

# MindFlow

Pablo Gutiérrez Valdín



1. Introducción .....	4
1.1 Contexto .....	4
1.2 Presentación .....	6
1.3 Objetivos .....	10
2. Análisis y diseño de la aplicación .....	10
2.1. Análisis de requisitos funcionales .....	10
2.1.1. Introducción.....	12
2.1.2. Tabla numerada de requisitos.....	12
2.2. Diagrama de casos de uso.....	13
2.2.1. Introducción.....	13
2.2.2. Diagramas y descripción .....	13
2.3. Diseño de interfaz.....	16
2.3.1. Introducción.....	16
2.3.2. Diagramas y descripción .....	16
2.4. Estructuras de datos usadas .....	21
2.4.1. Introducción.....	21
2.4.2. Diagramas y descripción .....	21
2.5. Fase de pruebas .....	22
2.5.1. Pruebas de validación de código .....	22
2.5.2. Pruebas de implantación en distintos soportes .....	25
3. Implementación .....	27
3.1. Tecnologías utilizadas en el desarrollo del proyecto .....	27
4. Puesta en marcha. Despliegue .....	28
4.1. Guía de despliegue del proyecto .....	28
5. CONCLUSIÓN .....	31
5.1 Valoración Personal del Trabajo Realizado. ....	31
5.2 Posibles Ampliaciones.....	31
6. BIBLIOGRAFÍA Y WEBGRAFÍA.....	32
6.1 MANUAL DE USUARIO II. ....	32
6.2 CÓDIGO COMPLETO DEL PROYECTO (Enlace a GitHub .....	33
7. APLICACIÓN EN EL AMBITO EMPRESARIAL .....	34
7.1. Idea de negocio .....	34

7.2. Planificación de necesidades de RRHH .....	36
7.3. Planificación de necesidades de producción .....	36
7.4. Planificación de la Prevención de Riesgos Laborales .....	38
7.5. Previsiones de inversión y financiación.....	39
7.6. Estudio de mercado y marketing operativo .....	40

# 1. Introducción

## 1.1 Contexto

**Título del proyecto:** MindFlow

**Descripción:**

Aplicación de productividad/Gestión tareas que cuenta con numerosas funcionalidades para llevar el seguimiento de tanto tareas cotidianas como trabajo real o incluso manejo de proyectos.

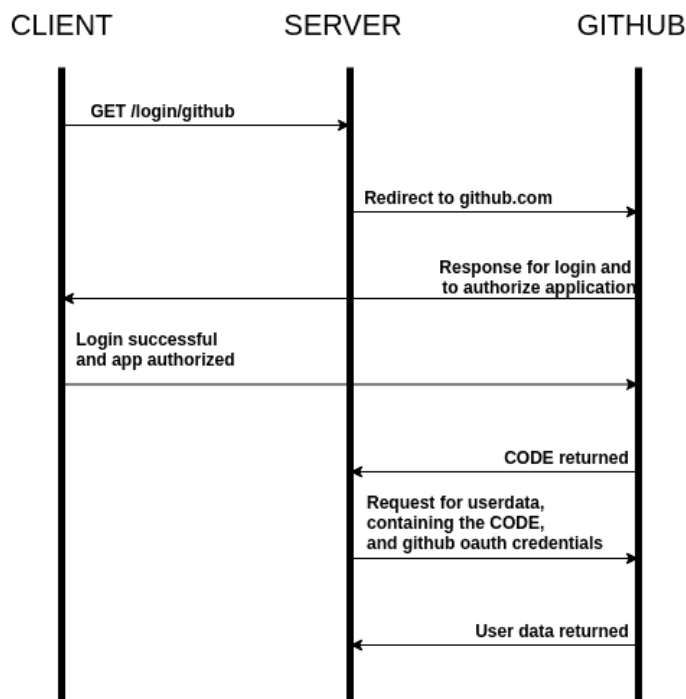
Es una aplicación muy centrada en la experiencia de usuario con interfaces atractivas y llamativas siguiendo estándares de web moderna.

Cuenta con las siguientes funcionalidades:

- **Todo List** : Se trata de una funcionalidad que nos permite crear listas de tareas con diferentes estados (*Todo, Doing, Done*) y que busca ayudar en el manejo de tareas. Puede enfocarse para el trabajo y tareas cotidianas. Además, busca dar una buena experiencia de uso al usuario ofreciendo funcionalidades como el drag and drop entre columnas.  
Cada tarea cuenta con un nivel de prioridad, una corta descripción y una fecha de caducidad.  
Además, cada tarea puede ser editada, eliminada o cambiada de estado. Los datos se hacen permanentes gracias a la API.
- **Calendar Task**: Esta funcionalidad, parecida a la anterior, busca dar un enfoque distinto al manejo de tareas permitiendo al usuario insertar la tarea dentro de un calendario con color (tag) que representa su tipo. Dentro de los tags aceptados se encuentran "*Personal*", "*Work*", "*Study*" y "*Other*".  
Dentro del calendario podemos movernos entre fechas y desplazar las tareas con drag and drop. El calendario está disponible actualmente solo para los días de la semana (De lunes a viernes). Esto se debe tanto a que busca seguir el calendario laboral y a que añadir hacer el calendario más largo podría empeorar la experiencia de usuario, sobre todo para dispositivos pequeños, ya que esta funcionalidad está diseñada especialmente para pantallas grandes. También se pueden borrar las tareas.

- **Github Data Manager:** Nos permite manejar el github del usuario sin necesidad de entrar en él. Puede ser útil para buscar rápidamente un proyecto y ver sus últimas novedades o información general mientras navegamos por MindFlow y trabajamos en otras cosas.

Esta funcionalidad nos muestra nuestros proyectos privados y públicos tras loguearnos con nuestro github. Una vez logueados se cargan a partir de la api de github. Tenemos opciones para navegar entre ellos y hacer búsquedas y filtrados. Una vez entramos en uno podemos ver su información y los últimos commits y issues en una interfaz moderna y bonita. Además, tenemos la opción de editar o cerrar issues. Para realizar esto según el usuario que tiene la persona se crea un **oAuth de github**. Funciona de la siguiente forma:



- **Notes Editor:** Nos permite crear páginas de notas (El límite lo establece el tipo de usuario) y en ellas podemos insertar texto. Además, contamos con un pequeño menú de herramientas para mejorar la experiencia usando el editor, nos permite, por ejemplo, poner texto en negrita o alinearlos a la derecha.

El editor actúa solamente en almacenaje local para proporcionar más velocidad en el guardado y actualización del contenido. Por lo tanto, no va enlazado a una cuenta concreta sino a un navegador de un dispositivo.

