



CENTRO INTEGRADO de FORMACIÓN PROFESIONAL
AVILÉS

CFGS: DESARROLLO DE APLICACIONES WEB

Modalidad: Presencial

Curso: 2022-2023

Agenda Web y Gestión

MEMORIA

Avilés, junio 2023

Autor: García Martínez, Luis Miguel

Tutor/a Individual: Iglesias, Yolanda

Tutor/a Colectivo: Iglesias, Yolanda

1. Índice

MEMORIA	1
1. Índice	2
2. Introducción	3
3. Objetivos del proyecto	3
3.1. Objetivo general del proyecto	3
3.2. Objetivos específicos	3
4. Análisis y Diseño	5
4.1. Requisitos funcionales	5
4.2. Casos de Uso	5
4.3. Modelo de datos (Diagrama E/R)	6
5. Implementación	8
5.1. Tecnologías/Herramientas empleadas	8
5.2. Descripción	9
6. Planificación y presupuesto	12
6.1. Diagrama de Gantt de planificación de tareas	12
6.2. Presupuesto	12
7. Futuras mejoras	14
8. Conclusiones sobre el proyecto.	14
9. Índice de figuras (imágenes / tablas)	15
10. Siglas y acrónimos	15
11. Bibliografía	15

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 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. Objetivos del proyecto

3.1. Objetivo general del proyecto

El objetivo es la correcta creación de una pagina web que pueda servir como una Agenda Web, en la cual se puedan crear tareas, con el añadido de la creación de grupos de trabajo, en los cuales poder entregar las tareas que solicitemos y/o nos soliciten, en además de esto, también se desea poder subir archivos, ya sea para una tarea, o para almacenarlos en la web para si mismos, los cuales posteriormente podrán descargar.

3.2. Objetivos específicos

Los objetivos específicos del proyecto son:

- La creación y gestión de usuarios:

Este es uno de los puntos indispensables de la aplicación, se espera que a través de la misma, se puedan crear, borrar y modificar usuarios, este es un punto de suma importancia, ya que las tareas serán asignadas a los usuarios, los grupos estarán compuestos de usuarios y los archivos son subidos por usuarios, además se espera que los usuarios puedan cambiar los datos de su cuenta como pueden ser el correo o el nombre del usuario.

- La creación y Gestión de tareas:

El principal objetivo de la aplicación es la creación de tareas, las cuales podamos gestionar, ya sea cerrándolas, borrándolas, o modificándolas, además de en el caso de los grupos limitar todas estas acciones solo a los administradores del grupo, ya que se entiende que tu eres el administrador de tus propias tareas individuales, además se realiza controles de entrega, a través de que la actividad este habilitada para ello y, si es que tiene fecha y/o hora fijada, te permita o no realizar las entregas dependiendo de la misma.

- La creación y Gestión de Grupos:

Otro de los puntos principales es la creación de grupos y su gestión, incluyendo esto, agregar usuarios, borrarlos del grupo y el borrado del grupo al completo, limitando estas cosas a los administradores y permitiendo a los usuarios no administradores únicamente salir de los grupos, en los grupos, es donde se crean, se entregan y se gestionan las actividades grupales, por lo que es un punto central de la aplicación.

- La subida y bajada de Archivos:

Otro punto indispensable es la subida y descarga de archivos, lo cual es necesaria tanto para las tareas, como para la subida de archivos para el propio usuario de forma personal, permitiendo a todos los usuarios subir a su propio espacio archivos los cuales podrán descargar posteriormente, y en el caso de las tareas, siempre que este permitido, subir archivos los cuales los administradores del grupo podrán descargar.

4. Análisis y Diseño

4.1. Requisitos funcionales

Se requiere realizar una aplicación para la asignación y gestión de tareas a través de una web, la cual se basa en que los usuarios ya sea formando grupos o ellos de forma individual, generen tareas, ya sea con fecha y hora o sin ella y entreguen tareas, las cuales se entregan enviando archivos, que deben ser accesibles, en el caso de los grupos por los administradores de estos, los cuales también pueden gestionar el mismo grupo.

Por lo tanto la aplicación ha de permitir:

- La creación y registro de usuarios.
- La creación, gestión y entrega de tareas individuales.
- La subida de archivos ya sea a tareas o para el propio usuario.
- La descarga de los archivos que sean subidos.
- La creación y gestión de grupos.
- La creación de tareas para estos mismos.
- El acceso por los administradores del grupo a las tareas enviadas por los miembros.

