# AUTOMATIZACIÓN DE IMÁGENES CON DOCKER

**Santiago Maya Cortés**

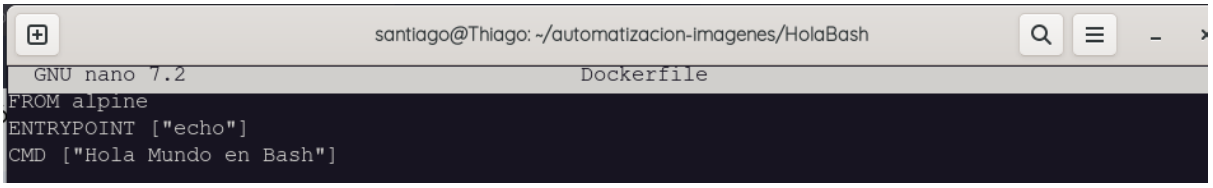
**Código: 1.092.850.328**


**Profesor: Carlos Eduardo Gomez Montoya**

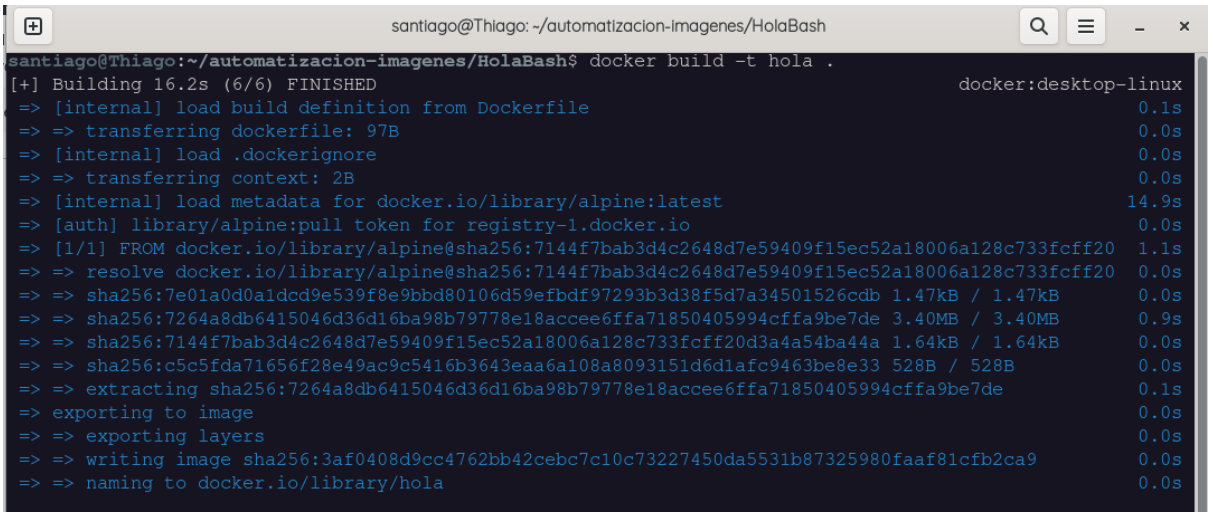
**Universidad del Quindío**
**Facultad de Ingeniería**
**Ingeniería de Sistemas y Computación**
**Computación en la Nube**
**Grupo: 01-D**
**Armenia, Quindío, 2023**

