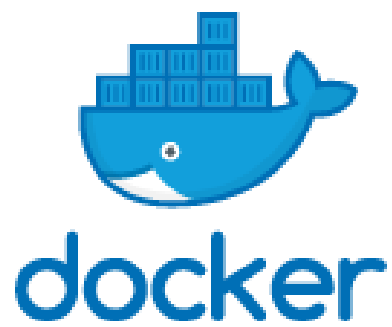




UTILIZACIÓN DE CONTENEDORES



Realizado por:

Daniel Rodríguez Fernández

Madrid 19 de Octubre de 2022

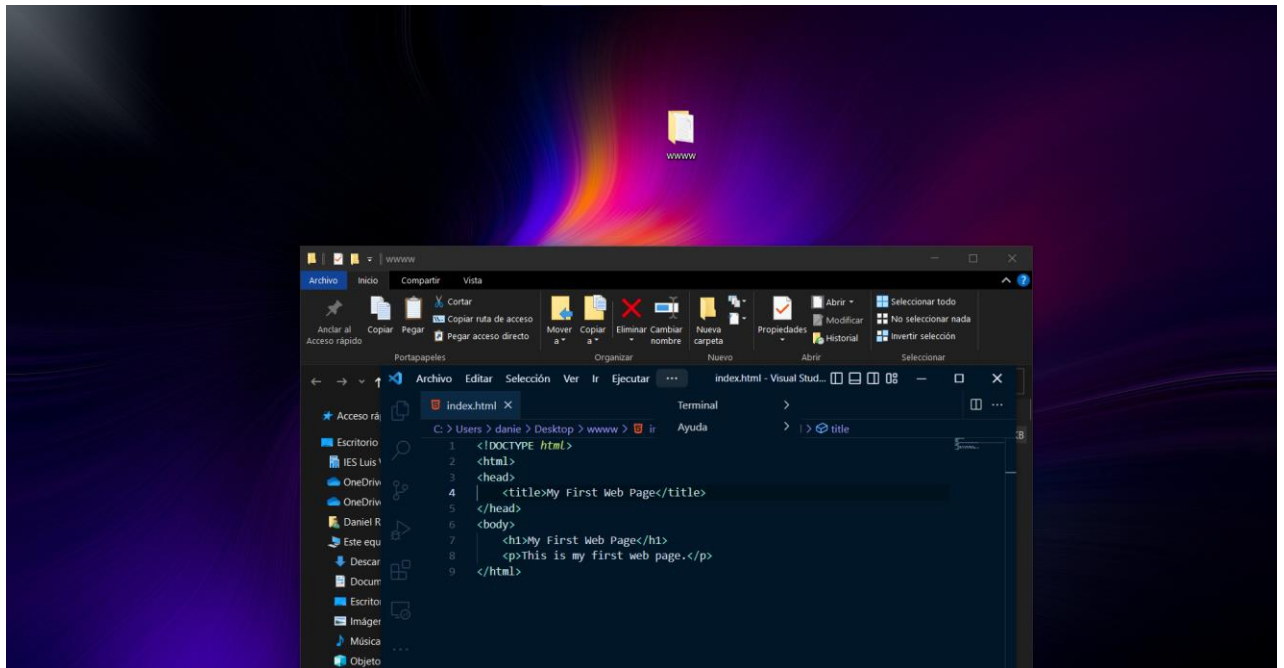
Contenido

1.1. Utilización de contenedores	3
1.1.1. Crea una página web sencilla en HTML, llámala index.html y guárdala en una carpeta llamada "www" en tu escritorio.	3
1.2. Descarga una imagen de Docker Hub de un servidor Apache o httpd en Docker Hub (pull).....	3
1.3. Ejecútalo y comprueba que funciona (start).....	4
1.4. Páralo y, si no vas a utilizar esta imagen, elimina el contenedor y la imagen.	4
1.5. Utiliza Docker para compilar y ejecutar una aplicación java que tengas en una carpeta de tu máquina (bind mount).	5
2. Creación de imágenes.....	5
2.1. Crea una imagen mediante un Dockerfile a partir de una imagen de apache que contenga tu web.	5
2.2. Expón el puerto 80.	5
2.3. Crea la imagen y llámala mi-web.....	5
2.4. Crea y ejecuta un contenedor que ejecute tu imagen en el puerto 80.	6
2.5. Crea una cuenta de docker y prueba a subir tu imagen a un repositorio público de DockerHub, después elimínala.	6
3. Ejecución de una aplicación basada en varios contenedores	7
3.1. Ejecuta una aplicación wordpress ejecutando manualmente los contenedores necesarios y uniéndolos en una red virtual.	7
3.2. Ejecuta una aplicación wordpress utilizando DockerCompose.	8
4. Instalación de Odoo mediante Docker	9
4.1. Sigue las instrucciones del repositorio oficial de Odoo y los conceptos aprendidos para:	9
4.1.1. Lanzar una instancia de Odoo utilizando la opción --link.	9
4.1.2. Lanzar una instancia de Odoo mediante una red virtual.	9
4.1.3. Lanzar una instancia de Odoo que tenga persistencia de datos de configuración y custom addons (bind mount), y de bbdd (volume).	10
4.1.4. Lanza una instancia de Odoo utilizando Docker Compose.....	10
5. Referencias.	11

1.1. Utilización de contenedores

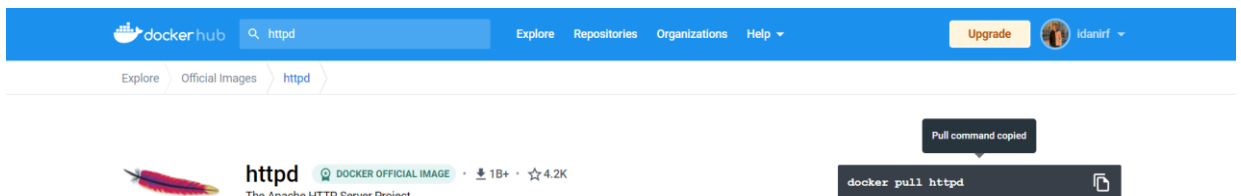
1.1. Crea una página web sencilla en HTML, llámala index.html y guárdala en una carpeta llamada "www" en tu escritorio.

Creamos la carpeta en el escritorio y creamos un archivo html básico.



