

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

WIGLAN ELLI PATRIEL TOSCAN MARIANI

**APLICATIVO IDENTIFICADOR DE ESPAÇOS ACESSÍVEIS PARA USO EM
DISPOSITIVOS MÓVEIS**

TRABALHO DE CONCLUSÃO DE CURSO

**PATO BRANCO
2014**

WIGLAN ELLI PATRIEL TOSCAN MARIANI

**APLICATIVO IDENTIFICADOR DE ESPAÇOS ACESSÍVEIS PARA USO EM
DISPOSITIVOS MÓVEIS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

Orientadora: Profa. Beatriz Terezinha Borsoi

Coorientadora: Profa. Rúbia Eliza de Oliveira Shultz Ascari

**PATO BRANCO
2014**

ATA Nº: 262

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DO ALUNO WIGLAN ELLI PATRIEL TOSCAN MARIANI.

As 10:00 hrs do dia 22 de dezembro de 2014, no Bloco V da UTFPR, Câmpus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Beatriz Terezinha Borsoi (Orientador), Andréia Scariot Beulke (Convidada) e Vinicius Pegorini (Convidado), para avaliar o Trabalho de Diplomação do aluno Wiglan Elli Patriel Toscan Mariani 01260650, sob o título **Aplicativo Identificador de Espaços Acessíveis para Uso em Dispositivos Móveis**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, COADS. Após a apresentação o candidato foi entrevistado pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. As 10:35 hrs foi encerrada a sessão.



Prof. Beatriz Terezinha Borsoi, Dr.
Orientadora



Prof. Andréia Scariot Beulke, M. Sc.
Convidado



Prof. Vinicius Pegorini, Esp.
Convidado



Prof. Eliane Maria de Bortoli Fávero, M.Sc
Coordenadora do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.
Coordenador do Curso

Dedico este trabalho aos meus pais, especialmente a minha mãe Elenir Mariani pelo apoio, incentivo e dedicação que possibilitou esta caminhada acadêmica. Aos demais familiares e amigos pelo incentivo e apoio prestado.

AGRADECIMENTOS

A Deus pela oportunidade de vida a cada dia, pelas oportunidades de aprendizado e experiência, pela sabedoria e tranquilidade na solução dos problemas encontrados, pela paz e conforto oferecido nos momentos de dificuldades.

Agradeço aos meus pais, Gelcir Mariani e Elenir Mariani, pelo esforço e dedicação que possibilitaram esta formação, pela educação e incentivos prestados. Ao meu irmão Wilian Mariani e demais familiares, pelo apoio, interesse e compreensão que me ajudaram durante esta caminhada.

Especialmente a Profa. Rúbia E. O. Schultz Ascari que com paciência, sabedoria e dedicação orientou este trabalho. Em especial a Professora/Orientadora Beatriz Borsoi, por toda ajuda prestada. Aos demais professores pelos ensinamentos, experiências e auxílios compartilhados.

Agradeço aos colegas de formação e de trabalho que contribuíram com os conhecimentos adquiridos. Especialmente ao colega Paulo Henrique Merlin pela amizade, pelo apoio e tempo dedicado, por não medir esforços na solução dos trabalhos e problemas.

A Todos, que diretamente ou indiretamente contribuíram para o alcance dos meus objetivos.

Devemos mudar nossa atitude tradicional em relação à construção de programas. Em vez de imaginar que nossa principal tarefa é instruir o computador sobre o que ele deve fazer, vamos imaginar que nossa principal tarefa é explicar aos seres humanos o que queremos que o computador faça.

Donald E. Knuth, Literate Programming

RESUMO

MARIANI, Wiglan Elli Patriel Toscan. Aplicativo identificador de espaços acessíveis para uso em dispositivos móveis. 2014. 58 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2014.

Este trabalho apresenta o desenvolvimento de um aplicativo construído com o propósito de melhorar a qualidade de vida de pessoas que apresentam algum tipo de dificuldade motora. Visando, assim, aplicar a TI (Tecnologia da Informação) para obter uma melhor orientação e conseqüente locomoção em ambientes desconhecidos para pessoas portadoras de dificuldade de mobilidade. O aplicativo desenvolvido é para uso em dispositivos móveis facilitando, assim, seu uso em ambientes aberto como estacionamentos, por exemplo. O aplicativo permite o cadastro e a posterior consulta de espaços acessíveis, ou seja, rampas, banheiros adaptados e outros. O aplicativo será utilizado inicialmente para mapear os espaços acessíveis disponíveis na Universidade Tecnológica Federal do Paraná (UTFPR), Câmpus Pato Branco, para melhorar a locomoção de acadêmicos, servidores ou visitantes que dependam de espaços acessíveis para locomoção.

Palavras-chave: Acessibilidade. Dispositivos móveis. Dificuldade motora.

ABSTRACT

MARIANI, Wiglan Elli Patriel Toscan. Application to use in mobile devices by persons with reduced mobility. 2014. 58 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2014.

This monograph presents the development of an application built aiming to improve the quality of life of people with some motor difficulty. Thus, applying information technology to get a better orientation and locomotion, resulting in help people with mobility difficulties in unknown environments. The developed application is to mobile devices, thus facilitating its use in open environments such as parking lots, for example. The application allows the registration and the subsequent accessible spaces consultation, or ramps and bathrooms adapted and others. The application will initially be used to map the accessible spaces available in UTFPR campus Pato Branco, to improve the mobility of academics, servers and visitors who depend on affordable spaces for locomotion.

Keywords: Accessibility. Mobiles Devices. Motor difficulty.

LISTA DE FIGURAS

Figura 1 - Tela inicial do aplicativo	24
Figura 2 - Menu lateral com opções de acesso	25
Figura 3 - Navegação livre	26
Figura 4 - Indicador de como adicionar novo Marcador	27
Figura 5 - Inserção de um novo ponto	28
Figura 6 - Ponto lançado no mapa	29
Figura 7 - Informativo adicionar marcador	29
Figura 8 - Edição do marcador com opção de exclusão	30
Figura 9 - Múltiplos marcadores cadastrados	31
Figura 10 - Ícone de remoção dos marcadores	31
Figura 11 - Remoção dos marcadores.....	31
Figura 12 - Listagem de pontos cadastrados	32
Figura 13 - Localização em foco para ser direcionada.....	32
Figura 14 - Direcionamento do ponto depois de selecionado.....	33
Figura 15 - Localização no mapa	34
Figura 16 - Ajuste de zoom aos pontos aplicados no mapa.....	35
Figura 17 - UTFPR - Câmpus Pato Branco	36
Figura 18 - Lista de categorias	37
Figura 19 - Adicionando novo ponto.....	38
Figura 20 - Identificação do ponto	39
Figura 21 - Exclusão/alteração do ponto	40
Figura 22 - Pontos adicionados.....	41
Figura 23 - Ações que estão definidas para serem mapeadas.....	46
Figura 24 - Estrutura do componente TMapImage.....	49
Figura 25 - Campos salvos no banco de dados.....	53

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais	23
Quadro 2 - Requisitos não funcionais	23

LISTAGENS DE CÓDIGOS

Listagem 1 - Visibilidade menu lateral	42
Listagem 2 - Abertura de tela navegação livre.....	42
Listagem 3 - Abertura de tela navegação Câmpus Pato Branco.....	42
Listagem 4 - Carregamento dos pontos já existentes.....	43
Listagem 5 - Evento BeforePost do cdsPontos	44
Listagem 6 - Comando para adicionar marcador ao mapa.....	45
Listagem 7 – Salvamento de latitude e longitude.....	45
Listagem 8 - Comando de remoção de pontos	46
Listagem 9 - Remoção do ponto da lista de pontos do mapa	46
Listagem 10 - Programação das ações	47
Listagem 11 - Ativação do GPS do dispositivo.....	47
Listagem 12 - Definição da localização atual.....	48
Listagem 13 - Evento em que o marcador estava sendo movido	49
Listagem 14 - Classe TMapImage	50
Listagem 15 - Classe TMarcador	51
Listagem 16 - Create do componente TMarcador.....	51
Listagem 17 - Desenhando identificador sobre o marcador	52
Listagem 18 - Classe TCircle.....	52
Listagem 19 - Evento Create da classe TCircle.....	53
Listagem 20 - Mapeando evento do componente	54
Listagem 21 - Apresentando ajuda e editando <i>marker</i>	54
Listagem 22 - Atualização do dispositivo.....	55

LISTA DE SIGLAS

e- MAG	Modelo de Acessibilidade de Governo Eletrônico
GPS	<i>Global Position System</i>
RAD	<i>Rapid Application Development</i>
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais.
UTFPR	Universidade Tecnológica Federal do Paraná
W3C	<i>World Wide Web Consortium</i>
WAI	<i>Web Accessibility Initiative</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 CONSIDERAÇÕES INICIAIS	12
1.2 OBJETIVOS	13
1.2.1 Objetivo Geral	14
1.2.2 Objetivos Específicos	14
1.3 JUSTIFICATIVA	14
1.4 ESTRUTURA DO TRABALHO	15
2 DISPOSITIVOS MÓVEIS E ACESSIBILIDADE	16
2.1 ACESSIBILIDADE	16
2.2 USO DE DISPOSITIVOS MÓVEIS PARA PROVER ACESSIBILIDADE	17
3 MATERIAIS E MÉTODO	19
3.1 MATERIAIS	19
3.2 MÉTODO.....	21
4 RESULTADOS	22
4.1 ESCOPO DO SISTEMA	22
4.2 MODELAGEM DO SISTEMA	22
4.3 APRESENTAÇÃO DO SISTEMA.....	23
4.4 IMPLEMENTAÇÕES DO SISTEMA	41
5 CONCLUSÃO	56
REFERÊNCIAS	57

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais que explicita o contexto do trabalho, os objetivos e justificativa, sendo finalizado com a apresentação da organização do texto por meio dos seus capítulos.

1.1 CONSIDERAÇÕES INICIAIS

A Norma Brasileira NBR 9050-2004 adota a seguinte definição de acessibilidade: "Possibilidade e condição de alcance, percepção e entendimento para a utilização com segurança e autonomia de edificações, espaço, mobiliário, equipamento urbano e elementos" (ABNT NBR 9050, 2005, p. 2). A própria norma complementa que para o espaço, edificação, mobiliário, equipamento ou elemento ser acessível é necessário que o mesmo possa ser alcançado, acionado, utilizado e vivenciado por qualquer pessoa, inclusive aquelas com mobilidade reduzida. Acessível está relacionado à acessibilidade física e de comunicação (ABNT NBR 9050, 2005, p. 2).

A Lei n° 10.098 de 19 de dezembro de 2000 (BRASIL, 2000) estabelece normas gerais e critérios básicos para a promoção da acessibilidade por pessoas portadoras de deficiência ou com mobilidade reduzida. As normas foram estabelecidas para assegurar a integração social por meio do exercício dos direitos individuais e sociais, embasadas no respeito à dignidade e na justiça social, no intuito de possibilitar às pessoas com limitação física acesso à educação, saúde, trabalho, lazer, previdência social, além de amparo à infância e maternidade (PAGLIUCA; ARAGÃO; ALMEIDA, 2007).

Segundo Godoy *et al.* (2000), a conquista por espaços livres de barreiras arquitetônicas implica a possibilidade e a condição de alcance para que portadores de deficiência utilizem com segurança e autonomia edificações, mobiliários, equipamentos urbanos, transportes e meios de comunicação.

Assim, para que cadeirantes e pessoas com mobilidade reduzida possam exercer seus direitos e fortalecerem sua participação como cidadãos, os ambientes que eles frequentam devem estar adaptados, disponibilizando rampas, estacionamentos, banheiros e espaços acessíveis. Atualmente, com o amparo legal,

muitos ambientes, especialmente os públicos, já oferecem diversos espaços acessíveis. Contudo, muitas vezes os portadores de deficiência têm dificuldade para identificar e localizar os espaços acessíveis existentes em um ambiente que lhes é ainda desconhecido.

Os avanços da TI têm facilitado a inclusão social e profissional das pessoas com deficiência. O uso de dispositivos móveis vem sendo amplamente difundido nesse público, principalmente por já existir diversos aplicativos desenvolvidos que visam prover acessibilidade aos portadores de algum tipo de deficiência.