# Hola Mundo en Bash Shell

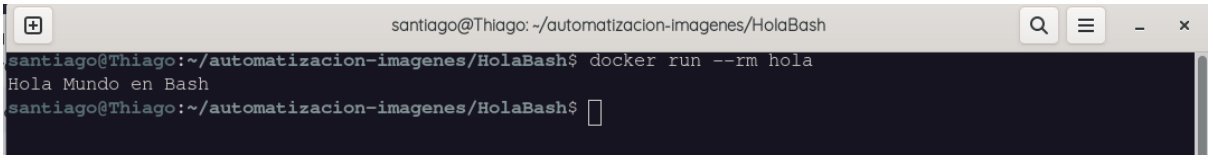
Archivo DockerFile




Construcción de la imagen hola



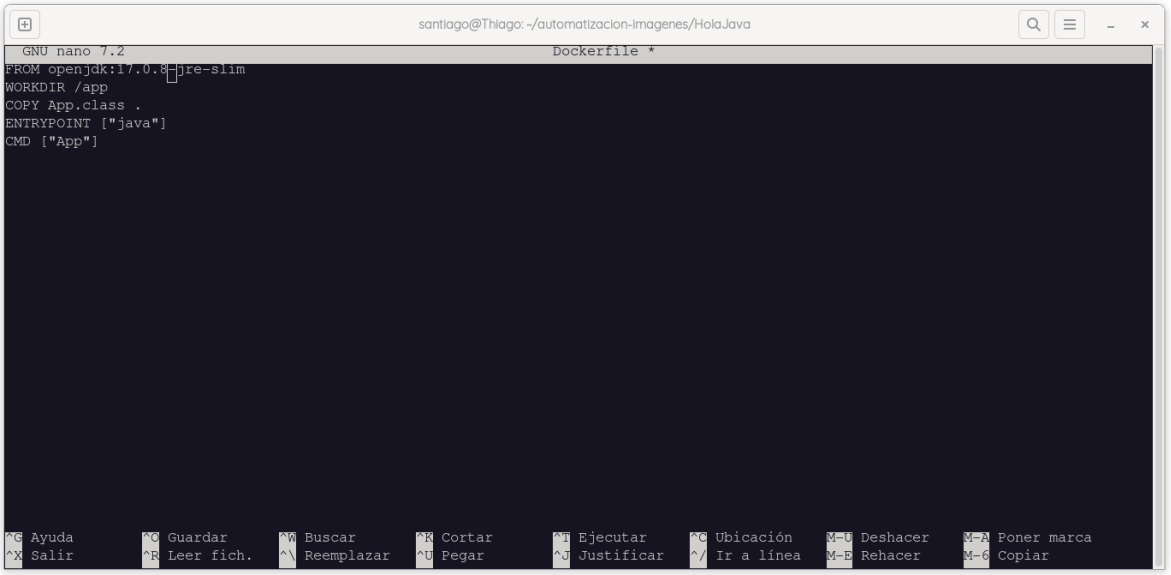

Creación del contenedor



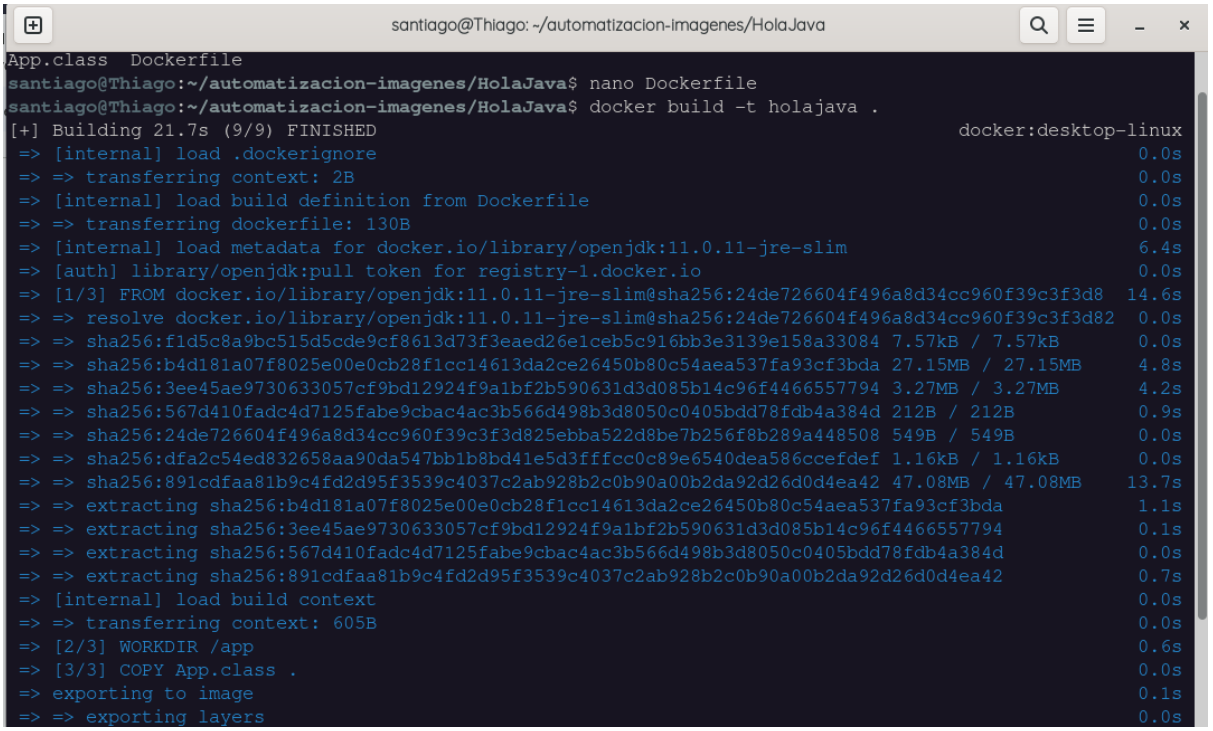
# Hola Mundo en Java

Aplicación JAVA que muestra un hola mundo

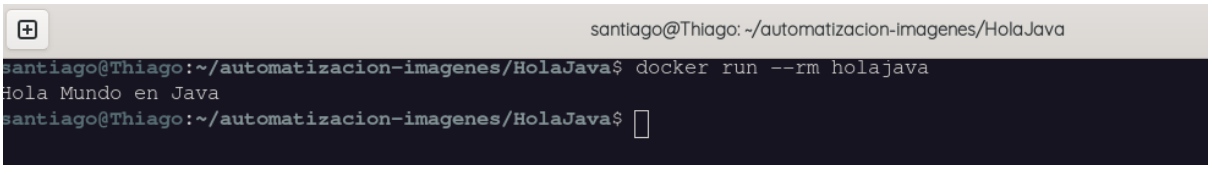

Archivo Dockerfile con el App.class(se utilizó el jdk 17 de java ya que en esa versión se compiló el programa)



