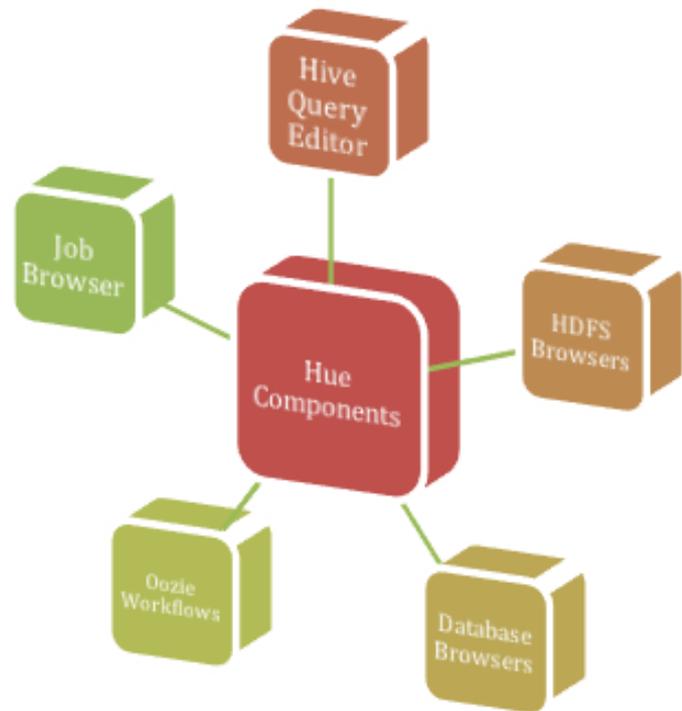


# Oficina HUE

- |       |                        |       |                |
|-------|------------------------|-------|----------------|
| I.    | Conceitos e Definições | X.    | Exercícios 1   |
| II.   | Componentes            | XI.   | Workflow Oozie |
| III.  | Navegação              | XII.  | Hands On 2     |
| IV.   | Job Browser            | XIII. | Exercícios 2   |
| V.    | Database Browser       |       |                |
| VI.   | Query Editor           |       |                |
| VII.  | Impala x Hive          |       |                |
| VIII. | Boas Práticas Impala   |       |                |
| IX.   | Hands On 1             |       |                |

Hue é um assistente de SQL para bancos de dados e DW, desenvolvido com o objetivo de acessar via browser os dados / arquivos armazenados em um cluster HDFS.

- HDFS Browser
- Job Browser
- Database Browser
- Query Editor
- Oozie Workflows



- **HDFS Browser** - Interface web cujo principal objetivo é permitir à interação do usuário com os arquivos armazenados no cluster Hadoop, como um sistema de arquivos.
- **Job Browser** - Ferramenta que nos permite acompanhar tarefas executadas no cluster Hadoop.
- **Database Browser** - Interface que nos permite visualizar de forma gráfica os “databases” existentes no HDFS, e seus componentes.
- **Query Editor** - Editor de consultas HQL que nos permite utilizar diversas ferramentas de consulta (Hive, Impala, etc) aos dados do cluster Hadoop de forma mais amigável.
- **Oozie Workflows** - Aplicação incorporada ao HUE que nos permite a criação de fluxos de trabalho e agendamento dos mesmos para execução no cluster Hadoop.

## Barra de atalhos da nova versão do HUE

The image shows the HUE interface with a sidebar on the left containing various shortcuts. Red arrows point from labels to specific icons in the sidebar:

- Editores** points to the code editor icon.
- Oozie** points to the Oozie workflow icon.
- Documentos** points to the document icon.
- Arquivos** points to the folder icon.
- Databases** points to the database icon.
- Wizard Import** points to the import icon.
- Profile / Sign out** points to the user profile icon.

The main workspace displays the Impala query editor. The top bar shows a search box and a 'Jobs' menu. The left sidebar shows the 'oficina\_hue' database with three tables: 'tb\_oficina\_hue\_tm', 'tb\_oficina\_hue\_txt\_ext', and 'tb\_oficina\_hue\_wf\_orc'. The main editor area contains a query:

```
Example: SELECT * FROM tablename, or press CTRL + space
```

Below the editor, the 'Query History' tab shows a query executed 11 minutes ago:

```
select
case when sg_uf = '' then 'Não informado' else
sg_uf end sg_uf,
dt_ref, ds_infracao,
cast(sum(qt_cnh_renovada) as int)
qt_cnh_renovada,
cast(sum(qt_novo_condutor_habilitado) as int)
qt_novo_condutor_habilitado,
cast(sum(qt_condutor_habilitado) as int)
qt_condutor_habilitado,
cast(sum(qt_ear) as int) qt_ear,
cast(sum(qt_exame_toxicologico) as int)
qt_exame_toxicologico,
cast(sum(qt_exames_toxicologicos_ear_cat_c_d_e)
as int) qt_exames_toxicologicos_ear_cat_c_d_e,
```

The right sidebar shows a filter for the 'impala' database with a list of categories: Aggregate, Analytic, Bit, Conditional, Date, Mathematical, Misc, String, and Type Conversion.

HDFS Browser, que permite visualizar, apagar, incluir, criar, compartilhar e movimentar arquivos e diretórios no HDFS através da interface web.

The screenshot displays the HDFS Browser interface. On the left, a sidebar lists various files and directories, including 'external', which is highlighted with a red arrow and labeled 'Arquivos / Diretórios'. The main area shows the 'File Browser' view for the path 'user / oficina\_hue / external', with a red arrow pointing to the path and labeled 'Caminho HDFS'. A table lists the files in this directory, with a red arrow pointing to the file 'log\_localidade.csv' and labeled 'Arquivo'.

Name	Size	User	Group	Permissions	Date
↑		oficina_hue	oficina_hue	drwxrwxrwx+	April 12, 2022 11:02 AM
.		oficina_hue	oficina_hue	drwxr-xr-x	April 07, 2022 10:04 AM
log_localidade.csv	589.2 KB	oficina_hue	oficina_hue	-rw-r--	April 07, 2022 10:04 AM

Por esta janela o usuário poderá, dependendo das permissões nos arquivos, realizar as ações disponíveis na dropdown “Ações” (renomear, mover, copiar, baixar, alterar permissões, alterar número de réplicas e compactar) para os arquivos selecionados através das *checkbox*, localizadas à esquerda de cada arquivo / diretório.

The screenshot displays the HDFS Browser interface. On the left, a sidebar shows a file tree with a search filter. The main area shows a file list with a table of file details. A red arrow points to the 'Actions' dropdown menu, which is open, showing options like Rename, Move, Copy, Download, Change permissions, Summary, Set replication, and Compress. Another red arrow points to the checkbox next to the selected file 'log\_localidade.csv'.

Name	Size	User	Group	Permissions	Date
log_localidade.csv		oficina_hue	oficina_hue	drwxrwxrwx+	April 12, 2022 11:02 AM
		oficina_hue	oficina_hue	drwxr-xr-x	April 07, 2022 10:04 AM
		oficina_hue	oficina_hue	-rw-r--r--	April 07, 2022 10:04 AM

Poderá também criar diretórios e arquivos através da dropbox “Novo” e fazer upload de arquivos no diretório corrente pelo botão carregar.

The screenshot displays the HDFS Browser interface. On the left is a sidebar with navigation icons and a file tree. The main area shows the 'File Browser' for the path 'user / oficina\_hue / external'. A search bar and 'Actions' menu are at the top. The 'Upload' and 'New' buttons are highlighted with red arrows, and a dropdown menu for 'New' is open, showing options for 'File' and 'Directory'. A table lists files and directories with columns for Name, Size, User, Group, Permissions, and Date.

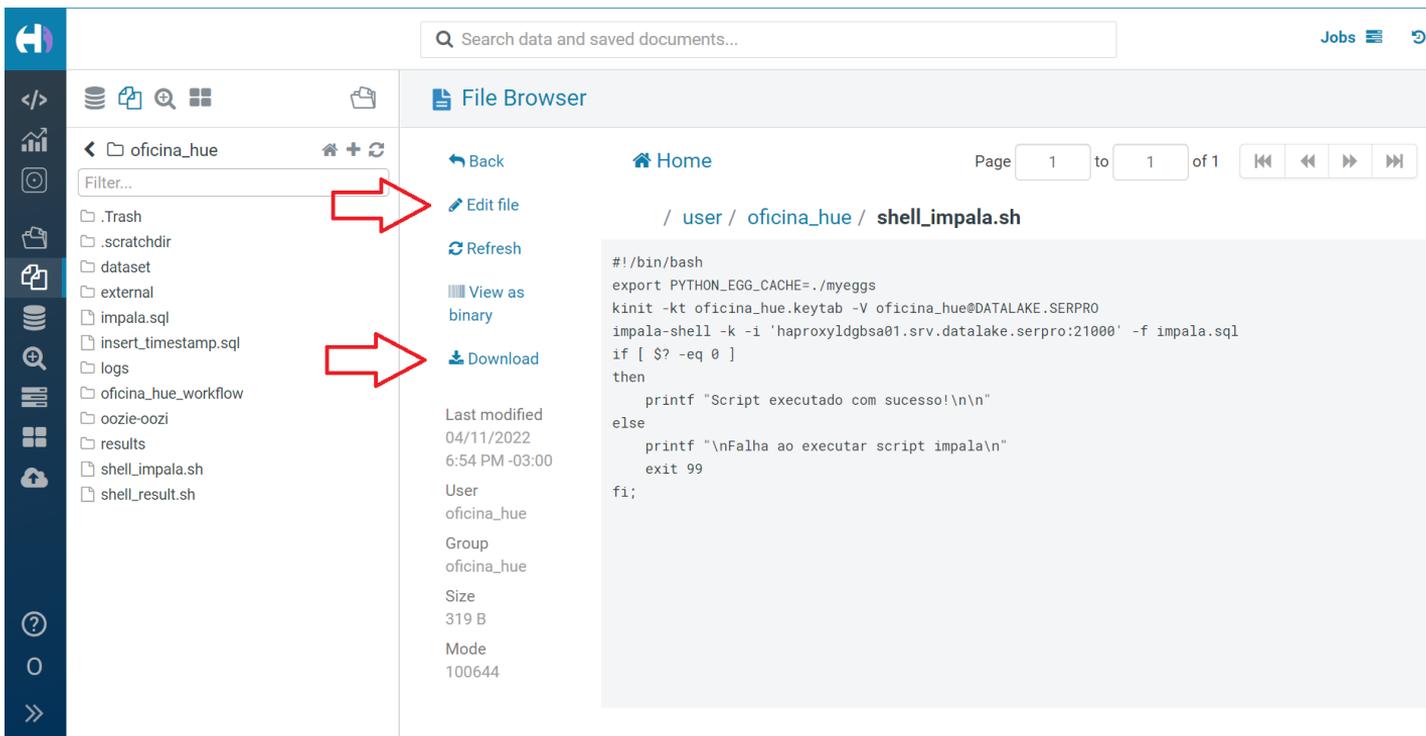
	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	↑		oficina_hue	oficina_hue	drwxrwxrwx+	April 12, 2022 11:02 AM
<input type="checkbox"/>	.		oficina_hue	oficina_hue	drwxr-xr-x	April 07, 2022 10:04 AM
<input type="checkbox"/>	log_localidade.csv	589.2 KB	oficina_hue	oficina_hue	-rw-r--	April 07, 2022 10:04 AM

Para apagar arquivos ou diretórios basta selecionar o referente pela checkbox e utilizar a dropbox “Move to trash”.

