

TP 5 - Herramientas de construcción de software

1- Ejemplo con C# y .NET Core

```
PS C:\Users\sofia\OneDrive\Escritorio\IdS3\miwebapi> cd MiProyectoWebAPI
PS C:\Users\sofia\OneDrive\Escritorio\IdS3\miwebapi\MiProyectoWebAPI> dotnet run
Compilando...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5148
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\sofia\OneDrive\Escritorio\IdS3\miwebapi\MiProyectoWebAPI
```

```
localhost:5148/weatherforecast
[[{"date": "2023-08-30", "temperatureC": 47, "temperatureF": 116, "summary": "Scorching"}, {"date": "2023-08-31", "temperatureC": 25, "temperatureF": 76, "summary": "Scorching"}, {"date": "2023-09-01", "temperatureC": 19, "temperatureF": 66, "summary": "Mild"}, {"date": "2023-09-02", "temperatureC": 1, "temperatureF": 33, "summary": "Freezing"}, {"date": "2023-09-03", "temperatureC": -11, "temperatureF": 13, "summary": "Scorching"}]]
```

```
MiProyectoWebAPI.csproj X
C: > Users > sofia > OneDrive > Escritorio > IdS3 > miwebapi > MiProyectoWebAPI > MiProyectoWebAPI.csproj
1  <Project Sdk="Microsoft.NET.Sdk.Web">
2
3  <PropertyGroup>
4    <TargetFramework>net7.0</TargetFramework>
5    <Nullable>enable</Nullable>
6    <ImplicitUsings>enable</ImplicitUsings>
7  </PropertyGroup>
8
9  <ItemGroup>
10   <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="7.0.10" />
11   <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
12 </ItemGroup>
13
14 </Project>
15
```

Luego de borrar los directorios bin y obj y ejecutar el comando **dotnet add package Newtonsoft.Json**

```
PS C:\Users\sofia\OneDrive\Escritorio\IdS3\miwebapi\MiProyectoWebAPI> dotnet run
Compilando...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5148
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\sofia\OneDrive\Escritorio\IdS3\miwebapi\MiProyectoWebAPI
```

```
MiProyectoWebAPI.csproj X
C: > Users > sofia > OneDrive > Escritorio > IdS3 > miwebapi > MiProyectoWebAPI > MiProyectoWebAPI.csproj
1 <Project Sdk="Microsoft.NET.Sdk.Web">
2
3   <PropertyGroup>
4     <TargetFramework>net7.0</TargetFramework>
5     <Nullable>enable</Nullable>
6     <ImplicitUsings>enable</ImplicitUsings>
7   </PropertyGroup>
8
9   <ItemGroup>
10    <PackageReference Include="Microsoft.AspNetCore.OpenApi" Version="7.0.10" />
11    <PackageReference Include="Newtonsoft.Json" Version="13.0.3" />
12    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
13  </ItemGroup>
14
15 </Project>
16
```

Revisar contenido de directorio bin/debug/net7.0

MiProyectoWebAPI > bin > Debug > net7.0				
Nombre	Fecha de modificación	Tipo	Tamaño	
appsettings.Development.json	29/8/2023 14:32	Archivo de origen ...	1 KB	
appsettings.json	29/8/2023 14:32	Archivo de origen ...	1 KB	
Microsoft.AspNetCore.OpenApi.dll	14/7/2023 23:54	Extensión de la ap...	65 KB	
Microsoft.OpenApi.dll	26/9/2022 09:32	Extensión de la ap...	207 KB	
MiProyectoWebAPI.deps.json	29/8/2023 14:44	Archivo de origen ...	6 KB	
MiProyectoWebAPI.dll	29/8/2023 14:44	Extensión de la ap...	10 KB	
MiProyectoWebAPI.exe	29/8/2023 14:44	Aplicación	151 KB	
MiProyectoWebAPI.pdb	29/8/2023 14:44	Archivo PDB	21 KB	
MiProyectoWebAPI.runtimeconfig.json	29/8/2023 14:44	Archivo de origen ...	1 KB	
Newtonsoft.Json.dll	8/3/2023 04:09	Extensión de la ap...	696 KB	
Swashbuckle.AspNetCore.Swagger.dll	11/1/2023 03:48	Extensión de la ap...	15 KB	
Swashbuckle.AspNetCore.SwaggerGen.dll	11/1/2023 03:48	Extensión de la ap...	95 KB	
Swashbuckle.AspNetCore.SwaggerUI.dll	11/1/2023 03:49	Extensión de la ap...	3.178 KB	

2- Ejemplo con nodejs

```
PS C:\Users\sofia\OneDrive\Escritorio\IdS3> cd my-app
PS C:\Users\sofia\OneDrive\Escritorio\IdS3\my-app> npm start

> my-app@0.1.0 start
> react-scripts start
```



Edit `src/App.js` and save to reload.

