



TECNOLÓGICO DE ESTUDIOS SUPERIORES DE ECATEPEC

DIVISIÓN DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

PERIODO 2022-1

ASIGNATURA:

(OPT_2_RSD) BASES DE DATOS PARA DISPOSITIVOS MÓVILES

PROFESOR:

CORTES BARRERA GRISELDA

GRUPO: 5801

Manual Reportes en Excel.

EQUIPO: MASTERS MOBILE

Aviña Velarde Daniela Michelle

Hernández Pacheco Andrea Alin

Reyes Mitznahuatl Brandon Jesús

Reyes Reyes Daniela Michelle

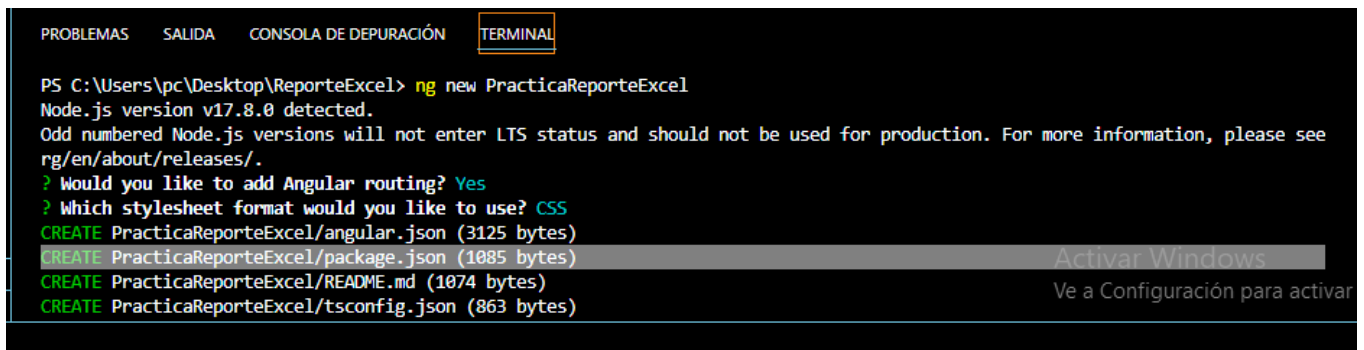
Villagrana Aparicio Brandon Jair

FECHA DE ENTREGA:

20-04-2022

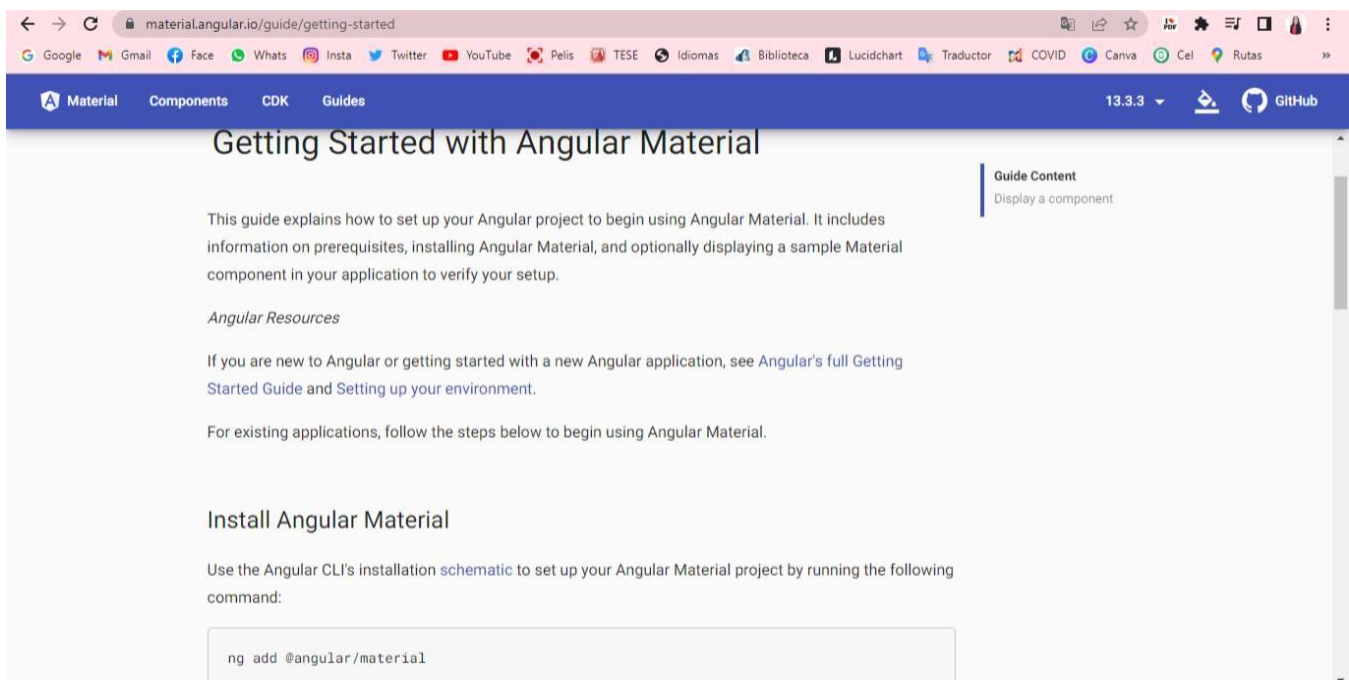
Manual para la construcción de una plantilla para obtener reportes en archivos de Excel.

Paso 1: El primer paso consiste en crear un nuevo proyecto de Angular. Para esto se debe abrir la consola, ingresar en el directorio que se desea colocar el proyecto y crearlo con el comando `ng new` seguido del nombre del proyecto.



```
PS C:\Users\pc\Desktop\ReporteExcel> ng new PracticaReporteExcel
Node.js version v17.8.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please see
ng/en/about/releases/.
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE PracticaReporteExcel/angular.json (3125 bytes)
CREATE PracticaReporteExcel/package.json (1085 bytes)
CREATE PracticaReporteExcel/README.md (1074 bytes)
CREATE PracticaReporteExcel/tsconfig.json (863 bytes)
```

Paso 2: El segundo paso es añadir el comando `ng add @angular/material` que se encuentra dentro de la página web de [angular.io](https://material.angular.io/guide/getting-started) para poder utilizar sus funciones más adelante.



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

PS C:\Users\pc\Desktop\ReporteExcel\PracticaReporteExcel>
PS C:\Users\pc\Desktop\ReporteExcel\PracticaReporteExcel> ng add @angular/material
Node.js version v17.8.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please see
https://nodejs.org/en/about/releases/.
i Using package manager: npm
✓ Found compatible package version: @angular/material@13.3.3.
✓ Package information loaded.

The package @angular/material@13.3.3 will be installed and executed.
Would you like to proceed? Yes
✓ Package successfully installed.
? Choose a prebuilt theme name, or "custom" for a custom theme: Indigo/Pink [ Preview: https://material.angular.io?theme=indigo-pink ]
? Set up global Angular Material typography styles? Yes
? Set up browser animations for Angular Material? Yes
UPDATE package.json (1151 bytes)
✓ Packages installed successfully.
UPDATE src/app/app.module.ts (502 bytes)
UPDATE angular.json (3289 bytes)
UPDATE src/index.html (588 bytes)
UPDATE src/styles.css (181 bytes)
PS C:\Users\pc\Desktop\ReporteExcel\PracticaReporteExcel>

