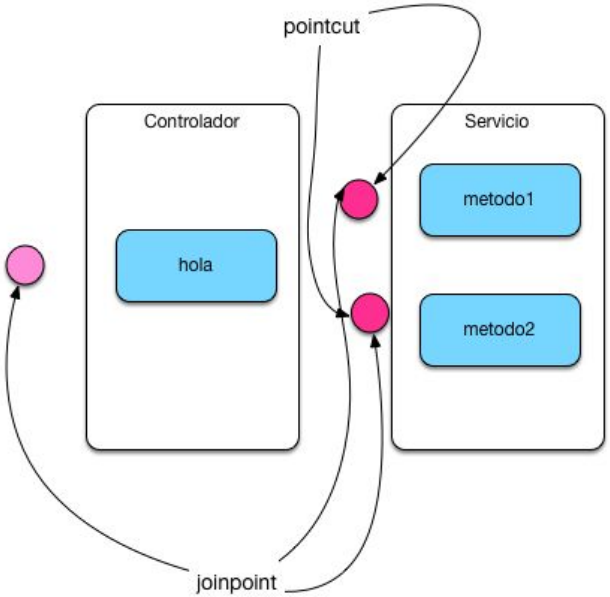
Join Point

Define los posibles puntos del programa en el que un Aspecto se puede aplicar.



PointCut

Subconjuntto de los JoinPoint a los que vamos a aplicar un aspecto determinado



Ejemplo en código

```
package com.ejemploasp.servicios;

import org.springframework.stereotype.Service;

@Service
public class ServicioA {

 public void metodo1() {

 try {
 Thread.currentThread().sleep(3000);
 } catch (InterruptedException e) {
 // TODO Auto-generated catch block
 e.printStackTrace();
 }
 System.out.println("metodo1");
 }

 public void metodo2() {

 System.out.println("metodo2");
 }
}
```

```
package com.ejemploasp;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

import com.ejemploasp.servicios.ServicioA;

@Controller
public class ControladorHola {

 @Autowired
 ServicioA miservicio;
 @RequestMapping("/hola")
 public String hola() {

 miservicio.metodo1();
 miservicio.metodo2();
 return "bienvenido";

 }
}
```

Clase SprintgBoot que tiene activado aspectos

```
package com.ejemploasp;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.EnableAspectJAutoProxy;

@SpringBootApplication
@EnableAspectJAutoProxy
@ComponentScan(basePackages = "com.ejemploasp")
public class SpringAspectosApplication {

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

¿Cómo implementamos un control de tiempo de ejecución con aspectos?

```
package com.ejemploasp;

import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.Around;
import org.aspectj.lang.annotation.Aspect;
import org.springframework.stereotype.Component;

@Aspect
@Component
public class Aspecto {

	@Around("execution(* com.ejemploasp.servicios.*.*())")
	public void tiempoPasado(ProceedingJoinPoint punto) throws Throwable {

		Long tiempo1 = System.currentTimeMillis();
		punto.proceed();
		Long tiempo2 = System.currentTimeMillis();
		Long total = tiempo2 - tiempo1;
		if (total > 2000)
			System.out.format("el metodo es : %s y el tiempo transcurrido %d\n", punto.getSignature().getName(),
		total);
	}
}
```