**Tecnológico de Estudios Superiores de Ecatepec**

**División de Ingeniería en Sistemas Computacionales**

**Academia en Ciencias de la Ingeniería**

**Materia** Base De Datos Para Dispositivos Móviles

**Grupo 5851**

**Equipo** Roma

**Integrantes**

Arreguin Campos Bryan Gustavo

Campero Granados Luis Daniel

Hernández Cervantes Geovanni

Rodríguez Fuentes Axel Román

Ruiz Teodocio José Pablo

**Profesora**

Griselda Cortes Barrera

***Módulo asignado funcionando - Manual***

# Manual

Para poder crear en archivo *.pdf* mediante lo primero que debemos hacer es instalar los paquetes necesarios:

1. Instalamos **Bootstrap**



1. Después instalamos los paquetes **jspdf@1.5.3 html2canvas**



Al tener instalado lo anterior, debemos instalar las dependencias de desarrollo:



1. Podemos visualizar las dependencias que instalamos, en el fichero **package.json**

Texto

Descripción generada automáticamente

Imagen que contiene botella, foto, tabla, alimentos

Descripción generada automáticamente

Añadimos las referencias de los estilos de Bootstrap en la sección de “**styles**” y añadir los scrips de jsPDF y html2canvas, esto se hará en el fichero de **angular.json**

Texto

Descripción generada automáticamente

Añadimos **“allowSyntheticDefaultImports”: true** dentro del apartado **compilerOptions,** esto se encontrara en la configuración de TypeScrip (**tsconfig.json**).

Texto

Descripción generada automáticamente

Importamos las librerías jsPDFen el **app.component.ts:**

**-import jsPDF from ‘jspdf’;**

**-import html2canvas from ‘html2canvas’;**

Texto

Descripción generada automáticamente

Para crear las tablas en nuestro documento, deberemos de seguir los siguientes pasos:

1. Crear un fichero en la carpeta “app”, este fichero deberá tener la extensión “**.json**”.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

1. En este fichero agregamos toda la información que nosotros deseemos mostrar.
2. Una vez agregada la información, deberemos de darle formato, este formato es como se desea que vaya estructurada la tabla.

Ejemplo:

Texto

Descripción generada automáticamente

1. Una vez agregada la información con formato, deberemos de dirigirnos al archivo ts.config.json y en la parte de **“CompileOptiones”** deberemos de agregar la línea de comando de **“resolvedJsonModule”: true**

Texto

Descripción generada automáticamente

1. Una vez agregado nos dirigimos al componente **“app.component.ts”** e importamos un archivo con el fichero que nosotros creamos.

Texto

Descripción generada automáticamente

1. Una vez importado deberemos de realizar un llamado al archivo junto con la palabra Data

Ejemplo:

**archivoData = archivoimportado;**

Texto

Descripción generada automáticamente

1. Para hacer nuestra tabla información se deberá de agregar una línea de comando debajo del constructor en el archivo *app.component.ts* que es:

Graphical user interface, logo

Description automatically generated

1. Una vez hecho esto se correrá el programa.
2. Una vez que hayamos corrido el programa, nos dirigimos **app.component.html** y añadimos lo siguiente, para crear una tabla con la información.

<div class="container">

<div class="row">

<div id="htmlData">

<h2>Lista de videojuegos en stock</h2>

<table class="table table-bordered mt-4">

<tr class="table-primary">

<th>Id</th>

<th>Name</th>

<th>Platform</th>

<th>Reference</th>

</tr>

<tr \*ngFor="Informacion">

<th>{{c.id}}</th>

<td>{{c.name}}</td>

<td>{{c.platform}}</td>

<td>{{c.reference}}</td></tr>

</table>

</div>

</div>

</div>

<div>

<button class="btn col col-lg-4 btn-danger btn-block ml-4" (click)="downloadPDF()">Download PDF</button>

</div>

Texto

Descripción generada automáticamente

1. Una vez hecho esto, procederemos a agregar un estilo con Bootstrap, para darle diseño a la tabla.
2. Nos dirigiremos a la parte de aoo.component.css y agregaremos el estilo de tabla que nosotros deseamos darle:

Ejemplo darle color a una fila:

Texto

Descripción generada automáticamente

1. Una vez teniendo el diseño de la tabla y donde muestre los datos vamos a agregar la configuración de la hoja del pdf donde obtendremos la tabla del **‘htmlData”** donde se muestran los datos de la API. Esto se agregara en la sección de **downloadPDF()**

Texto

Descripción generada automáticamente

1. Posteriormente haremos uso de html2canvas que ya habíamos importado con anterioridad.

Texto

Descripción generada automáticamente

1. En esa sección mandamos a llamar los datos y opciones del pdf, con el uso de un método que llamaremos *html2canvas que* añadiremos en el archivo *app.component.ts* añadiremos una imagen del canvas en formato PNG.

Interfaz de usuario gráfica, Texto

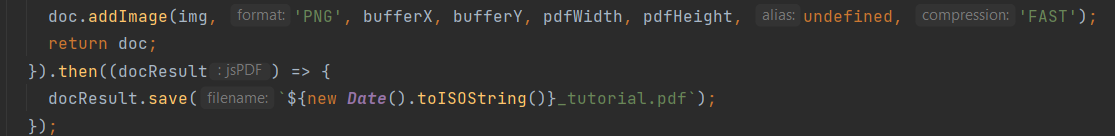
Descripción generada automáticamente

1. Añadimos la imagen del canvas para darle formato dentro de este mismo método.

Texto

Descripción generada automáticamente

1. Aquí añadimos la imagen, características y posteriormente devolvemos un documento. También nos va a mostrar la fecha de descarga y la hora dentro del mismo método.



