Alunos:
RM357067 - Edinaldo Rodrigues de Oliveira Junior
RM358158 - Henrique Cardoso
RM358067 - Jefferson de Souza Santos
RM357344 - Walace Vinicius Silva dos Santos

Data Architecture, Integration and Ingestion | QuantumFinance

Parte 1 - Framework de Dados proposto

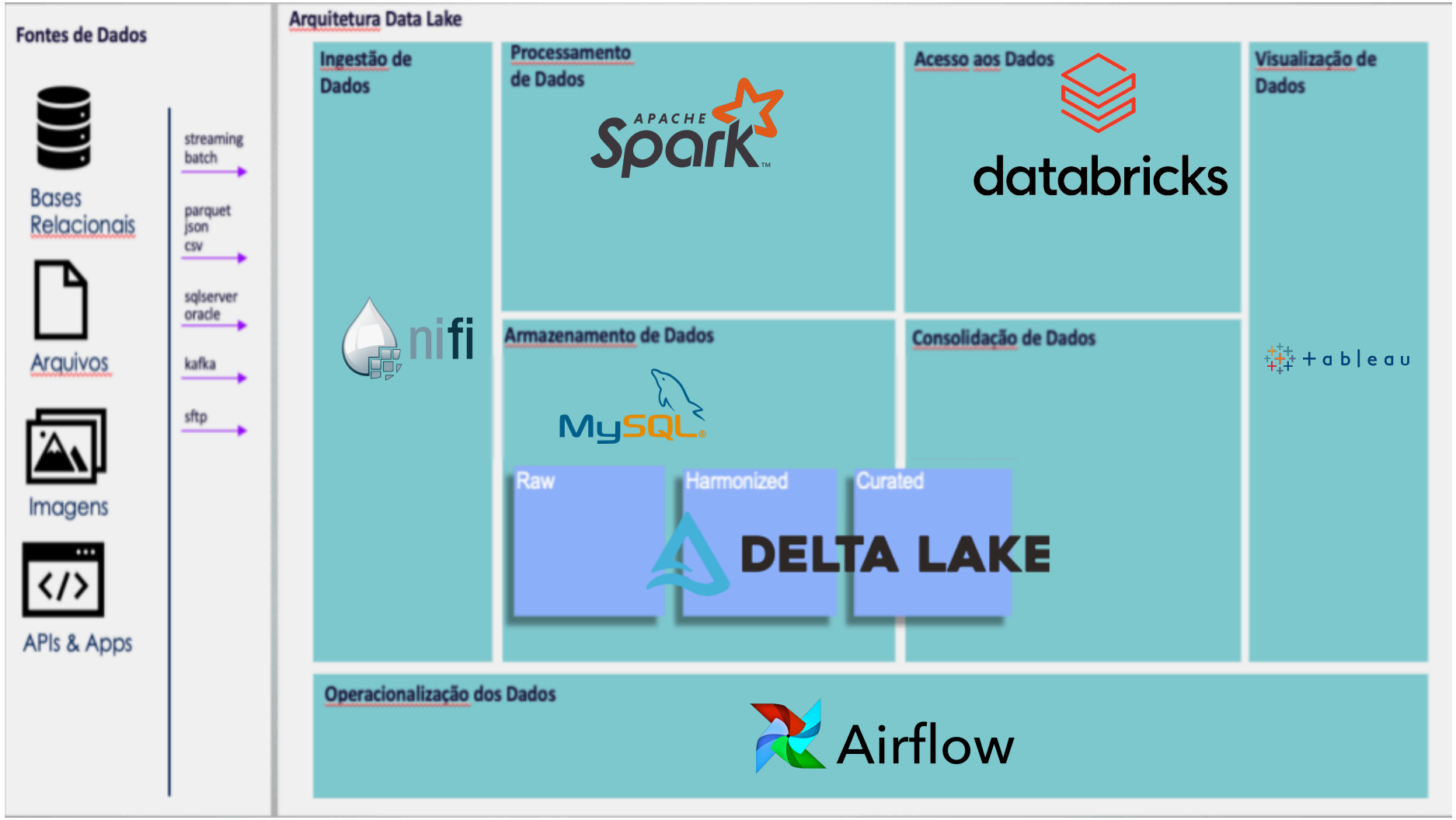
A QuantumFinance, sendo uma startup em ascensão no mercado financeiro, enfrenta desafios como a necessidade de escalabilidade, segurança e disponibilidade dos dados, fundamentais para operações financeiras robustas e confiáveis. Além disso, o cenário de inovação exige um ambiente de dados que suporte análises avançadas, inteligência artificial e tomadas de decisão ágeis.

