

**UNIVERSIDADE FEDERAL DE SANTA MARIA
COLÉGIO AGRÍCOLA DE FREDERICO WESTPHALEN
PÓS-GRADUAÇÃO EM GESTÃO DE TECNOLOGIA DA
INFORMAÇÃO**

**APLICAÇÃO PARA GERENCIAMENTO DE PROJETOS DE
SOFTWARE UTILIZANDO SCRUM COM INTERFACE
ADAPTADA PARA DISPOSITIVOS MÓVEIS**

MONOGRAFIA DE PÓS-GRADUAÇÃO

Dioni da Rosa

Frederico Westphalen, RS, Brasil

2013

APLICAÇÃO PARA GERENCIAMENTO DE PROJETOS DE
SOFTWARE UTILIZANDO SCRUM COM INTERFACE
ADAPTADA PARA DISPOSITIVOS MÓVEIS

Dioni da Rosa

Monografia de conclusão do Curso de Pós-Graduação em Gestão da
Tecnologia da Informação, da Universidade de Santa Maria, Colégio
Agrícola de Frederico Westphalen (UFSM / CAFW), como requisito
parcial para a obtenção do grau de
Especialista em Gestão da Tecnologia da Informação

Orientador Prof.^a Teresinha Letícia da Silva

Frederico Westphalen, RS, Brasil.

2013

© 2013

Todos os direitos autorais reservados a Dioni da Rosa. A reprodução de partes ou do todo deste trabalho só poderá ser feita com autorização por escrito do autor.

Endereço: Rua Rio Negro, n. 185- Apto 201, Bairro da Itapagé, Frederico Westphalen, RS, 98400-000

Fone (0xx)55 99031371; End. Eletrônico: dioni.r@live.com

**Universidade Federal de Santa Maria
Colégio Agrícola de Frederico Westphalen
Pós-Graduação em Gestão de Tecnologia da Informação**

**A Comissão Examinadora, abaixo assinada,
aprova a Monografia de Pós-graduação**

**APLICAÇÃO PARA GERENCIAMENTO DE PROJETOS DE
SOFTWARE UTILIZANDO SCRUM COM INTERFACE
ADAPTADA PARA DISPOSITIVOS MÓVEIS**

Elaborado por:
Dioni da Rosa

como requisito parcial para a obtenção de grau de
Especialista em Gestão da Tecnologia da Informação

COMISSÃO AVALIADORA

Teresinha Letícia da Silva, Ms.(UFSM/CAFW)
(Presidente Orientadora)

Adriana Soares Pereira, Dra.(UFSM/CAFW)

Bruno Batista Boniati, Ms.(UFSM/CAFW)

Frederico Westphalen, 19 de Julho de 2013.

Dedico meu trabalho as pessoas que estiveram ao meu lado e me apoiaram para que o mesmo fosse consolidado.

Agradecimento

Agradeço a Deus, por esse momento e pelos já idealizados nesta etapa de minha vida.

A família, por todo o apoio, compreensão, dedicação e incentivo durante as etapas de nossa vida.

A todos os outros membros da família que participaram do nosso crescimento profissional e pessoal.

Aos amigos que me ajudaram em momentos de dificuldade e comemoram comigo os momentos de alegria.

A Ana Paula Noro Grabowski pelo apoio, paciência, carinho e pela ajuda nos momentos de indecisão.

À orientadora Teresinha Letícia da Silva, que ajudou dando um rumo ao trabalho.

Em fim a todos aqueles que me ajudaram e me apoiaram de alguma forma para que essa etapa fosse cumprida.

“As pessoas não sabem o que querem, até mostrarmos a elas.”

Steve Jobs

RESUMO

Monografia

Pós-Graduação em Gestão da Tecnologia da Informação

Universidade Federal de Santa Maria – UFSM

Colégio Agrícola de Frederico Westphalen – CAFW

APLICAÇÃO PARA GERENCIAMENTO DE PROJETOS DE SOFTWARE UTILIZANDO SCRUM COM INTERFACE ADAPTADA PARA DISPOSITIVOS MÓVEIS

Autor: Dioni da Rosa

Orientadora: Teresinha Letícia da Silva

Local da Defesa e Data: Frederico Westphalen, 19 de julho de 2013.

A crescente evolução da tecnologia e redução de custos dos dispositivos móveis tem permitido a adaptação e execução de aplicações para estes dispositivos, as quais podem ser aplicação Web (*World Wide Web*) ou Nativas. Neste sentido, o trabalho em pauta aborda a adaptação de interface de um modelo de gerenciamento de projeto e tem como procedimento central do trabalho o estudo dos conceitos básicos e recursos disponíveis para serem utilizados em aplicações web para dispositivos móveis, bem como, a implementação de um protótipo de aplicação que integre a utilização da metodologia ágil para o desenvolvimento de software denominada *Scrum* para ser utilizada em um dispositivo móvel. Também, descreve alguns métodos disponíveis que podem ser utilizados para a melhor prática na gestão a fim de aperfeiçoar ou potencializar a qualidade e agilidade nos processos de desenvolvimento e engenharia de software, além de apresentar alguns conceitos de desenvolvimento para web responsiva com HTML5 e CSS3 e jQueryMobile visando facilitar a implementação de aplicações que possam ser executadas em vários dispositivos independente de tamanho de tela, sem a necessidade de reescrever códigos para determinadas plataformas ou sistema operacional.

Palavras-Chave: *Scrum*, HTML5, WEB Responsiva, Dispositivos Móveis.

ABSTRACT

Monograph

Postgraduate in Management of Information Technology

Universidade Federal de Santa Maria – UFSM

Colégio Agrícola de Frederico Westphalen – CAFW

APPLICATION FOR MANAGEMENT PROJECT SOFTWARE USING SCRUM WITH INTERFACE ADAPTED FOR MOBILE

Author: Dioni da Rosa

Advisor: Teresinha Letícia da Silva

Place and Date of Defense: Frederico Westphalen, 19 of July of 2013.

The growing evolution of technology and cost reduction of the mobile devices has made it possible the adaptation and execution of applications for these devices, which can be an application Web (World Wide Web) or Nativas. In this sense, the work in question discusses the interface adaptation a model for project management, the central procedure of the work is the study of the basic concepts and resources available to be utilized in web applications for mobile devices, as well as the implementing of a prototype of a application that integrates the utilization of agile methodology Scrum for the software development for be used on a mobile device. We also describes some methods available which can be utilized to the best practice in the management, in order to improvements or potentiate the quality and agility in development processes and software engineering, besides presenting some concepts of web development responsive with HTML5 and CSS3 and jQueryMobile, to facilitate the implementation of applications that can run on multiple devices irrespective of screen size, without the need to rewrite codes for determined platforms or operating system.

Keywords: Scrum, HTML5, WEB Responsive, Mobile devices.

LISTA DE ILUSTRAÇÕES

FIGURA 1 - PDA – FONTE: (ANALYSIS TECNOLOGIA, 2013).....	21
FIGURA 2 - SMARTPHONE- FONTE: (GSMARENA, 2013).....	21
FIGURA 3 - REPRESENTAÇÃO DE TABLETS – FONTE: (PITCHENGINE, 2013).....	22
FIGURA 4 - CÓDIGO DE WSDL - FONTE ARQUIVOS PESSOAIS	28
FIGURA 5 - EXEMPLO WEB SERVICE – FONTE: ARQUIVOS PESSOAIS.....	29
FIGURA 6 - EXEMPLO DE APLICAÇÃO NATIVA E APLICAÇÃO WEB – FONTE: (SIXREVISIONS, 2012)...	31
FIGURA 7 - ESTRUTURA DO PHONEGAP – FONTE: (PHONEGAP, 2013).	35
FIGURA 8 - INDICAÇÃO DE RECURSOS NATIVOS DO SO SUPORTAS PELO FRAMEWORK PHONEGAP. FONTE: (PHONEGAP, 2013).....	35
FIGURA 9 - REPRESENTAÇÃO DO WEB DESIGN RESPONSIVO – FONTE: (ZEMEL, 2012).	37
FIGURA 10 - REPRESENTAÇÃO DE CÓDIGO MEDIA QUERIES – FONTE: ARQUIVOS PESSOAIS.	39
FIGURA 11 - CÓDIGO <i>DOCTYPE</i> – FONTE: ARQUIVOS PESSOAIS	42
FIGURA 12 - REPRESENTAÇÃO DO MODELO CLÁSSICO – FONTE: ARQUIVOS PESSOAIS, BASEADO EM (SBROCCO E MACEDO, 2012).....	44
FIGURA 13 - REPRESENTAÇÃO DO MODELO ÁGIL XP	47
FIGURA 14 - REPRESENTAÇÃO DA METODOLOGIA AGIL – SCRUM. FONTE: (COHN, 2011).	48
FIGURA 15 - DIAGRAMA ENTIDADE RELACIONAMENTO DA BASE DE DADOS.....	54
FIGURA 16 - CASO DE USO USUÁRIO - <i>SCRUM MASTER</i>	55
FIGURA 17 - CASO DE USO DO USUÁRIO - PROGRAMADOR	56
FIGURA 18 - MARCAÇÃO MEDIA SCREEN NO CSS.....	57
FIGURA 19 - APRESENTAÇÃO DA TELA DE LOGIN EM RESOLUÇÕES MAIORES E MENORES QUE 480PX.....	58
FIGURA 20 - PAINEL DE TAREFAS DA TELA PRINCIPAL	59
FIGURA 21 - TELA DE REGISTRO DE OCORRÊNCIAS	59
FIGURA 22 - PAINEL DO MENU E O CÓDIGO DE SUA MONTAGEM.....	60
FIGURA 23 - TELA DO CADASTRO DE USUÁRIOS	61
FIGURA 24 - TELA DE CADASTRO DO PRODUTO BACKLOG.....	62
FIGURA 25 - TELA DE CADASTRO DE SPRINT.....	63
FIGURA 26 - TELA DE CADASTRO DE TAREFAS DA SPRINT	63
FIGURA 27 - CADASTRO DA ATA DE REUNIÃO.....	64
FIGURA 28 - APLICAÇÃO SENDO VISUALIZADA EM UM <i>SMARTPHONE</i>	65
FIGURA 29 - TELA PRINCIPAL SENDO VISUALIZADA EM UM <i>NOTEBOOK</i>	66

LISTA DE QUADROS

QUADRO 1 - ESPECIFICAÇÃO DOS MEDIA TYPE.....	38
QUADRO 2 - TIPOS DE MEDIA QUERIES	39
QUADRO 3 - 3 NÍVEIS DE SUPORTE DO JQUERY MOBILE	40
QUADRO 4 - NOVOS ELEMENTOS QUE FAZEM PARTE DO HTML5	41
QUADRO 5 - CARACTERÍSTICAS DO NOTEBOOK	52
QUADRO 6 - CARACTERÍSTICAS DO CASO DE USO: USUÁRIO – LOGIN	56
QUADRO 7 - CARACTERÍSTICAS DO SMARTPHONE SONY ERICSSON XPERIA NEO V	65

LISTA DE ABREVIATURAS E SIGLAS

3D: 3 Dimension

3G: Terceira geração de padrões e tecnologias de telefonia móvel

AJAX: Asynchronous JavaScript and XML

APP: Aplicação

ASN.1: Abstract Syntax Notation 1

CDMA: Code Division Multiple Access

CSMA/CA: Carrier Sense Multiple Access Protocol With Collision Svoidance

CSS: Cascading Style Sheets

DER: Diagrama de Entidade e Relacionamento

DSM: Adaptative Software Development

EDGE: Enhanced Data rates for GSM Evolution

Ghz: Gigahertz

GPRS: General packet radio service

GPS: Global Positioning System

GSM: Global System for Mobile

HTML: HyperText Markup Language

IDE: Integrated Development Environment

J2ME: Java 2 Micro Edition

LAN: Local Area Network

MAC: Media Access Control

MVC: Model View Controller

PC: Personal Computer ou Computador Pessoal

PCMCIA: Personal Computer Memory Card International Association

PDA: Personal Digital Assistant

PHP: Hypertext Preprocessor

PMBOK: Project Management Body of Knowledge

PMI: Project Management Institute

PX: Pixel

RAM: Random Access Memory

SDK: Software Development Kit

SDL: Specification and Description Language

SGBD: Sistema de Gerenciamento de Banco de Dados

SIG - Bluetooth Special Interest Group

Slots: Sua função é ligar os periféricos em espaços disponíveis

SO: Sistema Operacional

SOAP: Simple Object Access Protocol

SQL: Structured Query Language

TCP/IP: Transmission Control Protocol/Internet Protocol

TDMA: Time Division Multiple Access

UI: User Interface

W3C: World Wide Web Consortium

WDA: Wireless Digital Assistants

WEB: World Wide Web

WI-FI: inglês Wireless Fidelity, ou Fidelidade Sem Fio

WLAN: Wireless Local Area Network

WSDL: Web Service Description Language

XML: eXtensible Markup Language

XP: Extreme Programming

SUMÁRIO

INTRODUÇÃO	16
1 COMPUTAÇÃO MÓVEL	19
1.1 TIPOS DE DISPOSITIVOS	19
1.1.1 PDA (Personal Digital Assistant)	20
1.1.2 Smartphones	21
1.1.3 Tablets	22
1.2 SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS	22
1.2.1 Windows Phone	23
1.2.2 iPhone OS	23
1.2.3 Android	23
1.3 TECNOLOGIAS E MEIOS DE COMUNICAÇÃO MÓVEL	24
1.3.1 Wi-Fi (Wireless Fidelity)	24
1.3.2 Bluetooth	25
1.3.3 EDGE	26
1.3.4 3G	27
1.4 TRANSFERÊNCIA DE DADOS ENTRE DISPOSITIVO E SERVIDOS	27
1.5 APLICAÇÕES PARA DISPOSITIVOS MÓVEIS (SMARTPHONE E TABLETS)	29
1.5.1 Aplicativo Nativo	29
1.5.2 Aplicações Web	30
1.5.3 Método de instalação e execução das aplicações	30
1.5.4 Vantagens das aplicações móveis	31
1.5.5 Desvantagens das aplicações	32
1.5.6 Ferramentas para o desenvolvimento de aplicações móveis	32
2 ADAPTAÇÃO DE INTERFACES DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS	37
2.1 LAYOUT FLUIDO	37
2.2 IMAGENS E RECURSOS FLEXÍVEIS	38
2.3 MEDIA QUERIES	38
2.4 JQUERY MOBILE	40
2.5 HTML 5	41
3 METODOLOGIAS DE GERENCIAMENTO DE PROJETO DE SOFTWARE	43
3.1 METODOLOGIAS TRADICIONAIS	43
3.1.1 Modelo Clássico	44
3.1.2 PMBOK (Project Management Body of Knowledge)	44
3.2 METODOLOGIAS ÁGEIS	45
3.2.1 Extreme Programming (XP)	46
3.2.2 SCRUM	47
3.3 TRABALHOS RELACIONADOS	50
4 DESENVOLVIMENTO DE UMA APLICAÇÃO PARA GERENCIAMENTO DE PROJETOS DE SOFTWARE ADAPTADA PARA DISPOSITIVOS MÓVEIS	52

4.1 FERRAMENTAS UTILIZADAS	52
4.2 PROTÓTIPO	53
4.2.1 Estrutura da base de dados.....	53
4.2.2 Casos de uso e diagramas da aplicação.....	54
4.2.3 Telas do protótipo	57
4.3 TESTES DA APLICAÇÃO.....	65
4.4 CONSIDERAÇÕES FINAIS	67
5 CONCLUSÕES E TRABALHOS FUTUROS.....	68
5.1 TRABALHOS FUTUROS	69
6 REFERÊNCIAS BIBLIOGRÁFICAS	70
APÊNDICES.....	76
APÊNDICE A – DIAGRAMA ENTIDADE RELACIONAMENTO DA BASE DE DADOS.....	77

INTRODUÇÃO

No contexto de gerenciamento de projetos a utilização de metodologias ágeis no desenvolvimento de software cresce muito, isto ocorre pela necessidade de mudar os modelos tradicionais como prototipagem e incremental, que são utilizados no desenvolvimento e gerenciamento de projetos.

Através das metodologias ágeis existentes, como o *Scrum*, podem-se gerenciar melhor projetos de curta e longa duração, viabilizando a possibilidade de alterações constantemente no decorrer projeto. Desta forma, o principal objetivo da utilização de metodologias ágeis é proporcionar ao cliente a possibilidade de proveito da aplicação o mais rápido possível, fazendo com que o cliente receba constantemente partes do software na medida em que são concluídas.

O presente trabalho aborda a adaptação do *Scrum* para a utilização em dispositivos móveis visando o gerenciamento de projetos de software. Com a mesma, buscou-se verificar a disponibilidade e o funcionamento de determinados recursos e seus funcionamentos em diferentes dispositivos, independente de marca, resolução de tela ou sistema operacional.

Evidencia-se como proposta do trabalho o estudo dos conceitos básicos que englobam a utilização de aplicação web e aplicação nativa para adaptação de interfaces em dispositivos móveis. Para tanto, desenvolveu-se um protótipo de uma aplicação que demonstre algumas funcionalidades de uma metodologia ágil de gerenciamento de software utilizável em diferentes dispositivos móveis.

