

Design Responsivo para Software Livre: diretrizes para o OpenRedu Mobile¹

Raphael Salviano T. da Silva, Yuska Paola Costa Aguiar

Departamento de Ciências Exatas (DCX) - Universidade Federal da Paraíba (UFPB),
Campus IV
Rio Tinto - PB - Brasil

{raphael.salviano,yuska}@dcx.ufpb.br

Abstract. *In this paper we understand phenomena related to the consistency of interfaces based on Responsive Design, contributions to the free software communities, as well as studying bottlenecks in interfaces integration process in OpenRedu development environment. Through a case study, in which the evaluation of interfaces of the web version and mobile application was searched for inconsistencies, based on the evaluation, an interface guidelines document was proposed. The guideline document will be needed to standardize the creation of new screens in OpenRedu, and to guide the integrations carried out through a collaborative environment.*

Resumo. *Neste trabalho vamos entender fenômenos relativos à consistência de interfaces com base no Design Responsivo, em contribuições às comunidades do software livre, como também estudar gargalos no processo de integração de interfaces no ambiente de desenvolvimento do OpenRedu. Por meio de um estudo de caso, no qual foi realizado a avaliação de interfaces da versão web e aplicação mobile em busca de inconsistências, com base na avaliação foi proposto um documento de diretrizes de interface. O documento de diretrizes será necessário para padronizar a criação de novas telas no OpenRedu, e para guiar as integrações realizadas por meio de um ambiente colaborativo.*

1. Introdução

Atualmente pode-se dizer que é latente a mudança vivida pela sociedade em relação à tecnologia computacional. Segundo Shneiderman (2002) a tecnologia computacional considerada apenas o que os computadores podiam fazer pelo usuário, enquanto que agora há uma reflexão no que os usuários podem e pretendem fazer com seus dispositivos. Percebe-se uma mudança significativa no foco, que passa do dispositivo para o usuário (Puppi 2014).

Nesta perspectiva, observa-se que a última década foi marcada pela transição dos tipos de dispositivos de tecnologia computacional. O que antes tinha como produto tecnológico somente aparelhos fixos, como o computador pessoal (PC) e televisores,

¹ Trabalho de Conclusão de Curso apresentado pelo aluno **Raphael Salviano Trajano da Silva** sob a orientação da professora **Yuska Paola Costa Aguiar** como parte dos requisitos para obtenção do grau de Licenciado em Ciência da Computação na UFPB Campus IV.

vem dividindo espaço com a tecnologia parcialmente móvel, como *notebooks* e *ultrabooks*, e totalmente móvel como *smartphones* e *tablets*.

Com a nova realidade dos dispositivos tecnológicos está havendo uma readequação na forma que a informação é exibida ao usuário, ou seja, os sites e aplicações precisam se adaptar ao dispositivo utilizado pelo usuário. Segundo Bosco (2012) a ideia de a informação estar disponível a qualquer hora e em qualquer lugar já foi realizada. Mas existe um outro passo, no qual a informação deve estar disponível e adaptada para qualquer dispositivo. Com cada um deles possuindo formas diferentes de manipulação e proporcionando a experiência de uso adequada.

A diversidade de dispositivos teve como consequência um problema para os desenvolvedores de interfaces: como projetar um site ou aplicação que apresente o conteúdo em diferentes dispositivos sem comprometer a experiência de uso? Como resposta a este problema, foi desenvolvido um conjunto de técnicas para adaptar o conteúdo a diferentes dispositivos, o Design Responsivo.

Segundo Da Silva (2014) Design Responsivo é um princípio de desenvolvimento com foco *web* cujo o objetivo é adaptar o *layout* de páginas a qualquer dispositivo, tela e resolução, com propósito de garantir a boa experiência do usuário, possibilitando assim navegação e leitura confortáveis sem comprometer o acesso ao conteúdo a ser comunicado. Antes do Design Responsivo, era comum a criação de uma ou mais versões *mobile* do mesmo *site* ou aplicação, o que dificultava a manutenção do conteúdo destes.

Segundo da Cruz Pereira (2015) a ausência de diretrizes que definam padrões visuais e de comportamento da interface é responsável pela dificuldade em manter a consistência das novas telas em relação ao que era esperado inicialmente, e isso acontece durante a integração das colaborações realizadas por terceiros.

O desenvolvimento colaborativo em conjunto com o software livre e de código aberto (F/LOSS, do inglês: *Free/Libre and Open-Source Software*) vem ganhando mais espaço a cada dia, sendo crescente o número de projetos que passam a utilizar este tipo de modelo (Murillo 2009). Dessa forma surgem alguns problemas, como o de manter o código organizado e ter arquiteturas bem definidas, assim como o de manter a consistência de novas interfaces que serão desenvolvidas pelos colaboradores. Todo esse problema ocorre, por não existir um processo de integração definido dentro dos projetos de software livre (Reis 2003).

A maioria dos colaboradores de projetos de software livre lida frequentemente com dificuldades com a documentação disponível sobre o projeto. Este aspecto se agrava quando os colaboradores são novatos no projeto e necessitam aprender os aspectos técnicos e sociais do projeto para poder contribuir. De modo mais específico, de acordo com o trabalho de Redmiles (2014), os problemas encontrados estão mais relacionados à ausência de documentação, como: (i) falta de padrão na estrutura do projeto; (ii) falta de informações sobre as configurações do ambiente de trabalho; e os (iii) padrões de design; ocorrendo prioritariamente nesta sequência.

