Milestone 4: Presenting Your Findings (Storytelling)

Custormer

This project aims to incorporate specific denotations for the Sportstats client, using the Olympic Games dataset - 120 years of data. The base will analyze the influence of event management based on gender, nationality, age, sport and other social groupings. The implementation of the analysis on the age distribution of the medalists will allow the identification of patterns related to the ideal age for high performance in different sports, in addition to observing the evolution of the age group throughout the editions of the Olympic Games. This analysis can provide insights into longevity in sports careers, the influence of physiological factors, and the impact of technological advances on the extent of athletes' peak performance. The main objective is to examine patterns and trends among medalists, seeking to understand how age influences sports performance in each competition category.

HYPOTHESES

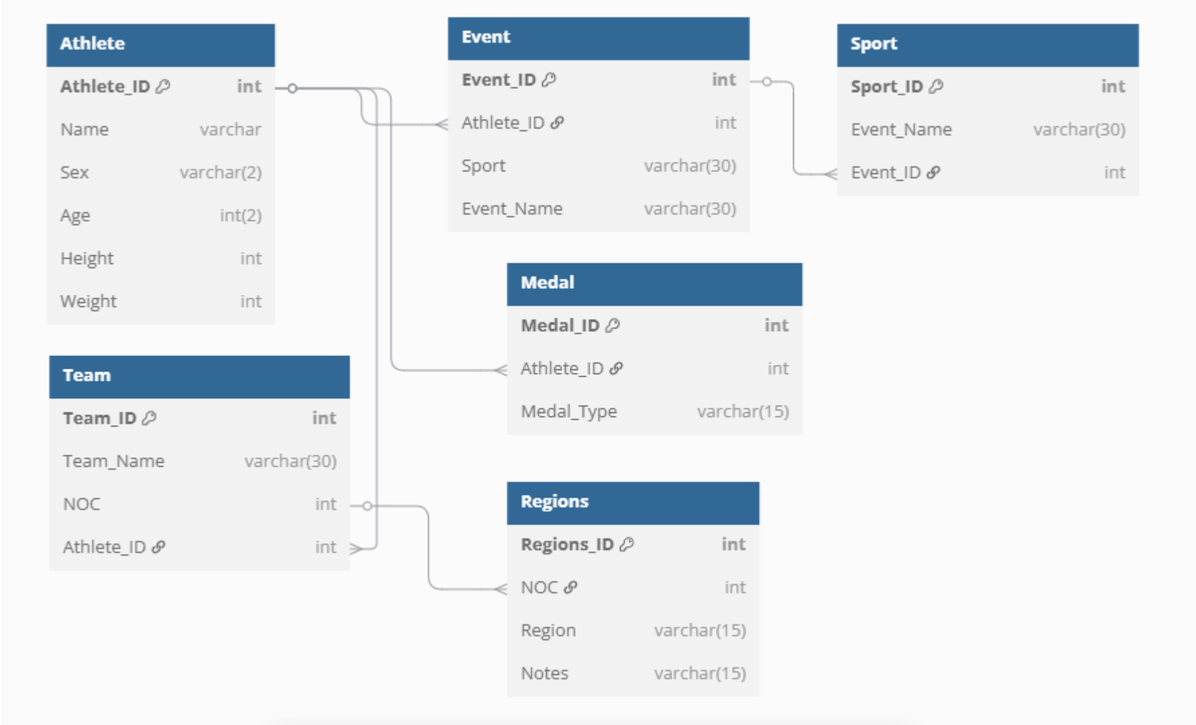
This project aims to develop a new analytical approach, exploring the trends of participation in the Olympic Games over the last 120 years. For this, we will use Python programming, ensuring an efficient and flexible structure for handling large volumes of data.

The integration of an advanced system will make it possible to identify historical patterns of participation, monitor the evolution of athletes throughout the editions of the Games and understand the main factors that impact sports performance over time. In addition, the use of advanced analytical tools will enable insights into technological changes, advances in training, and the influence of external variables on competitor performance.

APPROACH

This study seeks to understand the impact of age on the achievement of Olympic medals, revealing patterns that can help in the prediction of talent, the development of training strategies and the evaluation of changes in the age profile of athletes throughout Olympic history.

RELATIONSHIP DIAGRAM (DER)

