

# Trabalho de Infraestrutura Kafka

## Instituto Infnet

**Aluno: Charles de Araujo Melo**

1. Escolha 5 conceitos fundamentais sobre o Apache Kafka e os descreva.

**R: 1.** Kafka é uma plataforma distribuída de streaming de dados que tem basicamente 3 capacidades:

**Publicar e inscrever** fluxos (stream) de dados semelhante a uma fila de mensagens;

**Armazenar** stream de dados em um sistema tolerante a falhas;

**Processar** esse stream de dados;

**2.** Possui o Broker ou Kafka Node é uma instância de servidor que roda dentro do Kafka Cluster. O Kafka Cluster pode ser formado por um ou mais brokers. Além de atuar como um servidor, o broker tem a responsabilidade de armazenar partições dos tópicos.

**3.** Mensagem:

A mensagem dentro do Kafka é o elemento que transita entre todos os pontos da arquitetura. Ela contém os dados preenchidos pelo produtor, é armazenada dentro do tópico aguardando a leitura por parte um ou mais consumidores, até finalmente, ser utilizada pelo consumidor.

Cada mensagem é composta, sobretudo, por estes 3 itens:

**4.** Tópico (Topic):

O tópico é onde separamos as mensagens pelo seu objetivo, pela sua categoria.

Para ser criado, um tópico precisa de um nome, o número de partições (partitions) que ele terá e o fator de replicação (replication factor) que é o número de partições em sincronismo que esse tópico terá.

**5.** Produtor (Producer) e Consumidor (Consumer)

O produtor pode ser um comando no terminal ou qualquer aplicação responsável por enviar uma mensagem para um tópico do Kafka.

O consumidor é responsável por ler uma mensagem de uma ou mais partições contidas em um tópico. Quando um consumidor está online e pronto para ler uma mensagem, ele solicita ao tópico a próxima mensagem para leitura.

2. Descreva como é a arquitetura do Apache Kafka.

**R:** Sua arquitetura consiste em um log imutável de mensagens que podem ser organizadas em tópicos para consumo por vários usuários ou aplicativos. Um sistema de arquivos ou log de confirmação de banco de dados mantém um registro permanente de todas as mensagens para que o Kafka possa reproduzi-las para manter um estado de sistema consistente.

### 3. Apresente exemplos de utilização do Apache Kafka em bases NoSQL e SQL.

**R:** O Apache Kafka tem sido usado por empresas como Netflix, Spotify, Uber, LinkedIn e Twitter. E sua arquitetura é composta por producers, consumers e o próprio cluster.

O producer é qualquer aplicação que publica mensagens no cluster. O consumer é qualquer aplicação que recebe as mensagens do Kafka. O cluster Kafka é um conjunto de nós que funcionam como uma instância única do serviço de mensagem.

Um cluster Kafka é composto por vários brokers. Um broker é um servidor Kafka que recebe mensagens dos producers e as grava no disco. Cada broker gerencia uma lista de tópicos e cada tópico é dividido em diversas partições.

Depois de receber as mensagens, o broker as envia para os consumidores que estão registrados para cada tópico.

As configurações do Apache Kafka são gerenciadas pelo Apache Zookeeper, que armazena os metadados do cluster, como localização das partições, lista de nomes, lista de tópicos e nós disponíveis. Assim, o Zookeeper mantém a sincronização entre os diversos elementos do cluster.

Isso é importante porque o Kafka é um sistema distribuído, ou seja, as gravações e leituras são feitas por diversos clientes simultaneamente. Quando há uma falha, o Zookeeper elege um substituto e recupera a operação.

### 4. Descrevas os principais benefícios em utilizar o Apache Kafka.

**R: Escalabilidade:** o cluster pode ser facilmente redimensionado para atender ao aumento ou diminuição das cargas de trabalho.

**Distribuído:** o cluster Kafka opera com vários nós, dividindo o processamento. Replicado, particionado e ordenado: as mensagens são replicadas em partições nos nós do cluster na ordem em que chegam para garantir segurança e velocidade de entrega.

**Alta disponibilidade:** o cluster tem diversos nós (brokers) e várias cópias dos dados, assim.

### 5. O que é um pipeline de dados?

**R:** O pipeline de dados permite que dados de fontes diferentes sejam integrados com eficiência. Por meio dele, é possível analisar dados relativos ao comportamento do cliente-alvo, que pode ser o consumidor final, automação de processos, jornadas do comprador e experiências do cliente.

### 6. Dê 2 (dois) exemplos de aplicações onde os pipelines de dados são utilizados em seu dia-a-dia.

**R:** Contabilidade mensal como transações em bancos e utilização do netflix..

7. Selecione uma base de dados pública brasileira para utilizar neste exercício. Você pode baixá-la em algum formato que desejar (ex.: formato .csv). Informe onde e como você conseguiu os seus dados. Explique se são estruturados ou não estruturados. Cada linha/registro em seu banco de dados corresponde a quais informações? Cada registro possui quantas colunas associadas e quais atributos elas representam? Qual o tamanho do banco de dados escolhido?

