



CENTRO INTEGRADO de FORMACIÓN PROFESIONAL
AVILÉS

CFGS: DESARROLLO DE APLICACIONES WEB

Modalidad: Presencial

Curso: 2022-2023

Agenda Web y Gestión

MANUAL DE PROGRAMADOR

Avilés, junio 2023

Autor: García Martínez, Luis Miguel

Tutor/a Individual: Iglesias, Yolanda

Tutor/a Colectivo: Iglesias, Yolanda

1. Índice

MANUAL DE PROGRAMADOR	1
1. Índice	2
2. Introducción	3
3. Requerimientos técnicos de sistema	3
4. Descripción de las tablas de la base de datos	4
5. Definición y Descripción de las principales variables del sistema	6
6. Diccionario de Datos	7
10. Despliegue de la aplicación	9
10.1. Instalación	9
10.2. Administración y Configuración	9
11. Índice de figuras (imágenes / tablas)	10
12. Siglas y acrónimos	10
13. Bibliografía	10

2. Introducción

Este proyecto consiste en la creación de un programa web el cual pueda crear y gestionar grupos de trabajos, y fijar actividades atreves de estos, o para que los propios usuarios puedan marcarse sus propias actividades para sí mismos, todo esto atreves de una aplicación hecha en Laravel, y para el diseño principalmente realizado con Bootstrap. Este proyecto es debido a la dificultad para recordar las tareas que se han de realizar y para cuando deben de ser realizadas, además de facilitar la realización de tareas en grupo y almacenar archivos.

Este proyecto se basa en:

- Diseño de la Base de Datos.
- Creación de las Migrations para la Base de Datos.
- Creación de usuarios
- Controles de acceso a los usuarios
- Creación de Tareas
- Creación de Grupos
- Subida y Descarga de Archivos

3. Requerimientos técnicos de sistema

Los requerimientos técnicos son:

- Tener instalada Composer en el equipo, ya que sin el la instalación de Laravel es imposible.
- Tener Instalado Laravel, ya que atreves de sus variables, comandos y demás, será lanzaremos y permitiremos el correcto funcionamiento de nuestra aplicación web.
- Tener instalado XAMPP, el cual, pese a no utilizar todas las opciones que nos ofrece, necesitaremos para poder usar la base de datos con MySQL y permitirnos el acceso a la aplicación web atreves de Apache.
 - Dentro de XAMPP, tener configurado correctamente Apache, con el cual mostraremos la aplicación web.
 - Dentro de XAMPP, tener configurado correctamente MySQL, con el cual gestionaremos nuestra base de datos.

