

# Apache Hop

Aplicação de testes integrados a pipeline de dados com Apache Hop.



THE  
DEVELOPER'S  
CONFERENCE

Marcio Junior Vieira  
CEO & Data Scientist, Ambiente Livre

## Mini-cv

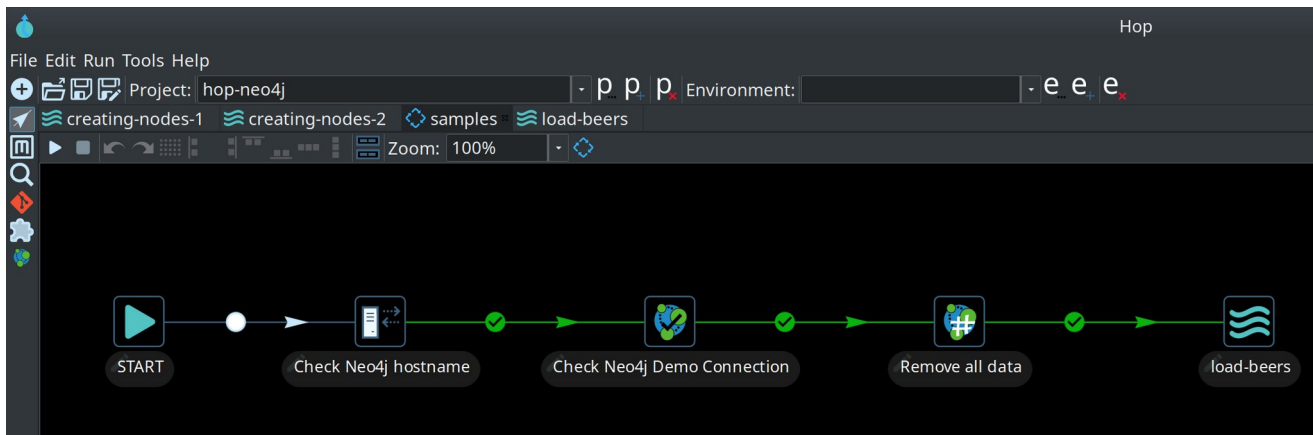
- 22 anos de experiência em TI, vivência em desenvolvimento, análise e ciência de dados.
- CEO da Ambiente Livre atuando como Cientista de Dados, Engenheiro de Dados e Arquiteto de Software.
- Professor dos MBAs em Big Data & Data Science, Inteligência Artificial e BI da Universidade Positivo.
- Professor do MBA Artificial Intelligence e Machine Learning da FIAP.
- Pesquisador do Laboratório de tecnologias de tomada de decisão da Universidade de Brasília (UNB/Latitude).
- Trabalhando com Free Software e Open Source desde 2000 com serviços de consultoria e treinamento.
- Graduado em Tecnologia em Informática(2004) e pós-graduado em Software Livre(2005) ambos pela UFPR.
- Palestrante FLOSS em: FISL, TDC, Latinoware, Campus Party, Pentaho Day, Ticnova, PgDay e FTSL.
- Organizador Geral: Pentaho Day 2017, 2015, 2019 e apoio nas ed. 2013 e 2014.
- Data Scientist, instrutor e consultor de Big Data e Data Science com tecnologias abertas.
- Ajudou a capacitar equipes de Big Data na IBM, Accenture, Tivit, Serpro, Natura, MP, Netshoes, Embraer, etc.
- Especialista em implantação e customização de Big Data com Hadoop, Spark, Pentaho, Cassandra e MongoDB.
- Contribuidor de projetos internacionais, tais como Apache Hop, Pentaho, LimeSurvey, SuiteCRM e Camunda.
- Especialista em implantação e customização de ECM com Alfresco e BPM com Activiti, Flowable e Camunda.
- Certificado (Certified Pentaho Solutions) pela Hitachi Vantara (Pentaho).
- Membro da The Order Of de Bee (comunidade Alfresco para desenvolver o ecossistema Alfresco independente).
- Premio: Camunda Champions 2022 (entre 35 no mundo e 4 Brasileiros).