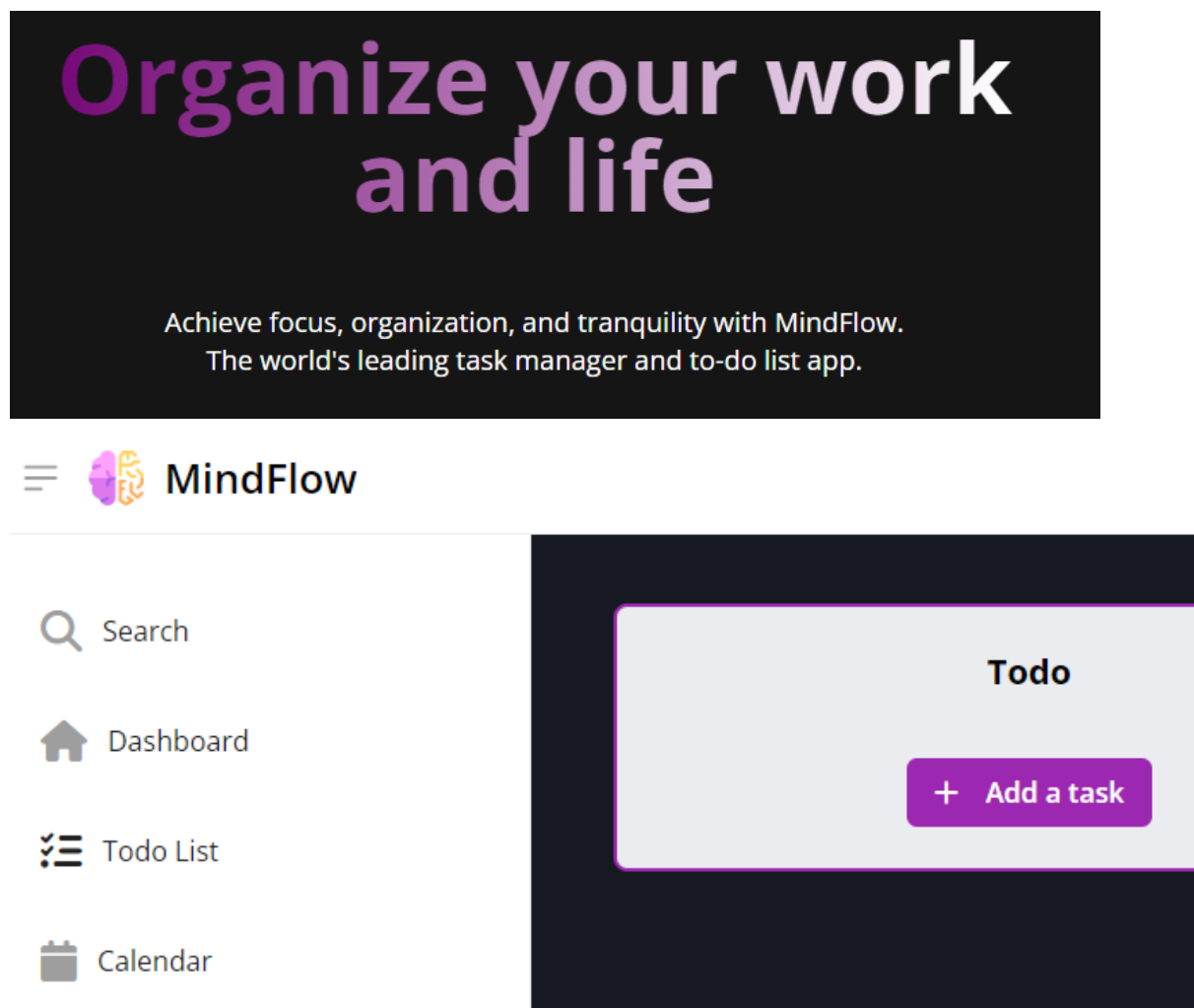
- **Statistics** : Funcionalidad que nos muestra información de la creación de tareas y el cierre de ellas tanto de “*Todo List*” como de “*Calendar tasks*” durante un periodo de tiempo. Esta información se muestra en gráficos y es exclusiva a un tipo de usuario privilegiado, ya sea el premium o admin.

## 1.2 Presentación

Como comento en el punto del diseño de la interfaz, la realización de un diseño atractivo y moderno fue muy importante para mí.

Mi aplicación MindFlow cuenta con un diseño oscuro que combina con blancos y grises claros y el color que identifica la aplicación, el morado.

Podemos ver el conjunto de todos estos en varios lugares de la aplicación:



La aplicación también cuenta con numerosos componentes de UI cómo:

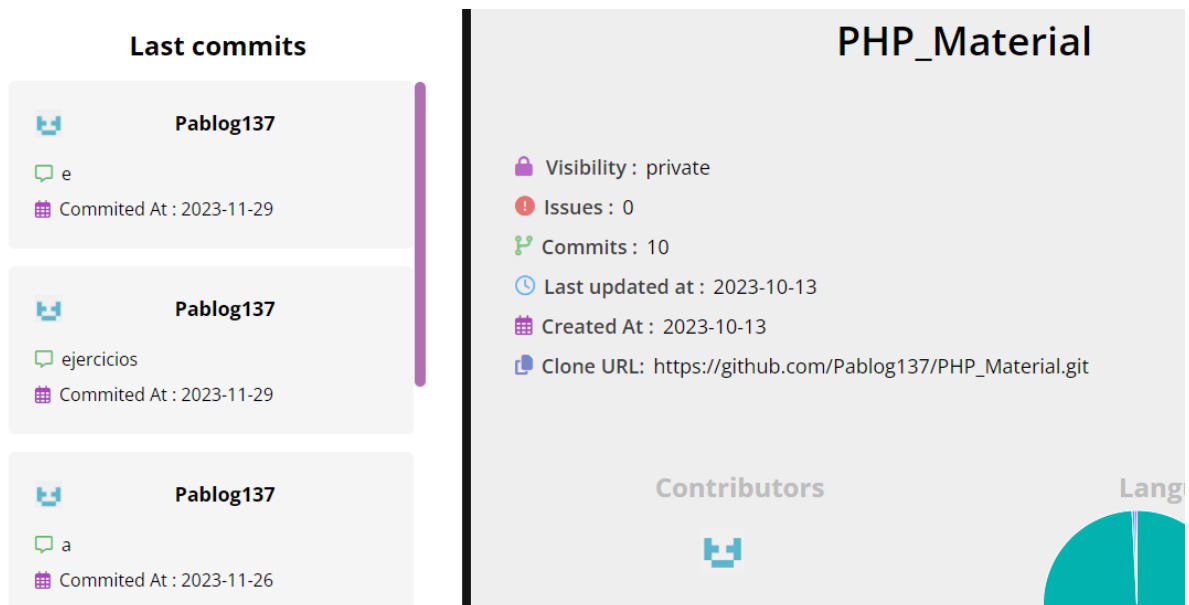
- Quotes
- Cards
- Buttons
- Accordions
- NavBars
- AsideBars
- Modales
- Alerts
- Formularios
- Iconos
- Gráficos

Además, la aplicación considera la reutilización de componentes y muchos se reutilizan varias veces. Para su desarrollo se utiliza tailwind, una tecnología nueva para mí.

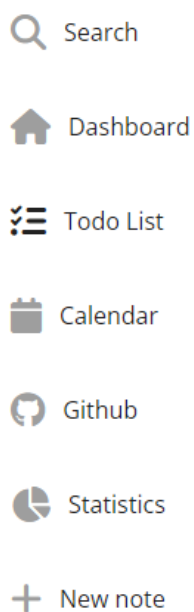
El logotipo de la aplicación cuenta con el color utilizado como tema y representa, como hace el nombre, la unión del cerebro con la tecnología, explicando la idea de un “segundo cerebro” dónde almacenamos nuestras ideas y tareas:



La aplicación busca siempre tener una buena accesibilidad y facilidad de entendimiento dando uso por lo tanto a numerosos iconos explicativos o ilustraciones:



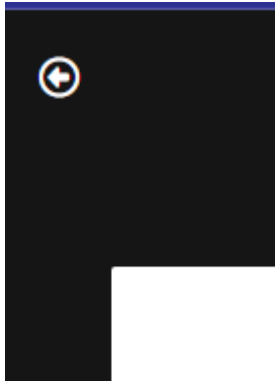
Además, la navegación es sencilla y explicativa mediante iconos. La aplicación ofrece el uso de una barra lateral dónde aparecen todas las secciones de la aplicación:



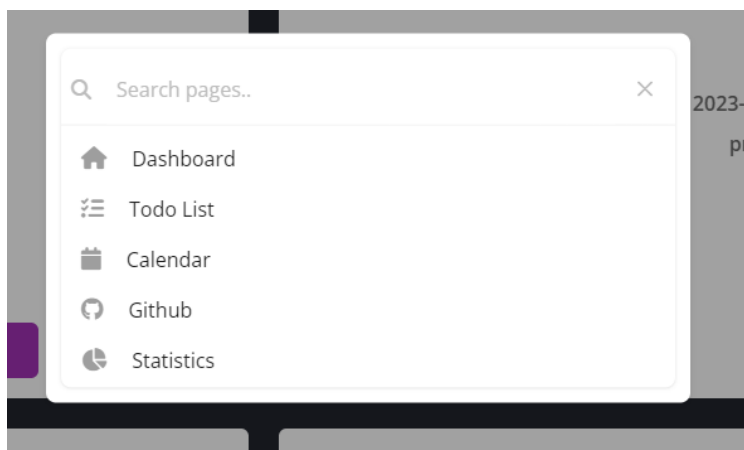


Además de que permite cerrarse y abrirse según el espacio que necesites para ver el resto de contenido.

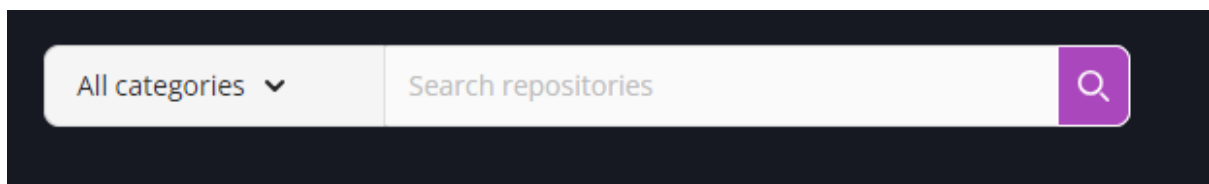
También se cuenta con flechas para volver hacía atrás:



O un buscador de páginas:



Lo mismo para los repositorios:



Por último, se considera el uso de herramientas semánticas para favorecer el SEO de la aplicación y se usan los elementos adecuados cuando son posibles.

