

# Olympics Dataset

Equipe: Liga Olímpica

Integrantes:

RA 213374 - Áureo Henrique e Silva Marques

RA 176566 - José Alexandre dos Santos Barros

RA 220407 - Lindon Jonathan Sanley dos Santos Pereira Monroe

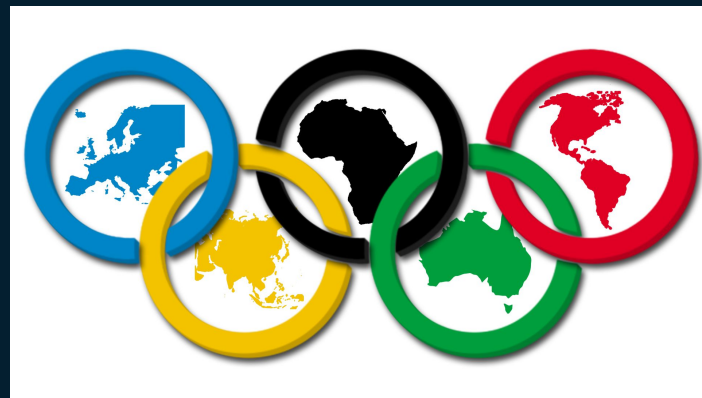
# Tema

- Histórico dos Jogos Olímpicos nos últimos anos.
- Os Jogos Olímpicos, ou Olimpíadas, são o maior evento esportivo do mundo e, de 4 em 4 anos, reúnem milhares de atletas de vários países. Embora suas origens sejam da Grécia Antiga, as primeiras Olimpíadas ocorreram oficialmente em 1896, organizadas pelo Comitê Olímpico Internacional (COI) e, portanto, desse ano até hoje, tratam-se de mais de 30 edições dos jogos olímpicos.

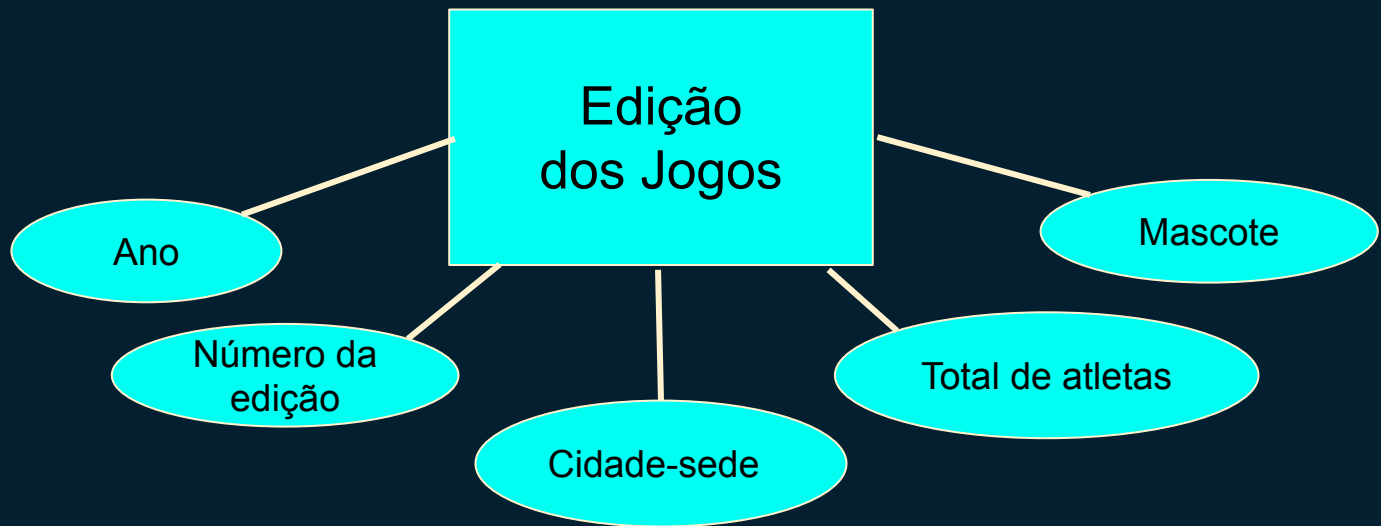


# Tema

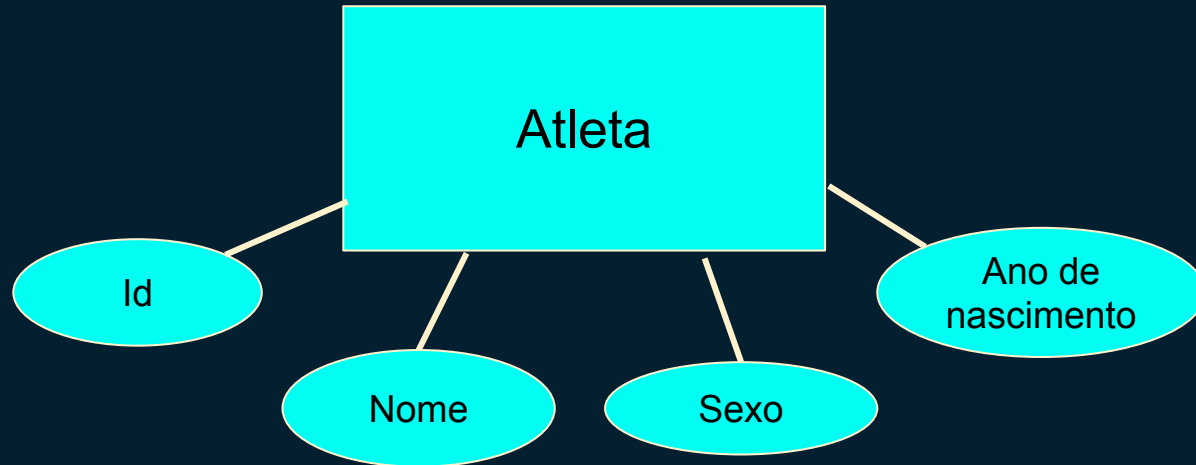
- Grande quantidade de informações sobre: atletas, países participantes, cidades-sede, medalhas, esportes, vencedores de cada modalidade, entre outros.
- É difícil encontrar bancos de dados atuais que possuam essas informações de forma centralizada e bem organizada.
- **Objetivo:** construir um dataset sobre os Jogos Olímpicos dos últimos anos que seja organizado e bem estruturado, permitindo diversos tipos de análises sobre o tema.



