

Taller de Arquitecturas de Servidores de Aplicaciones, Meta-protocolos de Objetos, Patrón IoC y Reflexión

Requisitos de Finalización

Descripción

En este taller, los estudiantes deben construir un **servidor Web tipo Apache** en Java.

El servidor debe:

- Entregar páginas **HTML**.
- Servir imágenes tipo **PNG**.
- Proveer un **framework IoC** para construir aplicaciones web a partir de **POJOs**.
- Permitir la construcción de una **aplicación Web de ejemplo** usando el servidor.
- Atender **múltiples solicitudes no concurrentes**.

El proyecto debe demostrar las capacidades **reflexivas de Java**, permitiendo:

- Cargar al menos un **bean (POJO)**.
- Derivar una aplicación Web a partir de dicho bean.

Se deberá entregar un prototipo mínimo al final del laboratorio. Luego se podrá complementar y pulir para la entrega final. **Se compararán los commits del inicio del laboratorio y de la entrega final.**

Sugerencias

1. Primera versión: cargar el POJO desde la línea de comandos

Similar al funcionamiento de los frameworks de pruebas.

Ejemplo de ejecución:

```
java -cp target/classes co.edu.escuelaing.reflexionlab.MicroSpringBoot \
co.edu.escuelaing.reflexionlab.FirstWebService
```

2. Usar la anotación **@GetMapping**

Publicar el servicio en la URI indicada. Limitar el retorno a **String**.

Ejemplo:

```
@RestController
public class HelloController {
```

```
@GetMapping("/")
public String index() {
    return "Greetings from Spring Boot!";
}
}
```

3. Versión final: escaneo automático de clases

El framework debe:

- Explorar el **directorio raíz** (o **classpath**)
 - Detectar clases anotadas con **@RestController**
 - Cargar **todas** las clases con dicha anotación automáticamente
 - Evitar tener que pasárlas manualmente por la línea de comandos
-

4. Soporte requerido

El framework debe soportar:

- **@GetMapping**
 - **@RequestParam**
-

5. Debe ser posible implementar este componente:

```
@RestController
public class GreetingController {

    private static final String template = "Hello, %s!";
    private final AtomicLong counter = new AtomicLong();

    @GetMapping("/greeting")
    public String greeting(
        @RequestParam(value = "name", defaultValue = "World") String name
    ) {
        return "Hola " + name;
    }
}
```

Entregables

1. **Código fuente del proyecto**, estructurado y manejado usando **Maven**.
2. El proyecto debe estar almacenado en un **repositorio público de GitHub** del estudiante.