

CENTRO UNIVERSITÁRIO ADVENTISTA DE SÃO PAULO
CAMPUS ENGENHEIRO COELHO
CURSO DE SISTEMAS PARA INTERNET

GABRIEL TAGLIARI RODRIGO
MATHEUS SILVA HOFART
ELISA BATTAGLINI DOMINGUES PEREIRA

INFOBEACONS

ENGENHEIRO COELHO

2016

GABRIEL TAGLIARI RODRIGO
MATHEUS SILVA HOFART
ELISA BATTAGLINI DOMINGUES PEREIRA

INFOBEACONS

Trabalho de Conclusão de Curso do Centro Universitário Adventista de São Paulo do curso de Sistemas para Internet, sob a orientação do Prof. Dr. Roberto Sussumu Wataya.

ENGENHEIRO COELHO

2016

Rodrigo, Gabriel Tagliari; Hofart, Matheus Silva; Pereira, Elisa Battaglini Domingues.

INFOBEACONS/ por Gabriel Tagliari Rodrigo, Matheus Silva Hofart, Elisa Battaglini Domingues Pereira. 2016.
25f.; 29,7cm.

Trabalho de Conclusão de Curso - UNASP-EC - Centro Universitário Adventista de São Paulo - Engenheiro Coelho, 2016.

"Orientação: Prof. Dr. Roberto Sussumu Wataya"

1. Beacons. 2. Informações. 3. Acessibilidade. I. Título

Trabalho de Conclusão de Curso do Centro Universitário Adventista de São Paulo, do curso de Sistemas para Internet apresentado e aprovado em 22 de Novembro de 2016.

Dr. Roberto Sussumu Wataya

Me. Percival Lucena

Me. Thales de Társis Cezare

Agradecimentos

- Ao Prof. Dr. Roberto Sussumu Wataya que nos auxiliou nas etapas do processo de desenvolvimento do projeto.
- Ao Prof. Me. Percival Lucena que esteve sempre disposto a nos ajudar.
- Aos amigos e familiares pelo constante apoio.

"As pessoas não sabem o que querem, até mostrarmos a elas"
(Steve Jobs)

Resumo

Conforme o tempo passa, notamos que as formas de se obter informações evoluem, a necessidade de se manter informado é grande. As pessoas buscam formas diferentes, rápidas e mais fáceis de conhecer e entender o que está a sua volta. O projeto tem por objetivo trazer às pessoas as informações necessárias do que lhes é exposto nos museus através de seus aparelhos *android* que lêem as informações cadastradas nos *beacons* de cada obra. Com o aplicativo aberto, ao aproximar o *smartphone* ou *tablet* serão exibidas as informações cadastradas para aquele determinado *beacon* mais próximo encontrado pelo *bluetooth* do dispositivo. Este aplicativo foi criado para atender essa necessidade de informação e principalmente de acessibilidade. Foi inspirado em projetos já existentes, porém diferentes, mas que trazem a mesma ideia de facilitar o acesso à informação. As tecnologias utilizadas para a criação do projeto foram *Java*, *Android*, *Volley*, *HTML5*, *CSS3*, *Spring*, *Hibernate*, *Bootstrap* e *MySQL*.

Palavras-chaves: *Beacons*; Informações; Acessibilidade.

Abstract

As time goes by, we note that the way to obtain information evolves, it's important to keep informed. What people seek are different, fastest and simple ways to know and understand what is around them. The project aims to bring people the necessary information that are displayed in museums through their Android devices that read the information registered for beacons of each work of art. With the app open, when approaching the smartphone or tablet, the information recorded for the nearest beacon find by bluetooth are displayed on the device screen. This app was created to attend the information needed and mostly the accessibility. It was inspired in existing projects, but different, it reflects the same idea to facilitate the information access. The technologies used for the creation of the project were Java, Android, Volley, HTML5, CSS3, Spring, Hibernate, Bootstrap and MySQL.

Keywords: Beacons; Information; Accessibility.

Sumário

1	Introdução	1
2	Metodologia	2
3	Fundamentação Teórica	3
3.1	Acessibilidade	3
3.2	Java	3
3.3	Spring	3
3.4	Hibernate	3
3.5	HTML5	4
3.6	CSS3	4
3.7	Bootstrap	4
3.8	Volley	5
3.9	MySQL	5
4	Desenvolvimento	6
4.1	Beacon	6
4.2	Bluetooth 4.0	6
4.3	Android	7
4.4	TalkBack	8
4.4.1	Gestos e Navegação	8
4.4.2	Como ativar o TalkBack	9
4.4.3	Como desativar o TalkBack	11
4.5	Desenvolvendo o aplicativo Infobeacons	11
4.5.1	Android Studio 2.0	16
4.5.2	Aparelho Android	17
4.6	Desenvolvendo o sistema de cadastramento de beacons	17
4.6.1	Ferramentas	17
4.6.2	Mapa mental do sistema de cadastramento de beacons	17
4.6.3	Telas	18
4.7	Banco de dados	21
4.7.1	Ferramentas	21
4.7.2	Tabelas	21
	Conclusões	23
	Referências	24

Lista de Figuras

Figura 1 – Sistemas Operacionais <i>Mobile</i> no Brasil	7
Figura 2 – Configurações	10
Figura 3 – Ativação <i>talkback</i>	10
Figura 4 – Desativar <i>TalkBack</i>	11
Figura 5 – Splash Screen	12
Figura 6 – Permissões	13
Figura 7 – Procurando	13
Figura 8 – Permissões	15
Figura 9 – Painel de ajuda	16
Figura 10 – Desenvolvimento <i>Android</i> Studio	16
Figura 11 – Mapa mental	18
Figura 12 – Tela de login	18
Figura 13 – Tela de listagem de <i>beacons</i>	19
Figura 14 – Tela de cadastrar <i>beacons</i>	19
Figura 15 – Tela de editar <i>beacons</i>	20
Figura 16 – Tela de criar contas	20
Figura 17 – Modelo Entidade Relacionamento	21

Lista de tabelas

Tabela 1 – Gestos básicos	9
Tabela 2 – Gestos vai-e-vem	9
Tabela 3 – Gestos angulados	9
Tabela 4 – Moto G	17
Tabela 5 – Samsung Galaxy J7	17

1 Introdução

O tema deste projeto é a acessibilidade em museus através de *beacons* e foi desenvolvido um aplicativo para *smartphones* e *tablets* que recebe informações dos *beacons* e exibe informações na tela. O tema veio de uma ideia em conjunto, inspirada em projetos já existentes e utilizados em museus.

Quando vamos a um museu, é comum não encontrarmos todas as informações referentes ao que é exposto. Os visitantes querem ver e saber sobre aquilo que já fez parte da história através das esculturas, buscam entender o sentimento que os pintores querem passar através de suas pinturas, então se faz necessário criar uma forma de atender a essa procura. Além dessa dificuldade de informações existem aqueles que possuem uma barreira maior, os deficientes visuais, que utilizam, neste caso, os sentidos do tato e da audição. Além do toque, ainda sim precisam de outras maneiras de obterem informações sobre as exposições.

Este estudo tem como objetivo criar um aplicativo baseado na tecnologia *beacon*, que aprimora o acesso às informações dos objetos expostos no museu, através dos dispositivos móveis por pessoas deficientes visuais ou não.

A importância deste trabalho se reflete em possibilitar aos visitantes do museu acesso as informações necessárias de uma forma rápida e dinâmica, melhorando a interação entre o museu e os usuários do aplicativo Infobeacons, trazendo consigo a inserção da acessibilidade, atributo essencial para locais que recebem visitantes com necessidades especiais ou que pretendem fazer essa inclusão.

