

# Magazine

## Objetivo del proyecto

- ✚ Desarrollar una Revista "Magazine".
- ✚ Creación de BD MySQL
- ✚ Creación de servidor con PHP
- ✚ Creación de Front para cliente
- ✚ Creación de Panel de control para CRUD
- ✚ Creación de Login para administradores

## Requisitos del proyecto

La interfaz debe tener las siguientes necesidades de parte del clientes:

1. Hacer login.
2. Hacer logout.
3. Listar Post.
4. Organizar las Artículo por categorías

Este interfaz solo debe realizarse si el usuario esta logeado a parte administrativo

1. Poder indicar la fecha de publicación de un post ( no será visible si la fecha es anterior a la fecha de publicación )

También desarrollarás un panel de control para poder gestionar los posts que cubrirá las siguientes necesidades Backend Admin:

1. Hacer login
2. Hacer logout
3. Listar posts
4. Listar Categorías
5. Crear un post
6. Leer un post
7. Editar un post
8. Remover un post

## Implementación del proyecto

Para llevar a cabo este proyecto será necesario que tengas en cuenta dos roles principales.

- ✚ El sistema de **backend en PHP** que se encarga de obtener la información de la base de datos
- ✚ El sistema de **frontend** que se encargará de consumir los datos que provee tu propio sistema de backend.

Es importante que tengas en cuenta que existen dos “Frontales”:

- ✚ Magazine público
- ✚ El panel de control que se usará para gestionar el Magazine

Es importante que tengas en cuenta los roles de usuario ya que un usuario **no logeado** no debería poder realizar las acciones de un usuario **logeado**.

## Especificaciones del proyecto

1. Debes de usar GIT desde el inicio del proyecto
2. Para la parte de desarrollo en **PHP** no deberás usar ningún framework
3. Deberás usar MySQL para gestionar la base de datos.
4. Todo el código debe de estar debidamente documentado
5. Debes de incluir una pequeña guía de usuario para entender cómo interactuar con la herramienta
6. La interfaz debe de ser totalmente **responsive**
7. Todos los comentarios incluidos en el código tienen que estar escritos en inglés
8. Usa el estilo de código camelCase
9. En el caso de estar usando HTML nunca uses estilos en línea
10. En el caso de estar usando diferentes lenguajes de programación siempre define la implementación en términos separados
11. Es recomendado dividir las tareas en varias subtarefas para que de esta forma puedas asociar cada paso en particular de la construcción con un commit específico
12. Para la documentación del proyecto es requerida una versión en PDF dentro del repositorio
13. Debes intentar en la medida de lo posible que los commits y las tareas planificadas sean lo mismo
14. Borra los ficheros que no se usen
15. Deberás crear un fichero README.md correctamente documentado en el directorio raíz del proyecto (ver pautas en **Recursos**)

# LISTA DE TAREAS A REALIZAR

Tarea	Prioridad	Horas
Documentación	Alta	1,00
Organización	Alta	2,00
Búsqueda Previa de información	Normal	1,00
Creación de repositorio	Baja	0,10
Documentando Proyecto	Normal	2,00
Creacion base de datos	Alta	2.00
Estructura HTML index php Principal	Baja	0,30
Htaces	Alta	0,30
Arquitectura MVC	Alta	2,00
Views	Alta	2.00
Models	Alta	2.00
Controllers	Alta	2.00
Creación README	Baja	2,00
Testing / Corrección Errores	Alta	1,00

