



UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

**SOLUÇÃO COLABORATIVA E GAMIFICADA PARA DISTRIBUIÇÃO DE
CARGAS DE MEDICAMENTOS: ESTUDO DE CASO DE APLICATIVO MÓVEL
COLABORATIVO**

FABRÍCIO TELES DUTRA GONÇALVES

São Luís, MA

2016

UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

FABRÍCIO TELES DUTRA GONÇALVES

**SOLUÇÃO COLABORATIVA E GAMIFICADA PARA DISTRIBUIÇÃO DE
CARGAS DE MEDICAMENTOS: ESTUDO DE CASO DE APLICATIVO MÓVEL
COLABORATIVO**

Monografia apresentada ao Curso de Engenharia da Computação da Universidade Estadual do Maranhão como pré-requisito para obtenção do título de Bacharelado em Engenharia de Computação. Sob orientação do Prof. MSc. Antonio Fernando Lavareda Jacob Junior.

São Luís, MA

2016

Gonçalves, Fabrício Teles Dutra

Solução colaborativa e gamificada para distribuição de cargas de medicamentos: estudo de caso de aplicativo movel colaborativo/ Fabrício Teles Dutra Gonçalves – São Luís, 2016.

95 f

Monografia (Graduação) – Curso de Engenharia de Computação, Universidade Estadual do Maranhão, 2016.

Orientador: Prof. Me Antonio Fernando Lavareda Jacob Junior

1. Solução colaborativa.2.Cargas de medicamentos.3. Aplicativo movel.I.Titulo.

CDU:004.4/.78:615.3

FABRÍCIO TELES DUTRA GONÇALVES

**SOLUÇÃO COLABORATIVA E GAMIFICADA PARA DISTRIBUIÇÃO DE
CARGAS DE MEDICAMENTOS: ESTUDO DE CASO DE APLICATIVO MÓVEL
COLABORATIVO**

Monografia apresentada ao curso de Bacharelado em Engenharia de Computação na Universidade Estadual do Maranhão como pré-requisito para obtenção do título de Bacharel.

Aprovada em ___/___/_____

MSc. Antonio Fernando Lavareda Jacob Junior
Orientador

Dr. Fernando Demétrio Cutrim
Primeiro membro

Dr. Reinaldo de Jesus da Silva
Segundo membro

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me concedido a vida e a capacidade para concluir este trabalho e concluir o curso.

Aos meus pais e meu irmão que sempre me apoiaram e compreenderam o esforço e dedicação investidos no curso, liberando o tempo necessário para a execução deste trabalho.

Ao Orientador pela paciência e esforço empregados neste trabalho, e todos os professores do curso, que foram tão importantes na minha vida acadêmica e no desenvolvimento desta monografia.

Aos meus amigos e colegas de curso pelo incentivo e ajuda no desenvolvimento desta monografia.

*“O sucesso nasce do querer, da determinação e
persistência em se chegar a um objetivo.
Mesmo não atingindo o alvo, quem busca e vence
obstáculos, no mínimo fará coisas admiráveis.”*
(José de Alencar)

LISTA DE FIGURAS

Figura 1 – Processo de solicitação e resposta.....	29
Figura 2 – Processo de requisição e resposta HTTP.	30
Figura 3 – Laço de eventos do Node.js.	32
Figura 4 – Diagrama representando o padrão MVC.....	34
Figura 5 – Arquitetura Android.	37
Figura 6 – Componentes da Camada HAL.....	38
Figura 7 – Arquitetura iOS.	39
Figura 8 – Ciclo de atualização com AngularJS.....	42
Figura 9 – Tecnologias utilizadas pelo framework NativeScript.	43
Figura 10 – Desenvolvimento Nativo x Cross-platform.	45
Figura 11 – Arquitetura geral da solução.	51
Figura 12 – Diagrama de caso de uso do sistema web.	54
Figura 13 – Diagrama de caso de uso do aplicativo.	56
Figura 14 – DER da base de medicamentos.	58
Figura 15 – DER de usuários e postos de saúde.....	59
Figura 16 – DER de perfil de acesso.	60
Figura 17 – DER de solicitação e autorização de solicitação.....	60
Figura 18 – DER de andamento de solicitação.....	61
Figura 19 – DER de atendimento e autorização de atendimento.	61
Figura 20 – DER de termo de recebimento/doação.....	62
Figura 21 – DER de gamificação.	62
Figura 22 – Tela de Login sistema web.....	63
Figura 23 – Tela principal sistema web.....	64
Figura 24 – Tela de atendimentos pendentes sistema web.....	65
Figura 25 – Tela de solicitações e atendimentos finalizados.....	65

Figura 26 – Tela de relação geral de solicitações e atendimentos do posto.	66
Figura 27 – Tela de login do aplicativo.	67
Figura 28 – Tela de seleção de posto de saúde.	67
Figura 29 – Tela de menu vertical acessado pelo funcionário.	68
Figura 30 – Tela principal acessado pelo funcionário.	69
Figura 31 – Tela de nova solicitação.	69
Figura 32 – Tela de seleção de quantidade.	70
Figura 33 – Tela de andamento de solicitação autorizada.	70
Figura 34 – Tela de andamento de solicitação sendo atendida.	71
Figura 35 – Tela de lista de solicitações de outros postos.	72
Figura 36 – Tela de atendimento de solicitação.	72
Figura 37 – Tela de histórico de solicitações.	73
Figura 38 – Tela de histórico de atendimentos.	73
Figura 39 – Tela de menu vertical acessado pelo administrador.	74
Figura 40 – Tela principal acessado pelo administrador.	75
Figura 41 – Tela de autorização de solicitação.	75
Figura 42 – Tela de lista de atendimentos pendentes de autorização.	76
Figura 43 – Tela de autorização de atendimento.	76
Figura 44 – Tela de cadastro de posto.	79
Figura 45 – Tela de cadastro de usuário.	79
Figura 46 – Tela de cadastro de usuário.	80
Figura 47 – Tela de nova solicitação.	80
Figura 48 – Tela de atendimento de solicitações.	81
Figura 49 – Tela inicial de aplicativo.	81
Figura 50 – Tela de seleção de posto.	82

LISTA DE TABELAS

Tabela 1 – Dinâmicas de jogo	22
Tabela 2 – Mecânicas de Jogo	23
Tabela 3 – Componentes de Jogo	24
Tabela 4 – Níveis de elementos de game design.	26
Tabela 5 – Vantagens e desvantagens do desenvolvimento nativo e multiplataforma.	46
Tabela 6 – Tabela de eventos.	57

LISTA DE ABREVIATURAS E SIGLAS

ANVISA – Agência Nacional de Vigilância Sanitária

API – *Application Programming Interface*

DER – Diagrama Entidade Relacionamento

IDE – *Integrated Development Environment*

JSON – *Javascript Object Notation*

MVC – *Model-View-Controller*

REST – *Representational State Transfer*

SDK – *Software Development Kit*

SQL – *Structured Query Language*

SUS – Sistema Único de Saúde

UML – *Unified Modeling Language*

UPA – Unidade de Pronto Atendimento

URI – *Uniform Resource Identifiers*

W3C – *World Wide Web Consortium*

XML – *Extensible Markup Language*

RESUMO

O uso de medicamentos é essencial para a manutenção da saúde da população, sendo o gerenciamento inadequado de medicamentos uma das causas de desperdícios e gastos de milhões dos cofres públicos. Diante desta problemática, o objetivo deste trabalho consiste no desenvolvimento e na implementação de um protótipo de aplicativo móvel e um sistema web como proposta para solução do problema de desperdícios de medicamentos, auxiliando no processo de distribuição e remanejamento dos mesmos. Para isso, foi realizado um estudo das principais tecnologias de desenvolvimento web e desenvolvimento mobile além de compreender como utilizar elementos de jogos para motivar usuários a utilizarem o sistema. Este estudo serviu de base para a definição do projeto de gamificação e de uma arquitetura que inclui um Web Service para troca de informações, o sistema web e o aplicativo móvel. Por meio de entrevistas in loco com profissionais das unidades de saúde, obteve-se a aceitação do trabalho como produto para auxílio à redistribuição de medicamentos.

Palavras-chave: Desperdício de Medicamentos. Web Service. Aplicativo. Sistema Web. Tecnologias. Redistribuição de Medicamentos.

ABSTRACT

The use of medicines is essential for maintaining population health and its inappropriate management is one of the reasons of wasting and expenses of millions from public resources. Having that in mind, the purpose of this work is the development and implementation of a prototype mobile application and web system as a proposal for solution of medicines wasting problem, helping in the distribution process. In this regard, a study was conducted to evaluate the current technologies of mobile and web development and to understand how to use games elements in order to motivate users. This study was important to define gamification project and the main architecture, which it is included a Web Service for information exchange, a web system and the mobile application. Through site visit interviews with professionals of healthcare units, acceptance of the work was obtained as a product for supporting medicines redistribution.

Keywords: Medicines Waste. Web Service. Mobile Application. Web System. Technologies. Medicines Redistribution.

SUMÁRIO

1. INTRODUÇÃO.....	14
1.1. Objetivos	15
1.1.1. Objetivo geral	15
1.1.2. Objetivos específicos	15
1.2. Metodologia	15
1.3. Estrutura do Documento	16
2. REVISÃO DE LITERATURA	17
2.1. Engenharia de Software	17
2.2. Usabilidade	18
2.3. Gamificação	20
2.3.1. Definição	21
2.4. Desenvolvimento Web.....	28
2.4.1. Arquitetura Cliente Servidor	29
2.4.2. Tecnologias de Desenvolvimento.....	30
2.4.3. Padrões de Projeto	33
2.4.4. Web Service.....	34
2.5. Desenvolvimento Mobile.....	35
2.5.1. Desenvolvimento Nativo	36
2.5.2. Desenvolvimento Cross-Platform.....	40
2.5.3. Comparativo entre Desenvolvimento Nativo e Cross-platform	44
3. DESENVOLVIMENTO	47
3.1. Formulação do problema	47
3.2. Proposta de Solução	49
3.3. Arquitetura Geral	50
3.4. Requisitos.....	52
3.5. Projeto de Gamificação	56
3.6. Modelagem do Banco de Dados	58
3.6.1. Base de Medicamentos	58
3.6.2. Usuários e Postos de Saúde	58
3.6.3. Perfil de Acesso	59

3.6.4. Solicitação	60
3.6.5. Atendimento	61
3.6.6. Gamificação.....	62
3.7. Prototipação.....	63
3.7.1. Sistema Web	63
3.7.2. Aplicativo	66
3.8. Memorial Descritivo	77
3.8.1. Base de Dados	77
3.8.2. Web Service.....	77
3.8.3. Sistema Web	78
3.8.4. Aplicativo	81
3.9. Resultados.....	82
4. CONSIDERAÇÕES FINAIS	84
4.1. Trabalhos Futuros.....	84
REFERÊNCIAS.....	85
APÊNDICE A – ROTEIRO DE ENTREVISTA	91
APÊNDICE B – SCRIPT PHP BASE DE DADOS	92

1. INTRODUÇÃO

Os avanços em alguns dos setores da saúde pública no Brasil foram significativos, principalmente os progressos tecnológicos da indústria farmacêutica, proporcionando medicamentos cada vez mais eficazes e seguros. A utilização de medicamentos tornou-se prática indispensável na contribuição para o aumento da qualidade e da expectativa de vida da população. (PORTELA et al., 2010).

A aquisição de medicamentos no setor público contribui para o sucesso e a credibilidade dos serviços farmacêuticos. Medicamentos definidos dentro de rigorosos critérios, armazenado em boas condições e administrados por profissionais capacitados, não atenderão às necessidades da rede de serviços se houver descontinuidade no suprimento de medicamentos. (MARIN et al., 2003).

A falta de medicamentos em unidades de saúde, por sua vez, é decorrente de problemas estruturais, organizacionais ou individuais, que permeiam as várias atividades dentro da vida útil dos mesmos. (MARIN et al., 2003).

A tecnologia de informação para dispositivos móveis em saúde tem se consolidado lentamente no mundo todo e através do desenvolvimento social e tecnológico, a internet tornou-se uma ferramenta para todas as instâncias profissionais. O mercado mobile está evoluindo constantemente e as soluções encontradas por empresas da área têm atendido a diversas necessidades de usuários.

A interatividade é um componente importante do design de produtos tecnológicos, tais como aplicativos móveis, destinados à mediação com seres humanos. O atual cenário tem como perfil usuários mais proativos com necessidade de responder às experiências de forma mais atuante e participativa, logo, o design de interação deve responder a esta questão ocasionando experiências agradáveis e significativas.

Considerando a interatividade como qualidade do produto e a interação representando a relação que ocorre entre o usuário e o ambiente, a utilização de mecânicas e técnicas adotadas no design de jogos tem por finalidade tornar os ambientes mais interativos e estreitar as comunicações entre usuário e sistema. A gamificação, portanto, tem como objetivo incrementar a interação do usuário, em contextos diversos, por meio da utilização de elementos de jogos.

A proposta deste trabalho é apresentar a solução para o problema de remanejamento de medicamentos em unidades de saúde a partir do desenvolvimento de aplicativo móvel e sistema web utilizando tecnologias atuais como ferramenta.

1.1. Objetivos

Nesse tópico, serão apresentados os objetivos gerais e os objetivos específicos deste trabalho.

1.1.1. Objetivo geral

Este trabalho tem por objetivo geral apresentar uma possível solução ao problema de distribuição e trocas de medicamentos entre unidades de saúde, utilizando um sistema informatizado e aplicativo móvel para solicitações e atendimentos tendo como drones o meio de transporte para as cargas dos medicamentos.

1.1.2. Objetivos específicos

- Utilizar metodologias da Engenharia de Software para levantamento de requisitos;
- Modelar abstração do sistema;
- Definir projeto de gamificação;
- Prototipação de telas;
- Desenvolvimento de base de medicamentos para auxílio no processo de solicitação e atendimentos;
- Desenvolvimento de Web Service, sistema web e aplicativo;

1.2. Metodologia

O trabalho iniciou com uma pesquisa a respeito do desperdício de medicamentos em postos e unidades de saúde. Nesse estudo foram definidos o tema, o contexto, os objetivos gerais e específicos e a estrutura do trabalho. Também foi utilizada a pesquisa exploratória para buscar metodologias e tecnologias que se enquadrassem no contexto do sistema a ser desenvolvido.

Essas metodologias de desenvolvimento, assim como as tecnologias, foram avaliadas e comparadas em suas principais características, verificando suas funcionalidades em relação a formulação do problema e proposta de solução.

A modelagem e desenvolvimento do sistema foi iniciado após a escolha das metodologias e tecnologias consideradas mais adequadas.

Na etapa de desenvolvimento do sistema será implementado um Web Service para a integração e troca de dados com o sistema web e o aplicativo móvel que proporcionará a centralização dos dados e uma possível integração com sistemas terceiros.

Após a solução alinhada, definida e executada, foi realizada entrevista com pessoas relacionadas ao domínio do problema para apresentação da proposta apresentada neste trabalho.

1.3. Estrutura do Documento

Este trabalho está estruturado da forma como segue. No Capítulo 2 é apresentada a revisão de literatura dos temas mais relevantes e de fundamental importância para a realização e elaboração da proposta apresentada neste trabalho. O Capítulo 3 descreve o processo de desenvolvimento da solução com a modelagem do sistema e prototipação. O Capítulo 4 são apresentadas as conclusões, resultados e considerações finais.

2. REVISÃO DE LITERATURA

2.1. Engenharia de Software

O software, muitas vezes associado a programas de computador, não representa apenas um programa, mas todos os dados de documentação e configuração associados e de fundamental importância para que o programa opere de forma correta. Um sistema de software é geralmente a execução de vários programas separados com vários arquivos de configuração utilizados para a configuração de cada programa, que descreve informações de versão e a estrutura do software e, por fim, a documentação do usuário que ilustra a forma de utilizar cada programa que constitui o sistema. (SOMMERVILLE, 2007).

Apesar de existir alguma semelhança entre o desenvolvimento de softwares e a fabricação de hardware, as duas atividades são diferentes. Em ambas, a qualidade é assegurada por um bom projeto, mas a fase de fabricação do hardware pode gerar problemas de qualidade, que são inexistentes de pessoas, mas a relação entre as pessoas envolvidas e o trabalho realizado é inteiramente diferente. As duas atividades requerem a construção de um produto, porém as abordagens são diferentes. Os custos do software são concentrados na engenharia. Isso significa que os projetos de software não podem ser geridos como se fossem projetos de fabricação uma vez que diferente do hardware, o software não se desgasta. (PRESSMAN, 2011).

A experiência inicial na construção de sistemas de software mostrou que o desenvolvimento informal de software não era suficiente. Projetos importantes apresentavam, algumas vezes, anos de atraso. O software, cujo custo superava as previsões, não era confiável, era difícil de manter e seu desempenho era insatisfatório. Novas técnicas e métodos eram necessários para controlar a complexidade inerente aos grandes sistemas de software. (SOMMERVILLE, 2007).

Engenharia de Software é a metodologia de desenvolvimento e manutenção de sistemas modulares, com as seguintes características: processo (roteiro) dinâmico, integrado e inteligente de soluções tecnológicas, adequação aos requisitos funcionais do negócio do cliente e seus respectivos procedimentos pertinentes; efetivação de padrões de qualidade, produtividade e efetividade em suas atividades e produtos; fundamentação na Tecnologia da Informação disponível, viável, oportuna e personalizada; planejamento e gestão de atividades, recursos, custos e datas. (REZENDE, 2005).

Maffeo (1992 apud REZENDE, 2005) considera que os objetivos primários da Engenharia de Software são o aprimoramento da qualidade dos produtos de software e o aumento da produtividade dos engenheiros de software, além do atendimento aos requisitos de eficácia e eficiência, ou seja, efetividade.

A metodologia deve auxiliar o desenvolvimento de projetos, sistemas ou software, de modo que os mesmos atendam de maneira adequada às necessidades do cliente ou usuário, com os recursos disponíveis e dentro de um prazo ideal definido em conjunto com os envolvidos. Não deve limitar a criatividade profissional, mas deve ser um instrumento que determine um planejamento metódico, que harmonize e coordene as áreas envolvidas. O que limita a criatividade não é a metodologia, mas os requisitos de qualidade, produtividade e efetividade de um projeto. (REZENDE, 2005).

Assim como a metodologia, os processos de software ou processos de desenvolvimento de software também são roteiros de elaboração de software que devem contemplar fases, subfases, produtos externados e pontos de avaliação de qualidade. (REZENDE, 2005).

Segundo Sommerville (2007), existem quatro atividades de processo fundamentais comuns a todos os processos de software. Essas atividades são:

1. Especificação do software: A funcionalidade do software e as restrições em sua operação devem ser definidas.
2. Desenvolvimento do software: O software deve ser produzido de modo que atenda às suas especificações.
3. Validação do software: O software tem de ser validado para garantir que ele faz o que o cliente deseja.
4. Evolução do software: O software deve evoluir para atender às necessidades mutáveis do cliente.

2.2. Usabilidade

Existem diversos atores envolvidos no desenvolvimento e uso dos sistemas computacionais interativos: fabricantes de hardware, de software, vendedores, profissionais de suporte e manutenção, provedores de acesso à internet, produtores de conteúdo, usuários, organizações, dentre outros. Todas essas partes interessadas costumam ser denominadas

stakeholders. Cada um enxerga a tecnologia sob um ponto de vista diferente, enfatizando alguns aspectos em detrimento de outros. (BARBOSA, 2010).

A usabilidade é um termo que começou a ser utilizado pela Ciência Cognitiva e depois pela Psicologia e Ergonomia em substituição ao termo “amigável” (DIAS, 2006). A usabilidade é um conjunto de propriedades de uma interface que reúne os seguintes componentes: fácil aprendizado, eficiência, capacidade de memorização, baixo índice de erros, satisfação e prazer ao uso (NIELSEN, 1994). Ainda de acordo com Nielsen e Loranger (2007) usabilidade é um atributo de qualidade relacionado à facilidade de uso de algo. Refere-se a rapidez com que os usuários podem aprender a usar alguma coisa, a eficiência deles ao usá-la, o quanto lembram daquilo, seu grau de propensão a erros e o quanto gostam de utilizá-la.

Para um usuário tirar proveito do apoio computacional oferecido pelo sistema, não podem existir barreiras que o impeçam de interagir com sua interface assim como deve estar claro para o usuário o que o sistema se propõe a fazer, e conseqüentemente, se usuário tiver uma boa compreensão das ferramentas do sistema a partir do design, ele terá melhor condição de fazer um uso produtivo e criativo do apoio computacional oferecido pelo sistema. (BARBOSA, 2010).

A norma ISO/IEC 9241-11 (2003) define alguns termos que são amplamente utilizados quando se trata de usabilidade:

- **Eficácia:** é a capacidade que os sistemas conferem a diferentes tipos de usuários para alcançar seus objetivos em número e com a qualidade necessária.
- **Eficiência:** é a quantidade de recursos que os sistemas solicitam aos usuários para a observação dos objetivos com o sistema.
- **Satisfação:** é a emoção que os sistemas proporcionam aos usuários em face dos resultados obtidos e dos recursos necessários para alcançar tais objetivos.
- **Tarefa:** Objetivo a ser atingido ou um resultado a obter.
- **Usuários:** Pessoas que utilizam alguma instância do sistema.
- **Contexto:** Conjunto de elementos, onde se destacam: o ambiente físico, a tecnologia utilizada e a motivação.

