

## **Cargue de Datos archivos CSV con Spring Boot.**

### **1. Creación de la Tabla en la Base de Datos**

Si no hemos creado la Base de Datos completa debemos crear la tabla que gestione los datos de los productos, en las siguientes líneas existe un ejemplo de ello.

- Crear Tabla

```
CREATE TABLE productos (  
    codigo INT NOT NULL,  
    nombre VARCHAR(255) NOT NULL,  
    nitproveedor int(15) NOT NULL,  
    precio_compra double NOT NULL,  
    iva double NOT NULL,  
    precio_venta double NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
ALTER TABLE productos  
    ADD PRIMARY KEY (codigo);
```

```
ALTER TABLE productos  
    ADD CONSTRAINT PRODUCTOS_PROVEEDOR FOREIGN KEY  
    (nitproveedor) REFERENCES proveedor (nit);
```

- Ejecución en MariaDB.

```
1 use proyecto;  
2  
3 CREATE TABLE productos (  
4     codigo INT NOT NULL,  
5     nombre VARCHAR(255) NOT NULL,  
6     nitproveedor int(15) NOT NULL,  
7     precio_compra double NOT NULL,  
8     iva double NOT NULL,  
9     precio_venta double NOT NULL  
10 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
11  
12 ALTER TABLE productos  
13     ADD PRIMARY KEY (codigo);  
14  
15 ALTER TABLE productos  
16     ADD CONSTRAINT PRODUCTOS_PROVEEDOR FOREIGN KEY (nitproveedor) REFERENCES proveedor (nit);  
17
```

## 2. Creación del Proyecto.

Creamos un proyecto en su IDE de preferencia con Spring Boot tipo “war”, seleccionamos las dependencias Spring web, Dev Tools y el conector a la Base de Datos de MySQL, nuestro proyecto se llamará CargarDatosCSV.

## 2. Código de la página de bienvenida predeterminada

Si tiene la página de bienvenida estática ( index.html ) para la aplicación, colóquela en el directorio src / main / resources / static. Además vamos a agregar el contenido de Bootstrap copiando en la carpeta del ejercicio la estructura de los archivos css, js y fonts, el archivo index.html posee el siguiente contenido:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Cargue de Productos desde csv</title>
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="css/bootstrap.min.css">
8  </head>
9  <body>
10     <div class="jumbotron text-center">
11         <h1>Desarrollo de Software.</h1>
12         <h2>Universidad El Bosque</h2>
13         <p>Cargue de Datos archivos CSV a Base de Datos Spring Boot!</p>
14     </div>
15     <div class="container-fluid">
16         <h1 style="text-align: center;">Listado de Productos</h1>
17         <h3 style="text-align: center;">Cargue de Archivo CSV</h3>
18         <br/><br/>
19         <div class="row">
20             <div class="col-sm-4">
21
22             </div>
23             <div class="col-sm-4">
24                 <form action="Controlador" method="post" class="form-control-static">
25                     <input type="file" name="nombre" accept=".csv"
26                         class="form-group" size="50">
27                     <br/>
28                     <input type="submit" value="Subir Archivo" class="btn btn-primary btn-lg"><br/>
29                 </form>
30             </div>
31         </div>
32     </div>
33     <hr /><br/><br/>
34     <footer class="text-center">
35         <p>Derechos de autor: Carlos Adolfo Beltrán Castro - osocarbel@gmail.com</p>
36     </footer>
37 </body>
38 </html>
```

### 1. Ejecución Netbeans:

- Agregar el servidor de Aplicaciones tomcat.

- Compilar si es necesario (F11)
- Ejecutar con Run.
- Ahora si ejecutamos el proyecto con la opción de Run, debe aparecer una pantalla como la siguiente:



## Lista de Productos

Cargue de Archivo CSV

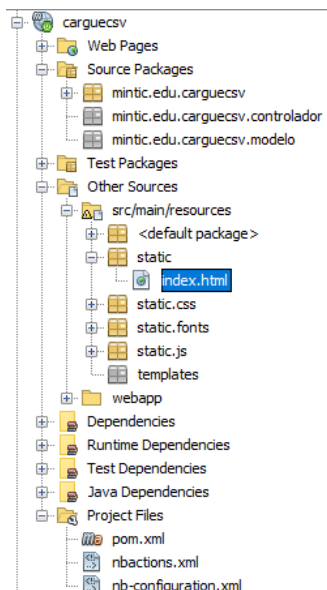
Examinar...

Subir Archivo

Derechos de autor: Carlos Adolfo Beltrán Castro - osocarbel@gmail.com

### 3. Crear estructura del Proyecto.

Agregamos un packages para el modelo y otro para el controlador dependiendo del principal creado con el proyecto.



#### 4. Clase Modelo de Datos.

Creamos la clase en modelo llamada Producto que tenga la misma estructura de la tabla de la Base de Datos, en la siguiente figura aparece la clase con los atributos y los constructores vacíos y completos:

