

Adaptação de Interfaces Web para Dispositivos Móveis

Darlan E. F. da Silva^{1,2}, Bruno B. Boniati¹, Roberto Franciscatto¹

¹Colégio Agrícola de Frederico Westphalen – Universidade Federal de Santa Maria
Caixa Postal 54 – 98.400-000 – Frederico Westphalen – RS – Brasil

²Graduação Tecnológica em Sistemas para Internet – Universidade Federal de Santa Maria
darlanfelisberto@gmail.com, {bruno,roberto}@cafw.ufsm.br

Abstract. *The adaptation of interfaces of web applications for mobile devices is a new requirement that web software developers need to consider. Through this work we intend to make a counterpoint on the development of native applications for mobile devices and adaptation of web interfaces to such devices. Through a case study aims to apply a set of best practices to enable a single web application can run on different devices satisfactorily.*

Resumo. *A adaptação de interfaces de aplicações web para dispositivos móveis é um novo requisito que desenvolvedores deste tipo de software precisam considerar. Por meio deste trabalho pretende-se fazer um contraponto acerca do desenvolvimento de aplicações nativas para dispositivos móveis e a adaptação de interfaces web para tais dispositivos. Através de um estudo de caso pretende-se aplicar um conjunto de boas práticas para permitir que uma mesma aplicação web possa ser executada em diferentes dispositivos de forma satisfatória.*

1. Introdução

Um dispositivo móvel é todo e qualquer equipamento que a partir da alimentação de uma bateria pode ser levado a qualquer lugar, sem a presença de cabos e é munido de recursos computacionais que permitem ao seu utilizador programá-lo, são exemplos os *notebooks*, *palmtops* e recentemente *tablets* e *smartphones* [Morimoto 2009]. Observa-se que com o advento e a popularização destes dispositivos, cada vez mais aplicações são migradas ou desenvolvidas especificamente para tais plataformas.

O desenvolvimento de *software* para dispositivos móveis é, em muitos casos, específico para cada sistema operacional e plataforma de hardware [Matos 2008]. Da mesma forma entende-se que cada vez mais, as aplicações web precisam considerar a sua execução em tais ambientes, que apresentam características de interface com o usuário um tanto distintas, como por exemplo, tamanho da tela reduzidos, e recursos de entrada diferenciados (teclados específicos, telas sensíveis ao toque, etc.).

Diante deste contexto o presente trabalho procura fazer um levantamento bibliográfico acerca das tecnologias envolvidas para a produção de *software* em diferentes plataformas de dispositivos móveis e estudar técnicas e padrões para a adaptação de interfaces web em tais equipamentos. Para tanto, um sistema computacional foi desenvolvido para auxiliar a coleta de avaliações de trabalhos. Tal sistema utiliza-se de um formulário para a coleta de notas e pretende-se utilizá-lo tanto em dispositivos tradicionais, como *desktops*, mas especialmente em dispositivos móveis (de modo que a avaliação possa ser feita a campo).

A intenção do trabalho é que a aplicação desenvolvida possa ser utilizada para a avaliação de trabalhos durante a Mostra de Ciências do Colégio Agrícola de Frederico Westphalen (CAFW), evento tradicional da instituição onde o projeto foi desenvolvido. A partir deste estudo pretende-se demonstrar as diferenças existentes entre o

desenvolvimento de aplicações nativas e aplicações web. Ao final do texto o protótipo desenvolvido será apresentado.

2. Plataformas de Desenvolvimento de Aplicações para Dispositivos Móveis

Na medida em que surgem novas tecnologias aparecem novas maneiras de se fazer o velho de novo. Geralmente quem mais se beneficia com o surgimento de novas tecnologias é o usuário final, que quer ter o mais recente e o que há de melhor em seu dispositivo móvel [Freire 2012]. Quando estas novas tecnologias são bem fundamentadas e bem elaboradas, elas trazem ao usuário uma nova experiência, tornando mais agradável, simples e atrativa a utilização de seu dispositivo.

