

Centro Universitário
Christus- UNICHRISTUS

Especialização em
Ciência de Dados e
Inteligência de Negócios
(Big Data e BI)

Prof. Dr. Manoel Ribeiro

Processamento de Dados em Tempo Real (Streaming)

Conteúdo da disciplina

- Dia 1 (sexta 18:00 às 22:00h)
 - Apresentação
 - Aula motivacional - Qual a importância do processamento de dados em tempo real?
- Dia 2 (sábado 8:00 às 18:00)
 - Fundamentos de sistema de processamento de tempo real
 - Fundamentos da arquitetura Apache Kafka, Apache ZooKeeper
 - Prática com Apache Kafka (tarde)

Conteúdo da disciplina

- Dia 3 (sexta 18:00 às 22:00h)
 - Fundamentos Apache Flume
 - Fundamentos Spark Streaming
 - Configuração Hadoop e Spark
- Dia 4 (sábado 8:00 às 18:00)
 - Fundamentos Introdução ao Apache Storm
 - Implementação de projeto prático/aplicado envolvendo Real Time Analytics
 - Avaliação

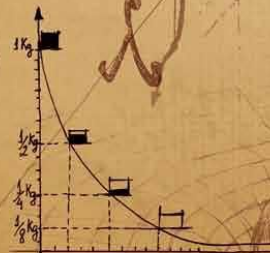
Repositório

<https://github.com/antoniomralmeida/streaming>

Fundamentação

$$\Delta x = v t$$
$$\Delta x = v_0 t + \frac{a t^2}{2}$$
$$v = v_0 + a t$$
$$v^2 = v_0^2 + 2 a \Delta x$$

$$\nabla \cdot \vec{E} = \frac{1}{\epsilon_0} \rho$$
$$\nabla \cdot \vec{B} = 0$$
$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$
$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t}$$



$$v = \frac{\Delta s}{\Delta t} = \frac{s - s_0}{t}$$



$$v_m = \frac{v + v_0}{2}$$

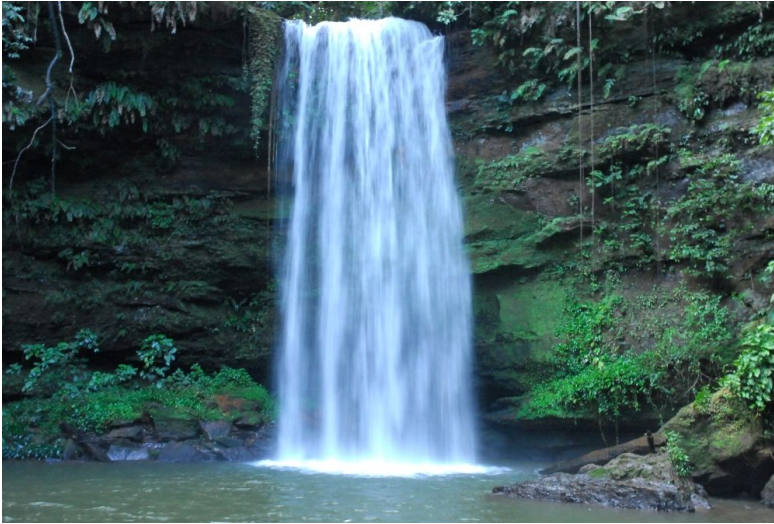
$$h = \frac{v^2 - v_0^2}{2g}$$

$$r = r_x + r_y + r_z$$
$$r = (r_x, r_y)$$
$$z = f_x x + f_y y + f_z z$$
$$g = f_x x + f_y y + f_z z$$
$$r = \sqrt{r_x^2 + r_y^2}$$
$$g = \frac{r_y}{r_x}$$

Streaming

Streaming

- Fluxo contínuo (contínuo \neq constante).



Streaming de dados

- Fluxo contínuo de dados.



Streaming de dados: Exemplos

- Sensores (IoT)
- Tráfego de rede
- Registros de call center
- Tendências em redes sociais
- Serviços de áudio e vídeo
- Análise de log
- Estatísticas de sites web



Tipos de streaming de dados

Dados de texto: web, log

Dados relacionais: tabelas, transações

Dados semi-estruturados: XML, json

Dados em grafo: redes sociais

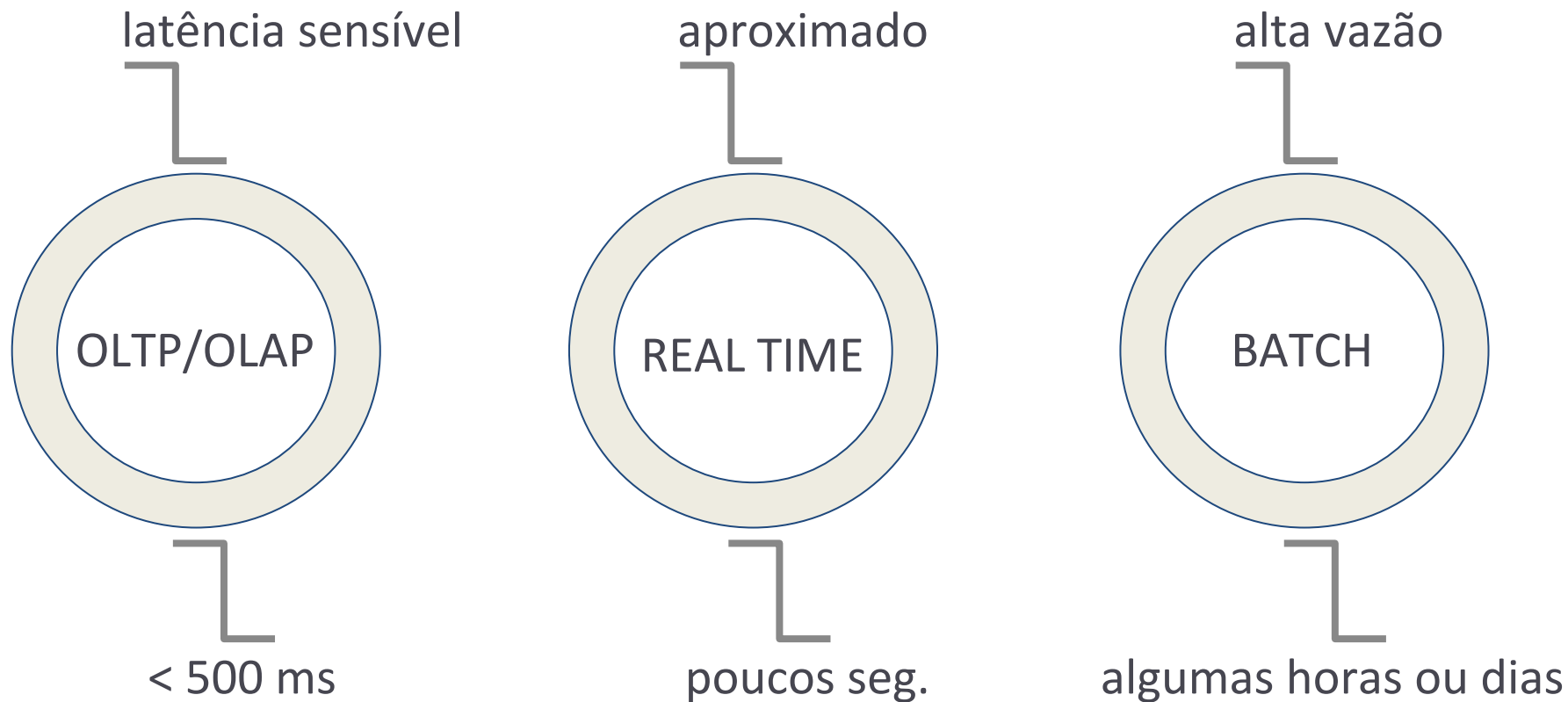
Dados de mobilidade: coordenadas geográficas x tempo

