UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CÂMPUS CORNÉLIO PROCÓPIO

DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

LEONARDO TEIXEIRA VIRGILIO

**SISTEMA DE ROTAS DE ÔNIBUS UTILIZANDO O ALGORITMO DE MELHOR CAMINHO**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2014

LEONARDO TEIXEIRA VIRGILIO

**SISTEMA DE ROTAS DE ÔNIBUS UTILIZANDO O ALGORITMO DE MELHOR CAMINHO**

Trabalho de conclusão de curso de graduação, apresentado à disciplina Trabalho de Diplomação, do curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas da Coordenação de Informática – COINF – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Danilo Sipoli Sanches

CORNÉLIO PROCÓPIO

2014

Dedico este trabalho primeiramente à Deus, às pessoas importantes que não estão mais presentes, aos meus familiares Elaine (mãe), Cil (pai) e Bianca (irmã), que sempre me apoiaram nesta trajetória, também gostaria de dedicar aos meus melhores amigos, dos quais sabem eles quem são, por terem me aguentado e me ajudado todo este tempo, juntamente para as pessoas demasiadamente importante para mim e que me acompanharam neste trajeto de enriquecimento profissional e sobre tudo pessoal, pessoas estas que teria o prazer de estar por perto sempre. E claro, não poderia esquecer o Seu Virço. Obrigado.

“Procure ser um homem de valor, em vez de ser um homem de sucesso.”

Albert Einstein.

**RESUMO**

VIRGILIO, Leonardo T.. **Sistema para rotas ônibus utilizando o algoritmo de Dijkstra (Melhor Caminho)**. 2014. 63 f. Tema de Trabalho de Conclusão de Curso (Graduação) - Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

Este trabalho propõe o desenvolvimento de um sistema que seja capaz de reconhecer e mapear o caminho com menor custo, bem como inserir, alterar e excluir os caminhos armazenados, além do desenvolvimento de um aplicativo para a plataforma Android que irá apresentar os dados do caminho com menor custo. Neste trabalho será utilizada a linguagem de programação Java, PHP com IDE Netbeans, Eclipse, banco de dados MySQL e como metodologia para o desenvolvimento será utilizado o Algoritmo de Dijkstra.

**Palavras-chave**: Algoritmo de Dijkstra, Android, Algoritmo Melhor Caminho, UTFPR.

.

**ABSTRACT**

VIRGILIO, Leonardo T.. **System of bus routes using Dijkstra's algorithm (the Best Way).** 2014. 63 p. Tema de Trabalho de Conclusão de Curso (Graduação) - Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014.

This work proposes the development of a system that is able to recognize and map the path of least cost as well as insert, update and delete the stored paths, and the development of an application for the Android platform that will display the path data lower cost. In this work will be used Java programming language, PHP IDE Netbeans, Eclipse, MySQL database and as a methodology for the development will be used Dijkstra's algorithm.

**Keywords**: Dijkstra's algorithm, Android, Algorithm the Best Way, UTFPR.

**LISTA DE FIGURAS**

[Figura 1 – Ilustração Cornélio Procópio - Belo Horizonte 16](#_Toc406169845)

[Figura 2 – Ilustração do grafo do Algoritmo de Dijkstra 17](#_Toc406169846)

[Figura 3 – Primeiro passo, execução do Algoritmo de Dijkstra 18](#_Toc406169847)

[Figura 4 – Segundo passo, execução do Algoritmo de Dijkstra 19](#_Toc406169848)

[Figura 5 - Terceiro passo, execução do Algoritmo de Dijkstra 19](#_Toc406169849)

[Figura 6 – Quarto passo, execução do Algoritmo de Dijkstra 20](#_Toc406169850)

[Figura 7 - Quinto passo, execução do Algoritmo de Dijkstra 20](#_Toc406169851)

[Figura 8 - Ilustração do Ciclo de Vida do SCRUM 24](#_Toc406169852)

[Figura 9 - Arquitetura do Sistema 28](#_Toc406169853)

[Figura 10 - Diagrama de Caso de Uso 30](#_Toc406169854)

[Figura 11 - Diagrama de Classe 33](#_Toc406169855)

[Figura 12 - Diagrama de Entidade e Relacionamento 34](#_Toc406169856)

[Figura 13 - Tela de busca da aplicação (imagem do emulador) 36](#_Toc406169857)

[Figura 14 - Execução da procedure dijresolve\_money 37](#_Toc406169858)

[Figura 15 - Tela de Processamento do Algoritmo de Melhor Caminho 37](#_Toc406169859)

[Figura 16 - Nós vizinhos com Cornélio Procópio (1º passo) 39](#_Toc406169860)

[Figura 17 - Nós vizinhos com Assis (2º passo) 40](#_Toc406169861)

[Figura 18 - Nós vizinhos com Belo Horizonte (3º passo) 41](#_Toc406169862)

[Figura 19 - Rota com menor custo 42](#_Toc406169863)

[Figura 20 - Tela de resultado do caminho com menor custo 43](#_Toc406169864)

**LISTA DE TABELAS**

[Tabela 1 - Algoritmo de Dijkstra 17](#_Toc406169879)

[Tabela 2 – Dados a partir do grafo do Algoritmo de Dijkstra, primeiro passo 18](#_Toc406169880)

[Tabela 3 – Dados a partir do grafo do Algoritmo de Dijkstra, segundo passo 19](#_Toc406169881)

[Tabela 4 – Dados a partir do grafo do Algoritmo de Dijkstra, terceiro passo 19](#_Toc406169882)

[Tabela 6 – Dados a partir do grafo do Algoritmo de Dijkstra, quarto passo 20](#_Toc406169883)

[Tabela 7 – Dados a partir do grafo do Algoritmo de Dijkstra, quinto passo 20](#_Toc406169884)

[Tabela 8- Funcionalidades do Sistema 31](#_Toc406169885)

[Tabela 9 - Rota alternativa CP/BH por Londrina/Bauru 44](#_Toc406169886)

[Tabela 10 - Rota alternativa CP/BH por Londrina/Araçatuba 44](#_Toc406169887)

[Tabela 11 - Rota alternativa CP/BH por Londrina/São José do Rio Preto 44](#_Toc406169888)

[Tabela 12 - Resultado da análise do questionário de avaliação 48](#_Toc406169889)

[Tabela 13 - Cronograma 49](#_Toc406169890)

**LISTA DE ABREVIATURAS**

|  |  |
| --- | --- |
| SRO | Sistema de Rotas de Ônibus |
| UTFPR | Universidade Tecnológica Federal do Paraná |
| UFSC  SISU | Universidade Federal de Santa Catarina  Sistema de Seleção Unificada |
| ENEM | Exame Nacional de Ensino Médio |
| CP | Cornélio Procópio |
| UML  JSP | Unified Modeling Language  JavaServer Pages |
| PR | Paraná |
| JVM  PHP  ADT | Java Virtual Machine  Personal Home Page  Android Developers Tools |
| IDE | Integrated Development Environment |
| XML | Extensible Markup Language |
| SGBD | Sistema de Gerenciamento de Banco de Dados |

SUMÁRIO

[1. INTRODUÇÃO 12](#_Toc406169922)

[2. JUSTIFICATIVA 14](#_Toc406169923)

[3. OBJETIVO GERAL 15](#_Toc406169924)

[4. METODOLOGIA DE DESENVOLVIMENTO 17](#_Toc406169925)

[4.1. Algoritmo de Dijkstra 17](#_Toc406169926)

[4.2. SCRUM 21](#_Toc406169927)

[5. DETALHES DA APLICAÇÃO 25](#_Toc406169928)

[5.1 Linguagens 25](#_Toc406169929)

[5.1.1 Java 25](#_Toc406169930)

[5.1.2 PHP 25](#_Toc406169931)

[5.2 Ferramentas 26](#_Toc406169932)

[5.2.1 Eclipse + Android Developer Tools 26](#_Toc406169933)

[5.2.2 Sublime Text 26](#_Toc406169934)

[5.2.3 MySQL 27](#_Toc406169935)

[5.2.4 NetBeans 27](#_Toc406169936)

[5.3 Arquitetura do Sistema 28](#_Toc406169937)

[5.4 Modelagem 28](#_Toc406169938)

[5.4.1 Diagrama de Caso de Uso 30](#_Toc406169939)

[5.4.2 Diagrama de Classe 33](#_Toc406169940)

[5.4.3 Diagrama de Entidade e Relacionamento 34](#_Toc406169941)

[6. VISÃO GERAL DA APLICAÇÃO 35](#_Toc406169942)

[7. CONCLUSÕES FINAIS 46](#_Toc406169943)

[7.2 Dificuldades Encontradas 46](#_Toc406169944)

[7.3 Trabalhos Futuros 46](#_Toc406169945)

[7.4 Validação do aplicativo 47](#_Toc406169946)

[8 CRONOGRAMA 49](#_Toc406169947)

[9 REFERÊNCIAS 50](#_Toc406169948)

[APÊNDICE A - Implementação das Procedures 52](#_Toc406169949)

[APÊNDICE B - Tabela De Dados Coletados 56](#_Toc406169950)

[APÊNDICE C - Questionário De Avaliação Da Aplicação 61](#_Toc406169951)

# 1. INTRODUÇÃO

Os transportes coletivos no Brasil têm grande importância no cotidiano da população, já que representam, em muitos casos, o único meio de deslocamento entre residência e trabalho ou escola. Dentre os transportes coletivos, o ônibus é o meio mais utilizado, pela sua maior acessibilidade e pelo atendimento amplo aos desejos de destino da população. Porém a demanda pelo sistema de ônibus no Brasil vem sofrendo um declínio nos últimos anos. Vários fatores podem ser listados para justificar este fato, dentre eles, a popularização do transporte aéreo e o aumento viagens por automóvel, etc. Com o aumento do número de automóveis particulares, os congestionamentos ocorrem com maior frequência. De acordo com (Zhang, 2011) para melhorar o trânsito, as empresas do transporte coletivo e o governo público necessitam aprimorar os serviços prestados. Os principais obstáculos que os usuários encontram quando optam pelo transporte coletivo são a ausência de informações relativas e/ou a baixa qualidade do serviço (Cutolo, 2003).

Neste cenário, que se caracteriza pela perda de espaço no mercado, torna-se fundamental a busca de referenciais competitivos. E, no processo de busca pela melhoria, já se pode contar com auxílio de tecnologias avançadas específicas para o uso no transporte por ônibus. Os desenvolvimentos de novas tecnologias de informação têm proporcionado grandes avanços no gerenciamento dos sistemas de transporte. Há várias tecnologias testadas e em operação que ajudam no papel de controle da busca pelo transporte coletivo rodoviário. Com o aumento e popularidade dos dispositivos móveis houve um crescente surgimento de novas aplicações que possibilitam atender às diversas necessidades dos usuários.

Inspirado nas novas tecnologias o presente trabalho refere-se ao desenvolvimento de um sistema para a plataforma Android. O SRO (Sistema para Rotas de Ônibus), uma aplicação para smartphones que utilizam o sistema operacional Android que fornecerá aos seus usuários a possibilidade de consultarem rotas de ônibus.

O propósito principal do SRO é que o usuário consiga realizar buscas por caminhos de ônibus por meio de filtro baseado no algoritmo de (E. W. Dijkstra, 1959). Diferente das consultas disponíveis nos sites das grandes empresas de transportes, em que somente são mostrados resultados da busca quando a cidade de origem e destino possuem uma rota direta, ou seja, quando o usuário do transporte coletivo não necessita de realizar conexões para chegar ao seu destino. O SRO realizará a consulta de caminhos (rotas de ônibus) mesmo havendo a possibilidade do usuário realizar conexões para chegar ao seu destino, por meio do algoritmo de Melhor Caminho do E. W. Dijkstra.

