




Proyecto de Sistemas Operativos – Sesión 1: ¡Al Contenedor!

<input checked="" type="checkbox"/> Completado	<input type="checkbox"/>
 Cursos	 <u>Sistemas Operativos</u>
 Estado	Sin empezar

Integrantes


Nombre	Cédula
Dylan Elizondo Alvarado	504610652
Luis David Salgado Gámez	208670670

Comandos de Docker Utilizados:

Comando para construir la imagen de la aplicación:

```
docker build . -t proyecto2_s1
```



El comando se uso en la carpeta donde esta el Dockerfile, en caso de querer construir la imagen desde otro punto en lugar  se debe poner la ruta donde se encuentra el archivo Dockerfile.

```

PS C:\Users\dylal\Documents\Ciclo_I_2025\Sistemas Operativos\Proyecto2_S0\src> docker build . -t proyecto2_s1
[+] Building 31.8s (12/12) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 694B
=> [internal] load metadata for docker.io/library/python:3.12-slim
=> [auth] library/python:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 28
=> [internal] load build context
=> => transferring context: 1.05kB
=> [1/6] FROM docker.io/library/python:3.12-slim@sha256:0175d8ff0ad1dc8ceca4bcf311c3e47d08807a940959fa1cdbcfa87841883a1
=> resolve docker.io/library/python:3.12-slim@sha256:0175d8ff0ad1dc8ceca4bcf311c3e47d08807a940959fa1cdbcfa87841883a1
=> sha256:2085f9e6ccae82a287654f932d6c768f0f9084a8f567ea03979491279dbd06c 248B / 248B
=> sha256:9e1259a465a5cb12ab4d313f1aacea70ac0462fa9387e76fdb5369810f7401f6 13.66MB / 13.66MB
=> sha256:ce1945dade9f42ba70b6318fa2e3862b64c774d98efc56b2f552a36f7dd58b82 3.51MB / 3.51MB
=> sha256:61320b01ae5e0798393ef25f2dc72faf43703e60ba089b07d7170acbabfb8f62 28.23MB / 28.23MB
=> extracting sha256:61320b01ae5e0798393ef25f2dc72faf43703e60ba089b07d7170acbabfb8f62
=> extracting sha256:ce1945dade9f42ba70b6318fa2e3862b64c774d98efc56b2f552a36f7dd58b82
=> extracting sha256:9e1259a465a5cb12ab4d313f1aacea70ac0462fa9387e76fdb5369810f7401f6
=> extracting sha256:2085f9e6ccae82a287654f932d6c768f0f9084a8f567ea03979491279dbd06c

```

Comando para construir el contenedor:

```
docker run --name proyecto2 -dp 5000:5000 -v "./app" -e WELCOME_MSG="¡Hola HotContainerz™ desde volumen!" proyecto2_s1
```



Este comando Lanza un contenedor llamado `proyecto2`, en segundo plano (`-d`), con:

- `p 5000:5000`: expone el puerto 5000 del contenedor al host.
- `v "./app"`: monta el código fuente local en `/app` (para que Flask vea los cambios).
- `e WELCOME_MSG=...`: pasa la variable de entorno `WELCOME_MSG`.
- `proyecto2_s1`: es la imagen usada para crear el contenedor.

Esto permite ver tu app en `http://localhost:5000` y cambiar el mensaje desde el entorno o el código sin reiniciar el contenedor.

Nota importante: al ejecutar este comando se debe estar en el directorio donde esta el código.

```
Windows PowerShell
PS D:\Tercero\I ciclo\operativos\proyecto2\Proyecto2_S0\src> docker run --name proyecto2 -dp 5000:5000 -v "C:\Tercero\I ciclo\operativos\proyecto2\Proyecto2_S0\src" -e WELCOME_MSG="¡Hola HotContainerz™ desde volumen!" proyecto2_s1
5fed39b1faced74e05a19c58360634276fd77c38dd2d07ab23284be36c74aac5
PS D:\Tercero\I ciclo\operativos\proyecto2\Proyecto2_S0\src>
```

Imágenes de evidencia desde el navegador:

Imagen al ejecutar el contenedor sin un mensaje como parámetro.

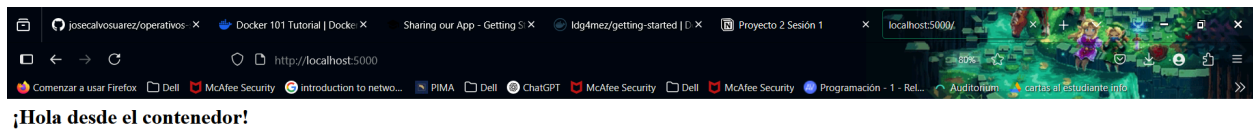


Imagen al ejecutar el contenedor enviando un mensaje como parámetro:

```
WELCOME_MSG="¡Hola HotContainerz™ desde volumen!"
```

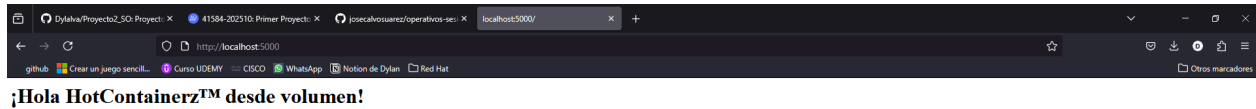


Imagen de Log al cambiar el código de `app.py`

```
PS C:\Users\dylan\Documents\Ciclo_I_2025\Sistemas Operativos\Proyecto2_S0\src> docker run -it --name proyecto2 -p 5000:5000 -v ".:/app" --e WELCOME_MSG="¡Hola HotContainerz™ desde volumen!" proyecto2_s1
* Serving Flask app 'app.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
* Restarting with stat
* Detected change in '/app/app.py', reloading
* Restarting with stat
```

Código del Dockerfile

```
FROM python:3.12-slim
```

```
WORKDIR /app
```

```
COPY requirements.txt .
```

```
RUN pip install --upgrade pip
```

```
RUN pip install --no-cache-dir -r requirements.txt
```

COPY . .

ENV FLASK_APP=app.py

ENV FLASK_ENV=development

EXPOSE 5000

CMD ["flask", "run", "--reload", "--host=0.0.0.0", "--port=5000"]