Este aplicativo utiliza a tecnologia *beacons*, um dispositivo que transmite sinais *bluetooth*, esses sinais podem ser captados por aplicativos de *smartphones* e *tablets*. São colocados *beacons* em cada obra que é exposta no museu, para que ao se aproximar com um *smartphone* ou *tablet* que possua este aplicativo instalado, será apresentada na tela e convertida em voz a informação cadastrada para aquele determinado *beacon* no banco de dados. Consequentemente, ao se aproximar de outro *beacon* o processo se repetirá com as informações referentes a ele.

2 Metodologia

A metodologia utilizada neste estudo consistiu em quatro etapas, iniciando com a fundamentação teórica, de trabalhos relacionados e estado da arte. Na segunda etapa - fundamentação teórica, foram feitos conceitos sobre acessibilidade às informações pelas pessoas com deficiência; Na terceira etapa, estudos técnicos do *beacons*, *Android*, *bluetooth*, *talkback*. Na etapa seguinte a criação do aplicativo com a tecnologia *beacons*, a criação do sistema de cadastramento de *beacons*, e finalmente as considerações finais.

As ferramentas que foram usadas para todo o desenvolvimento foram: *Android Studio*, *Eclipse*, *Google Chrome*, *Opera*, *MySQL WorkBench*, *Brackets* e *Photoshop*, *Wamp-Server*.

Foi utilizado um computador *desktop* com processador *AMD FX 8350*, com sistema operacional *Windows 8.1*, 6Gb de memória ram e um notebook *HP G42* com processador *Intel Core i5*, sistema operacional *Ubuntu 15.10* e 2Gb de memória ram.

Após a realização do desenvolvimento iniciou-se a parte dos testes para garantir a funcionalidade tanto do aplicativo quanto do sistema de cadastramento.

3 Fundamentação Teórica

3.1 Acessibilidade

A acessibilidade envolve não só o meio físico mas também o meio digital, que possibilita a inclusão de portadores de deficiência em tudo que é oferecido para a sociedade. As informações são acessíveis a partir do momento em que são divulgadas de formas variadas como, por imagem, texto e som. (TORRES *et al.*, 2002)

3.2 *Java*

A linguagem *Java* surgiu a partir de uma pesquisa financiada pela *Sun Microsystems* em 1991, iniciada sob o codinome de projeto *Green*. Com o intuito de se tornar uma linguagem para dispositivos inteligentes destinados ao usuário final e com a premissa de ser uma linguagem próxima as linguagens *C* e *C++* e que pudesse ser executado em diversos hardwares. (SAKURAI, 2011)

3.3 *Spring*

A história do *Spring* começou com um desenvolvedor em busca de qualidade no desenvolvimento Java para web. Esta pessoa, hoje um grande nome da comunidade Java, é Rod Johnson. Suas contribuições começaram com o livro chamado “Expert One-on-One J2EE Design and Development”. Nele, Rod mostra como utilizar as tecnologias *J2EE* (atualmente *Java EE*) para reduzir a complexidade do desenvolvimento, auxiliando ainda a resolver problemas e erros comuns. (ALEXANDRE, 2016)

3.4 *Hibernate*

O *Hibernate* é um framework para o mapeamento objeto-relacional escrito na linguagem Java, mas também é disponível em .Net como o nome *NHibernate*. Este programa facilita o mapeamento dos atributos entre uma base tradicional de dados relacionais e o modelo objeto de uma aplicação, mediante o uso de arquivos (*XML*) para estabelecer esta relação. (CRUZ, 2010)

3.5 *HTML5*

A sigla *HTML* significa *HyperText Markup Language* em português, linguagem de marcação de hipertexto. A primeira versão do *HTML* foi baseada na linguagem *SGML*. O *SGML* era utilizado para a estruturação de documentos e foi dele que o *HTML* herdou diversas *tags*, tais como: título `<h1>` ao `<h6>`, cabeçalho `<head>` e parágrafo `<p>`. A maior diferença entre essas duas linguagens de marcação é que o *HTML* implementava a *tag* `<a>` com o atributo *href*, permitindo assim a ligação (links) de uma página a outra. Esse conceito de interligação entre documentos é a base do funcionamento de toda Web. (WILLIAM, 2016)

Para a criação do sistema de cadastramento foi usado o *HTML5*, uma recente evolução do *HTML* e uma nova versão da linguagem, com novos elementos, atributos e comportamentos.

3.6 *CSS3*

O *Cascading Style Sheets (CSS)* foi proposto pela primeira vez em outubro de 1994, por Hakon Lie, que queria facilitar a programação de sites, que na época era muito mais complexa. As pessoas tinham que utilizar mais códigos para chegar a um resultado simples, como criar uma tabela. Em 1995 o *CSS1* foi desenvolvido pela W3C, um grupo de empresas do ramo da informática. (PACIEVITCH, 2016)

Segundo Pacievitch, a linguagem de estilos ganhou muito destaque entre 1997 e 1999, neste período ficou conhecido por grande parte dos programadores.

3.7 *Bootstrap*

Em 2011, o *Bootstrap* foi criado como uma solução interna para resolver inconsistências de desenvolvimento dentro da equipe de engenharia do *Twitter*. Embora inicialmente uma solução interna no *Twitter*, Mark e Jacob rapidamente perceberam que tinham descoberto algum muito maior. Em agosto de 2011, a estrutura *Bootstrap* foi lançada como um projeto de software livre no *Github*. Em alguns meses, milhares de desenvolvedores de todo o mundo contribuíram com o código e o *Bootstrap* se tornou o projeto de desenvolvimento de software livre mais ativo do mundo. (UTTERBACK, 2014)

3.8 Volley

O *Volley* é uma biblioteca mantida pelo Google com a proposta de tornar a implementação de comunicação *HTTP* em aplicações *Android* mais fácil e otimizada. Essa biblioteca também permite reduzir a quantidade de *Boilerplate Code* em relação a implementações baseadas nas classes *HTTP* nativas do *Android*, tornando o código mais simples e legível. (NIPO, 2014)

3.9 MySQL

O *MySQL* é um servidor robusto de bancos de dados *SQL* (*Structured Query Language*) - Linguagem Estruturada para Pesquisas) muito rápido, multitarefa e multiusuário. O Servidor *MySQL* pode ser utilizado em sistemas de produção com alta carga e missão crítica como também pode ser embutido em programa de uso em massa. (EDUCACAO, 2015)

4 Desenvolvimento

Neste capítulo será abordado o processo de desenvolvimento do aplicativo *Android*, do sistema de cadastramento, do *web service* e o banco de dados utilizado.

4.1 *Beacon*

Foram utilizados três *beacons* da marca *Estimote* para realização dos testes.

Estes *Beacons* são pequenos dispositivos com a configuração de *32-bit ARM® Cortex CPU* acompanham acelerômetro, sensor de temperatura e, o mais importante, *bluetooth 4.0 Smart*, também conhecido como *bluetooth de baixa energia*. Eles possuem um alcance máximo aproximado de 70 metros, porém, devido a interferências, em um contexto real estima-se entre 40-50 metros.(ESTIMOTE, 2014)

Eles possuem informações que são úteis para programar e identificar cada *beacon*.

UUID: É uma *string* de 16 byte que é utilizada para diferenciar um grupo grande de *beacons*. No caso do Infobeacons foi aplicado o mesmo *UUID* para todos os *beacons*.

