

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA – CT
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**UM APLICATIVO MÓVEL PARA REGISTRO
AUTOMÁTICO DA PRESENÇA ACADÊMICA VIA
BLUETOOTH**

TRABALHO DE GRADUAÇÃO

Fernando Weber Albiero

**Santa Maria, RS, Brasil
2014**

UM APLICATIVO MÓVEL PARA REGISTRO AUTOMÁTICO DA PRESENÇA ACADÊMICA VIA BLUETOOTH

Por

Fernando Weber Albiero

Trabalho de Graduação apresentado ao Curso de Ciência da Computação da
Universidade Federal de Santa Maria (UFSM, RS),
como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação.

Orientadora: Iara Augustin

Coorientador: João Carlos Damasceno de Lima

**Trabalho de Graduação N° 381
Santa Maria, RS, Brasil
2014**

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

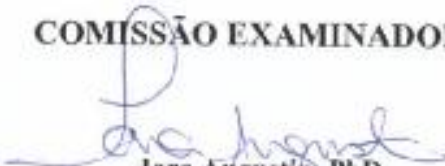
**UM APLICATIVO MÓVEL PARA REGISTRO AUTOMÁTICO DA
PRESENÇA ACADÊMICA VIA BLUETOOTH**

elaborado por
Fernando Weber Albiero

Fernando W. Albiero

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA


Lara Augustin, PhD
(Presidente/Orientadora)


Giovani Rubert Librelotto, PhD (UFSM)


Eduardo Kessler Piveta, PhD. (UFSM)

Santa Maria, 01 de Dezembro de 2014

AGRADECIMENTOS

Dedico os meus agradecimentos a minha família, que me ajudou, apoiou e aconselhou durante esta fase da minha vida. Meu pai Paulo Stangarlin Albiero e minha mãe Marli Terezinha Weber Carneiro. Agradeço em especial ao meu irmão, o professor Fábio Weber Albiero, que me inspirou a realizar o curso de graduação em Ciência da Computação além de sempre estar presente e disposto a ajudar nas dificuldades encontradas ao longo do curso. Gostaria também de agradecer a professora Iara Augustin e ao professor João Carlos Damasceno Lima pela sua orientação, permitindo que este trabalho fosse realizado. Agradeço também a todos os amigos que, de alguma forma, ajudaram durante este período na Universidade Federal de Santa Maria.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

UM APLICATIVO MÓVEL PARA REGISTRO AUTOMÁTICO DA PRESENÇA ACADÊMICA VIA BLUETOOTH

AUTOR: FERNANDO WEBER ALBIERO

ORIENTADORA: PROF. DRA. IARA AUGUSTIN

COORIENTADOR: PROF. DR. JOÃO CARLOS DAMASCENO DE LIMA

Data e Local da Defesa: Santa Maria, 01 de Dezembro de 2014.

O uso crescente de dispositivos móveis e a ampla difusão do uso de tecnologias sem fio, como o Bluetooth, vêm aumentando a demanda e estimulando cada vez mais o desenvolvimento de aplicativos. O uso desses aplicativos visa facilitar a vida dos usuários através da automação de tarefas simples do dia a dia. Algumas tarefas corriqueiras, como realizar o controle de presença acadêmica nas instituições de ensino, ainda é um processo realizado de forma manual e lenta, o que impacta diretamente no tempo de aula e em todo planejamento institucional. Diante disso, surgiu a necessidade de oferecer a essas instituições novas alternativas para o registro de frequência. Esse processo poderia ser feito de forma automática e transparente, tanto para os alunos quanto para os professores. Este projeto propõe uma forma alternativa para o controle de frequência acadêmica utilizando um aplicativo móvel no qual a frequência seria automaticamente apurada. A presença do aluno e do professor em uma determinada sala seria detectada através de dispositivos móveis que utilizem a tecnologia Bluetooth. Para validar a eficiência do aplicativo foram realizados testes em ambiente controlado e em sala de aula. Através dos testes foi possível validar a eficiência do aplicativo e alcançar os objetivos propostos neste trabalho.

Palavras-chave: Computação Móvel. Bluetooth. Frequência Acadêmica.

ABSTRACT

Trabalho de Graduação
Undergraduate Program in Computer Science
Universidade Federal de Santa Maria

A MOBILE APPLICATION FOR AUTOMATIC ACADEMIC RECORD ATTENDANCE BY BLUETOOTH

AUTHOR: FERNANDO WEBER ALBIERO

ADVISOR: PROF. DR. IARA AUGUSTIN

COADVISOR: PROF. DR. JOÃO CARLOS DAMASCENO DE LIMA

The increasing use of mobile devices and the widespread use of wireless technologies such as Bluetooth, are increasing demand and stimulating increasingly developing applications. The use of these applications seeks to make life easier for users by automating simple tasks of daily life. Some mundane tasks, such as performing control of academic presence in educational institutions, a process is still done manually and slowly, which impacts directly on class time and all over institutional planning. Thus, it became necessary to offer new alternatives to these institutions for the attendance record. This process could be done automatically and transparently, both for students and for teachers. This project proposes an alternative way to control academic frequency using a mobile application where the frequency would be automatically calculated. The presence of the student and the teacher in a particular room would be detected through mobile devices using Bluetooth technology. To validate the application efficiency were made tests in a controlled environment and in the classroom. Through the tests was possible to validate the application efficiency and achieve the objectives proposed in this work.

Keywords: Mobile Computing. Bluetooth. Academic Attendance.

LISTA DE ILUSTRAÇÕES

Figura 2.1 – Logotipo do Android	12
Figura 2.2 - Linha do tempo das versões Android	14
Figura 2.3 - Arquitetura do sistema operacional Android	15
Figura 2.4 - Ciclo de vida de uma atividade no Android	17
Figura 3.1 - Origem do logotipo Bluetooth	19
Figura 3.2 - Cronograma de evolução do Bluetooth	20
Figura 3.3 - Tipos de rede Bluetooth	22
Figura 3.4 - Topologia estabelecida entre dispositivos mestre e escravo	23
Figura 3.5 - Estados e fases da conexão em dispositivos Bluetooth	24
Figura 4.1 - Modelo 3D e componentes do dispositivo onBee	26
Figura 5.1 - Tela inicial do aplicativo BlueTApp Aluno	29
Figura 5.2 - Formado do arquivo.csv importado pelo aplicativo	30
Figura 5.3 - Tela inicial do aplicativo BlueTApp Professor .	31
Figura 5.4 - Formulário para inserção de novas turmas	31
Figura 5.5 - Tela inicial do aplicativo após a inserção de uma turma	32
Figura 5.6 - Tela exibida após um clique curto sobre alguma turmas	33
Figura 5.7 - Menu de opções exibido após um clique longo sobre alguma turma	34
Figura 5.8 - Tela exibida antes da realização de uma chamada	34
Figura 5.9 - Tela exibida ao término da busca por dispositivos	35
Figura 5.10 - "Chamada Manual"	36
Figura 5.11 - Formulário para inserção de um aluno	36
Figura 5.12 - "Remover Aluno"	37
Figura 5.13 - Confirmação da operação de exclusão de uma turma	38
Figura 5.14 - Registros de chamada	38
Figura 5.15 - Formato do arquivo.csv exportado pelo aplicativo	39
Figura 5.16 - Diagrama de casos de uso do aplicativo BlueTApp Professor	39
Figura 5.17 - Exemplo de utilização da coluna "Frequencia" no aplicativo	41
Figura 5.18 - Opções de chamada	42
Figura 5.19 - Diagrama de sequência	43
Figura 6.1 - Tempo para descoberta e pareamento	45
Figura 6.2 - Tempo para descoberta dos sinais Bluetooth	46

LISTA DE TABELAS

Tabela 3.1 - Classes de dispositivos Bluetooth	21
Tabela 4 1 - Diferenças entre o dispositivo onBee e o aplicativo proposto	27
Tabela 5.1 - Tabela turma.....	40
Tabela 5.2 - Tabela registro	41

SUMÁRIO

1 INTRODUÇÃO	10
2 ANDROID	12
2.1. Origem	13
2.2. Arquitetura	14
2.3. Desenvolvimento de Aplicações	16
3 BLUETOOTH	19
3.1. Origem	19
3.2. Características	20
3.3. Funcionamento	21
3.3.1 Conexão.....	23
3.4. Bluetooth no Android	25
4. AUTOMAÇÃO DO PROCESSO DE REGISTRO DA FREQUÊNCIA ACADÊMICA	26
5. O APLICATIVO	28
5.1 Versão Aluno	28
5.2 Versão Professor	30
5.2.1 Menu de opções.....	33
5.2.2 Persistência de Dados.....	40
5.2.4 Diagrama de Sequência.....	42
6. TESTES E VALIDAÇÃO DO APLICATIVO	44
7. CONCLUSÕES E TRABALHOS FUTUROS	47
REFERÊNCIAS	48

1 INTRODUÇÃO

Os alunos matriculados em uma disciplina devem possuir uma frequência mínima para a aprovação, como previsto na legislação Brasileira. A frequência mínima para efeitos de aprovação é de 75% (setenta e cinco por cento) sobre o total de aulas ministradas e demais atividades acadêmicas (BRASIL, 1996). A frequência acadêmica em qualquer nível é de grande importância para o um bom aprendizado. O simples fato de comparecer à aula e visualizar a explanação do professor agrega muito mais valor do que uma simples leitura do material de estudo. E além dos conceitos técnicos adquiridos, a participação nas atividades acadêmicas propicia a transmissão de normas, valores, e atitudes relacionadas à vida social que auxiliam no desenvolvimento do aluno como cidadão.

Na maioria das instituições de ensino, a frequência ainda é apurada da forma tradicional, através de papel e caneta, o que demanda tempo e tem impacto direto no planejamento institucional. Em alguns casos, uma fatia considerável do tempo de aula é gasta no processo, tempo esse que poderia ser utilizado para o ensino e aprendizagem dos alunos. Esse processo poderia ser feito de forma automática e transparente tanto para os alunos quanto para os professores através de um sistema ubíquo no qual a frequência seria automaticamente apurada por meio da detecção da presença do aluno e do professor em uma determinada sala de aula em um determinado horário (WEISER, 1991).

No mercado atual as opções para a automação do processo de frequência são limitadas, porém com a evolução da computação móvel e o surgimento de novas tecnologias de comunicação a curta distância, como o Bluetooth, a possibilidade de expandir esse número de opções vem se tornando cada vez maior.

Diante disso, esse Trabalho de Graduação (TG) visa oferecer às instituições de ensino uma forma alternativa, simples e sem custo de realizar o registro da frequência acadêmica através de um aplicativo móvel que faz uso do sistema operacional móvel Android e da tecnologia Bluetooth.

Este TG visa desenvolver um aplicativo móvel para o sistema operacional Android que, através do uso da tecnologia Bluetooth, automatize o processo de registro da frequência acadêmica. O aplicativo será responsável por todo o gerenciamento da atividade de chamada diminuindo assim o número de intervenções por parte do professor. Dessa maneira espera-se tornar o processo mais ágil de forma que sobre mais tempo para as atividades desenvolvidas em sala de aula.

Além disso, este trabalho propõe ainda um formato simples de arquivo para exportação, que pode vir a ser aproveitado futuramente pelas instituições de ensino, de forma a interpretar as informações capturadas pelo aplicativo.

Este trabalho está organizado da forma que segue. O capítulo 2 apresenta conceitos básicos sobre o sistema operacional móvel Android, seu funcionamento, e sua arquitetura. Tal capítulo apresenta ainda as vantagens pelas quais tal sistema operacional foi escolhido além de uma revisão sobre o desenvolvimento de aplicativos no Android.

O capítulo 3 revisa o uso da tecnologia Bluetooth apresentando suas características, funcionamento e limitações. Além disso, são apresentados também os tipos e fases de conexões no Bluetooth e como tal tecnologia é utilizada no sistema operacional Android.

O capítulo 4 faz uma revisão sobre os trabalhos relacionados para a automação do processo de registro da frequência acadêmica. É apresentada também uma tabela comparativa com o aplicativo proposto neste trabalho.

O capítulo 5 apresenta o aplicativo desenvolvido neste TG. Neste capítulo é explicado detalhadamente todos os níveis do aplicativo, desde sua primeira execução até o momento da exportação de dados.

O capítulo 6 mostra os testes realizados para validar o aplicativo e as dificuldades encontradas ao longo do desenvolvimento.

Por fim, o Capítulo 7 apresenta as conclusões do TG e os possíveis trabalhos futuros a serem desenvolvidos.

