

IES Valle Inclán



ALTA DISPONIBILIDAD

Y

DOCKER



NOMBRE Y APELLIDOS DEL AUTOR

Carlos González Martín y Rocío Ceballos Mateos.

ÍNDICE

1. Introducción.

2. Configuración de la Máquina y Preparación del Entorno.

3. Instalación de Docker.

4. Instalación de Portainer.

5. Implementación de MySQL en Docker con Portainer.

6. Instalación de Docker-Compose y Pi-hole.

7. Implementación de WordPress con Docker-Compose.

8. Extra: Script de Instalación Automática.

9. Conclusión.

1.INTRODUCCIÓN.

En el mundo de la tecnología, es importante encontrar formas más fáciles y eficientes de trabajar con programas y servidores. Docker es una herramienta que ayuda a organizar y ejecutar aplicaciones sin necesidad de instalarlas directamente en una computadora, permitiendo que funcionen en cualquier lugar sin problemas de compatibilidad.

Este trabajo explica paso a paso cómo instalar y utilizar Docker, además de algunas herramientas adicionales que hacen su uso más sencillo. Se verán ejemplos prácticos como la instalación de bases de datos, la gestión de aplicaciones y la administración de contenedores de manera visual.

El objetivo es aprender a utilizar Docker de una forma clara y sencilla, mostrando cómo puede facilitar el trabajo en informática y ayudar a organizar mejor los programas en un servidor.

2. Configuración de la Máquina y Preparación del Entorno.

- **Asignar Dirección IP**

Para facilitar la administración de la máquina, nos aseguramos de que tenga una dirección IP fija dentro de la red. Esto es útil para evitar que cambie cada vez que se reinicie y así poder conectarnos sin problemas.

```
Debian GNU/Linux 12 debian-12 tty1
debian-12 login: root
Password:
Linux debian-12 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

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: Mon Sep 23 09:05:51 CEST 2024 on tty1
root@debian-12:~# echo "\4" >> /etc/issue
root@debian-12:~# exit
```

- **Cambiar el Nombre de la Máquina**

Cambiar el hostname de la máquina es importante para identificarla de manera más sencilla dentro de la red. Esto se puede hacer con el comando:

```
Debian GNU/Linux 12 debian-12 tty1
192.168.1.107
debian-12 login:
```

```
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Feb 13 20:59:48 CET 2025 on tty1
root@dockey:~# hostnamectl set-hostname dockey_
```

- **Instalar los Paquetes Necesarios (curl y ssh)**

Antes de instalar Docker, nos asegurarnos de que tenemos instaladas algunas herramientas básicas:

- curl nos permitirá descargar archivos desde internet.
- ssh nos permitirá conectarnos de forma remota a la máquina.

Instalamos ambos paquetes con el siguiente comando:

sudo apt update && sudo apt install -y curl ssh

```
root@docker:~# apt update ; apt install curl ssh
Des:1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Des:2 http://deb.debian.org/debian bookworm InRelease [151 kB]
Des:3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Des:4 http://security.debian.org/debian-security bookworm-security/main Sources [145 kB]
Des:5 http://security.debian.org/debian-security bookworm-security/main amd64 Packages [245 kB]
Des:6 http://security.debian.org/debian-security bookworm-security/main Translation-en [146 kB]
Des:7 http://deb.debian.org/debian bookworm/non-free-firmware Sources [6.436 B]
Des:8 http://deb.debian.org/debian bookworm/main Sources [9.496 kB]
Des:9 http://deb.debian.org/debian bookworm/main amd64 Packages [8.792 kB]
Des:10 http://deb.debian.org/debian bookworm/main Translation-en [6.109 kB]
Des:11 http://deb.debian.org/debian bookworm/non-free-firmware amd64 Packages [6.240 B]
Des:12 http://deb.debian.org/debian bookworm-updates/main Sources.diff/Index [15,1 kB]
Ign:12 http://deb.debian.org/debian bookworm-updates/main Sources.diff/Index
Des:13 http://deb.debian.org/debian bookworm-updates/main amd64 Packages.diff/Index [15,1 kB]
Des:14 http://deb.debian.org/debian bookworm-updates/main Translation-en.diff/Index [15,1 kB]
Des:15 http://deb.debian.org/debian bookworm-updates/main amd64 Packages T-2025-01-14-2009.05-F-2024-11-27-1405.46.pd
Des:15 http://deb.debian.org/debian bookworm-updates/main amd64 Packages T-2025-01-14-2009.05-F-2024-11-27-1405.46.pd
Des:16 http://deb.debian.org/debian bookworm-updates/main Translation-en T-2025-01-14-2009.05-F-2024-11-27-1405.46.pd
Des:16 http://deb.debian.org/debian bookworm-updates/main Translation-en T-2025-01-14-2009.05-F-2024-11-27-1405.46.pd
```

- **Conectarse a la Máquina por SSH**

Para administrar la máquina de manera remota, nos conectaremos mediante SSH usando: **ssh usuario@IP**

```
C:\Users\carlo>ssh usuario@192.168.1.107
usuario@192.168.1.107's password:
Linux docker 6.1.0-25-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.106-3 (2024-08-26) x86_64

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: Thu Feb 13 21:52:56 2025 from 192.168.1.92
usuario@docker:~$
```

3. Instalación de Docker

- **Descargar el Script de Instalación de Docker**

Docker proporciona un script de instalación automática. Lo descargamos con:

curl -fsSL https://get.docker.com -o get-docker.sh

```
usuario@docker:~$ sudo curl -fsSL https://get.docker.com/ -o get-docker.sh
usuario@docker:~$ ls -l
total 24
-rw-r--r-- 1 root root 22592 feb 13 21:54 get-docker.sh
usuario@docker:~$
```

- **Dar Permisos y Ejecutar el Script**

Para poder ejecutarlo, le damos permisos con:

chmod +x get-docker.sh
./get-docker.sh

```
usuario@docker:~$ sudo chmod +x get-docker.sh
usuario@docker:~$ ./get-docker.sh
# Executing docker install script, commit: 4c94a56999e10efcf48c5b8e3f6afea464f9108e
+ sudo -E sh -c apt-get -qq update >/dev/null
+ sudo -E sh -c DEBIAN_FRONTEND=noninteractive apt-get -y -qq install ca-certificates curl >/dev/null
```

- **Docker Instalado**

Una vez ejecutado el script, Docker quedará instalado en nuestro sistema.

```
=====
To run Docker as a non-privileged user, consider setting up the
Docker daemon in rootless mode for your user:

    dockerd-rootless-setuptool.sh install

Visit https://docs.docker.com/go/rootless/ to learn about rootless mode.

To run the Docker daemon as a fully privileged service, but granting non-root
users access, refer to https://docs.docker.com/go/daemon-access/

WARNING: Access to the remote API on a privileged Docker daemon is equivalent
to root access on the host. Refer to the 'Docker daemon attack surface'
documentation for details: https://docs.docker.com/go/attack-surface/
=====
usuario@docker:~$
```

- **Añadir el Usuario al Grupo Docker**

Para poder ejecutar Docker sin necesidad de usar `sudo`, añadimos nuestro usuario al grupo Docker con:

sudo usermod -aG docker \$USER

```
usuario@docker:~$ sudo usermod -aG docker usuario
usuario@docker:~$
```

```
usuario@docker:~$ groups usuario
usuario : usuario cdrom floppy sudo audio dip video plugdev users netdev docker
usuario@docker:~$
```

