

MANUAL TÉCNICO DEL SISTEMA - FecGraph



**PYTHON + TKINTER
+ MATPLOTLIB**

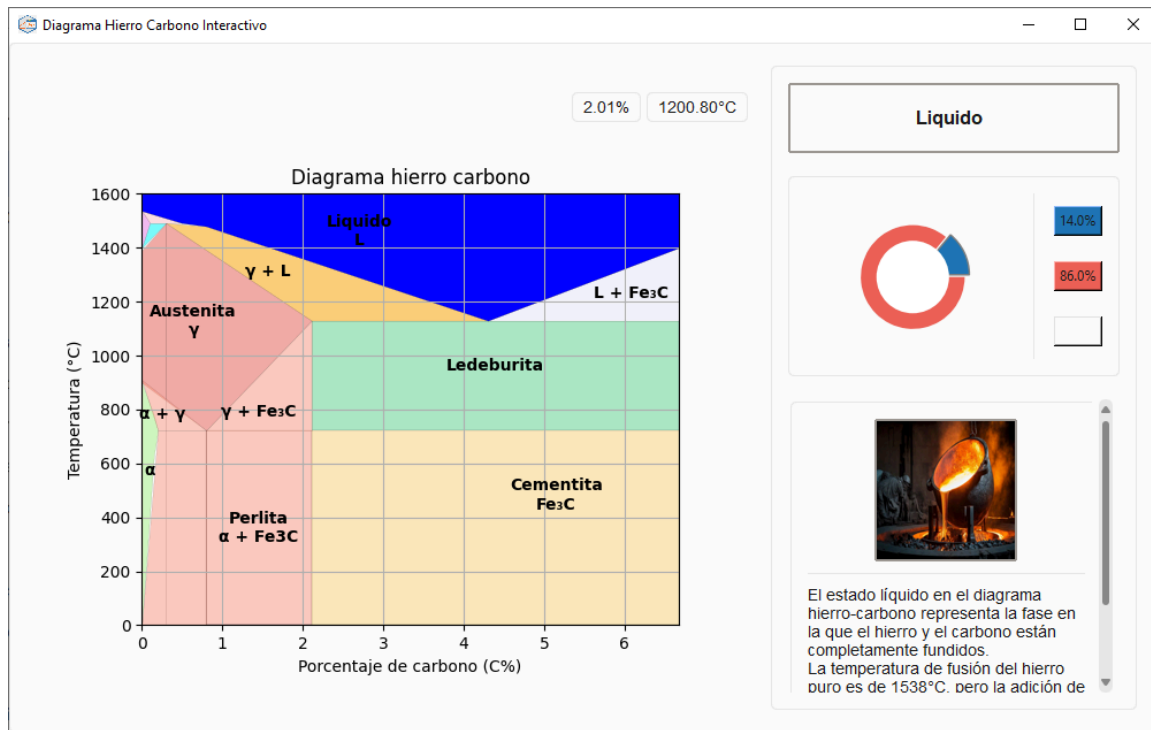
Universidad de La Guajira, Facultad de ingeniería

2025

1. Índice

1. Índice	2
2. Introducción	3
3. Objetivos del sistema	4
4. Requisitos del sistema	5
Figura 2. Sistemas operativos compatibles	5
5. Instalación	6
Figura 3. Pagina de github	6
Figura 4. Vista de Release en github	6
6. Diseño técnico del sistema	9
6.1 Requerimientos del sistema	9
6.2 Frameworks y librerías utilizadas	9
6.3 Modelo de datos	9
6.4 Funcionalidad y servicios	9
7. Despliegue y configuración	10
7.1 Organización de componentes	10
7.2 Instalación para desarrollo	11
7.3 Configuración	13
7.4 Despliegue	14
8. Resolución de problemas	14

2. Introducción



El presente manual técnico documenta los aspectos fundamentales del desarrollo, funcionamiento y despliegue de *GraphFec*, una aplicación interactiva para la visualización del diagrama hierro-carbono, ampliamente utilizado en el estudio del comportamiento térmico y estructural de aleaciones metálicas como el acero y el hierro fundido.

Esta herramienta ha sido concebida con el propósito de servir como recurso didáctico y técnico en contextos académicos, formativos y profesionales, permitiendo al usuario explorar visualmente los límites de las fases metalográficas en función del porcentaje de carbono y la temperatura.