A popularização dos celulares, *smartphones*, *tablets* não somente possibilitou mais praticidade no acesso às informações e a comunicação, como também, fez emergir a necessidade de dominar a utilização e o manuseio de tais dispositivos. A evolução destes e das tecnologias de comunicação tais como *wireless* e *Bluetooth*, entre outros recursos que os mesmos possuem, estão dando credibilidade em sua utilização na área de gerenciamento de software pelas empresas de desenvolvimento de aplicações, pois, devido ao seu tamanho torna-se mais fácil estar com o dispositivo sempre presente a fim de gerenciar melhor os procedimentos internos e externos durante a elaboração de um

software. Apesar das restrições quanto ao seu tamanho e interface, existem muitos recursos que podem ser utilizados para melhorar a utilização da tela diminuída dos dispositivos.

A necessidade de utilização de interfaces que se adaptem em diversos tipos de dispositivos móveis, buscando facilitar a interação com os usuários, faz emergir a exigência de diferentes aplicações, as quais sejam bem elaboradas, mais complexas e objetivas que possam ser utilizadas em diferentes aparelhos.

Com as tecnologias mais recentes tornou-se possível criar aplicações com interfaces que se adaptam a diferentes tipos de dispositivos dando uma maior qualidade e usabilidade aos dispositivos móveis. Um exemplo são as tecnologias de criação para web apps (*application*), que podem ser utilizadas em qualquer dispositivo móvel que suporte os recursos disponibilizados pela mesma. Outra tecnologia são as aplicações nativas que dependem de um desenvolvimento específico para o sistema operacional (SO) do dispositivo, pois, este compreende uma programação diferenciada cujo SDK (*Software Development Kit*) é disponibilizado pela empresa desenvolvedora do SO (Sistema Operacional).

Um dos problemas encontrados na adaptação de um modelo de gerenciamento de software para ser utilizado em dispositivos móveis é a limitação de recursos que se diferem a cada dispositivo. Apesar dos avanços tecnológicos, esses dispositivos ainda apresentam deficiências que aos poucos vem sendo reestruturadas visando facilitar aos usuários a sua utilização, exigindo assim do desenvolvedor a habilidade de aliar simplicidade com funcionalidade e riqueza de detalhes. Porém com a adaptação de algumas características que envolvem um modelo de gerenciamento de *software*, pode-se aliar simplicidade e usabilidade a uma aplicação, podendo ser utilizado por quaisquer dispositivos sem maiores exigência de recursos.

Mediante tal elaboração, o presente trabalho aborda o desenvolvimento de uma aplicação para gerenciamento de projetos de software utilizando a metodologia *Scrum* adaptado para dispositivos móveis. Também apresenta o estudo de algumas ferramentas para criação de aplicativos que possam ser utilizados em vários dispositivos, independente de SO ou tamanho de tela, além da elaboração de um protótipo de uma aplicação que utilize alguns dos recursos investigados.

Este trabalho apresenta o estudo de algumas tecnologias da computação móvel disponíveis para programação de aplicações, comunicação, e outros recursos que são apresentados no segundo capítulo. O terceiro capítulo apresenta recursos que podem ser

utilizados para a adaptação de interfaces para dispositivos móveis, a descrição de alguns conceitos de desenho e adaptação para vários tipos de resolução de tela. Já no quarto capítulo são apresentados alguns conceitos de metodologias de desenvolvimento de software disponíveis para serem utilizadas e também são apresentados alguns trabalhos relacionados ao tema abordado e que também utilizam recursos como HTML5 no desenvolvimento de aplicações para dispositivos móveis. No quinto capítulo é apresentado o desenvolvimento do trabalho com a descrição do protótipo desenvolvido, utilizando alguns recursos apresentado no trabalho. Por fim são feitas as conclusões e sugestões de trabalhos futuros.

1 COMPUTAÇÃO MÓVEL

Tecnologias móveis estão em contínuo avanço em termos de disponibilidade, funcionalidade e custos, tornando sua utilização mais atraente. Essas tecnologias permitem o desenvolvimento de uma grande variedade de aplicações, sendo estas delineadas pelos recursos disponíveis nos dispositivos. As aplicações podem ser nativas quando são desenvolvidas para aparelhos ou sistema operacional específico, ou aplicações web que podem ser acessadas de qualquer aparelho através de uma rede. Ambas se utilizam de interfaces adaptadas, que utilizam recursos disponíveis nos aparelhos (Exemplo: acesso a internet, GPS, câmera, etc).

Com a melhoria dos meios de comunicação e a facilidade na aquisição de dispositivos móveis, a utilização desses dispositivos cresceu rapidamente, devido comodidade e à mobilidade possibilitada por eles. Também melhorias na tecnologia para um melhor desempenho foram empregadas a esses dispositivos, dando oportunidade para o desenvolvedor inovar e gerando facilidade na utilização das aplicações elaboradas e adaptando processos que eram feitos apenas em *desktops* locais para serem utilizados em aparelhos móveis.

Também está aumentando a visibilidade dos desenvolvedores e empresas para esse mercado, onde o investimento em tecnologias e a criação de software para o mercado eletrônico móvel começa ser uma opção das empresas, abrindo assim novas fronteiras e oportunidades de negócio e gerenciamento, que visam facilitar o acesso e uso dos usuários. (HSIEH, 2007).

A mobilidade altera em muito a maneira como as pessoas e os clientes interagem, comunicam-se e colaboram entre si, e se espera que as aplicações móveis alterem a maneira como vivemos, jogamos e negociamos. (TURBAN e KING, 2004, p. 60).

1.1 Tipos de dispositivos

A utilização de *tablets*, PDA e *smartphones* contribuem para aumentar e qualificar a eficiência dos processos rotineiros. Os mesmos podem ser utilizados em várias tarefas, com coleta de informações, planilhas, calculadoras, automação, entre outras. Com essa

utilização diminuem-se as ocorrências de erros devido às falhas humanas no preenchimento de questionários, identificação de causas de eventos, erros em inspeções de campo e aumenta a disponibilidade de informações devido a sua mobilidade.

Os *tablets* e *smartphones* são vastamente utilizados, pois os mesmos apresentam um baixo custo de aquisição comercial o que os difere dos PDAs (*Personal Digital Assistant*) que em sua maioria são desenvolvidos especificamente para determinadas atividades elevando assim o seu curso, mas ambos têm ampla cobertura de serviços e uma grande potencialidade de utilização como dispositivo de aquisição de dados. Outra grande vantagem dos celulares e *smartphones* é permitir a sua utilização para comunicação de voz e dados. As recentes tecnologias desenvolvidas para esses dispositivos possuem telas coloridas com alta qualidade de imagem, porém com tamanho reduzido, o que muitas vezes limita o tipo de aplicação que pode ser suportada pelo dispositivo. (FERRARI, 2013).

Nas próximas sessões são apresentadas as principais características desses dispositivos moveis.

1.1.1 PDA (Personal Digital Assistant)

PDAs são dispositivos móveis com tela monocromática ou colorida e sistema de entrada de dados por tela sensível ao toque (*touch screen*). A maioria já possuem aplicativos como calendário, agenda, bloco de notas entre outros, inclusos juntamente com o sistema operacional, como o ilustrado na figura 1. Existem ferramentas comerciais e livres para desenvolvimento de aplicativos para estas plataformas. A interface física com outros equipamentos pode ser realizada através de comunicação serial, *bluetooth* e por alguns *slots* do tipo PCMCIA e *compact flash*. Existem comercialmente, interfaces com estes dispositivos para a utilização de transferência de dados via GPRS e também utilizando serviços da rede CDMA e *WI-FI*. Os dois principais sistemas operacionais para PDAs são o Palm OS da Palm Inc e o Pocket PC da Microsoft Inc. (NETO, 2005).



Figura 1 - PDA – Fonte: (ANALYSIS TECNOLOGIA, 2013)

1.1.2 Smartphones

Os *Smartphones* são telefones celulares com funções mais avançadas. O termo “*smart*” vem do inglês e significa “inteligente”, sendo assim “telefones inteligentes”, esses dispositivos permitem conexão com a internet através de meios sem fio (*wireless*), possuem a capacidade de executar software, executam em sua maioria um SO, possuem processamento gráfico e de informações entre outros recursos disponibilizados por eles, a figura 2 ilustra um desses dispositivos móveis. Atualmente a maior desvantagem é o tamanho da tela, a qual, geralmente, é menor que a de um *tablet* (OLIVEIRA, 2007).



Figura 2 - Smartphone- Fonte: (GSMARENA, 2013)

1.1.3 Tablets

Um *tablet* é um dispositivo pessoal em formato de prancheta que pode ser utilizado para acesso à Internet, organização pessoal, comunicação, acesso de informações, leitura de livros, jornais entre outros. Também pode ser utilizado para tirar fotos e gravar vídeos, além de possuir uma tela sensível ao toque (*touchscreen*) sem a necessidade da utilização de periféricos para navegar, por exemplo, mouse e teclado externos (EGGERS, 2012). A figura 3 - ilustra a representação de um *tablet*.



Figura 3 - Representação de Tablets – Fonte: (PITCHEngine, 2013)

1.2 Sistemas Operacionais para Dispositivos Móveis

Um sistema operacional nada mais é do que o principal software do dispositivo que é responsável por todo o funcionamento da máquina desde os demais softwares até o *hardware* instalado no mesmo. Todos os processos de um computador estão por de trás de uma programação complexa que comanda todas as funções que um utilizador impõe à máquina. O sistema operacional funciona com a inicialização de processos que este irá precisar para funcionar corretamente. Estes processos poderão ser arquivos que necessitem ser frequentemente atualizados, ou arquivos que processam dados úteis para o sistema (MAZIERO, 2013).

A seguir são apresentados alguns dos principais sistemas operacionais disponíveis em dispositivos móveis (*tablet / smartphone*).

1.2.1 Windows Phone

O sistema Windows Phone da Microsoft pode ser considerado o *MS Windows* para equipamentos móveis. Mas apesar da aparência semelhante com o *MS Windows* para os PC's, os sistemas são totalmente diferentes, e os equipamentos móveis só executam programas criados para a plataforma mobile, devido à compatibilidade com os recursos disponíveis (WINDOWS, 2013).

1.2.2 iPhone OS

O *iPhone OS* ou *Mac OS X iPhone* é o sistema operacional desenvolvido pela Apple Inc. para o *iPhone* e *iPod Touch*. Tal como o Mac OS X, da qual foi derivado, ele usa o sistema operacional da fundação Darwin. *iPhone OS* tem captação em quatro camadas: a camada Core OS, a camada *Core Services*, a camada média, e a camada *Cocoa Touch*.

Este sistema operacional não tinha um nome oficial até o lançamento da primeira versão beta do *iPhone SDK* em 6 de Março de 2008. Até então, a Apple comercializava e simplesmente declarava que o "*iPhone* usa OS X", uma referência ao sistema operacional para *desktop* ou notebooks da Apple, o Mac OS X (APPLE, 2013).

1.2.3 Android

O *Android* é um sistema operacional baseado em Linux, criado pela *Google* e direcionado a dispositivos móveis. Apesar de o sistema ser anunciado como um sistema da *Google* para dispositivos móveis, a responsabilidade do desenvolvimento do mesmo é da *Open Handset Alliance*, um conjunto de 47 empresas do ramo tecnológico e das telecomunicações.

A plataforma é suportada por vários modelos de dispositivos móveis, como *tablets* e de *smartphones*. Suporta uma grande variedade de tecnologias de conectividade incluindo *Bluetooth*, EDGE, 3G, e *Wi-Fi*.

O *Android* é totalmente capaz de fazer uso de câmaras de vídeo, *touchscreen*, GPS, acelerômetros, e aceleração de gráficos 3D melhorando a experiência do usuário ao utilizá-lo, a sua versão atual é a 4.2 e chamada de Jelly Bean (ANDROID, 2013).

1.3 Tecnologias e Meios de Comunicação Móvel

Uma arquitetura de redes de computadores define um conjunto de camadas e protocolos. Nos dispositivos móveis, a arquitetura de redes adotada é a TCP/IP. É natural que isso tenha acontecido já que o computador móvel deve ter acesso à rede física que usa a arquitetura TCP/IP. No entanto, os protocolos dessa arquitetura não foram projetados para esse ambiente que possui características particulares.

Computadores na arquitetura TCP/IP usada na internet possuem um endereço IP que determina o roteamento de pacotes a serem entregues a um destinatário. Por trás deste conceito está o fato que os computadores são fixos e o endereço determina a localização física de um computador em relação ao restante da rede. No entanto, no caso de computadores móveis, isto não é válido, já que a localização deles varia e o tipo de rede que utilizará também muda.

Se o endereço associado ao computador móvel permanecesse o mesmo, independente de sua localização, então o endereço não pode ser usado para rotear pacotes IP, já que pode não representar a localização atual de um computador móvel ou da rede que está sendo utilizada. Por outro lado, se um computador móvel possui um endereço que é em função de sua posição, então todas as outras entidades (computadores, aplicações, etc) em contato com um computador precisam ser informadas do novo endereço (ROMEIRO, Recife, 2005). Nas seguintes seções são abordados alguns dos principais meios de comunicação móvel.

1.3.1 *Wi-Fi (Wireless Fidelity)*

No início dos anos 90, já existia uma preocupação sobre as novas tecnologias que seriam utilizadas para a conexão de equipamentos em redes sem fio. Algumas empresas começaram a desenvolver um novo padrão para a transmissão wireless que se tornaria oficial em 1997, o padrão IEEE 802.11 (conhecido como *Wi-Fi - Wireless Fidelity*). As redes *Wi-Fi* utilizam a tecnologia de rádio no padrão IEEE 802.11 para conectar

computadores uns aos outros, à Internet e a outras redes cabeadas usando *Ethernet*. (FARIA, 2005).

O padrão de redes locais sem fio (802.11) define o protocolo e a compatibilidade de interconexão de equipamentos de comunicação de dados via ar, rádio ou infravermelho em uma rede local (LAN) usando o “*carrier sense multiple access protocol with collision avoidance*” (CSMA/CA) como mecanismo de compartilhamento do meio. O controle de acesso ao meio (MAC) suporta operações abaixo do controle do ponto de acesso assim como entre as estações independentes. O protocolo inclui autenticação, associação e reassociação de serviços, um procedimento opcional que cifra e decifra e gerenciamento de energia (JARDIM, 2007).

O padrão inclui a definição de base de informação gerenciável usando o *Abstract Syntax Notation 1* (ASN.1) e especifica o protocolo MAC formalmente, usando o *Specification and Description Language* (SDL) (JARDIM, 2007).

1.3.2 Bluetooth

Bluetooth é uma tecnologia para conexão sem fio do padrão IEE/802.15, que é de curta distância utilizada em dispositivos móveis como celulares, *palms*, fones de ouvido, microfones, computadores entre outros. A tecnologia desenvolvida inicialmente pela Ericsson (1944) com o objetivo de substituir os cabos que conectavam estes dispositivos ganhou o suporte da Intel, IBM, Toshiba, Nokia, Lucent, Motorola entre outras empresas que vieram a formar o *Bluetooth Special Interest Group* (SIG).

A tecnologia *Wireless Bluetooth* é um padrão de fato e uma especificação para enlaces entre dispositivos móveis, usando ondas de rádio de curto alcance. O SIG é um grupo industrial formado por empresas líderes em telecomunicações, computação e rede que possuem interesses no desenvolvimento desta tecnologia.

A proposta da tecnologia *Bluetooth* é habilitar os usuários para a conexão de uma grande variedade de dispositivos de computação e telecomunicações de maneira simples e fácil, sem a necessidade de cabos. Como a tecnologia é baseada em enlace via rádio, é fácil a transmissão rápida e segura de voz e dados na rede. A operação do *Bluetooth* é efetuada em uma banda de frequência entre 2.402 e 2.480 Ghz que está globalmente disponível e tem compatibilidade mundial, sendo portanto um padrão global para conectividade *wireless*. A tecnologia *Bluetooth* é muito resistente a dificuldades

eletromagnéticas, o que garante que as interferências sejam pequenas, porém a distância entre os dispositivos para estabelecer uma conexão ideal é pequena chegando apenas a 10 metros e no máximo 100 metros entre os dispositivos. (MANGANELLI e ROMANI, 2004).

1.3.3 EDGE

Enhanced Data rates for GSM Evolution (EDGE) ou *Enhanced GPRS* (EGPRS) é uma tecnologia digital para telefonia celular que permite melhorar a transmissão de dados e aumentar a confiabilidade da transmissão de dados. Embora seja tecnicamente uma tecnologia da 3ª Geração, geralmente é classificada como um padrão 2,75G, já que é uma melhoria feita nas redes 2,5G (GPRS) e não a criação de uma nova tecnologia (JARDIM, 2007).

Ela foi introduzida nas redes GSM por volta de 2003, inicialmente na América do Norte. E pode ser usada para qualquer troca de pacotes como uma conexão com a internet. Dados em alta-velocidade e serviços como *streaming* de vídeos, rádios ao vivo, transferência de arquivos são possíveis nesta tecnologia.

EDGE foi desenvolvida para capacitar a transmissão de uma grande quantidade de dados a altas taxas de velocidade (384 *kbit/s*). Ela usa o mesmo conceito da tecnologia TDMA, no que se referem à estrutura dos quadros, canais lógicos e largura de banda (JARDIM, 2007).

É uma atualização de rede eficiente e relativamente simples para a maioria de operadoras GSM. As implementações normalmente requerem apenas softwares e cartões para canais adicionais na infraestrutura GSM/GPRS existente. Esse formato reduz o custo de implementação, permitindo uma política de preços mais competitiva para os serviços.

