



UNICEPLAC

Centro Universitário do Planalto Central Aparecido dos Santos

Curso de Sistemas de Informação

Trabalho de Conclusão de Curso

Fretex: sistema para contratação de fretes

Brasília-DF

2020



UNICEPLAC

**GUSTAVO HENRIQUE ARAÚJO PALMEIRA
MATEUS DE SOUZA TELES
MIKAELLA DA SILVA ANTUNES**

Fretex: sistema para contratação de fretes

Artigo apresentado como requisito para conclusão do curso de Bacharelado em Sistemas de Informação pelo Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac.

Orientador: Prof. Esp. Hélder Line Oliveira

Brasília-DF

2020



UNICEPLAC

**GUSTAVO HENRIQUE ARAÚJO PALMEIRA
MATEUS DE SOUZA TELES
MIKAELLA DA SILVA ANTUNES**

Fretex: sistema para contratação de fretes

Artigo apresentado como requisito para conclusão do curso de Bacharelado em Sistemas de Informação pelo Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac.

Gama, 03 de dezembro de 2020.

Banca Examinadora

Prof. Hélder Line Oliveira
Orientador

Prof. Osman Brás de Souto
Examinador

Prof. Sebastião Ivaldo Carneiro Portela
Examinador



UNICEPLAC

Fretex: sistema para contratação de fretes

Gustavo Henrique Araújo Palmeira¹

Mateus de Souza Teles²

Mikaella da Silva Antunes³

Resumo:

A pesquisa tem como objetivo apresentar o processo de desenvolvimento de um software para auxiliar no transporte de cargas. A ideia para o desenvolvimento dessa pesquisa veio a partir de casos de reclamações de pessoas próximas ao grupo referente à falta de praticidade na contratação de fretes, com isso foi definido o desenvolvimento do software. As ferramentas utilizadas no desenvolvimento do trabalho foram React Native, PostgreSQL e as linguagens JavaScript e Java. O resultado esperado com o desenvolvimento dessa aplicação é facilitar o processo de comunicação e transporte das cargas entre as partes envolvidas com um baixo custo, ou seja, o cliente poderá chegar em um preço que agrade tanto ele quanto o prestador, sem que tenha um preço tabelado, e que também seja de confiança, pois toda a negociação e interação estará armazenada no banco de dados.

Palavras-chave: Cargas. Frete. Software

Abstract:

The research aims to present the process of developing a software to assist in the transport of cargo. The idea for the development of this research came from cases of complaints from people close to the group regarding the lack of practicality in the contracting of freight, with this was defined the development of the software. The tools used in the development of the work were React Native, PostgreSQL and the JavaScript and Java languages. The expected result with the development of this application is to facilitate the process of communication and transport of cargo between the parties involved with a low cost, that is, the customer can arrive at a price that pleases both, without having a list price, and that is also reliable, because all negotiation and interaction will be stored in the database.

Keywords: Cargo. Freight. Software

1 Graduando do Curso de Sistemas de Informação, do Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac. E-mail: gustavohenriquearaujo@hotmail.com.

2 Graduando do Curso de Sistemas de Informação, do Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac. E-mail: mateus_souza_teles@hotmail.com.

3 Graduanda do Curso de Sistemas de Informação, do Centro Universitário do Planalto Central Aparecido dos Santos – Uniceplac. E-mail: mantunes2903@gmail.com.

1 INTRODUÇÃO

Dentre os tipos de transporte de cargas presentes no Brasil, o transporte rodoviário de cargas “[...] é o modal predominante na matriz de transporte brasileira, sendo responsável por 60% da movimentação de cargas.” (CONFEDERAÇÃO NACIONAL DO TRANSPORTE, 2019b, p. 1). Isso mostra a importância desse segmento no país e também sua força econômica.

Tendo como base esse amplo mercado e com foco no segmento da matriz de transporte brasileiro, este projeto visa o desenvolvimento de um software para dispositivos móveis, que tem a finalidade de promover integração entre clientes que necessitam transportar cargas e os transportadores que podem realizar o fretamento. O software tem como objetivo principal facilitar o processo de negociação de fretes de cargas.

1.1 Problema de Pesquisa

Sabe-se que existe certa dificuldade em realizar a contratação de fretes com prestador de serviço devido a burocracia das transportadoras. Por essa razão, empresas e pessoas procuram transporte de forma informal, muitas vezes através de redes sociais, para conseguir fretamento de cargas. No entanto, a descentralização e falta de qualificação dos prestadores deste tipo de serviço implica em problemas como extravio de carga, descumprimento de prazos e baixa qualidade na prestação do serviço.

Sabendo-se desses problemas que são comumente encontrados nos processos de fretes de cargas, é necessária a adoção de uma estratégia para o ganho de produtividade e redução de custos em caráter permanente para obtenção de sucesso nas operações e confiabilidade entre as partes envolvidas.

Diante dessa necessidade, a tecnologia apresenta-se como uma ferramenta adequada para mediar as relações de trabalho entre transportadores (prestadores de serviço) e clientes (contratantes) a exemplo de grandes empresas como Uber, Ifood e Rappi que entregam valor a partir da redução da complexidade entre as partes interessadas em um determinado serviço.

Desta forma, este projeto aborda como elaborar uma solução de software que atenda as necessidades de negócio do ramo de fretes, para facilitar a busca de contratação e formalização deste serviço de forma digital e simplificada.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver um software para dispositivos móveis que tenha a finalidade de atender necessidades existentes nas atividades de transporte de cargas, otimizando os processos de contratação e gerenciamento de fretes com serviços que ofereçam aos transportadores e clientes maior praticidade. O principal objetivo do projeto é prover recursos para que clientes e transportadores possam negociar fretes utilizando um aplicativo em dispositivo móvel com praticidade e agilidade.

