



Relatório Integração de Sistemas de Informação

Trabalho Prático 1

Trabalho Realizado por:
Bruno Miranda nº26014

https://github.com/al26014/ISI_26014

25 de Outubro de 2024

Índice

1.	Introdução	3
2.	Problema	3
3.	Workflows	4
3.1.	Normalização	4
3.2.	Pagamentos Realizados	4
3.3.	Criação de Base de Dados	5
3.4.	Pagamentos Pendentes	5
3.5.	Pagamentos Fora de Prazo	6
3.6.	Transformações em XML e JSON	6
3.7.	Envio de Email	7
4.	Vídeo com demonstração	8
5.	Conclusão e Trabalhos Futuros	9
6.	Bibliografia	10

Índice de Figuras

Fig. 1 Normalização	4
Fig. 2 Pagamentos Realizados por Empresa	4
Fig. 3 Criação de Base de Dados	5
Fig. 4 Pagamentos Pendentes	5
Fig. 5 Pagamentos Fora de Prazo	6
Fig. 6 Transformação em XML e JSON	6
Fig. 7 Envio de Email	7

1. Introdução

Este trabalho prático insere-se no âmbito da disciplina de Integração de Sistemas de Informação, cujo foco é a aplicação e experimentação de ferramentas em processos de ETL (Extract, Transformation, and Load). Depois de avaliar as alternativas, foi decidido que para este trabalho seria utilizado o KNIME como a plataforma de desenvolvimento. Como tema do trabalho, foi escolhido a implementação de um processo ETL para automatizar a gestão de faturas e pagamentos numa empresa.

2. Problema

O principal problema que este projeto pretende resolver é a automação no processamento de faturas e pagamentos em grandes empresas. Muitas vezes, empresas enfrentam dificuldades em monitorar o estado dos pagamentos e identificar faturas que já passaram da data de vencimento. Isso pode gerar confusões e atrasos, resultando em multas para a empresa ou dificuldades no planejamento financeiro.

Neste trabalho, o objetivo é criar um sistema que:

- Verifique automaticamente se as faturas foram pagas e quais estão pendentes.
- Aplique penalizações em faturas não pagas dentro do prazo, aumentando o valor por cada dia de atraso.
- Agregue e organize os dados de forma que seja possível visualizar facilmente o status de todos os pagamentos.
- Exporte os resultados em formatos como XML ou JSON, para facilitar a integração com outros sistemas.

O desafio está em garantir que o sistema seja simples de usar, permitindo uma gestão eficaz dos dados de faturação e pagamento, sem necessidade de grande intervenção manual.

3. Workflows

Os diferentes workflows do projeto estão organizados em blocos específicos para facilitar a gestão e entendimento. Abaixo, segue-se a explicação de cada workflow representado nos diagramas.

1. Normalização

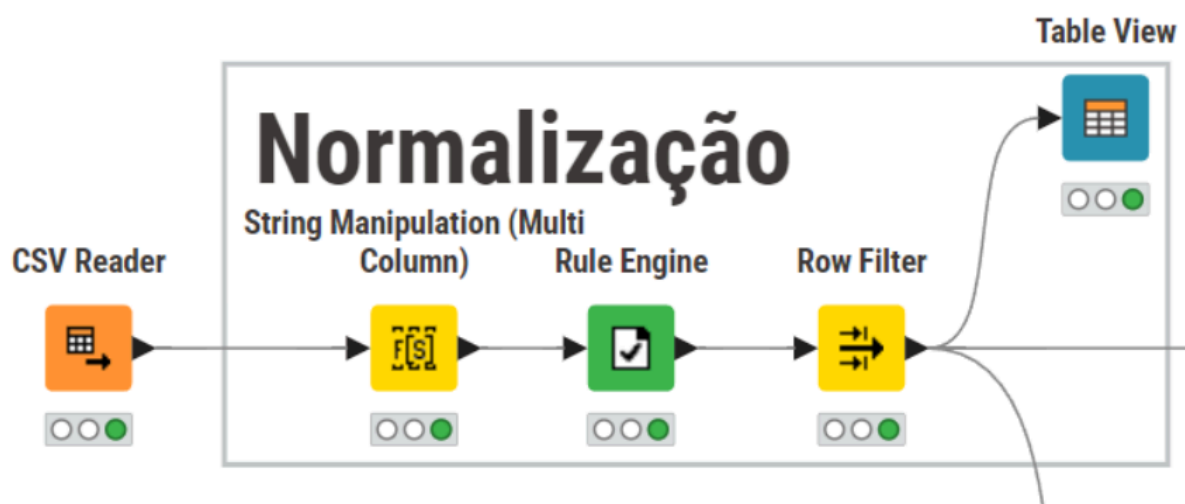


Fig. 1 Normalização

Este workflow é responsável por garantir que os dados lidos do ficheiro CSV estão no formato correto. Inclui a manipulação de strings (por exemplo, remoção de acentos ou espaços extras) e o uso de um **Rule Engine** para garantir que as informações de telefone estão no formato português. Uma vez validados, os dados seguem para o próximo estágio de processamento.

2. Pagamentos Realizados por Empresa

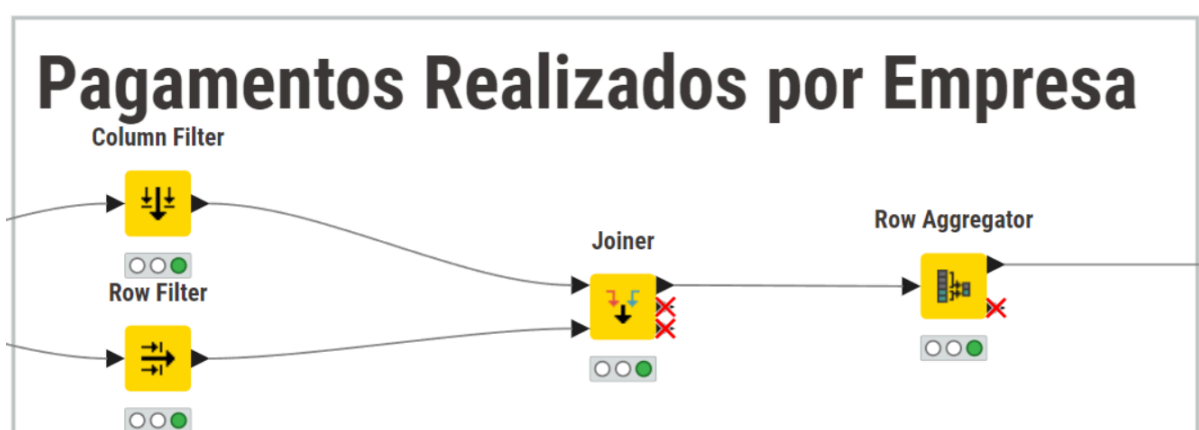


Fig. 2 Pagamentos Realizados por Empresa

Neste bloco, os dados são filtrados para identificar quais faturas foram pagas. Através do uso de um **Joiner** e um **Row Aggregator**, os dados são consolidados, permitindo uma visão agregada de todos os pagamentos realizados.

3. Criação de Base de Dados

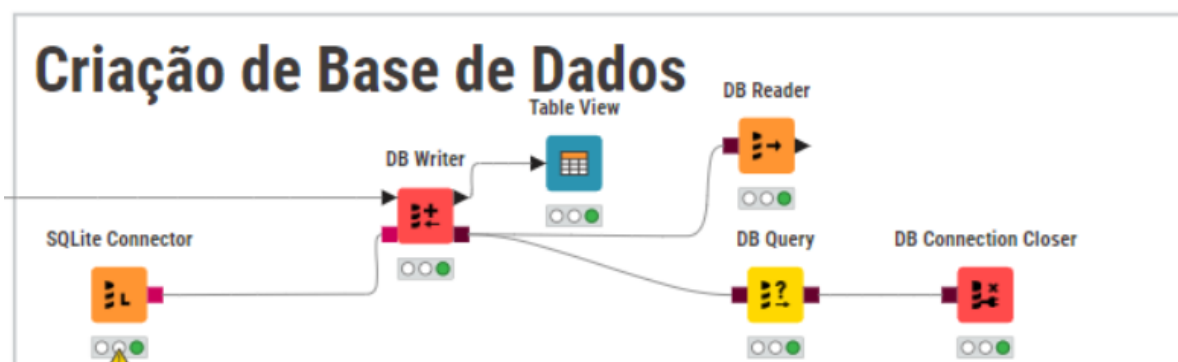


Fig. 3 Criação de Base de Dados

Este workflow é responsável por armazenar os dados num sistema de base de dados. Utiliza o **DB Writer** para escrever as informações diretamente na base de dados, facilitando o armazenamento e a futura consulta dos dados. O **DB Reader** é depois utilizado para verificar se o **DB Writer** funcionou sem erros. Também é utilizado um **DB Query** o qual apenas recolhe toda a informação de **DB Writer** de forma a que funcione o **Db Reader**, pois se estiverem a ser executados ao mesmo tempo o **DB Reader** e o **DB Connection Closer**, o **DB Connection Closer** irá parar o **Db Reader** de funcionar. Além disso, o **DB Connection Closer** é utilizado para garantir o encerramento adequado da conexão com a base de dados após a escrita dos dados, evitando problemas de conexão persistente ou bloqueio de recursos.

4. Pagamentos Pendentes

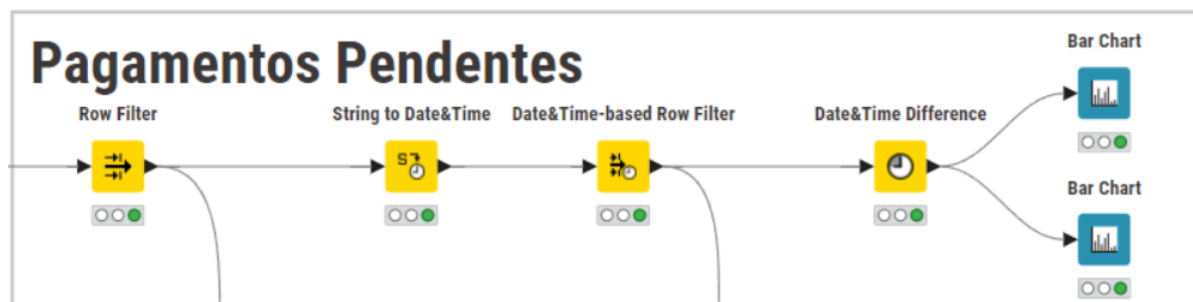


Fig. 4 Pagamentos Pendentes

Este workflow lida com as faturas que ainda não foram pagas. Um **Row Filter** identifica essas faturas, e o **Date&Time-based Row Filter** ajuda a calcular os dias restantes até o vencimento. Por fim, gráficos como **Bar Charts** são gerados para representar visualmente esses dados.

5. Pagamentos Fora de Prazo

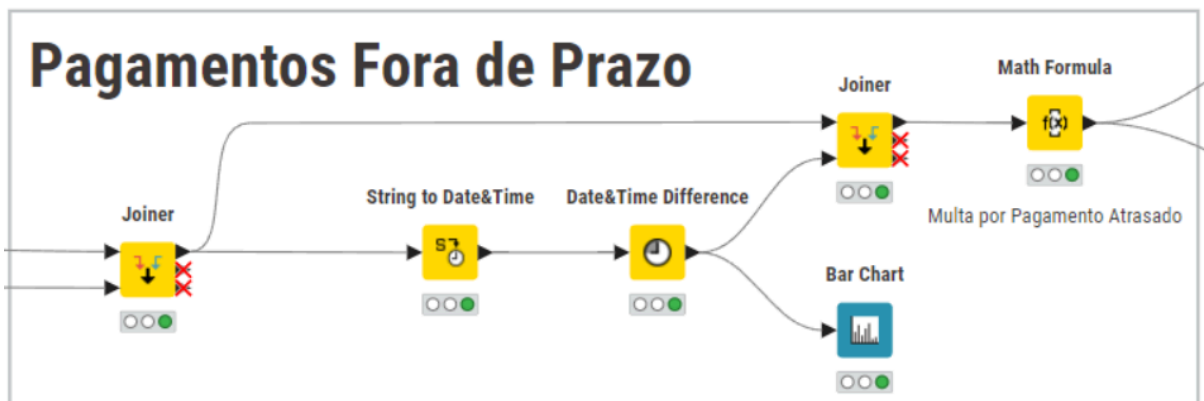


Fig. 5 Pagamentos Fora de Prazo

Aqui, as faturas que já passaram da data de vencimento são tratadas. O **Joiner** recebe o **Row Filter** e o **Date&Time-based Row Filter**, de modo a separar os pagamentos por pagar os quais ainda estão dentro de prazo e os que não, obtendo os que estão fora de prazo. Depois, a diferença de dias é calculada com o **Date&Time Difference**, e uma fórmula matemática (**Math Formula**) aplica uma multa com base no número de dias de atraso (2% por dia). O Gráfico é usado para visualizar o número de dias que já passaram desde quando deveriam ter sido efetuados os pagamentos.

6. Transformação em XML e JSON



Fig. 6 Transformação em XML e JSON

Este workflow converte os dados em formatos **XML** e **JSON** para facilitar a integração com outros sistemas ou o envio por APIs. São usados os nós **XML Column Combiner**, **XML Row Combiner**, **XML Writer**, **XML to JSON**, e finalmente o **JSON Writer**.

7. Envio de Email

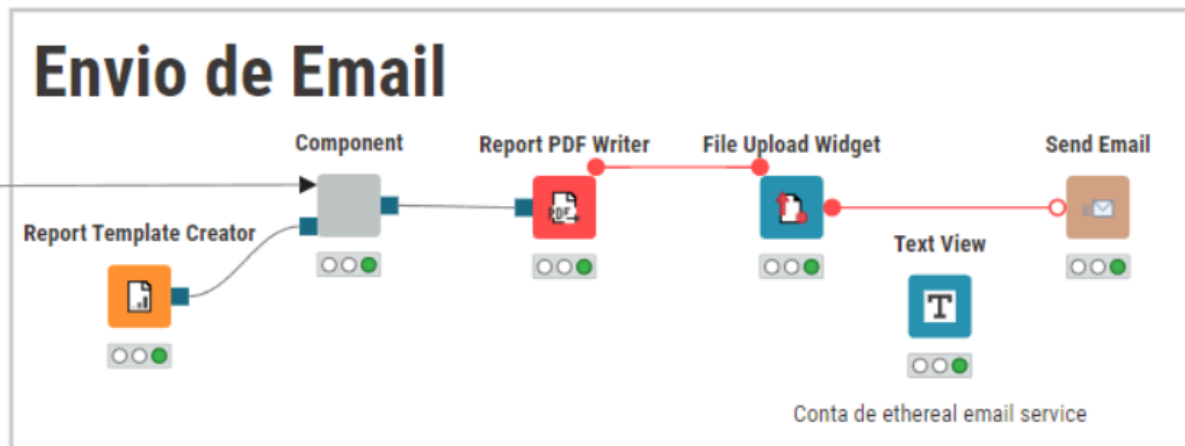


Fig. 7 Envio de Email

Para automatizar a notificação dos relatórios gerados, este bloco cria um relatório PDF e o envia por email. O **Report Template Creator** gera o relatório, o **File Upload Widget** é usado para carregar o arquivo, e o **Send Email** envia o relatório para o destinatário desejado. Este e-mail é enviado para uma conta de e-mail gerada através do Ethereum, um serviço de e-mail que permite o envio de mensagens e anexos para uma conta de e-mail temporária. O **Text View** serve para guardar a informação da conta criada para acesso futuro.

4. Vídeo com demonstração (QR Code)

Abaixo encontra-se um vídeo o qual demonstra o funcionamento do projeto, sem mostrar o email a ser enviado, pois isso estará em outro qr code mais abaixo.



Abaixo encontra-se a demonstração do envio do email e o que é visto em Ethereum.email.



5. Conclusão e Trabalhos futuros

Em conclusão, o processo de ETL desenvolvido conseguiu automatizar eficazmente o processo de faturas e pagamentos, incluindo a validação de dados e o cálculo automático de multas. Como trabalho futuro, seria de interesse integrar APIs externas para obter dados adicionais e explorar mais profundamente técnicas de visualização de dashboards para relatórios financeiros em tempo real.

6. Bibliografia

Ethereal Email

Knime Learning Center

Documentação fornecida pelo professor