

SISTEMA *MOBILE* DE *DELIVERY* DE COMIDA ONLINE

Terra, Ana Paula Cabral; Pereira, Frederico Novack Amaral; Ribeiro, Lucas Fonseca (1); Carvalho, Marcos Alberto de (2)

(1) Acadêmicos do Curso de Bacharelado em Ciência da Computação da UNIFENAS; (2) Orientador

RESUMO:

Os aplicativos para sistemas mobiles, assim como as WebApps, estão cada vez mais populares e se tornaram muito importantes nos últimos anos no cotidiano das pessoas, passando a terem um papel de suma importância na vida de seus usuários. Sendo assim, este trabalho tem como principal finalidade o desenvolvimento de um aplicativo mobile para as plataformas Android e IOS, a partir de um website já existente, chamado Cardápio Alfenas. Isso mostrará, apesar das várias funcionalidades que a WebApp possui, o referido oferece aos seus usuários diversas funcionalidades de forma rápida e prática na palma das suas mãos. Um dos principais objetivos na elaboração do trabalho é que o mesmo possa ser feito com tecnologias de desenvolvimento móvel atuais, que possuam características indispensáveis nos dias atuais, como responsividade, facilidade e agilidade.

PALAVRAS-CHAVE: aplicativos mobile, webapps.

ABSTRACT:

The applications for mobiles systems, as well as WebApps, are increasingly popular and have become very important in recent years in daily life, going to have a role of paramount importance in the life of the users. Therefore, this work has as main purpose of the development of a mobile application for Android and iOS platforms from an already existing website called Cardapio

Alfenas, showing that despite many features that WebApp has, a mobile application offers to the user users several features, quickly and conveniently, in the palm of their hands. One of the main objectives in the development of the work is that it has do be done with current mobile development technologies, which have essential characteristics today, as responsivity, facility and agility.

KEYWORDS: mobiles applications, webapps.

1. INTRODUÇÃO

1.1. Visão

A informática está, a cada dia, mais presente na vida do ser humano, seja em simples tarefas do dia-a-dia como utilizar o micro-ondas, a máquina de lavar ou em tarefas mais complexas, como a utilização de softwares para a descoberta da cura de doenças. E, com ela, ocorreu também a popularização dos dispositivos móveis que, diferentes dos de antigamente, possuem diversas finalidades além de apenas realizar ligações.

Segundo Lee, Shineider e Schell (2005), hoje, dispositivos que são muitas vezes mais rápidos, menores e mais poderosos que aqueles antigos equipamentos podem ser transportados confortavelmente na palma da de seus usuários. Com a contínua miniaturização dos chips e dispositivos existentes, dispositivos ainda menores serão fabricados no futuro.

Mobilidade pode ser definida como a capacidade de poder se deslocar ou ser deslocado facilmente. No contexto de computação móvel, mobilidade se refere ao uso pelas pessoas de dispositivos móveis portáteis funcionalmente poderosos que ofereçam a capacidade de realizar facilmente um conjunto de funções de aplicação, sendo também capazes de conectar-se, obter dados e fornecê-los a outros usuários, aplicações e sistemas (LEE; SHINEIDER; SCHELL, 2005, p. 1).

E a mobilidade é um dos maiores motivos pelo qual, no cenário atual, existe uma grande adaptação de sistemas web para as plataformas mobiles existentes. Outros motivos que também impulsionam essa mudança são a praticidade e a comodidade perante os usuários. Sistemas que, hoje, não se adaptam às novas tecnologias começam a ficar defasados e, conseqüentemente, perdendo usuários. Webapps que estão em fase de desenvolvimento encontram-se no momento ideal para a criação e adaptação mobile para as múltiplas plataformas hoje existentes. O essencial é desenvolver esses webapps para aplicações móveis com toda a qualidade e praticidade que o exigente mundo mobile requisita.

