

## SYS-STEM ArdLAB Setup

### 1 Configuración de la Raspberry PI

Para configurar la Raspberry PI debes seguir estos pasos:

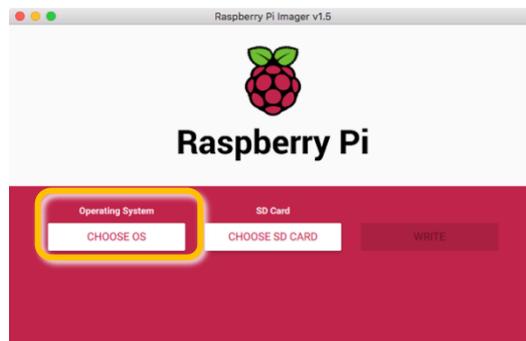
- Descarga el sistema operativo Raspberry PI OS desde <https://www.raspberrypi.com/software/>
- Graba la imagen del Raspberry PI OS en una tarjeta microSD
- Configura la conexión y los ajustes de seguridad del Raspberry PI OS

#### 1.1 Instala Raspberry PI OS

Descarga Raspberry PI Imager para el sistema operativo de tu ordenador desde el sitio web de Raspberry ([aquí](#)). Instala Raspberry PI Imager y realiza los siguientes pasos.

##### 1.1.1 Copia el Raspberry Pi OS a la tarjeta microSD

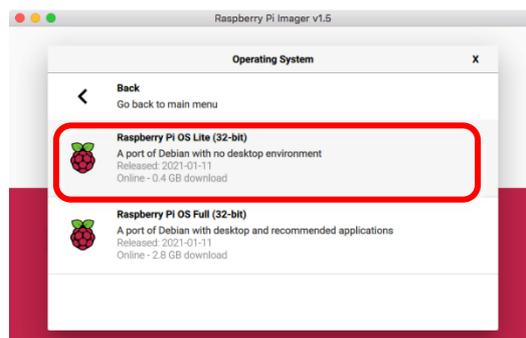
Selecciona “*CHOOSE OS*”



Selecciona “*Raspberry Pi OS (other)*”



Selecciona “*Raspberry Pi OS Lite (32-bit)*”



Selecciona “*CHOOSE SD CARD*”



Selecciona la tarjeta SD que vas a usar

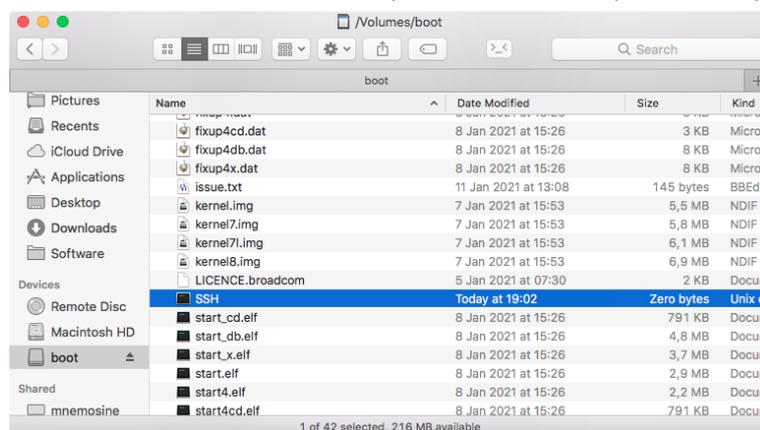


Por último, copia la imagen a la tarjeta microSD clicando “*WRITE*”



### 1.1.2 Habilita SSH antes del primer arranque

Para habilitar la conexión segura remota (SSH) a la Raspberry Pi antes del primer arranque, se debe crear un archivo vacío llamado SSH en la partición de arranque de la tarjeta microSD.



## 1.2 Primer arranque y configuración básica

Monta la tarjeta microSD en tu ordenador y crea el archivo (ten mucho cuidado en Windows, porque la mayoría de los editores siempre añaden una extensión y el comportamiento por defecto de Windows, es ocultar las extensiones conocidas)

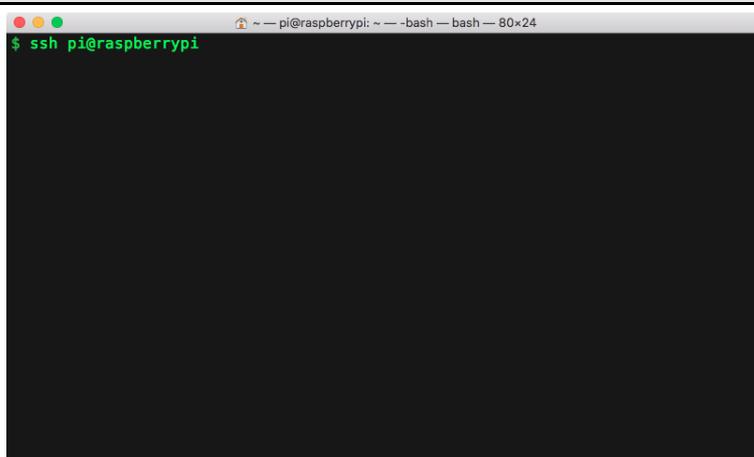
### 1.2.1 Instala la tarjeta microSD en la Raspberry Pi y haz la primera ejecución

Para evitar la necesidad de conectar un monitor y un teclado a la Raspberry Pi, le conectas un cable de red. Después de arrancar la Raspberry Pi, deberías poder conectarte usando SSH.

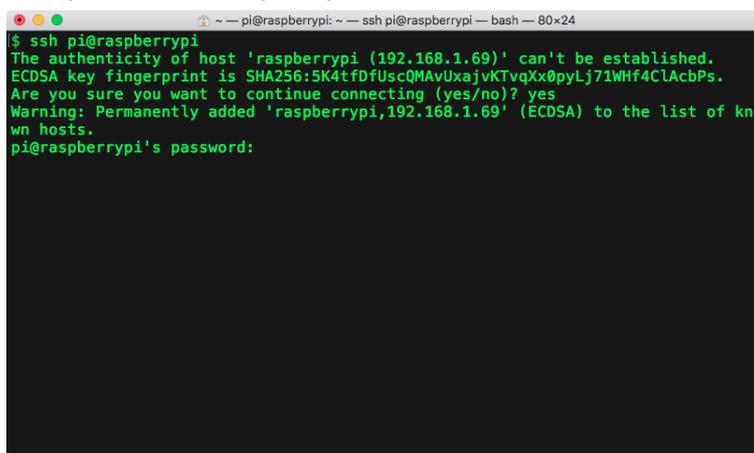
### 1.2.2 Conecta con la Raspberry Pi

- Para conectar con la Raspberry Pi:

```
ssh pi@raspberrypi
```



- Cuando te pregunte por la autenticidad del servidor, responde “yes” y entra con la contraseña por defecto “raspberrypi”.

