

## APLICATIVO PARA CONTROLE DE ENTRADA DE VISITANTES

Lucas Zimmer Martins<sup>1</sup>

Prof. Esp. Douglas Hermann<sup>2</sup>

Resumo: Este trabalho apresenta uma solução que visa permitir um controle mais informativo e dinâmico nos procedimentos de controle de visita das empresas. Esta foi criada para dispositivos móveis que utilizam o sistema operacional *Android* e tecnologias pertinentes à *smartphones* atuais, tais como o NFC, *QR Codes*, cartões inteligentes e câmeras nos dispositivos. Também possui integração com dispositivos *Android Wear*, entregando assim uma experiência completa para os usuários. Foram realizados testes de usabilidade do aplicativo, os quais permitiram a criação de uma interface agradável e intuitiva para o usuário, fornecendo uma forma simples de controlar e autorizar visitas. Por fim, concluiu-se que o aplicativo forneceu um melhor controle sobre os visitantes das empresas, permitindo uma maior agilidade e segurança durante estes procedimentos.

Palavras-chave: Android. Computação móvel. Controle de Visitantes. Android Wear. NFC. Cartões inteligentes. QRCode.

### 1 INTRODUÇÃO

A solução apresentada consiste em uma aplicação para dispositivos móveis *Android* cujo usuário será o recepcionista e/ou porteiro que realiza o procedimento de recepção da empresa e uma interface web que permite o pré-cadastramento de visitas. Neste aplicativo móvel estarão disponíveis, de forma ágil e fácil, todas as visitas previstas para o dia, bem como serão permitidas a ele algumas ações relacionadas ao processo de recepção de um visitante, tais como: captura de foto do visitante, associação de um *smartcard* a este visitante para permitir o seu acesso aos demais setores, solicitação de confirmação por parte do visitado, inclusão de uma nova visita que não estava prevista, entre outras.

Esta necessidade surgiu de uma empresa que possui um controle de visitantes que não é suficientemente eficiente e seguro. Atualmente, quando um visitante adentra a empresa, ele é atendido pelas recepcionistas e, então, seu cadastro é realizado, com informações como nome, CPF e RG. Após este cadastro inicial, a recepcionista efetua uma ligação para o responsável que irá receber os visitantes, para que o mesmo confirme que estava realmente esperando estes

---

<sup>1</sup> Lucas Zimmer Martins, graduado em Análise e Desenvolvimento de Sistemas pelo SENAC – [lucaszm@gmail.com](mailto:lucaszm@gmail.com)

<sup>2</sup> Douglas Hermann, graduado em Sistemas de Informação pela Feevale e pós-graduado em Gestão de Projetos pela Unisinos – [hermanndouglas@gmail.com](mailto:hermanndouglas@gmail.com)

visitantes e autorize a sua entrada. Todo este procedimento costuma levar algum tempo, o que costuma ocasionar algumas filas na recepção. Para resolver esse problema, esta solução foi idealizada e desenvolvida.

A seção 2 deste trabalho traz um referencial teórico com explicações sobre diversas tecnologias utilizadas nesta solução tais como: computação móvel, cartões inteligentes, RFID, NFC, *QRCode*, sistema operacional *Android* e *Android Wear*.

A seção 3 aborda trabalhos relacionados, mostrando algumas soluções já existentes e que possuam objetivos similares ao da solução apresentada, que é a de auxiliar no processo de recepção das empresas.

A seção 4 mostra a metodologia de desenvolvimento empregada durante a concepção desta solução, descrevendo etapa a etapa, desde a ideia até os testes de usabilidade e resultado final.

A seção 5 fornece uma visão geral da aplicação proposta, explicando todos os seus conceitos, requisitos e diagramas relacionados.

A seção 6 demonstra os resultados das avaliações e testes de usabilidade aplicados em um grupo de usuários selecionados, além de uma explicação sobre cada tela desenvolvida.

Por fim, a seção 7 apresenta a conclusão deste artigo, com explicações sobre o resultado obtido pela aplicação.

## **2 REFERENCIAL TEÓRICO**

Este capítulo pretende abordar as teorias que embasam o desenvolvimento deste trabalho. Primeiramente conceitualizando computação móvel, em seguida cartões inteligentes (*smart cards*), as tecnologias RFID e NFC, por último o sistema operacional *Android* e o *Android Wear*.

### **2.1 Computação Móvel**

Computação móvel pode ser definida como o ambiente computacional da mobilidade física. O usuário do ambiente de computação móvel poderá acessar dados, informações ou outros objetos lógicos a partir de qualquer dispositivo, de qualquer rede, a qualquer hora, em movimento (TALUKDER; AHMED; YAVAGAL, 2010).

Entre os aspectos distintivos dos sistemas de computação móvel estão a sua conectividade predominantemente sem fio, seu pequeno tamanho, a natureza móvel de seu uso, suas fontes de energia, e suas funcionalidades, que são particularmente adequadas para o usuário móvel. Por causa destas funcionalidades, os aplicativos de computação móvel são intrínsecos, diferentes de aplicações escritas para sistemas computacionais estacionários (B'FAR, 2005).

## 2.2 Cartões Inteligentes (*Smart Cards*)

Um cartão inteligente pode ser definido como um objeto que pode participar de uma transação eletrônica automatizada, pode armazenar dados com segurança, e pode armazenar ou executar uma série de funções e algoritmos de segurança. Um cartão inteligente não precisa possuir, obrigatoriamente, a forma de um cartão, podendo possuir formas e tamanhos variados (MAYES, 2008).

O principal componente de um cartão inteligente é o seu processador (CPU), que normalmente é cercado de outros quatro blocos funcionais: ROM (*“Read Only Memory”*, memória somente leitura), EEPROM (*“Electrical Erasable Programmable Read Only Memory”*, memória de somente leitura programável e eletricamente apagável), RAM (*“Random Access Memory”*, memória de acesso aleatório) e uma unidade que provê leitura e escrita. (RANKL; EFFING, 2010)

Os cartões inteligentes são divididos em dois grandes grupos, com base em suas formas de comunicação.

- Cartões inteligentes com contato: contêm um micro módulo contendo um chip único de silício com memória e microprocessador. Este chip, também conhecido como *“IC Card”* (*“Integrated Circuit Card”*, cartão de circuito integrado), é uma placa de contato de aproximadamente 1 cm<sup>2</sup>, posicionado na superfície de um cartão, e tipicamente de cor dourada. A conexão elétrica deste dispositivo é fornecida por um dispositivo externo quando este cartão é inserido no mesmo. A transmissão de comandos, dados e informações sobre o cartão é efetuado através desses pontos de contatos (COSKUN; OK; OZDENIZCI, 2012).
- Cartões inteligentes sem contato: são processados sem a necessidade de contato físico com um dispositivo externo. É a combinação de um cartão contendo um microchip e uma antena, isto permite que o cartão seja

monitorado à distância. Informações são armazenadas em um microchip que possui uma controladora e uma memória interna. Ao contrário dos cartões de contato, a fonte de energia vem de sua própria antena, que a captura através de campos eletromagnéticos, e é com a ajuda desta que a troca de dados entre o cartão e o dispositivo externo é efetuada (COSKUN; OK; OZDENIZCI, 2012).

Para esta aplicação serão utilizados cartões inteligentes sem contato do tipo MIFARE Classic. Estes cartões foram desenvolvidos pela empresa *Mikron* (posteriormente adquirida pela *Phillips*) e seu nome deriva de “*Mikron Fare Collection System*” (sistema de coleta de tarifas *Mikron*). Eles possuem como principais características: 768 bytes de armazenamento e permitem a leitura e escrita de informações (NFC Forum Type Tags, 2009).

### **2.3 RFID (*Radio-frequency identification*)**

O RFID (identificação por radiofrequência) é um termo genérico para a tecnologia que utiliza transmissores para identificar um objeto. A tecnologia tem sua origem no início do século 20 e foi inicialmente utilizado para identificar aeronaves militares como amigos/inimigos (EVDOKIMOV *et al.*, 2011).

Uma troca de informações por RFID envolve dois atores: uma *tag* e um leitor. O leitor inicia a comunicação ao gerar um campo de rádio e ficar ouvindo por ele, esperando por alguma resposta. A *tag* responde quando captura alguma transmissão deste leitor, inicialmente é fornecido um número de identificação único (“*Unique Identifier*”, ou UID), e, a partir daí, a comunicação está estabelecida (IGOE; COLEMAN; JEPSON, 2014).

Existem dezenas de padrões RFID aprovados pela *International Standards Organizations* (ISO), em conjunto com os maiores participantes do mercado de RFID. Entre estes padrões estão: ISO-11784 (originalmente desenvolvido para rastrear animais), EM4100 (opera na frequência 125kHz) e ISO-14443 (padrão desenvolvido para uso com cartões inteligentes e sistemas de pagamento, utiliza a frequência 13.56MHz) (IGOE; COLEMAN; JEPSON, 2014).

As tags com o padrão ISO-14443A são as compatíveis com o NFC, que será tratado abaixo.

