

Trabajo Práctico N°5

Ejercicio 1

The screenshot displays the Swagger UI for 'WebApplication2' at the URL `https://localhost:7044/swagger/index.html`. The interface shows the 'WeatherForecast' endpoint with a GET method. Below the endpoint, the 'Schemas' section lists 'WeatherForecast'. The 'Code' tab is active, showing a 200 status code and the 'Response body' in JSON format. The response body contains an array of five weather forecast objects, each with 'date', 'temperatureC', 'temperatureF', and 'summary' fields. The response headers are also visible at the bottom.

WebApplication2 1.0 OAS3
<https://localhost:7044/swagger/v1/swagger.json>

WeatherForecast

GET /WeatherForecast

Schemas

WeatherForecast >

Code Details

200

Response body

```
{
  "date": "2023-09-08T12:51:57.4285124-03:00",
  "temperatureC": 45,
  "temperatureF": 112,
  "summary": "Mild"
},
{
  "date": "2023-09-09T12:51:57.4289899-03:00",
  "temperatureC": 3,
  "temperatureF": 27,
  "summary": "Bracing"
},
{
  "date": "2023-09-10T12:51:57.4289927-03:00",
  "temperatureC": 13,
  "temperatureF": 55,
  "summary": "Bracing"
},
{
  "date": "2023-09-11T12:51:57.428993-03:00",
  "temperatureC": 21,
  "temperatureF": 69,
  "summary": "Balmy"
},
{
  "date": "2023-09-12T12:51:57.4289932-03:00",
  "temperatureC": 31,
  "temperatureF": 87,
  "summary": "Cool"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 07 Sep 2023 15:51:56 GMT
server: Kestrel
```

```

Usuario@DESKTOP-8G1HL0R MINGW64 ~/source/repos/WebApplication2
$ dotnet add package Newtonsoft.Json
Determinando los proyectos que se van a restaurar...
Writing C:\Users\Usuario\AppData\Local\Temp\tmpD37C.tmp
info : X.509 certificate chain validation will use the default trust store selected by .NET for code signing.
info : X.509 certificate chain validation will use the default trust store selected by .NET for timestamping.
info : Agregando PackageReference para el paquete "Newtonsoft.Json" al proyecto "C:\Users\Usuario\source\repos\WebApplication2\WebApplication2.csproj".
info : GET https://api.nuget.org/v3/registration5-gz-semver2/newtonsoft.json/index.json
info : OK https://api.nuget.org/v3/registration5-gz-semver2/newtonsoft.json/index.json 475 ms
info : Restaurando paquetes para C:\Users\Usuario\source\repos\WebApplication2\WebApplication2.csproj...
info : GET https://api.nuget.org/v3-flatcontainer/newtonsoft.json/index.json
info : OK https://api.nuget.org/v3-flatcontainer/newtonsoft.json/index.json 804 ms
info : GET https://api.nuget.org/v3-flatcontainer/newtonsoft.json/13.0.3/newtonsoft.json.13.0.3.nupkg
info : OK https://api.nuget.org/v3-flatcontainer/newtonsoft.json/13.0.3/newtonsoft.json.13.0.3.nupkg 31 ms
info : Se instaló Newtonsoft.Json 13.0.3 de https://api.nuget.org/v3/index.json con el hash de contenido Hrc5BXdl00IP9zeV+0Z848QWPaoCr9P3bDEZguI+gkLcBKA0xix/tLEAAHC+UvDNPv4a2d1810ReHMOagPa+zQ==.
info : El paquete "Newtonsoft.Json" es compatible con todos los marcos de trabajo especificados del proyecto "C:\Users\Usuario\source\repos\WebApplication2\WebApplication2.csproj".
info : Se agregó PackageReference para la versión "13.0.3" del paquete "Newtonsoft.Json" al archivo "C:\Users\Usuario\source\repos\WebApplication2\WebApplication2.csproj".
info : Generación de archivo MSBuild C:\Users\Usuario\source\repos\WebApplication2\obj\WebApplication2.csproj.nuget.g.props.
info : Generación de archivo MSBuild C:\Users\Usuario\source\repos\WebApplication2\obj\WebApplication2.csproj.nuget.g.targets.
info : Escribiendo el archivo de recursos en el disco. Ruta de acceso: C:\Users\Usuario\source\repos\WebApplication2\obj\project.assets.json
log : Se ha restaurado C:\Users\Usuario\source\repos\WebApplication2\WebApplication2.csproj (en 3.46 sec).