## 1.3 Objetivos

Los objetivos que busca cumplir la aplicación son:

- **Agrupar funcionalidades que mantengan relación entre ellas** para crear una aplicación robusta que no obligue al usuario a desplazarse a otras aplicaciones
- **Crear una aplicación con interfaz moderna y siguiendo patrones de UX y UI** para dar al usuario una experiencia agradable.
- **Realizar una aplicación que siga mis intereses y pueda ser útil** tanto para mí como para otras personas
- **Juntar mis conocimientos de tecnologías y desarrollo junto con nuevos conocimientos a aprender** para conseguir realizar un producto entero. Muchas herramientas utilizadas fueron completamente nuevas para mí.

## 2. Análisis y diseño de la aplicación

### 2.1. Análisis de requisitos funcionales

- **Creación e inicio de sesión con usuarios**
- **Reiniciar contraseña**
- **Realizar pagos**
- **Realizar filtrados de información**

- **Funcionalidad Todo List:**
  - Crear listas de tareas con diferentes estados (Todo, Doing, Done).
  - Drag and drop entre columnas.
  - Asignar nivel de prioridad, descripción corta y fecha de caducidad a cada tarea.
  - Editar, eliminar o cambiar el estado de las tareas.
  - Los datos son permanentes gracias a la API.
- **Funcionalidad Calendar Task:**
  - Insertar tareas en un calendario con colores que representan su tipo.
  - Tags aceptados: Personal, Work, Study, Other.
  - Navegación entre fechas y desplazamiento de tareas con drag and drop.
  - Disponible para los días de la semana (De lunes a viernes).
- **Funcionalidad Github Data Manager:**
  - Manejar el GitHub del usuario sin necesidad de entrar en él.
  - Ver proyectos privados y públicos después de iniciar sesión con GitHub.
  - Navegación, búsqueda y filtrado de proyectos.
  - Ver información, últimos commits e issues de los proyectos.
  - Editar o cerrar issues.
- **Funcionalidad Notes Editor:**
  - Crear páginas de notas con un límite establecido por el tipo de usuario.
  - Insertar texto con un pequeño menú de herramientas para mejorar la experiencia del editor.
  - Almacenamiento local para proporcionar velocidad en el guardado y actualización del contenido.
  - Borrado de nota.
- **Funcionalidad Statistics:**
  - Mostrar información de creación y cierre de tareas de Todo List y Calendar Tasks durante un período de tiempo.
  - Presentar la información en gráficos.
  - Acceso exclusivo para tipos de usuario privilegiados, como premium o admin.

### 2.1.1. Introducción

### 2.1.2. Tabla numerada de requisitos

Número	Requisito
1	Creación usuario
2	Login de usuario
3	Reinicio de contraseña
4	Desloguearse
5	Creación de tarea
6	Edición de tarea
7	Borrado de tarea
8	Cambio de estado de tarea
9	Búsqueda de páginas en lupa
10	Creación de tarea de calendario
11	Eliminación de tarea calendario
12	Cambio de fecha tarea calendario
13	Desplazamiento por calendario
14	Inicio de sesión con github
15	Búsqueda y filtrado de repositorios
16	Ver detalles de repositorio
17	Editar issue
18	Cerrar issue
19	Vista de estadísticas
20	Cambio de tipología en estadísticas
21	Cambio de periodo de tiempo en estadísticas
22	Creación página notas
23	Poner título/contenido página notas
24	Funcionalidades menú página notas
25	Eliminar página notas
26	Realizar pago
27	Cambio de divisas
28	Cambio de ejemplos de resoluciones
29	Selección de preguntas y respuestas
30	Cambios entre usuario con sus funcionalidades apropiadas
31	Diseño Responsive

## 2.2. Diagrama de casos de uso

### 2.2.1. Introducción

Existen 3 tipos de usuario en la aplicación, en el siguiente punto se indica cada usuario con sus características.

### 2.2.2. Diagramas y descripción

#### Usuario normal:

Tiene acceso a la mayoría de las funcionalidades menos a la sección de estadísticas y su creación de notas es muy limitada, solo puede tener 3 simultáneamente.



## Usuario premium:

El usuario premium tiene acceso a todas las funcionalidades menos a realizar pagos, ya que se intuye que ya los ha realizado para ser premium y no podrá volver a pagar hasta que pase su periodo y vuelva a usuario normal.

En el caso de las notas podrá tener hasta 10 simultáneamente.



## Administrador:

El usuario administrador es muy parecido al premium, tiene los mismos privilegios para ver todo menos el formulario de pagos, ya que no lo necesita, y no tiene límites en la creación de notas. Además, que su cuenta no expirará.

Ambos usuarios pueden ver la sección de estadísticas.



## 2.3. Diseño de interfaz

### 2.3.1. Introducción

Antes de empezar a desarrollar el proyecto me dediqué varios días a realizar vistas en figma. Algunas de estas fueron seguidas al pie de la letra mientras que otras sufrieron cambios durante el desarrollo.

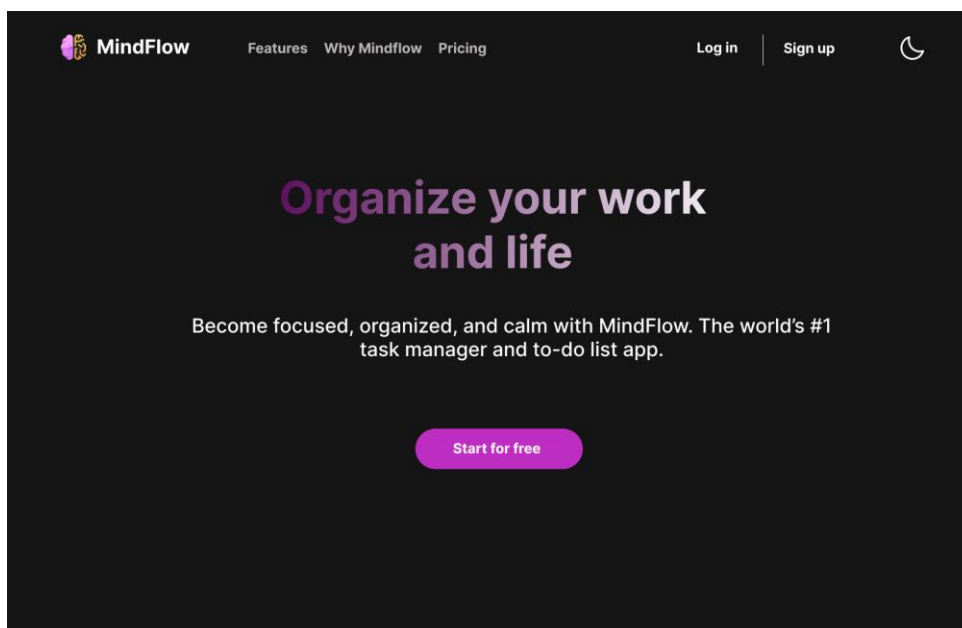
Usar figma me permitió crear diseños mucho más complejos y bonitos ya que me hizo centrarme en su diseño y creación antes de tocar código. Desde el punto de vista de terminar el proyecto fue quizás la mejor decisión que realicé.

- <https://www.figma.com/design/2yQpnIPqoS9RkEeWDearb6/MindFlow?node-id=0-1&t=UgkIsD5O0IWxjvDi-0>
- <https://www.figma.com/design/2yQpnIPqoS9RkEeWDearb6/MindFlow?node-id=13-2>
- <https://www.figma.com/design/2yQpnIPqoS9RkEeWDearb6/MindFlow?node-id=16-90>



### 2.3.2. Diagramas y descripción

Se incluyen algunos de los diseños de figma:

- **Landing Page:**






[Features](#)[Why Mindflow](#)[Pricing](#)[Log in](#)[Sign up](#)

## Choose your pricing plan

Price in EUR

### Basic




# Free

- ✓ All app features
- ✓ Create up to 5 pages

[Choose plan](#)

### Premium





# €3.99

/month

- ✓ All app features
- ✓ Unlimited pages
- ✓ Access to graphs and analytics


[Choose plan](#)



[Features](#)[Why Mindflow](#)[Pricing](#)[Log in](#)[Sign up](#)

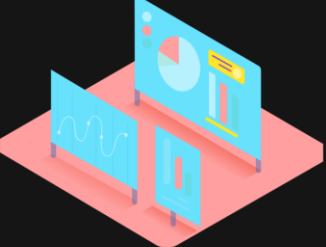
## Why choose us?

