

A PROGRAMAÇÃO COMO ELEMENTO POTENCIALIZADOR DO PROCESSO DE DESIGN DE INTERFACE

*Rebeca Medeiros de Andrade¹
Fábio Gonçalves Teixeira²*

Resumo: O Design de Interface é fundamental no desenvolvimento de soluções amigáveis, intuitivas e que visem a melhor experiência possível do usuário. Considerando as limitações de pessoas com baixa visão para utilizar computadores, que dependem de recursos que ajustem contrastes, tamanho e corpo de fonte, por exemplo, e também que alguns dos acessórios podem não ser adaptáveis, a proposta deste trabalho é apresentar a Programação como uma ferramenta que potencializa o desenvolvimento de projetos gráficos em ambiente digital. Para tanto, foram levantados princípios de usabilidade focados no usuário com baixa visão, por meio de levantamento bibliográfico, análise de três aplicativos calculadora, programação e teste de um aplicativo-piloto de calculadora básica. Como conclusão, propõe-se que a Programação e o Design de Interface representam uma conexão entre áreas de conhecimento com real potencial na valorização da diversidade em ambiente virtual.

Palavras-chave: Design de Interface, Tecnologia Assistiva, Deficiência Visual, Programação Orientada ao Objeto.

Abstract: Interface Design is fundamental in the development of graphical solutions that are friendly, intuitive and that enhance the user experience. Considering the limitations of people with low vision to use computers, which depend on features that adjust contrast, font size and body, for example, and also that some of the accessories may not be adaptable, the aim of the study work is the presentation of Programming as a tool that potentializes the development of graphic projects in a digital environment. For this purpose, user-focused usability principles with low vision were collected through a literature review, analysis of three applications calculator, programming and testing of a basic digital calculator pilot application. The study proposes that Programming and Interface Design represent a connection between areas of knowledge with real potential in promotion and appreciation of diversity in digital environments.

Keywords: Interface Design, Assistive Technology, Visual Impairment, Object-oriented Programming.

¹ Mestranda, PGDesign – UFRGS. rebeca.andrade@ufrgs.br.

² Professor Doutor, PGDesign – UFRGS. fabiogt@ufrgs.br.

1 Introdução

Sistemas operacionais de computador possuem acessórios, que são pequenos programas desenvolvidos com a finalidade de executar algumas tarefas básicas do dia-a-dia. Quanto às calculadoras, do ponto de vista dos acessórios de computadores, é possível perceber a existência de certo padrão em relação às atribuições oferecidas por diferentes tipos de calculadoras (considerando a existência de calculadoras básicas, avançadas e também outras que desempenham funções muito específicas e/ou complexas). Esses aplicativos geralmente apresentam uma interface que é, na verdade, uma metáfora de calculadoras físicas, apresentando a mesma disposição de botões e visor, por exemplo.

Tomando como base os três aplicativos analisados e considerando o fato de que eles não oferecem a opção de adaptações na configuração da interface como aumentar janela proporcionalmente (o que inclusive aumenta o tamanho dos botões e o tamanho do visor), ajustar a cor dos botões, alterar contraste entre botões e fundo e aumentar corpo e peso de caracteres, percebe-se que existe uma oportunidade de desenvolvimento de um novo programa que além de executar as funções básicas de uma calculadora, ainda atenda às necessidades de adaptação demandadas por parte de um público específico que tem necessidades também específicas. No contexto brasileiro atual, no que se diz respeito à Tecnologia Assistiva, percebe-se que as adaptações geralmente são produzidas a partir da exploração criativa, fazendo uso de materiais não convencionais e de baixo custo, sempre num esforço de equiparar as oportunidades que são oferecidas a pessoas típicas às oferecidas a pessoas com deficiência (HOHMANN; CASSAPIAN, 2011).

Nesse âmbito, o objetivo geral deste trabalho é, por meio da análise de três aplicativos de calculadora e da Programação Orientada ao Objeto (POO), desenvolver um aplicativo calculadora que atenda às necessidades demandadas por um público com deficiência visual, que tem necessidades específicas em relação à usabilidade de interfaces em ambiente virtual. Em relação aos requisitos básicos funcionais, o aplicativo deveria ser compatível com plataforma Microsoft® Windows e ser capaz de executar cálculos próprios de uma calculadora básica: somar, subtrair, multiplicar, dividir, calcular porcentagem e extrair raiz quadrada. O programa foi desenvolvido na linguagem *Delphi 10.2 Tokyo Starter*, com uma licença concedida a estudantes, portanto não demandou investimento financeiro. Além da análise de aplicativos já em uso em sistemas operacionais e desenvolvimento de uma nova proposta de aplicativo calculadora, também foi realizada pesquisa teórica em artigos científicos e

acadêmicos, e bibliografias que discorrem a respeito de Design de Interface, usabilidade, Tecnologia Assistiva, pessoa com deficiência visual de baixa visão e Programação Orientada a Objetos.

Do ponto de vista do profissional e/ou pesquisador do Design que se envolve no processo de desenvolvimento de projetos de interface em ambiente virtual, e que entende que essa é uma atividade de representação gráfica por meio de um determinado suporte, com finalidade prática, este trabalho apresenta uma possível aplicação da programação, mostrando essa como uma ferramenta que aliada do Design de Interface, pode melhorar a experiência do usuário com baixa visão.