Para el acceso a la aplicación lo primero que necesitaremos será que los usuarios tengan una cuenta, por lo que necesitaremos un sistema de autenticación, para esto hemos de permitir que se puedan registrar, y tras esto podrán iniciar sesión con la cuenta que se creó.

Los datos que almacenaremos por usuarios serán:

- Un Nombre, con el cual nos referiremos a él en el interior de la aplicación
- Su Email, el cual junto con la contraseña serán los campos que solicitaremos para el inicio de sesión después del registro
- Una contraseña, la cual cifraremos por motivos de seguridad

Para la creación de los grupos lo que necesitaremos será relacionar los grupos con sus miembros, y para esto guardaremos un nombre y una descripción para los grupos.

Para las tareas, ya sea grupales o individuales guardaremos un nombre de la tarea, una descripción de esta, un estado, con el cual podremos marcar si queremos que realicen entregas a la tarea, y limitando así las entregas a que la actividad esté abierta, y una fecha y una hora para marcar fechas límites de entrega, permitiendo por ejemplo que los usuarios puedan entregar hasta cierta hora o hasta cierto día.

Y por último para los archivos guardaremos un nombre, el cual será el nombre tanto del archivo como el nombre por el que guardaremos y descargaremos los archivos que subamos.

4.2. Casos de Uso

Para mostrar cuáles van a ser las funcionalidades de la aplicación podemos observar en *Ilustración 1 Casos de Uso de la aplicación* un diagrama de las posibles interacciones del usuario con nuestra aplicación.