2.3. Gamificação

A ideia de usar elementos de design de games em contextos que não são voltados para entretenimento, como saúde, gestão, economia, tem ganhado um alto crescimento de sua utilização em design de interações. Com a denominação de “Gamificação”, o objetivo principal é criar o engajamento de usuários aumentando as atividades realizadas e a apreensão dos mesmos às funcionalidades em sistemas computacionais. Inúmeras empresas já oferecem a Gamificação como camada de software de serviço de recompensas e sistemas de reputação com pontos, medalhas, níveis e tabelas de liderança. (DETERDING et al, 2011).

Complementando a definição proposta por Deterding (2011), Snijders, Dalpiaz e Brikkemper (2015) reafirmam que o uso dos elementos de design para sistemas gamificados devem ser característicos de design de jogos e possuem um papel importante na jogabilidade. A variedade de aplicações é abrangente, desde aplicativos móveis no qual o usuário é perseguido por zumbis, a medalhas com títulos de “expert” em websites colaborativos de perguntas e respostas como o Stack Overflow.

A gamificação de problemas reais costuma ser referenciada como Serious Games, ou Jogos Sérios. Em um exemplo conhecido de gamificação, a Universidade de Washington desenvolveu o jogo Foldit (UNIVERSITY OF WASHINGTON, 2008), com o objetivo de envolver os usuários na solução de um único problema: tentar compreender como determinada proteína poderia ser utilizada no combate à Aids. A atividade atraiu 46 mil participantes que em sua maioria não tinham qualquer ligação com a área médica e em apenas 10 dias conseguiram esclarecer um enigma que tomou 15 anos de vários cientistas sem a obtenção de nenhum resultado. (VIANNA et al, 2013).

Marins (2013 apud ZICHERMAN; CUNNINGHAM, 2011) ressalta que a gamificação não é apenas inserir elementos de jogos como medalhas em um produto, é preciso ter uma abordagem mais profunda. Mecânicas de jogos não irão corrigir problemas fundamentais de um negócio, não vão reconstruir uma infraestrutura deficiente e nem resolver o problema de um péssimo serviço ao cliente. Porém, transformar experiência em jogo pode produzir uma grande mudança comportamental. Sendo aplicada da forma correta, com uma abordagem cuidadosa, a gamificação pode ser bastante poderosa.

2.3.1. Definição

A gamificação, originado do termo *gamification* em inglês, corresponde ao uso de mecanismo de jogos orientados ao objetivo de resolver problemas práticos ou de despertar engajamento entre um público específico. (VIANNA et al, 2013).

Uma definição simples para a gamificação é “o uso de elementos de jogos e técnicas de game design em contextos não relacionados a jogos”. (MARINS, 2013 apud WERBACH; HUNTER, 2012).

Para detalhar a definição, é preciso conceituar jogo, elementos de jogos, design e ainda o que se entende por aplicações que não são do contexto de jogos.

Ao falar de jogos é importante saber que a definição de gamificação está relacionada a games (jogos) e não a play (brincadeira). O termo play pode ser definido como uma categoria mais ampla e diferente de game. Play apresenta um formato mais livre, mais improvisado, e até mesmo mais “tumultuado”. Games, diferentemente de play, são caracterizados por sistemas de regras explícitas e a competição de atores nesse sistema através de objetivos. Deterding (2011) explica a diferença entre os termos para concluir que o termo gamificação está diretamente relacionado a game.

Embora a maioria de exemplos atuais de gamificação seja digital, o termo não deve ser limitado a tecnologia digital. Não apenas a convergência da mídia e computação ubíqua estão confundido a distinção entre digital e não-digital: as definições de games e game design são por si só categorias transmediais. (DETERDING et al, 2011 apud JUUL, 2005).

Ao passo que jogos sérios descrevem o design pleno de jogos com propósitos não voltados a entretenimentos, aplicações “gamificadas” simplesmente incorporam elementos de jogos. Deterding (2011) afirma que a linha que separa ”jogos” e “artefato com elementos de jogos” pode ser tênue em certas situações.

Essa afirmação nos leva a outro questionamento: Que elementos pertencem ao conjunto de “elementos de jogos”? Os elementos de jogos são definidos como pequenos pedaços que constroem um jogo. Avatar, ambientes 3D, narrativa, feedback, reputações, rankings e níveis são exemplos de elementos encontrados em jogos. Porém, esses elementos também podem ser

encontrados fora dos mesmos, até mesmo em aplicações não “gamificadas”. (MARINS, 2013 apud WERBACH; HUNTER, 2012).

Werbach e Hunter (2012) identificaram três tipos de elementos (dinâmicas, mecânicas e componentes) como categorias aplicáveis aos estudos e desenvolvimento da gamificação. Tais categorias são organizadas em ordem decrescente de abstração de modo que cada mecânica se liga a uma ou mais dinâmicas, e cada componente a uma ou a mais mecânicas ou dinâmicas.

As dinâmicas de jogos representam um nível mais alto de abstração de elementos do jogo como ilustra a Tabela 1. São os temas em torno do qual o jogo se desenvolve, assim como aspectos do quadro geral do sistema de jogo levados em consideração, mas que não fazem parte diretamente do jogo. Esses elementos mostram quais são as forças subjacentes que existem em jogos. (MARINS, 2013 apud WERBACH; HUNTER, 2012).

Tabela 1 – Dinâmicas de jogo

Dinâmicas	Descrição
Emoções	Jogos podem criar diferentes tipos de emoções, especialmente a da diversão (reforço emocional que mantém as pessoas jogando)
Narrativa	Estrutura que torna o jogo coerente. A narrativa não tem que ser explícita, como uma história em um jogo. Também pode ser implícita, na qual toda a experiência tem um propósito em si
Progressão	Ideia de dar aos jogadores a sensação de avançar dentro do jogo
Relacionamentos	Refere-se à interação entre os jogadores, seja entre amigos, companheiros ou adversários
Restrições	Refere-se à limitação da liberdade dos jogadores dentro do jogo

Fonte: COSTA; MARCHIORI, 2016 apud WERBACH; HUNTER, 2012.

Portanto, as dinâmicas representam as interações entre o jogador e as mecânicas de jogo e compõem os aspectos do quadro geral de uma gamificação. Devem ser gerenciadas, mas não explicitadas obrigatoriamente no jogo. (WERBACH; HUNTER, 2012).

As mecânicas se referem aos elementos mais específicos que levam às ações também mais específicas. Elas orientam as ações dos jogadores em uma direção desejada delimitando que o jogador pode ou não fazer dentro do jogo. (ERIKSSON; MUSIALIK; WAGNER, 2012).

A Tabela 2 detalha as mecânicas e suas respectivas conceituações segundo Werbach e Hunter (2012).

Tabela 2 – Mecânicas de Jogo

Mecânicas	Descrição
Aquisição de recursos	O jogador pode coletar itens que o ajudam a atingir os objetivos
Avaliação	A avaliação permite que os jogadores vejam como estão progredindo no jogo
Chance	Os resultados de ação do jogador são aleatórios para criar uma sensação de surpresa e incerteza
Cooperação e competição	Cria-se um sentimento de vitória e derrota
Desafios	Os objetivos que o jogo define para o jogador
Recompensas	O benefício que o jogador pode ganhar a partir de uma conquista no jogo
Transações	Significa compra, venda ou troca de algo com outros jogadores no jogo
Turnos	Cada jogador no jogo tem seu próprio tempo e oportunidade para jogar. Jogos tradicionais, como jogos de cartas e jogos de tabuleiro muitas vezes dependem de turnos para manter o equilíbrio no jogo, enquanto muitos jogos de computador modernos trabalham em tempo real
Vitória	O “estado” que define ganhar o jogo

Fonte: COSTA; MARCHIORI, 2016 apud WERBACH; HUNTER, 2012.

As mecânicas auxiliam no funcionamento do jogo e orientam as ações do jogador, além de terem uma variedade ampla de estilos, dependendo da mecânica utilizada. (COSTA; MARCHIORI apud ZICHERMANN; CUNNINGHAM, 2011). Vários mecanismos podem estar incluídos em uma dinâmica tais como avaliação e recompensas. Dessa forma, cada mecânica é uma forma de atingir uma ou mais das dinâmicas descritas na Tabela 1. Um evento aleatório, como por exemplo um prêmio que aparece sem aviso, pode tanto estimular o senso de diversão e curiosidade dos jogadores como ser uma forma de obter novos participantes ou manter os jogadores mais experientes envolvidos. (COSTA; MARCHIORI apud WERBACH; HUNTER, 2012).

Costa e Marchiori (2016), ainda se baseando nas categorias de elementos de jogos identificadas por Werbach e Hunter (2012), definem componentes como aplicações específicas visualizadas e utilizadas na interface do jogo (Tabela 3). Os componentes representam o nível mais concreto dos elementos dos jogos e, assim como uma mecânica se liga com uma ou mais dinâmicas, vários componentes podem fazer parte de uma mecânica.

Tabela 3 – Componentes de Jogo

Componentes	Descrição
Avatar	Representação visual do personagem do jogador
Bens virtuais	Itens dentro do jogo que os jogadores podem coletar e usar de forma virtual e não real, mas que ainda tem valor para o jogador. Os jogadores podem pagar pelos itens ou moeda do jogo ou com dinheiro real
Boss	Um desafio geralmente difícil no final de um nível que tem de ser derrotado, a fim de avançar no jogo
Coleções	Formadas por itens acumulados dentro do jogo. Emblemas e Medalhas são frequentemente parte de coleções
Combate	Disputa que ocorre para que o jogador derrote oponentes em uma luta

Conquistas	Recompensa que o jogador recebe por fazer um conjunto de atividades específicas
Conteúdos desbloqueáveis	A possibilidade de desbloquear e acessar certos conteúdos no jogo se os pré-requisitos forem preenchidos. O jogador precisa fazer algo específico para ser capaz de desbloquear o conteúdo
Emblemas/medalhas	Representação visual de realizações dentro do jogo
Gráfico Social	Capacidade de ver amigos que também estão no jogo e ser capaz de interagir com eles. Um gráfico social torna o jogo uma extensão de sua experiência de rede social
Missão	Similar a “conquistas”. É uma noção de jogo de que o jogador deve fazer executar algumas atividades que são especificamente definidas dentro da estrutura do jogo
Níveis	Representação numérica da evolução do jogador. O nível do jogador aumenta à medida que o jogador se torna melhor no jogo
Pontos	Ações no jogo que atribuem pontos. São muitas vezes ligadas a níveis
Presentes	A possibilidade distribuir ao jogador coisas como itens ou moeda virtual para outros jogadores
Ranking	Lista jogadores que apresentam as maiores pontuações/conquistas/itens em um jogo
Times	Possibilidade de jogar com outras pessoas com mesmo objetivo

Fonte: COSTA; MARCHIORI, 2016 apud WERBACH; HUNTER, 2012.

Um desses elementos sozinho não pode ser considerado como um elemento específico para jogos. Os componentes acima podem assumir diferentes combinações, e essa escolha deve ser feita com base no que atende mais adequadamente as demandas de um determinado contexto. Combinar as dinâmicas, mecânicas e componentes de forma que sejam efetivas para

um determinado objetivo é a tarefa central de um projeto de gamificação. (WERBACH; HUNTER, 2012).

Deterding (2012) salienta ainda que “gamificar” não é a simples adição de recompensas e incentivos com pontos e medalhas dentro de um sistema e que, se aplicado dessa forma, os projetos gamificados estão destinados a fracassar.

Como visto, aplicações “gamificadas” não são apenas instâncias nos quais elementos de jogos tem sido atribuídos propósitos diferentes. Conceitualmente, o termo gamificação é mais utilizado como o uso de game design (design de jogo) do que como tecnologias baseadas em jogos ou práticas utilizadas no ramo de jogos. (DETERDING et al, 2011).

Deterding (2011) cita que em pesquisas realizadas na literatura, a respeito de jogos e gamificação, foi encontrado que os elementos de game design variam em níveis de abstração. Ordenado do concreto ao mais abstrato, 5 (cinco) níveis podem ser identificados de acordo como descrito na Tabela 4.

Tabela 4 – Níveis de elementos de game design.

Dinâmicas	Descrição	Exemplo
Padrões Design de Interface de Jogos	Design de interações de componentes e soluções para um problema conhecido em um contexto, incluindo prototipação	Medalhas, nível, quadro de liderança
Padrões de Design e Mecânicas de Jogos	Recorre a partes do design que influenciam na jogabilidade	Limites de tempo, recursos limitados
Princípios de Design e Heurística de Jogos	Guias avaliativas para abordar o problema de design ou analisar uma determinada solução de design	Metas claras, variedades de estilo do jogo

Modelos de Jogos	Modelos conceituais dos componentes de jogos ou experiência do jogo	Desafios, fantasia, curiosidade
Design de Métodos de Jogos	Práticas e processos específicos do design de jogos	Testes de Jogabilidade

Fonte: DETERDING et al, 2012.

Como pode ser analisado, o modelo de níveis proposto por Deterding (2011) faz a distinção entre padrões de design de interfaces e padrões de design de jogos ou mecânicas de jogos. Diferentemente dos padrões de interface de jogos, nem a mecânica de jogos nem os padrões de design de jogos se referem a soluções prototipadas, ambos podem ser implementados com diferentes elementos de interface. São, portanto, mais abstratos e, dessa forma, são tratados como distintos.

Observando da perspectiva do designer, o que difere gamificação de jogos de entretenimento ou jogos sérios é que o primeiro é desenvolvido com a intenção de ser um sistema que inclui elementos de jogos, e não um jogo propriamente dito. Da perspectiva do usuário, tais sistemas que englobam elementos de jogos podem ser experimentados como jogos, contendo motivação. (DETERDING et al, 2011).

Como descrito anteriormente, parte da definição de gamificação se atém a aplicar elementos de jogos em contextos que não referem a jogos, isto é, exclui o uso desses elementos como parte de um projeto de jogo. Portanto, para Deterding (2011), não se pode limitar o uso da gamificação para contextos específicos pois o mesmo pode ser utilizado em diversas áreas tais como treinamento, saúde e educação. Além disso, em Bunchball (2012) é ressaltado que a gamificação pode ser aplicada no meio empresarial, tanto na fidelização de clientes como no aumento de produtividade dos funcionários.

Concluindo, quando uma solução ou aplicação mistura elementos de jogos, design de jogos e o propósito principal não é que os usuários joguem um jogo, isso é gamificação. (MARINS apud WERBACH, 2012).

2.4. Desenvolvimento Web

Segundo Pressman (2011), a qualidade de uma aplicação móvel ou web pode ser definida em termos de usabilidade, funcionalidade, confiabilidade, eficiência, facilidade de manutenção, segurança, escalabilidade e tempo para colocação no mercado. Para que se possa alcançar tais atributos de qualidade, um bom projeto deve apresentar as seguintes características: simplicidade, consciência, identidade, robustez, navegabilidade e apelo visual. De tal maneira, o projeto deve ser estruturado em algumas fases que irão proporcionar uma melhor estruturação.

Na fase de interface, descrevemos a estrutura e organização da interface do usuário e é abrangido uma representação do layout da tela, uma definição dos modos de iteração e uma descrição dos mecanismos de navegação. A fase de projeto estético (ou gráfica) descreve o “aspecto” e inclui combinação de cores, layout geométrico, tamanho, fonte, posicionamento de textos entre outros. (PRESSMAN, 2011).

Pressman e Lowe (2009) listam as melhores práticas para o desenvolvimento de aplicações web:

- 1) O tempo deve ser utilizado para entender o objetivo do produto e sua necessidade de negócio;
- 2) A interação dos usuários com a aplicação deve ser descrita por meio de cenários;
- 3) Um plano de projeto deve ser desenvolvido, mesmo que breve;
- 4) Deve ser realizada modelagem de pontos mais obscuros do projeto;
- 5) Os modelos devem ser revistos quanto à consistência e qualidade;
- 6) A tecnologia deve permitir que os componentes utilizados para a construção possam ser reutilizados quantas vezes forem possíveis;
- 7) Utilizar design patterns desenvolvidos para reutilização de componentes;
- 8) Os testes devem ser criteriosos antes da liberação para implantação.

Esta seção abordará as tecnologias voltadas para o desenvolvimento de aplicações Web que são amplamente utilizadas em projetos atuais.

2.4.1. Arquitetura Cliente Servidor

A Web consiste em milhões de clientes e servidores, conectados através de redes cabeadas e sem fio como as redes wireless. Quando um determinado usuário acessa uma determinada Uniform Resource Locators (URL) pelo navegador, que representa o cliente, uma solicitação é enviada para o servidor Web, que representa o servidor. O servidor normalmente responde para o cliente enviando a página Web solicitada (DEITEL, 2004).

Um servidor Web recebe uma solicitação e devolve algo para o cliente. O browser ou navegador permite ao usuário solicitar um recurso. O servidor recebe a solicitação, encontra o recurso e devolve a resposta para o cliente. Um recurso pode ser uma página Web, figura, imagem ou vídeo. O cliente solicita um recurso através da URL e o servidor o disponibiliza. Se o recurso solicitado não for encontrado pelo servidor uma página de erro “404 Not Found” é retornada como resposta para o cliente. O processo de solicitação e resposta pode ser visualizado pela Figura 1.

Figura 1 – Processo de solicitação e resposta.



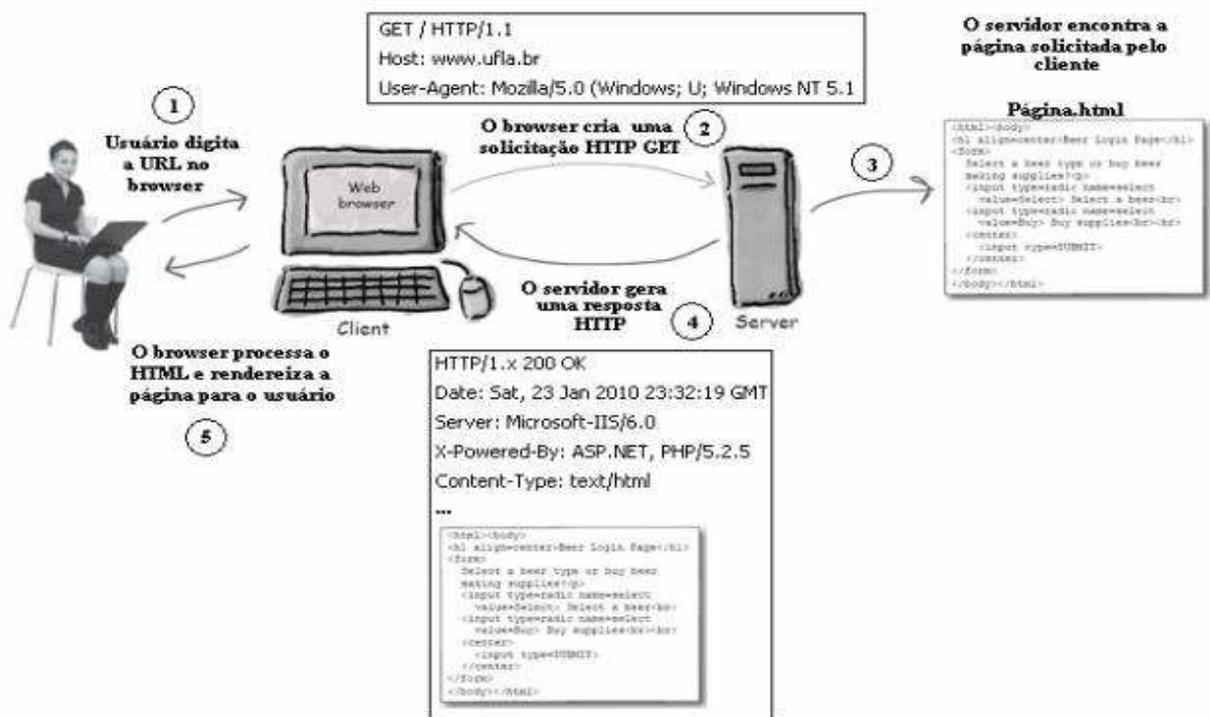
Fonte: BASHAM et al, 2008.

Na Web os clientes e servidores se comunicam através do protocolo HTTP (Hypertext Transfer Protocol). O TCP (Transmission Control Protocol – Protocolo de Controle de Transmissão) e o IP (Internet Protocol – Protocolo da Internet) são conhecidos coletivamente por TCP/IP. A definição do protocolo TCP pode ser encontrada no Request For Comments ou RFC 793. Dentre os serviços que o TCP disponibiliza podemos citar o serviço da internet orientado a conexão, ou seja, a aplicação pode confiar que todos os dados serão entregues sem erro e na ordem certa. Já o protocolo IP especifica o formato da informação enviada e recebida entre os roteadores, que representam os sistemas intermediários de comunicação, e os

hospedeiros como os computadores pessoais tradicionais, servidores, Web TV's, computadores portáteis, pagers e até mesmo torradeiras (KUROSE, 2003).

O HTTP é executado no topo das camadas e possui características específicas para Web, mas ele depende do protocolo TCP/IP para obter a solicitação e a resposta completa de um lugar para o outro. O processo completo de requisição e resposta pode ser visualizado pela Figura 2.

Figura 2 – Processo de requisição e resposta HTTP.



Fonte: BASHAM et al, 2008.