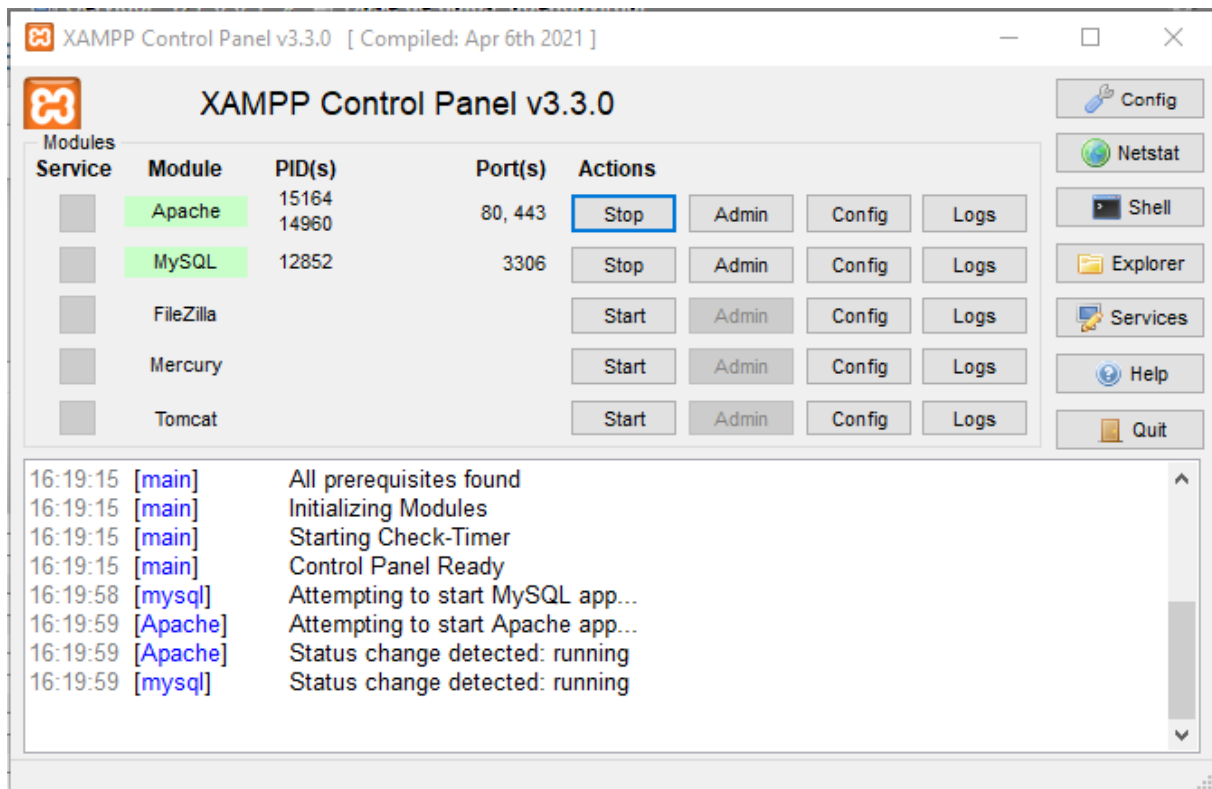


Ilustración 1 Correcto estado del XAMPP

4. Descripción de las tablas de la base de datos

Hay 9 tablas que pertenecen directamente al proyecto, las cuales son:

1. Actividades

- Cod_Actividad: Es la clave Primaria de la tabla, y es autoincrementar.
- Nombre_Actividad: Es donde almacenaremos el nombre poseerán nuestras actividades, es de tipo text.
- Descripcion_Actividad: Es donde almacenaremos la descripción que poseerán nuestras actividades, es de tipo text.
- Estado_Actividad: Es donde almacenaremos el estado nuestras actividades, es de tipo tinyint, siendo 0 cerrado y 1 abierto.
- Fecha_Cierre: Es donde almacenaremos la fecha de cierre de nuestras actividades, es de tipo date y puede ser fijada como nula, con ella controlaremos que los usuarios puedan entregar actividades.
- Hora_Cierre: Es donde almacenaremos la hora de cierre de nuestras actividades, es de tipo date y puede ser fijada como nula, con ella controlaremos que los usuarios puedan entregar actividades.

2. Archivos

- Cod_Archivos: Es la clave Primaria de la tabla, y es autoincrementar.
- Ruta_Archivos: Es donde almacenaremos el nombre de nuestros archivos, es de tipo text.

3. Grupos

- Cod_Grupo: Es la clave Primaria de la tabla, y es autoincrementar.
- Nombre_Grupo: Es donde almacenaremos el nombre poseerán nuestros grupos, es de tipo text.