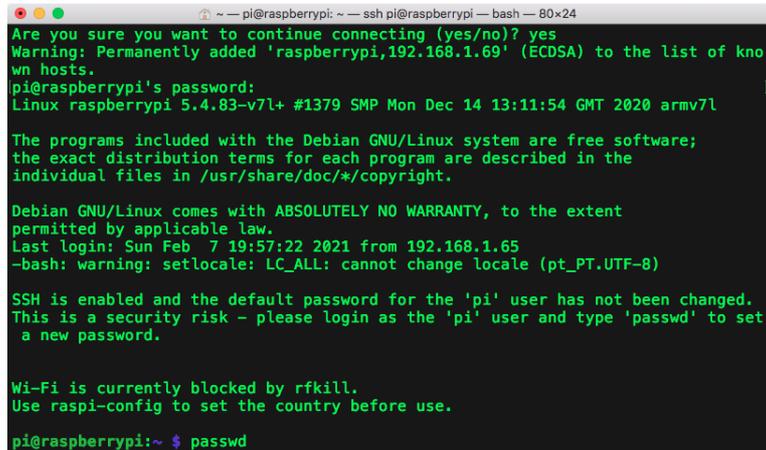


- Si no puedes conectarte, tendrás que añadir un monitor a la Raspberry PI, identificar la IP presentada en la consola y utilizarla en lugar de `raspberrypi`, por ej.:

```
ssh pi@192.168.1.69
```

- Después de iniciar la sesión, cambia la contraseña PI por una más segura. Usa el comando:

```
passwd
```



```

pi@raspberrypi: ~ -- ssh pi@raspberrypi -- bash -- 80x24
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'raspberrypi,192.168.1.69' (ECDSA) to the list of known hosts.
pi@raspberrypi's password:
Linux raspberrypi 5.4.83-v7l+ #1379 SMP Mon Dec 14 13:11:54 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Feb  7 19:57:22 2021 from 192.168.1.65
-bash: warning: setlocale: LC_ALL: cannot change locale (pt_PT.UTF-8)

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

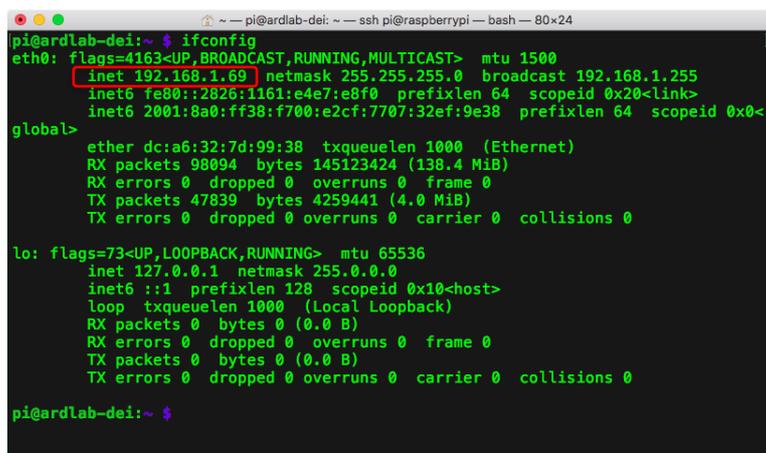
pi@raspberrypi:~ $ passwd

```

- Ejecuta el programa de información de la red:

```
ifconfig
```

Anota la dirección IP de la interfaz



```

pi@ardlab-dei:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.69 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::2826:1161:e4e7:e8f0 prefixlen 64 scopeid 0x20<link>
    inet6 2001:8a0:ff30:f700:e2cf:7707:32ef:9e38 prefixlen 64 scopeid 0x0<
global>
    ether dc:a6:32:7d:99:38 txqueuelen 1000 (Ethernet)
    RX packets 98094 bytes 145123424 (138.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 47839 bytes 4259441 (4.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@ardlab-dei:~ $

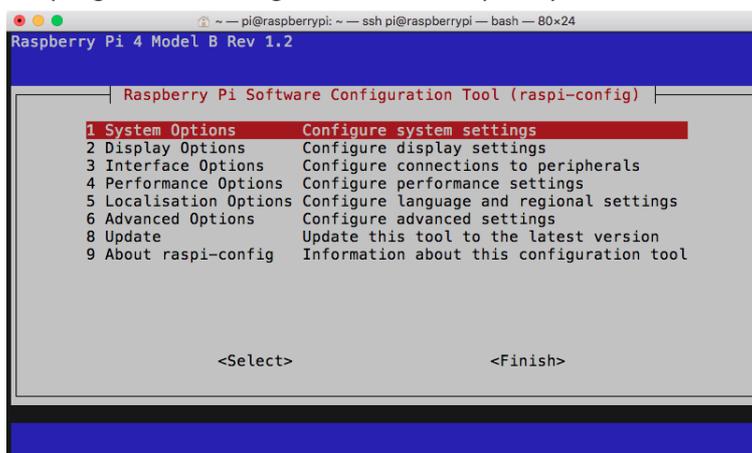
```

- A continuación, puedes establecer algunas de las configuraciones por defecto en la Raspberry Pi. Usa el comando:

```
sudo raspi-config
```

El comando `sudo` permite ejecutar `raspi-config` con permisos de root (administrador).

- Se abre el programa de configuración de la raspberry



- Configura las siguientes opciones:

1. Opciones del sistema (“*System Options*”)

**S4 Hostname** - Cambia el nombre del host de la raspberry, por ej.: ArdLAB-1. Si la IP de la Raspberry es pública, habrá que registrar este nombre en el servidor de DNS. Si la IP de la Raspberry es privada, debe ser estática y debe configurarse una regla de reenvío de puertos en el servidor NAT para el puerto 7575 de la Raspberry PI.

**S1 Wireless LAN, no recomendada** - Puedes configurar la conexión WiFi de la Raspberry Pi (deberías conectarte con un cable ethernet).

3. Opciones de interfaz

**P1 Camera** - Si estás utilizando la Raspberry PI como cámara de ArdLAB, debes habilitarla aquí.

5. Opciones de localización

**L2 Timezone** - Configura la zona horaria de tu servidor, por ejemplo Europe/Lisbon, Europe/Madrid, Europe/Athens, Europe/Zagreb, etc.

Después de salir de `raspi-config`, la Raspberry Pi puede pedir que se reinicie.

### 1.3 Actualiza el Raspberry Pi OS

A continuación, tienes que actualizar el sistema operativo de la Raspberry Pi utilizando los siguientes comandos en secuencia (el comando `apt-get` también se puede utilizar para actualizaciones):

```
sudo apt update
sudo apt upgrade
sudo reboot
```

Explicación de los comandos: “*update*” actualiza la información de los paquetes mediante descarga desde fuentes de internet configuradas; “*upgrade*” actualiza los paquetes actualmente instalados; “*reboot*” reinicia la Raspberry Pi

```

pi@ardlab-dei:~$ sudo apt update
Hit:1 http://archive.raspberrypi.org/debian buster InRelease
Hit:2 http://raspbian.raspberrypi.org/raspbian buster InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
28 packages can be upgraded. Run 'apt list --upgradable' to see them.
pi@ardlab-dei:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
 bluez-firmware ca-certificates device-tree-compiler file firmware-atheros
 firmware-brcm80211 firmware-libertas firmware-misc-nonfree firmware-realtek
 iproute2 libgnutls30 libldap-2.4-2 libldap-common libmagic-mgc libmagic1
 libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0
 python-rpi.gpio raspberrypi-bootloader raspberrypi-kernel
 raspberrypi-sys-mods rpi-eeeprom rpi.gpio-common sudo tzdata unzip
28 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 137 MB of archives.
After this operation, 3575 kB of additional disk space will be used.
pi@ardlab-dei:~$
  
