

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

GUILHERME LINCON BARCZYSZYN

**SISTEMA COLABORATIVO PARA PLANEJAMENTO DE
ROTAS PARA CADEIRANTES**

DISSERTAÇÃO

CURITIBA

2018

Dados Internacionais de Catalogação na Publicação

Barczykzyn, Guilherme Lincon

Sistema colaborativo para planejamento de rotas para cadeirantes[recurso eletrônico] / Guilherme Lincon Barczykzyn. – 2018.

1 arquivo texto (71 f.) : PDF ; 1.83 MB.

Modo de acesso: World Wide Web.

Texto em português com resumo em inglês.

Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Computação Aplicada. Área de concentração: Engenharia De Sistemas Computacionais, Curitiba, 2018.

Bibliografia: f. 67–70.

1. Computação - Dissertações. 2. Paraplégicos. 3. Deficientes. 4. Computadores e deficientes. 5. Arquitetura e deficientes físicos. 6. Deficientes - Orientação e mobilidade. 7. Projeto de acessibilidade 8. Acessibilidade - Simulação por computador –Dissertações. I. Kozievitch, Nádia Puchalski. II. Minetto, Rodrigo. III. Universidade Tecnológica Federal do Paraná- Programa de Pós-Graduação em Computação Aplicada. IV. Título.

CDD: Ed. 23 – 621.39

Biblioteca Central do Campus Curitiba – UTFPR

Bibliotecário: Adriano Lopes CRB-9/1429

GUILHERME LINCON BARCZYSZYN

**SISTEMA COLABORATIVO PARA PLANEJAMENTO DE
ROTAS PARA CADEIRANTES**

Dissertação apresentada ao Programa de Pós-Graduação em Computação Aplicada da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de “Mestre em Computação Aplicada” – Área de Concentração: Engenharia De Sistemas Computacionais.

Orientador: Prof^ª. Dr^ª. Nádia Puchalski
Kozievitch

Coorientador: Prof. Dr. Rodrigo Minetto

CURITIBA

2018

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Câmpus Curitiba

Programa de Pós-Graduação em Computação Aplicada

Título da Dissertação Nº 70

**SISTEMA COLABORATIVO PARA PLANEJAMENTO DE ROTAS
PARA CADEIRANTES**

por

GUILHERME LINCON BARCZYSZYN

Orientador: Prof^ª. Dr^ª. Nádia Puchalski Kozievitch

Coorientador: Prof. Dr. Rodrigo Minetto

Esta Dissertação foi apresentada como requisito à obtenção do grau de MESTRE EM COMPUTAÇÃO APLICADA - Área de concentração: Engenharia De Sistemas Computacionais pelo Programa de Pós-Graduação em Computação Aplicada- PPGCA – da Universidade Tecnológica Federal do Paraná- UTFPR, às 16 horas do dia 13 de Dezembro de 2018. O trabalho foi aprovado pela Banca Examinadora, composta pelos Doutores:

Prof. Dr. Artur Ziviani
(LNCC)

Prof^ª. Dr^ª. Rita Berardi
(UTFPR)

Prof^ª. Dr^ª.Nádia Puchalski Kozievitch
(UTFPR)

Visto da coordenação:

Prof. Adolfo Gustavo Serra Seca Neto, Dr.
(Coordenador do PPGCA)

A Folha de Aprovação assinada encontra-se na Coordenação do Programa de Pós-Graduação em Computação Aplicada

Dedico este trabalho à todos a
pesquisadores que superaram os desafios
que lhes foram impostos de maneira a se
tornarem profissionais/pessoas melhores.

Agradecimentos

Agradeço ao projeto BIGSEA, à FUNTEF, ao PPGCA e à UTFPR pelo apoio a esta pesquisa.

“Ao vencido, ódio ou compaixão; ao vencedor, as batatas”.

(Machado de Assis)

Resumo

BARCZYSZYN, Guilherme. **SISTEMA COLABORATIVO PARA PLANEJAMENTO DE ROTAS PARA CADEIRANTES**. 2018. 71 f.

Dissertação – Programa de Pós-Graduação em Computação Aplicada – Universidade Tecnológica Federal do Paraná. Curitiba, 2018.

Planejamento de rotas é um desafio para a computação urbana que, de maneira geral, envolve o processamento de uma grande quantidade de dados, além do *feedback* dos usuários. Tradicionalmente, os serviços de planejamento de rota são baseados em ruas, isto é, mesmo um caminho para um pedestre é fornecido utilizando ruas. Entretanto, tais modelos não são adequados para usuários com certas deficiências. A fim de resolver tal problema, essa dissertação apresenta um modelo baseado em calçadas visando suprir as necessidades de um cadeirante por meio de um serviço de planejamento de rotas. O modelo é matematicamente definido como um grafo, em que os vértices são as esquinas e as arestas são as calçadas ou cruzamentos. Os custos de cada aresta são obtidos de diversos fatores como distância, inclinação e a condição/manutenção das rampas, cruzamentos e calçadas. O modelo foi pensado de maneira que o *feedback* do usuário fosse considerado durante a atualização de suas informações com dados de cadeirantes sobre problemas de acessibilidade e de problemas resolvidos pelo departamento de planejamento urbano da cidade. É apresentado ainda um algoritmo de planejamento de rotas que fornece um conjunto de rotas alternativas baseadas em condições de acessibilidade e um algoritmo de recomendação de atalhos para auxiliar a tomada de decisões pelo departamento de planejamento urbano da cidade. Além disso, foi desenvolvida uma API para o fornecimento dos serviços de caminho mínimo em calçadas. Experimentos, usando *pgRouting* e o *Post-GIS* com dados abertos foram realizados em um bairro de uma cidade brasileira a fim de validar o modelo e o serviço de planejamento de rotas.

Palavras-chave: Cadeirante. Planejamento de rotas. Acessibilidade.

Abstract

BARCZYSZYN, Guilherme. **COLLABORATIVE SYSTEM FOR WHEELCHAIR ROUTE PLAN**. 2018. 71 p. Dissertation – Graduate Program in Applied Computing – Federal University of Technology – Paraná. Curitiba, 2018.

Route planning is a challenging problem for urban computing that usually involves the processing of a huge amount of data and collaborative user feedback. Traditionally, route planning services are street-based, that is, even paths for a pedestrian are suggested in terms of streets. However such models are not suitable for users with certain disabilities. To address this problem, we describe in this dissertation a sidewalk-based model to accommodate the needs for a wheelchair route planning service. The model is mathematically defined as a graph, where the vertices are the city block corners and the edges are the sidewalks or crosswalks. The edge costs are derived from important accessibility features, such as distance, path inclination, and existence and maintenance conditions of curb ramps, crosswalks and sidewalks. The model has been organized such as the user feedback is considered to help updating the model when accessibility issues are detected, by wheelchair users and companions, or solved, by the department of city planning. We also present a route planning algorithm that provides a set of alternative routes based on accessibility conditions, and a shortcut recommender algorithm to support accessibility-related decision making by the department of city planning. Besides, it was developed an API to provide shortest-path using sidewalks. Experiments, by using PgRouting and PostGIS with open data, are reported for a brazilian city neighborhood in order to validate the model and the route planning service.

Keywords: Wheelchair. Route planning. Accessibility.

Lista de Ilustrações

| | |
|---|----|
| Figura 1 – Modelo proposto baseado em grafo. | 15 |
| Figura 2 – Uma rota a partir de um modelo baseado em ruas (Google Maps) (a) e uma rota a partir de um modelo baseado em calçadas (b). | 16 |
| Figura 3 – Esquema de desenvolvimento do trabalho para atingir os objetivos propostos. | 19 |
| Figura 4 – Camadas da arquitetura de um GIS. | 21 |
| Figura 5 – Operações Topográficas (a) “disjunto”, “toca” e “sobrepõe” (b) “contém” e “cruza”. Fonte: Câmara et al.(1996). | 24 |
| Figura 6 – Exemplo de altimetria da cidade de Curitiba. | 25 |
| Figura 7 – Modelo de CI da IBM. | 28 |
| Figura 8 – Arquitetura baseada no modelo para computação urbana. | 38 |
| Figura 9 – Domínios de dados geográficos utilizados na implementação. | 39 |
| Figura 10 – Quadras da cidade e sua representação pelo SIG. | 40 |
| Figura 11 – Geometrias simplificadas e representação dos cruzamentos | 41 |
| Figura 12 – Grau de inclinação do caminho. | 42 |
| Figura 13 – Rotas mínimas de Yen. | 45 |
| Figura 14 – Extração dos contornos das quadras. | 48 |
| Figura 15 – Pontos que identificam cada cruzamento da cidade. | 49 |
| Figura 16 – Pontos das esquinas identificados. | 49 |
| Figura 17 – Raio agrupando pontos de esquinas de um cruzamento. | 50 |
| Figura 18 – Travessias dos cruzamentos. | 50 |
| Figura 19 – Fluxo de requisições para a API. | 53 |
| Figura 20 – Resultado da execução do algoritmo de Dijkstra na implementação do modelo baseado em calçadas. | 55 |
| Figura 21 – Exemplo de transformação de dados de altimetria em custo de inclinação. | 55 |
| Figura 22 – Máximo esforço nas ruas do bairro Batel. | 56 |
| Figura 23 – Local de maior inclinação no Batel (A). | 57 |
| Figura 24 – Local de maior inclinação no Batel (B). | 57 |
| Figura 25 – Infraestruturas de acessibilidade do Batel. | 58 |
| Figura 26 – Problemas encontrados nas calçadas. | 59 |
| Figura 27 – Caminhos mínimos. | 60 |
| Figura 28 – Resultado da execução do algoritmo de Dijkstra na implementação do modelo baseado em calçadas considerando toda a cidade de Curitiba. | 61 |
| Figura 29 – Testes comparativos com o <i>Google Maps</i> distância entre dois hospitais: do Hospital Ônix ao Hospital Geral de Curitiba. | 63 |
| Figura 30 – Testes comparativos com o <i>Google Maps</i> distância entre dois pontos: da Alameda Presidente Taunay (próximo ao Hospital de Olhos) ao Instituto Paranaense de Cegos. | 64 |
| Figura 31 – Testes comparativos com o <i>Google Maps</i> distância entre dois pontos: do Hospital Geral de Curitiba ao Shopping Crystal. | 65 |

Lista de Abreviaturas, Siglas e Acrônimos

| | |
|--------|---|
| SQL | <i>Structured Query Language</i> |
| IPPUC | Instituto de Pesquisa e Planejamento Urbano de Curitiba |
| SEDPCD | Secretaria Especial da Pessoa com Deficiência |
| IEEE | <i>Institute of Electrical and Electronics Engineers</i> |
| ITSC | <i>International Conference on Intelligent Transportation Systems</i> |
| ADA | <i>American Disability Association</i> |
| GPS | <i>Global Positioning System</i> |
| SIG | Sistema de Informações Geográficas |
| OSM | <i>Open Street Maps</i> |
| POI | <i>Points of Interest</i> |
| SGBD | Sistema Gerenciador de Banco de Dados |
| QGIS | <i>QuantumGIS</i> |
| JSON | <i>JavaScript Object Notation</i> |
| API | <i>Application Programming Interface</i> |
| IHC | Interação Humano-Computador |

Lista de Símbolos

Θ Complexidade algorítmica de tempo

Sumário

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 14 |
| 1.1 | OBJETIVO GERAL | 17 |
| 1.2 | OBJETIVOS ESPECÍFICOS | 17 |
| 1.3 | METODOLOGIA | 18 |
| 1.4 | PUBLICAÇÕES | 18 |
| 1.5 | ESTRUTURA DA DISSERTAÇÃO | 19 |
| 2 | CONCEITOS BÁSICOS E TRABALHOS RELACIONADOS | 20 |
| 2.1 | SIG | 20 |
| 2.1.1 | Geoprocessamento | 22 |
| 2.1.2 | Sistemas de Coordenadas e Georreferenciamento | 22 |
| 2.1.3 | Relações Espaciais | 23 |
| 2.1.4 | Topografia | 24 |
| 2.2 | ALGORITMOS | 25 |
| 2.3 | CIDADES INTELIGENTES | 27 |
| 2.3.1 | Definição | 27 |
| 2.3.2 | SIG para Cidades Inteligentes | 28 |
| 2.4 | TRABALHOS RELACIONADOS | 29 |
| 2.4.1 | Planejamento de Rotas | 29 |
| 2.4.2 | Acessibilidade | 33 |
| 2.4.3 | Proposta | 34 |
| 3 | MODELO TEÓRICO | 36 |
| 3.1 | AQUISIÇÃO DE DADOS | 36 |
| 3.2 | ARQUITETURA | 37 |
| 3.3 | GERENCIAMENTO DE DADOS URBANOS | 38 |
| 3.3.1 | Construção do Grafo | 39 |
| 3.3.2 | Inclinação | 41 |
| 3.3.3 | Coefficiente de Acessibilidade | 43 |
| 3.4 | FORNECIMENTO DE SERVIÇO | 43 |
| 3.4.1 | Planejamento de rotas com acessibilidade | 43 |
| 4 | IMPLEMENTAÇÃO E TESTES | 46 |
| 4.1 | TECNOLOGIAS UTILIZADAS | 46 |
| 4.2 | IMPLEMENTAÇÃO DO GRAFO | 46 |
| 4.2.1 | Extração de calçadas | 47 |
| 4.2.2 | Obtenção dos cruzamentos | 47 |

| | | |
|----------|---|-----------|
| 4.2.3 | Grafo baseado em calçadas | 50 |
| 4.2.4 | API | 51 |
| 4.3 | TESTES | 52 |
| 4.3.1 | Menor caminho em calçadas | 53 |
| 4.3.2 | Inclinação | 54 |
| 4.3.3 | Coefficiente de Acessibilidade | 56 |
| 4.3.4 | Planejamento de rotas com acessibilidade | 59 |
| 4.3.5 | Caminho mínimo em calçadas para toda Curitiba | 60 |
| 4.3.6 | Comparação com <i>GoogleMaps</i> | 61 |
| 5 | CONCLUSÃO | 66 |
| | REFERÊNCIAS | 68 |

CAPÍTULO 1

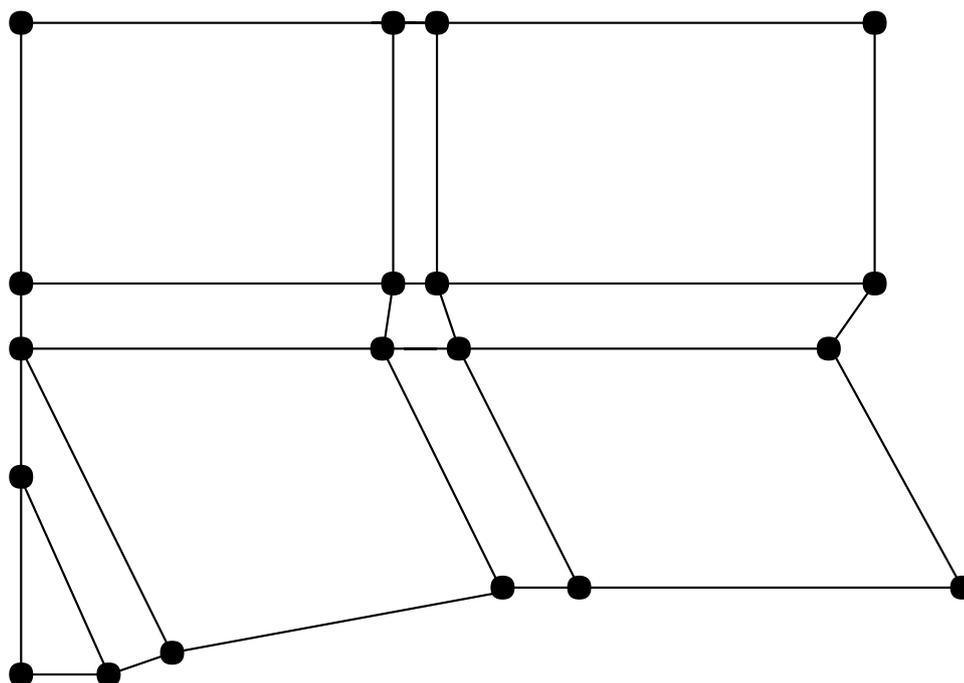
INTRODUÇÃO

Diversos esforços têm sido feitos para desenvolver serviços de planejamento de rotas destinados a pessoas com algum tipo de deficiência [(SOBEK; MILLER, 2006), (MENKENS et al., 2011), (SUMIDA et al., 2012), (WATTANAVARANGKUL; WAKAHARA, 2013), (NEIS, 2015)]. Essa questão é um problema desafiador que envolve o processamento de uma grande quantidade de dados - como mapas, imagens e topografia da cidade - além de uma grande quantidade de informações complexas oriundas de *feedbacks* de usuários e de agências municipais, a fim de definir corretamente caminhos confiáveis. Populares planejadores de rotas, como o Google Maps¹ são inadequados para tal propósito pois mesmo caminhos para pedestres são sugeridos em termos de ruas, conforme mostra a Figura 2a. Isso é um problema crítico para cadeirantes, pois calçadas de uma mesma rua podem apresentar diferentes condições de acessibilidade, além disso, não se tem informação de quão custoso é seguir uma rota na qual problemas de acessibilidade não são considerados durante o planejamento - note que uma rampa ausente no cruzamento de um pedestre pode mudar completamente o caminho traçado.

Serviços de rotas populares como o *Google Maps* são baseados em ruas e desconsideram caminhos baseados em calçadas. A fim de resolver tal questão, foi desenvolvido aqui um sistema colaborativo para planejamento de rotas para cadeirantes que usa um modelo baseado em calçadas, conforme mostra a Figura 2b. Especificamente, foram utilizadas rotas em termos de calçadas e cruzamentos, refletindo as condições que um cadeirante encontraria ao deslocar-se. Este modelo é matematicamente definido com um grafo, no qual os nós são as esquinas que formam as quadras da cidade e as arestas são as calçadas ou os cruzamentos conforme ilustra a Figura 1.

¹ Google Maps. <https://www.google.com/maps/>. Último acesso em 21/05/2018.

Figura 1 – Modelo proposto baseado em grafo.



Inicialmente, foi construído um grafo simplificado utilizando um conjunto de camadas de dados (topografia e outras) fornecidos pelo Instituto de Pesquisa e Planejamento Urbano de Curitiba (IPPUC²). Os custos das arestas foram obtidos a partir de fatores de acessibilidade como distâncias, inclinação, existência ou condições de manutenção de rampas e calçadas. Foi utilizado um Sistema de Informação Geográfica (SIG) para armazenar e manipular os dados via consultas SQL e uma biblioteca geoespacial com implementações de algoritmos de menor caminho.

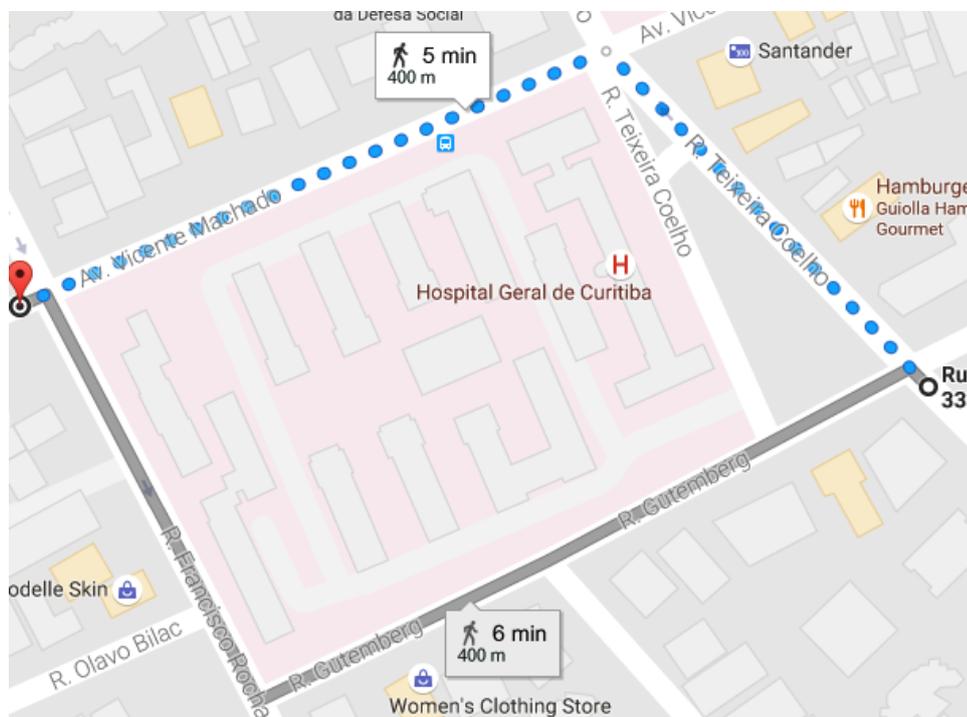
O *feedback* do usuário é também considerado para refinar os custos das arestas conforme a infraestrutura básica para cadeirantes. O grafo é, então, atualizado atribuindo valores “infinito” em arestas que representam cruzamentos que não possuem rampas (o resultado é o mesmo da remoção dessa aresta) ou em arestas que representam calçadas em péssimas condições de manutenção. Os valores também podem ser atualizados de maneira oposta quando a condição de manutenção é melhorada.

