

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

```
santiago@Thiago: ~/automatizacion-imagenes/HolaBash
GNU nano 7.2 Dockerfile
FROM alpine
ENTRYPOINT ["echo"]
CMD ["Hola Mundo en Bash"]
```

Construcción de la imagen hola

```
santiago@Thiago: ~/automatizacion-imagenes/HolaBash
santiago@Thiago:~/automatizacion-imagenes/HolaBash$ docker build -t hola .
[+] Building 16.2s (6/6) FINISHED docker:desktop-linux
=> [internal] load build definition from Dockerfile 0.1s
=> => transferring dockerfile: 97B 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/alpine:latest 14.9s
=> [auth] library/alpine:pull token for registry-1.docker.io 0.0s
=> [1/1] FROM docker.io/library/alpine@sha256:7144f7bab3d4c2648d7e59409f15ec52a18006a128c733fcff20 1.1s
=> => resolve docker.io/library/alpine@sha256:7144f7bab3d4c2648d7e59409f15ec52a18006a128c733fcff20 0.0s
=> => sha256:7e01a0d0a1dcd9e539f8e9bbd80106d59efbdf97293b3d38f5d7a34501526cdb 1.47kB / 1.47kB 0.0s
=> => sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de 3.40MB / 3.40MB 0.9s
=> => sha256:7144f7bab3d4c2648d7e59409f15ec52a18006a128c733fcff20d3a4a54ba44a 1.64kB / 1.64kB 0.0s
=> => sha256:c5c5fda71656f28e49ac9c5416b3643eaa6a108a8093151d6d1afc9463be8e33 528B / 528B 0.0s
=> => extracting sha256:7264a8db6415046d36d16ba98b79778e18accee6ffa71850405994cffa9be7de 0.1s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:3af0408d9cc4762bb42cebc7c10c73227450da5531b87325980faaf81cfeb2ca9 0.0s
=> => naming to docker.io/library/hola 0.0s
```

Creación del contenedor

```
santiago@Thiago: ~/automatizacion-imagenes/HolaBash
santiago@Thiago:~/automatizacion-imagenes/HolaBash$ docker run --rm hola
Hola Mundo en Bash
santiago@Thiago:~/automatizacion-imagenes/HolaBash$
```