```

Instala vim editor para tener un editor vi mejor.

```
sudo apt install vim
```

Para editar archivos en el terminal, puedes utilizar *vi* o *nano*. Si no estás familiarizado con *vi*, el editor *nano* es la mejor opción. Después de editar un archivo, pulsa *ctrl+X* para salir, y elige la respuesta adecuada para guardar las modificaciones.

```
nano file.txt
```

## 2 ArdLAB Setup

Para configurar ArdLAB debes seguir estos pasos:

- Instala Docker daemon en la Raspberry PI
- Clona el repositorio ArdLAB
- Configura e instala el o los contenedores Docker de ArdLAB
- Crea el archivo de configuración de ArdLAB
- Conecta el Arduino UNO a la Raspberry PI
- Prueba la configuración

### 2.1 Instala Docker en la Raspberry PI

**Docker** es un conjunto de productos de plataforma como servicio que utilizan la virtualización a nivel de sistema operativo para entregar el software en paquetes llamados “*containers*” (contenedores). Los contenedores están aislados unos de otros y agrupan su propio software, bibliotecas y archivos de configuración. Pueden comunicarse entre sí a través de canales bien definidos. Como todos los contenedores, comparten los servicios de un único núcleo del sistema operativo, por lo que utilizan menos recursos que las máquinas virtuales.

Para instalar dockers en la Raspberry PI hay que seguir estos pasos (basados en las instrucciones disponibles [aquí](#))

- Instala los requisitos.

```
sudo apt install apt-transport-https ca-certificates
software-properties-common -y
```

- Reinicia la Raspberry Pi.

```
sudo reboot
```

- Descarga el instalador de Docker y ejecútalo.

```
curl -fsSL get.docker.com -o get-docker.sh && sh get-docker.sh
```

- Da al usuario pi la capacidad de ejecutar Docker.

```
sudo usermod -aG docker pi
```

- Importa la clave CPG de Docker.

```
sudo curl -fsSL https://download.docker.com/linux/raspbian/gpg | sudo apt-key add -
```

- Añade el repositorio de docker a la lista de fuentes de apt.

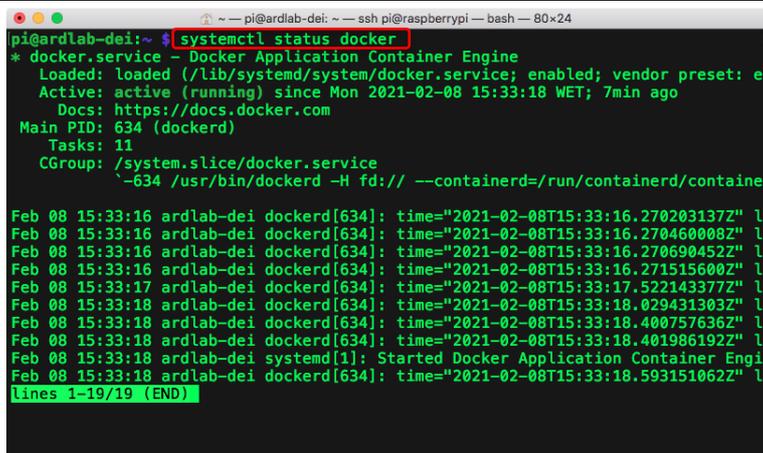
```
sudo sh -c 'echo "deb https://download.docker.com/linux/raspbian/ buster stable" >> /etc/apt/sources.list'
```

- Actualiza el Raspberry Pi OS y reinicia.

```
sudo apt update
sudo apt upgrade
sudo reboot
```

- Comprueba que el Docker daemon se está ejecutando (pulsas Q para salir del listado).

```
systemctl status docker
```



```
pi@ardlab-dei:~$ systemctl status docker
* docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Mon 2021-02-08 15:33:18 WET; 7min ago
     Docs: https://docs.docker.com
    Main PID: 634 (dockerd)
      Tasks: 11
   CGroup: /system.slice/docker.service
           └─634 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containe

Feb 08 15:33:16 ardlab-dei dockerd[634]: time="2021-02-08T15:33:16.270203137Z" l
Feb 08 15:33:16 ardlab-dei dockerd[634]: time="2021-02-08T15:33:16.27046008Z" l
Feb 08 15:33:16 ardlab-dei dockerd[634]: time="2021-02-08T15:33:16.270690452Z" l
Feb 08 15:33:16 ardlab-dei dockerd[634]: time="2021-02-08T15:33:16.271515600Z" l
Feb 08 15:33:17 ardlab-dei dockerd[634]: time="2021-02-08T15:33:17.522143377Z" l
Feb 08 15:33:18 ardlab-dei dockerd[634]: time="2021-02-08T15:33:18.029431303Z" l
Feb 08 15:33:18 ardlab-dei dockerd[634]: time="2021-02-08T15:33:18.400757636Z" l
Feb 08 15:33:18 ardlab-dei dockerd[634]: time="2021-02-08T15:33:18.401986192Z" l
Feb 08 15:33:18 ardlab-dei systemd[1]: Started Docker Application Container Engi
Feb 08 15:33:18 ardlab-dei dockerd[634]: time="2021-02-08T15:33:18.593151062Z" l
lines 1-19/19 (END)
```

Después de reiniciar, debería estar funcionando el servicio docker. Para gestionar el servicio docker puedes utilizar las siguientes opciones `systemctl`:

*start* - Inicia el servicio

*stop* - Detiene el servicio

*restart* - Reinicia el servicio, equivalente a *stop* + *start*

*reload* - Recarga los archivos de configuración del servicio

*enable* - Habilita el servicio (se ejecuta automáticamente en el arranque)

*disable* - Desactiva el servicio

- Comprueba la información del docker instalado.

```
docker info
```

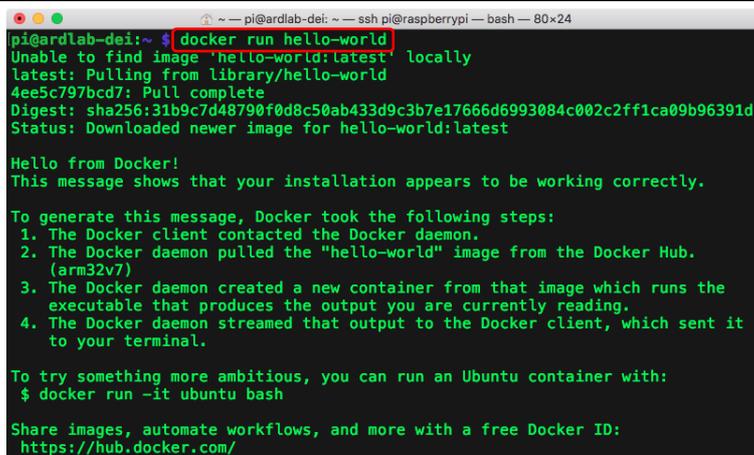
- Instala el docker compose

```
sudo apt install docker-compose
```

## 2.2 Crea y ejecuta el primer contenedor

- Para probar la instalación de Docker, crea y ejecuta un contenedor hello-world

```
docker run hello-world
```



```
pi@ardlab-dei:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
4ee5c797bcd7: Pull complete
Digest: sha256:31b9c7d48790f0d8c50ab433d9c3b7e17666d6993084c002c2ff1ca09b96391d
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (arm32v7)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/
```

Si ves el mensaje anterior, el contenedor ha sido creado y ejecutado. Para gestionar los contenedores y las imágenes, se pueden utilizar los siguiente comandos:

- Para mostrar imágenes de contenedores existentes localmente:

```
docker image ls
```

- Para mostrar la página de ayuda de Docker:

```
docker --help
```

- Para mostrar la página de ayuda del comando Docker:

```
docker image --help
```

- Eliminar la imagen local (utilizar el hash o nombre corto, utilizar `docker image ls` para mostrar los nombres o hash)

Utiliza la tecla TAB para autocompletar nombres/comandos.

