|  |  |  |
| --- | --- | --- |
| BurbujonDeLosHorrores | | |
|  |  |  |
|  | | |

Indice

[Configuración de Switch 2](#_Toc65181105)

[CMMS 3](#_Toc65181106)

[Oracle Database 19c 6](#_Toc65181107)

[SQL Developer 7](#_Toc65181108)

[Instalacion Symfony 7](#_Toc65181109)

[Creacion de servicio que lanza webservice 8](#_Toc65181110)

[Hardware 10](#_Toc65181111)

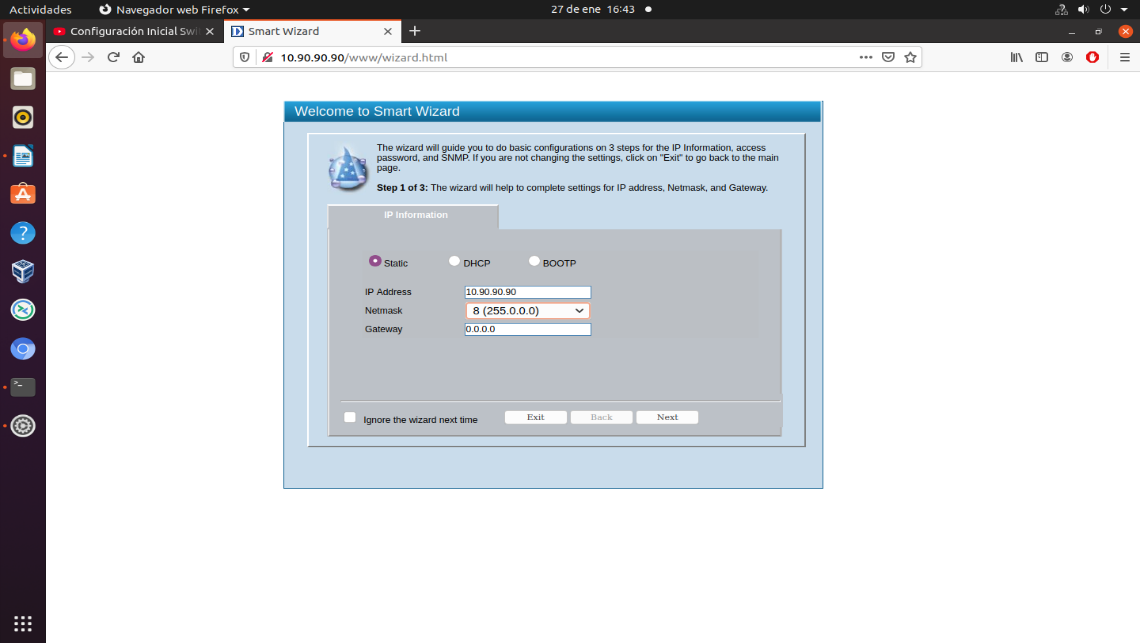
# Configuración de Switch

Para poder acceder al switch tenemos que reiniciar el switch introducir un clip o algo puntiagudo en el agujero donde pone reset y estar 10 segundos.

Una vez realizado el reset tenemos que configurar la red poniendo en la ip 10.90.90.x

mascara 255.255.255.0 y la puerta de enlace 10.90.90.90.

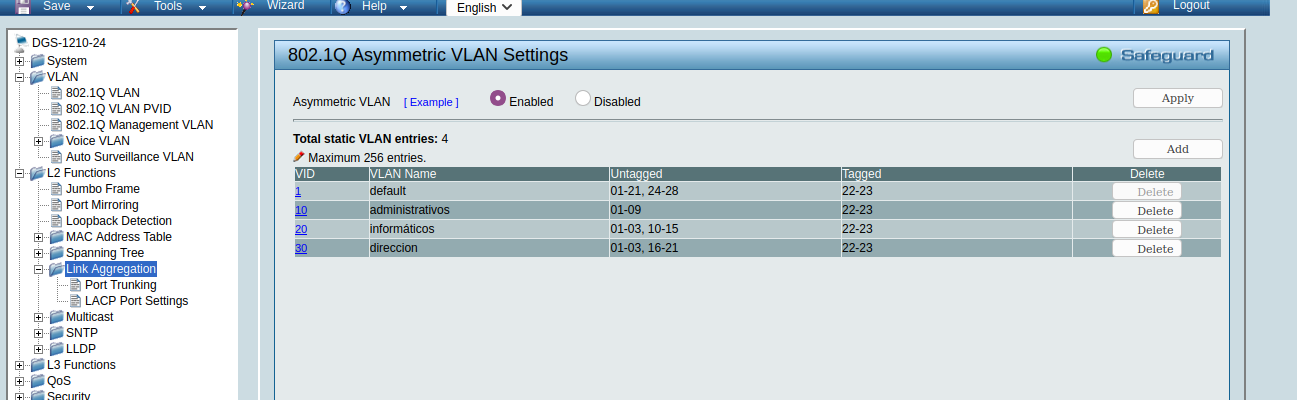
Una vez hecho eso introducimos en el buscador 10.90.90.90 y ya entraremos en el switch, nos pedirá una contraseña en la que tendremos que poner admin porque es la contraseña base del switch.