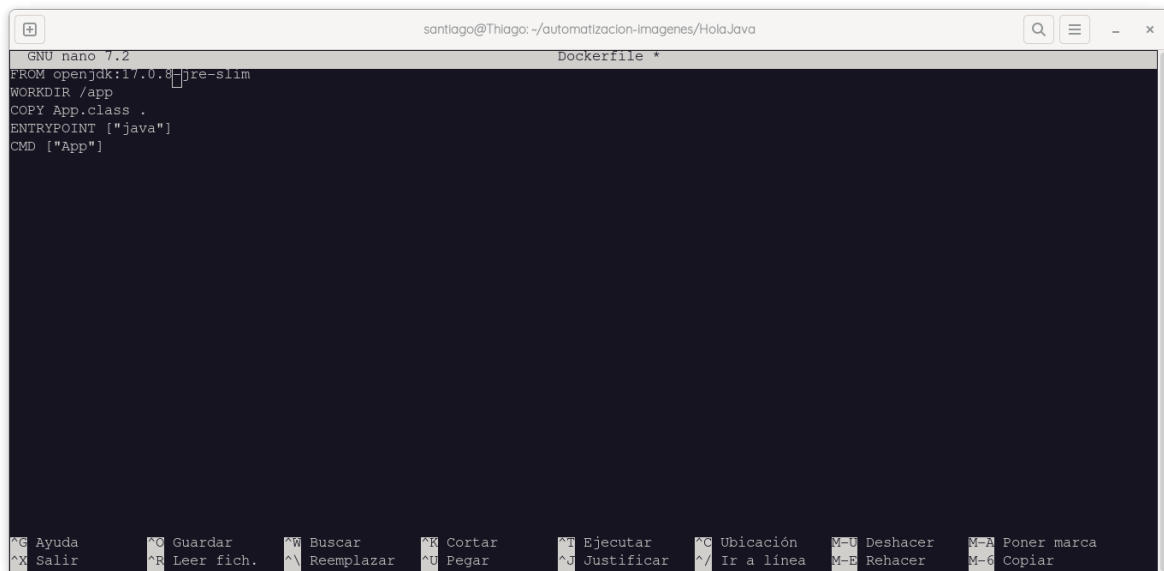
Hola Mundo en Java

Aplicación JAVA que muestra un hola mundo



```
src > J Appjava > ...
1  public class App {
2      public static void main(String[] args) throws Exception {
3          System.out.println("Hola Mundo en Java");
4      }
5  }
6  
```

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



```
santiago@Thiago: ~/automatizacion-imagenes/Hola.Java
GNU nano 7.2 Dockerfile *
FROM openjdk:17.0.8-jre-slim
WORKDIR /app
COPY App.class .
ENTRYPOINT ["java"]
CMD ["App"]

^C Ayuda      ^C Guardar   ^X Buscar    ^X Cortar    ^C Ejecutar   ^C Ubicación ^X-U Deshacer  ^X-A Poner marca
^X Salir     ^R Leer fich.^_ Reemplazar ^U Pegar      ^J Justificar ^_ Ir a línea  ^-E Rehacer    ^-C Copiar
```

Construcción de la imagen holajava

```
santiago@Thiago: ~/automatizacion-imagenes/HolaJava
App.class Dockerfile
santiago@Thiago:~/automatizacion-imagenes/HolaJava$ nano Dockerfile
santiago@Thiago:~/automatizacion-imagenes/HolaJava$ docker build -t holajava .
[+] Building 21.7s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                  0.0s
=> [internal] load build definition from Dockerfile             0.0s
=> => transferring dockerfile: 130B                              0.0s
=> [internal] load metadata for docker.io/library/openjdk:11.0.11-jre-slim 6.4s
=> [auth] library/openjdk:pull token for registry-1.docker.io  0.0s
=> [1/3] FROM docker.io/library/openjdk:11.0.11-jre-slim@sha256:24de726604f496a8d34cc960f39c3f3d8 14.6s
=> => resolve docker.io/library/openjdk:11.0.11-jre-slim@sha256:24de726604f496a8d34cc960f39c3f3d82 0.0s
=> => sha256:f1d5c8a9bc515d5cde9cf8613d73f3eae26e1ceb5c916bb3e3139e158a33084 7.57kB / 7.57kB 0.0s
=> => sha256:b4d181a07f8025e00e0cb28f1cc14613da2ce26450b80c54aea537fa93cf3bda 27.15MB / 27.15MB 4.8s
=> => sha256:3ee45ae9730633057cf9bd12924f9a1bf2b590631d3d085b14c96f4466557794 3.27MB / 3.27MB 4.2s
=> => sha256:567d410fadc4d7125fabe9cbac4ac3b566d498b3d8050c0405bdd78fdb4a384d 212B / 212B 0.9s
=> => sha256:24de726604f496a8d34cc960f39c3f3d825ebba522d8be7b256f8b289a448508 549B / 549B 0.0s
=> => sha256:dfa2c54ed832658aa90da547bb1b8bd41e5d3fffc0c89e6540dea586ccefdef 1.16kB / 1.16kB 0.0s
=> => sha256:891cdfaa81b9c4fd2d95f3539c4037c2ab928b2c0b90a00b2da92d26d0d4ea42 47.08MB / 47.08MB 13.7s
=> => extracting sha256:b4d181a07f8025e00e0cb28f1cc14613da2ce26450b80c54aea537fa93cf3bda 1.1s
=> => extracting sha256:3ee45ae9730633057cf9bd12924f9a1bf2b590631d3d085b14c96f4466557794 0.1s
=> => extracting sha256:567d410fadc4d7125fabe9cbac4ac3b566d498b3d8050c0405bdd78fdb4a384d 0.0s
=> => extracting sha256:891cdfaa81b9c4fd2d95f3539c4037c2ab928b2c0b90a00b2da92d26d0d4ea42 0.7s
=> [internal] load build context                                0.0s
=> => transferring context: 605B                                  0.0s
=> [2/3] WORKDIR /app                                          0.6s
=> [3/3] COPY App.class .                                      0.0s
=> exporting to image                                          0.1s
=> => exporting layers                                           0.0s
```

