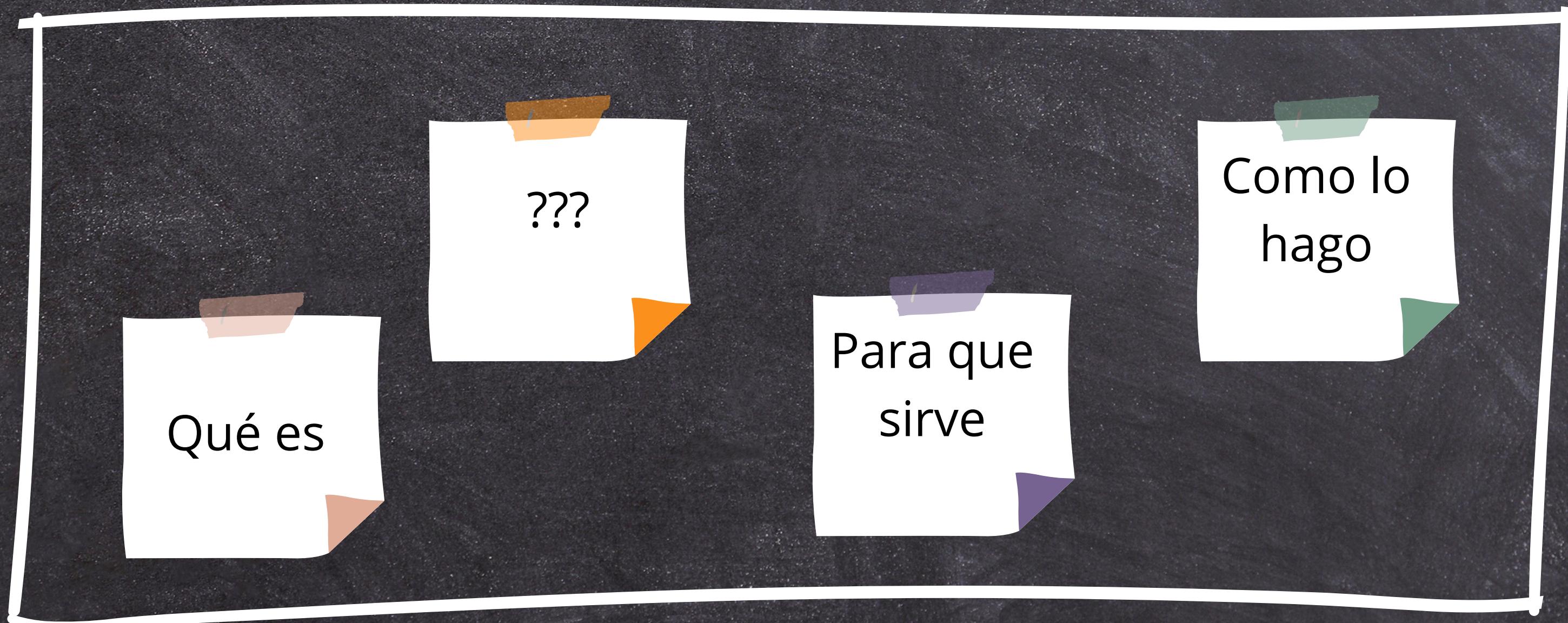


Documentación de proyecto.

MODULO IV. SESIÓN 1



CONCEPTO COLABORATIVO



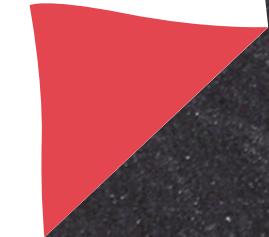
Documentación

La documentación de proyectos es una parte esencial en el desarrollo de software, ya que facilita la comprensión, mantenimiento y colaboración en los proyectos.