Esse trabalho tem como fundamentação experiências bem sucedidas observadas com relação ao desenvolvimento de aplicativos para dispositivos móveis, buscando prover melhor acessibilidade aos acadêmicos da Universidade Tecnológica Federal do Paraná (UTFPR), que venham a iniciar seus estudos em algum câmpus da Universidade. E, ainda, para visitantes desses câmpus que podem efetivamente não conhecer a sua infraestrutura.

Pretende-se desenvolver, portanto, um aplicativo capaz de apresentar de forma simples e prática, os espaços acessíveis que são disponíveis na estrutura atual do câmpus, ou seja, localização de rampas, vagas para estacionamento, banheiros adaptados, entre outros. Dessa forma, o acadêmico ou visitante com necessidades especiais pode se orientar e locomover da melhor forma possível desde o seu primeiro contato com o ambiente.

Além disso, o aplicativo permitirá ao usuário incluir novos pontos de acessibilidade, de acordo com sua necessidade, podendo expandir sua aplicação para identificar também outros locais que disponibilizam espaços acessíveis, como supermercados, restaurantes, cinemas, entre outros.

1.2 OBJETIVOS

O objetivo geral corresponde ao resultado principal da realização deste trabalho, ou seja, o desenvolvimento de um aplicativo para uso em dispositivos móveis. Os objetivos específicos complementam o objetivo geral, tanto em termos das tecnologias utilizadas como da finalidade do projeto.

1.2.1 Objetivo Geral

Desenvolver um aplicativo para dispositivos móveis capaz de identificar e registrar espaços acessíveis na UTFPR, câmpus Pato Branco, e em demais ambientes

1.2.2 Objetivos Específicos

- Proporcionar maior independência a portadores de deficiência, de forma que possam orientar-se adequadamente em ambientes desconhecidos, identificando facilmente a localização dos espaços acessíveis disponíveis.
- Criar um portfólio pessoal de experiências do portador de deficiência, que possa ser compartilhado com outras pessoas, a fim de melhorar a locomoção de pessoas com necessidades especiais.
- Proporcionar melhor receptividade a novos acadêmicos e visitantes da UTFPR que possuam necessidades especiais, disponibilizando uma ferramenta que identificará prontamente espaços acessíveis que possam melhorar sua locomoção pela estrutura dos câmpus.

1.3 JUSTIFICATIVA

O uso de Tecnologias de Informação e Comunicação (TICs) para apoio à mobilidade humana é apoiado por Conforto e Santarosa (2002, p. 89) que colocam que as tecnologias da inteligência e da comunicação ratificam a mobilidade humana e ao multiplicar proximidades cognitivas e afetivas apontam para a heterogeneidade do coletivo, que reconhece e valoriza a diversidade humana sem qualquer exclusão. Para esses autores essas tecnologias são as ferramentas digitais que por atuarem objetos catalisadores da inteligência coletiva, tornam-se entidades que propiciam o acolhimento da diversidade.

O aplicativo desenvolvido como resultado deste trabalho é realizado com o propósito de melhorar a qualidade de vida de portadores de deficiência, visando aplicar conhecimentos da tecnologia da informação para obter uma melhor orientação e consequente locomoção de pessoas com alguma dificuldade motora.

Assim, o sistema contribuirá contribuir para a independência em termos de mobilidade dos que possuem restrições naturais (natas ou adquiridas) de locomoção. O aplicativo foi desenvolvido visando os câmpus da UTFPR, mas pode ser utilizado em qualquer ambiente. Para o seu uso é necessário haver um mapa indicativo do local no qual são acrescentados os pontos de acessibilidade.

1.4 ESTRUTURA DO TRABALHO

O presente trabalho está organizado em capítulos, dos quais este é o primeiro e apresenta o conceito geral do trabalho, contendo os objetivos e a justificativa.

No Capítulo 2 é apresentado o referencial teórico sobre emprego de dispositivos móveis relacionados à temática de acessibilidade.

No Capítulo 3 estão os materiais e o método utilizados para o desenvolvimento do trabalho. Os materiais se referem ao que é necessário para modelar e implementar o sistema, incluindo as tecnologias, as ferramentas e os ambientes de desenvolvimento utilizados. O método se refere aos procedimentos utilizados no ciclo de vida do sistema, abrangendo da definição dos requisitos à implementação do sistema.

Os resultados obtidos com o trabalho, utilizando a metodologia proposta no Capítulo 3 estão contidos no Capítulo 4, apresentando imagens, gráficos e quadros para melhor compreensão.

Por fim, no Capítulo 5 está a conclusão do trabalho e as considerações finais.

2 DISPOSITIVOS MÓVEIS E ACESSIBILIDADE

Este capítulo apresenta a fundamentação conceitual do trabalho referindo-se à acessibilidade e implementação de aplicativos em dispositivos móveis. Isso porque o aplicativo desenvolvido visando auxiliar na mobilidade de portadores de deficiências é implementado para dispositivos móveis.

2.1 ACESSIBILIDADE

O conceito de acessibilidade faz relação às questões físicas de facilidades de acesso e à reabilitação física e profissional. Demonstra as condições de uso, como as interfaces são apresentadas ao usuário, como é feita essa troca de informação entre os usuários e a fonte de dados (sistemas, páginas *web*, etc.) e, principalmente, como se dará o acesso do usuário às informações disponíveis. A partir da década de 1980, no ano 1981, tempo em que ocorreu o “Ano Internacional das pessoas Deficientes”, a acessibilidade e a eliminação das barreiras físicas tornaram-se metas para todos os países em desenvolvimento.

No Brasil, o decreto de Lei nº 10098 de 19/12/2000 que estabelece normas gerais e critérios básicos para a promoção da acessibilidade, define acessibilidade como:

possibilidade e condição de alcance para utilização, com segurança e autonomia, dos espaços, mobiliários e equipamentos urbanos, das edificações, dos transportes e dos sistemas e meios de comunicação, por pessoa portadora de deficiência ou com mobilidade reduzida (BRASIL, 2008, p. 36).

Com o crescimento e a popularização da Internet, tornou-se necessário estabelecer padrões para que o conteúdo disponibilizado fosse considerado acessível a todos. Para que os sistemas computacionais fossem considerados acessíveis, na década de 1990, instituições como W3C (*World Wide Web Consortium* - Consórcio para a Web) e WAI (*Web Accessibility Initiative* - Iniciativa para acessibilidade na rede) estabeleceram padrões e protocolos que definiam se os sistemas desenvolvidos eram considerados acessíveis ou não.

Segundo ACESSOBRASIL (2014) acessibilidade:

Representa para o nosso usuário não só o direito de acessar a rede de informações, mas também o direito de eliminação de barreiras arquitetônicas, de disponibilidade de comunicação, de acesso físico, de equipamentos e programas adequados, de conteúdo e apresentação da informação em formatos alternativos (ACESSOBRASIL, 2014, p. 1)

Com o intuito de padronizar a criação e o desenvolvimento dos portais e sites do governo brasileiro, em 2007 tornou-se obrigatório seguir o conjunto de práticas estabelecidas no e-MAG (Modelo de Acessibilidade de Governo Eletrônico). Esse documento foi escrito com base nas necessidades brasileiras de acessibilidade e reúne um conjunto de recomendações e padronizações, de modelos de implementações, que facilitam e melhoram o acesso a sites do governo federal. O e-MAG também mantém de acordo as diretrizes internacionais de acessibilidade.

A Portaria SLT Nº 3, de 7 de maio de 2007 institucionaliza o e-MAG no âmbito do Sistema de Administração dos Recursos de Informação e Informática – SISP (BRASIL, 2008, p. 200).

2.2 USO DE DISPOSITIVOS MÓVEIS PARA PROVER ACESSIBILIDADE

Tecnologia Assistiva fornece possibilidades de propiciar uma maior independência, qualidade de vida e inclusão de pessoas portadoras de necessidades especiais (FAÇANHA, 2012).

Facilitar o acesso das pessoas que portam algum tipo de necessidade especial já é preocupação na maioria das empresas que desenvolvem algum tipo de produto. Um exemplo disso é a Google, que tem inovado seus aplicativos e ferramentas facilitando o uso. Já outras empresas fabricantes de sistemas operacionais como a Apple e Microsoft, disponibilizam ferramentas nativas voltadas a acessibilidade para o dispositivos produzidos.

Aplicativos gratuitos são disponibilizados em vários idiomas possibilitando desde a busca no Google até a compra de produtos *online*. Aplicativos instalados em *tablets* e celulares indicam aos portadores de deficiência que caminho seguir por meio de uso de sensores como *Global Position System* (GPS) e sensores de movimento.

O aplicativo Talkback desenvolvido pela Google e disponibilizado para a plataforma Android é um exemplo de melhoria de acessibilidade, pois ajuda

portadores de deficiência visual (ANDROID..., 2014). Ele faz a leitura de cada operação feita no dispositivo móvel, em paralelo à fala, a vibração também ocorre indicando que a operação foi realizada com sucesso. Além disso, o aplicativo reconhece comandos de voz.

A Apple se destaca no quesito acessibilidade, pois os dispositivos já têm diversos recursos nativos como, por exemplo, AssistiveTouch (APPLE, 2014; iOS, 2014). Esta interface permite adaptar a tela Multi-Touch do dispositivo com iOS às necessidades especiais dos seus usuários. Problemas como dificuldade para separar ou juntar os dedos são resolvidos substituindo comandos por toques únicos na tela ou é possível criar um movimento personalizado. O pressionamento do botão início pode ser substituído por um toque na tela. Os movimentos como girar e chacoalhar estão disponíveis mesmo se o dispositivo estiver preso a uma cadeira de rodas. Já o VoiceOver (VOICEOVER, 2014) é um leitor de tela baseado em movimentos que permite ouvir uma descrição de tudo o que estiver acontecendo na tela, de como está a carga da bateria, quem está ligando e até sobre qual aplicativo o dedo está posicionado. Também é possível adicionar monitores em braille com dispositivos iOS e sincronizar um monitor em *braille* Bluetooth para ler a voz do VoiceOver. Os monitores em Braille com teclas podem ser usados para controlar e gerenciar as atividades no dispositivo. O BrailleTouch (BRAILLE..., 2014) permite digitar usando *braille* direto na tela do dispositivo.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e o método utilizados para a análise e o desenvolvimento do aplicativo que é resultado da realização deste trabalho.

3.1 MATERIAIS

As ferramentas e as tecnologias utilizadas para as atividades de modelagem e implementação do sistema são: Delphi, Android, SQLite, FireDac, GPS, Tortoise SVN e ambientes de teste.

O ambiente utilizado para o desenvolvimento do aplicativo foi o Delphi, em sua versão Delphi XE7. O Delphi é uma Interface de Desenvolvimento Integrado (IDE), e uma ferramenta *Rapid Application Development* (RAD) () que permite o desenvolvimento rápido de aplicativos, tornando uma linguagem com o intuito de facilitar e diminuir o trabalho de um desenvolvedor. Sua essência é ser simples e evita digitar uma quantidade excessiva de códigos. No RAD, o processo de *design* e de desenvolvimento acontece simultaneamente, tornando-o mais ágil para um melhor prazo de entrega de um produto final (PISKE; SEIDEL, 2006).

A linguagem Delphi possibilita a publicação de aplicativos para as plataformas Android e iOS, o que torna-se uma vantagem a sua utilização. Delphi é uma solução tecnológica difundida para desenvolvimento *desktop* e que passou a atuar recentemente no mercado de desenvolvimento de aplicações para dispositivos móveis.

O aplicativo foi desenvolvido para o Android na versão 4.4.2. Em testes foram levantados problemas que impedem a compilação do aplicativo para versões posteriores até o momento.

O Banco de dados utilizado para o armazenamento das informações do aplicativo é o IBLite sob domínio da Interbase. O IBLite é um banco de dados escalável e é compatível com diversas plataformas como Android, iOS, OS X, Linux.

O IBLite pode utilizar recursos de múltiplos núcleos. Em questões de segurança, o IBLite oferece criptografia para o banco como um todo, ou em tabelas e até mesmo somente em colunas. Também é possível definir permissões para

múltiplos usuários e filtrar informações do que cada um pode visualizar. Ele pode ser distribuído tanto em sistema de computação em nuvens, quanto distribuição local, visando segurança e desempenho.

O IBLite é uma solução adequada para ser instalada em dispositivos móveis porque possui tamanho de arquivo reduzido. Ele é disponibilizado junto com a ferramenta Delphi XE5 e pode ser distribuído tanto para iOS como para Android.

