

INSTITUTO TECNOLÓGICO METROPOLITANO

Materia: Despliegue de Software

Taller 2 - Entregable

Integrantes:

Federico Rodriguez

Daihana Mora

Sofia Ortiz

Stefany Builes

Alejandra Alvarez

ENTREGABLE - TALLER 2

1. ¿Cómo se abordó el proceso de despliegue?

Se instala Docker

Se crea el proyecto ASP.net en Visual Studio

Se verifica que esté mapeado el Docker file.

Se ejecuta el proyecto.

Se activan los kubernetes desde Docker Desktop.

En el proyecto se exponen los servicios por el puerto 80 y se quitan los ifs del Swagger.

En gitbash nos dirigimos a nuestro archivo de docker en el proyecto y creamos nuestra imagen a partir del archivo docker del proyecto.

Construimos nuestro archivo .yaml y lo llamamos deployment.yaml.

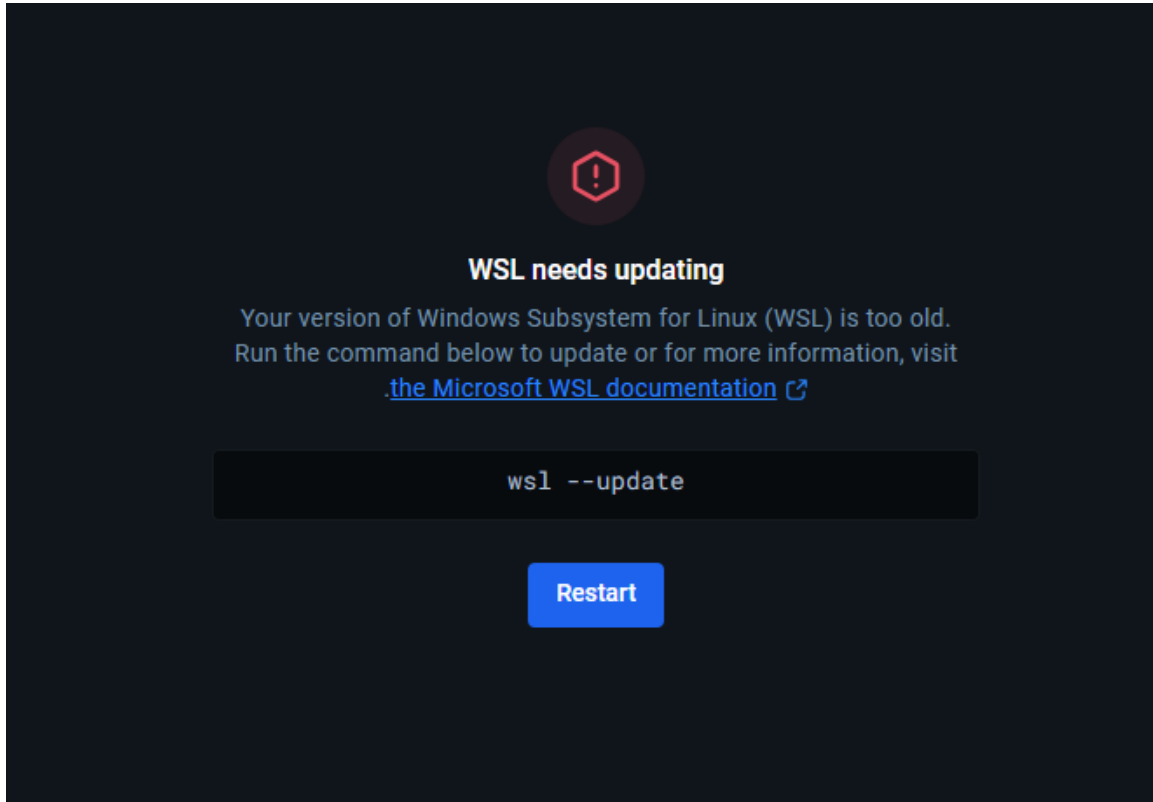
Desplegamos la api en nuestro kubernetes.

Se listan los pods corriendo en el kubernetes.

En el navegador se cargan los servicios expuestos por kubernetes.

2. ¿Qué errores encontramos y cómo los resolvimos?