**R:** Consegui no site de dados públicos pelo linky [https://www.kaggle.com/datasets/fidelissauro/Food\\_Preference](https://www.kaggle.com/datasets/fidelissauro/Food_Preference)

São dados estruturados por serem bem definidos possuindo linhas e tabelas ordenados.

Cada linha / registro no meu banco de dados corresponde ao registro de transações como levantamento de dados sobre os consumidores de uma lanchonete.

8. Formule pelo menos 2 perguntas sobre sua base de dados. O que você quer saber sobre os dados que escolheu?

**R:** **Quantidade** de consumidores do sexo feminino.

Quantidade de consumidores do sexo masculino.

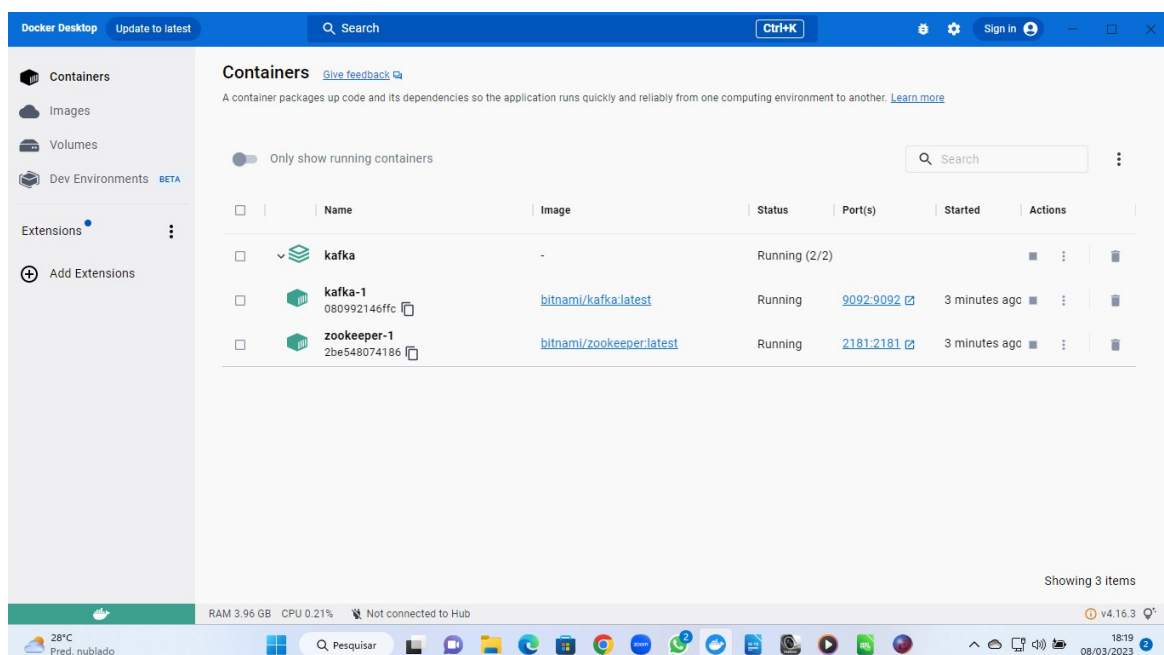
9. Formule uma hipótese sobre o que você acha que vai encontrar quando filtrar e analisar seus dados.

**R:** Hipótese que o sexo feminino consome mais na lanchonete do que o masculino.

10. Crie uma nova variável a partir de outras variáveis da base de dados que te auxilie na avaliação de sua hipótese.

**R:** summary\_quant

11. Importe a sua base de dados na infraestrutura Kafka. Inclua em seu relatório a forma que você realizou a importação.



```
Selecionar Jupyter Notebook (anaconda3)
[W 2023-03-08 18:20:57.004 LabApp] 'notebook_dir' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[W 2023-03-08 18:20:57.005 LabApp] 'notebook_dir' has moved from NotebookApp to ServerApp. This config will be passed to ServerApp. Be sure to update your config before our next release.
[I 2023-03-08 18:20:57.021 LabApp] JupyterLab extension loaded from C:\Users\fernu\anaconda3\lib\site-packages\jupyterlab
[I 2023-03-08 18:20:57.022 LabApp] JupyterLab application directory is C:\Users\fernu\anaconda3\share\jupyter\lab
[I 18:20:57.031 NotebookApp] Serving notebooks from local directory: C:\Users\fernu
[I 18:20:57.031 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 18:20:57.031 NotebookApp] http://localhost:8888/?token=a0921867f71a42adfcfe269f9283b6d168da10a74bf2bd57
[I 18:20:57.032 NotebookApp] or http://127.0.0.1:8888/?token=a0921867f71a42adfcfe269f9283b6d168da10a74bf2bd57
[I 18:20:57.033 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 18:20:57.074 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/fernu/AppData/Roaming/jupyter/runtime/nbserver-16936-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=a0921867f71a42adfcfe269f9283b6d168da10a74bf2bd57
or http://127.0.0.1:8888/?token=a0921867f71a42adfcfe269f9283b6d168da10a74bf2bd57
[I 18:21:16.448 NotebookApp] Uploading file to /Food_Preference.csv
```

