

Trabajo Práctico 6 - Construcción de Imágenes de Docker

1- Conceptos de Dockerfiles

FROM: La instrucción FROM es el punto de partida para una imagen de Docker. La elección de una imagen base adecuada es crucial para garantizar que tengamos el entorno necesario para nuestras aplicaciones. Dependiendo de nuestros requisitos, podemos optar por imágenes de sistemas operativos base o imágenes con software específico preinstalado.

RUN: Con la instrucción RUN, podemos realizar tareas de configuración y preparación en nuestra imagen. Esto es fundamental para instalar software adicional, configurar variables de entorno o realizar cualquier acción necesaria durante la construcción de la imagen.

ADD y COPY: Estas instrucciones nos permiten copiar archivos y directorios locales en nuestra imagen. ADD también puede manejar la extracción de archivos desde URL. Elegir entre ADD y COPY depende de nuestras necesidades, y estas instrucciones son esenciales para incluir recursos en la imagen.

EXPOSE: Al usar la instrucción EXPOSE, podemos especificar en qué puertos esperamos que escuche nuestro contenedor. Aunque esta instrucción no publica automáticamente los puertos, es fundamental para documentar y comunicar los puertos utilizados por nuestra aplicación.

CMD y ENTRYPOINT: Estas instrucciones definen comandos o aplicaciones predeterminadas para ejecutar cuando se inicia un contenedor. CMD es más flexible y se puede anular fácilmente al ejecutar el contenedor con un comando diferente. En cambio, ENTRYPOINT se utiliza cuando deseamos definir un punto de entrada fijo que no se anulará tan fácilmente.