Etc.

O que é tempo real?

Milissegundos, segundos, minutos?

O que é Tempo Real?



O que é Tempo Real?

REAL TIME TRENDS



Emerging break out trends in Twitter (in the form #hashtags)

REAL TIME CONVERSATIONS



Real time sports conversations related with a topic (recent goal or touchdown)

REAL TIME RECOMMENDATIONS



Real time product recommendations based on your behavior & profile

REAL TIME SEARCH



Real time search of tweets with a budget < 200 ms

Problemas em streaming

1. Como **obter** os dados a partir de várias fontes em um cluster em tempo real?
2. Como **processar** esses dados?



Apache Kafka

Apache Kafka - o que é

- *O que é o Apache Kafka?*
- *“Apache Kafka é uma plataforma distribuída de mensagens e streaming”.*

Apache Kafka

- Você **produz** uma mensagem.
- Essa mensagem é **anexada** em um tópico.
- Você então **consome** essa mensagem.
-

Por que usar?

- *“Se você quer mover e transformar um grande volume de dados em tempo real entre diferentes sistemas, então Apache Kafka pode ser exatamente o que você precisa”.*

Por que usar?

- message broke,

Apache Kafka

- Sistema de mensagens
 - Distribuído
 - Com alta vazão (*throughput*)
 - De geração (publicação) e leitura (sub-inscrição)
- Principais casos de uso:
 - Agregação de log
 - Processamento em tempo real
 - Monitoramento

Apache Kafka

- Originalmente desenvolvido pelo LinkedIn.
- Implementado em scala/Java.
- *Producers & Consumers.*
- Mensagens são associadas a tópicos, os quais representam um stream específico.
 - Logs web
 - Dados de sensores
- *Consumers* se inscrevem em um ou mais tópicos.