- c. Descripcion_Grupo: Es donde almacenaremos la descripción poseerán nuestros grupos, es de tipo text.
- 4. Users
 - a. Id: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Name: Es donde almacenaremos el nombre de nuestros usuarios, es de tipo varchar de tamaño máximo de 255.
 - c. Email: Es donde almacenaremos el correo de nuestros usuarios, es de tipo varchar de tamaño máximo de 255, y es utilizado como índice.
 - d. Password: Es donde almacenaremos la contraseña de nuestros usuarios, es de tipo varchar de tamaño máximo de 255, esta es cifrada antes de ser subida
- 5. Act_Arch
 - a. Cod_Act_Arch: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Cod_Actividad: Clave foránea, referencia la clave primaria de "Actividad", y es un bigint de tamaño máximo de 20
 - c. Cod_Archivos: Clave foránea, referencia la clave primaria de "Archivos", y es un bigint de tamaño máximo de 20
- 6. Grup_Act
 - a. Cod_Grup_Act: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Cod_Grupos
 - c. Cod_Actividades: Clave foránea, referencia la clave primaria de "Actividades", y es un bigint de tamaño máximo de 20
 - d. Codigo_Usuario: Clave foránea, referencia la clave primaria de "Usuarios", y es un bigint de tamaño máximo de 20
- 7. Usu_Act
 - a. Cod_Usu_Act: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Codigo_Usuario: Clave foránea, referencia la clave primaria de "Usuarios", y es un bigint de tamaño máximo de 20
 - c. Cod_Actividad: Clave foránea, referencia la clave primaria de "Actividad", y es un bigint de tamaño máximo de 20
- 8. Usu_Arch
 - a. Cod_Usu_Arch: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Codigo_Usuario: Clave foránea, referencia la clave primaria de "Users", y es un bigint de tamaño máximo de 20
 - c. Cod_Archivos: Clave foránea, referencia la clave primaria de "Archivos", y es un bigint de tamaño máximo de 20
- 9. Usu_Gru
 - a. Cod_Usu_Gru: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Codigo_Usuario: Clave foránea, referencia la clave primaria de "Usuario", y es un bigint de tamaño máximo de 20
 - c. Codigo_Grupos: Clave foránea, referencia la clave primaria de "Grupos", y es un bigint de tamaño máximo de 20
 - d. Administrador: Es donde almacenaremos si nuestros miembros del grupo son administradores o no del mismo, es de tipo tinyint, siendo 0 no administrador y 1 administrador.