A decisão de adotar uma arquitetura **Data Lakehouse** surge como a resposta ideal para essas demandas. Essa abordagem combina a flexibilidade do Data Lake, permitindo ingestão de dados não estruturados e semi-estruturados, com a governança e o desempenho analítico característicos de um Data Warehouse. Com essa arquitetura, a QuantumFinance será capaz de:

1. **Garantir Segurança e Confiabilidade:** Suporte a transações ACID e controle de acesso rigoroso, essenciais para operações financeiras críticas.
2. **Habilitar Escalabilidade e Agilidade:** Processamento distribuído para grandes volumes de dados e integração com ferramentas modernas de análise e ciência de dados.
3. **Facilitar a Inovação:** Fornecer acesso eficiente aos dados para cientistas, analistas e desenvolvedores, acelerando a entrega de insights e soluções personalizadas.
4. **Manter a Governança e Conformidade:** Implementação de políticas claras de gestão e qualidade de dados, fundamentais para atender regulamentos financeiros.

A escolha do Data Lakehouse não apenas atende às necessidades atuais da QuantumFinance, mas também prepara a empresa para um futuro de crescimento e competitividade em um setor dinâmico e altamente regulamentado.

Framework de dados:



Detalhe do FrameWork de Dados

1. Ingestão de Dados: Apache NiFi

Justificativa: O Apache NiFi é uma ferramenta poderosa para ingestão, roteamento e transformação de dados em tempo real e batch. Ele permite que você conecte múltiplas fontes de dados (como MySQL, APIs, arquivos CSV, entre outras) de forma simples e eficiente. Sua interface visual facilita a criação de fluxos de dados complexos sem a necessidade de código extensivo, o que é especialmente útil em um contexto de startup com necessidades ágeis e rápidas.

Conexões: NiFi pode facilmente integrar com MySQL (para dados transacionais) e Delta Lake (para dados analíticos) através de processadores de ingestão, permitindo uma movimentação fluida dos dados para os sistemas de armazenamento adequados.

2. Processamento de Dados: Apache Spark

Justificativa: O Apache Spark é uma plataforma de processamento de dados distribuída e de alto desempenho, capaz de lidar com grandes volumes de dados de maneira rápida e eficiente. Ele suporta tanto processamento batch quanto streaming, o que o torna flexível para diferentes necessidades, como transformar dados históricos e realizar análises em tempo real.

Conexões: O Spark se integra nativamente com Delta Lake, o que permite processar dados de forma eficiente, mantendo a integridade e a consistência dos dados. Além disso, o Spark pode ser executado na plataforma Databricks, otimizando o uso dos recursos de computação em nuvem.

3. Armazenamento de Dados: MySQL e Delta Lake

Justificativa:

MySQL: Utilizado para armazenar dados transacionais, como informações sobre transações financeiras, clientes e registros de sistemas operacionais. O MySQL é um banco relacional robusto, amplamente utilizado e conhecido por sua consistência e integridade de dados em sistemas de alto volume de transações (OLTP).

Delta Lake: Servindo como o repositório para dados analíticos e não estruturados (por exemplo, logs de transações, dados históricos e dados semi estruturados), o Delta Lake proporciona consistência ACID sobre dados armazenados em formato de Data Lake. Ele é ideal para processar e armazenar grandes volumes de dados de forma escalável e resiliente, com suporte a atualizações e backfills.

4. Consolidação de Dados: Delta Lake

Justificativa: O Delta Lake é a camada de consolidação de dados em uma arquitetura de Data Lakehouse, proporcionando o equilíbrio ideal entre segurança e flexibilidade. Ele oferece transações ACID, escalabilidade e controle sobre os

dados, além de permitir a atualização, exclusão e inserção de dados de maneira eficiente.
Conexões: Delta Lake recebe os dados processados pelo Apache Spark e serve como fonte central para análise e treinamento de modelos de dados. Ele também permite a consolidação de dados provenientes de múltiplas fontes, garantindo qualidade e consistência.

5. Acesso aos Dados: Databricks
Justificativa: O Databricks é uma plataforma integrada que fornece acesso fácil e otimizado aos dados no Delta Lake. Ele utiliza o Apache Spark para processamento e oferece uma interface interativa para análise, desenvolvimento de notebooks e execução de consultas SQL. Além disso, o Databricks oferece automação e escalabilidade, permitindo que o time de dados se concentre na criação de valor a partir dos dados, em vez de se preocupar com a infraestrutura.
Conexões: Databricks acessa os dados armazenados no Delta Lake e executa consultas analíticas, oferecendo uma plataforma fácil de usar para cientistas de dados e analistas de dados.

6. Visualização de Dados: Tableau
Justificativa: O Tableau é uma das ferramentas de visualização de dados mais populares e poderosas, capaz de criar dashboards interativos e dinâmicos que ajudam na exploração dos dados e na comunicação de insights de forma clara. A facilidade de uso e a integração com diversas fontes de dados (incluindo Delta Lake e MySQL) fazem do Tableau uma excelente escolha para startups que buscam agilidade e eficácia na visualização e tomada de decisões baseadas em dados.
Conexões: O Tableau pode se conectar diretamente ao Databricks e ao Delta Lake para acesso rápido a dados analíticos, além de poder integrar-se ao MySQL para visualização de dados transacionais, criando relatórios dinâmicos e interativos.