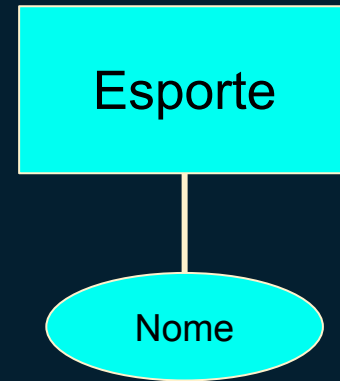
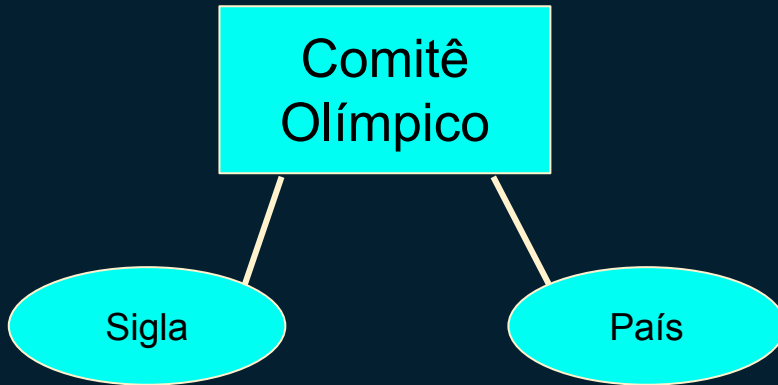
# Modelo conceitual (entidade Edição dos Jogos)



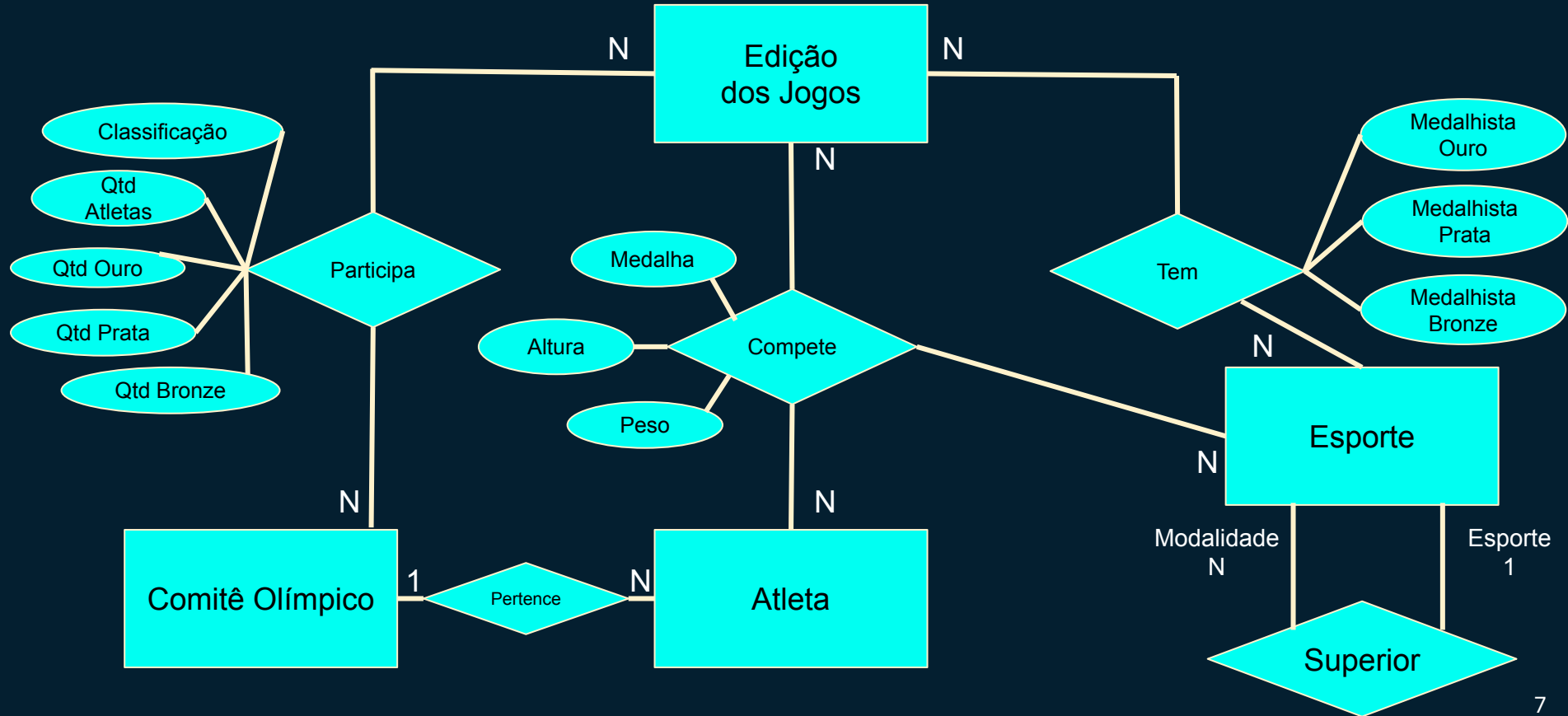
# Modelo conceitual (entidade Atleta)



# Modelo conceitual (entidade Comitê Olímpico e entidade Esporte)



# Modelo conceitual (geral)



# Fontes de dados

As fontes de dados que já pretendemos utilizar no dataset são as seguintes:

- “120 years of Olympic history: athletes and results”

<https://www.kaggle.com/heesoo37/120-years-of-olympic-history-athletes-and-results/discussion/69221>

Dataset histórico, uma tabela com dados das olimpíadas de 1896 a 2016. Foi criado a partir de dados do site [www.sports-reference.com](http://www.sports-reference.com).

- 2021 Olympics in Tokyo


<https://www.kaggle.com/arjunprasadsarkhel/2021-olympics-in-tokyo>

Dataset que consiste em uma tabela com dados específicos das olimpíadas de Tóquio em 2021.

- <https://olympics.com>


Site oficial do Comitê Olímpico Internacional (IOC) contendo uma base extensa de dados, notícias e informações sobre os Jogos Olímpicos e seus envolvidos, em geral.


Name	Sex	Age	Height	Weight	Team	NOC					
134732 unique values	M	73%	23	8%	NA	22%	NA	23%	United States	7%	USA
	F	27%	24	8%	180	5%	70	4%	France	4%	FRA
			Other (227521)	84%	Other (198453)	73%	Other (198816)	73%	Other (241281)	89%	Other (239)
A. Dijiang	M		24		188		88		China		CHN
A. Lamusi	M		23		178		68		China		CHN
Gunnar Nielsen Aaby	M		24		NA		NA		Denmark		DEN


Olympic GamesAthletesSportsNewsOlympic Channel


Athletes


Search for an athlete


  
SIMONE BILES  
USA, ARTISTIC G...


  
MICHAEL PHELPS  
USA, SWIMMING

  
PUSARLA VENKA...  
IND, BADMINTON

  
USAIN BOLT  
JAM, ATHLETICS

  
NAOMI OSAKA  
JPN, TENNIS

  
KATIE LEDECKY  
USA, SWIMMING

  
YUI HIRONAKA  
JPN, F...



# Modelos lógicos

- **Modelo relacional:** O modelo lógico relacional servirá para tratamento e melhor organização dos dados obtidos nas tabelas das fontes. Além disso, diversas análises estatísticas e comparativas são possíveis utilizando dados estruturados.
- **Modelo de documentos:** O modelo de documentos ajudará a encapsular melhor as informações sobre cada edição, através da hierarquia de elementos. Com ele, será possível obter informações mais diretas sobre um atleta ou um esporte em uma determinada olimpíada.

