



# ***ACTIVIDAD***

## ***Nº1***

***14/09/2021***

***SGE***

***DAVID SORIANO***

***ENGUIDANOS***

# Contenido

<b>Instal·lant LAMP + Wordpress</b>	<b>3</b>
Preparem el Contenidor	3
WORDPRESS	4
<b>Accedint a la interfície gràfica amb NoVNC</b>	<b>9</b>
Preparant Contenidor	9
<b>Creant imatge Ubuntu amb nano</b>	<b>10</b>
Preparant el Dockerfile i Creant la Imatge	10
<b>Creant imatge amb APP en Node</b>	<b>10</b>
Preparant el Dockerfile i Creant la Imatge	10
<b>Apache2 amb PHP usant Alpine</b>	<b>11</b>
Preparant el Dockerfile i Creant la Imatge	11
<b>Wordpress + MySQL</b>	<b>11</b>
Creant la Xarxa	11
Creant Contenidor MySQL	11
Creant contenidor amb WordPress	11
Instal·lant WordPress	12
Migrant Contenidor MySQL de 5.6 a 5.7	12
<b>WordPress amb Docker Compose</b>	<b>13</b>
Fitxer “Docker-Compose.YML” del cas pràctic	13
<b>Desenvolupant amb Visual Studio Code en un Contenidor</b>	<b>14</b>
Visual Studio Code I Plugins Per Docker	14
<b>Gestionant Docker amb Portainer CE</b>	<b>15</b>
INSTAL·LANT PORTAINER CE	15
<b>Comic Kubernetes</b>	<b>15</b>
Resum	15

## *Instal·lant LAMP + Wordpress*

### *Preparem el Contenidor*

Creem el contenidor LAMP en el nostre Sistema

```
sudo docker run -it -p 8080:80 --name LAMP ubuntu /bin/bash
```

```
soriano@soriano-VirtualBox:~$ sudo docker run -it -p 8080:80 --name LAMP ubuntu /bin/bash
[sudo] contraseña para soriano:
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
35807b77a593: Pull complete
Digest: sha256:9d6a8699fb5c9c39cf08a0871bd6219f0400981c570894cd8cbea30d3424a31f
Status: Downloaded newer image for ubuntu:latest
root@c12f349d4130:/#
```

Despres, tindrem que actualitzar-lo amb:

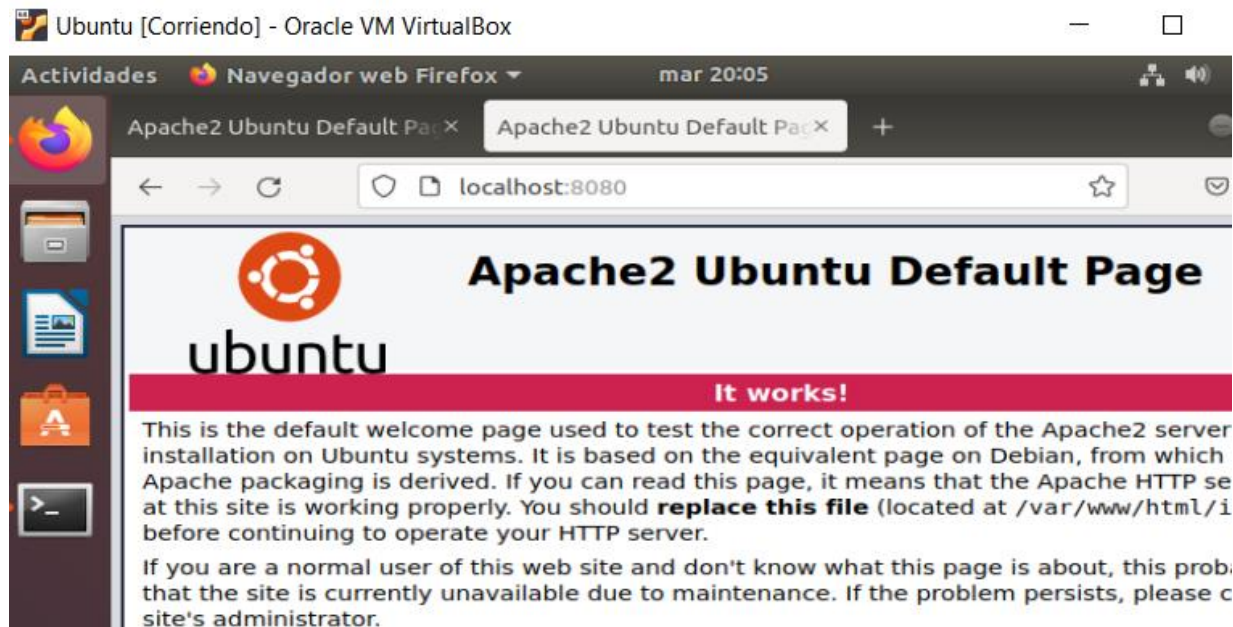
```
apt update
```

En acabant instal·larem els paquets necessaris i tindrem que confirmar la instal·lació, de la mateixa manera, també demanarà la nostra zona geogràfica, en el nostre cas seleccionem Europa, i per últim la zona horària, que seleccionarem la de Madrid.

Després de finalitzar l'instal·lació, llançarem el servei Apache:

```
service apache2 start
```

Seguidament, després de haver finalitzat l'inici d'Apache, podem comprovar que funciona accedint al nostre buscador i dirigint-se a la pàgina web <http://localhost:8080>



## WORDPRESS

Ara ja podem començar a instal·lar Wordpress, el primer pas que tenim que fer és instal·lar un editor de text, en este cas "nano".

L'instal·lació finalitzarà en menys d'un segon.

Ara haurem de crear un arxíu .conf en la ruta “/etc/apache2/sites-available/wordpress.conf”, i editarem el contingut.

```
root@c12f349d4130: /
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 4.8 /etc/apache2/sites-available/wordpress.conf
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
Options FollowSymLinks
AllowOverride Limit Options FileInfo
DirectoryIndex index.php
Order allow,deny
Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
Options FollowSymLinks
Order allow,deny
Allow from all
</Directory>
```

I guardem el arxíu.

Després haurem de carregar el arxíu, també tindrem que habilitar el URL i per últim recarregar l'Apache amb els següents comandaments.

```
a2ensite wordpress
```

```
ordrer a2enmod rewrite
```

```
service apache2 reload
```

Després accedim a <http://localhost:8080/blog>

I el resultat serà un error ja que no hi tenim configurat un arxíu

Ara escomençaré a preparar MySQL Server amb:

```
service mysql start
```

Una vegada fet, procedim a crear una contrasenya amb:

```
mysql_secure_installation
```

```
root@c12f349d4130:/# mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: █
```

```
All done!
```

Accedim a la base de dades:

```
mysql -u root -p
```

Una vegada accedit creem la base de dade i li anyadim un nou usuari i despres li donem privilegis, finalment apliquem el privilegis.

Fet aço podem eixir de mysql, i ara toca editar la configuració de wordpress, aixina:

```
GNU nano 4.8 /etc/wordpress/config-localhost.php
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'MiPass-2021');
define('DB_HOST', 'localhost');
define('DB_COLLATE', 'utf8_general_ci');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>█
```

Comprobem que tot ha funcionat bé.

# 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. Don't worry, you can always change these settings later.

**Site Title**

**Username**

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

**Password**

 Hide

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](#)



SORIANO SITE

Ara anem a fer que cunat iniciem l'ordinador s'inici apache i mysql, anyadint al fitcher .bashrc els següents commands:

```
service apache2 start  
service mysql start
```

