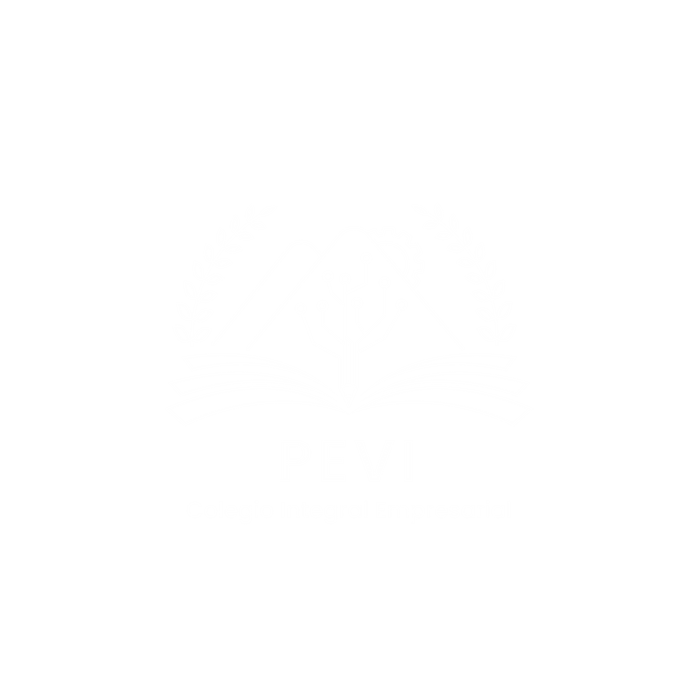
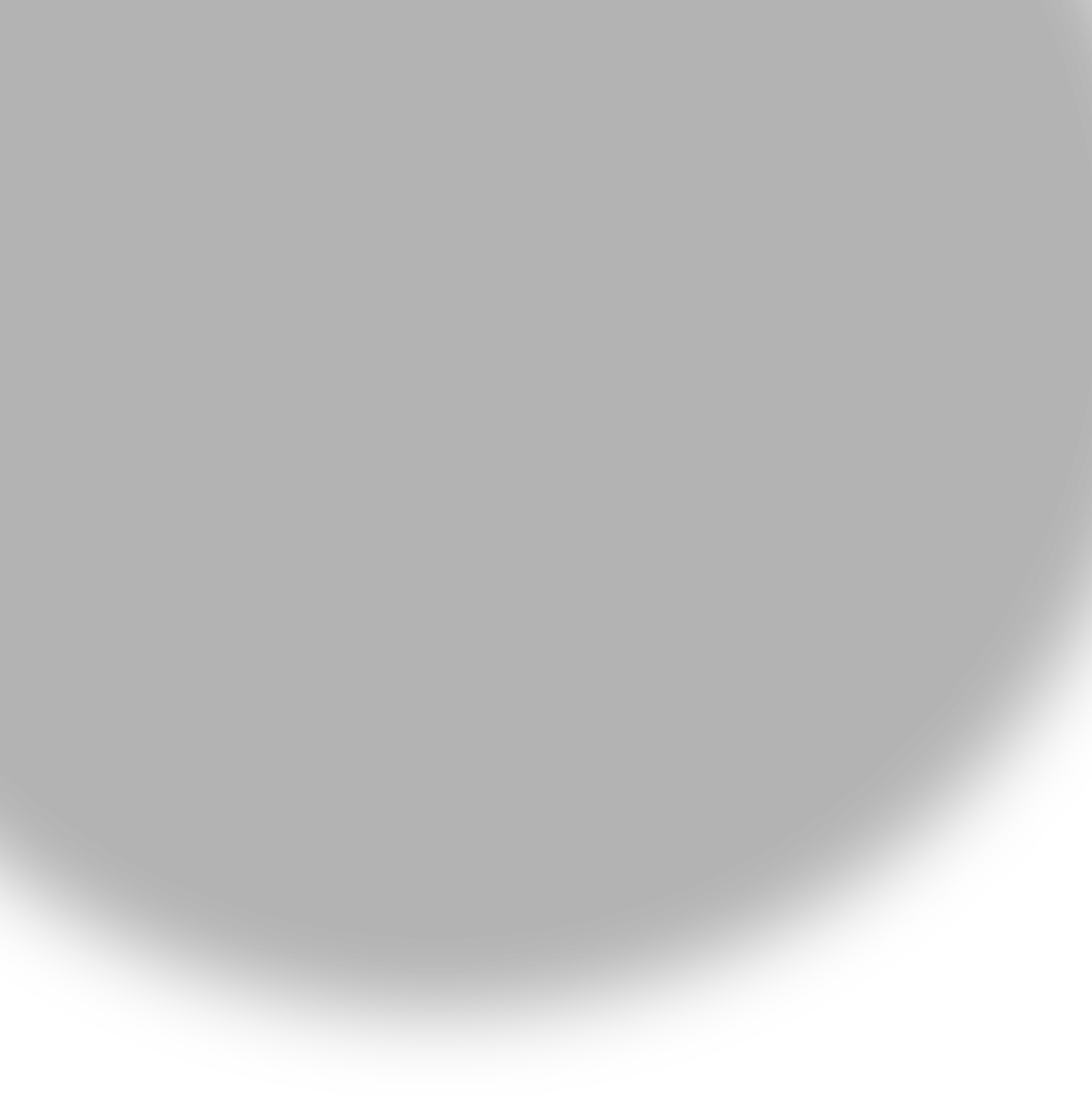
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente}

