

Streaming de Dados em Tempo Real: Aula 1

Prof. Felipe Timbó



Objetivos

Estudar/investigar princípios, técnicas e ferramentas necessárias para lidar com **streaming de dados**.

Desenvolver soluções em tempo real, isto é, à medida que dados são produzidos.

Resolver problemas relacionados a **streaming de dados em tempo real** com Kafka e Spark

Ementa

- **Dia 1:**
 - Introdução a Streaming de Dados
 - Introdução ao Apache Kafka
 - Setup do ambiente
- **Dia 2:**
 - Data Ingestion com Apache Kafka
 - Kafka Web Project

Ementa

- Dia 3:
 - Introdução ao Apache Spark
- Dia 4:
 - Processamento de dados de Streaming com Apache Spark

Tecnologias e Ferramentas deste Curso

Máquina Virtual (VirtualBox)

Linux (Ubuntu)

Python

VS Code

Apache Kafka

Apache Spark

Metodologia

Aulas expositivas com discussões

Práticas remotas

Leituras

Tarefas individuais

Projeto final (até 3 pessoas)

Recursos

Lista de e-mails:

uni7-ciencia-de-dados-turma10@googlegroups.com

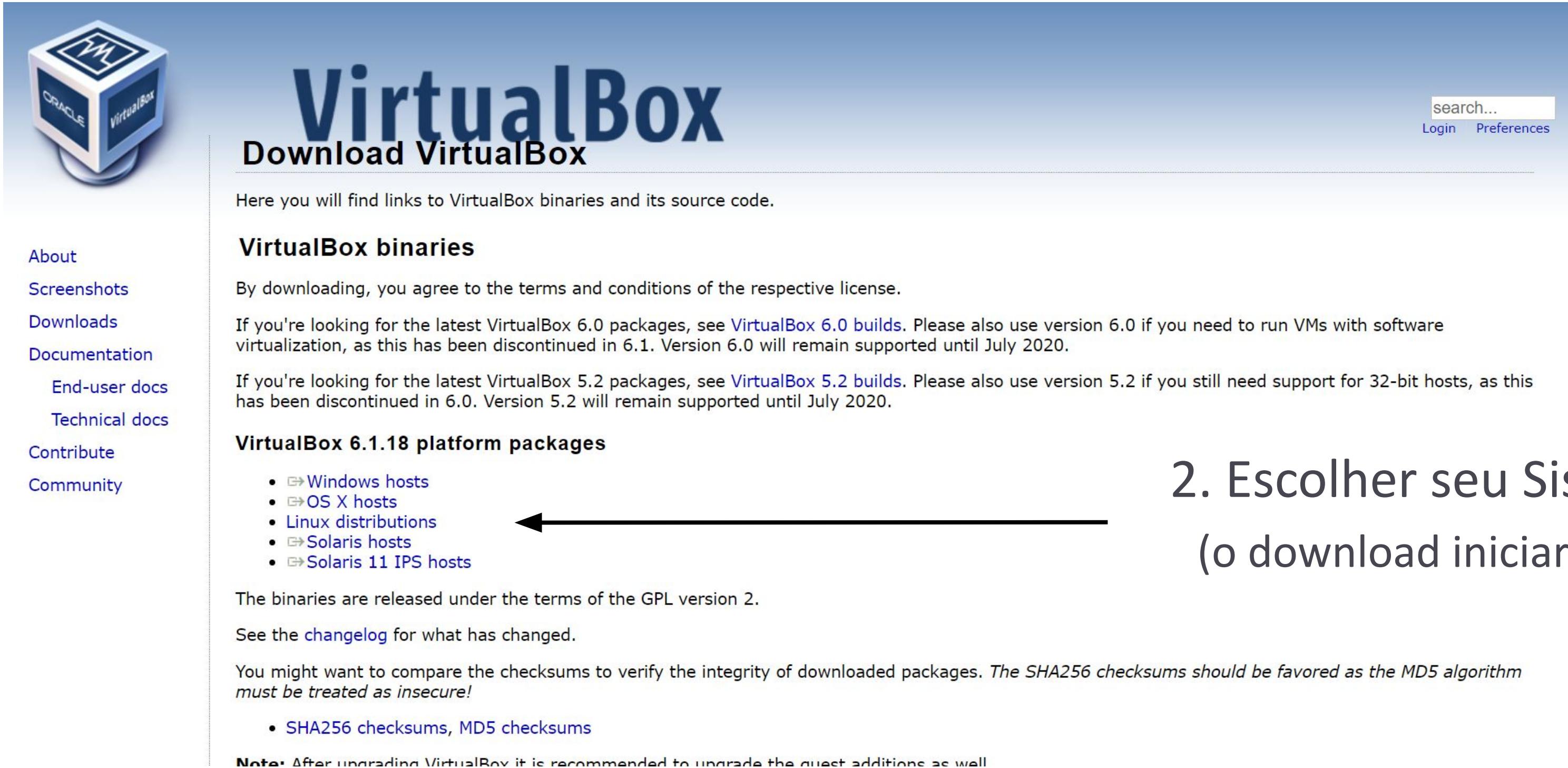
Arquivos (slides, livros e artigos)

Github: <https://github.com/felipetimbo/streaming-data-course>

Antes de tudo...

Download do VirtualBox

1. Acessar <https://www.virtualbox.org/wiki/Downloads>

A screenshot of the VirtualBox download page. The page features a large blue header with the Oracle VM VirtualBox logo on the left. The main title "VirtualBox" is prominently displayed in large blue letters, with "Download VirtualBox" below it. To the right of the title is a search bar and links for "Login" and "Preferences". On the left side, there is a sidebar with links for "About", "Screenshots", "Downloads", "Documentation", "End-user docs", "Technical docs", "Contribute", and "Community". The main content area contains sections for "VirtualBox binaries" and "VirtualBox 6.1.18 platform packages". The "VirtualBox binaries" section includes a note about GPL version 2 and a link to the "changelog". The "VirtualBox 6.1.18 platform packages" section lists options for Windows hosts, OS X hosts, Linux distributions, Solaris hosts, and Solaris 11 IPS hosts. A horizontal arrow points from the "VirtualBox 6.1.18 platform packages" section towards the second step of the guide.

Here you will find links to VirtualBox binaries and its source code.

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see [VirtualBox 6.0 builds](#). Please also use version 6.0 if you need to run VMs with software virtualization, as this has been discontinued in 6.1. Version 6.0 will remain supported until July 2020.

If you're looking for the latest VirtualBox 5.2 packages, see [VirtualBox 5.2 builds](#). Please also use version 5.2 if you still need support for 32-bit hosts, as this has been discontinued in 6.0. Version 5.2 will remain supported until July 2020.

VirtualBox 6.1.18 platform packages

- [Windows hosts](#)
- [OS X hosts](#)
- [Linux distributions](#)
- [Solaris hosts](#)
- [Solaris 11 IPS hosts](#)

The binaries are released under the terms of the GPL version 2.

See the [changelog](#) for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!*

- [SHA256 checksums](#), [MD5 checksums](#)

Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well

2. Escolher seu Sistema Operacional
(o download iniciará automaticamente)

Download do Ubuntu

3. Acessar: <https://ubuntu.com/download/desktop>

4. Realizar o download do
Ubuntu 22.04.2 LTS

Ubuntu 22.04.2 LTS

The latest LTS version of Ubuntu, for desktop PCs and laptops. LTS stands for long-term support — which means five years of free security and maintenance updates, guaranteed until April 2027.

[Ubuntu 22.04 LTS release notes](#)

Recommended system requirements:

- ✓ 2 GHz dual-core processor or better
- ✓ 4 GB system memory
- ✓ 25 GB of free hard drive space
- ✓ Internet access is helpful
- ✓ Either a DVD drive or a USB port for the installer media

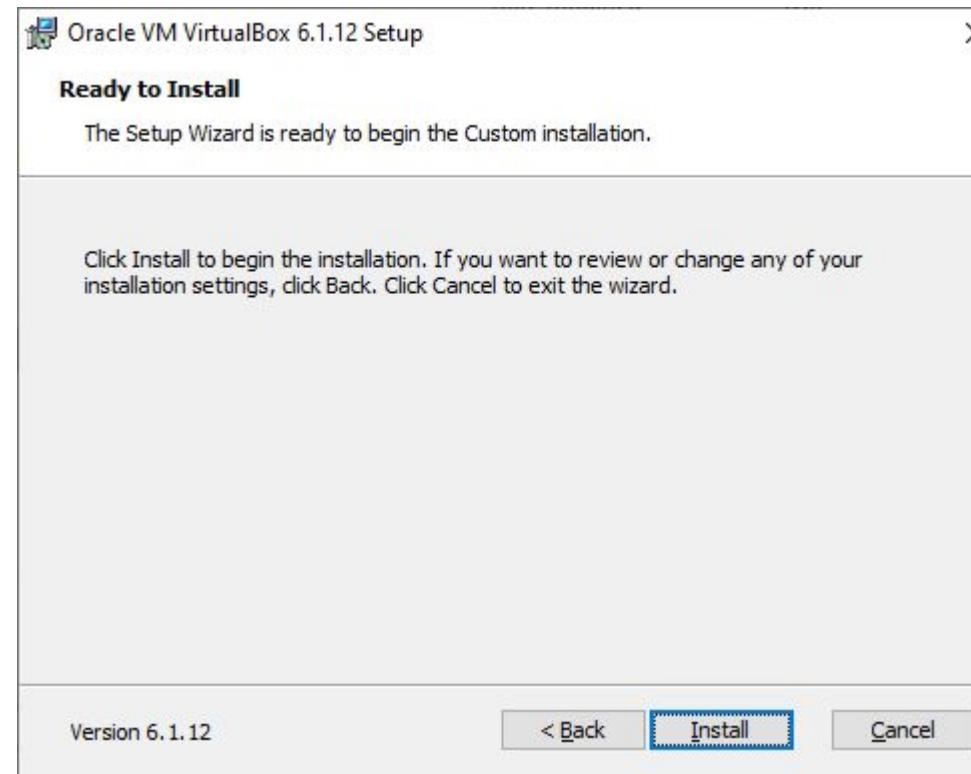
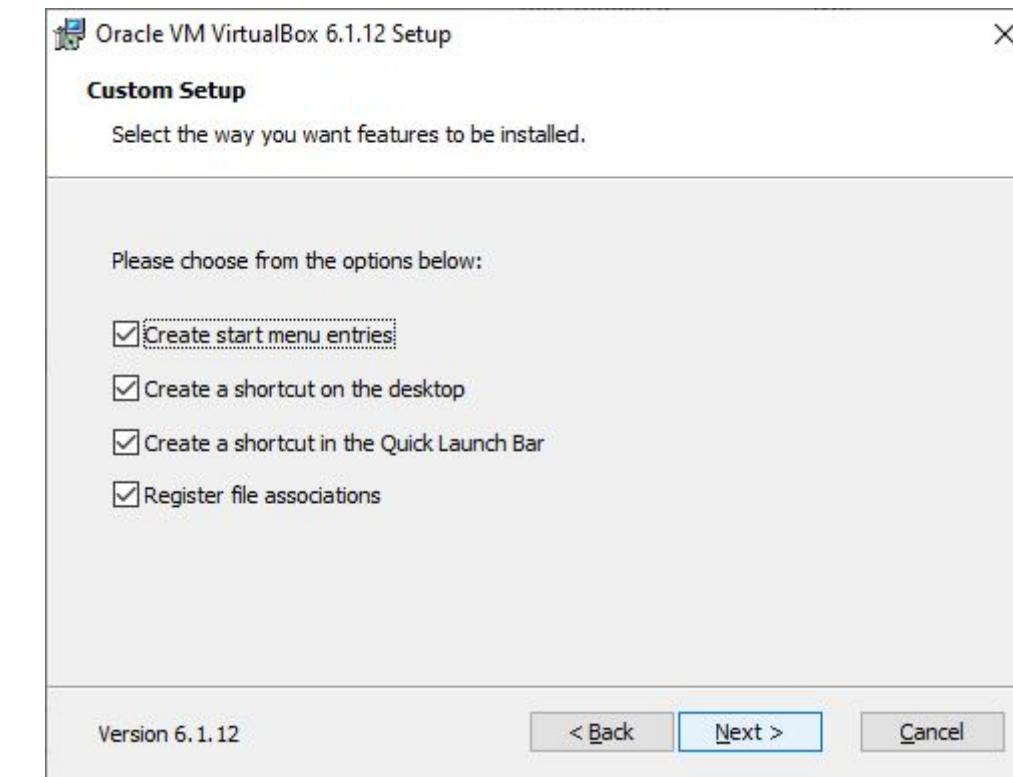
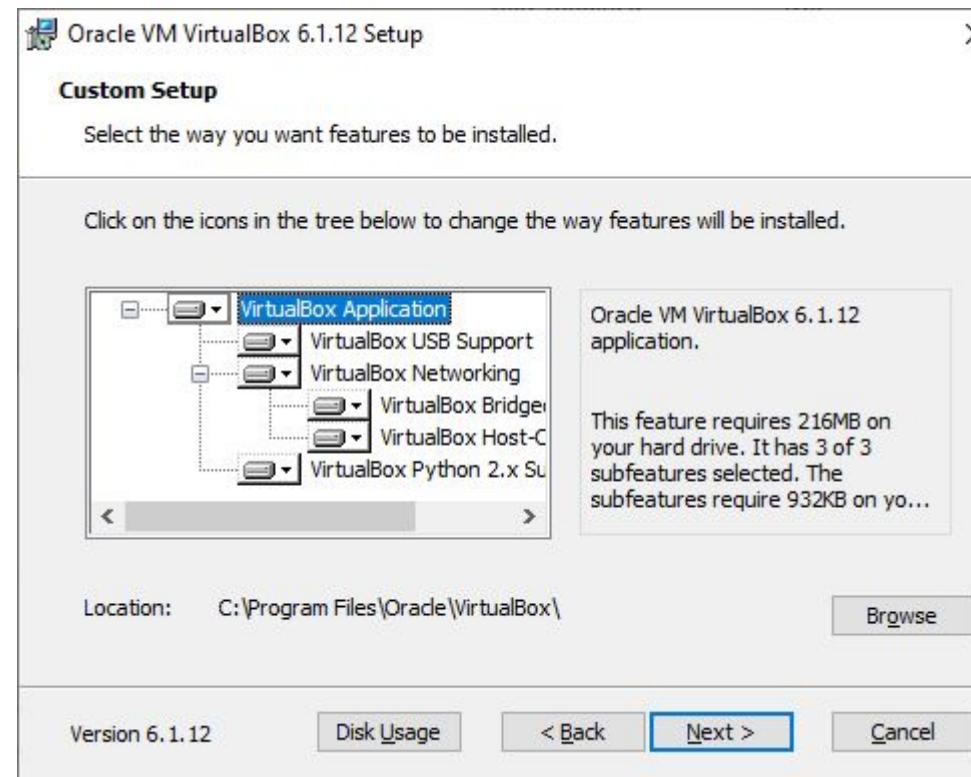


[Download](#)

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors and past releases [see our alternative downloads](#).

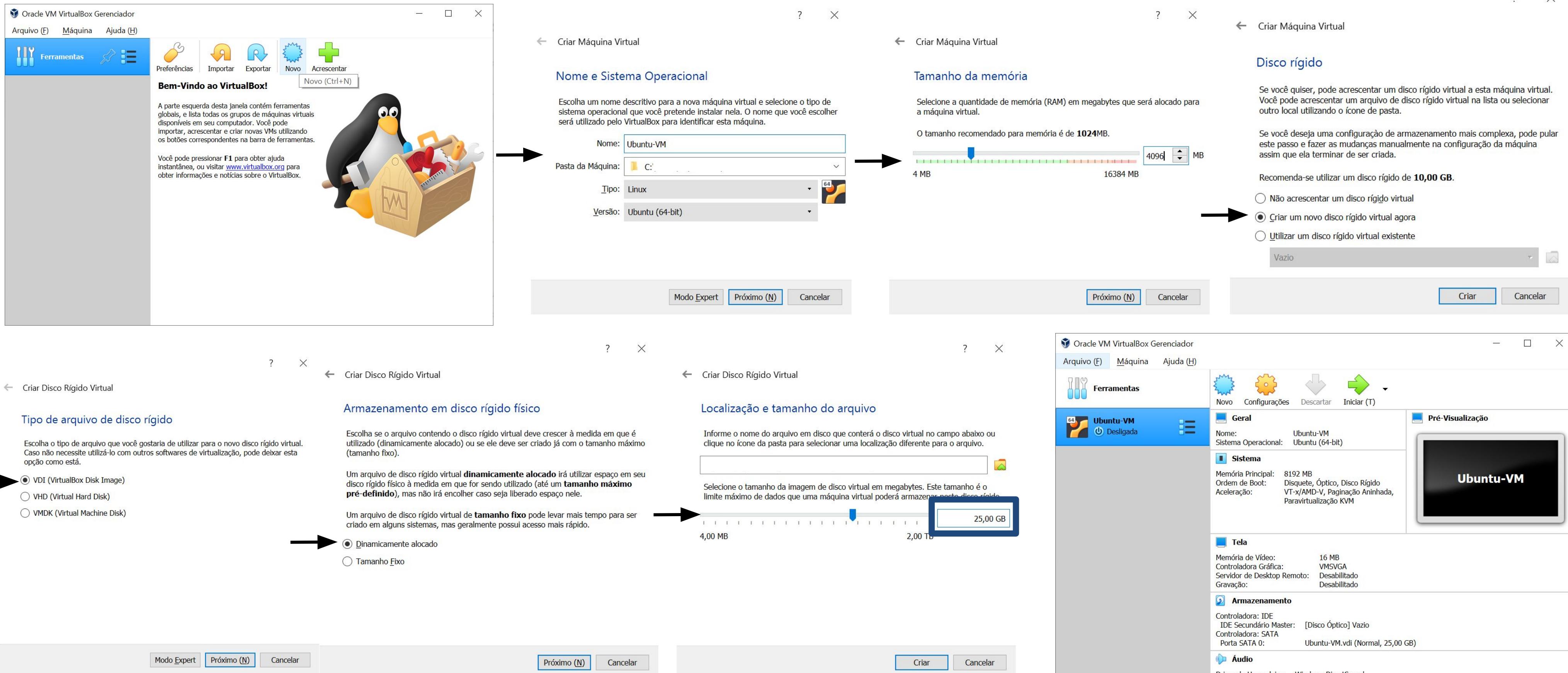
Instalação do VirtualBox

5. Instalar o VirtualBox



Criação de uma VM (Virtual Machine)