With MindFlow you have everything you need to keep life organized.

“MindFlow gives me the peace of mind knowing I am doing exactly what I need to do, exactly when I need to do it.”




[Features](#)[Why Mindflow](#)[Pricing](#)[Log in](#)[Sign up](#)



### With our premium subscription, you gain exclusive access to powerful analytics and dynamic charts

[Pick your plan](#)

- Log In / Register




## Log in to your account

Email

Password [Forgot password?](#)

[Log in](#)

Don't have an account? [Create an account](#)



## Create an account

Name

Email

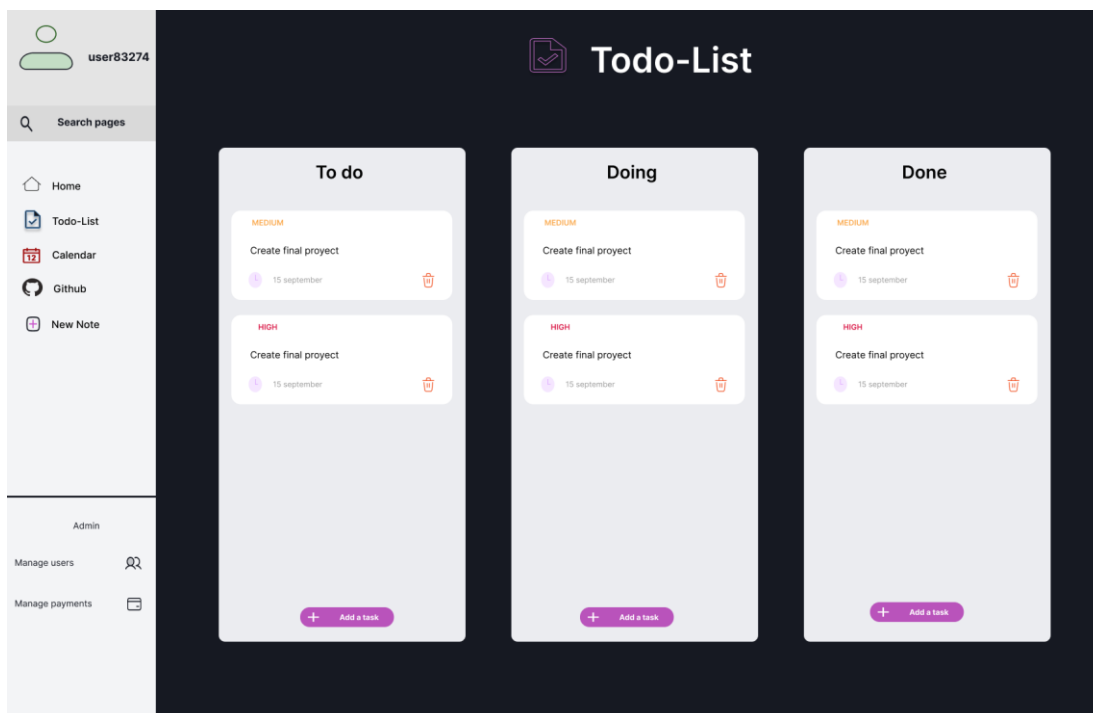
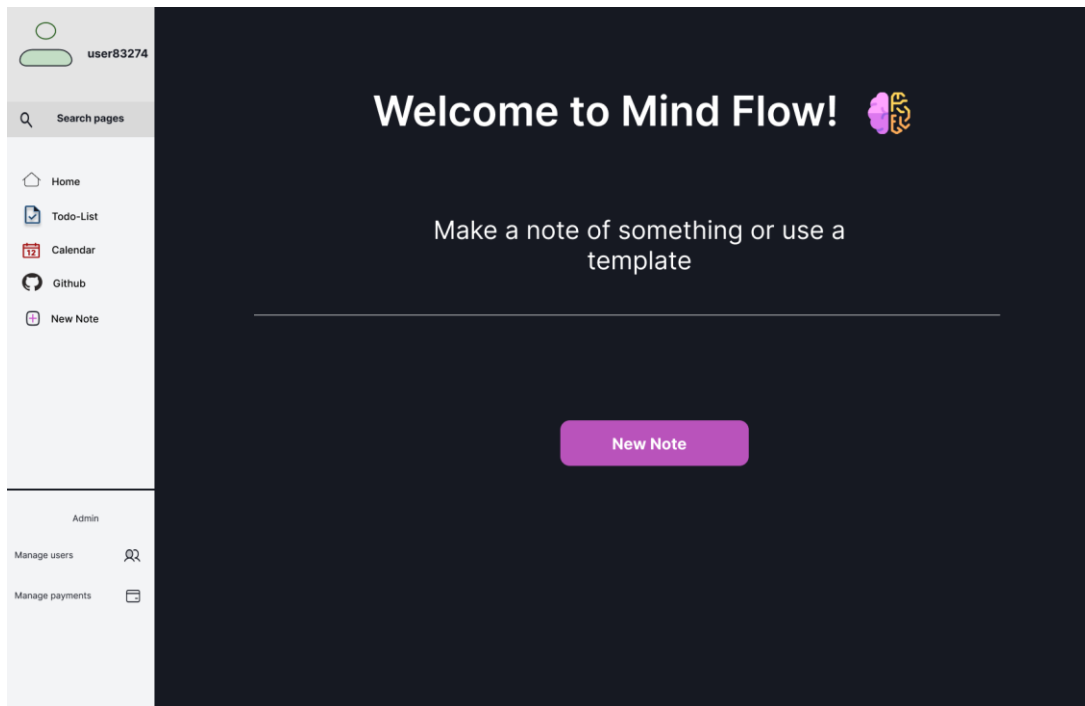
Password

Repeat password

[Sign up](#)

Already have an account? [Log in](#)

- Interior aplicación



My repos

Prueba repo

Last updated at : 2019-04-12

Visibility : Public

Total Issues : 120

Total Commits : 300

View More

Prueba repo

Last updated at : 2019-04-12

Visibility : Public

Total Issues : 120

Total Commits : 300

View More

Prueba repo

Last updated at : 2019-04-12

Visibility : Public

Total Issues : 120

Total Commits : 300

View More

Prueba repo

Last updated at : 2019-04-12

Visibility : Public

Total Issues : 120

Total Commits : 300

View More

Week

Month

< October >

2024

Monday	Tuesday	Wednesday	Thursday	Friday
31	31	31 <div>oafjlasoljfsd</div> <div>oahsdifohjsadf</div>	31	31
31	31	31	31	31
31	31	31	31	31
31	31	31	31	31
31	31	31	31	31