2 ANDROID

O Android é um sistema operacional completo para tecnologia móvel, envolvendo um pacote com programas, que conta com um, middleware, aplicativos e interface do usuário. Foi inicialmente desenvolvido pela Google e posteriormente pelo *Open Handset Alliance* (OHA), que é um consórcio com mais de 80 empresas entre as quais estão a HTC, LG, Motorola, Samsung, Sony Ericsson, Toshiba, Sprint Nextel, China Mobile, T-Mobile, Asus, Intel, Garmin e outras mais, com a intenção de permitir aos desenvolvedores criar aplicações que possam tirar total proveito do que um aparelho portátil possa oferecer. O Android é baseado no kernel do Linux e é composto por um conjunto de ferramentas que atua em todas as fases do desenvolvimento do projeto, desde a execução até a criação de softwares específicos (PEREIRA; SILVA 2009). Apesar de ter sido desenvolvido inicialmente para smartphones, hoje é usado em diversas outras aplicações como tablets, netbooks, relógios, etc. A figura 2.1 apresenta o logotipo do Android.



Figura 2.1 – Logotipo do Android.

Os aplicativos do Android são desenvolvidos utilizando Java e podem ser portados com bastante facilidade. O Android pode ser visto como uma máquina virtual Java rodando sobre o kernel do Linux, dando suporte para o desenvolvimento de aplicações Java através de um conjunto de bibliotecas e serviços (GOMES; FERNANDES, 2012).

O Android também inclui um motor gráfico 3D baseado no suporte de hardware, suporte de banco de dados alimentado por SQLite, e um navegador web integrado. Além disso, o layout de interface do usuário é baseado em XML (*eXtensible Markup Language*) (DIMARZIO, 2008).

Muitas características presentes no Android, tais como gráficos 3D e suporte a banco de dados nativo, também estão disponíveis em outras plataformas móveis. Porém o Android

conta com algumas vantagens. Apenas no Android há um componente que permite exibir e manipular um mapa do Google Maps, serviço de mapas do Google, dentro de uma aplicação. Outra vantagem do Android é que todos os aplicativos são criados igualmente. Ou seja, nele não há diferença entre aplicações nativas e as demais aplicações, o que possibilita uma grande personalização do sistema, ao permitir a substituição completa de aplicativos nativos por outros, criados por terceiros. Além disto, todos os aplicativos tem acesso às mesmas funcionalidades (MEIER, 2009).

2.1. Origem

Os celulares até meados 2002 serviam apenas para executar e receber chamadas, acompanhar seus contatos, e, eventualmente, enviar e receber mensagens de texto curtas, pois nem as câmeras em celulares eram facilmente encontradas. Ninguém ainda via a necessidade de navegação na Internet, tocar MP3, MP4, ou qualquer uma das muitas funções que estamos acostumados a usar hoje em dia. No entanto, mesmo se a necessidade estivesse presente na época, a falta de memória do dispositivo e capacidade de armazenamento reduzida eram obstáculos a serem superados (DIMARZIO, 2008)

Ao decorrer dos anos, com evolução da tecnologia, principalmente na questão de hardware e tamanhos dos dispositivos, os aparelhos móveis começaram a ganhar outras funções além da chamada por voz e o envio de SMS. Devido a essa evolução, foi necessário um avanço nas aplicações e nos sistemas operacionais para gerenciar as novas funções disponíveis nesses dispositivos (JÚNIOR; CASTRO, 2011).

De acordo com (CINDRAL, 2011), um dos grandes avanços ocorreu com o surgimento do Symbian, um SO mobile aberto. Adotado pela Nokia, o sistema permitiu que milhares de desenvolvedores no mundo pudessem criar aplicativos baseado nos novos recursos. Outros nomes como iOS, BlackBerryOS, além do Symbian, ganharam espaço no mercado, e lideravam praticamente absolutos até a Google lançar seu próprio Sistema Operacional (SO) junto com a OHA.

A história do sistema operacional Android inicia-se em julho de 2005 quando a Google adquiriu a Android Inc., uma pequena empresa em Palo Alto, California, USA e passou a desenvolver uma plataforma para sistemas móveis baseado em Linux, com o objetivo de ser uma plataforma flexível, aberta e de fácil migração para os fabricantes. Nascia

assim o embrião do sistema operacional Android. Esse sistema foi desenvolvido na plataforma Java e atualmente é mantido pela OHA.

A primeira versão comercial do Android, o Android 1.0, foi lançado em setembro de 2008. Desde abril de 2009, as versões Android foram batizadas com um codinome de doce e lançadas em ordem alfabética como mostrado na figura 2. Já se sabe que a próxima versão do Android será batizada de *Key Lime* (torta de limão) (JUNIOR; 2014).

A plataforma Android é a mais popular atualmente, o Sistema operacional Android, alcançou 81% de participação mundial no mercado de smartphones no terceiro trimestre de 2013. No Brasil não é diferente, no terceiro trimestre de 2013, foram comercializados 10,4 milhões smartphones no país. Destes smartphones vendidos, 90% utilizam sistema operacional Android (INTERNATIONAL DATA CORPORATION, 2013).



Figura 2.2 - Linha do tempo das versões Android. Fonte: JUNIOR, 2014.

2.2. Arquitetura

Como dito anteriormente, o Android é um sistema operacional baseado no kernel do Linux. Apesar disso, existe pouca coisa em comum com as distribuições Linux convencionais. Sua arquitetura possui basicamente quatro camadas, como mostrado na Figura 2.3.

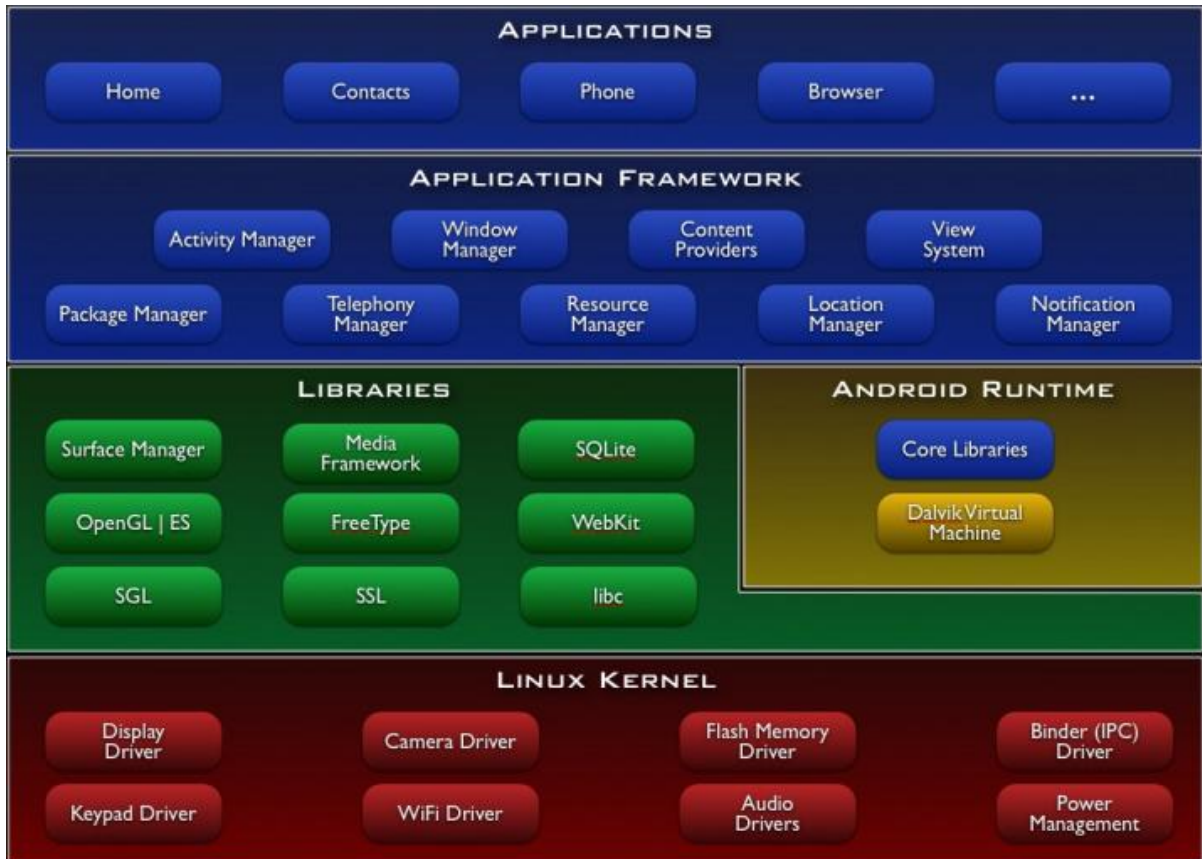


Figura 2.3 - Arquitetura do sistema operacional Android. Fonte: GOMES; FERNANDES, 2012.

Na base da pilha está um Linux Kernel versão 2.6. Além de permitir uma abstração entre o hardware e o software, ele é o responsável pelos principais serviços do sistema como gerenciamento de memória e gerenciamento de processos (MARTINS, 2009). Várias funções do kernel são utilizadas diretamente pelo Android, mas muitas modificações foram feitas para otimizar memória e tempo de processamento das aplicações. O Linux 2.6 foi escolhido por já conter uma grande quantidade de drivers de dispositivos sólidos e por ter um bom gerenciamento de memória e processos (GOMES; FERNANDES, 2012).

Embora as aplicações para Android sejam escritas utilizando a linguagem Java, elas não são executadas em uma máquina virtual Java tradicional e sim em outra, chamada *Dalvik*. Assim como diversos componentes da plataforma, esta máquina virtual é otimizada especialmente para dispositivos móveis (MARTINS, 2009). Essa máquina virtual foi construída pelos engenheiros da Google, para obter um consumo mínimo de memória e isolamento de processos. Ela permite que as aplicações escritas em linguagem Java sejam executadas normalmente (GOMES; FERNANDES, 2012). Cada aplicação do Android roda em seu próprio processo e cada processo é uma instância da máquina virtual Dalvik, criada

para que o dispositivo possa rodar múltiplas máquinas virtuais eficientemente. Os arquivos são executados no formato Dalvik Executable (.dex) e são otimizados para consumo de memória mínimo. Os arquivos são criados por um compilador Java, que converte o resultado no formato .dex (JÚNIOR; CASTRO, 2011).

A plataforma também inclui uma coleção de bibliotecas C/C++ utilizadas por vários componentes do sistema. Elas são as responsáveis por prover, dentre outros, áudio, vídeo, gráficos, banco de dados e navegador e são expostas para os desenvolvedores através do framework de aplicação (MARTINS, 2009). Para as bibliotecas gráficas, há uma biblioteca 3D para controle e acesso a todos os recursos do hardware de vídeo. A implementação foi baseada no OpenGL 9 (*Open Graphics Library*) (JÚNIOR; CASTRO, 2011).

A camada de framework foi desenvolvida para simplificar a reutilização dos componentes. A arquitetura deste framework nativo disponibiliza aos desenvolvedores as mesmas APIs (Applications Programming Interface) utilizadas para a criação de aplicações originais do sistema operacional Android. Desta forma qualquer desenvolvedor pode construir um aplicativo e disponibilizar suas “capacidades”, permitindo que elas sejam utilizadas por outros programas (JÚNIOR; CASTRO, 2011). Desta maneira o framework funciona como um link com a camada de bibliotecas do sistema operacional, as quais serão acessadas através de APIs contidas no framework (GOMES; FERNANDES, 2012).

O Android já vem equipado com diversas aplicações. Dentre elas, há cliente de e-mail, navegador, despertador, jogos, mapas e gerenciador de contatos. Estas aplicações estão localizadas no topo da pilha, juntamente com todas as demais, criadas por terceiros, instaladas no sistema (MARTINS, 2009).

2.3. Desenvolvimento de Aplicações

Como mencionado anteriormente, aplicações Android são escritas em linguagem Java. Para auxiliar no desenvolvimento das aplicações, normalmente são utilizadas IDE's (*Integrated Development Environment* ou Ambiente Integrado de Desenvolvimento), que são programas de computador que reúnem características e ferramentas de apoio ao desenvolvimento de *software*. Segundo (ANDROIDZ, 2013), um dos IDEs mais utilizados no desenvolvimento de aplicativos Java é o Eclipse (IDE), porém para o desenvolvimento deste projeto foi utilizado o Android Studio, a IDE da Google para o desenvolvimento Android.

Existem diversas ferramentas que auxiliam no desenvolvimento de aplicativos para o sistema Android. A mais completa é o kit de desenvolvimento Android SDK, que contém um ambiente com as características e especificações do Android e já vem instalado no Android Studio (GOMES; FERNANDES, 2012).