Ya dentro de la configuración del switch para hacer las Vlan tenemos que ir al apartador Vlan y dentro a 802.1Q Vlan. Asignaremos los grupos de vlan, el nombre y el rango de puerto de cada vlan y también asignaremos en cada vlan los los puertos de port channel y los ponemos en modo tangged, los puertos de las vlan en untagged.

Luego tendremos que ir a la opcion vlan en 802.1Q Vlan PVID, asignaremos a los puertos la vlan a la que pertenece.

Y por ultimo tendremos que asignar los puertos que van a ser port channel para eso tendremos que ir Vlan>L2 Functions>Link Aggregation>Port Trunking y pondremos los puertos que queremos hacer port channel.



# CMMS

**REQUISITOS:**

-Necesitamos una máquina virtual de win 10 de 32 bit.

-Necesitamos la carpeta CalemEAM que la descargamos de este link:

<https://sourceforge.net/projects/calemeam/files/CalemEAM%20Community%20Edition/Calem%20Community%20Edition%20R2.1e/>

-Necesitamos la version 1.7.2 de XAMPP, que la descargaremos en el siguiente link:

<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/1.7.2/>

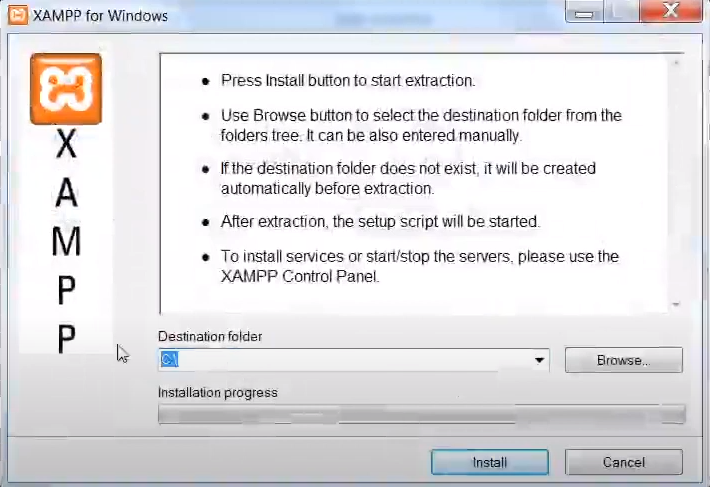
-Necesitamos el programa winrar

**PROCEDIMIENTO:**

-Lo primero que debemos hacer es instalar el servidor web XAMPP que nos aportará a nuestro buscador las herramientas apache y mysql.

Para ello nos metemos en el link que previamente he especificado y elegimos descargar el siguiente archivo:

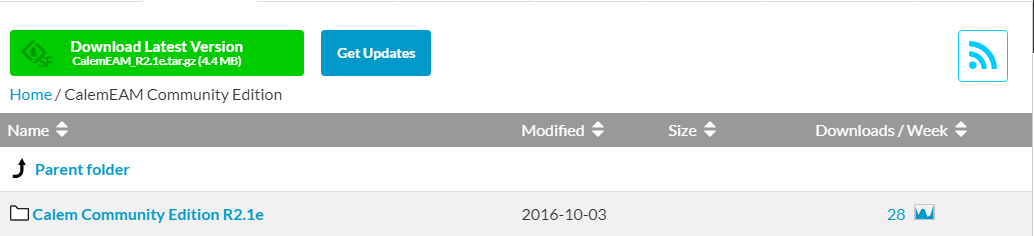
-Después de descargarlo lo ejecutamos y comenzamos la instalación del **XAMPP**



-Una vez instalemos el xampp abrimos el panel de control de xampp y arrancamos los servicios de apache y mysql.

-Una vez arrancados los servicios comenzamos con la descarga e instalación de la herramienta cmms(CalemEAM),para ello, accedemos al siguiente link:

<https://sourceforge.net/projects/calemeam/files/CalemEAM%20Community%20Edition/>

-Abrimos la siguiente carpeta

-Instalaremos el winrar para poder descomprimir el archivo descargado.