1.2. Justificativa

Nos últimos anos, com a popularização dos dispositivos móveis no Brasil, vem ocorrendo um grande aumento na compra e utilização dos mesmos, sejam eles para os mais diversos fins, deixando de lado as WebApps. E, com essa grande demanda, cada vez mais se faz necessária a melhoria das tecnologias empregadas nos mesmos, sendo elas tanto de hardware como de software.

Conforme o Instituto Brasileiro de Geografia Estatística (IBGE) e o Ministério de Desenvolvimento, Indústria e Comércio Exterior (MDIC), de 2003 a 2013, há um crescimento visível na produção, venda e importação de telefones celulares no Brasil. Como mostra na Tabela 1, no ano de 2003 foram produzidos cerca de 29,3 milhões de unidades de celulares no Brasil, enquanto, dez anos depois, em 2013, produziu-se mais do que o dobro, cerca de 62,5 milhões de unidades.

Tabela 1 - Produção, Exportação e Importação de celulares no Brasil 2003-2013

Milhões de unidades	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
Produção (1)	29,3	42,9	64,3	61,7	68,4	68,3	55,8	57,6	60,8	57,2	62,5
Produção vendida(1)	27,3	41,7	63,4	58,6	66,2	69,8	60,9	61,9	61,1	57,2	61,5
Exportação (2)	11,3	8,9	32,9	32,0	22,0	23,9	15,8	12,6	6,9	3,4	1,9
Importação (2)	1,4	1,3	4,5	3,0	3,9	6,2	3,3	5,8	13,8	15,6	10,2

(1) IBGE PIA; (2) MDIC.

Fonte: <http://www.teleco.com.br/celprod.asp>

1.3. Objetivos

Este trabalho buscou apresentar as tecnologias atuais para o desenvolvimento de aplicações *mobile* nas plataformas Android e iOS. E a partir dessas tecnologias desenvolveu-se um aplicativo mobile para cada plataforma.

1.4. Estrutura do Trabalho

O presente trabalho é composto de cinco capítulos. O primeiro trata de assuntos abordados como a visão da introdução, justificativa, objetivos gerais e específicos e a hipótese deste trabalho; o capítulo dois relata o referencial teórico; já o capítulo três apresenta o material e métodos utilizados na elaboração do projeto; no capítulo quatro são apresentados os resultados e discussão em relação ao trabalho elaborado e, finalmente, no capítulo cinco é relatado a conclusão em relação a este trabalho.

2. REFERENCIAL TEÓRICO

2.1. Quem está se tornando móvel?

A conveniência e a poderosa funcionalidade oferecidas pelos dispositivos móveis, tais como telefones celulares, tiveram como resultado muitas pessoas já terem optado pelo uso da mobilidade. Dois dos maiores grupos de pessoas que utilizam dispositivos móveis são os profissionais e os consumidores (LEE; SHINEIDER; SCHELL, 2005, p. 10).

Em relação aos profissionais, nada mais é do que a utilização de dispositivos móveis pelas empresas, para a melhoria dos serviços prestados aos clientes e, também, para a geração de lucros futuros. Nos dias atuais, a migração das empresas para a área móvel tornou-se uma questão crucial.

De acordo com Lee, Shineider e Schell (2005), existe um grande número de consumidores que utilizam dispositivos e soluções móveis e, entre eles, existem dois tipos mais importantes: os jovens e os adultos. Com relação aos consumidores jovens, pode-se dizer que eles utilizam de ferramentas móveis para a comunicação, entretenimento e também para finalidades educativas. Já os adultos utilizam as aplicações para o mesmo fim, porém pode-se notar um uso bem menor de troca de mensagens de texto.

A capacidade de se conectar e tornar as pessoas móveis têm consequências sutis e extraordinárias na maneira como interagimos e com tecnologias existentes (LEE; SHINEIDER; SCHELL, 2005, p. 19).

2.2. Intel XDK

A Intel XDK é um ambiente de desenvolvimento multiplataforma com um conjunto de ferramentas para desenvolvedores de software que permite desenvolver, testar, visualizar e implantar aplicativos web e aplicativos híbridos em HTML5 (INGLÊS; INTEL, 2015, tradução nossa).