6. Criar uma nova Máquina Virtual Ubuntu



The image shows a step-by-step guide for creating a new Virtual Machine (VM) in Oracle VM VirtualBox. It consists of five windows arranged horizontally:

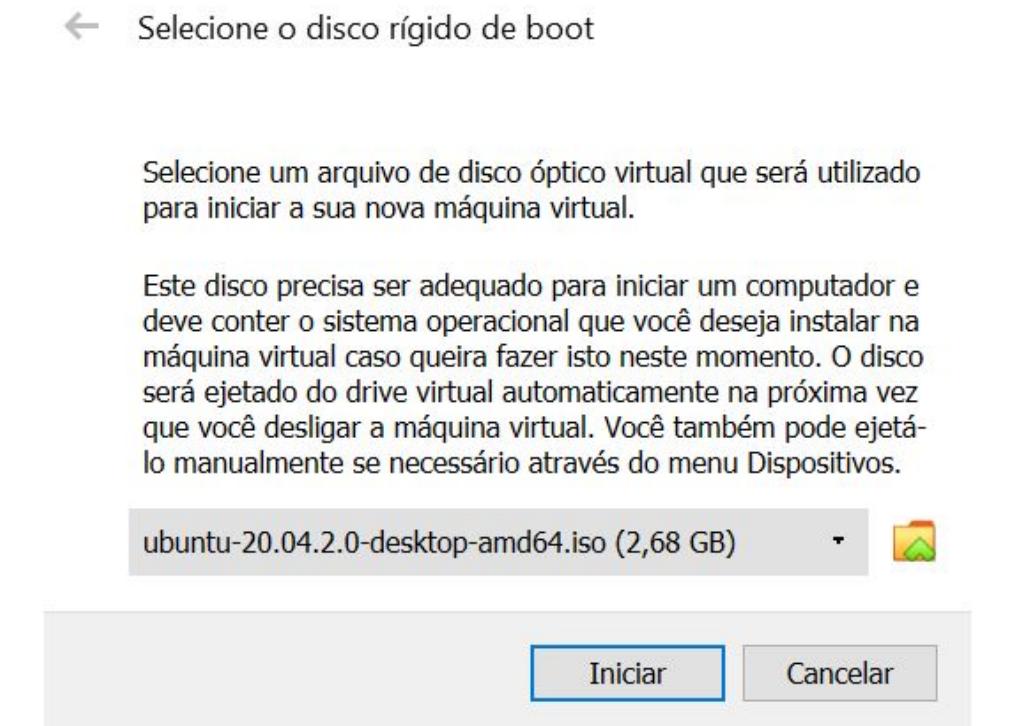
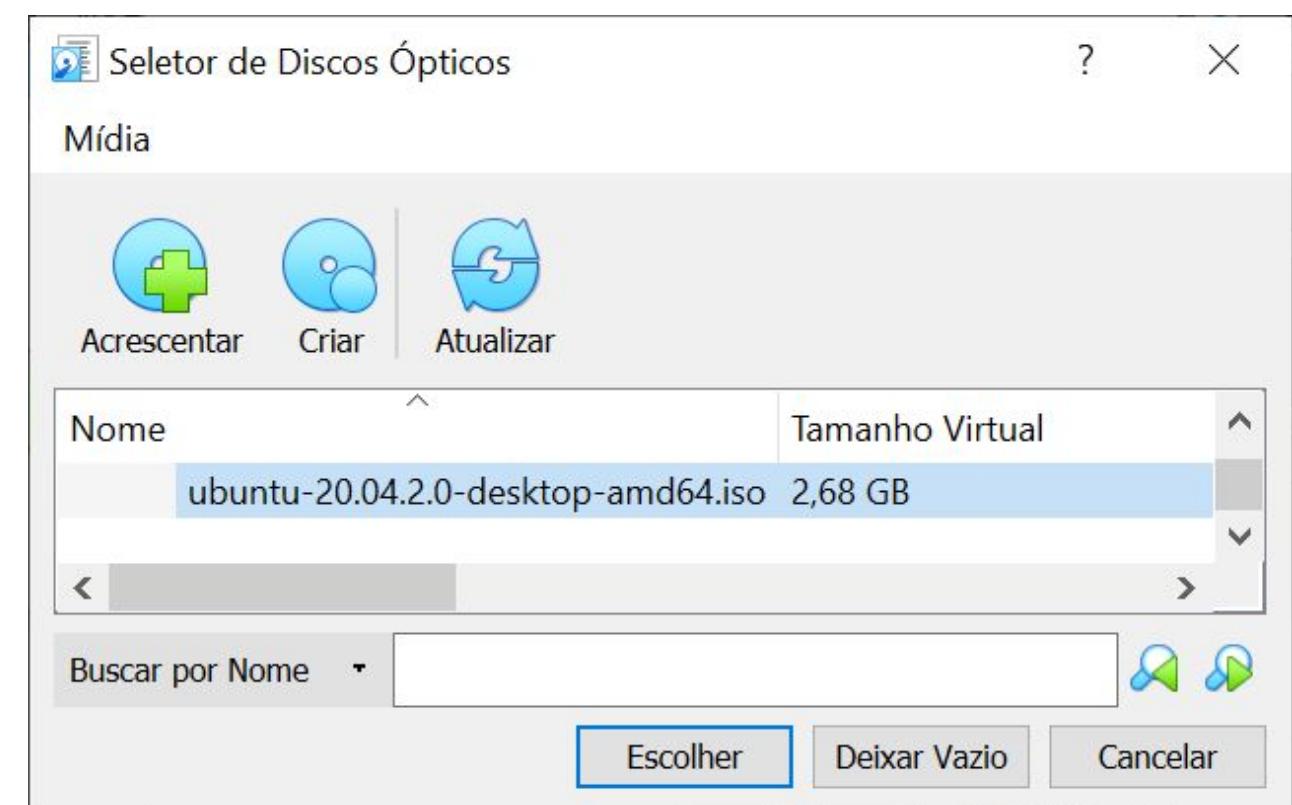
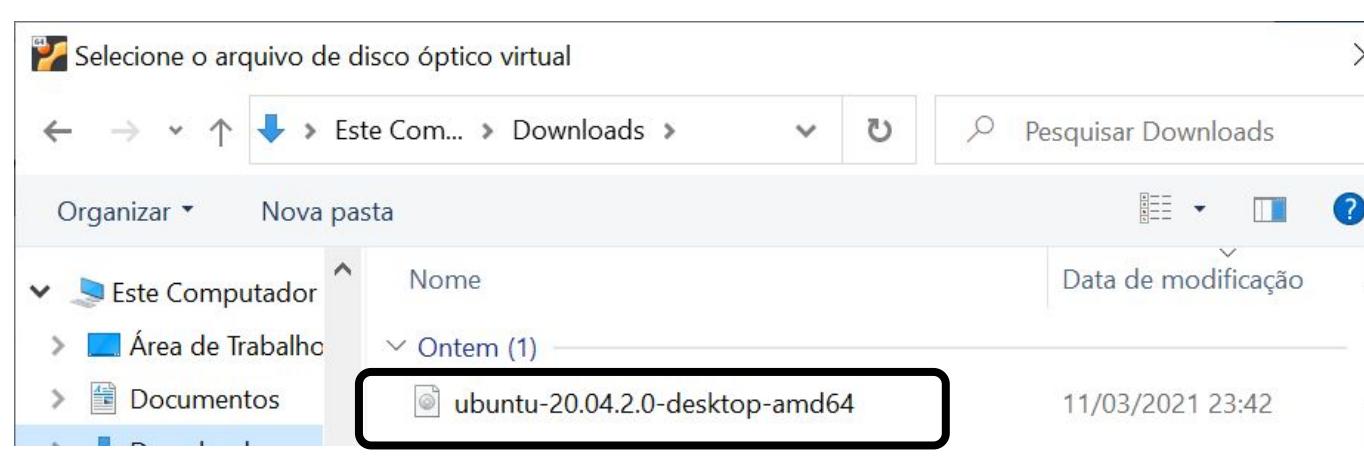
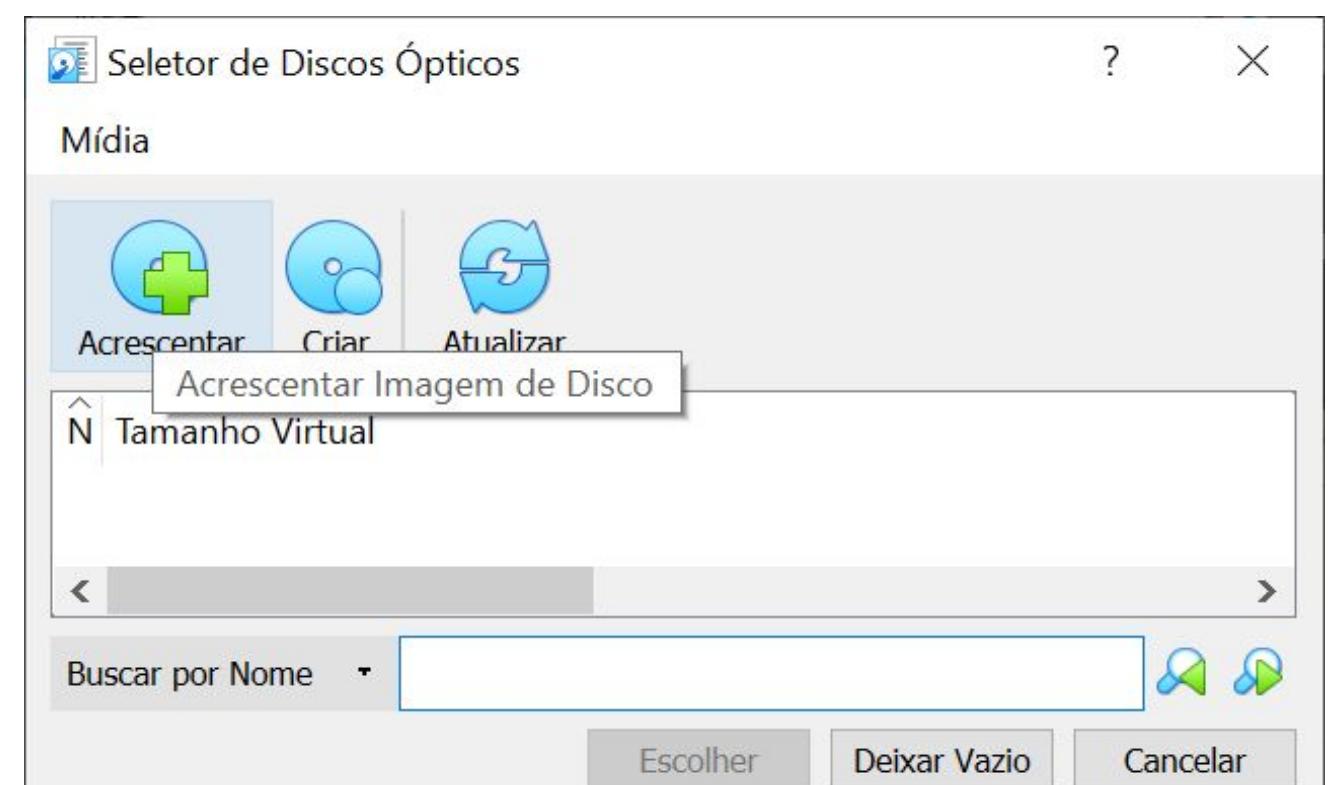
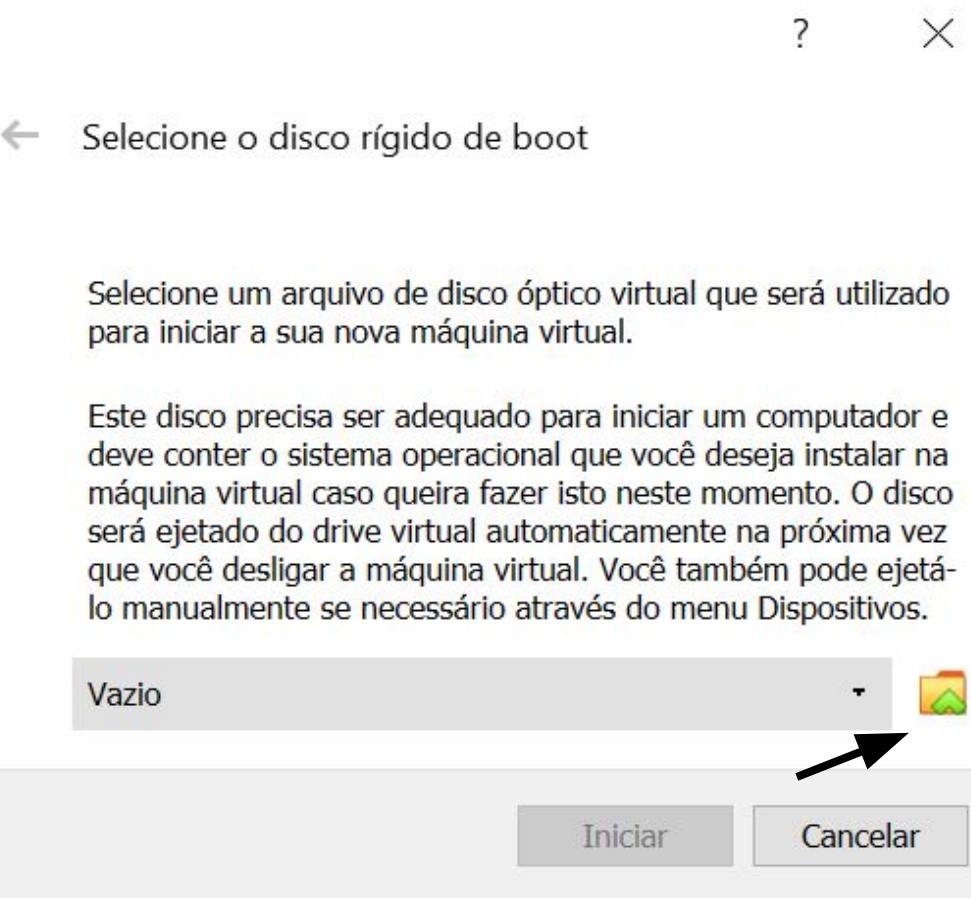
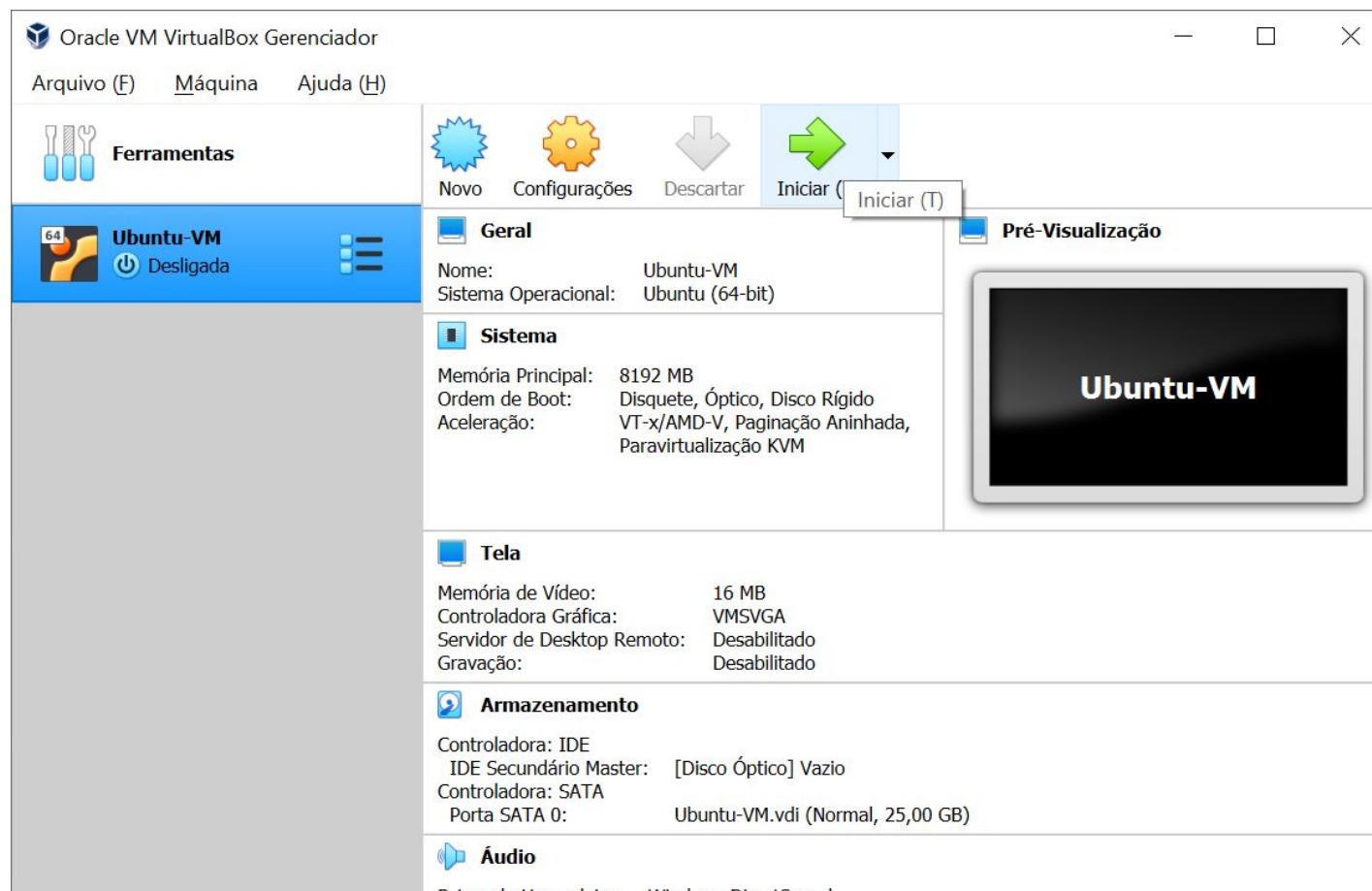
- Oracle VM VirtualBox Gerenciador:** Main interface showing the 'Bem-Vindo ao VirtualBox!' (Welcome to VirtualBox!) screen.
- Criar Máquina Virtual - Nome e Sistema Operacional:** Step 1 of the wizard. Name: Ubuntu-VM, Type: Linux, Version: Ubuntu (64-bit).
- Criar Máquina Virtual - Tamanho da memória:** Step 2 of the wizard. RAM size: 4096 MB (1024 MB recommended).
- Criar Máquina Virtual - Disco rígido:** Step 3 of the wizard. Option selected: Criar um novo disco rígido virtual agora.
- Criar Disco Rígido Virtual - Tipo de arquivo de disco rígido:** Step 4 of the wizard. Type selected: VDI (VirtualBox Disk Image).
- Criar Disco Rígido Virtual - Armazenamento em disco rígido físico:** Step 5 of the wizard. Allocation type: Dinamicamente alocado (Dynamic allocation).
- Criar Disco Rígido Virtual - Localização e tamanho do arquivo:** Step 6 of the wizard. File size: 25,00 GB.
- Oracle VM VirtualBox Gerenciador:** Final view of the VirtualBox Manager showing the newly created VM 'Ubuntu-VM' listed under 'Geral' (General) tab.

