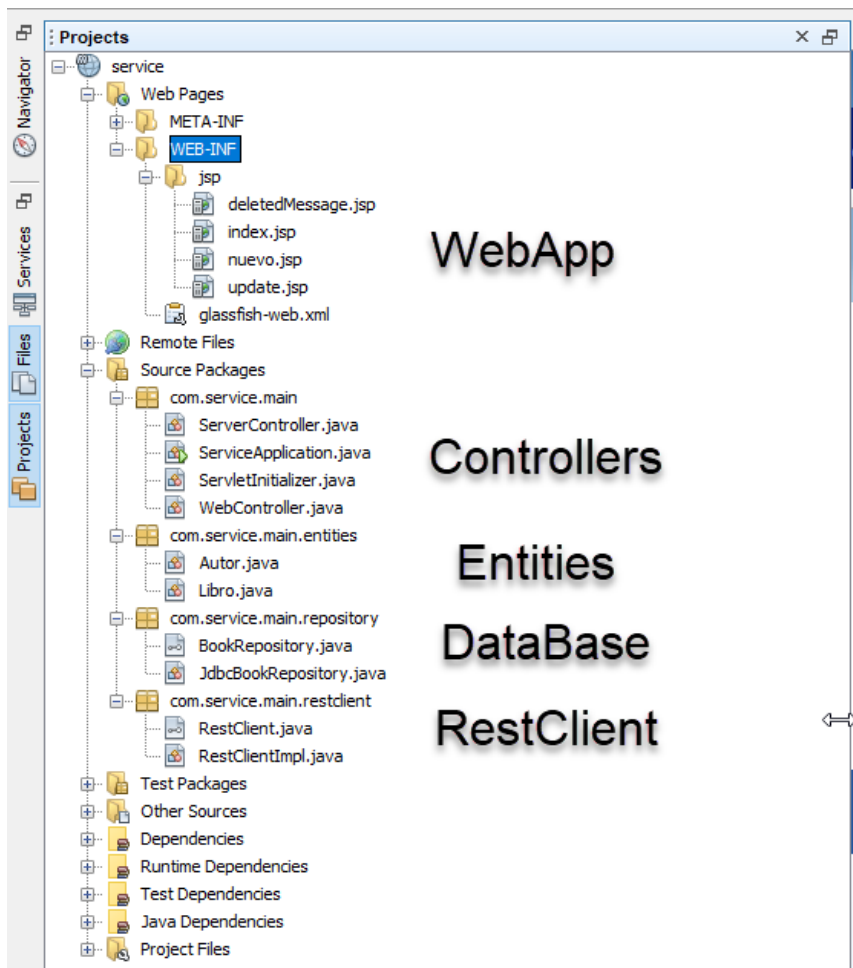


Respuesta a la Prueba

Estructura de proyecto, el proyecto principal contiene los dos componentes requeridos en la prueba que son: un WebService y un Cliente Web



1. Componente WebService

El componente de Servicios es un `@RestController` con todos los métodos necesarios para satisfacer los requerimientos

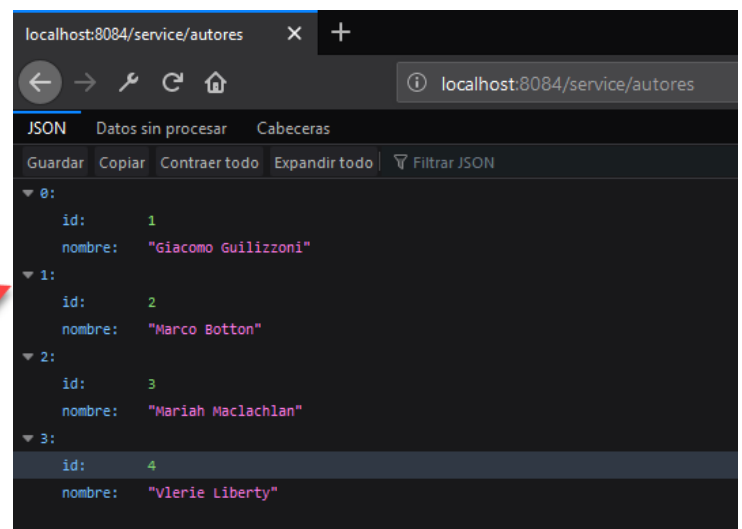
Vista de algunos métodos del `@RestController`

```
/**
 *
 * @author John
 */
@RestController
public class ServerController {

    @Autowired
    @Qualifier("jdbcBookRepository")
    private BookRepository repository;

    @GetMapping("/libros")
    public List<Libro> index() {
        return repository.libros();
    }

    @GetMapping("/autores")
    public List<Autor> autores() {
        return repository.autores();
    }
}
```



