



## ❖ Roteiro

- Apresentação
- Empresas
- Ferramentas
- Mão na massa
- Referências

# Apresentação

Nome: **Paulo Ricardo Ferreira**

Idade: **34 Anos**

Localidade: **Maracaí/SP**

Casado, Pai de família, Cristão

Linkedin: <https://www.linkedin.com/in/paulo-ricardo-ferreira/>

Atualmente trabalho na **Agroterenas - AGT** a 7 anos passando por alguns setores como: Agricultura de Precisão, Planejamento Agrícola e atualmente estou trabalhando como Analista de BI dentro da área de Custos e Orçamentos.

Minha jornada com dados iniciou por volta de 2017 com a necessidade de extrair dados do sistema de uma forma mais rápida e organizada.

Uma das rotinas de trabalho, era extrair informações do sistema através de relatórios em Excel, e movê-los para o local correto, que posteriormente seriam utilizados em painéis feitos Excel/Power BI, e buscando como melhorar/agilizar iniciei as buscas sobre o assunto e encontrei o Pentaho.

O Pentaho foi minha primeira ferramenta para tratamento de dados onde iniciei com pequenas tarefas e aos poucos fui evoluindo o processo.

Passei por diversas etapas até chegar nas tecnologias que uso atualmente, desde a utilização em um notebook, até o uso máquina virtual com scripts de agendamento, que posteriormente foi migrado para uso Docker para gerenciar os serviços, Integração Contínua com GitLab, Agendamento Apache Airflow e Apache Hop como orquestrador de dados

# Empresas



A **Linux Foundation** (LF) é uma [organização sem fins lucrativos](#) voltada para o desenvolvimento e aprimoramento de tecnologias de software livre e de código aberto (*open source*, na sigla em inglês). O principal projeto mantido pela Linux Foundation é o [Kernel Linux](#).

Fundada em 2000 pela fusão do Open Source Development Labs (OSDL) e o Free Standards Group (FSG), a Linux Foundation patrocina o trabalho do criador do Linux, Linus Torvalds, e é apoiada pelas empresas parceiras no desenvolvimento do Linux (aliança de desenvolvedores), que possui entre elas: [Canonical](#), [Red Hat](#), [Google](#), [Intel](#), [AMD](#), [Autodesk](#), [Petrobras](#), [Philips](#), [Samsung](#), [IBM](#), [Adobe](#), entre outras [empresas públicas](#), [privadas](#), organizações sem fins lucrativos, [universidades](#) e desenvolvedores independentes do todo o mundo.<sup>[1][2]</sup>

A Linux Foundation promove, protege, e padroniza Linux, "fornecendo um conjunto abrangente de serviços para competir eficazmente com plataformas fechadas".



A **Apache Software Foundation** (ASF) conhecida também apenas como **Apache Foundation** ou **Fundação Apache** é uma [organização sem fins lucrativos](#) criada para suportar os projetos de código aberto, principalmente os *Apache*, incluindo o [servidor web Apache HTTP Server](#).

A ASF é uma comunidade descentralizada de desenvolvedores de [software](#). Os *softwares* criados pela fundação são distribuídos sob a [licença Apache](#) e são conhecidos como [software livre](#) ou *open source software*. Os projetos Apache são caracterizados por um processo colaborativo e consensual e por uma licença aberta e pragmática. Os projetos são gerenciados por pessoas que são escolhidas, dentre os técnicos que contribuem mais ativamente, por todos participantes do projeto. A ASF é uma [meritocracia](#), isto é, para ser membro da fundação, o voluntário deve ter participado ativamente de projetos Apache.

Um dos objetivos da ASF é proteger legalmente os participantes dos seus projetos, e prevenir que o nome *Apache* seja utilizado por outras organizações sem a devida permissão.

Entre os seus integrantes, estão o chairman Greg Stein, os desenvolvedores Ken Coar, J. Aaron Farr, Cliff Schmidt, entre muitos outros. É mantida principalmente por doações e contando com o apoio de grandes corporações, como [IBM](#) e [Sun](#), tanto no que diz respeito ao desenvolvimento de produtos, quanto no fornecimento de [hardware](#) ou até mesmo no aspecto financeiro.