Annotations with arrows highlight specific steps and configurations:

- An arrow points from the 'Nome' field in the first window to the 'Nome' field in the second window.
- An arrow points from the 'Nome' field in the second window to the 'Nome' field in the third window.
- An arrow points from the 'Nome' field in the third window to the 'Nome' field in the fourth window.
- An arrow points from the 'Tipo' dropdown in the second window to the 'Tipo' dropdown in the third window.
- An arrow points from the 'Nome' field in the fourth window to the 'Nome' field in the fifth window.
- An arrow points from the 'Allocation type' dropdown in the fifth window to the 'Allocation type' dropdown in the sixth window.
- An arrow points from the 'File size' slider in the sixth window to the 'File size' input field in the seventh window.

Criação de uma VM (Virtual Machine)

7. Iniciar a Máquina Virtual Ubuntu

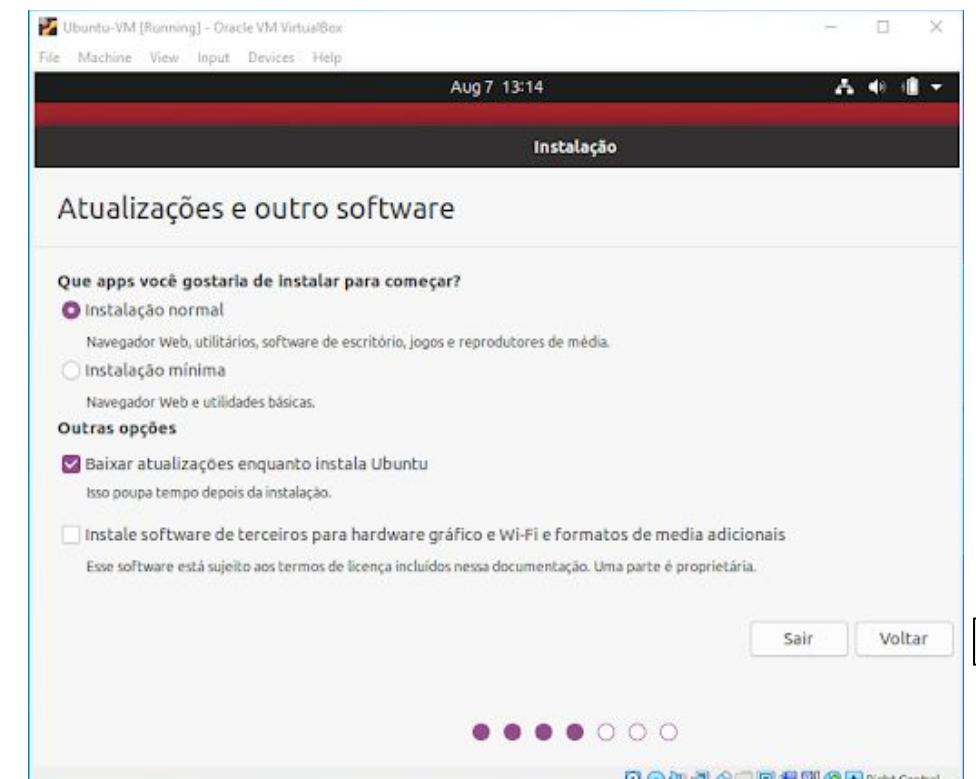
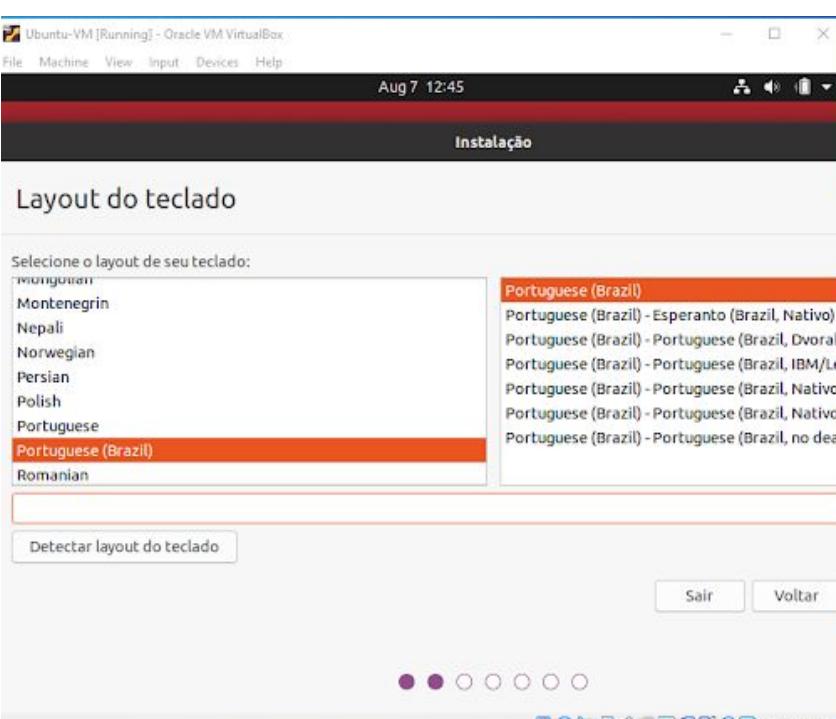


Instalação do Ubuntu

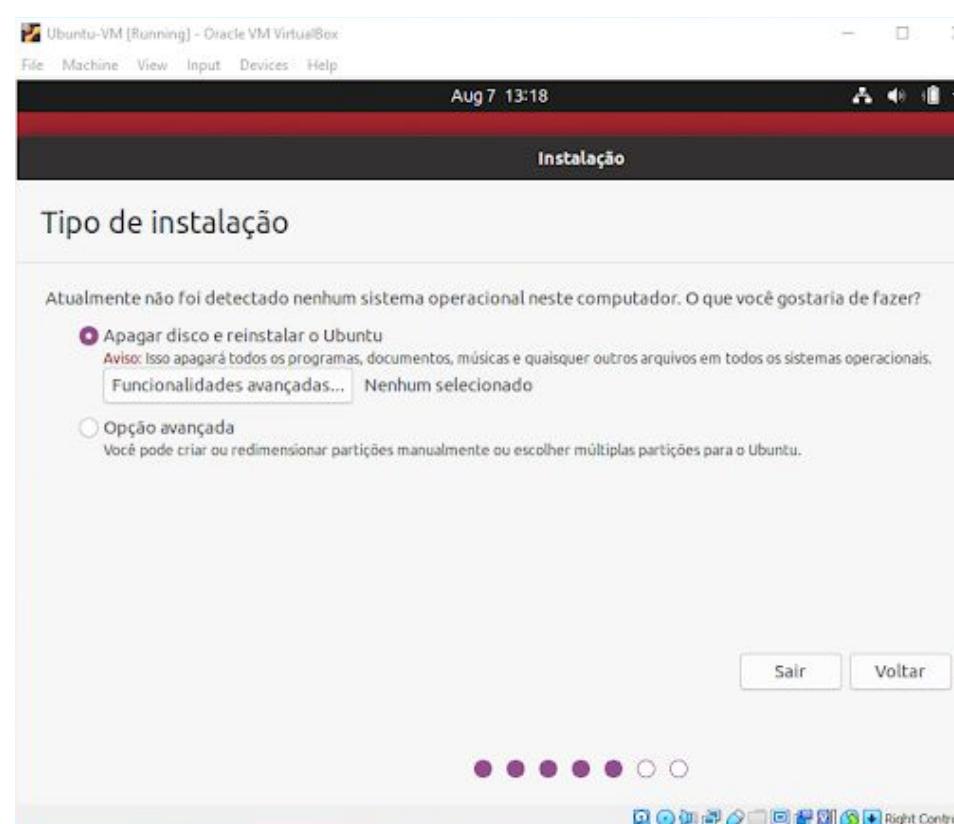
8. Instalar o Ubuntu na VM



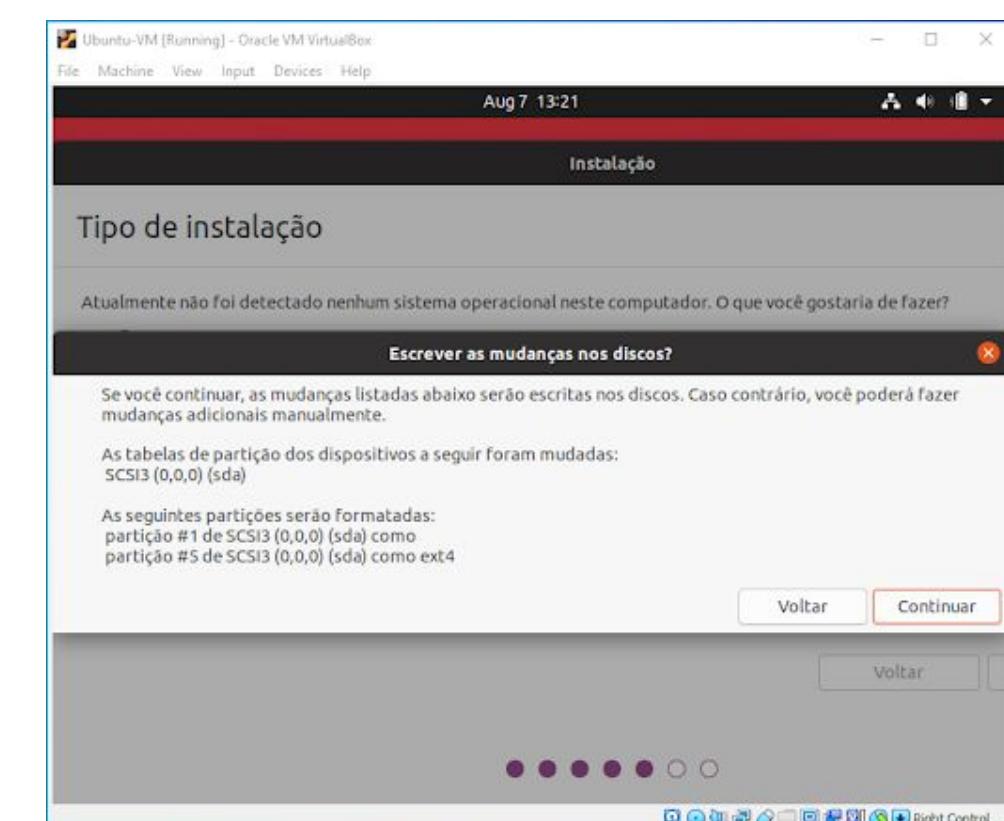
dois cliques



o botão 'continuar' pode
estar escondido aqui.
Acessá-lo via tecla 'tab'

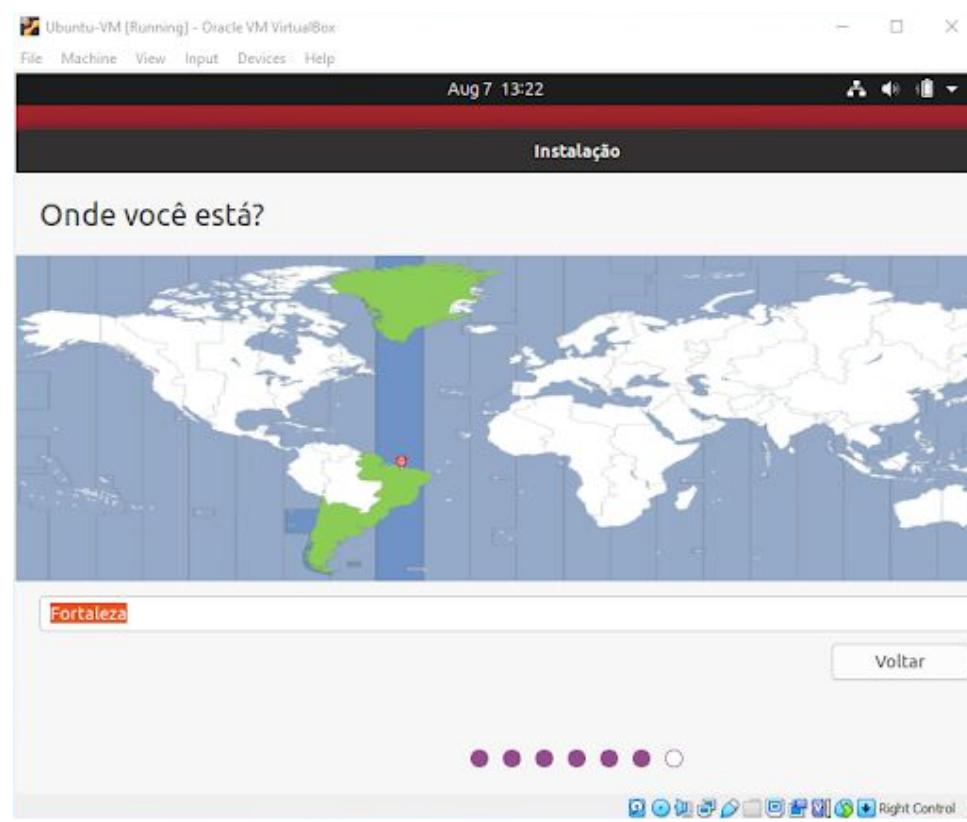


o botão 'continuar' pode
estar escondido aqui.
Acessá-lo via tecla 'tab'

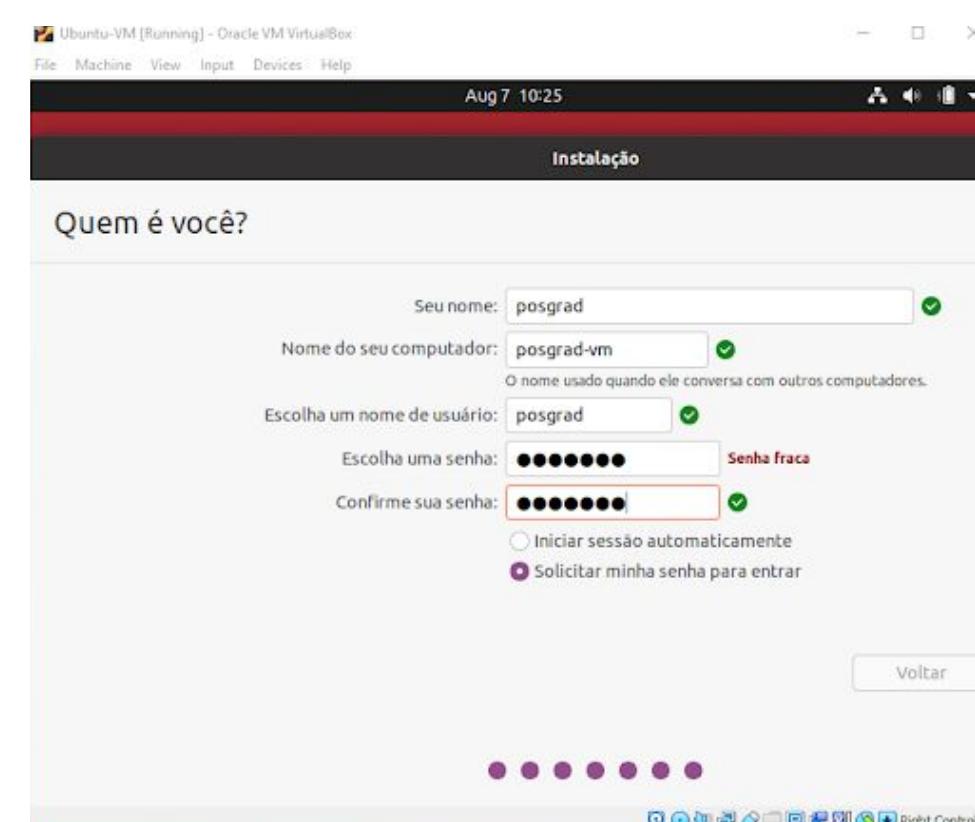


Instalação do Ubuntu

8. Instalar o Ubuntu na VM

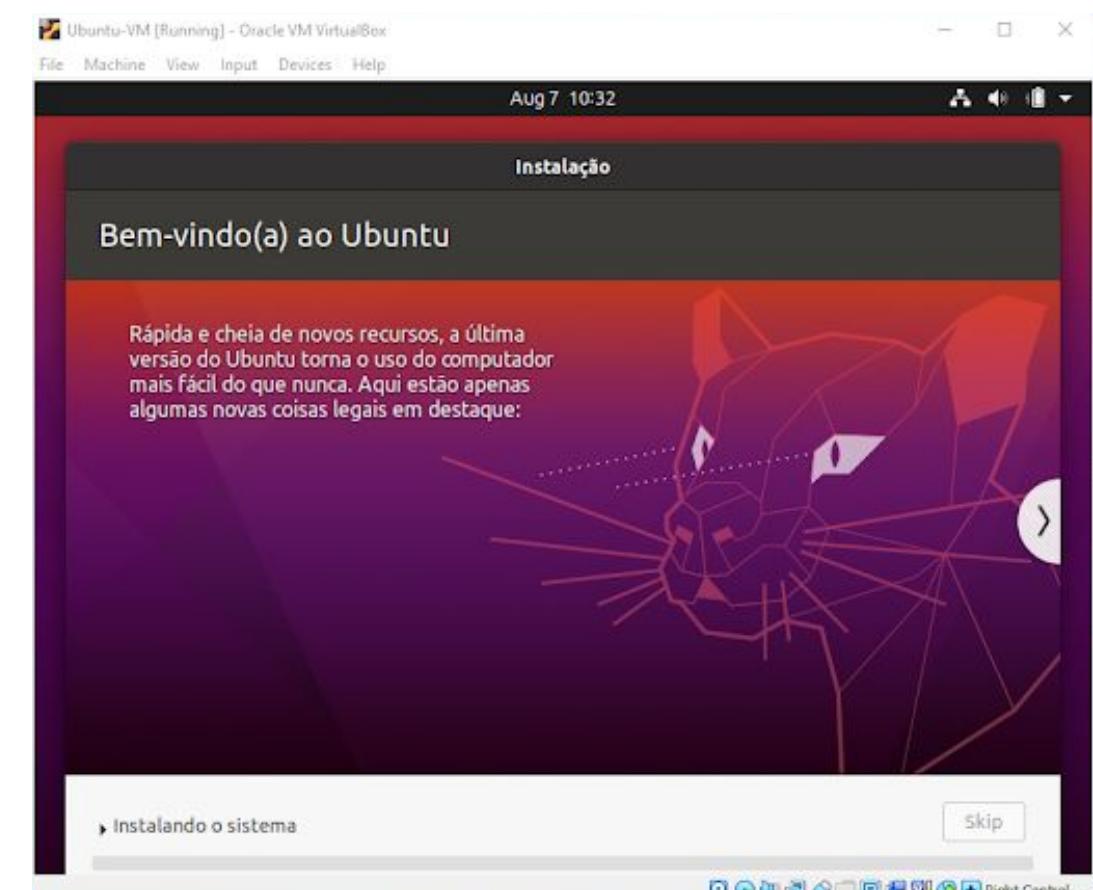


o botão 'continuar' pode
estar escondido aqui.
Acessá-lo via tecla 'tab'



o botão 'continuar' pode
estar escondido aqui.
Acessá-lo via tecla 'tab'

