

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**RENAN RISSÁ FRANCO
VINICIUS DE ANDRADE SAUTER**

**MODELAGEM E DESENVOLVIMENTO DE UM SISTEMA PARA
FACILITAR A NAVEGAÇÃO EM AMBIENTES DE MÉDIO PORTE**

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2014

RENAN RISSÁ FRANCO
VINICIUS DE ANDRADE SAUTER

**MODELAGEM E DESENVOLVIMENTO DE UM SISTEMA PARA
FACILITAR A NAVEGAÇÃO EM AMBIENTES DE MÉDIO PORTE**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação, do Departamento de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Daniel Costa de Paiva

PONTA GROSSA

2014



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Ponta Grossa
Nome da Diretoria
Nome da Coordenação
Nome do Curso



TERMO DE APROVAÇÃO

MODELAGEM E DESENVOLVIMENTO DE UM SISTEMA PARA FACILITAR A NAVEGAÇÃO EM AMBIENTES DE MÉDIO PORTE

por

RENAN RISSÁ FRANCO
VINICIUS DE ANDRADE SAUTER

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 22 de Julho de 2014 como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação. Os candidatos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Daniel Costa de Paiva
Prof. Orientador

Geraldo Ranthum
Membro titular

Willian Massami Watanabe
Membro titular

Ionildo José Sanches
Responsável pelos Trabalhos
de Conclusão de Curso

Gleifer Vaz Alves
Coordenador do Curso
UTFPR - Campus Ponta Grossa

À minha família, amigos e professores,
pelo incentivo e pelo apoio constante.

AGRADECIMENTOS

Primeiramente a Deus por ter me dado saúde e força para superar todas as nossas dificuldades.

Aos nossos pais, pelo exemplo, amor, apoio e incentivo, fundamentais na construção do nosso caráter.

A esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

Ao orientador Daniel Costa de Paiva, pelo suporte no pouco tempo que lhe coube, pelas suas correções e incentivos.

Agradeço a todos os professores por proporcionar o conhecimento não apenas racional, mas a manifestação de caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a nos alunos, não somente por terem ensinado, mas por terem nos feito aprender.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

A persistência é o caminho do êxito.
(Charles Chaplin)

RESUMO

FRANCO, Renan Rissá; SAUTER, Vinicius de Andrade. **Modelagem e Desenvolvimento de um sistema para facilitar a navegação em ambientes de médio porte.** 2014. 24 f. Trabalho de Conclusão de Curso Bacharelado Ciência da Computação - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

O desenvolvimento de maneiras para auxiliar as pessoas na navegação em ambientes torna-se bastante útil em lugares que há um grande volume de usuários, ajudá-los a se localizar e chegar a seus objetivos de maneira fácil e rápida, fornecendo uma economia em seu tempo e dinheiro, proporcionando-lhe informações sobre os locais e direções, deste modo, visando o desenvolvimento de um sistema inteligente adequado para ambientes com grande fluxo de usuários, este trabalho fornece uma solução de aplicativos para a criação e navegação em mapas elaborados a partir de plantas baixas. Os mapas virtuais criados podem ser disponibilizados aos usuários de duas maneiras sem a dependência de uma conexão externa (internet, GPS, etc.) para utilizar o sistema. Além disto, os usuários podem utilizar computadores presentes em lugares estratégicos do estabelecimento. Há ainda uma aplicação móvel disponível com um sistema operacional Android onde é possível a consulta off-line em qualquer lugar e momento.

Palavras-chave: Inteligência Artificial, Arquiteturas de Controle de Agentes, Agentes Autônomos, Agentes Inteligentes, Algoritmos de busca de caminhos.

ABSTRACT

FRANCO, Renan Rissá; SAUTER, Vinicius de Andrade. **Modeling and Development of a system to facilitate navigation in midsize environments.**

2014. 24 f. Trabalho de Conclusão de Curso Bacharelado Ciência da Computação - Federal Technology University - Parana. Ponta Grossa, 2014.

The development of ways to assist people in navigation in environments becomes quite useful in places that there is a large volume of users, helping them to find and reach your goals quickly and easily, providing an economy in its time and money by providing you with information about locations and directions, thus aiming at developing an intelligent system suitable for environments with large influx of users, this paper provides a solution for creating applications and navigation maps made from floorplans. The created virtual maps may be available to users in two ways without depending on an external connection (Internet, GPS, etc.) To use the system. In addition, users can use computers present in strategic locations of the property. There is even a mobile application available with an Android operating system where it is possible to query offline anywhere and time.

Keywords: Artificial Intelligence Architectures Control Agents, Autonomous Agents, Intelligent Agents, Algorithms Path Finding.

LISTA DE ILUSTRAÇÕES

Figura 1. Minicursos SAEQ	11
Figura 2. Mapa SAEQ	12
Figura 3. Planta Baixa - Shopping Interlagos São Paulo/SP	19
Figura 4. Interação com ambientes através de sensores e atuadores	20
Figura 5. Pontes de Königsberg	22
Figura 6. Exemplo de grafo infinito e finito	23
Figura 7. Exemplo de grafo com múltiplos ramos	24
Figura 8. Diagrama de Visibilidade.....	26
Figura 9. Campos Potenciais	27
Figura 10. Provável caminho escolhido.....	27
Figura 11. Decomposição em células exatas	28
Figura 12. Decomposição em células Aproximadas.....	28
Figura 13. Agente na origem e o ponto de interesse.....	37
Figura 14. Possível ação do agente em azul, em vermelho obstáculos encontrados	37
Figura 15. Agente ao decidir a próxima casa de destino.....	38
Figura 16. Matriz com pontos de interesse	39
Figura 17. Agentes encontrando pontos de interesse	40
Figura 18. Agente se movendo	40
Figura 19. Função "andaSegunda"	41
Figura 20. Buscando o menor caminho.....	41
Figura 21. Mapa inicial	42
Figura 22. Criação do agente e definição dos pontos de origem e destino	43
Figura 23. Análise de possíveis casas de destino.....	43
Figura 24. Escolha do próximo passo do agente	44
Figura 25. Caminho gerado pelo agente no mapa até chegarem ao destino	44
Figura 26. Campus UTFPR Ponta Grossa	48
Figura 27. Imagem final bloco C.....	49
Figura 30. Inserindo locais caminháveis	51
Figura 31. Locais caminháveis inseridos.....	52
Figura 32. Inserindo ponto de interesse	53
Figura 33. Ponto de interesse inserido.....	53
Figura 34. Pontos de interesse inseridos	54
Figura 35. Ícone salvar.....	54
Figura 36. Mensagem de Bem Vindo	55
Figura 40. Selecionando saída e entrada.....	57
Figura 41. Exemplo	58
Figura 42. Entrada de dados para aplicação mobile	59

Figura 43. Lista de plantas baixas para pesquisa de caminhos	60
Figura 44. Tela para seleção de origem e destino	60
Figura 45. Tela de Exibição do passo a passo até o destino.	61
Figura 46. Tela de Exibição do caminho encontrado	61

SUMÁRIO

1 INTRODUÇÃO	10
1.1 MOTIVAÇÃO	11
1.2 SOLUÇÃO PROPOSTA.....	12
1.3 DESAFIOS.....	14
1.3.1 Metodologia para representação de decomposição de Ambientes.....	14
1.3.2 Metodologia para o planejamento de caminhos e desvio de obstáculos	14
1.3.3 Criação de mapas virtuais para navegação com base em plantas baixas.....	14
1.3.4 Navegação em mapas virtuais.....	14
1.4 OBJETIVOS.....	15
1.4.1 Objetivo Geral.....	15
1.4.2 Objetivos Específicos.....	15
1.5 METODOLOGIA	15
2 POR QUE SISTEMA OFF-LINE?.....	17
3 TERMOS E ÁREAS TÉCNICAS	19
3.1 PLANTAS BAIXAS.....	19
3.2 AGENTES INTELIGENTES	20
3.2.1 Características de um agente	20
3.2.2 Arquiteturas de Agentes	21
3.3 INTRODUÇÃO À TEORIA DOS GRAFOS	22
3.3.1 Principais Conceitos	23
3.4 AMBIENTES E PLANEJAMENTO DE CAMINHOS	24
3.4.1 Representação e Decomposição de Ambientes	25
3.4.1.1 Mapas de rotas – <i>Roadmaps</i>	25
3.4.1.2 Campo de Potencial.....	26
3.4.1.3 Decomposição de células	27
3.4.2 Algoritmo de Dijkstra	28
3.4.3 Algoritmo B*	29
4 TRABALHOS RELACIONADOS	32
4.1 SHOPPING TAMBORÉ	33
5 PRIMEIRA VERSÃO DAS APLICAÇÕES	34
5.1 SISTEMA PARA LOCALIZAÇÃO DE PONTOS E CAMINHOS.....	34
5.1.1 Metodologia	34
5.1.2 Desenvolvimento	35
5.2 SISTEMA PARA LOCALIZAÇÃO DE UM PONTO DESVIANDO DE OBSTÁCULOS.....	36
5.2.1 Desenvolvimento	36
6 SEGUNDA VERSÃO DAS APLICAÇÕES.....	39
6.1 RESULTADO PARCIAL - SISTEMA PARA LOCALIZAÇÃO DE PONTOS E CAMINHOS	39

6.2 RESULTADO PARCIAL - SISTEMA PARA LOCALIZAÇÃO DE UM PONTO DESVIANDO DE OBSTÁCULOS.....	42
7 VERSÃO CONSOLIDADA DAS APLICAÇÕES	45
7.1 AMBIENTE DE DESENVOLVIMENTO	45
7.1.1 Aplicação <i>Desktop</i>	45
7.1.2 Aplicação <i>Mobile</i>	46
7.1.2.1 Eclipse IDE	46
7.1.2.2 Ambiente <i>Android</i>	46
7.2 ESTUDO DE CASO	47
7.3 APLICAÇÕES <i>DESKTOP</i>	49
7.3.1 Criando um novo projeto – Usuário Administrador	49
7.3.2 Utilizando O Sistema De Navegação	54
7.4 APLICAÇÃO <i>MOBILE</i>	58
7.4.1 Executando o Aplicativo.....	59
8 CONCLUSÃO.....	63
8.1 TRABALHOS FUTUROS	64
REFERÊNCIAS.....	65
APÊNDICE A - Arquivo gerado com os dados do Mapa Virtual pela aplicação <i>desktop</i>: BLOCO C1.xml.....	69

1 INTRODUÇÃO

A navegação em ambientes de médio porte, tais como Shopping Centers e Universidades, é muitas das vezes problema para o qual não se tem uma solução eficiente. Encontrar determinado local nestes ambientes é uma tarefa custosa que pode levar tempo, já que nos mesmos existem diferentes pisos e muitas das vezes não há placas ou uma sinalização adequada. É comum os visitantes que não conhecem detalhadamente os ambientes se perderem e gastarem muito tempo para encontrar o local que procuram e precisam.

Os dias atuais exigem cada vez mais tempo das pessoas para realizarem suas atividades, e essa exigência faz com que seja necessário haver um melhor gerenciamento desse tempo utilizado para executar tais atividades (Zanini, 2010). Afinal, com a vida moderna e as atividades cada vez exigindo mais tempo e pontualidade, o tempo se torna um fator raro, assim sua administração é imprescindível (TAMASHIRO, 2003). O tempo deve ser utilizado estrategicamente e desperdiçado o mínimo possível.

Com o crescimento das cidades, os ambientes fechados vêm crescendo cada vez mais, como acontecem com os shoppings center, instituições públicas e privadas. 43% dos shopping centers são considerados de médio porte e tem um fluxo médio diário de aproximadamente 21700 pessoas (IBOPE, 2011).

Em função da falta de sinalização adequada, cada vez mais pessoas perdem tempo para realizarem atividades cotidianas como encontrar uma loja em um shopping ou um departamento em uma Instituição de ensino. Tomando como base este cenário, neste trabalho se buscam soluções de forma rápida e de fácil acesso para disponibilizar as informações úteis aos usuários, seja nos seus dispositivos móveis ou em estações instaladas em locais específicos.

Uma solução hoje adotada em muitos shoppings center como o Shopping Tamboré e Shopping Jardim Sul, localizados respectivamente nas cidades de Barueri e São Paulo no estado de São Paulo, é a instalação de mapas em locais estratégicos para que o visitante consiga se localizar e decidir para onde ele deve seguir para chegar no seu local de destino.

Desta forma, o usuário pode encontrar e obter orientações e informações sobre trajetos para o deslocamento, ajudando-o a chegar ao seu destino, lugares específicos (salas, departamentos, lojas, dentre outros) do estabelecimento. Para o

sistema precisa ser pré-configurado, necessitando apenas da planta do local com informações definidas da localização de cada ponto importante no ambiente.

1.1 MOTIVAÇÃO

Como já dito, obter a localização e um possível caminho em ambientes fechados é um problema, um exemplo que fica evidente a dificuldade de se obter uma localização mesmo sendo pessoas que tenham certa facilidade com placas foi a Semana Acadêmica de Engenharia Química (SAEQ) realizada em 2013 no campus de Ponta Grossa da Universidade Tecnológica Federal do Paraná. Na entrada de um dos blocos do campus haviam cartazes com o nome dos minicursos e as salas e também cartazes com flechas que deveriam ajudar na localização do Centro de Convivência, bloco L, bloco Q e bloco R. Para quem não tinha conhecimento prévio do ambiente, certamente houve grande dificuldade para localizar os blocos mencionados com a indicação utilizada.



Figura 1. Minicursos SAEQ

Na Figura 1 vemos que são indicadas as salas que foram realizados os minicursos porém não há uma indicação de onde ficam as salas.



Figura 2. Mapa SAEQ

Como ilustrado na Figura 2, como tentativa de ajudar na localização foram usadas setas para indicar os blocos onde as salas ficam. Se fossem realmente seguir as setas todas elas dariam diretamente em paredes já que elas não tinham nenhuma precisão, afinal a localização do Bloco Q ou do Centro de Convivências não é da forma como estava mostrado em frente ao Auditório principal da universidade.

Desta forma, fica evidente que um sistema de localização em ambientes fechado seriam uma ferramenta de grande auxílio na localização, ajudando achar locais com rapidez evitando o desperdício de tempo e eventuais atrasos a compromissos ou minicursos como no caso citado.

1.2 SOLUÇÃO PROPOSTA

A solução proposta é, portanto, um sistema que tem três aplicações, duas *desktop* e uma *mobile*, de forma que as configurações sejam geradas em uma aplicação *desktop* e, após carregadas, podem ser consultadas em qualquer das opções.

A aplicação *desktop* base deve ser utilizada por um especialista que carrega a planta baixa do ambiente escolhido define os dados básicos e pontos estratégicos que posteriormente serão utilizados nas duas demais aplicações.

Uma aplicação móvel e duas *desktop* serão utilizadas para auxiliar os usuários no deslocamento e na localização, mostrando o caminho que deve ser seguido a partir de um ponto e do seu destino (informados por ele). As aplicações devem ser similares e apenas adaptadas aos dispositivos para atender não apenas aos usuários que possuem dispositivos Android, mas a todos os frequentadores do ambiente. Requisitos básicos para as aplicações são a não dependência de conexão com internet ou GPS (*Global Positioning System*). Para isto os dados devem ser previamente carregados e o mínimo de demanda local requerida após o usuário informar origem, destino e escolher a planta do ambiente onde se encontra.

As aplicações devem ser o mais genérico possível, dependendo apenas da planta e das marcações dos locais existentes no ambiente, ou seja, não serão desenvolvidas para um local específico. Visando ampliar a utilização a ferramenta *desktop* pretende-se que a mesma funcione também em versão Web, sendo útil inclusive para aqueles que ainda não estejam no ambiente ou que preferem acessar a aplicação deste modo, planejando antecipadamente os caminhos e atividades.

Para o desenvolvimento das aplicações *desktop* será utilizada a linguagem C#¹, que tem grande variedade de funções já que é uma linguagem considerada robusta (Microsoft, 2010). Outro motivo para a escolha desta linguagem é que quando se quer converter uma aplicação *desktop* em C# para uma aplicação Web, temos a facilidade pois a linguagem para Web .NET é compatível com o C# então toda a regra de negócio que será implementada poderá reutilizada quando da conversão da aplicação para a Web (DEITEL, 2004).

Para o desenvolvimento da aplicação móvel será utilizada a plataforma Android (GOOGLE, 2013) por esta permitir desenvolver e integrar aplicações de forma

¹ C# (pronuncia-se "C sharp") é uma linguagem de programação orientada a objetos, fortemente tipada, desenvolvida pela Microsoft para o desenvolvimento de uma variedade de aplicações que executam sobre o .NET Framework. (MICROSOFT, 2012)

simplificada utilizando Java como linguagem de programação e XML (*eXtensible Markup Language*).

“O Android é a plataforma *open-source* criada pelo Google para o desenvolvimento de aplicações para dispositivos móveis. É a nova sensação do momento e promete revolucionar o desenvolvimento de aplicações. Inclui um sistema operacional baseado no Linux e diversas aplicações, com uma rica interface gráfica, um browser para navegar na internet, integração com o Google Maps, suporte a multimídia, GPS, banco de dados integrado, jogos em 3D e muito mais.” (LECHETA, 2013).

1.3 DESAFIOS

1.3.1 Metodologia para representação de decomposição de Ambientes

Procurar por métodos para representar e decompor plantas baixas em Mapas Virtuais navegáveis.

1.3.2 Metodologia para o planejamento de caminhos e desvio de obstáculos

Procurar por métodos de planejamento de caminhos para serem aplicados nos Mapas Virtuais criados.

1.3.3 Criação de mapas virtuais para navegação com base em plantas baixas

Desenvolver uma aplicação para criação de Mapas Virtuais baseados em plantas baixas dos ambientes.

1.3.4 Navegação em mapas virtuais

Desenvolver mais 2 aplicações, utilizando os dados dos Mapas Virtuais criados, para realização da busca e geração de caminhos no ambiente.

1.4 OBJETIVOS

Os objetivos do trabalho estão descritos a seguir.

1.4.1 Objetivo Geral

Visando facilitar o deslocamento das pessoas em ambientes de médio porte, se tem como objetivo desenvolver um sistema composto por duas aplicações de exibição de caminhos em mapas locais que possam ser executados em ambientes com pouco poder de processamento de dados de maneira rápida e eficiente possibilitando o fácil acesso e mobilidade no local.

1.4.2 Objetivos Específicos

- Estudar conceitos importantes e trabalhos relacionados ao tema proposto;
- Buscar, estudar e adaptar plantas baixas para serem utilizadas nas aplicações;
- Desenvolver aplicações adaptáveis a qualquer planta e que sejam úteis para a sociedade;
- Avaliar as aplicações;
- Documentar as aplicações e os resultados.

1.5 METODOLOGIA

No desenvolvimento deste trabalho foi realizado um estudo e o desenvolvimento de aplicações para localização em ambientes de médio porte. Sendo assim ele foi dividido nas seguintes etapas que estão detalhadas nos próximos capítulos:

- Pesquisa e elaboração de plantas baixas:
Nesta etapa foram realizados estudos e a aquisição de plantas baixas que serão utilizadas na visualização em um ambiente virtual para serem utilizadas na execução do aplicativo.

