

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**JULIA DE ROCHI**

**PROTÓTIPO DE APLICATIVO ANDROID PARA IDENTIFICAÇÃO DE  
ANIMAIS UTILIZANDO A TECNOLOGIA NFC**

**CRICIÚMA  
2015**

**JULIA DE ROCHI**

**PROTÓTIPO DE APLICATIVO ANDROID PARA IDENTIFICAÇÃO DE  
ANIMAIS UTILIZANDO A TECNOLOGIA NFC**

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel no curso de Ciência da Computação, da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. Esp. Sergio Coral

Coorientador: Prof. Esp. William Bertan da  
Silva

**CRICIÚMA**

**2015**

**JULIA DE ROCHI**

**PROTÓTIPO DE APLICATIVO ANDROID PARA IDENTIFICAÇÃO DE ANIMAIS  
UTILIZANDO A TECNOLOGIA NFC**

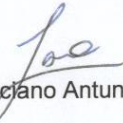
Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Automação.

Criciúma, 24 de novembro de 2015.

**BANCA EXAMINADORA**

  
Prof. Esp. Sérgio Corral - Especialista - (UNESC) – Orientador

  
Prof. William Bertan da Silva- Especialista - (UNESC) – Coorientador

  
Prof. Luciano Antunes - Mestre - (UNESC)

  
Prof. Gilberto Vieira da Silva - Especialista - (UNESC)

**Dedico a todos aqueles que de alguma forma estiveram e estão próximos de mim, fazendo esta vida valer cada vez mais a pena.**

## **AGRADECIMENTOS**

Agradeço primeiramente a minha família: minha mãe Gisele, minha Vó Felícia, meu irmão Luciano Octavio, meu pai de coração Natale e todos os demais membros que mesmo não citados a cima sempre estiveram presente e sempre me deram forças para continuar.

Agradeço ao meu namorado Jackson, que esse foi o que mais teve que me suportar durante o desenvolvimento deste projeto tendo uma grande compreensão do meu nervosismo, agradeço por ser essa pessoa calma, que está sempre disponível para me ajudar e me escutar.

Agradeço aos meus amigos, que estiveram ao meu lado durante esta caminhada, me incentivando e encorajando sempre.

Agradeço ao meu orientador Sergio Coral que me orientou de um modo que sempre estive à vontade para construir o meu trabalho, que me auxiliou na ideia inicial do projeto, trazendo o seu conhecimento e experiência para enriquecer a mesma.

Agradeço também ao meu Coorientador Willian, que foi fundamental, estando sempre disponível para responder as dúvidas que surgiam durante o desenvolvimento do projeto.

Aos demais docentes do curso de Ciência da Computação, que ao decorrer destes quatro anos e meio, compartilharam seus conhecimentos da melhor forma possível, agregando muito valor na minha vida acadêmica e pessoal.

Agradeço ao mundo por estar girando e sempre mudar as coisas, por nunca fazê-las serem da mesma forma, pois assim não teríamos o que questionar, pesquisar, o que descobrir e o que fazer, pois através disto consigo realmente crescer a cada dia.

**“A compaixão pelos animais está intimamente ligada a bondade de caráter, e quem é cruel com os animais não pode ser um bom homem. ”**

**Arthur Schopenhauer**

## RESUMO

Hoje em dia, o índice de abandono de animais é muito alto. As ONG's existentes hoje não conseguem cuidar e socorrer todos os animais abandonados, principalmente porque a maioria dessas instituições vive apenas de doações, não recebendo auxílio das autoridades. Muitos desses animais encontrados pelas ONG's podem ter sido perdidos e não abandonados, porém como o mesmo não pode ser identificado, ele acaba voltando para a rua caso não seja adotado no tempo em que permanece na instituição. Diante deste problema, a identificação do animal torna-se fundamental, pois facilita a localização do proprietário e obtém-se informações referentes ao animal. Um método indolor e permanente para a identificação de animais é o microchip, método já existente que funciona com a tecnologia Radio Frequency Identification (RFID), cujo chip já pode ser implantado em diversos locais. Entretanto, muitos estabelecimentos não possuem os leitores deste microchip para poder identificar o animal devido ao alto custo do mesmo. Uma tecnologia que vem se destacando cada vez mais no mercado é o Near Field Communication (NFC), isso por ser uma tecnologia simples e intuitiva ao usuário, permitindo a transmissão de dados apenas pelo contato do celular com a tag tal qual se deseja obter informações. Esta tecnologia está cada vez mais presente em dispositivos móveis. Diante deste problema e das tecnologias estudadas para poder resolver o mesmo, foi desenvolvido um protótipo de aplicativo na plataforma Android capaz de identificar tags NFC e cadastrar dados referentes ao animal e seu proprietário, contando com o auxílio de um servidor desenvolvido em Java que realiza a comunicação entre a aplicação e o banco de dados. A implementação da aplicação foi realizada através do ambiente de desenvolvimento Android Studio e os testes foram executados em um dispositivo LG G3. Os resultados obtidos foram satisfatórios, pois foi possível realizar a identificação da tag através do protótipo desenvolvido.

**Palavras-chave:** Identificação de animais. NFC. Android. Dispositivos moveis.

## ABSTRACT

Nowadays, the animals abandon levels are too high. The existing NGO's can't take care of and help all this abandoned animals, mostly because a lot of these organizations lives by donations, without support of the authorities. Some of these animals founded by the NGO's may be lost and not abandoned, but without the identification, he stay in the organization to adoption for a while and go back to the streets if he is not adopted. Facing this problem, the animal identification is primordial and helps the proprietary location, also giving information about the animal. A painless and permanent animal identification method is the microchip that works with the Radio Frequency Identification (RFID) technology, which is applicable in variable locations. However, some shops do not have the readers of the RFID microchip, because the reader has a high price. A technology that is growing is the Near Field Communication (NFC), because is a simple and interactive technology for the user, allowing data transfer by the contact of the smartphone with the tag that contain the information. That technology is becoming more present in the mobile devices. Facing this problem and the studied technologies, a prototype application was developed in the Android platform with the capabilities of identify NFC tags and write data about the animal and the proprietary, including a Java developed server that communicate the application with the database. The development was made in the integrated development environment Android Studio and the tests was made on a LG G3. The results are satisfactory, because was possible to identify the NFC tags with the developed prototype.

**Palavras-chave:** Animal Identification. NFC. Android. Mobile devices.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Aplicação Tag no animal .....	22
Figura 2 – Sistema RFID e seus componentes .....	24
Figura 3 – Derivação do Near Field Communication .....	25
Figura 4 – Modos de comunicação P2P, leitura/escritura e emulação de cartões ....	28
Figura 5 – Estrutura de uma <i>tag</i> NFC .....	30
Figura 6 – Arquitetura Android .....	33
Figura 7 – Ciclo de vida de uma Activity Android .....	38
Figura 8 – Versões do Android .....	39
Figura 9 – Web Service básico.....	43
Figura 10 – Comunicação Servlet e com três diferentes clientes.....	44
Figura 11 – Ciclo de vida de um Servlet.....	45
Figura 12 – Utilitário de Configuração SDK Manager.....	52
Figura 13 – Logcat Android Studio .....	52
Figura 14 – Equipamento utilizado para avaliação.....	53
Figura 15 – Modelagem do banco de dados .....	54
Figura 16 – Diagrama de casos de usos .....	55
Figura 17 – Estrutura de classes do Servlet.....	56
Figura 18 – Função para conversão de dados para o formato JSON .....	57
Figura 19 – Função de comunicação entre servidor e banco de dados .....	58
Figura 20 – Configuração do manifesto .....	59
Figura 21 – Tela de login.....	60
Figura 22 – Função LoginProprietario da classe AcessoServlet .....	61
Figura 23 – Classe AcessoServletAsync.....	62
Figura 24 – Versão SDK .....	62
Figura 25 – Configuração do filtro NFC.....	63
Figura 26 – Verificação se o NFC está ativo no dispositivo .....	64
Figura 27 – Função VerificaNFCAtivo .....	64
Figura 28 – Método enableForegroundDispatch .....	65
Figura 29 – Função para verificar se o NFC está ativo .....	65
Figura 30 – Método de reconhecimento da <i>tag</i> .....	66
Figura 31 – Método para realizar a gravação de dados na <i>tag</i> .....	67
Figura 32 – Método de conexão para gravação da <i>tag</i> .....	68

Figura 33 – Tela de início do protótipo .....	69
Figura 34 – Tela de cadastro de proprietário .....	70
Figura 35 – Tela de manipulação de dados do animal .....	70
Figura 36 – Tela cadastro do animal .....	71
Figura 37 – Calendário para selecionar a data.....	72
Figura 38 – Spinner com as raças de animais .....	72
Figura 39 – Dados do proprietário e do animal identificado .....	73
Figura 40 – Dados do proprietário e do animal identificado .....	74

## LISTA DE TABELAS

Tabela 1 – As cinco liberdades do bem-estar animal.....	19
Tabela 2 – Possíveis componentes de uma aplicação Android .....	37

## LISTA DE ABREVIATURAS E SIGLAS

ABINPET	Associação Brasileira da Indústria de Produtos para Animais de Estimação
API	Application Programming Interfaces
ART	Android Runtime
CPU	Central Processing Unit
EMCA	Computer Manufacturers Association
FAWC	Farm Animal Welfare Council
GPS	Global Positioning System
HTTP	HyperText Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ETSI	European Telecommunications Standards Institute
NDEF	Near Field Communication Data Exchange Format
NFC	Near Field Communication
NFCIP-1	Near Field Communication Interface and Protocol 1
OASIS	Organization for the Advancement of Structures Information Standards
OHA	Open Handset Alliance
P2P	Peer-To-Peer
RFID	Radio Frequency Identification
SAML	Security Assertion Markup Language
SDK	Android Software Development Kit
SOAP	Simple Object Access Protocol
SWP	Single Wire Protocol
UDDI	Universal Description Discovery and Integration
W3C	World Wide Web Consortium
WIFI	Wireless Fidelity
WSDL	Web Service Description Language
XML	Extensible Markup Language

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 OBJETIVO GERAL .....	15
1.2 OBJETIVOS ESPECÍFICOS .....	15
1.3 JUSTIFICATIVA .....	16
1.4 ESTRUTURA DO TRABALHO .....	17
<b>2 ABANDONO DE ANIMAIS</b> .....	<b>19</b>
2.1 GASTOS MÉDIOS COM ANIMAIS DE ESTIMAÇÕES .....	20
2.2 TIPOS DE IDENTIFICAÇÃO DE ANIMAIS .....	21
<b>2.2.1 Microchip</b> .....	<b>21</b>
<b>3 COMUNICAÇÃO DE DADOS ENTRE DISPOSITIVOS MÓVEIS</b> .....	<b>23</b>
3.1 RADIO FREQUENCY IDENTIFICATION .....	23
3.2 NEAR FIELD COMMUNICATION .....	25
<b>4 DISPOSITIVOS MÓVEIS</b> .....	<b>31</b>
4.1 SISTEMA OPERACIONAL ANDROID .....	32
<b>4.1.1 Plataforma Android</b> .....	<b>33</b>
4.1.1.1 Camada de aplicações .....	34
4.1.1.2 Camada do framework .....	34
4.1.1.3 Camada do framework .....	35
4.1.1.4 Camada do ambiente de execução .....	36
4.1.1.5 Linux Kernel .....	36
<b>4.1.2 Camada de aplicações</b> .....	<b>37</b>
<b>4.1.3 Versões do Android</b> .....	<b>39</b>
4.2 DESENVOLVIMENTO DO NFC DENTRO DA PLATAFORMA ANDROID .....	41
<b>5 APLICAÇÕES WEB</b> .....	<b>42</b>
5.1 WEB SERVICE .....	42
<b>5.1.1 Padronização</b> .....	<b>43</b>
5.2 SERVLET .....	44
<b>6 TRABALHOS CORRELATOS</b> .....	<b>46</b>
6.1 DESENVOLVIMENTO DE UM MÓDULO DE ACIONAMENTO PARA ABERTURA DA FECHADURA DE UM QUARTO DE HOTEL BASEADO NA TECNOLOGIA NFC .....	46

6.2 ACESSIBILIDADE EM BIBLIOTECAS: UTILIZANDO NFC E SMARTPHONES PARA AUXILIAR DEFICIENTES VISUAIS NA IDENTIFICAÇÃO DE LIVROS .....	46
6.3 NFC TICKETING: A PROTOTYPE AND USABILITY TEST OF AN NFC-BASED VIRTUAL TICKETING APPLICATION .....	47
6.4 DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL BASADA EN LA TECNOLOGÍA NFC PARA ACCESO A INFORMACIÓN DE LAS PIEZAS DE ARTE DE UN MUSEO .....	48
6.5 NFC TAGS-BASED NOTIFICATION SYSTEM FOR MEDICAL APPOINTMENTS .....	48
<b>7 APLICATIVO ANDROID PARA IDENTIFICAÇÃO DE ANIMAIS UTILIZANDO A TECNOLOGIA NFC .....</b>	<b>50</b>
7.1 METODOLOGIA.....	50
7.2 ESTUDO DAS FERRAMENTAS UTILIZADAS .....	51
7.3 MODELAGEM DE DADOS .....	53
7.4 DESENVOLVIMENTO DO PROTÓTIPO .....	56
<b>7.4.1 Estrutura do servidor web .....</b>	<b>56</b>
<b>7.4.2 Aplicação Android.....</b>	<b>59</b>
<b>7.4.3 Funcionamento do sistema .....</b>	<b>69</b>
7.5 RESULTADOS OBTIDOS .....	74
<b>8 CONCLUSÃO .....</b>	<b>77</b>
<b>REFERÊNCIAS.....</b>	<b>79</b>

## 1 INTRODUÇÃO

Atualmente, o índice de abandono de animais é muito grande, e isto pode ser comprovado pelo grande número de animais que lotam os abrigos brasileiros, principalmente durante as férias (MADI, 2014). Segundo a Organização não governamental (ONG) SOS - Vira Lata 2014 sediada em Criciúma, a identificação é essencial para que se tenha um controle sobre os animais, facilitando assim o trabalho da ONG. Atualmente a ONG possui cerca de 150 animais em lares temporários e clínicas a espera de adoção, porém, segundo a ONG, o número de animais abandonados é muito maior, pois muitos deles estão ainda nas ruas e a cada dia que passa este número aumenta.

Com o constante aumento no número de animais, os gastos da ONG com animais acidentados ou doentes encontrados na rua é muito alto, ficando em torno de cinco mil reais por mês, sendo que este dinheiro não vem de nenhum órgão público. A arrecadação é feita por doações, venda de produtos relacionados a ONG e bazar de roupas usadas.

Ao implantar uma identificação nos animais, a ONG poderia identificar o dono do animal, saber se o animal possui todas as vacinas em dia, idade, dentre outros dados que auxiliariam em um maior controle sobre a situação. O número de animais doados mensalmente pela ONG varia entre 25 e 35 animais, sendo que muitas vezes estes animais adotados, acabam sendo devolvidos ou abandonados a rua novamente.

Atualmente algumas prefeituras e ONGs já fazem a identificação de animais através de *tags*, facilitando assim a identificação do animal, porém para identificar os dados desses animais a pessoa que o encontra deve levar o mesmo até um veterinário que tenha um leitor que identifique os dados cadastrados na *tag*. Esta tecnologia ainda não é muito difundida entre os veterinários por ser um produto de alto custo.

A identificação dos animais de estimação vem sendo praticada desde a pré-história, há cerca de 14.000 – 18.000 anos, tempos estes que o homem o terá domesticado. Inicialmente, a esta identificação tinha como propósito a diferenciação dos animais entre eles mesmos, mas com o passar do tempo foi utilizada também para o controle da população e do bem-estar de tais animais (DÓRIA, 2010).

No passado, foram utilizadas técnicas de marcação simples como cortes nas orelhas com diferentes localizações e aspectos, bem como a marcação por queimadura à fogo ou à azoto líquido, ainda utilizados em alguns países e regiões, porém limitadas a algumas espécies animais, como cavalos e bovinos (DÓRIA, 2010).

Posteriormente, evoluiu-se tais técnicas para tatuagens nas orelhas e nas bragadas (zona interna das coxas, sem pelos), seguindo o uso de coleiras, anilhas e brincos e, mais recentemente, o uso de identificadores eletrônicos (DÓRIA, 2010).

Ao realizar uma pesquisa sobre soluções para este problema, não foram encontradas aplicações similares, estando disponíveis apenas aplicativos que utilizam etiquetas com QRCode na coleira dos animais. Esta solução não impede que um animal seja abandonado, pois basta retirar a *tag* da coleira e o animal ficará novamente sem identificação, não resolvendo assim o problema de abandono do mesmo. Foram encontradas soluções que utilizam o Radio Frequency Identification (RFID) com *tags* de biovidro, porém o custo muitas vezes não é viável para um pet shop, sendo que muitos dos estabelecimentos que inserem o chip não possuem o leitor devido ao seu alto custo.

Visando este problema, foi desenvolvido um protótipo de uma aplicação em Android para disponibilizar esta identificação para qualquer usuário que possua um celular com a tecnologia Near Field Communication (NFC).

A *tag* inserida no animal é feita de biovidro e deve ser implantada abaixo da pele do animal, geralmente na região da coxa. Este tipo de *tag* é considerado permanente e muito importante para a identificação do animal (WSPA, 2008, tradução nossa).

A tecnologia NFC é um tipo de comunicação entre dispositivos de curto alcance, para transferir ou receber dados, bastando que dois celulares ou um celular e um *tag* com a tecnologia NFC se encostem para efetuar a transferência de dados. (FINKENZELLER, 2010).

O NFC é uma evolução do RFID patenteada nos anos 1970 por Charles Walton e desenvolvido pela empresa NXP Semiconductors (antigamente conhecida como Philips) em parceria com a Sony no ano de 2002. Nestes anos foram realizados ainda muitos testes e a tecnologia não se expandiu rapidamente, pois o mercado não estava preparado para desfrutar de todo seu potencial (LANGER; ROLAND, 2010, tradução nossa).

Em 2003, o padrão NFC foi aprovado pela European Computer

Manufacturers Association (EMCA), International Organization for Standardization (ISO) e International Electrotechnical Commission (IEC) (GRATTON, 2007, tradução nossa).

No final do processo de padronização, foi possível uma verdadeira compatibilidade e interoperabilidade entre dispositivos habilitados. Em 2004 a Nokia, Sony e Philips criaram um fórum para dirigir a futura implantação do NFC, o NFC Fórum (RIBEIRO, 2013).

O primeiro celular que utilizou a tecnologia NFC foi o Samsung SHW-A170K, um aparelho que pretendia ser um dos protagonistas da iminente leva de produtos com esta tecnologia. Em 2010 o CEO do Google, Erich Schmidt, exibiu o Nexus S, aparelho produzido pela Samsung em parceria com o Google e uma das responsáveis pela popularização do Android. O Nexus S. foi um dos primeiros smartphones a ter a tecnologia que permite troca de informações por aproximação do NFC. (RIBEIRO, 2013).

O NFC opera em uma frequência de 13.56 MHz e tem um alcance de 5 cm ou 10 cm embora alguns fabricantes estão ampliando esta faixa para uma série de outras novas aplicações, tais como entretenimento de áudio pessoal (GRATTON, 2007, tradução nossa).

## 1.1 OBJETIVO GERAL

Desenvolver um protótipo de aplicativo Android de identificação de animais utilizando a tecnologia NFC integrado.

## 1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desta pesquisa consistem em:

- a) Coletar dados referentes ao índice de abandono;
- b) Compreender e aplicar o funcionamento da tecnologia NFC;
- c) Compreender o funcionamento da linguagem voltada ao desenvolvimento para dispositivos Android;
- d) Desenvolver um servidor para gravação das informações referente ao animal;

e) Desenvolver uma aplicação móvel que identifique as *tags* NFC e cadastre os dados necessários para a identificação do animal na plataforma Android.

### 1.3 JUSTIFICATIVA

Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), os brasileiros possuem 52 milhões de cães e 22 milhões de gatos. Isso significa que 44,3% dos lares têm, pelo menos, um cão e 17,7% têm ao menos um gato (GLOBO, 2015)

O número de animais abandonados é uma preocupação muito grande, tanto para o bem-estar do ser humano quanto para o bem-estar do animal. Visualizando o problema levantado, a identificação de animais se faz necessária para que diminua a quantidade de animais nas ruas e facilite a identificação dos mesmos, já que esses seres não têm a capacidade de dizer nome, endereço e de onde vêm (WSPA, 2008, tradução nossa).

Ultimamente, nas redes sociais, recebe-se diariamente posts sobre animais desaparecidos. Para as pessoas que são apegadas nestes animais, esta é uma situação desagradável que acaba tornando-se um problema para o proprietário e o pequeno animal pode acabar tendo traumas diante desta situação. A identificação dos animais de estimação é fundamental para que o animal seja encontrado em caso de desaparecimento, porém, o mesmo informa que infelizmente as pessoas lembram-se disto apenas quando ocorre o desaparecimento do animal (MARCIO, 2012).

A identificação é, portanto, um instrumento essencial para a gestão eficaz da população de cães e gatos. Esta identificação é ainda mais importante quando acoplado a um registro, isto é, um sistema em que detalhe cada animal e os dados de seu proprietário em um banco de dados central (WSPA, 2008, tradução nossa).

Em muitos países da Europa e nos Estados Unidos a identificação dos animais é uma obrigação, prática que deveria ser adotada o mais rápido possível pelo Brasil para poder assim ter um maior controle sobre a taxa de natalidade, abandono e também sobre outras informações importantes sobre os animais domésticos (WSPA, 2008, tradução nossa).

A cada dia que passa a capacidade de um dispositivo móvel é cada vez mais ampla. Os celulares não têm mais somente a função de realizarem ligações, mas

também são utilizados para armazenamento de fotos e vídeos pessoais e desenvolvimento de sistemas. Acredita-se, porém, que estes celulares podem ir além e fazer muito mais do que já fazem nos dias de hoje. Os mesmos já estão substituindo até mesmo os cartões de créditos e poderão substituir leitores NFC que são extremamente caros para usuários comuns (HEITOR, 2013).

Segundo a Teleco (2015) o mercado de smartphones no Brasil chegou a 93,3% de todos os celulares vendidos no primeiro trimestre de 2015, chegando a 95% no mês de março. Nesse primeiro trimestre foi vendido um total de 14,1 milhões de smartphones, contra 1,0 milhão de aparelhos tradicionais, chegando a um total de 15,1 milhões de celulares vendidos. Comparando os dados do primeiro trimestre de 2014 com os de 2015, o número de celulares tradicionais vendidos teve uma queda de 80,2% enquanto que as vendas de smartphones aumentaram 32,9% no mesmo período (TELECO, 2015).

Por este motivo, foi escolhido uma das tecnologias mais recentes para resolver este problema, um smartphone com a tecnologia NFC integrada, tecnologia essa que está presente em muitos dos novos celulares inteligentes. Após uma rápida pesquisa em sites de vendas de smartphones, foi possível encontrar o Sony Xperia E3, que possui a tecnologia NFC integrada, pelo valor médio de quatrocentos reais.

Com o protótipo de aplicativo que foi desenvolvido bastará a pessoa ter um celular com a tecnologia NFC para poder identificar a *tag* presente em um animal, apresentando uma solução mais viável para as pessoas, já que atualmente, segundo o analista Mary Meeker, o Brasil é o quarto país do mundo em número de smartphones com 70 milhões de unidades, sendo que este número cresce cada vez mais com o passar do tempo (GUIMARÃES, 2013).

#### 1.4 ESTRUTURA DO TRABALHO

Este trabalho de pesquisa é constituído por oito capítulos. A estrutura é formada pela introdução, o objetivo geral e seus objetivos específicos, e pela justificativa para o qual o projeto será desenvolvido e se encontra no primeiro capítulo.

O segundo capítulo traz uma breve pesquisa sobre o abandono de animais, gastos médios realizados com animais de estimações e sobre os tipos de identificação de animais trazendo uma maior importância sobre o microchip que será o tipo de

identificação utilizada nesse projeto.

No terceiro capítulo foram levantados aspectos sobre a comunicação entre dispositivos móveis, explorando também as tecnologias NFC e RFID.

No quarto capítulo foram fundamentadas as novas tecnologias existentes em telefones inteligentes e ressaltado a pesquisa sobre a plataforma Android, detalhando o seu funcionamento, suas camadas de aplicações e as versões existentes. Para finalizar foram realizadas pesquisas sobre o funcionamento do NFC na plataforma Android.

O quinto capítulo aborda dados referentes à Web Service e suas padronizações.

O sexto capítulo trata dos trabalhos relacionados a este projeto de pesquisa, onde os mesmos abordam, direta ou indiretamente, alguns objetivos semelhantes a este projeto.

Entrando no sétimo capítulo, é descrito todo o desenvolvimento, definindo as etapas metodológicas do processo, como o estudo de ferramentas e a modelagem dos dados. Seguindo com os detalhes da codificação realizada, foram expostos no subitem implementação, passando pelo fluxo do sistema e resultados obtidos.

Por fim, o oitavo capítulo que se compõe pela conclusão obtida após a realização do trabalho, dificuldades encontradas e sugestões para trabalhos futuros.

## 2 ABANDONO DE ANIMAIS

Ao longo da história da humanidade, homens e animais sempre tiveram uma boa interação, sendo utilizados como meio de transporte e como fonte de alimentação. Com o passar do tempo essa relação foi ficando mais profunda, sendo que atualmente os animais servem para companhia, tratamento de doenças, cães policiais, cães de guarda e outras formas de convivência (SILVANO et al, 2010).

O Brasil atualmente é o segundo no ranking de população de cães e gatos, contando com um número de 101,1 milhões de animais de estimação. Em primeiro lugar estão os Estados Unidos. O crescimento do número de animais de estimação está acontecendo, pois, as pessoas estão mudando os seus estilos de vida, se isolando mais em seus lares e dando um maior valor afetivo a outras espécies (FIOCCO, 2012).

