Proyecto práctico para validación y mejoramiento de habilidades en Programación Frontend (Angular)

Pre-requisitos:

- ☐ El desarrollo de este proyecto se centra en el uso del framework Angular, se sugiere trabajar con la versión 11+
- ☐ Instalar node.js en su última versión estable (recomendado: lts/gallium -> v16.16.0)
- ☐ Instalar npm en su última versión (actual: 8.11.0)
- ☐ Instalar Angular CLI
- ☐ Utilizar un IDE de tu preferencia (Visual Studio Code, Angular IDE, Brackets, entre otros)

Consideraciones:

- Este proyecto se ha diseñado con el objetivo de validar conocimientos esenciales para el desarrollo de aplicaciones web en Angular.
- Se recomienda seguir las buenas prácticas en la creación, inicialización y estructuración del proyecto.
- Hacer uso del Angular CLI para familiarizarse con los comandos propios del framework, como la creación de componentes, servicios, etc.

Enunciado

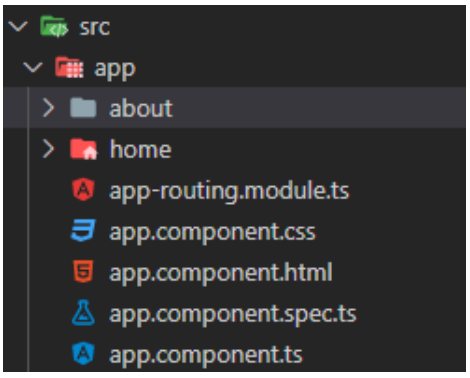
Vamos a crear el catálogo de una tienda en línea que ofrezca una gran variedad de productos. Los artículos se obtendrán a partir de la consulta a un servicio REST (mock) que puede ejecutarse de manera simultánea en el servidor local y que obtiene los datos de un archivo JSON (Ver apéndice A).

Como requerimientos del sistema, se tiene lo siguiente:

- Se deberían tener al menos 20 artículos cargados en el archivo JSON que servirá de base de datos (para obtener datos de prueba, se puede utilizar la librería Faker.js)
- Los atributos que debería tener cada producto son: id (número identificador único), nombre, descripción, precio, cantidad. (puede tener una imagen, pero es opcional - referencia para imagenes aleatorias: Lorem picsum)
- Se sugiere utilizar alguna librería o componente para elementos de interfaz gráfica (ej, Angular Material, Ng-Bootstrap)

Para efectos de este proyecto, se sugiere la creación de al menos un componente adicional al principal (app.component). De esta manera se puede reforzar el conocimiento del uso e integración de componentes en Angular.