- Las imágenes utilizadas en los contenedores no pueden eliminarse sin eliminar el contenedor.

```
docker image remove hello-world:latest
```

- Un comando alternativo para eliminar imágenes es:

```
docker rmi hello-world:latest
```

- Mostrar los contenedores que se están ejecutando

```
docker ps
```

- Mostrar todos los contenedores

```
docker ps -a
```

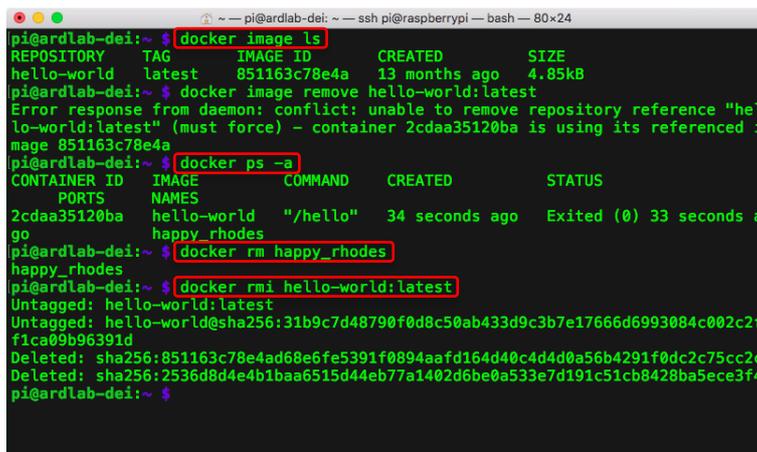
- Ahora puedes eliminar el contenedor hello-world utilizando el nombre/hash

```
docker rm happy_rhodes
```

- Comando alternativo para eliminar el contenedor utilizando la clave hash

```
docker rm 2cdaa3512ba
```

El resultado de algunos de los comandos anteriores:



```

pi@ardlab-dei:~$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
hello-world latest 851163c78e4a 13 months ago 4.85kB
pi@ardlab-dei:~$ docker image remove hello-world:latest
Error response from daemon: conflict: unable to remove repository reference "hello-world:latest" (must force) - container 2cdaa35120ba is using its referenced image 851163c78e4a
pi@ardlab-dei:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
2cdaa35120ba hello-world "/hello" 34 seconds ago Exited (0) 33 seconds ago
happy_rhodes
pi@ardlab-dei:~$ docker rm happy_rhodes
happy_rhodes
pi@ardlab-dei:~$ docker rmi hello-world:latest
Untagged: hello-world:latest
Untagged: hello-world@sha256:31b9c7d48790f0d8c50ab433d9c3b7e17666d6993084c002c2f1ca09b96391d
Deleted: sha256:851163c78e4ad68e6fe5391f0894aafd164d40c4d4d0a56b4291f0dc2c75cc2c
Deleted: sha256:2536d8d4e4b1baa6515d44eb77a1402d6be0a533e7d191c51cb8428ba5ece3f4
pi@ardlab-dei:~$

```

## 2.3 Configura el ArdLAB

- Clona el repositorio dentro del directorio principal del usuario pi, usando el siguiente comando:

```
git clone https://bitbucket.org/pssmatos/ardlab-setup.git
```

Se crea una nueva carpeta llamada `ardlab-setup` con el archivo de configuración del contenedor

### 2.3.1 Configura e instala el contenedor Docker de ArdLAB

- Cambia el directorio actual a `ardlab-setup` (puedes usar la tecla TAB para autocompletar el nombre del directorio)

```
cd ardlab-setup
```

- Muestra una lista del contenido del directorio:

```
ls -la
```

Sólo hay un archivo importante en este repositorio, el archivo de definición de docker-compose

```

/Volumes/Macintosh HD/Users/matos/Documents/cadeiras/UpSkill/Curso OutSystems/OutSystems/Reacti...
pi@ardlab-dei:~$ git clone https://pssmatos@bitbucket.org/pssmatos/ardlab-setup
.git
Cloning into 'ardlab-setup'...
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (4/4), done.
pi@ardlab-dei:~$ cd ardlab-setup/
pi@ardlab-dei:~/ardlab-setup$ ls -la
total 20
drwxr-xr-x  3 pi pi 4096 Feb 12 05:39 .
drwxr-xr-x 12 pi pi 4096 Feb 12 05:39 ..
drwxr-xr-x  8 pi pi 4096 Feb 12 05:39 .git
-rw-r--r--  1 pi pi   9 Feb 12 05:39 .gitignore
-rw-r--r--  1 pi pi 177 Feb 12 05:39 docker-compose.yml
pi@ardlab-dei:~/ardlab-setup$

```

- Inicia la configuración ejecutando el contenedor para crear el archivo de configuración con el siguiente comando:

```
docker-compose run --rm --entrypoint "python3 constellation.py" ardlab
```

El guión debería ejecutarse y presentar alguna información: la dirección IP pública, el nuevo código hash de ArdLAB, y un mensaje para registrar este hash en el sitio de la constelación (ya sea añadiendo un nuevo laboratorio o editando uno existente)

```

/Volumes/Macintosh HD/Users/matos/Documents/cadeiras/UpSkill/Curso OutSystems/OutSystems/Reacti...
pi@ardlab-dei:~/ardlab-setup$ ls -la
total 20
drwxr-xr-x  3 pi pi 4096 Feb 12 05:45 .
drwxr-xr-x 12 pi pi 4096 Feb 12 05:48 ..
drwxr-xr-x  8 pi pi 4096 Feb 12 05:39 .git
-rw-r--r--  1 pi pi   9 Feb 12 05:39 .gitignore
-rw-r--r--  1 pi pi 177 Feb 12 05:39 docker-compose.yml
pi@ardlab-dei:~/ardlab-setup$ docker-compose run --rm --entrypoint "python3 constellation.py" ardlab
Starting ArdLab Service...
After import_module
Hash ID is
Camera is OpenCV
Loading OpenCV camera module...
My public IP address is: 188.83.57.83
No hash, creating new hash 7c634113322042d9865f24675c46dd6b
You must add this hash to the constellation
pi@ardlab-dei:~/ardlab-setup$

