

Contar milisegundos desde medianoche



Completa el siguiente programa para contar milisegundos: `def run(hours: int, minutes: int, seconds: int) -> float:`
`# TU CÓDIGO AQUÍ`
`time_since_midnight = 'output'`

`return time_since_midnight`



Para completar el programa que cuenta milisegundos desde la medianoche, podemos convertir las horas, minutos y segundos a milisegundos y luego sumarlos. Aquí tienes el código completo:

python

```
def run(hours: int, minutes: int, seconds: int) -> float:
    # Convertir horas, minutos y segundos a milisegundos
    milliseconds = (hours * 3600 + minutes * 60 + seconds) * 1000
    time_since_midnight = milliseconds

    return time_since_midnight
```

Este código toma las horas, minutos y segundos, los convierte a milisegundos y devuelve el total de milisegundos desde la medianoche.



realiza un programa en python que calcule el área de un anillo, deberá preguntar el radio mayor y el radio menor



Claro, a continuación te proporciono un programa en Python que calcula el área de un anillo. Este programa pedirá al usuario el radio mayor y el radio menor, y luego calculará el área del anillo utilizando la fórmula $\pi(R^2 - r^2)$, donde R es el radio mayor y r es el radio menor.

python

```
import math

def calcular_area_anillo(radio_mayor: float, radio_menor: float) -> float:
    if radio_mayor <= radio_menor:
        raise ValueError("El radio mayor debe ser mayor que el radio menor.")

    area_anillo = math.pi * (radio_mayor**2 - radio_menor**2)
    return area_anillo

def main():
```

```
try:
    radio_mayor = float(input("Introduce el radio mayor: "))
    radio_menor = float(input("Introduce el radio menor: "))

    area = calcular_area_anillo(radio_mayor, radio_menor)
    print(f"El área del anillo es: {area:.2f} unidades cuadradas.")
except ValueError as e:
    print(f"Error: {e}")

if __name__ == "__main__":
    main()
```

Este programa funciona de la siguiente manera:

1. Define la función `calcular_area_anillo` que toma el radio mayor y el radio menor como argumentos y calcula el área del anillo.
2. En la función `main`, se solicita al usuario que introduzca los valores del radio mayor y el radio menor.
3. La función `calcular_area_anillo` es llamada con estos valores y el resultado se imprime en pantalla.
4. Si el usuario introduce valores inválidos (por ejemplo, si el radio mayor no es realmente mayor que el radio menor), se lanza una excepción y se imprime un mensaje de error.

Puedes ejecutar este programa en cualquier entorno de Python para calcular el área de un anillo.



Tengo un archivo llamado "calificaciones_alumnos.xlsx", el cual tiene un campo llamado "Nombre" y varias columnas de calificaciones llamadas Mat_CalculoIntegral, Mat_ProgramacionOOP y Mat_EstructuraDatos. Ayúdame a crear un programa en Python para graficar las calificaciones para cada alumno pero no quiero que se encimen sus etiquetas en el eje X, al inicio del código agrega un comentario explicando lo que hace el programa y documenta las líneas importantes del código



Aquí tienes un programa en Python que utiliza `pandas` para leer un archivo Excel y `matplotlib` para graficar las calificaciones de los alumnos. Se asegura de que las etiquetas en el eje X no se encimen mediante el uso de la rotación de etiquetas y ajuste de espacio.

python

```
"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Luego, grafica
estas calificaciones en un gráfico de barras para cada alumno, asegurándose de que
las etiquetas en el eje X no se encimen.
"""

import pandas as pd
import matplotlib.pyplot as plt

# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)

# Seleccionar columnas de interés
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_ProgramacionOOP',
```

```
'Mat_EstructuraDatos']

# Crear el gráfico
fig, ax = plt.subplots(figsize=(10, 6))

# Iterar sobre cada columna de calificación para crear una serie de barras
for idx, columna in enumerate(columnas_calificaciones):
    # Ajustar las posiciones de las barras para que no se encimen
    posiciones = [i + idx*0.2 for i in range(len(df))]
    ax.bar(posiciones, df[columna], width=0.2, label=columna)

# Configurar las etiquetas del eje X
ax.set_xticks([i + 0.2 for i in range(len(df))])
ax.set_xticklabels(df['Nombre'], rotation=45, ha='right')

# Agregar leyenda
ax.legend()

# Agregar títulos y etiquetas
ax.set_title('Calificaciones de los Alumnos')
ax.set_xlabel('Alumnos')
ax.set_ylabel('Calificaciones')

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```

Explicación del código:

1. Importaciones necesarias:

- `pandas` para manipular datos.
- `matplotlib.pyplot` para crear gráficos.

2. Leer el archivo Excel:

- `df = pd.read_excel(archivo_excel)` carga el archivo Excel en un DataFrame de pandas.

3. Seleccionar columnas de interés:

- `columnas_calificaciones` contiene los nombres de las columnas que corresponden a las calificaciones.

4. Crear el gráfico:

- `fig, ax = plt.subplots(figsize=(10, 6))` crea una figura y un eje para el gráfico con un tamaño de 10x6 pulgadas.

5. Iterar sobre cada columna de calificación:

- Para cada materia, se calcula una posición ajustada para las barras usando `posiciones = [i + idx*0.2 for i in range(len(df))]` para evitar que se encimen.
- `ax.bar(posiciones, df[columna], width=0.2, label=columna)` agrega las barras para la materia actual.

6. Configurar las etiquetas del eje X:

- `ax.set_xticks([i + 0.2 for i in range(len(df))])` establece las posiciones de las etiquetas en el eje X.
- `ax.set_xticklabels(df['Nombre'], rotation=45, ha='right')` ajusta las etiquetas con una rotación de 45 grados para evitar encimarse.

7. Agregar leyenda:

- `ax.legend()` agrega una leyenda al gráfico.

8. Agregar títulos y etiquetas:

- `ax.set_title('Calificaciones de los Alumnos')` agrega un título al gráfico.
- `ax.set_xlabel('Alumnos')` y `ax.set_ylabel('Calificaciones')` agregan etiquetas a los ejes X e Y respectivamente.

9. Mostrar el gráfico:

- `plt.tight_layout()` ajusta el diseño para que no haya solapamientos.
- `plt.show()` muestra el gráfico.

Este programa debería ayudarte a visualizar las calificaciones de los alumnos de una manera clara y organizada.



como importo las bibliotecas



Para importar las bibliotecas necesarias en Python, debes asegurarte de que estén instaladas en tu entorno. Las bibliotecas que necesitamos para este programa son `pandas` y `matplotlib`. Aquí te muestro cómo instalarlas y cómo importarlas en tu código.

Instalación de las bibliotecas

Si no tienes `pandas` y `matplotlib` instalados, puedes instalarlas usando `pip`. Abre una terminal o un símbolo del sistema y ejecuta los siguientes comandos:

```
sh
```

```
pip install pandas matplotlib
```

Importación de las bibliotecas

Una vez que hayas instalado las bibliotecas, puedes importarlas al inicio de tu script de Python. Aquí tienes el código completo, incluyendo las importaciones necesarias:

```
python
```

```
"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Luego, grafica
estas calificaciones en un gráfico de barras para cada alumno, asegurándose de que
las etiquetas en el eje X no se encimen.
"""
```

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)
```

```
# Seleccionar columnas de interés
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_ProgramacionOOP',
'Mat_EstructuraDatos']
```

```
# Crear el gráfico
fig, ax = plt.subplots(figsize=(10, 6))
```

```
# Iterar sobre cada columna de calificación para crear una serie de barras
for idx, columna in enumerate(columnas_calificaciones):
    # Ajustar las posiciones de las barras para que no se encimen
```

```
posiciones = [i + idx*0.2 for i in range(len(df))]
ax.bar(posiciones, df[columna], width=0.2, label=columna)

# Configurar las etiquetas del eje X
ax.set_xticks([i + 0.2 for i in range(len(df))])
ax.set_xticklabels(df['Nombre'], rotation=45, ha='right')

# Agregar leyenda
ax.legend()

# Agregar títulos y etiquetas
ax.set_title('Calificaciones de los Alumnos')
ax.set_xlabel('Alumnos')
ax.set_ylabel('Calificaciones')

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```

Explicación de las importaciones

- `import pandas as pd`: Esto importa la biblioteca `pandas` y le asigna el alias `pd`, que es una convención común y facilita el uso de sus funciones.
- `import matplotlib.pyplot as plt`: Esto importa la subbiblioteca `pyplot` de `matplotlib` y le asigna el alias `plt`, lo que también es una convención común y facilita la creación de gráficos.

Con estas importaciones al principio de tu script, el resto del código funcionará correctamente, permitiéndote leer el archivo Excel y graficar las calificaciones de los alumnos.



El programa me marca el error, con éste texto: Traceback (most recent call last):
 File "C:\Users\acost\Documents\ITP\Python 360\codigos\M2.1 Automatizando Excel\GraficaAlumnos.py", line 8, in <module>
 import pandas as pd
 ModuleNotFoundError: No module named 'pandas'



El error `ModuleNotFoundError: No module named 'pandas'` indica que la biblioteca `pandas` no está instalada en tu entorno de Python. Para resolver este problema, necesitas instalar `pandas` y `matplotlib` usando `pip`.

Pasos para instalar `pandas` y `matplotlib`

1. **Abrir la terminal o símbolo del sistema:**
 - En Windows, puedes abrir el símbolo del sistema (Cmd) o PowerShell.
 - En macOS y Linux, puedes abrir la terminal.
2. **Ejecutar los siguientes comandos para instalar las bibliotecas:**
 - Para instalar `pandas`:

```
sh
```

```
pip install pandas
```

- o Para instalar `matplotlib`:

```
sh
```

```
pip install matplotlib
```

Si estás usando un entorno virtual (virtual environment), asegúrate de que esté activado antes de ejecutar los comandos de instalación.

Ejemplo de activación de un entorno virtual

1. Crear un entorno virtual:

```
sh
```

```
python -m venv mi_entorno
```

2. Activar el entorno virtual:

- o En Windows:

```
sh
```

```
mi_entorno\Scripts\activate
```

- o En macOS y Linux:

```
sh
```

```
source mi_entorno/bin/activate
```

3. Instalar las bibliotecas en el entorno virtual:

```
sh
```

```
pip install pandas matplotlib
```

Verificación de la instalación

Después de instalar las bibliotecas, verifica que se instalaron correctamente ejecutando el siguiente script de Python:

python

```
import pandas as pd
import matplotlib.pyplot as plt

print("Pandas version:", pd.__version__)
print("Matplotlib version:", plt.__version__)
```

Si no hay errores y se imprimen las versiones de `pandas` y `matplotlib`, la instalación fue exitosa. Ahora, puedes ejecutar el programa original sin problemas.

Código completo con importaciones

A continuación, te dejo nuevamente el código completo para que lo ejecutes después de asegurarte de que las bibliotecas estén instaladas:

python

```
"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Luego, grafica
estas calificaciones en un gráfico de barras para cada alumno, asegurándose de que
las etiquetas en el eje X no se encimen.
"""

import pandas as pd
import matplotlib.pyplot as plt

# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)

# Seleccionar columnas de interés
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_ProgramacionOOP',
                           'Mat_EstructuraDatos']

# Crear el gráfico
fig, ax = plt.subplots(figsize=(10, 6))

# Iterar sobre cada columna de calificación para crear una serie de barras
for idx, columna in enumerate(columnas_calificaciones):
    # Ajustar las posiciones de las barras para que no se encimen
    posiciones = [i + idx*0.2 for i in range(len(df))]
    ax.bar(posiciones, df[columna], width=0.2, label=columna)

# Configurar las etiquetas del eje X
ax.set_xticks([i + 0.2 for i in range(len(df))])
ax.set_xticklabels(df['Nombre'], rotation=45, ha='right')

# Agregar leyenda
ax.legend()

# Agregar títulos y etiquetas
ax.set_title('Calificaciones de los Alumnos')
ax.set_xlabel('Alumnos')
ax.set_ylabel('Calificaciones')

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```

Este código debería ejecutarse correctamente una vez que las bibliotecas necesarias estén instaladas.



me sigue marcando el error: Traceback (most recent call last):
 File "C:\Users\acost\Documents\ITP\Python 360\codigos\M2.1 Automatizando Excel\GraficaAlumnos.py", line 8, in <module>
 import pandas as pd
 ModuleNotFoundError: No module named 'pandas'. Ayudame a eliminar los errores con un código diferente



Para asegurarte de que las bibliotecas `pandas` y `matplotlib` estén correctamente instaladas y el programa funcione, vamos a seguir estos pasos:

1. Verifica la instalación de las bibliotecas.
2. Proporciona un código alternativo para manejar el archivo Excel y graficar las calificaciones.

Verificar la instalación de las bibliotecas

Abre una terminal o símbolo del sistema y ejecuta los siguientes comandos para instalar `pandas` y `matplotlib`:

```
sh
```

```
pip install pandas matplotlib
```

Probar la instalación

Crea un archivo Python separado y prueba si las bibliotecas están correctamente instaladas:

```
python
```

```
import pandas as pd
import matplotlib.pyplot as plt

print("Pandas version:", pd.__version__)
print("Matplotlib version:", plt.__version__)
```

Ejecuta este archivo y asegúrate de que se imprimen las versiones de `pandas` y `matplotlib`.

Código alternativo para leer el archivo Excel y graficar las calificaciones

A continuación te proporciono un código alternativo que debería funcionar correctamente una vez que las bibliotecas estén instaladas:

```
python
```

```
"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Luego, grafica
estas calificaciones en un gráfico de barras para cada alumno, asegurándose de que
las etiquetas en el eje X no se encimen.
"""
```



```
import pandas as pd
import matplotlib.pyplot as plt

try:
    # Leer el archivo Excel
    archivo_excel = "calificaciones_alumnos.xlsx"
    df = pd.read_excel(archivo_excel)

    # Verificar que el archivo fue leído correctamente
    print("Datos leídos del archivo Excel:")
    print(df.head())

    # Seleccionar columnas de interés
    columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_Programacion00P',
'Mat_EstructuraDatos']

    # Crear el gráfico
    fig, ax = plt.subplots(figsize=(10, 6))

    # Iterar sobre cada columna de calificación para crear una serie de barras
    for idx, columna in enumerate(columnas_calificaciones):
        # Ajustar las posiciones de las barras para que no se encimen
        posiciones = [i + idx*0.2 for i in range(len(df))]
        ax.bar(posiciones, df[columna], width=0.2, label=columna)

    # Configurar las etiquetas del eje X
    ax.set_xticks([i + 0.2 for i in range(len(df))])
    ax.set_xticklabels(df['Nombre'], rotation=45, ha='right')

    # Agregar leyenda
    ax.legend()

    # Agregar títulos y etiquetas
    ax.set_title('Calificaciones de los Alumnos')
    ax.set_xlabel('Alumnos')
    ax.set_ylabel('Calificaciones')

    # Mostrar el gráfico
    plt.tight_layout()
    plt.show()
except FileNotFoundError:
    print(f"Error: No se encontró el archivo {archivo_excel}. Asegúrate de que el archivo existe en el directorio especificado.")
except Exception as e:
    print(f"Error: {e}")
```

Pasos a seguir:

1. Instalación de bibliotecas:

- Asegúrate de instalar `pandas` y `matplotlib` ejecutando `pip install pandas matplotlib`.

2. Verificación de la instalación:

- Crea un archivo Python con el código de verificación y ejecútalo para asegurarte de que las bibliotecas están instaladas correctamente.

3. Ejecutar el programa principal:

- Guarda el código proporcionado en un archivo Python y ejecútalo. Asegúrate de que el archivo `calificaciones_alumnos.xlsx` esté en el mismo directorio que el script o proporciona la ruta completa al archivo.

Este código maneja posibles errores, como la falta del archivo Excel, y muestra un mensaje adecuado si ocurre algún problema. También imprime los datos leídos del archivo Excel para verificar que se ha leído correctamente antes de intentar graficar las calificaciones.



Tengo un archivo llamado "calificaciones_alumnos.xlsx", el cual tiene un campo llamado "Nombre" y varias columnas de calificaciones llamadas Mat_CalculoIntegral, Mat_ProgramacionOOP y Mat_EstructuraDatos. Ayúdame a crear un programa en Python para agregar una columna llamada Mat_Fisica que contenga valores aleatorios entre 0 y 100 con un decimal, al inicio del código agrega un comentario explicando lo que hace el programa y documenta las líneas importantes del código



A continuación te proporciono un programa en Python que agrega una columna llamada `Mat_Fisica` al archivo `calificaciones_alumnos.xlsx`. Esta columna contendrá valores aleatorios entre 0 y 100 con un decimal. El programa usa las bibliotecas `pandas` para manipular el archivo Excel y `numpy` para generar los valores aleatorios.

Instalación de bibliotecas

Primero, asegúrate de tener instaladas las bibliotecas necesarias:

```
sh
```

```
pip install pandas numpy openpyxl
```

Código del programa

```
python
```

```
"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Luego, agrega una
columna llamada "Mat_Fisica" que contiene valores aleatorios entre 0 y 100 con un decimal
y guarda el archivo con la nueva columna.
"""

import pandas as pd
import numpy as np

# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)

# Generar valores aleatorios entre 0 y 100 con un decimal para la columna "Mat_Fisica"
df['Mat_Fisica'] = np.round(np.random.uniform(0, 100, len(df)), 1)

# Guardar el DataFrame modificado de nuevo en el archivo Excel
df.to_excel(archivo_excel, index=False)

print("Se ha agregado la columna 'Mat_Fisica' con valores aleatorios entre 0 y 100.")
```

Explicación del código

1. Importación de bibliotecas:

python

```
import pandas as pd
import numpy as np
```

Se importan las bibliotecas `pandas` para la manipulación del archivo Excel y `numpy` para generar valores aleatorios.

2. Leer el archivo Excel:

python

```
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)
```

Se lee el archivo Excel y se carga en un DataFrame de `pandas`.

3. Generar valores aleatorios para la columna "Mat_Fisica":

python

```
df['Mat_Fisica'] = np.round(np.random.uniform(0, 100, len(df)), 1)
```

Se generan valores aleatorios entre 0 y 100 con un decimal utilizando `numpy` y se agregan como una nueva columna llamada `Mat_Fisica` al DataFrame.

4. Guardar el DataFrame modificado en el archivo Excel:

python

```
df.to_excel(archivo_excel, index=False)
```

Se guarda el DataFrame modificado de nuevo en el archivo Excel, sobrescribiendo el archivo original sin incluir el índice de los DataFrame.

5. Mensaje de confirmación:

python

```
print("Se ha agregado la columna 'Mat_Fisica' con valores aleatorios entre 0 y 100.")
```

Se imprime un mensaje para confirmar que se ha agregado la nueva columna correctamente.

Ejecución del programa

Guarda el código anterior en un archivo Python, asegúrate de que el archivo `calificaciones_alumnos.xlsx` esté en el mismo directorio que el script o proporciona la ruta completa al archivo, y ejecuta el script. Esto agregará la columna `Mat_Fisica` con valores aleatorios y guardará los cambios en el archivo Excel.

Quiero saber cuántos registros y cuántos campos tiene mi tabla, al inicio del código agrega un comentario explicando lo que hace el programa y documenta las líneas importantes del código



Claro, a continuación te proporciono un programa en Python que agrega una columna llamada `Mat_Fisica` con valores aleatorios entre 0 y 100 con un decimal al archivo `calificaciones_alumnos.xlsx`, y también te muestra cuántos registros y campos tiene tu tabla. El programa usa `pandas` para manipular el archivo Excel y `numpy` para generar los valores aleatorios.

Instalación de bibliotecas

Primero, asegúrate de tener instaladas las bibliotecas necesarias:

```
sh
```

```
pip install pandas numpy openpyxl
```

Código del programa

```
python
```

```
"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Luego, agrega una
columna llamada "Mat_Fisica" que contiene valores aleatorios entre 0 y 100 con un decimal,
cuenta el número de registros (filas) y campos (columnas) en la tabla, y guarda el archivo
con la nueva columna.
"""

import pandas as pd
import numpy as np

# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)

# Obtener el número de registros (filas) y campos (columnas)
num_registros = len(df)
num_campos = len(df.columns)

# Imprimir el número de registros y campos
print(f"La tabla tiene {num_registros} registros y {num_campos} campos.")

# Generar valores aleatorios entre 0 y 100 con un decimal para la columna "Mat_Fisica"
df['Mat_Fisica'] = np.round(np.random.uniform(0, 100, len(df)), 1)

# Guardar el DataFrame modificado de nuevo en el archivo Excel
df.to_excel(archivo_excel, index=False)

print("Se ha agregado la columna 'Mat_Fisica' con valores aleatorios entre 0 y 100.")
```

Explicación del código

1. Importación de bibliotecas:

python

```
import pandas as pd
import numpy as np
```

Se importan las bibliotecas `pandas` para la manipulación del archivo Excel y `numpy` para generar valores aleatorios.

2. Leer el archivo Excel:

python

```
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)
```

Se lee el archivo Excel y se carga en un DataFrame de `pandas`.

3. Obtener el número de registros (filas) y campos (columnas):

python

```
num_registros = len(df)
num_campos = len(df.columns)
```

Se cuenta el número de filas y columnas en el DataFrame.

4. Imprimir el número de registros y campos:

python

```
print(f"La tabla tiene {num_registros} registros y {num_campos} campos.")
```

Se imprime el número de registros y campos en la tabla.

5. Generar valores aleatorios para la columna "Mat_Fisica":

python

```
df['Mat_Fisica'] = np.round(np.random.uniform(0, 100, len(df)), 1)
```

Se generan valores aleatorios entre 0 y 100 con un decimal utilizando `numpy` y se agregan como una nueva columna llamada `Mat_Fisica` al DataFrame.

6. Guardar el DataFrame modificado en el archivo Excel:

python

```
df.to_excel(archivo_excel, index=False)
```

Se guarda el DataFrame modificado de nuevo en el archivo Excel, sobrescribiendo el archivo original sin incluir el índice de los DataFrame.

7. Mensaje de confirmación:

```
python
```

```
print("Se ha agregado la columna 'Mat_Fisica' con valores aleatorios entre 0 y 100.")
```

Se imprime un mensaje para confirmar que se ha agregado la nueva columna correctamente.

Ejecución del programa

Guarda el código anterior en un archivo Python, asegúrate de que el archivo ``calificaciones_alumnos.xlsx`` esté en el mismo directorio que el script o proporciona la ruta completa al archivo, y ejecuta el script. Esto agregará la columna ``Mat_Fisica`` con valores aleatorios, mostrará el número de registros y campos en la tabla, y guardará los cambios en el archivo Excel.



Tengo un archivo llamado "calificaciones_alumnos.xlsx", el cual tiene un campo llamado "Nombre" y varias columnas de calificaciones llamadas Mat_CalculoIntegral, Mat_ProgramacionOOP y Mat_EstructuraDatos. Ayúdame a crear un programa en Python para encontrar el promedio total de calificaciones, al inicio del código agrega un comentario explicando lo que hace el programa y documenta las líneas importantes del código



Claro, a continuación te proporciono un programa en Python que lee un archivo Excel llamado ``calificaciones_alumnos.xlsx``, calcula el promedio total de las calificaciones en las columnas ``Mat_CalculoIntegral``, ``Mat_ProgramacionOOP`` y ``Mat_EstructuraDatos``, y lo imprime.

Instalación de bibliotecas

Primero, asegúrate de tener instaladas las bibliotecas necesarias:

```
sh
```

```
pip install pandas openpyxl
```

Código del programa

```
python
```

```
"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Luego, calcula el
promedio total de las calificaciones en las columnas "Mat_CalculoIntegral",
"Mat_ProgramacionOOP" y "Mat_EstructuraDatos", y lo imprime.
"""
```

```
import pandas as pd

# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)

# Seleccionar las columnas de calificaciones
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_ProgramacionOOP',
'Mat_EstructuraDatos']

# Calcular el promedio total de las calificaciones
promedio_total = df[columnas_calificaciones].mean().mean()

# Imprimir el promedio total de las calificaciones
print(f"El promedio total de las calificaciones es: {promedio_total:.2f}")
```

Explicación del código

1. Importación de la biblioteca:

python

```
import pandas as pd
```

Se importa la biblioteca `pandas` para la manipulación del archivo Excel.

2. Leer el archivo Excel:

python

```
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)
```

Se lee el archivo Excel y se carga en un DataFrame de `pandas`.

3. Seleccionar las columnas de calificaciones:

python

```
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_ProgramacionOOP',
'Mat_EstructuraDatos']
```

Se define una lista con los nombres de las columnas que contienen las calificaciones.

4. Calcular el promedio total de las calificaciones:

python

```
promedio_total = df[columnas_calificaciones].mean().mean()
```

- `df[columnas_calificaciones]` selecciona solo las columnas de calificaciones del DataFrame.
- `.mean()` calcula el promedio de cada columna.

- `.mean()` de nuevo calcula el promedio de esos promedios, obteniendo así el promedio total de todas las calificaciones.

5. Imprimir el promedio total de las calificaciones:

```
python
```

```
print(f"El promedio total de las calificaciones es: {promedio_total:.2f}")
```

Se imprime el promedio total de las calificaciones con dos decimales.

Ejecución del programa

Guarda el código anterior en un archivo Python, asegúrate de que el archivo `calificaciones_alumnos.xlsx` esté en el mismo directorio que el script o proporciona la ruta completa al archivo, y ejecuta el script. Esto calculará el promedio total de las calificaciones y lo imprimirá en la consola.



Tengo un archivo llamado "calificaciones_alumnos.xlsx", el cual tiene un campo llamado "Nombre" y varias columnas de calificaciones llamadas Mat_CalculoIntegral, Mat_ProgramacionOOP y Mat_EstructuraDatos. Ayúdame a crear un programa en Python para encontrar el promedio más alto de calificaciones, al inicio del código agrega un comentario explicando lo que hace el programa y documenta las líneas importantes del código



A continuación, te proporciono un programa en Python que lee un archivo Excel llamado `calificaciones_alumnos.xlsx`, encuentra el promedio más alto de las calificaciones en las columnas `Mat_CalculoIntegral`, `Mat_ProgramacionOOP` y `Mat_EstructuraDatos`, y lo imprime junto con el nombre del alumno correspondiente.

Instalación de bibliotecas

Primero, asegúrate de tener instaladas las bibliotecas necesarias:

```
sh
```

```
pip install pandas openpyxl
```

Código del programa

```
python
```

```
"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Luego, encuentra
el promedio más alto de las calificaciones en las columnas "Mat_CalculoIntegral",
```



```

"Mat_ProgramacionOOP" y "Mat_EstructuraDatos", e imprime el nombre del alumno
correspondiente y el promedio.
"""

import pandas as pd

# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)

# Seleccionar las columnas de calificaciones
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_ProgramacionOOP',
'Mat_EstructuraDatos']

# Calcular el promedio de las calificaciones para cada alumno
df['Promedio'] = df[columnas_calificaciones].mean(axis=1)

# Encontrar el promedio más alto y el nombre del alumno correspondiente
promedio_mas_alto = df['Promedio'].max()
alumno_con_promedio_mas_alto = df.loc[df['Promedio'] == promedio_mas_alto,
'Nombre'].values[0]

# Imprimir el nombre del alumno con el promedio más alto y el promedio
print(f"El alumno con el promedio más alto es: {alumno_con_promedio_mas_alto} con un
promedio de {promedio_mas_alto:.2f}")

```

Explicación del código

1. Importación de la biblioteca:

```
python
```

```
import pandas as pd
```

Se importa la biblioteca `pandas` para la manipulación del archivo Excel.

2. Leer el archivo Excel:

```
python
```

```
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)
```

Se lee el archivo Excel y se carga en un DataFrame de `pandas`.

3. Seleccionar las columnas de calificaciones:

```
python
```

```
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_ProgramacionOOP',
'Mat_EstructuraDatos']
```

Se define una lista con los nombres de las columnas que contienen las calificaciones.

4. Calcular el promedio de las calificaciones para cada alumno:

python

```
df['Promedio'] = df[columnas_calificaciones].mean(axis=1)
```

Se calcula el promedio de las calificaciones de cada alumno y se agrega como una nueva columna llamada `Promedio` en el DataFrame. El parámetro `axis=1` indica que el promedio se calcula a lo largo de las filas (para cada alumno).

5. Encontrar el promedio más alto y el nombre del alumno correspondiente:

python

```
promedio_mas_alto = df['Promedio'].max()
alumno_con_promedio_mas_alto = df.loc[df['Promedio'] == promedio_mas_alto,
'Nombre'].values[0]
```

- `df['Promedio'].max()` obtiene el valor del promedio más alto.
- `df.loc[df['Promedio'] == promedio_mas_alto, 'Nombre'].values[0]` obtiene el nombre del alumno con el promedio más alto.

6. Imprimir el nombre del alumno con el promedio más alto y el promedio:

python

```
print(f"El alumno con el promedio más alto es: {alumno_con_promedio_mas_alto} con un promedio de {promedio_mas_alto:.2f}")
```

Se imprime el nombre del alumno con el promedio más alto y el valor del promedio con dos decimales.

Ejecución del programa

Guarda el código anterior en un archivo Python, asegúrate de que el archivo `calificaciones_alumnos.xlsx` esté en el mismo directorio que el script o proporciona la ruta completa al archivo, y ejecuta el script. Esto calculará el promedio más alto de las calificaciones y lo imprimirá junto con el nombre del alumno correspondiente.



Tengo un archivo llamado "calificaciones_alumnos.xlsx", el cual tiene un campo llamado "Nombre" y varias columnas de calificaciones llamadas Mat_CalculoIntegral, Mat_ProgramacionOOP y Mat_EstructuraDatos. Ayúdame a crear un programa en Python para encontrar los registros que tengan un promedio menor a 70, al inicio del código agrega un comentario explicando lo que hace el programa y documenta las líneas importantes del código



Aquí tienes un programa en Python que lee un archivo Excel llamado `calificaciones_alumnos.xlsx`, calcula el promedio de las calificaciones para cada alumno, y encuentra los registros cuyos promedios

sean menores a 70. El programa también imprimirá estos registros.

Instalación de bibliotecas

Primero, asegúrate de tener instaladas las bibliotecas necesarias:

```
sh

pip install pandas openpyxl
```

Código del programa

```
python

"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Calcula el promedio
de las calificaciones en las columnas "Mat_CalculoIntegral", "Mat_ProgramacionOOP" y
"Mat_EstructuraDatos", y encuentra los registros cuyos promedios son menores a 70.
Finalmente, imprime estos registros.
"""

import pandas as pd

# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)

# Seleccionar las columnas de calificaciones
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_ProgramacionOOP',
'Mat_EstructuraDatos']

# Calcular el promedio de las calificaciones para cada alumno
df['Promedio'] = df[columnas_calificaciones].mean(axis=1)

# Encontrar los registros con promedio menor a 70
registros_bajo_70 = df[df['Promedio'] < 70]

# Imprimir los registros con promedio menor a 70
print("Registros con promedio menor a 70:")
print(registros_bajo_70)
```

Explicación del código

1. Importación de la biblioteca:

```
python

import pandas as pd
```

Se importa la biblioteca `pandas` para manipular el archivo Excel.

2. Leer el archivo Excel:

python

```
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)
```

Se lee el archivo Excel y se carga en un DataFrame de `pandas`.

3. Seleccionar las columnas de calificaciones:

python

```
columnas_calificaciones = ['Mat_CalculoIntegral', 'Mat_Programacion00P',
'Mat_EstructuraDatos']
```

Se define una lista con los nombres de las columnas que contienen las calificaciones.

4. Calcular el promedio de las calificaciones para cada alumno:

python

```
df['Promedio'] = df[columnas_calificaciones].mean(axis=1)
```

Se calcula el promedio de las calificaciones de cada alumno y se agrega como una nueva columna llamada `Promedio` en el DataFrame. El parámetro `axis=1` indica que el promedio se calcula a lo largo de las filas (para cada alumno).

5. Encontrar los registros con promedio menor a 70:

python

```
registros_bajo_70 = df[df['Promedio'] < 70]
```

Se filtran los registros cuyos promedios son menores a 70.

6. Imprimir los registros con promedio menor a 70:

python

```
print("Registros con promedio menor a 70:")
print(registros_bajo_70)
```

Se imprimen los registros que cumplen con la condición de tener un promedio menor a 70.

Ejecución del programa

Guarda el código en un archivo Python, asegúrate de que el archivo `calificaciones_alumnos.xlsx` esté en el mismo directorio que el script o proporciona la ruta completa al archivo, y ejecuta el script. Esto calculará el promedio de las calificaciones, filtrará los registros con promedios menores a 70, e imprimirá estos registros.



Tengo un archivo llamado "calificaciones_alumnos.xlsx", el cual tiene un campo llamado "Nombre" y varias columnas de calificaciones llamadas Mat_CalculoIntegral, Mat_ProgramacionOOP y Mat_EstructuraDatos. Ayúdame a crear un programa en Python para encontrar todos los nombres que contengan Martinez, al inicio del código agrega un comentario explicando lo que hace el programa y documenta las líneas importantes del código



Claro, aquí tienes un programa en Python que lee un archivo Excel llamado `calificaciones_alumnos.xlsx`, busca todos los nombres que contengan la cadena "Martinez" y los imprime. Este programa utiliza `pandas` para leer y manipular el archivo Excel.

Instalación de bibliotecas

Primero, asegúrate de tener instalada la biblioteca necesaria:

```
sh

pip install pandas openpyxl
```

Código del programa

```
python

"""
Este programa lee un archivo Excel llamado "calificaciones_alumnos.xlsx" que contiene
los nombres de los alumnos y sus calificaciones en varias materias. Busca todos los
nombres que contengan la cadena "Martinez" y los imprime.
"""

import pandas as pd

# Leer el archivo Excel
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)

# Filtrar los nombres que contienen "Martinez"
nombres_con_martinez = df[df['Nombre'].str.contains('Martinez', case=False, na=False)]

# Imprimir los nombres que contienen "Martinez"
print("Nombres que contienen 'Martinez':")
print(nombres_con_martinez['Nombre'])
```

Explicación del código

1. Importación de la biblioteca:

```
python

import pandas as pd
```

Se importa la biblioteca `pandas` para la manipulación del archivo Excel.

2. Leer el archivo Excel:

python

```
archivo_excel = "calificaciones_alumnos.xlsx"
df = pd.read_excel(archivo_excel)
```

Se lee el archivo Excel y se carga en un DataFrame de `pandas`.

3. Filtrar los nombres que contienen "Martinez":

python

```
nombres_con_martinez = df[df['Nombre'].str.contains('Martinez', case=False,
na=False)]
```

- `df['Nombre'].str.contains('Martinez', case=False, na=False)` busca la cadena "Martinez" en la columna `Nombre`, sin distinguir entre mayúsculas y minúsculas (`case=False`) y manejando valores faltantes (`na=False`).
- `df[df['Nombre'].str.contains(...)]` filtra las filas donde el nombre contiene la cadena "Martinez".

4. Imprimir los nombres que contienen "Martinez":

python

```
print("Nombres que contienen 'Martinez':")
print(nombres_con_martinez['Nombre'])
```

Se imprime la columna `Nombre` de los registros que contienen "Martinez".

Ejecución del programa

Guarda el código en un archivo Python, asegúrate de que el archivo `calificaciones_alumnos.xlsx` esté en el mismo directorio que el script o proporciona la ruta completa al archivo, y ejecuta el script. Esto imprimirá todos los nombres que contengan "Martinez".

ChatGPT puede cometer errores. Comprueba la información importante.