# Modelo lógico: Relacional

EdicaoDosJogos (Ano, NumeroDaEdicao, CidadeSede, TotalDeAtletas, Mascote)

Atleta (Id, Nome, AnoDeNascimento, Sexo)

ComiteOlimpico(Sigla, País)

EsporteModalidade(Id, Nome, EsportePai)

# Modelo lógico: Relacional

ParticipacaoComites(IdComite, AnoEdicao, QtdAtletas, QtdOuro , QtdPrata , QtdBronze, Classificacao)

- Chaves estrangeiras: IdComite -> Comiteloimpico(Sigla), AnoEdicao -> EdicaoDosJogos(Ano)

ParticipacaoAtletas(IdAtleta, AnoEdicao, IdModalidade, Altura, Peso, Medalha)

- Chaves estrangeiras: IdAtleta -> Atleta(Id), AnoEdicao -> EdicaoDosJogos(Ano),  
IdModalidade -> EsporteModalidade(Id)

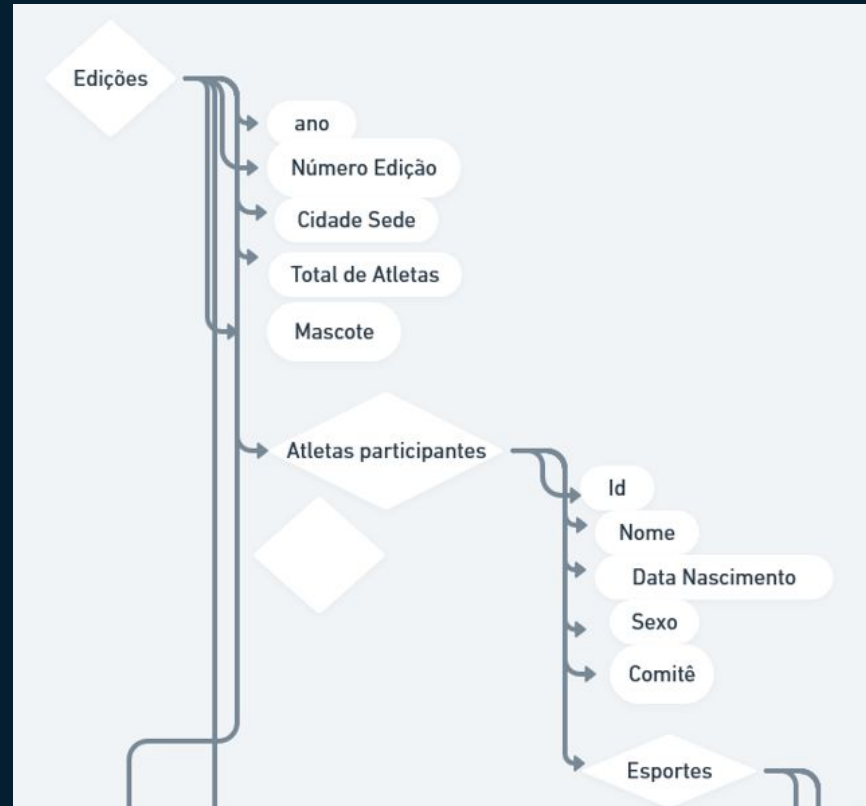
ComiteDosAtletas(IdComite, IdAtleta)

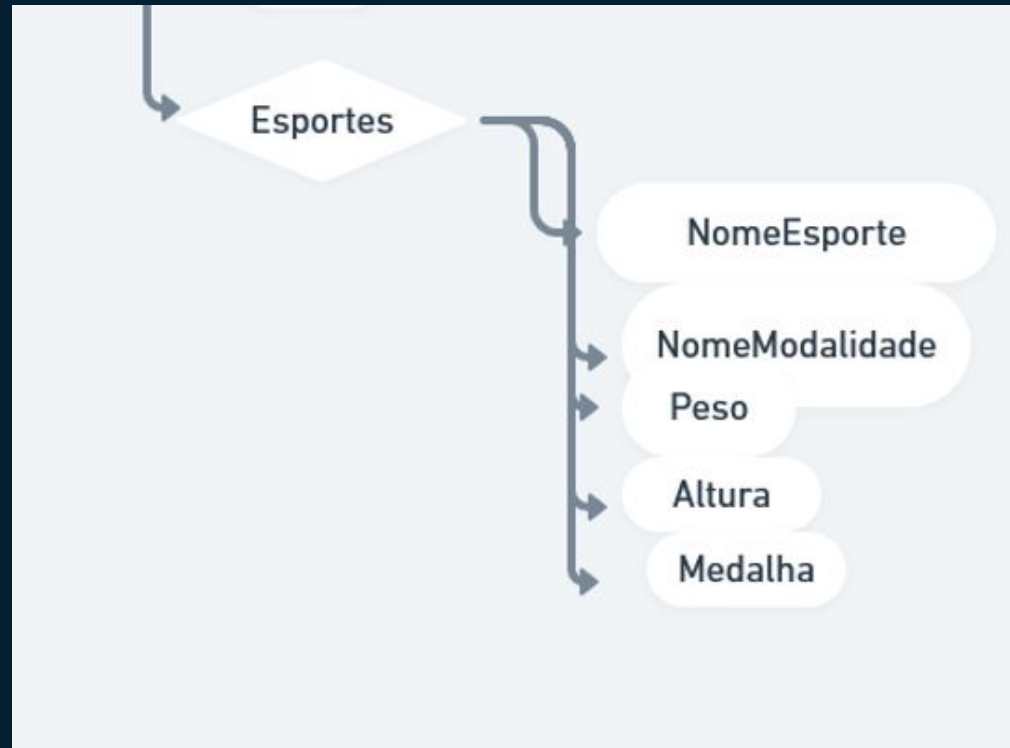
- Chaves estrangeiras: IdComite -> Comiteloimpico(Sigla), IdAtleta -> Atleta(Id)