# Nosso Ecossistema de Serviços

Big Data e Data Science	CRM e CMS	ECM e BPM	Business Intelligence
Análise de Dados da IoT Análise Preditiva Processamento Distribuído Banco de Dados Colunares  Big Data & Data Lake Big Data Analytics Machine Learning  Consultoria   Treinamento   Projeto	Marketing e Vendas Fidelização SAC e Pós-vendas Portais de Conteúdo  Customer Relationship Management Content Management System Pesquisa de Mercado & SLA  Consultoria   Treinamento   Projeto	Gestão de Documentos Gerenciamento de Mídias Processo de Negócio BPMN e BPMS  Enterprise Content Management Records Management Business Process Management  Consultoria   Treinamento   Projeto	Painéis de Indicadores Cubos de Análise Relatórios Gerenciais Tomada de Decisão  Business Intelligence & Analytics Dashboards e OLAP Data Integration & Data Mining  Consultoria   Treinamento   Projetos

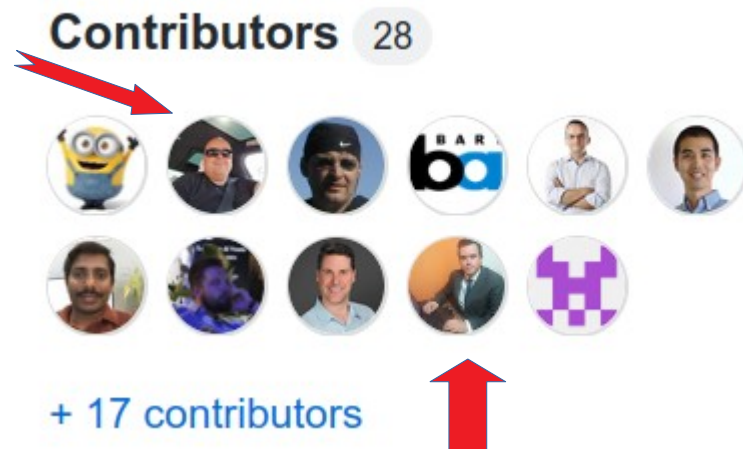


- Acronimo de: **H**op **O**rquestration **P**lataform
- Orquestração
  - **Dados** – Pipelines e Workflows.
  - **Metadata** – Edição, Manuseio e gerenciamento.
  - **Insights**: Execução e tratamento de dados, log do processo.
- - **Configurações**: Manuseio de ecossistemas complexos.



## Background

- Iniciativa de uma Comunidade (junto com Matt Casters)
- Fork do Kettle/Pentaho/PDI 8.2 + WebSpoon + Patches + plugins... (Somados são mais de 20 anos de desenvolvimento de software)
- Interface gráfica renovada.
- Back-end de metadados novo.
- Maior simplicidade.
- Muitos códigos refatorados.
- Licença Apache v2.0



## Apache Incubation

- Versão 1.0 lançada após 2 anos de refatoração e implementações.
- Atualmente estamos na versão 2.0.
- O projeto está encubado na Fundação Apache.
- Fontes em <https://github.com/apache/incubator-hop>
- Website: <https://hop.apache.org>








## Definição

- 491 Projetos Open Source.
- +7000 Committers, e com uma média de 50 novos mensais... Seja um!
- Data Science = Apache = Open Source
- **Apache é líder em Big Data e Data Science!**
- ~49 projetos da linha “Big Data” incluindo “Apache Hadoop” e “Spark”
- ~25 projetos de database incluindo “Apache Cassandra”



### • testing (5):

-  [Apache AntUnit](#)
-  [Apache Gump](#)
-  [Apache Jakarta Cactus \(in the Attic\)](#)
-  [Apache JMeter](#)
-  [Apache Yetus](#)

## Funcionalidades Tradicionais

- Usadas em projetos de data warehouse e data lakes.
- Usado para projetos de engenharia de dados.

## Funcionalidades Adicionais

- Migração de dados entre aplicações/banco de dados
- Exportar dados de banco de dados para arquivos texto
- Carregar massivamente dados em banco de dados
- Data Cleansing – disciplina de qualidade/limpeza de dados de data warehouse
- Integração de aplicações.
- Gerenciamento de Filesystem (File management )



## Funcionalidades

- Geração de Metadata: Sem Codificação (Low-code).
- Modular, plugável ou embutido.
- Rápido Start!
- Apache Beam com suporte a Apache Spark, Flink e GCP Dataflow.
- Pronto para uso com ferramentas simples.
- **Testes Integrados.**
- VFS File Systems (Local, AWS S3, Azure Blob, DropBox, Google Cloud Storage, Google Driver, HDFS, FTP, WebDav, RAM).



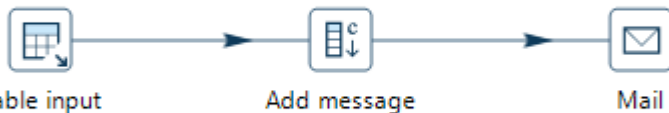
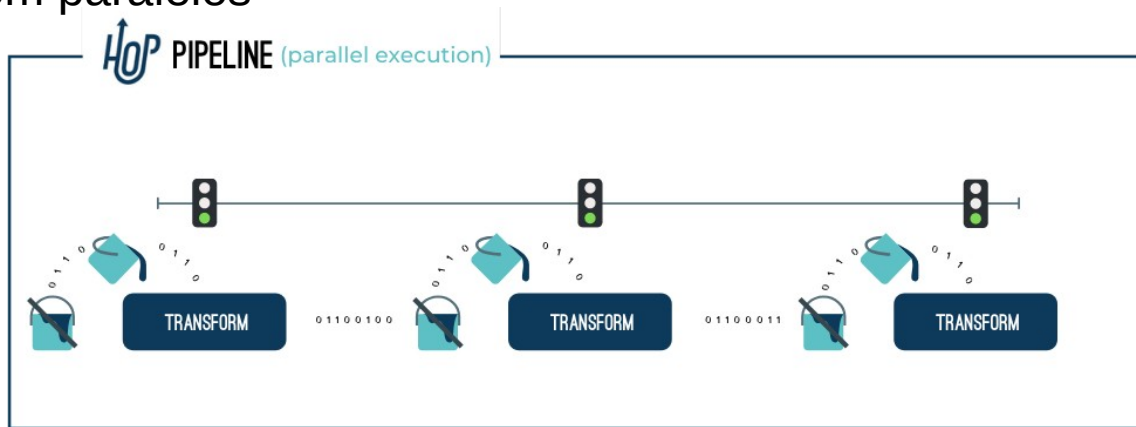
## Princípios

- Fácil.
- Rápido.
- Transparente.
- Inovador.
- Implementa melhores praticas.



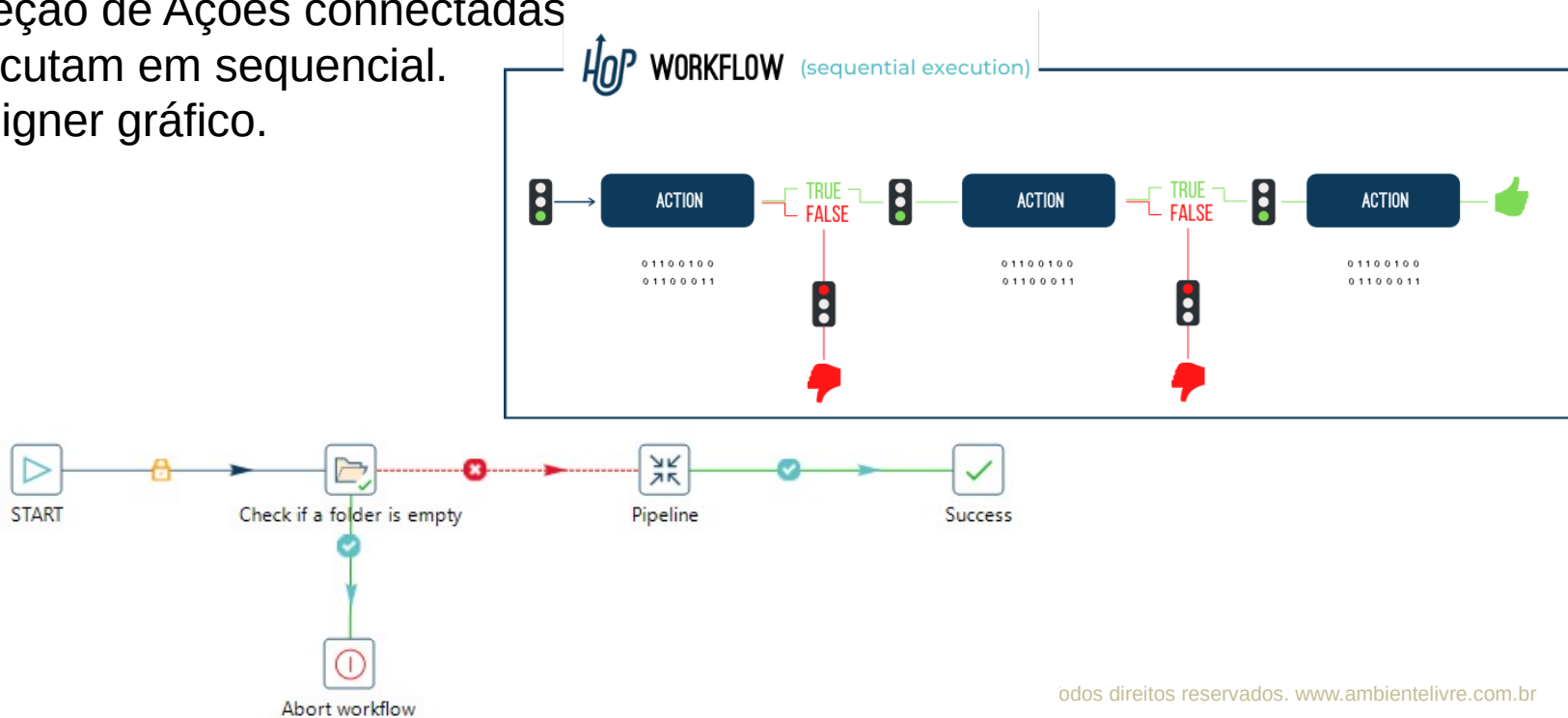
## Pipelines

- Coleção de transformações de dados conectadas.
- Execução de todos pipelines em paralelos
- Designer gráfico.



## Pipelines

- Coleção de Ações conectadas
- Executam em sequencial.
- Designer gráfico.