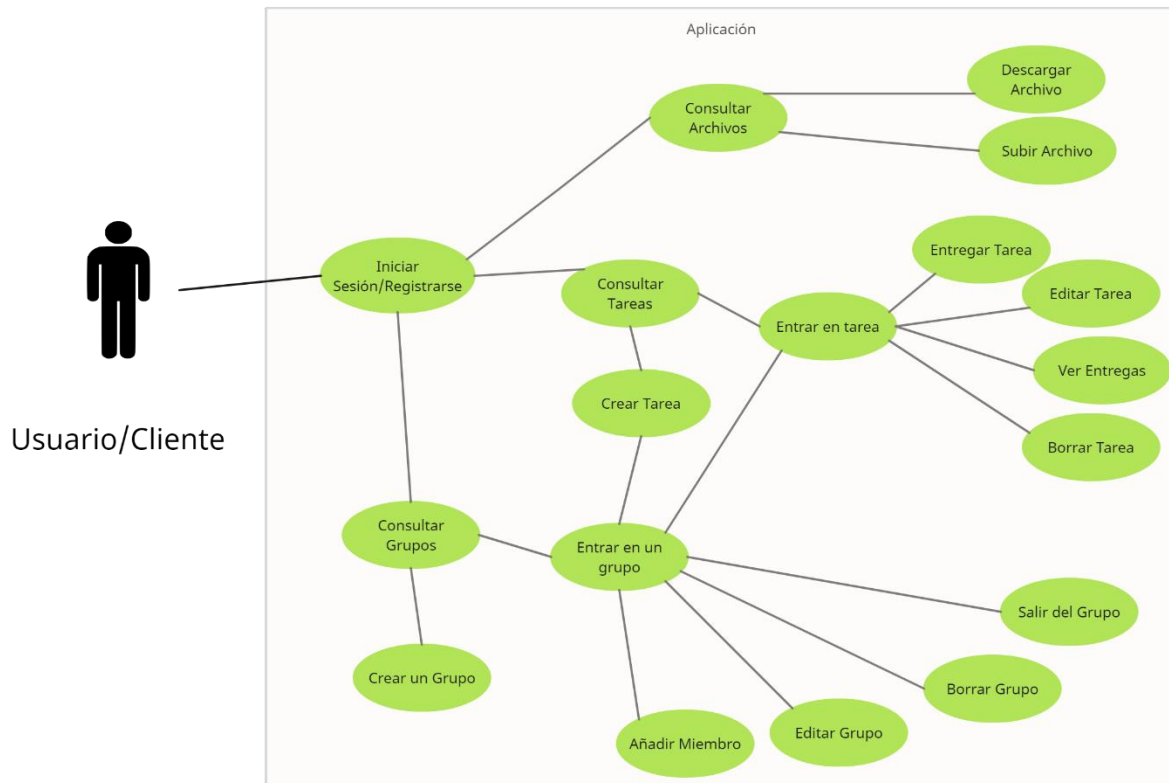


Ilustración 1 Casos de Uso de la aplicación

Como vimos en la ilustración anterior, el usuario tras iniciar sesión o registrarse tiene varios caminos, el primero de ellos entrar en Archivos, donde, en caso de que anterior mente hubiese subido archivos, ya fuese a una actividad o en archivos, podría descargar el archivo, o subir uno, tras esto en la ilustración podemos ver que el segundo camino sería “Consultar Tareas”, donde iríamos al apartado de Tareas y veríamos las tareas individuales que nos hemos fijado o podríamos crear una nueva tarea, al entrar en una de las tareas que hubiéramos creado anteriormente podremos subir un archivo como entrega, editar la tarea, cambiando todo lo que la define, ver las entregas que hemos realizado y borrar la tarea al completo, lo cual no conlleva el borrado de los archivos que subimos, por ultimo tenemos los grupos, donde podríamos crear uno o entrar a uno ya existente, en este dependiendo de si hemos sido fijado como administrador o no podremos salir del grupo, en caso de no ser administrador, o añadir miembros, editar el grupo , crear una tarea del grupo o bórralo en caso de serlo, pero en ambos casos lo que podremos es entrar a las tareas del grupo y dentro de la misma, dependiendo nuevamente de si somos administradores del grupo, podremos editar la tarea, ver las entregas o borrar la tarea en caso de ser administradores, o entregarla en caso de que este abierta si somos administradores o usuarios no Administradores.

4.3. Modelo de datos (Diagrama E/R)

En cuanto al modelo de datos, se puede ver que este compuesto por 4 entidades principales muy relacionadas entre sí, siendo el diagrama el siguiente:

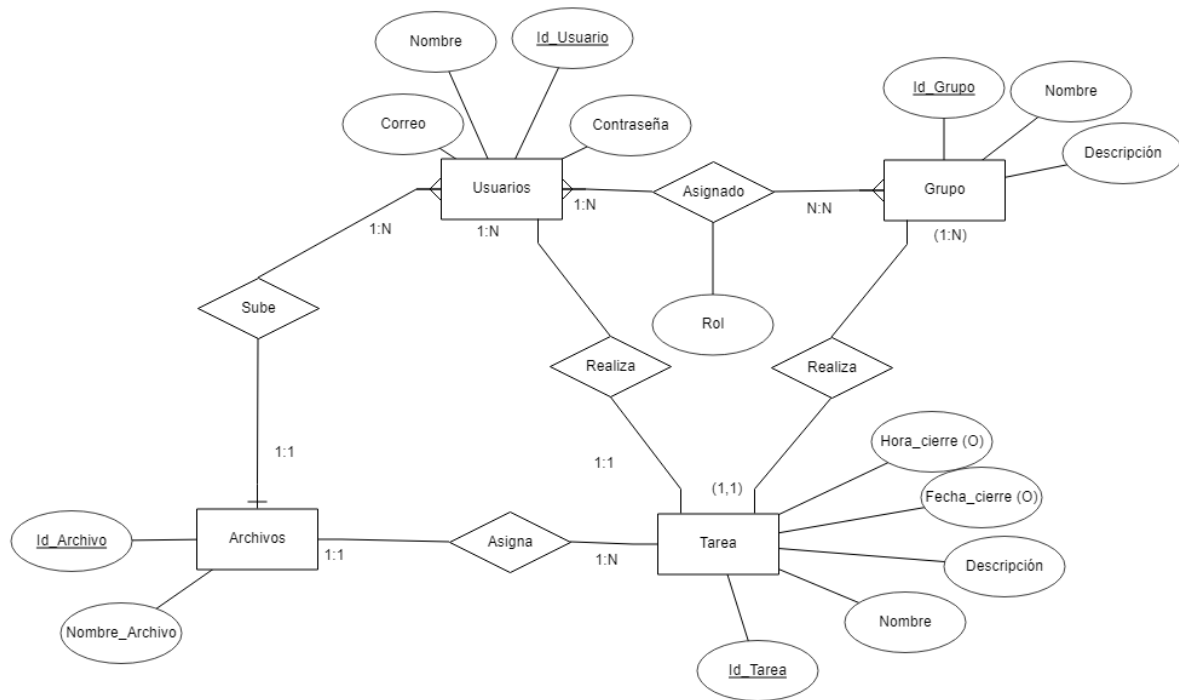


Ilustración 2 Diagrama del Modelo E/R

La entidad “Usuarios”, es la base de la aplicación, y los atributos “Id_Usuario” como clave primaria y “Nombre”, “Correo” y “Contraseña”, esta esta relacionada con “Grupo”, en cuya relación surge el atributo de “Rol”, con el cual marcaremos el rol de un “Usuario” en el “Grupo”, en la entidad “Grupo” encontramos como clave primaria “Id_Grupos”, y los atributos “Nombre” y “Descripción”, tanto “Usuarios” como “Grupos” están relacionados de la misma manera con “Tareas”, la cual como clave primaria tiene “Id_Tarea” y como atributos “Nombre” y “Descripción” y como atributos tambien, pero estos siendo opcionales, “Fecha_cierre” y “Hora_cierre”, como entidad tambien relacionada con “Usuarios” y esta vez con “Tarea” encontramos “Archivos”, el cual como clave primaria tiene “Id_Archivo” y como atributo tambien tiene “Nombre_Archivo”.

El anterior diagrama E/R, se tradujo a la base de datos como, la *Ilustración 3 Diseñador de la Base de Datos*, en la cual se puede ver como estas relaciones causaron la aparición de varias tablas para relacionar las entidades iniciales, surgiendo por ejemplo de la relación entre la entidad “Usuarios” y la entidad “Grupos” la tabla “Usu_gru”, en la cual se guarda el anterior atributo de “Rol”, convertido ahora en el campo “Administrador”.

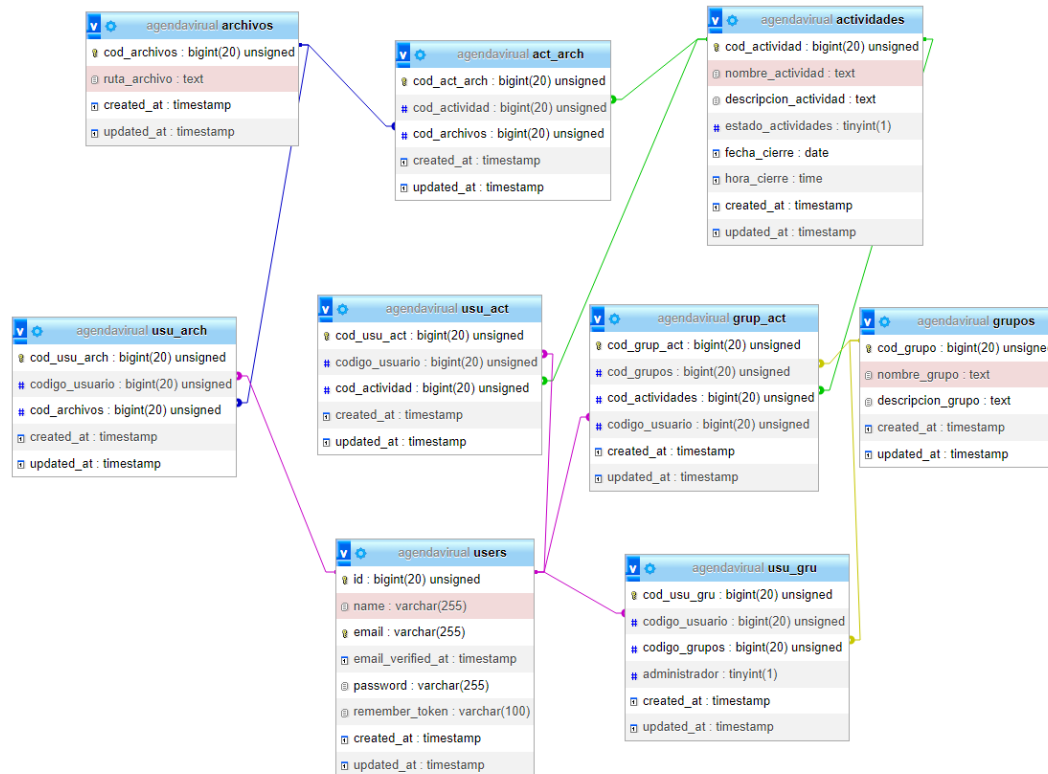


Ilustración 3 Diseñador de la Base de Datos

5. Implementación

5.1. Tecnologías/Herramientas empleadas

Las principales herramientas han sido Laravel, Composer, Bootstrap, PhpMyAdmin, Apache y PHP.