- **Reiniciar el Sistema**

Reiniciamos la máquina para aplicar los cambios de permisos y grupos:

sudo reboot

```
usuario@docker:~$ sudo reboot

Broadcast message from root@docker on pts/1 (Thu 2025-02-13 21:58:44 CET):

The system will reboot now!

usuario@docker:~$ Connection to 192.168.1.107 closed by remote host.
Connection to 192.168.1.107 closed.

C:\Users\carlo>
```

- **Probar con el Contenedor "Hello-World"**

Para comprobar que Docker funciona correctamente, ejecutamos el siguiente comando:

docker run hello-world

Si Docker está instalado correctamente, veremos un mensaje de bienvenida.

```
usuario@docker:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:e0b569a5163a5e6be84e210a2587e7d447e08f87a0e90798363fa44a0464a1e8
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.
   (amd64)
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/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

usuario@docker:~$
```

4. Instalación de Portainer

- **Descargar y Ejecutar el Contenedor de Portainer**

Portainer es una interfaz gráfica para administrar Docker. Lo instalamos con:

docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest

```
usuario@docker:~$ docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
Unable to find image 'portainer/portainer-ce:latest' locally
latest: Pulling from portainer/portainer-ce
dc8df9f2921e: Pull complete
c82aa9c9fb45: Pull complete
d48df1cd7a: Pull complete
a3939f2dc487: Pull complete
284b27bb324e: Pull complete
a53c848f28bf: Pull complete
9e1dad44be73: Pull complete
6f01ec19a2b: Pull complete
e2f767fa3835: Pull complete
793e77bf862e: Pull complete
4f4fb780ef54: Pull complete
Digest: sha256:bd8f7a6d98e2a512e18272c38914abdie92d663451f3c925d582a8557a3b92d7
Status: Downloaded newer image for portainer/portainer-ce:latest
42d983cdd77245819900783cd80275b395e5e8acdc99f1f0c87d28da5a8905f7
usuario@docker:~$
```

- **Ver si Portainer Está Corriendo**

Podemos verificar si el contenedor está en ejecución con:

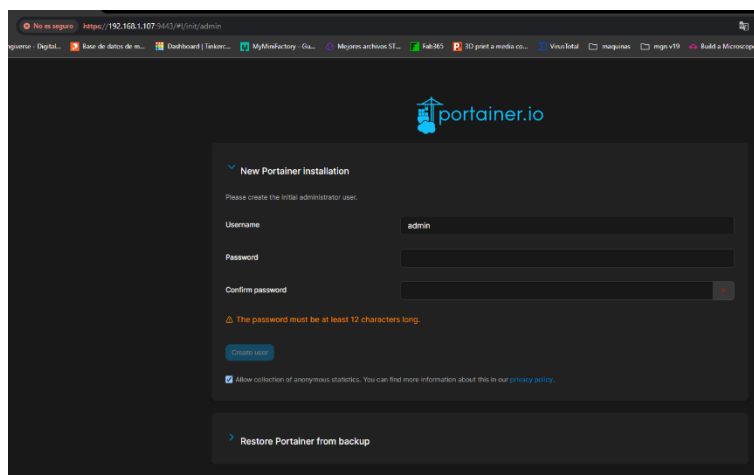
docker ps

```
usuario@docker:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
42d983cdd772   portainer/portainer-ce:latest       "/portainer"           36 seconds ago Up 34 seconds 0.0.0.0:8000->8000/tcp, :
::8000->8000/tcp, 0.0.0.0:9443->9443/tcp, ::9443->9443/tcp, 9000/tcp   portainer
usuario@docker:~$
```

- **Conectar a Portainer vía Web**

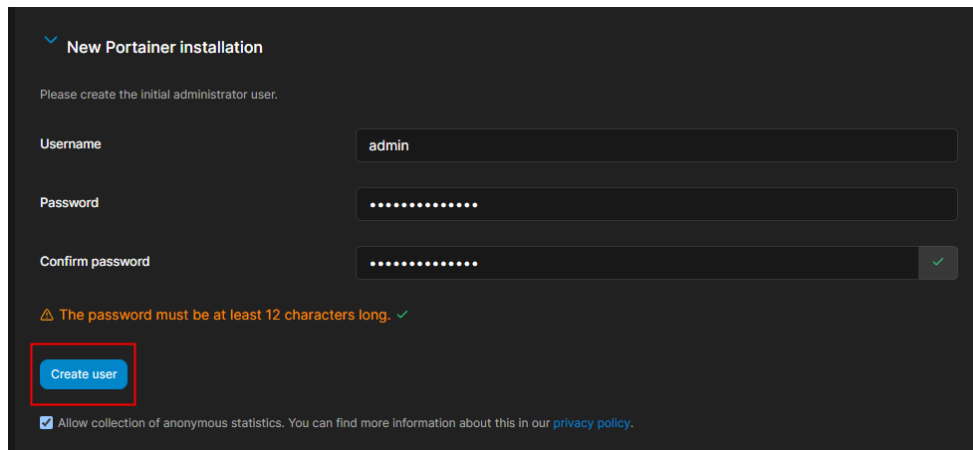
Abrimos un navegador y accedemos a la interfaz web con:

https://IP:9443



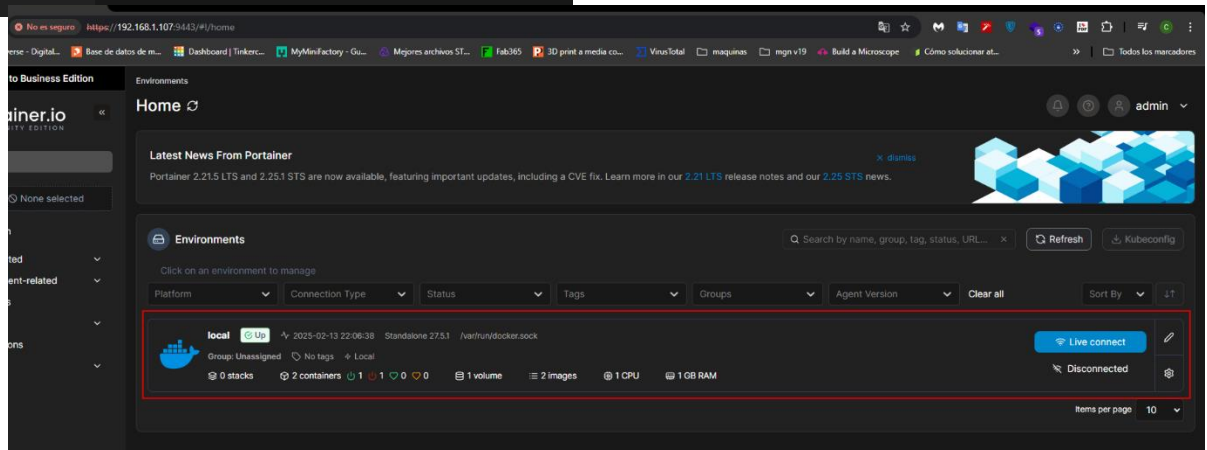
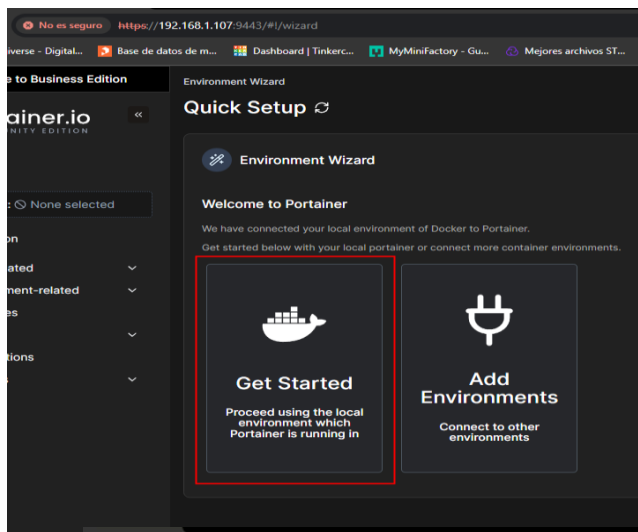
- **Crear Usuario en Portainer**

En la primera ejecución, debemos establecer una contraseña segura y hacer clic en "Create User".