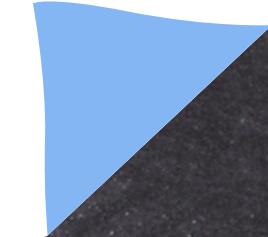
PARTES DE UNA DOCUMENTACION



RESUMEN DEL
PROYECTO



REQUISITOS DEL
PROYECTO



ARQUITECTURA
DEL PROYECTO



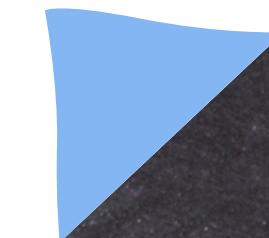
PARTES DE UNA DOCUMENTACION



INSTALACION Y
CONFIGURACION



USO DEL
PROYECTO



DESARROLLO Y
CONTRIBUCION



PARTES DE UNA DOCUMENTACION



MANTENIMIENTO



HISTORIAL DE
CAMBIOS

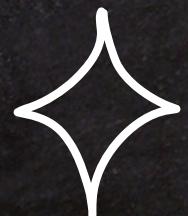


LICENCIA.



Resumen del proyecto

Es una sección clave de la documentación que proporciona una visión general concisa y clara del proyecto. Su propósito principal es ofrecer una descripción rápida y comprensible del proyecto para cualquier persona que lo lea, ya sean desarrolladores, partes interesadas, clientes o cualquier otra audiencia.



Elementos del resumen del proyecto

NOMBRE DEL PROYECTO:

EL TÍTULO OFICIAL DEL PROYECTO.

EJEMPLO: "GESTOR DE TAREAS".

DESCRIPCIÓN:

UN BREVE RESUMEN DEL PROPÓSITO, OBJETIVOS Y ALCANCE DEL PROYECTO. DEBE SER CLARO Y DIRECTO.

EJEMPLO:

"UNA APLICACIÓN WEB PARA GESTIONAR TAREAS DIARIAS, PERMITIENDO A LOS USUARIOS CREAR, EDITAR Y ELIMINAR TAREAS, ASÍ COMO MARCARLAS COMO COMPLETADAS."

ESTADO DEL PROYECTO:

EL ESTADO ACTUAL DEL PROYECTO. ESTO PUEDE INCLUIR SI EL PROYECTO ESTÁ EN DESARROLLO, COMPLETADO, EN MANTENIMIENTO, ETC.

EJEMPLO: "EN DESARROLLO"

Equipo de Desarrollo:

Una lista de los miembros del equipo y sus roles dentro del proyecto.

Ejemplo:

Juan Pérez - Desarrollador Backend

María López - Desarrolladora Frontend

Fecha de Inicio y Fin (Opcional):

Las fechas en las que el proyecto comenzó y se espera que termine.

Ejemplo:

Fecha de Inicio: 1 de enero de 2024

Fecha de Fin: 30 de junio de 2024

Objetivos del Proyecto:

Los objetivos clave que el proyecto pretende alcanzar.

Ejemplo: "Mejorar la productividad de los usuarios mediante una gestión eficiente de sus tareas."



Requisitos del proyecto

Son las especificaciones y condiciones que debe cumplir el proyecto para alcanzar sus objetivos de manera efectiva. Estos requisitos se dividen en dos categorías principales: requisitos funcionales y requisitos no funcionales. Cada categoría aborda diferentes aspectos del proyecto para garantizar que el producto final sea completo, usable y satisfaga las necesidades del usuario.





REQUISITOS FUNCIONALES:

ESPECIFICAN LO QUE EL SISTEMA DEBE HACER. INCLUYEN LAS CARACTERÍSTICAS Y FUNCIONALIDADES QUE EL SISTEMA DEBE TENER PARA CUMPLIR CON LOS OBJETIVOS DEL PROYECTO.

EJEMPLOS:

EL SISTEMA DEBE PERMITIR A LOS USUARIOS REGISTRARSE E INICIAR SESIÓN.

LOS USUARIOS DEBEN PODER CREAR, EDITAR Y ELIMINAR TAREAS.

EL SISTEMA DEBE ENVIAR NOTIFICACIONES POR CORREO ELECTRÓNICO A LOS USUARIOS.



Requisitos No Funcionales:

Especifican cómo debe comportarse el sistema. Incluyen restricciones y características que definen la calidad del sistema, como el rendimiento, la seguridad, la usabilidad, etc.

Ejemplos:

- El sistema debe ser capaz de manejar 1000 usuarios concurrentes.
- El tiempo de respuesta del sistema no debe exceder los 2 segundos bajo carga normal.
- Los datos de los usuarios deben estar encriptados en la base de datos.





Arquitectura del proyecto.