Contribuídores: <https://www.apache.org/foundation/thanks>

Fonte: [https://pt.wikipedia.org/wiki/Apache\\_Software\\_Foundation](https://pt.wikipedia.org/wiki/Apache_Software_Foundation)

# Ferramentas



# Visão Geral

## Ferramentas

Apache *HOP*



Orquestrador de  
Dados



Apache  
Airflow



Orquestração de  
Execução



PostgreSQL



Banco de Dados



## Tecnologias



docker



# Linux

# O que é o sistema Operacional Linux?



Linux é um Sistema Operacional, assim como o Windows e o Mac OS, que possibilita a execução de programas em um computador e outros dispositivos. Linux pode ser livremente modificado e distribuído.

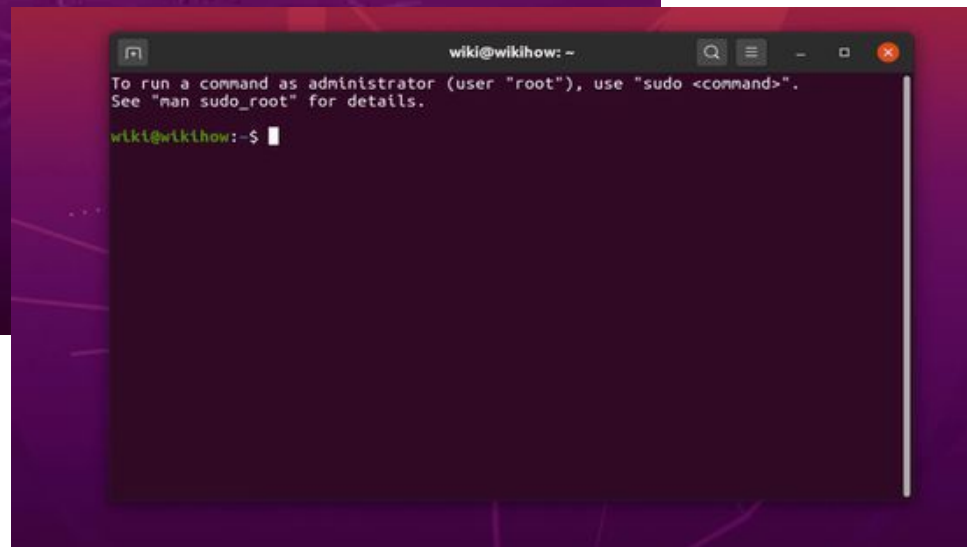
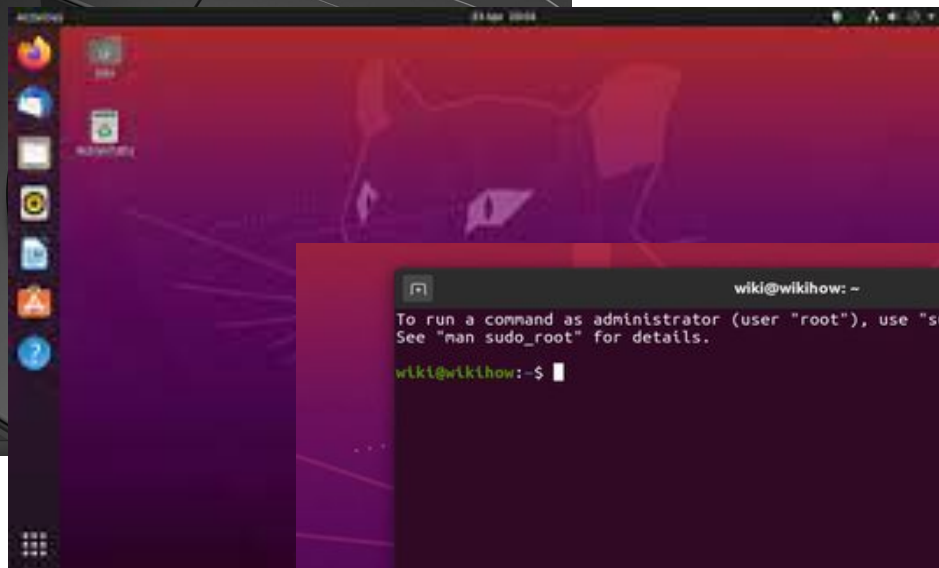
Apesar desta interpretação ser simplista é perfeitamente correta e aceitável. Mas, em uma definição mais profunda e técnica, Linux é o nome dado apenas ao núcleo do sistema operacional, chamado de Kernel.