- Pesquisa e desenvolvimento de um modelo para conversão da planta em um arquivo de dados:
Nesta etapa foi feita uma análise para a melhor utilização das plantas na aplicação e realizado um estudo das ferramentas a serem utilizadas e como será classificada cada área da planta e quais locais são importantes para as aplicações. Será gerado um arquivo com os dados necessários para a navegação na planta.
- Pesquisa sobre métodos de busca de caminhos:
Nesta etapa foi realizado um estudo a respeito de um método de busca de caminhos que possa ser executado em ambientes com pouco poder de processamento de dados (celular, tablet, etc.) de maneira rápida e eficiente.
- Desenvolvimento da ferramenta *desktop* para geração de arquivos de dados:
Nesta etapa foram realizadas a implantação dos métodos de interação e do método escolhido durante as pesquisas para a conversão das plantas baixas para um arquivo de dados. Após o término desta etapa a aplicação usa a figura de uma planta baixa e após o usuário informar os pontos importantes, será gerado um arquivo com os dados a serem utilizados nas aplicações de localização *mobile* e *desktop*.
- Desenvolvimento das aplicações de localização *mobile* e *desktop*:
Nesta etapa foram realizadas as implementações das aplicações *mobile* e *desktop* para a localização e busca de caminhos. Nelas foram utilizados métodos de interação e de busca de caminhos previamente escolhidos nas pesquisas anteriores. Esta atividade teve como objetivo principal gerar duas aplicações semelhantes que tenham, se não as mesmas, funções muito parecidas e com o mesmo resultado, tudo isso utilizando os dados que foram feitos na aplicação *desktop* base.

2 POR QUE SISTEMA *OFF-LINE*?

Apesar dos sistemas de localização em ambientes abertos já estarem bem estabelecidos, eles não possuem a precisão para ambientes de pequeno e médio porte. Este fato dificulta a utilização, além de demandar a utilização de rede GPS ou internet e, portanto, restringir o público que vai usufruir da aplicação.

Dentre os diversos sistemas que poderiam ser utilizados estão o GPS (*Global Positioning System*) que é um sistema americano, é um dos mais conhecidos para navegação em ambientes abertos com a ajuda de satélites (HUERTA, MANGIATERRA e NOGUERA, 2005). Existem outros sistemas que também são utilizados como o GLONASS (*Global Navigation Satellite System*) que está sobre controle da Rússia, Galileo que é um sistema de navegação independente controlado pela União Europeia e também o Beidou que também é um sistema de navegação por satélite porém este controlado pela China (JOKINEN, FENG, et al., 2013) (BAUCH, 2005) (ZOU, 2010).

O sistema de GPS utiliza uma rede de satélites que ficam espalhados pela órbita do globo terrestre, eles são sincronizados e enviam sinais em um certo intervalo de tempo para poder se obter a localização em tempo real (HUERTA, MANGIATERRA e NOGUERA, 2005). Uma dificuldade é encontrada em função destes sinais enviados pelos satélites para os dispositivos GPS muitas das vezes não terem potência suficiente para atravessar paredes de edifícios e alguns lugares na área urbana (MIU, 2002).

Com esse problema da perda de sinal do GPS algumas soluções apareceram, dentre essa algumas tomaram destaque como a RADAR que foi desenvolvida pela Microsoft e também tecnologias tomando como base o Zigbee (BAHL e PADMANABHAN, 2000) (CHEN, YANG, et al., 2009).

Nestes últimos anos foram desenvolvidos vários aplicativos *mobile* com objetivo de suprir essa necessidade ou pelo menos melhorar a atual situação, um dos métodos utilizados foi a atenuação do sinal, como o ângulo de chegada e a diferença do tempo de chegada. Mesmo se utilizando de recursos os resultados ainda foram muito limitados devido às reflexões das ondas de sinal no ambiente fechado (SOARES, 2013).

Não tendo sido contornado o problema de sinal e também pela precisão necessária em função do tamanho dos ambientes, optou-se por aplicações que não restrinjam o público e não demandem conexão com internet, GPS ou similares.

3 TERMOS E ÁREAS TÉCNICAS

Neste trabalho alguns termos e áreas são importantes uma vez que o desenvolvimento não envolve apenas a programação, mas o entendimento do ambiente, definição dos dados de entrada e saída, configurações e técnicas computacionais que podem ser utilizadas para busca de caminhos.

Nas próximas seções dados relativos a cada uma destas partes estão abordados. Na seção 3.1 está explicado o que se considera como planta baixa que será um dado de entrada. Na seção 3.2 estão informações a respeito de agentes computacionais, na 3.3 estão conceitos da teoria dos grafos, úteis para o entendimento deste trabalho e na seção 3.4 tipos de ambientes, além de algoritmos e técnicas para busca e planejamento de caminhos, estes os quais foram necessários para o embasamento teórico para o desenvolvimento das aplicações.

3.1 PLANTAS BAIXAS

Planta Baixa é o nome dado ao desenho feito de uma construção, em geral, a partir de um corte horizontal à altura de 1,5m a partir da base. É um diagrama dos relacionamentos entre salas, espaços e outros aspectos físicos em um nível de uma estrutura. Nela devem estar detalhadas em escala as medidas das paredes, portas, janelas, o nome de cada ambiente e seu respectivo nível.



Figura 3. Planta Baixa - Shopping Interlagos São Paulo/SP

3.2 AGENTES INTELIGENTES

"Um agente é algo capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por meio de atuadores" (STUART J. RUSSELL, 1995).

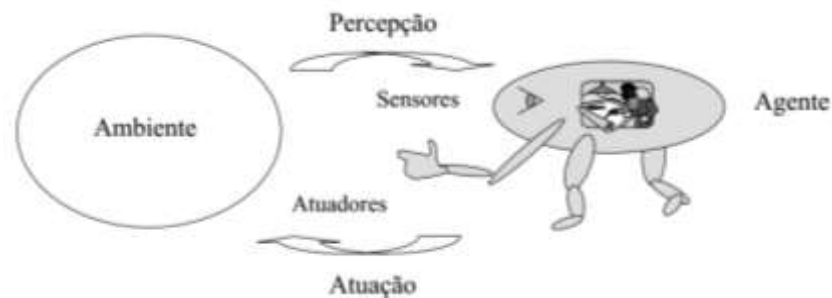


Figura 4. Interação com ambientes através de sensores e atuadores

Além de uma estrutura de navegação transitável é necessário obter uma maneira de navegação aplicável a este ambiente para encontrar os caminhos desejáveis. A utilização de agentes inteligentes torna-se um componente chave para a aplicação em que podem ser utilizadas algumas características comportamentais humanas em agentes artificiais, em quesitos como comportamento, conhecimento, intenção, etc. (FROZZA, 1997).

O comportamento de um agente em um dado ambiente depende de suas próprias metas e informações já conhecidas, os efeitos das ações do agente são notados através de eventos, que podem corresponder a modificações realizadas no ambiente pelo agente. Os agentes inteligentes são caracterizados como aqueles que possuem a capacidade de se adaptar e serem flexíveis ao ambiente, que possuem seus próprios objetivos podendo realizá-los de modo eficiente (WERNER e CASTELFRANCHI, 1992).

3.2.1 Características de um agente

Segundo Frozza (1997), um agente inteligente possui diversas características dentre as quais aquelas citadas a seguir são importantes para este trabalho:

- É uma entidade real ou virtual
- Está inserido em um ambiente, que pode mudar com as atuações do agente;
- Percebe o ambiente que está inserido através de percepções;

- Possui autonomia;
- Possui a capacidade de agir no ambiente, executar ações que podem mudar o estado do ambiente ou o estado interno do agente;
- Pode comunicar com outros agentes para auxiliar na conclusão de alguma tarefa;
- Tem capacidade social, podem interagir com o ambiente e outros agentes (reais ou artificiais);
- Possui metas;
- Pode agir em resposta ao ambiente ou por conta própria;
- Possui a capacidade de modificar seu comportamento em resposta a troca de informações ou aumento de conhecimento;
- Pode se mover;
- Possui algum tipo de conhecimento;
- Pode ser capaz de raciocinar sobre as atividades e informações de outros agentes para tomar suas decisões.

3.2.2 Arquiteturas de Agentes

Segundo Wooldridge e Jennings (1995) a constituição, estrutura e funcionamento de um agente podem ser realizados através de três tipos arquiteturais: deliberativas, reativas ou híbridas.

Arquiteturas Deliberativas seguem a abordagem clássica da Inteligência Artificial, onde os agentes contêm um modelo simbólico do mundo, explicitamente representado, e cujas ações são tomadas pelo raciocínio lógico, através da montagem de um plano de ações.

Arquiteturas Reativas são baseadas em comportamento, na proposta de que um agente pode desenvolver inteligência e interagir com seu ambiente, a escolha de uma ação é determinada pela situação em que o agente se encontra.

Arquiteturas Híbridas apresentam um misto de componentes das arquiteturas deliberativas e reativas com o objetivo de ser mais adequada e funcional para a construção de agentes para determinada aplicação.

3.3 INTRODUÇÃO À TEORIA DOS GRAFOS

Em uma parte deste trabalho, há o objetivo de calcular e mostrar qual é o caminho que deve ser percorrido pelo usuário para que o mesmo chegue ao objetivo. Por este motivo, alguns conceitos de teoria de grafos são fundamentais para deixar mais claro qual será a abordagem da solução, e será dada em especial atenção ao problema de busca de caminhos em grafos que é o objetivo do estudo. As nomenclaturas e as notações que padronizaremos agora, assim como as demonstrações dos teoremas seguem o exposto em GONZAGA (1978) e GONZAGA (2005).

A teoria dos grafos é uma ferramenta matemática simples e poderosa usada na resolução de inúmeros problemas tais como: busca de caminhos, fluxo de redes, logística. E ela vem de muito tempo atrás já que a citação mais antiga foi no ano de 1736, onde o matemático suíço Leonhard Euler apresentou o problema das sete pontes de Königsberg em seu artigo “The Seven Bridges of Königsberg”.

O problema apresentado por Euler era de uma cidade alemã chamada Königsberg, onde havia uma ilha, um rio que passava pela cidade e que logo após passar essa ilha ele se bifurcava, como pode ser visto na Figura 5. Sobre o rio ligando margem a margem haviam sete pontes e o problema era descobrir se seria possível em uma caminhada contínua passar por todas as pontes sem passar duas vezes sobre a mesma.

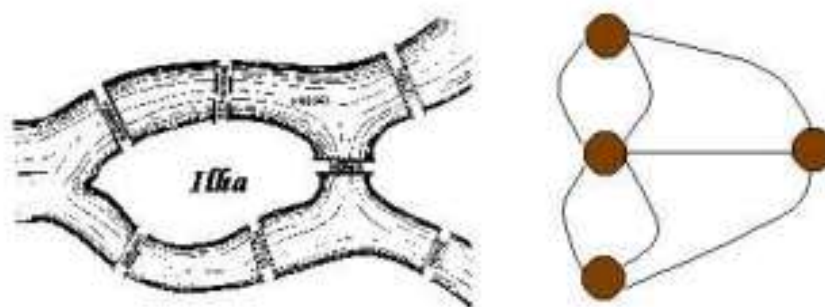


Figura 5. Pontes de Königsberg

Após Euler analisar o problema ele trocou as margens por pontos (nós) e as pontes por linhas (ramos) e provou que o problema não teria solução. Rabuske (1992) mostra uma análise de como Euler teria chegado a esta conclusão. Deste modo teve

o início da Teoria dos Grafos e após este experimento, muita evolução aconteceu e alguns conceitos úteis para este trabalho são apresentados a seguir.

3.3.1 Principais Conceitos

Um grafo consiste de dois grupos, nós e ramos $G = (N, M)$ em que N é um conjunto de nós (n_1, n_2, n_3, \dots) finito não podendo ser nulo e M é um conjunto de pares ordenados de elementos de N , $M \subset N \times N$, onde M é um conjunto de ramos $r = (n_i, n_j)$, sendo n_i e n_j as extremidades inicial e final respectivamente. Sempre que N e M forem finitos G será um grafo finito (ROSSETTO, 2007).

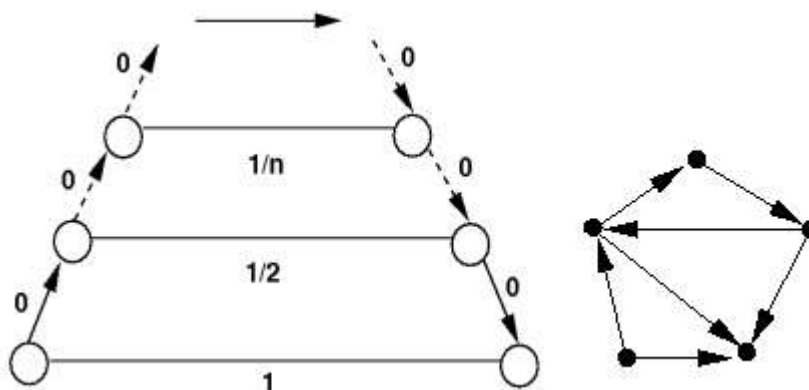


Figura 6. Exemplo de grafo infinito e finito

Se no grafo G não existir mais de um ramo ligado aos mesmos nós é considerado um grafo simples (ROSSETTO, 2007).

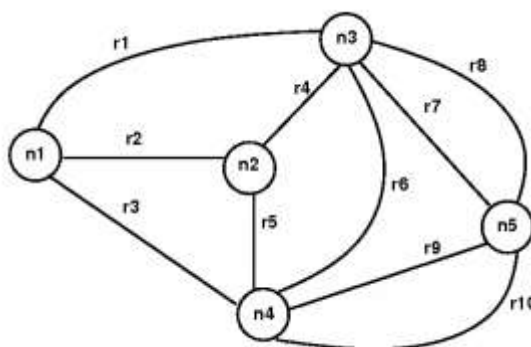


Figura 7. Exemplo de grafo com múltiplos ramos

Sempre que um nó for o início de um ramo ele deve ser chamado de cauda do ramo e sempre que o final de um ramo deve ser chamado de cabeça do ramo (ROSSETTO, 2007).

Dados dois nós, só pode-se dizer que um é sucessor do outro, quando os mesmos estão ligados por um mesmo ramo.

Um grafo só é considerado completo, se e somente se, todos os nós forem sucessores de todos os outros nós do grafo.

Na nossa aplicação os grafos são sempre orientados, já que os ramos são pares direcionais e não precisam ser completos em função das plantas baixas dos ambientes em questão. Para definir grafo não orientado, basta não utilizar a orientação, ou definir os nós como pares não orientados $\{n_i, n_j\}$, ou ainda exigir que sempre estejam presentes os ramos (n_i, n_j) e (n_j, n_i) .

3.4 AMBIENTES E PLANEJAMENTO DE CAMINHOS

O planejamento de caminhos é o processo no qual a partir de um mapa previamente modelado, seja possível ir de um ponto ao outro do ambiente. Neste processo, é importante encontrar uma boa maneira de representar o ambiente. Deste modo, para realizar a busca de caminhos utilizam-se as informações computadas pela decomposição do ambiente em subpartes para depois buscar caminhos entre os pontos de origem e destino. Esta opção será desenvolvida em um componente desktop que a partir de uma planta baixa como referência, o usuário irá determinar os espaços caminháveis (salas, corredores, etc.) e os prováveis pontos de interesse (destinos). Para este fim, foram estudadas possíveis técnicas para o desenvolvimento

das aplicações de a criação de mapas virtuais e de busca e planejamento de caminhos, estas técnicas estão descritas nos tópicos a seguir.

3.4.1 Representação e Decomposição de Ambientes

Existem muitos tipos de representação de ambientes e cada uma delas tem suas vantagens e desvantagens, o que torna a escolha da representação correta, extremamente importante. Uma das opções é que a representação descreva o ambiente em uma matriz onde cada célula representa uma área definida do mundo real. Cada célula então é carregada com informações específicas sobre aquela região como, por exemplo, se existe ou não obstáculo nela. A quantidade de memória necessária para o *grid* depende diretamente do tamanho do mapa, da escala utilizada e de quanta informação se deseja armazenar por célula (DEICU, 2012).

A fase de planejamento necessita, para sua execução, de uma configuração inicial e de uma final das definidas em um mesmo sistema de referência. Além disso, pressupõe o conhecimento de um mapa baseado no qual serão avaliadas as opções, a posição corrente por onde está se movendo e obstáculos existentes. A partir dessas informações, o planejador deve ser capaz de gerar uma curva que conduza de forma segura desviando de obstáculos quando existirem, entre as duas configurações (SANTANA, 2007).

As seções a seguir detalham estudos que embasaram a escolha dos métodos desenvolvidos nas aplicações.

3.4.1.1 Mapas de rotas – *Roadmaps*

Um mapa de rotas exhibe os caminhos possíveis de transporte, deste modo, para criar e representar esses mapas criam-se segmentos de retas que conectam os vértices de um obstáculo ao vértice de outro obstáculo de modo que este não passe pelo interior de nenhum outro obstáculo. Os segmentos de retas conectados formam um conjunto de caminhos denominado *Roadmaps*.

Neste método um nó é representado pelos vértices dos obstáculos, um nó do grafo só é conectado a outro se estes estão no campo de visão de ambos. Além disso, os pontos de origem, destino ou intermediários devem estar no campo de visão de

pelo menos um nó do grafo. O grafo criado é denominado grafo de visibilidade (SANTANA, 2007).

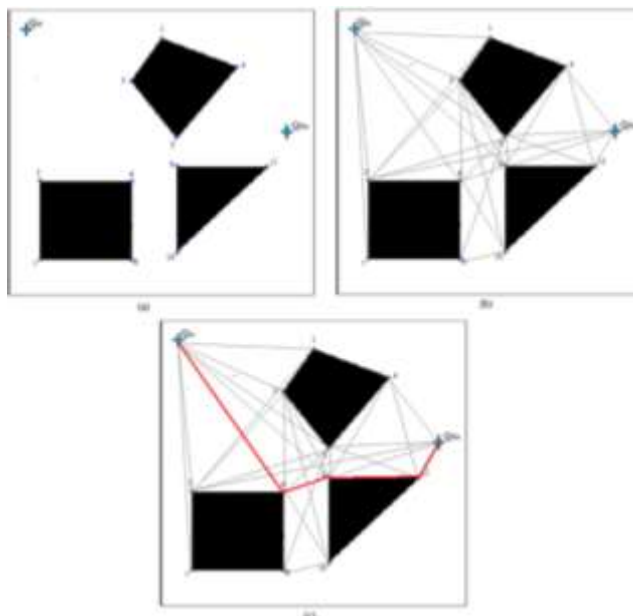


Figura 8. Diagrama de Visibilidade

O diagrama de visibilidade é construído a partir dos vértices dos obstáculos, da posição inicial e da posição final. Arestas são traçadas entre os vértices desde que não haja um obstáculo no caminho. Logo após, seleciona-se um algoritmo de planejamento discreto para encontrar um caminho viável entre o ponto inicial e o ponto final (DEICU, 2012).

3.4.1.2 Campo de Potencial

Outra possibilidade é a utilização de campos de potência, que, segundo Santana (2007), é um método de navegação e desvio de obstáculos quando não se tem conhecimento a princípio do mapa do ambiente, portanto os obstáculos são percebidos à medida que se navega no mapa. A implementação consiste em sair da origem e ir ao destino em pequenos passos direcionados sobre a ação de forças, no qual o destino possui uma força de atração e os obstáculos uma força de repulsão.