2 Referencial teórico

A seguir estão relacionados autores que discorrem sobre as áreas de interesse deste trabalho, como informações referentes à Baixa Visão, suas causas frequentes, graus e níveis dessa deficiência, e também dados estatísticos. São apresentadas também informações referentes à usabilidade no âmbito das interfaces projetadas para pessoas com deficiência visual de baixa visão, conceitos e dados relacionados à Tecnologia Assistiva e, por fim, informações referentes à Programação Orientada a Objetos.

2.1 Pessoas com baixa visão

A baixa visão, também conhecida como visão subnormal, decorre de uma grande perda da visão e que pode ser indicada quando a visão está abaixo de 20% nos dois olhos. Porém, ao contrário do que ocorre no caso de cegueira, o indivíduo com baixa visão possui alguma funcionalidade da visão preservada (DUARTE, 2015). Segundo a Cartilha do Censo 2010, no Brasil, considerando a população residente no país, 23,9% declarou possuir pelo menos uma das deficiências investigadas: visual, auditiva, motora e mental ou intelectual, sendo que a deficiência visual é a que apresentou maior ocorrência, afetando 18,6% da população. Do número total de pessoas com deficiência, 3,46% representa as com deficiência visual severa e 1,6% são totalmente cegas (BRASIL, 2012).

Indivíduos com baixa visão, quando bem orientados por oftalmologistas, podem ter acesso a recursos que podem facilitar sua vida, o capacitando a enxergar mais e melhor. No entanto é importante citar que a condição de baixa visão não pode ser corrigida ou reduzida com o uso de óculos, lentes de contato ou cirurgias, e as causas

principais dessa condição são Degeneração Macular Relacionada à Idade (DMRI), Glaucoma e Retinopatia Diabética (DUARTE, 2015).

Conforme aponta Silva (2013), existem diferentes graus de baixa visão e estes decorrem de diferentes níveis de agravamento de deficiência visual. As causas mais frequentes que levam a um diagnóstico de baixa visão estão exemplificadas a seguir, nas Figuras 1, 2, 3 e 4.

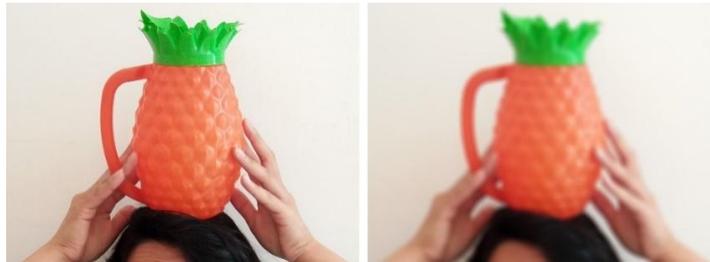


Figura 1 - Simulação comparativa entre visão de uma pessoa sem doença ocular e visão de uma pessoa com Catarata ou Deslocamento de Retina. Fonte: Adaptado de Kulpa (2009).



Figura 2 - Simulação comparativa entre visão de uma pessoa sem doença ocular e visão de uma pessoa com Glaucoma ou Retinose Pigmentar. Fonte: Adaptado de Kulpa (2009).



Figura 3 - Simulação comparativa entre visão de uma pessoa sem doença ocular e visão de uma pessoa com Retinopatia da Prematuridade ou Diabética. Fonte: Adaptado de Kulpa (2009).

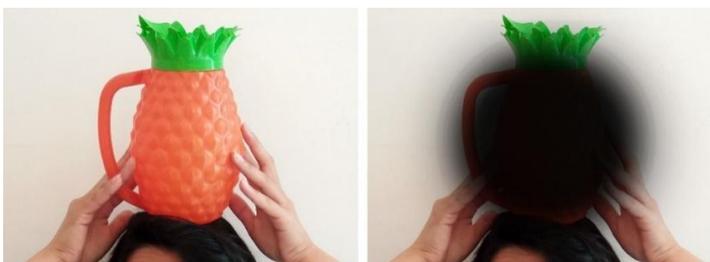


Figura 4 - Simulação comparativa entre visão de uma pessoa sem doença ocular e visão de uma pessoa com Degeneração Macular (DMRI) ou Toxoplasmose Congênita. Fonte: Adaptado de Kulpa (2009).

Como visto anteriormente, são várias as causas que levam à condição de baixa visão, portanto visando contemplar indivíduos com diferentes situações funcionais visuais, para o desenvolvimento do presente trabalho, foi eleito um modelo de cores que alcança diferentes diagnósticos de baixa visão (Figura 5).

Contrastes com Fundo em Preto	Elementos da Interface					
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto		Rodapé
Letra em Branco	Auxilia na identificação • Facilita a leitura • Emocionalmente neutro					
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto		Rodapé
Letra em Azul Claro	Melhor contraste na leitura • Auxilia na memorização e identificação					
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto		Rodapé
Letra Amarelo Luminoso	Melhor contraste na leitura • Auxilia na memorização e identificação • Fontes pequenas: negrito					
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto		Rodapé
Letra Verde Luminoso	Leitura adequada • Esteticamente apreciável					

Figura 5 - Combinações tonais que alcançam diferentes diagnósticos de baixa visão. Fonte: Kulpa, Teixeira e Silva, 2010.

De acordo com Kulpa, Teixeira e Silva (2010), as cores exercem papel de fundamental importância na Interação Humano Computador, sendo os elementos responsáveis por indicar aspectos da interface, destacar informações, criar conexões que favoreçam a memorização de conteúdos, direcionar fluxos de leitura e ainda permitem que os usuários criem novas significações por meio da interpretação da interface com a qual interage.