{ usuário: posgrad
senha: posgrad



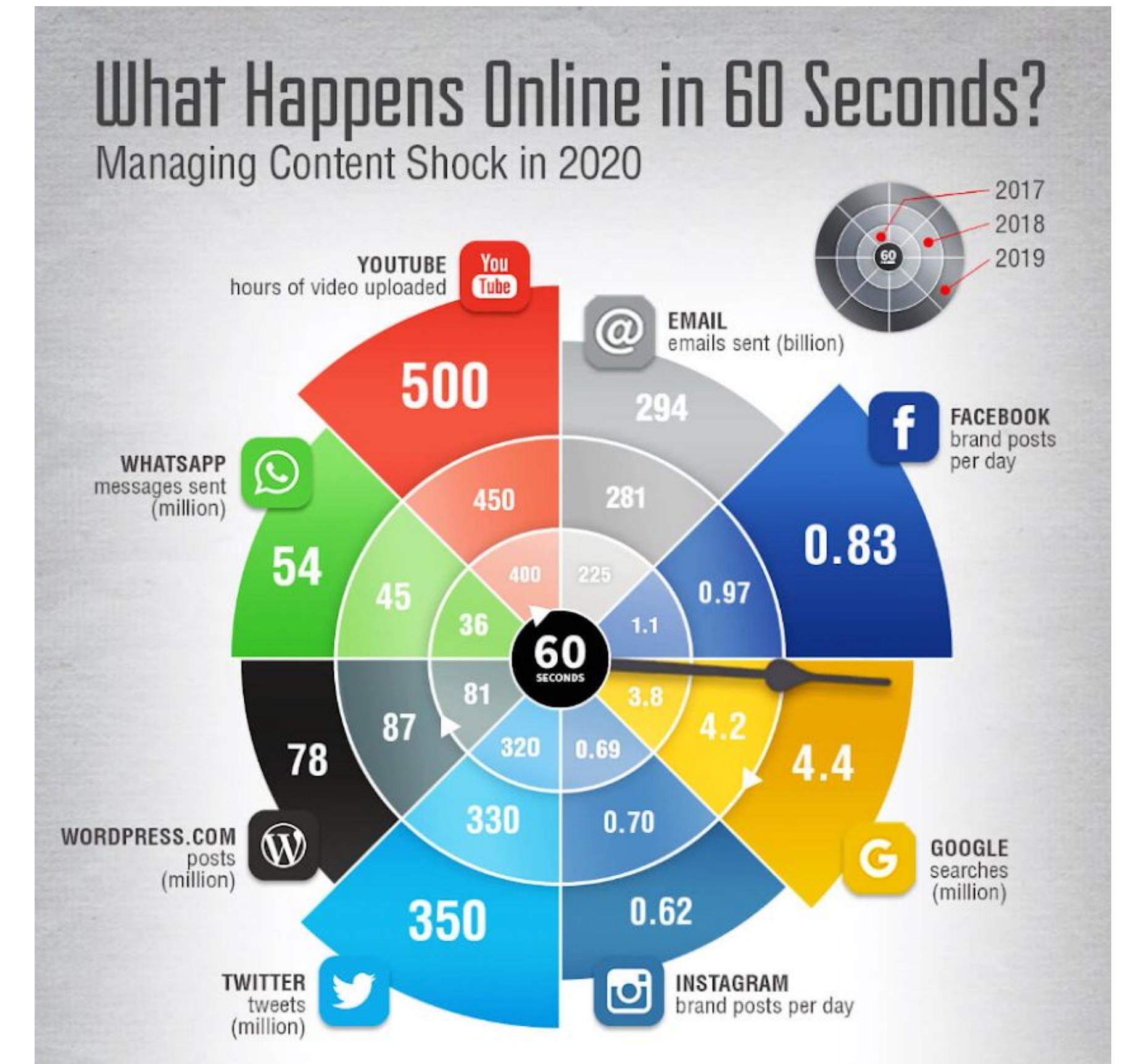
Deixar instalando...

Introdução

O que acontece em 60 segundos?

Upload de 500h de vídeo no YouTube

54 milhões de
mensagens no Whatsapp



Big Data

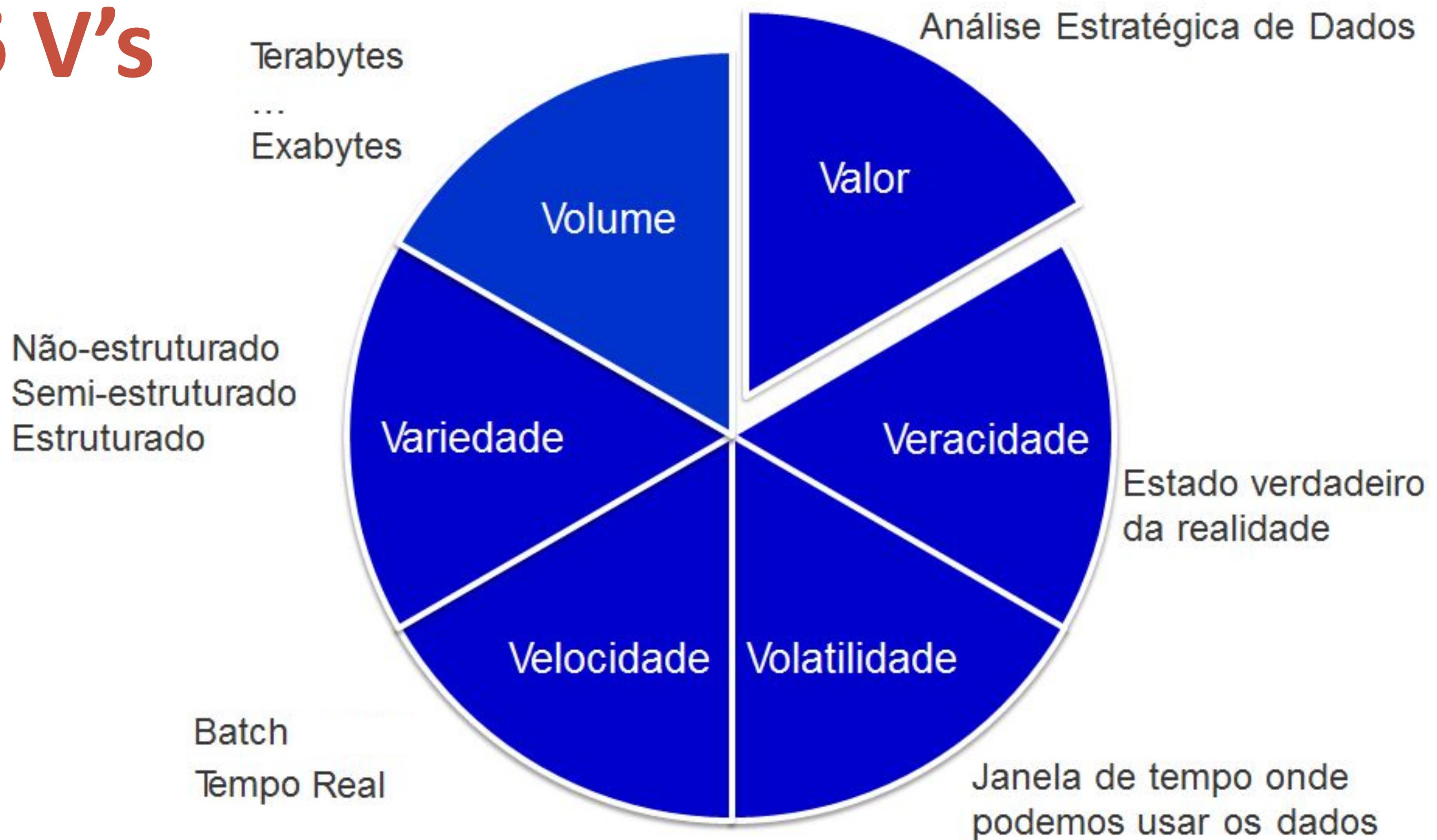
Big Data são dados que excedem o **armazenamento, o processamento e a capacidade dos sistemas convencionais**