A EDGE não requer a aquisição de espectro adicional pelas operadoras. Em vez disso, a tecnologia pode ser implementada nas bandas mais comuns, entre os quais 850, 900, 1800 e 1900 *MHz*. A oportunidade de implementar a EDGE no espectro existente significa um lançamento mais rápido de serviços 3G, num número maior de mercados e com um custo menor comparado com tecnologias que requerem espectro adicional (BALSEMÃO, 2008).

1.3.4 3G

A tecnologia 3G é a terceira geração de padrão IEEE 802.20 e tecnologias de telefonia móvel, que veio por sua vez suceder o 2G. É baseado na família de normas da União Internacional de Telecomunicações (UIT), no âmbito do Programa Internacional de Telecomunicações Móveis – 2000 (IMT-2000).

Esta tecnologia permite às operadoras da rede oferecer a seus usuários uma variedade de avançados serviços, já que possuem uma capacidade de rede maior por causa de uma melhora na eficiência espectral. Entre os serviços, há a telefonia por voz e a transmissão de dados a longas distâncias, tudo em um ambiente móvel. Normalmente, são fornecidos serviços com taxas de 5 a 10 *Megabits* por segundo.

Ao contrário das redes definidas pelo padrão IEEE 802.11, as redes 3G permitem telefonia móvel de longo alcance e evoluiu para incorporar redes de acesso à Internet em alta velocidade e Vídeo-telefonia. As redes IEEE 802.11 (mais conhecidas como *Wi-Fi* ou *WLAN*) são de curto alcance e ampla largura de banda e foram originalmente desenvolvidas para redes de dados, além de não possuírem muita preocupação quanto ao consumo de energia, aspecto fundamental para aparelhos que possuem pouca carga de bateria (BALSEMÃO, 2008).

1.4 Transferência de dados entre dispositivo e servidores

Os Web Services surgem com a ideia de solucionar os problemas de incompatibilidade entre os protocolos de comunicação, uso de portas e formatos binários rejeitados por firewalls, dentre outros problemas apresentados em softwares projetados com a arquitetura em camada. Web Services são sistemas de software desenhado para permitir a comunicação de máquina para máquina em uma rede de computadores. É uma solução utilizada na integração de sistemas e na comunicação entre tecnologias diferentes. Eles permitem a descrição de um serviço em uma determinada máquina, feita em uma linguagem baseada em XML (usualmente *WSDL-Web Service Description Language*) (CARISSIMI, ROCHOL e GRAVILLE, 2009, p. 335). Um exemplo de código de WSDL é ilustrado na figura 4.

```

<?xml version="1.0" encoding="UTF-8"?><!-- Published by JAX-WS RI at http://jax-ws.dev.java.net. RI's
<ns1:Policy xmlns:ns1="http://schemas.xmlsoap.org/ws/2004/09/policy" wsu:Id="WSCobaiaPortBindingPolic
<ns1:ExactlyOne>
<ns1:All></ns1:All>
</ns1:ExactlyOne>
</ns1:Policy>
<types>
<xsd:schema>
<xsd:import namespace="http://Servico/" schemaLocation="http://localhost:8080/coba1/WSCobaiaService?x
</xsd:schema>
</types>
<message name="ConsultaProdutoWS">
<part name="parameters" element="tns:ConsultaProdutoWS"></part>
</message>
<message name="ConsultaProdutoWSResponse">
<part name="parameters" element="tns:ConsultaProdutoWSResponse"></part>
</message>
<message name="InsereProdutoWS">
<part name="parameters" element="tns:InsereProdutoWS"></part>
</message>
<message name="InsereProdutoWSResponse">
<part name="parameters" element="tns:InsereProdutoWSResponse"></part>
</message>
<message name="TestaConexao">
<part name="parameters" element="tns:TestaConexao"></part>
</message>
<message name="TestaConexaoResponse">
<part name="parameters" element="tns:TestaConexaoResponse"></part>

```

Figura 4 - Código de WSDL - Fonte Arquivos Pessoais

Tal descrição é feita usando-se as regras do padrão SOAP (*Simple object access protocol*) para que, seguido o mesmo padrão, ferramentas do lado do cliente consigam automaticamente gerar códigos em linguagens como Java, para que do ponto de vista do desenvolvedor, ele esteja apenas acessando códigos em sua linguagem favorita, sem se importar com os detalhes como tráfego de rede, análise sintática do XML para consumo, entre outros. (ROCHA, 2009).

Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Os Web Services são componentes que permitem às aplicações enviar e receber dados em formato XML. Cada aplicação pode ter a sua própria "linguagem", que é traduzida para uma linguagem universal, o formato XML.

Existe uma grande motivação sobre a tecnologia Web Service, pois possibilita que diferentes aplicações comuniquem entre si e utilizem recursos diferentes. Utilizando essa tecnologia, uma aplicação pode invocar outra para efetuar tarefas simples ou complexas mesmo que as duas aplicações estejam em diferentes sistemas e escritas em linguagens diferentes. Por outras palavras, os Web Services fazem com que os seus recursos estejam disponíveis para que qualquer aplicação cliente possa operar e extrair os recursos fornecidos.

Um dos motivos que tornam os Web Services atrativos é o fato deste modelo ser baseado em tecnologias padrões. Eles são utilizados para disponibilizar serviços

interativos na Web, podendo ser acessados por outras aplicações usando, por exemplo, o protocolo SOAP (*Simple Object Access Protocol*). Segundo o W3C (*World Wide Web Consortium*) um Web Service define-se como: um sistema de software projetado para suportar a interoperabilidade entre máquinas sobre rede. Tem uma relação descritiva num formato *machine-processable*, especificamente WSDL (*Webservice Description Language*). A figura 5 descreve um exemplo de Web Service. (PAULON, 2004).



Figura 5 - Exemplo Web Service – Fonte: Arquivos Pessoais

1.5 Aplicações para Dispositivos Móveis (*Smartphone e Tablets*)

Existem ferramentas e *frameworks* construídos para ajudar no desenvolvimento de aplicativos que possam ter a sua implantação em múltiplas plataformas de sistema operacional móvel e *web browsers* como: *PhoneGap*, *Sencha Touch 2*, *Appcelerator Titanium* entre outros, que usam os conceitos de aplicação Web. Também existem tipos de aplicações mais específicas, construídas com ferramentas disponibilizadas pelo desenvolvedor do sistema operacional (SO) nativo, chamadas de Aplicações Nativas.

1.5.1 Aplicativo Nativo

Cada plataforma de desenvolvimento de aplicativos móveis (*iOS*, *Android*, *Windows Mobile*) requer seu próprio processo de desenvolvimento, onde cada plataforma tem a sua própria linguagem de programação nativa. O *Android* utiliza o Java, o *iOS* utiliza Objective-C, e o *Windows Phone* utiliza C++ essas são as principais linguagens de programação utilizadas pelas plataformas, porém ambas possuem compatibilidade com mais linguagens de programação disponíveis como C#, C, VB# (Visual Basic) ou Ruby. (LEE, SCHNEIDER e SCHELL, 2005).

Os Kits de desenvolvimento de software padronizados (SDKs), são ferramentas de desenvolvimento e de elementos de interface de usuário comuns (botões, campos de entrada de texto, entre outros) muitas vezes fornecidos pelo fabricante da plataforma.

Um aplicativo nativo tem em seu desenvolvimento a determinação à qual dispositivo móvel (*Smartphones, Tablet*, entre outros) ele será compatível na maioria das vezes especificando sistema operacional e modelo. São instalados diretamente no dispositivo.

1.5.2 Aplicações Web

As aplicações web são interpretadas pelo navegador do dispositivo móvel e executadas na web. Cada aplicação pode ter suas próprias características e peculiaridades, são desenvolvidas normalmente em HTML5 (*HyperText Markup Language 5*), CSS3 (*Cascading Style Sheets 3*), JavaScript e do lado do servidor linguagens ou *frameworks* de aplicação web de escolha do desenvolvedor como *PHP, Python, Rails, C#* entre outras. (INTEL, 2013).

Não há kits de desenvolvimento de software padrão (SDKs) que os desenvolvedores são obrigados a usar para fazer uma aplicação móvel deixando-os livre para escolher as ferramentas que pretendem utilizar.

Aplicativos desenvolvidos para a plataforma web utilizam a Internet para se comunicar e apresentar as funcionalidades específicas desenvolvidas para os dispositivos móveis. A aplicação é acessada através do navegador do dispositivo móvel e como utilizam a internet para isso não há a necessidade de serem instalados no dispositivo. (SUMMERFIELD, 2013).

1.5.3 Método de instalação e execução das aplicações

As aplicações nativas são transferidas para o dispositivo móvel através de um mercado de aplicações disponibilizado pelo desenvolvedor do sistema operacional, os usuários adquirem os aplicativos desses mercados como a *App Store* da Apple e o *Google Play* da Google que contém uma grande variedade de aplicações disponíveis para serem carregados nos dispositivos. São instalados e executados como um aplicativo independente sem a necessidade de um navegador, os usuários baixam e instalam manualmente ou habilitam para atualizar automaticamente os aplicativos podendo assim optar por ignorar uma atualização e executar uma versão diferente dos demais usuários.

Uma Aplicação Web é acessada através de um navegador presente no dispositivo móvel, por causa disso não há necessidade de ser instalada. As atualizações são feitas através do servidor web sem precisar com que o usuário intervenha. Para Aplicações Web não há loja de aplicativos, assim pode ser mais difícil para os usuários encontrar o aplicativo que deseje.

1.5.4 Vantagens das aplicações móveis

A usabilidade e interface que as aplicações oferecem ao usuário praticamente não se diferem muito e por isso muitas empresas optam por desenvolver tanto aplicativo nativos como aplicativos web para dispositivos móveis. A figura 6 demonstra lado-a-lado o aplicativo nativo e web do Facebook.

Pode-se notar em termos de interface e comandos contidos nos dois exemplos que não há muitas diferenças dando ao usuário a mesma experiência de uso.



Figura 6 - Exemplo de Aplicação Nativa e Aplicação WEB – Fonte: (SIXREVISIONS, 2012).

As aplicações nativas possuem algumas vantagens sobre as webs aplicações, pois as mesmas trabalham melhor com os recursos disponíveis no dispositivo como câmera, acelerômetro, uma vez que a linguagem usada habilita acesso ao *hardware* do dispositivo.

As aplicações nativas executam com melhor desempenho que as aplicações web, pois tem como aliado o desenvolvimento específico para a plataforma de execução,

porém as webs aplicações utilizam como base códigos de programação que executam na maioria das plataformas, além de poder aplicar o design responsivo e a possibilidade de executar a aplicação off-line de acordo com Mark (2013), armazenando dados na memória do dispositivo e ao estabelecer uma comunicação esses dados são transmitidos, evitando assim a inatividade da aplicação.

1.5.5 Desvantagens das aplicações

As aplicações nativas na maioria das vezes tem seu desenvolvimento com custo mais elevado, pois se for disponibilizada para múltiplas plataformas, requer desenvolvimento e atualização diferenciados para cada plataforma, além disso, os usuários poderem estar utilizando versões diferentes, fazendo assim com que haja dificuldade em manter e fornecer suporte para a aplicação.

Os aplicativos web não podem acessar todos os recursos do dispositivo, apesar de constantes esforços para que isso deixe de existir. O suporte a vários navegadores de celular pode resultar em maiores custos de desenvolvimento e manutenção. Os usuários podem estar em diferentes navegadores móveis e isso pode fazer com que a se torne difícil manter e fornecer suporte para a aplicação. Além da dificuldade que o usuário pode encontrar pela falta de uma loja de aplicativos centralizada. Observa-se no entanto, que algumas empresas como a Apple já estão se preocupando com isso e começam a disponibilizar mecanismos para encontrar aplicações Web mais facilmente.

1.5.6 Ferramentas para o desenvolvimento de aplicações móveis

Existe um grande gama de ferramentas e linguagens de programação disponíveis que podem ser utilizadas para o desenvolvimento de aplicativos móveis, muitas delas são gratuitas, porém existem ferramentas que para a utilização é necessário que seja efetuada a compra ou licenciamento da mesma. Assim basta o desenvolvedor definir o tipo de aplicação que melhor se adapte as características do projeto, podendo ser ela Web ou Nativa.

As ferramentas mais conhecidas para o desenvolvimento de aplicações nativas são as que a própria plataforma disponibiliza, já as aplicações web utilizam HTML5, CSS3,

JavaScript do lado do cliente e do lado do servidor linguagens ou *frameworks* de aplicação web como *PHP, Rails, Python e C#*.

1.5.6.1 Linguagens de desenvolvimento

As aplicações nativas para Windows Phone são desenvolvidas utilizando o C#, C++, XAML ou VB# (*Visual Basic*) como linguagens de programação e também o kit de desenvolvimento do Windows Phone disponibilizado pela Microsoft, que tem como base o Silverlight que é uma plataforma de desenvolvimento de interatividade e interfaces mais atraentes, que tem como linguagem de interface XAML sendo uma linguagem declarativa baseada no XML (*eXtensible Markup Language*). O *Android* utiliza o Java como sua principal linguagem de programação contida no seu SDK (*Software Development Kit*), mas também faz uso de outras linguagens como o C e C++ para o desenvolvimento de aplicações utilizando o *Android NDK* que é uma alternativa para o desenvolvimento de aplicações, de acordo com o próprio site de ferramentas de desenvolvimento da empresa (DEVELOPER ANDROID, 2013). Já o iOS utiliza apenas o Objective-C e há especulações de que em breve suporte C e C++ como as demais plataformas citadas acima (DEVELOPER APPLE, 2013)

Ambas as plataformas suportam e reconhecem as linguagens utilizadas para Aplicações Web como o HTML5, CSS3, *JavaScript, PHP, Rails, Python e C#* entre outras, dando assim alternativas para os desenvolvedores na criação de aplicações. A seguir são apresentados algumas das linguagens de programação, *frameworks* e algumas estruturas de programação disponíveis para os desenvolvedores de aplicações.

- **C#** - É uma linguagem de programação orientada a objetos, criada pela microsoft para a plataforma .NET que auxilia no desenvolvimento de aplicações.
- **C++** - É uma linguagem de programação que tem como base o C e utiliza o paradigma de programação orientada a objetos, utiliza também conceitos básicos de herança, argumentos de função padrão, além de todas as características da linguagem C e outras mais adicionadas (STROUSTRUP, 2013)
- **VB#** - É uma linguagem de programação projetada para criar aplicações tipadas e orientadas a objeto. Permitindo que os desenvolvedores utilizem os objetivo desenvolvidos para Windows, Web e dispositivos móveis (MICROSOFT, 2013).
- **Java** - É uma linguagem de programação e uma plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995 sua utilização é

gratuita, operando sobre *Java Runtime Environment* (JRE) que consiste no Java Virtual Machine (JVM), com classes centrais e bibliotecas de suporte da plataforma Java. O JRE representa a parte de tempo de execução do software Java (JAVA, 2013).

- **Objective-C** – É um *framework* desenvolvido pela Apple para criação de aplicações para o iOS, onde tem como base a estrutura da linguagem C.

XML – É uma linguagem de marcação em formato simples de texto, muito flexível derivado do SGML (ISO 8879). Originalmente concebido para enfrentar os desafios da publicação eletrônica em grande escala e de diversas fontes de dados padronizando assim a formatação dos mesmos (W3C, 2013a).

- **XAML** – É uma linguagem de interface utilizada para desenvolver elementos de interface do usuário para aplicativos do Windows Phone.

1.5.6.2 Linguagens e framework de desenvolvimento web

O desenvolvimento para WEB possibilita ao programador escolher dentre as várias linguagens e ferramentas que existem para serem utilizadas, além de facilitar a adaptação de interfaces de diferentes modelos reais para varias plataformas de SO móveis, podendo ser eles os de gerenciamento de software amplamente usado nas empresas pra melhor conceber seus sistemas. Algumas linguagens e *framework* disponíveis são:

PHP – O PHP (*Hypertext Preprocessor*) é uma linguagem de *scripting* de uso geral amplamente utilizada, onde é incorporado ao HTML sendo executada no servidor, os resultados gerados no servidor são enviados ao solicitante, montando assim a página em HTML (NIEDERAUER, 2011).

JavaScript - É uma linguagem de script criada pela Netscape em 1995 lançada junto com o navegador Netscape. Apesar de possuir o nome semelhante ao Java ela não é derivada, o JavaScript é um linguagem interpretada e flexível (DAMIANI, 2001).

PYTHON – Pode ser definida como uma linguagem de script orientada a objetos, comumente aplicada em funções de script (LUTZ e ASCHER, 2007, p. 31).

CSS3 - é a evolução do CSS (*Cascading Style Sheets*), com a adição de novos elementos, tendo como objetivo a formatação das informações de uma página HTML (ALVAREZ, 2013)

HTML5 – O HTML5 é conjunto das novas versões em desenvolvimento de Linguagem de Marcação de Hipertexto (*Hyper Text Markup Language* – HTML) e das

Folhas de Estilo em Cascata (*Cascading Style Sheets – CSS*), e de novas Interfaces para a Programação de Aplicações (*Application Program Interface – API*) para *ECMAScript*, que combinadas permitem a criação de interfaces web mais interativas (W3C, 2013b)

RAILS – É um *framework* de desenvolvimento gratuito e de código aberto que utiliza a linguagem de programação Ruby, esta que é interpretada e orientada a objetos, tem sua programação dinâmica reflexiva, permitindo alterações na sua estrutura durante a execução (URUBATAN, 2012).

