

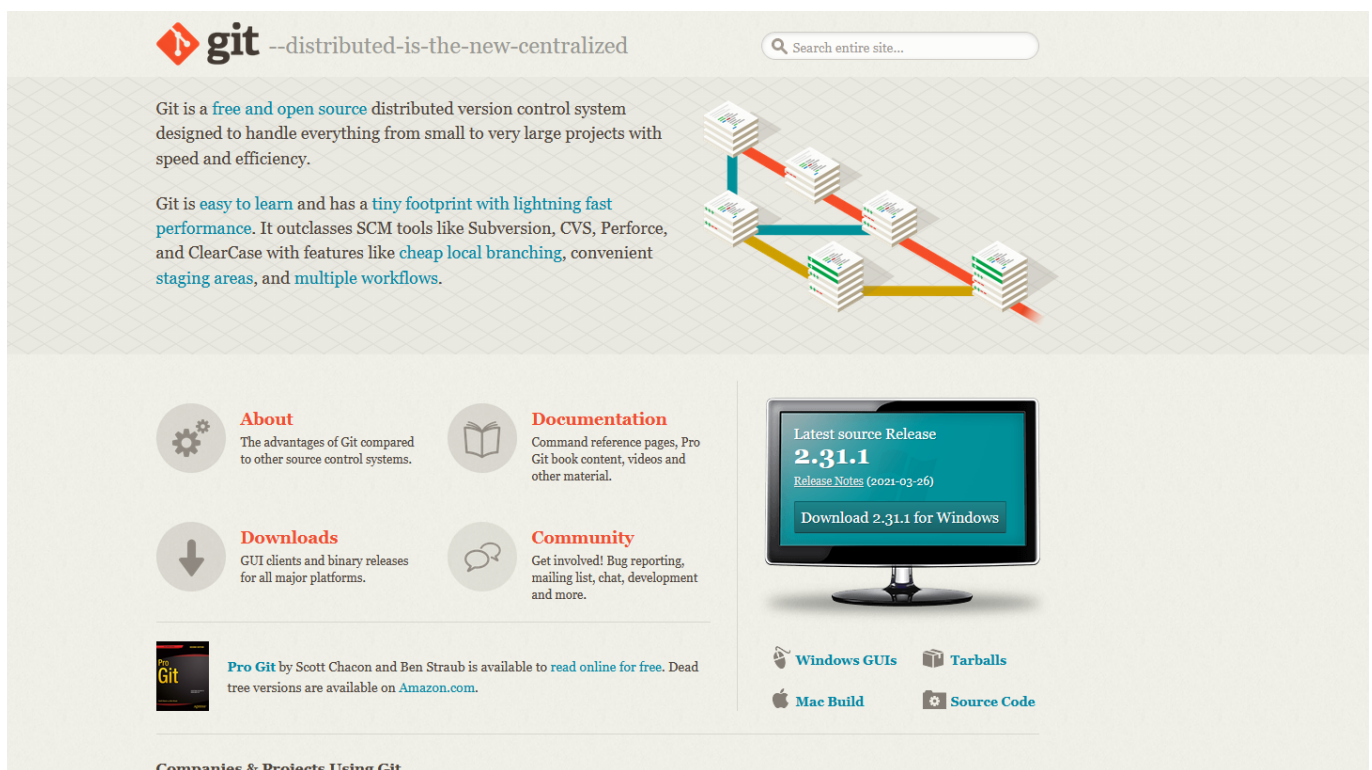
# Explicación de las tecnologías

## Git

Gracias a git contamos con las siguientes características que nos ayudan a mejorar nuestra forma de programar. Para ello nos ofrece las siguientes ventajas:

- Sistema de versiones, que nos permite llevar un gestión de los cambios realizados en nuestros códigos.
- Nos permite trabajar en equipo de manera sencilla. Gracias a un sistema de Merges.
- También nos permite volver a las versiones anteriores de nuestro código.

Para usar git nos vamos a su página web y nos descargamos el archivo de instalación que encontramos para windows.



