

Manual técnico Sistema de gestión documental Universidad Siglo XX

Base de datos

Realizado en PostgreSQL

Versiones: Postgres v14 o Lastest

1. Instalaciones necesarias

Para que se pueda crear y administrar una base de datos se debe ejecutar el siguiente comando (Linux): **sudo apt install phpmyadmin phppgadmin postgresql mariadb-server apache2**

2. Ingresar como usuario postgres (Linux)

Se ejecuta el siguiente comando: **sudo su - postgres**

3. Ingresar al administrador de postgres

Se ejecuta el comando: **psql**

4. Crear un usuario

Se lo crea con el siguiente comando:

```
postgres=# CREATE USER prueba with PASSWORD '123456';
```

5. Dar privilegios al usuario creado

```
postgres=# ALTER ROLE prueba WITH SUPERUSER;
```

6. Crear la base de datos y asignarle como dueño al usuario que creamos

```
postgres=# CREATE DATABASE db_de_prueba OWNER prueba;
```

7. Ejecutar script para generar base de datos

El script se descarga de la siguiente dirección:

<https://drive.google.com/file/d/134e0xu2AXvSbBbAESEIIBlcCSHFdIJ-R/view?usp=sharing>

Una vez creada la base de datos y el usuario se debe ejecutar el script que crea la base de datos y sus tablas, adicionalmente crea las unidades para el correcto funcionamiento

1. Backend

Realizado con un framework de PHP llamado Yii2

Versiones: PHP >=7.4.0 && Yii2 ~2.0.45

1. Clonar repositorio

Para clonar el repositorio se debe ingresar al siguiente enlace:

<https://github.com/parienteY/backend-sigloxx>. Dentro del cual podrá clonar el repositorio con el código.

2. Instalar dependencias

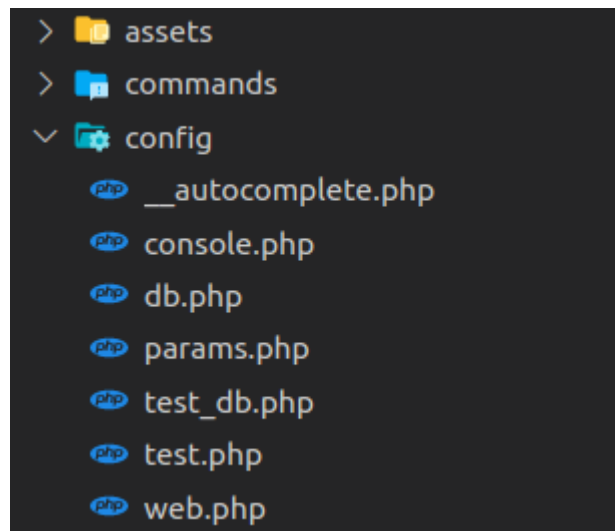
Para poder instalar las dependencias del proyecto correr el comando:
composer install

**Si no cuenta con composer instalado se debe instalar como se indica en este enlace:*

- <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-composer-on-ubuntu-20-04-es> (Ubuntu/Linux)
- <https://norvicsoftware.com/que-es-y-como-instalar-composer-en-windows-10/> (Windows)

3. Conexión a base de datos

Para poder configurar una conexión a la base de datos se debe ingresar a la carpeta **config** dentro de la raíz del proyecto y editar el archivo db.php



Dentro del archivo db.php se debe cambiar la siguiente configuración

```
'class' => 'yii\db\Connection',  
'dsn' => 'pgsql:host=localhost;dbname=gd_odontologia',  
'username' => 'siglo',  
'password' => '123456',  
'charset' => 'utf8'
```

Donde principalmente:

- **dns:** Es el host y separado por un punto y coma el nombre de la base de datos (dbname)
- **username:** Es el nombre del usuario con permisos para acceder a la base de datos
- **password:** Es la contraseña del usuario con permisos para acceder a la base de datos

Una vez configurado correctamente y si dicho usuario tiene acceso a la base de datos el proyecto ya estaría conectado a la base de datos.

4. Migración de tablas necesarias para Rbac

Para añadir las tablas necesarias para el acceso basado en roles dentro del backend se debe correr el siguiente comando: **php yii migrate -migrationPath=@yii/rbac/migrations**

6. Generar roles, permisos y usuarios

Descargar y ejecutar el siguiente script sql:

https://drive.google.com/file/d/1iKYc058hFkH2zKgG1B8-TxqiB8_VSbn0/view?usp=share_link

Los usuario que se crearan por defecto son los siguientes:

Correo	Contraseña
super@gmail.com	123456
extension@gmail.com	123456
gestion@gmail.com	123456
centro@gamil.com	123456
direccion@gmail.com	123456
internado@gmail.com	123456
investigacion@gmail.com	123456
clinicas@gmail.com	123456
academica@gmail.com	123456
sociedad@gmail.com	123456

