

## Exercise 4

```
In [1]:
```

```
import pandas as pd
# Leer los datos
import cv2
# Leer los datos
import matplotlib.pyplot as plt
import seaborn as sns
```

### Importar datos con Pandas

```
In [1]:
```

```
daily_prices = pd.read_csv('data/stock_daily_prices.csv', parse_dates=['date'])
daily_returns = pd.read_csv('data/stock_daily_returns.csv', parse_dates=['date'])

print(daily_prices)
print(daily_returns)

0      2008-01-02    5.849100    29.448424    47.058339    29.915334
1      2008-01-03    5.952452    29.246094    47.058338    29.690800
2      2008-01-04    5.498071    28.182682    46.870398    31.000002
3      2008-01-07    5.424478    28.868999    47.175980    29.180000
4      2008-01-08    5.229301    27.723747    47.239629    28.899999
...
3637 2022-06-13    131.880005    115.989998    138.149994    37.020000
3638 2022-06-14    132.709995    114.030998    133.839996    36.890000
3639 2022-06-15    135.429993    115.410004    132.509995    37.889999
3640 2022-06-16    130.859996    113.430000    132.300001    34.779999
3641 2022-06-17    131.559998    113.029999    132.360001    36.279999

[3642 rows x 5 columns]

0      2008-01-02    0.000000    0.000000    0.000000    0.000000
1      2008-01-03    0.046203    -0.687708    0.000000    -0.700233
2      2008-01-04    -0.420517    -2.268378    -0.401072    4.426454
3      2008-01-07    -1.338518    1.001718    0.652615    -5.870973
4      2008-01-08    -3.597157    -3.967066    0.262103    -16.413356
...
3637 2022-06-13    -3.828484    -2.977838    -2.677001    -10.083008
3638 2022-06-14    0.467020    -1.663937    -3.137796    -0.201034
3639 2022-06-15    2.011147    1.183593    -0.993725    2.430884
3640 2022-06-16    -3.861543    -1.715023    0.011384    -8.229792
3641 2022-06-17    1.153314    -0.352642    -0.720077    4.312824

[3642 rows x 5 columns]
```

Diagramas de líneas que muestran los precios diarios de las 4 acciones en una sola cifra.

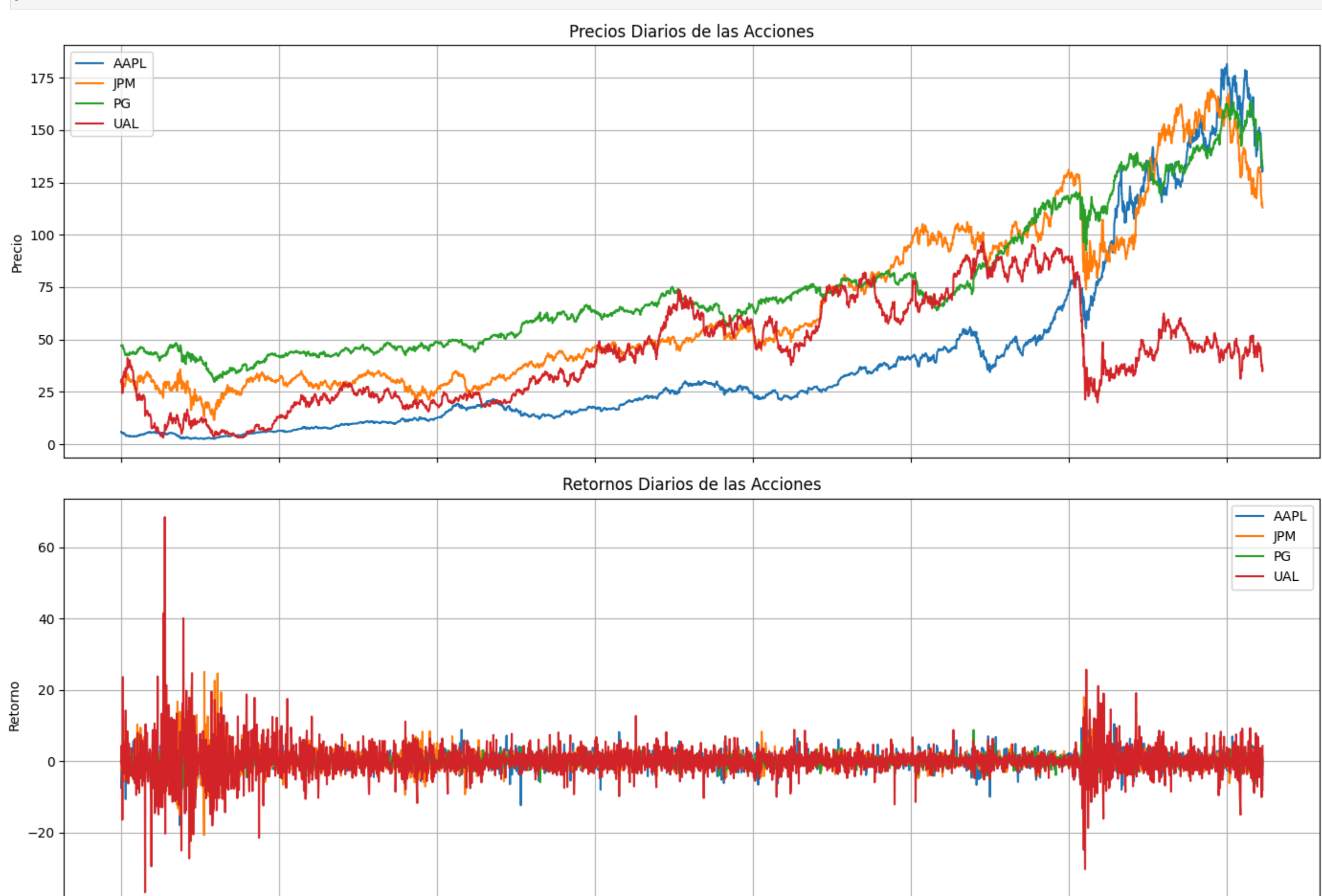
```
In [1]:
```

```
# Configuración de la figura
fig, axes = plt.subplots(4, 1, figsize=(14, 10), sharex=True)

# Graficar precios diarios
for column in ['AAPL', 'JPM', 'PG', 'UAL']:
    axes[0].plot(daily_prices['date'], daily_prices[column], label=column)
    axes[0].set_title(f'Precios Diarios de las Acciones')
    axes[0].set_ylabel('Precio')
    axes[0].legend()
    axes[0].grid(True)

# Graficar retornos diarios
for column in ['AAPL', 'JPM', 'PG', 'UAL']:
    axes[1].plot(daily_returns['date'], daily_returns[column], label=column)
    axes[1].set_title(f'Retornos Diarios de las Acciones')
    axes[1].set_ylabel('Retorno')
    axes[1].set_xlabel('Fecha')
    axes[1].legend()
    axes[1].grid(True)

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



### 4 precios diarios de acciones en varias subplots

