

Taller Práctico: Creación y Despliegue de una Aplicación FastAPI con Docker

1. Preparación del Entorno de Desarrollo

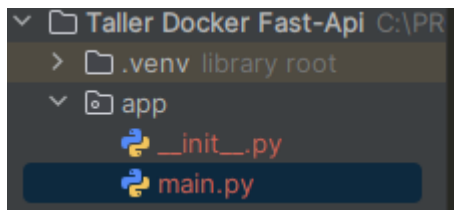
- Instale Docker en su sistema siguiendo las instrucciones proporcionadas en el curso o si desea utilice Play with Docker.
- Verifique la instalación ejecutando `docker --version` en la terminal.

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker --version
Docker version 26.0.0, build 2ae903e
```

2. Creación de una Aplicación FastAPI “Hello World” <https://fastapi.tiangolo.com>

<https://fastapi.tiangolo.com/deployment/docker/#create-the-fastapi-code>

- Cree un nuevo directorio para su proyecto y, dentro de este, un archivo `main.py` que contenga una aplicación FastAPI básica que responda con “Hello World” en la ruta raíz.

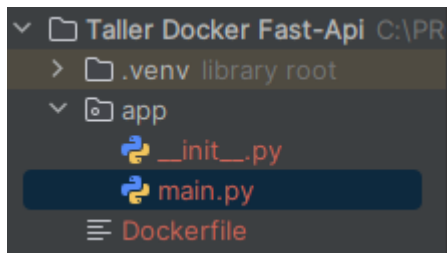


```
1  from typing import Union
2
3  from fastapi import FastAPI
4
5  app = FastAPI()
6
7
8  @app.get("/")
9  def read_root():
10     return {"Hello": "World"}
11
```

3. Construcción de un Dockerfile para la Aplicación

- En el mismo directorio del proyecto, cree un archivo Dockerfile que:

- Utilice una imagen base de Python adecuada.
- Instale las dependencias necesarias.
- Copie el código de la aplicación al contenedor.
- Configure el contenedor para ejecutar la aplicación utilizando un servidor ASGI como Uvicorn.



```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> pip install -r requirements.txt
Collecting fastapi<0.114.0,>=0.113.0 (from fastapi[standard]<0.114.0,>=0.113.0->-r requirements.txt (line 1))
  Downloading fastapi-0.113.0-py3-none-any.whl.metadata (27 kB)
Collecting pydantic<3.0.0,>=2.7.0 (from -r requirements.txt (line 2))
  Using cached pydantic-2.10.6-py3-none-any.whl.metadata (30 kB)
Collecting starlette<0.39.0,>=0.37.2 (from fastapi<0.114.0,>=0.113.0->fastapi[standard]<0.114.0,>=0.113.0->-r requirements.txt (line 1))
  Downloading starlette-0.38.6-py3-none-any.whl.metadata (6.0 kB)
Requirement already satisfied: typing-extensions>=4.8.0 in c:\users\yosel\appdata\local\packages\pythonsoftwarefoundation.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from fastapi<0.114.0,>=0.113.0->fastapi[standard]<0.114.0,>=0.113.0->-r requirements.txt (line 1)) (4.8.0)
Collecting annotated-types>=0.6.0 (from pydantic<3.0.0,>=2.7.0->-r requirements.txt (line 2))
```

1	<code>fastapi[standard]>=0.113.0,<0.114.0</code>
2	<code>pydantic>=2.7.0,<3.0.0</code>

```

1
2 FROM python:3.9
3
4
5 WORKDIR /code
6
7
8 COPY ./requirements.txt /code/requirements.txt
9
10
11 RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt
12
13 
14 COPY ./app /code/app
15
16
17 CMD ["fastapi", "run", "app/main.py", "--port", "80"]

```

4. Construcción de la Imagen Docker

- Utilice el comando `docker build` para construir la imagen Docker a partir del Dockerfile, asignándole un nombre que incluya su **nombre de usuario institucional**.