2.2 Usabilidade de interfaces para pessoas com deficiência visual

O termo usabilidade, conforme apontado por Carneiro e Ishitani (2014) e com base na definição proposta pela Associação Brasileira de Normas Técnicas (ABNT, 2002), faz referência à medida na qual um produto ou um serviço pode ser usado por usuários específicos visando alcançar objetivos específicos com efetividade, eficiência e satisfação, em um contexto específico de uso. A NBR 9241-11, norma que trata de requisitos ergonômicos para o trabalho em escritórios com computadores, indicando orientações a respeito de usabilidade, apresenta esses critérios fundamentais de usabilidade da seguinte forma:

- Eficácia: atendimento aos objetivos estabelecidos;
- Eficiência: recursos utilizados tendo em vista a abrangência dos resultados alcançados;
- Satisfação: inexistência do fator desconforto durante o uso do produto ou serviço.

Dessa forma, usabilidade é um atributo de produtos ou serviços que são fáceis de usar e de aprender (MELLO, 2016), que são fatores que promovem a autonomia do usuário. A usabilidade, no ambiente *web*, faz referência a cinco critérios fundamentais: facilidade de aprendizagem, eficiência no uso, facilidade de memorização, baixo grau de tendência a erros e satisfação do usuário (NIELSEN, 1993). Tais critérios são reforçados por conceitos apresentados por Preece, Rogers e Sharp (2008), que apresentam o que chamam de dimensões de usabilidade no projeto de um sistema interativo, que propõem metas de usabilidade bem similares às apresentadas por Nielsen (1993), são elas: apresentar eficácia e eficiência no uso, oferecer segurança, ser útil, ser de fácil aprendizado (assimilação) e ainda ter uma forma de uso fácil de lembrar.

Um dos fatores que indicam a qualidade de desempenho de uma interface é sua adequação às necessidades do usuário, considerando que o sistema deve sempre se adequar às necessidades dele, não o contrário (KULPA; TEIXEIRA; SILVA, 2010). Por possuírem limitações funcionais relativas à visão, indivíduos com baixa visão pouco são atendidos em suas necessidades quando necessitam interagir com interfaces que foram desenvolvidas com foco em um público geral (KULPA; TEIXEIRA; SILVA, 2010). Percebe-se, dessa forma, a necessidade de desenvolvimento de produtos e serviços focados nesse público, promovendo um maior grau de autonomia e independência em ambiente virtual.

Visando preencher uma lacuna nesse âmbito, Kulpa, Teixeira e Silva (2010) desenvolveram um estudo que gerou um Modelo de Cores para auxiliar designers e desenvolvedores durante o processo de construção de interfaces em ambiente virtual. O Modelo voltado especificamente para deficientes de baixa visão foi testado e visa potencializar a usabilidade em interfaces computacionais, promovendo uma melhora na Interação Humano Computador (IHC), que está diretamente associada ao modo como o usuário utiliza o computador, partindo do pressuposto que ele o utiliza como uma ferramenta para executar, simplificar ou apoiar tarefas (CUSIN; VIDOTTI, 2009).

2.3 Tecnologia Assistiva

De acordo com a *Assistive Technology Act* (1998), Tecnologia Assistiva compreende qualquer item adquirido comercialmente (podendo ser parte de equipamento ou equipamento assistivo), modificado ou personalizado, que é usado com a finalidade de melhorar o desempenho da capacidade funcional de um indivíduo com deficiência. Tecnologia Assistiva pode ser de baixa ou alta tecnologia.

De forma mais ampla e segundo apontamentos de pesquisadores brasileiros, a Tecnologia Assistiva faz referência a um campo científico que visa pesquisa, planejamento, criação e utilização de equipamentos e/ou dispositivos que potencializam a funcionalidade da pessoa com deficiência na realização de uma ou mais atividades (MELLO *et al.*, 2004; ROCHA; CASTIGLIONI, 2005).

Conforme aponta Leite (2016), atualmente existem sete classes diferentes de Tecnologia Assistiva: adaptações cognitivas, adaptações no meio ambiente (residencial, escolar, profissional e público), adaptações que facilitam o acesso ao computador, adaptações veiculares, adequação postural em cadeiras de rodas, comunicação alternativa (suplementar ou aumentativa) e equipamentos usados para esporte e recreação. Todas as classes citadas representam focos de pesquisas e intervenções clínicas, o que justifica os crescentes avanços tecnológicos que impactam diretamente no desenvolvimento de aparatos assistivos.

Assim como existem sete classes diferentes de Tecnologia Assistiva, existem sete aspectos que devem ser levados em consideração no processo de desenvolvimento de adaptações assistivas, conforme apontado por Hohmann e Cassapian (2011). São eles:

[...] análise da atividade, assimilação do problema, conhecimento dos princípios de compensação, sugestões de solução, pesquisa de recursos alternativos para a resolução do problema, manutenção periódica da adaptação e treino da adaptação na atividade (HOHMANN; CASSAPIAN, 2011, p. 11).

De acordo com Araújo (2007), o termo adaptação, no âmbito da Terapia Ocupacional, faz referência às modificações que são feitas no ambiente, tarefa ou método, que têm por objetivo a potencialização da funcionalidade do indivíduo, ao mesmo tempo que promove um maior grau de independência no desempenho de suas atividades.