Activar Windows
Ve a Configuración para activar Windows.
```

Paso 3: El tercer paso consiste en crear un nuevo componente para la tabla en la que se van a almacenar datos, este paso se realiza con el comando `ng g c` seguido del nombre del módulo y el nombre del componente.

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  Terminal  Ayuda  ReporteExcel - Visual Studio Code

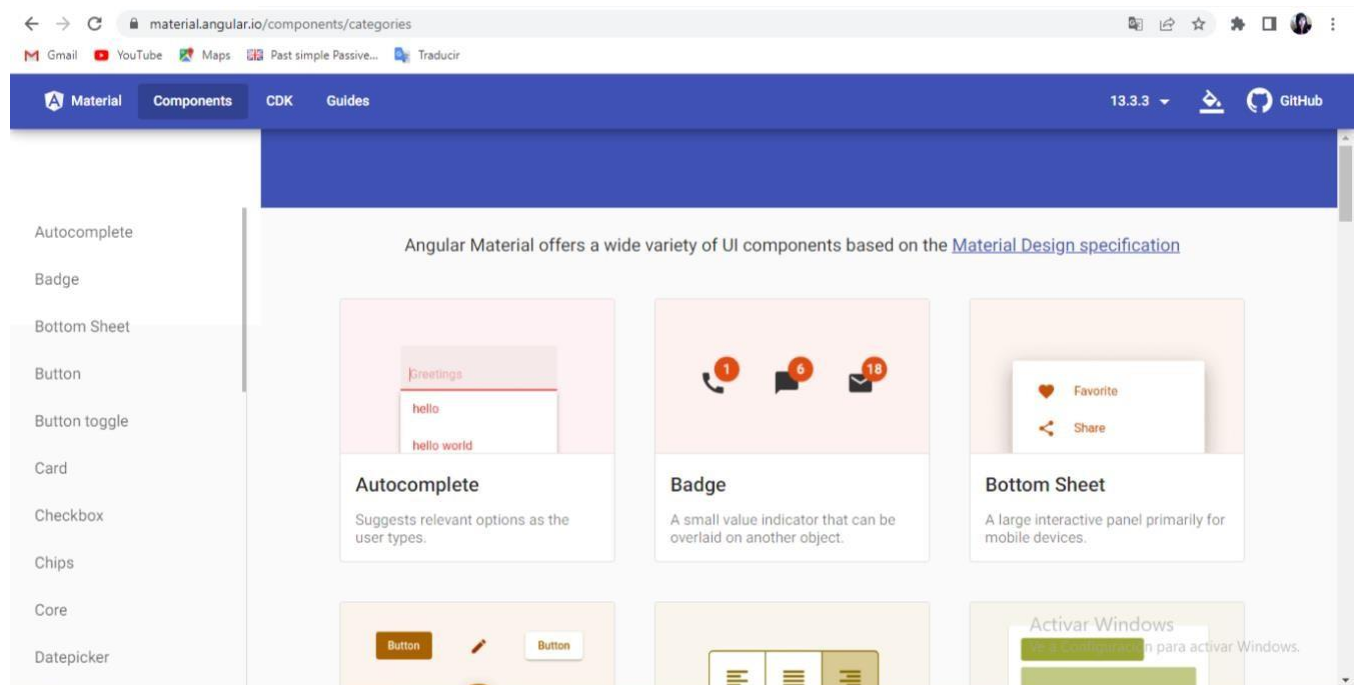
EXPLORADOR
└─ REPORTEEXCEL
  └─ PracticaReporteExcel
    ├── .vscode
    ├── node_modules
    └── src
      └── app
        ├── components
        │   └── table
        │       ├── table.component.css
        │       ├── table.component.html
        │       ├── table.component.spec.ts
        │       └── table.component.ts
        ├── app-routing.module.ts
        ├── app.component.css
        ├── app.component.html
        ├── app.component.spec.ts
        ├── app.component.ts
        ├── app.module.ts
        ├── assets
        ├── environments
        ├── favicon.ico
        ├── index.html
        ├── main.ts
        ├── polyfills.ts
        ├── styles.css
        ├── test.ts
        ├── .browserslistrc
        └── .editorconfig
      └── ESQUEMA
        └── LÍNEA DE TIEMPO

PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

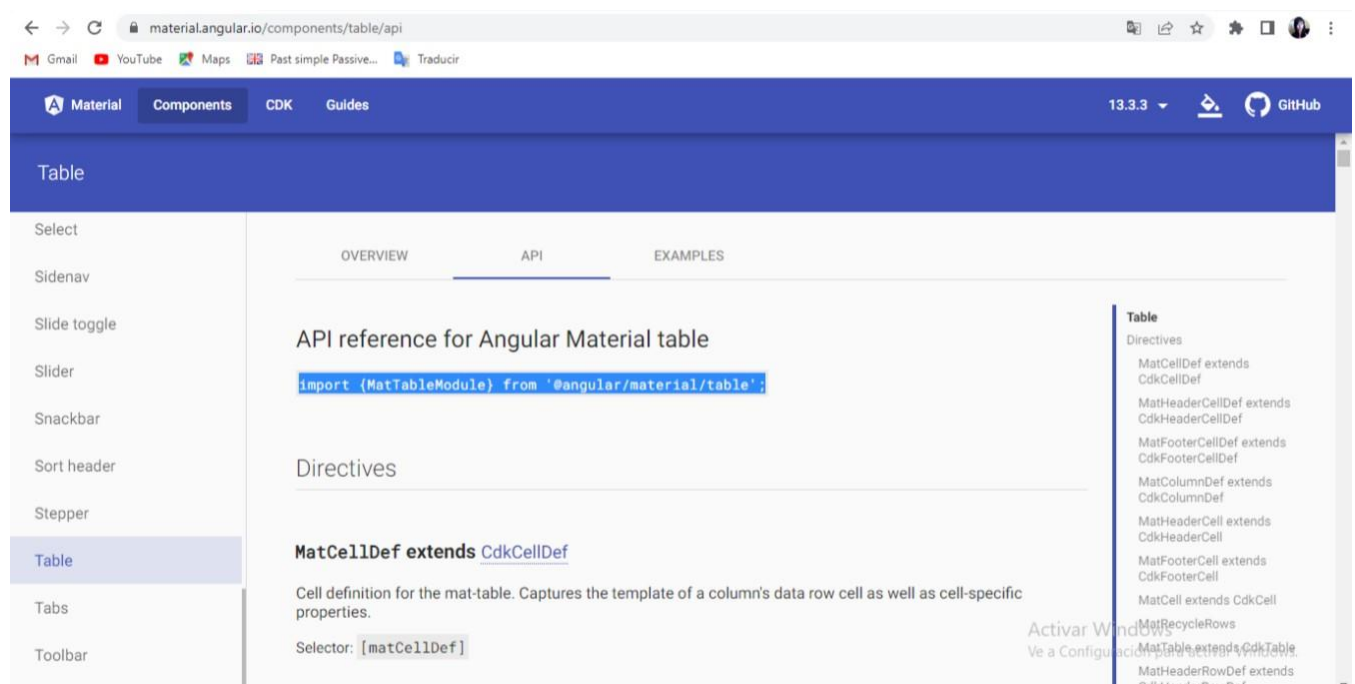
UPDATE src/styles.css (181 bytes)
PS C:\Users\pc\Desktop\ReporteExcel\PracticaReporteExcel> ng g c components/table
Node.js version v17.8.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please see
https://nodejs.org/en/about/releases/.
CREATE src/app/components/table/table.component.html (20 bytes)
CREATE src/app/components/table/table.component.spec.ts (619 bytes)
CREATE src/app/components/table/table.component.ts (271 bytes)
CREATE src/app/components/table/table.component.css (0 bytes)
UPDATE src/app/app.module.ts (591 bytes)
PS C:\Users\pc\Desktop\ReporteExcel\PracticaReporteExcel>

Activar Windows
Ve a Configuración para activar Windows.
```

Paso 4: El cuarto paso es ingresar a la documentación de angular material en la página <https://material.angular.io/components/categories> para después ir hacia los componentes.



Paso 5: El quinto paso es ir a los comandos y dentro de los comandos se selecciona la tabla que se desea agregar y en la opción de API se debe copiar el código que es la referencia para utilizar ese material.



Paso 6: El sexto paso es dirigirse al proyecto de Visual Studio Code y abrir el archivo que lleva por nombre app.module.ts y en la parte de import se pega el código previamente copiado.

```
import { MatTableModule } from '@angular/material/table';
```

```
TS app.module.ts 1 ●
PracticaReporteExcel > src > app > TS app.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { BrowserModule } from '@angular/platform-browser';
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
7  import { TableComponent } from './components/table/table.component';
8  import { MatTableModule } from '@angular/material/table';
9
```

Paso 7: El séptimo paso es agregar el módulo que se importó en el paso anterior en la parte de imports de todos los módulos.