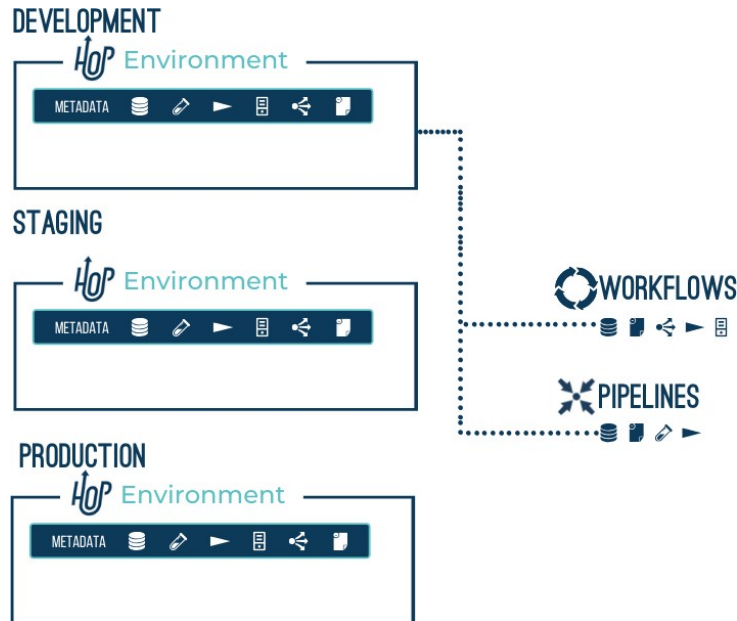


## Projetos

- Agrupamento de pipelines e workflow e configurações.

## Ambiente

- Instâncias de projetos  
(DEV, PROD, HOMOLOG, etc)



## Tools

- **Hop GUI:** Desenvolvimento e Design de workflow e pipelines.
- **Hop Conf:** Gerenciamento de variáveis e aspectos configuráveis.
- **Hop Encrypt:** Controle criptografados de dados de dados sensíveis (senhas, etc)
- **Hop Run:** executa pipelines e workflow por linha de comando e agendamento (cron, apache Airflow, schedules windows, etc).
- **Hop Search:** Busca em metadados do projeto.
- **Hop Server:** Interface web para gerenciamento de pipelines e workflows.
- **Hop Translator:** interface para não-tecnicos internacionalizar a ferramenta.

## Motivação

- Construir pipelines de dados é apenas o começo.
- Você deseja executar seus fluxos de trabalho e pipelines na produção de forma confiável e deseja garantir que seus dados sejam processados exatamente da maneira que você deseja.
- Esta última parte é onde o **teste unitário** entra.
- O teste permite que você crie projetos de engenharia de dados mais confiáveis.

## Unit Test

- Para testar se seus pipelines estão processando seus dados exatamente da maneira que você espera, o Apache Hop compara os dados gerados por um teste de unidade executado com um resultado conhecido.
- Esse resultado conhecido, também conhecido como **Golden Dataset**, é um conjunto de dados que foi adicionado ao seu projeto com resultados corretos garantidos.
- Quando os resultados produzidos pelo teste de unidade de pipeline correspondem exatamente ao Golden Dataset, o teste é aprovado. Se houver alguma diferença, o teste falha.
- Há uma série de ajustes que você pode aplicar aos seus unit tests



## Podem acelerar o desenvolvimento em vários casos

- Pipelines sem entrada de tempo de design: mapeamentos, thread único, etc.
- Quando os dados de entrada ainda não existem, estão em desenvolvimento ou não há acesso direto ao sistema de origem.
- Quando leva muito tempo para obter dados de entrada, consultas de execução longa, etc.

## conceitos (objetos de metadados)

- **Dataset** : um conjunto de linhas com um determinado layout, armazenado em um conjunto de dados CSV. Quando usado como entrada, chamamos de conjunto de dados de entrada. Quando usado para validar a saída de uma transformação, chamamos de conjunto de dados dourado .
- **Unit test**: a combinação de conjuntos de dados de entrada, conjuntos de dados golden, ajustes e um pipeline.
- **Unit test tweak**: a capacidade de remover ou ignorar uma transformação durante um teste.
- Você pode ter 0, 1 ou mais conjuntos de dados de entrada golden definidos em um unit test, assim como você pode ter vários unit tests definidos por pipeline.

## Runtime

- Quando um pipeline é executado no Hop GUI e um unit test é selecionado, acontece o seguinte:
  - ➔ Todas as transformações marcadas com um conjunto de dados de entrada são substituídas por uma transformação de Injetor.
  - ➔ Todas as transformações marcadas com um dataset golden são substituídas por uma transformação fictícia.
  - ➔ Todas as transformações marcadas com um ajuste "Bypass" são substituídas por uma transformação fictícia.
  - ➔ Todas as transformações marcadas com um ajuste "Remover" são removidas

## Steps de Unit Test

- A categoria 'Unit Test' na transformação contém as seguintes opções::



- ➔ **Set input data set:** para o unit test ativo, define qual dataset usar em vez da saída da transformação.
- ➔ **Clear input data set:** Remove um dataset de entrada definido deste unit test da transformação.
- ➔ **Set golden data set:** a entrada para esta transformação é obtida e comparada com o dataset golden que você está selecionando.
- ➔ **Clear golden data set:** Remova um dataset de entrada definido para este unit test de transformação.