```
1 package mintic.edu.carguecsv.modelo;
2
3 /**
4  *
5  * @author Carlos Beltrán
6  */
7 public class Producto {
8
9     private int codigo;
10    private String nombre;
11    private int nitproveedor;
12    private double precioCompra;
13    private double iva;
14    private double precioVenta;
15
16    // Constructores
17
18    public Producto() {
19    }
```

No olvidar agregar los getter, setter y el toString de la clase.

#### 5. Clase Conexion.

A continuación, dentro del package modelo cree una clase de llamada “Conexion”, que implemente los métodos para conectarnos a la Base de Datos, el contenido es similar al usado en los ejercicios anteriores.

#### 6. Clase LectorCSV.

Un CSV es un fichero de texto plano que consta de varios valores separados por comas (o punto y coma, según el caso), de modo que para procesar este tipo de ficheros en Java en principio debería bastar con leer el archivo línea a línea y partir las cadenas de texto por el separador correspondiente

A continuación, dentro del package modelo cree una clase de llamada “**LectorCSV**”, que implemente los métodos para leer los datos y vamos a convertirlo a una lista.

Se lee un archivo de texto tipo csv desde el index.html y vamos a ir al controlador e invocamos la clase que vamos a construir y después mostramos los datos en pantalla con una vista jsp.

```
1 package mintic.edu.carguecsv.modelo;
2
3 import java.io.BufferedReader;
4 import java.io.FileReader;
5 import java.io.IOException;
6 import java.util.Arrays;
7 import com.opencsv.CSVReader;
8 import java.util.ArrayList;
9 import java.util.List;
10
11 public class LectorCSV {
12     // Propiedades
13     private char separador;
14     private char comillas;
15
16     // Constructor
17
18     /**
19      * Inicializa el constructor definiendo el separador de los campos y las comillas usadas
20      * @param separador
21      * @param comillas
22      */
23     public LectorCSV(char separador, char comillas) {
24         this.separador = separador;
25         this.comillas = comillas;
26     }
27
28     // Métodos
29     /**
30      * Lee un CSV que no contiene el mismo caracter que el separador en su texto
31      * y sin comillas que delimiten los campos
32      * @param path Ruta donde está el archivo
33      * @throws IOException
34      */
35     public List<Producto> leerCSVSimple(String path) throws IOException {
36         // Abro el .csv en buffer de lectura
37         BufferedReader bufferLectura = new BufferedReader(new FileReader(path));
38         String linea = bufferLectura.readLine(); // Leo una línea del archivo
39         List<Producto> productos = new ArrayList<>();
40         while (linea != null) {
41             // Separa la línea leída con el separador definido previamente
42             String[] campos = linea.split(String.valueOf(this.separador));
43             Producto pro = new Producto(); // Instanciamos un objeto tipo Proveedor
44             pro.setCodigo(Integer.parseInt(campos[0]));
45             pro.setNombre(campos[1]);
46             pro.setNitproveedor(Integer.parseInt(campos[2]));
47             pro.setPrecioCompra(Double.parseDouble(campos[3]));
48             pro.setIva(Double.parseDouble(campos[4]));
49             pro.setPrecioVenta(Double.parseDouble(campos[5]));
50             productos.add(pro); // Agregarlo al ArrayList
51             System.out.println(Arrays.toString(campos));
52             // Vuelvo a leer del fichero
53             linea = bufferLectura.readLine();
54         }
55         // Cierro el buffer de lectura
56         if (bufferLectura != null) {
57             bufferLectura.close();
58         }
59         return productos;
60     }
61 }
```

## 7. Controlador.

Creamos un servlet sobre el package respectivo que nos permita recuperar el nombre del archivo y su contenido, se agrega un atributo de la clase LectorCSV y agregamos al método la ruta correspondiente, se recupera el valor del archivo, se crea una lista tipo Producto, se llena la lista con el método creado en la clase LectorCSV, se asigna el valor a la variable a usar en la vista y se direcciona a la vista.

```

1  package mintic.edu.carguecsv.controlador;
2  import java.io.IOException;
3  import java.io.PrintWriter;
4  import java.util.ArrayList;
5  import java.util.List;
6  import javax.servlet.ServletException;
7  import javax.servlet.http.HttpServlet;
8  import javax.servlet.http.HttpServletRequest;
9  import javax.servlet.http.HttpServletResponse;
10 import mintic.edu.carguecsv.modelo.LectorCSV;
11 import mintic.edu.carguecsv.modelo.Producto;
12
13 public class Controlador extends HttpServlet {
14
15     LectorCSV lectorCSV = new LectorCSV(',', ' ');
16     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
17         throws ServletException, IOException {
18         // Recuperamos el nombre del archivo proveniente de la vista
19         String nombre = request.getParameter("nombre");
20         // Creamos un arreglo de tipo Producto
21         List<Producto> productos = new ArrayList<Producto>();
22         // Llenamos el arreglo con el método leerCSVSimple
23         productos = lectorCSV.leerCSVSimple(nombre);
24         // Asignamos el nombre de la variable a usar en la vista con el valor de la lista
25         request.setAttribute("productos", productos);
26         // Direccionamos la salida a la vista.jsp
27         request.getRequestDispatcher("jsp/vista.jsp").forward(request, response);
28     }
29
30     HttpServlet methods. Click on the + sign on the left to edit the code.
68
69 }

```

## 8. Ajuste para JSP y Cargue de CSV.

Para poder usar páginas JSP con JSTL debemos agregar las siguientes tres dependencias en el archivo pom.xml:

```

<!-- Need this to compile JSP -->
<dependency>
    <groupId>org.apache.tomcat.embed</groupId>
    <artifactId>tomcat-embed-jasper</artifactId>

```

```

        <scope>provided</scope>
    </dependency>

    <!-- JSTL -->
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
    </dependency>

    <dependency>
        <groupId>com.opencsv</groupId>
        <artifactId>opencsv</artifactId>
        <version>3.8</version>
    </dependency>

```

## 9. Contenido de vista.jsp.

La vista la dividimos en dos secciones, en la primera parte para mostrar los mensajes o aviso y en la segunda la lista de datos, en la siguiente figura se aprecia la primera parte.