Volume de dados muito grande

Dados são gerados rapidamente

Dados não se encaixam nas estruturas de arquiteturas de sistemas atuais

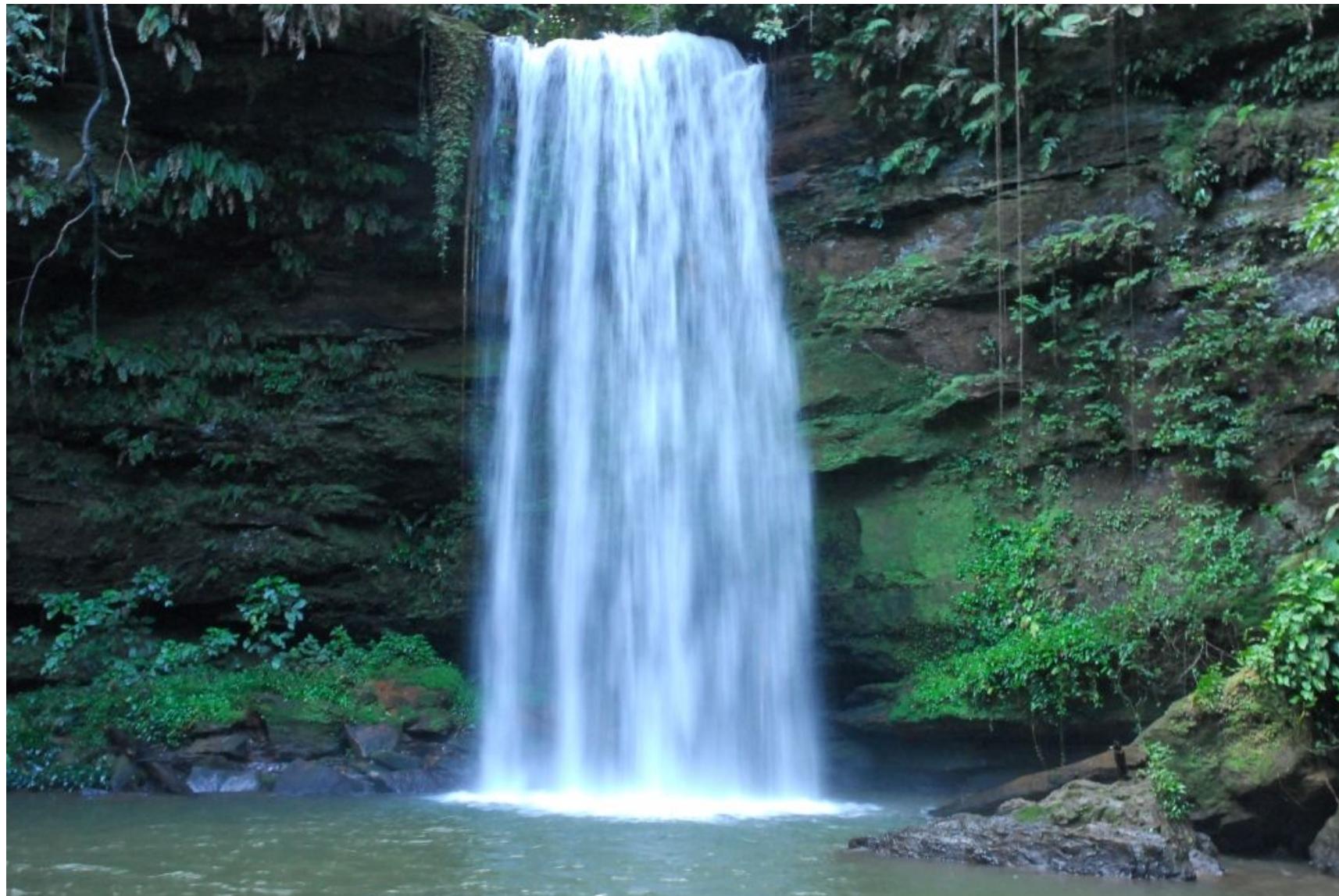
6 V's



Streaming

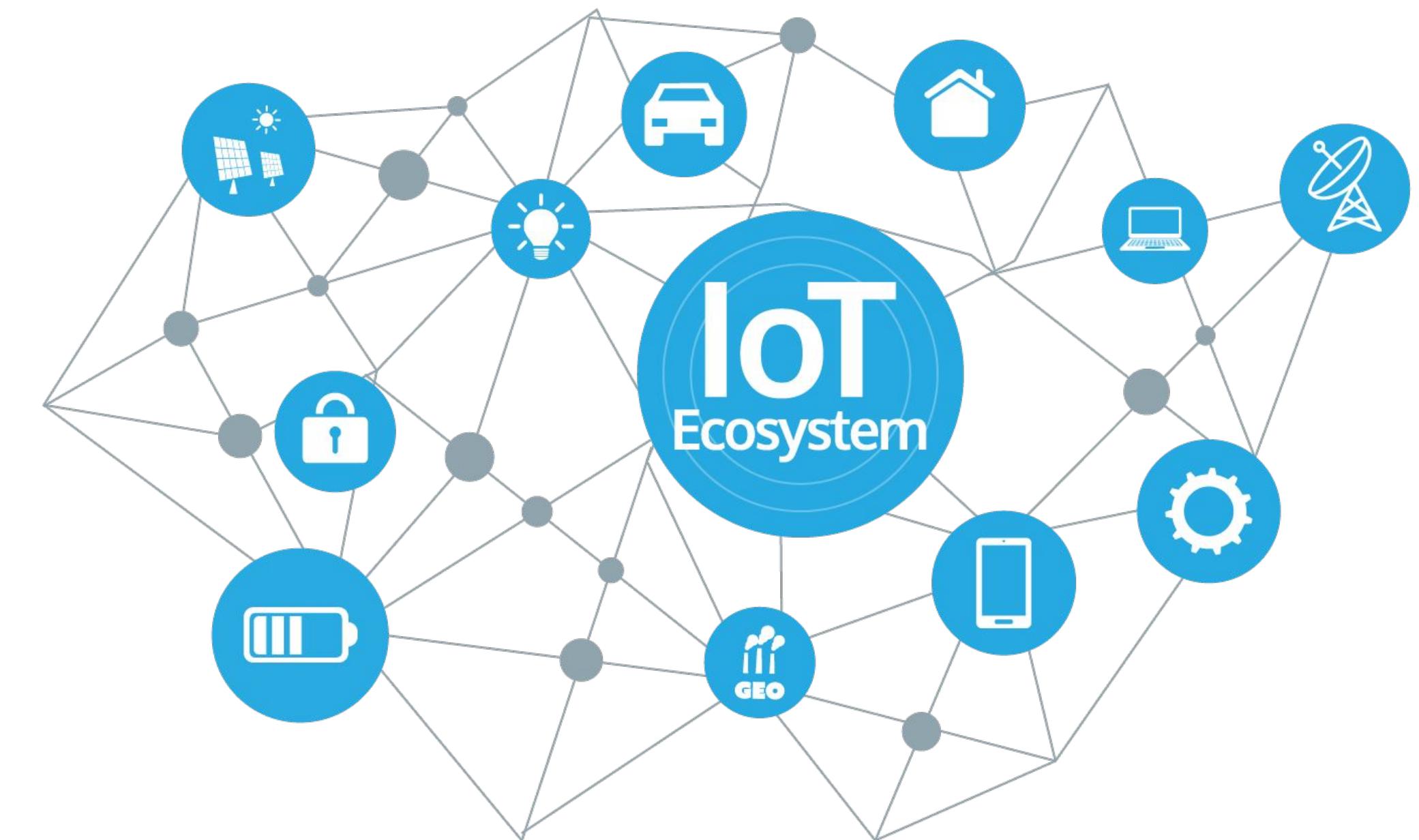
Streaming

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

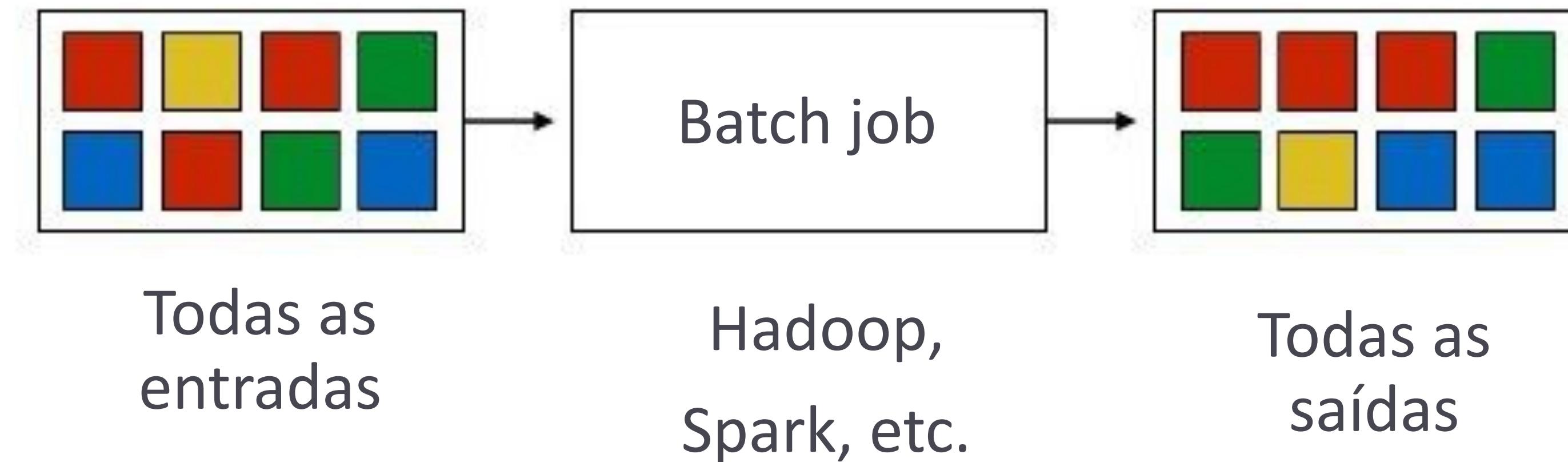
Dados semi-estruturados: XML, json

Dados em grafo: redes sociais

Dados de mobilidade: coordenadas geográficas x tempo

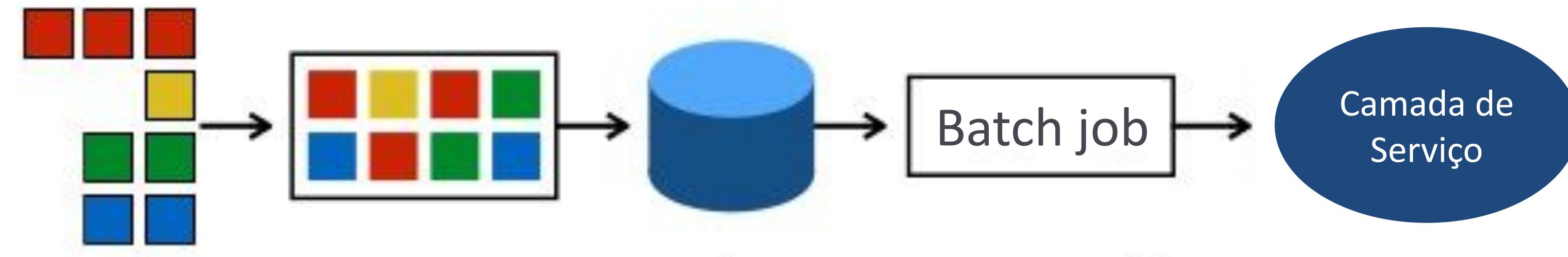
Etc.

Processamento em Batch



Processamento de Streaming

Em geral:



continuamente
produzido

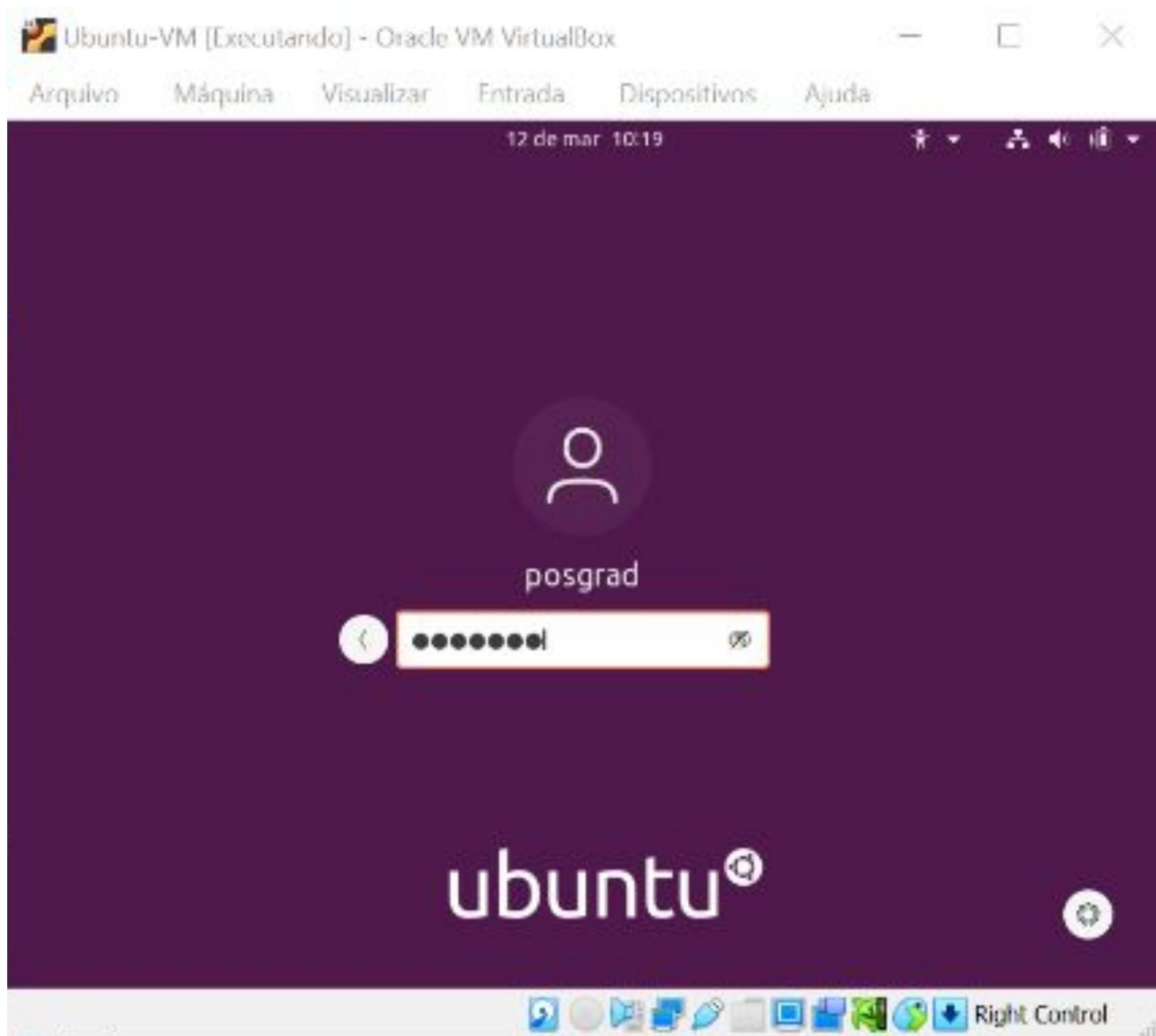
arquivos são
streams finitos

periodicamente
executado

Continuando
nossa Setup...

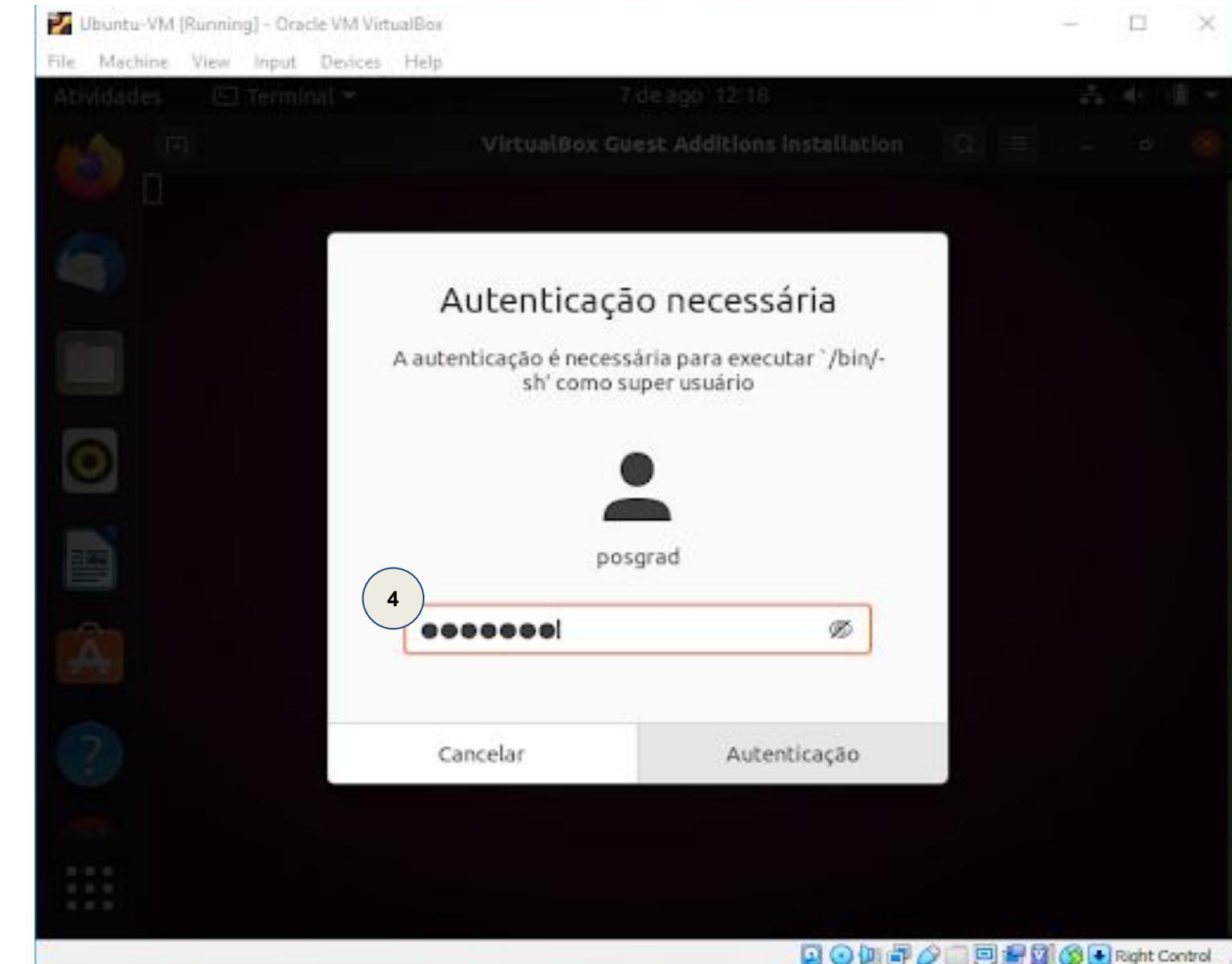
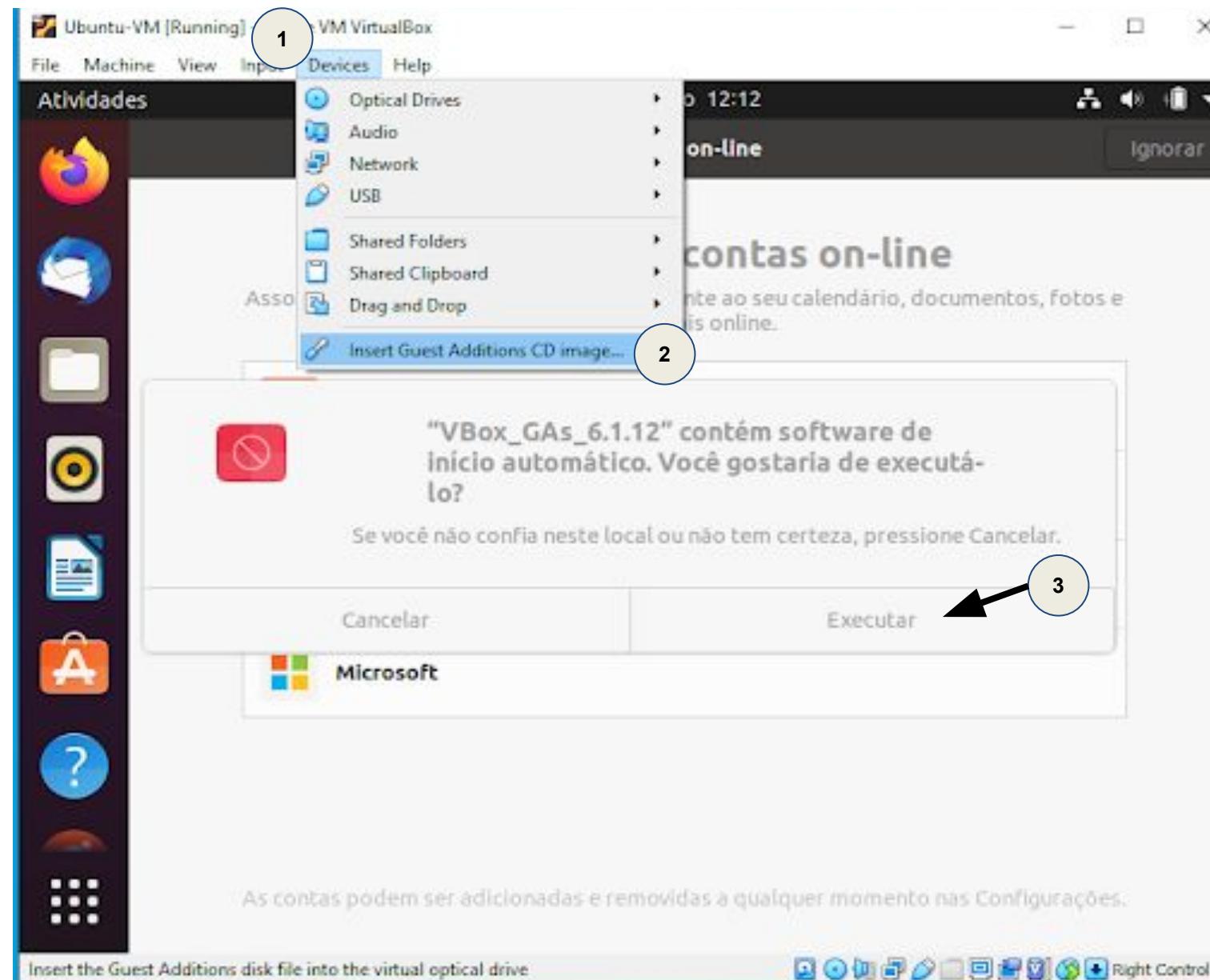
Configuração da VM Ubuntu

9. Logar na VM



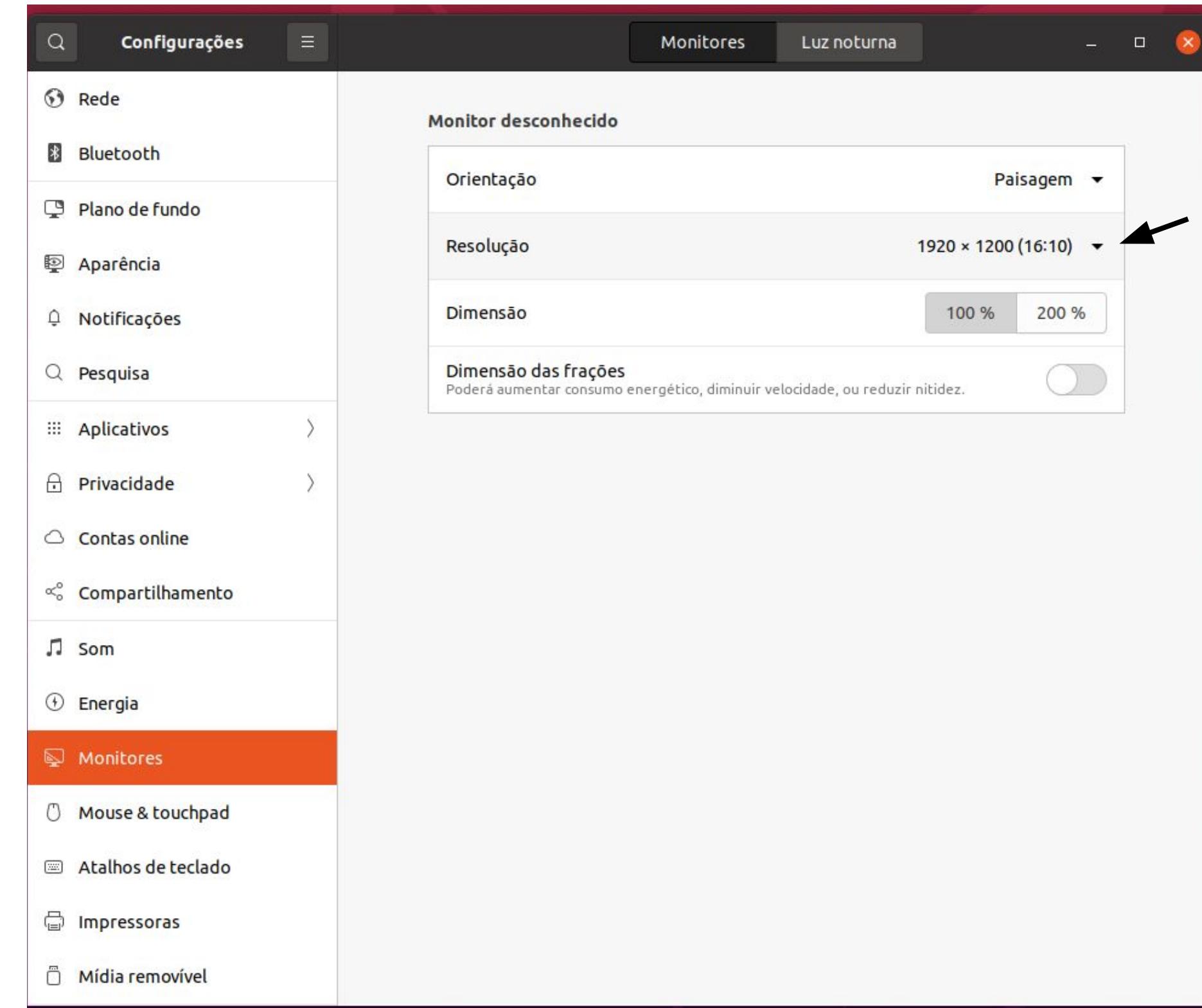
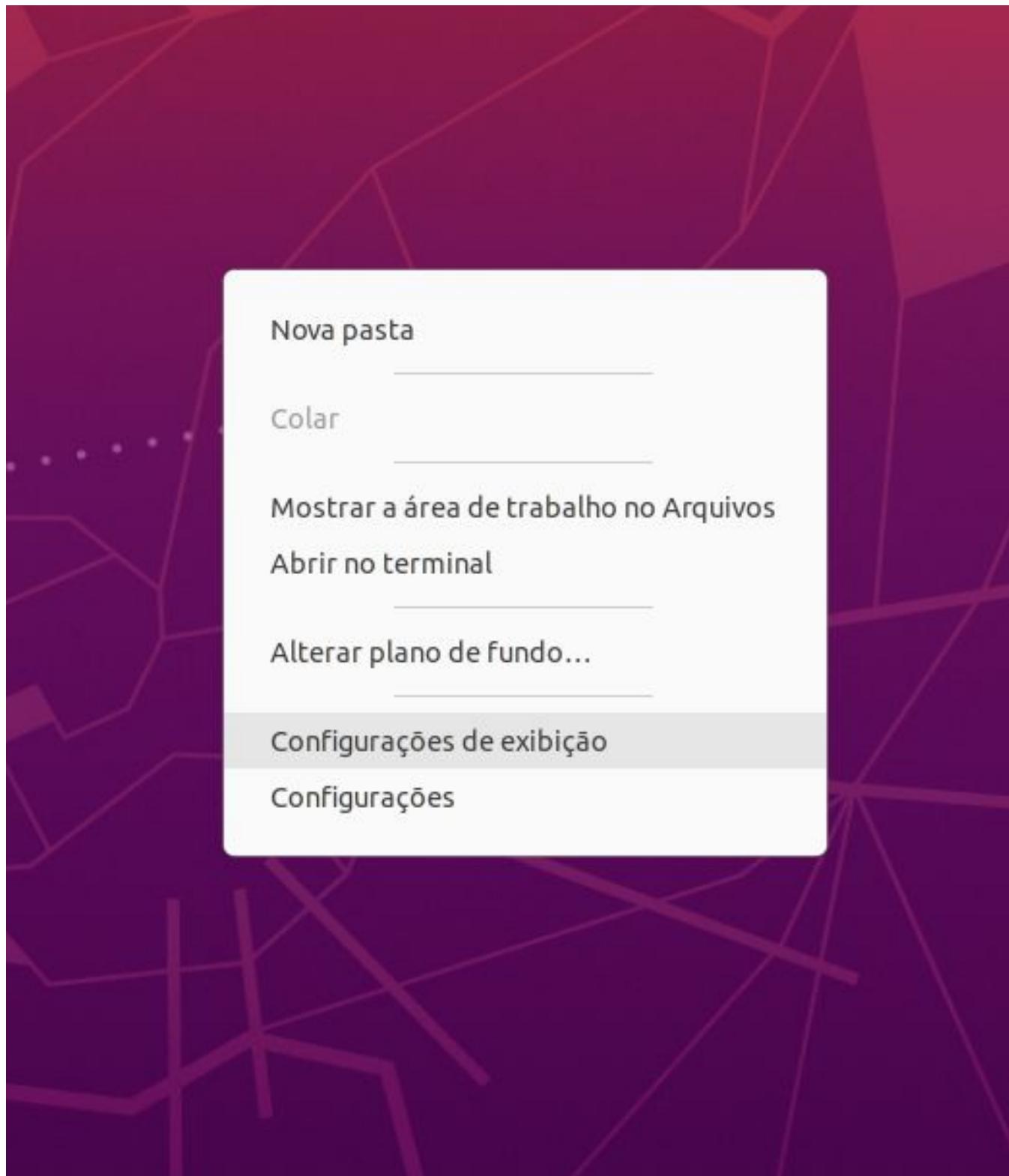
Configuração da VM Ubuntu

