DOCUMENTACION DEL SISTEMA DE GESTIÓN DE TAREAS

|  |  |
| --- | --- |
| Nombre del proyecto: | Task Manager |
| Fecha: | 04 DE SEPTIEMBRE DEL 2024 |
| Versión: | 1.0 |
| Creado por: | Oscar Guillermo Sierra Lozano, Yeferson Esmid Heredia Perdomo, Karen Johana Caicedo Arias, Jhon Sebastián Molina Fierro y Nicolas Obregón Rojas |

Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor | Descripción de la versión | Fecha de realización |
| 1.0 | Yeferson Heredia | Creación del modelado de mokac de la navegación en la página del sistema | 04/ 09 /2024 |
| 1.2 | Oscar Guillermo | Lanzamiento de las cualidades básicas y descrecimiento de los requerimientos | 08 / 09 / 2024 |
| 1.3 | Karen Caicedo | Descripción del sistema y limitación a los dispositivos que se piensa dar soporte del sistema general | 10/ 09 / 2024 |

Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Aprobador | Versión aprobada | Firma | Fecha |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |



Historial de aprobaciones

|  |  |  |  |
| --- | --- | --- | --- |
| Historial de aprobaciones | | | |
| Revisor | Versión revisada | Firma | Fecha |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[Índice 2](#_Toc474654897)

[Requisitos funcionales 5](#_Toc1894774589)

[Requisitos no funcionales 8](#_Toc1472966256)

# Índice



1. Introducción
   1. Alcance del producto
   2. Valor del producto
   3. Público objetivo
   4. Uso previsto
   5. Descripción general

2 Requisitos funcionales

1. Requisitos de la interfaz externa
   1. Requisitos de la interfaz de usuarios
   2. Requisitos de la interfaz de hardware
   3. Requisitos de la interfaz de software
   4. Requisitos de la interfaz de comunicación
2. Requisitos no funcionales
   1. Seguridad
   2. Capacidad
   3. Compatibilidad
   4. Confiabilidad
   5. Escalabilidad
   6. Mantenibilidad
   7. Facilidad de uso
   8. Otros requisitos no funcionales

5 Definiciones y acrónimos

1. Introducción

Llevar documentado el manejo del proyecto con dichas actualizaciones para buscar su mejora con el equipo de trabajo implementando dichas las sugerencias con lo que se recomiendo por el equipo y el administrador del proyecto.

## Alcance del producto

* Gestionar tareas y manejar con completa libertad de edición de dichas tareas.
* Gestionar trabajos grupales.
* Manejo y control de datos para los trabajos creados.
* Privacidad y roles para cada usuario.

## Valor del producto

## Público objetivo

* Empresas Agrícolas.
* Grupos Empresariales.
* Instituciones educativas.
* Aulas virtuales.
* Grupos de marketing.
* Programadores.
  1. Uso previsto
* Gestión de tareas.
* Control de actividades.
* Seguimiento de cumplimiento de trabajos.

## Descripción general

Este software tiene como propósito hacer una gestión de tareas para sus usuarios que consiste en poder crear tareas, modificarlas, poder eliminarlas, visualizarlas y que el usuario pueda subirlas al aplicativo. Tiene la capacidad de iniciar sesión, registrase o iniciar una cuenta existente, además se pueden asignar roles para los usuarios para generar una jerarquización y esto conlleve a un orden de responsabilidad y trabajo en equipo. También tiene la función de generar recordatorios y opciones de categorización de trabajos. Además, el sistema ofrecerá reportes y análisis sobre el progreso de las tareas y permitirá la colaboración en tiempo real de varios usuarios.

1. Requisitos funcionales

**Requisitos de diseño:**

* Diseño de la interfaz de usuario (UI).
* Arquitectura del sistema.
* Diseño de base de datos.
* Diseño de flujo de trabajo.
* Diseño de la interfaz de Programación (API).

**Requisitos Gráficos:**

* Paleta de colores.
* Tipografía.
* Iconografía.
* Imágenes y Gráficos.
* Espaciado y Márgenes.
* Animaciones y Transiciones.

**Requisitos de Sistema Operativo:**

* Compatibilidad con sistemas operativos de escritorio (Windows, macOS, Linux).
* Compatibilidad con sistemas operativos móviles (IOS, Android).
* Compatibilidad con navegadores web.
* Requisitos de red.

**Limitaciones del Producto:**

* Escalabilidad limitada.
* Dependencia de conexión a internet.
* Compatibilidad con dispositivos antiguos.

1. Requisitos de la interfaz externa
   1. Requisitos de

Describe la lógica detrás de las interacciones entre los usuarios y el software (diseños de pantalla, guías de estilo, etc.).

la interfaz de usuario

CRUD

**1. Autenticación (Login y Registro):**

* **Usuario:** Ingresa credenciales o completa el formulario de registro.
* **Backend:** Verifica credenciales o valida y guarda los datos de registro en la base de datos.
* **Resultado:** Acceso autorizado o error mostrado en la interfaz.