PARTICIPANTE:   
ANGEL DAVID COYOY RECANCOJ

**MANUAL DE**

**PROGRAMADOR**



NOMBRE DEL PROYECTO

Gestor de tareas avanzado

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteLOGOTIPO

Código QR

Descripción generada automáticamenteCÓDIGO QR

DESCRIPCIÓN:

El **Gestor de tareas avanzados** es una aplicación diseñada para permitir a los usuarios gestionar de manera eficiente sus tareas diarias, mejorando su organización personal. Desarrollado con el lenguaje de programación Python, este proyecto brinda una solución simple y eficaz para la planificación de actividades mediante una interfaz gráfica amigable y fácil de usar.

La aplicación se centra en ayudar a los usuarios a:

* **Agregar** nuevas tareas con detalles como el nombre, la prioridad y la fecha límite.
* **Modificar** o actualizar tareas existentes.
* **Eliminar** tareas que ya no son necesarias o que han sido completadas (no podrá eliminar tareas vencidas).
* **Visualizar** y filtrar tareas de acuerdo con su estado, prioridad, permitiendo a los usuarios enfocarse en lo más urgente o importante.
* **Organizar tareas** dependiendo su estado (pendiente, fuera de tiempo y tareas completas)

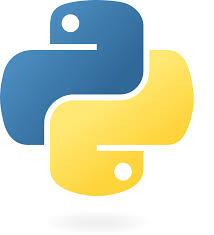
OBJETIVO PRINCIPAL:

Desarrollar una aplicación de escritorio en Python que permita a los usuarios gestionar y organizar sus tareas diarias de manera eficiente, facilitando la planificación y priorización de actividades a través de una interfaz simple, y fácil de usar, para mejorar su productividad y la organización de su tiempo.

MATERIALES Y APLICACIONES UTILIZADAS:

(CON IMAGEN Y PRECIO)

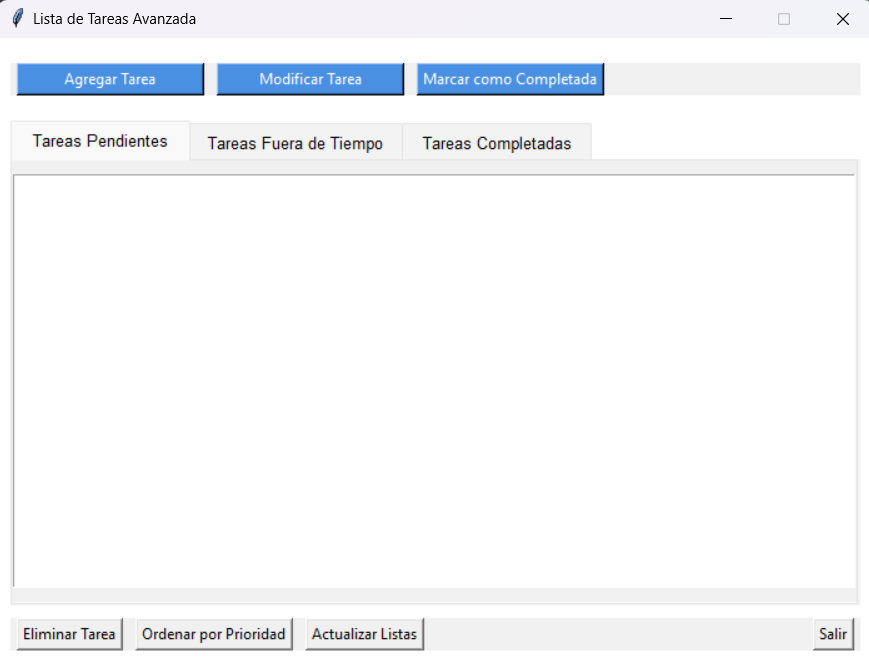
Python (sin costo) Editor de texto: Visual code (sin costo)