Ainda conforme Hohmann e Cassapian (2011), a avaliação do potencial de inclusão do indivíduo nas Atividades de Vida Diária é um aspecto importante e que deve ser considerado no processo de implementação de adaptações, para que seja possível mensurar o grau de independência proporcionado por elas e também para que possa orientar possíveis modificações que podem ser feitas em outros contextos.

Como citado, Tecnologia Assistiva pode ser de baixa ou alta tecnologia, sendo que as adaptações que são propostas e implementadas se enquadram em duas categorias. Adaptações que se enquadram na categoria de baixa tecnologia (ou de baixo custo) são chamadas de *low-tech* e compreendem equipamentos e/ou

dispositivos destinados ao auxílio nas Atividades de Vida Diária, enquanto as adaptações que se enquadram na categoria de alta tecnologia (ou de alto custo), chamadas de *high-tech*, compreendem equipamentos e/ou dispositivos que demandam o uso de tecnologias avançadas no seu processo de desenvolvimento e produção (TEIXEIRA; OLIVEIRA, 2007).

2.4 Programação Orientada a Objetos

Existem diversas linguagens de programação que podem ser usadas para o desenvolvimento de *softwares*, e essas linguagens podem seguir diferentes paradigmas. Atualmente, um dos paradigmas mais difundidos é o da Programação Orientada a Objetos (POO). A evolução desse paradigma tem ocorrido devido principalmente a duas questões: segurança e reaproveitamento de códigos, que são fatores de fundamental importância no desenvolvimento de aplicações modernas.

Segundo Cantù (2016), *Object Pascal* é uma linguagem que segue o paradigma da POO e é usada para programar desde aplicativos para computador até aplicativos do tipo cliente-servidor, sendo usada também para programar aplicativos para telefones celulares e *tablets*, sistemas de automação industrial, redes de internet etc. O autor ainda afirma que essas são apenas algumas das possibilidades de uso da linguagem, apesar de ela não se limitar a isso.

Conforme aponta Gasparotto (2014), a Programação Orientada a Objetos se fundamenta em quatro pilares: abstração, encapsulamento, herança e polimorfismo. Abstração é um dos pontos mais importantes dentro de qualquer linguagem POO. Como o programador está lidando com uma representação de um objeto real, três quesitos devem ser considerados: a identidade que será criada para o objeto, as propriedades (que são as características) que dizem respeito a esse objeto e, por fim, os métodos, que se referem às ações que esse objeto irá executar. Encapsulamento é a técnica responsável por conferir segurança à aplicação em uma POO, pois ela oculta suas propriedades, criando uma espécie de ambiente seguro, como uma metáfora de caixa-preta. A herança é a técnica que possibilita a reutilização de códigos, e representa uma grande vantagem, pois otimiza a produção em tempo e em linhas de código. E o polimorfismo, por sua vez, se refere a alteração do funcionamento interno de um método que foi herdado de um objeto-filho por um objeto-pai.

Os novos *softwares* são, geralmente, orientados a objetos (O'DOCHERTY, 2005). Quando se trata de desenvolvimento de *softwares* comerciais, por exemplo, o trabalho vai além do simples ato de escrever linhas de código, pois contempla etapas como a

pesquisa e investigação de requisitos do negócio, análise do problema e por fim o desenvolvimento de uma proposta de solução, que são etapas básicas do processo de design, como pode ser visto em Dresch, Lacerda e Júnior (2015).

A linguagem Delphi está compreendida no âmbito do paradigma da Programação Orientada a Objetos, e é tida como uma linguagem acadêmica, pois devido a sua facilidade de manutenção, se torna mais acessível ao aprendizado e, o que é mais importante, estabelece o contato do programador com conceitos de lógica de programação que podem, inclusive, ser utilizados em outras linguagens.

3 Metodologia

O desenvolvimento do presente trabalho se deu em três etapas: 1) seleção de três exemplos de aplicativos calculadora presentes do cotidiano da autora: um do *notebook* pessoal e dois de *desktops* do laboratório de pesquisa Virtual Design (ViD), na Faculdade de Engenharia da Universidade Federal do Rio Grande do Sul (UFRGS), 2) análise dos exemplos com base nos parâmetros abordados no referencial teórico sobre interface e usabilidade e 3) desenvolvimento do aplicativo-piloto por meio de Programação Orientada ao Objeto na linguagem Delphi. Cada uma das etapas será descrita de forma mais detalhada nos tópicos seguintes.

3.1 Apresentação e análise dos exemplares selecionados

Os exemplares selecionados estão dispostos juntamente com suas respectivas análises, sempre sob os apontamentos dos parâmetros definidos após a pesquisa teórica, que possibilitou a definição dos seguintes norteadores do processo de desenvolvimento da interface: tamanho da janela, tamanho dos botões, tamanho, corpo e peso dos caracteres utilizados e utilização de paleta cromática que possibilite alto contraste entre primeiro (visor e botões) e segundo (fundo) planos.

É importante ressaltar que o presente trabalho se propôs a desenvolver um aplicativo por meio de embasamento teórico e atendimento a parâmetros definidos após análise de exemplares de uso cotidiano que apresentavam alguns pontos de melhora.

A Figura 6 apresenta uma calculadora padrão, que é o primeiro exemplar selecionado, presente no *notebook* de uso pessoal da autora. O aplicativo ainda oferece funções além da calculadora padrão, como a científica, a de programador e também um conversor de medidas. O acessório abre com uma proporção de 135x240 pixels e não oferece a opção de aumentar ou redimensionar a janela (por meio da

função clica-arrasta-solta da seta), e quando o usuário clica na opção de maximizar a janela, ela apenas aumenta e redistribui os elementos na tela, não apresentando aumento no corpo dos caracteres, como pode ser visto na Figura 7.