PhoneGap – É um *framework* de código aberto e livre que utiliza HTML5, CSS3 e JavaScript, permitindo criar aplicativos móveis que usam APIs web, padronizando o desenvolvimento para qualquer plataformas móvel de acordo com o site da empresa (PHONEGAP, 2013). A figura 7 representa o *framework*.

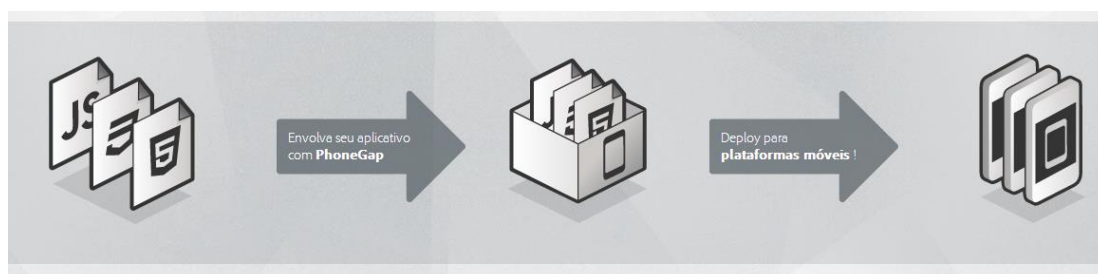


Figura 7 - Estrutura do PhoneGap – Fonte: (PHONEGAP, 2013).

A figura 8 ilustra alguns recursos nativos que podem ser utilizados pelo *framework* e também os SO que são suportados sendo eles *Android*, *iOS(iPhone)*, *Blackberry*, *Symbian*, *Windows Phone 7*, *WebOs* e *Bada*.

	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 5.x	Blackberry OS 6.0+	WebOS	Windows Phone 7 + 8	Symbian	Bada
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	X	✓	✓	X	✓
Contacts	✓	✓	✓	✓	✓	X	✓	✓	✓
File	✓	✓	✓	✓	✓	X	✓	X	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	X	X	✓	X	X
Network	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓	X

Figura 8 - Indicação de recursos nativos do SO suportadas pelo framework PhoneGap. Fonte: (PHONEGAP, 2013)

Sencha Touch 2 - É um *framework* que utiliza como base o HTML5 e CSS3, também possui um sistema de MVC (*Model view controller*) para auxiliar no desenvolvimento de aplicações rápidas que trabalham em iOS, *Android*, BlackBerry, Windows Phone entre outros, sua versão completa não é gratuita de acordo com o site da empresa.

jQuery Mobile – é um *framework* de sistema unificado, baseado em HTML5, que possui interface de usuário para ser utilizado em todas as plataformas de dispositivos móveis populares, constituído com jQuery e jQuery UI, com codificação JavaScript mais leve, possui um design flexível e pode ser facilmente personalizável.

Appcelerator Titanium - É um ambiente de desenvolvimento aberto e extensível para a criação de aplicações nativas em diferentes dispositivos móveis e sistemas operacionais, incluindo iOS, *Android*, Windows Phone e BlackBerry, bem como desenvolvimento híbridos e HTML5. Ele possui SDK e é compatível com sistema operacional móvel, possui uma IDE (*Integrated Development Environment*) baseado em Eclipse, ele é um *framework* MVC e possui o *Cloud Services* para construção de aplicativos moveis que se conectam.

2 ADAPTAÇÃO DE INTERFACES DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS

Existem muitas ferramentas e tecnologias para a adaptação de interfaces web e *desktops* para serem utilizadas e visualizadas em dispositivos móveis muitas dessas foram citadas no capítulo anterior, porém será apresentado mais detalhadamente a utilização do HTML5 com recurso de web design responsivo, aliando o CSS3 e JavaScript para melhorar a apresentação da interface.

De acordo ZEMEL (2012), o Web design responsivo é aquele que responde a qualquer dispositivo ou resolução de tela como está ilustrado na figura 9. Uma aplicação com layout responsivo pode se bem visualizada em qualquer um dos dispositivos, por exemplo, uma TV, *Tablet*, *Smartphone*, *desktop* ou até mesmo em novas geladeiras que possuem tela com conexão a internet. As tecnologias principais envolvidas são o layout fluido, imagens e recursos flexíveis e a utilização de media queries.



Figura 9 - Representação do web design responsivo – Fonte: (ZEMEL, 2012).

2.1 Layout fluido

O layout fluido é definido pela não utilização de medidas absolutas no CSS, pois ao especificar tamanhos, espaçamentos, margens, *padding*s ou qualquer medida no site

se impede a adaptação no mesmo a outros tipos de tela (ZEMEL, 2012), assim a melhor forma de especificar esses valores é a utilização de porcentagens e *em*¹.

2.2 Imagens e Recursos flexíveis

Para a utilização de imagens e recursos flexíveis no desenvolvimento de web design responsivos é necessário à utilização do CSS para determinar efeitos sobre os recursos utilizados em um site. (ZEMEL, 2012).

2.3 Media queries

Na versão anterior do CSS se utilizava uma função denominada *media type* que era utilizada para reconhecer um determinado dispositivo, sendo que possuía alguns tipos definidos, que estão representados no quadro 1 abaixo.

Quadro 1 - Especificação dos Media Type

Type	Descrição
All	A folha de estilo serve para todos os dispositivos
Braile	Para dar retorno quando se utiliza algum dispositivo tátil
Embossed	Impressoras em braille paginadas
Handheld	Dispositivos móveis
print	Para material paginado e visualização na tela no modo visualização de impressão
projection	Destinado para apresentação em projetores
screen	Definição para telas de computadores
speech	Para sintetizadores de voz
tty	Teleimpressores e terminais
tv	Para aparelhos do tipo TV

Fonte - (ZEMEL, 2012).

Com a evolução das tecnologias de desenvolvimento web, as diferenças entre dispositivos começaram a desaparecer, pois os dispositivos passaram a ter telas de boa qualidade e seus navegadores renderizam as páginas como um navegador normal *desktop*, um desses dispositivos é o iPhone que não há necessidade de preparar um

¹ *em* – é uma unidade escalável e os elementos *1em* são iguais ao tamanho atual da fonte do elemento pai.

layout e um CSS com o *media type handheld* para ele. (ZEMEL, 2012). Para resolver essas diferenças de tipos de dispositivos a versão do CSS3 utiliza media queries.

De acordo com Mazza (MAZZA, 2012, p. 177), as media queries são utilizadas para definir o *device* (dispositivo) e o CSS a ser utilizado, por exemplo, ao visualizar um site em um dispositivo móvel com tela pequena de 320px é carregado o CSS que tem adaptação definida para o mesmo, para isso é criado um CSS com seu estilo específico para os aparelhos que possuam essa resolução. A figura 10 ilustra como é feita a declaração na folha de estilo dos parâmetros *media queries*.

```

1  @media screen and (min-width:320px) {
2  body{
3      font-size:80%;
4  }
5  }
6  @media screen and (min-width:480px) {
7  body{
8      font-size:90%;
9  }
10 }
```

Figura 10 - Representação de código media queries – Fonte: Arquivos Pessoais.

Também se podem criar estilos específicos separados em mais arquivos de estilo e fazer a sua declaração específica no web site.

As media queries já são uma recomendação da W3C desde 2012, cujo tornou possível a evolução na especificação de estilos para web sites. Existe uma gama de parâmetros de media queries para serem utilizados, além de *media screen min-width* outros tipos são apresentados no quadro 2.

Quadro 2 - Tipos de media queries

Media Query	Descrição de Retorno
width	Retorna a largura da janela do navegador.
height	Retorna a altura da janela do navegador.
device-width	Retorna a largura da tela do dispositivo.
device-height	Retorna a altura da tela do dispositivo.
orientation	Verifica se a orientação é “em pé” (<i>portrait</i>) ou “deitada” (<i>landscape</i>). Se a tela for quadrada a orientação <i>portrait</i> prevalecerá.
aspect-ratio	Retorna a proporção da janela do navegador.
device-aspect-ratio	Retorna a proporção da tela do dispositivo
color	Indica o numero de bits por componente de cor do dispositivo
color-index	Descreve o numero de cores do dispositivo
monochrome	Indica o numero de bits por pixel em dispositivos monocromáticos
resolution	Trata da resolução do dispositivo, pode ser pontos por polegada(dpi) ou por centimetro(dpcm)

Scan	Trata do tipo de escaneamento que um dispositivo do tipo "TV" pode fazer.
Grid	Determina se o dispositivo é de grade ou de mapa de bits(bitmap)

Fonte - (ZEMEL, 2012).

2.4 jQuery Mobile

Em novembro de 2011 foi lançado uma versão jQuery Mobile, visando atender o mercado web móvel. Muitos navegadores dos dispositivos possuem suporte ao jQuery Mobile, alguns dispositivos móveis possuem suporte a sua totalidade de elementos e outros executam recursos mais brandos.

O jQuery Mobile tem suporte a grande maioria de *desktops* modernos, *smartphones e tablets*. É usado um nivelamento de suporte à plataforma jQuery Mobile de 3 níveis sendo que A (suporte total), B (suporte total menos Ajax), C (HTML básico), esse nivelamento é representado no quadro 3 que indica alguns navegadores e SO que possuem suporte aos determinados níveis.

Quadro 3 - 3 níveis de suporte do jQuery Mobile

A	B	C
Apple iOS 3.2*-6.1	Blackberry 5.0*	Internet Explorer 6 and older
<i>Android</i> 2.1-2.3	Opera Mini 7	iOS 3.x and older
<i>Android</i> 4.1 (Jelly Bean)	Nokia Symbian^3	Blackberry 4.x
<i>Android</i> 4.1 (Jelly Bean)	Internet Explorer 7	Windows Mobile
Windows Phone 7.5-7.8		
Blackberry 6-10		
Palm WebOS (1.4-3.0)		
Firefox Mobile 18		
Chrome for <i>Android</i> 18		
Opera Mobile 11.5-12		
Chrome Desktop 16-24		
Safari Desktop 5-6		
Firefox Desktop 10-18		
Internet Explorer 8-10		
Opera Desktop 10-12		

Fonte - *Mobile Graded Browser Support* (JQUERY MOBILE, 2013).

Sendo que a fidelidade visual da experiência e suavidade de transições de páginas é altamente dependente dos recursos de renderização de CSS do dispositivo e plataforma. Desta forma, nem todos os elementos terão um grau de pixel perfeito, pois

depende dos recursos disponíveis no dispositivo que está executando a renderização do web site ou aplicação.

2.5 HTML 5

Com a nova versão do HTML novos recursos foram apresentados para o desenvolvimento web, como novos elementos, novas funcionalidades entre outras diversas novidades que possibilitam novas experiências de interação com web sites (MAZZA, 2012). Alguns desses elementos adicionados na versão HTML5 estão apresentados no quadro 4 abaixo.

Quadro 4 - Novos Elementos que fazem parte do HTML5

TAG	Descrição
<article>	Especifica um artigo
<aside>	Especifica o conteúdo relacionado com o artigo dentro de um artigo ou página web
<audio>	Especifica um streaming de áudio em um documento HTML
<canvas>	Especifica a criação de gráficos em uma página web
<command>	Especifica um comando que o usuário pode invocar
<datalist>	Especifica um "autocomplete" em formulários
<datatemplate>	Especifica um template de dados
<details>	Especifica informações adicionais que o usuário pode visualizar ou esconder sob demanda
<device>	Representa um dispositivo seletor, que permite ao usuário dar o acesso à página para um dispositivo, por exemplo, uma câmera de vídeo.
<embed>	Especifica uma aplicação externa com conteúdo interativo
<eventsources>	Especifica um destino para eventos enviados por um servidor
<figure>	Especifica ilustrações, imagens, fotos, associado juntamente com alguma legenda
<footer>	Especifica um rodapé para a seção de uma página
<header>	Especifica um cabeçalho para uma seção de uma página
<hgroup>	Especifica um container para elementos de título do nível h1 ao h6
<keygen>	Especifica um par de chaves público-privadas
<mark>	Especifica um texto destacado para fins de referência
<meter>	Especifica medição dentro de um intervalo predefinido
<nav>	Especifica uma lista de links para navegação
<output>	Especifica o resultado de um cálculo, por exemplo, o efetuado por algum script
<progress>	Especifica o progresso de uma tarefa de qualquer tipo
<ruby>	Especifica texto Ruby (tipografia utilizada no Sudeste Asiático)
<rule>	Especifica as regras para a atualização de um <datatemplate>
<rp>	Especifica anotações ruby em benefício dos browsers que não suportam ruby
<rt>	Especifica um componente do texto ruby em uma anotação ruby
<section>	Especifica uma seção dentro de um artigo
<source>	Especifica múltiplos recursos multimídia em uma página
<time>	Especifica data e hora
<video>	Especifica um streaming de vídeo em um documento html

Fonte - HTML5 - Curso W3C Escritório Brasil (W3C, 2013b)

Uma das mudanças significativas que o HTML5 introduziu é a simplicidade na apresentação de elementos, pois diversos pontos se tornam opcionais ou desnecessários na construção de um web site ou aplicação web. Um exemplo dessa simplicidade é apresentado na figura 11, sendo que as primeiras 5 linhas de código são elaboradas utilizando o HTML5 e as linhas seguintes são a mesma representação, porém utilizando o HTML tradicional.

```
1 <!--HTML5 -->
2 <!doctype html>
3 <html>
4 <head>
5 <meta charset="utf-8">
6
7
1 |<!--HTML versões anteriores -->
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
7
```

Figura 11 - Código *doctype* – Fonte: Arquivos Pessoais

Com isso torna-se mais fácil e simples o desenvolvimento de aplicações ou sites web, ampliando possibilidades para melhorar a adaptação de variados tipos de conteúdo, até mesmos um modelo de engenharia de software.

3 METODOLOGIAS DE GERENCIAMENTO DE PROJETO DE SOFTWARE

A engenharia de software nada mais é do que a criação e utilização de metodologias e princípios existentes a fim de obter um software de qualidade, que seja confiável e que trabalhe eficientemente exercendo o seu propósito (SBROCCO e MACEDO, 2012). No entanto, é válido considerar que a regularidade do bom funcionamento não é uma constante pelo fato do uso de metodologias de desenvolvimento inadequadas ao propósito do projeto, desta forma podem-se enfrentar vários desafios dificultando o decorrer dos processos (SOMMERVILLE, 2003).

Existem várias metodologias de desenvolvimento de software, que podem ser analisadas e escolhidas de acordo com a necessidade do projeto. As metodologias ágeis e as tradicionais, apesar de diferentes, comportam processos e atividades da engenharia de software que são semelhantes, como a especificação de software, projeto e implantação, validação e evolução.

O aspecto principal para o gerenciamento de um projeto é a organização, planejada e supervisionada dos controles e procedimentos contidos no projeto, sendo um processo contínuo para que possa atingir o objetivo esperado (NBR10006, 2013). Para isso existem metodologias e normas que podem auxiliar para um gerenciamento mais efetivo, podendo ser elas metodologias ágeis e metodologias tradicionais.

3.1 Metodologias Tradicionais

Metodologias tradicionais fazem uso de um grande número de documentação sendo que cada etapa de um processo é documentada e anexada ao projeto principal. Vários modelos baseados na metodologia tradicional estão disponíveis, tais como o modelo clássico e PMBOK(*Project Management Body of Knowledge*) (SBROCCO e MACEDO, 2012).

3.1.1 Modelo Clássico

O modelo clássico também é conhecido como modelo em cascata, foi o primeiro processo publicado de desenvolvimento de software, modelo esse de fácil entendimento, pois é baseado em uma sequência de etapas, que são a viabilidade, análise, projeto, implementação, testes, implantação, operação e manutenção (SBROCCO e MACEDO, 2012).

Nesse modelo cada etapa é documentada e somente após aprovação da documentação padrão a próxima etapa é iniciada como apresenta a Figura 12.