1.2.2 Objetivos Específicos

- a) Realizar estudos das necessidades das partes interessadas, definindo os *stakeholders* bem como suas respectivas necessidades e funcionalidades;
- b) Levantar os requisitos funcionais do sistema, com base nas experiências esperadas que os usuários tenham ao utilizá-lo e os comportamentos esperados do sistema;
- c) Definir as ferramentas e tecnologias a serem utilizadas;
- d) Criar diagramas que definem a estrutura do sistema (diagrama de classes e fluxograma);
- e) Desenvolver a parte lógica do sistema (*back-end*), contendo as regras de negócio e persistência de dados;
- f) Desenvolver a parte visual do sistema (*front-end*), contendo telas com as interfaces com as quais o usuário interage, sendo através de formulários, links e botões;
- g) Realizar testes simulados, cadastrando usuários fictícios e realizado todos os processos do sistema, verificando assim se o sistema está funcional e se atende às necessidades levantadas.

2 REVISÃO DE LITERATURA

2.1 O uso da tecnologia na logística de transportes rodoviários

Mesmo que os investimentos em outros modais venham aumentando, o modal rodoviário ainda tem participação expressiva na matriz de transportes brasileira. Dentro deste

modal de transporte estão presentes as empresas transportadoras, com grandes frotas de caminhões e também os caminhoneiros autônomos. Estes caminhoneiros não possuem vínculos com empresas transportadoras e decidem sobre o seu trabalho, bem como são responsáveis pelas despesas e custos provenientes dele.

O caminhoneiro autônomo também é conhecido como Transportador Autônomo de Cargas (TAC). Este profissional é responsável por gerenciar seu próprio veículo e também as etapas do processo de transporte, como a retirada, o transporte e a entrega da carga (CARGOBR, 2019).

À medida que os smartphones se tornam mais populares entre os motoristas de caminhão, a negociação fretes de cargas pela internet se torna mais atraente. Os avanços da tecnologia vêm revolucionando a maneira como são prestados serviços em vários segmentos, e o transporte também pode fazer parte dessa revolução (BORTOLIN, 2014).

De acordo com a Confederação Nacional do Transporte (2019a), 98% dos caminhoneiros frequentemente acessam a internet por meio de um celular/smartphone. Isso mostra a presença significativa dos dispositivos móveis na profissão.

2.2 Arquiteturas tecnológicas para o desenvolvimento do sistema

Para o desenvolvimento do sistema foram definidas algumas arquiteturas tecnológicas e ferramentas que auxiliam nos processos de desenvolvimento de softwares, seguindo uma sequência de desenvolvimento. Essas arquiteturas são listadas na Tabela 1.

Tabela 1 – Utilização das arquiteturas tecnológicas

Arquitetura e seus usos	
Arquitetura tecnológica	Onde foi utilizada
Docker	Utilizado para disponibilizar o banco de dados em um container para facilitar o acesso e portabilidade.
PostgreSQL	Banco de dados relacional utilizado para armazenar os dados do sistema.
Java	Utilizado para programar a lógica da aplicação, ou seja, o <i>back-end</i> .

Spring	<i>Framework</i> utilizado junto ao Java para ajudar na construção das APIs (<i>Application Programming Interface</i> - Interface de Programação de Aplicativos) do sistema.
Spring Tool Suite	Ferramenta utilizada para facilitar o uso do <i>framework</i> Spring.
Insomnia	Utilizado para fazer as requisições http das APIs.
JavaScript	Utilizado para a construção do <i>front-end</i> , ou seja, a parte visível ao usuário.
React Native	<i>Framework</i> usado junto ao JavaScript para auxiliar na construção do <i>front-end</i> para a aplicação rodar nativamente tanto em plataformas Android quanto iOS.
Expo	Utilizado em conjunto com o React Native para a visualização da aplicação diretamente no celular físico.

Fonte: Dos autores, 2020.

As tecnologias apresentadas anteriormente são definidas a seguir.

2.2.1 Java

Tecnologia orientada a objetos (OOP) utilizada no desenvolvimento de softwares. O Java é uma linguagem de programação bastante presente no desenvolvimento de softwares, sendo encontrada em uma grande variedade de dispositivos, como laptops, consoles de games, supercomputadores e telefones celulares (JAVA, 2020).

O Java foi escolhido para ser utilizado no desenvolvimento deste projeto porque, segundo Java (2020), foi projetado para poder ser utilizado no desenvolvimento de aplicações portáteis que necessitam de bom desempenho e pode ser utilizado em vários tipos de plataformas de computação.

2.2.2 Spring

É uma plataforma Java que fornece suporte de infraestrutura abrangente para o desenvolvimento aplicativos utilizando a linguagem Java. O Spring abstrai configurações de infraestrutura do projeto para que o desenvolvedor possa se concentrar no desenvolvimento das regras de negócio. Spring torna a programação Java mais rápida, fácil e segura, se concentrando

em oferecer velocidade, simplicidade e produtividade no desenvolvimento de softwares. Isso o torna um dos *frameworks* Java mais popular do mundo (SPRING, 2020b).

2.2.3 Spring Tool Suite

Conhecido pela abreviação STS, é uma ferramenta de desenvolvimento de softwares projetada para desenvolver aplicativos Java baseados em Spring. De acordo com Spring (2020a), o STS é construído sobre o *Eclipse Integrated Development Environment (IDE)* e adaptado para o desenvolvimento de aplicativos corporativos usando Spring com complementos potentes que simplificam a criação e o trabalho, oferecendo ferramentas projetadas para utilizar o *framework* Spring de forma mais rápida e menos trabalhosa.

O Spring Tool Suite, em conjunto com o Spring, foi adotado neste projeto com o objetivo de simplificar o desenvolvimento do sistema, contendo abstrações de código que permitem aos desenvolvedores focarem mais na implementação das regras de negócio.

2.2.4 PostgreSQL

É um Sistema Gerenciador de Banco de Dados (SGBD) de código aberto que usa e estende a linguagem SQL combinada com muitos recursos que armazenam e escalam com segurança as cargas de trabalho de dados mais complicadas (POSTGRESQL, 2020).

O PostgreSQL foi escolhido para o desenvolvimento deste projeto por ser gratuito e de código aberto, além de um robusto gerenciador de banco de dados que permite o armazenamento e gerenciamento de dados.