Segundo Cybis *et al.* (2010) a usabilidade é a qualidade que caracteriza o uso de um sistema interativo. Ela se refere à relação que se estabelece entre usuário, tarefa, interface, equipamento e demais aspectos do ambiente no qual o usuário utiliza o

sistema. Neste contexto, é necessário que aplicações voltadas para o ensino, possam garantir uma boa experiência de uso. Entretanto, aplicações como o ambiente virtual de aprendizagem OpenRedu devem manter a consistência de interface entre suas versões web e mobile. O OpenRedu é um *software* livre, desenvolvido de forma colaborativa o que dificulta em manter uma compatibilidade de interfaces entre versões, uma vez que não há um direcionamento ou um conjunto de diretrizes para poder guiar a comunidade a elaboração de uma interface homogênea. Nesse contexto, este trabalho tem como principal objetivo sugerir um guia de diretrizes para contribuir com a manutenção da consistência de interfaces - com base no Design Responsivo, que serão desenvolvidas para a comunidade do OpenRedu.

O artigo está organizado em três seções, incluindo esta. Na seção 2 apresenta-se a fundamentação teórica. Na seção 3 descreve-se a metodologia adotada na execução da criação do guia.

2. Fundamentação Teórica

Neste capítulo apresentamos o problema de manutenção da consistência de interface em produtos de código aberto, desenvolvidos de forma colaborativa considerando as diretrizes do Design Responsivo. A revisão da literatura tem como objetivos compreender o contexto no qual está inserido o *OpenRedu* e explorar conceitos de usabilidade, como análise heurística e consistência de interfaces, dentro das comunidades de *software* livre.

2.1. OpenRedu

O OpenRedu é uma plataforma de código aberto para ensino com tecnologia, que permite criar, compartilhar e discutir conteúdo das mais diversas formas, estimulando a aprendizagem, colaboração e diversão (Redu 2016). A plataforma é desenvolvida pelo Centro de Informática da Universidade Federal de Pernambuco (CIn-UFPE).

Nesta plataforma os problemas de usabilidade e inconsistência de interfaces, de acordo com as diretrizes do Design Responsivo, são realidade, uma vez que não existe um processo de integração nem diretrizes definidas para o desenvolvimento de interfaces. Segundo da Cruz Pereira (2015) os desenvolvedores podem tomar decisões erradas. Tais decisões faz com que existam inconsistências na interface como um topo, principalmente em pontos específicos relacionados ao OpenRedu mobile que não estão contemplados em diretrizes mais gerais.

2.2. Princípios de Interface Homem-Computador

O desenvolvimento de um sistema com usabilidade depende da análise cuidadosa dos diversos componentes de seu contexto de uso e da participação ativa do usuário em seu projeto de interface. Porém, existe uma configuração base a partir da qual uma interface pode favorecer o estabelecimento da usabilidade na relação usuário - sistema, e essa configuração se faz por meio de critérios, princípios e heurísticas de usabilidade propostos na literatura e que devem ser considerados no projeto de interfaces. Para cada tipo de configuração os especialistas são orientados a avaliar se os componentes de interface estão de acordo com uma série de princípios, onde cada princípio se assemelham às recomendações de design de alto nível.

Dentre as proposições têm-se as “Heurísticas de Usabilidade” desenvolvidas por Nielsen (1994), onde o conjunto original de princípios surgiu a partir da análise de 249 problemas de usabilidade. Segundo Preece (2005), dez dessas heurísticas merecem destaque por ter foco em usabilidade, são elas: (i) visibilidade de *status* do sistema; (ii) compatibilidade do sistema com o mundo real; (iii) controle do usuário e liberdade; (iv) consistência e padrões; (v) ajudar os usuários a reconhecer, diagnosticar e corrigir erros; (vi) prevenção de erros; (vii) reconhecer, em vez de lembrar; (viii) flexibilidade e eficiência no uso; (ix) estética e *design* minimalista; e (x) ajuda e documentação.

Um outro princípio de usabilidade que foi desenvolvido por Ben Shneiderman (2004), denominado de “Regras de Ouro” propõe oito regras para o projeto e avaliação de interfaces. São eles: (i) perseguir a consistência; (ii) fornecer atalhos; (iii) fornecer feedback informativo; (iv) marcar o final dos diálogos; (v) fornecer prevenção e manipulação simples de erros; (vi) permitir o cancelamento das ações; (vii) fornecer controle e iniciativa ao usuário; (viii) reduzir a carga de memória de trabalho.

Segundo Cybis *et al.* (2010) a norma ISO 9241:10 (1998) propõe em sua parte 10, “Princípios de Diálogo”, que são sete princípios ergonômicos para o projeto e avaliação de Interfaces Humano-Computador (IHC) para aplicações de escritório. São eles: (i) adaptação à tarefa; (ii) autodescrição (*feedback*); (iii) controle ao usuário; (iv) conformidade às expectativas do usuário; (v) tolerância aos erros; (vi) facilidade de individualização; e (vii) facilidade de aprendizagem.