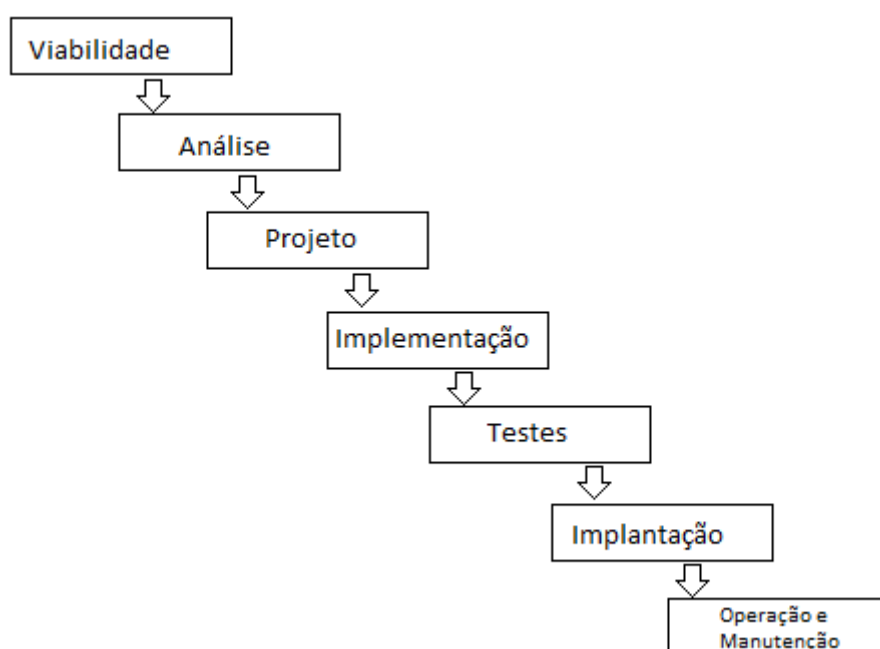


Figura 12 - Representação do Modelo Clássico – Fonte: Arquivos Pessoais, Baseado em (SBROCCO e MACEDO, 2012)

3.1.2 PMBOK (Project Management Body of Knowledge)

O PMBOK é um guia de conhecimentos e melhores práticas existente para os profissionais de gerência de projetos. Ele é genérico e serve para todas as áreas do conhecimento, independentemente se referir-se a um projeto de construção civil ou um de desenvolvimento de software. No guia constam práticas tradicionais de gestão de projetos, como também práticas inovadoras e mais avançadas (SBROCCO e MACEDO, 2012).

O PMBOK também tem como objetivo a padronização de termos utilizados na área de gerência de projetos. Ele é mantido pelo PMI (*Project Management Institute*), o qual consiste em uma associação sem fins lucrativos (DUNCAN, 2012).

De acordo com o PMBOK os processos gerenciais se resumem a, processo de iniciação, processo de planejamento, processo de execução, processo de controle e processo de finalização.

3.2 Metodologias Ágeis

No ano 2001 uma reunião muito importante em uma estação de esqui, em Nevada no estado norte-americano de Utah, marcou definitivamente o surgimento e a propagação de paradigmas de desenvolvimento de software ágeis. Nesta reunião estavam profissionais que já vinham trabalhando com metodologias como *Scrum*, XP (*Extreme Programming*), DSM (*Adaptative Software Development*), Cristal, entre outras, que até então eram conhecidas como metodologias leves (SBROCCO e MACEDO, 2012). O objetivo da reunião era discutir formas de melhorar o desempenho dos projetos. Após discussões, redigiram um documento que foi chamado de “Manifesto Ágil” (AGILE MANIFESTO, 2013), no conteúdo deste manifesto contém alguns conceitos chaves destacados à seguir:

- Indivíduos e interações mais que processos e ferramentas.
- Software em funcionamento mais que documentação abrangente.
- Colaboração com o cliente mais que negociação de contratos.
- Responder a mudanças mais que seguir um plano.

Os métodos ágeis são menos centrados na documentação voltando mais a codificação, também são metodologias adaptativas e não predeterminantes como as tradicionais, tirando assim todo o período de tempo que é dado para planejar (SBROCCO e MACEDO, 2012). No uso de metodologias ágeis as mudanças são aceitas ao longo do desenvolvimento de maneira natural e sem resistência, com custos mais aceitáveis perante as metodologias tradicionais.

De acordo com dados da *VersionOne* (VERSIONONE, 2012) empresa de estudo e desenvolvimento de ferramentas ágeis, apresentados no gráfico 1 e enfatizados em seu encontro anual em 2012, sendo que expõe algumas estáticas de várias empresas que adotaram algum tipo de metodologia ágil, e o resultado foi bem marcante, pois todas

apresentaram melhoras, sendo no tempo de resposta ou lançamento de um produto novo ao mercado, o aumento de produtividade por parte da equipe, melhor qualidade e redução de falhas e outras melhorias na adoção do desenvolvimento utilizando métodos ágeis.

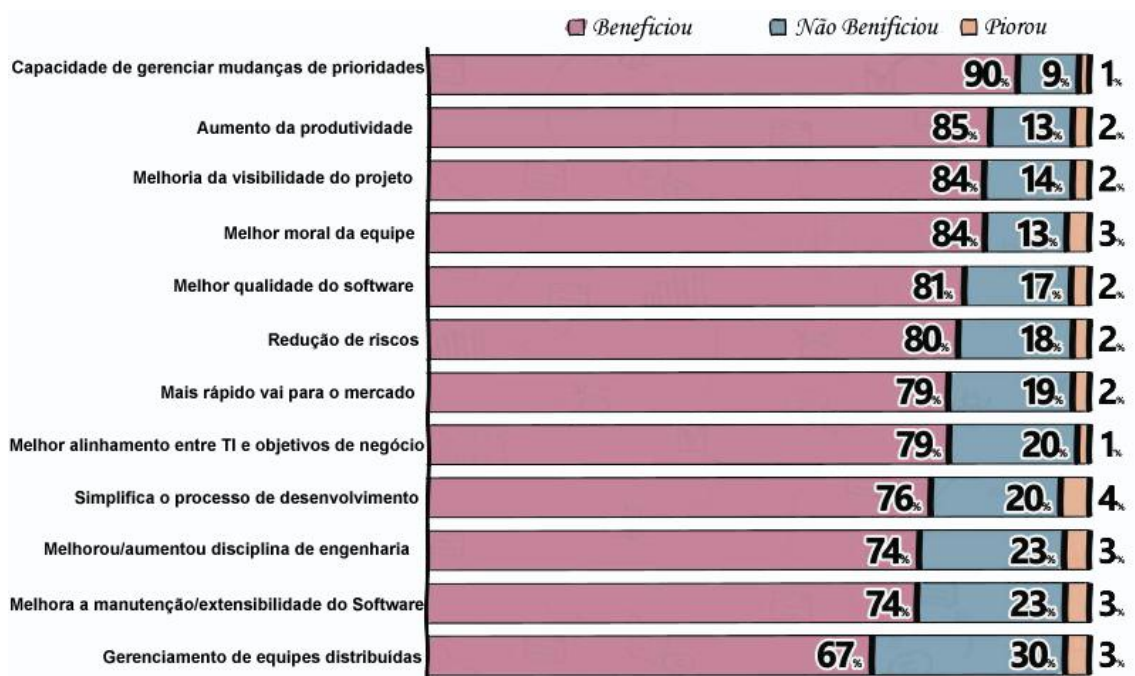


Gráfico 1 - Melhoria no Gerenciamento de Projetos de Empresas que Adotaram as Metodologias Ágeis.

Fonte – Versionone - 3rd Annual Survey: 2008 (VERSIONONE, 2012).

Outros aspectos visados na metodologia ágil é a priorização do cliente e das pessoas da equipe. O cliente é aproximando mais da equipe sendo reconhecido até como um membro dela. Como equipe há mais valorização das competências e equilíbrio da mesma, pois todos participam dos processos. Existem vários métodos ágeis disponíveis que podem ser utilizadas e adaptadas para qualquer área ou projetos.

3.2.1 Extreme Programming (XP)

É uma metodologia ágil voltada para pequenas e médias equipes de desenvolvimento com menos de 10 pessoas que trabalham em base com requisitos vagos e em constante mudança (SBROCCO e MACEDO, 2012). O XP possui algumas características que são marcantes nele como o *feedback* constante, abordagem incremental e propícia a comunicação entre as pessoas envolvidas na equipe.

A *Extreme Programming* tem a vantagem do ganho de eficiência e eficácia no desenvolvimento de sistemas, pois na metodologia a programação é feita em pares para evitar erros e melhorar o desenvolvimento em equipe, também é capaz de suportar as mudanças contínuas de mercado. A figura 13 ilustra os processos existentes no XP.



Figura 13 - Representação do Modelo Ágil XP

3.2.2 SCRUM

O *Scrum* se baseia em algumas características como a flexibilidade dos resultados, a flexibilidade dos prazos, equipe pequena, revisões frequentes, colaboração e orientação a objetos.

Um dos aspectos importantes do *Scrum* é que ele não é exclusivo para os projetos de desenvolvimento de software, ele pode ser usado para o desenvolvimento de qualquer produto ou até mesmo o gerenciamento de qualquer trabalho. Isso ocorre devido às suas características interativas e incrementais (COHN, 2011).

O produto é desenvolvido em partes (*Sprints*) e ao final de cada parte é feito um novo incremento até que chegue ao seu final, e cada parte finalizada é entregue ao cliente.

As principais vantagens na utilização do *Scrum* é a maior motivação da equipe com o comprometimento para atingir as metas estipuladas, as prioridades podem sofrer alterações, facilitando assim a manutenção e retenção de erros que podem surgir durante o desenvolvimento (FERNANDES e ABREU, 2012).

3.2.2.1 Processos de desenvolvimento

Os principais processos no desenvolvimento através da metodologia *Scrum* são ilustrados na figura 14, e se caracterizam com uma reunião entre clientes e desenvolvedores a fim de definir o *backlog* do produto, sendo esse uma lista de requisitos a serem elaborados. Nesse momento também são estimados o custo do projeto, prazo de entrega, é feita a análise de risco, ocorre à escolha das ferramentas de trabalho e também a definição do *Scrum Master*. Após a definição dessas etapas a equipe passa a definir a *Sprint Backlog* que contem uma lista de atividades a serem desenvolvidas e são distribuídas entre os membros da equipe. Ao final de cada *Sprint* ocorre um incremento do produto que é apresentado ao cliente, caso ocorra algum defeito é adicionado no *backlog do produto*. No decorrer do desenvolvimento são aplicados mecanismos de controle do *Scrum* como: ações sobre funcionalidades não entregues, problemas técnicos encontrados, correção de defeito entre outras funcionalidades (SBROCCO e MACEDO, 2012).

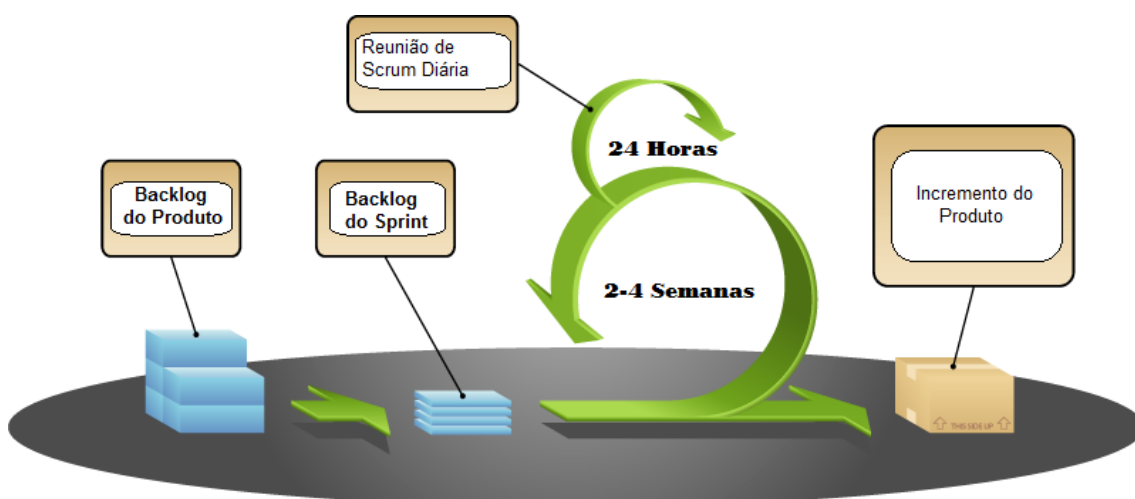


Figura 14 - Representação da Metodologia Ágil – Scrum. Fonte: (COHN, 2011).

3.2.2.2 Papéis

Na metodologia *Scrum* existem papéis definidos para os membros da equipe, que são divididos em *Product Owner*, *Scrum Master* e *Time Scrum*.

O *Product Owner* é a pessoa responsável por gerenciar o *Backlog* do Produto disponibilizado a todos, também gera e informa os requisitos do projeto, assim como as tarefas seguintes, priorizando as de maior valor ao projeto, já o *Scrum Master* é o responsável por implementar e assegurar que todos os envolvidos, inclusive o *Time Scrum* que é o grupo de desenvolvedores responsáveis pela interação e pelo projeto

como um todo, sigam as regras e praticas da metodologia (FERNANDES e ABREU, 2012).

3.2.2.3 Artefatos

Existem dois tipos de artefatos no *Scrum* que são os *Backlog* que é dividido em dois grupos o *Backlog* do Produto e o *Backlog* da *Sprint* que são tabelas de tarefas e interações que estão sendo desenvolvidas e o *Burndown* que são gráficos que informam o que ainda falta para ser desenvolvido.

Backlog do Produto: é um documento elaborado e mantido pelo *Product Owner*, onde é sempre atualizado com melhorias e novas atividades. Contém uma lista de tudo o que constitui as mudanças que serão realizadas no produto para suas futuras versões (FERNANDES e ABREU, 2012).

Backlog da *Sprint*: define as tarefas que o Time *SCRUM* deve executar para criar os incrementos do produto durante a execução de uma *Sprint*. As tarefas devem ser documentadas com seu status (não iniciado, em andamento, finalizado) e da quantidade de horas de trabalho restantes a cada dia da *Sprint* (FERNANDES e ABREU, 2012).

Os *Burndown Charts* mostram, graficamente, a quantidade de trabalho estimado ao longo do tempo (FERNANDES e ABREU, 2012).

A disposição dos elementos do *Scrum* apresentados são visualizados em sua maioria em lousas, planilhas e ainda muito pouco utilizadas em dispositivos móveis.

Através de lousa os elementos são disponibilizados em um quadro ou espaço viável podendo ser uma parede, vidro ou mesa onde se possa fazer a exposição das informações representadas através de post-its. Essas informações são chamadas de *Sprints backlog* tendo o objetivo de orientar e organizar as tarefas de uma *Sprint* para a visualização da equipe.

As planilhas é outra forma de organizar, sistematizar e mensurar informações para o processo de desenvolvimento do software. Essas informações podem ser visualizadas em diferentes planilhas (exemplo MS Excel, planilhas do Google Docs, LibreOffice Calc, etc).

Também se podem utilizar dispositivos móveis para exibir as atividades desempenhadas em um *Sprint*, mas, no entanto existem poucas aplicações que utilizam *Tablet e Smartphone* para visualização dos dados gerados pelos elementos anteriores citados. Algumas aplicações que disponibilizam essa visualização são: *OnTime Scrum* (disponível para multi-plataforma) e *ScrumDesk* (disponível para *iPhone, iPad e iPod*

Touch), disponíveis no mercado, porém ambas são ferramentas pagas, possuindo assim um custo para serem utilizadas.

3.3 Trabalhos relacionados

A utilização de aplicações para o gerenciamento de projetos em dispositivos móveis é uma realidade e um campo de pesquisa que tem muito a agregar, pois a mobilidade e recursos de comunicação oferecidos pelos dispositivos são de grande valia para quem se locomove muito e precisa estar em constante comunicação.

Braga (2012) em seu projeto o MB Project – Sistema de Controle de Projetos, para o gerenciamento de projetos que tem como linguagem de desenvolvimento o Java utiliza recursos para a visualização na web, sendo acessado por meio de um navegador em ambiente *desktop*.

Andrade (2012) em seu trabalho é apresentado uma abordagem de gerenciamento de projetos de software utilizando de dispositivos móveis para isso, onde visa à disponibilização de material para o apoio ao gerente de projetos na execução e planejamento de desenvolvimento de software.

Anacleto (2012) em sua dissertação aborda o desenvolvimento de aplicações web para *smartphones* com a utilização de tecnologias e linguagens web tais como o HTML5, JavaScript e CSS3 sendo capazes fornecer uma experiência de interação semelhante à de aplicações nativas, onde adaptou para um *smartphones* os recursos que executam a monitorização e controle de uma rede de painéis digitais.

Tais trabalhos apresentados tem em comum a visão de mobilidade, pois fazem o uso de dispositivos móveis para executar suas aplicações desenvolvidas. Destacam-se os trabalhos de Braga (2012) e Andrade (2012) por apresentar uma proposta de gerenciamento de software que possa ser utilizada em um dispositivo móvel dando maior mobilidade ao gerente de projetos, pois tudo que precisa para gerenciar está disponível em sua mão. O trabalho de Anacleto (2012) tem seu foco voltado ao gerenciamento de uma rede, porém fez o uso de ferramentas de desenvolvimento apresentadas nesse trabalho, como o HTML5, JavaScript e CSS3 para a criação de sua aplicação, dando uma maior abrangência ao seu trabalho, pois pode ser visualizado em qualquer dispositivo que possua um navegador WEB, desde um dispositivo móvel a um computador pessoal podendo ser ele um *Desktop* ou *Notebook*.

Paralelo aos trabalhos referidos acima o presente trabalho faz uma abordagem a gerenciamento do projeto de software apresentados também por Braga (2012) e Andrade (2012), porém utilizando a metodologia ágil *Scrum* para a sua elaboração, além de utilizar recursos de desenvolvimento WEB que também foram apresentados por Anacleto (2012) buscando aliar simplicidade na adaptação de uma metodologia muito usada no gerenciamento de projeto de software, podendo executar em vários dispositivos independente de suas características.

4 DESENVOLVIMENTO DE UMA APLICAÇÃO PARA GERENCIAMENTO DE PROJETOS DE SOFTWARE ADAPTADA PARA DISPOSITIVOS MÓVEIS

O propósito deste trabalho é apresentar o estudo sobre os recursos existentes para a criação de aplicações web e aplicações nativas, sendo que, o seu foco é voltado às tecnologias que permitam que a mesma aplicação possa ser executada em qualquer tipo de configuração de tela e dispositivo independente de sistema operacional.

