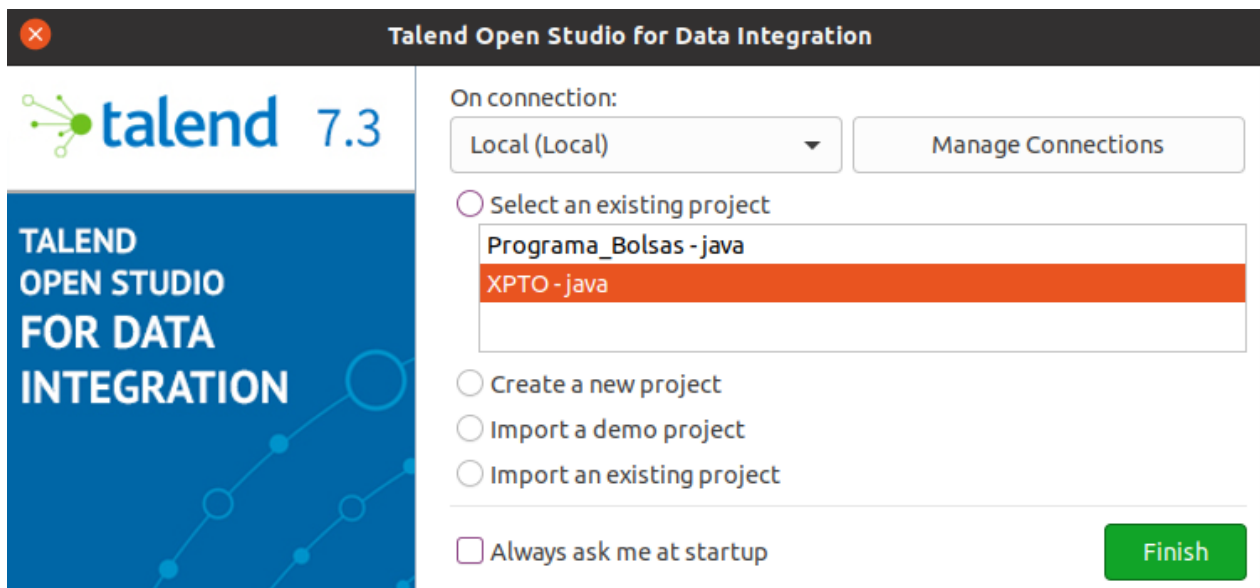


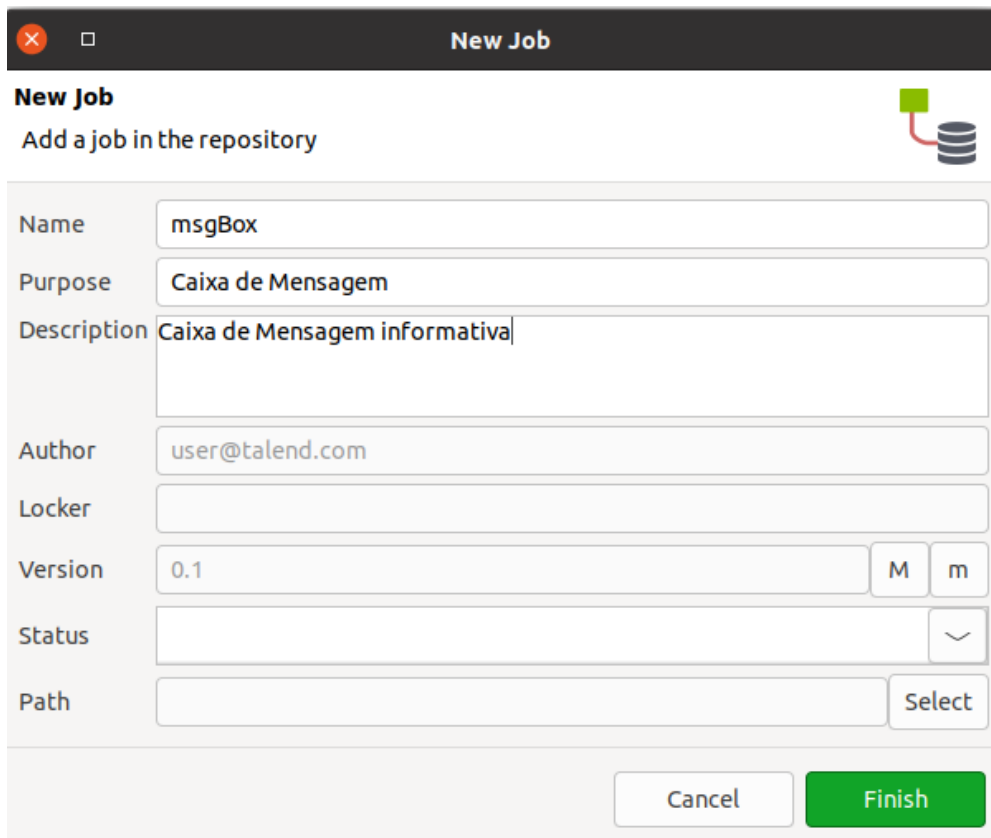
SEMANA 07 PROGRAMA DE BOLSAS COMPASSO
RAFAEL IGNAULIN

Instalando e configurando o Talend



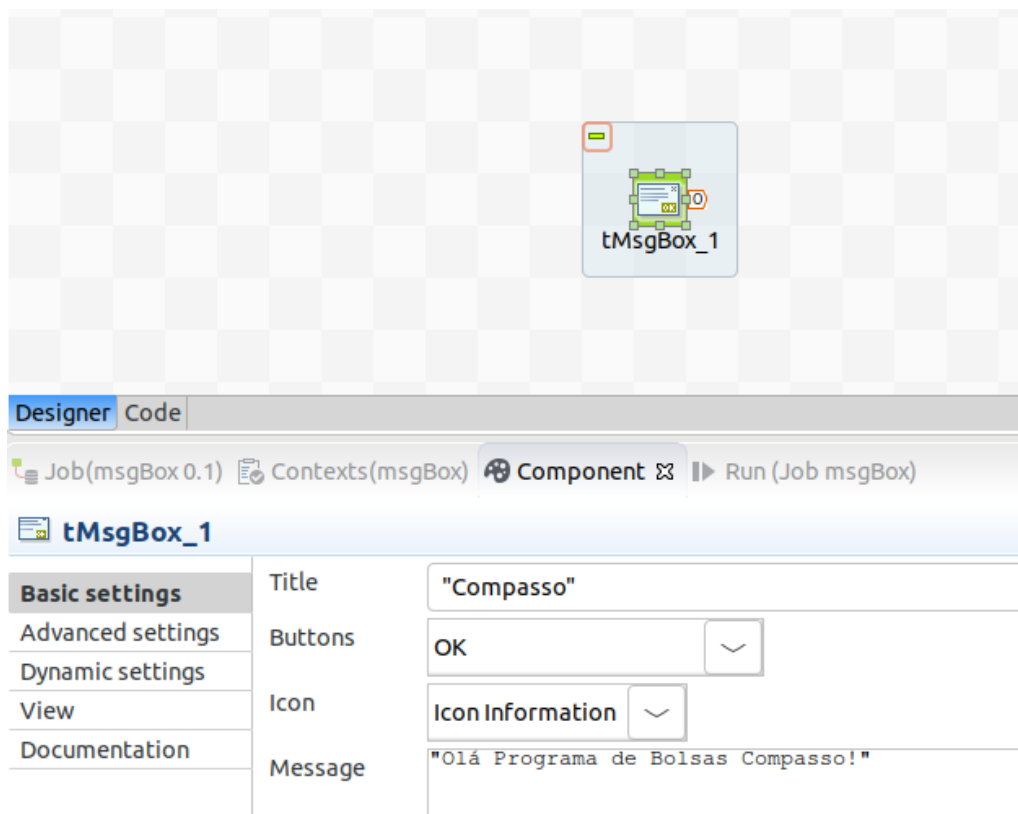
The screenshot shows the 'Talend Open Studio for Data Integration' installation window. On the left is a blue sidebar with the Talend 7.3 logo and the text 'TALEND OPEN STUDIO FOR DATA INTEGRATION'. The main area has a dark header with the title. Below the header, there's a section 'On connection:' with a dropdown menu set to 'Local (Local)' and a 'Manage Connections' button. Below this, there are three radio button options: 'Select an existing project' (selected), 'Create a new project', and 'Import a demo project'. Under 'Select an existing project', there's a list box showing 'Programa_Bolsas - java' and 'XPTO - java', with 'XPTO - java' highlighted in orange. Below the list box are two more radio button options: 'Import an existing project' and 'Always ask me at startup'. At the bottom right is a green 'Finish' button.

- Talend já instalado, e criado os dois projetos (lab01 e lab03)

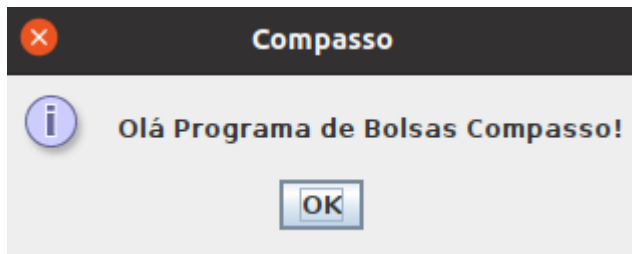


The screenshot shows the 'New Job' dialog box. It has a dark header with the title 'New Job'. Below the header, there's a section 'New Job' with a sub-header 'Add a job in the repository'. To the right of the sub-header is a small icon of a database cylinder. Below this, there are several input fields: 'Name' (containing 'msgBox'), 'Purpose' (containing 'Caixa de Mensagem'), 'Description' (containing 'Caixa de Mensagem informativa'), 'Author' (containing 'user@talend.com'), 'Locker' (empty), 'Version' (containing '0.1' with 'M' and 'm' buttons next to it), 'Status' (empty with a dropdown arrow), and 'Path' (empty with a 'Select' button). At the bottom right are two buttons: 'Cancel' and 'Finish'.

- Criando primeiro job para usar o msgBox



- Criado um MsgBox com uma mensagem específica e um título



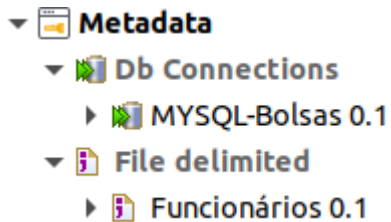
- Job Executado, mostrada a mensagem

Caso 2: Ler um arquivo CSV e inserir no Banco de dados

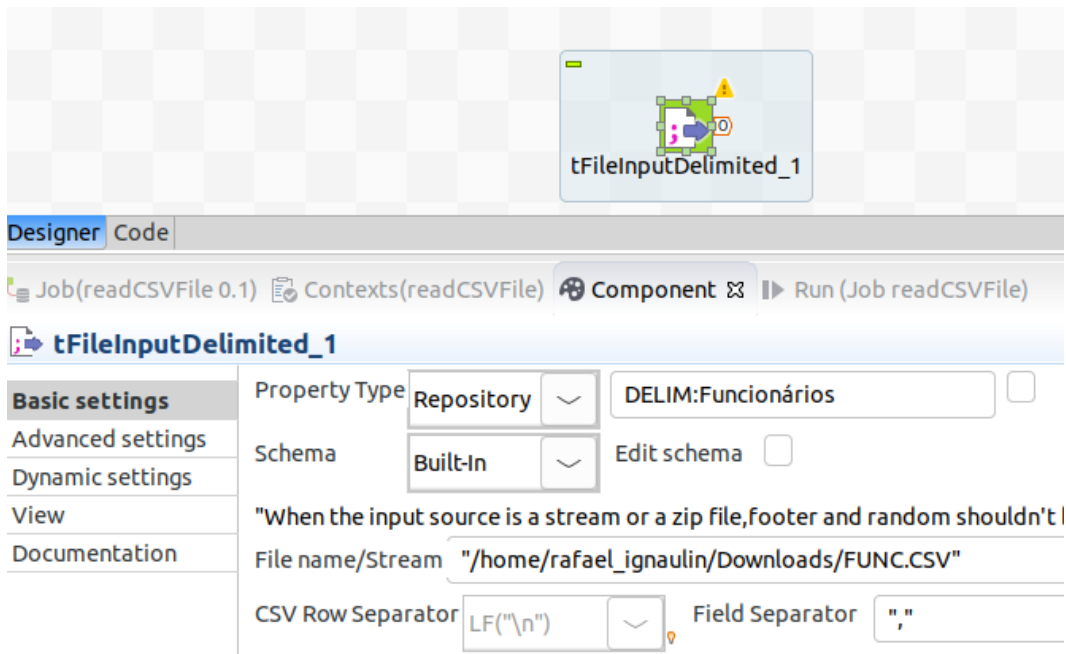
- Inicialmente criamos uma tabela no banco de dados MYSQL com as seguintes colunas:

```
CREATE TABLE FUNCIONARIO(
    CodFun INTEGER NOT NULL,
    Nome VARCHAR(50) NOT NULL,
    Nascimento DATE NOT NULL,
    Contratacao DATE NOT NULL,
    PRIMARY KEY (CodFun)
);
```

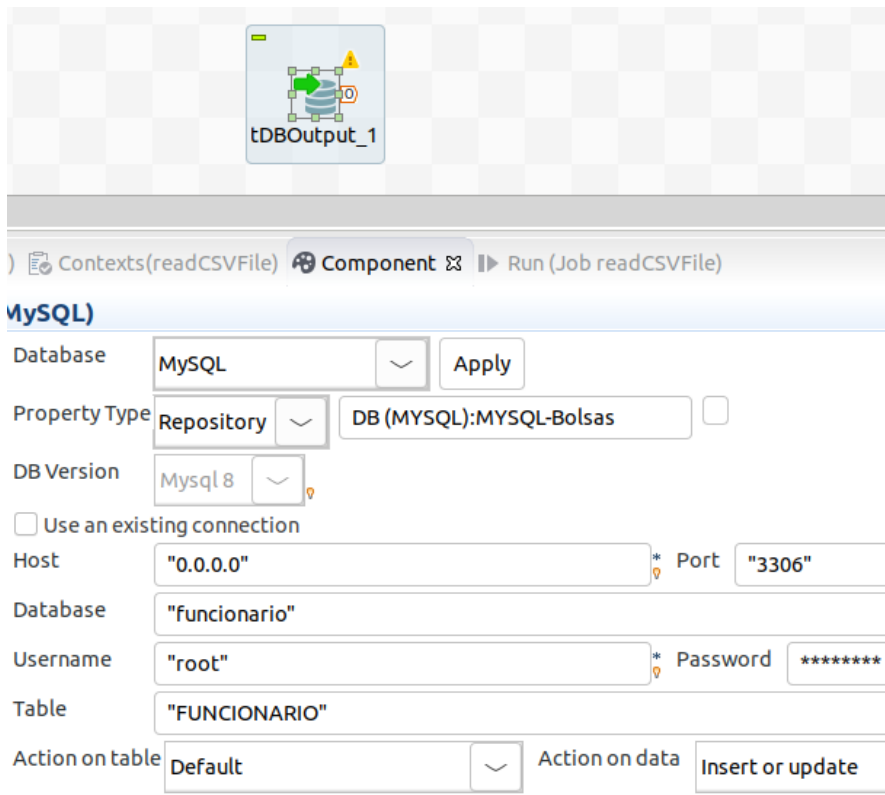
- Criado os metadados para entrada de dados (arquivo CSV), e saída de e dados (conexão com Database)



- Criado o FileInput em formato CSV, e configurado com o path do arquivo de entrada



- Criando o DB Output, para colocar dados no banco de dados



tDBOutput_1

Contexts(readCSVFile) Component Run (Job readCSVFile)

MySQL

Database: MySQL Apply

Property Type: Repository DB (MYSQL):MYSQL-Bolsas

DB Version: Mysql 8

☐ Use an existing connection

Host: "0.0.0.0" Port: "3306"

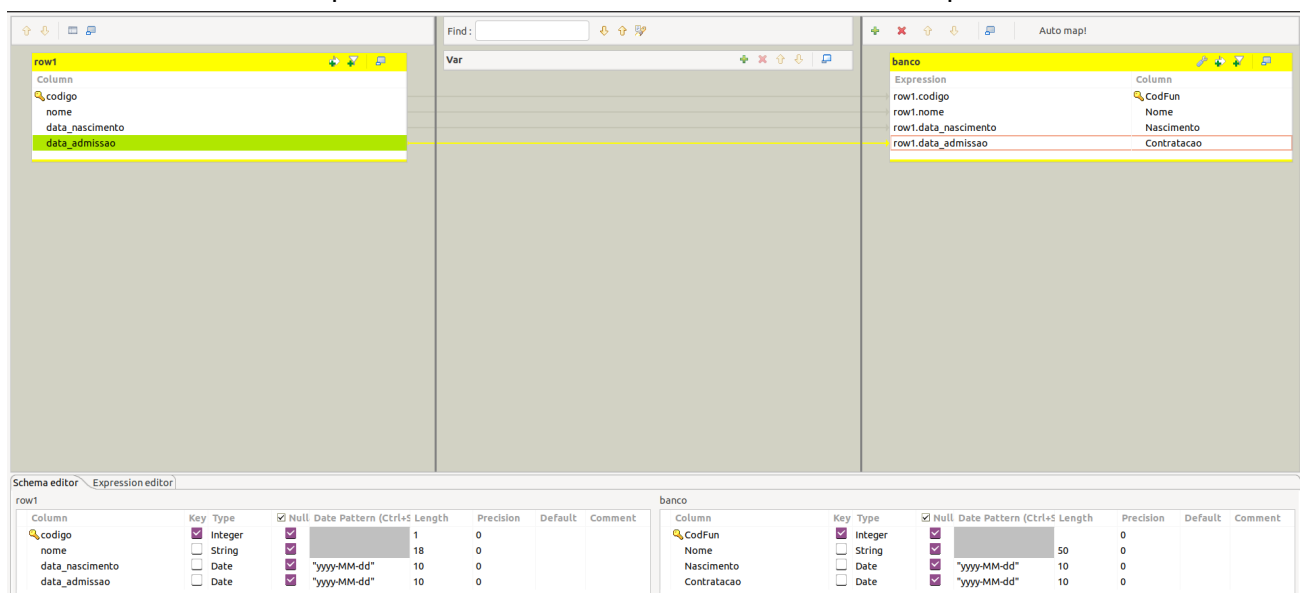
Database: "funcionario"

Username: "root" Password: "*****"

Table: "FUNCIONARIO"

Action on table: Default Action on data: Insert or update

- Criando o TMAP para fazer a conversão dos schemas do CSV para o banco de dados.



Find: Var Auto map!

row1

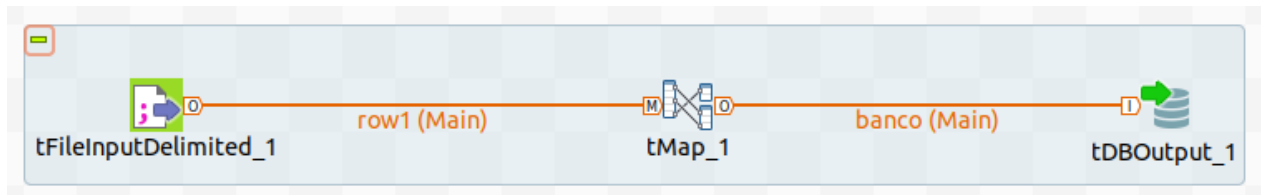
| Column | Expression | Column |
|-----------------|----------------------|-------------|
| codigo | row1.codigo | CodFun |
| nome | row1.nome | Nome |
| data_nascimento | row1.data_nascimento | Nascimento |
| data_admissao | row1.data_admissao | Contratacao |

banco

| Column | Key | Type | Null | Date Pattern (Ctrl+S) | Length | Precision | Default | Comment |
|-------------|-------------------------------------|---------|-------------------------------------|-----------------------|--------|-----------|---------|---------|
| CodFun | <input checked="" type="checkbox"/> | Integer | <input checked="" type="checkbox"/> | | 1 | 0 | | |
| Nome | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | | 18 | 0 | | |
| Nascimento | <input type="checkbox"/> | Date | <input checked="" type="checkbox"/> | "yyyy-MM-dd" | 10 | 0 | | |
| Contratacao | <input type="checkbox"/> | Date | <input checked="" type="checkbox"/> | "yyyy-MM-dd" | 10 | 0 | | |

Schema editor Expression editor

- O job completo ficou com esse formato:



- Após executar o job, temos os resultados que estavam escritos no CSV:

| CodFun | Nome | Nascimento | Contratacao |
|--------|--------------------|------------|-------------|
| 1 | Ana Paula | 1986-05-01 | 2017-10-20 |
| 2 | Carolina Ramos | 1975-02-08 | 1998-07-09 |
| 3 | Manuela dos Santos | 1985-11-20 | 2013-11-20 |
| 4 | Rafael Ignaulin | 1010-10-10 | 2020-08-20 |
| 5 | COMPASSO_USER | 2021-06-29 | 2021-06-29 |

2) DESAFIO XPTO : CRIAÇÃO DOS BUCKETS


| | |
|-----------------|---|
| xpto-raw-batch | Leste dos EUA (Norte da Virgínia) us-east-1 |
| xpto-raw-stream | Leste dos EUA (Norte da Virgínia) us-east-1 |
| xpto-refined | Leste dos EUA (Norte da Virgínia) us-east-1 |

- Em sequência, foram criados três buckets para usar no caso de uso da empresa XPTO. Os dois primeiros serão usados para a parte inicial de ingestão de dados, em formato Batch (lotes) e Streaming respectivamente. O último bucket servirá para guardar os dados depois da data de processamento.

3) DESAFIO XPTO: CRIAÇÃO DO JOB - INGESTÃO BATCH

- Criamos um job, para execução do processo
- Primeiro criamos o metadata do Banco de dados, com todas as suas informações.

Update Database Connection - Step 2/2

 You must press the Check Button to check the Database Setting



| | |
|---|--|
| DB Type | MySQL |
| Db Version | MySQL 8 |
| String of Connection | jdbc:mysql://hive-metastore.cnpzwadswd70.us-east-1.rds.amazonaws.com:3306/xpto |
| Login | programa_bolsas |
| Password | •• |
| Server | hive-metastore.cnpzwadswd70.us-east-1.rds.amazonaws.com |
| Port | 3306 |
| DataBase | xpto |
| Additional parameters | noDatetimeStringSync=true&serverTimezone=America/Sao_Paulo&useTimezone=true |
| <div>Test connection <input type="button" value="v"/></div> | |

- Depois criamos o metadata do CSV de saída, com todas as suas informações.

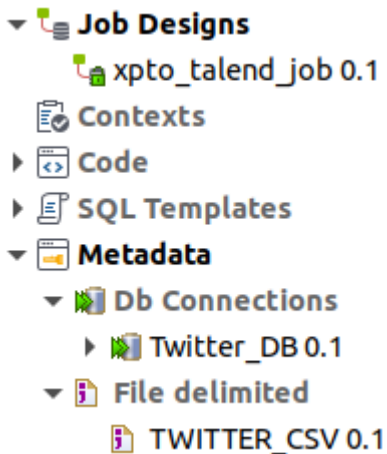
File - Step 3 of 3

Update an existing Metadata File on repository
Define the setting of the parse job



| | | |
|---|--|---|
| File Settings Encoding: <input type="text" value="UTF-8"/> Field Separator: <input type="text" value="Corresponding Character"/> <input type="text" value=","/> Row Separator: <input type="text" value="Corresponding Character"/> <input type="text" value="\n"/> | | Rows To Skip If any rows must be ignored, specify the following parameters Header: <input type="checkbox"/> <input type="text"/> Footer: <input type="checkbox"/> <input type="text"/> <input type="checkbox"/> Skip empty row |
| Escape Char Settings <input checked="" type="radio"/> CSV <input type="radio"/> Delimited Escape Char: <input type="text" value="\"/> Text Enclosure: <input "="" type="text" value="\"/> <input type="checkbox"/> Split row before field | | Limit Of Rows If the number of lines must be limited, specify this number Limit: <input type="text"/> |

- Aqui a lista contendo os 2 metadata que acabamos de criar.



- Colocamos o bloco DB Connection, para colocar as credenciais do banco de dados que iremos utilizar.

tDBConnection_1(MySQL)

| | | | |
|-------------------|---------------|--|---|
| Basic settings | Database | MySQL | Apply |
| Advanced settings | Property Type | Repository | DB (MySQL):Twitter_DB |
| Dynamic settings | DB Version | MySQL 8 | |
| View | Host | "hive-metastore.cnpzwadswd70.us-east-1.rds.am" | Port "3306" |
| Documentation | Database | "xpto" | Additional JDBC Parameters "noDatetimeStringSync=true&se" |
| | Username | "programa_bolsas" | Password ***** |

- Colocamos o bloco DB Input, para fazer uma query diretamente no banco de dados, que já está devidamente configurado no passo anterior.

tDBInput_1(MySQL)

Basic settings

Database: MySQL Apply

☐ Use an existing connection Component List: tDBConnection_1

Schema: Built-In Edit schema

Table Name: "xpto"

Query Type: Built-In Guess Query Guess schema

Query: *SELECT * FROM TWITTER_ELEICOES;*

- Colocamos o bloco de File Output em CSV, para criar um arquivo que será importado futuramente no Amazon S3.

tFileOutputDelimited_1

Basic settings

Property Type: Repository DELIM:TWITTER_CSV

☐ Use Output Stream

File Name: "/home/rafael_ignaulin/Desktop/COMPASSO/Sprint_4/week_07/xpto_csv/twitter.csv"

☐ Use OS line separator as row separator when CSV Row Separator is set to CR,LF or CRLF.

CSV Row Separator: LF("\n") Field Separator: ","

☐ Append ☐ Include Header ☐ Compress as zip file

Schema: Built-In Edit schema Sync columns

- Criamos o Map, e conectamos o esquema utilizado no banco de dados para um formato final que será usado no S3.

Schema editor

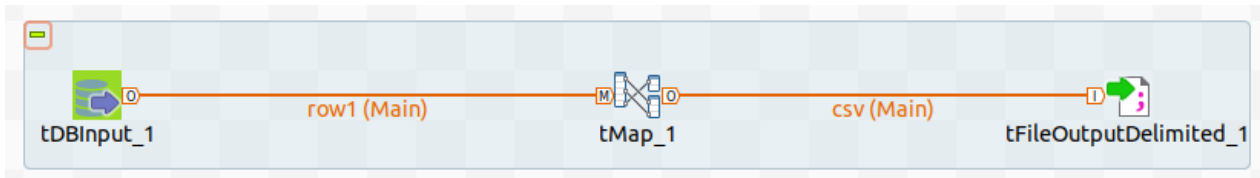
row1

| Column | Key | Type | Null | Date Pattern (Ctrl+S) | Length | Precision | Default | Comment |
|------------|-----|--------|-------------------------------------|-----------------------|--------|-----------|---------|---------|
| id | | Long | <input checked="" type="checkbox"/> | | | 0 | | |
| tweet_text | | String | <input type="checkbox"/> | | | 0 | | |
| tweet_date | | Date | <input checked="" type="checkbox"/> | yyyy-MM-dd HH:mm | 300 | 0 | | |

CSV

| Column | Key | Type | Null | Date Pattern (Ctrl+S) | Length | Precision | Default | Comment |
|------------|-----|--------|-------------------------------------|-----------------------|--------|-----------|---------|---------|
| id | | Long | <input checked="" type="checkbox"/> | | | 0 | | |
| tweet_text | | String | <input type="checkbox"/> | | | 0 | | |
| tweet_date | | Date | <input checked="" type="checkbox"/> | yyyy-MM-dd HH:mm | 300 | 0 | | |

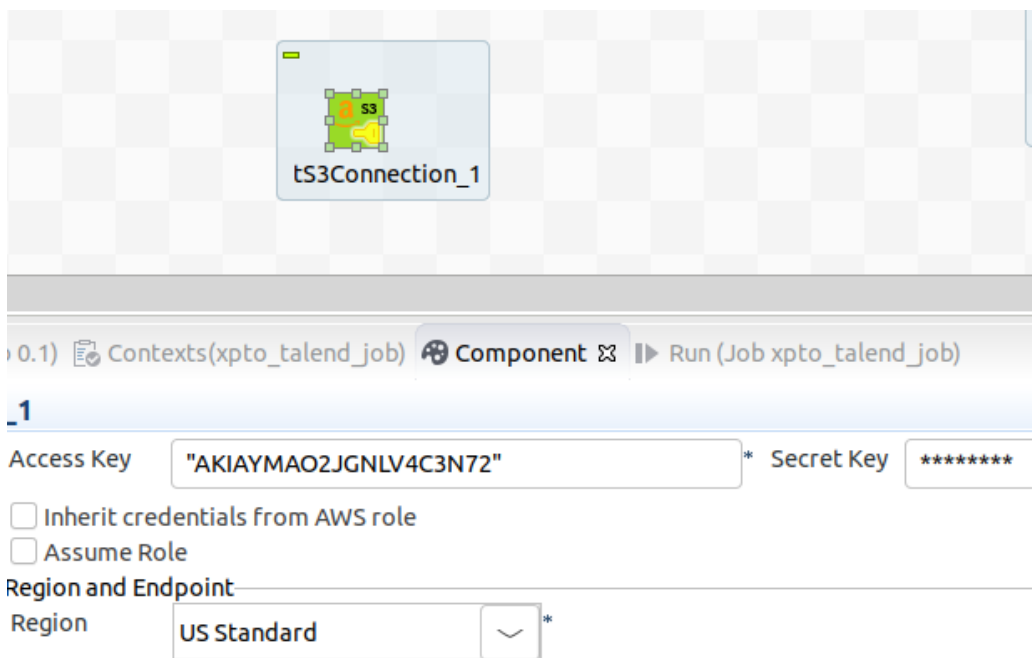
- Essa parte inicial de gravar os dados do banco de dados em um CSV ficou assim:



- Criando o arquivo twitter.csv (com os dados do banco de dados)

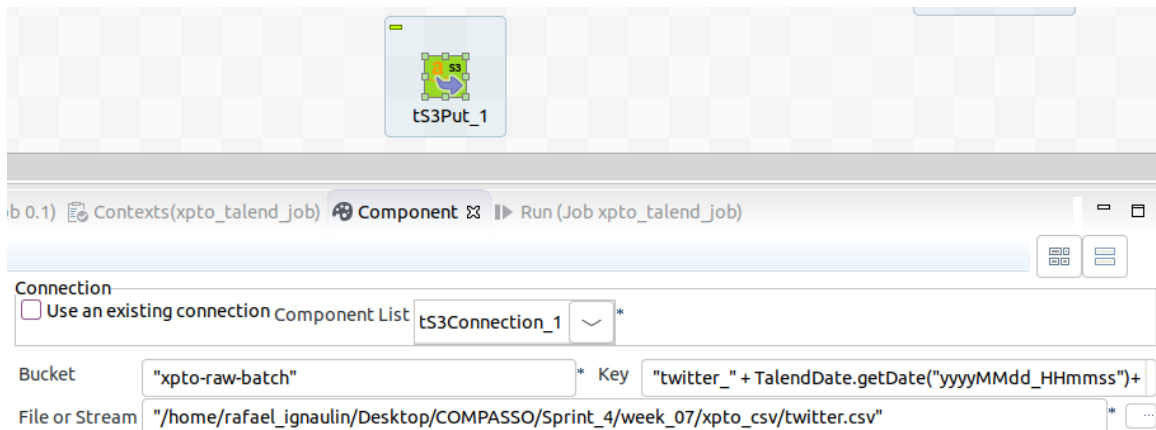


Agora iremos fazer a parte do upload desse arquivo para o Amazon S3:



- Inicialmente criamos a conexão com os serviços da Amazon. Primeiramente, criamos uma chave de acesso nas configurações de segurança da AWS e colocamos ela no s3 Connection.

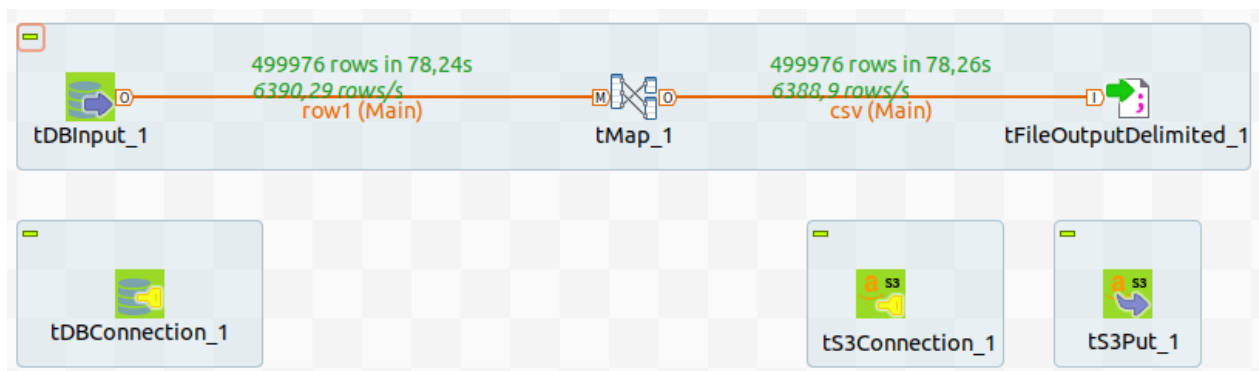
Usamos um S3 Put também para jogar arquivos dentro do S3:




- Aqui colocamos o bucket que será utilizado para o upload, o arquivo que será copiado e o nome do arquivo dentro do bucket, na "key".

*OBS: Foi criada uma data de timestamp concatenada na string do arquivo que será colocado no S3, para não ocorrer as substituições de arquivos antigos, assim mantendo a integridade e a imutabilidade de uma camada batch, como um requisito.

- O job inteiro ficou dessa forma:



- Com o job executado, será criado aquele mesmo arquivo "twitter.csv" comentado alguns parágrafos atrás, porém será colocado este mesmo arquivo dentro do bucket "xpto-raw-batch" com o seu respectivo timestamp.

| Nome |
|---|
|  twitter_20210702_143152.csv |