20

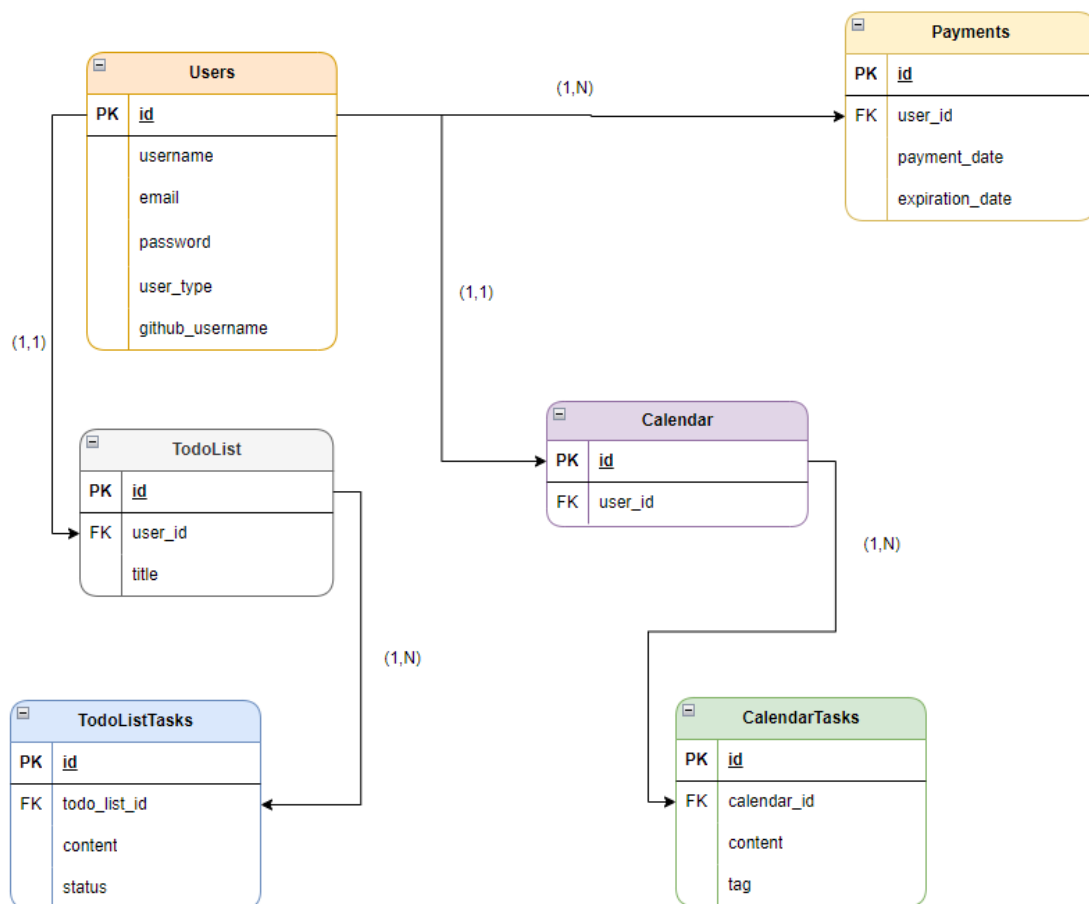
## 2.4. Estructuras de datos usadas

### 2.4.1. Introducción

Se usa una base de datos relacional, MySQL, para manejar los datos de la aplicación.

Para el almacenaje de datos de la aplicación contamos con 6 tablas, además de almacenaje en local storage. Por ejemplo, el de las notas. También Cookies para algún dato pequeño.

### 2.4.2. Diagramas y descripción



## 2.5. Fase de pruebas

Para validar el funcionamiento correcto de la aplicación se realizan tandas de pruebas de caja negra por cada sección / funcionalidad. Las pruebas se indican en el siguiente punto, tanto el título de la prueba como su resultado.

### 2.5.1. Pruebas de validación de código

#### Landing page

Prueba	Resultado
Funcionamiento correcto barra nav	OK
Funcionamiento correcto botones	OK
Funcionamiento botón Login/Register	OK

#### Login

Prueba	Resultado
Enviar formulario vacío. Debe no dejar	OK
Login inválido: Enviar con solo un campo. Debe no dejar	OK
Login con campos completos pero inválidos para la validación. Debe no dejar	OK
Login con usuario no existente. Debe no dejar	OK
Login válido	OK
Inyección SQL. Debe no funcionar	OK

#### Register

Prueba	Resultado
Enviar formulario vacío. Debe no dejar	OK
Register inválido: Algún campo vacío. Debe no dejar	OK
Register con campos completos pero inválidos para la validación. Debe no dejar	OK
Register con usuario existente. Debe no dejar	OK
Register válido	OK
Inyección SQL. Debe no funcionar	OK

#### Forgot password

Prueba	Resultado
Correo no válido. Debe mostrar error	OK
Correo no existente. Debe mostrar error	OK
Correo exitoso	OK
Recibe correo	OK
Crea nueva contraseña	OK
Crea nueva contraseña no valida. Debe mostrar error	OK

#### Interior aplicación

Prueba	Resultado
Funcionamiento aside bar	OK
Crear new note	OK
Abrir y cerrar aside	OK
Desloguearse	OK
Funcionamiento de la lupa	OK

#### Todo-List

Prueba	Resultado
Creación tarea vacía. Debe mostrar error	OK
Creación tarea con campos incorrectos. Debe mostrar error	OK
Creación tarea exitosa	OK
Reinicio de campos tarea	OK
Mover tarea de estatus	OK
Eliminar tarea	OK
Editar tarea con datos erróneos. Debe mostrar error	OK
Editar tarea exitosamente	OK

#### Calendario

Prueba	Resultado
Moverse entre fechas	OK
Creación tarea vacía. Debe mostrar error	OK
Creación tarea datos erróneos. Debe mostrar error	OK
Creación tarea exitosa	OK
Reiniciar formulario tarea	OK
Mover tarea entre días	OK
Borrar tarea	OK

## Github

Prueba	Resultado
Login con github erróneo. Debe mostrar error	OK
Login con github exitoso	OK
Búsqueda de repos y filtrado por categoría	OK
Ver detalles	OK
Eliminar	OK
Editar dejando vacío. Debe mostrar error	OK
Editar exitoso	OK

## Notas

Prueba	Resultado
Creación de nota	OK
Poner título	OK
Uso del menú de opciones, sin poder mezclar	OK
Almacenar datos	OK
Injectar SQL. Debe no dejar	OK
Eliminar nota	OK
Crear más notas que el límite. Debe no mostrar creación	OK

## Estadísticas

Prueba	Resultado
Cargado de datos y correctos	OK
Mostrar solo para usuarios privilegiados	OK
Cambiar entre todo list y calendario	OK
Cambiar durante fechas	OK

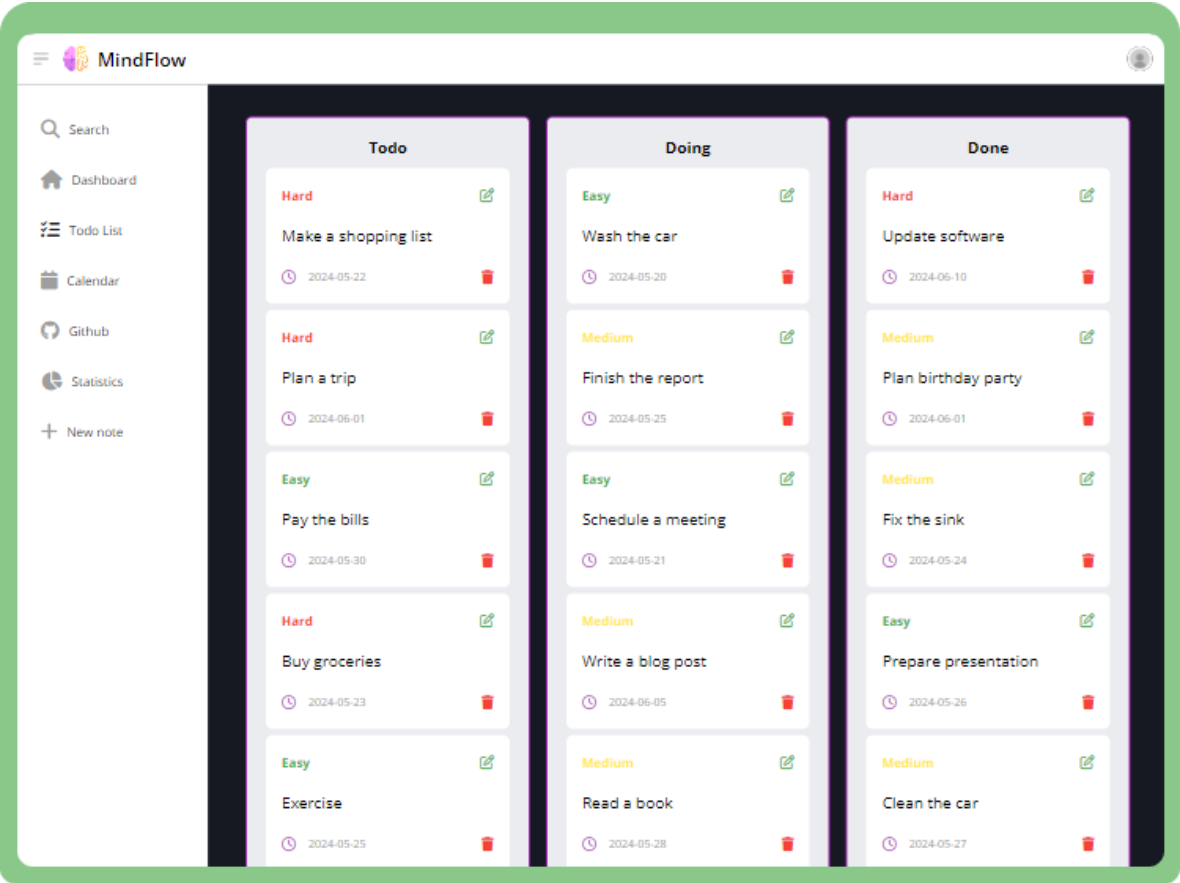


Prueba	Resultado
Realizar pago	OK
Cambio de usuario al pagar	OK
Se muestran funcionalidades apropiadas a usar	OK
Límite de notas por tipo usuario	OK