MatTableModule

```
15  imports: [
16    BrowserModule,
17    AppRoutingModule,
18    BrowserAnimationsModule,
19    MatTableModule
20  ],
```

Paso 8: El octavo paso consiste en instalar la dependencia del material de tabla que se va a ocupar para generar archivos de Excel, esta dependencia se encuentra como Angular Material Table Exporter y en la terminal se pone el comando para instalarla.

```
npm install --save mat-table-exporter
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL
PS C:\Users\pc\Desktop\ReporteExcel\PracticaReporteExcel> npm install --save mat-table-exporter
added 15 packages, and audited 939 packages in 2m
100 packages are looking for funding
  run `npm fund` for details
1 high severity vulnerability
Activar Windows
Ve a Configuración para activar Windows
Lín. 21, col. 17  Espacios: 2  UTF-8  LF  {} Ty
```

Paso 9: El noveno paso es crear una interfaz para los modelos, para esto se debe crear un módulo llamado models y un componente llamado tables.

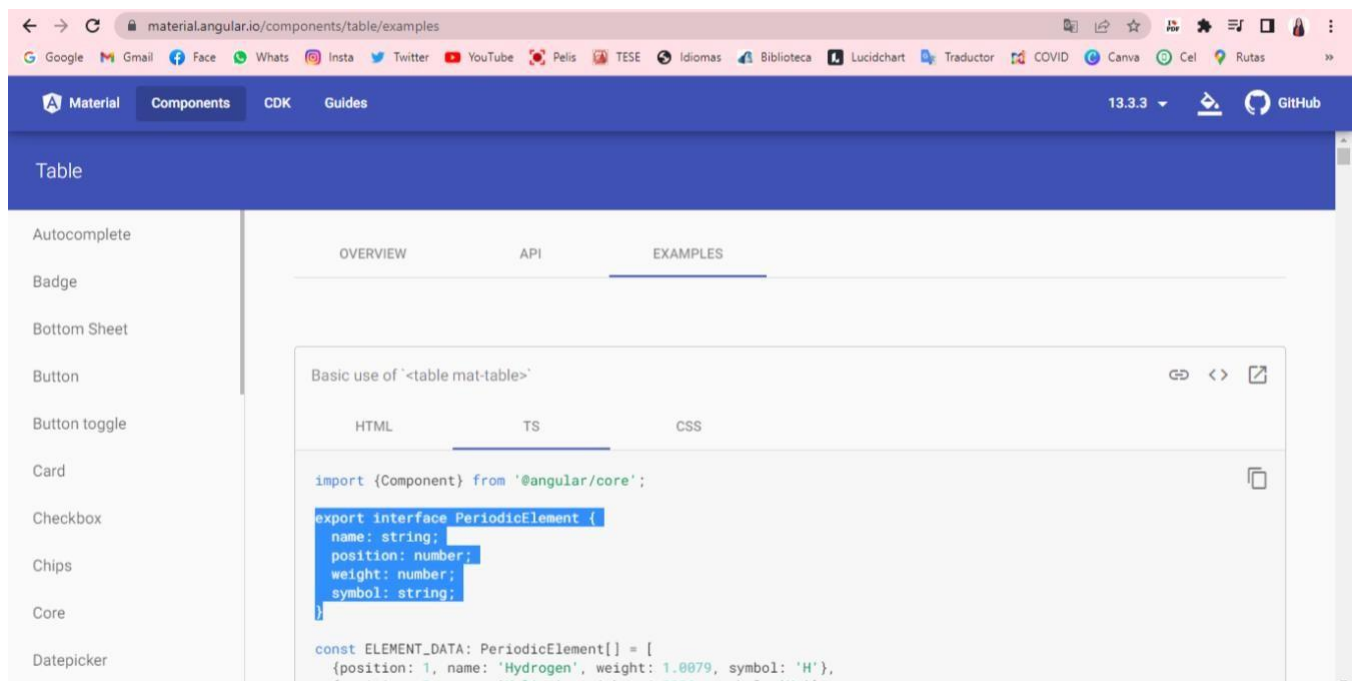
```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL

npm audit fix

Run `npm audit` for details.
PS C:\Users\pc\Desktop\ReporteExcel\PracticaReporteExcel> ng g i models/table
Node.js version v17.8.0 detected.
Odd numbered Node.js versions will not enter LTS status and should not be used for production. For more information, please see
e https://nodejs.org/en/about/releases/.
CREATE src/app/models/table.ts (27 bytes)
PS C:\Users\pc\Desktop\ReporteExcel\PracticaReporteExcel>

Lín. 1, col. 1  Espacios: 4  UTF-8  LF  { } 1
```

Paso 10: El décimo paso consiste en ir a la documentación de angular material y en la opción de ts (typescript) elegir la interfaz a utilizar y copiarla, para después pegarla en el archivo table.ts.

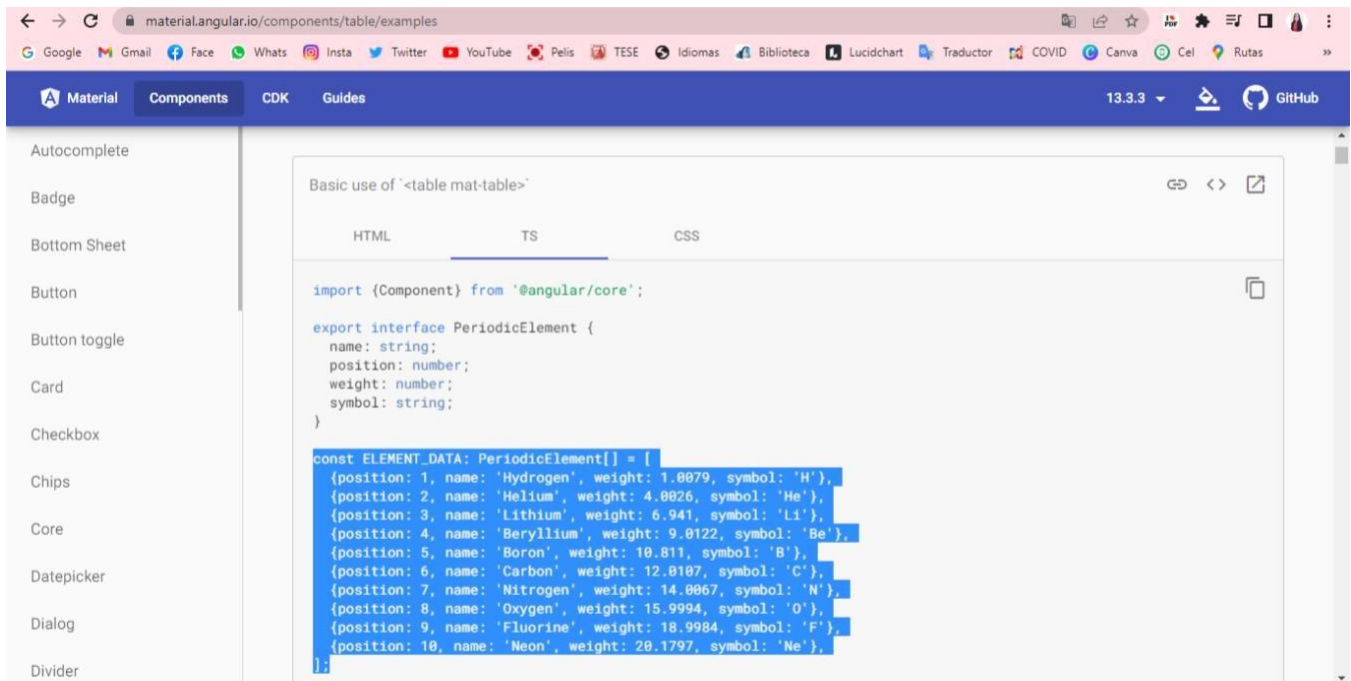


```
app.module.ts  TS table.ts  TS table.component.ts

PracticaReporteExcel > src > app > models > TS table.ts > ...

1  export interface PeriodicElement {
2      name: string;
3      position: number;
4      weight: number;
5      symbol: string;
6  }
7
```


Paso 11: El paso número once es copiar la constante de la tabla y después dirigirse al archivo table.component.ts y después del export se pega este código que es un arreglo para la tabla.