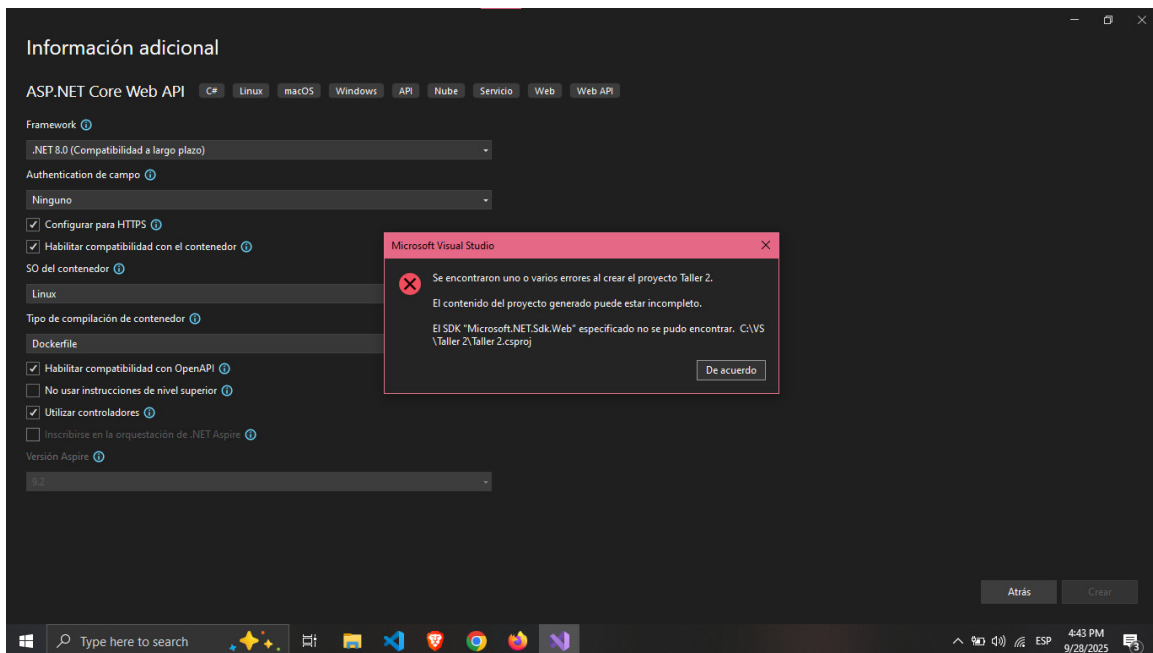
Al momento de instalar docker se requiere la actualización de WSL.



