El uso de Spring Boot properties es muy habitual cuando trabajamos con una aplicación de Spring Boot. A diferencia de otras aplicaciones clásicas de Spring Framework , Spring Boot hace uso del principio de convención sobre configuración y define un fichero por defecto de propiedades . Este fichero se encuentra en la carpeta resources de nuestro proyecto.

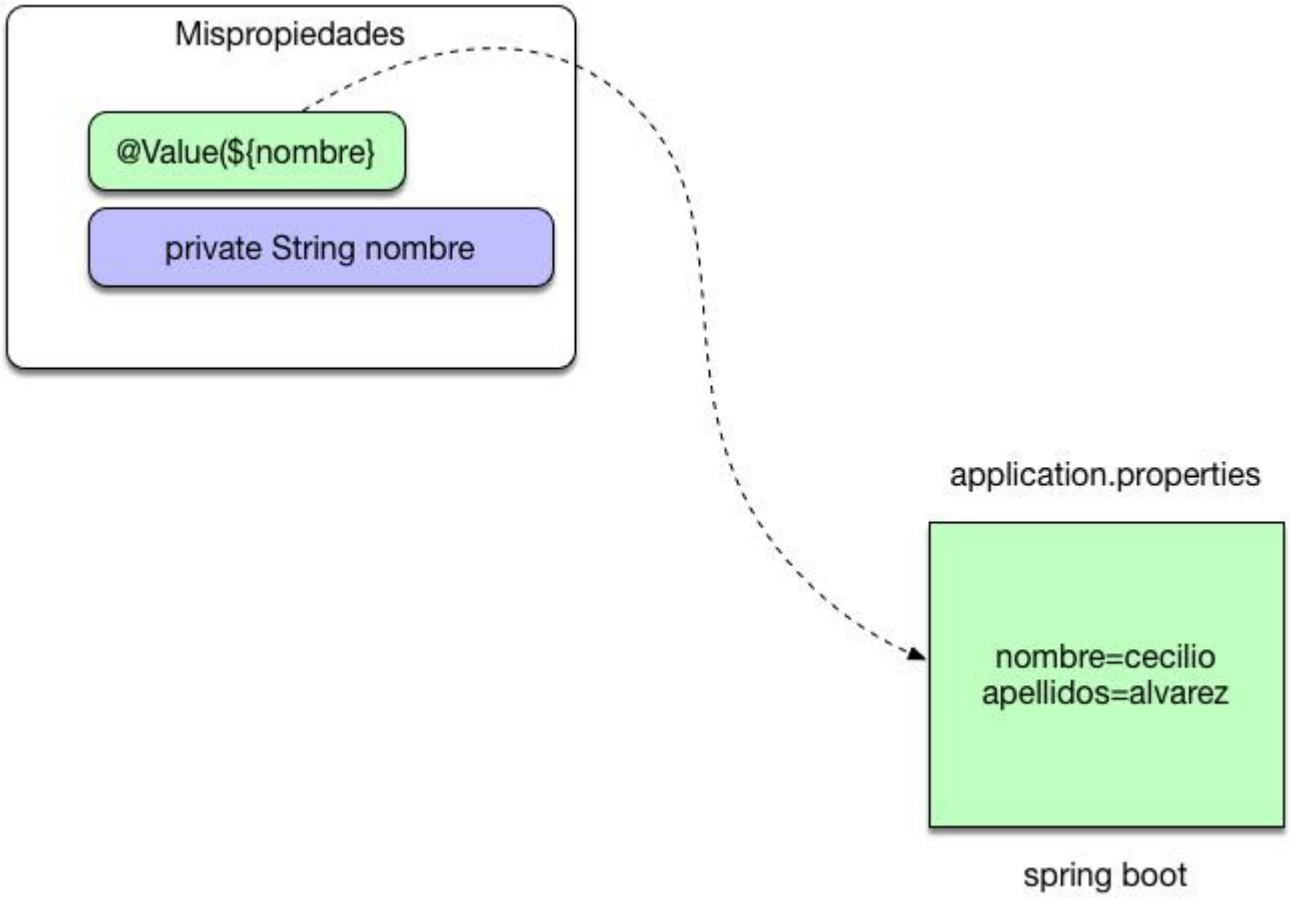
Spring Boot Properties

Vamos a usar este fichero de propiedades para dar de alta dos valores (un nombre y un apellido) y acceder a ellos desde nuestra aplicacion. Tenemos que configurar nuestra aplicación de Spring Boot con las anotaciones necesarias. En nuestro caso nos vamos a definir una clase Java sencilla que contenga ambas propiedades.

```
package com.arquitecturajava.springPrueba;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
```

```
@Component
public class MisPropiedades {
	@Value(&amp;amp;amp;quot;${nombre}&amp;amp;amp;quot;)
	private String nombre;
	@Value(&amp;amp;amp;quot;${apellidos}&amp;amp;amp;quot;)
	private String apellidos;
	public String getNombre() {
		return nombre;
	}
	public void setNombre(String nombre) {
		this.nombre = nombre;
	}
	public String getApellidos() {
		return apellidos;
	}
	public void setApellidos(String apellidos) {
		this.apellidos = apellidos;
	}

}
```

Como podemos observar nos hemos apoyado en la anotación @Value para inicializar las propiedades.

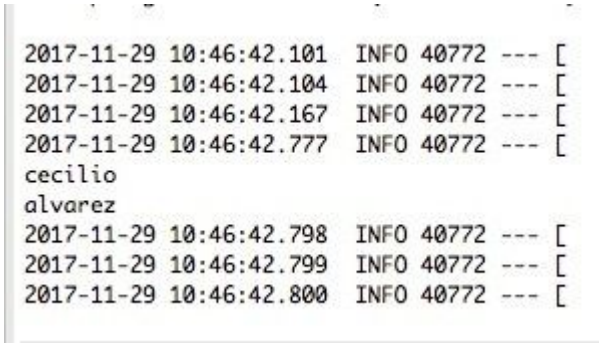


Hemos hecho uso de spring expression language para acceder a cada uno de los valores utilizando ${valor} . De esta forma tan sencilla seremos ya capaces de acceder a las propiedades definidas en el fichero application.properties. Nos queda simplemente inyectar nuestro clase en el proyecto de SpringBoot y ejecutarlo como proyecto de consola para ver el resultado.

```
package com.arquitecturajava.springPrueba;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
```

```
@SpringBootApplication
class SpringPruebaApplication implements CommandLineRunner {

	@Autowired
	MisPropiedades propiedades;

	public static void main(String[] args) {

		SpringApplication app = new
SpringApplication(SpringPruebaApplication.class);
		app.run(args);
	}

	@Override
	public void run(String... arg0) throws Exception {

		System.out.println(propiedades.getNombre());
		System.out.println(propiedades.getApellidos());

	}

}
```

Ahora nos es suficiente con ejecutar nuestra aplicación desde Eclipse y veremos como se imprime el nombre y los apellidos por la consola.



Acabamos de utilizar las capacidades de Spring Boot para manejo de propiedades.

Otros artículos relacionados

1. Spring Boot WAR sin Microservicios
2. ¿Qué es Spring Boot?
3. Spring Boot AOP y rendimiento
4. Spring Boot