- **Laravel:**
Laravel es un framework de PHP, a través del cual se facilita el desarrollo de las aplicaciones web al centralizar los recursos y asignarles un orden, este se distingue por el uso del modelo vista controlador, en el cual hay una parte que realiza las operaciones, el controlador, y otro que muestra la página, la vista, a través de esto puedes además de centralizar todas las operaciones aumentar la seguridad de la página al no ser accesible directamente el código de las operaciones al usuario.
- **Composer:**
Composer es un sistema de gestión de paquetes para programar en PHP, a través del cual se facilita el manejo de dependencias y librerías del propio PHP, este trabaja en conjunto con Laravel para la realización de varias descargas y extensiones.
- **Bootstrap:**
Bootstrap es una herramienta que sirve para facilitar el diseño de una página web, al recoger en clases muchos diseños pre hechos, los cuales en conjunto con su extensión para Visual Studio Code, hace muy sencillo el diseño, al tener ya registrados muchos de estos ya hechos.
- **PhpMyAdmin:**

PhpMyAdmin es un gestor de base de datos, el cual ya viene instalado con XAMPP, a través de este podremos gestionar nuestra base de datos, este trabaja con el lenguaje SQL como principal lenguaje a la hora de interactuar.

- Apache
Apache es un servidor web, con el cual nos encargamos del almacenamiento, procesamiento y muestra de la web realizada, este junto con PhpMySQL viene instalado por defecto en XAMPP.
- PHP:
PHP es un lenguaje de programación de entorno servidor, el cual es la base de Laravel, y junto con HTML y CSS es lo que constituye todas nuestras páginas

5.2. Descripción

- Creación del Proyecto:
Lo primero que se realizó fue la creación de un proyecto en Laravel y la modificación del “.env” para marcar el nombre de nuestra base de datos y el acceso a la misma, y la instalación del módulo de Breeze, el cual importa y pone en funcionamiento un simple pero efectivo sistema de autenticación y registros de usuarios, este además nos crea rutas y páginas, además de tablas en la base de datos para usuarios.
- Las migraciones:
Tras la creación del diagrama y su normalización se procedió a la creación de las migraciones para crear así la base de datos, creando 3 tablas principales, “Grupos”, “Actividades” y “Archivos”, ya que la tabla de “Users” que nos ha creado Breeze incluye todos los campos que nosotros íbamos a introducir, por lo que usaremos su tabla, tras ellos cree tablas para relacionar las 4 tablas principales, siendo las tablas, la unión entre “Actividades” y “Archivos” llamada “Act_Arch”, la unión entre “Grupos” y “Actividades” la llamamos “Grup_Act”, la unión entre “Users” y “Actividades” fue llamada “Usu_Act”, la unión entre “Users” y “Archivos” fue llamada “Usu_Arch”, y por último “Users” y “Grupos” fue llamada “Usu_Gru”.

Los campos de las tablas fueron, como se puede ver en *Ilustración 3 Diseñador de la Base de Datos*, fueron:

1. Actividades
 - a. Cod_Actividad: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Nombre_Actividad
 - c. Descripción_Actividad
 - d. Estado_Actividad
 - e. Fecha_Cierre: Puede ser fijada como nula.
 - f. Hora_Cierre: Puede ser fijada como nula.
2. Archivos
 - a. Cod_Archivos: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Ruta_Archivos
3. Grupos
 - a. Cod_Grupo: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Nombre_Grupo
 - c. Descripción_Grupo
4. Users

- a. Id: Es la clave Primaria de la tabla, y es autoincrementar.
 - b. Name
 - c. Email
 - d. Password
- 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"
 - c. Cod_Archivos: Clave foránea, referencia la clave primaria de "Archivos"
- 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"
 - d. Codigo_Usuario: Clave foránea, referencia la clave primaria de "Usuarios"
- 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"
 - c. Cod_Actividad: Clave foránea, referencia la clave primaria de "Actividad"
- 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
 - c. Cod_Archivos: Clave foránea, referencia la clave primaria de
- 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"
 - c. Codigo_Grupos: Clave foránea, referencia la clave primaria de "Grupos"
 - d. Administrador
- Las Rutas:

Se fijaron las rutas a utilizar en "web.php", además de marcar que paginas requerían un Inicio de Sesión previo para ser vistas, además de modificar los registros básicos de "/" y "/dashboard" para que apuntasen a "/inicio".
- Los Modelos:

Se crearon los Models de cada tabla de la base de datos, comprobando su funcionalidad introduciendo datos de prueba atreves de estos, tras comprobar su funcionalidad, se procedió a relacionar el controlador con estos Models.
- El Controlador y las Paginas:
 - Layout: Se modifiko el Layout de Laravel, para que tenga enlaces a todas las paginas de la web y aparezca la opción de inicio de sesión o de registro en caso de que no se esté con la sesión iniciada, o para que se muestre le nombre de usuario con un desplegable en el mismo, desde el cual se podrá acceder al perfil o salir de la sesión, además de que se muestren los links a las otras páginas.