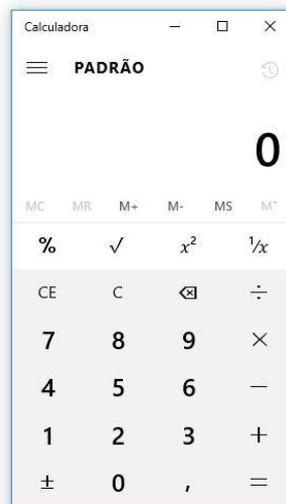


Figura 6 - Captura de tela de calculadora padrão *Windows 10 Pro* - primeiro exemplar.

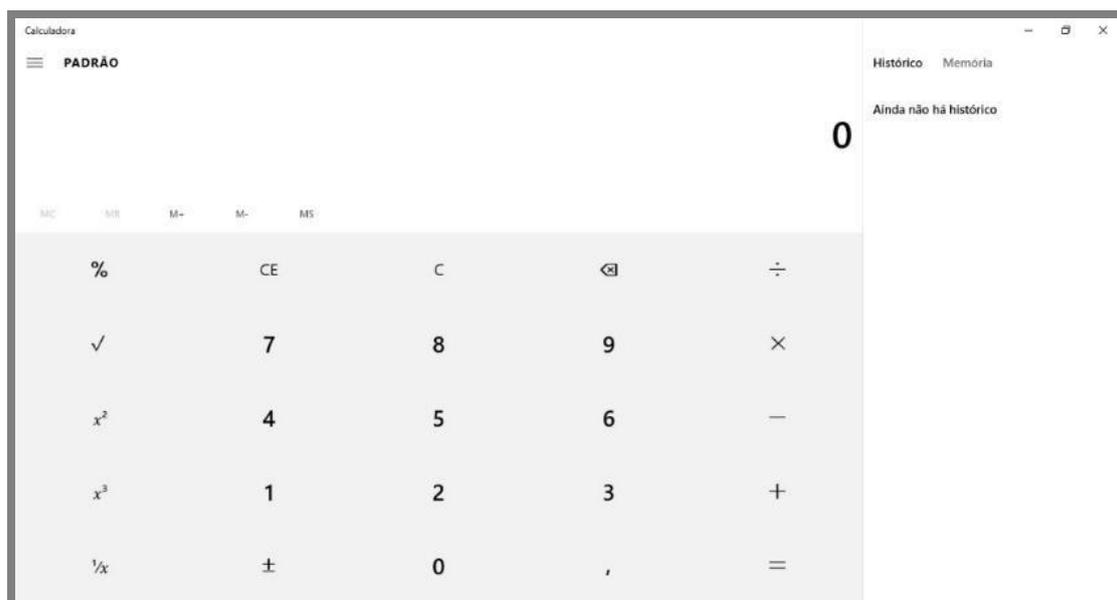


Figura 7 - Captura de tela de calculadora padrão *Windows 10 Pro* na visualização de tela cheia - primeiro exemplar.

A interface é bem organizada e limpa, porém não apresenta uma combinação tonal que favorece a experiência de uso do indivíduo com baixa visão de uma forma mais ampla, pois como visto no item 2.1 e apontado por Kulpa, Teixeira e Silva (2010), diferentes combinações cromáticas podem ser utilizadas para atender diferentes indivíduos com baixa visão, conforme modelo proposto pelos autores. A interface do

primeiro exemplar apresenta uma paleta cromática que favorecerá usuários com sensibilidade à luz, pois apresenta caracteres em preto sobre fundo cinza claro (KULPA; TEIXEIRA; SILVA, 2010), como mostra a Figura 8.

Contrastes com Fundo Cinza Claro	Elementos da Interface					
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto	Titulos	Rodapé
Letra em Azul Escuro	Facilita a leitura • Segurança de Navegação • Seriedade e Confiança					
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto	Titulos	
Letra em Preto	Facilita a leitura • Segurança de Navegação					
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto	Titulos	
Letra em Violeta Médio	Facilita a leitura					
		Menu principal	Sub-Menu		Titulos	
Letra em Vermelho Vivo	É indicado utilizar fonte em negrito					
		Menu principal				
Letra em Laranja	É indicado utilizar fonte em negrito					
			Sub-Menu			
Letra em Verde Escuro	É indicado utilizar fonte em negrito					

Figura 8 - Combinações tonais que favorecem a experiência de leitura de indivíduos com sensibilidade à luz. Fonte: Kulpa, Teixeira e Silva (2010).

O segundo exemplar (Figura 9) também é proveniente de plataforma *Windows* e é um dos exemplares selecionados de *desktops* do laboratório de pesquisa Virtual Design (ViD). Assim como o primeiro exemplar, o segundo exemplar não oferece opção de aumento ou redimensionamento da janela e tampouco permite que a janela seja maximizada – como pode ser visto na própria interface do aplicativo, que está com a opção maximizar desligada. A janela do acessório abre com uma proporção de 228x321 pixels, o que pode ser um fator limitador do uso por parte do indivíduo com baixa visão, já que os botões são pequenos e a interface foi desenvolvida em tons de azul, não apresentando um contraste muito efetivo entre caracteres, botões e fundo.