```
In [1]:
```

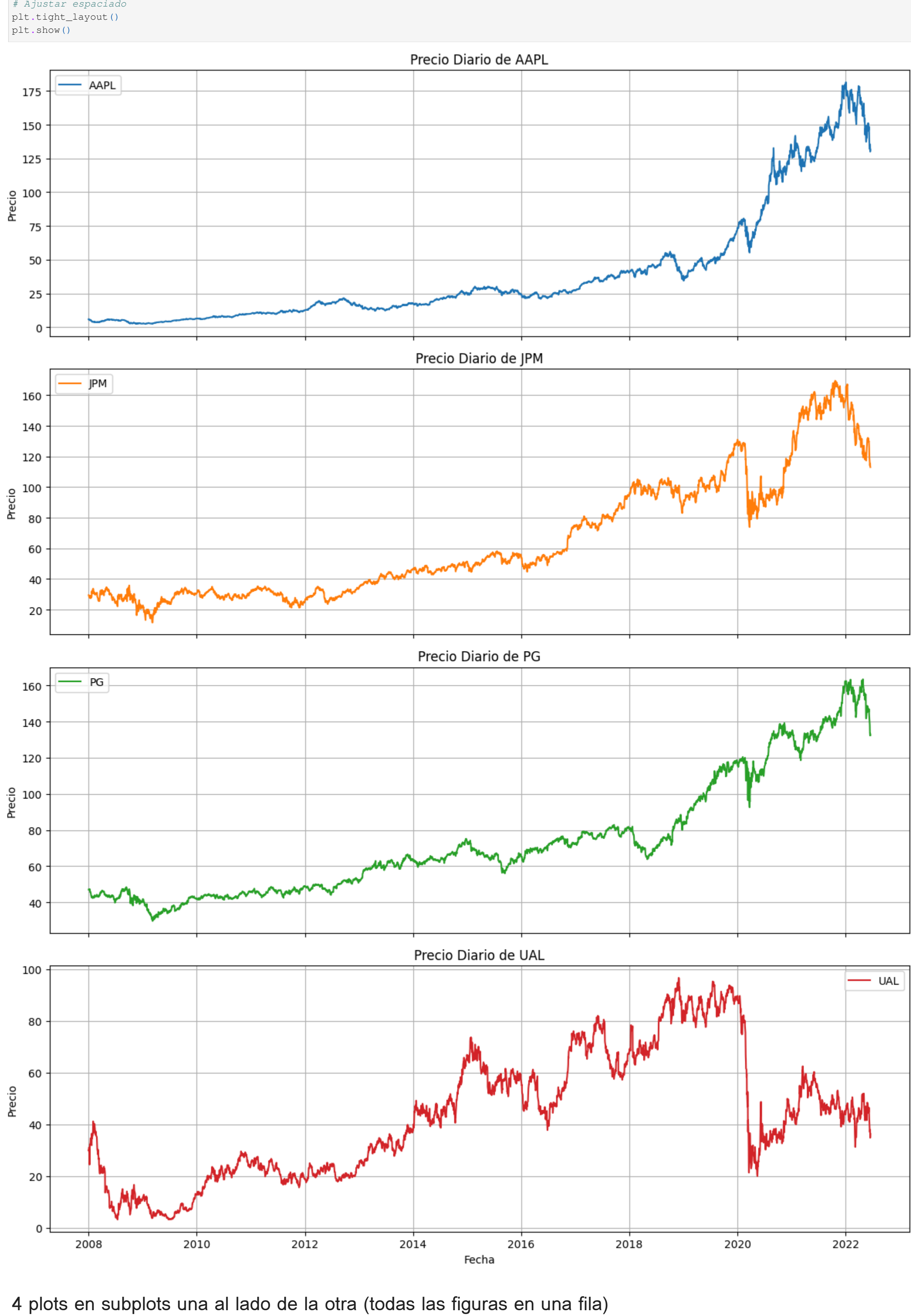
```
# Configuración de la figura y subplots
fig, axes = plt.subplots(4, 1, figsize=(12, 10), sharex=True)