```

1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
3  <!DOCTYPE html>
4  <html>
5  <head>
6      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7      <link rel="stylesheet" href="css/bootstrap.min.css">
8      <title>Datos del Archivo</title>
9  </head>
10 <body>
11     <div class="container-fluid">
12         <div class="row">
13             <h1 style="text-align: center;">Listado de Productos</h1>
14             <h3 style="text-align: center;">Contenido de Arreglo Subido desde el CSV</h3>
15             <br/><br/>
16             <div class="col-sm-2">
17                 <c:if test="${mensaje != null}" >
18                     <div class="alert alert-success alert-dismissible">
19                         <button type="button" class="close" data-dismiss="alert">&times;</button>
20                         <strong>Resultado!</strong> ${mensaje}
21                     </div>
22                 </c:if>
23                 <c:if test="${aviso != null}" >
24                     <div class="alert alert-danger alert-dismissible">
25                         <button type="button" class="close" data-dismiss="alert">&times;</button>
26                         <strong>Cuidado!</strong> ${aviso}
27                     </div>
28                 </c:if>
29             </div>

```

La parte final de la vista muestra la lista de datos usando el `forEach` de JSTL, agregamos el botón que nos permite regresar al inicio y los script que añaden jquery, opper y javascript.

```

30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

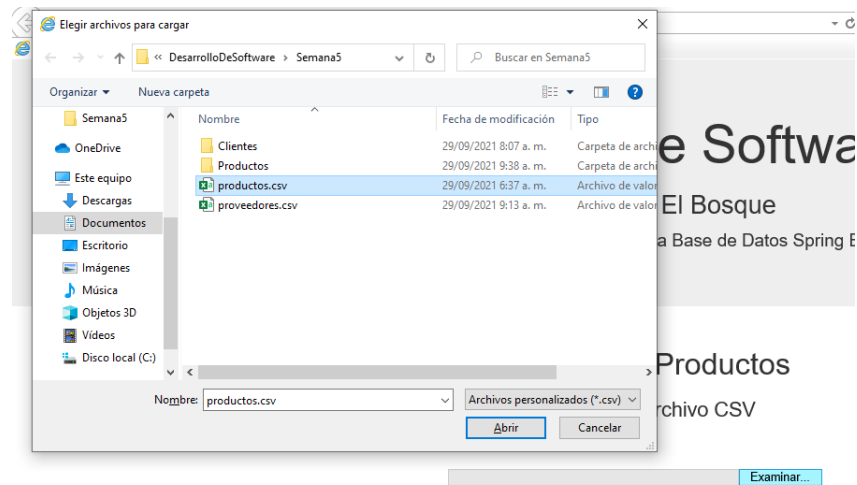
<div class="col-sm-6">
  <table border="0" cellspacing="5" cellpadding="5" class="table table-striped">
    <thead>
      <tr>
        <th>CODIGO</th>
        <th>NOMBRE</th>
        <th>PROVEEDOR</th>
        <th>PRECIO COMPRA</th>
        <th>IVA</th>
        <th>PRECIO COMPRA</th>
      </tr>
    </thead>
    <tbody>
      <c:forEach items="${productos}" var="pro">
        <tr>
          <td>${pro.codigo}</td>
          <td>${pro.nombre}</td>
          <td>${pro.nitproveedor}</td>
          <td>${pro.precioCompra}</td>
          <td>${pro.iva}</td>
          <td>${pro.precioVenta}</td>
        </tr>
      </c:forEach>
    </tbody>
  </table>
  <a href="/" class="btn btn-success btn-lg">Regresar al Inicio</a>
</div>
</div>
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpl"
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMl"
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6C"
</body>
</html>

```

## Prueba del proyecto.

Al presionar el botón de examinar aparece una pantalla que nos permite seleccionar el archivo a subir, nos podemos dar cuenta que solo muestra archivos .csv gracias a que en el index.html en el input tipo file usamos la propiedad `accept="*.csv"`. En la siguiente figura se muestra la pantalla referida.





Después de seleccionar el archivo aparece en la pantalla la ruta y el nombre del mismo y presionamos el botón de subir archivo.



Si el resultado es exitoso debe aparece la lista de productos como lo muestra la figura.

## Listado de Productos

Contenido de Arreglo Subido desde el CSV

Resultado!Carga exitosa

| CODIGO | NOMBRE      | PROVEEDOR | PRECIO COMPRA | IVA  | PRECIO COMPRA |
|--------|-------------|-----------|---------------|------|---------------|
| 1      | Melocotones | 1         | 25505.0       | 19.0 | 30351.0       |
| 2      | Manzanas    | 3         | 18108.0       | 19.0 | 21549.0       |
| 3      | Plátanos    | 4         | 29681.0       | 19.0 | 35320.0       |
| 4      | Lechuga     | 3         | 29788.0       | 19.0 | 35448.0       |
| 5      | Tomates     | 1         | 12739.0       | 19.0 | 15159.0       |
| 6      | Calabaza    | 1         | 21315.0       | 19.0 | 25365.0       |
| 7      | Apio        | 2         | 19249.0       | 19.0 | 22906.0       |
| 8      | Pepino      | 2         | 10958.0       | 19.0 | 13040.0       |
| 9      | Champiñones | 2         | 11046.0       | 19.0 | 13145.0       |
| 10     | Leche       | 5         | 21150.0       | 19.0 | 25169.0       |
| 11     | Queso       | 5         | 26571.0       | 19.0 | 31619.0       |

En caso contrario muestra la siguiente figura.

### Listado de Productos

Contenido de Arreglo Subido desde el CSV

**Cuidado!**Error al Cargue del Archivo ✕

| CODIGO | NOMBRE | PROVEEDOR | PRECIO COMPRA | IVA | PRECIO COMPRA |
|--------|--------|-----------|---------------|-----|---------------|
|--------|--------|-----------|---------------|-----|---------------|

Regresar al Inicio

## 10. Implementación del Cargue a la Base de Datos.

Para subir la información a la base de datos se puede procesar de varias formas, en cualquiera de ellas se debe crear un método en el DAO que inserte los registros de la lista o desde el csv.

### - Creación de la Clase ProductoDAO (Método agregarProducto).

Se crea una clase normal llamada ProductoDAO con la implementación de la creación de registros en la Base de Datos.