2- Generar imagen de docker

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ docker build -t miproyectowebapi .
[+] Building 50.9s (13/13) FINISHED                                docker:desktop-linux
=> [internal] load build definition from dockerfile                0.1s
=> => transferring dockerfile: 508B                                0.0s
=> [internal] load .dockerignore                                  0.1s
=> => transferring context: 2B                                       0.0s
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:7.0  2.4s
=> [build 1/8] FROM mcr.microsoft.com/dotnet/sdk:7.0@sha256:d0755fbf1ed34aecd79c127fd1dd01b80587 23.3s
=> => resolve mcr.microsoft.com/dotnet/sdk:7.0@sha256:d0755fbf1ed34aecd79c127fd1dd01b80587acdb8ff 0.1s
=> => sha256:d0755fbf1ed34aecd79c127fd1dd01b80587acdb8ff50e5a68d1adf8b076922b 1.79kB / 1.79kB 0.0s
=> => sha256:fbf7eed1efcc4cd2571e40f0c34949a09474f4a96594b0501b4e24b7f0767bcb 2.01kB / 2.01kB 0.0s
=> => sha256:bd36eac352ddb2d27a63f4714b9063fc8c7e9c615e430006dbeb7e3448cd5780 5.28kB / 5.28kB 0.0s
=> => sha256:e67fdae3559346105027c63e7fb032bba57e62b1fe9f2da23e6fdbb56384e00b 31.42MB / 31.42MB 1.3s
=> => sha256:0ab66724116f089079aab4d1403ce98586595f13ca4b4163005ec3733a5dab94 14.97MB / 14.97MB 2.7s
=> => sha256:14ccddebb1bc93dd113768cf39770236e66888c5d08fc5a554d2e47a0fe930e 32.46MB / 32.46MB 3.8s
=> => sha256:5e265b51b431d80b30bffa2e2f867af78d559418b418ea7b3edffaead2221b244 156B / 156B 1.6s
=> => extracting sha256:e67fdae3559346105027c63e7fb032bba57e62b1fe9f2da23e6fdbb56384e00b 7.8s
=> => sha256:3bda6efdfc5b92b1a9760092efa9eb1a4f50a0374789af843c6842acaf848b8f 10.12MB / 10.12MB 2.6s
=> => sha256:9a6d473c86f69f795e0ff2b6014ed32f2c9b35e743068b430a4adf1c1448176b 25.38MB / 25.38MB 8.8s
=> => sha256:5464f27bea88b4d6d19d0c0e15a62a967f65d74b9753611d0a174526bc08b1da 180.99MB / 180.99MB 8.7s
=> => sha256:3cc0e9253374a2ce8dd07c45550ebb8a207cc9898aa397219369e970962f7ab6 13.98MB / 13.98MB 7.1s
=> => extracting sha256:0ab66724116f089079aab4d1403ce98586595f13ca4b4163005ec3733a5dab94 1.5s
=> => extracting sha256:14ccddebb1bc93dd113768cf39770236e66888c5d08fc5a554d2e47a0fe930e 2.1s
=> => extracting sha256:5e265b51b431d80b30bffa2e2f867af78d559418b418ea7b3edffaead2221b244 0.0s
=> => extracting sha256:3bda6efdfc5b92b1a9760092efa9eb1a4f50a0374789af843c6842acaf848b8f 0.7s
=> => extracting sha256:9a6d473c86f69f795e0ff2b6014ed32f2c9b35e743068b430a4adf1c1448176b 3.3s
=> => extracting sha256:5464f27bea88b4d6d19d0c0e15a62a967f65d74b9753611d0a174526bc08b1da 4.6s
=> => extracting sha256:3cc0e9253374a2ce8dd07c45550ebb8a207cc9898aa397219369e970962f7ab6 0.4s
=> [internal] load build context                                  0.2s
=> => transferring context: 4.78MB                                   0.2s
=> [build 2/8] WORKDIR /src                                       0.1s
=> [build 3/8] COPY [MiProyectoWebAPI.csproj, .]                 0.0s
=> [build 4/8] RUN dotnet restore "./MiProyectoWebAPI.csproj"    11.8s
=> [build 5/8] COPY . .                                           0.1s
=> [build 6/8] WORKDIR /src/.                                     0.0s
=> [build 7/8] RUN dotnet build "MiProyectoWebAPI.csproj" -c Release -o /app/build 9.4s
=> [build 8/8] RUN dotnet publish "MiProyectoWebAPI.csproj" -c Release -o /app/publish /p:UseAppH 2.9s
=> exporting to image                                             0.4s
=> => exporting layers                                             0.4s
=> => writing image sha256:e790483ad63c23427718f2a0c4ddba0f354489fdc1607750581e2b4bda175d93 0.0s
=> => naming to docker.io/library/miprojectowebapi               0.0s
```

What's Next?

View a summary of image vulnerabilities and recommendations → [docker scout quickview](#)

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ docker run -p 8080:80 -it --rm miprojectowebapi
[info: Microsoft.Hosting.Lifetime[14]
Now listening on: http://localhost:5000
[info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
[info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Production
[info: Microsoft.Hosting.Lifetime[0]
Content root path: /src
```

```

root@23cf3b59a2ff:/# cd app
root@23cf3b59a2ff:/app# ls
Microsoft.AspNetCore.OpenApi.dll  Swashbuckle.AspNetCore.Swagger.dll
Microsoft.OpenApi.dll             Swashbuckle.AspNetCore.SwaggerGen.dll
SimpleWebAPI.deps.json            Swashbuckle.AspNetCore.SwaggerUI.dll
SimpleWebAPI.dll                  appsettings.Development.json
SimpleWebAPI.pdb                  appsettings.json
SimpleWebAPI.runtimeconfig.json  web.config
root@23cf3b59a2ff:/app# pwd
/app
root@23cf3b59a2ff:/app#