GraphFec está desarrollado en el lenguaje de programación Python, haciendo uso de bibliotecas como Tkinter para la interfaz gráfica, Matplotlib para el renderizado del diagrama y Pillow para el manejo de imágenes. Gracias a estas tecnologías, la aplicación logra un equilibrio entre funcionalidad, ligereza y compatibilidad, presentando al usuario una interfaz intuitiva en la que puede interactuar directamente con las diferentes regiones del diagrama. Al posicionar el cursor sobre una fase específica, esta se resalta visualmente y despliega un cuadro informativo (tooltip) que incluye su nombre y, al hacer clic, una imagen representativa junto con información detallada.

Este manual se dirige principalmente a desarrolladores, técnicos de sistemas o personal encargado del mantenimiento y despliegue de la aplicación, proporcionando una guía

estructurada sobre su arquitectura interna, requerimientos, instalación, configuración, solución de problemas y posibilidades de expansión. Asimismo, busca facilitar futuras mejoras o migraciones del sistema, fomentando su reutilización y sostenibilidad en el tiempo como herramienta educativa.

3. Objetivos del sistema



El principal objetivo del sistema FecGraph es proporcionar una herramienta interactiva, visual y educativa que facilite la comprensión del diagrama hierro-carbono, elemento fundamental en los estudios de metalurgia, ciencia de materiales e ingeniería mecánica. A través de una interfaz gráfica accesible y amigable, el sistema permite explorar de manera intuitiva las distintas fases del acero y del hierro fundido en función de su composición de carbono y temperatura, lo que tradicionalmente requiere una interpretación abstracta de gráficos complejos.

Mediante la interacción directa con las regiones del diagrama —resaltado visual, tooltips y paneles explicativos con imágenes— el usuario puede identificar fácilmente las características y comportamientos de fases como la austenita, la ferrita, la cementita, entre otras. Esto convierte al sistema no solo en un recurso visual, sino en una plataforma de aprendizaje activo que promueve la apropiación de conceptos metalográficos clave. Además, al estar desarrollado como una aplicación de escritorio ligera y portátil, FecGraph busca democratizar el acceso al conocimiento técnico, brindando a estudiantes, docentes y profesionales una alternativa digital moderna para el análisis de transformaciones

estructurales en los aceros, sin necesidad de conexión a internet ni software costoso o especializado. Entre sus objetos principales encontramos:

- Facilitar el análisis del diagrama Fe-C mediante una interfaz visual y accesible.
- Permitir interacción con las distintas fases mediante hover y clic.
- Mostrar información e imágenes asociadas a cada fase para apoyar el estudio metalográfico.

4. Requisitos del sistema



Figura 2. Sistemas operativos compatibles

- Sistema operativo: Windows 10 o superior
- Recomendado: resolución de pantalla 1366x768 o mayor
- Memoria RAM: 4 GB mínimo
- Espacio en disco: ~50 MB

Actualmente, GraphFec está disponible únicamente para sistemas operativos Windows, dado que el ejecutable ha sido empaquetado específicamente para esta plataforma. Sin embargo, se tiene previsto en futuras versiones lanzar versiones compatibles con otros sistemas operativos como Linux y macOS, ya sea mediante ejecución directa del código fuente o empaquetado multiplataforma.

Se le recuerda que a después del 14 de octubre de 2025, Microsoft dejará de ofrecer actualizaciones de seguridad gratuitas desde Windows Update, asistencia técnica y correcciones de seguridad de forma gratuita para Windows 10, así como para versiones anteriores del sistema operativo.

5. Instalación

1. Descargar el archivo ejecutable ``FecGraph-vX.Y.Z-(OS).exe`` desde la sección de Releases del repositorio oficial en GitHub según su dispositivo.