```

1  package mintic.edu.cargueCSV.modelo;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6  import java.sql.SQLException;
7  import java.sql.Statement;
8
9  public class ProductoDAO {
10     // Definir los Atributos. Capa de Datos. Se comunica con la BDs
11     Connection con = null; // Hacer la conexion a la BDs
12     Conexion cn = new Conexion();
13     Statement stm = null; // Separa el espacio para construir un comando SQL
14     ResultSet res = null; // Guarda el resultado de la consulta
15     PreparedStatement ps = null;
16

```

```

17 public boolean agregarProducto(Producto producto) {
18     boolean registrar = false; // Permite identificar si ya existe el producto
19     boolean encontrado = false; // Encuentra un producto con el correo Institucional
20     String buscar = "SELECT * FROM productos where codigo = " // Instrucción sql
21         + producto.getCodigo(); // Para buscar un registro con el mismo id
22     encontrado = buscar(buscar); // Ejecutamos el método con la consulta
23     if(!encontrado){
24         // La instrucción para insertar el registro
25         String sql = "INSERT INTO productos values (" + producto.getCodigo() + "," + producto.getNombre()
26             + "," + producto.getNitProveedor() + "," + producto.getPrecioCompra() + ","
27             + producto.getIva() + "," + producto.getPrecioVenta() + ")";
28         try {
29             con = cn.Conexion();
30             stm = con.createStatement();
31             stm.execute(sql);
32             registrar = true;
33             stm.close();
34             con.close();
35         } catch (SQLException e) {
36             System.out.println("Error: Clase ProductoDao, método agregarProducto");
37             e.printStackTrace();
38         }
39     }
40     return registrar;
41 }

43 public boolean buscar(String sql){
44     boolean encontrado = false;
45     con = cn.Conexion();
46     try {
47         stm = con.createStatement();
48         res = stm.executeQuery(sql);
49         while(res.next()){
50             encontrado = true;
51         }
52     } catch (SQLException e) {
53         System.out.println("Mensaje:"+e.getMessage());
54         System.out.println("Estado:"+e.getSQLState());
55         System.out.println("Codigo del error:"+e.getErrorCode());
56         System.out.println("Error: Clase ProductoDao, método agregarProducto"+e.getMessage());
57     }
58     return encontrado;
59 }
60
61 }

```

**- Ajustes al Controlador .**