Semelhantemente ao primeiro exemplar, o segundo exemplar apresenta uma interface amigável ao indivíduo com sensibilidade à luz, apesar de não possuir fundo cinza claro. O fundo azul claro em contraste com os caracteres em azul escuro se aproxima do modelo proposto por Kulpa, Teixeira e Silva (2010), na Figura 8.



Figura 9 - Captura de tela de calculadora *Windows* - segundo exemplar.

Proveniente de plataforma MAC OS, o terceiro exemplar (Figura 10) é o segundo dos exemplares selecionados de *desktops* do laboratório de pesquisa ViD. O acessório abre com uma janela na proporção de 220x285 pixels e oferece a opção de maximizar janela, porém o comando revelou duas curiosidades.

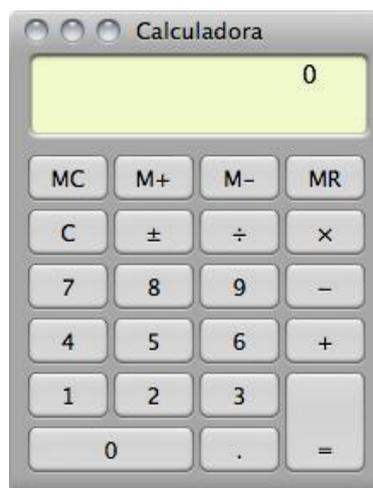


Figura 10 - Captura de tela de calculadora *Mac OS X* - terceiro exemplar.

Ao clicar no ícone maximizar janela representado pelo símbolo “+”, o aplicativo expande para a proporção 443x286 pixels (Figura 11) e acrescenta novas funções à calculadora padrão (Figura 10). O ícone maximizar continua presente na janela e ao ser clicado novamente, ele altera a proporção da janela para 363x362 pixels (Figura 12) e apresenta uma calculadora com funções bem diferentes das presentes nos exemplares anteriores. Dessa forma percebeu-se que a opção maximizar não se refere à possibilidade de aumentar a janela, mas acessar outras opções do aplicativo calculadora.

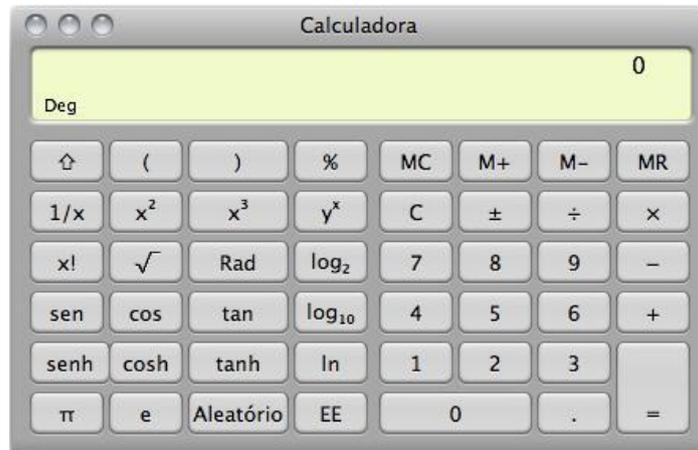


Figura 11 - Captura de tela de calculadora Mac OS X na opção tela cheia 1 - terceiro exemplar.

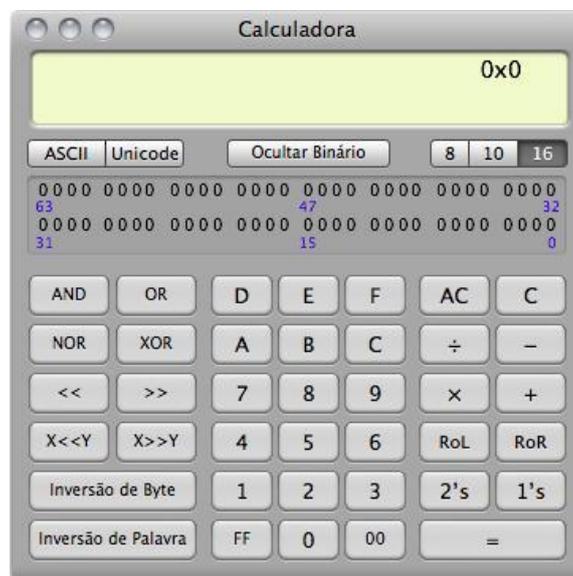


Figura 12 - Captura de tela de calculadora Mac OS X na opção tela cheia 2 - terceiro exemplar.

O terceiro exemplar apresenta botões com contorno bem definido e simulação de volume por meio de sombras suaves, o que favorece a definição de limites visuais por parte do usuário. Apesar de os botões serem bem definidos, apresentam dimensões que podem dificultar a experiência de uso, bem como o tamanho dos caracteres. A combinação de tons de cinza, assim como visto no primeiro e segundo exemplares, podem favorecer a experiência do usuário com sensibilidade à luz. E ainda de acordo com o Modelo de Cores proposto por Kulpa, Teixeira e Silva (2010), contrastes com fundo em cinza escuro (na faixa de 65%) harmonizam interfaces que apresentam muitas informações gráficas e cromáticas (Figura 13).

Contrastes com Fundo Cinza 65%	Elementos da Interface				
	Cabeçalho				
Letra em Branco	Contraste neutro • só utilizar com fonte de maior tamanho				
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto	
Letra em Azul Claro	Leitura adequada				
	Cabeçalho	Menu principal	Sub-Menu	Corpo de Texto	
Letra Amarelo Luminoso	Leitura adequada				

Figura 13 - Combinações tonais com fundo cinza escuro. Fonte: Kulpa, Teixeira e Silva (2010).