```

```

<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net6.0</TargetFramework>
    <Nullable>enable</Nullable>
    <ImplicitUsings>enable</ImplicitUsings>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="Newtonsoft.Json" Version="13.0.3" />
    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
  </ItemGroup>

</Project>

```

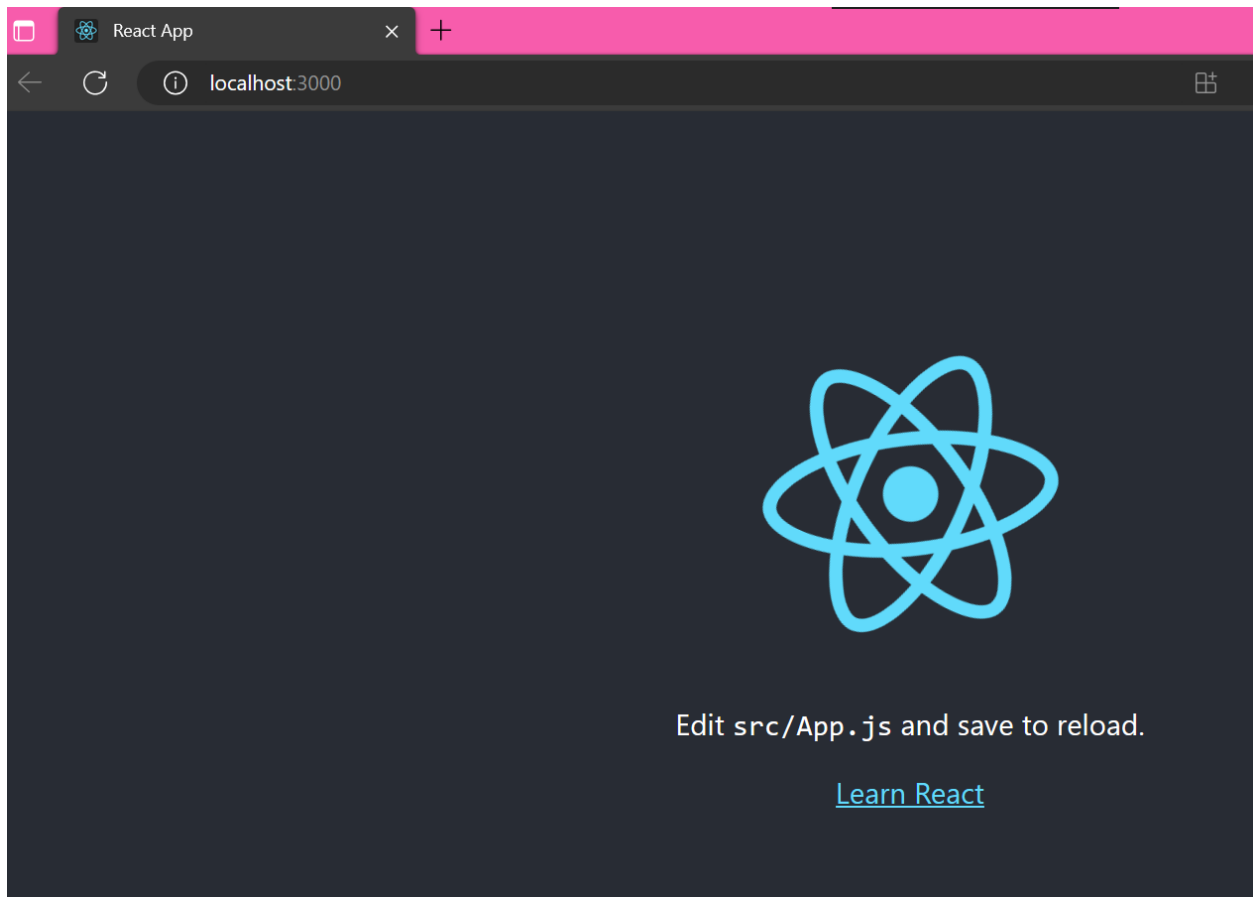
```

    "Newtonsoft.Json/13.0.3": {
      "type": "package",
      "compile": {
        "lib/net6.0/Newtonsoft.Json.dll": {
          "related": ".xml"
        }
      },
      "runtime": {
        "lib/net6.0/Newtonsoft.Json.dll": {
          "related": ".xml"
        }
      }
    },
  },