Os animais de estimação são completamente dependentes dos seres humanos, que devem fornecer a eles bem-estar físico, psicológico e emocional.

Segundo especialistas do Farm Animal Welfare Council (FAWC), conselho consultivo do bem-estar dos animais, existem alguns princípios essenciais para o bem-estar dos animais chamados de “Cinco Liberdades” explicadas na tabela 1, conforme definição do Conselho de Bem-estar de Animais de Produção. (DE PAULA, 2010).

Tabela 1 – As cinco liberdades do bem-estar animal

<b>Nomenclatura</b>	<b>Definição</b>
<b>Liberdade Nutricional</b>	Os animais devem estar livres de fome, sede e desnutrição.
<b>Liberdade Psicológica</b>	Os animais devem estar livres de medo e de estresse.
<b>Liberdade Sanitária</b>	Os animais devem estar livres de dor, doenças e ferimentos.
<b>Liberdade Ambiental</b>	Os animais devem ter liberdade de movimentos em ambientes adequados à sua espécie.
<b>Liberdade Comportamental</b>	Os animais devem ter liberdade para expressar o comportamento natural da sua espécie.

Fonte: Adaptada FAWC (2003, tradução nossa).

Em alguns estados existem leis mais específicas para o controle do bem-estar do animal, como por exemplo a cidade de Curitiba que criou a Lei Municipal 13.914 de 2011, onde afirma que os estabelecimentos que realizam vendas de animais vivos devem possuir alvarás cadastrados na prefeitura e relatórios de todos os animais nascidos, vendidos, permutados e doados. Todos os animais devem sair dos estabelecimentos de vendas com um microchip de identificação cadastrado no Sistema de Identificação Animal da Rede de Defesa e Proteção Animal de Curitiba (PREFEITURA MUNICIPAL DE CURITIBA, 2011).

Os locais de venda de animais devem dispor de um leitor para os microchips. Os animais revendidos devem também ter um atestado sanitário emitido por um médico veterinário que seja responsável pelo estabelecimento e o certificado deve conter o comprovante de vacinações do animal e condição reprodutora. No momento da comercialização, o local de venda deve disponibilizar documentos explicativos sobre a guarda responsável, orientações sobre a alimentação, higiene e cuidados médicos. No caso de descumprimento desta lei, os estabelecimentos podem sofrer sanções como, advertências por escrito, multa simples, multa diária, suspensão do alvará, entre outras (PREFEITURA MUNICIPAL DE CURITIBA, 2011).

## 2.1 GASTOS MÉDIOS COM ANIMAIS DE ESTIMAÇÕES

Muitas famílias fazem a aquisição de um animal de estimação sem planejamento financeiro, sendo esse um dos maiores fatores de abandono de animais no Brasil e no mundo (SILVA, 2006).

A Associação Brasileira da Indústria de Produtos para Animais de Estimação (ABINPET), divulgou uma pesquisa efetuada no estado de São Paulo com os gastos médios que podem ocorrer mensalmente com a aquisição de um animal de estimação. Conforme pesquisa, os gastos com um gato podem chegar a R\$ 84,19, enquanto os gastos com a aquisição de um cachorro são bem maiores, podendo chegar ao custo médio R\$ 308,00 para os donos de raças de grande porte e R\$ 133,00 para as raças de pequeno porte (ABINPET, 2012).

Para as pessoas que desejam ter um animal de estimação e que não podem gastar tanto com os mesmos recomenda-se aves, roedores ou peixes. O gasto médio mensal para roedores é de R\$ 25, enquanto para peixes é de R\$ 18 e para aves, que são os mais econômicos, R\$ 15 mensais (ABINPET, 2012).

## 2.2 TIPOS DE IDENTIFICAÇÃO DE ANIMAIS

A identificação é um instrumento fundamental na gestão das populações de cães e gatos. Existem inúmeras razões para que seja incentivada a identificação em animais de estimação, entre elas estão a facilidade de encontrar o dono de um animal desaparecido, diminuindo assim o número de animais nas ruas ou em abrigos, diminuindo também o encargo financeiro e administrativos para os órgãos públicos. Ao ligar o proprietário ao seu animal de estimação, poderá diminuir a probabilidade de abandono e incentivar os proprietários a assumir a responsabilidade de seus animais (WSPA, 2008, tradução nossa).

A identificação fica ainda mais eficaz quando acoplada a um sistema de identificação que seja unificado a um banco de dados central (normalmente nacional). A cada animal é atribuído um código de identificação permanente, quando pesquisado este código no banco de dados será possível encontrar o contato do proprietário do animal (WSPA, 2008, tradução nossa).

Existem vários métodos de identificação apropriados para animais de estimação sendo que o método escolhido deve ser o mais humano e eficiente possível. Os métodos são classificados de três formas, permanente, semipermanente e temporário; dentro desses temos as tatuagens, recorte da orelha, marca a frio, microchip, coleira de identificação entre outros (WSPA, 2008, tradução nossa).

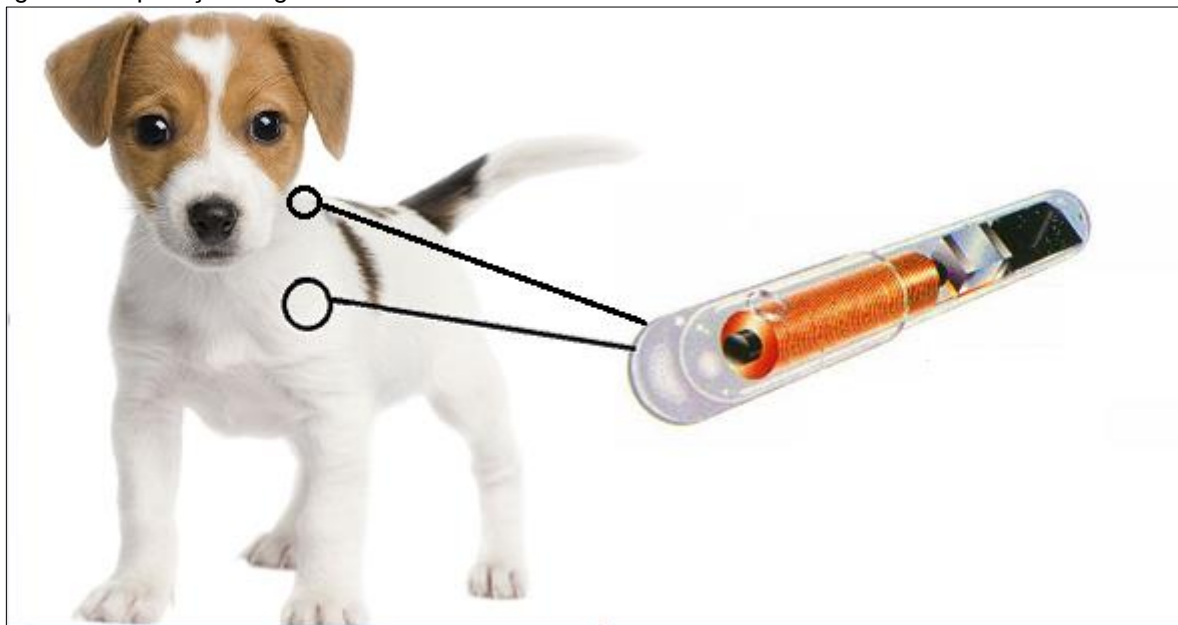
### 2.2.1 Microchip

O microchip é uma tecnologia inovadora para a monitorização de animais, e com várias vantagens se colocado em comparação com os demais métodos de identificação (MANSANO et al, 2013).

O microchip é composto de material esterilizado e é implantado abaixo a pele do animal, a sua aplicação não causa dor no animal e não necessita sedação. O mesmo possui um microcircuito contendo um código único e inalterado, inserido dentro de uma capsula de biovidro cirúrgico. O microchip é inativo e não possui uma bateria própria, o mesmo passa a transmitir informação apenas quando entra em contato com um leitor, retornando os dados referente ao mesmo. Cada microchip possui um ID único que seria um numero de quinze dígitos, que são utilizados para realizar a identificação do mesmo (PREFEITURA MUNICIPAL DE CURITIBA, 2011).

Na figura 1 é demonstrado o microchip e também os locais mais comuns onde o microchip pode ser inserido no animal, que seria o seu dorso ou no lateral da pata dianteira.

Figura 1 – Aplicação Tag no animal



Fonte: Cães Pro (2015)

O microchip tem uma vida de vinte e cinco anos, sendo assim um tipo de identificação permanente. O processo de implantação é rápida e indolor e não causará nenhum desconforto se implantado corretamente (WSPA, 2008, tradução nossa).

### 3 COMUNICAÇÃO DE DADOS ENTRE DISPOSITIVOS MÓVEIS

No momento que foram introduzidos os dispositivos móveis no mercado o único objetivo deles era a comunicação por voz. Com o passar do tempo e o avanço da tecnologia, foram introduzidos outros objetivos para esses dispositivos, tais como mensagens escritas, mensagens com imagens e Internet. Com a tecnologia melhorando com o passar dos anos, os dispositivos moveis vem ganhando mais funcionalidades, tais como sendo exemplos a introdução do infravermelho, Bluetooth, Global Positioning System (GPS), Wireless Fidelity (WiFi), NFC (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

No mercado de dispositivos móveis, está crescendo os dispositivos equipados com o NFC, tecnologia derivada do RFID, que vem sendo usada cada vez mais para tarefas do dia-a-dia, é uma tecnologia nova e potente para os usuários de dispositivos móveis (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

#### 3.1 RADIO FREQUENCY IDENTIFICATION

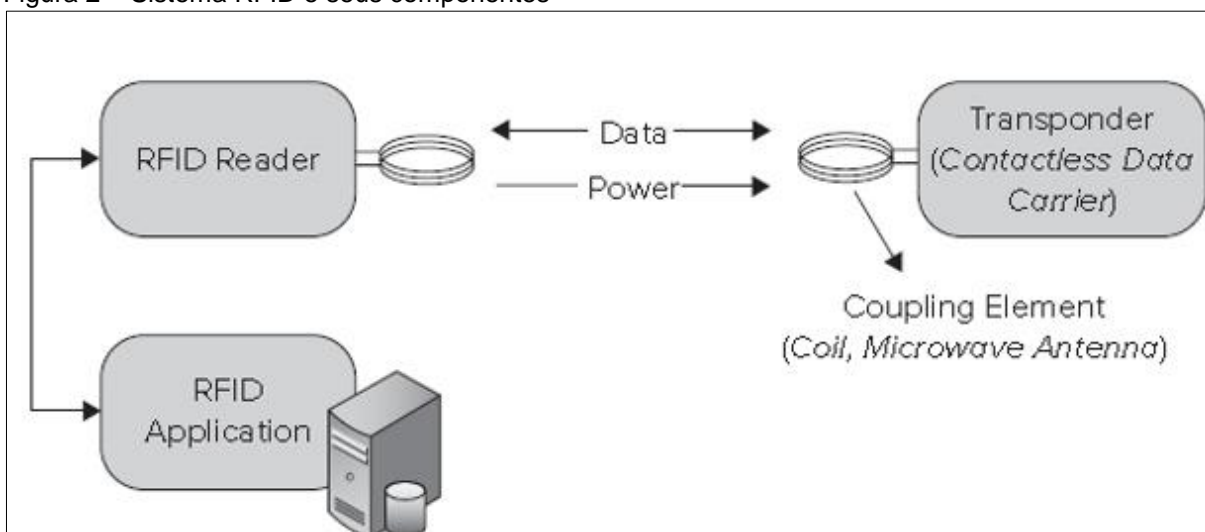
RFID é uma tecnologia de comunicação sem fio para troca de dados utilizando as ondas de rádio (FINKENZELLER, 2010, tradução nossa).

Ele tem gerado atenção nos últimos anos, pois está substituindo outros métodos de identificação, como por exemplo, o código de barras. Porém ao contrário do código a barras as etiquetas RFID não precisam estar visíveis para transmitir informações. Essa tecnologia foi criada na segunda guerra mundial para identificar aviões aliados, sendo usada em sequência para identificar e rastrear materiais nucleares (SAMPLE et al, 2008, tradução nossa).

Nos dias de hoje, essa tecnologia encontra-se em várias partes do nosso dia-a-dia como, por exemplo, nas chaves que abrem portões de apartamentos, no passe rápido para meios de transportes que se encontram nas rodovias, entre outros (IGOE, 2012, tradução nossa) .

Na figura 2 é ilustrado um sistema RFID simples, juntamente com seus componentes. O resultado da transmissão de dados varia dependendo da frequência utilizada. Os leitores podem ler e escrever dados nas *tags* também chamadas de transponder ou etiquetas (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

Figura 2 – Sistema RFID e seus componentes



Fonte: Coskun, Ok e Ozdenizci (2012)

Um sistema RFID é constituído principalmente por dois componentes: a etiqueta e o leitor. As etiquetas são dispositivos que são inseridos no objeto ao qual se deseja identificar. Existem dois tipos de etiquetas RFID, passivos e ativos.

As etiquetas passivas são compostas de um sistema com circuito integrado que tem um transmissor-receptor de rádio frequência e uma pequena parte de memória não volátil. A alimentação acontece através da corrente passada através do leitor da *tag*, essa alimentação recebida é suficiente apenas para que a etiqueta consiga transmitir os dados para o leitor.

As etiquetas ativas possuem um sistema com fonte de alimentação e um receptor de rádio próprio. Sistemas ativos tem um alcance muito maior que sistemas passivos, de consequência são menos propensos a erros e mais caros (IGOE, 2012, tradução nossa) .

O leitor contém tipicamente um modulo de alta frequência com um interpretador que serve para decodificar a informação transmitida pela etiqueta, uma unidade de controle e uma antena (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

O custo de um sistema RFID pode variar muito. A compra e instalação de um sistema ativo pode custar dezenas de milhares de dólares, enquanto um leitor e etiquetas passivas custam em média cem dólares (IGOE, 2012, tradução nossa).

Existem diversos protocolos RFID sendo eles de alta e baixa frequência. Na categoria de baixa frequência existem as bandas 125 kHz e 134,2 kHz, os leitores de alta frequência estão na banda de 13.56 MHz que permite taxas de leituras com

uma velocidade maior (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

O foco do RFID é automatizar a leituras de etiquetas para facilitar o rastreamento e monitoramentos de elementos, podendo também ser usado para identificação de animais, porém, o custo fica pouco viável para o usuário final que necessita de um equipamento para a identificação de seu animal de estimação (IGOE, 2012, tradução nossa) .

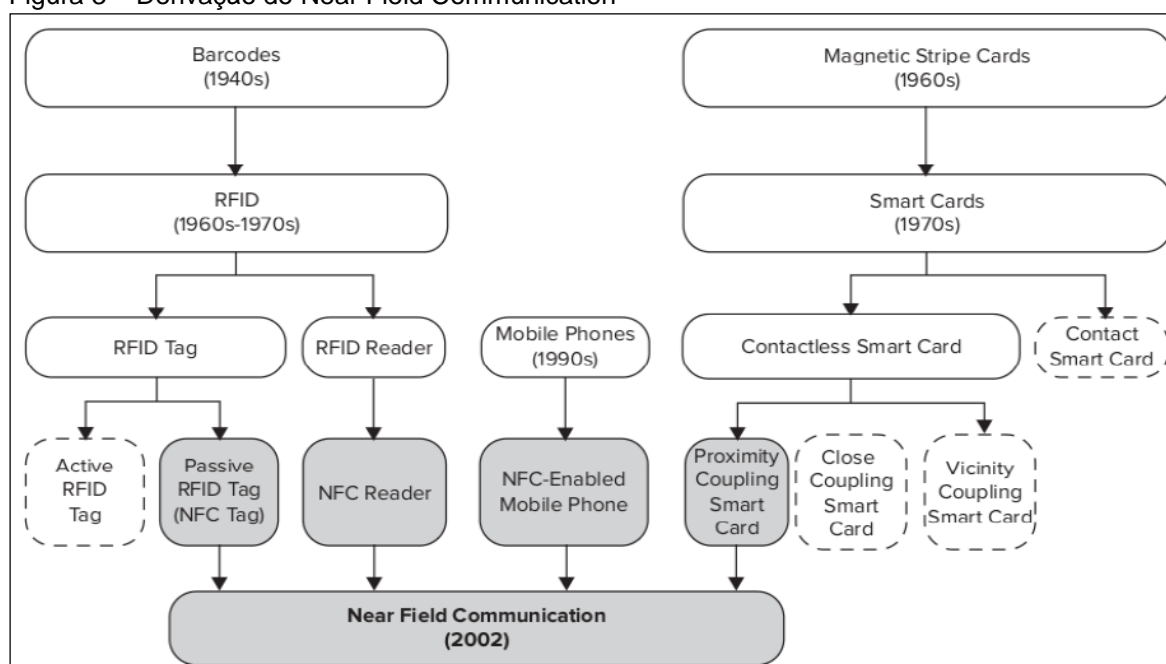
Tendo em vista o alto custo do RFID, foi desenvolvida uma nova tecnologia com uma maior segurança, o NFC que tem uma maior integridade nas informações e um custo acessível a muitas pessoas, já que essa tecnologia está presente na maiorias de celulares inteligentes existentes nos mercado (COSKUN; OK; OZDENIZCI, 2013, tradução nossa).

### 3.2 NEAR FIELD COMMUNICATION

O NFC é uma tecnologia de comunicação sem fio de curto alcance que permite o compartilhamento bidirecional de pequenas cargas de dados entre dispositivos eletrônicos (ANDROID DEVELOPERS, 2015c, tradução nossa).

Essa tecnologia foi baseada na tecnologia RFID, já existente e permitindo trocas mais complexas com o uso de etiquetas inteligentes conforme mostra a figura 3 (IGOE, 2012, tradução nossa) .

Figura 3 – Derivação do Near Field Communication



Fonte: Coskun, Ok e Ozdenizci (2012).

Esta tecnologia foi desenvolvida no ano de 2002 a partir de tecnologias já existentes, a MIFARE e FeliCa, produtos que foram desenvolvidos pela NXP Semiconductors (antiga Philips Semiconductors) em conjunto com a Sony (LANGER; ROLAND, 2010, tradução nossa).

No ano de 2003, a European Computer Manufacturers Association (EMCA), International Organization for Standardization (ISO) e International Electrotechnical Commission (IEC) aprovaram o padrão NFC. Essa padronização criou uma verdadeira compatibilidade entre aparelhos com NFC embarcados, sendo que no ano de 2004 a Philips, Nokia e Sony criaram um fórum chamado NFC Fórum, onde são definidos o futuro da adoção e do desenvolvimento da tecnologia (GRATTON, 2007, tradução nossa).

O NFC funciona na frequência central 13,56 MHz, e sua velocidade de transmissão de dados pode variar entre 106, 212 Kb/s e 424 Kb/s (kilobits por segundo). O padrão se chama Near Field Communication Interface and Protocol 1 (NFCIP-1) e os modos de comunicação para interface e protocolo são definidos pela norma ISO/IEC 18092. Esta norma define esquemas de modulação, codificações, velocidades de transferência, protocolo de transporte e controle de colisão entre outras especificações (INNOVISION RESEARCH & TECHNOLOGY, 2013, tradução nossa).

Na frequência 13,56 MHz pode-se chegar a aproximadamente 10 centímetros do dispositivo emissor do sinal, tornando assim o NFC também compatível com as normas RFID da mesma faixa. Atualmente, a taxa de transmissão de dados pode chegar a 106 kbit/s (baseado no hardware MIFARE), 212 kbit/s a 424 kbit/s (baseados no hardware FeliCa). (LANGER; ROLAND, 2010, tradução nossa).

A norma ISO/IEC 18092 define também os modos de comunicação que são os mesmos do RFID: ativo e passivo. No modo ativo os dispositivos transmitem os dados gerando cada um o seu próprio campo de rádio para a transmissão, enquanto o modo passivo apenas um dos dispositivos gera um campo de radio transmissão e o outro dispositivo também utiliza essa modulação para gerar dados. Conforme visto no protocolo de NFC o responsável por gerar o campo de rádio é o dispositivo de iniciação (INNOVISION RESEARCH & TECHNOLOGY, 2013, tradução nossa).

Os dispositivos NFC assim como os RFID, podem trabalhar também em modo ativo e em modo passivo. Além de existirem os modos ativos e passivos os aparelhos NFC tem a capacidade de trabalhar em três diferente modos de operação:

leitura/escrita, peer-to-peer (P2P) e emulação de cartões (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

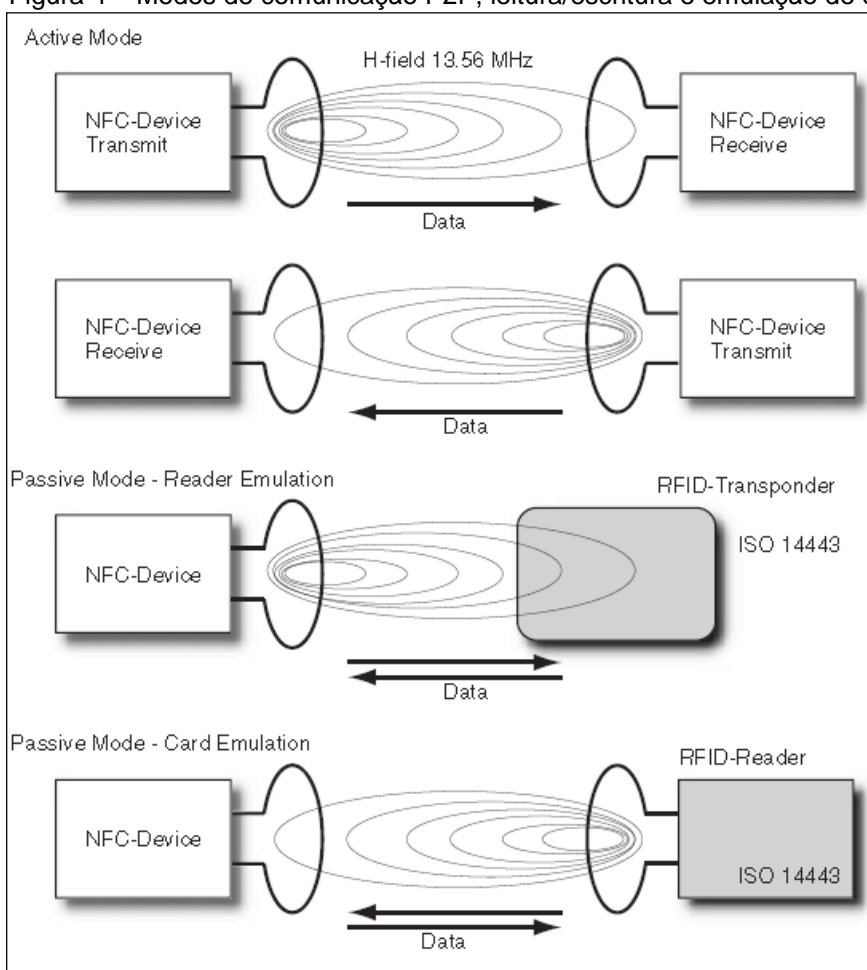
Os modos de operação leitura/escrita é conhecido por permitir assim leitura e gravação de etiquetas incluindo as smartcards MIFARE e FeliCa, que são as principais *tags* RFID e já são suportadas pelo NFC.

Neste modo o dispositivos móveis atuam como um iniciador e o *tag* passiva atua como um alvo. O dispositivo iniciador gera sinas de radio e o dispositivo de destino fica alimentado por este campo eletromagnético e responde ao dispositivo iniciador através da modulação existente no campo eletromagnético. Os exemplos são o smart-poster, que são cartazes inteligentes com a tecnologia NFC integrada no mesmo, possibilitando assim aos consumidores de celulares com tecnologia NFC, podem tocar no poster e receber informações de conteúdo, fazer downloads, receber cupons e entrar nos aplicativos de mídia digital (PATEL; KOTHARI, 2013).

O modo de operação P2P, ou terminal, cria uma conexão de rede ponto-a-ponto entre dois dispositivos ativos para a troca de informações, este modo é padronizado na norma ISO / IEC 18092 e permite que os dados sejam transmitidos em uma velocidade de 424 Kbit/s, pode-se citar como exemplo a transferências de arquivos entre dispositivos móveis (COSKUN; OK; OZDENIZCI, 2013, tradução nossa).

O modo de operação emulação de cartão é muito usado no acesso de controle de trânsito e nos cartões de proximidades. Neste modo um dos dispositivos NFC funciona como um cartão RFID e outros dispositivos NFC pode ler dados desse dispositivo, essas informações são utilizadas para outras operações, como pagamento, ou controle de acesso, entre outras funcionalidades (PATEL; KOTHARI, 2013).

Figura 4 – Modos de comunicação P2P, leitura/escritura e emulação de cartões



Fonte: Finkenzeller (2010).

Os leitores NFC são dispositivos ativos que permitem transferência de dados bidirecional com outro elemento NFC. Os leitores podem ler ou escrever dados, seja entre dispositivos ativos ou ativo e passivo. Ultimamente a integração da tecnologia NFC em dispositivos móveis cria oportunidade para exibir a facilidade de uso e consentimento do ecossistema (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

O NFC também trabalha com *tags*, que podem ser etiquetas, pulseiras, pequenos microchips com uma pequena antena que podem conter uma pequena quantidade de dados para transferir para outro dispositivo NFC, como um celular. (PATEL; KOTHARI, 2013, tradução nossa).