1.2. Descarga una imagen de Docker Hub de un servidor Apache o httpd en Docker Hub (pull).

Imagen de Docker Hub, obtenemos el comando para descargar la imagen.



Con el comando docker pull Httpd descargamos la imagen. Después creamos la imagen con sus puertos.

```
PS C:\Users\danie> docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
e9995326b091: Pull complete
ee55ccd48c8f: Pull complete
bc66ebee7efe: Pull complete
5d0f831d3c0b: Pull complete
e559e5380898: Pull complete
Digest: sha256:5fa96551b61359de5dfb7fd8c9e97e4153232eb520a8e883e2f47fc80dbfc33e
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
PS C:\Users\danie> docker create --name apache 8080:80 httpd
Unable to find image '8080:80' locally
Error response from daemon: pull access denied for 8080, repository does not exist or may require 'docker login': deni
: requested access to the resource is denied
PS C:\Users\danie> docker create --name apache -p 8080:80 httpd
075898bc63a16e95732deb3562350efd3ed294c1bb2b08e124546afdd628d470
PS C:\Users\danie>
```

1.3. Ejecútalo y comprueba que funciona (start).

Sintaxis → -p (puerto que vamos a abrir) -v(ruta donde vamos a guardar el volumen creado)

```
docker run -dit --name idanirfWeb1 -p 8080:80 -v C:\Users\danie\Desktop\www\:/usr/local/apache2/htdocs/ httpd
```

```
PS C:\Users\danie> docker run -dit --name idanirfWeb1 -p 8080:80 -v C:\Users\danie\Desktop\www\:/usr/local/apache2/htdocs/ httpd
963b34dcf6e116077ad1c16678fbc5294e9bad6d87057b5477cbb5a948d7681c
```



1.4. Páralo y, si no vas a utilizar esta imagen, elimina el contenedor y la imagen.

Vemos los servicios activos y detenemos la imagen lanzada.

```
S C:\Users\danie> docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS               NAMES
63b34dcf6e1   httpd     "httpd-foreground"      5 minutes ago Up 5 minutes    0.0.0.0:8080->80/tcp idanirfWeb1

S C:\Users\danie> docker stop idanirfWeb1
idanirfWeb1

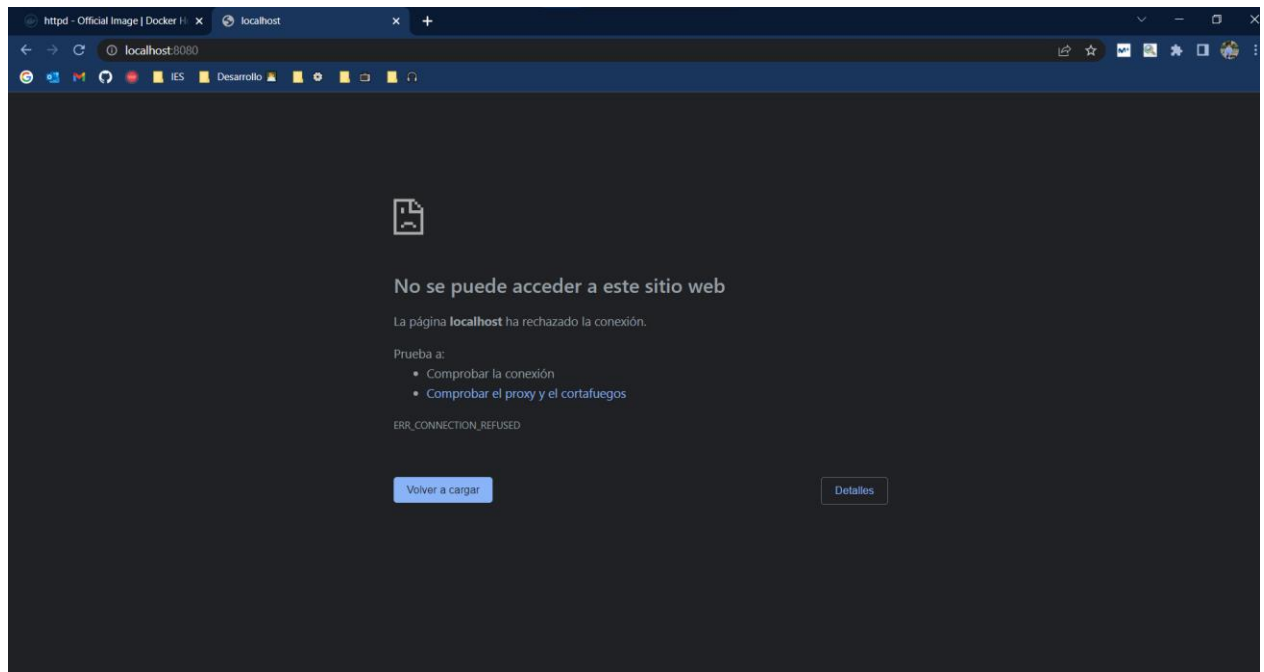
S C:\Users\danie> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS               NAMES
63b34dcf6e1   httpd     "httpd-foreground"      9 minutes ago Exited (0) 3 minutes ago                idanirfWeb1
75898bc63a1   httpd     "httpd-foreground"      15 minutes ago Created                                 apache

S C:\Users\danie> docker rm apache
apache

S C:\Users\danie> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS               NAMES
63b34dcf6e1   httpd     "httpd-foreground"      9 minutes ago Exited (0) 3 minutes ago                idanirfWeb1

S C:\Users\danie>
```

Podemos comprobar que al estar detenida la imagen no podemos acceder.