O *framework* de acesso a dados utilizado foi o FireDAC. Essa solução é disponibilizada pela Embarcadero e é uma ferramenta que permite a conexão nativa com diversos banco de dados como InterBase, SQLite, MySQL, SQL Server, Oracle, PostgreSQL, DB2, SQL Anywhere, Advantage DB, Firebird, Access, Informix, DataSnap, dentre outros.

GPS é um instrumento de navegação por satélite, que possui utilização comum em várias áreas. Para o mercado de dispositivos móveis, ele já se tornou um item obrigatório se for considerada a quantidade de aplicativos e recursos que oferecem serviços ou dependem do GPS para funcionar. Para este trabalho, a utilização do GPS é fundamental, uma vez que os principais recursos do aplicativo dependem do uso do GPS para funcionar.

A ferramenta Tortoise SVN tem objetivo de compartilhamento e versionamento de arquivos, permitindo a estruturação de pastas, repositórios, usuários, versões de arquivos e outras funcionalidades. Esta ferramenta foi fundamental para o desenvolvimento deste trabalho, uma vez que permitia o acesso rápido aos arquivos necessários para o desenvolvimento do projeto. Além de controlar e manter histórico das alterações realizadas, o que reduz a chance de erro na utilização dos códigos-fontes.

Para realizar os testes do aplicativo foram utilizados dispositivos reais e emuladores, ambos para a plataforma Android na versão 4.4.2. No caso do Android, o emulador está presente na instalação do Delphi XE7 e funciona como um dispositivo real que utiliza Android. No entanto, sua utilização não é muito viável considerando o tempo de resposta do emulador. Para testar o aplicativo para IOs, seria necessário a utilização de uma máquina com sistema operacional MacOS, além disso, existe toda uma configuração necessária do ambiente.

3.2 MÉTODO

As principais atividades realizadas para o desenvolvimento deste trabalho foram:

a) Estudo sobre o uso do ambiente de desenvolvimento Delphi XE7

Em sua versão mais recente, a Embarcadero lançou no mercado um ambiente para desenvolvimento de aplicativos que podem ser publicados na plataforma Android e iOS, OS X e *desktop* por meio de uma única codificação, ou seja, o mesmo código é interpretado em diversas plataformas. Por ser algo novo, não existe muito conteúdo sobre o desenvolvimento de aplicativos com esse *framework*. Para isso foi necessário realizar estudo sobre sua utilização e realizar simulações a fim de analisar a viabilidade de sua utilização para o desenvolvimento deste trabalho.

b) Levantamento de requisitos

Os requisitos foram levantados e definidos tendo como base a necessidade de desenvolver um sistema para identificação de espaços acessíveis. O levantamento de requisitos foi realizado pelo autor deste trabalho a partir de estudo sobre necessidades de pessoas portadoras de deficiência motora, de forma a facilitar a identificação e locomoção principalmente em ambientes desconhecidos.

c) Planejamento

Definição do escopo do sistema.

Busca por componente de registros de pontos em mapas.

d) Desenvolvimento

Modelagem do sistema e implementação das funcionalidades utilizando o ambiente de desenvolvimento Delphi XE7, visando apresentar o uso desta tecnologia no desenvolvimento de software para dispositivos móveis.

e) Testes

Os testes de navegabilidade, funcionalidade e compatibilidade foram realizados pelo autor do trabalho. Posteriormente o software será disponibilizado para que pessoas portadoras de deficiência motora para validação por usuários.

4 RESULTADOS

Neste capítulo são apresentados os resultados da realização deste trabalho, que é a modelagem e a implementação de um sistema para identificação de espaços acessíveis, para uso em dispositivos móveis.

4.1 ESCOPO DO SISTEMA

O aplicativo para registro e visualização de pontos de acessibilidade possui duas interfaces principais. A primeira é responsável por apresentar um mapa da UTFPR câmpus Pato Branco identificando os principais pontos acessíveis como rampas, acessos, estacionamentos e outros. O aplicativo também permitirá a inclusão e a exclusão dos pontos. Para cada ponto cadastrado, será possível acrescentar uma descrição a fim de facilitar a interpretação do local cadastrado.

A segunda interface apresenta um mapa, sendo possível registrar pontos de acessibilidade em qualquer lugar do mapa. Esses pontos podem ser estabelecimentos adaptados, acessos em ruas e calçadas, banheiros, entre outros. Os pontos devem fazer referência à localização obtida por GPS.

Cada vez que o usuário utilizar alguma interface, os pontos já registrados são carregados do banco de dados e apresentados ao usuário.

Em uma primeira versão o aplicativo não irá fazer o compartilhamento dos pontos cadastrados entre usuários.

4.2 MODELAGEM DO SISTEMA

Considerando que o aplicativo não comporta processos característicos de casos de uso, como por exemplo, realização de pedidos de venda e registrar novos projetos., ou seja, não estabelece um fluxo de informação contínuo, não houve necessidade de desenvolver os casos de uso, expansões de casos de uso e demais diagramas. Assim, foi estabelecido para modelagem apenas os requisitos funcionais e não funcionais, juntamente com a especificação das tabelas de banco de dados que armazenam as informações do aplicativo.

Os requisitos levantados para o desenvolvimento do aplicativo classificam-se em duas categorias, funcionais e não funcionais. Os requisitos funcionais representam as funcionalidades consideradas necessárias para o funcionamento do sistema. Os requisitos não funcionais explicitam regras de negócio, restrições ao sistema de acesso, como por exemplo, requisitos de qualidade, desempenho, segurança e outros.

Por convenção, a referência a requisitos pode ser feita por meio do identificador do requisito, usando a sigla RF para Requisitos Funcionais e RNF para Requisitos Não Funcionais.

O Quadro 1 apresenta a listagem dos requisitos funcionais do sistema identificados para o sistema.

ID	Descrição
RF 1	O aplicativo deve apresentar o mapa da UTFPR câmpus Pato Branco, possibilitando a inclusão/exclusão de pontos acessíveis.
RF 2	O aplicativo deve apresentar um mapa para navegação possibilitando a inclusão/exclusão de pontos acessíveis.
RF 3	Os pontos acessíveis cadastrados devem ser salvos em banco de dados local permitindo o registro de uma descrição do local.
RF 4	Os pontos cadastrados devem ser apresentados quando o usuário selecionar a modalidade desejada.
RF 5	Permitir ao usuário realizar a função de zoom sobre as telas.
RF 6	Para a tela de navegação em mapa, os pontos cadastrados devem fazer referência à posição obtida pelo sensor GPS.

Quadro 1 - Requisitos funcionais

O Quadro 2 apresenta a listagem dos requisitos funcionais do sistema.

ID	Descrição
RNF 1	O aplicativo deve ser disponibilizado para plataforma Android. Se possível para a plataforma IOs
RNF 2	O aplicativo deve utilizar banco de dados para armazenamento. Se Possível Utilizar o IBlite.

Quadro 2 - Requisitos não funcionais

4.3 APRESENTAÇÃO DO SISTEMA

O aplicativo desenvolvido como resultado deste trabalho é composto por duas interfaces principais. Por meio das quais é possível realizar a navegação entre os locais e registrar os pontos de acessibilidade. Os pontos podem ser adicionados,

editados ou removidos, e em alguns casos separados por categorias. Esses pontos também podem referenciar as situações mais diversas do usuário, como registro de localidades em geral, localização em um ambiente, identificação de lugares próprios e impróprios para determinada atividade, entre outros.

A Figura 1 apresenta a tela inicial do sistema que permite acesso ao menu de funcionalidades do aplicativo.

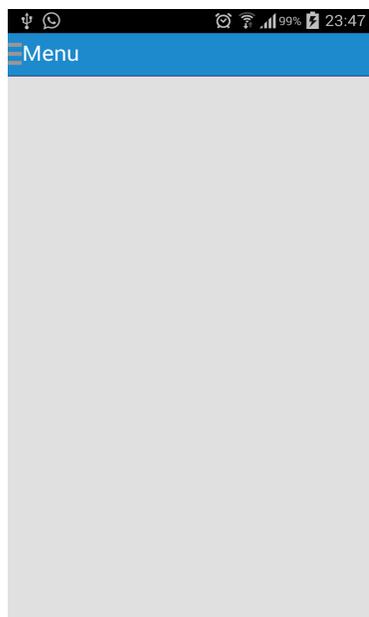


Figura 1 - Tela inicial do aplicativo

O botão Menu é visível em todas as telas do aplicativo. Este botão controla a visibilidade e apresentação/recolhimento do menu lateral do aplicativo. No canto superior está uma imagem que se refere à acessibilidade que é tema deste trabalho (área destacada na Figura 2). Nessa tela (apresentada na Figura 2) também é apresentado o nome do aplicativo: UTFPR Acessibilidade. Logo abaixo está o caminho de acesso para a utilização das duas principais funcionalidades do aplicativo. Ao clicar no botão, se o menu lateral já estiver expandido, o mesmo é recolhido, aumentando a área de atuação do usuário na funcionalidade escolhida. Caso o menu lateral esteja recolhido, o mesmo é apresentado fornecendo as opções de navegação.

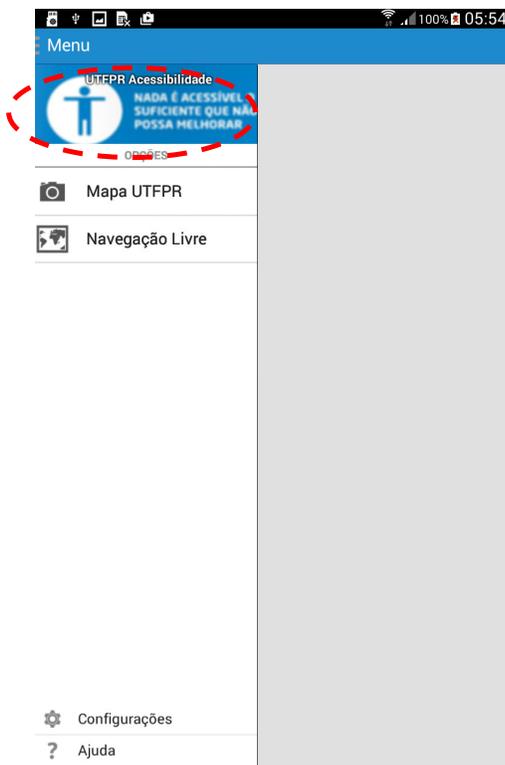


Figura 2 - Menu lateral com opções de acesso

As opções apresentadas na tela da Figura 2 são:

- Mapa UTFPR: apresenta o mapa da UTFPR Câmpus Pato Branco para navegação.
- Navegação Livre: apresenta um mapa de navegação, no mesmo padrão dos mapas apresentado pelo Google Maps.
- Configurações: apresenta as configurações do aplicativo.
- Ajuda: apresenta informações referentes ao desenvolvedor e ao projeto que originaram o aplicativo.

Ao escolher a funcionalidade de navegação livre é apresentado ao usuário um mapa para navegação, conforme apresenta a Figura 3.

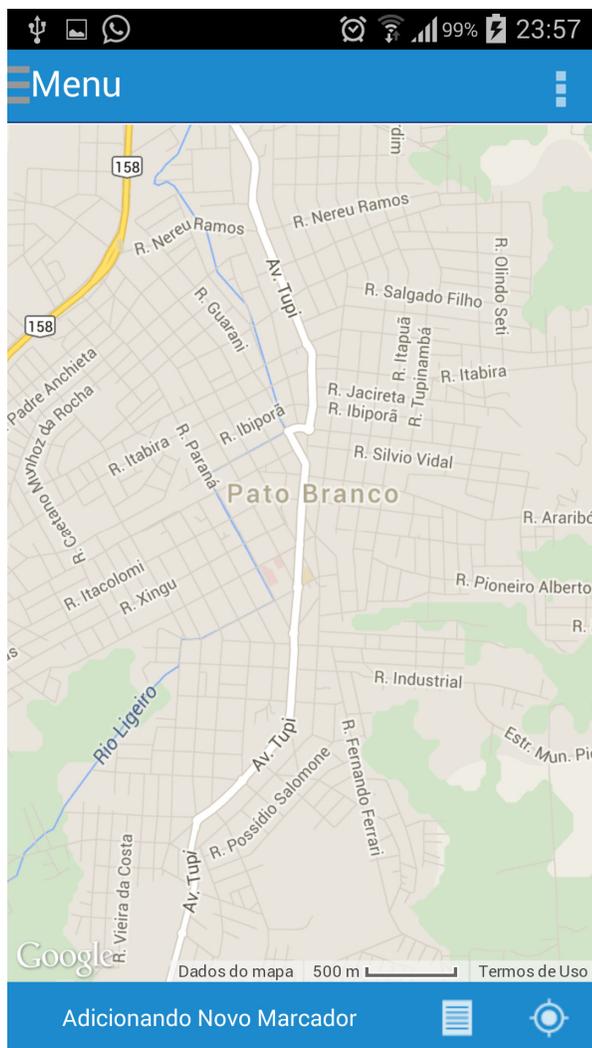


Figura 3 - Navegação livre

Neste mapa (Figura 3) é possível adicionar pontos clicando em um lugar específico. Ao clicar no ponto específico, o aplicativo aplicará um *zoom* aproximando o ponto clicado. Além disso, um informativo no rodapé do aplicativo indicará a ação do usuário como “Adicionando Novo Marcador” como apresenta a Figura 4.