Kafka: conceitos

Producers



Push

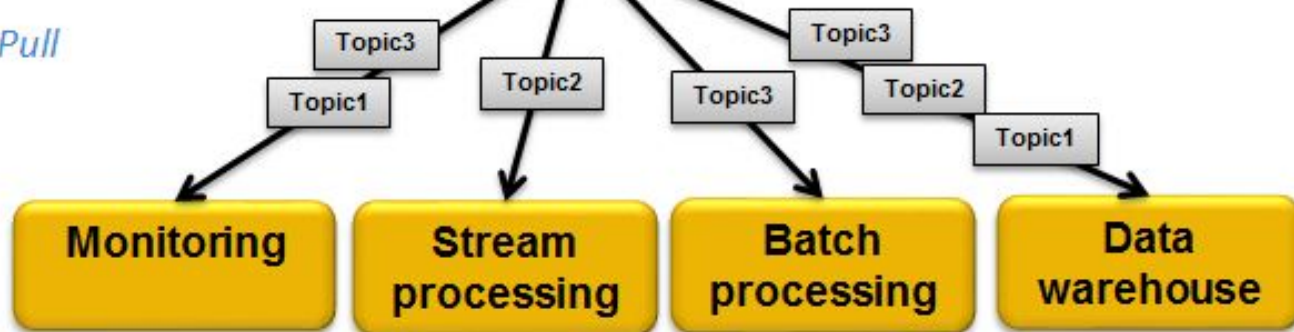
Broker



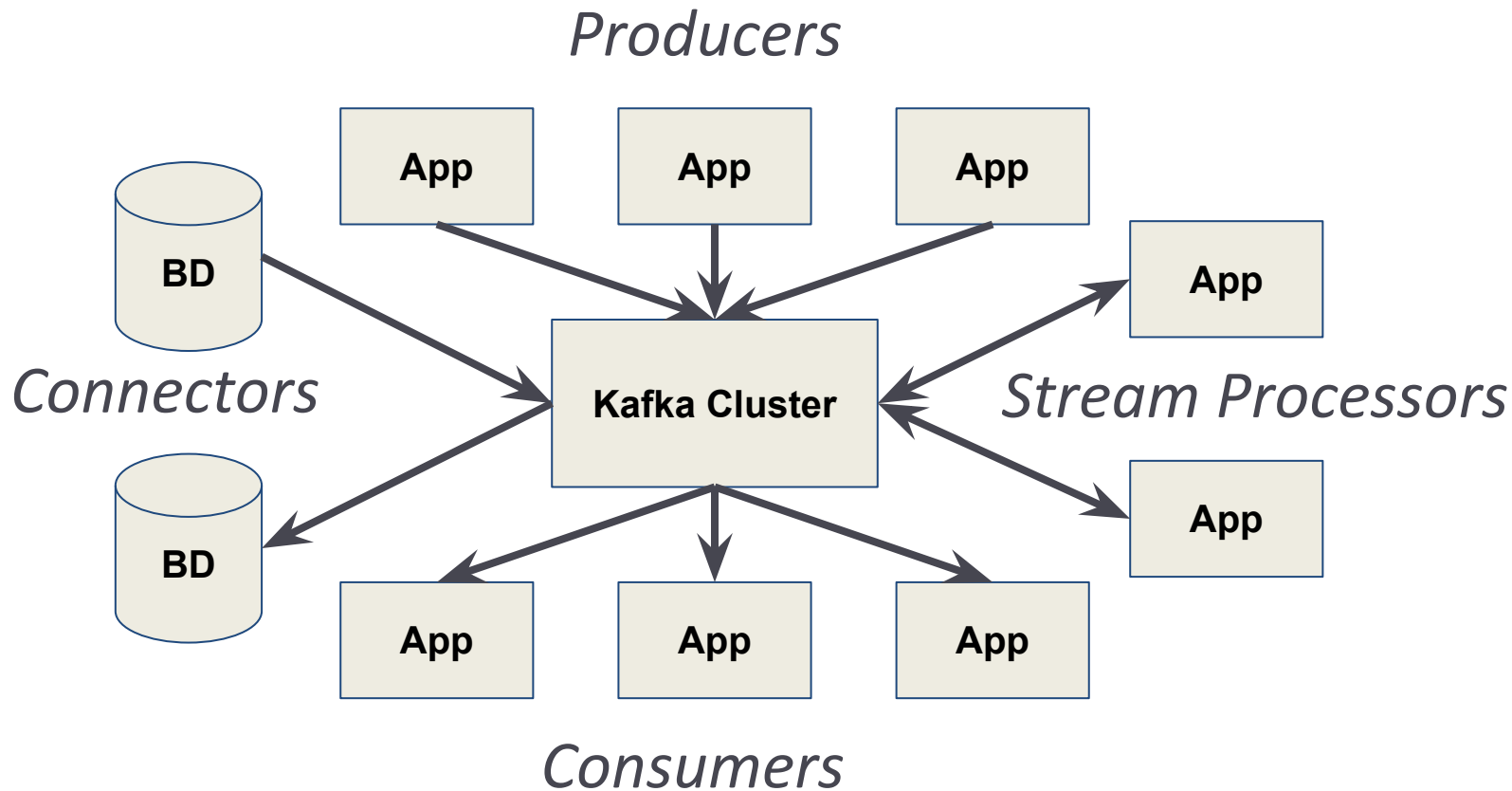
Zookeeper

Pull

Consumers

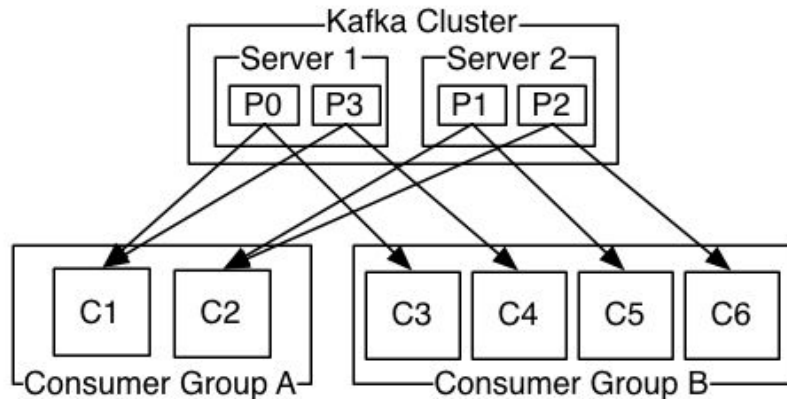


Kafka: arquitetura



Kafka: escalabilidade

- Kafka pode ser distribuído entre muitos processos em vários servidores.
- *Consumers* também podem ser distribuídos.
- Tolerante a falhas.



Fonte: <https://kafka.apache.org/intro.html>

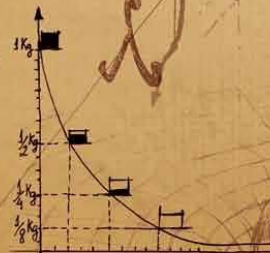
Kafka: pontos a considerar

- Simples sistema de mensagens, não de processamento.
- Não vive sem o **Zookeeper**, o qual pode se tornar um gargalo quando o número de tópicos/partições é muito grande (>>10000).
- Não otimizado para latências de milissegundos.

Prática

$$\Delta x = v_0 t$$
$$\Delta x = v_0 t + \frac{at^2}{2}$$
$$v = v_0 + at$$
$$v^2 = v_0^2 + 2a \Delta x$$

$$\nabla \cdot \vec{E} = \frac{1}{\epsilon_0} \rho$$
$$\nabla \cdot \vec{B} = 0$$
$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$
$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t}$$



$$v = \frac{\Delta s}{\Delta t} = \frac{s - s_0}{t}$$



$$v_m = \frac{v + v_0}{2}$$

$$h = \frac{v^2 - v_0^2}{2g}$$

$$r = r_1 + r_2 + r_3$$

$$r = (x, y)$$

$$z = f_1 x + f_2 x + f_3 x$$

$$y = f_1 y + f_2 y + f_3 y$$

$$r = \sqrt{r_x^2 + r_y^2}$$

$$r_{gx} = \frac{r_x}{r_z}$$

$$r_{gy} = \frac{r_y}{r_z}$$

Prática dia 2

- Instalar Oracle VirtualBox
- Criar instância para Linux Ubuntu
- Instalar Ubuntu
- Instalar Apache Kafka
 - Pré-requisito Zookeeper
- Tarefa 1 - Hello World Kafka
 - Criar tópico
 - Gerar dados no tópico
 - Consumir dados
- Tarefa 2 - produtor / consumidor em Python
 - instalar o python
 - instalar biblioteca kafka-python
 - executar código producer_consumer.py


Criar nova máquina virtual

Criar Máquina Virtual

Nome e Sistema Operacional

Escolha um nome descritivo para a nova máquina virtual e selecione o tipo de sistema operacional que você pretende instalar nela. O nome que você escolher será utilizado pelo VirtualBox para identificar esta máquina.

