

Tecnológico de Estudios Superiores de Ecatepec

División de Ingeniería en sistemas
Computacionales

Manual de uso de
Módulo de Excel

Asignatura:

Bases de Datos Para Dispositivos
Móviles

Profesor:

Cortes Barrera Griselda

Presentan Bichos Team

Gutiérrez García Jorge

Villasana Luna Ricardo

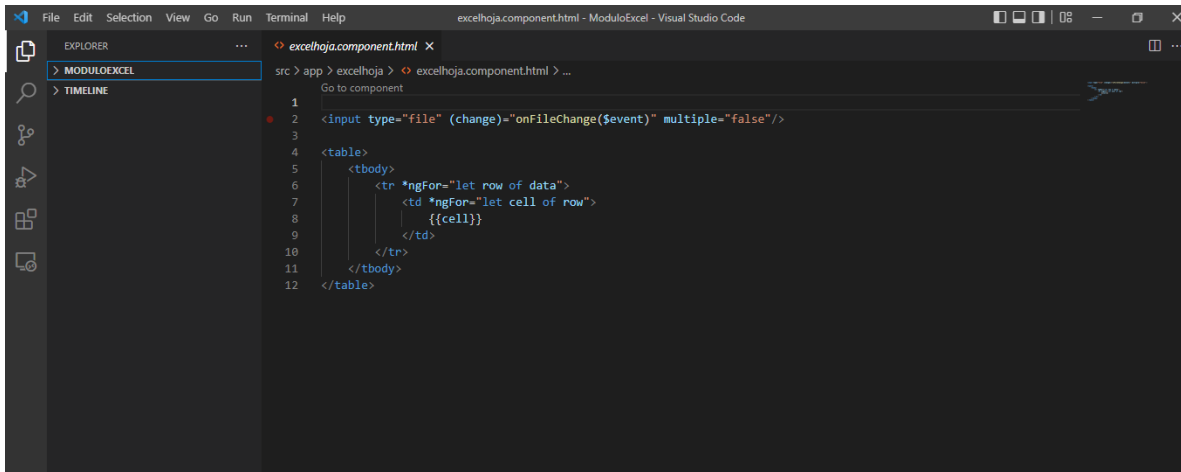
Valdez Montero Sergio Aaron

Grupo:

5801

Manual de uso de módulo de Excel en Angular.

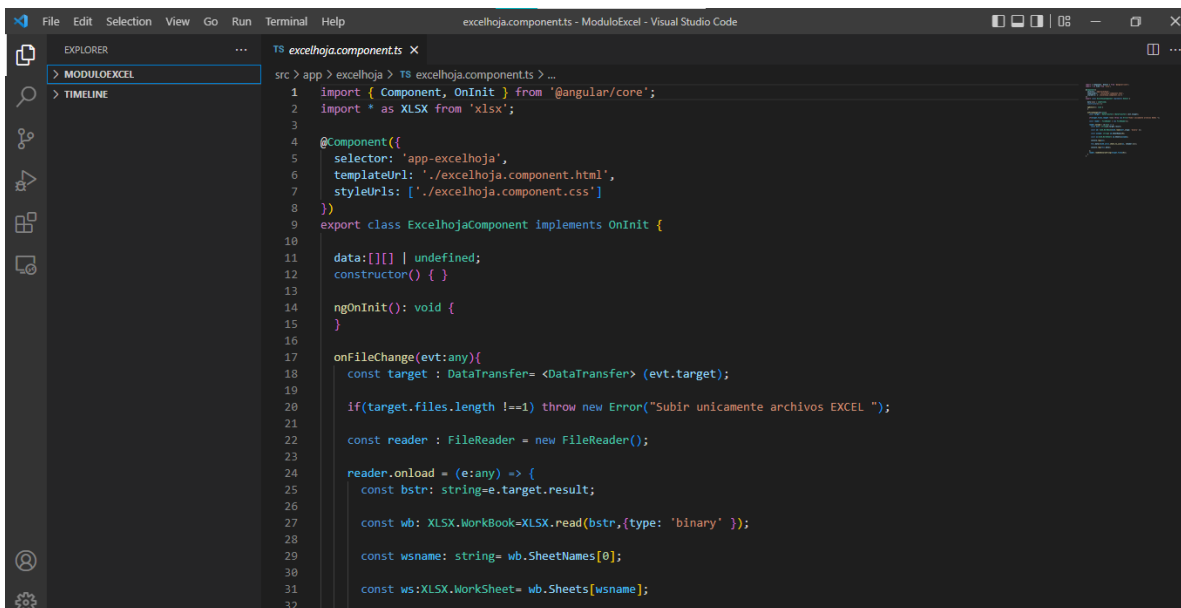
1. Para leer el archivo de Excel, se creo un componente llamado *excelhoja* el cual contiene la parte grafica donde se ingresará el archivo Excel y posteriormente lo mostrará en pantalla.