Major: É uma *string* de 2 bytes que é usada para distinguir um grupo menor de *beacons*.

Minor: É uma *string* de 2 bytes que é usada para identificar um único *beacon* dentro do grupo.

MAC: É uma *string* de 6 bytes utilizada como endereço físico para comunicação com a interface de rede.

TxPower: É empregado como mensurador da distância do dispositivo com relação ao *beacon*. Ele é calibrado com a distância equivalente a um metro.

Ao permacener com emissão constante de sinal *bluetooth* é possível identificar as informações acima para que sejam utilizadas no desenvolvimento.

4.2 *Bluetooth 4.0*

O *bluetooth* é uma tecnologia que faz a conexão de dispositivos eletrônicos entre si, através de ondas de rádio, sem o uso de cabos ou fios, necessitando apenas da

aproximação dos dispositivos. Essa tecnologia trás consigo a liberdade do contato com aparelhos residenciais como, ouvir música pelo chuveiro, controlar sua TV e até mesmo ascender as luzes da casa através de seu *smartphone* ou *tablet* (SIG, 2016).

A ideia de criar o *bluetooth* veio da companhia Ericsson que ao longo do tempo atraiu outras companhias como a Intel, IBM, Toshiba e Nokia entre outras que com o passar dos anos também aderiram a esse consórcio. A diversidade de companhias colaborou com o desenvolvimento dessa tecnologia para reconhecimento nos vários tipos de dispositivos (ALECRIM, 2013).

Como a evolução da tecnologia é constante, a versão 4.0 do *bluetooth* traz consigo seu benefício, economia de energia, caso o aparelho esteja com o *bluetooth* ativo, mas sem ser utilizado por um tempo, ele entra em economia de energia até que volte a ser usado.

4.3 Android

Foi escolhido o *Android* como linguagem principal, pois a maioria dos aparelhos em nosso meio possuem o *Android* como sistema operacional e pelo fato de ter maior acessibilidade para ser desenvolvido.

De acordo com a *Kantar World Panel*, a venda de *Smartphones Android* no Brasil em Janeiro de 2016 possuía a porcentagem de 92,4% com relação aos outros Sistemas operacionais. (KANTAR, 2016)

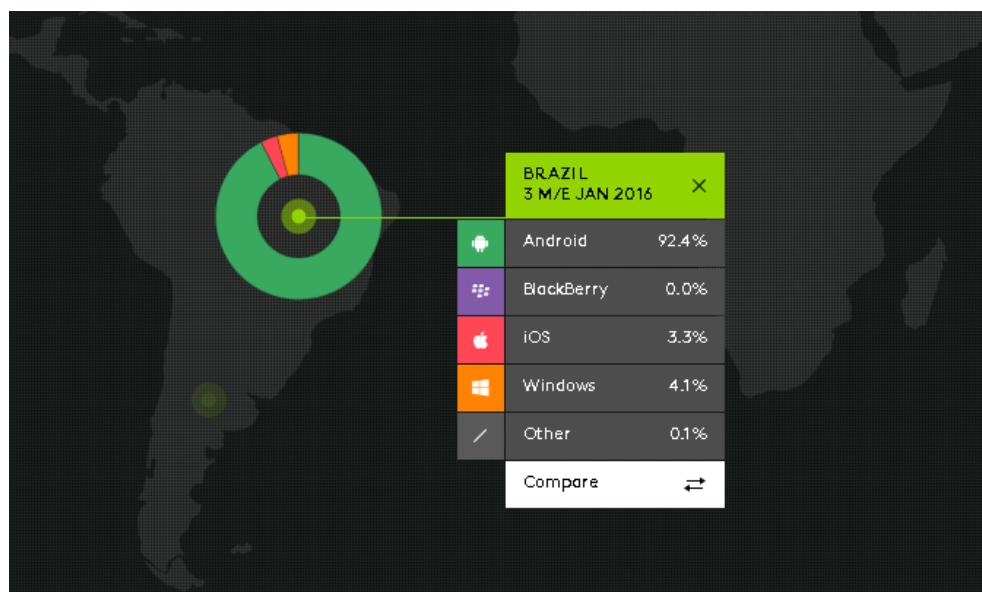


Figura 1 – Sistemas Operacionais *Mobile* no Brasil
Fonte: Autores

4.4 *TalkBack*

TalkBack não é uma coisa que você vai querer utilizar ao menos que você precise dele.

Pessoas que têm dificuldade em ver a sobrecarga de informação que um *smartphone* moderno tem para oferecer pois precisam de alguma ajuda. A *Google* fornece um conjunto muito abrangente de ferramentas de respostas. *Google TalkBack* é um serviço de acessibilidade que auxilia deficientes visuais e usuários com baixa visão a interagir e usufruir de seus dispositivos. Ele converte os textos em voz, emite vibrações e é possível saber o que pode ser feito conforme o toque na tela. É um serviço integrado com o *Android*, pois já vem instalado com o sistema operacional. (HILDENBRAND, 2014)

Ele funciona da seguinte maneira: o usuário utiliza os dedos para explorar a tela do dispositivo até o momento que atinja algum elemento ou texto. Logo, ao identificar este elemento, o *TalkBack* se encarregará de converter em voz sua informação. Esta informação pode ser uma mensagem, um texto ou até a etiqueta de conteúdo do elemento.

Segundo a Google (GOOGLE, 2016) "Os utilizadores de serviços de acessibilidade, como leitores de ecrã, recorrem a etiquetas de conteúdo para compreender o significado dos elementos numa interface.

Em alguns casos, como quando as informações são transmitidas graficamente num elemento, as etiquetas de conteúdo podem fornecer uma descrição de texto do significado ou da ação associada ao elemento.

Se os elementos numa interface do utilizador não fornecerem etiquetas de conteúdo, pode ser difícil para alguns utilizadores compreender as informações apresentadas ou efetuar ações na interface".

4.4.1 Gestos e Navegação

A priori, os gestos e a navegação do *TalkBack* pode ser complicado para quem possui boa visão, pois não é um recurso essencial para a navegação no Sistema Operacional do dispositivo. Contudo, existem usuários que necessitam deste recurso de acessibilidade para que seja possível navegar e utilizá-lo de forma plena no dia a dia.

Desta forma, eles precisam aprender alguns gestos para usufruir de todas as funcionalidades proposta pelo Sistema Operacional.

Segue alguns exemplos de gestos disponibilizados pela Google.

Ação	Gesto
Mover para o próximo item na tela	Deslizar para a direita
Mover para o item anterior na tela	Deslizar para a esquerda
Percorrer as configurações de navegação	Deslizar para cima ou para baixo
Selecionar item em foco	Tocar duas vezes

Tabela 1 – Gestos básicos

Ação	Gesto
Mover para o primeiro item na tela	Para cima e para baixo
Mover para último item na tela	Para baixo e depois para cima
Rolar para a frente (se estiver em uma página maior que a tela)	Para a direita e depois para a esquerda
Rolar para trás (se estiver em uma página maior que a tela)	Para a esquerda e depois para a direita
Mover controle deslizante para cima (por exemplo, o volume)	Para a direita e depois para a esquerda
Mover controle deslizante para baixo (por exemplo, o volume)	Para a esquerda e depois para a direita

Tabela 2 – Gestos vai-e-vem

Ação	Gesto
Botão "Início"	Para cima e para a esquerda
Botão "Voltar"	Para baixo e para a esquerda
Apps recentes	Para a esquerda e para cima
Notificações	Para a direita e para baixo (veja a observação abaixo)
Abrir menu de contexto local	Para cima e para a direita
Abrir menu de contexto global	Para baixo e para a direita

Tabela 3 – Gestos angulados

4.4.2 Como ativar o *TalkBack*

Para que seja ativado o *TalkBack* no *Android* o usuário deve seguir os seguintes passos:

Passo 1: Entrar nas "Configurações" do dispositivo, selecionar a opção "Acessibilidade" e depois clicar em "*TalkBack*".