Nome:

Tipo: 


Versão:

Criar Máquina Virtual

Tamanho da memória

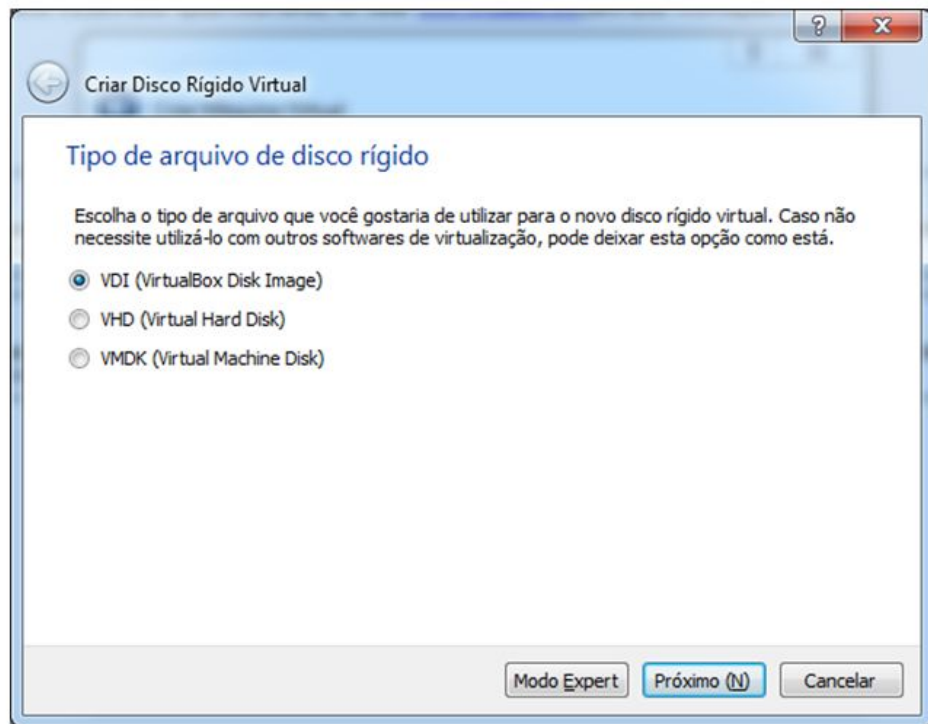
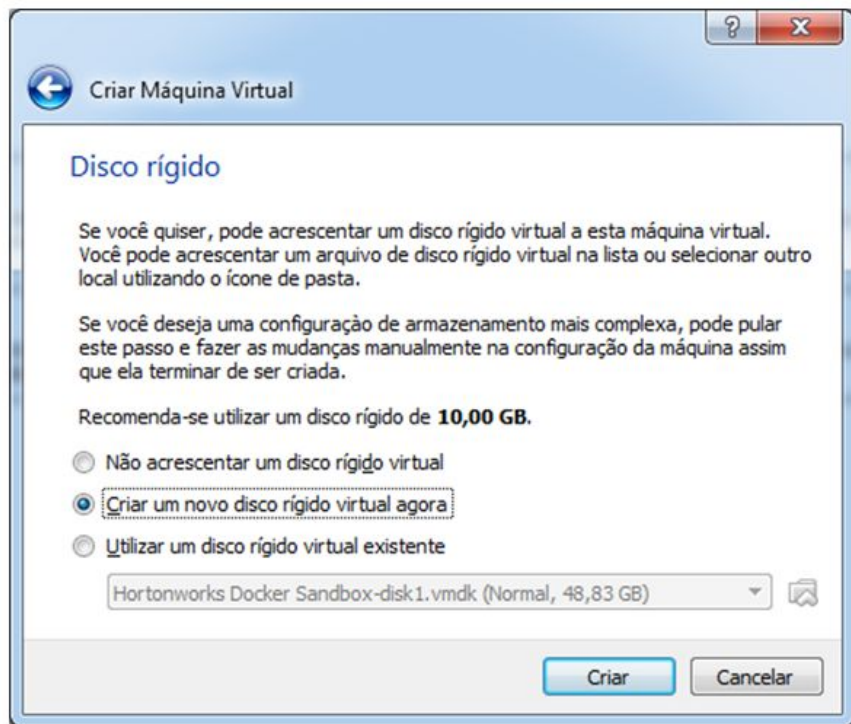
Selecione a quantidade de memória (RAM) em megabytes que será alocado para a máquina virtual.

O tamanho recomendado para memória é de **1024MB**.