- **Acceder a la Interfaz de Portainer**

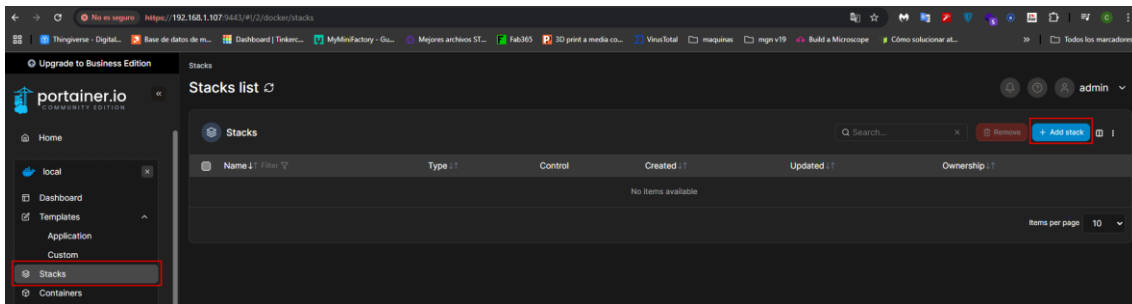
Después de iniciar sesión, hacemos clic en "Get Started" para acceder al panel principal.



5. Implementación de MySQL en Docker con Portainer

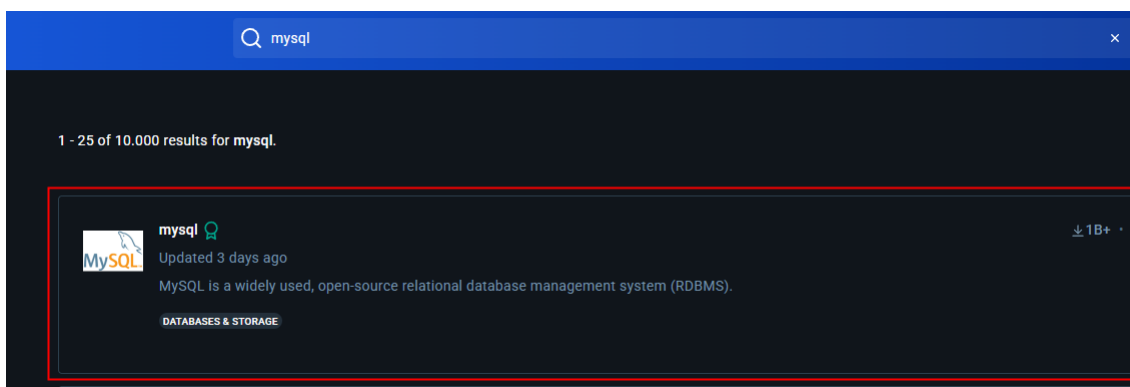
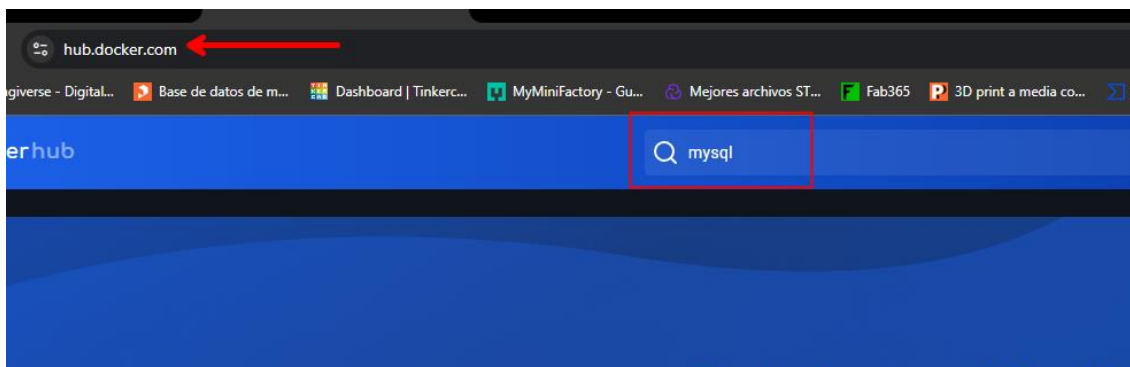
- **Ir a Stacks y Añadir un Stack**

En Portainer, nos dirigimos a la sección "Stacks" y añadimos un nuevo stack.