2.4.2. Tecnologias de Desenvolvimento

HTML

O Hyper Text Markup Language, mais conhecido como HTML, consiste em uma linguagem de marcação utilizada para produção de páginas web que podem ser interpretadas em praticamente qualquer tipo de computador e transmitidas pela internet. O desenvolvimento de códigos em HTML pode ser feito em arquivo de texto normal e visualizado em um navegador web, comumente conhecido como browser.

Como outras linguagens, o HTML já passou por diversas versões e processos de padronizações com o passar dos anos. Após esse tempo, a linguagem adquiriu características

marcantes em reflexo a evolução. Cada nova versão tinha como objetivo: adicionar novas características, melhorar a interoperabilidade, tornas a ferramenta compatível com a maioria dos browsers, plataformas e dispositivos do mercado, e ser acessível a todos os tipos de usuários.

CSS

A tecnologia web conhecida por CSS (Cascading Style Sheets) é uma linguagem de estilização que possibilita a criação de folhas de estilo no qual é definido o estilo de cada elemento HTML, isto é, é definido o layout da página web de uma forma eficiente, facilitando a alteração e manutenção do mesmo.

A estilização ocorre por meio de definição de regras que estão associadas a cada elemento que aparece na página web. Essas regras são responsáveis pela forma que um desses elementos são renderizados.

JavaScript

O JavaScript é uma linguagem de script incorporada ao documento HTML que oferece interação com a página modificando os elementos e estilos de forma dinâmica, permitindo estruturar o comportamento da aplicação web.

A linguagem surgiu como linguagem de script do lado do cliente (client-side) de páginas web. É uma linguagem dinâmica e utiliza o paradigma de orientação a objetos. Devido a facilidade de interação com o DOM (Document Object Model), os scripts desenvolvidos em JavaScript são amplamente integrados nas páginas web.

A ferramenta AJAX (Asynchronous JavaScript and XML) se utiliza da linguagem JavaScript e permite a manipulação do DOM de uma forma transparente pelo lado do cliente, sem a necessidade de carregamento da página. Ao usar AJAX no desenvolvimento de serviços para web, a informação é carregada de maneira mais simples e precisa. O usuário não precisa aguardar que a página seja totalmente recarregada quando faz uma requisição.

Node.js

O Node.js é uma plataforma desenvolvida a partir da engine JavaScript V8 da Google. Esta plataforma permite a construção de aplicações em rede escaláveis de forma trivial e

produtiva. Para isto, oferece um modelo orientado a eventos assíncronos com um esquema de recursos “não bloqueante”, que o faz uma solução leve e eficiente para aplicações em tempo real com intenso uso de dados. Diferente de várias plataformas open-source, o Node.js é fácil de inicializar e não requer recursos da máquina em excesso, como memória ou espaço de disco (NODE, 2016).

Atualmente, existem várias tecnologias consolidadas para front-end e back-end. O papel do Node.js é oferecer uma interface consistente entre estas duas tecnologias. O Node.js exerce seu trabalho através de um laço de eventos. O laço funciona da seguinte forma: (i) carrega e executa o programa; (ii) inicializa o laço de eventos; (iii) espera algum evento ser disparado; (iv) executa os manipuladores de eventos; (v) finaliza o processo, se o mesmo estiver ocioso. Sempre que um evento ocorrer, o Node.js irá executar as callbacks que estão escutando aquele evento. Apresenta-se na figura a seguir o laço de eventos do Node.js (RIBEIRO, 2015).

Figura 3 – Laço de eventos do Node.js.



Fonte: Adaptação de (WILSON, 2013).

As plataformas tradicionais que oferecem serviços web exigem que para cada requisição, uma thread seja criada. Cada nova thread consome uma quantidade específica de memória RAM. Uma aplicação contendo um alto fluxo de requisições necessitaria de uma infraestrutura consistente para funcionar adequadamente. O Node.js funciona de uma forma diferente, ele opera em uma única thread utilizando chamadas de entrada e saída não

bloqueantes. Assim, permitindo milhares de requisições concorrentes e poupando, consideravelmente, o consumo dos recursos computacionais (RIBEIRO, 2015).

Por padrão o Node.js é utilizado para conexões HTTP long-polling ou streaming. Com o advento do protocolo WebSockets, o Node.js precisava de uma ferramenta que conseguisse unir as duas tecnologias, com isso surgiu o Socket.IO (NODE, 2016a).

2.4.3. Padrões de Projeto

Segundo Gama (2000), um padrão de projeto são as descrições de objetos e classes que se comunicam e precisam ser personalizadas para resolver um problema geral de projeto num contexto particular. A essência de um padrão é uma solução reutilizável para um problema recorrente.

Gamma (2000) afirma que um padrão de projeto deve nomear, abstrair e identificar os aspectos chave de uma estrutura de projeto comum, que será útil para o desenvolvimento de um projeto orientado a objetos reutilizável. O padrão de projeto identifica as classes e instâncias participantes, seus papéis, colaborações e a distribuição de responsabilidades. Cada padrão tem um foco em um determinado problema ou um tópico particular de projeto orientado a objetos. Ele descreve em que situação pode ser aplicada, se ele pode ser aplicado em função de outras restrições de projeto e as consequências, custos e benefícios de sua utilização. Quando o projeto for implementado, um padrão de projeto deve fornecer exemplos de código para ilustrar a implementação do padrão.

A escolha da linguagem de programação é importante porque influencia o ponto de vista do projetista (usuário do padrão). Essa escolha determina o que pode ou não ser implementado facilmente. Alguns padrões podem ser expressos mais facilmente em uma linguagem que em outras.

Model View Controller

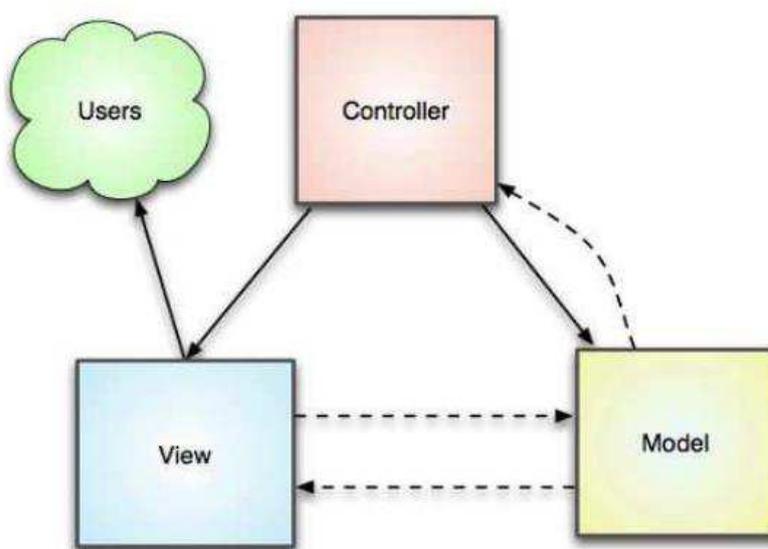
MVC é a abreviatura de Modelo-Visão-Controlador (Model-View-Controller), um padrão desenvolvido pelo Centro de Pesquisas da Xerox de Palo Alto (Xerox PARC) para a linguagem Smalltalk em 1979. (REENSKAUG, 1979) (SOUZA, 2007).

Afim de manter a manutenibilidade, aplicações web mais complexas precisaram realizar uma separação entre os dados (Model) e o layout (View). Assim, mudanças realizadas no layout

das páginas não afetam a manipulação de dados, e estes poderão ser reorganizados sem alterar o layout. O MVC resolve este problema através da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois: o controlador (Controller). (PRADO, 2015)

A Figura 4 exemplifica a relação entre Model, View, Controller e Usuários. As linhas sólidas indicam associação direta e as tracejadas associação indireta.

Figura 4 – Diagrama representando o padrão MVC.



Fonte: PRADO, 2015.

Os elementos da View (Visão) fazem uma representação das informações do Model (Modelo) para os usuários, a partir das quais os mesmos podem interagir com a aplicação. Essa interação é tratada pelo Controller (Controle) que pode alterar o status dos elementos do Model. Este que pode notificar a View tais alterações, fazendo com que esta última possa atualizar a informação apresentada ao usuário. (PRADO, 2015).

2.4.4. Web Service

Menéndez (2009) define Web Services como aplicações que aceitam solicitações de outros sistemas através da Internet.

Estas aplicações fornecem serviços, os quais são baseados em um conjunto de padrões da Internet definidos pelo W3C e não requerem configurações especiais nos firewalls, uma vez

que o protocolo HTTP atua como transporte na comunicação entre cliente e Web Service. (RECKZIEGEL, 2006).

A arquitetura de um Web Service é baseada na interação entre três personagens: Provedor de Serviços, Consumidor de Serviços e o Registro dos Serviços. As interações envolvem as operações de publicação, pesquisa e ligação. (KREGGER, 2001). Cada personagem possui funções como segue:

- **Provedor de serviços:** Entidade que cria o Web Service disponibilizando o serviço para que possa ser utilizado. (RECKZIEGEL, 2006).
- **Consumidor de serviços:** Qualquer cliente que utilize o Web Service criado por um provedor de serviços. O cliente conhece a funcionalidade do Web Service a partir da descrição disponibilizada pelo provedor de serviços. (LOPES, 2008).
- **Registro dos serviços:** Localização central na qual o provedor de serviços pode relacionar os seus serviços web e então o consumidor de serviços pode pesquisar serviços Web. (BASHA, et al, 2002).

2.5. Desenvolvimento Mobile

Segundo Pressman (2011), para que um software seja bem projetado, é necessário que os parâmetros sejam corretamente analisados e alinhadas, parâmetros tais como, escopo do software, riscos envolvidos, recursos necessários, tarefas necessárias, esforços e custos aplicados. Embora o desenvolvimento tradicional de aplicações tenha semelhanças com o desenvolvimento de aplicações móveis, este último apresenta características que o torna ligeiramente mais complexo, como por exemplo: crescente variedade de plataformas (Android, IOS, Windows Mobile, Firefox OS, BlackBerry, etc.), diferentes modelos de dispositivos, resolução e tamanho do display, diferentes configurações de memória e processador, consumo de bateria, entre outros.

De início, desenvolver aplicações para determinada plataforma, utilizando todo o ambiente da plataforma escolhida, era a única opção. Para cada plataforma existia uma linguagem específica. Segundo Prezotto e Boniati (2014), essa forma de desenvolvimento ficou conhecida como desenvolvimento nativo e consiste em um aplicativo projetado e desenvolvido especificamente para uma plataforma.

Com o aumento da demanda por aplicativos, surgiu a necessidade de desenvolvimento rápido para diferentes plataformas, aumentando o custo e a dificuldade de manutenção do software, visto que cada plataforma tinha um ambiente de desenvolvimento diferente das demais, o que conseqüentemente demandava da equipe diferentes áreas de conhecimento para desenvolver e manter inúmeros aplicativos. (PREZOTTO; BONIATI, 2014).

A partir dessa demanda, surgiu a necessidade da criação de apenas um aplicativo que pudesse ser publicado ou executado em diferentes plataformas. Essa forma de desenvolvimento é conhecida como cross-platform ou multiplataforma. (HEITKOTTER; HANSCHKE; MAJCHRZAK, 2013)

Segundo El-Kassas et al. (2015), algumas particularidades a serem consideradas no desenvolvimento mobile são:

- **Limitações de recursos:** As configurações de processamento e de armazenamento dos dispositivos móveis são limitadas.
- **Ecosistema heterogêneo:** Por existir diferentes sistemas operacionais, o desenvolvimento de aplicações móveis encontra-se em um contexto heterogêneo, implicando na necessidade de diferentes versões de uma mesma aplicação.
- **Experiência de uso:** As aplicações devem ser simples e possuir uma interface amigável que facilite a utilização e atenda às expectativas do usuário.
- **Manutenção:** Manutenções e atualizações constantes são necessárias para que a aplicação continue operando de forma correta.

2.5.1. Desenvolvimento Nativo

Aplicações nativas são desenvolvidas através do uso de ferramentas e linguagens específicas para determinada plataforma, utilizando o ambiente de desenvolvimento e SDK por ela providos. Os aplicativos ficam vinculados a esse ambiente, executando os dispositivos da plataforma escolhida. (EL-KASSAS et al., 2015).

Segundo Jobe (2013), Android e iOS são duas das plataformas mais conhecidas e utilizadas para desenvolvimento nativo de aplicações mobile, que serão descritas nas seções a seguir.

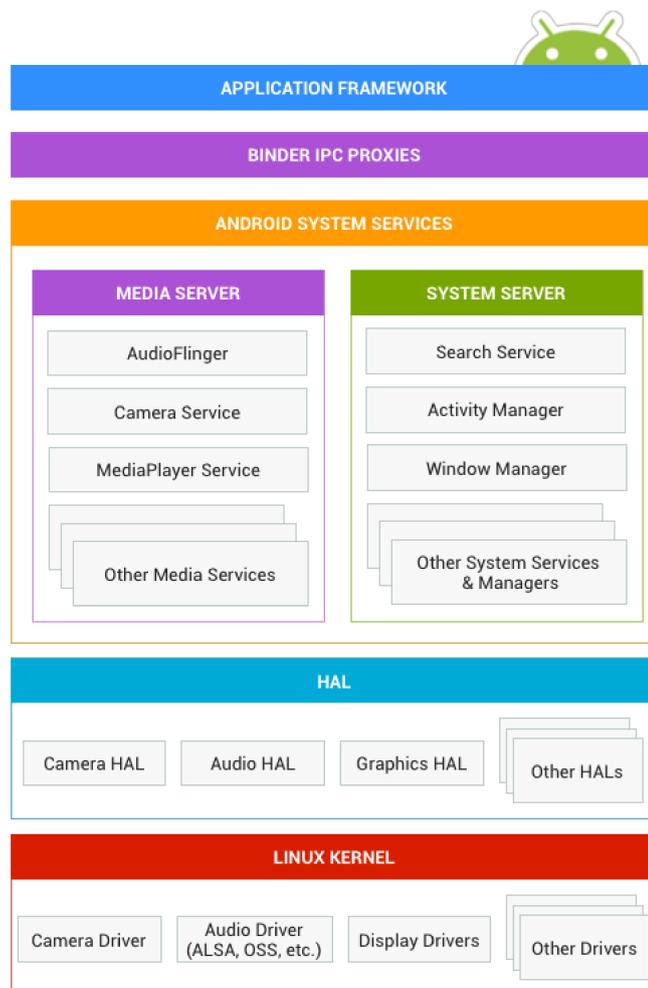
2.5.1.1.Android

O sistema operacional Android foi desenvolvido para aparelhos móveis e possui como base o Linux. Inicialmente, o projeto foi mantido pela empresa Android Inc., que em 2005 foi adquirida por um grupo de empresas denominado Open Handset Alliance, cujo o objetivo é fornecer um sistema operacional gratuito, completo e open source. (WOLFSON, 2013).

Arquitetura Android

A arquitetura do sistema Android é baseada na pilha de camadas de alto nível da Figura 5.

Figura 5 – Arquitetura Android.

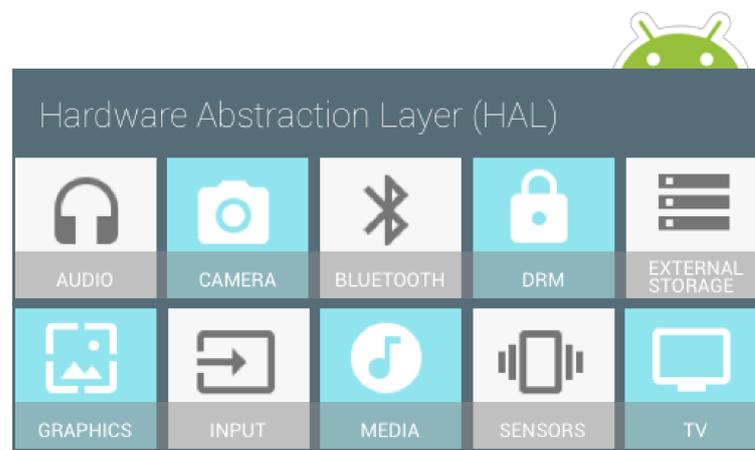


Fonte: ANDROID, 2016.

- **Application Framework:** A camada mais elevada na qual as aplicações desenvolvidas pelos desenvolvedores são instaladas. (ANDROID, 2016).

- Binder IPC Proxies: Camada que permite ser feita uma ponte entre a camada superior, application framework, e a próxima camada, system services. Essa característica possibilita que APIs de alto nível interajam com serviços do sistema. (ANDROID, 2016).
- System Services: APIs da camada de aplicação se comunicam com a camada de serviços para acessar os hardwares específicos. O sistema Android inclui dois grupos de serviços: system (serviços que envolvem notificações) e media (serviços que envolvem mídia tais como gravações). (ANDROID, 2016).
- Hardware Abstraction Layer (HAL): Permite que fornecedores de hardware criem interfaces e drivers que os mesmos oferecem, sendo possível então criar novas funcionalidades sem afetar as outras camadas do sistema. A Figura 6 abaixo ilustra alguns dos componentes da camada em questão. (ANDROID, 2016).

Figura 6 – Componentes da Camada HAL.



Fonte: ANDROID, 2016.

- Linux Kernel: Camada que representa a versão do kernel do Linux que possui algumas modificações como gerenciamento de memória, processo, energia, comunicação e logs. (ANDROID, 2016).

Desenvolvimento Android

Para o desenvolvimento de aplicações Android, pode ser utilizado diversas linguagens de programação, mas a linguagem Java é a mais utilizada tendo em vista que a plataforma de desenvolvimento Android é desenvolvida baseada em Java. (WOLFSON, 2013).

Em relação ao sistema operacional, não há nenhuma configuração específica como requisito inicial e a IDE indicada pela Google para o desenvolvimento de aplicações Android é o Android Studio, IDE oficial de desenvolvimento da plataforma. (ANDROID, 2016).

2.5.1.2.iOS

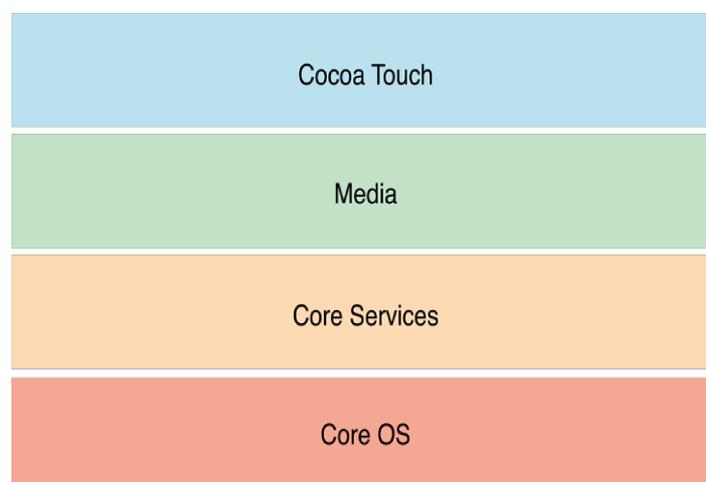
Com o lançamento do iPhone, em 2007, o sistema operacional embarcado no dispositivo tinha o nome de iPhone OS. Após a evolução dos dispositivos e a chegada do iPad, o sistema mudou o nome para iOS que tem como base o sistema operacional antecessor, o OS X. O sistema operacional funciona como uma interface entre as aplicações desenvolvidas e o hardware dos dispositivos. (APPLE INC., 2007).

A maioria das interfaces disponíveis para o uso são disponibilizadas por meio de frameworks, que podem ser adicionados ao projeto no Xcode, IDE oficial, contendo DSLs e os recursos necessários tais como, imagens, aplicativos auxiliares e arquivos header, para o framework funcionar de forma correta. (APPLE INC., 2014).

Arquitetura iOS

A arquitetura iOS é baseada nas camadas de alto nível listadas, conforme a Figura 7 abaixo.

Figura 7 – Arquitetura iOS.



Fonte: APPLE INC., 2014.

- Cocoa Touch: Camada mais alto nível na qual são fornecidos serviços básicos de interação com o usuário como por exemplo entradas baseada em toques, notificações

Push entre outras tecnologias que aprimoram a experiência do usuário. (APPLE INC., 2014).

- **Media:** Contém tecnologias e frameworks necessários para a implementação de funcionalidades multimídia com interação com áudio, vídeo e gráficos. (APPLE INC., 2014).
- **Core Services:** Camada mais próxima do hardware contendo acesso a funcionalidades de mais baixo nível como localização, SQLite e threads. (APPLE INC., 2014).
- **Core OS:** Camada na qual as funcionalidades de mais baixo nível são construídas e utilizadas por outros frameworks em outras camadas. (APPLE INC., 2014).

Desenvolvimento iOS

Diferentemente do desenvolvimento para Android, para a utilização de recursos avançados do SDK iOS, e para a distribuição na App Store, é exigida uma licença do Apple Developer Program, que possui o custo de US\$99,00 por ano. (APPLE INC., 2016).

Para instalar o SDK iOS, é necessário um computador com sistema Operacional OS X. Atualmente o SDK iOS é acompanhado com a ferramenta que auxilia a desenvolver o código, a IDE, denominada de Xcode. (APPLE INC., 2016).

O desenvolvimento é auxiliado por duas linguagens de programação, o Objective-C e o Swift. O Objective-C é baseado na linguagem de programação C, da qual herda a maior parte da sua sintaxe, além de tipos primitivos e fluxo de declarações, e adiciona sintaxe própria para definição de classes e métodos. O Swift foi lançado pela Apple como um sucessor do Objective-C e é uma linguagem software livre e pode ser integrada a códigos Objective-C. (APPLE INC., 2016).