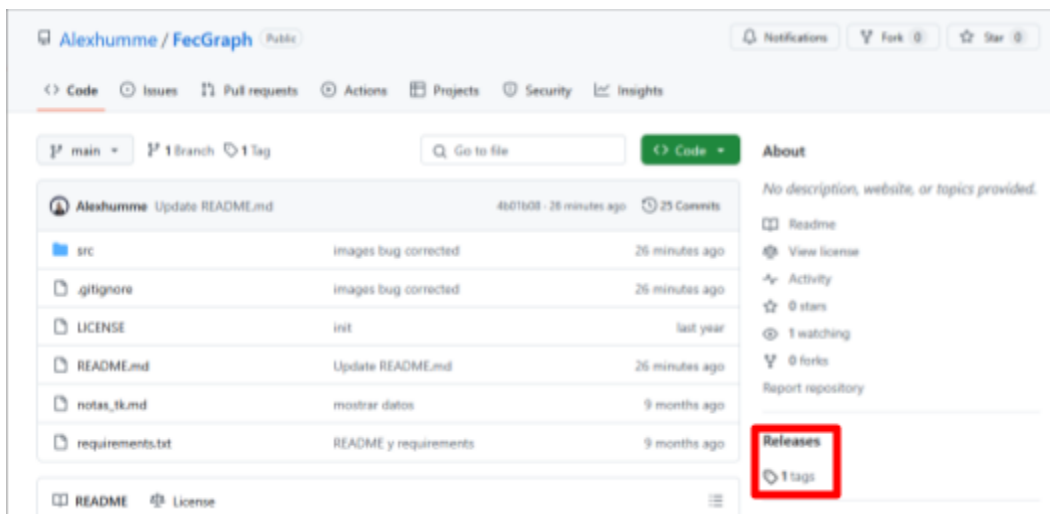


Figura 3. Pagina de github

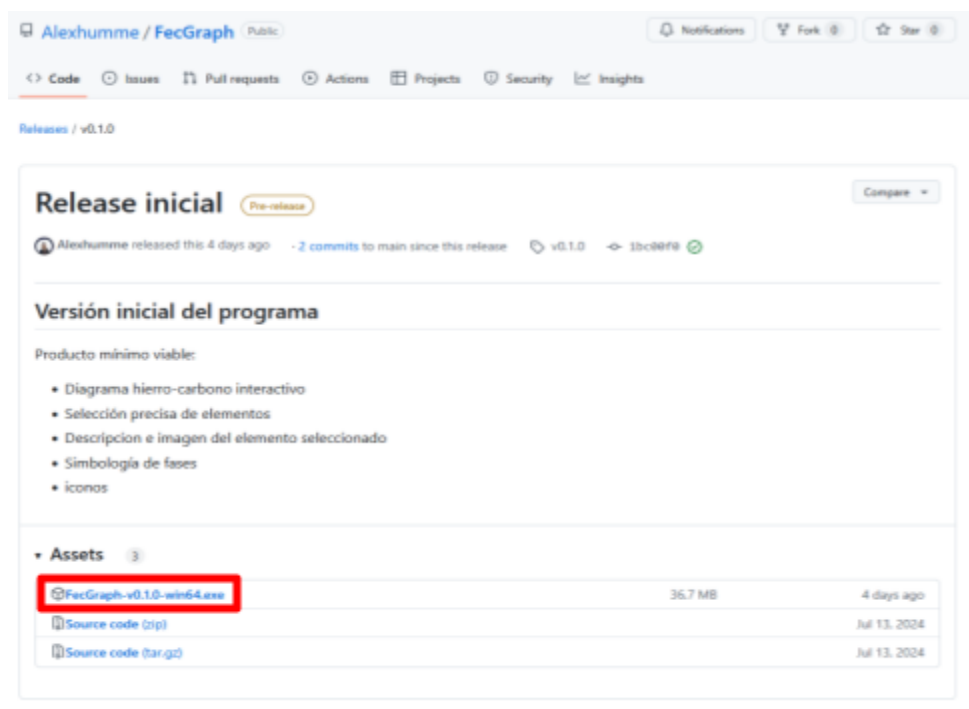


Figura 4. Vista de Release en github

2. Guardar el archivo en una carpeta segura.

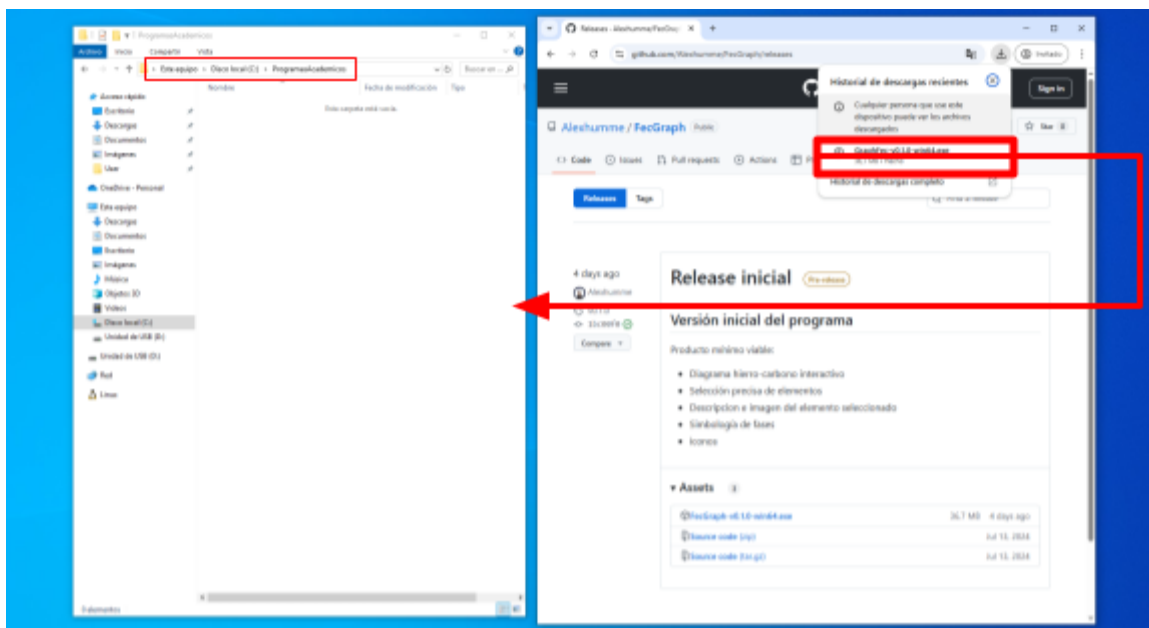


Figura 5. Guardar archivo descargado

3. Hacer doble clic sobre el archivo para ejecutar el programa.

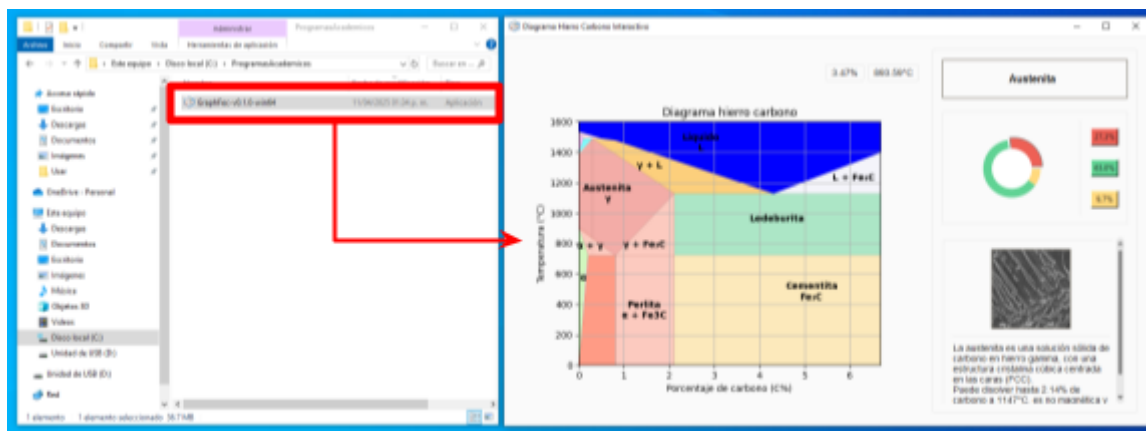


Figura 6. Ejecutar programa

En algunos casos, al ejecutar GraphFec por primera vez, Windows puede **bloquear su ejecución** con un mensaje de advertencia indicando que el archivo podría ser potencialmente peligroso. Esto es una medida de seguridad estándar del sistema operativo ante aplicaciones no firmadas digitalmente. Para continuar, haz clic en **“Más información”** y luego en **“Ejecutar de todas formas”**. Asegúrate de haber descargado el ejecutable desde la fuente oficial (como la sección de Releases del repositorio en GitHub) para garantizar su seguridad.

4. (Opcional) Crear un acceso directo en el escritorio.