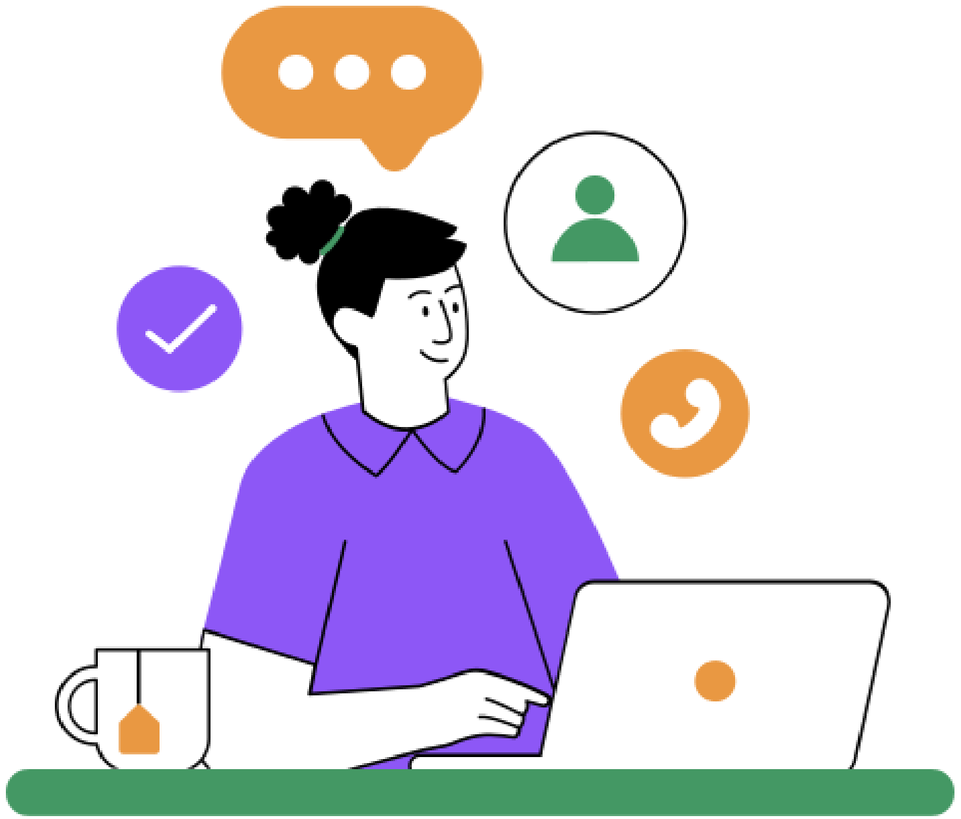


Relationship

In [10]:
```
import pandas as pd

# Carregar as bases de dados
athletes = pd.read_csv('athlete_events.csv')
regions = pd.read_csv('noc_regions.csv')

# Realizar a junção das tabelas usando 'NOC' como chave
dados_completos = athletes.merge(regions, on='NOC', how='left')

# Exibir as primeiras linhas do novo dataset
print(dados_completos.head())
```

```
   ID                      Name Sex   Age  Height  Weight            Team  \
0   1                 A Dijiang   M  24.0   180.0    80.0           China
1   2                  A Lamusi   M  23.0   170.0    60.0           China
2   3       Gunnar Nielsen Aaby   M  24.0     NaN     NaN         Denmark
3   4     Edgar Lindenau Aabye   M  34.0     NaN     NaN  Denmark/Sweden
4   5  Christine Jacoba Aaftink   F  21.0   185.0    82.0     Netherlands

   NOC        Games  Year  Season       City          Sport  \
0  CHN  1992 Summer  1992  Summer  Barcelona     Basketball
1  CHN  2012 Summer  2012  Summer     London           Judo
2  DEN  1920 Summer  1920  Summer  Antwerpen       Football
3  DEN  1900 Summer  1900  Summer      Paris     Tug-Of-War
4  NED  1988 Winter  1988  Winter    Calgary  Speed Skating

                           Event Medal       region notes
0         Basketball Men's Basketball   NaN        China   NaN
1     Judo Men's Extra-Lightweight   NaN        China   NaN
2           Football Men's Football   NaN      Denmark   NaN
3       Tug-Of-War Men's Tug-Of-War  Gold      Denmark   NaN
4  Speed Skating Women's 500 metres   NaN  Netherlands   NaN
```

Reading Database: athletes

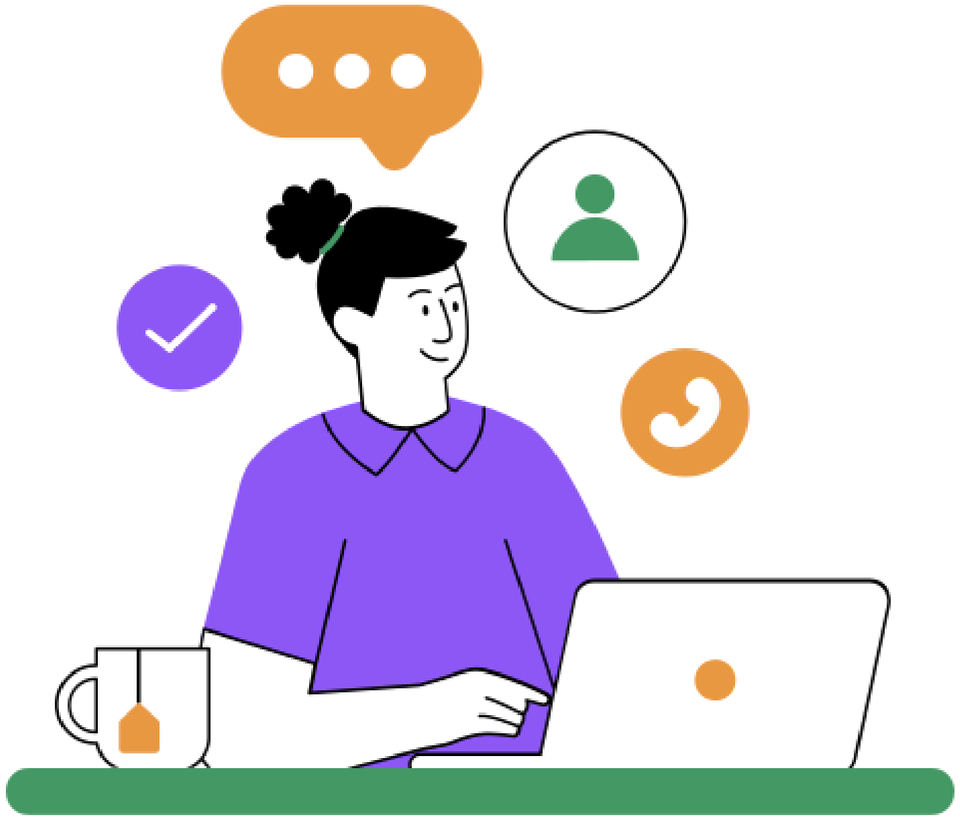


In [13]:
```
import pandas as pd

# Lendo o arquivo CSV
caminho_arquivo = 'athlete_events.csv'
dados = pd.read_csv(caminho_arquivo)

# Preenchendo valores ausentes e convertendo 'Age' para int
dados['Age'] = dados['Age'].fillna(0).astype(int)


# Exibindo os dados atualizados
dados
```