## 2.4 NFC (*Near-field Communication*)

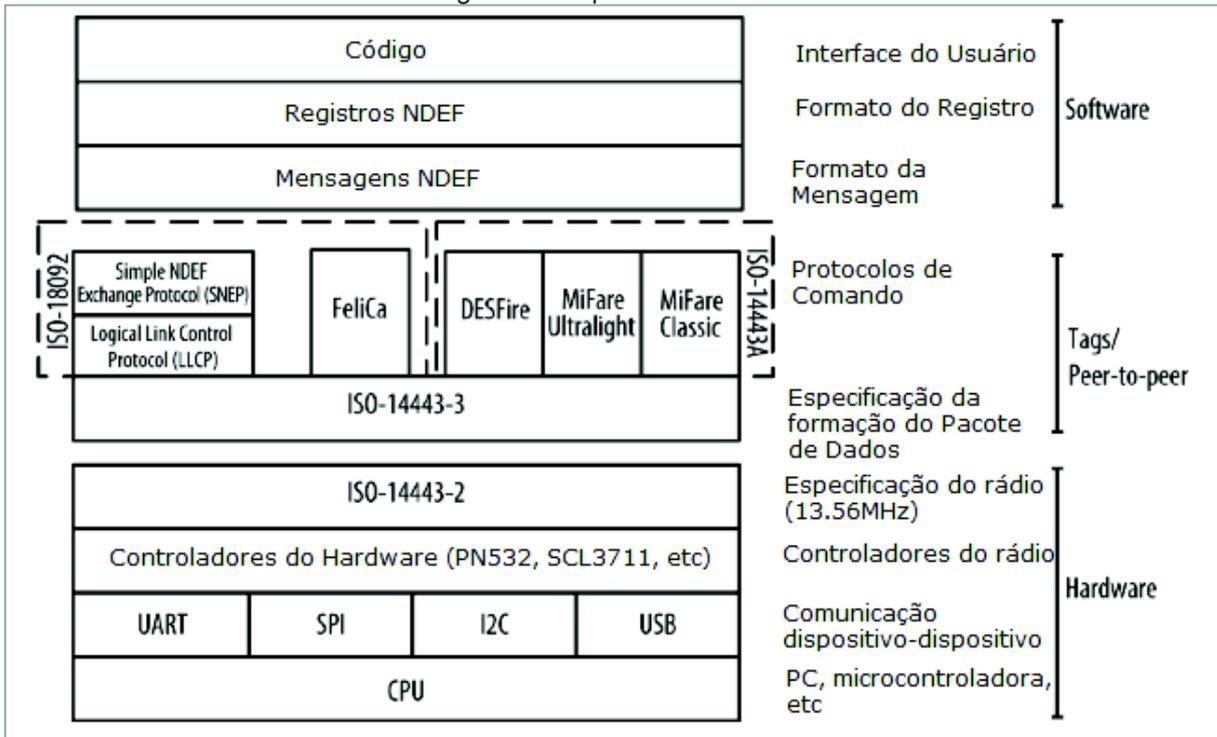
O NFC (comunicação por campo de proximidade) pode ser pensado como uma extensão do RFID. As trocas de dados por NFC também envolvem uma *tag* (alvo) e um leitor (iniciador), porém podem fazer mais além de trocar UIDs e ler/escrever dados. (IGOE; COLEMAN; JEPSON, 2014). Existem três modos principais no qual o NFC opera: leitura/escrita de *tags* (permite a leitura e escrita de informações em *tags*), *peer-to-peer* (permite que dois dispositivos equipados com NFC comuniquem-se entre si para realizar a troca de informações ou de arquivos) e emulação de cartões (permite que dispositivos ajam como cartões inteligentes, e comuniquem-se com leitoras externas) (NFC Forum: What It Does, 2015).

A troca de dados entre dispositivos NFC é realizada utilizando o formato NDEF (*NFC Data Exchange Format*). Este é um padrão em comum que funciona em todos os dispositivos NFC, independente do tipo de *tag* ou da tecnologia do dispositivo utilizado. Cada mensagem NDEF possui um ou mais registros NDEF, sendo que cada registro possui um tipo de dado em particular, um identificador único, um tamanho e os dados (IGOE; COLEMAN; JEPSON, 2014).

### 2.4.1 Arquitetura do NFC

A arquitetura de comunicação do NFC é dividida em três camadas principais: o *hardware*, as *tags* e o *software*.

Figura 1 - Arquitetura do NFC



Fonte: (IGOE; COLEMAN; JEPSON, 2014)

Na camada de *hardware* se encontra a CPU e os controladores de rádio que fazem a comunicação com as *tags* (ou outros dispositivos) diretamente. O NFC trabalha na especificação de rádio ISO-14443-2 do RFID, que descreve rádios de baixa-energia operando a 13.56MHz (IGOE; COLEMAN; JEPSON, 2014).

Na camada das *tags* estão as especificações dos pacotes de dados, seus protocolos e seus formatos. Existem diversos protocolos diferentes, baseados em duas especificações distintas, ISO-14443A e ISO-18092. A principal diferença entre essas duas especificações é a implementação de um modo de comunicação *peer-to-peer*, presente apenas na ISO-18092 (IGOE; COLEMAN; JEPSON, 2014).

Na terceira camada, fica a parte do *software*, com as suas mensagens, formatos e informações. (IGOE; COLEMAN; JEPSON, 2014)

## 2.5 QRCode

O QRCode (*Quick Response Code*, código de resposta rápida) foi desenvolvido pela Denso Wave Incorporated em 1994. Um QRCode não apenas possui uma alta capacidade, mas também pode armazenar diversos tipos de informação. Possui também uma alta taxa de correção de erros, podendo ser lido até 30% do código mesmo se estiver rasurado ou não legível (FURHT, 2011).

Também é conhecido por sua capacidade de codificar caracteres japoneses de forma direta (KATO; TAN; CHAI, 2010).

O QR Code possui até 40 tamanhos diferentes, cada tamanho é chamado de “versão”. Cada versão possui uma configuração e um número de módulos diferentes. A versão 1 é constituída de 21 por 21 módulos e pode armazenar até 41 caracteres numéricos, ou 25 caracteres alfanuméricos (letras maiúsculas, números e alguns caracteres especiais), ou 17 caracteres binários (no padrão ISO 8859-1, com caracteres maiúsculos, minúsculos, números e outros caracteres especiais), ou 10 *kanjis/kanas* (caracteres japoneses). A sua maior versão (40), é constituída de 177 por 177 módulos e pode armazenar até 7089 caracteres numéricos, ou 4296 caracteres alfanuméricos, ou 2953 caracteres binários, ou 1817 *kanjis/kanas* (QRCode: Information capacity and versions of the QR Code, 2015).

Figura 2 - Versões do QRCode



Fonte: <http://www.qrcode.com/en/about/version.html>

## 2.6 Plataforma Android

O *Android* é um sistema operacional móvel baseado em uma versão modificada do *Linux*. Foi originalmente desenvolvida por uma *startup* de mesmo nome, “*Android, Inc.*” (LEE, 2012). Essa *startup* havia sido fundada em 2003 por Andrew “Andy” Rubin (HOOG, 2011).

Em Julho de 2005, a “*Android, Inc.*” foi adquirida pelo *Google*. Esta aquisição foi acompanhada de registros de novas patentes e serviços envolvendo *mobile* por parte da *Google*, o que aumentou as especulações que a *Google* estava desenvolvendo seu próprio smartphone (HOOG, 2011).

No dia 5 de novembro de 2007, Andy Rubin anunciou que o *Google*, em conjunto com a *Open Handset Alliance* (um grupo formado pelas principais

operadoras de telefonia, fabricantes e empresas de software) estavam lançando uma plataforma abrangente para dispositivos móveis, que incluía um sistema operacional, uma interface para o usuário e aplicativos. Uma semana depois, a Google lançou o kit de desenvolvimento de software (SDK) para que os desenvolvedores pudessem utilizar (HOOG, 2011).

Em outubro de 2008 foi lançado o primeiro smartphone com *Android* disponível ao público, o *HTC Dream* (também conhecido como *T-Mobile G1*).

Figura 3 - HTC Dream



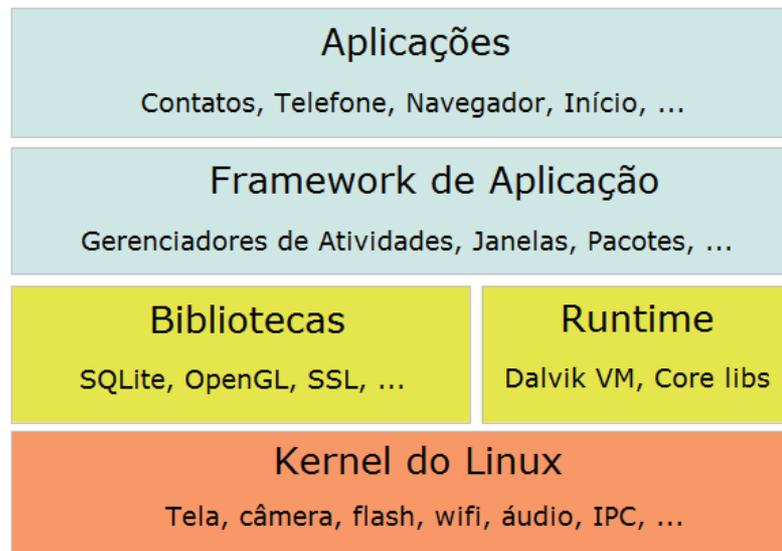
Fonte: <http://www.androidcentral.com/sites/androidcentral.com/files/postimages/684/htc-dream.jpg>

Por ser um sistema operacional de código aberto, pode ser sempre adaptado a fim de incorporar novas tecnologias, conforme estas forem surgindo. A plataforma vai estar sempre em evolução, já que as comunidades de desenvolvedores estarão trabalhando em conjunto para construir aplicações móveis inovadoras (PEREIRA; SILVA, 2009).

### 2.6.1 Arquitetura do *Android*

A arquitetura do sistema operacional *Android* está dividida em cinco partes, organizadas em quatro camadas (LEE, 2012):

Figura 4 - Arquitetura do Android



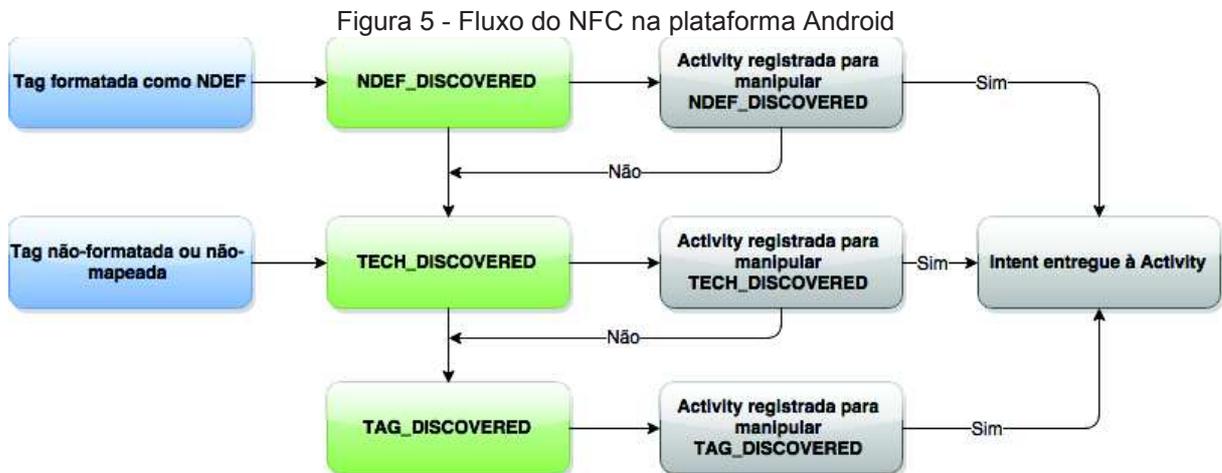
Fonte: <http://www.vogella.com/tutorials/Android/article.html>

- *Kernel do Linux*: é o *kernel* no qual o *Android* é baseado. Essa camada possui todos os drivers de baixo-nível dos componentes de *hardware*, como: câmera, *wi-fi*, áudio, etc.
- *Bibliotecas*: contém todo o código que disponibiliza os principais recursos de um sistema *Android*. A biblioteca do *SQLite*, por exemplo, disponibiliza todo suporte a uma base de dados para que uma aplicação possa utilizar para realizar a armazenagem de dados.
- *Runtime*: na mesma camada que as bibliotecas, o *Runtime* do *Android* provê um conjunto de bibliotecas centrais que permitem que desenvolvedores possam escrever aplicativos *Android* utilizando a linguagem de programação *Java*. O *Android Runtime* também inclui a *Dalvik Virtual Machine*, que possibilita que cada aplicativo *Android* rode em seu próprio processo, ela é uma máquina virtual desenvolvida especificamente para o *Android* e otimizada para dispositivos com memória e CPU limitados.
- *Framework* de Aplicação: expõe diversos recursos do sistema operacional para os desenvolvedores, para que possam ser utilizadas nas aplicações. Entre estes recursos estão: como caixas de textos, listas, botões, serviços *bluetooth*, serviços *wi-fi*, sensores de localização, etc).