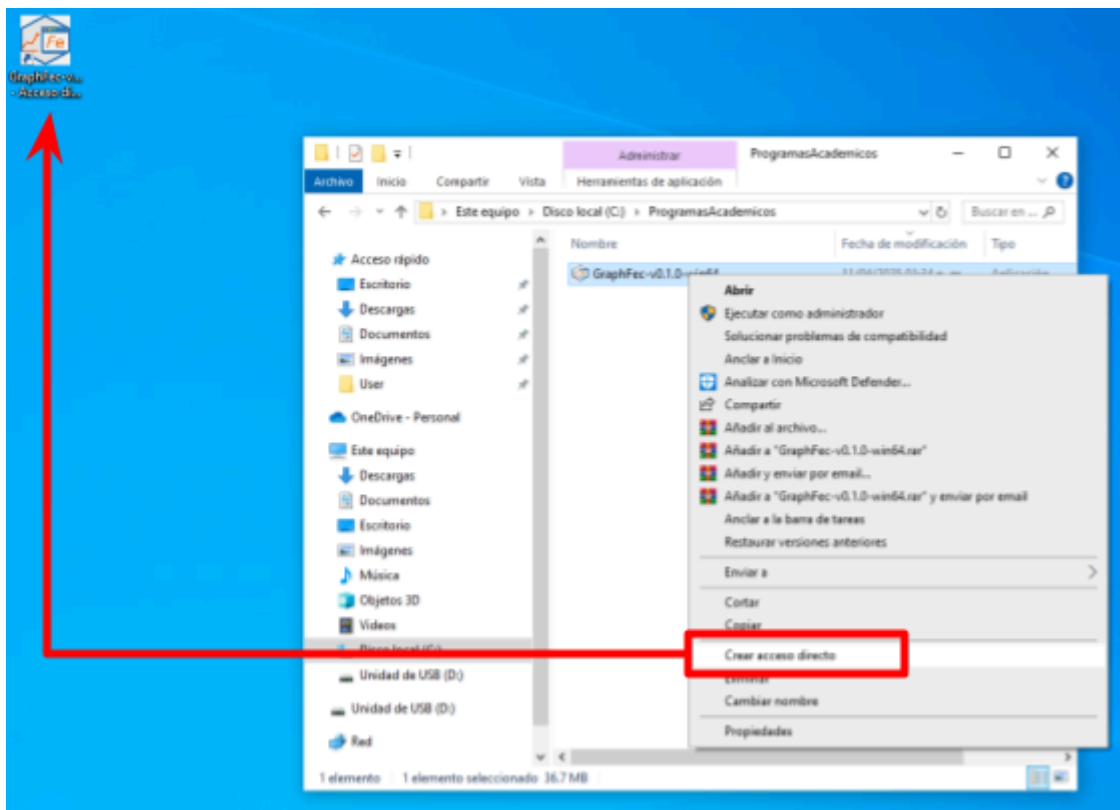


Figura 7. Crear acceso directo

6. Diseño técnico del sistema

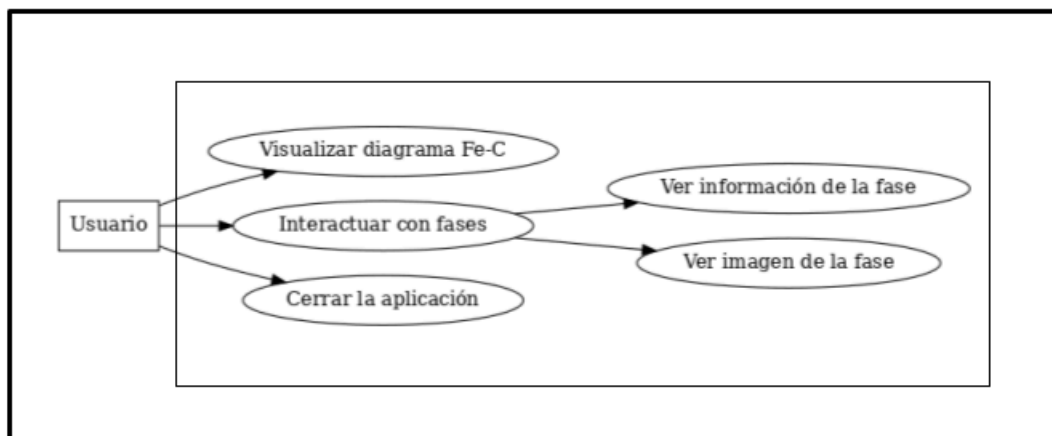
6.1 Requerimientos del sistema

- Sistema operativo: Windows 10 o superior
- Python versión: 3.10+ (solo durante desarrollo)
- Ejecutable final: .exe creado con PyInstaller
- Memoria RAM recomendada: 4 GB
- Almacenamiento requerido: ~50 MB