10. Instalar add-ons (para melhor experiência com o Ubuntu)



Configuração da VM Ubuntu

11. Ajustar resolução de tela



Atualizando o Ubuntu

Abrir o terminal (Alt+F2 e digitar gnome-terminal)

Atualizar o Ubuntu

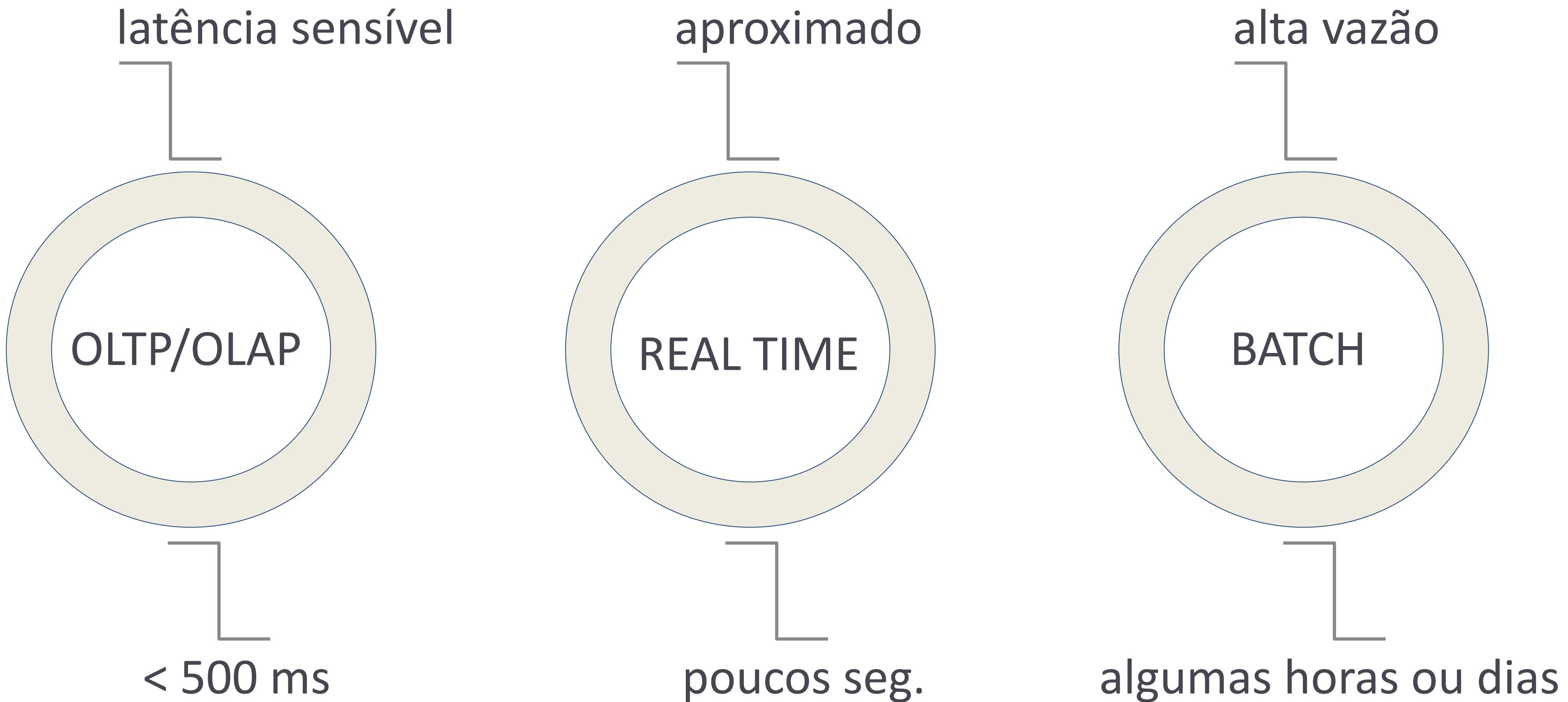
- sudo apt update
- sudo apt upgrade



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

Fonte: Real-Time Analytics with Apache Storm - <https://www.udacity.com/course/ud381>

Problemas em streaming

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



Finalizando
nossa Setup...

Instalando algumas libs

Instalar o curl

```
➤ sudo apt install curl
```

Instalar o VSCode

```
➤ sudo snap install --classic code
```

Instalar o Python

```
➤ sudo apt install python3-pip
```

Instalar o Java

```
➤ sudo apt install default-jdk
```

Deixar instalando...

Apache Kafka

Apache Kafka

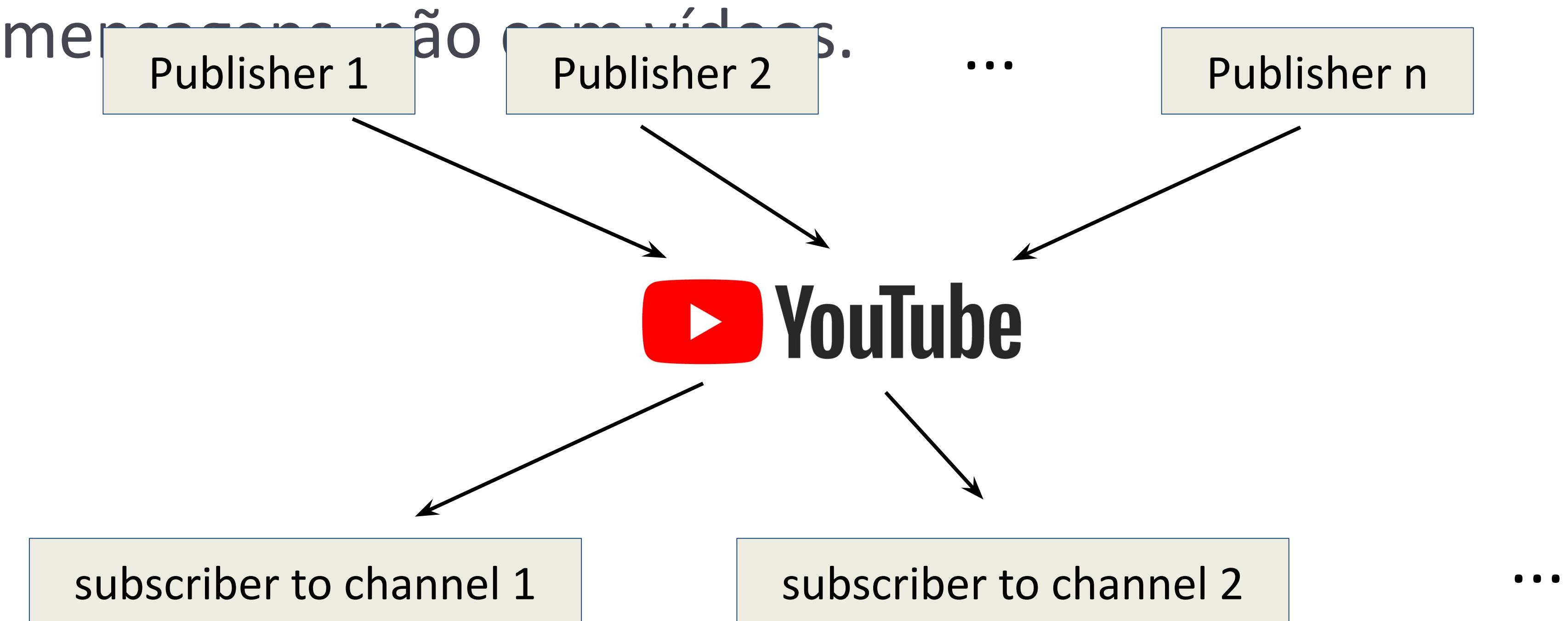
- Sistema de mensagens
 - Distribuído
 - Com alta vazão (*throughput*)
 - De geração (publicação) e leitura (sub-inscrição)
- Principais casos de uso:
 - Agregação de log
 - Mover/transformar conjuntos de dados em tempo real
 - Monitoramento

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.

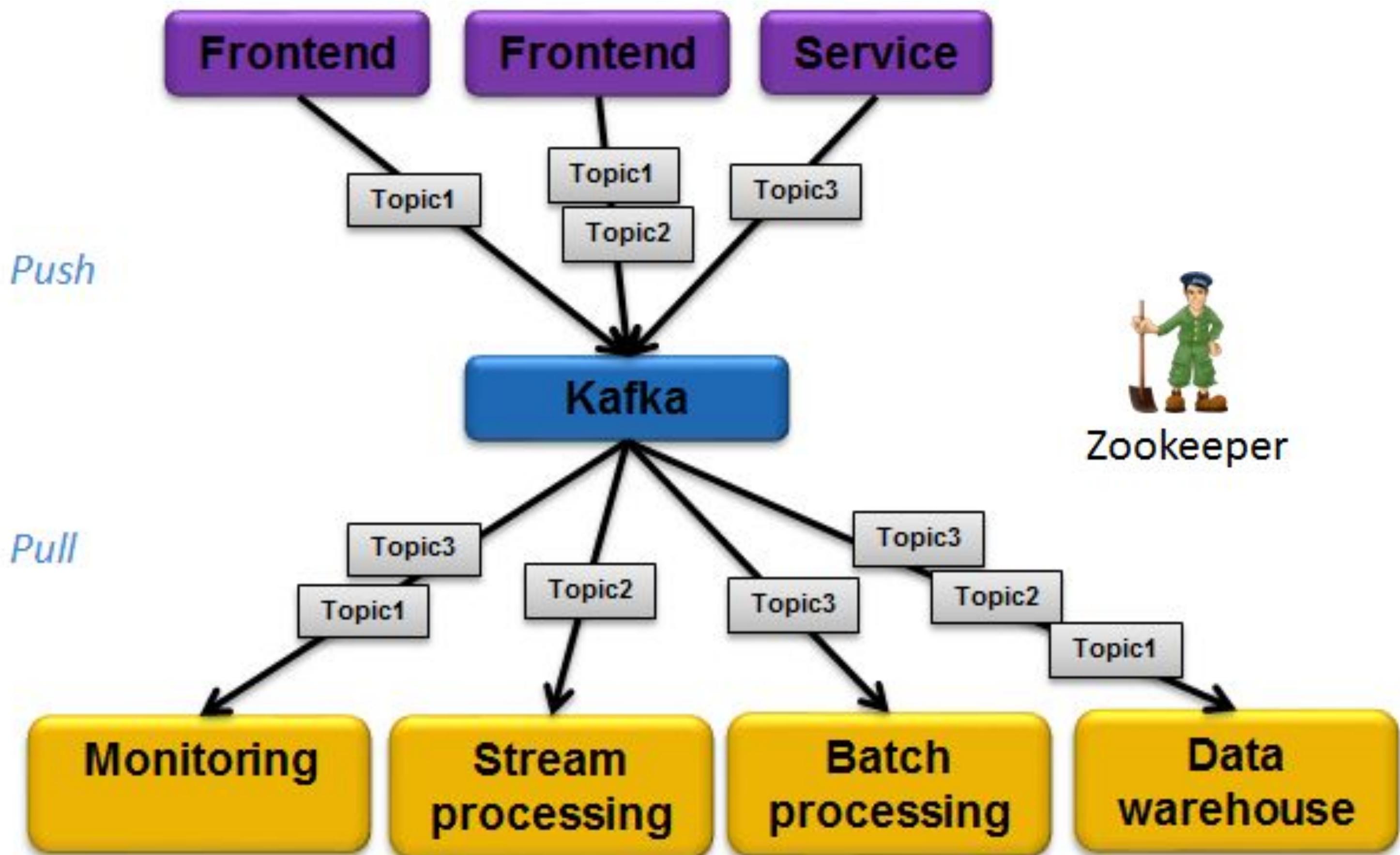
Publisher-subscriber system

- Kafka pode ser visto como um sistema publisher/subscriber, como o Youtube, mas com mensagens não como vídeos.



