

**FUNDAÇÃO UNIVERSIDADE DO TOCANTINS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**

**Cristhiann Tallys Rodrigues Martins**

**Desenvolvimento de ferramenta para auxiliar no desenvolvimento  
de aplicações adaptativas web mobile**

**Palmas/TO, 2015**

**FUNDAÇÃO UNIVERSIDADE DO TOCANTINS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**

**Cristhiann Tallys Rodrigues Martins**

**Desenvolvimento de ferramenta para auxiliar no desenvolvimento  
de aplicações adaptativas web mobile**

Trabalho de Conclusão de Curso de  
Sistemas de Informação da Fundação  
Universidade do Tocantins, apresentado  
como parte dos requisitos para obtenção do  
título de Bacharel em Sistemas de  
Informação.

Orientador: M.e Alex Coelho.

**Palmas-TO, 2015**

## DEDICATÓRIA

Primeiro a Deus por todos os ensinamentos e bênçãos alcançadas e por poder estar onde estou hoje. A toda minha família e a todos meus amigos que sempre me apoiaram e sempre se empenharam a jamais me deixar desistir e sempre buscar o melhor.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus pela graça, força e perseverança que me foram concedidas para superar todas as dificuldades e obstáculos que apareceram para enfim conseguir concluir o curso.

A meus pais, que mesmo passando por algumas dificuldades sempre me deram apoio e me auxiliaram em tudo que fosse preciso para ter o sucesso em toda carreira.

Agradeço também a todos meus amigos de infância por estarem comigo até hoje e por me apoiarem em todas minhas decisões. A meus amigos de faculdade que a cada trabalho e projeto foram de grande ajuda para que não somente o meu, mas o sucesso de todos fosse alcançado. Não poderia deixar de agradecer especialmente, do fundo do meu coração a Agnêlio Alves, Marinete Alves e Mauricio Frantz, pois serão amigos que levarei eternamente.

Agradeço aos professores que foram espelho e exemplos de profissionais e também de pessoa, agradeço por todo o tempo investido, toda a paciência e insistência para fazer com que o aprendizado fosse alcançado, lembrar-me-ei de todos com muito carinho e admiração. Agradeço especialmente ao Mestre Alex Coelho que me acompanhou desde o início, acreditou em meu potencial e jamais me deixou desamparado, mais que um professor, já o considero um amigo.

*“Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar a onde a maioria não chega, faça o que a maioria não faz. ”*

Bill Gates

## RESUMO

Como afirma em sua pesquisa, o site da Tudo Celular (2015) que em 2014 mais de 1,2 bilhões de smartphones foram vendidos, tendo-se nesse montante aparelhos com diferentes marcas, diferentes tamanhos e sistemas. Tendo isso, a necessidade de criação de aplicações adaptáveis aos diversos sistemas e tamanhos torna-se cada vez mais necessário. Dentro deste contexto, este trabalho tem por objetivo desenvolver uma ferramenta web que o usuário possa inserir um código HTML com as tags dos elementos que componham sua aplicação. O sistema será capaz de identificar os elementos existentes na aplicação, gerar três arquivos CSS onde encontra-se os layouts de cada elemento identificado padronizando de acordo com as recomendações da Apple, Android, e Windows Phone, alterar o código HTML para que seja possível a adaptação ao sistema, permitir download dos arquivos HTML5 e CSS's gerados. Com esses passos o código digitado que anteriormente era somente HTML, sem nenhum layout, agora é possível ver a diferença de layout de acordo com o sistema a ser executada.

**Palavras-chaves:** *Adaptabilidade; Android; Apple; Desenvolvimento web; Windows Phone.*

## ABSTRACT

As stated in their research, the All Mobile site (2015) that in 2014 more than 1.2 billion smartphones were sold, having this amount sets with different brands, different sizes and systems. With this, the need to create adaptable applications to the various systems and sizes becomes increasingly necessary. Within this context, this paper aims to develop a web tool that the user can enter an HTML code with the tags of the elements that compose your application. The system will be able to identify the existing elements in the application, create three files where CSS is the layouts of each element identified by standardizing according to the recommendations of Apple, Android, and Windows Phone, change the HTML code to make it possible to adapt the system, allow download of HTML 5 and CSS files generated's. With these steps the typed code that was previously only HTML without any layout, it is now possible to see the layout of difference in accordance with the system running.

**Keywords:** *Adaptability; Android; Apple; Web development; Windows Phone.*

## Sumário

1 INTRODUÇÃO.....	9
2 REFERENCIAL TEÓRICO .....	12
2.1 MOBILE FIRST .....	12
2.2 SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS.....	13
2.2.1 ANDROID .....	13
2.2.2 IOS .....	15
2.2.3 WINDOWS PHONE.....	17
2.3 ASPECTOS DE INTERFACE .....	19
2.3.1 HOMEM-MÁQUINA.....	21
2.3.1.1 INTERFACE.....	21
2.3.2 DISPOSITIVOS MÓVEIS .....	22
2.3.2.1 INTERFACE INTELIGENTE .....	22
2.3.2.2 INTERFACE ADAPTATIVA .....	23
2.3.2.3 INTERFACE RESPONSIVA .....	24
2.4 AVALIAÇÕES DOS APLICATIVOS PELAS APPS STORES.....	24
2.4.1 MOTIVOS PARA REPROVAÇÃO.....	27
2.5 HTML5 .....	30
2.5.1 O QUE MUDA COM O HTML5?.....	31
3 METODOLOGIA.....	33
4 RESULTADOS .....	35
4.1 O PROBLEMA.....	35
4.2 ARQUITETURA .....	36
4.3 APPWEB/HTML5.....	37
4.3.1 VERIFICAÇÃO DE APLICATIVOS.....	37
4.3.1.1 ANDROID .....	37
4.3.1.2 WINDOWS PHONE.....	38
4.3.1.3 IO's.....	39
4.4 DESENVOLVIMENTO .....	40
4.4.1 TRABALHOS RELACIONADOS.....	40
4.4.2 PÁGINA HTML5.....	41
4.4.3 IDENTIFICAÇÃO DA PLATAFORMA .....	41



4.4.4 MAPEAMENTO CSS.....	43
4.4.4.1 HTML5 .....	43
4.4.4.2 BOTÕES.....	44
4.4.4.3 INPUT.....	45
4.4.4.2 SWITCH.....	46
4.4.5 TESTE DE VERIFICAÇÃO DE PLATAFORMA .....	48
4.5 APLICAÇÃO DE INSERÇÃO DOS PADRÕES .....	49
4.5.1 IDENTIFICAÇÃO DOS COMPONENTES .....	50
4.5.2 GERAÇÃO DOS ARQUIVOS CSS.....	50
4.5.3 ALTERAÇÃO DO CÓDIGO HTML.....	51
4.5.4 DOWNLOAD DE ARQUIVOS.....	51
5. CONCLUSÃO .....	53
REFERÊNCIAS .....	55

# 1 INTRODUÇÃO

Com esse crescimento da quantidade de smartphones, o estudo de interfaces e de interação com o usuário demonstra ser de grande importância, tendo maior relevância agora do que em nos tempos em que se usava só e somente o computador de mesa. Conforme afirma Francisco Ribeiro (2014)

A usabilidade hoje em dia é um fator de sucesso para um software, ignorá-la pode significar o fracasso de um projeto, os desenvolvedores normalmente não se preocupam com esta característica. A análise da usabilidade de um sistema tem que ser parte integrante de qualquer ciclo de desenvolvimento, a proposta deste trabalho é realizar esta análise em um sistema desenvolvido para dispositivos móveis, usando técnicas e métodos de avaliação de interface, propondo melhorias ao sistema de acordo com os resultados obtidos nos testes.

Outros dados importantes são que um em cada cinco brasileiros costumam comentar, divulgar ou compartilhar experiências pela internet no exato momento em que elas acontecem (DATA FOLHA, 2014).

O tema “dispositivos móveis” sem dúvida alguma, hoje em dia, para quem estuda a área de tecnologia da informação, é um assunto que se deve ter uma maior atenção, pois seu crescimento é inegável e sendo uma área promissora. A quantidade de dispositivos móveis cresce cada dia mais e já representam a maioria dos sistemas computadorizados atualmente. Por isso torna-se cada vez mais importante criarmos aplicativos voltados para esse segmento (MEDEIROS, SD).

Mas não se deve preocupar somente em criar a aplicação, mas sim em como será utilizada, como será a melhor forma que o usuário poderá utilizar cada funcionalidade. A interface com o usuário tem importância fundamental em sistemas interativos, possibilitando a comunicação entre o usuário e o sistema, de modo que, quanto maior for o nível de usabilidade da interface, mais fácil será a usabilidade (SOUZA, SPINOLA, 2006).

No desenvolvimento de um aplicativo para dispositivos móveis, é necessário não somente criar a aplicação de qualquer forma, ou seja, é preciso o estudo detalhado de seu funcionamento e de sua usabilidade. Dentre as diversas tecnologias emergentes se encontra a HTML 5, sendo já concebida com elementos e características relevantes para sua adaptação aos dispositivos móveis. Deste

modo, uma questão deve ser respondida: quais são as maiores preocupações ao se desenvolver para mobile?

A adaptabilidade é possível a partir de uma combinação do HTML, CSS e *JavaScript*. Dessa forma o site desenvolvido será adaptado a qualquer tamanho de tela em que for executado.

La (2012) afirma que é possível criar um layout responsivo em três passos:

1. A *meta tag* “*viewport*”;
2. A estrutura HTML;
3. As *media queries*

No HTML, *meta tag* são as formas de fornecer para todos os tipos de sistemas todas as informações cabíveis para o conhecimento, e cada sistema processa as *meta tags* que entenda (Oliveira; Araújo, 2012).

O *viewport* tem por definição o tamanho disponível para exibição do site, ou seja, a área no qual é permitida que seja utilizado para o desenvolvimento do site. Define também ao navegador o tamanho de tela disponível para exibição do site.

Dentre os vários valores que a *viewport* pode receber, é recomendável utilizar “*width=device-width*”, informando ao navegador que o tamanho da *viewport* é igual ao tamanho da tela do dispositivo. O *initial-scale* especifica a renderização, ou seja, com o valor 1.0 a renderização do site não deverá conter nenhum zoom inicial (Silva, 2014).

Na estrutura HTML é o local onde se cria a estrutura básica do site, ou seja, os elementos que se ajustarão de acordo com o dispositivo, no qual por exemplo, pode-se criar um *layout* básico com cabeçalho de altura fixa em 180 *pixels*, conteúdo com largura de 600 *pixels*, e barra lateral de 300 *pixels* de largura.

As *media queries* são utilizadas para aplicar estilo CSS com as características do dispositivo que está a ser executado o navegador que tem acesso ao site em questão, ou seja, são as responsáveis pelas alterações no *layout* em aparelhos de diferentes tamanhos e resoluções, permitindo assim manter o mesmo código HTML, afirma SILVA (2014).

No desenvolvimento web uma das maiores preocupações está voltada ao design da aplicação, muitas vezes por não ter uma prática ou conhecimento de web design, o desenvolvedor pode ficar de certa forma desorientado. Tendo isso em vista, este trabalho tem por objetivos identificar qual a plataforma que a aplicação está a ser executada e qual o design correto de cada sistema, e dessa forma carrega o arquivo de configuração correto, e permitindo assim que o usuário faça o download dos arquivos criados pela aplicação.

Este trabalho foi desenvolvido para proporcionar ao desenvolvedor maior comodidade ao desenvolver aplicações web adaptativas, ou seja, desenvolver aplicação visando dispositivos móveis de diferentes sistemas.

## 2 REFERENCIAL TEÓRICO

Que o aumento do número de dispositivos móveis é crescente a cada dia está ficando concretizado mais e mais com o passar do tempo, mas antes de todos esses dados se tem questões a serem respondidas: o que é um dispositivo móvel? Quais são as características necessárias para que algum objeto tecnológico seja considerado móvel?

A Computação móvel permite que os usuários possam acessar livremente os serviços oferecidos por tais dispositivos, de forma independente de sua localização, requerendo assim suporte á mobilidade e existência de uma infraestrutura de comunicação sem a necessidade de estar conectado a nenhum tipo de fio, afirma Johnson.

Gavasso *et al.* (2006) afirmam que para ser considerado um computador de mão, os dispositivos devem ser constituídos de algumas características como funcionar sem cabos, exceto temporariamente (recarga elétrica ou sincronização com computadores de mesa), deve ser facilmente operado com uma mão, sem a necessidade de estar apoiado em uma mesa e ainda deve permitir a adição de novos aplicativos e suportar conexão à Internet.

Diante disso, a seguir serão expostos o elementos mais importantes dentro da relação de produção de aplicações para dispositivos móveis e considerações que devem ser trabalhadas para uma melhor relevancia no processo de desenvolvimento.

