

Do design à implementação: acessibilidade de websites e sistemas de informação digitais

From design to implementation: accessibility on websites and digital information systems

Pinheiro, Mauro; MSc.; Universidade Federal do Espírito Santo
mauropin@pobox.com

Resumo

Este artigo discute a acessibilidade em sistemas de informação, considerando a multiplicidade de meios de acesso a esses sistemas a partir do amadurecimento da computação pervasiva. São comparadas duas metodologias de desenvolvimento de websites, baseadas no princípio da *tolerância a falhas*, a saber: metodologia da *degradação graciosa* e metodologia de *melhorias progressivas*. Utiliza o caso do site Yahoo! como exemplo de aplicação de melhorias progressivas para garantir a acessibilidade. Ao final, destaca a importância de um maior envolvimento dos designers com questões técnicas de implementação de sistemas de informação digitais.

Palavras Chave: computação pervasiva; acessibilidade; webdesign.

Abstract

This article discusses the design of accessible information systems, taking into perspective the increasing ways of accessing these systems, due to the growth of pervasive computing. It compares two methodologies for developing websites, based on the principles of "fault tolerance": the "graceful degradation" methodology and the "progressive enhancements" methodology. It demonstrates the progressive enhancements can grant accessibility, using Yahoo!'s website as an example. As a conclusion, it highlights the need for designers getting involved with technical aspects concerning the implementation of digital information systems.

Keywords: *pervasive computing; accessibility; webdesign.*

Introdução

Há algum tempo alcançamos uma certa maturidade no projeto de sites no Brasil. O que na década de 90 era um esforço concentrado na figura quase mítica do *webmaster* hoje é um trabalho colaborativo entre arquitetos de informação, designers de interação, designers de experiência etc. A variedade de competências e profissionais envolvidos é grande, sendo um tema debatido entre os que atuam na área (PINHEIRO, 2007).

Avançamos conceitualmente, a tecnologia envolvida na produção de sistemas de informação interativos também evoluiu e multiplicaram-se as possibilidades de acesso a esses sistemas. Telefones celulares, consoles de *videogame*, televisores; hoje, diversas plataformas acessam a Internet e a Web não é mais uma exclusividade dos *browsers* em computadores do tipo *desktop*. Nos primeiros anos da Web, a questão limitava-se basicamente à adequação dos projetos aos diferentes navegadores existentes no mercado. Hoje a situação é mais complexa, envolvendo plataformas com características e limitações diferenciadas. No entanto, muitos projetos ainda desconsideram essa variedade de ambientes e de possibilidades de acesso.

A despeito de ser um tema amplamente debatido (SOUZA, 2007; SPELTA, 2007), a acessibilidade¹ dos sites ainda é ignorada por alguns profissionais de mídias interativas. Parte desse problema vem do desconhecimento de questões supostamente técnicas, no que se refere ao desenvolvimento e implementação dos sistemas, por parte daqueles que projetam a experiência do usuário.

No âmbito da arquitetura e do design, o conhecimento das tecnologias utilizadas no desenvolvimento dos projetos é condição *sine qua non* para uma atuação profissional competente. Um arquiteto que projete habitações em concreto armado deve conhecer as características dessa técnica construtiva. Não se espera que ele execute cálculos estruturais ou revire concreto em uma obra, mas é fundamental conhecer as limitações da tecnologia envolvida no processo construtivo pretendido, para que possa projetar habitações viáveis e para, idealmente, racionalizar a produção de forma a economizar tempo e recursos na sua construção. De maneira análoga, não se espera que um designer que projete sistemas interativos domine profundamente os aspectos técnicos envolvidos em sua produção, mas é necessário que tenha um conhecimento que vá além do senso comum sobre a tecnologia a ser utilizada. As etapas de design da interação e de arquitetura de informação, que definem como se dá a interação do usuário com o sistema (GARRET, 2002), devem ser pensadas tendo em vista as limitações tecnológicas dos meios de acesso existentes. No entanto, muitos designers permanecem alheios às questões técnicas, preocupando-se especialmente com as camadas mais evidentes dos sistemas interativos. Alguns designers chegam a se envolver unicamente

¹ Acessibilidade é a qualidade ou caráter do que é acessível, isto é, algo a que se pode ter acesso. No âmbito da Internet esta idéia está diretamente associada a garantir que sistemas de informação possam ser acessados por qualquer indivíduo. Embora a acessibilidade na Internet seja associada principalmente a pessoas com algum tipo de deficiência, nos últimos anos ampliou-se o conceito para além das limitações inerentes aos usuários. Atualmente a acessibilidade engloba as condições tecnológicas no momento de acesso, como por exemplo o uso de computadores com sistemas operacionais defasados, o acesso por meio de telefones celulares ou em ambientes com pouca velocidade de transmissão de dados. Conforme atesta Queiroz, "qualquer tipo de barreira para qualquer pessoa, mesmo sem deficiências ou apenas com limitações temporárias, passou a ser relacionado à acessibilidade. (...) Por exemplo, calçadas esburacadas, perigosas para mulheres grávidas que não podem enxergar os pés, ou um site na internet cujo código não permita o acesso por meio de celulares, passaram a ser inacessíveis. (...) Hoje, na prática, acessibilidade diz respeito à qualidade ou falta de qualidade de vida para todas as pessoas". (QUEIROZ, 2006).