Out[13]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
271111	135569	Andrzej ya	M	29	179.0	89.0	Poland-1	POL	1976	1976	Winter	Innsbruck	Luge	Luge Mixed (Men)'s	NaN

Reading Database: regions

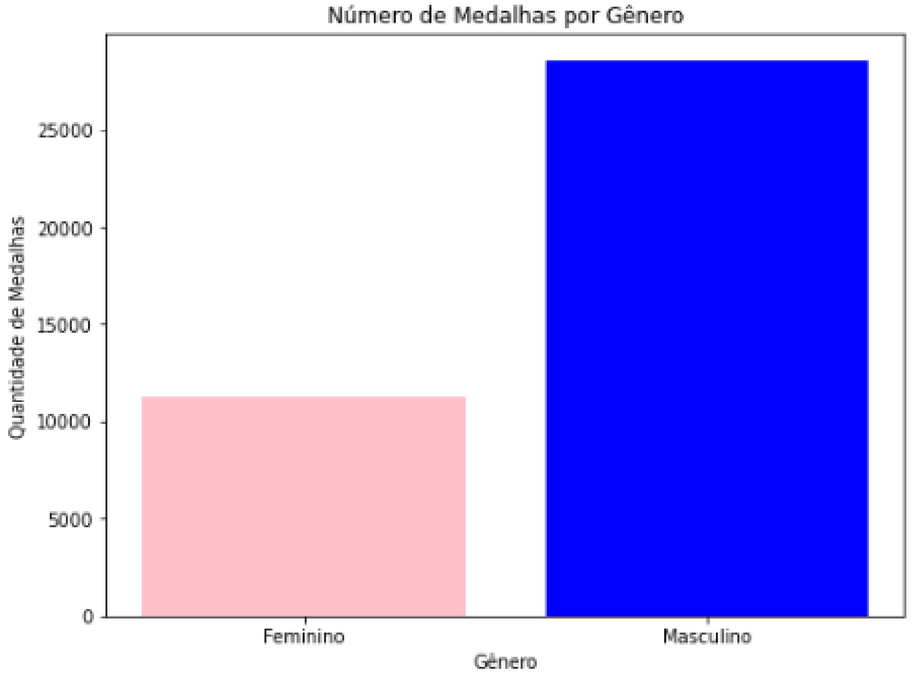


In [4]:
```
import pandas as pd

# Lendo o arquivo CSV
caminho_arquivo = 'noc_regions.csv'
dados = pd.read_csv(caminho_arquivo)

# Ajustando para mostrar todas as colunas
pd.set_option('display.max_columns', None)

# Exibindo o DataFrame completo
print(dados)
```

```
     NOC       region                notes
0    AFG  Afghanistan                  NaN
1    AHO      Curacao  Netherlands Antilles
2    ALB      Albania                  NaN
3    ALG      Algeria                  NaN
4    AND      Andorra                  NaN
..   ...          ...                  ...
225  YEM        Yemen                  NaN
226  YMD        Yemen           South Yemen
227  YUG       Serbia            Yugoslavia
228  ZAM       Zambia                  NaN
229  ZIM     Zimbabwe                  NaN

[230 rows x 3 columns]
```

Exercises 1 – Distributions of medasl by gender

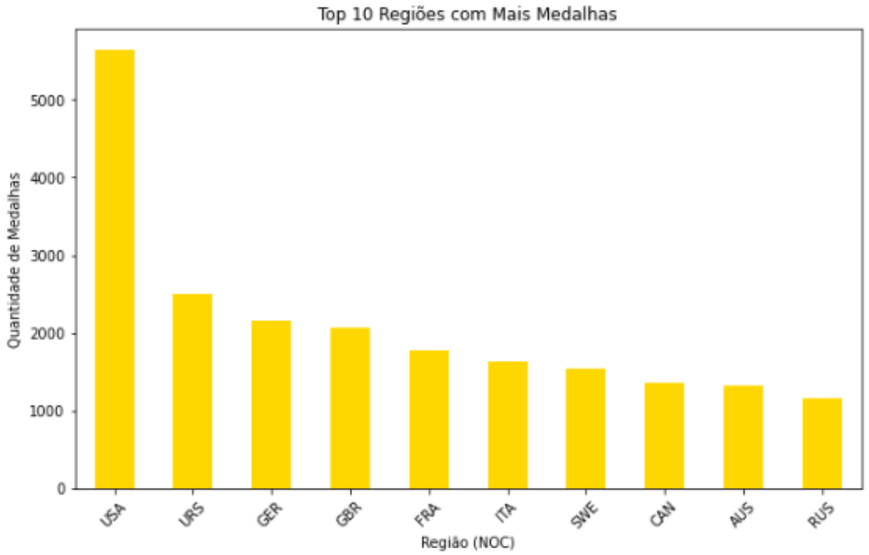
In [22]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Lendo o arquivo CSV
caminho_arquivo = 'athlete_events.csv'
dados = pd.read_csv(caminho_arquivo)