```

- Se agregó la entrada para la librería Newtonsoft.Json tanto en .csproj como en project.assets.json.
- bin y obj vuelven a crearse automáticamente al ejecutar el proyecto.

Ejercicio 2



You can now view my-app in the browser.

Local: `http://localhost:3000`
On Your Network: `http://192.168.56.1:3000`

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

Analizar el manejo de paquetes y dependencias realizado por npm:

Cuando `create-react-app` es ejecutado, se procede a la instalación de diversas dependencias esenciales para una aplicación React estándar. Entre estas dependencias se incluyen componentes clave como React, ReactDOM y otras herramientas interrelacionadas. Además, se efectúa una configuración de Webpack,

Babel y otros paquetes cruciales que permiten la compilación y ejecución fluida de la aplicación en un entorno de desarrollo.

Todas las dependencias específicas del proyecto, junto con las configuraciones asociadas, se registran meticulosamente en un archivo denominado `package.json`. Este archivo se crea y almacena en el directorio de tu proyecto, proporcionando un inventario detallado de los elementos necesarios para el funcionamiento de la aplicación, incluyendo sus versiones precisas.

Para mantener la coherencia y uniformidad en el desarrollo, npm también genera un archivo adicional llamado `package-lock.json`. Este archivo documenta de manera exacta las versiones de cada dependencia y sus subdependencias, asegurando que las mismas versiones de los paquetes sean empleadas de forma consistente en diversos entornos y por distintos miembros del equipo de desarrollo.

Ejercicio 3

1. Java:

- **Maven:** Maven es una herramienta de gestión de proyectos de código abierto que utiliza un archivo XML para describir el proyecto y sus dependencias.

2. Python:

- **setuptools:** Es el sistema de construcción y distribución estándar para proyectos de Python. También puede trabajar junto con herramientas como `pip` para instalar dependencias.

3. Ruby:

- **Rake:** Rake es una herramienta de construcción similar a Make para Ruby. Permite definir tareas y dependencias en un archivo `Rakefile`.

4. C++:

- **CMake:** CMake es una herramienta de construcción de código abierto que se utiliza para controlar el proceso de construcción de software utilizando archivos de configuración independientes del compilador.

5. Go:

- **go build:** Go tiene su propia herramienta de construcción incorporada. El comando `go build` compila los archivos del programa Go y produce un ejecutable.

6. **Rust:**

- **Cargo:** Cargo es la herramienta oficial de gestión de paquetes y construcción para el lenguaje de programación Rust.

Elegir al menos 10 lenguajes de la lista de top 20 o top 50 de tiobe:

1. **R:**

- Lenguaje de programación para estadísticas y análisis de datos.

2. **MatLab:**

- Utilizado para cálculos numéricos y técnicos, principalmente en ingeniería y matemáticas aplicadas.

3. **Fortran:**

- Lenguaje antiguo enfocado en aplicaciones científicas y de alto rendimiento.

4. **PHP:**

- Lenguaje de script para desarrollo web.

5. **SQL:**

- Lenguaje de consulta para bases de datos relacionales.

6. **C++:**

- Lenguaje eficiente y versátil, ampliamente utilizado en sistemas operativos y juegos.

7. **Java:**

- Lenguaje portátil, comúnmente usado en aplicaciones empresariales y Android.

8. **JavaScript:**

- Lenguaje del lado del cliente para interactividad web.

9. **Assembly Language:**

- Conjunto de instrucciones de bajo nivel para comunicarse con la CPU.

10. **Python:**

- Lenguaje de alto nivel versátil, usado en desarrollo web, análisis de datos y aprendizaje automático.