Construcción de la imagen holajava



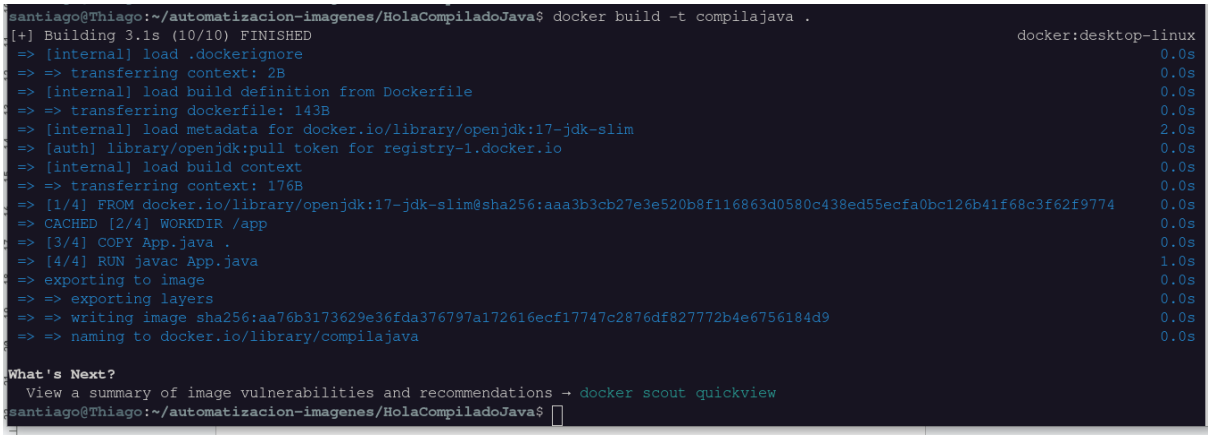

Creación del contenedor a partir de la imágen anterior



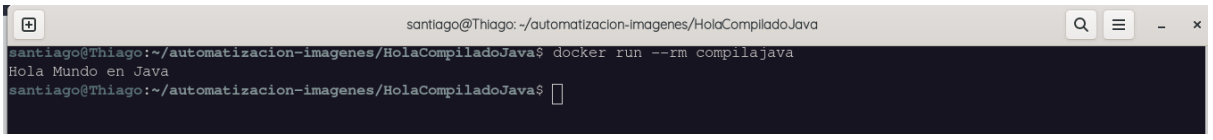
# Hola Java Compilado

Creación de archivo Dockerfile con App.java


Creación de la imágen compilajava

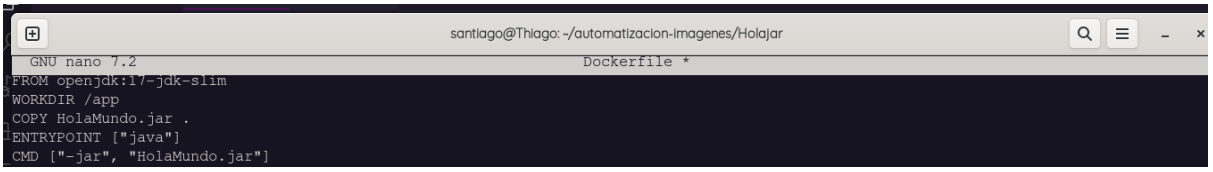



Ejecución del contenedor a partir de la imágen compilajava

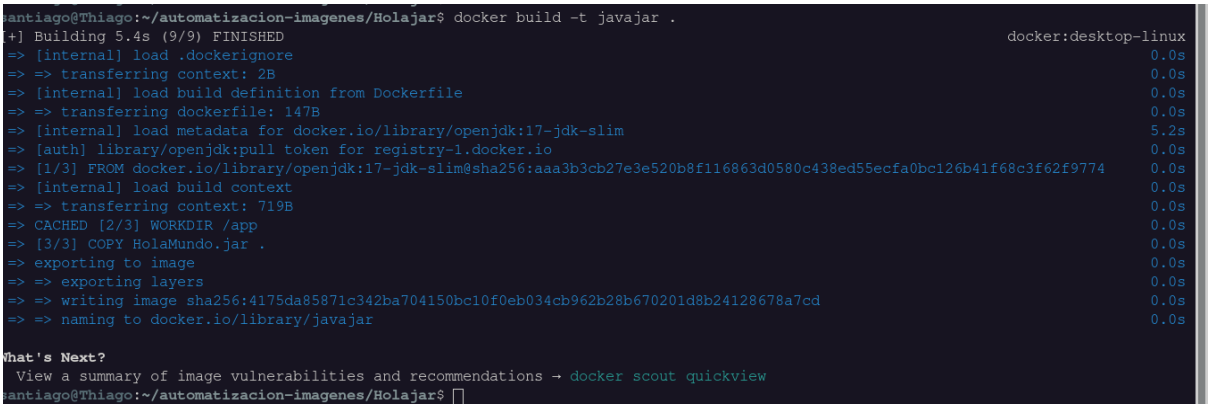


# Hola Java en archivo .JAR

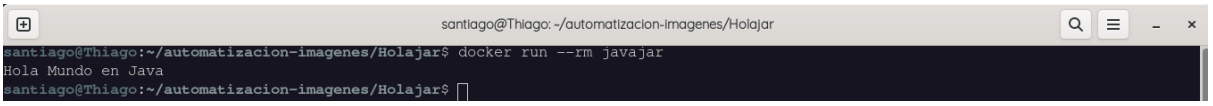
Archivo Dockerfile para la ejecución de el archivo HolaMundo.jar