En caso de usar Linux podemos instalarlo con los siguientes comandos `Sudo apt install git` o `sudo pacman install git`. En Mac `yum install git`

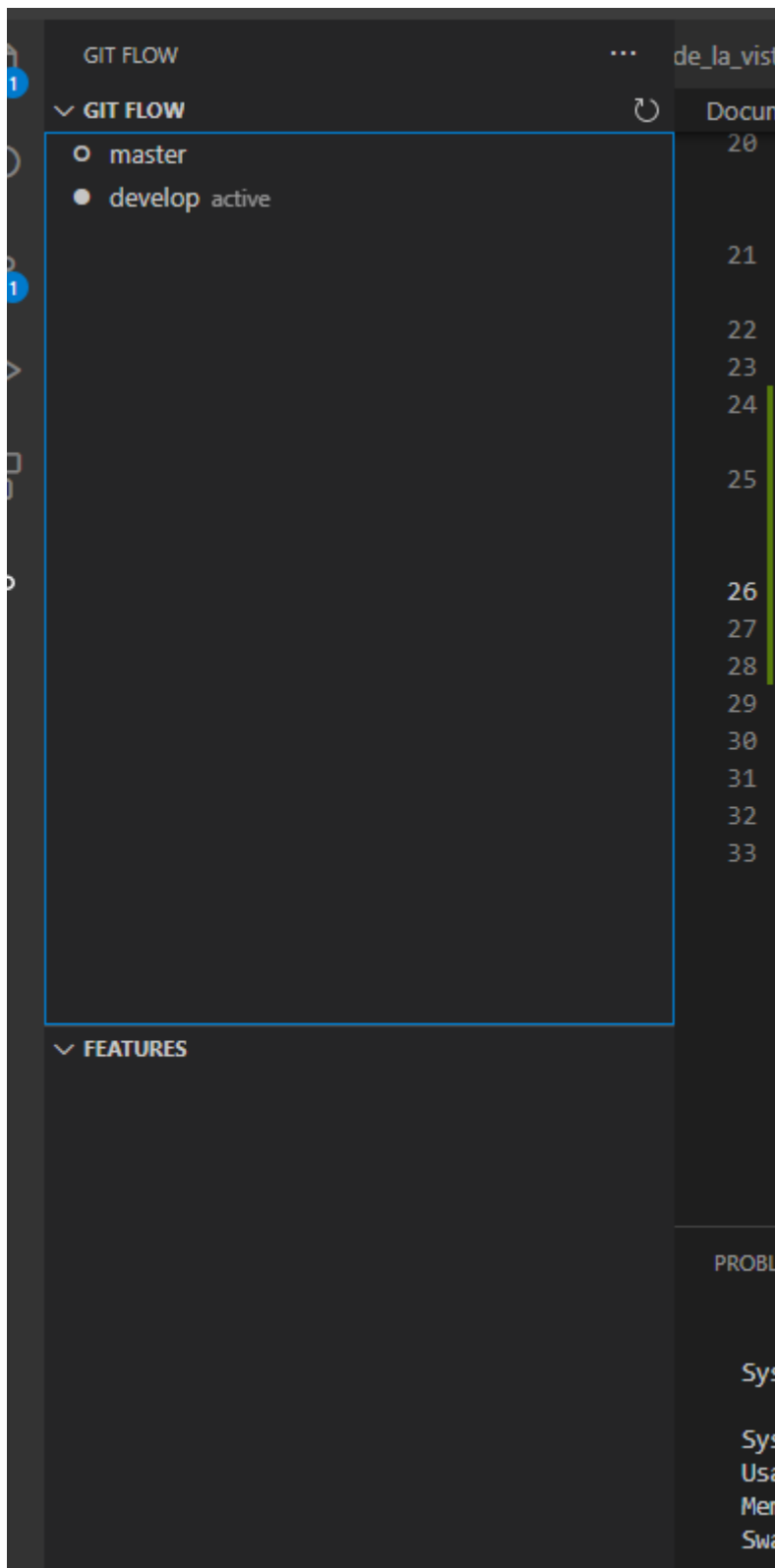
## Git-Flow

En cuanto a git flow es un agregado extra a las funciones git, nos mejora el flujo de trabajo. Para ello creamos una serie de ramas extras que nos permitirán mejorar la gestión de nuestros proyectos y no pisarnos el trabajo entre varias personas. Para ello cogemos la siguiente forma de actuar:

- Master y develop son intocables. 3 ramas nuevas de trabajo feature, hotfix y release.
- Explicaré el funcionamiento de estas ramas.
- Feature es una copia de develop, cómo no se trabaja directamente con esta da mas estabilidad a nuestro código. Cuando se termina una feature esta se mergea al develop. Pudiendo volver a una versión anterior de manera mucho más sencilla.