Essas *tags* NFC podem ser colocadas em qualquer local de modo a fornecer ou recolher dados entre o dispositivo móvel e as *tags*. Para que seja possível a transmissão de dados, bastam que ambos estejam perto e que o dispositivo suporte a tecnologia NFC. Essas *tags* são distribuídas por empresas de semicondutores, as

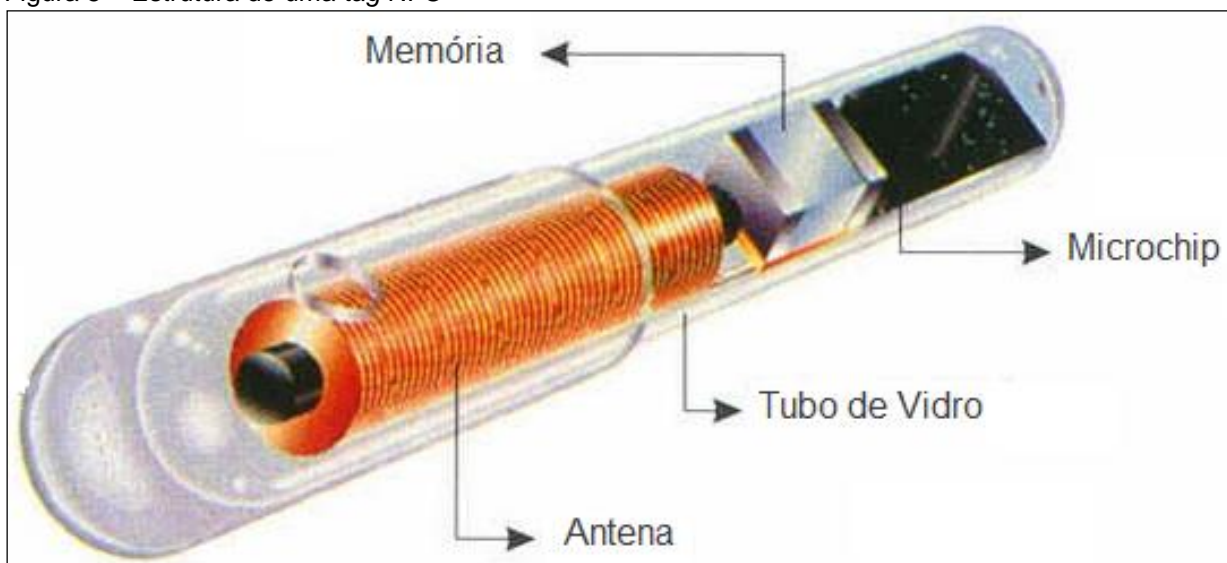
principais marcas são NXP, Broadcom, Qualcomm, Inside Secure, Renesas, Texas Instruments, e STMicroelectronics (NOH; LEE; CHOI, 2014).

Em média o custo de uma tag NFC é de 5 reais (SONNTAG, 2011).

A arquitetura de um *tag* NFC é composta por um chip de rádio ligado a uma antena e uma pequena área de memória. A antena é feita de alumínio ou cobre (figura 5). O cobre acaba se tornando mais caro, porém tem uma melhor execução. As antenas são indutores projetados para converter campos magnéticos das proximidades em energia. As *tags* são classificadas em quatro tipos:

- a) tipo 1: é baseado na ISO / IEC 14443A e tem um tamanho de memória que pode ser entre 96 bytes e 2 Kbytes e uma velocidade de comunicação de 106 kbit/s e tem a capacidade de ser protegida contra gravação.
- b) tipo 2: se diferencia da tipo 1 apenas pela memória que pode ser entre 48 bytes e 2 Kbytes.
- c) tipo 3: são baseadas na norma industrial japonesa (JIS) X 6319-4, o tamanho da memória pode ser de até 1 Mbyte e a velocidade de transmissão pode chegar até 212 kbit / s.
- d) tipo 4: podem armazenar até 32 Kbytes e são compatíveis com a ISO / IEC 14443 série padrão (A&B) e sua velocidade pode chegar até 106 kbit/s.

Figura 5 – Estrutura de uma tag NFC



Fonte: Alves (2008).

A tecnologia NFC facilita vida do usuário quando devidamente aplicada. Porém, pouco adianta proporcionar estes benefícios se não considerarmos a questão de segurança do NFC (ALECRIM, 2012).

A distância de comunicação entre os dispositivos é realmente pequeno, dificultando assim a interceptação do sinal, porém, não é impossível. Para resolver este problema foi criado o protocolo Single Wire Protocol (SWP). O SWP é considerada uma interface que proporciona uma comunicação segura entre o chip do celular e o chip NFC do aparelho. Mesmo que já existente, o protocolo SWP não é amplamente adotado pois não é uma solução com funcionamento completo, a mesma ainda encontra-se sob revisão final pelo European Telecommunications Standards Institute (ETSI) (ALECRIM, 2012).

Segundo Clark (2013, tradução nossa) até 2017 os aparelhos com NFC embarcados irão atingir uma taxa anual de crescimento de 65%, podendo chegar a 2,1 bilhões de unidades.

#### 4 DISPOSITIVOS MÓVEIS

A cada dia o mercado de celulares está crescendo cada vez mais. Metade da população mundial possui um celular. Os usuários de celulares estão cada vez mais atrás de dispositivos que vão além da simples ligação ou das mensagens de texto, querem celulares com diversos recursos, como câmera, músicas, acesso à internet, redes sócias, NFC, entre outros (LECHETA, 2010).

As empresas também estão crescendo e se tornando cada vez mais tecnológicas, e estão inserindo no seu dia-a-dia aplicações que utilizam os dispositivos móveis. Empresas estão sempre visando no aumento do lucro, e os dispositivos móveis podem ser uma grande relevância em um mundo em que a palavra “mobilidade” está cada vez mais conhecida e importante (LECHETA, 2010).

Para Lyytinen e Yoo (2002) a computação móvel veio para que as pessoas possam ter seu computador sempre presente, independentemente do local em que se vá, com as mesmas funcionalidades de um computador tradicional.

Em 2011 cerca de 712,5 milhões de smartphones foram vendidos no mundo, isso significa um aumento de mais ou menos 11% referente ao ano de 2010. Já no ano de 2012 o aumento em confronto ao ano de 2011 foi de 38,8% (TELECO, 2013).

Hoje uma grande quantia de smartphones já vem com a tecnologia NFC, e segundo a eMarketer, cerca de 1.9 bilhões de pessoas já possuem smartphones e em 2018 esse número poderá chegar a 2.5 bilhões, um terço dos consumidores em todo o mundo. Em 2018 esse número representará 51,7% de todos os usuários de telefonia móvel. Segundo a eMarketer o Brasil ocupa a 6º posição no *ranking* referente ao número de smartphone por número de usuário (EMARKETER, 2014).

A tradução de smartphone significa “Telefone Inteligente” e seus principais sistemas operacionais são IOS da Apple, o Android do Google, e o Windows Phone da Microsoft, BlackBerry, Symbian entre outros (BARROS, 2012).

Segundo a International Data Corporation, o Android e iOS foram responsáveis por mais que 96% de todos os embarques de smartphones no quarto trimestre de 2014.

O Android lidera esse mercado com 76,6% de Market Share no trimestre, e o iOS da Apple vem logo depois com 19,7% no trimestre. O Windows Phone teve uma participação de 2,8% no trimestre e o sistema operacional Blackberry permanece

abaixo de um por cento de participação (TELECO, 2015).

#### 4.1 SISTEMA OPERACIONAL ANDROID

Nessa nova era, os dispositivos móveis são os produtos de consumo mais utilizado no mundo, com 5 bilhões de aparelhos existentes, mais da metade da população mundial (PEREIRA; SILVA, 2012).

Com esse crescimento, o desenvolvimento de aplicativos vem aumentando e os desenvolvedores e empresas, buscam uma alternativa que seja moderna e ágil. Para que essa evolução da tecnologia seja satisfatória aos usuários, o Google deu continuação ao Android, que inicialmente foi desenvolvido pelo Android Inc., para ocupar esse espaço. Essa nova plataforma de desenvolvimento acabou causando impacto quando anunciada. Porém, não é apenas a Google que está por trás da plataforma, e sim um grupo de empresas, chamado Open Handset Alliance (OHA) (LECHETA, 2010).

A OHA é composta por líderes do mercado de dispositivos móveis. Alguns dos integrantes são HTC, LG, Motorola, Samsung, Toshiba, ASUS, Intel, Dell e outros. Durante o primeiro ano, a OHA contava com 84 membros. Um dos objetivos deste grupo é a criação de uma plataforma aberta para dispositivos móveis para que os clientes fiquem mais satisfeitos com o produto final. Outro objetivo da OHA é ter uma plataforma flexível e moderna para o desenvolvimento de aplicações corporativas (OPEN HANDSET ALLIANCE, 2012, tradução nossa).

O Android é uma plataforma completa para dispositivos móveis, que envolve um pacote com programas para celulares, um sistema operacional, middleware<sup>1</sup>, aplicativos e interface do usuário. Esta aplicação tem como intenção permitir que os desenvolvedores de aplicações para celulares possam obter o máximo proveito dos recursos que um dispositivo móvel possa oferecer. A plataforma também possui um conjunto de desenvolvimento, com ferramentas e Application Programming Interfaces (API) chamado Android Software Development Kit (SDK), este kit utiliza a linguagem de programação Java e se faz necessário para o desenvolvimento dos aplicativos Android (PEREIRA; SILVA, 2012).

Esta plataforma está em constante evolução, por ser open source. As

---

<sup>1</sup>Middleware: software que integra duas outras aplicações individuais, como por exemplo, conexão de um sistema de banco de dados a um servidor web.

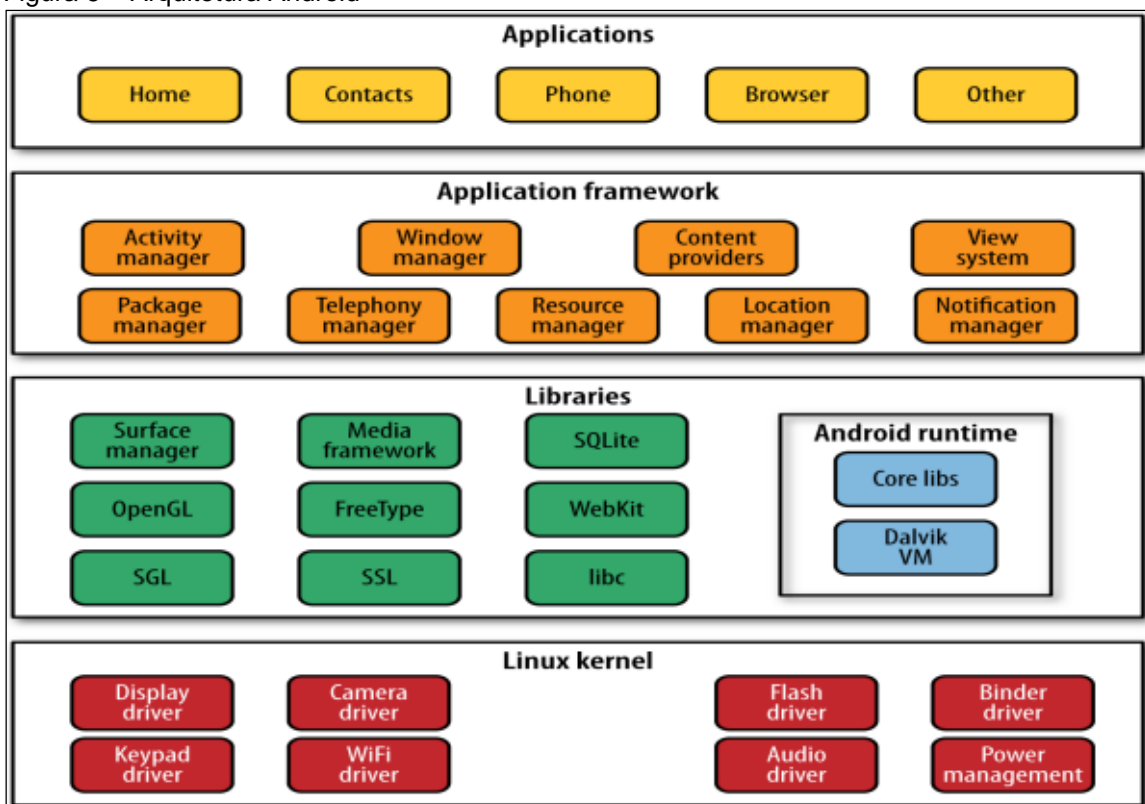
comunidades de desenvolvedores estão cada vez mais trabalhando em conjunto para criar aplicações móveis inovadoras. Mesmo sendo open source, o Android é muito seguro, pois o mesmo é executado em um Kernel Linux, sendo que é criado um novo usuário Linux com diretórios específicos toda vez que um aplicativo for instalado em uma estação.

Tendo em vista que os dados de cada aplicativo estão isolados uns dos outros, para que a informação seja compartilhada deve haver uma autorização do usuário, podendo ser negada essa autorização e de consequência não ocorrer a instalação do aplicativo (PEREIRA; SILVA, 2012).

#### 4.1.1 Plataforma Android

A arquitetura do sistema operacional Android está dividida em camadas conforme figura 6.

Figura 6 – Arquitetura Android



Fonte: Gargenta e Nakamura (2014).

Cada camada tem a responsabilidade de gerenciar seus respectivos processos. O Android é composto por um *Kernel Linux*, um conjunto de bibliotecas

C/C++ fornecidos por um framework de aplicação que fornece serviços e gerenciamento de tempo de execução e aplicações (MEIER, 2009, tradução nossa).

Cada camada possui características próprias, e em alguns casos elas não são claramente separadas, podendo acontecer que uma pode se infiltrar-se dentro da outra (GARGENTA, 2011, tradução nossa). As camadas serão especificadas abaixo.

#### 4.1.1.1 Camada de aplicações

Está é a última camada e a mais alta do diagrama de arquitetura. É a camada que o usuário final irá visualizar e interagir. Nesta camada se encontram todos os aplicativos do Android. Temos como exemplo, e-mail, programa de SMS, navegadores, mapas, calendários entre outros que podem ser criados pela comunidade de desenvolvedores (PEREIRA; SILVA, 2012).

Toda aplicação desenvolvida Android é compartilhada como um pacote, contém o código da aplicação, recursos de mídia utilizados e o manifesto do aplicativo (que identifica as ações, componentes e permissões requeridas para seu funcionamento). Este formato pode ser intimamente relacionado ao formato padrão de empacotamento da linguagem Java (.jar), definido especificamente para poder ser executado sem a necessidade de desempacotamento. Para a criação de um aplicativo Android, é necessário criar aplicativos em Java, que serão executados na máquina virtual Dalvik (JACKSON, 2011, tradução nossa).

#### 4.1.1.2 Camada do framework

É a camada no qual se encontram toda estrutura, recursos e APIs utilizadas para que uma aplicação possa ser criada no Android, como classe visuais, provedor de conteúdo, gerenciador de localização entre outros (PEREIRA; SILVA, 2012).

Através do framework de acesso às bibliotecas nativas, além de diversos recursos. Pode-se notar que a maior parte dessa camada é manager, ou seja, eles gerenciam algo (SCHMITZ, 2014). Pereira e Silva (2012) afirmam que os principais componentes desta camada são:

- a) activity manager: gerenciador de ciclo de vida das aplicações, incluindo a “pilha de telas” para a navegação do usuário;
- b) package manager: tem a responsabilidade de informar quais pacotes

- estão sendo utilizados pelo dispositivo;
- c) window manager: responsável pelo gerenciamento das apresentações de janelas;
  - d) location manager: utilizado para saber a localização do dispositivo através do GPS;
  - e) content providers: este componente disponibiliza o compartilhamento de informações entre aplicativos e a transferência de dados entre aparelhos;
  - f) view system: disponibiliza a parte gráfica para a aplicação.

#### 4.1.1.3 Camada do framework

As *Libraries* do Android contém todo o código fonte e são responsáveis em lidar diretamente com o núcleo do sistema (o Kernel). Esse conjunto de bibliotecas é escrito em C/C++ e possuem a função de dar acesso as principais funcionalidades do sistema operacional. Essas bibliotecas possuem códigos de acesso ao banco de dados SQLite, por exemplo e a biblioteca WebKit, para conseguir utilizar as funcionalidades da navegação na web (LEE, 2011).

Segundo Lee (2011) algumas das bibliotecas nativas inclui:

- a) surface manager: o Android utiliza um gerenciador de janelas de composição similar ao Vista ou Compiz, mas é muito mais simples. Ao invés de desenhar diretamente no buffer da tela, os comandos de desenho vão para outro buffer de imagem, que combinados com outras imagens formam o que o usuário visualiza. Este sistema possibilita a criação de efeitos, como as transições suaves entre as telas;
- b) freeType: biblioteca de renderização de fontes;
- c) codecs de mídia: possibilita a reprodução de vídeos e gravação e reprodução de áudios e vários formatos incluindo AAC, AVC (H.264), H.263, MP3 e MPEG-4;
- d) banco de dados SQL: Android inclui o banco de dados SQLite, com ele é possível utilizar de armazenamento de dados nas aplicações. A plataforma não é limitada nas bibliotecas nativas, sendo que a partir da versão 1.5 o desenvolvedor tem a possibilidade de criar e implantar suas próprias bibliotecas através do Native Development Toolkit (NDK) e testá-

las usando o Dalvik Virtual Machine (DVM) (SCHMITZ, 2014).

#### 4.1.1.4 Camada do ambiente de execução

A camada do ambiente de execução também chamada Android Runtime é a camada responsável pela execução do código compilado. Esta camada contém uma série de bibliotecas que permite que os desenvolvedores escrevam aplicativos usando a linguagem Java. É uma instância da máquina virtual Dalvik e somente é criada quando uma aplicação é executada (PEREIRA; SILVA, 2012).

A DVM é uma implementação criada pelo Google em Java, otimizada para dispositivos móveis, todos os códigos escritos para Android são em Java e executados dentro da DVM (BURNETTE, 2010, tradução nossa).

#### 4.1.1.5 Linux Kernel

O Android é construído em cima de uma fundação sólida e comprovada: o kernel do Linux. Criado por Linus Torvalds em 1991, o Linux pode ser encontrado hoje em tudo, desde relógios de pulso a supercomputadores. O Linux fornece uma camada de abstração de hardware para o Android, permitindo assim que o Android utilize uma grande variedade de plataformas (BURNETTE, 2010, tradução nossa).

A camada Linux Kernel, base do sistema operacional que utiliza a versão 2.6 do Kernel do Linux, é responsável pela segurança, gestão de processos, gestão de memória, pilhas de protocolos de rede e modelos de drivers. Esta camada possui todos os drivers de baixo nível para seus vários componentes (LEE, 2011).

A camada Linux Kernel tem à sua disposição um potente sistema próprio de gerenciamento de energia, onde um aplicativo requisita o gerenciamento de energia e o driver de energia do Kernel passa a checar periodicamente todos os dispositivos, caso algum deles não esteja sendo utilizado, o sistema de gerenciamento o desliga (PEREIRA; SILVA, 2012).

### 4.1.2 Camada de aplicações

Uma aplicação em Android é composta por várias funcionalidades, entre elas estão o toque de um alarme, abertura de um contato telefônico entre outros. Essas funcionalidades podem ser classificadas em quatro componentes Android diferentes, Tabela 2 (STEELE; TO, 2010).

Tabela 2 – Possíveis componentes de uma aplicação Android

Funcionalidade	Classe de Base Java	Exemplos
Foco nas ações que usuário pode realizar	Activity	Editar uma nota, jogar
Processo de segundo plano	Service	Tocar música, atualizar ícone do tempo
Receber mensagens	BroadcastReceiver	Disparar uma notificação sobre o evento
Armazena e recupera dados	ContentProvider	Abrir a agenda de contatos

Fonte: Adaptado de Steele e To (2010, tradução nossa)

Cada aplicação é composta por ao menos um dos componentes citado acima. Os mesmos são instanciados pelo sistema operacional, conforme necessário. Cada componente possui várias funcionalidades, sendo que cada um possui seu próprio ciclo de vida, sendo ele criado, focado, desfocado e destruído (STEELE; TO, 2010).

Cada componente de uma aplicação necessita de sua declaração dentro de um arquivo Extensible Markup Language (XML), chamado `AndroidManifest.xml`.

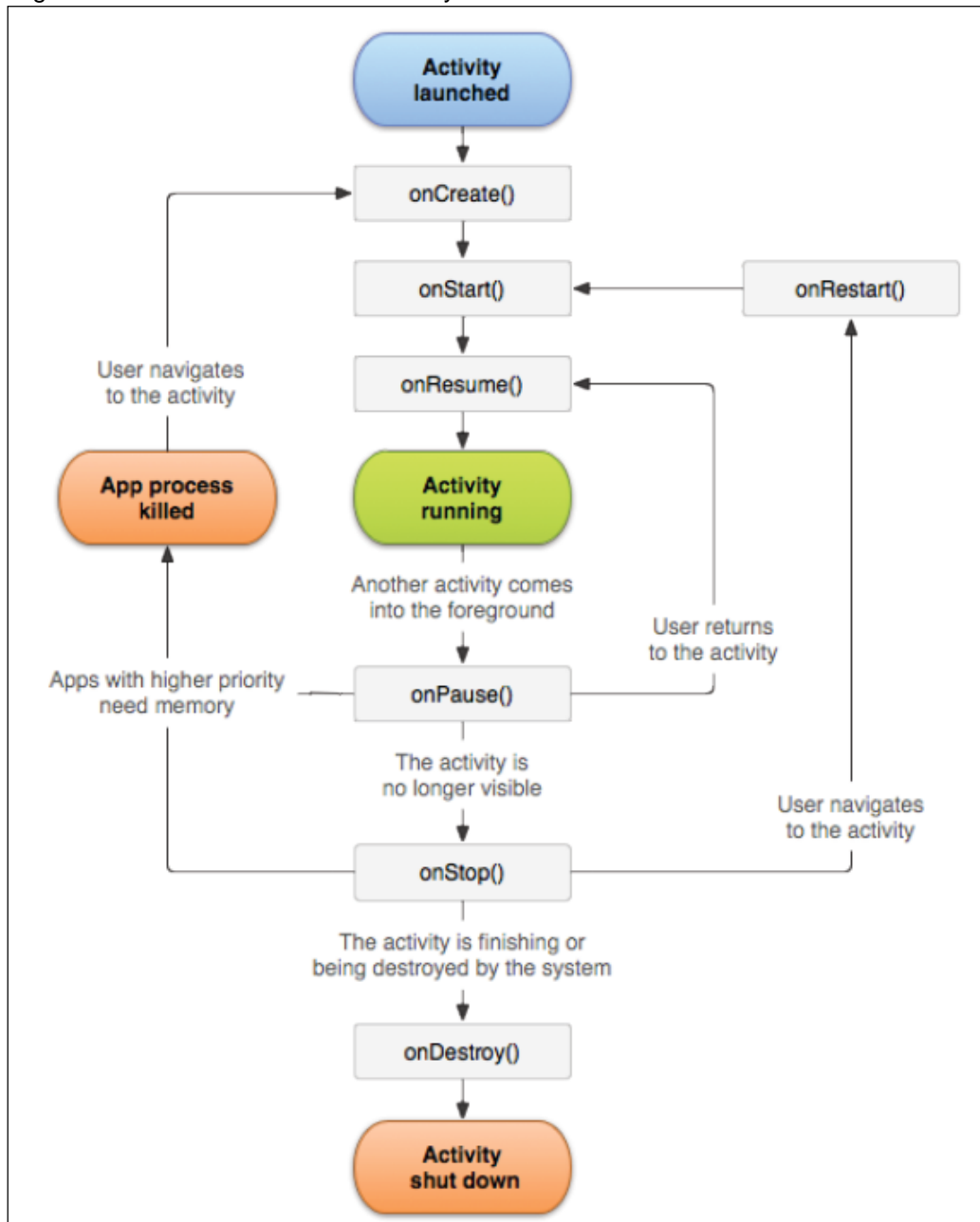
De todos os componentes a *Activity* é o mais utilizado, na maioria das vezes é representado por uma tela na aplicação que tem a responsabilidade de tratar eventos gerados nessa tela, por exemplo, quando o usuário seleciona um item do menu. Uma *Activity* representa uma atividade, ação ou funcionalidade que o usuário pode realizar dentro de uma aplicação (LECHETA, 2010).

Analogamente, assim como um site possui várias páginas, uma aplicação Android possui várias *activities*. Assim com o site tem um “home page”, a *activity* possui sua *activity main*, geralmente a primeira a ser exibida quando se inicia uma aplicação, podendo também navegar entre as *activities* ou mesmo saltar diretamente para outra de um aplicativo completamente independente (GARGENTA, 2011, tradução nossa).

Para realizar o gerenciamento de cada aplicação o sistema Android segue um ciclo de vida, Figura 7, ou seja, os possíveis estados em que ela se encontra. Cada processo em execução na plataforma Android é colocado em uma pilha. Quando

utilizado uma *activity* em primeiro plano, todo seu processo é colocado no topo da pilha de processos e o processo anterior é movido para o final da pilha (ABLESON; SEN, 2011, tradução nossa).

Figura 7 – Ciclo de vida de uma Activity Android



Fonte: Android Developers (2015a).

Método de um ciclo de vida de uma *Activity* segundo Pereira e Silva 2012:

- a) `onCreate()`: é chamado e criado quando uma atividade é iniciada, momentos antes dela aparecer para o usuário;
- b) `onStart()`: método que indica que a atividade está preste a se tornar

visível para o usuário;

- c) onResume(): é o topo da pilha de atividade, é chamado quando inicia a iteração com o usuário;
- d) onPause(): Este método é chamado quando a atividade está prestes a ir para segundo plano, geralmente porque uma nova atividade será iniciada. Neste momento devem ser salvos todos os dados que ainda não foram salvos;
- e) onStop(): este método é chamado quando a atividade não está sendo mais utilizada pelo usuário e perdeu foco para outra atividade, porém, ainda está na memória. Neste momento a atividade pode ser destruída, removida ou trazida para frente novamente;
- f) OnRestart(): este método é iniciado quando a atividade se torna visível novamente, após o evento onStop();
- g) g) OnDestroy(): chamado antes da aplicação ser destruída.

#### 4.1.3 Versões do Android

O Android vem melhorando com o passar do tempo, isso reflete nos números de versões lançadas, conforme Figura 8 (GLAUBER, 2015).