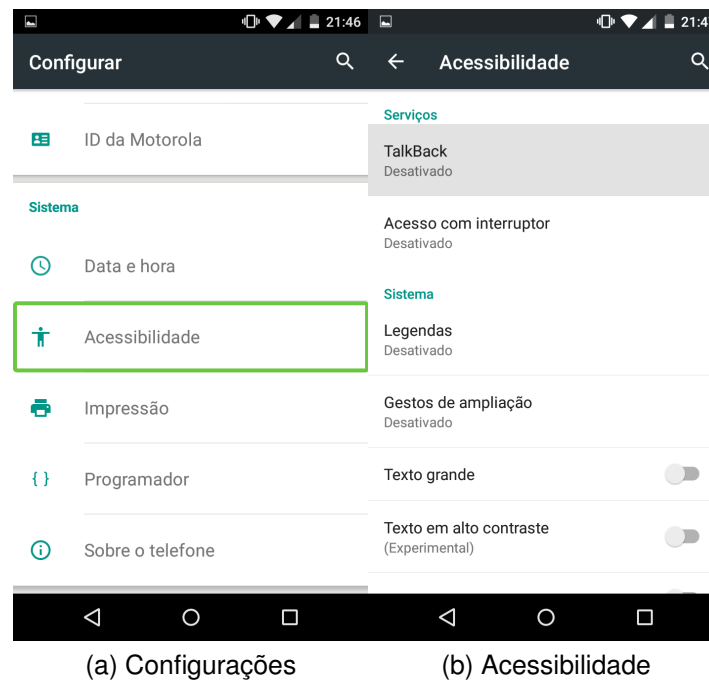


Figura 2 – Configurações
Fonte: Autores

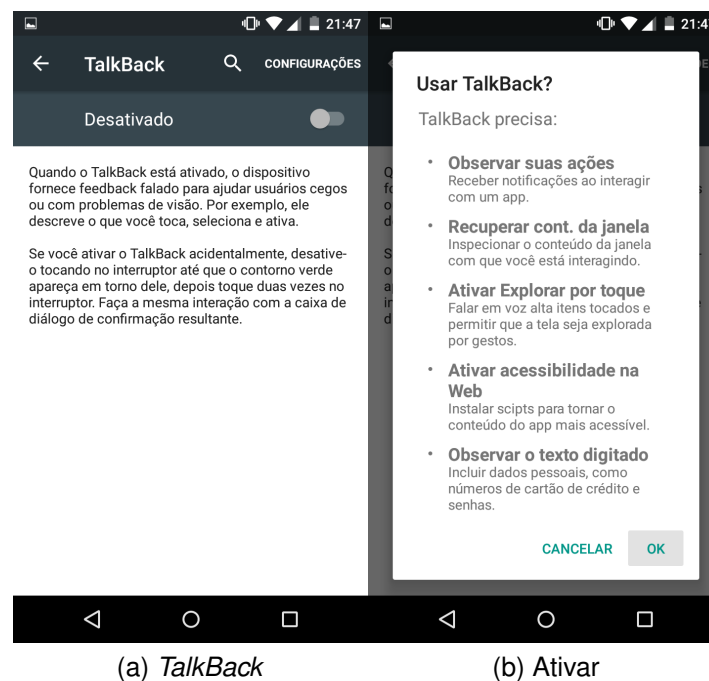


Figura 3 – Ativação *talkback*
Fonte: Autores

Passo 2: Clicar no *Switch* para ativar. Será aberta um alerta com as permissões

necessárias para o funcionamento do *TalkBack*.

Passo 3: Clicar em "OK".

4.4.3 Como desativar o *TalkBack*

Para desativar o *TalkBack* é necessário acessar novamente a opção *TalkBack* do menu de acessibilidade e clicar no elemento presente na parte superior direita da tela (mesma opção utilizada para a ativação). Logo, será exibida uma mensagem de confirmação para parar o *TalkBack* e clicando em "Ok" ele será desativado.

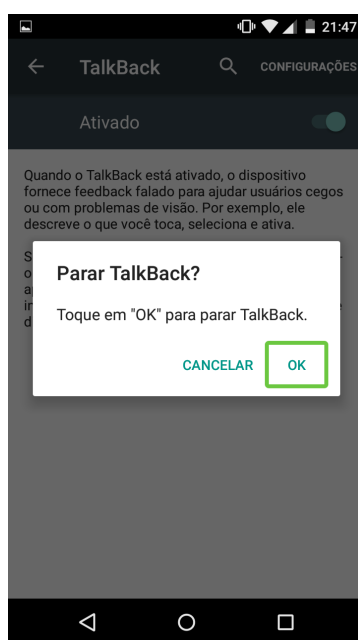


Figura 4 – Desativar *TalkBack*
Fonte: Autores

4.5 Desenvolvendo o aplicativo Infobeacons

O aplicativo foi desenvolvido em *Android*, utilizando o *Android Studio 2.0* como ferramenta. Ao abrir o aplicativo é apresentada a *splash screen* com o logo do *Infobeacons* e só dará lugar a tela seguinte quando o processamento dela estiver completo.

Splash Screen é uma tela inicial que, geralmente, possui o logo do aplicativo ou uma imagem e pode conter também uma barra de progresso. A função dela é referente a usabilidade do usuário quanto ao carregamento do aplicativo, pois permanece ativa enquanto é carregada a próxima tela e é destruída ao final deste carregamento. Desta forma, o usuário fica ciente que as informações estão sendo carregadas.



Figura 5 – Splash Screen
Fonte: Autores

Antes de começar a procurar os *beacons* se faz necessária a ativação do *bluetooth* e uma conexão ativa com a internet, seja por *Wi-Fi* ou *3G/4G*.

Bluetooth: É por meio do *bluetooth* que o dispositivo *Android* realiza o reconhecimento dos *beacons* e recebe as informações dele.

Internet: Possui o papel de realizar requisições *HTTP* no *web service* e recuperar informações lá cadastradas para o *beacon* cujo endereço físico foi passado como parâmetro da requisição.

Mediante ao *bluetooth* estar desativado ao iniciar o aplicativo, será exibida uma *pop-up* solicitando a permissão para que o *bluetooth* seja ativado pelo próprio aplicativo. Ao clicar na opção "permitir" será iniciada a busca pelos *beacons* normalmente, no entanto, caso seja escolhida a opção "recusar" o aplicativo exibirá um alerta em forma de *toast* informando que deverá ser ativado o *bluetooth* para que funcione corretamente.

Toast: Alerta nativo do *Android* que é exibido na parte inferior da tela e possui certa duração entre aparecer e desaparecer. A duração pode ser curta ou longa e ele é captado pelo *TalkBack* assim que é exibido.

Se o dispositivo não possuir conexão ativa com a internet enquanto o aplicativo está aberto, a qualquer momento pode ser exibido um alerta em forma de *toast* com a informação de que é necessária conexão com internet para o funcionamento correto.

Ambos são exemplificados pelas figuras abaixo.