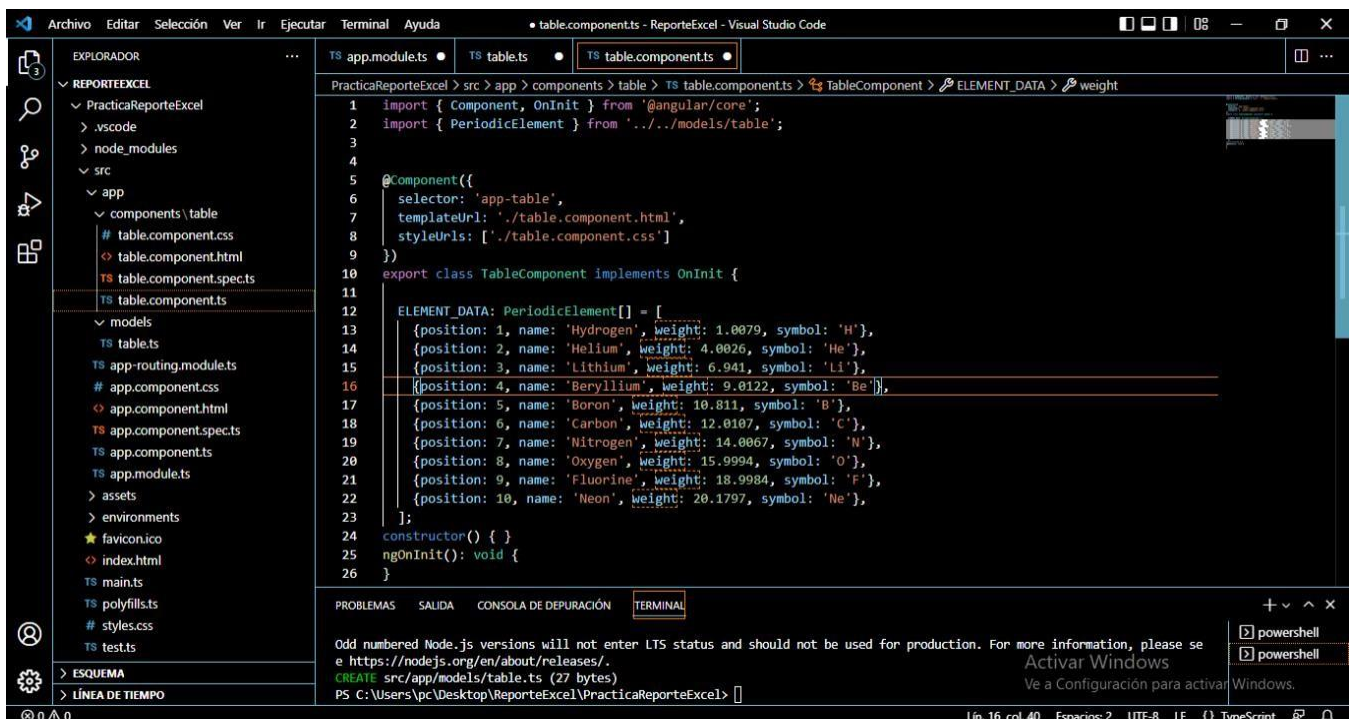


The screenshot shows the Material Angular website at material.angular.io/components/table/examples. The 'Components' tab is selected, and the 'Table' component is highlighted. The 'Basic use of <table mat-table>' example is shown, with the 'TS' tab selected. The code in the 'TS' tab defines a `PeriodicElement` interface and a `ELEMENT_DATA` array of objects representing periodic elements. The array is highlighted with a blue selection box.

```
import {Component} from '@angular/core';

export interface PeriodicElement {
  name: string;
  position: number;
  weight: number;
  symbol: string;
}

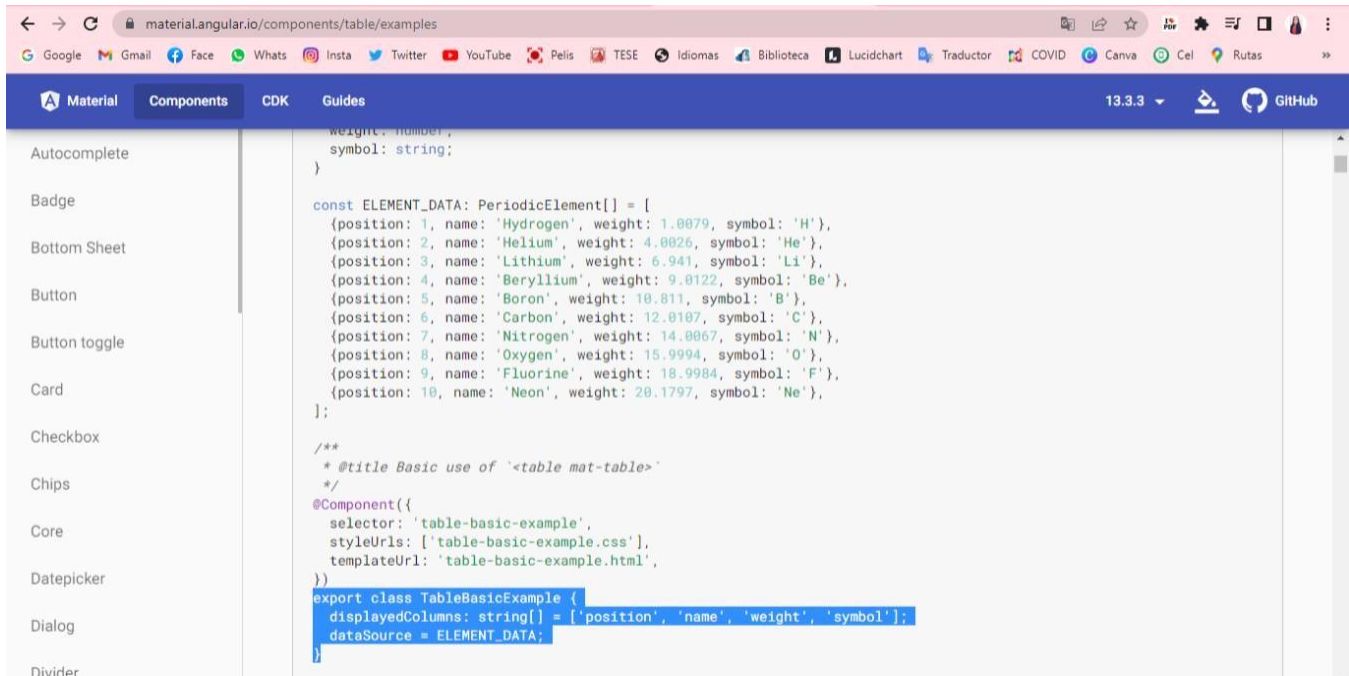
const ELEMENT_DATA: PeriodicElement[] = [
  {position: 1, name: 'Hydrogen', weight: 1.0079, symbol: 'H'},
  {position: 2, name: 'Helium', weight: 4.0026, symbol: 'He'},
  {position: 3, name: 'Lithium', weight: 6.941, symbol: 'Li'},
  {position: 4, name: 'Beryllium', weight: 9.0122, symbol: 'Be'},
  {position: 5, name: 'Boron', weight: 10.811, symbol: 'B'},
  {position: 6, name: 'Carbon', weight: 12.0107, symbol: 'C'},
  {position: 7, name: 'Nitrogen', weight: 14.0067, symbol: 'N'},
  {position: 8, name: 'Oxygen', weight: 15.9994, symbol: 'O'},
  {position: 9, name: 'Fluorine', weight: 18.9984, symbol: 'F'},
  {position: 10, name: 'Neon', weight: 20.1797, symbol: 'Ne'},
];
```



The screenshot shows the Visual Studio Code editor with the `table.component.ts` file open. The file is located in the `src/app/components/table` directory. The code defines a `TableComponent` class that implements `OnInit`. The `ELEMENT_DATA` array is defined in the `TableComponent` class, matching the array from the previous screenshot. The `constructor` and `ngOnInit` methods are also shown.

```
1 import { Component, OnInit } from '@angular/core';
2 import { PeriodicElement } from '../models/table';
3
4
5 @Component({
6   selector: 'app-table',
7   templateUrl: './table.component.html',
8   styleUrls: ['./table.component.css']
9 })
10 export class TableComponent implements OnInit {
11
12   ELEMENT_DATA: PeriodicElement[] = [
13     {position: 1, name: 'Hydrogen', weight: 1.0079, symbol: 'H'},
14     {position: 2, name: 'Helium', weight: 4.0026, symbol: 'He'},
15     {position: 3, name: 'Lithium', weight: 6.941, symbol: 'Li'},
16     {position: 4, name: 'Beryllium', weight: 9.0122, symbol: 'Be'},
17     {position: 5, name: 'Boron', weight: 10.811, symbol: 'B'},
18     {position: 6, name: 'Carbon', weight: 12.0107, symbol: 'C'},
19     {position: 7, name: 'Nitrogen', weight: 14.0067, symbol: 'N'},
20     {position: 8, name: 'Oxygen', weight: 15.9994, symbol: 'O'},
21     {position: 9, name: 'Fluorine', weight: 18.9984, symbol: 'F'},
22     {position: 10, name: 'Neon', weight: 20.1797, symbol: 'Ne'},
23   ];
24   constructor() { }
25   ngOnInit(): void {
26   }
27 }
```