Creación del contenedor a partir de la imagen anterior

```
santiago@Thiago: ~/automatizacion-imagenes/HolaJava
santiago@Thiago:~/automatizacion-imagenes/HolaJava$ docker run --rm holajava
Hola Mundo en Java
santiago@Thiago:~/automatizacion-imagenes/HolaJava$
```

Hola Java Compilado

Creación de archivo Dockerfile con App.java

```
santiago@Thiago: ~/automatizacion-imagenes/HolaCompiladoJava
GNU nano 7.2 Dockerfile *
FROM openjdk:17-jdk-slim
WORKDIR /app
COPY App.java .
RUN javac App.java
ENTRYPOINT ["java"]
CMD ["App"]
```

Creación de la imagen compilajava

```
santiago@Thiago:~/automatizacion-imagenes/HolaCompiladoJava$ docker build -t compilajava .
[+] Building 3.1s (10/10) FINISHED
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 143B 0.0s
=> [internal] load metadata for docker.io/library/openjdk:17-jdk-slim 2.0s
=> [auth] library/openjdk:pull token for registry-1.docker.io 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 176B 0.0s
=> [1/4] FROM docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774 0.0s
=> CACHED [2/4] WORKDIR /app 0.0s
=> [3/4] COPY App.java . 0.0s
=> [4/4] RUN javac App.java 1.0s
=> exporting to image 0.0s
=> => exporting layers 0.0s
=> => writing image sha256:aa76b3173629e36fda376797a172616ecf17747c2876df827772b4e6756184d9 0.0s
=> => naming to docker.io/library/compilajava 0.0s

What's Next?
View a summary of image vulnerabilities and recommendations -> docker scout quickview
santiago@Thiago:~/automatizacion-imagenes/HolaCompiladoJava$
```

Ejecución del contenedor a partir de la imagen compilajava

```
santiago@Thiago:~/automatizacion-imagenes/HolaCompiladoJava$ docker run --rm compilajava
Hola Mundo en Java
santiago@Thiago:~/automatizacion-imagenes/HolaCompiladoJava$
```

Hola Java en archivo .JAR

Archivo Dockerfile para la ejecución de el archivo HolaMundo.jar

```
santiago@Thiago: ~/automatizacion-imagenes/Holajar
GNU nano 7.2 Dockerfile *
FROM openjdk:17-jdk-slim
WORKDIR /app
COPY HolaMundo.jar .
ENTRYPOINT ["java"]
CMD ["-jar", "HolaMundo.jar"]
```

Creación de la imagen javajar a partir del archivo Dockerfile

```
santiago@Thiago:~/automatizacion-imagenes/Holajar$ docker build -t javajar .
[+] Building 5.4s (9/9) FINISHED                                docker:desktop-linux
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                    0.0s
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 147B                               0.0s
=> [internal] load metadata for docker.io/library/openjdk:17-jdk-slim 5.2s
=> [auth] library/openjdk:pull token for registry-1.docker.io   0.0s
=> [1/3] FROM docker.io/library/openjdk:17-jdk-slim@sha256:aaa3b3cb27e3e520b8f116863d0580c438ed55ecfa0bc126b41f68c3f62f9774 0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 719B                                    0.0s
=> CACHED [2/3] WORKDIR /app                                     0.0s
=> [3/3] COPY HolaMundo.jar .                                    0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:4175da85871c342ba704150bc10f0eb034cb962b28b670201d8b24128678a7cd 0.0s
=> => naming to docker.io/library/javajar                       0.0s

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
santiago@Thiago:~/automatizacion-imagenes/Holajar$
```