The screenshot displays the HDFS Browser interface. On the left is a sidebar with navigation icons and a file tree. The main area shows a search bar at the top, followed by a 'File Browser' header. Below this is a search input for file names and an 'Actions' dropdown menu. A red arrow points to the 'Move to trash' option in the dropdown, which has opened a sub-menu with a 'Delete forever' option. In the main file list, the file 'log\_localidade.csv' is selected, indicated by a checked checkbox and a red arrow pointing to it. The table below shows the details of the selected file.

	Name	Size	User	Group	Permissions	Date
<input type="checkbox"/>	↑		oficina_hue	oficina_hue	drwxrwxrwx+	April 12, 2022 11:02 AM
<input type="checkbox"/>	.		oficina_hue	oficina_hue	drwxr-xr-x	April 07, 2022 10:04 AM
<input checked="" type="checkbox"/>	log_localidade.csv	589.2 KB	oficina_hue	oficina_hue	-rw-r--r--	April 07, 2022 10:04 AM

O HDFS Browser ainda permite que realizemos alterações nos arquivos localizados no HDFS diretamente no Browser e download do mesmo, caso prefira editar em sua máquina.



The screenshot displays the HDFS Browser interface. On the left, a sidebar shows a file tree under the path `oficina_hue`. The tree includes folders like `.Trash`, `scratchdir`, `dataset`, `external`, `impala.sql`, `insert_timestamp.sql`, `logs`, `oficina_hue_workflow`, `oozie-oozi`, `results`, and files `shell_impala.sh` and `shell_result.sh`. Two red arrows point from the `Filter...` input and the `shell_impala.sh` file to the right-hand pane.

The right-hand pane, titled "File Browser", shows the selected file `shell_impala.sh`. It includes navigation controls (Back, Home, Page 1 to 1 of 1), action buttons (Edit file, Refresh, View as binary, Download), and file metadata (Last modified: 04/11/2022 6:54 PM -03:00, User: oficina\_hue, Group: oficina\_hue, Size: 319 B, Mode: 100644). The file content is displayed in a code editor:

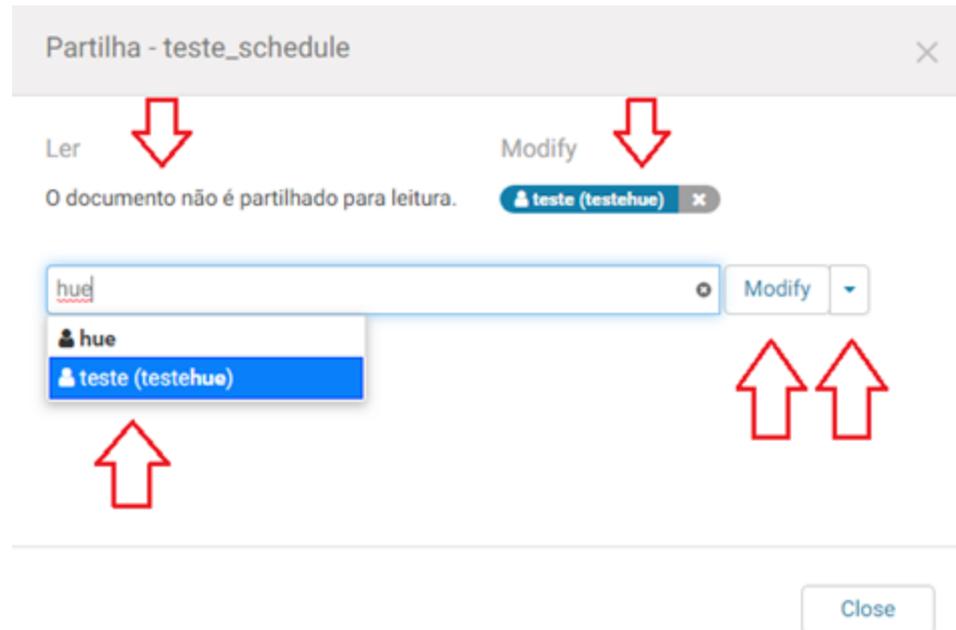
```
#!/bin/bash
export PYTHON_EGG_CACHE=./myeggs
kinit -kt oficina_hue.keytab -V oficina_hue@DATA LAKE.SERPRO
impala-shell -k -i 'haproxyldgbsa01.srv.data lake.serpro:21000' -f impala.sql
if [ $? -eq 0 ]
then
    printf "Script executado com sucesso!\n\n"
else
    printf "\nFalha ao executar script impala\n"
    exit 99
fi;
```

Podemos também realizar o compartilhamento de queries, workflows, schedules e outros documentos, com outros usuários do cluster. Acessando o menu principal, opção *Documents*, conseguimos visualizar os arquivos cuja edição foram feitas através das ferramentas do HUE (queries salvas, workflows oozie, schedules, diretórios, etc).

The screenshot displays the HUE HDFS Browser interface. The main area shows a table of documents under the 'My documents' folder. The table has columns for Name, Description, Type, Owner, and Last Modified. A context menu is open over the document 'sch\_send\_email', with the 'Share' option highlighted. Red arrows point to the 'Share' option and the document name.

Name	Description	Type	Owner	Last Modified
wkf_drdaniel		Oozie Workflow	oficina_hue	11/08/2020
My Workflow		Oozie Workflow	oficina_hue	11/08/2020
sch_exemplo_pgfn		Oozie Schedule	oficina_hue	11/08/2020
exemplo_pgfn		Oozie Workflow	oficina_hue	11/08/2020
sch_send_email		Oozie Schedule	oficina_hue	11/08/2020
wf_send_email	geração email	Oozie Workflow	oficina_hue	11/08/2020
My Workflow		Oozie Workflow	oficina_hue	08/24/2020
sch_test		Oozie Schedule	oficina_hue	08/24/2020
oficina_hue_hardson		Oozie Workflow	oficina_hue	08/10/2020
workflow_oficina_hue		Oozie Workflow	oficina_hue	08/10/2020

Na janela Partilha, o usuário deve digitar o nome do usuário (ou parte do nome) com quem deseja compartilhar o documento, e selecioná-lo quando mostrado. Para que seja efetivado o compartilhamento deve-se selecionar à direita qual tipo de permissão e clicar na descrição da permissão, para que a mesma apareça na parte superior da janela e seja assim efetivada.



O compartilhamento dos Documentos, conforme mostramos, pode ser realizado para usuários específicos, já o compartilhamento de arquivos dentro do HDFS só se realiza a nível de grupos ou abertura total do arquivo/diretório.

Para o compartilhamento de arquivos no HDFS devemos acessar o menu principal, opção *Files*, que nos levará para o diretório home do usuário. Selecionamos o arquivo a ser compartilhado através do checkbox à esquerda do arquivo e em seguida acessamos a dropdown *Ações* (*Actions*) selecionando a opção *Alterar Permissões* (*Change permissions*).

The screenshot displays the HDFS Browser interface. On the left, a sidebar shows a navigation menu with icons for home, search, and file management. The main area is titled 'File Browser' and shows a search bar at the top. Below the search bar, there are buttons for 'Actions', 'Move to trash', 'Upload', and 'New'. The interface shows a directory structure: 'Home / user / oficina\_hue'. A table of files is displayed with columns for Name, Group, Permissions, and Date. The file 'log\_localidade.csv' is selected, and the 'Actions' dropdown menu is open, highlighting the 'Change permissions' option. Red arrows point to the search bar, the 'Actions' button, the 'log\_localidade.csv' file, and the 'Change permissions' option in the dropdown menu.

Name	Group	Permissions	Date
↑	oficina_hue	drwxrwxrwx+	April 12, 2022 11:02 AM
.	oficina_hue	drwxr-xr-x	April 07, 2022 10:04 AM
log_localidade.csv	oficina_hue	-rw-r--	April 07, 2022 10:04 AM

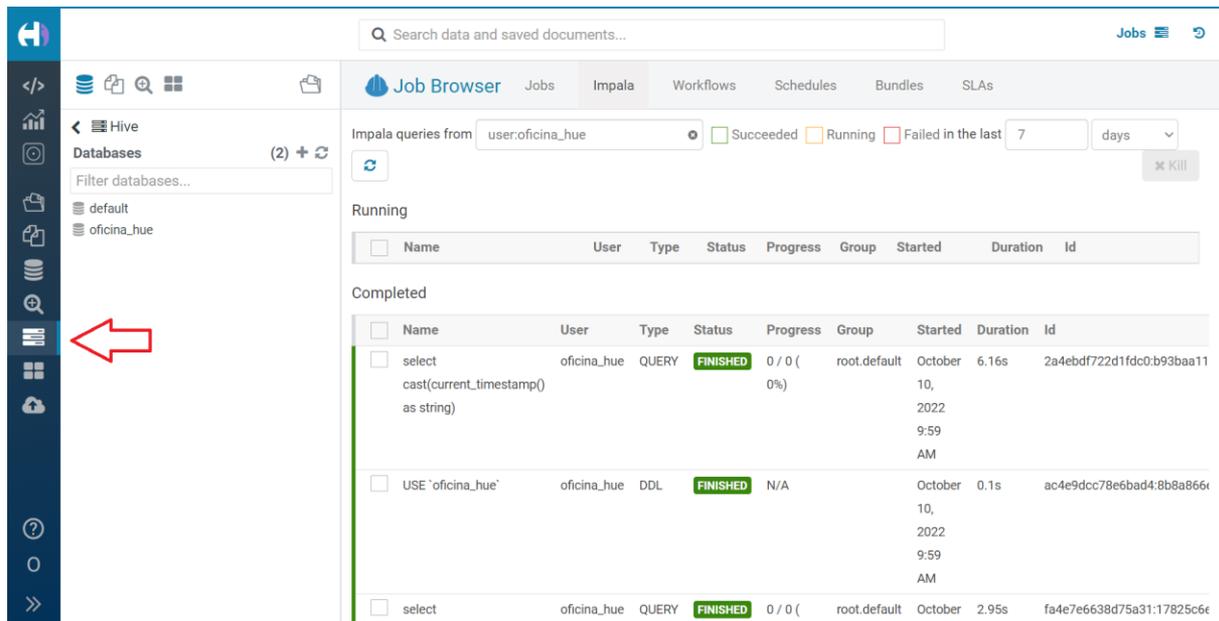
Por default, o usuário proprietário possui todas as permissões no arquivo, e os demais apenas permissão de leitura. Lembramos que mesmo que para leitura, os demais usuários ou o grupo do usuário corrente devem possuir permissão no diretório em que o arquivo se encontra, assim como nos demais diretórios caso ele não esteja no diretório raiz do usuário.

Para garantirmos que a permissão seja dada também aos diretórios de nível inferior ao diretório do arquivo, basta selecionar a opção Recursivo.

Change Permissions ✕

	User	Group	Other
Read	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Execute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sticky			<input type="checkbox"/>
Recursive			<input type="checkbox"/>

Utilizar o Hadoop implica na criação de muitos jobs e é bastante comum os desenvolvedores terem a necessidade de saber qual job está atualmente em execução no cluster do Hadoop ou qual o resultado de jobs finalizados (se houve sucesso ou falha na execução).



The screenshot shows the Hue Job Browser interface. The sidebar on the left contains various navigation icons, with a red arrow pointing to the 'Jobs' icon. The main content area displays job details for 'Impala queries from user:oficina\_hue'. The interface includes a search bar, a filter for 'Succeeded', 'Running', and 'Failed in the last 7 days', and a 'Kill' button. Below this, there are two tables: 'Running' and 'Completed'. The 'Completed' table shows three jobs with their status, progress, group, started time, duration, and ID.

Name	User	Type	Status	Progress	Group	Started	Duration	Id
select cast(current_timestamp() as string)	oficina_hue	QUERY	FINISHED	0 / 0 (0%)	root.default	October 10, 2022 9:59 AM	6.16s	2a4ebdf722d1fdc0:b93baa11
USE `oficina_hue`	oficina_hue	DDL	FINISHED	N/A		October 10, 2022 9:59 AM	0.1s	ac4e9dcc78e6bad4:8b8a866c
select	oficina_hue	QUERY	FINISHED	0 / 0 (0%)	root.default	October 10, 2022 9:59 AM	2.95s	fa4e7e6638d75a31:17825c6c

Por meio do navegador de jobs, podemos acessar todas as informações relacionadas ao job. Para isso, existe a opção no Hue que permite listar jobs e seus status.

Podemos ainda filtrar os jobs por status, busca textual e período em que foi executado.

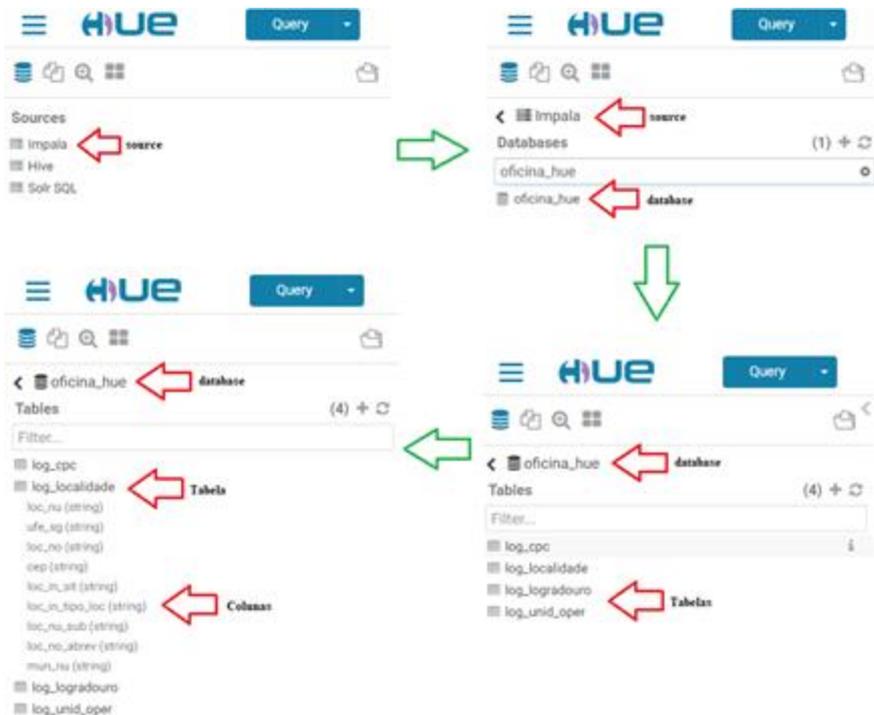
The screenshot shows the Job Browser interface with several red arrows pointing to specific features:

- Two arrows point to the 'Jobs' and 'Schedules' tabs in the top navigation bar.
- One arrow points to the 'Filter by id, name, user...' search box.
- Three arrows point to the 'Succeeded', 'Running', and 'Failed in the last' filter options.
- Three arrows point to the 'User', 'Type', and 'Status' columns in the job list table.

The 'Completed' section contains the following data:

<input type="checkbox"/>	Name	User	Type	Status	Progress	Group	Started	Duration	Id
<input type="checkbox"/>	src_relacoes_geral_dsx	05154921755	workflow	KILLED	100%		October 8, 2022 7:00 AM	7m, 26s	0003074-220806233739962-oozie-oozi-W
<input type="checkbox"/>	ALIENACAO_PAL_WF	72361417120	workflow	KILLED	100%		October 7, 2022 8:15 PM	21s	0003054-220806233739962-oozie-oozi-W
<input type="checkbox"/>	ALIENACAO_PAL_WF	72361417120	workflow	KILLED	100%		October 6, 2022 8:15 PM	19s	0003012-220806233739962-oozie-oozi-W
<input type="checkbox"/>	ALIENACAO_PAL_WF	72361417120	workflow	KILLED	100%		October 5, 2022 8:15 PM	18s	0002970-220806233739962-oozie-oozi-W

Através do HUE conseguimos visualizar de forma gráfica os “databases” existentes no HDFS, e seus componentes. Tal visualização é separada por fonte de dados capaz de apresentar as informações contidas nos databases (ex: hive, impala, solr, etc) até o nível de coluna.



Podemos visualizar também várias informações importantes sobre cada tabela clicando com o botão direito do mouse sobre a tabela desejada e selecionando a opção *Table Browser*.

The screenshot shows the Database Browser interface. The sidebar on the left contains navigation icons. The main content area is titled 'Table Browser' and shows the path 'Databases > oficina\_hue > tb\_oficina\_hue\_txt\_ext'. The 'Details' tab is active, displaying the following information:

**PROPERTIES**

- Table
- External and stored in location
- Created by oficina\_hue on Mon Apr 11 17:31:40 BRT 2022

**STATS**

- Data last updated on 05/02/2022 4:28 PM -03:00
- Schema last modified on 05/02/2022 4:28 PM -03:00 by s\_cargaserpro

**SCHEMA**

Column (9)	Type	Description	Sample
loc_nu	bigint		19 13
ufe_sg	string		AC AC
loc_no	string		Sena Madureira Plácido de Castro
cep	bigint		69940000 69928000
loc_in_sit	tinyint		0 0

O Query Editor além de permitir a criação de consultas às fontes de dados conectadas através do HUE, permite também que salvemos estas consultas no HDFS e mantém um histórico de todas as consultas executadas pelo usuário.

The screenshot displays the Hue Query Editor interface. On the left, a sidebar shows a navigation menu and a table browser for the 'oficina\_hue' database, listing tables like 'tb\_oficina\_hue\_tm' and 'tb\_oficina\_hue\_wf\_orc'. The main area shows a query editor with the SQL: `select cast(current_timestamp() as string);`. Below the editor, the 'Query History' and 'Saved Queries' tabs are visible. The 'Query History' tab shows three entries, with the first one highlighted and its ID `1244f21aad6e7ef0:869329fc00000000` highlighted. The 'Results (1)' tab shows the output: `cast(current_timestamp() as string)` with the value `2022-10-10 10:18:27.808422000`. On the right, a sidebar shows the 'impala' database and a list of data types, including 'Aggregate', 'Analytic', 'Bit', 'Conditional', 'Date', 'Mathematical', 'Misc', 'String', and 'Type Conversion'. Red arrows point to the 'Save' icon in the query editor toolbar, the highlighted query ID in the history, and the 'Type Conversion' category in the sidebar.

Possui uma aba lateral à direita que nos auxilia na utilização de funções de manipulação de dados, mostrando os nomes das mesmas e a sintaxe a ser utilizada.

The screenshot displays the Query Editor interface. On the left, a sidebar shows a list of tables under the 'oficina\_hue' database, including 'tb\_oficina\_hue\_tm', 'tb\_oficina\_hue\_txt\_ext', and others. The main area shows an Impala query editor with the following query:

```
1 | select cast(current_timestamp() as string);
```

Below the query, the execution results are shown in a table:

cast(current_timestamp() as string)
2022-10-10 10:18:27.808422000

On the right side, a function reference panel is open for the 'impala' database. It lists various functions such as 'Aggregate', 'Analytic', 'Bit', 'Conditional', 'Date', 'Mathematical', 'Misc', 'String', and 'Text'. A red arrow points to the 'impala' dropdown menu. Another red arrow points to the 'Conditional' category, and a third red arrow points to the 'cast' function within that category.

Existe ainda a possibilidade de o usuário compartilhar suas consultas salvas com outros usuários do cluster através do menu “Partilhar”. Basta digitar o nome do usuário na caixa de texto e selecionar qual tipo de permissão o usuário terá em sua consulta.

The screenshot displays the Query Editor interface with a modal window titled "Sharing - teste\_query" open. The modal contains two columns: "Read" and "Modify". Under "Read", there is a text input field containing "teste (testehue)" and a dropdown menu set to "Read". A red arrow points to the input field, and another red arrow points to the dropdown menu. The "Modify" column shows a similar setup but with a red arrow pointing to the "Read" dropdown. The background shows the main editor with a query: `select cast(current timestamp() as string);` and a table list on the left. A right sidebar shows a list of functions with red arrows pointing to the "Impala" header and the "Conditional" category.



X



## IMPALA

- Escrito em C++
- Data Marts
- Boa escolha para análise interativa e ad-hoc, pela rápida resposta
- Boa escolha para ferramentas de Business Intelligence que permitem aos usuários alterar consultas
- Usa Parquet como o formato de arquivo aconselhado
- Processamento em memória podendo falhar por falta de recurso.
- Utiliza Massively Parallel Processing (MPP)

## HIVE

- Escrito em Java
- Data Warehouse
- Boa escolha para consultas de longa duração que requerem transformações pesadas ou múltiplas junções
- Boa escolha para painéis que são predefinidos e não personalizáveis pelo visualizador
- Usa ORC como o formato de arquivo aconselhado
- Funciona melhor com JSON do que Impala
- Se ajusta ao recurso disponível, sendo mais tolerante a falhas.
- Utiliza modelo Map-reduce / Apache Tez

Memória x Disco

HIVE

VS

SAMSUNG

IMPALA

128GB

## PARQUET

O Impala permite que você crie, gereencie e consulte tabelas Parquet. Parquet é um formato de arquivo binário, orientado a colunas, altamente eficiente para os tipos de consultas em grande escala. O formato Parquet é extremamente aconselhado para consultas que verificam colunas específicas em uma tabela, por exemplo, para consultar tabelas "largas" com muitas colunas ou para realizar operações de agregação como SUM () e AVG () que precisam processar a maioria ou todos os valores de uma coluna. Cada arquivo de dados Parquet escrito pelo Impala contém os valores de um conjunto de linhas (referido como "grupo de linhas").

Em um arquivo de dados, os valores de cada coluna são organizados de forma que sejam todos adjacentes, permitindo uma boa compactação para os valores dessa coluna. As consultas em uma tabela Parquet podem recuperar e analisar esses valores de qualquer coluna rapidamente e com o mínimo de I/O.



Sales			
Product	Customer	Date	Sale
Beer	Thomas	2011-11-25	2 GBP
Beer	Thomas	2011-11-25	2 GBP
Vodka	Thomas	2011-11-25	10 GBP
Whiskey	Christian	2011-11-25	5 GBP
Whiskey	Christian	2011-11-25	5 GBP
Vodka	Alexei	2011-11-25	10 GBP
Vodka	Alexei	2011-11-25	10 GBP

Product	
ID	Value
1	Beer
2	Beer
3	Vodka
4	Whiskey
5	Whiskey
6	Vodka
7	Vodka

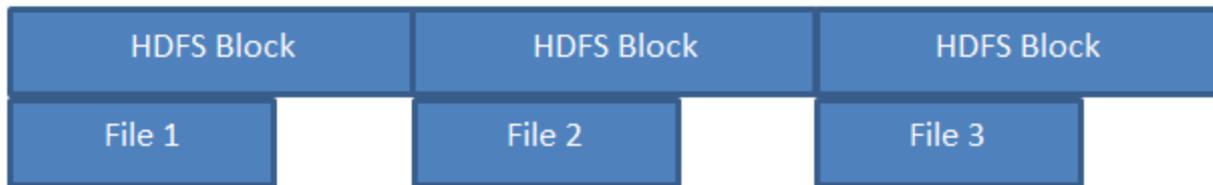
Customer	
ID	Customer
1	Thomas
2	Thomas
3	Thomas
4	Christian
5	Christian
6	Alexei
7	Alexei

## Formato mais apropriado de arquivo

Normalmente, para grandes volumes de dados (vários gigabytes por tabela ou partição), o tipo de arquivo Parquet tem um desempenho melhor devido a sua combinação de layout de armazenamento em forma de coluna, grandes requisições de I/O, compressão e codificação.

## Evite arquivos pequenos

- Arquivos menores que o blocos do HDFS ocuparão o bloco inteiro.
- Recuperar muitos arquivos tem um I/O maior que poucos, já que é lido bloco a bloco.
- Iniciar tarefas (tasks) para cada arquivo, em caso de pequenos arquivos, gerará um número muito maior de tasks, e com isso um custo muito maior de processamento.



## Evite arquivos pequenos

- Sempre use `INSERT ... SELECT` para copiar grandes volumes de dados de uma tabela para outra dentro do Impala. Evite `INSERT ... VALUES` para qualquer volume substancial de dados ou tabelas de performance crítica, pois cada cláusula cria um pequeno arquivo de dados separado.
- Caso seu processo de ingestão utilize o conceito de delta, e cada delta ingerido seja com um baixo volume de dados, talvez seja necessário de tempos em tempos gerar novamente sua tabela definitiva de dados, pois a ingestão provavelmente tem gerado pequenos arquivos parquet em cada execução e poderá fazer com que sua performance nas consultas seja prejudicada. Para isso pode utilizar o conceito de `INSERT OVERWRITE TABLE`.

## Granularidade das partições

- Quando você envia consultas que solicitam um valor específico ou um intervalo de valores para a coluna chave da partição, o Impala pode evitar a leitura de dados irrelevantes, resultando em uma grande economia de I/O em disco.
- Escolha uma estratégia de particionamento que insira pelo menos 256 MB de dados em cada partição para tirar vantagem do bulk de I/O do HDFS e das consultas distribuídas do Impala.
- Se você precisar reduzir o número de partições e aumentar o número de dados em cada partição, primeiro verifique as colunas chaves da partição que raramente são referenciadas ou que não são usadas por consultas críticas. Por exemplo, os dados de log do seu web site são particionados por ano, mês, dia e hora, mas a maioria das consultas são por dia, talvez então você precise apenas particionar por ano, mês e dia.

## Números inteiros menores para as colunas chaves das partições e para os joins

- Embora seja tentador usar strings para particionar coluna chave, já que esses valores se tornam nomes de diretórios HDFS, você pode diminuir a utilização de memória usando valores numéricos como campo chave de partição comum tais como ANO, MÊS e DIA. Use o menor número inteiro que possua o intervalo apropriado de dados, geralmente TINYINT para MÊS e DIA, e SMALLINT para ANO.
- Use a função `EXTRACT()` para remover campos de data e hora individualmente de um valor `TIMESTAMP`, e `CAST()` para retornar valor inteiro. Ao realizar consultas, de preferência para a utilização de colunas numéricas para a realização dos JOINS entre as tabelas, permitindo um cruzamento das informações sem necessidade de grandes varreduras ou utilização de memória.

## Minimize resultados se possível

Utilize as seguintes técnicas:

- Agregação: se você quer saber quantas linhas satisfazem uma determinada condição, o valor total de valores de alguma coluna, o menor ou maior valor, chame funções agregadas como COUNT( ), SUM( ), MIN( ) e MAX( ) na consulta para não precisar enviar o resultado para uma aplicação e fazer os cálculos na aplicação.
- Filtro: use os filtros aplicáveis na cláusula WHERE da consulta para eliminar linhas que não são relevantes.
- Cláusula LIMIT: se você precisa visualizar apenas alguns valores de um conjunto de dados, ou os valores superiores ou inferiores de uma consulta usando ORDER BY, inclua a cláusula LIMIT para reduzir o tamanho do retorno da consulta.

## Gerenciamento de Memória em Consultas

Em certas situações (queries complexas e/ou que tendem a demandar grandes quantidades de dados em memória) o otimizador do Impala pode tentar reservar um volume de memória maior que o necessário para o processamento da consulta gerando dois tipos de problemas:

- Query não ser executada por não haver recurso de memória suficiente no pool. Neste caso a query sequer será admitida para execução.
- Query é admitida para execução, porém reservará mais recurso de memória que o necessário, o que pode ocasionar a não execução de outras consultas no mesmo pool.

## Gerenciamento de Memória em Consultas

As soluções para resolvermos estes tipos de problema seriam:

- Gerenciar manualmente o recurso de memória a ser utilizado pelo Impala.  
Ex: `SET mem_limit="1G";`
- Adaptar sua consulta para execução no Hive, uma vez que o mesmo é mais robusto por usar `map_reduce`.
- Particionar sua consulta para utilização de uma quantidade de memória menor.  
Utilizar tabelas intermediárias, realizando assim uma consulta maior em várias etapas, reduzindo a utilização de memória em relação a uma única instrução `sql` mais complexa.

## Melhorando Performance e Consumo de Recursos Utilizando Estatísticas

Tanto o Impala quanto o Hive se utilizam das estatísticas geradas em suas tabelas para definição de plano de acesso e execução das consultas no Hadoop. Uma prática essencial para que consiga extrair a melhor performance de suas consultas é sempre manter as estatísticas das tabelas atualizadas. Para tanto devemos nos atentar em executar a atualização das estatísticas sempre que realizarmos alguma modificação nos dados das tabelas a serem consumidas.

Comandos:

Hive: `Analyze table nome_da_tabela compute statistics;`

Impala: `compute stats nome_da_tabela;`

## Hands on

### Criando estruturas e ingerindo dados

Search data and saved documents...

Jobs

### Import to table

1 Pick data from file /user/oficina\_hue/external/log\_localidade.csv

2 Move it to table oficina\_hue.log\_localidade

**SOURCE**

Type File

Path /user/oficina\_hue/external/log\_localidade.csv

**FORMAT**

Field Separator Choose... Record Separator New line Quote Character Double Quote

Has Header

Next

## Hands on

### Criando estruturas e ingerindo dados

Search data and saved documents...

Jobs

### Import to table

Pick data from file ✓ 2  
*/user/oficina\_hue/external/log\_localidade.csv* Move it to table oficina\_hue.log\_localidade

**DESTINATION**

Name oficina\_hue.log\_localidade

**PROPERTIES**

Format Text

Extras

Partitions + Add partition

Back Submit

## Hands on

Criando external table

```
CREATE EXTERNAL TABLE oficina_hue.tb_log_cpc_ext (cpc_nu bigint, ufe_sg string,  
loc_nu bigint, cpc_no string, cpc_endereco string, cep bigint)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ';'   
LOCATION '/user/oficina_hue/external'   
TBLPROPERTIES("skip.header.line.count"="1")
```

External tables funcionam como uma estrutura (esqueleto) de uma tabela que aponta diretamente para o arquivo/local, dentro do HDFS, indicado no momento de sua criação.

Quando dropamos uma external table, os arquivos referenciados por ela continuam no diretório em que se encontram, sendo eliminada apenas a estrutura de acesso.

## Hands on

Detalhe importante sobre internal (managed tables) e external tables:

Por definição, hoje no cluster de Brasília todas as tabelas criadas, independente de colocarmos de forma explícita que serão external, serão criadas como tal. Vale salientar que o conceito de external continuará valendo para as que explicitamente forem declaradas desta forma, logo se dropar uma tabela criada com o comando `CREATE EXTERNAL TABLE`, os dados continuarão no diretório cujo apontamento foi definido na criação da tabela, mas as tabelas external criadas sem explicitamente declaradas forem, quando dropadas, apagarão tanto a estrutura da mesma quanto os dados.

A exceção se dá para tabelas ACID (Atomicidade, Consistência, Isolamento e Durabilidade), estas serão criadas como tabelas internas gerenciadas pelo Hive (`TBLPROPERTIES('transactional'='true')`).

## Hands on

Ingerindo/criando tabela linha de comando

```
CREATE TABLE oficina_hue.tb_log_cpc_txt( cpc_nu bigint, ufe_sg string, loc_nu bigint,  
cpc_no string, cpc_endereco string, cep bigint)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ';'   
STORED AS TEXTFILE  
TBLPROPERTIES("skip.header.line.count"="1");
```

```
LOAD DATA INPATH '/user/oficina_hue/dataset' into table oficina_hue.tb_log_cpc_txt;
```

Quando utilizamos o comando *LOAD DATA* para inserirmos dados de um dado arquivo/diretório do HDFS para uma tabela específica, os arquivos são movidos para o diretório da tabela em questão e quando dropada a tabela, os arquivos serão literalmente apagados do HDFS.

## Hands on

Criando tabela Parquet utilizando uma tabela texto.

```
CREATE TABLE oficina_hue.tb_log_cpc_pq STORED AS PARQUET as  
select * from oficina_hue.tb_log_cpc_txt;
```

Podemos realizar a criação de tabelas tipo parquet, tanto pelo Impala (preferencialmente) quanto pelo Hive, através da consulta via HQL em uma tabela já existente.

## Hands on

### Criando Views

```
CREATE VIEW vw_log_cpc as  
SELECT * FROM log_cpc  
WHERE ufe_sg = 'AL';
```

Views podem ser criadas para armazenamento de queries pré-definidas como de costume, mas podem representar também uma restrição para um grupo de usuários com menos privilégio.

Obs: para se conseguir criar uma view, o usuário necessita ter acesso à todos os dados cuja view faz referência.

## Hands on

### Views Hive x Impala (incompatibilidades)

Ao criar uma view, é importante observarmos em que editor (hive ou impala) a view está sendo criada, devido ao comportamento das definições desta view relativos tanto às estruturas de dados, quanto às funções nativas de cada editor. Isto vale tanto para views quanto para tabelas.

Por exemplo, quanto ao uso de tipos e/ou estruturas de dados que funcionam exclusivamente em um dos editores. O impala, não conseguirá ler uma view que possua colunas do tipo binary ou utilizem funções específicas do Hive.

## Hands on

### Views / Tabelas Hive x Impala (incompatibilidades)

No Hive:

```
CREATE TABLE oficina_hue.teste_date(id_teste int, dt_teste date);
```

```
INSERT INTO oficina_hue.teste_date VALUES (1,'2021-01-01');
```

```
CREATE VIEW oficina_hue.vw_teste_date as
```

```
SELECT id_teste, date_format(dt_teste,'y') as year FROM oficina_hue.teste_date;
```

No Impala:

```
SELECT * FROM oficina_hue.vw_teste_date;
```

## Hands on

### Materialized views

*Materialized Views* não eram suportadas na versão do Hive que utilizávamos (Hive 2.1.1 - CDH 6.3.4), mas com a migração do cluster, já temos a versão Hive 3.1.3 - CDP 7.1.7 em produção, estando disponível à mesma. Para que a mesma funcione, deve-se criar a tabela cuja Materialized View irá realizar a consulta, utilizando o parâmetro 'transactional'='true' e um formato que permita que a tabela seja transacionada (Ex: ORC).

```
CREATE TABLE oficina_hue.teste_date(id_teste int, dt_teste date) stored as orc  
tblproperties('transactional'='true');
```

```
CREATE materialized VIEW oficina_hue.vw_teste_date as  
select id_teste, dt_teste from oficina_hue.teste_date;
```

## Hands on

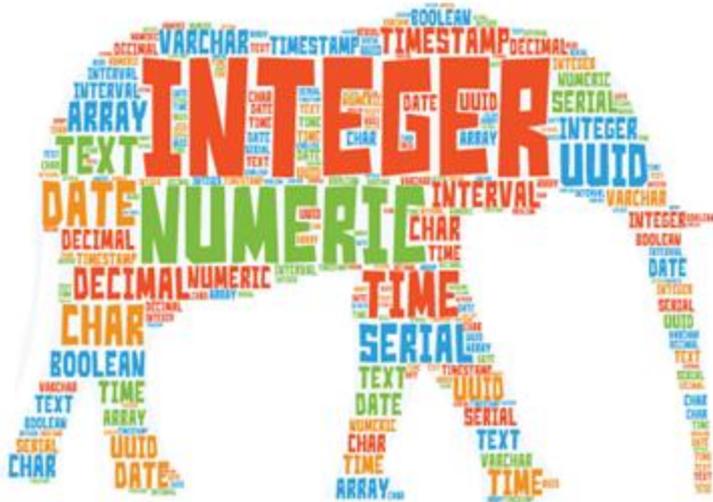
### Materialized views

*Para atualização dos dados das materialized views devemos usar o comando do exemplo abaixo:*

```
ALTER MATERIALIZED VIEW oficina_hue.vw_teste_date REBUILD;
```

*As tabelas ACID no hive suportam tanto insert, quanto update e delete, mas devemos salientar um detalhe. As materialized views criadas com base em tabelas que suportam as três operações de modificação de dados, não poderão ser lidas pelo Impala, o mesmo só aceita leitura de materialized views criadas em tabelas cujo ACID suportado seja apenas insert (`tblproperties 'transactional_properties'='insert_only'`)*

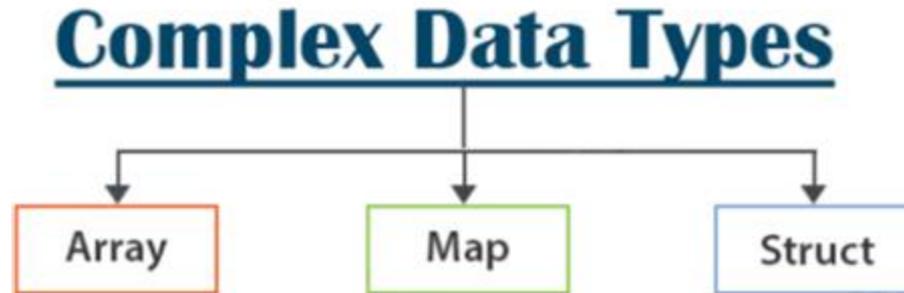
Considerando a versão Cloudera do condomínio atual, estes são os tipos de dados suportados, respectivamente, pelo hive e impala.



Hive	Impala
ARRAY*	ARRAY*
BIGINT	BIGINT
BOOLEAN	BOOLEAN
CHAR	CHAR
DATE	DATE
DECIMAL	DECIMAL
DOUBLE	DOUBLE
FLOAT	FLOAT
INT	INT
MAP*	MAP*
REAL	REAL
SMALLINT	SMALLINT
STRING	STRING
STRUCT*	STRUCT*
TIMESTAMP	TIMESTAMP
TINYINT	TINYINT
VARCHAR	VARCHAR
BINARY	-

## Complex types

Também conhecidos como tipos aninhados (nested types), esses tipos de dados permitem a representação de vários valores de dados em uma única posição de linha/coluna. Eles diferem dos tipos de coluna conhecidas, como BIGINT e STRING, conhecidos como tipos escalares ou tipos primitivos, que representam um único valor de dados em uma determinada posição de linha/coluna.



## Tipos dados Complex types

- **Array:** Um tipo complex type que pode representar um número arbitrário de elementos ordenados. Os elementos podem ser escalares ou outro tipo complexo (ARRAY, STRUCT ou MAP)
- **Map:** representa um conjunto arbitrário de pares de valores-chave. A parte chave deve ser, necessariamente, um tipo escalar, enquanto a parte do valor pode ser um escalar ou outro tipo complexo. Um exemplo simples de pensarmos em um uso para o tipo map, seria representarmos um conjunto de métricas para uma tabela de municípios, onde poderíamos ter informações como o código e nome do município como campos escalares e um campo do tipo map, contendo uma chave com o nome da métrica e um valor correspondente para o município (exemplos: 'Area:432244', 'Populacao':213685, 'PibPerCapta:R\$ 19.184,77')
- **Struct:** este tipo é útil para combinar duas tabelas relacionadas, de modo a minimizar a repetição. A maneira mais comum de representar esses dados é como um ARRAY de elementos STRUCT. O tipo STRUCT costuma ser mais útil como um item de um ARRAY ou a parte do valor do par de valores-chave em um MAP.

## Manipulando complex types - Impala x Hive

O Impala usa a notação de ponto para se referir a nomes de elementos ou elementos dentro de tipos complexos e notação de join para fazer referência cruzada de colunas escalares com os elementos de tipos complexos na mesma linha, em vez da cláusula LATERAL VIEW e da função EXPLODE () do HiveQL.

Uma consulta como `SELECT * FROM banco.tabela` feita pelo hive sobre uma tabela que contenha colunas do tipo complex type é aceita normalmente. Já no impala, apesar de não termos erro de execução, o retorno será apenas dos dados referentes às colunas escalares.

A versão atual do Impala não suporta a criação de novos dados com colunas do tipo complex type. Assim, o caminho que demonstraremos aqui é criarmos uma tabela do tipo parquet e alimentá-la com o hive, para então realizarmos consultas via impala.

## Hands on - Carregando dados Complex Type

```
[
  {
    "nome": "Joao",
    "sobrenome": "da Silva",
    "idade": "25",
    "endereco": {
      "logradouro": "Rua Abacaxi",
      "municipio": "Manaus",
      "uf": "AM",
      "cep": "11111-111"
    },
    "telefone": [
      {
        "tipo": "residencial",
        "numero": "92 1111-1234"
      },
      {
        "tipo": "celular",
        "numero": "92 3333-4567"
      }
    ]
  },
]
```

```
{
  "nome": "Maria",
  "sobrenome": "das Oliveiras",
  "idade": "29",
  "endereco": {
    "logradouro": "Rua Laranja",
    "municipio": "Feira de Santana",
    "uf": "BA",
    "cep": "22222-222"
  },
  "telefone": [
    {
      "tipo": "residencial",
      "numero": "75 2222-4321"
    },
    {
      "tipo": "celular",
      "numero": "75 4444-5678"
    }
  ]
},
```

```
{
  "nome": "Pedro",
  "sobrenome": "dos Santos",
  "idade": "40",
  "endereco": {
    "logradouro": "Rua Limoeiro",
    "municipio": "Pirenópolis",
    "uf": "GO",
    "cep": "33333-333"
  },
  "telefone": [
    {
      "tipo": "residencial",
      "numero": "62 3333-9876"
    },
    {
      "tipo": "celular",
      "numero": "62 5555-4321"
    }
  ]
}
]
```

## Hands on - Carregando dados Complex Type

```
{ "nome": "Joao", "sobrenome": "da Silva", "idade": "25", "endereco": { "logradouro": "Rua Abacaxi", "municipio": "Manaus", "uf": "AM", "cep": "11111-111" }, "telefone": [ { "tipo": "residencial", "numero": "92 1111-1234" }, { "tipo": "celular", "numero": "92 3333-4567" } ] }
```

```
{ "nome": "Maria", "sobrenome": "das Oliveiras", "idade": "29", "endereco": { "logradouro": "Rua Laranja", "municipio": "Feira de Santana", "uf": "BA", "cep": "22222-222" }, "telefone": [ { "tipo": "residencial", "numero": "75 2222-4321" }, { "tipo": "celular", "numero": "75 4444-5678" } ] }
```

```
{ "nome": "Pedro", "sobrenome": "dos Santos", "idade": "40", "endereco": { "logradouro": "Rua Limoeiro", "municipio": "Pirenópolis", "uf": "GO", "cep": "33333-333" }, "telefone": [ { "tipo": "residencial", "numero": "62 3333-9876" }, { "tipo": "celular", "numero": "62 5555-4321" } ] }
```

## Hands on - Carregando dados Complex Type

Browser de ficheiros

⏪ Voltar

🏠 Início

Página 1 to 1 de 1 ⏪ ⏩

✎ Editar ficheiro

🔄 Refresh

📄 Visualizar como binário

⬇️ Download

Última modificação  
24/10/2022 14:59  
-03:00

User  
oficina\_hue

Grupo  
oficina\_hue

Size  
955 B

Modo  
100644

/ user / oficina\_hue / json2hadoop / pessoa-formatted.json

```
{ "nome": "Joao", "sobrenome": "da Silva", "idade": "25", "endereco": { "logradouro": "Rua Abacaxi", "municipio": "Manaus", "uf": "AM", "cep": "11111-111" }, "telefone": [ { "tipo": "residencial", "numero": "92 1111-1234" }, { "tipo": "celular", "numero": "92 3333-4567" } ] }
{ "nome": "Maria", "sobrenome": "das Oliveiras", "idade": "29", "endereco": { "logradouro": "Rua Laranja", "municipio": "Feira de Santana", "uf": "BA", "cep": "22222-222" }, "telefone": [ { "tipo": "residencial", "numero": "75 2222-4321" }, { "tipo": "celular", "numero": "75 4444-5678" } ] }
{ "nome": "Pedro", "sobrenome": "dos Santos", "idade": "40", "endereco": { "logradouro": "Rua Limoeiro", "municipio": "Pirenópolis", "uf": "GO", "cep": "33333-333" }, "telefone": [ { "tipo": "residencial", "numero": "62 3333-9876" }, { "tipo": "celular", "numero": "62 5555-4321" } ] }
```





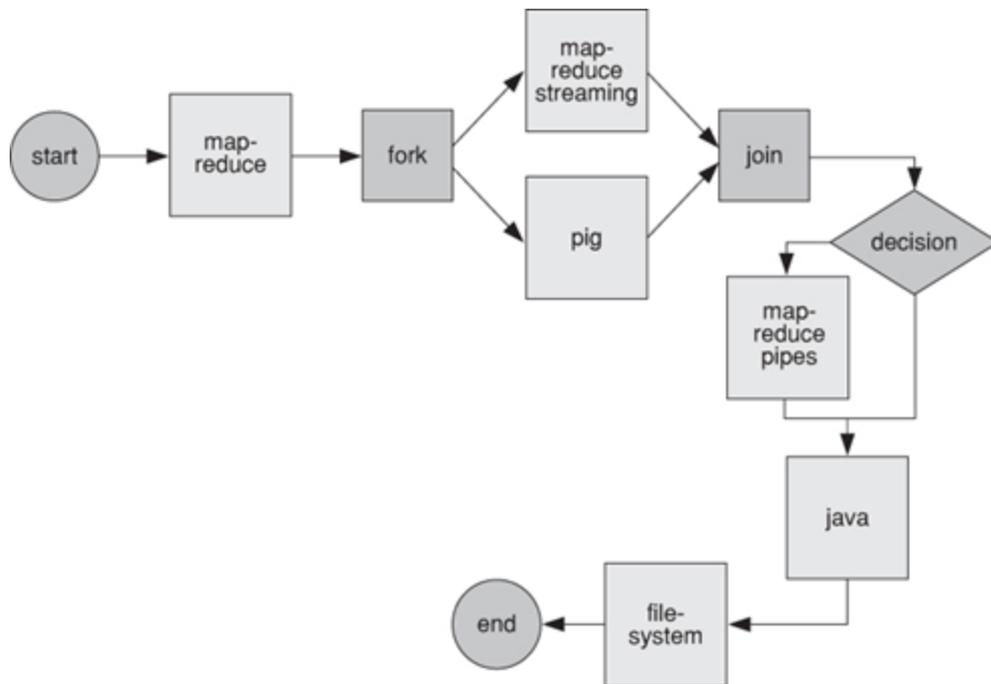




## Workflows Oozie

O Oozie é um mecanismo de workflow, baseado em servidor, especializado na execução de jobs com ações que executam tarefas em um cluster Hadoop.

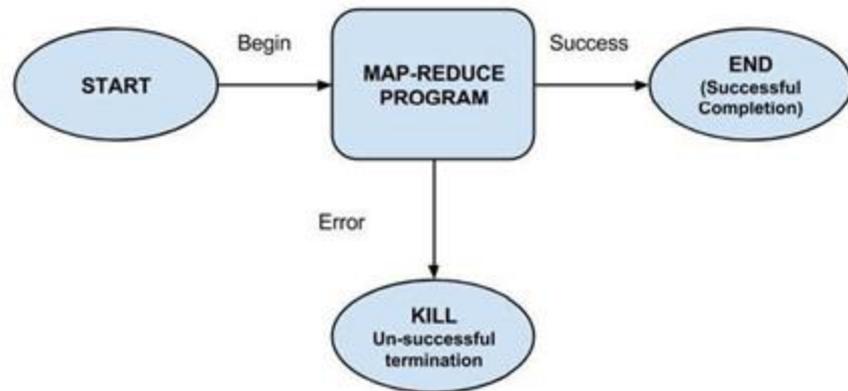
Para fins do Oozie, um Workflow é uma coleção de ações organizadas em um DAG (Direct Acyclic Graph) de controle de dependência.



## Workflows Oozie

As definições de workflows do Oozie são escritas em hPDL (uma linguagem de definição de processo XML semelhante ao JBOSS JBPM jPDL).

As ações dos workflows do Oozie iniciam trabalhos em sistemas remotos do Hadoop. Após a conclusão da ação, os sistemas remotos chamam o Oozie de volta para notificar a conclusão da ação, neste ponto o Oozie prossegue para a próxima ação no workflow.



## Workflows Oozie

Os workflows do Oozie contém nós de fluxo de controle e nós de ação.

Os nós de fluxo de controle definem o início e o fim de um workflow (nós de início, término e falha) e fornecem um mecanismo para controlar o caminho de execução do workflow (nós de decisão, bifurcação e junção).

Os nós de ação são o mecanismo pelo qual um fluxo de trabalho dispara a execução de uma tarefa de computação / processamento. O Oozie fornece suporte para diferentes tipos de ações: mapreduce do Hadoop, sistema de arquivos do Hadoop, SSH, HTTP, e-mail e subfluxo do Oozie. O Oozie pode ser estendido para oferecer suporte a tipos adicionais de ações.

## Hands on

### Workflow na prática

Por definição, tudo que criamos e salvamos com a interface do Hue é definido como Documento (queries, workflows, schedules, scripts, etc)

Para acessarmos a interface de construção de workflows do Oozie, bem como os fluxos já criados, devemos acionar o menu Documents, localizado no menu principal do Hue.

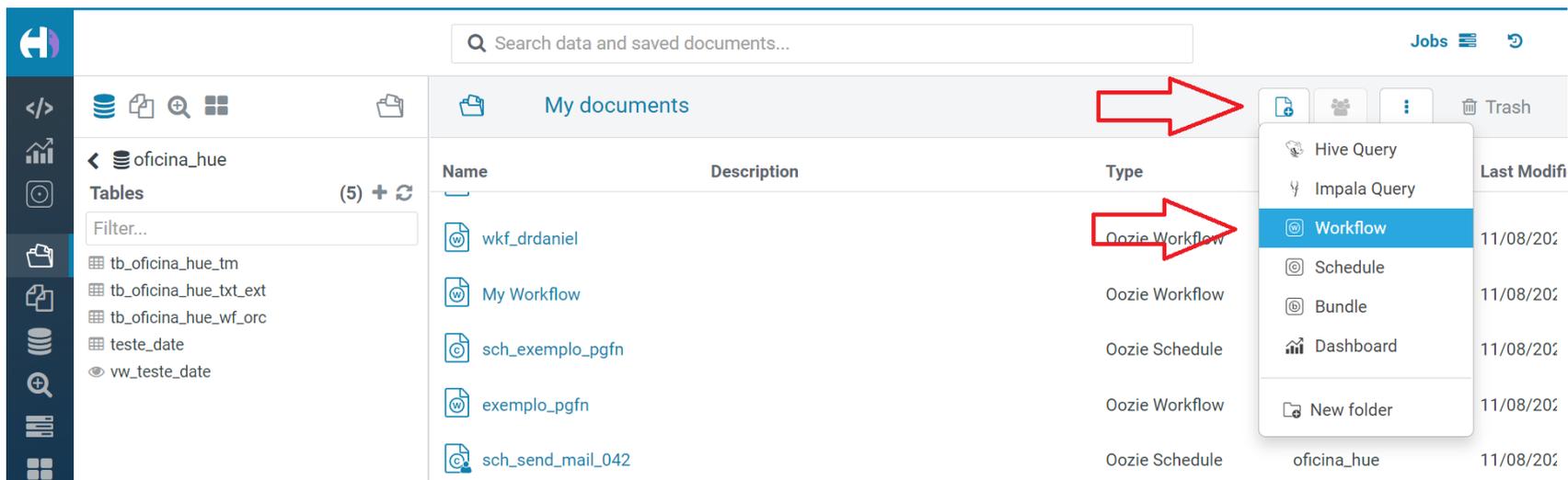
The screenshot shows the Hue web interface. At the top, there is a search bar and a 'Jobs' menu. Below that, the 'My documents' menu is highlighted with a red arrow. On the left sidebar, the 'Documents' icon is also highlighted with a red arrow. The main content area displays a table of documents:

Name	Description	Type	Owner	Last Modified
wkf_drdaniel		Oozie Workflow	oficina_hue	11/08/2020
My Workflow		Oozie Workflow	oficina_hue	11/08/2020
sch_exemplo_pgfn		Oozie Schedule	oficina_hue	11/08/2020
exemplo_pgfn		Oozie Workflow	oficina_hue	11/08/2020
sch_send_mail_042		Oozie Schedule	oficina_hue	11/08/2020
wf_send_mail_042	teste geração email	Oozie Workflow	oficina_hue	11/08/2020
My Workflow		Oozie Workflow	oficina_hue	08/24/2020
sch_teste_email		Oozie Schedule	oficina_hue	08/24/2020
oficina_hue_handson		Oozie Workflow	oficina_hue	08/10/2020
workflow_oficina_hue		Oozie Workflow	oficina_hue	08/10/2020

## Hands on

## Workflow na prática

Para criação de um novo workflow, devemos acessar o botão *Novo Documento* localizado na parte superior direita da janela do Hue, e selecionar a opção *Workflow*.



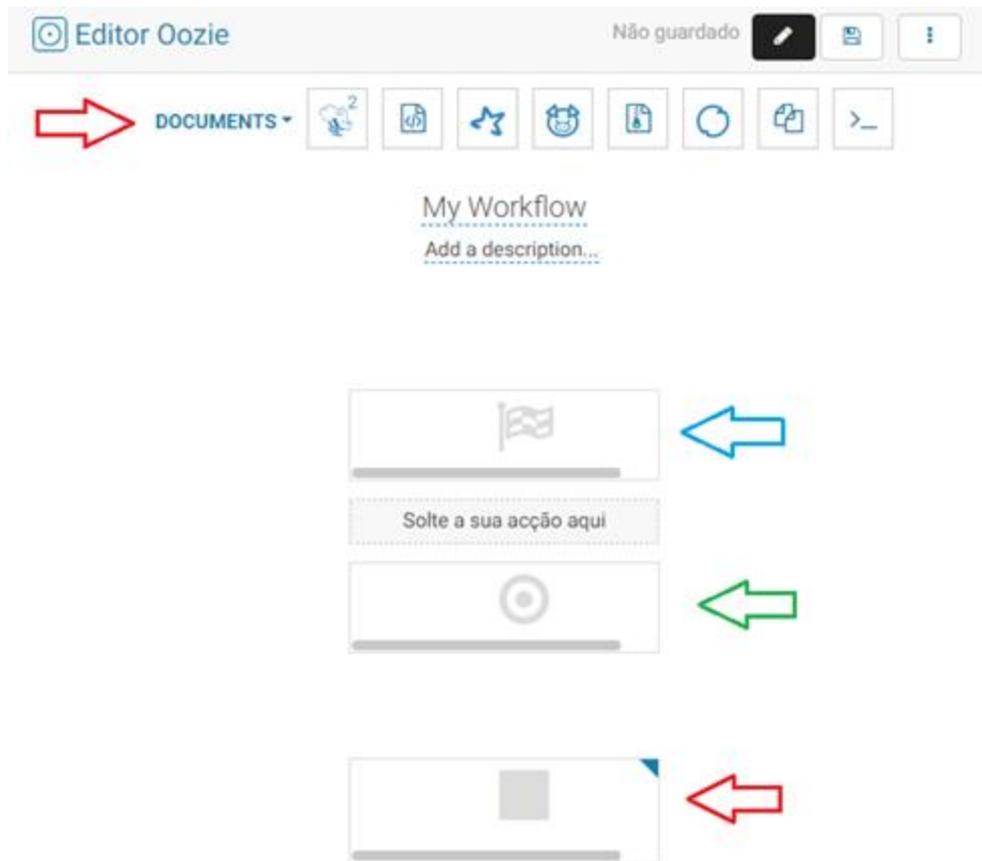
The screenshot shows the Hue interface with the 'Novo Documento' menu open. A red arrow points to the 'Novo Documento' button in the top right corner. Another red arrow points to the 'Workflow' option in the dropdown menu. The table below shows the current state of the document list.

Name	Description	Type	Last Modified
wkf_drdaniel		Oozie Workflow	11/08/202
My Workflow		Oozie Workflow	11/08/202
sch_exemplo_pgfn		Oozie Schedule	11/08/202
exemplo_pgfn		Oozie Workflow	11/08/202
sch_send_mail_042		Oozie Schedule	11/08/202

## Hands on

### Workflow na prática

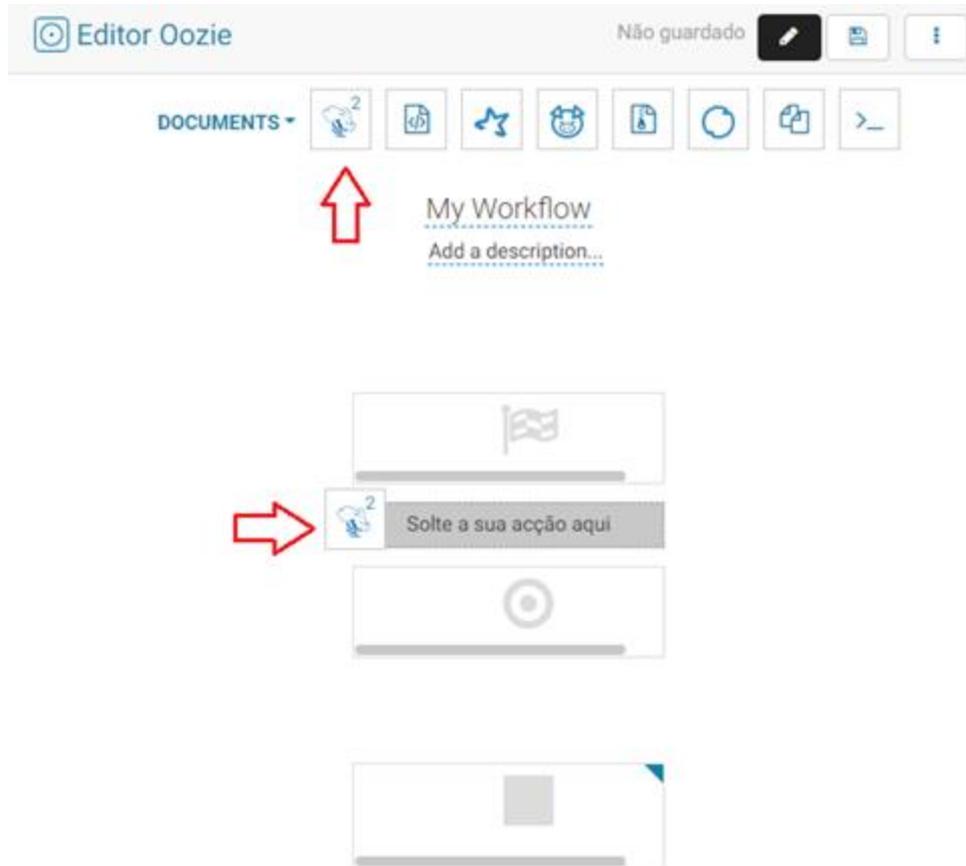
O Hue já trará uma estrutura básica de workflow com o ponto de início do mesmo, o ponto de finalização em caso de sucesso e o ponto de parada em caso de erro. Uma lista de documentos/ações que podemos implementar no fluxo também nos é apresentada.



## Hands on

### Workflow na prática

O framework trabalha com o esquema drag & drop. Você seleciona a ação que deseja e a arrasta para o local do workflow em que ela deve ser executada.



## Hands on

### Workflow na prática

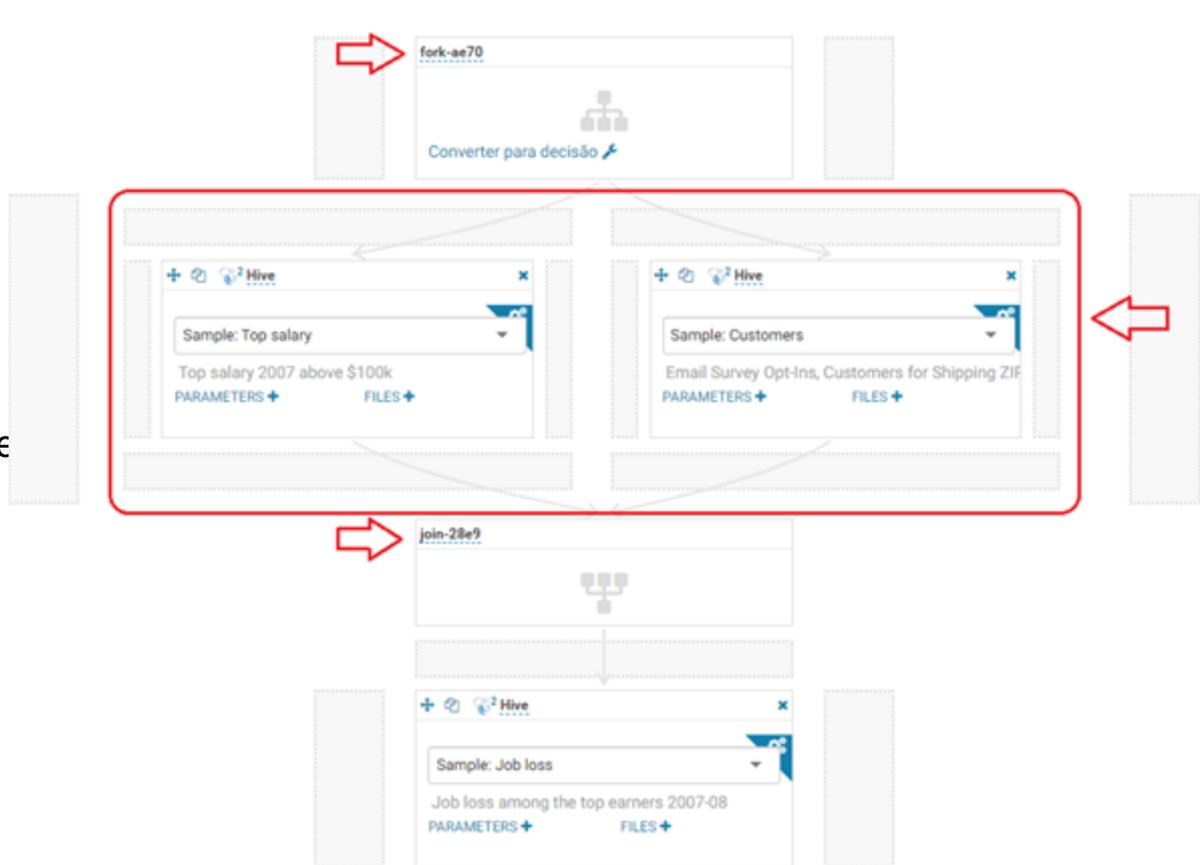
As dependências são criadas de acordo com o local em que se “solta” a ação que será executada, e pode ser vista clicando no ícone de engrenagens de cada etapa do fluxo, na aba *Transições*.

The image shows a screenshot of the Oozie workflow editor. It displays a vertical sequence of workflow elements: a start node (flag icon), two Hive action nodes, and an end node (circle icon). The first Hive action is titled "Sample: Customers" with the description "Email Survey Opt-Ins, Customers for Shipping ZIP Code". The second Hive action is titled "Sample: Job loss" with the description "Job loss among the top earners 2007-08". Red arrows point from the gear icon of the first action to the gear icon of the second action, and from the gear icon of the second action to the end node, indicating the flow of dependencies. A separate window in the bottom right corner shows the "Transições" (Transitions) tab for a Hive action, with "OK" set to "hive-c25d" and "KO" set to "Kill".

## Hands on

### Workflow na prática

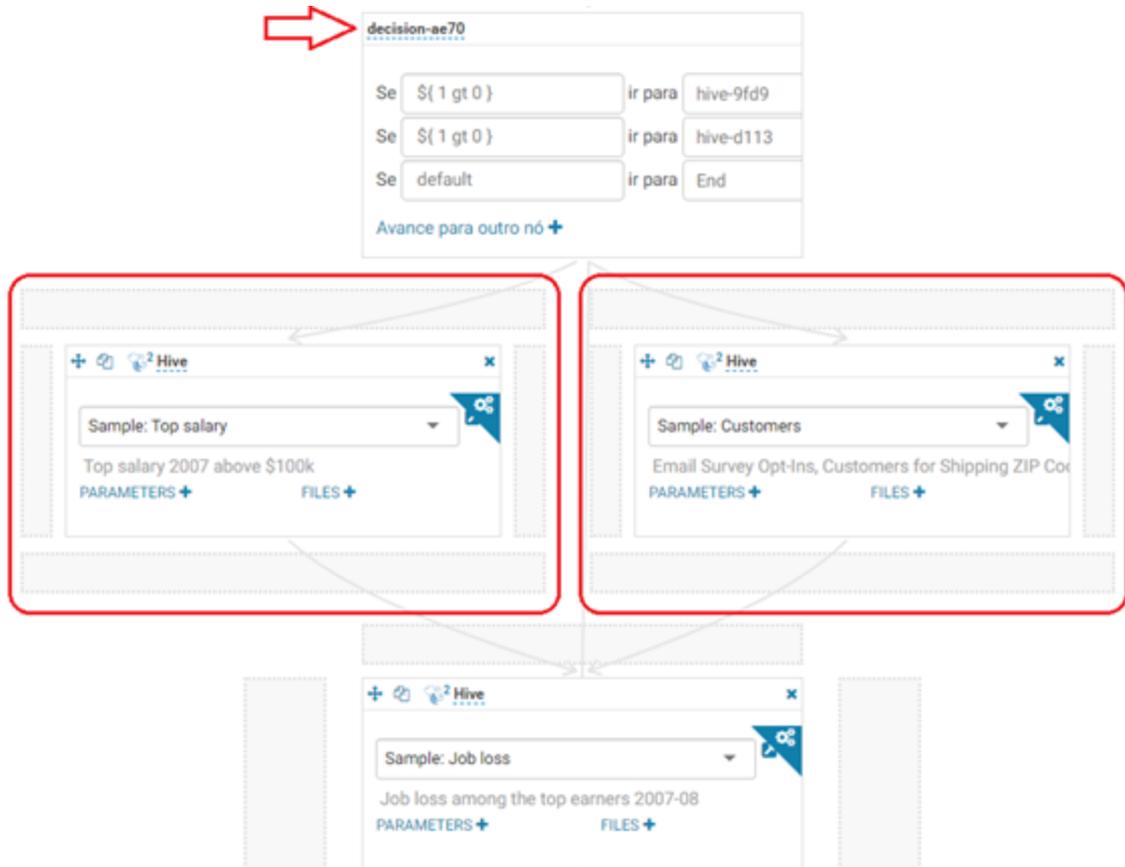
Quando houver necessidade de se executar duas ou mais ações em paralelo, ou que necessite de uma ação de decisão, deve-se arrastar a ação a ser executada para caixa lateral à ação já incluída no workflow. Isso fará com que sejam geradas duas novas etapas no fluxo, no caso de paralelismo serão geradas etapas de *fork* e *join*, unicamente para controle do fluxo.



## Hands on

### Workflow na prática

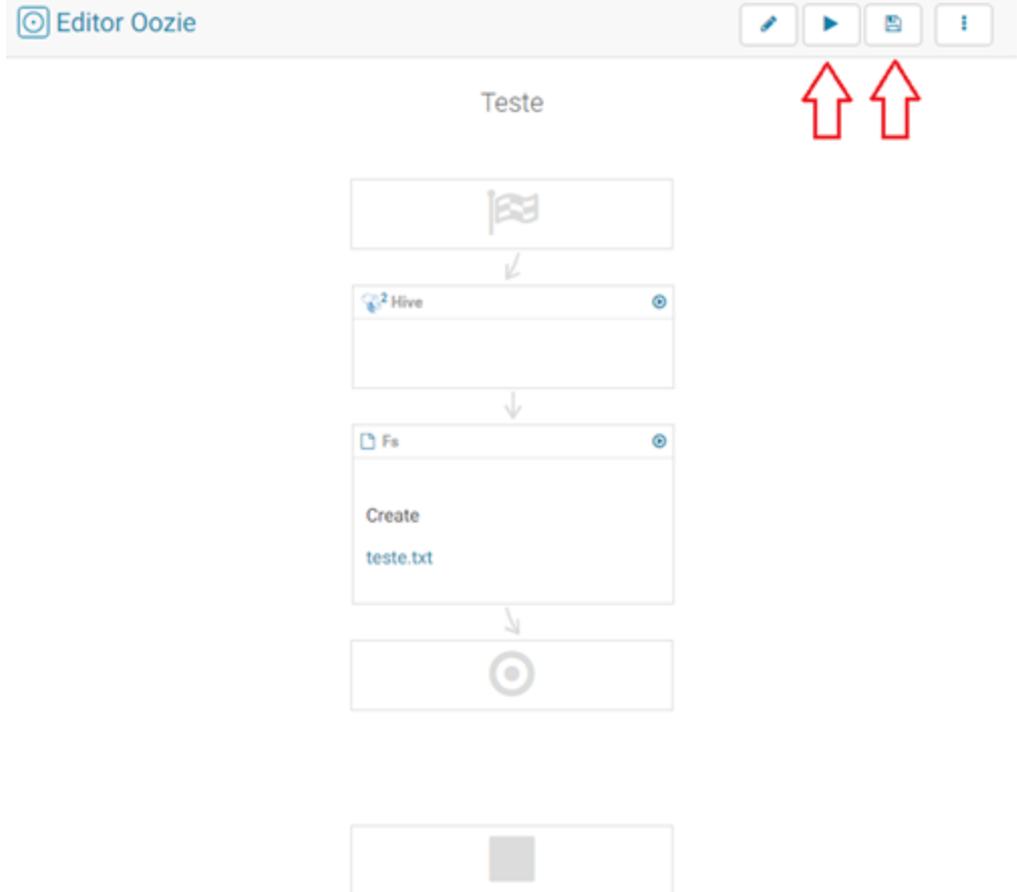
No caso de condicional, é gerada uma etapa *decision*, onde o usuário pode definir em quais situações o fluxo deve seguir determinado caminho.



## Hands on

### Workflow na prática

Para executar pontualmente o fluxo criado, deve-se primeiramente salvá-lo caso haja alterações recentes e clicar no botão play.



The screenshot shows the Oozie Editor interface. At the top, there is a toolbar with four icons: a pencil (edit), a play button (execute), a document (save), and a vertical ellipsis (more options). The play button is highlighted with a red arrow. Below the toolbar, the workflow is titled "Teste". The workflow consists of several steps: a flag icon, a Hive action (labeled "Hive"), a File System (Fs) action (labeled "Fs" with "Create teste.txt" below it), a target icon, and a square icon. Red arrows point to the play button and the Hive action.

## Hands on

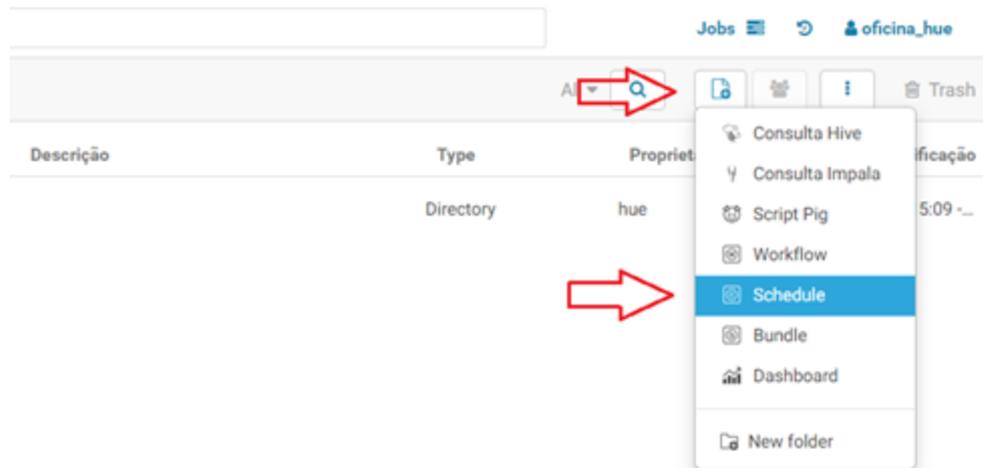
### Scheduler

Após a criação de seus Workflows oozie, você pode ainda realizar agendamento de suas execuções utilizando o programa Schedule (Oozie Coordinator).

Através do Schedule poderá agendar a execução de seus fluxos com base em tempo, eventos ou dados.

Para criação de um novo fluxo, deve-se acessar o Menu principal, Browser Documents.

No canto superior direito irá encontrar um botão para criação de novos documentos, e nesta lista a opção schedule.



## Hands on

### Scheduler

Os dados básicos a serem fornecidos para criação do agendamento são:

- Workflow a ser agendado
- Frequência de execuções: Definição de qual período será utilizado para definição das execuções. ex: Fluxo deve ser executado diariamente às 12 e as 00 horas.
- Timezone – Timezone do aplicativo schedule. Fuso horário que deverá ser utilizado como parâmetro para as execuções do fluxo.
- Período de existência do schedule: Por qual período (data/hora início e fim) o schedule deverá existir. ex: Schedule de 01/01/2021 10:00 até 21/12/2021 10:00.

The screenshot shows the 'Editor Oozie' interface for configuring a 'My Schedule'. The interface includes a 'Que workflow agendar?' section with a dropdown menu set to 'Teste'. Below this is the 'Com que frequência?' section, which includes a frequency selector set to 'Cada dia em 0:0', a 'Fuso horário' dropdown set to 'Etc/GMT', and date/time pickers for 'De' (2021-07-07 09:29) and 'Para' (2021-07-14 09:29). A 'Parameters' section at the bottom shows a dropdown menu and a text input field containing 'e.g. 1, 2, 3, /data/logs, S(coord.nominalTime())'. Red arrows point to the 'Teste' dropdown, the frequency selector, the 'Fuso horário' dropdown, the 'De' and 'Para' date/time pickers, and the 'Parâmetro' dropdown.

## Hands on

### Scheduler

Após salvar o schedule, para que o mesmo seja executado deve-se clicar no botão Play, localizado no canto superior direito da janela. Dessa forma o scheduler irá criar as instâncias de execução de acordo com as definições feitas pelo usuário no momento da criação do scheduler.

The screenshot shows the Oozie Editor interface for a workflow named "teste\_schedule". At the top right, there are four icons: a play button (highlighted with a red arrow), a pencil (edit), a trash can, and an information icon. Below the workflow name, there is a section titled "Que workflow agendar?" with a dropdown menu set to "teste". Another section titled "Com que frequência?" allows scheduling every hour from 0 to 58 minutes. Below this, there is a "Fuso horário" dropdown menu set to "Etc/GMT+2", which is highlighted with a red arrow. At the bottom, there are date and time pickers for "De" (2021-08-03 09:12) and "Para" (2021-08-03 18:00). A "Parameters" section at the bottom includes a link to "Adicionar parâmetro".

## Hands on

### Scheduler

Podemos ver na imagem seguinte o scheduler já iniciado, mostrando o percentual de execução do mesmo, quando foi enviado para execução e o status de execução de cada instância gerada. Podemos observar que foram geradas neste caso 12 instâncias de execução, mas reparem que ele já coloca como próxima execução o primeiro horário subsequente à última instância gerada.

The screenshot shows the Oozie Job Browser interface for a job named 'teste\_schedule'. The job is in a 'WAITING' state with 4% progress. The 'NEXT RUN' is scheduled for 'Tue, 03 Aug 2021 09:36:00'. A table below lists 12 instances of the job, with the first instance being 'RUNNING' and the others 'WAITING'.

Estado	Title	type	errorMessage	missingDependencies	number	errorCode	externalId	id	lastModifiedTime
WAITING	12-03 Aug 2021 09:34:00	schedule-task			12	-		0000808-210710155507184-oozie-oozi-Cj12	Tue, 03 Aug 2021 09:11:51
WAITING	11-03 Aug 2021 09:32:00	schedule-task			11	-		0000808-210710155507184-oozie-oozi-Cj11	Tue, 03 Aug 2021 09:11:51
WAITING	10-03 Aug 2021 09:30:00	schedule-task			10	-		0000808-210710155507184-oozie-oozi-Cj10	Tue, 03 Aug 2021 09:11:51
WAITING	9-03 Aug 2021 09:28:00	schedule-task			9	-		0000808-210710155507184-oozie-oozi-Cj9	Tue, 03 Aug 2021 09:11:51
WAITING	8-03 Aug 2021 09:26:00	schedule-task			8	-		0000808-210710155507184-oozie-oozi-Cj8	Tue, 03 Aug 2021 09:11:51
WAITING	7-03 Aug 2021 09:24:00	schedule-task			7	-		0000808-210710155507184-oozie-oozi-Cj7	Tue, 03 Aug 2021 09:11:51
WAITING	6-03 Aug 2021 09:22:00	schedule-task			6	-		0000808-210710155507184-oozie-oozi-Cj6	Tue, 03 Aug 2021 09:11:51
WAITING	5-03 Aug 2021 09:20:00	schedule-task			5	-		0000808-210710155507184-oozie-oozi-Cj5	Tue, 03 Aug 2021 09:11:51
WAITING	4-03 Aug 2021 09:18:00	schedule-task			4	-		0000808-210710155507184-oozie-oozi-Cj4	Tue, 03 Aug 2021 09:11:51
WAITING	3-03 Aug 2021 09:16:00	schedule-task			3	-		0000808-210710155507184-oozie-oozi-Cj3	Tue, 03 Aug 2021 09:11:51
WAITING	2-03 Aug 2021 09:14:00	schedule-task			2	-		0000808-210710155507184-oozie-oozi-Cj2	Tue, 03 Aug 2021 09:11:51
RUNNING	1-03 Aug 2021 09:12:00	schedule-task			1		0000809-210710155507184-oozie-oozi-W	0000808-210710155507184-oozie-oozi-Cj1	Tue, 03 Aug 2021 09:12:00

## Hands on

### Scheduler

Durante ou mesmo após a execução o usuário pode clicar em uma instância para verificar maiores detalhes da execução.

The screenshot displays the Oozie Job Browser interface. At the top, there are navigation tabs: Job Browser, Jobs, Queries, Workflows, Schedules, Bundles, and SLAs. A notification in the top right corner states "1 Hive jobs are running as the 'hive' user".

The main content area shows details for a job with ID "0000808-210710155507184-oozie-oozi-C" and name "teste". The job is in the "SUCCEEDED" state, run by the user "oficina\_hue", and completed in 30 seconds. A progress bar shows 100% completion. The job was submitted on August 3, 2021.

Navigation options include "Graph", "Properties", "Logs", "Tarefas", and "XML". The "Graph" tab is highlighted with a red box and a red arrow. To the right, there are control buttons: "Returne", "Suspend", "Renun", and "Eliminar".

Below the job details, a workflow diagram is visible, showing a sequence of steps: a Hive job named "teste.sql" and a final step represented by a square icon. Red arrows point to the "Graph" tab and the "100%" progress bar.



**Dúvidas ?**



# Instrutores

Bruno Mariz

[bruno.oliveira@serpro.gov.br](mailto:bruno.oliveira@serpro.gov.br)

Anderson Ceo

[anderson.ceo@serpro.gov.br](mailto:anderson.ceo@serpro.gov.br)

SUPAI - Janeiro/2023

 /serprobrasil

 @serprobrasil

 @serpro

 /serpro

 serpro.gov.br