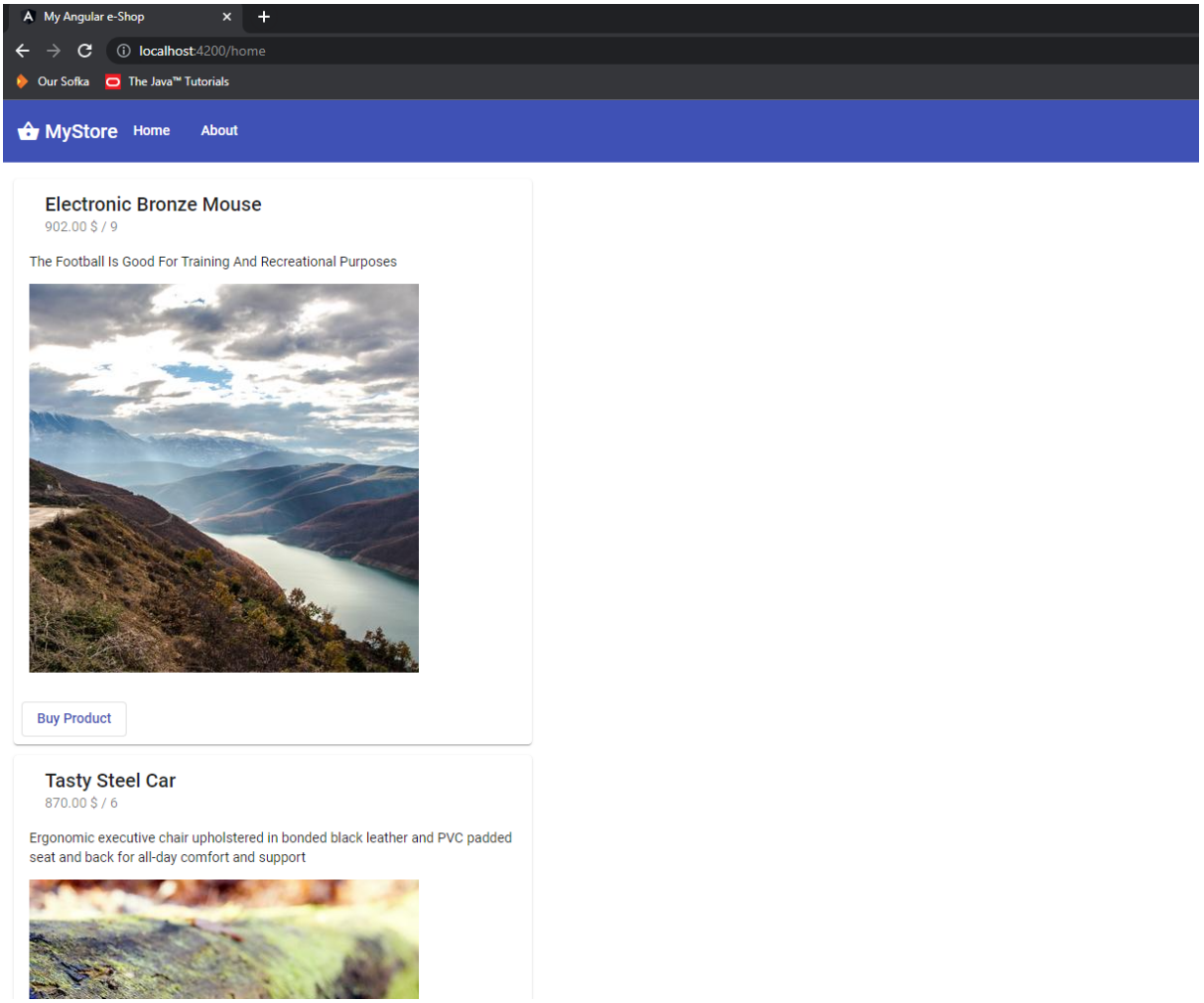
Ej.



También puede utilizarse este proyecto como práctica para manejo de repositorios git, por lo que se invita a experimentar con manejo de ramas (branches), restricciones de subida de archivos a través de un archivo .gitignore, etc.

➔ Recurso para la generación de archivos .gitignore basados en las tecnologías utilizadas: gitignore.io

Imagen de referencia de proyecto ejemplo:



Tiempo estimado sugerido para completar el proyecto: *2 días*

Apéndice A
Configuración de un servidor local para API Rest de prueba utilizando json-server

Documentación oficial del proyecto: https://github.com/typicode/json-server

Paso 1: Instalación de la librería json-server utilizando npm

- ☐ Utilizando el terminal de comandos de preferencia, ubicarse en el directorio raíz del proyecto.
- ☐ Ejecutar `npm install --save json-server`

Paso 2: Creación de directorio y archivo JSON de base de datos

- ☐ Crear un directorio llamado "server" dentro de la carpeta raíz del proyecto (utilizando el terminal de comandos o el explorador de archivos en el IDE)
- ☐ En el directorio "server", crear un archivo llamado "database.json"
- ☐ Editar el archivo "database.json" y escribir la siguiente estructura base para la BD de productos (ejemplo)

```
{
        "productos": [
                {
                        "id": 0,
                        "nombre": "Raqueta de tenis de mesa",
                        "descripcion": "Es una raqueta para jugar tenis de mesa",
                        "precio": "50.00",
                        "cantidad": "7"
                },
                { … }
        ]
}
```

