



Proyecto Final DAW - Organify

Desarrollado por **Miguel Hurtado**

Índice

Índice	2
1. Abstract	3
Español	3
Inglés	4
2. Justificación	4
Características Generales	4
Restricciones Generales	5
Aspectos a Cubrir y No Cubiertos	5
Aspectos a Cubrir:	5
Aspectos No Cubiertos:	5
Estudio de las Prestaciones de la Herramienta Frente a Otras Existentes en la Misma Categoría	6
3. Justificación de la Tecnología Empleada	6
Descripción y Justificación de las Tecnologías	7
Vue.js	7
Laravel	7
Tailwind CSS	7
Inertia.js	8
MySQL	8
Plesk	8
4. Requerimientos hardware y software:	9
Cliente (Frontend)	9
Servidor (Backend)	9
Requerimientos Combinados para Desarrollo	10
Requerimientos Combinados para Producción	11
5. Análisis y diseño	12
Diagrama de casos de uso	12
Descripción de la base de datos	13
6. Implementación	13
Hojas de Estilo	13
Plantillas	14
Formularios	14
Funciones de Envío de Datos	14
Conexión, Consultas y Selección de la BD	14
Uso de Fichero de Configuración	14
7. Evaluación y prueba	14
Testeo	14
Verificación de campos y la realimentación al usuario	15
8. Manual de estilos	15
Sketches	15

Criterios de accesibilidad	18
Criterios de usabilidad	19
Tipografía	19
Mapa de colores	19
Dispositivos/vistas	20
9. Software utilizado	20
10. Mejoras posibles y aportaciones	20



1. Abstract

Español

Descripción General:

Organify es una aplicación de gestión y organización de tareas diseñada tanto para uso personal como en entornos laborales. Facilita la creación, asignación y seguimiento de tareas diarias, ofreciendo herramientas de colaboración y planificación que optimizan la productividad de los equipos de trabajo. Con Organify, los usuarios pueden crear y gestionar tareas personales, unirse a grupos de trabajo, sugerir tareas y autoasignarse tareas del grupo. Los managers tienen funciones adicionales para gestionar usuarios, asignar horas de trabajo y aprobar tareas propuestas por el equipo.

Alcance del Proyecto:

- Creación y gestión de tareas personales.
- Asignación de tareas a días/horas específicas.
- Creación y gestión de grupos de trabajo.
- Unirse a grupos de trabajo públicos o solicitar acceso a grupos privados.
- Sugerencia y autoasignación de tareas dentro de grupos.
- Gestión de prioridades y estados de tareas.
- Funcionalidades específicas para managers, incluyendo la gestión de usuarios y aprobación de tareas.

Inglés

General Description:

Organify is a task management and organization application designed for both personal and work environments. It facilitates the creation, assignment, and tracking of daily tasks, offering collaboration and planning tools that optimize team productivity. With Organify, users can create and manage personal tasks, join work groups, suggest tasks, and self-assign tasks from the group board. Managers have additional functionalities to manage users, assign work hours, and approve tasks proposed by the team.

Project Scope:

- Creation and management of personal tasks.
- Assignment of tasks to specific days/hours.
- Creation and management of work groups.
- Joining public work groups or requesting access to private groups.
- Suggestion and self-assignment of tasks within groups.
- Management of task priorities and statuses.

- Specific functionalities for managers, including user management and task approval.

2. Justificación

Características Generales

Organify es una aplicación que ofrece una solución integral para la gestión de tareas tanto a nivel personal como laboral. Sus características clave incluyen:

- **Interfaz Intuitiva:** Diseño user-friendly que facilita la creación y gestión de tareas.
- **Creación y Asignación de Tareas:** Permite a los usuarios crear tareas, asignarles prioridades y fechas, y asignarlas a días y horas específicos.
- **Gestión de Grupos de Trabajo:** Los usuarios pueden crear y unirse a grupos de trabajo, gestionar miembros y colaborar en proyectos.
- **Sugerencia y Autoasignación de Tareas:** Los usuarios pueden sugerir tareas dentro del grupo y autoasignarse tareas disponibles.
- **Gestión de Prioridades y Estados de Tareas:** Cambiar prioridades y clasificar tareas como "to do", "doing" o "done".
- **Funcionalidades de Manager:** Gestión de usuarios, asignación de horas de trabajo, y aprobación de tareas propuestas por el equipo.
- **-Gestión de Archivos:** Posibilidad de subir, compartir y gestionar archivos adjuntos a las tareas.

Restricciones Generales

- **Conectividad:** Requiere conexión a internet para acceder y sincronizar datos.
- **Compatibilidad:** La aplicación está diseñada para navegadores modernos; versiones antiguas pueden no ser compatibles.

Aspectos a Cubrir y No Cubiertos

Aspectos a Cubrir:

- Gestión completa de tareas personales y laborales.
- Creación y gestión de grupos de trabajo.
- Herramientas de colaboración y planificación.
- Interfaz intuitiva y fácil de usar.
- Funcionalidades específicas para managers.
- Gestión de archivos, incluyendo subida, compartición y organización de documentos.

Aspectos No Cubiertos:

- No incluye integraciones con otras aplicaciones de productividad (e.g., Google Calendar, Microsoft Teams).
- No ofrece funcionalidades avanzadas de análisis de productividad o reportes detallados.