Uma aplicação é construída através de componentes que são instanciados no momento em que se tornam necessários. Existem quatro tipos de componentes básicos: atividades, serviços, provedores de conteúdo e receptores de broadcast. Uma atividade geralmente corresponde a uma tela para o usuário e é uma das mais importantes classes de um aplicativo Android. Serviços não possuem uma interface visual e são utilizados para executar processamentos em segundo plano. Os provedores de conteúdo servem para disponibilizar dados específicos de uma aplicação para outras aplicações. Por fim, os receptores de broadcast são componentes que ficam inativos e respondem a eventos (MARTINS, 2009).

Um dos pontos chaves para o desenvolvimento de aplicativos Android é o ciclo de vida das atividades. Com este conceito é possível entender seu funcionamento e realizar alterações no que for necessário (JÚNIOR; CASTRO, 2011).

Cada atividade de uma aplicação tem um ciclo de vida bem definido que inicia no momento em que é instanciado e acaba quando é destruído. Durante a execução, uma atividade pode ter seu estado alterado diversas vezes. Uma atividade possui essencialmente três estados: ativo, pausado e parado. As transições entre estes estados são notificadas através de chamadas a sete métodos especiais, conforme demonstrado na Figura 4. Estes métodos são utilizados para realizar as ações apropriadas a uma determinada mudança de estado (MARTINS, 2009).

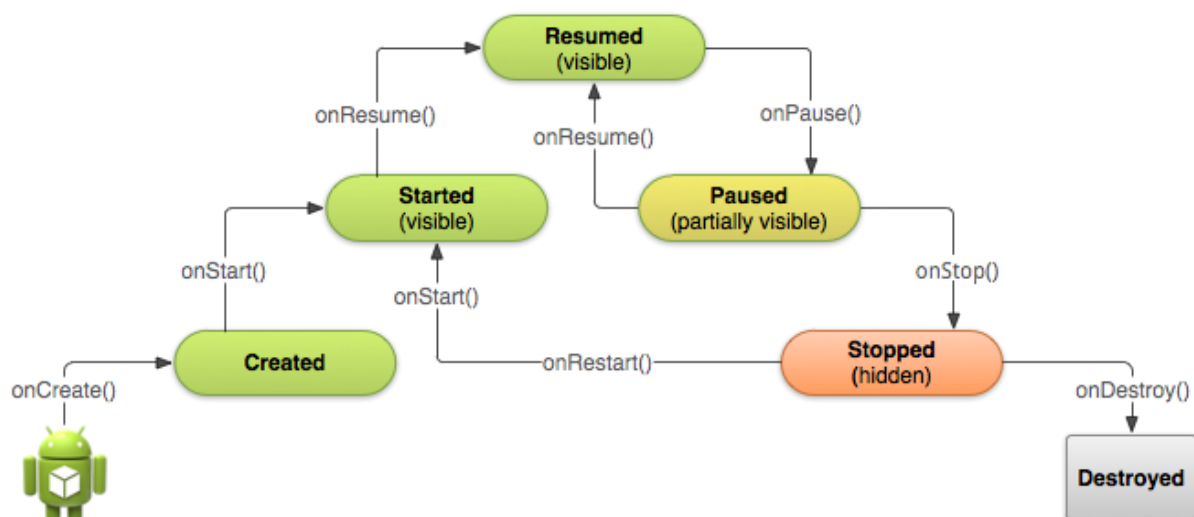


Figura 2.4 - Ciclo de vida de uma atividade no Android.

Durante o tempo entre as chamadas dos métodos *onStart()* e *onStop()*, a atividade pode ser vista pelo usuário. Entre os métodos *onResume()* e *onPause()* a atividade está na frente de todas as outras atividades e o usuário está interagindo com ela. E, devido ao fato de o *onPause()* ser o único método cuja chamada é garantida antes do processo ser eventualmente destruído, ele deve ser utilizado para armazenar, de forma persistente, dados para posterior restauração.

O desenvolvedor não tem controle sobre as mudanças de estado de uma aplicação, isso é responsabilidade do sistema operacional, no entanto, ele pode ser notificado quando um estado está prestes a mudar através da chamada de um método *on[Evento]()*. Sobrescrevendo esses métodos na sua classe de atividade, o Android vai executá-los no momento apropriado (BURNETTE, 2009).

O código, depois de compilado, é empacotado juntamente com outros recursos utilizados pela aplicação em um arquivo com o sufixo *.apk*. Este arquivo é o veículo de distribuição para os usuários instalarem a aplicação em seus dispositivos. Cada pacote *.apk* contém um arquivo de manifesto onde estão declarados todos os componentes da aplicação. O arquivo é estruturado em linguagem XML e também é utilizado para declarar as bibliotecas, permissões, versão e requisitos (MARTINS, 2009).

3 BLUETOOTH

A tecnologia Bluetooth é, basicamente, um padrão para comunicação sem fio de baixo custo e curto alcance, que opera através de conexões de rádio. A tecnologia Bluetooth permite a comunicação entre uma ampla variedade de dispositivos de computação, de telecomunicações e eletrodomésticos de uma forma bastante simples, sem a necessidade de qualquer tipo de cabeamento. A ideia é permitir a interligação desses dispositivos de uma forma automática e sem que o usuário necessite se preocupar com isso. (OLIVEIRA, 2005).

Baseada em tecnologia de rádio frequência (RF), a tecnologia Bluetooth busca promover uma solução de baixo consumo de energia para transmissão digital de dados e voz. Diferentemente da comunicação por infravermelho, a rádio frequência não necessita de linha de visão livre entre os pontos comunicantes. (MARTINCOSKI, 2003).

O nome Bluetooth foi inspirado no rei viking Harald Bluetooth, que unificou o povo escandinavo durante o século X (BISDIKIAN, 2001). De maneira semelhante, a tecnologia Bluetooth visa unir ou conectar diferentes tipos de dispositivos de diferentes fabricantes. Uma curiosidade acerca do logotipo, mostrado na figura 3.1, é que o símbolo é a união de algumas runas (conjunto de alfabetos relacionados que utilizam letras características) nórdicas, como no caso de: H - H (Hagall) e B - B (Berkanan), que são as letras iniciais do nome do rei viking (PASSOS, 2011).

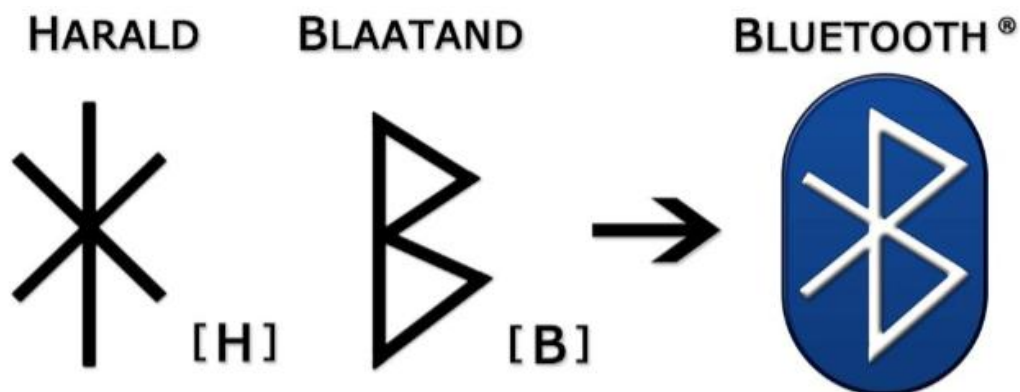


Figura 3.1 - Origem do logotipo Bluetooth. Fonte: PASSOS, 2011.

3.1. Origem

A ideia que eventualmente se transformou na tecnologia Bluetooth nasceu em 1994, quando engenheiros da Ericsson Mobile Communications pesquisavam a praticidade de uma

interface de rádio de baixo consumo de energia e pouco custo entre telefones e seus acessórios (OLIVEIRA, 2005). Em fevereiro de 1998, outras quatro grandes empresas (Intel, IBM, Toshiba e Nokia) uniram-se à Ericsson e formaram o *Bluetooth Special Interest Group* (SIG). Como resultado desta união, surge em julho de 1999, a versão 1.0 da especificação Bluetooth. Em dezembro de 1999, o grupo de promotores do Bluetooth foi ampliado com o ingresso de mais quatro grandes companhias: Microsoft, Lucent, 3Com e Motorola (MARTINCOSKI, 2003). Atualmente, o SIG é composto por mais de 2000 empresas.

De acordo com o site do Bluetooth (BLUETOOTH, 2014), ao final dos anos 2000 chega ao mercado o primeiro celular que fazia uso da tecnologia Bluetooth. Em seguida, em 2001, surgem a primeira impressora e o primeiro notebook utilizando Bluetooth. A partir daí, o crescimento de utilização dessa tecnologia foi exponencial, se disseminou pelo mercado e continua crescendo. A figura 3.2 apresenta o cronograma de evolução do Bluetooth ao longo dos anos:

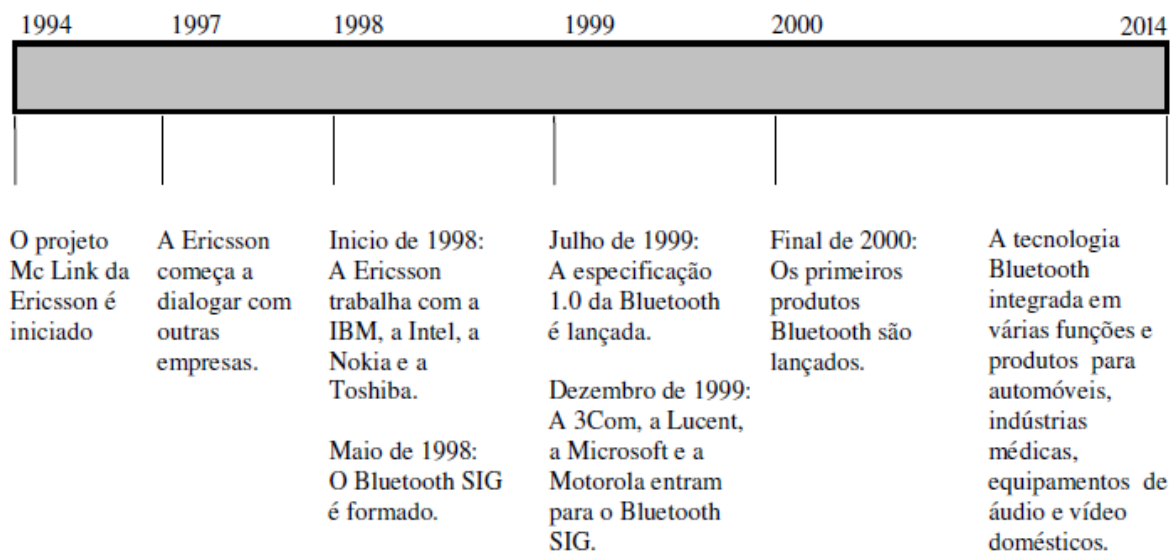


Figura 3.2 - Cronograma de evolução do Bluetooth. Fonte: OLIVEIRA, 2005.

3.2. Características

O padrão Bluetooth determina que todos os dispositivos se conectem por meio de um conjunto específico de protocolos e frequências de rádio (OLIVEIRA, 2005).

Os dispositivos Bluetooth operam na banda ISM (*industrial, scientific and medical*) que varia de 2,4 GHz a 2,4835 GHz. Esta banda de frequências está disponível mundialmente,

sendo destinada para usos gerais. As bandas ISM não requerem licença para operação, porém obedecem a um conjunto de especificações sobre interferência e emissão espectral de potência. Além do Bluetooth, diversas outras fontes de rádio frequência ocupam a banda ISM, como por exemplo, equipamentos de curto alcance (alarmes de carro, telefones sem fio, redes digitais *wireless*, etc.) e fontes de ruído aleatório (fornos de micro-ondas, lâmpadas de vapor de sódio, etc.). Este compartilhamento de banda contribui significativamente para a presença de sinais interferentes, tornando a banda ISM um meio não muito estável ou confiável (MARTINCOSKI, 2003).

Para minimizar este tipo de problema a tecnologia Bluetooth utiliza uma técnica denominada salto de frequência de alargamento de banda. É uma técnica comum onde o sinal salta de uma frequência para outra durante o curso de uma única transmissão (OLIVEIRA, 2005).

A tecnologia Bluetooth ainda especifica três níveis diferentes de potência de transmissão. A tabela 3.1 mostra as três classes possíveis de dispositivos e suas configurações.