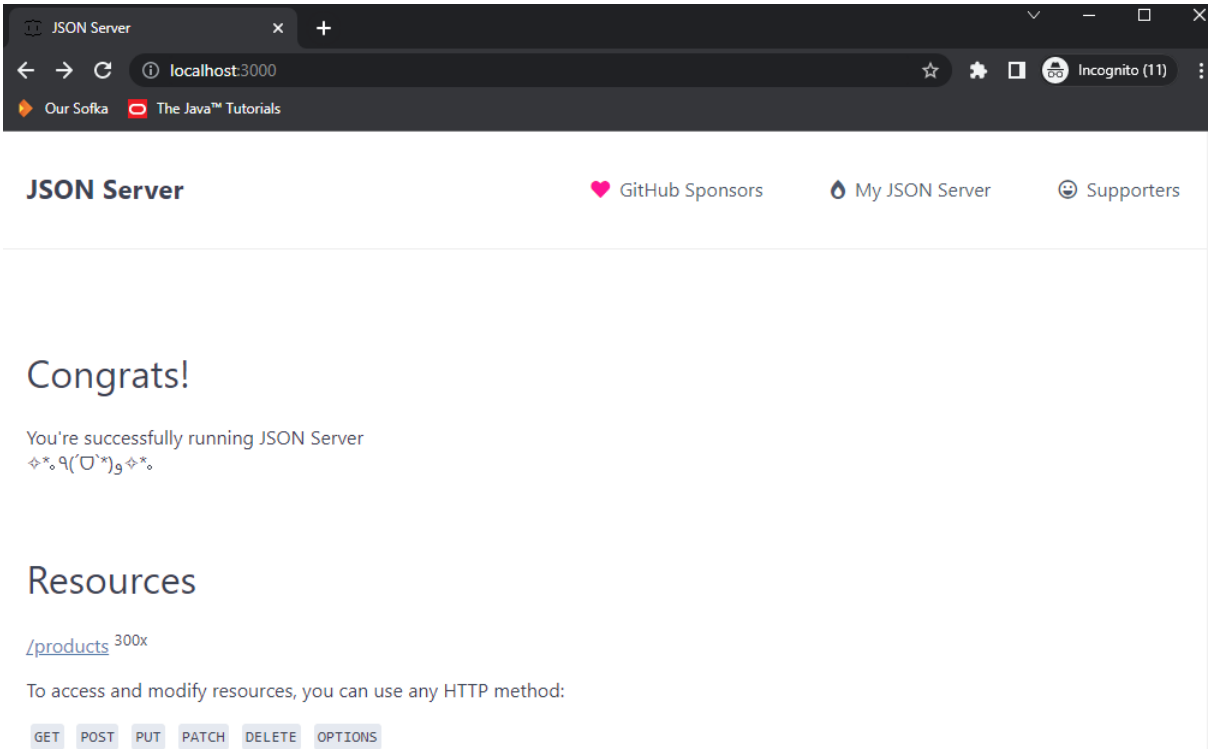
Reto adicional: Esta base de datos puede ser creada utilizando un servicio de datos ficticios, como Faker.js

Paso 3: Editar el archivo "package.json" para agregar el comando del servidor json-server

- ☐ Abrir el archivo package.json ubicado en el directorio raíz del proyecto
- ☐ En la sección de scripts, agregar la siguiente línea:
  "server": "json-server –watch ./server/database.json"

Paso 4: Ejecutar el servidor REST que tomara los datos de "database.json"

- ☐ En una consola, ubicarse en el directorio raíz del proyecto y ejecutar el siguiente comando:
  `npm run server`
- ☐ En una ventana del navegador de internet de su preferencia, se puede probar el resultado. El servidor se ejecuta por defecto en el puerto 3000 (http://localhost:3000). Debería verse como esto:


JSON Server
localhost:3000
Incognito (11)
Our Sofka
The Java™ Tutorials
JSON Server
GitHub Sponsors
My JSON Server
Supporters
Congrats!
You're successfully running JSON Server
✧*。٩(ˊᗜˋ*)و✧*。
Resources
/products 300x
To access and modify resources, you can use any HTTP method:
GET POST PUT PATCH DELETE OPTIONS