```

- El paso anterior también creó el archivo de configuración, incluido en el directorio config que se puede presentar utilizando:

```
cat config/config.cfg
```

```

/Volumes/Macintosh HD/Users/matos/Documents/cadeiras/UpSkill/Curso OutSystems/OutSystems/Reacti...
pi@ardlab-dei:~/ardlab-setup $ ls -la
total 24
drwxr-xr-x  4 pi  pi  4096 Feb 12 05:49 .
drwxr-xr-x 12 pi  pi  4096 Feb 12 05:48 ..
drwxr-xr-x  8 pi  pi  4096 Feb 12 05:39 .git
-rw-r--r--  1 pi  pi    9 Feb 12 05:39 .gitignore
drwxr-xr-x  2 root root 4096 Feb 12 05:49 config
-rw-r--r--  1 pi  pi   177 Feb 12 05:39 docker-compose.yml
pi@ardlab-dei:~/ardlab-setup $ ls -la config
total 12
drwxr-xr-x  2 root root 4096 Feb 12 05:49 .
drwxr-xr-x  4 pi  pi  4096 Feb 12 05:49 ..
-rw-r--r--  1 root root  402 Feb 12 05:49 config.cfg
pi@ardlab-dei:~/ardlab-setup $ cat config/config.cfg
ARDLAB_HASH="7c634113322042d9865f24675c46dd6b"
CONSTELATION="https://sys-stem.dei.isep.ipp.pt:8080/api/lab/"
# Camera connected to the RaspBerry PI, options are:
# "pi" - Use pi as video provider
# "OpenCV" - Use OpenCV as video provider
# "v4l2" - Use Video 4 Linux 2 video provider
#CAMERA_TYPE="pi"
#CAMERA_TYPE="OpenCV"
#CAMERA_TYPE= ""
CAMERA_TYPE="OpenCV"
OPENCV_CAMERA_SOURCE=""
pi@ardlab-dei:~/ardlab-setup $

```

Este archivo de configuración se editará en un paso posterior, para configurar tu nombre ArdLAB, el puerto y la cámara.

### 2.3.2 Conecta el Arduino UNO a la Raspberry PI

Ahora puedes conectar el Arduino UNO a la Raspberry PI utilizando uno de los puertos USB

### 3 Configurar dDNS y obtener certificados SSL

Para acceder al ArdLAB con encriptación (HTTPS), necesitamos utilizar un certificado SSL válido que, dependiendo del laboratorio, se puede obtener utilizando dos opciones distintas.

Si el ArdLAB tiene una dirección IP pública fija y un nombre DNS válido, continúa a la sección 3.3.

En caso de que tu ArdLAB tenga una de las siguientes restricciones, continúa a la sección correspondiente:

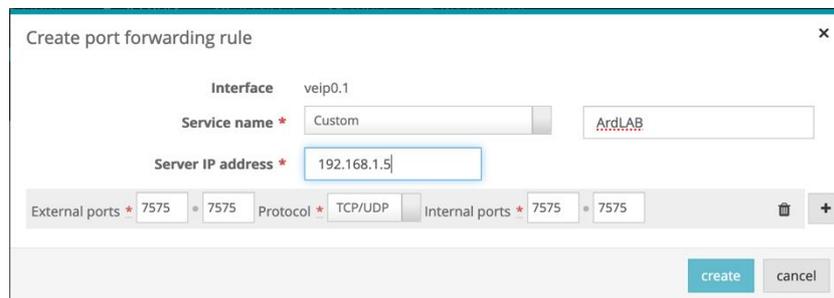
- Dirección IP fija, pero no se le puede asignar un nombre DNS (sección 3.2 y 3.3)
- Dirección IP pública dinámica (sección 3.2 y 3.3)
- Dirección IP privada (secciones 3.1, 3.2 y 3.3)

#### 3.1 Añadir regla de reenvío de puertos a tu ArdLAB

Si tu ArdLAB está detrás de un router NAT, tiene una dirección IP privada y el laboratorio no será accesible desde Internet. Para superar este problema, añade una “FORWARD PORT RULE” en el servidor NAT, para redirigir el tráfico entrante al puerto de comunicación de ArdLAB (7575).

Pide a tu administrador de red que cree dicha regla o, si tienes acceso al servidor NAT (o al router), configúrala tú mismo. El puerto por defecto de ArdLAB es el 7575 (puedes utilizar otro si el puerto 7575 no está disponible) y la dirección IP del servidor será la IP privada de ArdLAB.

La siguiente imagen muestra la configuración de una regla de este tipo en una página de configuración del router.



#### 3.2 Añadir tu Raspberry PI a un servicio DDNS

Si tu ArdLAB no tiene una IP fija con un nombre DNS válido, tendrás que crear una cuenta en un proveedor de servicios DDNS y crear un nuevo registro DNS para el ArdLAB.

En este ejemplo, utilizaremos el servicio Dynu dDNS (<https://www.dynu.com/>).

Es importante que el proveedor de servicios pueda crear registros TXT utilizando el script ACME Shell: acme.sh dnsapi (<https://github.com/acmesh-official/acme.sh/wiki/dnsapi>)

**NOTA:** Si tienes una dirección privada o una IP pública sin nombre DNS en tu ArdLAB, continúa a la siguiente sección, de lo contrario pasa a la sección 3.3.

### 3.2.1 Crear una cuenta en el servicio Dynu dDNS

- Accede al sitio <https://www.dynu.com/>:
- 1 Haz clic en el enlace “Create Account” en la parte superior derecha
- 2 Rellena tus datos
- 3 Envía la solicitud de “Create Account”.
- 4 Abre tu correo electrónico, busca el correo de verificación de Dynu y haz clic en la URL

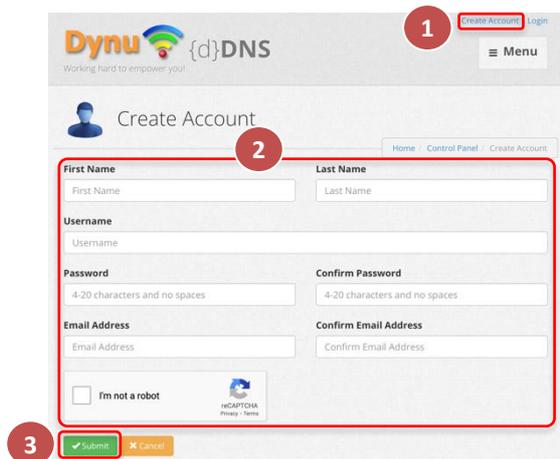


Figura 1 - Crear una cuenta en el servicio Dynu dDNS

### 3.2.2 Crear un registro DNS para tu ArdLAB

- Accede a tu cuenta en el sitio <https://www.dynu.com/> :
- 1 Accede al “Control Panel” (icono de engranaje)
- 2 En el “Control Panel”, selecciona “DDNS Services”
- 3 En la página “Dynamic DNS Service”, haz clic en “+Add”.
- 4 Rellena el nombre de tu ArdLAB (host) y elige un dominio de primer nivel (TLD).
- 5 Clic en “+Add”