Além destes princípios citados, o último conhecido como “Critérios ergonômicos” que foi desenvolvido por Scapin & Bastien (1993). Eles propuseram um conjunto de oito critérios ergonômicos principais que se subdividem em 18 subcritérios e critérios elementares. O objetivo é de minimizar a ambiguidade na identificação e classificação das qualidades e problemas ergonômicos do software interativo.

Porém a especificidade de algum produto torna necessário a escolha de um outro subconjunto de heurísticas ou elaboração de novas, ou seja, ao escolher um produto com um foco em uma determinada área, alguns critérios das heurísticas podem não serem aplicados por não abranger pontos específicos da área em que o produto está determinado. É possível adaptar os princípios de IHC listados acima, ou mesmo utilizar documentos de requisitos, pesquisas de mercado e recomendações de *design* como base na criação de heurísticas (Sharp *et al.* 2005). Em seguida é reunido um conjunto de heurísticas que se encaixa no contexto do produto e com a finalidade de obter um melhor resultado na análise.

Uma outra solução, é o Ergolist (Cybis 2011), que possui um *checklists* com 18 critérios que permitem avaliar a (i) concisão; (ii) mensagem de erro; (iii) flexibilidade; (iv) legibilidade; (v) significados; (vi) proteção contra erros; (vii) agrupamento de formatos; (viii) experiência do usuário; (ix) presteza; (x) controle do usuário; (xi) correção de erros; (xii) consistência; (xiii) agrupamento por localização; (xiv) densidade de erro; (xv) *feedback*; (xvi) compatibilidade; (xvii) ações explícitas e (xviii) ações mínimas.

2.3. Diretrizes de usabilidade

Os *guidelines*, ou documentos de diretrizes de usabilidade, têm como principal objetivo melhorar a usabilidade na interação humano-computador (Machate 2003). Tais documentos são baseados em teorias, dados empíricos e experiências práticas, que unidos são capazes de guiar desenvolvedores sobre como projetos de interface devem ser concebidos (Cronholm 2009).

Em *software* livre existem alguns projetos que já utilizam esse tipo de documentação. Este é o caso do LMS AMADEUS (Gomes *et al.* 2009), que é um ambiente de ensino a distância e que possui no documento de *guidelines* os princípios adotados na interface do produto, considerando: especificações de cores, modelo de corpo, tipografia, tabelas, listas, entre outros.

No caso do FirefoxOS (Mozilla 2016), que é um sistema operacional para dispositivos móveis que permite o desenvolvimento de aplicações por meio de um padrão visual bem definido. Nessas especificações há seções que tratam sobre a paleta de cores, tipografia utilizada, ícones, entre outros.

Já para o *Android*, um outro sistema operacional móvel de código aberto, existe também uma documentação sobre o estilo de interfaces adotado pela plataforma. Que a partir de sua versão 5 (*Android Lollipop*) ganhou um guia de usabilidade denominado Material Design (Google 2016), que por meio do site oficial, é possível entender quais os objetivos e princípios desse novo design, bem como conhecer os estilos de cor e fontes pré-definidas, assim como os padrões de interação por meio da tela de toque ou *mouse* e componentes disponíveis na plataforma.

Tabela 1. Quadro comparativo entre *guidelines*

	Tipografia	Paleta de Cores	Widgets	Ícones	Adaptável
FirefoxOS	Sim	Sim	Poucos	Sim	Não
Material Design	Sim	Sim	Vários	Sim	Sim
LMS Amadeus	Sim	Sim	Poucos	Sim	Não

Realizando uma análise inicial dos *guidelines* de cada ferramenta, ao comparar entre elas, é possível perceber que existe uma significativa diferença em relação a quantidade de *widgets* disponíveis e também a seus componentes serem adaptáveis. Em adicional, ao observar aplicativos que fazem uso de tais guias, apenas o *Material Design* e LMS Amadeus levam vantagem em relação ao *FirefoxOS*, uma vez que na loja de aplicativos do *Android* é vasta a quantidade de aplicativos que utilizam seu guia de diretrizes – o *Material Design*, e o LMS Amadeus por ser um guia de diretrizes específico para ambientes virtuais de aprendizagem.

2.4. Colaboração em projetos de Software Livre

Segundo da Cruz Pereira (2015) em software livre a forma de colaboração corre de forma descentralizada, ocorrendo por meio de comunidades, e devido a isso existe uma grande dificuldade no controle do que é feito. Porém, existem formas de suavizar tal problema, podendo ser utilizado sistemas de controle de versão, como o Git².

O GitHub (2016) é um serviço que permite armazenar repositórios de projetos utilizando controle de versão Git. Também é possível obter uma grande integração entre os membros das comunidades, pois é disponibilizado fóruns, *wiki* e um gerenciamento de falhas. O GitHub ainda disponibiliza a criação de uma cópia paralela a um projeto original, que pode ser administrada pelo próprio usuário, assim as melhorias são realizadas sem necessariamente ter acesso ao repositório original.

O GitHub sozinho não assegura que todos os problemas de *software* livre sejam resolvidos, especialmente os referentes a usabilidade, que normalmente dependem de padrões bem documentados e também da análise das contribuições por especialistas. Logo, a adoção de tais padrões de diretrizes e práticas é uma decisão de projeto, cabendo aos administradores do *software* definir ou não a forma como os colaboradores devem realizar contribuições (da Cruz Pereira 2015).