7. Operacionalização de Dados: Apache Airflow
Justificativa: O Apache Airflow é uma plataforma de orquestração de workflows, ideal para agendar e gerenciar pipelines de dados complexos, garantindo a execução ordenada e monitoramento de tarefas. Ele pode ser usado para automatizar o fluxo de dados, como ingestão, processamento e carregamento, além de lidar com dependências entre os diversos estágios da arquitetura de dados.
Conexões: O Airflow pode orquestrar processos como o Spark Jobs e a ingestão de dados via NiFi, além de garantir que as tarefas de processamento sejam executadas conforme o planejado, de maneira escalável e monitorada.

Cada ferramenta foi escolhida com base nas **necessidades específicas** da QuantumFinance, visando **escalabilidade, eficiência e agilidade** no tratamento e visualização de dados financeiros.

- O **Apache NiFi** garante uma ingestão robusta e escalável.
- O **Apache Spark** oferece processamento em larga escala, tanto para dados batch quanto streaming.
- O **MySQL** atende as necessidades de armazenamento transacional, enquanto o **Delta Lake** consolida dados analíticos de forma segura e eficiente.
- **Databricks** oferece acesso facilitado e otimizado a esses dados para análises, enquanto o **Tableau** torna a visualização e exploração de dados acessível.
- Finalmente, o **Apache Airflow** orquestra todos esses processos, garantindo a execução e monitoramento eficientes da pipeline de dados.

Parte 2 - MVP Processo de ingestão de registros de transações no MySQL

O exemplo de MVP de fluxo escolhido foi o de registro de transações no MySQL. Iremos montar esse utilizando o Apache NiFi para enviar dados de transações financeiras para um banco de dados MySQL, com base no modelo relacional OLTP (Online Transaction Processing). A modelagem relacional OLTP é a escolha ideal para dados de transações financeiras, pois ela assegura a integridade, a atomicidade e a consistência necessárias para o processamento de transações financeiras. O MVP irá realizar a ingestão de csv na tabela de Transacoes.

Criação dos dockers

--Nifi

```
docker container exec -it nifi bash
mkdir /tmp/nifi
exit
```

--MySQL

```
docker run --name MySQL -it ivangancev/ubuntusql:latest bash
```

Configurações gerais dos dockers (após criação dos mesmos)

--Configuracao do nifi:

```
docker container cp c:\dts\transacoes.csv nifi:/tmp/nifi
docker container cp c:\dts\core-site.xml nifi:/opt/nifi/nifi-current/conf/
docker container cp c:\dts\hdfs-site.xml nifi:/opt/nifi/nifi-current/conf/
docker container cp c:\dts\postgresql-42.7.1.jar nifi:/opt/nifi/nifi-current/lib/ (essencial para a conexão do nifi com o MySQL)
```

--Configuração de redes

```
docker network create my_network
docker run -d --name nifi --network my_network
docker run -d --name MySQL --network my_network
```

Configuração do MySQL:

--Criação do banco do database

```
create database db_dts;
```

--Criação do usuário e liberação de acesso

```
CREATE USER 'user_nifi'@'nifi.my_network' IDENTIFIED BY '';
GRANT ALL PRIVILEGES ON quantumfinance.* TO '%@'nifi.my_network';
FLUSH PRIVILEGES;
```