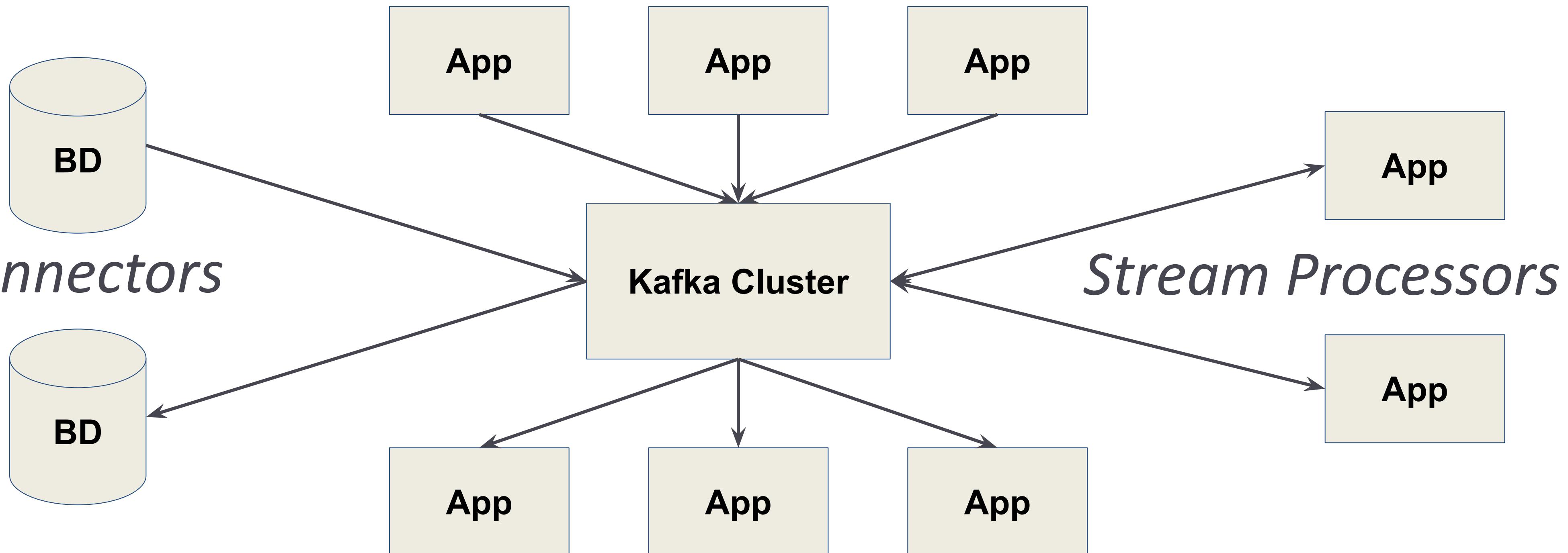
Kafka: conceitos

Producers



Kafka: arquitetura

Producers



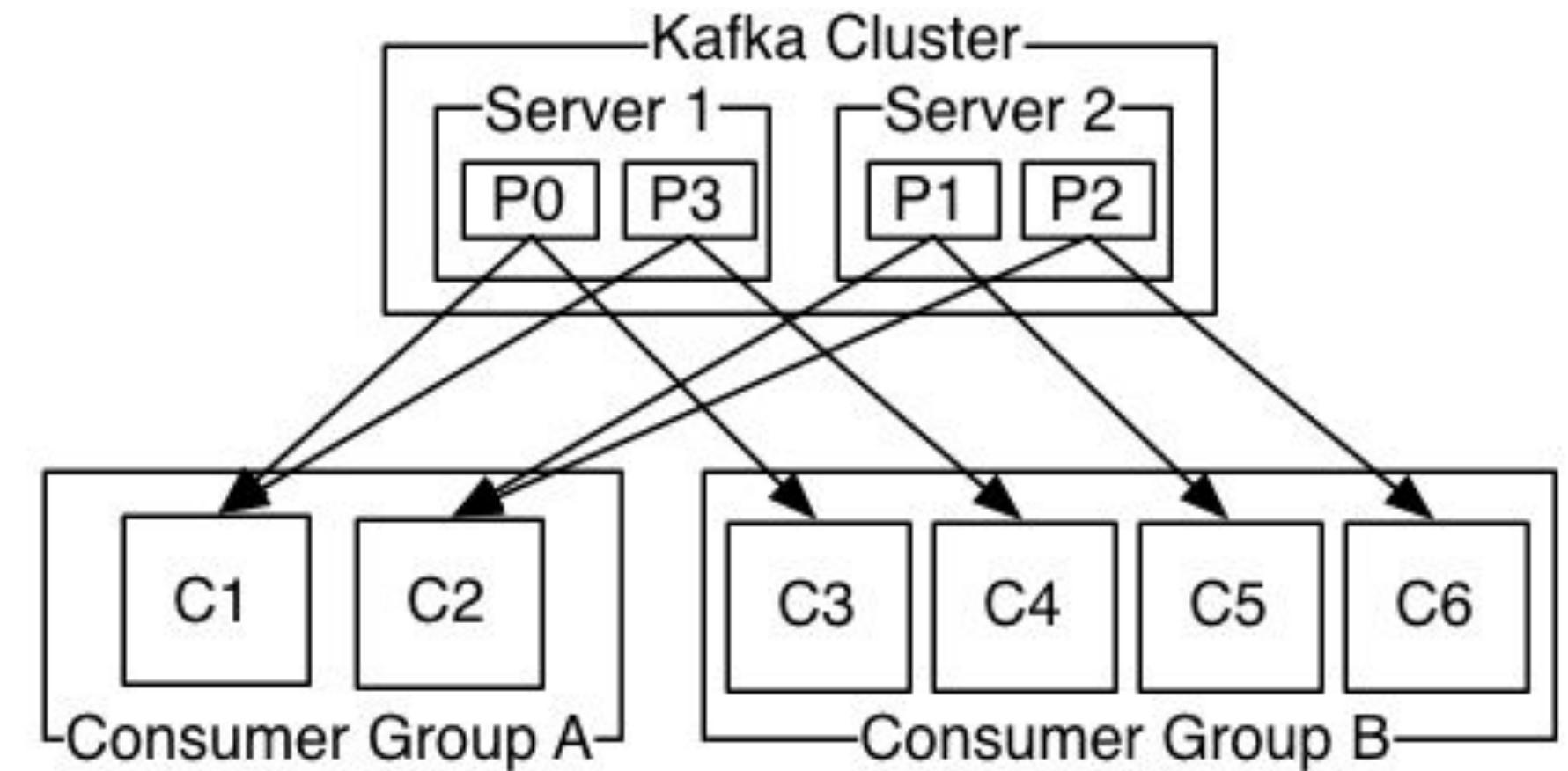
Connectors

Stream Processors

Consumers

Kafka: escalabilidade

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



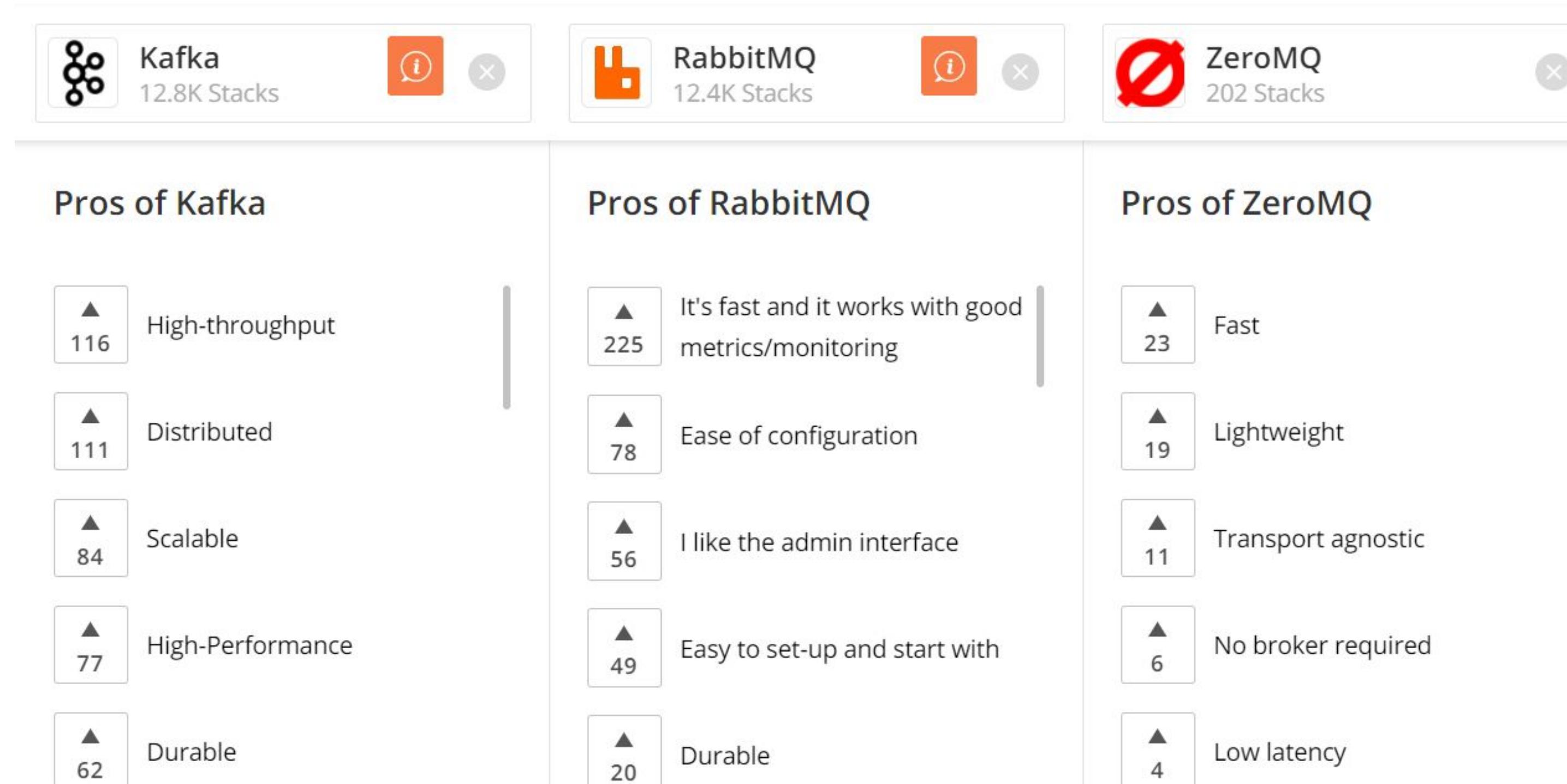
Fonte: <https://kafka.apache.org/intro.html>

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 ($>>10000$).

Kafka: quando comparado a outros concorrentes

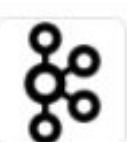
Prós



Fonte: <https://stackshare.io/stackups/kafka-vs-rabbitmq-vs-zeromq>

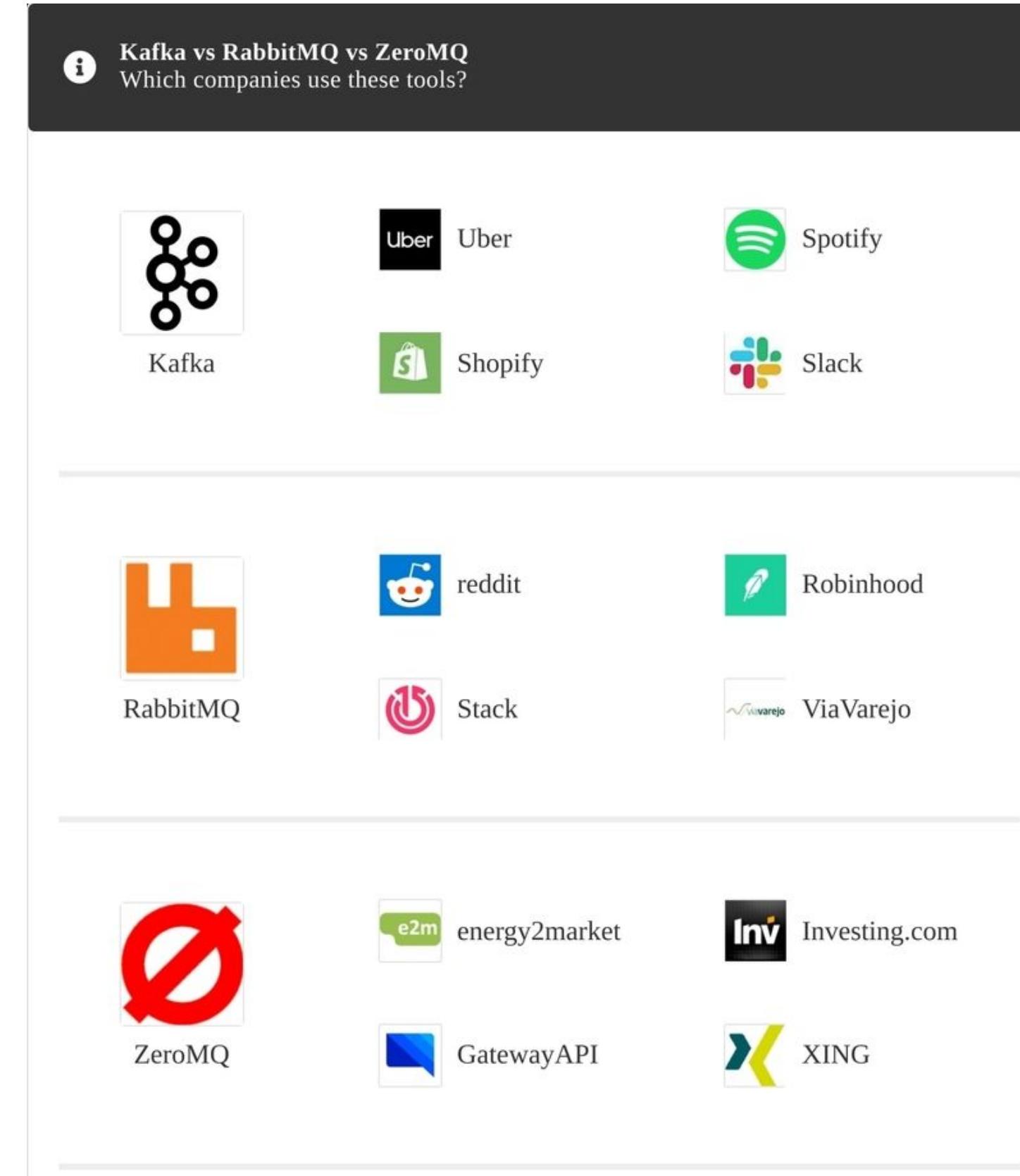
Kafka: quando comparado a outros concorrentes

Contras

 Kafka 12.8K Stacks	 RabbitMQ 12.4K Stacks	 ZeroMQ 202 Stacks
<h3>Cons of Kafka</h3> <ul style="list-style-type: none">▲ 26 Non-Java clients are second-class citizens▲ 25 Needs Zookeeper▲ 7 Operational difficulties▲ 1 Terrible Packaging	<h3>Cons of RabbitMQ</h3> <ul style="list-style-type: none">▲ 9 Too complicated cluster/HA config and management▲ 6 Needs Erlang runtime. Need ops good with Erlang runtime▲ 5 Configuration must be done first, not by your code▲ 4 Slow	<h3>Cons of ZeroMQ</h3> <ul style="list-style-type: none">▲ 5 No message durability▲ 3 Not a very reliable system - message delivery wise▲ 1 M x N problem with M producers and N consumers

Fonte: <https://stackshare.io/stackups/kafka-vs-rabbitmq-vs-zeromq>

Kafka: quando comparado a outros concorrentes



Fonte: <https://stackshare.io/stackups/kafka-vs-rabbitmq-vs-zeromq>

Instalando o Apache Kafka

Realizar o download do Apache Kafka:

```
➤ curl  
http://ftp.unicamp.br/pub/apache/kafka/2.8.1/ka  
fka_2.12-2.8.1.tgz -o ~/Downloads/kafka.tgz
```

```
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current  
                                         Dload  Upload   Total   Spent   Left  Speed  
100 62.6M  100 62.6M    0      0  528k      0  0:02:01  0:02:01 --:--:-- 345k  
posgrad@posgrad-vm:~$
```

Instalando o Apache Kafka

Criar um diretório chamado kafka e entrar nele:

- mkdir kafka
- cd kafka

Extrair os arquivos que estão na pasta download para o diretório criado:

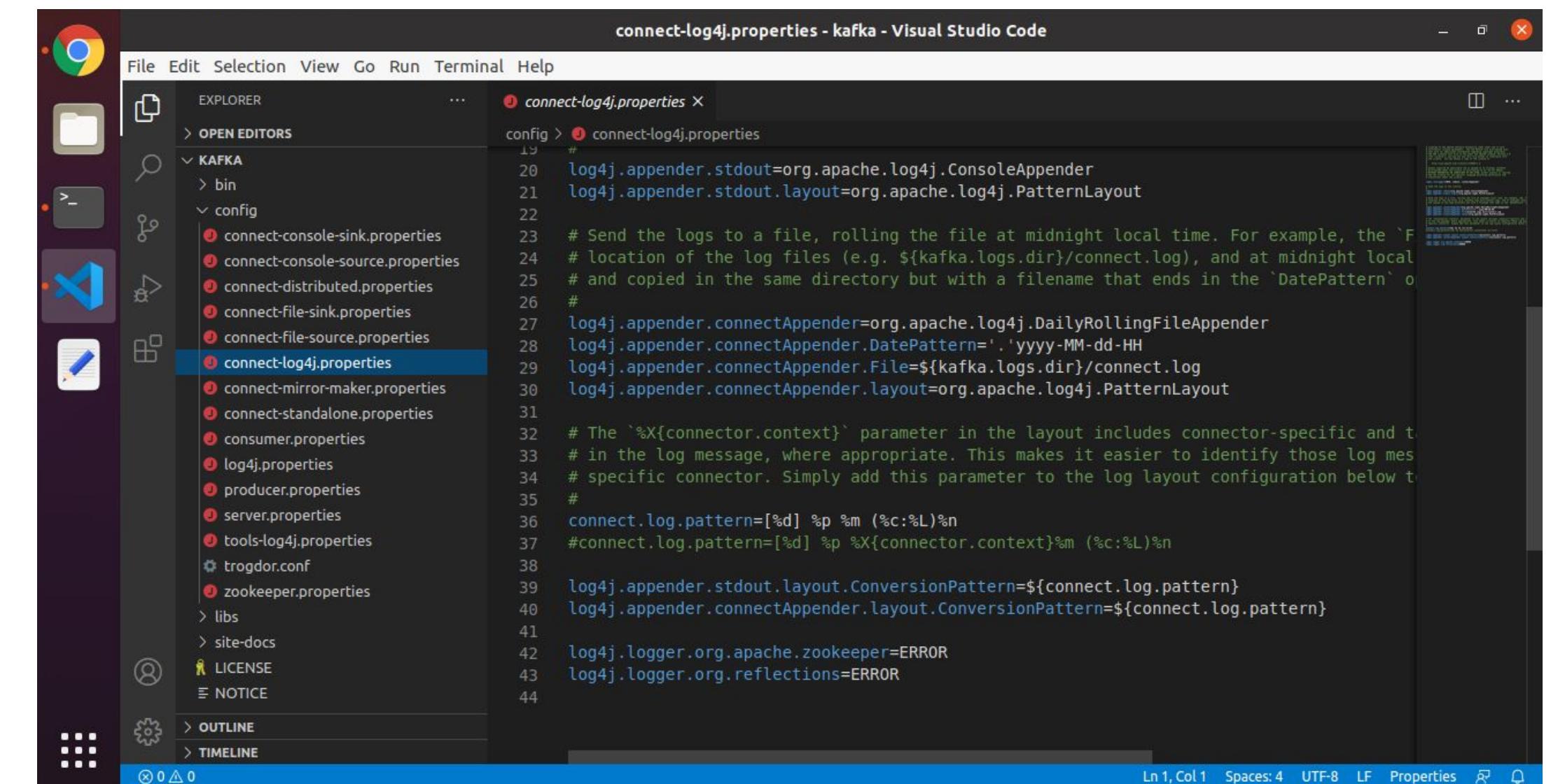
- tar -xvzf ~/Downloads/kafka.tgz --strip 1