Estudio de las Prestaciones de la Herramienta Frente a Otras Existentes en la Misma Categoría

Comparación con Otras Herramientas de Gestión de Tareas:

1. Trello:

- **Ventajas de Trello:** Popular, amplia adopción, muchas integraciones con otras herramientas.
- **Ventajas de Organify:** Funcionalidades específicas para managers, gestión de grupos de trabajo dentro de la misma plataforma.

2. Asana:

- **Ventajas de Asana:** Herramientas avanzadas de seguimiento de proyectos, análisis de productividad.
- **Ventajas de Organify:** Interfaz más intuitiva para usuarios no técnicos, integración más fluida entre frontend y backend gracias a Inertia.js, gestión de archivos integrada.

3. Microsoft To Do:

- **Ventajas de Microsoft To Do:** Integración con el ecosistema Microsoft, simplicidad en la gestión de tareas personales.

- Ventajas de Organify: Mejores capacidades de colaboración en equipo, funcionalidades de manager, planificación detallada de tareas, gestión de archivos.

3. Justificación de la Tecnología Empleada

El desarrollo de la aplicación **Organify** está basado en una combinación de tecnologías modernas y robustas que garantizan una experiencia de usuario fluida y un backend eficiente y escalable. A continuación, se enumeran y describen las tecnologías utilizadas, así como la justificación de su elección basada en un proceso de investigación exhaustivo.

Descripción y Justificación de las Tecnologías

Vue.js

- **Descripción:** Vue.js es un framework progresivo de JavaScript utilizado para construir interfaces de usuario. Es conocido por su enfoque flexible y su capacidad para integrarse fácilmente con otras bibliotecas o proyectos existentes.
- **Justificación:** Vue.js fue elegido por su curva de aprendizaje suave y su potente ecosistema. Ofrece reactividad y componentes reutilizables, lo que facilita el desarrollo de una interfaz de usuario dinámica y responsiva. Además, su comunidad activa y extensa documentación lo convierten en una opción ideal para este proyecto.

Laravel

- **Descripción:** Laravel es un framework de PHP que proporciona una sintaxis elegante y expresiva para el desarrollo de aplicaciones web. Incluye características como autenticación, manejo de sesiones, y enrutamiento, entre otras.
- **Justificación:** Laravel fue seleccionado por su robustez y facilidad de uso. Proporciona una estructura clara y funcionalidades integradas que aceleran el desarrollo del backend. Su ORM Eloquent facilita la interacción con bases de

datos y su capacidad para manejar la autenticación y autorización es crucial para las funcionalidades de gestión de usuarios y tareas.

Tailwind CSS

- **Descripción:** Tailwind CSS es un framework de CSS utilitario que permite construir interfaces de usuario personalizadas directamente en el HTML, mediante clases prediseñadas que se pueden combinar para diseñar cualquier layout.
- **Justificación:** Tailwind CSS fue elegido por su flexibilidad y capacidad para acelerar el desarrollo del frontend. A diferencia de otros frameworks CSS, Tailwind permite una personalización más granular sin necesidad de escribir CSS adicional. Esto facilita el mantenimiento del código y asegura consistencia en el diseño.

Inertia.js

- **Descripción:** Inertia.js actúa como un puente entre el frontend y el backend, permitiendo construir aplicaciones SPA (Single Page Application) utilizando frameworks de frontend como Vue.js y backend como Laravel sin necesidad de crear una API separada.
- **Justificación:** Inertia.js fue elegido para simplificar el desarrollo y la integración entre Vue.js y Laravel. Permite desarrollar aplicaciones modernas y dinámicas sin la complejidad de gestionar una API separada, lo que mejora la productividad y reduce el tiempo de desarrollo.

MySQL

- **Descripción:** MySQL es un sistema de gestión de bases de datos relacional, conocido por su fiabilidad y rendimiento.
- **Justificación:** MySQL fue seleccionado debido a su eficiencia y robustez en el manejo de grandes volúmenes de datos. Su amplia adopción y soporte en la comunidad de desarrollo también lo convierten en una opción confiable para almacenar y gestionar los datos de la aplicación.

Plesk

- **Descripción:** Plesk es una plataforma de gestión de hosting que permite administrar servidores web y dominios de manera intuitiva y eficiente.
- **Justificación:** Plesk fue elegido para la puesta en producción debido a su interfaz amigable y sus capacidades de gestión de servidores. Permite administrar el servidor de manera sencilla, configurar dominios, manejar bases de datos y realizar tareas de mantenimiento. La utilización de Plesk con el dominio **organify.es** asegura una administración centralizada y eficiente del entorno de producción.

4. Requerimientos hardware y software:

Para implementar y ejecutar el proyecto se necesita:

Cliente (Frontend)

1. Vue.js

- **Hardware:**
 - Procesador: Cualquier procesador moderno (Intel i3/i5/i7, AMD Ryzen).
 - Memoria RAM: Mínimo 4 GB (8 GB recomendado para un desarrollo más fluido).
 - Almacenamiento: 2 GB de espacio libre (para Node.js, npm/yarn, y dependencias del proyecto).
- **Software:**
 - Sistema Operativo: Windows, macOS, Linux.
 - Navegador Web: Últimas versiones de Google Chrome, Firefox, Safari o Edge.
 - Node.js: Versión 12.x o superior.
 - npm (incluido con Node.js) o Yarn para la gestión de paquetes.

2. Tailwind CSS

- **Hardware:**
 - Requisitos similares a los de Vue.js, ya que se integra en el mismo entorno de desarrollo.
- **Software:**
 - Node.js y npm/yarn (ya que Tailwind se instala como un paquete npm).

3. Inertia.js

- **Hardware:**
 - Requisitos similares a los de Vue.js, ya que se utiliza junto con Vue.js.
- **Software:**
 - Node.js y npm/yarn.
 - Integración con Vue.js y Laravel, por lo tanto, ambos deben estar instalados.

Servidor (Backend)

1. Laravel

- **Hardware:**
 - Procesador: Cualquier procesador moderno con soporte para múltiples núcleos.
 - Memoria RAM: Mínimo 2 GB (4 GB recomendado para un entorno de desarrollo).
 - Almacenamiento: Espacio suficiente para el código base, dependencias, y base de datos (mínimo 10 GB).
- **Software:**
 - Sistema Operativo: Ubuntu 18.04 o superior, CentOS 7 o superior, Windows Server 2016 o superior.
 - PHP: Versión 7.4 o superior.
 - Composer: Para la gestión de dependencias PHP.
 - Servidor web: Apache 2.4 o Nginx 1.17+.

2. MySQL

- **Hardware:**
 - Procesador: CPU con al menos 2 núcleos.
 - Memoria RAM: Mínimo 2 GB (8 GB o más recomendado para bases de datos grandes).
 - Almacenamiento: Espacio suficiente para la base de datos y los registros de transacciones (depende del tamaño de los datos, mínimo 20 GB).
- **Software:**
 - Sistema Operativo: Ubuntu, CentOS, Windows Server (compatible con la versión de MySQL que se desee instalar).
 - MySQL: Versión 5.7 o superior (preferiblemente la última versión estable).

3. Plesk

- **Hardware:**
 - Procesador: CPU de 1 GHz o superior.
 - Memoria RAM: Mínimo 1 GB (2 GB recomendado para un rendimiento óptimo).
 - Almacenamiento: Mínimo 10 GB (más espacio para los sitios web y bases de datos gestionados).

- **Software:**
 - Sistema Operativo: Varias versiones de Linux (Ubuntu, CentOS, Debian) y Windows Server son soportadas.
 - Dependencias adicionales: Servidor web (Apache/Nginx), PHP, MySQL/MariaDB.

Requerimientos Combinados para Desarrollo

Para un entorno de desarrollo que combina todas estas tecnologías:

- **Hardware:**
 - Procesador: Intel i5/i7 o AMD Ryzen 5/7.
 - Memoria RAM: Mínimo 8 GB (16 GB recomendado para múltiples tareas y servicios).
 - Almacenamiento: Mínimo 50 GB de espacio libre (para código, dependencias, bases de datos locales, etc.).
- **Software:**
 - Sistema Operativo: Windows 10, macOS, Ubuntu 18.04 o superior.
 - Node.js: Versión 12.x o superior.
 - PHP: Versión 7.4 o superior.
 - Composer.
 - npm/yarn.
 - MySQL: Versión 5.7 o superior.
 - Servidor web local: Apache o Nginx.

Requerimientos Combinados para Producción

Para un servidor de producción:

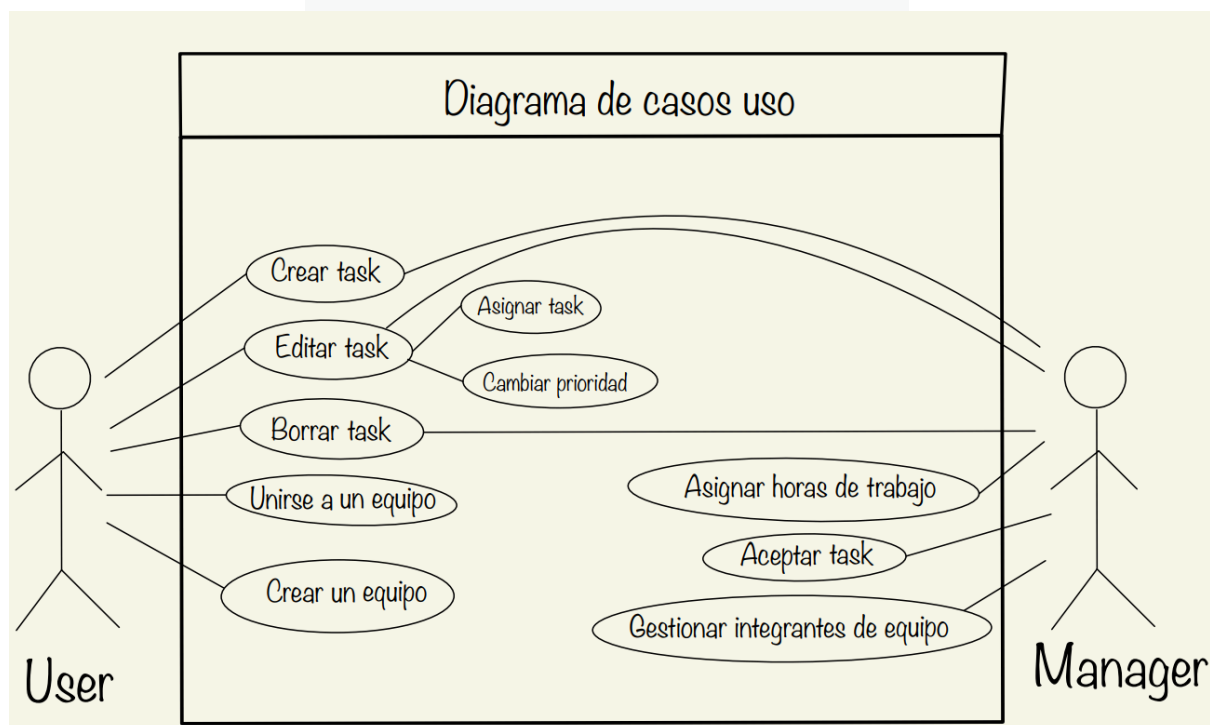
- **Hardware:**
 - Procesador: CPU con múltiples núcleos, preferiblemente Xeon o similar.
 - Memoria RAM: Mínimo 8 GB (16 GB o más recomendado).
 - Almacenamiento: SSD con espacio suficiente para la base de datos, código, y backups (mínimo 100 GB).
- **Software:**

- Sistema Operativo: Ubuntu 18.04 o superior, CentOS 7 o superior, Windows Server 2016 o superior.
- PHP: Versión 7.4 o superior.
- Composer.
- MySQL: Versión 5.7 o superior.
- Servidor web: Apache 2.4 o Nginx 1.17+.
- Plesk: Instalado y configurado para la gestión del servidor y sitios web.

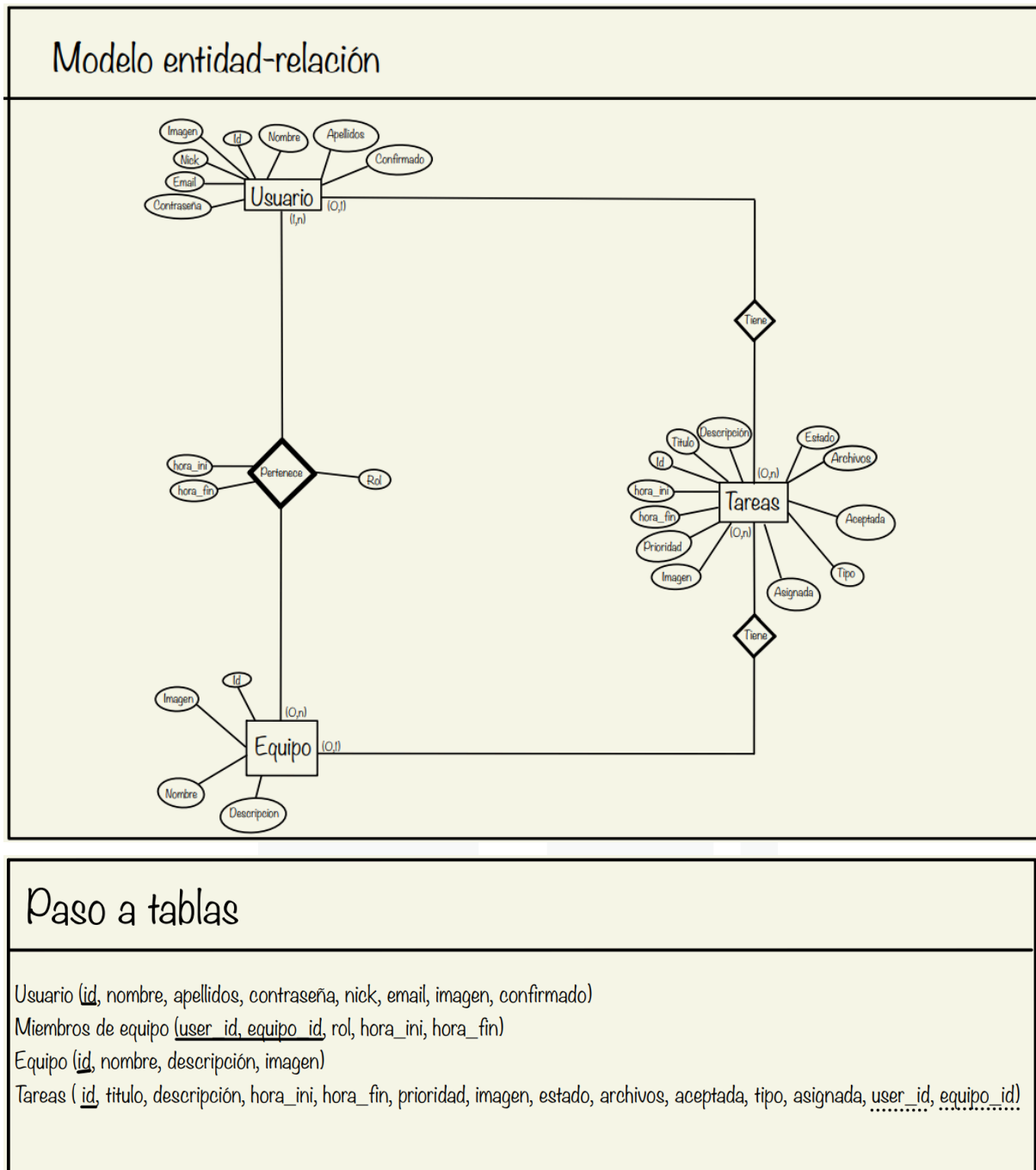
Estos requerimientos aseguran que las tecnologías mencionadas funcionen de manera eficiente tanto en entornos de desarrollo como en producción.

5. Análisis y diseño

Diagrama de casos de uso



Descripción de la base de datos



6. Implementación

Hojas de Estilo

- **Tailwind CSS:** Framework de utilidades CSS que emplea clases predefinidas para estilizar elementos HTML de manera eficiente.

Plantillas

- **Blade:** Estructura la aplicación y actúa como contenedor de componentes Vue.
- **Vue.js:** Se integra en Blade para definir componentes reutilizables que contienen su propia plantilla HTML, lógica JavaScript y estilos CSS.