[Learn React](#)

Analizar el manejo de paquetes y dependencias realizado por npm.

- Instalación de Node.js y npm: necesitas instalar Node.js para tener acceso a **npm**. **npm** es el sistema de gestión de paquetes para Node.js.
- Creación del Proyecto: se utiliza **npm create-react-app** para generar un proyecto con la estructura necesaria. **npm** se encarga de descargar las dependencias necesarias para la creación del proyecto.
- Dependencias en el package.json: **npm** utiliza el archivo package.json para gestionar las dependencias y configuraciones del proyecto. Cada vez que instalas un paquete, **npm** lo agrega automáticamente a este archivo.

```
{ } package.json X
C: > Users > sofia > OneDrive > Escritorio > IdS3 > my-app > { } package.json > ...
1  {
2    "name": "my-app",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.17.0",
7      "@testing-library/react": "^13.4.0",
8      "@testing-library/user-event": "^13.5.0",
9      "react": "^18.2.0",
10     "react-dom": "^18.2.0",
11     "react-scripts": "5.0.1",
12     "web-vitals": "^2.1.4"
13   },
14   "scripts": {
15     "start": "react-scripts start",
16     "build": "react-scripts build",
17     "test": "react-scripts test",
18     "eject": "react-scripts eject"
19   },
20   "eslintConfig": {
21     "extends": [
22       "react-app",
23       "react-app/jest"
24     ]
25   },
26 }
```

- Instalación de Dependencias: Cuando ejecutas **npm install**, npm lee el package.json y descarga todas las dependencias especificadas en él. Estas dependencias se almacenan en la carpeta node_modules. Para subir a github, se puede ignorar la carpeta de node_modules.

Escritorio > IdS3 > my-app > node_modules >

Nombre	Fecha de modificación	Tipo	Tamaño
.bin	29/8/2023 14:53	Carpeta de archivos	
.cache	29/8/2023 14:54	Carpeta de archivos	
@aashutoshtrathi	29/8/2023 14:52	Carpeta de archivos	
@adobe	29/8/2023 14:53	Carpeta de archivos	
@alloc	29/8/2023 14:52	Carpeta de archivos	
@ampproject	29/8/2023 14:52	Carpeta de archivos	
@babel	29/8/2023 14:52	Carpeta de archivos	
@bcoe	29/8/2023 14:52	Carpeta de archivos	
@csstools	29/8/2023 14:52	Carpeta de archivos	
@eslint	29/8/2023 14:52	Carpeta de archivos	
@eslint-community	29/8/2023 14:52	Carpeta de archivos	
@humanwhocodes	29/8/2023 14:52	Carpeta de archivos	
@istanbuljs	29/8/2023 14:52	Carpeta de archivos	
@jest	29/8/2023 14:53	Carpeta de archivos	
@jridgewell	29/8/2023 14:52	Carpeta de archivos	
@leichtgewicht	29/8/2023 14:52	Carpeta de archivos	
@nicolo-ribaudo	29/8/2023 14:52	Carpeta de archivos	

- Scripts en el package.json: **npm** permite definir scripts en el package.json. En el ejemplo de React, **npm start** es un script que inicia el servidor de desarrollo.
- Actualización de Dependencias: se puede actualizar las dependencias ejecutando **npm update**. **npm** también permite especificar rangos de versiones en el package.json para controlar cómo se actualizan las dependencias.

3 - Build tools para otros lenguajes

Hacer una lista de herramientas de build (una o varias) para distintos lenguajes, por ejemplo (Rust -> cargo)

- Python -> pipenv, Poetry
- Java -> Apache Maven, Gradle
- C -> Make, CMake
- C++ -> CMake, Bazel
- C# -> MSBuild, Cake
- JavaScript -> npm, Yarn
- PHP -> Composer
- Go -> go build, Dep, Go Modules

Elegir al menos 10 lenguajes de la lista de top 20 o top 50 de tiobe:
<https://www.tiobe.com/tiobe-index/>

1. Java
2. Python
3. C++
4. C
5. JavaScript
6. PHP
7. SQL
8. Assembly language
9. Go
10. R