# Lista de nombres de las acciones
stocks = ['AAPL', 'JPM', 'PG', 'UAL']

# Crear un subplot para cada acción
for i, stock in enumerate(stocks):
    axes[i].plot(daily_prices['date'], daily_prices[stock], label=stock, color='C'+str(i))
    axes[i].set_title(f'Precio Diario de {stock}')
    axes[i].set_ylabel('Precio')
    axes[i].legend()
    axes[i].grid(True)

# Etiqueta común para el eje X
axes[-1].set_xlabel('Fecha')

# Ajustar espaciado
plt.tight_layout()
plt.show()
```



### 4 plots en subplots una al lado de la otra (todas las figuras en una fila)

```
In [1]:
```

```
# Configuración de la figura y subplots
fig, axes = plt.subplots(1, 4, figsize=(20, 5), sharey=True)

# Lista de nombres de las acciones
stocks = ['AAPL', 'JPM', 'PG', 'UAL']

# Crear un subplot horizontal para cada acción
for i, stock in enumerate(stocks):
    axes[i].plot(daily_prices['date'], daily_prices[stock], label=stock, color='C'+str(i))
    axes[i].set_title(f'Precio de {stock}')
    axes[i].set_xlabel('Fecha')
    axes[i].legend()
    axes[i].grid(True)