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

```
santiago@Thiago: ~/automatizacion-imagenes/Holajar
santiago@Thiago:~/automatizacion-imagenes/Holajar$ docker run --rm javajar
Hola Mundo en Java
santiago@Thiago:~/automatizacion-imagenes/Holajar$
```

Hola Mundo en Python

Aplicación en python que muestra un Hola Mundo

```
Bienvenido Hola.py x
Hola.py
1 print("Hola mundo en Python")
```

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 imagen 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:b4042f895d5d1f8df415cae7c416f9dbcf0dc8867abb225955006de50b21f3 10.0s
=> => resolve docker.io/library/debian@sha256:b4042f895d5d1f8df415cae7c416f9dbcf0dc8867abb225955006de50b21f3 0.0s
=> => sha256:df89715852d4e9045673b3d22dc80b7846d563e18af1b14648d6bfe06839116f 1.46kB / 1.46kB 0.0s
=> => sha256:012c0b3e998c1a0c0bedcf712eaaafb188580529dd026a04aalce13fdb39e42b 49.56MB / 49.56MB 7.7s
=> => sha256:b4042f895d5d1f8df415cae7c416f9dbcf0dc8867abb225955006de50b21f3 1.85kB / 1.85kB 0.0s
=> => sha256:88b0908ef4de0f7015fd61b7fcbfa407854349af42d1e2081595768d575995c1 529B / 529B 0.0s
=> => extracting sha256:012c0b3e998c1a0c0bedcf712eaaafb188580529dd026a04aalce13fdb39e42b 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

