

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
RIO GRANDE DO SUL  
CAMPUS CANOAS  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

WELLINGTON DESSUY DE ALMEIDA

**Mysql Elastic - Software para uso de dados do MySQL com  
o Elasticsearch**

Canoas, 22 de novembro de 2025.

WELLINGTON DESSUY DE ALMEIDA

**Mysql Elastic Migrator - Um software para migração  
de dados do MySQL para o Elasticsearch**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Canoas.

Prof(a). Dr(a). Carla Odete Balestro Silva  
Orientador(a)

Canoas, 22 de novembro de 2025.



**Ministério da Educação**  
**Secretaria de Educação Profissional, Científica e Tecnológica**  
**Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul**  
**Campus Canoas**

**ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO**

Aos 02 dias do mês de dezembro de 2025, às 13h45 horas, em sessão pública no Auditório 2 do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Canoas, na presença da Banca Examinadora presidida pelo(a) Professor(a):

Prof. Dra. Carla Odete Balestro Silva, e

composta pelos examinadores:

1. Prof. Dr. Dieison Soares Silveira

2. Prof. Dra. Patricia Nogueira Hubler

3. \_\_\_\_\_

o(a) aluno(a) Wellington Dessuy de Almeida apresentou o Trabalho de Conclusão de Curso intitulado: Mysql Elastic Migrator - Software para migração de dados do MySQL para o Elasticsearch como requisito curricular indispensável para a integralização do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Após reunião em sessão reservada, a Banca Examinadora deliberou e decidiu pela APROVAÇÃO do referido trabalho, divulgando o resultado formalmente ao aluno e demais presentes e eu, na qualidade de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pelo aluno.

\_\_\_\_\_  
Presidente da Banca Examinadora

\_\_\_\_\_  
Examinador 01

\_\_\_\_\_  
Examinador 02

\_\_\_\_\_  
Examinador 03

\_\_\_\_\_  
Aluno



Documento assinado digitalmente  
**WELLINGTON DESSUY DE ALMEIDA**  
Data: 15/12/2025 12:04:10-0300  
Verifique em <https://validar.iti.gov.br>

## RESUMO

Esse trabalho consiste na elaboração de um software que tem como objetivo realizar a migração de dados de um banco MySQL para um ElasticSearch. Esse objetivo origina-se no contexto problemático que há quando um sistema já existente, que faz uso do banco MySQL, começa a possuir um grande volume de dados em suas tabelas. Nessa situação, consultas de dados podem começar a ter problemas de performance, causando problemas para quem está utilizando esse sistema. Com a finalidade de auxiliar nesse contexto, sem a necessidade da alteração do banco já utilizado, o MySQL Elastic Migrator funciona como uma ferramenta auxiliar para permitir que, independente da estrutura utilizada no MySQL, sejam migrados dados existentes para o ElasticSearch, que possui alta performance em consultas de dados, assim estando disponível essa tecnologia para uso em conjunto no sistema já consumido. Para elaboração do software nessa pesquisa aplicada, foram aplicados conceitos da metodologia Design Sprint, além do uso de tecnologias como Docker, Docker Compose e Laravel.

**Palavras-Chave:** Migração de dados. MySQL. ElasticSearch. PHP. Laravel.

## ABSTRACT

This work consists of developing a software application whose objective is to perform data migration from a MySQL database to Elasticsearch. This objective arises from the problematic context that occurs when an existing system that uses a MySQL database begins to have a large volume of data in its tables. In this situation, data queries may start to experience performance issues, causing problems for users of the system. In order to assist in this context, without the need to modify the existing database, the MySQL Elastic Migrator functions as an auxiliary tool to allow data to be migrated from MySQL, regardless of its structure, to Elasticsearch, which offers high performance in data queries, thus making this technology available for use alongside the existing system. For the development of the software in this applied research, concepts from the Design Sprint methodology were applied, in addition to technologies such as Docker, Docker Compose, and Laravel.

**Keywords:** Data migration. MySQL. Elasticsearch. PHP. Laravel.

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1 – Design Sprint: Mapeamento dos Problemas.....                | 19 |
| Figura 2 – Design Sprint: Mapeamento do Usuário Foco.....              | 19 |
| Figura 3 – Design Sprint: Mapeamento dos requisitos.....               | 20 |
| Figura 4 – Design Sprint: Tomada de decisão na jornada do usuário..... | 22 |
| Figura 5 – Design Sprint: Prototipação.....                            | 23 |
| Figura 6 – Diagrama de caso de uso.....                                | 24 |
| Figura 7 – Diagrama de classes.....                                    | 25 |
| Figura 8 – Diagrama de sequência.....                                  | 26 |
| Figura 9 – MySQL Elastic Migrator: Configuração MySQL.....             | 29 |
| Figura 10 – MySQL Elastic Migrator: Cadastro de consulta.....          | 30 |
| Figura 11 – MySQL Elastic Migrator: Execução.....                      | 31 |
| Figura 12 – MySQL Elastic Migrator: Geração de dados.....              | 32 |
| Figura 13 – Verificação de dados agregados por cliente.....            | 33 |
| Figura 14 – Verificação de dados agregados por produto.....            | 34 |

## SUMÁRIO

|   |           |
|---|-----------|
| <b>1 INTRODUÇÃO</b>                                   | <b>8</b>  |
| 1.1 MOTIVAÇÃO   | 9         |
| 1.2 OBJETIVOS   | 12        |
| 1.2.1 Objetivo Geral                                  | 12        |
| 1.2.2 Objetivos Específicos                           | 12        |
| <b>2 FUNDAMENTAÇÃO TEÓRICA</b>                        | <b>14</b> |
| 2.1 GITHUB  | 14        |
| 2.2 ELASTICSEARCH                                     | 14        |
| 2.2.1 Elastic Cloud                                   | 15        |
| 2.3 MYSQL   | 15        |
| 2.4 COMPOSER  | 15        |
| 2.5 LARAVEL FRAMEWORK                                 | 15        |
| 2.5.1 Filament PHP                                    | 16        |
| 2.6 DOCKER  | 16        |
| 2.6.1 Docker Compose                                  | 17        |
| 2.6.2 Docker Hub                                      | 17        |
| 2.7 DESIGN SPRINT                                     | 17        |
| <b>3 METODOLOGIA</b>                                  | <b>18</b> |
| 3.1 ELICITAÇÃO DE REQUISITOS                          | 18        |
| 3.2 CONFIGURAÇÃO DE AMBIENTE                          | 26        |
| 3.2.1 Docker e Docker Compose                         | 26        |
| 3.2.2 Criação do projeto Laravel Framework            | 27        |
| 3.3 IMPLEMENTAÇÃO                                     | 27        |
| 3.3.1 Migração dos dados                              | 27        |
| 3.3.2 Interface Web Complementar                      | 29        |
| 3.4 VERIFICAÇÃO DOS DADOS MIGRADOS                    | 31        |
| 3.4.1 Validação da aplicação por profissional da área | 34        |
| <b>4 CONCLUSÃO</b>                                    | <b>36</b> |
| <b>REFERÊNCIAS</b>                                    | <b>37</b> |

## 1 INTRODUÇÃO

Com um mercado atualmente de aproximadamente 58 milhões de dólares, o uso de software para resolução de problemas empresariais está em franco crescimento. Estima-se que esse mesmo mercado dobre de tamanho até 2029, gerando um cenário de muitas oportunidades (MORDOR INTELLIGENCE, 2024). Com muitas empresas investindo na criação, utilização e manutenção de softwares, o retorno desses investimentos também é algo esperado. Por isso, é necessário não só que os sistemas funcionem, mas que atendam as expectativas de eficiência que motivaram os aportes.

Softwares utilizam-se de Sistemas Gerenciadores de Banco de Dados (SGBD) para manipulação de suas bases de dados (ELMASRI e NAVATHE, 2011). No entanto, temos diversas opções, mas segundo o ranking da DB-Engines (2024), os bancos relacionais lideram quanto a utilização, estando Oracle e MySQL em primeiro e segundo lugar, respectivamente. No entanto, esses SGBDs mostraram-se não serem tão eficientes quando precisam gerenciar um grande volume de dados (OLIVEIRA, 2014).

Em contrapartida, bancos de dados não relacionais, a exemplo o ElasticSearch, que é o SGBD mais utilizado considerando a performance em consulta de dados (DB-ENGINES, 2024), aparecem como alternativa para manter a eficiência frente a um maior volume de informação. Então, considerando a grande utilização dos bancos relacionais, em especial, o MySQL, em softwares já existentes, e a barreira enfrentada quando há um aumento no volume de dados por ele manipulado, a migração desses dados para um banco de dados ElasticSearch pode ser a alternativa para a longevidade de um software com a resposta adequada quanto a performance.

Com isso, o objetivo dessa pesquisa foi desenvolver um software capaz de realizar a migração de dados de um banco de dados MySQL para um banco de dados ElasticSearch. Dessa forma, facilitando o processo de utilização do banco ElasticSearch junto a um sistema que já faça o uso do MySQL como principal SGBD. A pesquisa aplicada e de natureza qualitativa, teve o intuito de criar esse sistema migratório de dados através da utilização dos conceitos de desenvolvimento de software.

Para desenvolvimento do sistema, foi usada a linguagem PHP com a aplicação do framework Laravel<sup>1</sup> como base para esse software. Para execução do software, juntamente ao

---

<sup>1</sup> Laravel é um framework de aplicação web que tem como propósito facilitar o desenvolvimento através da disponibilização de vários recursos básicos, deixando o desenvolvimento para as questões mais específicas do sistema proposto. Para mais informações do framework acesse: <https://laravel.com/>

processo de criação da aplicação em si, também foi codificado um Dockerfile, bem como um arquivo de configuração para o Docker Compose, permitindo assim a execução desta aplicação através de um conjunto de containers Docker<sup>2</sup>. Esse software foi verificado, quanto ao procedimento de migração de dados ser bem sucedido, através de testes em um ambiente controlado com uma base de dados de grande volume, gerada especificamente para essa validação.

## 1.1 MOTIVAÇÃO

Atualmente, diversas tecnologias fazem parte da vida cotidiana da maior parte das pessoas, inclusive quando focado no contexto empresarial. O mercado de desenvolvimento de software, especificamente na área de produtividade empresarial, conforme pesquisa da Mordor Intelligence (2024), movimenta aproximadamente 58 bilhões de dólares anualmente e estima-se que dobre esse valor até 2029. Considerando esse crescimento estimado, muitos softwares serão criados, alterados e mantidos nesse contexto de inserção tecnológica que já é uma realidade necessária para qualquer empresa.

Com o aumento do uso de softwares, também ocorre um fenômeno descrito por Menezes (2016, p. 1) com relação a quantidade de dados armazenados:

Nas últimas décadas, inúmeras bases de dados estão tendenciando a possuir grande volume, alta velocidade de crescimento e grande variedade. Esse fenômeno é conhecido como Big Data e corresponde a novos desafios para tecnologias clássicas como Sistema de Gestão de Banco de Dados Relacional pois não tem oferecido desempenho satisfatório e escalabilidade para aplicações do tipo Big Data. (MENEZES, 2016, p. 1)

Evidenciando como a quantidade de dados tem sido cada vez maior, “o portal Finances Online [...] previu o aumento no consumo de dados, passando de 74 zettabytes, em 2021, para 149 zettabytes, em 2024” (O GLOBO, 2023).

Uma camada importante em um software é a de armazenamento de dados. É nela que temos a utilização de bancos de dados ou Sistemas Gerenciadores de Banco de Dados que são programas que permitem aos usuários criarem e manterem um banco de dados (ELMASRI e NAVATHE, 2011). Segundo Lóscio *et al* (2011), algumas das principais características de um

---

<sup>2</sup> O conceito de container Docker é explicado no capítulo 2.6.

SGBD são a segurança, a recuperação de falhas, gerenciamento do mecanismo de armazenamento de dados e controle das restrições de integridade do banco de dados.

Os SGBD podem ser classificados de acordo com alguns critérios, sendo um dos principais o modelo de dados (ELMASRI e NAVATHE, 2011). Dentre esses modelos temos as classificações de relacionais e não relacionais. O modelo relacional “[...] usa o conceito de relação matemática - que se parece com uma tabela de valores” (ELMASRI e NAVATHE, 2011). Dentre os diferentes SGBD que existem atualmente, quando falamos de bancos relacionais, MySQL é o *Open Source*<sup>3</sup> SQL (*Structured Query Language*) mais popular do mundo. Esse SGBD tem sido amplamente utilizado no mundo para desenvolvimento de softwares, pois, conforme ranqueamento da DB-Engines (2024), MySQL está na segunda posição, perdendo apenas para o Oracle que é um software que possui a mesma proposta mas que é proprietário<sup>4</sup>.

Mesmo sendo amplamente utilizado, o MySQL tem suas desvantagens em cenários com grandes volumes de dados pelo modelo relacional, que é a sua forma de estruturação dos dados armazenados. De acordo com Oliveira (2014, p. 187):

Com o passar do tempo e com o crescente volume de dados gerados a partir da expansão virtual, identificou-se que o modelo relacional não é tão escalável quanto necessário. Quando utilizado para gerenciar um grande volume de informações e cargas de trabalhos típicas de operações modernas de grande carga, incluindo o dimensionamento de conjuntos de dados, o banco de dados relacional perde sua performance consideravelmente. Neste sentido, quanto mais dados forem gerados, mais recursos de hardware serão necessários, como memórias e discos, para que a qualidade do serviço seja mantida. (OLIVEIRA, 2014, p. 187)

Nesses cenários, para os quais busca-se manipular massas de dados sem afetar a disponibilidade e a velocidade dos sistemas, é que alternativas tecnológicas para o armazenamento de dados são selecionadas. Nesse contexto, temos os bancos não relacionais, também chamados de NoSQL, que podemos caracterizar como “[...] um conjunto de conceitos que permitem o processamento de dados de forma rápida e eficiente com um foco em performance” (ANICETO e XAVIER, 2014, p. 17). Dados de uma pesquisa comparativa de performance entre um banco relacional e um não relacional realizada por FERREIRA e SILVA

---

<sup>3</sup> *Open Source* é um termo que caracteriza um software como código aberto, ou seja, o código desse software está disponível para qualquer pessoa.

<sup>4</sup> Um software proprietário é um sistema que é propriedade de uma empresa ou pessoa. O usuário do software não possui acesso ao código fonte e normalmente precisa de alguma licença do proprietário para utilização.

(2017) apontaram que, a consulta de dados, considerando aproximadamente 4 milhões de registros, levou em torno de 2,8 vezes mais tempo para retorno dos dados no banco relacional frente ao tempo em que o banco não relacional realizou a mesma consulta.

Observando-se os bancos não relacionais, ElasticSearch que é um “mecanismo de pesquisa e análise distribuído, armazenamento de dados escalável e um banco de dados vetorial” (ELASTICSEARCH..., 2024), vem como uma alternativa ao uso de bancos relacionais quando o propósito é manter os sistemas eficientes frente ao aumento significativo de dados armazenados. Quando a busca de informações no banco de dados é o objetivo principal na escolha do SGDB, o ElasticSearch tem sido muito utilizado, constando no ranking de SGDB focado em mecanismos de pesquisa da DB-Engines (2024) como primeiro colocado.

Como visto na citação de Oliveira (2014), o problema relacionado a performance de um sistema que usa um SGBD relacional ocorre com a utilização de um sistema cujos dados, em termos de quantidade, vão crescendo. O próprio pesquisador, desempenhou funções de desenvolvedor em sistema que se enquadram nesse cenário, sendo isso um impulsionador para a realização da pesquisa. É possível perceber que para manter um software com a utilização do banco relacional, mas possuir uma maneira de incrementar a performance na consulta dos dados é de extrema importância. A utilização conjunta de um banco NoSQL como o ElasticSearch pode ser a alternativa para a longevidade de um software apesar do aumento no volume de dados.

Nesse contexto, faz-se a necessidade de mecanismos com o propósito de migrar dados de uma base de dados relacional para uma não relacional. Assim, mesmo grandes sistemas, já estabelecidos e desenvolvidos utilizando um banco relacional, também conseguirão fazer uso de uma melhor tecnologia para solução de problemas na velocidade de acesso às suas informações. Ferramentas como o Estuary Flow (ESTUARY, 2024), surgem para facilitar a migração de dados, por exemplo, do MySQL para o ElasticSearch. A proposta desse modelo de ferramenta é justamente facilitar o caminho da equipe de desenvolvimento responsável por um software quanto à utilização do banco não relacional, nesse caso o ElasticSearch, em um sistema que já está em uso. Porém, esse serviço já existente, que é pago, pode não ser alternativa viável do ponto de vista econômico.

Essa pesquisa pretende contribuir para esse contexto através de um software que atua na migração dos dados de um banco MySQL para um ElasticSearch de código aberto e de livre

utilização. Esse sistema permite às equipes de desenvolvimento definir quais dados devem ser migrados bem como as informações para conexões nos dois SGBDs.

Para utilização desse software será necessária a criação de containers orquestrados via Docker Compose. Junto à aplicação encontra-se o arquivo de configuração pronto para inicialização de todos os containers. Através desses containers já inicializados, é possível acessar uma página web tanto para as configurações necessárias como para iniciar o processo de migração. Essa página web contém formulários para preenchimento das informações de conexão com os bancos de dados. Além disso, há um espaço para que o usuário defina as consultas que serão utilizadas para a busca dos dados a serem migrados, sendo então todos os campos retornados por elas salvos no ElasticSearch. Para salvar as informações configuradas no software ou quaisquer outro dado necessário para seu funcionamento, é utilizado um banco MySQL inicializado somente para esse propósito.

A equipe de desenvolvimento que utilizar o software, precisará configurar os bancos de dados para que possam ser acessados por esse container (com os dados de acessos já configurados na página web da aplicação) para que o software consiga buscar e salvar os dados durante o processo migratório. Para desenvolvimento desse sistema de migração foi utilizada a linguagem PHP com o uso do framework Laravel, tendo em vista a experiência prévia do pesquisador com ambos.

## 1.2 OBJETIVOS

Abaixo, serão abordados os objetivos geral e específicos da pesquisa.

### 1.2.1 Objetivo Geral

Desenvolver uma aplicação containerizada com interface web complementar capaz de realizar a migração de dados de um banco de dados MySQL para um banco de dados ElasticSearch.

### 1.2.2 Objetivos Específicos

- Realizar análise de requisitos através da utilização de processos da Design Sprint;
- Criar um Dockerfile para construção de imagem que servirá de base para o container Docker que executará o sistema de migração;
- Criar projeto Laravel para o desenvolvimento da aplicação;

- Codificar a aplicação para realizar a migração de dados;
- Verificar o funcionamento da aplicação em um contexto com grande volume de dados para validação;
- Realizar entrevistas com profissionais da área para captação de resultados do software.

## 2 FUNDAMENTAÇÃO TEÓRICA

Essa pesquisa utilizou-se de diversos recursos para a composição do software objetivo. Com a finalidade de contribuir no entendimento da solução criada, serão discutidos tópicos que explicam o que é cada uma das ferramentas, focando nas partes que têm relação com a pesquisa desenvolvida.

As seguintes ferramentas foram utilizadas nas mais diversas etapas do desenvolvimento da pesquisa. Algumas contemplam a etapa de elicitação de requisitos do software, outras são serviços ou aplicações usadas em conjunto com o software desenvolvido, além de ferramentas para a etapa do próprio desenvolvimento ou codificação.

### 2.1 GITHUB

Parte fundamental durante qualquer desenvolvimento de software é o versionamento do código, para o qual é amplamente utilizado o Git, que é um sistema de controle de versão (GITHUB, 2025). GitHub, que faz uso do sistema Git, “é uma plataforma baseada em nuvem em que é possível armazenar, compartilhar e trabalhar com outras pessoas para escrever códigos” (GITHUB, 2025). Por essa finalidade de compartilhamento, essa plataforma, dentro do contexto da pesquisa, consegue resolver questões referentes à disponibilização. De forma geral, a plataforma é um repositório para gerenciamento de códigos fontes de softwares.

### 2.2 ELASTICSEARCH

Dentre os diversos sistemas para bancos de dados, o ElasticSearch se destaca como um “mecanismo de busca de análise distribuído e de código aberto desenvolvido para velocidade, escala e aplicações de IA” (ELASTIC, 2025). Com sua grande performance na busca de informações, esse banco de dados se destaca na velocidade para retorno em meio a grandes quantidades de informações.

Um registro dentro do ElasticSearch é denominado documento, que é um objeto JSON (*Javascript Object Notation*), guarda os valores inseridos com seus pares de chave e valor. Com esse formato de informações, o ElasticSearch permite a busca desde dados mais simples como um número inteiro até textos complexos ou agregações de valores (ELASTIC, 2025).

### 2.2.1 Elastic Cloud

Para utilização de um banco de dados Elasticsearch, de uma maneira que não necessite um conhecimento aprofundado sobre questões de infraestrutura, usar um serviço diretamente de plataforma é mais simples e prático. Elastic Cloud é a plataforma oficial da Elastic para utilização de serviços como o Elasticsearch (ELASTIC, 2025).

Através dessa plataforma, que inclusive disponibiliza um tempo para avaliação gratuita, é possível testar o funcionamento do Elasticsearch, inserindo dados, realizando consultas e fazendo análises dos recursos utilizados. É possível validar até mesmo o custo que haveria pelo uso da plataforma durante o período de avaliação gratuita, permitindo a quem utilizar fazer o balanço se a contratação é algo viável.

### 2.3 MYSQL

MySQL é um banco de dados relacional, que armazena as suas informações em tabelas de dados. Esse banco utiliza a linguagem SQL (Structured Query Language), que é o padrão de linguagem mais comum para acesso a bancos de dados (ORACLE, 2025). Podem ser configuradas regras para o armazenamento de dados neste banco, que incluem obrigatoriedade de preenchimento, relacionamento entre colunas de diferentes tabelas, índices para facilitar a busca de registros, entre outras.

### 2.4 COMPOSER

O gerenciamento de dependências dentro de um projeto PHP é necessário e ferramentas como Composer tem essa finalidade. Com uma configuração, que é definida especificamente para cada projeto, é possível instalar ou atualizar bibliotecas que são dependências. (COMPOSER, 2025)

### 2.5 LARAVEL FRAMEWORK

Dentro do contexto de aplicações ou softwares, há diversas ferramentas, processos ou metodologias que visam organizar e/ou facilitar o desenvolvimento. Os frameworks, que são um

conjunto organizados de bibliotecas, componentes, padrões ou estruturas reutilizáveis, contemplam parte substancial do bom aproveitamento do tempo durante o desenvolvimento. Suas funções ou fluxos pré definidos, trazem agilidade e consistência na criação dos recursos para os quais o framework em específico se propõe.

O Laravel Framework serve para a criação de aplicações PHP, permitindo ao desenvolvedor focar no problema que é o foco do seu software ao invés de dedicar tempo com a criação de uma estrutura base para sua codificação (LARAVEL, 2025). Com sua imensa variedade de ferramentas e sua característica progressiva, um desenvolvedor iniciante ou avançado consegue usufruir dos seus recursos.

### 2.5.1 Filament PHP

Filament PHP também é um framework. No entanto, ele tem como propósito facilitar a etapa de criação de interfaces de usuários voltadas para painéis de administração ou gestão interna. Ele permite ao desenvolvedor criar essas interfaces através da manipulação de objetos PHP, incrementando a produtividade pois já disponibiliza através do uso das suas classes telas em formatos pré definidos (FILAMENTPHP, 2025).

### 2.6 DOCKER

A plataforma Docker permite que softwares sejam desenvolvidos, executados e entregues utilizando-se da estrutura de containers. Esses containers habilitam a execução das aplicações encapsuladas de tal forma que fique independente da infraestrutura na qual o container está rodando. Container é o nome dado a esse ambiente que é, até certo ponto, isolado. Por essas características, é possível a execução de diversos containers em uma mesma máquina, sem a interferência deles entre si (DOCKER, 2025).

Para a execução de um container como um processo isolado, é necessária a criação de uma imagem, a qual serve de base para esse container. Essa imagem é como um pacote com todas as necessidades para execução do software, desde arquivos, bibliotecas, configurações, pacotes ou qualquer outra especificação (DOCKER, 2025).

### 2.6.1 Docker Compose

Algumas aplicações necessitam de diversas tecnologias diferentes de infraestrutura funcionando simultaneamente. Com o uso de Docker containers, toda essa infraestrutura pode ser criada através de múltiplas criações desses containers. Como uma ferramenta de auxílio nesse processo de gerenciamento dos diversos containers, Docker Compose tem como finalidade defini-los e executá-los. Através de um arquivo de configuração YAML, toda estrutura dos serviços necessários são caracterizados, permitindo o uso deles em conjunto de forma fácil. Cada serviço define a criação de um container (DOCKER, 2025).

### 2.6.2 Docker Hub

Muitas necessidades no contexto de desenvolvimento de software são comuns, principalmente quando olhamos especificamente para alguma tecnologia ou ferramenta. Então, dentro do uso de containers, é possível a reutilização de imagens que já foram criadas ou definidas. Para isso, a plataforma Docker Hub é um catálogo de imagens já especificadas para contextos específicos, na qual é possível buscar e utilizar essas imagens para a criação de containers.

## 2.7 DESIGN SPRINT

De acordo com Silva (2024), “Design Sprint é uma metodologia centrada no usuário, iterativa, prática e colaborativa”. Essa metodologia, criada pela Google Ventures, visa a criação e validação de hipóteses rapidamente, para evitar grandes custos de desenvolvimento e tempo.

O processo de uma sprint de design é composta por 5 etapas: Entendimento e definição do problema, Divergência e proposição de diferentes ideias, Decisão por uma ideia através de votação, prototipação de média/alta fidelidade da solução proposta e validação do produto com usuários potenciais dando feedback real sobre a experiência de uso (SILVA, 2024). Essa metodologia foi feita para sua aplicação em equipes, permitindo assim uma maior troca de informações.

### 3 METODOLOGIA

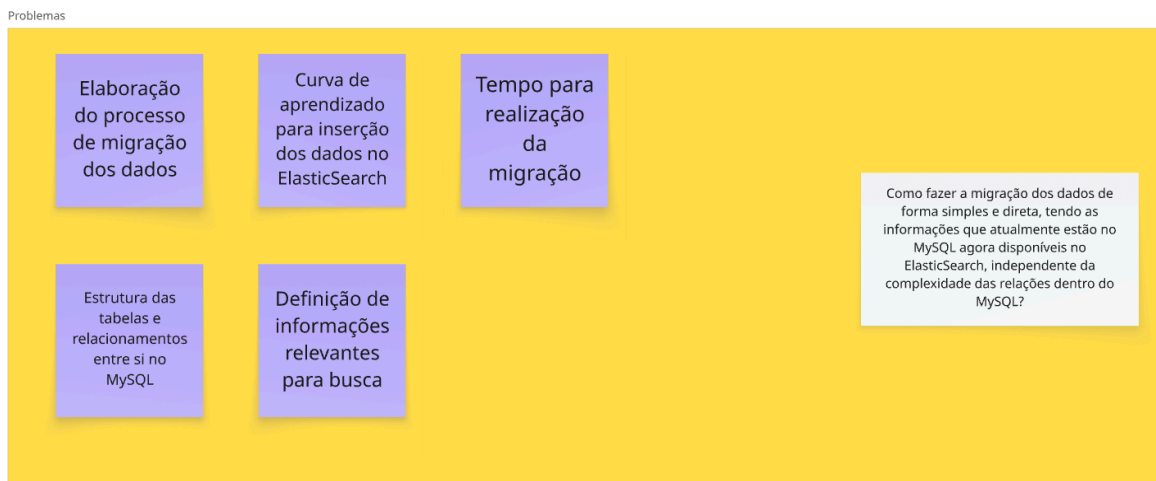
Essa pesquisa, que teve seu caráter qualitativo, desenvolveu-se em etapas. Pela sua classificação como pesquisa aplicada, que por sua vez “intenciona o desenvolvimento e inovação de produtos, em diversas áreas do conhecimento, aplicando conhecimento gerado pela pesquisa básica ou por ela mesma, com o intuito de resolver um problema” (PEREIRA, 2023), o software objetivo da pesquisa foi projetado, codificado e verificado. As etapas realizadas para cada uma das fases será detalhada nos tópicos a seguir.

#### 3.1 ELICITAÇÃO DE REQUISITOS

A primeira etapa constituiu-se no levantamento de requisitos do software. É necessário considerar a observação participante, tendo em vista que o autor deste trabalho atua como desenvolvedor em projeto que enfrenta a situação enquadrada no objetivo da pesquisa, além de pesquisa bibliográfica. Esses itens serviram de base para entendimento do cenário e aplicação de metodologia para definição das necessidades.

Assim, seguiu-se para a aplicação de partes da Design Sprint para obter os requisitos necessários da aplicação proposta. O primeiro fluxo utilizado foi o denominado Mapeamento. Como pode ser observado na Figura 1, o objetivo desse fluxo é trazer maior clareza a respeito do problema. Para isso, dores relacionadas ao contexto são listadas. Com essa lista, o primeiro objetivo da metodologia é a elaboração de uma pergunta que resume as dificuldades listadas, que foi a seguinte: Como fazer a migração dos dados de forma simples e direta, tendo as informações que atualmente estão no MySQL agora disponíveis no Elasticsearch, independente da complexidade das relações dentro do MySQL?

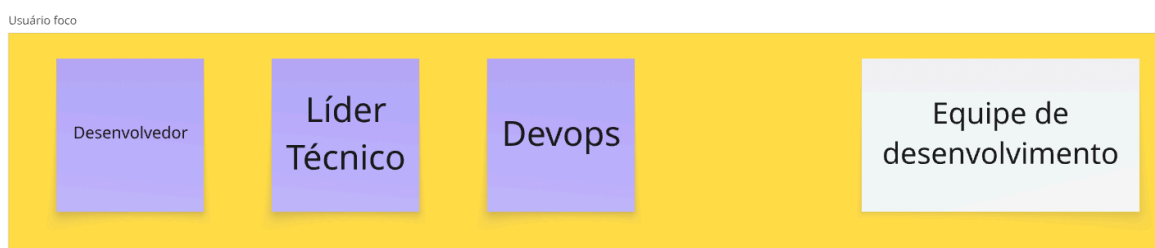
Figura 1 – Design Sprint: Mapeamento dos Problemas



Fonte: Autoria Própria, 2025.

A partir desse questionamento, seguindo o fluxo proposto metodologicamente, foram listados os possíveis usuários dessa aplicação, conforme a Figura 2. Com isso, viu-se que independente do papel específico do usuário dentro de um contexto de desenvolvimento de software, resume-se a equipe de desenvolvimento responsável pelo projeto que necessita da migração de dados.

Figura 2 – Design Sprint: Mapeamento do Usuário Foco



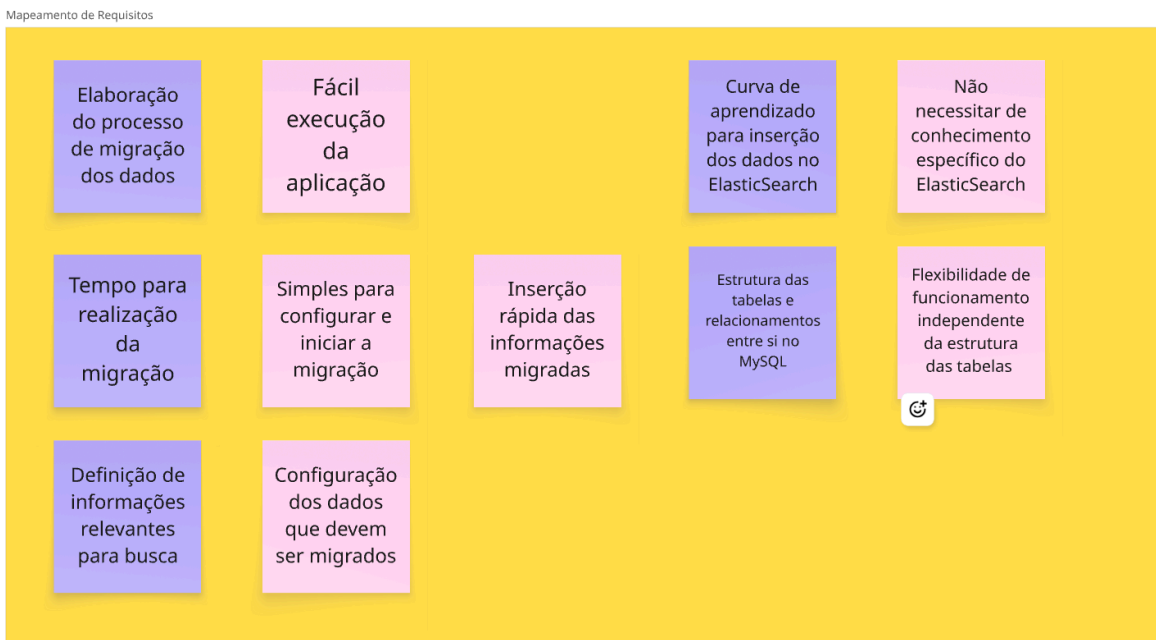
Fonte: Autoria Própria, 2025.

Com o usuário foco e os problemas listados, a próxima etapa foi apontar os possíveis requisitos para cada problema apresentado, levando em conta para qual tipo de usuário está sendo direcionado. Na Figura 3, vemos os requisitos levantados que foram os seguintes:

- Fácil execução da aplicação
- Simples para configurar e iniciar a migração

- Inserção rápida das informações migradas
- Configurações dos dados que devem ser migrados
- Não necessitar de conhecimento específico do ElasticSearch
- Flexibilidade de funcionamento independente da estrutura das tabelas

Figura 3 – Design Sprint: Mapeamento dos requisitos



Fonte: Autoria Própria, 2025.

Como parte final da etapa de mapeamento temos a criação da Jornada do Usuário. Nessa parte foi desenhado um fluxo de ações do usuário que tivessem como resultado final a resolução do problema apresentado. As ações também, em alguns casos, contam com inserções que são pontos relevantes dentro da sua execução. Os requisitos listados precisam de alguma maneira estarem vinculados à uma ação ou algumas dessas inserções complementares para garantir que a jornada desenhada cumpra o seu propósito na resolução do problema. Essa jornada contou com as seguintes ações e complementos:

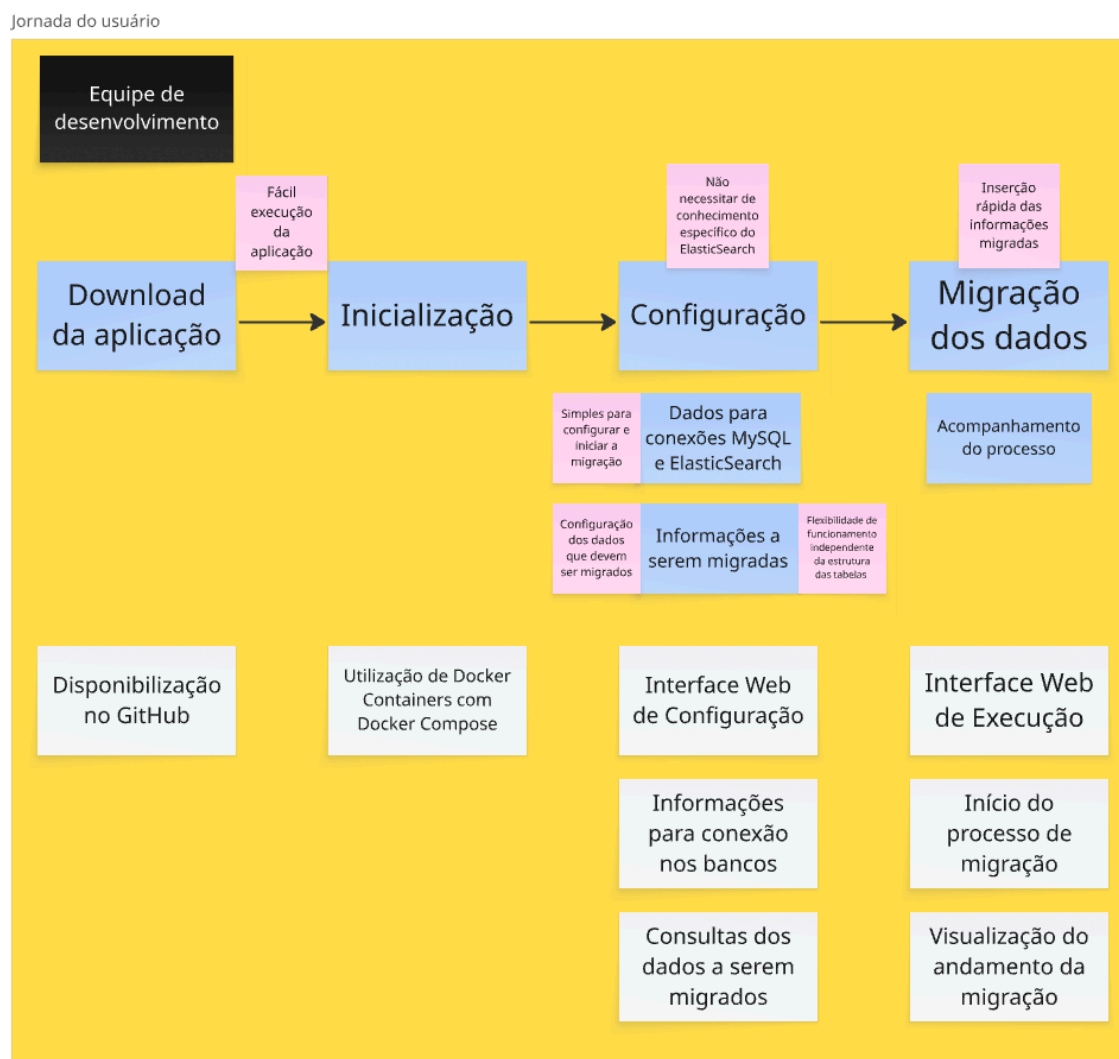
- Download da aplicação;
- Inicialização;
- Configuração:
  - Dados para conexões MySQL e ElasticSearch;

- Informações a serem migradas;
- Migração dos dados:
  - Acompanhamento do processo;

Com a etapa de Mapeamento finalizada, o próximo passo é a Tomada de Decisão. Como pode ser visto através da Figura 4, esse momento constituiu-se de uma análise frente a jornada do usuário resultante para definição de possíveis soluções. Para cada ação da jornada foram listadas possíveis soluções que conseguissem cumprir o requisito responsável por gerá-la. Segue a lista da ação com sua solução proposta:

- Download da aplicação: disponibilização no GitHub;
- Inicialização: utilização de Docker Containers com Docker Compose;
- Configuração: interface web de configuração, informações para conexão nos bancos e consultas dos dados a serem migrados;
- Migração dos dados: interface web de execução, início do processo de migração e visualização do andamento da migração.

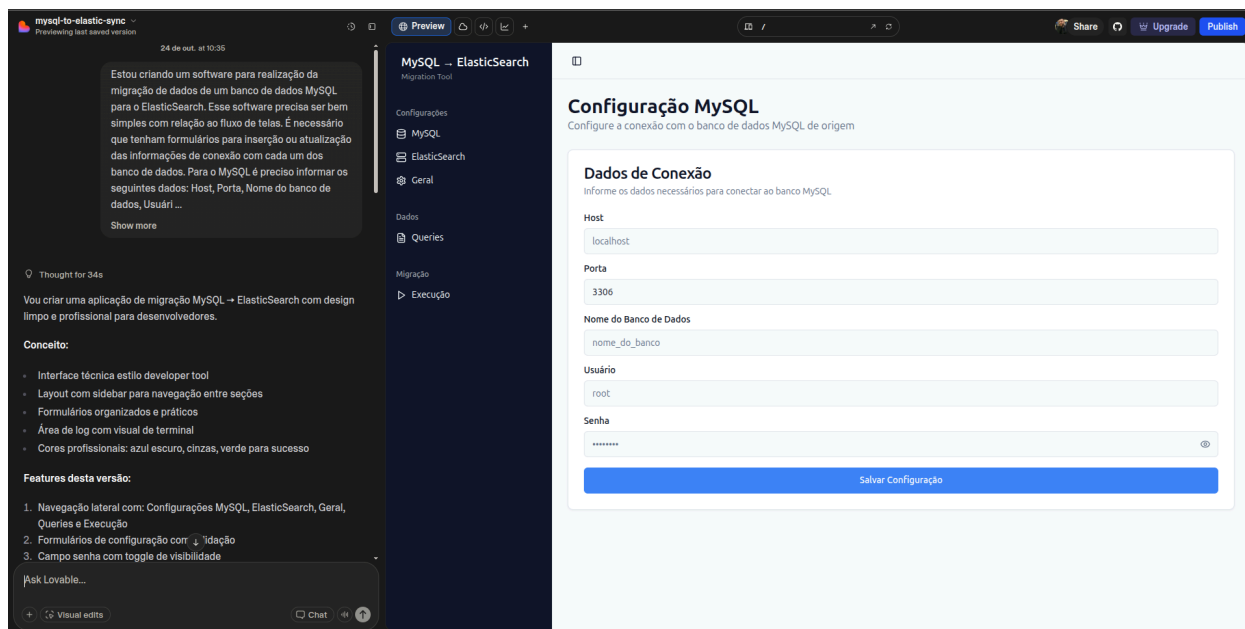
Figura 4 – Design Sprint: Tomada de decisão na jornada do usuário



Fonte: Autoria Própria, 2025.

Com as possíveis soluções, foi aplicada a etapa final dentro da Design Sprint: Prototipação. Para esse momento, com o objetivo de criar uma interface que suprisse os requisitos, jornada e possíveis soluções, foi utilizado a plataforma Lovable. Na Figura 5 é possível ver a plataforma citada no momento da criação. Essa plataforma gera aplicações através de inteligência artificial com base em textos em linguagem natural. Um prompt foi inserido para contextualizar e solicitar que fosse criada essa aplicação, gerando assim um protótipo navegável para validação do fluxo completo do usuário para realização da migração de dados.

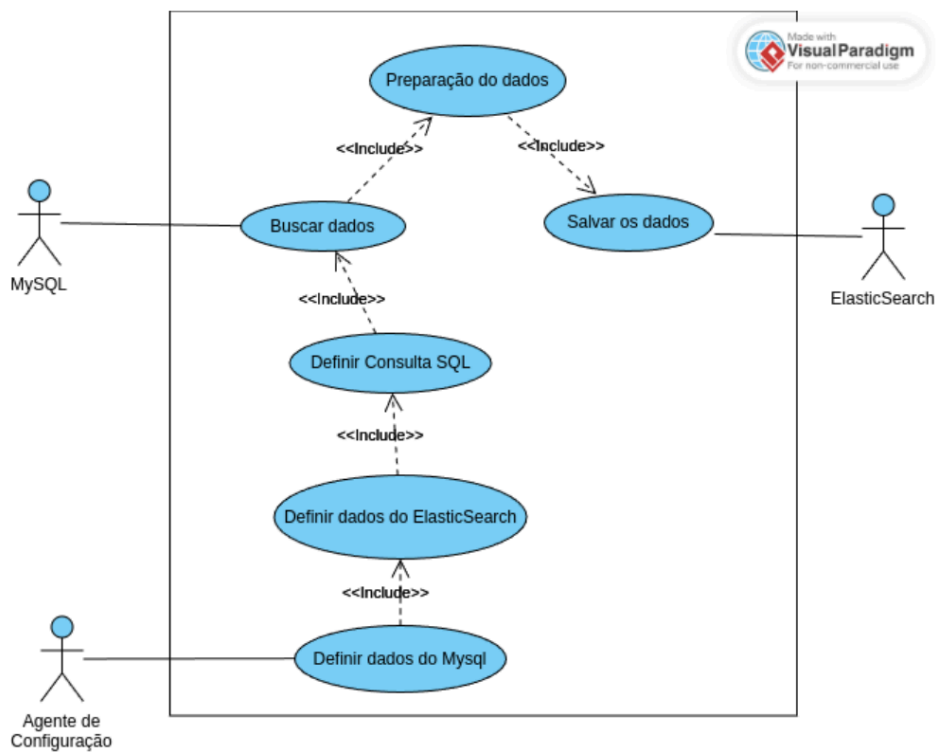
Figura 5 – Design Sprint: Prototipação



Fonte: Autoria Própria, 2025.

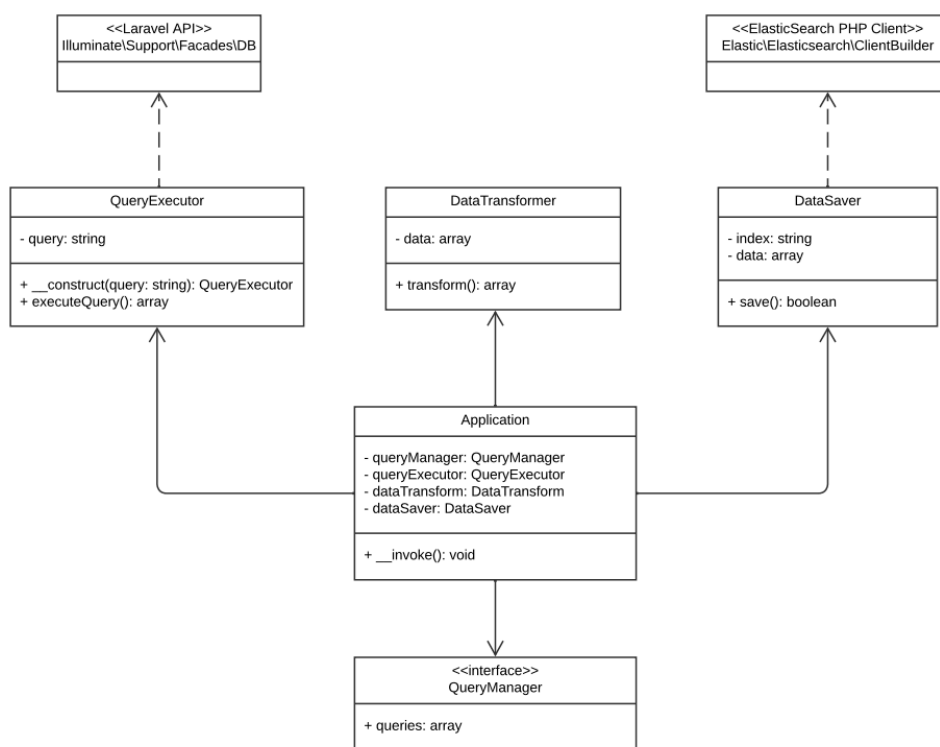
Para fins complementares a utilização da Design Sprint, também foram criados alguns diagramas com a utilização da UML (Linguagem de Modelagem Unificada). Como pode ser visto na Figura 6, o diagrama de caso de uso foi o primeiro criado. Após isso, o diagrama de classes conforme a Figura 7. E por fim, o diagrama de sequência visualizado através da Figura 8.

Figura 6 – Diagrama de caso de uso



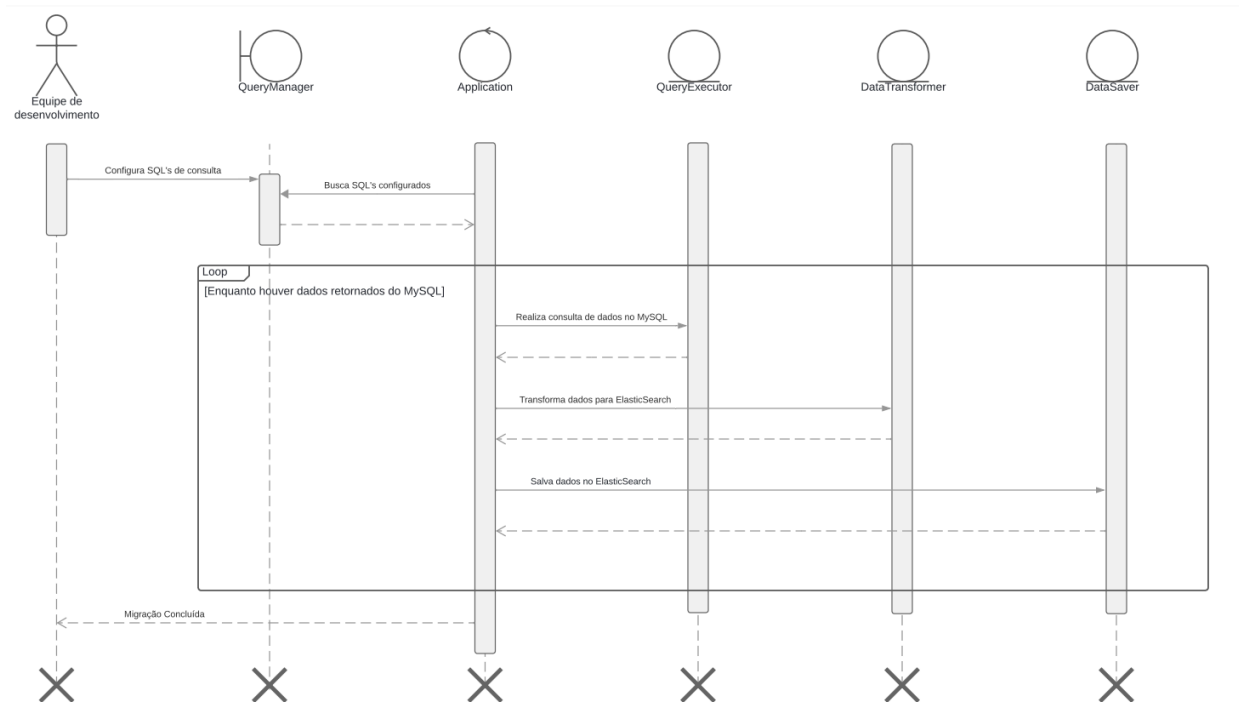
Fonte: Autoria Própria, 2025.

Figura 7 – Diagrama de classes



Fonte: Autoria Própria, 2025.

Figura 8 – Diagrama de sequência



Fonte: Autoria Própria, 2025.

Após todo material gerado pela aplicação da metodologia Design Sprint e os diagramas supracitados, seguiu-se para a etapa responsável pela configuração de ambiente.

### 3.2 CONFIGURAÇÃO DE AMBIENTE

Para possibilitar o download da aplicação por qualquer usuário, bem como fazer o versionamento do software, foi criado um repositório público no GitHub. Com esse repositório, a inicialização da aplicação foi a primeira etapa a ser resolvida. Para isso foram utilizados containers Docker orquestrados pela ferramenta Docker Compose.

#### 3.2.1 Docker e Docker Compose

Tendo em vista o apontamento do pesquisador para uso do PHP como linguagem de programação para desenvolvimento desta aplicação através da utilização do Laravel Framework, foi necessária a implementação de um container que suprisse os requisitos para um software com esses requisitos. Foi utilizada a plataforma Docker Hub para busca de imagens públicas que

pudessem servir de base para tanto esse container quanto os demais que posteriormente foram criados. Para o container responsável pela execução da aplicação foi desenvolvido um Dockerfile (arquivo base de construção de imagem para um container) que teve como imagem base a *php:8.3-fpm-bullseye*. Nesse Dockerfile foram adicionados passos para que cada necessidade para execução da aplicação fosse satisfeita. Assim, qualquer um, dentro de um sistema que possua o Docker, consegue executar a aplicação sem a necessidade de configurações específicas, cumprindo mais um requisito listado.

Com essa imagem, foi necessário a utilização do Docker Compose para o gerenciamento tanto desse container da aplicação quanto outro para inicialização do servidor web, responsável pela disponibilização da aplicação, bem como o container de banco de dados para guardar as configurações realizadas durante o uso da aplicação.

O arquivo de configuração do Docker Compose chamado *docker-compose.yml* foi codificado com um serviço para cada container necessário, assim permitindo que todos funcionem em conjunto para execução e disponibilização da aplicação.

### 3.2.2 Criação do projeto Laravel Framework

Para inicialização do projeto utilizando o Laravel Framework foi utilizado o próprio instalador disponível pela ferramenta através do Composer, que é um gerenciador de dependências para o PHP. Com um comando a estrutura do projeto foi criada permitindo o início da codificação do projeto. Apenas algumas configurações descritas na própria documentação do Laravel foram necessárias.

## 3.3 IMPLEMENTAÇÃO

Duas fases constituíram a codificação do software. A primeira delas referente à migração em si dos dados e a segunda responsável pela criação da interface web para configuração e utilização da aplicação.

### 3.3.1 Migração dos dados

Inicialmente a primeira necessidade para realização da migração foi justamente a conexão com os banco de dados que estariam envolvidos no processo. Para o MySQL, o Laravel

já disponibiliza uma maneira nativa de conexão com esse banco, somente sendo necessárias a configuração do container para que isso seja possível, o que foi realizado já na etapa de Configuração de Ambiente. No entanto, para conectar com o ElasticSearch não havia recurso nativo no framework. Então, foi necessária a inclusão de biblioteca para essa finalidade. Essa biblioteca, chamada ElasticSearch-PHP, é indicada na própria documentação do ElasticSearch e pode ser instalada via Composer.

Com o intuito de criar uma maneira de executar o código responsável pela migração dos dados, foi utilizado um recurso do Laravel chamado *Task Scheduling*. Esse recurso permite que determinada classe seja invocada sempre que um comando - *php artisan schedule:run* - seja executado.

A partir do diagrama de classes, foram implementadas as classes responsáveis pelo processo de migração dos dados. Esse processo possui algumas etapas, dentre as quais estão: verificação de conexão com os bancos de dados, busca das consultas de dados cadastradas e iteração pelas consultas.

No início do processo, a verificação de conexão com os bancos de dados é realizada com o propósito de garantir que o processo todo ocorra de forma. Caso a conexão com algum deles não seja estabelecida o usuário é informado pelo log da execução desse erro. Após isso, uma busca pelas consultas de dados cadastradas é realizada. Se não houver consultas cadastradas, o usuário também é informado, tendo em vista que são pré requisitos para busca dos dados que serão migrados.

Após isso, o processo de migração de forma direta, é iniciado. Ele constitui-se de uma iteração para cada consulta cadastrada. Em cada iteração, são processadas uma quantidade de registros que pode ser definida através de uma configuração. O índice definido para o recebimento dos dados no ElasticSearch é validado e, caso não exista, ele é criado. Então, os dados são buscados no MySQL, tratados para o formato esperado pela biblioteca ElasticSearch-PHP e enviados para o ElasticSearch. Essa iteração repete quantas vezes forem necessárias até que todos os registros retornados pela consulta sejam migrados. Esse processo pode ser acompanhado pelo log para acompanhamento.

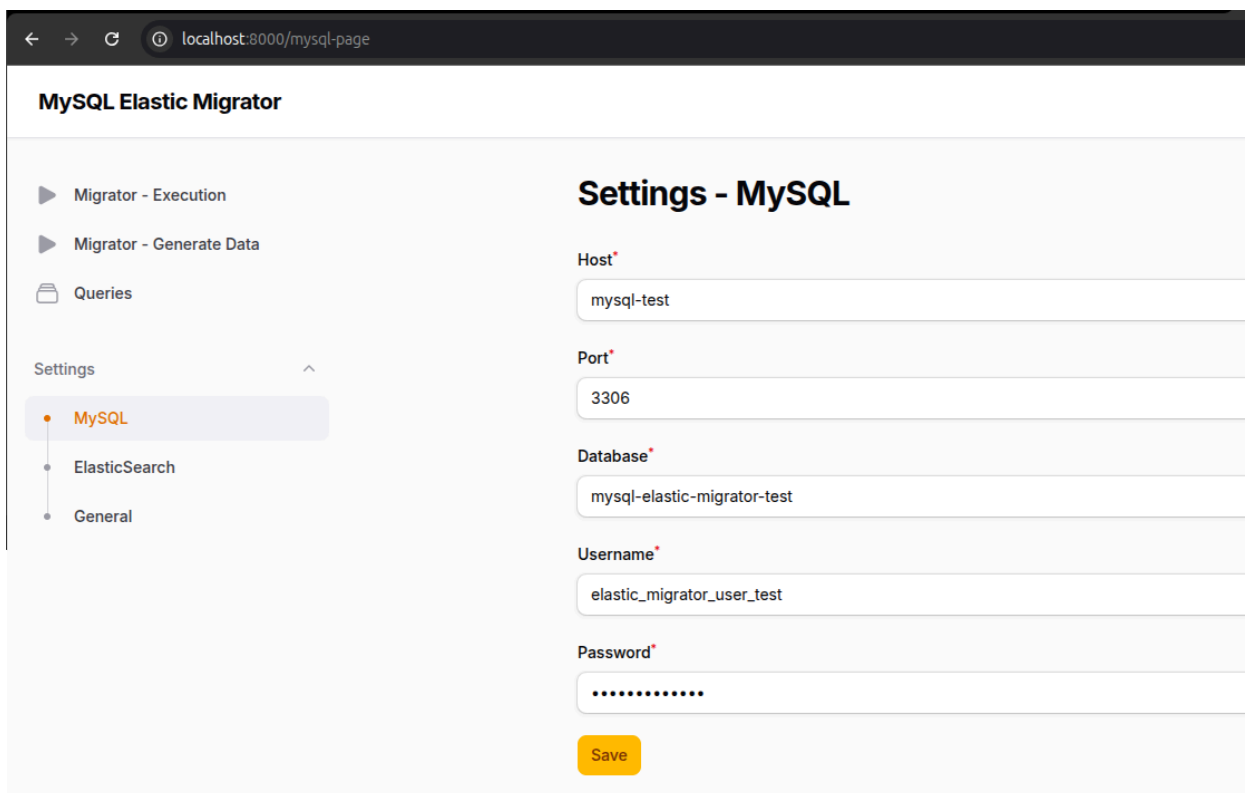
Com esse processo criado e verificado, o próximo passo foi a criação de uma interface web para seu gerenciamento. Facilitando para o usuário a configuração e a execução do processo de migração.

### 3.3.2 Interface Web Complementar

A partir do protótipo desenvolvido na etapa e elicitação de requisitos, foi iniciada a codificação da interface web com o auxílio do framework Filament PHP. Esse framework foi utilizado para a criação de uma interface que fosse robusta e atendesse a necessidade de configuração quanto à execução da aplicação de migração.

Para que o usuário tivesse fácil acesso às funcionalidades, foi aplicado o menu lateral para navegação. Com isso é possível acessar todas as telas de configurações, cadastro de consultas e execução da aplicação.

Figura 9 – MySQL Elastic Migrator: Configuração MySQL



The screenshot displays the MySQL Elastic Migrator web interface. The browser address bar shows 'localhost:8000/mysql-page'. The page title is 'MySQL Elastic Migrator'. On the left, a sidebar menu includes 'Migrator - Execution', 'Migrator - Generate Data', 'Queries', and 'Settings'. Under 'Settings', 'MySQL' is selected, with 'ElasticSearch' and 'General' also visible. The main content area is titled 'Settings - MySQL' and contains the following fields: 'Host' (mysql-test), 'Port' (3306), 'Database' (mysql-elastic-migrator-test), 'Username' (elastic\_migrator\_user\_test), and 'Password' (masked with dots). A yellow 'Save' button is located at the bottom of the form.

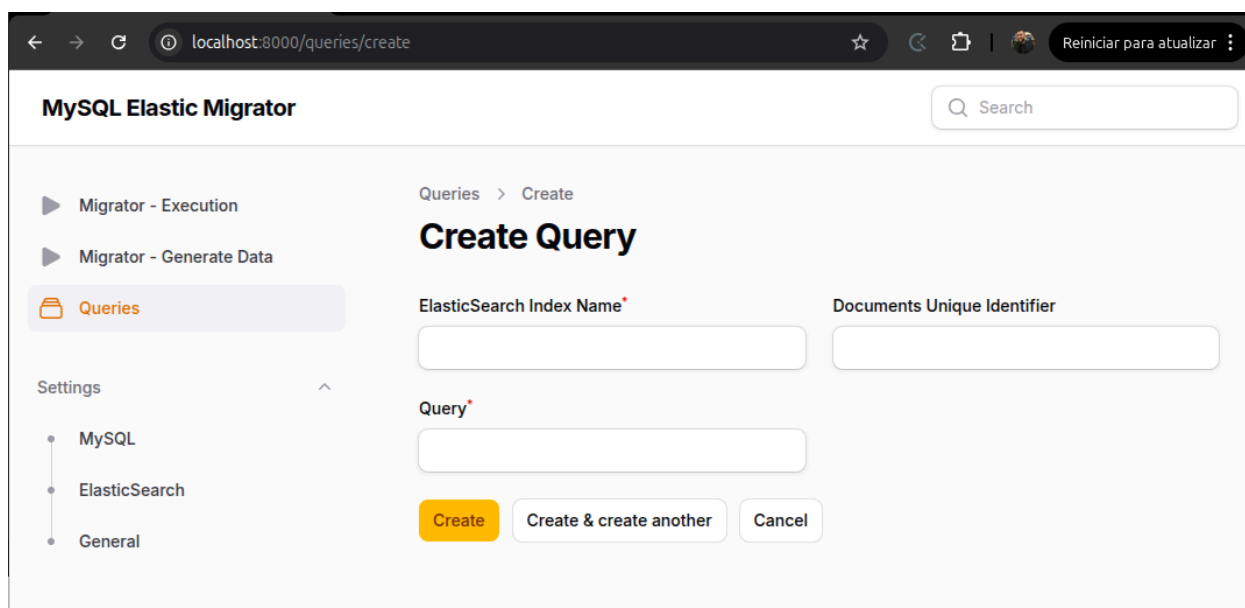
Fonte: Autoria Própria, 2025.

As telas de configurações constituem em três: MySQL, ElasticSearch e General. Cada uma das telas possui campos específicos de cada contexto. A do MySQL, que pode ser vista na Figura 9, e ElasticSearch servem para editar dados referentes à conexão com os bancos que fazem parte do processo de migração. Já a tela de configurações gerais, atualmente, possui apenas uma configuração que é referente a quantidade de registros processados em cada iteração

durante o processo de migração. Dessa forma, o usuário responsável durante a utilização do software pode fazer um ajuste para melhoria de performance da migração, tendo o devido cuidado com o valor porque sua variação influencia na quantidade de memória utilizada pela aplicação durante a migração.

O cadastro das consultas responsáveis pela busca dos dados que serão migrados é realizado através do menu Queries. Para cada consulta é necessário o preenchimento do nome do índice do ElasticSearch no qual serão salvas as informações, o comando SQL para consulta dos dados no MySQL e, de forma opcional, o preenchimento do campo retornado na consulta que será utilizado como id para os documentos criados no ElasticSearch. Essa tela de cadastro está exibida na Figura 10.

Figura 10 – MySQL Elastic Migrator: Cadastro de consulta

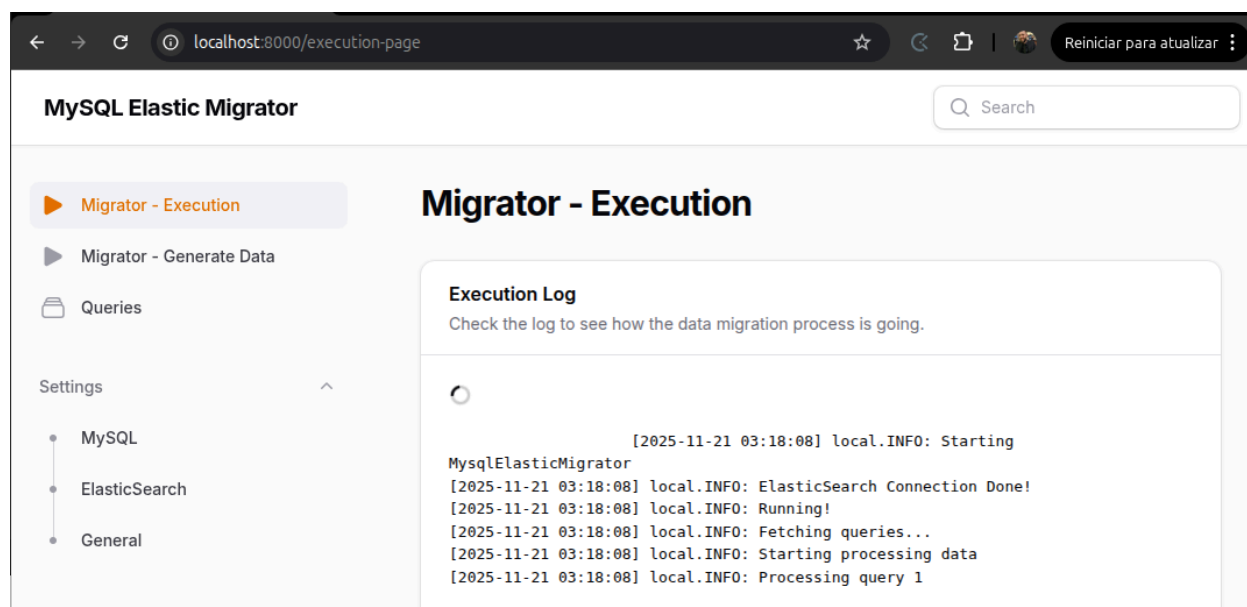


The screenshot shows the 'MySQL Elastic Migrator' web interface. The browser address bar indicates the URL 'localhost:8000/queries/create'. The page title is 'MySQL Elastic Migrator' and there is a search bar. The main content area is titled 'Create Query' and contains three input fields: 'ElasticSearch Index Name\*', 'Documents Unique Identifier', and 'Query\*'. Below the fields are three buttons: 'Create' (yellow), 'Create & create another', and 'Cancel'. On the left side, there is a sidebar menu with 'Queries' selected, and a 'Settings' section with 'MySQL', 'ElasticSearch', and 'General' options.

Fonte: Autoria Própria, 2025.

Com todas as configurações realizadas, a execução da migração é realizada diretamente no menu *Migrator - Execution*. Nesse ponto, é iniciado e acompanhado todo processo de migração até o seu encerramento como mostrado na Figura 11.

Figura 11 – MySQL Elastic Migrator: Execução



Fonte: Autoria Própria, 2025.

### 3.4 VERIFICAÇÃO DOS DADOS MIGRADOS

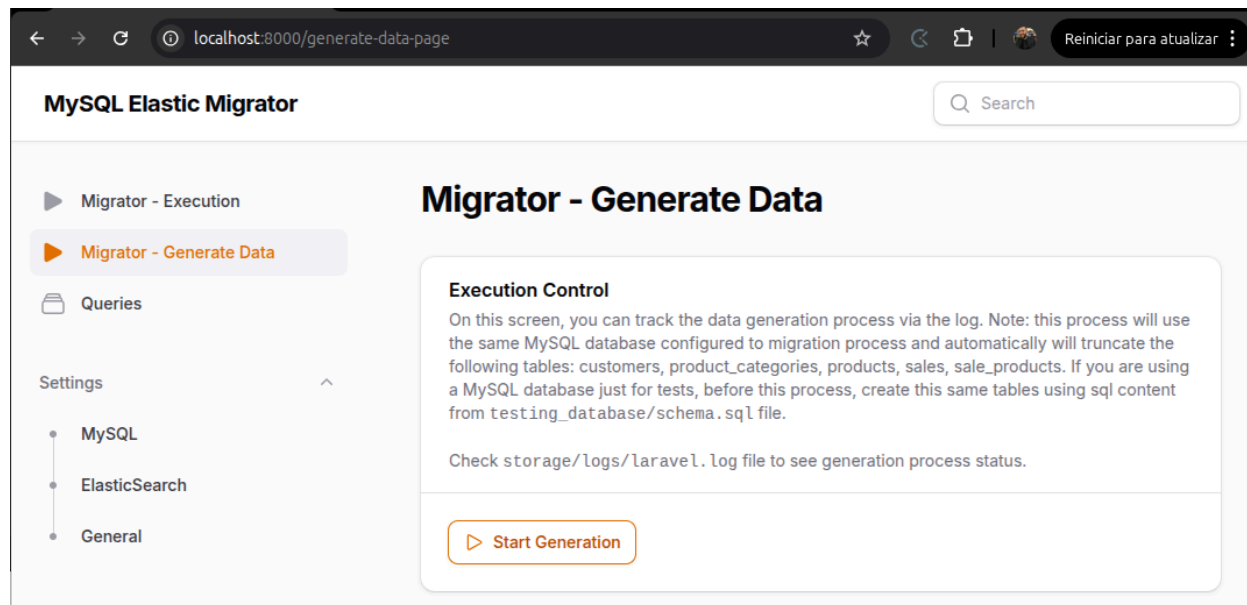
Juntamente com a configuração do Docker Compose para a aplicação, um serviço para a criação de um banco MySQL de teste também foi inserido. Assim, nesse banco foram criadas algumas tabelas básicas com vínculos entre si com o intuito de permitir consultas que pudessem ser utilizadas na aplicação de migração. Inicialmente foram inseridos alguns registros manualmente nesse banco de dados. O ElasticSearch, utilizado nos testes, foi criado na plataforma Elastic Cloud. A partir disso, foi executado o processo de migração de algumas dessas informações.

Inicialmente, para verificação dos dados migrados, foram realizados testes com pequenas quantidades de registros para de forma visual verificar se os registros estavam sendo migrados corretamente. Então, a partir de consultas no MySQL e visualização das informações migradas, diretamente na plataforma Elastic Cloud, foi possível verificar que o processo de migração ocorria corretamente.

No entanto, como a proposta é que essa aplicação funcione para contextos de grandes volumes de dados, foi implementada uma funcionalidade dentro da aplicação para gerar uma massa de dados de testes. A Figura 12 exhibe a tela para essa geração de dados. Essa

funcionalidade fica oculta na execução padrão da aplicação, mas através de mudança em variável de ambiente ela é disponibilizada no menu lateral.

Figura 12 – MySQL Elastic Migrator: Geração de dados



Fonte: Autoria Própria, 2025.

As tabelas, nas quais os dados são gerados, tem um contexto de vendas de produtos, constituindo um grupo de 5 tabelas: customers, product\_categories, products, sales e sale\_products. São gerados dados aleatórios para os campos, mas de forma coesa considerando os vínculos existentes entre os dados de tabelas diferentes. Com esse procedimento, foram então realizadas a criação de uma massa de dados de cerca de 725 mil registros resultantes em uma consulta para serem migrados.

Nessa primeira verificação, devido a grande quantidade de registros, foi verificado apenas se a quantidade de registros resultantes era igual ao de documentos criados no ElasticSearch. A migração foi executada e o número de documentos criados foi o número exato de registros retornados pela consulta.

Por fim, para aferir se os dados em si também estavam sendo migrados corretamente, não somente em quantidade mas também em conteúdo, foi realizado um teste com cerca de 20 mil registros. Apenas para complemento de informação, esse processo de migração ocorreu em 12 segundos. Como já mencionado, o contexto dos dados das tabelas é de uma estrutura de dados de vendas, portanto para verificação das informações foram feitas duas análises.

A primeira análise somente quantitativa referente ao total de registros a serem migrados, da mesma maneira que havia sido realizado anteriormente com o volume maior de dados. Uma consulta de contagem foi realizada no MySQL e, após o processo de migração, verificado no dashboard do Elastic Cloud a quantidade de documentos. No entanto, para verificação dos dados em si foram utilizados um conjunto de consultas no MySQL e ElasticSearch com agregações de dados, assim permitindo de forma indireta a realização da verificação de todos os dados.

Essas verificações consistiram na aferição das seguintes informações nos dois bancos de dados: somatório total do valor vendido, somatório total do valor vendido por cliente, somatório do total do valor vendido por produto, quantidade total de vendas por produto e quantidade total de unidades vendidas por produtos. Realizadas as consultas para obtenção desses valores em ambos os bancos os dados foram inseridos em arquivos excel. Nesses arquivos foram ordenados e comparados para verificar a correspondência. A Figura 13 mostra a comparação dos valores agregados por cliente enquanto a Figura 14 mostra a comparação dos valores agregados por produto.

Figura 13 – Verificação de dados agregados por cliente

|    | A                      | B            | C           | D                      | E                | F               | G       | H       | I       |
|----|------------------------|--------------|-------------|------------------------|------------------|-----------------|---------|---------|---------|
| 1  | cliente                | nº de vendas | valor total | SQL cliente            | SQL nº de vendas | SQL valor total |         |         |         |
| 2  | Aaron Wunsch           | 380          | 53441.3     | Aaron Wunsch           | 380              | 53441.30        | Correto | Correto | Correto |
| 3  | Abe Stoltenberg        | 295          | 42773       | Abe Stoltenberg        | 295              | 42773.00        | Correto | Correto | Correto |
| 4  | Alberta Glover         | 372          | 54160.4     | Alberta Glover         | 372              | 54160.40        | Correto | Correto | Correto |
| 5  | Chanelle Fay           | 404          | 55835.9     | Chanelle Fay           | 404              | 55835.90        | Correto | Correto | Correto |
| 6  | Connor Lind            | 356          | 46018.6     | Connor Lind            | 356              | 46018.60        | Correto | Correto | Correto |
| 7  | Dandre Crona           | 296          | 43330.8     | Dandre Crona           | 296              | 43330.80        | Correto | Correto | Correto |
| 8  | Danielle Marvin        | 272          | 39510.2     | Danielle Marvin        | 272              | 39510.20        | Correto | Correto | Correto |
| 9  | Dorothea Rosenbaum     | 394          | 58385.2     | Dorothea Rosenbaum     | 394              | 58385.20        | Correto | Correto | Correto |
| 10 | Dr. Adan Stanton       | 368          | 48971.9     | Dr. Adan Stanton       | 368              | 48971.90        | Correto | Correto | Correto |
| 11 | Dr. Hope Schinner      | 309          | 44593       | Dr. Hope Schinner      | 309              | 44593.00        | Correto | Correto | Correto |
| 12 | Dr. Maria Gutkowski II | 329          | 47597.5     | Dr. Maria Gutkowski II | 329              | 47597.50        | Correto | Correto | Correto |
| 13 | Dr. Reilly Kiehn V     | 303          | 44505.8     | Dr. Reilly Kiehn V     | 303              | 44505.80        | Correto | Correto | Correto |
| 14 | Dr. Willa Harvey       | 340          | 48797.9     | Dr. Willa Harvey       | 340              | 48797.90        | Correto | Correto | Correto |
| 15 | Edwin Kunze            | 356          | 49742.6     | Edwin Kunze            | 356              | 49742.60        | Correto | Correto | Correto |
| 16 | Evie Ratke             | 353          | 49096.1     | Evie Ratke             | 353              | 49096.10        | Correto | Correto | Correto |
| 17 | Foster McCullough      | 234          | 32085.4     | Foster McCullough      | 234              | 32085.40        | Correto | Correto | Correto |
| 18 | Frances Hudson         | 316          | 43998.4     | Frances Hudson         | 316              | 43998.40        | Correto | Correto | Correto |
| 19 | Hanna Gusikowski V     | 414          | 55871.6     | Hanna Gusikowski V     | 414              | 55871.60        | Correto | Correto | Correto |
| 20 | Hilbert Ankunding      | 352          | 49783.7     | Hilbert Ankunding      | 352              | 49783.70        | Correto | Correto | Correto |
| 21 | Hilton Buckridge       | 457          | 63567.4     | Hilton Buckridge       | 457              | 63567.40        | Correto | Correto | Correto |
| 22 | Howard Upton           | 502          | 73184.5     | Howard Upton           | 502              | 73184.50        | Correto | Correto | Correto |
| 23 | Jacquelyn Rowe DVM     | 334          | 49154.2     | Jacquelyn Rowe DVM     | 334              | 49154.20        | Correto | Correto | Correto |
| 24 | Jenifer Spinka         | 385          | 54392.8     | Jenifer Spinka         | 385              | 54392.80        | Correto | Correto | Correto |
| 25 | Jerald Pollich Jr.     | 489          | 69190.1     | Jerald Pollich Jr.     | 489              | 69190.10        | Correto | Correto | Correto |
| 26 | Judah Maggio           | 346          | 49239.8     | Judah Maggio           | 346              | 49239.80        | Correto | Correto | Correto |
| 27 | Juliana Robel          | 370          | 50435       | Juliana Robel          | 370              | 50435.00        | Correto | Correto | Correto |
| 28 | Kaylie Erdman          | 347          | 49121.2     | Kaylie Erdman          | 347              | 49121.20        | Correto | Correto | Correto |
| 29 | Lauren Pfeffer         | 467          | 65892.8     | Lauren Pfeffer         | 467              | 65892.80        | Correto | Correto | Correto |
| 30 | Lauriane Gutkowski II  | 402          | 57396       | Lauriane Gutkowski II  | 402              | 57396.00        | Correto | Correto | Correto |
| 31 | Leanna Rippin          | 338          | 48816.8     | Leanna Rippin          | 338              | 48816.80        | Correto | Correto | Correto |

Fonte: Autoria Própria, 2025.

Figura 14 – Verificação de dados agregados por produto

|    | A                       | B            | C                | D           | E                       | F          | G                    | H               | I       | J       | K       | L       |
|----|-------------------------|--------------|------------------|-------------|-------------------------|------------|----------------------|-----------------|---------|---------|---------|---------|
| 1  | produto                 | nº de vendas | quantidade total | valor total | SQL produto             | SQL vendas | SQL quantidade total | SQL valor total |         |         |         |         |
| 2  | Abelardo Kovacek        | 1041         | 26677            | 148298      | Abelardo Kovacek        | 1041       | 26677                | 148298.00       | Correto | Correto | Correto | Correto |
| 3  | Aliza Satterfield       | 1032         | 26413            | 149429      | Aliza Satterfield       | 1032       | 26413                | 149429.00       | Correto | Correto | Correto | Correto |
| 4  | Annabelle Stedemann     | 1048         | 26339            | 140200      | Annabelle Stedemann     | 1048       | 26339                | 140200.00       | Correto | Correto | Correto | Correto |
| 5  | Darion Mertz            | 1066         | 26498            | 147823.9    | Darion Mertz            | 1066       | 26498                | 147823.90       | Correto | Correto | Correto | Correto |
| 6  | Demond Ritchie          | 1042         | 26650            | 144705.7    | Demond Ritchie          | 1042       | 26650                | 144705.70       | Correto | Correto | Correto | Correto |
| 7  | Dr. Evans Kunze I       | 1054         | 26463            | 149069.7    | Dr. Evans Kunze I       | 1054       | 26463                | 149069.70       | Correto | Correto | Correto | Correto |
| 8  | Dr. Lera Brekke         | 1043         | 27125            | 146382.6    | Dr. Lera Brekke         | 1043       | 27125                | 146382.60       | Correto | Correto | Correto | Correto |
| 9  | Elmira Auer             | 1047         | 28088            | 154874.7    | Elmira Auer             | 1047       | 28088                | 154874.70       | Correto | Correto | Correto | Correto |
| 10 | Joseph Doolley          | 988          | 24690            | 135403.2    | Joseph Doolley          | 988        | 24690                | 135403.20       | Correto | Correto | Correto | Correto |
| 11 | Ludwig Wintheiser       | 1055         | 26545            | 149763.7    | Ludwig Wintheiser       | 1055       | 26545                | 149763.70       | Correto | Correto | Correto | Correto |
| 12 | Miss Palma Treutel DDS  | 1060         | 27731            | 150744.3    | Miss Palma Treutel DDS  | 1060       | 27731                | 150744.30       | Correto | Correto | Correto | Correto |
| 13 | Mr. Marvin Lowe PhD     | 1030         | 26181            | 145825.6    | Mr. Marvin Lowe PhD     | 1030       | 26181                | 145825.60       | Correto | Correto | Correto | Correto |
| 14 | Mrs. Burdette Jenkins   | 1055         | 27316            | 154756.8    | Mrs. Burdette Jenkins   | 1055       | 27316                | 154756.80       | Correto | Correto | Correto | Correto |
| 15 | Norberto Abshire        | 999          | 26144            | 143942.9    | Norberto Abshire        | 999        | 26144                | 143942.90       | Correto | Correto | Correto | Correto |
| 16 | Percival Schimmel       | 1015         | 24972            | 138953.4    | Percival Schimmel       | 1015       | 24972                | 138953.40       | Correto | Correto | Correto | Correto |
| 17 | Prof. Arch Schmeler III | 1069         | 28571            | 155855.5    | Prof. Arch Schmeler III | 1069       | 28571                | 155855.50       | Correto | Correto | Correto | Correto |
| 18 | Prof. Harold Will       | 1084         | 28035            | 155797.2    | Prof. Harold Will       | 1084       | 28035                | 155797.20       | Correto | Correto | Correto | Correto |
| 19 | Rosalie Greenfelder     | 1060         | 27589            | 150893.2    | Rosalie Greenfelder     | 1060       | 27589                | 150893.20       | Correto | Correto | Correto | Correto |
| 20 | Shania Gutkowski I      | 1051         | 27475            | 151194.1    | Shania Gutkowski I      | 1051       | 27475                | 151194.10       | Correto | Correto | Correto | Correto |

Fonte: Autoria Própria, 2025.

Após esse processo, foi verificado que todos os dados estavam correspondentes, confirmando que o processo de migração ocorreu corretamente. Aproveitando o nome do cliente e nome do produto, utilizados como agrupador nas verificações dos dados agregados, também foram verificados se os nomes correspondiam, validando também essa informação textual presente nos dados transferidos.

### 3.4.1 Validação da aplicação por profissional da área

Após o desenvolvimento da aplicação, um profissional da área, que atua em projeto que utiliza o MySQL como banco de dados base tendo grande volume de dados a serem manipulados via consultas SQL, se dispôs a realizar uma validação da aplicação já desenvolvida. Para a realização de momento por esse profissional, uma documentação foi criada no arquivo *README.md* do repositório do GitHub.

Para realização do teste apenas foi enviado o link do repositório do GitHub e solicitado que essa verificação fosse realizada, se possível, sem a interferência do pesquisador, utilizando apenas a documentação disponibilizada. Após o teste uma entrevista foi realizada para captação da experiência e resultados obtidos na utilização da aplicação.

Questionado nesta entrevista quanto ao funcionamento geral da aplicação, o entrevistado informou ter conseguido executar a aplicação corretamente e realizar a migração dos dados. Durante o processo de utilização do software, na parte de cadastro das consultas a serem realizadas, o entrevistado sugeriu a inserção de validação dessa consulta já no momento do

cadastro. Esse ponto surgiu devido a um erro de digitação cometido pelo entrevistado em meio ao teste, causando um erro somente no momento da execução da migração.

Ao ser indagado quanto ao êxito da aplicação na resolução do problema proposto, a resposta foi afirmativa. Dessa forma, o software foi considerado aprovado pelo entrevistado.

#### **4 CONCLUSÃO**

Com relação ao problema e ao software produto dessa pesquisa, pode-se afirmar que a resolução cumpriu com o objetivo de ter um sistema que se aplica no processo de migração de dados do MySQL para o ElasticSearch, independente da estrutura e formato dos dados presentes no MySQL. O uso pode ser feito tanto em um ambiente produtivo para migração dos dados e utilização de todos os benefícios propostos pelo ElasticSearch, tendo em vista a verificação de integridade dos dados que foram migrados.

Além disso, como um dos fatores que fizeram parte do contexto que gerou o problema da pesquisa é o crescente volume de dados em uma base MySQL de um sistema já existente, esse software pode ser utilizado como ferramenta para testar os recursos do ElasticSearch, migrando apenas algum conjunto de dados. Tornando possível a validação de um fluxo completo da utilização de ambos bancos de dados no sistema já existente.

Em conjunto ao êxito do software em questão após todas as verificações realizadas pelo pesquisador, bem como, a aprovação por profissional da área em teste para validação da ferramenta, também como parte do objetivo da pesquisa, que é a aplicação dos conhecimentos obtidos durante a realização do curso, foram possíveis, não só utilizar grande parte dos tópicos aprendidos no período do curso, como adquirir o aprendizado de conteúdos relacionados, como é o caso da Design Sprint, Docker, Docker Compose, Laravel e Filament PHP.

Apesar do funcionamento pleno do software, viu-se que há algumas melhorias que podem ser aplicadas na continuidade do projeto. Uma delas seria o desenvolvimento de uma etapa de verificação de integridade dos dados que foram migrados. Ao invés de necessitar uma validação posterior, como o que foi realizado nesta pesquisa, pode ser aplicado uma validação de cada registro, verificando a correspondência da informação entre os dois bancos de dados. Essa adição precisa ser avaliada quanto ao impacto de performance do processo de migração.

Também pode ser aplicado a interface web complementar páginas que venham a exibir tanto informações a respeito das consultas cadastradas quanto dos dados que estão no

ElasticSearch configurado. Dessa forma, o usuário responsável pela migração conseguiria ter centralizadas na mesma interface diversas informações para complementar esse processo.

## REFERÊNCIAS

ANICETO, Rodrigo Cardoso; XAVIER, Renê Freire. **Um estudo sobre a utilização do banco de dados NoSQL cassandra em dados biológicos**. Brasília, 2014. viii, 50 f. Monografia (Bacharelado em Ciência da Computação)—Universidade de Brasília. Disponível em: [https://bdm.unb.br/bitstream/10483/7927/1/2014\\_RodrigoCardosoAniceto\\_ReneFreireXavier.pdf](https://bdm.unb.br/bitstream/10483/7927/1/2014_RodrigoCardosoAniceto_ReneFreireXavier.pdf). Acesso em: 1 out. 2024.

COMPOSER. **Composer Documentation**. Disponível em: <https://getcomposer.org/doc/>. Acesso em: 17 nov. 2025.

DB-ENGINES. **DB-Engines Ranking of Relational DBMS**. Disponível em: <https://db-engines.com/en/ranking/relational+dbms>. Acesso em: 1 out. 2024.

DB-ENGINES. **DB-Engines Ranking of Search Engines**. Disponível em: <https://db-engines.com/en/ranking/search+engine>. Acesso em: 1 out. 2024.

DOCKER. **Docker Overview**. Disponível em: <https://docs.docker.com/get-started/docker-overview/>. Acesso em: 17 nov. 2025.

DOCKER. **Docker Compose Documentation**. Disponível em: <https://docs.docker.com/compose/>. Acesso em: 17 nov. 2025.

ELASTICSEARCH. **O que é Elasticsearch**. Disponível em: <https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro-what-is-es.html>. Acesso em: 20 nov. 2024.

ELASTIC. **Elasticsearch – Documentação oficial**. Disponível em: <https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html>. Acesso em: **17 nov. 2025**.

ELASTIC. **Elasticsearch Service on Elastic Cloud**. Elastic Documentation, [s.d.].

Disponível em: <<https://www.elastic.co/guide/en/cloud/current/ec-getting-started.html>>.  
Acesso em: 17 nov. 2025.

FILAMENTPHP. *FilamentPHP Documentation – Overview*. Disponível em:  
<https://filamentphp.com/docs/4.x/introduction/overview>. Acesso em: 21 nov. 2025.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6. ed. São Paulo: Addison Wesley, 2011.

ESTUARY. **Conexão em tempo real entre MySQL e Elasticsearch**. Disponível em:  
<https://estuary.dev/mysql-to-elasticsearch-real-time/>. Acesso em: 20 nov. 2024.

FERREIRA, Gilmar José da Silva; SILVA, Júlio Cesar Oliveira Ferreira. **Análise comparativa de desempenho de consultas entre um banco de dados relacional e um banco de dados não relacional**. Uberaba, 2017. Disponível em:  
<https://dspace.uniube.br:8443/handle/123456789/178>. Acesso em: 1 out. 2024.

GITHUB. **Sobre o GitHub e o Git**. GitHub Docs. Disponível em:  
<https://docs.github.com/pt/get-started/start-your-journey/about-github-and-git>. Acesso em: 17 nov. 2025.

LARAVEL. **Laravel Documentation**. *Installation: The PHP Framework For Web Artisans*. Disponível em: <https://laravel.com/docs/12.x>. Acesso em: 17 nov. 2025.

LÓSCIO, Bernadette F.; OLIVEIRA, H. R.; PONTES, J. C. S. **NoSql no desenvolvimento de aplicações Web colaborativas**. Disponível em:  
[https://www.researchgate.net/profile/Bernadette\\_Loscio/publication/268201466\\_NoSQL\\_no\\_desenvolvimento\\_de\\_aplicacoes\\_Web\\_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf](https://www.researchgate.net/profile/Bernadette_Loscio/publication/268201466_NoSQL_no_desenvolvimento_de_aplicacoes_Web_colaborativas/links/576aa72008aef2a864d1ef8c/NoSQL-no-desenvolvimento-de-aplicacoes-Web-colaborativas.pdf). Acesso em: 1 out. 2024.

MENEZES, Sandro Roberto Loiola de; FREITAS, Rebeca Schroeder; PARPINELLI, Rafael Stubs. **Mineração em Grandes Massas de Dados Utilizando Hadoop MapReduce e Algoritmos Bio-inspirados: Uma Revisão Sistemática**. Disponível em:  
<https://seer.ufrgs.br/rita/article/download/RITA-VOL23-NR1-69/37485>. Acesso em: 6 nov. 2024.

MORDOR INTELLIGENCE. **Tamanho do mercado de software de produtividade empresarial e análise de participação: tendências e previsões de crescimento (2024–2029).**

Disponível em:

<https://www.mordorintelligence.com/pt/industry-reports/global-business-productivity-software-market>. Acesso em: 6 nov. 2024.

O GLOBO. **Big data: análise de dados é aliada da indústria.** Disponível em:

<https://oglobo.globo.com/patrocinado/dino/noticia/2023/09/27/big-data-analise-de-dados-e-aliada-da-industria.ghtml>. Acesso em: 20 nov. 2024.

OLIVEIRA, Samuel Silva de. **Bancos de dados não-relacionais: um novo paradigma para armazenamento de dados em sistemas de ensino colaborativo.** Disponível em:

<https://www2.unifap.br/oliveira/files/2016/02/35-124-1-PB.pdf>. Acesso em: 20 nov. 2024.

ORACLE CORPORATION. *MySQL Documentation*. Disponível em:

<https://dev.mysql.com/doc/>. Acesso em: 15 nov. 2025.

PEREIRA, A. **Pesquisa Prática e Pesquisa Aplicada em Educação: Reflexões epistemo-metodológicas.** Disponível em:

<https://mestradoedoutoradoestacio.periodicoscientificos.com.br/index.php/reeduc/article/view/10598>. Acesso em: 04 mar. 2025.

SILVA, Rodrigo; COSTA, Catarina. **Aplicando a metodologia Design Sprint na produção de objetos educacionais.** In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 32., 2024, Brasília/DF. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024 . p. 274-284. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2024.2951>.

SOUZA, Rodrigo S.; VALENTIM, Natasha M. C.; COSTA, Catarina. **Aplicando Design Sprint em Sala de Aula Invertida: Um Estudo de Caso.** Disponível em:

<https://doi.org/10.5753/sbie.2023.235174>. Acesso em: 04 mar. 2025.