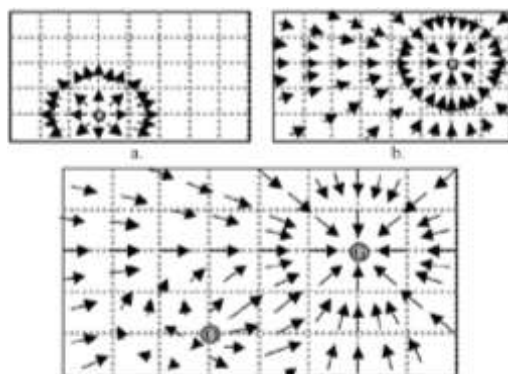


Figura 9. Campos Potenciais

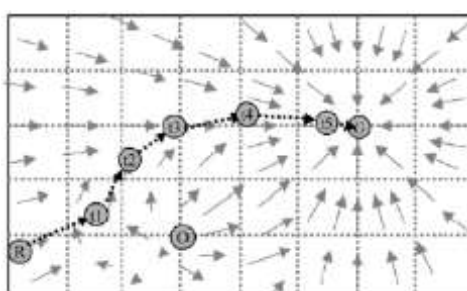


Figura 10. Provável caminho escolhido

3.4.1.3 Decomposição de células

Ainda de acordo com Santana (2007), o método de decomposição em células consiste em dividir o espaço livre em regiões convexas simples (células), de forma que um caminho entre quaisquer duas configurações em uma mesma célula possa ser facilmente obtido. Quando duas células possuem uma fronteira em comum, elas são adjacentes. A relação de adjacência ou conectividade entre as células é representada por um grafo não direcional. Este grafo é denominado grafo de conectividade.

Os métodos de decomposição em células podem ser divididos em métodos de decomposição exata e métodos de decomposição aproximada. Na decomposição exata, a união das células corresponde exatamente ao espaço de composição livre. A divisão de células é feita em pontos críticos de espaço, ou seja, pontos em que ocorre a divisão do espaço livre por um obstáculo (SANTANA, 2007).

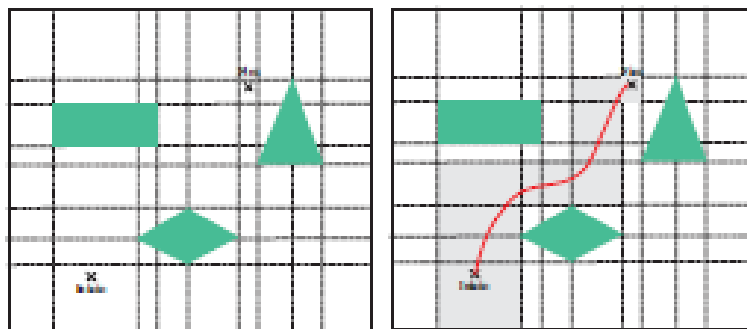


Figura 11. Decomposição em células exatas

Na decomposição aproximada, o espaço de configuração livre é decomposto em células de um formato específico, por exemplo, quadrados de dimensão fixa. A união destas células não corresponde exatamente ao espaço livre, mas está incluída nele (SANTANA, 2007).

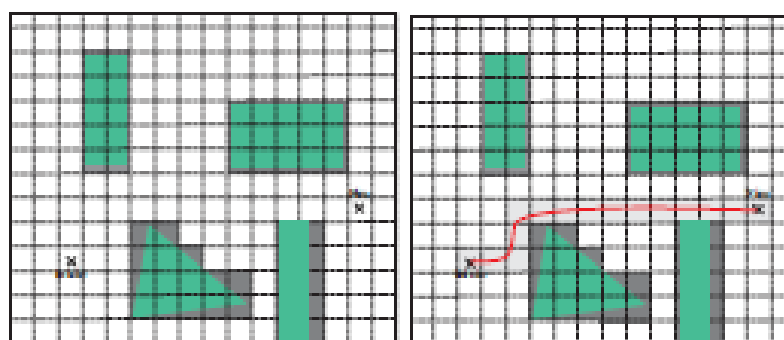


Figura 12. Decomposição em células Aproximadas

3.4.2 Algoritmo de Dijkstra

“O algoritmo de Dijkstra foi publicado em 1959 pelo holandês, Edsger Wybe Dijkstra”(DIAS). Este algoritmo é o mais famoso e aplicado na resolução de problemas de caminho mais curto em um grafo direcionado de pesos não negativos, pois caso haja presença de valores negativos, o algoritmo não garante exatidão da sua solução. O algoritmo para solucionar o problema em um grafo com pesos negativos é o algoritmo de Bellman-Ford, que possui mais tempo de execução que o Dijkstra(DIAS).

Este algoritmo parte de uma estimativa inicial para o custo mínimo e vai sucessivamente ajustando esta estimativa. Ele considera que um vértice estará fechado quando já tiver sido obtido um caminho de custo mínimo do vértice tomado como raiz da busca até ele, ou seja, onde a soma dos pesos das arestas seja mais

baixo entre todos os caminhos possíveis entre os pontos. Caso contrário ele é dito estar aberto (GONZAGA, 1978). Para entender o funcionamento do algoritmo de Dijkstra, ele segue os passos:

- 1- Inicialmente os vértices do grafo em estudo devem ser numerados;
- 2- Atribui valor aos vértices do grafo: o vértice de saída (raiz da busca) com o valor igual a 0 , e os demais vértices com valor igual a ∞ ;
- 3- A partir do vértice de raiz, consulta-se os vértices adjacentes a ele. Calcula-se sua distância entre os vértices (soma-se o peso das arestas);
- 4- Dentre todos os vértices adjacentes ao vértice de raiz, escolhe-se aquele com a menor distância e fecha-o. Recalcula os próximos vértices adjacentes a ele;
- 5- Dentre todos os vértices adjacentes não fechados, escolhe-se aquele com a menor distância. Recalcula os próximos vértices adjacentes a ele;
- 6- Dentre todos os vértices não fechados, escolhe-se aquele com a menor distância e fecha-o.
- 7- Por fim, escolhe-se como precedente aquele com menor distancia e fecha-o, assim a busca é finalizada.

3.4.3 Algoritmo B*

B* é um algoritmo de busca em grafos que encontra o caminho de menor custo a partir de um determinado nó inicial para qualquer nó objetivo (de um ou mais objetivos possíveis). Publicado pela primeira vez por Hans Berliner em 1979. Os intervalos algoritmo armazenam para cada nó da árvore, ao contrário de estimativas de valor de um só ponto. Então, nós folhas da árvore podem ser pesquisados até que um dos nós de nível superior tem um intervalo que é claramente "melhor" (BERLINER, 1979).

Nós folha de uma árvore B* são dadas por avaliações, que são intervalos em vez de números individuais. O intervalo deve conter o verdadeiro valor desse nó. Se todos os intervalos ligados a nós folha satisfizerem essa propriedade, em seguida, B* irá identificar um caminho ideal para o estado objetivo.

Para fazer backup dos intervalos dentro da árvore, o limite superior é definido de um pai para o máximo dos limites superiores dos filhos. Um limite inferior do pai é

definido para o máximo do limite inferior dos filhos. Note-se que os filhos diferentes podem fornecer esses limites.

B^* expande sistematicamente os nós, a fim de criar "separação", que ocorre quando o limite inferior de um filho direto da raiz é pelo menos tão grande como o limite superior de qualquer outro filho direto da raiz. Uma árvore que cria separação na raiz contém uma prova de que o melhor filho é pelo menos tão bom como qualquer outro filho.

Na prática, pesquisas complexas pode não terminar dentro de limites práticos de recursos. Assim, o algoritmo é normalmente aumentado com critérios de terminação artificiais, tais como o tempo ou limites de memória. Quando um limite artificial é atingido, então você deve fazer um julgamento sobre heurística que se movem para selecionar. Normalmente, a árvore poderia fornecê-lo com provas extensas, como os intervalos de nós raiz.

B^* é um processo que toda a árvore é mantida na memória, e repetidamente desce para encontrar uma folha em expansão. Esta seção descreve como escolher o nó para expandir.

Na raiz da árvore, o algoritmo aplica uma de duas estratégias, chamado prove-best e disprove-rest. Na estratégia de prove-best, o algoritmo seleciona o nó associado com o mais alto limite superior. A esperança é que a expansão desse nó irá elevar seu maior limite inferior do que qualquer outro nó do limite superior.

A estratégia de disprove-rest seleciona o filho da raiz, que tem o segundo maior limite superior. A esperança é que, expandindo o nó que você pode ser capaz de reduzir o limite superior para menos do que o limite inferior da melhor filho. Note-se que a aplicação da estratégia de disprove-resto é inútil até que o limite inferior do nó filho que tem o maior limite superior é o mais alto entre todos os limites inferior.

A descrição original do algoritmo não deu qualquer orientação adicional sobre qual a estratégia a ser selecionado. Existem várias alternativas razoáveis, como a expansão da escolha que tem a árvore menor. Uma vez que um filho da raiz foi selecionado (usando prove-best ou disprove-rest), então o algoritmo desce para um nó folha, selecionando repetidamente o filho que tem o maior limite superior.

Quando um nó folha é atingida, o algoritmo gera todos os nós sucessores e atribui intervalos usando a função de avaliação. Em seguida, os intervalos de todos os nós têm que ser feitos usando a operação de backup.

Quando transposições são possíveis, então a operação de backup pode precisar alterar os valores de nós que não se encontram no caminho de seleção. Neste caso, o algoritmo precisa ponteiros dos filhos de todos os pais, de modo que as alterações podem ser propagadas. Note-se que a propagação podem cessar quando uma operação de cópia de segurança não alterar o intervalo associado a um nó. Se os intervalos estão incorretos (no sentido em que o valor da teoria de jogos do nó não estiver contido dentro do intervalo), em seguida, B^* não pode ser capaz de identificar o caminho correto. No entanto, o algoritmo é bastante robusto a erros na prática. O algoritmo B^* de pesquisa tem sido utilizado para calcular a estratégia ideal em um jogo de soma de um conjunto de jogos combinatórios (SHEPPARD, 2002).

4 TRABALHOS RELACIONADOS

Uma solução comumente adotada em muitos *shoppings center* é a instalação de mapas em locais estratégicos para que o visitante consiga se localizar e decidir para onde ele deve seguir a fim de chegar ao seu local de destino.

Esta solução nem sempre é eficiente para os usuários, pois além do local procurado, eles precisam agora saber onde estão os mapas informativos.

O diferencial aqui é a possibilidade do usuário ter acesso no seu dispositivo móvel Android às informações que necessita para localização. Desta forma, o usuário pode encontrar e obter orientações e informações sobre trajetos para o deslocamento, ajudando-o a chegar ao seu destino, lugares específicos (salas, departamentos, lojas, dentre outros) do estabelecimento, necessitando apenas da planta do local com informações previamente definidas.

A empresa Apek (APEK, 2013) desenvolveu o Apek Trackmall, lançado no mercado com o objetivo de atingir os shopping centers. Este sistema tem como objetivo principal mostrar caminhos de uma loja a outra a visitantes e conta com outras funcionalidades, como suporte à programação de cinema, teatro, eventos, serviços, promoções.

Uma das grandes funcionalidades que é dita pelo fabricante é a fácil utilização e configuração, já que ele foi projetado para ser utilizado em telas *touchscreen* e todo o layout por ser personalizado.

Ele foi criado em plataforma Web, o que facilita muito quando a disponibilidade e compatibilidade. Aproveitando-se da tecnologia a Apek seguiu em frente e desenvolveu o mesmo aplicativo para aparelho móveis.

Atualmente esse sistema está ativo no shopping Boulevard na cidade de Londrina, PR e no shopping Eldorado, em São Paulo.

Trata-se de uma versão comercial do presente trabalho, mas que não havia conhecimento quando este foi iniciado. O diferencial é que nosso projeto não depende de qualquer conexão do usuário para consultas, buscas e navegação.

4.1 SHOPPING TAMBORÉ

No shopping Tamboré localizado na cidade de Barueri no estado de São Paulo, foi lançado um sistema de localização interna, pelo que foi informado pelo shopping, foram instalados seis terminais com o sistema, nesses terminais os usuários poderão busca lojas, produtos, enviar reclamações, sugestões e poderão preencher uma pesquisa de opinião.

Assim como o Trackmall, o sistema do shopping Tamboré também tem como objetivo principal mostrar caminhos de uma loja até a outra. Pelas informações disponibilizadas pelo shopping, esse sistema é inferior em relação ao número de recursos se comparado ao Trackmall, mas mesmo assim não deixa de ser um sistema interessante e com um objetivo semelhante ao desenvolvido neste trabalho.

5 PRIMEIRA VERSÃO DAS APLICAÇÕES

Inicialmente foram implementados dois modelos, os quais foram aperfeiçoados para a solução proposta. O primeiro modelo implementado foi uma aplicação simples para a busca de pontos de interesse e a busca do menor caminho entre uma lista de elementos, utilizando agentes reativos para a busca dos pontos e um agente inteligente para a busca do menor caminho em uma lista feita pelos agentes reativos.

A segunda aplicação teve como objetivo que um agente solto em um determinado ponto em uma matriz que seria o ambiente, busca um ponto de interesse. Porém nesta matriz ele tem obstáculos e o agente tem inteligência para poder desviar destes obstáculos e não passar no mesmo lugar por onde ele já passou.

5.1 SISTEMA PARA LOCALIZAÇÃO DE PONTOS E CAMINHOS

O ambiente foi constituído usando uma matriz 12X12, contendo cinco pontos de interesse, dois agentes. O objetivo é que os agentes conheçam todos os pontos de interesse e os marquem em um quadro negro.

Utilizando-se desta sequência foi colocado um agente que consultando essa lista no quadro negro percorre o menor caminho entre o primeiro e o segundo, do segundo para o terceiro e assim por diante.

Um caso muito conhecido assim são as formigas, já que elas saem em busca de alimento sem realmente saber onde os mesmos se encontram, mais assim que acham elas após deixarem um rastro de feromônio as outras formigas também conseguem chegar ao alimento. Deste modo elas criam um grafo invisível a olho nu onde a comida são os nós do grafo, as arestas são os rastros de feromônio deixados pelas formigas e as formigas são os agentes (BIRCH e HAYNES 1982).

5.1.1 Metodologia

A aplicação foi desenvolvida em C#, por meio de matrizes e arrays. Os agentes têm como entrada pontos de origem na matriz. Após alocado neste ponto eles sempre caminham para posição a sua direita da matriz caso essa posição seja

inexistente ele descerá uma linha e começará a se deslocar para a esquerda, assim por diante até que não haja mais nenhuma linha, ai ele começará a ir para a linha de cima até chegar ao topo. Quando achar um ponto de interesse será guardado em uma lista a posição do ponto de interesse e marcado no quadro negro.

Após a localização de todos os pontos de interesse e todos marcados no quadro negro, será retirado da matriz os agentes que encontraram os pontos de interesse e será colocado um agente no primeiro ponto de interesse que está marcado no quadro negro. Após alocado no primeiro ponto de interesse ele começará a percorrer o menor caminho para chegar no segundo ponto de interesse marcado no quadro negro, depois ele irá percorrer o menor caminho entre o segundo e o terceiro até que chegue no último ponto de interesse marcado no quadro.

5.1.2 Desenvolvimento

Os dois agentes utilizados para percorrer a matriz e encontrar todos os pontos de interesse são sempre alocados nos dois primeiros pontos de interesse da matriz e logo após serem alocados eles seguem as seguintes regras:

- Verifica se existe um ponto de interesse ao seu redor, nas quatro direções: esquerda, direita, cima, baixo;
- Caso haja algum ponto de interesse em alguma das direções ele verificará se já não conhece aquele ponto, caso não conheça ele irá para o ponto de interesse e marcará em sua tabela que conhece aquele ponto;
- Caso não exista nenhum ponto de interesse ao seu redor ele tentará andar na direção que está o seu status, a direção padrão é direita;
- Caso ele não possa andar nesta direção pois o ambiente termina onde ele está, ele descerá uma linha e inverterá a sua direção;
- Se os dois agentes se encontrarem os dois invertem as respectivas direções e continuam procurando os pontos de interesse;
- Eles continuarão procurando até que os dois agentes achem o último ponto de interesse.

Logo após são retirados os dois agentes que encontraram os pontos de interesse e colocado o agente que conhece todos os pontos pois ele sabe ler o quadro

negro onde estão anotados os pontos de interesse conhecidos, e assim ele segue as seguintes regras:

- Verifica se a posição de da linha a onde ele se encontra é menor que a linha do próximo ponto, caso seja verdadeiro ele descera uma linha, caso seja falso ele verifica se ela é maior, caso seja maior ele subirá uma linha, caso seja igual ele não faz nada em questão da linha;
- Verifica se a posição da coluna onde ele se encontra é menor que a coluna do próximo ponto, caso seja maior ele irá para próxima coluna, caso seja falso ele verifica se ela é maior, se for maior ele voltará uma coluna, caso seja igual ele não fará nada em questão da coluna;
- Ele fará isso até que chegue no último ponto de interesse conhecido da matriz que está anotado no quadro negro.

5.2 SISTEMA PARA LOCALIZAÇÃO DE UM PONTO DESVIANDO DE OBSTÁCULOS

Tendo um mapa predefinido no formato de uma matriz n por n com obstáculos no qual se determina o ponto de destino, dentro deste mapa, onde dada uma origem se deseja encontrar um caminho para o destino.

5.2.1 Desenvolvimento

Para resolver o problema foi inserido um agente em algum local da matriz. Este agente responde às percepções do ambiente interpretando a entrada de modo a tomar as decisões correspondentes as regras pré-estabelecidas, assim este analisa informações do ambiente, tais como, obstáculos, caminho livre, etc. e assim caminhar pela matriz até encontrar o ponto de interesse. A Figura 13 **Error! Reference source not found.** apresenta o estado inicial do mapa com as posições determinadas como origem, na escada, e destino do agente, na Coted.

									Escada	
Lab3	Lab1			Lab2	Lab4	A1		C1		
Lab5										A2
	Escada									
Hw										Demat
	D								Coted	

Figura 13. Agente na origem e o ponto de interesse

O agente analisa as casas ao redor (horizontal e vertical) e verifica se estas são espaços livres caminháveis, ele pode ir para frente e para os lados. A partir dos espaços livres ele verifica a distância destas casas e avança para a casa que tiver a menor distancia com o destino, assim move-se pela matriz até que uma das casas ao redor seja um ponto de interesse então move para casa do ponto de interesse encontrado.



Figura 14. Possível ação do agente em azul, em vermelho obstáculos encontrados

A figura 14 demonstra como o agente toma a decisão de qual seria a próxima casa, ele está saindo da escada então não pode retornar a posição onde já esteve, restando as 2 posições ao lado, a casa escolhida será a que possui menor distância entre ela e o destino, em linha reta, ignorando os obstáculos e caminhos.