Formularios

- **Vue.js & Laravel:** Se utilizan formularios en componentes Vue.js para capturar datos de entrada del usuario. Laravel maneja la recepción y procesamiento de estos datos en el backend, utilizando los métodos definidos en los controladores.

Funciones de Envío de Datos

- **Inertia.js:** Facilita la comunicación entre el frontend y el backend mediante peticiones sin recarga de página. Permite el envío de datos estructurados y la actualización dinámica de la interfaz de usuario.

Conexión, Consultas y Selección de la BD

- **Laravel & MySQL:** Laravel gestiona la conexión con la base de datos MySQL utilizando Eloquent ORM para realizar consultas y operaciones CRUD de manera eficiente y segura.

Uso de Fichero de Configuración

- **Laravel:** Se utiliza el archivo de configuración `.env` para definir variables de entorno y configuraciones específicas del entorno de desarrollo o producción.

7. Evaluación y prueba

Testeo

- **Pruebas Funcionales:** Verificación completa de todas las funcionalidades como.
- **Pruebas de Usabilidad e Interfaz de Usuario (UI):** Recopilación de feedback de usuarios para mejorar la navegación y experiencia.

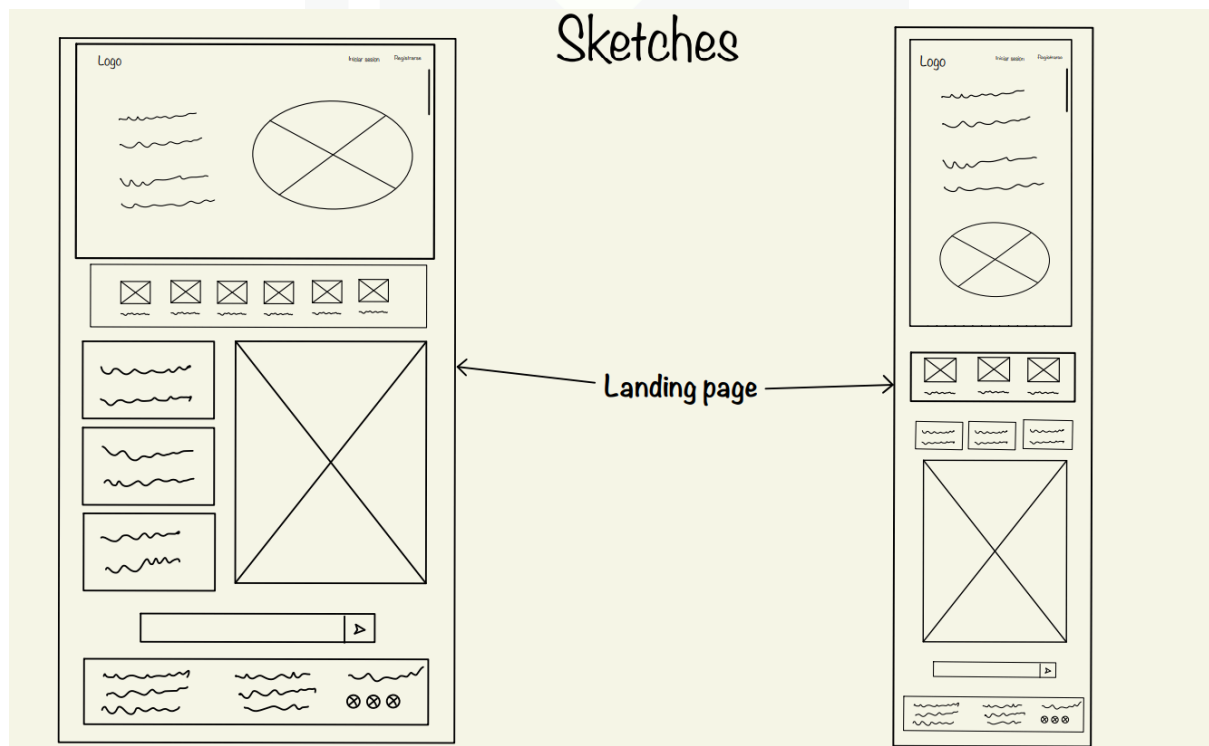
- **Pruebas de Rendimiento:** Evaluación a través de **PageSpeed Insights** para verificar y mejorar el rendimiento de la Web.
- **Pruebas de DB:** Se han utilizado **seeders** y **faker** para la inserción de datos falsos y comprobar el correcto funcionamiento de la Base de Datos