O modelo também fornece informações importantes para o departamento de planejamento urbano da cidade. Nos experimentos realizados, nota-se que o modelo pode tornar-se facilmente desconectado em algumas regiões e minimamente conectado em outras (a remoção de uma aresta pode desconectar a região). Isso ocorre pois diversas rampas

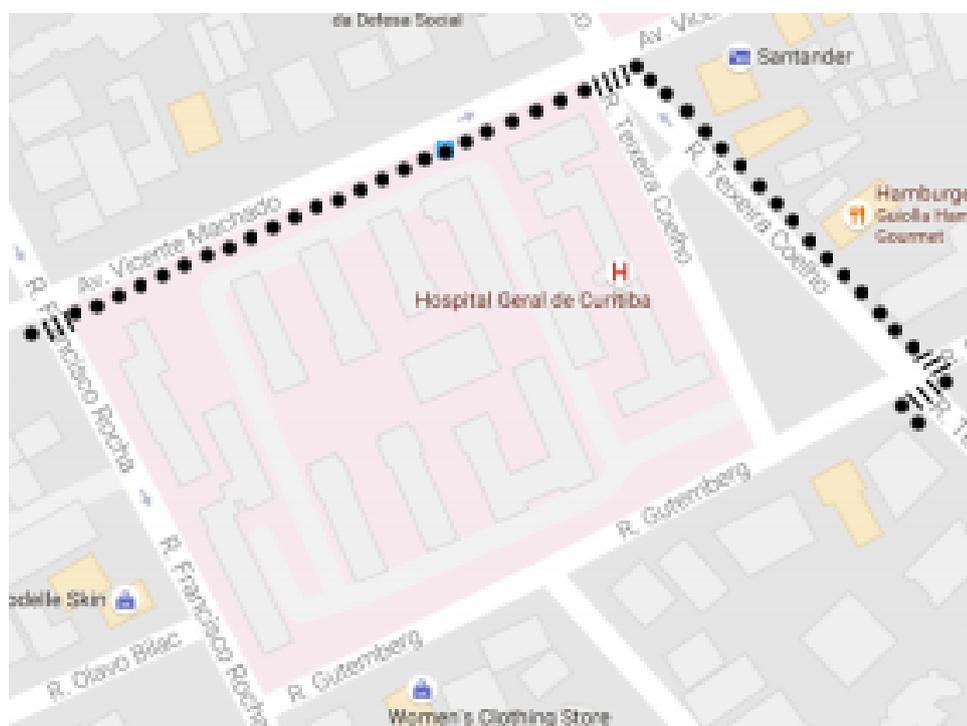
² IPPUC - <http://www.ippuc.org.br/>. Último acesso em 22/05/2018

Figura 2 – Uma rota a partir de um modelo baseado em ruas (Google Maps) (a) e uma rota a partir de um modelo baseado em calçadas (b).

(a) Modelo baseado em ruas



(b) Modelo baseado em calçadas



não existem e o custo das arestas é definido como infinito (removida). Em outras palavras, uma rota entre dois pontos separados por poucos metros pode representar quilômetros de distância ou mesmo ser impossível se o trajeto considerado depender da existência de

rebaixamentos para ser percorrido.

Neste trabalho, aborda-se também um problema de conectividade existente em muitas cidades, causado pela ausência de rampas e calçadas, e que pode ser melhorado adicionando um pequeno subconjunto de k rampas, estrategicamente localizadas. Este problema é conhecido em teoria de grafos como o problema da seleção de arestas de atalho [(MEYERSON; TAGIKU, 2009), (PAPAGELIS; BONCHI; GIONIS, 2011), (PAROTSIDIS; PINTOURA; TSAPARAS, 2015)] e possui aplicações em física, comunicação e redes sociais. No entanto, uma solução ótima para este problema de conectividade pertence a classe dos problemas NP-completos (CORMEN; STEIN; RIVEST, 2001), isto é, o algoritmo ideal possui complexidade de tempo exponencial. Por outro lado, soluções não ideais com tempo polinomial (MINETTO et al., 2016), como um algoritmo de recomendação de atalhos mostrados na Seção 4.3.4 são essenciais para melhorar a utilização de recursos e melhorar a mobilidade do cadeirante.

O sistema aqui descrito traz as seguintes contribuições para a área:

1. O desenvolvimento de um modelo baseado em calçada para planejamento de rotas para cadeirantes;
2. A inclusão de importantes fatores em rotas para cadeirantes, como inclinação do terreno, e existência e manutenção de rampas e calçadas;

1.1 OBJETIVO GERAL

O objetivo geral desta dissertação é a elaboração de um modelo teórico baseado em calçadas para fornecimento de rotas para cadeirantes, que considere aspectos importantes como a inclinação do terreno e a acessibilidade. Além disso, pretende-se, realizar a implementação deste modelo utilizando dados geoespaciais. Por fim, a partir da implementação, deseja-se construir um serviço para fornecimento de melhores rotas para cadeirantes.

1.2 OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral são considerados os seguintes objetivos específicos:

1. Explorar questões bibliográficas sobre (i) SIG, (ii) algoritmos para caminhos míni-

mos, e (iii) cidades inteligentes.

2. Explorar os trabalhos relacionados buscando modelos existentes e sistemas para planejamento de rotas para pessoas com necessidades especiais.
3. Seguindo a arquitetura proposta por (ZHENG et al., 2014), elaborar um modelo teórico baseado em calçadas para planejamento de rotas para cadeirantes.
4. Implementar o modelo elaborado utilizando dados geoespaciais abertos da cidade de Curitiba.
5. Fornecer um serviço para planejamento de rotas para cadeirantes.
6. Validar o a implementação do modelo desenvolvido, executando uma série de testes focando focando nos objetivos definidos para um deslocamento acessível.

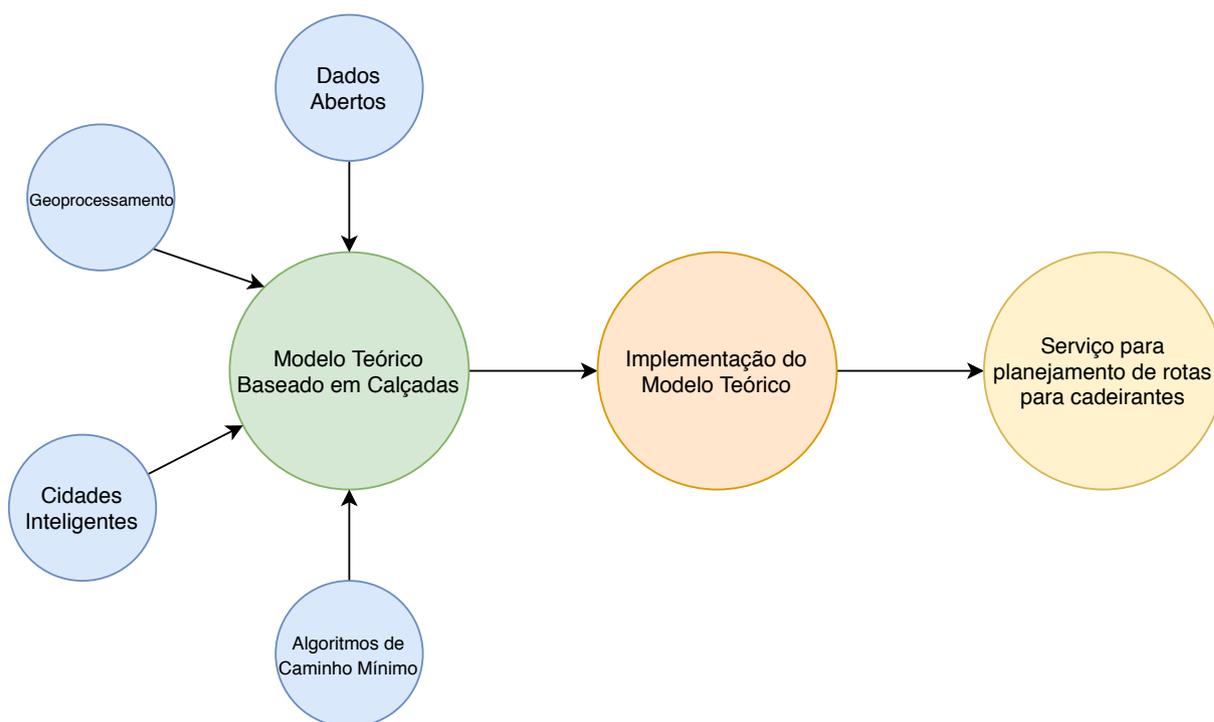
1.3 METODOLOGIA

A Figura 3 ilustra a metodologia seguida por este trabalho: os círculos em azul no diagrama mostram os conceitos que são necessários para desenvolver o modelo teórico. A partir dele é possível realizar uma implementação e disponibilizar um serviço para o planejamento de rotas para cadeirantes conforme mostram os círculos em rosa.

- Pesquisa bibliográfica sobre conceitos que embasam o desenvolvimento do Modelo Teórico;
- Desenvolvimento de Modelo Teórico baseado em Calçadas;
- Implementação do Modelo Teórico;
- Elaboração de Serviço para planejamento de rotas para cadeirantes;
- Validação do modelo e do serviço desenvolvidos.

1.4 PUBLICAÇÕES

As publicações diretamente relacionadas a esta dissertação (em ordem cronológica), são:

Figura 3 – Esquema de desenvolvimento do trabalho para atingir os objetivos propostos.

1. Guilherme L. Barczyszyn, Nádia P. Kozievitch, Rodrigo Minetto, Ricardo Dutra da Silva e Juliana de Santi: Utilização de dados de altimetria para o fornecimento de rotas acessíveis para cadeirantes. *GEOINFO* 2017: 104-109.
2. Guilherme L. Barczyszyn, Letícia M. O. Carmenar, Diego F. Nascimento, Nádia P. Kozievitch, Ricardo Dutra da Silva, Leonelo D. A. Almeida, Juliana de Santi e Rodrigo Minetto. A collaborative system for suitable wheelchair route planning. *ACM Transactions on Accessible Computing (TACCESS)*, v. 11, p. 1-26, 2018.

1.5 ESTRUTURA DA DISSERTAÇÃO

Este documento está organizado da seguinte maneira: o capítulo 2 apresenta conceitos importantes para o entendimento deste trabalho e apresenta trabalhos relacionados sobre modelos para planejamento de rotas acessíveis. O capítulo 3 apresenta a elaboração do modelo teórico proposto. O capítulo 4 demonstra como foi realizada a implementação deste modelo e a sua validação através de testes. Finalmente, o capítulo 5 apresenta as considerações finais e futuros trabalhos.

CAPÍTULO 2

CONCEITOS BÁSICOS E TRABALHOS RELACIONADOS

Neste capítulo, alguns conceitos que dão base a este trabalho são apresentados como aqueles referentes a Sistemas de Informações Geográficas, Algoritmos de cálculo de caminho mínimo e Cidades Inteligentes. Além disso, são apresentados os trabalhos relacionados.

2.1 SIG

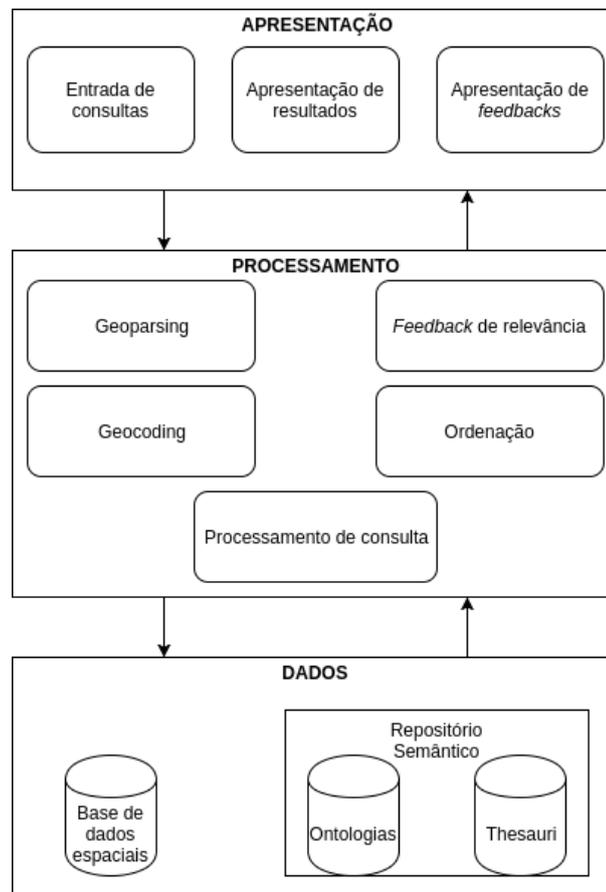
Sistemas de Informação Geográfica (SIG) são responsáveis pela manipulação de dados geográficos. Os SIG em geral são construídos seguindo uma arquitetura específica, conforme pode-se verificar na Figura 4, cujas camadas podem ser divididas da seguinte maneira:

- **Camada de apresentação:** Esta camada é responsável pela entrada da consulta do usuário, pela apresentação dos resultados das consultas e pela avaliação dos resultados pelo usuário. Nesta camada, existem preocupações que tangem a área de IHC (Interação Humano-Computador).
- **Camada de processamento:** Nesta camada ocorrem os processos de *geoparsing* e *geocoding*. O processo de *geoparsing*, conforme explicam (LI; TORRES, 2014), consiste na transformação de entidades geográficas para objetos digitais, além de sua desambiguação. O *geocoding*, para (LI; TORRES, 2014), consiste na associação de coordenadas geográficas a um objeto digital. Na camada de processamento encontra-se o processamento de consultas, que é responsável por interpretar e processar as consultas que o usuário utilizou como entrada. Encontra-se a avaliação dos resultados que tem o objetivo de melhorar os resultados baseados no feedback do usuário. A

camada de processamento também agrega a ordenação de resultados, responsável por ordenar os resultados baseados em sua relevância.

- **Camada de dados:** É dividida em base de dados espacial e semântica. A primeira consiste no armazenamento de objetos geográficos e fornece operações espaciais para consultas sobre tais objetos. Os objetos são armazenados utilizando-se pontos, linhas ou polígonos em algum sistema de coordenadas. A base de dados semântica armazena nome de lugares e definem de que forma elas estão organizadas e relacionadas umas às outras. Esses dados dão suporte para as técnicas de *geocoding* e *geoparsing* na camada de processamento.

Figura 4 – Camadas da arquitetura de um GIS.



Fonte: Adaptado de (LI; TORRES, 2014)

2.1.1 Geoprocessamento

O geoprocessamento caracteriza-se por ser uma área de conhecimento relativamente nova e ainda pouco explorada na resolução de problemas urbanos. Segundo (MOURA, 2012), o geoprocessamento surgiu no final do século XX como uma ferramenta de alta tecnologia para interpretações relacionadas a um espaço.

Pode-se definir o geoprocessamento como uma área que envolve a utilização de um conjunto de técnicas que permitem fazer análises espaciais, manipular e gerenciar informações georreferenciadas com uma agilidade e precisão que até antes de seu surgimento eram inimagináveis [(MOURA, 2012)].

A informação geográfica é caracterizada sempre pela existência de um atributo nas entidades que a relaciona com uma localização geográfica existente ou com uma outra entidade geográfica cuja localização geográfica é conhecida. Essa informação pode ser representada por dados em coordenadas geográficas, endereços completos ou referências únicas como o nome de uma universidade (Universidade Tecnológica Federal do Paraná) pode indicar o estado onde ela se encontra [(LI; TORRES, 2014)].

É importante conhecer o conceito de geoprocessamento pois ele fornecerá as principais ferramentas para manipular dados geográficos que auxiliarão o desenvolvimento desse trabalho. É graças aos geoprocessamento de dados fornecidos pelo departamento urbano da cidade que é possível elaborar e desenvolver o modelo aqui proposto.

2.1.2 Sistemas de Coordenadas e Georreferenciamento

O sistema de coordenadas é o sistema mais simples, popular e antigo de se utilizar até hoje. Nele, qualquer entidade geográfica está associada a um ponto de coordenadas (x e y). No globo terrestre, tem-se os meridianos, linhas paralelas que ligam os pólos Norte e Sul da Terra, e estão associadas, cada uma delas, a um valor de ângulo diferente de inclinação em relação a eles que pode ser chamado de longitude. Os paralelos, por sua vez são linhas que circundam a Terra horizontalmente. Cada um deles também associados a um ângulo de inclinação em relação ao centro da Terra, sendo o valor desse ângulo chamado de latitude (LI; TORRES, 2014).

No geoprocessamento existe um processo fundamental denominado georreferenciamento. Segundo o (IPPUC, 2013), georreferenciamento consiste em definir a localização

geográfica e, conseqüentemente, as coordenadas dos elementos passíveis de espacialização. Em suma, a partir de um ente geográfico, define-se a latitude e longitude deste ente. Isso pode ocorrer a partir de fontes que indiquem ou não as coordenadas deste ente.

O georreferenciamento é a ferramenta mais básica e mais essencial para o desenvolvimento deste trabalho. Trabalhar com objetos geográficos requer a identificação de sua localização, o que é garantido pelo georreferenciamento. Por exemplo, cada ponto existente em um modelo de fornecimento de rotas deve ser georreferenciado por suas coordenadas geográficas.

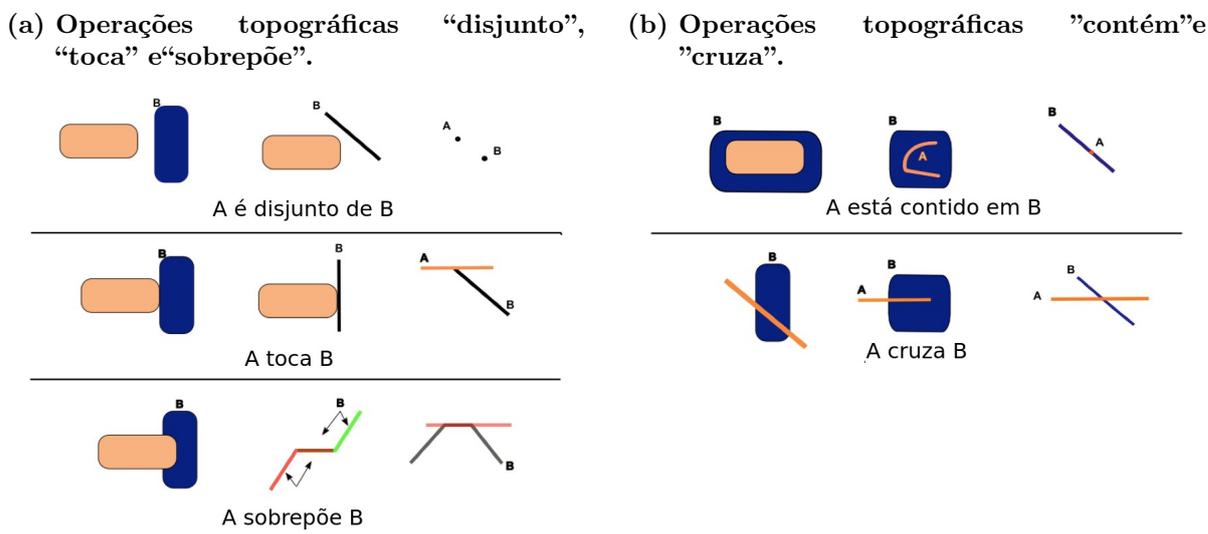
2.1.3 Relações Espaciais

Na informação geográfica, pode-se ter relações entre entidades geográficas expressadas de maneiras diferentes. (LI; TORRES, 2014) apresentam os conceitos dessas relações, retiradas da geometria clássica da seguinte maneira:

- **Relações Topográficas:** Nas relações topográficas, indica-se conexões entre objetivos do tipo “é adjacente”, “contém”, “é contido” mas não inclui medidas ou direção. Ainda pode-se ter “disjuntos”, “dentro”, “sobrepõe”, “cruza”, “toca”, “igual-se”, “assemelha-se” [(EGENHOFER, 1997), (CLEMENTINI; FELICE; OOSTEROM, 1993)]. Na Figura 5b, verifica-se uma representação gráfica dessas relações topológicas. A primeira delas é a disjunção (*disjoint*) entre os objetos A e B, nota-se que a disjunção ocorre não importando a forma em que os objetos assumem (uma entidade, uma linha ou pontos). Há também a relação toca (*touch*) em que os objetos podem assumir entidades ou linhas e se tocarem, sem se sobrepor. Outra relação representada é a de sobreposição (*overlap*) em que o objeto A se põe acima do objeto B, assumindo a forma de entidade ou de linhas. Há ainda, conforme ilustra a Figura 5a, as relações “está contido” (*in*) onde os objetos, linhas ou entidades, podem estar dentro (contidos) em outros e a relação de cruzamento (*cross*) onde um objeto pode cruzar o outro.
- **Relações Métricas:** Nas relações métricas, são expressadas medidas quantitativas como área, distância, comprimento e perímetro.
- **Relações Direcionais:** As relações direcionais expressam orientações baseadas nos pontos cardinais como Norte, Sul, Leste, Oeste ou ainda posições de ordem como “à

frente de”, “atrás de”, “sobre”, “sob”.