Vista de algunos métodos del @Repository para el acceso a los datos. Esta implementada con JdbcTemplate de Spring con un DataSource principal conectado con postgresQL.

```
147     @Override
148     public String getNameById(Long id) {
149         return jdbcTemplate.queryForObject(
150             "select name from books where id = ?",
151             new Object[]{id},
152             String.class
153         );
154     }
155
156     @Override
157     public List<Autor> autores() {
158         return jdbcTemplate.query(
159             "select * from autor",
160             (rs, rowNum)
161             -> new Autor(
162                 rs.getInt("id_autor"),
163                 rs.getString("nombre")
164             )
165         );
166     }
```

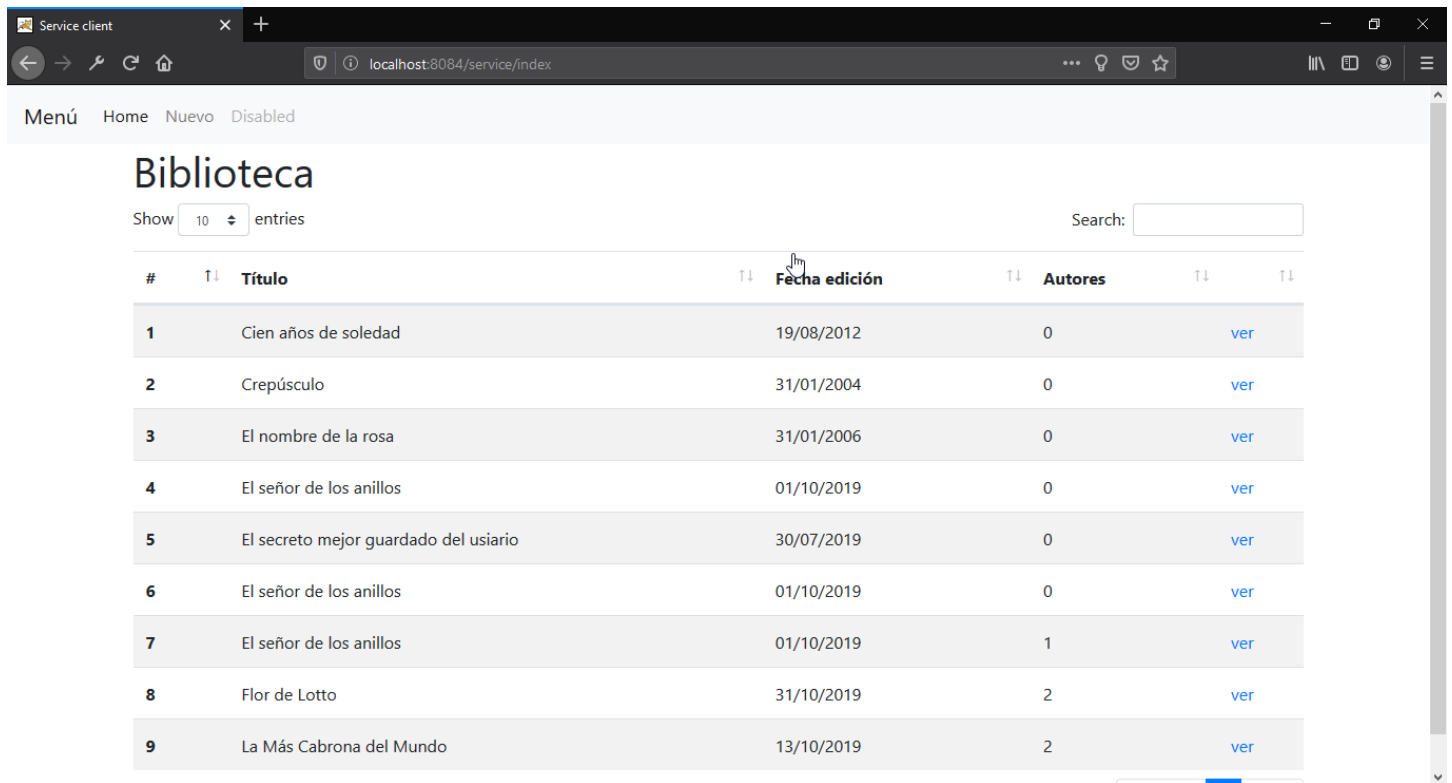
El archivo de Configuración de Spring Boot

```
1  ## Logging
2  logging.level.org.springframework=info
3  logging.level.com.mkyong=INFO
4  logging.level.com.zaxxer=DEBUG
5  logging.level.root=ERROR
6
7  logging.pattern.console=%-5level %logger{36} - %msg%n
8
9  ## PostgreSQL
10 spring.datasource.driver-class-name=org.postgresql.Driver
11 spring.datasource.url=jdbc:postgresql://localhost:5432/books
12 spring.datasource.username=root
13 spring.datasource.password=123
14
15 ## Web client
16 spring.mvc.view.prefix=/WEB-INF/jsp/
17 spring.mvc.view.suffix=.jsp
```