1.5. Utiliza Docker para compilar y ejecutar una aplicación java que tengas en una carpeta de tu máquina (bind mount).

Primero creamos un archivo java que contenga al menos un hola mundo. Una vez que la tengamos creado, ejecutamos el siguiente comando para ejecutar el contenedor y una vez su ejecución haya finalizado se eliminará, le pasamos la ruta, también el jdk y con javac el archivo .java y a continuación ejecutamos el siguiente comando donde le indicamos la ruta y con java el nombre del archivo Main y el archivo.java

```
at sun.launcher.LauncherHelper.checkAndLoadMain(Unknown Source)
PS C:\Users\danie\Desktop> docker run --rm -v C:\Users\danie\Desktop\:/tmp/java -w /tmp/java openjdk javac HolaMundo.java
Unable to find image 'openjdk:latest' locally
latest: Pulling from library/openjdk
50cbc88660a5: Pull complete
3f15da7b20d8: Pull complete
812b9f471c4d: Pull complete
Digest: sha256:448d8240b4e40a51e9710cf032d512457182233c4646681a58262efb15fd2d
Status: Downloaded newer image for openjdk:latest
PS C:\Users\danie\Desktop> docker run --rm -v C:\Users\danie\Desktop\:/tmp/java -w /tmp/java openjdk java HolaMundo Hola Mundo
PS C:\Users\danie\Desktop> docker run --rm -v C:\Users\danie\Desktop\:/tmp/java -w /tmp/java openjdk sh -c "javac HolaMundo.java && java HolaMundo"
Hola Mundo
PS C:\Users\danie\Desktop>
```

2. Creación de imágenes

2.1. Crea una imagen mediante un Dockerfile a partir de una imagen de apache que contenga tu web.

Ejecutamos el comando docker run le indicamos el puerto, la ruta y el servicio Httpd.

```
PS C:\Users\danie> docker run -dit -p 8080:80 -v C:\Users\danie\Desktop\www\:/usr/local/apache2/htdocs/ httpd
f672c45c645e0bdf8599ae9b750e9cb79e066995729c9f1a3a673ce1a0a93d0e
PS C:\Users\danie> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
f672c45c645e   httpd     "httpd-foreground"      14 minutes ago Up 14 minutes  0.0.0.0:8080->80/tcp    ecstatic_engelbart
PS C:\Users\danie> docker commit f672c45c645e ecstatic_engelbart
sha256:327fbb4fa86352048a7fe9953857aae771b8409d5f1c42d5df9582b8778fa23f
PS C:\Users\danie>
```

2.2. Expón el puerto 80.

Vemos que cuando hemos creado nuestro contenedor ya le hemos asignado el puerto 80.

```
PS C:\Users\danie> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
f672c45c645e   httpd     "httpd-foreground"      26 minutes ago Up 26 minutes  0.0.0.0:8080->80/tcp    ecstatic_engelbart
PS C:\Users\danie>
```

Vemos el contenido que hay en el puerto 80.



2.3. Crea la imagen y llámala mi-web.

Creamos una carpeta llamada docker en la unidad C en la ruta raíz.

```
Windows PowerShell
PS C:\Users\danie> cd ..
PS C:\Users> cd ..
PS C:\> mkdir docker

Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          31/10/2022   16:24         docker
```

A continuación creamos un Path hacia la ruta docker file indicando que es de tipo file.

```
PS C:\docker> new-item -path c:\docker\dockerfile -itemType file

Directorio: C:\docker

Mode                LastWriteTime         Length Name
----                -
-a-----          31/10/2022    16:25             0 dockerfile
```

A continuación creamos una carpeta que contiene los archivos y creamos un fichero html dentro de esta carpeta.

```
PS C:\docker> cd C:\docker\archivos\
PS C:\docker\archivos> echo "Hola Mundo" > index1.html
PS C:\docker\archivos> docker images

REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
miweb                latest             c215dda13b74       2 hours ago        145MB
ecstatic_engelbart  latest             327fbb4fa863       3 hours ago        145MB
httpd                latest             fe8735c23ec5       6 days ago         145MB
openjdk              latest             d3df331637f8       9 days ago         464MB
PS C:\docker\archivos>
```

A continuación modificamos el archivo Dockerfile.

```
*dockerfile: Bloc de notas
Archivo Edición Formato Ver Ayuda
FROM httpd
RUN MKDIR Documentos
COPY /archivos /Documentos
EXPOSE 8080
```

2.4. Crea y ejecuta un contenedor que ejecute tu imagen en el puerto 80.

Ejecutamos el comando build con docker sobre la imagen.

```
PS C:\docker\archivos> docker build -t miweb1 .
[+] Building 0.1s (2/2) FINISHED
=> CANCELED [internal] load .dockerignore
=> => transferring context:
failed to solve with frontend dockerfile.v0: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount3867331036/Dockerfile: no such file or directory
PS C:\docker\archivos>
```

Podemos ver el resultado.



2.5. Crea una cuenta de docker y prueba a subir tu imagen a un repositorio público de DockerHub, después elimínala.

Iniciamos sesión en docker hub con el comando docker login -u y nombre de usuario. Después hacemos un push y subimos la imagen.

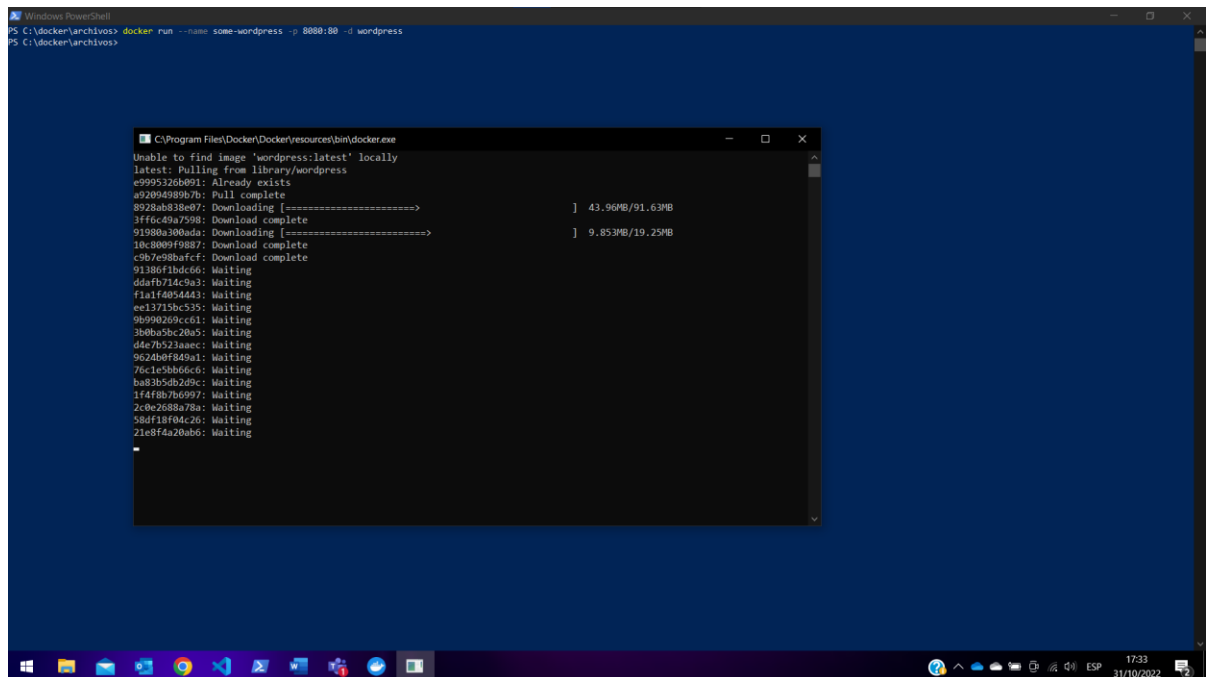
```
PS C:\docker\archivos> docker login -u idanirf
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
PS C:\docker\archivos> docker push miweb
Using default tag: latest
The push refers to repository [docker.io/library/miweb]
90976605ab66: Preparing
1617cfa5f5dd: Preparing
36de10a434ef: Preparing
6437b1170b0b: Preparing
13fb8799144c: Preparing
a12586ed027f: Waiting
denied: requested access to the resource is denied
PS C:\docker\archivos>
```