Computador (Ya adquirido/ sin costo)

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteCODIGO DEL PROGRAMA  
Para este programa se necesitaron 3 archivos:  
**main.py**: Archivo principal que inicia la aplicación y configura la ventana principal de la interfaz gráfica.  
**tarea\_logica.py**: Archivo que contiene la lógica de administración de tareas, como el almacenamiento, verificación de fechas límite y envío de notificaciones.  
**tarea\_ui.py**: Archivo encargado de la interfaz gráfica de usuario, permitiendo la creación, modificación y eliminación de tareas mediante componentes visuales.  
**tareas.json**: Almacena las tareas ingresadas (tareas pendientes, tarea s fuera de tiempo y tarea completadas) para que los datos se sigan mostrando cada vez que se ejecute el programa  
  
  
  
  
  
  
  
  
  
**main.py:** Código del archivo **main.py** es el principal para ejecutar y configurar la ventana principal

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media**Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTarea\_logica.py:** Maneja la lógica del código, se utilizaron las siguientes bibliotecas   
  
  
  
  
  
  
Se definen las listas para almacenar los datos según la función de cada una  
  
  
  
  
  
  
  
Crea las notificaciones del programa  
  
  
  
  
  
  
  
  
  
Código para agregar una nueva tarea

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteCódigo lógica para eliminar tareas

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente  
  
  
  
Lógica código para modificar una tarea  
  
  
  
Código para filtrar las tareas por prioridad  
  
  
  
  
  
  
Código para almacenar datos en el archivo JSON

Tarea\_ui.py: Archivo encargado de la interfaz gráfica de usuario, se usaron las siguientes Texto

Descripción generada automáticamente bibliotecas

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente  
  
  
Código para crear pestañas  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Código para manejar las funciones de agregar y eliminar tareas

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteCodigo para modificar tareas y marcar tareas como completadas  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Crear ventana principal

PROCEDIMIENTOS DE INSTALACIÓN Y CONFIGURACIÓN:

**Instalar Python:**

Descarga Python desde https://www.python.org/downloads/.

**Instalar Git:**

Descargar Git desde <https://git-scm.com/>.

**Clonar el repositorio**:  
 Abre una terminal, navega a la carpeta donde deseas clonar el proyecto y ejecuta:

“git clone <https://github.com/AngelCoyoy12/Gestor_Tareas_Avanzado>”  
  
 **Ejecuta el código en tu editor de texto:**

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteAbre el código clonado en tu editor de texto agregar las bibliotecas necesarias que requiera el código (el editor los ira pidiendo) Y ejecuta el archivo main.py y listo ya tiene le gestor de tareas avanzado.

RECOMENDACIONES:

**Registrar todas tus tareas**: Ingresa todas las actividades pendientes. Asegúrate de incluir detalles importantes como la fecha de vencimiento y la prioridad para un mejor control.

**Actualizar tus tareas regularmente**: Modifica o elimina las tareas según vayas avanzando, esto te permitirá mantener la lista siempre organizada.

**Planificar tus días**: Cada día, revisa las tareas pendientes y ajusta tus prioridades. Esto mejorará tu productividad

**Utilizar categorías y estados**: Clasifica las tareas por estado (pendiente, en progreso, completada) para tener un mejor control visual de tu avance.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente INFORMACIÓN DE CONTACTO: