UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS SOBRAL

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

ABRAÃO COSTA RODRIGUES LIMA

VIAJÁ: APLICAÇÃO WEB PARA CONSULTAR DISPONIBILIDADE DE HORÁRIOS E VALORES DE TRANSPORTES ALTERNATIVOS

SOBRAL-CE

2023

ABRAÃO COSTA RODRIGUES LIMA

VIAJÁ: APLICAÇÃO WEB PARA CONSULTAR DISPONIBILIDADE DE HORÁRIOS E VALORES DE TRANSPORTES ALTERNATIVOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Iális Cavalcante de Paula Júnior

SOBRAL-CE

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

L696v Lima, Abraão Costa Rodrigues.
VIAJÁ : APLICAÇÃO WEB PARA CONSULTAR DISPONIBILIDADE DE HORÁRIOS E VALORES DE TRANSPORTES ALTERNATIVOS / Abraão Costa Rodrigues Lima. – 2023.
20 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Sobral, Curso de Engenharia da Computação, Sobral, 2023.
Orientação: Prof. Dr. Iális Cavalcante de Paula Júnior.

1. engenharia de software. 2. desenvolvimento Web. 3. System Usability Scale. 4. topiques. 5. Ceará. I. Título.

CDD 621.39

ABRAÃO COSTA RODRIGUES LIMA

VIAJÁ: APLICAÇÃO WEB PARA CONSULTAR DISPONIBILIDADE DE HORÁRIOS E VALORES DE TRANSPORTES ALTERNATIVOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Campus Sobral da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Aprovada em: __/__/____.

BANCA EXAMINADORA

_____
Prof. Dr. Ialis Cavalcante de Paula Júnior
(Orientador)
Universidade Federal do Ceará (UFC)

_____
Prof. Me. Erick Aguiar Donato
Universidade Federal do Ceará (UFC)

_____
Prof. Me. Fernando Rodrigues de Almeida Júnior
Universidade Federal do Ceará (UFC)

Aos meus pais, Fátima e Lidomar.

AGRADECIMENTOS

À minha família, por todo o apoio durante minha vida e motivação para que conquistasse meus objetivos.

À minha namorada Jardielen Chaves, pelo seu apoio constante e incondicional. Sem sua ajuda, esse projeto não teria sido concluído.

Aos meus amigos, pela disposição de ajudar de todas as formas possíveis.

Ao Prof. Dr. Iális Cavalcante de Paula Júnior, pela paciência, apoio e excelente orientação.

Aos professores participantes da banca examinadora Erick Aguiar Donato e Fernando Rodrigues de Almeida Júnior pelo tempo, pelas valiosas colaborações e sugestões.

RESUMO

No Ceará, um dos meios mais comuns de transporte rodoviário são as topiques, nome comum das vans que executam o transporte intermunicipal de passageiros de forma alternativa aos ônibus. As vans, divididas em 27 cooperativas, juntas transportavam cerca de um milhão e 200 mil passageiros por mês já no ano de 2010. Entretanto, mesmo popularizado, o serviço de topiques não possui presença digital formalizada, disponibilizando horários e rotas disponíveis por postagens ou perfis de terceiros. Desta forma, definimos o escopo deste trabalho: desenvolver uma plataforma Web confiável e de fácil acesso que ofereça informações atualizadas sobre horários e valores de topiques. Foi necessário definir os requisitos para a aplicação assim como recolher dados sobre as rotas de topiques atuais. Durante a execução do trabalho, foi realizada engenharia de requisitos através de pesquisas com usuários para documentação das necessidades da aplicação. Para coletar as informações sobre as rotas, foram realizadas múltiplas entrevistas com motoristas na cidade de Sobral, cidade considerada polo econômico da região norte do Estado, agregando topiques que circulam em toda a região. O desenvolvimento foi seguido pela estruturação de um banco de dados e o desenvolvimento de uma API, assim como uma interface Web. Após o desenvolvimento e implantação, foi realizada a validação da aplicação utilizando o *System Usability Scale*(SUS) para avaliação, resultando em uma boa pontuação e, portanto, no sucesso do projeto.

Palavras-chave: engenharia de software; desenvolvimento Web; System Usability Scale; topiques; Ceará.

ABSTRACT

In Ceará, one of the most common means of road transportation is "topiques," which is the common name for vans that provide passenger transport between cities as an alternative to buses. The vans, divided into 27 cooperatives, collectively transported approximately one million and 200 thousand passengers per month as early as 2010. However, even though it became popular, topiques service lacks a formal digital presence, with schedules and routes being made available through third-party posts or profiles. Therefore, we defined the scope of this project: to develop a reliable and easily accessible Web platform that offers up-to-date information on topiques schedules and fares. It was necessary to define the application's requirements and gather data on current topiques routes. During the execution of the project, requirements engineering was performed through user research to register the application's needs. To collect information about the routes, multiple interviews were conducted with drivers in the city of Sobral, considered an economic hub in the northern region of the state, encompassing topiques that operate throughout the region. The development process involved the structuring of a database, the creation of an API, and the design of a Web interface. After development and implementation, the application was validated using the System Usability Scale (SUS) for evaluation, resulting in a good score and thus the success of the project.

Keywords: software engineering; Web development; System Usability Scale, topiques, Ceará.

LISTA DE FIGURAS

Figura 1 – Gráfico de pontuação SUS ….…..…… 22
Figura 2 – Diagrama ER das rotas no banco de dados…….. 26
Figura 3 - Endpoints do controlador de Rota…… 27
Figura 4 - Endpoints do controlador de Cidade…… 28
Figura 5 - Menu lateral do administrador……. 29
Figura 6 - Página inicial do administrador…… 30
Figura 7 - Listagem de rotas…….. 30
Figura 8 - Paradas de uma rota…….. 31
Figura 9 - Horários de uma rota…… 31
Figura 10 - Preços dos trajetos de uma rota…… 32
Figura 11 - Modal de alteração de distância e valores do trajeto…… 32
Figura 12 - Tela inicial do Viajá ……. 33
Figura 13 - Exemplo de resultado de consulta, card de horários disponíveis……. 33
Figura 14 - Exemplo de exibição de valor da passagem …… 34

LISTA DE ABREVIATURAS E SIGLAS

IPECE	Instituto de Pesquisa e Estratégia Econômica do Ceará
IBGE	Instituto Brasileiro de Geografia e Estatística
STIP-CE	Sistema de Transporte Rodoviário Intermunicipal de Passageiros do Estado do Ceará
ARCE	Agência Reguladora de Serviços Públicos Delegados do Estado do Ceará
VUP	Veículo utilitário de passageiro
VUM	Veículo utilitário misto
SEINFRA	Secretaria da Infraestrutura do Estado do Ceará
SUS	System usability scale
UFC	Universidade Federal do Ceará
MVC	Model-View-Controller
GUI	Guided user interface
CLI	Common Language Interface
IL	Intermediate Language
SQL	Structured Query Language
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
AWS	Amazon Web Services
RDS	Relational Database Service
SGBD	Sistema Gerenciador de Banco de Dados
EC2	Elastic Compute Cloud

SUMÁRIO

1	**INTRODUÇÃO**	14
1.1	***Justificativa***	16
1.2	***Objetivos***	16
1.2.1	***Objetivo geral***	16
1.2.2	***Objetivos específicos***	16
2	**FUNDAMENTAÇÃO TEÓRICA**	18
2.1	***Transporte alternativo de passageiros***	18
2.2	***Engenharia de requisitos***	18
2.3	***Tecnologias e métodos utilizados***	19
2.3.1	***Arquitetura Model-View-Controller***	19
2.3.2	***C#***	20
2.3.3	***Bancos de dados relacionais***	21
2.3.4	***PostgreSQL***	21
2.4	***System Usability Scale***	21
3	**METODOLOGIA**	23
3.1	***Pesquisa de mercado***	23
3.2	***Especificação de Requisitos***	24
3.2.1	***Requisitos funcionais***	24
3.2.2	***Requisitos não funcionais***	24
3.3	***Implementação***	25
3.3.1	***Banco de dados***	25
3.3.2	***Back-end***	26
3.3.3	***Front-end***	28
3.3.3.1	***Administrador***	29
3.3.3.2	***Usuário***	32
3.4	***Implantação***	34
3.5	***Avaliação de usabilidade***	35
4	**RESULTADOS E DISCUSSÃO**	36
4.1	***A pontuação SUS***	36

4.2 ***Discussão sobre respostas do formulário***..... 36
4.3 ***Pontos de melhoria***…..... 37
5 **CONCLUSÃO**…..……... 38

1 INTRODUÇÃO

No Brasil, o método predominante de deslocamento é o transporte rodoviário de passageiros, graças à sua consistência nos serviços prestados, ampla cobertura e tarifas acessíveis. Nesse contexto, o transporte rodoviário assume um papel crucial no movimento da população, desempenhando também um papel fundamental na estruturação econômica do país (FREITAS, 2011).

O Estado do Ceará é constituído por 189 municípios. Segundo o mais recente censo do IBGE em 2022, a população estadual aproxima-se de 8,792 milhões de habitantes, colocando-o atualmente na oitava posição no ranking de população brasileira.

Em 2006, visando unificar as definições das macrorregiões cearenses e simplificar o planejamento das ações governamentais em diversos setores, o Instituto de Pesquisa e Estratégia Econômica do Ceará (IPECE) propôs uma revisão da regionalização. Como desdobramento desse processo, os seguintes municípios foram designados como polos regionais: Itapipoca, Camocim, Paracuru, Sobral, Crateús, Tauá, Quixadá, Quixeramobim, Canindé, Baturité, Redenção, Cascavel, Aracati, Russas, Limoeiro do Norte, Jaguaribe, Crato, Juazeiro do Norte e Barbalha (IPECE, 2015).

Posteriormente, em 2009, o Governo do Estado lançou o edital de licitação do Sistema de Transporte Rodoviário Intermunicipal de Passageiros do Estado do Ceará (STIP-CE), introduzindo uma nova proposta de municípios polos. Esses municípios polos foram oficialmente designados para o transporte de passageiros entre os municípios cearenses e são: Aracati, Baturité, Canindé, Crateús, Crato, Iguatu, Itapipoca, Juazeiro do Norte, Limoeiro do Norte, Morada Nova, Quixadá, Russas, Sobral e Tauá (AGUIAR, 2015).

De acordo com a Agência Reguladora de Serviços Públicos Delegados do Estado do Ceará - ARCE (2019), o transporte rodoviário intermunicipal de passageiros no Ceará divide-se em dois serviços independentes: o serviço metropolitano e o serviço interurbano. Cada serviço possui duas categorias: o serviço regular, operado por ônibus, e o serviço regular complementar, operado por vans, micro-ônibus e/ou mini-ônibus, chamados popularmente de topiques. No caso do serviço regular complementar, ele é prestado por cooperativas ou motoristas autônomos associados a cooperativas de transporte de passageiros, utilizando

veículos utilitários de passageiros (VUP), veículos utilitários mistos (VUM), micro-ônibus ou mini-ônibus.

Segundo a SEINFRA, em 2010 o Ceará contava com 740 vans que operavam em 164 linhas que passavam por 139 municípios cearenses. Os dados são da época em que o transporte de passageiros foi regularizado pelo então governador do estado, Cid Gomes. Nesse período, por mês, as 27 cooperativas tinham capacidade de transportar um milhão e 200 mil passageiros (CEARÁ, 2010).

Para ilustrar o alcance do transporte por topiques no estado do Ceará, em Fortaleza, cerca de 58% dos passageiros que utilizam a malha rodoviária, fazem uso de topiques com alta ou baixa frequência, enquanto apenas 16,4% não utilizam de forma alguma (BARBOSA DE ALBUQUERQUE, 2013). No entanto, mesmo com tamanha abrangência do serviço, ainda é difícil encontrar informações sobre as rotas de transporte intermunicipal complementar em plataformas centralizadas.

Nos dias de hoje, a presença digital é um fator essencial para a manutenção e crescimento de um serviço. As gerações atuais, mais experientes com tecnologias, entendem que uma empresa que não possui uma plataforma online de qualquer natureza não existe ou não passa credibilidade (DANIEL; MUNOZ, 2016).

Diante disso, esse projeto consiste no desenvolvimento de uma plataforma Web que pode ser acessada por computadores ou aparelhos móveis com acesso à Internet que disponibilize, de acordo com a necessidade do usuário, rotas de topique disponíveis, com valores, horários e locais de embarque de acordo com a cidade da qual deseja partir e a cidade na qual deseja chegar.

O objetivo final do projeto foi a validação da usabilidade da solução, conduzida através da aplicação de um questionário usando o *System Usability Scale*(SUS) (BROOKE, 1995). Com a aplicação do questionário, foi possível reconhecer forças e fraquezas da solução e elaborar os futuros desenvolvimentos para que possa se tornar um produto mais eficaz.

1.1 Justificativa

O desenvolvimento de uma plataforma que centralize os dados sobre rotas de topiques disponíveis é parte da evolução natural do serviço. Quando se observa outros setores do transporte coletivo público e privado, ambos já desfrutam de plataformas que facilitam a viagem dos passageiros disponibilizando informações. Empresas de transporte intermunicipal por ônibus como a Guanabara facilitam a compra de passagens e a consulta de horários através de plataformas online. Iniciativas como a plataforma Moovit, com o foco na mobilidade dentro dos centros urbanos ajudam passageiros com necessidade de obter informações do transporte público a chegar em seus destinos com informações sobre paradas e horários de ônibus.

Estar no ambiente digital torna um serviço mais atrativo para as novas gerações que estão mais familiarizadas e preferem o acesso à informação dessa forma. Uma pesquisa feita online com residentes da região norte do Ceará, em sua maioria estudantes da UFC, mostrou que o surgimento de uma plataforma assim é desejada e ajudaria a sanar problemas que as pessoas apontam sobre o serviço de topiques.

Por último, quando pensamos pela ótica dos passageiros, isso seria uma melhora na qualidade de vida e facilitaria o planejamento de viagens para lazer ou trabalho, gerando um maior fluxo econômico entre as cidades que são conectadas pelas cooperativas.

1.2 Objetivos

1.2.1 Objetivo geral

O objetivo geral do trabalho é desenvolver uma aplicação Web eficiente, confiável e de fácil acesso que atenda às necessidades dos usuários, oferecendo informações atualizadas sobre horários e valores de topiques, contribuindo para uma experiência de viagem mais conveniente e organizada.

1.2.1 Objetivos específicos

Os objetivos específicos definidos para esse projeto, visando alcançar o objetivo geral são:

- Realizar uma pesquisa com usuários de topiques a fim de identificar as principais

necessidades dos usuários, as funcionalidades essenciais e as características técnicas que devem ser consideradas durante o desenvolvimento;

- Criar uma base de dados inicial contendo horários e rotas para serem utilizados nas pesquisas.
- Realizar o levantamento de requisitos do sistema que será implementado;
- Codificar o Sistema de Gerenciamento de horários e valores das topiques disponíveis.
- Validar a usabilidade da plataforma através de pesquisa com usuários.

2 FUNDAMENTAÇÃO TEÓRICA

Essa seção apresenta conceitos e características utilizados no desenvolvimento da aplicação assim como também as ferramentas escolhidas que utilizam desses conceitos. As descrições das ferramentas são adaptações de suas documentações oficiais, disponibilizadas pelas empresas responsáveis por seu desenvolvimento.

2.1 Transporte alternativo de passageiros

O transporte intermunicipal alternativo de passageiros tem sua definição, segundo a ARCE (2023), como um Serviço Regular Interurbano Complementar. Essa definição abrange o transporte de passageiros, realizado com Miniônibus, Micro-ônibus, Veículo Utilitário de Passageiros (VUP) ou Veículo Utilitário Misto (VUM), entre municípios fora da região metropolitana de Fortaleza. Os veículos utilizados para realizarem esse serviço são conhecidos, informalmente no estado do Ceará, como topiques.

2.2 Engenharia de requisitos

Como descrito por Sommerville (2007), os requisitos de um software podem ser funcionais ou não funcionais. Os requisitos funcionais definem o que o sistema deve fazer, de maneira objetiva, enquanto os requisitos não funcionais descrevem restrições ou exigências que o sistema deve cumprir.

Os requisitos não funcionais podem ser separados em: requisitos de produto, requisitos organizacionais e requisitos externos. Dentro dessas três categorias, outros aspectos do software serão definidos, como confiabilidade, eficiência, usabilidade, aspectos legais e proteção dos dados (SOMMERVILLE, 2007).