Figura 8 – Versões do Android



Fonte: Ibnlive (2014).

Cada versão adicionou novas funcionalidades para a plataforma: (GLAUBER, 2015):

- a) versão 1.5: foram adicionadas suporte para bibliotecas nativas em

aplicativos Android;

- b) versão 1.6: foram adicionadas funcionalidades referentes ao suporte de telas com tamanhos diferentes;
- c) versão 2.0: adicionou suporte para o multi-touch e melhor resolução para a tela;
- d) versão 2.2: melhorou ainda mais a resolução da tela e adicionou suporte a tecnologia Flash;
- e) versão 2.3: adicionou a tecnologia NFC e um novo coletor de lixo;
- f) versão 3.0: foi disponibilizada a versão para *tablets*;
- g) versão 4.0: foi desenvolvida para unificar *tablets* e smartphones em uma única versão e trouxe também o desbloqueio de tela por reconhecimento facial, entre outros;
- h) versão 4.1/4.2: versão otimizada para consumir menos bateria, redimensionamento do tamanho dos ícones das aplicações, foi disponibilizado também o aplicativo *Google Now* que fornece informações de restaurantes, transporte ou entretenimento, baseado na sua geolocalização do momento. Com a versão 4.2 também temos uma melhora no aplicativo de fotos, permitindo tirar fotos panorâmicas em 360°;
- i) versão 4.4: otimizado o desempenho do sistema, foi otimizado também a memória e aperfeiçoando a tecnologia *touchscreen* para que responda com mais rapidez e precisão. Nessa versão o Android suporta o perfil MAP para que carros com Bluetooth possam trocar mensagens com os dispositivos. O Android 4.4 oferece uma nova arquitetura aberta para realização de pagamentos NFC compatível com qualquer operadora móvel e permite que os apps gerenciem as informações de pagamento na nuvem ou no dispositivo;
- j) versão 5.0: versão compatível com relógios inteligentes, sistema com melhor desempenho e uma maior economia no consumo de bateria. Uma das características do Android 5.0 está nas notificações: as mensagens podem ser visualizadas sem a necessidade de desbloquear o smartphone e uma das principais mudanças foi que o mesmo não utiliza mais a *Dalvik* e sim agora é Android Runtime (ART) como máquina virtual.

## 4.2 DESENVOLVIMENTO DO NFC DENTRO DA PLATAFORMA ANDROID

Com o avanço da tecnologia, tanto o usuário quanto o ambiente podem interagir com o dispositivo de diferentes formas, uma delas é a comunicação por campo de curta distância, o NFC. O padrão permite que a tecnologia seja utilizada em três diferentes formas (ANDROID DEVELOPERS, 2015, tradução nossa):

- a) emulação de cartão: o dispositivo é um cartão que pode ser lido por outros leitores;
- b) modo leitor: o leitor pode ler *tags* NFC;
- c) modo P2P: dois dispositivos podem se comunicar e trocar dados.

O NFC no Android foi inserido na versão 2.3, codinome Gingerbread, onde foi apresentado a funcionalidade NFC Reader Mode, o modo leitor e a definição de mensagens e registros NFC Data Exchange Format (NDEF). A maior parte das APIs do framework é fundamentada no formato de dados NDEF, contudo, ele permite o uso de outros formatos como MIFARE ou a definição de um protocolo próprio (ANDROID DEVELOPERS, 2015b, tradução nossa).

Na versão 4.0 o modo P2P foi incluído através da funcionalidade nomeada de Android Beam™, que permite a troca de dados entre dispositivos Android, tendo em conta que a aplicação que deseja enviar dados deve estar em primeiro plano no dispositivo iniciador e que o receptor não esteja com a tela bloqueada. Uma função semelhante ao Beam, o NDEF push, existe desde a versão 2.3.3, contudo, é considerada antiquada e é recomendada apenas para suportar aparelhos antigos (ANDROID DEVELOPERS, 2015c, tradução nossa).

Na versão 4.4 foi inserida a opção de emulação de cartão, é um método adicional que não precisa de um chip separado do aparelho, chamado de elemento seguro. Com o elemento seguro a comunicação com o terminal NFC acontece através dele, sem que necessite do envolvimento de qualquer aplicação Android. Na emulação que acontece com a versão 4.4, os dados são encaminhados diretamente para a Central Processing Unit (CPU), assim a aplicação Android pode controlar os dados diretamente (ANDROID DEVELOPERS, 2015d, tradução nossa).

## 5 APLICAÇÕES WEB

A internet mudou o jeito de viver das pessoas, mudou o jeito de trabalhar e jogar. O advento da internet revolucionou também com a história da computação. A programação em Java juntamente com um servidor começou no ano de 1997, quando a Sun Microsystems lançou o Beta "Java™ Web Server" e Java Servlet Developers Kit (TEMPLE et al, 2004).

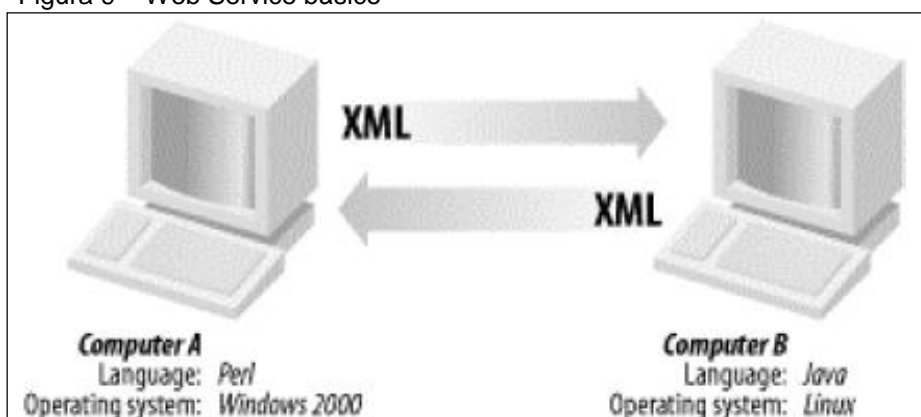
Logo se viu que o potencial de comunicação e iteração era muito grande e ia além da exibição de simples conteúdo. Com isso, as páginas estáticas não seriam mais suficientes. Com essa necessidade deu a origem as páginas dinâmicas. A maioria dos sites que acessos são páginas dinâmicas. O usuário realiza a requisição de algo ao servidor, que irá processar a requisição e retornar uma resposta ao usuário (CAELUM, 2015).

### 5.1 WEB SERVICE

Web Services surgiu como consequência natural da utilização da Internet. Com o crescimento da Internet no mercado corporativo, surgiu também a necessidade de integrar aplicações além das redes locais e, por consequência, em ambientes heterogêneos. Nesse contexto, e com uma união entre as empresas como IBM, Microsoft, BEA e outras que pertencem ao World Wide Web Consortium (W3C), surgiu a tecnologia de Web Services (GOMES, 2014).

Define-se web service como uma aplicação de software que pode ser acessada remotamente através de diversas máquinas distintas, Figura 9. É utilizado um sistema de mensagem de XML em um formato Simple Object Access Protocol (SOAP) específico, padronizado, e que não esteja amarrado a qualquer sistema operacional ou linguagem de programação (POTTS; KOPACK, 2003, tradução nossa).

Figura 9 – Web Service básico



Fonte: Cerami (2002).

### 5.1.1 Padronização

O W3C e a Organization for the Advancement of Structures Information Standards (OASIS) são as instituições responsáveis pela padronização das Web Services. Os padrões são úteis para que a indústria não trabalhe com projetos criados por uma única empresa e ficando dependendo dela. Sendo que os concorrentes se unem a estas organizações com o objetivo de criar padrões de consenso e com maior probabilidade de obtenção de credibilidade perante a indústria.

Dentre as principais organizações especificadoras, W3C controla as especificações SOAP, Web Service Description Language (WSDL), XML Schema e HyperText Transfer Protocol (HTTP), a OASIS controla as especificações Universal Description Discovery and Integration (UDDI), WS-Security e Security Assertion Markup Language (SAML) (POTTS; KOPACK, 2003, tradução nossa).

Os principais padrões mais comuns utilizados pela web services segundo Cerami (2002), são:

- a) SOAP: é um protocolo baseado em XML para a troca de informações entre computadores. Embora SOAP possa ser usado em uma variedade de sistemas de mensagens e pode ser entregue através de uma variedade de protocolos de transporte, o foco inicial de SOAP são as chamadas de procedimento remotas transportados via HTTP. Portanto, o SOAP permite que aplicativos clientes se conectem facilmente a serviços remotos e invoquem métodos remotos;
- b) WSDL: é uma gramática XML para especificar uma interface pública de um serviço web e tem como objetivo descrever as operações que

compõem uma Web Service, definindo como deve ser a entrada e saída dessas operações;

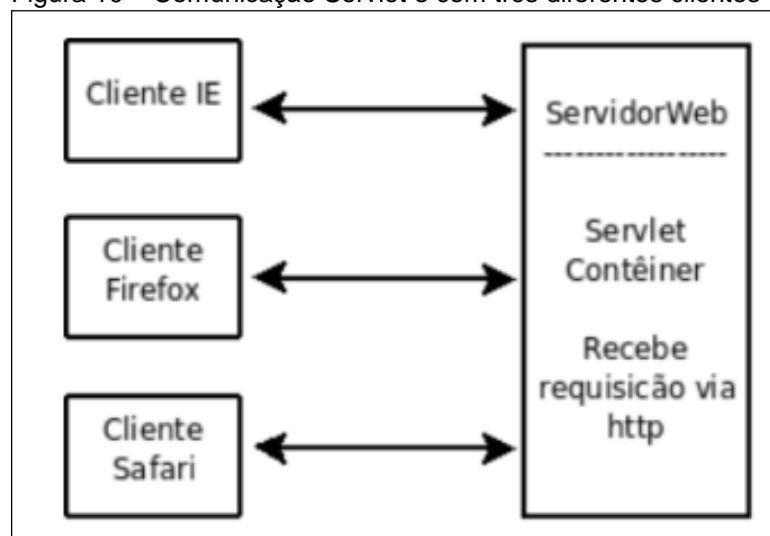
- c) XML: é a base em que os web services são criadas. O XML disponibiliza a descrição, o armazenamento, o formato da transmissão para troca de dados;
- d) UDDI: é um padrão desenvolvido para prover informações em forma de diretório sobre web services.

## 5.2 SERVLET

Servlets são um tipo de classe Java que é executada em um Servidor. Tem uma estrutura bem definida, e que, quando instaladas junto a um Servidor que permita a execução de Servlets, são chamados algumas vezes de Servidor de aplicações web (TEMPLE et al, 2004).

Cada Servlet é um objeto Java que através do request recebe as requisições a através do response, produz algo, como uma página HTML gerada dinamicamente, Figura 10, pode-se verificar o acesso de três diferentes clientes ao mesmo servidor através do protocolo HTTP (CAELUM, 2015).

Figura 10 – Comunicação Servlet e com três diferentes clientes

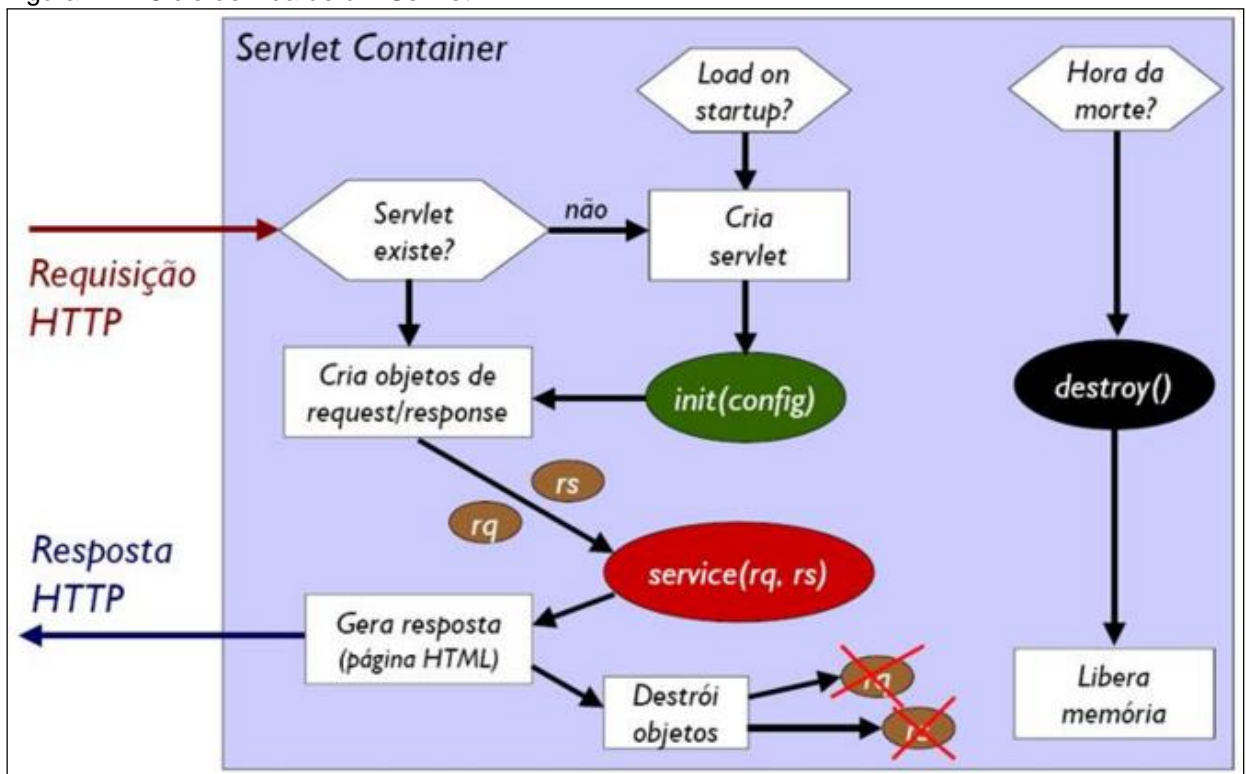


Fonte: Caelum (2015).

O ciclo de vida de um Servlet é controlado pelo container e é composto por 3 partes, inicialização, atendimento de uma requisição e finalização.

- a) Inicialização: ocorre quando o Servlet Container recebe uma requisição e a carrega em um Servlet. Após a inicialização, o Servlet pode atender as requisições. O container carrega a classe do Servlet na memória, cria uma instância da classe do Servlet e inicializa a instância chamando o método `init(ServletConfig)`.
- b) Atendimento: cada requisição é executada em um método `service` (`ServletRequest`, `ServletResponse`). O container realiza a criação de um objeto de requisição (`ServletRequest`) e de resposta (`ServletResponse`), após a criação chama o `service()` passando os objetos como parâmetros, depois o envio da resposta os objetos são destruídos.
- c) Finalização: quando o container decide remover o Servlet da memória, ele o finaliza chamando o método `destroy()`, na figura 11 pode-se compreender melhor este ciclo de vida (LEHRER, 2012).

Figura 11 – Ciclo de vida de um Servlet



Fonte: Palmeira (2013).

## 6 TRABALHOS CORRELATOS

Para a realização deste trabalho foram pesquisados alguns trabalhos semelhantes, adquirindo assim um maior embasamento do que já foi realizado. Foram encontrados diversos trabalhos nacionais e internacionais com o desenvolvimento de aplicativos que utilizam a tecnologia NFC.

### 6.1 DESENVOLVIMENTO DE UM MÓDULO DE ACIONAMENTO PARA ABERTURA DA FECHADURA DE UM QUARTO DE HOTEL BASEADO NA TECNOLOGIA NFC

Desenvolvido no ano de 2014 por Renato Ramos da Silva como trabalho de conclusão de curso de Pós-Graduação em Desenvolvimento de Produtos Eletrônicos do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, este projeto teve como objetivo desenvolver um módulo de acionamento capaz de abrir a fechadura de um quarto de hotel utilizando a tecnologia NFC a partir de dispositivos que utilizem o sistema operacional Android (SILVA, 2014)

A solução proposta para este trabalho é a utilização de smartphones com tecnologia NFC para ativar a abertura da fechadura do quarto de hotel. Ao realizar o check-in na recepção do hotel, o código NFC acoplado ao smartphone do hóspede recebe permissão para acessar um determinado quarto. No momento em que o usuário aproxima o smartphone da fechadura, o módulo de acionamento consulta o controlador do quarto que, por sua vez, consulta a permissão de acesso no banco de dados que caso seja permitida o módulo de acionamento abre a fechadura, caso contrário esta permanecerá na posição padrão, ou seja, fechada (SILVA, 2014).

### 6.2 ACESSIBILIDADE EM BIBLIOTECAS: UTILIZANDO NFC E SMARTPHONES PARA AUXILIAR DEFICIENTES VISUAIS NA IDENTIFICAÇÃO DE LIVROS

Este trabalho foi desenvolvido por Ederaldo de Oliveira Constantino e Vânia Paula de Almeida Nêris para a Universidade Federal de São Carlos como trabalho de conclusão do curso de Ciência da Computação no ano de 2014 e teve como objetivo o desenvolvimento de uma aplicação para pessoas com deficiência visual com o objetivo de melhorar a acessibilidade em uma biblioteca (CONSTANTINO; NERIS, 2014).

Ao aproximar o celular com o livro, são exibidos três botões permitindo assim que o usuário possa solicitar a versão em Braille, realizar o download do áudio do livro ou solicitar empréstimo do livro de maneira convencional. Para melhorar a acessibilidade das pessoas com deficiência e garantir o acesso ao aplicativo também foi instalado um software chamado Talkback, responsável pela leitura da tela e dos menus (CONSTANTINO; NERIS, 2014).

A tecnologia utilizada a este trabalho permitiu um ambiente onde o usuário com deficiência visual pode realizar o empréstimo do livro desejado com mais autonomia e segurança. Para isso basta haver a disposição um smartphone com o recurso NFC e leitor de tela ativado, espera-se que o usuário possa percorrer as estantes da biblioteca e escolher o livro desejado (CONSTANTINO; NERIS, 2014).

### 6.3 NFC TICKETING: A PROTOTYPE AND USABILITY TEST OF AN NFC-BASED VIRTUAL TICKETING APPLICATION

Este trabalho foi desenvolvido por Stefano Levaldi Ghiron, Serena Sposato, Carlo Maria Medaglia e Alice Moroni para a Universidade Sapienza de Roma, e tem como o desenvolvimento de um protótipo de aplicação móvel para o uso em bilhetagem eletrônica, também realizando um estudo sobre a usabilidade do software desenvolvido. A Aplicação desenvolvida permite ao usuário comprar e utilizar bilhetes de forma menos complexa, em uma combinação das tecnologias NFC e SMS (GHIRON et al, 2009, tradução nossa).

A arquitetura é formada por um smart poster (com uma etiqueta passiva MIFARE); uma aplicação, que permite comprar e utilizar os bilhetes; uma aplicação para proteger os bilhetes utilizando-se do módulo de segurança; uma aplicação, para o usuário validar os próprios bilhetes e uma última para verificação pelo gerenciador do sistema (GHIRON et al, 2009, tradução nossa).

Com o auxílio dos componentes se inicia um ciclo, onde o usuário inicia tocando o smartphone habilitado com NFC ao smart poster, que direciona o usuário a compra do bilhete eletrônico. Após a compra dos bilhetes o usuário pode utilizar o smartphone como bilhete nos pontos de verificação (GHIRON et al, 2009, tradução nossa).

O estudo também conseguiu testar usabilidade de forma a encontrar os pontos fracos de aplicação, sugerindo que estes possam ser eliminados em uma

versão futura (GHIRON et al, 2009, tradução nossa).

#### 6.4 DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL BASADA EN LA TECNOLOGÍA NFC PARA ACCESO A INFORMACIÓN DE LAS PIEZAS DE ARTE DE UN MUSEO

Este trabalho foi desenvolvido por Jesús Jorge Herrera Mires como trabalho de conclusão de curso de engenharia da computação da Universidade Católica do Peru no ano de 2013, e tem como objetivo o desenvolvimento de uma aplicação móvel voltada para o sistema operacional Android para agilizar e simplificar o acesso à informação das peças de arte em um museu (MIRES, 2013, tradução nossa).

A arquitetura do trabalho é composta por um aplicativo de leitor da *tag*, e onde também serão visualizados os dados referente a obra de arte, também é composto por uma aplicação web para cadastro das obras de artes pelo administrador do museu (MIRES, 2013, tradução nossa).

Com os componentes acima o usuário irá ler os dados da *tag* presente na peça do museu, em seguida busca através de um web service os dados referente a peça consultada, que foi cadastrada posteriormente pelo administrador. Caso não há dados referentes aos dados da peça, o sistema retornará que não existem dados para consultar (MIRES, 2013, tradução nossa).

A *tag* foi escrita através do aplicativo TagWriter, que é uma aplicação móvel desenvolvida pela NXP para armazenar dados e eventos executáveis na *tag* NFC, como contatos, números de telefone, condução Bluetooth, SMS, e-mail, texto simples, e muito mais.

#### 6.5 NFC TAGS-BASED NOTIFICATION SYSTEM FOR MEDICAL APPOINTMENTS

Este trabalho foi desenvolvido por Quratulain e Munazza (2014) como um artigo, pela Universidade de Engenharia e Tecnologia do Paquistão e foi publicado na revista International Journal of Automation and Smart Technology. O trabalho tem como objetivo um sistema de notificações utilizando *tags* NFC, para agendar uma próxima consulta.

O sistema proposto é dividido em duas partes, escrever informações na *tag* NFC e recuperar informações da *tag* NFC.

A arquitetura do sistema é composta por um sistema para inserir os dados da próxima consulta em uma base de dados central, em seguida os dados são enviados para um servidor. Através de um aplicativo Android serão acessados os dados gravados no servidor e grava esses dados em uma *tag* NFC passiva, ou no próprio telefone com tecnologia NFC do paciente. Em seguida o usuário irá ativar o aplicativo disponibilizado que irá ler a *tag* e transferir os dados sobre a consulta para o aplicativo, que irá cadastrar automaticamente um alerta para o dia da consulta. No trabalho foi utilizado *tags* NFC com armazenamento interno de 128 bytes

## 7 APLICATIVO ANDROID PARA IDENTIFICAÇÃO DE ANIMAIS UTILIZANDO A TECNOLOGIA NFC

Por meio do conhecimento adquirido durante o levantamento bibliográfico, foi possível criar um protótipo de aplicação de identificação de animais baseada na plataforma Android, utilizando a tecnologia NFC para possibilitar a identificação através da comunicação entre smartphones e *tag*.

A aplicação tem como principais funções:

- a) cadastro de usuário: que será identificado como o proprietário do animal;
- b) cadastro do animal: necessitando antes da identificação da *tag* NFC para realizar o cadastro;
- c) consulta do animal: na consulta, será listados todos os animais cadastrados para o proprietário, possibilitando também a exclusão e edição dos dados do animal;
- d) identificação do animal: que pode ser feita sem necessidade de um cadastro.

Tendo em vista que a identificação de animais é importante, o intuito deste projeto é a criação de algo que facilite essa identificação, evitando assim o sofrimento do animal e do dono que perdeu seu animal.

### 7.1 METODOLOGIA

O início deste projeto de pesquisa foi constituído pelo levantamento bibliográfico a fim de obter boas referências bibliográficas e conteúdo teórico. Foram também realizadas pesquisas sobre a tecnologia NFC, sua funcionalidade e também o modo de interagir entre *tag* e smartphone.

A seguir os estudos foram voltados para a idealização do sistema e a seleção de ferramentas de desenvolvimento.

A fundamentação teórica foi dividida em partes com objetivo de direcionar o trabalho e estruturar a implementação. Esse estudo iniciou com a pesquisa sobre o índice de abandono de animais e a importância da identificação dos mesmos.

Em seguida o foco foi direcionado ao estudo da tecnologia NFC, de onde surgiu, e também para um melhor entendimento sobre aspectos gerais de transmissão e as demais características.

Foi realizado também o estudo sobre o Sistema Operacional Android, que foi a plataforma escolhida para o desenvolvimento do trabalho proposto por ser a plataforma de código fonte aberto, fácil desenvolvimento devido ao maior conhecimento da parte do acadêmico e em seguida foi realizado um estudo sobre o funcionamento do NFC junto com o sistema Android juntamente com uma web service para manter as informações em um banco de dados, compartilhando as informações com outros dispositivos.

Com este embasamento teórico, se iniciou o desenvolvimento do protótipo. Primeiramente foi realizado um levantamento as ferramentas as serem utilizadas para o desenvolvimento do projeto.

Para entender melhor o funcionamento do NFC no android, foi criada uma aplicação teste, apenas com funcionalidades básicas do NFC, como, ler e escrever em uma *tag*. Em seguida se se iniciou a modelagem do banco de dados para a criação do banco. A próxima etapa foi o desenvolvimento da aplicação.

Os testes foram realizados durante o desenvolvimento da aplicação. Todas as compilações e execuções foram realizadas diretamente no dispositivo móvel com a plataforma android.