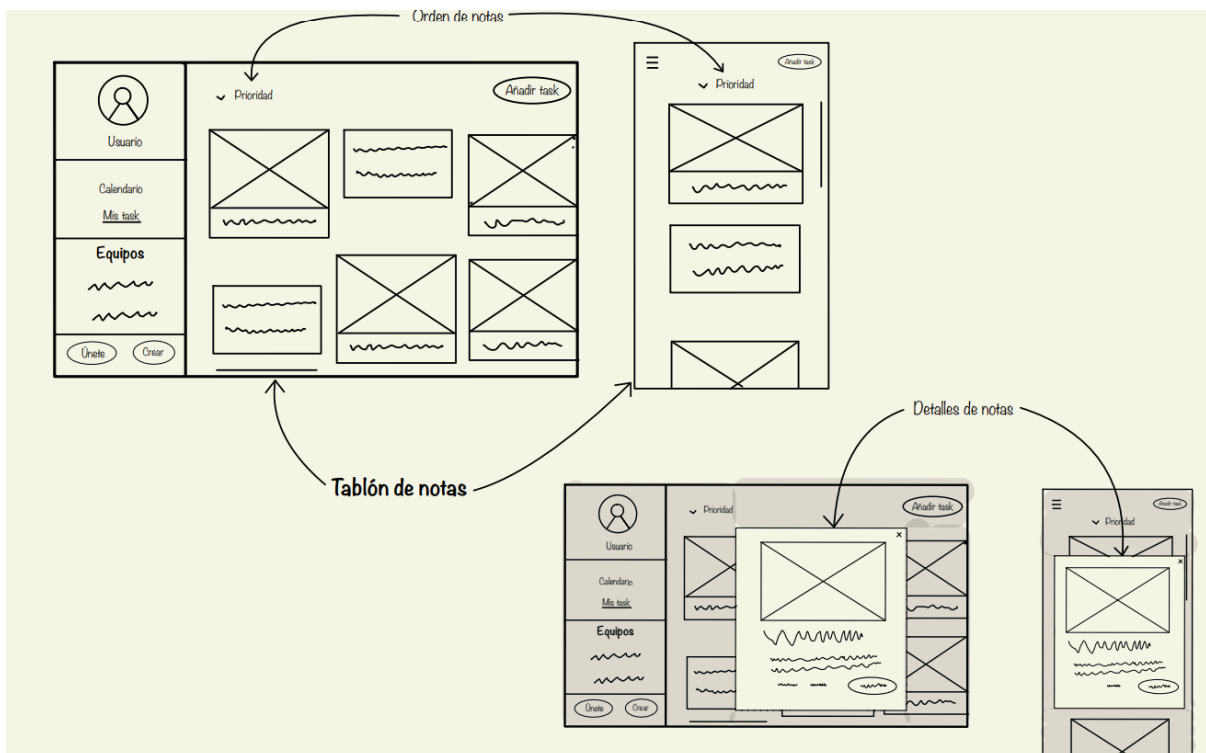
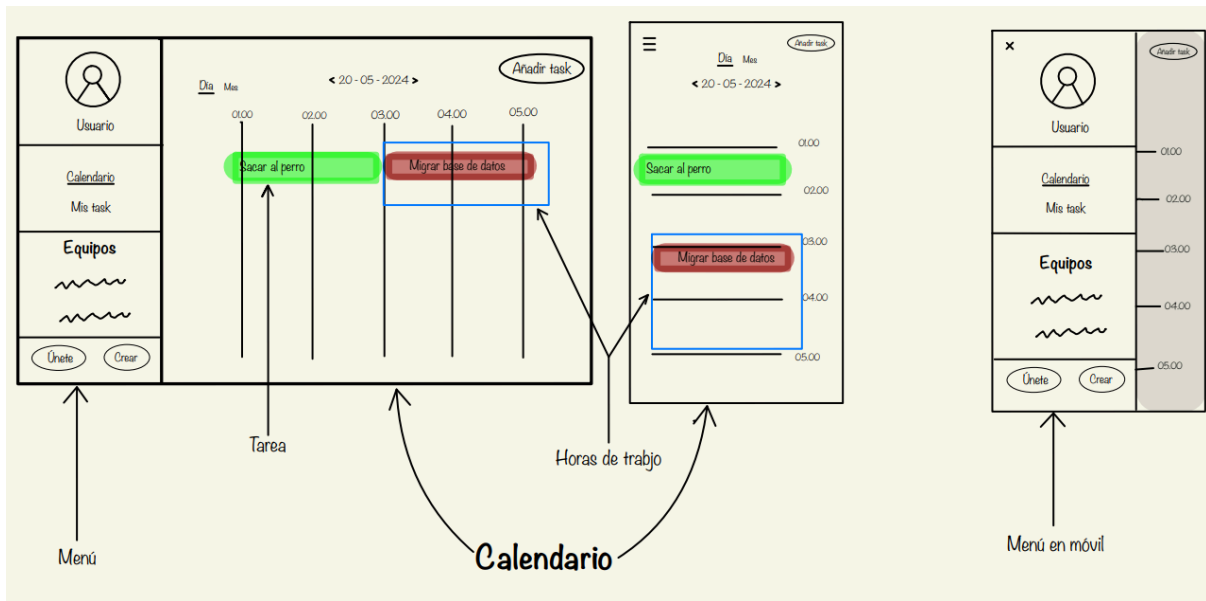
Verificación de campos y la realimentación al usuario