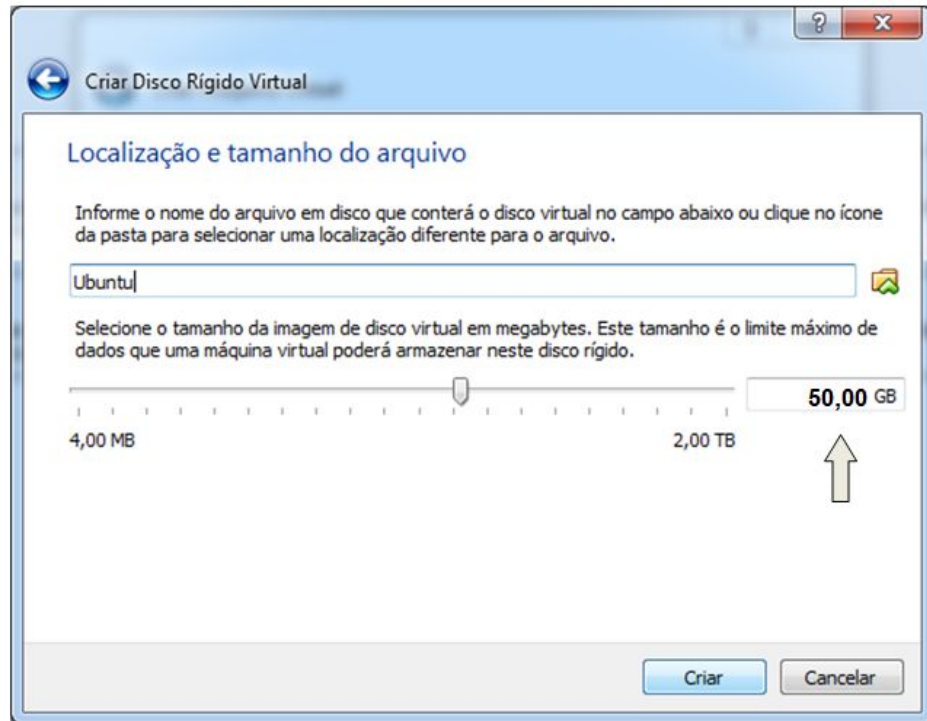
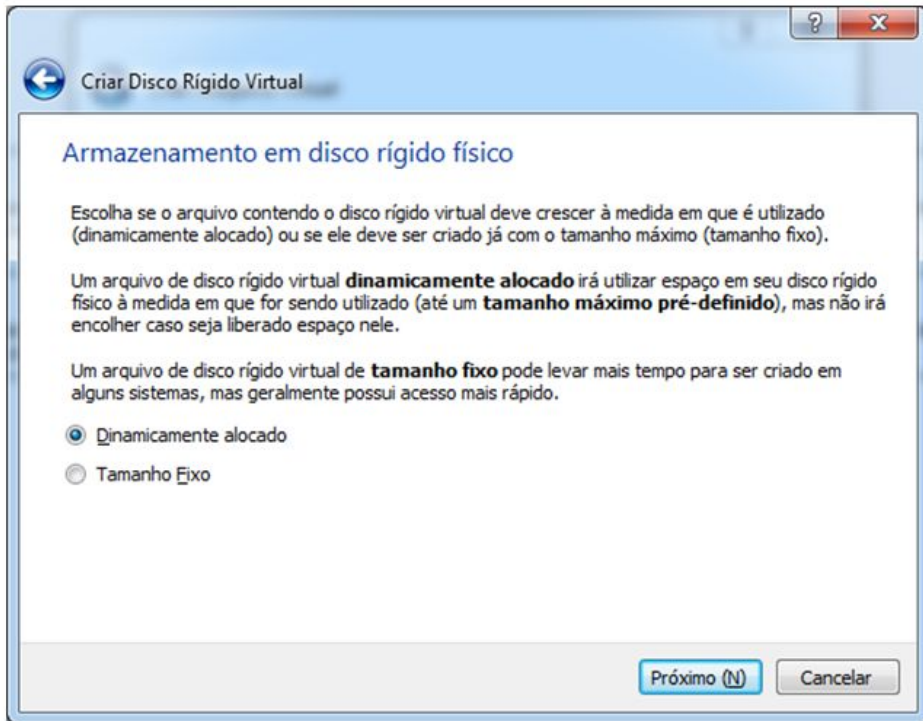
 4096 MB