Figura 5 – Operações Topográficas (a) “disjunto”, “toca” e “sobrepõe” (b) “contém” e “cruza”.
Fonte: Câmara et al.(1996).



Conhecer a relações espaciais é indispensável para o desenvolvimento desse trabalho. É necessário que se torne informações contendo diversos objetos geográficos em informação útil para o desenvolvimento do modelo. Por exemplo, se é necessário encontrar o ponto central de um cruzamento, utiliza-se a operação “cruza” entre duas ruas. Se for necessário encontrar todas as esquinas dentro de um raio em torno desse ponto central, utiliza-se a operação contém para agrupá-los.

2.1.4 Topografia

A Topografia (do grego: *topos*, lugar e *graphein*, descrever) é a ciência que trata dos princípios e métodos empregados para determinar as posições relativas de pontos da superfície terrestre por meio de medidas [(MÁRQUEZ, 2003)].

Dentro da Topografia, existem três operações que se aplicam na representação de um terreno, (MÁRQUEZ, 2003) define-as da seguinte maneira:

- **Planimetria** que estuda os instrumentos e métodos para projetar sobre uma superfície plana horizontal a exata posição dos pontos mais importantes do terreno e construir dessa maneira uma figura similar ao mesmo.
- **Altimetria** que determina as alturas dos diferentes pontos do terreno a respeito de uma superfície de referência; geralmente correspondente ao nível médio do mar.

- **Agrimensura** que compreende os procedimentos empregados para medir a superfície dos terrenos e para fracioná-los.

A Figura 6 foi gerada com dados de altimetria da cidade de Curitiba. Tal figura exemplifica a aplicação da operação de altimetria. Observe que existem diversas linhas e sobre cada uma delas há um valor. As linhas indicam a altura daquela região em relação ao nível do mar – esta medida é expressada, em metros, pelo número sobre a curva.

Figura 6 – Exemplo de altimetria da cidade de Curitiba.



Neste trabalho, a altimetria é utilizada para identificar declives em uma rota. É importante que se conheça o básico sobre tal conceito para possibilitar a extração de fatores importantes para um cadeirante que o envolva.

2.2 ALGORITMOS

Encontrar o menor caminho entre dois pontos é um problema fundamental dentro da Ciência da Computação e suas soluções fornecem respostas interessantes para outros problemas também (Frigioni et al., 1998). Conforme visto em seções anteriores, o geoprocessamento trabalha com pontos que representam uma coordenada geográfica na superfície terrestre. Utilizando tais pontos, é possível trabalhar com algoritmos de menor distância. Existem diversos algoritmos de busca de menor caminho, entre eles, destacam-se:

- **Algoritmo de Dijkstra (DIJKSTRA, 1959):** O Algoritmo de Dijkstra propõe a resolução do problema do caminho mínimo construindo uma árvore de menor

caminho entre os n nós de um grafo. Cada par de nós pode estar conectado por uma aresta com os pesos.

Para a construção da árvore de caminho mínimo, consideram-se os seguintes conjuntos (DIJKSTRA, 1959):

- **Conjunto I.** Arestas que fazem parte da árvore a ser construída, formando uma sub-árvore;
- **Conjunto II.** Arestas candidatas a serem a próxima a ser passada para o Conjunto I;
- **Conjunto III.** Arestas restantes (rejeitadas ou não consideradas);
- **Conjunto A.** Nós conectados às arestas do Conjunto I;
- **Conjunto B.** Nós restantes.

A construção da árvore inicia-se escolhendo um nó inicial e adicionando-o ao Conjunto A e suas arestas adjacentes ao Conjunto II. Inicialmente, o Conjunto I está vazio. Deve-se, então, executar os seguintes passos repetidamente:

- **Passo 1.** A aresta de menor custo/distância do Conjunto II é removida e adicionada ao Conjunto I. Como resultado, um nó é passado do Conjunto B para o Conjunto A;
- **Passo 2.** Consideram-se as arestas adjacentes ao nó transferido para o Conjunto A que se unem aos nós restantes no Conjunto B. Se a aresta em questão é mais custosa que a correspondente no Conjunto II, ela é rejeitada; caso contrário, ela substitui a aresta correspondente no Conjunto II.

Retorna-se para o Passo 1 até que o Conjunto II e B estejam vazios. As arestas no Conjunto I formam a árvore de menor caminho do grafo.

- **A* (RUSSEL; NORVIG, 1994):** O A* é um algoritmo de busca de caminho entre dois nós em um grafo. Este algoritmo utiliza uma estimativa heurística ao selecionar um nó para a rota menos custosa. Ao final, o A* sempre garantirá uma

rota de menor custo. Supondo que o algoritmo esteja analisando um nó n em um grafo, é realizado, então, o seguinte cálculo:

$$f(n) = g(n) + h(n) \quad (1)$$

Onde:

- $g(n)$ - função que retorna o custo do nó inicial até o nó n ;
- $h(n)$ - função que retorna a heurística (caminho estimado) do nó n até o nó final;
- $f(n)$ - função que retorna o custo estimado total.

No algoritmo de Dijkstra busca-se todos os vértices do grafo para determinar a rota de menor custo entre cada ponto. Já o A^* é uma modificação do algoritmo de Dijkstra que utiliza uma heurística para determinar quais vértices buscar. A heurística é o custo estimado (ou distância) do vértice inicial ao vértice desejado. Quanto menor a heurística, mais provável que se pesquise nas arestas do vértice em questão do que em arestas de vértices mais distantes. Devido a isso, a rota entre o vértice inicial e final é encontrada de maneira mais rápida. O A^* permite que se encontre o melhor resultado por tempo de CPU.

Por utilizar um modelo baseado em grafo, é importante conhecer algoritmos de menor caminho que serão utilizados no trabalho para encontrar a melhor rota. O algoritmos A^* e Dijkstra são os mais populares em termos de menores rotas. Encontrar uma rota ideal para um cadeirante considerando um ou mais custos pode ser um problema resolvido pelo algoritmo de Dijkstra ou A^* . Logo, é importante que se conheça cada um deles para o desenvolvimento desse trabalho.

2.3 CIDADES INTELIGENTES

2.3.1 Definição

Segundo (ALAWADHI et al., 2012), uma cidade inteligente é aquela cidade que utiliza tecnologias para integrar suas infraestruturas e serviços para melhorar a eficiência, a eficácia, a transparência e a sustentabilidade. Para (CTECNO, 2012), é um modelo de

cidade no qual convergem diversas correntes de desenvolvimento urbano (cidade sustentável, cidade inovadora, cidade digital, cidade de conhecimento) para melhorar a qualidade de vida e a gestão de recursos e de serviços, mediante o uso intensivo de tecnologia. Já em 2008, a IBM¹ formulou a visão "Planeta mais Inteligente"², que representa a utilização da tecnologia como um pilar fundamental para a melhoria dos principais setores econômicos, administração pública e população. O modelo de cidade inteligente (CI) da IBM, considera três pilares que são: infraestrutura, as operações de planejamento e gestão e os cidadãos (IBM, 2015).

Figura 7 – Modelo de CI da IBM.



Fonte: (IBM, 2015)

2.3.2 SIG para Cidades Inteligentes

Uma cidade, por definição, é uma entidade geográfica e, portanto, o uso dos SIG como um dos “sistemas de registros” chave, auxilia diversas entidades da cidade, para que possam trabalhar em colaboração nas fases de planejamento, design, engenharia, construção, gerenciamento de ativos, operações e desenvolvimento, gestão de situações de

¹ IBM - http://www.ibm.com/smarterplanet/es/es/smarter_cities/overview/. Último acesso em 27/06/2018

² IBM - <http://www.ibm.com/smarterplanet/br/pt/>. Último acesso em 27/08/2018

interrupção e durante emergências (ESRI, 2014). Durante muito tempo, na era digital da cidade, os mapas digitais e banco de dados foram integrados aos fluxos de trabalho de gestão da terra, como por exemplo, planejamento urbano e transporte, porém, os avanços tecnológicos da área não só permitem que os SIG sejam ferramentas de armazenamento e gestão, como também a realização de modelos espaciais que auxiliem no processo de tomada de decisões (WANG, 2013).

Este trabalho utiliza a tecnologia a fim de melhorar a qualidade de vida para os cidadãos em um contexto urbano e auxiliar órgãos públicos a na administração da cidade. É possível, portanto, classificar o trabalho como parte de cidades inteligentes.

2.4 TRABALHOS RELACIONADOS

Nesta seção, são apresentados alguns trabalhos sobre planejamento de rotas (modelos, aplicações), acessibilidade e trabalhos que precedem o desenvolvimento do trabalho aqui proposto.

2.4.1 Planejamento de Rotas

Existem diversos trabalhos que buscam desenvolver um sistema de planejamento de rotas para cadeirantes. Nesta seção, serão apresentados alguns deles.

Um sistema *web* (MAGUS) que considera calçadas e obstáculos de maneira detalhada para fornecer rotas para cadeirantes é proposto em (BEALE et al., 2006). O modelo foi construído combinando mapas e fotos aéreas, que foram manualmente digitalizados e atualizados por meio de *feedbacks* dos usuários. Como consequência desses dados digitalizados, os autores apontam um pobre detalhamento das informações de declives neste modelo. Apesar desse sistema fornecer informações para a planejadores urbanos, ele parece não prover informações para melhorias das calçadas.

Um sistema *web* simplificado (*U-ACCESS*) para identificar rotas e auxiliar pedestres a desviar de obstáculos em um campus universitário é proposto em (SOBEK; MILLER, 2006). Os autores argumentam que uma aquisição de dados detalhada como em (BEALE et al., 2006) requer muito tempo dos usuários, além de ser custosa para implantar e manter. O sistema computa rotas de acordo com três diferentes níveis de habilidades físicas: mobilidade não assistida, mobilidade parcial assistida e mobilidade de cadeira de rodas.

Apesar deste modelo levar em conta calçadas, ele não deixar claro como é feita a distinção entre calçadas e ruas. Além disso, não há um *feedback* dinâmico para reportar problemas de mobilidade.

Também buscando por um modelo simplificado, os autores em (HOLONE; MISUND, 2007) apresentam um protótipo de aplicação móvel para pedestres (*OurWay*), baseado em dados do OpenStreetMap [(HAKLAY; WEBER, 2008)], no qual usuários podem colaborar adicionando e avaliando rotas. O protótipo considerou ruas para pedestres, caminhos por parques e calçadas. A fim de prover um modelo simplificado, os autores consideraram três nível de acessibilidade: bom, desconfortável e inacessível. Os autores apontaram que o planejamento de rotas considera uma aresta como uma extensão de uma rua ou caminho, por exemplo, a geometria de uma rua não é considerada ao criar o grafo do mapa.

Em (VÖLKEKEL; WEBER, 2008), é apresentado um sistema colaborativo multi-critérios para planejamento de rotas personalizadas para pedestres com algum tipo de deficiência em termos de mobilidade [*RouteCheckr*]. Mapas da Universidade Técnica de Desden foram manualmente pré-processados para criar uma rede navegável que incluem calçadas. Os autores usaram avaliações com base no tempo para refletir precisamente as condições da rede. Para computar rotas, os autores propuseram a normalização de critérios como distância e segurança e os associaram com outros pesos de acordo com as preferências do usuário. Os autores apontam que há a necessidade de um número limitado de critérios, caso contrário, a influência das avaliações para um critério pode diminuir e não resultar em nenhuma adaptação de rota. Contudo, eles não indicaram o número adequado de critérios que devem ser utilizados.

Baseado em entrevistas com cadeirantes, padrões definidos na Lei dos Americanos com Deficiência (ADA) e na literatura, os autores de (KASEMSUPPAKORN; KARIMI, 2008) identificaram parâmetros de acessibilidade de calçadas para computar rotas personalizadas. Um banco de dados espacial do campus da Universidade de Pittsburgh foi utilizado. Informações sobre calçadas, prédios e entradas acessíveis, obtidas a partir de imagens de satélite foram manualmente adicionadas. Apesar de os autores mencionarem que a rede de calçadas é modelada como um grafo e os valores das elevações são calculados baseados em linhas de contornos, não foi esclarecido de que maneira o processamento das rotas é realizado. Os mesmos autores propuseram em (KASEMSUPPAKORN; KARIMI, 2009) uma estratégia de rota personalizada para cadeirantes que leva em conta obstáculos

em calçadas, tráfego em calçadas e preferências do usuário. Três métodos foram propostos para computar pesos específicos para o usuário para parâmetros de calçadas: restrição absoluta, restrição relativa e redução de caminho.

Em (CHEN et al., 2015), os autores propuseram um sistema de planejamento de viagens personalizadas em que busca maximizar o número de pontos de interesse da rota baseados no local de início e de fim da viagem, preferências do usuário, a duração da visita, o tempo de operação em cada ponto de interesse e a condição de tráfego da cidade. O sistema deverá automaticamente fornecer a melhor rota para os pontos de interesse especificados se os requisitos do usuário forem possíveis de obter, caso contrário, o usuário é questionado sobre a remoção de alguns pontos de interesse. A rede dinâmica é modelada baseada em um grafo direcionado - onde cada nó corresponde a um ponto de interesse com cinco atributos: tempo de operação, categoria, popularidade, localização geográfica e tempo de permanência - e a condição de tráfego é obtida a partir de dados de GPS de táxis. Os autores utilizaram heurísticas para recomendar uma rota personalizada ao usuário. Os testes rodaram com dados reais da cidade de São Francisco.

Um campo de pesquisa é o desenvolvimento de sistemas de navegação para cadeiras de rodas motorizadas para auxiliar usuários com deficiências físicas, cognitivas ou visuais. Este tipo de sistema [(VISWANATHAN et al., 2011), (CHAVEZ; CALTENCO; BERAN, 2014)] geralmente combina posicionamento por GPS e mapeamento automático do ambiente para localização e mapeamento, algoritmos de menor caminho para planejamento de rotas, interfaces de controle biométrico e câmera estéreo-visuais para detectar objetos de interesse (FRINTROP; ROME; CHRISTENSEN, 2010) e para reduzir colisões frontais com o auxílio da computação visual e algoritmos de aprendizado de máquina.

Em comparação, para requisitos específicos como rotas acessíveis poucos trabalhos foram publicados. Existem algumas bibliotecas *online* e serviços públicos específicos para rotas acessíveis [OpenRouteService³, Routino⁴ e OpenTripPlanner⁵]. Tais serviços, geralmente utilizam um SIG para armazenar, manipular, analisar, gerenciar e apresentar dados geográficos via consultas SQL. Apesar de tais serviços levarem em conta condições de acessibilidade, eles são baseados em ruas. Conseqüentemente, não existem informação específica sobre as condições de uma calçada.

³ OpenRouteService - <https://openrouteservice.org/>. Último acesso em 29/05/2018

⁴ Routino - <https://www.routino.org/>. Último acesso em 29/05/2018

⁵ OpenTripPlanner - <http://www.opentripplanner.org/>. Último acesso em 29/05/2018

Em (MENKENS et al., 2011), os autores propuseram um sistema móvel de navegação social e suporte para cadeirantes. Este sistema fornece informações de acessibilidade sobre diversos pontos de interesse como transporte público, lojas, banheiros, estacionamentos para portadores de deficiência, etc. O sistema de planejamento de rotas é moldado conforme as preferências do usuário. Diversos atributos foram usados para fornecer rotas relevantes como tipo de rua, existência de calçadas, inclinação, informações atualizadas de construções. Foi usada informação geográfica obtida de maneira voluntária, disponível no projeto Open Street Map, para informações de acessibilidade sobre pontos de interesse e para fornecer um serviço de rota. Uma comunidade auxilia os usuários a comunicar, trocar ideias e atualizar o sistema com informações sobre barreiras e obstáculos. No entanto, não é esclarecido como os requisitos do usuário são aplicados para fornecer o menor caminho. Além disso, o sistema não especifica rotas em termos de calçadas.

(NEIS, 2015) também propõe um sistema de planejamento de rotas para cadeirantes que leva em conta as preferências do usuário. Diversos atributos são levados em conta no cálculo como o tipo da rua, a existência de calçadas, a iluminação e a inclinação. O autor propõe um modelo baseado em grafo a partir de dados geográficos disponibilizados pelo projeto Open Street Map onde o peso de cada aresta é definido a partir da combinação linear das preferências individuais de cada usuário. O caminho acessível resultante será baseado apenas nas preferências definidas pelo usuário. A avaliação das rotas resultantes foi realizada verificando a diferença entre os requisitos do usuário e o caminho gerado. Apesar deste sistema considerar calçadas no lado esquerdo, direito ou em ambos os lados da rua, o caminho resultante não especifica qual delas deve ser usada. Além disso, não há *feedback* do usuário sobre problemas de mobilidade. Além disso, conforme observado pelo autor, se uma aresta não possui um dos atributos utilizados como parâmetro, o peso da aresta utilizará o pior valor durante o processamento da rota.

Em março de 2018⁶, o Google anunciou uma nova funcionalidade para o Google-Maps que torna o transporte mais acessível para cadeirantes ou pessoas com necessidades em mobilidade. Em seis grandes cidades ao redor do mundo, são mostradas estações e rotas adequadas para um cadeirante. São usados conceitos apresentados em (KOZIEVITCH et al., 2016) e (MINETTO et al., 2016) como informações colaborativas sobre acessibilidade e um modelo baseado em calçadas para selecionar melhores rotas. Apesar de ser

⁶ Google Blog. Último acesso - <https://www.blog.google/products/maps/introducing-wheelchair-accessible-routes-transit-navigation/>. Último acesso 22/07/2018

possível reportar locais com questões de acessibilidade como elevadores para cadeirantes, entradas ou estacionamentos acessíveis não são mencionados problemas de acessibilidade em calçadas. O Google afirma que está sempre atualizando as imagens do *StreetView*.

2.4.2 Acessibilidade

Utilizando a mesma fonte de dados utilizadas neste trabalho, (LOYOLA; MEIRA, 2018) propõe duas aplicações com o objetivo de auxiliar na consulta de dados levantados pela população e pela prefeitura sobre condições de calçadas e rebaixamentos. A primeira consiste em uma API para acesso e manipulação de dados de forma segura através de outras aplicações. Com foco nas visualizações dessas informações, a segunda consistem em uma aplicação Web que permite a visualização clusterizada e o filtro desses dados de forma que não exija ao usuário nenhuma formação técnica para seu uso. Neste trabalho também realizou-se uma análise de um questionário sobre calçadas e rebaixamentos na cidade de Curitiba com objetivo de compreender melhor o problema estudado, assim como duas entrevistas com funcionários do IPPUC⁷ para coleta de informações sobre o que é utilizado atualmente por eles, testes práticos da aplicação web e os resultados obtidos no desenvolvimento das aplicações.

Em (CAMENAR; NASCIMENTO; ALMEIDA, 2018), os autores buscam analisar problemas de acessibilidade enfrentados por pessoas com deficiências físicas em grandes centros urbanos no contexto de países em desenvolvimento, como o Brasil. Foi conduzido um estudo na cidade Curitiba. Baseado em estudos disponíveis na literatura, padrões de mobilidade urbana estabelecidos pelo ABNT 9050 e técnicas de IHC (Interação Humano-Computador) eles utilizaram um método padrão conduzir a análise. O método consiste em um *Workshop* Participativo e em Análise de Serviços de Rotas. No *Workshop* Participativo, os pesquisadores selecionaram um Grupo Focal e propuseram uma série de discussões sobre experiências de mobilidade na cidade. Este tipo de método permite que sejam coletados dados mais rápido do que entrevistas pessoais. Na Análise de Serviços de Rotas, eles aplicaram uma Entrevista Semi-estruturada sobre problemas de mobilidade encontrados na cidade envolvendo duas atividades: (i) o participante deve desenhar uma rota em um mapa impresso, baseado na leitura e análise de instruções em formato textual e (ii) os participantes devem descrever uma rota verbalmente analisando um mapa

⁷ IPPUC - <http://www.ippuc.org.br/>

impresso. O resultado de tal estudo permitiu compreender melhor os problemas de mobilidade em grandes cidades de países em desenvolvimento e novas formas de planejar rotas para um cadeirante, como utilizar modelos baseados em calçadas e não baseado em ruas.