# Etiqueta común para el eje Y
axes[0].set_ylabel('Precio')

# Ajustar espaciado
plt.tight_layout()
plt.show()
```



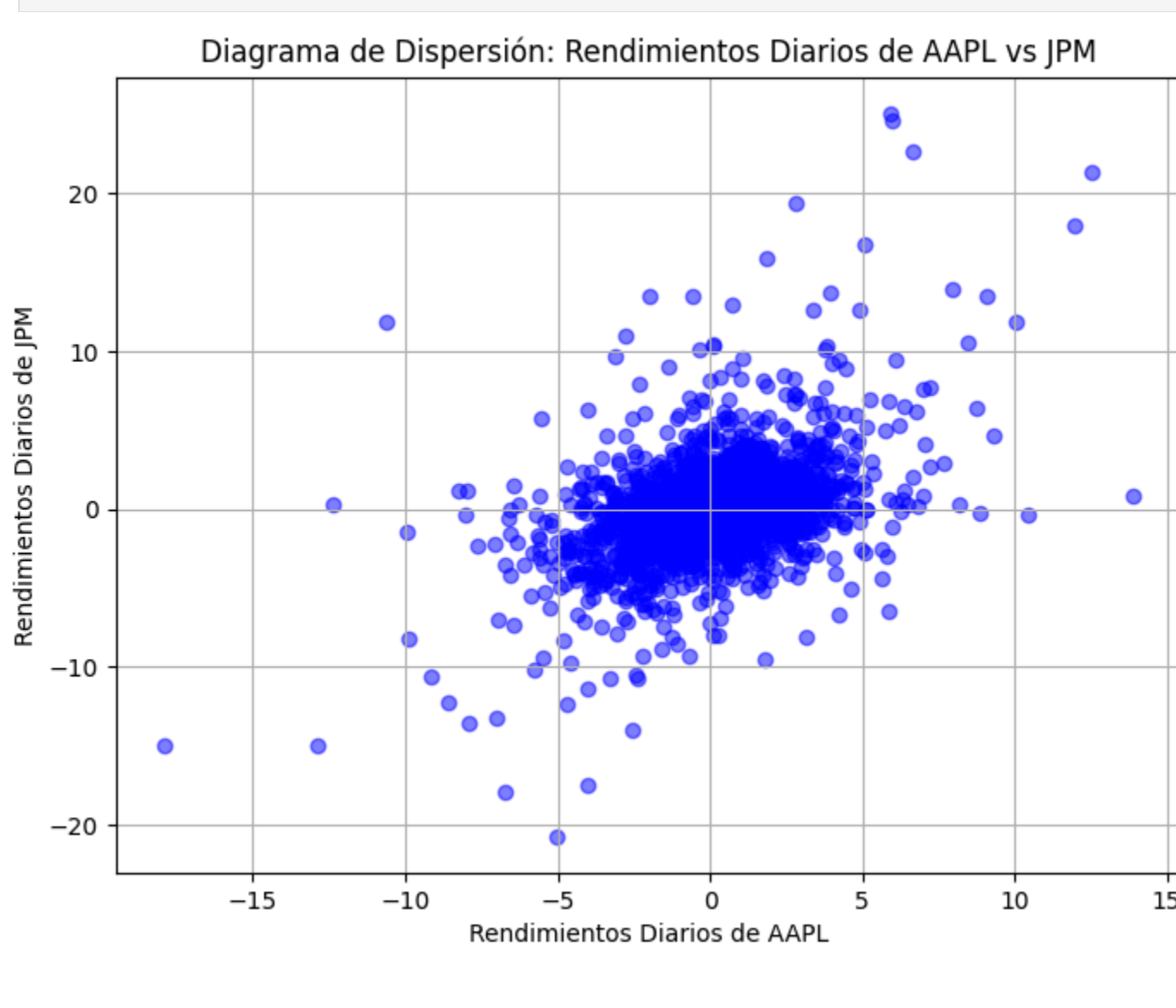
### Diagrama de dispersión entre los rendimientos diarios de Apple (AAPL) y JP Morgan (JPM)

```
In [1]:
```

```
# Crear el diagrama de dispersión
plt.figure(figsize=(8, 6))
plt.scatter(daily_returns['AAPL'], daily_returns['JPM'], alpha=0.5, color='blue')

# Configurar títulos y etiquetas
plt.title('Diagrama de Dispersión: Rendimientos Diarios de AAPL vs JPM')
plt.xlabel('Rendimientos Diarios de AAPL')
plt.ylabel('Rendimientos Diarios de JPM')
plt.grid(True)