# Filtrando dados e contando medalhas por gênero
medalhas_femininas = dados[(dados['Sex'] == 'F') & (dados['Medal'].notna())]['Medal'].count()
medalhas_masculinas = dados[(dados['Sex'] == 'M') & (dados['Medal'].notna())]['Medal'].count()

# Criando DataFrame para o gráfico
dados_medalhas = pd.DataFrame({
    'Gênero': ['Feminino', 'Masculino'],
    'Número de Medalhas': [medalhas_femininas, medalhas_masculinas]
})

# Criando o gráfico de barras
plt.figure(figsize=(8, 6))
plt.bar(dados_medalhas['Gênero'], dados_medalhas['Número de Medalhas'], color=['pink', 'blue'])
plt.title('Número de Medalhas por Gênero')
plt.xlabel('Gênero')
plt.ylabel('Quantidade de Medalhas')
plt.show()
```



Exercises 1 – TOP 10 regions that won the most medals

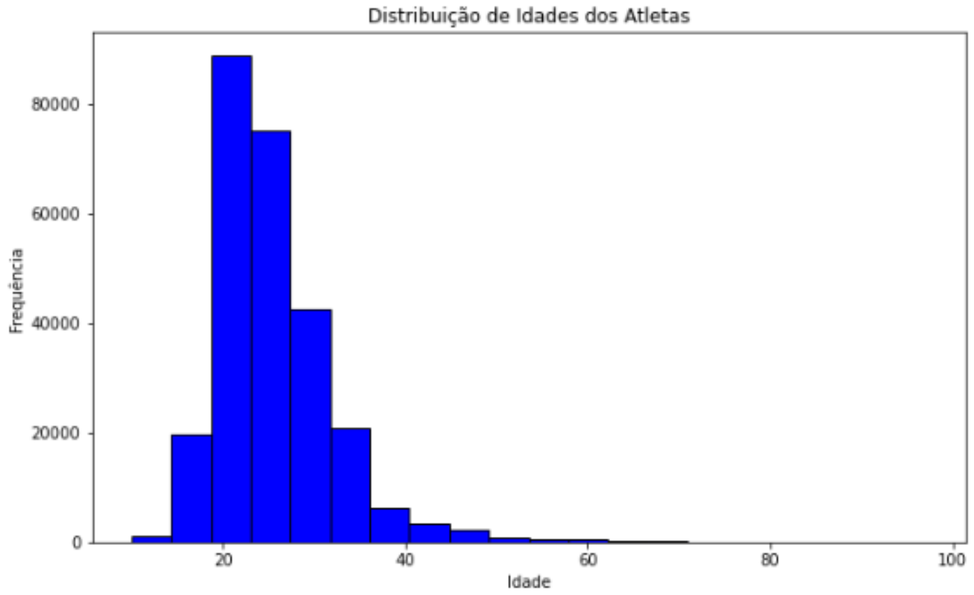
In [24]:
```python
import pandas as pd
import matplotlib.pyplot as plt

# Lendo o arquivo CSV
caminho_arquivo = 'athlete_events.csv'
dados = pd.read_csv(caminho_arquivo)

# Filtrando apenas registros com medalhas
medalhas_por_regiao = dados[dados['Medal'].notna()]['NOC'].value_counts()

# Obtendo as 10 regiões com mais medalhas
top_10_medalhas = medalhas_por_regiao.head(10)

# Exibindo os dados das 10 principais regiões
print(top_10_medalhas)

# Criando o gráfico
plt.figure(figsize=(10, 6))
top_10_medalhas.plot(kind='bar', color='gold')
plt.title('Top 10 Regiões com Mais Medalhas')
plt.xlabel('Região (NOC)')
plt.ylabel('Quantidade de Medalhas')
plt.xticks(rotation=45)
plt.show()
```

```
USA    5637
URS    2503
GER    2165
GBR    2068
```



Exercises 1 – Age Distribution

In [15]:
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Lendo o arquivo CSV
caminho_arquivo = 'athlete_events.csv'
dados = pd.read_csv(caminho_arquivo)

# Removendo valores NaN da coluna 'Age' para evitar erros
dados['Age'] = dados['Age'].dropna()

# Usando Matplotlib
plt.figure(figsize=(10, 6))
plt.hist(dados['Age'], bins=20, color='blue', edgecolor='black')
plt.title('Distribuição de Idades dos Atletas')
plt.xlabel('Idade')
plt.ylabel('Frequência')
plt.show()

# Usando Seaborn (gráfico mais estilizado)
plt.figure(figsize=(10, 6))
sns.histplot(dados['Age'], bins=20, kde=True, color='green')
plt.title('Distribuição de Idades dos Atletas (Seaborn)')
plt.xlabel('Idade')
plt.ylabel('Frequência')
plt.show()
```



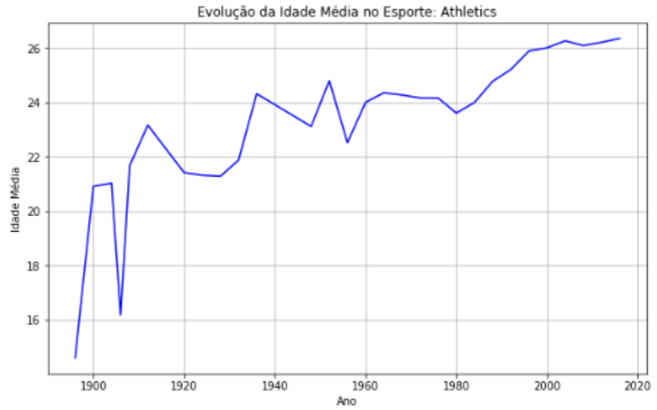
Exercises 2

Exploring the age distribution of Olympic athletes over the years is essential to identify patterns and trends that directly influence sports performance. This analysis will allow us to understand how age impacts the achievement of medals and the longevity of sports careers, revealing strategic insights into which modalities favor younger athletes and which require greater experience and physical maturity.