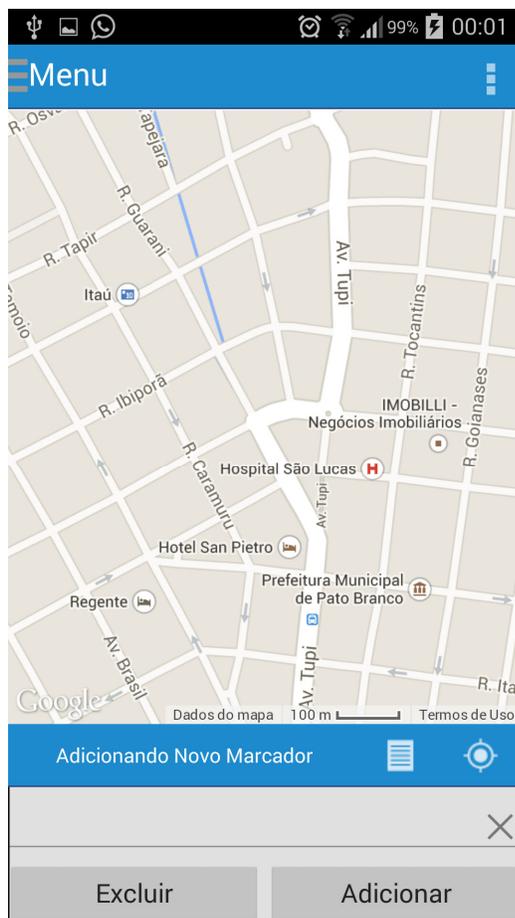


Figura 4 - Indicador de como adicionar novo Marcador

Com a realização dessa ação é apresentado ao usuário um campo para identificar o marcador antes de adicioná-lo. Nesse campo pode ser informado um texto que identifique ou explique o marcador no mapa. Após a inserção do texto informativo, o usuário deverá clicar no botão Adicionar, para que um marcador seja adicionado ao mapa como mostra a Figura 5.

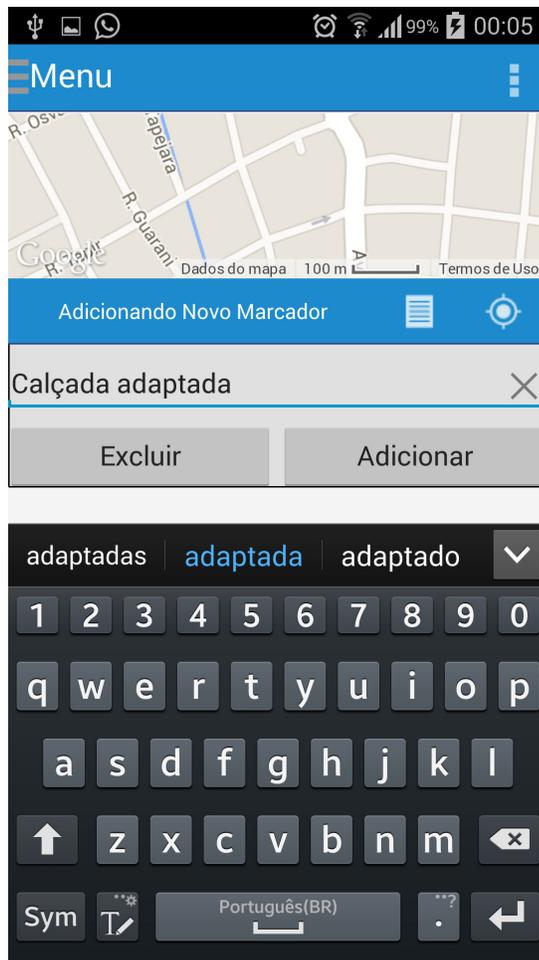


Figura 5 - Inserção de um novo ponto

Após clicar no botão Adicionar, o ponto é lançado no mapa, como mostra a Figura 6.

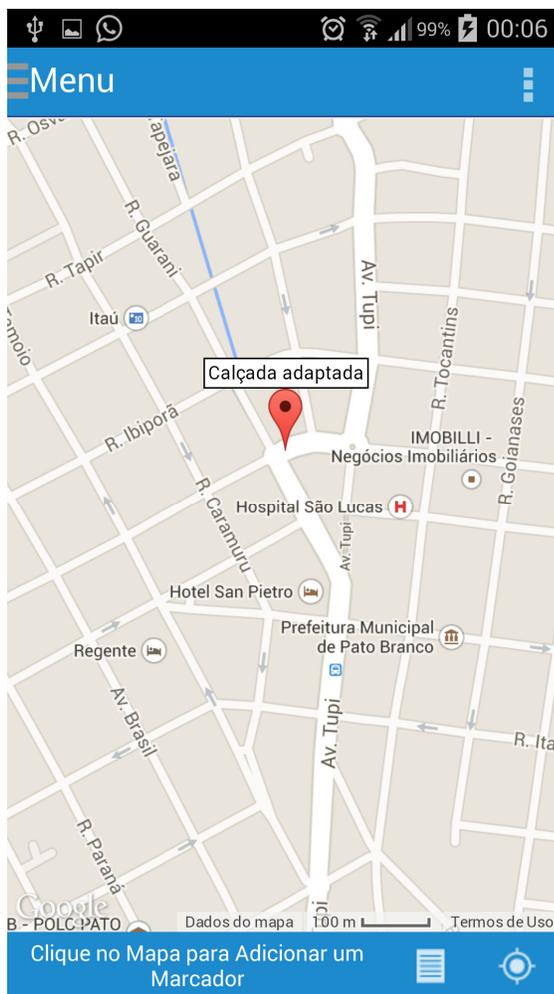


Figura 6 - Ponto lançado no mapa

Quando lançado o ponto no mapa, o indicador no rodapé indica ao usuário que deve ser clicado novamente no mapa para adicionar um novo marcador (Figura 7).



Figura 7 - Informativo adicionar marcador

Para o ponto adicionado, é apresentado um texto indicativo em cada ponto, além disso, é possível arrastá-lo pelo mapa procurando uma exatidão maior sobre o local desejado. Para isso, basta manter pressionado o toque sobre o ponto e arrastar ao local desejado. Ao tocar e soltar sobre um ponto Adicionado, na parte inferior do aplicativo são apresentadas as opções de edição e remoção do ponto, juntamente com o indicativo “Editando Marcador” (Figura 8).



Figura 8 - Edição do marcador com opção de exclusão

Neste passo é possível alterar a descrição do marcador clicando na caixa de texto e em seguida clicando no botão Adicionar para que o ponto no mapa seja atualizado com o novo texto informativo. Caso o usuário desejar remover o ponto do mapa, deverá clicar no botão Excluir. O mapa apresentado permite a adição de múltiplos pontos, como mostra a Figura 9.

No aplicativo também são apresentados dois ícones (destacados na Figura 12), no primeiro, apresenta uma lista com todos os pontos cadastrados. Esse ícone está destacado na Figura 12.

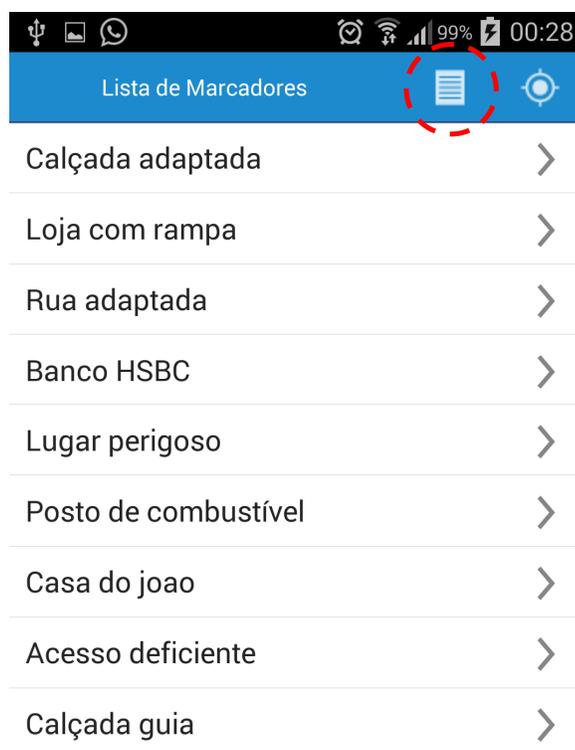


Figura 12 - Listagem de pontos cadastrados

Nesta listagem é possível retornar a tela anterior clicando no ícone da lista novamente apresentado no canto direito superior, destacado na Figura 13.



Figura 13 - Localização em foco para ser direcionada

Nesta mesma listagem ainda é possível navegar diretamente a um ponto específico já cadastrado mantendo pressionado o ponto desejado. Desta forma, o

aplicativo localizará o ponto no mapa por meio de suas coordenadas geográficas e identificá-lo no mapa (Figura 14).

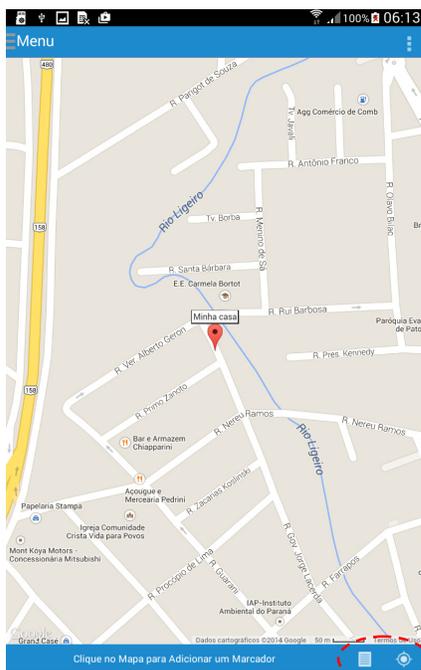


Figura 14 - Direcionamento do ponto depois de selecionado

O segundo ícone no rodapé representa a localização atual do dispositivo por meio do uso do GPS. Ao clicar no ícone, o aplicativo localiza sua posição no mapa geográfico e apresenta ao usuário aplicando um *zoom* para facilitar a visualização. O local em que se encontra o usuário é apresentado por um círculo na cor roxa (ou azul depende da visualização do aplicativo) no mapa como apresenta a Figura 15.