Durante o trabalho foi desenvolvido um protótipo nomeado de *My App Scrum*, com o objetivo de unir técnicas utilizadas para o gerenciamento de projetos através da metodologia *Scrum*, adaptando-a para a utilização em vários tipos dispositivos móveis e PC, que possuem algumas diferenças, sendo elas tamanho de tela ou desempenho de hardware. Desta forma, os estudos desenvolvidos bem como a criação do protótipo evidenciam a implementação de uma ferramenta de auxílio para o gerenciamento no desenvolvimento de um projeto software, apresentando as características principais da metodologia *Scrum* para a realização de tarefas de gerenciamento de software.

4.1 Ferramentas utilizadas

Para a criação do protótipo foi elaborado um estudo de algumas ferramentas para serem utilizadas na adaptação. Dentre estas, as tecnologias escolhidas para o desenvolvimento da aplicação foram o HTML5, CSS3 e jQuery Mobile com padrões de web responsivo utilizando alguns dos recursos que as tecnologias podem oferecer para o desenvolvimento de interface usuário de uma aplicação. Também utilizando o PHP (*Hypertext Preprocessor*) como linguagem para interação com o banco de dados MySQL instalado no servidor juntamente com o EasyPHP WebServer, que executa o apache e outras ferramenta para a apresentação, gerenciamento e desenvolvimento de web sites, ferramentas estas, instaladas em um *notebook* com configuração descritas no quadro 5.

Quadro 5 - Características do Notebook

Características do <i>Notebook</i>	
Modelo - CCE GT23	Processador - Intel® Core I3 330M
Memória RAM - 2GB DDR3	Tela - 14.1" LED Widescreen
Sistema Operacional - Windows 7 Professional	Disco Rígido(HD) - 320 GB

Placa Gráfica - INTEL® GRAPHIC MEDIA ACCELERATOR HD	Rede - 10/100 MBPS
Rede Sem Fio - Wireless Lan 802.11 B/G	Chipset - INTEL® HM55

Foi escolhido o HTML5, CSS3 e jQuery Mobile por serem tecnologias muito conhecidas e largamente utilizadas para o desenvolvimento WEB, além de executar em qualquer dispositivo móvel (*Smartphone, Tablet*) ou computadores pessoais (*notebook* ou *desktop*), que possua um navegador web (*browser*) e suporte a JavaScript.

Como ferramenta auxiliar de desenvolvimento foi adotado a utilização do Adobe Dreamweaver CS6, pois possui integração com todos os recursos utilizados no HTML5, CSS3 e jQuery Mobile disponíveis em seu conjunto de ferramentas para o desenvolvimento.

4.2 Protótipo

O protótipo é composto por uma base de dados MySQL instalada no servidor que armazena todas as informações geradas pelos usuários da aplicação, armazenada no servidor web, que contém todos os arquivos que irão interagir com o usuário e a base de dados

4.2.1 Estrutura da base de dados

A base de dados armazena todas as informações das interações dos usuários com a aplicação, utiliza o sistema de gerenciamento de banco de dados (SGBD) MySQL com linguagem de consulta SQL, disponibilizado assim um fonte de informações para os demais usuários com permissão de acesso visualizarem.

A figura 15 ilustra o diagrama de entidades e relacionamentos com algumas das principais tabelas contidas na base e no Anexo A consta o DER mais completo da base de dados.

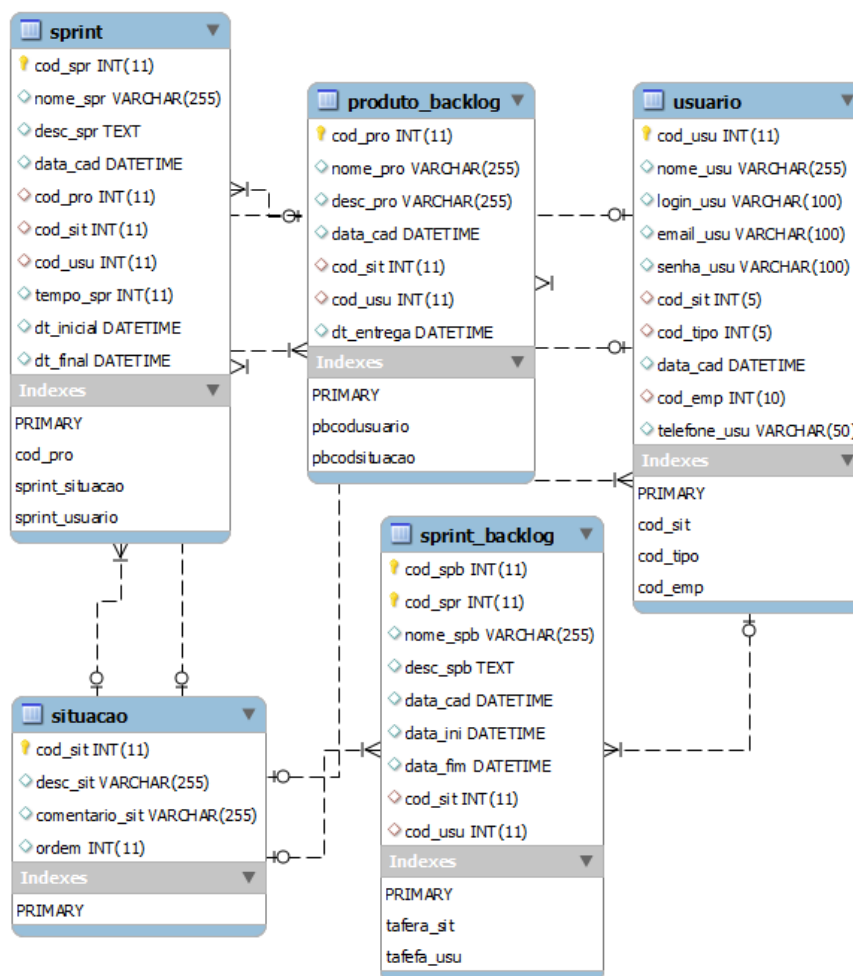


Figura 15 - Diagrama entidade relacionamento da Base de Dados

4.2.2 Casos de uso e diagramas da aplicação

A interação do usuário e a base de dados são feitas através da linguagem PHP, onde os usuários são divididos em dois grupos, sendo eles os *Scrum Master* e Programadores e são autenticados em sessão quando efetuam login na aplicação web dando assim maior segurança às atividades que serão realizadas pelo usuário.

O *Scrum Master* é responsável pela administração de todas as atividades incluídas no protótipo, sendo o responsável pelo bom funcionamento e alimentação de informações da aplicação, informando tarefas a serem feitas, detalhes para a realização do projeto, criação e manutenção dos usuários, identificação e cadastro de *sprints* e suas respectivas tarefas tendo o controle de tempo para o bom andamento do projeto, a figura 16 ilustra as atividades que o usuário *Scrum Master* irá executar na aplicação.

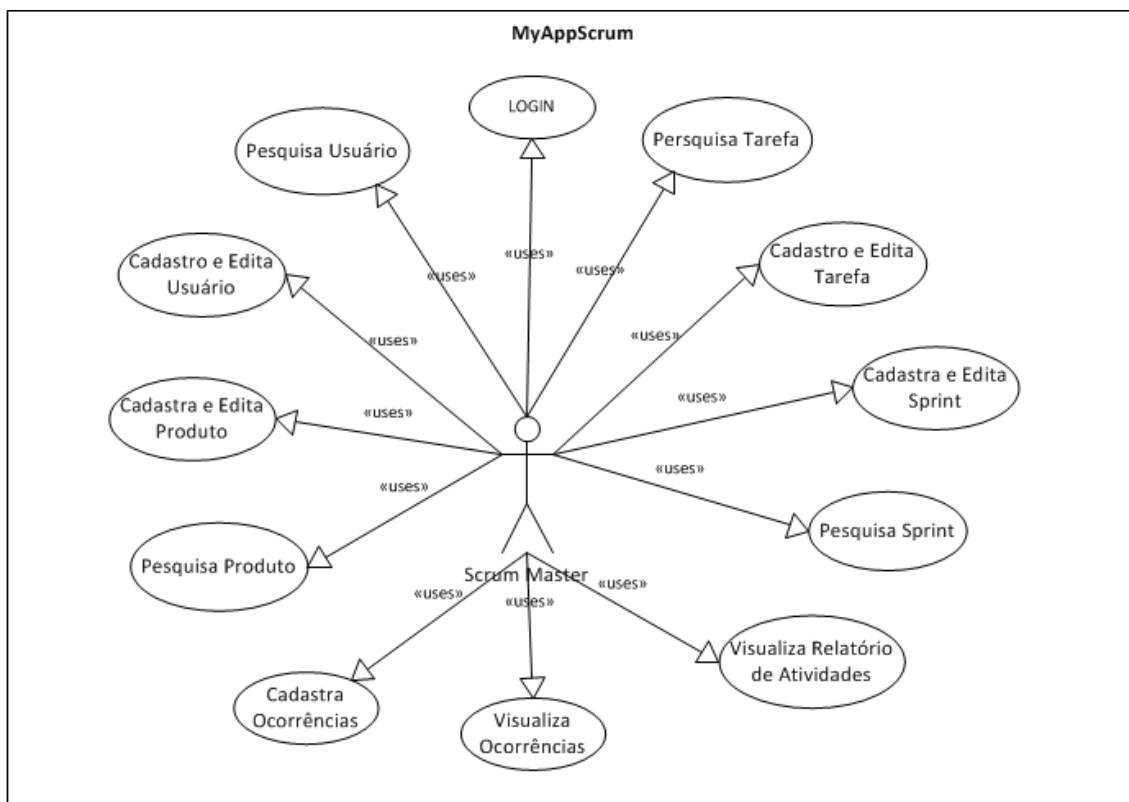


Figura 16 - Caso de Uso Usuário - *Scrum Master*

Já os usuários programados têm como atividades, aquelas vinculadas às tarefas que são inseridas pelo *Scrum Master* na aplicação e podem ser manipuladas pelo programador.

O programador pode abrir a tarefa para visualizar a situação da mesma, e se a tarefa estiver pendente o usuário programador com sessão iniciada pode alterar a situação da tarefa para “fazendo”. Caso a tarefa esteja feita porém não finalizada o usuário pode testar a tarefa e posteriormente finalizá-la, também inserir ocorrências relacionadas às tarefas que estava fazendo caso ocorram, além de editar e salvar informações pessoais de seu usuário, como nome e telefone. A figura 17 ilustra as atividades que o usuário Programador irá executar na aplicação.

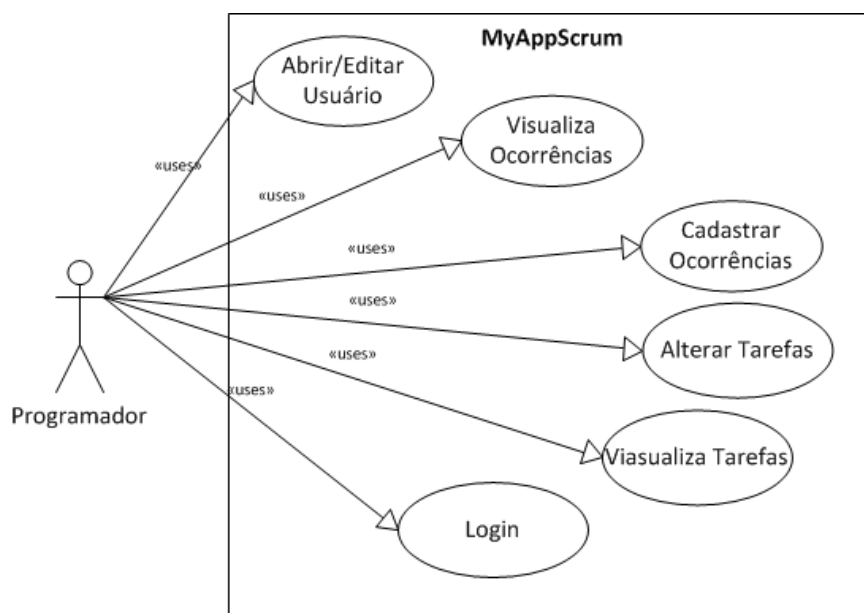


Figura 17 - Caso de Uso do Usuário - Programador

O login é a primeira tela de interação com o usuário, sendo que o mesmo deve informar os dados para ter acesso à aplicação independente do tipo de usuário, desta forma há um retorno se o que foi informado é válido ou não. No quadro 6 são apresentadas as características do caso de uso do login para ambos os tipos de usuários.

Quadro 6 - Características do caso de uso: Usuário – Login

Nome do caso de uso	Login
Sumário	Usuário usa a tela de login para criar uma sessão identificada com seus dados.
Ator primário	Scrum Master/Programador
Atores secundários	Usuário
Precondições	O Scrum Master/Programador possuir credenciais para acesso.
Fluxo Principal	<p>P1. A tela de login é apresentada.</p> <p>P2. O usuário informa suas credenciais para o login, sendo elas Usuário e Senha.</p> <p>P3. O usuário clica em enviar.</p> <p>P4. A aplicação valida a credencial informada.</p> <p>P5. O aplicativo registra o log do evento.</p> <p>P6. O usuário acessa a tela principal da aplicação.</p>
Fluxo Alternativo	<p>1 Entrar em contato para solicitar uma credencial</p> <p>1a. Se o usuário não possui uma credencial para acesso ao aplicativo, pode entrar em contato através do link de contato disponível na tela de login.</p> <p>1b. O usuário deve encaminhar um e-mail para o administrador da aplicação e aguardar um retorno dentro de 24 horas.</p> <p>1c. Retorna para a tela de login (P1).</p> <p>2 Validação dos dados</p> <p>2a. A aplicação apresenta a mensagem: “Falha no login. Entre em contato ou tente novamente!” (P1).</p> <p>3 Dados obrigatórios</p> <p>3a. Se usuário não preenchido a aplicação apresenta a mensagem: “Informe o usuário!” (P1).</p> <p>3b. Se senha não preenchida a aplicação apresenta a mensagem: “Informe a senha!” (P1).</p>
Pós-condições	O usuário tem suas credenciais validadas e acesso a tela principal e demais atividades que são de sua competência (P6).

O usuário a efetuar o login informa seus dados, após seus dados passam por um teste de validação, se o resultado for verdadeiro ele acessa a aplicação e encerra a atividade de login, se a validação retornar falso, o usuário é encaminhado para um processo onde pode escolher entrar em contato com o administrador da aplicação e encerrar a atividade de login, ou simplesmente encerrar a atividade de login e retornar ao início para tentar novamente.

4.2.3 Telas do protótipo

A primeira tela apresentada no acesso ao protótipo é a tela de login, que contém recursos CSS3 para estilizar a tela, também contendo características responsivas que se adaptam a várias resoluções através da utilização de *media screen*, que é uma *tag* de marcação CSS para definir a execução do estilo a partir de sua abertura até o seu fechamento na resolução definida nela. A figura 18 apresenta um trecho de código com essa marcação e na figura 19 é ilustrada a tela de login em resoluções maiores e menores que 480 px.

```
312 @media screen and (max-width: 480px) {
313     html {-webkit-text-size-adjust: none;}
314     .usu{display:none;}
315     #main {float: none;width: auto;padding: 20px;}
316     h1 a img { width: 185px;}
317     h2 {font-size: 2em;}
318     #form-oldschool {float: none; margin: 0; padding:
319         0px; border: 0; width: 100%; box-sizing: border-box;
320     }
321     div.sign-connections {width: 100%; box-sizing: border-box;
322         padding: 10px 0 0 0; margin-top: 20px;
323         border-top: 1px solid rgba(255, 255, 255, 0.1);
324     }
325     form fieldset { margin: 15px 0;}
326     label {margin-bottom: 5px;}
327     input.text-input { width: 100%; box-sizing: border-box;
328         border: none; font: 1.6em "Helvetica Neue", helvetica, sans-serif;
329         padding: 8px 10px; margin: 0px;
330     }
331     input.button { padding: 15px 0; margin: 15px 0 0;
332         width: 100%;
333     }
334     div.form-inline form fieldset, div.form-inline
335         form input.button { display: block;}
336     div.form-inline form input.text-input { width: 100%;}
337 }
```

Figura 18 - Marcação media screen no CSS



Figura 19 - Apresentação da tela de login em resoluções maiores e menores que 480px

Na tela de login o usuário efetua a sua autenticação através do usuário criado na base de dados, sendo que com a linguagem PHP e SQL é feito a consulta para autenticar o usuário e criar uma sessão para ele. Após esse processo é carregado à tela principal da aplicação que contém recursos diferentes dos apresentados na tela de login, pois é utilizado em sua codificação recursos de interface e usuário do *Framework* jQuery Mobile e mais recursos novos disponíveis no HTML5.

A tela principal da aplicação web executa recursos de interface do jQuery Mobile que aplica efeitos visuais na aplicação dando aspectos visuais mais parecidos com de aplicações de dispositivo móvel. O jQuery Mobile é um *framework* de interface de usuário com um design flexível, construído com base no jQuery, também possui um conjunto de recursos disponíveis para ser utilizado em aplicações web. O protótipo faz o uso de alguns desses recursos como efeito de transição, estilos de menu, botões, *forms* entre outros que são descritos nas telas de interação da aplicação, oferecendo uma interface mais interessante de ser usada pelo usuário, à figura 20 ilustra a tela principal da aplicação que é visualizada por todos os usuários.