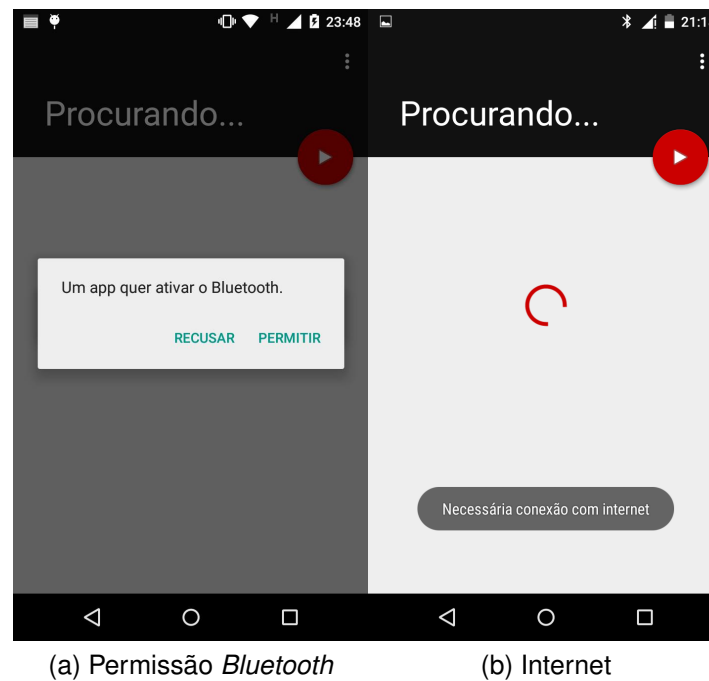


Figura 6 – Permissões
Fonte: Autores

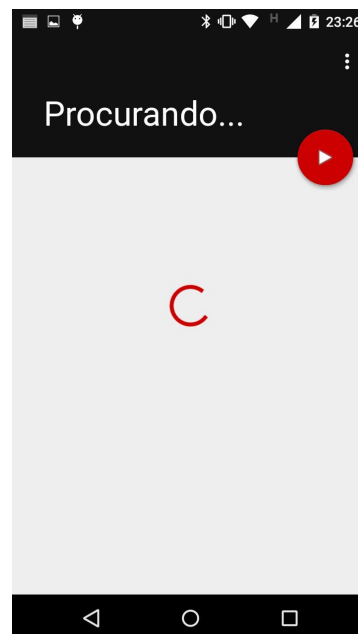


Figura 7 – Procurando
Fonte: Autores

Neste momento, com conexão a internet e com *bluetooth* ativado, o aplicativo começa a fazer a verificação dos *beacons* ao seu redor. Será permanecida a busca até o

momento que seja detectado algum *beacon* próximo, independente se o usuário estiver parado ou estar em movimento.

Existe um elemento gráfico, nativo do *Android*, localizado no centro da tela que fica em constante rotação indicando que o aplicativo está buscando os *beacons*.

Ao realizar o reconhecimento do *beacon* mais próximo são recebidas todas as informações dele tais como *UUID*, *Major*, *Minor*, *MAC*, dentre outras.

O dado mais importante e essencial neste cenário é o endereço *MAC*. Ele será adicionado como parâmetro em uma *URL*, será feita uma requisição *HTTP* por meio desta *URL*, chegará a um método no *web service* que buscará as informações cadastradas no banco de dados para aquele determinado *beacon* e será retornado um *JSON* para o aplicativo.

Ao realizar a requisição com sucesso e obter este retorno do objeto *JSON* o aplicativo organizará estas informações na tela do dispositivo da seguinte forma:

Barra superior: Alterada para o nome do *beacon* recebido pela requisição.

Abaixo da barra superior: Painel de imagem que exibirá a imagem recebida pela requisição.

Abaixo da imagem: Exibição do texto referente ao conteúdo recebido pela requisição.

É apresentada uma mensagem em forma de *toast* informando que os dados foram encontrados, pois ela será lida pelo *TalkBack* do dispositivo caso ele esteja ativo.

No entanto, se ocorrer alguma falha na requisição *HTTP* será exibida na tela a mensagem "[Erro] Tentando novamente..." e o aplicativo fará outra tentativa.

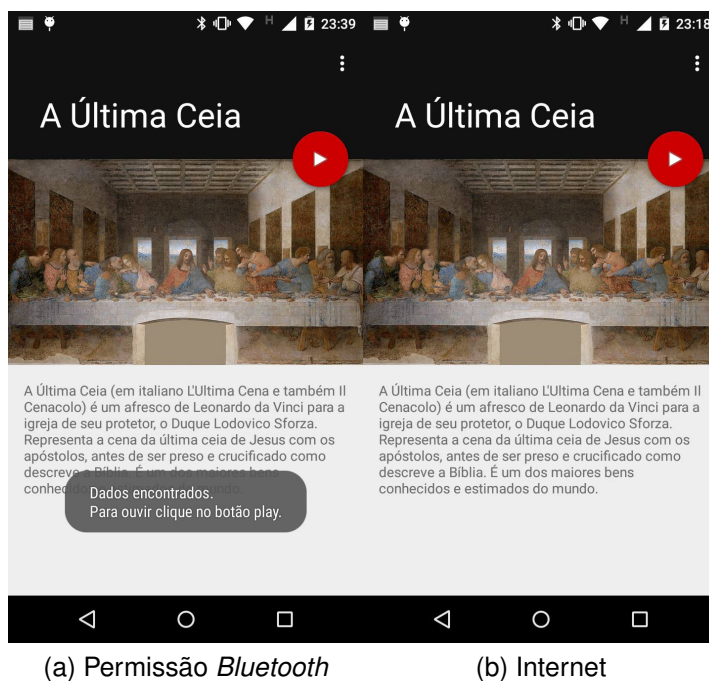
(a) Permissão *Bluetooth*(b) *Internet*

Figura 8 – Permissões

Fonte: Autores

A partir do momento que a informação está carregada na tela, o usuário consegue visualizar as informações ali descritas. Se o texto for muito extenso para caber na tela do dispositivo, é possível utilizar o gesto de *scroll* para ir rolando o texto conforme a leitura.

Foi criado um botão de play localizado na parte superior direita para que usuários em geral possam ouvir o texto referente ao *beacon*.

Este botão possui etiqueta de identificação para os usuários de *TalkBack* e, quando acionado, converte o texto exibido na tela em voz através do *Text to Speech*.

Text to Speech: É uma biblioteca nativa do *Android* responsável por converter o texto em voz.

Para auxiliar os usuários foi criado um painel de ajuda acessado pela opção localizada no canto superior direito. Este painel contém as informações sobre os requisitos para que o aplicativo funcione corretamente e uma breve explicação.