- Aplicações: nesta camada se encontram as aplicações que já vem embarcadas no dispositivo *Android* (como Telefone, Contatos, Navegador, etc), assim como as aplicações que são baixadas e instaladas a partir da *Play Store* (loja de aplicativos do *Android*, fornecida pela *Google*, antigamente conhecida como *Android Market*).

## 2.7 NFC na plataforma *Android*

A implementação do NFC na plataforma *Android* funciona baseado em “*Intents*”, que são uma descrição abstrata de uma operação a ser executada (Intent | Android Developers, 2015). No caso específico do NFC existem três *intents* que podem ser trabalhados, cada *intent* ocorre em um momento diferente ao aproximar um cartão inteligente do dispositivo. Na imagem abaixo podemos visualizar o fluxo utilizado pelo *Android* durante esta detecção, tanto para *tags* que sejam conhecidas (ou seja, que já estejam formatadas corretamente), quanto para *tags* não formatadas ou não mapeadas, podendo ser utilizado um ou mais *intent* em cada aplicação.



Fonte: <https://developer.android.com/guide/topics/connectivity/nfc/nfc.html>

- NDEF\_DISCOVERED: este *Intent* ocorre quando uma *tag* que contém uma mensagem NDEF é escaneada e é de um formato conhecido. Este é o *Intent* com a maior prioridade, e o sistema de *dispatch* tenta iniciar uma *Activity* com este *Intent* antes de qualquer outro *Intent*, quando possível.
- TECH\_DISCOVERED: se a *tag* escaneada não possuir nenhuma mensagem NDEF de um formato conhecido, ou se a *tag* não possui

nenhuma mensagem NDEF, mas seja de um formato conhecido, este *Intent* é disparado.

- TAG\_DISCOVERED: este *Intent* é iniciado quando nenhuma *Activity* intercepta os *Intents* anteriores.

Na implementação desta aplicação foi utilizado apenas o *Intent* “TAG\_DISCOVERED”, visto que necessitaremos apenas do identificador único (*UID*) de cada cartão MIFARE, não sendo necessária a utilização de mensagens NDEF.

## 2.8 Smartwatches (relógios inteligentes) e Android Wear

Os *smartwatches* são os dispositivos vestíveis predominantes no mercado atual. Inicialmente a ascensão dos *smartphones* fez com que o relógio de pulso se tornasse praticamente redundante. Hoje, entretanto, o sucesso e a proliferação dos *smartphones* e nossa dependência cada mais profunda nos mesmos pavimentou o caminho para os *smartwatches*, que oferecem uma interface menos intrusiva. Os *smartwatches* servem mais como uma extensão de um *smartphone*, eles tipicamente possuem um ou mais mecanismos de conectividade, tais como *Bluetooth*, *Wi-Fi*, *USB* e outros (MISHRA, 2015).

O *Android Wear* é uma API (“*Application Programming Interface*”, interface de programação de aplicações) do *Google* para *smartwatches* (relógios inteligentes), *Google Glass* e outros dispositivos vestíveis. (NARDONE; SILVA, 2015). Um dispositivo com *Android Wear* conceitualmente é uma extensão de um *smartphone*, permitindo estender as funcionalidades do mesmo e provendo uma interface de usuário mais simples e leve que permite que o usuário receba notificações e resolva interações triviais de uma forma menos invasiva. Este dispositivo normalmente possui uma tela simples e pode aceitar controle de voz e de toque (MISHRA, 2015).

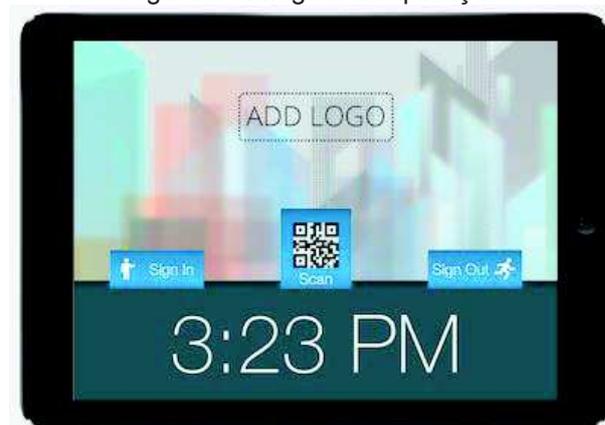
## 3 TRABALHOS RELACIONADOS

Na seção abaixo serão descritos aplicativos que possuem objetivos semelhantes, ou seja, permitem a confirmação da entrada de visitantes. Primeiramente será fornecido um resumo de cada aplicativo e após será exibido um quadro comparativo entre as soluções.

### 3.1 Inside

O primeiro aplicativo analisado foi o “*Inside – The App at your front desk*” (“O aplicativo na sua recepção”) da empresa [daVinci.io](http://daVinci.io). O aplicativo está disponível somente para o *iPad* (não possui uma versão para *iPhone* e nem para *Android*). Ele está mais voltado para o *check-in* realizado pelo próprio visitante e, portanto, possui alguns pontos diferentes com relação ao aplicativo que está sendo proposto neste trabalho. Permite a captura da assinatura do visitante, leitura de QRCode para agilizar a localização da visita pré-cadastrada, customização dos formulários de entrada, entre outras opções. Possui uma versão gratuita disponível (com menos recursos) e uma versão *premium*. Não está disponível na *AppStore* brasileira.

Figura 6 - Imagem da aplicação



Fonte: <https://atfrontdesk.com/>

### 3.2 Visitor Book

O segundo aplicativo analisado foi o “*Visitor Book*” (“Livro de Visitantes”) da empresa *Swiped On*. O mesmo se encontra disponível apenas para *iPad*. Possui uma versão gratuita e disponível na *AppStore* brasileira. A versão gratuita possui recursos reduzidos, não permitindo, por exemplo, a captura das fotos dos visitantes. Permite o controle das visitas por uma interface *web*, impressão de etiquetas de identificação, captura da assinatura, personalização da tela inicial.

Figura 7 - Imagem da aplicação



Fonte: <http://www.swipedon.com/visitor-registration-app>

### 3.3 Sign With Envoy

O terceiro aplicativo analisado foi o “*Sign With Envoy*” da empresa *Envoy*. Esta aplicação está disponível apenas para *iPad* e possui uma versão de demonstração por 15 dias. Após este período de demonstração o custo mensal varia de 99 dólares (para o plano Padrão e cobrança anual) a 499 dólares (para o plano Empresarial). Possui uma tela de controle/monitoramento na *web*, a captura de imagens dos visitantes, notificar os visitantes, bem como o pré-cadastramento dos visitantes (apenas nas versões Premium e Empresarial).

Figura 8 - Imagem da aplicação



Fonte: <https://envoy.co/>

### 3.4 iPad Receptionist

O último aplicativo analisado é o “*iPad Receptionist*” da empresa de mesmo nome. Possui versão apenas para *iPad*. Seu custo varia de 49 dólares por mês (versão Básica) a 249 dólares por mês (versão Empresarial). O Brasil não está

listado como opção na hora da compra. Possui as funcionalidades: notificações por mensagem de texto ou e-mail, tela de monitoramento na web, captura da foto do visitante, captura da assinatura.

Figura 9 - Imagem da aplicação



Fonte: <http://theipadreceptionist.com/>

### 3.5 Análise comparativa das aplicações

Abaixo se encontra uma tabela comparativa com as funcionalidades que cada alternativa possui, bem como as do aplicativo proposto.

Tabela 1 – Comparativo entre as aplicações apresentadas.

	<i>Inside</i>	<i>Visitor Book</i>	<i>Sign With Envoy</i>	<i>iPad Receptionist</i>	Aplicativo Proposto ( <i>Visitantes</i> )
Aplicativo Móvel	Sim (apenas para <i>iPad</i> )	Sim (Android)			
Identificação do Visitante utilizando QRCode	Sim	Não	Não	Não	Sim
Tela de Controle e/ou Monitoramento Web	Sim	Sim	Sim	Sim	Sim
Permite a associação de um cartão MIFARE ao Visitante	Não	Não	Não	Não	Sim
Captura da Foto do Visitante	Apenas na versão paga	Apenas na versão Premium	Sim	Sim	Sim

Pré-registro das Visitas	Apenas na versão paga	Não	Sim	Não	Sim
Integração com impressoras para geração de crachás	Sim	Sim	Sim	Sim	Não
Notificação do Visitado	Sim, por e-mail (apenas na versão paga)	Sim, por e-mail	Sim, por e-mail ou SMS	Sim, por e-mail ou SMS	Notificação no próprio aplicativo

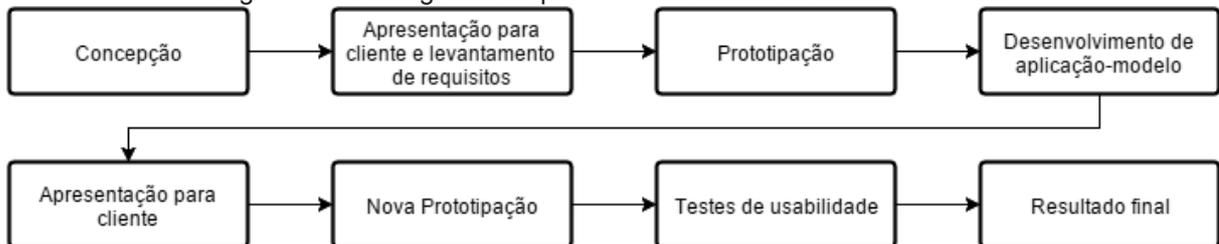
Fonte: o autor.