Diferente de outras plataformas para desenvolvimento móvel, a Intel XDK tem uma instalação muito mais fácil. Enquanto os seus concorrentes se veem obrigados a fazer uma configuração e adição de bibliotecas, essa plataforma só necessita de uma simples instalação.

E nela há a possibilidade de se desenvolver para qualquer plataforma, já que a compilação pode ser feita nas nuvens, fazendo disso uma novidade comparada a outras plataformas de desenvolvimento existentes no mercado atualmente.

A Intel XDK também possui um editor gráfico embutido, que é o que faz muita falta em todas as plataformas HTML5 para dispositivos móveis. É realmente lógico dar suporte a componentes HTML em editores *WYSIWYG* (*What You See Is What You Get*). Ele também suporta *frameworks* como Bootstrap e jQuery Mobile. Esses componentes de interface do usuário permitem a criação rápida de interfaces do seu aplicativo. (INGLÊS; INTEL, 2015, tradução nossa).

Outra funcionalidade bastante importante da Intel XDK é o seu emulador de dispositivos móveis. Ao criar o aplicativo, podem-se testar suas funcionalidades e capacidades, podendo traçar um perfil real da performance dele sem precisar instalar nada mais.

2.3. HTML 5

HTML 5 é uma linguagem para marcação de hipertexto, em inglês *HyperText Markup Language*, utilizada para criar sites.

Segundo Silva (2011), pode-se resumir hipertexto como todo o conteúdo inserido em um documento para a web e que tem como principal característica a possibilidade de se interligar a outros documentos da web. O que torna possível a construção de hipertextos são os links, presentes nas páginas de todos os sites que se visita quando a internet é utilizada.

A web foi inventada em 1992 por Sir Tim Berners-Lee. Atualmente Tim é diretor do World Wide Web Consortium (W3C), pesquisador sênior do Laboratório da Ciência da Computação e Inteligência Artificial (CSAIL) do Instituto de Tecnologia de Massachusetts (MIT) e professor de Ciência da Computação na Universidade de Southampton, na Inglaterra. Tim Berners-Lee trabalhava na Seção de Computação da Organização Europeia de Pesquisa Nuclear (CERN), com sede em Genebra, na Suíça, quando iniciou pesquisas visando a descobrir um método que possibilitasse aos cientistas do mundo inteiro compartilhar eletronicamente seus textos e pesquisas e que tivesse a funcionalidade de interligar os documentos. Estavam criadas as noções web e de links como são conhecidas atualmente. No ano de 1990, Tim criou um protótipo de um navegador para rodar em computadores da NeXT, uma companhia fundada em 1995 por Steve Jobs. Tim Berners-Lee acreditava que seria possível interligar hipertextos em computadores diferentes com uso de links globais, também chamados de hiperlinks. Ele desenvolveu um software próprio e um protocolo para recuperar hipertextos, denominado HTTP. O formato de texto que criou para o HTTP foi chamado de HTML. Tim tomou como base para criação da HTML a especificação SGML, que é um método internacionalmente reconhecido e aceito, contendo normas gerais para a criação de linguagens de marcação. A marcação para hiperlinks conduzindo a documentos que não estivessem em um mesmo computador obviamente não constava das normas para SGML e foi inventada por Tim, demonstrada pela primeira vez em 1990, em uma estação de trabalho NeXT, nos laboratórios da CERN. Estava criado o embrião da World Wide Web, bem como a primeira versão da linguagem HTML para a marcação de hipertextos (SILVA, 2011, p. 21).

2.4. JavaScript

A linguagem HTML tem como função principal estruturar uma página web, não se devendo empregá-la para adicionar estilos ou apresentação visual aos elementos que constituem a página. Essa tarefa é função das folhas de estilo em cascata (CSS).

O HTML também não possui funcionalidades que possam ou permitam adicionar interatividade avançada nas páginas. Isso será função das linguagens de programação.