com a apresentação visual dos sites, sem se preocupar com as etapas anteriores. Os profissionais que adotam esta postura pouco diferem dos usuários finais, no que diz respeito ao entendimento dos aspectos técnicos do processo produtivo, deixando para "o pessoal de tecnologia" essas questões — o que sem dúvida tem conseqüências negativas no desenvolvimento dos projetos.

Neste artigo discute-se como planejar sistemas de informação interativos acessíveis e com uma experiência de uso consistente, independente do ambiente no qual serão acessados. Para tanto, parte-se do princípio de **tolerância a falhas** e especialmente da metodologia de **melhorias progressivas**. Como recorte, são analisadas diferentes etapas do projeto de *websites*, de acordo com a metodologia mencionada. Ao final, destaca-se a necessidade dos profissionais de *user experience* se aproximarem das questões técnicas relativas à implementação de sistemas de interativos.

Múltiplos contextos: um olhar abrangente

Foi-se o tempo em que a maior preocupação dos designers envolvidos na criação de sites era a perfeita codificação dos *layouts* do Photoshop em páginas HTML, fiéis ao projeto original, nos diferentes navegadores existentes no mercado. Embora essa preocupação ainda persista, há questões mais complexas para as quais os designers de interação devem estar atentos. Uma delas é a pluralidade de meios de acesso existentes atualmente. Se antes o problema se resumia a um par de sistemas operacionais e uma meia dúzia de *browsers*, hoje há uma infinidade de dispositivos capazes de acessar a Internet, sem qualquer padronização entre eles.

Tomemos como exemplo os telefones celulares, cada vez mais utilizados como porta de entrada para o ambiente virtual. Os aparelhos existentes no mercado apresentam sistemas operacionais variados, com mecanismos de interação que vão do convencional teclado alfanumérico às telas de toque – acionadas com a ponta dos dedos (caso do iPhone, da Apple) ou através de apontadores similares a canetas (*stylus*). Cada interface demanda soluções diferentes para a realização de ações relativamente simples como preencher um formulário. Enquanto nos telefones com telas de toque o usuário literalmente pressiona uma área da tela, nos modelos mais antigos ele aperta botões no celular para alternar entre as áreas clicáveis do site, até selecionar o campo desejado do formulário. A mecânica de interação muda completamente. Sites projetados de maneira a depender exclusivamente que o ponteiro do mouse seja posicionado sobre determinado elemento da página (*mouseover*), não irão funcionar na maioria dos celulares, já que estes normalmente não trabalham com ponteiros como os PCs.

Além disso, mesmo para os celulares mais modernos, ainda há limitações da quantidade de dados transmitidos por segundo, e da capacidade de processamento dos aparelhos. Até o momento os celulares são muito inferiores aos computadores *desktop* nesse aspecto. Não fosse isso problema bastante, muitos telefones são incompatíveis com alguns componentes e linguagens de programação utilizados nos sites, como Flash, Javascript e AJAX. Dependendo de como o site for implementado, mesmo que o celular ignore os *scripts* e componentes utilizados esses dados serão transmitidos da mesma forma, desperdiçando tempo e consumindo banda inutilmente. Percebe-se portanto que um site originalmente planejado para ser visto em um PC, por mais que funcione em um celular, terá uma performance distinta neste ambiente

(PETINATTI, 2007). Pelo exposto, parece claro que a adequação de sites para celulares é algo bem mais complexo do que simplesmente pensar em termos de dimensões da tela.

É importante destacar que esta questão vai além dos PCs e celulares, considerando que os meios de acesso aos sistemas de informação não se limitam a esses aparelhos. Alguns consoles de *games* já permitem navegar na Internet, assim como a TV digital cedo ou tarde será mais um meio de acesso. O iPad da Apple, lançado em 2010, também tem limitações e características próprias, as quais deverão ser levadas em consideração para que os sistemas sejam acessíveis nesta plataforma. Uma vez que a tendência é que surjam outros meios de acesso a partir do amadurecimento da computação pervasiva (PINHEIRO e SPITZ, 2007), é praticamente impossível prever os contextos de uso e as características das plataformas que virão por aí.

Projetando em camadas

Parece clara a complexidade de projetar sistemas de informação para tamanha diversidade de ambientes. Apesar disso, há maneiras de garantir que mesmo em meios de acesso com poucos recursos se tenha uma experiência satisfatória. Embora este assunto não seja novidade, ainda é pouco discutido entre os designers de interação.