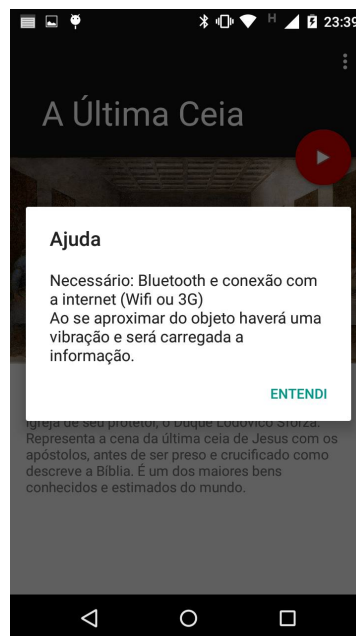


Figura 9 – Painel de ajuda
Fonte: Autores

4.5.1 Android Studio 2.0

Para o desenvolvimento na linguagem *Android* foi utilizado o *Android Studio 2.0*. O *Android Studio* é o ambiente de desenvolvimento integrado (*IDE*) oficial para o desenvolvimento de aplicativos *Android* e é baseado no *IntelliJ IDEA*. (STUDIO, 2016)

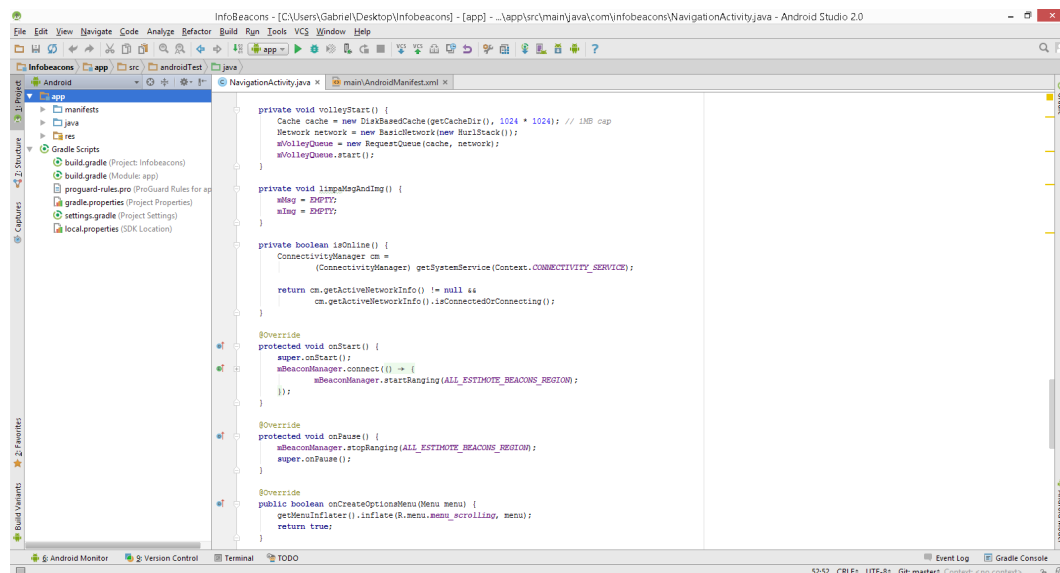


Figura 10 – Desenvolvimento *Android Studio*
Fonte: Autores

4.5.2 Aparelho *Android*

Os *Smartphones Android* utilizados para a realização dos testes do aplicativo foram:

Especificações técnicas	
Marca	Motorola
Modelo	Moto G
Sistema Operacional	Android 5.1 Lollipop
Tamanho da tela	4,5" polegadas

Tabela 4 – Moto G

Especificações técnicas	
Marca	Samsung
Modelo	Galaxy J7
Sistema Operacional	Android 6.0 Marshmallow
Tamanho da tela	5,5" polegadas

Tabela 5 – Samsung Galaxy J7

4.6 Desenvolvendo o sistema de cadastramento de *beacons*

4.6.1 Ferramentas

O Eclipse versão Luna foi a principal *IDE* utilizada para o desenvolvimento do sistema em *Java*.

4.6.2 Mapa mental do sistema de cadastramento de *beacons*

O mapa mental representa as rotas do sistema de cadastramento de *beacons*, com todos os fluxos que podem ser seguidos nele.

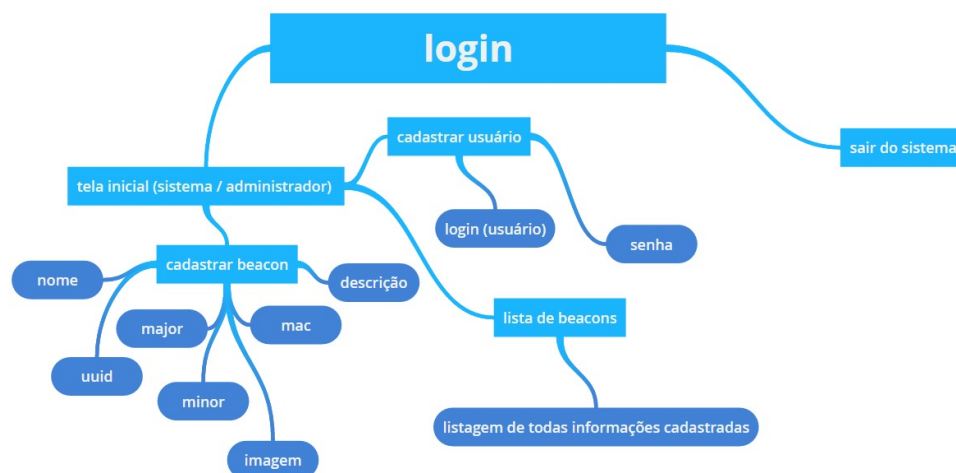


Figura 11 – Mapa mental
Fonte: Autores

4.6.3 Telas

A primeira tela do nosso sistema de cadastramento é um login. Por questões de segurança, o administrador principal do projeto deverá ter a conta criada pelo próprio desenvolvedor, com esse login em mãos ele poderá cadastrar novos usuários.

Caso o usuário preencha os dados pedido pelo formulário erroneamente, os campos digitados serão apagados automaticamente e em seguida o mesmo deve preenchê-lo novamente.

A imagem mostra a interface de login do sistema 'infobeacons'. No topo, há um ícone de um dispositivo eletrônico e o nome 'infobeacons'. Abaixo, há três campos de entrada: 'Usuário' (com ícone de pessoa), 'Senha' (com ícone de cadeado) e um botão 'Acessar'.

Figura 12 – Tela de login
Fonte: Autores

Quando as informações forem devidamente cadastradas em seus respectivos be-

acons, esta tela fará a listagem de todos esses objetos. Imagem, nome e *MAC* serão referências para que o usuário identifique qual conteúdo ele cadastrou em determinado *beacon*.

O usuário ainda poderá executar algumas ações, como por exemplo excluir o *beacon* ou apenas editá-lo.