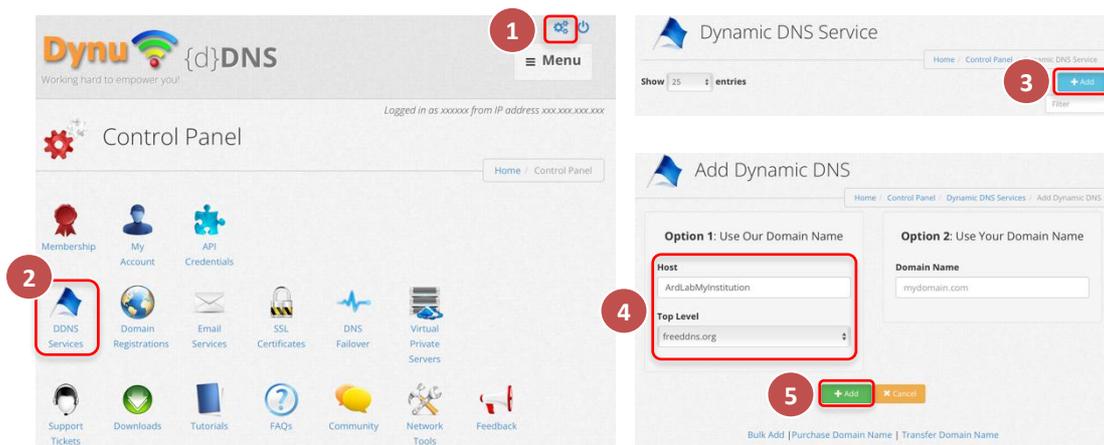


Figura 2 - Crear un registro DNS para tu ArdLAB

### 3.2.3 Gestiona tu registro ArdLAB dDNS

- Aparecerá el apartado “Manage Dynamic DNS Service”, y deberás:
  - 1 Cambiar tu dirección IPv4 por una “no válida”, por ejemplo: 1.1.1.1
  - 2 Clicka en “Save” para actualizar tu dirección IPv4
  - 3 Clicka en el botón “IP Update Password” en la parte inferior de la página
- En la página “Manage Credentials”:
  - 4 Introduce tu contraseña de Dynu
  - 5 Introduce una nueva contraseña de actualización de IP (esta contraseña se utilizará en el ArdLAB)
  - 6 Clicka en “Save” para actualizar la contraseña

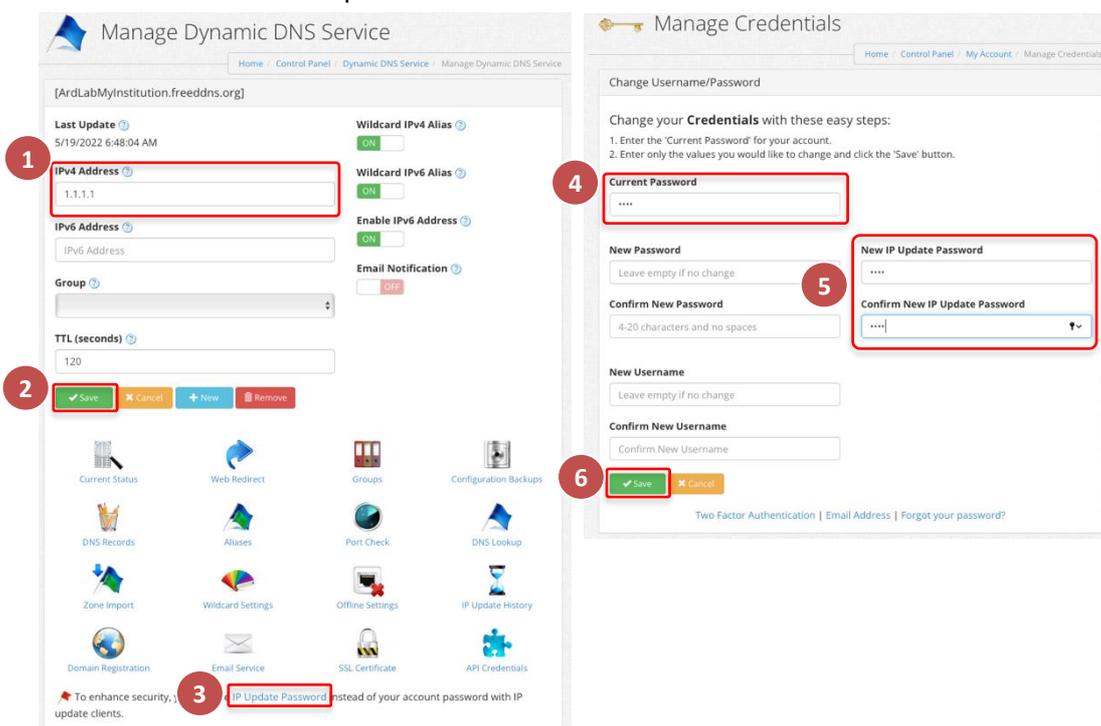


Figura 3 -Gestionar tu registro ArdLAB dDNS

### 3.2.4 Configurar el ArdLAB para actualizar automáticamente la dirección IP de Dynu

Ahora tenemos que configurar la raspberry para publicar la dirección pública ArdLAB al sitio Dynu (instrucciones en <https://www.dynu.com/DynamicDNS/IPUpdateClient/RaspberryPi-Dynamic-DNS>).

- En el directorio principal de la raspberry crea un nuevo directorio llamado dynudns y cambia a ese directorio

```
cd ~
mkdir dynudns
cd dynudns
```

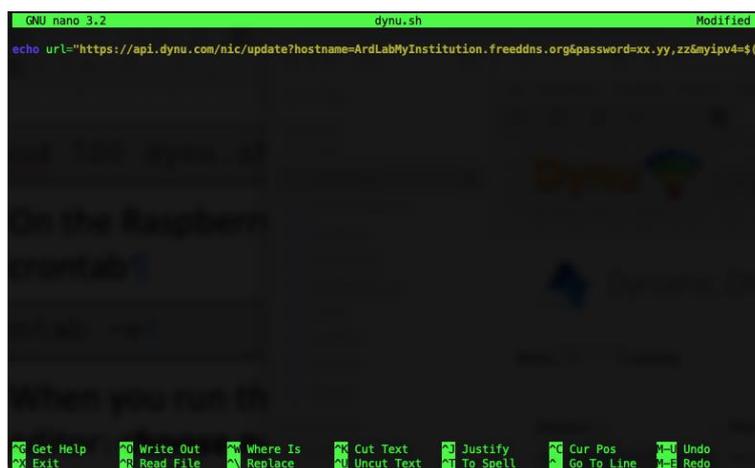
- Crea un nuevo guión llamado dynu.sh

```
nano dynu.sh
```

- Agrega esta línea al guión `dynu.sh`, cambiando `HOST` por el “Dominio” que creaste en el sitio *Dynu* y `IP_PASS` por la contraseña “*IP update password*” que definiste en el sitio *Dynu* .

```
echo
url="https://api.dynu.com/nic/update?hostname=HOST&password=IP_PASS&myipv4=$(curl -s -4
ifconfig.co)&myipv6=$(curl -s -6 ifconfig.co) " | curl
-k -o ~/dynudns/dynu.log -K -
```

Comprueba que el comando está escrito en una sola línea, y que el nombre de host y la contraseña están bien



The screenshot shows the nano 3.2 text editor with the file `dynu.sh` open. The content of the file is: `echo url="https://api.dynu.com/nic/update?hostname=ArdLabMyInstitution.freedom.org&password=xx.yy,zz&myipv4=$(`. The editor interface includes a status bar at the bottom with various navigation and editing shortcuts.

