



**MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CAMPUS CORNÉLIO PROCÓPIO
DEPARTAMENTO DE ANÁLISE E DESENVOLVIMENTO
DE SISTEMA**



LEONARDO TEIXEIRA VIRGILIO

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

TRABALHO DE DIPLOMAÇÃO

**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

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.

Cornélio Procópio, 20 de novembro de 2014

BANCA EXAMINADORA

Prof.

Prof.

Prof.

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

Albert Einstein

DEDICATÓRIA

Primeiramente à Deus e 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 suportado 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.

RESUMO

Sistema para rotas ônibus utilizando o algoritmo de Dijkstra (Melhor Caminho). 44 páginas. Tema de Trabalho de Diplomação (Tecnologia em Análise e Desenvolvimento de Sistemas). Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2014. Este trabalho propõe o desenvolvimento de um sistema do qual seja capaz de reconhecer e mapear o caminho com o menor custo, bem como inserir, alterar e excluir os caminhos armazenados bem como desenvolver 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 Scrum como metodologia para gerenciamento de cronograma.

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

.

ABSTRACT

System of bus routes using Dijkstra's algorithm (the Best Way). 38 pages. Graduation's work theme (Technology Analysis and Development of Systems). Federal Technological University of Paraná. Cornélio Procópio, 2014. This work proposes the development of a system which can be able to recognize and map the course with a lower cost, as well as inserting, changing and deleting stored routes, and also the development of an application for the Android Platform that will present the data of a route with the lowest cost. In this work will be used Java programming language, PHP with Netbeans IDE, Eclipse, MySQL database and Scrum as methodology for schedule management.

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

LISTA DE FIGURAS

Figura 1 - Grafo Cornélio Procópio - Belo Horizonte.....	15
Figura 2 - Nós vizinhos com Cornélio Procópio (1º passo).....	17
Figura 3 - Nós vizinhos com Assis (2º passo).....	18
Figura 4 - Nós vizinhos com Belo Horizonte (3º passo).....	19
Figura 5 - Rota com menor custo	20
Figura 6 - Arquitetura do Sistema.....	27
Figura 8 - Ilustração do Ciclo de Vida do SCRUM	34

LISTA DE TABELAS

Tabela 1 - Rota alternativa CP/BH por Londrina/Bauru.....	21
Tabela 2 - Rota alternativa CP/BH por Londrina/Araçatuba	21
Tabela 3 - Rota alternativa CP/BH por Londrina/São José do Rio Preto.....	21
Tabela 4 - Tabela de Funcionalidades do Sistema	22
Tabela 5 - Cronograma	35

LISTA DE ABREVIATURAS

SRO	Sistema de Rotas de Ônibus
UTFPR	Universidade Tecnológica Federal do Paraná
UFSC	Universidade Federal de Santa Catarina
SISU	Sistema de Seleção Unificada
ENEM	Exame Nacional de Ensino Médio
CP	Cornélio Procópio
SP	São Paulo
MG	Minas Gerais
PR	Paraná
JVM	Java Virtual Machine
PHP	Personal Home Page
ADT	Android Developers Tools
IDE	Integrated Development Environment
XML	Extensible Markup Language
SGBD	Sistema de Gerenciamento de Banco de Dados

SUMÁRIO

1 INTRODUÇÃO	12
2 JUSTIFICATIVA.....	13
3 OBJETIVOS.....	14
3.1 OBJETIVO GERAL	14
4 VISÃO GERAL DO TRABALHO DE CONCLUSÃO	16
5 ESCOPO DA APLICAÇÃO	22
5.1 DEFINIÇÃO DE ESCOPO	22
6 LIMITAÇÕES E RESTRIÇÕES DA APLICAÇÃO	24
6.1 LIMITAÇÕES	24
6.2 LIMITAÇÕES DE HARDWARE E SOFTWARE	24
7 DETALHES DA APLICAÇÃO	25
7.1 LINGUAGENS	25
7.1.1 JAVA.....	25
7.1.2 PHP	25
7.2 FERRAMENTAS.....	26
7.2.1 ECLIPSE + ANDROID DEVELOPER TOOLS	26
7.2.2 SUBLIME TEXT	26
7.2.3 MYSQL.....	26
7.2.4 NETBEANS.....	27
7.2 ARQUITETURA DO SISTEMA.....	27
7.3 ESPECIFICAÇÃO DE FUNCIONALIDADE	28
7.3.1 DIAGRAMA DE CASO DE USO	29
7.3.2 DIAGRAMA DE CLASSE.....	30
7.3.3 DIAGRAMA DE ENTIDADE E RELACIONAMENTO.....	31
8 METODOLOGIA DE DESENVOLVIMENTO.....	32
9 CRONOGRAMA	35
10 APÊNDICE A	37
10 APÊNDICE B	40
11 REFERÊNCIAS.....	ERRO! INDICADOR NÃO DEFINIDO.

1 INTRODUÇÃO

Este documento tem por objetivo apresentar um trabalho de diplomação do curso de Tecnologia em Análise e Desenvolvimento de Sistema da Universidade Tecnológica Federal do Paraná do Campus Cornélio Procópio (UTFPR-CP). O tema apresentado 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. 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 professor do Departamento de Informática e Estatística da UFSC (Universidade Federal de Santa Catarina):

O Algoritmo de Dijkstra (E.W. Dijkstra) 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 os limites e restrições da solução, as tecnologias a serem utilizadas e o cronograma que será obedecido.

2 JUSTIFICATIVA

Hoje em dia com o Sistema de Seleção Unificada (SISU), sistema informatizado gerenciado pelo Ministério da Educação no qual instituições públicas de ensino superior oferecem vagas para candidatos participantes do Exame Nacional de Ensino Médio (ENEM). Está cada vez mais comum a migração de estudantes para diversos lugares do Brasil. Muitos alunos estudam em universidades que ficam localizadas a mais de mil quilômetros de distância de 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 do ônibus que faz esse caminho diretamente.

3 OBJETIVOS

3.1 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.

O sistema funcionará da seguinte maneira:

Uma pessoa que precisa viajar de Cornélio Procopio – PR para a cidade de Belo Horizonte – MG a distância entre as duas cidades é de cerca 1034 KM (*Google Maps*) dependendo do caminho, essa pessoa leva o tempo de mais o menos 24 horas de viagem e gasta cerca 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.

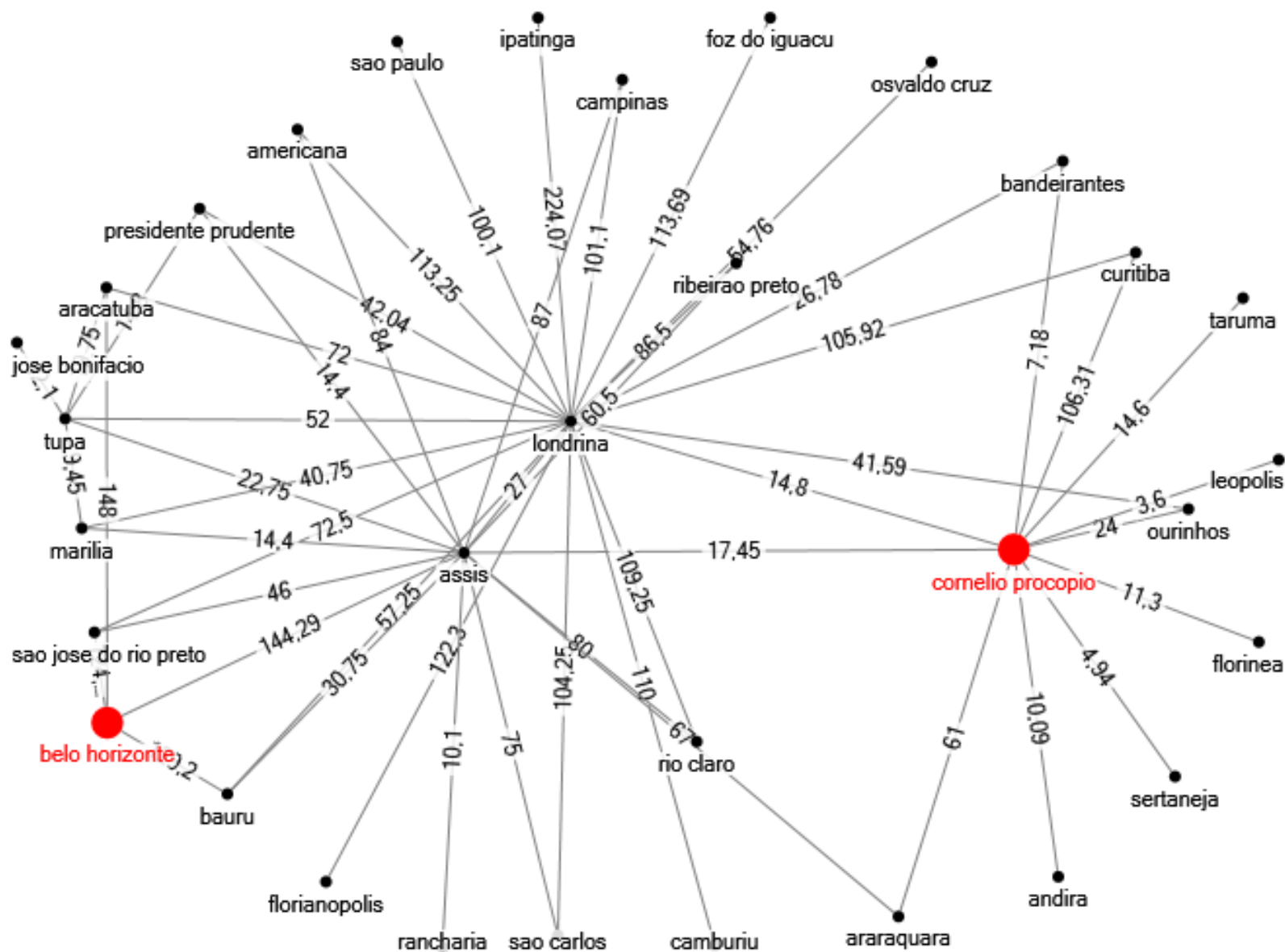


Figura 1 - Grafo Cornélio Procópio - Belo Horizonte

4 VISÃO GERAL DO TRABALHO DE CONCLUSÃO

Cenário proposto acima em simular uma viagem de ônibus com ponto de partida a cidade de Cornélio Procópio tendo seu destino final a capital do estado de Minas Gerais, Belo Horizonte. Para que esse cenário pudesse se tornar mais próximo do real, foi coletado dados como custos, tipo de veículo e tempo de viagem de algumas rotas de determinadas empresas.

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 foi manual, 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 serão explicados no tópico Detalhes e Limitações da Aplicação.

A partir dos dados coletados podemos 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 ressaltar que cada ligação entre os nós (cidades) há um valor intitulado de “Custo”. O “Custo” são os preços das passagens entre aquelas cidades. Tendo conhecimento desta informação o Algoritmo de Dijkstra tenta mapear uma rota com menor custo entre os pontos de origem e destino.

Primeiramente o algoritmo realiza 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. Identificando 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 5.

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

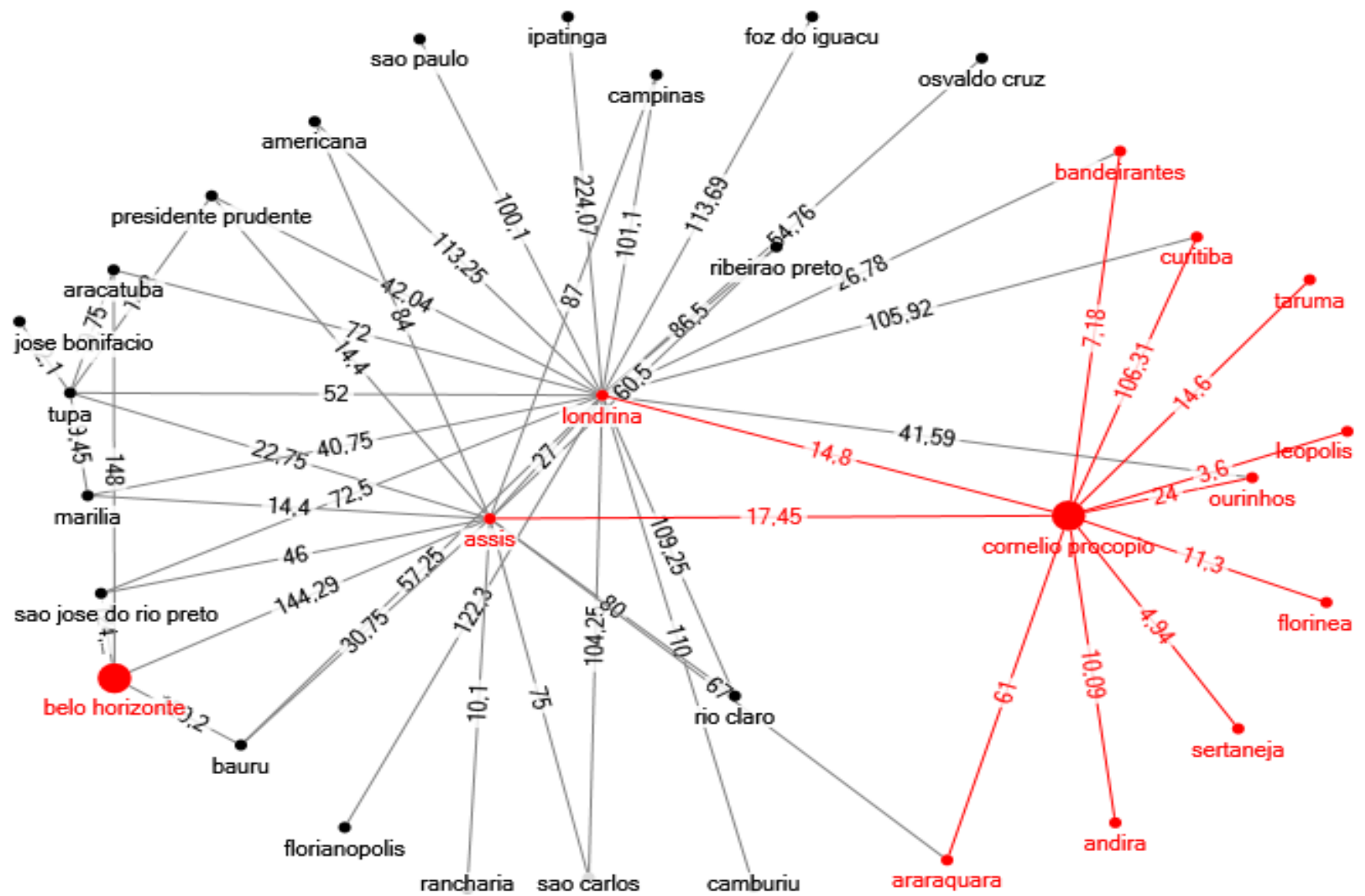


Figura 2 - Nós vizinhos com Cornélio Procópio (1º passo)

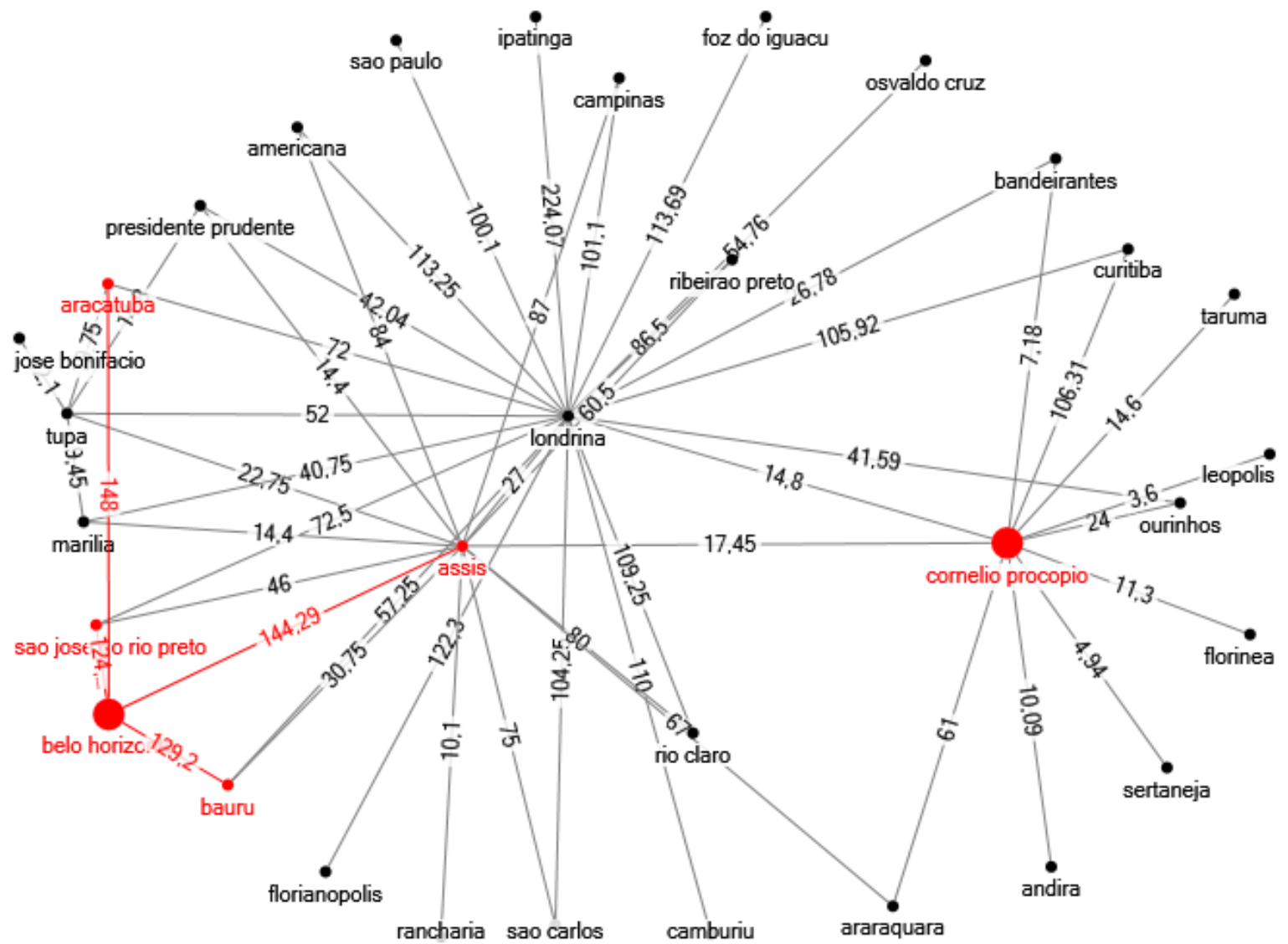


Figura 4 - Nós vizinhos com Belo Horizonte (3º passo)

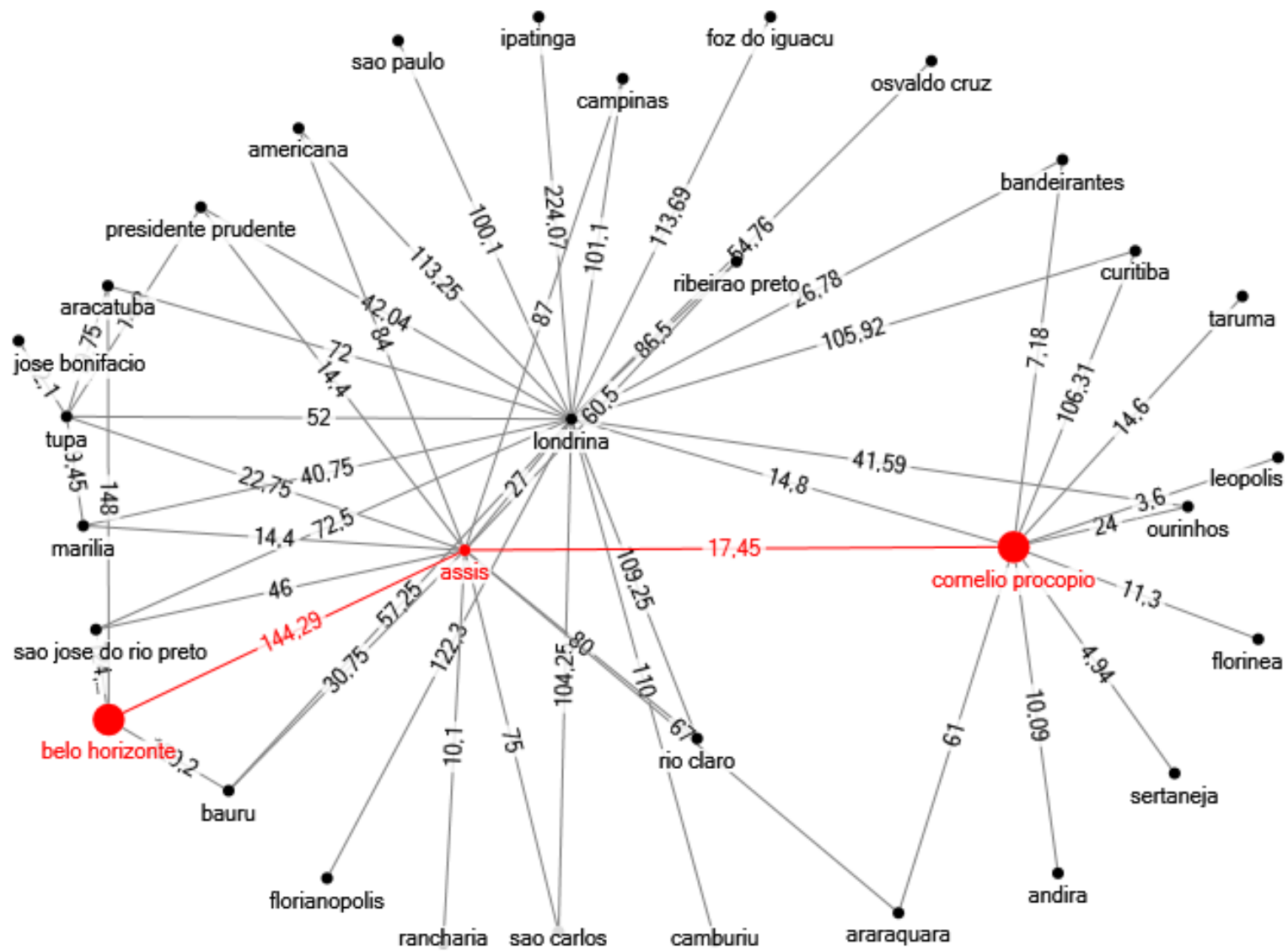


Figura 5 - Rota com menor custo

Pode-se concluir que, de acordo com os dados coletados e armazenados em nosso 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.

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

Tabela 1 - Rota alternativa CP/BH por Londrina/Bauru

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

Tabela 2 - Rota alternativa CP/BH por Londrina/Araçatuba

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

Tabela 3 - Rota alternativa CP/BH por Londrina/São José do Rio Preto

5 ESCOPO DA APLICAÇÃO

5.1 Definição de escopo

Nesta seção 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.

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

Tabela 4 - Tabela de Funcionalidades do Sistema

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 administrativa 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 administrativa 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 administrativa 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 administrativa 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.

6 LIMITAÇÕES E RESTRIÇÕES DA APLICAÇÃO

6.1 Limitações

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.

6.2 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 DETALHES DA APLICAÇÃO

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

7.1 Linguagens

7.1.1 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++.

7.1.2 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.

7.2 Ferramentas

7.2.1 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 [ADT, 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.

7.2.2 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`.

7.2.3 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.

7.2.4 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.

7.2 Arquitetura do Sistema

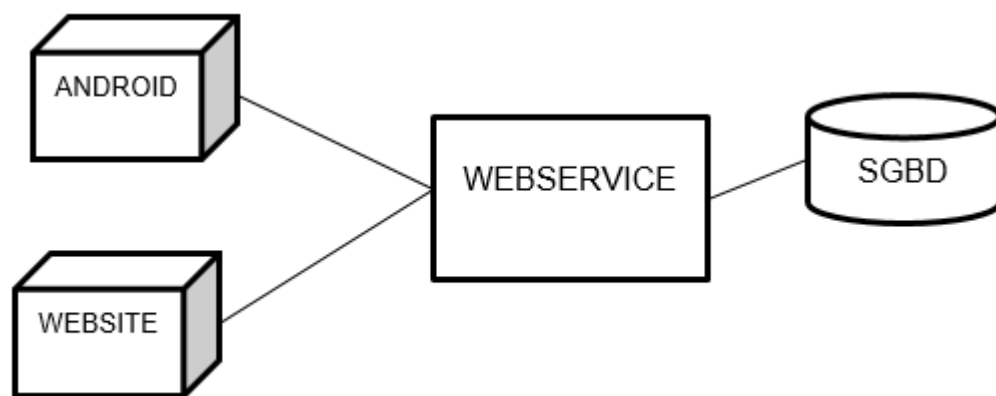


Figura 6 - Arquitetura do Sistema

As camadas que formam a arquitetura de software são descritas a seguir:

- *Android*: Toda a aplicação em *Android*, desde as *activities* que são responsáveis pela representação gráfica, utilizando componentes do tipo *textview*, *editview*, *listview*, *button*, implementadas em XML, até as classes implementadas em Java para realizar a conexão com o servidor.
- *Website*: Arquivos html, css, js e php que compõe toda a parte administrativa da aplicação.
- *Webservice*: Responsável pela comunicação entre a aplicação *Android* e o SGBD.
- *SGBD*: Responsável pelo gerenciamento das informações persistidas na aplicação.

7.3 Especificação de Funcionalidade

Nesta seção são apresentados o diagrama de classe (Figura 7), diagrama de casos de uso (Figura 8) e diagrama de entidade relacionamento (Figura 9).

7.3.1 Diagrama de Caso de Uso

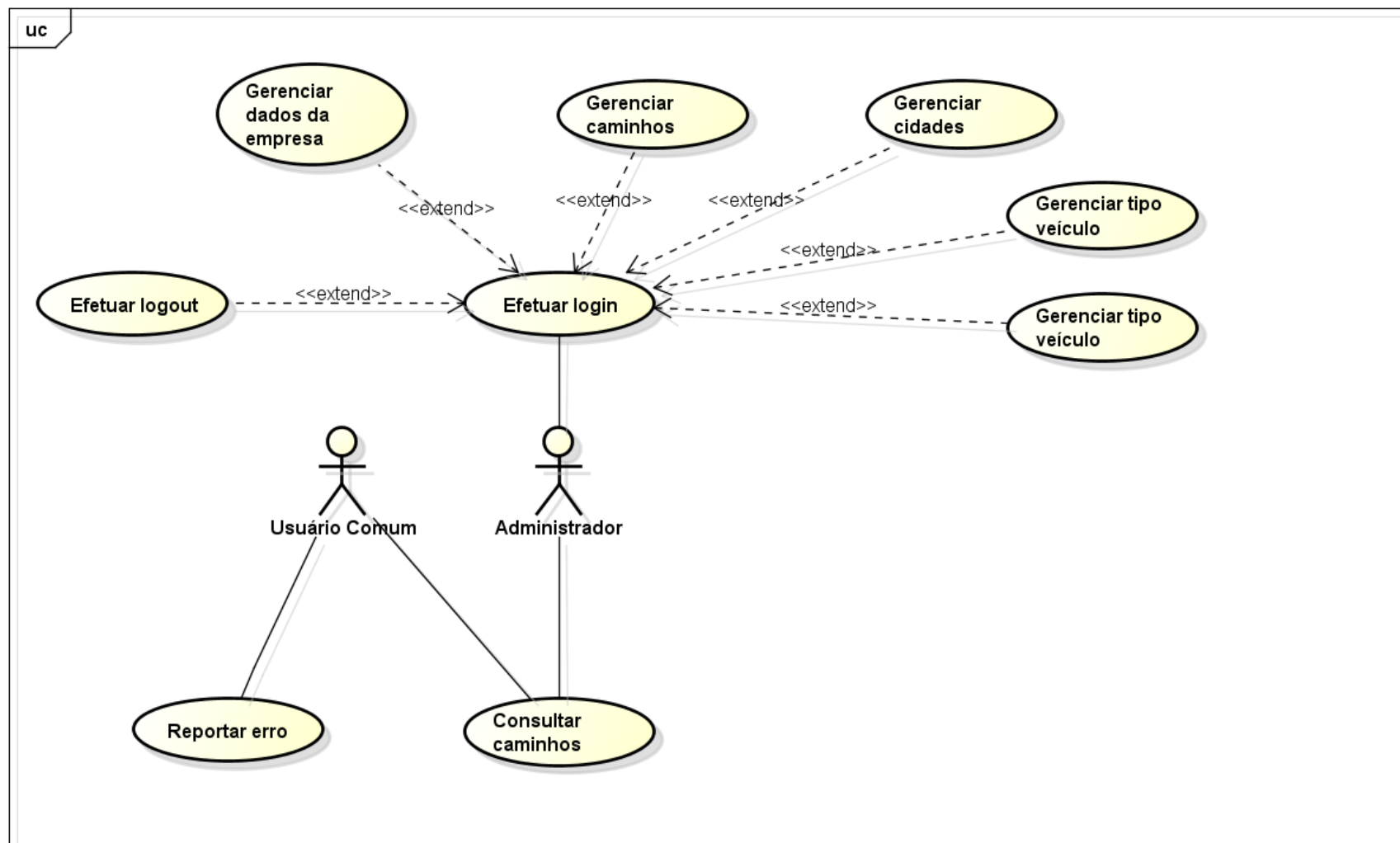


Figura 7 - Diagrama de Caso de Uso

7.3.2 Diagrama de Classe

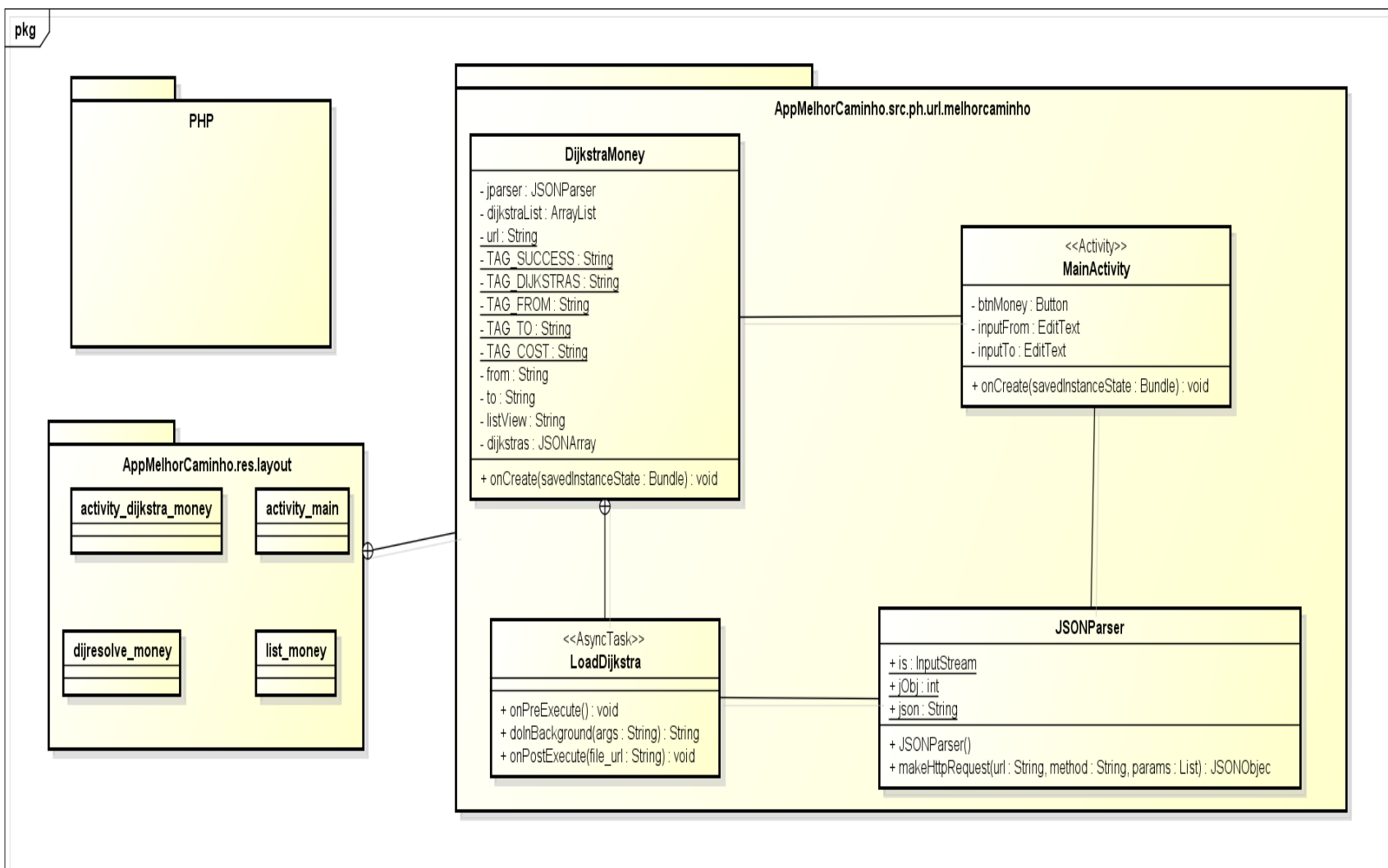


Figura 8 - Diagrama de Classe

7.3.3 Diagrama de Entidade e Relacionamento

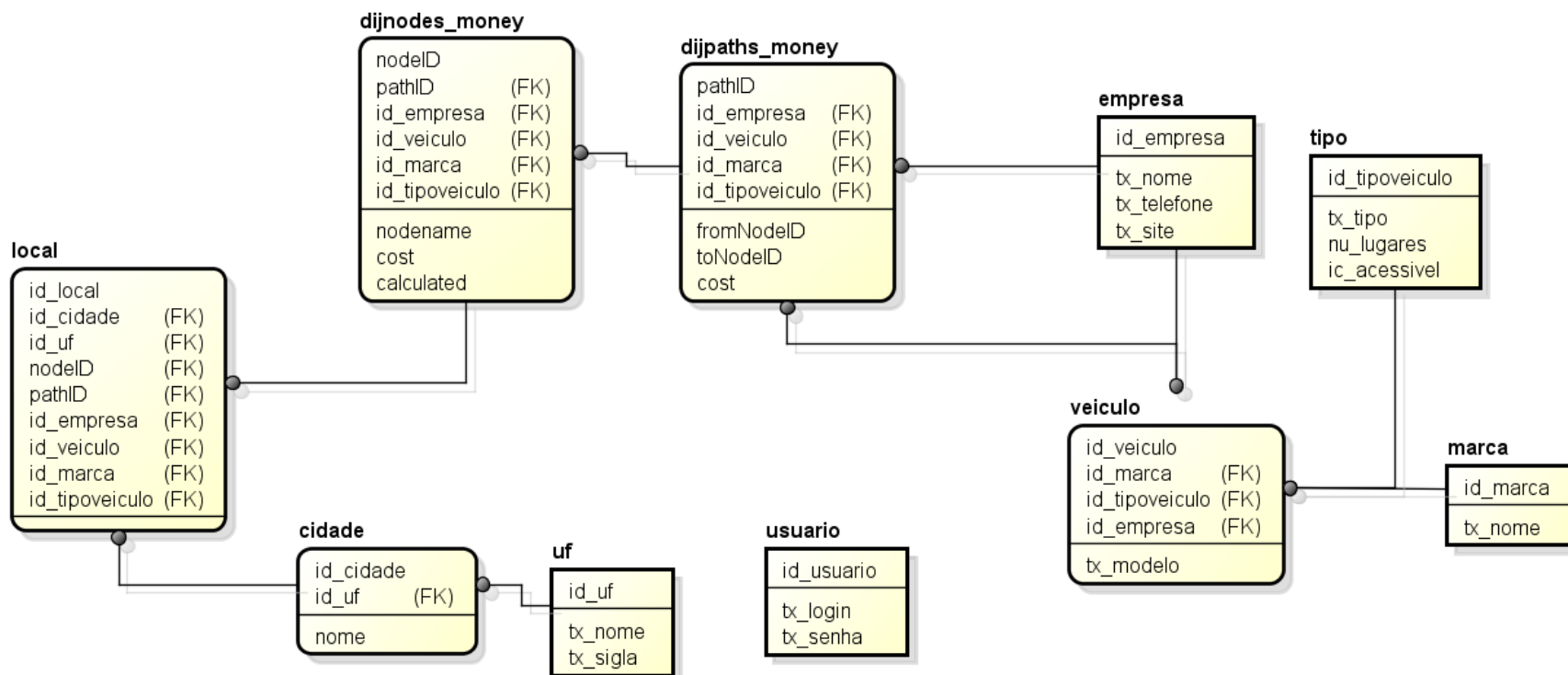


Figura 9 - Diagrama de Entidade e Relacionamento

8 METODOLOGIA DE DESENVOLVIMENTO

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

O Scrum é um framework de para resoluções de problemas, nada mais nada menos que ferramentas para auxiliar à gerenciar um projeto, desenvolvido por Ken Schwaber e Jeff Stherland sobre a necessidade de encontrar uma metodologia que abordasse o problema do desenvolvimento de software de forma não tradicional.

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.

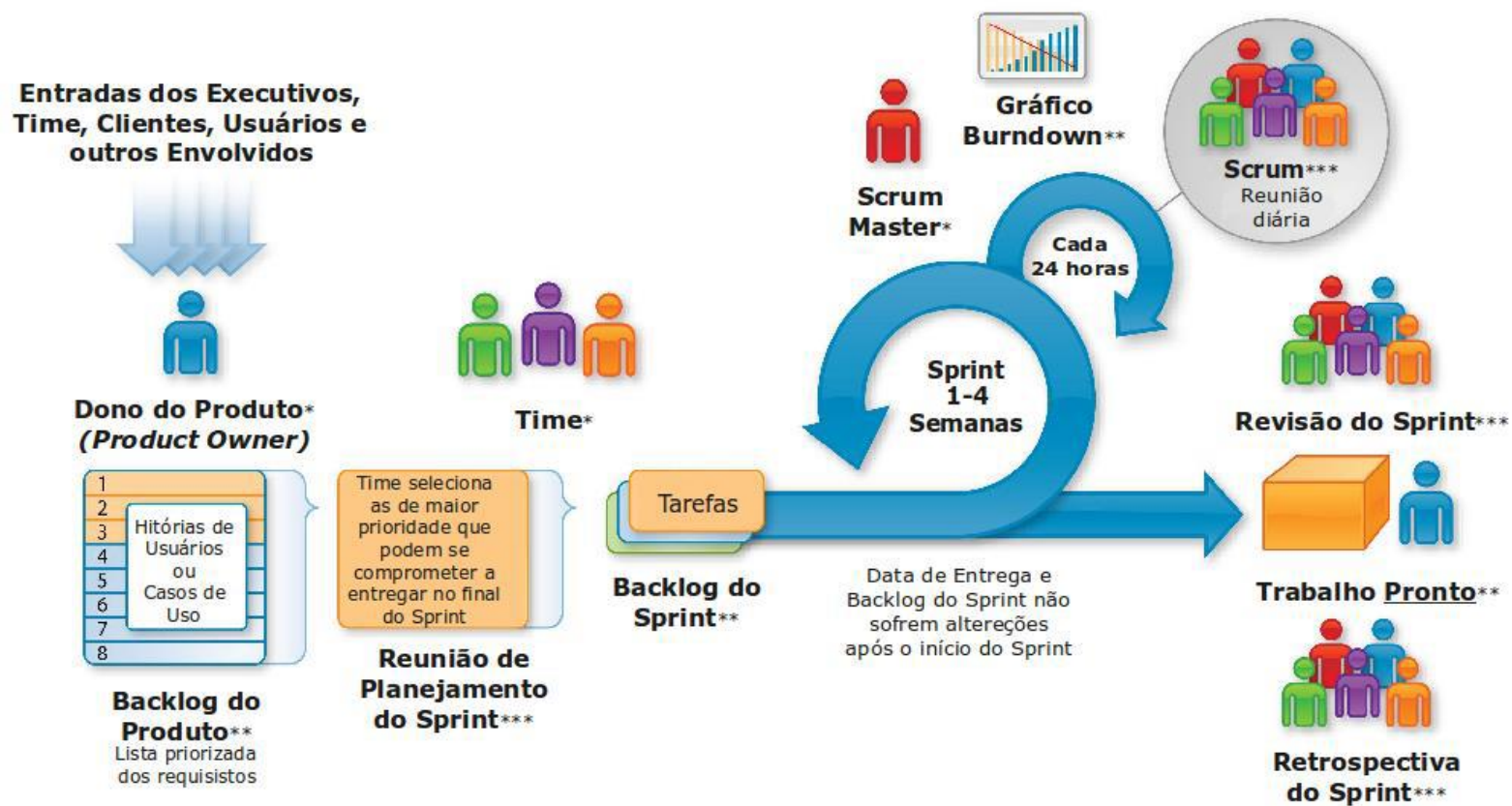
Segundo (Soares, 2004) o Scrum pode ser utilizado por pequenas ou grandes equipes, porém nada impede de ser utilizado por uma pessoa só. Só que neste caso você deve cumprir todos os papéis do clico de vida (veja a Figura 1).

O Scrum possui a seguinte hierarquia:

- ProductOwner – E 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 realizados 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? O objetivo 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.



* Papel, ** Artefato, *** Cerimônia

Figura 10 - Ilustração do Ciclo de Vida do SCRUM

9 CRONOGRAMA

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

Tabela 5 - Cronograma

10 REFERÊNCIAS

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

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

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

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

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.

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.

10 APÊNDICE A

Segue abaixo as implementações para do Algoritmo de Dijkstra:

PROCEDURE DIJADDPATH_MONEY

```

1. DELIMITER $$
2.
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. );
10.  IF vFromNodeID IS NULL THEN
11.    BEGIN
12.      INSERT INTO dijnodes_money (NodeName, Calculated) VALUES (pFromNodeName, 0);
13.      SET vFromNodeID = LAST_INSERT_ID();
14.    END;
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 $$
2.
3. CREATE DEFINER=`root`@`localhost` PROCEDURE `dijResolve_money`( pFromNodeName VARCHAR(2
4.   0), pToNodeName VARCHAR(20) )
5. BEGIN
6.   DECLARE vFromNodeID, vToNodeID, vNodeID, vPathID INT;
7.   DECLARE vCost FLOAT;

```

```

7.  DECLARE vFromNodeName, vToNodeName VARCHAR(20);
8.  -- null out path info in the nodes table
9.  UPDATE dijnodes_money SET PathID = NULL, Cost = NULL, Calculated = 0;
10. -- find nodeIDs referenced by input params
11. SET vFromNodeID = ( SELECT NodeID FROM dijnodes_money WHERE NodeName = pFromNodeName
12. );
13. IF vFromNodeID IS NULL THEN
14.     SELECT CONCAT('From node name ', pFromNodeName, ' not found.' );
15. ELSE
16.     BEGIN
17.         -- start at src node
18.         SET vNodeID = vFromNodeID;
19.         SET vToNodeID = ( SELECT NodeID FROM dijnodes_money WHERE NodeName = pToNodeName
20. );
21. IF vToNodeID IS NULL THEN
22.     SELECT CONCAT('From node name ', pToNodeName, ' not found.' );
23. ELSE
24.     BEGIN
25.         -- calculate path costs till all are done
26.         UPDATE dijnodes_money SET Cost=0 WHERE NodeID = vFromNodeID;
27.         WHILE vNodeID IS NOT NULL DO
28.             BEGIN
29.                 UPDATE
30.                     dijnodes_money AS src
31.                     JOIN dijpaths_money AS paths ON paths.FromNodeID = src.NodeID
32.                     JOIN dijnodes_money AS dest ON dest.NodeID = Paths.ToNodeID
33.                     SET dest.Cost = CASE
34.                         WHEN dest.Cost IS NULL THEN src.Cost + Paths.Cost
35.                         WHEN src.Cost + Paths.Cost < dest.Cost THEN src.Cost +
36.                             Paths.Cost
37.                         ELSE dest.Cost
38.                     END,
39.                     dest.PathID = Paths.PathID
40.                 WHERE
41.                     src.NodeID = vNodeID
42.                     AND (dest.Cost IS NULL OR src.Cost + Paths.Cost < dest.Cost)
43.                     AND dest.Calculated = 0;
44.                 UPDATE dijnodes_money SET Calculated = 1 WHERE NodeID = vNodeID;
45.                 SET vNodeID = ( SELECT nodeID FROM dijnodes_money
46.                     WHERE Calculated = 0 AND Cost IS NOT NULL
47.                     ORDER BY Cost LIMIT 1
48.                 );
49.             END;
50.         END WHILE;
51.     END;
52. END IF;
53. IF EXISTS( SELECT 1 FROM dijnodes_money WHERE NodeID = vToNodeID AND Cost IS NULL ) T
54. HEN
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;
77.
78.             INSERT INTO Map(FromNodeName,ToNodeName,Cost) VALUES(vFromNodeName,vToNodeName,
79.                 vCost);
80.             SET vToNodeID = (SELECT FromNodeID FROM dijpaths_money WHERE PathID = vPathID
81.                 );
82.             END;
83.         END WHILE;
84.         SELECT FromNodeName,ToNodeName,Cost FROM Map ORDER BY RowID DESC;
85.         DROP TEMPORARY TABLE Map;
86.     END IF;
87. END

```

10 APÊNDICE B

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 IGUAÇU	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 IGUAU	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