Classe	Potência Máxima de Transmissão (mW)	Potência Máxima de Transmissão (dBm)	Alcance
Classe 1	100 mW	20 dBm	100 m
Classe 2	2.5 mW	4 dBm	10 m
Classe 3	1 mW	0 dBm	1 m

Tabela 3.1 - Classes de dispositivos Bluetooth. Fonte: TAKASHI, 2003.

Estas classes de potência possibilitam diferentes alcances para os dispositivos Bluetooth. Atualmente a maioria dos fabricantes está produzindo módulos de um mega watt (classe 3), com alcance de até 1 metros.

3.3. Funcionamento

Para haver uma conexão entre dois dispositivos, eles devem conter um rádio Bluetooth, sendo este um pequeno chip de computador que estabelece e gerencia as conexões individuais com baixo consumo de energia. O padrão Bluetooth determina que todos os dispositivos se conectem por meio de um conjunto específico de frequências de rádio e também determina os protocolos necessários para transmitir e receber dados, sendo que todos

os dispositivos Bluetooth devem usar os mesmos protocolos para que todos falem a mesma linguagem eletrônica (OLIVEIRA, 2005).

Quando dois dispositivos Bluetooth realizam a conexão, diz-se que eles estão emparelhados. O emparelhamento é o processo de criação e troca de uma chave-senha entre os dois dispositivos a fim de estabelecer uma nova relação entre eles. Após emparelhar os dispositivos, uma chave-senha é armazenada para que a autenticação ocorra automaticamente durante conexões futuras.

De acordo com as especificações Bluetooth, oito é o número máximo de dispositivos que podem ser conectados de cada vez. No entanto, o uso de três a quatro dispositivos é um limite prático, dependendo do tipo dos dispositivos e perfis que estão sendo usados (APPLE SUPPORT, 2014).

Quando dois dispositivos estabelecem uma conexão se cria uma rede denominada Piconet. Várias Piconets podem ser conectadas formando uma Scatternet como mostrado na Figura 7.

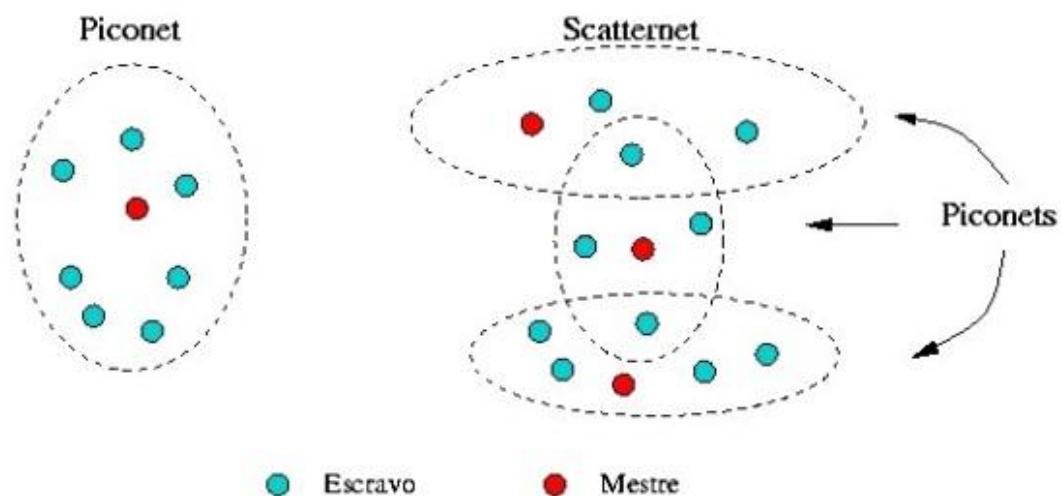


Figura 3.3 - Tipos de rede Bluetooth. Fonte: OLIVEIRA, 2005.

As Piconets são formadas por um dispositivo mestre que pode comunicar-se simultaneamente com até sete dispositivos escravos, sendo que os dispositivos escravos só podem se comunicar diretamente com o mestre e nunca com outro escravo, formando assim uma topologia em estrela como mostrado na figura 8. O dispositivo mestre é quem requisita a conexão e o escravo é quem aceita.

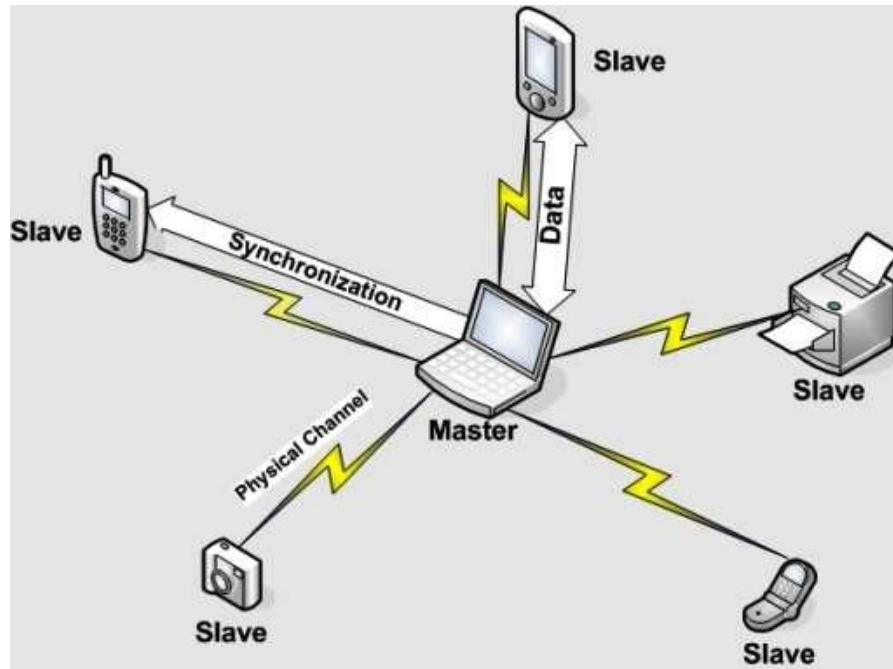


Figura 3.4 - Topologia estabelecida entre dispositivos mestre e escravo. Fonte: ZANON; FELDEN [s.d.].

A dimensão de oito dispositivos é suficiente para agrupar um microcomputador com todos os seus periféricos ou dispositivos Bluetooth transportados por uma pessoa (ZANON; FELDEN, [s.d.]).

3.3.1 Conexão

Os dispositivos Bluetooth possuem dois estados operacionais de conexão. O primeiro, quando o dispositivo já pertence a uma piconet, é chamado de estado Conectado. No segundo, quando o dispositivo não pertence a nenhuma piconet, nem se encontra em processo de formação de uma piconet ou união com uma já existente, é chamado de estado *Stand-By* (espera). O estado *Stand-By* é o estado operacional padrão para um dispositivo Bluetooth.

Quando dois dispositivos Bluetooth estiverem dentro da distância de alcance mútuo, é possível estabelecer uma conexão para comunicação. O estabelecimento das conexões é dado em duas fases. Primeiro, o dispositivo mestre entra em modo *Inquiry* (investigação), onde busca por dispositivos visíveis. Para um dispositivo escravo ser visível, é necessário habilitar o modo *Inquiry Scan* (busca por investigações). Através do *Inquiry*, o dispositivo mestre obtém uma lista dos dispositivos que estão dentro do seu alcance e seus respectivos

endereços. Cada dispositivo Bluetooth possui um endereço (Bluetooth Address) de 48 bits que é único no mundo.

Após escolher com qual dispositivo fará a conexão, o dispositivo mestre entra em modo *Paging* (convocação). Neste modo, o dispositivo mestre requisita a conexão ao dispositivo escravo de interesse, utilizando para isso o Bluetooth Address do dispositivo escravo obtido no modo *Inquiry*. O dispositivo escravo pode ser configurado para responder ou não às convocações do dispositivo mestre. Para responder, precisa estar com o modo *Page Scan* (busca por convocações) habilitado. Em novas conexões, caso o dispositivo mestre já conheça o endereço do dispositivo escravo, não há a necessidade de passar novamente pelo modo *Inquiry*.

Os estados intermediários *Inquiry* e *Page* implementam conceitos de “identificabilidade” e “conectabilidade” para dispositivos Bluetooth (MARTINCOSKI, 2003). A figura 3.6 mostra os estados da conexão Bluetooth.

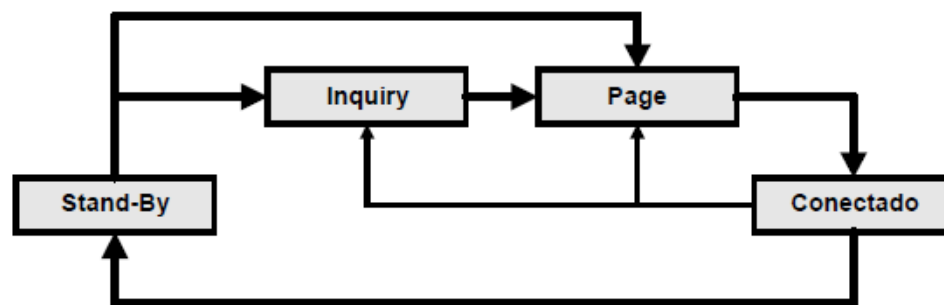


Figura 3.5 - Estados e fases da conexão em dispositivos Bluetooth. Fonte: MARTINCOSKI, 2003.

De acordo com (OLIVEIRA 2005), no estado Conectado, o dispositivo pode estar em quatro modos de conexão:

1. *Active*: O dispositivo está transmitindo ou recebendo ativamente dentro de uma piconet, sincronizando automaticamente com a unidade mestra;
2. *Sniff*: O dispositivo monitora a piconet a uma taxa reduzida, diminuindo o seu consumo de energia;
3. *Hold*: Este modo é aplicado geralmente a dispositivos com transferências de dados simples, apenas o timer interno do dispositivo fica ativo;
4. *Park*: Neste modo, o dispositivo permanece sincronizado com a piconet, mas não participa do tráfego de dados, suspende o seu endereço MAC (*Media Access Control*).

3.4. Bluetooth no Android

A plataforma Android inclui suporte para a pilha de rede Bluetooth, permitindo que um dispositivo troque dados com outros dispositivos sem a necessidade de cabos. A estrutura dos aplicativos fornece acesso à funcionalidade Bluetooth através das APIs do Android Bluetooth. Essas APIs permitem que as aplicações estabeleçam conexão ponto-a-ponto ou multiponto com outro dispositivo Bluetooth (ANDROID DEVELOPERS, 2014).

Através das APIs do Bluetooth uma aplicação Android pode executar diferentes funções como: verificar a existência de outros dispositivos Bluetooth no seu raio de alcance, estabelecer conexão com outros dispositivos, enviar e receber dados, gerenciar múltiplas conexões, entre outras. Todas as APIs do Bluetooth estão disponíveis dentro do pacote `android.bluetooth`. Dentre as principais classes necessárias para criar uma conexão Bluetooth estão:

BluetoothAdapter: Representa o adaptador Bluetooth local. É o ponto de entrada para toda a interação Bluetooth. Usando isso, é possível descobrir outros dispositivos Bluetooth, consultar a lista de dispositivos conectados (emparelhados), instanciar um dispositivo usando um endereço MAC conhecido, e criar um socket para escutar as comunicações de outros dispositivos.

BluetoothDevice: Representa um dispositivo Bluetooth remoto. Usado para solicitar uma conexão com um dispositivo remoto através de um socket ou para consultar informações sobre o dispositivo, como seu nome, endereço, classe e estado da ligação.

BluetoothSocket: Representa a interface para um socket de Bluetooth. Este é o ponto de conexão que permite que um aplicativo troque dados com outro dispositivo Bluetooth via `InputStream` e `OutputStream`.

BluetoothClass: Descreve as características gerais e capacidades de um dispositivo Bluetooth. Esta é um conjunto de propriedades *read-only* (somente leitura) que definem as classes de dispositivos e seus serviços.

4. AUTOMAÇÃO DO PROCESSO DE REGISTRO DA FREQUÊNCIA ACADÊMICA

O processo de controle de frequência, que ocorre nas instituições de ensino, atualmente é executado de forma manual utilizando papel e caneta, já está defasado, demanda tempo e acaba por comprometer um período importante da aula que poderia ser utilizado para o ensino dos alunos. O impacto no tempo de aula significa que menos tempo é dedicado para o processo de ensino aprendizagem, ou até mesmo a não realização da apuração da frequência pelos professores. Esse processo poderia ser feito de forma automática e transparente para os alunos e professores através de um sistema ubíquo no qual a frequência seria automaticamente apurada por meio da detecção da presença do aluno e do professor em uma determinada sala de aula em um determinado horário (CHAMON, 2014).