```

1  package mintic.edu.cargueCSV.controlador;
2
3  import java.io.IOException;
4  import java.util.ArrayList;
5  import java.util.List;
6  import javax.servlet.ServletException;
7  import javax.servlet.http.HttpServlet;
8  import javax.servlet.http.HttpServletRequest;
9  import javax.servlet.http.HttpServletResponse;
10 import mintic.edu.cargueCSV.modelo.LectorCSV;
11 import mintic.edu.cargueCSV.modelo.Producto;
12
13 public class Controlador extends HttpServlet {
14
15     // Instanciar un objeto de la Clase LectorCSV
16     LectorCSV lectorCSV = new LectorCSV(',', ' ');
17     String mensaje = null;
18     String aviso = null;
19
20     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
21         throws ServletException, IOException {
22         try { // Recibimos el nombre y la ruta del archivo a cargar
23             String nombre = request.getParameter("nombre");
24             // Crear una lista tipo Producto
25             List<Producto> productos = new ArrayList<Producto>();
26             // Llenamos la lista con el método creado en LectorCSV
27             productos = lectorCSV.leerCSVSimple(nombre);
28             if (productos.size() != 0) {
29                 mensaje = "Archivo Cargado Exitosamente";
30                 aviso = null;
31             } else {
32                 mensaje = null;
33                 aviso = "Datos Leídos Inválidos";
34             }
35             // Definimos una variable a usar en la vista
36             request.setAttribute("productos", productos);
37         } catch (Exception e) {
38             System.out.println("No se seleccionó Archivo para cargar");
39             mensaje = null;
40             aviso = "No se seleccionó Archivo para cargar";
41         }
42
43         request.setAttribute("mensaje", mensaje);
44         request.setAttribute("aviso", aviso);
45         // Direccinamos la salida la vista deseada
46         request.getRequestDispatcher("jsp/vista.jsp").forward(request, response);
47     }
48
49     +
50     +
51     +
52     +
53     +
54     +
55     +
56     +
57     +
58     +
59     +
60     +
61     +
62     +
63     +
64     +
65     +
66     +
67     +
68     +
69     +
70     +
71     +
72     +
73     +
74     +
75     +
76     +
77     +
78     +
79     +
80     +
81     +
82     +
83     +
84     +
85     +
86     +
87     +
88     +
89     +
90     +
91     +
92     +
93     +
94     +
95     +
96     +
97     +
98     +
99     +
100    +
101    +
102    +
103    +
104    +
105    +
106    +
107    +
108    +
109    +
110    +
111    +
112    +
113    +
114    +
115    +
116    +
117    +
118    +
119    +
120    +
121    +
122    +
123    +
124    +
125    +
126    +
127    +
128    +
129    +
130    +
131    +
132    +
133    +
134    +
135    +
136    +
137    +
138    +
139    +
140    +
141    +
142    +
143    +
144    +
145    +
146    +
147    +
148    +
149    +
150    +
151    +
152    +
153    +
154    +
155    +
156    +
157    +
158    +
159    +
160    +
161    +
162    +
163    +
164    +
165    +
166    +
167    +
168    +
169    +
170    +
171    +
172    +
173    +
174    +
175    +
176    +
177    +
178    +
179    +
180    +
181    +
182    +
183    +
184    +
185    +
186    +
187    +
188    +
189    +
190    +
191    +
192    +
193    +
194    +
195    +
196    +
197    +
198    +
199    +
200    +
201    +
202    +
203    +
204    +
205    +
206    +
207    +
208    +
209    +
210    +
211    +
212    +
213    +
214    +
215    +
216    +
217    +
218    +
219    +
220    +
221    +
222    +
223    +
224    +
225    +
226    +
227    +
228    +
229    +
230    +
231    +
232    +
233    +
234    +
235    +
236    +
237    +
238    +
239    +
240    +
241    +
242    +
243    +
244    +
245    +
246    +
247    +
248    +
249    +
250    +
251    +
252    +
253    +
254    +
255    +
256    +
257    +
258    +
259    +
260    +
261    +
262    +
263    +
264    +
265    +
266    +
267    +
268    +
269    +
270    +
271    +
272    +
273    +
274    +
275    +
276    +
277    +
278    +
279    +
280    +
281    +
282    +
283    +
284    +
285    +
286    +
287    +
288    +
289    +
290    +
291    +
292    +
293    +
294    +
295    +
296    +
297    +
298    +
299    +
300    +
301    +
302    +
303    +
304    +
305    +
306    +
307    +
308    +
309    +
310    +
311    +
312    +
313    +
314    +
315    +
316    +
317    +
318    +
319    +
320    +
321    +
322    +
323    +
324    +
325    +
326    +
327    +
328    +
329    +
330    +
331    +
332    +
333    +
334    +
335    +
336    +
337    +
338    +