2.4.3 Proposta

A partir de um conjunto de camadas de mapas base (ruas, calçadas e quadras), (KOZIEVITCH et al., 2016) propõe a construção de um grafo com custos em suas arestas. A novidade neste trabalho é a possibilidade de utilizar dados abertos para considerar vários fatores (calçadas diferentes em uma rua ou ausência de rampas) para encontrar a melhor rota para uma cadeirante. Este grafo pode ser calibrado conforme as preferências individuais, este passo requer intervenção humana. Por exemplo, uma rota que não passe por grandes avenidas mas maximize o número de pontos de ônibus pode ser preferida por um cadeirante. Do ponto de vista de geometria, as ruas são uma ou mais linhas (que podem ser representadas como arestas do grafo). Quadras podem ser decompostas em linhas (contornos) e tais linhas em pontos. Com estes dados transformados é possível formalmente definir a partir do conjunto P de pontos uma rede $G=(V,E)$, onde V representa os vértices e E representa as arestas. Neste grafo, o peso de uma aresta é considerada a distância entre dois pontos. O último passo é identificar os cruzamentos que basicamente são extraídos a partir do cruzamento de duas ruas. Em cada cruzamento, haverá um conjunto de pontos de esquinas, que hipoteticamente podem ser considerados como rampas. É proposto um modelo baseado em calçadas mas ainda não considera inclinação como peso. No trabalho proposto aqui, complementa-se este modelo considerando ainda a inclinação de cada aresta como peso.

Em (MINETTO et al., 2016), buscando um serviço de rotas para cadeirantes mais realístico, os autores complementam o modelo proposto por (KOZIEVITCH et al., 2016) levando em conta o *feedback* do usuário para identificar problemas de acessibilidade. Construindo um modelo semelhante ao apresentado por (KOZIEVITCH et al., 2016), assume-se que nem sempre ele será acessível devido a problemas que podem existir nas calçadas por exemplo: ausência de rampas, calçadas fora de especificação, etc. A fim de resolver tal problema, é proposto que o usuário colabore com informações sobre problemas de acessibilidade que existam no mundo real. Tais informações devem sustentar o sistema para refinar futuras requisições de rotas. A fim de garantir a qualidade das informações

fornechas pelo usuário, eles consideraram duas análises complementares: métrica de massa crítica (GRUDIN, 1994) e verificação empírica realizada pelo órgão público responsável. A partir da colaboração dos usuários nos testes realizados em um bairro da cidade de Curitiba, foi possível identificar diversos problemas de acessibilidade reais. Os autores concluem afirmando que uma boa abordagem desse problema envolve a combinação de dados, domínio de conhecimento e *feedback* do usuário. Como trabalho futuro, a fim de refinar os resultados, eles sugerem a inclusão de dados de topografia, o que será realizado no trabalho aqui proposto.

CAPÍTULO 3

MODELO TEÓRICO

Este capítulo apresenta o modelo teórico desenvolvido neste trabalho. Ele está dividido da seguinte maneira: Aquisição de Dados, Arquitetura e Fornecimento de Serviço.

3.1 AQUISIÇÃO DE DADOS

Os dados adquiridos para este sistema vêm principalmente das três seguintes fontes: *sensoreamento urbano, feedback de cadeirantes e outros cidadãos e feedback de autoridades de planejamento urbano*. A agência de planejamento urbano provê informações geoespaciais sobre a rede de transportes (ruas, rodovias, calçadas, ciclovias e transporte público), localização de pontos de interesse (POIs) (hospitais, escolas, polícia, etc), topografia ou mapas de elevação, assim como informações dinâmicas sobre tráfego, ruído, acidentes, meteorologia. Em específico, neste trabalho foi utilizado um conjunto de camadas de mapas base, fornecido pelo Instituto de Pesquisa e Planejamento Urbano (IPPUC¹) que agrega as geometrias da cidade – ruas e quadras – bem como uma coleção de objetos baixo nível como pontos, linhas, polígonos com suas respectivas latitudes, longitudes e dados de elevação a fim de construir um modelo baseado em calçadas e cruzamentos para planejamento de rotas para cadeirantes. O tamanho das quadras da cidade variam de 0,002 a 6.094,127 metros quadrados.

O Banco de Dados foi construído utilizando PostGIS, um plugin para PostGres que permite a manipulação de dados geográficos. A fim de melhorar o desempenho das tabelas utilizadas, foram criados índices específicos para dados geográficos em cada tabela. Na tabela do grafo (*network*) foram criados ainda índices sobre as colunas que correspondem aos vértices.

¹ IPPUC - <http://www.ippuc.org.br/>. Último acesso em 05/01/2018.

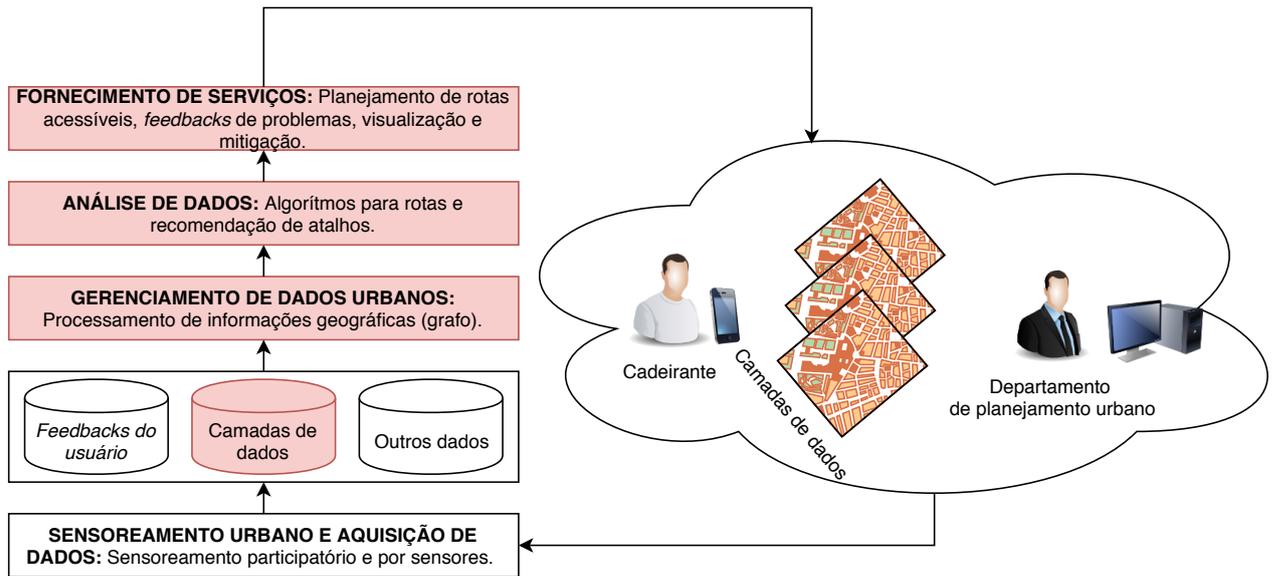
A novidade aqui é a maneira essas calçadas e cruzamentos são explorados por meio de arestas de um modelo de grafo: inicialmente utilizando seus tamanhos e, em uma segunda abordagem, utilizando sua inclinação. O modelo pode ser atualizado interativamente. Cadeirantes ou outros cidadãos podem reportar problemas, ajudando a construir um mapa de acessibilidade confiável da cidade e mantê-lo atualizado. O departamento de planejamento urbano age de acordo com esse *feedback* a fim de identificar gargalos nas rotas e melhorar as condições de acessibilidade da cidade.

3.2 ARQUITETURA

A arquitetura do sistema aqui proposto, ilustrada na Figura 8, é baseada no *framework* para computação urbana proposto por (ZHENG et al., 2014). Este *framework* é composto pelas seguintes camadas: *Sensoreamento urbano e Aquisição de dados*, *Gerenciamento de dados urbanos*, *Análise de dados e Fornecimento de Serviços*. A camada de Sensoreamento urbano é responsável por coletar dados através de sensores em geral, celulares, redes sociais e *feedback* do usuário. A **Camada de Gerenciamento de Dados Urbanos** é responsável por estruturar os dados para facilitar a sua análise de maneira eficiente. Na **Camada de Análise de Dados** é onde são usados algoritmos de otimização, de mineração de dados, aprendizado de máquina, otimização, visualização, e, no caso deste trabalho, de planejamento de rotas. Por fim, a **Camada de Fornecimento de Serviços** provê algum serviço específico ao usuário, no caso desse trabalho fornece ao usuário a menor rota, a visualização e o reporte de problemas. A Figura 8 destaca em vermelho as camadas que serão utilizadas neste trabalho. As seções 3.3 e 3.4 detalham de que maneira este trabalho seguiu a arquitetura aqui apresentada.

(BARCZYSZYN, 2015) realizou a integração das diversas camadas dos dados fornecidos pelo IPPUC (ruas, quadras, escolas, hospitais, etc) formando um banco de dados geográfico. Derivado de tal banco de dados será criado um novo banco de dados no qual serão utilizadas algumas camadas pertencentes ao domínio de arruamento (quadras, ruas e calçadas) e ao domínio de altimetria (curvas de nível). A Figura 9a mostra a visualização dos dados do domínio de quadras combinados com o fundo do *OpenStreetMaps*. Já a Figura 9b mostra visualização dos dados do domínio de altimetria também com o *OpenStreetMaps*.

Figura 8 – Arquitetura baseada no modelo para computação urbana.



Fonte: Adaptado de (ZHENG et al., 2014)

O script a seguir mostra de criação da tabela de quadras, onde informações de tamanho da área e do comprimento (colunas *shape_area* e *shape_len*, respectivamente), geometria da quadra (coluna *geom*) são armazenadas:

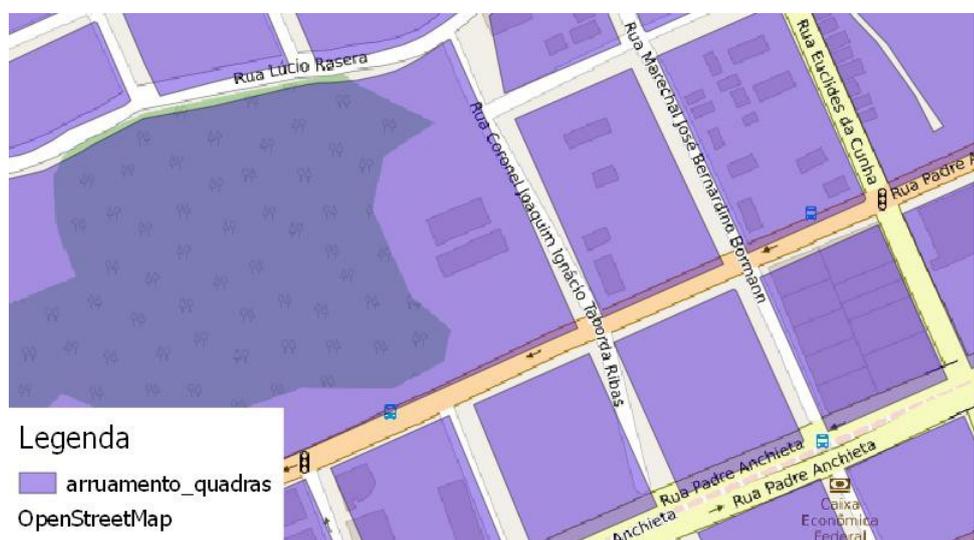
```
CREATE TABLE arruamento.arruamento_quadras
(
    gid SERIAL NOT NULL,
    tipo CHARACTER VARYING(50),
    shape_area NUMERIC,
    shape_len NUMERIC,
    geom GEOMETRY(MultiPolygon, 4326),
    CONSTRAINT arruamento_quadras_pkey PRIMARY KEY (gid)
)
```

3.3 GERENCIAMENTO DE DADOS URBANOS

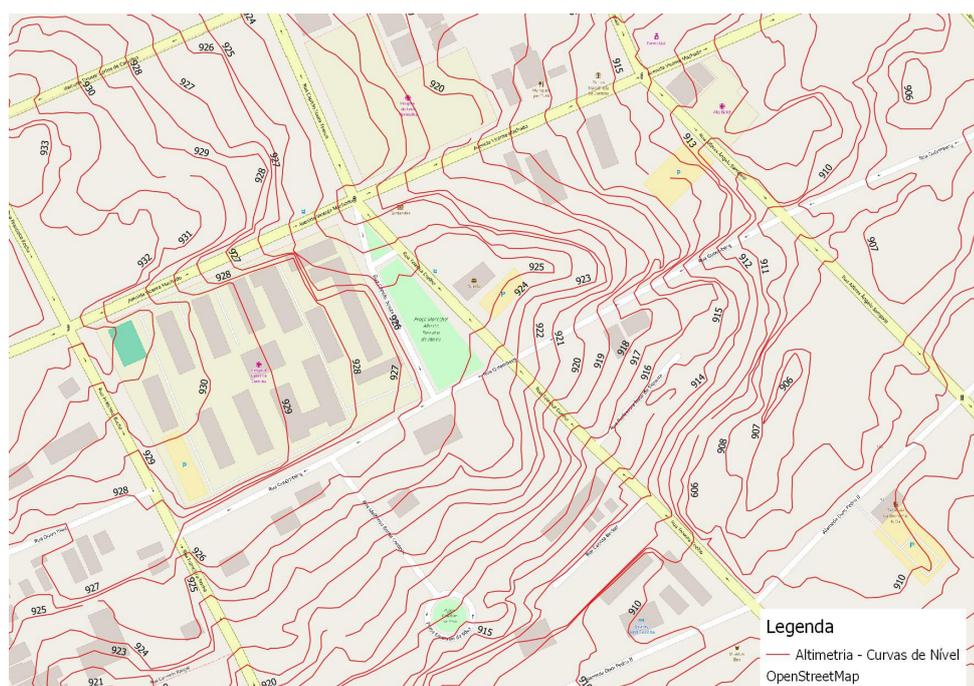
Conforme dito na seção 3.2 este trabalho segue a arquitetura proposta por (ZHENG et al., 2014). Esta seção descreve de que maneira a camada de Gerenciamento de Dados de tal arquitetura foi desenvolvida neste trabalho.

Figura 9 – Domínios de dados geográficos utilizados na implementação.

(a) Domínio de arruamento e OpenStreetMaps.



(b) Domínio de altimetria e OpenStreetMaps.



3.3.1 Construção do Grafo

As Figuras 10 e 11 mostram um exemplo da construção do grafo baseado em calçadas. Os polígonos das quadras podem possuir diversos vértices e arestas a fim de representar precisamente as suas delimitações. Comumente, o número de vértices em uma quadra aparentemente retangular é maior que quatro (ver Figura 10b). Os vértices em excesso geralmente são devido a detalhes na geometria das quadras. Foi realizada então uma simplificação dos polígonos removendo vértices em locais de pequenas curvaturas

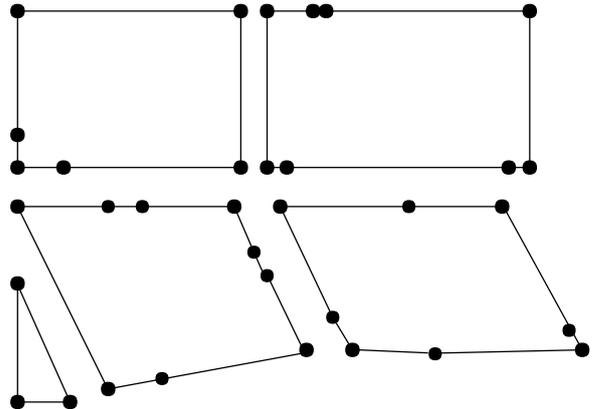
(realizado através de uma função espacial do banco dados chamada *st_simplify*). O motivo para tal simplificação são os seguintes: (a) a modelagem das calçadas tornam-se mais simples e (b) permanecem vértices das esquinas, onde geralmente estão localizados rampas e/ou cruzamentos (ver Figura 11b).

Figura 10 – Quadras da cidade e sua representação pelo SIG.

(a) Imagem de satélite de quadras da cidade de Curitiba



(b) Linhas e pontos usados para representar as quadras em um SIG

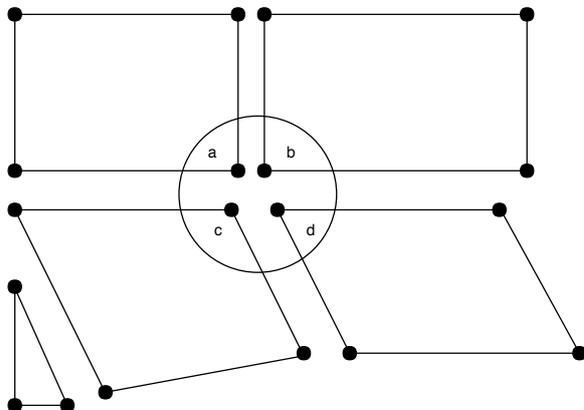


Assumindo que rampas estão localizadas nos vértices dos polígonos simplificados, foi realizada uma delimitação em cada cruzamento e agrupados os vértices dentro dessa região. A Figura 11a demonstra o agrupamento dos pontos *a*, *b*, *c* e *d*. As arestas das bordas do polígono simplificado formado utilizando pontos agrupados (criado utilizando a função espacial do banco de dados *st_convexhull*) foram inclusas no modelo para representar cruzamentos e, conseqüentemente, rampas. A Figura 11b mostra um exemplo com todos os cruzamentos já identificados.

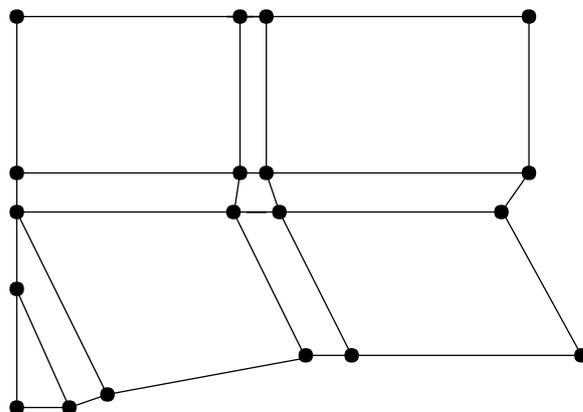
Formalmente, as geometrias das calçadas e cruzamentos foram modeladas como um grafo planar não-direcionado $G = (V, E)$. O conjunto de vértices V corresponde às esquinas das quadras ou também as rampas. Calçadas ou cruzamentos são representadas pelo conjunto de arestas $E \in (V \times E)$. Portanto, uma calçada ou cruzamento é uma aresta, um par de vértices (u, v) conectando os pontos u e v em V . Cada vértice $v \in V$ possui latitude, longitude e valores de inclinação e cada aresta $(u, v) \in E$ possui um peso primário $d(u, v)$ que corresponde à distância, em metros, entre os pontos u e v e um valor adicional para inclinação (ver Seção 3.3.2) e coeficiente de acessibilidade (ver Seção 4.3.3). Essas medidas estão descritas nas próximas seções.

Figura 11 – Geometrias simplificadas e representação dos cruzamentos

(a) Quadras simplificada e região utilizada para agrupar pontos (a , b , c e d) onde há rampas.



(b) Pontos conectados para representar os cruzamentos do modelo.



3.3.2 Inclinação

A inclinação de um caminho deve ser levada em conta em um sistema de planejamento de rotas para cadeirantes a fim de fornecer um caminho adequado. Portanto, considerou-se as linhas topográficas da cidade, conforme mostra a Figura 12, como elementos relevantes para a obtenção dos pesos das arestas.

Inicialmente, as interseções entre as arestas do grafo e as linhas topográficas foi obtida através do algoritmo de intersecção segmento-segmento proposto por (BENTLEY; OTTMANN, 1979) realizado pelas funções *st_dump* e *st_intersection* do PostGIS. Dada uma aresta (u, v) (uma calçada ou cruzamento, por exemplo), tal que $\{p_0 = u, p_1, p_2, \dots, p_n = v\}$ é o conjunto de pontos formado pela união dos pontos extremos u e v e as interseções pela aresta com um conjunto de linhas topográficas (ver Figura 12a).

Para cada segmento $(p_i, p_{i+1}), 0 \leq i < n$, calcula-se o percentual do grau de inclinação da seguinte maneira [(ABNT, 2015)]:

$$g(p_i, p_j) = \frac{100 \times d_v(p_i, p_j)}{d_h(p_i, p_j)} \quad (2)$$

onde $d_v(p_i, p_j)$ denota o deslocamento vertical de p_i para p_j (subida), $d_h(p_i, p_j) = \sqrt{d(p_i, p_j)^2 - d_v(p_i, p_j)^2}$ é o deslocamento horizontal (descida) e $d(p_i, p_j)$ é a distância entre dois pontos. A Figura 12b mostra exemplos de valores calculados para cada segmento sobre uma aresta (u, v) . Note que g é negativo para uma descida de p_i para p_j , positivo para uma subida e zero para uma região plana. O valor para essa métrica não deve ser maior

do que 8% para um acesso sem dificuldades, e é comumente usado como referencial para especificação de rampas ou outros obstáculos para cadeirantes como descreve a norma NBR9050 (ABNT, 2015).