I te que quedar aixina:

```
GNU nano 4.8                                     .bashrc  
unset color_prompt force_color_prompt  
  
# If this is an xterm set the title to user@host:dir  
case "$TERM" in  
xterm*|rxvt*)  
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"  
    ;;  
*)  
    ;;  
esac  
  
# enable color support of ls and also add handy aliases  
if [ -x /usr/bin/dircolors ]; then  
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || ev  
    alias ls='ls --color=auto'  
    #alias dir='dir --color=auto'  
    #alias vdir='vdir --color=auto'  
  
    alias grep='grep --color=auto'  
    alias fgrep='fgrep --color=auto'  
    alias egrep='egrep --color=auto'  
fi  
  
# some more ls aliases  
alias ll='ls -alF'  
alias la='ls -A'  
alias l='ls -CF'  
  
# Alias definitions.  
# You may want to put all your additions into a separate file like  
# ~/.bash_aliases, instead of adding them here directly.  
# See /usr/share/doc/bash-doc/examples in the bash-doc package.  
  
if [ -f ~/.bash_aliases ]; then  
    . ~/.bash_aliases  
fi  
  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then  
#    . /etc/bash_completion  
#fi  
  
service apache2 start  
service mysql start
```

Per finalitzar comprovarem en:

```
docker stop LAMP
```

```
docker start LAMP
```

Accedim a <http://localhost:8080/blog> i te que estar aixina:



---

# Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

✎ soriano 🕒 September 16, 2021  
📁 Uncategorized 💬 1 Comment ✎ Edit

## *Accedint a la interfície gràfica amb NoVNC*

### *Preparant Contenidor*

Primerr de tot anem a crear un contenidor nou:

```
docker run --rm -it -p 8080:8080  
theasp/novnc
```

Despres accdim a  
<http://localhost:8080/vnc.html>:

I te que quedar algo aixina.



## Creant imatge Ubuntu amb nano

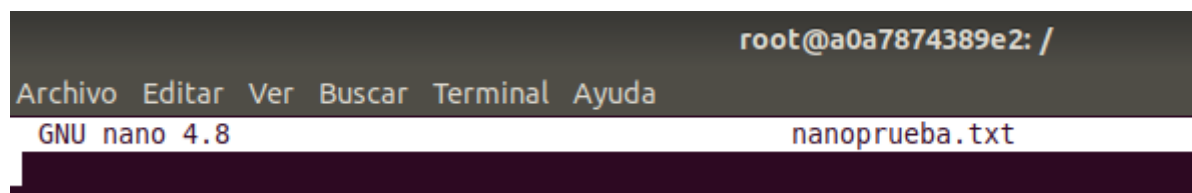
### Preparant el Dockerfile i Creant la Imatge

Creem una imatge de docker amb:  
docker build -t ubuntunano ./

```
Successfully built b2ed69458adf
Successfully tagged ubuntunano:latest
soriano@soriano-VirtualBox:~/Escritorio/ubuntunano$
```

Comprobarem que funciona:  
docker run -it ubuntunano

```
soriano@soriano-VirtualBox:~/Escritorio/ubuntunano$ sudo docker run -it ubuntunano
root@a0a7874389e2:/#
```



## Creant imatge amb APP en Node

### Preparant el Dockerfile i Creant la Imatge

Utilitzant el archiu proporcionat en el tema i amb el comandament:  
docker build -t sampledocker ./

Obtenim el següent resultat:

```
Successfully built a3db938da8fa
Successfully tagged sampledocker:latest
soriano@soriano-VirtualBox:~/Escritorio/sampledocker/app$
```

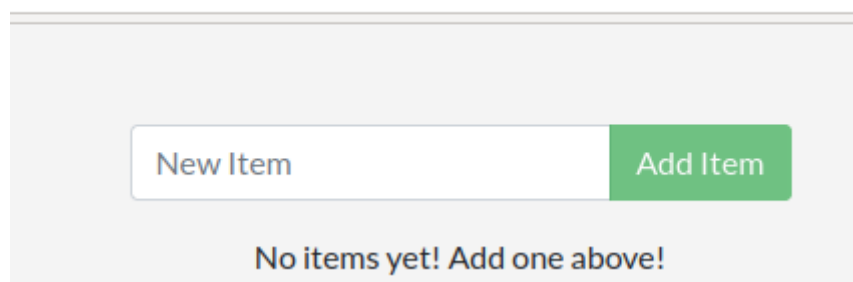
Ara per comprovar que tot ha funcionat bé, executem:  
docker run -dp 3000:3000 sampledocker

```
soriano@soriano-VirtualBox:~/Escritorio/sampledocker/app$ sudo docker run -dp 3000:3000 sampledocker
84b80e2b8d4f00f56de6238355b5f9ceb1a404ea960874254155a7ec37e494be
```

