

A instrução Env

```
ENV var=valor
```

```
Successfully tagged fernando:latest
faofernando@war-machine: ~/Projetos/docker/09 - Conhecendo o Dockerfile em Detalhes/A instrução ENV$ docker run -it fernando /bin/bash
root@d5fc7fd48530:/# env
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sq=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.tar.xz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lrz=01;31:*.lzex=01;31:*.xex=01;31:*.zst=01;31:*.tztst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tze=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
HOSTNAME=d5fc7fd48530
PWD=/
HOME=/root
var1=fernando
var2=luiz
TERM=xterm
SHELL=/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
=/usr/bin/env
```

Basicamente, a instrução env permite definir a variável de ambiente. No exemplo que foi ensinado na aula, definimos a instrução com dois valores, a partir disso é criado uma imagem com o comando

```
faofernando@war-machine:~/Projetos/docker/09 - Conhecendo o Dockerfile em Detalhes/A instrução ENV$ docker build -t="fernando" .
Sending build context to Docker daemon 2.048kB
Step 1/3 : FROM ubuntu:18.04
--> 8d5df41c547b
Step 2/3 : RUN apt-get update && apt-get install -y iputils-ping apache2
--> Running in c7447543bde5
Get:1 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2938 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [136 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [1141 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [29.9 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [2311 kB]
Get:13 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [22.8 kB]
Successfully tagged fernando:latest
faofernando@war-machine:~/Projetos/docker/09 - Conhecendo o Dockerfile em Detalhes/A instrução ENV$ docker run -it fernando /bin/bash
root@d5fc7fd48530:/# env
```

Depois que a imagem é criada, é necessário utilizar o comando docker run p/ executar em um container.

```
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
GNU nano 6.2
#Working with instructions
FROM ubuntu:18.04
RUN apt-get update && apt-get install -y iputils-ping apache2
ENV var1=ping var2=8.8.8.8
CMD $var1 $var2
#ENTRYPOINT ["apache2ctl"]
```

```
faofernando@war-machine: ~/Projetos/docker/09 - Conhecendo o Dockerfile em Detalhes/A instrução ENV$ nano Dockerfile
faofernando@war-machine:~/Projetos/docker/09 - Conhecendo o Dockerfile em Detalhes/A instrução ENV$ docker build -t="pinger" .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM ubuntu:18.04
----> 8d5df41c547b
Step 2/4 : RUN apt-get update && apt-get install -y iputils-ping apache2
----> Using cache
----> 0dec97c379e
Step 3/4 : ENV var1=ping var2=8.8.8.8
----> Running in 99bd2bef6109
Removing intermediate container 99bd2bef6109
----> abeb3d3d42dc
Step 4/4 : CMD $var1 $var2
----> Running in 30ba703a793b
Removing intermediate container 30ba703a793b
----> 7b3414e8ba2d
Successfully built 7b3414e8ba2d
Successfully tagged pinger:latest
faofernando@war-machine:~/Projetos/docker/09 - Conhecendo o Dockerfile em Detalhes/A instrução ENV$
```

```
faofernando@war-machine:~/Projetos/docker/09 - Conhecendo o Dockerfile em Detalhes/A instrução ENV$ docker logs bd2622e3e066
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=11.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=36.3 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=10.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=49.7 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=7.54 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=115 time=6.47 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=115 time=9.65 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=115 time=10.4 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=115 time=12.9 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=115 time=20.7 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=115 time=9.97 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=115 time=11.4 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=115 time=79.8 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=115 time=7.48 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=115 time=12.4 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=115 time=13.2 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=115 time=11.3 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=115 time=12.7 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=115 time=17.7 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=115 time=7.48 ms
64 bytes from 8.8.8.8: icmp_seq=21 ttl=115 time=11.0 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=115 time=7.81 ms
64 bytes from 8.8.8.8: icmp_seq=23 ttl=115 time=7.52 ms
64 bytes from 8.8.8.8: icmp_seq=24 ttl=115 time=29.9 ms
64 bytes from 8.8.8.8: icmp_seq=25 ttl=115 time=13.0 ms
64 bytes from 8.8.8.8: icmp_seq=26 ttl=115 time=8.29 ms
64 bytes from 8.8.8.8: icmp_seq=27 ttl=115 time=12.6 ms
64 bytes from 8.8.8.8: icmp_seq=28 ttl=115 time=10.9 ms
64 bytes from 8.8.8.8: icmp_seq=29 ttl=115 time=11.2 ms
64 bytes from 8.8.8.8: icmp_seq=30 ttl=115 time=8.21 ms
64 bytes from 8.8.8.8: icmp_seq=31 ttl=115 time=11.0 ms
64 bytes from 8.8.8.8: icmp_seq=32 ttl=115 time=8.72 ms
64 bytes from 8.8.8.8: icmp_seq=33 ttl=115 time=13.4 ms
64 bytes from 8.8.8.8: icmp_seq=34 ttl=115 time=13.1 ms
```

```
=18.1 ms
=18.1 ms
=18.1 ms
=17.1 ms
=18.1 ms
s + ' $(cat /dev/urandom | tr -dc a-z0-9 | fold -n 32 | xargs echo | sha256sum | cut -d _ -f 1)
```

docker logs -f CONTAINER_ID

Exibe o log em tempo real