Creando archivo o proyecto en Linux

1. $ mkdir ‘carpeta’/ (crear carpeta en dirección actual)
2. $ sudo chmod –R 777 ‘carpeta’/ (dar permisos a la carpeta creada)
3. $ curl –sS <https://getcomposer.org/installer> | php (instalar composer en carpeta creada)
4. $ chmod +x /’carpeta’/composer.phar (dar permisos al composer)
5. $ php composer.phar (validar composer)
6. $ sudo php composer.phar create-project symfony/framework-standard-edition ‘nombre a la carpeta que se va a crear (Symfony)’ 2.4.0 (instalación de Symfony)

Se pedirá el sistema de base de datos (pdo\_pgsql para postgres si se requiere MySQL se da ENTER), el nombre de la base de datos, usuario y contraseña cuando se esté realizando la instalación de Symfony

Antes de continuar se debe de crear una base de datos y un usuario para vincularlo con el proyecto.

Creación de base de datos en Shell SQL

1. Sudo su postgres (ingresar a postgres)
2. create database ‘nombre’; (creación de la base de datos)
3. create user ‘nombre’ password ‘pass’; (creación del usuario y contraseña)
4. alter role ‘nombre’ with password ‘pass’; (cambiar el usuario)
5. \c ‘nombre BD’; usar la base de datos

**NOTA:** Los nombres que se le darán al usuario y a la base de datos tienen que ser en minúsculas porque postgres no acepta mayúsculas.

**Creación de tablas y formularios**

Después de crear la base de datos y vincularla al proyecto se realizaran los siguientes pasos para crear tablas y formularios llamados **entity.**

1. $ php app/console generate:doctrine:entity (creación de las entidades)

En esta parte se pedirá que ingresemos la carpeta Acme y lo haremos de la siguiente forma:

AcmeDemoBundle:’nombre que le daremos al archivo’

Ejemplo: **AcmeDemoBundle:ejemplo**

Se pedirá con qué tipo de lenguaje se trabajara XML, YML, PHP y ANNOTATION seleccionaremos **YML**.

Después nos pedirán los nombres, tipos y longitudes de los campos de las tablas.

Una vez realizada esta labor crearemos las tablas en SQL

1. $ php app/console doctrine:schema:update --dump-sql (consulta todo el doctrine para crear el script SQL)
2. $ sudo rm –fr app/cache/\* (limpiar cache)
3. $ php app/console doctrine:schema:update --force (ejecuta los Script)
4. $ php app/console generate:doctrine:crud (crea los formularios)

Cuando empecemos a crear los formularios insertaremos el nombre de la entidad que creamos al principio ejemplo:

* AcmeDemoBubdle:ejemplo
* Yes
* YML
* Daremos enter
* Yes
* Yes

Y con esto tendremos creado nuestro proyecto con formularios y tablas para hacer altas, consultas y modificaciones

instalacion de assets

1. sudo php app/console assets:install (comando que copia todos los archivos de la carpeta resources /public y las mete en la carpeta web)

Installar git desde terminal

1. sudo apt-get install git-core (instala el programa git)
2. git clone https: (clona un archivo que tengamos en gibhub desde una URL)
3. sudo apt-get install gitk giggle git-cola git-gui gitg (instalacion de git gui)
4. git gui (abre el git gui)
5. git config --global user.name ('usuario') (confirmar usuario en git)
6. git config --global user.email ('correo') (confirmar correo en git)
7. git remote add origins ssh:[//root@linx4.dscorp.com.mx](mailto://root@linx4.dscorp.com.mx):22023/var/www/html/userbundle (descargar formularios de un servidor ) o git init (inicializar git), git pull origins master

Nota: para la realizacion del siguiente comando se necesita tener instalado composer de manera global

1. para realizar la actualizacion del composer es necesario dar permisos al comando (sudo composer.phar update).

Nota: antes de meter los siguientes comandos se necesita abrir en un editor de texto el archivo config.yml que se encuentra en la carpeta app/config

y cambiar lo el parametro por “% database\_driver%”

1. Despues se generara las sentencias sql con el comando (sudo php app/console doctrine:schema:update –dump-sql)
2. al concluir se introducira las sentencias sql generadas anteriormente con el comando (php app/console doctrine:schema:update --force)