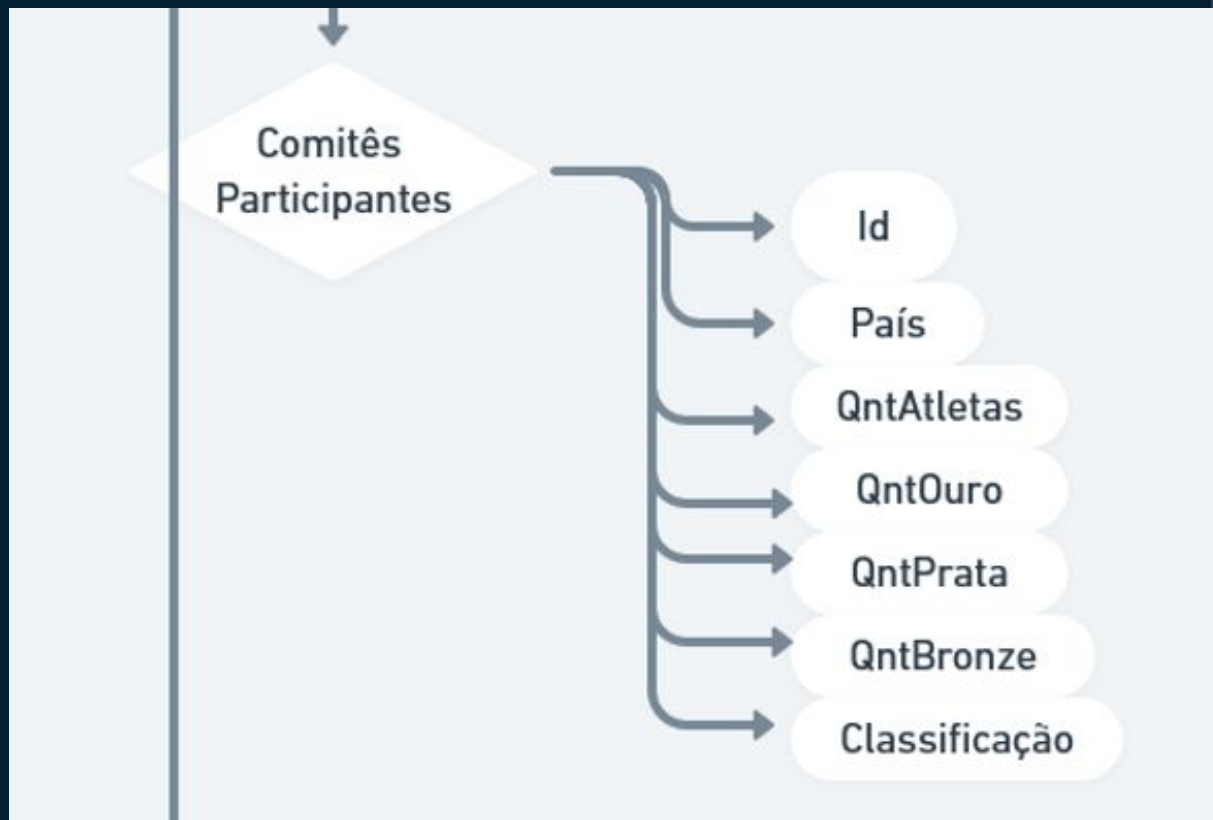
EsportesDasEdicoes(AnoEdicao, IdModalidade, Ouro, Prata, Bronze)

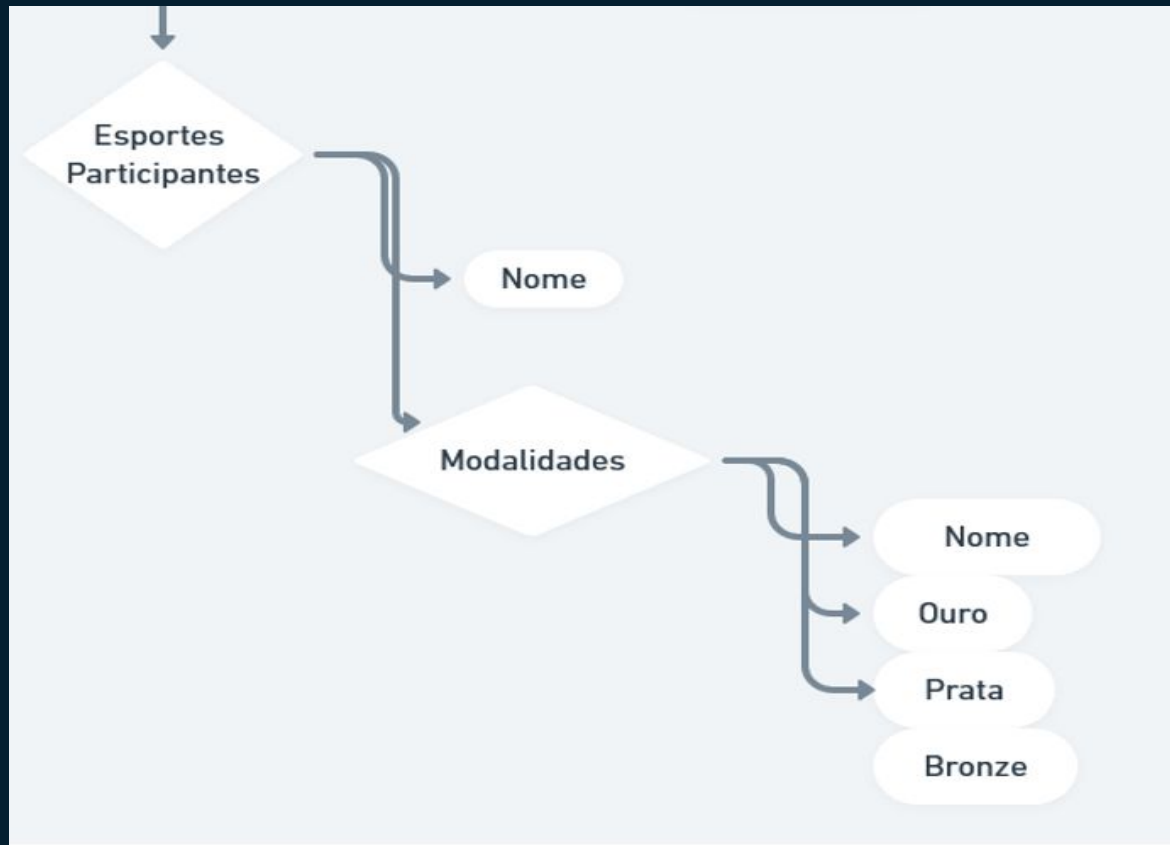
- Chaves estrangeiras: AnoEdicao -> EdicaoDosJogos(Ano), IdModalidade -> EsporteModalidade(Id)

# Modelo lógico: Documentos









# Operações aplicadas aos bancos

- **Extração:** Usada para complementação dos dados dos datasets estruturados, extraindo dados do site <http://olympics.com>.
- **Integração:** O dataset integra dados das nossas 3 fontes principais e de mais algumas auxiliares.
- **Tratamento:** Foram tomadas medidas em relação a dados faltantes, como anos de nascimento e sexo dos atletas.
- **Transformação:** Os dados foram transformados de forma a obtermos as tabelas especificadas no modelo lógico a partir dos dados brutos, tornando a análise mais prática e eficiente.



# Tratamento do Dataset:

## “120 years of Olympic history: athletes and results”

1. Leitura do CSV do dataset.
2. Seleção apenas dos jogos de verão de 2000 em diante.

```
[82] import pandas as pd
```

```
1  
▶ atletas_120 = pd.read_csv('athlete_events.csv')  
atletas_120=atletas_120.loc[(atletas_120["Season"] != 'Winter') & (atletas_120["Year"] >=2000)]  
atletas_120=atletas_120.reset_index(drop=True)  
atletas_120["Id"] = atletas_120.index  
2  
  
display(atletas_120)
```

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season
0	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer
1	12	Jyri Tapani Aalto	M	31.0	172.0	70.0	Finland	FIN	2000 Summer	2000	Summer
2	13	Minna Maarit Aalto	F	34.0	159.0	55.5	Finland	FIN	2000 Summer	2000	Summer

# Tratamento do Dataset:

## "120 years of Olympic history: athletes and results"

Tabela 1 - Edição dos Jogos

Ano	NumeroDaEdicao	CidadeSede	TotalDeAtletas	Mascote
-----	----------------	------------	----------------	---------

Podemos ver o padrão de criação das tabelas:

1. Seleção das colunas desejadas que estão no dataset fonte.
2. Renomeação adequada das colunas.
3. Criação de novas colunas vazias.
4. Preenchimento de dados, nesse caso, de forma manual (exceção).
5. Preenchimento de dados através de função e lógica, nesse caso, o número de atletas em uma certa edição é calculado selecionando-se os atletas de um certo ano e eliminando os atletas repetidos.
6. Preenchimento de dados nulos com hífen.

```
[84] edicaoDosJogos=pd.DataFrame(data=atletas_120[['Year', 'City']])
1   edicaoDosJogos=edicaoDosJogos.drop_duplicates("Year")
   edicaoDosJogos=edicaoDosJogos.sort_values(by=["Year"])
   edicaoDosJogos = edicaoDosJogos.reset_index(drop=True) #Reseta os index da tabela
2   edicaoDosJogos = edicaoDosJogos.rename({"Year": "Ano"}, axis=1)
   edicaoDosJogos = edicaoDosJogos.rename({"City": "Cidade"}, axis=1)
3   edicaoDosJogos["NumerodaEdicao"] = None
   edicaoDosJogos["TotaldeAtletas"] = None

4   for i in range(len(edicaoDosJogos.index)):
       edicaoDosJogos.at[i, 'NumerodaEdicao']=27+i

   edicaoDosJogos["Mascote"]=['Olly, Syd e Millie','Athena e Phevos','Beibei, Jingjing, H

5   # Função que ajuda a encontrar a qtd de atletas
   def setNumeroDeAtletas (row):
       atletasDeUmAno=pd.DataFrame(data=atletas_120[['Year', 'Name']])
       atletasDeUmAno=atletasDeUmAno.loc[atletasDeUmAno["Year"] ==row.Ano]
       atletasDeUmAno=atletasDeUmAno.drop_duplicates("Name")
       row.TotaldeAtletas=len(atletasDeUmAno.index)
       return row

   edicaoDosJogos = edicaoDosJogos.apply(lambda x: setNumeroDeAtletas(x),axis=1)

6   edicaoDosJogos=edicaoDosJogos.fillna("-")
   display(edicaoDosJogos)
```

	Ano	Cidade	NumerodaEdicao	TotaldeAtletas	Mascote
0	2000	Sydney	27	10639	Olly, Syd e Millie
1	2004	Athina	28	10537	Athena e Phevos
2	2008	Beijing	29	10880	Beibei, Jingjing, Huanhuan, Yingying e Nini
3	2012	London	30	10502	Wenlock
4	2016	Rio de Janeiro	31	11174	Vinicius

# Tratamento do Dataset:

## "120 years of Olympic history: athletes and results"

Tabela 2 - Atleta

Id	Nome	DataDeNascimento	Sexo
----	------	------------------	------

1. Cálculo do ano de nascimento através da subtração do ano da edição pela idade que o atleta tinha ao participar.

```
[119] atleta = pd.DataFrame(data=atletas_120[['Name', 'Sex', 'Age', 'Year']])
      atleta = atleta.drop_duplicates("Name")
      atleta = atleta.sort_values(by=["Name"])

      atleta = atleta.rename({"Name": "Nome"}, axis=1)
      atleta = atleta.rename({"Sex": "Sexo"}, axis=1)
      atleta = atleta.rename({"Age": "AnoDeNascimento"}, axis=1)

1  atleta["AnoDeNascimento"] = atleta["Year"] - atleta["AnoDeNascimento"]

      atleta = atleta.drop(columns=["Year"])
      atleta = atleta.reset_index(drop=True) #Reseta os index da tabela
      atleta["Id"] = atleta.index
      atleta = atleta.reindex(columns=["Id", "Nome", "AnoDeNascimento", "Sexo"])

      atleta = atleta.fillna("-")
      atleta.head(50)
```

	Id	Nome	AnoDeNascimento	Sexo
0	0	Gabrielle Marie "Gabby" Adcock (White-)	1991	F
1	1	Th Anh	1996	F
2	2	Th Ngn Thng	1989	F
3	3	A Lamusi	1989	M
4	4	A. Joshua "Josh" West	1977	M
5	5	Aadam Ismaeel Khamis	1989	M
6	6	Aagje Varwalleghe	1988	F

# Tratamento do Dataset:

## "120 years of Olympic history: athletes and results"

Tabela 3 - Comitê Olímpico	
Id	País

Seleção da coluna de países e de sua sigla  
NOC (National Olympic Committee).

```
[90] #Esse comando foi só pra criar uma nova tabela com base na coluna de outra
comiteOlimpico = pd.DataFrame(data=atletas_120[['NOC', 'Team']])
comiteOlimpico = comiteOlimpico.drop_duplicates() #Remove linhas duplicadas
comiteOlimpico = comiteOlimpico.sort_values("NOC") #Ordena pela coluna 'NOC'
comiteOlimpico = comiteOlimpico.reset_index(drop=True) #Reseta os index da tabela
comiteOlimpico = comiteOlimpico.rename({"NOC": "Id"}, axis=1)
comiteOlimpico = comiteOlimpico.rename({"Team": "País"}, axis=1)
#comiteOlimpico = comiteOlimpico.drop_duplicates(['Id']) #Remove linhas duplicadas
comiteOlimpico = comiteOlimpico.reindex(columns=["Id", "País"])
```

```
[91] comiteOlimpico=comiteOlimpico.fillna("-")
display(comiteOlimpico)
```

	Id	País
0	AFG	Afghanistan
1	AHO	Netherlands Antilles
2	ALB	Albania
3	ALG	Algeria
4	AND	Andorra

# Tratamento do Dataset:

## “120 years of Olympic history: athletes and results”

Tabela 4 - Esporte/Modalidade

Id	Nome	EsportePai
----	------	------------

1. Tabela de modalidades e os esportes das quais pertencem.
2. Tabela esportes (que não pertencem a nenhum outro, contendo valor nulo na coluna “EsportePai”).
3. Concatenação das duas tabelas.

```
▶ esporteModalidade1 = pd.DataFrame(data=atletas_120[['Event', 'Sport']])
esporteModalidade1 = esporteModalidade1.drop_duplicates()
esporteModalidade1 = esporteModalidade1.sort_values("Event")
1 esporteModalidade1 = esporteModalidade1.reset_index(drop=True)
esporteModalidade1 = esporteModalidade1.rename({"Event": "Nome"}, axis=1)
esporteModalidade1 = esporteModalidade1.rename({"Sport": "EsportePai"}, axis=1)

esporteModalidade2 = pd.DataFrame(data=atletas_120[['Sport']])
esporteModalidade2 = esporteModalidade2.rename({"Sport": "Nome"}, axis=1)
2 esporteModalidade2 = esporteModalidade2.drop_duplicates()
esporteModalidade2["EsportePai"] = None
esporteModalidade2 = esporteModalidade2.sort_values("Nome")
esporteModalidade2 = esporteModalidade2.reset_index(drop=True)

esporteModalidade = pd.concat([esporteModalidade1, esporteModalidade2])
esporteModalidade = esporteModalidade.sort_values("Nome")
3 esporteModalidade = esporteModalidade.reset_index(drop=True)
esporteModalidade["Id"] = esporteModalidade.index
esporteModalidade = esporteModalidade.reindex(columns=["Id", "Nome", "EsportePai"])

[93] esporteModalidade = esporteModalidade.fillna("-")
esporteModalidade.head(50)
```