Es una descripción detallada de la estructura técnica del sistema y de cómo sus componentes interactúan entre sí. Proporciona una visión general de la organización del sistema, destacando los principales componentes, sus responsabilidades y las tecnologías utilizadas. Esta sección es crucial para entender cómo está construido el sistema y cómo debe ser mantenido y escalado.



Componentes de la arquitectura.

Diagrama de Arquitectura:

Un **diagrama visual** que muestra los principales **componentes** del sistema y sus **interacciones**. Puede incluir **diagramas de componentes**, **diagramas de despliegue**, y **diagramas de flujo de datos**.

Descripción de Componentes:

Una **explicación detallada** de cada **componente** del sistema, sus **responsabilidades** y cómo interactúan con otros componentes.

Tecnologías Utilizadas:

Una lista de las **tecnologías**, **lenguajes de programación**, **frameworks**, y **herramientas** que se utilizan en el proyecto.



Instalación y configuración.

La sección de instalación y configuración proporciona instrucciones detalladas sobre cómo instalar y configurar el proyecto para que funcione correctamente en diferentes entornos. Esta parte es esencial para garantizar que cualquier desarrollador o usuario pueda poner en marcha el proyecto de manera eficiente y sin problemas.



Uso del proyecto

La sección de "Uso del Proyecto" proporciona instrucciones detalladas sobre cómo utilizar el sistema una vez que está instalado y configurado. Esto incluye cómo acceder a las funcionalidades principales, ejemplos de uso, y cualquier otra información relevante para que los usuarios puedan interactuar con el sistema de manera efectiva.

Desarrollo y contribuciones.

"Desarrollo y Contribuciones" en la documentación de un proyecto proporciona pautas y procedimientos para que otros desarrolladores puedan contribuir al proyecto de manera efectiva. Esto incluye cómo configurar el entorno de desarrollo, las normas de codificación, el flujo de trabajo de desarrollo, y cómo enviar contribuciones.

Mantenimiento.

La sección de "Mantenimiento" en la documentación de un proyecto aborda cómo mantener el sistema funcionando correctamente a lo largo del tiempo. Esto incluye procedimientos para actualizar el sistema, gestionar dependencias, realizar copias de seguridad, monitorear el rendimiento y manejar incidentes.

Historial de cambios

La sección de "Historial de Cambios" en la documentación de un proyecto, también conocida como "Changelog," proporciona un registro cronológico de todas las modificaciones significativas realizadas en el proyecto. Esto incluye nuevas funcionalidades, mejoras, correcciones de errores y otros cambios relevantes. Un buen changelog ayuda a los usuarios y desarrolladores a entender la evolución del proyecto y facilita la gestión de versiones.



Licencias.

La sección de "Licencia" en la documentación de un proyecto especifica los términos bajo los cuales el software puede ser utilizado, modificado y distribuido. Elegir y declarar claramente una licencia es crucial para proteger los derechos del autor y establecer las condiciones para los usuarios y contribuidores del proyecto.



Ejemplo de licencias.

Ejemplos de Licencias Comunes

1. MIT License:

- Permisiva, permite prácticamente cualquier uso con mínima restricción.
- Requiere incluir el aviso de derechos de autor y la renuncia de responsabilidad.

2. Apache License 2.0:

- Similar a la MIT, pero incluye una cláusula de concesión de patente.
- Requiere que las modificaciones sean claramente documentadas.

3. GNU General Public License (GPL):

- Copyleft, requiere que las obras derivadas también se distribuyan bajo la misma licencia.
- Asegura que el software permanezca libre y abierto.



Licencias privativas



1. **Restricciones de Uso:** Establecen condiciones específicas para el uso del software, como el número de usuarios, el ámbito de uso (personal o comercial), o el período de validez de la licencia.
 2. **Limitaciones de Modificación:** Algunas licencias privativas prohíben la modificación o adaptación del software sin autorización explícita del propietario.
 3. **No Divulgación del Código Fuente:** En general, las licencias privativas no requieren que el código fuente del software esté disponible públicamente, lo que significa que los usuarios no tienen acceso al código interno del programa.
 4. **Costo:** En muchos casos, el uso del software bajo una licencia privativa implica el pago de una tarifa o suscripción, ya sea por el uso del software en sí o por el acceso a funciones adicionales y soporte técnico.
- 

Preguntas.