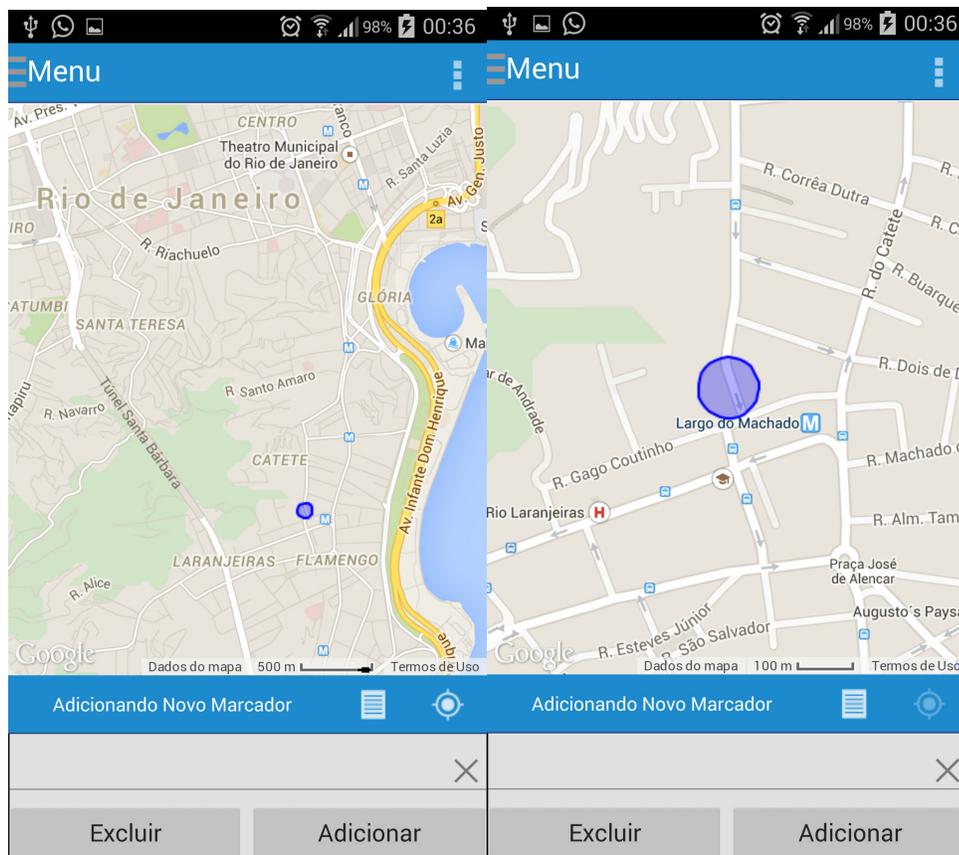


Figura 15 - Localização no mapa

Nesse caso também é apresentada a possibilidade de adicionar um ponto ao mapa, como nos demais processos apresentados. Para facilitar a navegação, o mapa no geral permite o aumentar e diminuir o zoom. Os pontos inseridos são automaticamente ajustados ao zoom aplicado no mapa (Figura 16).

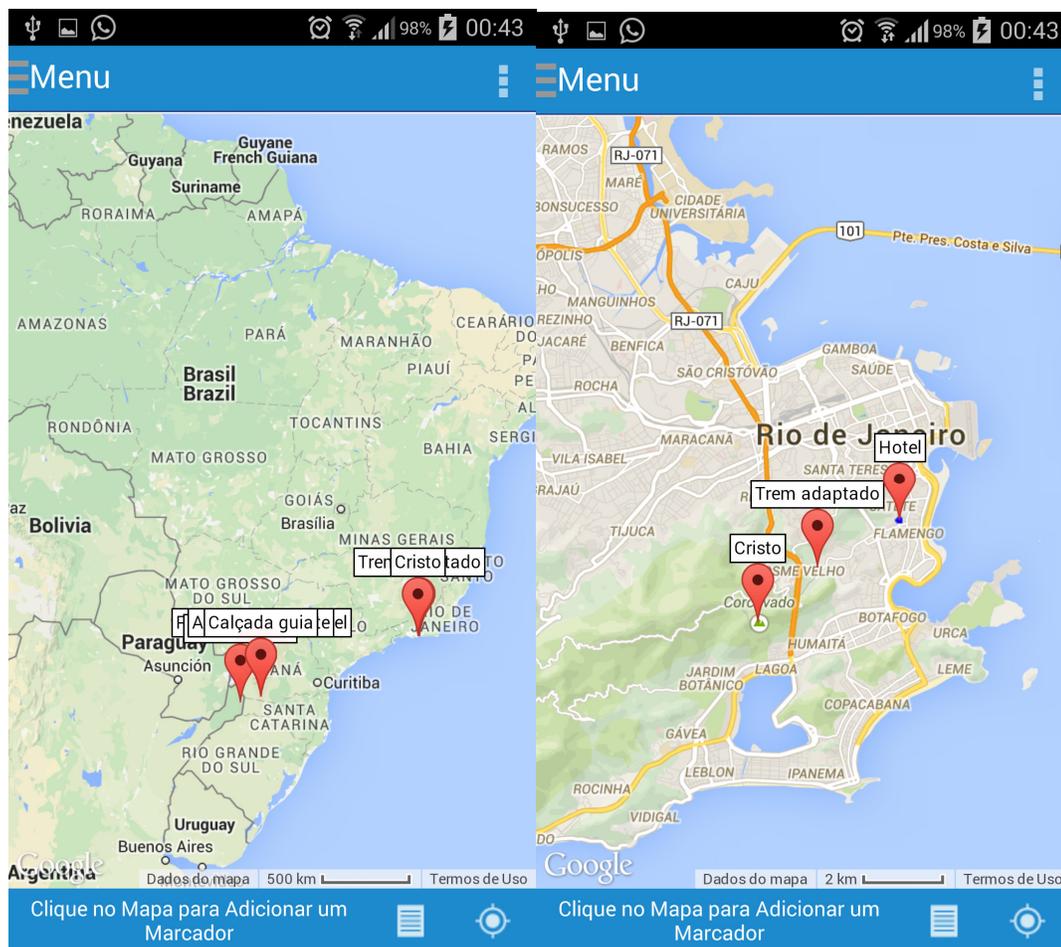


Figura 16 - Ajuste de zoom aos pontos aplicados no mapa

Ao fechar o aplicativo e abri-lo novamente, todos os pontos cadastrados permanecerão salvos.

A funcionalidade Mapa UTFPR apresenta um mapa real da estrutura do câmpus Pato branco. Neste mapa são apresentados os blocos e os caminhos que podem ser utilizados no câmpus (Figura 17).



Figura 17 - UTFPR - Câmpus Pato Branco

Neste mapa, é possível aumentar e diminuir o zoom por meio dos botões identificados com sinal de mais (+) e menos (-) localizados no canto inferior esquerdo (área destacada na Figura 17). Além disso, também é possível adicionar múltiplos pontos ao mapa. Para facilitar a identificação dos pontos foi criada uma categorização conforme apresenta a Figura 18.

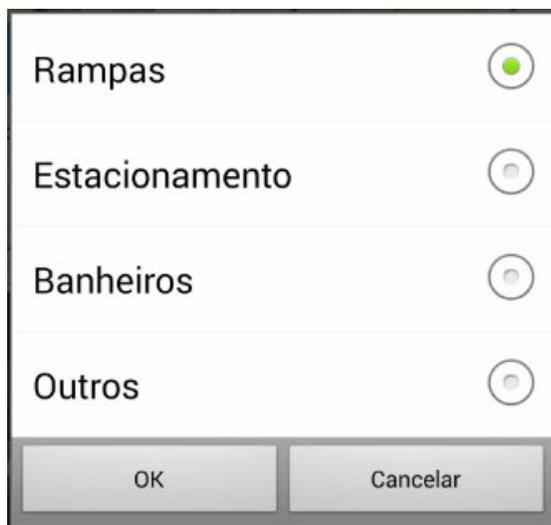


Figura 18 - Lista de categorias

As categorias padrões também são divididas por cor, sendo: rampas na cor verde, estacionamento na cor rosa, banheiros na cor azul escuro e a categoria outros na cor azul claro.

Para adicionar um marcador, o usuário deverá manter pressionado o local aproximado que deseja adicionar o ponto. Após manter pressionado por alguns instantes, a tela da Figura 19 apresentada.

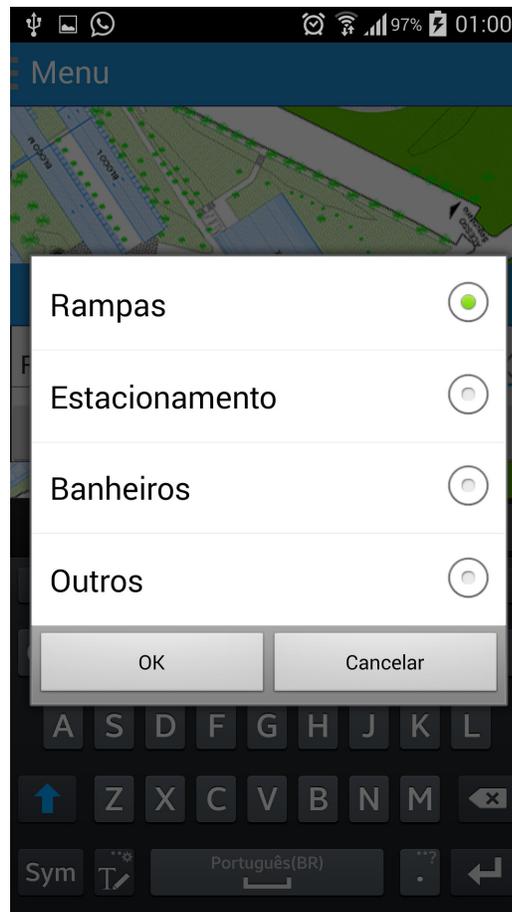


Figura 19 - Adicionando novo ponto

Ao selecionar a categoria, o usuário deverá pressionar o botão OK. Neste momento solicitará uma descrição ao ponto, o qual será adicionado (Figura 20).

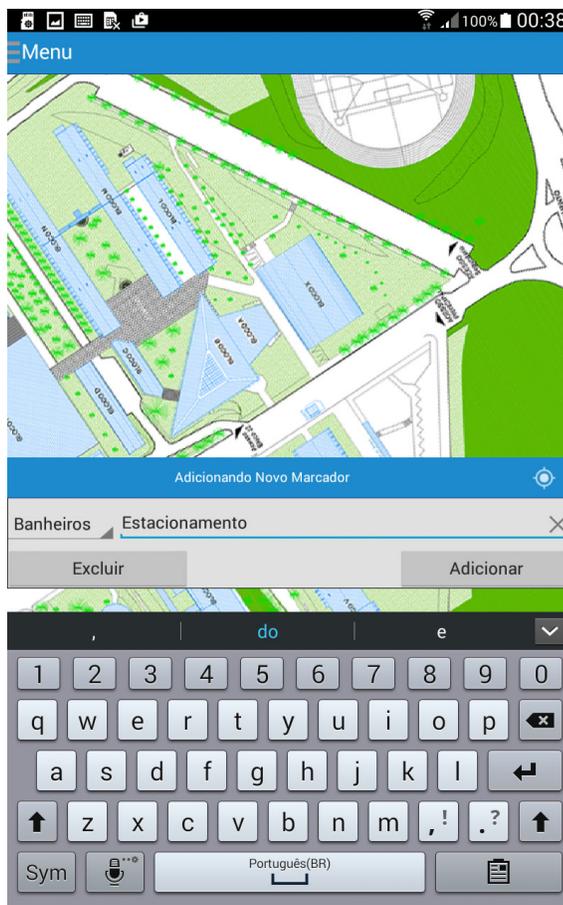


Figura 20 - Identificação do ponto

Neste momento o usuário informa o descritivo e classifica o ponto em uma das categorias disponíveis. Para concluir, é necessário clicar em Adicionar.

Com os pontos adicionados é possível realizar três operações: excluir um, editar e remover pontos. Para excluir ou editar um ponto, basta clicar sobre o ponto desejado que a tela de edição será apresentada. Nessa tela apresentada na Figura 21, o usuário pode trocar a categoria, o descritivo ou então excluir o ponto do mapa.

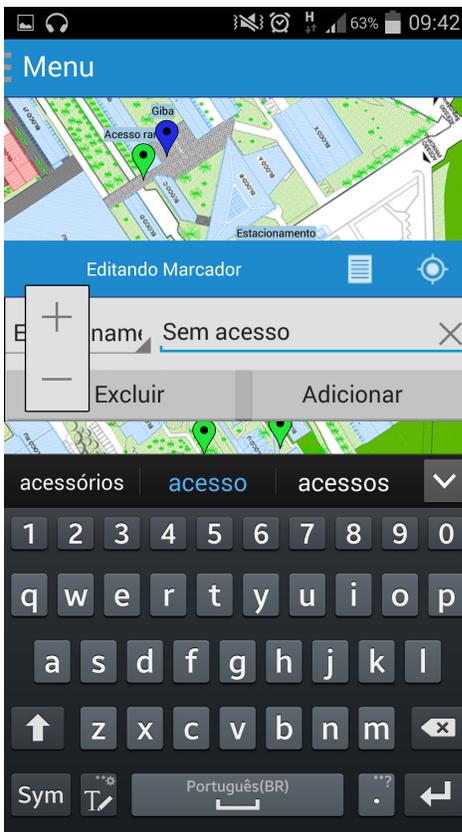


Figura 21 - Exclusão/alteração do ponto

Por último, para melhor posicionar o ponto no mapa, é possível arrastá-lo por toda a área apresentada. No momento em que o usuário soltar o ponto, este permanecerá no local indicado. Na Figura 22 é apresentado um demonstrativo de diversos pontos adicionados.



Figura 22 - Pontos adicionados

Apesar de o aplicativo ter sido desenvolvido para localizar e registrar pontos de acessibilidade, o usuário pode utilizá-lo como um guia de localização de seus interesses pessoais, como, por exemplo, localização de blocos de aula, biblioteca, cantinas, lugares de risco, entre outros.

4.4 IMPLEMENTAÇÕES DO SISTEMA

O leiaute do aplicativo é composto por duas telas principais e a tela de acesso para estas funcionalidades. A tela inicial contém o menu de navegação que permite o acesso às outras telas. O menu de navegação é disponibilizado em todas as telas, e por meio do botão 'btnMenu', este é apresentado ou ocultado do usuário. Ao clicar no botão o código da Listagem 1 é executado, definindo a visibilidade do menu lateral.

```

procedure TFMain.actShowMenuLateralExecute(Sender: TObject);
begin
  actShowMenuLateral.Enabled := False;
  actShowMenuLateral.Tag := not actShowMenuLateral.Tag;
  if Boolean(actShowMenuLateral.Tag) then begin
    RectMenu.BringToFront;
    rectMenuLateral.BringToFront;
  end;
end;

```

```

    ImgMenu.Margins.Left := 0;
    rectMenuLateral.Visible := True;
end else begin
    ImgMenu.Margins.Left := 8;
    rectMenuLateral.Visible := False;
end;
actShowMenuLateral.Enabled := True;
end;

```

Listagem 1 - Visibilidade menu lateral

Após a escolha do usuário de qual tela abrirá o aplicativo verifica se existe uma tela ativa. Caso exista uma tela ativa, a mesma é fechada e é instanciada a tela escolhida. Caso a tela escolhida já esteja instanciada (ativa), somente é configurada a visibilidade e a tela é rerepresenta. Este funcionamento é padrão para todas as telas. Na Listagem 1 está o código para abertura da tela de navegação livre (Listagem 2).

```

procedure TFMain.liGMapsClick(Sender: TObject);
begin
    if Assigned(FMapImage) then
        FMapImage.Close;
    if not Assigned(FMapGMaps) then
        Application.CreateForm(TFMapGMaps, FMapGMaps);
    actShowMenuLateral.Execute;
    FMapGMaps.TMSFMXWebGMaps1.Visible := True;
    FMapGMaps.TMSFMXWebGMaps1.Repaint;
    FMapGMaps.liGMaps.IsSelected := True;
    FMapGMaps.Show;
end;

```

Listagem 2 - Abertura de tela navegação livre

Na Listagem 3 está a abertura da tela para navegação no Câmpus Pato Branco da UTFPR.

```

procedure TFMain.liImagemClick(Sender: TObject);
begin
    if Assigned(FMapGMaps) then
        FMapGMaps.Close;
    if not Assigned(FMapImage) then
        Application.CreateForm(TFMapImage, FMapImage);
    actShowMenuLateral.Execute;
    FMapImage.liImagem.IsSelected := True;
    FMapImage.Show;
end;

```

Listagem 3 - Abertura de tela navegação Câmpus Pato Branco

Uma das principais funcionalidades do aplicativo é a tela em que o usuário navega livremente pelo mapa adicionando pontos a ele. Para a apresentação do mapa, foi utilizado um componente chamado “TMSFMXWebGMaps” do pacote de componentes da “TTMSFMXWebGMaps”. Com este componente é possível

adicionar “*markers*” (pontos de localização na tela) ao mapa. Todo ponto adicionado se baseia no registro de suas coordenadas geográficas, latitude e longitude. Ao adicionar um ponto ao mapa são registradas em um banco de dados de nome ‘Maps.wmap’ as informações referentes ao ponto, como Latitude, Longitude, Título, Index. Ao abrir a tela, os pontos já registrados são carregados para o ‘cdsPontos’ (componente da classe TClientDataSet, responsável por receber as informações do arquivo de dados e apresenta-las em forma de registros que possam ser manipulados) e adicionados ao mapa.

A Listagem 4 apresenta o código responsável por carregar os pontos já existentes.

```

procedure TMapGMaps.CarregarMarcadores;
begin
  if FileExists(GetHomePath + '/Maps.wmap') then begin
    try
      if TMSFMXWebGMaps1.Markers.Count > 0 then
        TMSFMXWebGMaps1.Markers.Clear;
      cdsPontos.EmptyDataSet;
      cdsPontos.LoadFromFile(GetHomePath + '/Maps.wmap');
    except
      cdsPontos.Close;
      cdsPontos.CreateDataSet;
    end;
  end else begin
    cdsPontos.Close;
    cdsPontos.CreateDataSet;
    cdsPontos.EmptyDataSet;
  end;
  if (cdsPontos.RecordCount > 0) then begin
    Operacao := MarkerOpcao.Adicionando;
    cdsPontos.First;
    while not cdsPontos.Eof do begin
      cdsPontos.Edit;
      cdsPontos.Post;
      cdsPontos.Next;
    end;
  end;
end;

```

Listagem 4 - Carregamento dos pontos já existentes

Nessa etapa é verificado se o arquivo de banco de dados ‘Maps.wmap’ existe. Caso não exista, o componente cdsPontos é criado em memória e, posteriormente, receberá os pontos adicionados pelo usuário. Caso o arquivo de banco já exista, suas informações são carregadas para o cdsPontos por meio da função LoadFromFile. A segunda etapa consiste em adicionar os pontos ao Mapa, pois até o momento eles estão apenas carregados ao componente cdsPontos. Para isso, caso o cdsPontos tenha registros, é realizada uma ação de edição e confirmação de

alteração (cdsPontos.Post) para que os eventos do componente sejam executados. Ao executar a linha de comando cdsPontos.Post, o evento BeforePost do componente é executado. Este evento é responsável por adicionar o ponto ao mapa, uma vez que as informações necessárias já estão em formato de registro, divididos em campos no componente. A Listagem 5 apresenta o código executado no evento BeforePost.

```

procedure TFMapGMaps.cdsPontosBeforePost(DataSet: TDataSet);
var
  vMarker: TMarker; i: Integer;
begin
  inherited;
  if Operacao = MarkerOpcao.Adicionando then begin
    vMarker :=
TMSFMXWebGMaps1.Markers.Add(cdsPontosla.AsFloat,cdsPontoslo.AsFloat,cdsPont
osTitle.AsString,',',True,True,
    True, False, True, 0);
    vMarker.MapLabel.Text := cdsPontosTitle.AsString;
    if cdsPontosIndex.AsInteger <= 0 then begin
      Operacao := MarkerOpcao.Nenhuma;
      if TMSFMXWebGMaps1.Markers.Count >= 2 then
        cdsPontosIndex.AsInteger :=
TMSFMXWebGMaps1.Markers[TMSFMXWebGMaps1.Markers.Count - 2].Tag + 1
      else
        cdsPontosIndex.AsInteger := 1;
      Operacao := MarkerOpcao.Adicionando;
    end;
    vMarker.Tag := cdsPontosIndex.AsInteger;
  end
  else if Operacao = MarkerOpcao.Editando then begin
    for i := 0 to TMSFMXWebGMaps1.Markers.Count - 1 do
      if TMSFMXWebGMaps1.Markers[i].Tag = cdsPontosIndex.AsInteger then begin
        TMSFMXWebGMaps1.Markers[i].MapLabel.Text := cdsPontosTitle.AsString;
        Break;
      end;
    end;
  end;
end;

```

Listagem 5 - Evento BeforePost do cdsPontos

O evento BeforePost é, basicamente, dividido em duas partes, a edição do ponto localizado no *marker* (ponto) do mapa e atualização da descrição do ponto com base no atributo Tag, que vincula o ponto da lista de pontos do mapa com o registro no cdsPontos. A outra parte é a inserção de um novo *marker*, para isso há uma variável do tipo TMarker que recebe o endereço de memória do *marker* instanciando no Componente TMSFMXWebGMaps1 por meio da função Markers.Add(). Após instanciado, é definido o número do *index* que este *marker* receberá de acordo com os demais pontos que já foram adicionados. Em seguida, o *index* atribuído no registro do cdsPontos, também é atribuído a propriedade TAG no componente, para que eles fiquem sincronizados.

Quando o usuário está adicionado um novo *marker* no mapa e clica no botão Adicionar, o aplicativo executa os mesmos eventos descritos no parágrafo anterior. Primeiro, define a descrição do Ponto e depois executa o evento BeforePost por meio do comando `cdsPontos.Post`. Sempre que um novo ponto é adicionado, é necessário ter a informação da sua latitude e longitude. Para isso sempre que o usuário clica no mapa, essas duas informações já são preenchidas, para que quando o usuário clicar em Adicionar, o registro no `cdsPontos` já contenha essas duas informações. O código para adicionar marcador no mapa é apresentado na Listagem 6.

```
procedure TFMMapGMaps.btAdicionarClick(Sender: TObject);
begin
  inherited;
  if Operacao = MarkerOpcao.Nenhuma then
    Operacao := MarkerOpcao.Adicionando;
  cdsPontos.Edit;
  cdsPontos.Title.AsString := edtDescricao.Text;
  cdsPontos.Post;
  cdsPontos.SaveToFile(GetHomePath + '/Maps.wmap');
  OcultarBarraEdicaoInsercaoMarcador;
  txtAjuda.Text := 'Clique no Mapa para Adicionar um Marcador';
  Operacao := MarkerOpcao.Nenhuma;
end;
```

Listagem 6 - Comando para adicionar marcador ao mapa

Assim que o ponto é adicionado/atualizado no mapa, o aplicativo salva as alterações no arquivo de dados por meio do comando `cdsPontos.SaveToFile()`. Vale lembrar que quando clicado no botão adicionar, a latitude e longitude necessárias para criar o ponto no mapa já estão salvas no componente `cdsPontos`, como apresenta a listagem 7.

```
cdsPontos.FieldName('la').AsFloat := Latitude;
cdsPontos.FieldName('lo').AsFloat := Longitude;
```

Listagem 7 – Salvamento de latitude e longitude

As informações de latitude e longitude são obtidas no evento `MapClick` do componente `TMSFMXWebGMaps`.

Quando o usuário utiliza a opção de exclusão do ponto, este é removido do `cdsPontos`, por meio do comando `cdsPontos.Delete`. A Listagem 8 apresenta o código para a remoção de um ponto do mapa.

```

procedure TFMapGMaps.btRemoverClick(Sender: TObject);
begin
  inherited;
  cdsPontos.Delete;
  OcultarBarraEdicaoInsercaoMarcador;
  txtAjuda.Text := 'Clique no Mapa para Adicionar um Marcador';
end;

```

Listagem 8 - Comando de remoção de pontos

Ao executar o comando `cdsPontos.delete`, o evento `BeforeDelete` é acionado. Neste evento, o ponto que até o momento estava sendo removido apenas do `cdsPontos`, é localizado na listagem de pontos do mapa através da propriedade TAG e removido (Listagem 9).

```

procedure TFMapGMaps.cdsPontosBeforeDelete(DataSet: TDataSet);
var
  i: Integer;
begin
  inherited;
  for i := 0 to TMSFMXWebGMaps1.Markers.Count - 1 do
    if TMSFMXWebGMaps1.Markers[i].Tag = cdsPontosIndex.AsInteger then begin
      TMSFMXWebGMaps1.Markers.Delete(i);
      Break;
    end;
  end;
end;

```

Listagem 9 - Remoção do ponto da lista de pontos do mapa

Na listagem dos pontos cadastrados no mapa, ao pressionar um ponto em específico, o aplicativo direciona o usuário até este ponto e aplica um *zoom* para facilitar a visualização. Para isso, a listagem de pontos é apresentada em um componente do tipo `TListView` e as ações que ocorrem sobre este `ListView` de nome `lvMarcadores` são mapeadas, conforme apresenta a Figura 23.

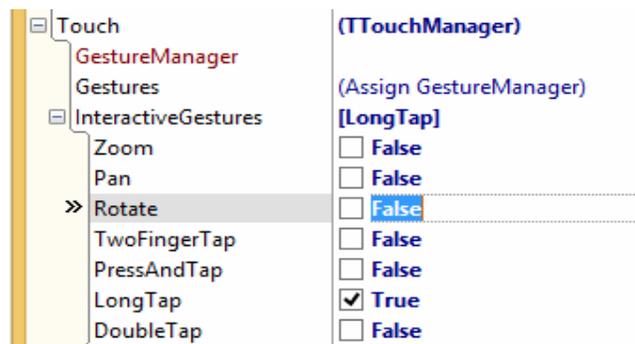


Figura 23 - Ações que estão definidas para serem mapeadas

O conjunto de ações apresentadas no grupo 'Interactive Gestures' representa as ações disponíveis que podem ser mapeadas no componente. Essas ações também podem ser mapeadas em outros componentes, como, por exemplo, em um

TImage. Neste caso, decidiu-se que apenas a ação LongTap, ou seja, quando o usuário manter pressionado por um intervalo de tempo maior, será mapeado.

Após definir as ações que serão mapeadas, é necessário indicar para o aplicativo o que deve ser feito quando uma determinada ação acontecer. Para isso o evento OnGesture do lvMarcadores (TListView) é programado (Listagem 10).

```

procedure TMapGMaps.lvMarcadoresGesture(Sender: TObject;
  const EventInfo: TGestureEventInfo; var Handled: Boolean);
var
  bounds : TBounds;
begin
  inherited;
  if (EventInfo.GestureID = System.UITypes.igiLongTap) then begin
    cdsPontos.RecNo := lvMarcadores.ItemIndex + 1;
    bounds := FMX.TMSWebGMapsCommonFunctions.TBounds.Create;
    bounds.NorthEast.Latitude := cdsPontosla.AsFloat + 0.002;
    bounds.NorthEast.Longitude := cdsPontoslo.AsFloat + 0.002;
    bounds.SouthWest.Latitude := cdsPontosla.AsFloat - 0.002;
    bounds.SouthWest.Longitude := cdsPontoslo.AsFloat - 0.002;
    TMSFMXWebGMaps1.MapZoomTo(bounds);
    ShowListClick(ShowList);
  end;
end;

```

Listagem 10 - Programação das ações

Neste caso, está sendo verificado se a ação ocorrida é a que está sendo mapeada. Em resultado positivo, é aplicado um *zoom* ao ponto selecionado pelo usuário na lista apresentada. O *zoom* é aplicado por meio da função MapZoomTo() disponível no componente TMSFMXWebGMaps. Em seguida, a lista de pontos é fechada e o ponto escolhido é apresentado no mapa.

Outra funcionalidade desta tela é o acompanhamento da localização atual em que o usuário se encontra. Para isso é necessário clicar na imagem de localizador (GPSLocation). No evento onClick da imagem foi programada a função de ativar o GPS do dispositivo (Listagem 11).

```

procedure TMapGMaps.GPSLocationClick(Sender: TObject);
begin
  inherited;
  GpsAnimation.Enabled := True;
  if not LocationSensor1.Active then begin
    LocationSensor1.Active := True;
  end;
end;

```

Listagem 11 - Ativação do GPS do dispositivo

Uma vez ativado o sensor de localização, o evento LocationChanged é executado a cada vez que a localização do dispositivo mudar. Neste caso, esse evento é utilizado para desenhar um círculo no local em que o dispositivo é

encontrado pelo GPS. Além disso, quando encontrado o dispositivo, é aplicado um *zoom* sobre as coordenadas recebidas (local do dispositivo), como apresentado na Listagem 12.

```

procedure TFMapGMaps.LocationSensor1LocationChanged(Sender: TObject;
const OldLocation, NewLocation: TLocationCoord2D);
var
  bounds: FMX.TMSWebGMapsCommonFunctions.TBounds;
begin
  inherited;
  LocationSensor1.Active := False;
  TMSFMXWebGMaps1.MapPanTo(NewLocation.Latitude, NewLocation.Longitude);
  bounds := FMX.TMSWebGMapsCommonFunctions.TBounds.Create;
  bounds.NorthEast.Latitude := NewLocation.Latitude + 0.002;
  bounds.NorthEast.Longitude := NewLocation.Longitude + 0.002;
  bounds.SouthWest.Latitude := NewLocation.Latitude - 0.002;
  bounds.SouthWest.Longitude := NewLocation.Longitude - 0.002;
  TMSFMXWebGMaps1.MapZoomTo(bounds);
  if not Assigned(PolygonLocationItem) then begin
    PolygonLocationItem := TMSFMXWebGMaps1.Polygons.Add;
    CircleLocation := PolygonLocationItem.Polygon;
    CircleLocation.PolygonType :=
FMX.TMSWebGMapsCommon.TPolygonType.ptCircle;
    CircleLocation.BackgroundOpacity := 80;
    CircleLocation.BorderWidth := 2;
  end;
  CircleLocation.Radius := 45;
  CircleLocation.Visible := True;
  CircleLocation.Center.Latitude := NewLocation.Latitude;
  CircleLocation.Center.Longitude := NewLocation.Longitude;
  TMSFMXWebGMaps1.DeleteAllMapPolygon;
  TMSFMXWebGMaps1.CreateMapPolygon(CircleLocation);
  GpsAnimation.Enabled := False;
  GPSLocation.Opacity := 1;
  GPSLocation.Repaint;
end;

```

Listagem 12 - Definição da localização atual

Caso já exista um círculo indicando a localização, este é removido e um novo é adicionado com as coordenadas atuais. O componente responsável por obter a localização atual do dispositivo é o 'LocationSensor1', que desempenha este papel ao ser ativado.

Os *markers* (pontos) adicionados ao mapa, apresentam a funcionalidade de ser arrastados para atingir uma precisão maior do lugar escolhido. Para isso, foi implementado o evento OnMarkerDragEnd do componente TMSFMXWebGMaps, fazendo com que as localizações finais do ponto sejam atualizadas no arquivo de dados. A implementação desse evento é apresentada na Listagem 13.

```

procedure TFMapGMaps.TMSFMXWebGMaps1MarkerDragEnd(Sender: TObject;
  MarkerTitle: string; IdMarker: Integer; Latitude, Longitude: Double);
begin
  inherited;

```

```

Operacao := MarkerOpcao.Editando;
cdsPontos.First;
cdsPontos.FindKey ([TMSFMXWebGMaps1.Markers[IdMarker].Tag]);
cdsPontos.Edit;
cdsPontos.FieldName('la').AsFloat := Latitude;
cdsPontos.FieldName('lo').AsFloat := Longitude;
cdsPontos.Post;
cdsPontos.SaveToFile(GetHomePath + '/Maps.wmap');
end;

```

Listagem 13 - Evento em que o marcador estava sendo movido

A funcionalidade Mapa UTFPR, a segunda interface do aplicativo, consiste na apresentação do mapa da UTFPR câmpus Pato Branco. Neste mapa fornecido pelo setor de Engenharia da Universidade, estão apresentados os blocos, bem como, os caminhos e a representação de todos os edifícios do câmpus até a presente data em que o material foi fornecido. Esta interface também permite ao usuário a adição de pontos com base nas coordenadas geográficas, categorizando esses pontos conforme a necessidade.

Para o desenvolvimento da funcionalidade Mapa UTFPR foi necessário implementar um componente que herde a estrutura básica e o funcionamento de um componente TImage. A Figura 24 apresenta a estrutura do componente desenvolvido.

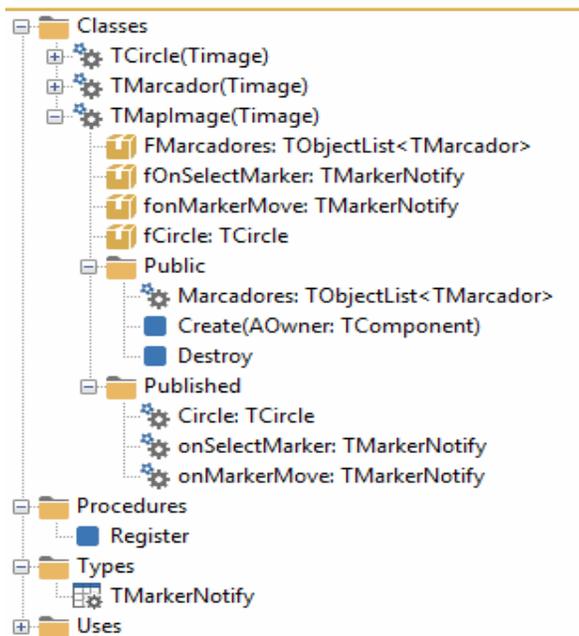


Figura 24 – Estrutura do componente TMapImage

O componente TMapImage foi desenvolvido pelo autor desde trabalho com o objetivo de facilitar as operações durante a utilização do aplicativo. Uma de suas funcionalidades é para instanciar os pontos no mapa, ou seja, para armazenar os

pontos adicionados ao mapa. Pra isso foi criado uma lista da classe T Marcador, chama F Marcadores. Cada ponto adicionado ao mapa, corresponde a um objeto instanciado da classe T marcador adicionado na lista F Marcadores. A Listagem 14 apresenta a codificação do componente.

```
//classe do componente TMapImage
TMapImage = class(TImage)
private
  { Private declarations }
  FMarcadores: TObjectList<TMarcador>;
  fOnSelectMarker: TMarkerNotify;
  fonMarkerMove: TMarkerNotify;
  fCircle: TCircle;
protected
  { Protected declarations }
public
  { Public declarations }
  constructor Create(AOwner: TComponent); override;
  destructor Destroy; override;
  Property Marcadores: TObjectList<TMarcador> read FMarcadores
    write FMarcadores;
published
  property Circle: TCircle read fCircle;
  property onSelectMarker: TMarkerNotify read fOnSelectMarker
    write fOnSelectMarker;
  property onMarkerMove: TMarkerNotify read fonMarkerMove write
fonMarkerMove;
  { Published declarations }
end;
```

Listagem 14 - Classe TMapImage

O componente MapImage é herdado da classe TImage. Dentro da classe MapImage há uma lista de marcadores do tipo T Marcador. Cada marcador é um ponto adicionado ao mapa apresentado que também é herdado da classe TImage. Além disso, há também um objeto do tipo TCircle que apresenta a localização atual no mapa. A classe T Marcador é apresentada na Listagem 15.

```
//classe do marcador
TMarcador = class(TImage)
private
  fCaption: String;
  function GetX: Single;
  function GetY: Single;
  procedure SetX(const Value: Single);
  procedure SetY(const Value: Single);
  procedure SetCaption(const Value: String);
protected
  FLastPosition: TPointF;
  procedure Paint; override;
  procedure Click; override;
  procedure DoGesture(const EventInfo: TGestureEventInfo;
    var Handled: Boolean); override;
public
  property Caption: String read fCaption write SetCaption;
  constructor Create(AOwner: TComponent; X, Y: Single; ACaption: String;
```

```

    Categoria: Integer); reintroduce;
published
    property X: Single read GetX write SetX;
    property Y: Single read GetY write SetY;
end;

```

Listagem 15 - Classe TMarcador

Cada marcador contém o atributo chamado *Caption* que recebe a identificação do marcador. As funções *Get* e *Set* de *X* e *Y* são necessárias ao utilizar as posições do marcador, pois ao utilizar o *X* e *Y* do marcador o seu tamanho deve ser considerado. Essas funções retornam as posições corretas descontando o tamanho (*X,Y*) do objeto marcador. Ao ser instanciado, o marcador recebe por parâmetro a localização em que deve ser criado, além de receber o texto descritivo e qual categoria pertence. O código do componente *create* do componente *Tmarcador* é apresentado na Listagem 16.

```

constructor TMarcador.Create(AOwner: TComponent; X, Y: Single; ACaption:
String;
    Categoria: Integer);
var
    InStream: TResourceStream;
begin
    inherited Create(AOwner);
    Self.Parent := TFmxObject(AOwner);
    Self.Height := 60;
    Self.Width := 30;
    Self.X := X;
    Self.Y := Y;
    Self.Anchors := [];
    InStream := TResourceStream.Create(HInstance,
    'PngImage_' + Categoria.ToString(), RT_RCDATA);
    Self.MultiResBitmap.Items[0].Bitmap.LoadFromStream(InStream);
    Self.Caption := ACaption;
    Self.TouchManager.InteractiveGestures := [TInteractiveGesture.Pan,
    TInteractiveGesture.LongTap];
end;

```

Listagem 16 - Create do componente TMarcador

Como esse objeto é criado somente em tempo de execução, a imagem que representa ele na tela esta adicionada como *resource* no projetoLo e é obtida por meio do carregamento em *Stream*. Para o marcador, também é habilitado quais são as ações que ele irá responder, neste caso, estão ativadas as ações *LongTap* e *Pan*. Quando adicionado à tela, é necessário desenhar a descrição do marcador sobre ele. O código da Listagem 17 se refere a funcionalidade de desenhar o identificador sobre o marcador. Por meio do uso do *canvas* é desenhado a descrição sobre ponto no mapa.

```

procedure TMarcador.Paint;
var
  vRect: TRectF;
  vTextWidth: Single;
begin
  inherited;
  Canvas.Stroke.Kind := TBrushKind.bkSolid;
  Canvas.Stroke.Color := $FFA4C9DF;
  Canvas.Stroke.Thickness := 3;
  Canvas.Fill.Color := $FF000202;
  Self.Canvas.BeginScene();
  vTextWidth := Self.Canvas.TextWidth(Self.Caption) + 2;
  vRect := RectF(10 - (vTextWidth / 2), -15, (vTextWidth / 2) + 10, 4);
  Self.Canvas.FillRect(vRect, 2, 2, AllCorners, 100, Canvas.Stroke);
  Self.Canvas.FillText(vRect, Self.Caption, false, 100, [],
  TTextAlign.Center);
  Self.Canvas.EndScene();
end;

```

Listagem 17 - Desenhando identificador sobre o marcador

Outro Objeto da classe MapImage é o Circle, também herdado da classe TImage, descrito na Listagem 18.

```

//Objeto que apresenta a posicao atual obtida por GPS
TCircle = class(Timage)
  function GetX: Single;
  function GetY: Single;
  procedure SetX(const Value: Single);
  procedure SetY(const Value: Single);
public
  property X: Single read GetX write SetX;
  property Y: Single read GetY write SetY;
  constructor Create(AOwner: TComponent);
end;

```

Listagem 18 - Classe TCircle

O objeto TCircle contém apenas as coordenadas em que será apresentado no mapa. Quando instanciado, somente recebe a imagem do *resource* e fica visível na tela (Listagem 19).

```

constructor TCircle.Create(AOwner: TComponent);
var
  InStream: TResourceStream;
begin
  inherited Create(AOwner);
  Self.Parent := TFmxObject(AOwner);
  Self.Height := 15;
  Self.Width := 15;
  Self.Anchors := [];
  InStream := TResourceStream.Create(HInstance, 'MapCircle', RT_RCDATA);
  Self.MultiResBitmap.Items[0].Bitmap.LoadFromStream(InStream);
end;

```

Listagem 19 - Evento Create da classe TCircle

Com o componente desenvolvido, foi criado a Unit MapImage e adicionado o componente MapImage1 na tela. A imagem da UTFPR foi adicionada diretamente ao componente MapImage1 presente na Unit MapImage. Os pontos adicionados ao mapa são salvos da mesma forma que na primeira interface (navegação livre). No entanto, nesse caso, as informações são salvas em arquivo de dados chamado 'MapaUtfpr.wmap', utilizando o mesmo componente TClientDataSet, de nome cdsDados. Para esta interface, além da Latitude, Longitude, Título, Index, a categoria do ponto também é salva. A Figura 25 apresenta os campos salvos no banco de dados.



Figura 25 - Campos salvos no banco de dados

Para adicionar o ponto ao mapa, é necessário manter pressionado por alguns instantes sobre o local que deseja adicionar o ponto. Neste caso, o evento 'Gesture', cujo código é apresentado na Listagem 20, do componente MapImage1 é executado.

```

procedure TFMapImage.MapImage1Gesture(Sender: TObject;
  const EventInfo: TGestureEventInfo; var Handled: Boolean);
begin
  inherited;
  if (EventInfo.GestureID = System.UITypes.igiLongTap) then begin
    Handled := True;
    txtAjuda.Text := 'Adicionando Novo Marcador';
    EdtDescricao.Text := '';
    cbCategorias.DropDown;
    MostrarBarraEdicaoInsercaoMarcador;
    EdtDescricao.SetFocus;
  end;
end;

```

Listagem 20 - Mapeando evento do componente

Neste caso, está sendo mapeado o LongTap. Caso ele ocorra, a lista de categorias do Ponto será exibida por meio do comando `cbCategoria.DropDown` (comboBox). Juntamente com a listagem das categorias, pelo procedimento 'MostrarBarraEdicaoInsercaoMarcador' é apresentado o campo para preencher a descrição do ponto. Neste mesmo procedimento é apresentada, também, a opção de excluir/editar o ponto no mapa.

Após informadas a descrição e categoria do ponto, o evento `BeforePost` do `cdsDados` será executado, adicionando o ponto a lista de marcadores pertencentes ao componente `MapImage1`. Da mesma forma que é realizada na interface de navegação livre, o atributo `TAG` do componente *marker* (ponto) é utilizada para sincronizar com o registro no `cdsDados`.

Quando selecionado um ponto, por meio de clique sobre ele, o evento `SelectMarker` é executado, fazendo com que o texto de ajuda no rodapé seja alterado e possibilitando manutenção do ponto.

A Listagem 21 apresenta o código para apresentação da Ajuda e edição do *Marker*.

```

procedure TFMapImage.MapImage1SelectMarker(marker: TMarcador);
begin
  inherited;
  Operacao := MarkerOpcao.Editando;
  cdsDados.First;
  cdsDados.FindKey([marker.Tag]);
  txtAjuda.Text := 'Editando Marcador';
  EdtDescricao.Text := cdsDadosTitle.AsString;
  MostrarBarraEdicaoInsercaoMarcador;
  Operacao := MarkerOpcao.Nenhuma;
end;

```

Listagem 21 - Apresentando ajuda e editando *marker*

Uma das principais usabilidades da interface de Localização GPS no Câmpus

é a obtenção da localização do dispositivo dentro do câmpus. Quando o usuário clica na imagem do localizador (GPSLocation), ele ativa o GPS que busca pela localização atual do dispositivo. Diferentemente da primeira interface, o componente 'LocationSensor1', responsável pela localização do dispositivo fica ativo, mantendo atualizada a localização conforme é alterada. Para tornar essa alteração de localização visível, no evento OnLocationChanged do componente LocationSensor1, o procedimento AtualizaLocalGPS é executado. O código para a atualização do dispositivo é apresentado na Listagem 22.

```

procedure TFMapImage.AtualizaLocalGPS;
const
  totWid = 0.009452;
  totHeig = 0.005901;
  LongIni = -52.695230; // final -52.685778
  latIni = -26.199538; // final -26.193637
var
  dif: Single;
begin
  inherited;
  dif := edtlong.Text.ToSingle() - LongIni;
  MapImage1.Circle.X := (dif / totWid) * MapImage1.Width;
  dif := edtlat.Text.ToSingle() - latIni;
  MapImage1.Circle.Y := MapImage1.Height - (dif / totHeig) *
  MapImage1.Height;
end;

```

Listagem 22 - Atualização do dispositivo

Conforme as coordenadas da localização atual mudam, o Circle (objeto do componente MapImage1) recebe a atualização das posições, se movendo conforme necessário. Para isso, foi necessário realizar um cálculo para obter a proporção da distância percorrida nas coordenadas em relação ao tamanho do mapa.

5 CONCLUSÃO

Para portadores de necessidades especiais a acessibilidade promove a participação social de todas as pessoas na sociedade, uma vez que viabiliza o acesso e uso de ambientes e equipamentos não disponíveis, sem recursos específicos. Acessibilidade implica equiparação de oportunidades em todos os espaços da vida.

O uso da tecnologia voltada para a acessibilidade favorece e facilita a reintegração e aprendizado de portadores de necessidades especiais, uma vez que rompe barreiras e limitações antes existentes. Assim verificou-se a oportunidade de desenvolver um aplicativo para dispositivos móveis que facilitasse a acessibilidade no câmpus Pato Branco da UTFPR.

O aplicativo foi desenvolvido conforme planejando e foram implementadas as funcionalidades pretendidas para a identificação de pontos no mapa que indiquem locais que atendem às especificações de acessibilidade como rampas de acesso e banheiros adaptados. No mapa, também, podem ser pontuadas localizações de perigo no sentido de restringirem ou mesmo impedirem a mobilidade e o acesso. A criação do portfólio se refere aos pontos armazenados pelo próprio usuário. Esses pontos podem ser compartilhados com outros usuários.

Com o aplicativo proposto espera-se proporcionar maior independência aos portadores de deficiência motora, de forma que possam se orientar melhor em ambientes desconhecidos, identificando facilmente a localização dos espaços acessíveis disponíveis. Além disso, será possível ao usuário criar um portfólio pessoal de experiências, e compartilhar com outras pessoas com necessidades especiais.

Com base no aplicativo desenvolvido são indicadas algumas melhorias que poderiam ser realizadas como implementações futuras:

- Na tela de navegação livre, implementar um campo de busca por endereço para facilitar a utilização da funcionalidade.
- Implementar o compartilhamento de pontos entre usuários via *bluetooth*.
- Categorizar os pontos adicionados na navegação livre.
- Adicionar os pontos padrões no mapa da UTFPR Câmpus Pato Branco.

REFERÊNCIAS

ABNT NBR 9050. **Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos.** Disponível em:

<http://www.pessoacomdeficiencia.gov.br/app/sites/default/files/arquivos/%5Bfield_generico_imagens-filefield-description%5D_24.pdf>. Acesso em: 09 set. 2014.

ACESSOBRASIL. **O que é acessibilidade?** Disponível em: <http://www.acesso brasil.org.br/index.php?itemid=45>. Acesso em: 18 fev. 2014.

ANDROID ACESSIBILITY. **Enable TalkBack.** Disponível em: <<https://support.google.com/accessibility/android/answer/6007100?hl=en>>. Acesso em: 22 dez. 2014.

APPLE. **Using AssistiveTouch on your iOS device.** Disponível em: <<http://support.apple.com/en-la/HT202658>>. Acesso em: 22 dez. 2014.

BRASIL. **Lei n. 10.098, de 19 de dezembro de 2000 - Estabelece normas gerais e critérios básicos para promoção de acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida e dá outras providências.** Brasília; 2000.

BRASIL. **Coordenadoria nacional para integração da pessoa portadora de deficiência - acessibilidade.** Brasília: Secretaria Especial dos Direitos Humanos, 2008..

BRAILLE TOUCH. **Braille touch.** Disponível em: <<http://brailletouchapp.com/>>. Acesso em: 22 dez. 2014.

CONFORTO, Débora; SANTAROSA, Lucilia M. C. **Acessibilidade à web: Internet para todos. Revista Informática na Educação: Teoria & Prática.** v. 5 n. 2. Porto Alegre, 2002.

FAÇANHA, Agebson Rocha. **Uma proposta para acessibilidade visual e tátil em dispositivos touchscreen.** Dissertação (mestrado). Universidade Federal do Ceará. Centro de Ciências. Departamento de Computação. 2012. Disponível em: <http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CD EQFjAC&url=http%3A%2F%2Fwww.mdcc.ufc.br%2Fteses%2Fdoc_download%2F186-&ei=tj-YVJ35MZbGsQS8IIHYAg&usg=AFQjCNFp2sIDgFtQtqRqAw9pziunm6hP9Q&sig2=AaOTKebZ45_ZlpRqPkpkIA&bvm=bv.82001339,d.cWc&cad=rja>. Acesso em: 22 dez. 2014.

GODOY, Andréa et al. **Cartilha da inclusão dos direitos da pessoa com deficiência.** Belo Horizonte: PUC/MG; 2000.

IOS. **iOS: a wide range of features for a wide range of needs.** Disponível em: <<https://www.apple.com/accessibility/ios/>>. Acesso em: 22 dez. 2014.

PAGLIUCA, Lorita M. F.; ARAGÃO, Antônia E. D. A.; ALMEIDA, Paulo C. **Acessibilidade e deficiência física: identificação de barreiras arquitetônicas em áreas internas de hospitais de Sobral, Ceará.** Rev. esc. enferm. USP, v. 41 n. 4. São Paulo, Dec. 2007.

PISKE; Otavio R. SEIDEL, Fábio A. **Rapid application development.** 2006. Disponível em: <<http://www.angusyoung.org/arquivos/artigos/rad.pdf>>. Acesso em: 3 dez. 2014.

VOICEIVER. **VoiceOver for OS X: a feature tha speaks for itself.** Disponível em: <<https://www.apple.com/accessibility/osx/voiceover/>>. Acesso em: 3 dez. 2014.