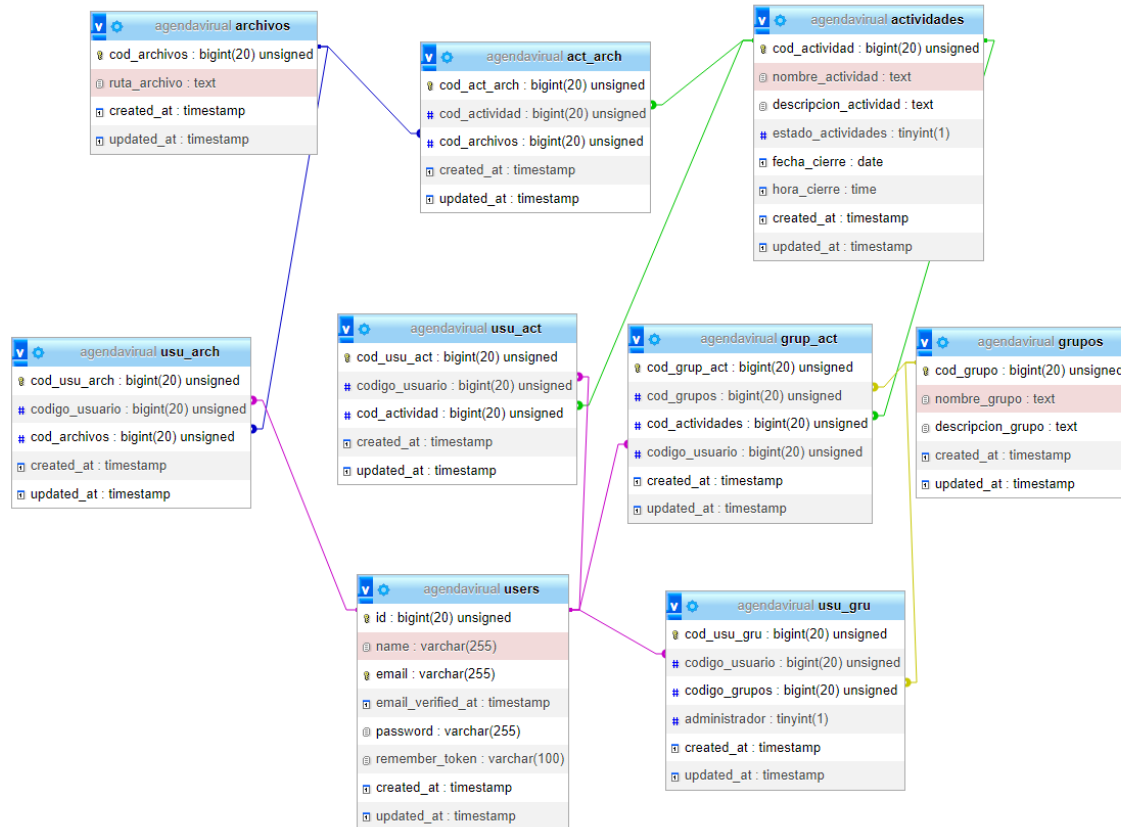


Ilustración 2 Vista de las Tablas de la Base de Datos

5. Definición y Descripción de las principales variables del sistema

El centro neurológico de nuestra aplicación es “ControladorAgenda.php”, aquí se encuentran todas nuestras funciones y es donde obtenemos los datos que serán mostrados al usuario, aquí podríamos dividir el controlador en 6 partes, una por cada agrupación de paginas que controla, pero de estas 6 hay 3 que destacan sobre las demás.

Las encargadas de las tareas individuales son unas de estas tres, donde nos encontramos con “getTareas”, con el cual se muestra al usuario las tareas individuales que están relacionadas con su cuenta a través de “Usu_Act”, permitiendo le así ver sus tareas propias, también nos encontramos con “getTareaEspecifica”, el cual requiere que le pasemos el id de la tarea que deseamos visualizar por la URL, con esto lo que logramos es que el usuario vea todos los datos de esa tarea y le permitimos a través de otras funciones desde descargar o cargar los archivos de la tarea, borrar la o editar la con “editarTareaIndi”, con el cual podemos editar la tarea en la que nos encontramos, cambiando todos sus datos, y marcando le fecha y hora de cierre o cerrar la tarea.

Otra de las tres agrupaciones que destaca es aquella con la que podemos subir y descargar archivos de nuestra página, pasando le la ruta del archivo y almacenando nuestro servidor los archivos que subimos con el método “postArchivos_Create” con el cual lo que hacemos es registrar en la base datos tanto en archivo sus datos como en “Usu_arch” que es archivo que subimos es perteneciente a ese usuario y almacenando lo en nuestro equipo en

“/app/public/Archivos” dentro de la carpeta de XAMPP en donde creamos nuestro proyecto de Laravel, otra de las funciones mas importantes es la realizada por nuestra funcion “getDowload”, con la cual descargamos los archivos del usuario que seleccionemos.

Y por último, la agrupación que me parece mas importante de todas ,no solo por mezclar las anteriores funcionalidades de una forma distinta en su interior sino porque permite que nuestros usuarios se agrupen y realicen cada uno sus entregas a un mismo sitio, siendo este grupos, en el cual además de crear grupos con “postGrupos_Create”, relacionamos a estos a los usuarios y les asignamos las tareas del grupo con “postAdd”, con el cual añadimos usuarios al grupo en cuestión y le relacionamos con las actividades que posee el grupo, y dependiendo de tu rol en el grupo, en la actividad se te permitirá realizar los mismo cambios ,como los que podrías realizar en tu actividad individual, en el caso de ser administrador, además de poder descargar todas las subidas de los otros usuarios, o solamente poder subir tareas si están abiertas en el caso de no serlo, y todo esto es gestionado por la función de “getTareasGrupoEspecificas”.

```

306 }
307
308 public function getTareasGrupoEspecificas($id,$cod)
309 {
310     $comprobar=Grup_act::where('cod_actividades',$cod)->where('cod_grupos',$id)->where('codigo_usuario',Auth::user()->id)->get();
311     $Archivos=Act_arch::where('cod_actividad',$cod)->get();
312     $lista_arch=array();
313     for($i=0;$i<count($Archivos);$i++)
314     {
315         $lista_arch[$i]=Archivos::where('cod_archivos',$Archivos[$i]['cod_archivos']->get());
316     }
317     if(isset($comprobar[0])){
318         $tareas=Actividades::where('cod_actividad',$cod)->get();
319         $date_now = time(); //tiempo actual
320         $date_convert = strtotime($tareas[0]['fecha_cierre']." ".$tareas[0]['hora_cierre']);
321         if($date_now < $date_convert || $tareas[0]['fecha_cierre'] == NULL && $tareas[0]['hora_cierre'] == NULL) {
322             $entregas=Act_arch::where('cod_actividad',$cod)->get();
323             $admin = Usu_gru::where('codigo_grupos',$id)->where('codigo_usuario',Auth::user()->id)->get();
324             return view('Agenda.tareas_grupo_especi', array('Tarea'=>$tareas,'Entregas'=>$entregas,'estado'=>1,'Archivos'=>$lista_arch,'admin'=>$admin));
325         }
326         else
327         {
328             $entregas=Act_arch::where('cod_actividad',$cod)->get();
329             $admin = Usu_gru::where('codigo_grupos',$id)->where('codigo_usuario',Auth::user()->id)->get();
330             return view('Agenda.tareas_grupo_especi', array('Tarea'=>$tareas,'Entregas'=>$entregas,'estado'=>0,'Archivos'=>$lista_arch,'admin'=>$admin));
331         }
332     }
333     else {
334         return redirect()->back();
335     }
336 }