Visualizando os arquivos do Kafka

Abrir o VS Code

Clicar em Open Folder

Selecionar a pasta kafka



The screenshot shows the Visual Studio Code interface with the title bar "connect-log4j.properties - kafka - Visual Studio Code". The left sidebar displays a file tree under the "KAFKA" folder, with "connect-log4j.properties" selected. The main editor area shows the content of the "connect-log4j.properties" file:

```
connect-log4j.properties - kafka - Visual Studio Code

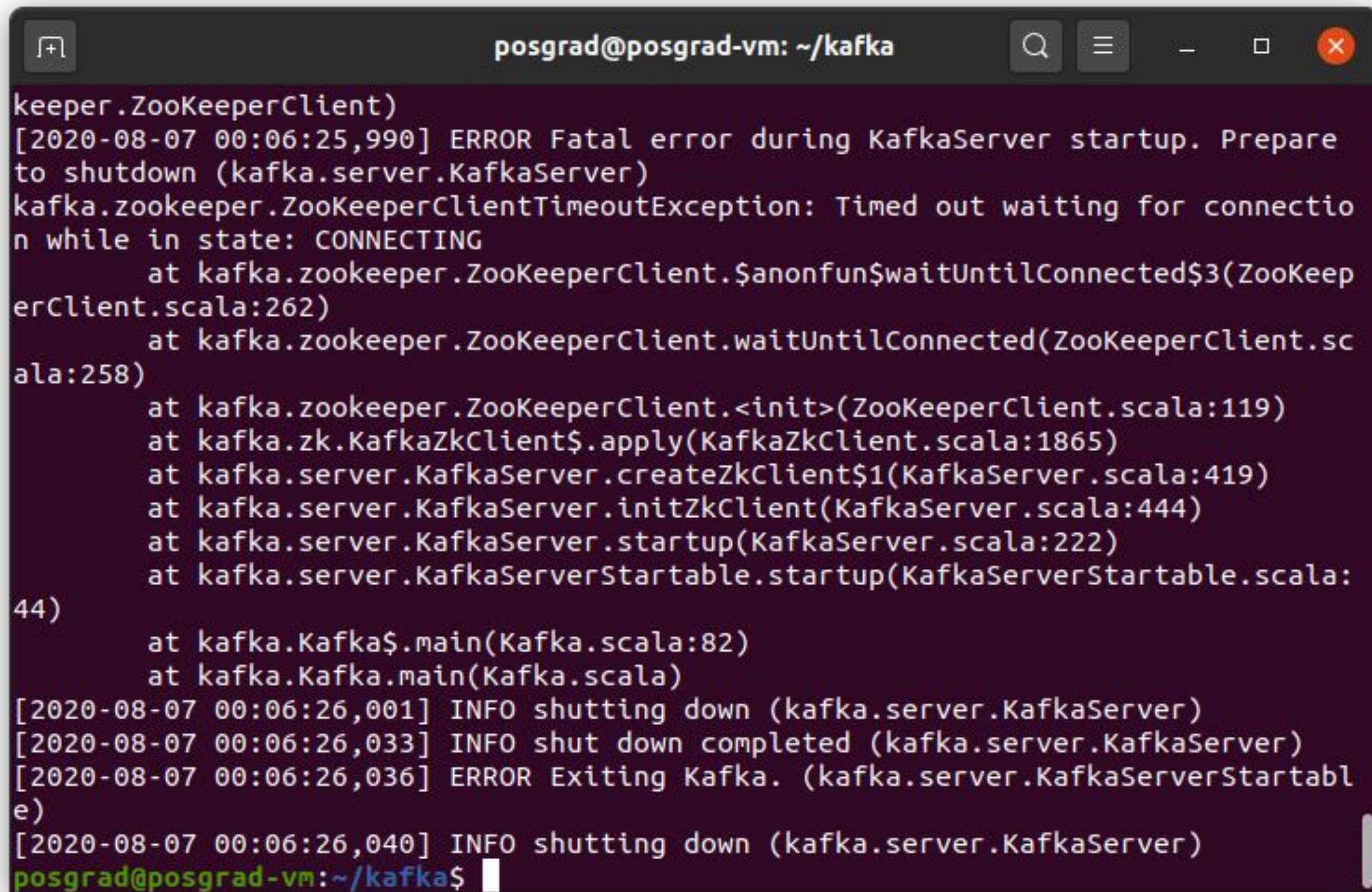
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
KAFKA
> bin
config
  connect-console-sink.properties
  connect-console-source.properties
  connect-distributed.properties
  connect-file-sink.properties
  connect-file-source.properties
  connect-log4j.properties
  connect-mirror-maker.properties
  connect-standalone.properties
  consumer.properties
  log4j.properties
  producer.properties
  server.properties
  tools-log4j.properties
  trogdr.conf
  zookeeper.properties
> libs
> site-docs
LICENSE
NOTICE
OUTLINE
TIMELINE

connect-log4j.properties ×
config > connect-log4j.properties
19 #
20 log4j.appenders.stdout=org.apache.log4j.ConsoleAppender
21 log4j.appenders.stdout.layout=org.apache.log4j.PatternLayout
22
23 # Send the logs to a file, rolling the file at midnight local time. For example, the `F
24 # location of the log files (e.g. ${kafka.logs.dir}/connect.log), and at midnight local
25 # and copied in the same directory but with a filename that ends in the `DatePattern` o
26 #
27 log4j.appenders.connectAppender=org.apache.log4j.DailyRollingFileAppender
28 log4j.appenders.connectAppender.DatePattern='yyyy-MM-dd-HH
29 log4j.appenders.connectAppender.File=${kafka.logs.dir}/connect.log
30 log4j.appenders.connectAppender.layout=org.apache.log4j.PatternLayout
31
32 # The `'%X{connector.context}'` parameter in the layout includes connector-specific and t
33 # in the log message, where appropriate. This makes it easier to identify those log mes
34 # specific connector. Simply add this parameter to the log layout configuration below t
35 #
36 connect.log.pattern=[%d] %p %m (%c:%L)%n
37 #connect.log.pattern=[%d] %p %X{connector.context}%m (%c:%L)%n
38
39 log4j.appenders.stdout.layout.ConversionPattern=${connect.log.pattern}
40 log4j.appenders.connectAppender.layout.ConversionPattern=${connect.log.pattern}
41
42 log4j.logger.org.apache.zookeeper=ERROR
43 log4j.logger.org.reflections=ERROR
44
```

The status bar at the bottom indicates "Ln 1, Col 1 Spaces: 4 UTF-8 LF Properties ⚙️ ⌂".

Inicializando o Servidor Kafka

➤ bin/kafka-server-start.sh config/server.properties



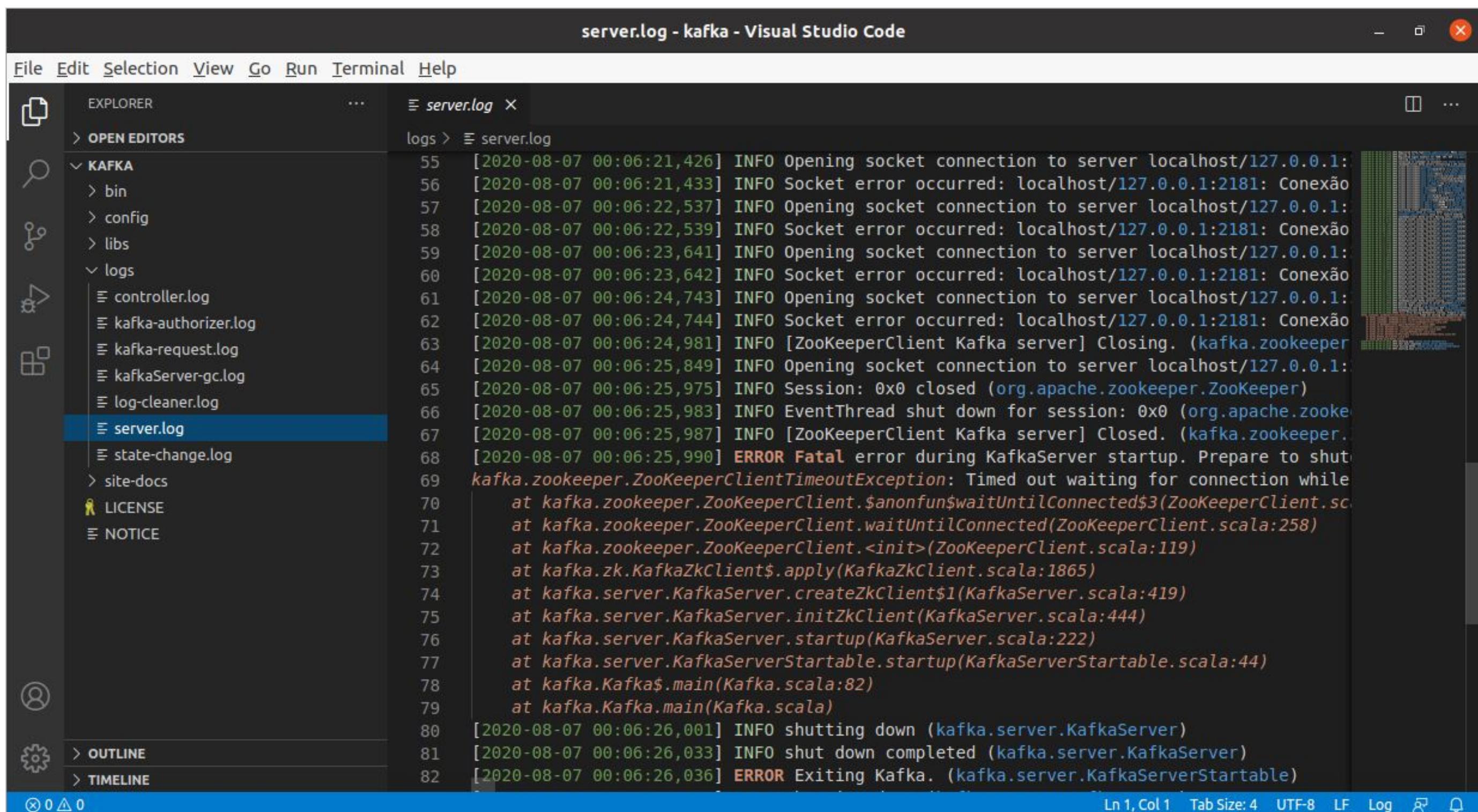
A screenshot of a terminal window titled "posgrad@posgrad-vm: ~/kafka". The window contains a stack trace and log messages. The stack trace is as follows:

```
keeper.ZooKeeperClient)
[2020-08-07 00:06:25,990] ERROR Fatal error during KafkaServer startup. Prepare
to shutdown (kafka.server.KafkaServer)
kafka.zookeeper.ZooKeeperClientTimeoutException: Timed out waiting for connectio
n while in state: CONNECTING
    at kafka.zookeeper.ZooKeeperClient.$anonfun$waitUntilConnected$3(ZooKeep
erClient.scala:262)
    at kafka.zookeeper.ZooKeeperClient.waitUntilConnected(ZooKeeperClient.sc
ala:258)
    at kafka.zookeeper.ZooKeeperClient.<init>(ZooKeeperClient.scala:119)
    at kafka.zk.KafkaZkClient$.apply(KafkaZkClient.scala:1865)
    at kafka.server.KafkaServer.createZkClient$(KafkaServer.scala:419)
    at kafka.server.KafkaServer.initZkClient(KafkaServer.scala:444)
    at kafka.server.KafkaServer.startup(KafkaServer.scala:222)
    at kafka.server.KafkaServerStartable.startup(KafkaServerStartable.scala:
44)
    at kafka.Kafka$.main(Kafka.scala:82)
    at kafka.Kafka.main(Kafka.scala)
[2020-08-07 00:06:26,001] INFO shutting down (kafka.server.KafkaServer)
[2020-08-07 00:06:26,033] INFO shut down completed (kafka.server.KafkaServer)
[2020-08-07 00:06:26,036] ERROR Exiting Kafka. (kafka.server.KafkaServerStartabl
e)
[2020-08-07 00:06:26,040] INFO shutting down (kafka.server.KafkaServer)
```

The terminal prompt "posgrad@posgrad-vm:~/kafka\$" is visible at the bottom.

Observando os logs de erro do kafka

No VS Code, abrir a pasta logs, e o arquivo server.log



The screenshot shows a Visual Studio Code window titled "server.log - kafka - Visual Studio Code". The left sidebar has "OPEN EDITORS" expanded, showing a tree view of a "KAFKA" directory containing files like "bin", "config", "libs", "logs", and several log files: "controller.log", "kafka-authorizer.log", "kafka-request.log", "kafkaServer-gc.log", "log-cleaner.log", "server.log" (which is selected and highlighted in blue), and "state-change.log". The main editor area displays the contents of the "server.log" file, which is a log of Kafka server startup. Lines 55 through 82 are shown, detailing socket connections, errors, and finally the shutdown of the Kafka server. The log ends with "shutting down" at line 80 and "shut down completed" at line 81, followed by an error message at line 82.

```
55 [2020-08-07 00:06:21,426] INFO Opening socket connection to server localhost/127.0.0.1:  
56 [2020-08-07 00:06:21,433] INFO Socket error occurred: localhost/127.0.0.1:2181: Conexão  
57 [2020-08-07 00:06:22,537] INFO Opening socket connection to server localhost/127.0.0.1:  
58 [2020-08-07 00:06:22,539] INFO Socket error occurred: localhost/127.0.0.1:2181: Conexão  
59 [2020-08-07 00:06:23,641] INFO Opening socket connection to server localhost/127.0.0.1:  
60 [2020-08-07 00:06:23,642] INFO Socket error occurred: localhost/127.0.0.1:2181: Conexão  
61 [2020-08-07 00:06:24,743] INFO Opening socket connection to server localhost/127.0.0.1:  
62 [2020-08-07 00:06:24,744] INFO Socket error occurred: localhost/127.0.0.1:2181: Conexão  
63 [2020-08-07 00:06:24,981] INFO [ZooKeeperClient Kafka server] Closing. (kafka.zookeeper.  
64 [2020-08-07 00:06:25,849] INFO Opening socket connection to server localhost/127.0.0.1:  
65 [2020-08-07 00:06:25,975] INFO Session: 0x0 closed (org.apache.zookeeper.ZooKeeper)  
66 [2020-08-07 00:06:25,983] INFO EventThread shut down for session: 0x0 (org.apache.zooke  
67 [2020-08-07 00:06:25,987] INFO [ZooKeeperClient Kafka server] Closed. (kafka.zookeeper.  
68 [2020-08-07 00:06:25,990] ERROR Fatal error during KafkaServer startup. Prepare to shut  
69 kafka.zookeeper.ZooKeeperClientTimeoutException: Timed out waiting for connection while  
70 at kafka.zookeeper.ZooKeeperClient.$anonfun$waitForConnected$3(ZooKeeperClient.sc  
71 at kafka.zookeeper.ZooKeeperClient.waitForConnected(ZooKeeperClient.scala:258)  
72 at kafka.zookeeper.ZooKeeperClient.<init>(ZooKeeperClient.scala:119)  
73 at kafka.zk.KafkaZkClient$.apply(KafkaZkClient.scala:1865)  
74 at kafka.server.KafkaServer.createZkClient$1(KafkaServer.scala:419)  
75 at kafka.server.KafkaServer.initZkClient(KafkaServer.scala:444)  
76 at kafka.server.KafkaServer.startup(KafkaServer.scala:222)  
77 at kafka.server.KafkaServerStartable.startup(KafkaServerStartable.scala:44)  
78 at kafka.Kafka$.main(Kafka.scala:82)  
79 at kafka.Kafka.main(Kafka.scala)  
80 [2020-08-07 00:06:26,001] INFO shutting down (kafka.server.KafkaServer)  
81 [2020-08-07 00:06:26,033] INFO shut down completed (kafka.server.KafkaServer)  
82 [2020-08-07 00:06:26,036] ERROR Exiting Kafka. (kafka.server.KafkaServerStartable)
```

Inicializando o Zookeeper

➤ bin/zookeeper-server-start.sh config/zookeeper.properties

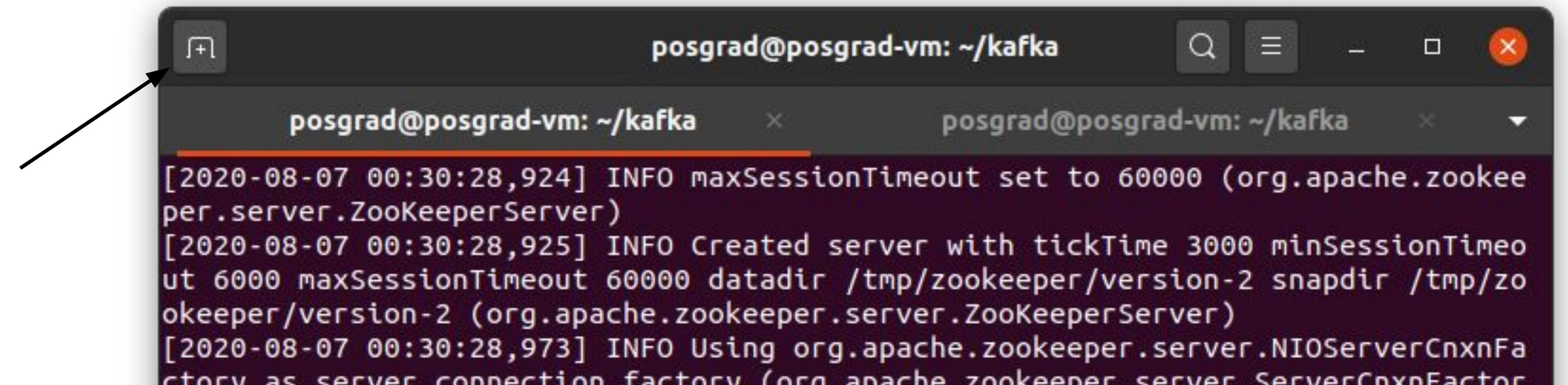
Zookeeper por padrão escuta a porta 2181

Observar arquivo ***zookeeper.properties*** na pasta ***config***

```
connect-standalone.properties      15 # the directory where the snapshot is stored.
                                16 dataDir=/tmp/zookeeper
                                17 # the port at which the clients will connect
                                18 clientPort=2181
                                19 # disable the per-ip limit on the number of connections since this is a non-production
                                20 maxClientCnxns=0
```

Inicializando o Servidor Kafka (novamente)

Em outro terminal



```
[2020-08-07 00:30:28,924] INFO maxSessionTimeout set to 60000 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-08-07 00:30:28,925] INFO Created server with tickTime 3000 minSessionTimeout 6000 maxSessionTimeout 60000 datadir /tmp/zookeeper/version-2 snapdir /tmp/zookeeper/version-2 (org.apache.zookeeper.server.ZooKeeperServer)
[2020-08-07 00:30:28,973] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.NIOServerCnxnFactory)
```

Inicie o Kafka novamente:

➤ bin/kafka-server-start.sh config/server.properties

Voilà

Zookeeper **localhost:2181**

Kafka server (broker) **localhost:9092**

Dúvidas?