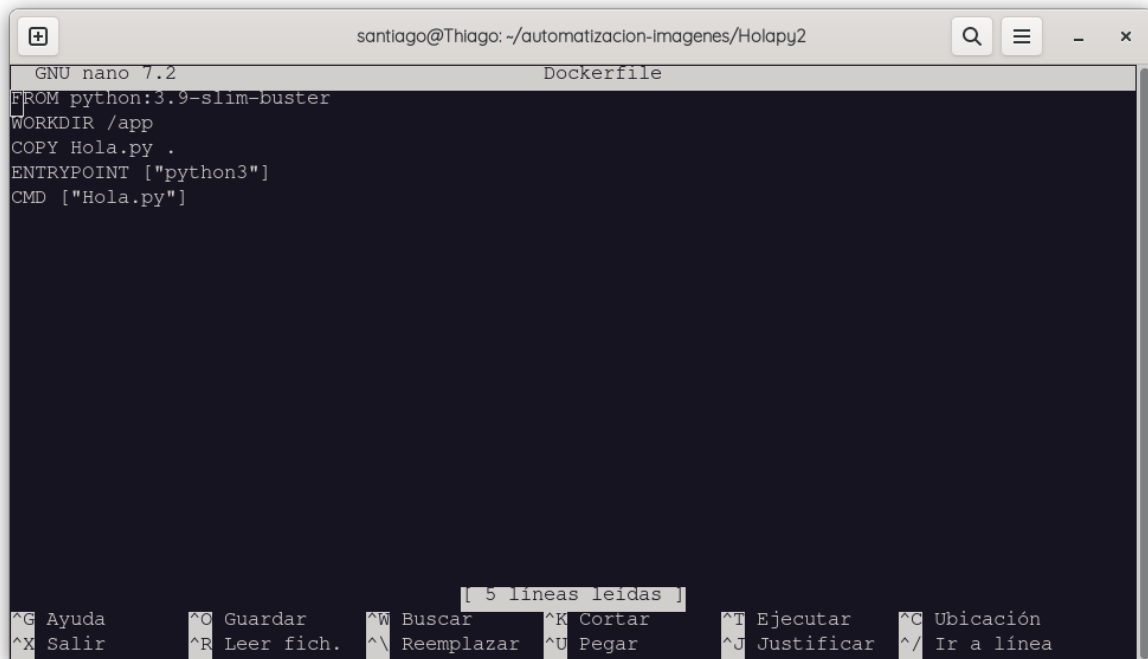
What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview
```

Creacion y ejecucion del contenedor haciendo uso de la imagen anterior

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


Hola Mundo en Python 2

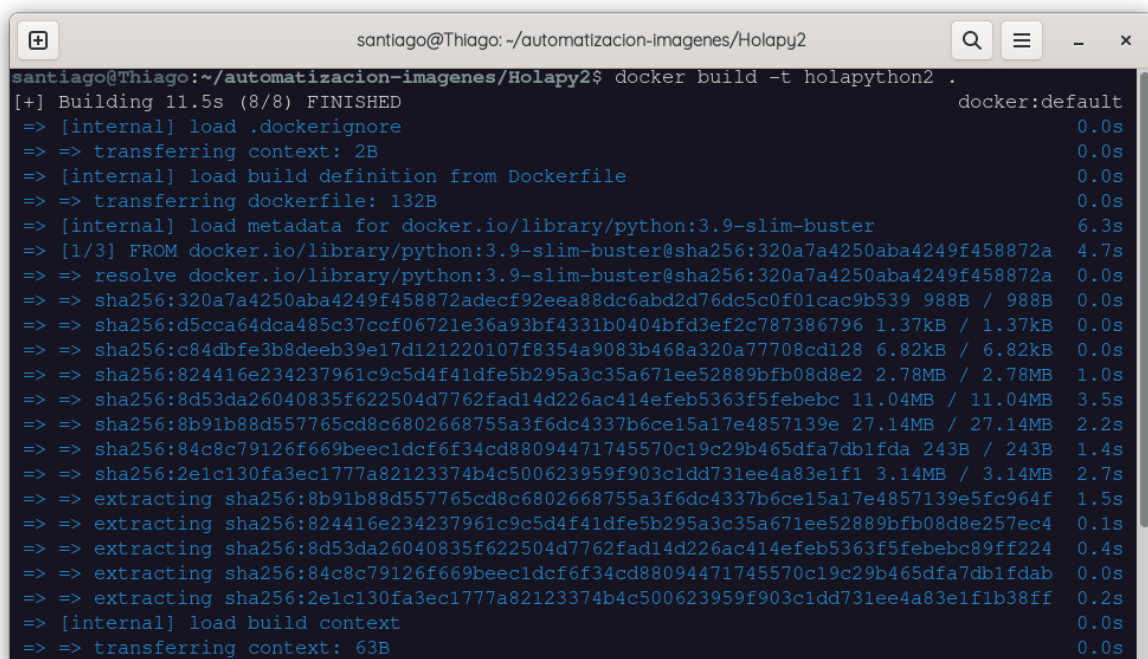
Archivo Dockerfile haciendo uso de la imagen de python 3.9 que se encuentra en Docker Hub



```
santiago@Thiago: ~/automatizacion-imagenes/Holapy2
GNU nano 7.2 Dockerfile
FROM python:3.9-slim-buster
WORKDIR /app
COPY Hola.py .
ENTRYPOINT ["python3"]
CMD ["Hola.py"]