2.5.2.Desenvolvimento Cross-Platform

Como descrito anteriormente, para o desenvolvimento de uma aplicação nativa, é necessário que se codifique para cada plataforma desejada, o que exige tempo, investimento financeiro, e uma equipe completa com conhecimento específico das diversas tecnologias usadas pelas plataformas alvo. O desenvolvimento cross-platform ou multiplataforma, possibilita a implementação de uma aplicação que pode ser executada em diferentes

plataformas. (EL-KASSAS et al., 2015). Segundo Heitkotter, Hanschke e Majchrzak (2013), as soluções cross-platform são comumente categorizadas em web apps e aplicações híbridas.

Aplicações denominadas de web apps são implementadas utilizando tecnologias web tais como as descritas na seção de desenvolvimento web deste trabalho, considerando aspectos específicos dos dispositivos móveis, tais como tamanho de tela e o modo de utilização. (HEITKOTTER; HANSCHKE; MAJCHRZAK, 2013). Segundo Lengstorf e Leggetter (2013), web apps estão disponíveis para qualquer dispositivo que possui um browser, ou navegador web, que suporta as tecnologias utilizadas pela aplicação. A maioria dos dispositivos de hoje em dia possuem navegadores web que suportam HTML5, CSS3 e JavaScript.

Conforme Heitkotter, Hanschke e Majchrzak (2013), a solução híbrida possui a combinação da utilização de tecnologias web com a possibilidade de acesso às funcionalidades nativas.

As seções que seguem apresentam as tecnologias atuais utilizadas para o desenvolvimento multiplataforma.

AngularJS

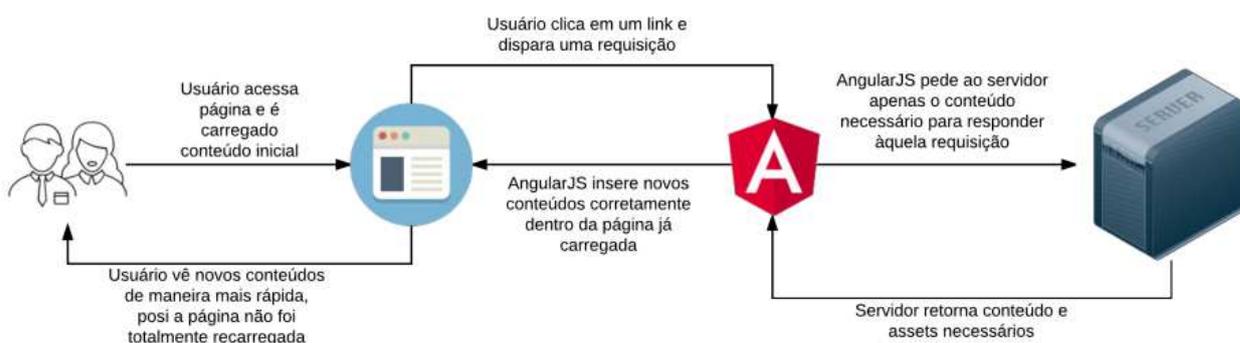
A necessidade de desenvolver aplicações JavaScript de grande porte tem dado origem ao nascimento de diversos frameworks JavaScript que se propõem ser opções para a codificação de aplicações web de grande porte tendo como características, fácil manutenção, escalável e ambiente preparado e propício para testes. Dentre os frameworks que surgiram, o AngularJS se caracteriza como um dos mais populares, mantido pela Google e open source. (RUEBBELKE; FORD, 2015).

Conforme Chandermani (2015), a proposta de manter o código mais fácil de manutenção, legível e organizado se deve ao padrão MVC que o AngularJS utiliza como base, padrão o qual, como explicado na seção de Desenvolvimento Web, se propõe realizar a separação de responsabilidades segundo o papel de cada componente do padrão MVC:

- Model: Armazena os dados da aplicação. (CHANDERMANI, 2015).
- View: Representa o modelo na interface do usuário. (CHANDERMANI, 2015).
- Controller: Responsável pela coordenação entre os componentes model e view. (CHANDERMANI, 2015).

Qualquer mudança no componente Model é visível no componente View e qualquer alteração realizada pelo usuário enquanto interagindo com o componente View é refletido no componente Model. No meio termo existe o Controller que tem o papel de coordenar as alterações e é responsável por manter Model e View sincronizadas. (CHANDERMANI, 2015). Esse ciclo é esquematizado na Figura 8.

Figura 8 – Ciclo de atualização com AngularJS.



Fonte: MATOS; SILVA, 2016 apud URSINO; CAPANNA, 2015, p. 30.

Ionic

Criado em 2013 pela empresa Drifty Co., Ionic é um framework open source para desenvolvimento de aplicativos híbridos utilizando as tecnologias web descritas anteriormente, tais como, HTML, CSS e JavaScript, otimizadas para a plataforma alvo dos dispositivos móveis. (DRIFTY CO., 2016).

Segundo Drafty Co. (2016), uma aplicação Ionic possui as seguintes subdivisões conforme a arquitetura base do framework:

- Views: Camada na qual as informações são mostradas para o usuário, normalmente, arquivos HTML separados, que são atualizados utilizando o AngularJS. (DRIFTY CO., 2016).
- Controllers: Responsável por controlar o fluxo de dados e a lógica da aplicação, apresentando ao usuário as Views e se comunicando com as camadas de dados da aplicação. Possui todas as informações que são necessárias para a criação da View. (DRIFTY CO., 2016).
- Data: Camada mais próxima da camada model do padrão MVC que encapsula dados da aplicação e os fornece por meio de um web service. Essa camada se comunica

com os Controllers respondendo as requisições com os dados a serem utilizados para a criação da View. (DRIFTY CO., 2016).

- App Configuration: Se comunica com os Controllers por meio de rotas, sendo possível criar rotas padrão, caso não haja nenhuma rota que esteja sendo identificada. (DRIFTY CO., 2016).
- Directives: Camada responsável pela interface e criação de comportamentos específicos em elementos de uma página HTML. (DRIFTY CO., 2016).

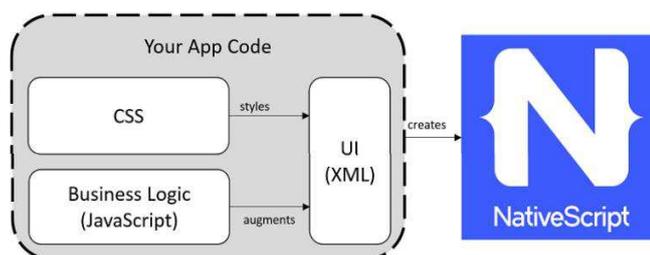
NativeScript

Uma das desvantagens do desenvolvimento multiplataforma é a redução de performance das aplicações que cria de certa forma resultados inesperados e dificulta a manutenção do código base tendo em vista que para melhorar o desempenho, diferentes implementações são feitas. O framework também open source e tecnologia mais recente, conhecido como NativeScript, se propõe a resolver esses problemas recorrentes no desenvolvimento cross-platform, prometendo ser um framework de abordagem verdadeiramente write-once, isto é, a codificação ocorre apenas uma vez e é distribuído para várias plataformas. (BRANSTEIN M., BRANSTEIN N., 2016).

A diferença que o framework propõe é o acesso a elementos nativos através da plataforma desenvolvida em JavaScript. Criado e mantido pela empresa Telerik, NativeScript permite o acesso a todas as bibliotecas do sistema operacional através do código em JavaScript da aplicação. (ANDERSON, 2016).

As aplicações desenvolvidas em NativeScript são codificadas com a combinação das tecnologias JavaScript, XML, e CSS. (BRANSTEIN M., BRANSTEIN N., 2016). A Figura 9 ilustra os componentes de uma aplicação desenvolvida com o framework.

Figura 9 – Tecnologias utilizadas pelo framework NativeScript.



Fonte: BRANSTEIN M., BRANSTEIN N., 2016.

Como pode ser visto na Figura 9, a arquitetura de uma aplicação NativeScript possui os seguintes componentes:

- JavaScript: Componente em que a lógica de negócio é executada, permitindo acesso aos dados e controle de fluxo da aplicação. (BRANSTEIN M., BRANSTEIN N., 2016).
- XML: Utilizado para criar os templates e a interface dentro da aplicação para interação com o usuário. (BRANSTEIN M., BRANSTEIN N., 2016).
- CSS: Responsável pela estilização da interface codificada em XML, que possui semelhanças com a estilização de aplicações HTML. (BRANSTEIN M., BRANSTEIN N., 2016).

2.5.3.Comparativo entre Desenvolvimento Nativo e Cross-platform

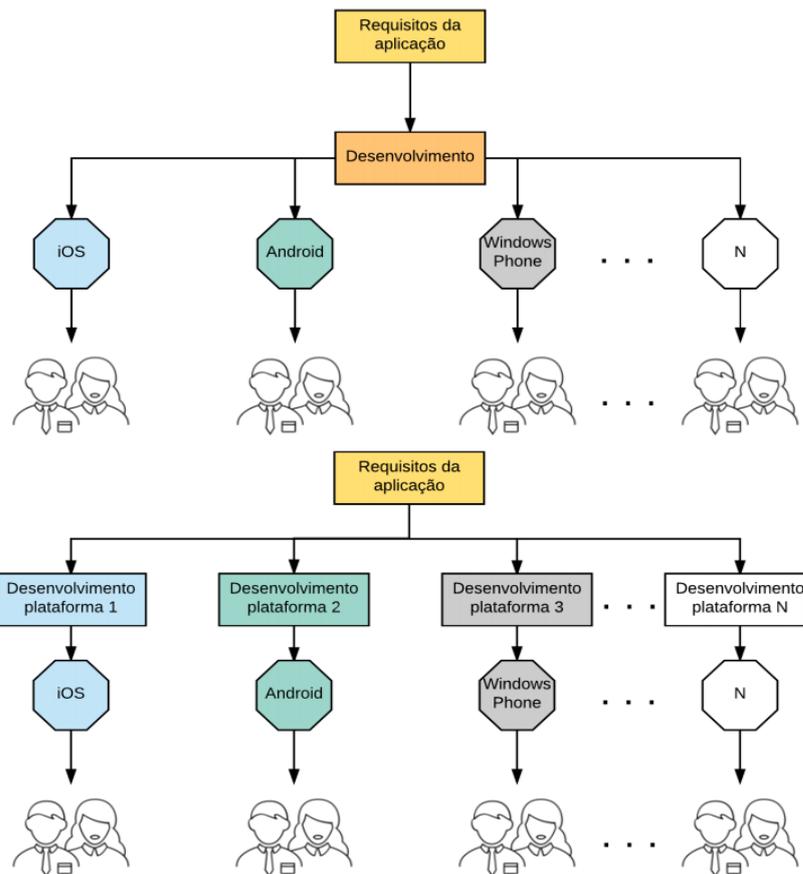
Após a análise do problema que a solução se propões a resolver, a equipe de desenvolvimento tem que decidir qual será a plataforma alvo, sempre considerando que ao excluir uma plataforma, uma grande parcela do mercado pode estar sendo restringida, ocasionando a redução do alcance da aplicação e conseqüentemente os lucros da aplicação. (CORRAL; JANES; REMENCIUS, 2012).

Ao escolher a abordagem nativa, a equipe se depara com o desenvolvimento de várias aplicações para cada plataforma alvo, precisando passar por todo o processo de desenvolvimento visto que cada uma possui sua arquitetura, look and feel, padrões de código e APIs específicas. (HOLZINGER; TREITLER; SLANY, 2012).

Ao utilizar a abordagem cross-platform ou multiplataforma, a aplicação não fica tão dependente de plataforma, considerando que a arquitetura será única assim como o ambiente de desenvolvimento e conjunto de ferramentas. Após o desenvolvimento da aplicação utilizando o desenvolvimento multiplataforma, algumas customizações podem ser feitas para aprimorar a interface referente a cada plataforma alvo. (CORRAL; JANES; REMENCIUS, 2012).

A Figura 10 ilustra as diferenças entre o processo de desenvolvimento nativo e cross-platform.

Figura 10 – Desenvolvimento Nativo x Cross-platform.



Fonte: MATOS; SILVA, 2016 apud CORRAL; JANES; REMENCIUS, 2012, p.36.

A Tabela 5 ilustra algumas vantagens e desvantagens de cada abordagem.

Tabela 5 – Vantagens e desvantagens do desenvolvimento nativo e multiplataforma.

	Vantagens	Desvantagens
Nativo	<ul style="list-style-type: none"> • Explora todas as capacidades do dispositivos • Maior performance • Oferece experiência nativa de usabilidade para o usuário • Integração próxima com o sistema e outros aplicativos 	<ul style="list-style-type: none"> • Necessário um aplicativo por plataforma • Mais caro e mais difícil de manter, por ter que manter vários <i>apps</i> diferentes • Requer domínio de vários ambientes e linguagens para cada plataforma
Multiplataformas	<ul style="list-style-type: none"> • Só é preciso dominar uma linguagem e um ambiente • Desenvolve apenas um código e pode distribuir em várias lojas de <i>apps</i>, o que faz com que o alcance do <i>app</i> seja maior • Reduz custo, tempo e esforço de desenvolvimento e manutenção 	<ul style="list-style-type: none"> • Não possui acesso a todas as funcionalidades do dispositivo • Menor performance comparada ao nativo, linguagens interpretadas e compiladas • Não possui a mesma usabilidade e experiência de uso das aplicações nativas • Não possui as últimas atualizações lançadas pelo <i>SO</i>, pois para cada atualização do <i>SO</i>, é preciso ser feita uma atualização na plataforma <i>cross</i>

Fonte: MATOS; SILVA, 2016.

3. DESENVOLVIMENTO

Neste capítulo, apresenta-se o processo metodológico elaborado para concretizar os objetivos definidos no início deste trabalho. Para isto, é definido um processo de desenvolvimento para a solução aqui proposta.

No decorrer deste capítulo são expostas as etapas do desenvolvimento até se chegar ao produto final deste projeto.

3.1. Formulação do problema

A saúde da população não depende apenas do serviço de saúde pública e da distribuição de medicamentos, mas estes são extremamente importantes para obtenção de resultados que proporcionem a minimização, controle e redução de efeitos causados por doenças. A disponibilidade de medicamentos em setores públicos é determinante para a resolubilidade da atenção e dos serviços em saúde envolvendo muitos recursos públicos e, atualmente, é um dos setores de maior impacto financeiro, e a tendência da demanda de medicamentos é cada vez maior. A ausência de um gerenciamento eficiente pode acarretar vários problemas e grandes desperdícios. (BRASIL, 2004).

A aquisição de medicamentos no setor público, é uma das peças que contribuem para a credibilidade dos serviços farmacêuticos. A organização do processo de aquisição e distribuição possui diversas partes que compõem o sistema, e a ausência de uma delas, ou execução inadequada, impede o correto funcionamento de todo o ciclo, podendo assim comprometer o objetivo do programa, que é selecionar, programar, adquirir, armazenar, distribuir e dispensar medicamentos com qualidade e segurança à população. (MARIN et al., 2003).

Conforme Brito e Bastos (2015), a evolução dos gastos com medicamentos no Brasil tem aumentado em um índice superior à inflação e ao crescimento do Produto Interno Bruto – PIB, um dado alarmante tendo em vista que, se essa tendência continuar e o nível do cuidado de saúde for mantido, os governos terão que adotar algumas medidas para sustentar o financiamento de medicamentos tais como, aumentar impostos e cortar os gastos em outras áreas. Porém, pode-se perceber, que caminhos alternativos podem ser repensados, uma vez que adquiridos os medicamentos, os processos de armazenamento e gestão, devem ser sempre

reavaliados a fim de que o controle entre a aquisição e a dispensação ocorram da forma mais viável possível.

Os governos federais, estaduais e municipais desperdiçam por ano, cerca de um bilhão de medicamentos. Em estudos realizados pelo Conselho Federal de Farmácia (CRF), em média, 20% dos medicamentos comprados no varejo pelo poder público e pelos hospitais privados são desperdiçados anualmente por falhas no processo de seleção, aquisição e no armazenamento dos medicamentos e ainda mais, por problemas na gestão realizada por profissionais não capacitados, ou até mesmo a falta deste nas unidades básicas de saúde. (BENEVIDES; GALDO, 2010).

Pode-se citar como fatores relevantes nesse processo de desperdício, a ineficiência da gestão farmacêutica e o uso irracional de medicamentos. O problema levou o Conselho Federal de Farmácia (CFF) a encaminhar uma proposta ao Ministério da Saúde para que o governo realize a implantação de farmácias públicas em todos os municípios ou postos de medicamentos emergenciais para cidades com menos de 15 mil habitantes. Nesses municípios, os farmacêuticos fariam a gerência dos medicamentos de como selecionar adequadamente e evitar descarte desnecessário, evitando com que os medicamentos expirem os prazos de validade ou estraguem nas prateleiras por não terem gestão e acompanhamento sobre os mesmos. (BENEVIDES; GALDO, 2010).

Para se ter uma visão do investimento que os medicamentos representam para os cofres públicos, enquanto os gastos totais com saúde aumentaram em 9,6%, os gastos com medicamentos tiveram incremento de 123,9% no período de 2002 a 2006. Esse dado mostra como o gerenciamento desses medicamentos precisa ser monitorado para identificar fatores que podem ter contribuído para essa discrepância. Como pode-se concluir, esse percentual vem crescendo cada vez mais em diversos Estados e Municípios brasileiros, que ainda não possuem estruturas, e profissionais preparados de forma satisfatória tanto para controlar bem esses gastos, como atenderem a população nas suas necessidades. (VIEIRA; MENDES, 2007).

No estado do Maranhão, foi realizada pesquisas de avaliação para o diagnóstico de implementação do programa de Assistência Farmacêutica e foi identificado problemas com aquisição de medicamentos constatando que 27% das farmácias dos municípios do estado geram um grande desperdício de medicamentos, estando os mesmos com prazo de validade vencido, demonstrando a inobservância da falta do profissional farmacêutico para atuar frente

a programas similares, ocasionando falhas no processo e aumento de gasto do setor público, devido ao não comprometimento e cumprimento de normas já estabelecidas. (JANNUZZI, 2011).

3.2. Proposta de Solução

Atualmente, a distribuição de medicamentos por parte do governo para os postos de saúde possui várias deficiências, dentre elas, podemos citar que os medicamentos são distribuídos sem demanda específica para o tipo de medicamento, e sim pela quantidade de atendimentos que aquele posto recebe, e até em certos casos, dependendo da região, os postos de saúde recebem medicamentos de forma igualitária.

Porem desta forma, é comum certos postos que possuem maior demanda de atendimentos por atenderem uma maior região, acabam esgotando seus medicamentos em um curto período de tempo. Enquanto isso, em postos de saúde com menor demanda de atendimento, sobram medicamentos, sem nenhuma possibilidade de serem usados em outros postos de saúdes, ocasionando os gastos descritos anteriormente.

O processo de redistribuição desses medicamentos é inexistente. Nenhum posto, apesar de ter um estoque considerável, consegue ou pode suprir a necessidade de outros postos de saúde. Como resultado, remédios e compostos químicos que passam da validade, não podem ser administrados em pacientes e exames, e são desperdiçados.

A falta de interação entre os postos de saúde e a redistribuição dos medicamentos é o fator chave desse problema, e foco deste projeto.

A proposta envolve duas partes: 1) o desenvolvimento do sistema web e aplicativo móvel colaborativo e gamificado; 2) a solução para a distribuição de cargas de medicamentos utilizando drones.

Este trabalho irá contemplar a primeira parte da proposta, que é a apresentação de uma solução de software que possibilita o compartilhamento de medicamentos entre postos de saúde, auxiliando na diminuição de gastos e redução de desperdícios que ocorrem no ato de dispensa de medicamentos. Como partes integrantes da solução, será apresentado o protótipo de um

sistema web e um aplicativo. A segunda parte foi desenvolvida em outro trabalho de conclusão de curso de Engenharia de Computação da UEMA¹.

Tomando-se como exemplo, a funcionalidade e necessidade de se registrar o recebimento de um medicamento proveniente de outro posto e também a sua saída requer impressão de relatórios, apresentação de análises entre os gestores de cada local e, com isso, o sistema web se mostrou mais eficaz visto que pode ser manuseado em um desktop/notebook com fácil acesso a impressoras e conexão a projetores.

Por outro lado, se sabe que o número de computadores em postos de saúde, quando existem, é restrito a setores, isto é, cada setor possui apenas um computador desktop e se torna inviável o acesso ao mesmo por cada funcionário para solicitar ou atender medicamentos. Essa estrutura foi o ponto principal para que a proposta de um aplicativo também estivesse incluída na solução, não só pela mobilidade, mas pela agilidade que o usuário terá para rapidamente solicitar um medicamento e visualizar os atendimentos de outros postos.