O ponto principal a se ter em mente é que os sistemas devem ser acessíveis em todos os ambientes, mesmo naqueles que não suportam determinadas funcionalidades. No passado era comum que os sites fossem planejados para o melhor ambiente possível: durante o projeto, partia-se da situação ideal e testava-se o site em ambientes menos capazes, fazendo as alterações necessárias para que nesses casos o usuário ainda tivesse uma oferta de conteúdo satisfatória. Isto é chamado de **degradação graciosa** (*graceful degradation*), método derivado da **tolerância a falhas** (*fault-tolerance*), isto é, os sistemas devem funcionar mesmo na ocorrência de uma falha em um de seus componentes. Na prática, no entanto, muitas vezes isso não é feito, seja por pressa, por falta de recursos, ou por incompetência. É comum webdesigners optarem por ignorar os usuários que não usam navegadores adequados ao site. É o caso de projetos feitos totalmente em Flash, ou que utilizam Javascript/AJAX para exibir alguns elementos da página, e não apresentam alternativas em HTML. Nesses casos, os usuários que usam *browsers* com limitações pagam o preço por seu "atraso", deixando de ver determinados conteúdos ou mesmo perdendo funções, projetadas originalmente para ambientes mais avançados. Para muitos supostos profissionais de mídia interativa, o problema é do usuário que não usa o navegador adequado, ou ainda, o problema é do responsável pela implementação do site, que tem a tarefa praticamente impossível de fazer um projeto mal elaborado adequar-se a qualquer tipo de navegador.

Ora, projetar a experiência do usuário não é tarefa de quem implementa o site, e sim do designer de interação. É seu papel planejar a melhor experiência possível, sabendo que nem sempre os navegadores utilizados serão os mais modernos. Isto é um dado do projeto. É preciso pensar alternativas que permitam que o sistema seja utilizado em qualquer ambiente. Isso não é uma questão de implementação, e sim de projeto.

Há uma metodologia alternativa à idéia de degradação graciosa, segundo a qual deve-se projetar os sistemas de maneira que eles tenham **melhorias progressivas** (*progressive enhancements*). A diferença é sutil, mas tem um impacto grande especialmente no que diz

respeito ao projeto do sistema. Com a degradação graciosa projeta-se a situação ideal e ao longo do desenvolvimento o projeto é adaptado às diferentes situações de acesso. Com as melhorias progressivas, parte-se do nível mais básico e novas "camadas" são adicionadas ao sistema ao longo do tempo. Quanto mais avançado o ambiente usado para acessar o sistema, mais "camadas" ele consegue interpretar. Como o projeto parte do ponto mais básico, mesmo os usuários em condições limitadas podem acessar o conteúdo. Enquanto na degradação graciosa o foco é na apresentação, com as melhorias progressivas o foco é no conteúdo.

Uma diferença importante entre os dois métodos de desenvolvimento é que enquanto na degradação graciosa mesmo as informações que não são interpretadas corretamente por navegadores limitados são transmitidas, nos sistemas desenvolvidos através de melhorias progressivas apenas as informações compatíveis com o navegador utilizado são enviadas, economizando tempo e recursos de ambas as partes (cliente e servidor). Isso é extremamente interessante ao levarmos em conta os celulares, que apresentam limitações de transferência de dados e de adequação a tecnologias mais avançadas, quando comparados aos PCs.

Na figura 1 estão representadas os componentes ou camadas dos sistemas de informação, no contexto de projeto de sites², na perspectiva das melhorias progressivas. Cada "camada" será discutida em detalhes a seguir.

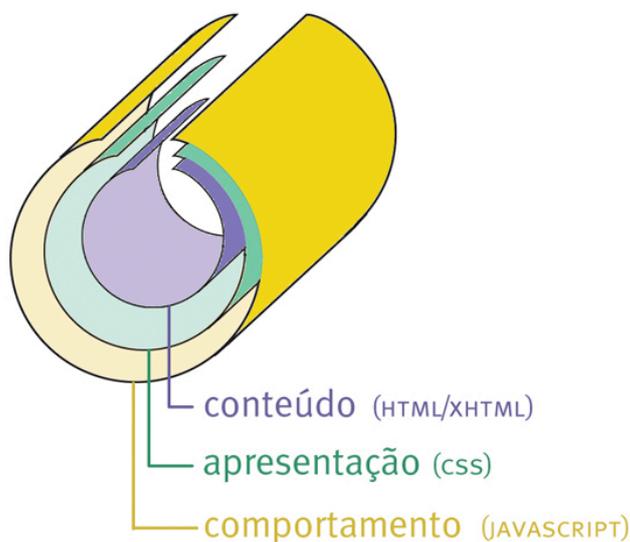


Figura 1: componentes dos sistemas de informação no contexto de projeto de sites

A primeira camada: conteúdo

A camada central é o conteúdo. Conceber bem a estrutura de um site, relativa ao conteúdo, estabelecer hierarquias entre os elementos do texto utilizando HTML semântico, é