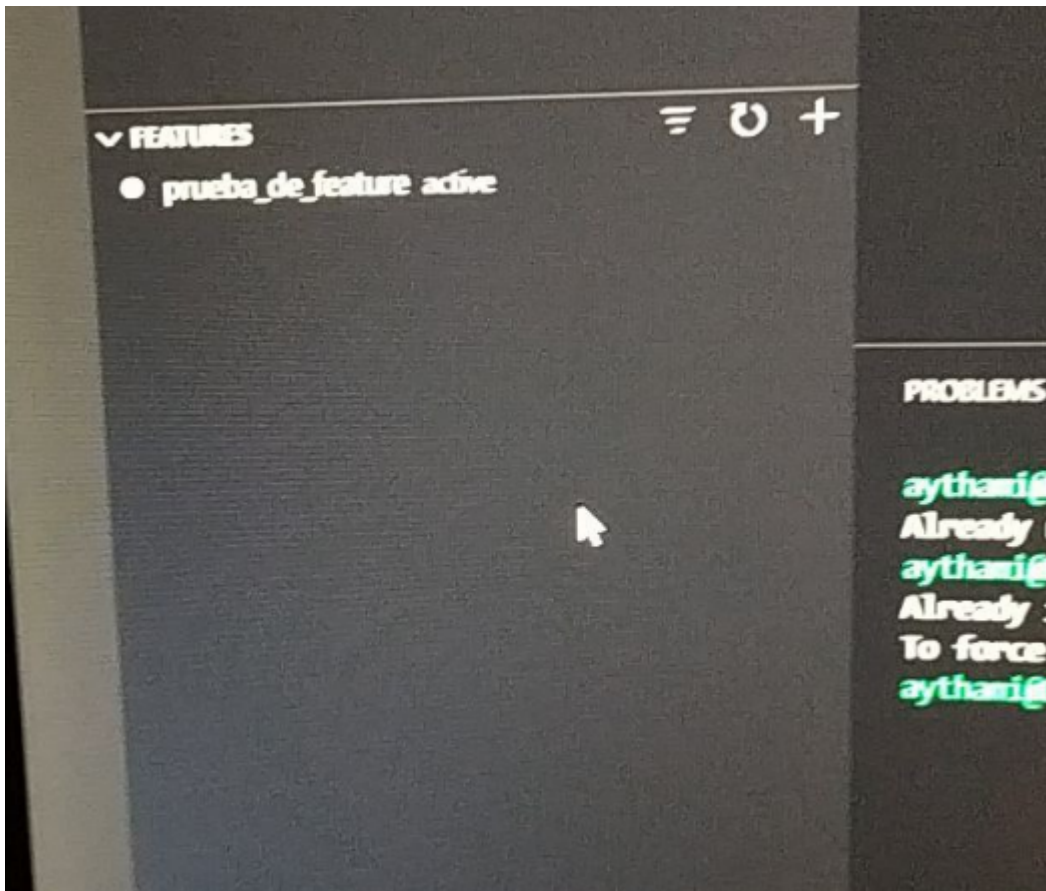
- Release es la copia de develop que se va a pasar a producción donde se pueden testear posibles fallos de las mismas. y es el que lleva con sigo los cambios de versiones.
- Hotfix es una copia de master que sirve para arreglar los errores.

A la hora de usar gitflow primero necesitamos instalarlo para ello usaremos el comando `Sudo apt install git-flow` o en caso de window su instalador correspondiente. para usar gitflow de manera comoda sin comandos, tambien nos descagaremos esta extension del [visual estudio code](#).