```

Ilustración 3 Vista de getTareasGrupoEspecificas

6. Diccionario de Datos

1. Actividades

#	Nombre	Tipo	Atributos	Nulo	Clave
1	cod_actividad	bigint(20)	UNSIGNED	No	Primaria
2	nombre_actividad	text		No	
3	descripcion_actividad	text		No	
4	estado_actividades	tinyint(1)		No	
5	fecha_cierre	date		Si	
6	hora_cierre	time		Si	

Tabla 1 Diccionario Actividades

2. Archivos

#	Nombre	Tipo	Atributos	Nulo	Clave
1	cod_archivos	bigint(20)	UNSIGNED	No	Primaria

2	ruta_archivo	text		No	
---	---------------------	------	--	----	--

Tabla 2 Diccionario Archivos

3. Grupos

#	Nombre	Tipo	Atributos	Nulo	Clave
1	cod_grupo	bigint(20)	UNSIGNED	No	Primaria
2	nombre_grupo	text		No	
3	descripcion_grupo	text		No	

Tabla 3 Diccionario Grupos

4. Users

#	Nombre	Tipo	Atributos	Nulo	Clave
1	id	bigint(20)	UNSIGNED	No	Primaria
2	name	varchar(255)		No	
3	email	varchar(255)		No	Indice
4	password	varchar(255)		No	

Tabla 4 Diccionario Users

5. Act_Arch

#	Nombre	Tipo	Atributos	Nulo	Clave
1	cod_act_arch	bigint(20)	UNSIGNED	No	Primaria
2	cod_actividad	bigint(20)	UNSIGNED	No	Foranea
3	cod_archivos	bigint(20)	UNSIGNED	No	Foranea

Tabla 5 Diccionario Act_Arch

6. Grup_Act

#	Nombre	Tipo	Atributos	Nulo	Clave
1	cod_grup_act	bigint(20)	UNSIGNED	No	Primaria
2	cod_grupos	bigint(20)	UNSIGNED	No	Foranea
3	cod_actividades	bigint(20)	UNSIGNED	No	Foranea
4	codigo_usuario	bigint(20)	UNSIGNED	No	Foranea

Tabla 6 Diccionario Grup_Act

7. Usu_Act

#	Nombre	Tipo	Atributos	Nulo	Clave
1	cod_usu_act	bigint(20)	UNSIGNED	No	Primaria
2	codigo_usuario	bigint(20)	UNSIGNED	No	Foranea
3	cod_actividad	bigint(20)	UNSIGNED	No	Foranea

Tabla 7 Diccionario Usu_Act

8. Usu_Arch

#	Nombre	Tipo	Atributos	Nulo	Clave
1	cod_usu_arch	bigint(20)	UNSIGNED	No	Primaria
2	codigo_usuario	bigint(20)	UNSIGNED	No	Foranea
3	cod_archivos	bigint(20)	UNSIGNED	No	Foranea