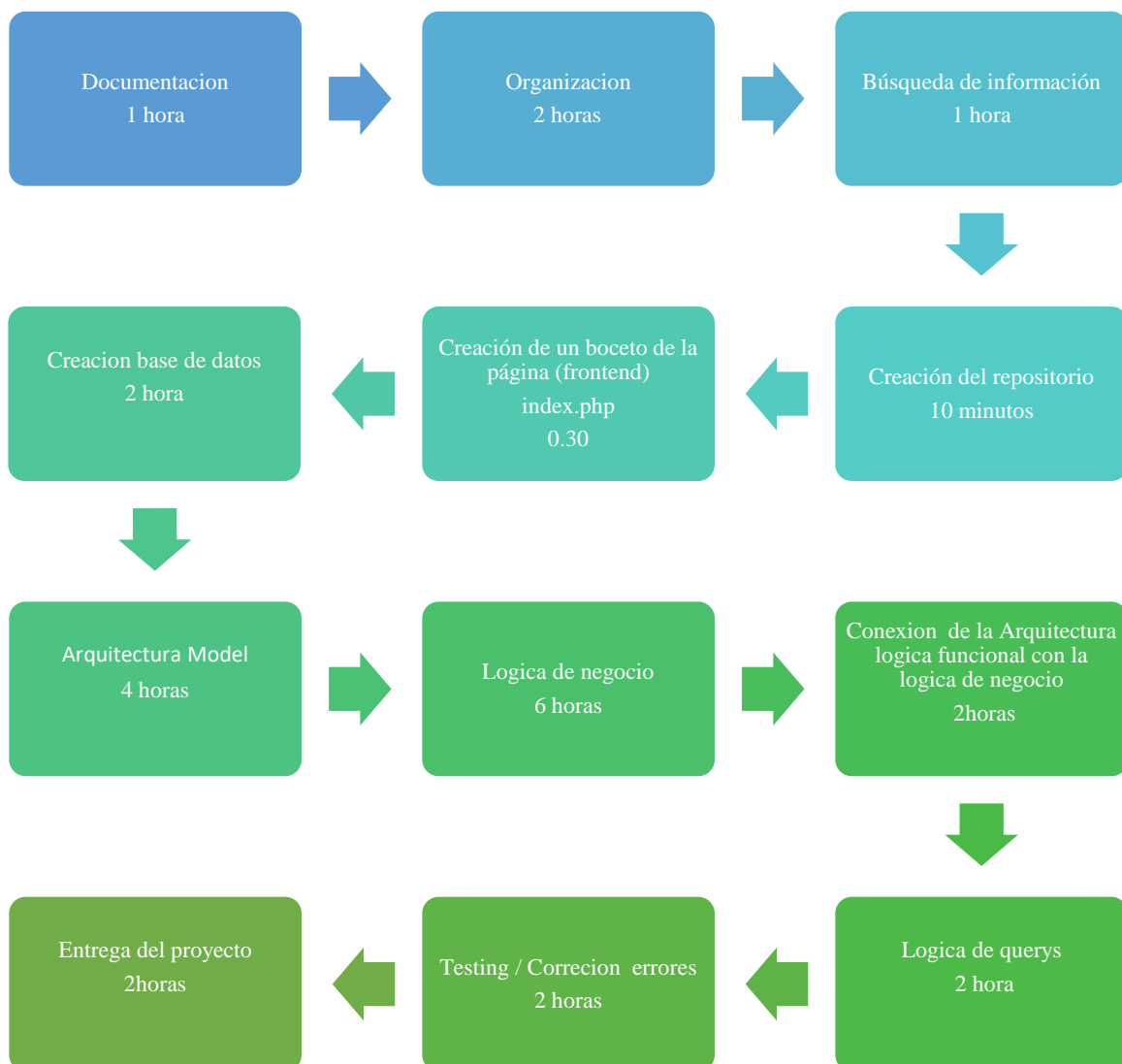
Tenemos previsto terminar el proyecto en 18 Horas

En las cual damos 8 horas más para prever incidencias en el proyecto

[Registro De Incidencia que se ha detectado durante el proyecto](#)

- Petición a la base de datos solución hay varias petición como hacerlo
- Controlador llamando al model solución con un `requiere_once` del model
- Controlador enviando datos a views correctamente solución petición desde views desde action con el controlador mas el método
- En rutamiento solución arreglo del htaccess
- Tiempo

### Seguimiento Calendario del proyecto



### Métricas de Calidad

A pesar de que el proyecto debe adherirse a todos los requisitos y especificaciones del proyecto, hay algunas condiciones que, si se cumplen adecuadamente, añaden una sensación de calidad y robustez al propio proyecto. Estas condiciones son:

1. Levantar el WampServer conectándose a mysql y localhost.
2. El código php debe estar libre de pelusas, como lo indica un INTELLIGENCE de PHP de confianza.
3. La aplicación web debe responder.
4. Diseño responsive

## Documentación de Riesgos

- Retrasos de proyecto.
- Pérdida o daños de material de trabajo.