### 2.1 MOBILE FIRST

Devido a grande quantidade de *smatphones* em todo mundo, deu-se inicio a uma ideia de que poderia ser mais viável e até mesmo rentável criar aplicações ou sites primeiramente visando dispositivos móveis. Essa é a definição do mobile first, segundo Eis (2011).

Eis (2011) complementa ainda que, colocando os *mobiles* em primeiro lugar, uma série de vantagens serão alcançadas como: a utilização das técnicas permitidas pelos mobiles como acelerômetro, GPS, *multitouch*, giroscópio, e uma série de funções que os computadores *desktop* jamais permitiriam o usuários utilizarem. O foco, usabilidade e acessibilidade também são outras vantagens, pois

concentra-se nas ações essenciais, e preocupar-se com o layout do sistema é muito importante e de fato mais facilitado quando o assunto é desenvolvimento mobile.

## 2.2 SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS

Quando o assunto é “sistema para dispositivo móvel” é praticamente impossível não pensar nos três grandes do mercado: *Android*, *IO's* e *Windows Phone*. Cada um tem seu próprio modelo para aplicativos, suas próprias exigências quanto ao *layout* do sistema.

Cada sistema segue um diferente modelo de *design* para suas aplicações, alguns exigindo bastante das aplicações quanto a seu *layout* gerando dessa forma alterações no projeto quando se é desenvolvido para vários aplicativos que rodam em diferentes dispositivos e sistemas operacionais.

### 2.2.1 ANDROID

Hoje em dia, é o sistema que está presente em grande maioria dos dispositivos móveis do mundo. Uma estimativa recente conta que 84,70% dos aparelhos utilizam o sistema e com a incrível marca de um bilhão de usuários ativos. Por ser o sistema com maior ascensão, o Android a cada dia que passa atrai novos desenvolvedores, tendo hoje um total de 1,3 milhões de aplicativos na Google Play (HAMANN, 2014).

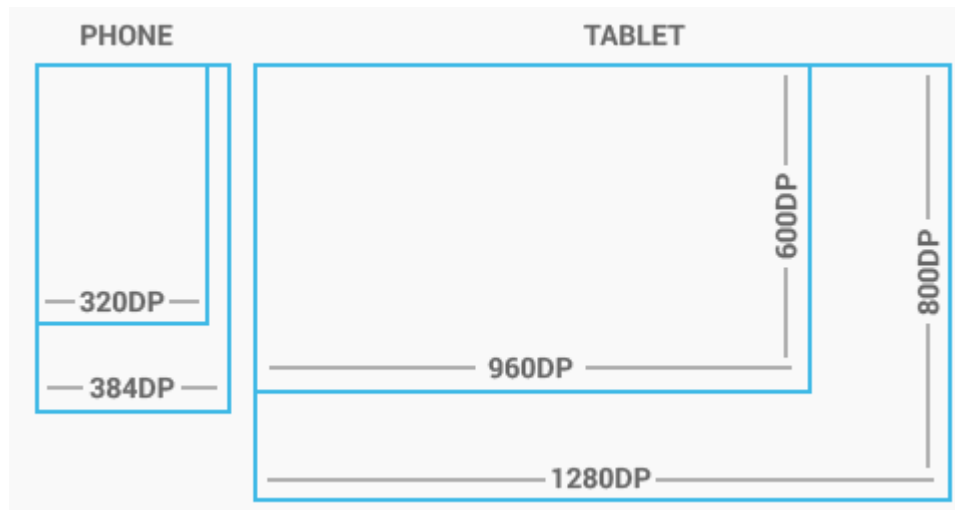
Em seu site oficial, o GOOGLE dá algumas recomendações para que os desenvolvedores criem aplicativos com os padrões exigidos pela empresa. Não necessariamente que esses padrões sejam obrigatórios, mas que sejam feitos projetos com um visual e uma usabilidade com padrões agradáveis para os usuários. Dentre tais recomendações pode-se citar:

- **Encante-me:** A superfície bonita, a animação cuidadosamente colocados, ou um efeito sonoro na hora certa é uma alegria para experimentar. Permitir que o usuário dê toques pessoais para se sentir em casa e no controle. Aprender as preferências dos usuários ao longo do tempo de utilização da aplicação.

- **Simplifique minha vida:** Utilize frases curtas, com palavras simples. Dê preferência às imagens. Dê palpites nas decisões do usuário.
- **Faça-me surpreendente:** Faça a aplicação mais fácil de aprender. Seja sutil ao levar as pessoas a fazerem as correções, usuário gostam de se sentir inteligentes quando utilizam os aplicativos.

Desenvolver para esse sistema pode ser complicado devido a grande quantidade de aparelhos de diferentes tamanhos existentes. Em seu site oficial, o Android oferece alguns exemplos de como se devem compor os ícones, layout's, cores e alguns outros elementos para que a aplicação seja bem desenvolvida.

A Figura 1 mostra desde o menor ao maior tamanho de tela existente tanto para *smartphones* quanto para *tablets* que utilizam o sistema.

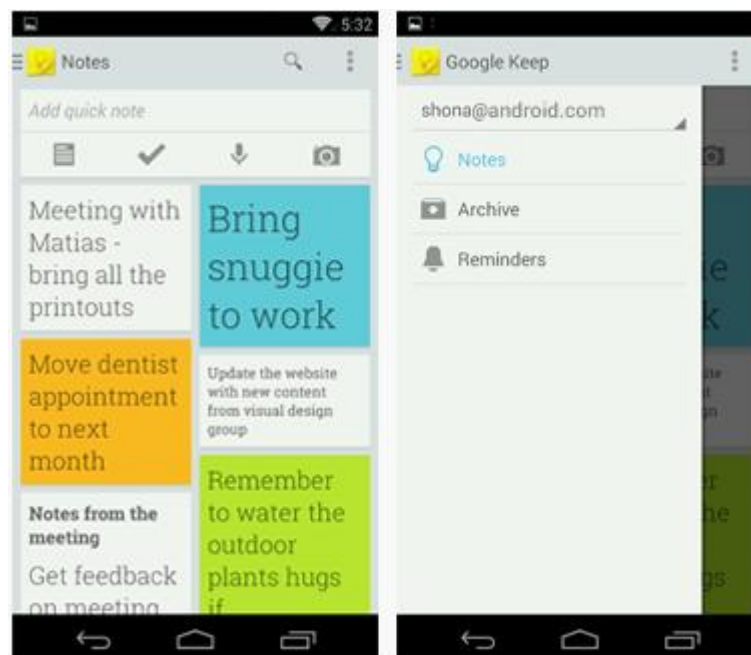


**Figura 1:** Métricas e Grades

**Fonte:** Developer Android

Ao criar uma aplicação, é importante a preocupação da adaptação do aplicativo para os diferentes tamanhos.

Na Figura 2 é apresentada a demonstração gráfica da utilização de alguns componentes disponíveis para desenvolvimento para Android.



**Figura 2:** Common App UI (Aplicativo Comum)

**Fonte:** Developer Android

O site oficial do Android apresenta uma típica aplicação para o sistema que utiliza barras de ação, navegação gaveta e área de conteúdo, sendo definidos como:

- **Barras de ação:** Essa barra é onde o usuário faz controle dos modos de exibições da aplicação, é o centro de comando para o aplicativo. Para a visual atual do Android, é uma das ações mais importantes para os projetos que estão sendo desenvolvidos;
- **Navegação gaveta:** Funciona como um menu do aplicativo, ou seja, quando a estrutura da aplicação é mais complexa a gaveta de navegação exibi as principais opções da navegação. A gaveta se expande a partir da borda esquerda da tela, sobrepondo a área de conteúdo;
- **Área de Conteúdo:** É o espaço onde o conteúdo do aplicativo é exibido.

Acima tem-se a explicação de como se dá o corpo de uma aplicação Androi, itens básicos para um aplicativo padrão.

## 2.2.2 IOS

Diferente do Android, o iOS aos poucos começa a perder espaço no cenário mundial de smartphones. Hoje em dia 11,07% dos aparelhos utilizam o sistema. Entretanto, demorará certo tempo para que perca sua posição de destaque para

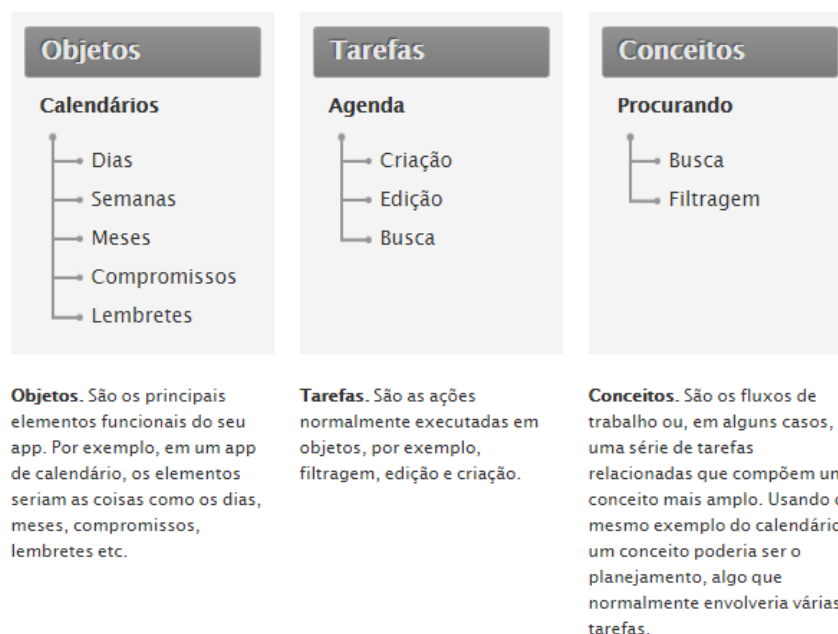


outros sistemas operacionais com menor uso que o iOS. Hoje existem 1,2 milhões de aplicativos para o sistema (HAMANN, 2014).

Levando em conta dimensões e versões do sistema, desenvolver para os dispositivos da Apple pode ser mais vantajoso do que desenvolver para os demais sistemas. Essa facilidade dá-se devido ao menor número de aparelhos e de versões de sistema em que se deve testar a aplicação e se adaptar.

Em seu site oficial, a Apple dá algumas dicas para desenvolver uma aplicação seguindo os critérios impostos pela empresa:

- **Coloque-se no lugar do usuário:** Pensar em tarefas simples, que o usuário faça no dia-a-dia e identifique como um app poderia facilitar tais tarefas;
- **Envolva-se com os usuários:** Convidar usuários que tenham boas ideias a participarem do projeto. Deixar que o usuário final faça parte do desenvolvimento pode aperfeiçoar o funcionamento do aplicativo;
- **Design para toque:** Neste tópico, a Apple convida o desenvolver a fazer uma análise melhor em um iPhone ou iPad, para entender melhor o diferencial dos dispositivos.
- **Mantenha o foco do seu app:** quando uma aplicação em iOS define e matem o foco na função principal, fica mais agradável e pratico de usar. A criação de listas ajudará a apresentar a interface organizada e focalizada. Tais considerações são demonstradas na Figura 3 apresentada a seguir.



**Figura 3:** Listas de objetos de uma aplicação iOS

**Fonte:** Developer iOS

Ao seguir as técnicas recomendadas, a aprovação da aplicação pela Apple é praticamente certa. Com os itens acima é verificado que para ser aprovado o aplicativo não precisa somente ter uma infinidade de itens, funcionalidades inovadoras, mas sim preocupar-se com a interação com o usuário.

### 2.2.3 WINDOWS PHONE

Vários sites de pesquisas relacionados a sistemas de dispositivos móveis ressaltam o crescimento do sistema Windows Phone no mercado mundial. Porém, ainda está longe dos grandes *players* do mercado. Dono de 2,50% dos aparelhos mundiais, em 2014, estão ativos cerca de 60 milhões de usuários [HAMANN, 2014].

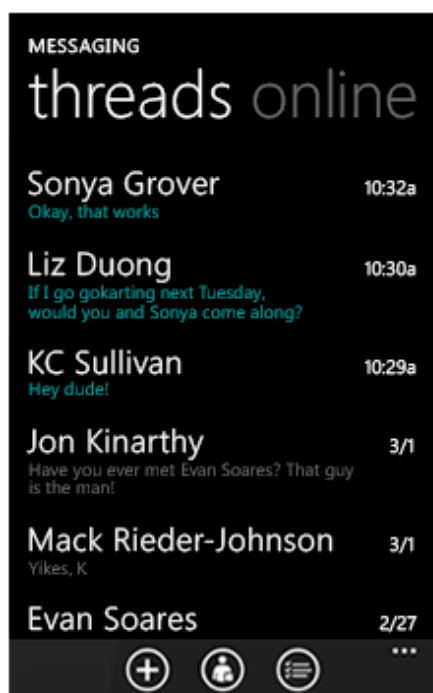
Devido a baixa quantidade de usuários, ainda é baixo o desenvolvimento de aplicações para a plataforma, existindo hoje algo em torno de 300 mil aplicativos para o sistema. Como os outros sistemas anteriormente citados, o Windows Phone também disponibiliza em seu site oficial, seus princípios de designer de aplicações, conforme pode ser constatado na Figura 4.



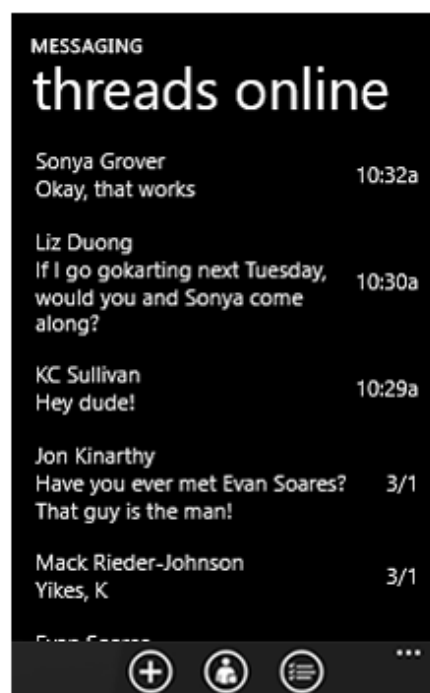
**Figura 4:** Conceito de Movimento

**Fonte:** Princípios de design da Microsoft

A Microsoft aposta no **Design com movimento** demonstrado na Figura 4. Em seu princípio de design, a explicação é de que o movimento dá vida às experiências e transmite uma sensação de elegância. Essa e outras inspirações levaram aos princípios de design, como: Dedicação ao refinamento do aplicativo; Faça mais com menos; Rápido e flexível; Autenticidade digital; Aproveite todos os recursos de integração, bem como o trabalho com modelo hierárquico demonstrado a seguir.



Hierarquia



Hierarquia insuficiente

**Figura 5:** Modelo Hierárquico

**Fonte:** Princípios de design da Microsoft

A Figura 5 ilustra alguns dos elementos do design da Microsoft. Segundo a mantenedora do SO, a utilização dos grids tornam a aplicação moderna e coerente, chamada grade tipográfica. Tal conceito é muito utilizado na retratação e utilização do modelo hierárquico.

## 2.3 ASPECTOS DE INTERFACE

Hoje em dia, a criação de padrões de interface é uma das maiores preocupações dos desenvolvedores ao criar uma aplicação, tendo-se assim um estudo bem específico dentro de quais recursos são utilizados para padronizar a interface da aplicação.

Em seu artigo, Antonio Barbosa (2013) afirma que os padrões de navegação de interface gráfica para dispositivos móveis são classificados em: padrões primários e secundários. O autor classifica ainda as seguintes categorias de padrões: formulários, tabelas e listas, busca ordenação e filtragem, ferramentas, gráficos, convites, *feedback* e *affordance*, ajuda e antipadrões. A Tabela 1 apresenta os padrões definidos como primários de navegação.

<b>Padrões primários de navegações</b>	
<b>Springboard</b>	Caracteriza-se por uma página inicial de opções de menu que agem como um ponto de partida para o aplicativo.
<b>Menu de listas</b>	Caracteriza-se na forma de lista, pode ser simples, agrupadas e avançadas com recursos adicionais para busca, navegação ou filtragem. Cada item da lista é um ponto de partida para o aplicativo.
<b>Menu de abas</b>	Caracteriza-se pelo seu design na forma de abas, que pode ser superior ou inferior.
<b>Galeria</b>	Caracteriza-se pela exibição na forma de itens de conteúdo que podem ser organizados em um carrossel, uma grade ou em slide show.
<b>Dashboard</b>	Caracteriza-se pela forma de painéis de instrumento que fornece resumo de indicadores principais de desempenho (Key Performance Indicators – KPIs).
<b>Metáfora</b>	Caracteriza-se por uma página inicial modelada para refletir a metáfora do aplicativo. Pode ser utilizado para ajudar a catalogar e categorizar itens em um aplicativo.
<b>Megamenu</b>	Caracteriza-se por um grande painel sobreposto com formatação e agrupamento personalizados das opções de menu.
<b>Padrões secundários de navegações</b>	
<b>Carrossel de páginas</b>	Caracteriza-se pelo gesto de arrastar os dedos para navegar rapidamente em um conjunto de páginas.
<b>Carrossel de imagens</b>	Caracteriza-se pelo uso em dimensões 2D ou coverflow (uma interface tridimensional para navegação em biblioteca).
<b>Lista expandida</b>	Caracteriza-se pela permissão de que em uma única tela seja acessada para revelar mais informações.

**Tabela 1:** Padrões primários e secundários de navegações.

**Fonte:** [BARBOSA, 2013]

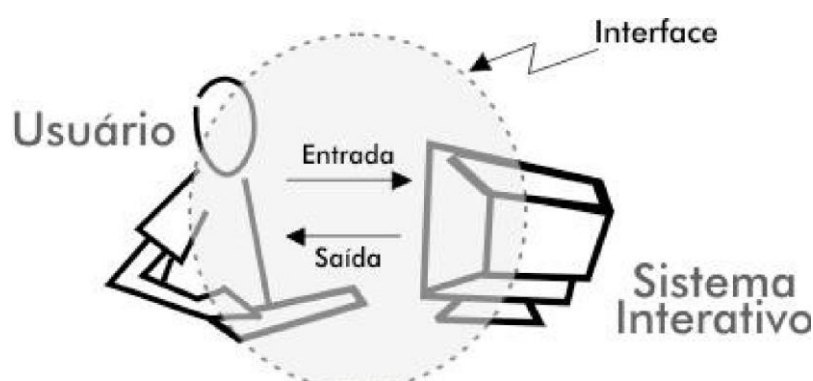
Na Tabela 1 podemos visualizar a divisão dos padrões definidos por Antonio Barbosa (2013), onde mostra dividido os itens de um aplicativo, com nomes específicos de cada, demonstrando os padrões, com as funções que cada um exerce na aplicação.

### 2.3.1 HOMEM-MÁQUINA

Mesmo antes de se ter o termo “dispositivo móvel” já havia uma preocupação com a interface e a usabilidade com que o usuário terá acesso. Para iniciar o estudo do conteúdo de interação homem-máquina devemos conhecer o conceito interface.

#### 2.3.1.1 INTERFACE

Segundo Bida de Oliveira (2015), pode-se visualizar hoje em dia como conceito de interface a visão de janelas, menus, ícones e barra de rolagem, entretanto sua abrangência e conceituação é bem maior. Todos os comportamentos observáveis externamente em que o usuário e o computador possam passar podem ser considerados uma interface homem-máquina, onde se tem uma entrada e uma saída de dados. A Figura 6 demonstra que tal relacionamento nasce dos conceitos básicos da interação, considerando as perspectivas de entrada e saída.



**Figura 6:** Funcionamento da interface homem-máquina

**Fonte:** Bida de Oliveira, 2015

Podem-se observar vários fatores para justificar a grande preocupação atual voltada para interfaces homem-máquina. Destaca-se a de que está ficando mais

precoce a utilização de recursos de informática, prevendo-se que todo ser humano irá utilizar computadores no futuro de uma forma ou de outra.

Além disso, no mercado as lojas afirmam que nas vendas de produtos de informática, a saída maior são dos produtos com que o usuário tem maior acesso à funcionalidade fornecida pelo sistema. No custo de um sistema computacional agrega-se também o treinamento de usuários, portanto quanto mais difícil for a usabilidade da interface do sistema, maior o custo de treinamento, afirma Bida de Oliveira (2011).

### 2.3.2 DISPOSITIVOS MÓVEIS

O desenvolvimento de uma aplicação para dispositivos móveis com uma interface aparentemente agradável e de boa usabilidade é um grande desafio devido a gama de diferentes dispositivos com diferentes tamanhos, sistemas e hardwares que englobam a computação móvel. Assim, a preocupação anteriormente existente quanto a interface para computadores, toma proporções mais amplas quanto aos dispositivos móveis.

#### 2.3.2.1 INTERFACE INTELIGENTE

Carla, Ferreira, Sant'ana (2004) afirma ainda que o que torna uma interface inteligente é a possibilidade de adaptação às diferentes necessidades dos usuários. Assim verifica-se que: *“Interface inteligente é considerada aquela que entenda os objetivos e metas do usuário e saiba atingi-los ou que facilite uma interação mais natural, com uma maior tolerância a erros e com formatos mais agradáveis.”*.

Machado (2013) completa ainda que para ser considerado inteligente, uma interface deverá possuir os seguintes componentes:

- **Modelo de utilizador:** é uma compilação, ou seja, um conjunto de informações que descreve o usuário. É utilizada para determinar a apresentação de dados, que tipo de ajuda dar, e como o usuário irá interagir com a interface;
- **Ajuda inteligente:** apresenta ao usuário a ajuda que ele precisa para um determinado tempo, ou determinada situação;

- **Adaptabilidade da interface:** o usuário pode configurar como quer que a interface seja. O sistema também pode adaptar-se a preferência de outras aplicações;
- **Reconhecimento dos planos:** usado para deduzir o que o usuário planeja fazer. Neste reconhecimento faz-se uso do modelo do utilizador.

Para a melhor interação do usuário, a interface deverá adaptar-se às diferentes proporções dos dispositivos que será acessado o sistema.

### 2.3.2.2 INTERFACE ADAPTATIVA

Aplicações de alta complexidade que utilizam uma alta quantidade de informações ao mesmo tempo e que necessitam de que o usuário faça o uso de várias funções ao mesmo tempo, tendem a ter a necessidade de que sua interface faça uma adaptação a sua necessidade. Logo:

“A interface é considerada adaptável, quando realiza as adaptações unicamente no momento em que o usuário a requisita, ou seja, o usuário adapta o sistema a seu modo. Adaptar o usuário ao sistema significa oferecer-lhe treinamento, documentação, tutores, facilidades de ajuda entre outras, enquanto o sistema permanece fixo” [CARLA, FERREIRA e SANT'ANA, 2004]“.

A adaptação pode ser empregada em dois tipos:

- **Adaptação estática da interface do usuário:** realizada durante o desenvolvimento do software. Está relacionada a modificação e a manutenção do sistema, no momento de expansões e de atualizações;
- **Adaptação dinâmica da interface do usuário:** realizada durante a execução do software.

A adaptação é feita utilizando o modelo do utilizador descrito anteriormente, onde há uma verificação do perfil e da utilização do usuário e assim adaptando-se automaticamente ao perfil necessário. Sendo utilizada essa interface, o sistema pode ser personalizado para estilos cognitivos individuais, necessidades de informações e tarefas personalizadas.



### 2.3.2.3 INTERFACE RESPONSIVA

Interface adaptativa consiste na resolução do problema da visualização de aplicativos web nos diversos tamanhos de aparelhos e diferentes tamanhos de telas existentes no mercado. Como Ferreira (2011) afirma, um design responsivo inclui:

- Adaptar o layout da página de acordo com a resolução em que está sendo visualizada.
- Redimensionar as imagens automaticamente para que caibam na tela e para que não sobrecarreguem a transferência de dados em um celular, por exemplo.
- Simplificar elementos da tela para dispositivos móveis, onde o usuário normalmente tem menos tempo e menos atenção durante a navegação.
- Ocultar elementos desnecessários nos dispositivos menores.
- Adaptar tamanho de botões e links para interfaces touch onde o ponteiro do mouse é substituído pelo dedo do usuário.
- Utilizar de forma inteligente recursos mobile como geolocalização e mudança na orientação do aparelho (horizontal ou vertical).

Com esse cuidado, o desenvolvedor garante a visualização de sua aplicação seja amigável a todos os aparelhos que será executada.

### 2.4 AVALIAÇÕES DOS APLICATIVOS PELAS APPS STORES

Para que o aplicativo tenha aceitação de comercialização nas apps stores as empresas de sistemas fazem uma verificação do aplicativo antes de disponibilizá-lo para sua loja de aplicativos.

Para serem aceitos, os aplicativos devem seguir os mínimos padrões citados acima. O Android é mais flexível em relação ao designer da aplicação, pois com o desenvolvimento desde projeto não foi encontrado algum check list que a Google utiliza para fazer alguma checagem. Com a chegada da nova versão do Android, a 5.0 nomeada Lollipop houve uma alteração considerável em sua interface, é

esperado que a exigência em designer de aplicativos comece a ter uma atenção maior.

“Uma notável reformulação nas suas funcionalidades, interface, novas animações e interações baseadas em gestos. O lançamento da nova versão do Android marcou o fim dos cantos quadrados e das cores primárias que o caracterizavam. Com a chegada do novo sistema operacional há mudanças, também, no acesso às notificações. De forma ainda mais dinâmica, o Android Lollipop vem acompanhado de um novo acesso interativo, que permite ao usuário ter acesso à mensagens e alertas sem precisar desbloquear o aparelho. Também será possível ver as notificações na parte superior direita da tela de bloqueio do celular, especificamente. Com isso, o usuário vai poder descartar as mensagens menos importantes apenas clicando nelas ou arrastando-as só com simples toques na tela bloqueada. Essa possibilidade de ler e até mesmo responder uma notificação diretamente da tela de bloqueio torna o seu uso mais fácil e prático” (TECHTUDO, 2015).

Na Figura 7 é possível visualizar a nova interface da última versão do Android.

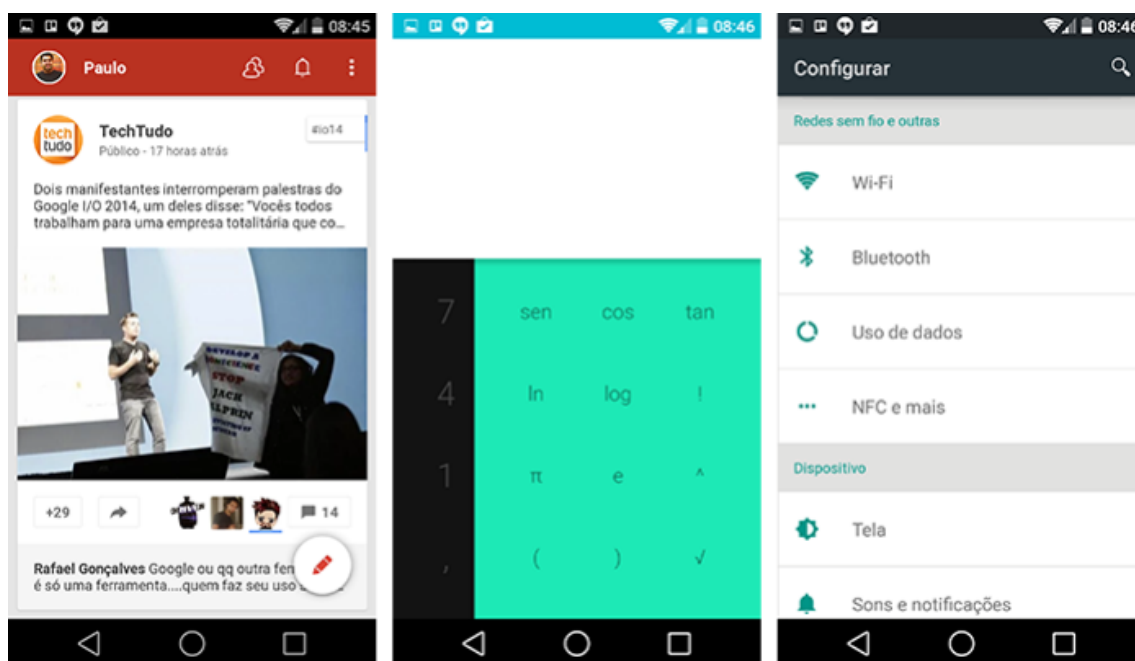


Figura 7: Android 5.0

Na Figura 7 *Design* mais limpo, sistema de *grids*, mudança nos ícones. Essas são algumas diferenças do sistema 4.0 para a nova versão do Android, a 5.0.

Em seu site, a Apple é mais rigorosa nesse quesito, tendo um check list a ser seguido para assim poder disponibilizar a aplicação. A seguir temos alguns itens que estão disponibilizados no site da Apple que podem ser observados no *check list*:

1. Não exiba produtos Apple menores que 25 mm de comprimento para materiais impressos e 200 pixels em tela.
2. Manter o tamanho relativo correto entre os produtos sempre que diversos produtos forem mostrados, ou seja, manter proporção para os tamanhos de aparelhos;
3. Não sobreponha imagens individuais de produtos Apple; em vez disso, use os layouts de produtos sobrepostos fornecidos.
4. Não altere, distorça nem modifique as imagens de produtos Apple fornecidas.
5. Não adicione reflexos ou sombras.
6. Não corte, recorte ou obstrua qualquer parte das imagens de produtos Apple.
7. Não incline nem altere o ângulo das imagens de produtos.
8. Não anime, incline, rotacione nem gire as imagens de produtos.
9. Não adicione elementos gráficos que pareçam entrar ou sair da tela do produto.
10. Não realce o contorno de um produto
11. Não utilize 3D nem crie qualquer simulação de um produto Apple. Não utilize ilustração para reproduzir um produto Apple, exceto para materiais de instrução. Caso dispositivos portáteis genéricos sejam ilustrados, não inclua detalhes que sejam exclusivos dos produtos Apple, como o botão de início ou de controle de volume.
12. Não crie botões ou ícones com imagens de produtos.
13. Não utilize o selo da App Store ou imagens de produtos Apple em embalagens descartáveis ou promoções da indústria de alimentos. Utilize o nome do produto apenas em texto.

Igualmente o Android, ao acessar o endereço eletrônico do Windows Phone não há disponibilizado check list para aprovação de publicação de aplicativos. Ao

fazer a verificação de aplicativos, foi observada uma grande exigência para a utilização de layout de grades conforme exposto na Figura 9.

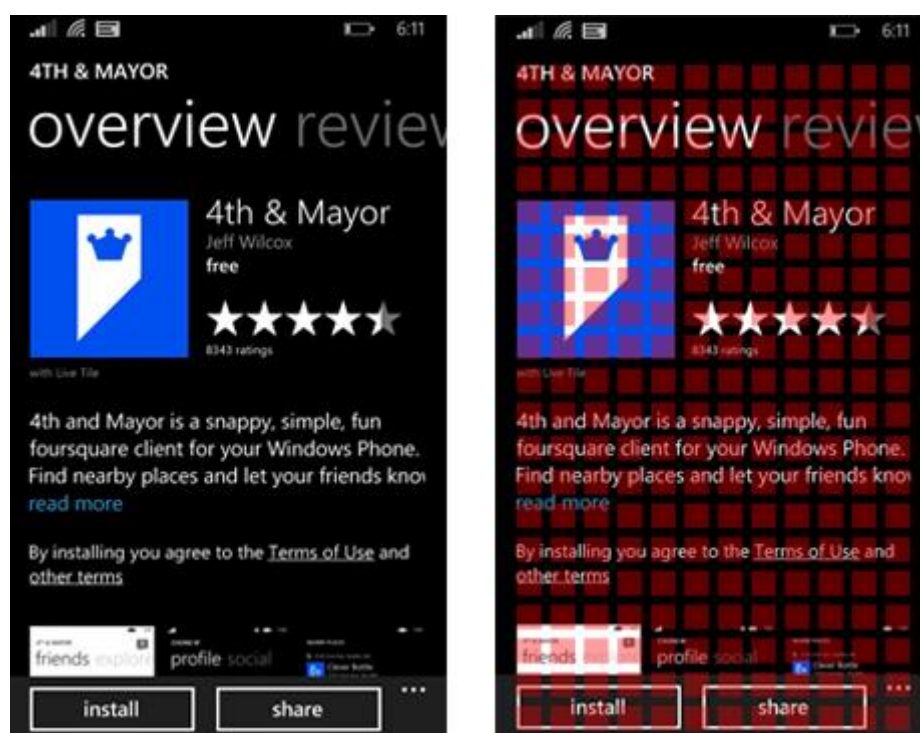


Figura 8: Layout de grades

Ao fazer a verificação as aplicações, foi notado que os aplicativos são organizados em forma de grids, tendo assim uma igualdade de design a grande maioria dos aplicativos.

#### 2.4.1 MOTIVOS PARA REPROVAÇÃO

Além de motivos de design da aplicação, o endereço eletrônico Olhar Digital (2014) listou alguns fatores que levam um aplicativo a ser rejeitado pela Apple:

- **Falta de informações:** o aplicativo deve conter detalhes dentro de uma seção “Apple Review”, especificando determinadas necessidades de hardware, assim como configurações extras ou criação de conta;
- **Bugs:** para a Apple, um aplicativo deve ser publicado somente se passar por todos os testes e não conter nenhum bug;
- **Interface amigável:** a interface deve seguir a ideia de ser “amigável, refinada e clean”. Além disso, o design deve seguir o guia de design da Apple e os “Sim e Não’s de Design de Interface de Usuário.”;

- **Aplicativos repetidos:** submeter aplicativos que são a mesma coisa diminuem a chance da aprovação;
- **Descrições insuficientes:** a descrição de um app deve ser clara e explicar a funcionalidade e seu objetivo;
- **Falta de funcionalidades:** a multiplicidade de nichos e a adição de algo a mais na vida do usuário são algumas essenciais funcionalidades que uma aplicação deve conter para conseguir a aprovação.

Assim, cabe salientar e expor o que é descrito pela Google em seu endereço eletrônico, no qual descreve que:

Políticas de conteúdo aplicam-se a qualquer conteúdo que seu aplicativo exiba ou ao qual se vincule, incluindo quaisquer anúncios exibidos aos usuários e quaisquer conteúdos gerados por usuários que o aplicativo hospede ou aos quais se vincule [Google Play, 2015].

No termo, fica especificado:

- Não são permitidos aplicativos que contenham e explicitem conteúdos pornográficos;
- Representação de violência, os aplicativos não podem conter matérias que ameacem, assediem ou intimidem outros usuários;
- Não é permitido conteúdo contra grupos de pessoas com base em raça ou origem étnica, religião, deficiência, sexo, idade, condição de veterano ou orientação/identidade sexual;
- Não é permitido atividades ilegais como vendas de produtos não legalizados;
- Conteúdo que envolva jogos de azar online é proibido;
- Não é permitido conteúdo que prejudique, interfira na operação ou acesse, de forma não autorizada, redes, servidores ou outra infraestrutura:
  - o Não transmita vírus, worms, defeitos, cavalos de troia, malware ou qualquer outro item que possa criar vulnerabilidades de segurança ou danificar dispositivos, aplicativos ou dados pessoais dos usuários.

- o Aplicativos que coletam informações (como a localização ou o comportamento do usuário) sem o conhecimento do usuário (spyware) são proibidos.
- o Scripts maliciosos e golpes de phishing de senha também são proibidos no Google Play, assim como aplicativos que levam o usuário a fazer o download ou a instalação de aplicativos de fontes fora do Google Play involuntariamente.
- o Um aplicativo transferido do Google Play somente modifica, substitui ou atualiza seu próprio código binário de APK usando o mecanismo de atualização do Google Play.
- Interferência do sistema:
  - o Um aplicativo transferido do Google Play (ou seus componentes ou elementos derivados) não podem fazer alterações ao dispositivo do usuário fora do aplicativo sem o conhecimento e o consentimento do usuário.
  - o Isso inclui comportamentos como a substituição ou a reorganização da apresentação padrão de aplicativos, widgets ou configurações no dispositivo. Se um aplicativo fizer essas alterações com o conhecimento e o consentimento do usuário, será necessário que a informação sobre qual aplicativo fez a modificação seja clara para o usuário. Além disso, será preciso que essa alteração possa ser facilmente revertida ou desinstalada do aplicativo pelo usuário.
  - o Os aplicativos e seus anúncios não podem modificar nem adicionar marcadores de página e configurações do navegador. Além disso, também não podem adicionar atalhos para a tela inicial nem ícones no dispositivo do usuário como um serviço para terceiros ou com fins de publicidade.
  - o Os aplicativos e seus anúncios não podem exibir propaganda por meio de notificações de nível do sistema no dispositivo do usuário, a menos que as notificações sejam provenientes de um recurso integral

fornecido pelo aplicativo instalado. Por exemplo, um aplicativo de companhia aérea que notifica os usuários sobre ofertas especiais ou um jogo que notifica os usuários sobre promoções inseridas em jogos.

- o Os aplicativos não podem estimular, incentivar ou enganar os usuários para que removam ou desativem aplicativos de terceiros, exceto se isso fizer parte de um serviço de segurança fornecido pelo aplicativo.

Como verificado acima, esses elementos levam à reprovação de aplicativos pela loja da Apple, esses elementos devem ser evitados para que a aplicação não seja banida. No desenvolvimento de uma aplicação web também deve-se ficar atento para não conter tais elementos.

## **2.5 HTML5**

A cada site que acessamos, a cada página da internet que visitamos, independente do conteúdo, seja uma rede social ou site de notícias, em todos existem uma coisa em comum: o HTML. Nota-se que:

“Código HTML está presente nas páginas da web há mais de dez anos e tem uma larga aceitação dos usuários por todo o mundo. Com certeza, qualquer pessoa que tem uma ligação um pouco mais estreita com computadores e internet já ouviu falar em HyperText Markup Language (Linguagem de marcação de hipertexto), ou seja, o HTML. A principal característica desse tipo de programação são as “tags” – as marcações para inserção de algum tipo de objeto ou característica de um objeto através de comandos em inglês e números.” BARWINSKI (2009).

A utilizar o sistema de divisão de marcações com tags, na nova versão, o HTML5 essas etiquetas de marcação são mais objetivas, facilitando tanto a navegação, quanto o desenvolvimento.

“Será o novo padrão para a estruturação e apresentação de conteúdo na Word Wide Web trazendo melhorias significativas com novas funcionalidades de semântica e acessibilidade, além de melhorar o suporte aos mais recentes conteúdo multimídias.” BOTELHO (2011).

Segundo Medeiros, o HTML5 propicia ao desenvolvedor Web o poder de criar jogos que variam em complexidade, alguns chegando a ser comparados a jogos desenvolvidos em plataformas complexas.

### 2.5.1 O QUE MUDA COM O HTML5?

As novas características do HTML5 permitem utilizar menos recursos de terceiros, como o Flash. Afirma Medeiros que o HTML 5 facilita e melhora a apresentação e estrutura do conteúdo da Web através de novas tags para a utilização de canvas, tocar vídeos, drag-and-drop, etc. Todos esses recursos eram possíveis apenas com Flash.

Barwinski (2009) cita as novidades do HTML5 como:

- **Tag Canvas:** especializadas em renderizar imagens em bitmap, as tags canvas serão específicas para a edição breve de imagens através de APIs ou JavaScript. Esse tipo de edição acontece pura e exclusivamente de maneira muito similar a outros geradores de imagem em duas dimensões (2D). Além disso, as Canvas Tags são compatíveis com as folhas CSS;
- **Tags de vídeo:** Incluir vídeos em HTML como conhecemos hoje, utilizando códigos para o “embed”, ou seja, incorporar vídeos à página será muito mais simples. O HTML 5 possui tags específicas para a inserção de vídeos no corpo da página. Basta inserir a tag de vídeo assim como se faz com aquela destinada às imagens. Assim, será necessário indicar uma “src”, ou seja, “source” (fonte) que nada mais é do que a origem do vídeo;
- **Base de dados:** apesar de não ser um recurso muito utilizado por usuários não desenvolvedores, os bancos de dados são importantes para os sites. No HTML 5, algumas vantagens serão implementadas como a possibilidade de entradas de valores ou palavras chave, além do banco de dados SQL.

Para o desenvolvimento em HTML5, Sánches (2013) dá algumas dicas como:



1. **Escolha um estilo de código:** O HTML5 permite que a marcação seja escrita em maiúsculas, minúsculas ou uma mistura delas. Além disso, ninguém é obrigado a criar todos os atributos em elementos como <img>. Você pode até omitir o atributo type quando usar <style>, <link> ou <script>. O estilo é uma escolha sua, mas se está trabalhando com uma equipe, verifique se todos estão em sintonia. E ainda há um número de elementos e atributos que estão ultrapassados em HTML5, como <big>, <center> e <font>. Melhor deixá-los para lá;
2. **Leia as specs:** as especificações são uma parte importante. Ler as specs é a única maneira de entender alguns elementos;
3. **Planeje sua marcação:** Usar o conceito de documento outline vai ajudar a criar uma marcação bem estruturada. No processo de criação do design, é possível criar uma lista alinhada à página, o que vai obrigá-lo a pensar os níveis de títulos que devem ser usados e quais elementos serão colocados em cada lugar;
4. **Escolha o elemento certo:** Pode ser difícil escolher o elemento certo para cada projeto. Para resolver voltamos à premissa de usar um elemento para descrever o significado do conteúdo, e não a sua aparência.
5. **Guerra de codecs:** Quando criar conteúdo multimídia para o seu site codifique seus arquivos múltiplas vezes e inclua opções para browser com capacidades inferiores.

Um dos desafios de se trabalhar com web é acompanhar as novidades tecnológicas que aparecem e desaparecem, se renovam e se superam em uma velocidade impressionante.

### 3 METODOLOGIA

O presente trabalho foi realizado utilizando-se de diversas fontes bibliográficas, no qual o assunto se encontrasse referendado. Assim, foram considerados artigos de pesquisadores renomados, trabalhos de conclusão de curso, dentre outras fontes como os sítios eletrônicos dos próprios mantenedores dos sistemas operacionais, pois em seu conteúdo continham várias informações que poderiam passar despercebidas por alguns pesquisadores ou até mesmo que possam ter sido incorporadas aos sites após os estudos anteriores.

Para o desenvolvimento da aplicação e para a realização de testes foram utilizados os seguintes itens:

- Notebook;
- Nokia Lumia 820;
- iPhone 4S;
- Motorola Moto G;
- Motorola Razr;
- Nokia Lumia 925;
- Samsung Galaxy Mega

O trabalho e posterior desenvolvimento foi inicialmente guiado com a sua divisão em três itens partes:

1. **Identificação da plataforma;**
2. **Mapeamento CSS;**
3. **Aplicação para inserção da adaptabilidade.**

Com isso, passou a realizar trabalhos inerentes e específicos as tecnologias a serem utilizadas neste processo.

As aplicações contidas neste trabalho foram desenvolvidas utilizando:

- Linguagem de programação Java, seguindo JSP (Java Server Pages);
- Desenvolvido na plataforma NetBeans 8.1
- Servidor Apache TomCat 8.0

O maquinário utilizado foi um notebook Acer Aspire, Core 2 Duo, com 2 Gb de memória RAM.

Foi desenvolvido no laboratório de hardware da Fundação Universidade do Tocantins (UNITINS). O desenvolvimento deu-se no período de Outubro a Novembro, os testes das aplicações foram ocorrendo quando verificava-se a necessidade, mas os testes finais deram-se de Novembro a Dezembro.

## 4 RESULTADOS

Dentre os vários aspectos relevantes para uma adequada apresentação, todos os conteúdos trabalhados foram importantes para a produção do presente trabalho. Assim, para o desenvolvimento de um sistema responsivo e adaptativo foi necessário utilizar-se das tecnologias *HTML 5*, *CSS* e *JavaScript*.

### 4.1 O PROBLEMA

Frente ao problema proposto que foi a criação de uma aplicação adaptável ao sistema em que está a ser executada a aplicação, trabalhando assim as diretivas de designer impostas por cada sistema operacional, foi necessário ter-se bem definida a arquitetura do sistema, ou seja, como funcionaria a aplicação, os arquivos, e todos os componentes necessários para que a adaptação ocorra.

Para se desenvolver a aplicação foco do trabalho foi necessário alguns itens que fizessem a adaptabilidade e a identificação de tamanhos de telas e outros elementos necessário para se tornar o sistema adaptável, diante das características dos dispositivos e sistemas operacionais em questão.

Tão logo, fora levantada a necessidade de criação de arquivos responsáveis pela adaptabilidade do sistema e também para alterar as diretrizes de acordo com o sistema operacional do dispositivo a ser executado, cada um com sua função bem definida, como:

- **Arquivos *HTML*:** No arquivo HTML foram definidas as *div's*, ou seja, onde consta todos os componentes da aplicação como botões, formulários, modais, caixas de mensagens e todos os elementos necessários para a criação da aplicação;
- ***JavaScript*:** O JavaScript contém o conteúdo responsável por fazer a identificação de qual sistema que a aplicação está a ser executada, podendo estar incluso no arquivo *HTML* ou estar em um arquivo separado.
- **Arquivos *CSS*:** O CSS foi responsável por identificar qual o tamanho da tela está sendo exibida a aplicação, para isso fora utilizado as *medias queries*. O CSS também fica responsável pelo designer dos elementos incluídos no HTML.

Tendo-se o conhecimento de quais itens utilizados, e também tendo-se bem definido qual função de cada elemento da aplicação foi necessário entender a arquitetura do sistema, de como funciona toda a tramitação de dados do sistema.

## 4.2 ARQUITETURA

Para desenvolver a arquitetura do sistema foi levado em consideração quais elementos deveriam ser utilizados para conseguir a adaptação e a responsividade dos elementos, chegando-se a arquitetura exposta da Figura 10.



**Figura 9:** Arquitetura do sistema

Inicialmente a arquitetura ficou definida com cinco elementos básicos: o *HTML5*, o *Javascript* e três arquivos *CSS* distintos para se verificar quais e como os elementos HTML utilizados responderiam a cada tipo de plataforma móvel. Esses arquivos foram necessários para que a adaptabilidade e o *material design* correto

fosse carregado e então mapeado para a aplicação que realizaria a transformação considerando a adaptabilidade e forma de resposta de cada S.O móvel.

### **4.3 APPWEB/HTML5**

Para o desenvolvimento da aplicação web adaptável ao sistema foi necessário primeiro fazer a identificação de quais elementos utilizar em uma aplicação, ou seja, quais elementos são mais comuns nas aplicações nos diversos tipos de Sistemas Operacionais existentes para os dispositivos móveis.

Para definição de tais elementos, foi feita a verificação dos aplicativos mais relevantes dos sistemas Android, Windows Phone e IO's, para ser verificada a adaptabilidade do sistema e quais elementos eram mais comumente utilizado nos aplicativos *mobile*.

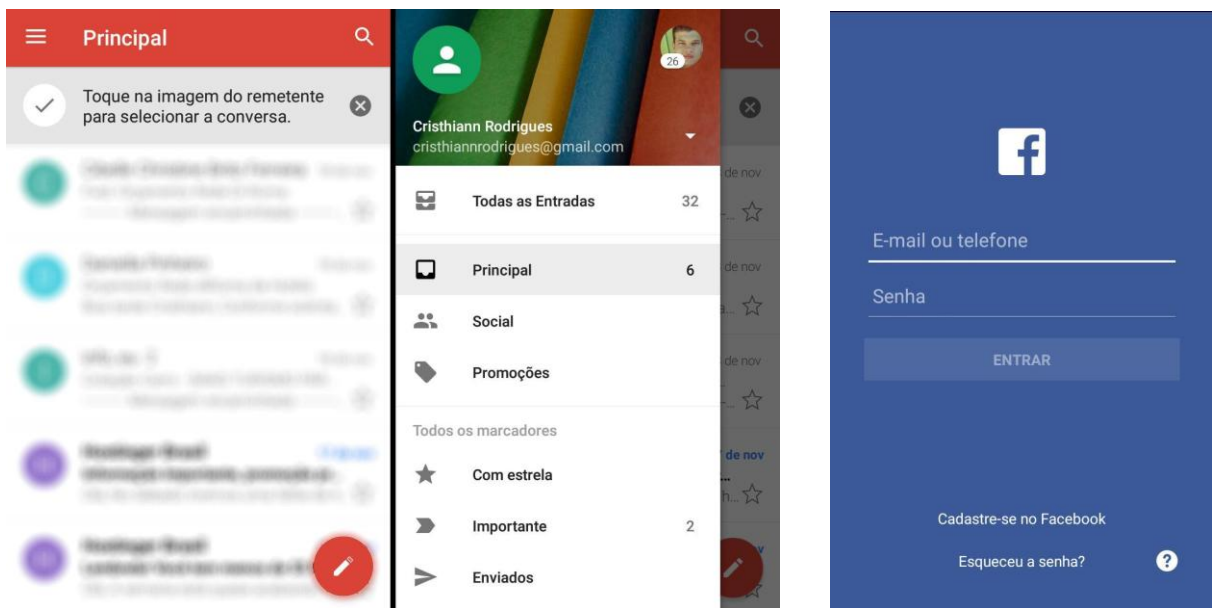
Desta forma, uma pesquisa foi feita nas lojas de aplicativos dos sistemas, sendo selecionados os aplicativos que estavam nos aplicativos mais baixados de cada plataforma.

#### **4.3.1 VERIFICAÇÃO DE APLICATIVOS**

Para a escolha dos elementos foi feito um estudo em vários aplicativos existentes nas lojas de cada sistema.

##### **4.3.1.1 ANDROID**

Para a plataforma Android foram verificados os aplicativos do Gmail e Facebook, conforme demonstrado na Figura 10.

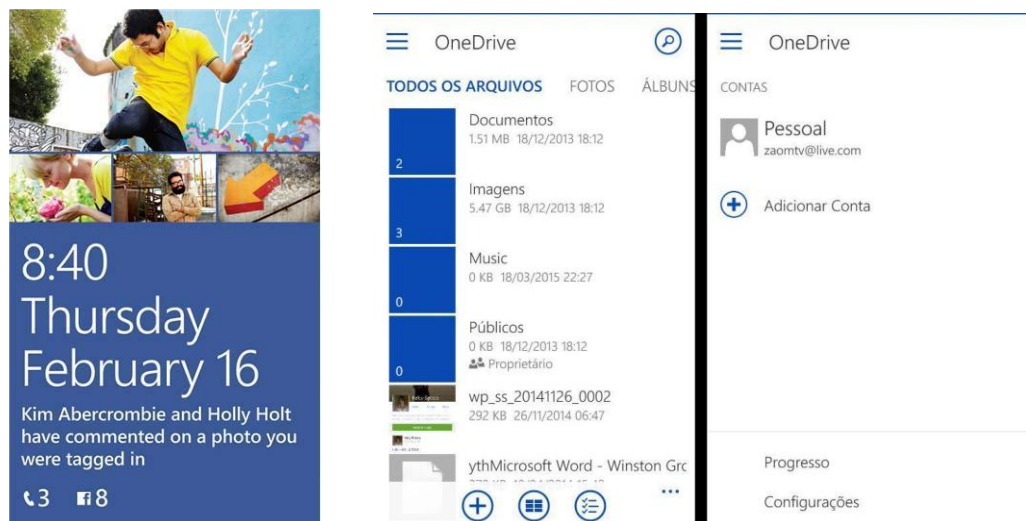


**Figura10:** Gmail e Facebook.

Nos aplicativos verificados no sistema Android, foi detectado em sua maioria a presença de formulários, botões, listas, e menu como elementos mais comuns e sustentáveis a uma adaptabilidade em novas aplicações.