4 MB 8192 MB

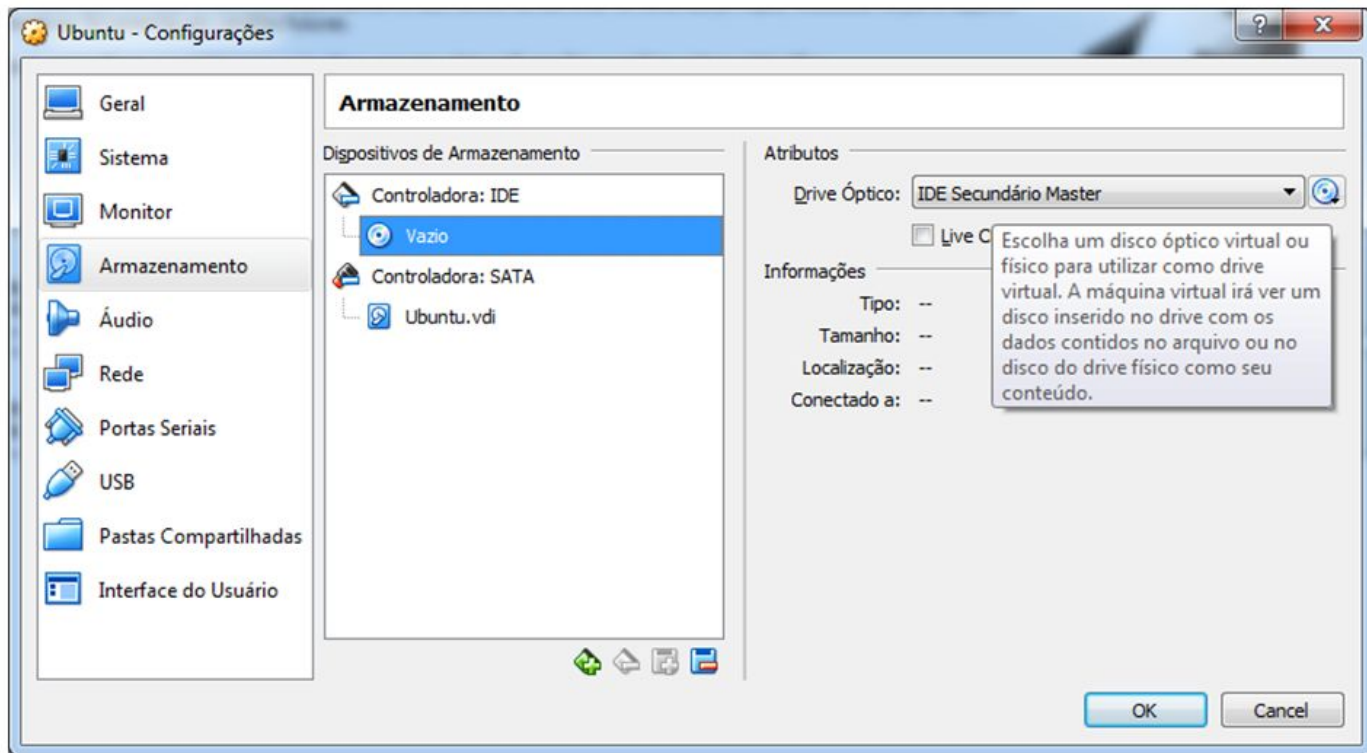
Criar nova máquina virtual



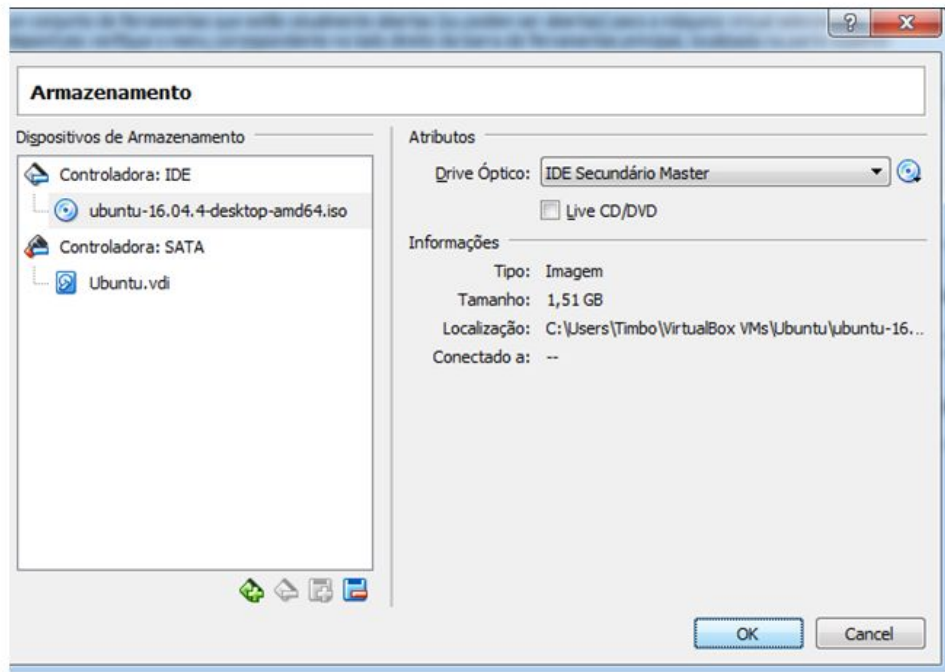
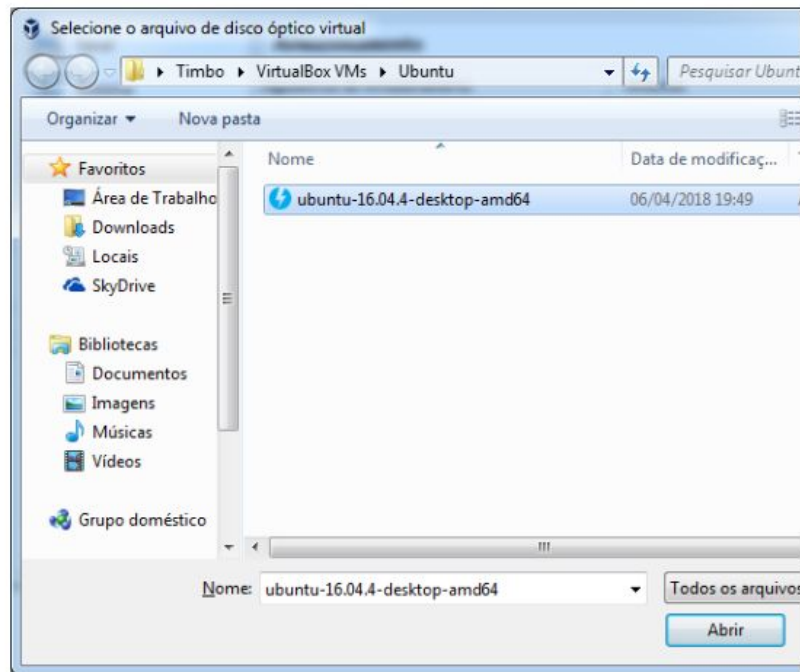
Criar nova máquina virtual



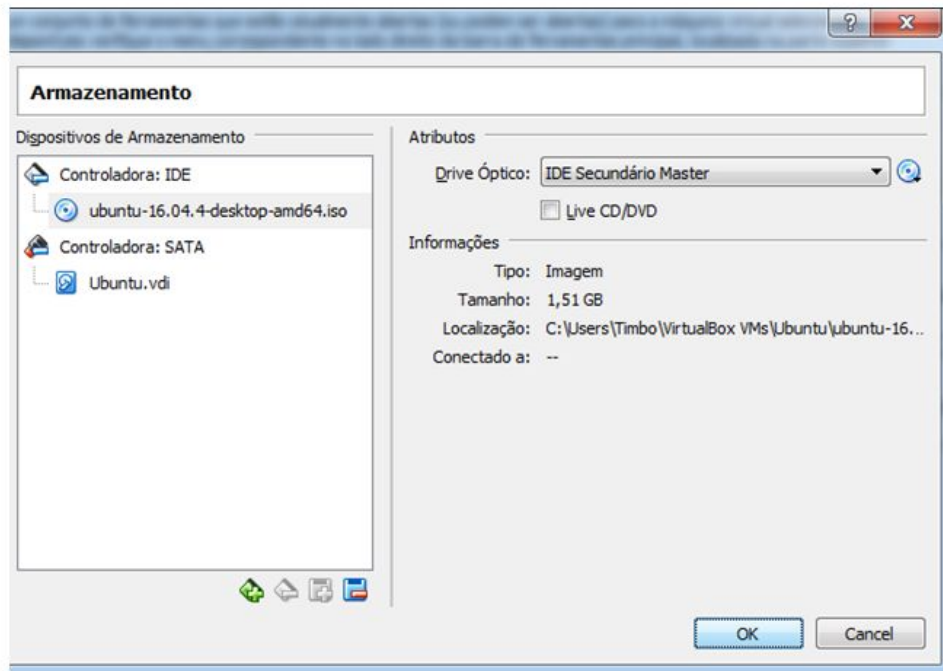
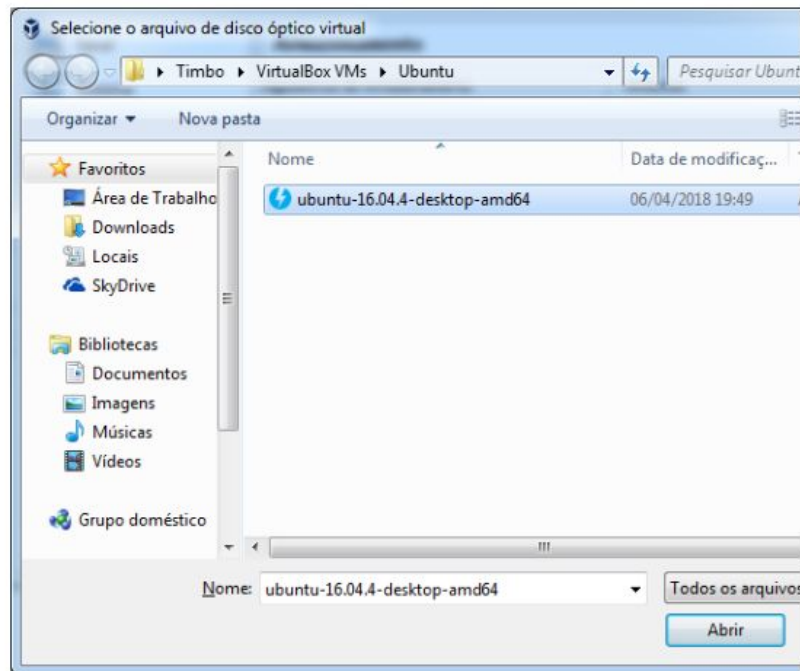
Configurações -> Armazenamento