De acordo com (A. C. Mariani, 2001) professor do Departamento de Informática e Estatística da UFSC (Universidade Federal de Santa Catarina):

O Algoritmo de Dijkstra (E.W. Dijkstra, 1959) é um dos algoritmos que calcula o caminho de custo mínimo entre vértices de um grafo. Escolhido um vértice como raiz da busca, este algoritmo calcula o custo mínimo deste vértice para todos os demais vértices do grafo.

As próximas seções apresentam o objetivo geral, os limites e restrições da solução, as tecnologias a serem utilizadas e o cronograma que será obedecido.

# 2. JUSTIFICATIVA

No ano de 2009 o Ministério de Educação brasileiro desenvolveu uma plataforma para que os estudantes que realizaram o Exame Nacional do Ensino Médio (ENEM) possam se inscreverem em instituições de ensino superior que aderirem totalmente ou parcial, à nota do Enem como forma de ingresso, em substituição ao vestibular. Com o Sistema de Seleção Unificada (SISU), o candidato pode escolher até duas opções de curso, sendo permitidas alterações durante o período de inscrição, desta maneira possibilitou que o mesmo não necessite realizar vestibular na cidade em que a instituição de ensino superior pertence. Por sua vez está mais comum a migração de estudantes para diversos lugares do Brasil. Muitos alunos estudam em universidades que ficam localizadas distantes sua residência. Essa distância, de certa forma é um grande problema que o estudante enfrenta ao sair de casa, pois na maioria dos casos eles dependem do transporte coletivo para deslocar-se de sua cidade natal para a cidade em que irá estudar.

A partir deste problema, foi realizado entrevistas informais com alunos da Universidade Tecnológica Federal do Paraná do Campus Cornélio Procópio (UTFPR-CP) e muitos deles citaram a dificuldade em encontrar rotas de ônibus para chegar ou sair de Cornélio Procópio, pois há poucas empresas que oferecem esse serviço para destinos com grandes distâncias em Cornélio Procópio. Pensando nessa necessidade de que o estudante tem em planejar toda a logística de sua viagem foi que surgiu a ideia do aplicativo para a plataforma Android que propõem minimizar esse problema.

Um pequeno exemplo para demonstrar o problema:

Se um estudante precisar ir para Presidente Prudente – SP, cerca de duzentos quilômetros de distância, ele precisa de ir para Londrina – PR, Assis – SP ou Ourinhos – SP, a cidade de Cornélio Procópio não possui uma rota/linha de ônibus que realiza este caminho diretamente.

# 3. OBJETIVO GERAL

O objetivo geral do trabalho é desenvolver um sistema para estudantes ou passageiros em geral que dependem do transporte coletivo para se locomoverem entre uma cidade e outra sem precisar de realizar grandes buscas e por meio de filtro onde irá mostrar o caminho de menor custo.

O sistema funcionará da seguinte maneira:

Uma pessoa que precisa viajar de Cornélio Procópio – PR para a cidade de Belo Horizonte – MG a distância entre as duas cidades é de cerca 1034 quilômetros (Google Maps) podendo variar dependendo do caminho escolhido. Essa pessoa leva cerca de 22 horas de viajem e gasta em média de R$ 200,00 com passagens. Com o aplicativo, esse viajante poderá pesquisar caminhos com menor custo possibilitando uma melhor economia.

Esta situação é um problema comum que estudantes sofrem ao ir morar longe de casa e acontece frequentemente quando a cidade de origem e destino são cidades pequenas, pois há poucas rotas de ônibus e o tempo que se gasta é elevado para chegarem ao seu destino.

A figura abaixo (Figura 1) traz uma versão ilustrada das possibilidades de caminhos que um usuário de transporte coletivo poderá escolher para chegar em seu ponto de destino. Nas próximas seções há uma explicação com maior riqueza em detalhes.

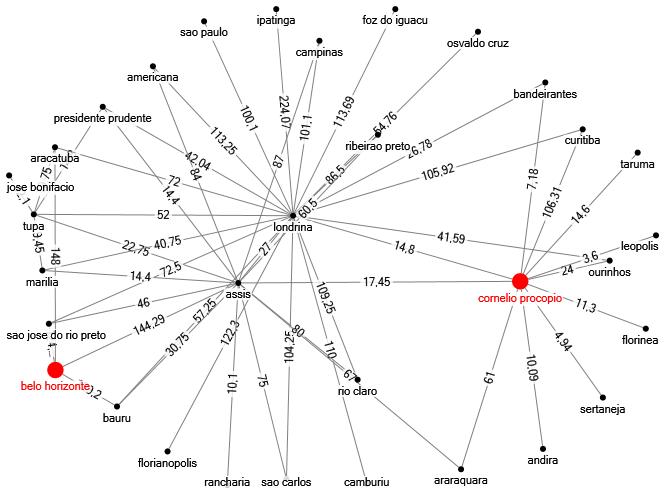


Figura 1 – Ilustração Cornélio Procópio - Belo Horizonte

# 4. METODOLOGIA DE DESENVOLVIMENTO

# 4.1. Algoritmo de Dijkstra

Das mais diversas utilizações dos grafos (genética, ciências sociais, robótica, etc) uma delas é a determinação de um caminho com custo mínimo. Assim, uma rede de estradas por exemplo que liga diversas cidades (nós) pode ser representada por meio de um grafo. Através da concepção de um grafo e do Algoritmo de Dijkstra há a possibilidade de calcular o caminho com menor custo para realizar um determinado percurso.

O Algoritmo de Dijkstra é um dos algoritmos que calcula o custo mínimo entre vértices de um grafo, através do nó raiz da busca este algoritmo calcula o custo mínimo deste nó para todos os demais nós.

Este algoritmo parte de uma estimativa inicial para o custo mínimo e vai sucessivamente ajustando esta estimativa. Ele considera que um nó estará fechado quando já tiver sido obtido um caminho de custo mínimo do nó tomado como raiz da busca até ele. Caso contrário ele dito estar aberto.

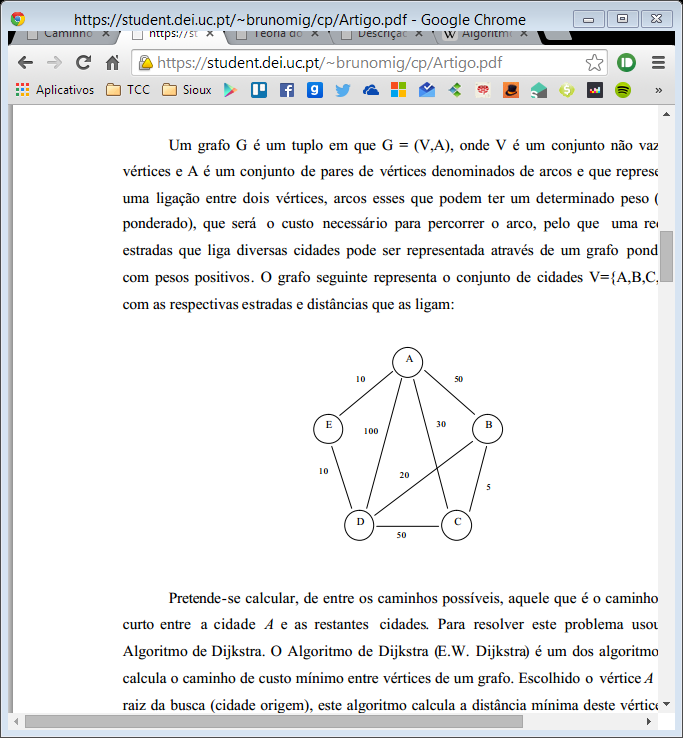


Figura 2 – Ilustração do grafo do Algoritmo de Dijkstra

Tabela 1 - Algoritmo de Dijkstra

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NÓS | **A** | **B** | **C** | **D** | **E** |
| CUSTO | 0 | ∞ | ∞ | ∞ | ∞ |
| PRECEDENTE | - | - | - | - | - |
| FECHADO | N | N | N | N | N |

Inicialmente todos os nós possuem um custo infinito, exceto o vértice raiz, que logicamente seu custo é zero. Após o nó que tem um custo menor em relação à todos os outros nós e encontra-se “aberto”. O nó selecionado é “fechado” e são recalculados de acordo com a raiz da busca o custo às arestas respectivas. Na medida que irá descobrindo novas rotas inferiores às que se encontram armazenadas nas tabelas, procede-se à substituição por novas rotas e também é alterado o nó precedente.

Segue abaixo o primeiro passo:

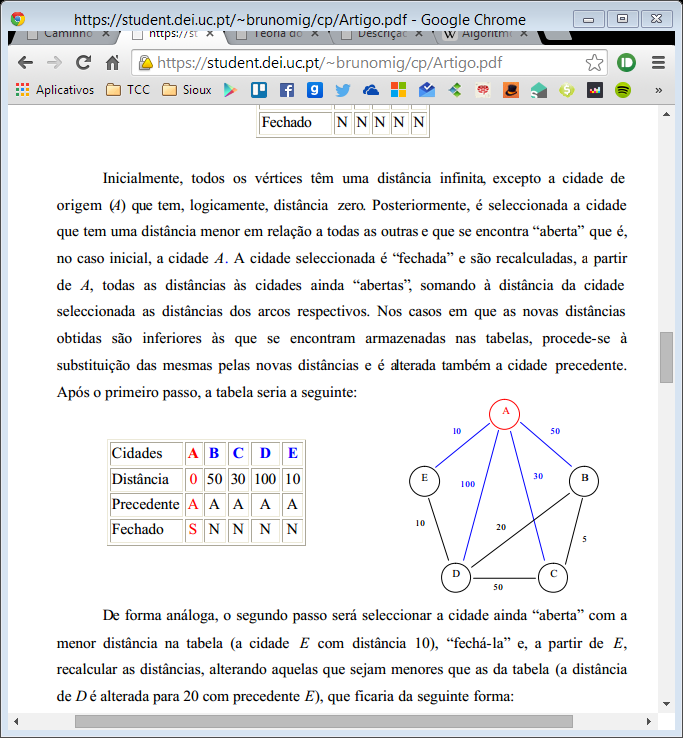


Figura 3 – Primeiro passo, execução do Algoritmo de Dijkstra

Tabela 2 – Dados a partir do grafo do Algoritmo de Dijkstra, primeiro passo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NÓS | **A** | **B** | **C** | **D** | **E** |
| CUSTO | 0 | 50 | 30 | 100 | 10 |
| PRECEDENTE | A | A | A | A | A |
| FECHADO | S | N | N | N | N |

O segundo passo é selecionar o nó que ainda está “aberto” com o menor custo (o vértice E com custo 10), a partir de *E*, o próximo passo é recalcular os custos, alterando aquelas que são menores que as da tabela (o custo de *D* é alterado para 20 com precedente o nó *E*).