-Este es el archivo que debemos descomprimir para poder comenzar con la instalación de la herramienta CMMS



-Una vez descomprimido el archivo nos saldra una carpeta llamada CalemEAM,esta la cortaremos y pegaremos en la carpeta **C:/xampp/htdocs**

-Después de esto, abrimos la shell de mysql desde el panel de control de xampp y procedemos a crear una base de datos en la cual almacenaremos los datos del CalemEAM,

para esto introducimos el siguiente comando:

**mysql -u root -p**

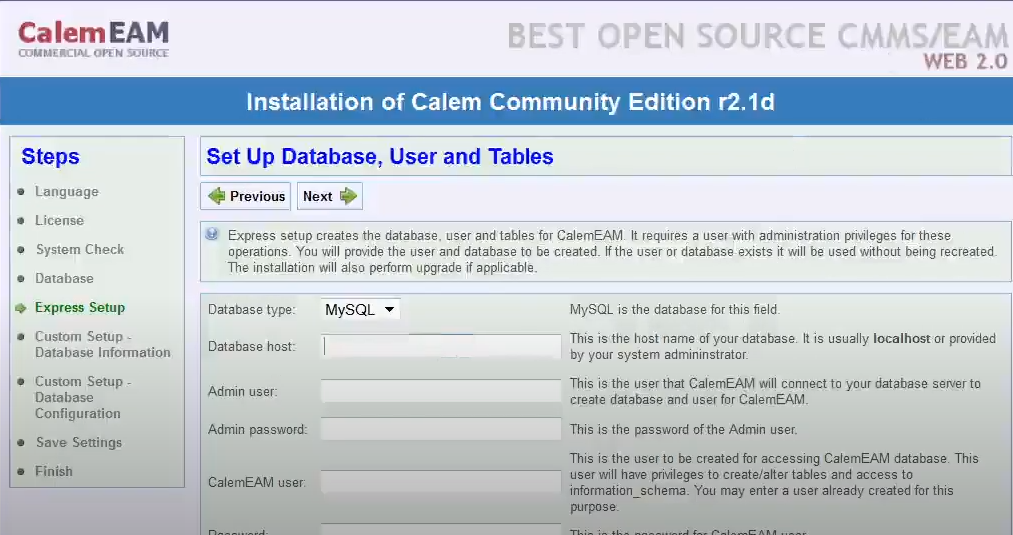
**create database calemeam**

**exit**

-Después de esto accedemos al buscador e introducimos la siguiente ruta:

**localhost/CalemEAM/installation/**

-Si todo está bien nos accedera a la pantalla de instalación del CalemEAM

****

-Introducimos los credenciales de la base de datos, en el database host introducimos (localhost),en el admin user introducimos (root), en la contraseña de admin de db no introducimos nada ya que el usuario root predeterminado de mysql no tiene ninguna password,en el calemeam user introducimos (admin), en la contraseña de este (admin),después introducimos el nombre de la base de datos creada y la damos a siguiente.

-Después de terminar la instalación debemos borrar la carpeta installation y acceder al siguiente link:

<http://localhost/CalemEAM/>

-Introducimos los credenciales predeterminados

USER: **admin**

PASSWORD: **admin\_password**

-Una vez terminado el logue podemos comenzar a utilizar la herramienta.

# Oracle Database 19c

Para empezar, descargamos Oracle Database de la pagina de Oracle, esta nos proveerá de la versión mas reciente (en este caso la 19c), para ello será necesario registrarse proporcionando un email y un teléfono.

Antes de proceder es necesario crear un nuevo usuario de Windows que requerirá un usuario y una contraseña y asegurarse de que no es creada una cuenta de administrador.

Una vez descargado y descomprimido el instalador lo ubicamos en una carpeta dedicada en C (para que todos los usuarios puedan verla)

Para esto es necesario iniciar sesión en Windows con el nuevo usuario y volver a la carpeta en la que habíamos descomprimido el instalador para ejecutar setup.exe con permisos de administrador.

En el primer paso seleccionamos crear y configurar una base de datos de instancia única y pasamos al segundo donde seleccionaremos la clase de escritorio para que sea tan eficiente como pueda. En el tercer paso escogeremos una cuenta Windows existente e introduciremos el nombre de usuario y contraseña creados anteriormente y, en el paso posterior, escogeremos la ruta en la que deseamos realizar la instalación. Para la base de Oracle la ruta será C:\OracleBase19c y es importante quedarse con el nombre de la base de datos global (orcl por defecto) y escoger una contraseña alfanumérica y para terminar otro dato importante es el de la base de datos insertable (orclpdb por defecto).

