**Documentación Completa del Módulo de Gráficas (modulos/graficas.py)**

**Propósito General del Módulo**

El **módulo de gráficos** está diseñado para generar visualizaciones de datos mediante dos enfoques principales: gráficos estáticos y gráficos interactivos. Utiliza las bibliotecas **matplotlib** para gráficos estáticos y **plotly** para gráficos interactivos. El módulo tiene como objetivo facilitar la creación de representaciones visuales de datos en forma de gráficos de líneas, barras y dispersión.

Este módulo es modular y flexible, permitiendo la personalización de los gráficos según el tipo de visualización y las preferencias del usuario (interactividad o estática).

**Funciones del Módulo de Gráficas**

**1. generar\_grafico(datos, tipo='linea', grafico\_interactivo=False)**

* **Propósito**: Esta función es el punto de entrada principal para generar gráficos. Dependiendo del tipo de gráfico (estático o interactivo) y los datos proporcionados, la función delega la tarea a las funciones correspondientes para generar gráficos estáticos con **matplotlib** o interactivos con **plotly**.
* **Parámetros**:
  + **datos** (list o dict):
    - Si es una lista, se asume que es una sola serie de datos.
    - Si es un diccionario, se asume que las claves son las leyendas o nombres de las series y los valores son las series de datos asociadas.
    - Ejemplo: {"Serie 1": [10, 20, 30], "Serie 2": [30, 40, 50]}
  + **tipo** (str, opcional, por defecto 'linea'):
    - El tipo de gráfico a generar. Puede ser:
      * 'linea': Gráfico de líneas.
      * 'barra': Gráfico de barras.
      * 'dispercion': Gráfico de dispersión (puntos).
  + **grafico\_interactivo** (bool, opcional, por defecto False):
    - Si es True, genera un gráfico interactivo usando **plotly**.
    - Si es False, genera un gráfico estático usando **matplotlib**.
* **Salida**:
  + No devuelve un valor. La salida es el gráfico generado y mostrado en la interfaz.
* **Descripción**: Esta función decide, según el valor de grafico\_interactivo, si se debe generar un gráfico estático con **matplotlib** o un gráfico interactivo con **plotly**. Luego, llama a las funciones correspondientes para generar el gráfico.

**Flujo**:

* + Recibe los datos y el tipo de gráfico deseado.
  + Si grafico\_interactivo es True, llama a generar\_grafico\_interactivo().
  + Si grafico\_interactivo es False, llama a generar\_grafico\_estatico().

**Ejemplo de uso**:

datos = [10, 20, 30, 40]

generar\_grafico(datos, tipo='linea', grafico\_interactivo=False) # Gráfico estático

**2. generar\_grafico\_estatico(datos, tipo='linea')**

* **Propósito**: Genera un gráfico estático utilizando **matplotlib**. Se encarga de crear gráficos con los datos proporcionados, ya sea de tipo línea, barra o dispersión.
* **Parámetros**:
  + **datos** (list o dict):
    - Si es una lista, se asume una única serie de datos.
    - Si es un diccionario, las claves representan las leyendas y los valores son las series de datos.
  + **tipo** (str, opcional, por defecto 'linea'):
    - El tipo de gráfico a generar. Puede ser 'linea', 'barra', o 'dispercion'.
* **Salida**:
  + No devuelve valores. Muestra el gráfico generado en una ventana emergente.
* **Descripción**: Utiliza **matplotlib** para crear un gráfico estático. Si los datos son una lista, se genera un gráfico con una sola serie. Si los datos son un diccionario, se genera un gráfico con múltiples series. La función soporta tres tipos de gráficos:
  + **Gráfico de líneas** ('linea'): Una representación de los datos en líneas.
  + **Gráfico de barras** ('barra'): Las series de datos se representan como barras.
  + **Gráfico de dispersión** ('dispercion'): Los puntos de los datos se grafican como puntos dispersos en el plano cartesiano.

**Flujo**:

* + Verifica el tipo de gráfico.
  + Dependiendo del tipo de gráfico y los datos proporcionados, genera el gráfico correspondiente usando **matplotlib**.
  + Muestra el gráfico.

**Ejemplo de uso**:

datos = [10, 20, 30, 40, 50]

generar\_grafico\_estatico(datos, tipo='barra')

**3. generar\_grafico\_interactivo(datos, tipo='linea')**

* **Propósito**: Genera un gráfico interactivo usando **plotly**. La interactividad permite al usuario acercar, mover o visualizar los datos de forma más dinámica.
* **Parámetros**:
  + **datos** (list o dict):
    - Si es una lista, se genera un solo gráfico de la serie.
    - Si es un diccionario, se generan múltiples gráficos con las leyendas especificadas.
  + **tipo** (str, opcional, por defecto 'linea'):
    - El tipo de gráfico a generar. Puede ser 'linea', 'barra', o 'dispercion'.
* **Salida**:
  + No devuelve valores. Muestra el gráfico interactivo en una ventana del navegador.
* **Descripción**: Utiliza **plotly** para crear un gráfico interactivo. El usuario podrá hacer zoom, mover el gráfico, y obtener información interactiva sobre los puntos de datos. Al igual que en la versión estática, soporta gráficos de tipo línea, barra y dispersión.

**Flujo**:

* + Verifica el tipo de gráfico.
  + Dependiendo del tipo de gráfico y los datos, genera el gráfico interactivo correspondiente.
  + Muestra el gráfico en una ventana del navegador.

**Ejemplo de uso**:

datos = [10, 20, 30, 40, 50]

generar\_grafico\_interactivo(datos, tipo='linea')

**Descripción General del Flujo de Datos y Ejecución**

1. **Entrada**: El controlador recibe la solicitud para generar un gráfico con ciertos datos y el tipo de gráfico (línea, barra, dispersión) y si debe ser interactivo o no.
2. **Selección de Función**: Dependiendo de los parámetros, se llama a la función correspondiente:
   * Si el parámetro grafico\_interactivo es True, se llama a la función generar\_grafico\_interactivo().
   * Si el parámetro grafico\_interactivo es False, se llama a la función generar\_grafico\_estatico().
3. **Generación de Gráfico**:
   * La función seleccionada genera el gráfico con **matplotlib** o **plotly**, dependiendo del tipo de gráfico solicitado.
4. **Salida**: El gráfico se muestra al usuario ya sea de forma estática en una ventana emergente o de forma interactiva en una página web.

**Consideraciones Importantes**

* **Interactividad**: El uso de **plotly** proporciona una visualización dinámica, permitiendo al usuario explorar los datos de forma más detallada, mientras que **matplotlib** ofrece gráficos más simples y adecuados para visualizaciones estáticas.
* **Flexibilidad**: El módulo soporta gráficos de líneas, barras y dispersión, lo que cubre la mayoría de las necesidades de visualización de datos.
* **Integración**: Este módulo está diseñado para ser fácilmente invocado desde el controlador central, lo que permite que el sistema funcione de manera coherente con las demás funcionalidades del programa.

**Conclusión**

El **módulo de gráficos** ofrece una solución robusta y flexible para la visualización de datos. Con soporte para gráficos estáticos e interactivos, cubre una variedad de casos de uso comunes en la representación gráfica de datos. Su integración con el controlador y otros módulos asegura una experiencia de usuario fluida y eficiente.