

Lista de Comandos

**Guía para desarrollo de WebApp**

 Habilitar extensión para PostgreSQL en Xampp

 extension=pdo\_pgsql

* A
  + B
* Symfony CLI
* Crear proyecto:
  + symfony new <name-proyect> --webapp
* Iniciar el servidor
  + symphony serve
* Añadir el DATABASE\_URL:

DATABASE\_URL="postgresql://postgres:admin@127.0.0.1:5433/DB?serverVersion=15&ch

arset=utf8"

* Escribir un script sql para crear las tablas en el MDB de postgresql
* Importar las tablas como entidades dentro del proyecto
  + symfony console doctrine:mapping:import "App\Entity" annotation --path=src/Entity
* Añadir la anotación para crear un repositorio para cada entidad
  + @ORM\Entity(repositoryClass="App\Repository\UsuarioRepository")
* Regenerar las entidades, crea los getters, setters y repositorios
  + symfony console make:entity --regenerate App
* Crear entidades
  + symfony console make:entity
* Crear un Controlador de Vistas
  + symfony console make:controller
* Crear el Login
  + symfony console make:auth
* Modificar el archivo security.yaml
* Generar el Register
  + symfony console make:registration-form
* Crear CRUD para una Entidad
  + symfony console make:crud
* Generar un Formulario para una Entidad
  + symfony console make:form <name-entity>

# Symfony console

* Version
  + symfony console -V
* Limpiar Cache
  + symfony console cache:clear
* Mostrar las Routes de la App
  + symfony console debug:route
* Crear una Database
  + symfony console doctrine:database:create
* Realizar migration
  + symfony console make:migration



* + symfony console doctrine:migration:migrate
* Creates a Guard authenticator of different flavors
  + symfony console make:auth
* Creates a new console command class
  + symfony console make:command
* Creates a new controller class
  + symfony console make:controller
* Creates CRUD for Doctrine entity class
  + symfony console make:crud

# Symfony Composer

* Display this application versión
  + symfony composer -V
* Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json
  + symfony composer install
* Adds required packages to your composer.json and installs them
  + symfony composer require
* Updates your dependencies to the latest version according to composer.json, and updates the composer.lock file
  + symfony composer update

# Git

* git init
* git remote add <name-remote> [url]
* git remote remove <name-remote>
* git config –global user.aname <name>
* git config –global user.email <email>
* git clone [url]
* git branch -M main
* git branch -a
* git checkout
* git checkout -b
* git add .
* git commit -m “message”
* git pull <name-remote> <name-rama>
* git push <name-remote> <name-rama>
* git log
* git log --oneline
* git reset –soft HEAD~1

# Docker

* Listar todas las Imágenes descargadas
  + docker images
* Descargar Imagen
  + docker pull <name-image>
  + docker pull <name>:<version>
* Eliminar Image
  + docker image rm <name>:<version>
* Crear Contenedor
  + docker create <name-image>
* Crear Docker con nombre personalizado
  + docker create –name <name-docker> <name-image>
* Crear Docker mappeando los puertos
  + docker create -p<port-f>:<port-docker> --name <name-docker> <name-image>
* Ejecutar Contenedor
  + docker start <id-docker>
* Detener Contenedor
  + docker stop <id-docker>
* Mostrar lista de contenedores ejecutados
  + docker ps
* Mostrar lista de todos los contenedores
  + docker ps -a
* Inspeccionar Docker
  + docker inspect <id-docker or name-docker>
* Eliminar Docker
  + docker rm <id-docker or name-docker>
* Eliminar todos los contenedores ejecutados
  + docker container prune
* Crear una Imagen Local
  + docker build <name-imageLocal>
* Enlistar las Imagenes locales
  + docker image ls
  + docker logs –follow <name-docker>
  + docker run –d <image>
  + dicker runn –name <name-docker> -p27017:27017 –d <image>
* $ ls ~/.ssh/
* $ ssh-keygen
* $ cat ~/.ssh/id\_rsa.pub > /dev/clipboard
  + $ git remote add git@gitlab.com:pruebas-telconet-sasf/proyecto-27-03-2023.git