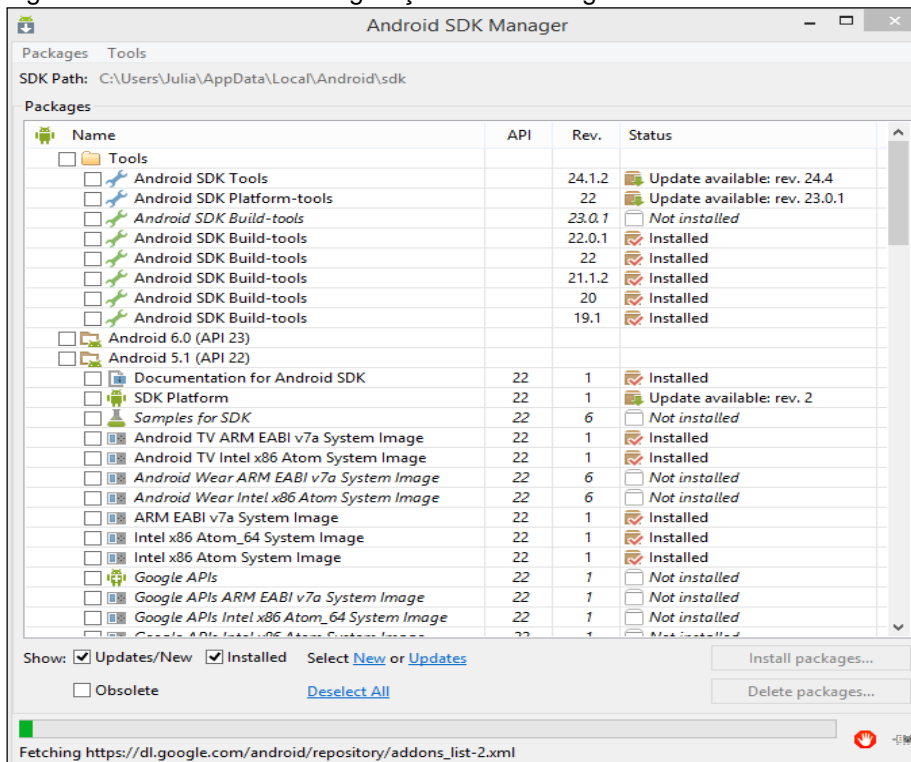
## 7.2 ESTUDO DAS FERRAMENTAS UTILIZADAS

Para dar início ao desenvolvimento do aplicativo android foi utilizado o software Android Studio<sup>2</sup> na versão 1.2.2, juntamente com o Kit de Desenvolvimento para o Android (Standard Development Kit - SDK) na API 21 da plataforma utilizada no projeto e configurado conforme figura 12.

---

<sup>2</sup> O Android Studio pode ser baixado no link a seguir <http://developer.android.com/sdk/>.

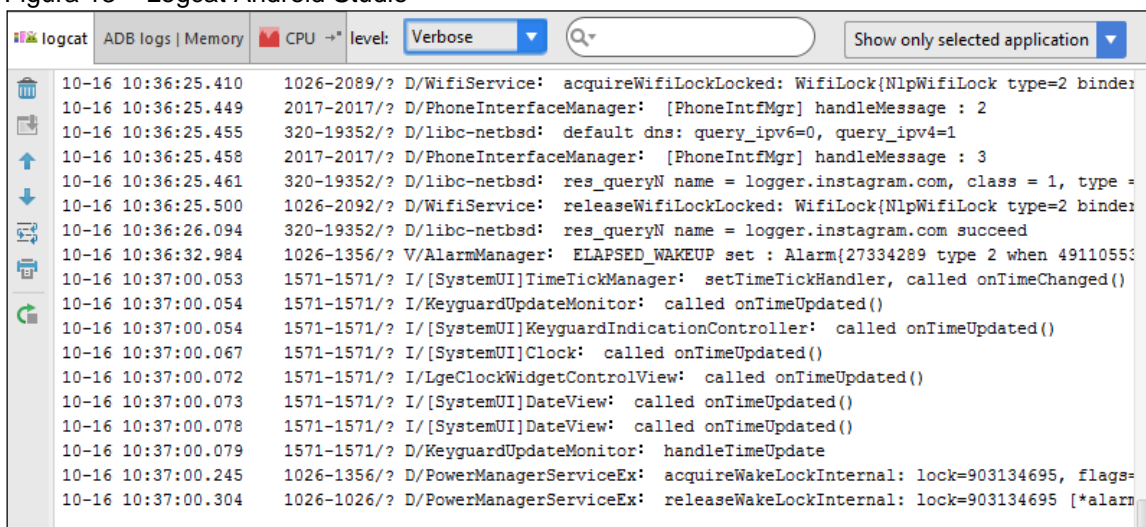
Figura 12 – Utilitário de Configuração SDK Manager.



Fonte: Do autor

A ferramenta utilizada no desenvolvimento e que já disponibilizada pelo Android SDK foi o LogCat demonstrado na figura 13, que tem a funcionalidade todos os logs do sistema operacional, como mensagens implementas pelo desenvolvedor para serem mostradas na console da IDE de desenvolvimento.

Figura 13 – Logcat Android Studio



Fonte: Do autor

O dispositivo ao qual foram submetidos os testes foi o LG G3 (figura 14),

tal aparelho roda a versão 5.0 do Android API 21, o aparelho possui uma tela de 5.5 polegadas sensível ao toque com resolução de 1440 x 2560 pixel e possui a tecnologia NFC integrada.

Figura 14 – Equipamento utilizado para avaliação



Fonte: LG (2015).

Para o desenvolvimento do serviço de comunicação responsável por receber os dados do aplicativo no computador o *serverlet*, foi utilizado a IDE Netbeans<sup>3</sup> 8.0.2 devido à familiaridade no desenvolvimento web, em conjunto com a JDK 1.8.

O banco de dados utilizado para o desenvolvimento do banco de dados foi o postgresSQL na versão 9.4, devido a ser um banco de dados gratuito, de boa performance e familiaridade.

### 7.3 MODELAGEM DE DADOS

Para a criação do banco foi necessário realizar uma pesquisa sobre os dados mais importantes para se saber sobre um animal encontrado.

A modelagem do banco de dados facilita o desenvolvimento do projeto, pois através desse modelo será possível ter uma percepção maior de como irá ficar a ligação entre os cadastros, diminuindo assim possíveis erros que poderiam ocorrer no momento do desenvolvimento.

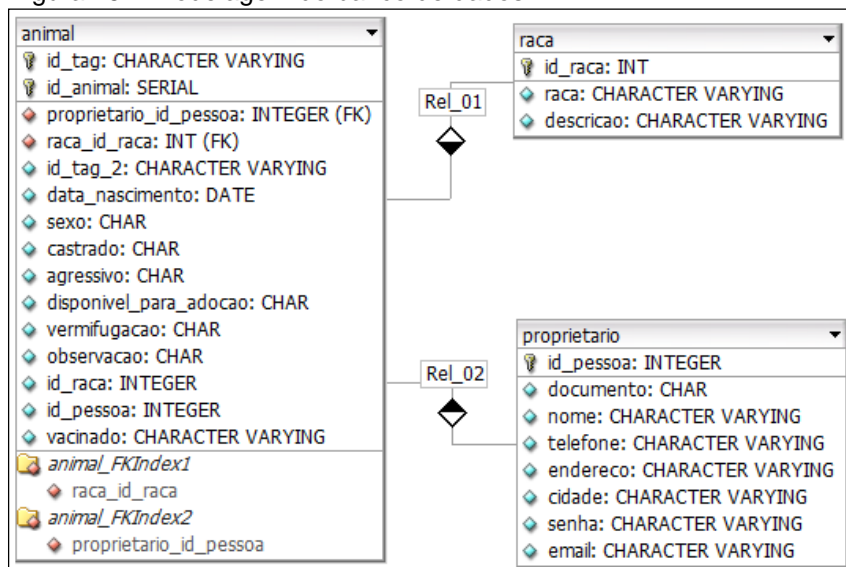
Para a modelagem do banco de dados, foi utilizado o DBDesigner 4 que é

---

<sup>3</sup> O Netbeans pode ser baixado no link a seguir  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

um software livre utilizado para a modelagem de dados visual. Segundo Medeiros 2015, DBDesigner pode ser utilizado para qualquer tipo de banco de dados principalmente para MySQL, Oracle, MSSQL e PostgreSQL, sendo este último o banco utilizado para desenvolvimento do projeto sendo assim, a figura 15, apresenta o modelo físico do banco de dados presente no servidor.

Figura 15 – Modelagem do banco de dados

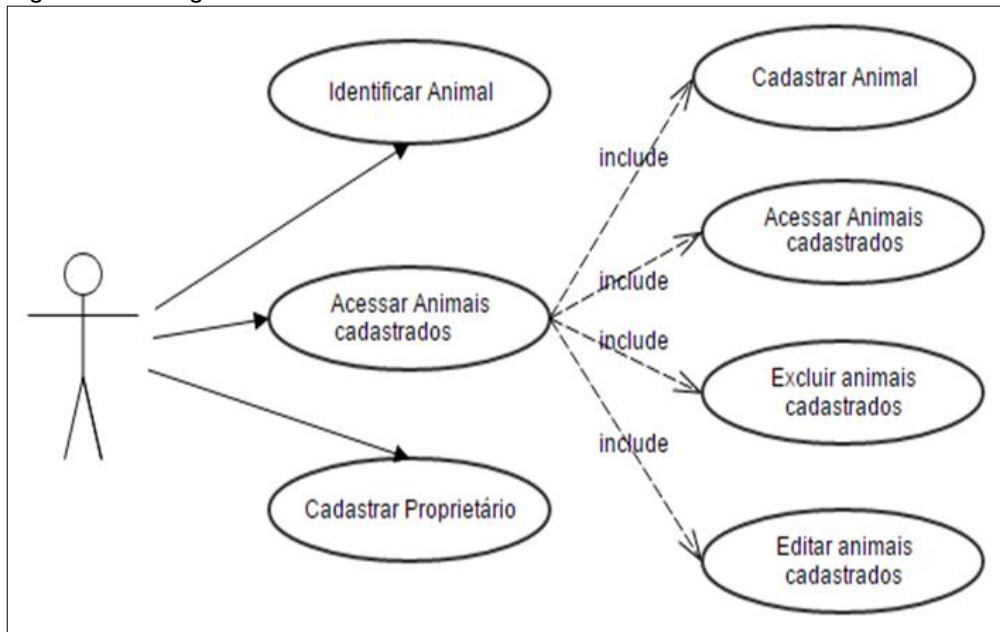


Fonte: Do autor

O protótipo é composto por uma aplicação desenvolvida com a Android SDK, nomeada como IdentificaPet.

O IdentificaPet possui todas as funções necessárias para a identificação do animal, desde o cadastro do proprietário, cadastro do animal entre outras funcionalidades que se comunicam com o servidor web para busca e gravação de dados. Para um melhor entendimento, o diagrama de caso de uso, demonstra todas as funcionalidades da aplicação, cujo pode ser conferido na figura 16.

Figura 16 – Diagrama de casos de usos



Fonte: Do autor

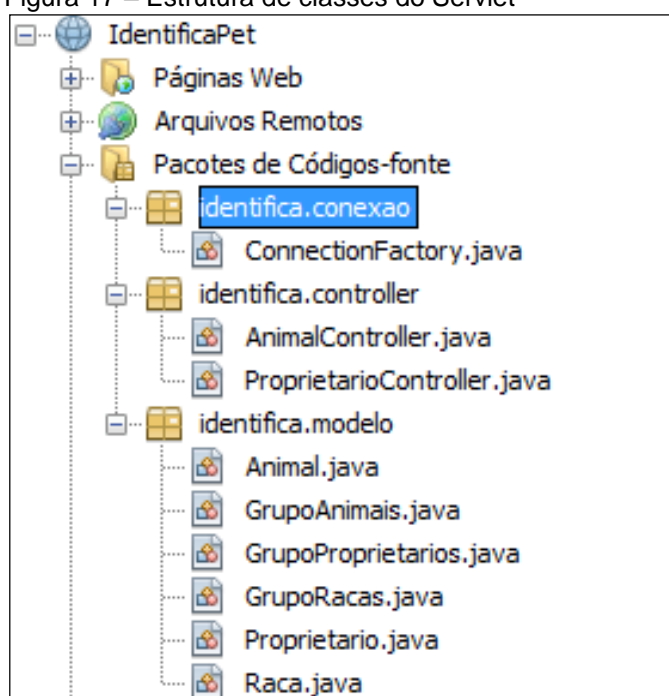
## 7.4 DESENVOLVIMENTO DO PROTÓTIPO

Após todas as ferramentas configuradas e instaladas corretamente se deu início ao desenvolvimento do protótipo do aplicativo de identificação. Inicialmente foi criada apenas uma aplicação de teste, onde se aplicou a funcionalidade da tecnologia NFC no Android. Após um bom entendimento se deu início ao projeto principal.

### 7.4.1 Estrutura do servidor web

A criação do Servlet foi realizada através do NetBeans juntamente com o servidor GlassFish na versão 4.0. O mesmo foi construído em Java através do Servlet classe padrão que fica sobre controle do container, nesse caso o servidor GlassFish. Na figura 17, pode-se conferir as classes desenvolvidas.

Figura 17 – Estrutura de classes do Servlet



Fonte: Do autor

A troca de dados entre a aplicação Android e o serviço acontece através de trocas de dados JSON ou através de confirmações sobre a operação. Quando o aplicativo Android faz uma requisição para listar os animais referentes a certo proprietário, o mesmo irá retornar um Array de JSON com todos os animais referentes

ao proprietário, na figura 18 podemos ver a função que realiza essa operação. Caso a aplicação Android faça a requisição de excluir um animal o serviço irá retornar para aplicação Android apenas uma string com o sucesso ou fracasso da operação.

Figura 18 – Função para conversão de dados para o formato JSON

```
private void listarAnimaisProp(HttpServletRequest request, HttpServletResponse response) {
    String sCodigo = (String)request.getParameter("codigo_proprietario");
    System.out.println("codigo proprietario: " + sCodigo);

    Integer codigo = Integer.valueOf(sCodigo);

    List<Animal> lista = GrupoAnimais.getAnimais(codigo);

    String listaAnimal = "{\"animais\":[";
    for (int i=0; i<lista.size(); i++) {
        Animal c = lista.get(i);
        if (i+1 < lista.size()) {
            listaAnimal += "{\"nome\":\"" + c.getNome() + "\", \"codigo\":\"" + c.getCodigo() + "\"},";
        }else{
            listaAnimal += "{\"nome\":\"" + c.getNome() + "\", \"codigo\":\"" + c.getCodigo() + "\"}";
        }
    }
    listaAnimal += "]}";
    try {
        PrintWriter out = response.getWriter();
        out.print(listaAnimal);
        out.flush();
    }catch(IOException ex) {

    }
}
```

Fonte: Do autor

Para a realização da conexão com o banco de dados foi criada a classe ConnectionFactory, com a configurações necessárias para a comunicação entre a aplicação web e o banco de dados. No caso da aplicação IdentificaPet o banco de dados utilizado foi o PostgreSql, na imagem 19 pode-se ver as configurações necessárias para essa comunicação.

Figura 19 – Função de comunicação entre servidor e banco de dados

```
package petshop.conexao;

import java.sql.Connection;
import java.sql.DriverManager;

public class ConnectionFactory {
    private static final String DRIVER = "org.postgresql.Driver";
    private static final String URL = "jdbc:postgresql://localhost/identificapet";
    private static final String USERNAME = "postgres";
    private static final String PASSWORD = "postgre";

    /**
     *
     * @return
     * @throws Exception
     */
    public static Connection getConnection() throws Exception {
        Class.forName(DRIVER);
        Connection conn = DriverManager.getConnection(URL, USERNAME, PASSWORD);
        return conn;
    }
}
```

Fonte: Do autor

Foi criada também uma interface simples, apenas para consulta dos proprietários cadastrados, que será utilizada apenas para testes.

Na estrutura da figura 17, também foram criadas classes de manipulação de dados e as duas classes Servlet chamada *AnimalController* e *ProprietarioController*, que estende a classe *HttpServlet*. Essa classe age como uma camada intermediária entre uma requisição realizada pelo aplicativo e o banco de dados. Através dessas classes, pode-se ler os dados que são passados como parâmetros pela aplicação Android, procurar as informações referente a requisição efetuada, formatar o resultado e retornar o mesmo para aplicação Android.

## 7.4.2 Aplicação Android

Para dar início ao desenvolvimento da aplicação mobile, foi definido no manifesto da aplicação as permissões para acesso aos hardwares necessários para execução do projeto. Para o IdentificaPet, foi necessário a liberação do hardware do NFC, através das entradas *android.permission.NFC*, do hardware da INTERNET, para possibilitar as comunicações com o servidor (*android.permission.INTERNET*) e também a permissão para *android.permission.CALL\_PHONE*, que permite ao usuário de clicar no número de telefone, onde abrirá a tela de chamada com o número para realizar a mesma.

Para que a aplicação não seja instalada em aplicativos que não possuem o NFC, também no manifesto definimos a condição na tag *uses-feature* informando *true* na tag *required*, fazendo assim que o Google Play não mostre o aplicativo para usuários sem suporte ao NFC.

Figura 20 – Configuração do manifesto

```

<uses-permission android:name="android.permission.NFC" />
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<uses-feature
    android:name="android.hardware.nfc"
    android:required="true" />

<application
    android:allowBackup="true"
    android:icon="@drawable/caogato"
    android:label="IdenticaPet"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".Principal.Login"
        android:label="IdenticaPet" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>

        <intent-filter>
            <action android:name="android.nfc.action.NDEF_DISCOVERED" />
            <category android:name="android.intent.category.DEFAULT" />
            <data android:mimeType="text/plain" />
        </intent-filter>
    </activity>
</application>

```

Fonte: Do autor

A primeira *activity* a ser iniciada é a *Login.class*, através da qual é possível acessar todas as funcionalidades do aplicativo, iniciando as *activities* *CadastroProprietario.class*, *LeTag.class* e *IdentificaTag.class*.

Na figura 21 podemos ver o layout da tela de *login.class* e na imagem 20 e a configuração para que a *activity Login* seja a *activity* principal da aplicação.

Figura 21 – Tela de login



Fonte: Do autor

Através do botão azul o proprietário do animal poderá realizar o seu cadastro, onde irá informar os seus dados que servirá de informação no momento de identificar o animal. Após o seu cadastro o mesmo poderá cadastrar um animal acessando o seu perfil através da tela de *login*.

Ao informar o seu e-mail e senha na tela acima, o aplicativo, irá realizar a requisição ao servidor, enviado como parâmetro usuário e senha, para verificar se o mesmo possui um cadastro.

Quando necessário a comunicação com o servidor, é chamada a função desejada da classe *AcessoServlet*, no caso do *login* é chamada a *LoginProprietario* que está dentro da classe *AcessoServlet*.

Todas as funções da classe *AcessoServlet* tem como parâmetro a *activity* e o objeto que implementa a interface, e os parâmetros necessários para passar ao servidor.

Inicialmente a classe *AcessoServlet* realizava diretamente a comunicação com o servidor, causando assim um travamento da tela no momento da comunicação. Para resolver esse problema foi criada uma classe chamada *AcessoServletAsync* que estende *AsyncTask*, permitindo realizar a requisição em segundo plano e publicar os resultados na interface, sem manipulação de threads.

A interface criada se chama *IProgressoCompleto* que possui a função *completouProgresso*, que retornará a *activity* o retorno do servidor.

A função da classe *AcessoServlet* passa para *AcessoServletAsync* todas as informações necessárias para a comunicação com o servidor, imagem 22, são atribuídas três informações.

Através da função *setUrl* é atribuída a URL para acessar o servidor, a função *setAcao* foi criada para atribuir a ação que se está realizando, já que uma *activity* pode realizar mais de uma requisição ao servidor, sendo necessário no momento do retorno saber qual requisição foi utilizada, já a função *execute* será passada os parâmetros e a mesma irá executar a chamada ao servidor.

Figura 22 – Função *loginProprietario* da classe *AcessoServlet*

```
void loginProprietario(Activity act, IProgressoCompleto callback, String Email, String Senha){
    try {
        AcessoServletAsync servletAsync = new AcessoServletAsync(act, callback);

        servletAsync.setUrl("http://" + ipv4 + "/IdentificaPet/login");
        servletAsync.setAcao("LOGIN");
        servletAsync.execute("email", Email, "senha", Senha.toString());

    } catch (Exception e){
    }
}
```

Fonte: Do autor

Para que o *ProgressDialog* fosse exibido na interface enquanto o acesso ao Servlet era realizado, foi necessário que a classe *AcessoServletAsync* (esta, herdada de *AsyncTask*) possuísse um construtor que se especificasse a *Activity* em que o *ProgressDialog* apareceria e o objeto que implementa a interface *IProgressoCompleto*, para que a função de *callback* fosse chamada assim que a

requisição no Servlet estivesse terminada. Imagem 23.

Figura 23 – Classe AcessoServletAsync

```
public class AcessoServletAsync extends AsyncTask<String, Void, String> {
    private ProgressDialog progressDialog;
    private String url;
    private String acao;
    private IProgressoCompleto callback;

    public AcessoServletAsync(Activity act, IProgressoCompleto callback){
        progressDialog = new ProgressDialog(act);
        this.callback = callback;
    }

    @Override
    protected String doInBackground(String... params) {...}

    @Override
    protected void onPreExecute() {...}

    @Override
    protected void onPostExecute(String result) {
        if (progressDialog.isShowing()) {
            progressDialog.dismiss();
        }
        callback.completouProgresso(result, getAcao());
    }
}
```

Fonte: Do autor

Para realizar a leitura da *tag* NFC é importante definir no manifesto a versão mínima SDK de nível 10, pois o NFC é suportado apenas pela versão 2.3.3. Na figura 24 a configuração do *targetSdkVersion* e do *minSdkVersion*.

Figura 24 – Versão SDK

```
<uses-sdk
    android:targetSdkVersion="19"
    android:minSdkVersion="10"/>
```

Fonte: Do autor

No manifesto foi definido também o filtro para quando o smartphone identificar uma *tag* NFC, seja aberto automaticamente o aplicativo IdentificaPet, caso o celular possua mais de um aplicativo para leitura de tags, será aberto uma janela para que o usuário possa escolher qual deles utilizar, na figura 25, configuração do filtro.

O filtro deve ficar dentro da *activity* que se deseja abrir do aplicativo, no nosso caso será aberta a de login. A tag *data android:mimeType* serve para identificar o tipo de dados que se está procurando na tag NFC, no caso do aplicativo IdentificaPet o tipo de dados gravada na tag é do tipo texto.

Na tag *action android:name* é definido o tipo da ação será realizada pelo filtro, que nesse caso foi definido tag com dados NDEF\_DISCOVERED. A tag *category android:name="android.intent.category.DEFAULT"* serve para receber intenções implícitas, caso não declarado no filtro de intenções, nenhuma intenção implícita retornará a sua *activity*.

Figura 25 – Configuração do filtro NFC

```
<intent-filter>
  <action android:name="android.nfc.action.NDEF_DISCOVERED" />
  <category android:name="android.intent.category.DEFAULT" />
  <data android:mimeType="text/plain" />
</intent-filter>
```

Fonte: Do autor

O modo de operação escolhido para implementação foi o modo passivo, onde apenas o dispositivo irá gerar um sinal de radiofrequência para a tag, que será alimentada pelo mesmo e irá transmitir os dados do tipo NDEF para a aplicação.

Para realizar um cadastro de um animal primeiramente devemos identificar a tag ao qual será associado esse animal, isso para poder identificar o ID da tag para gravar no banco de dados. Para isso foi criada uma *activity* chamada *IdentificaTag*, para realizar essa operação.

Antes de abrir a *activity* que irá identificar a tag NFC, deve-se verificar se o NFC está ativo no celular, para isso é necessário a importação da classe *android.nfc.NfcAdapter* que contém as informações referente a tecnologia NFC do dispositivo utilizado.

Para buscar o estado do *NfcAdapter* após a importação da biblioteca, basta utilizar a função *getDefaultAdapter* passando como parâmetro *activity* atual, figura 26.

Figura 26 – Verificação se o NFC está ativo no dispositivo

```

nfcAdapter = NfcAdapter.getDefaultAdapter(ConsultaAnimal.this);

NFCFuncao ClasseNFC = new NFCFuncao();

lbNFCAtivo = ClasseNFC.VerificaNFCAtivo(nfcAdapter);
if (!lbNFCAtivo){
    Toast.makeText(ConsultaAnimal.this, "Favor ativar o NFC antes de cadastrar um animal.", Toast.LENGTH_SHORT).show();
}else {

```

Fonte: Do autor

Para evitar a duplicidade de código, foi criada a função *VerificaNFCAtivo* para verificar se o mesmo existe no celular e se está ativo, o parâmetro passado é o *NfcAdapter* e o retorno é uma *Boolean* onde será informado se o mesmo está ativado ou não, figura 27.

Figura 27 – Função VerificaNFCAtivo

```

Boolean verificaNFCAtivo(NfcAdapter nfcAdapter) {
    if (nfcAdapter != null && nfcAdapter.isEnabled()) {
        return true;
    }else{
        return false;
    }
}

```

Fonte: Do autor

Após verificar que o NFC está ativo corretamente, foi criado um *PendingIntent* que será passado ao método *enableForegroundDispatch*, imagem 28.

Este método deverá ser chamado dentro do thread principal e apenas quando a *activity* estiver em primeiro plano, ou seja, depois que o *OnResume()* tiver sido chamado.

Figura 28 – Método enableForegroundDispatch

```

protected void onResume() {
    super.onResume();

    enableForegroundDispatchSystem();
}

@Override
protected void onPause() {...}

@Override
protected void onNewIntent(Intent intent) {...}

private void enableForegroundDispatchSystem(){

    Intent intent = new Intent(this, IdentificaTag.class).addFlags(Intent.FLAG_RECEIVER_REPLACE_PENDING);

    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, 0);
    IntentFilter[] intentFilter = new IntentFilter[]{};

    nfcAdapter.enableForegroundDispatch(this, pendingIntent, intentFilter, null);
}

```

Fonte: Do autor

O *enableForegroundDispatch* tem como finalidade dar prioridade a aplicação ativa, caso contrário o sistema android irá mostrar uma lista com todas os aplicativos que tem a capacidade de ler uma *tag* NFC.

Após *activity* deixar o primeiro plano logo após o *OnPause()*, é chamado o método *disableForegroundDispatch* para retirar a prioridade da mesma, figura 29.