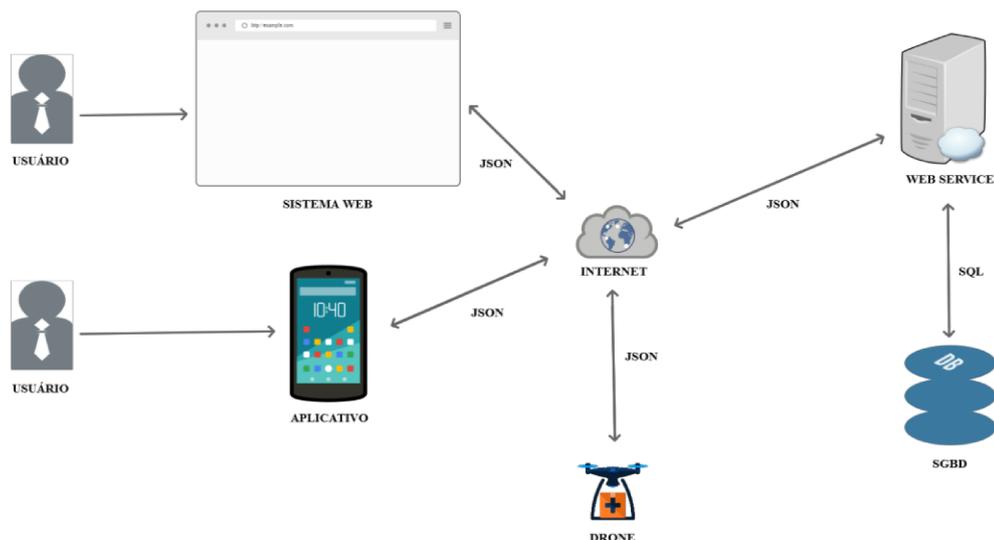
As seções que seguem, apresentam os itens do projeto essenciais que tratam do desenvolvimento da solução de software, tais como o levantamento de requisitos, modelagem do banco de dados, definição dos protótipos de interface, arquitetura e o processo de desenvolvimento como resultado do trabalho. Paralelo aos itens essenciais, será apresentado também um projeto de gamificação objetivando motivar a utilização da solução pelos usuários que terão interação com o sistema.

3.3. Arquitetura Geral

A arquitetura projetada para a solução deste projeto consiste na comunicação com um Web Service desenvolvido e hospedado no servidor remoto que fará a ponte com o banco de dados também hospedado no mesmo servidor. Na arquitetura proposta, o Web Service também se comunica com o drone, fornecendo dados de origem, destino e o caminho a percorrer para que seja entregue as cargas de medicamentos atendidas. Tanto o sistema web quanto o aplicativo se comunicarão com o webservice, como ilustrado na Figura 11.

¹ GUEDES, SAULO. **SOLUÇÃO COLABORATIVA E GAMIFICADA PARA DISTRIBUIÇÃO DE CARGAS DE MEDICAMENTOS: ESTUDO DE CASO UTILIZANDO DRONES**. 2016. No prelo.

Figura 11 – Arquitetura geral da solução.



Fonte: AUTOR, 2016.

Em linhas gerais, o Web Service da solução apresentada tem como objetivo expor as funcionalidades para os usuários do sistema web e do aplicativo através do protocolo HTTP. Cada funcionalidade representa uma aplicação identificada por uma URI, cujas interfaces podem ser definidas através de artefatos JSON, estrutura de dados escolhida para troca de mensagens entre servidor e cliente.

A estrutura de dados JSON é um formato de troca de dados definido como um conjunto de nomes e valores separados por dois pontos e delimitados por vírgulas. Algumas das vantagens da utilização do JSON são:

- Suporte a objetos
- Leitura mais simples
- Velocidade maior na execução e troca de dados devido a sua estrutura
- Arquivo com tamanho compacto

Além das suas vantagens, o JSON é uma ferramenta bastante utilizada para auxiliar a execução de chamadas AJAX nas quais o retorno obtido da execução no servidor obedece aos padrões de criação de objetos JSON, tornando-se mais fácil a leitura dos dados retornados em JavaScript.

3.4. Requisitos

No início de toda a atividade de desenvolvimento de software é necessário levantar os requisitos do sistema proposto. Sommerville (2007) apresenta como proposta, um processo genérico de levantamento de requisitos que contém as seguintes atividades: compreensão do domínio e coleta dos requisitos. A compreensão do domínio foi realizada na formulação do problema. Quanto ao levantamento de requisitos, foi levantado os seguintes requisitos funcionais que indicam o que o sistema deve fazer, de forma abstrata:

- O sistema deverá conter controle de acesso para os usuários que representam os funcionários de um posto de saúde, que deverão solicitar medicamentos e efetuar atendimentos a solicitações e também os que representam os administradores de cada posto, que o papel é autorizar tanto as solicitações quanto os atendimentos;
- O sistema deve exigir credenciais de acesso para o controle de acesso;
- O sistema deverá solicitar a escolha do posto caso o usuário tenha relação com mais de um;
- O sistema deve ser capaz de fornecer uma lista de medicamentos com nomenclatura familiar referente ao domínio do problema, que será utilizada para a solicitação de medicamentos;
- O sistema deve fornecer a lista de solicitações sendo atendidas caso a mesma não tenha sido finalizada;
- O sistema também deverá ser capaz de consultar o histórico de todas as solicitações;
- O sistema deverá restringir a escolha de apenas um medicamento no ato da solicitação, sendo necessário a informação de quantidade de unidades do medicamento solicitado;
- Ao ser clicado um item da lista de solicitações pendentes, o sistema deverá mostrar informações a respeito do status da solicitação em questão;
- O sistema deve fornecer a lista de solicitações de outros postos que ainda estão pendentes e podem ser atendidas;
- O sistema possibilitará a escolha de um item da lista de solicitações de outros postos, sendo necessário a informação da quantidade de unidades do medicamento que será atendido;
- O sistema também deverá ser capaz de consultar o histórico de todos os atendimentos efetuados;

- Para cada solicitação realizada pelos usuários que representam os funcionários do posto, deverá ser autorizada pelo usuário que representa o Administrador do mesmo posto;
- Para cada atendimento de solicitação de outros postos realizado pelo usuário que representam os funcionários do posto, deverá ser autorizada pelo usuário que representa o Administrador do mesmo posto;
- O sistema deve possibilitar a geração de termos de recebimento e de saída decorrentes do atendimento de uma solicitação apenas se confirmado a autorização de algum o Administrador do posto que está atendendo;
- O sistema deve possibilitar a consulta de relatórios e/ou gráficos que ilustrem a comparação do número de atendimentos e solicitações de um posto;
- Sistema deve enviar os dados do atendimento para o sistema externo de entrega que será encarregado de encaminhar a quantidade doada do medicamento ao posto solicitante;
- Todos as solicitações e atendimentos deverão ser visíveis em todos os postos em tempo real através de notificações e/ou atualização das listas.

Como constatado através da compreensão do domínio, a complexidade e o tamanho do problema é grande, existindo diversos fatores presentes. Visto que a solução do trabalho se propõe a resolver um dos fatores do problema, isto é, a redução de desperdícios de medicamentos através do compartilhamento dos mesmos, se faz necessário o levantamento de requisitos não funcionais que definem o que o software não se propõe a solucionar e as suas restrições:

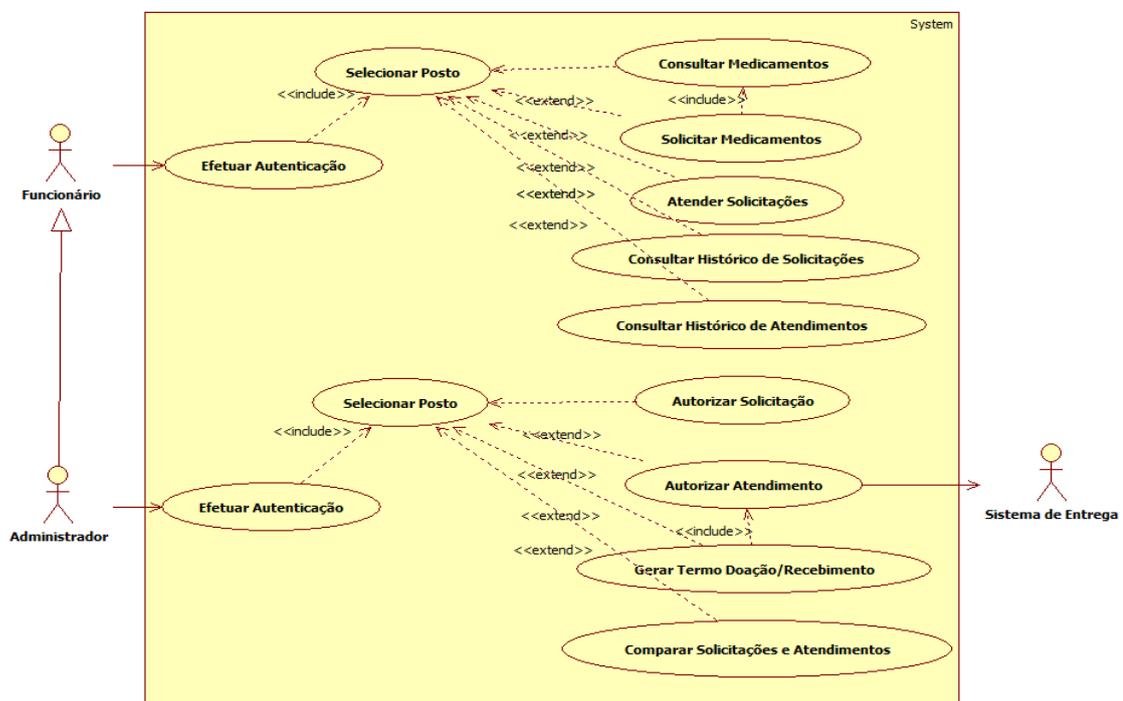
- O sistema não armazenará qualquer informação de estoque de medicamentos assim como não haverá cadastro dos mesmos pelo usuário;
- As informações sobre os medicamentos deverão conter apenas dados básicos e relevantes para a realização de solicitações, dados estes conhecidos e familiares a todos os profissionais que se relacionam com o domínio;
- O aplicativo conterà todas as funcionalidades do sistema web com exceção das funcionalidades de gerar termo e consulta de relatórios e gráficos.

Para poder representar inicialmente as funcionalidades propostas, foi utilizado um diagrama que facilita o entendimento das fronteiras do sistema, conhecido como diagrama de caso de uso.

Segundo o Rational Unified Process (RUP, 2013), o Diagrama de Caso de Uso representa um cenário simplificado das funcionalidades do sistema. Ele é composto de casos de uso e atores. Os atores representam uma entidade externa ao sistema, podendo ser pessoas ou outros sistemas, e os casos de uso representam as funções ou funcionalidades disponibilizadas pelo sistema.

A Figura 12 ilustra o diagrama de caso de uso do sistema modelado de acordo com os requisitos levantados anteriormente.

Figura 12 – Diagrama de caso de uso do sistema web.



Fonte: AUTOR, 2016.

Como pode ser observado, o diagrama é manipulado pelos seguintes atores:

- **Funcionário:** Ator que representa os usuários que trabalham no posto de saúde e poderão solicitar medicamentos e efetuar atendimentos;

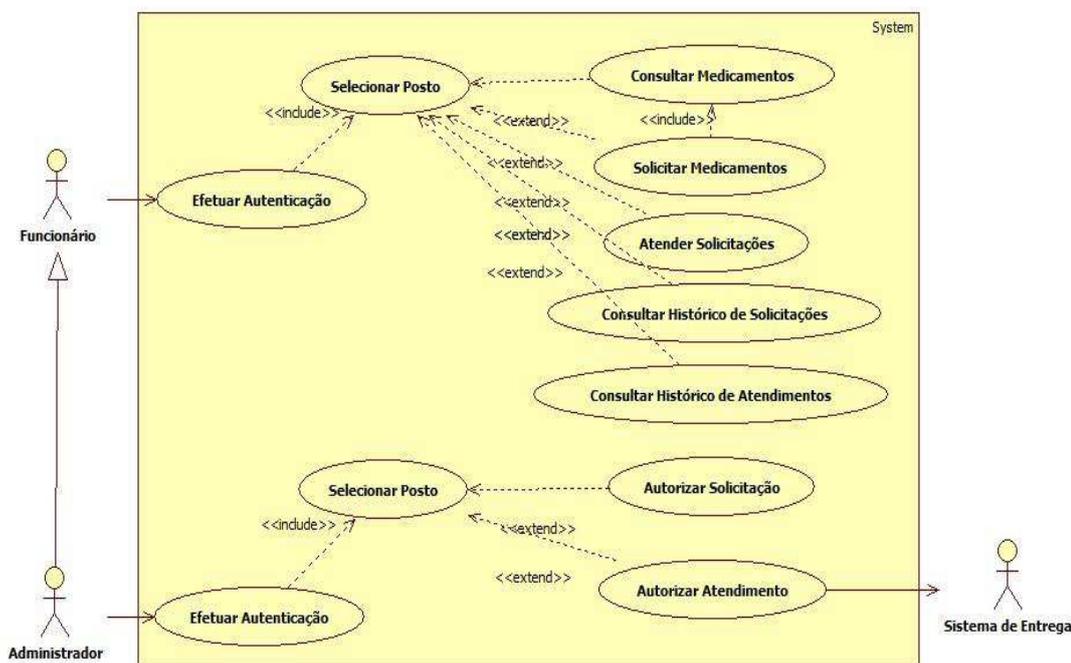
- Administrador: Uma especialização do Funcionário, o ator Administrador representa os usuários do posto de saúde que possuem permissão para autorizar as solicitações e os atendimentos;
- Sistema de Entrega: Agente externo que representa o sistema de envio dos medicamentos de atendimentos autorizados.

Os casos de uso que compõem o diagrama são:

- Efetuar Autenticação: Este caso de uso que representa o acesso ao sistema é obrigatório tanto para o ator Funcionário quanto para o ator Administrador;
- Selecionar Posto: Assim como a autenticação, a interação com o sistema só ocorrerá se o usuário especificar qual posto ele se encontra e realizará as atividades;
- Consultar Medicamentos: Representa o acesso a funcionalidade de consulta de medicamentos cadastrados na base de medicamentos;
- Solicitar Medicamentos: Representa a interação do usuário para a solicitação de um medicamento;
- Atender Solicitações: Representa o atendimento de solicitações de outros postos de saúde;
- Consultar Histórico de Solicitações: Caso de uso que define a possibilidade de consultar todas as solicitações realizadas;
- Consultar Histórico de Atendimentos: Representa o acesso ao histórico de todos os atendimentos efetuados;
- Autorizar Solicitação: Caso de uso restrito ao ator Administrador, representa a funcionalidade de autorizar as solicitações realizadas pelos outros usuários;
- Autorizar Atendimento: Também restrito ao Administrador, possibilita a autorização dos atendimentos efetuados pelos outros usuários do posto;
- Gerar Termo Doação/Recebimento: Caso de uso que somente ocorrerá se um atendimento tiver sido autorizado, possibilitando a geração do Termo de Atendimento para o posto que está atendendo à solicitação e do Termo de Recebimento para o posto solicitante;
- Comparar Solicitações e Atendimentos: Caso de uso referente às consultas de relatórios e gráficos.

A Figura 13 ilustra os casos de usos referentes ao aplicativo que de forma resumida permitirá solicitar e atender solicitações de medicamentos e também autorizá-las.

Figura 13 – Diagrama de caso de uso do aplicativo.



Fonte: AUTOR, 2016.

3.5. Projeto de Gamificação

O projeto de gamificação vem com o objetivo de despertar o interesse do usuário em utilizar o sistema proposto, utilizando os mecanismos de jogos tais como as técnicas de game design e elementos descritos na revisão de literatura deste trabalho.

Considerando o perfil dos usuários que utilizarão o software, funcionários de postos de saúde que possuem inúmeras atividades e que diariamente atendem um número muito alto de pessoas, a missão do projeto de gamificação é fazer com que esses usuários se sintam estimulados a utilizar o software, isto é, a relatarem a necessidade de um medicamento e atender a necessidade de outros postos de saúde.

Estabelecido então a missão, foi definido as ações que cada usuário realizará dentro do domínio da gamificação e que terá um feedback, positivo ou negativo. A Tabela 6 ilustra as ações referentes a cada tipo de usuário do sistema, denominadas de eventos:

Tabela 6 – Tabela de eventos.

Perfil de Acesso	Eventos
Funcionário	Solicitar medicamentos; Atender a solicitações; Confirmar recebimento
Administrador	Solicitar medicamentos; Atender a solicitações; Confirmar recebimento; Autorizar/Negar solicitação; Autorizar/Negar atendimento

Fonte: AUTOR, 2016.

Para cada ação descrita na Tabela 6, existe um feedback que neste projeto engloba recompensas e penalizações como seguem:

- **Solicitar medicamento:** 5 pontos positivos;
- **Medicamento solicitado não autorizado:** 5 pontos negativos;
- **Medicamento solicitado autorizado:** 5 pontos positivos;
- **Atender à solicitação:** 10 pontos positivos;
- **Medicamento atendido não autorizado:** 10 pontos negativos;
- **Medicamento atendido autorizado:** 10 pontos positivos;
- **Autorizar/Negar solicitação:** 5 pontos positivos;
- **Autorizar/Negar atendimento:** 10 pontos positivos;

Ao decorrer do tempo, os usuários terão realizados um conjunto de ações, acumulando dessa forma, pontos que poderão ser analisados em relatórios e gráficos acessíveis pelo sistema web. De acordo com o acúmulo de pontos relacionados a cada evento, cada usuário receberá badges, isto é, títulos que indicam a sua maior atividade dentro do sistema:

- **Doador:** Usuário que acumulou mais 500 pontos atendendo a solicitações de outros postos;
- **Suplicante:** Usuário que acumulou mais de 500 pontos realizando solicitações;
- **Mediador:** Usuário que acumulou mais que 500 pontos autorizando e/ou negando tanto solicitações quanto atendimentos;

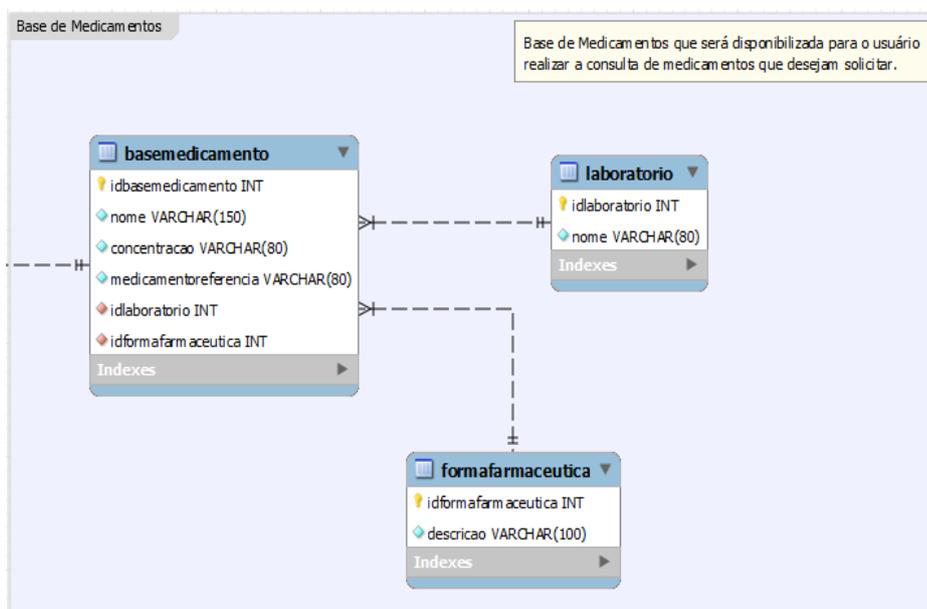
3.6. Modelagem do Banco de Dados

Após análise dos requisitos do sistema, modelagem de casos de usos e definição do projeto de gamificação, o Banco de Dados do sistema foi modelado de acordo com módulos representando as principais funcionalidades e requisitos do sistema e a proposta de gamificação da solução. As subseções que seguem descrevem a modelagem das funcionalidades principais.

3.6.1. Base de Medicamentos

Como descrito no levantamento de requisitos, o sistema não permitirá cadastro de medicamentos e nem terá controle de estoque. Para cumprir tal especificação, será necessário que o sistema já possua uma base de dados com os medicamentos cadastrados. A Figura 14 ilustra as relações oriundas dessa especificação:

Figura 14 – DER da base de medicamentos.



Fonte: AUTOR, 2016.

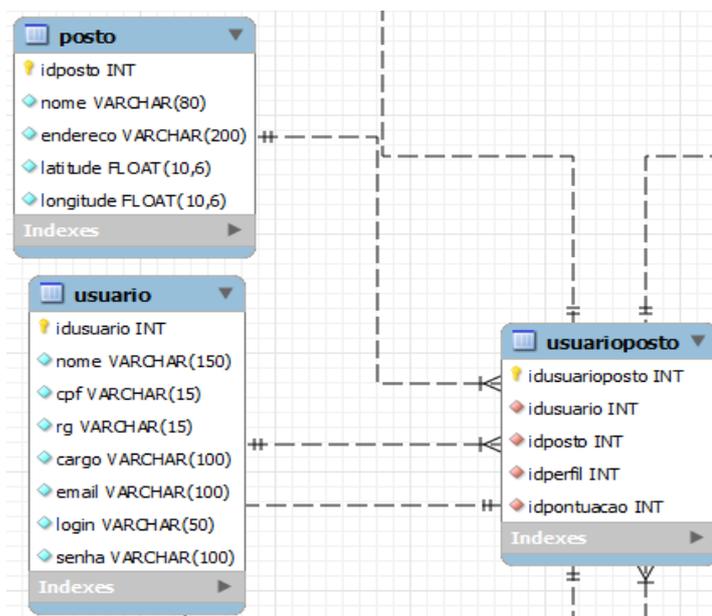
3.6.2. Usuários e Postos de Saúde

Dentre os atributos principais dos postos, tais como nome e endereço, será armazenado também as informações de latitude e longitude, necessárias ao envio de dados ao sistema externo de entrega.