² No âmbito da Engenharia de Software, usa-se terminologia semelhante para referir-se a divisão estrutural de componentes dos sistemas, isto é, a divisão em camadas com funções distintas no sistema; nesse caso a quantidade de camadas varia de acordo com o modelo de arquitetura de software utilizado. Neste artigo discute-se fundamentalmente questões relativas ao projeto de sites, com uma ótica voltada para os profissionais de design de interação, buscando relacionar os componentes do sistema com etapas do processo de design e implementação *client-side* de sites.

uma tarefa essencial que muitas vezes não recebe a devida atenção. Não se trata de uma etapa meramente técnica da implementação do site, mas da definição da importância dos seus elementos. No campo do design gráfico esta é uma prática antiga: analisar o texto, extraíndo dele as categorias que ajudam a definir a hierarquia de cada parte em relação ao todo, separando elementos como títulos, subtítulos, legendas, notas de rodapé. Cada elemento tem uma função específica no conjunto da obra, e merece um tratamento particular. Isto é fundamental para a etapa seguinte, a de estabelecer critérios para a apresentação do conteúdo. Um site com uma hierarquia clara entre seus elementos será mais facilmente compreendido, não só pelo usuário, mas pelos diferentes ambientes que vão interpretar o site — navegadores, leitores de tela, celulares etc.

Costuma-se discutir o que deve ser considerado conteúdo, especialmente entre os adeptos de Flash como principal ferramenta para a construção de sites. Há quem argumente que a forma é também conteúdo, não cabendo uma separação estrita entre texto, imagem, movimento: tudo seria conteúdo. Embora essa argumentação faça sentido para alguns projetos, na maioria dos casos é preciso distinguir forma e conteúdo. Flash é uma ferramenta poderosa que possibilita soluções interessantes na criação de sites, mas trata-se de uma maneira de trabalhar a apresentação de conteúdos, e não de um conteúdo em si. É importante separar bem o que é apresentação e o que é conteúdo, especialmente se levarmos em consideração que existem ambientes que não suportam Flash, como os leitores de tela utilizados por pessoas cegas, e grande parte dos celulares. Se ao construir o site não houver uma preocupação específica em criar mecanismos que garantam alternativas de fornecer o conteúdo para esses ambientes, o site será "invisível". O mesmo vale para os robôs dos mecanismos de busca, que indexam os conteúdos dos sites baseado no texto presente no código HTML das páginas. Há quem acredite que basta fornecer informações nas *meta tags* inseridas no HTML, como maneira de descrever aquele site. Mas isso não é suficiente, uma vez que os robôs de busca terão apenas aquele conteúdo para indexar, ignorando o que for colocado dentro do Flash. Isso limita a possibilidade de fornecer dados sobre o site.

Se de qualquer forma for necessário utilizar Flash, é importante que se tenha o cuidado de gerar alternativas para tornar o site acessível, para que o conteúdo possa ser compreendido em outros ambientes, caso contrário o site será inútil para uma parcela do público. Thomas Vander Wal (2006) alerta que um site "precisa degradar bem. Precisa ser acessível. Precisa ser usável. Senão, será uma bobagem; bacana, mas inútil para algumas ou muitas pessoas".

A segunda camada: apresentação

A segunda camada é a de apresentação. É a etapa que costuma receber mais atenção dos designers, embora não seja necessariamente a mais importante. É neste momento que a estrutura semântica do conteúdo, definida anteriormente, toma forma nos navegadores, "traduzida" através de estilos criados para este fim. É importante destacar que a apresentação não é apenas visual, uma vez que leitores de tela utilizados por cegos também são capazes de interpretar o conteúdo. A apresentação, nestes casos, será sonora, e não visual. Por isso é fundamental definir bem a estrutura semântica do conteúdo na etapa anterior, uma vez que nem sempre a apresentação do site será aquela definida na segunda etapa. De qualquer forma,

mesmo que utilize soluções de apresentação tão distintas como uma família tipográfica em negrito na tela ou um tom mais grave no sintetizador de voz, o sistema deverá ser preparado para manter as relações hierárquicas definidas a priori, independente do meio de acesso utilizado. Como destacado anteriormente, se projetado em termos de melhorias progressivas, o sistema estará apto a oferecer uma experiência satisfatória com as mínimas condições de acesso.