 infobeacons

Bem Vindo, matheus

Lista de Beacons

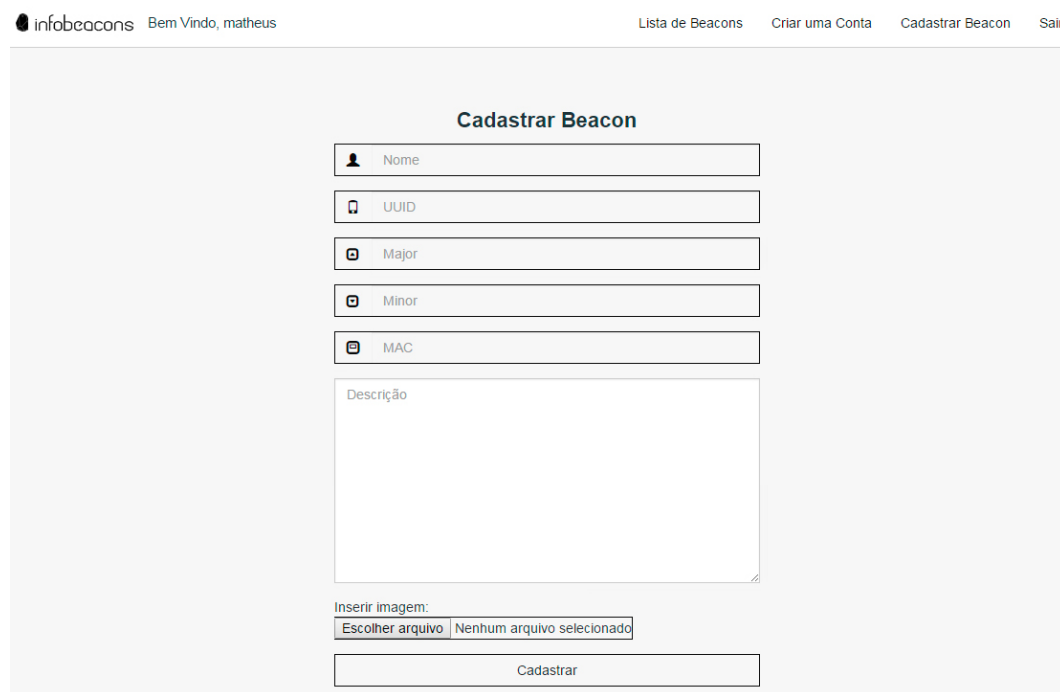
Criar uma Conta

Cadastrar Beacon

Sair

Imagem	Nome	Mac	Ações	
	O Grito	E9:F7:CF:EB:35:0D	Editar	Excluir
	A Última Ceia	E6:6A:91:0A:D1:48	Editar	Excluir
	Mona Lisa	F7:CC:43:EA:C4:27	Editar	Excluir
	A Ronda Noturna	DA:CD:77:58:59:52	Editar	Excluir

Figura 13 – Tela de listagem de *beacons*
Fonte: Autores



infobeacons Bem Vindo, matheus

Lista de Beacons Criar uma Conta Cadastrar Beacon Sair

Cadastrar Beacon

Descrição

Inserir imagem:

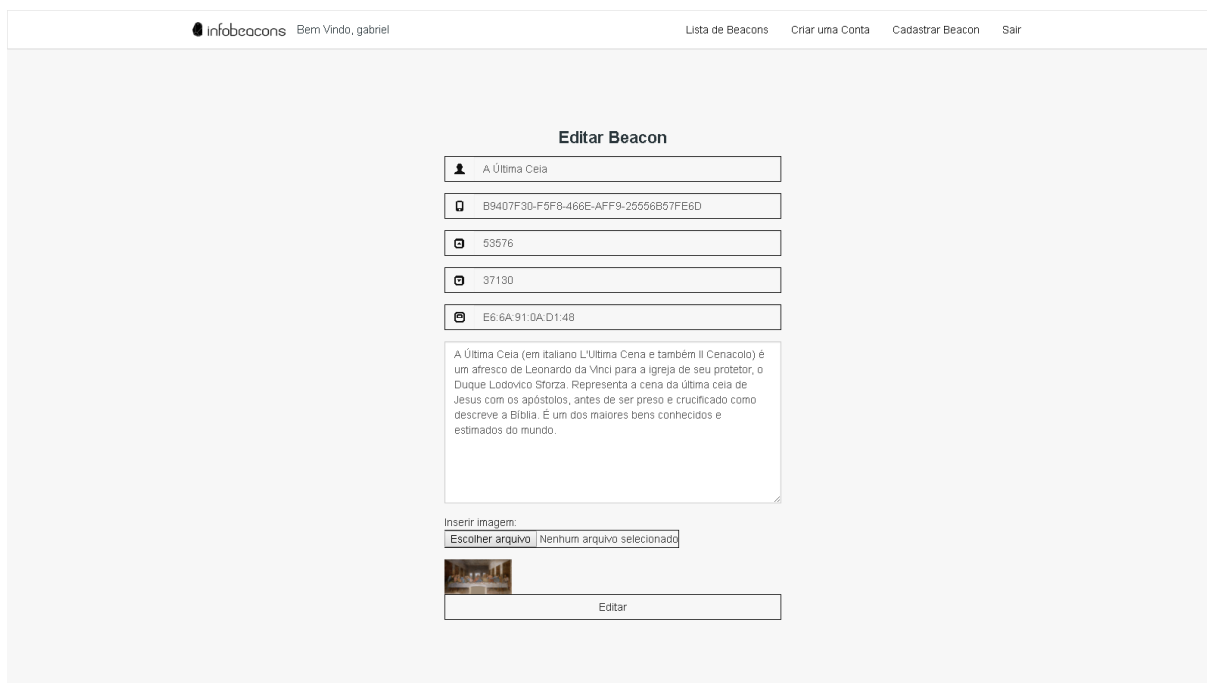
Nenhum arquivo selecionado

Figura 14 – Tela de cadastrar *beacons*
Fonte: Autores

Os *beacons* serão cadastrados de acordo com o que é pedido pelo formulário. Nome, *UUID*, *major*, *minor* e *MAC* são necessários para que o usuário tenha êxito neste processo.

A imagem é o único campo que não é requerida, sendo assim, você pode terminar de cadastrar o *beacon* normalmente, caso queira adicionar a imagem depois, basta entrar na tela de edição e subir uma imagem com até 300Kb.

A tela de edição é exatamente igual a tela de cadastro, todas as informações cadastradas estará presente nela. Caso o usuário tenha escrito alguma palavra errada, adicionado a imagem errada ou até mesmo ter colocado alguma outra informação errada referente ao *beacon*, ele poderá fazer as devidas correções.



The screenshot shows the 'Editar Beacon' form in the infobeacons application. The header includes the logo 'infobeacons', the user 'Bem Vindo, gabriel', and navigation links: 'Lista de Beacons', 'Criar uma Conta', 'Cadastrar Beacon', and 'Sair'. The form itself is titled 'Editar Beacon' and contains several input fields: a text field for the name 'A Última Ceia', a text field for the ID 'B9407F30-F5F8-466E-AFF9-2556B57FE6D', a text field for the category '53576', a text field for the location '37130', and a text field for the email 'E6:6A:91:0A:D1:48'. Below these fields is a text area containing a description of 'A Última Ceia' (The Last Supper) by Leonardo da Vinci. At the bottom, there is an 'Inserir imagem:' section with a button 'Escolher arquivo' and a note 'Nenhum arquivo selecionado'. A small thumbnail image of the Last Supper is visible, and an 'Editar' button is at the bottom right of the form.

Figura 15 – Tela de editar *beacons*
Fonte: Autores



The screenshot shows the 'Criar Conta' form in the infobeacons application. The header is identical to the previous figure, showing the logo 'infobeacons', the user 'Bem Vindo, matheus', and navigation links: 'Lista de Beacons', 'Criar uma Conta', 'Cadastrar Beacon', and 'Sair'. The form is titled 'Criar Conta' and contains four input fields: a text field for the username 'Usuário', a text field for the password 'Senha', a text field for the password confirmation 'Confirmar senha', and a button 'Criar' at the bottom.

Figura 16 – Tela de criar contas
Fonte: Autores

O administrador do projeto poderá cadastrar um novo usuário no sistema, basta ele entrar nesta tela e preencher as informações que serão solicitadas pelo formulário, todos os campos são obrigatórios para que ele tenha êxito ao finalizar o processo de criação. Todos os usuários que forem criados poderão adicionar novos *beacons* e usuários.

4.7 Banco de dados

4.7.1 Ferramentas

Como ferramenta de administração do banco de dados para este projeto foi usado o *MySQL Workbench*. Ela facilita o processo de criação, manipulação e geração do modelo entidade relacionamento do banco de dados.

4.7.2 Tabelas

Foi utilizado o *MySQL* como banco de dados para que sejam armazenados as informações cadastradas e recuperadas pelo sistema de cadastramento.