```

PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-Api> docker build -t bjacuna1 .
[+] Building 1.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 284B
=> [internal] load metadata for docker.io/library/python:3.9
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/5] FROM docker.io/library/python:3.9@sha256:c17c71e1f5f258803a6b7c391f8013adbf84285af54c2a811de4a5a1ac5a8676
=> [internal] load build context
=> => transferring context: 123B
=> CACHED [2/5] WORKDIR /code
=> CACHED [3/5] COPY ./requirements.txt /code/requirements.txt
=> CACHED [4/5] RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt
=> CACHED [5/5] COPY ./app /code/app
=> exporting to image
=> => exporting layers
=> => writing image sha256:0f0ea8219d070470b4652049b86ba76f94886deb1e8ee6d44d2140066d50a632
=> => naming to docker.io/library/bjacuna1

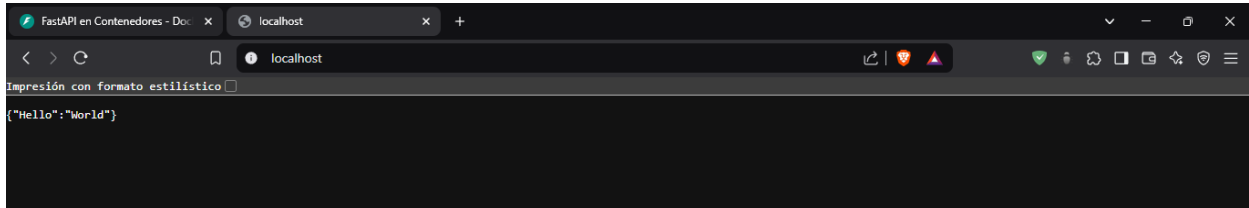
```

5. Ejecución y Prueba del Contenedor

- Inicie un contenedor basado en la imagen creada utilizando docker run.

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker run -d --name my-container -p 80:80 bjacuna166b68187b00e05cc6ecf8fb2535da42ad075abb11ae7697ac63360dd502091ff
```

- Verifique que la aplicación responde correctamente accediendo en su navegador a <http://localhost:8000> o al URL que le corresponda.



6. Gestión de Contenedores

- Liste los contenedores en ejecución y detenga el contenedor de la aplicación.

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
66b68187b00e	bjacuna1	"fastapi run app/mai..."	About a minute ago	Up About a minute	0.0.0.0:80->80/tcp	my-container

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker stop 66
66
```

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

- Elimine el contenedor detenido.

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker rm 66
66
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

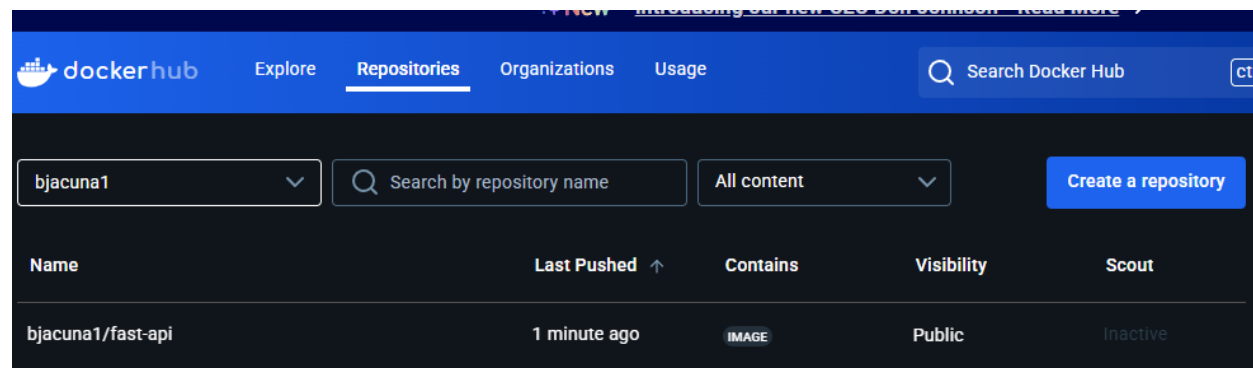
7. Publicación de la Imagen en Docker Hub

- Etiquete la imagen Docker con su nombre de usuario de Docker Hub.
- Inicie sesión en Docker Hub desde la terminal.

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker login -u bjacuna1
Password:
Login Succeeded
```

- Suba la imagen al repositorio de Docker Hub utilizando docker push.

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker push bjacuna1/fast-api:latest
The push refers to repository [docker.io/bjacuna1/fast-api]
52a3ae0a5afd: Pushed
032ea5c7c2b7: Pushed
00877e003334: Pushed
4c84eee35ab6: Pushed
60a159600b22: Mounted from library/python
ee959616fc20: Mounted from library/python
d0e85779261a: Mounted from library/python
dafb8aed9f7f: Mounted from library/python
41d4dc7516bb: Mounted from library/python
c0f51bbdc37d: Mounted from library/python
91b542912d12: Mounted from library/python
latest: digest: sha256:0a5674ad69e422d6e42033c8c2f63a8a0653a670d618a9e68979fc74c7e69864 size: 2628
```



Name	Last Pushed	Contains	Visibility	Scout
bjacuna1/fast-api	1 minute ago	IMAGE	Public	Inactive

8. Documentación y Control de Versiones

- Cree un repositorio en GitHub para su proyecto.
- Suba el código fuente, el Dockerfile y cualquier otro archivo relevante al repositorio.
- Incluya en el repositorio capturas de pantalla o registros que evidencien la ejecución exitosa de la aplicación y la publicación de la imagen en Docker Hub.

