

## CUE: SPRING

### DRILLING: BIBLIOTECARIO PEREZOSO

Para resolver este ejercicio, debe haber revisado la lectura y los videos de la CUE Spring.

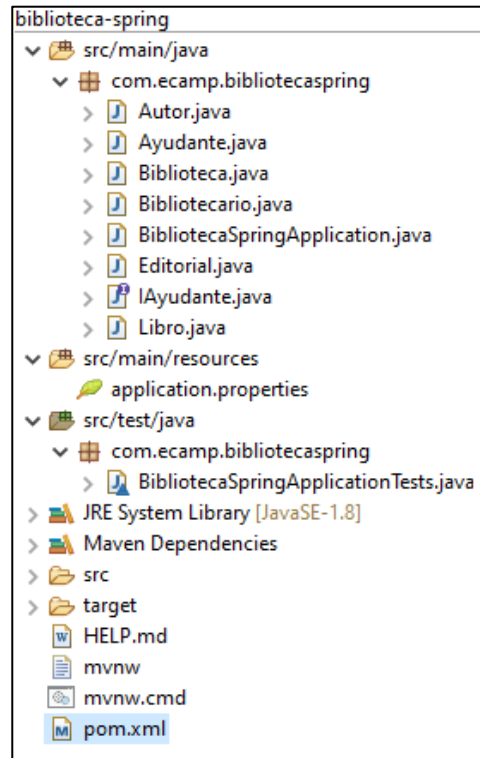
### EJERCICIO:

Se requiere crear una aplicación Spring que simule parcialmente el funcionamiento de una Biblioteca que cuenta con un bibliotecario y un ayudante. El bibliotecario es perezoso y le entrega todo el trabajo al ayudante. Lo único que hace es dar las ordenes al ayudante y reportar lo que sucedió. Nunca sale de recepción, donde atiende al público.

Como el bibliotecario puede tener varios ayudantes, y no le importa como hagan su trabajo, creó un contrato donde especifica el comportamiento de los ayudantes. Así podrá entregarles las mismas órdenes a todos y obtener los mismos resultados. Este contrato se representa en la forma de la siguiente interfaz Java:

```
1 public interface IAyudante {  
2  
3     public List<Libro> buscarLibro(String frase);  
4     //se obtienen libros segun la frase buscada en el nombre,  
5     autor, editorial, genero, etc.  
6     public List<Libro> obtenerLibros();  
7     //Se obtienen todos los libros de la biblioteca  
8     public void agregarLibro(Libro libro);  
9     //se agrega un nuevo libro a la biblioteca  
10    public Libro cambiarDisponibilidad(int id);  
11    //Disponible: disponible = true, Prestado disponible = false  
12 }
```

El proyecto debe tener la siguiente estructura:



Note los archivos: *Autor*, *Ayudante*, *Biblioteca*, *Bibliotecario*, *Editorial*, *IAyudante*, *Libro*.

Estos son los que deben crearse para completar el ejercicio, solo estas clases e interfaz. A continuación, se explica las asociaciones y atributos de las clases.

## **LIBRO:**

Tiene Los siguientes atributos:

```
1 int id
2 String titulo
3 Autor autor
4 Editorial editorial
5 int pagina
6 String genero
7 boolean disponible
```

donde *Autor* y *Editorial* son clases que deben ser inyectadas por Spring.

## **AUTOR:**

Tiene los siguientes atributos:

```
1 String nombre
2 String apellido
```

## EDITORIAL:

Tiene los siguientes atributos:

```
1 String nombre;  
2 String pais;
```

## BIBLIOTECA:

Solo contiene una lista de *Libros*. Solo acá se guardan todos los libros de la biblioteca.

## AYUDANTE:

Solo contiene un atributo de clase *Biblioteca*. Este debe ser inyectado por Spring. Además, esta clase debe implementar la interfaz **IAyudante**. Es el único que tiene acceso a la *Biblioteca* y su lista de libros.

## BIBLIOTECARIO:

Solo posee un atributo *ayudante* de tipo **IAyudante**, el cual le es asignado por Spring. Además, debe realizar las siguientes acciones **con el mínimo de esfuerzo posible**:

```
1 public void solicitanTodosLosLibros() {  
2 //ordenar al ayudante entregar la lista de todos los libros  
3 //imprimir por pantalla el resultado de lo ordenado (como ud.  
4 Desea)  
5 }  
6  
7 public void agregarUnLibro(Libro libro) {}  
8 //ordenar al ayudante a agregar un libro a la biblioteca  
9 //imprimir por pantalla el resultado de lo ordenado (como ud.  
10 Desea)  
11 }  
12  
13 public void devolucionOPrestamoDeLibro(int id) {  
14 //ordenar al ayudante a cambiar la disponibilidad de un libro por  
15 id  
16 //imprimir por pantalla el resultado de lo ordenado (como ud.  
17 Desea)  
18 }  
19  
20 public void buscarLibro(String frase) {  
21 //ayudante! busca una lista de libros según esta frase  
22 //imprimir por pantalla el resultado de lo ordenado (como ud.  
23 Desea)  
24 }
```

El proyecto debe ejecutar, con el método `run()` de `CommandLineRunner`, los métodos de una instancia de `Bibliotecarios`: 4 agregaciones de libros, 2 devoluciones o prestamos, solicitar todos los libros, buscar un libro por frase a elección.