Com o montante de plataformas para o desenvolvimento existentes hoje no mercado global, ocorre uma grande oferta de dispositivos dos mais variados tipos, tamanhos, modelos e marcas, executando em si diversos sistemas operacionais, com diversas tecnologias de desenvolvimento de software. Com essa diversidade de tecnologias para o desenvolvimento de *software*, torna-se cada vez mais complicada a tarefa do desenvolvedor, de manter compatível uma mesma versão de aplicação entre diversas plataformas [IBM 2012]. Tal situação exige que um profissional que deseje desenvolver um aplicativo para mais de uma plataforma precise ter conhecimento de diversas linguagens de programação e ferramentas de desenvolvimento.

O desenvolvimento de aplicativos para dispositivos móveis iniciou juntamente com a criação do protocolo de rede WAP (*Wireless Application Protocol*). Esse protocolo foi criado por um conglomerado de empresas para tornar possível a utilização da internet em dispositivos móveis. Porém as aplicações que utilizam essa tecnologia utilizam uma linguagem de marcação baseado em XML¹, o WML (*Wireless Markup Language*). O WML não agradou os desenvolvedores e devido a isso, o WAP não decolou. Então em 2002, surgiu o WAP 2.0, que trouxe consigo suporte a toda a pilha TCP/IP, assim como suporte a *xHTML Basic*, mantendo o suporte a WML e todas as camadas de rede da pilha WAP [Tanenbaum 2003].

O que se observou na sequência foi o surgimento de diferentes plataformas para desenvolvimento de aplicações nativas para dispositivos móveis. Uma aplicação nativa é um software criado utilizando uma linguagem de programação compilada, (ex. C, C++, Objective-C, Java, etc.) de forma específica para determinada arquitetura (hardware e/ou sistema operacional). Esse tipo de aplicação costuma ser adquirida e baixada de uma loja virtual. Aplicações nativas normalmente não possuem dependência de uma rede permanente de dados e conseguem acessar recursos avançados de hardware do dispositivo (câmeras, GPS, acelerômetro, etc.)

Devido não haver uma padronização, entre as novas plataformas que surgiram, a mesma aplicação pode não funcionar em duas plataformas distintas. Da mesma forma existem plataformas que são específicas para um determinado *hardware* e/ou sistema operacional. A incompatibilidade existente entre as plataformas existe devido à maneira como elas são concebidas, devido à linguagem utilizada e até mesmo devido aos nomes das classes/métodos que diferem umas das outras ou simplesmente inexistem em plataformas distintas.

Em [Lionbridge 2012] são listadas algumas características típicas de uma aplicação nativa: visual padronizado com a interface do dispositivo, acesso direto ao *hardware* através da API do dispositivo, execução *off-line* e alto custo para o seu desenvolvimento. Este último item ocorre devido a necessidade de disponibilizar a aplicação para execução em diversos sistemas operacionais, sem a existência de uma

¹ *eXtensible Markup Language* (XML) é uma linguagem para marcação utilizada para descrever e representar dados.

compatibilidade binária entre eles. São exemplos de plataformas de dispositivos móveis: Android², IOS³, BlackBerry OS⁴, Windows Phone⁵ e Java ME⁶.

3. Aplicações Web e Dispositivos Móveis

Uma *WebApp* é uma aplicação criada para ser acessada por meio de um navegador, a partir de um dispositivo móvel ou *desktop* [Fernandes 2011]. Tais aplicações podem ser utilizadas a partir de qualquer dispositivo, com qualquer tamanho e sistema operacional, devido ao fato da mesma abstrair a plataforma adjacente e ser exibida e executada por um navegador. Boa parte dos navegadores utilizam padrões definidos pelo W3C (*World Wide Web Consortium*), de forma que o código que é renderizado em um navegador da Apple seja renderizado da mesma maneira que um navegador da Mozilla.

A W3C foi criada por diversas empresas comerciais e educacionais, com o objetivo de definir padrões nas áreas do desenvolvimento pra *web*. Padronizar e definir normas para o desenvolvimento para web foi necessário em função da existência de diversos navegadores criados de forma independente, sendo que alguns não compartilham do mesmo motor de renderização. Sem essa padronização seria necessário possuir diversas versões de *WebApps*, uma para cada navegador, da mesma forma como acontece com as aplicações nativas [Oliveira 2010].