Normalmente a camada de apresentação de sites é definida com estilos, através de CSS³. Isso é relativamente conhecido e utilizado há algum tempo. No entanto, muitos designers ainda não atentam para detalhes que podem fazer a diferença na hora de acessar os sistemas em ambientes diversos. É importante trabalhar de maneira a garantir flexibilidade na apresentação das informações, permitindo por exemplo que o tamanho da tipografia possa ser alterado nos navegadores. Da mesma forma, a estrutura ou *grid* da página devem ser passíveis de redimensionamento proporcional, garantindo que o site mantenha sua integridade, sem que elementos da página se sobreponham uns aos outros quando o site for visto ampliado (como, por exemplo, quando acessado em celulares com telas pequenas ou mesmo em navegadores em computadores *desktop*, para pessoas com visão deficiente).

Outro detalhe que muitas vezes passa despercebido é o uso de imagens para títulos e elementos de navegação. Sabe-se que são limitadas as opções de tipografia na Web. Uma vez que a aplicação das famílias tipográficas é feita no computador do usuário, as opções de famílias tipográficas mais seguras são aquelas que normalmente estão presentes na maioria dos computadores, independente do sistema operacional utilizado. Isso garantiria, em tese, que o site fosse visto com a tipografia que foi planejada pelo designer. Para contornar a limitação de famílias tipográficas, é comum o uso de imagens, especialmente para títulos, para menus de navegação e outros elementos pontuais. Isso no entanto tem o problema de adequação para ambientes limitados. É conhecida a necessidade de se colocar um texto no atributo ALT das imagens, para que *browsers* que não capacitados a exibir imagens possam apresentar alguma informação alternativa. O mesmo vale para ambientes que não interpretam CSS, como alguns celulares. Nesses casos, o site deve ser construído de forma a apresentar as mesmas informações que as imagens utilizadas. Novamente, há que se ter o cuidado de não apenas apresentar as informações de forma esteticamente agradável, mas de possibilitar que diferentes plataformas de acesso possam interpretar essas informações corretamente, dentro de suas especificidades.

O que foi dito para imagens também se aplica quando componentes em Flash são utilizados. Normalmente menus feitos em Flash são ignorados em ambientes limitados. Portanto, devem ser criados menus alternativos, que serão exibidos somente nos ambientes incompatíveis com Flash. Caso contrário, o site não funcionará para parte dos usuários.

A terceira camada: comportamento

A terceira camada é a do comportamento. É aí que os mecanismos de interação são mais evidentes, especialmente após a popularização do Javascript para criação de sites. Áreas da página que se descortinam quando clicadas, que mudam o conteúdo sem requisitar acesso ao

³ CSS, ou *Cascading Style Sheets* é uma maneira de adicionar estilos (por exemplo, cores, espaçamento) a documentos web. O CSS normalmente é um documento separado do conteúdo do site. Com isto separa-se a forma (CSS) do conteúdo (HTML ou XHTML).

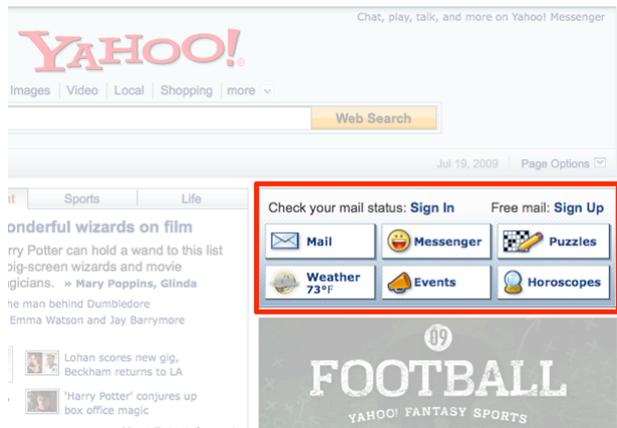
servidor e/ou sem precisar recarregar a página inteira, são cada vez mais comuns, acrescentando um aspecto lúdico e uma experiência mais agradável, na maioria dos casos.

Assim como na camada de apresentação, é importante ter em mente ao projetar o comportamento dos sites que nem todos os ambientes estarão aptos a interpretar corretamente os componentes utilizados. É preciso prover alternativas para suprir essas limitações.

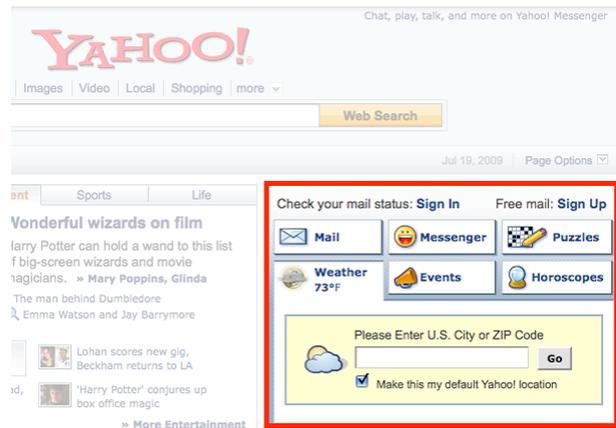
Ressalte-se, mais uma vez, que comportamento não é conteúdo. Embora em alguns casos o comportamento planejado para a situação ideal não seja possível, é necessário garantir meios de acessar o conteúdo. É perfeitamente viável construir sites que excluam componentes que não serão interpretados corretamente sem que isso prejudique o acesso ao conteúdo. Nesse caso, apenas a última camada do processo é alterada.