2.2.5 Docker

É uma plataforma que possibilita a criação, execução, publicação e gerenciamento de containers. No cenário de desenvolvimento de softwares, o Docker simplifica e acelera o fluxo e trabalho, enquanto dá aos desenvolvedores a liberdade de inovar com sua escolha de ferramentas, pilhas de aplicativos e ambientes de implantação para cada projeto (DOCKER, 2020).

Neste projeto, o uso do Docker foi definido para instalação do banco de dados da aplicação em container, o que elimina a necessidade de instalar o banco de dados no computador no qual o projeto foi desenvolvido. Isso torna o banco mais fácil de ser inicializado e de ser

portado para outros ambientes.

2.2.6 JavaScript

Segundo Souza (2019), o JavaScript é uma linguagem de programação de alta complexidade, mas fácil de usar, projetada para criar elementos em aplicativos da web, como sites, aplicativos e sistemas on-line. Nos últimos anos, seu uso tem aumentado de forma expressiva, tornando-a uma das linguagens mais populares do mundo e uma ótima ferramenta para desenvolvedores.

Inicialmente o JavaScript foi desenvolvido para rodar apenas em navegadores web, porém esta linguagem vem ganhando grande espaço do desenvolvimento *back-end* (lógica da aplicação, geralmente contida em um servidor) e *mobile* (para dispositivos móveis).

Essa tecnologia foi definida para ser usada no projeto por ser uma linguagem de programação que vem se tornando cada vez mais utilizada no desenvolvimento de aplicativos para dispositivos móveis e também por ser a linguagem utilizada pelo *framework* React Native, também utilizado no projeto e descrito a seguir.

2.2.7 React Native

É um *framework* que possibilita o desenvolvimento de aplicações para dispositivos móveis. Segundo Cabral (2016), esta tecnologia foi desenvolvida pelos engenheiros do Facebook e é constituída por uma série de ferramentas que viabilizam a criação de aplicações móveis nativas para as plataformas iOS e Android, utilizando o que há de mais moderno no desenvolvimento *front-end*.

Esta tecnologia foi definida para ser utilizada no projeto por possibilitar que um aplicativo seja desenvolvido uma única vez e disponibilizado tanto para celulares com sistema operacional Android quanto iOS. Com isso o tempo de desenvolvimento é reduzido de forma significativa.

2.2.8 Expo

É um conjunto de ferramentas e serviços criados em torno do React Native e de plataformas nativas que ajudam a desenvolver, construir, implantar e iterar rapidamente em aplicativos iOS, Android e web a partir da mesma base de código (EXPO, 2020).

O Expo também possibilita que a aplicação seja executada e testada diretamente em um celular físico através da rede *wireless*, sem a necessidade de conexão cabeada entre o dispositivo e o computador. Isso oferece melhor experiência para o desenvolvedor além de permitir que a interface da aplicação seja visualizada em um dispositivo real. Por essa razão o Expo foi adotado no desenvolvimento do projeto.

2.2.9 Insomnia

É um software gratuito que tem por objetivo proporcionar ao usuário interação com APIs - *Application Programming Interface* (Interface de Programação de Aplicativos) baseadas no protocolo HTTP. O Insomnia oferece uma interface fácil de usar, com funcionalidades avançadas, como auxiliares de autenticação, geração de código e variáveis de ambiente. Com esta ferramenta é possível fazer requisições a *APIs Rest* de forma rápida e fácil (INSOMNIA, 2020).

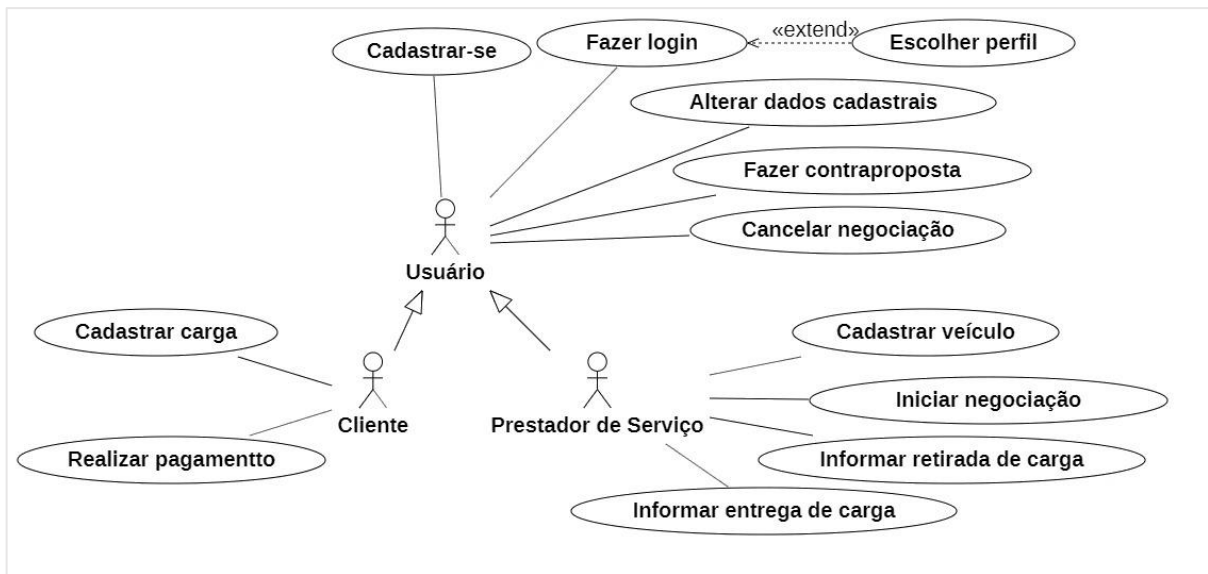
Por esta razão, essa tecnologia foi definida para ser usada no desenvolvimento do projeto para realizar testes durante o desenvolvimento das *APIs* do sistema, simulando o papel do *front-end*.