## Steps de Unit Test

- ➔ **Create data set:** crie um dataset vazio com os campos de saída desta transformação.
- ➔ **Write rows to data set:** Execute o pipeline atual e grave os dados em um Dataset.
- ➔ **Remove from test:** quando este unit test for executado, não inclua esta transformação.
- ➔ **Include in test:** Execute o pipeline atual e grave os dados em um dataset.
- ➔ **Bypass in test:** quando este unit test for executado, ignore esta transformação (substitua por um dummy)
- ➔ **Remove bypass in test:** Não ignora esta transformação no pipeline atual durante o teste.

## Para Desenvolvedores do Hop

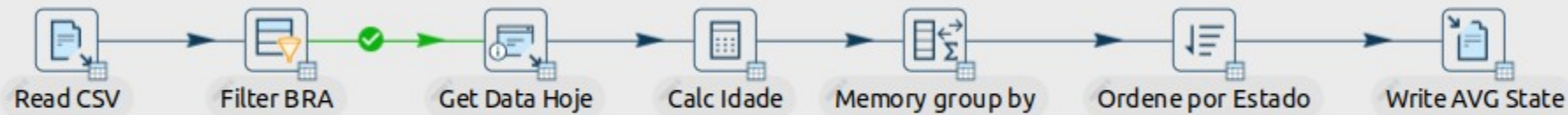
- Executa centenas de testes validando as funcionalidades do Apache Hop.
- Utilizado para validar a Própria Ferramenta.
- Integração com Unit Test do Maven.

## Passos do Live Demo

- Criar os conjuntos de dados.
- Gravar dados nos conjuntos de dados.
- Criar o Unit Test.
- Definir dataset de entrada Golden.
- Execute o Unit Test.

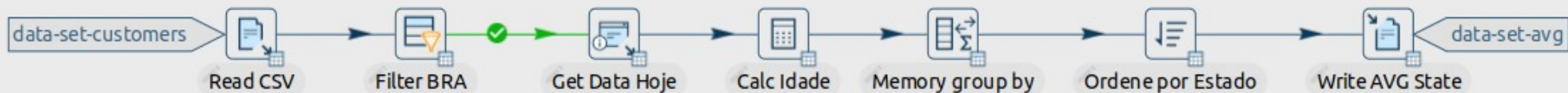
# Pipeline Demo.

## Gera Media de Idade de Cliente do Brasil por estado.





# Pipeline com Unit Test



# Pipeline com Unit Test - Success

1≡ Transform Metrics 2≡ Logging



```
2022/12/07 14:53:49 - pipeline_clientes - Executing this pipeline using the Local Pipeline Engine with run configuration 'local'
2022/12/07 14:53:49 - pipeline_clientes - Execution started for pipeline [pipeline_clientes]
2022/12/07 14:53:49 - Read CSV.0 - Finished processing (I=0, O=0, R=6, W=6, U=0, E=0)
2022/12/07 14:53:49 - Filter BRA.0 - Finished processing (I=0, O=0, R=6, W=5, U=0, E=0)
2022/12/07 14:53:49 - Get Data Hoje.0 - Finished processing (I=0, O=0, R=5, W=5, U=0, E=0)
2022/12/07 14:53:49 - Calc Idade.0 - Finished processing (I=0, O=0, R=5, W=5, U=0, E=0)
2022/12/07 14:53:49 - Memory group by.0 - Finished processing (I=0, O=0, R=5, W=3, U=0, E=0)
2022/12/07 14:53:49 - Ordene por Estado.0 - Finished processing (I=0, O=0, R=3, W=3, U=0, E=0)
2022/12/07 14:53:49 - Write AVG State.0 - Finished processing (I=0, O=0, R=3, W=3, U=0, E=0)
2022/12/07 14:53:49 - pipeline_clientes - Unit test 'pipeline_clientes UNIT' passed successfully
2022/12/07 14:53:49 - pipeline_clientes - _____
2022/12/07 14:53:49 - pipeline_clientes - Write AVG State - data-set-avg : Test passed successfully against golden data set
2022/12/07 14:53:49 - pipeline_clientes - Unit test was successfully executed.
2022/12/07 14:53:49 - pipeline_clientes - _____
2022/12/07 14:53:49 - pipeline_clientes - Pipeline duration : 0.05 seconds [ 0.050" ]
2022/12/07 14:53:49 - pipeline_clientes - Execution finished on a local pipeline engine with run configuration 'local'
```