Dois exemplos disso podem ser vistos no site Yahoo! internacional. No primeiro caso, na página principal alguns elementos contêm informações que só são exibidas quando o ponteiro do mouse está posicionado sobre eles, eliminando a necessidade de navegar para outra página. No segundo caso, é possível navegar entre fotos de uma galeria sem atualizar toda a página. Esses comportamentos dependem de navegadores compatíveis com Javascript. No entanto, os usuários que utilizam *browsers* limitados não são penalizados, podendo acessar o mesmo conteúdo. Nesses casos o site terá um comportamento diferente. O Yahoo! foi planejado de forma a manter inalterada a camada central (o conteúdo), alterando apenas a camada externa (o comportamento), seguindo o método de melhorias progressivas.

No exemplo abaixo (figura 2), ao posicionar o mouse sobre o botão Weather, um box para entrada de dados é exibido, no qual deve-se inserir o nome da cidade sobre a qual se deseja ter informações climáticas. O resultado é apresentado na mesma tela, sem atualizar a página toda. Esse comportamento é construído com Javascript.



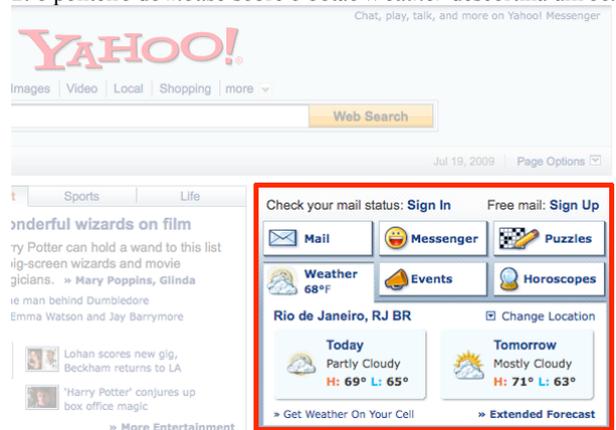
1: a página principal



2: o ponteiro do mouse sobre o botão *Weather* descortina um box



3: o usuário pode solicitar informações sem sair da página



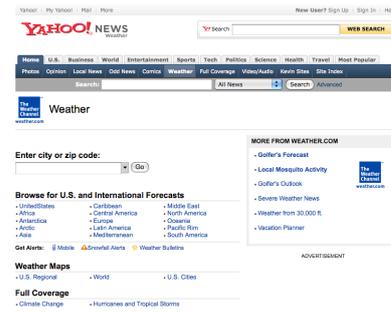
4: a resposta é exibida ainda na página principal

Figura 2: consulta de condições climáticas, utilizando Javascript, no site do Yahoo!

Em navegadores incompatíveis com Javascript, o comportamento é diferente: o posicionamento do ponteiro do mouse sobre o elemento não dispara ação alguma, mas ao clicar no mesmo elemento acessa-se a página de Tempo, na qual pode-se solicitar as informações sobre condições climáticas. Apesar do comportamento do site variar de acordo com o *browser* utilizado, em ambos os casos o conteúdo disponível é o mesmo, mudando apenas a mecânica de interação. A melhoria progressiva permite um acesso mais rápido ao conteúdo, sem impedir que o mesmo seja visível para usuários com navegadores limitados.



1: na página principal, o botão *Weather* funciona como *link*



2: a página de *Weather*, selecionando o continente



3: navegação interna, selecionando o país



4: navegação interna, selecionando a cidade



5: a página com as condições climáticas da cidade selecionada

Figura 3: consulta de informações climáticas, sem Javascript, no site do Yahoo!

Um outro exemplo da aplicação de melhorias progressivas pode ser visto na galeria de fotos da seção de esportes do site Yahoo! internacional. Nos navegadores com Javascript (figura 4), é possível navegar entre as fotos da galeria sem atualizar toda a página. Nessa versão, além dos botões de avançar e retroceder posicionados acima da foto, há ainda a opção de exibir as fotos em *slideshow*, e a possibilidade de abrir uma galeria com todas as fotos (figura 5).