Porém atualmente, no mercado, ainda não existem muitas alternativas de aplicativos ou tecnologias que realizem essa tarefa. A Universidade Católica de Minas desenvolveu em parceria com a Universidade Centro Leste um sistema ubíquo para o registro automático da presença acadêmica de alunos. Bem diferente do proposto por este projeto, tal sistema funciona através do uso de um cartão eletrônico integrado com um microchip de rádio frequência que é alimentado por uma pequena bateria e busca por uma rede sem fio. Uma vez encontrada a rede, o microchip envia informações para a rede, identificando e localizando o aluno que precisa permanecer no mínimo quinze minutos ao alcance da mesma para ter sua presença registrada. Todos esses componentes formam um pequeno dispositivo portátil como mostrado na figura 4.1. Chamado de onBee, tal dispositivo é carregado pelos alunos e tem custo de vinte e cinco reais cada (CHAMON, 2014).



Figura 4.1 - Modelo 3D e componentes do dispositivo onBee. Fonte: CHAMON, 2014.

Diante disso, esse projeto visa oferecer às instituições de ensino uma forma nova, mais simples e mais barata de realizar o registro de frequência de seus alunos através de um aplicativo móvel que automatize o processo de controle de frequência acadêmica utilizando a plataforma Android e a tecnologia Bluetooth.

A tabela 4.1 apresenta as principais diferenças entre o dispositivo onBee e o aplicativo proposto neste projeto.

	onBee	Aplicativo
Tecnologia utilizada	Wi-fi	Bluetooth
Tempo para o aluno ser localizado	15 - 60 min	15-30 seg
Tempo de vida	8 anos e 8 meses	Ilimitado
Custo	R\$25,00	Sem custo

Tabela 4.1 - Diferenças entre o dispositivo onBee e o aplicativo proposto neste trabalho.

5. O APLICATIVO

O aplicativo proposto nesse projeto foi desenvolvido em linguagem Java para a versão 2.3 (Gingerbread) do Android, o que o possibilita sua utilização em 99,2% dos dispositivos móveis que utilizem a plataforma Android. O aplicativo foi batizado de BlueTApp e visa automatizar e agilizar o processo de controle de frequência realizado nas salas de aula diminuindo o número de intervenções por parte do professor durante o processo.

A ideia central do aplicativo é capturar os sinais Bluetooth dos dispositivos móveis dos alunos e através das informações obtidas a partir desses sinais verificar a presença ou ausência do aluno em sala de aula. Para tornar isso possível o aplicativo foi desenvolvido em duas versões, uma para os alunos e outra para os professores.

Como mencionado anteriormente o aplicativo foi desenvolvido para a versão 2.3 do Android, porém como essa é uma versão muito antiga, o aplicativo também foi testado e adaptado para funcionar em versões mais recentes. Os testes foram realizados nas versões 2.3 e 4.4 da plataforma.

5.1 Versão Aluno

O aplicativo versão aluno é bastante simples e tem como objetivo principal padronizar o nome dos dispositivos móveis dos alunos. Através dessa padronização, o dispositivo móvel do professor poderá reconhecer e diferenciar cada dispositivo através da captura do sinal Bluetooth.

Essa versão foi batizada de BlueTApp Aluno e funciona da seguinte maneira, após o preenchimento do campo “matricula”, que tem valor numérico e é de preenchimento obrigatório (não pode ser vazio, caso isso ocorra um alerta é exibido), e a confirmação da operação com o clique no botão “confirmar” o aplicativo solicita ao aluno permissão para utilizar o adaptador Bluetooth do aparelho e torná-lo visível para outros dispositivos. Se a solicitação for aceita, o aplicativo verifica o estado atual do adaptador Bluetooth. Se o adaptador estiver desligado o aplicativo modifica o seu estado para “ligado” e altera o nome do dispositivo para a matrícula informada no campo de texto. Se o adaptador já estiver ligado o aplicativo simplesmente modifica o nome do dispositivo para a matrícula informada. Após essa operação o aplicativo torna o dispositivo visível para outros dispositivos e uma mensagem é mostrada ao aluno informando que seu aparelho está pronto para a chamada por Bluetooth.

O tempo de visibilidade do dispositivo varia para cada versão do Android. Conforme a documentação do próprio Android as versões 2.* tem um tempo máximo de 300 segundos, já a partir das versões 3.* esse limite foi aumentado podendo chegar a até 3600 segundos (ANDROID DEVELOPERS, 2014). O tempo máximo de visibilidade foi utilizado nesse projeto.

A versão BlueTApp Aluno não é de uso obrigatório para o funcionamento da versão BlueTApp Professor, ela é simplesmente uma ferramenta facilitadora que pode ser substituída pela forma manual de execução dos passos que o aplicativo realiza. Para isso o aluno deve acessar, na versão 2.3 do Android, Menu > Configurações > Conexões sem fio e rede > Configurações Bluetooth e ativar o adaptador Bluetooth, alterar o nome do dispositivo para sua matrícula e torna-lo visível durante o tempo desejado.

Na versão 4.4 do Android o aluno deve acessar Menu > Configurações > Bluetooth para executar as operações.

A figura 5.1 apresenta o aplicativo BlueTApp Aluno.

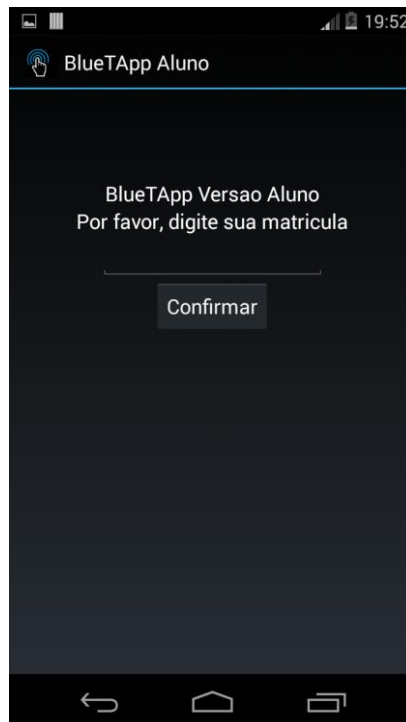


Figura 5.1 - Tela inicial do aplicativo BlueTApp Aluno.

5.2 Versão Professor

O aplicativo na versão professor foi batizado de BlueTApp Professor e é quem realiza todo gerenciamento do processo de chamada de forma automática e transparente. Antes da execução da aplicação o professor deve acessar o portal do professor da UFSM e fazer o download dos arquivos .csv referente as turmas as quais ele irá realizar a chamada. Os dados contidos nos arquivos .csv serão importados e armazenados pelo aplicativo no momento em que o professor tentar inserir uma nova turma. A figura 5.2 mostra o formato do arquivo.csv utilizado pela aplicação.

```

nome,matrícula,data de inclusão na turma,curso,e-mail
ALAN PERALTA DUTRA,201222150,28/07/2014,Bacharelado em Sistemas de Informação,adutra@inf.ufsm.br
ANDRE LUIS GIACOMINI BRENDLER,201240234,28/07/2014,Bacharelado em Sistemas de Informação,andre_brendler@hotmail.com
CRISTOFER ROVIAN CLARO PEDROZO,2921130,25/08/2014,Curso de Engenharia de Computação,crisrcp@gmail.com
EMIR NATAL BRITO DE CAMPOS JUNIOR,2920918,28/07/2014,Bacharelado em Sistemas de Informação,emircamposjunior@gmail.com
FERNANDO QUATRIN CAMPAGNOLO,201120061,11/08/2014,Bacharelado em Sistemas de Informação,fcampagnolo@inf.ufsm.br
HERSON FLACH NADALON,201040150,28/07/2014,Bacharelado em Sistemas de Informação,hnadalon@inf.ufsm.br
JUNIOR DE SOUZA,201020579,22/08/2014,Curso de Engenharia de Computação,junior.souza2102@gmail.com
MARCO ANTONIO DALCIN,201020318,25/08/2014,Curso de Engenharia de Computação,marco_dalcin@hotmail.com
MICHEL RAMOS MENDONCA,201020865,28/07/2014,Bacharelado em Sistemas de Informação,mmendonca@inf.ufsm.br
PEDRO GRASSI XAVIER,201040106,25/08/2014,Curso de Engenharia de Computação,pedro_gx@hotmail.com
ROMEU ISAAC CASAROTTO,201420058,15/08/2014,Bacharelado em Sistemas de Informação,casarotto@live.com

```

Figura 5.2 - Formado do arquivo.csv importado pelo aplicativo.

Ao executar a aplicação pela primeira vez, é criado um diretório dentro da pasta nativa de diretórios do Android, com o nome de BlueTApp, onde posteriormente serão armazenados arquivos referentes ao funcionamento da mesma. Ainda na primeira execução o professor fica limitado a inserir uma nova turma com mostrado na figura 5.3.

Ao clicar no botão “Inserir Turma” um formulário, como mostrado na figura 5.4, para preenchimento com os dados referentes à turma é exibido. É nesse formulário que, além de indicar o nome da turma e a carga horária da disciplina que será lecionada, o professor indicará também o arquivo .csv proveniente do portal do professor. Ao clicar na caixa de texto “nome do arquivo .csv”, caso o dispositivo móvel do professor contenha algum gerenciador de arquivos instalado, ele será invocado automaticamente, de forma que o professor possa navegar entre os diretórios do seu dispositivo móvel à procura do arquivo. Caso contrário, o professor deve digitar o nome do arquivo, que deve estar na pasta “download” nativa do Android.

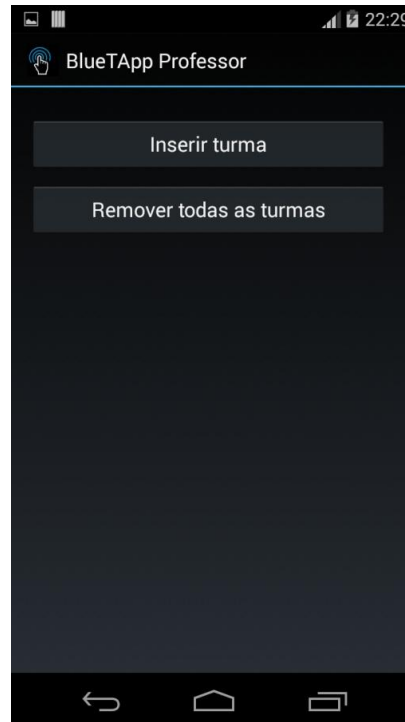


Figura 5.3 - Tela inicial do aplicativo BlueTApp Professor.

A parte do formulário onde os dias letivos são informados juntamente com o horários das aulas servirão para trabalhos futuros e serão explicados mais a frente.



Figura 5.4 - Formulário para inserção de novas turmas.

Após preencher os campos do formulário e clicar no botão “Inserir Turma”, a turma é criada e passa a ser exibida na tela inicial da aplicação como mostrado na figura 5.5. Uma mensagem também é exibida ao professor confirmando que a operação foi realizada com sucesso.

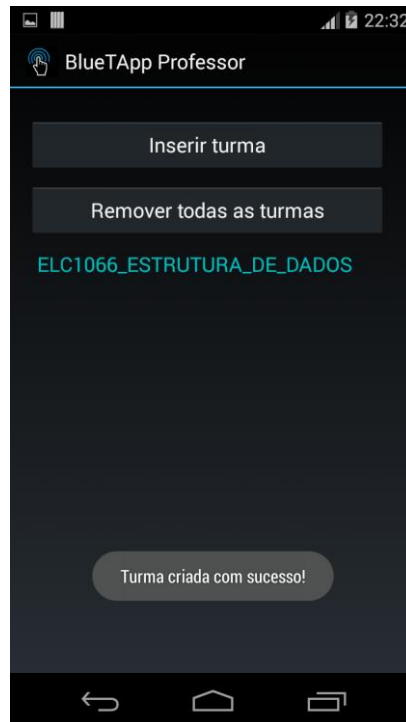


Figura 5.5 - Tela inicial do aplicativo após a inserção de uma turma.

Agora, ao executar a aplicação, as turmas anteriormente adicionadas são listadas e o professor não fica mais limitado a inserir turmas. Ao efetuar um clique curto sobre alguma das turmas existentes na lista, as informações, que foram importadas do arquivo .csv, sobre cada aluno da turma, são exibidas como mostrado na figura 5.6. Além disso, o número e a porcentagem de faltas de cada aluno também são exibidos. A porcentagem de faltas é calculada através da seguinte fórmula:

$$Pfaltas = \frac{Nfaltas * 100}{Chr}$$

Onde:

- *Pfaltas* é a porcentagem de faltas de cada aluno;
- *Nfaltas* é o número de faltas de cada aluno;

- *Chr* é a carga horária total da disciplina.