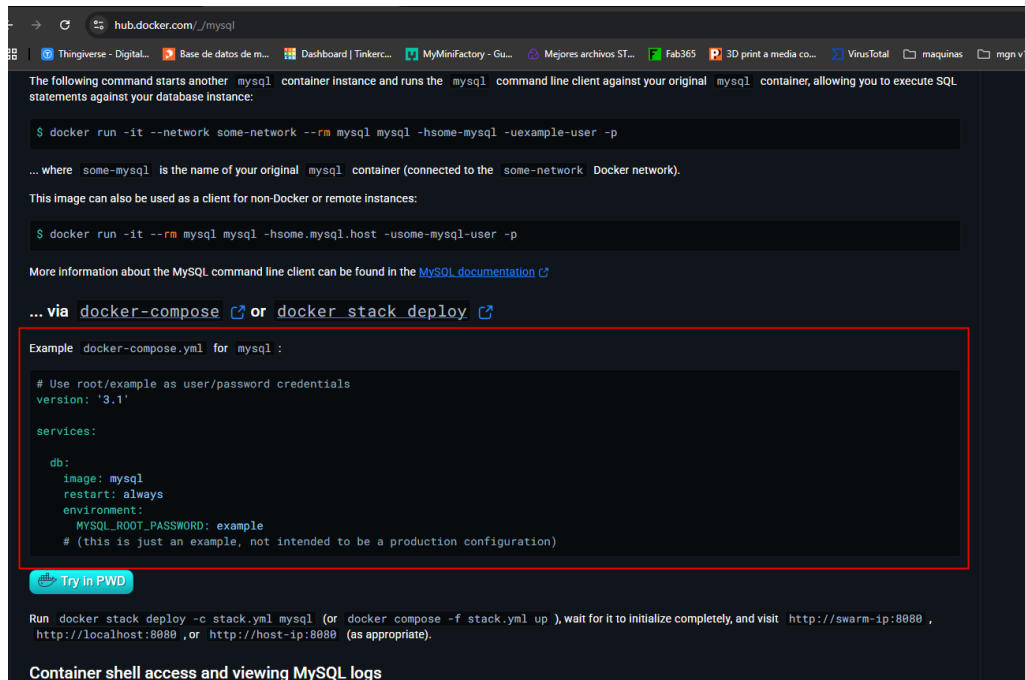


- **Buscar MySQL en Docker Hub**

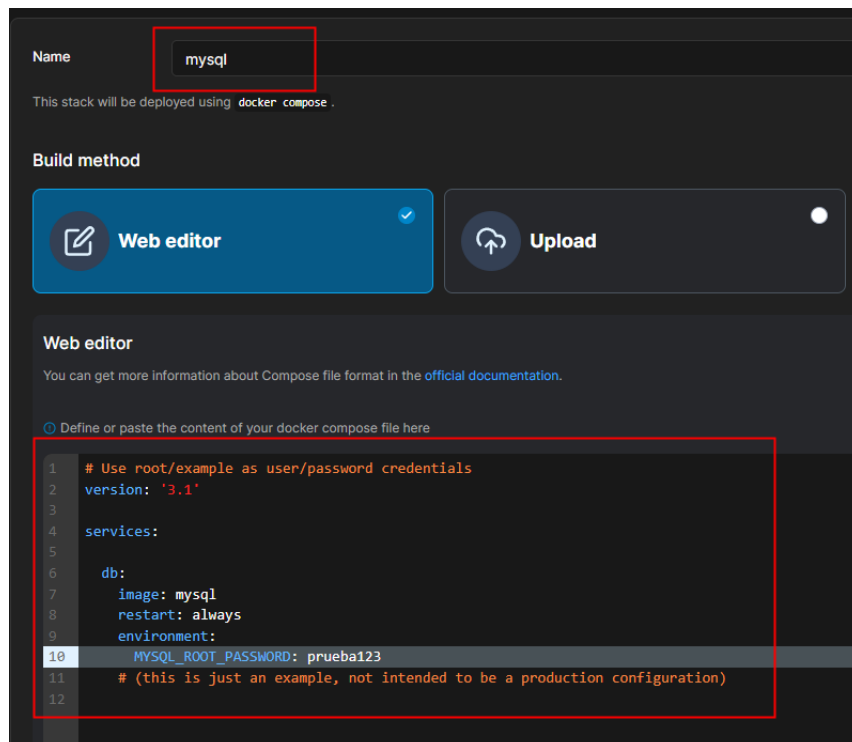
En la web hub.docker.com buscamos la imagen oficial de MySQL.



- **Copiar la Configuración del “docker-compose.yml”**
Desde la página de MySQL en Docker Hub, copiamos el código “docker-compose.yml” “que define la configuración del contenedor.

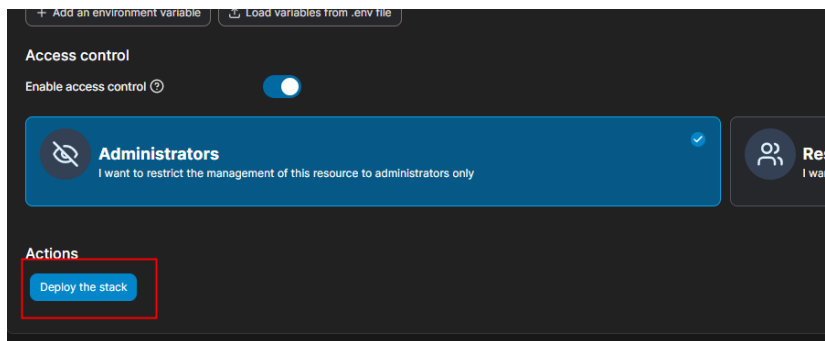


- **Nombrar el Stack y Pegar la Configuración**
En Portainer, asignamos el nombre "mysql" al stack y pegamos la configuración del docker-compose.yml.



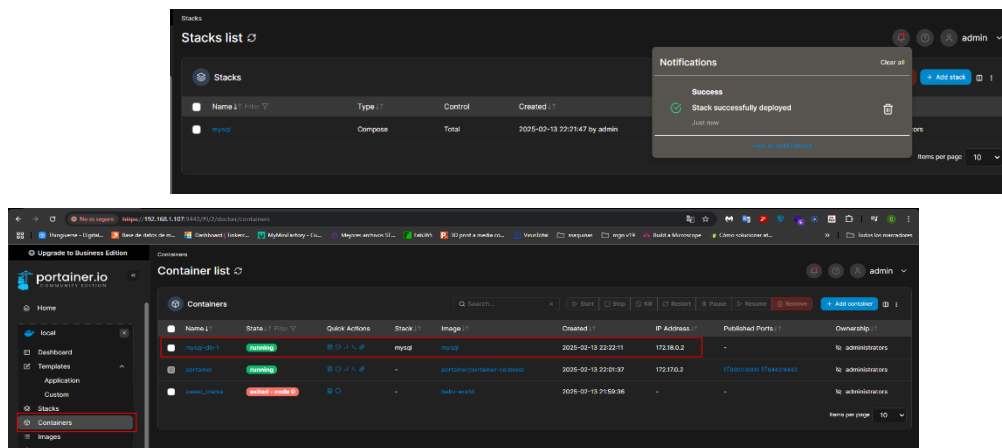
- **Desplegar el Stack**

Bajamos hasta el final de la página y hacemos clic en "Deploy the Stack" para iniciar el servicio.



- **Verificar que MySQL Está en Ejecución**

En la sección "Containers" de Portainer, verificamos que el contenedor de MySQL está funcionando.



- **Conectarse a MySQL Dentro del Contenedor**

Accedemos al contenedor con:

docker exec -it mysql bash

```

ERROR: Response from daemon: No such container: mysql
usuario@docker:~$ docker exec -it mysql-db-1 bash
bash-5.1# mysql -u root
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: NO)
bash-5.1# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 9.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>

```

- **Listar las Bases de Datos en MySQL**

Entramos a MySQL y vemos las bases de datos disponibles con:

mysql -u root -p
SHOW DATABASES;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql>
```

6. Instalación de Docker-Compose y Pi-hole

- **Instalar Docker-Compose**

Instalamos Docker-compose con:

```
sudo curl -L "https://github.com/docker/compose/releases/download/$(curl -s  

https://api.github.com/repos/docker/compose/releases/latest | grep 'tag_name'  

| cut -d\" -f4)/docker-compose-$(uname -s)-$(uname -m)" -o  

/usr/local/bin/docker-compose  

sudo chmod +x /usr/local/bin/docker-compose
```

```
usuario@docker:~$ sudo curl -L "https://github.com/docker/compose/releases/download/$(curl -s https://api.github.com/repos/docker/compose/releases/latest | grep 'tag_name' | cut -d\" -f4)/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100 70.1M  100 70.1M    0     0  33.4M      0  0:00:02  0:00:02 --:--:-- 44.9M
usuario@docker:~$ sudo chmod +x /usr/local/bin/docker-compose
usuario@docker:~$ docker-compose --version
Docker Compose version v2.33.0
usuario@docker:~$
```

- **Instalar Pi-hole con Docker-Compose**

Descargamos y configuramos Pi-hole como un contenedor Docker.