- Inicio: Se designo la página “inicio.blade.php” como página de “/inicio”, la cual es la única que puede ser visible antes de Iniciar Sesión, y se comprobó, tanto que es accesible por usuarios que iniciaron sesión. como por usuarios que no.

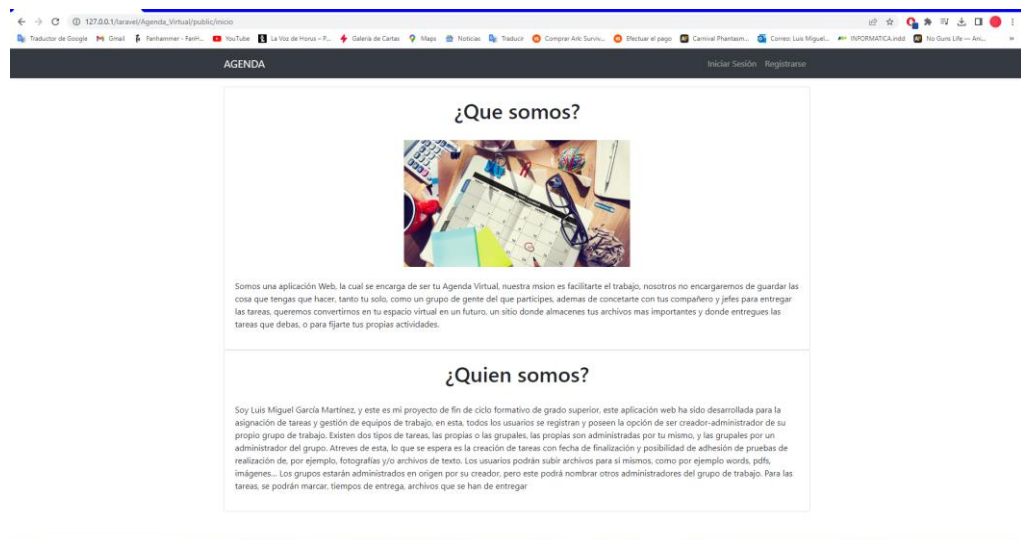


Ilustración 4 Pagina de inicio

- Perfil: Se crearon los controladores para la pagina de perfil, tanto uno para modificarla, como otro para verla.
- Grupos:
Se crearon, primero una vista que te permitía ver todos los grupos en los que un usuario esta añadido y otra pagina para crear grupos, tras eso se desarrollo un vista para el interior de cada grupo, en la cual la gente marcada como administradora puede añadir y quitar gente y borrar el grupo, y para la gente que no esta designada como administradora, se les permite abandonar el grupo, luego se desarrollo la creación de actividades grupales, a las cuales se les marco el acceso atreves del grupo, y dentro, al igual que para la gestión de los grupos, dependiendo de si era administrador o no, se le permite modificar la actividad, borrar la , o ver y descargar las entregas realizadas, en el controlador de esta pagina tambien se extrae si se esta a tiempo para subir la tarea o no.
- Tareas:
En tareas, se desarrollo una función con la cual crear nuevos grupos, y se marco la pagina de acceso junto con los datos a mostrar en ella, siendo muy similar a la de grupos, pero en esta al ser del propio usuario no fue necesario delimitar las opciones de administración.
- Archivos:
Se desarrollo un método para el almacenamiento y descarga de archivos a Laravel, además de permitir ver y descargar a los usuarios los archivos que cargo a cualquier parte, ya sea una actividad o al propio apartado de archivos.