In [13]:
```
import pandas as pd

# Carregar os dados
dados = pd.read_csv('athlete_events.csv')

# Remover valores ausentes e converter a coluna 'Age' para inteiro
dados = dados.dropna(subset=['Age'])
dados['Age'] = dados['Age'].astype(int)

# Criar o totalizador por idade
totalizador_idade = dados['Age'].value_counts().sort_index()

# Exibir o total de atletas por idade
print(totalizador_idade)
```

```
10      1
11     13
12     39
13    187
14    837
     ...
81      2
84      1
88      3
96      1
97      1
Name: Age, Length: 74, dtype: int64
```

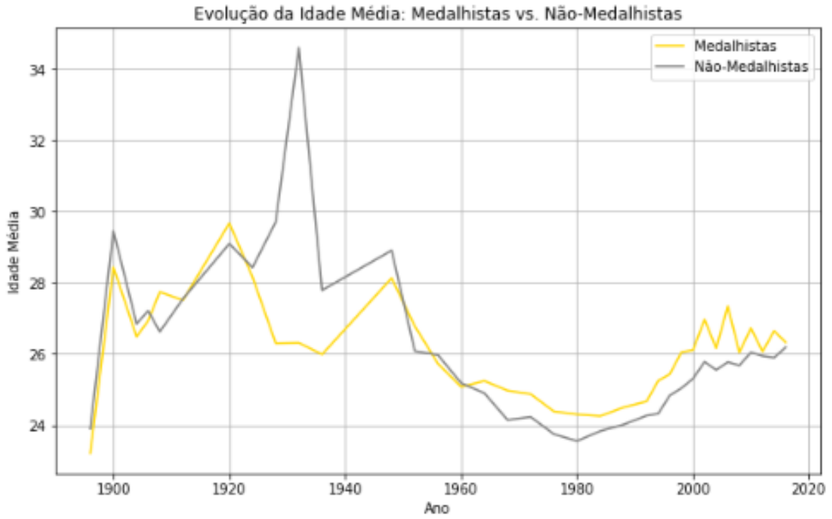
In [14]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Carregar o arquivo CSV
caminho_arquivo = 'athlete_events.csv'
dados = pd.read_csv(caminho_arquivo)

# Preencher valores ausentes e garantir que 'Age' e 'Year' sejam numéricos
dados['Age'] = dados['Age'].fillna(0).astype(int)
dados['Year'] = pd.to_numeric(dados['Year'], errors='coerce')

# 1. Evolução das idades por esporte
# Calculando a média de idade por ano para um esporte específico, exemplo: 'Athletics'
esporte_especifico = 'Athletics'
idade_media_esporte = dados[dados['Sport'] == esporte_especifico].groupby('Year')['Age'].mean()

plt.figure(figsize=(10, 6))
idade_media_esporte.plot(kind='line', color='blue')
plt.title(f'Evolução da Idade Média no Esporte: {esporte_especifico}')
plt.xlabel('Ano')
plt.ylabel('Idade Média')
plt.grid()
plt.show()
```



Exercises 2

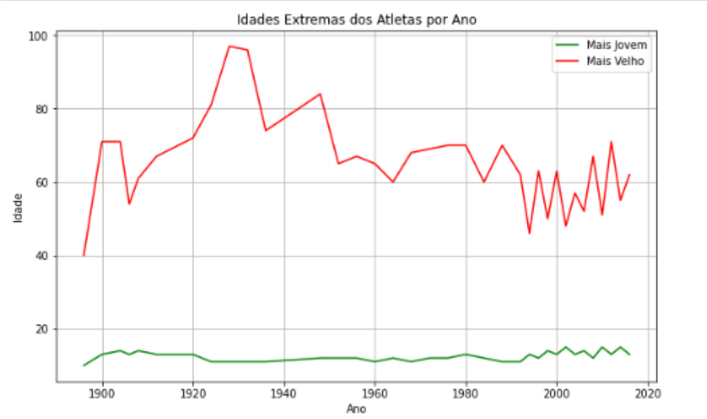
In [15]:

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

# Carregar o arquivo CSV
caminho_arquivo = 'athlete_events.csv'
dados = pd.read_csv(caminho_arquivo)

# 2. Evolução das idades: Medalhistas vs. Não-Medalhistas
# Criar uma coluna para identificar medalhistas
dados['Medalist'] = dados['Medal'].notna()

# Calcular a idade média por ano para medalhistas e não-medalhistas
idade_media_medalhistas = dados[dados['Medalist']].groupby('Year')['Age'].mean()
idade_media_nao_medalhistas = dados[~dados['Medalist']].groupby('Year')['Age'].mean()

plt.figure(figsize=(10, 6))
idade_media_medalhistas.plot(kind='line', label='Medalhistas', color='gold')
idade_media_nao_medalhistas.plot(kind='line', label='Não-Medalhistas', color='gray')
plt.title('Evolução da Idade Média: Medalhistas vs. Não-Medalhistas')
plt.xlabel('Ano')
plt.ylabel('Idade Média')
plt.legend()
plt.grid()
plt.show()
```