En el siguiente paso el instalador comprobara los requisitos previos, incluida la RAM disponible y, una vez realizada esta tarea, nos mostrara y dejara exportar los datos de la instalación, si seleccionamos instalar comenzara el proceso cuya duración variara en base al hardware. El firewall de Windows bloqueara algunas funciones del instalador a no ser que, cuando lo pregunte, se le conceda al instalador permiso para comunicarse. Una vez finalizada la instalación nos dará una URL para conectar mediante el explorador (<https://localhost:5500/em>)

# SQL Developer

Desde SQL PLUS (incluido con la instalación de Oracle Database 19c) podemos iniciar sesión introduciendo: ‘/’ as sysdba

Después crearemos un nuevo usuario común (contienen c## como prefijo): create user c##scott identified by tiger;

Y le concederemos los siguientes permisos con: grant resource,contect,dba to c##scott;

Tras salir volveremos a entrar, pero esta vez como c##scott y contraseña tiger para verificar que podemos contarnos con este usuario y, de ser así, podremos cerrarlo y proceder a conectar SQL Developer.

En la esquina superior izquierda de SQL Developer veremos un gran + verde y si lo pulsamos nos saldrá la ventana para realizar una nueva conexión a una base de datos.

El nombre de la conexión es arbitrario (pero no puede dejarse vacío), el usuario y su contraseña deben ser los creados antes desde SQL PLUS.

El tipo de conexión será básica, el nombre del host localhost, el puerto será el 1521 y el SID será orcl (si no hemos cambiado ninguno de estos valores durante la instalación de Oracle Database19c). Y con esto tendríamos acceso completo a la base de datos desde SQL Developer.

# Instalacion Symfony

Lo primero necesitamos un máquina virtual ubuntu server.

Una vez dentro instalamos todos los modulos de php 7.2:

* **apt install php7.2 php7.2-cli php7.2-common php7.2-mbstring php7.2-gd php7.2-intl php7.2-xml php7.2-mysql php7.2-zip php7.2-pgsql php7.2-dev php7.2-bcmath php7.2-xsl php7.2-xmlrpc**

Después de instalar los módulos instalamos mcrypt:

* **apt install mcrypt**

Luego actualizamos las librerías:

* **apt upgrade**

Lo siguiente que haremos es instalar php-mbstring phpunit:

* **apt install php-mbstring phpunit**

Mas adelante instalaremos los modulos que faltan de php:

* **apt install php7.2-common php7.2-cli php7.2-mysql php7.2 php7.2-sqlite3 php7.2-json php7.2-opcache php7.2-readline libapache2-mod-php7.2 php7.2-curl**

Para finalizar instalaremos el git, curl y composer que son gestores de dependencias

* **apt install git curl composer**

# Creacion de sCript De requisitos(Librerias)

Normalmente en una pagina web no es necesario instalar ninguna librería ya que el servidor tiene todo lo basico para el funcionamiento y creacion de servicios pero en nuestro caso al haber elegido un framework el cual esta orientado a objetos ademas de creacion de APIs complejas deberemos de instalar ciertas librerias para el correcto funcienamiento, por lo tanto sabiendo esto y teniendo en cuenta que estas librerias son necesarias instalarlas la primera vez que se pone en marcha el servidor decidimos hacer un script y olvidarnos de estar atento de que cada vez que se acabase la instalacion ejecutaramos la siguiente. En nuestro caso al tener dos maquinas una para la pagina y otra para el webservice (concretamente la maquina que hace las consultas a la BBDD, ya que esto nos parecio mas seguro en vez que el cliente de por si accediese a la BBDD directamente, de esta forma el cliente hace la peticion al webservice y este se encargara de manejar esa peticion, podriamos llamarlo “firewall”, pero como todo tiene sus fallos etc.). Por lo tanto tendremos dos scripts el del webservicerequisites.sh y requisites.sh.

requisites.sh:

webservicerequisites.sh:

# Creacion de servicio que lanza webservice

Para crear este servicio que queremos que se pueda tambien detener ademas, de iniciar cuando el servidor se inicie, necesitaremos dos scripts .sh (ya que el webservice esta definido en una maquina ubuntu)

En el primer script que sera el de inicio escribiremos lo siguiente:

#!/bin/bash

cd /home/nombreUsuario/NombreCarpetaProyecto

git pull origin

cd /home/nombreUsuario/NombreCarpetaProyecto/WS

php bin/console server:run 192.168.4.x:8000

En estas lineas basicamente lo que estamos haciendo es acceder a la ruta donde esta el proyecto actualizarlo en el branch principal (original) y ejecutar el webservice en modo servidor en una ip especifica y en un puerto especifico como en este caso.