Dada a aresta (u, v) de uma calçada e cruzamento, realiza-se a somatória do deslocamento vertical para calcular o *esforço de elevação*, isto é

$$E_{elev} = \sum_{p \in (u,v), p_i \neq v} |d_v(p_i, p_{i+1})| \quad (3)$$

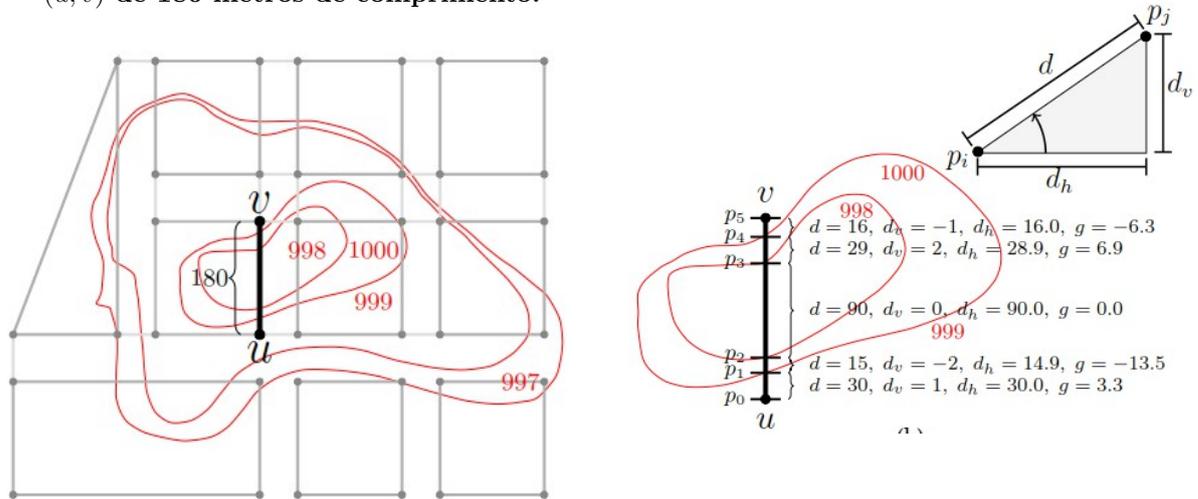
e do grau percentual de inclinação g deriva-se a métrica de *máximo esforço* da seguinte maneira

$$E_{max}(u,v) = \max_{p \in (u,v), p_i \neq v} \{|g(p_i, p_{i+1})|\} \quad (4)$$

Figura 12 – Grau de inclinação do caminho.

(a) Grafo sobreposto por quatro linhas topográficas (destacadas em vermelho) com elevações de 997, 998, 999 e 1000 metros sobre uma aresta de calçada (u, v) de 180 metros de comprimento.

(b) As interseções sobre a aresta (u, v) com as curvas de nível decompõe (u, v) em cinco segmentos com diferentes graus de acessibilidade $g(p_i, p_{i+1})$



Foram usados valores em módulo $|\cdot|$ em tais métricas pois considerou-se que tanto subidas quanto descidas são igualmente custosas em termos de mobilidade. Note que o cálculo dessas medidas é realizado apenas uma vez. Os valores são armazenados como pesos para a aresta (u, v) e não há necessidade de incluir interseções no grafo espacial. Deve-se ter cuidado ao definir a elevação nos vértices u e v . Se considerar $d(p_0, p_1) = 0,7$, na Figura 12, serão obtidos valores inválidos para d_h e g . Tais casos devem ser tratados

separadamente. A interpolação do valor de elevação de p_0 pode ser usado, por exemplo.

3.3.3 Coeficiente de Acessibilidade

Após a resolução de questões sobre a inclinação, assume-se que todas as calçadas e cruzamentos são acessíveis para cadeirantes. No entanto, tal afirmação as vezes pode não ser verdadeira, assim como pode-se nota na Figura 26. É possível que uma aresta deste modelo conecte duas quadras e que não haja rampas no cruzamento. No entanto, este modelo deve refletir tal problema. Foram encontradas situações de pobre manutenção e obstruções que tornariam uma calçada impossível de se transitar. As vezes, a calçada até mesmo não existe. Não obstante, uma métrica realística de acessibilidade deve levar em conta o *feedback* dos usuários para cada uma dessas situações.

Para isso, define-se neste trabalho o coeficiente de acessibilidade, que consiste em uma métrica que indica se uma calçada possui ou não tais problemas de acessibilidade. O coeficiente de acessibilidade para uma aresta (u, v) apresenta valor zero na ausência de problemas de acessibilidade e infinito, caso contrário, então a aresta não pode ser utilizada. A medida é dinâmica e depende da interação entre os cidadãos e o departamento de planejamento urbano da cidade. Tal serviço é melhor descrito na seção 3.4. Por fim, note que o refinamento do grafo pode ser feito inicialmente utilizando imagens do *Google Street View*. No entanto, tais imagens são geralmente desatualizadas, portanto, essa metodologia irá funcionar apenas com o *feedback* do usuário.

3.4 FORNECIMENTO DE SERVIÇO

Seguindo a arquitetura proposta por (ZHENG et al., 2014), essa seção demonstra de que maneira a camada de fornecimento de serviço foi utilizada neste trabalho.

3.4.1 Planejamento de rotas com acessibilidade

Dados um local de origem v_0 e um local de destino v_k ($v_0, v_k \in V$), a tarefa do planejador de rota é computar um caminho que minimize o custo de viagem de acordo com algum critério. Um caminho $p(v_0, v_k) = (v_0, v_1, \dots, v_k)$ é uma sequência de arestas iniciando em v_0 e terminando em v_k . O custo do caminho é definido como a soma dos pesos das arestas pelo caminho, isto é, $w(p) = \sum_{i=1}^k w(v_{i-1}, v_i)$. Formalmente, planejadores de

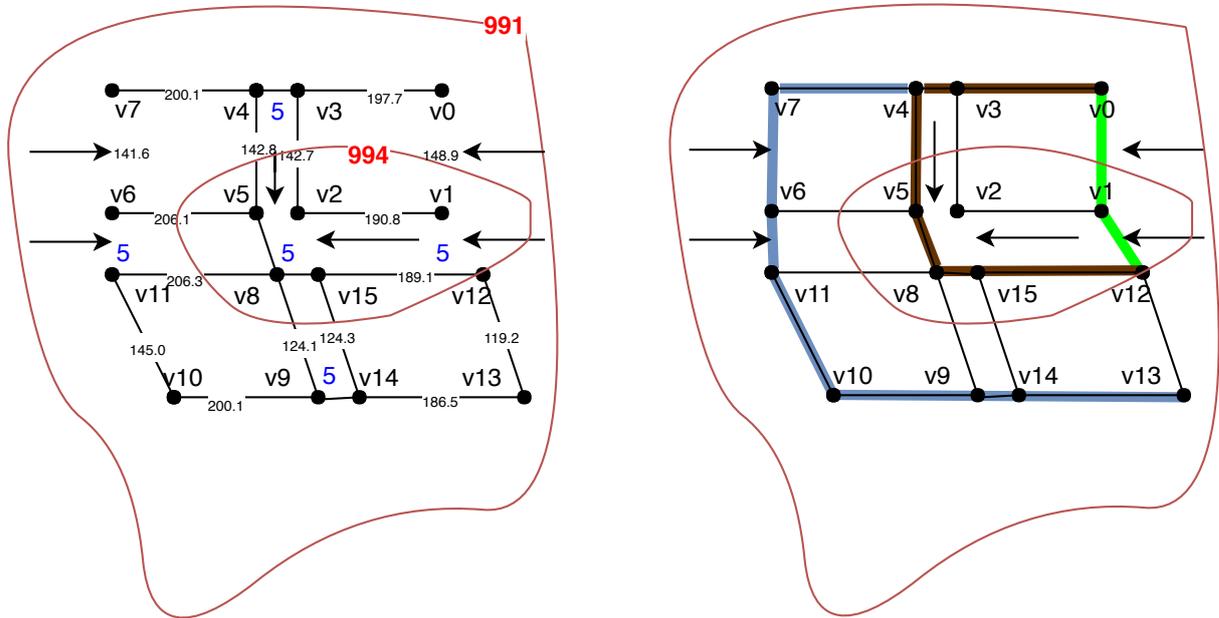
rotas geralmente assumem que o peso para computar o melhor caminho é a distância. $w(u, v) = d(u, v)$. No entanto, em um contexto de cadeirantes, é desejável encontrar um caminho que minimize vários custos distintos simultaneamente, como distância, inclinação e problemas de acessibilidade. Infelizmente, um algoritmo padrão de menor caminho, como o Dijkstra [(CORMEN; STEIN; RIVEST, 2001)], não garante uma rota que minimize diversos fatores ao mesmo tempo. Uma alternativa seria combinar linearmente os custos existentes em um único ou encontrar um caminho de custo mínimo cujo consumo de recursos obedeça alguns limites – por exemplo, encontrar o caminho de menor distância sem problemas de acessibilidade e com inclinação máxima de cinco metros. No entanto, tal estratégia geralmente leva ao problema de como unir o conhecimento adquirido de múltiplas fontes dados heterogêneas [(ZHENG et al., 2014)]. Tal problema é conhecido como um problema NP-Completo [(ZIEGELMANN, 2007)], isto é, o algoritmo ótimo para resolvê-lo possui complexidade exponencial.

A estratégia adotada para este problema de rota com custos múltiplos é utilizar o algoritmo de Dijkstra com cada custo separado – para computar os limites inferiores para distância, inclinação e problemas de acessibilidade. Usando a distância como custo primário, calcula os $n - 1$ melhores casos de menores caminhos utilizando o algoritmo de Yen [(YEN, 1970)]. Então, dentro desses $n - 1$ caminhos, busca-se por aqueles com menores inclinações e problemas de acessibilidade. Note que os caminhos com menores inclinações e problemas de acessibilidade de todo o modelo podem não estar na lista dos melhores $n - 1$ caminhos encontrados anteriormente, no entanto, neste caso, sabe-se que estes caminhos ótimos não podem ser obtidos dentro de uma distância menor do que aquela do $(n - 1)$ -ésimo caminho. Em suma, o planejador de rota aqui apresentado sempre trará o caminho de menor distância – o primeiro na lista de Yen – e, se este caminho não satisfazer todos os limites inferiores, seleciona-se dos $n - 1$ caminhos restantes – ordenados ascendentemente em termos de distância - um caminho de menor inclinação (Seção 3.3.2) e outro com o menor número de problemas de acessibilidade (Seção 4.3.3). A complexidade de tempo do algoritmo de Yen é $\Theta(nV(E + V \log V))$ para um grafo com n arestas, V vértices e E arestas. Um exemplo de Rota de Yen é ilustrada na Figura 13.

Na Figura 13 encontra-se a lista de menores caminhos de Yen. Na esquerda, as distâncias e a topografia de quatro quadras – as seis flechas indicam problemas de acessibilidade em duas calçadas e em quatro cruzamentos; direita, três exemplos do caminho de Yen: o caminho de menor distância (em verde), o caminho com menor problema de

acessibilidade (em marrom) e o caminho com menor inclinação (em azul):

Figura 13 – Rotas mínimas de Yen.



- $p(v_0, v_k) = \{v_0, \dots, v_k\} = (\text{distância, inclinação, problemas de acessibilidade})$
- $p(v_0, v_{13}) = \{v_0, v_1, v_{12}, v_{13}\} = (272, 2, 6, 2)$
- $p(v_0, v_{13}) = \{v_0, v_3, v_4, v_5, v_8, v_{15}, v_{14}, v_{13}\} = (652, 9, 6, 0)$
- $p(v_0, v_{13}) = \{v_0, v_3, v_4, v_7, v_6, v_{11}, v_{10}, v_9, v_{14}, v_{13}\} = (1086, 0, 2)$

CAPÍTULO 4

IMPLEMENTAÇÃO E TESTES

Com o modelo teórico desenvolvido, é possível utilizá-lo para implementar o sistema proposto neste trabalho. Este capítulo tem o objetivo de demonstrar de maneira prática como foi realizada a implementação deste modelo e apresentar alguns testes realizados.

4.1 TECNOLOGIAS UTILIZADAS

Para o desenvolvimento deste trabalho, as seguintes tecnologias foram utilizadas: como Sistema Gerenciador de Banco de Dados (SGBD) foi utilizado o PostgreSQL¹ versão 9.5. A extensão PostGIS² versão 2.2 foi utilizada para o manejo de dados geoespaciais. Para a visualização dos dados em mapas, foi utilizado o QuantumGIS³ versão 2.16.3. Por fim, para o cálculo de rotas, utilizou-se o plugin PgRouting versão 2.2.2. O PgRouting⁴ é uma extensão para PostGIS que fornece diversas funções no próprio banco de dados para cálculo de caminhos mínimos utilizando como entrada dados geográficos. O PgRouting traz diversas implementações de algoritmos de caminho mínimos como Dijkstra, A*, entre outros. Neste trabalho, as implementações desses algoritmos no PgRouting serão utilizadas para encontrar as rotas desejadas.

4.2 IMPLEMENTAÇÃO DO GRAFO

Nesta seção, demonstra-se como o grafo foi implementado seguindo o modelo teórico apresentado na seção 3.3.1. Com esta implementação, busca-se que o grafo esteja

¹ PostgreSQL - <https://www.postgresql.org/>. Último acesso em 23/07/2018

² PostGIS - <https://postgis.net/>. Último acesso em 23/07/2018

³ QuantumGIS - https://www.qgis.org/pt_BR/site/ltimoacessoem23/07/2018.

⁴ PgRouting - <https://pgrouting.org/>. Último acesso em 23/07/2018.

criado dentro de um contexto de banco de dados geográfico e que seja possível executar algoritmos de menor caminho dentro deste mesmo contexto.

Inicialmente, tinham-se as geometrias dos blocos das quadras da cidade. Era necessário, a partir delas, extrair seus contornos, onde estão as calçadas. Tendo tais contornos, foi necessário identificar os cruzamentos e criar as suas linhas. Por fim, foi possível gerar um grafo utilizando as linhas obtidas até então.

4.2.1 Extração de calçadas

No banco de dados fornecido pelo IPPUC⁵ existiam geometrias do tipo polígono das quadras da cidade conforme ilustra a Figura 14a. Esses polígonos foram o ponto inicial para a criação do grafo. Utilizando a função *ST_ExteriorRing*⁶ do *PostGIS* foi possível obter o contorno de tais geometrias, o que foi considerado-se como as calçadas. A Figura 14b mostra as linhas resultantes que representam as calçadas.

4.2.2 Obtenção dos cruzamentos

Para criar os cruzamentos, foi necessário identificar, agrupar pontos iniciais e finais das travessias e, por fim, gerar linhas dessas travessias. O primeiro passo para a obtenção dos cruzamentos foi descobrir a localização de cada um deles. Para isso, utilizou-se o domínio de rua. Nele, geometrias do tipo linha são utilizadas para representar as ruas da cidade. Utilizando a função *ST_Intersects*⁷ do *PostGIS*, foi possível gerar pontos onde existiam cruzamentos na cidade conforme mostra a Figura 15.

O segundo passo foi identificar os pontos do cruzamento, isto é, pontos iniciais e finais em caso de travessia. Para isso, foi necessário identificar nas linhas das calçadas os pontos onde elas formavam esquinas. Para isso utilizou-se a função *ST_Dump*⁸ do *PostGIS*. A fim de evitar que fossem gerados pontos em locais que não faziam parte de cruzamentos, foi necessário simplificar as linhas de calçadas e ignorar pequenas quebras

⁵ IPPUC - <http://www.ippuc.org.br/>. Acesso em 19/02/2019

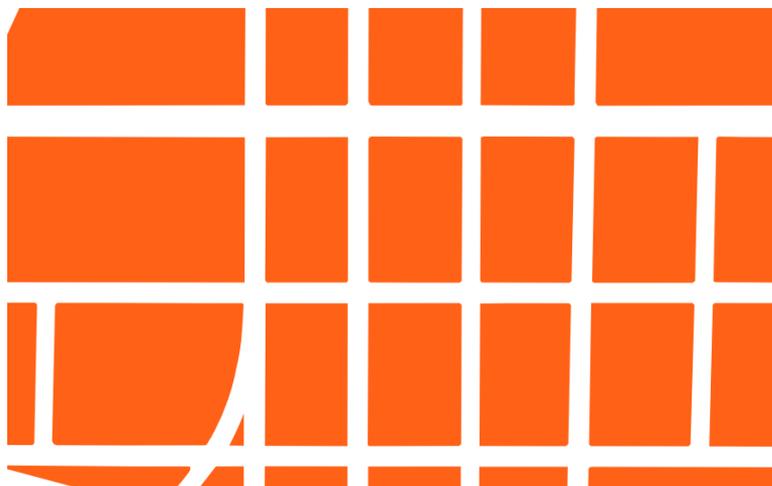
⁶ PostGIS (Documentação): https://postgis.net/docs/ST_ExteriorRing.html. Acesso em 08/05/2018

⁷ PostGIS (Documentação): https://postgis.net/docs/ST_Intersects.html. Acesso em 08/05/2018

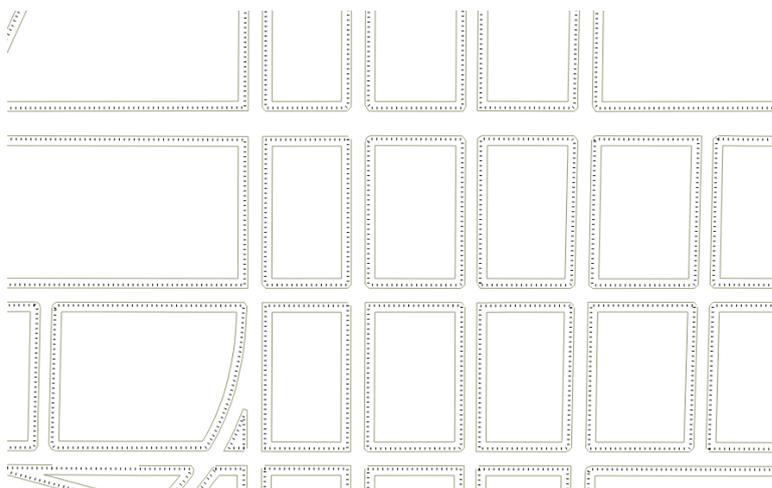
⁸ PostGIS (Documentação): https://postgis.net/docs/ST_Dump.html. Acesso em 08/05/2018

Figura 14 – Extração dos contornos das quadras.

(a) Geometrias (polígonos) iniciais das quadras.



(b) Linhas resultantes da extração dos contornos das quadras - as calçadas.



através da função *ST_Simplify*⁹ do *PostGIS*. A Figura 16 mostra os pontos que foram identificados após essa etapa.

Foi necessário então identificar quais pontos faziam parte de cada cruzamento. Para isso, gerou-se um raio em torno de cada ponto central de cruzamento gerado. Utilizando a função *ST_Within*¹⁰ foi possível identificar quais pontos de esquina estavam dentro do raio e, conseqüentemente, quais faziam parte de cada cruzamento. Foi possível, portanto, agrupar os pontos das esquinas por cruzamento. A Figura 17 mostra um raio agrupando todos os pontos que fazem parte de um determinado cruzamento.

⁹ PostGIS (Documentação): https://postgis.net/docs/ST_Dump.html. Acesso em 08/05/2018

¹⁰ PostGIS (Documentação): https://postgis.net/docs/ST_Within.html. Acesso em 08/05/2018

Figura 15 – Pontos que identificam cada cruzamento da cidade.