339    +
340    +
341    +
342    +
343    +
344    +
345    +
346    +
347    +
348    +
349    +
350    +
351    +
352    +
353    +
354    +
355    +
356    +
357    +
358    +
359    +
360    +
361    +
362    +
363    +
364    +
365    +
366    +
367    +
368    +
369    +
370    +
371    +
372    +
373    +
374    +
375    +
376    +
377    +
378    +
379    +
380    +
381    +
382    +
383    +
384    +
385    +
386    +
387    +
388    +
389    +
390    +
391    +
392    +
393    +
394    +
395    +
396    +
397    +
398    +
399    +
400    +
401    +
402    +
403    +
404    +
405    +
406    +
407    +
408    +
409    +
410    +
411    +
412    +
413    +
414    +
415    +
416    +
417    +
418    +
419    +
420    +
421    +
422    +
423    +
424    +
425    +
426    +
427    +
428    +
429    +
430    +
431    +
432    +
433    +
434    +
435    +
436    +
437    +
438    +
439    +
440    +
441    +
442    +
443    +
444    +
445    +
446    +
447    +
448    +
449    +
450    +
451    +
452    +
453    +
454    +
455    +
456    +
457    +
458    +
459    +
460    +
461    +
462    +
463    +
464    +
465    +
466    +
467    +
468    +
469    +
470    +
471    +
472    +
473    +
474    +
475    +
476    +
477    +
478    +
479    +
480    +
481    +
482    +
483    +
484    +
485    +
486    +
487    +
488    +
489    +
490    +
491    +
492    +
493    +
494    +
495    +
496    +
497    +
498    +
499    +
500    +
501    +
502    +
503    +
504    +
505    +
506    +
507    +
508    +
509    +
510    +
511    +
512    +
513    +
514    +
515    +
516    +
517    +
518    +
519    +
520    +
521    +
522    +
523    +
524    +
525    +
526    +
527    +
528    +
529    +
530    +
531    +
532    +
533    +
534    +
535    +
536    +
537    +
538    +
539    +
540    +
541    +
542    +
543    +
544    +
545    +
546    +
547    +
548    +
549    +
550    +
551    +
552    +
553    +
554    +
555    +
556    +
557    +
558    +
559    +
560    +
561    +
562    +
563    +
564    +
565    +
566    +
567    +
568    +
569    +
570    +
571    +
572    +
573    +
574    +
575    +
576    +
577    +
578    +
579    +
580    +
581    +
582    +
583    +
584    +
585    +
586    +
587    +
588    +
589    +
590    +
591    +
592    +
593    +
594    +
595    +
596    +
597    +
598    +
599    +
600    +
601    +
602    +
603    +
604    +
605    +
606    +
607    +
608    +
609    +
610    +
611    +
612    +
613    +
614    +
615    +
616    +
617    +
618    +
619    +
620    +
621    +
622    +
623    +
624    +
625    +
626    +
627    +
628    +
629    +
630    +
631    +
632    +
633    +
634    +
635    +
636    +
637    +
638    +
639    +
640    +
641    +
642    +
643    +
644    +
645    +
646    +
647    +
648    +
649    +
650    +
651    +
652    +
653    +
654    +
655    +
656    +
657    +
658    +
659    +
660    +
661    +
662    +
663    +
664    +
665    +
666    +
667    +
668    +
669    +
670    +
671    +
672    +
673    +
674    +
675    +
676    +
677    +
678    +
679    +
680    +
681    +
682    +
683    +
684    +
685    +
686    +
687    +
688    +
689    +
690    +
691    +
692    +
693    +
694    +
695    +
696    +
697    +
698    +
699    +
700    +
701    +
702    +
703    +
704    +
705    +
706    +
707    +
708    +
709    +
710    +
711    +
712    +
713    +
714    +
715    +
716    +
717    +
718    +
719    +
720    +
721    +
722    +
723    +
724    +
725    +
726    +
727    +
728    +
729    +
730    +
731    +
732    +
733    +
734    +
735    +
736    +
737    +
738    +
739    +
740    +
741    +
742    +
743    +
744    +
745    +
746    +
747    +
748    +
749    +
750    +
751    +
752    +
753    +
754    +
755    +
756    +
757    +
758    +
759    +
760    +
761    +
762    +
763    +
764    +
765    +
766    +
767    +
768    +
769    +
770    +
771    +
772    +
773    +
774    +
775    +
776    +
777    +
778    +
779    +
780    +
781    +
782    +
783    +
784    +
785    +
786    +
787    +
788    +
789    +
790    +
791    +
792    +
793    +
794    +
795    +
796    +
797    +
798    +
799    +
800    +
801    +
802    +
803    +
804    +
805    +
806    +
807    +
808    +
809    +
810    +
811    +
812    +
813    +
814    +
815    +
816    +
817    +
818    +
819    +
820    +
821    +
822    +
823    +
824    +
825    +
826    +
827    +
828    +
829    +
830    +
831    +
832    +
833    +
834    +
835    +
836    +
837    +
838    +
839    +
840    +
841    +
842    +
843    +
844    +
845    +
846    +
847    +
848    +
849    +