1. Para generar el pdf añadiremos un botón en el **app.component.html** donde llamaremos a la función de **downloadPDF**

Texto

Descripción generada automáticamente

1. Una vez agregado todo, procederemos a ingresar en la consola “npm start” para inicializar todo y que quede guardado correctamente, y el resultado se mostrara así:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Agregar imágenes y estas se muestren en los archivos .pdf

En el archivo *app.component.html* agregaremos una etiqueta img de la siguiente forma

Graphical user interface, text

Description automatically generated

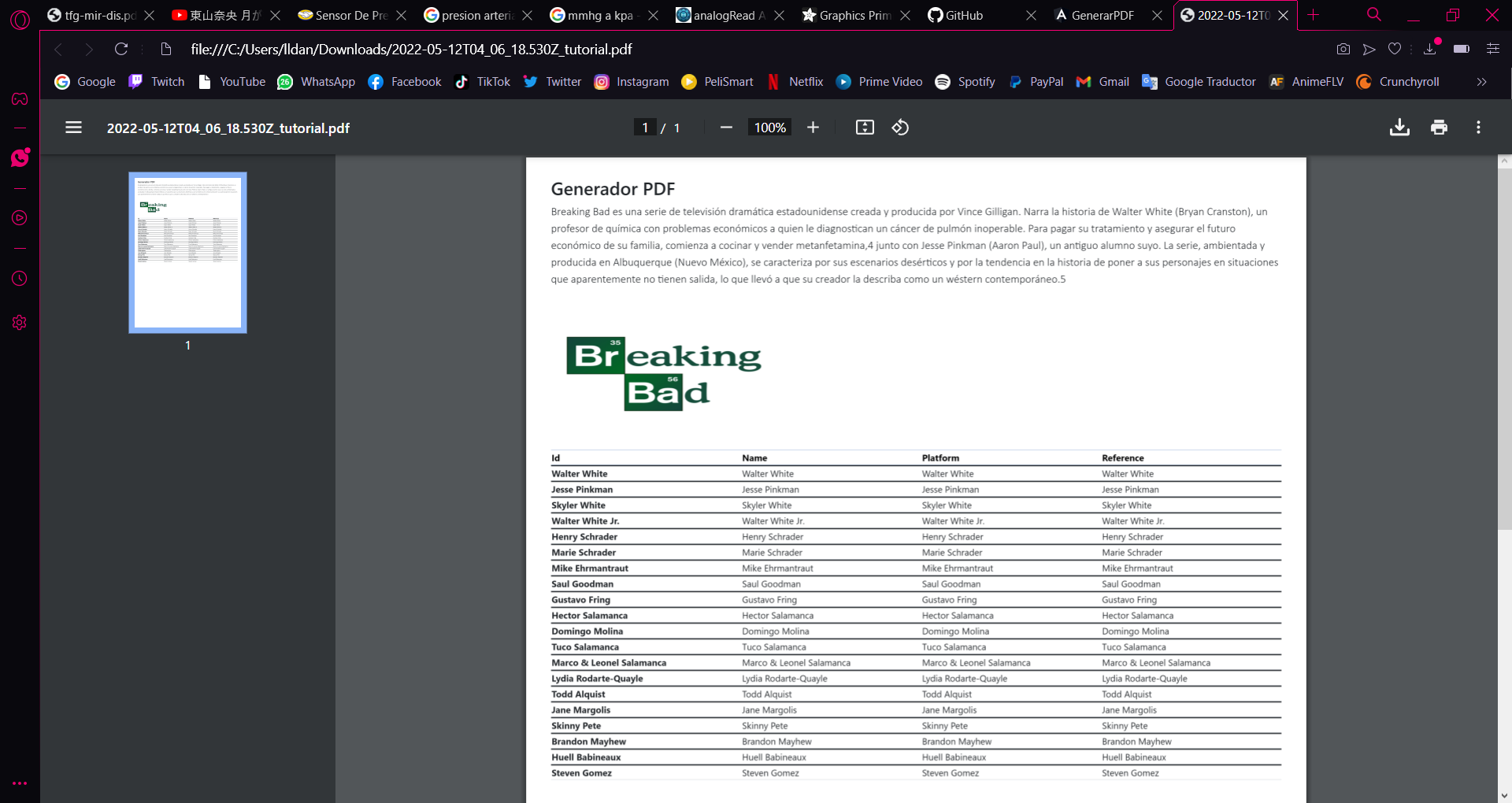
En nuestro archivo *app.component.css* podemos modificarle el diseño a nuestra imagen

A screenshot of a computer

Description automatically generated with medium confidence

Nota: la imagen deberá de estar dentro de nuestra carpeta *assets*

Listo, una vez que generemos nuestro pdf podremos ver nuestra imagen dentro de el:



Link del repositorio en GitHub:

<https://github.com/ghc4884/PROYECTO-PAQUETERIA-EQUIPO-ROMA>