

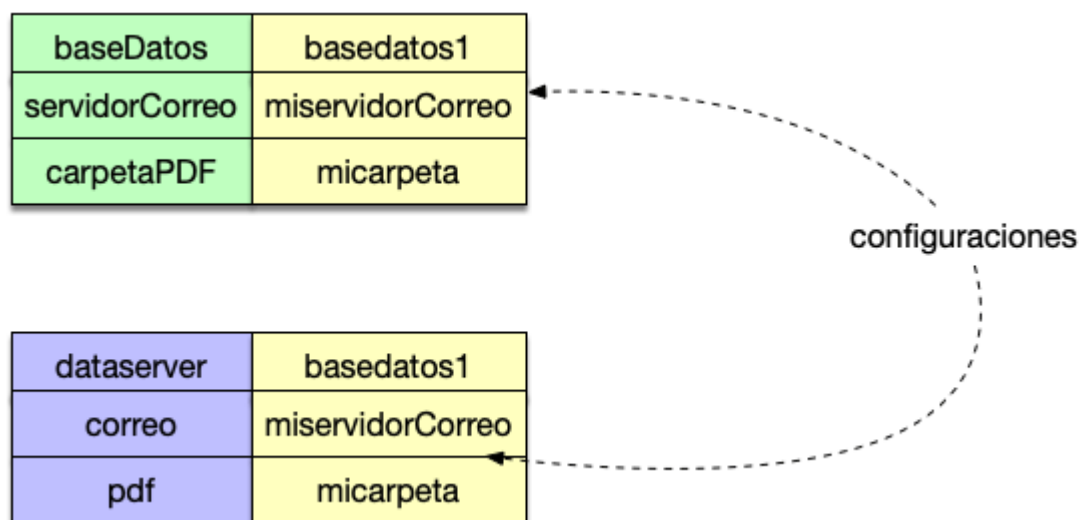
El principio de Convención sobre Configuración es uno de los principios más importantes de ingeniería de software. ¿Para qué sirve este principio? . Se trata de un principio que su objetivo es simplificar el día a día del desarrollo. La mayor parte de las veces cuando desarrollamos software hay una parte del tiempo de desarrollo que lo dedicamos a tareas de configuración de ese software. Por ejemplo podemos tener un fichero de propiedades que ponga

```
baseDatos= basedatos1
servidorCorreo=miservidorCorreo.com
carpetaPDF=micarpeta
```

Esto es simplemente una labor de configuración que haremos por cada aplicación que demos de alta. Sin embargo nos podemos encontrar que otro desarrollador decidió que su fichero de configuración sea el siguiente:

```
dataserver= basedatos1
correo=miservidorCorreo.com
pdf=micarpeta
```

Es cierto que almacena la misma información para también es cierto que la configuración no se parece en nada . Esto siempre genera muchos problemas.



Porque abre las puertas a opciones como la siguiente cuando uno realiza la configuración.

```
baseDatos= basedatos1
servidoCorreo=miservidorCorreo.com
carpetaPDF=micarpeta
```

Los datos parecen correctos como al principio pero lamentablemente “servidoCorreo” le falta la “r” de “servidor” .

baseDatos	basedatos1
servidoCorreo	miservidorCorreo
carpetaPDF	micarpeta

Un pequeño fallo, pero un fallo que nos puede volver locos. El principio de convención sobre configuración nos dice que definamos una convención “fija” y nos olvidemos de “configurar” continuamente las parametrizaciones siempre que sea posible. Un ejemplo sencillo sería

```
database.server=basedatos1
mail.server=miservidorCorreo.com
app.folder=micarpeta
```

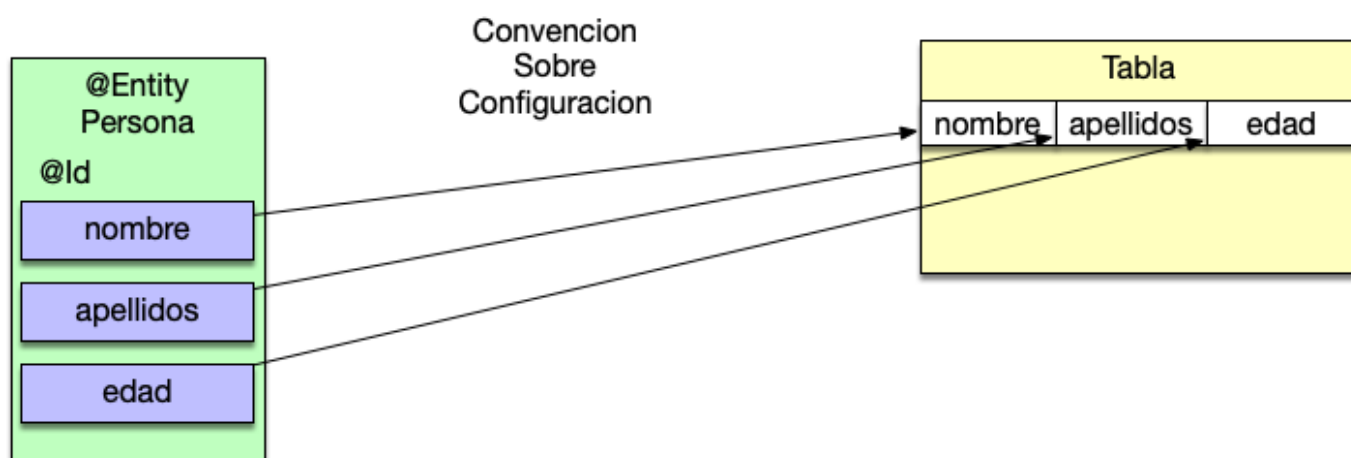
Los nombres son más flexibles y asumimos que por convención siempre se denominan database.server , mail.server y app.folder . Aquí aunque nos cueste verlo también estamos asumiendo otra convención que es que cada palabra está separada por un “.”. Por lo tanto copiaremos y pegaremos el fichero y modificaremos valores , no propiedades.