Figura 20 - Painel de Tarefas da Tela Principal

Quando o usuário clicar no ícone ao lado da descrição da tarefa, é gerado um *page pop*² onde o utilizador pode escolher informar algum tipo de ocorrência ou alterar a situação da tarefa, a figura 21 ilustra o *page pop* de ocorrência gerado. Na tela da ocorrência o usuário utilizador informa o produto, a Sprint e a tarefa que a ocorrência pertence, os tipos de ocorrência são padronizados e cadastrados pelo *Scrum Master* e os demais utilizadores fazem uso dos tipos de ocorrências disponibilizados.

Figura 21 - Tela de registro de ocorrências

² Página criada em forma de pop-up, que é uma janela extra que abre no navegador

O utilizador pode informar na ocorrência uma descrição do que ocorreu, permitindo um melhor entendimento pelos demais utilizadores que poderão visualizar a ocorrência.

Também ao clicar em menu na página principal, o mesmo é deslizado da esquerda para direita, ficando sobre a página que o solicitou sendo ela a principal ou qualquer outro contida na aplicação, mostrado em forma de painel, esse efeito é possível pela utilização de recurso do HTML5, CSS3 e principalmente do jQuery Mobile, sendo que o conjunto de código gera esse efeito e estilo no menu. A figura 22 mostra o código utilizado para montar o menu e o menu ao seu lado.

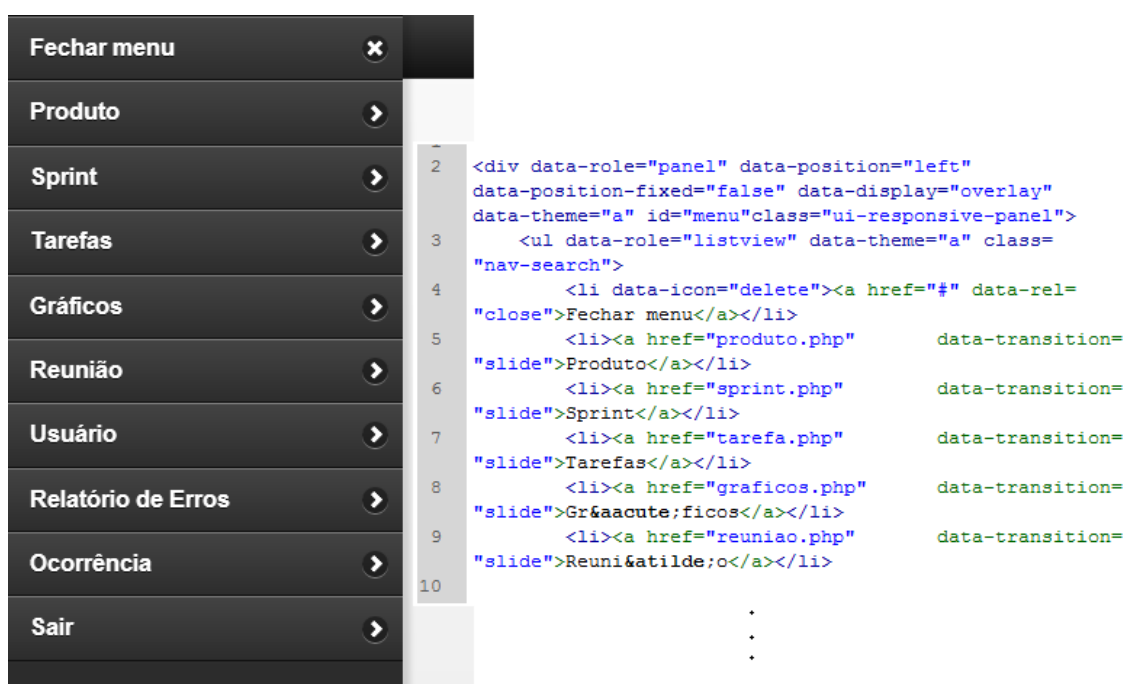


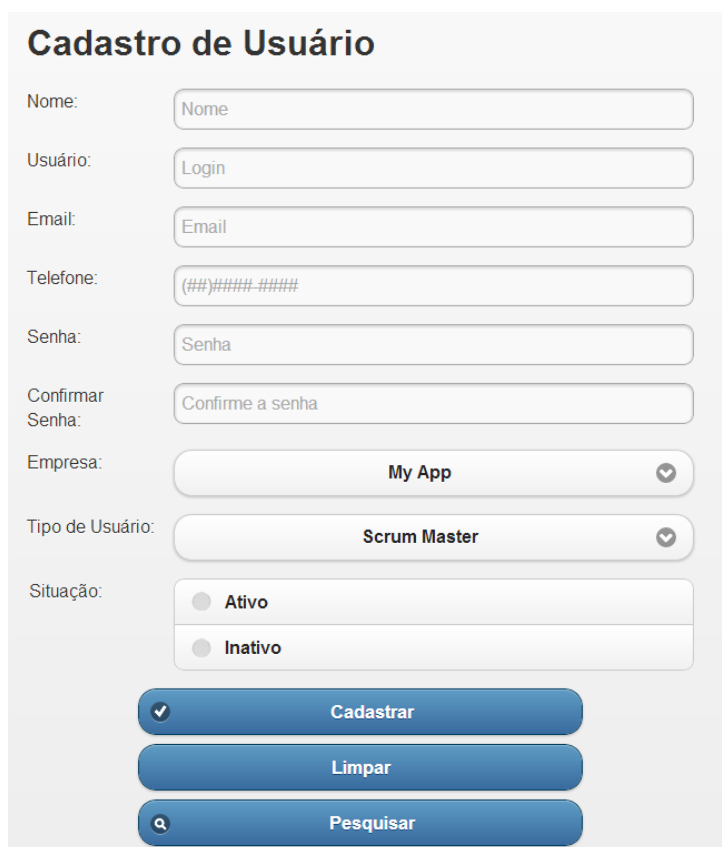
Figura 22 - Painel do menu e o Código de sua montagem

No código da figura 22 percebem-se algumas *tags* familiares `<div>`, `` e ``, mas os atributos menos conhecidos como *data-role*, *data-position*, *data-display* e *data-theme* que se apresentam fazem parte da versão cinco sendo que o *data-* indica um atributo de referência a um objeto que neste caso é um elemento do jQuery Mobile, mas também pode ser um elemento criado através do JavaScript.

Nos links que referenciam outras páginas desenvolvidas para o protótipo é apresentado mais um elemento que faz parte do *Framework* jQuery Mobile, que é o efeito de transição entre as páginas apresentado pelo atributo *data-transition* na tag HTML ``, o atributo pode ser definido por vários tipos de efeitos de transição já definidos no *framework*, como *pop*, *fade*, *flip*, *flow*, *slide*, *slideup* e outros disponíveis e

especificados na documentação do *framework*, também pode-se editar esses efeitos para se adaptar ao que o desenvolvedor deseja apresentar na aplicação.

Para definir as permissões de acesso às telas do sistema, o usuário administrador irá definir o tipo de usuário que está cadastrando na tela de cadastro de usuário que é ilustrada na figura 23, podendo ser do tipo *Scrum Master* ou Programador, o *Scrum Master* tem acesso a todas as telas da aplicação e é definido como administrador, já o tipo Programador tem acesso a algumas telas que fazem parte de sua competência, essas atividades estão definidas nos casos de uso apresentados na figura 16 para o *Scrum Master* e na figura 17 para o Programador que informa as telas que o usuário tem acesso.



A tela de cadastro de usuário apresenta o seguinte formulário:

- Nome:** Campo de texto com o placeholder "Nome".
- Usuário:** Campo de texto com o placeholder "Login".
- Email:** Campo de texto com o placeholder "Email".
- Telefone:** Campo de texto com o placeholder "(##)##### ####".
- Senha:** Campo de texto com o placeholder "Senha".
- Confirmar Senha:** Campo de texto com o placeholder "Confirme a senha".
- Empresa:** Menu suspenso com o valor selecionado "My App".
- Tipo de Usuário:** Menu suspenso com o valor selecionado "Scrum Master".
- Situação:** Grupo de botões com duas opções: "Ativo" (selecionado) e "Inativo".

Na base do formulário, há três botões de ação:

- Botão "Cadastrar" com um ícone de marca de seleção.
- Botão "Limpar".
- Botão "Pesquisar" com um ícone de lupa.

Figura 23 - Tela do Cadastro de Usuários

Na tela de cadastro do usuário devem ser informados os dados solicitados no formulário, caso alguma informação essencial não for informada, ao clicar no botão cadastrar uma mensagem será apresentada na tela solicitando que o campo seja preenchido. Essa tela possui alguns campos novos de dados da versão do HTML5, como o *email* e *tel*, que possuem uma validação própria e não é necessário criar um código de validação dos campos, pois ao sair do campo já é informado se as informações ali inseridas são válidas ou não. Ao clicar no botão pesquisar o usuário

administrador visualiza todos os usuários cadastrados e pode editar algum usuário e atualizar os dados do seu cadastro.

Clicando no menu e no link “produto” o utilizador será redirecionado para a tela de cadastro de Produto *Backlog*, que tem como objetivo informar o projeto que será desenvolvido. Na tela de cadastro do produto se encontra outros elementos visuais do *framework* jQuery Mobile juntamente com novos elementos disponibilizados no HTML5 como o novo tipo de dado *date* da *tag input*. A tela de cadastro de produto é acessada apenas pelo usuário *Scrum master* e tem como objetivo informar o produto/projeto que será trabalhando, sendo que o usuário pode inserir mais que um produto para ser trabalhado e mudar a situação do produto caso seja necessário. A figura 24 ilustra os itens que são apresentados na tela de cadastro do Produto *Backlog*



A imagem mostra a interface de usuário para o cadastro de um produto backlog. O formulário contém os seguintes campos e elementos:

- Nome:** Campo de texto com o placeholder "Nome".
- Descrição:** Campo de texto com o placeholder "Descrição".
- Situação:** Grupo de botões de rádio com as opções "Ativo" e "Inativo".
- Data de entrega:** Campo de data com o formato "dd/mm/aaaa" e ícones de navegação.
- Botões de Ação:** Três botões azuis: "Cadastrar" (com ícone de checkmark), "Limpar" e "Pesquisar" (com ícone de lupa).

Figura 24 - Tela de Cadastro do Produto Backlog

A figura 24 do cadastro do produto *backlog* o usuário pode efetuar pesquisas e editar produtos cadastrados clicando no botão pesquisar e também ao informar os dados solicitados no formulário de cadastro e clicando em cadastrar, um novo produto é inserido na base de dados, também já é disponibilizado para efetuar vínculos no Cadastro de uma nova Sprint na Tela de cadastro de Sprint que é ilustrada na figura 25.

Sprint

Projeto: Projeto TCC

Nome:

Objetivo:

Tempo para a Tarefa:

Data de Início:

Data de Término:

Situação:

Cadastrar

Limpar

Pesquisar

Figura 25 - Tela de Cadastro de Sprint

Na tela de cadastro de Sprint o usuário seleciona a qual produto a Sprint pertence e informa o nome e objetivo da *Sprint*, determina também o tempo destinado para realização da *Sprint* assim como a data de início e final da mesma. O usuário também deve informar corretamente a situação da *Sprint*. A situação é cadastrada e a nomenclatura é definida pelo usuário *Scrum Master*. Após o cadastro da Sprint que será trabalhada o usuário *Scrum Master* já poderá informar às tarefas que pertencem a Sprint na tela de cadastro de tarefas da *Sprint* que é ilustrada na figura 26.

Tarefas da Sprint

Sprint: asd

Nome:

Descrição:

Data de Início:

Data de Término:

Situação:

Cadastrar

Limpar

Pesquisar

Figura 26 - Tela de Cadastro de Tarefas da Sprint

As tarefas que são cadastradas, de acordo com a metodologia de desenvolvimento ágil *Scrum*, são as mesmas que discutidas em reuniões e listadas pelo *Scrum Master* onde é definida uma data inicial e final para a realização de cada tarefa. Como todas as telas de cadastro, a tela tarefas da *Sprint* também contém os botões de cadastro, limpar e pesquisa com as mesmas funções, cadastrar uma nova tarefa, limpar o formulário e pesquisar tarefas cadastradas para visualizar e editar se for necessário.

A aplicação possui uma tela opcional que pode ser utilizada pelo *Scrum Master* se preferir informar os acontecimentos das reuniões diárias com a equipe, esses registros são informados nas telas de cadastro das atas de reuniões, cujo objetivo central é manter os registros das reuniões, o usuário deve informar os campos existentes no formulário e clicar em cadastrar, como é feito nas demais telas da aplicação, também possui uma opção de pesquisa para visualizar as atas de reuniões já cadastradas, podendo também editá-las. A figura 27 ilustra a tela de cadastro das atas das reuniões realizadas e também os campos que nela contém e devem ser informados pelo usuário, caso o usuário não preencha um campo que é requerido uma mensagem irá avisá-lo referente à qual informação não foi informada.



Ata de Reunião

Número da Ata:

Sprint:

Pautas:

Descrição:

Atividades:

Data da Reunião:

Cadastrar

Limpar

Pesquisar

Figura 27 - Cadastro da Ata de Reunião

4.3 Testes da aplicação

O teste com o aplicativo foi realizado em um dispositivo móvel (*Smartphone Xperia Neo V*) e em um *notebook* (CCE GT23), sendo que o aparelho móvel utilizado tem suas características detalhadas no quadro 7 abaixo e o *notebook* foi o mesmo utilizado como servidor e suas características estão detalhadas no quadro 5.

Quadro 7 - Características do Smartphone Sony Ericsson Xperia neo V

Características	
OS	Android OS, v4.0
Chipset	Qualcomm MSM8255 Snapdragon
CPU	1 GHz Scorpion
Memória	1 GB (320 MB disponível ao usuário), 512 MB de RAM
GPU	Adreno 205
Sensores	Acelerômetro, proximidade, bússola
Navegador	HTML5, Adobe Flash
GPS	Sim, com suporte A-GPS
Java	Sim, via Java MIDP emulador
WLAN	WI-FI 802.11 b / g / n, DLNA, WI-FI hotspot
Tela	480 x 854 pixels, 3,7 polegadas

Fonte: gsmarena (GSMARENA, 2013).

O *Smartphone* se comportou bem ao abrir a aplicação, as telas se adaptaram a resolução do dispositivo, melhorando a disposição dos elementos apresentados na tela em execução no mesmo ilustrado na figura 28. O aparelho possui um processamento e memória RAM mediano, possibilitando uma melhor experiência para usuário ao executar a aplicação.

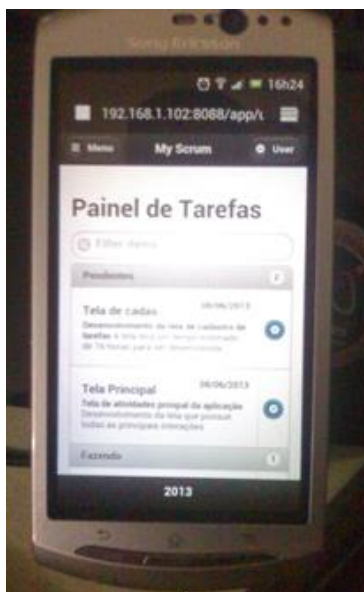


Figura 28 - Aplicação sendo visualizada em um *Smartphone*

Ao visualizar a aplicação no *notebook*, os elementos aumentaram seu tamanho, ajustando-se ao tamanho disponibilizado pela resolução de tela do computador, como está ilustrado na figura 29. O cache de memória do *browser* utilizado pelo *notebook* é maior, alguns efeitos de transição que executaram com um pouco de lentidão no *Smartphone*, se comportam melhor, isto é, com mais rapidez, melhorando a interação do usuário com a aplicação.

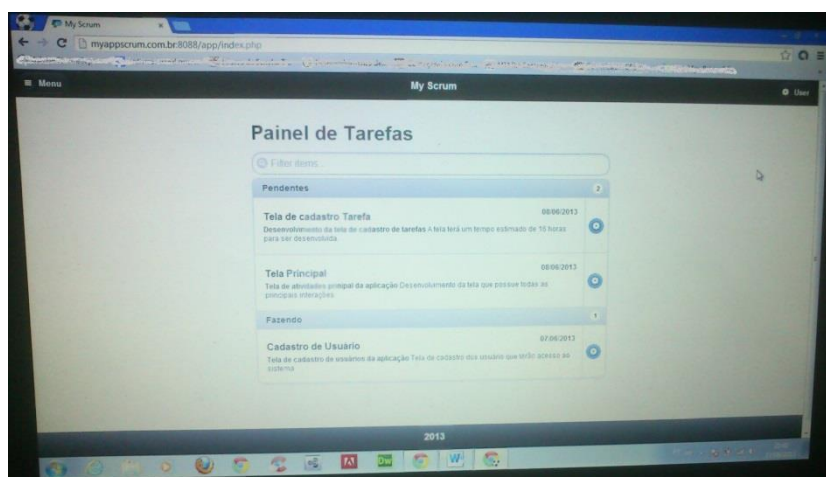


Figura 29 - Tela principal sendo visualizada em um *notebook*

O fato de se ter um pouco de lentidão ao executar a aplicação web *My App Scrum* em um *Smartphone* de desempenho mediano em sua maioria é devido à quantidade de memória para cache disponibilizada pelo navegador dos dispositivos móveis, no entanto mediante as vantagens apresentadas no desenvolvimento de aplicações Web essa lentidão pode ser reparada com a melhor configuração de memória para o navegador, pois muitas plataformas e dispositivos poderão utilizá-la, melhorando a interatividade no gerenciamento de projetos.