Figura 29 – Função para verificar se o NFC está ativo

```

protected void onPause() {
    super.onPause();
    disableForegroundDispatchSystem();
}

@Override
protected void onNewIntent(Intent intent) {...}

private void enableForegroundDispatchSystem() {...}

private void disableForegroundDispatchSystem() {
    NfcAdapter.getDefaultAdapter(this).disableForegroundDispatch(this);
}

```

Fonte: Do autor

Assim que o sistema identifica a *tag* NFC, é chamado o método *onNewIntent()*, que irá ter como parâmetro a *intent* criada e passada no método *enableForegroundDispatch*.

Nesse método através da função *getParcelableExtra()* pode-se obter todas as informações referente a *tag* NFC, no caso do *IdentificaPet* a informação que precisamos é o *id* da *tag*, que será obtido através da função *getNfcAdapterExtraID(intent)* que irá retornar o *ID* da *tag* NFC como podemos ver na codificação da figura 30 .

Após a identificação do *ID* da *tag* NFC o sistema irá passar o código para o cadastro do animal ou pegará esse código para realizar a identificação do mesmo.

Figura 30 – Método de reconhecimento da *tag*

```
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    if(intent.hasExtra(NfcAdapter.EXTRA_TAG)) {
        String lsID;
        lsID = getNfcAdapterExtraID(intent);

        if (Proprietario.equals("0")){
            Intent intentAnimal = new Intent(IdentificaTag.this, IdentificaAnimal.class);
            intentAnimal.putExtra("codi_nfc", lsID);
            startActivity(intentAnimal);
        }else {
            Intent intentAnimal = new Intent(IdentificaTag.this, CadastroAnimal.class);
            intentAnimal.putExtra("proprietario", Proprietario);
            intentAnimal.putExtra("id_animal", lsID);
            startActivity(intentAnimal);

            Toast.makeText(this, lsID, Toast.LENGTH_SHORT).show();
        }
    }
}
```

Fonte: Do autor

Após o usuário inserir os dados e grava-los no banco de dados, será necessário aproximar novamente a *tag* para poder gravar dados na mesma, para que mesmo se o usuário não possua internet, a identificação do animal possa ser realizada, trazendo apenas os dados essenciais.

Os dados a serem gravados na *tag* são o nome do animal, nome e telefone do proprietário. Para isso foi criada a *activity LeTag.class()*, os métodos *OnResume()*, *OnPause()* são iguais ao da *activity IdentificaTag*, a única diferença ocorre no método *onNewIntent()* que será explicado a seguir.

Para gravar os dados na *tag* mais algumas bibliotecas foram necessárias, como *android.nfc.NdefMessage*, *android.nfc.NdefRecord*, *android.nfc.Tag*, *android.nfc.tech.Ndef* e *android.nfc.tech.NdefFormatable*.

Através da função *getParcelableExtra*, é possível pegar todos os dados referente a tag. Porém antes de pegar os dados necessários para gravar um texto na tag é verificado se o ID é o mesmo da tag inicial, isso para evitar que sejam gravados dados na tag errada. Após essa verificação e pegar os dados da tag será montado a string que será gravada na mesma.

A mesma é criada através da função *createNdefMessage* disponibilizada pelo framework do Android figura 31.

Figura 31 – Método para realizar a gravação de dados na tag

```
protected void onNewIntent(Intent intent) {
    super.onNewIntent(intent);
    if(intent.hasExtra(NfcAdapter.EXTRA_TAG)) {
        Toast.makeText(this, "Tag identificada!", Toast.LENGTH_SHORT).show();
        String lsID, lsIDAux;
        lsID = getNfcAdapterExtraID(intent);
        lsIDAux = lsID.substring(2, 16);
        lsIDAux = lsIDAux.toUpperCase();

        if (Tag.equals(lsIDAux)){
            android.nfc.Tag tag = intent.getParcelableExtra(NfcAdapter.EXTRA_TAG);
            String lstag = NomeAnimal + "-" + Proprietario + "-" + TelefoneProprietario;
            NdefMessage ndefMessage = createNdefMessage(lstag + "");

            writeNdefMessage(tag, ndefMessage);

            Intent intentAnimal = new Intent(GravarTag.this, ConsultaAnimal.class);
            intentAnimal.putExtra("proprietario", Codi_Proprietario);
            startActivity(intentAnimal);
        }else {
            Toast.makeText(this, "A TAG NFC deve ser a mesma informada no momento inicial do cadastro.",
        }
    }
}
```

Fonte: Do autor

Na função *writeNdefMessage* será passado como parâmetro a tag encontrada e a mensagem no formato *NdefMessage* que será gravada na mesma. Para gravar os dados na tag, primeiramente deve-se chamar o *connect()*, após a conexão o sistema irá verificar se a tag identificada é gravável, e em seguida é chamada a função *writeNdefMessage* da biblioteca *android.nfc.tech.Ndef* que irá gravar a *ndefMessage* na tag, figura 32.

Figura 32 – Método de conexão para gravação da *tag*

```

private void writeNdefMessage(Tag tag, NdefMessage ndefMessage){
    try{
        if(tag == null)
        {
            Toast.makeText(this, "Tag object cannot be null", Toast.LENGTH_SHORT).show();
            return;
        }
        Ndef ndef = Ndef.get(tag);

        if(ndef == null){
            formatTag(tag, ndefMessage);
        }else{
            ndef.connect();
            if(!ndef.isWritable()){
                Toast.makeText(this, "Tag is not writable", Toast.LENGTH_SHORT).show();
                ndef.close();
                return;
            }
            ndef.writeNdefMessage(ndefMessage);
            ndef.close();
        }
    }catch (Exception e)
    {
        Log.e("writeNdefMessage", e.getMessage());
    }
}

```

Fonte: Do autor

Após a gravação do registro na *tag* o sistema irá disponibilizar uma lista com todos os animais cadastrados para o proprietário corrente.

A identificação da *tag* utiliza a mesma *activity* (IdentificaTag) para identificar a tag no momento do cadastro do animal e possui o mesmo funcionamento com uma única diferença. Após identificar a ID da tag, através de uma classe *AsyncTask*, aplicação conecta-se ao servidor buscando pelo ID identificado, caso esta *tag* possua um cadastro no banco de dados será exibido os dados do animal e de seu proprietário, possibilitando assim a identificação.

Caso o usuário não esteja conectado na internet também será possível identificar a tag, porém apenas os dados básicos como o nome do animal, nome e telefone do proprietário.

### 7.4.3 Funcionamento do sistema

O fluxo do protótipo desenvolvido se dá início na tela de login figura 33, a mesma é composta pelo campo de acesso restrito a apenas usuários já cadastrados no sistema, opção para realizar o cadastro do proprietário e acesso para identificar o animal. O login é efetuado através do preenchimento do e-mail e senha cadastrados através do botão “CADASTRE-JÁ”.

Figura 33 – Tela de início do protótipo

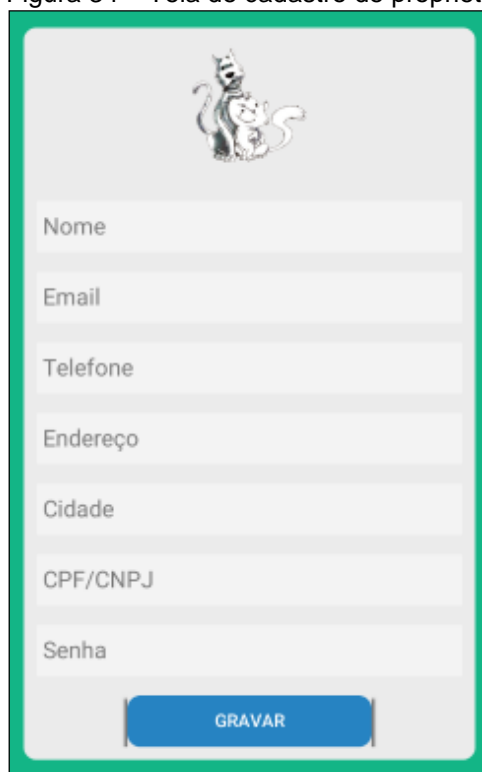
A tela de início do protótipo, intitulada "Identifica Pet", apresenta um cabeçalho verde com o título em branco. Abaixo do cabeçalho, há uma ilustração de dois gatos, um maior e um menor, ambos com coleiras. O formulário de login contém dois campos de entrada: o primeiro para o e-mail, com o texto "juliaderochi@hotmail.com" preenchido, e o segundo para a senha, com pontos para ocultar o conteúdo. Abaixo dos campos, há três botões: um verde com o texto "ENTRAR", um azul com o texto "NÃO TEM UM CADASTRO? CADASTRE-JÁ", e um vermelho com o texto "IDENTIFICAR ANIMAL".

Fonte: Do autor

O cadastro de usuário, que seria o proprietário do animal, possui campos com informações que será relevante no momento da identificação do animal, figura 34.

Após realizar o cadastro o sistema irá levar o usuário direto para a tela de consulta de animais cadastrados, onde também é possível cadastrar um novo animal, excluir e editar os dados de um animal já cadastrado.

Figura 34 – Tela de cadastro de proprietário

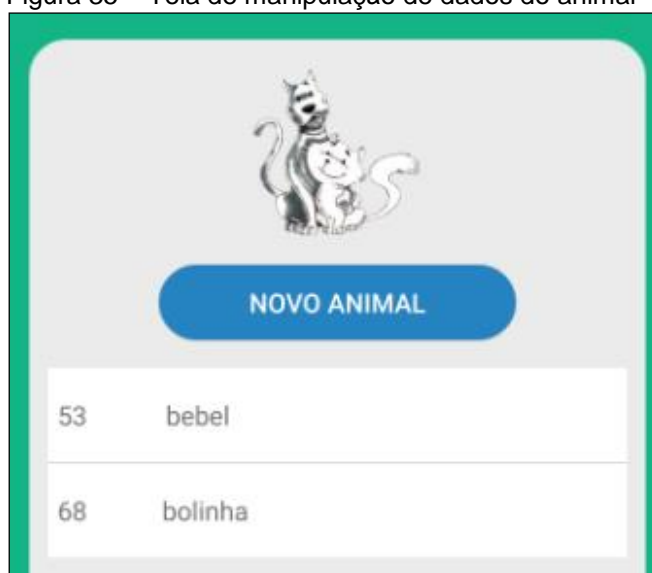


A tela de cadastro de proprietário apresenta um formulário com campos para Nome, Email, Telefone, Endereço, Cidade, CPF/CNPJ e Senha. No topo, há um ícone de dois gatos. Um botão azul com o texto 'GRAVAR' está localizado na base do formulário.

Fonte: Do autor

Para realizar a exclusão de um animal basta ficar segurando o mesmo por alguns segundos, fazendo assim a exclusão automática do registro do banco de dados. Para editar os dados do animal, basta clicar uma vez em cima do mesmo, onde abrirá a tela de cadastro com os dados já cadastrados, possibilitando assim a edição podemos ver o layout desta tela na figura 35.

Figura 35 – Tela de manipulação de dados do animal



A tela de manipulação de dados do animal mostra um ícone de dois gatos no topo. Abaixo dele, há um botão azul com o texto 'NOVO ANIMAL'. Abaixo do botão, há uma lista de animais com duas colunas: uma para o número de identificação e outra para o nome. Os dados exibidos são:

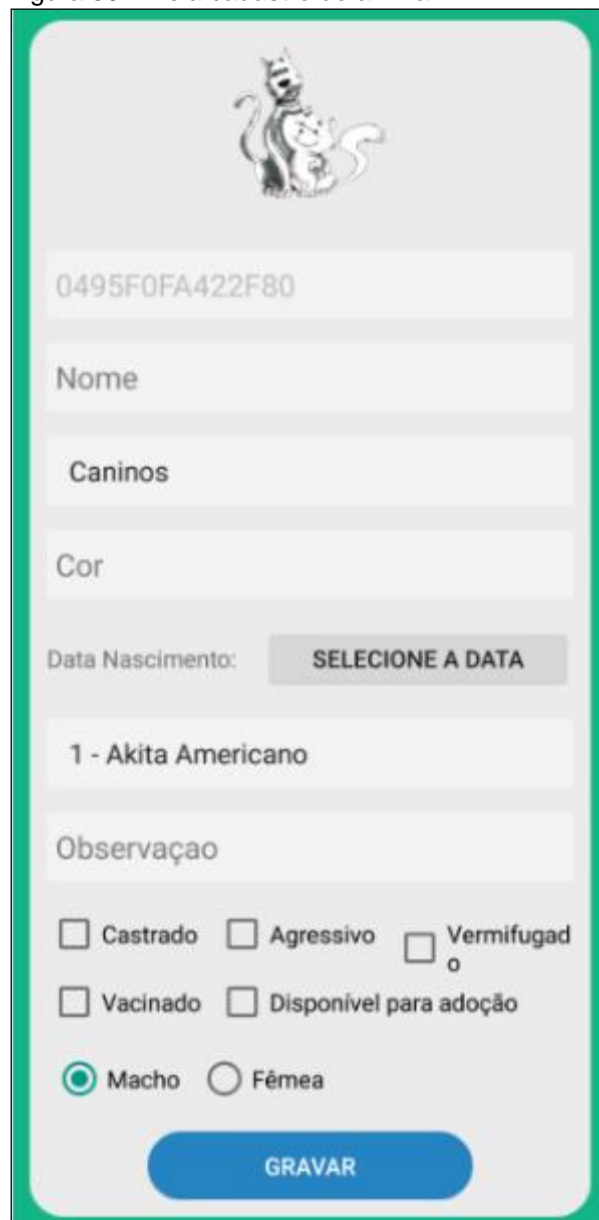
53	bebel
68	bolinha

Fonte: Do autor

Ao clicar no botão “Novo Animal” o usuário será levado até uma tela onde necessita a identificação da *tag* NFC para realizar o cadastro do animal.

Após a identificação da *tag* será aberto a tela de cadastro onde já estará disponível apenas para visualização a informação referente ao ID da *tag* NFC, os demais campos ficaram disponível para o preenchimento das informações, figura 36.

Figura 36 – Tela cadastro do animal



A imagem mostra a interface de usuário para o cadastro de um animal. No topo, há um ícone de um gato e um cachorro. Abaixo, um campo de texto exibe o ID da tag NFC: 0495F0FA422F80. Seguem campos para Nome, Caninos, Cor e Data Nascimento (com um botão "SELECIONE A DATA"). A raça selecionada é "1 - Akita Americano". Há um campo de Observação e uma seção de opções com caixas de seleção para: Castrado, Agressivo, Vermifugado, Vacinado, Disponível para adoção, Macho (selecionado) e Fêmea. Um botão azul "GRAVAR" está na base.

Fonte: Do autor

Para o campo “data”, foi criado um *DatePickerDialog* que abrirá um calendário assim que clicar no botão “SELECIONE A DATA” onde o usuário poderá selecionar a data desejada, figura 37.

Figura 37 – Calendário para selecionar a data



Fonte: Do autor

As raças disponíveis vêm diretamente do banco de dados, onde foram cadastradas com antecedência pelo desenvolvedor, as raças ficam dentro de *Spinner* onde será possível escolher a raça desejada, figura 38.

Caso a raça não esteja cadastrada pode ser escolhido a opção outros. Após gravar os dados o usuário deverá aproximar novamente a *tag* ao celular para que os dados possam ser gravados internamente da mesma, possibilitando assim a identificação mesmo sem conexão à internet.

Figura 38 – Spinner com as raças de animais

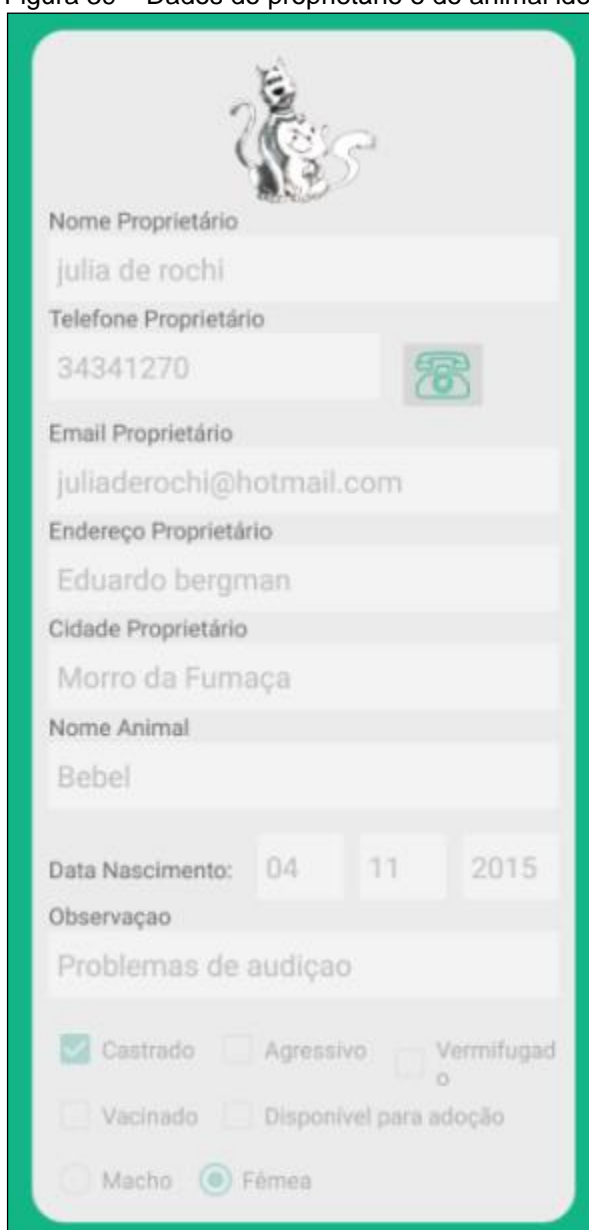


Fonte: Do autor

Para realizar a identificação basta o usuário selecionar o botão “IDENTIFICAR ANIMAL” da tela de login onde será necessário o usuário aproximar a *tag* próximo ao animal para realizar a identificação do mesmo.

Após identificação o usuário poderá visualizar as informações referente ao animal encontrado e também dados do proprietário do animal pode-se ver os dados cadastrados para o animal “Bebel”.

Figura 39 – Dados do proprietário e do animal identificado



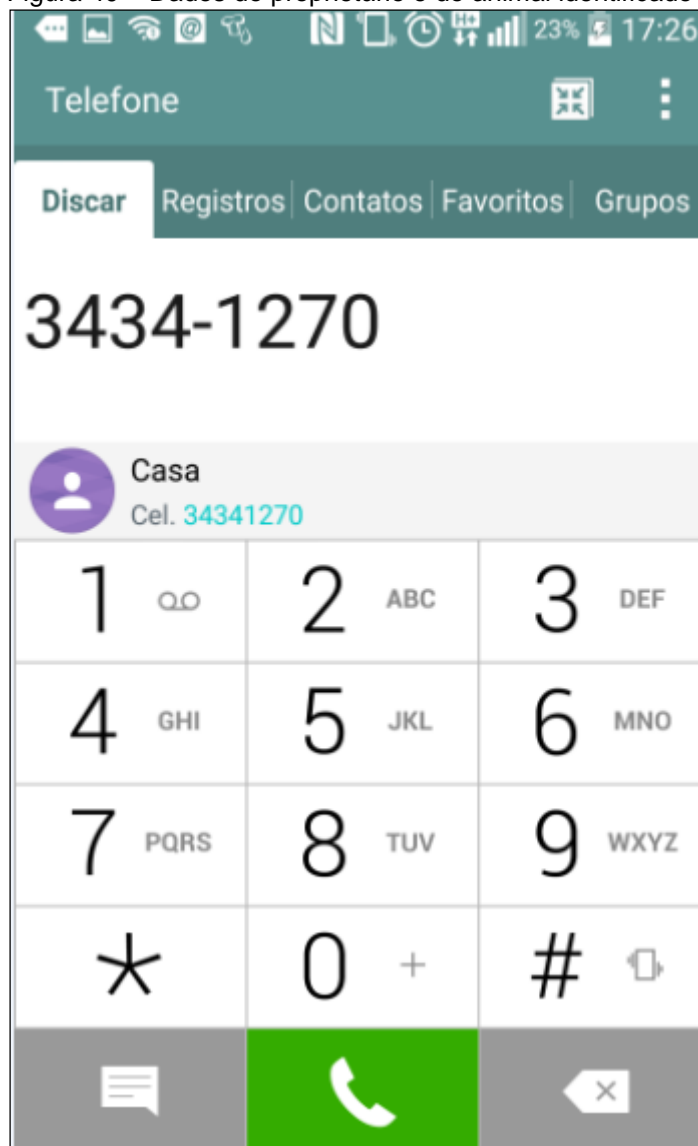
A imagem mostra uma interface de usuário com um formulário de dados. No topo, há um ícone de dois gatos. Abaixo, o formulário contém os seguintes campos:

- Nome Proprietário: julia de rochi
- Telefone Proprietário: 34341270 (acompanhado de um ícone de telefone)
- Email Proprietário: juliaderochi@hotmail.com
- Endereço Proprietário: Eduardo bergman
- Cidade Proprietário: Morro da Fumaça
- Nome Animal: Bebel
- Data Nascimento: 04 / 11 / 2015
- Observação: Problemas de audição
- Formulário de características com opções de seleção:
  - Castrado
  - Agressivo
  - Vermifugado
  - Vacinado
  - Disponível para adoção
  - Macho
  - Fêmea

Fonte: Do autor

Na figura 40, verifica-se a tela que será mostrada, caso o usuário deseje ligar para o proprietário do animal, basta clicar na imagem do telefone que o sistema irá levar o número diretamente para a tela de chamada facilitando assim o contato entre usuário e proprietário do animal.

Figura 40 – Dados do proprietário e do animal identificado



Fonte: Do autor

## 7.5 RESULTADOS OBTIDOS

A partir da etapa de levantamento de requisitos realizada neste trabalho, foi possível levantar material referencial importante sobre todos os temas tratados na implementação do sistema proposto, sendo esse seu principal objetivo.

Foram destacados no projeto o grande índice de abandono de animais e a importância se todos os animais pudessem ser identificados. Foram estudados todos os métodos de identificação e pesquisado qual seria o melhor método para identificação de um animal, sendo esse o microchip por ser um método indolor, permanente e de baixo custo.

Outro levantamento importante que consta nessa pesquisa é o estudo da tecnologia NFC, descrevendo primeiramente a tecnologia RDIF que é de onde deriva o NFC continuando com seu surgimento do NFC, características, modo de operação, frequência utilizada, seguindo com as características de um *tag* NFC.

Através da fundamentação teórica foi possível também compreender o funcionamento da linguagem de programação Android facilitando e possibilitando o desenvolvimento da pesquisa.

A aplicação do funcionamento do NFC foi muito satisfatória, sendo que existe muita documentação para compreender seu funcionamento. Não foram encontrados trabalhos iguais ou similares a esse, o qual a *tag* utilizada fosse a microchip NFC, deixando uma dúvida no começo da fundamentação sobre seu real funcionamento, que, porém, foram todas esclarecidas e compreendidas durante o desenvolvimento do projeto.

Os testes foram realizados em apenas um animal, a *tag* foi implantada no dorso do animal por sugestão do veterinário. A implantação foi realizada com sucesso e sem dificuldades, o animal não demonstrou dor após a aplicação da *tag*, apenas no momento devido a grossura da agulha.

Após a implantação podemos sentir a *tag* no animal, a mesma não fica sempre no ponto aplicado, podendo migrar um pouco, porém não muito distante do local da aplicação.

Foram realizados vários testes de leitura, após posicionar o telefone próximo ao local onde se encontra a *tag* conseguimos após algumas tentativas a identificação da mesma. O posicionamento do celular também influencia na identificação.

Após a leitura da *tag*, os dados referentes ao proprietário e ao animal foram demonstrados com sucesso para o usuário final.

Para o tema proposto por este trabalho de conclusão de curso, os resultados obtidos foram satisfatórios em relação aos objetivos apresentados. O objetivo principal que se trata do desenvolvimento de um protótipo de aplicativo

Android capaz de identificar *tag* NFC aplicando o conhecimento obtido sobre o seu funcionamento e cadastrar os dados referente ao animal e seu proprietário na plataforma Android, contanto com o auxílio do servidor desenvolvido que serve como comunicador entre a aplicação e o banco de dados foi alcançado com sucesso.

## 8 CONCLUSÃO

O número de animais abandonados está crescendo cada dia mais, aumentando assim o número de animais em ONG's e canis, que não estão dando conta de manter todos esses animais, pois o custo está cada vez mais elevado.

Diante desta situação a identificação de animais se torna essencial para que ONG's ou outros órgãos responsáveis possam ter reconhecimento do proprietário e informações importantes para o bem-estar do animal.

Diante deste problema foi desenvolvido o protótipo de aplicativo para a identificação de *tag* NFC através da aplicação Android.

Durante o desenvolvimento do projeto foram encontrados alguns obstáculos. Um dos maiores problemas foi encontrar a *tag* NFC na frequência correta, já que o NFC pede uma frequência de 13.56 MHz, não sendo flexível. A obtenção da *tag* se teve após a pesquisa em um site chinês e em contato diretamente com o fornecedor, se pode encontrar a correta *tag* que levou em média de três meses para chegar ao Brasil. O custo da *tag* não foi um problema já que seu valor não ultrapassou dez reais, tendo em vista que hoje um pet shop no Brasil cobra em média cem reais para inserir a *tag* no animal.

Durante as fases de testes foi constatado que a distância da *tag* e do celular deve ser menor que 10 cm, dificultando assim o reconhecimento da mesma. O pelo do animal também dificultou a leitura da *tag*, porém não impossibilitou a identificação. A *tag* foi implantada na parte do dorso superior, porém após os testes e verificando a dificuldade na identificação, juntamente a um veterinário, ficou definido que seria melhor a implementação da *tag* no ombro, por ser uma superfície lisa que facilitaria a sua localização e identificação.

Os demais testes realizados, foram com a *tag* dentro de um envelope, para assim simular a pele do animal.

Algumas dificuldades foram encontradas para o desenvolvimento da aplicação Android, como a função para o não travamento da tela durante a consulta com servidor, sendo assim necessária a criação de funções que modificaram a lógica de programação do protótipo.