Creación de la imágen javajar a partir del archivo Dockerfile

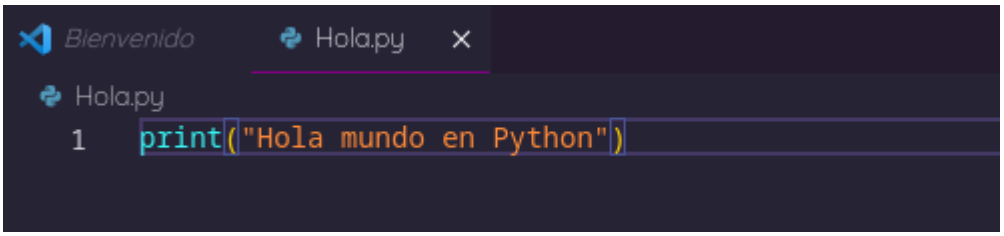



Ejecución del contenedor a partir de la imágen creada en el punto anterior



# Hola Mundo en Python

Aplicación en python que muestra un Hola Mundo




Archivo Dockerfile que instala python3 y ejecuta el programa Hola.py

```
FROM debian
RUN apt-get update && apt-get install -y python3
WORKDIR /app
COPY Hola.py .
ENTRYPOINT ["python3"]
CMD ["Hola.py"]
```