#### 4.3.1.2 WINDOWS PHONE

Quanto a plataforma Windows Phone foram verificados os aplicativos Facebook e OneDrive, que consistiam nos mais acessados e obtidos em sua loja.



**Figura 11:** Facebook e One Drive.

Assim como no sistema Android, foi verificado os itens menu, botões e botões. Foi verificado também a presença do layout em *grid* que caracteriza todos os aplicativos Windows Phone.

#### 4.3.1.3 IO's

Por fim, na identificação dos principais componentes características de da S.O. No IO's foram verificados os aplicativos Facebook e Whatsapp.

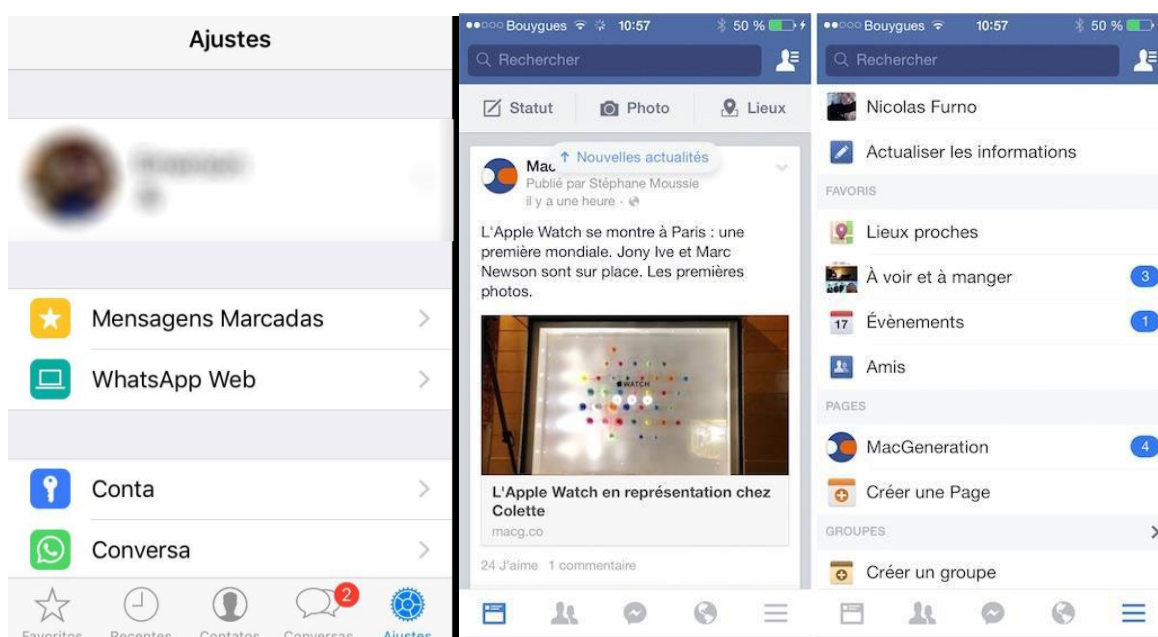


Figura 12: Whatsapp e Facebook no IO'S.

No IO's foram identificados, como nos outros sistemas, a presença de menu, formulários e botões. Diferenciado dos demais o menu do iPhone tem uma localidade diferenciada. Enquanto os demais sistemas utilizam o menu na parte superior da tela, o IO's utiliza os menus na parte inferior.

Feita essa verificação ficaram definidos alguns componentes a ser colocado na aplicação, como:

- **Formulários;**
- **Botões;**
- **Campos de Texto;**
- **Listas de componentes;**
- **Tabelas;**
- **Links.**



Esses itens aparecem em todas as aplicações verificadas, ou seja, para uma visualização de uma adaptabilidade esses elementos são essenciais.

#### **4.4 DESENVOLVIMENTO**

No decorrer do desenvolvimento foram feitos vários testes para identificar o comportamento da adaptabilidade, a responsividade e vários outros itens que são necessários para a aplicação. Deste modo, importante se torna descrever o processo, apresentando os principais trechos deste processo.

##### **4.4.1 TRABALHOS RELACIONADOS**

Antes de partir para o desenvolvimento da ferramenta foi feito uma pesquisa se existe algum sistema que faça a mesma função ou algum serviço semelhante ao proposto.

Na pesquisa foram identificados dois sites que fazem trabalhos semelhantes ao trabalho proposto.

O site do PROTO.IO, gera automaticamente o CSS de um *switch* no modelo das três plataformas neste trabalho estudadas. O site gera o código CSS para que o usuário copie o código e coloque no código CSS da aplicação que está desenvolvendo.

O site da CSS3Maker permite que o usuário configure da forma que quiser o item disponível. Quando alterado, o site gera em tempo real o código CSS em que o usuário está alterando.

Feito a verificação desses sites foram verificadas algumas vantagens da aplicação a ser desenvolvida neste trabalho em relação às aplicações verificadas:

- Identificação automática de elementos;
- Permite download de arquivos, sendo assim não necessitando de copiar texto;
- Código CSS completo;
- Download tanto do código CSS quanto HTML;

#### 4.4.2 PÁGINA HTML5

Para solicitar o código digitado pelo usuário, foi desenvolvido uma página em HTML5, solicitando que usuário entre com sua aplicação desenvolvida em HTML (HTML5 ou HTML). A aplicação faz a varredura no arquivo a procura de alguns elementos como:

- **Declaração *Doctype*:** se não houver a declaração ou a declaração do mesmo estiver padrão HTML4 a aplicação faz a alteração ou a inclusão da mesma no padrão HTML5;
- **A tag *<head>*:** A aplicação faz a varredura do código, se a tag não estiver presente, a aplicação retornará uma mensagem de erro informando ao usuário que seu código está incompleto;
- **Tag *<body>*:** Na ausência da tag *<body>* a aplicação retorna mensagem ao usuário de que falta o componente;
- **Componentes:** Após as verificações acima forem feitas, a aplicação fará a varredura de todo o *<body>* a procura dos componentes que compõem uma aplicação web como botões, inputs e etc.

Os itens acima são componentes que compõem uma página em HTML5, portanto a presença de cada componente se dá de bastante importância, por isso houve-se a necessidade de tais verificações.

#### 4.4.3 IDENTIFICAÇÃO DA PLATAFORMA

O primeiro ponto a ser destacado quanto a implementação consistiu em identificar a plataforma. Para isso, a forma mais adequada uma vez que se trabalha com a linguagem HTML foi a criação de uma função *JavaScript*, a mesma e apresentada a seguir.

```
1 function sistema(){
2   if (navigator.userAgent.indexOf('Linux') != -1) {
3     var so = "Linux";
4   }
5   else if ((navigator.userAgent.indexOf('Win') != -1)) {
6     document.write();
7     var so = "Windows";
8   }
9   else if (navigator.userAgent.indexOf('Mac') != -1) {
10    document.write();
11    var so = "Macintosh";
```

12	}
13	return so;
14	}

**Figura 13:** *JavaScript* para teste

A variável *so* foi utilizada para armazenar qual sistema o aplicativo está a ser executado, definindo assim, como os elementos deveriam se comportar diante do S.O. acessado. Ponto importante a se considerar é que as plataformas móveis tem por base as estruturas utilizadas no S.O's dos computadores e tão logo, os navegadores da mesma forma conseguem interpretar a plataforma que no momento esta visualizando o conteúdo.

Assim, na aplicação foi utilizado o código descrito acima, obtendo informações do sistema, no qual a plataforma que a aplicação está a ser executada e escrevendo no código *HTML5* qual arquivo correto *CSS* que deve ser importado pela aplicação.

A função responsável por passar qual sistema está a ser executado o aplicativo é o método *navigator.userAgent.indexOf* que:

- ***navigator***: é um objeto onde fica armazenada as informações do navegador;
- ***userAgent***: essa propriedade retorna o valor do cabeçalho *user-agent* enviado pelo navegador para o servidor, contendo informações como nome, versão, e a plataforma do *browser*;
- ***indexOf***: esse método retorna a ocorrência de um valor especificado de uma sequencia. Retorna -1 se o valor da pesquisa for o valor da pesquisa não ocorrer.

Tendo-se a identificação da aplicação feita e sabendo-se em qual plataforma a aplicação está a ser executada, passa-se a desenvolver os itens da aplicação JSP que uma vez identificados os elementos HTML, no momento de sua execução no navegador do dispositivo móvel, em tempo de execução acionará o *CSS* adequado .

Para os testes do mapeamento original para as aplicações criou-se uma tela de *login*, acrescida de campos para inserção de login e senha, um botão para

confirmação de login, e dois *links*. Desta forma foi possível, antes da implementação da aplicação visualizar se o mapeamento dos componentes HTML estavam corretos para cada plataforma.

Na página criada como teste, verificou-se ainda menu com dois itens de exemplo para que seja verificada a adaptação à plataforma, o que era importante para o contexto.

#### 4.4.4 MAPEAMENTO CSS

Diante da análise das aplicações mercadológicas mais utilizadas em cada uma das plataformas demonstradas anteriormente, foram selecionados os componentes que frequentemente são mais constantes nas aplicações para dispositivos móveis. Os arquivos CSS foram definidos com o intuito de que a aplicação identificasse o layout de cada elemento incluído no arquivo HTML, dependendo de qual plataforma está a ser executada a aplicação.

##### 4.4.4.1 HTML5

Iniciando a padronização dos layouts e também como identificado anteriormente, um dos itens selecionados foram os links, e para que seja verificada essa adaptação e também gerar a estilização para todas as tipografias da aplicação foi gerado o código a seguir:

<pre>html, body {   font-family:     "Helvetica", "Arial",     sans-serif;   font-size: 14px;   font-weight: 400;   line-height: 20px; }</pre>	<pre>html, body, a {   font-family:     "Segoe UI", "Open     Sans", serif;   font-size: 14px;   font-weight: 400;   line-height:     20px; }</pre>	<pre>html, body {   font-family:     "Helvetica Neue",     HelveticaNeue,     Helvetica, sans-     serif;   font-size: 14px;   font-weight: 400;   line-height:     20px; }</pre>
(a) Android	(b) Windows Phone	(c) IO's




**Figura 14:** CSS para a tag HTML

A Figura 14 acima representa o código para estilização de todas as fontes inseridas no arquivo, ou seja, todos os links e toda tipografia inserida na aplicação será estilizada com as fontes padrões de cada plataforma.

#### 4.4.4.2 BOTÕES

Como verificado nas aplicações, cada plataforma tem seu padrão de *design* e os códigos CSS abaixo padronizam os botões de acordo com cada plataforma.

<pre> button{   background: #0099CC;   border: none;   border-radius: 2px;   color: rgb(0,0,0);   position: relative;   height: 36px;   min-width: 64px;   padding: 0 16px;   display: inline-block;   font-family:     "Roboto", "Helvetica",     "Arial", sans-serif;   font-size: 14px;   font-weight: 500;   text-transform:     uppercase;   line-height: 1;   letter-spacing: 0;   overflow: hidden;   will-change: box-     shadow, transform;   -webkit-transition:     box-shadow 0.2s cubic-     bezier(0.4, 0, 1, 1),     background-color 0.2s     cubic-bezier(0.4, 0,     0.2, 1), color 0.2s     cubic-bezier(0.4, 0,     0.2, 1);   transition:     box-shadow 0.2s cubic-     bezier(0.4, 0, 1, 1),     background-color 0.2s     cubic-bezier(0.4, 0,     0.2, 1), color 0.2s     cubic-bezier(0.4, 0,     0.2, 1);   outline: none;   cursor: pointer;   text-decoration:     none;   text-align: center;   line-height: 36px;   vertical-align:     middle;   box-shadow: 0 2px 5px     0 rgba(0, 0, 0, 0.16),     0 2px 10px 0 rgba(0, 0,     0, 0.12); } </pre>	<pre> button{   padding: 0     1rem;   height:     2.125rem;   width: 120px;   text-align:     center;   vertical-     align: middle;   background-     color: #2E8DEF;   border: 1px ;   color:     #FFFFFF;   cursor:     pointer;   display:     inline-block;   outline: none;   font-size:     .875rem;   line-height:     1;   margin:     .15625rem 0;   position:     relative;   font-family:     "Segoe UI", "Open     Sans", serif; } </pre>	<pre> button{   border: none;   background: none;   display: block;   cursor: pointer;   text-decoration: none;   font: -apple-system-     caption1;   font-size: 13pt;   -webkit-transition:     all 0.35s ease-out;   transition: all 0.35s     ease-out;   -webkit-box-sizing:     auto;   box-sizing: auto;   padding: 8px 0px;   position: relative;   white-space: nowrap;   overflow: hidden;   text-overflow:     ellipsis;   vertical-align:     middle;   text-align: center;   color: #434343;   border-color: #434343;   background-color:     rgba(255, 255, 255, 0.01);   -webkit-font-     smoothing: antialiased;   margin: 0 5px;   color: #585d63;   border: solid 1px     #585d63;   border-radius: 10px;   display: block;   margin: 0 auto;   padding: 10px 20px;   -webkit-box-sizing:     border-box;   box-sizing: border-     box;   width: auto;   height: auto;   min-width: 200px;   max-width: 300px; } </pre>
---	---	---

		
(a) Android	(b) Windows Phone	(c) IO's

**Figura 15:** CSS dos botões nas diversas plataformas móveis.

Os códigos acima geram os botões nos padrões de cada plataforma. Na Figura 15 (a) segue o padrão de botões Android, que apresenta em seu código a geração do padrão com bordas arredondadas, sombreamento, fonte em *uppercase* além da cor padrão de botão primário Android.