2. Componente Cliente Web

El index de la aplicación: /service/index

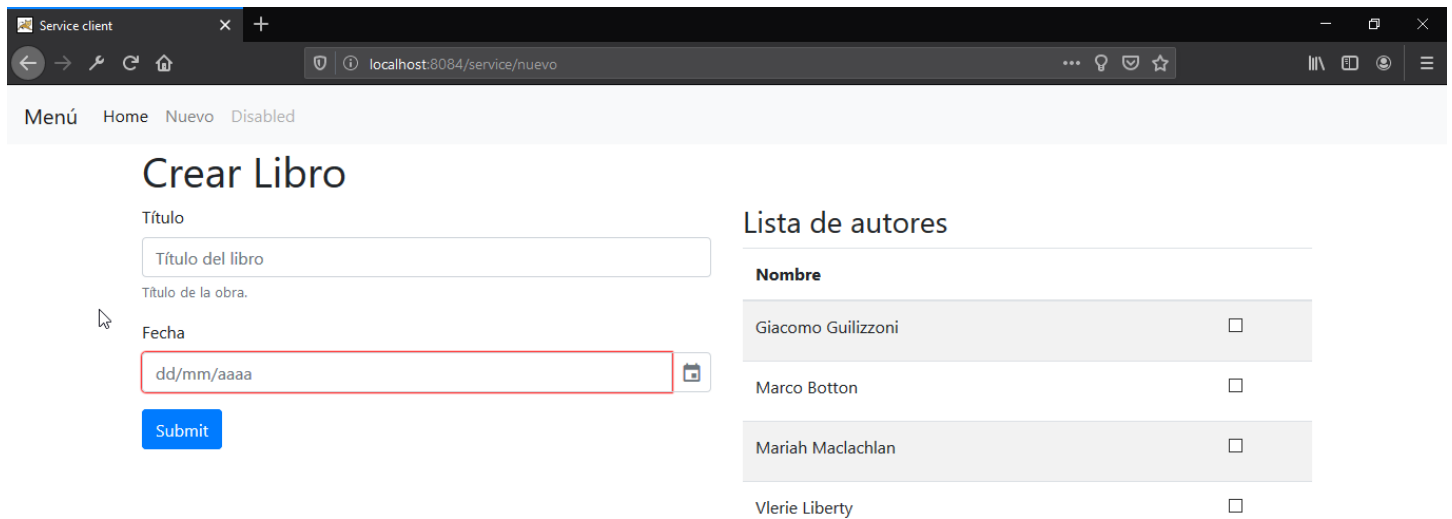
La tabla HTML se ha potenciado con DataTables para tareas de ordenamiento, filtros, además de la posibilidad de incrementar tareas de exportación de datos entre otras ventajas.