Creación de la imágen holapython a partir del Dockerfile

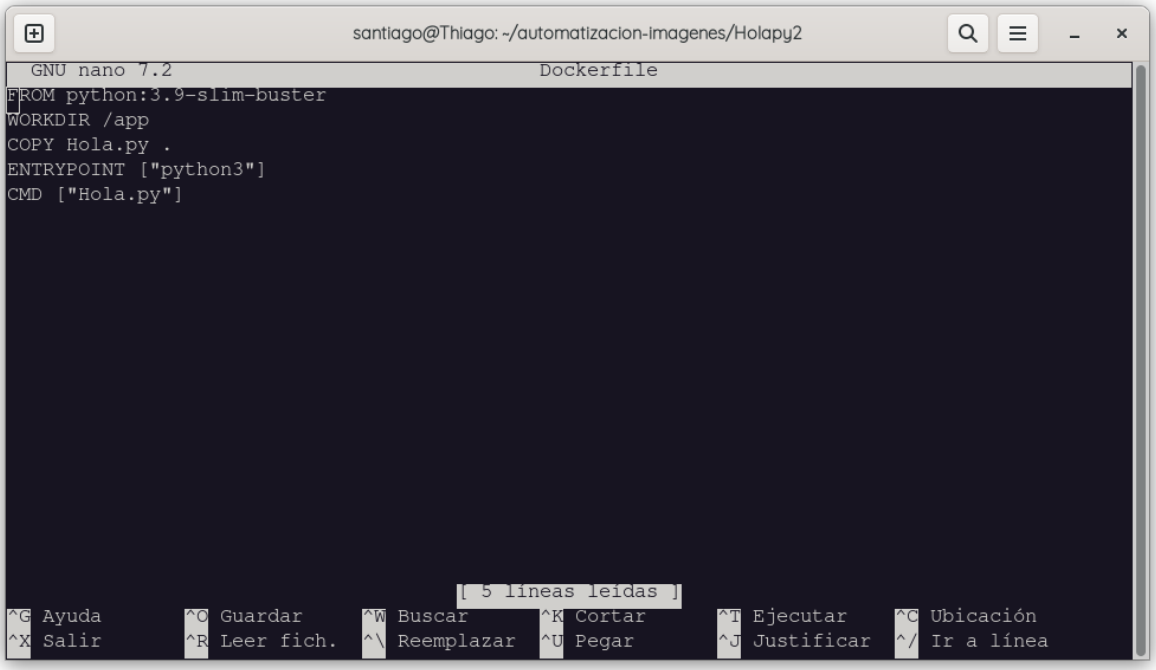
```
santiago@Thiago:~/automatizacion-imagenes/Holapython$ docker build -t holapython .
[+] Building 34.0s (10/10) FINISHED                                                                    docker:desktop-linux
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 167B                                                                                0.0s
 => [internal] load metadata for docker.io/library/debian:latest                                                    5.7s
 => [auth] library/debian:pull token for registry-1.docker.io                                                       0.0s
 => [1/4] FROM docker.io/library/debian@sha256:b4042f895d5d1f8df415caebe7c416f9dbcf0dc8867abb225955006de50b21f3     10.0s
 => => resolve docker.io/library/debian@sha256:b4042f895d5d1f8df415caebe7c416f9dbcf0dc8867abb225955006de50b21f3      0.0s
 => => sha256:df89715852d4e9045673b3d22dc80b7846d563e18af1b14648d6bfe06839116f 1.46kB / 1.46kB                      0.0s
 => => sha256:012c0b3e998c1a0c0bedcf712eaaafb188580529dd026a04aa1ce13fdb39e42b 49.56MB / 49.56MB                    7.7s