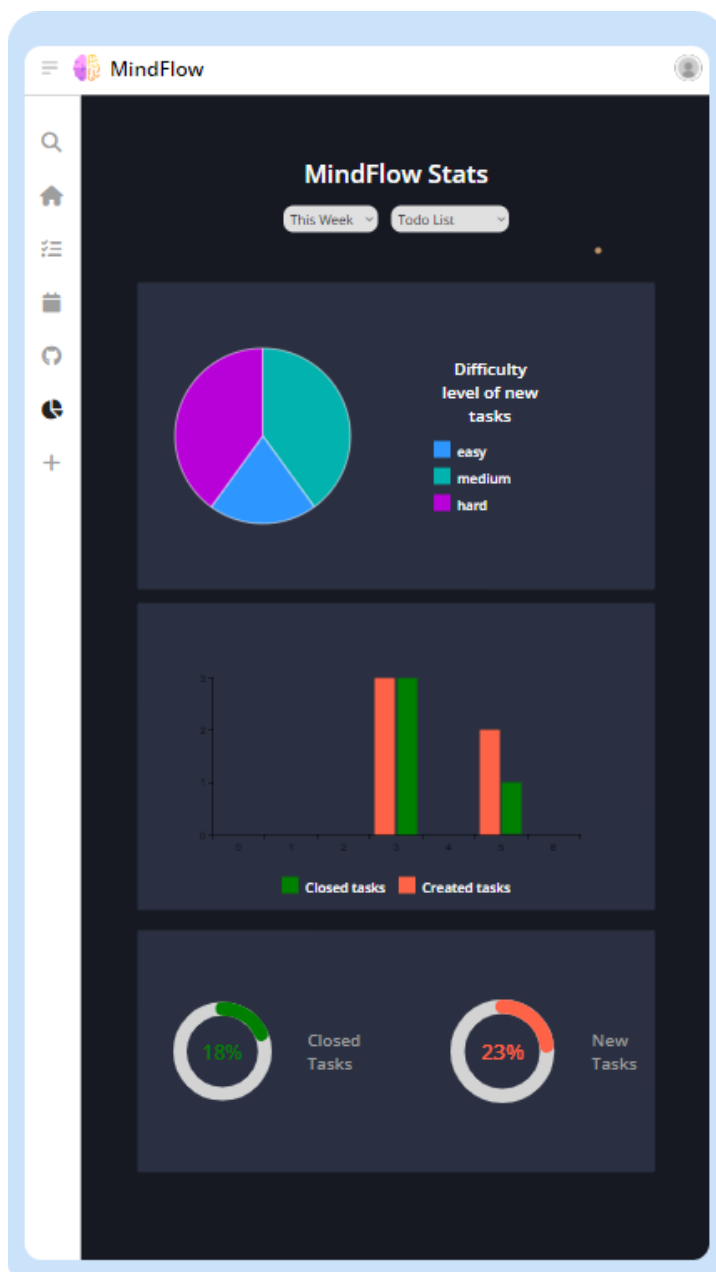
2.5.2. Pruebas de implantación en distintos soportes

El proyecto está probado para distintas resoluciones y es totalmente responsive. Ha de tenerse en cuenta que es un proyecto destinado a usarse en pantallas medianas/grandes (Ordenadores o tablets) ya que su experiencia de usuario es mucho mejor. Si se quiere usar para móvil puede reducirse su experiencia ya que no es una interfaz sencilla de mostrar en pequeño.

Muestra en pantalla mediana/grande



## Muestra en pantalla pequeña



## 3. Implementación

### 3.1. Tecnologías utilizadas en el desarrollo del proyecto

Front end	
Tecnología	Razón de uso
React	Fue usado para realizar las interfaces de usuario. Por su rapidez, reusabilidad y acceso a documentación es la mejor opción. También se emplearon librerías como react dnd (drag and drop)
TypeScript	La razón de su uso fue para mejorar el código JavaScript y evitar errores que de no usarlo hubieran sucedido. También para aprendizaje personal.
Tailwind CSS	Para realizar diseños rápidamente y muy personalizables ya que extiende las funcionalidades de Bootstrap. Se utilizan varias librerías de componentes como MUI y Flowbite
Tecnologías y recursos extra	También se emplean recursos para la UI como Font awesome, manypixels, spinkit, etc.

Back end	
Tecnología	Razón de uso
Laravel	Por su facilidad de uso y familiaridad con la tecnología, se emplea para realizar API. Para el acceso a datos se usa Eloquent y para las plantillas de correos Blade.
MySQL	Por todo lo que nos ofrece, por su gran uso y acceso a información y por familiaridad con la tecnología

Despliegue	
Tecnología	Razón de uso

<b>Docker</b>	Uso docker por la facilidad que nos da para poder compartir y desplegar el proyecto
<b>AWS</b>	Se desplegó el proyecto tanto en el servidor del instituto como en AWS.

## Otras tecnologías y herramientas

- **VSCode** para la producción de código por su facilidad de uso y extensiones
- **Git y Github** para llevar versiones del código y tenerlo en un repositorio
- **Trello** para planificar trabajo durante Sprints
- **Figma** para los diseños
- **Postman** para probar APIs
- **APIs Externas:**
  - API Github
  - API Chatgpt

## 4. Puesta en marcha. Despliegue

### 4.1. Guía de despliegue del proyecto

Actualmente la aplicación está desplegada en **AWS** y en el **servidor del instituto**. Cada uno tiene una instalación distinta.

En el caso de **AWS** deberíamos crear una instancia EC2 con la distribución de nuestra elección, así como las características de hardware. También es muy importante que abramos al público los puertos que utilizamos, sino será imposible acceder a la aplicación.

En **ambos despliegues** nos conectaremos a la instancia por medio de ssh. El servidor del instituto ya viene configurado con las instalaciones, pero para la EC2 necesitaremos instalar **Git y Docker**.

Una vez hecho esto, para cualquiera de los dos servidores necesitamos **clonar el proyecto del github**.

Cuando tengamos el proyecto clonado, el primer paso es comprobar que los **.env**, tanto del **back** como del **front** tienen los datos correctos para el despliegue. Por defecto se proporciona en el repositorio un **.env.example** que debería ser renombrado

**.env**. Este contiene los datos necesarios. Si se quisiera editar algún dato estos son los necesarios para que la aplicación funcione:

- **.env Back:**

- URL Base utilizada en el cliente. Esta es necesaria para manejar el envío de correos desde el back (Específicamente para indicar la URL que el usuario debe seguir para el reinicio de contraseña).
- Datos de conexión a la base de datos como pueden ser el puerto, host, y contraseña
- Datos de la aplicación OAuth de Github. Esto es una funcionalidad utilizada en la aplicación y requiere de un ID de cliente y un ID de cliente secreto que proporcione para que github reconozca mi aplicación.
- Datos de correo. Aquí se indica desde dónde se mandan los correos.

- **.env Front:**

- URL Server: Aquí se indica la URL del servidor con su puerto para realizar las peticiones.
- ID de cliente de github. Mencionado anteriormente, es el mismo.

**\*\* Importante \*\*** Si se quiere probar la funcionalidad de github deben cambiarse tanto el id de cliente como el id secreto ya que oAuth solo deja enlazar una URL de aplicación. Con lo cual existen tres aplicaciones oAuth con diferentes IDs (Local, AWS y servidor instituto).

El siguiente paso es crear los contenedores Docker. Para esto la aplicación tiene un Docker file en el back y otro en el front. Para relacionar todo esto y crear los contenedores se emplea un Docker compose ubicado en la raíz del proyecto. Este indica la creación de cada contenedor: El back, el front y la base de datos. También crea volúmenes para almacenar información de la aplicación en caso de parar o eliminar algún contenedor.

Para finalmente crear los contenedores se accede a la raíz del proyecto y se ejecuta el comando:

```
PS C:\Users\pablo\OneDrive\Documentos\MindFlow> docker-compose up -d
```

Cuando se acaben de crear los contenedores se necesita acceder al contenedor de laravel para ejecutar las migraciones (También se puede automatizar en el Docker file, pero he decidido hacerlo a mano por errores pasados).