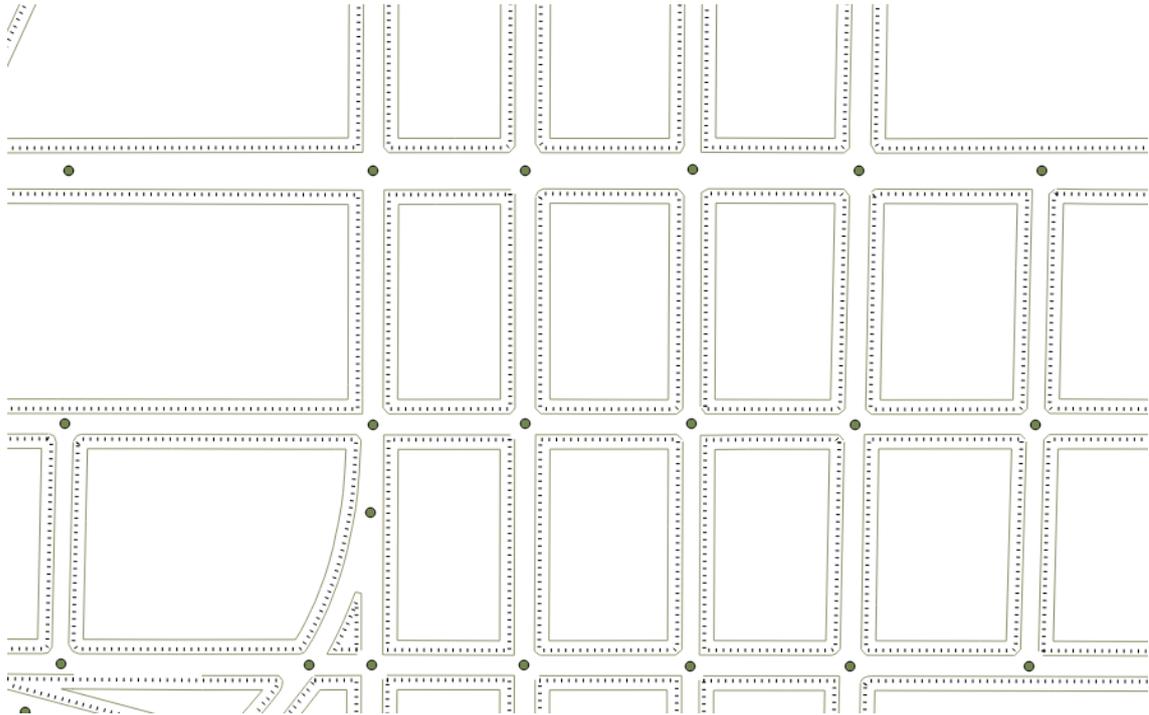
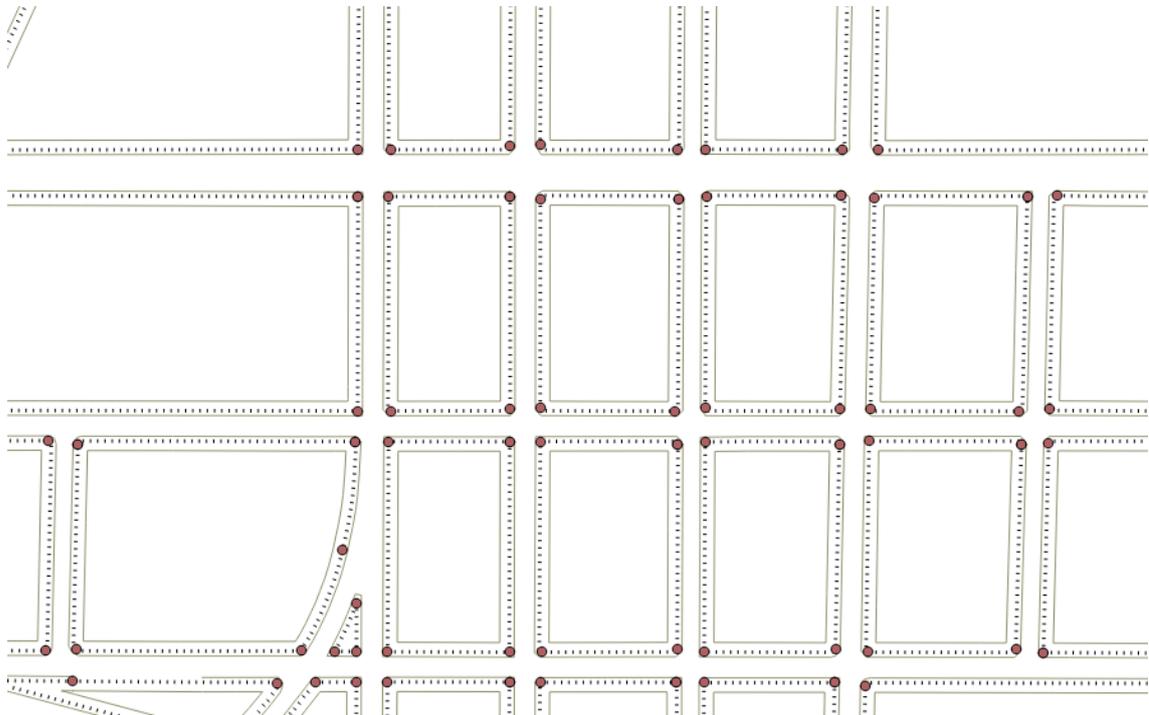
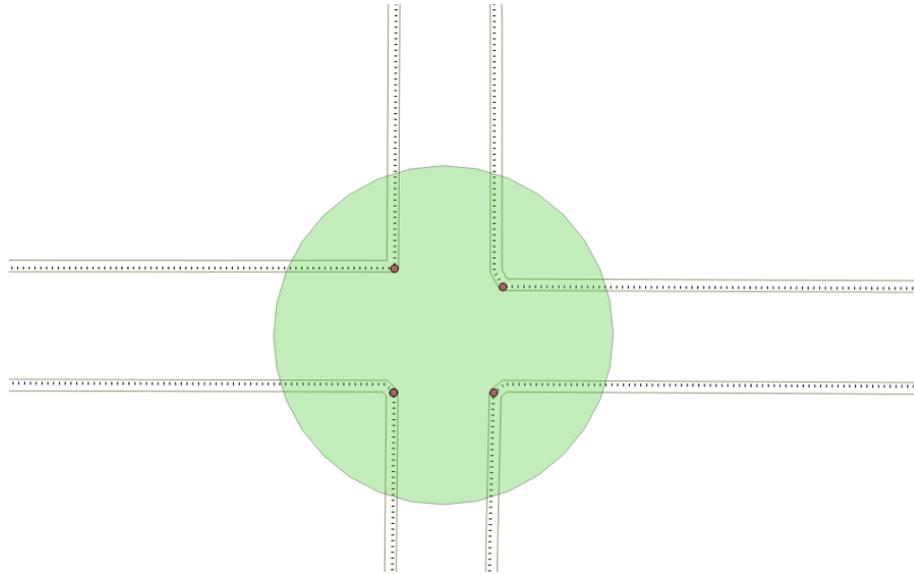
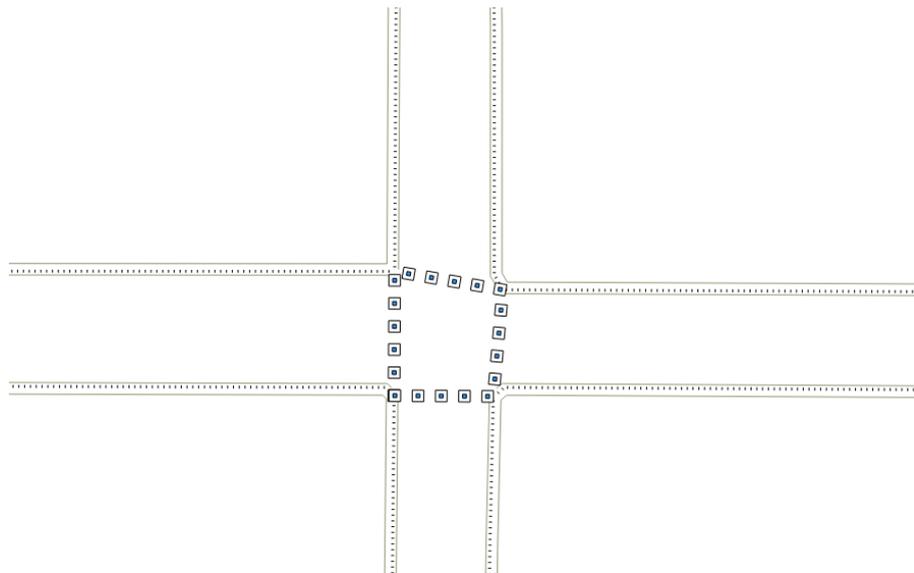


Figura 16 – Pontos das esquinas identificados.



A última etapa para a obtenção dos cruzamentos foi ligar cada uma das esquinas de maneira que formassem as linhas das travessias. Utilizando a função *ST_Convexhull*¹¹ do *PostGIS* foi possível gerar polígonos entre esses pontos. Retirando o contorno desses polígonos, obteve-se linhas que formavam as travessias conforme mostra a Figura 18.

¹¹ PostGIS (Documentação): https://postgis.net/docs/ST_Convexhull.html. Acesso em 08/05/2018

Figura 17 – Raio agrupando pontos de esquinas de um cruzamento.**Figura 18 – Travessias dos cruzamentos.**

4.2.3 Grafo baseado em calçadas

Para gerar o grafo, foi necessário separar as linhas das calçadas e cruzamentos em dois objetos: arestas e vértices. Para os vértices foram considerados todos os pontos de um cruzamento; as arestas, cada linha entre esses pontos.

Para cada aresta, foi necessário identificar os seus vértices iniciais e finais. Em seguida, foi necessário identificar cada vértice através de sua localização e atribuir um identificador distinto (número inteiro) para cada um. Finalmente, foi possível gerar uma tabela onde cada registro armazena uma aresta do grafo com os identificadores dos vértices.

tices inicial e final bem como suas geometrias e seus custos (distância e coeficiente de inclinação).

4.2.4 API

Foi desenvolvida uma *API* a fim de fornecer o serviço de planejamento de rotas usando o caminho mínimo e inclinação. Com a disponibilização do serviço, é possível que outros trabalhos possam utilizá-lo desenvolvendo aplicações clientes.

Esta API consiste em uma aplicação servidor que recebe como parâmetro de entrada as coordenadas inicial e final da rota desejada. Como resposta, o serviço retornará um vetor de elementos contendo os vértices da rota resultante com suas coordenadas, seu número de sequência, custo e custo acumulado. Tal aplicação foi desenvolvida seguindo a arquitetura de comunicação REST. Com tecnologias usou a linguagem Java¹² com o framework para serviços *SpringBoot*¹³. Seu padrão de dados segue o formato JSON.

O fluxo básico de requisição é demonstrado na Figura 19. Uma aplicação cliente (*smartphone* ou aplicação *web*) fazem a requisição REST para a camada da API enviando as coordenadas iniciais e finais. A API irá se comunicar com o Postgres, requisitando que os dados sejam processados utilizando um algoritmo de caminho mínimo através do PgRouting através da consulta a seguir.

```
SELECT seq , path_seq , node , edge , cost , agg_cost
FROM grafo.network
JOIN
(SELECT * FROM pgr_dijkstra (
SELECT gid AS id ,
start_id::int4 AS source ,
end_id::int4 AS target ,
ST_Length(ST_MakeLine(st_startpoint , st_endpoint)) AS cost
FROM grafo.network ' ,
79746 ,
77823 ,
false)) AS route
```

¹² Java - https://www.java.com/pt_BR/. Último acesso em 25/06/2018.

¹³ SpringBoot - <https://spring.io/projects/spring-boot>. Último acesso em 25/06/2018

ON

```
network.gid = route.edge;
```

O PgRouting utilizará como grafo uma tabela resultante dos passos anteriores para encontrar o melhor caminho. O PgRouting retorna o resultado em formato JSON para a API que por sua vez o entrega para a aplicação cliente.

Um aspecto que vale ressaltar é que pelo fato de os caminhos serem baseados em pontos finais e iniciais que são representados pelos vértices do grafo, o serviço utilizará como vértices inicial e final do caminho desejado aqueles que estão mais próximos das coordenadas solicitadas.

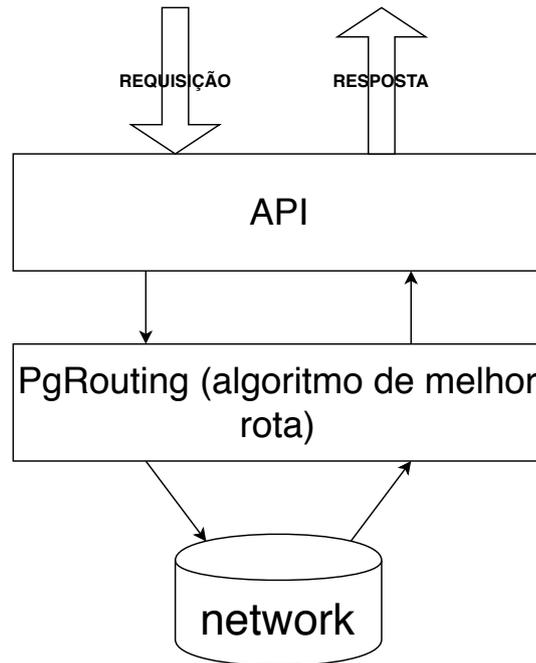
Como resposta, o serviço retornará um vetor ordenado em formato *JSON* das coordenadas dos vértices pertencentes ao caminho contendo: i) o identificador único de cada rota (*routeId*), ii) a sequência do vértice (*sequence*), iii) a latitude e vi) longitude de cada ponto, v) a distância de cada trecho (*distance*) e vi) a distância acumulada até aquele vértice (*accumulatedDistance*) conforme o *payload* a seguir:

```
[
  {
    "routeId": 0, (i)
    "sequence": 1, (ii)
    "latitude": -49.28266601336963, (iii)
    "longitude": -25.432840098324153, (vi)
    "distance": 123.51022004324217, (v)
    "accumulatedDistance": 0 (vi)
  }
]
```

4.3 TESTES

Após a elaboração de um modelo baseado em calçadas que considera inclinação e questões de acessibilidade e a sua implementação, foram realizados testes para cada um dos seus pontos propostos.

Figura 19 – Fluxo de requisições para a API.



4.3.1 Menor caminho em calçadas

A fim de validar a implementação do grafo baseado em calçadas conforme proposto na seção 3.3.1, foram realizados testes executando o algoritmo de Dijkstra através do PgRouting sobre o grafo implementado. Os parâmetros necessários para rodar este teste são nós dos pontos iniciais e nós dos pontos finais da rota desejada. O teste foi delimitado ao bairro Batel, com pontos iniciais e finais selecionados ao acaso.

Nesta consulta, indica-se na tabela do grafo (*network*) quais campos serão utilizados como *source* e *target* que são os campos onde o algoritmo buscará, respectivamente, os *id* dos vértices inicial e final em cada registro de aresta. Além de indicar qual campo será considerado para calcular o custo (*cost*). Indica-se ainda qual é o vértice inicial e qual é o vértice desejado para todo o caminho. Como resultado da consulta, tem-se as seguintes informações:

- *seq* – número sequencial do resultado.
- *path_seq* – número sequencial do caminho encontrado.
- *node* – identificador do vértice referente àquele registro.
- *edge* – identificador da aresta referente àquele registro.
- *cost* – custo da aresta.

- *agg_cost* – custo acumulado até aquele vértice.

A consulta a seguir exemplifica de que maneira executa-se o algoritmo de Dijkstra utilizando o pgRouting:

```
SELECT seq , path_seq , node , edge , cost , agg_cost
FROM grafo.network
JOIN
(SELECT * FROM pgr_dijkstra (
SELECT gid AS id ,
start_id::int4 AS source ,
end_id::int4 AS target ,
ST_Length(ST_MakeLine(st_startpoint , st_endpoint)) AS cost
FROM grafo.network ' ,
79746 ,
77823 ,
false)) AS route
ON
network.gid = route.edge;
```

É possível ainda visualizar o resultado da execução utilizando o QGIS¹⁴, conforme ilustra a Figura 20. A fim de verificar se o caminho fornecido ocorre sempre nas bordas das quadras (calçadas), utilizou-se o *Google Maps* como *background*.

4.3.2 Inclinação

Os testes de inclinação foram realizados no bairro Batel. Na seção 3.3.2 foi demonstrado teoricamente como os dados de inclinação foram convertidos em informações de custo para o modelo proposto. A Figura 21 exemplifica a transformação de dados de altimetria em pequenos segmentos nas aresta, dentro dos quais serão calculado os esforço máximo. Utilizou-se o mesmo grafo dos demais testes. No entanto, não foi utilizada a distância como peso mas sim o esforço máximo.

¹⁴ QGIS - <https://www.qgis.org/en/site/>

Figura 20 – Resultado da execução do algoritmo de Dijkstra na implementação do modelo baseado em calçadas.

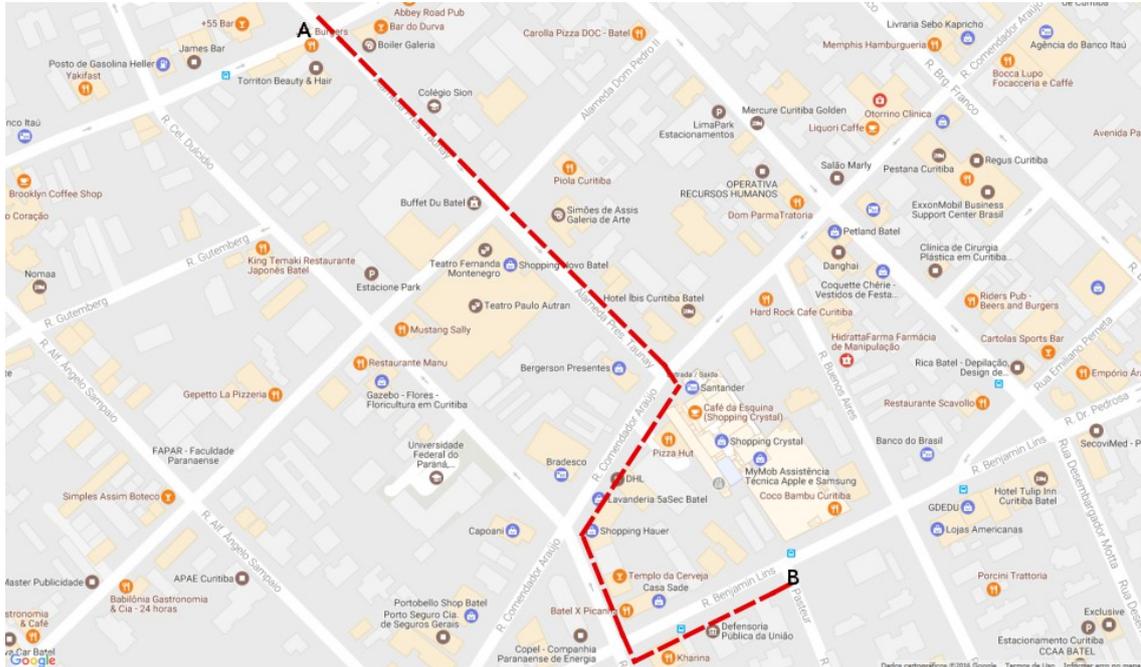
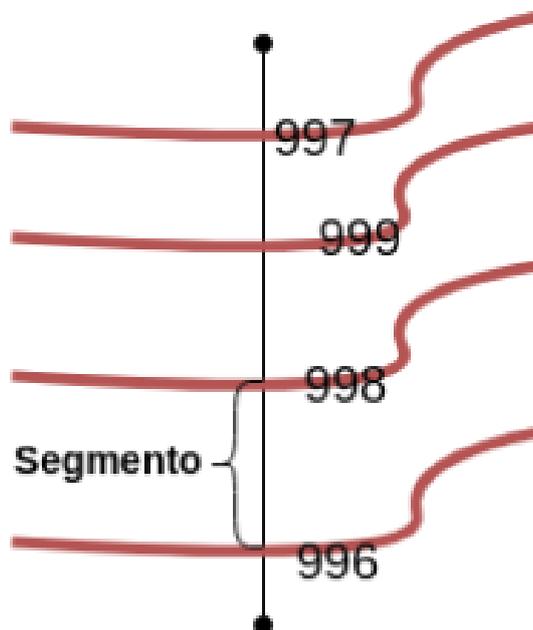


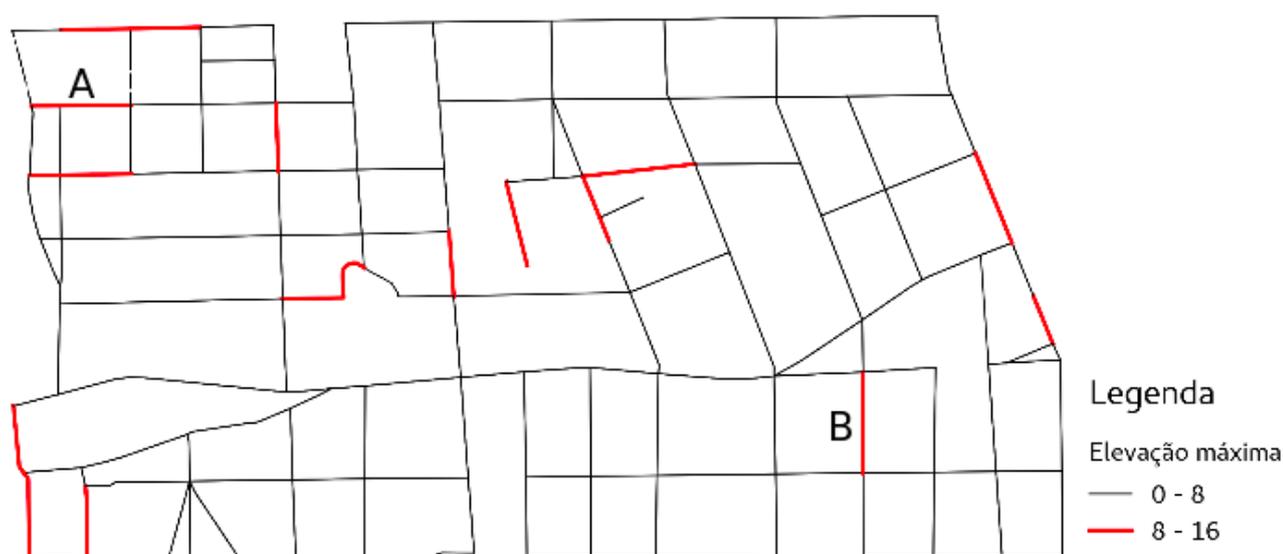
Figura 21 – Exemplo de transformação de dados de altimetria em custo de inclinação.