6. Planificación y presupuesto

6.1. Diagrama de Gantt de planificación de tareas

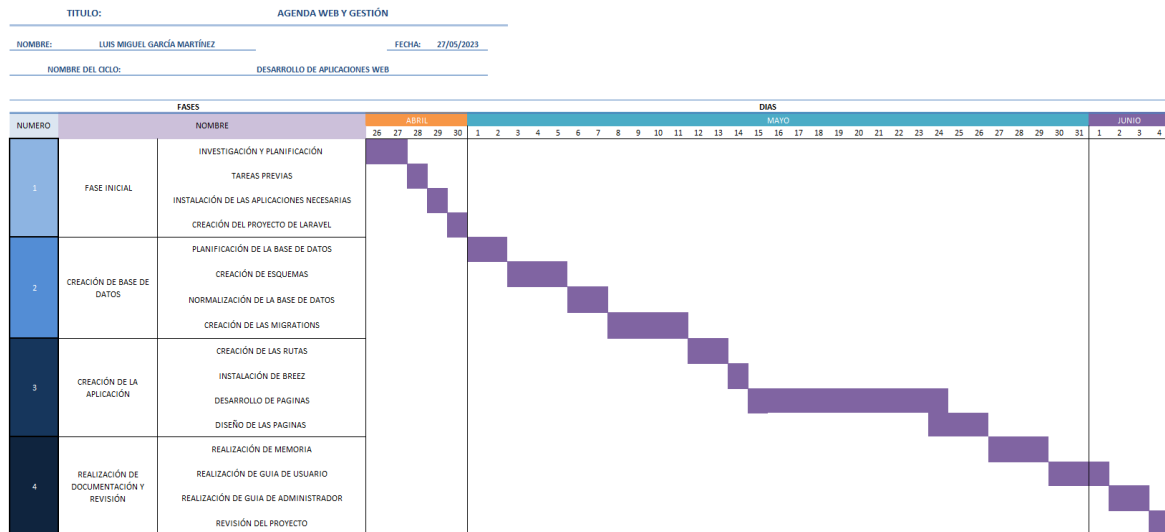


Ilustración 5 Diagrama de Gantt

Como se puede observar en el diagrama, el desarrollo del proyecto se dividió en 4 partes, una fase inicial, en la cual se busco información y se descargaron las aplicaciones que van a ser necesarias, como pude ser Composer, tras ello , se planifico la basa de datos, realizando el Diagrama E\R y tras ellos se procedió a su normalización, tras esto, se creo el proyecto de Laravel y se crearon las migraciones, con las cuales se compuso la base de datos, después de esto, comenzó la tercera fase, en la cual instalamos Breeze y lanzamos las migraciones para tener la base de datos ya operativa, se designaron las rutas que serian utilizadas y se marcaron en el "web.php", en esta fase lo único que mostraban era un mensaje para demostrar su operatividad, tras lo cual se procedió al desarrollo de las paginas y sus controladores, empezando por Archivos, donde permitimos la subida y bajada de archivos, y al ya tener desarrollada la subida y bajada de archivos, pasamos a las tareas, donde a la hora de subir una tarea reutilizamos el método para subir archivos, al acabar esto realizamos los grupos, y al acabar con estos, procedimos a realizar el diseño de las vistas de la página, donde elegí usar principalmente el blanco y el gris para las paginas y en azul en el caso de la mayoría de los botones que realizan una acción, como subir un archivo, al acabar con el diseño, se procedió con la redacción de los archivos y la revisión de las paginas.

6.2. Presupuesto

- **Coste de Programador**

El servicio tendrá un coste de 70€ por día de trabajo, calculado una duración aproximada de 2 semanas, el coste en mano de obra rondaría los 700€, incluyendo el desarrollo de la base de datos , el desarrollo y la puesta en marcha.

- **Coste de hardware**

- **Hardware**

Tabla 1 Precio Hardware

Hardware	Número de Unidades	Precios/Unidad	Precio Total
HP M01-F2042ns Intel Core i5	1	675,10€	675,10€

- **Software**

Tabla 2 Precio Software

Software	Numero de Claves	Precio/Unidad	Precio Total
Windows 10 Pro	1	259 €	259 €

- **Coste de total**

Tabla 3 Coste Total

Coste Total	
Coste Hardware	675,10€
Mano de Obra	700€
Coste Software	259 €
Coste Total (sin IVA)	1.634,10€
Coste Total (con IVA)	326,82€
Coste Total	1.960,92€

7. Futuras mejoras

El programa, pese a ser muy completo, podría expandirse, y algunas de las adhesiones que realizaría, serían:

- Envío de correo al realizar el registro de un usuario, en el cual hayas de entrar para así activar la cuenta, por ejemplo a través de un link, con esto al menos podríamos tener la seguridad de que el correo existe.
- Borrado de Archivos, otro añadido que realizaría sería permitir a los usuarios borrar sus archivos individuales, y a los administradores de las tareas, el borrado de los archivos entregados, tanto para la actividad como para el usuario.
- Limitación de números de archivos a entregar, otro añadido que realizaría, sería permitir a los administradores limitar el número de archivos que un usuario puede entregar a una actividad.
- Un chat para el grupo, otra funcionalidad que sería interesante añadir, sería añadir un chat del grupo, en el cual los usuarios puedan hablar, y así por ejemplo, que un usuario por ejemplo pueda avisar a un administrador que la actividad está cerrada.
- Servidor FTP, otro añadido interesante, sería crear un servidor ftp, en el cual guardaríamos todos los archivos de los usuarios, sería conveniente, que este se encontrara en un equipo diferente, con lo que además de reducir el consumo de espacio de los discos del servidor, aumentaríamos el rendimiento y la organización de los archivos.
- Lista de “amigos”, otro añadido interesante, sería la creación de una lista, donde poder guardar los usuarios y sus códigos, que más usamos, o con los que más relacionados estamos, facilitando así para los usuarios añadir los miembros con los que más se relaciona a sus grupos.

8. Conclusiones sobre el proyecto.

Mis conclusiones son, que es un desarrollo interesante, que plantea sus dificultades, por ejemplo en la relación entre Laravel y la base de datos, al utilizar Laravel sus propias formulas, las cuales, pese a ser correctas algunas de las que se realizan no funcionan y has de buscar el método exacto que es dictado por Laravel.

Otro punto que me pareció interesante es la relación entre las tablas, por ejemplo, la de “Grup_Act”, donde a la hora de relacionar los grupos con las Actividades podemos ver que al querer asignarle las actividades a los miembros de los grupos, y no a los grupos como tal, hemos de añadir la clave foránea del Id de “Users”, haciendo así una asignación directa los miembros de los grupos y no directamente a los grupos como tal.

Otra característica que despertó mi interés, fue la de cargar datos, ya que permite muchísimas posibilidades, desde hacer que sea privado, hasta externalizar lo, refiriendo me a la capacidad de asignar esta característica a otra máquina, también te permite diferentes niveles de protección para los archivos y marcar las carpetas donde deseamos que se almacenen los datos, por ejemplo para las actividades de grupo, lo que hacemos es almacenar, tanto en la carpeta personal del usuario que la ha subido, como en una carpeta de la actividad, para que así los usuarios tengan su propia copia del archivo que subieron y los administradores otra.

Y por último, me pareció interesante también es el control de datos y su filtro desde el controlador, para así limitar el acceso a ciertas acciones únicamente a los usuarios que deseamos, un ejemplo de ello es en las tareas de grupo, donde permitiremos a los usuarios hacer o no hacer dependiendo de los permisos que se les asignaron en la tabla que relaciona “Grupos” y “Usuarios”.

9. Índice de figuras (imágenes / tablas)

Ilustración 1 Casos de Uso de la aplicación	6
Ilustración 2 Diagrama del Modelo E/R	7
Ilustración 3 Diseñador de la Base de Datos	8
Ilustración 4 Pagina de inicio	11
Ilustración 5 Diagrama de Grantt	12
Tabla 1 Precio Hardware	13
Tabla 2 Precio Software	13
Tabla 3 Coste Total	13

10. Siglas y acrónimos

Nombre	Descripción
FTP	Un protocolo de transferencia de archivos, File Transfer Protocol, que se traduce como Protocolo de Transferencia de Archivos.
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

11. Bibliografía

Redacción de referencias en IEEE.

- [1] Web : <https://getbootstrap.com/docs/4.0/components/modal/>
- [2] Web : <https://laravel.com/docs/10.x/migrations>
- [3] Web : <https://laravel.com/docs/10.x/filesystem>
- [4] Web : <https://creately.com/blog/es/diagramas/tutorial-diagrama-caso-de-uso/#:~:text=El%20diagrama%20de%20caso%20de,roles%20interact%C3%BAan%20con%20el%20sistema.>

- [5] Web : <https://stackoverflow.com/questions/36883728/save-vs-update-in-laravel>
- [6] Web : <https://stackoverflow.com/questions/29594572/laravel-eloquent-update-record-without-loading-from-database>
- [7] Web : <https://stackoverflow.com/questions/45458074/laravel-eloquent-orm-delete-method>