Figura 4: galeria com recursos Javascript, para navegar entre as fotos sem atualizar toda a página

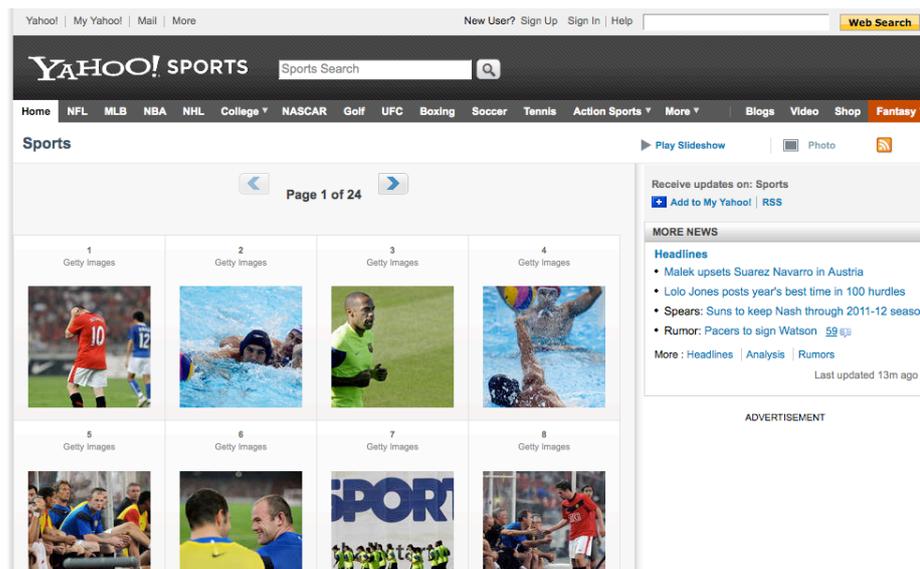
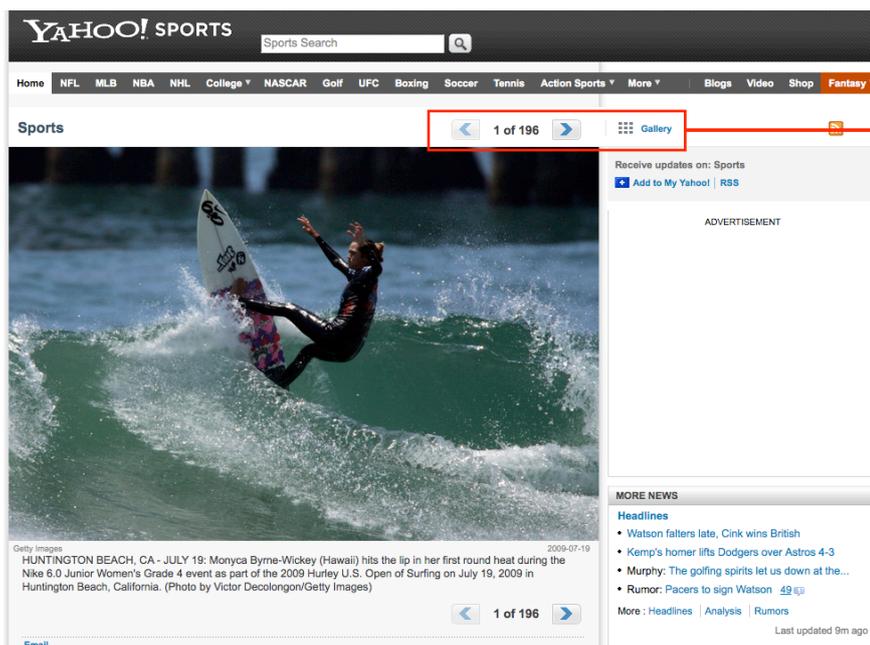


Figura 5: galeria de fotos com recursos Javascript

Nos navegadores incompatíveis com Javascript, somente os botões da navegação geral aparecem, isto é, aqueles que permitem navegar linearmente entre as fotos e apresentar a galeria completa (figura 6). Nesse caso, ocultam-se os elementos que não funcionam nesses navegadores. Ainda assim o sistema permite acessar o mesmo conteúdo da versão mais avançada (as fotos e a galeria completa), mudando apenas o comportamento da interface.



Somente os links de navegação entre fotos e de abrir a galeria completa estão ativos. Cada link abre uma nova página

Figura 6: galeria de fotos sem recursos Javascript. Cada link atualiza toda a página.

Percebe-se que houve uma preocupação em oferecer uma experiência consistente nos diferentes meios de acesso, sem penalizar os usuários com navegadores limitados. Isto exige atenção em todas as etapas, tendo o cuidado em separar o que é conteúdo, apresentação e comportamento, e prever todas as situações possíveis sem privilegiar um determinado ambiente ou sistema.

Considerações finais

Neste artigo não se pretendeu tratar profundamente os métodos, técnicas e alternativas para o desenvolvimento de sistemas de informação acessíveis. Para isto há uma série de artigos e sites especializados no assunto que serão melhores referências. A intenção era, a partir dos exemplos apresentados, chamar atenção para um aspecto pouco discutido sobre este tema e destacar a importância dos designers de interação se envolverem nestas questões.