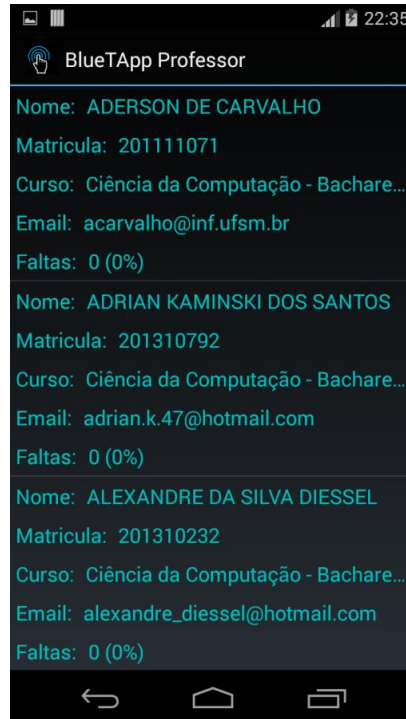


Figura 5.6 - Tela exibida após um clique curto sobre alguma turmas.

5.2.1 Menu de opções

Ao efetuar um clique longo sobre alguma das turmas um menu com diversas opções é exibido, como mostrado na figura 5.7. A seguir é explicado detalhadamente como o aplicativo se comporta a escolha de cada uma das opções disponíveis no menu.

- **Chamada por Bluetooth:** é a opção que realiza a chamada de forma automática. Ao escolher essa opção uma nova tela é exibida, onde o professor, antes de realizar a chamada deve indicar o assunto da aula e a carga horária referente à chamada que será realizada, como mostrado na figura 5.8. Após a confirmação dos dados o aplicativo verifica se o adaptador Bluetooth do dispositivo está ligado, caso não esteja, uma solicitação de permissão é exibida. Após o professor aceitar a solicitação, o aplicativo ativa o adaptador Bluetooth e inicia a busca pelos sinais dos dispositivos móveis dos alunos. Todos dispositivos encontrados são listados para o professor e ao final da busca o número de dispositivos encontrados, o número de alunos presentes e o tempo total de busca são informados, como mostrado na figura 5.9.

Após o término da busca por dispositivos a frequência é atribuída de forma automática para cada aluno.

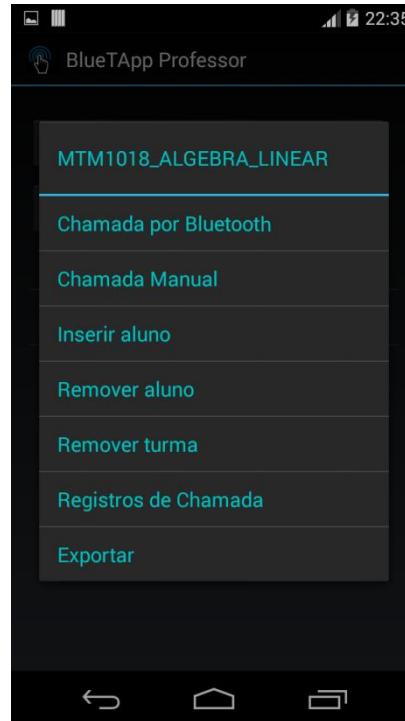


Figura 5.7 - Menu de opções exibido após um clique longo sobre alguma turma.

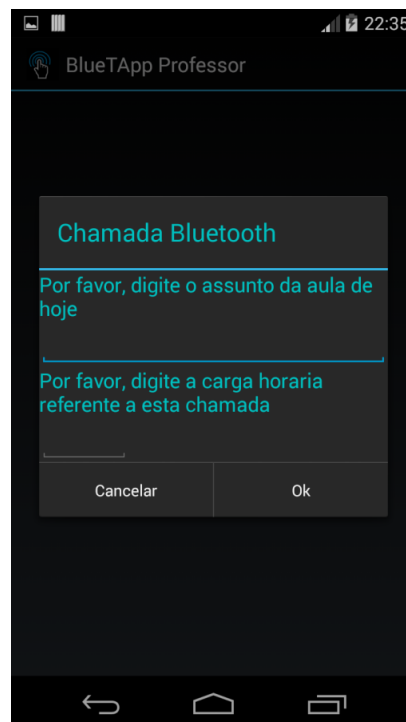


Figura 5.8 - Tela exibida antes da realização de uma chamada.

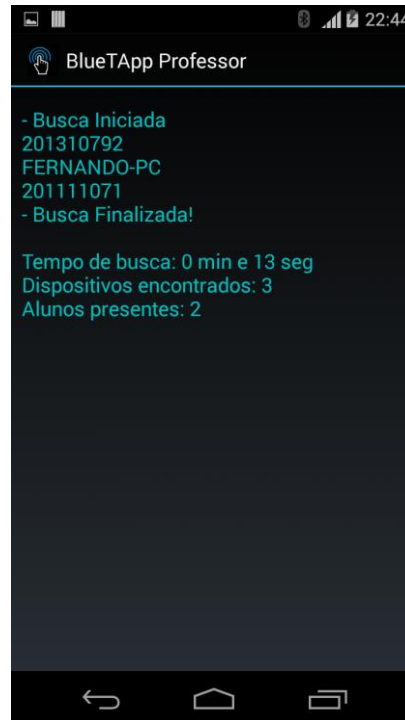


Figura 5.9 - Tela exibida ao término da busca por dispositivos.

- **Chamada Manual:** essa opção permite ao professor realizar a chamada de forma manual. Assim como na chamada por Bluetooth, o professor precisa indicar o assunto da aula e a carga horária referente à chamada que será realizada. Após a confirmação dos dados uma lista com o nome de cada aluno da turma é exibida juntamente com um checkbox, permitindo ao professor marcar quais alunos estão presentes. A chamada manual é concluída com o clique sobre o botão “Salvar”. A figura 5.10 mostra a tela de seleção dos alunos apresentada pela opção “Chamada Manual”.

- **Inserir Aluno:** essa opção permite ao professor inserir um aluno na turma. Quando escolhida, um formulário para preenchimento com as informações referentes ao aluno é exibido, como mostrado na figura 5.11. Caso algum dos campos não seja preenchido, um alerta é gerado. Isso também ocorre caso o professor tente inserir um aluno já existente na turma. A operação de inserção é concluída com o clique sobre o botão “Inserir Aluno”. Essa opção é de grande utilidade para o período pós-ajustes de matrícula, onde o professor pode inserir os novos alunos na turma sem a necessidade de recarregar todo o arquivo .csv.



Figura 5.10 - "Chamada Manual".

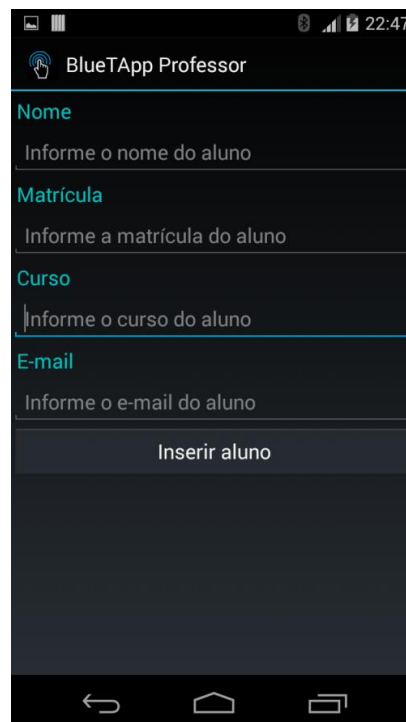


Figura 5.11 - Formulário para inserção de um aluno.

- **Remover Aluno:** permite que o professor remova um aluno da turma. A busca pelo aluno é realizada através da sua matrícula, que é única. Um alerta é gerado quando a matrícula informada não é encontrada. A figura 5.12 apresenta o formulário de remoção de um aluno.

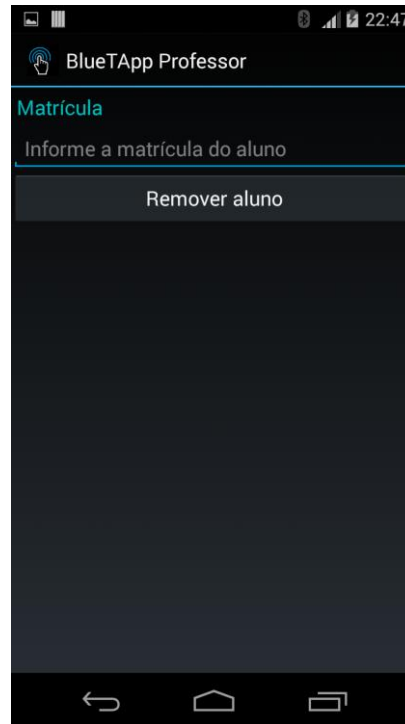


Figura 5.12 - "Remover Aluno".

- **Excluir Turma:** permite ao professor excluir uma turma. Ao clicar nessa opção uma caixa de dialogo é exibida solicitando a confirmação da operação. Dessa forma, evita-se que uma turma seja excluída por um erro de clique. A figura 5.13 apresenta a tela exibida ao escolher essa opção.

- **Registros de Chamada:** é a opção que exhibe uma espécie de “log” das chamadas realizadas. Ao clicar nessa opção é exibida uma tela que mostra os dias em que as chamadas foram realizadas, o número de alunos que estavam presentes naquela aula e o assunto da aula lecionada pelo professor. A figura 5.14 mostra a tela referente à opção “Registros de Chamada”.

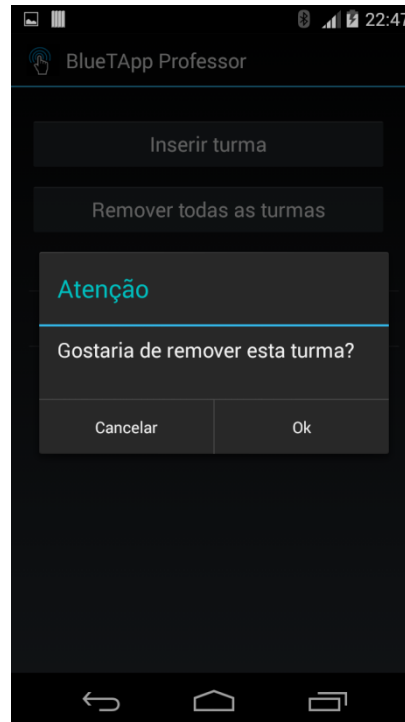


Figura 5.13 - Confirmação da operação de exclusão de uma turma.



Figura 5.14 - Registros de chamada.

- **Exportar:** essa opção permite que o professor exporte os dados referente à frequência acadêmica de cada aluno para um arquivo no formato .csv. O arquivo gerado leva o nome da turma seguido da palavra “Frequencias” e é armazenado na pasta BlueTApp criada pelo aplicativo. O formato do arquivo é apresentado na figura 5.15. Como mostrado na figura é possível notar que o arquivo é composto de duas colunas, uma referente à matrícula do aluno e a outra ao vetor de frequências do mesmo.

```

matricula,frequencia
201111071,APPAAPAAAPAAAPPPAPPAPAAAPPAPAAAPPAPAAAPPPPPAAPPAAAPAAAPAAAPPPAPPAPAAAPPAPAAAPPAPAA
201310792,PPAAAPPPPPPPAAAPPPAAAPAAAPAAAPPPAPAPPPPPAAAPPPPPPPAAAPPAAPPPAAAPPPAAAPAAAPPPA
201310232,AAAPPPAAAPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPAAAPAAAPPPAPPAPAAAPPAPAAAPPAPAA
201130305,AAAPAAAPPPAPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201410154,AAAPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201310316,APAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201212732,AAAPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201311093,PPPPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201210149,APAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201030216,APAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201312353,PPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201310739,PPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA
201230058,PPPPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAAAPPPAA

```

Figura 5.15 - Formato do arquivo.csv exportado pelo aplicativo.

O diagrama de casos de uso esclarece as opções disponíveis no BlueTApp professor. O diagrama é apresentado na figura 5.16.