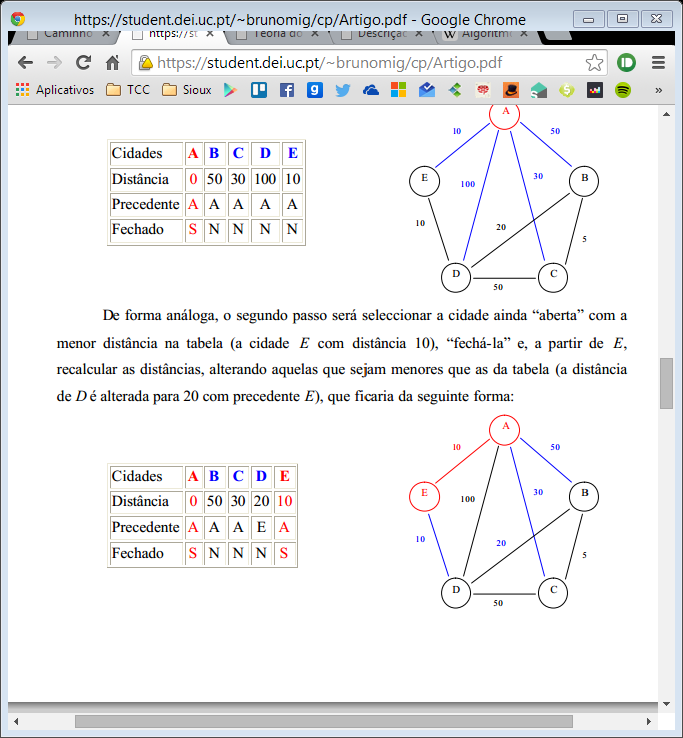


Figura 4 – Segundo passo, execução do Algoritmo de Dijkstra

Tabela 3 – Dados a partir do grafo do Algoritmo de Dijkstra, segundo passo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NÓS | **A** | **B** | **C** | **D** | **E** |
| CUSTO | 0 | 50 | 30 | 20 | 10 |
| PRECEDENTE | A | A | A | E | A |
| FECHADO | S | N | N | N | S |

Desde modo, o algoritmo repete-se até que todos os nós tenham sido “fechados”. Segue as seguintes tabelas:

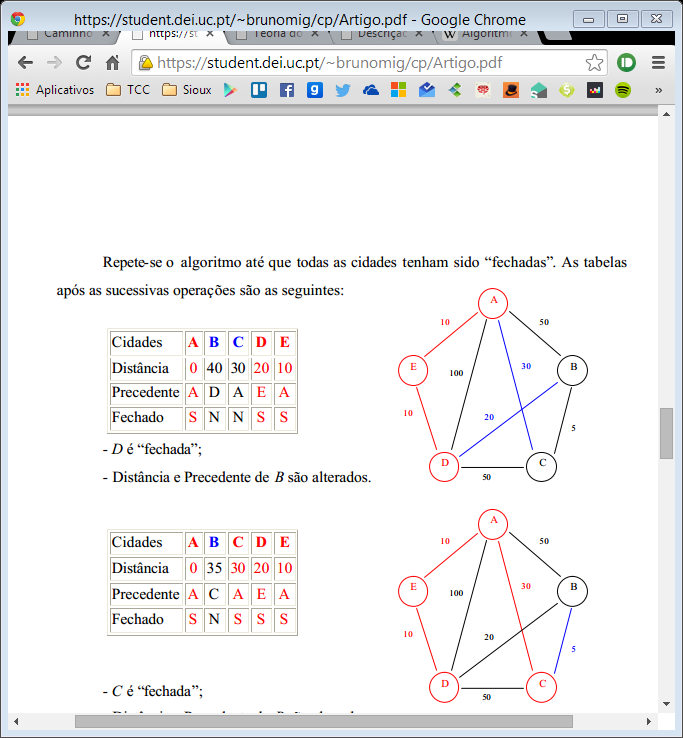


Figura 5 - Terceiro passo, execução do Algoritmo de Dijkstra

Tabela 4 – Dados a partir do grafo do Algoritmo de Dijkstra, terceiro passo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NÓS | **A** | **B** | **C** | **D** | **E** |
| CUSTO | 0 | 40 | 30 | 20 | 10 |
| PRECEDENTE | A | D | A | E | A |
| FECHADO | S | N | N | S | S |

O nó *D* é “fechado” e o custo e precedente do nó *B* são alterados.

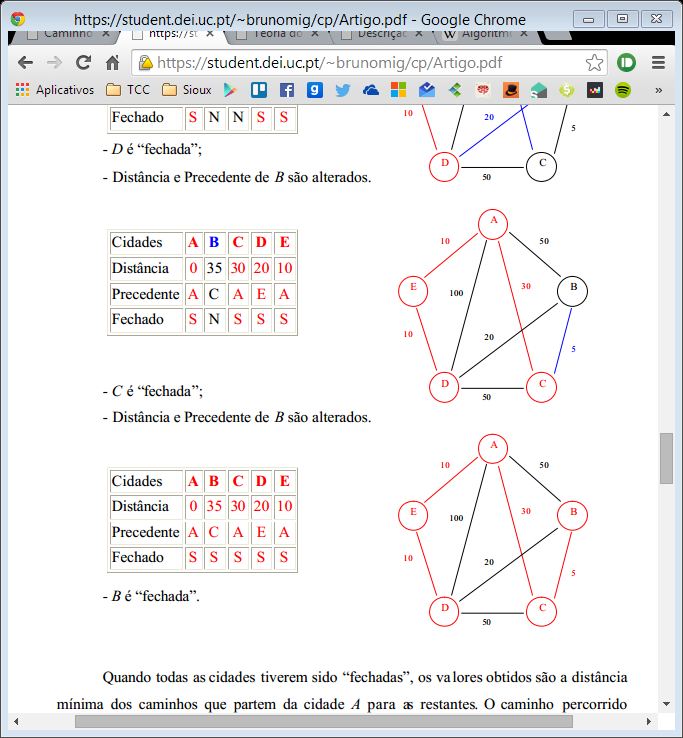


Figura 6 – Quarto passo, execução do Algoritmo de Dijkstra

Tabela 5 – Dados a partir do grafo do Algoritmo de Dijkstra, quarto passo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NÓS | **A** | **B** | **C** | **D** | **E** |
| CUSTO | 0 | 35 | 30 | 20 | 10 |
| PRECEDENTE | A | C | A | E | A |
| FECHADO | S | N | S | S | S |

O nó *C* é “fechado” e o custo e precedente do nó *B* são alterados.

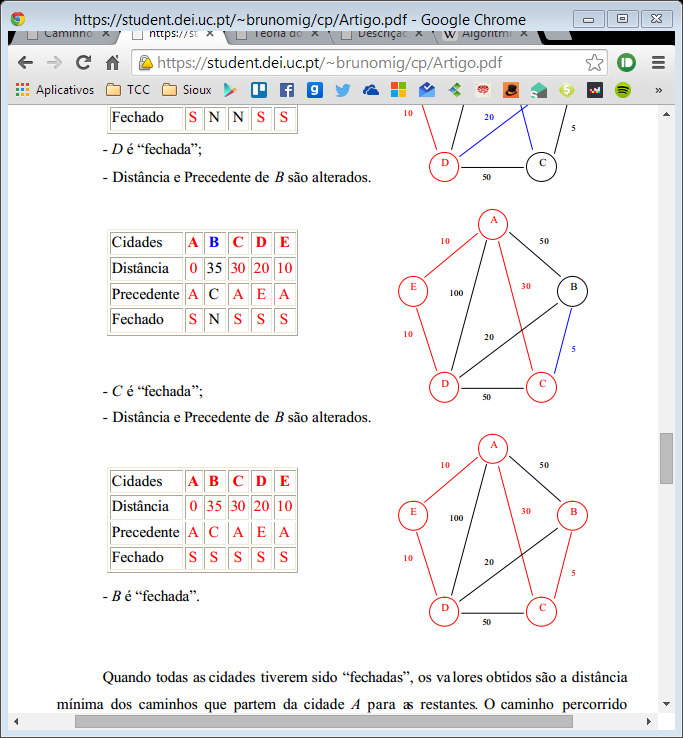


Figura 7 - Quinto passo, execução do Algoritmo de Dijkstra

Tabela 6 – Dados a partir do grafo do Algoritmo de Dijkstra, quinto passo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NÓS | **A** | **B** | **C** | **D** | **E** |
| CUSTO | 0 | 35 | 30 | 20 | 10 |
| PRECEDENTE | A | C | A | E | A |
| FECHADO | S | S | S | S | S |

O nó *B* é “fechado.

Quanto todos os nós tiverem sido “fechados”, os valores obtidos serão os custos mínimos das rotas que partem do nó tomando como raiz da busca até os demais nós do grafo. O caminho percorrido nesse trajeto a partir dos nós precedentes. Para exemplificar, o caminho que irá do nó *A* até o *B,* cujo o custo mínimo é de 35. O vértice precedente de *B* é o vértice *C.* Portando o caminho é *A →* C *→ B.* Conclui-se que o precedente de *C* é A. Portando a rota final é *A → C → B,* com o custo mínimo de 35.

Vale ressaltar que o Algoritmo de Dijkstra só pode ser utilizado com valores em grafos com valores unicamente positivos. Ele não garante, contudo, a exatidão da solução caso haja a presença de arestas com valores negativos.

# 4.2. SCRUM

A metodologia adotada para o desenvolvimento do sistema será o processo ágil Scrum.

O Scrum foi desenvolvido por (Ken Shwaber e Jeff Stherland, 2001) para ajudar com a necessidade de encontrar uma metodologia que abordasse o problema do desenvolvimento de software de forma não tradicional. Neste contexto, podemos definir o Scrum como um framework para auxiliar resoluções de problemas permitindo um melhor gerenciamento do projeto.

O processo ágil de gerência de projetos Scrum não resolve todos os problemas, mas os torna mais visíveis. Além disso utiliza boas práticas para motivar pessoas, satisfazer necessidades de um cliente real e ensina formas de aperfeiçoar o ambiente de trabalho, é baseado no empirismo, ou seja, de acordo com Jungle (2004, p. 5) o empirismo é um conhecimento baseado nas experiências adquiridas por processos de experimentação e testes.

Segundo (Soares, 2004) o Scrum pode ser utilizado por pequenas ou grandes equipes, porém nada impede de ser utilizado por uma pessoa só. Só que neste caso você deve cumprir todos os papeis do clico de vida (veja a Figura 8).

O Scrum possui a seguinte hierarquia:

* ProductOwner – É o dono do produto, e o responsável por maximizar o valor do produto e o esforço despendido da sua equipe, ele também é o único responsável por organizar o BacklogProduct.
* BacklogProduct – Lista de definições que compõe o software, ou seja, é uma lista de atividades que dará origem ao software completo, sendo eles seus requisites funcionas e não funcionais do software proposto.
* Planning Meeting – Plano para cada Sprint, geralmente ocorre antes do início de cada Sprint, o ScrumMaster mostra a sua equipe o que será necessário desenvolver até o final deste Sprint, essas funcionalidades expostas aos integrantes da equipe ficam definidas no BacklogProduct.
* Daily Scrum – Reuniões diárias da que a equipe se reúne e compartilha o que foi feito no dia anterior, basicamente são realizadas algumas questões, como: O que você fez no projeto desde a última Daily Scrum? O que você planeja fazer no projeto no período entre essa reunião e a próxima? O que está te impedindo de cumprir suas metas para esse projeto? Os objetivos destas reuniões são para sincronização das tarefas e identificar problemas dos quais a equipe então procurar soluções rápidas.
* Sprint Review – Reunião antes do Planning Meeting, a equipe mostra ao ScrumMaster o que fora desenvolvido referente ao Sprint referido, geralmente nestas reuniões são integrantes também os usuários do sistema, para que então eles homologuem o que fora feito pela equipe de desenvolvimento.



Figura 8 - Ilustração do Ciclo de Vida do SCRUM

# 5. DETALHES DA APLICAÇÃO

Descrição das ferramentas e tecnologias utilizadas para a construção da aplicação serão apresentadas abaixo:

# 5.1 Linguagens

Nesta seção serão apresentadas as linguagens de programação que foram utilizadas.

# 5.1.1 Java

É uma linguagem de programação e uma plataforma de computação desenvolvida pela Sun Microsystems. (Oracle, 2010). É uma linguagem orientada a objetos de alto desempenho, suportando multithreading, compilada para um bytecode que é interpretada por uma máquina virtual (JVM).

Suas principais características são a portabilidade que permite o Java ser executado em qualquer plataforma que possua o interpretador Java. É orientado a objetos baseada no modelo Simular. Há inúmeros recursos e rotinas que facilitam a cooperação de protocolos de redes de computadores. Possui sua sintaxe similar às linguagens de programação C/C++.

# 5.1.2 PHP

De acordo com a documentação oficial, "PHP: Hypertext Preprocessor", é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a Web e pode ser mesclada dentro do código HTML. É uma linguagem de script open source de uso geral, muito utilizada e especialmente guarnecida para o desenvolvimento de aplicações Web inserida dentro do HTML.

O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente rapidamente, mas você pode fazer muito mais do que isso com PHP.

5.2 Ferramentas

Nesta seção serão apresentadas as ferramentas que foram utilizadas para o desenvolvimento da aplicação.

5.2.1 Eclipse + Android Developer Tools

“O Eclipse é o ambiente de desenvolvimento preferido pelo Google, e há um plug-in chamado ADT (Android Development Tools) para facilitar o desenvolvimento, os testes e a compilação do projeto.” (LACHETA, 2010, 28).

Segundo (Android Development Tools, 2014) o ADT amplia os recursos da IDE Eclipse para que seja possível configurar rapidamente novos projetos Android. Possibilita de criar interface de usuário do aplicativo, adicionar pacotes com base na API Framework Android, depurar os aplicativos usando as ferramentas do SDK do Android até mesmo exportar arquivos .apk a fim de distribuir a aplicação.

5.2.2 Sublime Text

Um editor de texto e código fonte multiplataforma, escrito em linguagem C++, simples mas com grande potencial, inicialmente utilizado para realizar modificações nos arquivos .php, .css, .js e .html.

# 5.2.3 MySQL

Segundo [MySQL, 2014] o MySQL é um software de banco de dados mais popular de código aberto do mundo, com mais de 100 milhões de cópias de seu software baixado ou distribuído por toda a sua história. Com sua velocidade superior, confiabilidade e facilidade de uso, o MySQL se tornou a escolha preferida para Web, Web 2.0, SaaS, ISV.

MySQL é uma parte fundamental de LAMP (Linux, Apache, MySQL, PHP / Perl / Python), Mais e mais empresas estão usando LAMP como uma alternativa aos caros software proprietário por causa de seu baixo custo e à liberdade de plataforma. MySQL foi originalmente fundada e desenvolvida na Suécia por dois suecos e um finlandês: David Axmark, Allan Larsson e Michael "Monty" Widenius, que havia trabalhado juntos desde a década de 1980.

# 5.2.4 NetBeans

Segundo (NetBeans, 2014) O NetBeans IDE oferece suporte superior para desenvolvedores de C/C++ e PHP, oferecendo editores e ferramentas abrangentes para os frameworks e tecnologias relacionadas. Além disso, o IDE tem editores e ferramentas para XML, HTML, PHP, Groovy, Javadoc, JavaScript e JSP.

É possível instalar o NetBeans IDE em todos os sistemas operacionais que suportam Java, desde sistemas Windows, Linux a Mac OS. Escreva Uma Vez, Execute em Qualquer Lugar é uma frase tão aplicável ao NetBeans IDE quanto às suas próprias aplicações... porque o NetBeans IDE é escrito em Java também.

# 5.3 Arquitetura do Sistema

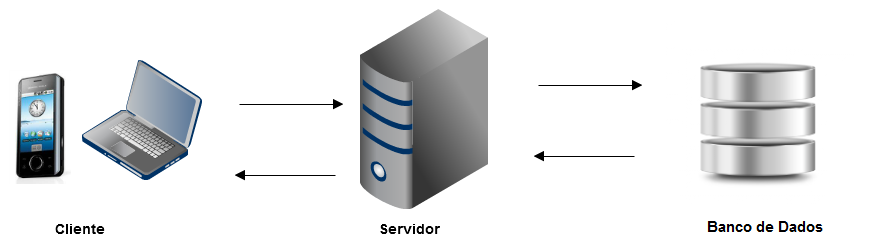


Figura 9 - Arquitetura do Sistema

No ano de 2000 surgiu o conceito de Web Service, que de acordo com Nicolai M. Josuttis, 2007, os Web Services referem-se a uma coleção de padrões que provem a capacidade de um sistema comunicação de se comunicar com outro sistema e podem ser definidos como uma tecnologia em sistemas distribuídos que tem o intuito de solucionar o problema da comunicação entre aplicações diferentes.

Por meio desta tecnologia é possível realizar interações com aplicações de forma padronizada, enviando e recebendo dados em um formato pré-estabelecido, mesmo que as aplicações sejam desenvolvidas em linguagens distintas.

5.4 Modelagem

Somente com o auxílio da modelagem do sistema é possível manipular a implementação do sistema de maneira eficaz. Há vários métodos de destaque como Booch de Grady, Booch, OOSE de Ivar Jacobson e OMT de James Rumbaugh. Houve a necessidade de criar-se uma nova linguagem unificada, intitulada de UML (Linguagem de Modelagem Unificada) que é uma tentativa de padronizar a modelagem orientada a objetos de forma que qualquer sistema possa ser modelado.

Nesta seção são apresentados o diagrama de classe (Figura 10), diagrama de casos de uso (Figura 11) e diagrama de entidade relacionamento (Figura 12).

5.4.1 Diagrama de Caso de Uso

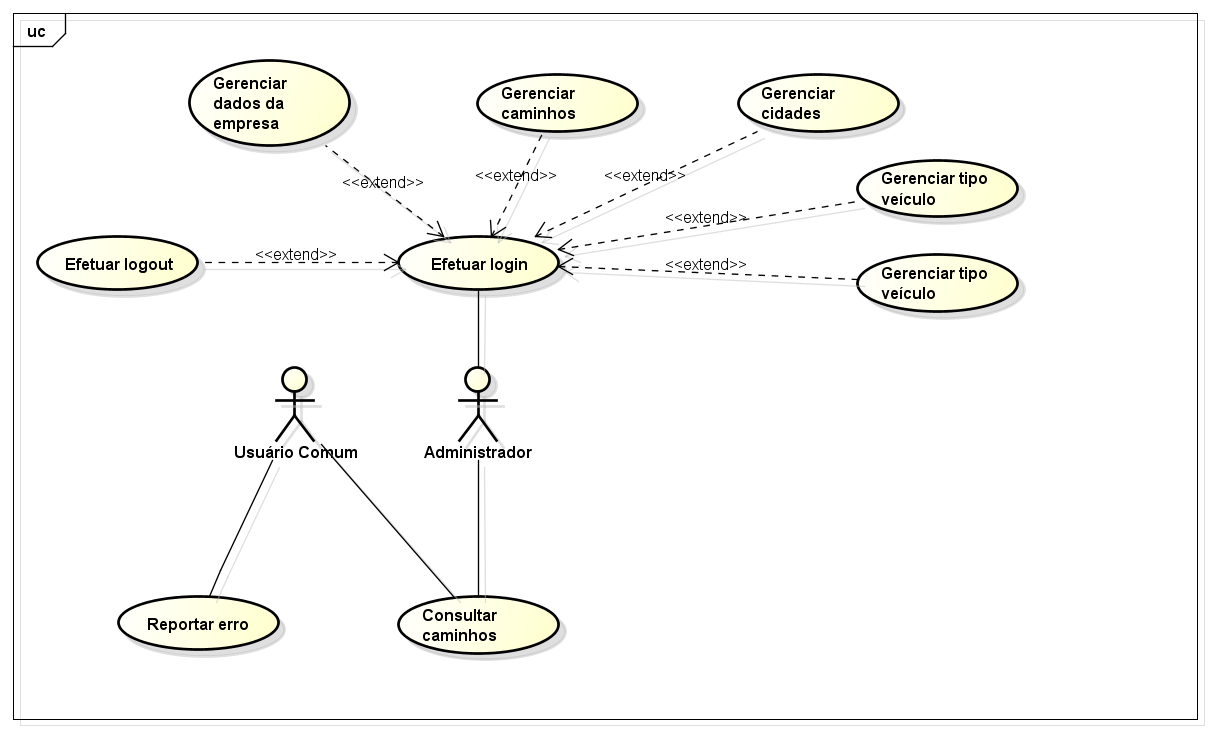


Figura 10 - Diagrama de Caso de Uso

Abaixo está descrito as funcionalidades do sistema a serem desenvolvidas e estão classificadas de acordo com as seguintes prioridades:

Essencial: Deverá ser desenvolvida de forma completa, inicial e prioritária;

Importante: Deverá ser desenvolvida de forma completa, porém dá prioridade anteriormente às funcionalidades de precedência essencial.

Desejável: Poderá ser desenvolvida após as de caso de prioridade essencial e importante, podendo em caso de atraso no cronograma, ser descartada.

Tabela 7- Funcionalidades do Sistema

|  |  |  |
| --- | --- | --- |
| IDENTIFICAÇÃO | FUNCIONALIDADE | PRIORIDADE |
| F1 | Consultar caminhos com menor custo | Essencial |
| F2 | Consultar caminhos com menor tempo | Desejável |
| F3 | Gerenciar caminhos | Essencial |
| F4 | Gerenciar dados das empresas | Essencial |
| F5 | Gerenciar veículos | Importante |
| F6 | Gerenciar cidades | Importante |
| F7 | Reportar erro | Desejável |

**F1** – O usuário que possuir o aplicativo Android poderá consultar os caminhos cadastrados no sistema filtrando por menor custo.

**F2** – O usuário que possuir o aplicativo Android poderá consultar os caminhos cadastrados no sistema filtrando por menor tempo.

**F3** – Os usuários que possuem acesso à área administrativa do sistema poderão inserir, alterar e excluir os caminhos armazenados no banco de dados através da interface administrava web.

**F4** – Os usuários que possuem acesso à área administrativa do sistema poderão inserir, alterar e excluir os dados das empresas armazenados no banco de dados através da interface administrava web.

**F5** – Os usuários que possuem acesso à área administrativa do sistema poderão inserir, alterar e excluir os veículos armazenados no banco de dados através da interface administrava web.

**F6** – Os usuários que possuem acesso à área administrativa do sistema poderão inserir, alterar e excluir as cidades armazenados no banco de dados através da interface administrava web.

**F7** – Os usuários que utilizarem o sistema, poderão reportar algum erro ou sugestão através de um canal de comunicação entre o usuário e o administrador do sistema.

5.4.2 Diagrama de Classe

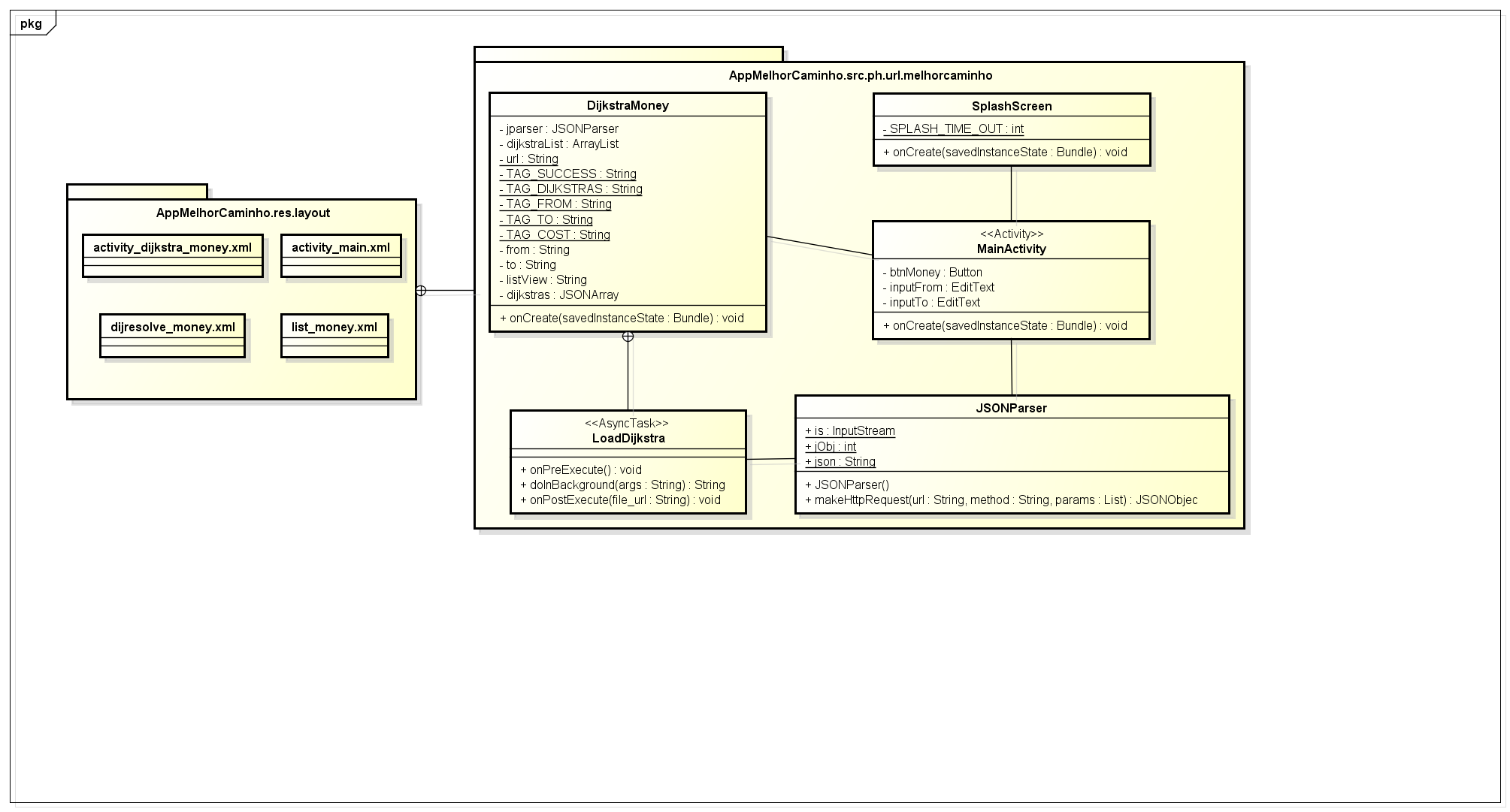


Figura 11 - Diagrama de Classe

5.4.3 Diagrama de Entidade e Relacionamento

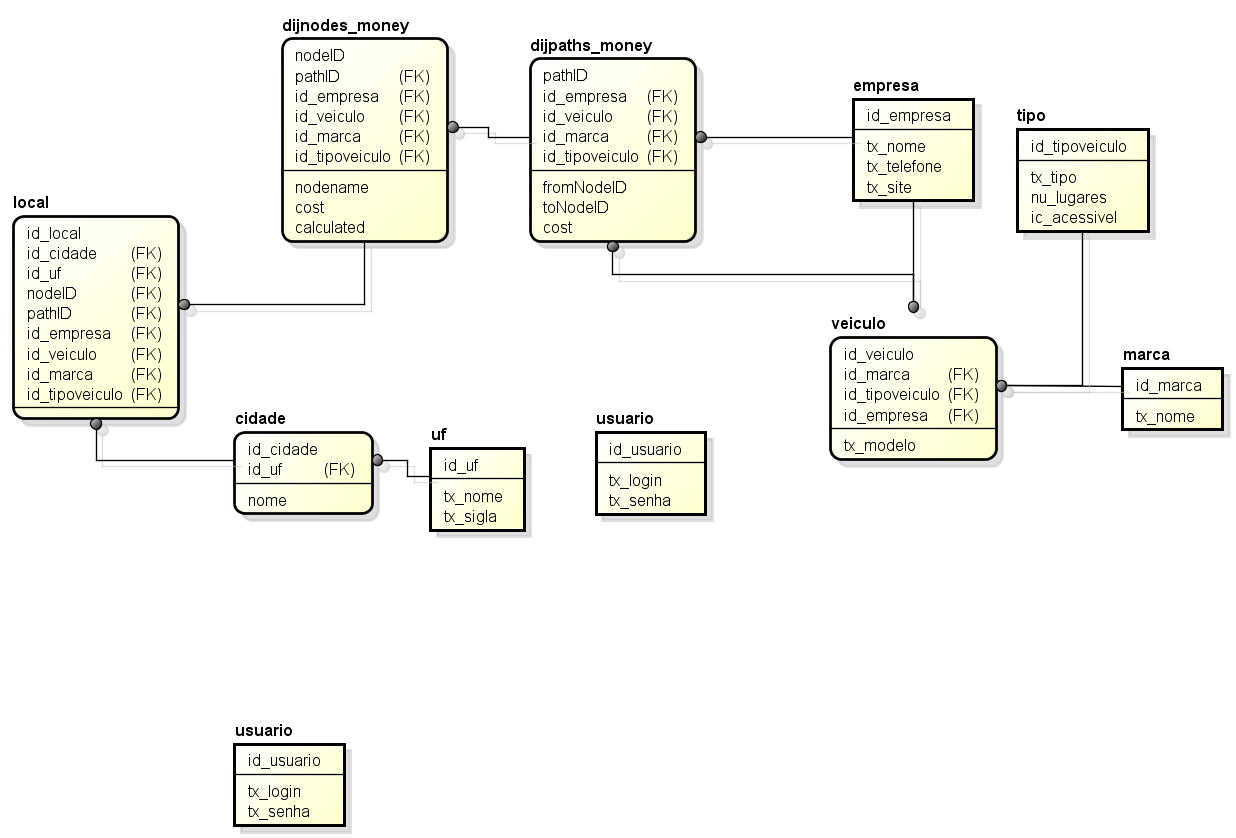


Figura 12 - Diagrama de Entidade e Relacionamento

# 6. VISÃO GERAL DA APLICAÇÃO

Com o objetivo de ilustrar o funcionamento da aplicação proposta neste trabalho, será apresentado uma simulação de uma viagem de ônibus partindo da cidade de Cornélio Procópio com destino a cidade de Belho Horizonte, Minas Gerais. Para tanto, foram coletados dados reais como custos, tipo de veículo e tempo de viagem de algumas rotas de determinadas empresas. As empresas escolhidas foram, Viação Garcia, Guerino Seiscentos e Gontijo.

Os dados coletados foram feitos de maneira manual, através do sistema de consultas de passagens que as empresas disponibilizam em seus websites. O ideal seria obter toda a base de dados com informações sobre as rotas/caminhos que aquela determinada empresa realiza. Porém como a maneira de se obter esses dados não foi a mais adequada, não foi possível ter conhecimento de todas as rotas/caminhos que as empresas disponibilizam em seus itinerantes, todos os dados obtidos das empresas de transportes podem ser visualizados no Apêndice B. Mais detalhes sobre as limitações do software serão explicados no tópico Detalhes e Limitações da Aplicação.

A partir dos dados coletados pode-se observar que há inúmeras possibilidades de chegar no destino final, Belo Horizonte, tendo como ponto de origem Cornélio Procópio conforme a Figura 1. É de importância salientar que cada ligação entre os nós (cidades) há um valor intitulado de “Custo”. O “Custo” são os preços das passagens entre um nó (cidade) e outro. Possuindo o conhecimento desta informação o Algoritmo de Dijkstra consegue mapear uma rota com menor custo entre os pontos de origem e destino.

Abaixo pode-se visualizar a tela principal da aplicação, a tela de buscas. Interface simples, intuitiva. A tela de buscas contém dois campos de texto intitulados, origem e destino, e apenas um botão com o nome pesquisar. Possibilitando desde modo a consulta do caminho de forma simples e direta.



Figura 13 - Tela de busca da aplicação (imagem do emulador)

Desta forma, a aplicação conecta com o servidor web com auxílio do framework JSON enviando os dados de origem e destino para a realização da consulta por meio do Algoritmo de Dijkstra implementado em duas procedures localizadas diretamente no banco de dados.

A primeira procedure, dijaddpath\_money(‘’,’’,0.0), é responsável por adicionar o no banco de dados o nó de origem, destino e custo no banco de dados. A segunda procedure intitulada de dijResolve\_money(‘’,’’) é responsável por fazer a consulta do nó de origem e nó destino, calculando a rota que possui o menor custo entre eles e retorna uma tabela temporária intitulada Map, onde há os dados de origem (FromNodeName) e destino (ToNodeName) e custo do caminho (Cost).

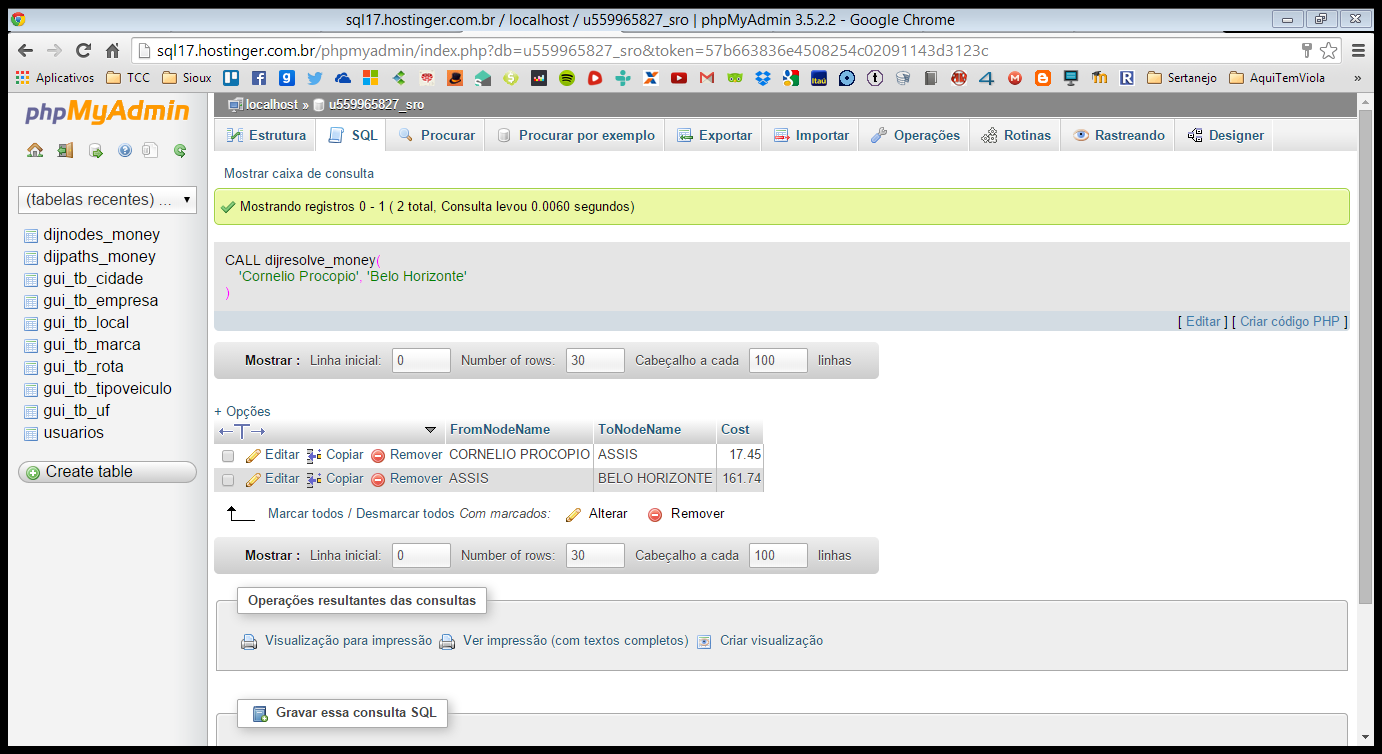


Figura 14 - Execução da procedure dijresolve\_money

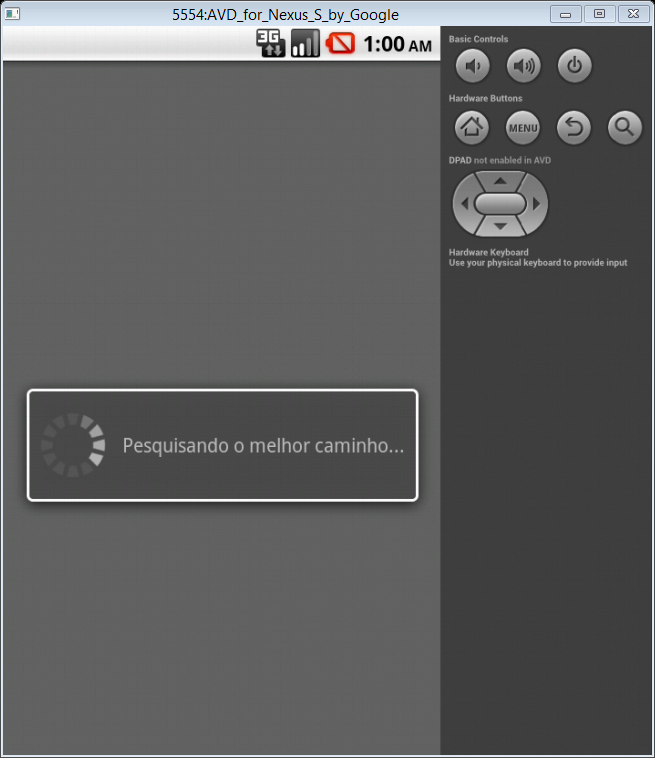


Figura 15 - Tela de Processamento do Algoritmo de Melhor Caminho

A primeira etapa do processamento da procedure dijresolve\_(‘Cornelio Procopio’,’Belo Horionte’); em backend é realizar uma busca entre os nós vizinhos do ponto de partida que é Cornélio Procópio, comparando os valores entre eles identificando a aresta que possui o menor custo. Deste modo algoritmo consegue verificar que o vértice Assis é a melhor opção. Partindo deste princípio o algoritmo realiza todo este procedimento para os nós vizinhos do vértice Assis, e assim até chegarmos no destino final, conforme podemos visualizar na Figura 19.

Para melhor entendimento, abaixo temos ilustrações de todos os grafos gerados com os passos realizados de acordo com o Algoritmo de Dijkstra:

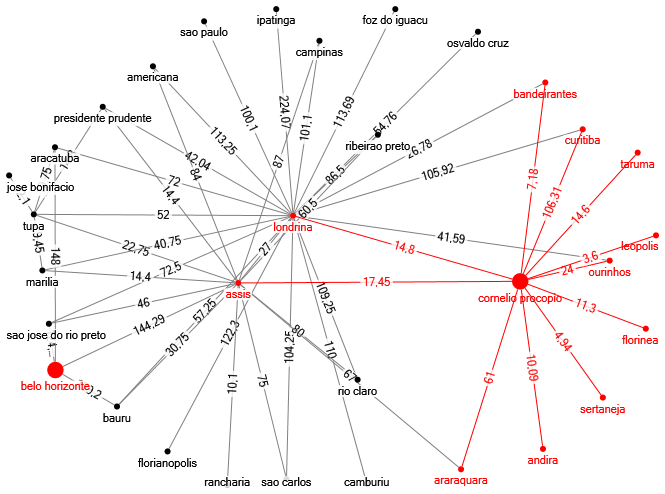


Figura 16 - Nós vizinhos com Cornélio Procópio (1º passo)

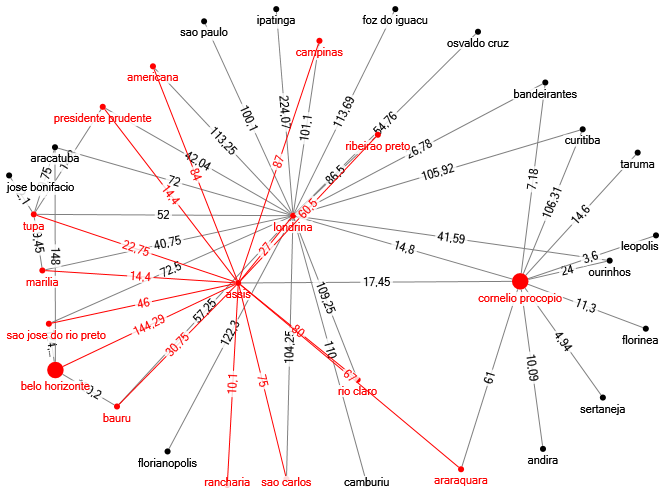


Figura 17 - Nós vizinhos com Assis (2º passo)

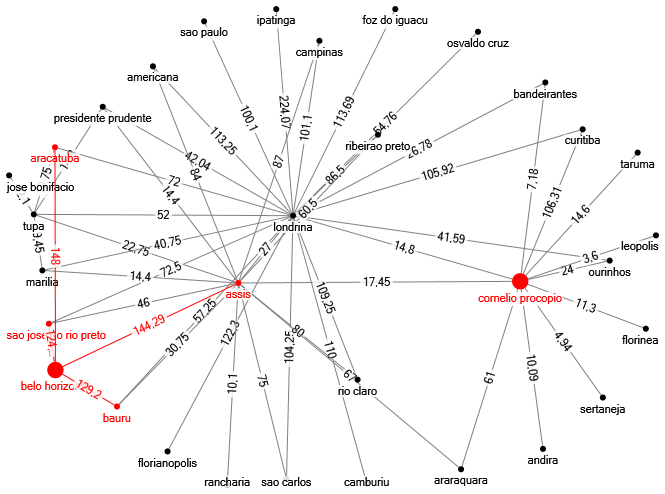


Figura 18 - Nós vizinhos com Belo Horizonte (3º passo)

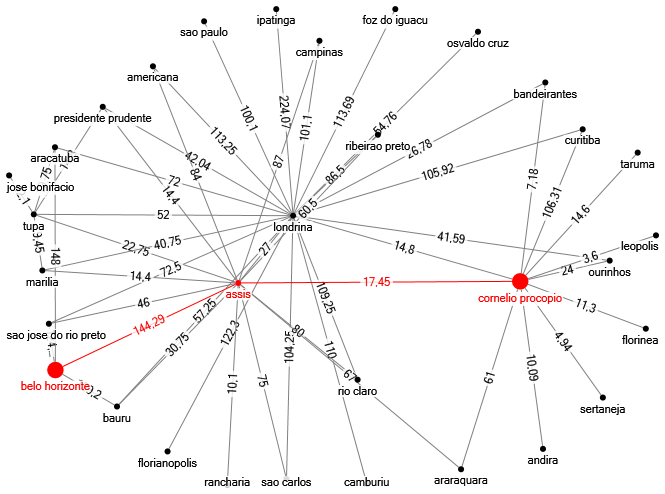


Figura 19 - Rota com menor custo

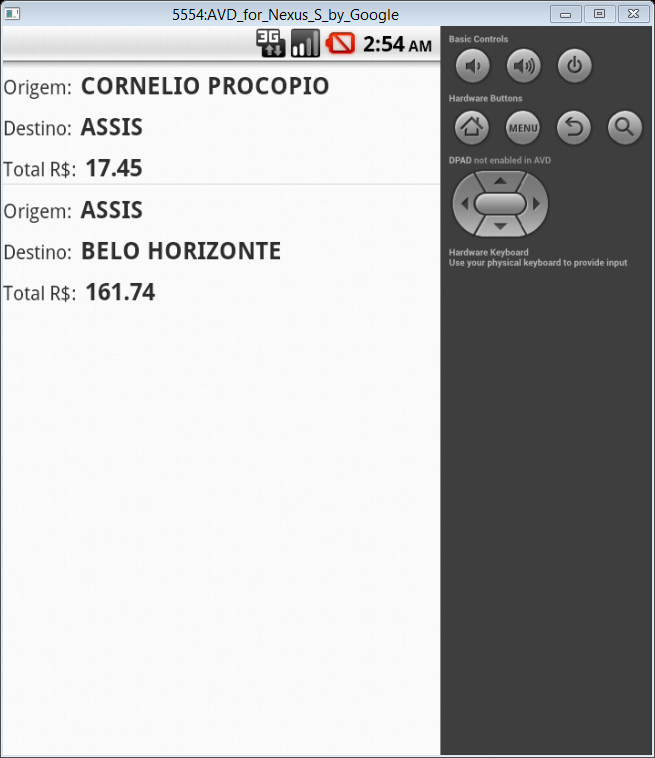


Figura 20 - Tela de resultado do caminho com menor custo

Pode-se concluir que, de acordo com os dados coletados e armazenados no banco de dados relacional, a rota com o custo mínimo partindo do vértice Cornélio Procópio com destino ao nó Belo Horizonte custa R$ 161.74. Uma economia média de cerca de R$ 40,00, se o passageiro utilizasse o caminho por Londrina em vez do indicado pelo algoritmo.

Tabela 8 - Rota alternativa CP/BH por Londrina/Bauru

|  |  |  |
| --- | --- | --- |
| **ORIGEM** | **DESTINO** | **CUSTO** |
| CORNELIO PROCOPIO | LONDRINA | 14,80 |
| LONDRINA | BAURU | 30,75 |
| BAURU | BELO HORIZONTE | 129,20 |
| **TOTAL** | | **174,75** |

Tabela 9 - Rota alternativa CP/BH por Londrina/Araçatuba

|  |  |  |
| --- | --- | --- |
| **ORIGEM** | **DESTINO** | **CUSTO** |
| CORNELIO PROCOPIO | LONDRINA | 14,80 |
| LONDRINA | ARACATUBA | 72,00 |
| ARACATUBA | BELO HORIZONTE | 148,00 |
| **TOTAL** | | **234,80** |

Tabela 10 - Rota alternativa CP/BH por Londrina/São José do Rio Preto

|  |  |  |
| --- | --- | --- |
| **ORIGEM** | **DESTINO** | **CUSTO** |
| CORNELIO PROCOPIO | LONDRINA | 14,80 |
| LONDRINA | SÃO JOSE DO RIO PRETO | 72,50 |
| SÃO JOSE DO RIO PRETO | BELO HORIZONTE | 124,00 |
| **TOTAL** | | **211,3** |

# 7. CONCLUSÕES FINAIS

7.1 Limitações de Hardware e Software

As restrições têm como base o sistema onde foi desenvolvida e testada a aplicação, podendo então ter uma margem de tolerância para mais ou para menos nos requisitos necessários ou nos testes obtidos em outros computadores de hardwares diferentes.

* Para o correto funcionamento da aplicação é indispensável a conexão do smartphone com a internet. Recomendado uma conexão com velocidade mínima.
* A aplicação exige a versão 2.2 (Frozen Yogurt) ou superior do Android.
* É necessário habilitar a opção de permitir a instalação de aplicativos de fontes desconhecidas ou possuir acesso a loja oficial do Google para a instalação da aplicação.

# 7.2 Dificuldades Encontradas

Foram encontradas diversas dificuldades durante o desenvolvimento, a mais grave ocasionou a retirada de um requisito do projeto. Por falta de conhecimentos adequados no desenvolvimento da aplicação necessários para torna-la a aplicação com maior usabilidade ao usuário. Requisito não implementado, tornaria possível a consulta de caminhos utilizando tempo da viajem como filtro de busca.

Possibilitando assim o usuário escolher entre o caminho com menor valor ou o que possui um tempo menor de viagem.

# 7.3 Trabalhos Futuros

A aplicação final será uma versão de teste, que poderá ser incrementada para uso profissional ou para fins acadêmicos, podendo ela ser comercializada ou então continuada para estudo e pesquisa.

O ideal seria realizar uma integração entre o banco de dados da aplica com as bases de dados das empresas de ônibus, assim obtendo uma maior quantidade de rotas melhorando desta forma a utilidade da aplicação.

Hoje a pessoa que utilizar a aplicação possui a limitação de apenas filtrar o caminho desejado por o menor custo. Para se tornar uma aplicação mais interessante e com finalidade de ser comercializada, a aplicação teria que possuir o filtro de menor tempo, possibilitando assim o usuário de escolher uma rota mais econômica ou que irá demorar menos para chegar em seu destino.

Em um futuro próximo, há necessidade de avaliar essas melhorias com mais detalhes e possivelmente serem desenvolvidas.

# 7.4 Validação do aplicativo

Por meio de questionário de avaliação que foi aplicado aos usuários que colaboraram com o período de teste da aplicação. Foi realizada uma análise dos resultados, permitindo atingir o objetivo central do que a aplicação propõe, sobre usabilidade. De acordo com a análise dos resultados sobre as questões de campo aberto, onde os usuários poderiam descrever de forma livre sobre sugestões, melhorias e críticas. Pode-se concluir que houve uma aceitação da aplicação pelos avaliados com uma ressalva sobre o filtro de menor tempo de viajem que ficou claro a necessidade desta funcionalidade não desenvolvida. Além desta sugestão, surgiram mais algumas novas funcionalidades novas e aprimoramento das funcionalidades já existentes.

Abaixo a Tabela 11 ilustra os resultados da análise dos questionários de avaliação do software.

Tabela 11 - Resultado da análise do questionário de avaliação

|  |  |
| --- | --- |
| **AVALIADO** | **NOTA** |
| Usuário 1 | 8.0 |
| Usuário 2 | 8.5 |
| Usuário 3 | 9.0 |
| Usuário 4 | 8.5 |
| **MÉDIA** | **8.5** |

# 8 CRONOGRAMA

Tabela 12 - Cronograma

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ATIVIDADES** | **ABR** | **MAI** | **JUN** | **JUL** | **AGO** | **SET** | **OUT** | **NOV** |
| Requisitos |  |  |  |  |  |  |  |  |
| Análise |  |  |  |  |  |  |  |  |
| Projeto |  |  |  |  |  |  |  |  |
| Codificação |  |  |  |  |  |  |  |  |
| Teste |  |  |  |  |  |  |  |  |
| Reparos |  |  |  |  |  |  |  |  |
| Escrita Final TD |  |  |  |  |  |  |  |  |
| TCC |  |  |  |  |  |  |  |  |

# 9 REFERÊNCIAS

**Arquitetura do Android.** Disponível em: < http://developer.android.com/images/system-architecture.jpg>, Acesso em: 8 de abr 2014.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **Unified Modeling Language User Guide.** Não Disponível: Addison Wesley, 1998. 512 p.

CARVALHO, P. S. Bruno Miguel. **ALGORITMO DE DIJKSTRA.** Universidade de Coimbra, Coimbra, Portugal. Disponível em: http://student.dei.uc.pt/~brunomig/cp/Artigo.pdf>, Acesso em: 8 de abr 2014.

CUTOLO, F. A. (2003) “**Diretrizes para sistema de informação ao usuário**”. In: III Seminário Internacional PROMOTEO, Porto Alegre-RS.

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA DA UNIVERSIDADE FEDERAL DE SANTA CATARINA. **Algoritmo de Dijkstra para cálculo do Caminho de Custo Mínimo** Disponível em: < http://www.inf.ufsc.br/grafos/temas/custo-minimo/dijkstra.html>, Acesso em: 8 de jan 2014.

**Desenvolvimento ágil utilizando Scrum**. Disponível em: < http://tableless.com.br/desenvolvimento-agil-utilizando-scrum/>. Acesso em: 6 de jan 2014.

DIJKSTRA, Edsger W. **A note on two problems in connexion with graphs.** **Numerische mathematik**, v. 1, n. 1, p. 269-271, 1959.

**Dijkstra’s shortest path algorithm** Disponível em: < http://www.artfulsoftware.com/infotree/qrytip.php?id=766>, Acesso em: 15 de nov 2014.

**History Android.** Disponível em: <http://www.android.com/history/>, Acesso em: 8 de nov 2014.

JOSUTTIS, Nicolai M. **SOA in practice: the art of distributed system design.** " O'Reilly Media, Inc.", 2007.

LACHETA, Ricardo R. **Google Android Aprenda a criar aplicações para dispositivos móveis com Android SDK.** 2ªEdição. São Paulo: Novatec, 2010. 608 p.

MARIANI, Antonio Carlos. **Teoria dos grafos**. Universidade Federal de Santa Catarina, Departamento de Informática e Estatística, 2001.

Melo, Jairo Simão Santana. **Arquitetura Orientada a Serviços Para Integração de Tecnologias Aplicadas a um Atlas Tridimensional Interativo da Anatomia Mamária.** 2012. 139 f. Tese (Doutorado em Engenharia Elétrica) – Faculdade de Tecnologia, Universidade de Brasília, Brasília.

**MYSQL**, A. B. ‘MySQL Documentation. Disponível em http://www.mysql. com/doc, 2000. Acesso em: 15 de nov 2014.

**NETBEANS**, I. D. E. Disponível em:< http://www. netbeans. org/>. Acesso em: 15 de nov 2014.

**Obtenha Informações sobre a Tecnologia Java**. Disponível em: <http://www.java.com/pt\_BR/about/>, Acesso em: 8 de jan 2014.

**PHP Documentation** Disponível em: < http://php.net/manual/pt\_BR/preface.php>, Acesso em: 15 de nov 2014.

RILDO F SANTOS. **SCRUM Experience**. Disponível em: < http://www.etecnologia.com.br/scrum/Scrum%20Experience%20%5BO%20Tutorial%20SCRUM%5D%20v16.pdf>. Acesso em: 6 de jan 2014.

RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. **Unified Modeling Language Reference Manual, The**. Pearson Higher Education, 2004.

**Unified Modeling Language (UML)** Disponível em: < http://www.deinf.ufma.br/~acmo/MOO\_Intro.pdf>, Acesso em: 15 de nov 2014.

ZHANG, J., WANG, F., WANG, K., Lin, W., Xu, X. and CHEN, C. (2011) “**Data-Driven Intelligent Transportation Systems: A Survey**”, In: IEEE Transactions on ntelligent Transportation Systems. vol. 12, no. 4, pp. 1624-1639.

# APÊNDICE A - Implementação das Procedures

**Procedure Dijaddpath\_Money**

1. DELIMITER $$
3. **CREATE** DEFINER=`root`@`localhost` **PROCEDURE** `dijaddpath\_money`(
4. pFromNodeName **VARCHAR**(20), pToNodeName **VARCHAR**(20), pCost **FLOAT**
5. )
6. **BEGIN**
7. **DECLARE** vFromNodeID, vToNodeID, vPathID **INT**;
8. **SET** vFromNodeID = ( **SELECT** NodeID **FROM** dijnodes\_money **WHERE** NodeName = pFromNodeName  );
9. IF vFromNodeID **IS** NULL **THEN**
10. **BEGIN**
11. **INSERT** **INTO** dijnodes\_money (NodeName,Calculated) **VALUES** (pFromNodeName,0);
12. **SET** vFromNodeID = LAST\_INSERT\_ID();
13. **END**;
14. **END** IF;
15. **SET** vToNodeID = ( **SELECT** NodeID **FROM** dijnodes\_money **WHERE** NodeName = pToNodeName );
16. IF vToNodeID **IS** NULL **THEN**
17. **BEGIN**
18. **INSERT** **INTO** dijnodes\_money(NodeName, Calculated)
19. **VALUES**(pToNodeName,0);
20. **SET** vToNodeID = LAST\_INSERT\_ID();
21. **END**;
22. **END** IF;
23. **SET** vPathID = ( **SELECT** PathID **FROM** dijpaths\_money
24. **WHERE** FromNodeID = vFromNodeID AND ToNodeID = vToNodeID
25. );
26. IF vPathID **IS** NULL **THEN**
27. **INSERT** **INTO** dijpaths\_money(FromNodeID,ToNodeID,Cost)
28. **VALUES**(vFromNodeID,vToNodeID,pCost);
29. **ELSE**
30. **UPDATE** dijpaths\_money **SET** Cost = pCost
31. **WHERE** FromNodeID = vFromNodeID AND ToNodeID = vToNodeID;
32. **END** IF;
33. **END**

**Procedure Dijresolve \_Money**

1. DELIMITER $$
3. **CREATE** DEFINER=`root`@`localhost` **PROCEDURE** `dijResolve\_money`( pFromNodeName **VARCHAR**(20), pToNodeName **VARCHAR**(20) )
4. **BEGIN**
5. **DECLARE** vFromNodeID, vToNodeID, vNodeID, vPathID **INT**;
6. **DECLARE** vCost **FLOAT**;
7. **DECLARE** vFromNodeName, vToNodeName **VARCHAR**(20);
9. **UPDATE** dijnodes\_money **SET** PathID = NULL,Cost = NULL,Calculated = 0;
11. **SET** vFromNodeID = ( **SELECT** NodeID **FROM** dijnodes\_money **WHERE** NodeName = pFromNodeName );
12. IF vFromNodeID **IS** NULL **THEN**
13. **SELECT** CONCAT('From node name ', pFromNodeName, ' not found.' );
14. **ELSE**
15. **BEGIN**
17. **SET** vNodeID = vFromNodeID;
18. **SET** vToNodeID = ( **SELECT** NodeID **FROM** dijnodes\_money **WHERE** NodeName = pToNodeName );
19. IF vToNodeID **IS** NULL **THEN**
20. **SELECT** CONCAT('From node name ', pToNodeName, ' not found.' );
21. **ELSE**
22. **BEGIN**
24. **UPDATE** dijnodes\_money **SET** Cost=0 **WHERE** NodeID = vFromNodeID;
25. WHILE vNodeID **IS** NOT NULL DO
26. **BEGIN**
27. **UPDATE**
28. dijnodes\_money **AS** src
29. JOIN dijpaths\_money **AS** paths **ON** paths.FromNodeID = src.NodeID
30. JOIN dijnodes\_money **AS** dest **ON** dest.NodeID = Paths.ToNodeID
31. **SET** dest.Cost = CASE
32. **WHEN** dest.Cost **IS** NULL **THEN** src.Cost + Paths.Cost
33. **WHEN** src.Cost + Paths.Cost < dest.Cost **THEN** src.Cost + Paths.Cost
34. **ELSE** dest.Cost
35. **END**,
36. dest.PathID = Paths.PathID
37. **WHERE**
38. src.NodeID = vNodeID
39. AND (dest.Cost **IS** NULL OR src.Cost + Paths.Cost < dest.Cost)
40. AND dest.Calculated = 0;
42. **UPDATE** dijnodes\_money **SET** Calculated = 1 **WHERE** NodeID = vNodeID;
44. **SET** vNodeID = ( **SELECT** nodeID **FROM** dijnodes\_money
45. **WHERE** Calculated = 0 AND Cost **IS** NOT NULL
46. **ORDER** **BY** Cost LIMIT 1
47. );
48. **END**;
49. **END** WHILE;
50. **END**;
51. **END** IF;
52. **END**;
53. **END** IF;
54. IF EXISTS( **SELECT** 1 **FROM** dijnodes\_money **WHERE** NodeID = vToNodeID AND Cost **IS** NULL ) **THEN**
55. -- problem,  cannot proceed
56. **SELECT** CONCAT( 'Node ',vNodeID, ' missed.' );
57. **ELSE**
58. **BEGIN**
59. -- write itinerary to map table
60. **DROP** **TEMPORARY** **TABLE** IF EXISTS map;
61. **CREATE** **TEMPORARY** **TABLE** map (
62. RowID **INT** **PRIMARY** **KEY** AUTO\_INCREMENT,
63. FromNodeName **VARCHAR**(20),
64. ToNodeName **VARCHAR**(20),
65. Cost **FLOAT**
66. ) ENGINE=MEMORY;
67. WHILE vFromNodeID <> vToNodeID DO
68. **BEGIN**
69. **SELECT**
70. src.NodeName,dest.NodeName,dest.Cost,dest.PathID
71. **INTO** vFromNodeName, vToNodeName, vCost, vPathID
72. **FROM**
73. dijnodes\_money **AS** dest
74. JOIN dijpaths\_money **AS** Paths **ON** Paths.PathID = dest.PathID
75. JOIN dijnodes\_money **AS** src **ON** src.NodeID = Paths.FromNodeID
76. **WHERE** dest.NodeID = vToNodeID;
78. **INSERT** **INTO** Map(FromNodeName,ToNodeName,Cost) **VALUES**(vFromNodeName,vToNodeName,vCost);
80. **SET** vToNodeID = (**SELECT** FromNodeID **FROM** dijpaths\_money **WHERE** PathID = vPathID);
81. **END**;
82. **END** WHILE;
83. **SELECT** FromNodeName,ToNodeName,Cost **FROM** Map **ORDER** **BY** RowID **DESC**;
84. **DROP** **TEMPORARY** **TABLE** Map;
85. **END**;
86. **END** IF;
87. **END**

# APÊNDICE B - Tabela De Dados Coletados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EMPRESA** | **TIPO** | **ORIGEM** | **DESTINO** | **CUSTO** |
| GUERINO SEISCENTO | CONVENCIONAL | AMERICANA | ASSIS | 84,00 |
| GUERINO SEISCENTO | CONVENCIONAL | AMERICANA | LONDRINA | 113,25 |
| GARCIA | CONVENCIONAL | ANDIRA | CORNELIO PROCOPIO | 10,09 |
| GONTIJO | CONVENCIONAL | ARACATUBA | BELO HORIZONTE | 148,00 |
| GARCIA | CONVENCIONAL | ARACATUBA | LONDRINA | 72,00 |
| GUERINO SEISCENTO | SUBURBANO | ARACATUBA | TUPA | 19,75 |
| GUERINO SEISCENTO | CONVENCIONAL | ARARAQUARA | ASSIS | 67,00 |
| GARCIA | CONVENCIONAL | ARARAQUARA | CORNELIO PROCOPIO | 61,00 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | AMERICANA | 84,00 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | ARARAQUARA | 67,00 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | BAURU | 30,75 |
| GONTIJO | CONVENCIONAL | ASSIS | BELO HORIZONTE | 144,29 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | CAMPINAS | 87,00 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | CORNELIO PROCOPIO | 17,45 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | LONDRINA | 27,00 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | MARILIA | 14,40 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | PRESIDENTE PRUDENTE | 14,40 |
| GUERINO SEISCENTO | SUBURBANO | ASSIS | RANCHARIA | 10,10 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | RIBEIRAO PRETO | 60,50 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | RIO CLARO | 80,00 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | SAO CARLOS | 75,00 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | SAO JOSE DO RIO PRETO | 46,00 |
| GUERINO SEISCENTO | CONVENCIONAL | ASSIS | TUPA | 22,75 |
| GARCIA | CONVENCIONAL | BANDEIRANTES | CORNELIO PROCOPIO | 7,18 |
| GARCIA | CONVENCIONAL | BANDEIRANTES | LONDRINA | 26,78 |
| GUERINO SEISCENTO | CONVENCIONAL | BAURU | ASSIS | 30,75 |
| GONTIJO | CONVENCIONAL | BAURU | BELO HORIZONTE | 129,20 |
| GUERINO SEISCENTO | CONVENCIONAL | BAURU | LONDRINA | 57,25 |
| GONTIJO | CONVENCIONAL | BELO HORIZONTE | ARACATUBA | 148,00 |
| GONTIJO | CONVENCIONAL | BELO HORIZONTE | ASSIS | 144,29 |
| GONTIJO | CONVENCIONAL | BELO HORIZONTE | BAURU | 129,20 |
| GONTIJO | CONVENCIONAL | BELO HORIZONTE | SAO JOSE DO RIO PRETO | 124,92 |
| GARCIA | CONVENCIONAL | CAMBURIU | LONDRINA | 110,00 |
| GUERINO SEISCENTO | CONVENCIONAL | CAMPINAS | ASSIS | 87,00 |
| GARCIA | CONVENCIONAL | CAMPINAS | LONDRINA | 101,10 |
| GARCIA | CONVENCIONAL | CORNELIO PROCOPIO | ANDIRA | 10,09 |
| GARCIA | CONVENCIONAL | CORNELIO PROCOPIO | ARARAQUARA | 61,00 |
| GUERINO SEISCENTO | CONVENCIONAL | CORNELIO PROCOPIO | ASSIS | 17,45 |
| GARCIA | CONVENCIONAL | CORNELIO PROCOPIO | BANDEIRANTES | 7,18 |
| GARCIA | EXECUTIVO | CORNELIO PROCOPIO | CURITIBA | 106,31 |
| GUERINO SEISCENTO | CONVENCIONAL | CORNELIO PROCOPIO | FLORINEA | 11,30 |
| GUERINO SEISCENTO | CONVENCIONAL | CORNELIO PROCOPIO | LEOPOLIS | 3,60 |
| GARCIA | CONVENCIONAL | CORNELIO PROCOPIO | LONDRINA | 14,80 |
| GARCIA | CONVENCIONAL | CORNELIO PROCOPIO | OURINHOS | 24,00 |
| GUERINO SEISCENTO | CONVENCIONAL | CORNELIO PROCOPIO | SERTANEJA | 4,94 |
| GUERINO SEISCENTO | CONVENCIONAL | CORNELIO PROCOPIO | TARUMA | 14,60 |
| GARCIA | EXECUTIVO | CURITIBA | CORNELIO PROCOPIO | 106,31 |
| GARCIA | CONVENCIONAL | CURITIBA | LONDRINA | 105,92 |
| GARCIA | CONVENCIONAL | FLORIANOPOLIS | LONDRINA | 122,30 |
| GUERINO SEISCENTO | CONVENCIONAL | FLORINEA | CORNELIO PROCOPIO | 11,30 |
| GARCIA | CONVENCIONAL | FOZ DO IGUACU | LONDRINA | 113,69 |
| GONTIJO | CONVENCIONAL | IPATINGA | LONDRINA | 224,07 |
| GUERINO SEISCENTO | CONVENCIONAL | JOSE BONIFACIO | TUPA | 42,10 |
| GUERINO SEISCENTO | CONVENCIONAL | LEOPOLIS | CORNELIO PROCOPIO | 3,60 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | AMERICANA | 113,25 |
| GARCIA | CONVENCIONAL | LONDRINA | ARACATUBA | 72,00 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | ASSIS | 27,00 |
| GARCIA | CONVENCIONAL | LONDRINA | BANDEIRANTES | 26,78 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | BAURU | 57,25 |
| GARCIA | CONVENCIONAL | LONDRINA | CAMBURIU | 110,00 |
| GARCIA | CONVENCIONAL | LONDRINA | CAMPINAS | 101,10 |
| GARCIA | CONVENCIONAL | LONDRINA | CORNELIO PROCOPIO | 14,80 |
| GARCIA | CONVENCIONAL | LONDRINA | CURITIBA | 105,92 |
| GARCIA | CONVENCIONAL | LONDRINA | FLORIANOPOLIS | 122,30 |
| GARCIA | CONVENCIONAL | LONDRINA | FOZ IGUACU | 113,69 |
| GONTIJO | CONVENCIONAL | LONDRINA | IPATINGA | 224,07 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | MARILIA | 40,75 |
| GARCIA | CONVENCIONAL | LONDRINA | OSVALDO CRUZ | 54,76 |
| GARCIA | CONVENCIONAL | LONDRINA | OURINHOS | 41,59 |
| GARCIA | CONVENCIONAL | LONDRINA | PRESIDENTE PRUDENTE | 42,04 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | RIBEIRAO PRETO | 86,50 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | RIO CLARO | 109,25 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | SAO CARLOS | 104,25 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | SAO JOSE DO RIO PRETO | 72,50 |
| GARCIA | CONVENCIONAL | LONDRINA | SAO PAULO | 100,10 |
| GUERINO SEISCENTO | CONVENCIONAL | LONDRINA | TUPA | 52,00 |
| GUERINO SEISCENTO | CONVENCIONAL | MARILIA | ASSIS | 14,40 |
| GUERINO SEISCENTO | CONVENCIONAL | MARILIA | LONDRINA | 40,75 |
| GUERINO SEISCENTO | SUBURBANO | MARILIA | TUPA | 9,45 |
| GARCIA | CONVENCIONAL | OSVALDO CRUZ | LONDRINA | 54,76 |
| GARCIA | CONVENCIONAL | OURINHOS | CORNELIO PROCOPIO | 24,00 |
| GARCIA | CONVENCIONAL | OURINHOS | LONDRINA | 41,59 |
| GUERINO SEISCENTO | CONVENCIONAL | PRESIDENTE PRUDENTE | ASSIS | 14,40 |
| GARCIA | CONVENCIONAL | PRESIDENTE PRUDENTE | LONDRINA | 42,04 |
| GUERINO SEISCENTO | SUBURBANO | PRESIDENTE PRUDENTE | TUPA | 14,80 |
| GUERINO SEISCENTO | SUBURBANO | RANCHARIA | ASSIS | 10,10 |
| GUERINO SEISCENTO | CONVENCIONAL | RIBEIRAO PRETO | ASSIS | 60,50 |
| GUERINO SEISCENTO | CONVENCIONAL | RIBEIRAO PRETO | LONDRINA | 86,50 |
| GUERINO SEISCENTO | CONVENCIONAL | RIO CLARO | ASSIS | 80,00 |
| GUERINO SEISCENTO | CONVENCIONAL | RIO CLARO | LONDRINA | 109,25 |
| GUERINO SEISCENTO | CONVENCIONAL | SAO CARLOS | ASSIS | 75,00 |
| GUERINO SEISCENTO | CONVENCIONAL | SAO CARLOS | LONDRINA | 104,25 |
| GUERINO SEISCENTO | CONVENCIONAL | SAO JOSE DO RIO PRETO | ASSIS | 46,00 |
| GONTIJO | CONVENCIONAL | SAO JOSE DO RIO PRETO | BELO HORIZONTE | 124,92 |
| GUERINO SEISCENTO | CONVENCIONAL | SAO JOSE DO RIO PRETO | LONDRINA | 72,50 |
| GARCIA | CONVENCIONAL | SAO PAULO | LONDRINA | 100,10 |
| GUERINO SEISCENTO | CONVENCIONAL | SERTANEJA | CORNELIO PROCOPIO | 4,94 |
| GUERINO SEISCENTO | CONVENCIONAL | TARUMA | CORNELIO PROCOPIO | 14,60 |
| GUERINO SEISCENTO | SUBURBANO | TUPA | ARACATUBA | 19,75 |
| GUERINO SEISCENTO | CONVENCIONAL | TUPA | ASSIS | 22,75 |
| GUERINO SEISCENTO | CONVENCIONAL | TUPA | JOSE BONIFACIO | 42,10 |
| GUERINO SEISCENTO | CONVENCIONAL | TUPA | LONDRINA | 52,00 |
| GUERINO SEISCENTO | SUBURBANO | TUPA | MARILIA | 9,45 |
| GUERINO SEISCENTO | SUBURBANO | TUPA | PRESIDENTE PRUDENTE | 14,80 |

# 

# APÊNDICE C - Questionário De Avaliação Da Aplicação

Questionário de Avaliação do Sistema

Sistema de Rotas de Ônibus

**Nome:**

1. **O sistema possui uma interface de fácil entendimento?**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **As informações são que apresentadas no resultado são coerentes?**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Foi possível obter o caminho desejado de forma rápida e prática?**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Qual versão do Android seu smartphone possui?**

* Android 2.2 – Froyo
* Android 2.3 – Gingerbread
* Android 3.0/3.1/3.2 – Honeycomb
* Android 4.0 – Ice Cream Sandwich
* Android 4.1/4.2/4.3 - Jelly Bean
* Android 4.4 – KitKat
* Outro

1. **O que você sugere que seja feito para evolução e aprimoramento da aplicação?**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **De uma nota de 0 (zero) a 10 (dez) ao sistema.**

**Nota: \_\_\_\_\_\_**