Outros problemas foram encontrados durante o desenvolvimento da aplicação, mas foram se resolvendo à medida que os estudos se aprofundavam.

Sendo assim o objetivo geral e os objetivos específicos dessa pesquisa

foram todos atingidos com êxito, tendo em vista que foi possível a identificação da *tag* NFC através de uma aplicação Android.

Este trabalho abre uma porta para a sociedade e para os meios públicos, facilitando a identificação de animais perdidos ou abandonados, podendo também ter um controle sobre a situação dos animais em cada cidade, caso a identificação fosse aplicada corretamente.

O trabalho também foi submetido como proposta para o projeto Sinapse da inovação<sup>4</sup> na categoria de tecnologia social para o setor de administração pública, já que o mesmo possui uma relevância social muito grande para a sociedade.

Diante da importância da identificação do animal, diversos pontos surgiram durante todo o processo de execução do trabalho, que podem ser utilizados como sugestão de novos objetivos ao longo de novas pesquisas nesta área. Alguns destes pontos são:

- a) Criação de uma web service mais robusto, com um banco de dados e arquitetura de maior confiabilidade;
- b) Inserir a funcionalidade de buscar a localização do celular, permitindo assim no momento da leitura da *tag* o envio de um e-mail automático para o dono do animal, informando a localização ao qual foi realizada a leitura da *tag*;
- c) Criação de uma aplicação Web para que os órgãos públicos possam ter um maior controle sobre os animais cadastrados;

Por fim conclui-se que o trabalho teve realmente o resultado esperado, atingindo todos os itens propostos com a conclusão do desenvolvimento do protótipo de identificação de animais utilizando a tecnologia NFC.

---

<sup>4</sup> IdentificaPet - Programa de identificação de animais utilizando NFC, segue link para acesso a proposta submetida. <http://sc5.sinapsedainovacao.com.br/sc5/ideia/identificapet-programa-de-identificacao-de-animais-utilizando-nfc>

## REFERÊNCIAS

- ABINPET. Associação Brasileira da Indústria de Produtos para animais de Estimação (São Paulo). **Aves são os animais que menos pesam no orçamento familiar**. 2012. Disponível em: <<http://abinpet.org.br/imprensa/noticias/abinpet-divulga-media-de-gasto-com-pets/>>. Acesso em: 03 abr. 2015.
- ABLESON, F. SEN, R. **Android in Action**. Greenwich, Conn: Manning Publications, 2011. Disponível em: <<http://www.lcc.ufrn.br/~thiagoalves/minicurso-android/android%20in%20action%203rd%20edition.pdf> >Acesso em: 12 jun. 2013.
- ALECRIM, Emerson. **O que é NFC (Near Field Communication)?** 2012. Disponível em: <<http://www.infowester.com/nfc.php>>. Acesso em: 05 maio 2015.
- ALVES, Joel. Chip. 2008. Disponível em: <<http://chipdabesta.blogspot.com.br/>>. Acesso em: 05 maio 2015.
- ANDROID DEVELOPERS. **Near Field Communication**. Disponível em: <<http://developer.android.com/guide/topics/connectivity/nfc/index.html>>. Acesso em: 27 abr. 2015.
- ANDROID DEVELOPERS. **Activity**. Disponível em: <<http://developer.android.com/reference/android/app/Activity.html>>. Acesso em: 14 jun. 2015a.
- ANDROID DEVELOPERS. **Advanced NFC**. Disponível em: <<https://developer.android.com/guide/topics/connectivity/nfc/advanced-nfc.html>>. Acesso em: 14 jun. 2015b.
- ANDROID DEVELOPERS. **NFC Basics**. Disponível em: <<http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#p2p>>. Acesso em: 14 jun. 2015c.
- ANDROID DEVELOPERS. **Host-based Card Emulation**. Disponível em: <<http://developer.android.com/guide/topics/connectivity/nfc/hce.html>>. Acesso em: 10 jun. 2015d.
- BARROS, Thiago. **O que é smartphone e para que serve?** 2012. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2011/12/o-que-e-smartphone-e-para-que-serve.html>>. Acesso em: 19 jun. 2015.
- BURNETTE, E. **Hello, Android: Introducing Google's Mobile Development Platform**. 3. ed. Raleigh, N.C: Pragmatic Bookshelf, 2010.
- CAELUM (São Paulo). **APOSTILA JAVA PARA DESENVOLVIMENTO WEB: Servlets**. 2015. Disponível em: <<http://www.caelum.com.br/apostila-java-web/>>. Acesso em: 19 out. 2015.
- CERAMI, Ethan. **Web Services Essentials**. Sebastopol: O'reilly, 2002.

CLARK, S. **One in three mobile phones to come with NFC** by 2017. Disponível em: <<http://www.nfcworld.com/2013/06/05/324448/one-in-three-mobile-phones-to-come-with-nfc-by-2017/>>. Acesso em: 5 abr. 2014.

CONSTANTINO, Ederaldo de Oliveira; NERIS, Vânia Paula de Almeida. Acessibilidade em Bibliotecas: utilizando NFC e Smartphones para auxiliar Deficientes Visuais na Identificação de Livros. **T.i.s. São Carlos**, São Carlos, v. 8, n. 2, p.144-150, 01 ago. 2014. Disponível em: <<http://revistatis.dc.ufscar.br/index.php/revista/article/download/84/78>>. Acesso em: 14 jun. 2015.

COSKUN. **Identification methods for dogs and cats**. 2008. Disponível em: <[http://www.icam-coalition.org/downloads/Identification methods for dogs and cats.pdf](http://www.icam-coalition.org/downloads/Identification%20methods%20for%20dogs%20and%20cats.pdf)>. Acesso em: 30 set. 2014.

COSKUN, V.; OK, K.; OZDENIZCI, B. **Near Field Communication: From theory to practice**. West Sussex, Reino Unido: John Wiley & Sons Ltd, 2012.

\_\_\_\_\_. **NFC Application Development for Android**. West Sussex, Reino Unido: John Wiley & Sons Ltd, 2013.

PREFEITURA MUNICIPAL DE CURITIBA. **Lei nº 13.914**, de 23 de dezembro de 2011.

DE PAULA, Patricia Madureira Castro. **Estratégias adicionais no controle populacional de cães de rua**. 2010. 72 f. Dissertação (Mestrado) - Curso de Ciências Veterinárias, Medicina Veterinária, Universidade Federal do Paraná, Curitiba. Disponível em: <[http://dspace.c3sl.ufpr.br/dspace/bitstream/handle/1884/23726/ESTRATEGIAS ADICIONAIS NO CONTROLE POPULACIONAL DE CAES DE RUA.pdf?sequence=1](http://dspace.c3sl.ufpr.br/dspace/bitstream/handle/1884/23726/ESTRATEGIAS_ADICIONAIS_NO_CONTROLE_POPULACIONAL_DE_CAES_DE_RUA.pdf?sequence=1)>. Acesso em: 03 abr. 2015.

DÓRIA F. **A Identificação de animais de companhia**. Disponível em: <[http://www.vetbiblios.pt/ARTIGOS\\_TECNICOS/Diversos/IDENTIFICACAO\\_ANIMAIS\\_COMPANHIA.pdf](http://www.vetbiblios.pt/ARTIGOS_TECNICOS/Diversos/IDENTIFICACAO_ANIMAIS_COMPANHIA.pdf)>. Acesso em: 04 out. 2014.

EMARKETER. **2 Billion Consumers Worldwide to Get Smart(phones) by 2016 - See more at: <http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694#sthash.oFQMAHdv.dpuf>**. 2014. Disponível em: <<http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694>>. Acesso em: 20 jun. 2015.

FAWC. **Report on the Welfare of Farmed Animals at Slaughter or Killing**. 2003. Disponível em: <<http://www.fawc.org.uk/reports/pb8347.pdf>>. Acesso em: 19 nov. 2014.

FINKENZELLER, K. **RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication**. 3. ed. West Sussex, Reino Unido: John Wiley & Sons Ltd, 2010.

FIOCCO, Alessandro. **População de gatos crescerá 8% em 2012 e a de cães 4%**.

2012. Disponível em: <<http://f5.folha.uol.com.br/bichos/1122747-populacao-de-gatos-crescera-8-em-2012-a-de-caes-4.shtml>>. Acesso em: 15 out. 2014.

GARGENTA, M. **Learning Android**. Sebastopol, Calif: O'Reilly Media, 2011.

GARGENTA, Marko; NAKAMURA, Masumi. **Learning Android: Develop Mobile Apps Using Java and Eclipse**. 2. ed. Gravenstein Highway North, Sebastopol: O'reillynMedia, 2014, p. 270.

GHIRON, S. L. et al. **NFC Ticketing: A Prototype and Usability Test of an NFC-Based Virtual Ticketing Application**. In: FIRST INTERNATIONAL WORKSHOP ON NEAR FIELD COMMUNICATION (NFC '09). Hagenberg, Áustria

GLAUBER, Nelson. **Dominando o Android: Do Básico ao Avançado**. São Paulo: Novatec Editora, 2015. 792 p.

GLOBO. **Brasileiros têm 52 milhões de cães e 22 milhões de gatos, aponta IBGE**. 2015. Disponível em: <<http://g1.globo.com/natureza/noticia/2015/06/brasileiros-tem-52-milhoes-de-caes-e-22-milhoes-de-gatos-aponta-ibge.html>>. Acesso em: 21 jun. 2015.

GOMES, Daniel Adorno. **Web Services SOAP em JAVA: Guia prático para o desenvolvimento de web services em Java**. 2. ed. São Paulo: Novatec Editora, 2014.

GRATTON, D. A. Developing Practical Wireless Applications. In: Near Field Communications: The Smart Choice for Enabling Connectivity. Burlington (EUA): Digital Press, 2007. p. 216–224.

GUIMARÃES, Pereira Saulo. **Brasil é o quarto país do mundo em número de smartphones**. Exame. maio. 2013. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/brasil-e-o-quarto-pais-do-mundo-em-numero-de-smartphones>> Acesso em: 10 maio 2015.

HEITOR, Victor. **O que é celular com NFC?**. 2013. Disponível em: <<http://www.heitorvictorio.com.br/celular-com-nfc-near-field-communication/>>. Acesso em: 01 out. 2014.

IBNLIVE. **Android 5.0 Lollipop: 10 highlights of the latest version of Google's Android operating system**. Disponível em: <<http://www.ibnlive.com/news/india/android-5-0-lollipop-10-highlights-of-the-latest-version-of-googles-android-operating-system-720503.html>>. Acesso em: 14 jun. 2015.

IGOE, Tom. **Getting Started with RFID: Identify Objects in the Physical World with Arduino**. Sebastopol: O'reilly, 2012.

INNOVISION RESEARCH & TECHNOLOGY. **Near Field Communication in the real world: Turning the NFC promise into profitable, everyday applications**. Disponível em: <[http://www.nfc-forum.org/resources/white\\_papers/Innovision\\_whitePaper1.pdf](http://www.nfc-forum.org/resources/white_papers/Innovision_whitePaper1.pdf)>.

Acesso em: 2 maio 2015.

JACKSON, W. **Android Apps for Absolute Beginners**. Berkeley, Calif. : New York, NY: Apress, 2011.

LANGER, J; ROLAND, M. **Anwendungen und Technik von Near Field Communication(NFC)**. Berlim: Springer Berlin Heidelberg, 2010.

LECHETA, Ricardo R.. **Google Android: aprenda a criar aplicações para dispositivos móveis com android SDK**. 2. ed. São Paulo: Novatec Editora, 2010. 608 p.

LEE, Wei-Meng. **Introdução ao Desenvolvimento de Aplicativos para Android**. Rio de Janeiro: Ciência Moderna, 2011. 442 p.

LYYTINEN, Kalle; YOO, Youngjin. Ubiquitous computing. **Communications Of The Acm**. New York, p. 63-96. jan. 2002. Disponível em: <[http://pahl.de/download/dissertation/ds2os.lab/files/commun.\\_acm\\_2002\\_lyytinen.pdf](http://pahl.de/download/dissertation/ds2os.lab/files/commun._acm_2002_lyytinen.pdf)>. Acesso em: 19 jun. 2015.

LG. **Smartphone LG G3**. Disponível em:< <http://www.lge.com/br/celulares/lg-d855-g3> >. Acesso em: 19 out. 2015.

LEHRER, Cristiano. **Tecnologias Web: Java Servlets**. 2012. Disponível em: <<http://www.ybadoo.com.br/tutoriais/web/04/slides.pdf>>. Acesso em: 19 out. 2015.

MADI, Raquel. **Número de abandono de animais cresce durante férias escolares**. Terra Jan. 2014. Disponível em: <<http://mulher.terra.com.br/numero-de-abandono-de-animais-cresce-durante-ferias-escolares,514bf41101fa3410VgnVCM5000009ccce b0a RCRD.html>>. Acesso em: 30 ago. 2014.

MANSANO, C et al. **Avaliação de diferentes locais de implante de microchip para identificação eletrônica de reprodutores de rã-touro e sua validação por meio da análise de imagem**. Cbra. São Paulo, p. 295-297. 2013. Disponível em: <[http://www.cbra.org.br/pages/publicacoes/rbra/v37n3/pag295-297\\_\(RB436\).pdf](http://www.cbra.org.br/pages/publicacoes/rbra/v37n3/pag295-297_(RB436).pdf)>. Acesso em: 03 maio 2015.

MARCIO, Wellinton. **Coleira de identificação pode salvar seu animalzinho**. 2012. Disponível em: <<http://www.douradosagora.com.br/dourados/coleira-de-identificacao-pode-salvar-seu-animalzinho>>. Acesso em: 01 set. 2014.

MEDEIROS, Higor. **DBDesigner: Modelagem e Implementação de banco de dados**. 2015. Disponível em: <<http://www.devmedia.com.br/dbdesigner-modelagem-e-implementacao-de-banco-de-dados/30897>>. Acesso em: 16 out. 2015.

MEIER, R. **Professional Android Application Development**. 1. ed. Indianapolis, IN: Wiley Publishing, Inc, 2009.

MIRES, Jesús Jorge Herrera. **Diseño e implementación de una aplicación móvil basada en la tecnología nfc para acceso a información de las piezas de arte de un museo**. 2013. 102 f. TCC (Graduação) - Curso de Engenharia da Computação,

Universidade Católica do Perú, Perú, 2013. Disponível em: <[http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5351/herrera\\_jesus\\_diseño\\_aplicacion\\_movil\\_tecnologia\\_nfc\\_acceso\\_informacion\\_piezas\\_arte\\_museo.pdf?sequence=1](http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/5351/herrera_jesus_diseño_aplicacion_movil_tecnologia_nfc_acceso_informacion_piezas_arte_museo.pdf?sequence=1)>. Acesso em: 15 jun. 2015.

NOH, Sun-kuk; LEE, Seong-ro; CHOI, Dongyou. **Proposed M-Payment System Using Near-Field Communication and Based on WSN-Enabled Location-Based Services for M-Commerce**. Hindawi Publishing Corporation. Gwangju, p. 1-8. 15 abr. 2014. Disponível em: <<http://www.hindawi.com/journals/ijdsn/2014/865172/>>. Acesso em: 04 maio 2015.

OPEN HANDSET ALLIANCE. **Alliance Overview**. Disponível em: <<http://www.openhandsetalliance.com/>>. Acesso em: 11 maio 2015.

PALMEIRA, Thiago Vinícius Varallo. **Ciclo da vida do Servlet**. 2013. Disponível em: <<http://www.devmedia.com.br/ciclo-da-vida-do-servlet/27919>>. Acesso em: 19 out. 2015.

PATEL, Jignesh; KOTHARI, Badal. Near field communication - the future technology for an interactive world. **International Journal Of Engineering Research And Science & Technology**. Hyderabad, p. 55-59. 01 maio 2013. Disponível em: <[http://ijerst.com/download.php?file=ijerstadmin/upload/IJEETC\\_518299068c8a9.pdf&iid=27](http://ijerst.com/download.php?file=ijerstadmin/upload/IJEETC_518299068c8a9.pdf&iid=27)>. Acesso em: 03 maio 2015.

PEREIRA, Lucio Camilo Oliva; SILVA, Michel Lourenço da. **Android para Desenvolvedores**. 3. ed. Rio de Janeiro: Brasport, 2012. 240 p.

POTTS, Stephen; KOPACK, Mike. **Aprenda Web Services em 24 Horas**. Campus, Rio de Janeiro, RJ. 2003

QURATULAIN, Masud; MUNAZZA, Fatima. NFC Tags Based Notification System for Medical Appointments. Ausmt, Pakistan, v. 4, n. 4, p.191-195, 1 dez. 2014. **International Journal of Automation and Smart Technology**. DOI: 10.5875/ausmt.v4i4.742.

RIBEIRO, Gabriel. **A evolução do Nexus: veja como o Google projetou seus smartphones**. 2013. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2013/04/evolucao-do-nexus-veja-como-o-google-projetou-seus-smartphones.html>>. Acesso em: 25 set. 2014.

SAMPLE, A.p. et al. Design of an RFID-Based Battery-Free Programmable Sensing Platform. **Ieee Trans. Instrum. Meas.**, [s.l.], v. 57, n. 11, p.2608-2615, nov. 2008. Institute of Electrical & Electronics Engineers (IEEE). DOI: 10.1109/tim.2008.925019.

SCHMITZ, Daniel. **Dominando Android**. São Paulo: Leanpub, 2014. 8 p.

SILVA, Marcelo de Oliveira Caron et al. POSSE RESPONSÁVEL DE CÃES E GATOS NO MUNICÍPIO DE GARÇA/SP. **Revista Científica Eletrônica de Medicina Veterinária**, Garça, n. 6, p.4-9, jan. 2006. Semestral. Disponível em: <[http://faef.revista.inf.br/imagens\\_arquivos/arquivos\\_destaque/3DHv9yBzqLrhlo2\\_20](http://faef.revista.inf.br/imagens_arquivos/arquivos_destaque/3DHv9yBzqLrhlo2_20)>

13-5-20-15-26-52.pdf>. Acesso em: 19 maio 2015.

SILVA, Renato Ramos da. **Desenvolvimento de um módulo de acionamento para abertura da fechadura de um quarto de hotel baseado na tecnologia NFC**. 2014. 81 f. Monografia (Especialização) - Curso de Desenvolvimento de Produtos Eletrônicos, Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Florianópolis, 2014. Disponível em: <[http://professorpetry.com.br/Ensino/Defesas\\_Pos\\_Graduacao/Defesa\\_47\\_Renato\\_Ramos\\_da\\_Silva\\_Desenvolvimento\\_de\\_um\\_Modulo\\_de\\_Acionamento\\_Para\\_Abertura\\_de\\_Fechadura\\_de\\_um\\_Quarto\\_de\\_Hotel\\_Baseado\\_na\\_Tecnologia\\_NFC.pdf](http://professorpetry.com.br/Ensino/Defesas_Pos_Graduacao/Defesa_47_Renato_Ramos_da_Silva_Desenvolvimento_de_um_Modulo_de_Acionamento_Para_Abertura_de_Fechadura_de_um_Quarto_de_Hotel_Baseado_na_Tecnologia_NFC.pdf)>. Acesso em: 14 jun. 2015.

SILVANO, D. et al. Divulgação dos princípios da guarda responsável: uma vertente possível no trabalho de pesquisa a campo. **Revista Eletrônica Novo Enfoque**, v.09, n.09, p. 64-86, 2010. Disponível em: <<http://www.castelobranco.br/sistema/novoenfoque/files/09/artigos/06.pdf>>. Acesso em: 12 jun. 2013.

SONNTAG. **NFC Tag Types**. 2011. Disponível em: <<http://www.nfc.cc/technology/nfc-tag-types/>>. Acesso em: 05 maio 2015.

STEELE, J.; TO, N. **The Android Developer's Cookbook: Building Applications with the Android SDK: Building Applications with the Android SDK**. 1. ed. Upper Saddle River, NJ: Addison-Wesley Professional, 2010.

TELECO, **Inteligência em Telecomunicações. SMARTPHONE**. 2013. Disponível em: <<http://www.teleco.com.br/smartphone.asp>>. Acesso em: 04 nov. 2013.

TELECO. **Sistemas operacionais**. 2015. Disponível em: <[http://www.teleco.com.br/sist\\_operacional.asp](http://www.teleco.com.br/sist_operacional.asp)>. Acesso em: 20 jun. 2015.

TEMPLE, André et al. **Jsp, Servlets e J2EE**. 2004. Disponível em: <<http://www.inf.ufsc.br/~bosco/downloads/livro-jsp-servlets-j2ee.pdf>>. Acesso em: 19 out. 2015.

**APÊNDICE(S)**

## APÊNDICE A – ARTIGO CIENTÍFICO

**Protótipo de Aplicativo Android para Identificação de Animais Utilizando a Tecnologia NFC**Julia de Rochi<sup>1</sup>, Sergio Coral<sup>2</sup>

<sup>1</sup>Acadêmico do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias (UnaCET) – Universidade do Extremo Sul Catarinense (UNESC) Av. Universitária, 1105 - Bairro Universitário – Criciúma – SC – Brasil

<sup>2</sup>Professor do Curso de Ciência da Computação – Unidade Acadêmica de Ciências, Engenharias e Tecnologias (UnaCET) – Universidade do Extremo Sul Catarinense (UNESC) Av. Universitária, 1105 - Bairro Universitário – Criciúma – SC – Brasil

juliaderoche@hotmail.com, sergiocoral@unesc.net

**Abstract.** *Nowadays, the animals abandon levels are too high. The existing NGO's can't take care of and help all this abandoned animals. Facing this problem, the animal identification is primordial and helps the proprietary location, also giving information about the animal. Facing this problem and the studied technologies, a prototype application was developed in the Android platform with the capabilities of identify NFC tags and write data about the animal and the proprietary, including a Java developed server that communicate the application with the database.*

**Resumo.** *Hoje em dia, o índice de abandono de animais é muito alto. As ONG's existentes hoje não conseguem cuidar e socorrer todos os animais abandonados. Diante deste problema, a identificação do animal torna-se fundamental, pois facilita a localização do proprietário e obtém-se informações referentes ao animal. Diante deste problema e das tecnologias estudadas para poder resolver o mesmo, foi desenvolvido um protótipo de aplicativo na plataforma Android capaz de identificar tags NFC e cadastrar dados referentes ao animal e seu proprietário, contando com o auxílio de um servidor desenvolvido em Java que realiza a comunicação entre a aplicação e o banco de dados.*

## 1. Introdução

Atualmente, o índice de abandono de animais é muito grande, e isto pode ser comprovado pelo grande número de animais que lotam os abrigos brasileiros, principalmente durante as férias (MADI, 2014).

No passado, foram utilizadas técnicas de marcação simples como cortes nas orelhas com diferentes localizações e aspectos, bem como a marcação por queimadura à fogo ou à azoto líquido. Posteriormente, evoluiu-se tais técnicas para tatuagens nas orelhas e nas bragadas (zona interna das coxas, sem pelos), seguindo o uso de coleiras, anilhas e brincos e, mais recentemente, o uso de identificadores eletrônicos (DÓRIA, 2010).

Visando este problema, foi desenvolvido um protótipo de uma aplicação em Android para disponibilizar esta identificação para qualquer usuário que possua um celular com a tecnologia Near Field Communication (NFC).

A tecnologia NFC é um tipo de comunicação entre dispositivos de curto alcanc, para transferir ou receber dados, bastando que dois celulares ou um celular e um tag com a tecnologia NFC se encostem para efetuar a transferência de dados. (FINKENZELLER, 2010).

## **2. Abandono de animais**

O Brasil atualmente é o segundo no ranking de população de cães e gatos, contando com um número de 101,1 milhões de animais de estimação. O crescimento do número de animais de estimação está acontecendo, pois, as pessoas estão mudando os seus estilos de vida, se isolando mais em seus lares e dando um maior valor afetivo a outras espécies (FIOCCO, 2012).

Os animais de estimação são completamente dependentes dos seres humanos, que devem fornecer a eles bem-estar físico, psicológico e emocional.

Em alguns estados existem leis mais específicas para o controle do bem-estar do animal, como por exemplo a cidade de Curitiba que criou a Lei Municipal 13.914 de 2011, onde afirma que os estabelecimentos que realizam vendas de animais vivos devem possuir alvarás cadastrados na prefeitura e relatórios de todos os animais nascidos, vendidos, permutados e doados. Todos os animais devem sair dos estabelecimentos de vendas com um microchip de identificação cadastrado no Sistema de Identificação Animal da Rede de Defesa e Proteção Animal de Curitiba (PREFEITURA MUNICIPAL DE CURITIBA, 2011).

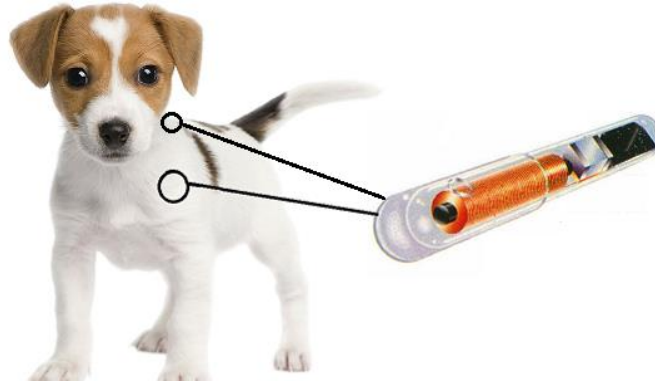
### **2.1. Identificação de Animais**

A identificação é um instrumento fundamental na gestão das populações de cães e gatos. Existem inúmeras razões para que seja incentivada a identificação em animais de estimação, entre elas estão a facilidade de encontrar o dono de um animal desaparecido, diminuindo assim o número de animais nas ruas ou em abrigos (WSPA, 2008, tradução nossa).