# Tratamento do Dataset:

## “120 years of Olympic history: athletes and results”

Tabela 5 - Participação Comitês						
IdComite	AnoEdicao	QtdAtletas	QtdOuro	QtdPrata	QtdBronze	Classificacao

1. Seleciona os atletas com medalha de ouro, prata ou bronze, para determinado país em determinado ano.  
Remove as modalidades repetidas, já que pode haver mais de uma medalha na mesma modalidade, se for um time.
2. Seleciona todos os atletas de um país em determinado ano. Remove os nomes duplicados.
3. Converte os 3 tipos de medalha em uma pontuação, para calcular a classificação em seguida.

```
def setQtdMedalhasAtletas (row):  
    infoMedalhas=pd.DataFrame(data=atletas_120[['Year', 'NOC', 'Event', 'Medal', 'Name']])  
  
    1 infoOuroPais=infoMedalhas.loc[(infoMedalhas["Year"] ==row.AnoEdicao) & (infoMedalhas["NOC"] == row.IdComite) & (infoMedalhas["Medal"]=="Gold")]  
    infoOuroPais=infoOuroPais.drop_duplicates ("Event")  
    row.QtdOuro=len(infoOuroPais.index)  
  
    infoPrataPais=infoMedalhas.loc[(infoMedalhas["Year"] ==row.AnoEdicao) & (infoMedalhas["NOC"] == row.IdComite) & (infoMedalhas["Medal"]=="Silver")]  
    infoPrataPais=infoPrataPais.drop_duplicates ("Event")  
    row.QtdPrata=len(infoPrataPais.index)  
  
    infoBronzePais=infoMedalhas.loc[(infoMedalhas["Year"] ==row.AnoEdicao) & (infoMedalhas["NOC"] == row.IdComite) & (infoMedalhas["Medal"]=="Bronze")]  
    infoBronzePais=infoBronzePais.drop_duplicates ("Event")  
    row.QtdBronze=len(infoBronzePais.index)  
  
    2 infoQtdAtletasPais=infoMedalhas.loc[(infoMedalhas["Year"] ==row.AnoEdicao) & (infoMedalhas["NOC"] == row.IdComite)]  
    infoQtdAtletasPais=infoQtdAtletasPais.drop_duplicates('Name')  
    row.QtdAtletas=len(infoQtdAtletasPais.index)  
  
    3 row.PontuacaoTemp=10000*row.QtdOuro+100*row.QtdPrata+1*row.QtdBronze  
  
    return row  
  
participacaoComites= participacaoComites.apply(lambda x: setQtdMedalhasAtletas(x),axis=1)
```

# Tratamento do Dataset:

## "120 years of Olympic history: athletes and results"

Tabela 5 - Participação Comitês

IdComite	AnoEdicao	QtdAtletas	QtdOuro	QtdPrata	QtdBronze	Classificacao
----------	-----------	------------	---------	----------	-----------	---------------

1. É criada uma tabela para cada ano e os países são ordenados pela pontuação.
2. Função que determina a classificação para cada tabela, levando em conta que países com a mesma pontuação (quantidade de medalhas de cada tipo exatamente igual) tem a mesma classificação, e há um salto para a classificação seguinte.
3. Concatenação das tabelas de cada ano em uma só.

```
quadro2008 = quadro2008.reset_index(drop=True)
```

```
quadro2012=participacaoComites.loc[(participacaoComites["AnoEdicao"] == 2012)]  
quadro2012=quadro2012.sort_values(by=['PontuacaoTemp'], ascending=False)  
quadro2012 = quadro2012.reset_index(drop=True)
```

```
quadro2016=participacaoComites.loc[(participacaoComites["AnoEdicao"] == 2016)]  
quadro2016=quadro2016.sort_values(by=['PontuacaoTemp'], ascending=False)  
quadro2016 = quadro2016.reset_index(drop=True)
```

```
def setClassificacao(quadro):  
    classificacao=0  
    salto=0  
    pontuacao_anterior=10000000  
    for i in range(len(quadro.index)):  
        if quadro.at[i, 'PontuacaoTemp']<pontuacao_anterior:  
            classificacao+=1+salto  
            salto=0  
            pontuacao_anterior=quadro.at[i, 'PontuacaoTemp']  
            quadro.at[i, 'Classificacao']=classificacao  
            if pontuacao_anterior==0:  
                quadro.at[i, 'Classificacao']="-."  
        else:  
            quadro.at[i, 'Classificacao']=classificacao  
            salto+=1  
            if pontuacao_anterior==0:  
                quadro.at[i, 'Classificacao']="-."
```

```
setClassificacao(quadro2000)  
setClassificacao(quadro2004)  
setClassificacao(quadro2008)  
setClassificacao(quadro2012)  
setClassificacao(quadro2016)
```

```
frames=[quadro2000, quadro2004, quadro2008, quadro2012, quadro2016]  
participacaoComites = pd.concat(frames)
```

# Tratamento do Dataset:

## “120 years of Olympic history: athletes and results”

Tabela 6 - Participação Atletas

IdAtleta	AnoEdicao	IdModalidade	Altura	Peso	Medalha
----------	-----------	--------------	--------	------	---------

1. Funções que percorrem as tabelas de atletas e esportes/modalidades para substituir seus nomes por seus respectivos Id.

```
[102] participacaoAtletas=pd.DataFrame(data=atletas_120[['Name', 'Year', 'Event', 'Height', 'Weight', 'Medal']])
participacaoAtletas=participacaoAtletas.rename({"Name":"IdAtleta"}, axis=1)
participacaoAtletas=participacaoAtletas.rename({"Year":"AnoEdicao"}, axis=1)
participacaoAtletas=participacaoAtletas.rename({"Event":"IdModalidade"}, axis=1)
participacaoAtletas=participacaoAtletas.rename({"Height":"Altura"}, axis=1)
participacaoAtletas=participacaoAtletas.rename({"Weight":"Peso"}, axis=1)
participacaoAtletas=participacaoAtletas.rename({"Medal":"Medalha"}, axis=1)

def setIdAtleta(row):
    nomeAtleta = row.IdAtleta
    id = atleta[atleta['Nome']==nomeAtleta].Id.values[0]
    row.IdAtleta = id
    return row

def setIdModalidade(row):
    nomeModalidade = row.IdModalidade
    id = esporteModalidade[esporteModalidade['Nome']==nomeModalidade].Id.values[0]
    row.IdModalidade = id
    return row

participacaoAtletas = participacaoAtletas.apply(lambda x: setIdAtleta(x),axis=1) #Aplico a função acima po
participacaoAtletas = participacaoAtletas.apply(lambda x: setIdModalidade(x),axis=1) #Aplico a função acir
participacaoAtletas = participacaoAtletas.reset_index(drop=True)
```