850    +
851    +
852    +
853    +
854    +
855    +
856    +
857    +
858    +
859    +
860    +
861    +
862    +
863    +
864    +
865    +
866    +
867    +
868    +
869    +
870    +
871    +
872    +
873    +
874    +
875    +
876    +
877    +
878    +
879    +
880    +
881    +
882    +
883    +
884    +
885    +
886    +
887    +
888    +
889    +
890    +
891    +
892    +
893    +
894    +
895    +
896    +
897    +
898    +
899    +
900    +
901    +
902    +
903    +
904    +
905    +
906    +
907    +
908    +
909    +
910    +
911    +
912    +
913    +
914    +
915    +
916    +
917    +
918    +
919    +
920    +
921    +
922    +
923    +
924    +
925    +
926    +
927    +
928    +
929    +
930    +
931    +
932    +
933    +
934    +
935    +
936    +
937    +
938    +
939    +
940    +
941    +
942    +
943    +
944    +
945    +
946    +
947    +
948    +
949    +
950    +
951    +
952    +
953    +
954    +
955    +
956    +
957    +
958    +
959    +
960    +
961    +
962    +
963    +
964    +
965    +
966    +
967    +
968    +
969    +
970    +
971    +
972    +
973    +
974    +
975    +
976    +
977    +
978    +
979    +
980    +
981    +
982    +
983    +
984    +
985    +
986    +
987    +
988    +
989    +
990    +
991    +
992    +
993    +
994    +
995    +
996    +
997    +
998    +
999    +
1000   +
1001   +
1002   +
1003   +
1004   +
1005   +
1006   +
1007   +
1008   +
1009   +
1010   +
1011   +
1012   +
1013   +
1014   +
1015   +
1016   +
1017   +
1018   +
1019   +
1020   +
1021   +
1022   +
1023   +
1024   +
1025   +
1026   +
1027   +
1028   +
1029   +
1030   +
1031   +
1032   +
1033   +
1034   +
1035   +
1036   +
1037   +
1038   +
1039   +
1040   +
1041   +
1042   +
1043   +
1044   +
1045   +
1046   +
1047   +
1048   +
1049   +
1050   +
1051   +
1052   +
1053   +
1054   +
1055   +
1056   +
1057   +
1058   +
1059   +
1060   +
1061   +
1062   +
1063   +
1064   +
1065   +
1066   +
1067   +
1068   +
1069   +
1070   +
1071   +
1072   +
1073   +
1074   +
1075   +
1076   +
1077   +
1078   +
1079   +
1080   +
1081   +
1082   +
1083   +
1084   +
1085   +
1086   +
1087   +
1088   +
1089   +
1090   +
1091   +
1092   +
1093   +
1094   +
1095   +
1096   +
1097   +
1098   +
1099   +
1100   +
1101   +
1102   +
1103   +
1104   +
1105   +
1106   +
1107   +
1108   +
1109   +
1110   +
1111   +
1112   +
1113   +
1114   +
1115   +
1116   +
1117   +
1118   +
1119   +
1120   +
1121   +
1122   +
1123   +
1124   +
1125   +
1126   +
1127   +
1128   +
1129   +
1130   +
1131   +
1132   +
1133   +
1134   +
1135   +
1136   +
1137   +
1138   +
1139   +
1140   +
1141   +
1142   +
1143   +
1144   +
1145   +
1146   +
1147   +
1148   +
1149   +
1150   +
1151   +
1152   +
1153   +
1154   +
1155   +
1156   +
1157   +
1158   +
1159   +
1160   +
1161   +
1162   +
1163   +
1164   +
1165   +
1166   +
1167   +
1168   +
1169   +
1170   +
1171   +
1172   +
1173   +
1174   +
1175   +
1176   +
1177   +
1178   +
1179   +
1180   +
1181   +
1182   +
1183   +
1184   +
1185   +
1186   +
1187   +
1188   +
1189   +
1190   +
1191   +
1192   +
1193   +
1194   +
1195   +
1196   +
1197   +
1198   +
1199   +
1200   +
1201   +
1202   +
1203   +
1204   +
1205   +
1206   +
1207   +
1208   +
1209   +
1210   +
1211   +
1212   +
1213   +
1214   +
1215   +
1216   +
1217   +
1218   +
1219   +
1220   +
1221   +
1222   +
1223   +
1224   +
1225   +
1226   +
1227   +
1228   +
1229   +
1230   +
1231   +
1232   +
1233   +
1234   +
1235   +
1236   +
1237   +
1238   +
1239   +
1240   +
1241   +
1242   +
1243   +
1244   +
1245   +
1246   +
1247   +
1248   +
1249   +
1250   +
1251   +
1252   +
1253   +
1254   +
1255   +
1256   +
1257   +
1258   +
1259   +
1260   +
1261   +
1262   +
1263   +
1264   +
1265   +
1266   +
1267   +
1268   +
1269   +
1270   +
1271   +
1272   +
1273   +
1274   +
1275   +
1276   +
1277   +
1278   +
1279   +
1280   +
1281   +
1282   +
1283   +
1284   +
1285   +
1286   +
1287   +
1288   +
1289   +
1290   +
1291   +
1292   +
1293   +
1294   +
1295   +
1296   +
1297   +
1298   +
1299   +
1300   +
1301   +
1302   +
1303   +
1304   +
1305   +
1306   +
1307   +
1308   +
1309   +
1310   +
1311   +
1312   +
1313   +
1314   +
1315   +
1316   +
1317   +
1318   +
1319   +
1320   +
1321   +
1322   +
1323   +
1324   +
1325   +
1326   +
1327   +
1328   +
1329   +
1330   +
1331   +
1332   +
1333   +
1334   +
1335   +
1336   +
1337   +
1338   +
1339   +
1340   +
1341   +
1342   +
1343   +
1344   +
1345   +
1346   +
1347   +
1348   +
1349   +
1350   +
1351   +
1352   +
1353   +
1354   +
1355   +
1356   +
1357   +
1358   +
1359   +
1360   +
1361   +