5. Iniciar el proyecto

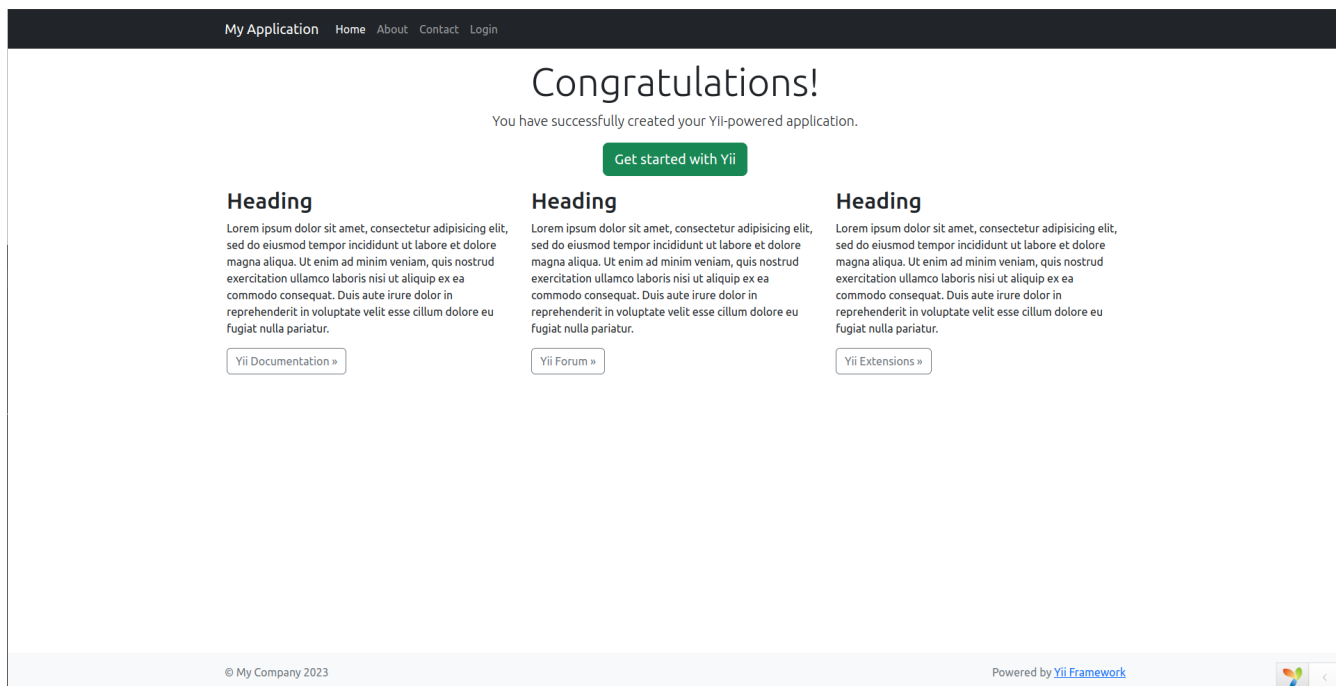
Si se configuro todo correctamente ya se puede iniciar el backend con el siguiente comando: `php -S 192.168.1.12:8080`

Donde la ip 192.168.1.12:8080 es solo un ejemplo ahí se podría poner cualquier dirección ip y puerto dentro de la red de su preferencia

6. Comprobar que el proyecto se ha iniciado

Se debe ingresar a la dirección con la ip y puerto que se ha especificado en el paso anterior.

Ej: <http://192.168.1.12:8080>



Si aparece una pagina como esta el proyecto se ha iniciado correctamente.

Frontend

Realizado con la librería de Javascript llamada React JS

Versiones: Node 16.18.1 && React 18.2.0

1. Instalaciones necesarias

Para instalar NodeJS y NPM ingresar al siguiente enlace y seguir los pasos

- <https://www.cursosgis.com/como-instalar-node-js-y-npm-en-4-pasos/> (Windows)
- <https://www.hostinger.es/tutoriales/instalar-node-js-ubuntu/> (Ubuntu/Linux)

2. Clonar el código del repositorio

Descargar el código desde el siguiente enlace:

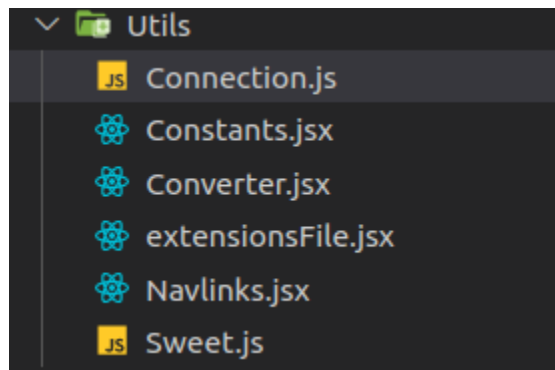
<https://github.com/Stumbleine/odontologia-xx>

3. Instalar dependencias

Para poder instalar las dependencias del proyecto se debe ejecutar el siguiente comando: **npm i**

4. Configurar conexión al backend

Para configurar lo se debe ingresar a la carpeta **Utils** dentro de la raíz del proyecto y editar el archivo **Connection.js**



Se debe editar la siguiente parte del código

```
export const URL = 'http://192.168.1.12:8080';  
const API = axios.create({  
  baseURL: URL,  
  responseType: 'utf8',  
});  
  
export default API;
```

Cambiar la dirección dentro de la constante "URL" a la dirección en la cual se está corriendo el backend

4. Configurar IP y puerto para ejecución del proyecto

Para poder configurar la ejecución del proyecto en un puerto e ip específico se debe configurar el archivo **package.json** que se encuentra en la raíz del proyecto.

4. Ejecutar el proyecto

Una vez instaladas todas las dependencias y configurada la conexión a la base de datos ya se puede ejecutar el proyecto con el siguiente comando: **npm start**

Buscar las siguientes líneas de código:

```
"scripts": {  
  "start": "HOST=127.0.0.1 PORT=3001 react-scripts start",  
  "build": "react-scripts build",  
  "test": "react-scripts test",  
  "eject": "react-scripts eject"  
},
```

Y configurar el script "start" las variables **HOST** para la ip y **PORT** para el puerto