Pulsa `CTRL+X` para guardar, pulsa `Y` para confirmar que se guarda el documento, y luego `ENTER` para confirmar el nombre del archivo

- Cambia los permisos del guión a 700

```
chmod 700 dynu.sh
```

- En el terminal de la Raspberry Pi, ejecuta el siguiente comando, para añadir el guión a tu `crontab`

```
crontab -e
```

Cuando ejecutes este comando por primera vez, debes seleccionar tu editor de texto favorito: **elige nano**

- Pega la siguiente línea en la parte inferior del editor de texto

```
* /5 * * * * ~/dynudns/dynu.sh >/dev/null 2>&1
```

Pulsa CTRL+X para guardar, pulsa Y para confirmar que se guarda el documento, y luego ENTER para confirmar el nombre del archivo

```
GNU nano 3.2 /tmp/crontab.7nMj51/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
*/5 * * * * ~/dynudns/dynu.sh >/dev/null 2>&1

Get Help  Write Out  Where Is  Cut Text  Justify   Cur Pos   Undo
Exit      Read File  Replace   Uncut Text To Spell  Go To Line Redo
```

Ahora tienes un cronjob que se ejecuta cada 5 minutos y actualizará tu IP pública en el servidor dDNS de Dynu. Puedes comprobar en tu subdominio Dynu, para ver si la dirección IP cambia en los próximos 5 minutos.

Si la IP no se actualiza, puedes ver el archivo `~/dynudns/dynu.log` y comprobar el error. Si obtienes `badauth`, significa que has suministrado un dominio o una contraseña de actualización de IP equivocados.

### 3.3 Emisión y renovación de certificados SSL

Para acceder al servidor usando HTTPS, se necesita un certificado SSL válido. Para simplificar la obtención y renovación de los certificados, se necesita una herramienta de automatización de certificados. Utilizaremos el guión `acme.sh` (ACME Shell), que puede obtener y renovar certificados de varias Autoridades de Certificación (CAs) y utilizar varios “Challenge Types” (Tipos de Desafío) para confirmar la propiedad del dominio.

#### 3.3.1 Instala el guión `acme.sh` para automatizar la renovación del certificado

- Instala el guión `acme.sh`

```
cd ~
curl https://get.acme.sh | sh -s email=my@example.com
```

- Tras la instalación, debes cerrar el terminal actual y volver a abrirlo para que el alias surta efecto. Inicia sesión de nuevo y ejecuta el siguiente comando

```
acme.sh --version
```

Si ves la versión `acme.sh`, el guión está correctamente instalado y puedes proceder a generar el certificado.

**Nota:** Si tienes una versión antigua de Ardlab, ahora tendrás que actualizar el contenedor.

```
cd ~/ardlab-setup
git pull
docker-compose down
docker-compose pull
docker-compose up --detach
```

### 3.3.2 Generar el certificado para ArdLAB en la IP privada (utilizando el reenvío de puertos)

Si tienes una IP privada y has creado un registro dDNS de Dynu, ahora tienes que generar el certificado utilizando un desafío DNS (si tienes una dirección IP pública, continúa con la sección 3.3.3.)

En tu directorio principal, ejecuta los siguientes comandos, sustituyendo las `ClientId` y `Secret` con sus claves api, obtenidas en la sección “*Manage Dynamic DNS Service*” → “*api credentials*” de Dynu (ver Figura 3). También necesitas reemplazar el `DOMAIN` después del `-d` con tu propio dominio Dynu dDNS.

**Nota:** Sólo es necesario exportar las credenciales en la primera ejecución, porque `acme.sh` almacena las claves en sus archivos de configuración.

- Estos pasos de configuración se presentan en la página de `acme.sh` “*How to use DNS API*” (<https://github.com/acmesh-official/acme.sh/wiki/dnsapi#24-use-dynu-api>)

```
export Dynu_ClientId=xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxxx
export Dynu_Secret=yyyyyyyyyyyyyyyyyyyyyyyy
acme.sh --issue --dns dns_dynu -d DOMAIN \
    --cert-file /home/pi/ardlab-setup/certs/cert.pem \
    --key-file /home/pi/ardlab-setup/certs/key.pem \
    --fullchain-file /home/pi/ardlab-
setup/certs/fullchain.pem \
    --reloadcmd "docker restart ardlab-setup_ardlab_1"
```

- Una vez finalizado el comando, puedes comprobar los certificados emitidos con el siguiente comando

```
acme.sh -list
```

Ahora puedes continuar con la sección 0.

### 3.3.3 Generar el certificado en ArdLAB con IP pública

- Para generar el certificado en el ArdLAB con IP pública, es necesario instalar el `socat`

```
sudo apt install socat
```

- Entonces debes permitir que `socat` acepte conexiones en puertos restringidos

```
sudo setcap 'cap_net_bind_service=+ep' /usr/bin/socat
```

- Por último, puedes ejecutar el `acme.sh` para solicitar el nuevo certificado. No olvides sustituir `DOMAIN` después de `-d` por tu propio dominio ArdLAB.

```
acme.sh --issue --standalone -d DOMAIN \
    --cert-file /home/pi/ardlab-setup/certs/cert.pem \
    --key-file /home/pi/ardlab-setup/certs/key.pem \
    --fullchain-file /home/pi/ardlab-
setup/certs/fullchain.pem \
    --reloadcmd "docker restart ardlab-setup_ardlab_1"
```



Co-funded by the  
Erasmus+ Programme  
of the European Union

- Una vez finalizado el comando, puedes comprobar los certificados emitidos con el siguiente comando

```
acme.sh -list
```

## 4 Añadir el ArdLAB a la Constelación

Para añadir el ArdLAB a la Constelación Sys-Stem ArdLAB, debes seguir estos pasos:

- Crea una cuenta de usuario de la Constelación ArdLAB (sólo si no tienes una)
- Añade tu ArdLAB a la Constelación (debe ser aprobado por el administrador de la Constelación)
- Ejecuta la configuración de ArdLAB
- Ejecuta el contenedor ArdLAB
- Añade sketches de muestra y prueba tu ArdLAB en el Sitio de la Constelación

### 4.1 Configura el ArdLAB en el Constellation Hub

Para añadir tu ArdLAB al Constellation Hub, necesitas tener una cuenta y luego añadir los parámetros de tu ArdLAB al hub.

#### 4.1.1 Crear una cuenta de usuario de ArdLAB Constellation

- Accede al sitio de la Constelación ArdLAB SYS-STEM (<https://sys-stem.dei.isep.ipp.pt/>)
- Si ya tienes una cuenta, inicia sesión; de lo contrario, utiliza el botón de registro para crear una nueva cuenta de usuario.