A principal vantagem da *WebApp* contra a *App Nativa* é a padronização junto a W3C [Mahemoff 2011], garantindo que todos os browsers executem e entendam da mesma maneira os padrões HTML⁷, CSS⁸ e JavaScript⁹, sendo estes as principais tecnologias para o seu desenvolvimento. Existem ainda algumas tecnologias proprietárias que podem ser utilizadas em algumas plataformas, como o Adobe Flash¹⁰ e Microsoft Silverlight¹¹. Tais tecnologias, para serem utilizadas exigem a instalação de *plug-ins* pelo usuário do dispositivo móvel, em algumas situação não há versões para algumas plataformas, tornando a sua utilização não recomendável.

4. Técnicas para Adaptação de Interfaces

O modelo de aplicação apresentado por este trabalho utiliza as melhores práticas de desenvolvimento para *web*, com o foco em dispositivos móveis. Com este objetivo foram consideradas algumas das recomendações mais relevantes criadas pela W3C [W3C 2008] e [W3C 2010]. Essas normas visam permitir uma maior facilidade para a exploração de recursos provida pelos dispositivos, proporcionando uma melhor experiência por parte do usuário.

Algumas dessas especificações podem trazer uma série de problemas de compatibilidade da aplicação, com dispositivos móveis antigos, tornando inviável a sua utilização. Esses problemas são ocasionados pela utilização de tecnologias muito recentes, que os dispositivos antigos não possuem o suporte completo ou havendo a ausência dessa tecnologia por completo. Devido a esses problemas, essas especificações

² Android, http://www.openhandsetalliance.com/android_overview.html ou <http://www.android.com>

³ Apple IOS, <http://www.apple.com/br/ios/>

⁴ BlackBerry OS, <http://blackberry.com/services/blackberry7/>

⁵ Microsoft Windows Phone, <http://www.microsoft.com/windowsphone/>

⁶ Java Micro Edition, <http://www.oracle.com/javame>

⁷ *HyperText Markup Language*, linguagem de marcação utilizada para estruturar o conteúdo de aplicações e páginas web.

⁸ *Cascading Style Sheets*, linguagem para definição dos estilos e efeitos visuais do conteúdo web.

⁹ JavaScript, linguagem de programação interpretada pelo navegador e que em conjunto com HTML e CSS permite utilizar recursos programáveis em páginas web.

¹⁰ Adobe Flash, <http://get.adobe.com/br/flashplayer/>

¹¹ Microsoft Silverlight, <http://www.microsoft.com/silverlight/>

não foram tidas como obrigatórias, mas sim como um padrão que pode ser seguido por desenvolvedores de aplicações *web* para dispositivos móveis.

As melhores práticas foram divididas em diversas especificações, entretanto aqui serão apresentadas somente as especificações de maior relevância para o projeto e que foram utilizadas para a construção do protótipo. Como a *WebApp* utiliza-se de acesso a internet (provida pelo dispositivo móvel ou pela operadora), entende-se que a mesma poder estar sujeita a atrasos em função da baixa qualidade e/ou da pequena capacidade de transferência dos dados pela rede. Devido a isso foram seguidas as especificações redigidas pela W3C que dizem respeito à *cookies*, *Ajax*, otimização de requisição e tempo de carregamento das telas.

Os *cookies* (“testemunhos de conexão”) são muito utilizados para armazenar informações no lado cliente (navegador), são armazenados em arquivos que são enviados ao servidor a cada requisição de uma nova tela, o que motivou a redução de seu uso pela aplicação. Em seu lugar foi utilizado o conceito de sessões. Uma sessão é um arquivo armazenado no servidor da aplicação, esse arquivo contém todas as variáveis correspondentes à sessão vigente e no cliente é armazenado apenas um *cookie* contendo o identificador da sessão. Construindo a aplicação com um número limitado de *cookies*, garante-se que o cliente necessite enviar menos informações para o servidor e conseqüentemente economizar na utilização da franquia de dados.

Outra técnica muito utilizada em sites para *desktop* e recomendada pela W3C para a utilização em dispositivos móveis, é a utilização de *Ajax* para realizar requisições como servidor. Com a sua utilização é eliminado uma grande parte do tempo levado para o cliente realizar o carregamento da página, pois apenas parte da estrutura da página é carregada, tornando a navegação pelo cliente muito mais rápida. No presente trabalho, pretendia-se manter a compatibilidade com dispositivos antigos, por isso essa técnica não foi utilizada.