I ara l'aplicació web si tot ha funcionat bé te que donar aquest resultat:

<http://localhost:3000>

localhost:3000



# Apache2 amb PHP usant Alpine

## Preparant el Dockerfile i Creant la Imatge

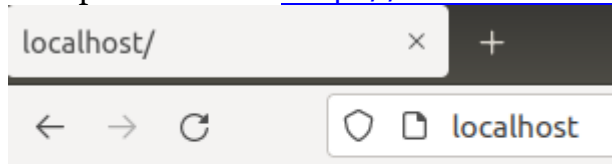
Utilitzem el següent comandament per a inicial el Dockerfile:

```
docker build -t alpineapache ./
Successfully built 949aa0eca7a9
Successfully tagged alpineapache:latest
soriano@soriano-VirtualBox:~/Escritorio/alpineapache$
```

Ara inicialitzare l'aplicació al port 80 de localhost:

```
docker run -dp 80:80 alpineapache
soriano@soriano-VirtualBox:~/Escritorio/alpineapache$ sudo docker run -dp 80:80 alpineapache
f6ec6e211219b0090b0297c7eecd2c64978ac909fcfd03794771f0477d43845f
```

I després entrant a <http://localhost:80>



# It works!

## Wordpress + MySQL

### Creant la Xarxa

Per a escomençar a crear la xarxa executarem:

```
docker network create redwp
soriano@soriano-VirtualBox:~$ sudo docker network create redwp
a47834530b37d677f0f53bcfe4fe4ab357bd05562d4844f9b0557bc1c1a92d59
```

### Creant Contenidor MySQL

Ara crearem el contenidor per al MySQL amb:

```
sudo docker run --name nuestromysql --network redwp -v
/home/soriano/mysqldata:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=cefireroot -
e MYSQL_USER=cefireuser -e MYSQL_PASSWORD=cefirepass -e
MYSQL_DATABASE=cefiredb -d mysql:5.6
```

### Creant contenidor amb WordPress

Després crearem el contenidor amb WordPress amb:

```
Docker run --name nuestrowp --network redwp -p 8080:80 -d worpress
```


## Instal·lant WordPress

Accedim a <http://localhost:8080>



¡Muy bien! Ya has terminado esta parte de la instalación. Ahora WordPress puede comunicarse con tu base de datos. Si estás listo, es el momento de...

Ejecutar la instalación

  SORIANO SITE

## Migrant Contenedor MySQL de 5.6 a 5.7

Primer de tot pararem el contenidor:

```
docker stop nuestromysql
```

```
docker rm nuestromysql
```

```
soriano@soriano-VirtualBox:~$ sudo docker stop nuestromysql
nuestromysql
soriano@soriano-VirtualBox:~$ sudo docker rm nuestromysql
nuestromysql
```

Ara crearem el nou contenidor, l'informació de l'antiga versió encara existir, pero la habem tingut que borrar, per a poder instalar la nova, amb:

```
docker run --name nuestromysql --network redwp -v
/home/soriano/mysqldata:/var/lib/mysql -d mysql:5.7
```

## WordPress amb Docker Compose

### Fitxer “Docker-Compose.YML” del cas pràctic

Obrim la ubicació de la carpeta “CasoPractico1-Wordpress” i escribim en la terminal:

```
docker-compose up -d
```

```
soriano@soriano-VirtualBox:~/Escritorio/CasoPractico1-Wordpress$ sudo docker-compose up -d
[sudo] contraseña para soriano:
Creating network "casopractico1-wordpress_default" with the default driver
Creating volume "casopractico1-wordpress_db_data" with default driver
Creating casopractico1-wordpress_db_1 ... done
Creating casopractico1-wordpress_wordpress_1 ... done
soriano@soriano-VirtualBox:~/Escritorio/CasoPractico1-Wordpress$
```

Accedim a <http://localhost:8000>



Ara pararem el sistema amb:

```
docker-compose down
```

```
soriano@soriano-VirtualBox:~/Escritorio/CasoPractico1-Wordpress$ sudo docker-compose down
Stopping casopractico1-wordpress_wordpress_1 ... done
Stopping casopractico1-wordpress_db_1 ... done
Removing casopractico1-wordpress_wordpress_1 ... done
Removing casopractico1-wordpress_db_1 ... done
Removing network casopractico1-wordpress_default
```

Per ultim relancem el sistema:

```
docker-compose up -d
```

# Desenvolupant amb Visual Studio Code en un Contenedor

## Visual Studio Code I Plugins Per Docker

Instalem Visual Studio Code a Ubuntu en <https://code.visualstudio.com/>.

Ara crearem un contenedor que al·macenarà Apache+PHP amb:

```
sudo docker run -d --name servidordesarrollo -p 8080:80 php:7.2-apache
```

```
Status: Downloaded newer image for php:7.2-apache
8624cf6416a85eca7e2c9587d30fdd71af6ee76114b01ed439c2026aa44ce387
soriano@soriano-VirtualBox:~$
```

Després anem a <http://localhost:8080> i comprovem que tot ha funcionat bé.


## Forbidden

You don't have permission to access this resource.


*Apache/2.4.38 (Debian) Server at localhost Port 8080*

Ens ixirà aquest error, es normal.

Finalment:


**PHP Version 7.2.34**

System	Linux 21bec17a59 5.4.0-84-generic #94~18.04.1-Ubuntu SMP Thu Aug 26 23:17:46 UTC 2020 x86_64
Software de Ubuntu	
Build Date	Dec 11 2020 10:50:00
Configure Command	./configure '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file=/usr/local/etc/php/conf.d' '--enable-option-checking=fatal' '--with-mhash' '--with-pic' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-password-argon2' '--with-sodium=sodium' '--with-sqlite=/usr' '--with-sqlite3=/usr' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' '--disable-cgi' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php
Loaded Configuration File	(none)
Scan this dir for additional .ini files	/usr/local/etc/php/conf.d
Additional .ini files parsed	/usr/local/etc/php/conf.d/docker-php-ext-sodium.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.mbstring, chunkencode, chunkdecode, chunkencode, chunkdecode

This program makes use of the Zend Scripting Language Engine: 

# Gestionant Docker amb Portainer CE

## INSTAL·LANT PORTAINER CE

Container list 

Containers

admin



















[my account](#) [log out](#)

Containers

Columns Settings

Start Stop Kill Restart Pause Resume Remove Add container

Search...  
Audio Code

<input type="checkbox"/>	Name	State  Filter 	Quick actions	Stack	Image
<input type="checkbox"/>	<u>sorianoserver</u>	running	   	-	nginx:latest
<input type="checkbox"/>	portainer	running	   	-	portainer/portainer
<input type="checkbox"/>	casopractico1-wordpress_wordpress_1	running	   	casopractico1-wordpress	wordpress:latest
<input type="checkbox"/>	casopractico1-wordpress_db_1	running	   	casopractico1-wordpress	mysql:5.7

## Comic Kubernetes

### Resum

Kubernetes es un comic que parla d'un informàtic que porta una empresa que ha fet un videojoc popular. I han sorgit molts problemes que ell no pot solucionar. Despres apareix Atenea, que l'explica com Kubernetes funciona es un programa que funciona amb contenidos el videojoc s'adapta es portable i escalable. A més, l'explica quins son els punts forts de l'aplicació. Al final, Atenea l'explica a l'informàtic com començar a utilitzar-lo.