The screenshot shows the Visual Studio Code editor with the file `excelhoja.component.html` open. The Explorer sidebar on the left shows the project structure with `MODULOEXCEL` and `TIMELINE` folders. The main editor area displays the following HTML code:

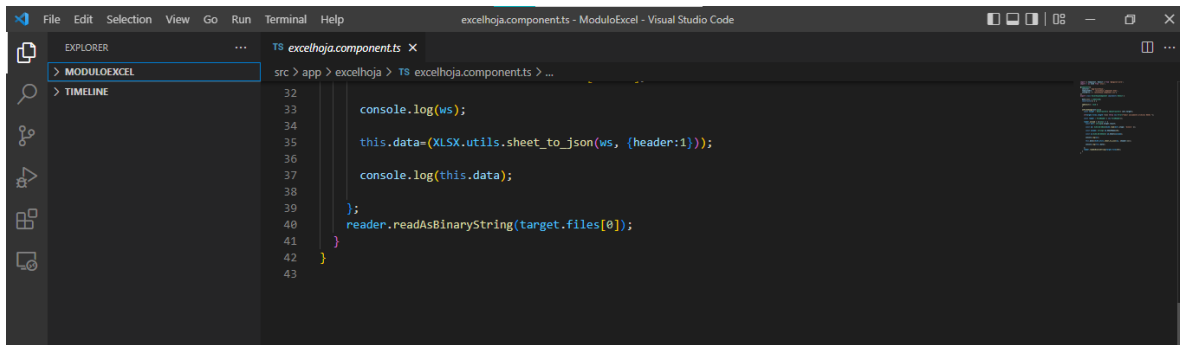
```
1 <input type="file" (change)="onFileChange($event)" multiple="false"/>
2
3
4 <table>
5   <tbody>
6     <tr *ngFor="let row of data">
7       <td *ngFor="let cell of row">
8         {{cell}}
9       </td>
10    </tr>
11  </tbody>
12 </table>
```

2. Como segundo paso es desarrollar la lógica en el *excelhoja.component.ts*, dentro de este archivo es donde crearemos el método el cual leerá el archivo Excel.(NOTA: Para que se pueda leer el archivo Excel se usa la extensión XLSX, la cual solo se puede usar instalando la dependencia la cual es la siguiente: **ng install xlsx**).