Convención sobre Configuración y transparencia

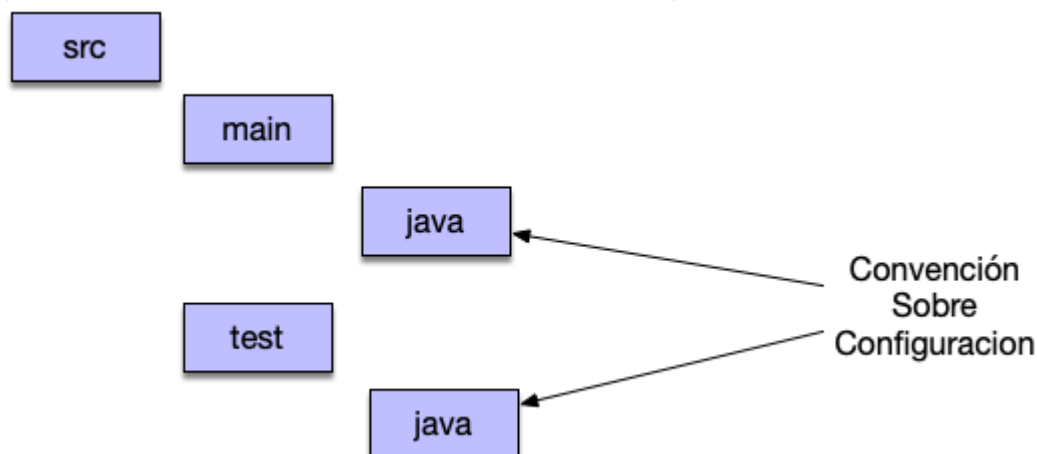
A veces parece que este principio se usa poco , pero la realidad es la contraria , se usa en

prácticamente todos los lugares , lo que sucede es que muchas veces ni nos paramos a pensar en ello. Veamos algunos ejemplos

- Cuando nosotros definimos una entidad en JPA con @Entity y @Id , estamos usando el principio de Convención sobre configuración ya que JPA entiende que el nombre de la tabla de la base de datos coincide con el JavaBean y los campos coinciden con las columnas. Hemos aplicado una convención.

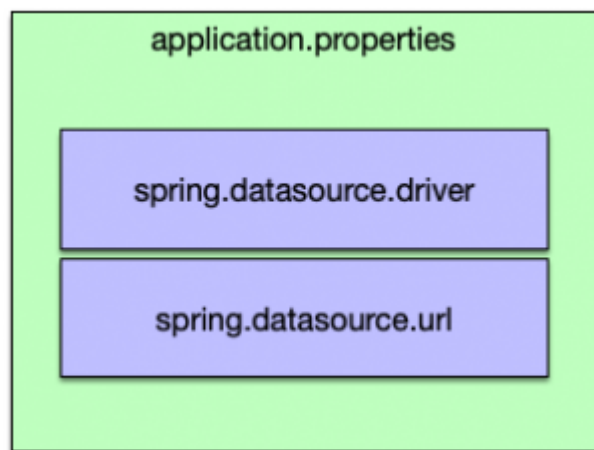


- Cuando construimos un proyecto de maven y las clases Java se almacenan en `src/main/java` y las clases de test se almacenan en `src/test/java`. Estamos ante otro uso del principio COC.



- Cuando usamos Spring Boot y configuramos un DataSource en el fichero de

application.properties estamos ante otro claro ejemplo de uso de Convención sobre Configuración ya que el fichero se ha de llamar así y las propiedades tienen también una denominación fija.



Existen otras muchas situaciones en las que ese principio se usa , pero cómo he dicho una de las características que destaca de él es su transparencia. En muchas ocasiones ni nos damos cuenta de que se está aplicando. Pero es uno de los principios que más ayuda a generar soluciones sencillas y elegantes.

Otros artículos relacionados

1. [DRY vs DAMP dos principios importantes](#)
2. [Adaptadores y patrones y el principio OCP](#)
3. [Java Fechas y el principio SRP](#)
4. [Wikipedia](#)