

Trabajo Práctico 5 - Herramientas de construcción de software

1- Ejemplo con C# y .NET Core

```
tincho@IdeaPad-5-14ALC05:~$ dotnet --version
7.0.112
```

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5$ dotnet new webapi -n MiProyectoWebAPI
The template "ASP.NET Core Web API" was created successfully.

Processing post-creation actions...
Restoring /home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/MiProyectoWebAPI.csproj:
  Determining projects to restore...
  Restored /home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/MiProyectoWebAPI.csproj
  (in 256 ms).
Restore succeeded.

tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5$ cd MiProyectoWebAPI/
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ dotnet run
Building...
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5237
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: /home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI
```

```
[{"date": "2023-10-17", "temperatureC": 54, "temperatureF": 129, "summary": "Cool"}, {"date": "2023-10-18", "temperatureC": 24, "temperatureF": 75, "summary": "Sweltering"}, {"date": "2023-10-19", "temperatureC": -2, "temperatureF": 29, "summary": "Cool"}, {"date": "2023-10-20", "temperatureC": 46, "temperatureF": 114, "summary": "Warm"}, {"date": "2023-10-21", "temperatureC": -2, "temperatureF": 29, "summary": "Scorching"}]
```

```
Open  MiProyectoWebAPI.csproj  Sa
~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI
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="7.0.12" />
11    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
12  </ItemGroup>
13
14 </Project>
```

```

tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ dotnet add package
Newtonsoft.Json
Determining projects to restore...
Writing /tmp/tmpR4piHo.tmp
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/lib/dotnet/sdk/7.0.112/trustedroots/codesignctl.pem'.
info : X.509 certificate chain validation will use the fallback certificate bundle at '/usr/lib/dotnet/sdk/7.0.112/trustedroots/timestampctl.pem'.
info : Adding PackageReference for package 'Newtonsoft.Json' into project '/home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/MiProyectoWebAPI.csproj'.
info : CACHE https://api.nuget.org/v3/registration5-gz-semver2/newtonsoft.json/index.json
info : Restoring packages for /home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/MiProyectoWebAPI.csproj...
info : Package 'Newtonsoft.Json' is compatible with all the specified frameworks in project '/home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/MiProyectoWebAPI.csproj'.
info : PackageReference for package 'Newtonsoft.Json' version '13.0.3' added to file '/home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/MiProyectoWebAPI.csproj'.
info : Generating MSBuild file /home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/obj/MiProyectoWebAPI.csproj.nuget.g.props.
info : Generating MSBuild file /home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/obj/MiProyectoWebAPI.csproj.nuget.g.targets.
info : Writing assets file to disk. Path: /home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/obj/project.assets.json
log : Restored /home/tincho/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI/MiProyectoWebAPI.csproj (in 113 ms).

```

```

tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ rm -r bin
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ rm -r obj
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ ls
appsettings.Development.json  Controllers          Program.cs  WeatherForecast.cs
appsettings.json              MiProyectoWebAPI.csproj  Properties

```

```

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="7.0.12" />
11    <PackageReference Include="Newtonsoft.Json" Version="13.0.3" />
12    <PackageReference Include="Swashbuckle.AspNetCore" Version="6.5.0" />
13  </ItemGroup>
14
15 </Project>

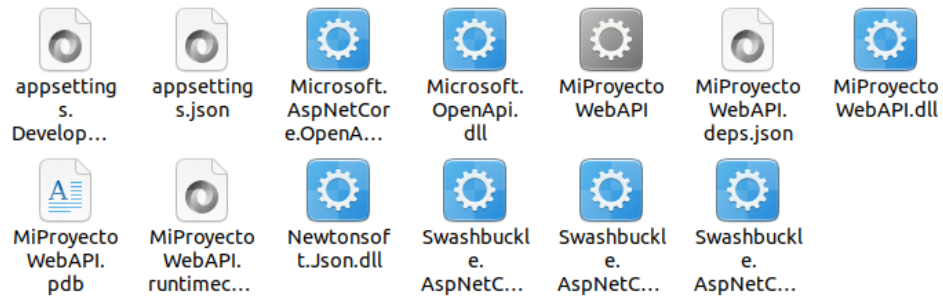
```

En cuanto a obj/debug/project.assets.json, el nuevo archivo de proyecto parece estar utilizando versiones más recientes de las bibliotecas y paquetes, lo que podría indicar actualizaciones o cambios en las dependencias de tu proyecto. Además, la estructura del archivo proporciona detalles más precisos sobre la configuración y los archivos relacionados con cada paquete.

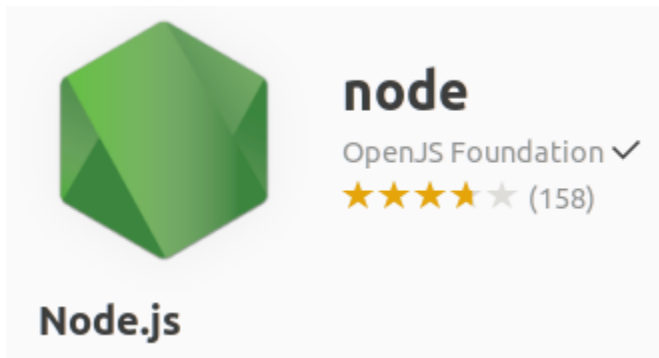
```

tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ rm -r bin
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ rm -r obj
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ ls
appsettings.Development.json  Controllers          Program.cs  WeatherForecast.cs
appsettings.json              MiProyectoWebAPI.csproj  Properties

```



2- Ejemplo con nodejs



```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5$ npx create-react-app my-app
Need to install the following packages:
create-react-app@5.0.1
Ok to proceed? (y) y
npm WARN deprecated tar@2.2.2: This version of tar is no longer supported, and will not receive security
updates. Please upgrade asap.

Creating a new React app in /home/tincho/Documents/Ingenieria de Software III/TP5/my-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...

added 1000 packages in 58s

39 packages are looking for funding
  run `npm fund` for details

Success! Created my-app at /home/tincho/Documents/Ingenieria de Software III/TP5/my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd my-app
  npm start

Happy hacking!
```



Welcome to React

To get started, edit `src/App.js` and save to reload.

Todas las dependencias y scripts específicos del proyecto se enumeran en un archivo llamado `package.json`. Este archivo es crucial para la gestión de dependencias en un proyecto. Cuando ejecuto `npm install` en el directorio de mi proyecto, `npm` lee el `package.json` y descarga e instala todas las dependencias necesarias. Puedo ver estas dependencias divididas en dos categorías: "dependencies" para las necesarias en producción y "devDependencies" para las necesarias solo durante el desarrollo.

También puedo agregar nuevas dependencias al proyecto ejecutando comandos como npm install <nombre-del-paquete>.

Además, npm permite especificar las versiones de las dependencias para mantener la compatibilidad de un proyecto. Esto es fundamental para evitar problemas de compatibilidad y garantizar que mi aplicación funcione de manera consistente en diferentes entornos.

3- Build tools para otros lenguajes

- Rust: Cargo
- Java: Apache Maven, Gradle
- C: Make, CMake
- Python: setuptools, Distutils, pipenv
- C++: CMake, Make, Bazel
- JavaScript: npm (Node.js Package Manager), Yarn
- C#: MSBuild (Microsoft Build Tools), NuGet
- PHP: Composer
- Swift: Swift Package Manager (SwiftPM)
- Kotlin: Gradle (también se usa para Android)
- Ruby: RubyGems, Bundler
- Go: Go Modules