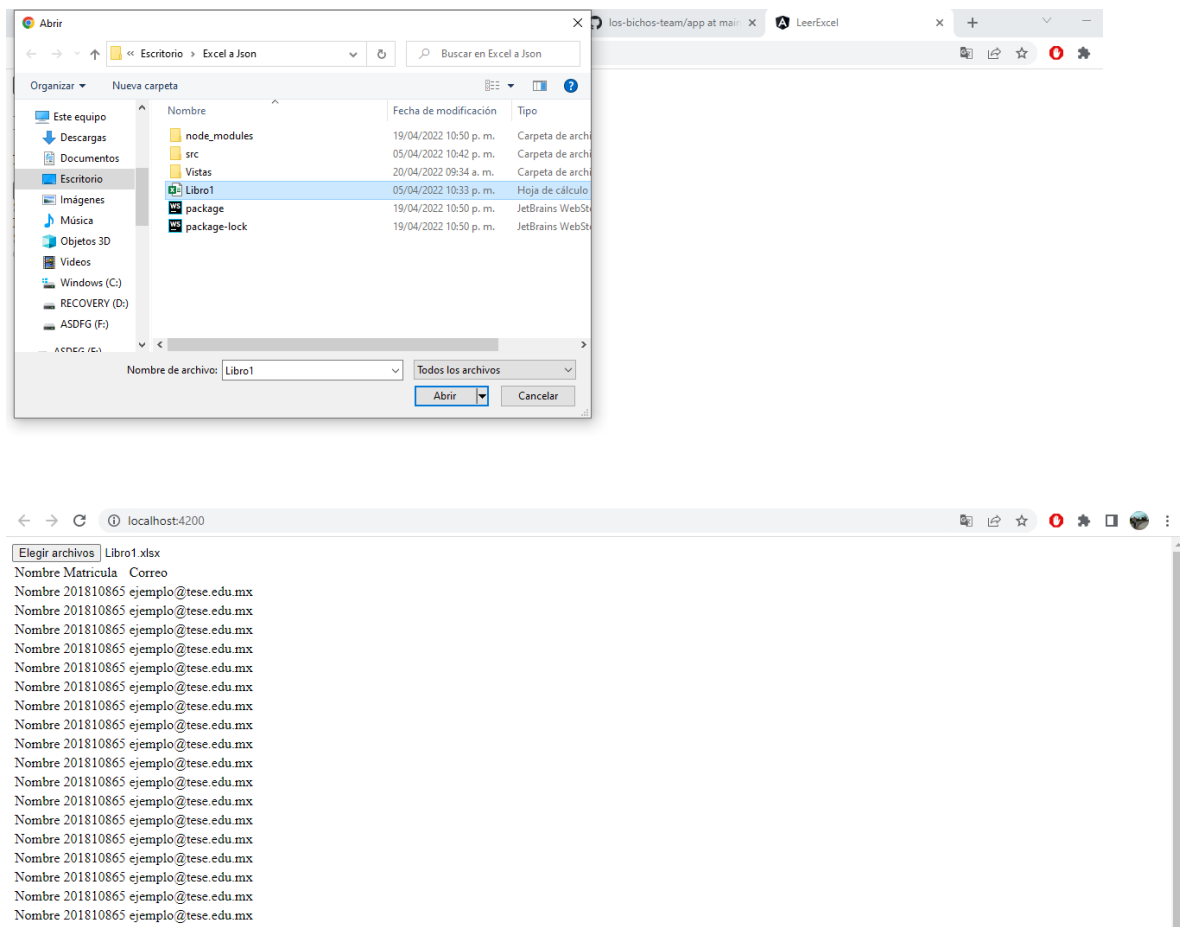


The screenshot shows the Visual Studio Code editor with the file `excelhoja.component.ts` open. The Explorer sidebar on the left shows the project structure with `MODULOEXCEL` and `TIMELINE` folders. The main editor area displays the following TypeScript code:

```
1 import { Component, OnInit } from '@angular/core';
2 import * as XLSX from 'xlsx';
3
4 @Component({
5   selector: 'app-excelhoja',
6   templateUrl: './excelhoja.component.html',
7   styleUrls: ['./excelhoja.component.css']
8 })
9 export class ExcelhojaComponent implements OnInit {
10
11   data:[][] | undefined;
12   constructor() { }
13
14   ngOnInit(): void {
15   }
16
17   onFileChange(evt:any){
18     const target : DataTransfer= <DataTransfer> (evt.target);
19
20     if(target.files.length !==1) throw new Error("Subir unicamente archivos EXCEL ");
21
22     const reader : FileReader = new FileReader();
23
24     reader.onload = (e:any) => {
25       const bstr: string=e.target.result;
26
27       const wb: XLSX.WorkBook=XLSX.read(bstr,{type: 'binary' });
28
29       const wsname= wb.SheetNames[0];
30
31       const ws:XLSX.WorkSheet= wb.Sheets[wsname];
32     }
33   }
```

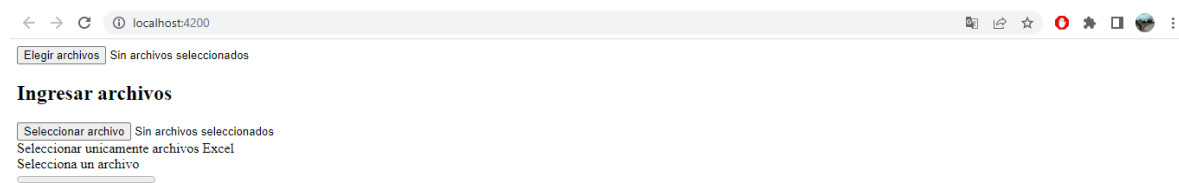


- Una vez ejecutamos en la consola el módulo (con **ng serve -o**) lo que nos mostrara es lo que contiene el archivo *excelhoja.component.html* el cual es un botón para seleccionar nuestro archivo Excel. Una vez hecho eso automáticamente nos leerá el archivo y lo mostrará en pantalla.

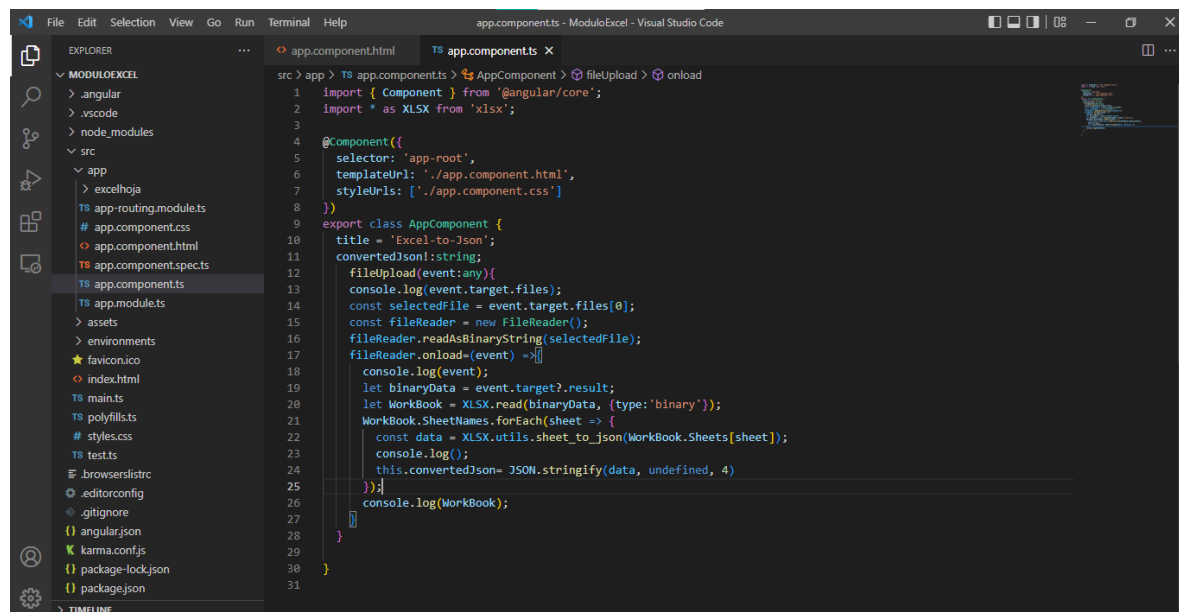


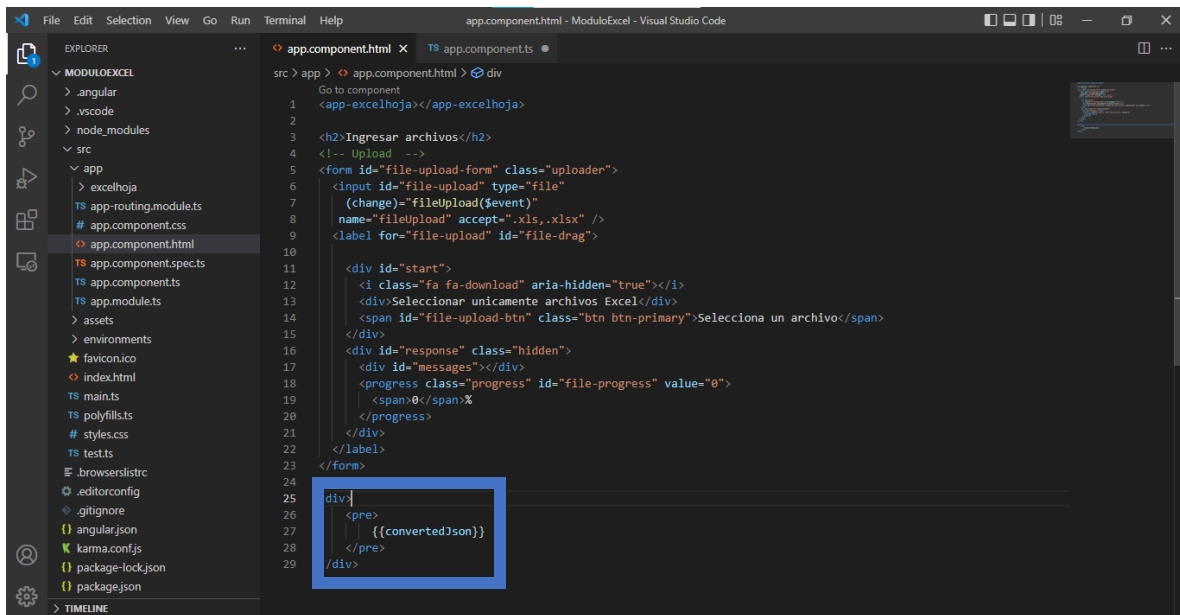
Podemos observar que el contenido de mostrado en pantalla es el mismo que el que esta en nuestro archivo Excel.

De este modo en nuestro modulo tendremos la opción de leer Excel y convertirlo a Json en uno mismo.



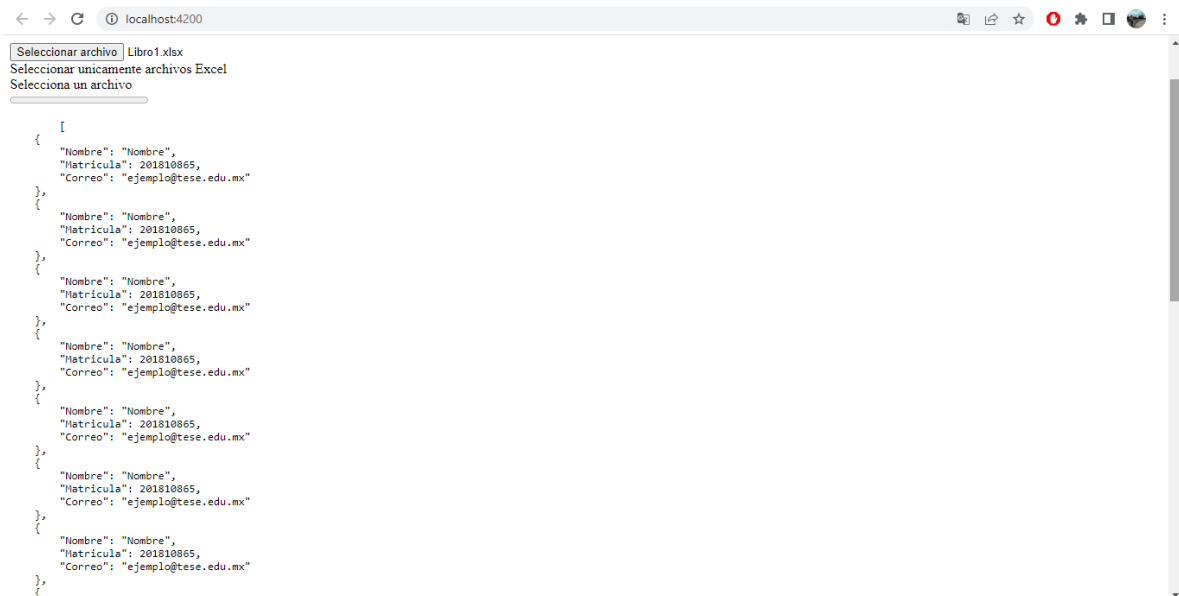
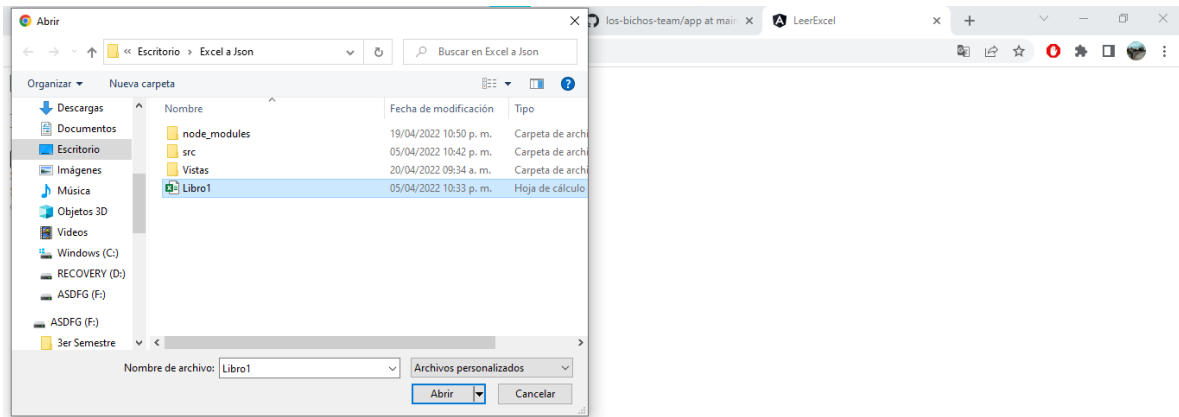
5. Para la lógica de como convertir el archivo Excel a Json usaremos el *app.component.ts* en este apartado esta todo el proceso de leer la hoja de Excel así como la conversión a Json, únicamente hay una variación en donde usamos la variable creada llamada **convertedJson** la cual la desarrollamos y mandamos a llamar en el *app.component.html* para su correcto uso.





```
1 <app-excelhoja></app-excelhoja>
2
3 <h2>Ingresar archivos</h2>
4 <!-- Upload -->
5 <form id="file-upload-form" class="uploader">
6   <input id="file-upload" type="file"
7     (change)="fileUpload($event)"
8     name="fileUpload" accept=".xls,.xlsx" />
9   <label for="file-upload" id="file-drag">
10
11     <div id="start">
12       <i class="fa fa-download" aria-hidden="true"></i>
13       <div>Seleccionar unicamente archivos Excel</div>
14       <span id="file-upload-btn" class="btn btn-primary">Selecciona un archivo</span>
15     </div>
16     <div id="response" class="hidden">
17       <div id="messages"></div>
18       <progress class="progress" id="file-progress" value="0">
19         <span>0</span></span>
20     </div>
21   </div>
22 </label>
23 </form>
24
25 <div>
26   <pre>
27     {{convertedJson}}
28   </pre>
29 </div>
```

6. Como paso final, ejecutamos el proyecto y nos aparecera la opcion de leer excel o convertir excel a Json. En este caso usaremos la de convertir excel a Json y una vez ingresando el archivo automaticamente nos convertira el archivo excel a Json.



Podemos comprobar que el archivo excel contiene el mismo contenido que lo que se convirtió en Json. Ya que usamos como ejemplo el mismo archivo que leyo. Pero se puede usar cualquier archivo que sea excel.