Importei minha base de dados para o kafka utilizando o jupyter

12. Realize pré-processamento dos dados importados. Inclua eu seu relatório os códigos utilizados para o pré-processamento e criação de novas variáveis.

```
jupyter Charles_Producer Last Checkpoint: Ontem às 21:35 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

print(f'Iniciando o Kafka producer at {dt.datetime.utcnow()}')
Iniciando o Kafka producer at 2023-03-08 23:16:53.126909

In [3]: #Configurar um contador e definir o caminho do arquivo
counter = 0
file = 'Food_Preference.csv'

In [4]: with open(file, 'r') as new_obj:
        csv_dict_reader = DictReader(new_obj)
        for row in csv_dict_reader:
            key = str(counter).encode()
            ack = producer.send(topic='Food', key=key, value=json.dumps(row).encode('utf-8'))
            metadata = ack.get()
            counter = counter + 1
            print(metadata.topic, metadata.partition, key)

Food 0 b'270'
Food 0 b'271'
Food 0 b'272'
Food 0 b'273'
Food 0 b'274'
Food 0 b'275'
Food 0 b'276'
Food 0 b'277'
Food 0 b'278'
Food 0 b'279'
Food 0 b'280'
Food 0 b'281'
Food 0 b'282'
Food 0 b'283'
Food 0 b'284'
Food 0 b'285'
Food 0 b'286'
Food 0 b'287'
Food 0 b'288'
```

### #Configurar um contador e definir o caminho do arquivo

```
counter = 0
```

```
file = 'Food_Preference.csv'
```

### #Pré processamento dos dados

```
with open(file, 'r') as new_obj:
```

```
    csv_dict_reader = DictReader(new_obj)
```

```
    for row in csv_dict_reader:
```

```
        key = str(counter).encode()
```

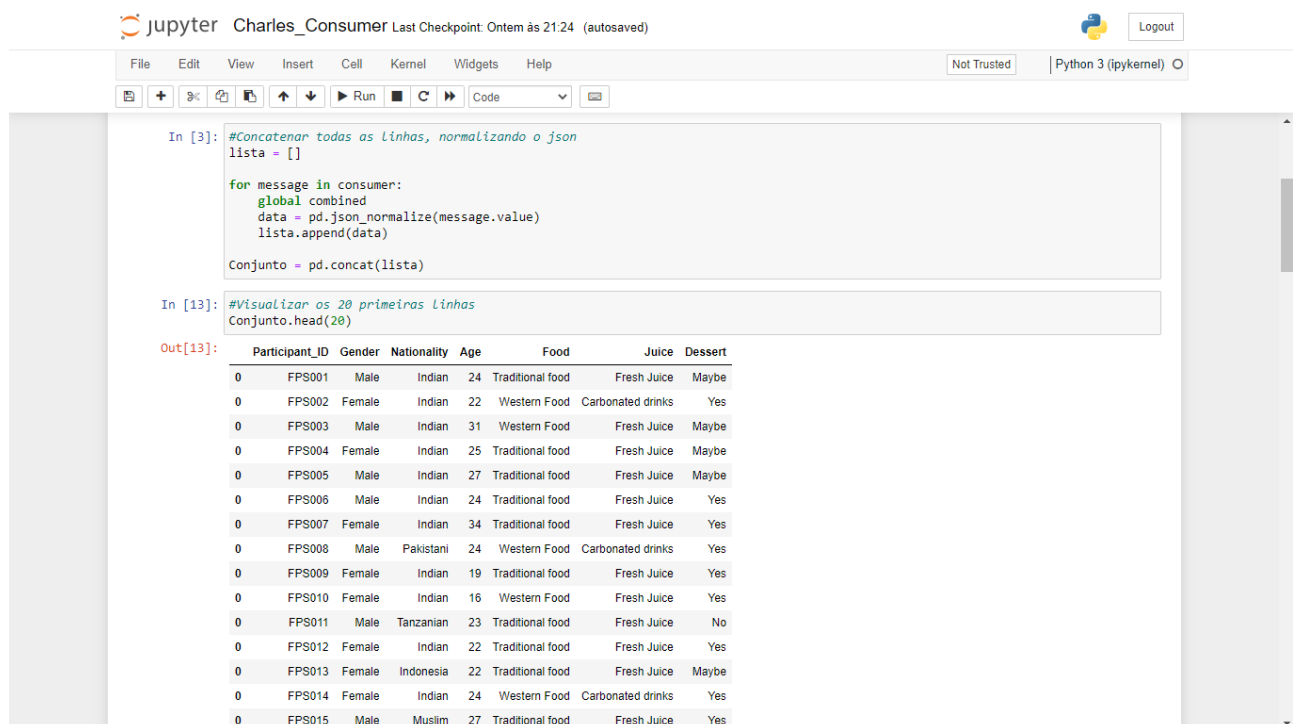
```
        ack = producer.send(topic='Food', key=key, value=json.dumps(row).encode('utf-8'))
```

```
        metadata = ack.get()
```

```
        counter = counter + 1
```

```
    print(metadata.topic, metadata.partition, key)
```