A identificação do animal fica ainda mais eficaz quando acoplada a um sistema de identificação que seja unificado a um banco de dados central (normalmente nacional). A cada animal é atribuído um código de identificação permanente, quando pesquisado este código no banco de dados será possível encontrar o contato do proprietário do animal (WSPA, 2008, tradução nossa).

Existem vários métodos de identificação apropriados para animais de estimação sendo que o método escolhido deve ser o mais humano e eficiente possível. O microchip é uma tecnologia inovadora para a monitorização de animais, e com várias vantagens se colocado em comparação com os demais métodos de identificação (MANSANO et al, 2013).

O microchip é composto de material esterilizado e é implantado abaixo a pele do animal, a sua aplicação não causa dor no animal e não necessita sedação. O mesmo possui um microcircuito contendo um código único e inalterado, inserido dentro de uma capsula de biovidro cirúrgico. O microchip é inativo e não possui uma bateria própria, o mesmo passa a transmitir informação apenas quando entra em contato com um leitor, retornando os dados referente ao mesmo (PREFEITURA MUNICIPAL DE CURITIBA, 2011). Na figura 1 é demonstrado o microchip e também os locais mais comuns onde o microchip pode ser inserido no animal, que seria o seu dorso ou no lateral da pata dianteira.



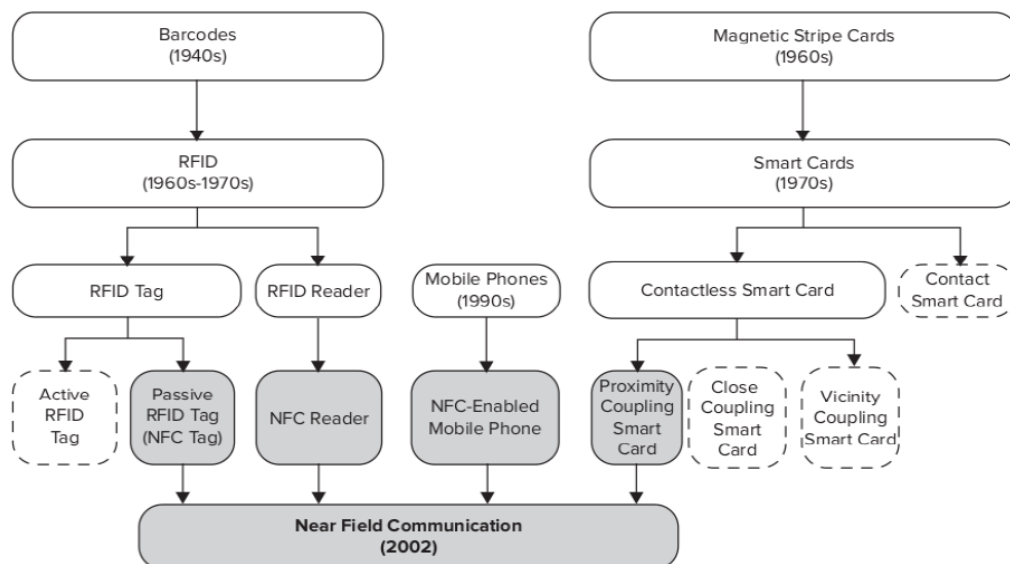
**Figura 1. Aplicação Tag no animal**

O microchip tem uma vida de vinte e cinco anos, sendo assim um tipo de identificação permanente. O processo de implantação é rápida e indolor e não causará nenhum desconforto se implantado corretamente (WSPA, 2008, tradução nossa).

### 3. Near Field Communication

O NFC é uma tecnologia de comunicação sem fio de curto alcance que permite o compartilhamento bidirecional de pequenas cargas de dados entre dispositivos eletrônicos (ANDROID DEVELOPERS, 2015c, tradução nossa).

Essa tecnologia foi baseada na tecnologia RFID, já existente e permitindo trocas mais complexas com o uso de etiquetas inteligentes conforme mostra a figura 3 (IGOE, 2012, tradução nossa).



**Figura 2. Derivação do Near Field Communication**

Esta tecnologia foi desenvolvida no ano de 2002 a partir de tecnologias já existentes, a MIFARE e FeliCa, produtos que foram desenvolvidos pela NXP Semiconductors (antiga Philips Semiconductors) em conjunto com a Sony (LANGER; ROLAND, 2010, tradução nossa).

O NFC funciona na frequência central 13,56 MHz, e sua velocidade de transmissão de dados pode variar entre 106, 212 Kb/s e 424 Kb/s (kilobits por segundo). A norma ISO/IEC

18092 define também os modos de comunicação que são os mesmos do RFID: ativo e passivo. No modo ativo os dispositivos transmitem os dados gerando cada um o seu próprio campo de rádio para a transmissão, enquanto o modo passivo apenas um dos dispositivos gera um campo de rádio transmissão e o outro dispositivo também utiliza essa modulação para gerar dados. Conforme visto no protocolo de NFC o responsável por gerar o campo de rádio é o dispositivo de iniciação (INNOVISION RESEARCH & TECHNOLOGY, 2013, tradução nossa).

Os dispositivos NFC assim como os RFID, podem trabalhar também em modo ativo e em modo passivo. Além de existirem os modos ativos e passivos os aparelhos NFC tem a capacidade de trabalhar em três diferente modos de operação: leitura/escrita, peer-to-peer (P2P) e emulação de cartões (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

Os leitores NFC são dispositivos ativos que permitem transferência de dados bidirecional com outro elemento NFC. Os leitores podem ler ou escrever dados, seja entre dispositivos ativos ou ativo e passivo. Ultimamente a integração da tecnologia NFC em dispositivos móveis cria oportunidade para exibir a facilidade de uso e consentimento do ecossistema (COSKUN; OK; OZDENIZCI, 2012, tradução nossa).

O NFC também trabalha com tags, que podem ser etiquetas, pulseiras, pequenos microchips com uma pequena antena que podem conter uma pequena quantidade de dados para transferir para outro dispositivo NFC, como um celular. A arquitetura de um tag NFC é composta por um chip de rádio ligado a uma antena e uma pequena área de memória. A antena é feita de alumínio ou cobre. O cobre acaba se tornando mais caro, porém tem uma melhor execução. As antenas são indutores projetados para converter campos magnéticos das proximidades em energia. (PATEL; KOTHARI, 2013, tradução nossa).

Segundo Clark (2013, tradução nossa) até 2017 os aparelhos com NFC embarcados irão atingir uma taxa anual de crescimento de 65%, podendo chegar a 2,1 bilhões de unidades.

#### **4. Aplicativo Android para Identificação de Animais Utilizando a Tecnologia NFC**

Por meio do conhecimento adquirido durante o levantamento bibliográfico, foi possível criar um protótipo de aplicação de identificação de animais baseada na plataforma Android, utilizando a tecnologia NFC para possibilitar a identificação através da comunicação entre smartphones e tag.

Tendo em vista que a identificação de animais é importante, o intuito deste projeto é a criação de algo que facilite essa identificação, evitando assim o sofrimento do animal e do dono que perdeu seu animal.

##### **4.1. Metodologia**

O início deste projeto de pesquisa foi constituído pelo levantamento bibliográfico a fim de obter boas referências bibliográficas e conteúdo teórico. Foram também realizadas pesquisas sobre a tecnologia NFC, sua funcionalidade e também o modo de interagir entre tag e smartphone. A seguir os estudos foram voltados para a idealização do sistema e a seleção de ferramentas de desenvolvimento.

A fundamentação teórica foi dividida em partes com objetivo de direcionar o trabalho e estruturar a implementação. Esse estudo iniciou com a pesquisa sobre o índice de abandono de animais e a importância da identificação dos mesmos.

Em seguida o foco foi direcionado ao estudo da tecnologia NFC, de onde surgiu, e também para um melhor entendimento sobre aspectos gerais de transmissão e as demais características. Foi realizado também o estudo sobre o Sistema Operacional Android, que

foi a plataforma escolhida para o desenvolvimento do trabalho proposto por ser a plataforma de código fonte aberto, fácil desenvolvimento devido ao maior conhecimento da parte do acadêmico e em seguida foi realizado um estudo sobre o funcionamento do NFC junto com o sistema Android juntamente com uma web service para manter as informações em um banco de dados, compartilhando as informações com outros dispositivos.

Com este embasamento teórico, se iniciou o desenvolvimento do protótipo. Primeiramente foi realizado um levantamento as ferramentas as serem utilizadas para o desenvolvimento do projeto. Para entender melhor o funcionamento do NFC no android, foi criada uma aplicação teste, apenas com funcionalidades básicas do NFC, como, ler e escrever em uma tag. Em seguida se se iniciou a modelagem do banco de dados para a criação do banco. A próxima etapa foi o desenvolvimento da aplicação.

Os testes foram realizados durante o desenvolvimento da aplicação. Todas as compilações e execuções foram realizadas diretamente no dispositivo móvel com a plataforma android.

## **5. Desenvolvimento do protótipo**

Após todas as ferramentas configuradas e instaladas corretamente se deu início ao desenvolvimento do protótipo do aplicativo de identificação. Inicialmente foi criada apenas uma aplicação de teste, onde se aplicou a funcionalidade da tecnologia NFC no Android. Após um bom entendimento se deu início ao projeto principal.

### **5.1. Estrutura do servidor web**

A criação do Servlet foi realizada através do NetBeans juntamente com o servidor GlassFish na versão 4.0. O mesmo foi construído em Java através do Servlet classe padrão que fica sobre controle do container, nesse caso o servidor GlassFish.

A troca de dados entre a aplicação Android e o serviço acontece através de trocas de dados JSON ou através de confirmações sobre a operação. Quando o aplicativo Android faz uma requisição para listar os animais referentes a certo proprietário, o mesmo irá retornar um Array de JSON com todos os animais referentes ao proprietário. Caso a aplicação Android faça a requisição de excluir um animal o serviço irá retornar para aplicação Android apenas uma string com o sucesso ou fracasso da operação.

Para a realização da conexão com o banco de dados foi criada a classe ConnectionFactory, com a configurações necessárias para a comunicação entre a aplicação web e o banco de dados. No caso da aplicação IdentificaPet o banco de dados utilizado foi o PostgreSQL.

### **5.2. Aplicação Android**

Para dar início ao desenvolvimento da aplicação mobile, foi definido no manifesto da aplicação as permissões para acesso aos hardwares necessários para execução do projeto que podemos ver na figura 3.

Para que a aplicação não seja instalada em aplicativos que não possuem o NFC, também no manifesto definimos a condição na tag uses-feature informando true na tag required, fazendo assim que o Google Play não mostre o aplicativo para usuários sem suporte ao NFC.

```

<uses-permission android:name="android.permission.NFC" />
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<uses-feature
    android:name="android.hardware.nfc"
    android:required="true" />

```

**Figura 3. Configuração do manifesto**

Quando necessário a comunicação com o servidor, é chamada a função desejada da classe `AcessoServlet`, no caso do login é chamada a `LoginProprietario` que está dentro da classe `AcessoServlet`.

Todas as funções da classe `AcessoServlet` tem como parâmetro a `activity` e o objeto que implementa a interface, e os parâmetros necessários para passar ao servidor. Inicialmente a classe `AcessoServlet` realizava diretamente a comunicação com o servidor, causando assim um travamento da tela no momento da comunicação. Para resolver esse problema foi criada uma classe chamada `AcessoServletAsync` que estende `AysncTask`, permitindo realizar a requisição em segundo plano e publicar os resultados na interface, sem manipulação de threads.

Para realizar a leitura da tag NFC é importante definir no manifesto a versão mínima SDK de nível 10, pois o NFC é suportado apenas pela versão 2.3.3. O modo de operação escolhido para implementação foi o modo passivo, onde apenas o dispositivo irá gerar um sinal de radiofrequência para a tag, que será alimentada pelo mesmo e irá transmitir os dados do tipo NDEF para a aplicação.

Antes de abrir a `activity` que irá identificar a tag NFC, deve-se verificar se o NFC está ativo no celular, para isso é necessário a importação da classe `android.nfc.NfcAdapter` que contém as informações referente a tecnologia NFC do dispositivo utilizado. Para buscar o estado do `NfcAdapter` após a importação da biblioteca, basta utilizar a função `getDefaultAdapter` passando como parâmetro `activity` atual.

Após verificar que o NFC está ativo corretamente, foi criado um `PendingIntent` que será passado ao método `enableForegroundDispatch`, Este método deverá ser chamado dentro do thread principal e apenas quando a `activity` estiver em primeiro plano, ou seja, depois que o `OnResume()` tiver sido chamado. O `enableForegroundDispatch` tem como finalidade dar prioridade a aplicação ativa, caso contrário o sistema android irá mostrar uma lista com todas os aplicativos que tem a capacidade de ler uma tag NFC. Após `activity` deixar o primeiro plano logo após o `OnPause()`, é chamado o método `disableForegroundDispatch` para retirar a prioridade da mesma.

Assim que o sistema identifica a tag NFC, é chamado o método `onNewIntent()`, que irá ter como parâmetro a `intent` criada e passada no método `enableForegroundDispatch`. Nesse método através da função `getParcelableExtra()` pode-se obter todas as informações referente a tag NFC, no caso do `IdentificaPet` a informação que precisamos é o id da tag, que será obtido através da função `getNfcAdapterExtraID(intent)` que irá retornar o ID da tag NFC que irá servir para gravar os dados no banco dados.

Após o usuário inserir os dados e grava-los no banco de dados, será necessário aproximar novamente a tag para poder gravar dados na mesma, para que mesmo se o usuário não possua internet, a identificação do animal possa ser realizada, trazendo apenas os dados essenciais. Para gravar os dados na tag mais algumas bibliotecas foram necessárias, como `android.nfc.NdefMessage`, `android.nfc.NdefRecord`, `android.nfc.Tag`, `android.nfc.tech.Ndef` e `android.nfc.tech.NdefFormatable`.

Na função `writeNdefMessage` será passado como parâmetro a tag encontrada e a mensagem no formato `NdefMessage` que será gravada na mesma. Para gravar os dados na tag, primeiramente deve-se chamar o `connect()`, após a conexão os sistema irá verificar se a tag identificada é gravável, e em seguida é chamada a função `writeNdefMessage` da biblioteca `android.nfc.tech.Ndef` que irá gravar a `ndefMessage` na tag. Após a gravação do registro na tag o sistema irá disponibilizar uma lista com todos os animais cadastrados para o proprietário corrente.

### 5.3. Funcionamento do sistema

O fluxo do protótipo desenvolvido se dá início na tela de login figura 4, a mesma é composta pelo campo de acesso restrito a apenas usuários já cadastrados no sistema, opção para realizar o cadastro do proprietário e acesso para identificar o animal. O cadastro de usuário, que seria o proprietário do animal, possui campos com informações que será relevante no momento da identificação do animal.

Após realizar o cadastro o sistema irá levar o usuário direto para a tela de consulta de animais cadastrados, onde também é possível cadastrar um novo animal, excluir e editar os dados de um animal já cadastrado. Para realizar a exclusão de um animal basta ficar segurando o mesmo por alguns segundos, fazendo assim a exclusão automática do registro do banco de dados. Para editar os dados do animal, basta clicar uma vez em cima do mesmo, onde abrirá a tela de cadastro com os dados já cadastrados, possibilitando assim a edição.

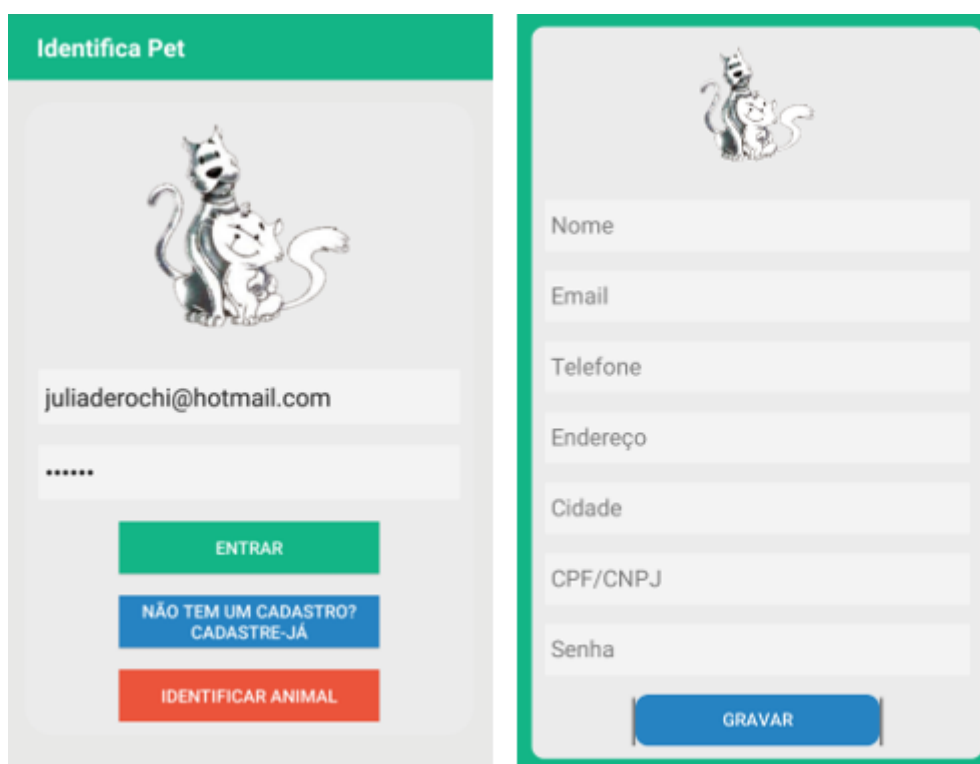


Figura 4. Tela de *login* e cadastro de proprietário

Ao adicionar um novo animal o usuário será levado até uma tela onde necessita a identificação da tag NFC para realizar o cadastro do animal. Após a identificação da tag será aberto a tela de cadastro onde já estará disponível apenas para visualização a informação referente ao ID da tag NFC, os demais campos ficaram disponível para o preenchimento das informações.

Para realizar a identificação basta o usuário selecionar o botão “IDENTIFICAR ANIMAL” da tela de login onde será necessário o usuário aproximar a tag próximo ao animal para realizar a

identificação do mesmo. Após identificação o usuário poderá visualizar as informações referente ao animal encontrado e também dados do proprietário do animal.

## **6. Resultados obtidos**

A partir da etapa de levantamento de requisitos realizada neste trabalho, foi possível levantar material referencial importante sobre todos os temas tratados na implementação do sistema proposto, sendo esse seu principal objetivo.

A aplicação do funcionamento do NFC foi muito satisfatória, sendo que existe muita documentação para compreender seu funcionamento. Não foram encontrados trabalhos iguais ou similares a esse, o qual a tag utilizada fosse a microchip NFC, deixando uma dúvida no começo da fundamentação sobre seu real funcionamento, que, porém, foram todas esclarecidas e compreendidas durante o desenvolvimento do projeto.

Os testes foram realizados em apenas um animal, a tag foi implantada no dorso do animal por sugestão do veterinário. A implantação foi realizada com sucesso e sem dificuldades, o animal não demonstrou dor após a aplicação da tag, apenas no momento devido a grossura da agulha.

Após a implantação podemos sentir a tag no animal, a mesma não fica sempre no ponto aplicado, podendo migrar um pouco, porém não muito distante do local da aplicação. Foram realizados vários testes de leitura, após posicionar o telefone próximo ao local onde se encontra a tag conseguimos após algumas tentativas a identificação da mesma. O posicionamento do celular também influencia na identificação. Após a leitura da tag, os dados referentes ao proprietário e ao animal foram demonstrados com sucesso para o usuário final.

Para o tema proposto por este trabalho de conclusão de curso, os resultados obtidos foram satisfatórios em relação aos objetivos apresentados. O objetivo principal que se trata do desenvolvimento de um protótipo de aplicativo Android capaz de identificar tag NFC aplicando o conhecimento obtido sobre o seu funcionamento e cadastrar os dados referente ao animal e seu proprietário na plataforma Android, contanto com o auxílio do servidor desenvolvido que serve como comunicador entre a aplicação e o banco de dados foi alcançado com sucesso.

## **7. Conclusão**

O número de animais abandonados está crescendo cada dia mais, aumentando assim o número de animais em ONG's e canis, que não estão dando conta de manter todos esses animais, pois o custo está cada vez mais elevado.

Diante desta situação a identificação de animais se torna essencial para que ONG's ou outros órgãos responsáveis possam ter reconhecimento do proprietário e informações importantes para o bem-estar do animal.

Durante o desenvolvimento do projeto foram encontrados alguns obstáculos. Um dos maiores problemas foi encontrar a tag NFC na frequência correta, já que o NFC pede uma frequência de 13.56 MHz, não sendo flexível. A obtenção da tag se teve após a pesquisa em um site chinês e em contato diretamente com o fornecedor, se pode encontrar a correta tag que levou em média de três meses para chegar ao Brasil. O custo da tag não foi um problema já que seu valor não ultrapassou dez reais, tendo em vista que hoje um pet shop no Brasil cobra em média cem reais para inserir a tag no animal.

Durante as fases de testes foi constatado que a distância da tag e do celular deve ser menor que 10 cm, dificultando assim o reconhecimento da mesma. O pelo do animal também dificultou a leitura da tag, porém não impossibilitou a identificação. A tag foi implantada na parte do dorso superior, porém após os testes e verificando a dificuldade na identificação, juntamente a um

veterinário, ficou definido que seria melhor a implementação da tag no ombro, por ser uma superfície lisa que facilitaria a sua localização e identificação.

Outros problemas foram encontrados durante o desenvolvimento da aplicação, mas foram se resolvendo à medida que os estudos se aprofundavam.

Sendo assim o objetivo geral e os objetivos específicos dessa pesquisa foram todos atingidos com êxito, tendo em vista que foi possível a identificação da tag NFC através de uma aplicação Android.

Este trabalho abre uma porta para a sociedade e para os meios públicos, facilitando a identificação de animais perdidos ou abandonados, podendo também ter um controle sobre a situação dos animais em cada cidade, caso a identificação fosse aplicada corretamente. O trabalho também foi submetido como proposta para o projeto Sinapse da inovação na categoria de tecnologia social para o setor de administração pública, já que o mesmo possui uma relevância social muito grande para a sociedade.

Por fim conclui-se que o trabalho teve realmente o resultado esperado, atingindo todos os itens propostos com a conclusão do desenvolvimento do protótipo de identificação de animais utilizando a tecnologia NFC.

## References

ANDROID DEVELOPERS. NFC Basics. Disponível em: <<http://developer.android.com/guide/topics/connectivity/nfc/nfc.html#p2p>>. Acesso em: 14 jun. 2015c.

COSKUN RESEARCH & TECHNOLOGY. Near Field Communication in the real world: Turning the NFC promise into profitable, everyday applications. Disponível em: <[http://www.nfc-forum.org/resources/white\\_papers/Innovision\\_whitePaper1.pdf](http://www.nfc-forum.org/resources/white_papers/Innovision_whitePaper1.pdf)>. Acesso em: 2 maio 2015.

DÓRIA F. A Identificação de animais de companhia. Disponível em: <[http://www.vetbiblios.pt/ARTIGOS\\_TECNICOS/Diversos/IDENTIFICACAO\\_ANIMAIS\\_COMPANHIA.pdf](http://www.vetbiblios.pt/ARTIGOS_TECNICOS/Diversos/IDENTIFICACAO_ANIMAIS_COMPANHIA.pdf)>. Acesso em: 04 out. 2014.

FINKENZELLER, K. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication. 3. ed. West Sussex, Reino Unido: John Wiley & Sons Ltd, 2010.

FIOCCO, Alessandro. População de gatos crescerá 8% em 2012 e a de cães 4%. 2012. Disponível em: <<http://f5.folha.uol.com.br/bichos/1122747-populacao-de-gatos-crescera-8-em-2012-a-de-caes-4.shtml>>. Acesso em: 15 out. 2014.

IGOE, Tom. Getting Started with RFID: Identify Objects in the Physical World with Arduino. Sebastopol: O'reilly, 2012.

INNOVISION RESEARCH & TECHNOLOGY. Near Field Communication in the real world: Turning the NFC promise into profitable, everyday applications. Disponível em: <[http://www.nfc-forum.org/resources/white\\_papers/Innovision\\_whitePaper1.pdf](http://www.nfc-forum.org/resources/white_papers/Innovision_whitePaper1.pdf)>. Acesso em: 2 maio 2015.

LANGER, J; ROLAND, M. Anwendungen und Technik von Near Field Communication(NFC). Berlim: Springer Berlin Heidelberg, 2010.

MADI, Raquel. Número de abandono de animais cresce durante férias escolares. Terra Jan. 2014. Disponível em: <<http://mulher.terra.com.br/numero-de-abandono-de-animais-cresce-durante-ferias-escolares,514bf41101fa3410VgnVCM5000009ccce b0a RCRD. html>>. Acesso em: 30 ago. 2014.

MANSANO, C et al. Avaliação de diferentes locais de implante de microchip para identificação eletrônica de reprodutores de rã-touro e sua validação por meio da análise de imagem. Cbra. São Paulo, p. 295-297. 2013. Disponível em: <[http://www.cbra.org.br/pages/publicacoes/rbra/v37n3/pag295-297 \(RB436\).pdf](http://www.cbra.org.br/pages/publicacoes/rbra/v37n3/pag295-297 (RB436).pdf)>. Acesso em: 03 maio 2015.

PATEL, Jignesh; KOTHARI, Badal. Near field communication - the future technology for an interactive world. International Journal Of Engineering Research And Science & Technology. Hyderabad, p. 55-59. 01 maio 2013. Disponível em: <[http://ijerst.com/download.php?file=ijerstadmin/upload/IJEETC\\_518299068c8a9.pdf&iid=27](http://ijerst.com/download.php?file=ijerstadmin/upload/IJEETC_518299068c8a9.pdf&iid=27)>. Acesso em: 03 maio 2015.

PREFEITURA MUNICIPAL DE CURITIBA. Lei nº 13.914, de 23 de dezembro de 2011.