```
GNU nano 7.2 docker-compose.yml
1 # More info at https://github.com/pi-hole/docker-pi-hole/ and https://docs.pi-hole.net/
2 services:
3   pihole:
4     container_name: pihole
5     image: pihole/pihole:latest
6     ports:
7       - "53:53/tcp"
8       - "53:53/udp"
9     # Default HTTP Port
10    - "80:80/tcp"
11    # Default HTTPS Port. FTL will generate a self-signed certificate
12    - "443:443/tcp"
13    # Uncomment the below if using Pi-hole as your DHCP Server
14    #- "67:67/udp"
15  environment:
16    # Set the appropriate timezone for your location (https://en.wikipedia.org/wiki/List_of_tz_database_time_zones), e.g:
17    TZ: 'Europe/Madrid'
18    # Set a password to access the web interface. Not setting one will result in a random password being assigned
19    FTLCONF_webserver_api_password: 'correct horse battery staple'
20    # If using Docker's default 'bridge' network setting the dns listening mode should be set to 'all'
21    FTLCONF_dns_listeningMode: 'all'
22  # Volumes store your data between container upgrades
23  volumes:
24    # For persisting Pi-hole's databases and common configuration file
25    - './etc-pihole:/etc/pihole'
26    # Uncomment the below if you have custom dnsmasq config files that you want to persist. Not needed for most starting fresh with Pi-hole v6. If you're
27    #- './etc-dnsmasq.d:/etc/dnsmasq.d'
28  cap_add:
29    # See https://github.com/pi-hole/docker-pi-hole#note-on-capabilities
30    - NET_ADMIN
31    # Required if you are using Pi-hole as your DHCP server, else not needed
32    - SYS_TIME
33    # Required if you are using Pi-hole as your NTP client to be able to set the host's system time
34    # Optional, if Pi-hole should get some more processing time
35    - SYS_NICE
36  restart: unless-stopped
37
```

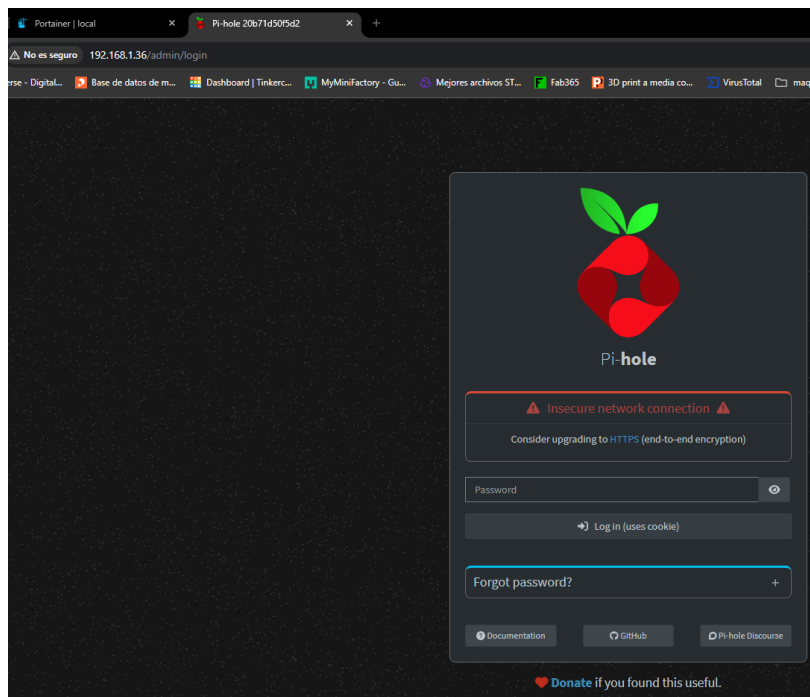
- **Descargar la Imagen de Pi-hole**

Docker-compose up -d

```
usuario@docker:~/piholes$ docker-compose up -d
[+] Running 2/12
  * pi-hole Pulled
    * 83abf496f1b8 Pull complete
    * 5e81a42a3169 Pull complete
    * 4f4fb700ef54 Pull complete
    * 2002978e8b06 Pull complete
    * 6a0c2d37028d Pull complete
    * 90f38802a58f Pull complete
    * ddd77a26e02 Pull complete
    * b6d5e77b1ab Pull complete
    * dc10559df0b4 Pull complete
    * 1bb371d39743 Pull complete
    * 08c048d5368d Pull complete
[+] Running 2/2
  * Network pi-hole_default Created
  * Container pi-hole Started
usuario@docker:~/piholes$ ls
docker-compose.yml  etc-pi-hole
```

- **Comprobar si Pi-hole Funciona**

Accedemos a la IP del servidor en el navegador para ver si está funcionando.



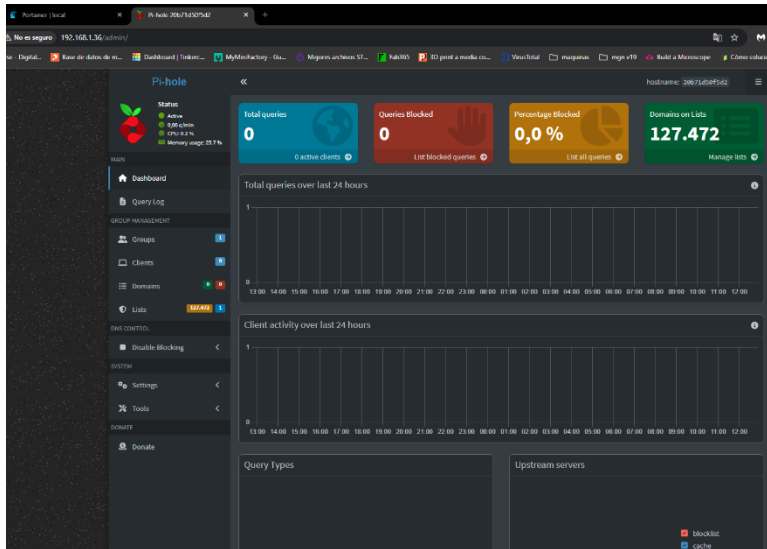
- **Cambiar la Contraseña de Pi-hole con Portainer**

Desde Portainer, accedemos a la configuración y cambiamos la contraseña.

```
20b71d50f5d2:/$ sudo pihole setpassword
Enter New Password (Blank for no password):
Confirm Password:
[✓] New password set
20b71d50f5d2:/$
```

- **Iniciar Sesión en Pi-hole**

Usamos la nueva contraseña para acceder al panel de administración.



Configurar Pi-hole en el Router

En la configuración DHCP del router, establecemos la IP del servidor como DNS primario.

7.Implementación de WordPress con Docker-Compose

- **Ejecutar docker-compose para WordPress**

Usamos docker-compose up -d para desplegar WordPress y su base de datos.

Para evitar conflictos con la instalación de WordPress, eliminamos la instancia anterior de MySQL.