3.2 Programação em linguagem Delphi

Como abordado anteriormente, a linguagem Delphi está compreendida no âmbito da Programação Orientada a Objetos e foi escolhida para o desenvolvimento do presente trabalho pelo fato de ser uma linguagem acadêmica com interface intuitiva e também pelo fato de ser a linguagem de programação com a qual a autora tem mais proximidade. Para o desenvolvimento dessa etapa, foi utilizado o *software Delphi 10.2 Tokyo Starter*, por meio de uma licença educacional.

A vantagem de se trabalhar com esse *software* é o fato de o desenvolvedor trabalhar a parte gráfica (a interface) ao mesmo tempo em que programa o código, tendo apenas que alternar entre as respectivas abas. Como o presente trabalho visava atender determinados requisitos referentes à interface e usabilidade focadas na experiência de uso de indivíduos com baixa visão, a possibilidade de trabalhar interface e programação de código ao mesmo tempo significou otimização de tempo.

4 Apresentação e discussão de resultados

Visando contemplar indivíduos com diferentes situações funcionais relacionadas à visão, foi usado o modelo de cores que alcança diferentes diagnósticos de baixa visão, desenvolvido por Kulpa, Teixeira e Silva (2010), conforme Figura 5, apresentada no item 2.1 deste trabalho.

Conforme Kulpa, Teixeira e Silva (2010), contrastes com fundo preto combinados com caracteres em branco auxiliam na identificação dos elementos da interface, facilitam a leitura e ainda são emocionalmente neutros. O fundo preto potencializa o contraste entre os botões (que estão em cinza escuro) que calçam e destacam os caracteres (letras, números e símbolos matemáticos) que estão dispostos em branco, como pode ser visto na Figura 14.

A janela do aplicativo abre na proporção de 364x521 pixels e apresenta botões maiores do que os presentes nos exemplares que foram apresentados no item 3.1, o visor também apresenta resultados com caracteres com um corpo de fonte maior e os caracteres dos botões também são maiores. Todos esses pontos de melhora podem ser observados, de forma mais direta, também na Figura 14.



Figura 14 - Interface do aplicativo desenvolvido e seus respectivos pontos de melhora.

A programação do código se deu de forma paralela ao desenvolvimento da interface da calculadora e demandou pesquisas e uma série de testes para validação dos comandos e ações, bem como correções e adaptações. O aplicativo-piloto foi testado apenas internamente, não tendo sido aplicado a pessoas com deficiência visual. Quanto a isso cabe ressaltar o fato de que o Modelo de Cores utilizado como base é resultado de uma pesquisa já publicada, e foi testado por 10 usuários (colaboradores voluntários) com baixa visão, conforme Kulpa, Teixeira e Silva (2010). Em relação ao objetivo geral deste trabalho, essa não é uma questão comprometedora, dado que a proposta geral é apresentar a Programação como uma ferramenta que pode ser aplicada para potencializar o desenvolvimento de projetos gráficos em ambiente digital. Entretanto essa é uma oportunidade de prosseguimento do trabalho, a aplicação a usuários com baixa visão, para colher opiniões, corrigir possíveis falhas, realizar novos testes e propor novas aplicações e funcionalidades.

5 Considerações finais

A programação, seja ela estruturada ou orientada a objetos, é uma ferramenta que pode ser usada visando melhorar o desempenho pessoal, acadêmico ou profissional do designer em suas tarefas, mas que também pode ser usada na proposição de resolução de problemas ou adequações em tarefas desempenhadas por terceiros, já que a programação, de forma mais generalizada, está presente em muitos aspectos do cotidiano e ainda é um instrumento que pode ser usado para estimulação do desenvolvimento do pensamento lógico.

Atualmente a metodologia usada para desenvolvimento de *softwares* é, em boa parte, orientada a objetos, o que faz com que a linguagem usada para o desenvolvimento seja, da mesma forma, orientada a objetos. O foco no objeto tem sido um dos grandes impulsionadores da indústria de programação, e a tendência é que continue da mesma forma por um longo tempo, devido ao aumento da demanda de desenvolvimento de aplicativos nativos de telefones celulares como também dos híbridos, que podem ser executados em telefones celulares, computadores e *tablets* por exemplo.

A programação de códigos, tendo como base requisitos definidos por meio de critérios de acessibilidade e usabilidade, representa a conexão valiosa entre áreas de conhecimento que, juntas, proporcionam o desenvolvimento de ferramentas que podem facilitar a vida de usuários e também melhorar o seu desempenho em determinadas tarefas. O design de interface é, sem dúvida, um fator de potencialização da acessibilidade, atuando como um elemento agregador na programação de códigos.

O aplicativo desenvolvido e apresentado neste trabalho é um exemplo do fato de que a programação orientada ao objeto pode ser uma ferramenta de real utilidade no desenvolvimento de projetos gráficos em ambiente digital, tanto para um público típico, quanto para um público com algum tipo de deficiência. É de fundamental importância que o designer compreenda a extensão e a potencialidade da sua possibilidade de atuação como um profissional socialmente ativo, e diretamente envolvido na proposição de soluções para problemas humanos contemporâneos, atuando também como um promotor do respeito às diferenças e da valorização da diversidade em ambiente virtual.