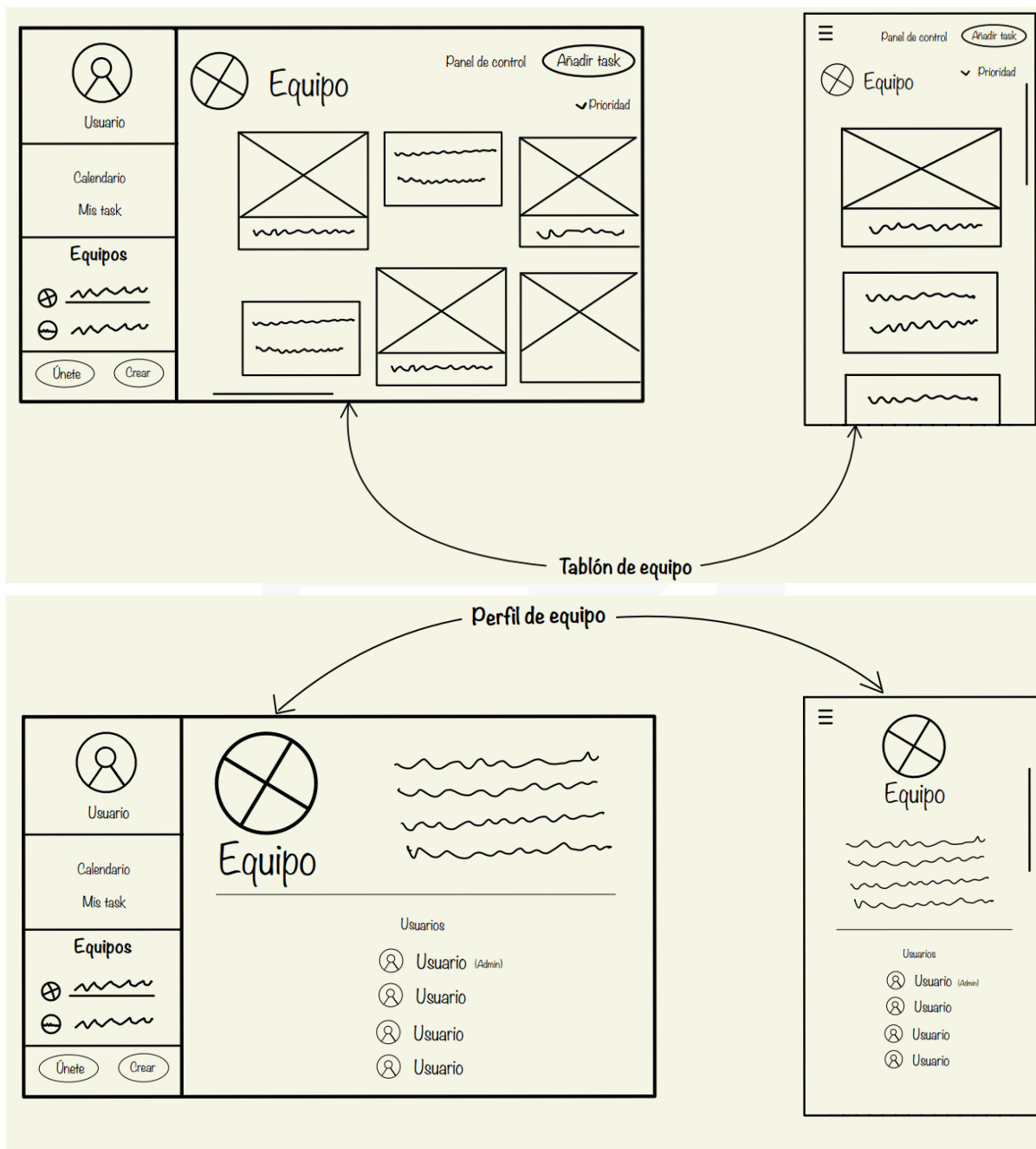
- **Verificación de Campos:** Validación de formatos y requisitos de datos en todos los formularios tanto en el frontend como en el backend.
- **Realimentación al Usuario:** Mensajes claros y precisos para guiar al usuario durante la interacción con la aplicación.

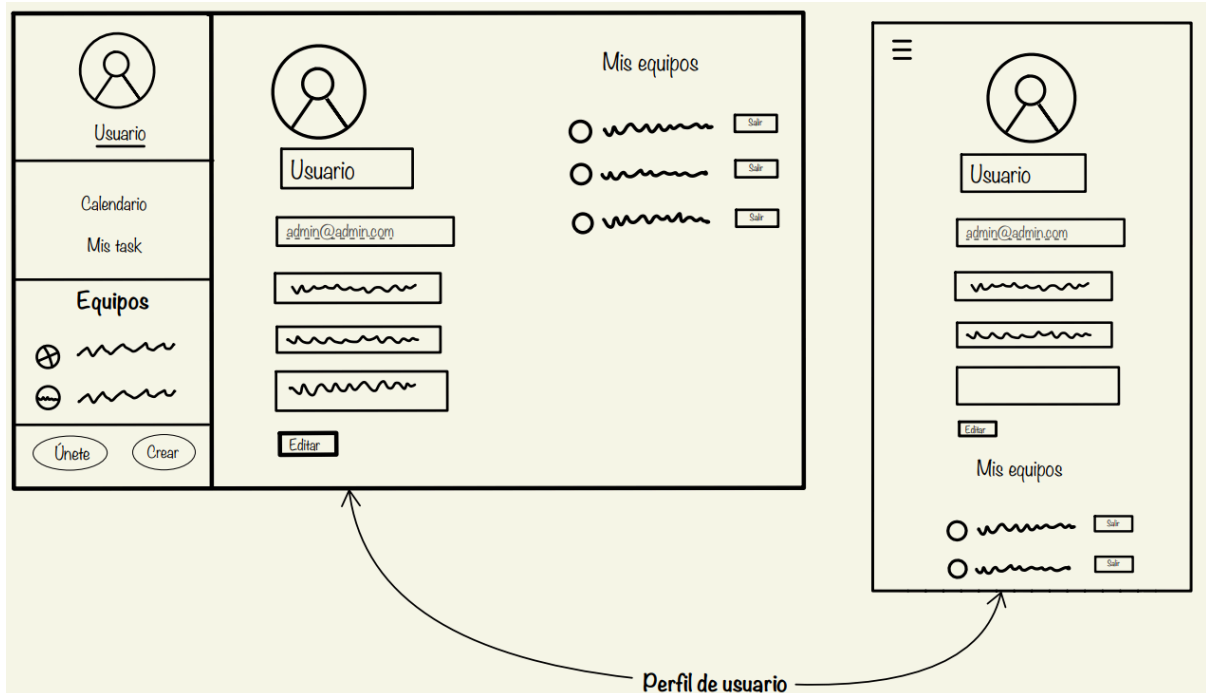
8. Manual de estilos

Sketches









Criterios de accesibilidad

He diseñado **Organify** cumpliendo los estándares de accesibilidad WCAG 2.1 para garantizar que todos los usuarios, incluidos aquellos con diversas capacidades, puedan interactuar eficazmente con la aplicación.