13. Inclua em seu relatório o código fonte necessário para definir e executar um pipeline que implemente, na ordem correta, todos os passos de pré-processamento que você escolheu para analisar sua base de dados.



The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [3]: #Concatenar todas as linhas, normalizando o json
lista = []

for message in consumer:
    global combined
    data = pd.json_normalize(message.value)
    lista.append(data)

Conjunto = pd.concat(lista)


In [13]: #Visualizar os 20 primeiras linhas
Conjunto.head(20)
```

Out[13]:

	Participant_ID	Gender	Nationality	Age	Food	Juice	Dessert
0	FPS001	Male	Indian	24	Traditional food	Fresh Juice	Maybe
0	FPS002	Female	Indian	22	Western Food	Carbonated drinks	Yes
0	FPS003	Male	Indian	31	Western Food	Fresh Juice	Maybe
0	FPS004	Female	Indian	25	Traditional food	Fresh Juice	Maybe
0	FPS005	Male	Indian	27	Traditional food	Fresh Juice	Maybe
0	FPS006	Male	Indian	24	Traditional food	Fresh Juice	Yes
0	FPS007	Female	Indian	34	Traditional food	Fresh Juice	Yes
0	FPS008	Male	Pakistani	24	Western Food	Carbonated drinks	Yes
0	FPS009	Female	Indian	19	Traditional food	Fresh Juice	Yes
0	FPS010	Female	Indian	16	Western Food	Fresh Juice	Yes
0	FPS011	Male	Tanzanian	23	Traditional food	Fresh Juice	No
0	FPS012	Female	Indian	22	Traditional food	Fresh Juice	Yes
0	FPS013	Female	Indonesia	22	Traditional food	Fresh Juice	Maybe
0	FPS014	Female	Indian	24	Western Food	Carbonated drinks	Yes
0	FPS015	Male	Muslim	27	Traditional food	Fresh Juice	Yes

14. Insira em seu relatório um esquema que represente o funcionamento de seu pipeline de dados.

15. Exporte os seus dados processados em formato .csv e importe em um software de visualização. Se possível, você também pode integrar diretamente o Apache Kafka com uma ferramenta de visualização.

 jupyter

QuitLogout

FilesRunningClusters

Select items to perform actions on them.

UploadNew↺

0 / Downloads

NameLast ModifiedFile size

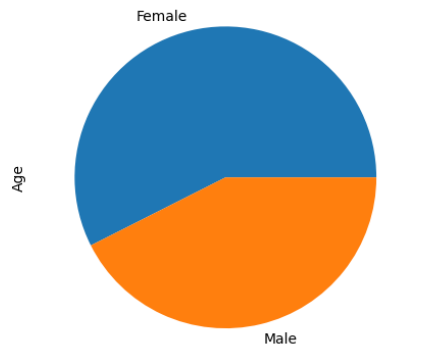
<input type="checkbox"/>	Food_Preference.csv		
<div>UploadCancel</div>			
<input type="checkbox"/>	..	poucos segundos atrás	
<input type="checkbox"/>	Charles_Consumer.ipynb	Running 4 minutos atrás	74.9 kB
<input type="checkbox"/>	Charles_Producer.ipynb	Running uma hora atrás	9.42 kB
<input type="checkbox"/>	Completo_Consumer.ipynb	Running 3 horas atrás	73.2 kB
<input type="checkbox"/>	Completo_Producer.ipynb	3 horas atrás	2.45 kB
<input type="checkbox"/>	2023-02-27 20_01_09 - Infraestrutura Kafka [23E1_2] (Video).mp4	um dia atrás	290 MB
<input type="checkbox"/>	53bb6156bed750a1167cc554e0142fe5.pdf	9 dias atrás	128 kB
<input type="checkbox"/>	Anaconda3-2022.10-Windows-x86_64.exe	9 dias atrás	651 MB
<input type="checkbox"/>	Aula7.mp4	2 dias atrás	285 MB
<input type="checkbox"/>	Aula8.mp4	2 dias atrás	409 MB
<input type="checkbox"/>	Docker Desktop Installer.exe	20 dias atrás	623 MB
<input type="checkbox"/>	Food_Preference.csv	2 horas atrás	16.9 kB
<input type="checkbox"/>	kafka-3.4.0-src.tgz	20 dias atrás	10.6 MB
<input type="checkbox"/>	Kafka_Connect (1).pdf	2 dias atrás	733 kB
<input type="checkbox"/>	Kafka_Connect.pdf	2 dias atrás	733 kB
<input type="checkbox"/>	offsetexplorer_64bit.exe	20 dias atrás	41.2 MB
<input type="checkbox"/>	online_retail_II_short500.csv	2 dias atrás	45.6 kB

Dados em formato csv sendo importados no jupyter

1.1.Utilizando a ferramenta de visualização, crie gráficos (no mínimo dos gráficos, um de barras e um de dispersão) um suportem as suas conclusões com relação às hipóteses investigadas.