6.2 Frameworks y librerías utilizadas

- Tkinter: interfaz gráfica (GUI)
- Matplotlib: renderizado del gráfico
- Pillow: manejo de imágenes PNG
- PyInstaller: empaquetado a .exe

6.3 Casos de uso



El sistema GraphFec está diseñado en torno a una serie de casos de uso centrados en la interacción del usuario con el diagrama hierro-carbono. El caso de uso principal es la visualización del diagrama, que permite al usuario explorar de forma gráfica la relación entre la temperatura y el porcentaje de carbono en las distintas fases del acero. A partir de este caso central, se derivan otros casos como la interacción con las fases, donde el usuario puede posicionar el cursor sobre una región específica del gráfico para resaltar la fase correspondiente y recibir información visual inmediata mediante tooltips. Si se realiza clic en una fase, se despliega información más detallada acompañada de una imagen

representativa. Finalmente, el sistema contempla el cierre de la aplicación como un caso de uso esencial para garantizar una experiencia de usuario completa. Estos casos de uso permiten definir claramente los límites funcionales del sistema y orientar su diseño desde una perspectiva orientada al usuario.

6.3 Modelo de datos

Los datos de las fases están estructurados como un arreglo de objetos tipo diccionario, donde cada fase incluye nombre, símbolo, descripción, propiedades, lista de puntos para graficar (line_x, line_y), color y ruta de imagen asociada.

```
data: list = [  
  {  
    "name": "Austenita",  
    "symbol": "γ",  
    "description": "La austenita es una solución sólida de carbono en hierro gamma, con una estructura cristalina cúbica centrada en las caras (FCC).\nPuede disolver hasta 2.14% de carbono a 1147°C, es no magnética y relativamente dúctil y tenaz.",  
    "line_x": [0, 0.3, 2.11, 0.8, 0],  
    "line_y": [1390, 1492, 1130, 723, 900],  
    "color": "#EE6055",  
    "img": "austenita.png"  
  },  
  ...  
]
```

6.4 Funcionalidad y servicios

- Renderizado del diagrama con límites ajustables.
- Detección de colisión para interacción con regiones del gráfico (fases).
- Visualización de tooltip e imagen explicativa.
- Cambios visuales dinámicos al hacer hover (alpha, borde, etc).

```
class InteractivePlot(ttk.Frame):
    def __init__(self, master): ...
    def create_controls(self): ...
    def create_data_bar(self): ...
    def create_plot(self): ...
    def on_hover(self, event): ...
    def update_cards(self, event): ...

class PercPlot(ttk.Frame):
    def __init__(self, master): ...
    def create_controls(self): ...
    def load_phases(self, phases, t): ...

class MainApp(tk.Tk):
    def __init__(self): ...
    def handle_graph_click(self, event): ...
    def on_close(self): ...

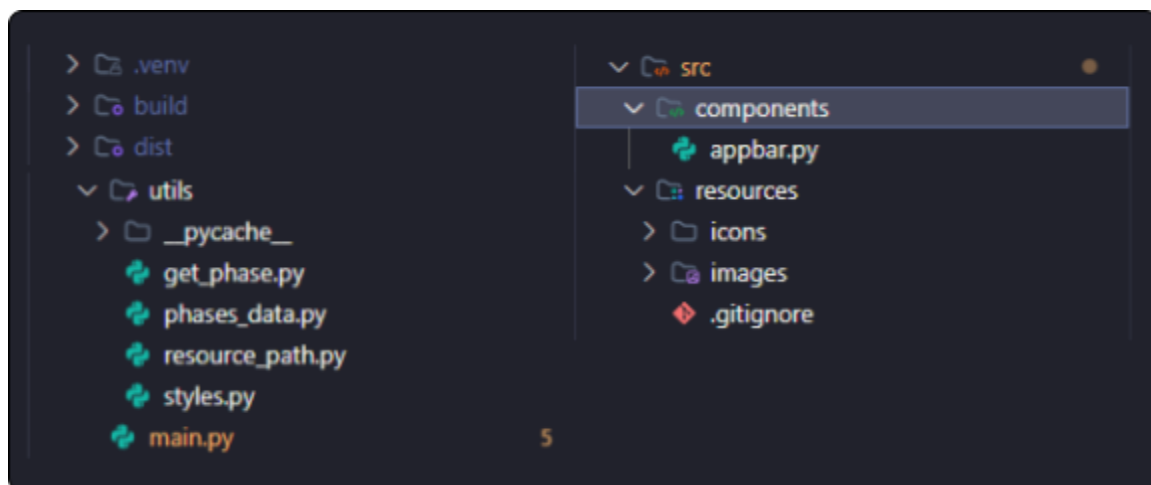
class ScrollableFrame(ttk.Frame):
    def __init__(self, container, *args, **kwargs): ...

class Sidebar(ttk.Frame):
    def __init__(self, master): ...
    def create_controls(self): ...
    def handle_graph_click(self, event): ...
    def load_data(self, index): ...
```