3. Metodologia

O desenvolvimento do presente trabalho ocorreu em várias etapas. A primeira etapa foi a de realizar uma revisão bibliográfica sobre os tópicos trabalhados, em seguida foi realizado um estudo das interfaces do OpenRedu em suas versões *web* e *mobile*. A partir disso foi elaborado um documento de diretrizes, para detalhamento dos padrões de interface a serem seguidos pelo *software*.

3.1. Objetivos

Cada um dos métodos descritos nas seções seguintes foi importante para atingir os objetivos dessa etapa.

- Identificar, na literatura, processos/métodos com base em Design Responsivo que possam ser utilizados de forma a minimizar as inconsistências identificadas entre o *OpenRedu Web* e *Mobile*;
- Diagnosticar inconsistências de interface, de acordo com diretrizes definidas pelo Design Responsivo;
- Conceber um documento de diretrizes para o *OpenRedu mobile* de modo a permitir uma análise heurística, com foco na revisão de *guidelines* e inspeção de inconsistência.

3.1. Revisão bibliográfica

A revisão bibliográfica foi realizada em três etapas: a partir de busca manual, utilizando palavras-chave que tenham relação com o tema, em páginas que agrupam publicações acadêmicas, como o Google Acadêmico; por meio dos anais de conferências de *software* livre; e por meio de livros relacionados a usabilidade, avaliação heurística e *Design Responsivo*.

² Git: <https://git-scm.com/>

Dentro do tema abordado alguns tópicos foram considerados mais relevantes: *design* responsivo, avaliação heurística, diretrizes de usabilidade e *software* livre. Na tabela 2 encontra-se os tópicos e as palavras-chaves relacionadas em inglês.

Tabela 2. Tópicos e palavras-chaves usados na pesquisa bibliográfica

Tópico	Palavras-chaves
Design Responsivo	responsive design;
Avaliação Heurística	heuristic evaluation; heuristic analysis;
Diretrizes de usabilidade	product guidelines; user interface guidelines; usability guidelines;
Software Livre	open source; free software; floss;

Todas as palavras-chave foram pesquisadas no Google Acadêmico. Os tópicos de Design Responsivo, avaliação heurística e diretrizes de usabilidade trouxeram de forma isolada várias opções. Então foi realizada uma restrição adicionando palavras-chave relacionadas ao tópico de *software* livre.

Na revisão bibliográfica foram utilizados dois livros relacionados à Engenharia de Software e Usabilidade, onde o primeiro deles foi ‘*Software Engineering: A practitioner’s Approach*’ de PRESSMAN (2009), e o segundo foi o de ‘*Design de Interação – Além da Interação Homem-Computador*’ de PREECE (2005).

As pesquisas no Google Acadêmico e nos anais de conferências se mostraram bastante úteis, uma vez que foram encontrados trabalhos relacionados de forma prática. Alguns exemplos como a ‘*AcademicMidTrek ’13*’ no qual Rajanen (2013) publicou um artigo que discute práticas de usabilidade que são utilizadas em *software* livre, sendo esse considerado um trabalho relevante para aprofundar conhecimentos no tema.

3.2. Diagnóstico de inconsistência de interfaces

O processo de diagnóstico foi fundamentado através do Ergolist (Cibys 2011), que é uma técnica analítica de inspeção ergonômica através de *checklist*, com o objetivo de avaliar a interface de uma determinada aplicação, *software* ou *site*, baseado em recomendações ergonômicas. Ela fundamenta-se na aplicação dos critérios ergonômicos citados no item 2.2, por meio de questões apresentadas na lista de verificação, apoiado em glossário e informações detalhadas sobre o critério que está sendo avaliado. O avaliador assinala as respostas e o sistema as armazena, no fim do processo, o sistema gera um laudo final sobre as questões que apresentaram conformidade ergonômica, as que não são aplicáveis e as que não apresenta conformidade ergonômica. Todo o processo foi conduzido por um avaliador com experiência na área de IHC, sob supervisão de um professor formado também na área de IHC.

3.3. Concepção do Guia

O guia foi elaborado utilizando as documentações oficiais presentes nas páginas *web* das empresas detentoras dos sistemas operacionais *Android* e *FirefoxOS*, além do guia de recomendações do Ambiente Virtual de Aprendizagem (AVA) LMS AMADEUS (Gomes *et al.* 2009). Cada uma dessas empresas disponibiliza um conjunto de diretrizes para o *design* de suas aplicações, tomando como base o uso adequado de componentes de interface existentes nos respectivos sistemas operacionais, a fim de que o usuário disponha o máximo de usabilidade possível. Contudo, nota-se que a terminologia usada para descrever componentes de interface varia sensivelmente de acordo com o fabricante, o que dificulta pessoas com conhecimento específico em determinada plataforma utilize os componentes corretos para poder criar a interface com o usuário. Além disso, algum elemento de um dado guia de diretrizes pode não existir em outro guia, o que pode levar a desenvolver interfaces inconsistentes ou impossíveis de serem implementadas.

Por causa de tais problemas, uma análise destas documentações supracitadas foi realizada. Para cada elemento abordado em uma documentação, foi procurado um equivalente na documentação das demais plataformas. Após esta etapa de mapeamento dos componentes, foi realizada uma fusão das diretrizes para cada componente, a fim de gerar as diretrizes gerais.