Foram criadas quatro tabelas: *user*, *role*, *user_role* e *beacon*. Tais demonstradas pelo modelo entidade relacionamento abaixo:

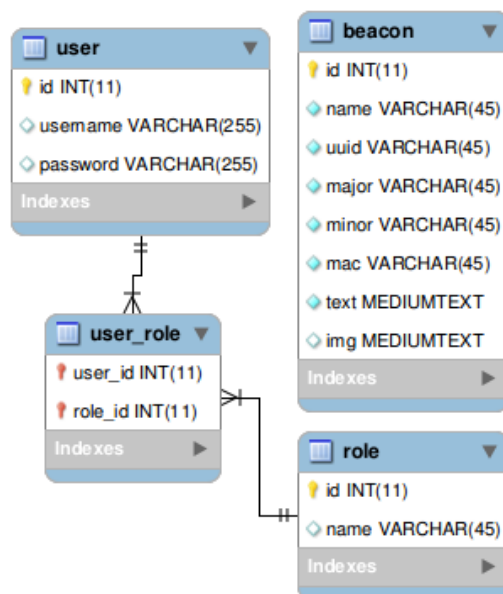


Figura 17 – Modelo Entidade Relacionamento
Fonte: Autores

A tabela *beacon* armazena os dados dos *beacons* que são cadastrados pelo sistema de cadastramento e possui as seguintes colunas:

Id: É o identificador único de cada registro na tabela e é constituído por um inteiro que é incrementado automaticamente a cada novo registro.

Nome: Identificação de nome para o *beacon*.

Uuid: Identificador alfa-numérico.

Major: Número entre 0 e 65523 para identificar com maior precisão um pequeno grupo de *beacons*.

Minor: Número entre 0 e 65523 para identificar com maior precisão um único *beacon*.

Mac: Endereço físico único de cada *beacon*.

Texto: Texto que será exibido no aplicativo.

Img: Imagem convertida em Base64 que será exibida no aplicativo.

A tabela *user* foi criada para que sejam armazenados os dados de usuários, tais como:

Id: É o identificador único de cada registro na tabela e é constituído por um inteiro que é incrementado automaticamente a cada novo registro.

Username: É o nome que o usuário utilizará para entrar no sistema de cadastramento.

Password: É a senha, criptografada nesta coluna, que será utilizada para que o usuário entre no sistema de cadastramento.

A tabela *role* armazena as informações referentes a permissão do usuário no sistema, sendo que podem ser cadastrados diferentes perfis para determinadas rotas do sistema. O padrão utilizado é o de administrador. Ela possui os seguintes campos:

Id: É o identificador único de cada registro na tabela e é constituído por um inteiro que é incrementado automaticamente a cada novo registro.

Name: É um nome para identificar o tipo de permissão.

A tabela *user_role* é o relacionamento muitos para muitos entre as tabelas *user* e *role*. Ela é responsável por ligar um usuário a uma permissão e possui os seguintes campos:

User_id: Identificador único da tabela *user*.

Role_id: Identificador único da tabela *role*.

Considerações finais

Neste projeto abordamos sobre como resolver o problema de obter informações dentro de um museu pelos visitantes em geral, porém com atenção para aqueles que possuem deficiência visual.

Um dos objetivos iniciais do projeto era não utilizar internet para que os usuários pudessem utilizar o aplicativo em qualquer local. Contudo, foi decidido incorporar um sistema de cadastramento e um *web service* para que a manutenção e o cadastramento de *beacons* do projeto fosse realizada facilmente. Outro ponto que levou a esta tomada de decisão foi o fácil acesso atual a planos de 3g, logo, não haveria problema em utilizar dados móveis.

Consideramos que tudo que foi apresentado neste trabalho faz com que este problema encontrado seja sanado.

Este projeto foi de grande valor, visto que nos permitiu desenvolver algo que colaborasse tanto para resolver a dificuldade de informação quanto no quesito acessibilidade que leva em consideração a inclusão de portadores de deficiência visual em locais despreparados.

Perspectivas Futuras

- Hospedagem do *web service*
- Publicar o aplicativo na *Google Play*

Referências

ALECRIM, E. *Tecnologia Bluetooth: o que é e como funciona?* 2013. Disponível em: <<http://www.infowester.com/bluetooth.php>>. Acesso em: 20 de Out. 2016.

ALEXANDRE, J. *Todo poderoso Spring - Revista Java Magazine 100 - Parte 2*. 2016. Disponível em: <<http://www.devmedia.com.br/todo-poderoso-spring-revista-java-magazine-100-parte-2/23539>>. Acesso em: 21 de Out. 2016.

CRUZ, E. A. *Hibernate conceitos*. 2010. Disponível em: <<http://pt.slideshare.net/elifranio/hibernate-conceitos>>. Acesso em: 21 de Out. 2016.

EDUCACAO, P. *História do MySQL*. 2015. Disponível em: <<https://www.portaleducacao.com.br/informatica/artigos/66679/historia-do-mysql>>. Acesso em: 21 de Out. 2016.

ESTIMOTE. *Developer Docs*. 2014. Disponível em: <<http://developer.estimote.com/>>. Acesso em: 25 de Abr. 2016.

GOOGLE. *Etiquetas de conteúdo*. 2016. Disponível em: <<https://support.google.com/accessibility/android/answer/7158690?hl=pt>>. Acesso em: 30 de Abr. 2016.

HILDENBRAND, J. *What is Google TalkBack?* 2014. Disponível em: <<http://www.androidcentral.com/what-google-talk-back>>. Acesso em: 30 de Abr. 2016.

KANTAR. *Smartphone OS sales market share*. 2016. Disponível em: <<http://www.kantarworldpanel.com/global/smartphone-os-market-share/>>. Acesso em: 5 de Out. 2016.

NIPO, F. *HTTP em Android com Volley*. 2014. Disponível em: <<https://felipenipo.wordpress.com/2014/10/17/http-em-android-com-volley>>. Acesso em: 21 de Out. 2016.

PACIEVITCH, Y. *Cascading Style Sheets (CSS)*. 2016. Disponível em: <<http://www.infoescola.com/informatica/cascading-style-sheets-css>>. Acesso em: 21 de Out. 2016.

SAKURAI, R. G. *Java - Origem e evolução da linguagem Java*. 2011. Disponível em: <<http://www.universidadejava.com.br/materiais/origem-evolucao-java>>. Acesso em: 21 de Out. 2016.

SIG, B. *What is Bluetooth technology?* 2016. Disponível em: <<https://www.bluetooth.com/what-is-bluetooth-technology/bluetooth-technology-basics/low-energy>>. Acesso em: 20 de Out. 2016.

STUDIO, A. *Conheça o Android Studio*. 2016. Disponível em: <<https://developer.android.com/studio/intro/index.html>>. Acesso em: 10 de Out. 2016.

TORRES, E. F.; MAZZONI, A. A.; ALVES, J. d. M. A acessibilidade à informação no espaço digital. *Ciência da Informação, Brasília*, SciELO Brasil, v. 31, n. 3, p. 83–91, 2002.

UTTERBACK, B. *O que é o Bootstrap? – Verdades e mitos: Parte 1 de 2*. 2014. Disponível em: <<https://www.prestashop.com/blog/pt/2014/03/06/o-que-e-o-bootstrap-verdades-e-mitos-parte-1-de-2>>. Acesso em: 21 de Out. 2016.

WILLIAM, D. *A História Do HTML*. 2016. Disponível em: <<http://www.frontendbrasil.com.br/artigos/a-historia-do-html>>. Acesso em: 21 de Out. 2016.