Cada usuário terá também um relacionamento com o posto de saúde, podendo um usuário pertencer a vários postos, uma especificação decorrente da realidade domínio, no qual muita das vezes os funcionários trabalham em diferentes postos de saúde.

A Figura 15 ilustra modelagem do diagrama entidade relacionamento das relações usuário, posto, e o relacionamento usuarioposto.

Figura 15 – DER de usuários e postos de saúde.

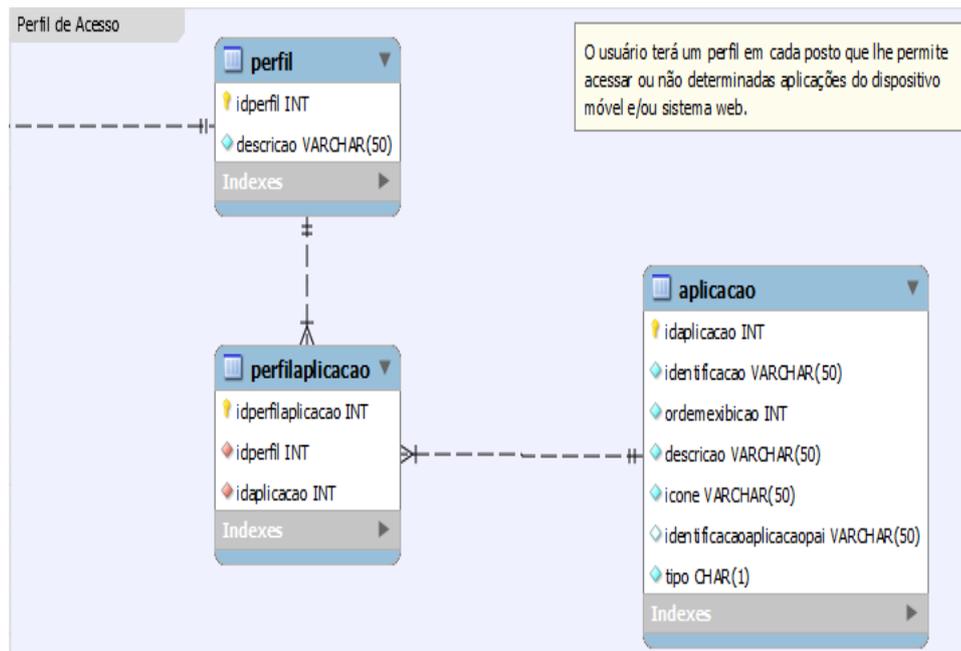


Fonte: AUTOR, 2016.

3.6.3. Perfil de Acesso

Outra especificação a ser cumprida pelo sistema é o controle de acesso no qual foram levantados dois perfis, o do Funcionário e do Administrador. Além disso, é preciso especificar as aplicações acessíveis por cada perfil. A Figura 16 detalha a estrutura que será mantida no banco de dados.

Figura 16 – DER de perfil de acesso.

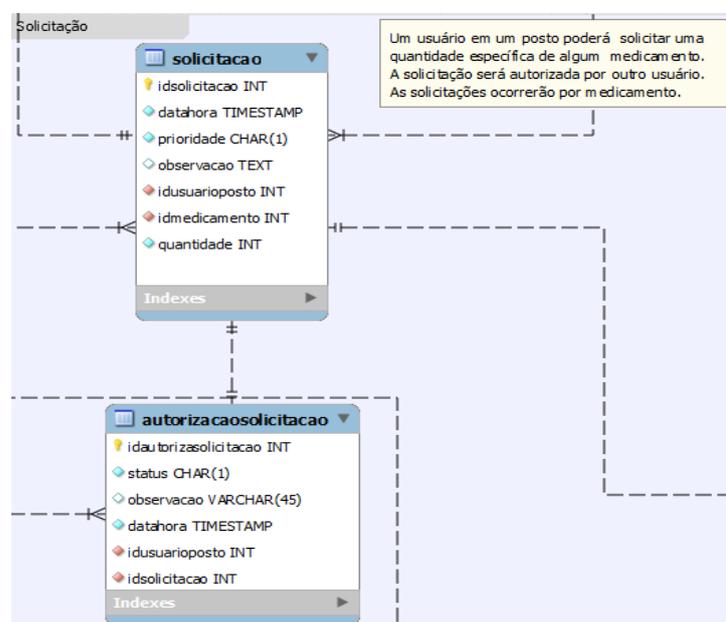


Fonte: AUTOR, 2016.

3.6.4. Solicitação

Todas as solicitações realizadas pelo usuário requerem autorização prévia do Administrador do posto de saúde para que sejam visualizadas por outros postos. A Figura 17 demonstra a modelagem para a solicitação e a autorização da mesma.

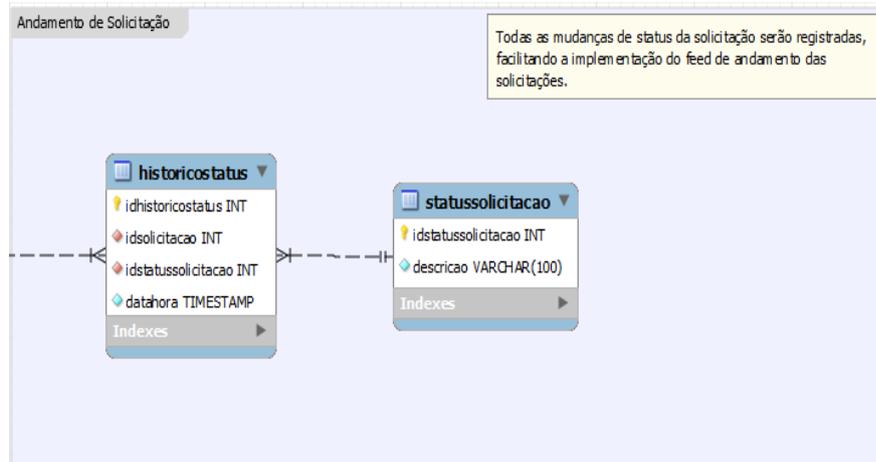
Figura 17 – DER de solicitação e autorização de solicitação.



Fonte: AUTOR, 2016.

Também envolvendo as solicitações, é necessário que andamento de cada uma seja armazenado no banco de dados, para melhor análise e acompanhamento. A Figura 18 detalha a estrutura modelada para o andamento de solicitações.

Figura 18 – DER de andamento de solicitação.

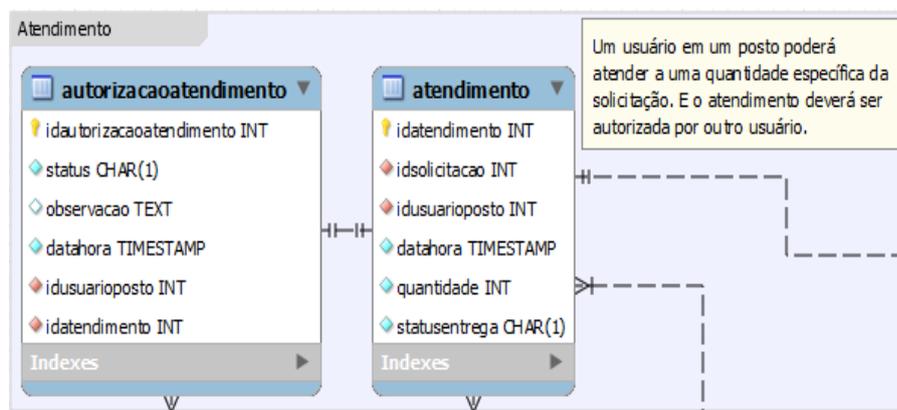


Fonte: AUTOR, 2016.

3.6.5. Atendimento

Da mesma forma que a solicitação deverá ser autorizada previamente, o atendimento a solicitação de outros postos de saúde também deverá ser autorizado, fazendo-se necessário a modelagem especificada na Figura 19.

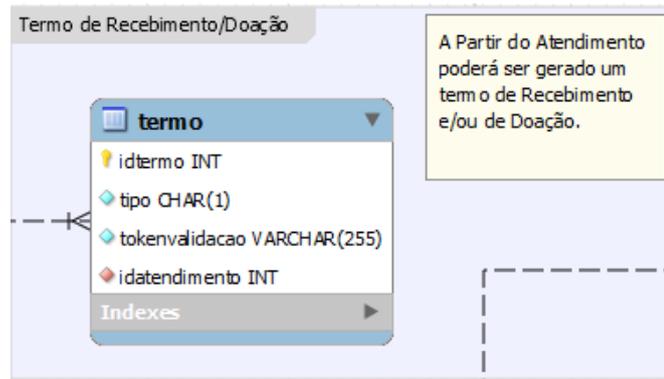
Figura 19 – DER de atendimento e autorização de atendimento.



Fonte: AUTOR, 2016.

O atendimento de uma solicitação e sua autorização permitirá a geração de termos de recebimento e de saída, sendo necessário a modelagem da relação ilustrada pela Figura 20.

Figura 20 – DER de termo de recebimento/doação.

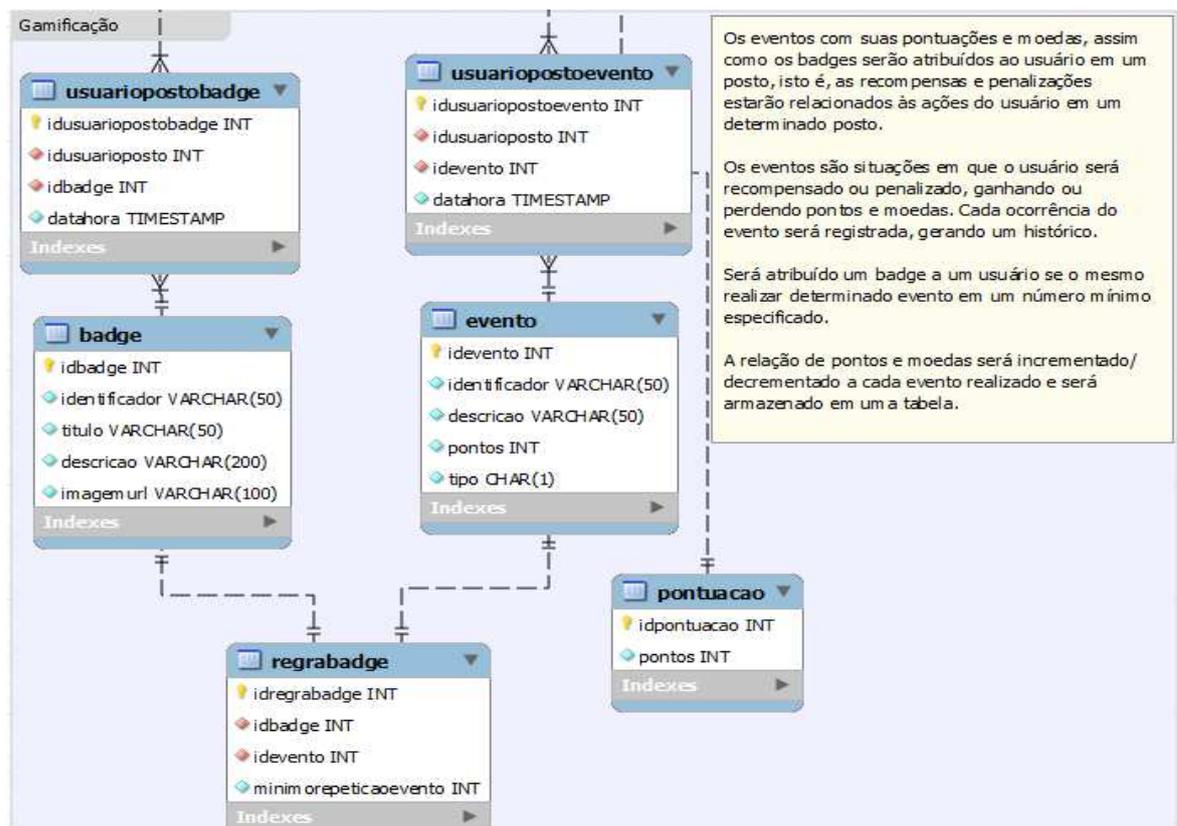


Fonte: AUTOR 2016.

3.6.6. Gamificação

Por último, foi modelado o diagrama entidade relacionamento do projeto de gamificação que fornecerá suporte a uma implementação futura. A Figura 21 detalha a modelagem do projeto de gamificação.

Figura 21 – DER de gamificação.



Fonte: AUTOR, 2016.

3.7. Prototipação

Uma técnica que serve de complemento à análise de requisitos é a construção de protótipos que representam um esboço de partes do sistema, tais como telas de entrada, telas de saída, subsistemas, ou até mesmo o sistema como um todo. Os protótipos podem ser utilizados para validação dos requisitos do sistema para que seja assegurado que os mesmos foram bem entendidos.

As seções que seguem descrevem os protótipos descartáveis construídos para validação do sistema web e aplicativo. Todos os protótipos foram desenvolvidos com auxílio da ferramenta CASE de prototipação denominada Pencil.

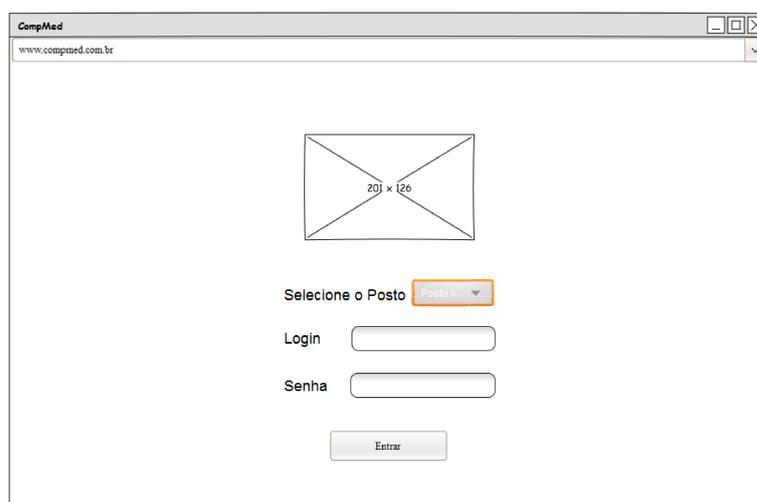
3.7.1. Sistema Web

O sistema web diverge do aplicativo pelas funcionalidades. Enquanto o aplicativo fornece apenas as ações de solicitar, atender solicitações, autorizar solicitações e autorizar atendimentos, o sistema web foi projetado para ser restrito a usuários com perfil de Administrador que poderão emitir termos, consultar relatórios, gráficos, entre outras funcionalidades restritas que poderão ser adicionadas no futuro.

Login

Na aplicação web, terá um campo para seleção do posto de saúde que o funcionário está atendendo no momento. A Figura 22 mostra um esboço da tela de login do sistema web.

Figura 22 – Tela de Login sistema web.



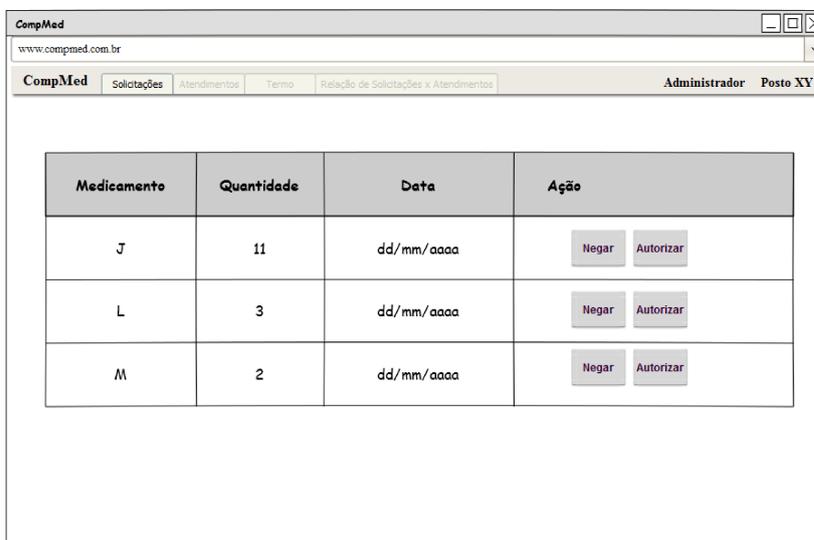
Fonte: AUTOR, 2016.

Tela Principal

Como explanado anteriormente, a aplicação web terá algumas funcionalidades restritas ao Administrador, tal como ilustrada na Figura 22, a relação de solicitações pendentes de autorização, a relação de atendimentos também pendentes de autorização, a relação de solicitações e atendimentos finalizados para geração de termos e gráfico ilustrando os dados gerais.

Cada solicitação realizada pelos outros usuários com perfil de Administrador será registrada, inicialmente, como pendente de autorização, e será papel do Administrador autorizá-la ou negá-la, como demonstrado na Figura 23.

Figura 23 – Tela principal sistema web.



The screenshot shows a web browser window titled 'CompMed' with the URL 'www.compmed.com.br'. The interface includes a navigation menu with 'Solicitações', 'Atendimentos', 'Termo', and 'Relação de Solicitações x Atendimentos'. The user is logged in as 'Administrador' in 'Posto XY'. The main content area displays a table with the following data:

Medicamento	Quantidade	Data	Ação
J	11	dd/mm/aaaa	Negar Autorizar
L	3	dd/mm/aaaa	Negar Autorizar
M	2	dd/mm/aaaa	Negar Autorizar

Fonte: AUTOR, 2016.

Atendimentos

Assim como as solicitações, os atendimentos também devem ser autorizados pelo administrador. A Figura 24 ilustra a tela do sistema web que lista todas as autorizações que ainda não foram autorizadas.

Figura 24 – Tela de atendimentos pendentes sistema web.

Medicamento	Atendido	Data	Ação
J	11	dd/mm/aaaa	Negar Autorizar
L	3	dd/mm/aaaa	Negar Autorizar
M	2	dd/mm/aaaa	Negar Autorizar

Fonte: AUTOR, 2016.

Termo de Recebimento x Doação

A tela da Figura 25 ilustra a relação de solicitações e atendimentos do posto de saúde em que o usuário está logado, todas finalizadas que podem ser gerados termos.

Figura 25 – Tela de solicitações e atendimentos finalizados.

Solicitações Finalizadas

Medicamento	Quantidade	Status	Origem	Ação
D	1	Finalizado	Posto B	Gerar Termo
E	4	Finalizado	Posto C	Gerar Termo
F	11	Finalizado	Posto A	Gerar Termo

Atendimentos Finalizados

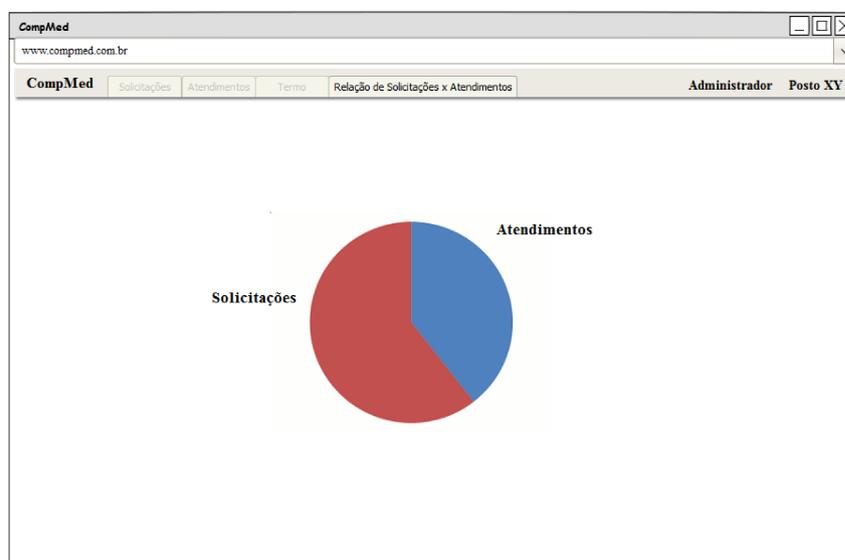
Medicamento	Quantidade	Status	Destino	Ação
K	10	Finalizado	Posto A	Gerar Termo
J	7	Finalizado	Posto B	Gerar Termo
A	9	Finalizado	Posto A	Gerar Termo

Fonte: AUTOR, 2016.

Relação de Solicitações x Atendimentos

Um dos requisitos levantados é a opção de relatórios oriundos de todas as solicitações e atendimentos finalizados, esboçado na prototipação e ilustrado na Figura 26.

Figura 26 – Tela de relação geral de solicitações e atendimentos do posto.



Fonte: AUTOR, 2016.

3.7.2. Aplicativo

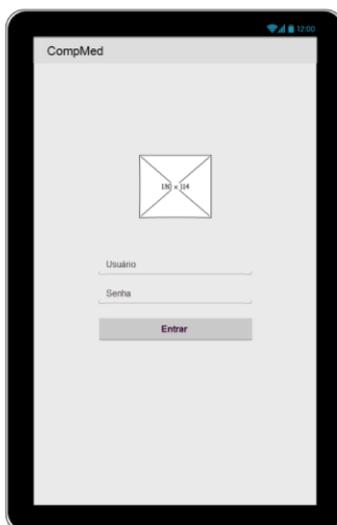
Considerando que o aplicativo atende aos requisitos de solicitar medicamentos, atender solicitações, autorizar solicitações e atendimentos, foi projetado os protótipos de telas descritas abaixo.