3 PROCEDIMENTOS METODOLÓGICOS

Para que um software tenha bons resultados em seu processo de desenvolvimento, é necessário que seja feita a especificação de itens de aplicabilidade de acordo com suas respectivas funções. Nessa especificação podem ser utilizados documentos e diagramas que detalham as funcionalidades e a estrutura do sistema. Para atender essa prática, este projeto contemplou em seus processos iniciais a construção dos diagramas de caso de uso e de classes, demonstrados nas figuras 1 e 2, e também o documento de levantamento de requisitos do sistema, apresentado no APÊNDICE A.

Para construção do projeto foi realizada avaliação para definir quais as funções e comportamentos que deveriam ser adotadas pelo sistema. Essas funções são demonstradas no Diagrama de casos de uso ilustrado na Figura 1, apresentada a seguir.

Figura 1 – Diagrama de casos de uso

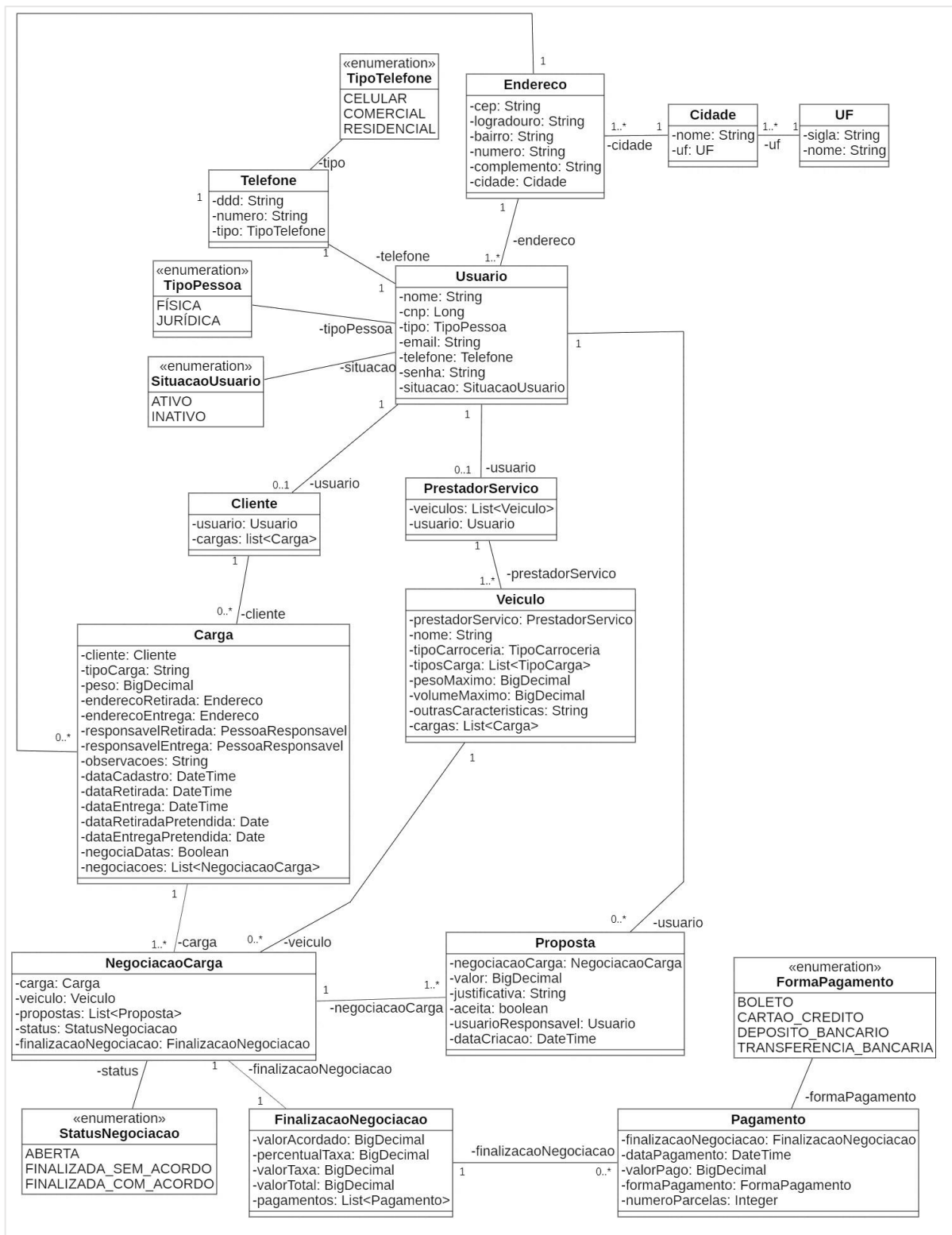


Fonte: Dos autores, 2020.

O Diagrama de casos de uso demonstra os atores e as ações que podem realizar no sistema. Nele são apresentadas as operações ligadas a cada um dos atores. No diagrama pode-se observar que existem funcionalidades que são comuns tanto para os clientes quanto os prestadores de serviço. Isso é demonstrado através da herança do ator “Usuário” para os atores “Cliente” e “Prestador de Serviço”, onde os atores filhos possuem todas as funcionalidades do ator pai.

Para demonstrar a estrutura do sistema, a Figura 2, apresentada a seguir, ilustra o Diagrama de classes, que define como foram divididas as informações sistematicamente, onde foram inseridos os atributos e sua respectiva tipagem.

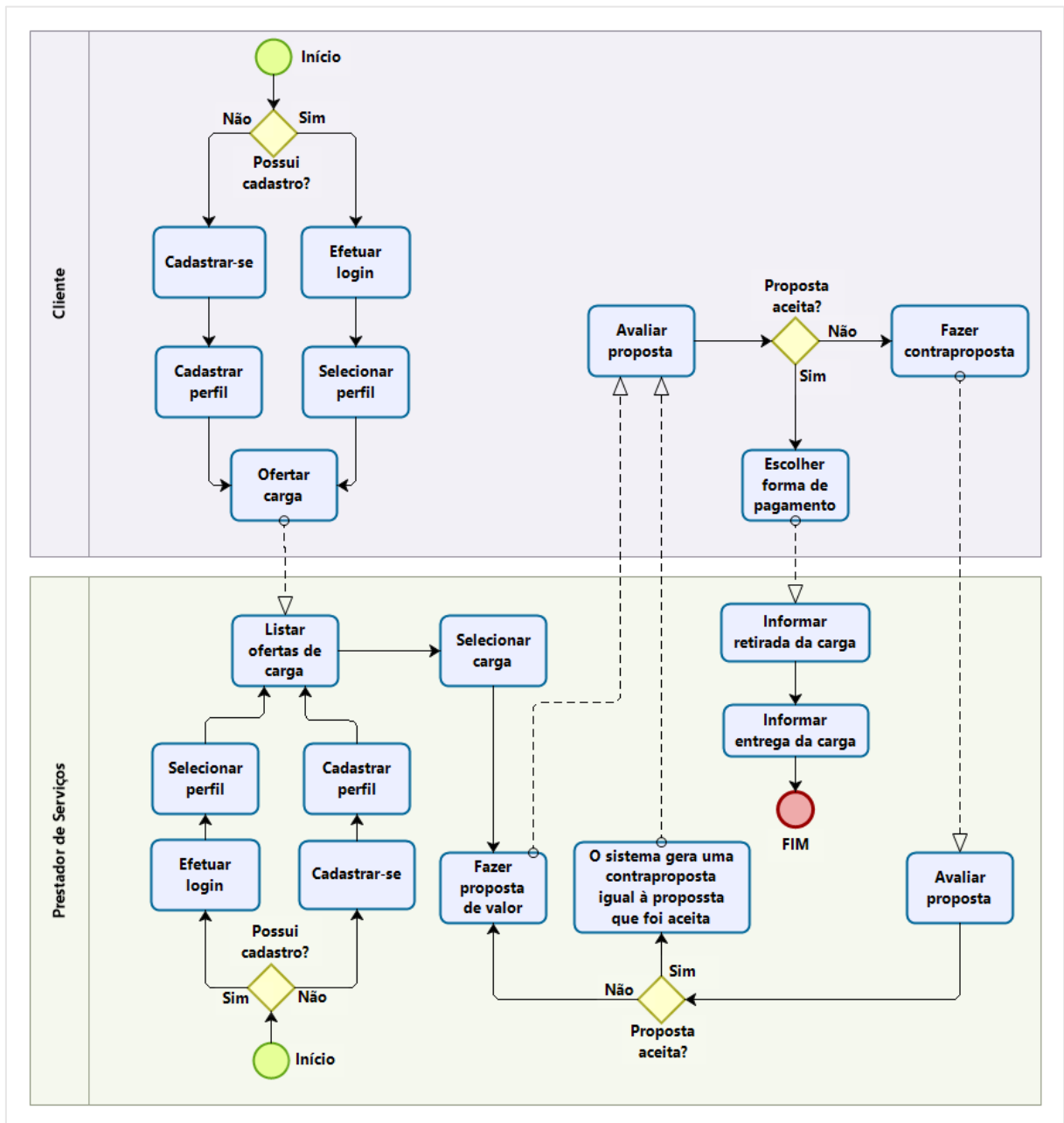
Figura 2 – Diagrama de classes



Fonte: Dos autores, 2020.

A Figura 3, apresentada a seguir, ilustra o fluxograma do sistema. O fluxograma representa graficamente a sequência de etapas e processos a serem executados pelo sistema para atingir um resultado, obtido a fim de atender ao problema de pesquisa de forma que possibilite interações entre clientes e transportadores a fim de negociar fretes de cargas. Através dele é possível identificar o fluxo de ações realizadas por cada ator desde o acesso ao sistema até o resultado final das interações entre eles.

Figura 3 – Fluxograma



Fonte: Dos autores, 2020

4 APRESENTAÇÃO E ANÁLISE DOS DADOS

Foi criada uma tela para realizar o cadastro dos usuários, contendo apenas os dados pessoais do usuário. Após ser realizado o cadastro, o usuário escolhe qual o perfil que deseja utilizar, sendo eles o de cliente e o de prestador de serviços. Um usuário pode cadastrar os dois perfis, usando um por vez e podendo alternar qual perfil deseja utilizar quando quiser, conforme a Figura 4.

Figura 4 – Telas de cadastro e seleção de perfil

The figure displays three sequential screenshots of the FRETEx mobile application interface:

- Left Screenshot (15:55):** Titled "Cadastro de usuário". It features a form with the following fields: "Nome completo: *" (input: "Nome"), "CPF/CNPJ: *" (input: "CPF/CNPJ"), "E-mail: *" (input: "E-mail"), "Telefone: *" (split into "DDD" and "Número" inputs), and "Senha: *" (input: "Senha"). A blue "Salvar" button is at the bottom.
- Middle Screenshot (15:50):** Titled "Cadastro de usuário". It shows the same form with red error messages: "Nome completo: * Obrigatório", "CPF/CNPJ: * Deve ter no mínimo 11 caracteres", "E-mail: * Digite um e-mail válido", and "Senha: * Obrigatório". The inputs contain "Nome", "123", "aaa", and "Senha".
- Right Screenshot (16:06):** Titled "Seleção de perfil". It shows a profile selection screen with the heading "Selecione o perfil:". There are two buttons: a purple "Cliente" button and a green "Prestador de serviços" button.

Fonte: Dos autores, 2020

O usuário com perfil de cliente pode realizar solicitações de frete cadastrando informações sobre a carga a ser transportada e acompanhar os detalhes da solicitação, conforme a Figura 5. A tela de detalhes da carga é visível tanto para o cliente quanto para o prestador de serviços, sendo importante para a negociação.

Figura 5 – Tela de descrição da carga e informações do frete

The image displays two screenshots of the FRETEX mobile application interface. The left screenshot, titled 'Solicitação de frete', shows a form for creating a freight request. It includes fields for 'Carga a ser transportada: *' (with a placeholder 'Tipo de carga. Ex: geladeira'), 'Peso: *' (with a placeholder 'Peso da carga (Kg)'), 'Observações:' (with a placeholder 'Observações sobre a carga'), 'Data pretendida para retirada:' (with a placeholder 'Data pretendida para a carga ser ret'), 'Data pretendida para entrega:' (with a placeholder 'Data pretendida para a carga ser eni'), and 'Endereços:' (with a placeholder 'Endereço de partida *'). The right screenshot, titled 'Detalhes da carga', displays the details for a specific shipment. It lists: 'Tipo de carga: Gado nelore', 'Peso: 2500 Kg', 'Observações: O gado é um pouco bravo', 'Endereço de partida: Rua Desembargador Paula Motta, 01, , Vila Oliveira, Ponte Nova - MG, CEP: 35430123', 'Endereço de entrega: Rua Benedito Valadares, 150, . Centro, Ponte Nova - MG, CEP: 35430012', 'Data de cadastro: 21/11/2020 16:18', 'Data de retirada: 23/11/2020 08:00', 'Data de entrega: 23/11/2020 16:18', and 'Negociações:'. At the bottom, it shows 'Veículo: Caminhão gaiola' and 'Status: FINALIZADA_COM_ACORDO' with a right-pointing arrow.

Fonte: Dos autores, 2020

Na tela de negociação, ilustrada na Figura 6, é apresentado um *card* para ambos os usuários que estão participando da negociação, parte principal da aplicação, em que se realizará o *link* entre cliente e prestador de serviços. Nessa tela haverá a troca de propostas para confirmação do frete.

Figura 6 – Tela de negociação

The figure displays two screenshots of the FRETEx mobile application interface. The left screenshot, titled 'Proposta', shows a form for creating a proposal. It includes fields for 'Data de retirada: *' (23/11/2020), 'Data de entrega: *' (23/11/2020), 'Valor: *' (Valor do frete), and 'Justificativa:' (Justificativa para o valor). A 'Salvar' button is at the bottom. The right screenshot, titled 'Detalhes da negociação', shows negotiation details. It includes 'Carga: Gado nelore', 'Veículo: Caminhão gaiola - (suporta até 5000 Kg)', and 'Situação: FINALIZADA_COM_ACORDO'. Below this are two proposal cards. The first card is for the 'Prestador' (Provider), dated 21/11/2020 at 16:10:39, with a value of R\$1000,00 and status 'NÃO ACEITA'. The second card is for the 'Cliente' (Client), dated 21/11/2020 at 16:17:41, with a value of R\$800,00 and status 'NÃO ACEITA'. Both cards have 'Aceitar' and 'Contrapor' buttons.

Fonte: Dos autores, 2020

Na etapa de configuração do ambiente de desenvolvimento, foram encontradas dificuldades na utilização da tecnologia Docker quanto ao acesso ao banco de dados pela aplicação. Após algumas horas de análise, foi descoberto que problema era referente à porta de acesso do container. O problema foi corrigido criando novo container e expondo a porta padrão do banco de dados.

O sistema desenvolvido permite de forma prática e sem burocracia a adesão prestadores de serviços que possuam algum veículo que possa transportar cargas. Isso pode ser considerado um diferencial se comparado a outros sistemas já existentes no mercado que possuem as mesmas finalidades. Desta forma, torna-se um produto mais acessível para prestadores de serviços e clientes de pequeno porte, não se limitando apenas a grandes empresas.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento do software foi realizado com base em requisitos funcionais definidos pelos próprios desenvolvedores, levando em consideração possíveis situações que o projeto venha a solucionar. A criação foi baseada em ferramentas que auxiliam nos processos de desenvolvimento de softwares. As maiores dificuldades encontradas foram o aprofundamento no conhecimento da solução Docker e React Native e determinação de requisitos funcionais na visão do prestador de serviço de transporte.

O produto final foi baseado na plataforma mobile, devido ao tempo para levantamento de requisitos, documentação e desenvolvimento ser restrito, tendo como funcionalidades as opções de solicitação de frete, negociação de valor e acompanhamento da carga. O protótipo é funcional e corrobora a viabilidade da aplicação e de seus objetivos.

Como projetos futuros pode ser desenvolvida uma versão web para utilização do sistema em navegadores de internet e outros módulos podem ser agregados, tais como: cadastro de postos de gasolina, restaurantes e oficinas, por partes de atores específicos, que visem a prestação de serviços ao transportador. Outro projeto que pode ser desenvolvido e assim adicionar maior segurança e melhorar o sistema é a implementação da finalização de pagamentos das negociações, sendo por geração de boletos, cobrança por meio de cartão de crédito ou outras formas de pagamento possam ser disponibilizadas.

REFERÊNCIAS

BORTOLIN, Nelson. **Carga Pesada**. 2014. Disponível em: <https://cargapesada.com.br/2014/08/13/popularizacao-dos-smartphones-impulsiona-centrais-de-fretes-on-line>. Acesso em: 26 nov. 2020.

CABRAL, Carlos. **Construa aplicações móveis nativas com JavaScript**. 2016. Disponível em: <https://tableless.com.br/react-native-construa-aplicacoes-moveis-nativas-com-javascript>. Acesso em: 28 set. 2020.

CARGOBR. **A DIFERENÇA entre caminhoneiro autônomo, agregado e colaborador**. 2019. Disponível em: <https://blog.cargobr.com/caminhoneiro-autonomo>. Acesso em: 23 nov. 2020.

CONFEDERAÇÃO NACIONAL DO TRANSPORTE (Brasil). **Pesquisa CNT Perfil dos Caminhoneiros**. 2019a. Disponível em: <https://cnt.org.br/pesquisas>. Acesso em: 23 nov. 2020.

CONFEDERAÇÃO NACIONAL DO TRANSPORTE (Brasil). **Transporte em Movimento: as vantagens de transportar sem sobrecarga**. 2019b. Disponível em:

<https://cdn.cnt.org.br/diretorioVirtualPrd/4b20b79f-1d61-42d9-9909-f021c408f58e.pdf>.
Acesso em: 23 abr. 2020.

DOCKER. **Developing with Docker**. [2020]. Disponível em: <https://www.docker.com/why-docker>. Acesso em: 28 set. 2020.

EXPO. **Introduction to Expo**. [2020]. Disponível em: <https://docs.expo.io>. Acesso em: 28 set. 2020.

INSOMNIA. **Getting Started with Insomnia**. 2020. Disponível em: <https://support.insomnia.rest/article/11-getting-started>. Acesso em: 28 set. 2020.

JAVA. **Obtenha Informações sobre a Tecnologia Java**. [2020]. Disponível em: <https://www.java.com/pt-BR/about>. Acesso em: 18 nov. 2020.

POSTGRESQL. **What is PostgreSQL?** [2020]. Disponível em: <https://www.postgresql.org/about>. Acesso em: 28 set. 2020.

SOUZA, Ivan. **JavaScript: o que é, como funciona e por que usá-lo no seu site**. 2019. Disponível em: <https://rockcontent.com/br/blog/javascript>. Acesso em: 23 nov. 2020.

SPRING. **Spring Tools 4**. [2020a]. Disponível em: <https://spring.io/tools>. Acesso em: 28 set. 2020.

SPRING. **Why Spring?** [2020b]. Disponível em: <https://spring.io/why-spring>. Acesso em: 23 nov. 2020.

Fretext: sistema para contratação de fretes Levantamento de Requisitos

Histórico de revisões do modelo

Versão	Data	Autor	Descrição
1.0	14/09/2020	Mikaella Antunes	Versão inicial
1.1	15/09/2020	Mateus Souza	Revisão do documento
1.2	01/11/2020	Mateus Souza	Revisão do documento
2.0	20/11/2020	Mateus Souza	Revisão, alteração da estrutura e adição de informações no documento

Aprovadores

Nome	Função
Gustavo Araújo	Desenvolvedor de Software
Mateus Souza	Desenvolvedor de Software
Mikaella Antunes	Analista de Documentação

1. Introdução

1.1. Propósito

Especificar os requisitos do sistema a ser desenvolvido, que será disponibilizado na versão *mobile*.

1.2. Público Alvo

O documento se destina aos desenvolvedores de software e *stakeholders*.

1.3. Escopo

Elicitar os requisitos de um sistema que tem o objetivo de atender necessidades existentes nas atividades de transporte de cargas otimizando os processos de contratação e gerenciamento de fretes.

1.4. Abreviações

Neste documento são utilizadas algumas abreviações, que são definidas a seguir:

- RF: requisito funcional;
- RNF: requisito não funcional;
- SGBD: Sistema Gerenciador de Banco de Dados.

1.5. Glossário

- **Requisito funcional** – Funcionalidade que descreve o que deve fazer em termo de tarefas e serviços.
- **Requisito não funcional** – Aspectos não funcionais de um sistema, como as restrições nas quais o sistema deve operar.
- **Prestador de serviço** – Transportador que disponibilizará o seu veículo para realizar fretes.

- **Cliente** – Usuário que solicita o serviço de frete.

2. Visão Geral do Produto

Este documento apresenta uma descrição geral do sistema e suas funcionalidades, especificando entradas e saídas dos requisitos funcionais e a descrição dos requisitos não funcionais contidos no sistema.

2.1. Descrição dos usuários no sistema

Os usuários do sistema poderão ser pessoas físicas ou jurídicas. Um usuário poderá ter tanto o perfil de cliente quando o de prestador de serviço.

2.2. Premissas e restrições

- **Acesso:** O usuário deve estar cadastrado no sistema para fazer qualquer solicitação.
- **Cadastro:** Não pode haver dois cadastros com o mesmo número de CPF/CNPJ ou e-mail. Um usuário pode ter os dois perfis, tanto de cliente, quanto de prestador de serviços. O usuário se auto cadastra no sistema.

3. Requisitos Funcionais

3.1. RF001: Cadastro de usuário

Descrição: O usuário se cadastra no sistema.

Entrada: Nome, e-mail, CPF/CNPJ, telefone e senha.

Processo: O cadastro será incluído no banco de dados.

Saída: Mensagem de confirmação do cadastro caso tenha sido efetuado, senão, mensagem de erro.

3.2. RF002: Efetuar Login

Descrição: O usuário que já possui cadastro pode efetuar login para acesso ao sistema.

Entrada: Informações de e-mail ou CPF/CNPJ e senha.

Processo: O sistema verifica se as credenciais informadas são válidas.

Saída: Redirecionamento para a tela de seleção do perfil, se não, mensagem de credenciais inválidas.

3.3. RF003: Solicitar redefinição de senha

Descrição: O usuário pode solicitar a redefinição de senha caso a esqueça.

Entrada: E-mail cadastrado no sistema.

Processo: O sistema verifica se o e-mail informado está cadastrado e, caso esteja, envia um código de recuperação para o e-mail.

Saída: Redirecionamento para a tela de redefinição de senha, senão, mensagem de erro.

3.4. **RF004: Redefinir senha**

Descrição: O usuário redefine a senha cadastrada.

Entrada: Código de recuperação e nova senha a ser cadastrada.

Processo: O sistema verifica se o código informado é válido e, caso seja, atualiza a senha do usuário no banco de dados.

Saída: Redirecionamento para a tela de login, senão, mensagem de erro.

3.5. **RF005: Selecionar perfil**

Descrição: O usuário pode escolher qual perfil deseja utilizar.

Entrada: Perfil selecionado

Processo: Quando um perfil é selecionado pelo usuário pela primeira vez, o sistema cadastra o perfil e vincula ao usuário. Para cadastrar o perfil de prestador de serviços, o usuário deve cadastrar um veículo. Quando o perfil selecionado já foi cadastrado anteriormente pelo usuário, o sistema personaliza todas as opções das telas de acordo com o perfil.

Saída: Redirecionamento para a tela principal com as opções de ações disponíveis para o perfil selecionado, senão, mensagem de erro.

3.6. **RF006: Cadastrar veículo**

Descrição: O usuário com perfil de prestador de serviços pode cadastrar um ou mais veículos.

Entrada: Informações sobre as características do veículo.

Processo: O cadastro é incluído no banco de dados.

Saída: Mensagem de confirmação do cadastro caso tenha sido efetuado, senão, mensagem de erro.

3.7. **RF007: Solicitar frete de carga**

Descrição: O cliente realiza solicitação de frete para uma carga.

Entrada: Informações sobre a carga a ser transportada.

Processo: O cliente solicita o frete com suas descrições e informa o local de retirada e entrega da carga.

Saída: O sistema grava a solicitação do frete no banco de dados.

3.8. **RF008: Selecionar a carga**

Descrição: O prestador de serviço escolhe uma solicitação de frete entre as que aparecem na listagem.

Entrada: Lista das solicitações de frete cadastradas por clientes.

Processo: O prestador de serviço visualiza a lista de solicitações e seleciona a desejada.

Saída: Redirecionamento para a tela com detalhes da solicitação escolhida.

3.9. RF009: Enviar proposta

Descrição: O prestador de serviços, após avaliar o frete, inicia uma negociação enviando uma proposta para realização do serviço.

Entrada: Datas do serviço e valor proposto.

Processo: O sistema salva a proposta e disponibiliza para o cliente aceitar ou contrapor.

Saída: Redirecionamento para a tela de detalhes da negociação com a lista de propostas atualizada, senão, mensagem de erro.

3.10. RF010: Realizar contraproposta

Descrição: Quando é feita uma proposta por um usuário, o outro usuário participante da negociação pode não concordar e fazer uma contraproposta.

Entrada: Datas do serviço e valor proposto.

Processo: O sistema salva a proposta e disponibiliza para o outro usuário aceitar ou contrapor.

Saída: Redirecionamento para a tela de detalhes da negociação com a lista de propostas atualizada, senão, mensagem de erro.

3.11. RF011: Aceitar proposta

Descrição: Estando de acordo com a última proposta recebida, o usuário aceita a proposta.

Entrada: O usuário aciona a opção de aceitar a proposta e confirma a ação.

Processo: Após a confirmação o sistema pode ter dois comportamentos:

1 - Caso o usuário que está aceitando seja o cliente, o sistema finaliza a negociação e redireciona o cliente para a tela de finalização do pagamento.

2 - Caso o usuário que está aceitando seja o prestador de serviço, o sistema gera uma nova proposta em nome do prestador para aprovação final do cliente, com as mesmas informações da proposta que ele está aceitando.

Saída: Redirecionamento para a tela de finalização do pagamento ou para a tela de detalhes da negociação com a lista de propostas atualizada, senão, mensagem de erro.

3.12. RF012: Finalizar pagamento

Descrição: Após a finalização da negociação, o cliente confirma a forma de pagamento.

Entrada: O cliente visualiza os detalhes do pagamento e seleciona a forma de pagamento desejada.

Processo: O sistema grava as informações no banco de dados.

Saída: Redirecionamento para a tela de negociação finalizada, senão, mensagem de erro.

3.13. RF013: Cancelar negociação

Descrição: Caso qualquer um dos usuários participantes de uma negociação queira, poderá cancelar a negociação.

Entrada: O usuário acessa a tela de detalhes da negociação, aciona o botão de cancelar e confirma a ação.

Processo: O sistema encerra a negociação, não permitindo mais a realização de propostas.

Saída: Mensagem de confirmação do cancelamento, senão, mensagem de erro.

3.14. RF014: Informar a retirada da carga

Descrição: O prestador de serviços informa que retirou a carga para transporte.

Entrada: Data e hora que a carga foi retirada.

Processo: O sistema insere a informação no banco de dados.

Saída: Mensagem de sucesso, senão, mensagem de erro.

3.15. RF015: Informar a entrega da carga

Descrição: O prestador de serviços informa que entregou a carga no destino final.

Entrada: Data e hora que a carga foi entregue.

Processo: O sistema insere a informação no banco de dados.

Saída: Mensagem de sucesso, senão, mensagem de erro.

3.16. RF016: Alterar dados cadastrais

Descrição: O usuário pode alterar os dados pessoais cadastrados no sistema.

Entrada: Preenchimento dos campos referentes às informações a serem alteradas.

Processo: O sistema atualiza as informações no banco de dados.

Saída: Mensagem de sucesso, senão, mensagem de erro.

4. Requisitos Não Funcionais

4.1. RNF001: Software

O SGBD utilizado será o PostgreSQL, por ser um gerenciador seguro e *open-source*.

4.2. RNF002: Linguagem de programação

O sistema será construído baseado nas linguagens de programação Java e Java Script. No qual são conhecidas no mercado para desenvolvimento de software.

5. Atributos

5.1. Disponibilidade

O sistema deve estar sempre disponível, caso ocorra alguma interrupção ele deve ser restaurado o mais rápido possível.

5.2. Segurança

O sistema deverá ser o mais seguro possível para que pessoas não autorizadas sejam impossibilitadas de acessar as informações dos usuários.

Agradecimentos

Agradecemos ao professor orientador Hélder Line Oliveira, que nos orientou ao longo de todo o trabalho de forma muito clara e objetiva e ao professor Sebastião Portela que, no papel de responsável pela disciplina de TCC, foi muito solícito quanto a dúvidas sobre o trabalho. Mesmo com as dificuldades e limitações das orientações à distância, esses dois estiveram sempre dispostos a nos ajudar.

Agradecemos também aos professores que compareceram na qualificação do trabalho e fizeram suas sugestões de melhorias que poderiam ser feitas.

Por fim, agradecemos a todos que de alguma forma contribuíram para o bom êxito deste trabalho. Em especial aos professores das demais disciplinas cursadas por nós no mesmo período de realização do trabalho, que foram compreensíveis ao passar os trabalhos menos extensos a fim de permitir que pudéssemos manter nossa concentração e esforços mais direcionados ao TCC.