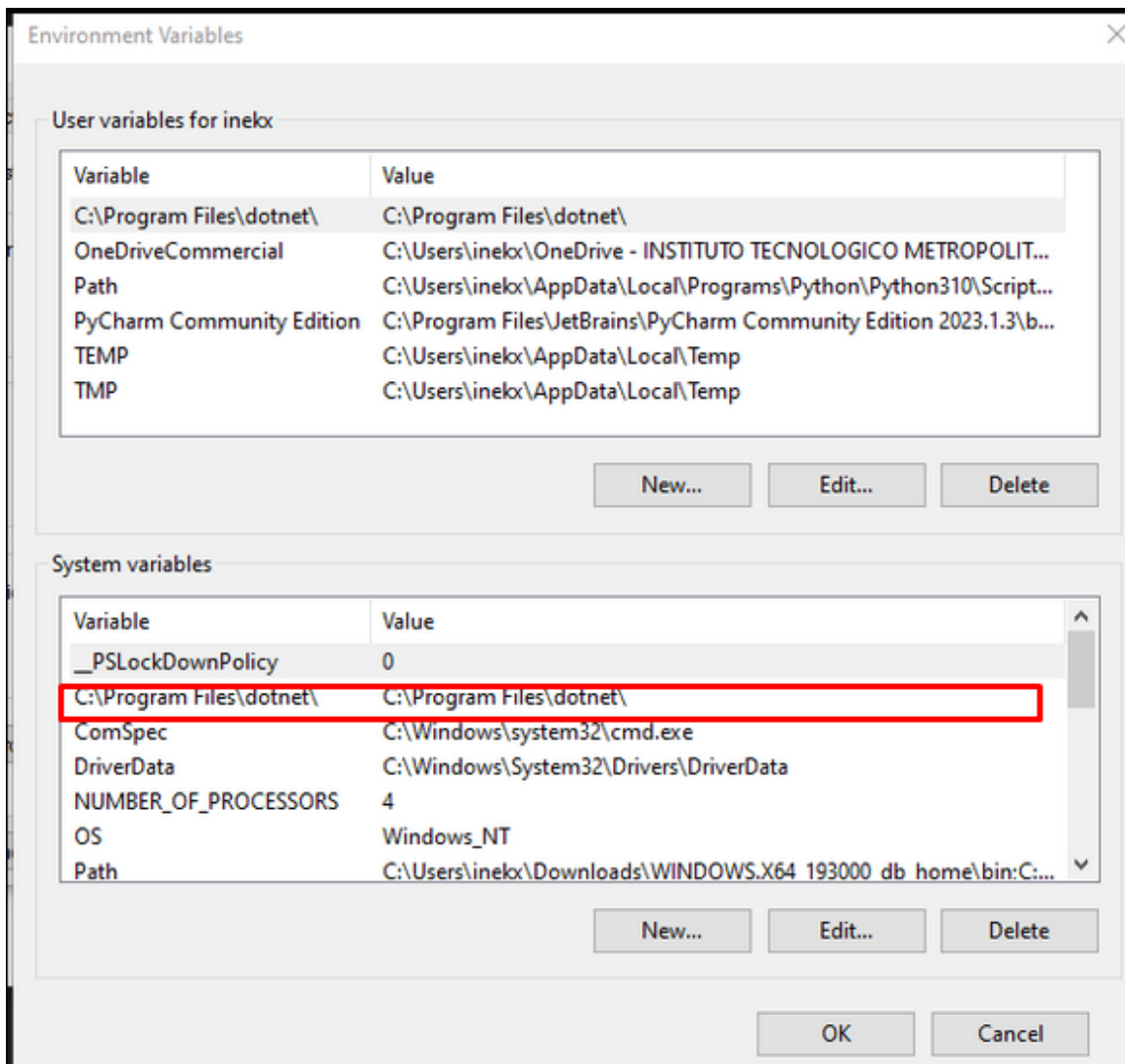
Solución: Se abre power shell y se ejecuta el comando `wsl --update` para luego utilizar el comando `wsl --version` para verificar.

El documento YAML debe ir con minúsculas porque sino, no carga la API en el navegador.

Se presentó un problema por parte de la compañera Sofía Ortiz al momento de crear el web api en Visual Studio por problemas de Net 8.0.



Se intentó dar solución reinstalando Visual Studio, descargando Net 8.0 por medio del instalador de visual studio y de internet, agregando la instalación a las variables de entorno del equipo y no funcionó.



```

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\inekx> where dotnet
PS C:\Users\inekx> ^C
PS C:\Users\inekx> dotnet --info

Host:
  Version:      9.0.8
  Architecture: x86
  Commit:       aae90fa090
  RID:          win-x86

.NET SDKs installed:
  No SDKs were found.

.NET runtimes installed:
  Microsoft.AspNetCore.App 5.0.6 [C:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 8.0.19 [C:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.AspNetCore.App 9.0.8 [C:\Program Files (x86)\dotnet\shared\Microsoft.AspNetCore.App]
  Microsoft.NETCore.App 5.0.6 [C:\Program Files (x86)\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 8.0.19 [C:\Program Files (x86)\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.NETCore.App 9.0.8 [C:\Program Files (x86)\dotnet\shared\Microsoft.NETCore.App]
  Microsoft.WindowsDesktop.App 8.0.19 [C:\Program Files (x86)\dotnet\shared\Microsoft.WindowsDesktop.App]
  Microsoft.WindowsDesktop.App 9.0.8 [C:\Program Files (x86)\dotnet\shared\Microsoft.WindowsDesktop.App]

Other architectures found:
  x64 [C:\Program Files\dotnet]
    registered at [HKLM\SOFTWARE\dotnet\Setup\InstalledVersions\x64\InstallLocation]

Environment variables:
  Not set

global.json file:
  Not found