Na Figura 15 (b) o código gera o botão no padrão Windows Phone, o mesmo tem um visual mais limpo, ou seja, não tem a presença de arredondamento nas extremidades e a fonte não tem o padrão sempre maiúscula.

Na Figura 15 (c) o código gera o botão no padrão IO's, comparado ao padrão Android, o botão tem as extremidades mais arredondadas, e em comparação com o padrão Windows Phone a fonte segue o mesmo layout, ou seja, não está em *uppercase*.

Nos códigos CSS é possível identificar as diferenças das aparências como a borda arredondada do botão no padrão Android que pode ser verificado no trecho `border-radius: 2px`. O padrão do Windows Phone é sem borda, por este motivo é que existe a ausência desse elemento no código do mesmo. Para o IO's que tem uma borda mais arredondada, pode ser verificado no código `border-radius: 10px`.

Outra característica do Android que pode ser verificada em código é a presença de sombra do botão que é gerada pelo código: `box-shadow 0.2s cubic-bezier(0.4, 0, 1, 1), background-color 0.2s cubic-bezier(0.4, 0, 0.2, 1), color 0.2s cubic-bezier(0.4, 0, 0.2, 1)`.

#### 4.4.4.3 INPUT

Os inputs podem ser de vários tipos, como *input text*, *checkbox*, *radio button*, e alguns outros que possam ser verificados.

<pre>input{   border: none;   border-bottom: 1px solid   rgba(0,0,0, 0.12);</pre>	<pre>input{   display: inline-   block;   height:</pre>	<pre>input{   text-align: left;   width: 120pt;   height: 15pt;</pre>
---	---	---




<pre>display: block; font-size: 16px; margin: 0; padding: 4px 0; width: 100%; background: none; text-align: left; color: inherit; }</pre>	<pre>2.125rem; position: relative; vertical-align: middle; margin: .325rem 0; line-height: 2; }</pre>	<pre>border-radius: 10px; background-color: #FFFFFF; border-color: #CCCCCC; }</pre>
		
(a) Android	(b) Windows Phone	(c) IO's

Figura 16: Input

O código acima gera os *input text* nos padrões de cada plataforma. Na figura 16 (a) o código gera os padrões Android, na Figura 16 (b) o layout gerado é do Windows Phone, na figura 16 (c) o padrão gerado é do IO's.

Como visto na Figura 16, para representação do padrão Android de formulário com a presença de borda somente na parte inferior é vista no trecho `border-bottom: 1px solid rgba(0,0,0, 0.12)`. Para o Windows Phone, o padrão em grid é representado em `display: inline-block`. Para o IO's as bordas mais arredondadas são verificadas no código: `border-radius: 10px`.

#### 4.4.4.2 SWITCH

Verificado como um input, o switch consiste em uma marcação de um formulário, ou seja, pode ser considerado uma extensão da tag.

<pre>.switch { position: relative; width: 55px; -webkit-user- select:none; -moz- user-select:none; - ms-user-select: none; } .switch-checkbox { display: none; } .switch-label { display: block; overflow: hidden; cursor: pointer; height: 20px; padding: 0; line- height: 20px; border: 0px solid</pre>	<pre>.switch { position: relative; width: 90px; -webkit-user- select:none; -moz-user- select:none; -ms-user- select: none; } .switch-checkbox { display: none; } .switch-label { display: block; overflow: hidden; cursor: pointer; border: 2px solid #999999; border-radius: 0px; } .switch-inner {</pre>	<pre>.switch { position: relative; width: 60px; -webkit-user- select:none; -moz- user-select:none; - ms-user-select: none; } .switch-checkbox { display: none; } .switch-label { display: block; overflow: hidden; cursor: pointer; height: 36px; padding: 0; line- height: 36px;</pre>
---	--	---

<pre> #FFFFFF; border- radius: 30px;     background-color: #9E9E9E; } .switch-label:before {     content: "";     display: block; width: 30px; margin: -5px;     background: #FFFFFF;     position: absolute; top: 0; bottom: 0;     right: 31px;     border-radius: 30px;     box-shadow: 0 6px 12px 0px #757575; } .switch- checkbox:checked + .switch-label {     background-color: #42A5F5; } .switch- checkbox:checked + .switch-label, .switch- checkbox:checked + .switch-label:before {     border-color: #42A5F5; } .switch- checkbox:checked + .switch-label .switch-inner {     margin-left: 0; } .switch- checkbox:checked + .switch-label:before {     right: 0px;     background-color: #2196F3;     box-shadow: 3px 6px 18px 0px rgba(0, 0, 0, 0.2); } </pre>	<pre> display: block; width: 200%; margin- left: -100%;     transition: margin 0.3s ease-in 0s; } .switch-inner:before, .switch-inner:after {     display: block; float: left; width: 50%; height: 30px; padding: 0; line-height: 26px;     font-size: 14px; color: white; font- family: Trebuchet, Arial, sans-serif; font- weight: bold;     box-sizing: border- box;     border: 2px solid transparent;     background-clip: padding-box; } .switch-inner:before {     content: "";     padding-left: 10px;     background-color: #2E8DEF; color: #FFFFFF; } .switch-inner:after {     content: "";     padding-right: 10px;     background-color: #CCCCCC; color: #333333;     text-align: right; } .switch-switch {     display: block; width: 25px; margin: 0px;     background: #000000;     position: absolute; top: 0; bottom: 0;     right: 65px;     transition: all 0.3s ease-in 0s; } .switch-checkbox:checked + .switch-label .switch- inner {     margin-left: 0; } .switch-checkbox:checked + .switch-label .switch- switch {     right: 0px; } </pre>	<pre> border: 2px solid #CCCCCC; border-radius: 36px;     background- color: #FFFFFF;     transition: background-color 0.3s ease-in; } .switch- label:before {     content: "";     display: block; width: 36px; margin: 0px;     background: #FFFFFF;     position: absolute; top: 0; bottom: 0;     right: 22px;     border: 2px solid #CCCCCC; border-radius: 36px;     transition: all 0.3s ease-in 0s; } .switch- checkbox:checked + .switch-label {     background- color: #49E845; } .switch- checkbox:checked + .switch-label .switch-label:before {     border-color: #49E845; } .switch- checkbox:checked + .switch-label .switch-label:before {     right: 0px; } </pre>
--	--	---





**Figura 17:** Switch

A Figura 17 mostra os códigos que geram um *switch* de acordo com o padrão de cada plataforma: Android - (a), Windows Phone - (b) e IO's – (c).

Para criação do switch assim como alguns outros elementos para o desenvolvimento da aplicação é necessário a criação de algumas div's específicas. Para criação do switch por exemplo é necessário incluir a div a seguir:

1	<code>&lt;div class="switch"&gt;</code>
2	<code>&lt;input type="checkbox" name="switch" class="switch-checkbox" id="myswitch" checked&gt;</code>
3	<code>&lt;label class="switch-label" for="myswitch"&gt;</code>
4	<code>&lt;span class="switch-inner"&gt;&lt;/span&gt;</code>
5	<code>&lt;span class="switch-switch"&gt;&lt;/span&gt;</code>
6	<code>&lt;/label&gt;</code>
7	<code>&lt;/div&gt;</code>

**Figura 18:** Switch

Tendo o código acima inserido no HTML é somente aplicar o CSS contido na Figura 17 que para será gerado um switch na aplicação.

#### 4.4.5 TESTE DE VERIFICAÇÃO DE PLATAFORMA

Foi desenvolvido uma página *html* com o código em *javascript* no corpo da página, onde o mesmo faz a verificação de qual plataforma o sistema está a ser a executado.



**Figura 19:** (a) Página Android, (b) Página Windows Phone, (c) Página IO's

Na Figura 19, é representado a verificação da veracidade das informações retornadas pelo javascript de identificação da plataforma. A aplicação foi executada em três diferentes aparelhos: Motorola Moto G com sistema Android 5.0 (Lollipop), Nokia Lumia 820 e um Iphone 4S.

#### 4.5 APLICAÇÃO DE INSERÇÃO DOS PADRÕES

Uma vez mapeado os principais componentes e tags do HTML, passou-se A trabalhar na criação da aplicação para a inserção dos padrões anteriormente estudados e gerando assim uma página HTML5 adaptativa e responsiva a cada plataforma. A aplicação permite ao usuário inserir código HTML de forma genérica, ou seja, um código com as tags básicas para o desenvolvimento de uma aplicação, e o sistema fará a leitura e a manipulação do código, gerando nova interface com elementos e características de cada plataforma e com HTML5 quando possível.

Assim que o usuário digitar seu código ou mesmo fazer o upload para a aplicação o sistema faz a leitura do código digitado e faz a identificação dos componentes que compõem a aplicação.

### 4.5.1 IDENTIFICAÇÃO DOS COMPONENTES

O sistema faz a varredura de todo o código inserido, sendo que está pré-configurado os itens anteriormente citados para a alteração de acordo com a plataforma, como demonstra a figura a seguir:

```
1  if (html.contains("<button")) {
2      ...
3  }
4  if (html.contains("<input")) {
5      ...
6  }
7  if (html.contains("switch")) {
8      ...
9  }
```

**Figura 20:** Identificação dos elementos

Na Figura 20 é possível visualizar a identificação dos elementos. A identificação de uma palavra em uma *string* é feita pela função javascript *.contains()*, que faz a verificação em todo o código digitado se há o termo informado.

### 4.5.2 GERAÇÃO DOS ARQUIVOS CSS

Após a identificação dos componentes, o sistema gera 3 arquivos CSS distintos, um para cada plataforma (Android, Windows Phone, IO's). Assim que os arquivos são gerados, são salvos para que no final do processo seja permitido fazer o download do arquivo.

```
1      if (html.contains("<button")) {
2          codigoCSSAndroid += Android.button();
3          codigoCSSWP += WP.button();
4          codigoCSSIOs += IOs.button();
5      }
6      if (html.contains("<input")) {
7          codigoCSSAndroid += Android.input();
8          codigoCSSWP += WP.input();
9          codigoCSSIOs += IOs.input();
10     }
11     if (html.contains("switch")) {
12         codigoCSSAndroid += Android.geraSwitch();
13         codigoCSSWP += WP.geraSwitch();
14         codigoCSSIOs += IOs.geraSwitch();
15     }
```

**Figura 21:** Funções Javascript que geram o CSS

Na Figura 21 mostra o código que faz a geração dos arquivos CSS. É chamada pelo método de identificação anteriormente citado.

O código recebe como parâmetro o código HTML digitado pelo usuário na aplicação. A aplicação pega o código digitado pelo usuário faz a verificação com o método `.contains()` que faz a verificação se uma palavra está em uma string. Verificado a presença do elemento é adicionado a três variáveis do tipo string que fica responsável por armazenar o código CSS do elemento identificado.

### 4.5.3 ALTERAÇÃO DO CÓDIGO HTML

Para que seja carregado o CSS correto para cada plataforma, a aplicação fará a verificação da tag `<head></head>` na aplicação HTML. Após a identificação, a aplicação irá inserir o código javascript para a importação correta do CSS para determinada plataforma, bem como a alteração de marcações HTML por HTML5 se possível.

```
public static String alterarHtml(String html){  
    String htmlAlterado = html.replace("<head>", "...");  
    return(htmlAlterado);  
}
```

**Figura 22:** Alteração HTML

Na Figura 22 é identificado o método de alteração do código HTML digitado pelo usuário. O código recebe como parâmetro o código HTML digitado e pelo método `.replace` altera a tag `<head>` incluindo novamente a tag mas com a adição do script para identificação da plataforma e inclusão do CSS correto para cada sistema.

### 4.5.4 DOWNLOAD DE ARQUIVOS

Ao final do processo a aplicação permitirá que seja feito o download dos arquivos já configurados para adaptar-se a plataforma a ser executada. Os arquivos gerados serão: HTML alterado para permitir a adaptação e três arquivos CSS, um para cada plataforma.