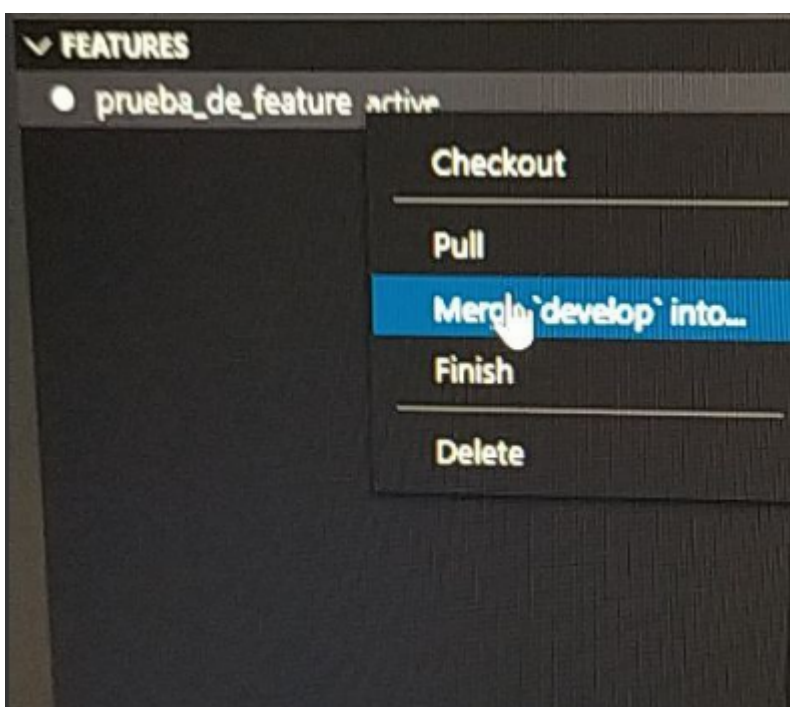


Lo primero para hacer uso de gitflow pondremos el comando `git flow init` en nuestra carpeta principal para agregarle las funciones de git flow a nuestro proyecto.

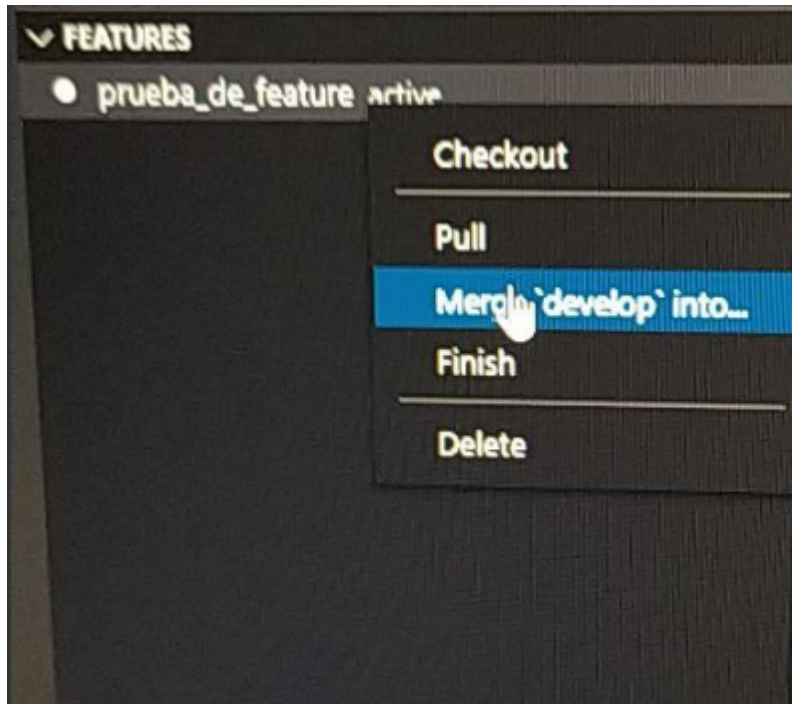
Ahora en nuestro plugin de gitflow de daremos al + que se ve en features esto nos permitirá crear una copia de developer de forma sencilla.



Trabajaremos en esta rama hasta terminar nuestro trabajo petición o función y luego usaremos el botón que pone merge.