The screenshot shows a web browser window with the address bar at localhost:8084/service/index. The page has a navigation bar with 'Menú', 'Home', 'Nuevo', and 'Disabled'. The main heading is 'Biblioteca'. Below it, there's a 'Show 10 entries' dropdown and a search box. A table lists books with columns for '#', 'Título', 'Fecha edición', 'Autores', and a 'ver' link. The table contains 9 rows of data.

#	Título	Fecha edición	Autores	
1	Cien años de soledad	19/08/2012	0	ver
2	Crepúsculo	31/01/2004	0	ver
3	El nombre de la rosa	31/01/2006	0	ver
4	El señor de los anillos	01/10/2019	0	ver
5	El secreto mejor guardado del usuario	30/07/2019	0	ver
6	El señor de los anillos	01/10/2019	0	ver
7	El señor de los anillos	01/10/2019	1	ver
8	Flor de Lotto	31/10/2019	2	ver
9	La Más Cabrona del Mundo	13/10/2019	2	ver

El formulario de alta de nuevos libros



The screenshot shows a web browser window with the address bar at localhost:8084/service/nuevo. The page has a navigation bar with 'Menú', 'Home', 'Nuevo', and 'Disabled'. The main heading is 'Crear Libro'. Below it, there's a form with fields for 'Título' (with a sub-label 'Título del libro'), 'Fecha' (with a sub-label 'Título de la obra.' and a date format 'dd/mm/aaaa'), and a 'Submit' button. To the right, there's a section titled 'Lista de autores' with a list of authors and checkboxes: Giacomo Guilizzoni, Marco Botton, Mariah Maclachlan, and Vlerie Liberty.

Crear Libro

Título
Título del libro

Título de la obra.
Fecha
dd/mm/aaaa

[Submit](#)

Lista de autores

Nombre

- Giacomo Guilizzoni ☐
- Marco Botton ☐
- Mariah Maclachlan ☐
- Vlerie Liberty ☐

Vista de la interfaz de RestClient

```
7  /**
8   *
9   * @author John
10  */
11  public interface RestClient {
12
13      List<Autor> autores();
14
15      List<Libro> libros();
16
17      void guardarLibro(Libro libro);
18
19      Libro getLibroById(Integer id);
20
21      void eliminarLibro(Libro libro);
22  }
23
```