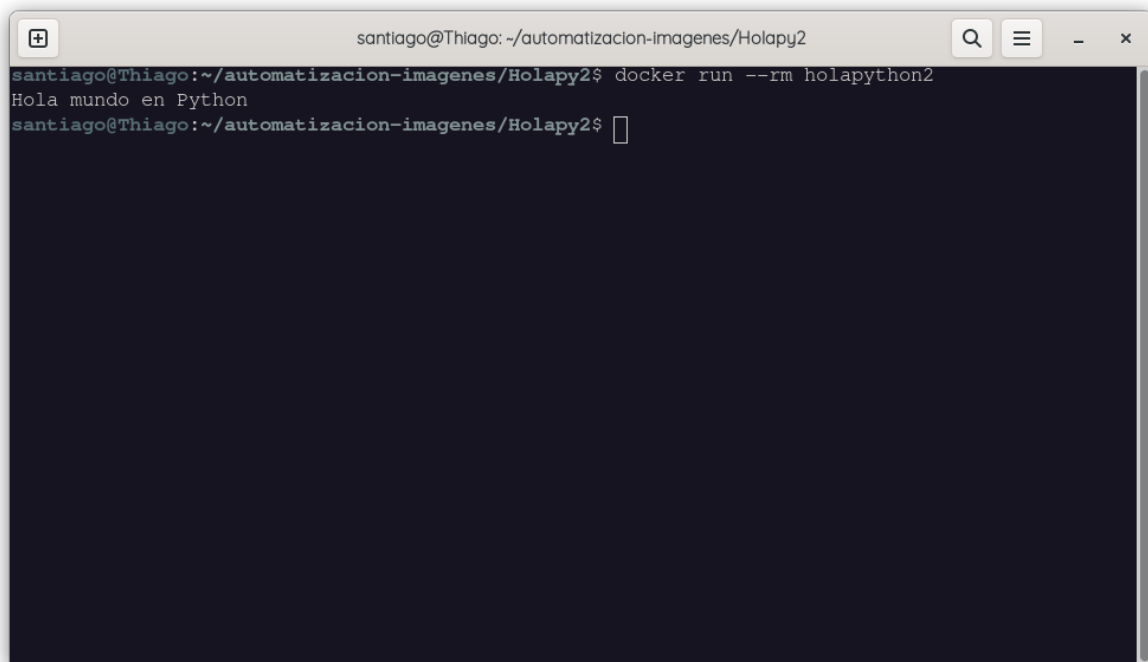
[ 5 líneas leídas ]
^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar ^/ Ir a línea
```

Creación de la imagen holapython2 a partir del Dockerfile anterior



```
santiago@Thiago: ~/automatizacion-imagenes/Holapy2$ docker build -t holapython2 .
[+] Building 11.5s (8/8) FINISHED
=> [internal] load .dockerignore                                docker:default 0.0s
=> => transferring context: 2B                                   0.0s
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 132B                               0.0s
=> [internal] load metadata for docker.io/library/python:3.9-slim-buster 6.3s
=> [1/3] FROM docker.io/library/python:3.9-slim-buster@sha256:320a7a4250aba4249f458872a 4.7s
=> => resolve docker.io/library/python:3.9-slim-buster@sha256:320a7a4250aba4249f458872a 0.0s
=> => sha256:320a7a4250aba4249f458872adecf92eea88dc6abd2d76dc5c0f01cac9b539 988B / 988B 0.0s
=> => sha256:d5cca64dca485c37ccf06721e36a93bf4331b0404bfd3ef2c787386796 1.37kB / 1.37kB 0.0s
=> => sha256:c84dbfe3b8deeb39e17d121220107f8354a9083b468a320a77708cd128 6.82kB / 6.82kB 0.0s
=> => sha256:824416e234237961c9c5d4f41dfe5b295a3c35a671ee52889bfb08d8e2 2.78MB / 2.78MB 1.0s
=> => sha256:8d53da26040835f622504d7762fad14d226ac414efeb5363f5febebc 11.04MB / 11.04MB 3.5s
=> => sha256:8b91b88d557765cd8c6802668755a3f6dc4337b6ce15a17e4857139e 27.14MB / 27.14MB 2.2s
=> => sha256:84c8c79126f669beec1dcf6f34cd88094471745570c19c29b465dfa7db1fda 243B / 243B 1.4s
=> => sha256:2e1c130fa3ec1777a82123374b4c500623959f903c1dd731ee4a83e1f1 3.14MB / 3.14MB 2.7s
=> => extracting sha256:8b91b88d557765cd8c6802668755a3f6dc4337b6ce15a17e4857139e5fc964f 1.5s
=> => extracting sha256:824416e234237961c9c5d4f41dfe5b295a3c35a671ee52889bfb08d8e257ec4 0.1s
=> => extracting sha256:8d53da26040835f622504d7762fad14d226ac414efeb5363f5febebc89ff224 0.4s
=> => extracting sha256:84c8c79126f669beec1dcf6f34cd88094471745570c19c29b465dfa7db1fdab 0.0s
=> => extracting sha256:2e1c130fa3ec1777a82123374b4c500623959f903c1dd731ee4a83e1f1b38ff 0.2s
=> [internal] load build context                                0.0s
=> => transferring context: 63B                                   0.0s
```

Creación y ejecución del contenedor a partir de la imagen holapython2



A terminal window titled 'santiago@Thiago: ~/automatizacion-imagenes/Holapy2'. The window shows the command 'docker run --rm holapython2' being executed. The output of the command is 'Hola mundo en Python'. The prompt 'santiago@Thiago:~/automatizacion-imagenes/Holapy2\$' is visible at the bottom of the terminal.