Na tentativa de realizar o mínimo possível de requisições ao servidor e mantendo o funcionamento correto da *WebApp*, faz-se necessário a realização de pelo menos três redirecionamentos realizados pelo cliente: *login*, seleção do trabalho a ser avaliado e submissão das notas para gravação. Objetivando otimizar o tempo de carregamento das páginas e garantir uma maior velocidade na exibição do conteúdo, foi abolido a utilização de imagens no protótipo. Em sendo necessária sua utilização, o W3C recomenda o uso do *AppCache*, uma das especificações do HTML5 que permite a utilização de elementos que compõe o site a partir de uma *cache* mantida pelo navegador.

Na composição do conteúdo a ser exibido pelo *WebApp*, foram adicionados apenas elementos necessários e mínimos para a utilização do aplicativo (ex. nome e um pequeno resumo do trabalho - exibido através de um pequeno quadro do tipo *tooltip*). Outra recomendação que foi utilizada é o conceito de *URLs* amigáveis, mantendo-as curtas, e simplificando sua digitação em alguns dispositivos móveis com teclados ou dispositivos de entrada limitados.

Em relação a organização do layout a W3C recomenda a não utilização de frames e tabelas aninhadas substituindo-os por contêineres que utilizam propriedades CSS. Em relação aos valores de medidas (dimensões dos contêineres e outros elementos) recomenda-se o uso de medidas relativas, como em¹², ex¹³, de forma que o

¹² em é uma medida relativa, a um tamanho de determinada propriedade como *font-size*, *margin*, *width*, previamente defino ou não. Por exemplo, a propriedade “font-size” caso ela não seja defina o padrão dela em muitos navegadores é 16px, então 16px=1em, 2em=32px.

¹³ “ex” é equivalente ao tamanho da letra “x” em minúscula da fonte vigente.

navegador do dispositivo móvel possa adaptar o conteúdo às dimensões da tela [Silva 2011].

5. Estudo de Caso: Avaliação de Trabalhos da Mostra de Ciências do CAFW

Baseado nas referências apresentadas e objetivando o desenvolvimento de uma *WebApp* que se adapte a diferentes dispositivos, desenvolveu-se um protótipo para ser utilizado em avaliações de trabalhos. O protótipo consiste de uma ficha de avaliação que permitirá que diferentes avaliadores possam atribuir notas em critérios pré-definidos aos trabalhos visitados a campo. O estudo de caso consiste na utilização do protótipo desenvolvido em substituição à ficha de papel utilizada nas edições anteriores do evento.

Optou-se pelo desenvolvimento de uma *WebApp* em função dos requisitos do software, em especial a necessidade de o mesmo ser executado sobre diferentes plataformas. Também se considerou a presença de rede de internet no local do evento, eliminando a necessidade da aplicação ser executada de forma off-line. Foram utilizadas tecnologias livres e multiplataforma. Caso a opção fosse pelo desenvolvimento de uma aplicação nativa, seriam necessários conhecimentos em diferentes plataformas de desenvolvimento e mesmo assim a mesma não seria tão abrangente como uma *WebApp*. Neste caso seria necessário reescrever a aplicação para cada plataforma, uma vez que não há compatibilidade entre as mesmas. Outro fator que foi considerado é o fato de que uma *WebApp* é acessada diretamente pelo navegador, não existindo necessidade de instalação prévia no dispositivo.

O funcionamento da aplicação é bastante simples, por meio da identificação de um usuário avaliador (previamente cadastrado) o sistema apresenta a relação de trabalhos a serem avaliados. A lista de trabalhos de cada avaliador é definida previamente pelo próprio *software* considerando alguns critérios como categorias de trabalhos preferenciais (ensino médio, ensino técnico ou ensino de graduação) e também os participantes do mesmo (impedindo que um avaliador avalie trabalhos onde o mesmo é integrante da equipe). A figura 1 a janela de escolha do trabalho a ser avaliado. A primeira parte da figura (a) demonstra a interface da aplicação em um emulador Android e a segunda parte (b) ilustra a mesma tela da interface sendo apresentada em *notebook* no navegador Firefox.