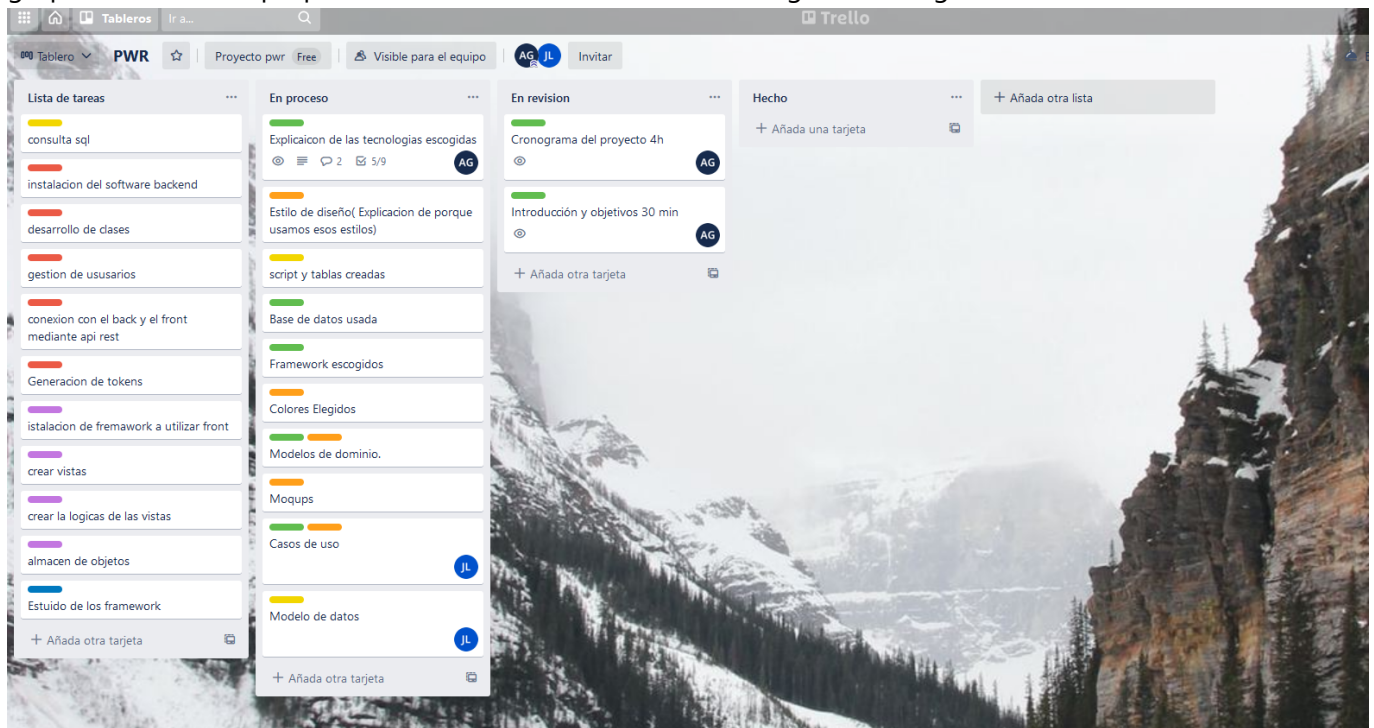
Podemos cerrar la rama en finish para no acumular muchas ramas de features.



Podemos crear la rama con el comando `git flow feature start feature_branch` que es una copia de develop y `git flow feature finish feature_branch` que cierra la rama y la mergea automáticamente.

## Trello

Trello es una herramienta que nos permite gestionar los trabajos que realizan cada uno de los integrantes del grupo. Para ello nos proporciona un tablero como se ve en la siguiente imagen.




Dando la capacidad de poner comentarios, hacer listas de tareas e ir cambiando la tarea según va avanzando en nuestro caso hemos puesto 4