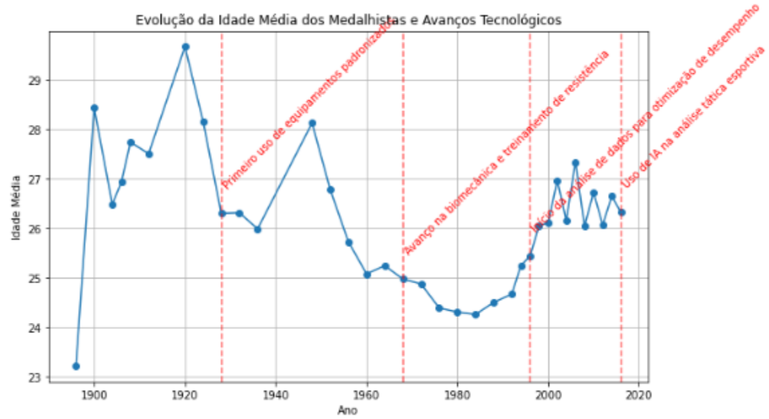
In [16]:

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

# Carregar o arquivo CSV
caminho_arquivo = 'athlete_events.csv'
dados = pd.read_csv(caminho_arquivo)

# 3. Atletas mais jovens e mais velhos em cada edição dos Jogos
# Encontrar os atletas mais jovens e mais velhos por ano
idades_extremas = dados.groupby('Year')['Age'].agg(['min', 'max'])

plt.figure(figsize=(10, 6))
idades_extremas['min'].plot(kind='line', label='Mais Jovem', color='green')
idades_extremas['max'].plot(kind='line', label='Mais Velho', color='red')
plt.title('Idades Extremas dos Atletas por Ano')
plt.xlabel('Ano')
plt.ylabel('Idade')
plt.legend()
plt.grid()
plt.show()
```



Exercises 2

```
In [18]: import pandas as pd
         import matplotlib.pyplot as plt

         # Carregar os dados dos atletas olímpicos
         dados = pd.read_csv('athlete_events.csv')

         # Remover valores ausentes e converter a idade para inteiro
         dados = dados.dropna(subset=['Age'])
         dados['Age'] = dados['Age'].astype(int)

         # Filtrar apenas medalhistas
         medalhistas = dados[dados['Medal'].notna()]

         # Calcular idade média dos medalhistas ao longo dos anos
         idade_media_por_ano = medalhistas.groupby('Year')['Age'].mean()

         # Criar gráfico para visualizar a evolução da idade média
         plt.figure(figsize=(10, 6))
         plt.plot(idade_media_por_ano.index, idade_media_por_ano.values, marker='o', linestyle='-')
         plt.title('Evolução da Idade Média dos Medalhistas e Avanços Tecnológicos')
         plt.xlabel('Ano')
         plt.ylabel('Idade Média')
         plt.grid()

         # Destacar marcos tecnológicos que podem ter impactado a longevidade esportiva
         marcos_tecnologicos = {
             1928: "Primeiro uso de equipamentos padronizados",
             1968: "Avanço na biomecânica e treinamento de resistência",
             1996: "Início da análise de dados para otimização de desempenho",
             2016: "Uso de IA na análise tática esportiva"
         }

         for ano, evento in marcos_tecnologicos.items():
             if ano in idade_media_por_ano.index:
                 plt.axvline(x=ano, color='red', linestyle='--', alpha=0.6)
                 plt.text(ano, idade_media_por_ano[ano] + 0.5, evento, rotation=45, fontsize=10, color='red')

         plt.show()
```



In this item, a new hypothesis was created by adding information from the new technologies. The average age of medalists has gradually increased over the years, which suggests that sporting longevity has been extended. This effect may be related to improvements in training methods, advancement in sports medicine, and muscle recovery strategies, allowing athletes to remain competitive for longer.

The inclusion of technological events in the graph highlights moments when significant advances may have influenced athletes' performance:

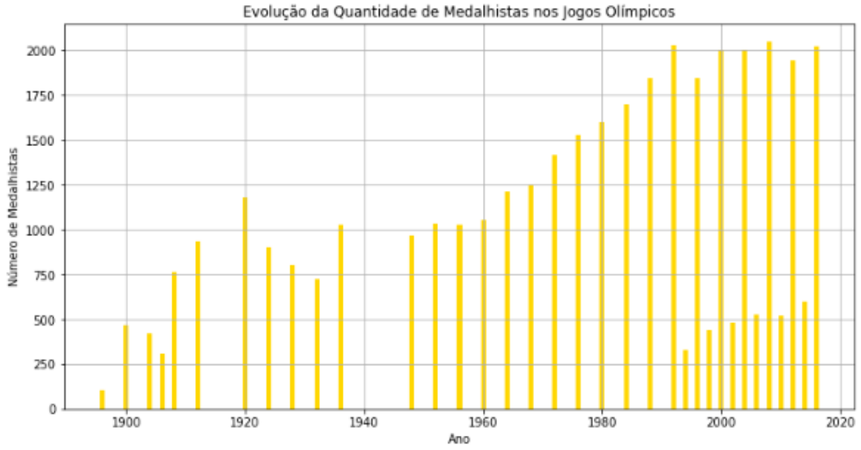
- 1928 – Standardization of → Equipment Enabled greater equality in competitions and better adaptation of athletes.
- 1968 – Advancement in Biomechanics → Development of techniques for optimizing movements and physical endurance.
- 1996 – Data Analysis for Training → Allowed personalized adjustments in the physical preparation of athletes, maximizing their performance.
- 2016 – Artificial Intelligence in Sports → Application of data technology to predict performance patterns and improve competitive strategies.

The impact of these advances can be observed in the maintenance of a higher average age of the medalists in the following periods.

.

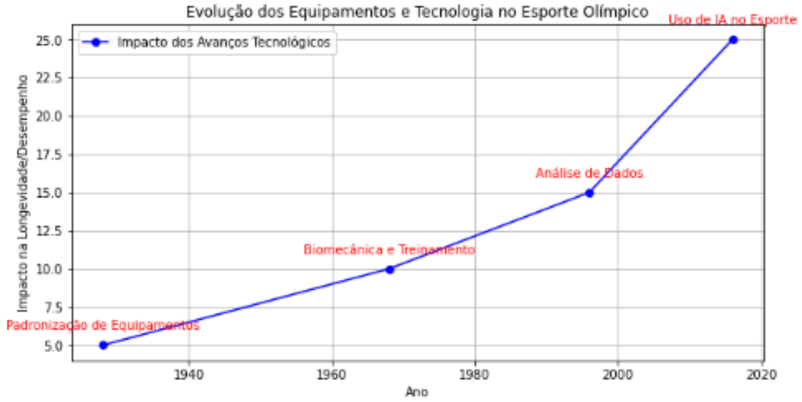
Exercises 2