Login

Cada usuário deverá ser cadastrado previamente, sendo fornecido suas credenciais de acesso, tais como identificação do login e senha.

Para acesso às funcionalidades do aplicativo, o usuário deverá acessar o sistema efetuando o seu login. A tela de login ilustrada na Figura 27 será uma das telas comuns tanto ao perfil de Administrador quanto ao perfil de Funcionário.

Figura 27 – Tela de login do aplicativo.

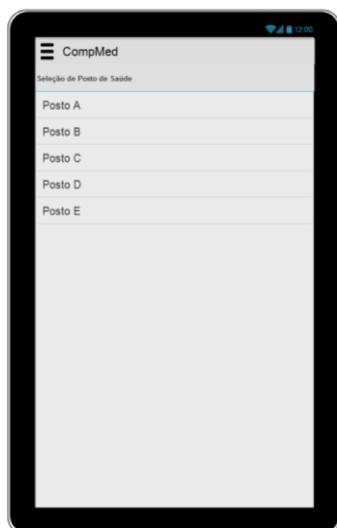


Fonte: AUTOR, 2016.

3.7.2.1. Seleção de Posto

Após o usuário logar no aplicativo, o mesmo deverá escolher um dos postos de saúde a qual faz parte e foi previamente cadastrado. A Figura 28 ilustra a seleção de um dos postos de saúde.

Figura 28 – Tela de seleção de posto de saúde.



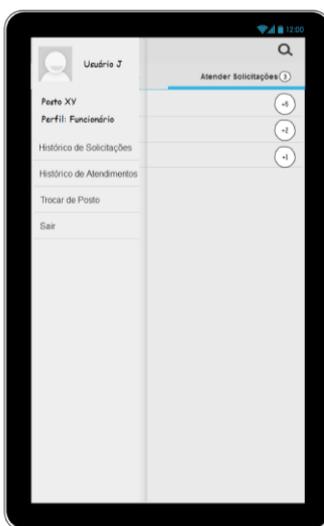
Fonte: AUTOR, 2016.

Menu Vertical acessado pelo Funcionário

Após o posto de saúde ser devidamente selecionado o usuário acessará as aplicações principais assim como as aplicações listadas no menu vertical. A Figura 29 detalha o evento de abertura do menu vertical.

É importante ressaltar que esta tela é uma das telas que serão dinâmicas de acordo com o perfil do usuário, que no caso descrito, representa o perfil de Funcionário.

Figura 29 – Tela de menu vertical acessado pelo funcionário.



Fonte: AUTOR, 2016.

Tela Principal acessado pelo Funcionário

A tela principal acessada pelo Funcionário é ilustrada pela Figura 30. Esta contém uma aba “Minhas Solicitações”, que aparecerá informações de solicitações já realizadas pelo usuário com a opção de adicionar uma nova. Se o usuário não tiver realizado nenhuma solicitação ainda, aparecerá uma mensagem "Ainda não foi realizada nenhuma solicitação."

A outra aba, “Atender Solicitações”, conterà informações de solicitações de outros postos, que o usuário escolherá atender ou não de acordo com a realidade do seu posto de saúde.

Figura 30 – Tela principal acessado pelo funcionário.

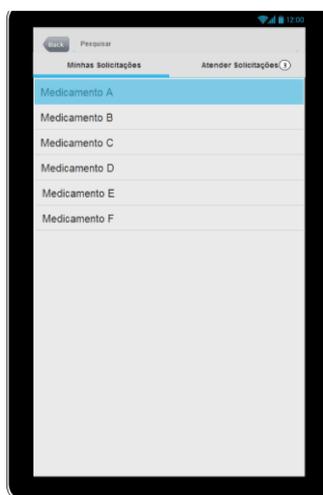


Fonte: AUTOR, 2016.

Solicitação

Ao adicionar uma nova solicitação aparecerá uma lista de medicamentos que estão disponíveis nas farmácias populares e postos de saúde com a possibilidade de serem pesquisados por nome. A Figura 31 simula o usuário efetuando uma nova solicitação e pesquisando o medicamento.

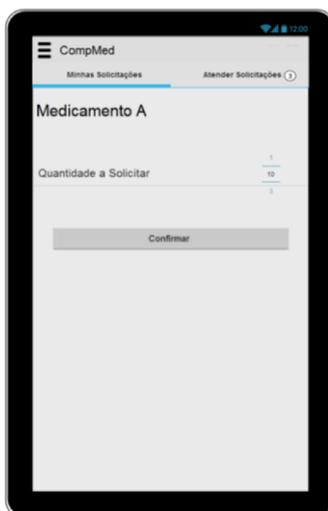
Figura 31 – Tela de nova solicitação.



Fonte: AUTOR, 2016.

Quando o medicamento é escolhido, há a necessidade de entrar com os dados indicando a quantidade solicitada, como ilustra a Figura 32.

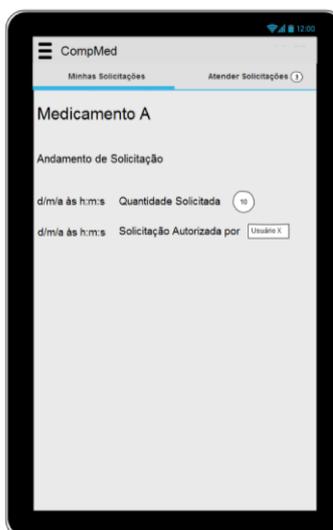
Figura 32 – Tela de seleção de quantidade.



Fonte: AUTOR, 2016.

Após o medicamento ser solicitado, o mesmo aparece na lista de minhas solicitações com o status inicial de “Pendente de Autorização”, que indica que sua solicitação ainda não foi autorizada pelo administrador. A Figura 33 simula quando o item da lista de solicitações, que contém um medicamento já autorizado e com o status de “Pendente de Atendimento”, é selecionado detalhando as informações do andamento da solicitação.

Figura 33 – Tela de andamento de solicitação autorizada.

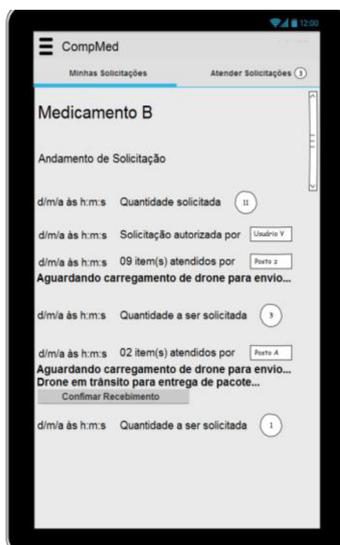


Fonte: AUTOR, 2016.

O medicamento solicitado será visível por outros postos apenas se for previamente autorizado pelo usuário com perfil de Administrador do posto de origem. A partir do momento

que o mesmo é autorizado, todos os usuários de outros postos de saúde poderão contribuir com parte da quantidade solicitada até que toda a quantidade seja suprida. A Figura 34 ilustra o andamento de uma solicitação sendo atendida, detalhando as informações dos postos doadores e o decremento da quantidade inicial ao decorrer dos atendimentos.

Figura 34 – Tela de andamento de solicitação sendo atendida.



Fonte: AUTOR, 2016.

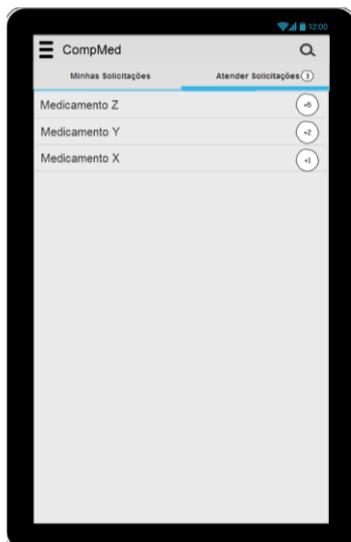
Outro detalhe ilustrado pela Figura 34, é a necessidade que o usuário terá de relatar que a quantidade do medicamento atendida pelo posto doador foi recebida.

Atendimento

A aba denominada de “Atender Solicitações” ilustrada na Figura 35, mostra os medicamentos solicitados por outros postos com as suas devidas quantidades.

Importante ressaltar que as solicitações apenas serão atendidas por completo se e somente se todas as quantidades solicitados pela unidade forem supridas por completo. Desta forma, várias unidades de saúde poderão contribuir para o atendimento de uma mesma solicitação, uma vez que nem todos os postos poderão doar a quantidade requisitada de uma só vez.

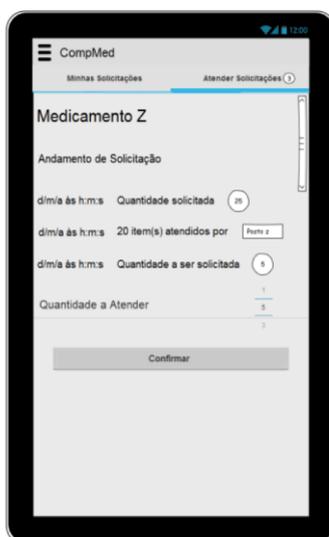
Figura 35 – Tela de lista de solicitações de outros postos.



Fonte: AUTOR, 2016.

Ao usuário selecionar um dos medicamentos da lista especificada na Figura 35, ele terá a opção de atender o medicamento, sendo necessário apenas especificar a quantidade e confirmar, como mostra a Figura 36. Quando uma determinada quantidade é atendida, a quantidade inicial solicitada é decrementada, para que os outros postos atendem apenas o que estiver faltando ser atendido.

Figura 36 – Tela de atendimento de solicitação.



Fonte: AUTOR, 2016.

Histórico

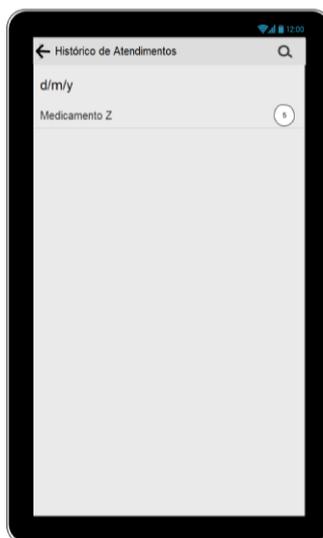
Toda a relação de solicitações e atendimentos podem ser visualizadas pelo usuário no aplicativo. A Figura 37 ilustra a tela de todas as solicitações realizadas pelo usuário e a Figura 38 detalha todas as informações de atendimentos realizados ao longo da utilização do aplicativo.

Figura 37 – Tela de histórico de solicitações.



Fonte: AUTOR, 2016.

Figura 38 – Tela de histórico de atendimentos.



Fonte: AUTOR, 2016.

Menu Vertical acessado pelo Administrador

Um dos requisitos especificado na seção 3.3. é o controle de acesso, que infere na disponibilidade de diferentes funcionalidades dependendo do perfil do usuário. Como exemplo dessas diferenças, a Figura 39 simula a abertura do menu vertical visto pelo usuário com o perfil de Administrador, que contém alguns itens a mais que a visão do Funcionário.

Figura 39 – Tela de menu vertical acessado pelo administrador.



Fonte: AUTOR, 2016.

Tela Principal acessado pelo Administrador

Outra divergência é a tela principal, ilustrada na Figura 40, acessada pelo Administrador que assim como a do Funcionário está dividida em duas abas, porém representam funcionalidades diferentes. A aba “Autorizar Solicitações” detalha todos os medicamentos pendentes de solicitação, e a aba “Autorizar Atendimentos” que lista todos os atendimentos de solicitações que ainda não foram autorizados.

Figura 40 – Tela principal acessado pelo administrador.

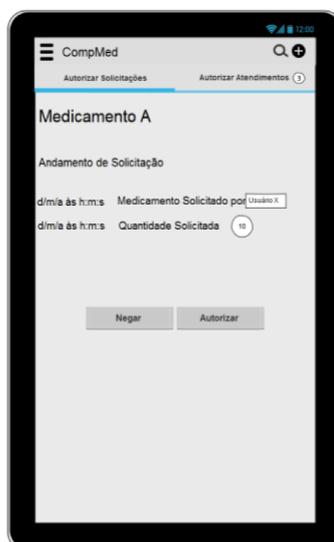


Fonte: AUTOR, 2016.

Autorização de Solicitação

Quando o usuário com perfil de Administrador seleciona um dos itens da lista de solicitações pendentes de autorização, é fornecido informações da quantidade da solicitação e do usuário que realizou a ação. Abaixo das informações, o usuário poderá autorizar ou negar de acordo com a situação do seu posto de saúde, como detalhado na Figura 41.

Figura 41 – Tela de autorização de solicitação.



Fonte: AUTOR, 2016.

Autorização de Atendimento

A aba “Autorizar atendimentos”, detalhada na Figura 42, lista todos os atendimentos dos usuários que ainda não foram atualizados, assim como a quantidade de cada atendimento.

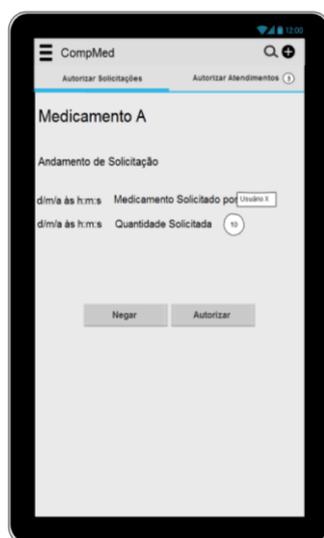
Figura 42 – Tela de lista de atendimentos pendentes de autorização.



Fonte: AUTOR, 2016.

Quando selecionado um dos itens, as informações de atendimento são fornecidas e o Administrador poderá escolher entre autorizar e negar o atendimento em questão, como ilustrado na Figura 43.

Figura 43 – Tela de autorização de atendimento.



Fonte: AUTOR, 2016.

3.8. Memorial Descritivo

Será explanado nessa seção as implementações da solução apresentada neste trabalho, tanto da base de dados quanto do Web Service, do sistema web e do aplicativo.

3.8.1. Base de Dados

Conforme analisado na seção 3.3, foi concluído que ao invés de lidar com o controle de estoque e cadastro de atributos dos medicamentos, seria melhor se ter uma base de dados de todos os medicamentos fornecidos na rede do SUS e de distribuição de medicamentos do governo.

Após análise e pesquisa sobre os principais medicamentos, foi encontrado em uma relação fornecido pela ANVISA e atualizada em 27/03/2009, uma lista de medicamentos e suas respectivas referências com informações de fármaco, laboratório, concentração e forma farmacêutica.

A lista retirada do repositório Lista de Medicamentos de Referência (2009), foi analisada de acordo com os conceitos presentes também no domínio da ANVISA e encontradas no repositório Conceitos Gerais sobre Medicamentos. A lista foi processada utilizando um script desenvolvido em PHP (Apêndice B), na estrutura da modelagem definida na Figura 14 da seção 3.6. Os dados lidos pelo script foram inseridos em um banco de dados MySQL armazenado no servidor remoto.

3.8.2. Web Service

Segundo Menéndez (2009), um Web Service pode ser definido simplesmente como uma aplicação que aceita solicitações de outros sistemas através da Internet, podendo ser identificada por uma URI, cujas interfaces podem ser definidas através de estrutura de dados baseadas em XML ou JSON.

O servidor de aplicação foi desenvolvido com a utilização da ferramenta Node.js e do framework Express para definição de rotas que, de forma abstrata, representam URIs acessíveis por outros sistema para comunicação e transmissão de dados.

Como fase inicial para a validação da ideia, foram estabelecidas as principais rotas de acesso:

- `Medicamento.js`: Rota que fornece os dados registrados na base de medicamentos;
- `Postos.js`: Rota definida para cadastro e consulta de postos cadastrados;
- `Solicitacao.js`: Rota responsável pelo processamento de solicitações efetuadas;
- `Users.js`: Rota que recebe credenciais do usuário e valida autenticação.

3.8.3.Sistema Web

A partir da criação do Web Service e da definição de rotas, foi possível a realização de testes para a transmissão de dados.

Foi iniciado então a etapa de desenvolvimento do sistema web, que como escopo inicial consistia no cadastro de postos, usuários, solicitação de medicamentos e atendimento.

O sistema foi desenvolvido utilizando as tecnologias web atuais HTML5, CSS3, JavaScript e o framework AngularJS. O AngularJS além de auxiliar na definição de controle e interface da aplicação fazia a comunicação com o servidor via HTTP utilizando JSON para transmissão e troca de dados.

A Figura 44 detalha a tela de cadastro do posto e integração com API do Google Maps para auxílio no cadastro do endereço e consulta de informações de latitude e longitude.

Esta funcionalidade de cadastro foi desenvolvida apenas para a realização de testes durante a fase de implementação e agilizou na inserção de dados reais no banco de dados, como por exemplo as informações de latitude e longitude da UPA do Araçagy.

Figura 44 – Tela de cadastro de posto.

CompMed Cadastro ▾ Consulta ▾ Relatório ▾ Administrador UPA Araçagy ▾

Novo Posto de Atendimento

Nome

Endereço
[Buscar](#)

Latitude

Longitude



Fonte: AUTOR, 2016.

A Figura 45 ilustra a tela de cadastro de usuário, utilizada para cadastrar os usuários que possuem acesso ao sistema web e aplicativo.

Figura 45 – Tela de cadastro de usuário.

CompMed Cadastro ▾ Consulta ▾ Relatório ▾ Administrador UPA Araçagy ▾

Novo Usuário

Nome

CPF

RG

Cargo

Email

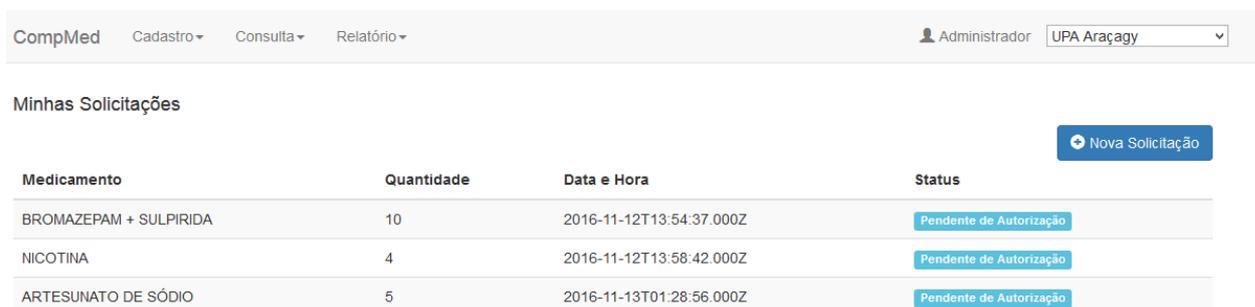
Login de Acesso

Senha de Acesso

Fonte: AUTOR, 2016.

A Figura 46 demonstra a lista de solicitações do usuário e o andamento de cada uma.

Figura 46 – Tela de cadastro de usuário.



Medicamento	Quantidade	Data e Hora	Status
BROMAZEPAM + SULPIRIDA	10	2016-11-12T13:54:37.000Z	Pendente de Autorização
NICOTINA	4	2016-11-12T13:58:42.000Z	Pendente de Autorização
ARTESUNATO DE SÓDIO	5	2016-11-13T01:28:56.000Z	Pendente de Autorização

Fonte: AUTOR, 2016.

Ao realizar uma nova solicitação, o sistema solicita o medicamento e a especificação de quantidade. Ao digitar o nome do medicamento, o sistema efetua a busca na base de dados e mostra para o usuário todos os medicamentos existentes com as informações de concentração e forma farmacêutica, como ilustrado na Figura 47.

Figura 47 – Tela de nova solicitação.



Fonte: AUTOR, 2016.

A Figura 48 ilustra a tela de solicitações realizadas por outros postos de saúde e que ainda não foram atendidas.

Figura 48 – Tela de atendimento de solicitações.



Origem	Endereço	Medicamento	Quantidade	Data e Hora	Status
UPA Araçagy	Avenida dos Holandeses, s/n	ACETATO DE DESMOPRESSINA	10		<input type="button" value="Atender"/>
UPA Cidade Operária	Av. Principal, s/n, Cidade Operária	ACETATO DE LEUPRORRELINA	20		<input type="button" value="Atender"/>
UPA Araçagy	Avenida dos Holandeses, s/n	ACECLOFENACO	13		<input type="button" value="Atender"/>

Fonte: AUTOR, 2016.

3.8.4. Aplicativo

O desenvolvimento do aplicativo também foi inicializado e focado inicialmente na autenticação do usuário em um determinado posto de saúde o qual foi previamente cadastrado.

A ferramenta utilizada para o desenvolvimento do aplicativo foi o NativeScript, previamente detalhado neste trabalho na revisão de literatura e que possibilita a criação de aplicativos móveis nativos a partir de uma abordagem de desenvolvimento multiplataforma. A comunicação assim, como no sistema web, é realizada via protocolo HTTP com o Web Service hospedado em servidor externo.

A Figura 49 detalha a tela inicial do aplicativo móvel com os campos de login e senha.

Figura 49 – Tela inicial de aplicativo.



Fonte: AUTOR, 2016.

A Figura 50 ilustra a seleção do posto de saúde que o usuário se encontra no momento e que realizará solicitações e atendimentos.

Figura 50 – Tela de seleção de posto.



Fonte: AUTOR, 2016.