En el segundo script que servira para parar el servicio escribiremos lo siguiente:

#!/bin/bash

cd /home/nombreUsuario/NombreCarpetaProyecto/WS

php bin/console server:stop

Este script es mas corto y sencillo y despues de haber visto el de antes, es facil deducir que lo que hacemos es acceder a la carpeta y parar el webservice (o servidor) que esta lanzado en la ip especifica y el puerto especifico de antes.

Lo siguiente sera ubicar estos dos archivos en la siguiente ruta: /usr/local/bin. Les daremos permiso de ejecucion, es decir hacer un chmod 755 nombreDelArchivo.sh.

Por ultimo en /etc/systemd/system crearemos un archivo .service (el archivo que se ejecutara nada mas iniciar el ordenador o mas conocido como servicio como su propio nombre indica).

En este archivo lo que escribiremos sera lo siguiente:

[Unit]

Description = descripcion que quieras poner

After=networking.target

[Service]

Type=oneshot

ExecStart=/usr/local/bin/nombreArchivo.sh

ExecStop=/usr/local/bin/nombreArchivo.sh

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

De esta forma definimos que script se tiene que ejecutar dependiendo si el ordenador se esta iniciando o estemos diciendo que queremos parar el servicio etc.

Antes de darlo por finalizado haremos ciertas comprobaciones de si el servicio esta funcionando bien, las comprobaciones son las siguientes:

systemctl start nombreServicio.service

systemctl status --no-pager nombreServicio.service

systemctl stop nombreServicio.service

Si todo va bien y en las comprobaciones dice que el servicio esta activado, lo ultimo que habra que hacer es habilitar el servicio para que cuando se reinicie el servicio se inicie automaticamente, y el comando es el siguiente:

systemctl enable nombreServici

# Creacion de webservice

Teniendo en cuenta que solo hay que recibir informacion, tratarla para despues devolver esa informacion tratada (idea basica de un webservice), tenemos que tener en mente tres cosas la primera que no vamos a necesitar diseño ya que no necesitamos mostrar nada por pantalla, segundo que las peticiones se envian por metodos concretos (POST, GET, etc.) por lo tanto hay que tener en cuenta que datos queremos que se muestren en la url ya que no queremos injecciones sql (este la diferencia entre post y get, de todas formas sigue habiendo formas de acceder a la bbdd sin necesidad de que utilizemos esto), y por ultimo si hay alguna forma de generar menos accesos es decir menos urls a las que hacer peticiones (ejemplo: <http://192.168.4.x:8000/crearModUsu>) y podemos juntar distintas acciones en una misma url el funcionamiento sera mas eficaz ya que hay menos codigo ejecutar por cada peticion.

Por lo tanto teniendo en cuenta esto generaremos una funcion publica, que va a ser la que mas vamos a utilizar y es la devolucion de los datos en formato json, apartir de ahí solo necesitamos crear la funciones que basicamente lo que haran es recibir los datos, ver que es lo que hay que hacer con ellos es decir, insertar, actualizar, eliminar, mostrar, etc. Despues generar la respuesta en este caso un array y llamar a la funcion que la convierte en json y la devuelve. El mas complicado de tratar ha sido el de la creacion y modificacion de usuario ya que al tener dos tipos de usuarios (Cliente y Trabajador) se ha tenido que crear una peticion que espicifica que tipo de accion se va a realizar y que tipo de usuario va a ser el que va a recibir esa accion. Otras de las complicadas ha sido la peticion de creacion y modificacion de ya que estas tienen id en el diseño de la bbdd, en cambio en la peticion llega el nombre, de esta forma tienes que hacer dos consultas y recorrer dos tipos de datos distintos la tabla artista y la tabla maquetas (otra forma hubiera sido crear una clase propia para despues crear un dql y extraer el dato mediante el nombre y aunque en eficacia sea mas optimo decimos que no ibamos a implementarlo, almenos de momento)

# Hardware

TORRE CERRADA:

gigabyte H55M-S2H LGA1150 + intel HD graphics

intel core i3 540

ram ddr3 8gb (2x4gb) 2133mhz

hdd: SEAGATE BARRACURA 7200RPM ST3500418AS

TORRE ABIERTA:

GIGABYTE GA-H81M-S2H LGA1150 + intel HD graphics

ram ddr3 8gb (2x4gb) 2133mhz

hdd SEAGATE BARRACUDA 7200RPM ST3170827AS

Intel(R) Core(TM) i5-4460 CPU@ 3.20GHz

SWITCH X2:

D-LINK DGS-1210-24