```
In [35]: #Gerar um gráfico de pizza
Ticket.plot.pie()
```

```
Out[35]: <AxesSubplot:ylabel='Age'>
```



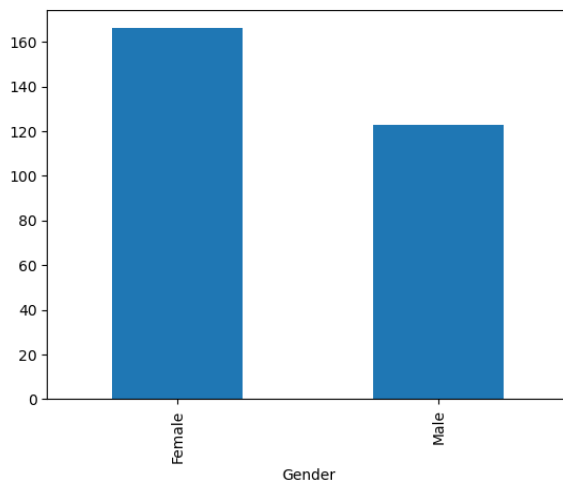
```
In [47]: #Gera um gráfico de linha
summary_quant.plot.line()
```

```
Out[47]: <AxesSubplot:xlabel='Gender'>
```

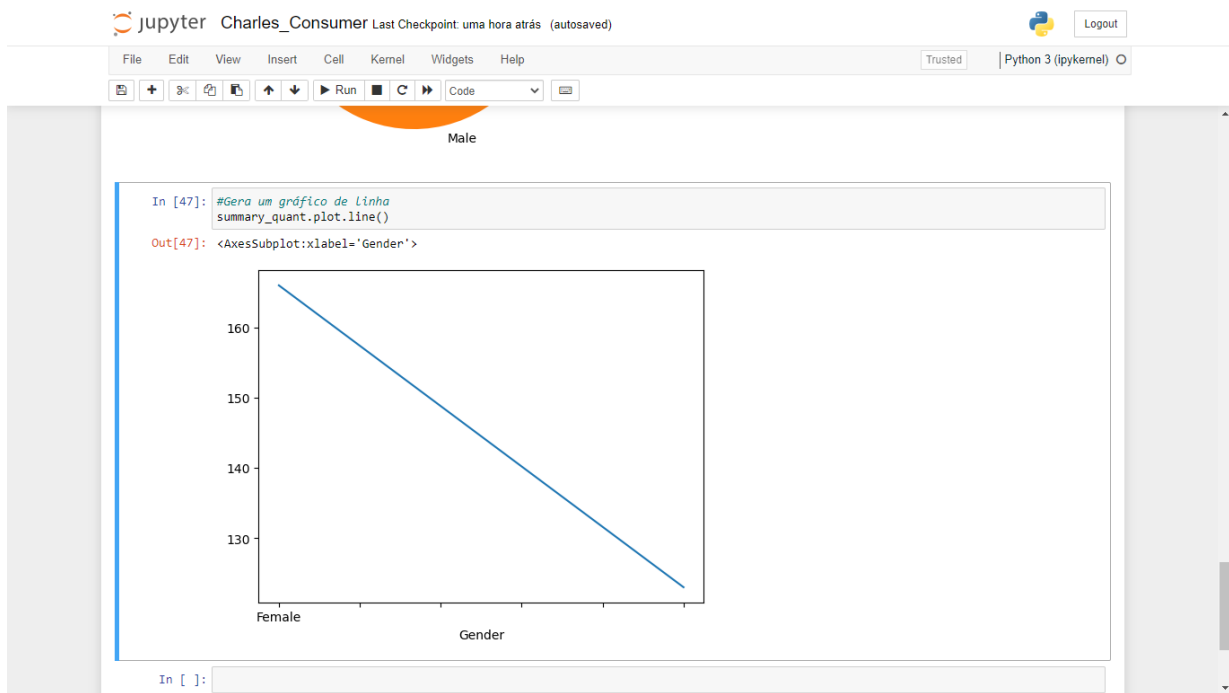
```
In [33]: #Importar a biblioteca matplotlib
import matplotlib.pyplot as plt
```