Os passos citados acima fazem com que um código HTML puro que seja digitado na aplicação seja possível que em dispositivos de sistemas diferente seja alterado o seu layout, ou seja, seja importados arquivos diferentes dependendo de sua plataforma a ser executado.

No desenvolvimento deste trabalho foram encontradas várias dificuldades que podem ser citadas como:

- **Identificação de tamanhos dos dispositivos:** houve-se essa dificuldade inicialmente, pois para se desenvolver uma aplicação web adaptável ao sistema é necessário levar-se em conta o tamanho da tela do dispositivo que está a ser executado, pois cada elemento deverá moldar-se de acordo com o tamanho em que se é exibido;
- **Javascript:** Identificação de elementos javascript que fossem compatíveis nos navegadores dos dispositivos móveis;
- **Fontes e cores:** Levantamento dos estilos e adequação ao suportado no CSS;
- **Dificuldade com a linguagem:** o CSS3 com suas novas técnicas e modos de se tratar cada item torna-se difícil de às vezes de se identificar qual item a se desenvolver.

Por fim, durante o desenvolvimento da aplicação foram feitos vários testes para a verificação e validação dos dados e também a visualização da adaptabilidade e responsividade, sendo em sua maioria obtido resultados adequados nas 3 plataformas propostas.

## 5. CONCLUSÃO

Este trabalho apresentou o desenvolvimento de uma ferramenta de auxílio a desenvolvedores web, que permite o usuário entrar com seu código HTML para que seja identificado os elementos, e seja gerado o código CSS juntamente com a adaptação do código inserido para que a aplicação seja adaptável à plataforma. Com esta aplicação a necessidade de alteração de códigos CSS ficará facilitada, pois é permitido a alteração do CSS de cada plataforma, assim o desenvolvedor se quiser fazer alguma alteração em algum layout terá de forma organizada cada elemento CSS. Assim, os objetivos propostos foram alcançados com sucesso através da ferramenta criada.

O trabalho desenvolvido contribui não somente à desenvolvedores, mas abrange também a comunidade que tenha curiosidade em entender como funciona aplicações adaptáveis e também que estejam em busca de maneiras simples e rápidas para desenvolverem suas próprias aplicações.

Como sugestão para trabalhos futuros pode-se destacar itens como:

- **Identificação de itens** a serem exibidos em uma página dependendo do tamanho da tela. Com o estudo para desenvolver o presente projeto foi verificado que existe a possibilidade de se apresentar somente os itens mais importantes para o usuário, ou seja, apresentar somente o conteúdo que o usuário possa estar procurando. Por exemplo, um site de vendas de bilhetes para cinema. Na aplicação vista por um computador o site apresentar a opção da compra do ingresso, mais todas as opções de filmes que estão em cartaz, endereços dos cinemas próximos, trailers dos filmes, e vários outros dados que possam fazer com que o usuário acabe comprando mais ingressos para outros filmes. Mas quando o site for acessado de um dispositivo com a tela reduzida, a aplicação faça a verificação de itens que levarão o usuário a finalizar sua compra rapidamente.
- **Desenvolvimento de mais elementos:** para este projeto foram desenvolvidos as marcações e componentes mais importantes levantados, e como sugestão para próximos estudos a criação de mais elementos, estilização de tags essenciais e também a criação de novas div's.

Com o estudo e o desenvolvimento deste trabalho pôde-se verificar que a preocupação com a usabilidade do usuário é de máxima importância por todas as plataformas, tendo-se um estudo abrangente da interação homem-máquina, desde testes aplicados como verificação de estudos anteriormente feitos.

Este trabalho serve de inspiração para próximas pesquisas, inspira mais desenvolvedores a criar mais layouts para várias e várias tags existentes e até mesmo o desenvolvimento de mais div's que possam fazer a criação de interface web para itens presentes nas aplicações mobile.

## REFERÊNCIAS

ANTONIO BARBOSA, Nelson. **Padrões de Projeto de Interface para Aplicativos Android para o Google TV (2013)**. Disponível em: <<http://www.espweb.uem.br/site/files/tcc/2012/Nelson%20Antonio%20Barbosa%20-%20Padroes%20de%20projeto%20de%20interface%20para%20aplicativos%20Android%20para%20Google%20TV.pdf>>; Acesso em: outubro de 2015.

APPLE. **Negócios - Apple (2015)** Disponível em: <<https://www.apple.com/br/business/accelerator/>> Acesso em: setembro de 2015.

APPLE STORE. **Guia de Marketing da Apple Store (2015)**. Disponível em: <<https://developer.apple.com/app-store/marketing/guidelines/br/#messaging>> Acesso em: setembro de 2015.

BARWINSKI, Luísa. **O que é HTML5? (2009)** Disponível em: <<http://www.tecmundo.com.br/navegador/2254-o-que-e-html-5-.htm>>; Acesso em: outubro de 2015

BIDA DE OLIVEIRA, Frederico. **Interfaces Usuário-Máquina (2011)**. Disponível em <[https://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1618984280\\_Apostila-Interfaces-Homem-Maquina.pdf](https://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1618984280_Apostila-Interfaces-Homem-Maquina.pdf)> Acesso em: setembro de 2015.

CARLA, Giani; FERREIRA, Mauricio; SANT'ANA, Nilson. **Utilização de Interfaces Adaptativas para a Computação Móvel. (2004)** Disponível em: <<http://mtc-m18.sid.inpe.br/col/lac.inpe.br/worcap/2004/09.30.13.53/doc/artigoGiani.pdf>>; Acesso em: setembro de 2015

CSS3Maker. **CSS3Maker**. Disponível em: <<http://www.css3maker.com/>> Acessado em: novembro de 2015

DATA FOLHA. **43 milhões de brasileiros acessam internet por dispositivos móveis.** São Paulo. 2014. Disponível em: <<http://datafolha.folha.uol.com.br/mercado/2014/01/1400618-43-milhoes-de-brasileiros-acessam-internet-por-dispositivos-moveis.shtml>>; Acesso em: setembro de 2015



DEVELOPER ANDROID. Disponível em: <<http://developer.android.com/index.html>>; Acesso em: outubro de 2015

EIS, Diego. **Mobile First – A arte de pensar com foco (2011)**. Disponível em: <<http://tableless.com.br/mobile-first-a-arte-de-pensar-com-foco/>>; Acesso em: setembro de 2015

FRANCISCO RIBEIRO, D. **Estudo de Interface Humano-Máquina em Dispositivos Móveis (2007)**. Disponível em: <[https://projetos.inf.ufsc.br/arquivos\\_projetos/projeto\\_521/ArtigoDaniel.pdf](https://projetos.inf.ufsc.br/arquivos_projetos/projeto_521/ArtigoDaniel.pdf)> Acesso em: outubro de 2015

HAMANN, Renan. **iOS, Android e Windows Phone: números dos gigantes comparados [infográfico] (2014)**. Disponível em: <<http://www.tecmundo.com.br/sistema-operacional/60596-ios-android-windows-phone-numeros-gigantes-comparados-infografico.htm>>; Acesso em: agosto de 2015

GAVASSO, Gabriel; CALDERARO, Rodrigo; SPOLAVORI, Carlos Eduardo. **A usabilidade em dispositivos móveis. (2006)** Disponível em: <<http://guaiba.ulbra.br/seminario/eventos/2006/artigos/sistemas/143.pdf>>; Acesso em: setembro de 2015

Google. **Material Design**. Disponível em: <<https://www.google.com/design/spec/material-design/introduction.html>>; Acesso em: outubro de 2015

JOHNSON, Thienne. **Java para dispositivos móveis – Desenvolvendo aplicações com J2ME**. Disponível em: <<http://novatec.com.br/livros/javamoveis/capitulo9788575221433.pdf>>; Acesso em: outubro de 2015

LA, N. **RESPONSIVE DESIGN IN 3 STEPS. (2011)** Disponível em: <<http://webdesignerwall.com/tutorials/responsive-design-in-3-steps>>; Acesso em: outubro de 2015

MACHADO NETO, O. J. **Usabilidade da interface de dispositivos móveis: heurísticas e diretrizes para o design**. São Carlos, 2013.

MACHADO, Eugenio. **Interação Homem-Máquina (2013)**. Disponível em: <<http://homaquina.blogspot.com.br/2013/02/interface-inteligente.html>>; Acesso em: setembro de 2015

MEDEIROS, Higor. **Projetando e criando Aplicativos para Dispositivos Móveis**. Disponível em: <<http://www.devmedia.com.br/projetando-e-criando-aplicativos-para-dispositivos-moveis/30671>>; Acesso em: setembro de 2015

MICROSOFT. **Princípios de design da Microsoft**. Disponível em: <<http://msdn.microsoft.com/pt-br/library/windows/apps/hh781237.aspx>>; Acesso em: outubro de 2015

MOSIMANN NETTO, Max. **Design para Dispositivos Móveis – O primeiro passo para um sistema bem sucedido, é a interação com o usuário**. Disponível em: <<http://www.linhadecodigo.com.br/artigo/648/design-para-dispositivos-moveis-o-primeiro-passo-para-um-sistema-bem-sucedido-e-a-interacao-com-o-usuario.aspx>>

NICOLAS. **Facebook n'est pas (encore) à jour pour les iPhone6 (2014)**. Disponível em: <<http://www.igen.fr/app-store/2014/09/facebook-nest-pas-encore-jour-pour-les-iphone-6-87892>>; Acesso em: novembro de 2015

OLHAR DIGITAL. **10 motivos para um aplicativo ser rejeitado pela Apple (2014)**. Disponível em: <<http://olhardigital.uol.com.br/noticia/10-motivos-para-um-aplicativo-ser-rejeitado-pela-apple/43838>>; Acesso em: setembro de 2015

OLIVEIRA, G. A.; ARAUJO, W. J. **USAR OU NÃO USAR – qual a relevância das meta tags na recuperação da informação pelos mecanismos de busca? (2012)**. Disponível em: <<http://www.biblionline.ufpb.br/ojs/index.php/biblio/article/view/11865/7510>>; Acesso em: novembro de 2015

PLAFFO. **The best App: Aprile 2013 – Le migliori applicazioni per Windows Phone scelte da Plaffo**. Disponível em: <<http://www.plaffo.com/2013/05/the-best-app-aprile-2013-le-migliori-applicazioni-per-windows-phone-scelte-da-plaffo/>>; Acesso em: outubro de 2015

PROTO.IO. **On/Off FlipSwitch**. Disponível em: <<https://proto.io/freebies/onoff/>> Acesso em: novembro de 2015.

SALLES MANZOTTI, Caio. **Design de Interface em Dispositivos Móveis.** Disponível em:

<[http://www.ensaiovisual.com.br/media/caio\\_manzotti\\_design\\_interface.pdf](http://www.ensaiovisual.com.br/media/caio_manzotti_design_interface.pdf)>;

Acesso em: setembro de 2015

SÁNCHEZ, I. **Conselhos essenciais em HTML5, CSS, responsivo e SEO (2013).**

Disponível em: <<http://imasters.com.br/search/seo/conselhos-essenciais-em-html5-css-responsivo-e-seo/?trace=1519021197&source=single>> Acesso em: novembro de 2015

SILVA, A. A. P. **DESIGN RESPONSIVO: TÉCNICAS, FRAMEWORKS, E FERRAMENTAS.** (2014) Disponível em:

<<http://uniriodb.uniriotec.br/tcc/201412Almeida.pdf>>; Acesso em: setembro de 2015

SOARES DE SOUZA, Luciano; DE MESQUITA SPINOLA, Mauro. **Requisito de usabilidade em projetos de interface centrado no usuário de software de dispositivos móveis.** (2006) Disponível em:

<[http://www.abepro.org.br/biblioteca/ENEGEP2006\\_TR470319\\_7324.pdf](http://www.abepro.org.br/biblioteca/ENEGEP2006_TR470319_7324.pdf)>; Acesso em: outubro de 2015

TECHTUDO. **Android Lollipop: conheça o sistema operacional móvel do Google (2015).** Disponível em: <<http://www.techtudo.com.br/tudo-sobre/android-l.html>>; Acesso em: setembro de 2015

TEIXEIRA, F. **O que é Responsive Web Design? (2011).** Disponível em <<http://arquiteturadeinformacao.com/mobile/o-que-e-responsive-web-design/>> Acesso em: novembro de 2015.

TUDO CELULAR. **Mais de 1,2 bilhão de smartphones foram vendidos em 2014 e Samsung continua caindo (2015).** Disponível em: <

<http://www.tudocelular.com/android/noticias/n50616/Segundo-pesquisa-continua-em-declinio.html>> Acesso em: dezembro de 2015

W3CSCHOOL. **THE WORLD'S LARGEST WEB DEVELOPER SITE.** Disponível em: <[http://www.w3schools.com/jsref/prop\\_nav\\_useragent.asp](http://www.w3schools.com/jsref/prop_nav_useragent.asp)>; Acesso em: setembro de 2015