```

3- Dockerfiles Multi Etapas

```

tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP5/MiProyectoWebAPI$ docker build -t mip
rojectwebapi .
[+] Building 11.6s (18/18) FINISHED                                docker:desktop-linux
=> [internal] load .dockerignore                                  0.0s
=> => transferring context: 2B                                     0.0s
=> [internal] load build definition from dockerfile               0.0s
=> => transferring dockerfile: 581B                                0.0s
=> [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:7.0 2.1s
=> [internal] load metadata for mcr.microsoft.com/dotnet/sdk:7.0  1.2s
=> [build 1/7] FROM mcr.microsoft.com/dotnet/sdk:7.0@sha256:d0755fbf1ed34aec9c127fd1dd01b80587a 0.0s
=> [internal] load build context                                  0.0s
=> => transferring context: 4.03kB                                  0.0s
=> [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:7.0@sha256:77ea66eb200fa06d2929cc937b75724bb2f 0.1s
=> => resolve mcr.microsoft.com/dotnet/aspnet:7.0@sha256:77ea66eb200fa06d2929cc937b75724bb2f5cdf2 0.0s
=> => sha256:77ea66eb200fa06d2929cc937b75724bb2f5cdf2bcf6dc6aaf0851bb75bb631f 1.79kB / 1.79kB 0.0s
=> => sha256:4cecefeb3ad7531e38b0915ad8f4e297a888d8260634162f899e7bc83f25e2ef 1.37kB / 1.37kB 0.0s
=> => sha256:b06452c8ee916a8c0a9863aad05f86b728ec72a1e43bcbfd8e7bd304953285a1 2.36kB / 2.36kB 0.0s
=> CACHED [build 2/7] WORKDIR /src                                0.0s
=> CACHED [build 3/7] COPY [MiProyectoWebAPI.csproj, .]          0.0s
=> CACHED [build 4/7] RUN dotnet restore "/MiProyectoWebAPI.csproj" 0.0s
=> [build 5/7] COPY . .                                           0.1s
=> [build 6/7] WORKDIR /src/.                                     0.0s
=> [base 2/2] WORKDIR /app                                         0.0s
=> [final 1/2] WORKDIR /app                                        0.1s
=> [build 7/7] RUN dotnet build "MiProyectoWebAPI.csproj" -c Release -o /app/build 6.9s
=> [publish 1/1] RUN dotnet publish "MiProyectoWebAPI.csproj" -c Release -o /app/publish /p:UseAp 2.2s
=> [final 2/2] COPY --from=publish /app/publish .                 0.1s
=> exporting to image                                             0.1s
=> => exporting layers                                           0.1s
=> => writing image sha256:0c992654e31ccf76dd25f47d25f6f1be556fb3bf4fa595f3792858f14bf9f38f 0.0s
=> => naming to docker.io/library/miprojectwebapi                0.0s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview

```

La primera etapa, llamada base, es similar a la etapa anterior. Utiliza la imagen `mcr.microsoft.com/dotnet/aspnet:7.0` como base, establece el directorio de trabajo en `/app` y expone el puerto 80.

La segunda etapa, llamada build, también es similar a la etapa anterior. Utiliza la imagen `mcr.microsoft.com/dotnet/sdk:7.0` como base y se encarga de la compilación del proyecto. Establece el directorio de trabajo en `/src`, copia el archivo de proyecto `MiProyectoWebAPI.csproj`, restaura las dependencias con `dotnet restore`, copia todos los archivos del proyecto y luego compila el proyecto en la carpeta `/app/build`.

La tercera etapa, llamada publish, se basa en la etapa build y se encarga de la publicación del proyecto. Utiliza dotnet publish para generar los archivos de salida en la carpeta /app/publish. Se establece la opción /p:UseAppHost=false para indicar que no se debe usar el host de la aplicación.

La cuarta etapa, llamada final, es similar a la etapa anterior y se basa en la etapa base. Configura el directorio de trabajo en /app, copia los archivos generados en la etapa de publish en la carpeta /app y configura el punto de entrada para la aplicación utilizando ENTRYPOINT.

```
root@20c77a72293d:/# ls
app  boot  etc   lib   media  opt   root  sbin  sys  usr
bin  dev   home  lib64  mnt   proc  run   srv   tmp  var
root@20c77a72293d:/# cd app/
root@20c77a72293d:/app# ls
MiProyectoWebAPI.deps.json      Microsoft.OpenApi.dll          appsettings.Development.json
MiProyectoWebAPI.dll            Newtonsoft.Json.dll            appsettings.json
MiProyectoWebAPI.pdb            Swashbuckle.AspNetCore.Swagger.dll  web.config
MiProyectoWebAPI.runtimeconfig.json  Swashbuckle.AspNetCore.SwaggerGen.dll
Microsoft.AspNetCore.OpenApi.dll     Swashbuckle.AspNetCore.SwaggerUI.dll
root@20c77a72293d:/app#
```

4- Imagen para aplicación web en Nodejs

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP6/trabajo-practico-06/nodejs-docker$ npx create-react-app my-app

Creating a new React app in /home/tincho/Documents/Ingenieria de Software III/TP6/trabajo-practico-06/nodejs-docker/my-app.

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

added 1000 packages in 54s

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

Success! Created my-app at /home/tincho/Documents/Ingenieria de Software III/TP6/trabajo-practico-06/nodejs-docker/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!