Kernel é um conjunto de instruções que controla como será usado o processador, a memória, o disco e dispositivos periféricos. É o software presente em todo sistema operacional que determina como o computador deve funcionar. O Kernel Linux foi criado pelo Linus Torvalds, com a primeira versão oficial lançada em 1991.

O Kernel por si só não tem utilidade prática. É preciso uma série de programas adicionais para seu uso efetivo, como interpretadores de comandos, compiladores para que seja possível o desenvolvimento de novos programas, editores de textos e assim por diante.

Sobre o Kernel Linux, empresas como Canonical, RedHat entre outras, desenvolvem sistemas operacionais sobre o Kernel Linux





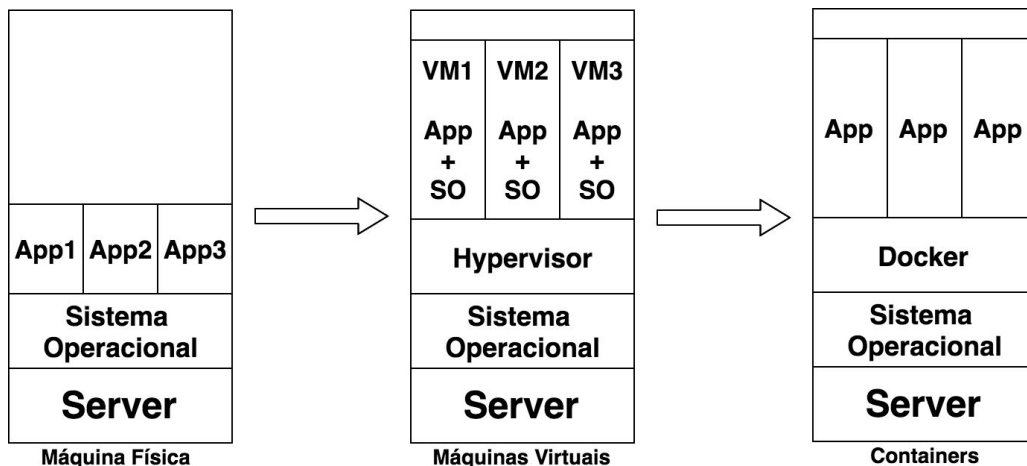
# Docker

# O que é Docker?



De forma bem resumida, podemos dizer que o Docker é uma plataforma aberta, criada com o objetivo de facilitar o desenvolvimento, a implantação e a execução de aplicações em ambientes isolados. Foi desenhada especialmente para disponibilizar uma aplicação da forma mais rápida possível.

## Comparativo entre Máquina Física / VM / Container



Fonte: <https://github.com/badtuxx/DescomplicandoDocker/blob/main/media/image3.png>

Fonte: <https://stack.desenvolvedor.expert/appendix/docker/oquee.html>

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
9fae634671f3	hop-server	1.57%	5.846GiB / 7.741GiB	75.51%
333a0c319b60	airflow-scheduler	14.05%	423.2MiB / 7.741GiB	5.34%
86bad0ccdaa5	airflow-webserver	0.07%	94.11MiB / 7.741GiB	1.19%
9c901a17844c	airflow-postgres	1.30%	33.58MiB / 7.741GiB	0.42%

Sistema Operacional Linux Ubuntu

Processador com 4 Cores

8 GB de Memória Ram

Mínimo de 60 GB de HD

\*Estes requisitos é para manter os serviços ligados 24/7 em uma VM, considerando que o banco de dados esteja em outros local

# Apache Hop





## Como surgiu o Apache Hop e o Que Ele Oferece?

O Apache Hop é uma ferramenta de integração de dados de código aberto, que é um fork do Pentaho Data Integration (PDI) ou Kettle.

Ele oferece uma ferramenta de desenvolvimento visual que pode tornar os Engenheiros e Arquitetos de Dados mais produtivos, principalmente os que preferem construir seus pipelines sem escrever nenhum código.