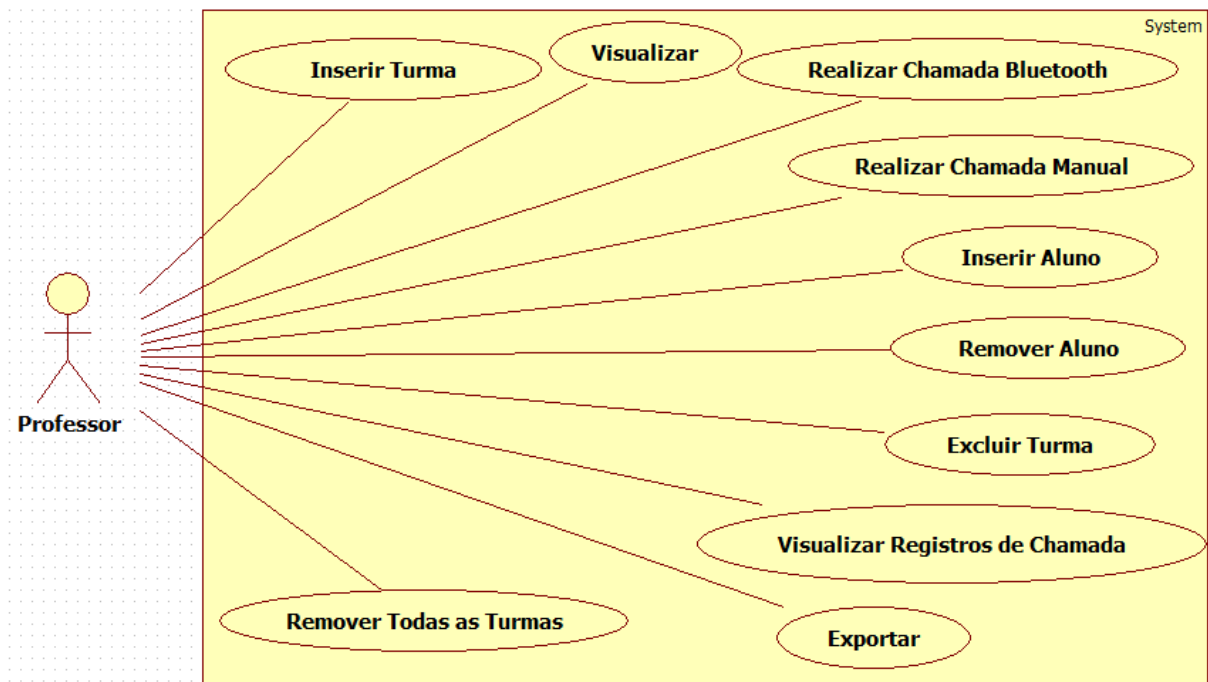


Figura 5.16 - Diagrama de casos de uso do aplicativo BlueTApp Professor.

5.2.2 Persistência de Dados

A persistência de dados na aplicação foi desenvolvida utilizando o banco de dados nativo da plataforma Android, o SQLite. Os dados referentes à aplicação foram armazenados em duas tabelas. As tabelas 5.1 e 5.2 apresentam o formato das tabelas e um exemplo de entrada de dados para cada uma.

Id	Nome	Matricula	Frequência	Bluetooth Address	Curso	Email
1	Fulano	123456	PPPPAA	0C:DF:A4:C0:9E:ED	Ciência da Computação	fulano@gmail.com

Tabela 5.1 - Tabela turma.

A tabela apresentada acima leva o nome da turma e é criada no momento em que o professor adiciona uma nova turma. A coluna Id é preenchida automaticamente pelo banco de dados de forma auto incremental a partir de cada aluno importado do arquivo .csv. As colunas Nome, Matricula, Curso e Email são preenchidas a partir das informações importadas do arquivo .csv.

A coluna Frequência permanece vazia até o momento em que uma chamada é realizada. Essa coluna é representada por uma String onde cada caracter equivale à uma hora aula e é incrementada cada vez que uma chamada é realizada. A letra “P” representa uma presença enquanto a letra “A” representa uma ausência. O número de caracteres adicionado a String é determinado pelo professor no momento anterior a realização da chamada quando ele indica a carga horária da mesma.

Já a coluna Bluetooth Address só é preenchida no momento em que uma chamada por Bluetooth é efetuada. Ao encontrar os sinais Bluetooth dos dispositivos móveis dos alunos, o aplicativo compara o nome do dispositivo encontrado, que no caso é a matricula do aluno, com a matrícula presente no banco de dados, caso as duas sejam iguais, a coluna Bluetooth Address, associada ao aluno, é preenchida com esse endereço. Assim, nas próximas vezes que a chamada por Bluetooth for realizada esse aluno só ganhará presença caso o Bluetooth Address encontrado seja o mesmo ao anteriormente associado a ele. Isso evita que um aluno se passe por outro através da troca do nome do dispositivo para a matrícula de outro aluno.

Id	Data	Presenças	Ementa	Posição Inicial	Posição Final
1	15/11/2014	28	Orientação a Objetos - Classes	2	3

Tabela 5.2 - Tabela registro.

A tabela apresentada acima funciona como um registro de chamada. Ela é criada no momento em que uma chamada é realizada e tem como nome o nome da turma seguido da palavra “Registro”. A coluna data é referente à data em que a chamada foi realizada, a coluna presenças indica o número de alunos que estavam presentes quando esta chamada foi realizada e a coluna ementa é referente ao assunto da aula, indicado pelo professor no momento anterior à realização da chamada.

As colunas posição inicial e final indicam a posição, da chamada realizada, no vetor de frequências. A figura 5.17 esclarece a função dessas colunas utilizando como exemplo a entrada de dados da tabela 5.2.

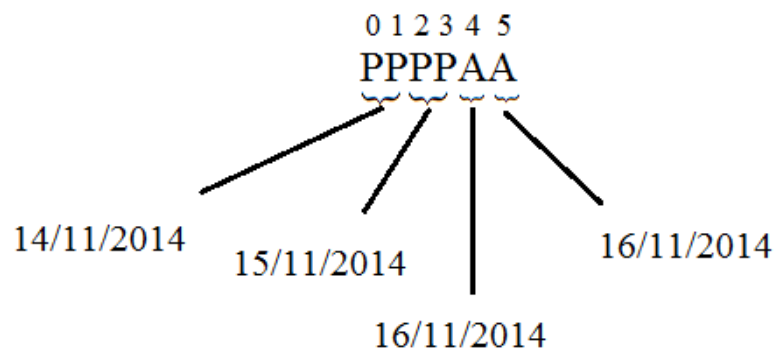


Figura 5.17 - Exemplo de utilização da coluna "Frequencia" no aplicativo.

Através dos registros de chamada é possível verificar a quantidade de chamadas realizadas pelo professor. Os registros de chamada são utilizados também quando o professor tenta realizar mais de uma chamada por aula. O aplicativo verifica que uma chamada já foi realizada no corrente dia e oferece ao professor as opções de realizar uma nova chamada ou sobrescrever a anterior, como mostrado na figura 5.18.

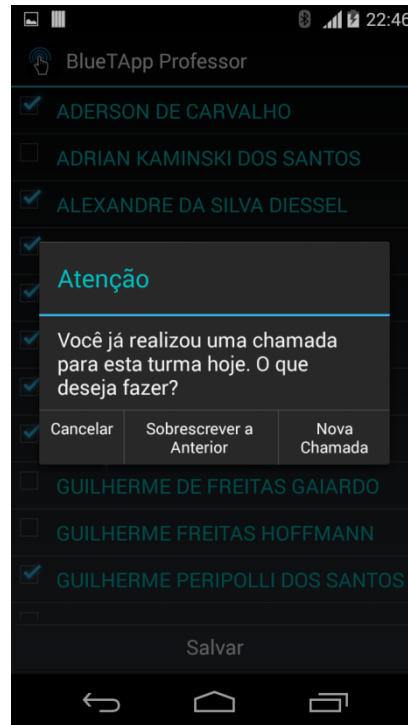


Figura 5.18 - Opções de chamada.

Ao escolher “Nova Chamada” um novo registro é adicionado à tabela de registros e o vetor de frequências é atualizado com novas informações. Se a opção escolhida for “Sobrescrever a Anterior”, a última chamada realizada é editada, assim como o ultimo registro de chamada de forma que os caracteres referentes a ultima chamada no vetor de frequências são atualizados.

Quando o professor realiza mais de uma chamada por aula, espera-se que a carga horária da aula, informada no momento anterior a realização da chamada, seja dividida pelo número de chamadas. Por exemplo, se a aula tem duração de duas horas e o professor deseja realizar duas chamadas, uma no início da aula e outro no final, a carga horária referente a cada chamada deve ser de uma hora. Dessa forma, ao final do semestre o tamanho da String de frequências será relativo ao número de horas aula.

5.2.4 Diagrama de Sequência

A seguir é apresentado o diagrama de sequência relativo ao funcionamento completo do aplicativo.

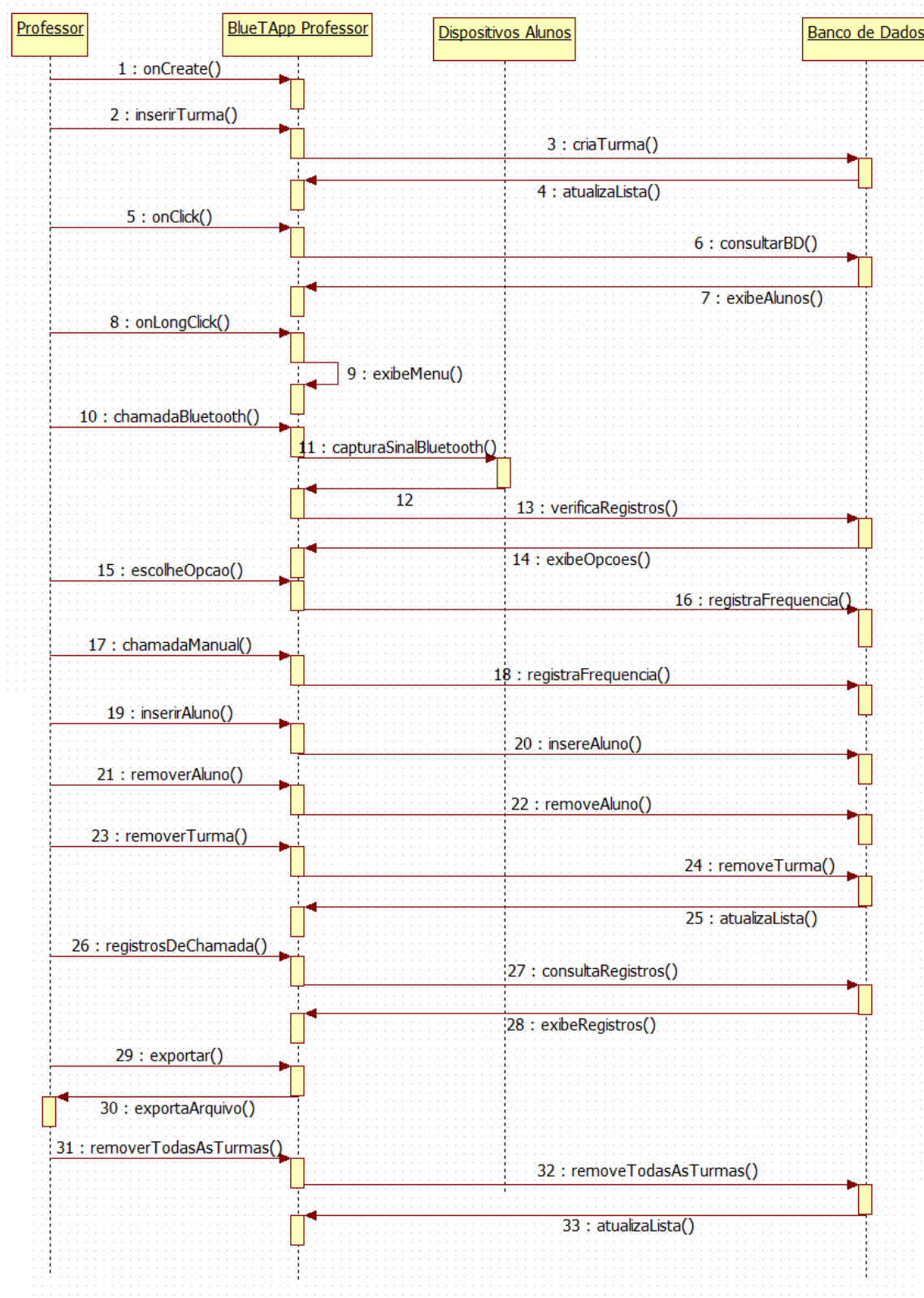


Figura 5 19 - Diagrama de seqüência.