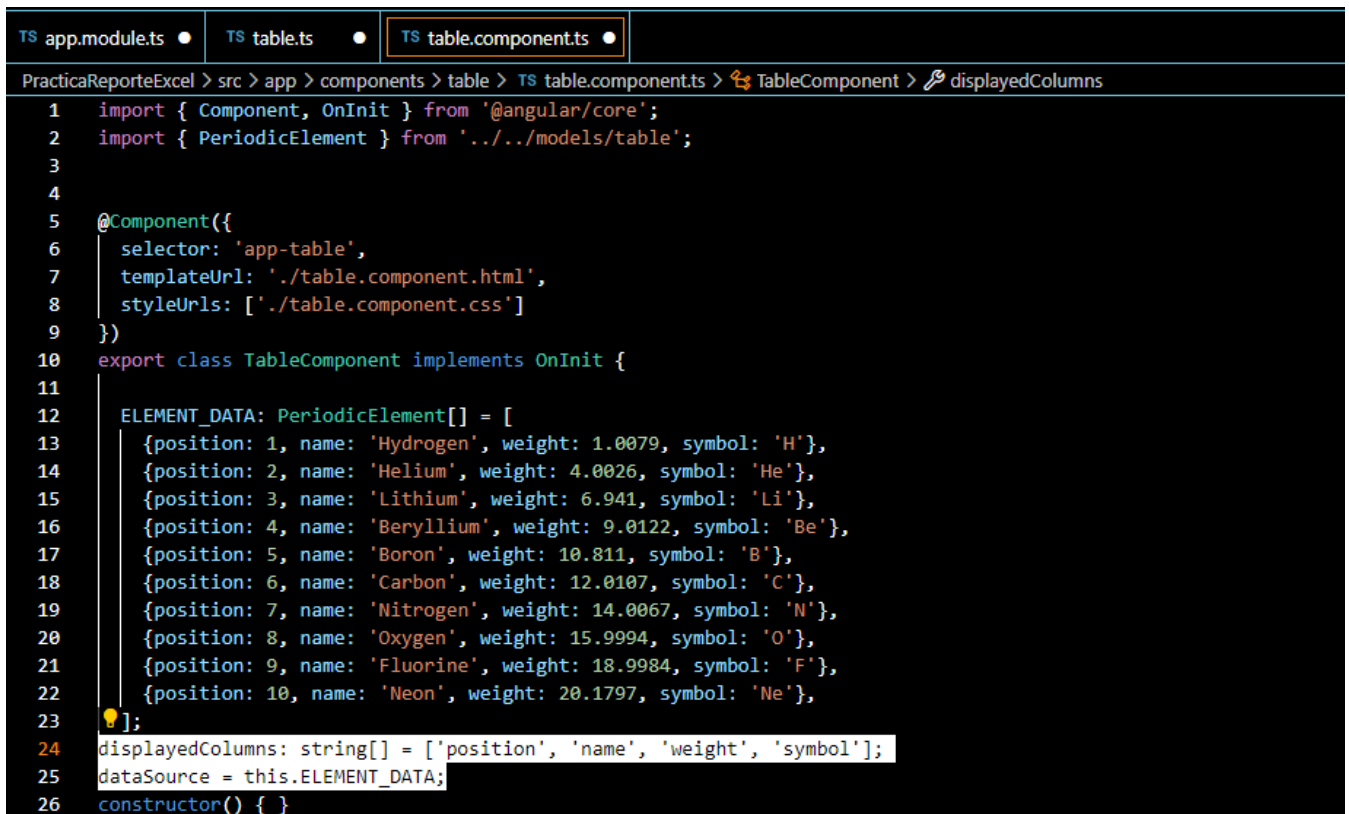
Paso 12: El paso número doce es copiar las variables de la tabla y después dirigirse al archivo table.component.ts y después de la constante se pega este código que es para tener las columnas y el datasource de la tabla.



```
weight: number;
symbol: string;
}

const ELEMENT_DATA: PeriodicElement[] = [
  {position: 1, name: 'Hydrogen', weight: 1.0079, symbol: 'H'},
  {position: 2, name: 'Helium', weight: 4.0026, symbol: 'He'},
  {position: 3, name: 'Lithium', weight: 6.941, symbol: 'Li'},
  {position: 4, name: 'Beryllium', weight: 9.0122, symbol: 'Be'},
  {position: 5, name: 'Boron', weight: 10.811, symbol: 'B'},
  {position: 6, name: 'Carbon', weight: 12.0107, symbol: 'C'},
  {position: 7, name: 'Nitrogen', weight: 14.0067, symbol: 'N'},
  {position: 8, name: 'Oxygen', weight: 15.9994, symbol: 'O'},
  {position: 9, name: 'Fluorine', weight: 18.9984, symbol: 'F'},
  {position: 10, name: 'Neon', weight: 20.1797, symbol: 'Ne'},
];

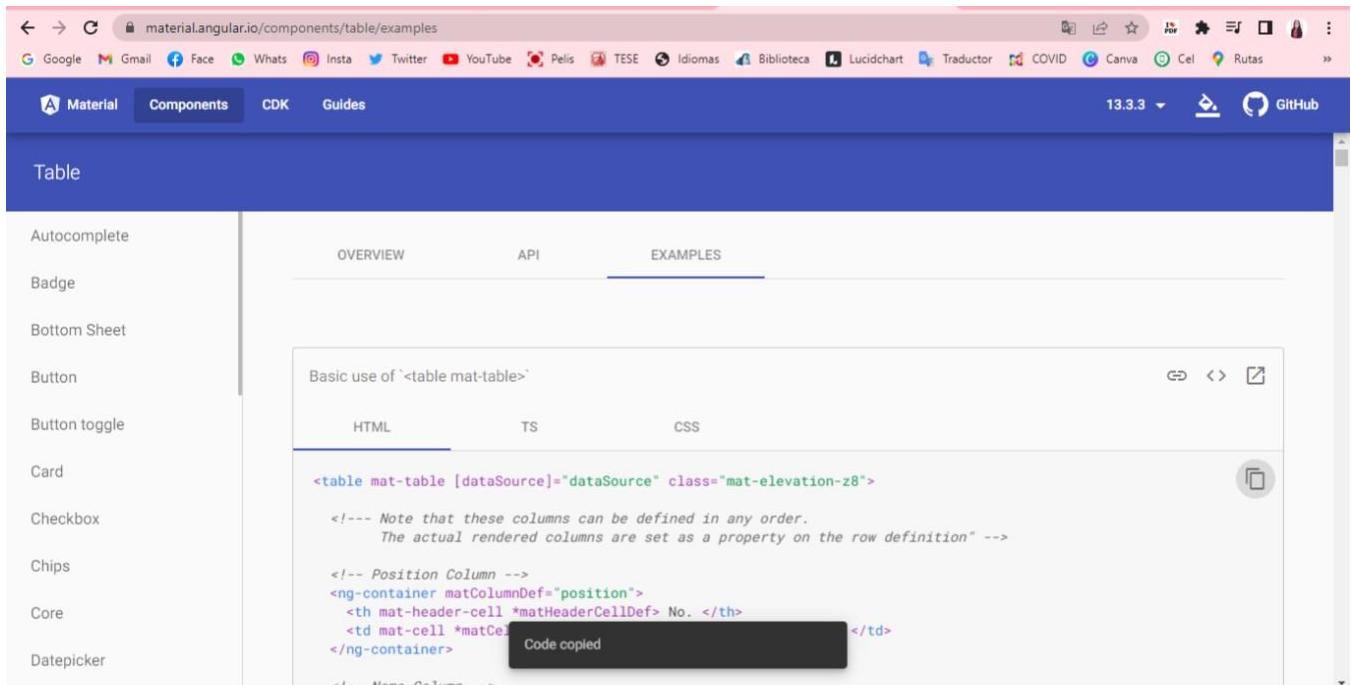
/**
 * @title Basic use of `<table mat-table>`
 */
@Component({
  selector: 'table-basic-example',
  styleUrls: ['table-basic-example.css'],
  templateUrl: 'table-basic-example.html',
})
export class TableBasicExample {
  displayedColumns: string[] = ['position', 'name', 'weight', 'symbol'];
  dataSource = ELEMENT_DATA;
}
```



```
TS app.module.ts • TS table.ts • TS table.component.ts •
PracticaReporteExcel > src > app > components > table > TS table.component.ts > TableComponent > displayedColumns

1 import { Component, OnInit } from '@angular/core';
2 import { PeriodicElement } from '../models/table';
3
4
5 @Component({
6   selector: 'app-table',
7   templateUrl: './table.component.html',
8   styleUrls: ['./table.component.css']
9 })
10 export class TableComponent implements OnInit {
11
12   ELEMENT_DATA: PeriodicElement[] = [
13     {position: 1, name: 'Hydrogen', weight: 1.0079, symbol: 'H'},
14     {position: 2, name: 'Helium', weight: 4.0026, symbol: 'He'},
15     {position: 3, name: 'Lithium', weight: 6.941, symbol: 'Li'},
16     {position: 4, name: 'Beryllium', weight: 9.0122, symbol: 'Be'},
17     {position: 5, name: 'Boron', weight: 10.811, symbol: 'B'},
18     {position: 6, name: 'Carbon', weight: 12.0107, symbol: 'C'},
19     {position: 7, name: 'Nitrogen', weight: 14.0067, symbol: 'N'},
20     {position: 8, name: 'Oxygen', weight: 15.9994, symbol: 'O'},
21     {position: 9, name: 'Fluorine', weight: 18.9984, symbol: 'F'},
22     {position: 10, name: 'Neon', weight: 20.1797, symbol: 'Ne'},
23   ];
24   displayedColumns: string[] = ['position', 'name', 'weight', 'symbol'];
25   dataSource = this.ELEMENT_DATA;
26   constructor() { }
```