 => => sha256:b4042f895d5d1f8df415caebe7c416f9dbcf0dc8867abb225955006de50b21f3 1.85kB / 1.85kB                      0.0s
 => => sha256:88b0908ef4de0f7015fd61b7fcbfa407854349af42d1e2081595768d575995c1 529B / 529B                          0.0s
 => => extracting sha256:012c0b3e998c1a0c0bedcf712eaaafb188580529dd026a04aa1ce13fdb39e42b                           1.6s
 => [internal] load build context                                                                                   0.0s
 => => transferring context: 63B                                                                                    0.0s
 => [2/4] RUN apt-get update && apt-get install -y python3                                                         17.7s
 => [3/4] WORKDIR /app                                                                                              0.0s
 => [4/4] COPY Hola.py .                                                                                            0.0s
 => exporting to image                                                                                              0.4s
 => => exporting layers                                                                                             0.4s
 => => writing image sha256:d0a6ac8afbe90266502af7257d1f501069c3f4881a7956313e676793d41025e2                        0.0s
 => => naming to docker.io/library/holapython                                                                       0.0s

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

Creacion y ejecucion del contenedor haciendo uso de la imágen anterior

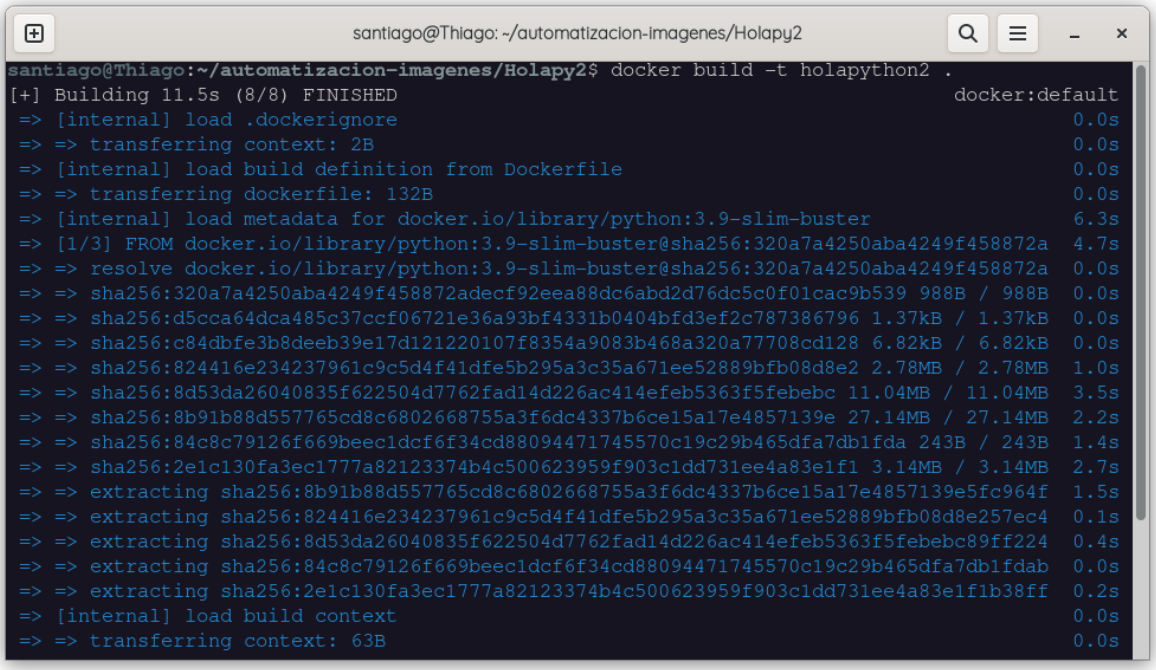
```
santiago@Thiago:~/automatizacion-imagenes/Holapython$ docker run --rm holapython
Hola mundo en Python
santiago@Thiago:~/automatizacion-imagenes/Holapython$
```

## Hola Mundo en Python 2

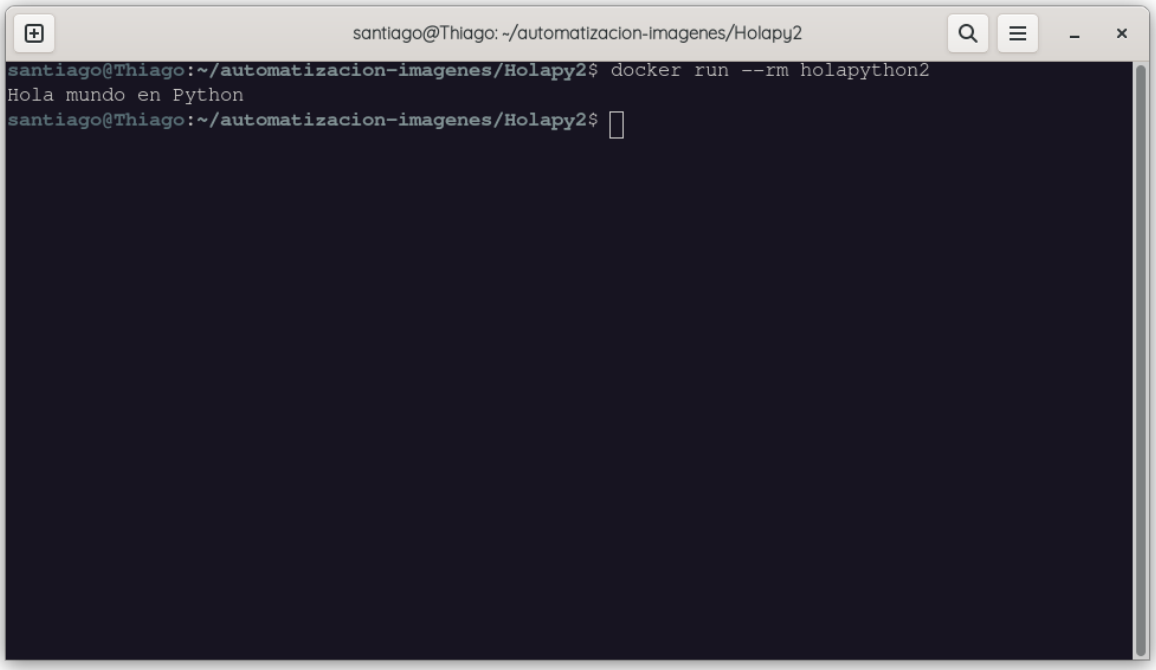
Archivo Dockerfile haciendo uso de la imagen de python 3.9 que se encuentra en Docker Hub




Creación de la imágen holapython2 a partir del Dockerfile anterior

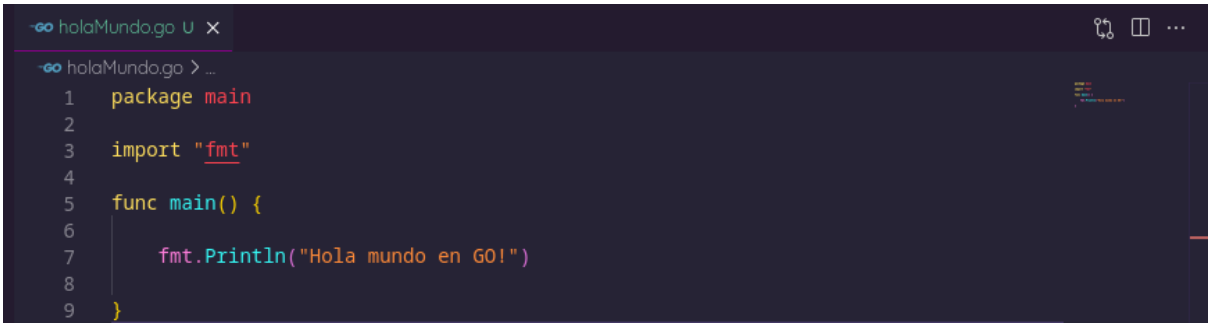


Creación y ejecución del contenedor a partir de la imágen holapython2

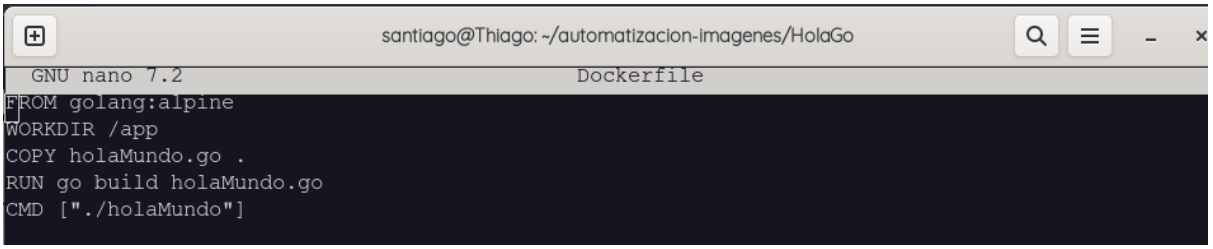


# Hola Mundo En GO

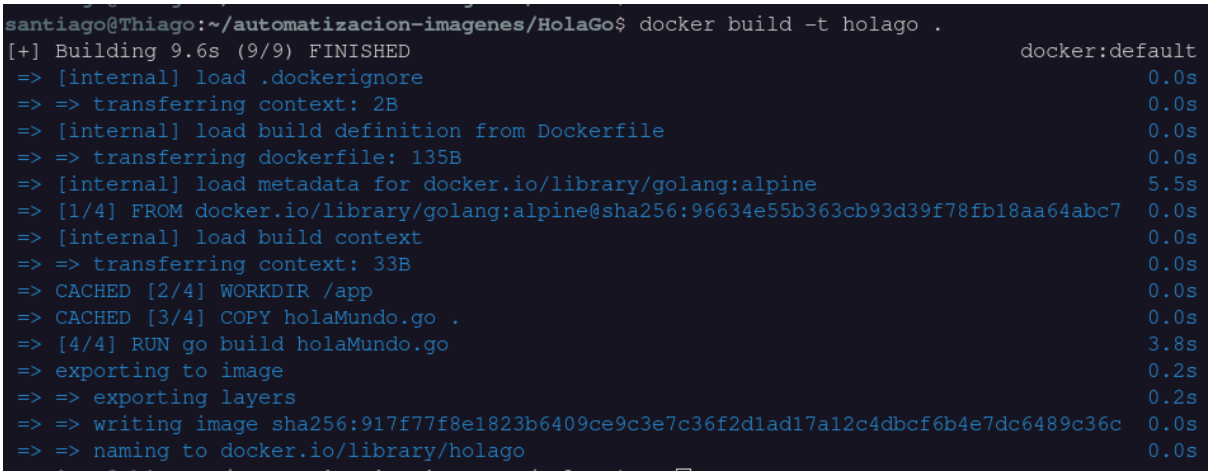
Aplicación que muestra un Hola mundo en Go




Archivo Dockerfile que permite la compilación del código anterior haciendo uso de la imágen de alpine de Go




Construcción de la imágen holago



Creación y ejecución del contenedor que permite visualizar la salida de la aplicación holaMundo.go