

**UNIVERSIDADE DE TAUBATÉ**  
**Vitor Gonçalves Dias Giacomini**

**HUB DE ESTIMATIVAS:**  
**Uma aplicação web para otimização do processo**  
**de estimativa de custos**

**Taubaté - SP**  
**2023**

**Vitor Gonçalves Dias Giacomini**

**HUB DE ESTIMATIVAS:**

**Uma aplicação web para otimização do processo  
de estimativa de custos**

Trabalho apresentado para obtenção do grau  
Bacharelado pelo Curso Engenharia de  
Computação do Departamento de Informática da  
Universidade de Taubaté, Área de Concentração:  
Engenharia de Computação

Orientador: Prof. Me. Dawilmar Guimarães Araújo

**Taubaté - SP**

**2023**

**Grupo Especial de Tratamento da Informação - GETI  
Sistema Integrado de Bibliotecas – SIBi  
Universidade de Taubaté - Unitau**

G429h    Giacomini, Vitor Gonçalves Dias  
          Hub de estimativas: uma aplicação web para otimização do processo de  
          estimativa de custos / Vitor Gonçalves Dias Giacomini. -- 2023.  
          51 f. : il.

          Monografia (graduação) – Universidade de Taubaté, Departamento de  
          Informática, 2022.  
          Orientação: Prof. Me. Dawilmar Guimarães Araújo, Departamento de  
          Informática.

          1. Aplicação web. 2. Estimativa de custos. 3. Metodologia SOLID. I.  
          Universidade de Taubaté. Departamento de Informática. Graduação em  
          Engenharia da Computação. II. título.

CDD – 005.1

**VITOR GONÇALVES DIAS GIACOMINI**

**Hub de Estimativas: Uma aplicação web para otimização do processo de estimativa de custos**

Trabalho apresentado para obtenção do grau Bacharelado pelo Curso Engenharia de Computação do Departamento de Informática da Universidade de Taubaté, Área de Concentração: Engenharia de Computação

Orientador: Prof. Me. Dawilmar Guimarães Araújo

Data: 05/12/2023

Resultado: APROVADO

**BANCA EXAMINADORA**

Prof. Me. Dawilmar Guimarães Araújo

Universidade de Taubaté

Assinatura: \_\_\_\_\_

Prof. Esp. Esgar Israel

Universidade de Taubaté

Assinatura: \_\_\_\_\_

Prof. Me. Luiz Eduardo Souza Evangelista

Universidade de Taubaté

Assinatura: \_\_\_\_\_

Dedico este trabalho à minha família, aos meus colegas de curso e aos inspiradores professores da universidade. Todo o apoio que recebi foi essencial para a minha evolução acadêmica e profissional.

## AGRADECIMENTOS

A minha família por todo o apoio dado desde bem antes da confecção desse trabalho, mais precisamente desde vinte e um de novembro do ano dois mil.

Ao Prof. Mestre Dawilmar Guimarães Araújo pela orientação com relação ao desenvolvimento desse trabalho e por todas as aulas ministradas durante o curso.

A Universidade de Taubaté por sua estrutura e seus professores que fizeram parte do meu desenvolvimento educacional nestes últimos anos.

## RESUMO

Dentro da área de estimativa de custos há necessidade de um constante controle de informações. Estimativas são geradas diariamente e seus detalhes devem estar facilmente disponíveis para todos da área, incluindo alterações em tempo real que podem ser feitas nestas. Surge, portanto, a necessidade de uma plataforma que possa atender todas as demandas apresentadas pela área. Este trabalho tem como objetivo abordar a implementação de uma aplicação web de gerenciamento de estimativas de custos dentro da área de estimativa de custos, aplicação esta que busca organizar melhor o processo de estimativas desde o ponto de cadastro pela parte do estimador, passando pela análise dessa estimativa, revisão e ação do supervisor e organização e visualização posterior das estimativas existentes no sistema. Para a confecção deste trabalho foram estudados os problemas enfrentados pelos estimadores e supervisores da área de estimativas de custo, bem como as ideias que poderiam melhorar tanto os processos que envolviam estimativas quanto um aperfeiçoamento na forma como o setor se apresenta para o resto da empresa, além da metodologia SOLID para a construção de um *software* de qualidade. Portanto, com esse *software*, busca-se suprir as necessidades já existentes dentro da área de estimativa de custos, como um gerenciamento de estimativas mais simples, como também aumentar a capacidade da área em se relacionar com outras áreas dentro de uma empresa, simplificando seus processos e destacando seus resultados.

Palavras-chave: Aplicação web, estimativa de custos, metodologia SOLID

## ABSTRACT

Within the area of cost estimation there is a need for constant control of information. Estimates are generated daily, and their details must be easily available to everyone in the area, including real-time changes that can be made to them. Therefore, the need for a platform that could meet all the demands presented by the area arises. This work aims to address the implementation of a management web application within the cost estimation area, an application that seeks to better organize the estimation process from the point of registration by the estimator, passing through the analysis of this estimate, review and supervisor's action and organization and subsequent visualization of estimates existing in the system. For the preparation of this work, the problems faced by estimators and supervisors in cost estimating were studied, as well as the ideas that could improve both the processes that involved estimates and an improvement in the way the sector presents itself to the rest of the company, in addition to the SOLID methodology for building quality software. Therefore, with this software, an attempt is made to meet the needs that already exist within the cost estimation area, such as simpler estimation management, as well as increasing the area's ability to relate to other areas within a company, simplifying its processes and highlighting their results.

Keywords: Web application, cost estimation, SOLID methodology



## LISTA DE ILUSTRAÇÕES

FIGURA 01: Interface do Visual Studio Code.....	19
FIGURA 02: Interface do IntelliJ IDEA.....	20
FIGURA 03: Interface do MySQL Workbench.....	21
FIGURA 04: Interface do Insomnia.....	22
FIGURA 05: Interface do GitHub no projeto estimations-hub (front-end).....	23
FIGURA 06: Fluxo padrão na área de estimativa de custos.....	25
FIGURA 07: Protótipo da página inicial.....	26
FIGURA 08: Protótipo da página de visão geral.....	27
FIGURA 09: Protótipo da página de estimativas.....	28
FIGURA 10: Protótipo da página de produtos.....	29
FIGURA 11: Protótipo da página de criação de nova estimativa.....	30
FIGURA 12: Módulo de estimativas.....	32
FIGURA 13: Página inicial.....	34
FIGURA 14: Página de visão geral.....	35
FIGURA 15: Página de estimativas.....	36
FIGURA 16: Página de produtos.....	37
FIGURA 17: Página de criação de estimativa.....	38
FIGURA 18: Diagrama de entidades do Spring Boot.....	42
FIGURA 19: <i>Endpoints</i> presentes no controlador do produto.....	43
FIGURA 20: <i>Endpoints</i> presentes no controlador do estimador.....	44
FIGURA 21: <i>Endpoints</i> presentes no controlador da estimativa.....	44
FIGURA 22: Resposta com código 200 (OK): Estimativa encontrada com sucesso.....	45
FIGURA 23: Resposta com código 400 ( <i>Bad Request</i> ): Requisição incorreta.....	46
FIGURA 24: Resposta com código 401 ( <i>Unauthorized</i> ): Sem autorização.....	46
FIGURA 25: Resposta com código 404 ( <i>Not Found</i> ): Estimativa não encontrada.....	47
FIGURA 26: Resposta com código 500 ( <i>Internal Server Error</i> ): Erro interno.....	47

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO</b> .....	11
<b>1.1.</b>	<b>Objetivos</b> .....	12
<b>1.2.</b>	<b>Justificativa</b> .....	12
<b>2.</b>	<b>REVISÃO BIBLIOGRÁFICA</b> .....	13
<b>2.1.</b>	<b>Estimativa de Custos</b> .....	13
<b>2.2.</b>	<b>Aplicação Web</b> .....	14
<b>2.3.</b>	<b>Metodologia SOLID</b> .....	15
<b>2.4.</b>	<b>Tecnologias Utilizadas</b> .....	16
2.4.1.	Linguagens.....	16
2.4.2.	Ferramentas de Desenvolvimento.....	18
<b>3.</b>	<b>DESENVOLVIMENTO</b> .....	24
<b>3.1.</b>	<b>Definição dos requisitos</b> .....	24
<b>3.2.</b>	<b>Protótipo</b> .....	25
<b>3.3.</b>	<b>Definição do tipo do sistema</b> .....	30
<b>4.</b>	<b>RESULTADOS</b> .....	31
<b>4.1.</b>	<b>Front-end</b> .....	31
4.1.1.	Estrutura.....	31
4.1.2.	Página Inicial.....	33
4.1.3.	Visão Geral.....	34
4.1.4.	Estimativas.....	35
4.1.5.	Produtos.....	36
4.1.6.	Criação de estimativa.....	38
<b>4.2.</b>	<b>Back-end</b> .....	39

4.2.1. Estrutura.....	39
4.2.2. Entidades.....	41
4.2.3. Endpoints.....	43
<b>5. CONCLUSÕES.....</b>	<b>48</b>
5.1. Trabalhos futuros.....	49
<b>6. REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>50</b>

## 1. INTRODUÇÃO

Em uma economia global, que inclui empresas cada vez mais competitivas, onde eficiência e inovação são indispensáveis para o sucesso, os sistemas baseados em computadores tornaram-se um pilar insubstituível, onde uma vez já foi apenas um diferencial, agora mostra-se um requisito mínimo.

Para que empresas consigam perdurar em um ambiente tão competitivo como é a indústria atual, a adoção e eficiente utilização de *softwares* é indispensável. Esses programas têm a capacidade de aumentar a produtividade, facilitar a tomada de decisão baseada em dados, aumentar a segurança do processo, confiabilidade, organização, eficiência e muitos outros atributos. (ProEst Autodesk, 2021).

Baseado na essencialidade do *software* na empresa atual, surge também a necessidade daqueles que conseguem construir, implementar, manter e melhorar tais programas. Conforme os *softwares* tornam-se cada vez mais específicos, ao ponto que as empresas atualmente possuem diversas dessas aplicações não só externas como internas também, é imensurável a importância dos programadores neste processo.

Dadas as incontáveis diferenças que existem entre esses programadores, pessoas que vivem no mundo inteiro, com educações, princípios, visões, pensamentos e vidas diferentes, algumas convenções começaram a surgir para buscar padronizar a construção desses *softwares*, sendo uma dessas metodologias a SOLID, recebendo este nome como um acrônimo de seus cinco princípios fundamentais aprofundados no capítulo seguinte.

Trazer tais *softwares* para áreas fabris como a de estimativa de custos mostra-se como um desafio, uma vez que boa parte dessas áreas executam suas atividades bem antes de *softwares* estarem disponíveis de forma comercial, com datas anteriores a 1980 (Thunderbolt Innovation, 2019).

Esse desafio, quando superado, costuma gerar notáveis índices positivos para as empresas (ProEst Autodesk, 2021), evidenciando a importância do processo completo da construção de *software*, desde o planejamento em conjunto com os clientes que

encomendaram o programa até a manutenção do resultado, mantendo-o atualizado conforme as necessidades que surgirem.

## 1.1 Objetivos

Este trabalho busca implementar uma aplicação web de gerenciamento de estimativa de custos, cobrindo desde o ponto de cadastro pela parte do estimador, passando pela análise dessa estimativa, revisão e ação do supervisor e organização e visualização posterior das estimativas existentes no sistema, seguindo os princípios da metodologia SOLID para garantir um *software* eficiente e de fácil manutenção.

## 1.2. Justificativa

A motivação por trás da escolha do tema “HUB DE ESTIMATIVAS: Uma aplicação web para otimização do processo de estimativa de custos” surgiu da experiência profissional que o autor, um engenheiro de computação em formação, teve dentro da área fabril de estimativa de custos, onde a principal tecnologia utilizada para o controle do setor inteiro era uma planilha.

Uma vez observado o potencial que um *software*, mesmo transacional, teria para contribuir com as variadas dificuldades apresentadas por todos os níveis da área, desde estagiários até membros da supervisão, floresceu a ambição de desenvolver tal sistema.

O processo de desenvolvimento apresentou-se como uma ótima oportunidade para entender por completo como funciona esse processo na prática, incluindo não só as tarefas mais óbvias como a programação de fato como também a elaboração de protótipos e a colheita de requisitos de aceite para que o programa tenha todas as funcionalidades esperadas.

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1. Estimativa de Custos

A estimativa de custos é o processo de coletar e analisar dados históricos e aplicar modelos quantitativos, técnicas, ferramentas e bancos de dados para prever uma estimativa do custo futuro de um item, produto, programa ou tarefa. A estimativa de custo é a aplicação da arte e da tecnologia de aproximar o valor provável (ou custo), extensão ou caráter de algo com base nas informações disponíveis no tempo (MISLICK; NUSSBAUM, p.11).

Os estimadores profissionais utilizam técnicas definidas para criar estimativas de custos que são utilizadas para avaliar a viabilidade financeira, orçar custos e monitorar os gastos de projetos. Uma estimativa precisa de custos é fundamental para decidir assumir um projeto, para determinar o escopo eventual de um projeto e para garantir que os projetos permaneçam financeiramente viáveis e evitem excessos de custos (EBY, 2017).

A estimativa de custo prepara o terreno para um orçamento que fornecerá o financiamento adequado para concluir um projeto no prazo e de acordo com os padrões de qualidade. Dar um passo extra e atribuir orçamentos para várias atividades ajuda a equipe a ficar dentro do orçamento maior e evitar gastos desnecessários. Também ajuda a liderança do projeto a entender onde podem ocorrer excedentes para que possam mover recursos conforme necessário para cobrir os custos (MONDAY, 2022).

A estimativa de custos é um processo vital que envolve prever os requisitos financeiros de um projeto ou atividade. Ele fornece insights valiosos que ajudam você a tomar decisões informadas, definir orçamentos, alocar recursos com eficiência, gerenciar riscos e obter uma vantagem competitiva. Usando a estimativa de custos, as organizações podem garantir o sucesso de seus projetos, entregá-los no prazo e dentro do orçamento e manter sua reputação no setor (IntelliPaat, 2023).

## 2.2 Aplicação Web

Uma aplicação web é um *software* executado em um navegador da Internet. Qualquer site que contenha elementos interativos pode ser chamado de aplicação web. Isso significa que o usuário pode interagir com a plataforma pressionando botões, preenchendo formulários, solicitando preços ou realizando compras. Semelhante ao *software* de computador desktop ou a um aplicativo móvel, um aplicativo da web fornece uma interface de usuário, oferece utilidade ou entretenimento e a capacidade de acessar, criar, armazenar ou modificar dados (RUDNIK, 2020).

Como definido por William (2022), uma arquitetura de aplicação web normalmente é composta por três componentes principais:

- *Web Browser*: O navegador ou o componente front-end é o componente chave que interage com o usuário, recebe a entrada e gerencia a lógica de apresentação enquanto controla as interações do usuário com o aplicativo. As entradas do usuário também são validadas, se necessário.
- *Servidor da Web*: O servidor da Web, também conhecido como componente de back-end ou componente do lado do servidor, lida com a lógica de negócios e processa as solicitações do usuário, roteando as solicitações para o componente certo e gerenciando todas as operações do aplicativo. Ele pode executar e supervisionar solicitações de uma ampla variedade de clientes.
- *Servidor de banco de dados*: O servidor de banco de dados fornece os dados necessários para o aplicativo. Ele lida com tarefas relacionadas a dados. Em uma arquitetura multicamadas, os servidores de banco de dados podem gerenciar a lógica de negócios com a ajuda de procedimentos armazenados.

De modo geral, aplicações web fazem parte do dia a dia de consumidores, empresas e profissionais de vários ramos diferentes, fazendo com que o desenvolvimento, implementação e controle de qualidade desses sistemas acabe ganhando maior destaque no mercado digital (XP EDUCAÇÃO, 2022).

## 2.3 Metodologia SOLID

Metodologia SOLID é um conjunto popular de princípios de design usados no desenvolvimento de *software* orientado a objetos.

SOLID é um acrônimo que representa cinco princípios-chave de design, todos os cinco são comumente usados por engenheiros de *software* e fornecem alguns benefícios importantes para os desenvolvedores (WATTS, 2020).

Como teorizado por Martin (2000) e posteriormente aprofundado e consolidado no acrônimo SOLID por Feathers (2004), é possível detalhar os princípios como:

- *Princípio da Responsabilidade Única*: Uma classe deve ter, obrigatoriamente, apenas um único motivo para mudar.
- *Princípio Aberto-Fechado*: As classes podem estar abertas para extensão, entretanto devem estar fechadas para modificação.
- *Princípio da Substituição de Liskov*: As classes derivadas devem ser completamente substituíveis por suas classes bases sem efeito na integridade do sistema.
- *Princípio da Segregação de Interfaces*: Uma classe não deve implementar interfaces ou métodos que não serão usados por ela.
- *Princípio da Inversão de Dependência*: Os módulos de alto e baixo níveis devem depender apenas de abstrações. Essas abstrações não devem depender de detalhes, mas sim o oposto, detalhes devem depender de abstrações.



## 2.4 Tecnologias Utilizadas

### 2.4.1 Linguagens

Nesta seção serão apresentadas quais as três linguagens utilizadas no desenvolvimento dessa aplicação, sendo duas delas linguagens de programação, o Typescript e Java, e a outra uma linguagem de consulta estruturada, o SQL.

#### . *Typescript*

Typescript é uma linguagem de programação de código aberto desenvolvida pela Microsoft, tratando-se de um superconjunto da linguagem Javascript. Em outras palavras, é uma linguagem que possui tudo o que o Javascript possui e algumas outras funcionalidades a mais, sendo a mais notável a tipagem estática, isto é, a definição dos tipos de dados em tempo de compilação, proporcionando assim maior coerência e um código mais bem definido e de fácil manutenção (Duarte, 2020).

Por tais características, foi escolhida a linguagem Typescript para compor o front-end da aplicação, isto é, a parte visual do programa que fica em contato direto com o usuário por meio de seu navegador. Para facilitar a construção da aplicação e a união do Typescript com os arquivos HTML e CSS foi utilizado o framework Angular.

Angular é definido pela sua própria documentação oficial<sup>1</sup> como uma “plataforma de desenvolvimento para criar aplicativos de página única eficientes e sofisticados”. Sua utilização em conjunto com arquivos HTML e CSS vão compor a parte visual da aplicação, conhecida como front-end. Nela ficará toda a lógica das múltiplas páginas do *software*, estilização e scripts que farão requisições para o back-end.

---

<sup>1</sup> Disponível em: <<https://angular.io/docs>>. Acesso em: 12 de julho de 2023.

## . Java

Java, segundo sua documentação oficial<sup>2</sup>, é uma linguagem de programação de alto nível e orientada a objetos utilizada amplamente em desenvolvimento de *software*. Criada pela Sun Microsystems em 1995 e posteriormente adquirida junto com a empresa pela Oracle Corporation em 2010, é conhecida pelo seu amplo uso, segurança e robustez.

Por tais características positivas, a linguagem de programação Java foi escolhida para compor o back-end da aplicação, que é a sessão responsável por fazer o papel de intermediador entre o que é requisitado pelo front-end e o que é retornado direto do banco de dados da aplicação. Buscando estruturar o projeto de uma forma prática e de fácil manutenção, foi utilizado o framework Spring Boot.

Spring Boot, como definido em seu site oficial<sup>3</sup>, é um framework de desenvolvimento de aplicativos Java que simplifica a criação de aplicativos baseados em Spring, possuindo um servidor web embutido que facilita a configuração e execução de aplicações.

Adicionalmente, serão utilizadas algumas bibliotecas externas para complementar o desenvolvimento utilizando o Spring Boot:

- *MySQL Driver*: Permite a comunicação com um banco de dados MySQL;
- *Lombok*: Automatiza a criação de código repetitivo utilizando anotações;
- *Spring Web*: Fornece suporte para mapeamento de URL e tratamento de requisições HTTP;
- *H2 Database*: Fornece um banco de dados embutido, facilitando testes do código;
- *Spring Data JPA*: Automatiza a criação de código SQL repetitivo;

---

<sup>2</sup> Disponível em: <[https://www.java.com/en/download/help/whatis\\_java.html](https://www.java.com/en/download/help/whatis_java.html)>. Acesso em: 12 de julho de 2023.

<sup>3</sup> Disponível em: <<https://spring.io/projects/spring-boot#overview>>. Acesso em: 12 de julho de 2023.

## . SQL

A Linguagem de consulta estruturada, ou *Structured Query Language* em inglês, é uma linguagem utilizada para gerenciar dados relacionais dentro de um banco de dados e fluxos de dados dentro de um sistema de gerenciamento de dados.

Como definido no site oficial dos serviços web da Amazon, a AWS, é possível “usar instruções SQL para armazenar, atualizar, remover, pesquisar e recuperar informações do banco de dados. Também pode usar SQL para manter e otimizar a performance do banco de dados.”<sup>4</sup>

Para esse projeto, a linguagem SQL será usada como interface por um sistema de gerenciamento de banco de dados chamado MySQL.

MySQL é um sistema de gerenciamento de banco de dados (SGBD) que utiliza a linguagem SQL. Ele servirá como uma espécie de arquivo onde serão guardadas as informações exibidas pela aplicação web: Os estimadores, estimativas e produtos.

### 2.4.2 Ferramentas de desenvolvimento

Para auxiliar o desenvolvimento dessa aplicação foram utilizados alguns *softwares*, sendo dois deles IDEs: O Visual Studio Code e o IntelliJ, utilizados para escrita, refatoração e *debugging* do código, Insomnia para testagem da API com requisições e o MySQL Workbench para fornecer um serviço de banco de dados.

---

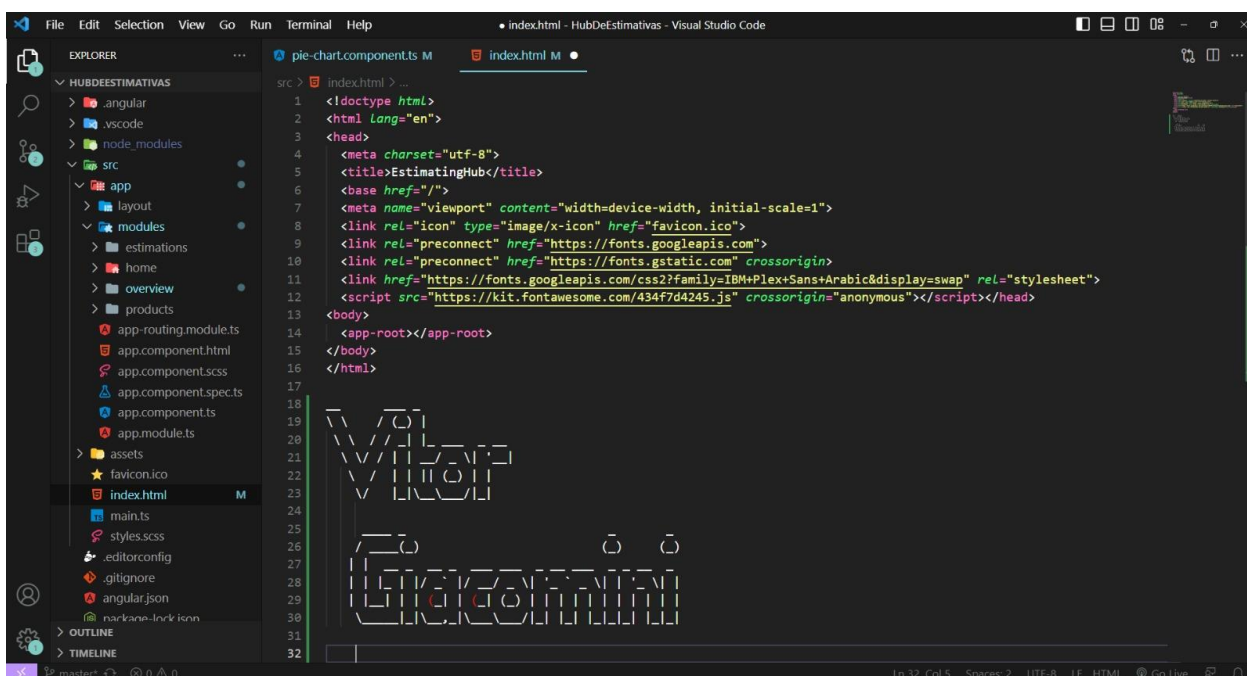
<sup>4</sup> Disponível em: <<https://aws.amazon.com/pt/what-is/sql/>>. Acesso em: 13 de julho de 2023.

## . Visual Studio Code

O Visual Studio Code, mais conhecido como VSCode, é definido pela sua documentação oficial<sup>5</sup> como um editor de código fonte desenvolvido pela empresa Microsoft, suportando diversas linguagens de programação, além de fornecer recursos como destaque de sintaxe, *snippets*, refatoração e *debugging* integrados, além do seu extenso suporte a diferentes extensões.

Um exemplo da interface deste editor pode ser visto na figura abaixo:

Figura 01: Interface do Visual Studio Code



Fonte: Elaborado pelo autor, 2023

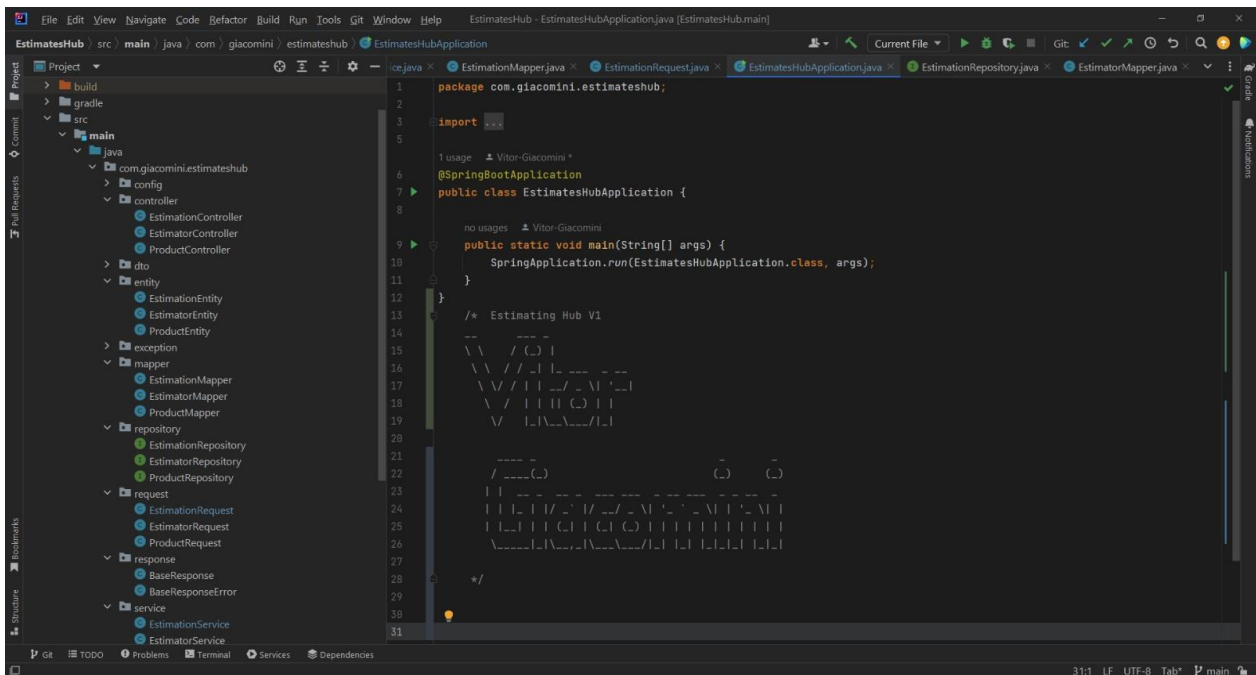
<sup>5</sup> Disponível em: <<https://code.visualstudio.com/docs>>. Acesso em: 13 de julho de 2023.

## . IntelliJ IDEA

O IntelliJ é, segundo sua documentação oficial<sup>6</sup>, uma IDE criada pela JetBrains, conhecida por inteligência e eficiência, destacando-se por ser um editor avançado que possui uma profunda compreensão do código, fornecendo assim uma navegação rápida e experiências eficientes ao usuário.

Um exemplo da interface desta IDE pode ser visto na figura abaixo:

Figura 02: Interface do IntelliJ IDEA



Fonte: Elaborado pelo autor, 2023

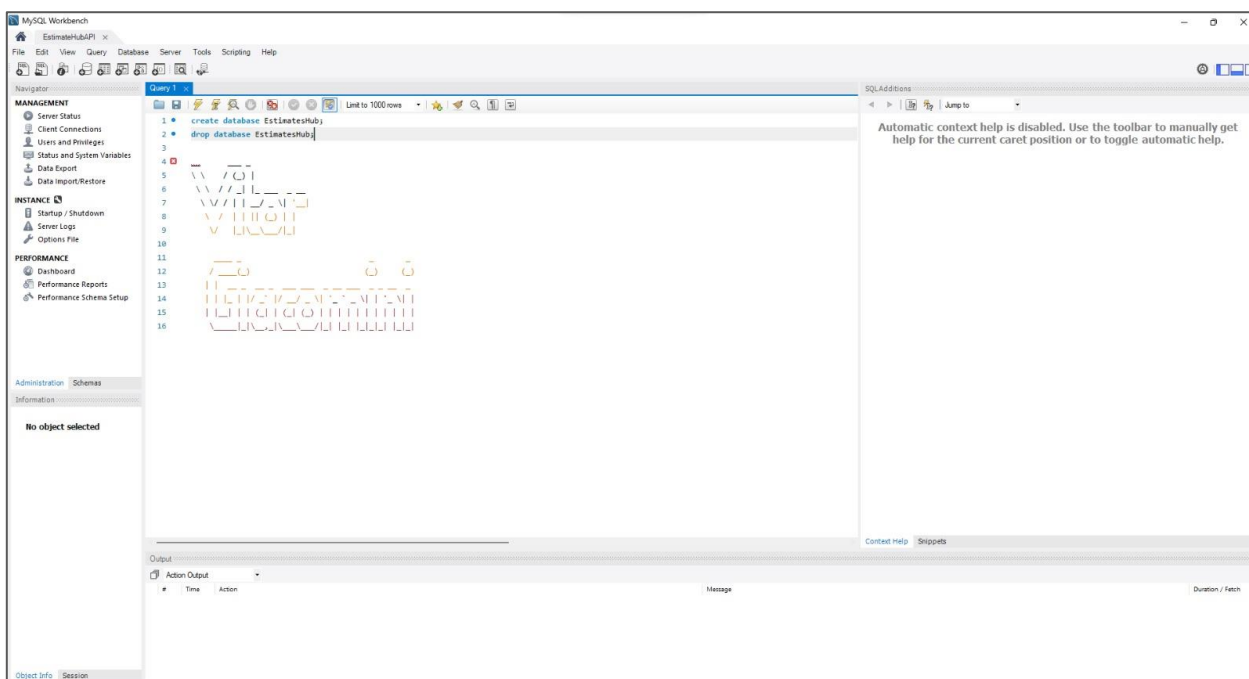
<sup>6</sup> Disponível em: <<https://www.jetbrains.com/pt-br/idea/>>. Acesso em: 14 de julho de 2023.

## . MySQL Workbench

MySQL Workbench é, como definido em seu site oficial<sup>7</sup>, uma ferramenta visual unificada para arquitetos de banco de dados, desenvolvedores e DBAs, fornecendo modelagem de dados, desenvolvimento de SQL e ferramentas de administração abrangentes para configuração de servidor, administração de usuários, backup e muito mais, fornecendo recursos robustos como suporte a migrações e dashboards.

Um exemplo da interface desta ferramenta pode ser visto na figura abaixo:

Figura 03: Interface do MySQL Workbench



Fonte: Elaborado pelo autor, 2023

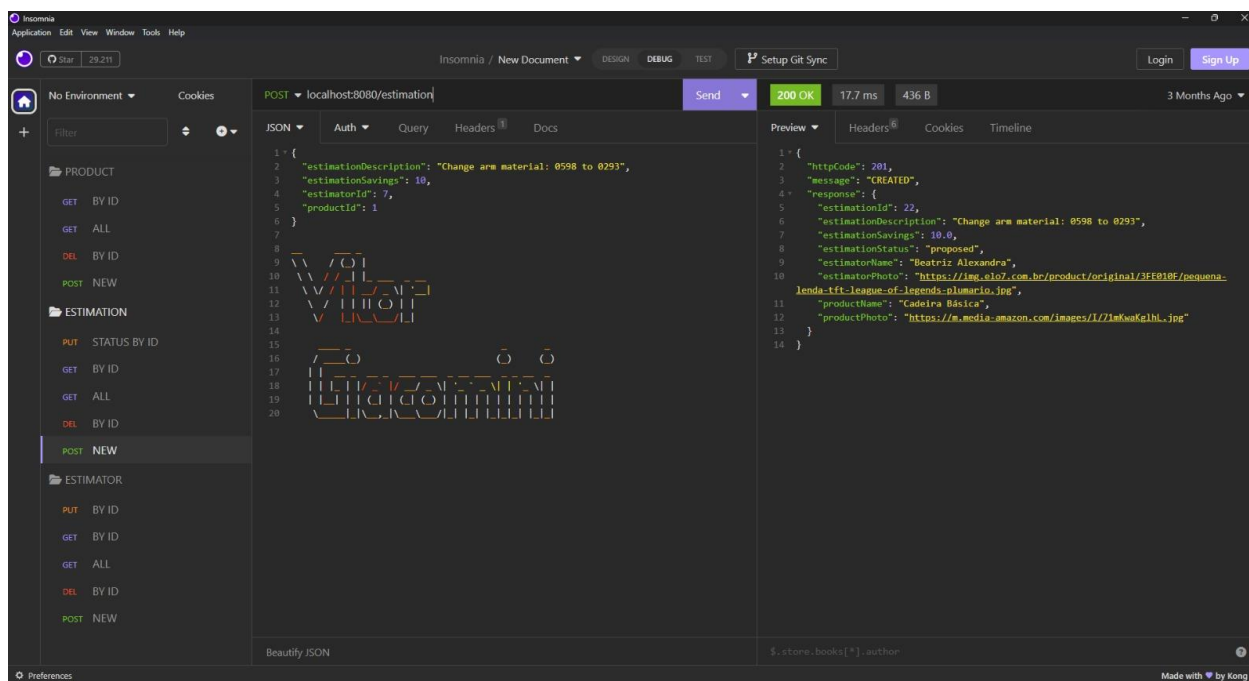
<sup>7</sup> Disponível em: <<https://www.mysql.com/products/workbench/>>. Acesso em: 15 de julho de 2023.

## . Insomnia

Insomnia, de acordo com sua documentação oficial<sup>8</sup>, é uma ferramenta de testes de API que permite aos desenvolvedores criar, gerenciar, documentar e testar chamadas para variados tipos de APIs por meio de uma interface gráfica intuitiva que facilita a construção de chamadas HTTP e a visualização da resposta da API.

Um exemplo da interface desta ferramenta pode ser visto na figura abaixo:

Figura 04: Interface do Insomnia



Fonte: Elaborado pelo autor, 2023

<sup>8</sup> Disponível em: <<https://docs.insomnia.rest/>>. Acesso em: 15 de julho de 2023.

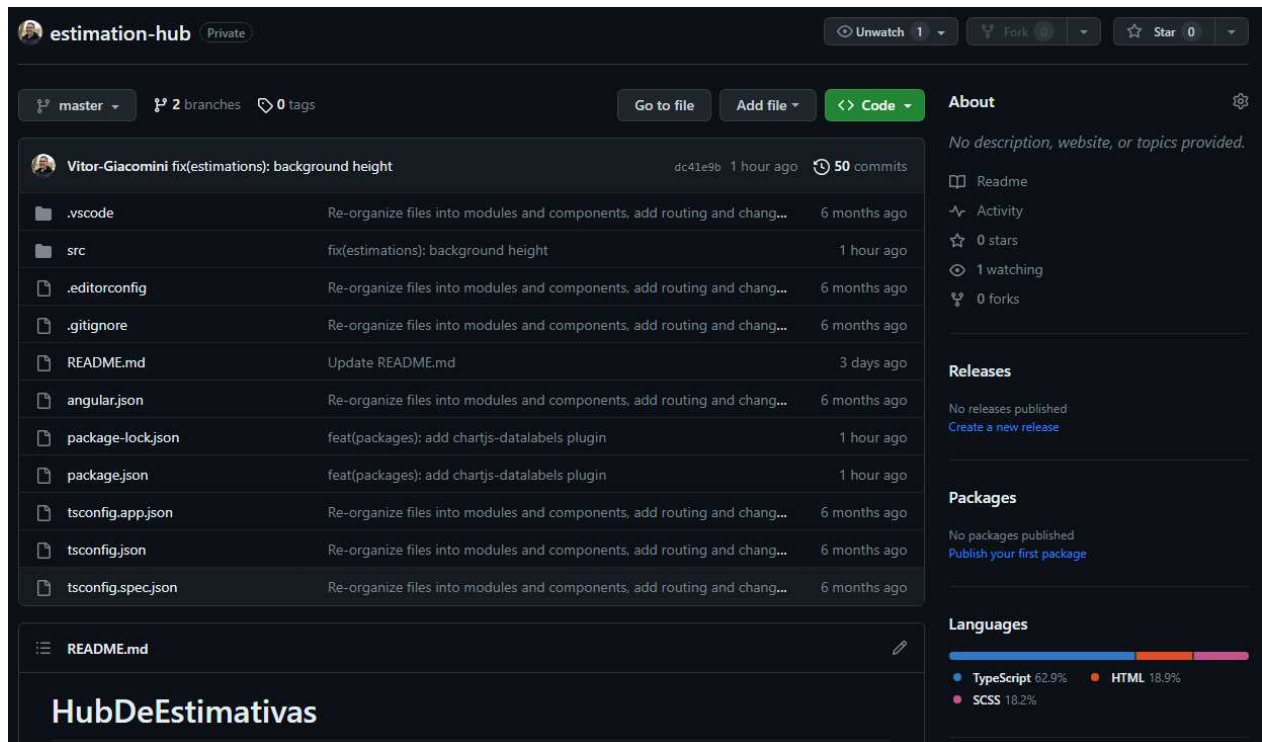
## . Github

Com o objetivo de facilitar o controle das versões do código, documentar as mudanças feitas nas aplicações e manter um backup seguro do que estava sendo desenvolvido, optou-se pelo uso do Github.

Github é, segundo apresentado em sua documentação oficial<sup>9</sup>, uma plataforma de hospedagem de código fonte com controle de versões. O programa permite com que usuários possam criar e gerenciar repositórios de código, possibilitando desde um controle simples de versões até colaboração simultânea com outros programadores.

A visão do repositório que compõe o front-end pode ser vista na imagem abaixo:

Figura 05: Interface do GitHub no projeto estimation-hub (front-end)



Fonte: Elaborado pelo autor, 2023

<sup>9</sup> Disponível em: <<https://docs.github.com/en/>>. Acesso em: 16 de julho de 2023.



### 3. DESENVOLVIMENTO

#### 3.1 Definição dos requisitos

Devido ao fato de a área de estimativa de custos ter a possibilidade de executar ou não uma grande quantidade de tarefas distintas, dependendo do ramo ou natureza de negócio da empresa, definiu-se antes do início do desenvolvimento do *software* qual seria o contexto em que o trabalho seria inserido: Uma empresa responsável pela confecção de móveis.

Nesta empresa, a área de estimativa de custos é responsável por analisar os produtos já existentes e buscar neles oportunidades de economia sem que o cliente seja afetado negativamente. Para fins de simplificação, não serão incluídos nos cálculos os fatores de mão de obra, maquinário, fornecedores e lucro, concentrando-se apenas no custo agregado de todos esses fatores em forma do preço final do produto.

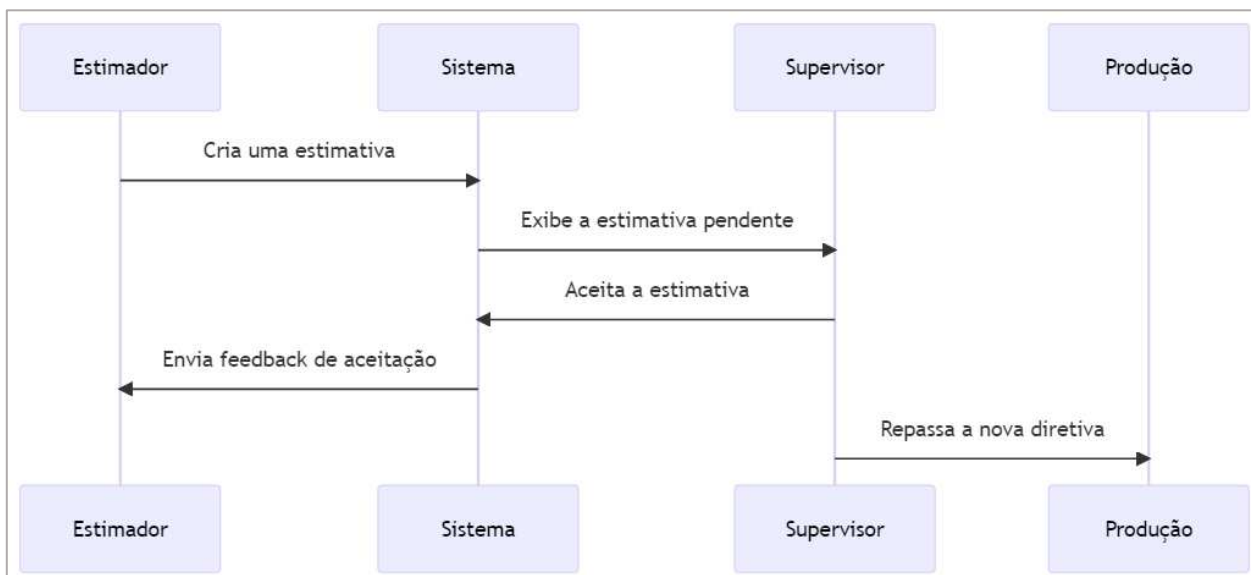
Dentro desta área, há a necessidade de uma plataforma unificada em que os estimadores possam inserir suas estimativas e acompanhar o estado delas em tempo real. Nesta mesma plataforma, os supervisores devem conseguir visualizar todas as estimativas, além de mudar o estado delas dependendo do seu veredito, aceitando-as ou não.

Como ponto adicional a um sistema de gerenciamento convencional, esse programa também deve fornecer uma forma simples de ter uma visualização geral sobre todas as estimativas existentes nele, local onde pode ser analisado de forma objetiva o quanto a área contribui em economias para empresa, bem como outros dados relevantes.

Para ilustrar o funcionamento esperado do *software* e seu papel dentro do fluxo padrão que acontece dentro da área de estimativa de custos, foi desenvolvido um diagrama de sequência por meio da linguagem Mermaid.

Este diagrama pode ser visualizado na figura em sequência:

Figura 06: Fluxo padrão na área de estimativa de custos



Fonte: Elaborado pelo autor, 2023

### 3.2 Protótipo

Após a definição de quais os requisitos existirão no *software*, foi criado um protótipo de alto nível para alinhar entre programador e idealizador da ferramenta como seria a aparência do programa.

O protótipo foi feito utilizando a ferramenta Canva, uma plataforma online de design gráfico. Por se tratar de um protótipo de alto nível, o *layout* está sujeito a sofrer pequenas alterações no produto.

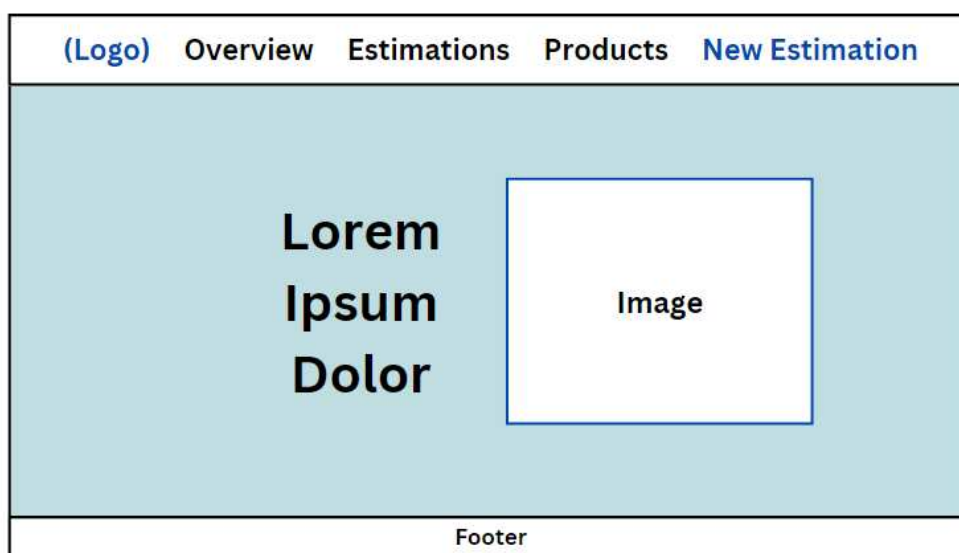
Com a construção do protótipo, buscou-se minimizar a quantidade de páginas desnecessárias e informações redundantes no sistema, melhorando a experiência do usuário e ao mesmo tempo deixando a aplicação mais leve e veloz.

Este protótipo contou com os seguintes elementos:

- *Página Inicial*: Popularmente conhecida como “Home Page”, será a primeira página a ser exibida pela aplicação, e o caminho padrão para o qual o usuário será redirecionado ao não colocar nenhum parâmetro depois do endereço da aplicação na barra de navegação.

Com uma representação visual em acordo com a seguinte figura:

Figura 07: Protótipo da página inicial

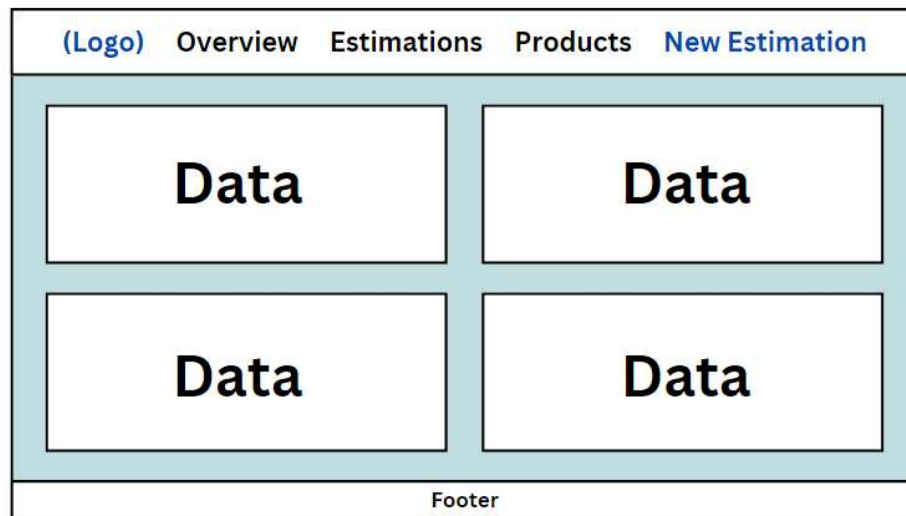


Fonte: Elaborado pelo autor, 2023

- *Visão Geral*: Definida na aplicação como “Overview”, trata-se de uma página feita para mostrar, em gráficos, alguns dados de rápido acesso sobre a área de estimativa de custos, levando em conta as informações existentes no banco de dados. O objetivo dessa página é possibilitar uma forma fácil e clara dos estimadores mostrarem o seu impacto dentro da empresa.

Com uma representação visual em acordo com a seguinte figura:

Figura 08: Protótipo da página de visão geral



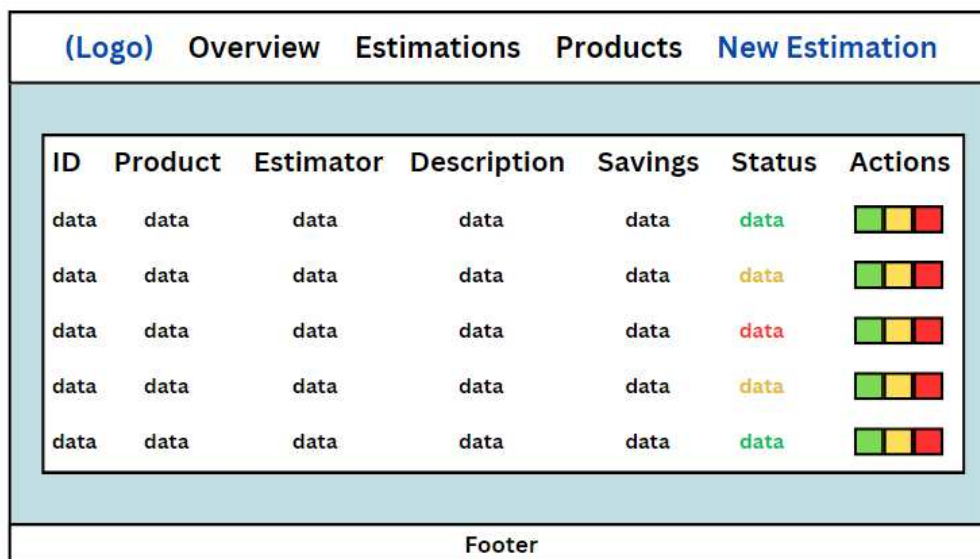
Fonte: Elaborado pelo autor, 2023

- *Estimativas*: Em inglês, “*Estimations*”, trata-se de uma página composta principalmente por uma lista com as estimativas existentes no sistema, incluindo o produto ao qual se referem e outras informações como o estado da estimativa e

o estimador. Essa página busca possibilitar o acesso fácil dos administradores as estimativas feitas pelos funcionários da área, permitindo fazer o fácil gerenciamento destas.

Com uma representação visual em acordo com a seguinte figura:

Figura 09: Protótipo da página de estimativas



O protótipo da página de estimativas apresenta uma interface com um menu de navegação no topo contendo: (Logo), Overview, Estimations, Products e New Estimation. Abaixo do menu, há uma tabela com sete colunas: ID, Product, Estimator, Description, Savings, Status e Actions. A tabela contém cinco linhas de dados, cada uma com um ícone de status composto por três quadrados (verde, amarelo e vermelho). Na base da interface, há uma seção rotulada como Footer.

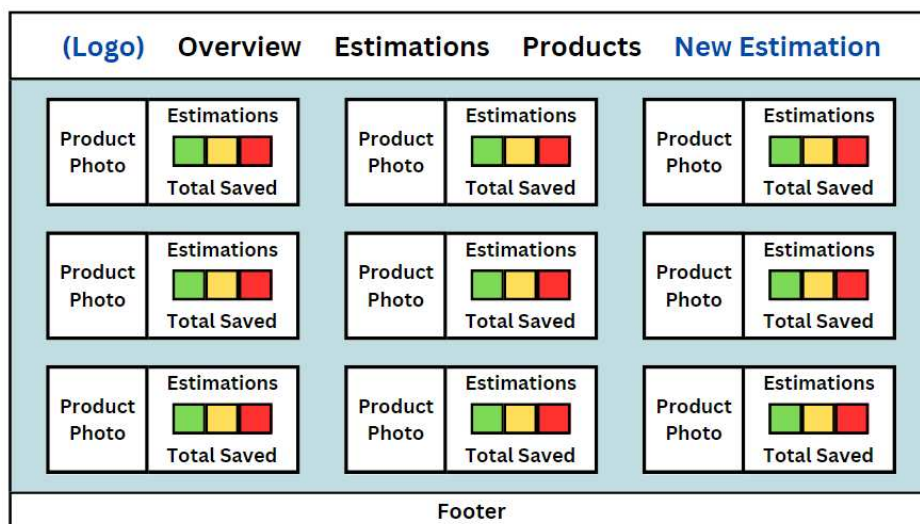
ID	Product	Estimator	Description	Savings	Status	Actions
data	data	data	data	data	data	
data	data	data	data	data	data	
data	data	data	data	data	data	
data	data	data	data	data	data	
data	data	data	data	data	data	

Fonte: Elaborado pelo autor, 2023

- *Produtos*: A aba de produtos, definida como “*Products*” na aplicação, será uma página em que estarão organizados todos os produtos da empresa, tendo eles recebido estimativas ou não. Busca-se com a inclusão dessa página facilitar a visualização de quais produtos são mais eficientes desde seu lançamento na empresa, e, da mesma forma, quais são mais problemáticos.

Com uma representação visual em acordo com a seguinte figura:

Figura 10: Protótipo da página de produtos

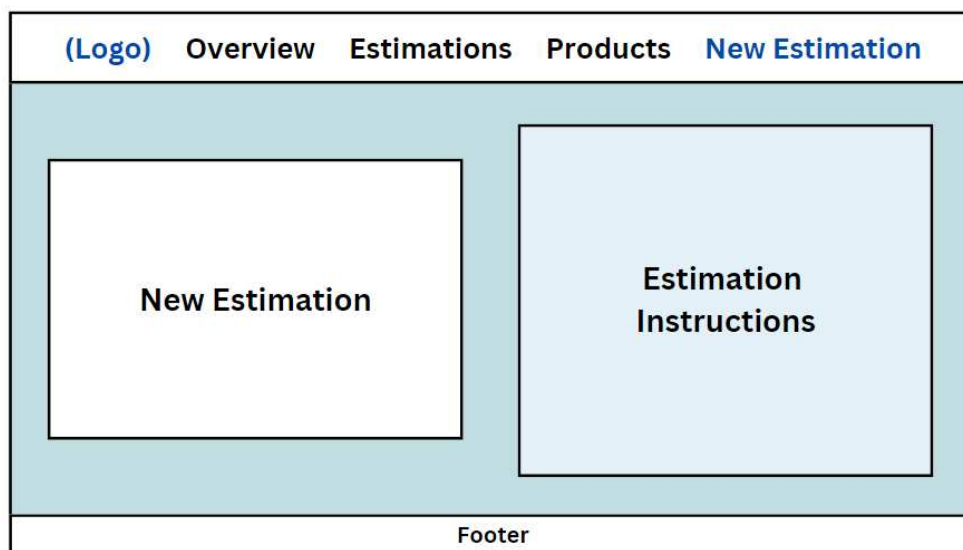


Fonte: Elaborado pelo autor, 2023

- *Criação de nova estimativa:* Tal página busca permitir de forma eficiente a criação de uma nova estimativa ao mesmo tempo que lembra ao estimador quais são os requisitos necessários para a formulação correta dela.

Com uma representação visual em acordo com a seguinte figura:

Figura 11: Protótipo da página de criação de nova estimativa



Fonte: Elaborado pelo autor, 2023

### 3.3. Definição do tipo do sistema

Uma vez definidos os requisitos e o protótipo, define-se o tipo de sistema que será construído pelo programador responsável, neste caso, um sistema transacional.

Segundo Barbosa (2012, p. 4), um sistema de informação transacional “auxilia o nível mais elementar (operacional), monitorando atividades diárias de uma empresa como vendas, receitas, fluxo de materiais”, e “executa e registra transações rotineiras necessárias para conduzir o negócio”, exatamente o que se busca atingir com o Hub de Estimativas.

## **4. RESULTADOS**

Este capítulo contém a apresentação da aplicação, sendo dividida em duas subseções: Front-end, que ilustra a visualização web que estará diretamente em contato com o usuário e back-end, contendo a estrutura escolhida para compor este bloco do *software*.

### **4.1. Front-end**

Sendo diretamente acessado pelo usuário, o front-end desenvolvido será explicado nessa subseção, tendo primeiro sua estrutura apresentada e, posteriormente, como ficaram as páginas que compõem a aplicação.

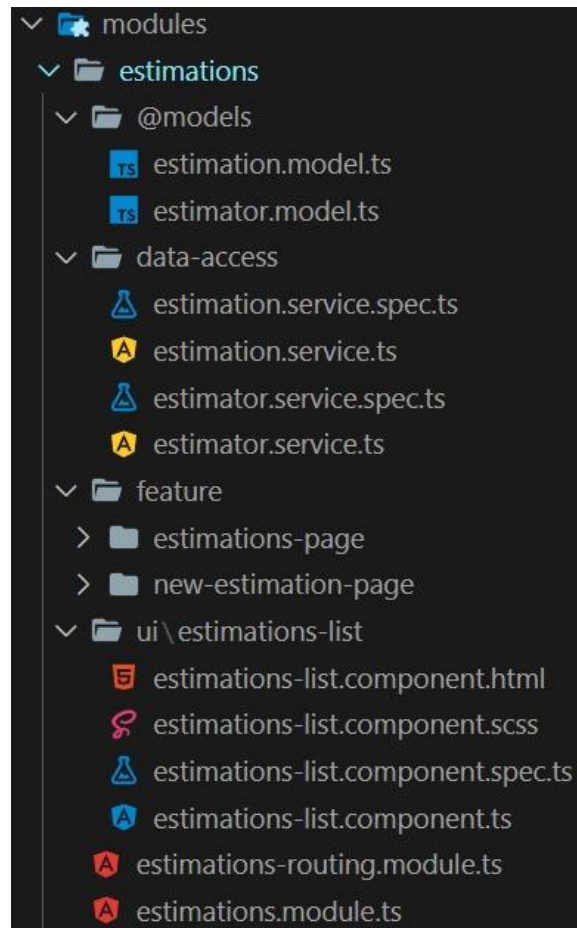
#### **4.1.1. Estrutura**

A aplicação segue uma estrutura modular, em que cada uma das entidades principais contidas no programa tem seu próprio pacote, contendo seus respectivos modelos, serviços e componentes, sejam eles lógicos ou apenas ilustrativos.

Um exemplo pode ser visto na imagem da Figura 12 a seguir, envolvendo o pacote “Estimativas”:



Figura 12: Módulo de Estimativas



Fonte: Elaborado pelo autor, 2023

Cada um desses pacotes possui uma função específica dentro do projeto:

- **@Models:** Contém as entidades diretamente ligadas ao pacote em que estão localizadas. No caso, essas entidades são Estimativa e Estimador.
- **Data-access:** Apresenta um ou mais serviços relacionados a entidade principal daquele pacote. No exemplo, os serviços de Estimativa e Estimador. Nestes arquivos estão localizados os métodos que enviarão requisições HTTP para o back-end, recebendo de volta informações como uma lista de todos os estimadores, uma confirmação de que a estimativa foi atualizada, dentre outros.

- **Feature:** Contém os chamados “componentes inteligentes” da aplicação, isto é, os componentes que possuem dentro deles uma lógica que pode afetar o resto da aplicação ou os dados existentes na API. No exemplo, existem duas *features* relacionadas ao módulo de estimativas: A página de visualização de todas as estimativas e a página de criação de uma nova estimativa.
- **UI:** Apresenta os chamados “componentes burros” da aplicação, isto é, os componentes que possuem nenhuma ou uma quantidade mínima de lógica dentro deles, funcionando na maioria das vezes apenas como funcionalidades ilustrativas. Neste caso, o componente Estimations-list funciona apenas para receber uma lista de estimativas e exibi-la para o usuário.

Ainda dentro do pacote de estimativas, é possível ver também dois arquivos que não estão localizados em nenhuma subpasta. São os módulos, sendo o primeiro, o *routing module*, responsável por definir a rota no navegador que levará para esse módulo específico e o segundo, apenas *estimations module*, sendo utilizado para declaração de componentes utilizados dentro do pacote em que este está localizado, no caso, *estimations*.

#### 4.1.2. Página Inicial

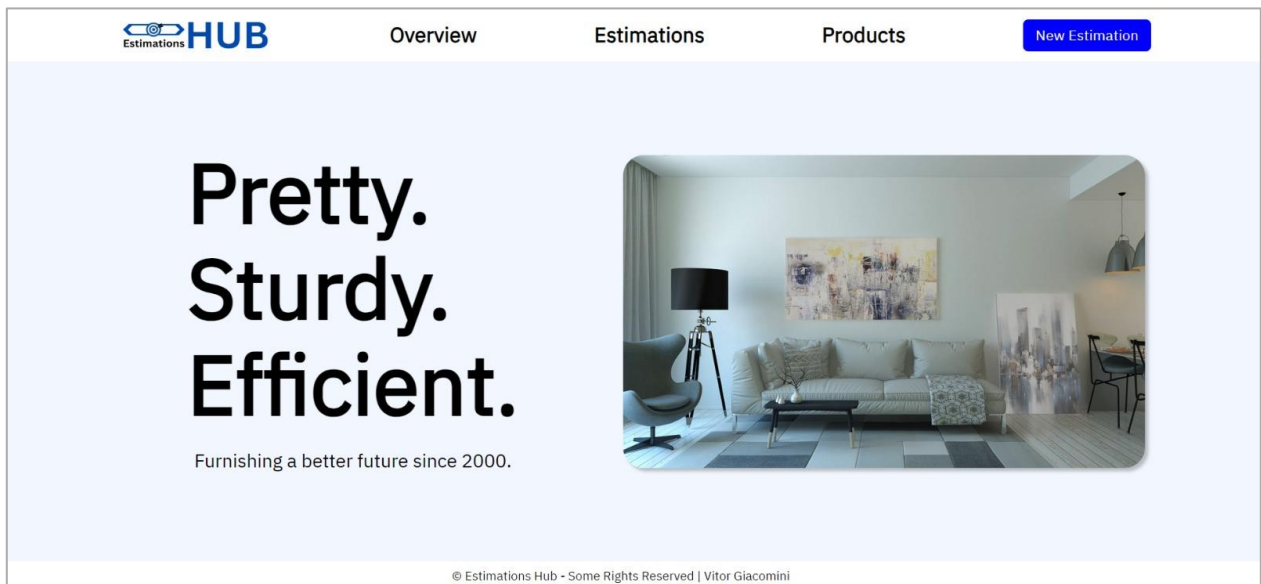
A página inicial da aplicação pode ser dissecada em três sessões:

- **Header:** Barra horizontal localizada no topo da tela onde é possível encontrar textos que, ao serem clicados, levam o usuário para diferentes páginas dentro da aplicação. É um componente presente em todas as páginas do programa.

- **Conteúdo:** Apresentando uma frase escolhida como lema da empresa fictícia, um subtítulo e uma imagem de uma sala de estar, é um elemento estático cujo objetivo é recepcionar de forma agradável o usuário.
- **Footer:** Barra horizontal localizada no final da tela onde é possível encontrar um pequeno texto com o nome da aplicação, direitos e nome do autor. Bem como o header, esse componente também está presente em todas as páginas.

A aparência final da página inicial pode ser vista na figura abaixo:

Figura 13: Página inicial



Fonte: Elaborado pelo autor, 2023

#### 4.1.3. Visão Geral

A página de visão geral apresenta informações de fácil entendimento sobre o desempenho da área de estimativa de custos, como requisitado durante a criação do protótipo. Ela é dividida em duas sessões:

- **Cartões:** Essa sessão contém três informações de maior relevância, sendo elas o total monetário poupado pela área, a quantidade de estimativas aprovadas e a maior estimativa aprovada desde o presente momento.
- **Gráfico:** Essa sessão apresenta um gráfico que mostra de forma simples um resumo sobre os resultados da área. O espaço ocupado por cada uma das sessões do gráfico está diretamente relacionado a quantidade de estimativas aprovadas, e o texto nele presente exhibe a soma de suas economias.

A aparência final da página de visão geral pode ser vista na figura abaixo:

Figura 14: Página de visão geral



Fonte: Elaborado pelo autor, 2023

#### 4.1.4. Estimativas

A página de estimativas apresenta uma lista com todas as estimativas presentes no banco de dados do sistema. Possui sete colunas contendo, respectivamente: ID, Nome do Produto, Estimador, Descrição, Economia, Status e Ações. Enquanto as seis

primeiras colunas são apenas de visualização, sendo possível usar os campos de pesquisa para filtrar os resultados, a última coluna é interativa, permitindo com que o usuário possa alterar o estado das estimativas no sistema.

As estimativas são exibidas por ordem de data de criação, permitindo com que estimativas recém-criadas apareçam no topo, tornando mais fácil para o administrador notar quando há uma adição ao sistema e, posteriormente, atribuir a ela um estado.

A aparência final da página de estimativas pode ser vista na figura abaixo:

Figura 15: Página de estimativas

ID	Product	Estimator	Description	Savings	Status	Actions
33	Bedside Table	Thomas Maroni	Change top material: S349 to S301	\$3	Proposed	✓ ⌂ ✕
32	Wood Table	Evair Esmael	Change leg material: W803 to W837	\$5	Proposed	✓ ⌂ ✕
31	Wood Stool	Antônidas Oslo	Change seat material: W240 to W040	\$1	Proposed	✓ ⌂ ✕
30	Royalty Chair	Josinelson Rodrigo	Change arm material: H423 to H410	\$3	Accepted	✓ ⌂ ✕
29	Vintage Stool	Luiz Hernando	Change leg material: L102 to L148	\$6	Accepted	✓ ⌂ ✕
28	Wood Stool	Wilmar Guimarães	Change leg material: W494 to W499	\$5	Accepted	✓ ⌂ ✕
27	Royalty Chair	Josinelson Rodrigo	Change cloth material: C438 to R499	\$3	Accepted	✓ ⌂ ✕
26	Saloon Stool	Edward Enary	Change top material: R390 to R399	\$1	Accepted	✓ ⌂ ✕
25	Bedside Table	Wilmar Guimarães	Change top material: P383 to P398	\$3	Rejected	✓ ⌂ ✕
24	Wood Stool	Evair Esmael	Change seat material: W844 to W843	\$5	Rejected	✓ ⌂ ✕

Fonte: Elaborado pelo autor, 2023

#### 4.1.5. Produtos

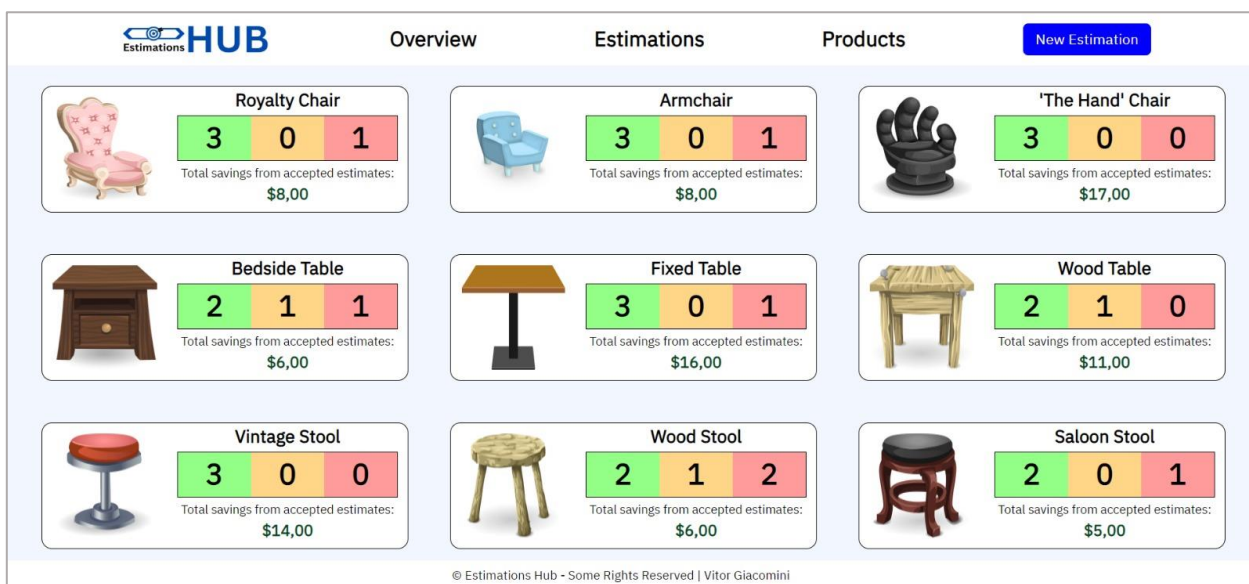
A página de produtos apresenta uma lista com os cartões de cada um dos produtos. Nestes cartões estão localizadas algumas informações básicas sobre o produto, como seu nome e uma imagem para agilizar o reconhecimento dele.

Adicionalmente, há uma fita contendo três valores que exibem a quantidade de estimativas cujo produto é o aquele do cartão: Estimativas aprovadas, pendentes e negadas, respectivamente. Para facilitar a compreensão, essa fita utiliza para cada estado de estimativa o mesmo esquema de cores da página de lista de estimativas.

Por fim, há um número em verde que, como dito pelo texto acima dele, exibe a soma das economias de todas as estimativas aprovadas daquele respectivo produto. No terceiro cartão da primeira linha, por exemplo, é possível ver que existem duas estimativas aprovadas para o móvel “Assento de Mão”, e a soma das economias dessas duas estimativas são de \$16.

A aparência final da página de produtos pode ser vista na figura abaixo:

Figura 16: Página de produtos



Fonte: Elaborado pelo autor, 2023

#### 4.1.6. Criação de estimativa

A página de criação de nova estimativa pode ser dividida em duas sessões:

- **Cartão de nova estimativa:** Contendo um formulário de quatro campos, esse cartão requisita algumas informações para que o estimador possa criar uma estimativa, sendo elas a descrição, o nome do estimador, o produto e a economia dessa estimativa. Ao colocar valores válidos em todos os campos, o botão “Submit” (Enviar, em português), fica ativo, permitindo a criação da estimativa.
- **Instruções de estimativa:** Este cartão contém as instruções básicas para a criação de uma nova estimativa, buscando lembrar o estimador das boas práticas no momento de cadastro e de algumas outras informações sobre o estado da aplicação e o funcionamento do formulário.

A aparência final da página de criação de estimativas pode ser vista na figura abaixo:

Figura 17: Página de criação de estimativa

The screenshot shows the 'New Estimation' page in the Estimations Hub application. The page has a navigation bar with 'Overview', 'Estimations', 'Products', and a 'New Estimation' button. The main content area is divided into two sections: a 'New Estimation' form and 'Estimation Instructions'.

**New Estimation Form:**

- Description:** A text input field with the placeholder text 'Describe your estimation using the S.D.P.'
- Estimator:** A dropdown menu with the placeholder text 'Select an estimator'.
- Product:** A dropdown menu with the placeholder text 'Select a product'.
- Savings:** A text input field with a dollar sign icon and the value '0'.
- Submit:** A button labeled 'Submit'.

**Estimation Instructions:**

- **No fields can be left empty**  
If the savings are null, the value should be zero.
- **The estimation's description should follow the Standard Declaration Pattern**  
E. g.: Change (part) material: XXXX to XXXX.
- **Savings is always calculated on a per-unit basis**  
Same goes for the "Products" page.
- **The estimation's status will always start as "proposed"**  
The status can only be changed on the "Estimations" page.
- **Ensure that all the information is correct before submitting the estimation**  
Inaccurate or incomplete estimations will be promptly rejected.

© Estimations Hub - Some Rights Reserved | Vitor Giacomini

Fonte: Elaborado pelo autor, 2023

## 4.2. *Back-end*

O *back-end*, divisão da aplicação responsável por lidar com as requisições enviadas pelo *front-end* e seus efeitos no banco de dados, será apresentado nessa subseção.

### 4.2.1. *Estrutura*

Buscando construir uma aplicação alinhada com os padrões SOLID e convenções de desenvolvimento em *Spring Boot*, o *back-end* foi dividido em pacotes, cada um possuindo uma função específica:

- **Controller:** O pacote de controladores contém as classes que são responsáveis por gerenciar as requisições HTTP que chegam até o *back-end* e, posteriormente, retornar uma resposta para o cliente. Essas classes fazem tal controle por meio de *endpoints*, URLs que são acessadas dependendo da ação que o usuário busca fazer. Na aplicação foram utilizados três controladores, um para estimativa, um para estimador e um para produto.
- **DTO:** O pacote de *Data Transfer Objects* (Objetos de Transferência de Dados em português) é composto por classes que são utilizadas para a transferência de data entre diferentes partes da aplicação. Não possui regras de negócio, apenas os campos necessários das entidades. Na aplicação foram utilizados DTOs para cada uma das entidades e suas relações específicas, além de um DTO para erros, totalizando 6 dessas classes dentro do pacote.
- **Entity:** O pacote de entidades contém as classes que correspondem as entidades do banco de dados. Essas classes não possuem regras de negócio em seu código, apenas campos que definem seus atributos no sistema. A aplicação possui três entidades, sendo elas estimativa, estimador e produto. Seus atributos serão explicados no capítulo seguinte.



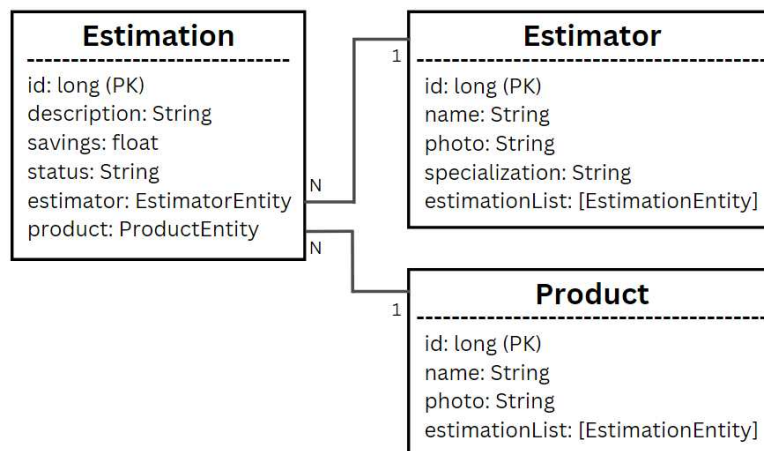
- **Exception:** O pacote de exceções contém classes que são utilizadas para fazer o tratamento de erros da aplicação. Por exemplo, ao tentar criar uma entidade que já existe no banco de dados, em vez de receber somente um código 409, que é o de conflito, a classe “EntityAlreadyExistsException” (em português, exceção de entidade já existente) é chamada e utilizada para detalhar o erro ao usuário. A aplicação é composta por dois arquivos principais de exceção, o de entidade já existente e o de entidade não encontrada.
- **Mapper:** O pacote de mapeadores é composto por classes responsáveis por converter objetos de um tipo para outro. Especificamente nessa aplicação os mapeadores foram usados para converter as classes de entidades para DTOs e vice-versa. O programa é composto por três classes mapeadoras, sendo uma para estimativa, outra para estimador e outra para produto.
- **Repository:** O pacote de repositórios possui classes que se comunicam diretamente com o banco de dados. Essas classes são compostas por métodos básicos de um CRUD, como encontrar uma entidade por meio do ID, alterar um campo em uma entidade por meio do nome dela, entre outros. A aplicação possui três repositórios, sendo um para estimativa, outro para estimador e outro para produto.
- **Request:** O pacote de requisições é composto por classes que definem a estrutura necessária dentro das requisições HTTP enviadas para o back-end para que o processo requisitado seja iniciado. Por exemplo, para a criação de uma nova estimativa no sistema os campos necessários, como estimador, descrição e afins, estão citados na respectiva classe de estimativa dentro do pacote *request*, enquanto campos como ID, que é automaticamente definido pelo back-end, não existem no *request* e, portanto, não devem vir junto com a requisição de criação de nova estimativa. A aplicação possui três classes de requisição, sendo uma para estimativa, outra para estimador e outra para produto.

- **Response:** O pacote de respostas é formado por classes que definem a estrutura da resposta que o back-end enviará ao final de cada requisição. No caso dessa aplicação, o formato padrão de resposta é composto por um código HTTP, uma mensagem de texto com o nome desse código e uma resposta da API, podendo ser um outro texto ou uma entidade. Na aplicação foram utilizados dois arquivos para lidar com as respostas, sendo um deles para respostas normais e outro para respostas de erros.
- **Service:** O pacote de serviços é composto por classes que contém as regras de negócio da aplicação. Essas classes fazem uso de diversos métodos de outros pacotes, como as mudanças de entidade para DTOs, a busca dentro de repositórios e afins. São responsáveis por lidar com as requisições enviadas para elas por meio dos controladores. A aplicação possui três classes de serviço, sendo uma para estimativa, outra para estimador e outra para produto.

#### 4.2.2. Entidades

Como apresentado na seção 4.2.1, no *back-end* foram escolhidas três entidades para a construção do sistema. Essas entidades possuem relações umas com as outras, como é possível ver no diagrama da Figura 18 a seguir:

Figura 18: Diagrama de entidades do Spring Boot



Fonte: Elaborado pelo autor, 2023

Essas relações, entretanto, não são iguais para os dois lados. Entre as entidades presentes na aplicação existem algumas relações N:1, lida como “muitos-para-um”. Esse tipo de associação ocorre quando uma entidade pode, por meio de um atributo, estar relacionada a mais de uma outra entidade.

Definidos esses conceitos, destacam-se as individualidades de cada entidade:

- **Estimation:** A entidade de estimativas possui os atributos ID, descrição, economias, estado, estimador e produto. Esses dois últimos atributos formam ligações muitos para um com suas respectivas entidades, pois uma estimativa limita-se somente a um estimador e produto específicos.
- **Estimator:** A entidade de estimador possui os atributos ID, nome, foto, especialização e lista de estimativas. Esse último atributo possui uma relação de um para muitos com a entidade de estimativas, pois o mesmo estimador pode ter várias estimativas no sistema.
- **Product:** A entidade de produto possui os atributos ID, nome, foto e lista de estimativas. Esse último atributo possui uma relação de um para muitos com a entidade de estimativas, pois o mesmo produto pode estar presente em mais de uma estimativa.

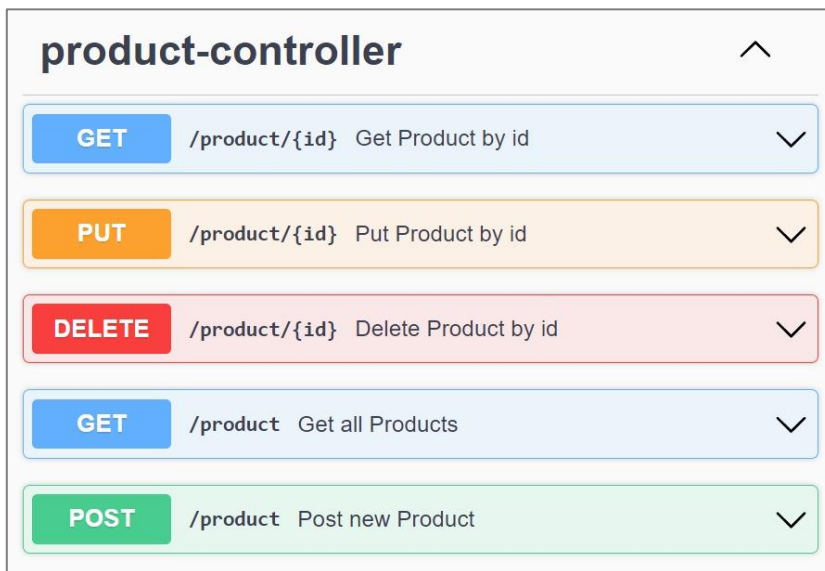
### 4.2.3. Endpoints

Como definido anteriormente, *endpoints* são URLs acessadas pelo usuário dependendo da ação que ele busca fazer, sendo tais funções correspondentes a métodos específicos de uma API, como os métodos GET (pegar), POST (criar), PUT (editar) ou DELETE (deletar).

Para documentar quais são os *endpoints* existentes na aplicação, foi utilizada uma biblioteca chamada Swagger, que providencia uma interface que permite fácil visualização de quais são os *endpoints* presentes na aplicação, bem como suas respostas possíveis.

Nas figuras a seguir (19 até 21) é possível observar os *endpoints* da aplicação listados por meio da interface do Swagger:

Figura 19: *Endpoints* presentes no controlador do produto



Fonte: Elaborado pelo autor, 2023

Figura 20: *Endpoints* presentes no controlador do estimador

estimator-controller		
GET	/estimator/{id}	Get Estimator by id
PUT	/estimator/{id}	Put Estimator by id
DELETE	/estimator/{id}	Delete Estimator by id
GET	/estimator	Get all Estimators
POST	/estimator	Post new Estimator

Fonte: Elaborado pelo autor, 2023

Figura 21: *Endpoints* presentes no controlador da estimativa

estimation-controller		
GET	/estimation/{id}	Get Estimation by id
PUT	/estimation/{id}	Put Estimation Status by id
DELETE	/estimation/{id}	Delete Estimation by id
GET	/estimation	Get all Estimations
POST	/estimation	Post new Estimation

Fonte: Elaborado pelo autor, 2023

Cada um desses *endpoints* possui mais de uma resposta possível. Por exemplo, o método GET do controlador do estimador não retorna necessariamente só o estimador

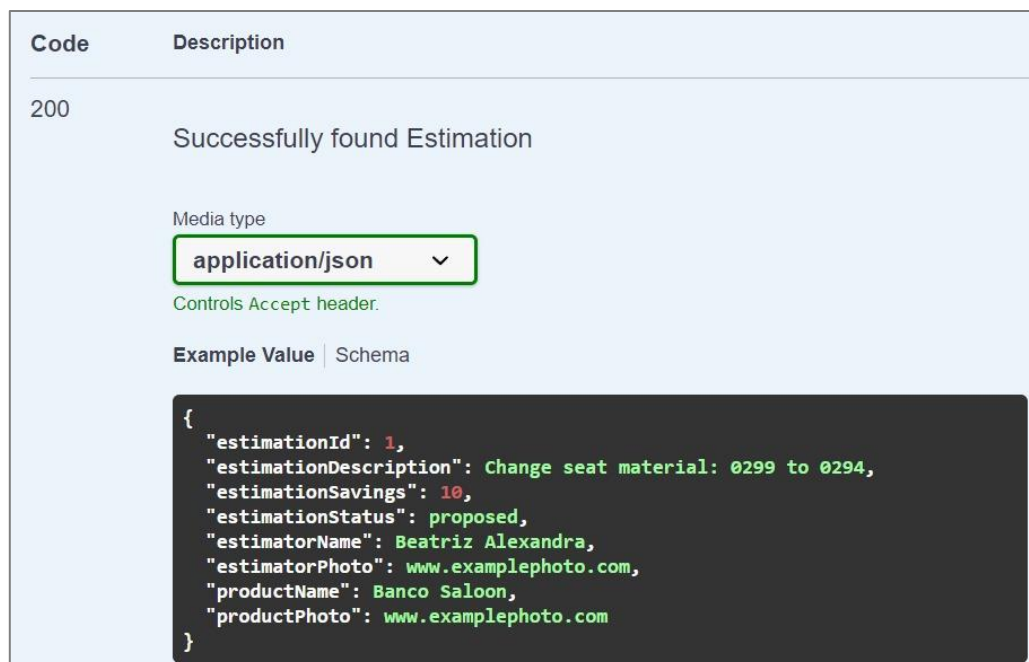
com aquele ID. É possível que não exista no sistema um estimador com aquela identificação, e nesse caso é necessário que o *endpoint* retorne uma mensagem de aviso.

Para documentar cada um dos retornos possíveis de cada um desses *endpoints*, foi utilizado um outro recurso dentro da biblioteca Swagger, chamado de ApiResponse. Com a adição de algumas diretivas de código no programa, foi adicionada à essa interface uma lista de respostas possíveis de cada um desses *endpoints*.

Como todos os *endpoints* tem algumas respostas padronizadas como os códigos 401 *Unauthorized* (Não autorizado) e 500 *Internal Server Error* (Erro interno do servidor), será apresentada a documentação de resposta de apenas um dos *endpoints* para evitar repetição.

Nas figuras a seguir (22 até 26) é possível observar, por meio da interface do Swagger com destaque para o método GET por ID, as respostas programadas:

Figura 22: Resposta com código 200 (OK): Estimativa encontrada com sucesso



Code	Description
200	Successfully found Estimation

Media type

Controls Accept header.

Example Value | Schema

```
{
  "estimationId": 1,
  "estimationDescription": Change seat material: 0299 to 0294,
  "estimationSavings": 10,
  "estimationStatus": proposed,
  "estimatorName": Beatriz Alexandra,
  "estimatorPhoto": www.examplephoto.com,
  "productName": Banco Saloon,
  "productPhoto": www.examplephoto.com
}
```

Fonte: Elaborado pelo autor, 2023

Figura 23: Resposta com código 400 (*Bad Request*): Requisição incorreta

400

Bad Request

Media type

application/json

Example Value | Schema

```
{
  "statusCode": 400,
  "message": Bad Request,
  "details": Bad Request
}
```

Fonte: Elaborado pelo autor, 2023

Figura 24: Resposta com código 401 (*Unauthorized*): Sem autorização

401

Unauthorized

Media type

application/json

Example Value | Schema

```
{
  "statusCode": 401,
  "message": Unauthorized,
  "details": Unauthorized
}
```

Fonte: Elaborado pelo autor, 2023

Figura 25: Resposta com código 404 (*Not Found*): Estimativa não encontrada

404

Estimation not found

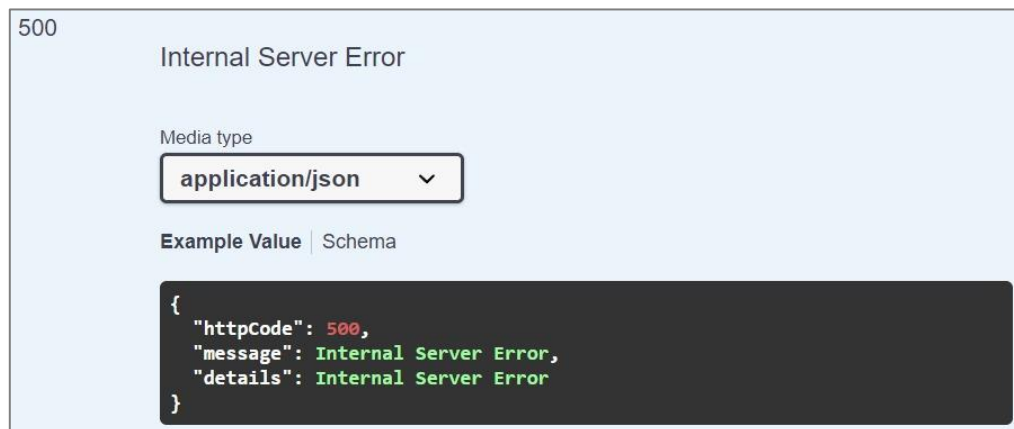
Media type

**application/json** ▾

**Example Value** | Schema

```
{
  "statusCode": 404,
  "message": "Not Found",
  "details": "Estimation not found"
}
```

Fonte: Elaborado pelo autor, 2023

Figura 26: Resposta com código 500 (*Internal Server Error*): Erro interno do servidor

500

Internal Server Error

Media type

**application/json** ▾

**Example Value** | Schema

```
{
  "statusCode": 500,
  "message": "Internal Server Error",
  "details": "Internal Server Error"
}
```

Fonte: Elaborado pelo autor, 2023



## 5. CONCLUSÕES

Neste projeto, a criação de uma aplicação web destinada a auxiliar a área de estimativa de custos foi executada com êxito, inserindo os requisitos definidos antes do início do projeto dentro de um programa com tecnologias modernas e práticas modernas de desenvolvimento, como a metodologia SOLID.

Foi desenvolvida uma aplicação robusta e ao mesmo tempo intuitiva, que permite com que os usuários da área de estimativa de custos possam criar estimativas e alterar seus estados, bem como acessar facilmente informações não só sobre essas estimativas como também dos produtos e da área no geral, satisfazendo todas as reclamações anotadas antes do início do desenvolvimento.

A escolha do Angular no front-end mostrou-se muito eficaz para a construção de um *software* escalável, leve e bem-organizado, facilitando o desenvolvimento de quaisquer funcionalidades que surgiram durante a construção do programa, bem como manutenção futura.

A utilização do Spring Boot no back-end mostrou-se igualmente efetiva, permitindo com que tenha sido feita uma clara divisão de funcionalidade em cada uma das classes criadas durante o desenvolvimento, garantindo uma aplicação padronizada de fácil compreensão e manutenção.

Conclui-se, portanto, que o desenvolvimento do Hub de Estimativas resultou em um produto com potencial para contribuir diretamente para a área de estimativa de custos, além de um eficaz exercício de programação para o desenvolvedor envolvido, não só pelas tecnologias utilizadas como também pela aplicação na prática de teorias como a metodologia SOLID e a orientação a objetos.

## 5.1. Trabalhos futuros

Por mais completo que seja um programa, é da visão do autor que sempre há espaço para melhorias, e com o Hub de Estimativas não é diferente.

Tratando-se, teoricamente, de um programa interno de uma empresa, não foi definida a necessidade de criação de permissões, tornando desnecessário qualquer implementação de um sistema de contas. Tal implementação, entretanto, traria um nível a mais de solidez para o *software*, permitindo com que, por exemplo, somente administradores poderiam fazer modificações nos estados das estimativas, enquanto estimadores poderiam apenas criar estimativas no sistema.

Ademais, seria possível também implementar um sistema de registro dentro da aplicação, local onde seria guardado um histórico de quais foram as mudanças feitas no programa e quem foi o autor destas. Isso seria útil em situações em que, por exemplo, existissem dois ou mais administradores responsáveis por alterar os estados das estimativas.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA, Claudio. **Sistemas Transacionais Gerenciais e Estratégicos**. UNIP, 2012. Disponível em: <<https://pt.slideshare.net/s3k7or/sistemas-transacionais-gerenciais-e-estrategicos>>. Acesso em: 21/10/2023.

DUARTE, Luiz. **Como usar TypeScript com Node.js**. LuizTools, 2020. Disponível em: <<https://www.luiztools.com.br/post/como-usar-typescript-com-node-js>>. Acesso em: 12 de julho de 2023.

EBY, Kate. **O guia completo para estimar custos de projeto**. Smartsheet, 2017. Disponível em: <<https://pt.smartsheet.com/ultimate-guide-project-cost-estimating>>. Acesso em: 25 de junho de 2023.

INTELLIPAAT. **What is Cost Estimation?** 2023. Disponível em: <<https://intellipaat.com/blog/what-is-cost-estimation/?US>>. Acesso em: 26 de junho de 2023.

MARTIN, Robert C. **Design Principles and Design Patterns**. Object Mentor, 2000. Disponível em: <[http://staff.cs.utu.fi/jounsmad/doos\\_06/material/DesignPrinciplesAndPatterns.pdf](http://staff.cs.utu.fi/jounsmad/doos_06/material/DesignPrinciplesAndPatterns.pdf)>. Acesso em: 29 de junho de 2023.

MISLICK, Gregory K.; NUSSBAUM A., Daniel. Cost Estimation: **Cost Estimation Methods and Tools**. Nova Jersey: Wiley, 2015.

MONDAY. **Cost estimate in project management: a critical success factor**. 2022. Disponível em: <<https://monday.com/blog/project-management/cost-estimation-in-project-management/>>. Acesso em: 25 de junho de 2023.

PROEST AUTODESK. **Benefits of construction cost estimating software**. 2021. Disponível em: <<https://proest.com/construction/estimating/benefits-of-cost-estimation/>>. Acesso em: 21 de outubro de 2023.

RUDNIK, Anastasia. **What is a web application and how does it work**. Medium, 2020. Disponível em: <<https://medium.com/web-space-team/what-is-a-web-application-and-how-does-it-work-2774e90797ba>>. Acesso em: 10 de julho de 2023.

THUNDERBOLT INNOVATION. **The Evolution of construction estimating**. 2019. Disponível em: <[https://www.thunderboltinnovation.com/thunderbolt\\_university/the-evolution-of-construction-estimating/](https://www.thunderboltinnovation.com/thunderbolt_university/the-evolution-of-construction-estimating/)>. Acesso em: 20 de outubro de 2023.

WATTS, Stephen. **The Importance of SOLID Design Principles**. BMC, 2020. Disponível em: <<https://www.bmc.com/blogs/solid-design-principles/>>. Acesso em: 29 de junho de 2023.

WILLIAM, Jackson. **Web Application Architecture**. ClickITTech, 2022. Disponível em <<https://www.clickittech.com/devops/web-application-architecture/>>. Acesso em: 10 de julho de 2023.

XP EDUCAÇÃO. **Aplicação web: entenda sua importância + para que serve**. 2022. Disponível em: <<https://blog.xpeducacao.com.br/aplicacao-web/>>. Acesso em: 11 de julho de 2023.