```
In [34]: #Gerar um gráfico de barras
Ticket.plot.bar()
```

```
Out[34]: <AxesSubplot:xlabel='Gender'>
```



```
In [35]: #Gerar um gráfico de pizza
```



1.2. Por fim, escreva um texto em seu relatório fazendo uma análise final, tendo em vista os resultados obtidos. Responda às perguntas que fez no início do exercício e discuta se sua hipótese foi confirmada ou refutada.

**R:** Estima-se que 61% dos **consumidores** que, por alguma razão, são do sexo feminino como comprovado pelo gráfico através dos dados obtidos com isso confirmamos a hipótese.



### Prints das telas utilizadas no jupyter:

```
#Importando as bibliotecas
from kafka import KafkaConsumer
import json
from json import loads
import pandas as pd
from pandas.io.json import json_normalize
```

In [2]:

```
#Consuma todas as mensagens do tópico, mas não marque como 'lidas'
(enable_auto_commit=False)
#para que possamos relê-los quantas vezes quisermos.
consumer = KafkaConsumer('Food',
group_id = 'food-consumer-group',
bootstrap_servers=['localhost:9092'],
value_deserializer=lambda m: json.loads(m.decode('utf-8')),
auto_offset_reset = 'earliest',
enable_auto_commit=False,
consumer_timeout_ms = 1000)
```

In [3]:

```
#Concatenar todas as linhas, normalizando o json
lista = []
```

```
for message in consumer:
    global combined
    data = pd.json_normalize(message.value)
    lista.append(data)
```

```
Conjunto = pd.concat(lista)
```

In [13]:

```
#Visualizar os 20 primeiras linhas
Conjunto.head(20)
```

Out[13]:

	Participant_ID	Gender	Nationality	Age	Food	Juice	Dessert
0	FPS001	Male	Indian	24	Traditional food	Fresh Juice	Maybe
0	FPS002	Female	Indian	22	Western Food	Carbonated drinks	Yes
0	FPS003	Male	Indian	31	Western Food	Fresh Juice	Maybe
0	FPS004	Female	Indian	25	Traditional food	Fresh Juice	Maybe
0	FPS005	Male	Indian	27	Traditional food	Fresh Juice	Maybe
0	FPS006	Male	Indian	24	Traditional food	Fresh Juice	Yes
0	FPS007	Female	Indian	34	Traditional food	Fresh Juice	Yes
0	FPS008	Male	Pakistani	24	Western Food	Carbonated drinks	Yes
0	FPS009	Female	Indian	19	Traditional food	Fresh Juice	Yes
0	FPS010	Female	Indian	16	Western Food	Fresh Juice	Yes
0	FPS011	Male	Tanzanian	23	Traditional food	Fresh Juice	No
0	FPS012	Female	Indian	22	Traditional food	Fresh Juice	Yes
0	FPS013	Female	Indonesia	22	Traditional food	Fresh Juice	Maybe

	Participant_ID	Gender	Nationality	Age	Food	Juice	Dessert
0	FPS014	Female	Indian	24	Western Food	Carbonated drinks	Yes
0	FPS015	Male	Muslim	27	Traditional food	Fresh Juice	Yes
0	FPS016	Male	Pakistan	25	Traditional food	Fresh Juice	Maybe
0	FPS017	Male	Maldivian	26	Traditional food	Fresh Juice	No
0	FPS018	Male	MY	22	Traditional food	Fresh Juice	No
0	FPS019	Female	Malaysian	38	Traditional food	Fresh Juice	Maybe
0	FPS020	Female	Indian	31	Traditional food	Fresh Juice	Maybe

In [5]:

```
#Obter informações do Dataframe
Conjunto.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 289 entries, 0 to 0
Data columns (total 7 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Participant_ID        289 non-null    object
1   Gender                289 non-null    object
2   Nationality           289 non-null    object
3   Age                  289 non-null    object
4   Food                 289 non-null    object
5   Juice                289 non-null    object
6   Dessert              289 non-null    object
dtypes: object(7)
memory usage: 18.1+ KB
```

In [14]:

```
#Converter as colunas Age e Nationality
Conjunto['Age'] = pd.to_numeric(Conjunto['Age'])
```

In [16]:

```
#Obter informações do Dataframe
Conjunto.head(10)
```

Out[16]:

	Participant_ID	Gender	Nationality	Age	Food	Juice	Dessert
0	FPS001	Male	Indian	24	Traditional food	Fresh Juice	Maybe
0	FPS002	Female	Indian	22	Western Food	Carbonated drinks	Yes
0	FPS003	Male	Indian	31	Western Food	Fresh Juice	Maybe
0	FPS004	Female	Indian	25	Traditional food	Fresh Juice	Maybe
0	FPS005	Male	Indian	27	Traditional food	Fresh Juice	Maybe
0	FPS006	Male	Indian	24	Traditional food	Fresh Juice	Yes
0	FPS007	Female	Indian	34	Traditional food	Fresh Juice	Yes
0	FPS008	Male	Pakistani	24	Western Food	Carbonated drinks	Yes
0	FPS009	Female	Indian	19	Traditional food	Fresh Juice	Yes
0	FPS010	Female	Indian	16	Western Food	Fresh Juice	Yes

In [31]:

```
#Soma das quantidades por nacionalidade
```

```
summary_quant = Conjunto.groupby('Gender')['Age'].count()
print(summary_quant)
```

```
Gender
Female    166
Male      123
Name: Age, dtype: int64
```

In [32]:

```
Ticket = summary_quant
print(Ticket)
```

```
Gender
Female    166
Male      123
Name: Age, dtype: int64
```

In [33]:

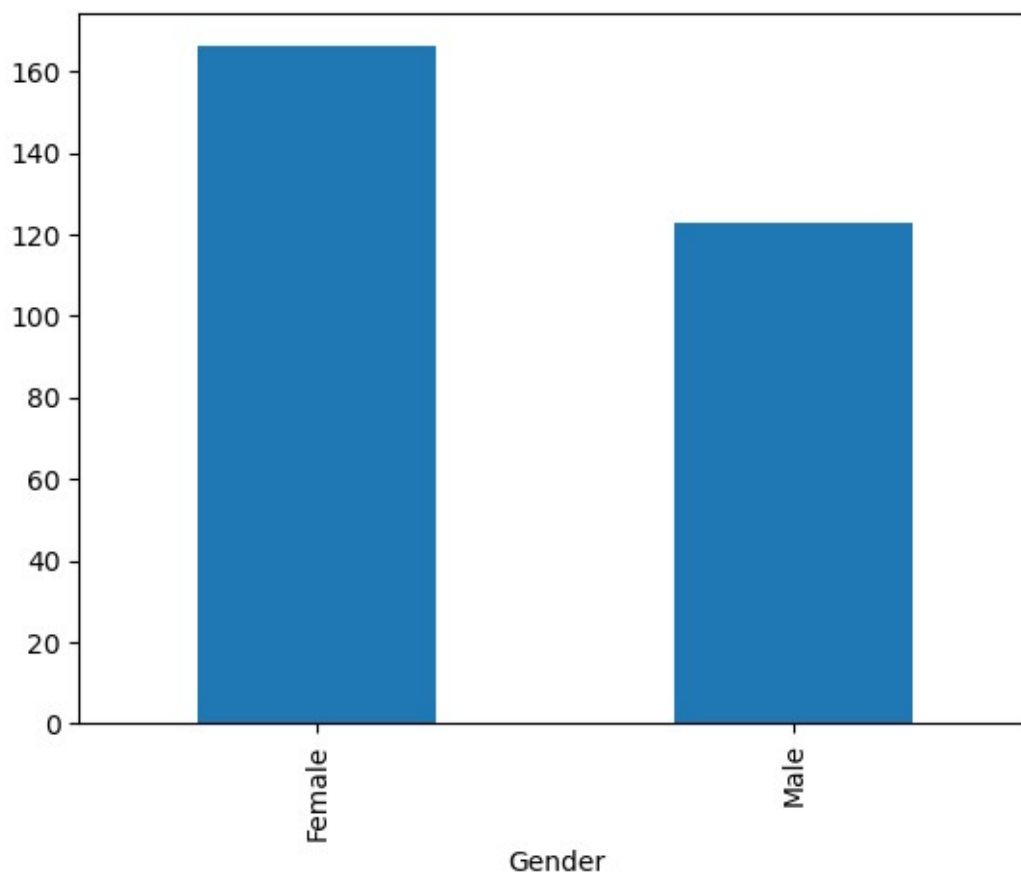
```
#Importar a biblioteca matplotlib
import matplotlib.pyplot as plt
```

In [34]:

```
#Gerar um gráfico de barras
Ticket.plot.bar()
```

Out[34]:

<AxesSubplot:xlabel='Gender'>

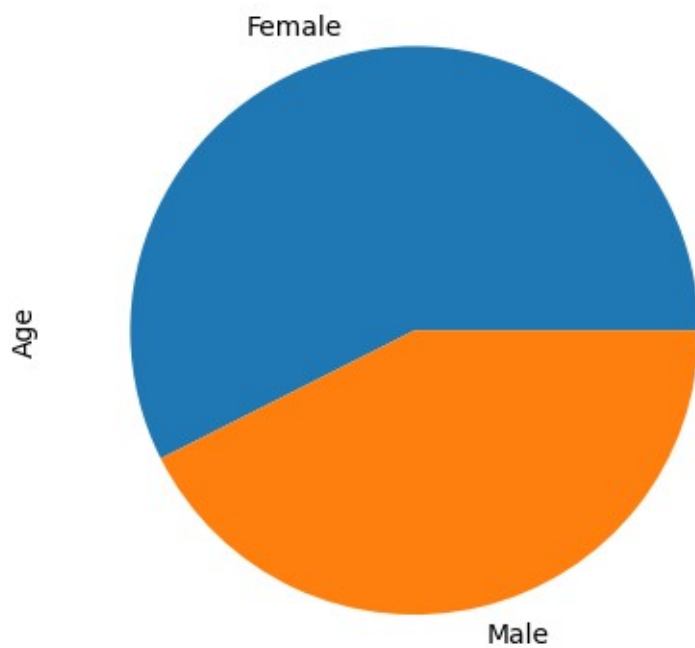


In [35]:

```
#Gerar um gráfico de pizza  
Ticket.plot.pie()
```

Out[35]:

<AxesSubplot:ylabel='Age'>

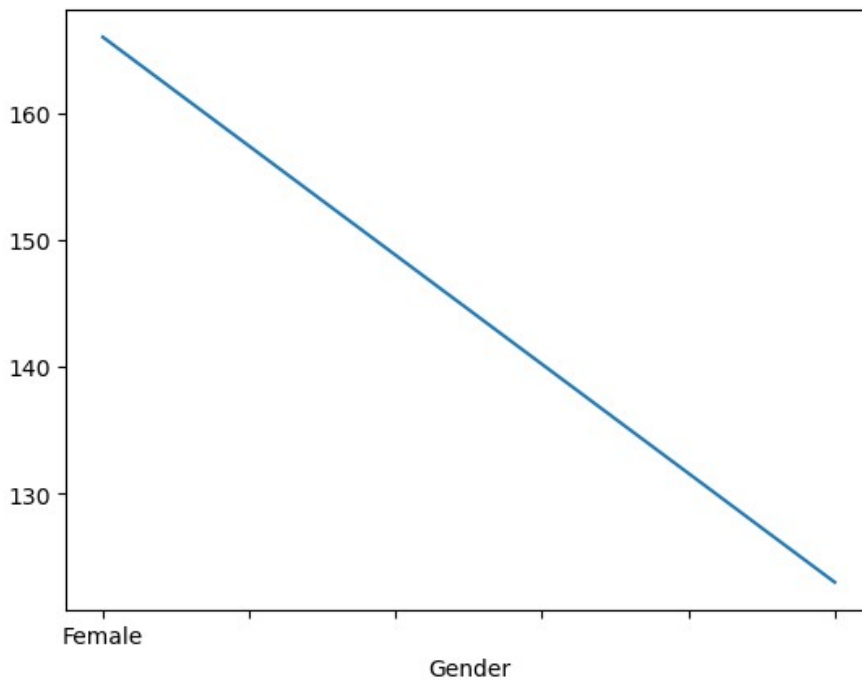


In [47]:

```
#Gera um gráfico de linha  
summary_quant.plot.line()
```

Out[47]:

<AxesSubplot:xlabel='Gender'>



```
#Importa as bibliotecas
import pandas as pd
import json
from csv import DictReader
import datetime as dt
from kafka import KafkaProducer
```

In [2]:

```
#Inicializar o cliente Kafka Producer
producer = KafkaProducer(bootstrap_servers=['localhost:9092'])

print(f'Iniciando o Kafka producer at {dt.datetime.utcnow()}')
```

Iniciando o Kafka producer at 2023-03-08 23:16:53.126909

In [3]:

```
#Configurar um contador e definir o caminho do arquivo
counter = 0
file = 'Food_Preference.csv'
```

In [4]:

```
with open(file, 'r') as new_obj:
    csv_dict_reader = DictReader(new_obj)
    for row in csv_dict_reader:
        key = str(counter).encode()
        ack = producer.send(topic='Food', key=key,
value=json.dumps(row).encode('utf-8'))
        metadata = ack.get()
        counter = counter + 1
        print(metadata.topic, metadata.partition, key)
```

```
producer = kafka_producer({bootstrap_servers=['localhost:9092']})
```

```
print(f'Iniciando o Kafka producer at {dt.datetime.utcnow()}')
```

Iniciando o Kafka producer at 2023-03-08 23:16:53.126909

```
In [3]: #Configurar um contador e definir o caminho do arquivo
counter = 0
file = 'Food_Preference.csv'
```

```
In [4]: with open(file, 'r') as new_obj:
        csv_dict_reader = DictReader(new_obj)
        for row in csv_dict_reader:
            key = str(counter).encode()
            ack = producer.send(topic='Food', key=key, value=json.dumps(row).encode('utf-8'))
            metadata = ack.get()
            counter = counter + 1
            print(metadata.topic, metadata.partition, key)
```

```
Food 0 b'270'
Food 0 b'271'
Food 0 b'272'
Food 0 b'273'
Food 0 b'274'
Food 0 b'275'
Food 0 b'276'
Food 0 b'277'
Food 0 b'278'
Food 0 b'279'
Food 0 b'280'
Food 0 b'281'
Food 0 b'282'
Food 0 b'283'
Food 0 b'284'
Food 0 b'285'
Food 0 b'286'
Food 0 b'287'
Food 0 b'288'
```