-- Criação da tabela

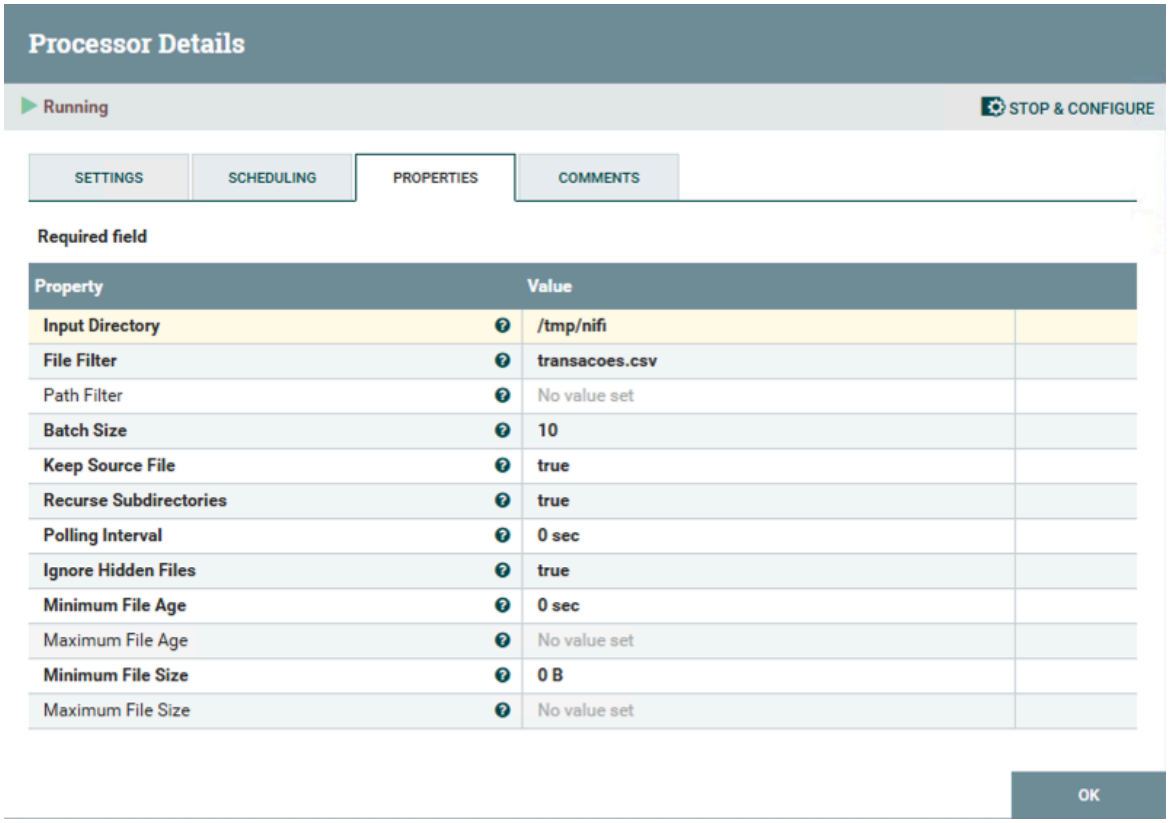
```
CREATE TABLE quantumfinance.Transacoes (
   transacao_id VARCHAR(9) PRIMARY KEY,
   conta_id_origem VARCHAR(6),
   conta_id_destino VARCHAR(6),
   valor FLOAT(15),
   tipo_transacao VARCHAR(3),
   data TIMESTAMP
);
```

Acessar o nifi pelo browser

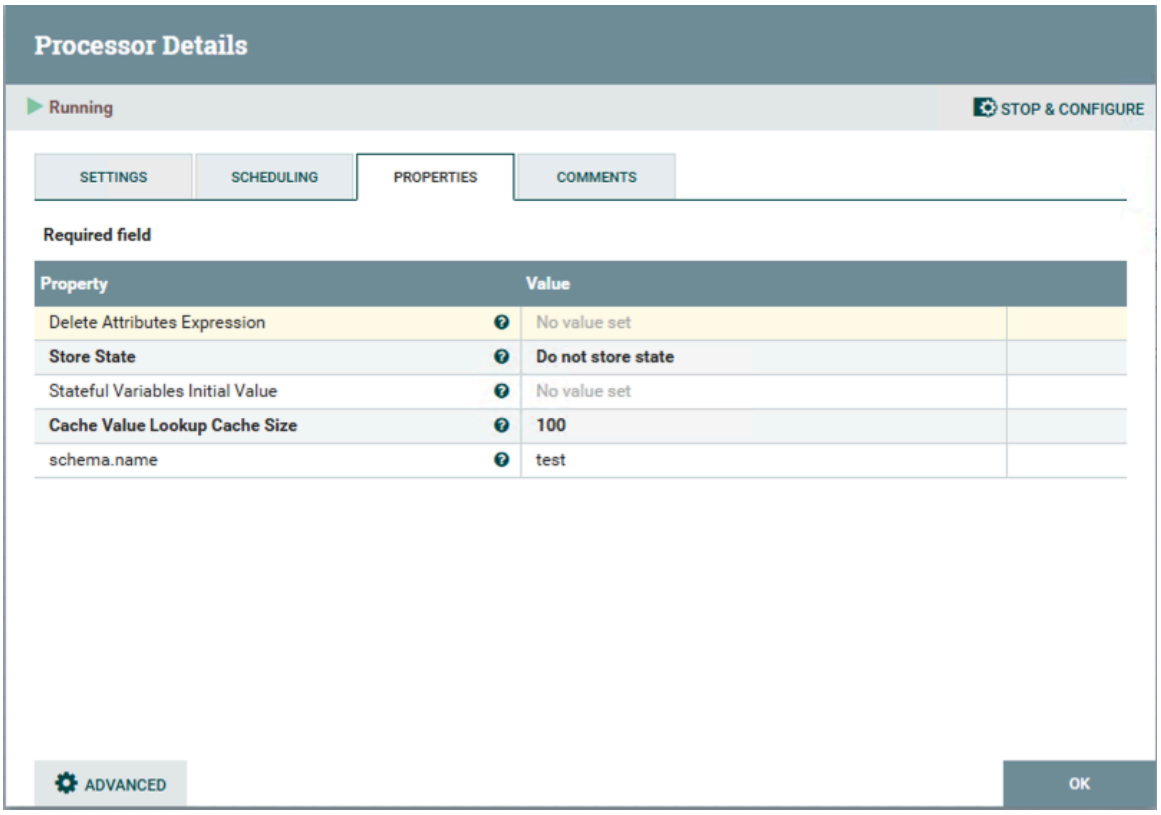
http://localhost:9090/nifi

Configurações do Nifi

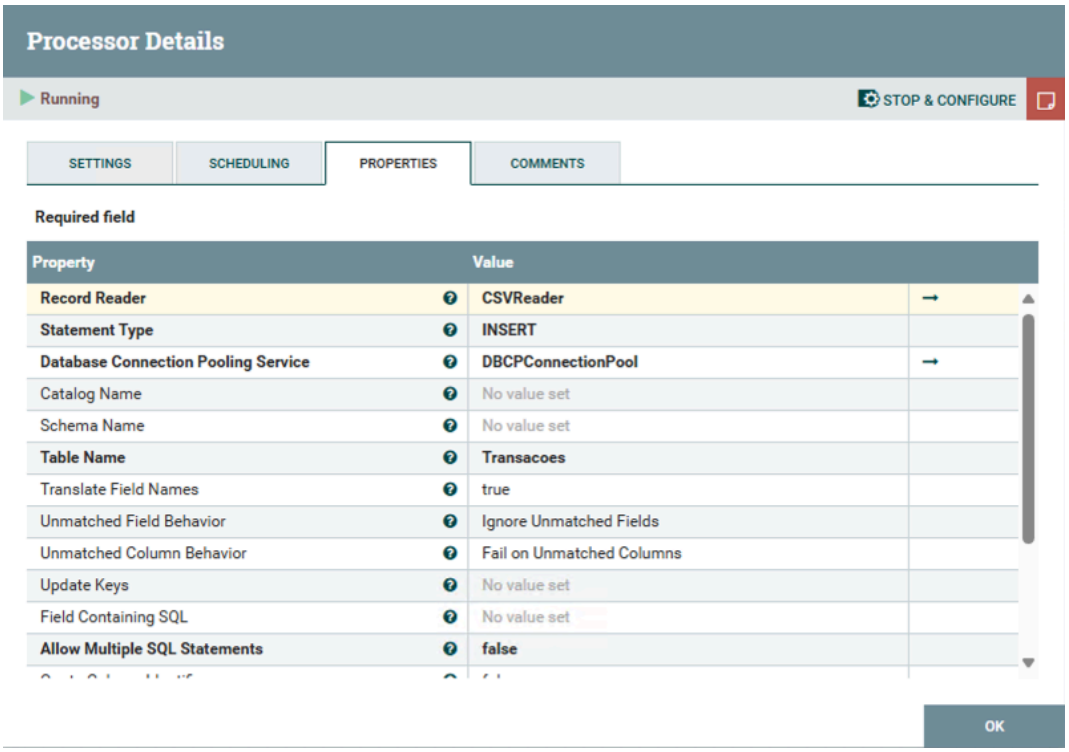
Dentro do NiFi, iniciamos o processo com a leitura do arquivo CSV utilizando o Processor GetFile, configurado da seguinte forma:



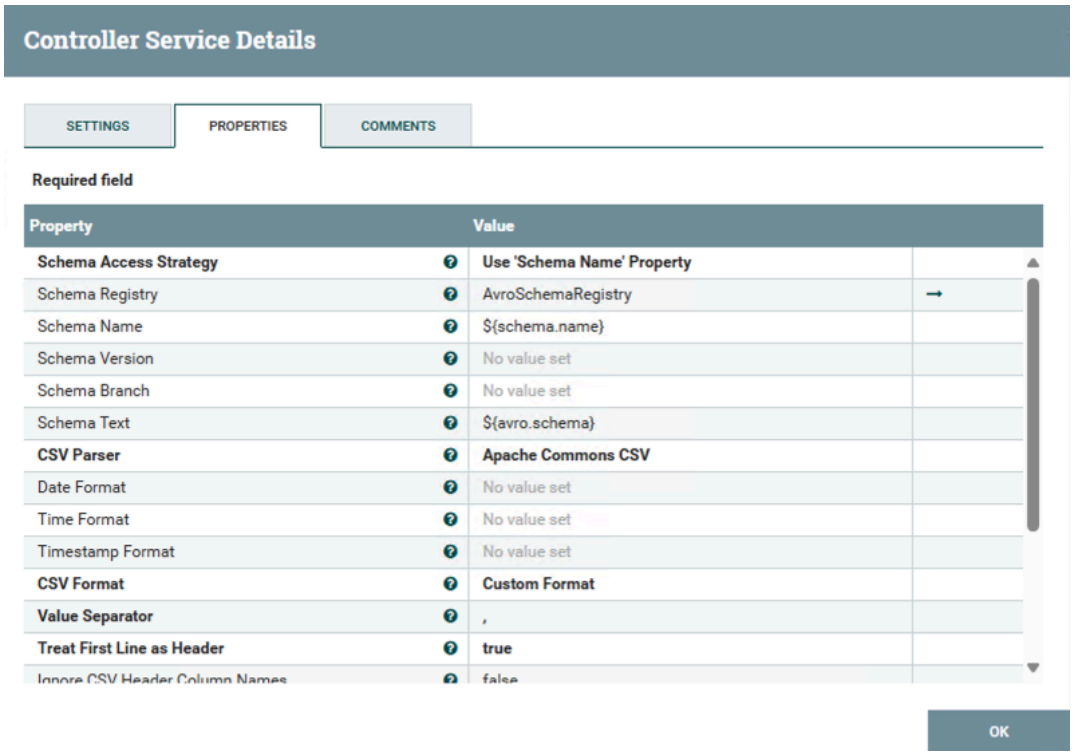
Como o CSV não contém um schema.name nas suas configurações padrão, uma etapa adicional foi inserida com o Processor Update Attribute, que denominamos "test". Nessa etapa, futuramente atribuíremos o schema ao arquivo.

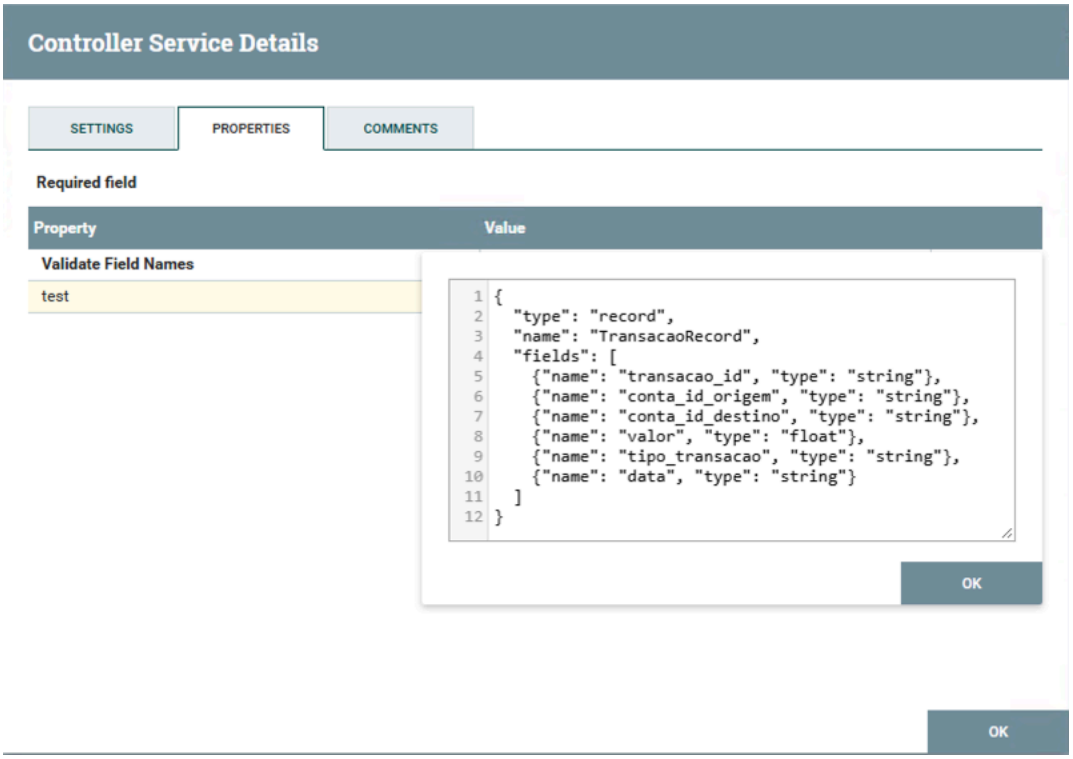


Na sequência, utilizamos o Processor PutDatabaseRecord, responsável pela transformação do CSV em um flowfile, que será enviado para o MySQL. Para essa configuração, são utilizados dois Controller Services: um para a leitura do arquivo CSV e outro para a conexão com o banco MySQL.

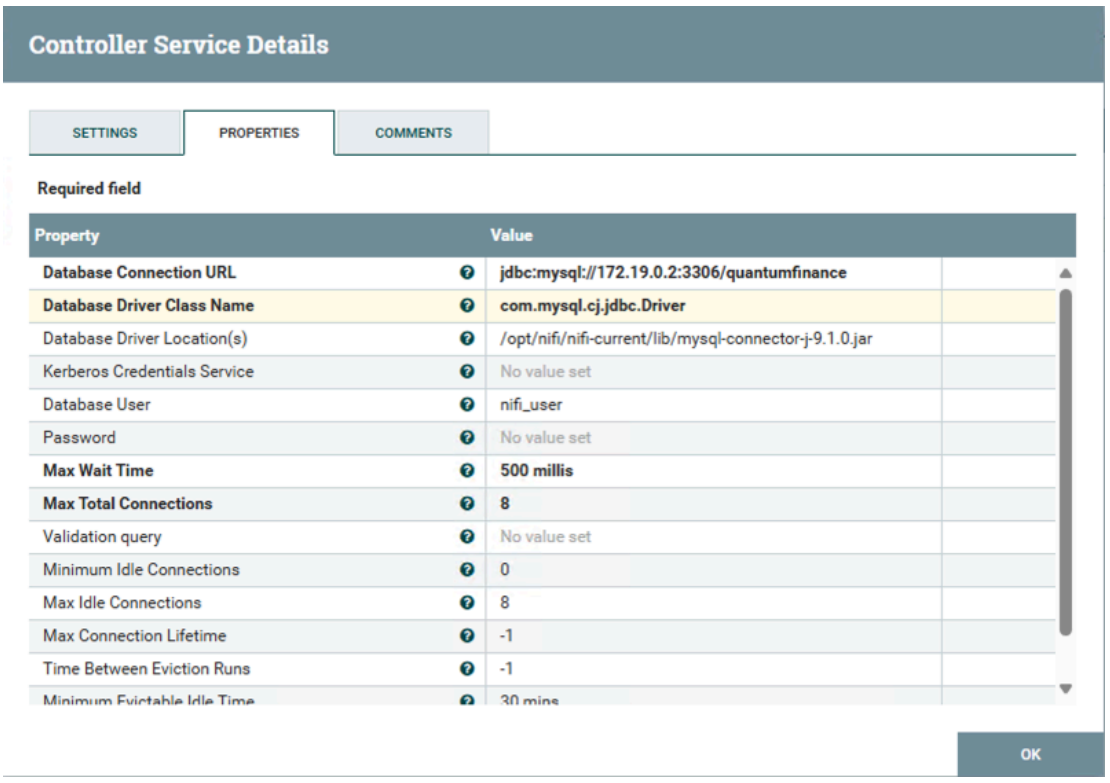


O CSVReader possui as configurações adequadas, sendo que um terceiro Controller Service é utilizado para configurar o schema e passá-lo para o atributo "test" que definimos anteriormente.



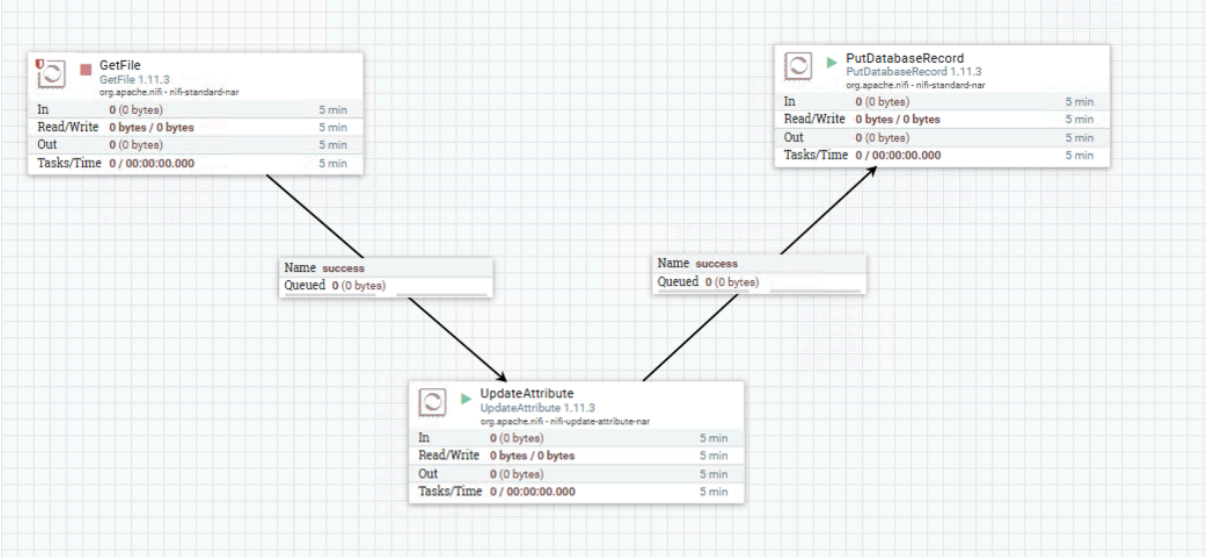


Já o DBCPConnectionPool é encarregado da conexão com o MySQL. Para configurá-lo, é necessário informar o Database Connection URL, que inclui o IP, a porta do Docker do MySQL e o database em uso. Além disso, é preciso referenciar o local do arquivo do Driver (o mesmo que foi carregado anteriormente via terminal).



Por fim, o fluxo MVP é composto pelos Processors GetFile > UpdateAttribute > PutDatabaseRecord, que formam a base para a ingestão e transformação dos

dados. Esse fluxo pode ser facilmente consultado e replicado por meio do template QuantumFinance_Transacoes.xml, garantindo uma visualização clara de todo o processo e facilitando a implementação de futuras integrações ou ajustes na arquitetura de dados.




```
mysql> select * from quantumfinance.Transacoes;
+--------------+-----------------+------------------+---------+----------------+---------------------+
| transacao_id | conta_id_origem | conta_id_destino | valor   | tipo_transacao | data                |
+--------------+-----------------+------------------+---------+----------------+---------------------+
| 101851338    | 219791          | 824691           | 4050.86 | DOC            | 2024-11-10 00:13:53 |
| 114992769    | 856943          | 324815           | 4606.74 | PIX            | 2024-11-01 09:38:16 |
| 117806137    | 862505          | 937918           |  5197.8 | PIX            | 2024-11-24 02:50:49 |
| 179194945    | 611759          | 382245           |  4747.2 | TED            | 2024-10-31 16:58:04 |
| 183776839    | 104675          | 825682           | 8120.67 | DOC            | 2024-11-11 22:38:36 |
| 275584748    | 516417          | 816190           | 4654.72 | PIX            | 2024-11-22 05:22:42 |
| 301884039    | 102785          | 828552           | 5032.88 | PIX            | 2024-11-14 13:06:00 |
| 308284747    | 954943          | 179633           | 5895.89 | DOC            | 2024-11-05 00:16:31 |
| 339807174    | 693987          | 646765           | 9703.18 | PIX            | 2024-11-02 00:59:21 |
| 345648656    | 470073          | 701737           | 9955.58 | PIX            | 2024-11-30 04:06:28 |
| 361140453    | 420631          | 159695           | 3175.72 | TED            | 2024-11-16 16:52:09 |
| 371345537    | 406620          | 858746           |   571.1 | TED            | 2024-11-08 19:54:23 |
| 373753592    | 839565          | 742432           | 9237.83 | TED            | 2024-11-11 03:23:31 |
| 387087861    | 447350          | 785158           | 6073.62 | PIX            | 2024-11-21 02:13:11 |
| 425492296    | 273028          | 876143           | 9284.86 | TED            | 2024-11-25 18:17:49 |
| 444041889    | 696270          | 379408           | 5549.95 | PIX            | 2024-11-15 12:09:35 |
| 453428389    | 825304          | 771863           |    5799 | PIX            | 2024-11-30 13:33:03 |
| 454728268    | 172363          | 601114           | 4392.77 | DOC            | 2024-11-28 00:42:21 |
| 470107732    | 759337          | 505268           | 3363.32 | TED            | 2024-11-13 10:24:44 |
| 521434454    | 386314          | 234390           | 8432.27 | DOC            | 2024-11-13 17:26:57 |
| 521525625    | 273535          | 823920           | 9174.47 | DOC            | 2024-11-19 07:49:17 |
| 539007895    | 609527          | 294981           |  785.48 | PIX            | 2024-11-02 04:54:50 |
| 549455354    | 103895          | 171872           | 4186.32 | TED            | 2024-11-10 01:47:07 |
| 573509053    | 702965          | 431379           | 6740.85 | PIX            | 2024-11-29 09:42:24 |
| 604263386    | 528332          | 606341           | 6095.21 | DOC            | 2024-11-21 13:14:33 |
| 628310600    | 637044          | 833948           | 9585.44 | TED            | 2024-11-06 02:24:47 |
| 642344737    | 603060          | 967038           | 9262.51 | TED            | 2024-11-07 08:24:05 |
| 673874974    | 148025          | 784798           | 5784.69 | PIX            | 2024-11-09 08:36:05 |
| 687684683    | 129213          | 590252           | 6938.04 | PIX            | 2024-11-01 10:17:49 |
| 689869129    | 141071          | 896209           |   996.4 | PIX            | 2024-11-07 06:16:17 |
| 705820949    | 163070          | 763262           | 6801.33 | DOC            | 2024-11-21 01:57:03 |
| 712252874    | 998694          | 703023           | 1885.32 | PIX            | 2024-11-20 17:18:54 |
| 712673554    | 637595          | 209146           | 6796.78 | PIX            | 2024-11-19 08:42:27 |
| 726947679    | 611514          | 458439           |  448.03 | DOC            | 2024-11-16 15:10:23 |
| 729292522    | 192080          | 115341           | 8750.87 | DOC            | 2024-11-18 00:44:12 |
| 762110979    | 216297          | 687078           | 5049.98 | PIX            | 2024-11-26 05:54:47 |
| 774935765    | 729212          | 622949           |    4302 | PIX            | 2024-11-25 00:59:13 |
| 775817716    | 417822          | 263569           | 1463.49 | PIX            | 2024-11-05 13:20:06 |
| 809547195    | 637689          | 774948           | 1447.74 | PIX            | 2024-11-03 04:35:31 |
| 830319250    | 216934          | 938593           | 5714.53 | DOC            | 2024-11-29 03:19:36 |
```