```
# Criar gráfico para visualizar a evolução da idade média
plt.figure(figsize=(12, 6))
plt.plot(idade_media_por_ano.index, idade_media_por_ano.values, marker='o', linestyle='-', label='Idade Média')
plt.ylabel('Idade Média dos Medalhistas')
plt.xlabel('Ano')
plt.title('Evolução da Idade Média dos Medalhistas e Avanços Tecnológicos')
plt.grid()
plt.legend()

# Destacar marcos tecnológicos que podem ter impactado a longevidade esportiva
marcos_tecnologicos = {
    1928: "Padronização de equipamentos",
    1968: "Biomecânica e treinamento de resistência",
    1996: "Análise de dados no treinamento",
    2016: "Uso de IA na análise tática esportiva"
}

for ano, evento in marcos_tecnologicos.items():
    if ano in idade_media_por_ano.index:
        plt.axvline(x=ano, color='red', linestyle='--', alpha=0.6)
        plt.text(ano, idade_media_por_ano[ano] + 0.5, evento, rotation=45, fontsize=10, color='red')

plt.show()

# Criar gráfico para visualizar a contagem de medalhistas ao longo dos anos
plt.figure(figsize=(12, 6))
plt.bar(contagem_medalhistas_por_ano.index, contagem_medalhistas_por_ano.values, color='gold')
plt.ylabel('Número de Medalhistas')
plt.xlabel('Ano')
plt.title('Evolução da Quantidade de Medalhistas nos Jogos Olímpicos')
plt.grid()
plt.show()
```



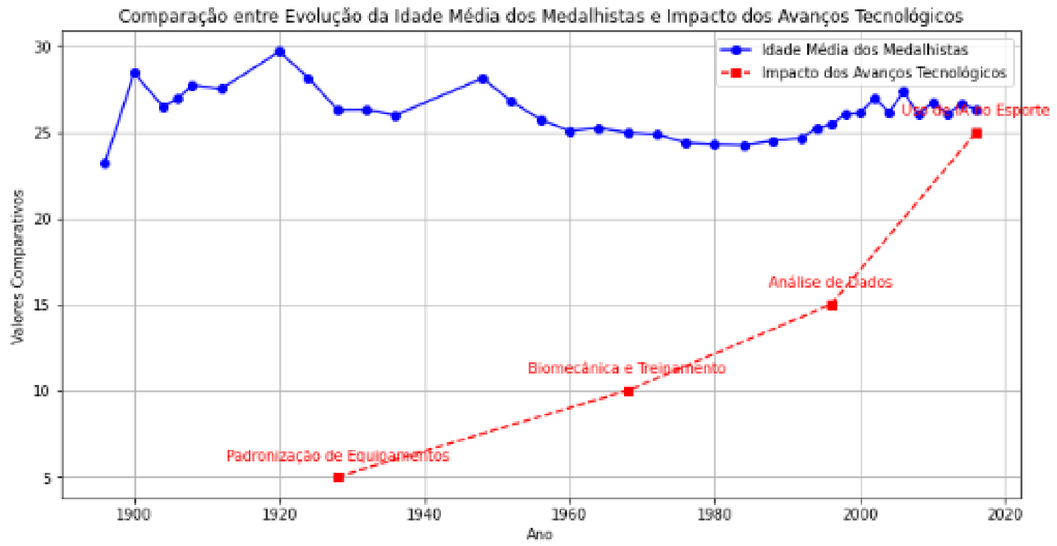
In [21]:

```
import matplotlib.pyplot as plt

# Definição de marcos tecnológicos e seus impactos na longevidade esportiva
anos = [1928, 1968, 1996, 2016]
impacto = [5, 10, 15, 25]  # Representa o aumento no rendimento/longevidade devido aos avanços tecnológicos
eventos = ["Padronização de Equipamentos", "Biomecânica e Treinamento", "Análise de Dados", "Uso de IA no Esporte"]

# Criando o gráfico
plt.figure(figsize=(10, 5))
plt.plot(anos, impacto, marker='o', linestyle='-', color='blue', label="Impacto dos Avanços Tecnológicos")