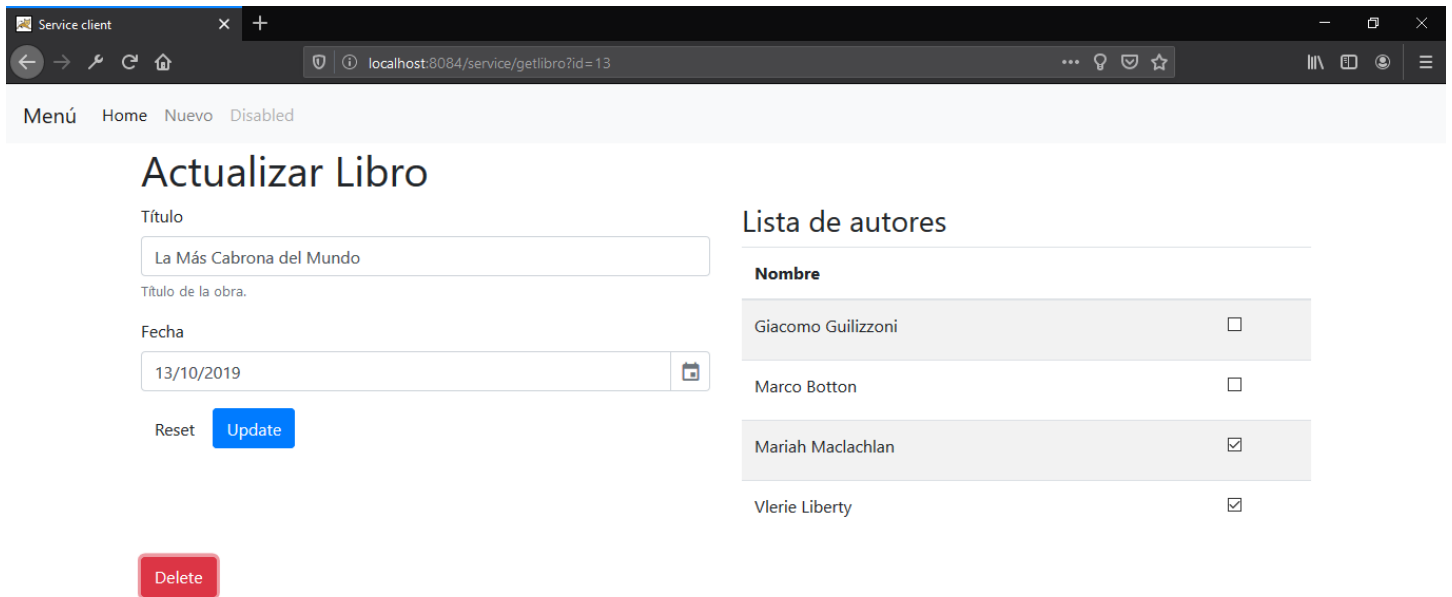
La implementación muestra la técnica utilizada para la integración.

```
16  * @author John
17  */
18  @Service
19  public class RestClientImpl implements RestClient {
20
21      final private String host = "http://localhost:8084/service";
22
23      @Override
24      public List<Autor> autores() {
25          String theUrl = host + "/autores";
26
27          RestTemplate restTemplate = new RestTemplate();
28          ResponseEntity<List<Autor>> response = restTemplate.exchange(
29              theUrl,
30              HttpMethod.GET,
31              null,
32              new ParameterizedTypeReference<List<Autor>>() {
33              });
34          List<Autor> result = response.getBody();
35          return result;
36      }
37
```

WebService

RestTemplate

Vista de Libro y edición de sus datos, que además incluye la función de eliminar



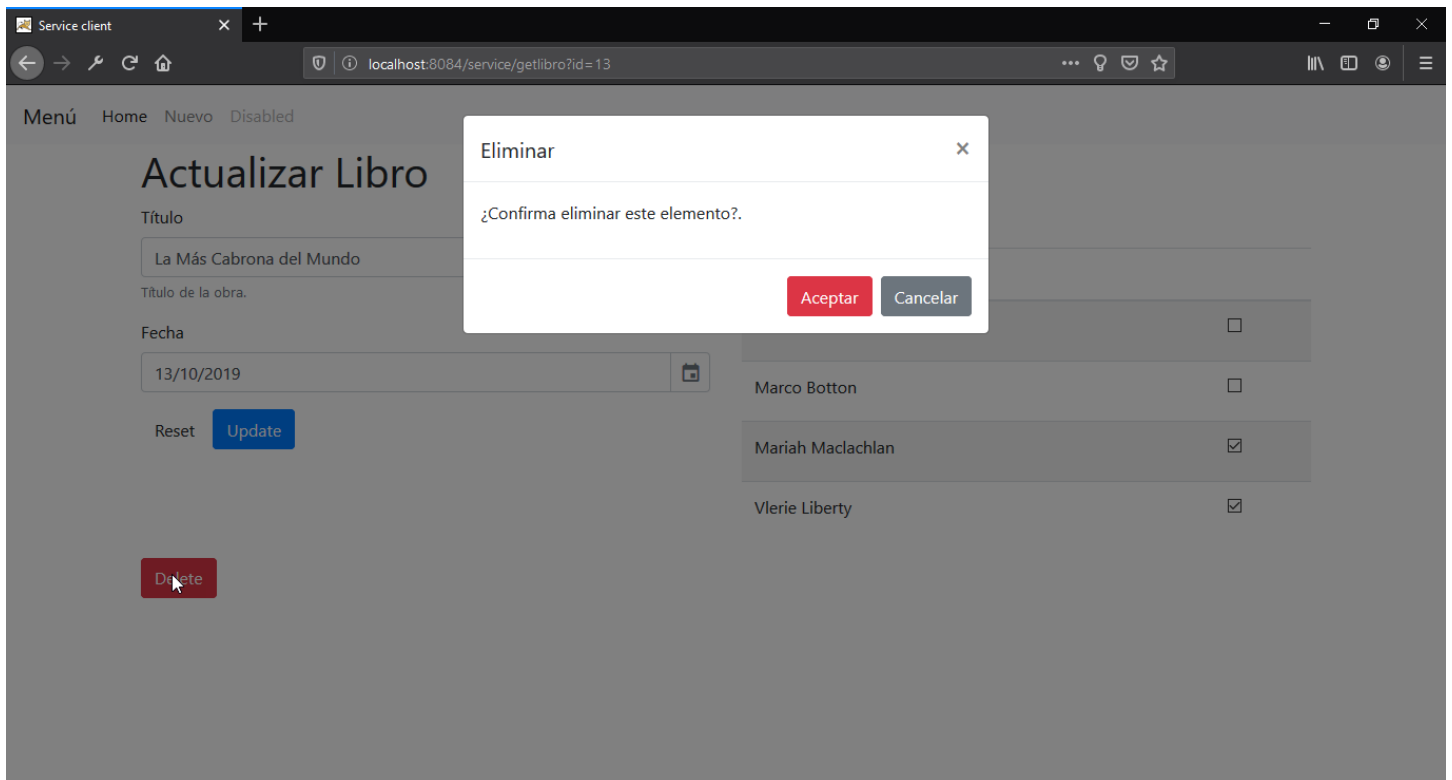
The screenshot shows a web application interface with a dark-themed browser window. The address bar shows 'localhost:8084/service/getlibro?id=13'. The page has a navigation bar with 'Menú', 'Home', 'Nuevo', and 'Disabled'. The main content area is titled 'Actualizar Libro'. It contains a form with the following fields:

- Título:** A text input field containing 'La Más Cabróna del Mundo'.
- Título de la obra:** A text input field.
- Fecha:** A date input field containing '13/10/2019'.
- Buttons:** 'Reset' and 'Update' (blue).

Below the form is a red 'Delete' button. To the right, there is a section titled 'Lista de autores' with a table of authors:

Nombre	
Giacomo Guilizzoni	<input type="checkbox"/>
Marco Botton	<input type="checkbox"/>
Mariah Maclachlan	<input checked="" type="checkbox"/>
Vlerie Liberty	<input checked="" type="checkbox"/>

Eliminar contiene un paso extra de confirmación



This screenshot shows the same web application interface as the previous one, but with a confirmation dialog box open over the 'Delete' button. The dialog box is titled 'Eliminar' and contains the text '¿Confirma eliminar este elemento?.'. It has two buttons: 'Aceptar' (red) and 'Cancelar' (gray). The background of the application is dimmed.

Conclusiones

El código y la base de datos se encuentran en un repositorio GitHub

<https://github.com/johngnu/springws-demo>

Proyecto desarrollado con NetBeans 8.2 con Tomcat

Saludos

John Castillo

02 de Octubre de 2019