```
santiago@Thiago:~/automatizacion-imagenes/Holapy2$ docker run --rm holapython2
Hola mundo en Python
santiago@Thiago:~/automatizacion-imagenes/Holapy2$
```

Hola Mundo En GO

Aplicación que muestra un Hola mundo en Go

```
holaMundo.go U x
holaMundo.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6
7     fmt.Println("Hola mundo en GO!")
8
9 }
```

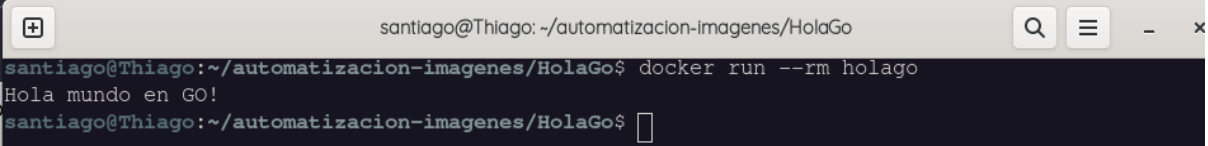
Archivo Dockerfile que permite la compilación del código anterior haciendo uso de la imagen de alpine de Go

```
santiago@Thiago: ~/automatizacion-imagenes/HolaGo
GNU nano 7.2 Dockerfile
FROM golang:alpine
WORKDIR /app
COPY holaMundo.go .
RUN go build holaMundo.go
CMD [ "./holaMundo" ]
```

Construcción de la imagen holago

```
santiago@Thiago:~/automatizacion-imagenes/HolaGo$ docker build -t holago .
[+] Building 9.6s (9/9) FINISHED docker:default
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 135B 0.0s
=> [internal] load metadata for docker.io/library/golang:alpine 5.5s
=> [1/4] FROM docker.io/library/golang:alpine@sha256:96634e55b363cb93d39f78fb18aa64abc7 0.0s
=> [internal] load build context 0.0s
=> => transferring context: 33B 0.0s
=> CACHED [2/4] WORKDIR /app 0.0s
=> CACHED [3/4] COPY holaMundo.go . 0.0s
=> [4/4] RUN go build holaMundo.go 3.8s
=> exporting to image 0.2s
=> => exporting layers 0.2s
=> => writing image sha256:917f77f8e1823b6409ce9c3e7c36f2dlad17a12c4dbcf6b4e7dc6489c36c 0.0s
=> => naming to docker.io/library/holago 0.0s
```

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

A terminal window with a title bar showing 'santiago@Thiago: ~/automatizacion-imagenes/HolaGo'. The terminal contains the command 'docker run --rm holago' and its output 'Hola mundo en GO!'. The prompt is 'santiago@Thiago:~/automatizacion-imagenes/HolaGo\$' followed by a cursor.

```
santiago@Thiago: ~/automatizacion-imagenes/HolaGo$ docker run --rm holago
Hola mundo en GO!
santiago@Thiago:~/automatizacion-imagenes/HolaGo$
```