**2. Creación y Gestión de Tareas:**

* **Usuario:** Crea, edita o elimina tareas desde la interfaz.
* **Backend:** Procesa los datos enviados, valida y guarda las tareas en la base de datos.
* **Resultado:** Actualización de tareas reflejada en tiempo real en la interfaz.

**3. Consulta de Tareas:**

* **Usuario:** Navega por las tareas, usando filtros o categorías.
* **Backend:** Recupera las tareas correspondientes desde la base de datos según los filtros.
* **Resultado:** Muestra las tareas organizadas en la interfaz.

**4. Cambio de Estado de Tareas:**

* **Usuario:** Marca tareas como completadas o cambia su estado.
* **Backend:** Actualiza el estado de la tarea en la base de datos.
* **Resultado:** Tarea actualizada reflejada en la interfaz.

**5. Eliminación de Tareas:**

* **Usuario:** Solicita eliminar una tarea.
* **Backend:** Borra la tarea en la base de datos tras confirmar permisos.
* **Resultado:** La tarea desaparece de la lista.

**6. Colaboración y Notificaciones:**

* **Usuario:** Asigna o colabora en tareas con otros usuarios.
* **Backend:** Envía notificaciones y actualizaciones en tiempo real.
* **Resultado:** Los usuarios reciben notificaciones de cambios o asignaciones.
  1. Requisitos de

Incluye las conexiones entre tu producto y otros componentes de software, como el marco de front-end y back-end,

las bibliotecas, etc.

la interfaz de software

* **Compatibilidad con navegadores web (para versiones basadas en web):**

Cualquier navegador en su versión más reciente y estable ya sean Google Chrome, Edge, Mozilla Firefox, Safari para una mejor compatibilidad en los sistemas operativos de Windows, macOS y Linux.

* **Requisitos de red:**

Conectividad: Conexión a internet de alta velocidad para garantizar una buena sincronización de datos y uso de servicios en la nube.

Protocolos de red: Soporte para HTTP/HTTPS, y WebSocket si se requiere si se requieren comunicaciones en tiempo real.

Configuraciones de firewall: Permitir conexiones salientes de puertos especificos utilizados por la aplicación, como el puerto 443 para HTPPS.

* 1. Requisitos de la interfaz de comunicación

Enumera los requerimientos de los programas de comunicación

que usará tu producto, como los emails o los formularios integrados.

* **Gmail:** Servicio de correo electrónico fácil de usar, con integración con Google Drive, Google Calendar y otras herramientas de Google.
* **Outlook:** Plataforma de correo electrónico de Microsoft que se integra con Office 365, OneDrive y Teams, ideal para trabajo colaborativo.
* **Google Forms:** Herramienta gratuita para crear formularios y encuestas, con integración directa a Google Sheets para análisis de datos.
* **Microsoft Forms:** Solución de formularios de Microsoft que permite crear encuestas y recopilar información, integrada con Excel y otras herramientas de Office.
* **WhatsApp:** Aplicación de mensajería instantánea ampliamente utilizada, que permite chats grupales, llamadas de voz y video, y compartir archivos fácilmente.

# Requisitos no funcionales



4

* 1. Seguridad

Incluye las normas de privacidad y protección de datos que deben cumplirse.

Al momento de la creación de los usuarios tiene se le otorga un rol que conlleve a unas limitaciones del sistema que maneja. Se limita el manejo de los usuarios y los datos enlazados a ellos mismos para mantener la privacidad.

* 1. Capacidad

Describe la demanda de almacenamiento actual y futura de tu software.

El sistema asegura una alta compatibilidad con los navegadores web más recientes y populares, garantizando un rendimiento óptimo en múltiples plataformas, incluyendo Windows, macOS, Linux, iOS y Android. Hemos definido una capacidad de red que exige una conexión de alta velocidad, soportando protocolos esenciales como HTTP/HTTPS y WebSocket para comunicaciones en tiempo real. Además, hemos priorizado la seguridad de las conexiones a través de la configuración de firewalls para conexiones seguras por el puerto 443. A nivel de diseño gráfico, el sistema maneja una interfaz de usuario intuitiva y visualmente atractiva, con una paleta de colores cuidadosamente seleccionada, iconografía clara y animaciones suaves que no afectan el rendimiento. Si bien la escalabilidad está limitada por la infraestructura, hemos asegurado que el sistema pueda manejar adecuadamente un volumen moderado de usuarios y que dependa de una conexión estable para su funcionamiento eficiente.

* 1. Compatibilidad

Enumera los requisitos mínimos de hardware para el software.