## Diagrama de Gannt



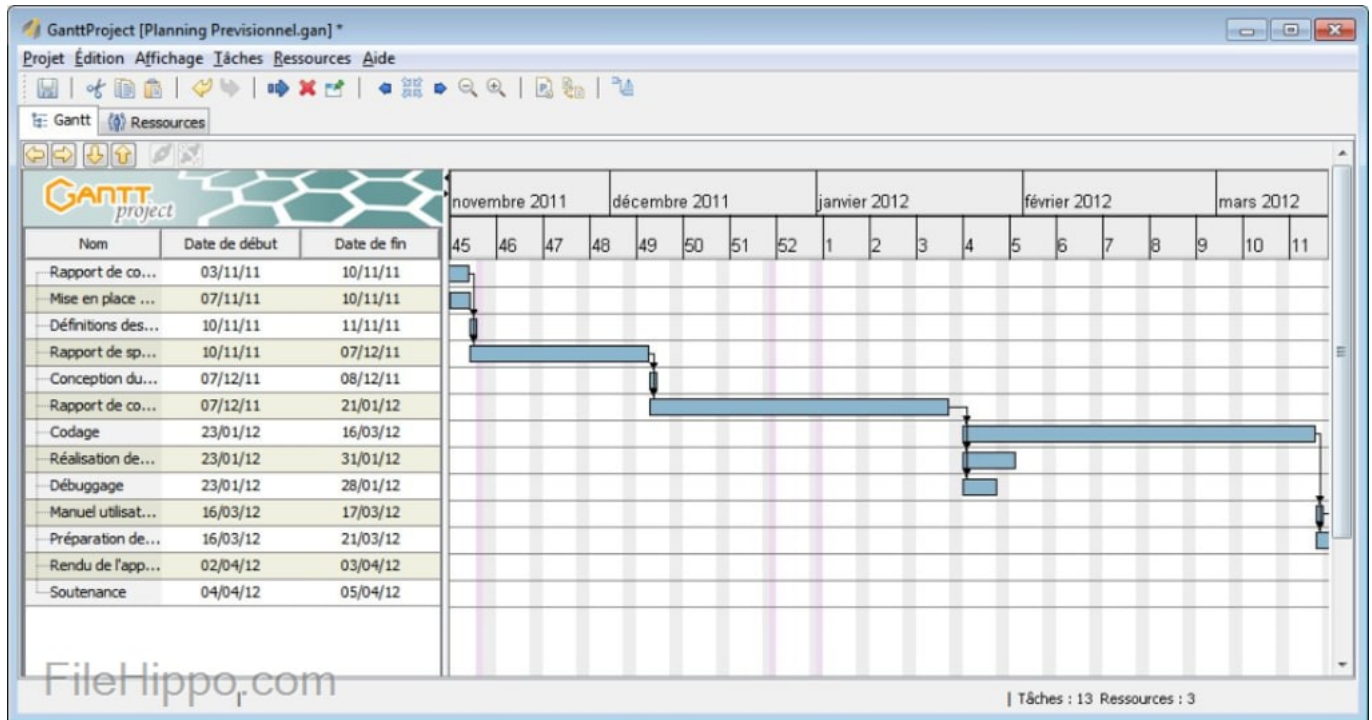
El diagrama de gantt es una forma útil de gestión que tiene que completarse en un proyecto y sus fechas.



Nombre	Fecha de inicio	Fecha de fin	Duración
● Introducción	20/3/21	20/3/21	2h
● Objetivos	21/3/21	21/3/21	2h
□ ● Tecnología Escogidas para usar	22/3/21	24/3/21	6h
● Base de datos usada	22/3/21	22/3/21	1h
● Explicacion de las Tecnolo...	23/3/21	23/3/21	3h
● Framework usados	24/3/21	24/3/21	2h
□ ● Diseño de la aplicacion	25/3/21	27/3/21	10h
● Cronologia	25/3/21	25/3/21	4h
● Casos de uso	26/3/21	26/3/21	4h
● Modelos de dominio	27/3/21	27/3/21	2h
□ ● Estilo de diseño	28/3/21	10/4/21	25h
● Moqups	28/3/21	4/4/21	12h
● Diseño de la vista	5/4/21	10/4/21	12h
● Colores Elegidos	5/4/21	5/4/21	1h
□ ● Base de datos	20/3/21	10/4/21	6h
● Modelo de datos	20/3/21	31/3/21	6h
● Script y tablas creadas	1/4/21	1/4/21	2h
● Consulta Sql	1/4/21	10/4/21	10h
□ ● Backend	12/4/21	3/5/21	
● Intalaciopn del software	12/4/21	12/4/21	4h
● Desarrollo de clases	13/4/21	18/4/21	10h
● Desarrollo de controladores	19/4/21	23/4/21	30h
● Gestion de usuarios	24/4/21	28/4/21	20h
● Conexion del back con el ...	29/4/21	3/5/21	10h
□ ● Front	24/4/21	16/5/21	
● Instalacion del software	24/4/21	24/4/21	4h
● Crear Vistas	25/4/21	5/5/21	30h
● Crear la logica de las vistas	6/5/21	11/5/21	30h
● Almacen de objetos	12/5/21	16/5/21	10h
● Tiempo para improvisos	17/5/21	28/5/21	
□ ● Estudio de las Tecnologias a ...	20/3/21	28/5/21	
● Documentacion	29/5/21	6/8/21	

como vemos en la imagen El diagrama de gantt cuenta con un nombre para las tareas, fecha de inicio , fecha de fin y tiempo para realizar las tareas. Estos elementos puesto como se ven en la imagen nos proporcionan las duración estimada del proyecto. Cuánto tiempo y en qué fechas se deben entregar cada una de las tareas del mismo.