Como dito anteriormente, há uma situação curiosa no mercado: muitos designers consideram este assunto uma questão técnica, de responsabilidade dos profissionais ligados à implementação *client-side*. No entanto, parece claro não ser este o caso. O projeto de sistemas de informação acessíveis, embora necessite de um *background* técnico, demanda sobretudo uma abordagem conceitual mais abrangente. Para uma prática de design eficiente nesse campo, é fundamental que haja uma aproximação maior com os profissionais responsáveis pela implementação dos sistemas, que têm um olhar mais técnico sobre o assunto e com certeza têm muito a contribuir na busca de soluções adequadas para a multiplicidade de ambientes e situações de uso possíveis. O sucesso de um site não é responsabilidade de indivíduos isolados, mas um trabalho de equipe, que exige necessariamente a colaboração entre os seus membros. Foi-se o tempo do trabalho em cascata, no qual a responsabilidade de cada profissional terminava com a entrega da sua parte para o responsável pela próxima etapa.

Os exemplos apresentados focaram especificamente em sites, acessados em navegadores de computadores *desktop*, por serem esses os casos mais comuns e facilitarem o entendimento do assunto. Mas com o amadurecimento da pervasividade computacional, objetos diversos serão também portas de entrada para o ambiente virtual. Isto torna a questão da acessibilidade de sistemas de informação ainda mais complexa, demandando uma atenção especial dos profissionais ligados ao projeto da experiência do usuário.

Referências

GARRET, Jesse James. **The elements of user experience: user-centered design for the web**. New York: AIGA/New Riders, 2002.

GUSTAFSON, Aaron. Understanding progressive enhancement. **A list apart**, 7 out. 2008. Disponível em: <<http://bit.ly/gustafson01>>. Acesso em: 18 jul. 2009.

_____. Progressive enhancement with CSS. **A list apart**, 21 out. 2008. Disponível em: <<http://bit.ly/gustafson02>>. Acesso em: 18 jul. 2009.

_____. Progressive enhancement with javascript. **A list apart**, 4 nov. 2008. Disponível em: <<http://bit.ly/gustafson03>>. Acesso em: 18 jul. 2009.

KEITH, Jeremy. Best practices. In: **DOM scripting: web design with javascript and the document object model**. Berkeley, California, USA: Friends of ED, set. 2005.

_____. Hijax: progressive enhancement with AJAX. In: **XTech 2006**, Amsterdam, 16-19 mai. 2006. Disponível em: <<http://bit.ly/keith-xtech2006>>. Acesso em: 18 jul. 2009.

NIELSEN, Jakob. Mobile Usability. **Jakob Nielsen's Alertbox**, 20 jul. 2009. Disponível em: <<http://www.useit.com/alertbox/mobile-usability.html>>. Acesso em: 20 jul. 2009.

PETINATTI, Lucas. Challenges of interface design for mobile devices. **Yahoo! User Interface Blog**, 2 out. 2007. Disponível em: <<http://bit.ly/petinatti2007>>. Acesso em: 17 jul. 2009.

PINHEIRO, Mauro. Do design de interface ao design da experiência. **Revista Design em Foco**, v.IV n.2, jul./dez. 2007. Salvador: EDUNEB, 2007, p.9-23.

PINHEIRO, Mauro; SPITZ, Rejane. O design de interação em ambientes de ubiquidade computacional. In: Congresso Internacional de Design da Informação, 3., 8 a 10 out. 2007, Curitiba. **Anais...CD-ROM**. Curitiba, 2007.

QUEIROZ, Marco Antonio de. Acessibilidade Web: tudo tem sua primeira vez. **Bengala Legal**, 2006. Disponível em: <<http://www.bengalalegal.com/capitulomaq.php>>. Acesso em: 20 ago. 2010.

SPELTA, Lêda. Acessibilidade web: 7 mitos e um equívoco. In: Encontro Brasileiro de Arquitetura de Informação (EBAI), 1., 19 a 20 out. 2007, São Paulo. **Anais...São Paulo**, 2007.

SOUZA, Edson Rufino de. Acessibilidade ainda é um sonho distante. **Webinsider**, 25 mai. 2007. Disponível em: <<http://bit.ly/webinsider-souza-2007>>. Acesso em: 17 jul. 2009.

STEARNS, Geoff. SWFObject: javascript flash player detection and embed script. **Deconcept**. S/D. Disponível em: <<http://blog.deconcept.com/swfobject/>>. Acesso em: 18 jul. 2009.

THE YAHOO! USER INTERFACE LIBRARY. **Graded Browser Support**. Disponível em: <<http://bit.ly/yahoo-ui-gbs>>. Acesso em: 18 jul. 2009.

WAL, Thomas Vander. For many AJAX is not degrading, but it must. **Vanderwal.net**, 1 jan. 2006. Disponível em: <<http://bit.ly/vanderwal-2006-jan-01>>. Acesso em: 18 jul. 2009.