```
GNU nano 7.2 docker-compose.yml
1 services:
2
3   wordpress:
4     image: wordpress
5     restart: always
6     ports:
7       - 8080:80
8     environment:
9       WORDPRESS_DB_HOST: db
10      WORDPRESS_DB_USER: exampleuser
11      WORDPRESS_DB_PASSWORD: examplepass
12      WORDPRESS_DB_NAME: exampledb
13     volumes:
14       - wordpress:/var/www/html
15
16   db:
17     image: mysql:8.0
18     restart: always
19     environment:
20       MYSQL_DATABASE: exampledb
21       MYSQL_USER: exampleuser
22       MYSQL_PASSWORD: examplepass
23       MYSQL_RANDOM_ROOT_PASSWORD: '1'
24     volumes:
25       - db:/var/lib/mysql
26
27 volumes:
28   wordpress:
29   db:
30
```

- **Convertir Archivos con dos2unix (si es necesario)**

Si transferimos archivos desde Windows, convertimos los saltos de línea con:

dos2unix archivo.sh

```

usuario@wordpress:~/wordpress$ dos2unix docker-compose.yml
dos2unix: convirtiendo archivo docker-compose.yml a formato Unix...
usuario@wordpress:~/wordpress$ docker-compose up -d
[+] Running 5/5
  db Pulled
    4375993d4f6 Pull complete
    126d9d3809b3 Pull complete
    157a7468a5d3 Pull complete
    c4d8af899494 Pull complete
    c92147eb3382 Pull complete
    b3ec9882b976 Pull complete
    ad072341f6b0 Pull complete
    bdb7e2eca8d4 Pull complete
    b6b5c46ac97d Pull complete
    8cd34fa224e6 Pull complete
    b4144aa75def Pull complete
  wordpress Pulled
    7cf63256a31a Pull complete
    859c077b5003 Pull complete
    59e01f001c00 Pull complete
    7d7543348a2e Pull complete
    ee6fbc7f6018 Pull complete
    7ac282ed1b18 Pull complete
    ac27beebac1c Pull complete
    849a107069e4 Pull complete
    a02f50cc1f1f Pull complete
    897474ecb9dc Pull complete
    7b3a964a341f Pull complete
    5b3467e9e61d Pull complete
    9e276231f1e4 Pull complete
    4f4fb700ef54 Pull complete
    115a3dfab727 Pull complete
    4cdeb0350bf5 Pull complete
    0af6e19e3a6a Pull complete
    e9ab1aac14f2 Pull complete
    2e0cbc7f9407 Pull complete
    9b2cf0brcbcd Pull complete
    edf9e7080704 Pull complete
    33559abf5deb Pull complete
[+] Running 5/5
  Network wordpress_default Created
  Volume "wordpress_db" Created
  Volume "wordpress_wordpress" Created
  Container wordpress-db-1 Started
  Container wordpress-wordpress-1 Started
usuario@wordpress:~/wordpress$

```

- **Comprobar que WordPress Funciona**

Accedemos al navegador e ingresamos la IP donde está corriendo WordPress.

WordPress Installation

Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Names can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password

JcLt)VJPDY7D*OvXsG Strong

Important: You will need this password to log in. Please store it in a secure location.

Your Email

Double-check your email address before continuing.

Search engine visibility

☐ Discourage search engines from indexing this site

It is up to search engines to honor this request.

Install WordPress

- **Balanceador.**

Mostramos el archivo de configuración docker-compose.yml y nginx.conf. En el docker-compose.yml se han definido los servicios que se ejecutarán en los contenedores, especificando la imagen base, los puertos y las redes utilizadas. En el nginx.conf se ha configurado el balanceador de carga, estableciendo las reglas de redirección del tráfico hacia los distintos servidores.

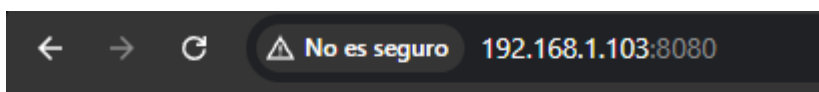
```
GNU nano 7.2 docker-compose.yml
1 version: '3'
2
3 services:
4   web1:
5     image: nginx
6     container_name: web1
7     volumes:
8       - ./web:/usr/share/nginx/html
9     networks:
10      - web_network
11
12   web2:
13     image: nginx
14     container_name: web2
15     volumes:
16       - ./web2:/usr/share/nginx/html #./web2 en caso de diferenciar index.html
17     networks:
18      - web_network
19
20   balancer:
21     image: nginx
22     container_name: balancer
23     volumes:
24       - ./nginx.conf:/etc/nginx/nginx.conf:ro
25     ports:
26       - "8080:80" #si ya se está usando el puerto 80 de la máquina real, cambiar el primer 80 por otro puerto
27     networks:
28      - web_network
29
30 networks:
31   web_network:
32     driver: bridge
33
```