Usar estos diagramas nos aporta claridad en el proyecto, una vista general del mismo, datos sobre el rendimiento , mejor gestión del tiempo y flexibilidad.



#### Claridad

Nos aporta claridad gracias a ver las tareas desglosadas y las líneas temporales en una sola vista del proyecto.

#### Comunicación

Esto ayuda a las diferentes partes del equipo a llevar un tiempo de entrega, reduciendo las reuniones.


#### Motivación

Al tener unos tiempos de entrega tan marcados se motivan a entregar el trabajo para no retrasar el resto del proyecto.

#### Flexibilidad

Los programas nos permiten editar el tiempo y las horas de trabajo para ajustar el tiempo de entrega ya que estos suelen variar según las circunstancias.

El programa usado en esta ocasión para hacer el diagrama es [Ganttproject](#) aprobenchando que es de libre uso. Como se ve en la imagen tenemos la opción de crear el proyecto con nombre , fecha inicio y fecha fin.



Nombre	Fecha de inicio	Fecha de fin	Duración
● Introduccion	20/3/21	20/3/21	2h
● Objetivos	21/3/21	21/3/21	2h
□ ● Tecnología Escogidas para usar	22/3/21	24/3/21	6h
● Base de datos usada	22/3/21	22/3/21	1h
● Explicacion de las Tecnolo...	23/3/21	23/3/21	3h
● Framework usados	24/3/21	24/3/21	2h
□ ● Diseño de la aplicacion	25/3/21	27/3/21	10h
● Cronologia	25/3/21	25/3/21	4h
● Casos de uso	26/3/21	26/3/21	4h
● Modelos de dominio	27/3/21	27/3/21	2h
□ ● Estilo de diseño	28/3/21	10/4/21	25h
● Moqups	28/3/21	4/4/21	12h
● Diseño de la vista	5/4/21	10/4/21	12h
● Colores Elegidos	5/4/21	5/4/21	1h
□ ● Base de datos	20/3/21	10/4/21	6h
● Modelo de datos	20/3/21	31/3/21	6h
● Script y tablas creadas	1/4/21	1/4/21	2h
● Consulta Sql	1/4/21	10/4/21	10h
□ ● Backend	12/4/21	3/5/21	
● Instalaciopn del software	12/4/21	12/4/21	4h
● Desarrollo de clases	13/4/21	18/4/21	10h
● Desarrollo de controladores	19/4/21	23/4/21	30h
● Gestion de usuarios	24/4/21	28/4/21	20h
● Conexion del back con el ...	29/4/21	3/5/21	10h
□ ● Front	24/4/21	16/5/21	
● Instalacion del software	24/4/21	24/4/21	4h
● Crear Vistas	25/4/21	5/5/21	30h
● Crear la logica de las vistas	6/5/21	11/5/21	30h
● Almacen de objetos	12/5/21	16/5/21	10h
● Tiempo para imprevistos	17/5/21	28/5/21	
□ ● Estudio de las Tecnologias a ...	20/3/21	28/5/21	
● Documentacion	29/5/21	6/8/21	

El programa nos permite agregar nuevas éticas y hacer subTareas. Entonces agregue duración y para tener en cuenta cuánto deben durar las tareas

## Usos de las Tecnologías en nuestro proyecto y otros datos de interés.

### Como hacer los commit

En nuestro caso hemos decidido hacer los commit de la siguiente manera 'información de lo que has realizado fecha y duración ' quedando de la siguiente manera los commit 'Bibliografía y bitacora actualizada 04/04/2021 5min'.

### Como Trabajamos con git-flow

Aprovechamos la funcionalidad de git flow que nos crea una rama master y develop, y las funciones de features para crear cada una de las funcionalidades de nuestro código. Estas ramas se crear con el nombre de la tarea y a que parte del código pertenece por ejemplo login\_backend.

## Base de datos usada

La base de datos que hemos decidido usar es Mysql o Mariadb. Para esta elección hemos tenido en cuenta las bases de datos que puede usar nuestro framework de backend que son:

- MySQL
- Postgresql
- SQLite3
- SQL Server

Luego hemos tenido en cuenta la experiencia usando las distintas bases de datos nombradas anteriormente, las dos que hemos usado más son Mysql y Sqlite.

En cuanto a sqlite está diseñada para funcionar en memoria o con dispositivos móviles.