# Pipeline com Unit Test - Fail

### Unit test results

Here are the results of the unit test validations: (1 rows)

Pipeline	Unit test	Data set	Transform	Error?	Comment
1 pipeline_clientes	pipeline_clientes UNIT	data-set-avg	Write AVG State	Y	Incorrect number of rows received from transform, golden data set 'data-set-avg' has 3 rows in it and we received 1

Close

Transform Metrics Logging



2022/12/07 14:56:56 - Hop - Started the pipeline execution.  
2022/12/07 14:56:56 - pipeline\_clientes - Executing this pipeline using the Local Pipeline Engine with run configuration 'local'  
2022/12/07 14:56:56 - pipeline\_clientes - Execution started for pipeline [pipeline\_clientes]  
2022/12/07 14:56:56 - Read CSV.0 - Finished processing (I=0, O=0, R=6, W=6, U=0, E=0)  
2022/12/07 14:56:56 - Filter BRA.0 - Finished processing (I=0, O=0, R=6, W=1, U=0, E=0)  
2022/12/07 14:56:56 - Get Data Hoje.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)  
2022/12/07 14:56:56 - Calc Idade.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)  
2022/12/07 14:56:56 - Memory group by.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)  
2022/12/07 14:56:56 - Ordene por Estado.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)  
2022/12/07 14:56:56 - Write AVG State.0 - Finished processing (I=0, O=0, R=1, W=1, U=0, E=0)  
2022/12/07 14:56:56 - pipeline\_clientes - Unit test 'pipeline\_clientes UNIT' failed, 1 errors detected, 1 comments to report.  
2022/12/07 14:56:56 - pipeline\_clientes - \_\_\_\_\_  
2022/12/07 14:56:56 - pipeline\_clientes - Write AVG State - data-set-avg : Incorrect number of rows received from transform, golden data set 'data-set-avg' has 3 rows in it and we received 1  
2022/12/07 14:56:56 - pipeline\_clientes - \_\_\_\_\_  
2022/12/07 14:56:56 - pipeline\_clientes - Pipeline duration : 0.049 seconds [ 0.049"]  
2022/12/07 14:56:56 - pipeline\_clientes - Execution finished on a local pipeline engine with run configuration 'local'

## Web

- <https://hop.apache.org/>
- <https://www.know-bi.be/blog/5-minutes-to-build-unit-tests-in-apache-hop>

# Obrigado

Marcio Junior Vieira

[marcio@ambientelivre.com.br](mailto:marcio@ambientelivre.com.br)

@marviojvieira @ambientelivre

@ambientelivreopensource

<https://www.linkedin.com/in/mvieira1/>

**Slide da Palestra será publicada em:**

Marcio Junior Vieira

Linkedin....: <https://www.linkedin.com/in/mvieira1/>

Source Code: [https://github.com/ambientelivre/samples\\_hop](https://github.com/ambientelivre/samples_hop)

Nos siga nas redes: @ambientelivre @ambientelivreopensource