VUE JS

# Visión general

El objetivo es entender el framework para renderizar  vista para construir interfaces de usuario y aplicaciones de una sola página.

# Tabla de contenidos

[**Fase I - Iniciación del Proyecto**](#_w1vzj3h1laqw) **1**

[Requisitos del proyecto](#_rf0k2roeckt1) 1

[Especificaciones del proyecto](#_7gxddv3dxn6r) 1

[**Fase II - Planificación de Proyectos**](#_47d6lr53hx3l)2

[Razonamiento](#_sa0tgkpsanuv) 2

Implementaciones 3

[Planificación de tareas](#_yk4g8lqtyomw) 3

[Seguimiento](#_v4kuirxnt5lf) del calendario de proyecto 3

[Flujo de trabajo de Git](#_525hxfjvck8g) 4

[Herramientas](#_1baqvpbig0cx) 4

[**Fase III - Ejecución del proyecto**](#_llw0m4o3mrui)5

[Incidentes](#_8q5vkahqigl5) 5

[Lecciones](#_qghm3c6m59y2) 5

[**Fase IV - Proyecto Closure**](#_jb72b8bwfrvr) **6**

[Comentarios generales](#_doygaox2gvoh) 6

# Fase I - Iniciación del Proyecto

## Requisitos del proyecto

* Peticiones de librerias con npm tambien podemos con cdn
  + Librería axios .
  + Verificación del nuevo directorio llamado Vendor

## Especificaciones del proyecto

* El proyecto debe desarrollarse en Vuejs o Javascript, Html, SCSS.
* Crear un repositorio GIT
* No subir la dependencia
* La estructura de directorios del proyecto debe estar bien definida y organizada.
* El código debe documentarse correctamente utilizando el idioma inglés.
* El código debe usar un estilo *camelCase.*
* En caso de usar HTML no usar estilos en linea
* El proyecto no debe contener archivos no utilizados.
* El proyecto debe desarrollarse mediante *git*, utilizando mensajes deconfirmación explícitos y concisos.
* Borrar ficheros que no sean necesario para evaluar el proyecto
* El proyecto debe contener un archivo *README* escrito en *Markdown* que muestre una breve descripción y los pasos para runél.

# 

# Fase II - Planificación de proyectos

## Razonamiento

Esta píldora se desarrollar una spa que muestre galeria de imágenes, que se cargara grupos de 5 imágenes a medida que el usuario haga scroll vertical.

En cuenta

* Posicion del scroll del usuario que esta visitando la pagina
* Consumir la API, obtener bloques de imágenes en la peticion.
* Controlar el numeros de peticiones que se realiza
* Realizar un componente para no repetir el codigo con los datos data
* Poner en practica los metodos de un componente

## Organizacio del codigo

En este pequeño proyecto nos hemos enfocado en poner a prueba **PHPUnit**, por lo que no hemos tenido en cuenta la organización de nuestro código. Es muy importante que organices de forma adecuada. Crea los siguientes directorios:

* app ( será el responsable de contener el código fuente de tu app )
* test ( será el responsable de contener los test )
  + test/app ( este directorio se crea a modo de mantener la misma estructura que la app original para facilitar la ubicación de los test de cada uno de los ficheros)

## Planificación de tareas



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tarea** | **Prioridad** | **Horas** | **Dificultad** | **ID** |
| Documentación | Alta | 1,00 | Alta | 1 |
| Organización | Alta | 2,00 | Alta | 2 |
| Búsqueda Previa de información | Normal | 3,00 | Normal | 3 |
| Creación de repositorio | Baja | 0,15 | Baja | 4 |
| Estructura índex.html | Baja | 0.30 | Normal | 5 |
| Installar Libreria Necesaria como axios | Normal | 0.10 | Baja | 6 |
| Creacion Metodos | Normal | 0.30 | Normal | 7 |
| Creación README | Baja | 0,30 | Baja | 12 |
| Testing / Corrección Errores | Alta | 0,30 | Normal | 13 |
| Entrega de proyecto | Alta | 0.20 | Alta | 14 |

**Seguimiento Calendario del proyecto**

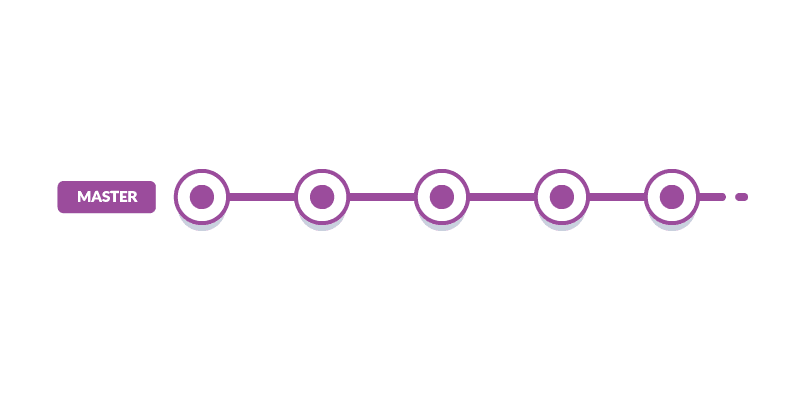
## Documentación WORKFLOW DE GIT

* Creación Git Hub [**https://github.com/robertfox11/PillsVuejs.git**](https://github.com/robertfox11/PillsVuejs.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>



## Herramientas

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

* ***git: un potente sistema de control de*** versiones que ayuda a realizar un seguimiento de los cambios en el árbol de trabajo.
* ***Visual Studio Code: un editor de*** código optimizado para crear y depurar aplicaciones web modernas.
* ***Vuejs, JavaScript, Html, SCSS***
* ***Librería*** *Composer y PHP UNIT*
* ***Herramientas para desarrolladores de Google Chrome:*** se utiliza para depurar el código JavaScript y para probar los ajustes de diseño.
* ***Documentos de Google:*** se utiliza para escribir la documentación del proyecto.
* [***Validador W3C***](https://validator.w3.org/)***:*** utilizado para validar el código HTML y CSS.
* [***ESLint***](https://eslint.org/demo)***:*** utilizado para validar el código JavaScript.
* ***nano: editor de texto básico que utiliza la interfaz de línea de*** comandos.
* ***curl:*** herramienta de línea de comandos utilizada para transferir datos utilizando varios protocolos de red.
* ***Documentos de Google:*** se utiliza para escribir la documentación del proyecto.

# Fase III - Ejecución del proyecto

## Incidentes

¡Ninguno, por suerte!

## Lessons

Todas las tareas se completaron sin tener que hacer frente a ningún obstáculo importante.

# Fase IV - Cierre del proyecto

## Comentarios generales

La píldora se completó con éxito en el intervalo de tiempo que se predijo en [*la planificación*](#_yk4g8lqtyomw)de tareas.