Para acceder al contenedor se puede usar el siguiente comando:

```
PS C:\Users\pablo\OneDrive\Documentos\MindFlow> sudo docker exec -it ID /bin/bash
```

Dónde ID es el id del contenedor. Este podemos encontrarlo haciendo simplemente un docker ps.

```
PS C:\Users\pablo\OneDrive\Documentos\MindFlow> docker ps
CONTAINER ID   IMAGE                COMMAND                  CREATED        STATUS        PORTS                               NAMES
3daf0573f7b4   mindflow-react-app   "nginx -g 'daemon of..." 46 hours ago   Up 4 minutes   0.0.0.0:80->80/tcp                 mindflow-react-app-1
30f65a7edfb4   mindflow-laravel-app "docker-php-entrypoi..." 46 hours ago   Up 4 minutes   0.0.0.0:8000->80/tcp                 mindflow-laravel-app-1
56d0753bcb8e   mysql:5.7            "docker-entrypoint.s..." 46 hours ago   Up 4 minutes   0.0.0.0:3306->3306/tcp, 33060/tcp   mysql
```

Una vez dentro simplemente se necesita hacer el comando **php artisan migrate**,

Y seleccionar Si cuando nos pregunte si crea también la base de datos.

Ya hecho esto tenemos todo listo para usar la aplicación. El primer paso será acceder a la url de esta y crear un usuario o usar uno de los existentes.

- Para el desplegado en el instituto se accede mediante el dominio:  
<https://pablogv24.iesmontenaranco.com/>
- Para el desplegado en aws se accede por la IP pública, si se quisiera comprobar tengo que iniciar el laboratorio anteriormente:  
<https://3.218.109.164/>

## 5. CONCLUSIÓN

### 5.1 Valoración Personal del Trabajo Realizado.

El proyecto ha salido tal como tenía pensado ya que es algo que quería realizar desde tiempo. No he tenido grandes dificultades y he sabido resolver todo a su tiempo. El mayor problema al que me he enfrentado es el tiempo ya que, aunque haya empezado desde el día 1 este proyecto ha llevado muchísimas horas, tanto el diseño como el desarrollo y finalmente la realización de documentación, pruebas y documentación.

Lo que más me ha gustado de él es seguir mis intereses y tener que hacer toda la realización del software, tanto mis puntos débiles como fuertes. Estoy convencido de que esto me ha mejorado como programador y me servirá para mis futuros proyectos personales.

Me hubiera gustado haber hecho el back en otra tecnología para aprender, tenía pensado node.js o .NET, pero me hubiera llevado más tiempo ya que ya había decidido a introducir las nuevas tecnologías en el front y serían demasiadas cosas nuevas.

Como curiosidad, haciendo el proyecto descubrí que también me interesa bastante la parte de despliegue ya que anteriormente no le había dedicado tanto tiempo. Me hubiera gustado con más tiempo probar otros servicios de cloud como Azure.

En conclusión, he cumplido todo lo que me propuesto con trabajo muy duro y por lo tanto estoy muy contento con mi desempeño. El proyecto ha sido de gran ayuda para seguir progresado como programador.

### 5.2 Posibles Ampliaciones.

Como posibles ampliaciones me gustaría como comentario en la valoración cambiar el back a otra tecnología y probar otros servicios de cloud.

También me gustaría poder dedicar un esfuerzo a realizar tests unitarios en el front, con tecnologías como Jest y React Testing Library.

Dentro de la aplicación estoy muy contento con mi trabajo y he realizado todas las funcionalidades que tenía en mente. Si tuviera que implementar algo más elegiría el uso de chatgpt en mi aplicación, que es algo que tenía en mente realizar para el final del proyecto en caso de terminar antes. Finalmente, he decidido priorizar otras cosas y dejar esto de lado. La idea era introducir un endpoint para poder realizar preguntas a chatgpt dentro de la funcionalidad de notas.

## 6. BIBLIOGRAFÍA Y WEBGRAFÍA

Para el desarrollo de MindFlow he recopilado información de varias fuentes:

- **StackOverflow / ChatGpt** : Para buscar soluciones a problemas / bugs
- **Documentación oficial**: Para buscar como usar funcionalidades. Para esto he utilizado bastante la documentación de Tailwind ya que este framework era nuevo para mí. También he consultado para documentación de MUI, Drag and drop react, github API, Typescript, etc.
- En momentos puntuales he consultado algún **video** para una pequeña funcionalidad o para informarme sobre despliegues.
- **Foros/Blogs**: Para buscar como arreglar ciertas cosas

### 6.1 MANUAL DE USUARIO II.

Se proporcionan tres usuarios distintos, cada uno con su tipo de privilegios:

- **Usuario administrador**

Usuario: pablogutierrezvaldin@gmail.com

Contraseña: admin666

- **Usuario Premium**

Usuario: premium@gmail.com

Contraseña: admin666

- **Usuario Estándar**

Usuario: user@gmail.com

Contraseña: admin666

También se puede crear una cuenta, pero por defecto serás un usuario estándar. Los usuarios premium se crean al realizar la compra del premium y los administradores no se pueden crear, son proporcionados por la persona que se encargue del desarrollo de la aplicación.

El usuario **normal** puede acceder a todas las funcionalidades menos a las estadísticas y como máximo podrá tener 3 notas simultáneamente. Además, podrá ver la opción para pagos.



El usuario **premium** puede acceder a todas las funcionalidades incluido las estadísticas y tener un máximo de 10 notas simultáneamente.

El usuario **admin** podrá ver todas las funcionalidades y no tiene límites de notas.

La primera vista de la aplicación es la Landing Page. Esta se muestra seas usuario o no. Te muestra de que trata la aplicación. Una vez inicies sesión o te registres entrarás a la aplicación, al Dashboard. Para todos los usuarios es igual.

Una vez dentro de la aplicación puedes desplazarte en la barra lateral entre las distintas funcionalidades:

- En el TodoList podrás crear, editar y borrar tareas además de moverlas entre columnas de diferentes estados.
- En el calendar podrás crear tareas con una tipología y moverla dentro del mes.
- En el Github Data Manager podrás manejar tus repositorios y editar o cerrar issues.
- En las estadísticas (solo las verás si eres “admin” o “premium”) podrás ver en gráficos la creación de tareas y el cierre de ellas en comparación al total de ellas.
- En el símbolo de creación, en la barra lateral podrás crear notas. Tendrás un límite asociado al tipo de usuario.
- Finalmente, en la esquina superior derecha saldrá un perfil de usuario en el que clicándolo podrás cerrar sesión y te sacará de la aplicación.

## 6.2 CÓDIGO COMPLETO DEL PROYECTO (Enlace a GitHub)

Enlace: <https://github.com/daw2-23-24-pdaw/MindFlow-Pablo-Gutierrez>.

Este repositorio contiene tanto el front como el back. El front se encuentra en la carpeta front de la raíz del proyecto.

## 7. APLICACIÓN EN EL AMBITO EMPRESARIAL

### 7.1. Idea de negocio

**MindFlow** es una aplicación web de productividad y gestión de tareas diseñada para usuarios que buscan una solución para manejar tanto tareas cotidianas como proyectos profesionales. La aplicación ofrece una experiencia de usuario optimizada con interfaces modernas y atractivas, siguiendo los estándares de diseño web actual. Sus funcionalidades principales incluyen:

1. **Todo List:** Permite crear listas de tareas con estados (Todo, Doing, Done), ofreciendo un manejo dinámico. Las tareas pueden ser priorizadas, descritas brevemente y asociadas a una fecha de caducidad.
2. **Calendar Task:** Facilita la organización de tareas dentro de un calendario semanal (lunes a viernes), categorizadas por tags de color (Personal, Work, Study, Other).
3. **Github Data Manager:** Integra la gestión de proyectos de Github (Una herramienta que permite guardar repositorios de código y compartirlo con gente), permitiendo ver y manipular información de repositorios sin salir de MindFlow. Los usuarios pueden buscar y filtrar proyectos, y realizar acciones.
4. **Notes Editor:** Ofrece un espacio para crear y editar notas con herramientas de formato básico. Similar a la funcionalidad de Word, pero más simple.
5. **Statistics:** Proporciona gráficos y datos sobre la creación y cierre de tareas en Todo List y Calendar Task, disponible solo para usuarios premium o admin. La suscripción cuesta \$1.99 al mes.

**El valor añadido de MindFlow** se basa en la facilidad de manejo de proyectos y tareas personales y el conjunto de varias herramientas que podemos encontrar en aplicaciones distintas lo que nos permite no tener que disponer de otras aplicaciones y hacer todo aquí, todo por un precio muy asequible.

Además de su enfoque moderno y que asegura una gran experiencia de usuario.

## **Análisis PEST:**

### **Político:**

- Normativas de protección de datos: Asegurarse de cumplir con GDPR y otras regulaciones de privacidad.
- Políticas de ciberseguridad: Adaptarse a las normativas sobre seguridad de la información.

### **Económico:**

- Competencia en precios: Necesidad de establecer precios competitivos para suscripciones premium.