# Mostrar el gráfico
plt.show()
```



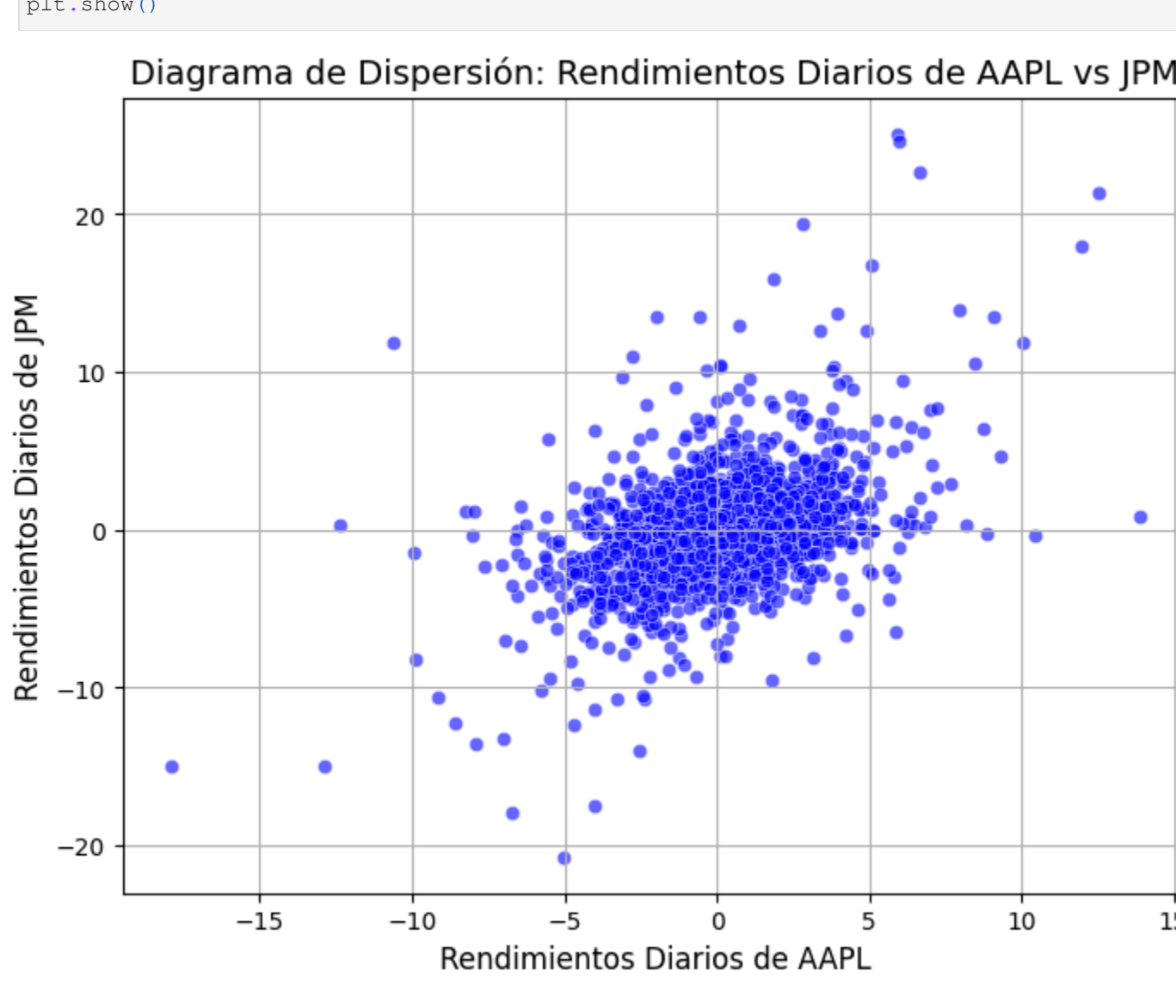
### Diagrama de dispersión similar entre los rendimientos diarios de Apple y JP Morgan con Seaborn

```
In [1]:
```

```
# Crear el diagrama de dispersión con Seaborn
plt.figure(figsize=(8, 6))
sns.scatterplot(
    x=daily_returns['AAPL'],
    y=daily_returns['JPM'],
    alpha=0.5,
    color='blue'
)

# Configurar títulos y etiquetas
plt.title('Diagrama de Dispersión: Rendimientos Diarios de AAPL vs JPM', fontsize=14)
plt.xlabel('Rendimientos Diarios de AAPL', fontsize=12)
plt.ylabel('Rendimientos Diarios de JPM', fontsize=12)
plt.grid(True)

# Mostrar el gráfico
plt.show()
```



### Asignar el 70% sobre AAPL y el resto entre las otras 3 acciones. Gráfico circular que muestre estas asignaciones utilizando explode