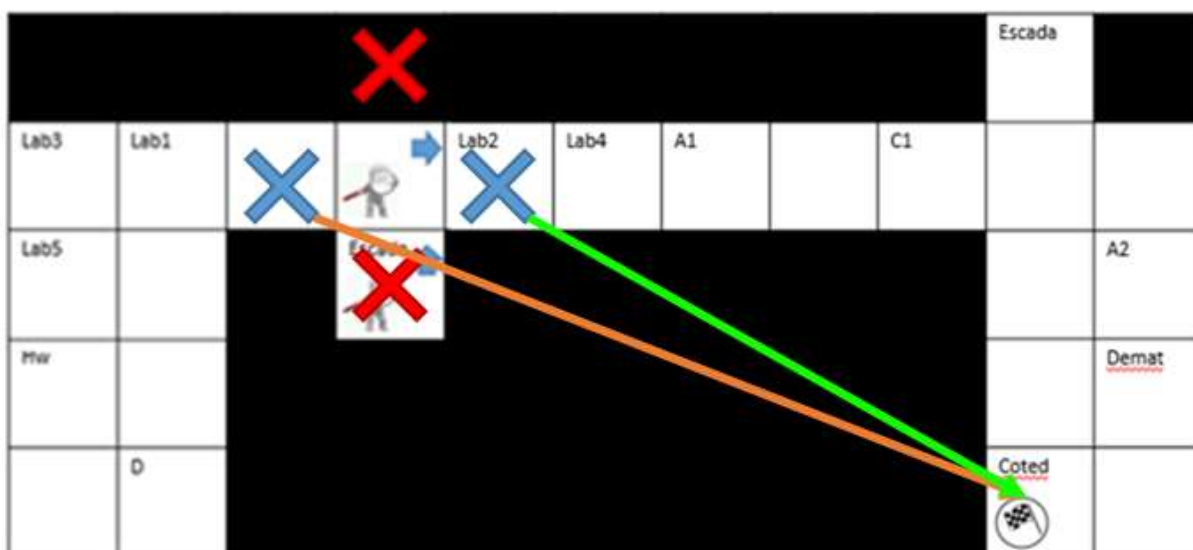


Figura 15. Agente ao decidir a próxima casa de destino

Deste modo é possível encontrar o caminho de maneira mais eficiente que se estiver andando a esmo no ambiente, pois a cada passo o agente acaba desviando dos obstáculos e chegando cada vez mais perto de seu destino. Porém é necessário saber o local do ponto de destino e garantir que há caminho até lá, pois uma situação onde o destino está atrás de uma parede fará com que o agente siga o caminho "errado" e tenha que corrigir.

6 SEGUNDA VERSÃO DAS APLICAÇÕES

Ambas aplicações tiveram seu objetivo alcançado, cada uma com um tipo de implementação diferente e, a partir de agora, o desafio é fundir as duas técnicas para que eles funcionem em conjunto para a resolução do problema proposto.

6.1 RESULTADO PARCIAL - SISTEMA PARA LOCALIZAÇÃO DE PONTOS E CAMINHOS

Nesta parte foram usados dois tipos de agentes, os que buscam os pontos de interesse são reativos que se preocupam apenas em achar o ponto de interesse e tem suas regras. No caso da busca de caminho foram utilizados agentes inteligentes que têm memória para poder saber onde se encontram os pontos de interesse e poder traçar o menor caminho conforme as regras previamente definidas.

Na Figura 16 é apresentada a matriz com cinco pontos de interesse marcados com o valor 0.

1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1

Figura 16. Matriz com pontos de interesse

Dois agentes reativos foram alocados nos dois primeiros pontos de interesse dispostos na matriz, para que a partir deles comecem a percorrer a matriz em busca dos outros pontos de interesse.

Na figura 17 é um exemplo mostrando que quando um agente acha um ponto de interesse assim que ele se deslocar deste mesmo ele irá mostrar uma mensagem com as informações deste ponto de interesse com as coordenadas X e Y.

```

achou x = 0 y = 11
achou x = 1 y = 6
  1  1  1  1  1  1  1  1  1  1  1  0
  1  1  1  1  1  1  0  2  1  1  1  2
  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1
  1  0  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  0  1  1  1  1  1
  1  1  1  1  1  1  0  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  1

```

Figura 17. Agentes encontrando pontos de interesse

Na Figura 18 é um exemplo com uma matriz um pouco menor que foi utilizado no resto do trabalho, ela foi feita apenas para exemplificar, mostrando o que acontece quando um agente se movimenta, no caso do exemplo ele está com a sua direção voltada para a direita e conforme ele anda e chega no final do ambiente logo ele toma a decisão de descer uma linha e inverte a sua direção caminhando a partir de agora para a esquerda na busca do próximo ponto de interesse.

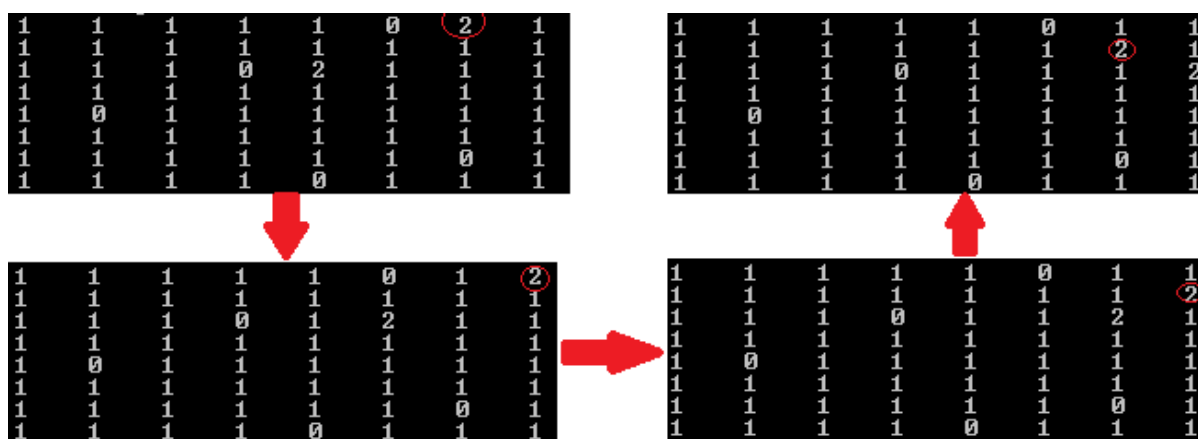


Figura 18. Agente se movendo

Para decidir qual o caminho que deve ser tomado é feito uma série de testes através da função “andaSegunda” que é mostrada na Figura 19. A função faz com que o agente primeiramente verifique se existe algum ponto de interesse em uma das quatro direções: esquerda, direita, cima e baixo. Sempre que tiver um ponto de interesse ele dará preferência para esta posição já que um dos objetivos é buscar os pontos.

```

public static int[,] andaSegunda(int[,] especo)
{
    int cont = 0;
    int flag = 0;

    for (int i = 0; i < 2; i++)
    {
        cont = 0;

        if ((leg[i].posy - 1) > -1)
        {
            if (especo[leg[i].posx, leg[i].posy - 1] == 0)
            {
                for (int h = 0; h < leg[i].listaX.Count; h++)
                {
                    if ((leg[i].listaX[h] == leg[i].posx) && (leg[i].listaY[h] == leg[i].posy - 1))
                    {
                        cont = 1;
                        break;
                    }
                }
            }
            if (cont == 0)
            {
                if (leg[i].valor == 3)
                {
                    especo[leg[i].posx, leg[i].posy - 1] = 3;
                    especo[leg[i].posx, leg[i].posy] = 0;
                    leg[i].listaX.Add(leg[i].posx);
                    leg[i].listaY.Add(leg[i].posy);
                    if (existe(leg[i].posx, leg[i].posy))
                    {
                        c.pX.Add(leg[i].posx);
                        c.pY.Add(leg[i].posy);
                    }
                }
            }
        }
    }
}

```

Figura 19. Função "andaSegunda"

Sempre que os dois agentes se encontrarem eles também invertem os seus *status* de direção, deste modo evitando a colisão entre dois agentes. Eles continuam andando em busca dos pontos até que achem todos os pontos ou que cheguem ao final da matriz.

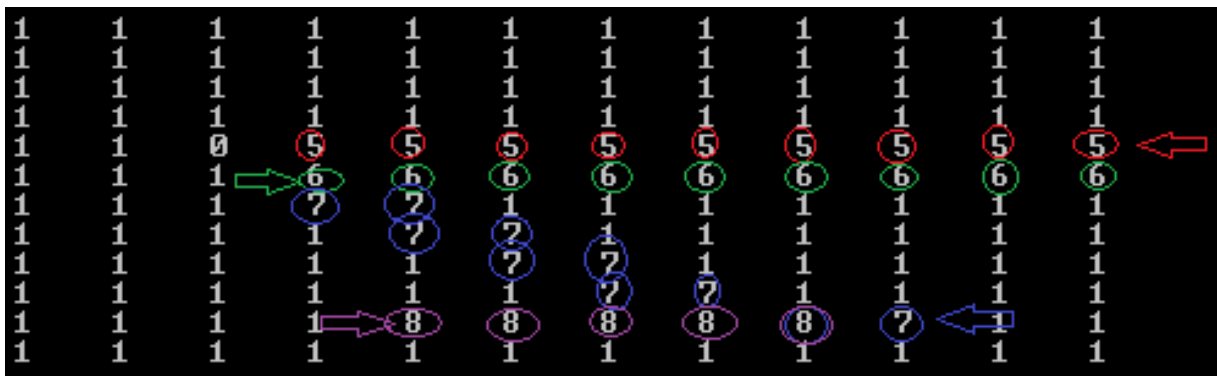


Figura 20. Buscando o menor caminho

Na Figura 20 um agente inteligente foi colocado na matriz para percorrer o menor caminho na sequência de pontos que foram gravados, assim que ele começar a andar ele deixa um rastro pela matriz para ficar evidente as células que ele passou mostrando o menor caminho, sempre que ele chega em um ponto de interesse, ele incrementa o número que ele está deixando o rastro. Para isto, ele começa com o valor 5 e vai incrementando até achar todos os caminhos que foram pré-determinados. Na figura 20 estão marcados com as cores vermelho, verde, azul e roxo os diferentes caminhos que foram percorridos. Também estão indicados com flechas os locais onde existem pontos de interesse.

Assim foi observado que os agentes reativos utilizam-se da força bruta para poder encontrar os pontos de interesse, apesar disso eles cumprem perfeitamente o objetivo que é a busca dos pontos. Os agentes utilizados para a busca dos pontos são agentes reativos, logo eles não aprendem e não trocam informações e nada mais complexo eles apenas reagem ao ambiente como no caso deste trabalho que ele reage aos pontos de interesse.

A busca do menor caminho entre os pontos da lista é feita por um agente inteligente, com processamento muito menor já que ele não se utiliza da força bruta por já saber a posição exata dos pontos bastando apenas percorrer o menor caminho em relação ao próximo ponto de interesse.

6.2 RESULTADO PARCIAL - SISTEMA PARA LOCALIZAÇÃO DE UM PONTO DESVIANDO DE OBSTÁCULOS

Neste projeto inicialmente foi determinado um mapa fixo com obstáculos que podem ser paredes ou objetos no caminho. Estes obstáculos são representados com o caractere X no mapa e os espaços livres são representados com o caractere ESPAÇO como apresenta na imagem a seguir:

```
static final String mapa = ""
+ "XXXXXXXXX X\n"
+ "          \n"
+ " X XXXXX  \n"
+ " XXXXXXXX \n"
+ " XXXXXXXX \n";
```

Figura 21. Mapa inicial

Para a criação do agente é necessário conhecer o ambiente em que ele será inserido. Na criação e instancia do agente, são passados como parâmetros a matriz e o número de linhas e colunas do mapa. Para iniciar o movimento do agente foi criado a função “moverAg (posXIni, posYIni, posXFim, posYFim)” em que são passadas as posições X e Y (posXIni, posYIni) iniciais do agente e as posições de destino (posXFim, posYFim), como demonstra a Figura 22.

```
AgenteSab ag = new AgenteSab(matriz, tCol, tLin);
ag.moverAg(2, 2, 9, 4);
```

Figura 22. Criação do agente e definição dos pontos de origem e destino

Para a escolha das possíveis casas de destino primeiramente é verificado se existe algum vizinho na posição em que o agente se encontra. Caso existe, verifica se este vizinho não é um obstáculo ou se este já não foi percorrido. As possíveis casas são armazenadas em um array de possíveis destinos, como apresentado no trecho de código abaixo:

```
while (!achou) {
    ArrayList<Integer> camX = new ArrayList<>();
    ArrayList<Integer> camY = new ArrayList<>();
    if (posY > 0) {
        char temp = matrizMapa[posX][posY - 1];
        char tempMemo = matrizMemoria[posX][posY - 1];
        if (temp == ' ' && tempMemo != '-') {
            camX.add(posX);
            camY.add(posY - 1);
        }
    }
}
```

Figura 23. Analise de possíveis casas de destino

Após conhecer os vizinhos que podem ser caminhados é verificado qual deles possui menor distância com o ponto de destino, este será a próxima casa escolhida pelo agente.

```

posXant = posX;
posYant = posY;
double proxPeso = Integer.MAX_VALUE;
for (int v = 0; v < camX.size(); v++) {
    double peso = Math.sqrt(Math.pow((posXfim - camX.get(v)), 2) + Math.pow((posYfim - camY.get(v)), 2));
    if (camX.get(v) == posXfim && camY.get(v) == posYfim) {
        achou = true;
        posX = posXfim;
        posY = posYfim;
    } else if (peso < proxPeso) {
        proxPeso = peso;
        posX = camX.get(v);
        posY = camY.get(v);
    }
}

```

Figura 24. Escolha do proximo passo do agente

Desta maneira é possível encontrar um caminho ao destino de uma maneira mais eficiente que se o agente estiver caminhando a esmo no ambiente, pois a cada passo o agente se encontra mais próximo do destino. A Figura 25 demonstra o caminho percorrido pelo agente do ponto de origem (Escada), ao destino (COTED).

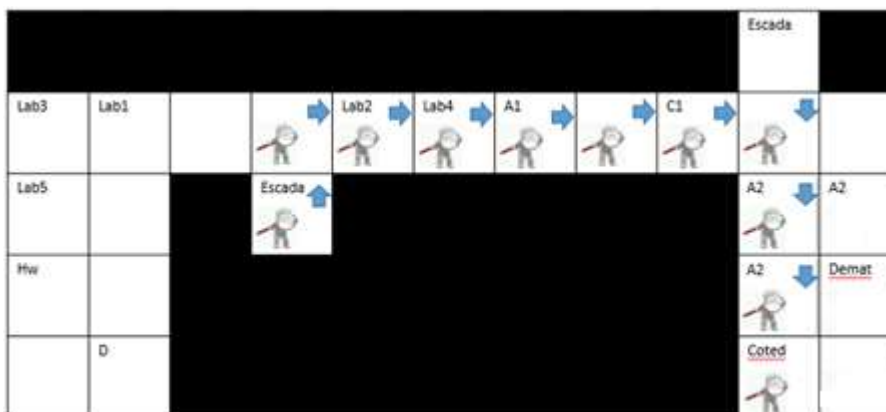


Figura 25. Caminho gerado pelo agente no mapa até chegarem ao destino

7 VERSÃO CONSOLIDADA DAS APLICAÇÕES

Como objetivo para a resolução do problema foram desenvolvidas três aplicações sendo duas *desktop* e uma *mobile*. Numa das aplicações *desktop* é feita a criação do projeto para a futura navegação e, na outra aplicação *desktop* pode se realizar a navegação. Na aplicação *mobile* só é possível realizar a navegação em um projeto criado na aplicação *desktop*, ou seja, não é possível criar um projeto na aplicação *mobile*.

7.1 AMBIENTE DE DESENVOLVIMENTO

Nesta seção estão descritos os ambientes utilizados para o desenvolvimento de cada uma das aplicações.

7.1.1 Aplicação *Desktop*

A aplicação para a plataforma Windows é desenvolvida na linguagem C# utilizando o Microsoft Visual Studio 2012 que é um pacote de programas da Microsoft para o desenvolvimento de software.

O Microsoft Visual Studio é especialmente dedicado ao .NET Framework e às linguagens Visual Basic, C, C++, C# e J#. Também é um grande produto de desenvolvimento na área web, usando a plataforma do ASP.NET. As linguagens com maior frequência nessa plataforma são: VB.NET (Visual Basic.Net) e o C#. Assim como o Windows ele também é um produto comercial com preços variados para cada um de suas versões, sendo grátis a com o menor número de recursos. O aplicativo desenvolvido para a plataforma Windows na linguagem C# foi-se utilizado o Microsoft Visual Studio 2012 lançado no final do ano de 2012 (SOARES, 2013).

Já a linguagem C# é orientada a objetos, fortemente tipada, desenvolvida pela Microsoft como parte da plataforma .NET. A sua sintaxe foi baseada no C++ mas inclui muitas influências de outras linguagens de programação, como Object Pascal e Java (BASIURA et al, 2003).

7.1.2 Aplicação Mobile

Para a plataforma *mobile* é desenvolvido um sistema baseado no sistema operacional Android, para isso, é utilizado o Eclipse IDE com built-in ADT (*Android Developer Tools*) e uma biblioteca de API e ferramentas de desenvolvedor Android, teste e depuração de aplicativos que podem ser obtidos com o Android SDK.

7.1.2.1 Eclipse IDE

O Eclipse é uma IDE (*integrated development environment*) Java open-source com suporte a outras linguagens de programação a partir do download de plug-ins para C/C++, PHP, ColdFusion, Python, Android, etc.

O projeto Eclipse é formado por uma comunidade de indivíduos e organizações que desejam colaborar em software open-source comercialmente amigável. Seus projetos são focados na construção de uma plataforma de desenvolvimento aberta composta de estruturas extensíveis, ferramentas e tempos de execução para a construção, implantação e gerenciamento de software em todo o ciclo de vida (Eclipse, 2014).

7.1.2.2 Ambiente Android

Android é o sistema operacional móvel da Google, desenvolvido pela Google Inc. com parceria com Open Handset, e fornece uma estrutura rica em recursos dos dispositivos que lhe permite criar aplicativos inovadores e jogos para dispositivos móveis em um ambiente de linguagem Java (Android Developer, 2014).

Aplicativos Android são escritos na linguagem de programação Java com as bibliotecas de recursos de cada dispositivo. As ferramentas do SDK compilam o código, juntamente com todos os arquivos de dados e recursos, em um arquivo compilado “.apk” que se trata de um pacote Android, no formato de um arquivo que contém todo o conteúdo de um aplicativo para Android e este é o arquivo que os dispositivos com o sistema operacional utilizam para instalar o aplicativo (Android Developer, 2014).

O sistema operacional Android é um sistema multiusuário do Linux, em que cada aplicativo é um usuário diferente.

Por padrão, o sistema atribui a cada aplicativo uma ID de usuário exclusivo Linux (o ID é usado apenas pelo sistema e é desconhecido pelo aplicativo). O sistema define as permissões para todos os arquivos de modo que apenas o ID de usuário atribuído a esse aplicativo pode acessá-los.

Cada processo tem a sua própria máquina virtual e o código de um aplicativo é executado isoladamente de outros.

Por padrão, cada aplicativo é executado em seu próprio processo Linux. Android inicia o processo quando qualquer um dos componentes do aplicativo precisa ser executado, em seguida, desliga o processo quando ele não é mais necessário, ou quando o sistema deve recuperar a memória para outros aplicativos.

7.2 ESTUDO DE CASO

Para explicar o funcionamento da aplicação *desktop* será utilizada uma representação, do piso térreo do Bloco C da Universidade Tecnológica Federal do Paraná, campus de Ponta Grossa. A Figura 26 mostra uma imagem de satélite do campus, com seus limites identificados por uma linha vermelha, e o bloco C delimitado por linhas amarelas.



Figura 26. Campus UTFPR Ponta Grossa

Essa representação foi inicialmente um rascunho em um pedaço de papel, nele foi desenhado a mão o formato do local e demarcado as portas que neste caso são os nossos pontos de interesse. Juntamente com isso foi também feita uma legenda, indicando o que cada porta significa. Esse rascunho foi feito no mês de agosto do ano de 2013. Logo após esse rascunho pronto, copiamos isso para a forma digital usando o programa Paintbrush, e como resultado pode ser visto na Figura 27.

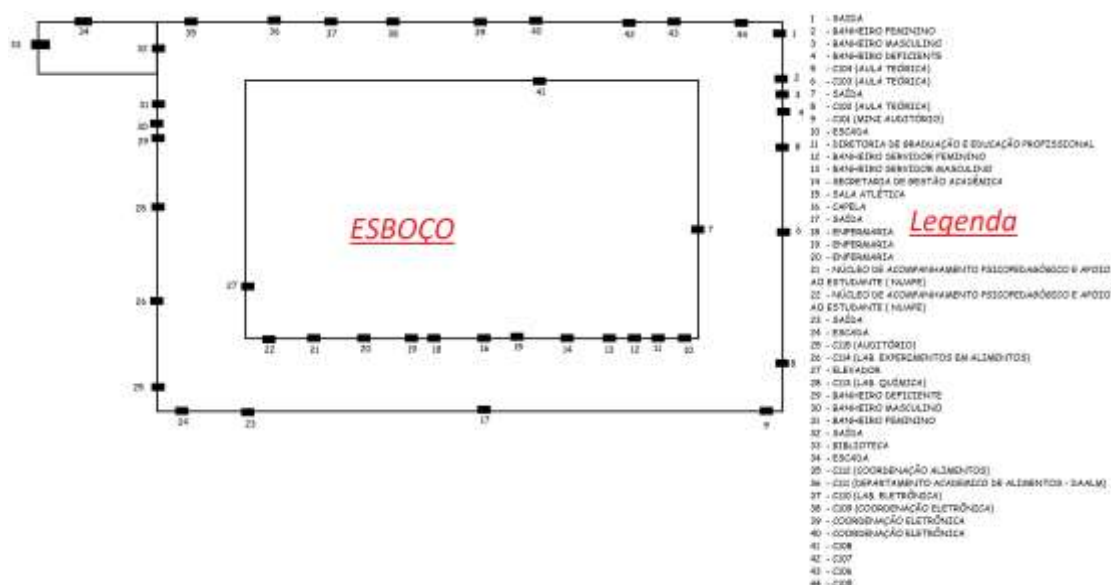


Figura 277. Imagem final bloco C

7.3 APLICAÇÕES *DESKTOP*

Como já dito anteriormente, existem duas aplicações *desktop*, uma que é utilizada para gerar o mapa virtual e a outra que é utilizada para navegar. Nos tópicos abaixo serão descritos o funcionamento de cada uma delas.

7.3.1 Criando um novo projeto – Usuário Administrador

Para criar um projeto utilizaremos como exemplo o esboço do bloco C mostrado na Figura 27. Assim que selecionada a opção “Criar um novo projeto” irá abrir uma tela na qual deverá ser selecionado o arquivo de imagem de planta baixa ou similar a qual se deseja criar um projeto. A Figura 28 mostra a tela de abrir a imagem, onde foi selecionada a imagem chamada BLOCO C1.jpg.

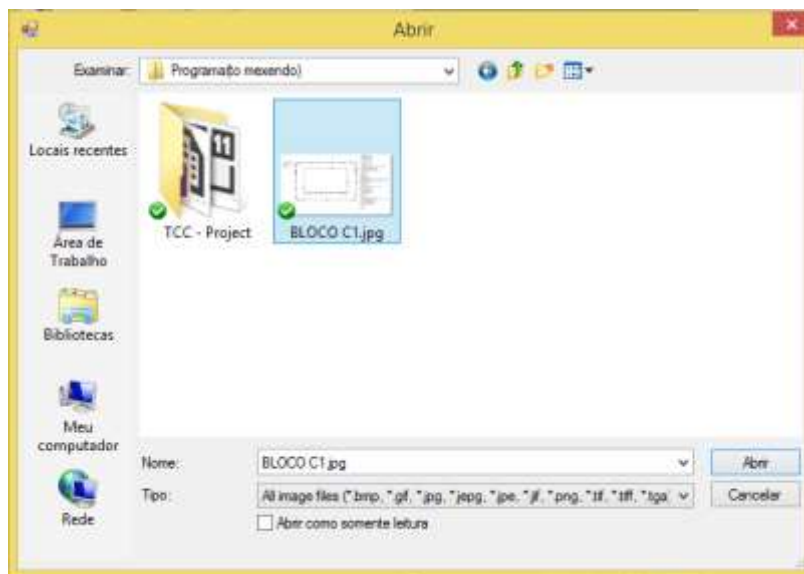


Figura 28. Tela abrir imagem

Após aberta a imagem aparece a tela da Figura 29. Nessa tela deve ser ampliada a imagem da planta para posterior indicação dos pontos e dos possíveis caminhos.

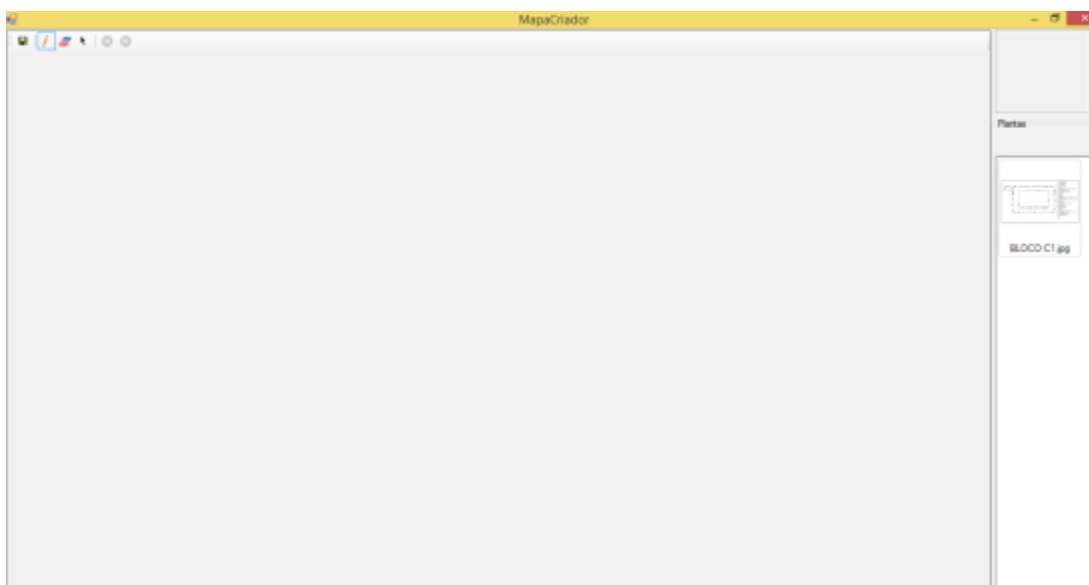


Figura 29. Tela Principal MapaCriador

Assim clica-se no menu da direita que contém uma miniatura da imagem escolhida anteriormente, logo que selecionado ela será exibida no espaço vazio na cor cinza na tela. A partir deste ponto será iniciada a indicação de onde poderá caminhar, com um clique insere-se um ponto, este ponto será o inicial para se criar

uma cadeia de linhas, que irão identificar os caminhos possíveis. Na Figura 30 é inserido um desses pontos em um canto de um corredor do bloco C.

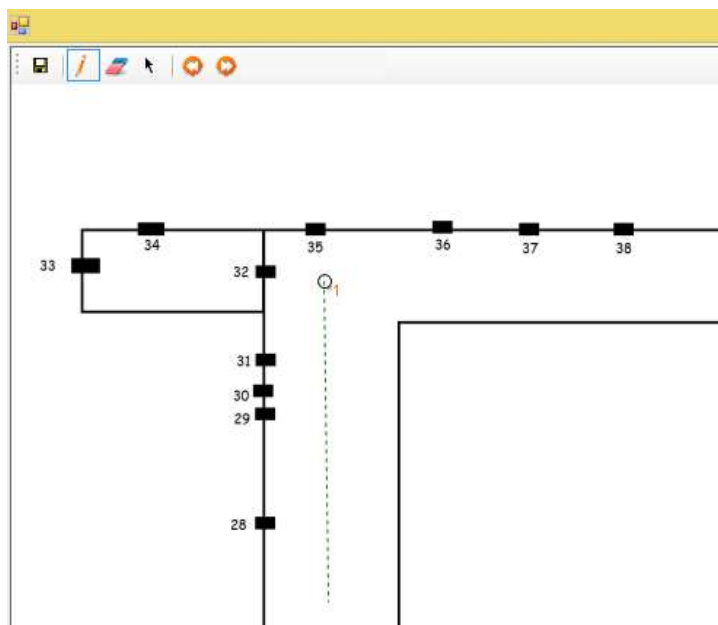


Figura 280. Inserindo locais caminháveis

Após indicados todos os locais caminháveis, a imagem deve ficar parecida com a Figura 31, nela foram necessários quatro pontos para que fossem desenhados todos os caminhos possíveis.

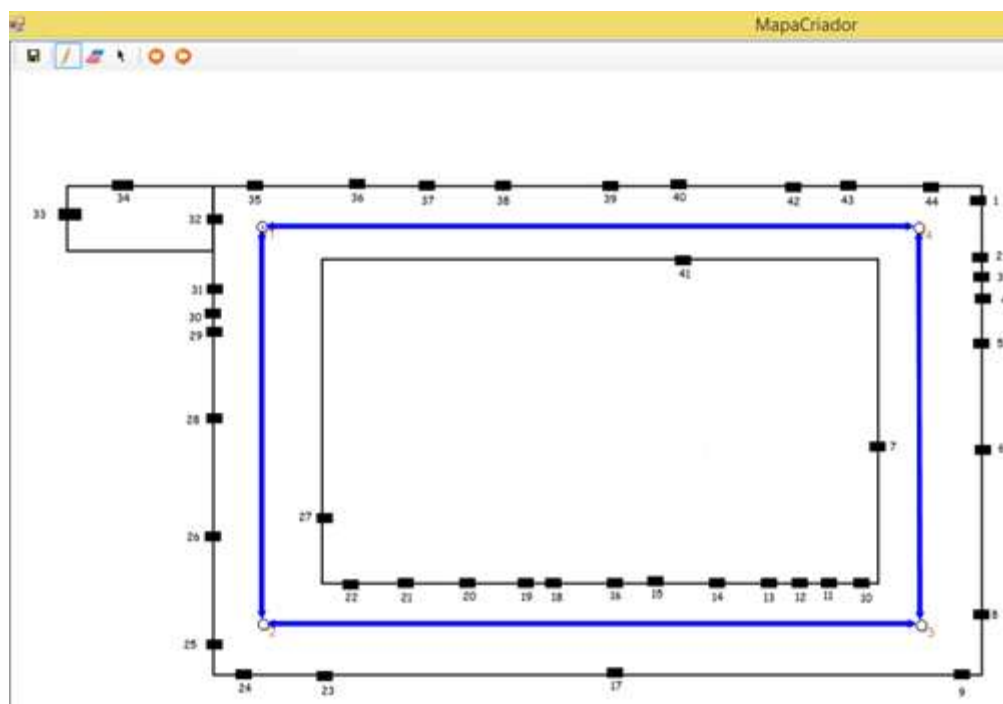


Figura 291. Locais caminháveis inseridos

Agora deve clicar na seta vermelha indicando no canto superior esquerdo da tela, e começar a selecionar os pontos de interesse, no estudo de caso cada retângulo preto indica um ponto de interesse, começamos pelo ponto de interesse 1, segundo a legenda 1 é uma “Saída”, assim que clicarmos com o botão da esquerda do mouse no retângulo preto irá abrir uma tela como mostra a Figura 32. Nesta tela, se deve inserir um código para este ponto, uma nomenclatura e uma descrição, que neste caso foram “S01”, “SAIDA01” e “Saída 01” respectivamente. Pode-se observar que uma linha amarela ligou-se automaticamente a um local possível de se caminhar anteriormente indicado. Depois de tudo indicado deve-se clicar no botão “OK”.

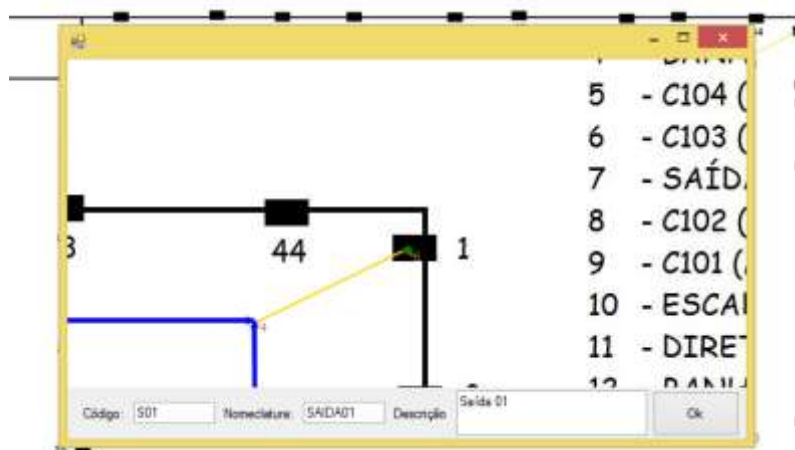


Figura 302. Inserindo ponto de interesse

Assim que inserido o ponto de interesse ele irá aparecer com a sua nomenclatura em cima da imagem e uma linha na cor amarela ligando o mesmo a um local que pode-se caminhar, como ilustra a Figura 33.

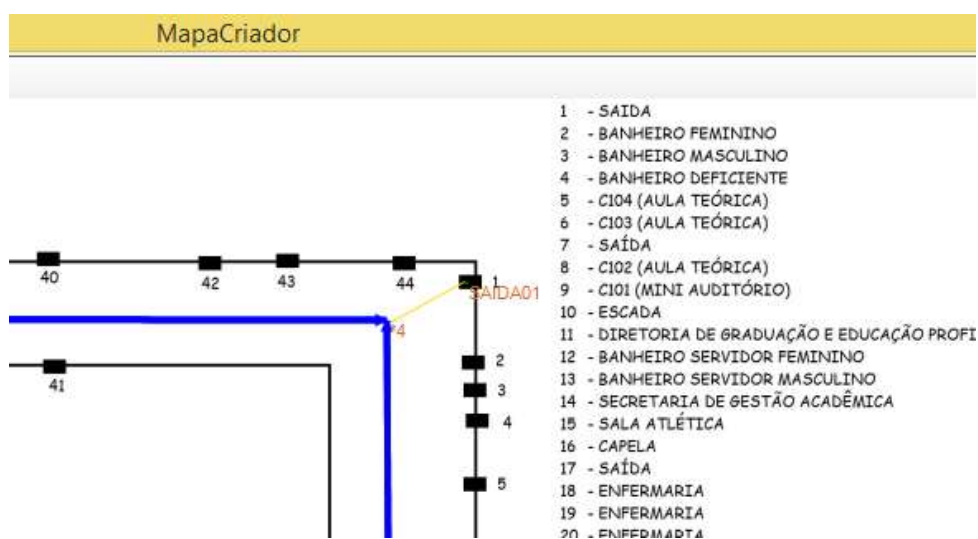


Figura 313. Ponto de interesse inserido

Deste modo devem ser inseridos todos os pontos de interesse que existirem na planta. No estudo de caso utilizado como exemplo foram inseridos todos os pontos e a Figura 34 ilustra como ficou.

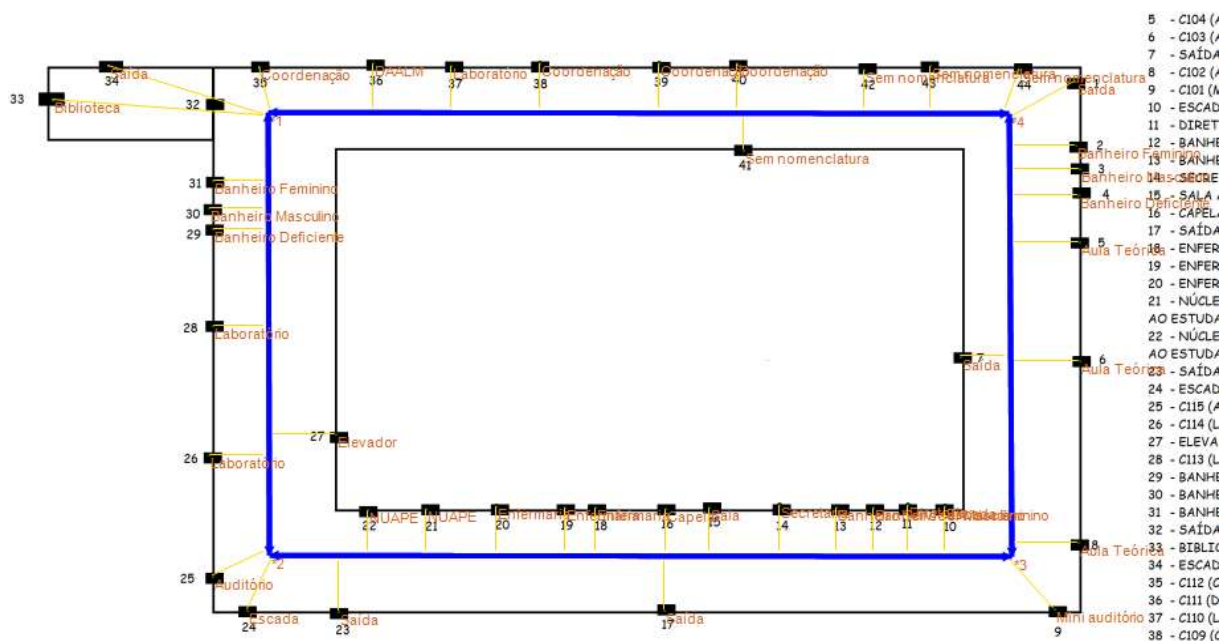


Figura 324. Pontos de interesse inseridos

Finalizada a inserção dos pontos de interesse, está feita a criação do projeto. A opção de salvar está demonstrada na Figura 35 demarcado com um retângulo vermelho em volta de um disquete.

Assim que salvo, será gerado um arquivo no formato .XML que ficará no mesmo caminho que a imagem utilizada está. Este arquivo será usado no sistema de navegação, nele contém os dados dos pontos e caminhos.

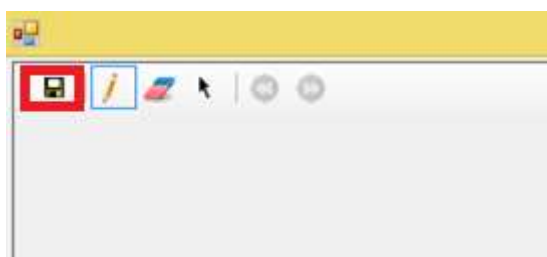


Figura 335. Ícone salvar.

7.3.2 Utilizando O Sistema De Navegação

Com o arquivo no formato .XML gerado, será utilizado o sistema para busca caminhos. Primeiramente devemos abrir a aplicação, irá aparecer a tela ilustrada na Figura 36, que indica que devemos primeiro selecionar o arquivo de imagem e depois o arquivo .XML.

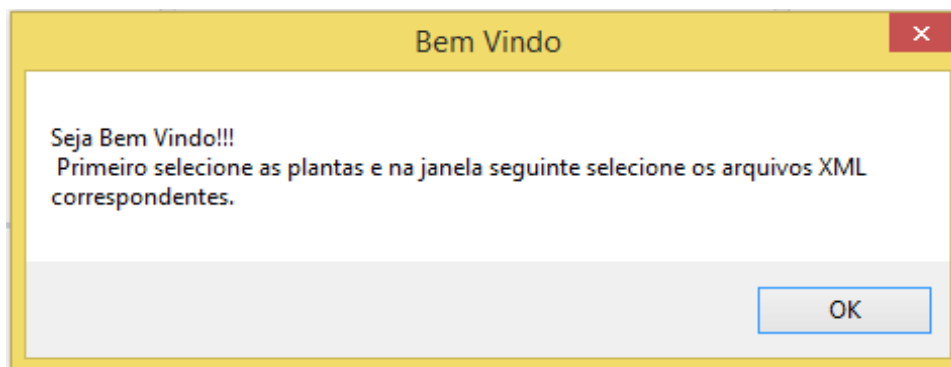


Figura 346. Mensagem de Bem Vindo

Na figura 37 está a tela para abertura de um arquivo de imagem, o arquivo deve ser o mesmo que foi utilizado para a criação do projeto no item 7.2.1.

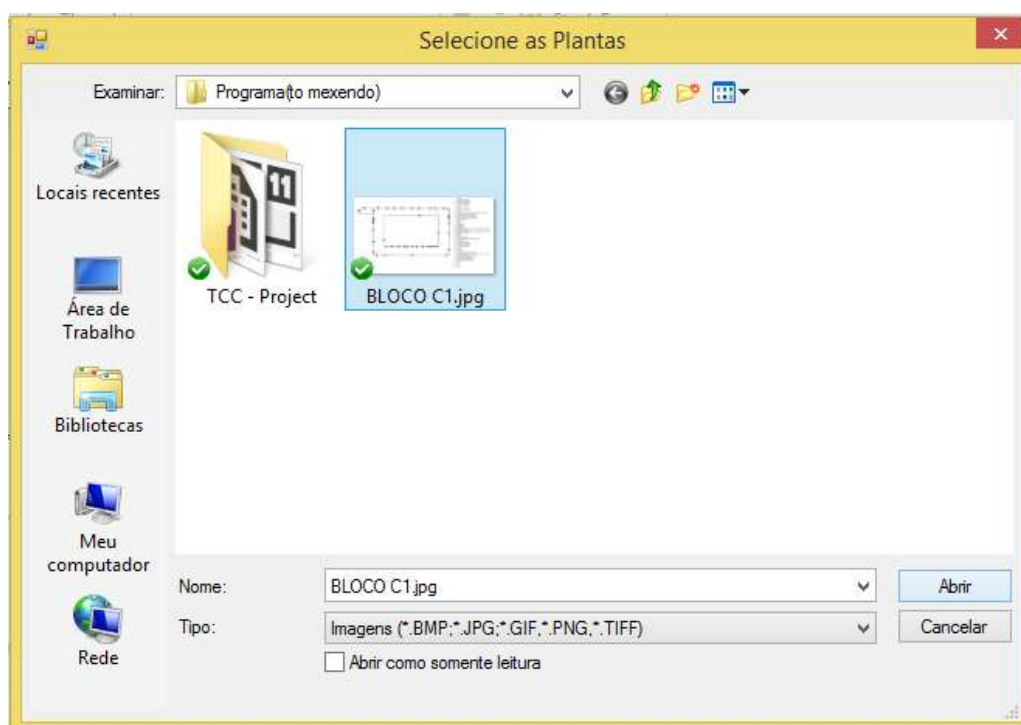


Figura 37. Selecionar planta

Imediatamente irá abrir uma tela semelhante, figura 38, porém agora deverá ser aberto um arquivo com a extensão .XML. O arquivo criado com o procedimento explicado na seção 7.2.1 este arquivo tem o mesmo nome da imagem e está no formato XML.

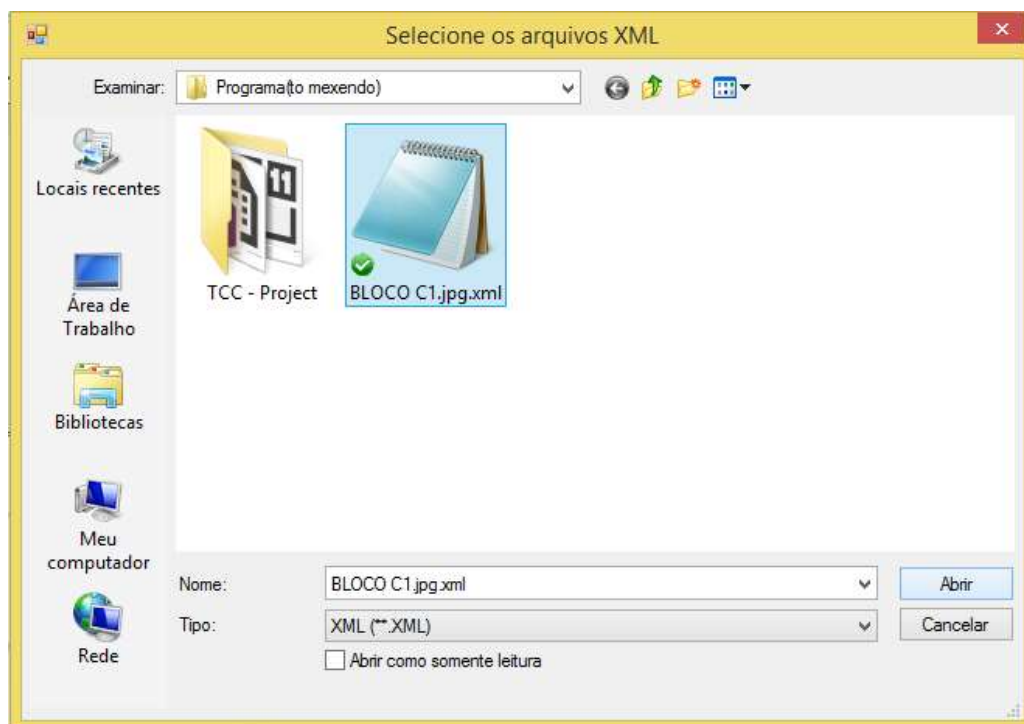


Figura 38. Selecionar arquivo XML

Deste modo se os arquivos carregados foram os corretos o sistema exibe uma tela com duas *combobox*, um botão no canto direito superior e logo mais abaixo uma miniatura do arquivo de imagem selecionado (Figura 39). A imagem selecionada será projetada no resto da tela e círculos indicarão os pontos de interesse.

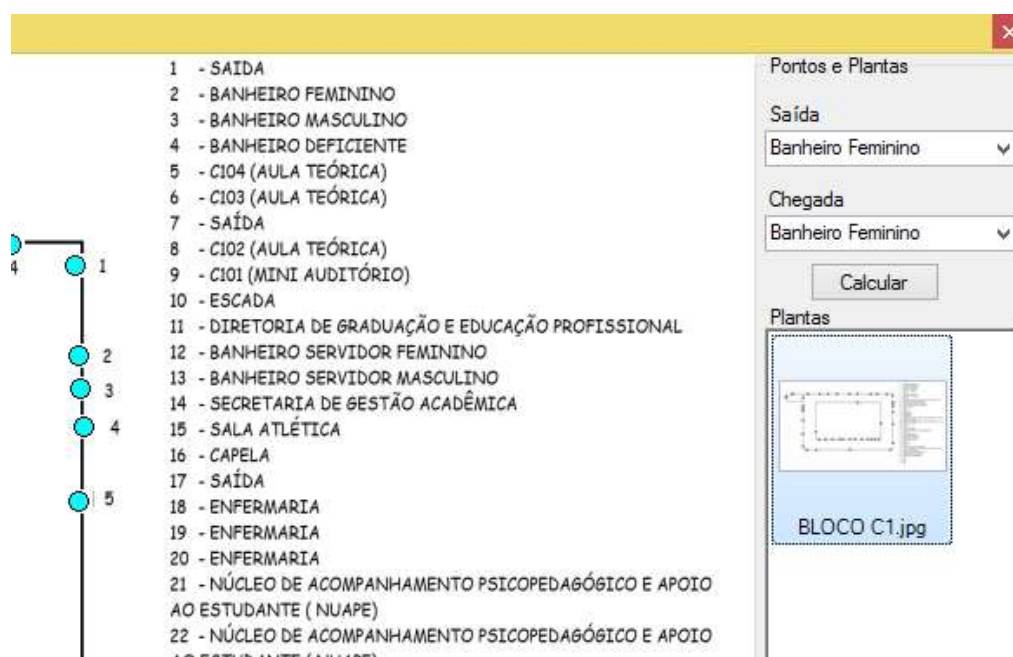


Figura 39. Tela principal

Agora iremos iniciar uma busca de um ponto de interesse a outro. Primeiramente devemos selecionar um item na *combobox* de saída, nela estarão listados todos os pontos de interesse por sua nomenclatura. No exemplo foi selecionada a “Capela” como ponto de saída e o “DAALM” como ponto de chegada, Figura 40.

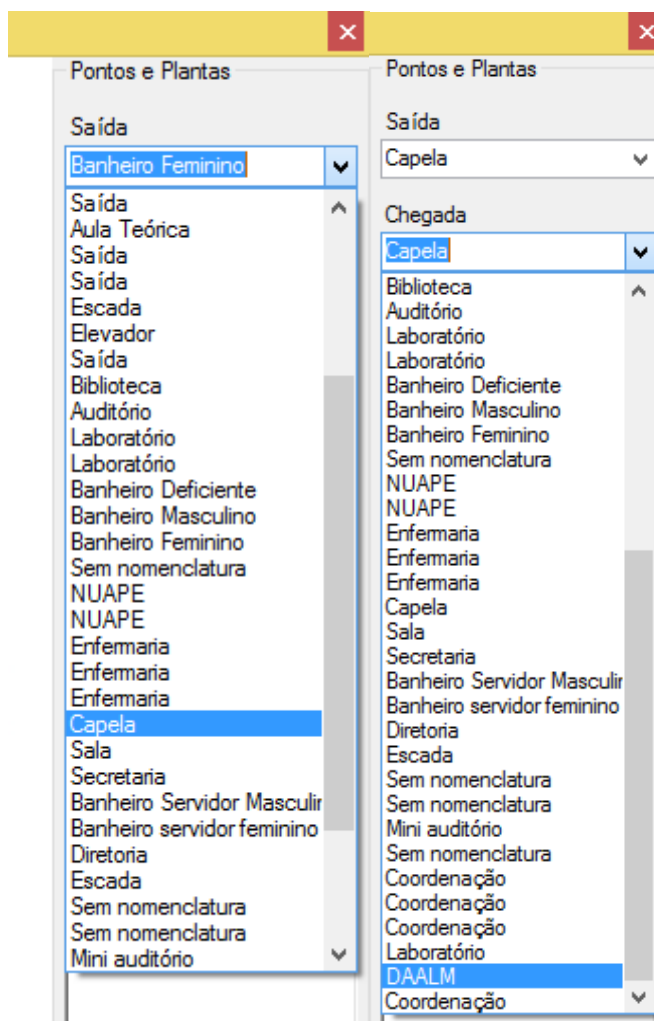


Figura 350. Selecionando saída e entrada

Com os pontos de interesse de saída e chegada definidos deve-se clicar no botão “Calcular” localizado abaixo da *combobox*. Ele irá procurar o menor caminho utilizando o algoritmo de Dijkstra.

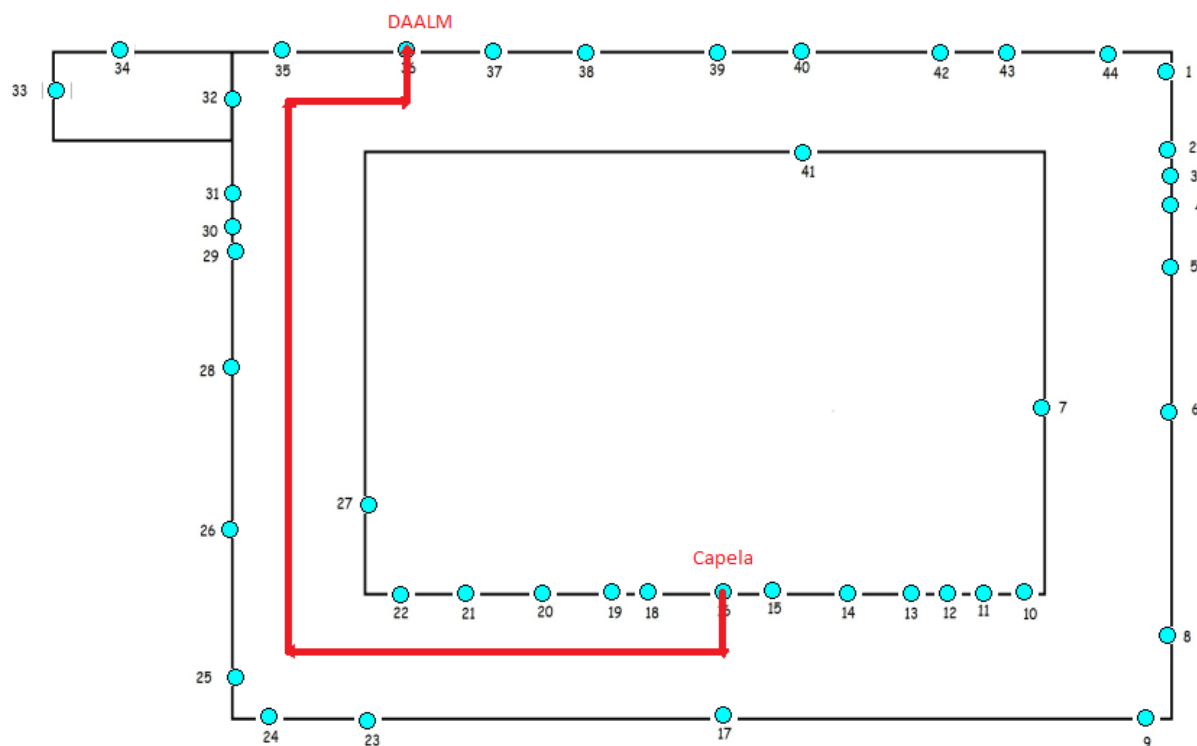


Figura 361. Exemplo

Como pode-se observar na Figura 41, foi criado um caminho do ponto 16 "Capela" ao ponto 36 "DAALM" levando em consideração os pontos de interesse e caminhos possíveis, indicados na criação do projeto no item 7.2.1.

Do mesmo modo como apresentado aqui, quaisquer dois pontos podem ser selecionados pelo usuário e sem a necessidade de conexão com internet ou GPS será feito o cálculo do caminho a ser percorrido da origem ao destino.

Importante ressaltar que a correção das informações, da localização e da planta dependem da configuração inicial, fato que pode ser trabalhado futuramente com um algoritmo adaptável que armazene novos nomes ou locais quando visitados por usuários, similar ao que ocorre com o Foursquare (TECHCRUNCH, 2010), por exemplo.

7.4 APLICAÇÃO *MOBILE*

A aplicação *mobile* possibilita a qualquer usuário com um dispositivo Android uma maneira fácil e portátil de busca e navegação em qualquer ambiente previamente definido em um projeto. Para isto ele depende apenas de baixar os arquivos de imagem e XML com os possíveis pontos de interesse para não haver mais

necessidade de outra comunicação externa, disponibilizando o acesso ao conteúdo em qualquer lugar.

7.4.1 Executando o Aplicativo

O pacote com o aplicativo android (arquivo .apk) será disponibilizado no google play store (<https://play.google.com/store>) para download direto no celular. O mapa com a planta baixa e os arquivos de navegação podem ser disponibilizados em uma página da web para download ou algum outro método de transferência de dados como bluetooth, NFC (Near Field Communication), etc. Estes arquivos baixados devem ser movidos para pasta “Navegador” que é gerada pela aplicação em sua primeira execução.

Após gerada a planta baixa e os arquivos de navegação na aplicação *desktop*, os arquivos podem ser carregados em dispositivos *mobile* com o Sistema Operacional Android conforme o esquema apresentado na Figura 42.

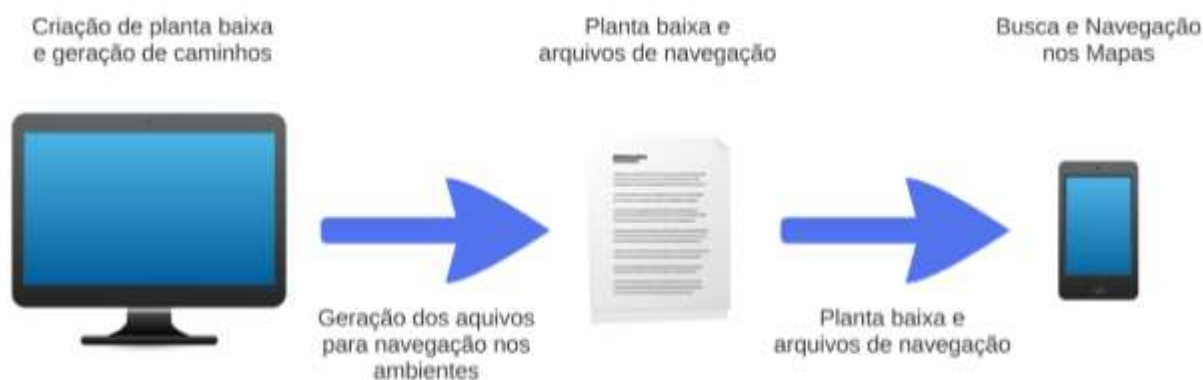


Figura 372. Entrada de dados para aplicação mobile

As plantas baixas e os arquivos de navegação são armazenados em uma pasta padrão (“Navegador”) que será utilizada pela aplicação. Para que o usuário possa escolher em qual planta baixa ele deseja navegar, para isso, será exibida uma lista das plantas que ele possui, como demonstrado na Figura 43.

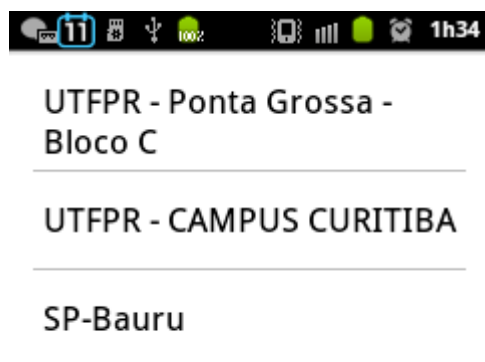


Figura 383. Lista de plantas baixas para pesquisa de caminhos

Os possíveis pontos de origem (A) e destino (B) são exibidos possibilitando ao usuário efetuar a busca, Figura 44.



Figura 394. Tela para seleção de origem e destino

Quando a busca é solicitada a aplicação executa o algoritmo de Dijkstra no grafo da planta para encontrar o caminho mais curto partindo da origem até encontrar o destino, se este caminho for encontrado a aplicação exibe a tela com a o passo a passo do caminho encontrado, Figura 45 e a tela com a planta com caminho Figura 46.

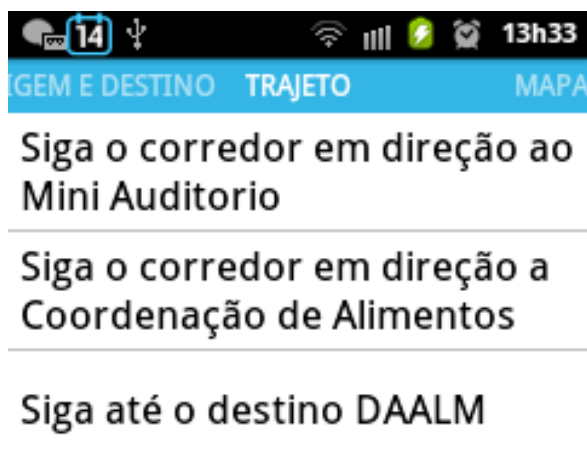


Figura 405. Tela de Exibição do passo a passo até o destino.

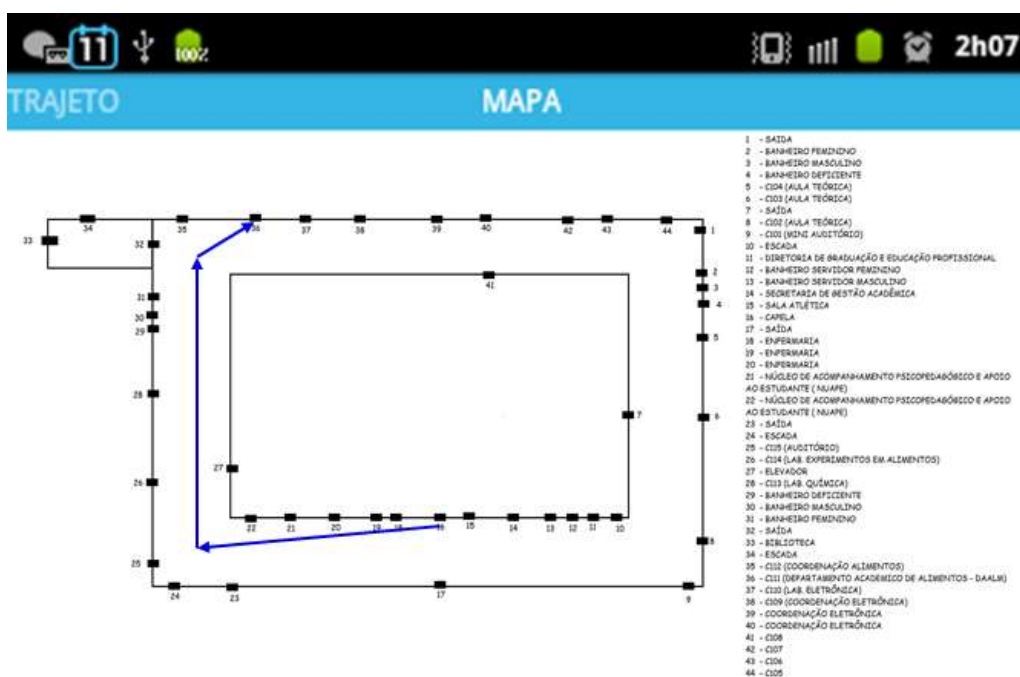


Figura 416. Tela de Exibição do caminho encontrado

O usuário do aplicativo ao pressionar 2 vezes em certo lugar da planta, será realizado um Zoom na posição que foi pressionada, como demonstrado na Figura 47.

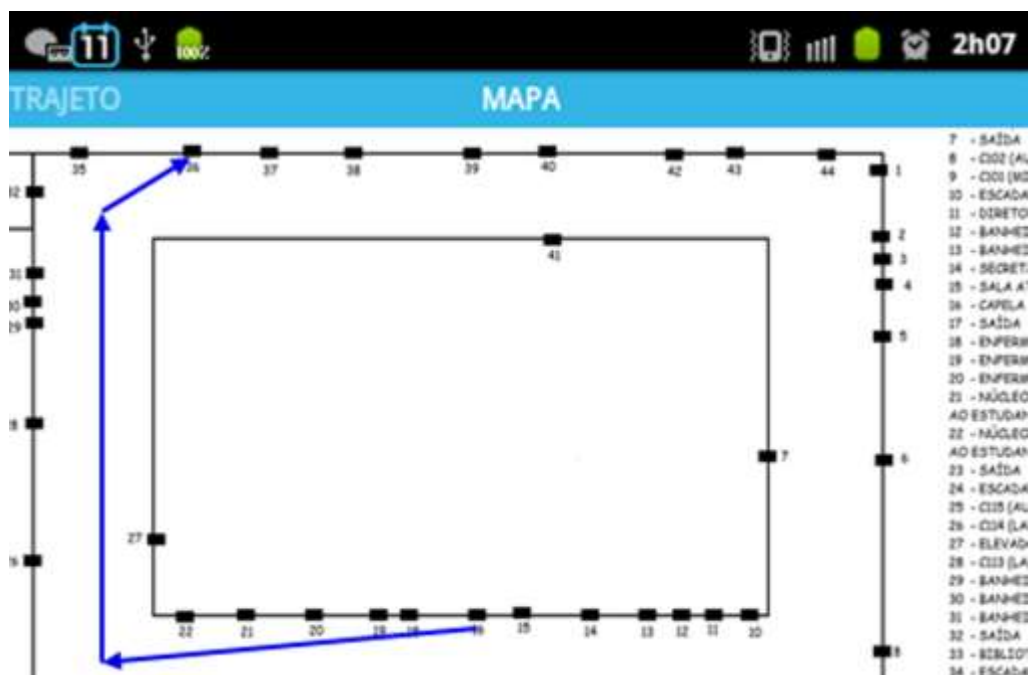


Figura 47. Tela de exibição do caminho encontrado com Zoom

Caso não seja encontrado o caminho, é apresentada a mensagem “Não foi possível encontrar um caminho ao destino selecionado!”.

8 CONCLUSÃO

Este capítulo apresenta as considerações finais sobre o estudo, confrontando os objetivos gerais e específicos com os resultados obtidos através dos experimentos realizados.

O objetivo geral de criar um sistema de localização em ambientes fechados que facilitasse a vida de pessoas no seu cotidiano e que fosse adaptável a inúmeros casos, foi alcançado com sucesso. Assim foram desenvolvidas duas aplicações sendo uma para dispositivo móvel e outra *desktop*, onde em ambas o usuário pode fazer a busca de caminhos entre pontos de interesse.

Durante o desenvolvimento do trabalho, foram necessárias algumas mudanças para que se tornasse viável o desenvolvimento das aplicações, já que ambas partiram do zero e em linguagens diferentes dificultam o reuso do código. Alguns requisitos tiveram que ser descartados pelo fato do nível de complexidade ser alto demandando muito tempo para o desenvolvimento.

Quanto ao objetivo específico de estudar conceitos e trabalhos relacionados ao tempo, foi cumprido com sucesso, apesar que no começo do trabalho trabalhos relacionados eram muito poucos e muitas das vezes eram relacionados só em um aspecto, mais com o passar do tempo enquanto o trabalho era desenvolvido foram lançados aplicativos no mercado que tinham a mesma finalidade proposta neste trabalho.

Os objetivos específicos de busca, estudar, adaptar plantas baixas para a utilização nas aplicações, desenvolver aplicações adaptáveis a qualquer planta e que sejam úteis para a sociedade também foi cumprido com sucesso, já que pode-se usar a aplicação utilizando-se de uma planta baixa sem nenhuma alteração tanto como de um esboço feito a partir de uma mera observação ocular do local.

Já os objetivos específicos de avaliar e documentar as aplicações e documentar os resultados, foram cumpridos parcialmente. A avaliação das aplicações não foi possível ser feita pelo motivo que as aplicações não ficaram prontas com uma sobra de tempo suficiente para serem instaladas em um local para a utilização de um grupo de pessoas. Porém a documentação das aplicações foi feita e está descrita neste trabalho.

Assim é possível concluir que esse trabalho cumpriu com a maioria dos objetivos proposto, além de ser considerado uns dos primeiros trabalhos acadêmicos

a tratar de um sistema para criação e navegação em ambientes com um grande fluxo de usuários.

As aplicações podem ser utilizadas experimentalmente na UTFPR seja através de totens ou mesmo disponível aos alunos e visitantes.

8.1 TRABALHOS FUTUROS

Com base nos resultados obtidos, algumas sugestões para trabalhos futuros podem ser listadas, por exemplo, a abordagem de algoritmos que acumulem novos locais e nomes dos pontos de interesse, facilitando a navegação, atingindo um número maior de pessoas e se adequando a modificações de nomes dos locais, como ocorreu durante este projeto na UTFPR.

Realizar experimentos práticos e avaliar a facilidade de uso e a opinião dos usuários, uma vez que a busca de salas ou departamentos e a navegação em locais de médio porte é alvo de reclamações frequentes.

Realizar testes de robustez e verificar quais as restrições do sistema quanto ao tamanho e formato dos dados de entrada e dos dispositivos móveis, além de dependência de escala.

Adaptar o sistema para que o mesmo facilite a busca de caminhos adequados a pessoas com restrições físicas e para que seja acessível.

REFERÊNCIAS

- ANDROID DEVELOPER, About Android. Disponível em: <<http://developer.android.com/about/index.html>>. Acesso em 15 mar. 2014.
- APEK, (2013). **Apek TrackMall**. Disponível em <<http://www.apek.com.br/trackmall-diretorio-virtual-shoppings.html>> Acesso em: 14 jul. 2014.
- BAHL, P.; PADMANABHAN, V. N. RADAR: An In-Building RF-based User Location. **Proceedings of the IEEE Infocom**, p. 775–784, 2000.
- BASIURA, R. et al. (2003). **Professional ASP.NET, Web Services**. São Paulo, Pearson.
- BAUCH, A. Galileo enters the next stage. **Physikalisch-Technischen Bundesanstalt**, 2005. Disponível em: <<http://www.ptb.de/cms/index.php?id=2504&L=1>>. Acesso em: 20 Setembro 2013.
- BIRCH, M. C. & HAYNES, K. F. 1982. **Insect pheromones**. Londres, Edward Arnold. 60 p.
- CHEN, Y.-T. et al. A RSSI-based Algorithm for Indoor Localization Using ZigBee in Wireless Sensor Network. **International Conference on Distributed Multimedia Systems**, p. 70-75, 2009.
- DEICU, L. E. **Planejadores de Caminho baseados em Campos Potenciais**. UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. RIO GRANDE DO SUL. 2012.
- DEITEL, H. M. et al. **C# Como Programar**. São Paulo: Pearson Education, 2003.
- DIAS, K. N.; STELLO, C. C. **Algoritmo de Dijkstra e Algoritmo de Kruskal**. Universidade Federal do Pampa Alegrete.
- FROZZA, R. **SIMULA - Ambiente Para Desenvolvimento de Sistemas Multiagentes Reativos**. Universidade Federal do Rio Grande do Sul. Porto Alegre. 1997.

GONZAGA, C.(1978). **Busca de Caminhos em Grafos e Aplicações**. I Reunião de Matemática Aplicada, IBM, Rio de Janeiro.

GONZAGA, C.(2005). **Notas de aula de Pesquisa Operacional**. Universidade Federal de Santa Catarina, Florianópolis.

GOOGLE (2013). **Android, the world's most popular mobile platform**. Disponível em <<http://developer.android.com/about/index.html>> Acesso em: 29 jun. 2013.

HARTER, A. et al. The Anatomy of a Context-Aware Application. **Wireless Networks**, Cambridge, v. 8, p. 187-197, 2002.

HUERTA, E.; MANGIATERRA, A.; NOGUERA, G. **GPS Posicionamiento Satelital**. 1nd. ed. Rosario: UNR Editora, 2005.

IBOPE. SHOPPING CENTERS. **Geonotícias**, maio. 2011. Disponível em: <<http://www4.ibope.com.br/download/shoppingcenters.pdf>>. Acesso em: 18 set. 2013.

JOKINEN, A. et al. GLONASS Aided GPS Ambiguity Fixed Precise Point Positioning. **Journal of Navigation**, Nottingham, v. 66, n. 03, p. 1469-7785, Maio 2013.

LECHETA, R. R. (2013). **Google Android - Aprenda a Criar Aplicações Para Dispositivos Móveis Com o Android Sdk**. Editora Novatec, 3ª Ed 2013.

MICROSOFT (2010). **Introduction to the C# Language and the .NET Framework, Visual Studio**. Disponível em <<http://msdn.microsoft.com/pt-br/library/vstudio/z1zx9t92%28v=vs.100%29.aspx>> Acesso em: 29 jun. 2013.

MICROSOFT. (2012). **Visual C#, Visual Studio**. Disponível em <<http://msdn.microsoft.com/pt-br/library/vstudio/kx37x362.aspx>> Acesso em: 27 jun. 2013.

MIU, A. K. L. **Design and Implementation of an Indoor Mobile Navigation System**. Massachusetts Institute Of Technology. Massachusetts, p. 60. 2002.

RABUSKE, M. A. (1992). **Introdução à teoria de grafos**, Editora da UFSC.ROSSETTO, D. R. (2007). **Algoritmos de Busca de Caminhos em Grafos**

aplicados aos problemas: Alinhamento de Proteínas e Quebra-cabeça de 15 peças. Universidade Federal de Santa Catarina, Florianópolis.

SANTANA, A. M. **Localização e Planejamento de Caminhos para um Robô Humanoide e um Robô Escravo com Rodas.** Universidade Federal do Rio Grande do Norte. Natal-RN, p. 31-37. 2007.

SOARES, E. E. D. M. **Localização INDOOR via KDE assinaturas de RSSI.** Universidade Federal do Rio de Janeiro. Rio de Janeiro, p. 65. 2013.

SOARES, H. N. (2013). **Desenvolvimento de Protótipo de Veículo Controlado por Realidade Virtual.** Universidade Federal de Ouro Preto.

STUART J. RUSSELL, P. N. **Artificial Intelligence: A Modern Approach.** New Jersey, USA: Alan Apt, 1995.

TAMASHIRO, E. C. **Administração do Tempo Gerenciando o Tempo Como Aliado.** Universidade Candido Mendes. Rio de Janeiro, p. 45. 2003.

TECHCRUNCH, (2010). **Foursquare now 3 million strong.** Disponível em <<http://techcrunch.com/2010/08/29/foursquare-now-3-million-strong>> Acesso em: 27 jun. 2013.

WERNER, E.; CASTELFRANCHI, C. **Artificial Social Systems.** San Martino al Cimino, Itália: Springer-Verlag Berlin Heidelberg, v. I, 1992.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent Agents: Theory and Practice. **Knowledge Engineering Review**, Manchester, 1995. 115-152.

ZANINI, E. **Como administrar melhor seu tempo.** 1º. ed. São Paulo: Biblioteca 24 horas, 2010.

ZOU, W. **China Military Online**, 2010. Disponível em: <http://eng.chinamil.com.cn/news-channels/china-military-news/2010-05/20/content_4222569.html>. Acesso em: 20 Setembro 2013.

BERLINER, H. **The B* Tree Search Algorithm. A Best-First Proof Procedure**, 1979.

SHEPPARD, B. **World-championship-caliber Scrabble**, 2002.

APÊNDICE A - Arquivo gerado com os dados do Mapa Virtual pela aplicação *desktop*:
BLOCO C1.xml

```
<?xml version="1.0" encoding="utf-8"?><obj_Planta Nome="Bloco C1"
Descricao="Bloco C1 UTFPR Ponta Grossa" nomePlanta="BLOCO
C1.jpg"><graphNodes><node nome="" abreviatura="*1" descricao=""
point="{X=622,Y=375}" raioPonto="4" tipo="Interseccao" /><node nome=""
abreviatura="*2" descricao="" point="{X=625,Y=1331}" raioPonto="4"
tipo="Interseccao" /><node nome="" abreviatura="*3" descricao=""
point="{X=2248,Y=1334}" raioPonto="4" tipo="Interseccao" /><node nome=""
abreviatura="*4" descricao="" point="{X=2242,Y=377}" raioPonto="4"
tipo="Interseccao" /><node nome="BF02" abreviatura="Banheiro Feminino"
descricao="Banheiro Feminino 02" point="{X=2387,Y=445}" raioPonto="4"
tipo="PontoReferencia" /><node nome="BM03" abreviatura="Banheiro Masculino"
descricao="Banheiro Masculino 03" point="{X=2396,Y=496}" raioPonto="4"
tipo="PontoReferencia" /><node nome="BD03" abreviatura="Banheiro Deficiente"
descricao="Banheiro Deficiente 04" point="{X=2393,Y=552}" raioPonto="4"
tipo="PontoReferencia" /><node nome="C104" abreviatura="Aula Teórica"
descricao="C104 - Sala de aula" point="{X=2396,Y=654}" raioPonto="4"
tipo="PontoReferencia" /><node nome="C103" abreviatura="Aula Teórica"
descricao="C103 - Sala de aula" point="{X=2396,Y=911}" raioPonto="4"
tipo="PontoReferencia" /><node nome="SD01" abreviatura="Saída"
descricao="Saída 01" point="{X=2387,Y=307}" raioPonto="4" tipo="PontoReferencia"
/><node nome="SD07" abreviatura="Saída" descricao="Saída 07"
point="{X=2135,Y=902}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C102" abreviatura="Aula Teórica" descricao="C102 - Sala de aula"
point="{X=2393,Y=1303}" raioPonto="4" tipo="PontoReferencia" /><node
nome="SD17" abreviatura="Saída" descricao="Saída 17" point="{X=1487,Y=1449}"
raioPonto="4" tipo="PontoReferencia" /><node nome="SD23" abreviatura="Saída"
descricao="Saída 23" point="{X=775,Y=1452}" raioPonto="4" tipo="PontoReferencia"
/><node nome="ES24" abreviatura="Escada" descricao="Escada 24"
point="{X=575,Y=1449}" raioPonto="4" tipo="PontoReferencia" /><node
nome="EL27" abreviatura="Elevador" descricao="Elevador 27"
point="{X=769,Y=1069}" raioPonto="4" tipo="PontoReferencia" /><node
nome="SD34" abreviatura="Saída" descricao="Saída 34" point="{X=272,Y=273}"
raioPonto="4" tipo="PontoReferencia" /><node nome="BB33"
abreviatura="Biblioteca" descricao="Biblioteca 33" point="{X=150,Y=346}"
raioPonto="4" tipo="PontoReferencia" /><node nome="C115" abreviatura="Auditório"
descricao="C115 - Auditório" point="{X=500,Y=1376}" raioPonto="4"
tipo="PontoReferencia" /><node nome="C114" abreviatura="Laboratório"
descricao="C114 - Laboratório experimentos em alimentos" point="{X=492,Y=1114}"
raioPonto="4" tipo="PontoReferencia" /><node nome="C113"
abreviatura="Laboratório" descricao="C113 - Laboratório química"
point="{X=500,Y=834}" raioPonto="4" tipo="PontoReferencia" /><node
nome="BD29" abreviatura="Banheiro Deficiente" descricao="Banheiro Deficiente 29"
point="{X=500,Y=626}" raioPonto="4" tipo="PontoReferencia" /><node
nome="BM30" abreviatura="Banheiro Masculino" descricao="Banheiro Masculino 30"
point="{X=492,Y=581}" raioPonto="4" tipo="PontoReferencia" /><node nome="BF31"
abreviatura="Banheiro Feminino" descricao="Banheiro feminino 31"
point="{X=500,Y=521}" raioPonto="4" tipo="PontoReferencia" /><node nome="C108"
abreviatura="Sem nomenclatura" descricao="Sem descrição"
```



```

point="{X=1661,Y=456}" raioPonto="4" tipo="PontoReferencia" /><node
nome="NUAPE21" abreviatura="NUAPE" descricao="Núcleo de Acompanhamento
Psicopedagógico e Apoio ao Estudante" point="{X=839,Y=1232}" raioPonto="4"
tipo="PontoReferencia" /><node nome="NUAPE21" abreviatura="NUAPE"
descricao="Núcleo de Acompanhamento Psicopedagógico e Apoio ao Estudante"
point="{X=966,Y=1229}" raioPonto="4" tipo="PontoReferencia" /><node
nome="ENF20" abreviatura="Enfermaria" descricao="Enfermaria 20"
point="{X=1120,Y=1224}" raioPonto="4" tipo="PontoReferencia" /><node
nome="ENF19" abreviatura="Enfermaria" descricao="Enfermaria 19"
point="{X=1270,Y=1227}" raioPonto="4" tipo="PontoReferencia" /><node
nome="ENF18" abreviatura="Enfermaria" descricao="Enfermaria 18"
point="{X=1337,Y=1227}" raioPonto="4" tipo="PontoReferencia" /><node
nome="CAP16" abreviatura="Capela" descricao="Capela 16"
point="{X=1490,Y=1229}" raioPonto="4" tipo="PontoReferencia" /><node
nome="SA15" abreviatura="Sala" descricao="Sala atlética 15"
point="{X=1586,Y=1227}" raioPonto="4" tipo="PontoReferencia" /><node
nome="SGA14" abreviatura="Secretaria" descricao="Secretaria de Gestão
Acadêmica" point="{X=1739,Y=1221}" raioPonto="4" tipo="PontoReferencia"
/><node nome="BSM13" abreviatura="Banheiro Servidor Masculino"
descricao="Banheiro Servidor Masculino 13" point="{X=1863,Y=1227}" raioPonto="4"
tipo="PontoReferencia" /><node nome="BSF12" abreviatura="Banheiro servidor
feminino" descricao="Banheiro servidor feminino 12" point="{X=1944,Y=1227}"
raioPonto="4" tipo="PontoReferencia" /><node nome="DGEP11"
abreviatura="Diretoria" descricao="Diretoria de graduação e educação profissional"
point="{X=2020,Y=1221}" raioPonto="4" tipo="PontoReferencia" /><node
nome="ES10" abreviatura="Escada" descricao="Escada 10"
point="{X=2101,Y=1224}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C105" abreviatura="Sem nomenclatura" descricao="Sem descrição"
point="{X=2266,Y=282}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C106" abreviatura="Sem nomenclatura" descricao="Sem descrição"
point="{X=2069,Y=273}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C101" abreviatura="Mini auditório" descricao="C101 - Mini Auditório"
point="{X=2341,Y=1447}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C107" abreviatura="Sem nomenclatura" descricao="Sem descrição"
point="{X=1924,Y=282}" raioPonto="4" tipo="PontoReferencia" /><node
nome="CE40" abreviatura="Coordenação" descricao="Coordenação eletrônica"
point="{X=1646,Y=267}" raioPonto="4" tipo="PontoReferencia" /><node
nome="CE39" abreviatura="Coordenação" descricao="Coordenação eletrônica"
point="{X=1476,Y=267}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C109" abreviatura="Coordenação" descricao="C109 - Coordenação
eletrônica" point="{X=1215,Y=267}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C110" abreviatura="Laboratório" descricao="Laboratório eletrônica"
point="{X=1021,Y=273}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C111" abreviatura="DAALM" descricao="Departamento academico de
alimentos" point="{X=850,Y=267}" raioPonto="4" tipo="PontoReferencia" /><node
nome="C112" abreviatura="Coordenação" descricao="Coordenação Alimentos"
point="{X=601,Y=276}" raioPonto="4" tipo="PontoReferencia"
/><graphNodeNeighbors><node nome="" abreviatura="*1"><neighbor nome=""
abreviatura="*2" liga="*1" /><neighbor nome="" abreviatura="*4" liga="*1"
/><neighbor nome="EL27" abreviatura="Elevador" liga="*1" /><neighbor

```

nome="SD34" abreviatura="Saída" liga="*1" /><neighbor nome="BB33"
 abreviatura="Biblioteca" liga="*1" /><neighbor nome="C115" abreviatura="Auditório"
 liga="*1" /><neighbor nome="C114" abreviatura="Laboratório" liga="*1" /><neighbor
 nome="C113" abreviatura="Laboratório" liga="*1" /><neighbor nome="BD29"
 abreviatura="Banheiro Deficiente" liga="*1" /><neighbor nome="BM30"
 abreviatura="Banheiro Masculino" liga="*1" /><neighbor nome="BF31"
 abreviatura="Banheiro Feminino" liga="*1" /><neighbor nome="C108"
 abreviatura="Sem nomenclatura" liga="*1" /><neighbor nome="C105"
 abreviatura="Sem nomenclatura" liga="*1" /><neighbor nome="C106"
 abreviatura="Sem nomenclatura" liga="*1" /><neighbor nome="C107"
 abreviatura="Sem nomenclatura" liga="*1" /><neighbor nome="CE40"
 abreviatura="Coordenação" liga="*1" /><neighbor nome="CE39"
 abreviatura="Coordenação" liga="*1" /><neighbor nome="C109"
 abreviatura="Coordenação" liga="*1" /><neighbor nome="C110"
 abreviatura="Laboratório" liga="*1" /><neighbor nome="C111" abreviatura="DAALM"
 liga="*1" /><neighbor nome="C112" abreviatura="Coordenação" liga="*1"
 /></node><node nome="" abreviatura="*2"><neighbor nome="" abreviatura="*1"
 liga="*2" /><neighbor nome="" abreviatura="*3" liga="*2" /><neighbor nome="SD17"
 abreviatura="Saída" liga="*2" /><neighbor nome="SD23" abreviatura="Saída"
 liga="*2" /><neighbor nome="ES24" abreviatura="Escada" liga="*2" /><neighbor
 nome="EL27" abreviatura="Elevador" liga="*2" /><neighbor nome="SD34"
 abreviatura="Saída" liga="*2" /><neighbor nome="BB33" abreviatura="Biblioteca"
 liga="*2" /><neighbor nome="C115" abreviatura="Auditório" liga="*2" /><neighbor
 nome="C114" abreviatura="Laboratório" liga="*2" /><neighbor nome="C113"
 abreviatura="Laboratório" liga="*2" /><neighbor nome="BD29"
 abreviatura="Banheiro Deficiente" liga="*2" /><neighbor nome="BM30"
 abreviatura="Banheiro Masculino" liga="*2" /><neighbor nome="BF31"
 abreviatura="Banheiro Feminino" liga="*2" /><neighbor nome="NUAPE21"
 abreviatura="NUAPE" liga="*2" /><neighbor nome="NUAPE21"
 abreviatura="NUAPE" liga="*2" /><neighbor nome="ENF20"
 abreviatura="Enfermaria" liga="*2" /><neighbor nome="ENF19"
 abreviatura="Enfermaria" liga="*2" /><neighbor nome="ENF18"
 abreviatura="Enfermaria" liga="*2" /><neighbor nome="CAP16"
 abreviatura="Capela" liga="*2" /><neighbor nome="SA15" abreviatura="Sala"
 liga="*2" /><neighbor nome="SGA14" abreviatura="Secretaria" liga="*2" /><neighbor
 nome="BSM13" abreviatura="Banheiro Servidor Masculino" liga="*2" /><neighbor
 nome="BSF12" abreviatura="Banheiro servidor feminino" liga="*2" /><neighbor
 nome="DGEP11" abreviatura="Diretoria" liga="*2" /><neighbor nome="ES10"
 abreviatura="Escada" liga="*2" /><neighbor nome="C101" abreviatura="Mini
 auditório" liga="*2" /></node><node nome="" abreviatura="*3"><neighbor nome=""
 abreviatura="*2" liga="*3" /><neighbor nome="" abreviatura="*4" liga="*3"
 /><neighbor nome="BF02" abreviatura="Banheiro Feminino" liga="*3" /><neighbor
 nome="BM03" abreviatura="Banheiro Masculino" liga="*3" /><neighbor
 nome="BD03" abreviatura="Banheiro Deficiente" liga="*3" /><neighbor nome="C104"
 abreviatura="Aula Teórica" liga="*3" /><neighbor nome="C103" abreviatura="Aula
 Teórica" liga="*3" /><neighbor nome="SD01" abreviatura="Saída" liga="*3"
 /><neighbor nome="SD07" abreviatura="Saída" liga="*3" /><neighbor nome="C102"
 abreviatura="Aula Teórica" liga="*3" /><neighbor nome="SD17" abreviatura="Saída"
 liga="*3" /><neighbor nome="SD23" abreviatura="Saída" liga="*3" /><neighbor
 nome="ES24" abreviatura="Escada" liga="*3" /><neighbor nome="NUAPE21"

abreviatura="NUAPE" liga="*3" /><neighbor nome="NUAPE21"
 abreviatura="NUAPE" liga="*3" /><neighbor nome="ENF20"
 abreviatura="Enfermaria" liga="*3" /><neighbor nome="ENF19"
 abreviatura="Enfermaria" liga="*3" /><neighbor nome="ENF18"
 abreviatura="Enfermaria" liga="*3" /><neighbor nome="CAP16"
 abreviatura="Capela" liga="*3" /><neighbor nome="SA15" abreviatura="Sala"
 liga="*3" /><neighbor nome="SGA14" abreviatura="Secretaria" liga="*3" /><neighbor
 nome="BSM13" abreviatura="Banheiro Servidor Masculino" liga="*3" /><neighbor
 nome="BSF12" abreviatura="Banheiro servidor feminino" liga="*3" /><neighbor
 nome="DGEP11" abreviatura="Diretoria" liga="*3" /><neighbor nome="ES10"
 abreviatura="Escada" liga="*3" /><neighbor nome="C101" abreviatura="Mini
 auditório" liga="*3" /></node><node nome="" abreviatura="*4"><neighbor nome=""
 abreviatura="*3" liga="*4" /><neighbor nome="" abreviatura="*1" liga="*4"
 /><neighbor nome="BF02" abreviatura="Banheiro Feminino" liga="*4" /><neighbor
 nome="BM03" abreviatura="Banheiro Masculino" liga="*4" /><neighbor
 nome="BD03" abreviatura="Banheiro Deficiente" liga="*4" /><neighbor nome="C104"
 abreviatura="Aula Teórica" liga="*4" /><neighbor nome="C103" abreviatura="Aula
 Teórica" liga="*4" /><neighbor nome="SD01" abreviatura="Saída" liga="*4"
 /><neighbor nome="SD07" abreviatura="Saída" liga="*4" /><neighbor nome="C102"
 abreviatura="Aula Teórica" liga="*4" /><neighbor nome="C108" abreviatura="Sem
 nomenclatura" liga="*4" /><neighbor nome="C105" abreviatura="Sem nomenclatura"
 liga="*4" /><neighbor nome="C106" abreviatura="Sem nomenclatura" liga="*4"
 /><neighbor nome="C107" abreviatura="Sem nomenclatura" liga="*4" /><neighbor
 nome="CE40" abreviatura="Coordenação" liga="*4" /><neighbor nome="CE39"
 abreviatura="Coordenação" liga="*4" /><neighbor nome="C109"
 abreviatura="Coordenação" liga="*4" /><neighbor nome="C110"
 abreviatura="Laboratório" liga="*4" /><neighbor nome="C111" abreviatura="DAALM"
 liga="*4" /><neighbor nome="C112" abreviatura="Coordenação" liga="*4"
 /></node><node nome="BF02" abreviatura="Banheiro Feminino"><neighbor
 nome="" abreviatura="*3" liga="Banheiro Feminino" /><neighbor nome=""
 abreviatura="*4" liga="Banheiro Feminino" /></node><node nome="BM03"
 abreviatura="Banheiro Masculino"><neighbor nome="" abreviatura="*3"
 liga="Banheiro Masculino" /><neighbor nome="" abreviatura="*4" liga="Banheiro
 Masculino" /></node><node nome="BD03" abreviatura="Banheiro
 Deficiente"><neighbor nome="" abreviatura="*3" liga="Banheiro Deficiente"
 /><neighbor nome="" abreviatura="*4" liga="Banheiro Deficiente" /></node><node
 nome="C104" abreviatura="Aula Teórica"><neighbor nome="" abreviatura="*3"
 liga="Aula Teórica" /><neighbor nome="" abreviatura="*4" liga="Aula Teórica"
 /></node><node nome="C103" abreviatura="Aula Teórica"><neighbor nome=""
 abreviatura="*3" liga="Aula Teórica" /><neighbor nome="" abreviatura="*4"
 liga="Aula Teórica" /></node><node nome="SD01" abreviatura="Saída"><neighbor
 nome="" abreviatura="*3" liga="Saída" /><neighbor nome="" abreviatura="*4"
 liga="Saída" /></node><node nome="SD07" abreviatura="Saída"><neighbor
 nome="" abreviatura="*4" liga="Saída" /><neighbor nome="" abreviatura="*3"
 liga="Saída" /></node><node nome="C102" abreviatura="Aula Teórica"><neighbor
 nome="" abreviatura="*3" liga="Aula Teórica" /><neighbor nome="" abreviatura="*4"
 liga="Aula Teórica" /></node><node nome="SD17" abreviatura="Saída"><neighbor
 nome="" abreviatura="*2" liga="Saída" /><neighbor nome="" abreviatura="*3"
 liga="Saída" /></node><node nome="SD23" abreviatura="Saída"><neighbor
 nome="" abreviatura="*2" liga="Saída" /><neighbor nome="" abreviatura="*3"

liga="Saída" /></node><node nome="ES24" abreviatura="Escada"><neighbor
 nome="" abreviatura="*3" liga="Escada" /><neighbor nome="" abreviatura="*2"
 liga="Escada" /></node><node nome="EL27" abreviatura="Elevador"><neighbor
 nome="" abreviatura="*1" liga="Elevador" /><neighbor nome="" abreviatura="*2"
 liga="Elevador" /></node><node nome="SD34" abreviatura="Saída"><neighbor
 nome="" abreviatura="*2" liga="Saída" /><neighbor nome="" abreviatura="*1"
 liga="Saída" /></node><node nome="BB33" abreviatura="Biblioteca"><neighbor
 nome="" abreviatura="*2" liga="Biblioteca" /><neighbor nome="" abreviatura="*1"
 liga="Biblioteca" /></node><node nome="C115" abreviatura="Auditório"><neighbor
 nome="" abreviatura="*1" liga="Auditório" /><neighbor nome="" abreviatura="*2"
 liga="Auditório" /></node><node nome="C114" abreviatura="Laboratório"><neighbor
 nome="" abreviatura="*1" liga="Laboratório" /><neighbor nome="" abreviatura="*2"
 liga="Laboratório" /></node><node nome="C113"
 abreviatura="Laboratório"><neighbor nome="" abreviatura="*1" liga="Laboratório"
 /><neighbor nome="" abreviatura="*2" liga="Laboratório" /></node><node
 nome="BD29" abreviatura="Banheiro Deficiente"><neighbor nome=""
 abreviatura="*1" liga="Banheiro Deficiente" /><neighbor nome="" abreviatura="*2"
 liga="Banheiro Deficiente" /></node><node nome="BM30" abreviatura="Banheiro
 Masculino"><neighbor nome="" abreviatura="*1" liga="Banheiro Masculino"
 /><neighbor nome="" abreviatura="*2" liga="Banheiro Masculino" /></node><node
 nome="BF31" abreviatura="Banheiro Feminino"><neighbor nome=""
 abreviatura="*1" liga="Banheiro Feminino" /><neighbor nome="" abreviatura="*2"
 liga="Banheiro Feminino" /></node><node nome="C108" abreviatura="Sem
 nomenclatura"><neighbor nome="" abreviatura="*1" liga="Sem nomenclatura"
 /><neighbor nome="" abreviatura="*4" liga="Sem nomenclatura" /></node><node
 nome="NUAPE21" abreviatura="NUAPE"><neighbor nome="" abreviatura="*2"
 liga="NUAPE" /><neighbor nome="" abreviatura="*3" liga="NUAPE" /></node><node
 nome="NUAPE21" abreviatura="NUAPE"><neighbor nome="" abreviatura="*2"
 liga="NUAPE" /><neighbor nome="" abreviatura="*3" liga="NUAPE" /></node><node
 nome="ENF20" abreviatura="Enfermaria"><neighbor nome="" abreviatura="*2"
 liga="Enfermaria" /><neighbor nome="" abreviatura="*3" liga="Enfermaria"
 /></node><node nome="ENF19" abreviatura="Enfermaria"><neighbor nome=""
 abreviatura="*2" liga="Enfermaria" /><neighbor nome="" abreviatura="*3"
 liga="Enfermaria" /></node><node nome="ENF18"
 abreviatura="Enfermaria"><neighbor nome="" abreviatura="*2" liga="Enfermaria"
 /><neighbor nome="" abreviatura="*3" liga="Enfermaria" /></node><node
 nome="CAP16" abreviatura="Capela"><neighbor nome="" abreviatura="*2"
 liga="Capela" /><neighbor nome="" abreviatura="*3" liga="Capela" /></node><node
 nome="SA15" abreviatura="Sala"><neighbor nome="" abreviatura="*2" liga="Sala"
 /><neighbor nome="" abreviatura="*3" liga="Sala" /></node><node nome="SGA14"
 abreviatura="Secretaria"><neighbor nome="" abreviatura="*2" liga="Secretaria"
 /><neighbor nome="" abreviatura="*3" liga="Secretaria" /></node><node
 nome="BSM13" abreviatura="Banheiro Servidor Masculino"><neighbor nome=""
 abreviatura="*2" liga="Banheiro Servidor Masculino" /><neighbor nome=""
 abreviatura="*3" liga="Banheiro Servidor Masculino" /></node><node
 nome="BSF12" abreviatura="Banheiro servidor feminino"><neighbor nome=""
 abreviatura="*2" liga="Banheiro servidor feminino" /><neighbor nome=""
 abreviatura="*3" liga="Banheiro servidor feminino" /></node><node
 nome="DGEP11" abreviatura="Diretoria"><neighbor nome="" abreviatura="*2"
 liga="Diretoria" /><neighbor nome="" abreviatura="*3" liga="Diretoria"

```

/></node><node nome="ES10" abreviatura="Escada"><neighbor nome=""
abreviatura="*2" liga="Escada" /><neighbor nome="" abreviatura="*3" liga="Escada"
/></node><node nome="C105" abreviatura="Sem nomenclatura"><neighbor
nome="" abreviatura="*1" liga="Sem nomenclatura" /><neighbor nome=""
abreviatura="*4" liga="Sem nomenclatura" /></node><node nome="C106"
abreviatura="Sem nomenclatura"><neighbor nome="" abreviatura="*1" liga="Sem
nomenclatura" /><neighbor nome="" abreviatura="*4" liga="Sem nomenclatura"
/></node><node nome="C101" abreviatura="Mini auditório"><neighbor nome=""
abreviatura="*2" liga="Mini auditório" /><neighbor nome="" abreviatura="*3"
liga="Mini auditório" /></node><node nome="C107" abreviatura="Sem
nomenclatura"><neighbor nome="" abreviatura="*1" liga="Sem nomenclatura"
/><neighbor nome="" abreviatura="*4" liga="Sem nomenclatura" /></node><node
nome="CE40" abreviatura="Coordenação"><neighbor nome="" abreviatura="*1"
liga="Coordenação" /><neighbor nome="" abreviatura="*4" liga="Coordenação"
/></node><node nome="CE39" abreviatura="Coordenação"><neighbor nome=""
abreviatura="*4" liga="Coordenação" /><neighbor nome="" abreviatura="*1"
liga="Coordenação" /></node><node nome="C109"
abreviatura="Coordenação"><neighbor nome="" abreviatura="*1"
liga="Coordenação" /><neighbor nome="" abreviatura="*4" liga="Coordenação"
/></node><node nome="C110" abreviatura="Laboratório"><neighbor nome=""
abreviatura="*1" liga="Laboratório" /><neighbor nome="" abreviatura="*4"
liga="Laboratório" /></node><node nome="C111" abreviatura="DAALM"><neighbor
nome="" abreviatura="*1" liga="DAALM" /><neighbor nome="" abreviatura="*4"
liga="DAALM" /></node><node nome="C112"
abreviatura="Coordenação"><neighbor nome="" abreviatura="*4"
liga="Coordenação" /><neighbor nome="" abreviatura="*1" liga="Coordenação"
/></node></graphNodeNeighbors></graphNodes></obj_Planta>

```