#### 4 METODOLOGIA

Esta seção irá descrever como foi realizado o processo de desenvolvimento desta solução descrevendo cada etapa do processo, desde a sua origem, até o resultado final.

As etapas realizadas estão demonstradas no fluxograma abaixo e serão detalhadas a seguir.

Figura 10 - Fluxograma do processo de desenvolvimento utilizado.



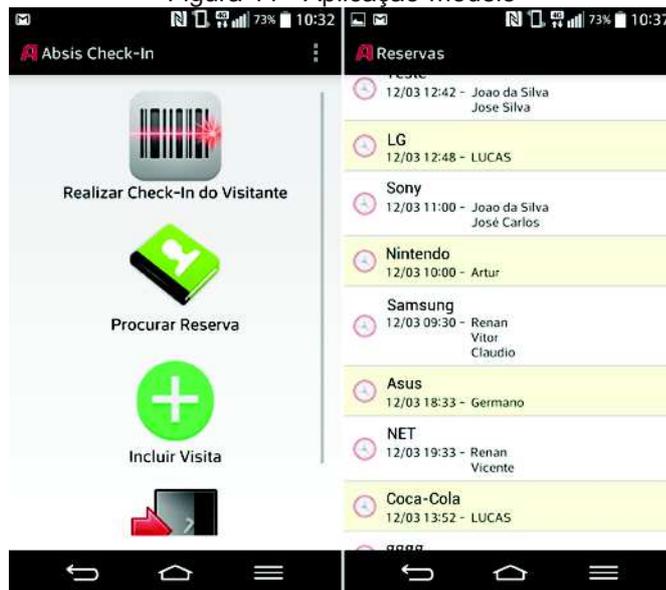
Fonte: o autor.

A primeira etapa realizada foi a da concepção da ideia da aplicação. Nesta etapa foi definido o objetivo da aplicação: auxiliar e otimizar o processo de entrada e controle de visitantes de uma empresa utilizando *smartphones* e todas as tecnologias disponíveis neles, como câmera e NFC.

Após a concepção da ideia da aplicação, foi realizada, com o responsável pela área de controle de acesso de uma empresa, uma reunião para apresentação da ideia geral da solução e a realização do levantamento de requisitos funcionais e não funcionais.

Para as próximas etapas foram realizadas a prototipação e o desenvolvimento de uma aplicação-modelo, com um visual mais simples e apenas com os requisitos básicos implementados, apenas para demonstração das funcionalidades.

Figura 11 - Aplicação-modelo



Fonte: o autor.

A partir desta etapa foi iniciado um novo processo de prototipação e desenvolvimento de *mockups* para uma nova versão da aplicação, contendo todas as funcionalidades descritas nos requisitos. Usando estes *mockups* como base, a nova versão foi desenvolvida.

Com o desenvolvimento desta nova versão a próxima etapa executada consistiu em validar a experiência dos usuários durante a utilização do aplicativo. Para este propósito, foi definido que seriam realizados testes de usabilidade com os usuários. Estes testes ajudam a identificar problemas que pessoas podem ter alguma interface em específico e ajudam a revelar tarefas que o usuário possa considerar difícil de executar, ou que possuam alguma linguagem confusa (GOODMAN; KUNIAVSKY; MOED, 2012).

O teste de usabilidade escolhido para ser aplicado foi o “*System Usability Scale*” (“escala de usabilidade do sistema”), ou SUS. Este teste foi desenvolvido em 1986 por John Brooke e foi, inicialmente, utilizado para testar a usabilidade em aplicações para terminais de computadores, mas atualmente é utilizado para as mais diversas tecnologias como: websites, hardwares e até mesmo páginas-amarelas (BANGOR; KORTUM; MILLER, 2008). Ele é um questionário composto de 10 perguntas padrões, com 5 opções de resposta cada (variando de “discordo completamente” à “concordo completamente”). Este teste, em específico, permite que seja calculada uma pontuação ao final de sua aplicação. A nota atribuída a cada resposta encontra-se na tabela abaixo.

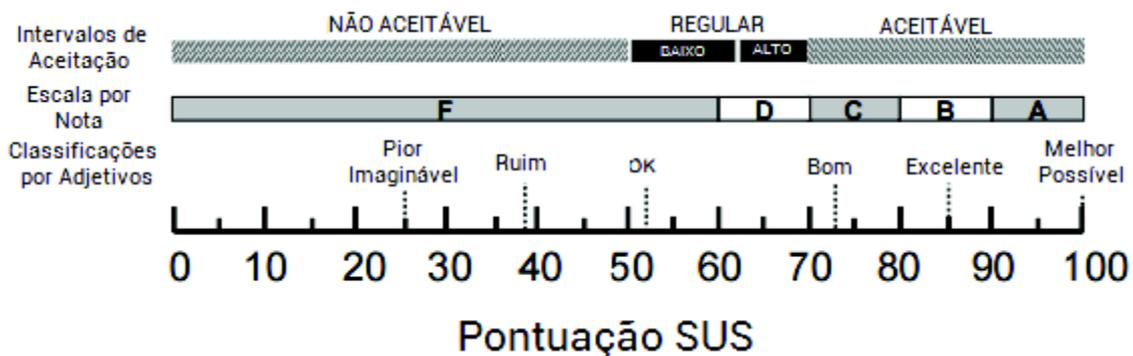
Tabela 2 – Pontuação SUS

Numeração da Questão	Avaliação				
	1 (Discordo Totalmente)	2	3	4	5 (Concordo Completamente)
Questões Ímpares	0	1	2	3	4
Questões Pares	4	3	2	1	0

Fonte: o autor.

O resultado de cada questionário é obtido através de uma soma simples dos valores de cada resposta assinalada e multiplicando-se este resultado por 2,5. A partir desta nota é possível atribuir um conceito geral à aplicação, conforme a imagem abaixo (BANGOR; KORTUM; MILLER, 2009).

Figura 12 - Pontuação SUS



Fonte: BANGOR; KORTUM; MILLER, 2009.

## 5 IMPLEMENTAÇÃO

Neste capítulo será fornecida uma visão geral da solução, incluindo uma breve descrição do sistema, detalhamento de todos os requisitos, diagrama de casos de uso, diagrama de classes e alguns *mockups*.

### 5.1 Sistema Proposto

O sistema proposto possui um objetivo principal, que é o de controlar a entrada de visitantes em empresas, permitindo que as visitas sejam cadastradas antecipadamente e para que possa ser fornecido um cartão inteligente (*smartcard*) para identificação dos visitantes.

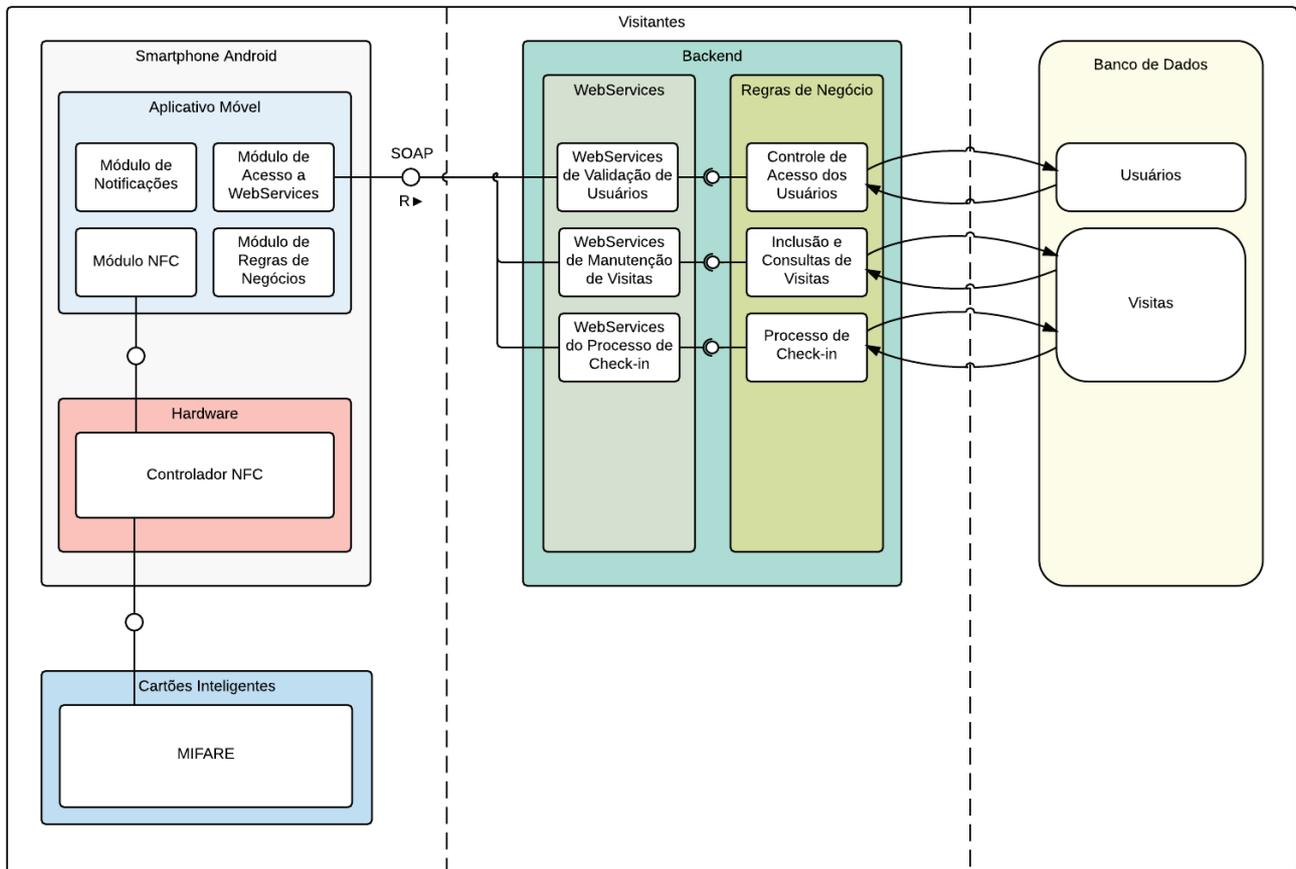
O sistema proposto consiste de uma aplicação móvel (desenvolvida para Android) e uma aplicação web (desenvolvida utilizando a tecnologia *ASP.NET*). A comunicação entre essas aplicações será realizada utilizando *Web Services*. Além disso também estará contemplado na solução o suporte à *smartwatches* com

*Android Wear* permitindo que as notificações sejam recebidas e respondidas no próprio relógio.