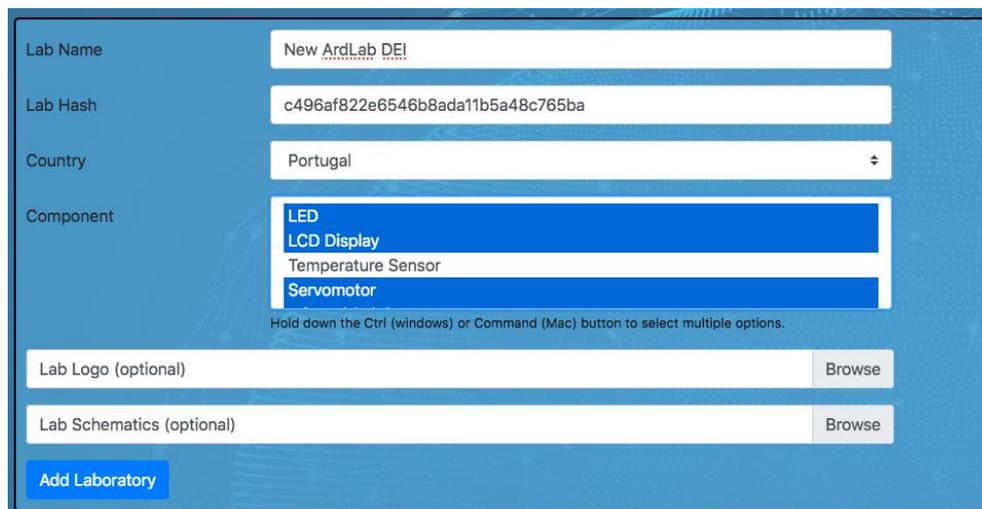


#### 4.1.2 Añade tu ArdLAB a la Constelación

- Selecciona la opción de menú "Add your ArdLAB" → "Add your Laboratory to the Constellation".



- Rellena el formulario New ArdLAB, utiliza el hash creado anteriormente, y pulsa el botón "Add Laboratory".



El Laboratorio necesita ser aprobado por uno de los administradores de la Constelación SYS-STEM, por lo que antes de proceder a la siguiente configuración, tienes que esperar a recibir la autorización (te llegará por correo electrónico después de la autorización de ArdLAB).

## 4.2 Ejecuta la configuración de ArdLAB

- Ahora podemos editar el archivo de configuración de ArdLAB, para configurar el nombre DNS de ArdLAB, el puerto y la cámara

```
cd ~/ardlab-setup
sudo nano config/congif.cfg
```

- A continuación se presenta una muestra de config.cfg con algunas de las configuraciones más habituales

```
ARDLAB_HASH="a826b23115b84f0d8c0d040b89739b"
CONSTELLATION="https://sys-
stem.dei.isep.ipp.pt:8080/api/lab/"
HOSTNAME="ardlab.myInstitution.com"
FORWARD_PORT=7575
# Camera connected to the RaspBerry PI, options are:
#   "pi"      - Use pi as video provider
#   "OpenCV" - Use OpenCV as video provider
#CAMERA_TYPE="pi"
CAMERA_TYPE="OpenCV"
OPENCV_CAMERA_SOURCE="0"
```

Los parámetros que puedes cambiar son:

**HOSTNAME** - es el nombre que has registrado en el servidor DNS para tu ArdLAB

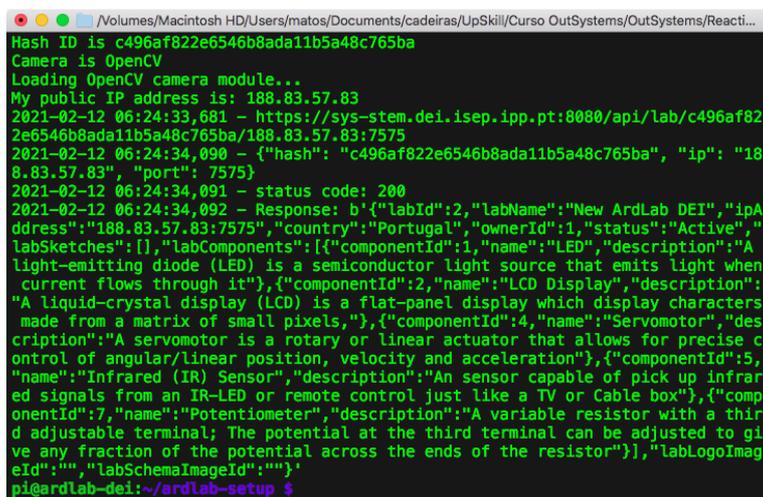
**FORWARD\_PORT** - es el puerto que reenvía en tu router NAT. Sólo es necesario añadir esta entrada si el puerto del router NAT es distinto del 7575

**CAMERA\_TYPE** - utiliza PI para la cámara Raspberry PI y OpenCV para una cámara USB

- Cuando se apruebe el Laboratorio, hay que volver a ejecutar el contenedor para “conectarse” a la Constelación

```
docker-compose run --rm --entrypoint "python3
constellation.py" ardlab
```

Si el comando tiene éxito, el guión `constellation.py` presenta la siguiente información



```
Hash ID is c496af822e6546b8ada11b5a48c765ba
Camera is OpenCV
Loading OpenCV camera module...
My public IP address is: 188.83.57.83
2021-02-12 06:24:33,681 - https://sys-stem.dei.isep.ipp.pt:8080/api/lab/c496af82
2e6546b8ada11b5a48c765ba/188.83.57.83:7575
2021-02-12 06:24:34,090 - {"hash": "c496af822e6546b8ada11b5a48c765ba", "ip": "18
8.83.57.83", "port": 7575}
2021-02-12 06:24:34,091 - status code: 200
2021-02-12 06:24:34,092 - Response: b'{"labId":2,"labName":"New ArdLab DEI","ipA
ddress":"188.83.57.83:7575","country":"Portugal","ownerId":1,"status":"Active","
labSketches":[],"labComponents":[{"componentId":1,"name":"LED","description":"A
light-emitting diode (LED) is a semiconductor light source that emits light when
current flows through it"}, {"componentId":2,"name":"LCD Display","description":
"A liquid-crystal display (LCD) is a flat-panel display which display characters
made from a matrix of small pixels,"}, {"componentId":4,"name":"Servomotor","des
cription":"A servomotor is a rotary or linear actuator that allows for precise c
ontrol of angular/linear position, velocity and acceleration"}, {"componentId":5,
"name":"Infrared (IR) Sensor","description":"An sensor capable of pick up infrar
ed signals from an IR-LED or remote control just like a TV or Cable box"}, {"comp
onentId":7,"name":"Potentiometer","description":"A variable resistor with a thir
d adjustable terminal; The potential at the third terminal can be adjusted to gi
ve any fraction of the potential across the ends of the resistor"}],"labLogoImag
eId":"","labSchemaImageId":""}'
pi@ardlab-dei:~/ardlab-setup $
```

#### 4.2.1 Ejecuta el contenedor ArdLAB

- Por último, se puede ejecutar el contenedor ArdLAB para servir la cámara y la API de programación

```
docker-compose up --detach
```

El contenedor se reiniciará cuando la Raspberry PI se inicie o si el servidor web del contenedor falla.

- Si necesitas detener el contenedor, utiliza el comando

```
docker-compose down
```

- Si en un futuro necesitas actualizar el contenedor, debes hacerlo (después de bajar el contenedor):

```
docker-compose pull
```

La actualización es silenciosa y puede tardar algunos minutos. Después de actualizar el contenedor, debes ejecutar el contenedor de nuevo, utilizando el primer comando de esta sección.

Puedes comprobar si el ArdLAB está funcionando, accediendo al siguiente enlace:

<https://ArdLAB-DNS-Name:7575>