## Documentación WORKFLOW DE GIT

- Creación Git Hub <https://github.com/robertfox11/Magazine.git>
- Hacemos commits de la estructura de la página principal.
- Probabilidad de que ocurra 80%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación) Pedir ayuda a compañeros
- Probabilidad de que ocurra 30%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación) Pedir ayuda a compañeros
- No encontrar con facilidad la información relacionada con el proyecto
- Probabilidad de que ocurra 30%
- Impacto en el proyecto 60%
- Posible alternativa (mitigación)
- Pedir ayuda a compañeros

A partir de la realización de la estructura se continuó trabajando solamente en la

rama "master", a través del Workflow "Gitflow".

Mas información --> <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow>



## Herramiento del proyecto

Se utilizaron diferentes herramientas en el desarrollo del proyecto. Son los siguientes:

1. **git: un potente sistema de control de** versiones que ayuda a realizar un seguimiento de los cambios en el árbol de trabajo.
2. **Visual Studio Code: un editor de** código optimizado para crear y depurar aplicaciones web modernas.
3. **WampServer**, viene integrado el servidor Web Apache, openSSL para soporte SSL, base de datos MySQL y lenguaje PHP
4. **Herramientas para desarrolladores de Google Chrome:** se utiliza para depurar el código JavaScript y para probar los ajustes de diseño.
5. **Documentos de Google:** se utiliza para escribir la documentación del proyecto.
6. **Validador W3C:** utilizado para validar el código HTML y CSS.
7. **ESLint:** utilizado para validar el código JavaScript.

## Flujo de trabajo de Git

Todas las confirmaciones se van a insertar en la rama **maestra**, siguiendo un criterio personal de cargar solo instantáneas que son funcionales y que funcionan correctamente, sin contar errores menores. No hay otras ramas, ya que ralentizaría el proceso de desarrollo.

En otro lado, los mensajes de confirmación finalizan con su objetivo principal indicado entre corchetes: por ejemplo.

- Las relaciones principal para la logica funcional es [class]
- las confirmaciones relacionadas principalmente con los cambios CSS comienzan con **[styling]**;
- los relacionados principalmente con el diseño de la página, **[layout]**;
- los relacionados con el proyecto añadiendo carpetas o documentacion, plugins, library **[us]**;

- 🚦 relacionados principalmente con la documentación, **[documentation]**.
- 🚦 Para php vista la relacionamos con [index]
- 🚦 Estructura mvc

## Estructura del archivo

Los archivos del proyecto se organizarán de esta manera:

APPMagazine/	
assets/	
img/	Carpeta que contiene todas las imágenes utilizadas en la interfaz.
bootstrap/	Carpeta que contiene css responsive utilizados en la interfaz de usuario.
css/	Carpeta que contiene todos los estilos utilizados en la interfaz de usuario.
Config/config.php	Configuración de enrutamiento principal
DataBase.php	Petición de conexión a la base de datos MySQL
Controller/	Controladores al modelo y al views
Documentation/	Documentation del proyecto
Helpers/	Unas clases con funcionalidades bases
Models/	Modelos de class para la petición de datos y funcionalidades
Views/	Vistas de rutas para la acción de los métodos que viene de controllers
	/layout
	Vistas principales de header footer de php
.gitignore	Carpeta utilizada por <i>git</i> para contener información sobre el repositorio.
.htacces	archivo que permite modificar el engine de apache para crear el enrutamiento RewriteEngine ON
DataBase/	Base de datos sql Magazine que utilizaremos
se encuentra también el diagrama de la base de datos	
Usaremos para realizar las peticiones	

Autoload.php/ se encargara la carga automática vinculando las partes del programa desde el almacenamiento masivo

index.php Página principal de la aplicación web.

README.md Archivo que contiene instrucciones sobre cómo ejecutar el proyecto.