```
GNU nano 7.2 nginx.conf
1 events { }
2
3 http {
4   upstream backend {
5     server web1:80;
6     server web2:80;
7   }
8
9   server {
10    listen 80;
11
12    location / {
13      proxy_pass http://backend;
14      proxy_set_header Host $host;
15      proxy_set_header X-Real-IP $remote_addr;
16      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
17    }
18  }
19 }
20
```


- Ejecutamos el comando `docker-compose up -d` en la terminal. Este comando se encarga de iniciar los contenedores en modo **detached**, permitiendo que se ejecuten en segundo plano. De esta manera, el balanceador de carga y los servicios web quedan en funcionamiento sin necesidad de mantener la terminal abierta.

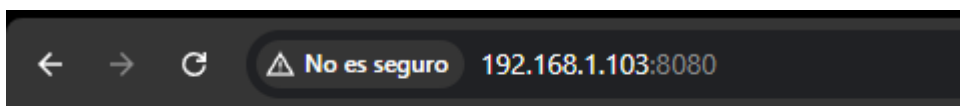
```
usuario@docker:~/nginx$ docker-compose up -d
WARN[0000] /home/usuario/nginx/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 3/3
✔ Container web1      Started    0.75s
✔ Container web2      Started    0.65s
✔ Container balancer   Started    0.65s
usuario@docker:~/nginx$
```

- Mostramos la verificación del servicio accediendo desde un navegador a la dirección IP:8080. Esto confirma que los contenedores han sido iniciados correctamente y que la configuración del balanceador de carga permite acceder a la aplicación web a través de esta dirección.



esto es la web1

- En esta captura se ha recargado la página web, y ahora se puede observar que el tráfico ha sido redirigido a web2. Esto valida que el balanceador de carga está funcionando correctamente, distribuyendo las solicitudes entre los diferentes servidores configurados.



esto es la web2

8.Extra: Script de Instalación Automática

Script de Instalación de Docker, Docker-Compose y Portainer

Para automatizar todo el proceso, cree (Carlos) (un pequeño script para instalar Docker, Docker-compose y portainer

- podemos usar el script disponible en:
https://github.com/C4rl0s1103/Menu_docker.

9.Conclusión

Docker ha facilitado la manera en que se instalan y gestionan programas, permitiendo que funcionen de forma ordenada y sin afectar el sistema principal. Su uso ahorra tiempo y evita problemas al mover aplicaciones entre diferentes máquinas o servidores.

Además, herramientas como Portainer hacen que su administración sea más sencilla y accesible, incluso para quienes no tienen experiencia previa. Gracias a su flexibilidad, Docker se ha convertido en una opción muy útil para organizar y optimizar el trabajo con distintos programas.

LEGISLACIÓN



ÍNDICE

1. PROTECCIÓN DE LA INFORMACIÓN EN MEDIOS DIGITALES Y PROTECCIÓN DE DATOS
2. RGPD, LA LOPDGDD Y LA CCPA ORIENTADA A LA INFORMÁTICA
3. ISO 27001: SEGURIDAD DE LA INFORMACIÓN EN INFORMÁTICA
4. CONCLUSIÓN

1. Protección de la Información en Medios Digitales y Protección de Datos

La protección de la información en medios digitales es fundamental para garantizar la seguridad y privacidad de los datos personales y corporativos. En la era digital, la información puede ser vulnerable a ataques cibernéticos, filtraciones y accesos no autorizados. Por ello, es esencial aplicar medidas de protección alineadas con las normativas de protección de datos, como el Reglamento General de Protección de Datos (RGPD) en Europa y leyes específicas en otros países.

Principios de la Protección de Datos

Los principios clave que rigen la protección de datos en medios digitales incluyen:

- Licitud, lealtad y transparencia: El tratamiento de datos debe ser legal y transparente para el usuario.
- Limitación de la finalidad: Los datos deben ser recopilados con un propósito específico y no usarse para fines distintos sin consentimiento.
- Minimización de datos: Solo deben recolectarse los datos estrictamente necesarios.
- Exactitud: Los datos deben ser precisos y actualizarse regularmente.
- Limitación de almacenamiento: No se deben conservar los datos más tiempo del necesario.
- Integridad y confidencialidad: Se deben proteger los datos contra accesos no autorizados y ataques cibernéticos.

Medidas de Protección en Medios Digitales

Para garantizar la seguridad de los datos, se deben aplicar medidas técnicas y organizativas adecuadas:

a) Seguridad Informática

- Cifrado de datos: Utilización de algoritmos de cifrado para proteger información en tránsito y almacenamiento.
- Autenticación de doble factor (2FA): Añadir una segunda capa de seguridad en inicios de sesión.
- Actualización de software y sistemas: Mantener los sistemas operativos, aplicaciones y antivirus actualizados.
- Copias de seguridad (Backups): Realizar respaldos periódicos en servidores seguros o almacenamiento en la nube con cifrado.
- Control de accesos: Asignar permisos y roles adecuados a cada usuario para limitar el acceso a información sensible.

b) Protección en Redes y Dispositivos

- Uso de VPN: Para proteger la transmisión de datos en redes públicas.
- Firewalls y antivirus: Para prevenir ataques cibernéticos y malware.
- Seguridad en dispositivos móviles: Implementar políticas de seguridad para evitar accesos no autorizados a información empresarial.

c) Normativas y Buenas Prácticas

- Cumplimiento del RGPD, LOPDGDD y CCPA: Adaptar las políticas de privacidad a las normativas vigentes.
- Uso de políticas de privacidad claras: Informar a los usuarios sobre el uso de sus datos y obtener su consentimiento explícito.
- Gestión de incidentes: Contar con un plan de respuesta a incidentes de seguridad en caso de fuga de datos.

Protección de Datos Personales en Internet

Los datos personales son especialmente vulnerables en internet, por lo que es clave protegerlos mediante:

- Uso de contraseñas seguras: Combinación de letras, números y caracteres especiales, evitando datos predecibles.
- Revisión de permisos en aplicaciones y redes sociales: Evitar otorgar acceso excesivo a datos personales.
- Evitar compartir información sensible: No divulgar datos personales en sitios web no confiables o redes sociales públicas.
- Cuidado con ataques de phishing: No hacer clic en enlaces sospechosos o compartir información en sitios no verificados.

Protección de Datos en Empresas

Las organizaciones deben aplicar medidas de seguridad adicionales para proteger la información de clientes y empleados:

- Políticas de seguridad de la información: Crear regulaciones internas sobre el manejo de datos.
- Concienciación y formación del personal: Capacitar a empleados sobre buenas prácticas en ciberseguridad.
- Uso de herramientas de protección: Implementar soluciones como DLP (Data Loss Prevention) para evitar fugas de datos.
- Contratos de confidencialidad: Establecer acuerdos para el manejo seguro de la información.

2.RGPD, LA LOPDGDD Y LA CCPA ORIENTADA A LA INFORMÁTICA

RGPD, LA LOPDGDD Y LA CCPA orientadas a la informática abarcando sus implicaciones en el tratamiento de datos, la seguridad de la información y la gestión de sistemas:

a) Reglamento General de Protección de Datos (RGPD)

Ámbito y aplicación:

El RGPD es una normativa europea que regula el tratamiento de datos personales de ciudadanos de la Unión Europea. Aplica tanto a empresas ubicadas en Europa como a aquellas fuera de la UE que procesen datos de residentes europeos.

Implicaciones para la informática:

- **Diseño y desarrollo seguro:** Los sistemas informáticos deben diseñarse desde el “privacy by design”, es decir, incorporando medidas de protección y privacidad en el desarrollo de aplicaciones, bases de datos y servicios en la nube.
- **Cifrado y seguridad:** Es esencial implementar técnicas de cifrado para proteger los datos en tránsito y en reposo, reduciendo el riesgo de accesos no autorizados.
- **Gestión de consentimientos:** Las aplicaciones deben incorporar mecanismos claros para recabar el consentimiento explícito del usuario y gestionar sus preferencias.
- **Notificación de brechas:** En caso de incidentes o vulneraciones, se debe contar con sistemas de detección y protocolos de respuesta que permitan notificar a las autoridades y a los afectados en un plazo máximo de 72 horas.
- **Portabilidad de datos:** Los sistemas deben facilitar la exportación de datos personales a solicitud del usuario, promoviendo la interoperabilidad entre plataformas.

b) Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales (LOPDGDD)

Ámbito y aplicación:

La LOPDGDD es la normativa española que adapta y complementa el RGPD. Incorpora además aspectos específicos sobre los derechos digitales en el ámbito laboral y otros entornos.

Implicaciones para la informática:

- **Derechos digitales:** Los sistemas y plataformas deben contemplar la protección de derechos como el derecho a la desconexión digital, lo cual afecta la gestión de herramientas de comunicación interna y sistemas de videoconferencias.
- **Medidas de seguridad específicas:** Se exige la implementación de medidas técnicas y organizativas para proteger la integridad, confidencialidad y disponibilidad de los datos, lo que incide en la configuración de redes, servidores y software de seguridad.

- Evaluación de riesgos: Los responsables de sistemas deben llevar a cabo evaluaciones de impacto (DPIA) para identificar y mitigar riesgos en el tratamiento de datos personales.
- Acceso y control: La LOPDGDD impulsa el uso de controles de acceso estrictos en sistemas informáticos, garantizando que solo personal autorizado pueda interactuar con datos sensibles.

c) California Consumer Privacy Act (CCPA)

Ámbito y aplicación:

La CCPA es una ley de privacidad de datos vigente en California (EE.UU.) que otorga a los consumidores derechos sobre el uso y la venta de sus datos personales. Aunque se centra en la protección de los residentes de California, muchas empresas globales adoptan medidas similares para cumplir con esta normativa.

Implicaciones para la informática:

- Transparencia en el tratamiento: Las plataformas deben implementar mecanismos para informar a los usuarios sobre qué datos se recopilan, cómo se utilizan y con quién se comparten.
- Derechos de acceso y eliminación: Las aplicaciones deben permitir que los usuarios accedan, soliciten la corrección o eliminen su información personal, lo que requiere una infraestructura que facilite estas operaciones de forma automatizada.
- Seguridad de los datos: La CCPA incentiva la adopción de prácticas de seguridad robustas para evitar brechas que puedan exponer datos personales, afectando la forma en que se gestionan bases de datos, servidores y servicios en la nube.
- Auditorías y cumplimiento: Los sistemas informáticos deben incluir registros y trazabilidad para demostrar el cumplimiento de la normativa, lo que implica desarrollar soluciones de auditoría interna y monitoreo de accesos.

Cada una de estas normativas impone retos específicos a nivel informático: desde el diseño seguro de software y la implementación de mecanismos de cifrado, hasta la necesidad de crear sistemas que faciliten la gestión de consentimientos, la portabilidad y la eliminación de datos.

- El RGPD y la LOPDGDD enfatizan un enfoque proactivo de la privacidad, exigiendo que la protección de datos se integre en el núcleo del desarrollo.

3.ISO 27001: SEGURIDAD DE LA INFORMACIÓN EN INFORMÁTICA

La ISO/IEC 27001 es un estándar internacional para la gestión de la seguridad de la información (SGSI). Su objetivo es establecer un marco de referencia para la protección de datos en sistemas informáticos, empresas y organizaciones. Esta norma ayuda a identificar, evaluar y gestionar los riesgos relacionados con la seguridad de la información, garantizando su confidencialidad, integridad y disponibilidad.

¿Qué es la ISO 27001?

Es una norma publicada por la Organización Internacional de Normalización (ISO) y la Comisión Electrotécnica Internacional (IEC). Se enfoca en la implementación de un Sistema de Gestión de Seguridad de la Información (SGSI), proporcionando directrices y controles para proteger los datos.

Objetivos principales:

- Proteger la información sensible frente a accesos no autorizados.
- Garantizar la integridad de los datos, evitando alteraciones o manipulaciones no deseadas.
- Asegurar la disponibilidad de la información, permitiendo el acceso cuando sea necesario.
- Cumplir con las regulaciones legales y normativas (RGPD, LOPDGDD, CCPA, etc.).
- Mejorar la ciberseguridad en empresas y sistemas informáticos.

Principios Clave en la Seguridad de la Información

La ISO 27001 se basa en el triángulo CIA, que son los tres pilares de la seguridad de la información:

- Confidencialidad: Solo las personas autorizadas pueden acceder a la información.
- Integridad: La información debe ser precisa, completa y no alterada sin autorización.
- Disponibilidad: La información debe estar accesible cuando sea requerida por usuarios autorizados.

Requisitos de la ISO 27001

Para obtener la certificación ISO 27001, una empresa debe cumplir con ciertos requisitos y seguir un proceso estructurado.

a) Implementación del Sistema de Gestión de Seguridad de la Información (SGSI)

El SGSI es el núcleo de la ISO 27001 y debe incluir:

- Política de Seguridad de la Información: Documento donde la empresa define sus objetivos y compromisos en seguridad.
- Análisis de riesgos: Identificar posibles amenazas y vulnerabilidades en la infraestructura informática.
- Evaluación de impacto: Determinar el impacto de una brecha de seguridad en la empresa.
- Plan de mitigación de riesgos: Aplicar medidas de control para minimizar riesgos de seguridad.
- Control de acceso: Definir quién puede acceder a qué información y bajo qué condiciones.
- Gestión de incidentes: Protocolo para actuar en caso de ataque cibernético o pérdida de datos.

b) Controles de Seguridad

La ISO 27001 define 114 controles de seguridad agrupados en 14 dominios, entre ellos:

- Gestión de accesos: Controlar quién tiene acceso a qué información.
- Cifrado de datos: Protección de información mediante criptografía.
- Seguridad en redes: Implementación de firewalls, VPNs y segmentación de redes.
- Protección en la nube: Aplicación de seguridad en entornos cloud.
- Gestión de incidentes: Protocolo de respuesta ante ciberataques.
- Copias de seguridad: Implementación de backups para recuperación de datos.

Beneficios de la ISO 27001 en Informática

Implementar la ISO 27001 en entornos informáticos ofrece múltiples ventajas:

a) Mayor seguridad en sistemas y datos

- Protección frente a ataques informáticos como ransomware, phishing y accesos no autorizados.
- Reducción del riesgo de filtraciones de datos y pérdida de información sensible.

b) Cumplimiento legal

- Facilita el cumplimiento de normativas como RGPD, LOPDGDD y CCPA.
- Evita sanciones económicas por incumplimientos en protección de datos.

c) Mejora de la confianza y reputación

- Empresas certificadas en ISO 27001 generan mayor confianza en clientes y socios.

- Reducción de impactos negativos en caso de ciberataques o fugas de información.

d) Eficiencia y reducción de costes

- Optimización de procesos mediante una gestión estructurada de la seguridad.
- Reducción de costes asociados a incidentes de seguridad o recuperación de datos.

Proceso de Certificación ISO 27001

Para obtener la certificación ISO 27001, una empresa debe seguir estos pasos:

1. Análisis de riesgos: Identificar vulnerabilidades en los sistemas informáticos.
2. Implementación del SGSI: Aplicar controles de seguridad adecuados.
3. Auditoría interna: Evaluar el cumplimiento de la norma dentro de la empresa.
4. Auditoría externa: Un organismo certificador evalúa la empresa y otorga la certificación.
5. Mantenimiento y mejora continua: Monitorear y actualizar las medidas de seguridad.

Comparación entre ISO 27001 y otras Normas de Seguridad

Característica	ISO 27001	NIST (800-53, CSF, etc.)	COBIT	PCI-DSS	GDPR
Enfoque principal	Gestión de seguridad de la información (SGSI)	Controles de seguridad informática	Gobierno de TI y control	Seguridad en datos de tarjetas de pago	Protección de datos personales
Organización responsable	ISO (International Organization for Standardization)	NIST (National Institute of Standards and Technology - EE.UU.)	ISACA	PCI Security Standards Council	Unión Europea
Aplicabilidad	General para cualquier organización	Principalmente para entidades gubernamentales y empresas en EE.UU.	Empresas que buscan gobernanza de TI	Empresas que manejan datos de tarjetas de crédito	Organizaciones que manejan datos personales de ciudadanos de la UE
Certificación disponible	Sí	No (pero se puede usar como marco de referencia)	No (sirve como modelo de madurez)	Sí	No (es un marco regulatorio con cumplimiento obligatorio)
Principales requisitos	Gestión de riesgos, seguridad de la información, mejora continua	Controles de seguridad basados en riesgos, defensa en profundidad	Gestión de procesos de TI alineados con objetivos de negocio	Seguridad en la transmisión, almacenamiento y procesamiento de datos de tarjetas	Consentimiento, derechos de los usuarios, privacidad por diseño
Flexibilidad	Adaptable a diferentes sectores y empresas	Amplio y detallado, pero puede ser complejo	Más enfocado en gobernanza, con alto nivel de abstracción	Estricto y específico para pagos electrónicos	Obligatorio para empresas con datos personales de la UE

4.CONSLUSIÓN

La seguridad de la información y la protección de datos son esenciales en la era digital para prevenir accesos no autorizados, filtraciones y ataques cibernéticos. Para ello, se han desarrollado normativas y estándares como el RGPD, la LOPDGDD, la CCPA e ISO 27001, que establecen principios y medidas para garantizar la confidencialidad, integridad y disponibilidad de la información.

Las organizaciones deben adoptar enfoques proactivos, incorporando seguridad desde el diseño en sus sistemas, aplicando cifrado, controles de acceso y planes de respuesta a incidentes. Además, el cumplimiento normativo no solo evita sanciones legales, sino que también fortalece la confianza de clientes y socios.

La implementación de un marco de gestión de seguridad, como ISO 27001, permite estructurar medidas de protección y optimizar procesos, reduciendo costos y riesgos. En definitiva, la ciberseguridad no es solo una exigencia legal, sino una necesidad estratégica para la continuidad y reputación de cualquier entidad que maneje información en entornos digitales.