Os principais módulos do aplicativo móvel são: módulo com as regras de negócio, módulo NFC (que realiza a leitura do cartão MIFARE), módulo de notificações e o módulo que realiza a comunicação com o *backend*, utilizando *web services*.

Na parte de *backend* encontram-se os servidores da aplicação (*IIS - Internet Information Services*), o qual possui a parte *Web* da solução, junto com os *Web Services* de comunicação utilizados pela aplicação móvel, e o banco de dados (*SQL Server*).

Figura 13 - Estrutura da Aplicação



Fonte: o autor.

## 5.2 Requisitos Funcionais e Não-Funcionais

Baseado nas necessidades dos usuários finais deste sistema, os requisitos funcionais e não funcionais foram definidos, estes estão descritos abaixo:

### 5.2.1 Requisitos Funcionais:

O sistema deve permitir:

- Realizar uma pesquisa em todas as reservas já cadastradas, utilizando os seguintes parâmetros: data/hora inicial e final, nome da empresa, status da reserva, nome do visitante, documento de identificação do visitante.

- Realizar o *check-in* de uma reserva que já esteja cadastrada.

- Associar uma nova imagem ao visitante durante o *check-in*.

- A associação de um cartão MIFARE (utilizando o NFC) a um visitante durante o processo de *check-in*.

- Inclusão de visitantes adicionais a uma reserva já cadastrada, durante o *check-in*.

- Disponibilizar a visualização do histórico de visitas do Visitante durante o processo de *check-in*.

- Incluir uma nova reserva manualmente, realizando o *check-in* neste momento.

- Visualização de todas as reservas que já estejam cadastradas para a data atual e que estejam pendentes de entrada já na tela inicial do aplicativo.

- Permitir a busca da reserva através de um QRCode a ser lido através da câmera do dispositivo.

- Permitir que seja solicitada uma confirmação do visitado (pessoa que irá receber a visita) através de uma notificação. Esta notificação permitirá duas ações: “negar” e “aprovar” a entrada do visitante. Esta resposta irá retornar à recepcionista.

- Informar o visitado da entrada do visitante esperado através de uma notificação.

### 5.2.2 Requisitos Não-Funcionais:

- O sistema se comunicará com um servidor de aplicações através de *Web Services*.

- O sistema funcionará em um dispositivo com sistema operacional *Android* (versão mínima 4.0), com câmera e conectividade NFC.

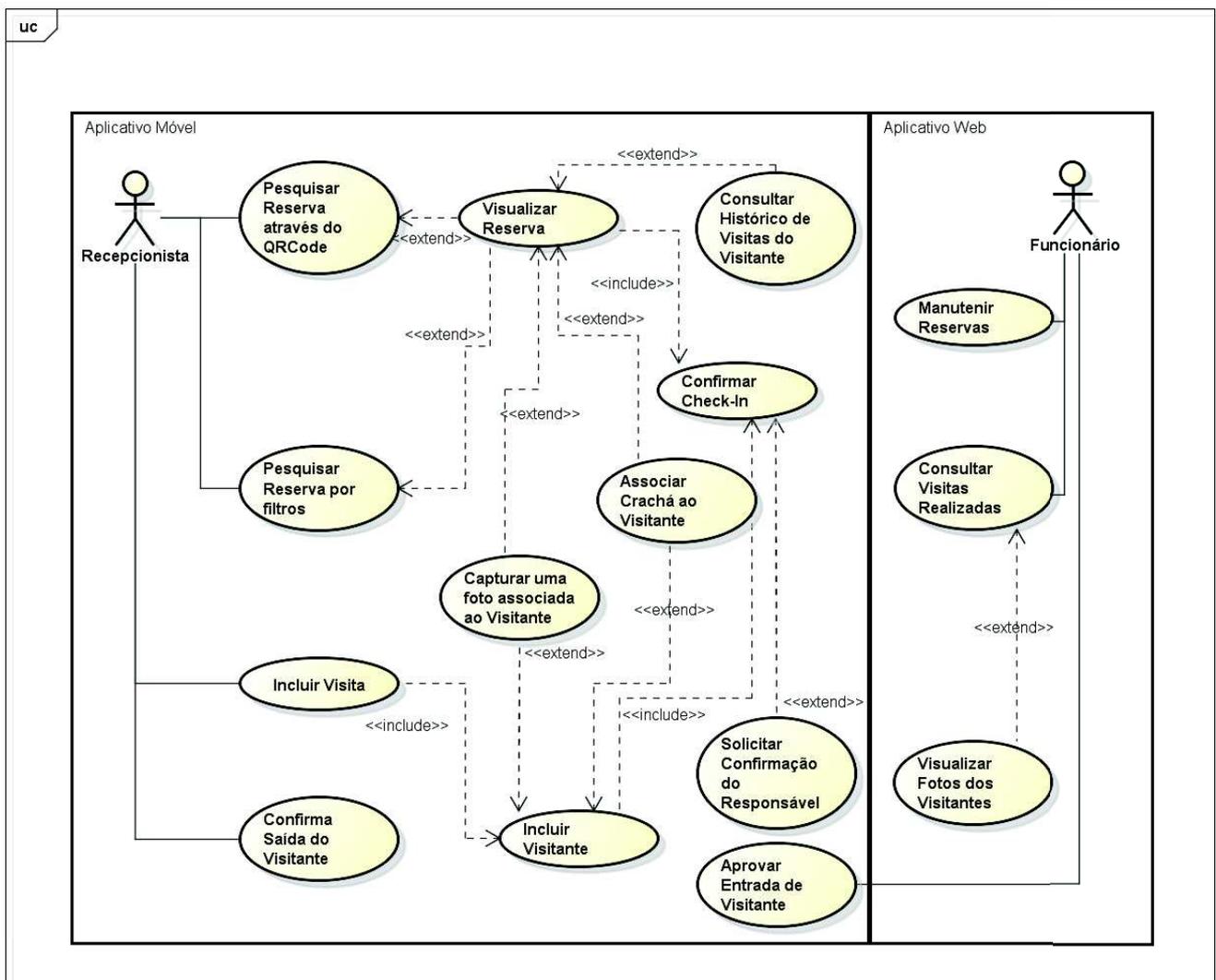
- O sistema deverá permitir a integração com dispositivos *Android Wear* para realizar as interações com as notificações da aplicação.

### 5.3 Casos de Uso

Nesta seção serão abordados os casos de uso da aplicação. Primeiramente será exibido o diagrama contendo todos os casos de uso, e logo após cada caso de uso será explicado separadamente.

Há dois atores principais que irão interagir com o sistema, são eles: Funcionário e Recepcionista (ou Porteiro).

Figura 14 - Casos de Uso



powered by Astah

Fonte: o autor

Ao Funcionário são fornecidas as seguintes ações: dar manutenção às reservas, ou seja, permite a este ator incluir, alterar, visualizar e excluir as mesmas. Para as visitas que já foram realizadas, é permitida a opção de visualizar a foto dos visitantes, caso elas tenham sido capturadas durante o processo de *check-in*. Todas

estas ações são realizadas via interface web. No aplicativo móvel é permitida ao funcionário a função de aprovar a entrada do visitante.

À Recepcionista são permitidas as seguintes ações: pesquisar a reserva através da leitura de um QRCode, realizar a pesquisa através de outros filtros, incluir uma nova visita ou confirmar a saída de algum visitante. Ao localizar a reserva desejada (ou ao iniciar o cadastramento de uma nova visita) são permitidas à Recepcionista as seguintes operações: consultar o histórico de visitas do visitante, capturar uma nova imagem associada ao visitante, associação de um *smartcard* ao visitante, confirmar a entrada do visitante (*check-in*).

Durante o processo de *check-in* pode ser solicitado ao responsável (funcionário) uma aprovação da entrada, que este deverá aprovar utilizando o aplicativo móvel.

Abaixo encontra-se uma tabela descritiva com todos os casos de uso citados no diagrama.

Tabela 3 – Casos de Uso.

Caso de Uso	Descrição
Pesquisar Reserva através do QRCode	Ao escolher esta opção no principal, uma nova tela com a imagem da câmera é aberta, permitindo a leitura de um QR Code com o número da reserva. Ao localizar um código, o aplicativo irá requisitar ao <i>Web Service</i> a reserva em questão. Caso encontre, irá para a tela de “Visualizar Reserva”.
Pesquisar Reservas por Filtros	Através desta opção a Recepcionista poderá realizar uma pesquisa na lista de reservas já cadastradas, utilizando os seguintes campos: data/hora de início, data/hora final, nome da empresa, código localizador, nome do visitante e documento do visitante. Após a consulta, será exibida uma lista com todas as reservas encontradas. Ao clicar em uma das reservas, irá para a tela de “Visualizar Reserva”.

Incluir Visita	É permitido ao usuário realizar a inclusão de uma nova visita, sem que ela esteja previamente cadastrada. Ao abrir a tela de inclusão, será necessário informar o nome da empresa e realizar o cadastramento dos visitantes. Após, o cadastramento de pelo menos um visitante é permitido ao usuário confirmar a entrada ( <i>check-in</i> ) destes visitantes.
Confirmar Saída do Visitante	Ao se escolher esta opção na tela inicial, será aberta para o usuário uma lista com todas as visitas que estejam atualmente em andamento. Ao escolher a visita desejada, será solicitada uma confirmação e então o visitante terá a sua saída confirmada.
Visualizar Reserva	Quando o usuário escolhe uma reserva, será aberta uma tela para visualização das seguintes informações da reserva: data/hora, nome da empresa, nome do responsável pela reserva, lista com os visitantes, contendo nome e documento de cada um. Dentro deste caso de uso é permitido ao usuário realizar uma ou mais destas ações: associar um crachá a qualquer visitante, visualizar o histórico de visitas do visitante, associar uma nova foto a algum visitante ou incluir mais algum visitante, caso seja necessário. Ao terminar de realizar todas as ações que julgar necessário, o usuário poderá, então, confirmar a entrada destes visitantes.
Capturar uma foto associada ao Visitante	Durante este caso de uso será permitido ao usuário a captura de uma foto do visitante utilizando a câmera do dispositivo. Ao capturar a foto, a mesma será associada ao visitante selecionado.
Incluir Visitante	Durante a visualização da reserva é permitido ao usuário realizar a inclusão de um novo visitante. Os campos solicitados e obrigatórios para inclusão do visitante são: nome e número do documento do