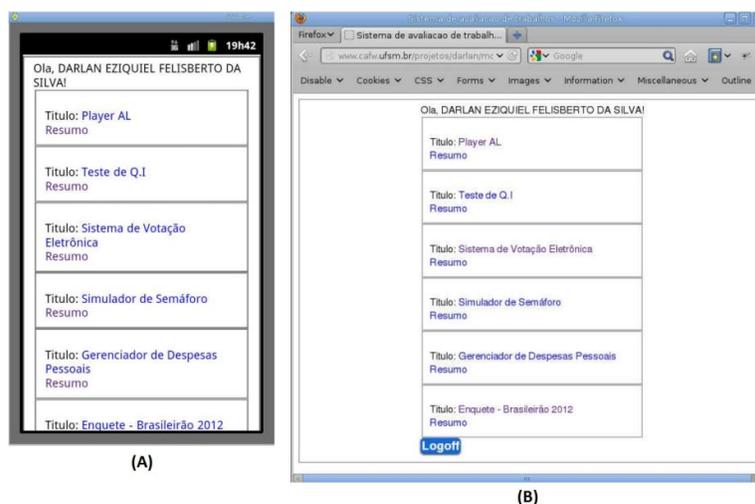


Figura 1. Janela para Escolha do Trabalho a ser Avaliado

Uma vez que o trabalho foi selecionado, na tela seguinte o usuário deverá informar as notas para os critérios pré-definidos. De forma a facilitar a entrada de dados, especialmente em dispositivos móveis desprovidos de teclado, optou-se pela utilização de caixas de seleção onde o usuário apenas escolhe uma nota sem a necessidade de digitação. A figura 2 ilustra a janela onde a avaliação é realizada. Pode-se observar que o avaliador tem duas opções: gravar as notas para posterior alteração ou submetê-las como definitivas, neste caso estando aptas para compor a pontuação final do trabalho.

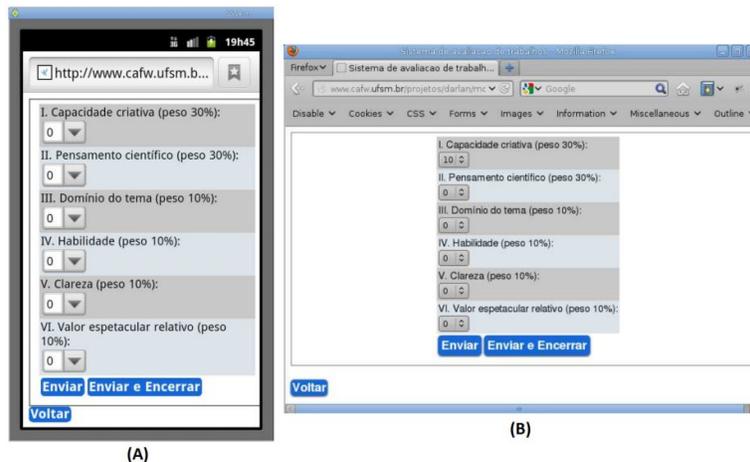


Figura 2. Janela de Avaliação dos Trabalhos

6. Considerações Finais

Ao longo deste trabalho discutiu-se a problemática que muitos desenvolvedores de aplicações para dispositivos móveis se deparam: desenvolver de forma específica e nativa para uma determinada plataforma (Android, IOS, WindowsPhone, etc.) ou utilizar tecnologias compatíveis com diferentes dispositivos. Ao longo do texto algumas diferenças dessas duas formas de se desenvolver aplicações foram apresentadas.

É importante considerar que dependendo do escopo do projeto uma alternativa pode ser melhor do que a outra, como por exemplo, se há necessidade de acessar diretamente o hardware do dispositivo ou se existem restrições de conexão com a rede, uma aplicação nativa certamente será mais interessante. Por outro lado, se há necessidade permanente de conexão com a internet (aplicação on-line) e não existem necessidades específicas de hardware, uma aplicação *web* (*WebApp*) pode ser tornar uma alternativa interessante.