# Adicionando os eventos no gráfico
for i in range(len(anos)):
    plt.text(anos[i], impacto[i] + 1, eventos[i], fontsize=10, ha='center', color='red')

plt.xlabel("Ano")
plt.ylabel("Impacto na Longevidade/Desempenho")
plt.title("Evolução dos Equipamentos e Tecnologia no Esporte Olímpico")
plt.grid()
plt.legend()
plt.show()
```



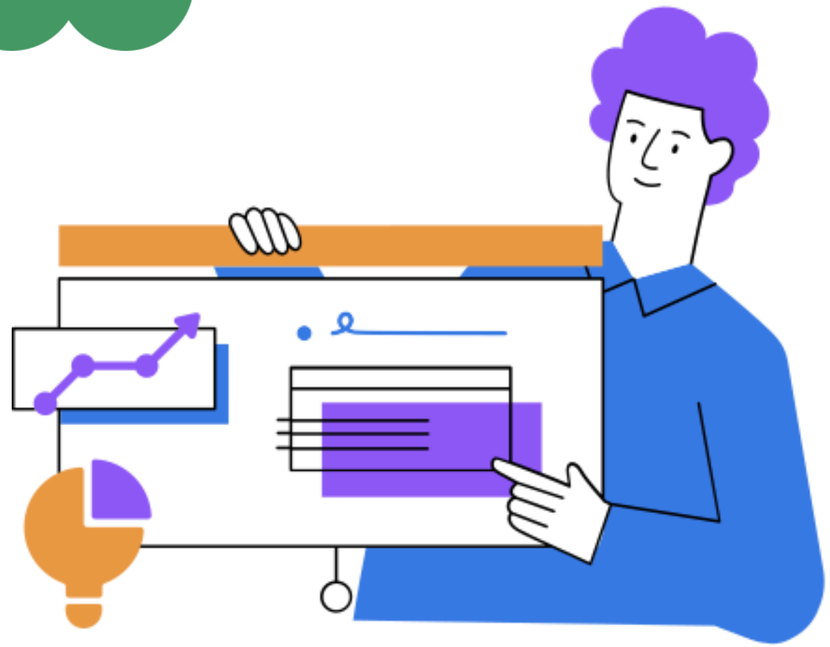
Exercises 2

Correlation between age of medalists and technological advances: If we observe that the average age of medalists increases after certain technological advances, this may indicate that new methods and equipment help to prolong the career of athletes.
Technological milestones impacting performance: The red curve highlights key moments in the evolution of sports, allowing you to assess whether there are peaks or significant changes in performance after these advances.
Periods of greater transformation: If there are jumps in the average age of medalists close to technological advances, this may suggest that more modern training methods have allowed older athletes to maintain a competitive performance.




```
In [22]: import pandas as pd
         import matplotlib.pyplot as plt

         # Carregar os dados dos atletas olímpicos
         dados = pd.read_csv('athlete_events.csv')

         # Remover valores ausentes e converter a idade para inteiro
         dados = dados.dropna(subset=['Age'])
         dados['Age'] = dados['Age'].astype(int)

         # Filtrar apenas medalhistas
         medalhistas = dados[dados['Medal'].notna()]

         # Calcular idade média dos medalhistas ao longo dos anos
         idade_media_por_ano = medalhistas.groupby('Year')['Age'].mean()

         # Definição de marcos tecnológicos e seus impactos na longevidade esportiva
         anos_tecnologia = [1928, 1968, 1996, 2016]
         impacto_tecnologico = [5, 10, 15, 25]  # Representa o aumento no rendimento devido aos avanços tecnológicos
         eventos_tecnologicos = ["Padronização de Equipamentos", "Biomecânica e Treinamento", "Análise de Dados", "Uso de IA no Esporte"]

         # Criando o gráfico comparativo
         plt.figure(figsize=(12, 6))

         # Plotando evolução da idade média dos medalhistas
         plt.plot(idade_media_por_ano.index, idade_media_por_ano.values, marker='o', linestyle='-', label="Idade Média dos Medalhistas",

         # Plotando o impacto da tecnologia ao longo dos anos
         plt.plot(anos_tecnologia, impacto_tecnologico, marker='s', linestyle='--', label="Impacto dos Avanços Tecnológicos", color='red'

         # Adicionando eventos no gráfico
         for i in range(len(anos_tecnologia)):
             plt.text(anos_tecnologia[i], impacto_tecnologico[i] + 1, eventos_tecnologicos[i], fontsize=10, ha='center', color='red')

         plt.xlabel("Ano")
         plt.ylabel("Valores Comparativos")
         plt.title("Comparação entre Evolução da Idade Média dos Medalhistas e Impacto dos Avanços Tecnológicos")
         plt.legend()
         plt.grid()
         plt.show()
```

Conclusions



Evolution of the Age of Medalists and Impact of Technology on the Olympic Games

The average age of medalists has increased in recent decades.

Technological advancements and modern training methods allow for longer careers.

Endurance modalities favor older athletes, while explosive sports favor younger ones.

Impact of Technological Advances

1928 → Standardization of equipment improves competitiveness.

1968 → Biomechanics and resistance training increase peak performance.

1996 → Data analysis optimizes fitness and recovery.

2016 → AI applied to sports strategies improves efficiency and longevity.

Hypotheses for the Future

Sports longevity will continue to grow with new medical and technological advancements.

Countries that invest in science and technology in sports will have a competitive advantage.

Technical and endurance sports will continue to allow older athletes at a high level.

Technology plays a fundamental role in the evolution of athletes' performance, allowing for longer careers and changing participation patterns in the Olympic Games.

Thank You