Referências

- ARAUJO, P. M. P. Adaptações. In: GREVE, J. M. A. **Tratado de Medicina de Reabilitação**. São Paulo: Roca, 2007. p. 325-329.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). NBR 9241-11 - **Requisitos ergonômicos para trabalho de escritórios com computadores: parte 11 - orientações sobre usabilidade**. Rio de Janeiro: 2002.
- BRASIL. Secretaria de Direitos Humanos da Presidência da República, Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência. **Cartilha do Censo 2010: pessoas com deficiência** [Internet]. Brasília, DF; 2012. Disponível em: <<https://goo.gl/cPkM3p>>. Acesso em: jul. 2018.
- CANTÙ, M. **Object Pascal Handbook**. Delphi 10.1 Berlin Edition ed. [S.l.]: Peter W A Wood, 2016.
- CARNEIRO, R. V.; ISHITANI, L. Aspectos de usabilidade de mobile learning voltado para usuários com restrições decorrentes da idade. **Revista Brasileira de Computação Aplicada**, 2014. v. 6, n. 1, p. 81–94.
- CUSIN, Cesar Augusto; VIDOTTI, Silvana Aparecida Borsetti Gregorio. Inclusão digital via acessibilidade web. **Liinc em Revista**, 2009. v.5, n.1, p. 45-65. Disponível em: <<http://revista.ibict.br/liinc/article/view/3189/2851>>. Acesso em: jul. 2018.
- DRESCH, Aline; LACERDA, Daniel Pacheco; JÚNIOR, José Antonio Valle Antunes. **Design Science Research: método de pesquisa para o avanço da ciência e tecnologia**. 1ª Ed. Porto Alegre: Bookman, 2015.
- DUARTE, F. V. Baixa Visão. **IORJ | Consultas, Exames e Cirurgias Oftalmológicas**, [S.l.], 7 set. 2015. Disponível em: <<http://www.iorj.med.br/baixa-visao/>>. Acesso em: 18 jul. 2017.
- GASPAROTTO, Henrique Machado. POO: Os 4 pilares da Programação Orientada a Objetos. **DevMedia - Códigos para quem faz Código**, [S.l.], 30 out. 2014. Disponível em: <<https://goo.gl/Zh358a>>. Acesso em: 4 jul. 2017.
- HOHMANN, P.; CASSAPIAN, M. R. Adaptações de baixo custo: uma revisão de literatura da utilização por terapeutas ocupacionais brasileiros. **Revista de Terapia Ocupacional da Universidade de São Paulo**, 2011. v. 22, n. 1, p. 10–18.
- KULPA, C. C. **A contribuição de um modelo de cores na usabilidade das interfaces computacionais para usuários de baixa visão**. Porto Alegre, Brasil: Universidade Federal do Rio Grande do Sul, 2009. Dissertação de Mestrado.
- KULPA, C. C; TEIXEIRA, F. G.; SILVA, R. P. Um modelo de cores na usabilidade das interfaces computacionais para os deficientes de baixa visão. **Design e Tecnologia**, set. 2010. v. 1, n. 01, p. 66–78.
- LEITE, A. Tecnologia Assistiva: o “Bê a bá”! **REAB**, [S.l.], 2016. Disponível em: <<http://www.reab.me/tecnologia-assistiva-o-be-a-ba/>>. Acesso em: 24 jun. 2017.

MELLO, A. G. Deficiência, incapacidade e vulnerabilidade: do capacitismo ou a preeminência capacitista e biomédica do Comitê de Ética em Pesquisa da UFSC. **Ciência & Saúde Coletiva**, 2016. v. 21, n. 10, p. 3265–3276.

MELLO, M. A. F.; CAPANEMA, V. M.; LUZO, M. P. Recursos tecnológicos em Terapia Ocupacional - Órtese e Tecnologia Assistiva. In: DE CARLO, M. P.; LUZO, M. C. M. **Terapia Ocupacional - Reabilitação Física e Contextos Hospitalares**. São Paulo: Roca, 2004. p. 99-125.

NIELSEN, J. **Usability engineering**. San Francisco: Morgan Kaufmann; 1993.

O'DOCHERTY, M. **Object-Oriented Analysis and Design: understanding System Development with UML 2.0**. England: John Wiley & Sons Ltd, 2005.

PREECE, J.; ROGERS, Y.; SHARP, H. **Design de interação**. Porto Alegre: Bookman, 2008.

ROCHA, E. F.; CASTIGLIONI, M. Do C. Reflexões sobre recursos tecnológicos: ajudas técnicas, tecnologia assistiva, tecnologia de assistência e tecnologia de apoio. **Revista de Terapia Ocupacional da Universidade de São Paulo**, 1 set. 2005. v. 16, n. 3, p. 97–104.

SILVA, R. S. **Contribuições do Design de Produto e Usabilidade no Projeto de Brinquedos: um Estudo Focado na Criança com Deficiência Visual**. Porto Alegre, Brasil: Universidade Federal do Rio Grande do Sul, 2013. Dissertação de Mestrado.

TEIXEIRA, E.; OLIVEIRA, M. C. Adaptações. In: FERNANDES, A. C., CASALIS, M. E. P.; RAMOS, A. C. R. **Medicina e Reabilitação** - Princípios e Práticas. São Paulo: Roca, 2007. p. 671-706.

U.S. Government Printing Office. Public Law 105-394. Assistive Technology Act of 1998. 1998.