JavaScript foi criada pela Netscape em parceria com a Sun Microsystems, com a finalidade de fornecer um meio de adicionar interatividade a uma página web. A primeira versão, denominada JavaScript 1.0, foi lançada em 1995 e implementada em março de 1996 no navegador Netscape Navigator 2.0 quando o mercado era dominado pela Netscape (SILVA, 2011, p. 23).

Segundo Silva (2011, p. 23): “Não há como fazer funcionar um formulário HTML com o uso de elementos HTML. A HTML limita-se a criar os rótulos e campos de um formulário para serem preenchidos pelo usuário e nada mais. Com HTML, não conseguimos processar os dados nem mesmo enviá-los ao servidor ou a outra máquina qualquer. Para cumprir tais tarefas, é necessário utilizar um programa que consiga manipular e processar os dados. Entre as várias linguagens destinadas a adicionar e processar dados em páginas web, destacam-se PHP, ASP, Java, Ruby, Python e ColdFusion, entre outras. ”

As linguagens de programação, como as citadas anteriormente, foram desenvolvidas para rodar no lado do servidor, isto é, dependem de uma máquina remota onde estão hospedadas as funcionalidades capazes de interpretar e fazer funcionar os programas. JavaScript é uma linguagem desenvolvida para rodar no lado do cliente, ou seja, a interpretação e o funcionamento da linguagem dependem de funcionalidades hospedadas no navegador do usuário. Isso é possível porque existe um interpretador JavaScript hospedado no navegador (SILVA, 2011, p. 23).

2.5. Ruby On Rails

Um framework nada mais é do que uma coleção de bibliotecas e ferramentas que são usadas para facilitar o desenvolvimento de alguma aplicação. Um bom framework, normalmente, fornece uma base, porém completa, infraestrutura para construir toda a aplicação.

O Rails é um framework de aplicações web escritas na linguagem Ruby. Ele é projetado para tornar a programação de aplicações web mais fácil, através de suposições de que cada desenvolvedor precisará para começar. Isso permite que se escreva muito mais código enquanto se faz do que muitas outras linguagens de programação ou frameworks. Desenvolvedores mais experientes em Rails também relatam que isso torna o desenvolvimento de aplicações web muito mais divertido (RailsGuides, [ca. 2011], tradução nossa).

2.6. Padrão MVC

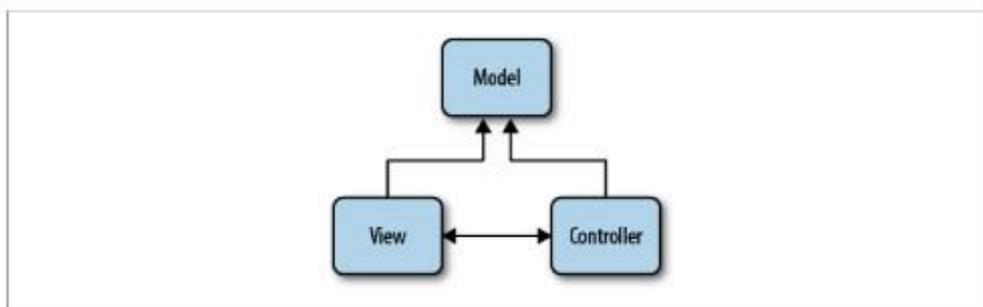
Inventado para ser utilizado na linguagem Smalltalk por Trygve Reenskaug nos anos 70, o padrão MVC foi se popularizar na web em 2003 com o surgimento do Ruby on Rails (PALERMO, et. al., 2010, p. 7).