# Tratamento do Dataset:

## "120 years of Olympic history: athletes and results"

Tabela 7 - Comitê dos Atletas

IdComite	IdAtleta
----------	----------

1. Função que percorre a tabela de atletas para substituir seus nomes pelos respectivos Id.

```
[104] comiteDosAtletas=pd.DataFrame(data=atletas_120[['NOC', 'Name']])  
comiteDosAtletas=comiteDosAtletas.drop_duplicates ("Name")
```

```
comiteDosAtletas=comiteDosAtletas.rename({"Name":"IdAtleta"}, axis=1)  
comiteDosAtletas=comiteDosAtletas.rename({"NOC":"IdComite"}, axis=1)
```

```
1 def setIdAtleta(row):  
    nomeAtleta = row.IdAtleta  
    id = atleta[atleta['Nome']==nomeAtleta].Id.values[0]  
    row.IdAtleta = id  
    return row
```

```
comiteDosAtletas = comiteDosAtletas.apply(lambda x: setIdAtleta(x),axis=1)
```

# Tratamento do Dataset:

## “120 years of Olympic history: athletes and results”

Tabela 8 - Esportes das Edições

AnoEdicao	IdModalidade	Ouro	Prata	Bronze
-----------	--------------	------	-------	--------

1- Função que, através da pesquisa por ano, modalidade e medalha de ouro, prata ou bronze, encontra um atleta que ganhou uma medalha que atinja essas especificações, pega o país a qual ele pertence, e coloca na posição esperada na tabela 8.

```
esportesDasEdicoes=pd.DataFrame(data=atletas_120[['Year', 'Event']])
esportesDasEdicoes=esportesDasEdicoes.rename({"Year": "AnoEdicao"}, axis=1)
esportesDasEdicoes=esportesDasEdicoes.drop_duplicates(['AnoEdicao', 'Event'])
esportesDasEdicoes["Ouro"] = None
esportesDasEdicoes["Prata"]=None
esportesDasEdicoes["Bronze"]=None
```

```
def setPodio(row, infoPodio):
```

```
    try:
        infoPodioOuro=infoPodio.loc[(infoPodio["Year"] ==row.AnoEdicao) & (infoPodio["Event"] == row.Event) & (infoPodio["Medal"]=="Gold")]
        infoPodioOuro=infoPodioOuro.reset_index(drop=True)
        row.Ouro=infoPodioOuro.at[0, 'NOC']
```

```
    except:
        row.Ouro="-"
```

```
    try:
        infoPodioPrata=infoPodio.loc[(infoPodio["Year"] ==row.AnoEdicao) & (infoPodio["Event"] == row.Event) & (infoPodio["Medal"]=="Silver")]
        infoPodioPrata=infoPodioPrata.reset_index(drop=True)
        row.Prata=infoPodioPrata.at[0, 'NOC']
```

```
    except:
        row.Prata="-"
```

```
    try:
        infoPodioBronze=infoPodio.loc[(infoPodio["Year"] ==row.AnoEdicao) & (infoPodio["Event"] == row.Event) & (infoPodio["Medal"]=="Bronze")]
        infoPodioBronze=infoPodioBronze.reset_index(drop=True)
        row.Bronze=infoPodioBronze.at[0, 'NOC']
```

```
    except:
        row.Bronze="-"
```

```
    return row
```

# Tratamento do Dataset: "2021 Olympics in Tokyo"

- Fornece dados básicos sobre os atletas e sobre as medalhas de cada comitê.
- Dataset com menos informações. Não possui modalidades dos esportes nem informações como ano de nascimento, sexo, altura e peso dos atletas.

## Atletas

```
#Comando para carregar o dataset. Se o seu dataset for em  
atletas_original = pd.read_excel('Athletes.xlsx')  
  
#Comando 'head' mostra as 5 primeiras linhas da tabela  
atletas_original.head()
```

	Name	NOC	Discipline
0	AALERUD Katrine	Norway	Cycling Road
1	ABAD Nestor	Spain	Artistic Gymnastics
2	ABAGNALE Giovanni	Italy	Rowing
3	ABALDE Alberto	Spain	Basketball
4	ABALDE Tamara	Spain	Basketball

```
[194] medalhas1 = pd.read_excel("Medals.xlsx")  
medalhas1.head()
```

	Rank	Team/NOC	Gold	Silver	Bronze	Total	Rank by Total
0	1	United States of America	39	41	33	113	1
1	2	People's Republic of China	38	32	18	88	2
2	3	Japan	27	14	17	58	5
3	4	Great Britain	22	21	22	65	4
4	5	ROC	20	28	23	71	3



SIMONE BILES



United States of America



Artistic Gymnastics

Olympic Medals

4

G

1

S

2

B

Games participations

2

First Olympic Games

Rio 2016

Year of Birth

1997

Social Media



Olympic Results



Site "olympics.com"

RESULTS	EVENT	SPORT
Tokyo 2020		
#n/a	Women's All-Around	Artistic Gymnastics
B	Women's Balance Beam	Artistic Gymnastics
#n/a	Women's Floor Exercise	Artistic Gymnastics
S	Women's Team	Artistic Gymnastics
#n/a	Women's Uneven Bars	Artistic Gymnastics
#n/a	Women's Vault	Artistic Gymnastics
Rio 2016		
B	Balance Beam	Artistic Gymnastics
G	Floor Exercise	Artistic Gymnastics
G	Horse Vault	Artistic Gymnastics

```
# Trecho de código que realiza requisições ao site "olympics.com" e obtém  
# ano de nascimento do atleta e modalidades em que participou  
for i in range(len(atletas)):
```

```
    ###  
    # Tratamento do nome do atleta  
    ###
```

```
    |  
    try:  
        r = requests.get("https://olympics.com/en/athletes/"+nomeUrl)  
        tree = html.fromstring(r.content)
```

Requisição ao site para  
obter dados de um  
atleta específico

```
        details = tree.xpath('//ul[@class="detail_list"]/li/div/text()')  
        b = details.index("Year of Birth")  
        year = details[b+1]  
        atletas.loc[i, "Ano"] = year
```

```
        tabela_part = tree.xpath('//table[@class="sm-mb6 has-header"]/tbody/tr')
```

```
        nomeOlimp = tabela_part[0].xpath('./h2/text()')
```

```
        if len(nomeOlimp)>0:
```

```
            if nomeOlimp[0] == "Tokyo 2020":
```

```
                for tr in tabela_part:
```

```
                    td = tr.xpath('./td')
```

```
                    medalha = td[1].xpath('./div')[0].text_content().replace("'", '').replace("\n", "").replace("\r", "").replace(" ", "")
```

```
                    modalidade = td[2].text_content()
```

```
                    esporte = td[3].text_content()
```

```
                    atletasEsportes.loc[len(atletasEsportes)] = [nomeReal, nomeUrl, esporte, modalidade, medalha]
```

```
except:
```

```
    print(nomeReal)
```