Para facilitar o entendimento do documento de diretrizes gerais, optou-se por expor cada recomendação de acordo com um único formato. Dessa forma, a primeira versão do guia ficou com uma estrutura bem definida e fácil de ser consultada. Tendo em vista que neste ponto do estudo o resultado do diagnóstico já é conhecido, novas diretrizes tiveram que ser elaboradas para serem utilizadas de acordo com cada componente, a fim de enriquecer o documento final. O documento resultante desse processo pode ser contemplado na sessão 4.2.

4. Resultados

Neste capítulo serão apresentados os resultados obtidos na pesquisa, sendo ele o resultado do diagnóstico de interfaces do OpenRedu *Web* e *Mobile* e um guia de diretrizes *web* para o OpenRedu *Mobile*.

4.1. Diagnóstico de inconsistência entre interfaces Web e Mobile

O resultado do diagnóstico de inconsistência de interfaces do OpenRedu, apresentou um leve grau de divergência entre as versões *web* e *mobile* (Figura 1). Na versão *mobile*, o OpenRedu traz um número maior de inconformidades em relação a sua versão *web*, com um total de 09 critérios de 18 possíveis. A maior diferença pode ser notada no critério de Legibilidade (Figura 1-A), onde a versão *web* atingiu uma marca de 8 itens não contemplados, contra 5 itens de sua versão *mobile*.

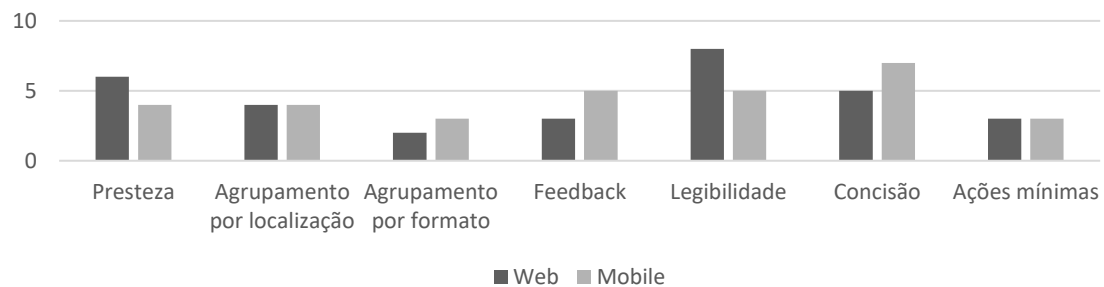


Figura 1-A. Inconformidades: Web X Mobile

Outra divergência que se pode notar, é que no critério de Concisão (Figura 1-A) a versão *mobile* traz um maior índice de itens não contemplados, 7 itens para a versão *mobile* contra 5 da versão *web*. Os critérios de “*agrupamento por localização*” (Figura 1-A), “*ações mínimas*” (Figura 1-A), “*densidade informacional*” (Figura 1-B), “*flexibilidade*” (Figura 1-B) e “*mensagens de erro*” (Figura 1-B), apresentaram média igual, desta forma foi possível observar que ambas as versões trazem falhas nestes 05 critérios.

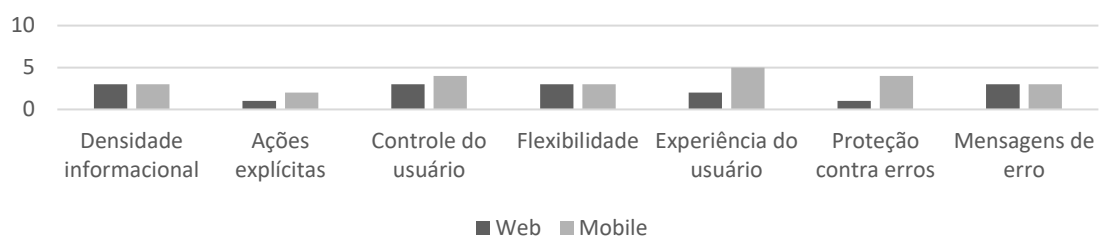


Figura 1-B. Inconformidades: Web X Mobile

Um outro ponto que se pode observar é que a versão *web*, traz um melhor resultado nos critérios de “*ações explícitas*” (Figura 1-B), “*proteção contra erros*” (Figura 1-B) e “*significados*” (Figura 1-C), trazendo uma quantidade de itens não contemplados bem inferior que sua versão *mobile*. Por fim, revela-se que a avaliação completa do diagnostico pode ser conferida por meio de apêndice externo disponível na internet³.

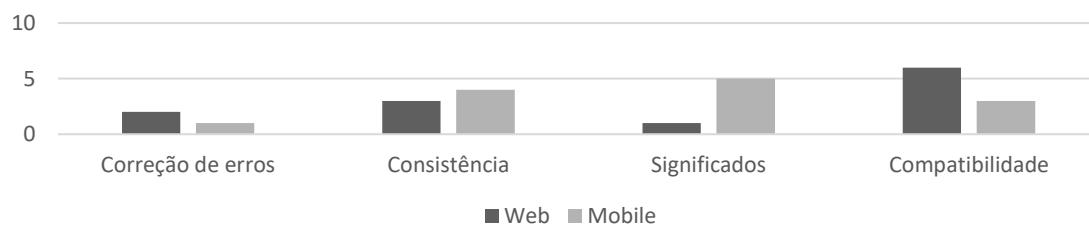


Figura 1-C. Inconformidades: Web X Mobile

4.2. Guia de Diretrizes do OpenRedu Mobile

³ Link de acesso: <https://1drv.ms/w/s!AugTYoo4GHwkhqQSsbhctZTXIYSFbw>

O Guia de Diretrizes do OpenRedu *Mobile*⁴, é um *website* que possui diretrizes de interface com base no Design Responsivo e nos princípios de IHC, com o intuito de padronizar o *layout* e usabilidade das telas da aplicação. O guia ainda oferece junto ao documento, telas prontas codificadas, que estão disponíveis em um repositório no *Bitbucket*⁵ de forma pública e gratuita.