Os requisitos de produto definem o comportamento do *software*, neles estão descritos os recursos necessários para que a solução cumpra os requisitos de confiabilidade e eficiência também descritos neles. Um exemplo de um requisito de confiabilidade de uma aplicação seria de um aplicativo de *internet banking*, que deve estar disponível vinte e quatro horas por dia, com o mínimo de interrupções possíveis. Isso poderia levar a equipe de engenharia a desenhar uma solução com um sistema que utiliza múltiplos servidores que podem compensar falhas em um servidor único (SOMMERVILLE, 2007).

Nos requisitos organizacionais podemos incluir a forma com a qual será guiado o

processo de desenvolvimento, os recursos e a linguagem de programação utilizada (SOMMERVILLE, 2007).

2.3 Tecnologias e métodos utilizados

Uma parte importante do desenvolvimento de qualquer solução de *software* é a escolha das tecnologias e metodologias utilizadas na estruturação da aplicação. Para o projeto descrito aqui, foram escolhidas tecnologias que se adequavam a pontos essenciais do projeto. Pontos esses que são: o tamanho reduzido da equipe, o tempo de desenvolvimento diminuto e a familiaridade com as ferramentas, uma vez que o escopo do projeto e o tempo disponível não permitiriam o desenvolvimento consistente da aplicação junto do aprendizado de novas linguagens ou arquiteturas.

2.3.1 Arquitetura Model-View-Controller

O Model-View-Controller (MVC) é uma arquitetura de software idealizada inicialmente para criação de interfaces guiadas de usuário ou *guided user interfaces* (GUI) mas é popularmente utilizado nos dias de hoje na programação Web (HOPKINS, 2013). O MVC data de 1979, criado por Trygve Reenskaug enquanto projetava o Smalltalk-79, uma linguagem de programação orientada a objetos utilizada para o ensino de programação (WIKIPÉDIA, 2023).

No padrão MVC, a aplicação possui três camadas essenciais que separam a regra de negócio da manipulação dos dados e da exibição dos mesmos ao usuário final. Essas três camadas são: a camada de modelo (*Model*), a camada de visualização (*View*) e a camada de controle (*Controller*) (HOPKINS, 2013).

A primeira camada descrita é a camada de *Model.* Nela está definida a lógica da aplicação, assim como é nessa camada que é feita a comunicação entre as camadas de *Controller* e *View*. Na camada de modelo está definido também a modelagem dos dados da aplicação e todas as funções que os manipulam. As funções de manipulação definida no modelo não podem ser acessadas por outras camadas, sendo possível apenas a chamada delas pela camada de controle (DEACON, 2009).

Exibindo os dados gerenciados pela primeira camada, temos a camada de *View*, no qual os dados são interpretados e organizados para a exibição para o usuário. Uma *View* pode ser uma tabela, um gráfico ou um relatório. Qualquer componente que exibe os dados é

considerado uma visão, então não é necessário limitarmos a exibições estruturadas como citadas acima (GAMMA, 1998). Na interpretação atual do MVC, em aplicações Web, a página HTML exerce a função da *View*.

Por último, temos a terceira camada, que seria a camada de controle, responsável por receber requisições do usuário e redirecionar de acordo com a necessidade. O controlador é quem decide qual modelo deve ser invocado para executar uma regra de negócio ou qual visão deve ser exibida ao usuário. Em uma aplicação Web, por exemplo, o *controller* seria o responsável por interpretar cliques nos botões e selecionar visões de acordo com a necessidade do usuário (MASSARI, 2017).

2.3.2 C#

O C# (lê-se *C-Sharp*) é uma linguagem de programação orientada a objetos, desenvolvida e mantida pela Microsoft. A linguagem é fortemente tipada e possui características únicas que favorecem o desenvolvimento de aplicações robustas ou com muito potencial de crescimento (BILLWAGNER, 2023).

Dentre essas características estão a coleta de lixo que gerencia o alocamento de memória pelo desenvolvedor, recuperando o espaço de objetos cujos ciclos de vida já foram encerrados (GEWARREN, 2023). A linguagem permite a definição de tipos anuláveis, protegendo os objetos de variáveis sem valor alocado e também conta com um tratamento de exceções estruturado e extensível (BILLWAGNER, 2023).

A plataforma de desenvolvimento do C# é conhecida como .NET (lê-se dotNet), ela é um conjunto de bibliotecas combinados em um sistema de execução virtual baseado em uma implementação própria da Microsoft do CLI (*Common Language Interface*). Toda aplicação em C# é compilada inicialmente em IL (do inglês, Linguagem Intermediária) que está de acordo com o CLI e então é interpretada para instruções nativas de máquina (BILLWAGNER, 2023).

.NET tem um grande foco em compatibilidade e versatilidade. Ela possui recursos que aperfeiçoam o desenvolvimento de aplicações diversas como código assíncrono, delegados, exceções, tipos genéricos e outros que auxiliam desenvolvedores (GEWARREN, 2023). Essas facilidades são utilizadas em frentes variadas de desenvolvimento com diferentes produtos, desde o desenvolvimento para aplicações desktop, aplicações Web usando o *framework* ASP.NET, utilizado no projeto descrito nesse trabalho; aplicações mobile usando a biblioteca

Xamarin ou até mesmo jogos, usando o Unity Framework (BILLWAGNER, 2023).

2.3.3 Bancos de dados relacionais

Ramez Elmasri e Navathe (2005) descrevem um banco de dados relacional como uma coleção de relações na qual cada relação pode ser representada coloquialmente como uma "tabela" que contém uma coleção de registros. Cada registro de uma tabela, segundo Ramez Elmasri e Navathe (2005), representa uma entidade do mundo real e cada coluna auxilia, descrevendo um aspecto daquela entidade. Usando a terminologia correta, cada registro de uma tabela pode ser chamado de tupla, enquanto cada coluna é definida como um atributo. É importante lembrar que no modelo relacional, todos os valores sob um mesmo atributo devem ser do mesmo tipo.

2.3.4 PostgreSQL

PostgreSQL é um banco de dados relacional, de código aberto e que usa a linguagem SQL para definir suas estruturas e gerenciar seus registros. O sistema foi lançado em 1986 e desde 2001 tem comprovada eficácia ACID, que é um conjunto de características para transações em bancos de dados que garantem atomicidade, consistência, isolamento e durabilidade (POSTGRESQL, 2023).

O PostgreSQL tenta manter conformidade com o núcleo da linguagem SQL padrão, atualmente apresentando 170 das 179 funções declaradas no SQL Core de 2016. Todas as funcionalidades mais conhecidas são implementadas com sintaxe igual ou semelhante àquela do SQL como conhecemos (POSTGRESQL, 2023).

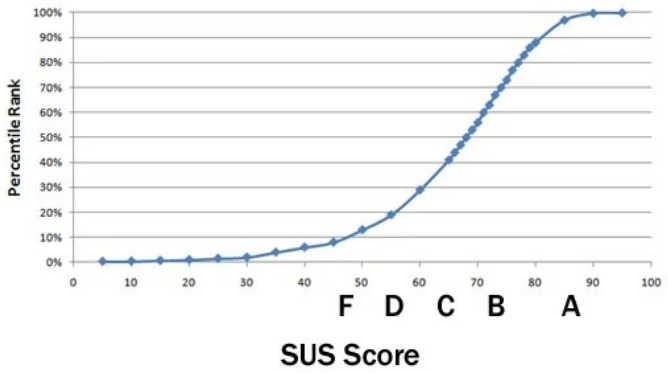
2.4 System Usability Scale (SUS)

Segundo Brooke (1995), usabilidade não pode ser mensurada em nenhuma forma real ou claramente quantificada. A usabilidade pode ser medida apenas como um nível de conformidade com o propósito de uma solução. De tal forma, a usabilidade de uma aplicação deve ser medida de acordo com o contexto no qual será utilizada e sua conformidade ao contexto (BROOKE, 1995).

Pensando nisso, John Brooke idealizou, em 1986, o SUS(*System Usability Scale*), um questionário simples e rápido de perguntas e respostas que pode avaliar diversos produtos,

sejam eles de tecnologia ou não. O SUS consiste em dez perguntas que tem respostas que variam dentro de cinco níveis que vão de “discordo fortemente” a “concordo fortemente”. Uma vez que o questionário tenha sido respondido, o avaliador fará um cálculo de acordo com as respostas e assim calcular a pontuação SUS (TEIXEIRA, 2016).

Figura 1: Gráfico de pontuação SUS



Fonte: (TEIXEIRA, 2016)

A Figura 1 descreve o gráfico de classificação da pontuação SUS obtida após calcular a média das pontuações de cada usuário que respondeu o questionário. As letras representam notas da escala de avaliação por letras para melhor visualização do que seriam resultados considerados aceitáveis ou não, partindo da pior nota representada por F e progredindo gradativamente até o A que seria a melhor nota. Exemplificando, uma nota abaixo de 50 seria o equivalente a uma nota F, considerada ruim na graduação com letras, enquanto a melhor nota, no caso, seria a nota A, com notas próximas ou acima de 85.

De modo a gerar dados mais próximos da realidade possível, o usuário deve responder o questionário de forma natural, sem dedicar muito tempo a cada pergunta, assim como também não deve ser entrevistado ou ter discutido acerca do produto de qualquer forma. Quando um usuário não souber o que responder a uma questão, ele deverá escolher a opção central, marcando o número três na escala (BROOKE, 1995).

Para que um produto seja considerado “viável”, deve ter sua pontuação maior ou igual a 68, representada na Figura 1 como a nota C, um resultado mediano. Quando um produto tem uma nota menor que essa, é percebida a necessidade de maior investimento no design do produto para que sua usabilidade tenha seus defeitos sanados. É importante salientar que o SUS é um sistema generalista e quantitativo, por isso ele deve ser combinado com outras avaliações para que a definição específica dos problemas do produto seja formulada (TEIXEIRA, 2016).

3 METODOLOGIA

Na seção de metodologia, será descrito o processo de desenvolvimento da solução, detalhando o passo a passo, partindo da pesquisa de mercado, para a especificação de requisitos, depois para o desenho da solução e por último, sua implementação. Na área de desenvolvimento de software, o ato de documentar os processos é responsável por manter uma solução acessível a novos desenvolvedores, assim como diminuir o custo de manutenção de funcionalidades implementadas.

3.1 Pesquisa de mercado

Quando pensamos no usuário de transporte alternativo, podemos imaginar duas personas. A primeira é o usuário de rotina, que mora em uma cidade distante da sua atividade principal, como trabalho ou estudo, enquanto a segunda persona seria o usuário casual. O usuário casual não utiliza o serviço sempre e nem sempre faz a mesma rota.

Para compreender os hábitos comportamentais do público-alvo da aplicação, foi aplicado um formulário eletrônico através do *Google Forms*, no qual o usuário respondia sobre a cidade na qual reside, sua frequência no uso das topiques (ou se já utilizou uma topique de qualquer forma), assim como uma pergunta objetiva sobre a utilidade de uma plataforma de verificação de horários, valores e lugares das topiques.

Todas as respostas do formulário vieram de moradores da macrorregião econômica de Sobral. Dentre as respostas, 97% afirmou que utiliza ou já utilizou o serviço, enquanto cerca de 39% afirmou que utiliza as topiques com uma frequência semanal (1 ou mais vezes na semana). Alguns usuários usaram o campo de resposta da frequência para numerar problemas com o sistema atual das topiques, destacando o acesso limitado às informações das topiques.

Para o usuário casual, o processo de verificar disponibilidade de horários, locais de embarque e desembarque, assim como os valores das passagens, pode ser oneroso e cansativo. As cooperativas não possuem plataformas online que disponibilizam os horários, sejam aplicativos, sites ou páginas de redes sociais.

Existem perfis de notícias locais nas redes sociais que postam, esporadicamente, os horários que contemplam uma rota, mas não são confiáveis na questão de quão atualizados estão os horários ou se há apenas aquela rota na cidade.

O mais próximo de um produto semelhante ao proposto no projeto é o site “Vou de topic” que dispõe de horários de rotas que passam pela cidade de Mucambo. A plataforma

oferece, gratuitamente, horários e locais de embarque e desembarque, assim como sinalização de horários diferentes em fins de semana e feriados.

O diferencial do aplicativo proposto no projeto será o número de cidades abrangidas, assim como a disponibilidade do valor das passagens entre as cidades.

3.2 Especificação de Requisitos

Os requisitos da solução proposta no trabalho serão definidos inicialmente com o objetivo principal dela. Em seguida, serão descritos seus requisitos funcionais, descrevendo suas funcionalidades necessárias. A última parte descreverá os requisitos não funcionais da aplicação, na qual descreveremos exigências que a aplicação deve cumprir para satisfazer aspectos que serão especificados.

3.2.1 Requisitos funcionais

O objetivo principal do trabalho é criar uma aplicação que permita a um passageiro consultar os horários, locais de partida e valores de topiques, de acordo com a rota que deseja fazer.

Dessa forma, o usuário administrador do sistema terá que realizar:

- Gerenciamento de rotas das topiques.
- Gerenciamento de horários das topiques.
- Gerenciamento do custo das passagens das topiques.

Enquanto o usuário passageiro terá que realizar, dados origem e destino da viagem:

- Consulta de horários das viagens.
- Consulta de lugar de partida das viagens.
- Consulta de local de chegada das viagens.
- Consulta de valor das viagens.

3.2.2 Requisitos não funcionais

Para os requisitos não funcionais da aplicação, foram levados em consideração os aspectos de segurança, usabilidade, escalabilidade e manutenibilidade. No quesito de segurança temos que o sistema deve garantir segurança dos dados dos administradores e das rotas cadastradas para evitar alterações não desejadas ou homologadas.

Sobre usabilidade, ficou definido que o sistema deve ter um uso intuitivo e simplificado por parte dos passageiros, fluído por parte dos administradores, de forma a não atrapalhar no processo de criação ou alteração de uma rota já existente. Ainda na usabilidade, a aplicação deve poder ser utilizada em dispositivos móveis diversos com diferentes tamanhos de tela, assim como computadores ou laptops.

Analisando o quesito de escalabilidade, a estrutura do banco de dados e do servidor deve permitir um grande número de consultas simultâneas e um grande volume de dados de rotas sem perder performance ou integridade dos dados.

Quanto a manutenibilidade, cada função da aplicação deve ser desenvolvida com o máximo de modularidade e o mínimo de acoplamento para facilitar a implementação de testes unitários ou de integração. O baixo acoplamento permitirá a implementação de novas funcionalidades com maior segurança de não comprometer funcionalidades já existentes.

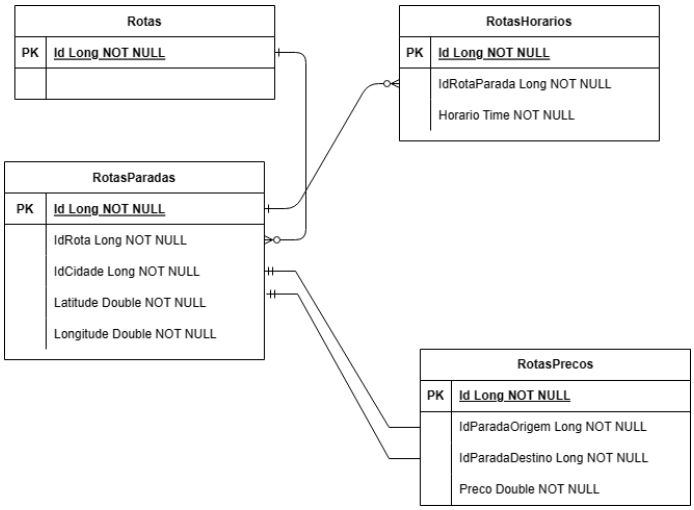
3.3 Implementação

O projeto foi implementado em três estruturas principais, sendo elas: a estrutura do banco de dados no PostgreSQL, o *back-end* com a API que vai consultar os dados em C# usando *framework* ASP.NET e o *front-end* usando o padrão MVC e o *framework* ASP.NET.

3.3.1 Banco de dados

Para o banco de dados, foi necessário abstrair cada aspecto de uma rota executada por uma topique. Pensando nessa abstração, foram idealizadas quatro tabelas: Rotas, Paradas, Horários e Preços. Inicialmente, foi idealizado que a tabela de Rotas unificasse todas as outras com referências de chaves estrangeiras a ela, entretanto durante o desenho da solução, as relações ficaram mais distribuídas.

Figura 2: Diagrama ER das rotas no banco de dados.



Fonte: Elaborada pelo autor.

Na Figura 2, podemos ver uma parte resumida das relações das tabelas e a estrutura das rotas no banco de dados. A tabela de rotas serve apenas como um agregador de paradas, enquanto as paradas representam as coordenadas dentro das cidades nas quais as topiques vão parar para o embarque dos passageiros.

Os horários, ligados sempre a uma parada, representam os horários estimados nos quais o carro está saindo da parada. São armazenados os valores das viagens, levando em consideração apenas cidade de origem e a cidade destino, uma vez que as cooperativas entrevistadas não fazem distinção de serviço de acordo com os atributos do carro, mas sim de acordo com a distância viajada. No banco de dados também existem tabelas que armazenam as informações dos usuários administradores e as informações das cidades como nome e Estado.

3.3.2 Back-end

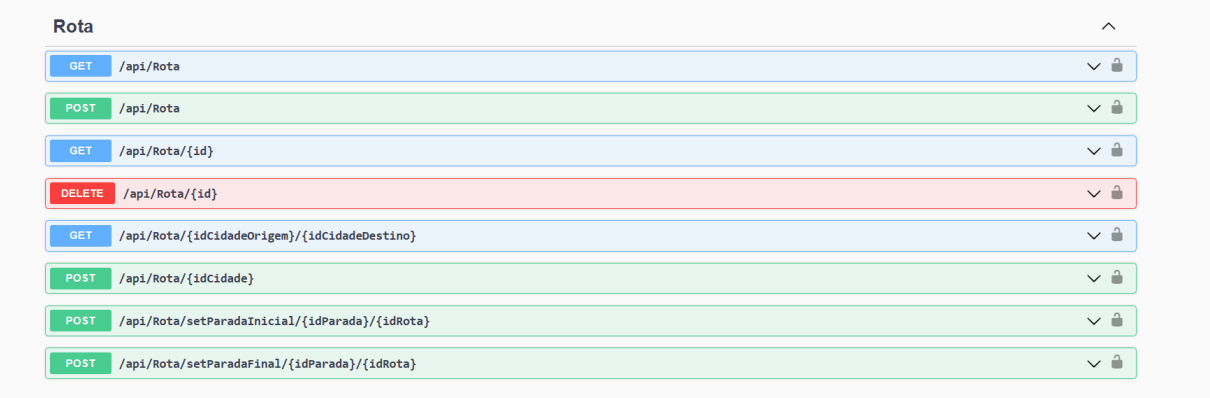
No *back-end*, temos duas partes da arquitetura MVC, sendo eles a camada de modelo, com a regra de negócio da solução, assim como a camada de controle, responsável por redirecionar os comandos das *views* para os devidos modelos, para que então ocorra a manipulação ou recuperação dos dados.

O *back-end* da aplicação foi desenvolvido no intervalo de três meses, trabalhando cerca de 15 horas semanais no formato de uma API REST, uma aplicação que serve de ponte entre o cliente no navegador e os dados do banco de dados. A API vai receber a requisição em um de seus *endpoints*, ou rotas, organizados dentro de controladores no *framework* ASP.NET,

que funcionam na prática como prefixos que separam as rotas entre os modelos. Um *endpoint*, ou uma rota, é um endereço que pode receber requisições HTTP utilizando verbos como GET, POST, PUT ou DELETE para invocar, no caso do projeto atual, uma função de dentro da API.

Para melhor visualização das rotas durante o desenvolvimento, foi utilizado o pacote *Swagger*, capaz de gerar uma interface gráfica Web para visualização e teste dos resultados das requisições. O pacote foi configurado apenas para ambiente de desenvolvimento, não sendo possível o acessar a página de visualização no ambiente de produção (disponível ao usuário final).

Figura 3: *Endpoints* do controlador de Rota



Fonte: Elaborado pelo autor.

A Figura 3 demonstra a representação de um dos controladores na interface gerada pelo *Swagger*. Ao todo, a API possui 7 controladores: User, Authentication, Cidades, Rota, RotaParada, RotaParadaHorarios e RotaPreco. As rotas indicadas em azul, serão requisições HTTP usando o verbo GET, no qual a requisição não altera dado algum, apenas retorna dados. As rotas em verde são requisições que utilizam o verbo POST, no qual o cliente envia dados, através do corpo da requisição ou parâmetros no endereço da requisição, que serão salvos no sistema. A rota em vermelho representa uma requisição que usa o verbo DELETE, no qual o usuário indicará um registro ou dado a ser apagado do sistema.

Os termos entre chaves "{}" são parâmetros da requisição, o usuário altera-os utilizando as entradas do sistema. Os valores enviados nas requisições estão normalmente ocultos ao usuário e não possuem valor semântico para a utilização do sistema.

O controlador *User* possui rotas que gerenciam os usuários, seja para cadastro, edição, remoção ou apenas recuperação dos dados. Junto dele, o controlador *Authentication* é responsável por validar o acesso de um usuário administrador ao sistema. Um usuário padrão do sistema não precisa ser cadastrado, visto que o sistema, em sua ideia atual, não precisa dos

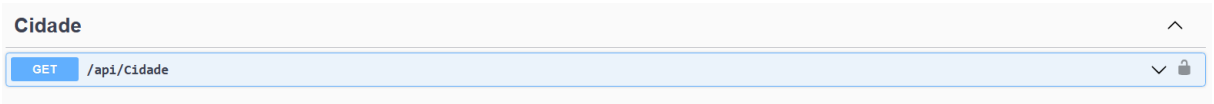
dados de um usuário para que ele efetue as consultas. Entretanto, um administrador precisa ser homologado para que tenha permissão de alterar os dados sobre rotas, sendo seus horários, valores ou paradas. A decisão diz respeito ao requisito de confiabilidade dos dados do sistema.

Falando sobre os controladores de Rota e RotaParada, temos controladores utilizados para gerenciar as duas tabelas mais referenciadas do banco de dados. Como visto na Figura 2, uma rota agrega inúmeras paradas enquanto uma parada agrega inúmeros horários e pode ser referenciada em duplas por registros de preço dos trajetos.

Sob os controladores de RotaParadaHorarios e RotaPreco, estão os *endpoints* responsáveis por dados específicos das rotas das topiques. O controlador de horários pode retornar os horários referentes a apenas uma parada ou uma rota inteira, enquanto o controlador de preços pode ser consultado pelo usuário final na consulta de rotas para consultar o preço de um trajeto ou então o administrador requisita todos os valores de trajeto de uma rota para validá-los ou realizar quaisquer alterações.

O controlador de cidade não permite inserção ou edição de novas cidades, apenas a recuperação dos dados. Para evitar problemas com dados inconsistentes ou redundantes, foi decidido que os registros de cidades seriam atualizados manualmente.

Figura 4: *Endpoints* do controlador de Cidade



Fonte: Elaborado pelo autor.

Podemos perceber na Figura 4 que o controlador Cidade possui apenas um endpoint que usa o método GET para retornar a lista de cidades cadastradas. Inicialmente foram cadastradas apenas cidades da macrorregião de planejamento Sobral-Ibiapaba (SEPLAG-CE, 2023).

3.3.3 Front-end

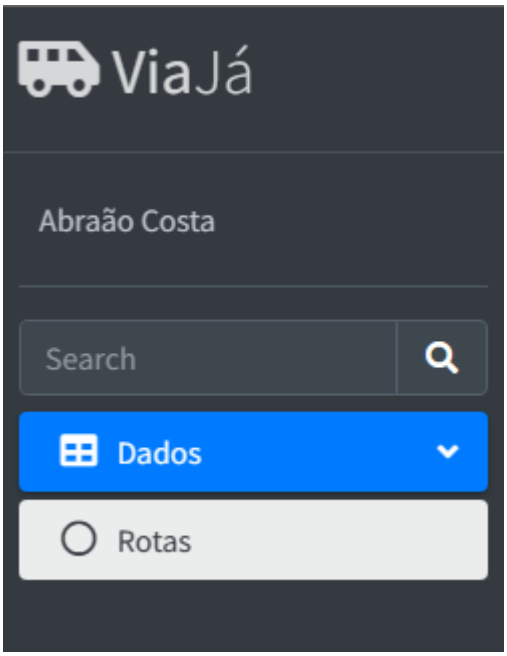
Sobre a interface da solução, podemos dividi-la em dois grupos de *views*, ou telas que foram desenvolvidas paralelamente ao *back-end* no intervalo de três meses, utilizando cerca de 10 horas semanais. O primeiro conjunto é o de telas do administrador, que gerencia o cadastro de rotas, horários e valores. Enquanto isso, o segundo conjunto de telas seria o do usuário final, que pode consultar as rotas e suas informações.

3.3.3.1 Administrador

Para as telas do administrador, foram criados controladores (semelhantes aos controladores do *back-end*) que retornam telas. Seu funcionamento é semelhante aos controladores da API, entretanto todas as chamadas utilizam o verbo GET e retornam um arquivo HTML com a interface que será utilizada. Na arquitetura MVC, teríamos no *front-end* controladores (*controllers*) e visões (*views*) como os nomes dos componentes sugeriram anteriormente.

A organização das telas foi realizada utilizando um *template* que utiliza o *framework CSS Bootstrap*, chamado AdminLTE, constituído por um conjunto de páginas pré marcadas, com classes CSS e funções Javascript pré programadas para que o desenvolvedor ganhe bastante tempo sem precisar desenvolver toda a interface do zero. O AdminLTE é o *template* para sistemas administrativos mais utilizado no mundo, sendo gratuito e de código aberto, ele proporciona uma variedade de personalizações sem perder a facilidade de implementá-lo quase como um produto de prateleira.

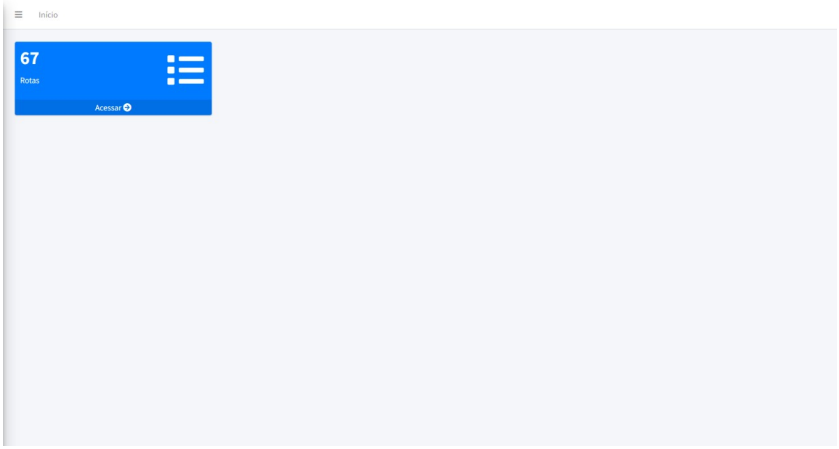
Figura 5: Menu lateral do administrador.



Fonte: Elaborada pelo autor.

No menu das telas do administrador, demonstrado na Figura 5, podemos ver as páginas disponíveis (marcadas pelo ícone circular) dentro de um submenu (marcado pelo ícone personalizado). A estrutura de submenus foi escolhida para acomodar novas funções que podem ser incluídas no futuro. Dessa forma, o teste e a manutenção do sistema poderão ser feitos de forma segura.

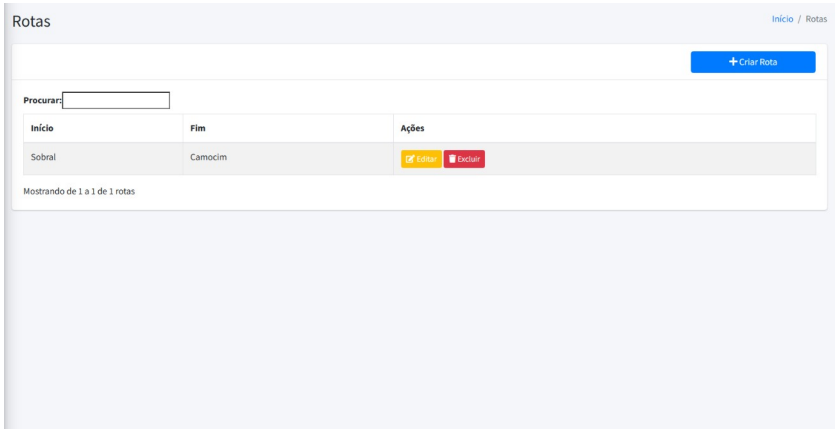
Figura 6: Página inicial do administrador



Fonte: Elaborada pelo autor.

Representada na Figura 6, está a página inicial do administrador. Essa página terá acesso rápido a todas as páginas importantes do sistema. Seguindo o raciocínio do menu, essa estrutura de atalhos pequenos foi pensada para que o sistema seja modular e expansível, com baixo custo de manutenção.

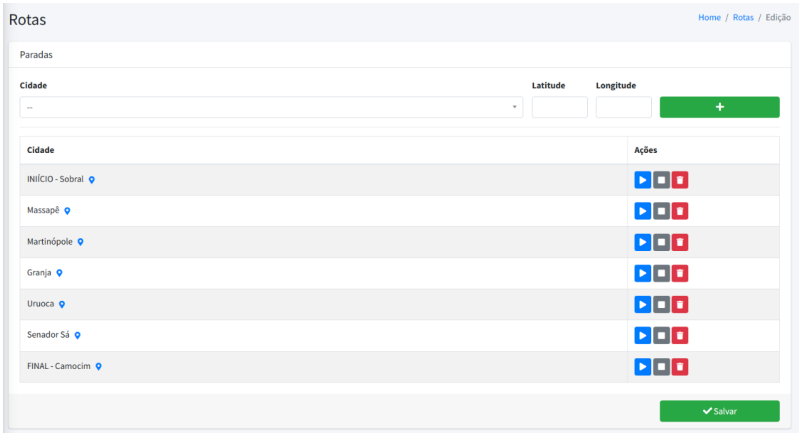
Figura 7: Listagem de rotas



Fonte: Elaborada pelo autor.

Ao acessar a tela de rotas, seja pelo menu ou pelo atalho na tela inicial, o administrador é redirecionado para a tela na Figura 7, que exibe a lista de rotas cadastradas, uma opção para adicionar novas rotas e atalhos para edição ou remoção de rotas existentes.

Figura 8: Paradas de uma rota.

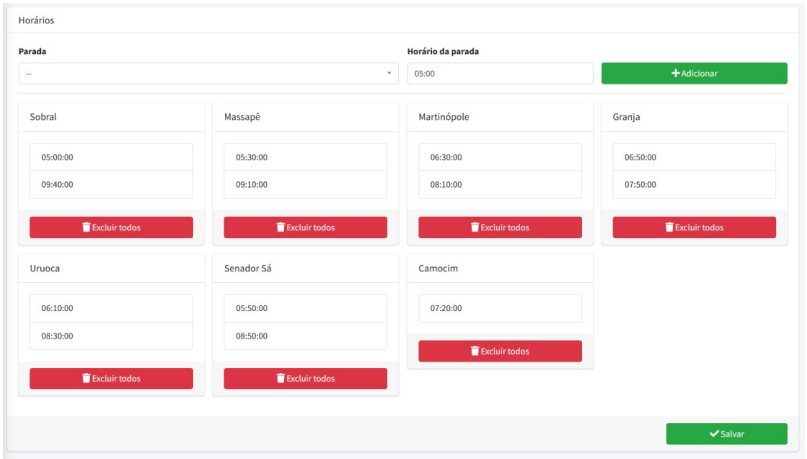


Fonte: Elaborada pelo autor.

Quando o administrador escolhe por editar uma rota, temos uma tela dividida em *cards*, ou cartões, que representam as paradas, os horários e os valores da rota. Na Figura 8, podemos ver o *card* das paradas. Nele, o administrador poderá adicionar ou remover paradas da rota e definir qual será sua parada inicial e sua parada final, na qual a topique fica esperando por novos passageiros, ou, caso não cumpra a lotação, até o próximo horário de viagem.

Atualmente, a plataforma ainda não possui a funcionalidade de escolher uma localidade diretamente no mapa e coletar sua latitude e longitude, portanto ela deve ser inserida manualmente. Tal funcionalidade pode ser implementada como um desenvolvimento futuro.

Figura 9: Horários de uma rota.



Fonte: Elaborada pelo autor.

O segundo *card* da tela representa os horários da rota, no qual o administrador pode adicionar ou remover horários para cada parada, como visto na Figura 9. Para o administrador, cada rota representa o roteiro que um carro fará no dia. Podem existir várias

rotas que passem pela mesma parada mas com horários diferentes entre si. Essa funcionalidade foi idealizada quando se tornou de conhecimento do desenvolvimento que as topiques não seguem horários fixos, mas sim escalas temporárias.

Figura 10: Preços dos trajetos de uma rota.

Valores

-	Sobral	Massapê	Senador Sá	Uruoca	Martinópole	Granja	Camocim
Sobral	-	R$ 0,00	R$ 0,00	R$ 0,00	R$ 0,00	R$ 27,60	R$ 34,35
Massapê	18,5 Km	-	R$ 0,00	R$ 0,00	R$ 0,00	R$ 22,70	R$ 29,45
Senador Sá	46,6 Km	27,9 Km	-	R$ 0,00	R$ 0,00	R$ 15,30	R$ 22,05
Uruoca	58 Km	39,5 Km	11,6 Km	-	R$ 0,00	R$ 12,25	R$ 19,00
Martinópole	83 Km	64,5 Km	36,6 Km	25 Km	-	R$ 5,60	R$ 12,35
Granja	104,2 Km	85,7 Km	57,8 Km	46,2 Km	21,2 Km	-	R$ 6,75
Camocim	129,6 Km	111,1 Km	83,2 Km	71,6 Km	46,6 Km	25,4 Km	-

Fonte: Elaborada pelo autor

Por último, temos o *card* com a representação dos valores dos trajetos que a rota pode fazer. O *card*, representado na Figura 10, é uma representação da matriz exibida dentro das topiques que percorrem uma rota. Quando o administrador clica em uma distância ou valor, pode alterar essa dupla de valores do trajeto.

Figura 11: *Popup* de alteração de distância e valores do trajeto.

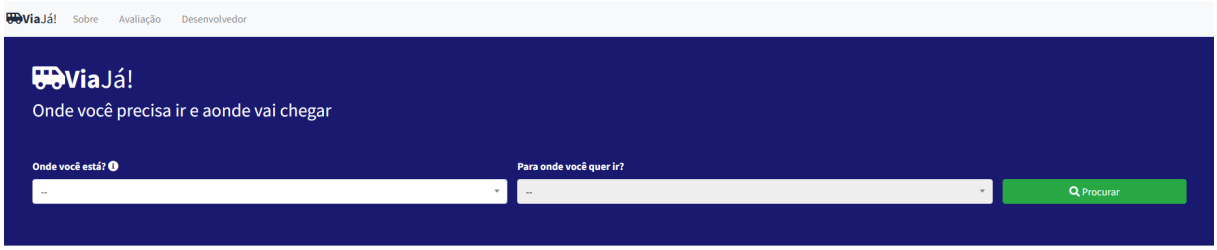
Fonte: Elaborada pelo autor

Exemplificando o comportamento de edição de distância e/ou valor de um trajeto, temos a Figura 11. O modal representado pode ser aberto ao clicar em ambas as células que representam um dado sobre o trajeto, seja a distância ou o valor.

3.3.3.2 Usuário

Quanto ao usuário final, é disponibilizado o acesso apenas a uma tela, a tela responsável pela consulta de informações de rotas de acordo com uma cidade de origem e uma cidade de destino. A tela de consulta, para a maioria dos usuários, por escolha, será a identidade do sistema.

Figura 12: Tela inicial do Viajá.

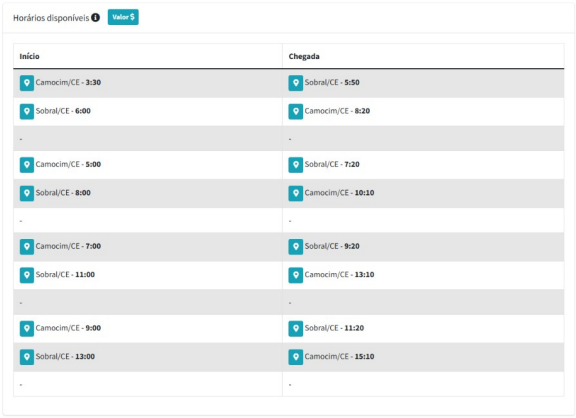


Fonte: Elaborada pelo autor.

Ao acessar a plataforma, o usuário se depara com dois campos esperando seleção e um botão para executar a consulta, como demonstrado na Figura 12. Quando o usuário seleciona as duas cidades e confirma apertando no botão, uma requisição à API é feita para que o banco de dados seja consultado retornando todos os horários de paradas que passem por aquelas cidades, agrupando-os por rota e organizando-os segundo seus horários.

Ao executar a consulta, caso existam rotas disponíveis entre as cidades escolhidas, um novo *card* será exibido na tela. Ele exibirá informações de horários e locais de embarque das topiques na cidade de origem, assim como o local e horário de desembarque na cidade de chegada.

Figura 13: Exemplo de resultado de consulta, card de horários disponíveis.



Fonte: Elaborada pelo autor.

A Figura 13 mostra um exemplo de consulta. Um usuário consultou rotas que saem da cidade de Sobral e chegam ou passam pela cidade de Camocim.

Ao clicar no botão de valor, uma pequena janela abre para informar o valor da passagem para o trajeto entre as cidades.

Figura 14: Exemplo de exibição de valor da passagem.

Fonte: Elaborada pelo autor.

Na Figura 14, um modal exibe o valor da passagem e a distância entre as cidades selecionadas.

3.3.4 Implantação

O processo de implantação descreve o processo de disponibilização da aplicação ao usuário final. Em aplicações *standalone*, seria representado pela instalação do software no computador do usuário. No caso de uma aplicação Web, ele descreve o processo de preparação dos servidores e hospedagem da aplicação.

Para a implantação do 'ViaJá!', nome dado a plataforma desenvolvida, foram utilizados os serviços da *Amazon Web Services* (AWS). A AWS dispõe de diversas soluções de computação em nuvem, das quais três foram utilizadas no projeto.

O banco de dados foi hospedado em uma instância do *Relational Database Service* (RDS) configurada para o postgreSQL com um limite de 30GBs de armazenamento. O uso de um Sistema Gerenciador de Banco de Dados (SGBD) *open source* é uma vantagem aqui, uma vez que não foi necessário comprar uma licença para configurar a instância do RDS.

A API e a aplicação foram hospedadas separadamente, em duas instâncias de máquinas virtuais do *Elastic Compute Cloud* (EC2), ambas usando o sistema operacional CentOS (distribuição do Linux exclusiva da Amazon), com 1GB de memória RAM e apenas 8GB de armazenamento cada. As duas máquinas foram configuradas com servidores Apache para funcionarem como servidores que podem receber requisições de fora da rede privada e IPs fixos através do sistema de endereços de IP elásticos da AWS.

Quanto ao endereço da aplicação, foi escolhido o endereço 'viajah.link'. O domínio foi comprado por um ano e custou 5 dólares, convertidos para cerca de 25 reais na cotação

atual. O sufixo “link” foi escolhido devido ao seu custo reduzido. Seu redirecionamento foi configurado através do *Amazon Route 53*, serviço de DNS da AWS.

3.4 Avaliação de usabilidade

Para coletar as respostas das avaliações da usabilidade, um formulário eletrônico foi aplicado através da ferramenta *Google Forms* para um público composto majoritariamente de estudantes universitários dos campi da UFC e de outras universidades da região norte do Ceará. A outra parte do público que respondeu à pesquisa foi composta pela rede de contatos próximos do autor do trabalho, em sua maioria residentes de Sobral, no Ceará. No total, foram coletadas respostas de 26 usuários.

O sistema utilizado para mensurar a usabilidade da solução foi o SUS. Composto por 10 afirmações, com respostas dispostas em uma escala Likert de 1 a 5, representando inclinações em diferentes níveis de intensidade quanto à conformidade com a afirmativa, sendo 1 o nível de “discordo fortemente” e 5 o nível de “concordo fortemente”. As afirmativas dispostas no formulário foram:

1. Eu acho que gostaria de usar esse sistema com frequência
2. Eu acho o sistema desnecessariamente complexo.
3. Eu achei o sistema fácil de usar.
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
5. Eu acho que as várias funções do sistema estão muito bem integradas.
6. Eu acho que o sistema apresenta muita inconsistência.
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
8. Eu achei o sistema atrapalhado de usar.
9. Eu me senti confiante ao usar o sistema.
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Além das afirmativas padrão do sistema SUS, foi adicionado um campo de texto aberto à sugestões de melhoria do sistema assim como relatório de erros encontrados durante os testes.

Para o cálculo da pontuação SUS de uma aplicação, a pontuação das respostas foi tratada da seguinte maneira: para as afirmativas 1,3,5,7 e 9, subtraímos 1 do número correspondente a resposta. Para as afirmativas 2,4,6,8 e 10, subtraímos o número da resposta

de 5. Após a soma de todas as pontuações, foi obtido uma soma que pode variar entre 0 e 40. Depois de calcular a pontuação da resposta de todos os participantes, foi tirada a média dos resultados e multiplicamos pela constante 2,5.

4 RESULTADOS E DISCUSSÃO

A seção de resultados servirá para demostrar o resultado da aplicação de um formulário eletrônico que implementa o sistema SUS de avaliação de usabilidade e uma breve discussão sobre a opinião e sugestões de melhorias vindas de usuários e empresários para os quais o projeto foi apresentado.

4.1 A pontuação SUS

Para o cálculo da pontuação SUS, os dados das respostas coletadas foram normalizados e então tratados como Brooke (1995) recomenda. Logo após, foi calculada a média de cada resposta pelos usuários para que depois fossem somadas e multiplicadas pela constante 2,5. O resultado desse cálculo foi uma pontuação de 91,5, muito acima da média apontada por Brooke (1995) em sua escala de 0 a 100. Para exemplificar, uma aplicação média obteria uma pontuação de 68.

4.2 Discussão sobre respostas do formulário

Mesmo que a métrica principal utilizada para avaliar a aplicação leve em consideração a média de todas as respostas do formulário, existem perguntas nas quais houve uma distribuição diferente de respostas. Essa diferença pôde indicar falhas de desenho da aplicação que não foram verbalizadas como sugestões mas que também merecem atenção.

Na afirmação 1 (Eu acho que gostaria de usar esse sistema com frequência.), foi encontrado um percentual de 12% de respostas que foram neutras ou negativas. Isso indica uma possível tendência de abandono de usuários no futuro que merece atenção.

Já nas afirmações 2 (Eu acho o sistema desnecessariamente complexo.) e 5 (Eu acho que as várias funções do sistema estão muito bem integradas), 7,6% das respostas apresentaram resultados neutros ou negativos, podendo indicar também um fluxo confuso para o usuário. Melhorar a legibilidade da interface de forma a guiar o usuário na utilização das funcionalidades poderia reduzir o percentual de respostas com essa opinião.

4.3 Pontos de melhoria

Diferente de um formulário SUS normal, foi incorporada uma nova pergunta. A última pergunta era um campo aberto a sugestões de melhoria e relatório de erros encontrados pelos usuários. A presença de um espaço aberto para sugestões se provou útil para definir futuros passos para o desenvolvimento de versões aprimoradas do projeto.

Alguns usuários apontaram dificuldade de leitura das informações das rotas, o que mostra uma possível falha na experiência do usuário, causada pela poluição visual. Uma forma de sanar futuramente esse problema seria uma reimaginação da disposição das informações, aumentando a legibilidade das informações.

Houve sugestões também da implementação do projeto na forma de um aplicativo *mobile*, justificadas por uma possível melhora na acessibilidade. O desenvolvimento de um aplicativo *mobile* poderia também abrir portas para a coleta dos dados de uso dos usuários e a construção de uma experiência personalizada para o público-alvo.

Um ponto levantado na perspectiva de negócio da aplicação foi seu estado de aplicação com alto índice de abandono, visto que um usuário médio só precisaria consultar uma rota uma única vez e depois não tem motivos para voltar a utilizar a plataforma. Possíveis soluções para esse problema seriam a integração com motoristas e cooperativas para oferecimento de promoções para usuários da plataforma ou a possibilidade de compra de passagens diretamente pela plataforma.

5 CONCLUSÃO

Através da pesquisa de mercado e contato com o possível público alvo, foi possível fazer o levantamento dos requisitos essenciais da ferramenta e validar a necessidade de resolução do problema percebido. Todos os usuários que responderam aos questionários preliminares explicitaram o desejo por uma plataforma que agregasse essas informações.

Graças a entrevistas aos motoristas locais, foi possível montar uma base de dados sólida para os testes da aplicação. Entretanto, uma futura colaboração direta com as cooperativas poderia render uma base de dados maior e mais detalhada, uma vez que entrevistar cada um dos motoristas leva tempo e devido ao caráter informal do projeto, não promove colaboração total no projeto.

Os requisitos funcionais foram levantados após a pesquisa de outras soluções que atuavam em setores semelhantes com funcionalidades úteis ao sistema, enquanto os requisitos não funcionais foram obtidos com sucesso através da comunicação com o potencial público-alvo.

Sobre o desenvolvimento da aplicação, conclui-se que foi bem-sucedido. Nos testes iniciais não foram encontrados erros críticos ou falhas de desenho que impedissem a utilização. Durante a avaliação dos usuários, foram apontadas algumas funcionalidades que fariam bom uso de refatoração, enquanto também foi pontuada a falta de legibilidade da informação na tabela principal com informação de horários.

Por final, a avaliação de usabilidade da plataforma ocorreu como desejado. Com um número satisfatório de respostas, foi possível mensurar de maneira confiável a qualidade da interface e funcionalidades da solução, assim como também foi possível listar pontos de melhoria nos quais deve ser colocado o foco no desenvolvimento futuro da aplicação.

REFERENCIAS

DANIEL, M.; MUNOZ, N. **Influência da presença digital das marcas no comércio eletrônico.** SP, Brasi: Business School São Paulo, 2016.

ARCE. **Relatório anual 2019**. Fortaleza 2019. Disponível em: https://www.arce.ce.gov.br/download/transportes/. Acesso em: 18 jun 2023.

ARCE. **Transportes**. Disponível em: https://www.arce.ce.gov.br/coordenadorias/transportes/. Acesso em: 29 out. 2023.

CEARÁ. Secretaria de Infraestrutura. **Serviço de transporte complementar é formalizado.** 2010. Disponível em: https://www.seinfra.ce.gov.br/2010/12/08/servico-de-transportecomplementar-e-formalizado/. Acesso em: 18 jun. 2023.

MVC. In: **WIKIPÉDIA, a enciclopédia livre**. Flórida: Wikimedia Foundation, 2023. Disponível em: https://pt.wikipedia.org/w/index.php?title=MVC&oldid=65860004. Acesso em: 12 mai. 2023.

DEACON, J. **Model-View-Controller (MVC) Architecture**. JOHN DEACON Computer Systems Development, Consulting & Training, 2009.

GAMMA, E.; ADDISON-WESLEY PUBLISHING COMPANY. **Design patterns CD : elements of reusable object-oriented software.** Reading, Ma: Addison-Wesley, 1998.

MASSARI, J. **Padrão MVC | Arquitetura Model-View-Controller**. Disponível em: https://www.portalgsti.com.br/2017/08/padrao-mvc-arquitetura-model-view-controller.html. Acesso em: 20 jun. 2023.

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Pearson, 2007. Acesso em: 20 jun. 2023.

BILLWAGNER. **Um tour por C# – Visão geral**. Disponível em: https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/. Acesso em: 21 jun. 2023.

GEWARREN. **Coleta de lixo no .NET**. Disponível em: https://learn.microsoft.com/pt-br/dotnet/standard/garbage-collection/. Acesso em: 21 jun. 2023.

RAMEZ ELMASRI; NAVATHE, S. B. **Sistemas de banco de dados**. São Paulo (Sp):

Pearson Addison Wesley, 2005.

POSTGRESQL. **PostgreSQL: About**. Disponível em: https://www.postgresql.org/about/.Acesso em: 20 jun. 2023.

TEIXEIRA, F. **O que é o SUS (System Usability Scale) e como usá-lo em seu site**. Disponível em: https://brasil.uxdesign.cc/o-que-%C3%A9-o-sus-system-usability-scale-e-como-us%C3%A1-lo-em-seu-site-6d63224481c8. Acesso em: 21 jun. 2023.

BARBOSA DE ALBUQUERQUE, D. **COMPORTAMENTO DO CONSUMIDOR EM RELAÇÃO AO TRANSPORTE COLETIVO URBANO POR ÔNIBUS E TOPICS EM FORTALEZA-CE**. TCC—UNIVERSIDADE FEDERAL DO CEARÁ: CURSO DE ADMINISTRAÇÃO, 2013 .

BROOKE, J. SUS: **A quick and dirty usability scale Fault diagnosis training View project System Usability Scale View project**. 1995.

FREITAS, André Luís Policani; REIS FILHO, Carlos Augusto Couto; RODRIGUES, Fernanda Ramos. **Avaliação da qualidade do transporte rodoviário intermunicipal e interestadual de passageiros: uma abordagem exploratória**. Revista Transportes, v. 19, n. 3, p. 49-61, 2011.

SEPLAG-CE. **Perfil da Macrorregião Sobral/Ibiapaba PERFIL DA MACRORREGIÃO SOBRAL/IBIAPABA Caracterização**. Disponível em: https://www.seplag.ce.gov.br/wp-content/uploads/sites/14/2011/05/Perfil-Regional-Sobral-Ibiapaba.pdf. Acesso em: 9 out. 2023.