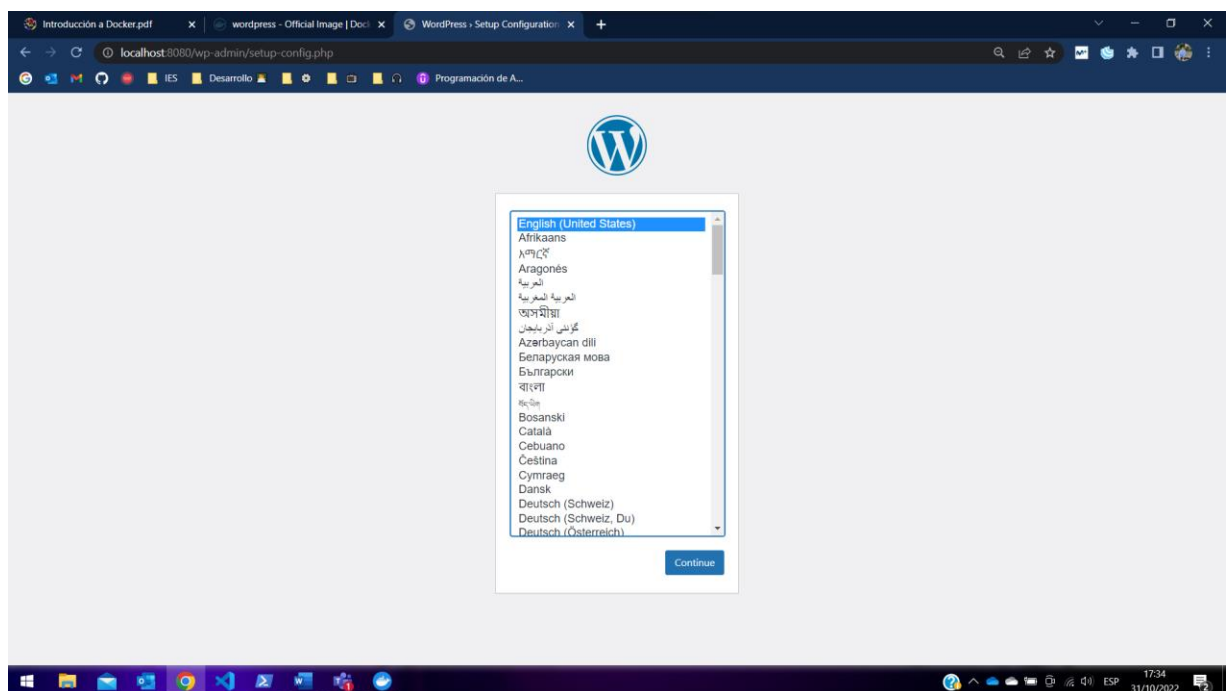
3. Ejecución de una aplicación basada en varios contenedores

3.1. Ejecuta una aplicación wordpress ejecutando manualmente los contenedores necesarios y uniéndolos en una red virtual.

Ejecutamos un nuevo contenedor la imagen de wordpress como no tenía la imagen descargada se está descargando y después se ejecutará el contenedor.



Podemos ver como resultado que ha funcionado y podemos acceder a Wordpress.



3.2. Ejecuta una aplicación wordpress utilizando DockerCompose.

Obtenemos el archivo de docker hub lo dejamos el escritorio y lo llamamos docker-compose.yaml

Obtenemos el archivo de docker hub lo dejamos el escritorio y lo llamamos docker-compose.yaml

```

1 version: '3.1'
2
3 services:
4
5   wordpress:
6     image: wordpress
7     restart: always
8     ports:
9       - 8080:80
10    environment:
11      WORDPRESS_DB_HOST: db
12      WORDPRESS_DB_USER: exampleuser
13      WORDPRESS_DB_PASSWORD: examplepass
14      WORDPRESS_DB_NAME: exampledb
15    volumes:
16      - wordpress:/var/www/html
17
18   db:
19     image: mysql:5.7
20     restart: always
21     environment:
22       MYSQL_DATABASE: exampledb
23       MYSQL_USER: exampleuser
24       MYSQL_PASSWORD: examplepass
25       MYSQL_RANDOM_ROOT_PASSWORD: '1'
26     volumes:
27       - db:/var/lib/mysql
28
29 volumes:
30   wordpress:
31   db:

```

Subimos el servicio docker-compose up.