6. TESTES E VALIDAÇÃO DO APLICATIVO

O primeiro desafio deste trabalho era escolher qual tipo de comunicação, realizada entre o dispositivo móvel do professor e os dispositivos móveis dos alunos, era mais vantajosa em termos de desempenho e velocidade para o aplicativo. Tal comunicação poderia se dar de duas formas, com pareamento e abertura de um canal para troca de mensagens ou sem pareamento e sem troca de mensagens. Para descobrir qual tipo de comunicação era mais viável, foram realizados testes em ambiente controlado, com quatro dispositivos móveis, um fazendo o papel do professor e três atuando como alunos.

Este primeiro teste se deu da seguinte forma, após a descoberta dos sinais Bluetooth dos dispositivos que estavam atuando como alunos, o aparelho do professor solicitou o pareamento com tais dispositivos. Através deste teste foi possível verificar que o pareamento não pode ser efetuado de forma simultânea com todos os dispositivos, ou seja, o aparelho do professor precisa criar uma fila de pareamento, na ordem dos sinais que foram encontrados, e enviar solicitações de pareamento uma a uma. Isso ocorre porque, ao solicitar o pareamento, uma chave única é compartilhada entre os dois dispositivos que estão tentando efetuar a operação, como a operação de pareamento é bloqueante, se mais algum dispositivo tentar efetuar o pareamento antes do primeiro ter sido completado, a segunda chave é perdida e o procedimento não consegue ser completado.

O tempo de pareamento com até três dispositivos é apresentado na figura 6.1. Os dados apresentados na figura foram calculados através de três testes com cada número de dispositivos e então calculada a média aritmética dos tempos. Por meio da figura é possível notar que conforme aumenta o número de dispositivos envolvidos no processo, o tempo para a realização da descoberta de sinal mais o pareamento também aumenta, se tornando inviável para este trabalho.

Através deste teste foi possível verificar que efetuando o pareamento entre os dispositivos, os objetivos proposto neste projeto estariam comprometidos. Optou-se então por abrir mão da troca de mensagens entre os dispositivos e realizar a automação do registro da frequência acadêmica sem realizar o pareamento.



Figura 6.1 - Tempo para descoberta e pareamento.

Após a escolha do tipo de comunicação, um segundo teste foi realizado para medir o tempo de descoberta dos sinais Bluetooth sem a realização do pareamento. Este teste foi realizado em ambiente controlado e também em sala de aula.

Para tornar este teste possível, um tempo de aula de Fundamentos de Banco de Dados lecionada pelo professor João Carlos Damasceno de Lima, da Universidade Federal de Santa Maria, foi cedido. Neste teste foi solicitado aos alunos que instalassem a versão aluno do aplicativo em seus dispositivos móveis. Aos alunos que estavam utilizando dispositivos de outras plataformas que não o Android, foi solicitado que efetuassem de maneira manual os passos realizados pelo aplicativo versão aluno. No dia dos testes estavam presentes somente nove alunos, dessa maneira ficamos limitados a testar o aplicativo com no máximo nove dispositivos. Nosso objetivo era medir o tempo de busca dos sinais Bluetooth conforme a quantidade de dispositivos envolvidos no processo. Para tornar isso possível, várias chamadas por Bluetooth foram efetuadas. Na primeira chamada foram utilizados somente quatro dispositivos. Após cada teste o número de dispositivos envolvidos era acrescido de um até

chegarmos ao nosso limite de nove dispositivos. A figura 6.2 mostra o tempo de descoberta dos sinais Bluetooth dos dispositivos móveis dos alunos.

Este período em sala de aula serviu também para validar a eficiência do aplicativo, que em 100% dos casos, conseguiu capturar os sinais Bluetooth de todos os alunos envolvidos no teste e registrar a frequência de forma automática, alcançando assim o os objetivos propostos neste trabalho.

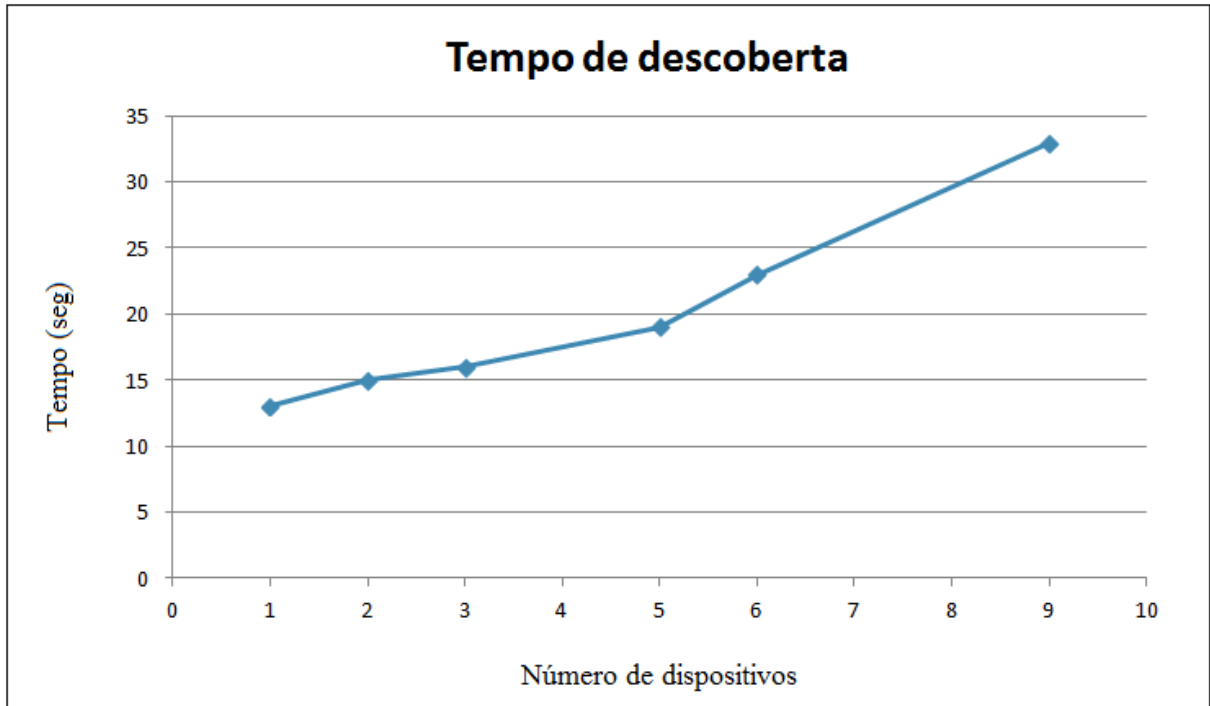


Figura 6.2 - Tempo para descoberta dos sinais Bluetooth.

7. CONCLUSÕES E TRABALHOS FUTUROS

Este TG apresentou um aplicativo móvel que além da automação e gerenciamento do registro de frequência acadêmica serve também como uma ferramenta de consulta para os professores, que podem verificar a situação dos alunos ou das aulas já ministradas. Além disso, um formato de arquivo para futuras interpretações dos dados obtidos pelo aplicativo foi proposto.

No futuro, espera-se aumentar as funcionalidades do aplicativo, começando pela execução do mesmo de forma automática. Isso seria possível a partir das informações obtidas no formulário de inserção de uma nova turma (dias letivos e horário das aulas). Esta funcionalidade, porém, não foi desenvolvida neste trabalho uma vez que a aplicação necessitaria rodar em segundo plano, o que acarretaria em um consumo significativo de bateria e um gerenciamento de consumo se faria necessário. Além disso, esta funcionalidade estaria fugindo dos objetivos propostos, que tem foco em desempenho e velocidade.

Outra possibilidade de trabalho futuro é permitir que os alunos utilizem mais de um dispositivo móvel para participar do processo de registro da frequência acadêmica e além disso, expandir a fronteira do aplicativo para os alunos, de tal maneira que possam consultar sua situação de frequência de forma independente do professor.

Espera-se que futuramente uma parceria com a Universidade Federal de Santa Maria seja alcançada, de forma que o aplicativo seja adotado pela instituição para a realização do registro de frequência acadêmica e o arquivo proposto utilizado para a interpretação dos dados.

Através dos testes realizados foi possível validar a eficiência do aplicativo e alcançar os objetivos propostos. Espera-se que no futuro, este aplicativo ganhe seu espaço dentro das instituições de ensino facilitando a tarefa de realização da chamada tanto para professores quanto para alunos.

REFERÊNCIAS

- ANDROID DEVELOPERS. **Bluetooth**. 2014. Disponível em <<http://developer.android.com/guide/topics/connectivity/bluetooth.html>>. Acesso em 01 out. 2014.
- ANDROIDZ. **Guia para Iniciantes**. 2013. Disponível em <<http://www.androidz.com.br/portal/o-que-e-o-sdk-do-android.html>>. Acesso em 01 out. 2014.
- APPLE SUPPORT. **Wireless input devices: Bluetooth frequently asked questions**. 2014. Disponível em <<http://support.apple.com/kb/ht3887?>>. Acesso em 19 set. 2014.
- BISDIKIAN, C. An overview of the Bluetooth wireless technology. **IEEE Commun Mag**, v. 22109, n. June, 2001.
- BLUETOOTH. **History of the Bluetooth Special Interest Group**. 2014. Disponível em <<http://www.bluetooth.com/Pages/History-of-Bluetooth.aspx>>. Acesso em 19 set. 2014.
- BRASIL. Lei n. 9.394, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional. **Diário Oficial [da] República Federativa do Brasil**, Brasília, DF, 23 dez. 1996. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/19394.htm>. Acesso em: 17 nov. 2014.
- BURNETTE, E. **Hello, Android: Introducing Google's Mobile Development Platform**. Texas: The Pragmatic Bookshelf, 2009. p. 247
- CHAMON, J. P. M. et al. **Registro Ubíquo de Controle Acadêmico: Localização em Ambiente Interno Utilizando Ciclo de Trabalho Dinâmico**. In: XXXIV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 2014, Brasília/DF.
- CIDRAL, B.; Afinal, o que é Android?. **TechTudo**, 2011. Disponível em <<http://www.techtudo.com.br/artigos/noticia/2011/01/afinal-o-que-e-android.html>>. Acesso em: 27 set. 2014.
- DIMARZIO, J. **Android, a programmers guide**. [s.l.] McGraw-Hill, 2008. p. 337
- GOMES, R.; FERNANDES, J. **Sistema Operacional Android**. p. 31, 2012.
- INTERNATIONAL DATA COPORATION. **Releases**. 2013. Disponível em: <<http://br.idclatin.com/releases/news.aspx?id=1547>>. Acesso em 29 set. 2014.

JUNIOR, J. B.; **Manual do Usuário de Android**. 2014. Disponível em <<http://books.google.com.br>>. Acesso em 25 set. 2014.

JÚNIOR, M.; CASTRO, R. DE O. **Um estudo de caso da plataforma Android com Interfaces Adaptativas**. p. 1–16, 2011.

MARTINCOSKI, D. H. **Sistema para Telemetria de Eletrocardiograma Utilizando Tecnologia Bluetooth**. 2003. 113 f. Dissertação (Mestrado em Eng. Elétrica)-Universidade Federal de Santa Catarina, 2003.

MARTINS, R. **Desenvolvimento de Aplicativo para Smartphone com a Plataforma Android**. 2009. Monografia (Graduação em Engenharia da Computação)-Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2009.

MEIER, R. **Android™ Application Development**. [s.l.] John Wiley & Sons, Inc., 2009. p. 434.

OLIVEIRA, A. X. et al. **Projeto Tecnologia Wireless – Bluetooth**. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, 2005, Resende. Disponível em <<http://www.aedb.br/seget/artigos.php?pag=8>>. Acesso em: 12 ago. 2014.

PASSOS, B. **Sistema Bluetooth para Controle de Acessórios Veiculares Utilizando Smartphone com Android**, 2011. Monografia (Graduação em Engenharia da Computação)-Centro Universitário de Brasília, Brasília, 2011.

PEREIRA, L. C. O.; SILVA, M. L. **Android para Desenvolvedores**. Rio de Janeiro: Brasport Livros e Multimídia Ltda, 2009.

TAKASHI, C. K. **Arquitetura de Mobilidade Bluetooth**. 2003. Dissertação (Mestrado em Ciência da Computação)-Universidade Federal de Pernambuco, Recife, 2003.

ZANON, T.; FELDEN, E. DOS S. **Uso e Aplicação da Tecnologia Bluetooth**. ctec.unicruz.edu.br, p. 0-3, [s,d].

WEISER, M.; **The Computer for the 21st Century**. Scientific american, (1991). Disponível em <http://wiki.daimi.au.dk/pca/_files/weiser-orig.pdf>. Acesso em 20 de out. 2014.