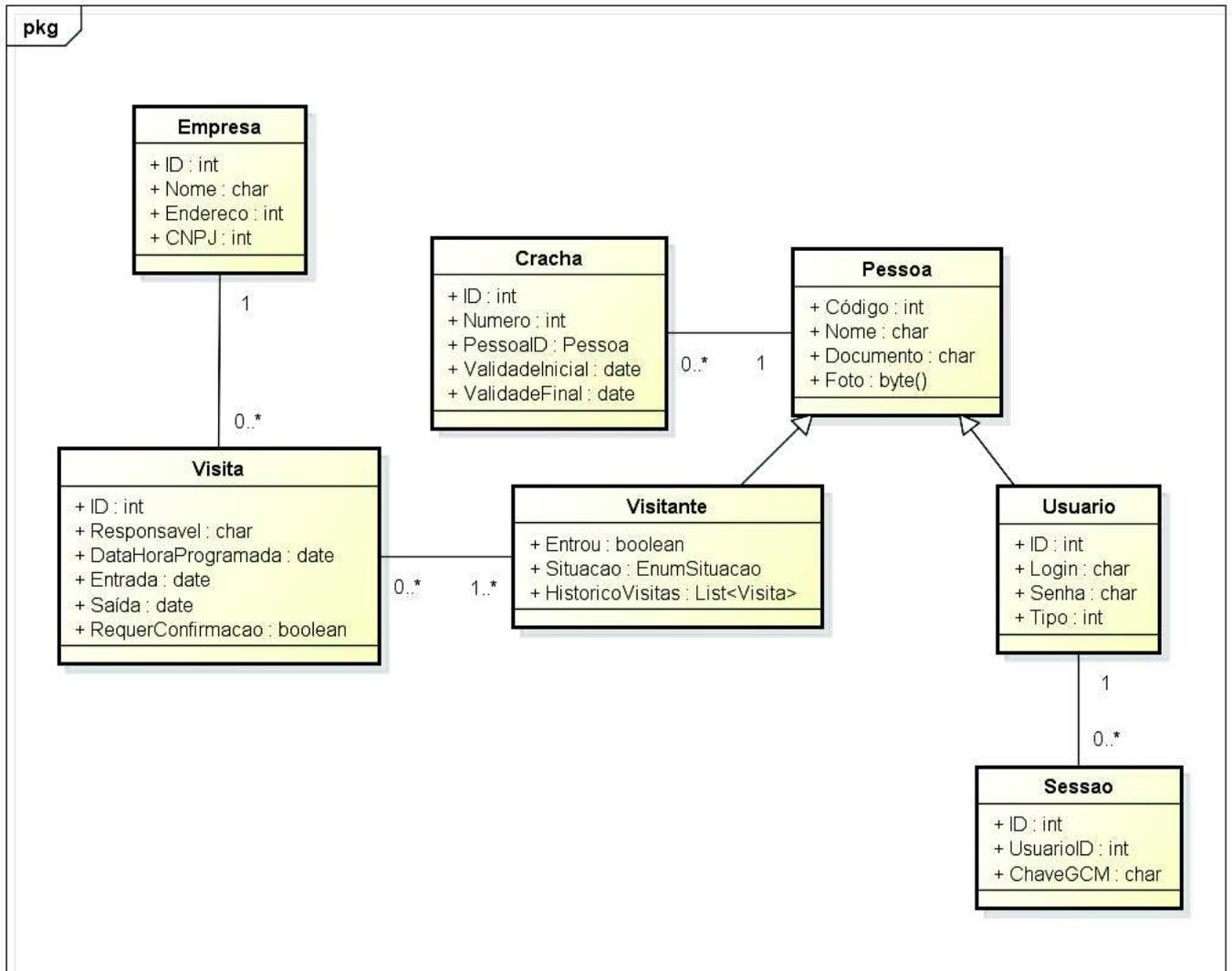
	mesmo.
Associar Crachá ao Visitante	Ao escolher esta opção será disponibilizada ao usuário a opção de associar um crachá a um visitante. Este crachá será associado ao realizar a aproximação de um smartcard MIFARE do leitor NFC do dispositivo. O UID ( <i>Unique Identifier</i> , identificador único) será lido automaticamente e será associado ao visitante. Essa associação permitirá ao sistema de controle de acesso realizar as validações necessárias para permitir, ou não, o acesso do visitante à determinados locais por portas e/ou catracas.
Consultar Histórico de Visitas do Visitante	Neste caso de uso será mostrada ao usuário uma lista com o histórico de visitas do visitante selecionado. Nesta lista os seguintes dados serão exibidos: data/hora de entrada, data/hora de saída, nome da empresa pela qual o usuário fez a visita.
Confirmar Check-in	Neste caso de uso será confirmada a entrada do visitante. Este processo será realizado através de uma requisição a um <i>Web Service</i> . Durante este processo poderá ser solicitada a confirmação do responsável.
Solicitar Confirmação do Responsável	Durante este caso de uso o responsável pela visita receberá uma notificação através da aplicação solicitando uma confirmação. As ações possíveis serão: “Aprovar” e “Rejeitar”.
Aprovar Entrada de Visitante	As ações permitidas ao Funcionário serão “Aprovar” e “Rejeitar”. Esta resposta retornará para a Recepcionista, e, no caso de “Confirmar”, o mesmo liberará o acesso e o check-in será confirmado.

Fonte: o autor.

## 5.4 Diagrama de Classes

Na figura abaixo vemos o diagrama de classes utilizado pela aplicação. Estas classes estão restritas ao aplicativo móvel, não contemplando a parte *web*.

Figura 15 - Diagrama de Classes



powered by Astah

Fonte: o autor

## 5.5 QRCode

O aplicativo móvel utilizará o QRCode para armazenar informações referentes à visita. Será utilizada a versão 1, que permite o armazenamento de até 10 caracteres alfanuméricos. Esta quantidade de caracteres será suficiente, pois será armazenado apenas um código localizador de 8 letras/números, que será utilizado para auxiliar na pesquisa e localização da reserva.

Figura 16 - Exemplo de QRCode

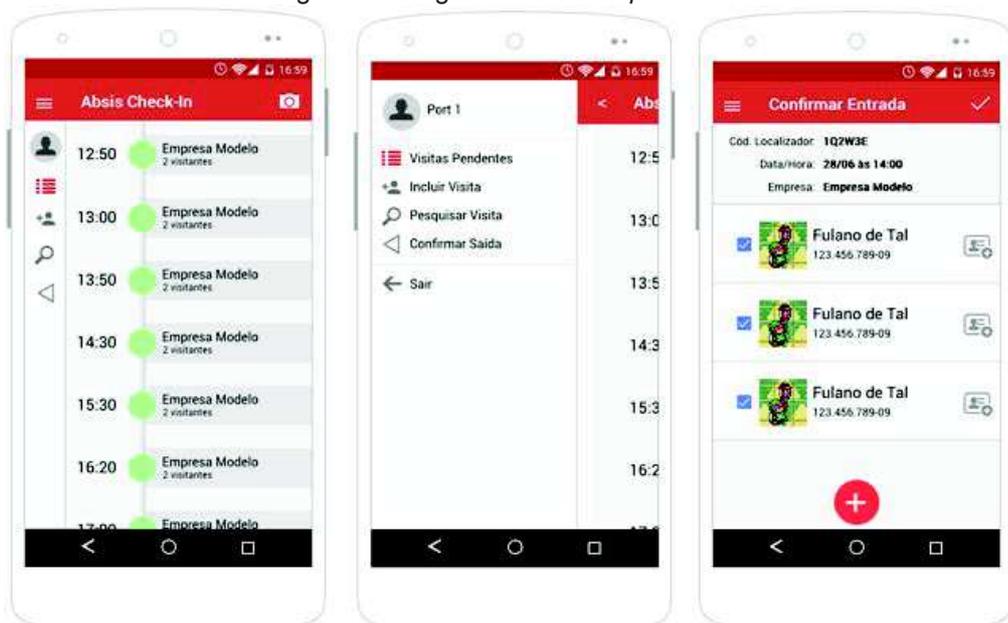


Fonte: o autor

A leitura deste código QR Code pelo aplicativo móvel será realizada pela biblioteca *ZXing* (uma biblioteca de código aberto, disponível em <https://github.com/zxing/zxing>, que fornece funcionalidades de leitura e criação de códigos de barras em geral, entre eles o QRCode).

## 5.6 Mockups

*Mockups* são modelos inventados ou simulados que imitam apenas certos aspectos do objeto ou conceito final. Eles são construídos para destacar partes ou funções essenciais (SOULIER, 1981). Durante a concepção da aplicação foram criados diversos *mockups* com o visual esperado da aplicação com o objetivo de realizar validações e testes de usabilidade com os usuários. Na figura abaixo podemos ver algumas das telas criadas durante esta etapa do desenvolvimento:

Figura 17 - Alguns dos *mockups* criados.

Fonte: o autor.

## 6 RESULTADOS

Baseado nos modelos e especificações criadas anteriormente as seguintes telas foram desenvolvidas:

Tabela 4 – Telas desenvolvidas

Tela	Descrição
Tela inicial da aplicação (Anexo A)	Nesta tela o usuário deverá informar o seu usuário e senha para acessar o aplicativo
Menu principal (Anexo B)	Exibe todas as visitas que estejam pendentes de entrada. No menu lateral são fornecidas as principais funcionalidades do sistema: “Visitas Pendentes”, “Incluir Visita”, “Pesquisar Visita” e “Confirmar Saída”. Na barra superior, no canto direito, fica disponível a opção de leitura de QRCode.
Menu principal com as opções abertas (Anexo C)	Expande o menu lateral, exibindo as opções disponíveis.
Opção Leitura de QR Code (Anexo D)	Tela que será aberta ao escolher a opção “Leitura de QRCode” no menu principal, irá fornecer a imagem da câmera traseira em tempo real para permitir a leitura de um QRCode. Ao localizar um QRCode será encaminhado para a tela “Informações sobre a Visita”
Informações sobre a Visita (Anexo E)	Ao se escolher uma visita, tanto pelo “Menu principal” quanto pela “Leitura de QRCode”, irão aparecer informações relacionadas à Visita, tais como: código localizador, data/hora programada da visita, nome da empresa e uma lista com todos os visitantes, incluindo suas fotos, caso existam. Nesta tela estarão disponíveis duas opções: confirmar <i>check-in</i> (botão localizado no canto superior direito) e incluir novos visitantes (botão localizado no canto inferior da tela).
Inclusão de novos visitantes	Exibe a opção para incluir novos visitantes à visita existente, sendo necessário informar o nome e o documento do visitante.