7. Despliegue y configuración

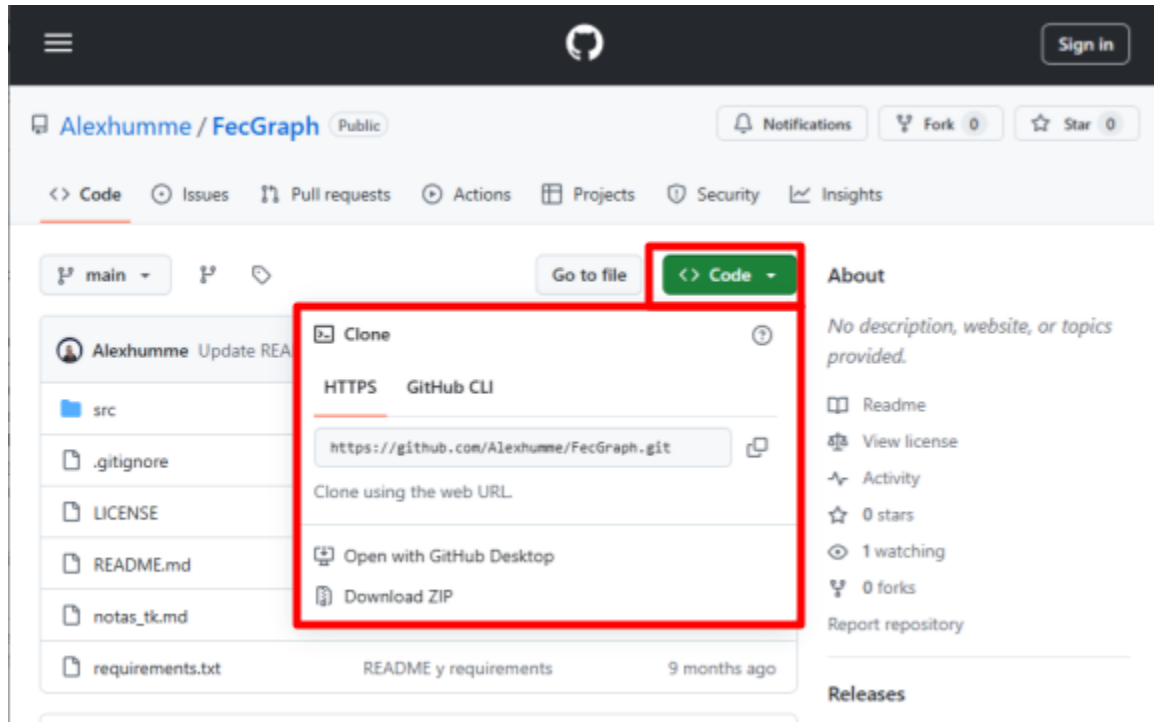
7.1 Organización de componentes

- /src – Contiene el código fuente.
- /resources/images – Imágenes de cada fase.
- /utils/resource_path.py – Función para rutas compatibles con PyInstaller.