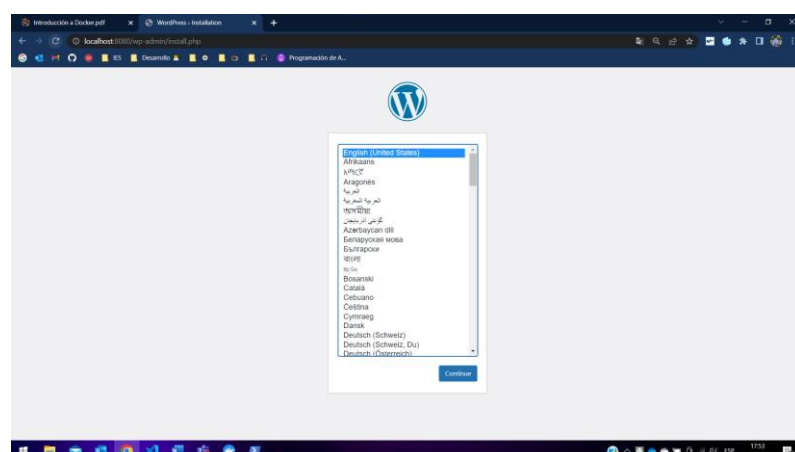
```

C:\Windows PowerShell
PS C:\Users\danie\Desktop> docker-compose up

    Network desktop_default      Created                                0.8s
    Container desktop-db-1       Created                                0.1s
    Container desktop-wordpress-1 Created                                0.1s
Attaching to desktop-db-1, desktop-wordpress-1
desktop-db-1                    2022-10-31 16:48:28.0000 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.40-1.el7 started.
desktop-wordpress-1             2022-10-31 16:48:28.0000 [Note] [Entrypoint]: Copying root...
desktop-wordpress-1             2022-10-31 16:48:28.0000 [Note] [Entrypoint]: Switching to dedicated user 'mysql'.
desktop-wordpress-1             2022-10-31 16:48:28.0000 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.40-1.el7 started.
desktop-wordpress-1             Completed wordpress has been successfully copied to /var/www/html
    1 no wp-config.php found in /var/www/html, but WORDPRESS_... variables supplied; copying 'wp-config-docker.php' (WORDPRESS_DB_HOST WORDPRESS_DB_NAME WORDPRESS_DB_PASSWORD WORDPRESS_DB_USER)
    '1/var/lib/mysql/mysql.sock' -> '/var/run/mysql/mysql.sock'
desktop-wordpress-1             AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
desktop-wordpress-1             [Mon Oct 31 16:48:28.388595 2022] [error] [pid 1] AH00016: Apache/2.4.54 (Debian) PHP/7.4.32 configured -- resuming normal operations
desktop-wordpress-1             [Mon Oct 31 16:48:28.389825 2022] [corenotice] [pid 1] AH00042: Command line: 'apache2 -D FOREGROUND'
desktop-wordpress-1             [Mon Oct 31 16:48:28.400000 2022] [warn] [pid 1] Using TIMESTAMPTO with implicit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp server option (see documentation for more details).
desktop-db-1                    2022-10-31 16:48:28.427602Z [Note] mysqld (mysqld 5.7.40) starting as process 1 ...
desktop-wordpress-1             2022-10-31 16:48:28.429342Z [Note] InnoDB: PUNCH HOLE support available
desktop-wordpress-1             2022-10-31 16:48:28.429522Z [Note] InnoDB: Mutexes and rw_locks use GCC atomic builtins
desktop-wordpress-1             2022-10-31 16:48:28.429542Z [Note] InnoDB: Uses event mutexes
desktop-wordpress-1             2022-10-31 16:48:28.429562Z [Note] InnoDB: GCC builtin __atomic_thread_fence() is used for memory barrier
desktop-wordpress-1             2022-10-31 16:48:28.429572Z [Note] InnoDB: Compressed tables use zlib 1.2.12
desktop-wordpress-1             2022-10-31 16:48:28.429572Z [Note] InnoDB: Using Linux native AIO
desktop-wordpress-1             2022-10-31 16:48:28.430136Z [Note] InnoDB: Number of pools: 1
desktop-wordpress-1             2022-10-31 16:48:28.430312Z [Note] InnoDB: Using CPU crc32 instructions
desktop-wordpress-1             2022-10-31 16:48:28.431022Z [Note] InnoDB: Initializing buffer pool, total size = 120M, instances = 1, chunk size = 120M
desktop-wordpress-1             2022-10-31 16:48:28.431901Z [Note] InnoDB: Completed initialization of buffer pool
desktop-wordpress-1             2022-10-31 16:48:28.441324Z [Note] InnoDB: If the mysqld execution user is authorized, page cleaner thread priority can be changed. See the man page of setpriority().
desktop-wordpress-1             2022-10-31 16:48:28.451496Z [Note] InnoDB: Highest supported file format is Barracuda.
desktop-wordpress-1             2022-10-31 16:48:28.456032Z [Note] InnoDB: Creating shared tablespace for Temporary tables
desktop-wordpress-1             2022-10-31 16:48:28.457019Z [Note] InnoDB: Setting file './ibtmp1' size to 12 MB. Physically writing the file full; Please wait ...
desktop-wordpress-1             2022-10-31 16:48:28.459172Z [Note] InnoDB: File './ibtmp1' size is now 12 MB.
desktop-wordpress-1             2022-10-31 16:48:28.461917Z [Note] InnoDB: 90 redo rollback segment(s) found. 90 redo rollback segment(s) found. 90 redo rollback segment(s) are active.
desktop-wordpress-1             2022-10-31 16:48:28.462800Z [Note] InnoDB: 32 non-redo rollback segment(s) are active.
desktop-wordpress-1             2022-10-31 16:48:28.463936Z [Note] InnoDB: 5.7.40 started; log sequence number 12140270
desktop-wordpress-1             2022-10-31 16:48:28.463936Z [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
desktop-wordpress-1             2022-10-31 16:48:28.463936Z [Note] Plugin 'FEDERATED' is disabled.
desktop-wordpress-1             2022-10-31 16:48:28.465170Z [Note] InnoDB: Buffer pool(s) load completed at 221031 16:48:28
desktop-wordpress-1             2022-10-31 16:48:28.469432Z [Note] Found ca.pem, server-cert.pem and server-key.pem in data directory. Trying to enable SSL support using them.
desktop-wordpress-1             2022-10-31 16:48:28.469432Z [Note] Skipping generation of SSL certificates as certificate files are in data directory.
desktop-wordpress-1             2022-10-31 16:48:28.469491Z [Warning] A deprecated TLS version TLSv1 is enabled. Please use TLSv1.2 or higher.
desktop-wordpress-1             2022-10-31 16:48:28.469492Z [Warning] A deprecated TLS version TLSv1.1 is enabled. Please use TLSv1.2 or higher.
desktop-wordpress-1             2022-10-31 16:48:28.469904Z [Warning] CA certificate ca.pem is self signed.
desktop-wordpress-1             2022-10-31 16:48:28.469958Z [Note] Skipping generation of RSA key pair as key files are present in data directory.
desktop-wordpress-1             2022-10-31 16:48:28.500575Z [Note] Server hostname (bind-address): '*'; port: 3306
desktop-wordpress-1             2022-10-31 16:48:28.500575Z [Note] IPv6 is available.
desktop-wordpress-1             2022-10-31 16:48:28.500522Z [Note] '!' resolves to '!!';
desktop-wordpress-1             2022-10-31 16:48:28.500521Z [Note] Server socket created on IP: '::'.
desktop-wordpress-1             2022-10-31 16:48:28.500500Z [Warning] Insecure configuration for '--pid-file': Location '/var/run/mysqld' in the path is accessible to all OS users. Consider choosing a different directory.
desktop-wordpress-1             2022-10-31 16:48:28.510747Z [Note] Event Scheduler: Loaded 0 events
desktop-wordpress-1             2022-10-31 16:48:28.511982Z [Note] mysqld: ready for connections.
Version: '5.7.40' socket: '/var/run/mysql/mysql.sock' port: 3306 MySQL Community Server (GPL)