(Anexo F)	
Confirmação de entrada (Anexo G)	Ao escolher a opção “confirmar check-in” na tela “informações sobre a visita” aparecerá uma mensagem para confirmar a entrada, podendo o usuário cancelar e voltar para a tela anterior.
Solicitar confirmação por parte do visitado (Anexo H)	Caso a visita escolhida para realizar o <i>check-in</i> possua o atributo de “Solicitar Confirmação por parte do Visitado” assinalado (esta informação virá do cadastro realizado na parte <i>web</i> da aplicação) a aplicação irá solicitar ao usuário que ele realize a captura de uma foto dos visitantes para que o visitado a visualize e autorize/negue a entrada dos mesmos.
Mensagem de Espera (Anexo I)	Enquanto o visitado não realiza a ação de autorizar/negar esta mensagem ficará em exibição na tela.
Notificação do Visitado (Anexo J)	O visitado, caso a visita possua o atributo de solicitar confirmação, receberá esta notificação, com a foto capturada do visitante, já com as ações permitidas a ele: Negar e Autorizar.
Notificação do Visitado (Anexo K)	Simultaneamente a isso, o visitado irá receber a notificação no <i>smartwatch</i> , exibindo a foto capturada do visitante e também com as ações permitidas a ele. As telas são alternadas a partir de um <i>swipe</i> para a direita ou para a esquerda.
Notificação de Aviso do Visitado (Anexo L)	Logo após o visitante possuir sua entrada confirmada, o visitado irá receber uma notificação avisando que os visitantes entraram.
Opção Confirmar Saída (Anexo M)	Ao escolher a opção “Confirmar Saída” no menu principal serão exibidas todas as visitas que se encontram atualmente em andamento (que já possuem o registro de entrada, mas ainda não possuem o registro de saída).
Confirmação de saída (Anexo N)	Ao escolher alguma visita na tela anterior será exibida uma mensagem para confirmação da saída do registro selecionado.

Fonte: o autor.

Com a aplicação completamente desenvolvida foram escolhidas cinco pessoas, sendo três potenciais usuários da aplicação, para que os testes de

usabilidade fossem aplicados. Para estes testes foram definidas cinco ações que os usuários deveriam executar e, após isso, deveriam responder um questionário com as suas impressões sobre a aplicação. As ações escolhidas foram:

1. Localizar uma visita que esteja prevista para o dia de hoje, realizando a confirmação da entrada destes visitantes.
2. Incluir uma nova visita que não estivesse na lista de visitas previstas para o dia.
3. Realizar uma busca de uma visita a partir do número de documento do visitante.
4. Localizar uma visita através de um QRCode impresso.
5. Confirmar a saída de um visitante.

O resultado das avaliações encontra-se no quadro a seguir, as respostas variam de “1”, para “discordo completamente”, a “5”, para “concordo completamente”.

Tabela 5 – Questionário dos testes de usabilidade.

Questões		Avaliação				
		1	2	3	4	5
1	Eu gostaria de utilizar esta aplicação.	-	-	-	-	100%
2	Eu achei o sistema desnecessariamente complexo.	40%	20%	20%	20%	-
3	Eu achei o sistema fácil de utilizar.	-	-	20%	-	80%
4	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	60%	20%	-	20%	-
5	Eu achei que as funções do sistema estavam bem integradas.	-	-	-	20%	80%
6	Eu achei que havia muitas inconsistências no sistema.	60%	40%	-	-	-
7	Eu acredito que outras pessoas aprenderiam o sistema facilmente.	-	-	-	20%	80%
8	Eu achei o sistema muito incômodo para ser utilizado.	100%	-	-	-	-
9	Eu me senti confiante ao usar o sistema.	-	-	20%	-	80%
10	Eu precisei aprender muitas coisas novas antes de conseguir utilizar o sistema de forma satisfatória.	80%	20%	-	-	-

Fonte: o autor.

A partir destas respostas o método SUS permite que seja calculado um valor como resultado. Após o cálculo de todos os questionários, foi efetuada uma média aritmética simples, obtendo, assim uma pontuação final de 90.5, que corresponde a uma avaliação “Excelente” de acordo com Bangor, Kortum e Miller (2009).

Algumas questões foram possíveis de serem observadas durante o uso, tais como: ícone de “confirmar check-in” não estava claro o suficiente e alguns usuários demoraram a encontrá-lo; um usuário não conseguiu realizar a ação de “incluir visita” de forma independente, sendo necessário um auxílio; a maioria dos usuários não percebeu que o menu lateral expandia e permitia visualizar os textos das opções, mas, apesar disso, ainda assim a maioria conseguiu localizar as ações desejadas.

## 7 CONCLUSÃO

Este trabalho apresentou uma solução focada em otimizar o processo de recepção de visitantes em uma empresa. O desenvolvimento de uma aplicação para dispositivos móveis permitiu a utilização de novas tecnologias, como o NFC, QRCode e a captura de imagens de forma facilitada.

A utilização destas tecnologias nesta solução mostrou que ainda há espaço para a otimização de outros processos no nosso dia-a-dia. Poderíamos, por exemplo, utilizar o NFC dos *smartphones* para nos identificarmos na portaria de um prédio, ou nas entradas de nossas residências para a abertura das portas. Seria possível, também, uma adoção maior por parte dos QRcodes em lugar dos códigos de barras tradicionais, trazendo assim algumas vantagens, tais como: maior capacidade de armazenamento de informação, permitir que dados sejam lidos mesmo que haja alguma rasura, entre outras.

Ao utilizar a aplicação foi possível notar uma considerável redução no tempo de execução da operação de recepcionar um visitante, pois todas as visitas previstas estavam sempre visíveis e de fácil alcance. Além disso, ficou visível uma melhora na segurança já que existe a solicitação de uma aprovação por parte do visitado utilizando uma imagem capturada na hora do visitado. Outro ponto que a aplicação contribuiu foi em relação ao visitado poder receber uma notificação que o visitante entrou, dispensando assim a utilização de uma ligação por parte da

recepcionista, permitindo que seja possível receber o aviso mesmo que o visitado esteja longe de sua mesa.

Como trabalhos futuros, planeja-se realizar alterações na aplicação, com dois objetivos principais: permitir a utilização da aplicação em outras localidades, tais como condomínios, além de implementar a possibilidade do check-in ser realizado pelo próprio visitante, mantendo a aplicação sempre em execução em um *smartphone/tablet* que esteja colocado em um *stand*. Além disso planeja-se a integração com o sistema de controle de acesso da empresa, para permitir que seja realizado o bloqueio do visitante em catracas e/ou portas.

### APPLICATION FOR VISITORS' ENTRANCE CONTROL

**Abstract:** This paper presents a solution aiming to allow a more informative and dynamic control over visit control procedures. It was created for mobile devices that use *Android* operational system and relevant technologies to current devices, such as NFC, QRcodes, smartcards and device cameras. It also has integration with *Android Wear* devices, thus delivering a complete experience for users. Application usability tests were performed, which allowed the creation of an enjoyable and intuitive user interface, providing a simple way to control and authorize visits. Lastly, it was concluded that the application provided a better control over companies' visitors, allowing greater agility and security during those procedures.

**Keywords:** Android. Mobile Computing. Visitors' Control. Android Wear. NFC. Smartcards. QRcodes.

### REFERÊNCIAS

- AHSON, S. A.; ILYAS, M. **Near Field Communications Handbook**. Estados Unidos: CRC Press, 2012.
- ALLEN, S.; GRAUPERA, V.; LUNDRIGAN, L. **Pro Smartphone Cross-Platform Development**. Estados Unidos: Apress, 2010.
- ANDROID 2.3.3 APIs. **Android Developers**. Disponível em: <<http://developer.android.com/about/versions/android-2.3.3.html>>.
- BANGOR, A.; KORTUM, P. T.; MILLER, J. T. **An Empirical Evaluation of the System Usability Scale**. [S.l.]: International Journal of Human-Computer Interaction, 2008.
- BANGOR, A.; KORTUM, P.; MILLER, J. **Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale**. [S.l.]: Journal of Usability Studies, 2009.
- B'FAR, R. **Mobile Computing Principles**. Estados Unidos: Cambridge University Press, 2005.
- CECCHINATO, M. E.; BIRD, J.; COX, A. I. **Smartwatches: the Good, the Bad and the Ugly?** Seoul: [s.n.], 2015.
- CHIRICO, U. **Smart Card Programming**. Estados Unidos: lulu.com, 2014.
- COSKUN, V.; OK, K.; OZDENIZCI, B. **Near Field Communication (NFC): From Theory to Practice**. Reino Unido: John Wiley & Sons Ltd, 2012.