Note: the project was bootstrapped with an old unsupported version of tools.
Please update to Node >=14 and npm >=6 to get supported tools in new projects.
```

```

1 # Etapa 1: Imagen de construcción
2 FROM node:13.12.0-alpine as build
3 WORKDIR /app
4 COPY package*.json ./
5 RUN npm install
6 COPY . .
7 RUN npm run build
8
9 # Etapa 2: Imagen de producción
10 FROM node:13.12.0-alpine
11 WORKDIR /app
12 COPY --from=build /app .
13 EXPOSE 3000
14 CMD ["npm", "start"]

```

```

tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP6/trabajo-practico-06/nodejs-docker$ cd
my-app/
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP6/trabajo-practico-06/nodejs-docker/my-
app$ docker build -t test-node .
[+] Building 68.0s (13/13) FINISHED          docker:desktop-linux
=> [internal] load .dockerignore              0.0s
=> => transferring context: 2B                 0.0s
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 317B            0.0s
=> [internal] load metadata for docker.io/library/node:13.12.0-alpine 9.5s
=> [auth] library/node:pull token for registry-1.docker.io 0.0s
=> [build 1/6] FROM docker.io/library/node:13.12.0-alpine@sha256:cc85e728fab3827ada20a181ba280cae 5.7s
=> => resolve docker.io/library/node:13.12.0-alpine@sha256:cc85e728fab3827ada20a181ba280cae1f8b62 0.0s
=> => sha256:cc85e728fab3827ada20a181ba280cae1f8b625f256e2c86b9094d9bfe834766 1.19kB / 1.19kB 0.0s
=> => sha256:ed06820d0fb6f4711e0a6f50c9f147fb2596399866319e1bb3b0a52393c5615f 1.16kB / 1.16kB 0.0s
=> => sha256:483343d6c5f5d658963b7c16e4299247f765bef4025012457ee105481ea1afc1 6.77kB / 6.77kB 0.0s
=> => sha256:aad63a9339440e7c3e1fff2b988991b9bfb81280042fa7f39a5e327023056819 2.80MB / 2.80MB 0.7s
=> => sha256:a00bd932208e2de29cd2b7e0bab954325913d146401effb482fff3d8775aaeb 35.29MB / 35.29MB 1.9s
=> => sha256:c57f2c59b93778192e60e0e213949630f9ad30324736bbb0a71e12adf8a16d46 2.24MB / 2.24MB 0.5s
=> => sha256:f3446470f297e51c1e27f90f7ae9a94f1ee7b6c8e529aec83aa1a04ad70531c0 284B / 284B 0.8s
=> => extracting sha256:aad63a9339440e7c3e1fff2b988991b9bfb81280042fa7f39a5e327023056819 0.2s
=> => extracting sha256:a00bd932208e2de29cd2b7e0bab954325913d146401effb482fff3d8775aaeb 3.3s
=> => extracting sha256:c57f2c59b93778192e60e0e213949630f9ad30324736bbb0a71e12adf8a16d46 0.2s
=> => extracting sha256:f3446470f297e51c1e27f90f7ae9a94f1ee7b6c8e529aec83aa1a04ad70531c0 0.0s
=> [internal] load build context              3.9s
=> => transferring context: 95.31MB           3.6s
=> [build 2/6] WORKDIR /app                   0.2s
=> [build 3/6] COPY package*.json ./          0.1s
=> [build 4/6] RUN npm install                 38.4s
=> [build 5/6] COPY . .                      4.3s
=> [build 6/6] RUN npm run build               5.0s
=> [stage-1 3/3] COPY --from=build /app .      1.9s
=> exporting to image                         1.2s
=> => exporting layers                       1.2s
=> => writing image sha256:b10a63cbec84568846372b3d34fb8c9afb1145fecb1b12a8f81239680726d7b 0.0s
=> => naming to docker.io/library/test-node 0.0s

```

What's Next?

View a summary of image vulnerabilities and recommendations → [docker scout quickview](#)

```

tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP6/trabajo-practico-06/nodejs-docker/my-
app$ docker run -p 3000:3000 -d test-node
be3ed32d304d1a7378e9190c92b5c5db5fb624469db5f9d9f55d3dc0997b6430a

```



Welcome to React

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

5- Publicar la imagen en Docker Hub.

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP6/trabajo-practico-06/nodejs-docker/my-app$ docker push martinlescano/test-node:latest
The push refers to repository [docker.io/martinlescano/test-node]
880c9dd615f4: Pushed
112f8b639ff2: Pushed
65d358b7de11: Mounted from library/node
f97384e8ccbc: Mounted from library/node
d56e5e720148: Mounted from library/node
beee9f30bc1f: Mounted from library/node
latest: digest: sha256:80d916ab92c2d2a8c07387af192bd5b560d076324a38cec6d785ff727fb6ea88 size: 1576
```



`martinlescano/test-node` ☆

By [martinlescano](#) • Updated a minute ago

Image