Uso de xpath para  
identificação de  
elementos no html



# Informações obtidas do site

	Nome	Esporte	Modalidade	Medalha
100	Valentina Acosta Giraldo	Archery	Mixed Team	#26
101	Valentina Acosta Giraldo	Archery	Women's Individual	#=33
102	Yenny Acuna	Football	Women	#11
103	Kazuya Adachi	Canoe Slalom	Men's Kayak	#16
104	Seiya Adachi	Water Polo	Men	#10
105	Amal Adam	Archery	Mixed Team	#29
106	Amal Adam	Archery	Women's Individual	#=33
107	Constantin Adam	Rowing	Men's Eight	#7
108	Klaudia Adamek	Athletics	Women's 4 x 100m Relay	#n/a
109	Patrycja Adamkiewicz	Taekwondo	Women -57kg	#=11
110	Liam Adams	Athletics	Men's Marathon	#24
111	Paul Adams	Shooting	Skeet Men	#21
112	Taeyanna Adams	Swimming	Women's 100m Breaststroke	#n/a
113	Yasemin Adar	Wrestling	Women's Freestyle 76kg	B

Id	Nome	Ano	Sexo
0	Katrine Aalerud	1994	F
1	Nestor Abad	1993	M
2	Giovanni Abagnale	1995	M
3	Alberto Abalde	NaN	None
4	Tamara Abalde	1989	F
5	Luc Abalo	1984	M
6	Cesar Abaroa	1996	M
7	Abobakr Abass	1998	M
8	Hamideh Abbasali	1990	F
9	Islam Abbasov	1996	M
10	Lois Abbingh	1992	F
11	Emily Abbot	1997	None
12	Monica Abbott	1985	None
13	Abubaker Haydar Abdalla	1996	M
14	Maryam Abdalla	NaN	None

# Problemas enfrentados

- Como não há o nome correto de todos os atletas e alguns atletas possuem informações faltantes no site, a tabela ficou com alguns campos incompletos.
- Para alguns dados, realmente não havia fontes de onde retirar a informação, como altura e peso dos atletas.
- Ainda será necessária uma análise detalhada para conseguir integrar os dados de Tokyo com os dos jogos anteriores.

```
Id      0
Nome    0
Ano     1009
Sexo    1950
dtype: int64
```

# Conjunto inicial de perguntas

- Quais os países que mais ganharam medalhas e os países que menos ganharam medalhas em uma determinada Olimpíada?
- Qual o número médio de medalhas de um país nas Olimpíadas que ele participou?
- Qual o número de atletas por país em uma determinada olimpíada?
- Quais foram os países ganhadores de medalha de ouro no esporte X nas últimas 5 olimpíadas?
- Qual a proporção de atletas do sexo masculino e do sexo feminino participando nos Jogos Olímpicos?
- Em quantas Olimpíadas um determinado atleta participou e quantas medalhas ele ganhou?
- Quais países que mais trazem atletas para os Jogos Olímpicos?
- Para um determinado esporte, existe algum país que constantemente está no pódio?
- Para os países que trazem um número reduzido de atletas, em quais esportes eles costumam participar?



1. Quais os países que mais ganharam medalhas e os países que menos ganharam medalhas em uma determinada Olimpíada?

```
/*Pergunta 1*/  
SELECT DISTINCT C.PAIS, CAST (P.Classificacao AS INT) Classificacao  
FROM ParticipacaoComites P, ComiteOlimpico C  
WHERE C.Sigla=P.IdComite AND AnoEdicao=2016 AND Classificacao<>'-'  
ORDER BY Classificacao
```

index	PAIS	CLASSIFICACAO			
0	United States	1	80	United Arab Emirates	78
1	Great Britain	2	81	Morocco	78
2	China	3	82	Portugal	78
3	Russia	4	83	Finland	78
4	Germany	5	84	Austria	78
			85	Estonia	78

2. Qual o número médio de medalhas de um país nas Olimpíadas que ele participou?

```
/*Pergunta 2*/  
SELECT C.PAIS, (SUM(P.QTDouro)+SUM(P.QTDPrata)+SUM(P.QTDBronze))/COUNT(*) MEDIA  
FROM PARTICIPACAOComites P, COMITEOLIMPICO C  
WHERE P.IDComite=C.SIGLA  
GROUP BY C.PAIS  
ORDER BY MEDIA DESC
```

index	PAIS	MEDIA
0	United States	105
1	Russia	77
2	China	75
3	Great Britain	47
4	Germany	46
5	Australia	43
6	France	37
7	Japan	31
8	Italy	29
9	South Korea	27
10	Cuba	21

3. Quais foram os países ganhadores de medalha de ouro no esporte X nas últimas 5 olimpíadas?

```
/*Pergunta 3*/  
SELECT DISTINCT E.AnoEdicao, M.Nome, E.Ouro  
FROM EsportesDasEdicoes E, EsporteModalidade M  
WHERE E.IdModalidade=M.Id AND E.AnoEdicao >= 2000 AND M.Id=8  
ORDER BY E.AnoEdicao
```

index	ANOEDICAO	NOME	OURO
0	2000	Athletics Men's 100 metres	USA
1	2004	Athletics Men's 100 metres	USA
2	2008	Athletics Men's 100 metres	JAM
3	2012	Athletics Men's 100 metres	JAM
4	2016	Athletics Men's 100 metres	JAM

4. Qual a proporção de atletas do sexo masculino e do sexo feminino participando nos Jogos Olímpicos?

```
/*Pergunta 4*/  
SELECT P.AnoEdicao, A.Sexo, COUNT(*) Total  
FROM Atleta A, ParticipacaoAtletas P  
WHERE A.ID=P.IDATLETA  
GROUP BY P.AnoEdicao, A.Sexo  
ORDER BY P.AnoEdicao
```

index	ANOEDICAO	SEXO	TOTAL
0	2000	M	8384
1	2000	F	5437
2	2004	F	5546
3	2004	M	7897
4	2008	M	7776
5	2008	F	5826
6	2012	M	7108
7	2012	F	5812
8	2016	F	6220
9	2016	M	7468

5. Em quantas Olimpíadas um determinado atleta participou e quantas medalhas ele ganhou?

*/\*Pergunta 5\*/*

```
SELECT A.NOME, P.MEDALHA, COUNT(*) TOTAL
FROM PARTICIPACAOATLETAS P, ATLETA A
WHERE P.IDATLETA=A.ID AND A.ID=1467 AND P.MEDALHA<>'-'
GROUP BY P.IDATLETA, P.MEDALHA
```

index	NOME	MEDALHA	TOTAL
0	Alfredo Rota	Bronze	1
1	Alfredo Rota	Gold	1

```
SELECT NOME, COUNT(*) TOTALOLIMPIADAS
FROM (SELECT A.NOME, COUNT(*) TOTALJOGOS
FROM PARTICIPACAOATLETAS P, ATLETA A
WHERE P.IDATLETA=A.ID AND A.ID=1467
GROUP BY P.IDATLETA, P.ANOEDICAO)
GROUP BY NOME
```

index	Key	Value
0	NOME	Alfredo Rota
1	TOTALOLIMPIADAS	3

The background is a dark navy blue. In the top-left and bottom-left corners, there are overlapping, semi-transparent geometric shapes in shades of green, blue, orange, and pink. In the top-right and bottom-right corners, there are similar overlapping shapes in shades of green, blue, purple, and orange. The word "Obrigado!" is centered in the middle of the slide in a white, bold, sans-serif font.

**Obrigado!**