Baseado no padrão para cadeirantes, foi calculada a métrica de esforço máximo (Equação 4), que usa a distância e a elevação, para distinguir entre calçadas críticas (em vermelho) e acessíveis (em azul). Conforme especificado pelo padrão NBR9050 [(ABNT, 2015)], o grau percentual de inclinação (Equação 2) não deve ser maior que 8% para garantir a acessibilidade.

A Figura 22 mostra o esforço máximo para o Batel. De acordo com o valor de esforço máximo, o mapa mostra se uma rua é acessível ou não. Ou seja, ela deve apresentar um grau percentual de inclinação menor que 8% para que seja acessível do ponto de vista de inclinação. Ruas acessíveis são destacadas em azul enquanto os maiores valores de inclinação e, conseqüentemente, de menor acessibilidade são destacados em vermelho. Note que existem duas ruas destacadas com as letras A e B. Tais pontos são os que apresentam maior coeficiente de acessibilidade no bairro e podem ser vistos em detalhes nas Figuras 23 e 24.

Figura 22 – Máximo esforço nas ruas do bairro Batel.



4.3.3 Coeficiente de Acessibilidade

Para a realização dos teste aqui apresentados, considerou-se a região do Batel em Curitiba como objeto de estudo. Na Figura 25a o grafo espacial é apresentado com todas as arestas possíveis. Após o *feedback* do usuário, o grafo é atualizado, conforme mostra a Figura 25b. Os problemas são detalhados na Figura 26. Conforme pode-se notar na Figura 26, foram identificados os seguintes problemas: ausência de calçada (0), em obras (1), danificada (2), obstáculo (3), fora da especificação (4) e para rampas ausência de

Figura 23 – Local de maior inclinação no Batel (A).



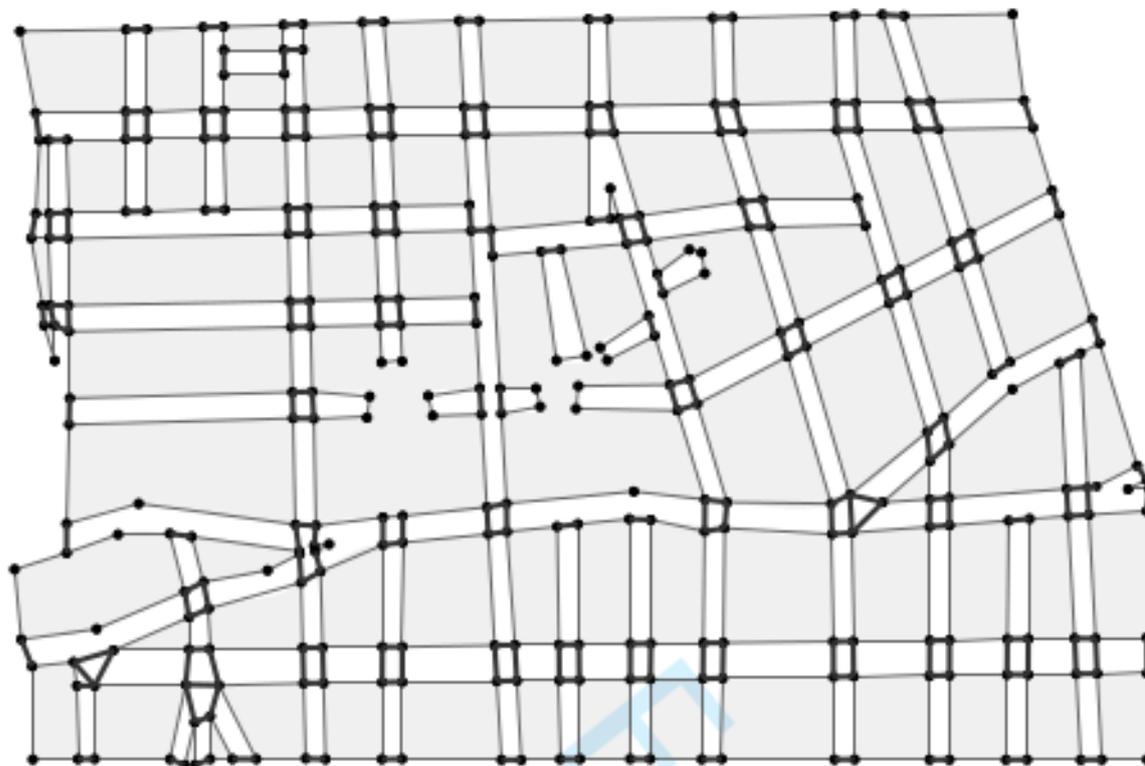
Figura 24 – Local de maior inclinação no Batel (B).



rampas (5 e 6), fora de especificação (7) e danificada (8 e 9).

Figura 25 – Infraestruturas de acessibilidade do Batel.

(a) Infraestrutura de calçadas e cruzamentos esperada.



(b) Infraestrutura identificada após *feedback* de usuários

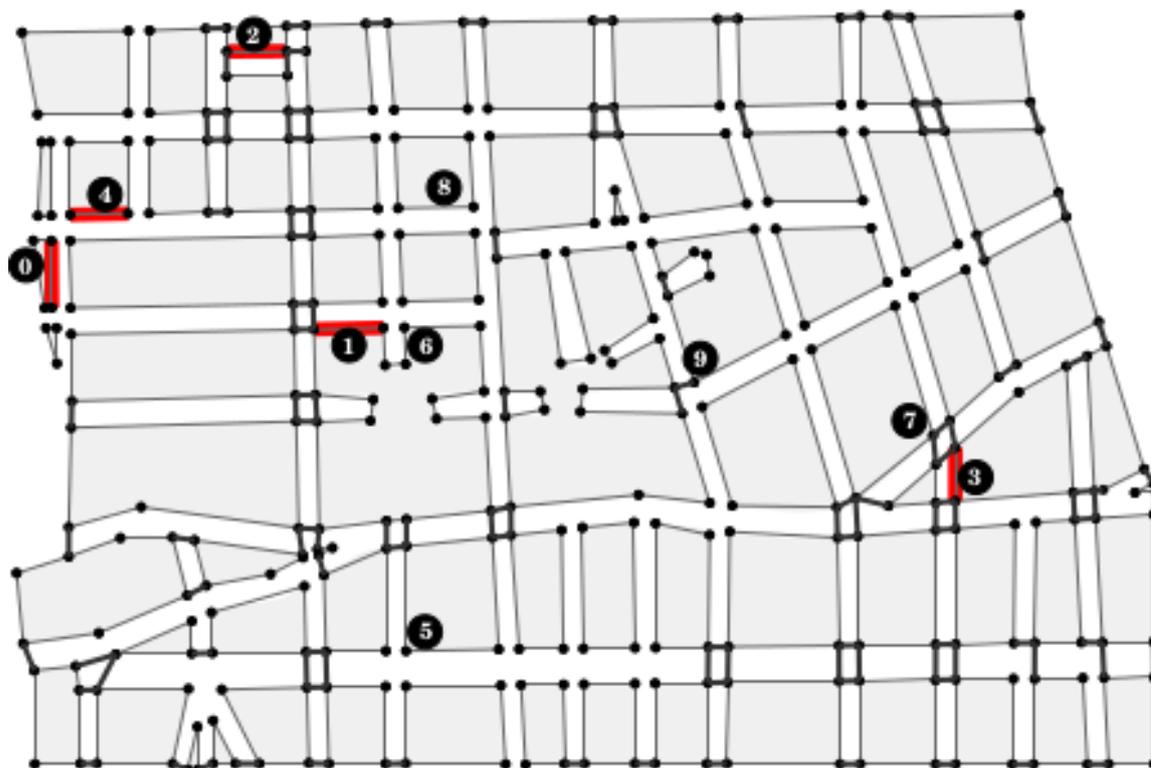


Figura 26 – Problemas encontrados nas calçadas.



4.3.4 Planejamento de rotas com acessibilidade

A Figura 27 mostra os menores caminhos para as métricas relacionadas à distância e inclinação para um par origem-destino na região do Batel. Considerando a métrica de distância (a), o planejador de rota gerou um caminho de 907,47 metros e um esforço máximo de 13,04%. Para inclinação (b), foi gerado um caminho de 1021,91 metros e um esforço máximo de 6,99%, que está sob o limite superior de 8% sugerido pelo NBR9050 [(ABNT, 2015)]. Tais resultados reforçam a importância de um planejador de rotas gerar caminhos de acordo com as necessidades de mobilidade dos usuários.

Foram usadas consultas dos algoritmos de Dijkstra e K-Dijkstra no pgRouting¹⁵ para encontrar menores rotas dentro de um contexto de dados geoespaciais. Resultados intermediários (como cruzamentos e métricas de esforço máximo) foram processadas por funções espaciais utilizando o PostGIS¹⁶. Posteriormente, estes caminhos foram visualizados no QGIS¹⁷, que permite a integração com o *Open Street Maps*.

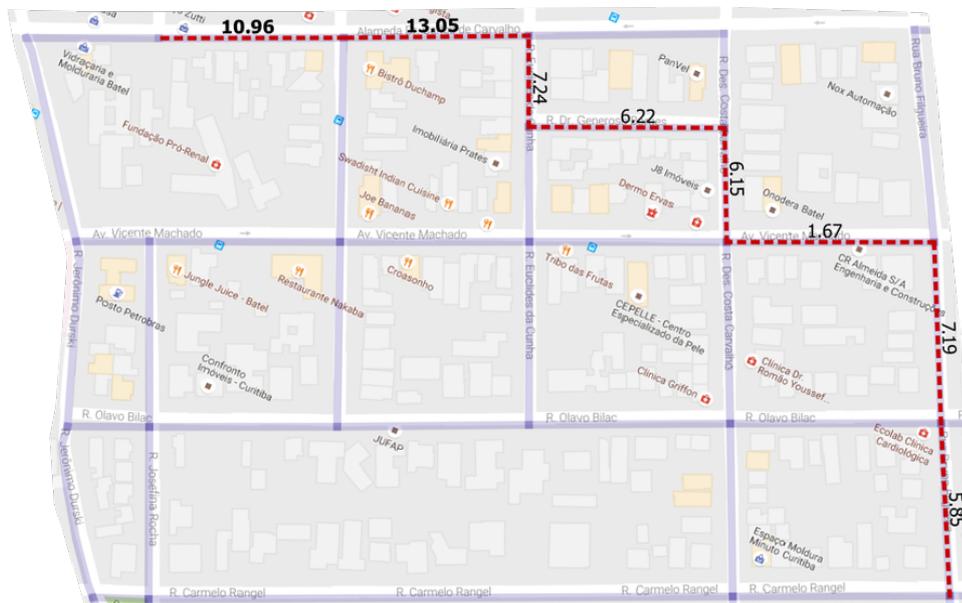
¹⁵ pgRouting - <https://pgrouting.org/>. Último acesso em 05/06/2018.

¹⁶ PostGIS - <https://postgis.net/>. Último acesso em 05/06/2018.

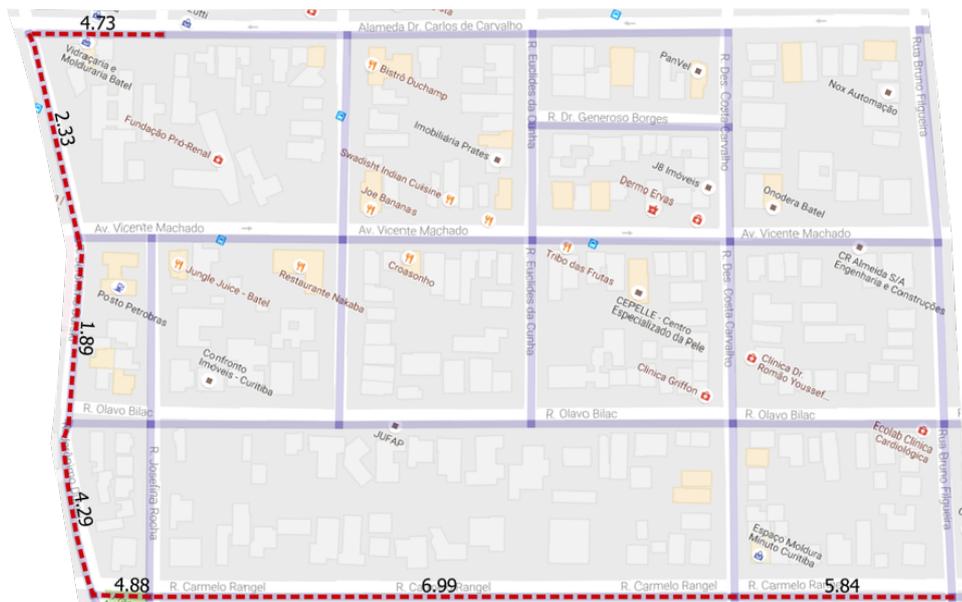
¹⁷ QGIS - <https://qgis.org/en/site/>. Último acesso em 05/06/2018.

Figura 27 – Caminhos mínimos.

(a) Menor caminho encontrado utilizando distância como métrica.



(b) Menor caminho resultante de uma execução de K-Dijkstra no pgRouting usando distância como métrica mas que minimiza a inclinação.

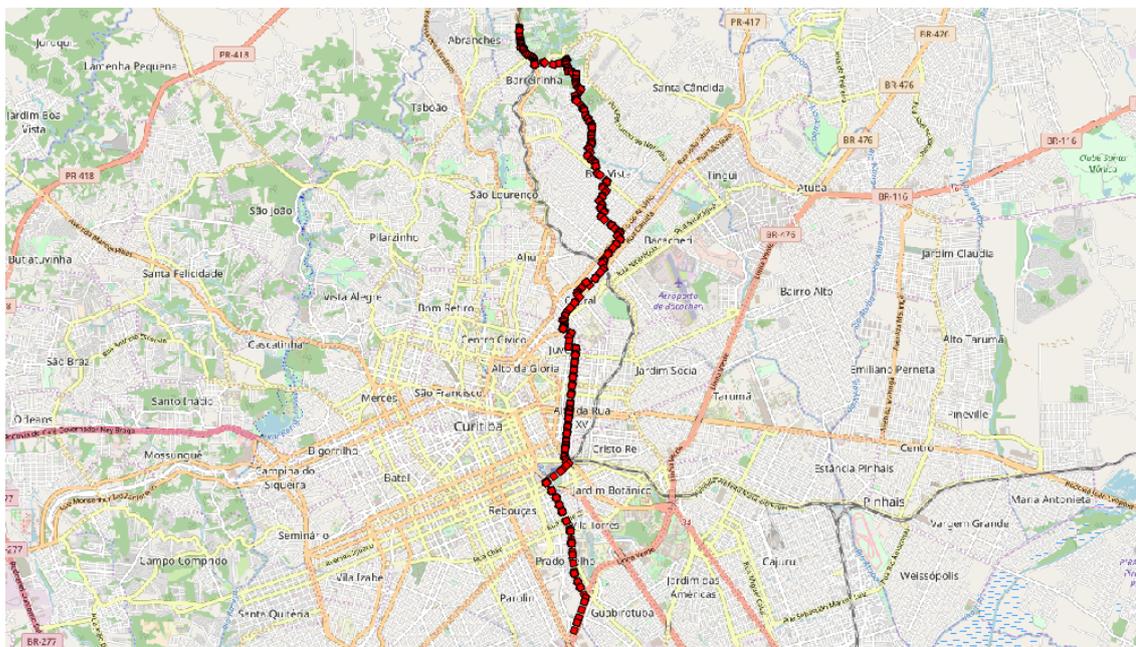


4.3.5 Caminho mínimo em calçadas para toda Curitiba

A fim de descobrir se o planejamento de rotas é escalável em um cenário maior que apenas um bairro, foi executado um teste considerando o contexto de toda a cidade de Curitiba (75 bairros). Foram selecionados dois pontos aleatórios em bairros diferentes e executado o Dijkstra utilizando-os como ponto de início e de destino. O resultado (usando um caminho entre o Abranches e Guabirotuba) é mostrado pela Figura 28. Executando

o algoritmo diretamente no banco de dados, a execução levou 1,8 segundos e gerou um caminho com um total de 14,98 quilômetros de distância.

Figura 28 – Resultado da execução do algoritmo de Dijkstra na implementação do modelo baseado em calçadas considerando toda a cidade de Curitiba.



Foram realizados testes para toda Curitiba utilizando a API implementada. Os pontos inicial e final ($\{-25.359034, -49.2633057\}$ e $\{-25.4663085, -49.2526824\}$) foram os mesmos utilizados nos testes executados diretamente pelo banco de dados. A execução, neste caso, levou 9,8 segundos com um total de 14,98 quilômetros de distância.

4.3.6 Comparação com *GoogleMaps*

Nesta seção são demonstrados testes realizados a fim de comparar as rotas resultantes da API com as rotas fornecidas pelo *GoogleMaps*. O intuito deste teste é encontrar as principais diferenças entre um modelo baseado em ruas com o modelo aqui desenvolvido (baseado em calçadas e que considera inclinação).

Selecionou-se três pares de pontos iniciais e finais e, para cada par, realizadas três execuções: i) pelo *GoogleMaps*, ii) pela API, considerando calçadas e distância e iii) pela API, considerando calçadas e inclinação.

Nas Figuras 29, 30 e 31 são apresentados testes com rotas entre dois pontos. As Figuras 29a, 30a e 31a mostra a execução sobre a implementação do modelo desenvolvido

neste trabalho. Nesta execução foi utilizado o algoritmo Dijkstra considerando como custo a distância de cada calçada. Similarmente, as Figuras 29b, 30b e 31b mostram execuções sobre tal implementação, no entanto, o custo considerado pelo algoritmo de Dijkstra foi o índice de inclinação de cada calçada. Por fim, as Figuras 29c, 30c e 31c apresentam rotas entre os mesmos pontos executada pelo *GoogleMaps*, que fornece rotas baseadas em ruas.

A distância da rota da Figura 29a apresentou uma distância de 950m e um custo de inclinação máximo de 13,4% e e mínimo de 2,7%, a distância da rota da Figura 29b apresentou uma distância de 1.500m com custo de inclinação máximo de 6,4% e mínimo de 1,8% e a distância da rota da Figura 29c foi de 1000m.

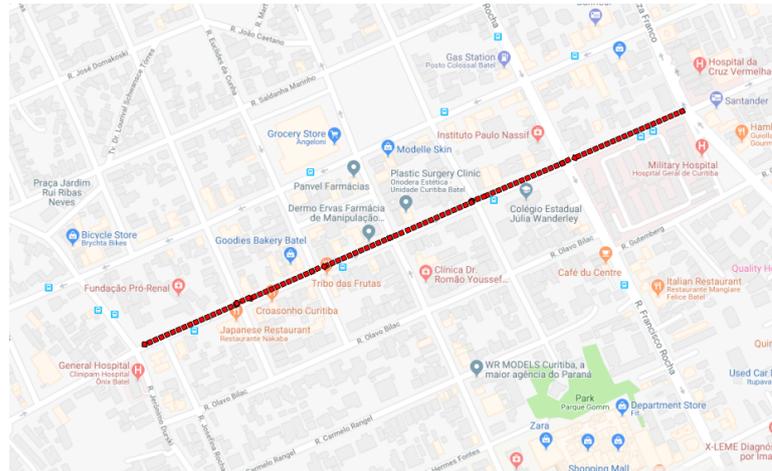
A distância da rota da Figura 30a apresentou uma distância de 950m e um custo de inclinação máximo de 27,8% e e mínimo de 3,1%, a distância da rota da Figura 30b apresentou uma distância de 1.300m com custo de inclinação máximo de 7,8% e mínimo de 4,3% e a distância da rota da Figura 30c foi de 950m.

As rotas nas Figuras 31a, 31b e 31c coincidiram apresentando uma distância de 900m e um custo de inclinação máximo de 8,8% e mínimo de 1,6%.

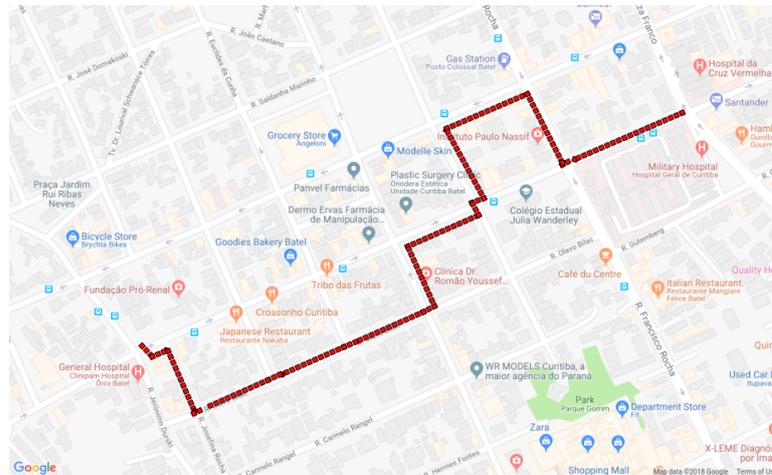
Note que as execuções considerando a inclinação podem fornecer rotas possivelmente mais extensas, contudo, tais rotas são mais acessíveis em termos de inclinação. As execuções considerando distância forneceram resultados semelhantes ao do *GoogleMaps*, contudo, as rotas retornadas foram traçadas considerando calçadas e cruzamentos.