3.9. Resultados

Além dos objetivos do trabalho aqui proposto, é importante ressaltar o resultado inicial que representa o conhecimento agregado ao decorrer das pesquisas realizadas e da avaliação de metodologias e tecnologias que possibilitam a solução do problema.

Além do conhecimento adquirido e como forma de validar a solução aqui proposta, foi realizado uma entrevista com funcionários que atuam no domínio do problema e que lidam dia-a-dia com as necessidades de unidades de atendimento que atendem, em média, mais de 900 pacientes por dia.

Em virtude da abrangência da área e complexidade do problema, foi elaborado um roteiro com perguntas abertas (Apêndice A) visando confirmar os problemas encontrados em pesquisas e apresentar uma possível solução.

Conforme as perguntas foram respondidas, foi constatado que de fato existe desperdício de medicamentos que se intensifica quando não há o remanejamento dos mesmos, isto é, quando é constatado que em determinada unidade está sobrando e não é tomado nenhuma medida para a

doação destes medicamentos para outras unidades que constam a falta. Atualmente há o remanejamento por meios alternativos, porém devidamente documentados, como por exemplo, o uso de telefone e de aplicativos de comunicação como o WhatsApp.

Quando apresentado a solução proposta neste trabalho, foi questionado de início se a base de dados utilizada possuía relevância e seria válida e constatou-se que a relação de medicamentos de referência da ANVISA seria uma das bases principais, visto que nela contém todos os medicamentos fornecidos pelo SUS, mas não seria a única uma vez que foi ressaltada a existência de diversas relações de medicamentos hospitalares que não estão presentes da relação da ANVISA.

O outro questionamento realizado na entrevista foi a respeito dos benefícios que tal solução aqui proposta traria para a gestão de medicamentos e foi relatado que o benefício principal seria o auxílio de remanejamento e dispensa de medicamentos evitando desperdícios desnecessários de medicamentos que demoram meses para serem adquiridos e repassados para os consumidores finais que são os pacientes.

4. CONSIDERAÇÕES FINAIS

A relevância da otimização de serviços na sociedade com o objetivo do desenvolvimento da mesma é algo crucial que tem chamado a atenção dos governos e empresas. O trabalho proposto teve como objetivo a implementação de um sistema web e aplicativo móvel utilizando tecnologias atuais para auxiliar no remanejamento de medicamentos entre postos de saúde, algo que quando não realizado, gera prejuízo aos cofres públicos.

Através do resultado obtido no trabalho, é possível afirmar que a solução proposta pode ser utilizada para auxiliar na redução de gastos e diminuição de desperdício com medicamentos otimizando redistribuição destes. Um dos principais desafios de projetos assim, além da aceitação por parte do governo, é a parte da logística. Na logística, os custos relacionados ao transporte são muito elevados, e qualquer novo sistema que possa gerar alguma redução de custo é muito valorizado e possui grandes impactos.

Em relação a modelagem e desenvolvimento, buscou-se apresentar um sistema modular que se utiliza de metodologias e tecnologias atuais encontradas no mercado, que possibilitaram a agilidade no processo.

4.1. Trabalhos Futuros

Sugere-se para trabalho futuro a finalização do desenvolvimento do aplicativo e sistema web realizando modificações na base de medicamentos, ampliando a escolha do usuário e adicionando a utilização de sockets para implementar a funcionalidade de notificações.

Além de alterações nas funcionalidades, outra sugestão seria a implementação do projeto de gamificação, realizando testes com o objetivo de avaliar se os usuários efetuam mais ações no sistema.

Pensando em uma ampliação do sistema, outro trabalho futuro seria integrar o gerenciamento da compra de medicamentos possibilitando uma melhor distribuição de acordo com a real demanda de cada unidade de saúde.

REFERÊNCIAS

- ANDERSON, N. J. **Getting Started with NativeScript**. Packt Publishing Ltd., 2016.
- ANDROID. **Android Interfaces and Architecture | Android Open Source Project**. 2016. Disponível em: <<http://source.android.com/devices/index.html>>. Acesso em: 06 nov. 2016.
- ANDROID. **Meet Android Studio | Android Studio**. 2016. Disponível em: <<https://developer.android.com/studio/intro/index.html>>. Acesso em: 06 nov. 2016.
- ANVISA. **Conceitos Gerais sobre Medicamentos**. Disponível em: <<http://www.anvisa.gov.br/hotsite/genericos/profissionais/conceitos.htm>>. Acesso em: 01 ago. 2016.
- ANVISA. **Lista de Medicamentos de Referência**. Março, 2009. Disponível em: <<http://www.anvisa.gov.br/medicamentos/referencia/lista.pdf>>. Acesso em: 01 ago. 2016.
- APPLE INC. **About Objective-C**. 2014. Disponível em: <<https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>>. Acesso em: 06 nov. 2016.
- APPLE INC. **Apple Developer Program - Apple Developer**. 2016. Disponível em: <<https://developer.apple.com/programs/>>. Acesso em: 06 nov. 2016.
- APPLE INC. **News, Apple - Hot News. 2007**. Disponível em: <<http://web.archive.org/web/20071020040652/http://www.apple.com/hotnews>>. Acesso em: 06 nov. 2016.
- BARBOSA, S. **Interação humano computador**. Rio de Janeiro: Elsevier, 2010.
- BASHA, S. Jeelani; CABLE, Scott; GALBRAITH, Ben; HENDRICKS, Mack; IRANI, Romin; MILBERY, James; MODI, Tarak; TOST, Andre; TOUSSAINT, Alex. **Professional java web services**. Birmingham: Wrox Press, 2002.
- BASHAM, B.; SIERRA K.; BATES B. **Head First Servlets & JSP**. 2 Edition. O'REILLY, 2008.

BENEVIDES, C.; GALDO, R. **R\$ 1 bilhão em remédios no lixo: Falhas na compra e no armazenamento levam a desperdícios de medicamentos no país**. Rio de Janeiro, 2010. Disponível em: <<http://www.ecodebate.com.br/2010/04/26/r-1-bi-em-remedios-no-lixo-falhas-nacompra-e-no-armazenamento-levam-a-desperdicio-de-medicamento-no-pais>>.

Acesso em: 29 out. 2016.

BRANSTEIN, M. E.; BRANSTEIN N. J. **NativeScript in Action**. Manning Publications Co., 2016.

BRASIL, CONSELHO NACIONAL DE SECRETÁRIOS DE SAÚDE. **Para entender a gestão do Programa de Medicamentos de dispensação em caráter excepcional**. Conselho Nacional de Secretários de Saúde. Brasília, 2004.

BRITO, P. S.; BASTOS, F. M. **Aquisição de Medicamentos: Um Paralelo Entre A Compra E O Desperdício Dos Medicamentos Adquiridos Pelo Setor Público**. n.1, v.1, p.57-71, 2015.

BUNCHBALL. (2012). **Bunchball**. Disponível em: <<http://www.bunchball.com/>>. Acesso em: 08 out. 2016.

CHANDERMANI. **AngularJS by Example**. Packt Publishing Ltd., 2015.

CORRAL, L.; JANES, A.; REMENCIUS, T. **Potential Advantages and Disadvantages of Multiplatform Development Frameworks - A Vision on Mobile Environments**. Procedia Computer Science, v. 10, p. 1202–1207, jan. 2012. ISSN 1877-0509. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050912005303>>. Acesso em: 08 nov. 2016.

COSTA, A. C. S., MARCHIORI, P. Z. **Gamificação, elementos de jogos e estratégia: uma matriz de referência**. InCID: R. Ci. Inf. e Doc., Ribeirão Preto, v. 6, n. 2, p. 44-65, set. 2015/fev. 2016.

DEITEL, H.M. e DEITEL, P.J. **Java How to Program**. 6th Edition. Prentice Hall, 2004.

DE OLIVEIRA, F. G., SEABRA J. M. P. **Metodologias de desenvolvimento de software: Uma análise no desenvolvimento de sistemas na web**. Periódico Científico Tecnologias em Projeção | v.6 | n.1 Junho 2015

DETERDING, S. **Gamification: designing for motivation**. *Interactions*, v.19, n. 4, p. 14-17, jul./ago., 2012. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2212877.2212883>>. Acesso em: 08 out. 2016.

DETERDING S., DIXON, D., KHALED, R., NACKLE, L. **From game design elements to gamefulness: defining gamification**, Proc. of MindTrek: Envisioning Future Media Environments, 2011.

DO PRADO, Rodolfo Costa. **Aplicação do método FrameWeb no desenvolvimento de um sistema de informação utilizando o framework VRaptor 4**. 2015. 87 f. Trabalho de Conclusão de Curso (Bacharel em Ciência da Computação)-Curso de Ciência da Computação do Departamento de Informática, Universidade Federal do Espírito Santo, 2015.

DRIFTY. **About Ionic - Ionic Components**. 2016. Disponível em: <<http://ionicframework.com/docs/overview/#about>>. Acesso em: 11 nov. 2016.

EL-KASSAS, W. S. et al. **Taxonomy of Cross-Platform Mobile Applications Development Approaches**. *Ain Shams Engineering Journal*, 2015.

FOLDIT. Disponível em: <<https://fold.it/portal/>>. Acesso em: 07 out. 2016.

FORD, B.; RUEBBELKE, L. **AngularJS in Action**. Manning Publications Co., 2015.

GAMMA, E. **Padrões de Projeto: soluções de software orientado a objetos**. Porto Alegre: Bookman, 2000.

HEITKOTTER, H.; HANSCHKE, S.; MAJCHRZAK, T. A. **Comparing Crossplatform Development Approaches for Mobile Applications**. *Web information systems and technologies*, 2013.

HOLZINGER, A.; TREITLER, P.; SLANY, W. **Making Apps Useable on Multiple Different Mobile Platforms: On Interoperability for Business Application Development on Smartphones**. In: QUIRCHMAYR, G. et al. (Ed.). *Multidisciplinary Research and Practice for Information Systems*. Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, 7465). p. 176–189. ISBN 978-3-642-32497-0 978-3-642-32498-7. DOI: 10.1007/978-3-642-32498-7_14. Disponível em: <http://link.springer.com/chapter/10.1007/978-3-642-32498-7_14>. Acesso em: 08 nov. 2016.

JANNUZZI P.M. **Avaliação de Programas Públicos por meio da análise estruturada dos relatórios de auditoria da Controladoria Geral da União**. Cadernos Gestão Pública e Cidadania. São Paulo, n.59, v. 16, 2011. MARIN N. et al. **Assistência Farmacêutica para Gerentes Municipais**, Rio de Janeiro: OPAS/MS 2003.

JOBE, W. **Native Apps Vs. Mobile Web Apps**. International Journal of Interactive Mobile Technologies (iJIM), v. 7, n. 4, p. 27, out. 2013. ISSN 1865-7923.

JUUL, J. **Half-real: video games between real rules and fictional worlds**. MIT Press, Cambridge, Ma, 2005.

KREGER, H. **Web Services Conceptual Architecture (WSCA 1.0)**. IBM Software Group, 2001.

LENGSTORF, J., LEGGETTER, P. **Realtime Web Apps: With HTML5 WebSocket, PHP, and jQuery**. Springer Science and Business Media New York.

LOPES, Camilo. **Serie 2:Web Service, Aplicação & Benefícios**. Disponível em: <<http://blog.camilolopes.com.br/serie-2-web-service-aplicacao-beneficios/>>. Acesso em: 10 dez. 2016.

MARIN N. et al. **Assistência Farmacêutica para Gerentes Municipais**, Rio de Janeiro: OPAS/MS 2003.

MARINS, Diego Ribeiro. **Um processo de gamificação baseado na teoria da autodeterminação**. 2013. 125 f. Dissertação (Mestrado em Engenharia de Sistemas e Computação) - Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia, Universidade Federal do Rio de Janeiro, 2013.

MATOS, B. R. D.; E SILVA, J. G. De B. **Estudo Comparativo entre o desenvolvimento de aplicativos móveis utilizando plataformas nativas e multiplataforma**. 2016. 63 f. Trabalho de Conclusão de Curso (Bacharel em Engenharia de Software) - Curso de Engenharia de Software, Universidade Brasília, 2016.

MENÉNDEZ, A. I. M. **Uma ferramenta de apoio ao desenvolvimento de Web Services**. Dissertação de Mestrado, Universidade Federal de Campina Grande, curso de Pós-Graduação em Informática, 2002.

NIELSEN J. **Heuristic Evaluation**. New York: John Wiley & Sons; 1994.

NODE. **Node.js**. Disponível em: <<https://nodejs.org/en/>>. Acesso em: 18.ago. 2016.

NODE. **Node.js**. Disponível em: <<https://nodejs.org/api/>>. Acesso em: 18 ago. 2016.

PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. 7ª Edição. São Paulo: AMGH Editora Ltda, 2011.

PRESSMAN, R. S; LOWE, David. **Web engineering: a practitioner's approach**. McGraw-Hill, 2009.

PREZOTTO, E.; BONIATI, B. **Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas**. 2014.

PORTELA, A.S. et al. Políticas Públicas de Medicamentos: trajetória e desafios, **Revista de Ciências Farmacêuticas Básica e Aplicada**, n.1, v. 31, p. 09-14, 2010.

RECKZIEGEL, Mauricio. **Entendendo os WebServices**. Disponível em: <<http://imasters.com.br/artigo/4245/web-services/entendendo-os-webservices/>>. Acesso em: 10 dez. 2016.

REENSKAUG, T. **THING-MODEL-VIEW-EDITOR, an Example from a planning system**. 1979.

REZENDE, D. **Engenharia de software e sistemas de informação**. BRASPORT LIVROS E Multimídia Ltda., 2005.

RIBEIRO, C. **Aplicações Web real-time com Node.js**. Casa do Código, 2015.

SNIJDERS, R. et al. **REfine: A Gamified Platform for Participatory Requirements Engineering**.

SOMMERVILLE, I. **Engenharia de Software**. 8ª Ed. São Paulo: Pearson, 2007.

SOUZA, V. E. S. **FrameWeb: um Método baseado em Frameworks para o Projeto de Sistemas de Informação Web**. Dissertação (Programa de Pós-Graduação em Informática) - Universidade Federal do Espírito Santo, 2007.

STARK, J. **Building iPhone Apps with HTML, CSS, and JavaScript**. O'Reilly Media, 2010.

VIANNA, Ysmar et al. **Gamification, Inc: como reinventar empresas a partir de jogos**. 1. Ed. – Rio de Janeiro: MJV Press, 2013.

VIEIRA F.S.; ZUCCHI P. **Distorções causadas pelas ações judiciais à política de medicamentos no Brasil**. Revista de Saúde Pública, n.2, v.41, p.214-222, 2007.

WERBACH, K., HUNTER, D. (2012). **For the Win: How Game Thinking Can Revolutionize Your Business**. W. D. Press, Ed. Wharton Digital Press.

WILSON, J. R. **Node.js the Right Way: Practical, Server-Side JavaScript That Scales**. Pragmatic Bookshelf, 2013.

WOLFSON, M. **Android Developer Tools Essentials**. 1ª Ed. Sebastopol, California: O'Reilly, 2013.

ZICHERMANN, G., CUNNINGHAM, C. (2011). **Gamification by design: Implementing game mechanics in web and mobile apps**. O. Media, Ed. O'Reilly Media.

APÊNDICE A – ROTEIRO DE ENTREVISTA

1. Faltam, de fato, medicamentos na unidade?
2. Quando há registros de falta de medicamentos, qual a instrução para a população?
3. Como se dá o controle de estoques dos medicamentos?
4. Existe um modelo de fornecimento gratuito na unidade?
5. Quais os meios de divulgação do estoque de medicamentos?
6. Há meios emergenciais de aquisição de medicamentos, que não se subordine às licitações?
7. Existe o compartilhamento de medicamentos entre outras unidades?
8. Se existe o compartilhamento, o mesmo é permitido por lei? Como se dá esse compartilhamento?
9. Existe um controle efetivo, informatizado pela unidade de medicamentos que são descartados?
10. Na sua opinião, quais benefícios um sistema como esse traria aos postos de saúde?

APÊNDICE B – SCRIPT PHP BASE DE DADOS

```

include_once __DIR__ . '/inc/_all.php';

// Abre conexão com o BD
$conn = GetConnection();

// Arquivo a ser lido
$arquivo = './lista.csv';

// Processamento
if(is_file($arquivo)){//SE LEU O ARQUIVO...

// Abrir arquivo
$lendo = @fopen($arquivo, "r");

// Falha em abertura de arquivo
if (!$lendo){
echo "Erro ao abrir a URL.";
exit;
}

$i = 0;

$qtydRegistros = 0; //Contador de Registros Lidos

//LÊ A 1ª LINHA DO ARQUIVO (CABECALHO)
$linha = fgets($lendo,9999);

while (!feof($lendo)){ //PARA CADA LINHA DO ARQUIVO

$i++;

$linha = fgets($lendo,9999);

// Se possui campos
if (strpos($linha, ",") {

$linha_legivel = explode(",", $linha);

$qtydRegistros++;

echo "<br/><br/><br/>";

//print_r($linha_legivel);

echo "<br/><br/><br/>";

$farmaco = (empty($linha_legivel[0])) ? null : $linha_legivel[0];

$laboratorio = (empty($linha_legivel[1])) ? null : $linha_legivel[1];

$medicamentoreferencia = (empty($linha_legivel[2])) ? null : $linha_legivel[2];

$concentracao = (empty($linha_legivel[3])) ? null : $linha_legivel[3];

```

```

$formafarmaceutica = (empty($linha_legivel[4])) ? null : $linha_legivel[4];

// Cadastra Informações
$insert = "INSERT INTO basemedicamento_copy(farmaco, laboratorio, medicamentoreferencia, concentracao,
formafarmaceutica, created_at) VALUES('$farmaco', '$laboratorio', '$medicamentoreferencia', '$concentracao',
'$formafarmaceutica', Now())";

$results = $conn -> query($insert);

$error = $conn -> error;

if ($error) {

    printr($error);

    exit("<br/><br/>Erro em inserção da linha $i");
}

}

} //FIM PARA CADA LINHA DO ARQUIVO...

// Fecha Arquivo
fclose($lendo);

// Fecha Conexão
$conn -> close();

if($qtdRegistros > 0){ //SE HOUVE ALGUM PAGAMENTO...

    echo "<b>$qtdRegistros registros lidos</b>";

} else{ //SE NENHUM PAGAMENTO FOI ENCONTRADO

    echo "Nenhum Registro Lido!";

}

} else{ //SE NÃO LEU O ARQUIVO

    echo "<b>Erro na Leitura do Arquivo!</b><br />";

}

// Laboratorio
$select = "SELECT DISTINCT laboratorio FROM basemedicamento_copy ORDER BY laboratorio
ASC";

$ret = FetchAll($select);

for($i = 0; $i < count($ret); $i++){

    $descricao = $ret[$i]['laboratorio'];

    $insert = "INSERT INTO laboratorio(nome) VALUES('$descricao)";

```

```

$results = $conn -> query($insert);

$error = $conn -> error;

if ($error) {

    print_r($error);

    exit("<br/><br/>Erro em inserção da linha $i do laboratorio<br/><br/>");
}
}

// Forma Farmaceutica
$select = "SELECT DISTINCT formafarmaceutica FROM basemedicamento_copy ORDER BY
formafarmaceutica ASC";

$ret = FetchAll($select);

for($i = 0; $i < count($ret); $i++){

    $descricao = $ret[$i]['formafarmaceutica'];

    $insert = "INSERT INTO formafarmaceutica(descricao) VALUES('$descricao')";

    $results = $conn -> query($insert);

    $error = $conn -> error;

    if ($error) {

        print_r($error);

        exit("<br/><br/>Erro em inserção da linha $i do formafarmaceutica<br/><br/>");
    }
}

// Base Medicamento
$select = "SELECT farmaco,laboratorio,medicamentoreferencia,concentracao,formafarmaceutica
FROM basemedicamento_copy ORDER BY farmaco ASC";

$ret = FetchAll($select);

for($i = 0; $i < count($ret); $i++){

    $farmaco = $ret[$i]['farmaco'];

    $medicamentoreferencia = $ret[$i]['medicamentoreferencia'];

    $concentracao = $ret[$i]['concentracao'];

    $laboratorio = $ret[$i]['laboratorio'];

    $formafarmaceutica = $ret[$i]['formafarmaceutica'];
}

```

```
$selectLaboratorio = "SELECT idlaboratorio FROM laboratorio WHERE nome = '$laboratorio'";  
  
$rs = FetchAll($selectLaboratorio);  
  
$laboratorio = $rs[0]['idlaboratorio'];  
  
$selectFormaFarmaceutica = "SELECT idformafarmaceutica FROM formafarmaceutica WHERE  
descricao = '$formafarmaceutica'";  
  
$rs = FetchAll($selectFormaFarmaceutica);  
  
$formafarmaceutica = $rs[0]['idformafarmaceutica'];  
  
$insert = "INSERT INTO basemedicamento(nome, medicamentoreferencia, concentracao,  
idlaboratorio, idformafarmaceutica) VALUES('$farmaco', '$medicamentoreferencia', '$concentracao',  
$laboratorio, $formafarmaceutica)";  
  
if($i >= 1308){  
  
$results = $conn -> query($insert);  
  
$error = $conn -> error;  
  
if ($error) {  
  
echo $insert . "<br/>";  
  
print_r($error);  
  
exit("<br/><br/>Erro em inserção da linha $i da basemedicamento<br/><br/>");  
}  
}  
}  
  
// Fecha Conexão  
$conn -> close();
```