No caso de uma *WebApp*, é importante ressaltar que a mesma não precisa ser específica para determinada plataforma, e neste caso é desejável que sua interface com o usuário se adapte ao dispositivo em que está sendo executada (se for um *tablet* ou um *desktop* poderá utilizar a tela de forma mais ampla, se for um *smartphone* ou PDA precisará compactar sua apresentação visual). De forma a validar as técnicas debatidas ao longo do trabalho para adaptar interfaces web em dispositivos móveis, um protótipo de uma *WebApp* foi desenvolvido. Através de tal aplicação uma ficha de avaliação é apresentada ao usuário podendo ser utilizada para avaliação de trabalhos a campo. Este é o caso dos requisitos apresentados pela Mostra de Ciências do CAFW, evento anual tradicional da instituição onde o trabalho foi desenvolvido.

Após a implementação do protótipo, testes foram executados em diferentes dispositivos para demonstrar a aderência da solução aos requisitos do evento ao qual o mesmo se propunha a ser utilizado. Uma avaliação de trabalhos simulada foi proposta

em uma disciplina de forma a incentivar os alunos a utilizarem o sistema para avaliar os trabalhos dos colegas. Ao longo do teste validou-se o software desenvolvido assim como algumas incompatibilidades entre dispositivos foram observadas.

Cabe ressaltar que o software desenvolvido pode ser utilizado para qualquer tipo de avaliação de trabalhos onde há necessidade de um avaliador atribuir notas utilizando critérios pré-definidos em trabalhos inscritos. A característica da mobilidade permite ao avaliador utilizar o sistema ao mesmo tempo em que visita os trabalhos (situação característica da Mostra de Ciências do CAFW). Espera-se que o sistema seja utilizado na próxima edição da Mostra de Ciências, prevista para Novembro de 2012 onde o mesmo será validado por um conjunto maior de usuários e dispositivos e será utilizado para apurar os resultados finais do evento.

Como sugestão de trabalhos futuros, seria importante estudar como organizar a interface do protótipo de maneira que possam ser utilizadas as mais recentes tecnologias para a *web*, seguindo todas as boas praticas definidas pela W3C. Também se pretende estudar *frameworks* e projetos relacionados (ex. Wurfl¹⁴ e Bootstrap¹⁵), que em função de configurações prévias conseguem reorganizar a interface da aplicação em função do dispositivo utilizado.

Referências

- Fernandes, B. (2011), “Webapp, Já Posso Investir Nisso”. In: *COLAPHP – Congresso Latino Americano de PHP*, Foz do Iguaçu – PR.
- Freire, F. (2011) “Faça-se Presente no Mundo Móvel”. In: *Revista Wide*, Ed. 83.
- IBM (2012) “Native, Web or Hybrid Mobile App Development? - White Paper”, <ftp://public.dhe.ibm.com/software/pdf/mobile-enterprise/WSW14182USEN.pdf>.
- Lionbridge (2012). “Mobile Web Apps vs. Mobile Native Apps: How to Make the Right Choice”, <http://en-us.lionbridge.com/kc/mobile-web-apps-vs-mobile-native-apps.htm>
- Mahemoff, M. (2011), “HTML5 vs Native: the mobile app debate”, <http://www.html5rocks.com/en/mobile/nativedebate/>.
- Matos, P. M. C. P. P. (2008), “Visualização de Informação Geo-referenciada em Dispositivos Móveis”, Dissertação de Mestrado, Universidade de Lisboa.
- Morimoto, C. E. (2009) *Smartphones Guia Prático*, GDH Press e Sul Editores.
- Oliveira, L. (2010) “Web Mobile – a Internet como Unificadora de Apps Móveis”, <http://www.mobileasy.com.br/dotheevolution/2010/11/25/web-mobile-a-internet-como-unificadora-de-apps-moveis/>
- Silva, M. S. (2011) *CSS3: Desenvolva Aplicações Web Profissionais com o uso dos Poderosos Recursos de Estilização do CSS3*, Editora Novatec.
- Tanenbaum, A. S. (2003), *Computer Networks*, Prentice Hall, 4ª Edição.
- W3C (2008), “Mobile Web Best Practices 1.0”, <http://www.w3.org/TR/mobile-bp>
- W3C (2010), “Mobile Web Application Best Practices”, <http://www.w3.org/TR/mwabp>

¹⁴ Projeto Wurfl - Wireless Universal Resource FiLe - www.scientiamobile.com/

¹⁵ Twitter Bootstrap - <http://twitter.github.com/bootstrap/>