## Componentes do Apache Hop

O Hop tem três componentes principais a seguir:

**Hop GUI:** É um editor de interface gráfica para construção de pipelines (transformações) e fluxos de trabalho (jobs). Ele permite que você crie tarefas complexas de ETL (Extract, Transformation, Load) sem escrever nenhum código. Ele fornece uma interface de arrastar e soltar que permite criar, editar, executar ou depurar um pipeline ou fluxo de trabalho.

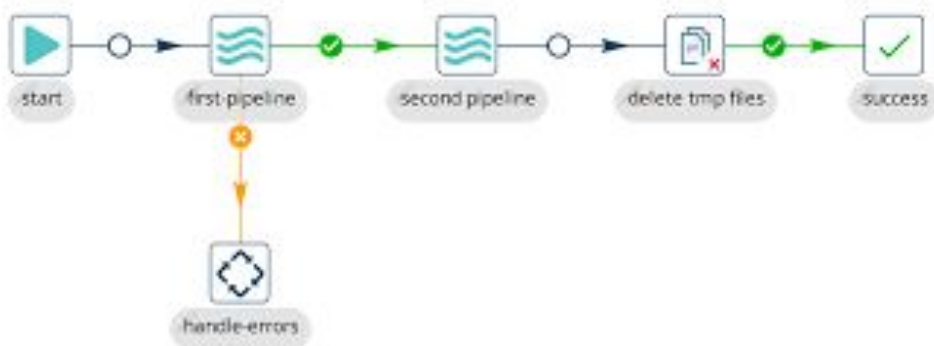
**Hop Run:** É um utilitário CLI (Command Line Interface) autônomo que pode ser usado para executar pipeline e fluxo de trabalho.

**Hop Server:** É um contêiner web leve que permite executar o pipeline e o fluxo de trabalho em um servidor remoto e que pode ser implantado em vários servidores. Ele também fornece uma API REST para invocar remotamente seu fluxo de trabalho e pipeline.

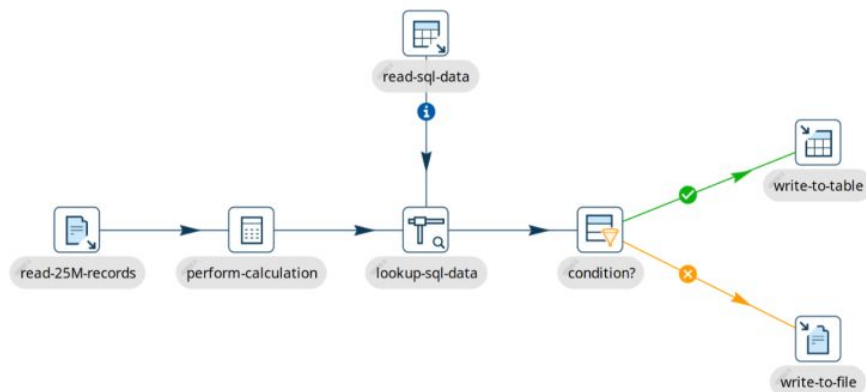
Fonte: <https://www.cienciaedados.com/pipelines-de-dados-com-apache-hop/>

## Exemplos Apache Hop / Pentaho

### Workflow / Job



### Pipeline / Transformer



# Apache Airflow

# O que é o Apache Airflow



O projeto do Apache Airflow se iniciou como um piloto dentro do Airbnb, em 2015, e desde então vem sendo adotado pelas maiores empresas do mundo todo, se tornando hoje a principal referência em ferramentas de orquestração no universo de dados. No final de 2020, a versão 2.0 foi oficialmente lançada, mostrando a maturidade do projeto e contínua evolução com apoio da comunidade, incluindo diversas melhorias tanto na experiência do usuário (melhorias na *UI*, *Task Flow API*, etc.), quanto em **segurança e infraestrutura**.

## Principais conceitos para entender o Apache Airflow

Para um entendimento rápido de como funciona a arquitetura dos fluxos no Airflow, podemos focar nos principais componentes:

**DAGs:** Abreviação de Direct Acyclic Graph, é a estrutura principal que representa um fluxo de dados. Poderia ser equivalente a um pipeline de dados. Geralmente, dentro de uma empresa, teremos várias DAGs, e cada uma terá uma função específica e geralmente independente (ex: Pipeline Dados A, Pipeline Dados B).

