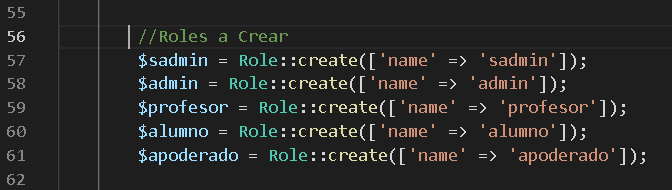
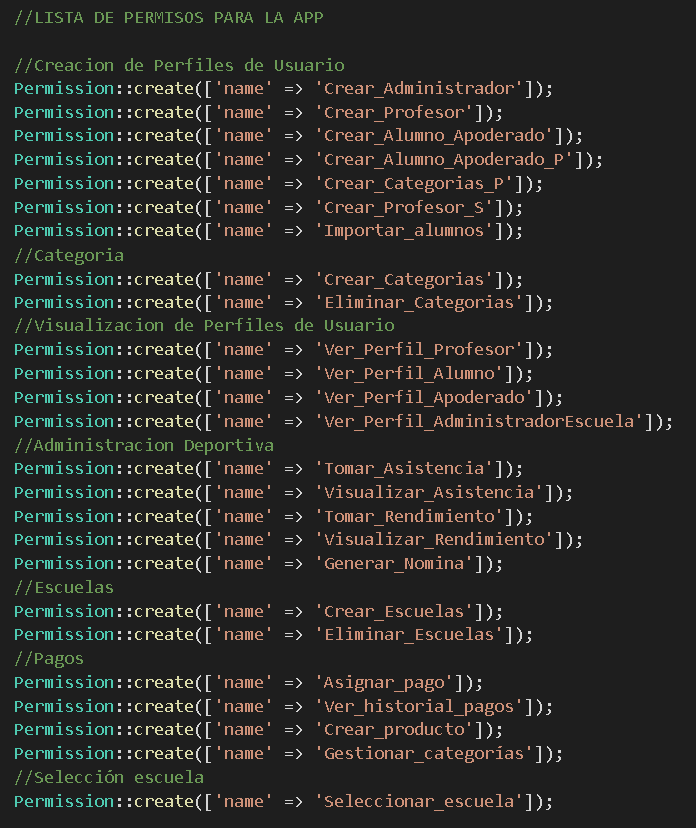
# Tutorial de Totalges para desarrolladores.

Dentro de las cosas puntuales a tener en cuenta al momento de desarrollar son la creación de los roles y permisos para cada usuario.

## Creación de roles:

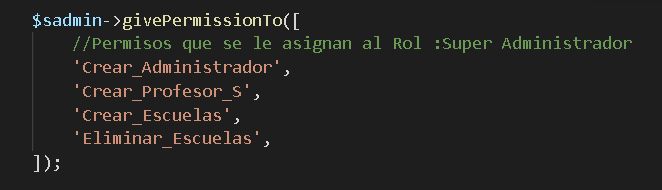


## Creación de permisos:



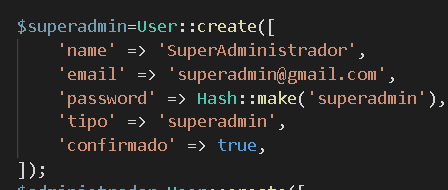
Esto lo que hace es crear en la base de datos los permisos y roles necesarios para que funcione la aplicación.

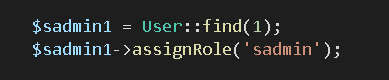
## Asignación de permisos:



En el caso del superadmin tiene 4 permisos asignados solamente.

Cada vez que se cree un usuario se le debe asignar su respectivo rol para que dicho usuario pueda tener acceso a esos permisos:

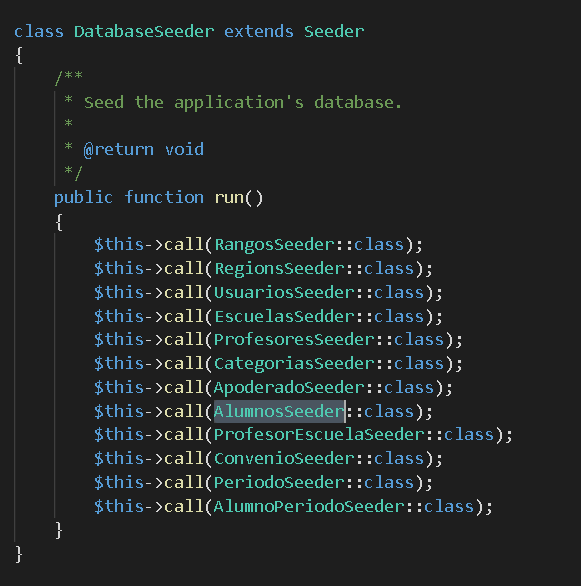
****



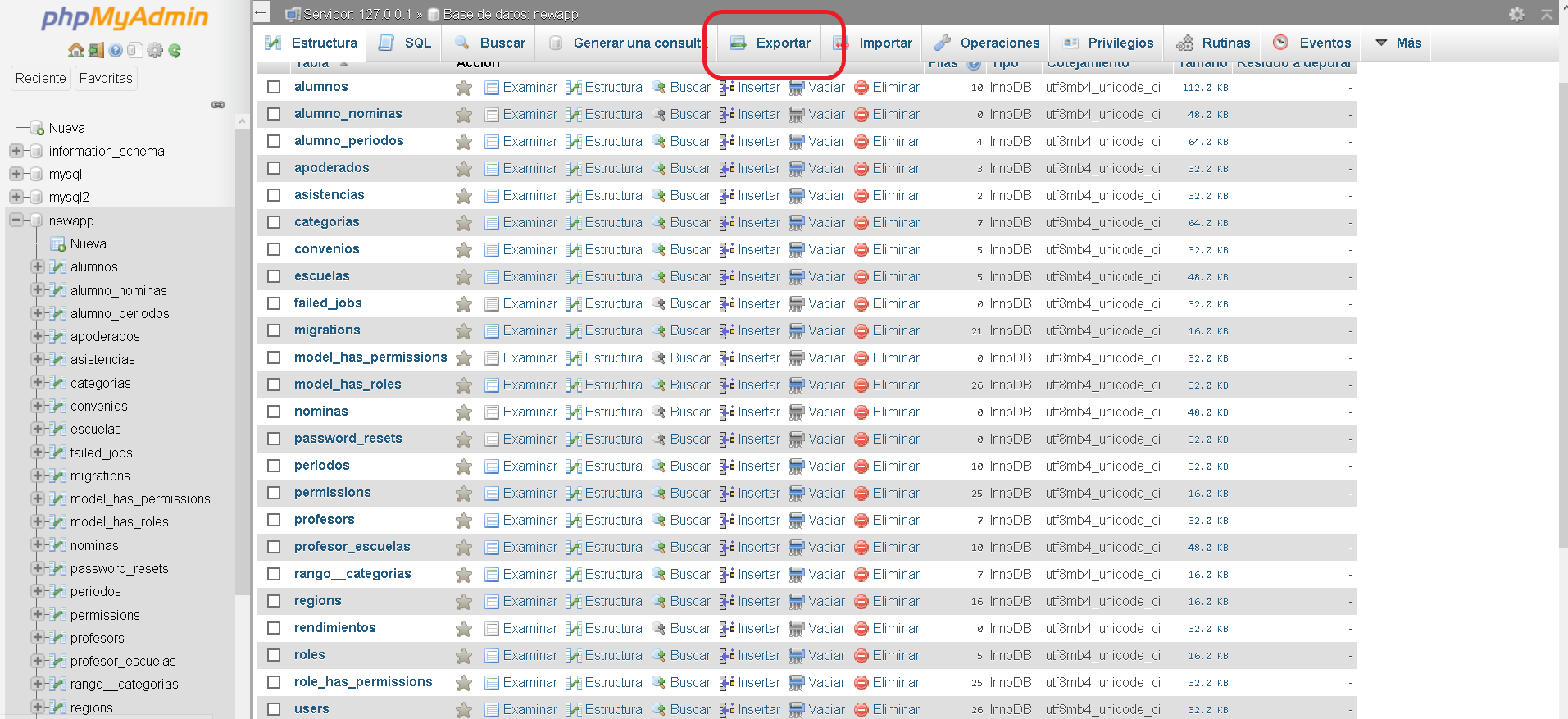
Obtenemos el primer usuario creado que fue el superadmin y le asignamos el rol sadmin con la función assignoRole, este proceso de assignRole se debe hacer cada vez que se cree un usuario.

Con esos datos establecidos ya se puede usar la aplicación, pero primero hay que cargar los datos en la base de datos con:  
php artisan migrate:fresh --seed

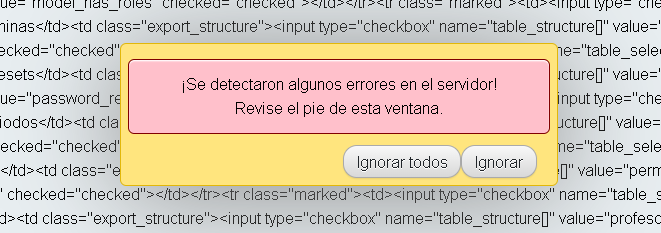
Lo que hace este comando es cargar en la base de datos toda la información que contienen los Seeder que se llamaron en la función run() del archivo DatabaseSeeder.php:



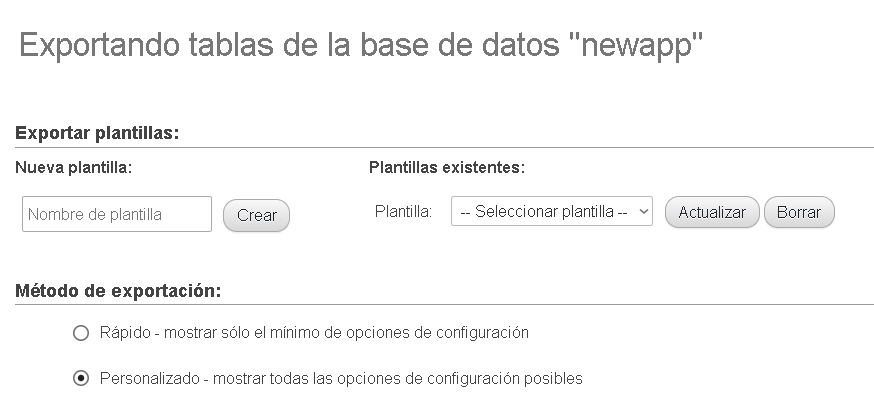
Una vez con la base de datos cargada vamos a <http://localhost/phpmyadmin>, donde seleccionamos la opción exportar en la base de datos en cuestión:



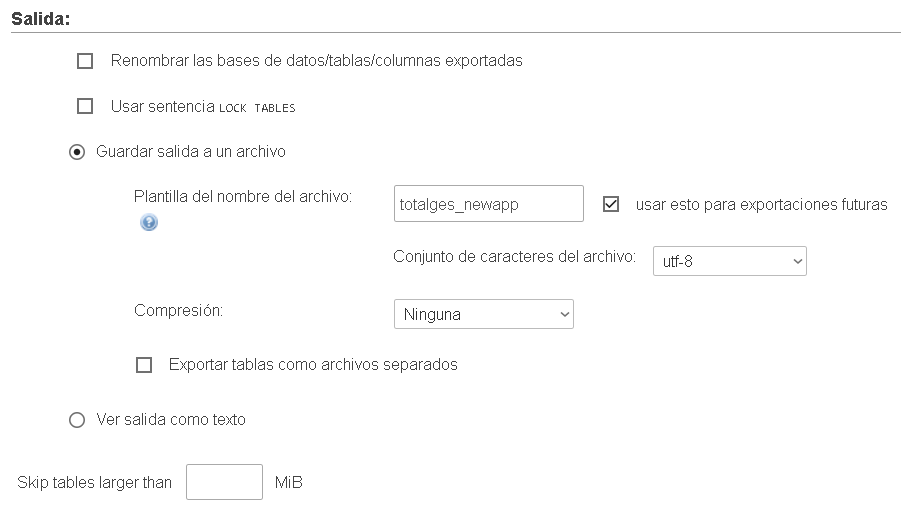
Hay que tener en cuenta algunos aspectos al momento de exportar. Primero nos dará un mensaje de error, a lo cual debemos seleccionar “ignorar todos”:



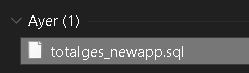
Después debemos seleccionar un método de exportación personalizado:



Vamos bajando y en la sección de salida debemos poner el nombre que la base de datos tiene en el servidor, que en este caso era “totalges\_newapp”, es importante este punto, sino dará error al momento de importar.



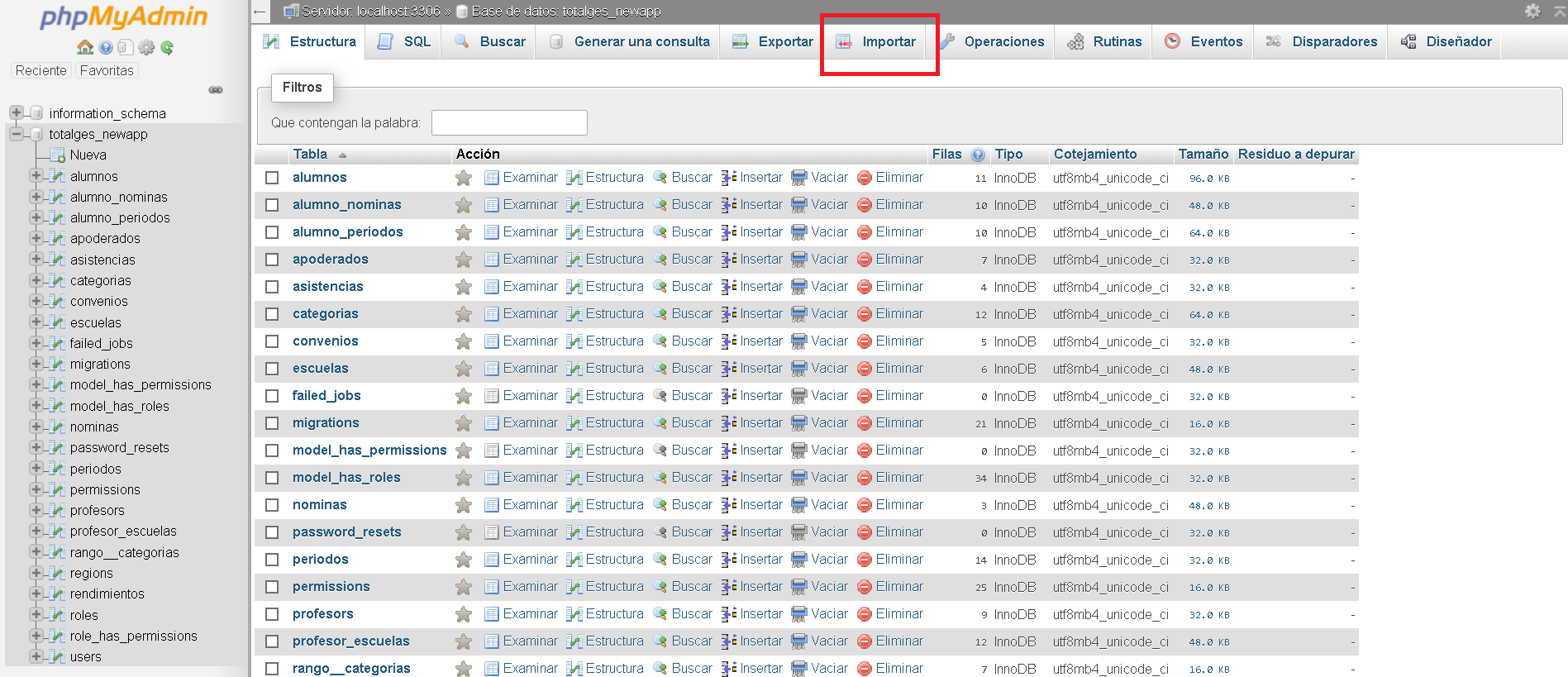
Bajamos hasta el final y seleccionamos continuar, y se nos descargará un archivo .sql con el nombre que le dimos:



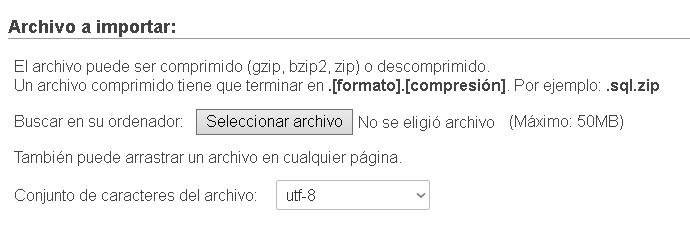
El comando “php artisan migrate”no funciona en el servidor por lo cual hay que importar la base de datos en local a la base de datos en el servidor, para lo cual se selecciona la opción phpMyAdmin en CPanel:



Después se abre la base de datos y se selecciona la opción importar en la base de datos indicada:

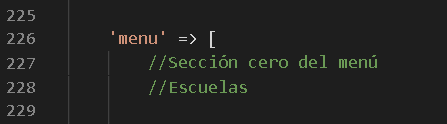


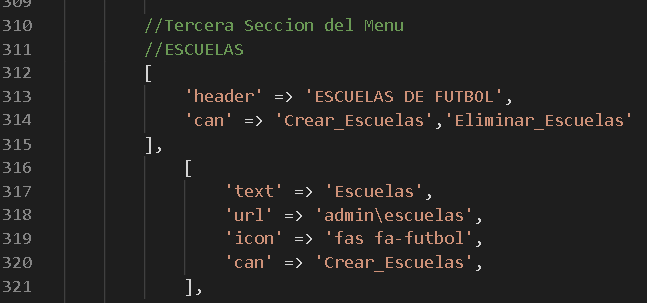
Entonces nos pedirá el archivo .sql que se descargó, lo ingresamos, bajamos al final y apretamos en continuar.



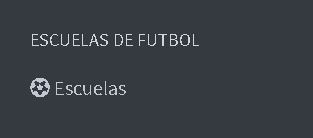
Con eso ya tendríamos la base de datos con sus permisos y roles en el servidor.

Para poder enlazar los permisos a opciones en el panel lateral es necesario editar el archivo adminlte.php en la sección menu:





Por ejemplo, podemos agregar un encabezado y las opciones que tendrá dicho encabezado. Esta configuración se refleja en el panel lateral de la sesión del superadmin:



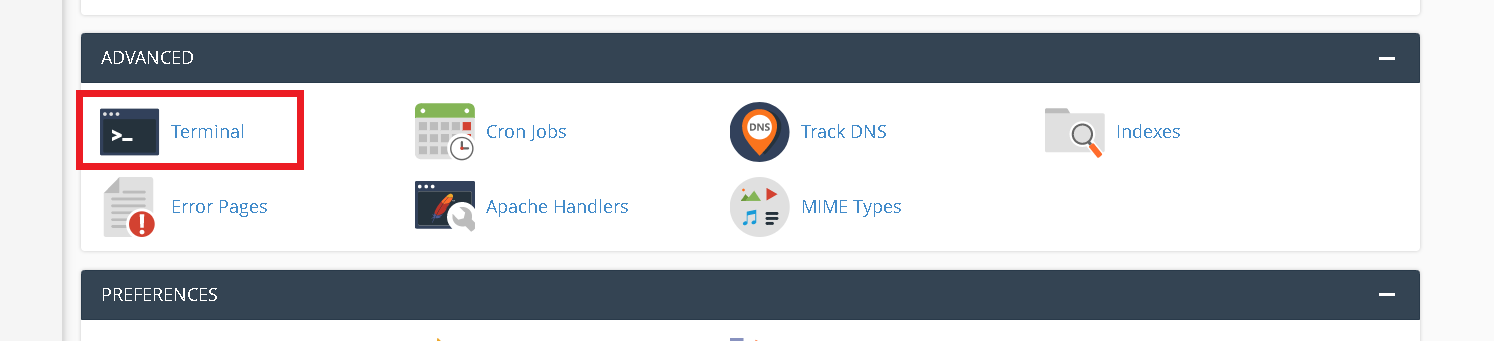
Cada vez que se haga un cambio en el archivo adminlte.php o en cualquier archivo de configuración del proyecto, se deben ejecutar estos comandos en la terminal:

**php artisan config:cache**

**php artisan cache:clear**

**php artisan config:clear**

En caso de estar trabajando en el servidor será necesario acceder a la terminal en las opciones avanzadas:



Una vez dentro de la terminal estaremos en la raíz del directorio principal, por lo cual deberemos ir al directorio en el que se encuentra el proyecto:

**cd public\_html**

Y luego de eso ejecutar los siguientes comandos:

**php artisan config:cache**

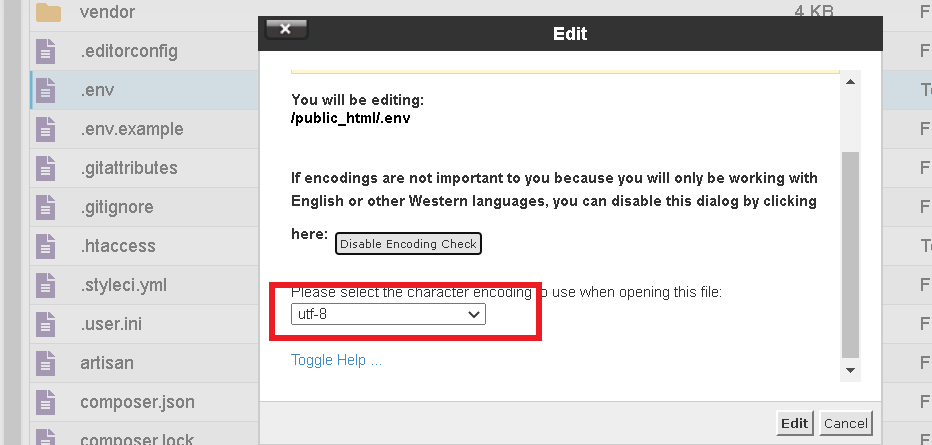
**php artisan cache:clear**

**php artisan config:clear**

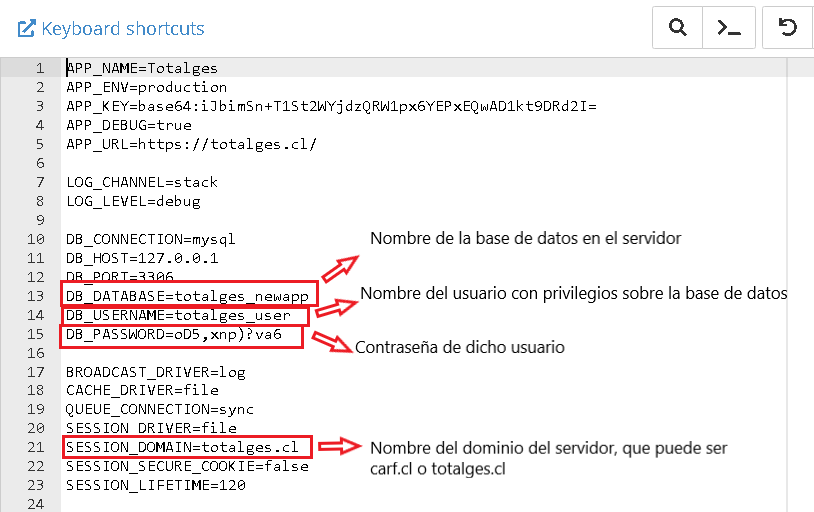
Con eso la configuración en el servidor habrá sido reiniciada y los cambios realizados se podrán apreciar.

Luego de ejecutar esos comandos se habrá reiniciado la configuración.

Cada vez que se edite un archivo en el servidor se recomienda verificar que la codificación es utf-8, por ejemplo, si deseamos editar el archivo .env:

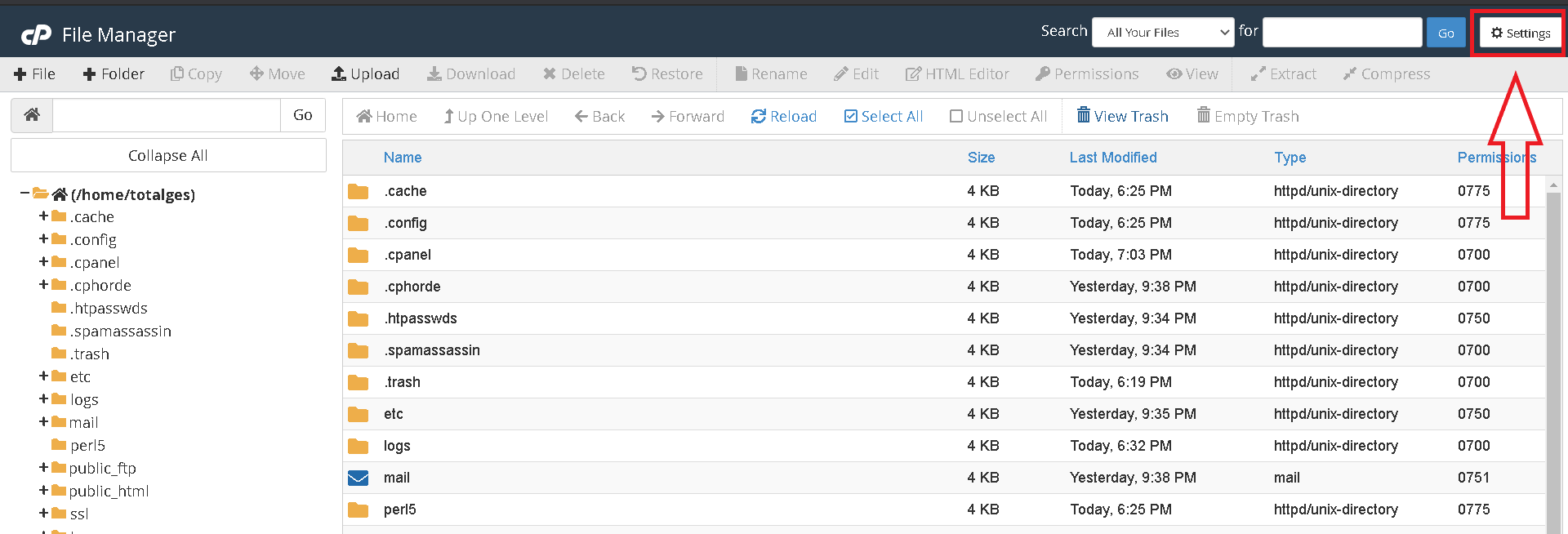


Otra cosa importante es que dentro del archivo .env se debe prestar mucho cuidado a los siguientes valores:

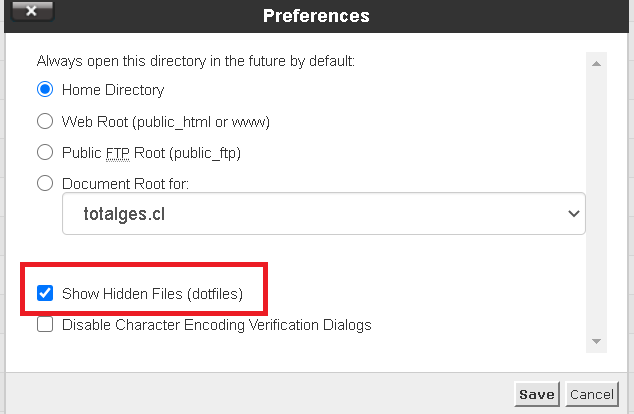


Cabe mencionar que luego de cualquier cambio en el archivo .env es necesario también ejecutar los comandos para reiniciar la configuración.

En caso de no encontrar el archivo .env en el directorio raíz del proyecto es necesario configurar el administrador de archivos en CPanel:

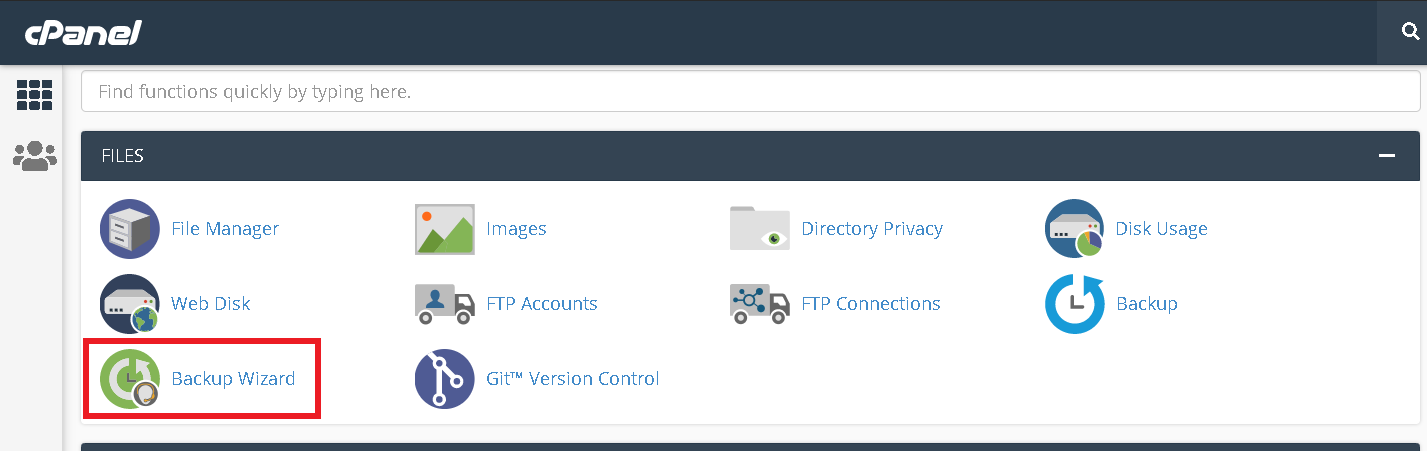


Luego de que se desplieguen las opciones debemos marcar la de ver archivos ocultos:

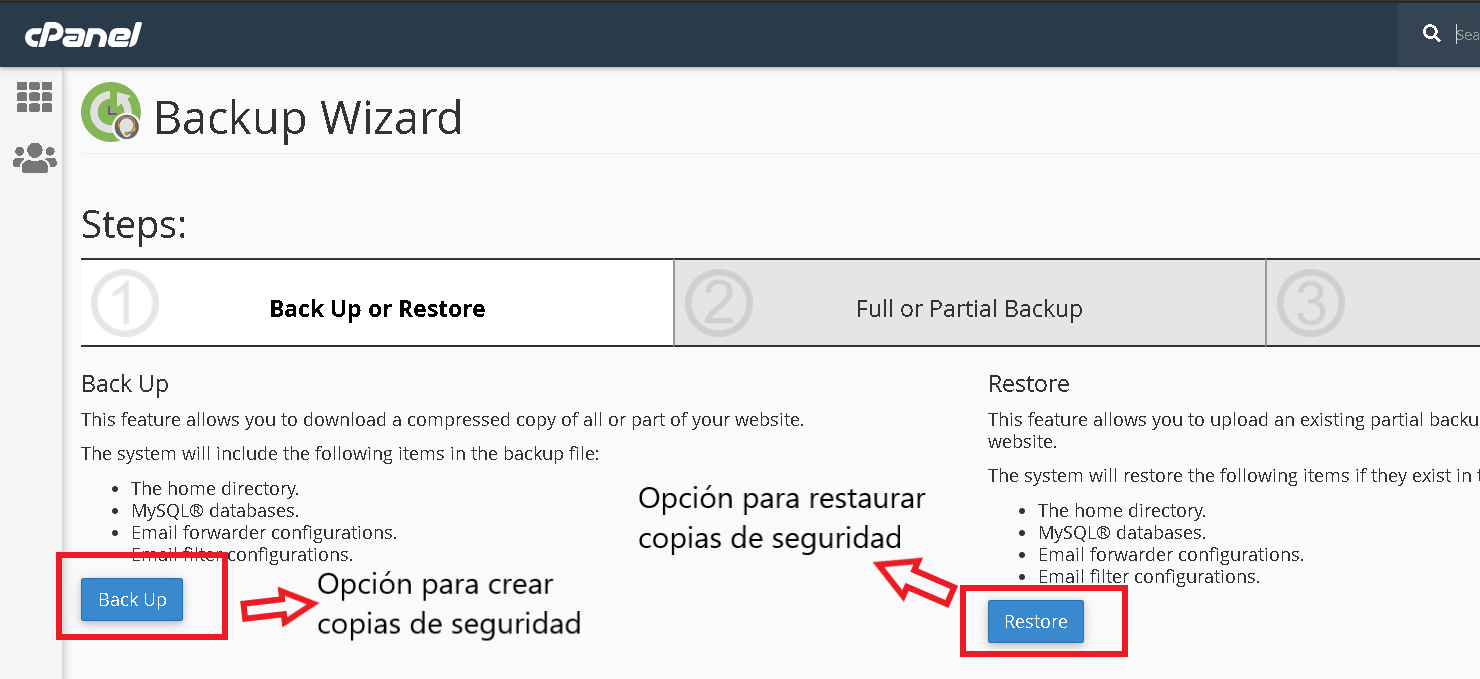


Una vez hecho esto ya podremos ver los archivos ocultos incluyendo al archivo .env.

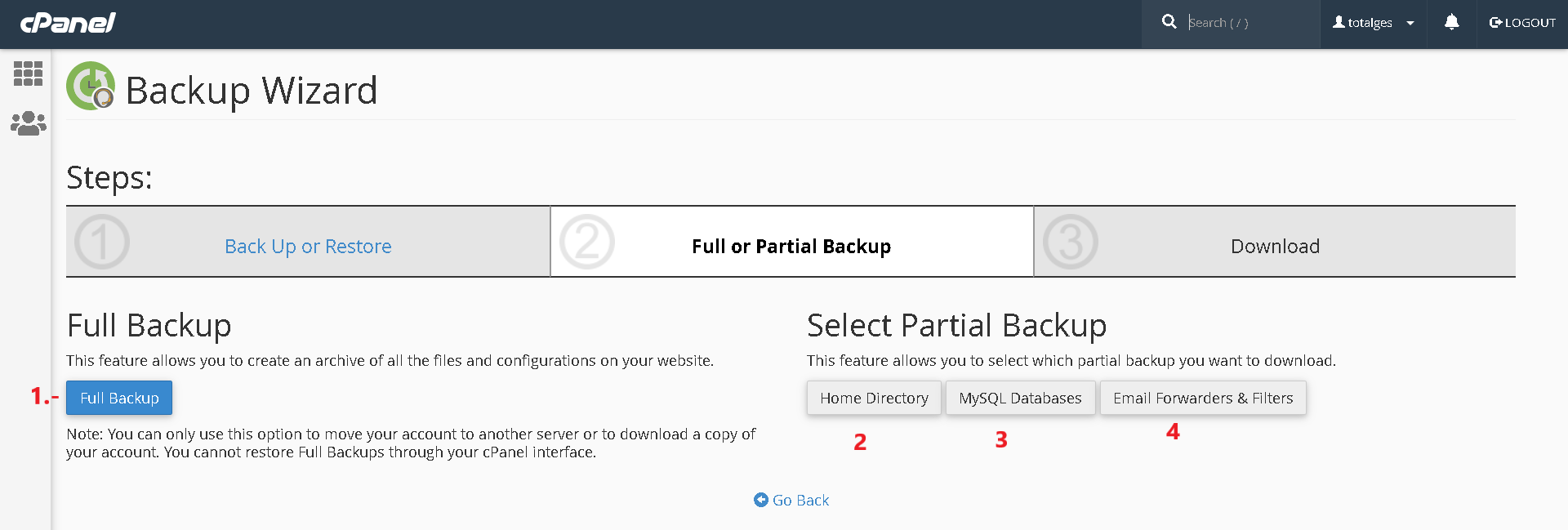
Para poder manejar las copias de seguridad debemos acceder a la opción Backup Wizard en la sección de archivos:



En esta sección encontraremos las opciones de descargar y restaurar copias de seguridad:



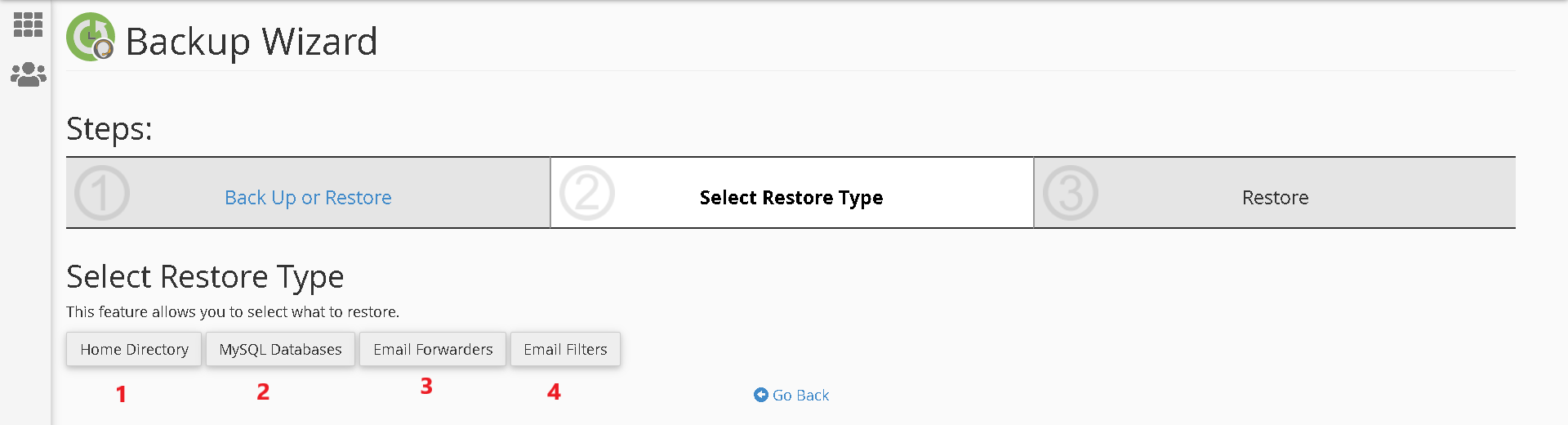
En caso de seleccionar la opción de crear la copia de seguridad, encontraremos las siguientes opciones:



1. Hacer una copia de seguridad de todo el dominio.
2. Hacer una copia de seguridad de solo los archivos del dominio
3. Hacer una copia de seguridad de la base de datos
4. Hacer una copia de seguridad del servidor de correo

Una vez seleccionada una opción se nos redireccionará a una nueva ventana donde tendremos la posibilidad de descargar dicha copia de seguridad.

En caso de seleccionar la opción de restaurar la copia de seguridad, podremos seleccionar las siguientes opciones:



1. Restaurar los archivos del dominio
2. Restaurar las bases de datos
3. Restaurar los registros de email
4. Restaurar los filtros de email.

Un detalle importante también es que cada vez que quieran restaurar una copia de seguridad, ésta debe ser cargada en un fichero que tenga formato ZIP.

Comandos de Laravel:

Estos comandos son los que más he usado en mi tiempo programando, como se puede ver no hay un comando para crear vistas, por lo cual es necesario crearlas uno mismo desde el IDE.

**php artisan serve ->** Inicia la aplicación de Laravel en un servidor local.

**php artisan make:model carpeta/Nombre --migration ->** Crea un modelo y automáticamente crea su tabla de migraciones, en caso de no introducir una carpeta contenedora el modelo se creará en la dirección raíz de la carpeta Model.

**php artisan make:controller carpeta/NombreController ->** Crea un controlador dentro de una carpeta o en la dirección raíz del directorio Controller (En caso de no introducir una carpeta).

**php artisan migrate:fresh --seed ->** Crea las bases de datos según las migraciones que estén creadas y carga los datos definidos en los Seeders.

**php artisan make:seeder NombreSeeder ->** Crea un seeder de alguna entidad/modelo para cargar datos automáticamente.