```

Podemos ver como resultado que ha funcionado y podemos acceder a Wordpress.



4. Instalación de Odoo mediante Docker

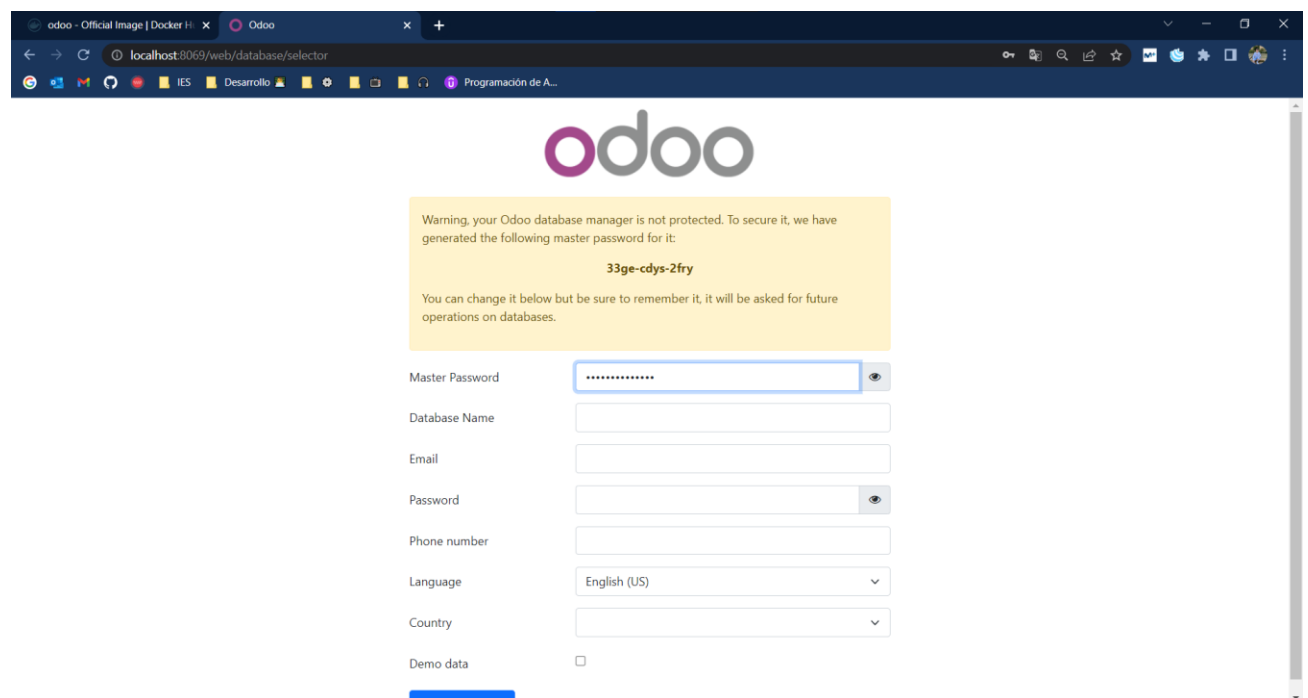
4.1. Sigue las instrucciones del repositorio oficial de Odoo y los conceptos aprendidos para:

4.1.1. Lanzar una instancia de Odoo utilizando la opción --link.

Accedemos a docker hub y obtenemos el comando para ejecutar docker con la opción --link.

```
Windows PowerShell
PS C:\Users\daniel> docker run -d --name postgres --link db:postgres --name db postgres:13
Unable to find image 'postgres:13' locally
13: Pulling from library/postgres
a9995126b091: Already exists
8b383f17886: Pull complete
bb26f7e78134: Pull complete
c8e073b7ae91: Pull complete
9965b1679915: Pull complete
556528f6c3c5: Pull complete
89ac8466672c: Pull complete
4effb95d5849: Pull complete
97f42548f1e1: Pull complete
43e7713b7769: Pull complete
2888936d5b2e: Pull complete
b4b731b0864d: Pull complete
7bd79522d4dc: Pull complete
Digest: sha256:20316c2ba82d687bb91d66e69c2534c74107db3192f22a04de3b6425905
Status: Downloaded newer image for postgres:13
dad4942e0f1829ee574c1324450847c443958623fa75e8a5b431b6a4cce2f2
PS C:\Users\daniel> docker run -p 8069:8069 --name odoo --link db:db -t odoo
2022-11-01 12:03:41,621 INFO odoo: Odoo version 16.0-20221025
2022-11-01 12:03:41,621 INFO odoo: Using configuration file at /etc/odoo/odoo.conf
2022-11-01 12:03:41,621 INFO odoo: addons paths: ['/usr/lib/python3/dist-packages/odoo/addons', '/var/lib/odoo/addons/16.0', '/mnt/extra-addons']
2022-11-01 12:03:41,621 INFO odoo: database: odoo@172.17.0.2:5432
2022-11-01 12:03:41,705 INFO odoo.addons.base.models.ir_actions_report: A package may be missing to print Data Matrix barcodes: pylibdmtx or libdmtx.
2022-11-01 12:03:41,791 INFO odoo.addons.base.models.ir_actions_report: Will use the Wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2022-11-01 12:03:42,088 INFO odoo.service.server: HTTP service (werkzeug) running on f0bf8406f6b6:8069
2022-11-01 12:03:52,420 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:52] "GET / HTTP/1.1" 303 - 1 0.003 0.026
2022-11-01 12:03:52,444 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:52] "GET /web HTTP/1.1" 303 - 2 0.005 0.013
2022-11-01 12:03:52,893 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:52] "GET /web/database/selector HTTP/1.1" 200 - 2 0.005 0.031
2022-11-01 12:03:52,987 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:52] "GET /web/static/lib/bootstrap/dist/css/bootstrap.css HTTP/1.1" 200 - 1 0.003 0.022
2022-11-01 12:03:52,993 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:52] "GET /web/static/src/lib/fontawesome/css/font-awesome.css HTTP/1.1" 200 - 1 0.002 0.027
2022-11-01 12:03:52,994 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:52] "GET /web/static/lib/jquery/jquery.js HTTP/1.1" 200 - 1 0.004 0.023
2022-11-01 12:03:52,995 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:52] "GET /web/static/lib/bootstrap/js/dist/dom/event-handler.js HTTP/1.1" 200 - 1 0.002 0.022
2022-11-01 12:03:52,996 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:52] "GET /web/static/lib/bootstrap/js/dist/dom/data.js HTTP/1.1" 200 - 1 0.003 0.024
2022-11-01 12:03:53,003 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:53] "GET /web/static/lib/bootstrap/js/dist/dom/manipulator.js HTTP/1.1" 200 - 1 0.005 0.027
2022-11-01 12:03:53,029 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:53] "GET /web/static/lib/bootstrap/js/dist/modal.js HTTP/1.1" 200 - 1 0.003 0.014
2022-11-01 12:03:53,030 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:53] "GET /web/static/lib/bootstrap/js/dist/dom/selector-engine.js HTTP/1.1" 200 - 1 0.009 0.019
2022-11-01 12:03:53,032 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:53] "GET /web/static/lib/bootstrap/js/dist/base-component.js HTTP/1.1" 200 - 1 0.002 0.015
2022-11-01 12:03:53,037 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:53] "GET /web/static/src/public/database_manager.js HTTP/1.1" 200 - 1 0.003 0.017
2022-11-01 12:03:53,055 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:53] "GET /web/static/img/logo.png HTTP/1.1" 200 - 1 0.002 0.005
2022-11-01 12:03:53,264 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:53] "GET /web/static/src/lib/fontawesome/fonts/fontawesome-webfont.woff2?v=4.7.0 HTTP/1.1" 200 - 1 0.003 0.018
2022-11-01 12:03:53,491 INFO werkzeug: 172.17.0.1 - - [01/Nov/2022 12:03:53] "GET /web/static/img/favicon.ico HTTP/1.1" 200 - 1 0.002 0.008
```

Podemos ver que su ejecución ha funcionado y podemos hacer la instalación de odoo.

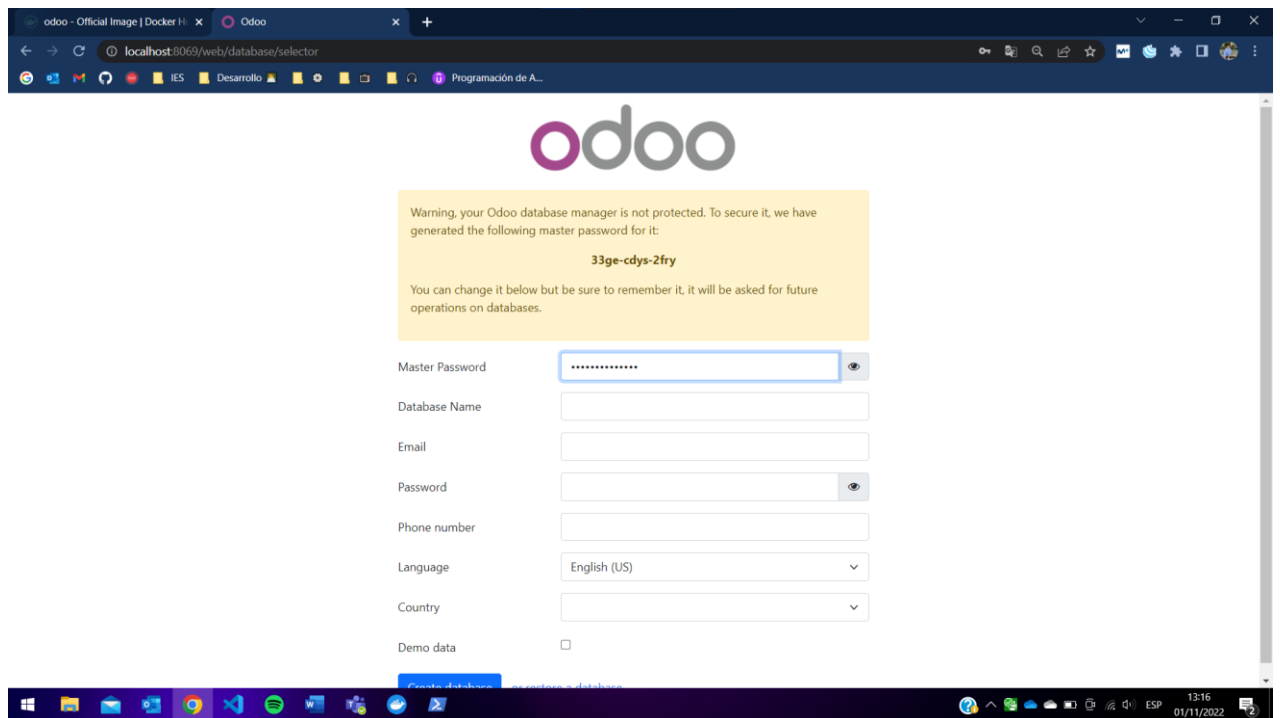


4.1.2. Lanzar una instancia de Odoo mediante una red virtual.

Creamos una red virtual con el comando docker network create y el nombre de la red. Y después lanzamos la imagen de odoo.

```
Windows PowerShell
PS C:\Users\daniel> docker network create redodoo
PS C:\Users\daniel> docker run -p 8069:8069 --name odoo --link db:db -t odoo
2022-11-01 12:15:30,675 INFO odoo: Odoo version 16.0-20221025
2022-11-01 12:15:30,675 INFO odoo: Using configuration file at /etc/odoo/odoo.conf
2022-11-01 12:15:30,675 INFO odoo: addons paths: ['/usr/lib/python3/dist-packages/odoo/addons', '/var/lib/odoo/addons/16.0', '/mnt/extra-addons']
2022-11-01 12:15:30,675 INFO odoo: database: odoo@172.17.0.2:5432
2022-11-01 12:15:30,853 INFO odoo.addons.base.models.ir_actions_report: A package may be missing to print Data Matrix barcodes: pylibdmtx or libdmtx.
2022-11-01 12:15:30,856 INFO odoo.addons.base.models.ir_actions_report: Will use the Wkhtmltopdf binary at /usr/local/bin/wkhtmltopdf
2022-11-01 12:15:31,036 INFO odoo.service.server: HTTP service (werkzeug) running on d84373305f5e:8069
```

Podemos ver el resultado y ya podríamos realizar la instalación de odoo.