**Tasks:** Tarefas que serão executadas dentro da sua DAG. Uma DAG pode ter uma ou várias tasks atreladas. Alguns exemplos de tarefas são: execuções de scripts em Python, Bash, Spark, entre outras. Para que possamos escolher qual tipo de tarefa a Task irá executar e seus parâmetros, devemos atribuir um Operator para ela. As dependências entre uma task e outra são declaradas via script de maneira prática, formando a lógica do fluxo da DAG.

**Operators:** São os componentes pré-definidos (template) para executar as Tasks. Os componentes mais comuns são o BashOperator, PythonOperator, EmailOperator, entre outros. Existem de centenas a milhares de componentes das mais diversas tecnologias, prontos para serem utilizados. Porém, é possível desenvolver operadores personalizados caso necessário.

**Executor:** É o mecanismo que será responsável por executar as Tasks – o motor de execução. O Apache Airflow só poderá ter um executor definido para seu ambiente. Alguns exemplos são: KubernetesExecutor, SequentialExecutor e **LocalExecutor**

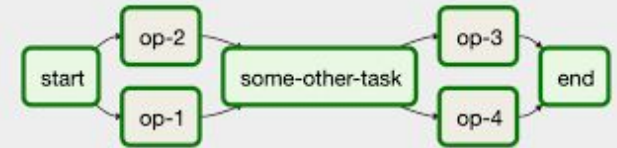
**Scheduler:** Um dos principais componentes do Airflow, é o responsável por monitorar as execuções das DAGs e iniciar novas tarefas quando elas estiverem disponíveis, nos horários estimados.

## Tela Principal

### DAGs

All 26		Active 10	Paused 16	Filter DAGs by tag	Search DAGs		
DAG	Owner	Runs	Schedule	Last Run	Recent Tasks	Actions	Links
<input checked="" type="checkbox"/> example_bash_operator example example2	airflow	2	0 0 ***	2020-10-26, 21:08:11	9	<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input checked="" type="checkbox"/> example_branch_dop_operator_v3 example	airflow		* / 1 * * * *			<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input type="checkbox"/> example_branch_operator example example2	airflow	1	@daily	2020-10-23, 14:09:17	11	<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input checked="" type="checkbox"/> example_complex example example2 example3	airflow	1	None	2020-10-26, 21:08:04	37	<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input checked="" type="checkbox"/> example_external_task_marker_child	airflow	1	None	2020-10-26, 21:07:33	2	<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input checked="" type="checkbox"/> example_external_task_marker_parent	airflow	1	None	2020-10-26, 21:08:34	1	<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input checked="" type="checkbox"/> example_kubernetes_executor example example2	airflow		None			<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input checked="" type="checkbox"/> example_kubernetes_executor_config example3	airflow	1	None	2020-10-26, 21:07:40	5	<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input checked="" type="checkbox"/> example_nested_branch_dag example	airflow	1	@daily	2020-10-26, 21:07:37	9	<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	
<input type="checkbox"/> example_passing_params_via_test_command example	airflow		* / 1 * * * *			<a href="#">▶</a> <a href="#">🔄</a> <a href="#">🗑️</a> ...	

### DAG e suas Task's



Mã~o na massa

# Referências



Rafael Arruda -> <https://www.youtube.com/@arrudaconsulting>  
Cloud, Pentaho, Datawarehouse

Apache Hop -> <https://www.youtube.com/@ApacheHop>  
Canal Oficial Apache Hop

Apache Airflow -> <https://www.youtube.com/@MarcLamberti>  
Divulgador Apache Airflow

Engenharia de Dados -> <https://www.youtube.com/@LuanMorenoMMacie>  
BigData, Engenharia de Dados e o que o mercado tem praticado

Linux Tips -> <https://www.youtube.com/@LinuxTips>  
Docker, Kubernetes, DevOps

Fabricao Veronez -> <https://www.youtube.com/@fabricaoveronez>  
Docker, Kubernetes, DevOps

Full Cycle -> <https://www.youtube.com/@FullCycle>  
Docker, Kubernetes, DevOps

Damavis -> <https://github.com/damavis/airflow-hop-plugin>  
Plugin Apache Airflow para Apache Hop

Vagner Fonseca -> <https://www.youtube.com/@FonsecaVagner>  
Linux

Obrigado a Todos