```
In [1]:
```

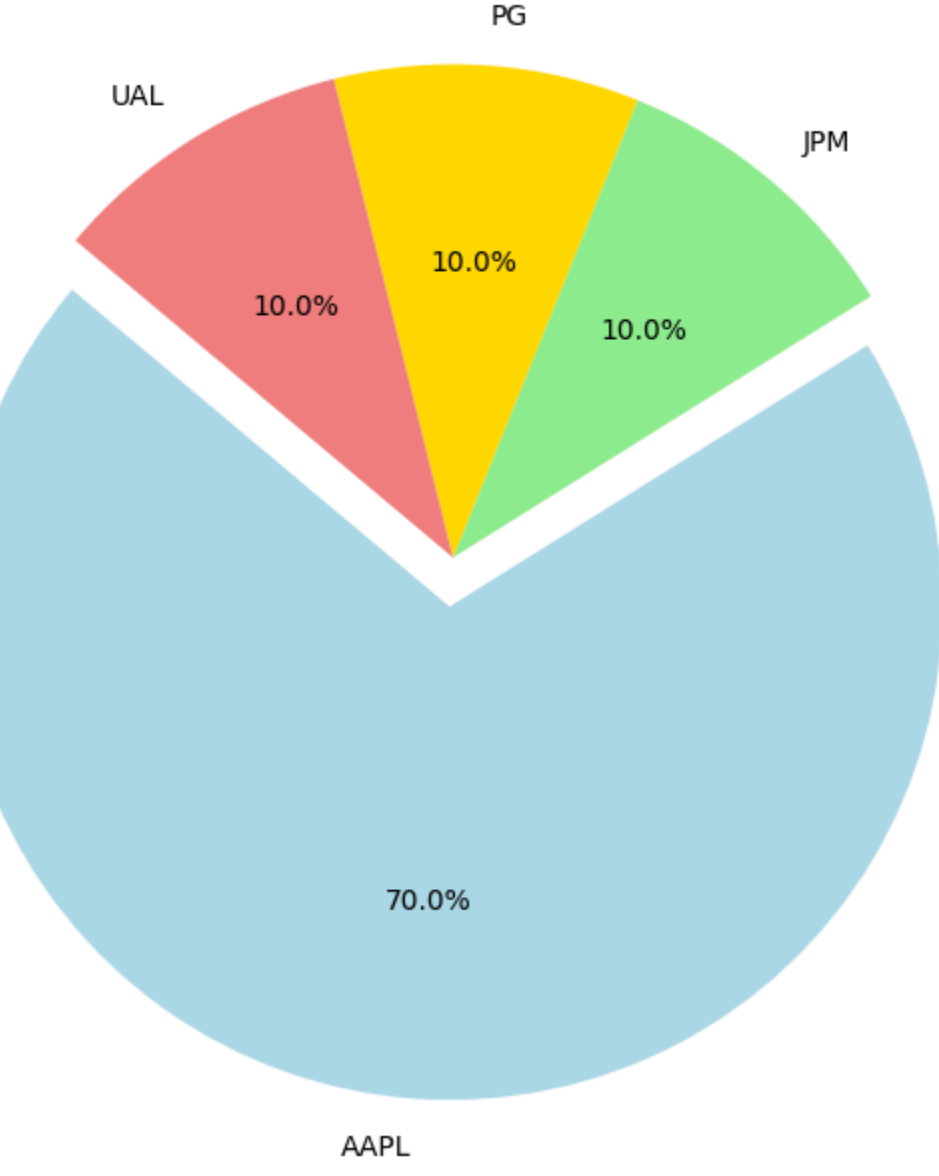
```
# Datos de la asignación de activos
labels = ['AAPL', 'JPM', 'PG', 'UAL']
sizes = [70, 10, 10, 10] # 70% para AAPL y 10% cada uno para el resto
explode = (0, 0, 0, 0) # Separación para AAPL

# Crear el gráfico circular
plt.figure(figsize=(8, 8))
plt.pie(
    sizes,
    labels=labels,
    explode=explode,
    autopct='%1.1f%%',
    startangle=40,
    colors=['lightblue', 'lightgreen', 'gold', 'lightcoral']
)

# Configurar título
plt.title('Asignación de Activos: AAPL y Otras Acciones', fontsize=14)

# Mostrar el gráfico
plt.show()
```

### Asignación de Activos: AAPL y Otras Acciones



Histograma para los retornos de United Airlines y P&G usando 40 divisiones con color rojo, donde se muestra la media y la desviación estándar para ambas acciones en la parte superior de la figura.

```
In [1]:
```

```
# Extraer los retornos de UAL y PG
ual_returns = daily_returns['UAL']
pg_returns = daily_returns['PG']