**Compatibilidad para sistemas operativos de escritorio:**

* **Windows:**

Requisitos mínimos: Procesador de 64 bits, 4 GB de RAM, 2 GB de espacio en disco.

* **MacOS:**

Requisitos mínimos: Procesador Intel o Apple M1, 4 GB de RAM, 2 GB en disco.

* **Linux:**

Requisitos mínimos: Procesador de 64 bits, 4 GB de RAM, 2 GB de espacio en disco.

**Compatibilidad para sistemas operativos móviles:**

* **IOS:**

Requisitos mínimos: iPhone XS o superior, 4 GB de RAM.

* **Android:**

Requisitos mínimos: Dispositivo con procesador Qualcomm Snapdragon 765G o equivalente, 4 GB de RAM.

* 1. Confiabilidad

Calcula cuál es el tiempo de fallo crítico esperado de tu producto con relación a un uso normal.

Dentro de la práctica prototipada para la evaluación del proyecto se utilizó un tiempo de 144 horas (6 días) dentro de un uso normal, teniendo un tiempo promedio entre fallos de λ=0.0069.

Tasa de fallos

Ahora, la tasa de media de tiempo entre fallos, analizando los tiempos y la cantidad de errores se llegó a la conclusión de 144.92 representa el tiempo promedio que esperas que pase del proyecto en uso normal antes de que ocurra un fallo crítico.

En el tiempo utilizado (2 semanas) se estima que el sistema se use en 12 horas durante 7 días a la semana. Dándonos un tiempo de 84 horas/semana, entonces sabiendo esto, se concluye que se puede esperar un fallo aproximadamente cada 1.72 semanas.

* 1. Escalabilidad

Calcula la cantidad máxima de trabajo con la que tu software puede funcionar según lo esperado.

Un software se puede evaluar y optimizar mediante un proceso de métricas y fórmulas. Primero, se mide el rendimiento, que se calcula de la siguiente manera:

Esta fórmula ayuda a determinar cuántas transacciones puede procesar el sistema en un período de tiempo determinado y proporciona una base para comprender su capacidad actual. Si medimos cuántas tareas pueden crearse, asignarse, actualizarse o completarse en un periodo determinado, Podríamos gestionar 500 tareas por minuto de forma eficiente, el rendimiento es de 500 tareas/minuto. Esto nos ayuda a saber la cantidad de trabajo que puede procesar la aplicación sin que se degrade el rendimiento.

que indica qué porcentaje de recursos del sistema (como CPU y memoria) se están utilizando. Si el servidor está usando el 80% de la CPU y la memoria cuando hay 100 usuarios activos gestionando tareas, significa que está cerca de los límites de capacidad y podría ser necesario añadir más recursos o servidores, para asegurar que el sistema no colapse bajo cargas mayores.

ya que un aumento en el tiempo de respuesta a medida que aumenta la carga de trabajo puede ser una señal de que el sistema necesita más capacidad. Si los usuarios notan que la aplicación se vuelve más lenta a medida que se incrementa el número de tareas o usuarios, esto indica que el sistema necesita mejoras de escalabilidad. Tenemos que Mantener un bajo tiempo de respuesta para ofrecer una experiencia fluida.

se puede utilizar para predecir el comportamiento del sistema bajo diferentes condiciones de carga y planificar mejoras. Esto nos permitirá estimar cuándo necesitaremos más servidores o aumentar la capacidad del sistema para garantizar que funcione correctamente con una carga mayor.

* 1. Mantenibilidad

Describe cómo se debe usar la integración continua para implementar funciones o reparar errores rápidamente.

El proyecto mejora significativamente mediante el uso de la Integración Continua (CI), ya que las soluciones se pueden identificar, corregir e implementar de manera rápida y eficiente. Siempre que hay un cambio en el código, CI ejecuta pruebas automatizadas que ayudan a detectar errores tempranamente. Cuando se detecta un problema, los desarrolladores pueden solucionarlo y volver a ejecutar el proceso de prueba comprometiéndose a garantizar que no se produzcan nuevos errores. Además, CI permite la implementación automática de la corrección en entornos de producción, acelerando el tiempo de entrega de la solución sin interrupciones innecesarias. En caso de errores críticos, el sistema puede volver automáticamente a la última versión estable, minimizando el impacto en los usuarios. De esta forma, la CI no sólo facilita la corrección rápida de errores, sino que también evita la acumulación de errores, dando como resultado un mantenimiento ágil, seguro y sostenible en el tiempo.

* 1. Facilidad de uso

Describe qué tan fácil resultará el uso del software a los usuarios finales.

El software se plantea de manera intuitiva para el usuario, ya sea a nivel de menús, gráficos, imagenes y tablas para tener una experiencia fluida y agradable.

* 1. Otro

Enumera cualquier otro requisito no funcional.

1. Definiciones y acrónimos



5

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |