



Universidade Federal do ABC

UNIVERSIDADE FEDERAL DO ABC  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

## **ESTÁGIO SUPERVISIONADO III**

FELIPE AUGUSTO DOS SANTOS

Orientador: Prof. Dr. Daniel Morgato Martin

Santo André – SP  
2019

**FELIPE AUGUSTO DOS SANTOS**

**ANALISTA DE DESENVOLVIMENTO PLENO NA EMPRESA TENDA DIGITAL**

Relatório de estágio apresentado à banca do Bacharelado em Ciência da Computação da Universidade Federal do ABC como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Daniel Morgato Martin

Santo André – SP  
2019

## **DEDICATÓRIA**

Aos inquietos do mundo, que teimam em enxergar oportunidades onde outros só veem desespero.

## **AGRADECIMENTOS**

Ofereço meus sinceros agradecimentos:

Ao professor orientador Daniel, pela paciência e por sempre estar disposto à ouvir minhas ideias. E principalmente, por se empolgar tanto quanto eu pela oportunidade de aprender coisas novas.

À família Tenda, por proporcionar tudo o que um ambiente de trabalho deveria: companheirismo, desafios, obstáculos, risadas e muita, mas muita vontade de crescer.

"Os covardes nunca começam, os fracos morreram ao longo do caminho - e nós estamos aqui."

---

KNIGHT [4]

## **Resumo**

Este relatório tem como objetivo principal descrever as atividades realizadas como Analista de Desenvolvimento de Software na empresa Tenda Digital. O período descrito se refere à um sprint de um mês, onde foi desenvolvido a funcionalidade de busca por texto em diversas partes do sistema.

**Palavras-Chave:** Desenvolvimento Web, Desenvolvimento Mobile, Tenda Digital, JavaScript, Engenharia de Software

## **Abstract**

The main goal of this report is to describe the activities accomplished as an Analyst of Software Development at Tenda Digital. The period described here refers to a one-month sprint, where it was developed a text search feature across the entire system.

**Keywords:** Web Development, Mobile Development, Tenda Digital, JavaScript, Software Engineering

# Sumário

<b>1</b>	<b>Introdução</b>	<b>8</b>
1.1	Caracterização e análise da empresa Tenda Digital . . . . .	8
1.2	Instituição concedente: Tenda Digital Ltda. . . . .	9
1.3	Objetivos a serem alcançados . . . . .	9
<b>2</b>	<b>Descrição das atividades desenvolvidas</b>	<b>10</b>
2.1	Visao Geral . . . . .	10
2.2	Pesquisa . . . . .	12
2.3	Infraestrutura . . . . .	12
2.4	Sincronização de dados entre MongoDB e ElasticSearch . . . . .	14
2.5	Conversor de queries HTTP para ElasticSearch . . . . .	15
2.6	Estender back-end com rotas de busca . . . . .	17
2.7	Estender front-end para suportar buscas textuais . . . . .	18
<b>3</b>	<b>Fundamentação</b>	<b>20</b>
3.1	Disciplinas correlacionadas . . . . .	20
3.2	Conhecimento aplicado . . . . .	20
<b>4</b>	<b>Considerações finais</b>	<b>21</b>
4.1	Contribuições . . . . .	21



## Lista de Figuras

1	Logo da Tenda Digital . . . . .	9
2	Aplicativo publicado na Play Store . . . . .	10
3	Sistema administrativo . . . . .	11
4	Criação do cluster na Google Cloud Platform . . . . .	13
5	Visão geral do cluster . . . . .	13
6	Elasticsearch e Kibana em produção . . . . .	14
7	Interface CRUD de publicações . . . . .	18
8	Implementação da barra de busca . . . . .	19
9	Barra de busca em funcionamento . . . . .	19

# 1 Introdução

Nesta seção, apresenta-se a instituição concedente, a caracterização do estágio e da empresa, e uma visão geral do estágio.

## 1.1 Caracterização e análise da empresa Tenda Digital

O estágio foi realizado na empresa Tenda Digital, localizado na Estrada das Lágrimas, 111 - São Caetano do Sul (CNPJ 28.493.420/0001-75). A Tenda Digital é uma startup (dez funcionários) que desenvolve projetos de hardware e software (web e mobile). Entretanto não é uma fábrica de software, pois escolhe em quais projetos participar, pois normalmente faz parte societária do projeto em questão.

A Tenda Digital foi fundada em 2007 por Ivan Seidel, conhecido nos Torneios de Robótica e campeão mundial do Torneio First LEGO League (FLL) em Atlanta, já foi competidor, juiz e criou o sistema que gerenciou 27 ligas de robótica na RoboCup WorldCup Brasil e na Olimpíada Brasileira de Robótica.

A motivação por trás da fundação da empresa veio principalmente para poder atuar em projetos de terceiros. Entretanto, com o passar do tempo a necessidade da Tenda Digital se transformar foi aumentando devido a diversos projetos próprios que foram sendo desenvolvidos com o passar dos anos.

Então, em 2015 Ivan se juntou a Danilo Yoneshige e eles iniciaram seu primeiro projeto juntos como sócios, desenvolvendo um produto que hoje consta no portfólio da Tenda Digital chamado WisePix, um sistema que analisa fotos tiradas em eventos e as envia automaticamente para o Facebook do participante.

Em 2017, os negócios se expandiram com os investimentos de Marcos Wesley na empresa. Empresário com mais de 20 anos de experiência no setor de educação tecnológica, ele viu potencial em um aplicativo educacional para facilitar a comunicação de pais e alunos com a escola. E hoje, grande parte da equipe mantém seu foco neste projeto, denominado Tenda Edu.

Alguns dos projetos que atualmente constam no portfólio da Tenda Digital são:

- WisePix
- ZOOM Realidade Aumentada
- Tournamenter

Uma das características mais marcantes é o ambiente de trabalho descontraído e os horários flexíveis, que facilitam o balanceamento entre o estágio e a universidade.

## 1.2 Instituição concedente: Tenda Digital Ltda.



Figura 1: Logo da Tenda Digital

**Endereço:** Estrada das Lágrimas, 111

**CEP:** 09581-300

**Cidade:** São Caetano do Sul

**Estado:** SP

**CNPJ:** 28.493.420/0001-75

**Representado por:** Danilo Yoneshige

**Cargo:** Sócio

**Supervisor:** Danilo Yoneshige

**Horário:** 09h às 18h

**Total semanal:** 44 horas.

**Atividades:**

- Estruturação da equipe de desenvolvedores;
- Modelagem e implementação de Banco de Dados;
- Desenvolvimento de sistema web;
- Análise de Sistemas e/ou planejamento de Interfaces;

## 1.3 Objetivos a serem alcançados

Durante o período descrito neste relatório, o objetivo principal foi estender as funcionalidades do produto principal desenvolvido pela Tenda Digital, o Tenda Edu, principalmente no que se refere à criação de um sistema de busca por texto onde os usuário pudessem de forma simples e eficiente encontrarem os melhores resultados para suas buscas.

## 2 Descrição das atividades desenvolvidas

Esta seção tem por objetivo descrever as atividades efetuadas na realização do Estágio Supervisionado III, explicando como foram aplicados os conhecimentos obtidos no Bacharelado em Ciências da Computação.

### 2.1 Visao Geral

O produto principal da Tenda Digital é chamado Tenda Edu, um aplicativo que tem o objetivo facilitar o relacionamento entre escola, familiares e alunos. Para atingir este objetivo ele possui diversas funcionalidades, dentre elas:

- **Agenda Personalizada:** Responsáveis podem visualizar a agenda de cada aluno organizada em um calendário personalizado
- **Linha do Tempo:** Um resumo dos comunicados e atividades, organizados em uma linha do tempo interativa.
- **Helpdesk Integrado:** Responsáveis podem abrir chamados diretamente com a escola em diversos canais (secretaria, financeiro, etc...)

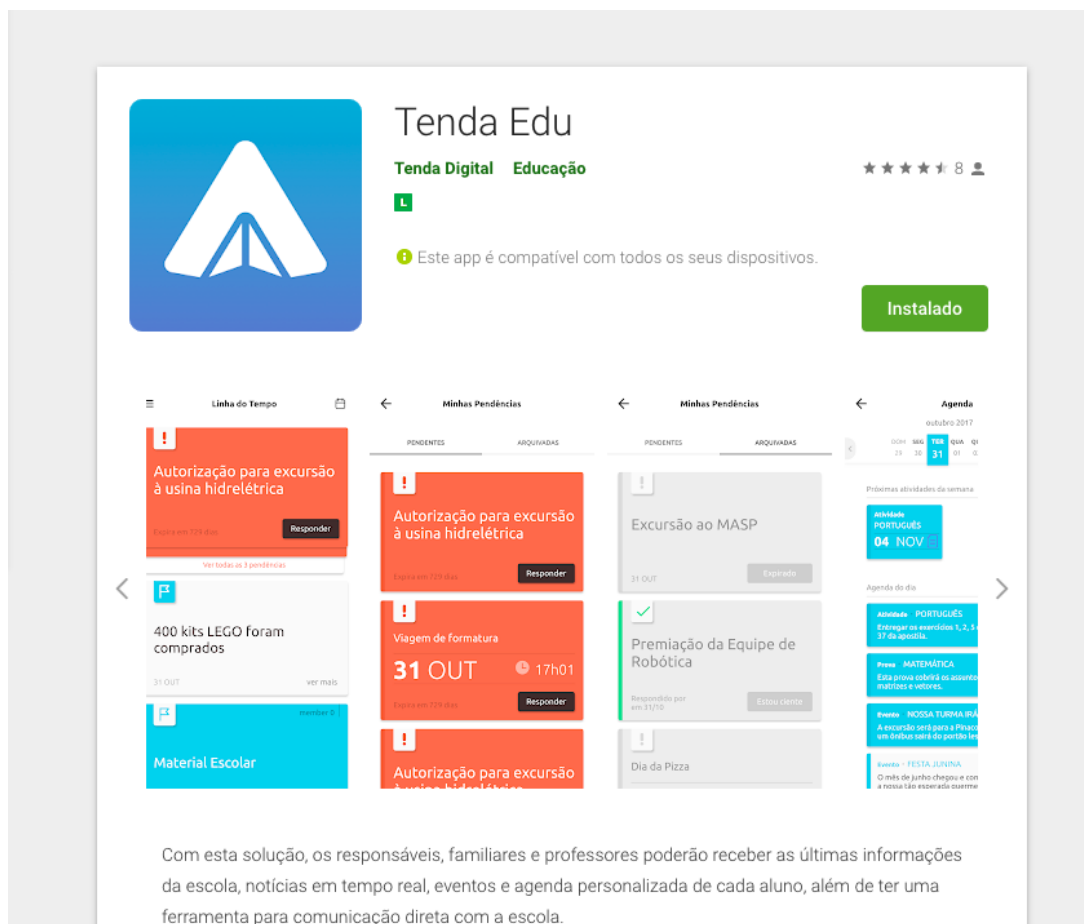


Figura 2: Aplicativo publicado na Play Store

Para facilitar a gestão por parte da escola, foi desenvolvido uma interface administrativa onde diversos membros conseguem tomar ações de gestão, como por exemplo: criar novos usuários, ensalar alunos, agendar publicações na linha do tempo e até mesmo responder solicitações dos pais e responsáveis.

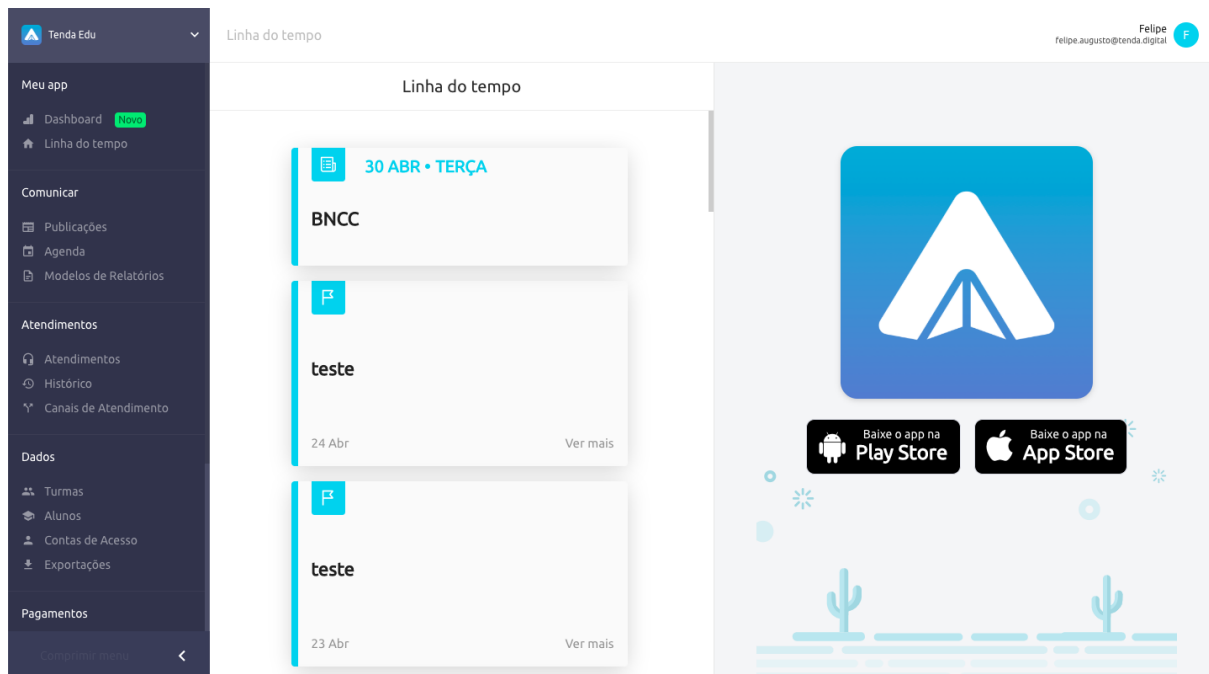


Figura 3: Sistema administrativo

Entretanto, uma das funcionalidades mais importantes ainda precisava ser desenvolvida, a busca por texto. O objetivo dessa funcionalidade é permitir que o usuário encontre os melhores resultados para sua busca textual, esteja buscando por uma publicação feita anteriormente, o nome de um responsável pai para verificar se o cadastro dele está ativo ou até mesmo uma turma.

Dada a importância desta funcionalidade, foi adicionada a *pipeline* de desenvolvimento a tarefa de implementar uma ferramenta de busca textual em diversas partes no sistema. As etapas a seguir foram efetuadas até se atingir a implementação total da funcionalidade, entretanto, tais etapas não estavam claras *a priori* e foram sendo modificadas conforme a equipe ganhava mais conhecimento e entendimento das opções e possíveis soluções para o problema.

- Pesquisa;
- Infraestrutura
- Sincronização de dados entre MongoDB e Elasticsearch;
- Conversor de queries HTTP para Elasticsearch;
- Estender *back-end* com rotas de busca;
- Estender *front-end* para suportar buscas textuais;

A execução e implementação destas funcionalidades serão descritas em seguida.

## 2.2 Pesquisa

A primeira etapa foi a de pesquisa para entender como o problema poderia ser resolvido, quais algoritmos poderiam ser utilizados e quais estruturas de dados implementadas. Durante a pesquisa encontrou-se duas ferramentas open-source especializadas em efetuar buscas textuais em tempo hábil:

- Apache Solr <sup>1</sup>
- ElasticSearch<sup>2</sup>

Analisando melhor as duas ferramentas, percebeu-se que o ElasticSearch possuía mais funcionalidades além de busca textual e a empresa que o desenvolve possui diversos produtos integrados a ele, como por exemplo o Kibana, uma ferramenta de visualização de dados. Além disso, ele é usado e confiado por diversas empresas, como por exemplo Facebook, Uber e Netflix. A etapa de pesquisa foi concluída com a decisão de se utilizar o ElasticSearch como ferramenta de busca textual.

## 2.3 Infraestrutura

Existem duas opções para se criar uma infraestrutura contendo ElasticSearch, pode-se hospedar diretamente em qualquer servidor ou contratar uma empresa que provenha esse tipo de serviço (*SaaS - Software as a Service*). A opção preferida era contratar uma destas empresas, pois elas lidam com diversos problemas de escalabilidade, atualização, rede, segurança, entre outros que ocorrem quando se gerencia a própria infraestrutura, tanto em cloud quando localmente.

Encontraram-se diversas empresas que prestavam tal serviço, porém os custos estavam muito além do que imaginou-se, por isso decidiu-se hospedar diretamente em uma infraestrutura cloud, mais especificamente a Google Cloud Platform<sup>3</sup>. Para isso criou-se um cluster Kubernetes<sup>4</sup> na Google Cloud Platform e seguiu-se o próprio tutorial da Google para configurar o ambiente de produção.

YANSYAH [6]

---

<sup>1</sup>Apache Solr. Disponível em: <https://lucene.apache.org/solr/>

<sup>2</sup>ElasticSerch. Disponível em: <https://www.elastic.co/>

<sup>3</sup>Google Cloud Plataform. Disponível em: <https://cloud.google.com/>

<sup>4</sup>Kubernetes. Disponível em: <https://kubernetes.io/>

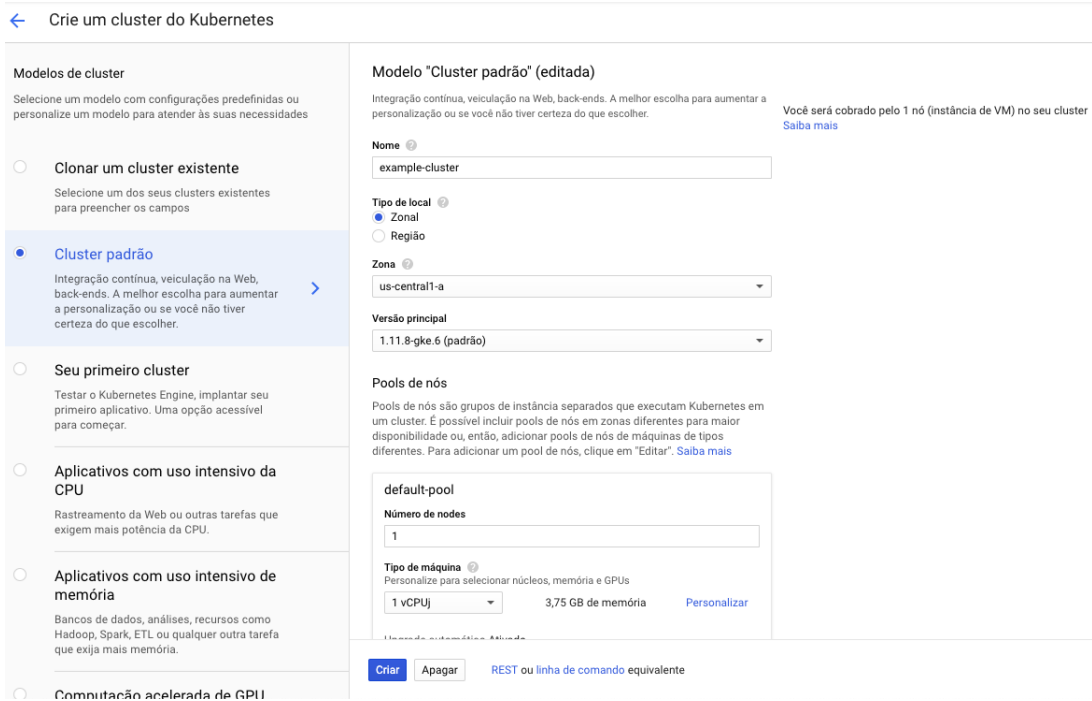


Figura 4: Criação do cluster na Google Cloud Platform

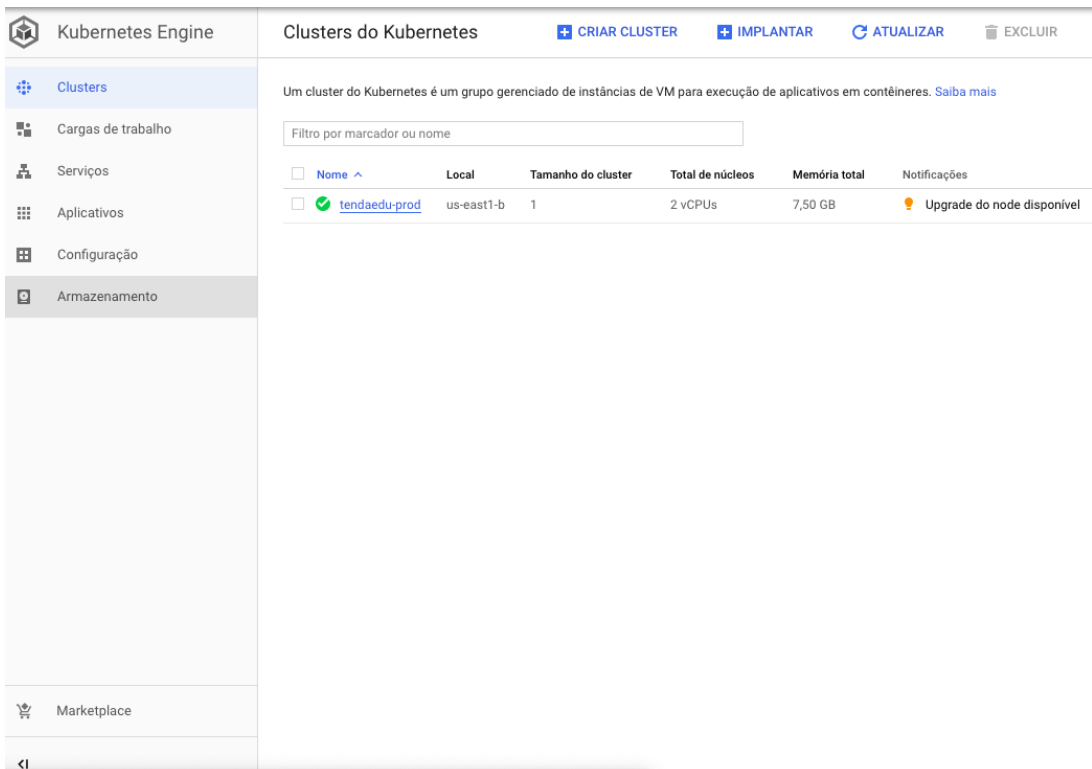


Figura 5: Visão geral do cluster

Kubernetes Engine

Cargas de trabalho

ATUALIZAR IMPLANTAR EXCLUIR

As cargas de trabalho são unidades de computação implantáveis que podem ser criadas e gerenciadas em um cluster.

É objeto do sistema: Falso Filtrar cargas de trabalho Colunas

Nome	Status	Tipo	Pods	Namespace	Grupo
elasticsearch	OK	Stateful Set	2/2	infra	tendaedu-prod
kibana-kibana	OK	Deployment	1/1	infra	tendaedu-prod

Figura 6: Elastisearch e Kibana em produção

## 2.4 Sincronização de dados entre MongoDB e ElasticSearch

O banco de dados usado no Tenda Edu é o MongoDB,<sup>5</sup> um banco de dados não-relacional, altamente escalável. Para interagir com o banco utiliza-se a ferramenta de ODM (Object-Document Mapping) chamada *Mongoose*<sup>6</sup>. ODM's estão para bancos não-relacionais assim como ORM's (Object Relational-Mapping) para banco relacionais. A função de ambos é mapear instâncias de classes para registros em banco de dados, podendo-se interagir com as instâncias ao invés de efetuar operações diretamente no banco. O'NEIL [5]

Durante a fase de pesquisa e quando encontrou-se o ElasticSearch percebeu-se que ele seria capaz de armazenar os dados de forma muito similar ao MongoDB, por isso a primeira dúvida foi se seria necessário migrar completamente os dados do MongoDB para o ElasticSearch e utilizar apenas este.

Entretanto, após novas pesquisas descobriu-se que a melhor prática é usar o ElasticSearch em conjunto com outro banco, seja ele não-relacional (MongoDB), ou relacional (MySQL, PostgreSQL). Ou seja, na prática deveria haver uma sincronização entre os dados do MongoDB com o ElasticSearch, toda vez que um documento fosse criado, atualizado ou excluído no MongoDB a mesma operação deveria ser feita no ElasticSearch.

Como é de costume, ao invés de implementar uma solução do zero, sempre pesquisa-se se alguém já desenvolveu uma solução para o problema, nesse caso encontrou-se uma solução chamada *mongoosastic*<sup>7</sup>, um plugin para o *Mongoose*, que efetua a sincronização de forma automática. A etapa de desenvolvimento se resumiu à instalar o plugin no *back-end*, efetuar sua configuração e

<sup>5</sup>MongoDB. Disponível em: <https://www.mongodb.com/>

<sup>6</sup>Mongoose. Disponível em: <https://mongoosejs.com/>

<sup>7</sup>Mongoosastic. Disponível em: <https://github.com/mongoosastic/mongoosastic>



efetuar testes para verificar se a sincronização estava funcionando como deveria. Quando verificou-se que tudo estava correto, prosseguiu-se para a próxima etapa.

## 2.5 Conversor de queries HTTP para Elasticsearch

O *back-end* do Tenda Edu foi desenvolvido em Node.js <sup>8</sup>, a decisão de se usar JavaScript no servidor foi tomada com o intuito de diminuir a quantidade de linguagens utilizadas nos projetos da empresa, visto que tanto o *front-end* quanto o aplicativo (híbrido) também se utilizam desta linguagem.

JavaScript é uma linguagem tanto orientada à objetos quanto funcional [1], isso propociona uma liberdade extra na hora de modelar o sistema. Possibilitando tanto o uso de conceitos como encapsulamento, herança, polimorfismo [3], assim como imutabilidade, funções puras e funções de alta ordem. [2]

O sistema utiliza o framework web Express <sup>9</sup> para disponibilizar uma API HTTP, que cria uma interface com diversas coleções do MongoDB (análogo à tabelas em banco relacionais), provendo operações básicas de CRUD assim como diversas outras.

Uma das funcionalidades que o sistema já possuía em praticamente todas as rotas de listagem do sistema (devido a outra biblioteca chamada *api-query-params*<sup>10</sup>) é a capacidade de converter parâmetros de query strings de URL's HTTP para queries do MongoDB. Por exemplo, a seguinte URL:

```
https://api.example.com/users?  
  active=true&  
  sort=-createdAt&  
  name=felipe&  
  createdAt<=2019-04-28
```

É convertida pela seguinte query em MongoDB:

```
{  
  "filter": {  
    "active": true,  
    "createdAt": {  
      "$lte": "2019-04-28T00:00:00.000Z"  
    },  
    "name": "felipe"  
  },  
  "sort": {  
    "createdAt": -1  
  }  
}
```

---

<sup>8</sup>Node.js. Disponível em: <https://nodejs.org/>

<sup>9</sup>Express. Disponível em: <http://expressjs.com/>

<sup>10</sup>api-query-params. Disponível em: <https://www.npmjs.com/package/api-query-params>

Isso facilita na generalização do código, pois em qualquer rota de listagem o código é praticamente o mesmo, como pode-se observar no exemplo abaixo.

```
// Convert query from URL to MongoDB
const query = require('api-query-params')(request.query)

// Execute query on a generic collection
const documents = await Collection
  .find(query.filter)
  .sort(query.sort)
  .limit(Math.min(query.limit, 100))
  .skip(query.skip)
```

O desafio agora era converter a query para ElasticSearch da mesma forma, procurou-se alguma biblioteca que efetuava essas operações, entretanto, não foi encontrada nenhuma que satisfazia as necessidades. Portanto, foi necessário desenvolver do zero esse conversor. Inicialmente pensou-se em fazer um conversor de URL's HTTP diretamente para ElasticSearch, entretanto, decidiu-se escrever um conversor de MongoDB para ElasticSearch, pois como explicado anteriormente o conversor de URL's HTTP para MongoDB já existia.

O custo extra para se fazer duas conversões ao invés de uma é compensado por haver diversas rotas no sistema onde se faz necessário o uso de um conversor de MongoDB para ElasticSearch diretamente e a primeira ideia não seria suficiente para cobrir esse caso.

Com o conversor implementado, após a conversão da query de MongoDB para ElasticSearch, obtêm-se o seguinte resultado:

```

{
  "query": {
    "bool": {
      "filter": [
        {
          "term": {
            "active": "true"
          }
        },
        {
          "range": {
            "createdAt": {
              "lte": "2019-04-28T14:49:31.375Z"
            }
          }
        }
      ],
      {
        "term": {
          "name.keyword": "felipe"
        }
      }
    ]
  },
  "sort": {
    "createdAt": "desc"
  }
}

```

Existe um planejamento para deixar esse conversor open-source, pois acreditamos que a comunidade se beneficiaria com essa solução.

## 2.6 Estender back-end com rotas de busca

De posse do conversor, seria possível criar novas rotas do back-end para que a API efetuasse buscas no Elasticsearch ao invés do MongoDB. Para isso, implementou-se o seguinte padrão de rotas:

```

GET /usuarios          (MongoDB)
GET /usuarios/busca   (ElasticSearch)

```

Esse padrão não vale apenas para a coleção de usuários, mas para diversas outras como por exemplo: turmas, postagens e tickets de atendimento. Desenvolveu-se o código de todas as rotas que foram consideradas essenciais possuem busca textual.

## 2.7 Estender front-end para suportar buscas textuais

O sistema web do Tenda Edu já possuía um painel administrativo com diversas interfaces para operações de CRUD. As tecnologias utilizadas para o desenvolvimento da interface foram HTML<sup>11</sup> e CSS<sup>12</sup>, que são normalmente utilizadas em qualquer projeto que seja acessável via browser.

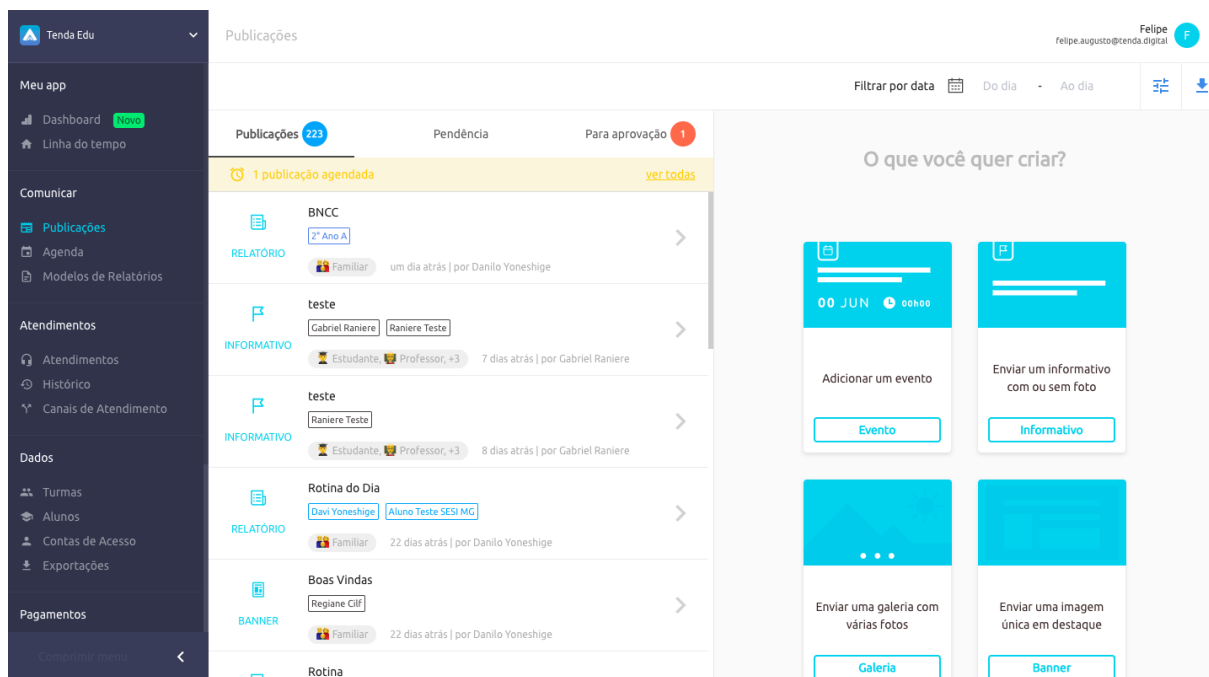


Figura 7: Interface CRUD de publicações

Além disso também utilizou-se um framework JavaScript de front-end chamado Vue.js<sup>13</sup>. Utilizando-se de componentes pré estilizados de algumas bibliotecas de *UI* famosas de Vue.js, como ElementUI<sup>14</sup> e Vuetify<sup>15</sup>, sendo necessários apenas alguns ajustes de CSS.

Desenvolveu-se uma barra de busca superior para que os usuários pudessem efetuar buscas, além disso, a rota de API que o front-end fazia as requisições foi modificada, trocou-se a URL de listagem do MongoDB para a do Elasticsearch. O resultado final pode ser observado nas imagens abaixo:

<sup>11</sup>HTML. Disponível em: <https://www.w3schools.com/html/>

<sup>12</sup>CSS. Disponível em: <https://www.w3schools.com/css/>

<sup>13</sup>Vue.js. Disponível em: <https://vuejs.org/>

<sup>14</sup>ElementUI. Disponível em: <https://element.eleme.io/>

<sup>15</sup>Vuetify. Disponível em: <https://vuetifyjs.com/>

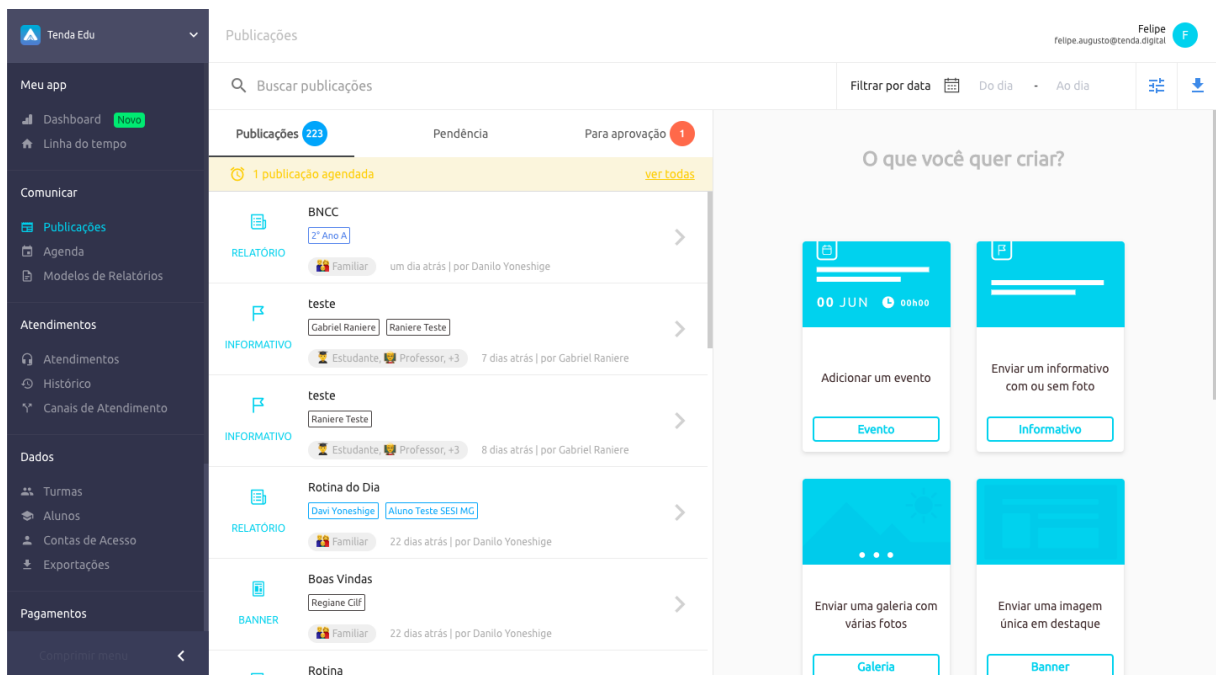


Figura 8: Implementação da barra de busca

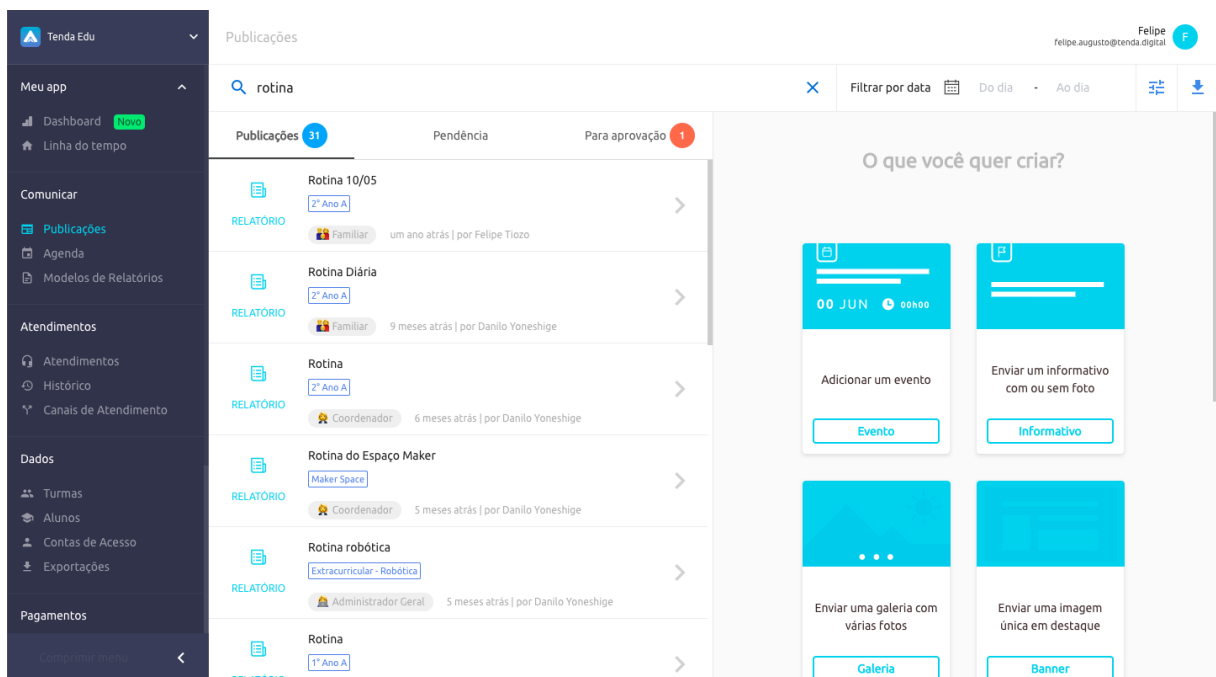


Figura 9: Barra de busca em funcionamento

### 3 Fundamentação

#### 3.1 Disciplinas correlacionadas

Para a execução das tarefas foram tomadas como base de conhecimento, principalmente, as matérias contidas na tabela 1.

Lógica Básica	<ul style="list-style-type: none"><li>• Operadores lógicos</li></ul>
Programação Orientada a Objetos	<ul style="list-style-type: none"><li>• Paradigma de programação orientada a objetos</li></ul>
Algoritmos e Estrutura de Dados	<ul style="list-style-type: none"><li>• Ter opinião formada e criticidade ao escolher certos tipos de algoritmos</li><li>• Questionamento sobre as linguagens e quais algoritmos seus métodos utilizam</li></ul>
Banco de Dados	<ul style="list-style-type: none"><li>• Modelos de banco de dados</li><li>• Linguagem SQL</li></ul>
Redes de Computadores	<ul style="list-style-type: none"><li>• Conhecimentos dos protocolos TCP/IP e HTTP e funcionamento da internet de forma geral</li></ul>
Paradigmas de Programação	<ul style="list-style-type: none"><li>• Programação funcional</li></ul>
Programação para WEB	<ul style="list-style-type: none"><li>• Padrão MVC, servidores, etc...</li><li>• HTML, CSS</li></ul>
Interface Humano-Computador	<ul style="list-style-type: none"><li>• Heurísticas de Nielsen</li><li>• Design de Interfaces</li><li>• Usabilidade</li></ul>

Tabela 1: Matérias relacionadas ao estágio

#### 3.2 Conhecimento aplicado

Das disciplinas correlacionadas, especifica-se as que tiveram maior impacto e que foram imprescindíveis durante o estágio.

**Programação para WEB** Provavelmente a matéria com maior impacto, pois mesmo sendo ofertada pela UFABC em Java, apresenta diversos elementos utilizados com frequência em qualquer projeto orientado para WEB: Padrão MVC, desenvolvimento de API's, HTML, CSS, etc...

**Programação Orientada a Objetos** Diversos conceitos de programação orientada a objetos foram utilizados, utilizou-se classes, objetos, herança e encapsulamento. Principalmente no desenvolvimento da API que segue o padrão MVC.

**Redes de Computadores** O entendimento de como a internet funciona, IP, DNS, latência, redes privadas e como os pacotes de dados trafegam na rede são os pontos mais importantes e aplicáveis da disciplina durante este estágio.

## **4 Considerações finais**

Essa seção tem como objetivo mostrar as contribuições do estágio para a empresa, para o estagiário e as maiores dificuldades encontradas durante o período retratado .

### **4.1 Contribuições**

O estágio é essencial para a formação profissional da área de Computação, é nele que o mundo acadêmico se choca com o mundo corporativo e então o aprendizado feito em sala de aula ganha forma e praticidade. As atividades realizadas permitiram que uma carreira e um foco profissional fosse tomado, bem como auxiliou no acompanhamento das disciplinas e no melhor entendimento do conteúdo exposto pelos Docentes.

Um ponto crucial destacável neste período foi o fato de não apenas aprender, ou reforçar, algumas linguagens e seu contexto, mas também e mais importante, como programar de forma limpa, coerente e eficaz. Além disso, o aprendizado e a dinâmica de um projeto e do desenvolvimento de um software será de suma importância para a carreira como Bacharel em Ciência da Computação.

## Referências

- [1] L. ATENCIO. *Functional Programming in JavaScript*. Manning, 2016.
- [2] R.; WADLER P. BIRD. *Introduction to Functional Programming*. Prentice Hall, 1994.
- [3] P. J. DEITEL, H. M.; DEITEL. *Java: como programar*. Bookman, 2005.
- [4] P. KNIGHT. *Shoe Dog: A Memoir by the Creator of Nike*. Scribner, 2016.
- [5] E. J. O'NEIL. *Object/relational mapping 2008: hibernate and the entity data model (edm)*. Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008.
- [6] B. A. YANSYAH. A guide to deploy elasticsearch cluster on google kubernetes engine, 2018. URL <https://medium.com/google-cloud/a-guide-to-deploy-elasticsearch-cluster-on-google-kubernetes-engine-52f67743ee98>.