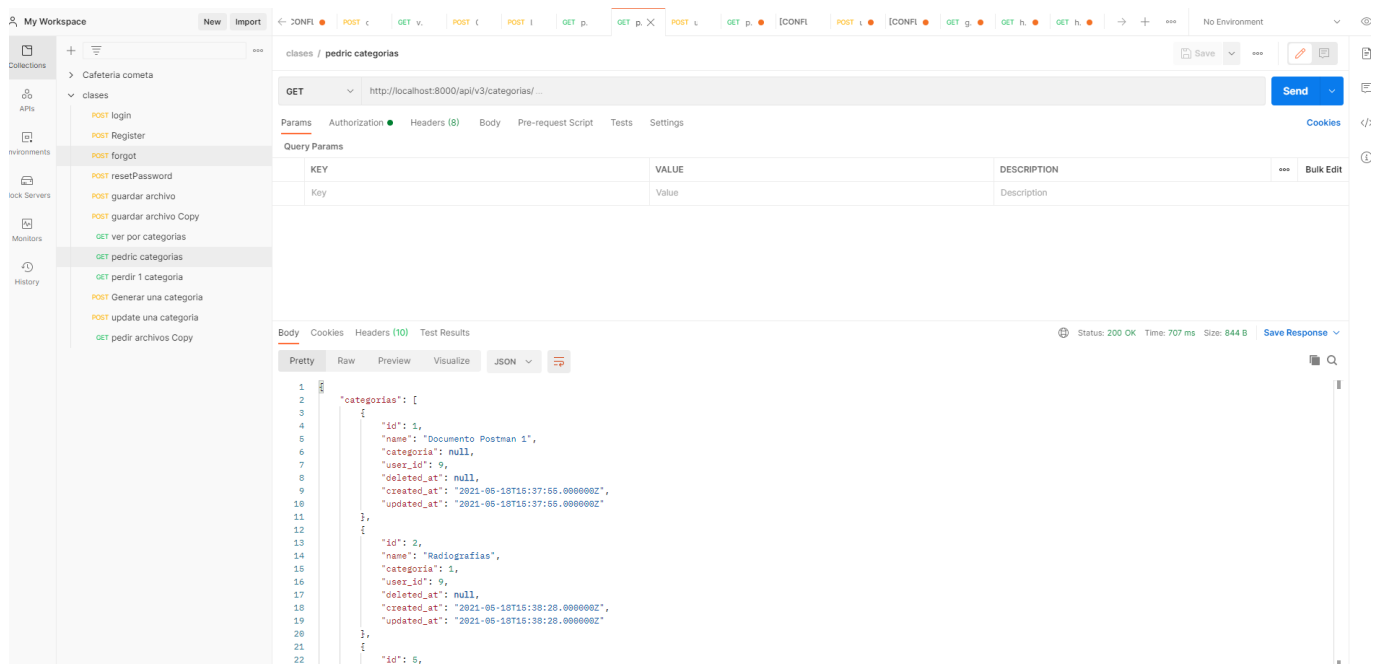
Mysql está más diseñada para páginas web con un servidor integrado, también es una de las que más hemos usado y viene instalada junto con Xampp que será nuestro software de desarrollo.

Las otras bases de datos no las hemos tratado en el instituto y están orientadas a grandes flujos de datos.

## Postman

Postman es una herramienta que se utiliza, sobre todo, para el testing de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de sistemas. ... Quizás sea una de las herramientas más utilizadas para hacer testing exploratorio de este tipo de sistemas.

En este caso posman lo hemos utilizado para ir haciendo pruebas del funcionamiento de los endpoint de nuestra api en la documentación de backend vemos en detalle su funcionamiento.



Pero nos permite hacer peticiones de todo tipo y recibir la correspondiente respuesta del servidor.

## Framework

En cuanto a framework este año no hemos visto muchos framework en el front hemos visto Angular y en el back apenas hemos llegado a ver laravel y sinfonía en la parte de php y Spring en la parte de java(Gracias a la documentación de JC).



Otro de los hándicap a tomar en cuenta ha sido la dificultad de los integrantes del grupo a la hora de programas en una u otro lenguaje de programación y el lenguaje que se está usando en la prácticas de ambos para poder tener un poco de ayuda en caso de necesitarla.

Por ello hemos elegido en el front Angular ya que es el lenguaje que está usando Javier en el front y Laravel por la documentación y que Aythami está usando otro framework muy parecido de back en la empresa de practicas llamado CodeIgniter 3.