```

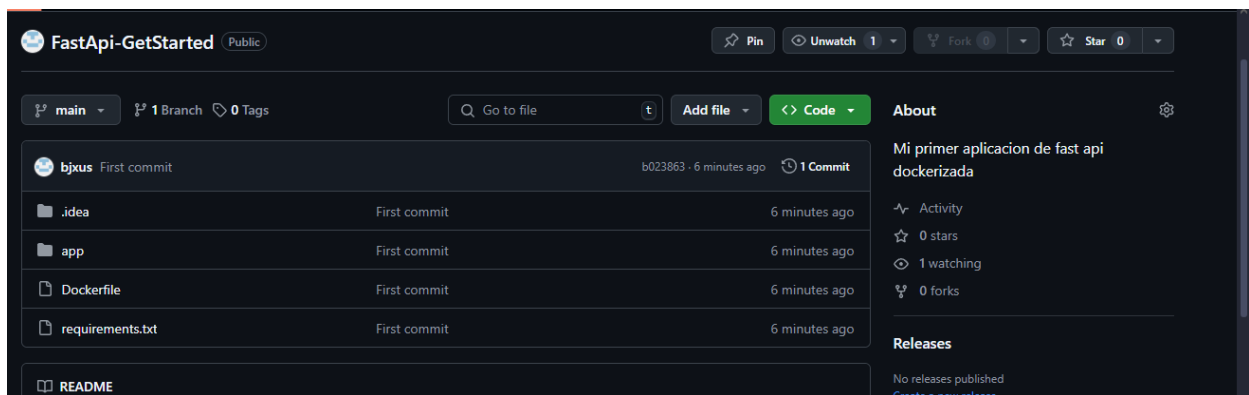
yose1@BRAYAN MINGW64 /c/PROGRAMACION/PYTHON/Taller Docker Fast-API (master)
$ git add .

yose1@BRAYAN MINGW64 /c/PROGRAMACION/PYTHON/Taller Docker Fast-API (master)
$ git commit -m "First commit"
[master (root-commit) b023863] First commit
 9 files changed, 106 insertions(+)
 create mode 100644 .idea/Taller Docker Fast-API.iml
 create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
 create mode 100644 .idea/modules.xml
 create mode 100644 .idea/vcs.xml
 create mode 100644 .idea/workspace.xml
 create mode 100644 Dockerfile
 create mode 100644 app/__init__.py
 create mode 100644 app/main.py
 create mode 100644 requirements.txt

yose1@BRAYAN MINGW64 /c/PROGRAMACION/PYTHON/Taller Docker Fast-API (master)
$ git branch -m master main

yose1@BRAYAN MINGW64 /c/PROGRAMACION/PYTHON/Taller Docker Fast-API (main)
$ git remote add origin https://github.com/bjxus/Fast-API-GetStarted.git

```



9. Limpieza de Imágenes y Contenedores

- Elimine las imágenes y contenedores innecesarios de su sistema para liberar espacio, utilizando comandos como `docker rmi` y `docker container prune`.

```

PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker rmi bjacuna1/fast-api bjacuna1 bjacuna fast-api myimage
Untagged: bjacuna1/fast-api:latest
Untagged: bjacuna1/fast-api@sha256:0a5674ad69e422d6e42033c8c2f63a8a0653a670d618a9e68979fc7c74c7e69864
Untagged: bjacuna1:latest
Untagged: bjacuna:latest
Untagged: fast-api:latest
Untagged: myimage:latest
Deleted: sha256:0f0ea8219d070470b4652049b86ba76f94886deb1e8ee6d44d2140066d50a632

```

```
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker image prune
WARNING! This will remove all dangling images.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
PS C:\PROGRAMACION\PYTHON\Taller Docker Fast-API> docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
```

10. Entrega del Proyecto

- Proporcione el enlace al repositorio de GitHub que contiene el proyecto completo al correo scadena@unicesar.edu.co.
- Incluya en el repositorio un archivo **README.md** con instrucciones claras sobre cómo construir y ejecutar la imagen Docker de la aplicación.