Segundo Gamma et al. (2000, p. 20): “O MVC é composto por três tipos de objetos. O modelo é o objeto de aplicação, a visão é a apresentação na tela; e o controlador define a maneira como a interface do usuário reage às suas entradas. Antes do MVC, os projetos de interface para o usuário tendiam em agrupar esses objetos. Com o MVC, aumentaram-se a flexibilidade e a reutilização. “

A camada de modelo faz todo o trabalho da aplicação, enquanto a camada de visão encarrega-se da interface do usuário e a camada de controle executa todas as ações da aplicação e faz a troca de mensagens entre o modelo e a visão. Para cada objeto de visão, existe um objeto de controle, mas, para cada modelo, podem existir vários objetos de visão e controle (KRASNER; POPE, 1988, tradução nossa).

A FIG. 1 mostra como o padrão MVC trabalha e se comunica entre as camadas modelo, visão e controle.

Figura 1 - The MVC Architecture



Fonte: Chadwick, Snyder, Panda (2012)

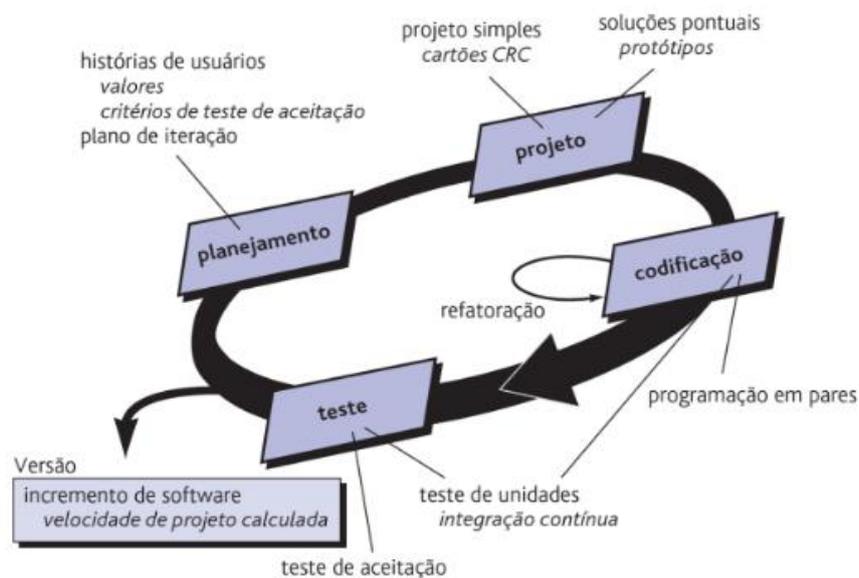
2.7. Extreme Programming (XP)

Segundo Sommerville (1995), um método de desenvolvimento de software é um conjunto de atividades que auxiliam na sua produção. O resultado dessas atividades é um produto que reflete a forma como todo processo foi conduzido.

No projeto, utilizou-se uma das abordagens mais amplamente usadas no desenvolvimento de software ágil, a Extreme Programming.

De acordo com Pressman (2011, p. 72): “a Extreme Programming emprega a metodologia orientada a objetos como seu paradigma de desenvolvimento e envolve um conjunto de regras e práticas constantes no contexto de quatro atividades metodológicas: planejamento, projeto, codificação e testes. A FIG. 2 ilustra o processo XP e destaca alguns conceitos e tarefas-chave associados a cada uma das atividades metodológicas.”

Figura 2 - O Processo de Extreme Programming (XP).



Fonte: Pressman (2011)

2. MATERIAL E MÉTODOS

3.1. Arquitetura Utilizada

- Padrão MVC;
- Extreme Programming (XP).

3.2. Etapas do Desenvolvimento

O projeto foi dividido nas seguintes etapas:

1. Estudo das tecnologias referentes e, já citadas no Referencial Teórico, ao desenvolvimento de dispositivos móveis;
2. Definição das tecnologias que melhor se adaptaram ao problema para serem utilizadas no projeto;
3. Desenvolvimento do projeto, utilizando as tecnologias selecionadas;
4. Simulação, teste e análise da aplicação.

3.3. Desenvolvimento

Este trabalho consistiu no desenvolvimento de um aplicativo mobile do website Cardápio Alfenas, situado no domínio <http://www.cardapioalfenas.com.br>, no qual já se encontrava ativo e com acessos diários. A Figura 3 abaixo mostra a tela principal do website. O website a ser utilizado como modelo para o aplicativo mobile permite que o usuário, ao utilizá-lo, procure estabelecimentos comerciais e alimentos de interesse no município de Alfenas em Minas Gerais, seja para apenas simplesmente visualizar o cardápio de algum estabelecimento em questão, buscar endereços ou telefones de estabelecimentos ou a sua maior finalidade: realizar pedidos de

alimentos via website, caso o estabelecimento esteja logado e em operação no momento da visualização.

Figura 3 - Tela Inicial do Cardápio Alfenas.



Fonte: Print screen do website www.cardapioalfenas.com.br

Para que fosse possível desenvolver o aplicativo com todas as funções que o website possui, porém de uma forma mais inovadora e que se fosse capaz de produzi-lo em várias plataformas comumente utilizadas pelas pessoas, como Android, iOS e Windows Phone, utilizou-se a plataforma de desenvolvimento mobile da Intel, a Intel XDK, onde o desenvolvedor programa em HTML 5 e JavaScript e o projeto é compilado pelo Apache Cordova.

A IDE da Intel mostrou-se muito útil para gerar a interface de aplicativos simples, sendo bem fácil manipular componentes já prontos, como botões, labels, dropbox, menus interativos e outros. Como o objetivo deste trabalho era transformar uma aplicação web em uma aplicação mobile, a interface do nosso

aplicativo é semelhante ao site, para que o usuário possa se sentir confortável, utilizando uma interface já conhecida e sua *user xperience* já praticada.

Para que fosse possível tal semelhança de visual, optou-se por desenvolver todos os componentes gráficos via código HTML, e assim foi feito o aproveitamento das classes e elementos do CSS do próprio site a ser transformado em mobile, o que tornou o trabalho gráfico um pouco mais fácil. Algumas alterações no CSS tiveram que ser feitas, já que a questão de responsividade foi tratada muitas vezes pelo próprio Cordova.

As funções em JavaScript do projeto enviam requisições para o servidor do site, que foi feito em Ruby on Rails, com Puma e com o proxy reverso NGINX, sendo todo esse projeto em back-end, estando em um repositório do Docker. Todas as operações de webservice utilizadas no projeto do aplicativo possuem o mesmo embasamento lógico e operacional do website, o que tornou a criação da webservice bem prática e rápida, sendo a única alteração nessas funções o modo de retorno. Enquanto o servidor devolve para o cliente web um HTML, o servidor deve retornar para a aplicação uma linguagem demarcada como o XML ou JSON. Optou-se por utilizar o JSON, pois sua visualização de dados e o tratamento são mais simples que o XML. Então, mesmo que as funções responsáveis por usuários web do servidor façam as mesmas operações lógicas, teve-se que criar novas funções que realizavam a mesma operação, porém retornando em um JSON.

O JSON respondido pelo servidor é tratado pelo Javascript do aplicativo, fazendo, assim, com que o aplicativo renderize a página de acordo com o retorno do servidor e a operação realizada. Todas as funções tiveram o mesmo processo.

As chamadas para o servidor para realizar qualquer operação com o banco de dados eram feitas em protocolo RESTful. Utilizava-se o host das chamadas o endereço do Cardápio Alfenas próprio para o webservice (www.cardapioalfenas.com.br/web-service).

O banco de dados era manipulado inteiramente pelo servidor. Qualquer operação de consulta, alteração, manipulação, adição ou deleção de dados deveria ser respondida pelo servidor. O app então se comunica com o webservice do servidor por meio de protocolo REST e o servidor avalia a

veracidade e qualidade dos dados e altera dentro do banco. O design do banco então nunca era reconhecido pelo aplicativo, mas sim pelo webservice desenvolvido em REST.

4. RESULTADOS E DISCUSSÃO

Foram encontrados vários desafios ao longo do desenvolvimento do trabalho, como, por exemplo, a realização do login do Facebook por meio do aplicativo, qual API do Facebook utilizar, ou, até mesmo, a publicação da aplicação na loja App Store, mas com um árduo esforço e muito trabalho, essas dificuldades foram ultrapassadas.

Em relação à ferramenta utilizada, a Intel XDK, ela se mostrou muito eficaz para produzir aplicativos simples e que requerem pouco ou nenhum uso de dados. Para aplicações mais complexas como o sistema do Cardápio Alfenas, não só o Intel XDK como o próprio Cordova deixaram muito a desejar.

O código desenvolvido no aplicativo não pode ser modularizado, pode apenas ser desenvolvido em um arquivo, contendo todas as funções em Javascript e outro arquivo único contendo todas as <div> que representam as telas. A impossibilidade de manutenção mais profunda no projeto desenvolvido para iOS fez com que a publicação do aplicativo na App Store da Apple fosse quase impossível.

A solução para tal problema em sistemas iOS foi o de incluir no arquivo info.plist (arquivo XML responsável pelas configurações e informações de extrema importância no projeto) uma tag responsável por permitir que o aplicativo faça chamada a servidores que não possuem uma conexão segura. A Intel XDK, infelizmente, não possui essa possibilidade, já que, para incluir tal informação, é necessário alterar as configurações de compilação do Cordova. É impossível tal ato já que, para compilar um projeto do Intel XDK, o projeto inteiro é enviado aos servidores da Intel, onde o Cordova está compilado e retornando para o desenvolvedor.

5. CONCLUSÃO

O presente trabalho demonstrou o poder da tecnologia mobile nos dias atuais aliado ao cotidiano das pessoas, assim ajudando-as cada vez mais em suas tarefas e atividades. Faz-se necessário, atualmente, o uso de celulares móveis para uma boa parte das atividades do dia-a-dia e isso se torna ainda mais fácil e menos estressante quando essa atividade já possui um aplicativo mobile responsivo e criado com tecnologias atuais, como é o caso da aplicação desenvolvida.

Notou-se que 57,28% do público que utiliza o sistema do Cardápio Alfenas são usuários mobile. Provando que existe um engajamento maior de usuários mobile com o sistema. Apesar da tecnologia usada (Intel XDK) não ter sido eficiente na produção de um aplicativo para iOS, a versão em Android conseguiu ser publicada e teve um alto rendimento para o site. Para a versão em iOS recomenda-se o desenvolvimento em linguagem nativa, como Objective-C ou Swift.

6. REFERÊNCIAS

CHADWICK, Jess. SNYDER, Todd. PANDA, Hrusikesh. **ASP.NET MVC 4**. 1 ed. Sebastopol: O'Reilly, 2012.

GAMMA, Erich et al. **Padrões de Projeto: Soluções reutilizáveis de software Orientado a Objetos**. Porto Alegre: Bookman, 2000.

INTEL. **Introduction to the Intel XDK**. 29 out. 2015. Disponível em: <https://software.intel.com/pt-br/xdk/docs/intel-xdk-introduction> Acesso em: 14 nov. 2015

KRASNER, Glenn; POPE, Stephen. **A Description of the Model-View-Controller User Interface Paradigm in the Smalltalk-80 System**. ParcPlace Systems, 1988.

LEE, Valentino; SCHNEIDER, Heather; SCHELL, Robert. **Aplicações Móveis**. 1. ed. São Paulo: Makron Books, 2005.

PALERMO, Jeffrey et al. **APS.NET MVC 2 In Action**. Manning Publications – Stamford, 2010.

PRESSMAN, Roger S. **Engenharia de Software**. 7^o ed. São Paulo: McGraw-Hill, 2011.

RailsGuides. **Getting Started with Rails**. Disponível em:
http://guides.rubyonrails.org/getting_started.html Acesso em: 15 fev. 2016

SILVA, Maurício Samy. **HTML5**. 1^a ed. São Paulo: Novatec, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 5^a ed. Boston: PEARSON: Addison-Wesley, 1995.