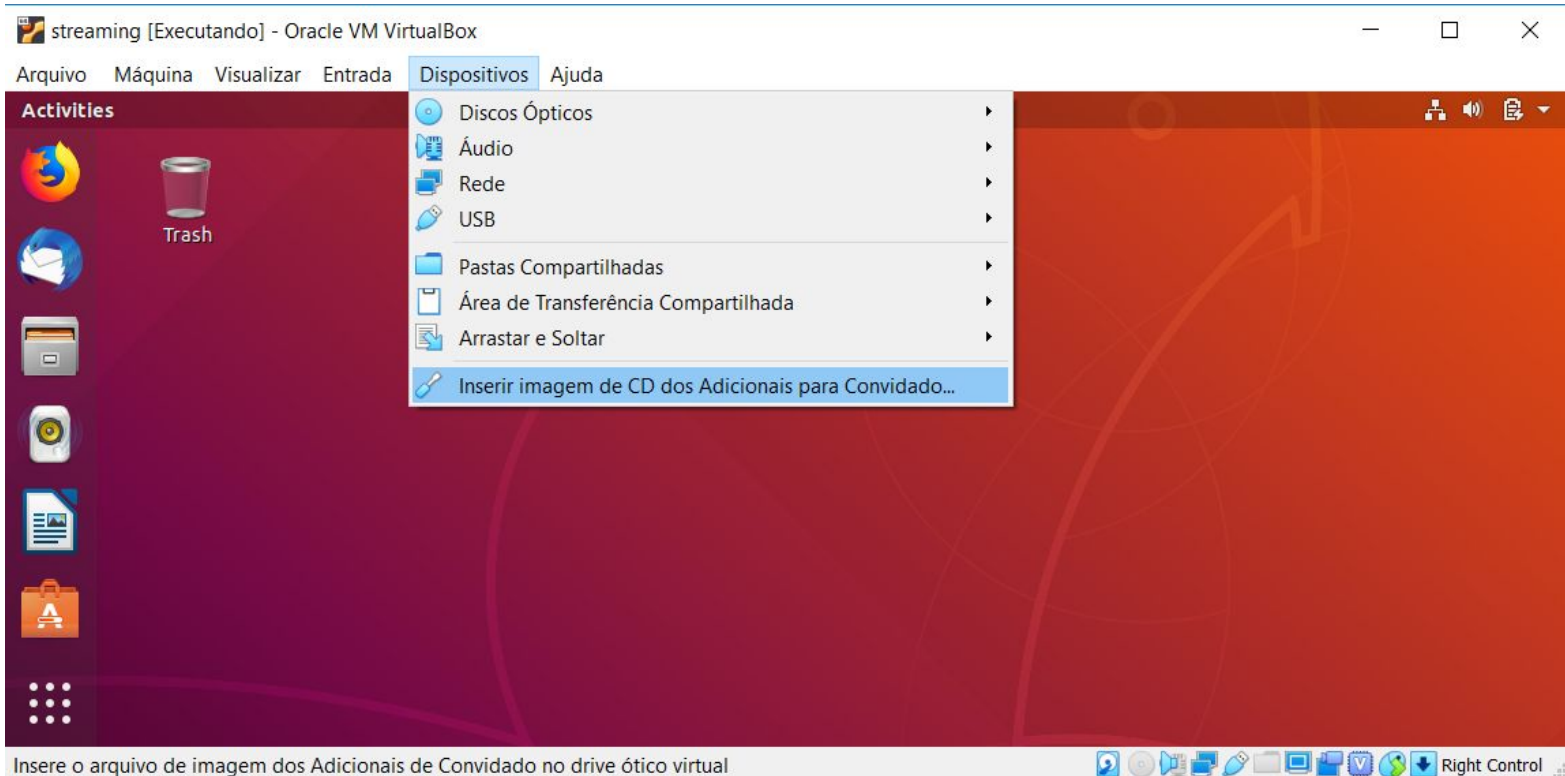
Selecionar arquivo de disco óptico virtual



Selecionar arquivo de disco óptico virtual



Configurar tela cheia



KAFKA do ZERO

Primeiros passos

1. Acessar máquina virtual
2. Acessar terminal (gnome-terminal)
3. Atualizar Ubuntu 16.04
 - `sudo apt-get update -y`
 - `sudo apt-get upgrade -y`
4. Esperar alguns minutos