1362   +
1363   +
1364   +
1365   +
1366   +
1367   +
1368   +
1369   +
1370   +
1371   +
1372   +
1373   +
1374   +
1375   +
1376   +
1377   +
1378   +
1379   +
1380   +
1381   +
1382   +
1383   +
1384   +
1385   +
1386   +
1387   +
1388   +
1389   +
1390   +
1391   +
1392   +
1393   +
1394   +
1395   +
1396   +
1397   +
1398   +
1399   +
1400   +
1401   +
1402   +
1403   +
1404   +
1405   +
1406   +
1407   +
1408   +
1409   +
1410   +
1411   +
1412   +
1413   +
1414   +
1415   +
1416   +
1417   +
1418   +
1419   +
1420   +
1421   +
1422   +
1423   +
1424   +
1425   +
1426   +
1427   +
1428   +
1429   +
1430   +
1431   +
1432   +
1433   +
1434   +
1435   +
1436   +
1437   +
1438   +
1439   +
1440   +
1441   +
1442   +
1443   +
1444   +
1445   +
1446   +
1447   +
1448   +
1449   +
1450   +
1451   +
1452   +
1453   +
1454   +
1455   +
1456   +
1457   +
1458   +
1459   +
1460   +
1461   +
1462   +
1463   +
1464   +
1465   +
1466   +
1467   +
1468   +
1469   +
1470   +
1471   +
1472   +
1473   +
1474   +
1475   +
1476   +
1477   +
1478   +
1479   +
1480   +
1481   +
1482   +
1483   +
1484   +
1485   +
1486   +
1487   +
1488   +
1489   +
1490   +
1491   +
1492   +
1493   +
1494   +
1495   +
1496   +
1497   +
1498   +
1499   +
1500   +
1501   +
1502   +
1503   +
1504   +
1505   +
1506   +
1507   +
1508   +
1509   +
1510   +
1511   +
1512   +
1513   +
1514   +
1515   +
1516   +
1517   +
1518   +
1519   +
1520   +
1521   +
1522   +
1523   +
1524   +
1525   +
1526   +
1527   +
1528   +
1529   +
1530   +
1531   +
1532   +
1533   +
1534   +
1535   +
1536   +
1537   +
1538   +
1539   +
1540   +
1541   +
1542   +
1543   +
1544   +
1545   +
1546   +
1547   +
1548   +
1549   +
1550   +
1551   +
1552   +
1553   +
1554   +
1555   +
1556   +
1557   +
1558   +
1559   +
1560   +
1561   +
1562   +
1563   +
1564   +
1565   +
1566   +
1567   +
1568   +
1569   +
1570   +
1571   +
1572   +
1573   +
1574   +
1575   +
1576   +
1577   +
1578   +
1579   +
1580   +
1581   +
1582   +
1583   +
1584   +
1585   +
1586   +
1587   +
1588   +
1589   +
1590   +
1591   +
1592   +
1593   +
1594   +
1595   +
1596   +
1597   +
1598   +
1599   +
1600   +
1601   +
1602   +
1603   +
1604   +
1605   +
1606   +
1607   +
1608   +
1609   +
1610   +
1611   +
1612   +
1613   +
1614   +
1615   +
1616   +
1617   +
1618   +
1619   +
1620   +
1621   +
1622   +
1623   +
1624   +
1625   +
1626   +
1627   +
1628   +
1629   +
1630   +
1631   +
1632   +
1633   +
1634   +
1635   +
1636   +
1637   +
1638   +
1639   +
1640   +
1641   +
1642   +
1643   +
1644   +
1645   +
1646   +
1647   +
1648   +
1649   +
1650   +
1651   +
1652   +
1653   +
1654   +
1655   +
1656   +
1657   +
1658   +
1659   +
1660   +
1661   +
1662   +
1663   +
1664   +
1665   +
1666   +
1667   +
1668   +
1669   +
1670   +
1671   +
1672   +
1673   +
1674   +
1675   +
1676   +
1677   +
1678   +
1679   +
1680   +
1681   +
1682   +
1683   +
1684   +
1685   +
1686   +
1687   +
1688   +
1689   +
1690   +
1691   +
1692   +
1693   +
1694   +
1695   +
1696   +
1697   +
1698   +
1699   +
1700   +
1701   +
1702   +
1703   +
1704   +
1705   +
1706   +
1707   +
1708   +
1709   +
1710   +
1711   +
1712   +
1713   +
1714   +
1715   +
1716   +
1717   +
1718   +
1719   +
1720   +
1721   +
1722   +
1723   +
1724   +
1725   +
1726   +
1727   +
1728   +
1729   +
1730   +
1731   +
1732   +
1733   +
1734   +
1735   +
1736   +
1737   +
1738   +
1739   +
1740   +
1741   +
1742   +
1743   +
1744   +
1745   +
1746   +
1747   +
1748   +
1749   +
1750   +
1751   +
1752   +
1753   +
1754   +
1755   +
1756   +
1757   +
1758   +
1759   +
1760   +
1761   +
1762   +
1763   +
1764   +
1765   +
1766   +
1767   +
1768   +
1769   +
1770   +
1771   +
1772   +
1773   +
1774   +
1775   +
1776   +
1777   +
1778   +
1779   +
1780   +
1781   +
1782   +
1783   +
1784   +
1785   +
1786   +
1787   +
1788   +
1789   +
1790   +
1791   +
1792   +
1793   +
1794   +
1795   +
1796   +
1797   +
1798   +
1799   +
1800   +
1801   +
1802   +
1803   +
1804   +
1805   +
1806   +
1807   +
1808   +
1809   +
1810   +
1811   +
1812   +
1813   +
1814   +
1815   +
1816   +
1817   +
1818   +
1819   +
1820   +
1821   +
1822   +
1823   +
1824   +
1825   +
1826   +
1827   +
1828   +
1829   +
1830   +
1831   +
1832   +
1833   +
1834   +
1835   +
1836   +
1837   +
1838   +
1839   +
1840   +
1841   +
1842   +
1843   +
1844   +
1845   +
1846   +
1847   +
1848   +
1849   +
1850   +
1851   +
1852   +
1853   +
1854   +
```

|   |   |   |
|---|---|---|
|  | <b>MINTIC 2022</b><br><b>DESARROLLO DE SOFTWARE</b><br><b>Cargue de Datos archivos CSV con Spring Boot.</b> |  |
|---|---|---|