Neste documento de diretrizes são destacados os princípios utilizados na elaboração de interfaces do projeto, o esquema de cores e fontes, e como os componentes devem estar organizados na tela.

⁴ Link de acesso do Guia de Diretrizes: <http://guiaopenredu.businesscatalyst.com/index.html>

⁵ Link de acesso do repositório do Bitbucket:
<https://raphaelsalviano@bitbucket.org/raphaelsalviano/openredu-mobile.git>

4.2.1. Princípios

Considerando o contexto da elaboração de interfaces para *mobile* há dez (10) princípios básicos, definidos por Nielsen (1994) e selecionados por Preece (2005), e que devem ser seguidos na elaboração de uma interface *mobile*:

1. Visibilidade de status do sistema: o sistema deve manter o usuário informado sobre o que está acontecendo, incluindo *feedbacks* em tempo ágil para as ações do usuário. Na prática podemos citar o uso de mensagens de sucesso e erros como peças chaves para seguir este princípio;
2. Compatibilidade do sistema com o mundo real: as telas deverão possuir uma linguagem simples e acessível para o público, por isso é necessário utilizar um vocabulário adequado ao contexto;
3. Controle do usuário e liberdade: é necessário permitir que o usuário possa ir e vir de diferentes telas. Como forma de atingir esse princípio se faz uso do sistema de navegação e de botões para retornar ou avançar para determinadas telas;
4. Consistência a padrões: também é necessário manter ações consistentes dentro da aplicação, de modo que as ações similares sempre ocorram de modo esperado, com um retorno visual coerente;
5. Ajudar os usuários a reconhecer, diagnosticar e corrigir erros: é necessário que a aplicação informe mensagens claras em caso de erro, e que dê possibilidades ao usuário para solucionar-la de forma adequada;
6. Prevenção de erros: na elaboração das telas é necessário analisar a facilidade do usuário cometer erros e entender o porquê;
7. Reconhecer, em vez de lembrar: significa reduzir a carga cognitiva e o uso excessivo de memória do usuário para armazenar informações sobre caminhos complexos ou comandos para acessar determinada página ou funcionalidade. Por isso, é necessário estimular o reconhecimento, de modo que o usuário realize atividades de forma intuitiva;
8. Flexibilidade e eficiência no uso: é necessário que a aplicação disponha de atalhos para que usuários avançados realizem as atividades de forma mais ágil;
9. Estética e design minimalista: nas telas da aplicação é necessário manter apenas informações relevantes, de modo que o usuário não encontre obstáculos por causa da grande quantidade de informações desnecessárias;
10. Ajuda e documentação: em todas as telas é necessário manter acesso a informações de ajuda ou a dicas que ajudem o usuário na execução de atividades dentro da aplicação.

4.2.2. Grids

A partir dos *wireframes* da aplicação foi construído o *grid* que representa a estrutura geral das páginas. Também foram determinados os tamanhos de altura, largura e espaçamento das áreas. Este tipo de informação norteia o desenvolvimento de novas funcionalidades que requisitem o uso de uma ou mais regiões do *layout* padrão. Na Figura 2 é destacada a finalidade de cada área de um *grid* da tela de *login* da aplicação. Além disso o guia traz informações sobre tamanho dos componentes e unidades de medidas a serem utilizados para deixar todo o *layout* responsivo.