4.1.3. Lanzar una instancia de Odoo que tenga persistencia de datos de configuración y custom addons (bind mount), y de bbdd (volume).

Este apartado no lo he conseguido completar, he encontrado los comandos necesarios para completarlo. Pero una vez que los ejecuto me salen errores que no consigo solucionar.

Primeramente ejecutamos nuestro contenedor con una configuración personalizada para ello utilizamos el siguiente comando.

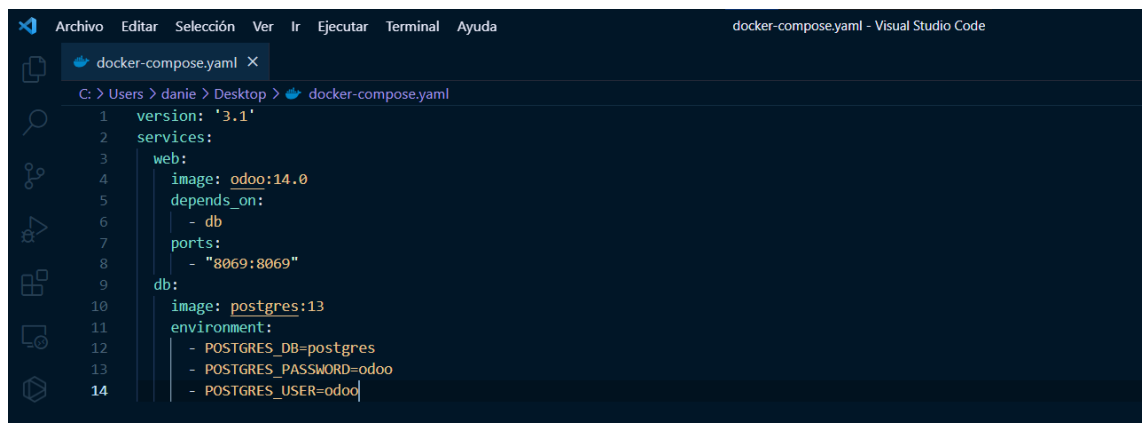
```
PS C:\Users\danie\Desktop> docker run -v /path/to/config:/etc/odoo -p 8069:8069 --name odoo --link db:db -t odoo
docker: Error response from daemon: could not get container for db: No such container: db.
See 'docker run --help'.
PS C:\Users\danie\Desktop>
```

Ahora con el siguiente comando podemos montar la ruta para complementos personalizados.

```
b2 C:\Users\danie\Desktop> docker run -v /path/to/config:/etc/odoo -v /path/to/addons:/usr/lib/python3/dist-packages/odoo/addons -p 8069:8069 --name odoo --link db:db -t odoo
b2 C:\Users\danie\Desktop>
```

4.1.4. Lanza una instancia de Odoo utilizando Docker Compose.

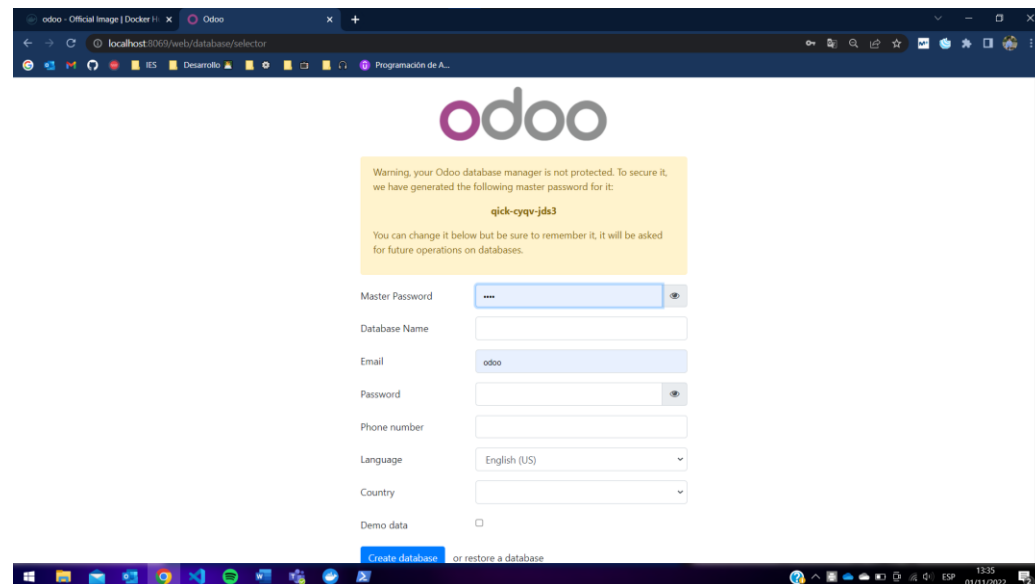
Obtenemos el texto de docker hub lo insertamos en el escritorio en un archivo llamado docker-compose.yaml



Lanzamos el servicio con docker-compose up -d para que se ejecute tanto PostgreSQL y odoo.

```
PS C:\Users\danie\Desktop> docker-compose up -d
[+] Running 3/3
 - Network desktop_default Created
 - Container desktop-db-1 Started
 - Container desktop-web-1 Started
PS C:\Users\danie\Desktop>
```

Podemos ver que podemos acceder a odoo y realizar su instalación.



5. Referencias.

https://docs.docker.com/get-started/02_our_app/

https://www.youtube.com/watch?v=J_ja2soiUZk

<https://atareao.es/tutorial/docker/crear-una-imagen-docker-desde-cero-y-paso-a-paso/>

<https://www.returngis.net/2019/02/publicar-tu-imagen-en-docker-hub/>