Tabla 8 Diccionario Usu_Arch

9. Usu_Gru

#	Nombre	Tipo	Atributos	Nulo	Clave
1	cod_usu_gru	bigint(20)	UNSIGNED	No	Primaria
2	codigo_usuario	bigint(20)	UNSIGNED	No	Foranea
3	codigo_grupos	bigint(20)	UNSIGNED	No	Foranea
4	administrador	tinyint(1)		No	

Tabla 9 Diccionario Usu_Gru

10. Despliegue de la aplicación

10.1. Instalación

La instalación con el paquete “.rar” es sencilla, consiste en inicio en instalar XAMPP y configurar Apache y MySQL, tras esto instalaremos Composer, iniciaras Apache y MySQL y abriendo la consola de comandos, ya sea con “Windows + R” o a través del buscador abrimos el “CMD”, con el abierto, nos aseguraremos de tener una carpeta propia en “C:/xampp/htdocs/” y tras posicionarnos en nuestra carpeta con el “CMD” lanzaremos el comando “composer global require laravel/installer” y tras la instalación lanzaremos “laravel new ejemplo-app” en la carpeta, con esto habremos creado nuestro proyecto en Laravel, con lo que podremos en primer lugar realizar la instalación de Breeze, para crear el sistema de Autenticación y que se creen sus pantallas y una parte de las migraciones, a estas, les sumaremos las que están en nuestro comprimido, también agregaremos los modelos y el controlador, y por último añadiremos nuestras páginas, con esto ya agregado, modificaremos el “.env” para que apunte nuestra base de datos y le asignamos el nombre que le queramos poner y el usuario con el que lo que ramos lanzar todo, con esto ya hecho lanzaremos las migraciones y si todo ha salido bien al intentar acceder ya debería de estar completamente operativa.

10.2. Administración y Configuración

Uno de los puntos más importantes de la administración será cada cierto tiempo realizar limpiezas de los archivos almacenados en el equipo, ya que los usuarios no pueden borrarlos y estos no tienen un límite de espacio, además de esto hay algunos cambios en la configuración que se podrían realizar como por ejemplo que las tareas por defecto se creen abiertas, que en el caso actual no es así, al considerar que al crear un tarea, no es requerida para ese mismo momento y por lo tanto no es preciso que nadie realice la entrega en el mismo instante de su creación.

11. Índice de figuras (imágenes / tablas)

Ilustración 1 Correcto estado del XAMPP-----	4
Ilustración 2 Vista de las Tablas de la Base de Datos-----	6
Ilustración 3 Vista de getTareasGrupoEspecific-----	7
Tabla 1 Diccionario Actividades _____	7
Tabla 2 Diccionario Archivos _____	8
Tabla 3 Diccionario Grupos _____	8
Tabla 4 Diccionario Users _____	8
Tabla 5 Diccionario Act_Arch _____	8
Tabla 6 Diccionario Grup_Act _____	8
Tabla 7 Diccionario Usu_Act _____	8
Tabla 8 Diccionario Usu_Arch _____	8
Tabla 9 Diccionario Usu_Gru _____	9

12. Siglas y acrónimos

Siglas	
XAMPP	XAMPP es un paquete de software, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes PHP y Perl. El nombre es en realidad un acrónimo: X, queriendo decir que sirve para cualquier sistema operativo, Apache, MariaDB/MySQL, PHP, Perl

13. Bibliografía

Redacción de referencias en IEEE.

- [1] Web : <https://laravel.com/docs/10.x/starter-kits>
- [2] Web : <https://cynoteck.com/es/blog-post/installing-laravel-8-on-windows-10-xampp/>
- [3] Web: <https://politicadedatos.cdmx.gob.mx/cultura/quias/diccionario#:~:text=Un%20diccionario%20de%20datos%20es,datos%20y%20sus%20posibles%20usuarios.>
- [4] Web : <https://www.ibm.com/docs/es/spss-statistics/29.0.0?topic=variables-system>