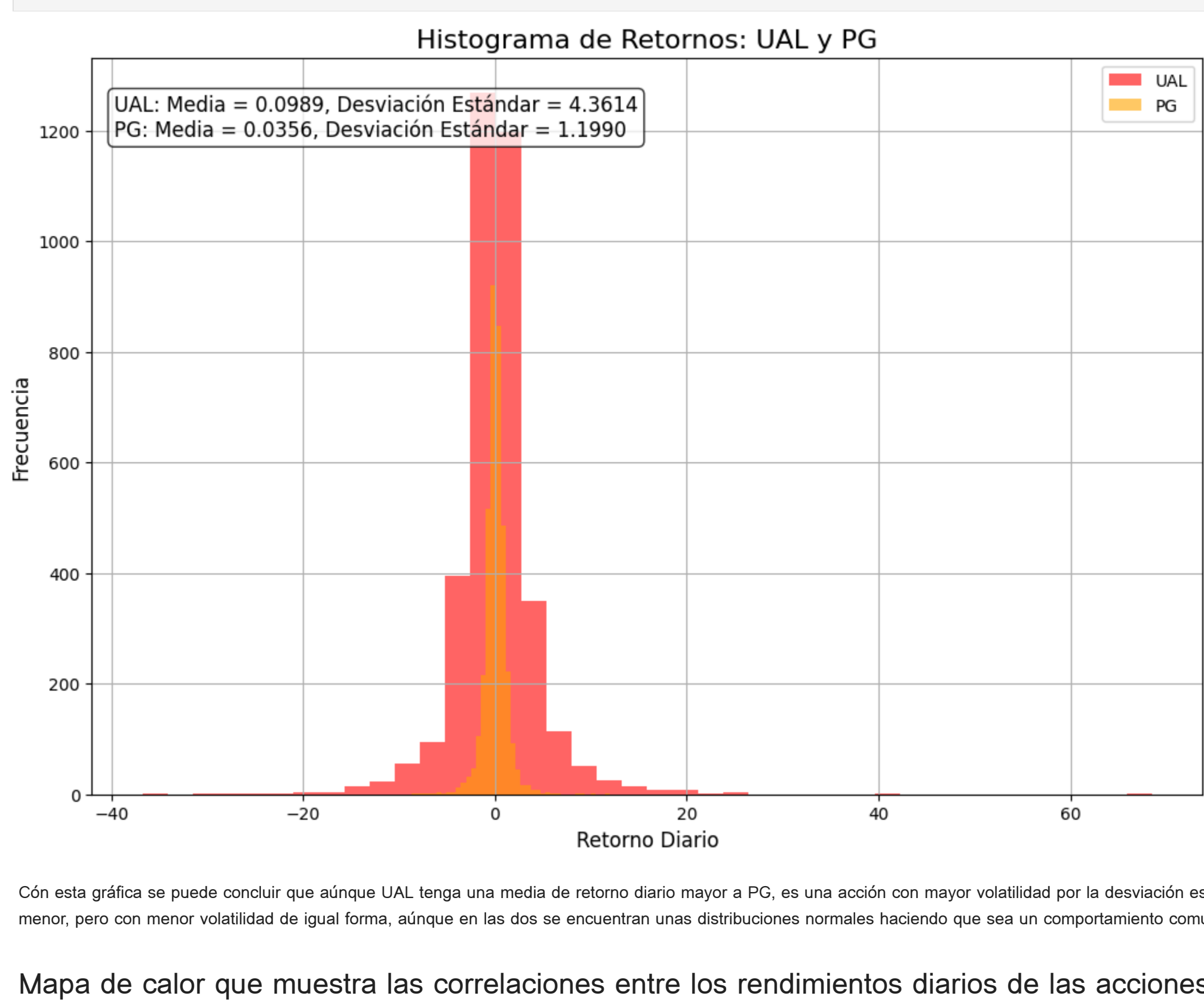
# Calcular media y desviación estándar
ual_mean, ual_std = ual_returns.mean(), ual_returns.std()
pg_mean, pg_std = pg_returns.mean(), pg_returns.std()

# Crear el histograma
plt.figure(figsize=(12, 8))
plt.hist(ual_returns, bins=40, alpha=0.6, color='red', label='UAL')
plt.hist(pg_returns, bins=40, alpha=0.6, color='orange', label='PG')

# Agregar información de estadísticas en la parte superior
plt.text(
    0.02, 0.55,
    f'UAL: Media = {ual_mean:.4f}, Desviación Estándar = {ual_std:.4f}\n'
    f'PG: Media = {pg_mean:.4f}, Desviación Estándar = {pg_std:.4f}',
    transform=plt.gca().transAxes,
    fontsize=12,
    verticalalignment='top',
    bboxdict=dict(boxstyle='round', facecolor='white', alpha=0.8)
)

# Configurar títulos y etiquetas
plt.title('Histograma de Retornos: UAL y PG', fontsize=16)
plt.xlabel('Retorno Diario', fontsize=12)
plt.ylabel('Frecuencia', fontsize=12)
plt.grid(True)

# Mostrar el gráfico
plt.show()
```



Como se puede ver, AAPL tiene una media de retorno diario mayor a PG, es una acción con mayor volatilidad por la desviación estándar, aparte de tener una mayor frecuencia en los datos observados. Mientras PG tiene unas ganancias o retorno diario menor, pero con menor volatilidad de igual forma, aunque en las dos se encuentran unas distribuciones normales haciendo que sea un comportamiento común en las dos acciones.

### Mapa de calor que muestra las correlaciones entre los rendimientos diarios de las acciones.