- DOBKIN, D. M. **The RF in RFID: UHF RFID in Practice**. Estados Unidos: Newnes, 2013.
- EVDOKIMOV, S. et al. **RFID and the Internet of Things: Technology, Applications, and Security Challenges**. Estados Unidos: now Publishers Inc., 2011.
- FURHT, B. **Handbook of Augmented Reality**. Estados Unidos: Springer, 2011.
- GLOVER, B.; BHATT, H. **RFID Essentials**. Estados Unidos: O'Reilly, 2006.
- GOODMAN, E.; KUNIAVSKY, M.; MOED, A. **Observing the User Experience: A Practitioner's Guide to User Research**. Estados Unidos: Elsevier, Inc., 2012.
- HOOG, A. **Android Forensics: Investigation, Analysis and Mobile Security for Google Android**. Estados Unidos: Syngress, 2011.
- IBM Knowledge Center. **IBM Knowledge Center**, 2015. Disponível em: <[http://www-01.ibm.com/support/knowledgecenter/SS5JSH\\_9.1.1/com.ibm.xtools.modeler.doc/topics/cdepd.html?lang=pt-br](http://www-01.ibm.com/support/knowledgecenter/SS5JSH_9.1.1/com.ibm.xtools.modeler.doc/topics/cdepd.html?lang=pt-br)>. Acesso em: 29 jun. 2015.
- IGOE, T.; COLEMAN, D.; JEPSON, B. **Beginning NFC: Near Field Communication with Arduino, Android & Phoneygap**. Estados Unidos: O'Reilly, 2014.
- INTENT | Android Developers. **Android Developers**, 2015. Disponível em: <<http://developer.android.com/reference/android/content/Intent.html>>. Acesso em: 27 jun. 2015.
- KATO, H.; TAN, K. T.; CHAI, D. **Barcodes for Mobile Devices**. Reino Unido: Cambridge University Press, 2010.
- LEDFORD, J.; ZIMMERLY, B.; AMIRTHALINGAM, P. **Web Geek's Guid to the Android-Enabled Phone**. Estados Unidos: Pearson Education, Inc., 2010.
- LEE, W.-M. **Beginning Android 4 Application Development**. Estados Unidos: John Wiley & Sons, Inc., 2012.
- MATEUS, G. R.; LOUREIRO, A. A. F. **Introdução à Computação Móvel**. Minas Gerais, Brasil: UFMG, 2010.
- MAYES, K. **Smart Cards, TOkens, Security and Applications**. Estados Unidos: Springer, 2008.
- MISHRA, S. M. **Wearable Android: Android Wear and Google FIT App Development**. Nova Jersey: John Wiley & Sons, Inc., 2015.
- NARDONE, M.; SILVA, V. **Discovering Android Wear**. Helsinki, Finlândia: Apress, 2015.
- NASUTION, S. M.; HUSNI, E. M.; WURYANDARI, A. I. **Prototype of Train Ticketing Application Using Near Field Communication (NFC) Technology on Android Device**. Bandung, Indonesia: School of Electrical Engineering and Informatics, Bandung Institute of Technology, 2012.
- NFC Forum Type Tags. **NFC Forum**, 2009. Disponível em: <[http://members.nfc-forum.org/resources/white\\_papers/NXP\\_BV\\_Type\\_Tags\\_White\\_Paper-Apr\\_09.pdf](http://members.nfc-forum.org/resources/white_papers/NXP_BV_Type_Tags_White_Paper-Apr_09.pdf)>. Acesso em: 20 jun. 2015.
- NFC Forum: What It Does. **NFC Forum**, 2015. Disponível em: <<http://nfc-forum.org/what-is-nfc/what-it-does/>>. Acesso em: 20 jun. 2015.
- OPEN Handset Alliance - Members. **Open Handset Alliance**, 2015. Disponível em: <[http://www.openhandsetalliance.com/oha\\_members.html](http://www.openhandsetalliance.com/oha_members.html)>. Acesso em: 20 jun. 2015.
- PEREIRA, L. C. O.; SILVA, M. L. D. **Android para desenvolvedores**. Rio de Janeiro: Brasport, 2009.
- QR CODE: Information capacity and versions of the QR Code. **QRCode.com**, 2015. Disponível em: <<http://www.qrcode.com/en/about/version.html>>. Acesso em: 21 jun. 2015.
- RANKL, W.; EFFING, W. **Smart Card Handbook**. Estados Unidos: John Wiley & Sons, Inc., 2010.
- SOULIER, J. S. **The Instructional Media Library**. Estados Unidos: Education Technology Publications, Inc., 1981.

TALUKDER, A. K.; AHMED, H.; YAVAGAL, R. R. **Mobile Computing:** Technology, Applications and Service Creation. 2<sup>a</sup>. ed. Delhi, Índia: Tata McGraw Hill Education Private Limited, 2010.

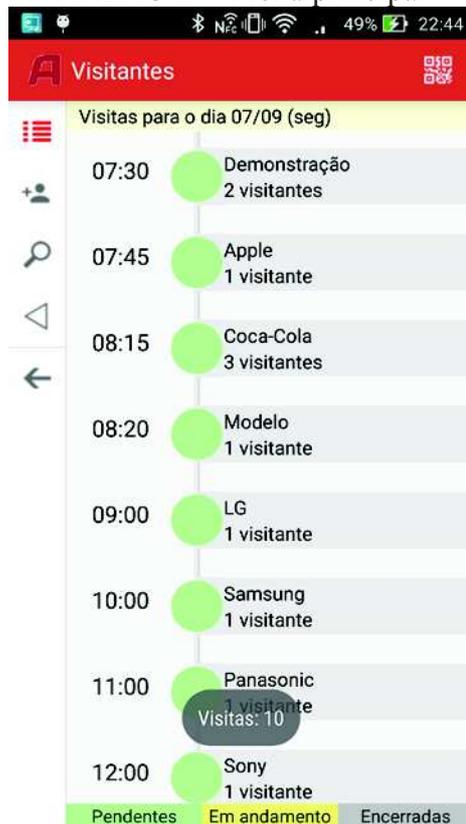
UDELL, S. **Pro Web Gadgets Across iPhone, Android, Windows, Mac, iGoogle and More.** Estados Unidos: Apress, 2009.

## ANEXOS

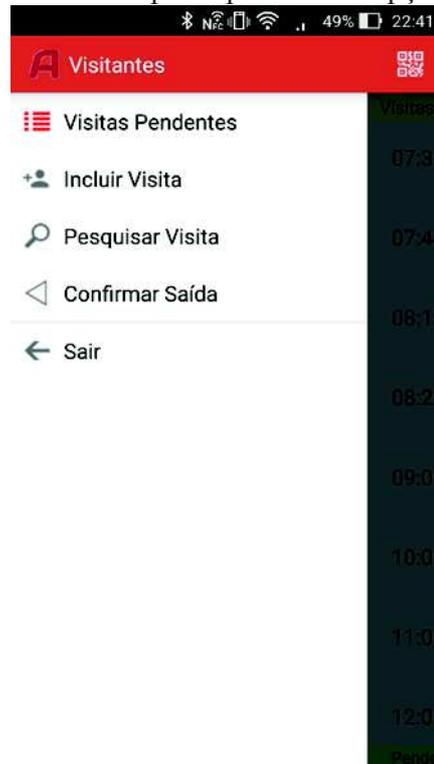
### ANEXO A – Tela inicial da Aplicação



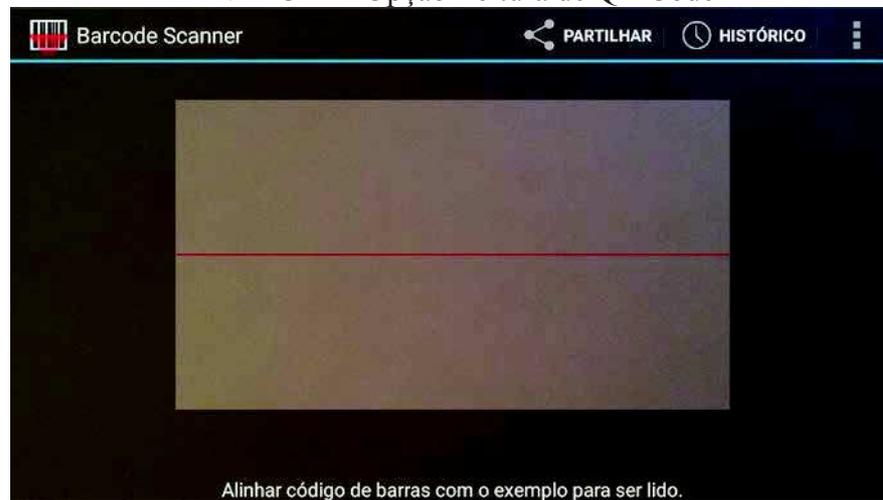
### ANEXO B – Menu principal



## ANEXO C – Menu principal com as opções abertas



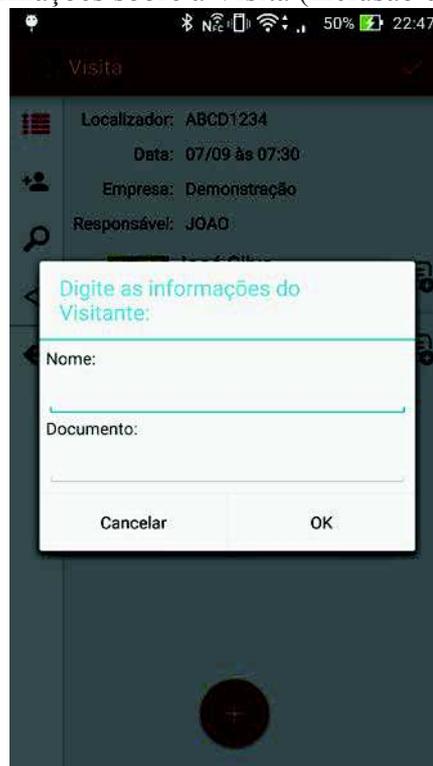
## ANEXO D – Opção Leitura de QR Code



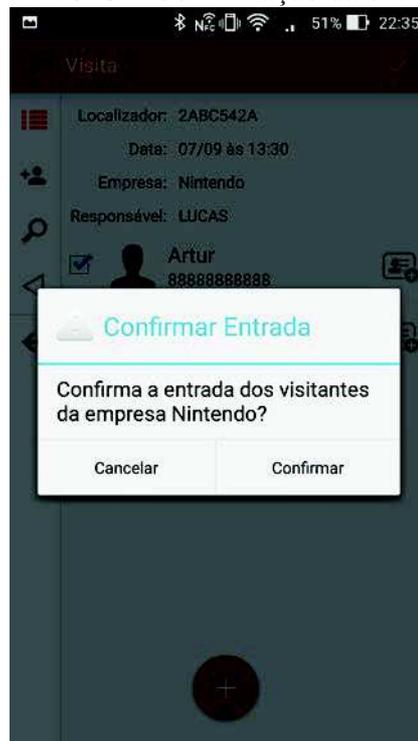
## ANEXO E – Informações sobre a Visita



## ANEXO F – Informações sobre a Visita (inclusão de novos visitantes)



## ANEXO G – Confirmação da entrada



## ANEXO H – Confirmação de Entrada



## ANEXO I – Mensagem de Espera



## ANEXO J – Notificação do Visitado



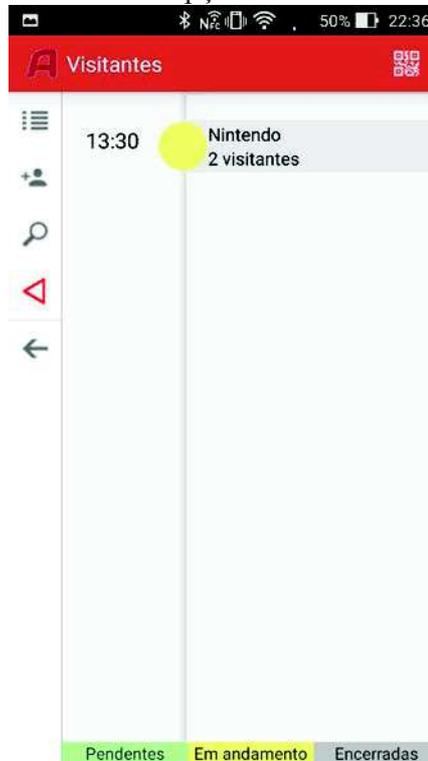
## ANEXO K – Notificação do Visitado (Smartwatch)



## ANEXO L – Notificação de Aviso do Visitado



## ANEXO M – Opção Confirmar Saída



### ANEXO N – Confirmação de Saída