He implementado:

- **Percepción:** Me he asegurado que toda la información en el sitio web sea clara y legible para todos los usuarios, independientemente de sus capacidades visuales. Esto incluye el uso de colores y contrastes adecuados, así como el ajuste del tamaño y estilo de la fuente para mejorar la legibilidad según las preferencias del usuario.
- **Operabilidad:** El sitio web facilita la navegación y la interacción fluida con diferentes dispositivos de entrada, como teclados y dispositivos móviles. La interfaz ha sido optimizada para garantizar una experiencia de usuario consistente y accesible en diversas plataformas, asegurando que todos los usuarios puedan acceder y utilizar el contenido sin dificultades operativa
- **Comprensión:** He establecido una estructura de contenido coherente y predecible en todo el sitio, facilitando a los usuarios la localización y navegación eficiente a través de la información. Los mensajes de error son claros y están diseñados para proporcionar orientación contextual que ayuda a los usuarios a resolver problemas y completar tareas con éxito.

- **Robustez:** El desarrollo del sitio web lo he realizado con un enfoque en la robustez y la compatibilidad con una amplia gama de tecnologías y plataformas. El código ha sido validado para cumplir con estándares web actuales y futuros, asegurando que todas las funcionalidades del sitio sean accesibles y operativas en diferentes navegadores y versiones, proporcionando una experiencia confiable y consistente para todos los usuarios.

Criterios de usabilidad





La usabilidad de **Organify** la he centrado en proporcionar una experiencia de usuario eficiente y satisfactoria:

- **Facilidad de Uso:** Se ha diseñado el sitio web con un enfoque claro en la usabilidad, asegurando que los usuarios puedan navegar de manera intuitiva y realizar acciones fácilmente.
- **Eficiencia en la Interacción:** Las funciones y características del sitio web se han organizado de manera que los usuarios puedan completar tareas de manera eficiente y con el menor número de pasos posibles.
- **Aprendizaje y Adaptación:** La estructura y el diseño del sitio web permiten a los usuarios aprender rápidamente cómo navegar y utilizar sus funciones, sin necesidad de instrucciones complicadas.
- **Satisfacción del Usuario:** Se ha priorizado la experiencia del usuario, asegurando que cada interacción con el sitio web sea satisfactoria y cumpla con las expectativas del usuario.

Tipografía

Tipografía	Uso	Tamaño
ui-monospace	Titulos y subtitulos	Títulos: 36px Subtítulos: 24px
ui-sans-serif	Texto y botones	Texto: 16px Botones: 18px

Mapa de colores

Color	Hexadecimal	RGB	Uso
	#1f2937	(31, 41, 55)	Color Principal de la web, usado en header, menús y pop ups
	#4ade80	(74, 222, 128)	Color secundario , usado para botones, detalles y texto destacado
	#ffffff	(255, 255, 255)	Usado en fondo de la página y texto sobre fondo oscuro
	#000000	(0,0,0)	Usado para texto sobre fondo blanco

Dispositivos/vistas

Organify se adapta perfectamente a cualquier dispositivo gracias a las funciones de responsividad de Tailwind CSS, asegurando una experiencia de usuario consistente y efectiva en todos los dispositivos.

9. Software utilizado

He utilizado los siguientes softwares para el desarrollo y despliegue de la web:

- **Visual Studio Code:** Para escribir, editar y depurar el código de la aplicación.
- **Plesk Obsidian:** Para la configuración y subida de archivos al servidor de hosting
- **Navegadores:** He utilizado distintos navegadores como Brave, Google Chrome, Safari y Firefox para asegurar la compatibilidad de mi página con todos ellos.
- **GitHub Desktop:** Para el control de versiones de mi proyecto y publicar mi código en Github.

10. Mejoras posibles y aportaciones

Para mejorar la web tengo pensado:

- **Limpieza y optimización del código** para mejorar el rendimiento y el tiempo de carga de la web.
- **Integrar Organify** con herramientas populares de productividad como Slack, Microsoft Teams o Google Workspace para mejorar la colaboración y la comunicación dentro de los equipos.
- Realizar una **revisión de diseño** centrada en el usuario para optimizar la experiencia de usuario (UX/UI), asegurando una navegación más intuitiva y una interfaz más atractiva.
- Permitir a los usuarios **personalizar la apariencia** de la interfaz de usuario (temas oscuros, colores personalizados) y configurar preferencias de visualización según sus necesidades individuales.
- **Habilitar funcionalidades offline** para permitir a los usuarios trabajar sin conexión y sincronizar automáticamente los datos cuando recuperen la conexión a internet.
- Integrar **herramientas de comunicación** como chat en vivo o comentarios directamente dentro de la aplicación para facilitar la colaboración entre equipos y la comunicación sobre tareas específicas.
- Permitir la **integración de calendarios compartidos** para mejorar la coordinación y la asignación eficiente de tareas dentro de los equipos de trabajo.