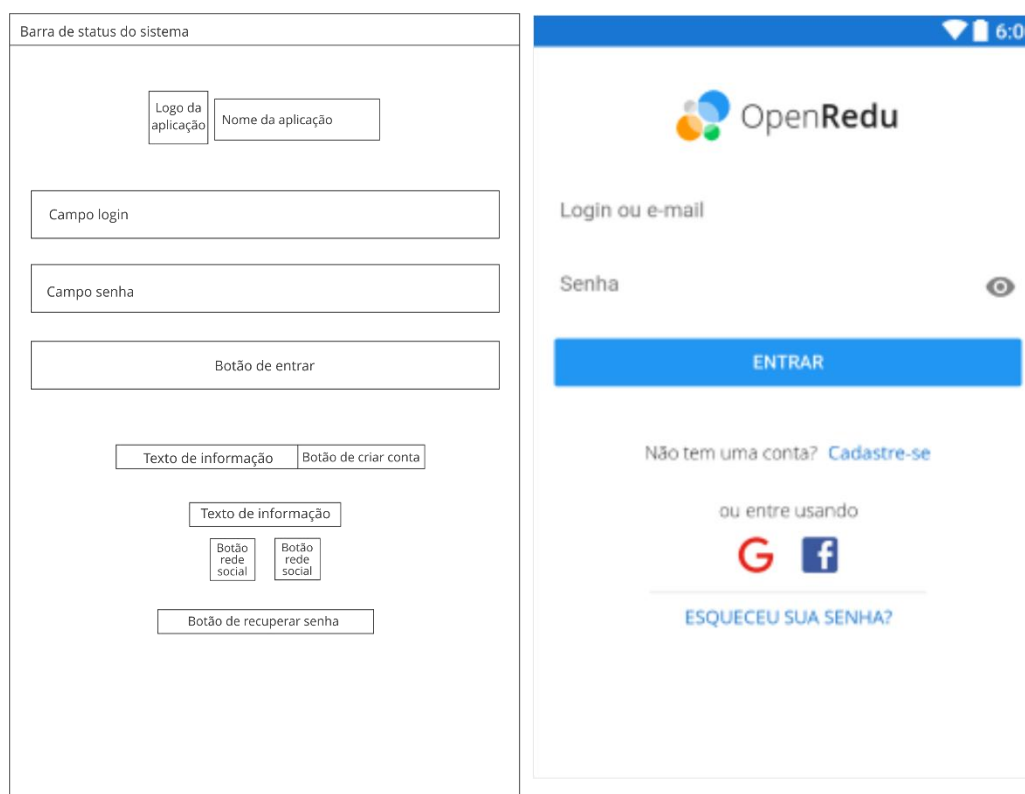


Figura 2. Área de *grid* da tela de Login à esquerda e Tela de Login pronta à direita

4.2.3. Cores

A paleta de cores foi definida de forma a manter a identidade visual do projeto, mas também para realçar informações relevantes com base no senso comum, como por exemplo as mensagens de sucesso e erro. Seguindo as diretrizes encontradas no Material Design da Google, foram selecionadas 3 cores para a identidade visual da aplicação, ciano como cor primária da aplicação, uma variação escura do ciano para a cor primária escura e uma variação do laranja para a cor de destaque. Na Figura 3 é representada as cores definidas na paleta, onde no círculo contém a cor de fato. Abaixo de cada círculo há o valor hexadecimal e os valores no modo RGB (*red, green and blue*).

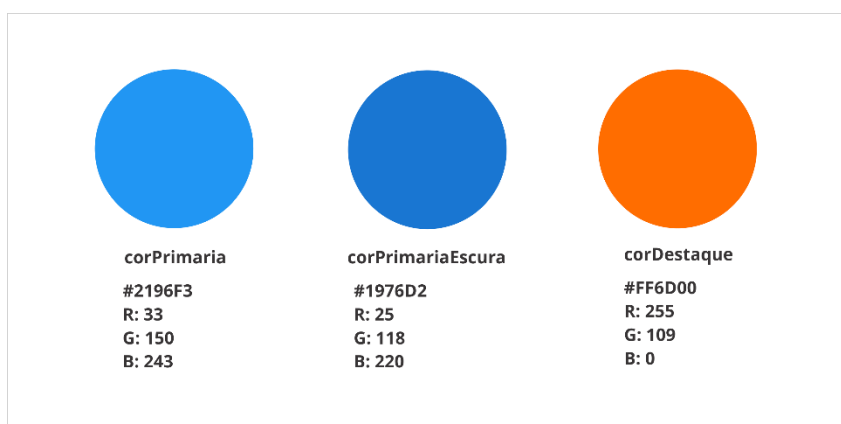


Figura 3. Paleta de cores primária da aplicação

4.2.4. Tipografia

A tipografia é o processo de organizar e definir estilos em fontes de texto, e tem papel importante dentro do *design* da aplicação. É através dela que se torna possível estabelecer hierarquias e adicionar uma ordem estrutural dentro da tela, os artifícios mais conhecidos é a variação de tamanho e tipos de fonte. Para o projeto do OpenRedu Mobile a família de fontes escolhida foi a *Open Sans*, que possui uma vasta opção de espessuras. Na folha de estilos (Figura 4) as fontes estão personalizadas com tamanhos na unidade *sp*, que é padrão no sistema operacional Android.

Display 4	Light 112sp
Display 3	Regular 56sp
Display 2	Regular 45sp
Display 1	Regular 34sp
Cabeçalho	Regular 24sp
Título	Semibold 20sp
Sub-título	Regular 16sp (Device), Regular 15sp (Desktop)
Corpo 2	Regular 14sp (Device), Regular 13sp (Desktop)
Corpo 1	Regular 14sp (Device), Regular 13sp (Desktop)
Rúbrica	Regular 12sp
Botão	SEMIBOLD (ALL CAPS) 14sp

Figura 4. Estilos de fontes

5. Conclusão

Em projetos de *software* livre os colaboradores têm dificuldades de encontrar uma forma de começar a contribuir, seja pela falta de entendimento das necessidades do projeto, quanto não haver documentação suficiente para tal. Também é difícil encontrar

especificações do *design* usado no projeto, por isso quando colaboradores decidem contribuir, não o fazem mantendo as interfaces consistentes com o todo.

Com o uso de técnicas de inspeção ergonômicas para medir o índice de usabilidade de um certo *software*, é essencial para o diagnóstico de inconsistências entre interfaces. Dessa forma foi realizada uma avaliação da interface do OpenRedu em suas versões *web* e *mobile*, afim de verificar se havia inconsistência entre elas. Durante o processo de diagnóstico foi encontrado inconsistências entre as interfaces, sendo isso mais perceptível na versão *mobile* do OpenRedu.