### **Social:**

- Tendencias en teletrabajo: Aumento de la demanda de herramientas de gestión de tareas y proyectos.
- Preferencias del usuario: Aceptación de interfaces modernas y atractivas.

### **Tecnológico:**

- Avances en tecnologías web: Uso de las últimas tecnologías para mejorar la experiencia de usuario.
- Integración con APIs: Capacidad para integrar nuevas funcionalidades basadas en otras plataformas (ej. Github)

## **Análisis DAFO**

### **Debilidades:**

- Falta de adaptación para el uso de muchos usuarios simultáneamente
- Opción premium necesitaría tener más opciones en un uso real por usuarios. Debe ofrecer más.

### **Amenazas:**

- Aplicaciones existentes que realizan funciones comunes o parecidas con presupuestos multimillonarios. Por ejemplo: Word, Notion, Google Calendar.
- Cambios en las políticas de APIs de terceros (ej. Github) que puedan afectar la funcionalidad.

### Fortalezas:

- Interfaz moderna y centrada en la experiencia de usuario.
- Integración de múltiples funcionalidades (Todo List, Calendar Task, Github Data Manager, Notes Editor) en una sola plataforma.
- Opciones de suscripción premium accesibles y funcionales.

### Oportunidades:

- Crecimiento del mercado de herramientas de productividad debido al aumento del trabajo remoto.

## 7.2. Planificación de necesidades de RRHH

### Recursos materiales:

#### Infraestructura Tecnológica:

- **Servidores:** Es Esencial el uso de servidores, actualmente usa estos, pero suscripciones reducidas. Para la publicación de la aplicación a los usuarios se alquilarían servidores más caros y potentes.

#### Recursos personales:

- **Equipo de desarrollo:** La aplicación está realizada exclusivamente por mí, para un entorno de venta real se necesitaría de más personal. Tanto más desarrolladores como personal de ciberseguridad, despliegue, bases de datos, marketing, recursos humanos, etc. Depende del alcance de la aplicación el número de personal.

## 7.3. Planificación de necesidades de producción

Para un entorno real de venta por medio de la aplicación, el **organigrama** podría ser este para tener el producto establecido en el mercado:

#### Dirección General

- CEO/director general: Responsable de la visión y estrategia general de la empresa, toma de decisiones clave y supervisión de todos los departamentos. De esto me encargaría yo, al haber creado la aplicación.

#### Desarrollo de Producto

- CTO/director de Tecnología: Supervisa el desarrollo técnico, define la arquitectura y asegura la calidad del producto.
- Equipo de Desarrollo:
  - Desarrolladores Front-End: 3 personas
  - Desarrolladores Back-End: 3 personas
  - Ingenieros de DevOps: 2 personas

#### Diseño y Experiencia de Usuario

- Director de Diseño: Responsable de la estrategia de diseño y la experiencia del usuario.
- Diseñadores UX/UI: 2 personas

#### Calidad y Pruebas

- QA/Testers: 2 personas

#### Gestión de Proyectos

- Gerente de Proyecto: Coordina las tareas del equipo, mantiene los plazos y supervisa el progreso del proyecto.

#### Marketing y Ventas

- Director de Marketing: Responsable de las estrategias de marketing y ventas.
- Especialistas en Marketing: 2 personas
- Equipo de Ventas: 2 personas

#### Atención al Cliente

- Gerente de Soporte: Supervisa las actividades de soporte al cliente.
- Agentes de Soporte: 2 personas
- Administración y Finanzas

Director Financiero: Responsable de la gestión financiera y contable.

Contadores: 1 persona

Recursos Humanos: 1 persona

Hasta llegar a este punto se empezará con los empleados justos y luego se irá escalando según se obtengan los primeros beneficios.

## Obligaciones Fiscales y Laborales

### Obligaciones Fiscales:

- Registro de la Empresa: Inscribir la empresa en el registro mercantil y obtener un número de identificación fiscal (NIF).
- Impuestos Sobre la Renta: Declarar y pagar el impuesto sobre sociedades. Presentación del Impuesto de Sociedades (Impuesto sobre la Renta de las Personas Jurídicas).
- IVA: Inscripción en el registro de IVA, declaración y pago trimestral del IVA.
- Retenciones e Ingresos a Cuenta: Retenciones en los pagos a trabajadores y contratistas y su ingreso a la Hacienda Pública.

### Obligaciones Laborales:

- Contratos Laborales: Cumplimiento de la legislación laboral vigente.
- Alta en la Seguridad Social: Inscripción de la empresa y de todos los empleados en la Seguridad Social.
- Prevención de Riesgos Laborales: Implementación de un plan de prevención de riesgos laborales y realización de formaciones obligatorias.
- Registro de Jornada Laboral: Mantenimiento de un registro diario de la jornada laboral de cada empleado.
- Políticas de Igualdad
- Formación y Desarrollo: Ofrecer programas de formación y desarrollo.
- Si se tienen oficinas asegurar el cumplimiento de la ley para cada característica. Por ejemplo: Iluminación, temperatura, distancia, material ...

## 7.4. Planificación de la Prevención de Riesgos Laborales

Los posibles riesgos asociados a la ejecución de un proyecto así pueden ser:

- Riesgos ergonómicos: Los trabajadores pueden sufrir dolores o fatiga al estar prologadas horas frente al ordenador
- Riesgos Psicosociales: Estrés laboral debido a plazos ajustados, alta carga de trabajo y necesidad de constante innovación.
- Riesgos de seguridad Informática: Posibles brechas de seguridad que pueden comprometer la información y datos sensibles del proyecto.

- Riesgos físicos: Accidentes en el lugar de trabajo, como caídas, golpes o incendios.

#### **Medidas de prevención:**

- Evaluación Ergonómica: Realizar evaluaciones periódicas de las estaciones de trabajo para asegurar una correcta postura.
- Mobiliario Ergonómico
- Gestión de Carga de trabajo: Distribuir la carga equitativamente sin saturar a los trabajadores
- Flexibilidad laboral para mejorar la calidad laboral y evitar estrés y problemas asociados
- Ciberseguridad para prevenir riesgos informáticos
- Plan de emergencia y capacitación en primeros auxilios para prevenir riesgos físicos

### **7.5. Previsiones de inversión y financiación**

#### **Gastos:**

- Infraestructura tecnológica:
  - o Servidores y Hosting en Cloud
  - o Ordenadores y material para empleados
  - o Licencias de software y suscripciones
  - o Herramientas de comunicación
- Salarios
- Otros gastos
  - o Espacio de oficina (Si se usa)
  - o Marketing y publicidad
  - o Mantenimiento material
  - o Gastos legales

#### **Financiación:**

Los gastos iniciales serán altos ya que se necesita de personal y servidores. Para financiarme tengo en cuenta las siguientes financiaciones:

- Capital Propio
- Inversión de socios

- Presentar idea a inversores y business angels

## 7.6. Estudio de mercado y marketing operativo

### Estrategia de Precios:

**Modelo Freemium:** La aplicación será gratuita con funcionalidades básicas disponibles para todos los usuarios.

**Suscripción Premium:** \$1.99/mes para acceder a funciones avanzadas, como estadísticas detalladas y características adicionales.

### Justificación del Precio:

**Accesibilidad:** El precio de \$1.99/mes es accesible para la mayoría de los usuarios, tanto individuales como profesionales.

**Competitividad:** Comparado con otras aplicaciones de productividad, el precio es competitivo y atractivo.

**Valor Añadido:** Las funcionalidades premium, como las estadísticas y la integración con Github, justifican el costo adicional.

### Estrategia de Comunicación:

**Redes Sociales:** Campañas en Facebook, Twitter, LinkedIn, y Instagram para alcanzar a diferentes segmentos de usuarios. Ofrecer colaboraciones a influencers para recomendar el producto.

**Marketing de Contenidos:** Creación de blogs, tutoriales y charlas sobre el producto para alcanzar otro tipo de público y llamarles la atención

**Email Marketing:** Campañas de correo electrónico para convertir usuarios gratuitos en suscriptores premium.



## **Distribución y servicio Postventa**

### **Distribución**

- La aplicación está disponible a través de internet para cualquier navegador y tipo de dispositivo
- Optimizado para pantallas grandes pero accesible en dispositivos móviles

### **Servicio Postventa:**

- Soporte 24/7 a través de redes sociales y correo electrónico
- Documentación del producto. Guías de usuario y tutoriales.
- Encuestas de Satisfacción
- Actualizaciones frecuentes
- Se valora el hacer pruebas gratuitas de premium