```
In [1]:
```

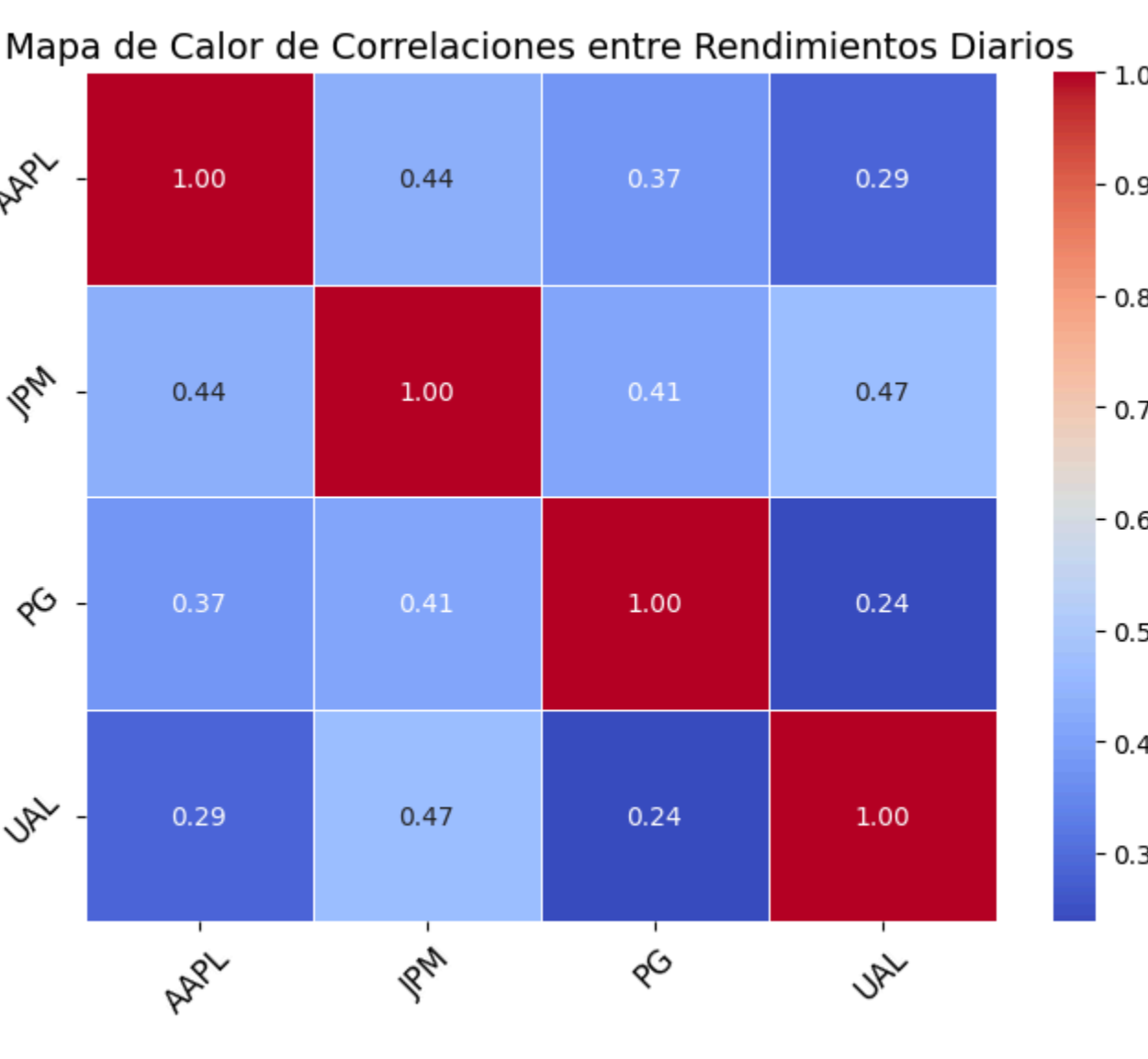
```
# Calcular la matriz de correlación
correlation_matrix = daily_returns[['AAPL', 'JPM', 'PG', 'UAL']].corr()

# Crear el mapa de calor
plt.figure(figsize=(8, 6))
sns.heatmap(
    correlation_matrix,
    annot=True,
    cmap='coolwarm',
    cbar=True,
    linewidth=0.5
)

# Configurar títulos y etiquetas
plt.title('Mapa de Calor de Correlaciones entre Rendimientos Diarios', fontsize=14)
plt.xticks(rotation=45, fontsize=12)
plt.yticks(rotation=45, fontsize=12)

# Mostrar el gráfico
plt.show()
```





Entre PG y UAL hay muy poca relación, ya que podemos observar que es la relación más pequeña entre todas las demás donde AAPL y JPM son las acciones que tienen más correlación, esta falta de correlación o de ser la menor tal vez se debe al ser la empresa que menos relación tiene con las otras 3, ya que una es de tecnología la otra es de banca la otra es de aerolínea, y esta es de productos de cuidado de piel, donde parece ser que es la que más alejada está en relación a otras y más aún con una aerolínea.