Ao utilizar um auxílio para o gerenciamento de um projeto, o gerente pode estar sempre em contato com sua equipe, tendo um aliado para a comunicação e representação de afazeres, os dispositivos móveis podem exercer essa função de auxiliar nos processos de desenvolvimento de um software por vários meios, o qual pode ser ele um *smartphone* ou *tablet* executando uma aplicação web sem a necessidade de codificação para as particularidades dos dispositivos móveis. Com apenas um código pode-se executar uma aplicação em vários dispositivos móveis e PCs, melhorando a interação entre gestor e equipe dando mais qualidade e agilidade ao desempenhar tarefas do gerenciamento de software.

4.4 Considerações finais

As facilidades no desenvolvimento utilizando HTML5, CSS3 e jQuery Mobile é grande, pois são utilizadas linguagens familiares que tem um grande tempo de existência e de fácil compreensão, além de vários dispositivos móveis terem suporte e executarem essas linguagens. Porém vale enfatizar, que apesar da qualidade e desempenho dos dispositivos móveis evoluírem bastante e poderem ser até comparados com alguns computadores de baixo custo, os navegadores dos dispositivos ainda tem dificuldades na renderização de grandes quantidades de arquivos. O cache de memória empregado em seus navegadores é relativamente menor do que é disponibilizado pelos navegadores *desktop*, dificultando a interação com uma tela que possua muitos elementos para serem renderizados em um processo de transição da aplicação, no entanto alguns dispositivos móveis permitem configurar seu navegador, podendo assim disponibilizar mais memória para ser utilizada como cache do mesmo, garantindo um melhor desempenho ao utilizar uma aplicação web.

5 CONCLUSÕES E TRABALHOS FUTUROS

A utilização de vários dispositivos com formatos de visualização e resolução diferentes além de SO distintos foi exigindo a criação de uma mesma aplicação para várias demandas e com a utilização de linguagens de programação diferentes para cada uma, tentando facilitar o uso dos usuários em seus vários dispositivos. Paralelo a tais mudanças, o avanço tecnológico dos dispositivos móveis (*smartphones* e *tablets*) tem permitido a execução de recursos para desenvolvimento de aplicativos cada vez mais complexos, que apresentam interfaces cada vez mais ricas e com funcionalidades que antes só eram possíveis de se visualizar em ambientes *desktops*. Além da disponibilidade de executar uma aplicação tanto em um navegador web com a utilização de linguagens como HTML5, CSS3 e JavaScript, executadas no browser do dispositivo móvel, também é possível instalar uma aplicação no próprio dispositivo criada com ferramentas e linguagens disponibilizadas pela empresa detentora do sistema operacional que é executado no dispositivo, através do mercado de aplicações disponibilizado pela empresa.

Da mesma forma que os dispositivos móveis evoluem, outras alternativas de linguagens de programação surgem para o desenvolvimento de aplicações que são executadas nos dispositivos os quais cada vez disponibilizam mais recursos.

A utilização da web permite a execução de um único aplicativo em vários dispositivos independente do SO ou resolução de tela. Utilizando os recursos do HTML5, CSS3 e jQuery tem-se uma aplicação que pode ser executada em qualquer dispositivo que contenha um navegador e compatibilidade com os recursos utilizados na aplicação, sem a necessidade de criação diferenciada para cada dispositivo. Além da facilidade no desenvolvimento das aplicações utilizando linguagens de programação já conhecida para desenvolvimento permite usar o mesmo código fonte para rodar em vários dispositivos, proporcionando experiências novas na adaptação de interfaces para uma demanda maior. Assim a adaptação de interfaces gráficas em dispositivos móveis é de grande importância, pois ajuda o usuário a ter uma melhor interação com o aparelho utilizado e a aplicação desenvolvida se difunde rapidamente.

Nesse sentido este trabalho apresentou uma das formas de desenvolvimento de aplicações cuja interface se adapta a variados dispositivos desde *desktops* a

Smartphones, apresentando um protótipo desenvolvido para web que adaptou o *Scrum* sendo esta uma metodologia de desenvolvimento ágil de software. A aplicação contém algumas das principais atividades exercitadas no *Scrum*, como descrição de Sprint e Tarefas, auxiliado assim o gerenciamento de projetos de software.

No entanto, ainda há certo receio na utilização da web para aplicações, devido à necessidade de uma conexão para a comunicação de dados, sendo que, apesar das melhorias por parte das empresas de comunicação e evoluções como 3g, ainda há lentidão no acesso a internet através de dispositivos móveis fora das grandes capitais, dificultando o acesso de uma gama de possíveis usuários. Outra dificuldade é o poder computacional dos dispositivos móveis mais acessíveis, pois apesar de muitas melhorias empregadas nos dispositivos alguns recursos apresentam lentidão, como a renderização da interface, e aparelhos com melhor desempenho os quais são muito caros para serem comprados.

Desta forma, o trabalho em pauta destaca o desenvolvimento de uma aplicação web cujo foco é a adaptação em qualquer interface e plataforma, pois, evita gasto em seu desenvolvimento além de sua manutenção ser efetuada em apenas um código fonte.

5.1 Trabalhos futuros

Como sugestões para trabalhos futuros, poderia ser desenvolvida uma aplicação compilada com execução *off-line* e *online*, ou seja, podendo ter ou não uma conexão com a internet, com a execução de uma base temporária local ao executar *off-line* e quando estiver conectado enviar os dados para base de dados remota, garantindo assim o armazenamento dos dados, também, em caso de ausência de conectividade, abrindo assim maiores possibilidades na utilização da tecnologia de aplicações web.

Já na área de adaptação de interfaces para dispositivos móveis pode-se utilizar *frameworks* como o Sencha e PhoneGap para o aperfeiçoamento da aplicação apresentada visando somente a utilização em dispositivos móveis como *Smartphones* e *tablets* para usufruir dos recursos contidos nesses dispositivos como câmera, geolocalização, acelerômetro, contatos entres outros disponíveis, que podem ser extraídos com a utilização do PhoneGap e apresentados de uma melhor forma com a utilização do Sencha, sendo que ambos utilizam o HTML5, CSS para a criação da interface que o usuário ira interagir com a aplicação.

6 REFERÊNCIAS BIBLIOGRÁFICAS

AGILE MANIFESTO. **Princípios por trás do Manifesto Ágil**, 2013. Disponível em: <<http://agilemanifesto.org/iso/ptbr/principles.html>>. Acesso em: 16 fev. 2013.

ALVAREZ, M. A. **Introdução a CSS3**, 2013. Disponível em: <<http://www.criarweb.com/artigos/introducao-a-css3.html>>. Acesso em: 19 mar. 2013.

ANACLETO, J. A. D. C. **Desenvolvimento de uma aplicação web para dispositivos móveis - Monitorização e controlo de uma rede de digital signage**, Dissertação de Mestrado - UNIVERSIDADE DO MINHO. Largo do Paço - Braga, 2012.

ANALYSIS TECNOLOGIA. **O PDA**, 2013. Disponível em: <<http://analysistecnologia.wordpress.com/tipos-de-computadores/o-pda/>>. Acesso em: 20 jul. 2013.

ANDRADE, S. C. D. **Uma abordagem de gerenciamento de projetos de software para dispositivos móveis**, Dissertação de Mestrado - UNIVERSIDADE ESTADUAL DE MARINGÁ. Maringá, 2012.

ANDROID. **Android, the world's most popular mobile platform**, 2013. Disponível em: <<http://developer.android.com/about/index.html>>. Acesso em: 18 abr. 2013.

APPLE. **iPhone**, 2013. Disponível em: <<http://apple-history.com/iphone>>. Acesso em: 16 abr. 2013.

BALSEMÃO,. **Gerência e Monitaramento de Redes Através de Dispositivos Móveis**, 2008. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/15969/000695245.pdf>>. Acesso em: 21 jun. 2013.

BRAGA, M. L. **MB Project – Sistema de Controle de Projetos**, Projeto de Graduação - UNIVERSIDADE CATÓLICA DE PELOTAS. Pelotas, 2012.

CARISSIMI, A. S.; ROCHOL, J.; GRAVILLE, L. Z. **Redes de Computadores**. Porto Alegre: Bookman, 2009. 391 p.

COHN, M. **Desenvolvimento de software com Scrum: Aplicando Métodos Ágeis com sucesso**. Porto Alegre: Bookman, 2011. 496 p.

DAMIANI, E. B. **Guia de consulta Rápida JavaScript**. São Paulo: Novatec ditora Ltda, 2001.

DEVELOPER ANDROID. **Android NDK**, 2013. Disponível em: <<http://developer.android.com/tools/sdk/ndk/index.html>>. Acesso em: 18 mar. 2013.

DEVELOPER APPLE. **Developer Tools**, 2013. Disponível em: <<https://developer.apple.com/technologies/tools/>>. Acesso em: 27 jul. 2013.

DUNCAN, R. **A Guide to The Project Management Body of Knowledge**, 2012. Disponível em: <http://www.unipi.gr/akad_tmhm/biom_dioik_tech/files/pmbok.pdf>. Acesso em: 16 fev. 2013.

EGGERS, I. **Tablet: As experiências do produto entre estudantes de concurso público**, 2012. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/65606/000870714.pdf>>. Acesso em: 21 jun. 2013.

FARIA, Y. L. **Redes sem fio: A evolução da Comunicação Utilizando WiMax**, Trabalho de Conclusão de Curso - INIMINAS. Uberlândia, 2005.

FERNANDES, A. A.; ABREU, V. F. **Implantando a Governança de TI da Estratégia à Gestão dos Processos e Serviços**. 3. ed. Rio de Janeiro: Brasport e Multimídia Ltda, 2012. 615 p.

FERRARI, B. **8 tecnologias móveis para 2010, segundo Gartner**, 2013. Disponível em: <<http://info.abril.com.br/professional/tendencias/8-tecnologias-moveis-para-2010-segundo-gartner.shtml>>. Acesso em: 18 fev. 2013.

GSMARENA. **Sony Ericsson Xperia neo V**, 2013. Disponível em: <http://www.gsmarena.com/sony_ericsson_xperia_neo_v-4122.php>. Acesso em: 22 jun. 2013.

HSIEH, C.-T. **Mobile commerce: Assessing new business opportunities**, 2007. Disponível em: <<http://www.iima.org/CIIMA/13%20CIIMA%207-1-07%20Hsieh%2087-100.pdf>>. Acesso em: 09 fev. 2013.

INTEL. **Iniciando no Intel® HTML5 Development Environment**, 2013. Disponível em: <<http://software.intel.com/pt-br/articles/intel-html5-development-environment-getting-started-guide>>. Acesso em: 21 mar. 2013.

JARDIM, F. M. **Treinamento Avançado em Redes Wireless**. São Paulo: Digerati Books, 2007. 128 p.

JAVA. **Qual é a diferença entre o JavaScript e o Java?**, 2013. Disponível em: <http://www.java.com/pt_BR/download/faq/java_javascript.xml>. Acesso em: 19 mar. 2013.

JQUERY MOBILE. **Mobile Graded Browser Support**, 2013. Disponível em: <<http://jquerymobile.com/gbs/>>. Acesso em: 18 abr. 2013.

LEE, V.; SCHNEIDER, ; SCHELL, R. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. São Paulo: Pearson Makron Books, 2005. 328 p.

LUTZ, M.; ASCHER, D. **Aprendendo Python**. Tradução de João Tortello. 2ª. ed. São Paulo: Artmed, 2007. 566 p.

MANGANELLI, E. C.; ROMANI, J. **Protocolos de Sincronização de Dados em Ambiente Wireless. Um estudo de caso**, Trabalho de Conclusão de Curso - UNIVERSIDADE FEDERAL DE SANTA CATARINA. Florianópolis, 2004.

MAZIERO,. **Sistemas Operacionais: Conceitos e Mecanismos**, 2013. Disponível em: <<http://dainf.ct.utfpr.edu.br/~maziero/lib/exe/fetch.php/so:so-cap01.pdf>>. Acesso em: 21 jun. 2013.

MAZZA, L. **HTML5 e CSS3 - Domine a web do futuro**. São Paulo: Casa do Código, 2012.

MICROSOFT. **Visual Basic**, 2013. Disponível em: <[http://msdn.microsoft.com/pt-br/library/2x7h1hfk\(v=vs.90\).aspx](http://msdn.microsoft.com/pt-br/library/2x7h1hfk(v=vs.90).aspx)>. Acesso em: 18 abr. 2013.

NBR10006. **Gerenciamento da Qualidade - Diretrizes para a qualidade em gerenciamento de projetos**, 2013. Disponível em: <http://bagatin.tech.nom.br/fgvpr_gpp2004/Principios%20de%20Gerencia%20de%20Pr ojetos%20e%20Gerencia%20de%20Integracao/NBR10006.PDF>. Acesso em: 19 mar. 2013.

NETO, J. M. R. **A Cardiologia no Século XXI. O Uso do PDA é Indispensável?**, 2005. Disponível em: <<http://www.scielo.br/pdf/abc/v85n6/27407.pdf>>. Acesso em: 21 jun. 2013.

NIEDERAUER, J. **Desenvolvendo Websites com PHP**. 2ª. ed. São Paulo: Novatec, 2011. 304 p.

OLIVEIRA, A. S. D. **Smartphones e trabalho imaterial: Uma Ernografia virtual sobre sujeitos Usuários de dispositivos móveis convergentes**, Dissertação de Mestrado - UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Porto Alegre, 2007.

PAULON, R. **Web Services in JAVA**, 2004. Disponível em: <<http://imasters.com.br/artigo/1863/java/web-services-in-java/>>. Acesso em: 16 abr. 2013.

PHONEGAP. **about**, 2013. Disponível em: <<http://phonegap.com/about/>>. Acesso em: 18 fev. 2013.

PILGRIM, M. **Dive Into HTML5**, 2013. Disponível em: <<http://diveintohtml5.info/table-of-contents.html#offline>>. Acesso em: 18 mar. 2013.

PITCHENGINE. **What are the differences between tablets and notebooks**, 2013. Disponível em: <<http://new.pitchengine.com/pitches/266f65a9-dfb1-42dd-9162-d145bab37d4a>>. Acesso em: 20 jul. 2013.

ROCHA, D. Utilizando web services em aplicações móveis. **WebMobile**, n. 16, p. 82, 2009.

ROMEIRO, B. G. B. D. A. **Desenvolvimento de Aplicativos para Dispositivos Móveis na Plataforma J2ME**, Trabalho de Conclusão de Curso - ESCOLA POLITÉCNICA DE PERNAMBUCO. Recife, 2005.

SBROCCO, H. T. D. C.; MACEDO, P. C. **Metodologias Ágeis: Engenharia de Software sob Medida**. 1. ed. São Paulo: Érica, 2012. 237 p.

SIXREVISIONS. **Native App vs. Mobile Web App: A Quick Comparison**, 2012. Disponível em: <<http://sixrevisions.com/mobile/native-app-vs-mobile-web-app-comparison/>>. Acesso em: 18 fev. 2013.

SOMMERVILLE, I. **Engenharia de Software**. 6. ed. [S.l.]: Addison Wesley, 2003. 568 p.

STROUSTRUP, B. **A History of C++: 1979– 1991**, 2013. Disponível em: <<http://www.stroustrup.com/hopl2.pdf>>. Acesso em: 18 jun. 2013.

SUMMERFIELD, J. **Mobile Website vs. Mobile App (Application): Which is Best for Your Organization?**, 2013. Disponível em: <<http://www.hswsolutions.com/services/mobile-web-development/mobile-website-vs-apps>>. Acesso em: 21 mar. 2013.

TURBAN, E.; KING, D. **Comércio Eletrônico: Estratégia e Gestão**. São Paulo: Pearson / Prentice Hall, 2004. 456 p.

URUBATAN, R. **Ruby on Rails - Desenvolvimento Fácil e Rápido de Aplicações WEB**. 2ª. ed. São Paulo: Novatec, 2012. 336 p.

VERSIONONE. **7th Annual State of Agile Development Survey**, 2012. Disponível em: <<http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>>. Acesso em: 20 jul. 2013.

W3C. **Extensible Markup Language (XML)**, 2013a. Disponível em: <<http://www.w3.org/XML/>>. Acesso em: 12 jun. 2013.

W3C. **HTML5 Curso W3C Escritório Brasil**, 2013b. Disponível em: <<http://www.w3c.br/pub/Cursos/CursoHTML5/html5-web.pdf>>. Acesso em: 12 jun. 2013.

WINDOWS. **Histórico de atualizações do Windows Phone 7**, 2013. Disponível em: <<http://www.windowsphone.com/pt-br/how-to/wp7/basics/update-history>>. Acesso em: 21 jun. 2013.

ZEMEL, T. **Web Design Responsivo Páginas adaptáveis para todos os dispositivos.**

São Paulo: Casa do Código, 2012.

APÊNDICES

Apêndice A – Diagrama entidade relacionamento da Base de Dados