7.2 Instalación para desarrollo

1. Clonar o descomprimir el repositorio.



2. Crear y activar entorno virtual (solo para desarrollo).

```
PS C:\Users\User\Proyectos\FecGraph> python -m venv .venv
```

```
PS C:\Users\User\Proyectos\FecGraph> .\.venv\Scripts\activate
```

3. Instalar dependencias con:

```
(.venv) PS C:\Users\User\Proyectos\FecGraph> pip install -r .\requirements.txt
```

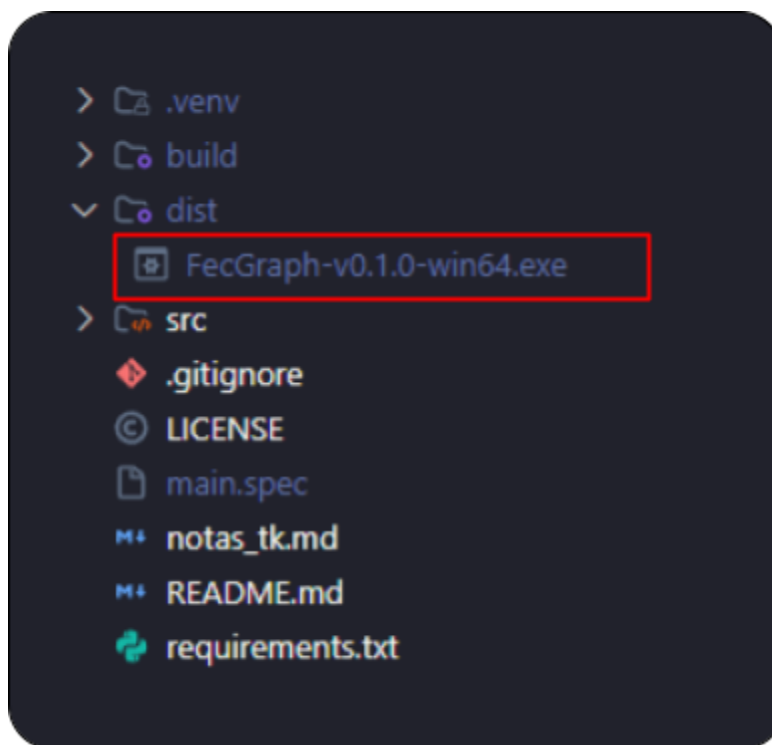
4. Ejecutar el programa (modalidad de desarrollo):

```
(.env) PS C:\Users\User\Proyectos\FecGraph> python .\src\main.py
```

5. Una vez completada una nueva versión lanzable, generar el .exe:

```
(.env) PS C:\Users\User\Proyectos\FecGraph> pyinstaller --noconsole --onefile --  
icon=./src/resources/icons/favicon.ico --add-data "src/resources/images;src/  
resources/images" src/main.py
```

6. El ejecutable estará en /dist.



7.3 Configuración

- No requiere base de datos.
- Las rutas internas usan `resource_path()` para compatibilidad con PyInstaller.

```
# src\utils\resource_path.py

import sys
import os

def resource_path(relative_path):
    """Obtiene la ruta absoluta del recurso, compatible con PyInstaller."""
    try:
        # PyInstaller crea una carpeta temporal y guarda el path en _MEIPASS
        base_path = sys._MEIPASS
    except AttributeError:
        base_path = os.path.abspath(".")

    return os.path.join(base_path, relative_path)
```

- Compatible con pantallas HD y escalables.

7.4 Despliegue

- Subir el .exe renombrado como FecGraph.exe a la sección Releases en GitHub.
- Asegurarse de incluir instrucciones de uso y permisos de ejecución si aplica.

8. Sobre licencias de uso y derechos de autor

Este programa está protegido por una licencia de libre uso y distribución para fines educativos; el código puede ser descargado libremente por los estudiantes de la universidad de la Guajira.

El software incluye figuras e imágenes recuperadas de la página web **commons.wikimedia.org** y están amparadas bajo licencias de libre uso y distribución para fines educativos. La acreditación y fuentes para este material se puede encontrar en el archivo "LICENCES.txt" en el repositorio del proyecto.

9. Resolución de problemas

Error	Causa común	Solución
FileNotFoundError	El .exe no encuentra las imágenes	Usar --add-data en PyInstaller y resource_path()
El programa no termina al cerrar	Falta sys.exit() o handler de cierre	Usar root.protocol("WM_DELETE_WINDOW", on_close)
Las imágenes no se cargan	Pérdida de referencia a PhotoImage	Guardar referencia como self.imgtk = ImageTk.PhotoImage(...)