Paso 13: El paso número trece es copiar el código de html para después pegarlo en el archivo table.component.html.



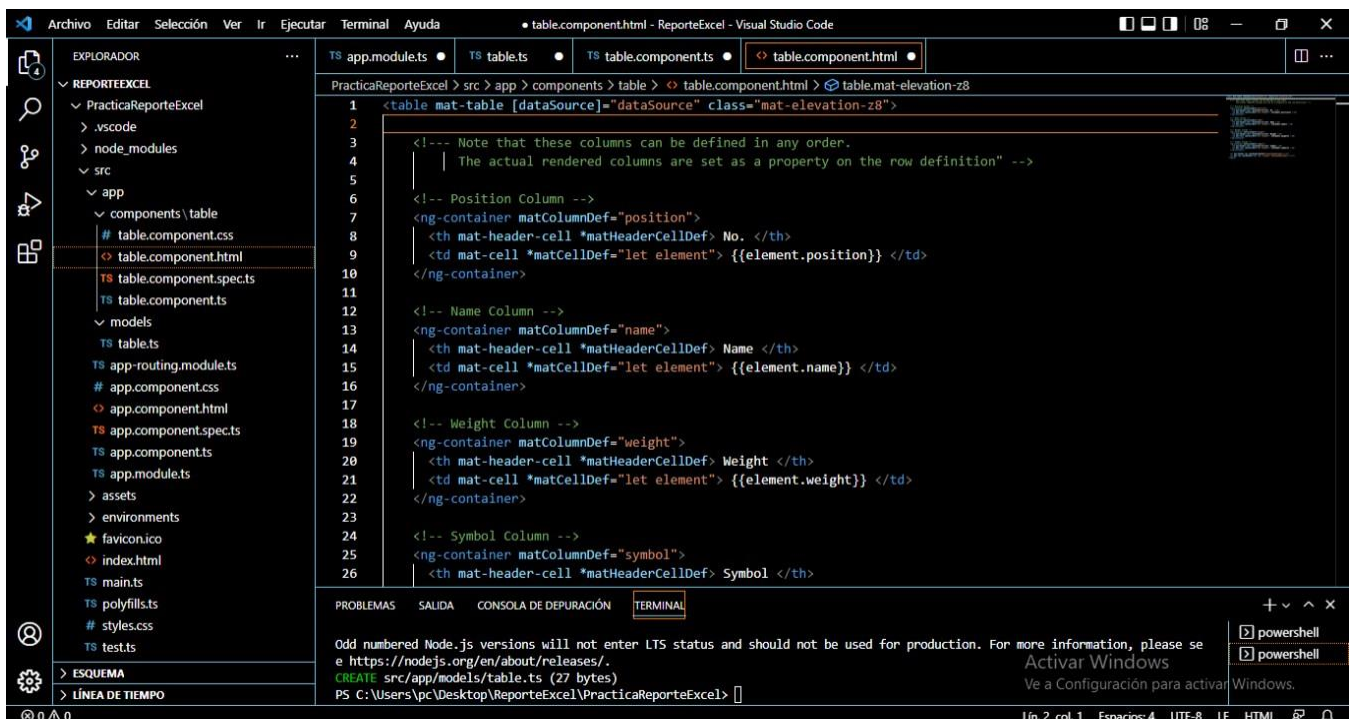
The screenshot shows the Material Angular website at material.angular.io/components/table/examples. The 'Table' component is selected in the left sidebar. The 'EXAMPLES' tab is active, displaying the 'Basic use of <table mat-table>' example. The HTML code is shown, and a 'Code copied' tooltip is visible over the code.

```
<table mat-table [dataSource]="dataSource" class="mat-elevation-z8">

  <!-- Note that these columns can be defined in any order.
        The actual rendered columns are set as a property on the row definition" -->

  <!-- Position Column -->
  <ng-container matColumnDef="position">
    <th mat-header-cell *matHeaderCellDef> No. </th>
    <td mat-cell *matCellDef="let element"> {{element.position}} </td>
  </ng-container>

  <!-- Name Column -->
```



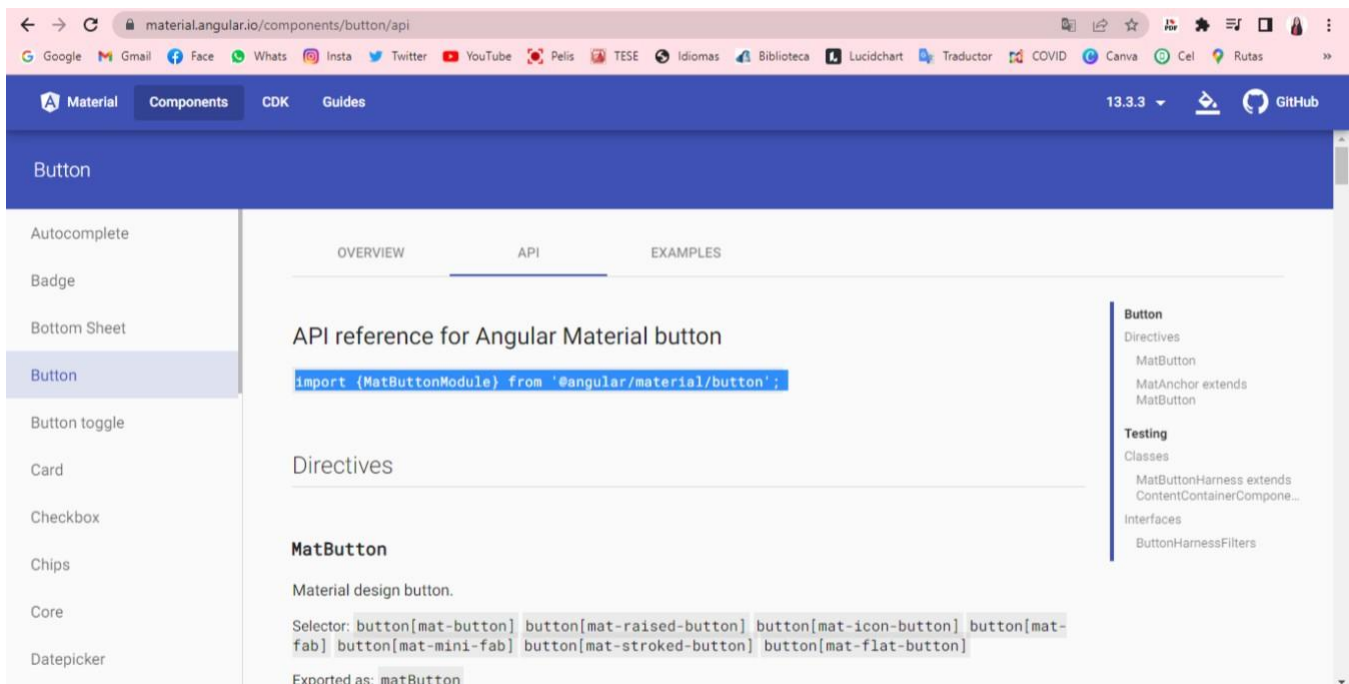
The screenshot shows the Visual Studio Code editor with the 'table.component.html' file open. The file content matches the HTML code from the previous screenshot. The Explorer sidebar on the left shows the project structure, including 'PracticaReporteExcel' and 'table.component.html'. The Terminal at the bottom shows a message about Node.js versions and a PowerShell prompt.

```
1 <table mat-table [dataSource]="dataSource" class="mat-elevation-z8">
2
3 <!-- Note that these columns can be defined in any order.
4 | The actual rendered columns are set as a property on the row definition" -->
5
6 <!-- Position Column -->
7 <ng-container matColumnDef="position">
8 | <th mat-header-cell *matHeaderCellDef> No. </th>
9 | <td mat-cell *matCellDef="let element"> {{element.position}} </td>
10 </ng-container>
11
12 <!-- Name Column -->
13 <ng-container matColumnDef="name">
14 | <th mat-header-cell *matHeaderCellDef> Name </th>
15 | <td mat-cell *matCellDef="let element"> {{element.name}} </td>
16 </ng-container>
17
18 <!-- Weight Column -->
19 <ng-container matColumnDef="weight">
20 | <th mat-header-cell *matHeaderCellDef> Weight </th>
21 | <td mat-cell *matCellDef="let element"> {{element.weight}} </td>
22 </ng-container>
23
24 <!-- Symbol Column -->
25 <ng-container matColumnDef="symbol">
26 | <th mat-header-cell *matHeaderCellDef> Symbol </th>
```

Paso 14: El paso número catorce es abrir el archivo app.component.html, se debe borrar todo lo que contiene este archivo y después se debe llamar el router outlet y el componente de la tabla.

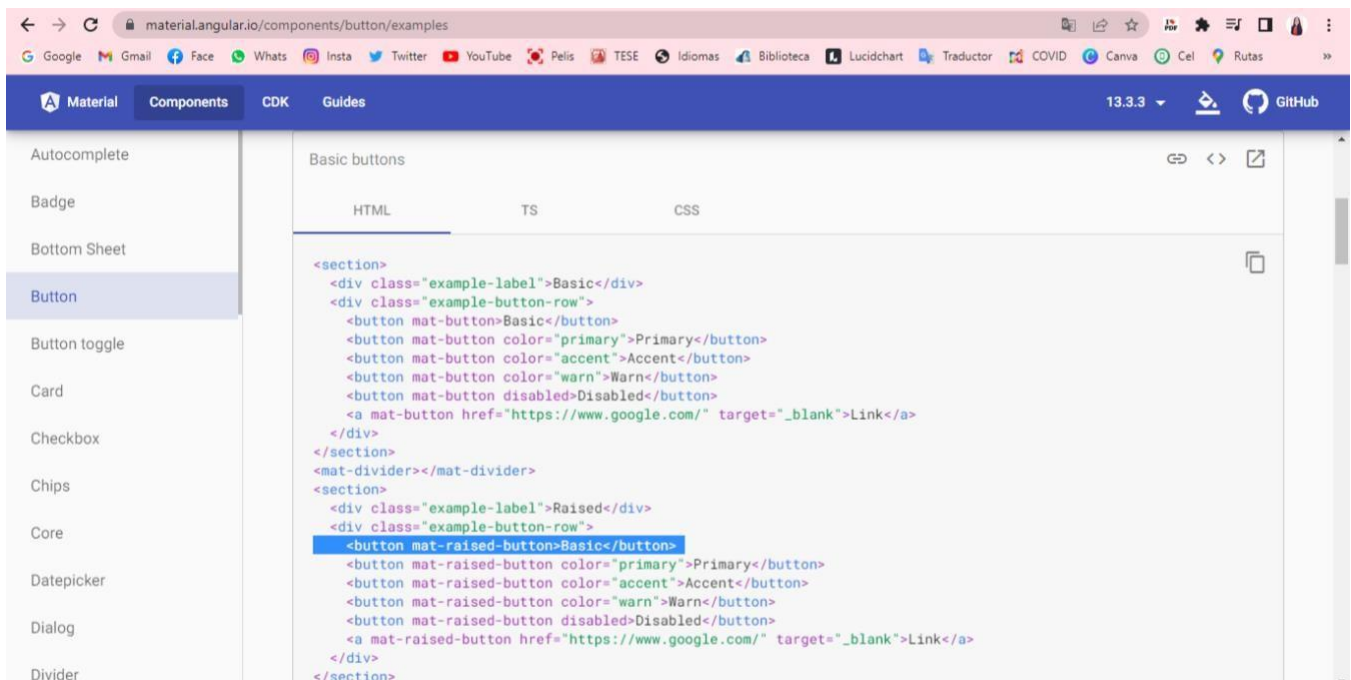
```
TS app.module.ts • TS table.ts • TS table.component.ts • <> table.component.html • <> app.component.html •
PracticaReporteExcel > src > app > <> app.component.html > div > app-table
1 <div>
2   <app-table></app-table>
3   <router-outlet></router-outlet>
4 </div>
5
```

Paso 15: El paso número quince consiste en dirigirse a angular material, ir a los componentes, después a los botones, elegir el API, copiarlo para posteriormente pegarlo y así importarlo en app.module.ts.



```
TS app.module.ts • TS table.ts TS table.component.ts <> table.component.html • <> app.component.html •
PracticaReporteExcel > src > app > TS app.module.ts > AppModule
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
7 import { TableComponent } from './components/table/table.component';
8 import { MatTableModule } from '@angular/material/table';
9 import { MatButtonModule } from '@angular/material/button';
10 import { MatTableExporterModule } from 'mat-table-exporter';
11
12 @NgModule({
13   declarations: [
14     AppComponent,
15     TableComponent
16   ],
17   imports: [
18     BrowserModule,
19     AppRoutingModule,
20     BrowserAnimationsModule,
21     MatTableModule,
22     MatButtonModule,
23     MatTableExporterModule
24   ],
25 })
```

Paso 16: El paso número dieciséis es ir a los ejemplos de los botones en angular material y elegir el que se va a utilizar, copiar su código para después agregarlo en el archivo table.component.html.



The screenshot shows the Angular Material documentation website at material.angular.io/components/button/examples. The 'Button' component is selected in the left sidebar. The main content area displays two sections of button examples: 'Basic buttons' and 'Raised buttons'. The 'Basic buttons' section shows a row of buttons with different colors (Basic, Primary, Accent, Warn, Disabled) and a link. The 'Raised buttons' section shows a row of raised buttons with the same color variations and a link. The code for the 'Basic buttons' section is highlighted in the image.

```
<section>
  <div class="example-label">Basic</div>
  <div class="example-button-row">
    <button mat-button>Basic</button>
    <button mat-button color="primary">Primary</button>
    <button mat-button color="accent">Accent</button>
    <button mat-button color="warn">Warn</button>
    <button mat-button disabled>Disabled</button>
    <a mat-button href="https://www.google.com/" target="_blank">Link</a>
  </div>
</section>
<mat-divider></mat-divider>
<section>
  <div class="example-label">Raised</div>
  <div class="example-button-row">
    <button mat-raised-button>Basic</button>
    <button mat-raised-button color="primary">Primary</button>
    <button mat-raised-button color="accent">Accent</button>
    <button mat-raised-button color="warn">Warn</button>
    <button mat-raised-button disabled>Disabled</button>
    <a mat-raised-button href="https://www.google.com/" target="_blank">Link</a>
  </div>
</section>
```

TS app.module.ts	TS table.ts	TS table.component.ts	<> table.component.html	<> app.component.html
PracticaReporteExcel > src > app > components > table > <> table.component.html > mat-table.mat-elevation-z8				
<pre> 14 <!-- Weight Column --> 15 <ng-container matColumnDef="weight"> 16 <th mat-header-cell *matHeaderCellDef> Weight </th> 17 <td mat-cell *matCellDef="let element"> {{element.weight}} </td> 18 </ng-container> 19 20 <!-- Symbol Column --> 21 <ng-container matColumnDef="symbol"> 22 <th mat-header-cell *matHeaderCellDef> Symbol </th> 23 <td mat-cell *matCellDef="let element"> {{element.symbol}} </td> 24 </ng-container> 25 26 <tr mat-header-row *matHeaderRowDef="displayedColumns"></tr> 27 <tr mat-row *matRowDef="let row; columns: displayedColumns;"></tr> 28 </table> 29 <button mat-raised-button (click)="exporter.exportTable('csv')" >Export to csv</button> 30 31 </pre>				

Paso 17: El paso número diecisiete consiste en agregar la directiva o método que se encuentra en la documentación para darle función al botón de exportación a archivos Excel, como importaciones para exportar y la función al presionar el botón.

```

1 <mat-table matTableExporter [dataSource]="dataSource" class="mat-elevation-z8" #exporter="matTableExporter">
2 <!-- Position Column -->
3 <ng-container matColumnDef="position">
4   <mat-header-cell *matHeaderCellDef> No. </mat-header-cell>
5   <mat-cell *matCellDef="let element"> {{element.position}} </mat-cell>
6 </ng-container>
7

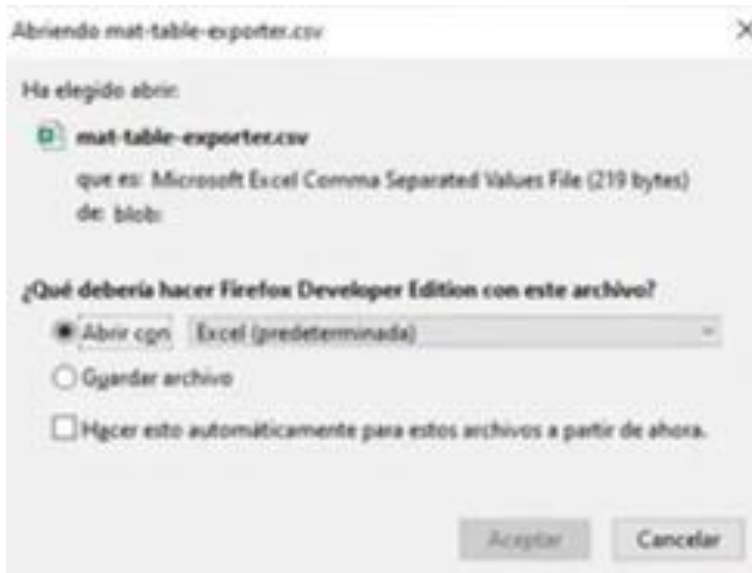
```

```

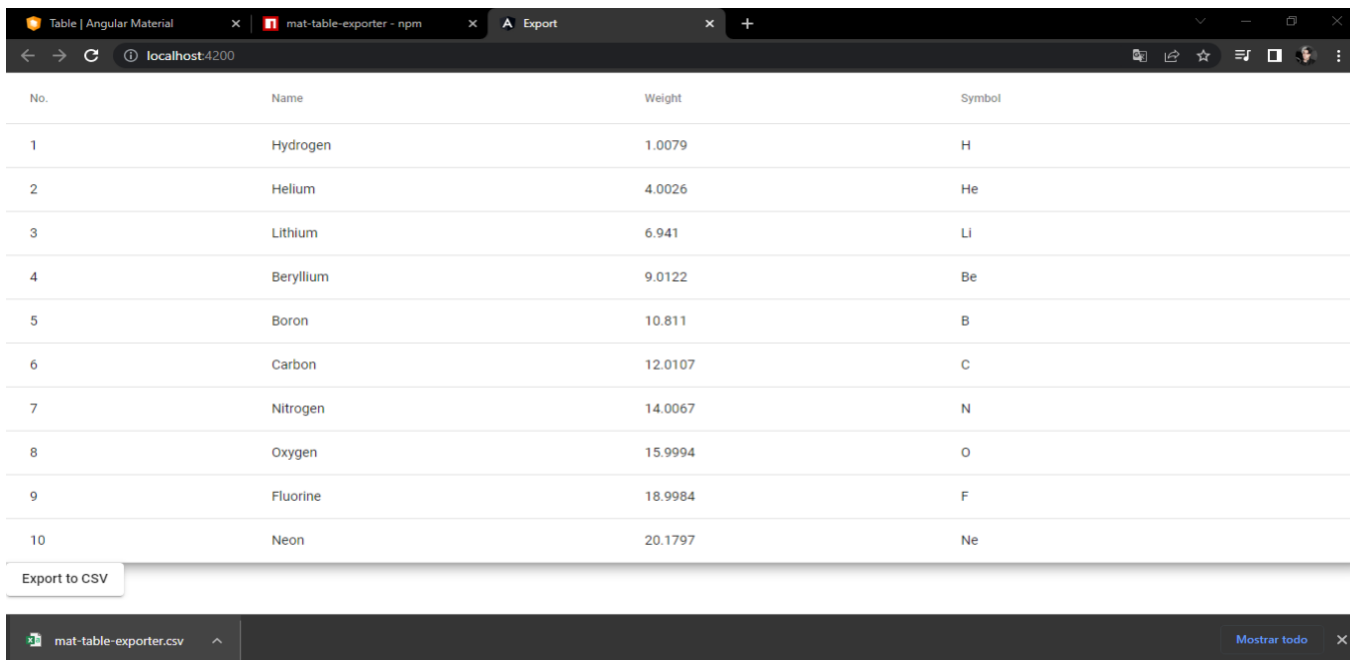
src > app > app.module.ts > AppModule
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { BrowserModule } from '@angular/platform-browser/animations';
7 import { TableComponent } from './components/table/table.component';
8 import { MatTableModule } from '@angular/material/table';
9 import { MatButtonModule } from '@angular/material/button';
10 import { MatTableExporterModule } from 'mat-table-exporter';
11
12
13
14 @NgModule({
15   declarations: [
16     AppComponent,
17     TableComponent
18   ],
19   imports: [
20     BrowserModule,
21     AppRoutingModule,
22     BrowserModule,
23     MatTableModule,
24     MatButtonModule,
25     MatTableExporterModule
26   ],

```

Paso 18: El siguiente paso consiste en compilar y ejecutar el proyecto creador para que se abra en el navegador con la dirección de localhost y verificar que se encuentre la tabla y el botón, así como la función del botón que en el momento que se le dé clic se inicie la descarga del archivo para exportar la información en un Excel.



Paso 19: El penúltimo paso es aceptar la descarga para que comience y una vez que termine poder abrir el archivo.



Paso 20: El último paso es abrir el archivo de Excel para comprobar que los datos que se muestran en la tabla son los mismos datos que se ingresaron en el proyecto de Angular.

The screenshot shows a web browser window with a table of elements and an Excel spreadsheet overlaying it. The table in the browser has columns: No., Name, Weight, and Symbol. The Excel spreadsheet also has columns: No., Name, Weight, and Symbol. The data in both tables matches.

No.	Name	Weight	Symbol
1	Hydrogen	1.0079	H
2	Helium	4.0026	He
3	Lithium	6.941	Li
4	Beryllium	9.0122	Be
5	Boron	10.811	B
6	Carbon	12.0107	C
7	Nitrogen	14.0067	N
8	Oxygen	15.9994	O
9	Fluorine	18.9984	F
10	Neon	20.1797	Ne

Export to CSV

Autoguardado mat-table-exporter (1) Brandon Evann

Archivo Inicio Insertar Disposición de página Fórmulas Datos Revisar Vista Ayuda Comentarios

Calibri 11 Fuente Alineación Número Estilos

POSIBLE PÉRDIDA DE DATOS Algunas características del libro se pueden perder si lo guarda como CSV (delimitado por comas). Para conservar estas características, guárdelo como archivo de Excel.

No mostrar de nuevo Guardar con

mat-table-exporter (1)

Listo Accesibilidad: No disponible Promedio: 8.39848 Recuento: 44 Suma: 167.9696

Link del Repositorio:

<https://github.com/BrandonMitznahuatl/Modulo-Actividad>

Link del funcionamiento con GitHub Pages:

<https://brandonmitznahuatl.github.io/Modulo-Actividad/>