```

```

ALEJANDRILAPTOP-SFLRNBQJ MINGW64 /c:/Proyectos/ITM/WebApplication1/WebApplication1
$ docker build -t webapplication1-api:local .
[+] Building 22.1s (15/17) docker:desktop-linux
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 1.54kB 0.0s
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:8.0 0.9s
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:8.0 0.4s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [build 1/7] FROM mcr.microsoft.com/dotnet/sdk:8.0sha256:3cef19377b2 13.5s
=> => resolve mcr.microsoft.com/dotnet/sdk:8.0sha256:3cef19377b2ef2a017 0.0s
=> => sha256:d98d17a5531938676b94521e26c99ea7cda25ea 16.98MB / 16.98MB 3.0s
=> => sha256:40956778c3851952134f441b55fd547edfcd3727ec3 2.63kB / 2.63kB 0.7s
=> => sha256:744ee46f25dfcfell008b8730e5797d36322 177.34MB / 177.34MB 10.7s
=> => sha256:46ca3aca6a9f4d8499b19ee6313f4cd3a883de6a8 30.89MB / 30.89MB 3.9s
=> => extracting sha256:46ca3aca6a9f4d8499b19ee6313f4cd3a883de6a81fb646e 0.6s
=> => extracting sha256:744ee46f25dfcfell008b8730e5797d363226d6f0ec82c 2.5s
=> => extracting sha256:40956778c3851952134f441b55fd547edfcd3727ec3650bd 0.0s
=> => extracting sha256:d98d17a5531938676b94521e26c99ea7cda25ea11e63 0.3s
[base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:8.0sha256:e88f90b6d9f4e9 0.0s
=> => resolve mcr.microsoft.com/dotnet/aspnet:8.0sha256:e88f90b6d9f4e9 0.0s
=> [internal] load build context 0.4s
=> => transferring context: 3.08MB 0.4s
=> CACHED [base 2/2] WORKDIR /app 0.0s
=> CACHED [final 1/2] WORKDIR /app 0.0s
=> [build 2/7] WORKDIR /src 0.2s
=> [build 3/7] COPY [webapplication1.csproj, webapplication1/] 0.0s
=> [build 4/7] RUN dotnet restore "/webApplication1/webApplication1.csp 4.4s
=> [build 5/7] COPY . 0.0s
=> [build 6/7] WORKDIR /src/webApplication1 0.0s
=> ERROR [build 7/7] RUN dotnet build "/webApplication1.csproj" -c Release 2.9s
-----
> [build 7/7] RUN dotnet build "/webApplication1.csproj" -c Release -o /app/bu
ild:
0.80s Determining projects to restore...
1.100 All projects are up-to-date for restore.
2.881 CSC : error CS5001: Program does not contain a static 'Main' method suitab
le for an entry point [/src/WebApplication1/WebApplication1.csproj]
2.889
2.889 Build FAILED.
2.889
2.889 CSC : error CS5001: Program does not contain a static 'Main' method suitab
le for an entry point [/src/WebApplication1/WebApplication1.csproj]
2.889 0 Warning(s)
2.889 1 Error(s)
2.890
2.890 Time Elapsed 00:00:02.38
-----
Dockerfile:22
-----
20 | COPY
21 | WORKDIR "/src/webApplication1"
22 | >>> RUN dotnet build "/webApplication1.csproj" -c $BUILD_CONFIGURATION -
o /app/build
23 |
24 | # Esta fase se usa para publicar el proyecto de servicio que se copia
ra en la fase final.
-----
ERROR: failed to build: failed to solve: process "/bin/sh -c dotnet build "/we
bApplication1.csproj" -c $BUILD_CONFIGURATION -o /app/build" did not complete s
uccessfully: exit code: 1

```

3. ¿Cómo se distribuyeron las responsabilidades en el equipo?

Federico Rodriguez

Creación del repositorio con el readme.

Parte uno: Creación de web API, uso de Docker.

Daihana Mora

Creación del repositorio con el readme.

Parte uno: Creación de web API, uso de Docker.

Stefany Builes

Creación del repositorio con el readme.

Parte uno: Creación de web API, uso de Docker.

Parte dos: Creación de imagen, construcción de archivo YAML, Despliegue de API en kubernetes.

Creación de video.

Sofía Ortiz

Creación de entregables.