Instalar o Java

```
sudo add-apt-repository -y ppa:webupd8team/java
```

```
gpg: keyring `/tmp/tmpkjr4mnm/secring.gpg' created  
gpg: keyring `/tmp/tmpkjr4mnm/pubring.gpg' created  
gpg: requesting key EEA14886 from hkp server keyserver.ubuntu.com  
gpg: /tmp/tmpkjr4mnm/trustdb.gpg: trustdb created  
gpg: key EEA14886: public key "Launchpad VLC" imported  
gpg: no ultimately trusted keys found  
gpg: Total number processed: 1  
gpg:         imported: 1 (RSA: 1)  
OK
```

```
sudo apt-get update
```

```
sudo apt-get install oracle-java8-installer -y
```

```
sudo java -version
```

```
java version "1.8.0_66"  
Java(TM) SE Runtime Environment (build 1.8.0_66-b17)  
Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)
```

Instalar o ZooKeeper

- `sudo apt-get install zookeeperd`

Para testar a instalação:

- `netstat -ant | grep :2181`

```
tcp6      0      0 :::2181          :::*              LISTEN
```

Instalar o Kafka

Baixar o kafka:

- `wget https://archive.apache.org/dist/kafka/0.10.0.1/kafka_2.10-0.10.0.1.tgz`

Criar um diretório para a instalação do kafka:

- `sudo mkdir kafka`

Extrair os dados para o diretório:

- `sudo tar -xvf kafka_2.10-0.10.0.1.tgz -C kafka/`

Inicializar o Servidor Kafka

- o `cd kafka/kafka_2.10-0.10.0.1/bin`
- o `sudo ./kafka-server-start.sh ~/kafka/kafka_2.10-0.10.0.1/config/server.properties`

```
[2016-08-22 21:43:48,279] WARN No meta.properties file under dir /tmp/kafka-logs/meta.properties  
(kafka.server.BrokerMetadataCheckpoint)
```

```
[2016-08-22 21:43:48,516] INFO Kafka version : 0.10.0.1 (org.apache.kafka.common.utils.AppInfoParser)
```

```
[2016-08-22 21:43:48,525] INFO Kafka commitId : a7a17cdec9eaa6c5 (org.apache.kafka.common.utils.AppInfoParser)
```

```
[2016-08-22 21:43:48,527] INFO [Kafka Server 0], started (kafka.server.KafkaServer)
```

```
[2016-08-22 21:43:48,555] INFO New leader is 0 (kafka.server.ZookeeperLeaderElector$LeaderChangeListener)
```


Criar novo tópico

7. Criar e listar um novo tópico

- `./kafka-topics.sh --create`
 `--zookeeper localhost:2181`
 `--replication-factor 1 --partitions 1`
 `--topic any`
- `./kafka-topics.sh --list --zookeeper`
 `localhost:2181`

Gerar dados de um tópico

8. Executar o *producer*

- `./kafka-console-producer.sh --broker-list localhost:6667 --topic any`

Consumir dados de um tópico

9. Inicializar outra janela de terminal

10. Acessar diretório KAFKA

- `cd kafka/kafka_2.10-0.10.0.1/bin`

11. Obter os dados de um determinado tópico

```
./kafka-console-consumer.sh --zookeeper  
localhost:2181 --bootstrap-server localhost:9092  
--topic TutorialTopic --from-beginning
```

Produtor x Consumidor de dados

```

maria_dev@sandbox:/usr/hdp/current/kafka-broker/bin
Using username "maria_dev".
maria_dev@127.0.0.1's password:
Last login: Mon Feb 13 19:31:16 2017 from 10.0.2.2
[maria_dev@sandbox ~]$ cd /usr/hdp/current/kafka-broker/bin
[maria_dev@sandbox bin]$ ./kafka-console-consumer.sh --bootstrap-server sandbox.hortonworks.com:6667 --zookeeper localhost:2181 --topic fred --from-beginning
{metadata.broker.list=sandbox.hortonworks.com:6667, request.timeout.ms=30000,
ient.id=console-consumer-89447, security.protocol=PLAINTEXT}
This is a line of data
I am sending this on the fred topic
Here is yet another line
█

kafka-consumer-perf-test.sh      kafka-zookeeper-run-class.sh
kafka-mirror-maker.sh           windows
kafka-preferred-replica-election.sh  zookeeper-security-migration.sh
kafka-producer-perf-test.sh        zookeeper-server-start.sh
kafka-reassign-partitions.sh       zookeeper-server-stop.sh
kafka-replay-log-producer.sh       zookeeper-shell.sh
[maria_dev@sandbox bin]$ ./kafka-topics.sh --create --zookeeper sandbox.hortonwo
rks.com:2181 --replication-factor 1 --partitions 1 --topic fred
Created topic "fred".
[maria_dev@sandbox bin]$ ./kafka-topics.sh --list --zookeeper sandbox.hortonwork
s.com:2181
ATLAS_ENTITIES
ATLAS_HOOK
  _consumer_offsets
fred
log-topic
log-topic2
test - marked for deletion
[maria_dev@sandbox bin]$ ./kafka-console-producer.sh --broker-list sandbox.horto
nworks.com:6667 --topic fred
This is a line of data
I am sending this on the fred topic
Here is yet another line
█
```

Para sair: `ctrl+C`

Atividade 1



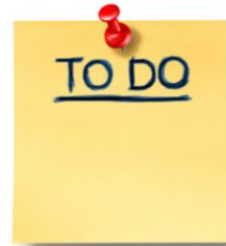
1. Crie um tópico com o nome de vocês;
2. Liste os tópicos e verifique se o seu foi criado;
3. Gere dados para o tópico criado.

Obs.:

Endereço do zookeeper: `localhost:2181`

broker-list: `localhost:9092`

Atividade 2 (vale pt)



1. Imprima apenas o conteúdo da tupla;
2. Gere dados de quatro producers simultâneos;
3. Aumente a frequência de geração das tuplas (geração mais rápida);
4. Filtre e imprima apenas por tuplas que possuem valores de peso maiores que 80;
5. Filtre e imprima apenas por tuplas que possuem valores de IMC acima de 35 ($IMC = peso/altura^2$).

Fim