Figura 29 – Testes comparativos com o *Google Maps* distância entre dois hospitais: do Hospital Ônix ao Hospital Geral de Curitiba.

(a) Execução pela API considerando calçadas e distância.



(b) Execução pela API considerando calçadas e inclinação.



(c) Execução pelo *GoogleMaps*.

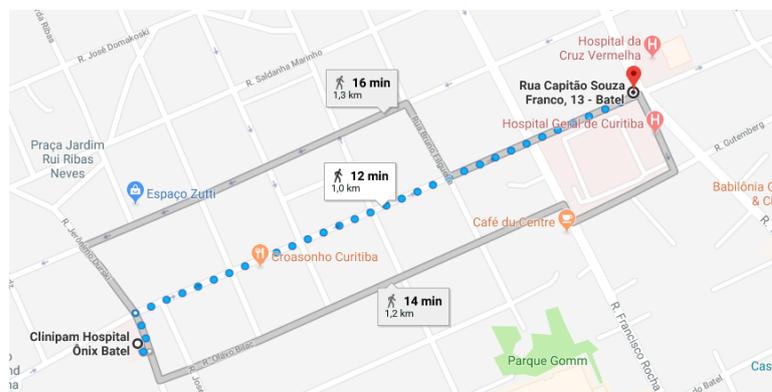
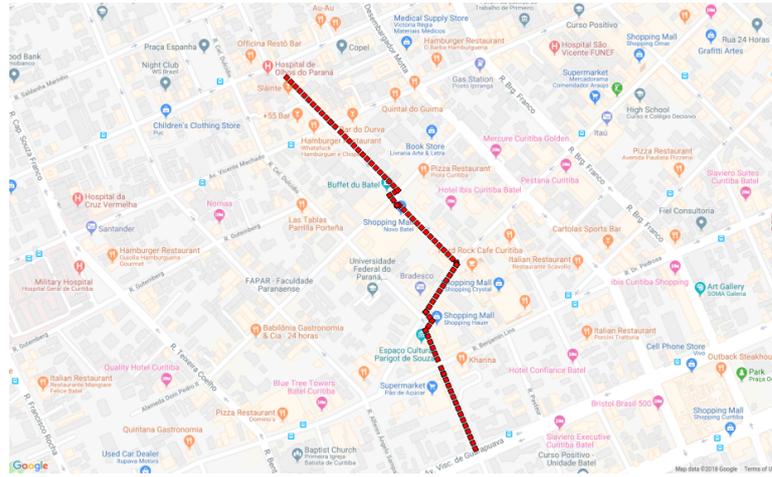
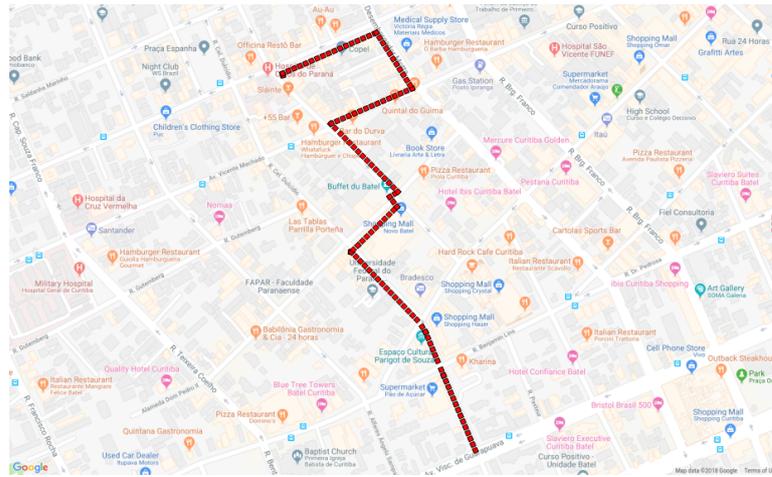


Figura 30 – Testes comparativos com o *Google Maps* distância entre dois pontos: da Alameda Presidente Taunay (próximo ao Hospital de Olhos) ao Instituto Paranaense de Cegos.

(a) Execução pela API considerando calçadas e distância.



(b) Execução pela API considerando calçadas e inclinação.



(c) Execução pelo *GoogleMaps*.

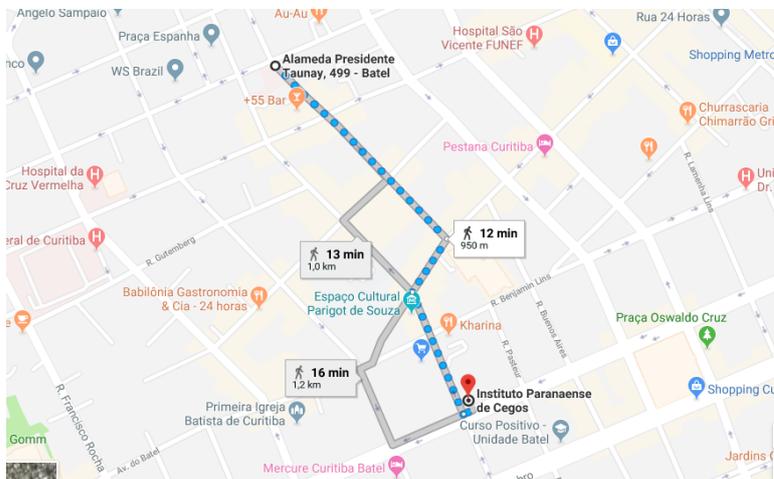
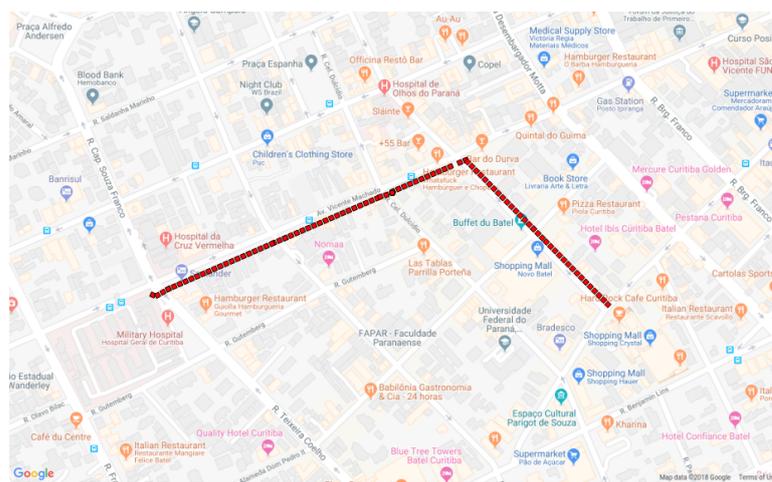
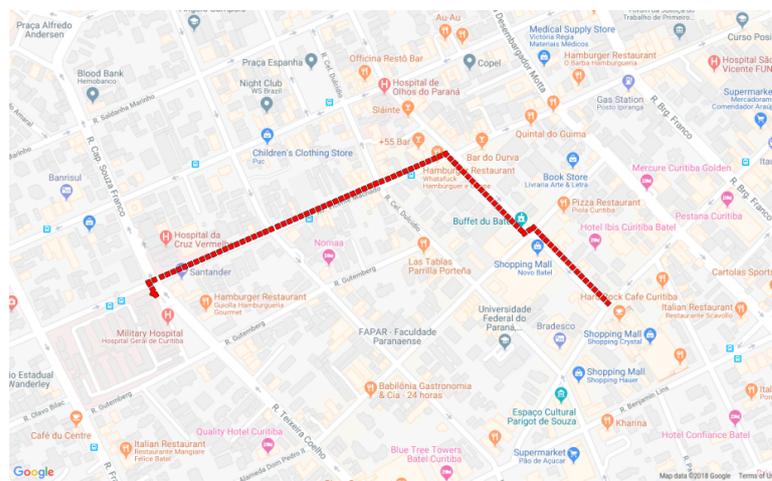


Figura 31 – Testes comparativos com o *Google Maps* distância entre dois pontos: do Hospital Geral de Curitiba ao Shopping Crystal.

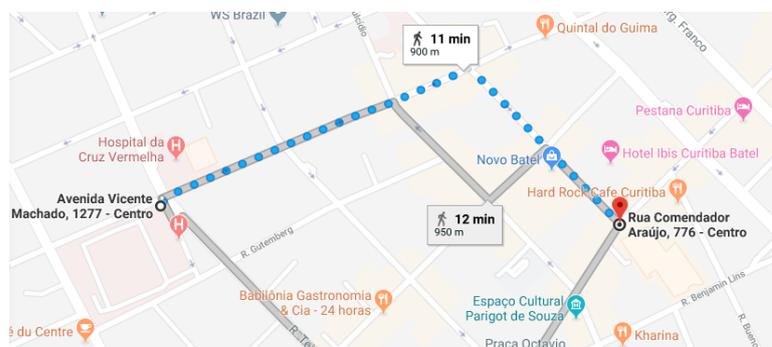
(a) Execução pela API considerando calçadas e distância.



(b) Execução pela API considerando calçadas e inclinação.



(c) Execução pelo *GoogleMaps*.



CAPÍTULO 5

CONCLUSÃO

Um planejamento de rotas é um problema desafiador que envolve o processamento de uma grande quantidade de dados - como mapas, imagens e topografia da cidade - além de uma grande quantidade de informações complexas oriundas de *feedbacks* de usuários e de agências municipais, a fim de definir corretamente caminhos confiáveis.

O objetivo desse trabalho foi utilizar a computação para proporcionar uma maior comodidade e qualidade de vida aos cadeirantes, tornando o seu deslocamento menos custoso. Para tanto, este trabalho apresentou um modelo teórico que contempla os principais pontos que interferem no deslocamento dos cadeirantes em um cenário urbano como rampas, inclinação, acessibilidade, fatores esses que trabalhos anteriores não apresentam (ver seção 2.4).

Os objetivos específicos alcançados por este trabalho foram, portanto:

- A realização de uma revisão bibliográfica nos seguintes conceitos: SIG, algoritmos e cidades inteligentes.
- Foi realizada uma pesquisa explorando os principais trabalhos que elaboraram modelos semelhantes ao apresentado aqui.
- Foi elaborado um novo modelo teórico utilizando a arquitetura proposta por (ZHENG et al., 2014) pois nenhum dos trabalhos relacionados pesquisados na literatura apresentava um modelo que fosse baseado em ruas e que considerasse a inclinação de um caminhos e fatores de acessibilidade como obstáculos e situação das calçadas (ver seção 2.4).
- Utilizando dados geográficos e um banco de dados geoespacial, foi possível, então, a partir desse modelo, implementar um sistema que atendesse a todos esses aspectos essenciais para um cadeirante (ver seção 4.2).

- O modelo foi implementado e disponibilizado para uso através de uma API, conforme mostra a seção 4.2.4.
- Os testes foram realizados na seção 4.3 e validaram a implementação deste modelo.

Derivado dessa dissertação, foram publicados dois trabalhos: o primeiro ocorreu na conferência Geoinfo 2017, *shortpaper* (BARCZYSZYN et al., 2017). O segundo consiste em um Journal publicado pela ACM Transactions on Accessible Computing, (BARCZYSZYN et al., 2018).

As principais dificuldades aqui encontradas foram a manipulação das funções corretas para fazer a transformação de dados principalmente para a geração correta de arestas de cruzamentos. A obtenção dos pontos de cruzamento entre aresta de calçada e curvas de altimetria. A posterior aplicação de todas as transformações para toda a cidade de Curitiba (inicialmente, foram realizadas apenas para o bairro Batel) foi uma das grandes dificuldades por envolver uma enorme quantidade de registros.

Como trabalhos futuros, pode-se citar:

- A inclusão de pontos de interesses a partir da utilização de outras camadas de dados apresentadas em (BARCZYSZYN, 2015) (como ciclovias, pontos de ônibus, entre outros);
- O complemento do modelo com a utilização de outras informações como dados de trânsito para identificar cruzamentos perigosos.
- A integração com *crowdsourcing* e a implementação de uma aplicação que forneça uma interface para interação do usuário também são possíveis trabalhos futuros.
- A utilização de dados da camada de transporte público para incluir no modelo, por exemplo, pontos de ônibus.
- Considerar diferentes perfis ao gerar as rotas: além de cadeirantes, contemplar pedestres com malas, carrinho de bebê.
- Cálculo de um índice de qualidade que reflète a diferença da rota real e a rota ideal.

REFERÊNCIAS

- ABNT. Accessibility to buildings, equipment and the urban environment.technical report. Brazilian Technical Standards Association., 2015.
- ALAWADHI, S. et al. Smart 172 cities and service integration initiatives in north american cities: A status report. **ACM New York**, 2012.
- BARCZYSZYN, G. L. Integração de dados geográficos para planejamento urbano da cidade de Curitiba. Universidade Tecnológica Federal do Paraná. Trabalho de Conclusão de Curso. 2015.
- BARCZYSZYN, G. L. et al. A collaborative system for suitable wheelchair route planning. **Transactions on Accessible Computing**, p. 92–105, 2018.
- BARCZYSZYN, G. L. et al. Utilização de dados de altimetria para o fornecimento de rotas acessíveis para cadeirantes. **Geoinfo**, p. 104–109, November 2017.
- BEALE, L. et al. Magus: Modelling access with gis in urban systems: An application for wheelchair users in northamptonshire. **6TH ERCIM Workshop "User Interfaces for All"**, 2006.
- BENTLEY, J. L.; OTTMANN, T. A. Algorithms for reporting and counting geometric intersections. **IEEE Trans. Comput.**, p. 643–647, 1979.
- CAMENAR, L. M. O.; NASCIMENTO, D. F.; ALMEIDA, L. D. A. A method for analyzing mobility issues for people with physical disabilities in the context of developing countries. In: **HCI International, 2018, Las Vegas. Lecture Notes in Computer Science - Universal Access in Human-Computer Interaction. Methods, Technologies, and Users. Switzerland: Springer International Publishing AG, 2018.**
- CHAVEZ, A.; CALTENCO, H.; BERAN, V. Comparison of different powered-wheelchair control modes for individuals with severe motor impairments. **IEEE Int. Conf. on Informatics in Control, Automation and Robotics**, p. 353–359, 2014.
- CHEN, C. et al. Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. **IEEE Transactions on Intelligent Transportation Systems**, p. 1259–1273, 2015.
- CLEMENTINI, E.; FELICE, P. D.; OOSTEROM, P. A. van. A small set of formal topological relationships suitable for end-user interaction. **International Symposium on Advances in Spatial Databases**, p. 277–295, 1993.
- CORMEN, T. H.; STEIN, C.; RIVEST, C. E. Leiserson R. L. Introduction to algorithms. McGraw-Hill Higher Education, p. 246–253, 2001.

- CTECNO. **Hoja de Ruta para la Smart City**. 2012. Acessado em 27/06/2018. Disponível em: <http://www.ctecno.cat/wp-content/uploads/2012/03/Hoja-de-Ruta-Smart-Cities_def.pdf>.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische Mathematik**, p. 269–271, 1959.
- EGENHOFER, M. J. Query processing in spatial-query-by-sketch. **Journal of Visual Languages Computing**, 1997.
- ESRI. **White Paper - Gis for Smart Cities**. 2014. Acessado em 27/06/2018. Disponível em: <<http://www.esriindia.com/industries/government/~media/esri-%20india/files/pdfs/indus%20tries/white-paper-gis-for-smart-cities>>.
- FRINTROP, S.; ROME, E.; CHRISTENSEN, H. I. Computational visual attention systems and their cognitive foundations: A survey. **ACM Transactions on Applied Perception**, p. 1–6, 2010.
- GRUDIN, J. Groupware and social dynamics: Eight challenges for developers. **Commun. ACM**, vol. **37**, no. **1**, p. 92–105, 1994.
- HAKLAY, M.; WEBER, P. Openstreetmap: User-generated street maps. **IEEE Pervasive Computing** **7**, p. 12–18, 2008.
- HOLONE, H.; MISUND, G. People helping computers helping people: Navigation for people with mobility problems by sharing accessibility annotations. **ICCHP 2008**, p. 11–14, 2007.
- IBM. **IBM, um planeta mais inteligente**. 2015. Acessado em 27/06/2018. Disponível em: <http://www.ibm.com/smarterplanet/es/es/smarter_cities/overview/>.
- IPPUC. Relatório descritivo das atividades de levantamento de informações e montagem do banco de dados. 2013.
- KASEMSUPPAKORN, P.; KARIMI, H. A. Data requirements and a spatial database for personalized wheelchair navigation. **Singapore Therapeutic, Assistive Rehabilitative Technologies (START) Centre**, p. 31–34, 2008.
- KASEMSUPPAKORN, P.; KARIMI, H. A. Personalised routing for wheelchair navigation. **J. Locat. Based Serv**, p. 24–54, 2009.
- KOZIEVITCH, N. P. et al. An Alternative and smarter route planner for wheelchair users - Exploring Open Data. 5th International Conference on Smart Cities and Green ICT Systems. Rome. Proceedings of the 5th International Conference on Smart Cities and Green ICT Systems, p. 94–99, 2016.
- LI, L. T.; TORRES, R. S. Digital libraries applications: Cbir, education, social networks, escience/simulation, and gis. in capítulo 5. Morgan and Claypool Publishers, p. 63–117, 2014.
- LOYOLA, G. F. M.; MEIRA, M. T. Proposta de aplicação de visualização de dados focada em planejamento urbano acessível. Trabalho de Conclusão de Curso. 2018.

- MENKENS, C. et al. Easywheel - a mobile social navigation and support system for wheelchair users. **In: ITNG '11**, p. 859–866, 2011.
- MEYERSON, A.; TAGIKU, B. Minimizing average shortest path distances via shortcut edge addition. **Int. Workshop on Algorithms and Techniques for Approximation, Randomization, and Combinatorial Optimization**, p. 272–285, 2009.
- MINETTO, R. et al. Shortcut Suggestion based on Collaborative User Feedback for Suitable Wheelchair Route Planning. 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), p. 2372–2377, 2016.
- MOURA, A. C. M. Learning topics in urban planning at ufmg: Geoprocessing to support analysis, planning and proposal of the urban landscape at neighborhood scale. **5th International Seminar on Environmental Planning and Management: Urban Responses for Climate Change**, p. 15, 2012.
- MÁRQUEZ, F. G. Curso básico de topografía: planimetría, agrimensura, altimetría. Editorial Pax México., p. 1–3, 2003.
- NEIS, P. Measuring the reliability of wheelchair user route planning based on volunteered geographic information. **Transactions in GIS**, p. 188–201, 2015.
- PAPAGELIS, M.; BONCHI, F.; GIONIS, A. Suggesting ghost edges for a smaller world. **ACM Int. Conf. on Information and Knowledge Management (CIKM)**, p. 2305–2308, 2011.
- PAROTSIDIS, N.; PINTOURA, E.; TSAPARAS, P. Selecting shortcuts for a smaller world. **SIAM Int. Conf. on Data Mining**, p. 28–26, 2015.
- RUSSEL, S.; NORVIG, P. Artificial intelligence. Prentice Hall, p. 98–101, 1994.
- SOBEK, A.; MILLER, H. U-access: a web-based system for routing pedestrians of differing abilities. **Journal of Geographical Systems**, p. 269–287, 2006.
- SUMIDA, Y. et al. Development of a route finding system for manual wheelchair users based on actual measurement data. **IEEE Int. Conference on Ubiquitous Intelligence and Computing**, p. 17–23, 2012.
- VISWANATHAN, P. et al. Navigation and obstacle avoidance help (noah) for older adults with cognitive impairment: A pilot study. **ACM SIGACCESS Conference on Computers and Accessibility**, p. 43–50, 2011.
- VÖLKEL, T.; WEBER, G. Routecheckr: Personalized multicriteria routing for mobility impaired pedestrians. **Springer Netherlands**, p. 233–241, 2008.
- WANG, T. Interdisciplinary urban gis for smart cities: advancements and opportunities. **Geo-spatial Information Science**, p. 25–34, 2013.
- WATTANAVARANGKUL, N.; WAKAHARA, T. Indoor navigation system for wheelchair using smartphones. in information technology convergence: Security, robotics, automations and communication. **Springer Netherlands**, p. 233–241, 2013.
- YEN, J. Y. An algorithm for finding shortest routes from all source nodes to a given destination in general networks. **Quart. Appl. Math.** **27**, p. 526–530, 1970.

ZHENG, Y. et al. Urban computing: Concepts, methodologies, and applications. **ACM Transactions on Intelligent Systems and Technology**, 2014.

ZIEGELMANN, M. Constrained shortest paths and related problems - constrained network optimization. **Verlag**, 2007.