Por fim, com o diagnóstico concluído foi necessário desenvolver um guia com diretrizes para guiar os colaboradores do OpenRedu a projetar interfaces mais consistentes e com uma usabilidade melhorada e adaptativa para diversos dispositivos. Pode-se entender que, com um processo bem definido e uma documentação de interfaces disponível para os colaboradores, é possível manter projetos de *software* livre com interfaces consistentes em novas contribuições a comunidade, assim o projeto pode se tornar mais atrativo a novos colaboradores, promovendo a comunidade e permitindo uma melhor evolução, tanto de códigos quanto de interfaces.

5.1. Limitações

A principal limitação deste trabalho é que devido ao tempo não foi possível realizar uma avaliação do Guia por parte da comunidade do OpenRedu.

5.2. Trabalhos Futuros

Além de melhorar pontos definidos nas limitações, podemos citar como trabalho futuro a avaliação do atual guia de diretrizes por parte da comunidade de colaboradores do OpenRedu. Também podemos sugerir uma adaptação do guia para abranger também a versão *web* do ambiente, unificando assim a experiência entre plataformas.

Referências

- Bastien, J. C., & Scapin, D. L. (1993). Ergonomic criteria for the evaluation of human-computer interfaces (Doctoral dissertation, Inria).
- Bosco, T. Responsive Web. Wide, n.88, jan/fev 2012, p.24-27.
- Cronholm, S. (2009, November). The usability of usability guidelines: a proposal for meta-guidelines. In Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7 (pp. 233-240). ACM.
- CYBIS, W. ErgoList. 2011. Disponível em: <<http://www.labiutil.inf.ufsc.br/ergolist>>. Acesso em: 29 jul. 2016.
- CYBIS, W., BETIOL, H., & FAUST, R. Ergonomia e Usabilidade: Conhecimentos, Métodos e Aplicações.[SI: sn], 2010. Citado, 10, 30.
- da Cruz Pereira, P. H. CONSISTÊNCIA DE INTERFACES EM SOFTWARE LIVRE: PROCESSO E GUIA DE RECOMENDAÇÕES PARA O LMS AMADEUS.

- Da Silva, A. D. A. P. (2014) Design Responsivo: técnicas, frameworks e ferramentas.
- FREE SOFTWARE FOUNDATION. O que é software livre? Free Software Foundation. Disponível em: <<http://www.gnu.org/philosophy/free-sw.html>>. Acesso em: 29 jul. 2016.
- GITHUB. About. GitHub. Disponível em: <<http://github.com/about>>. Acesso em: 14 jul. 2016.
- Gomes, A. S., Carvalho, R. S., Melo Filho, I. J., Rolim, A. L. S., Monteiro, B. S., & Oliveira, G. R. S. (2009). Amadeus: Novo modelo de sistema de gestão de aprendizagem. *Revista Brasileira de Aprendizagem Aberta e a Distância*, 8, 1-15.
- GOOGLE. Introduction. Material design. Disponível em: <<http://www.google.com/design/spec/material-design/introduction.html>>. Acesso em: 13 jul. 2016.
- ISO 9241. Ergonomic requirements for office work with visual display terminals (VDTs). Part 10: Dialogue principles.
- Machate, M. B. J. (2003). Creative Design of Interactive Products and Use of Usability Guidelines-a Contradiction?. *Human-Computer Interaction: Theory and Practice*, 1, 43.
- MOZILLA. Firefox OS Guidelines. Mozilla Style Guide. ISSN <https://www.mozilla.org/en-US/styleguide/products/firefox-os/>. Acesso em: 20 jul. 2016.
- Murillo, L. F. R. (2009). Tecnologia, política e cultura na comunidade brasileira de software livre e de código aberto (Doctoral dissertation, UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL).
- Nielsen, J., & Mack, R. L. (Eds.). (1994). *Usability inspection methods* (pp. 1-414). New York, NY: Wiley & Sons.
- Preece, J., Rogers, Y., & Sharp, H. (2005). *Design de interação*. Bookman.
- Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.
- Puppi, M. B. (2014). Diretrizes para design de interface de aplicativos em smartphones para alemão como língua estrangeira: um estudo sobre mobile learning.
- Rajanen, M., & Iivari, N. (2013, October). Open Source and Human Computer Interaction Philosophies in Open Source Projects: Incompatible or Co-Existent?. In *Proceedings of International Conference on Making Sense of Converging Media* (p. 67). ACM.
- REDMILES, D. et al. The Hard Life of Open Source Software Project Newcomers. CHASE 2014 Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering, Hyderabad, India, 2-3 jun. 2014. 72-78.
- Reis, C. R., & de Mattos Fortes, R. P. (2003). Caracterização de um Processo de Software para Projetos de Software Livre (Doctoral dissertation, PhD thesis, University of Sao Paulo, Brazil).

- SHARP, H., Rogers, Y., & Preece, J. (2005). Design de Interação: além da interação homem-computador. Artmed.
- Shneiderman, B. & Plaisant, C. (2004) Design the user interface: strategies for effective human-computer interaction. 4. Ed. Addison-Wesley Publishing Company.
- Shneiderman, B. Leonardo's Laptop: human needs and the new computing technologies. Cambridge, MA: The MIT Press, 2002.