

## **A INTERFACE DO SOFTWARE E AS FORMAS MARCADAS E NÃO MARCADAS DA PRESENÇA DE OUTREM\***

*Aguinaldo Gomes de Souza – ETE-PE/HIPERGED-UFC*

**RESUMO:** Neste trabalho analisamos as marcas da intersubjetividade deixadas em interfaces de softwares. O que entendemos por isso é que, tanto nos softwares quanto nas interfaces desses softwares, existem formas marcadas e não marcadas de presença do outro, deixadas por desenvolvedores, que perpassam desde o processo de fabricação até o produto em uso por outros sujeitos. Para tanto, fazemos uma breve retomada sobre a origem e evolução das interfaces de software para estudarmos as interfaces pelo viés da Linguística. Concluímos que as formas de enunciações materializadas em uma interface de software são material de base por meio do qual o usuário do software estabelecerá relação com outros sujeitos além do software.

**PALAVRAS-CHAVE:** Interface. Intersubjetividade. Linguística.

### **A ORIGEM DAS INTERFACES DE SOFTWARE**

A interface gráfica de um software é um elemento introdutório que facilita o uso do software pelo usuário. Ela é composta por elementos textuais, de imagens, sonoros e por sequências de controle que permitem o ser humano interagir não só com os softwares mas também com outros sujeitos. É por meio da interface gráfica (do inglês, graphical user interface - GUI) que nós, enquanto usuários desses artefatos, podemos manter relação com o software – podemos digitar um artigo, escrever um e-mail, enviar um formulário etc. É também por meio da interface gráfica do software que reconhecemos os diversos tipos de softwares que existem.

A história da interface se confunde com a história dos primeiros softwares modernos, sabe-se que a interface gráfica como conhecemos hoje foi desenvolvida no início da década de 1970 pelos pesquisadores da Xerox PARC; como aponta Johnson (2001 p.51), a Xerox era um lugar intelectualmente desafiante. Nessa empresa foi desenvolvida não só a primeira interface gráfica como também o mouse, as janelas e os mapeamentos de bits o qual nos deu a imagem visual para a organização das informações.

O desenvolvimento de softwares com janelas mudou substancialmente o modo de operar um computador. Aos poucos, essa invenção, viabilizada na prática pelos profissionais

\* Acesso ao registro da comunicação em Fórum: <<http://www.textolivre.org/forum/viewtopic.php?f=12&t=3835>>.

da Xerox, causou um impacto tão grande no campo da informática que hoje é praticamente impossível pensar em softwares sem interface gráfica. As janelas estabelecem limites entre os programas, por meio delas é feita a delimitação do espaço do software. Elas representaram uma evolução tão grande no conceito de como operar um computador, que um dos sistemas operacionais mais usados no mundo recebeu o nome de Windows (janelas em inglês). Os softwares, os diretórios e os documentos que antes eram acessados por meio de comandos digitados (ver figura 01), passaram a ganhar um novo modo de operação: comandos enviados através de janelas. Uma interface gráfica que tem por base janelas, em vez de linhas de comando, permite ao usuário do computador executar inúmeras tarefas mais facilmente.

```

Directory of D:\
25/04/2009 18:16 <DIR>      _paulo_autran_-_4_seculos_da_poesia
18/03/2009 01:54      63.968.916 _paulo_autran_-_4_seculos_da_poesia.rar
09/04/2009 09:47          2.745 J_3_5v.gif
27/02/2009 16:23      338.368 758847877_08-B3-2009.pdf
25/04/2009 18:16 <DIR>      adobe_photoshop_cs3_br(com_plugins)
02/05/2009 17:31 <DIR>      aguinaldo_arquivos
11/04/2009 19:27          1.217 emotoposs.txt
25/03/2009 00:23      514.869 as_tecnologias_da_inteligencia.txt
01/12/2009 16:16 <DIR>      aulas
18/04/2009 22:03          648 calpe.log
18/04/2009 22:03      78.778 calpe97892009.pdf
23/04/2009 22:20      147.964.196 confli.5ell.death.suite.gfsilva.uww.therche
s.de(1).rar
13/04/2007 12:30          32.489 confli.5ell.death.suite.gfsilva.uww.therche
s.de(1).rar
16/07/2009 23:10 <DIR>      converteroutput
07/04/2009 08:53      2.206.510 cris.pdf
01/01/2000 02:51      677.657 da4.jpg
24/07/2009 17:00          846.336 dianna.doc
25/07/2009 00:21 <DIR>      downloads
17/11/2009 12:25 <DIR>      downloadsp
15/04/2009 22:57          314.368 ergo.aguinaldoseuza.doc
09/11/2009 00:42 <DIR>      fotos
19/04/2009 13:43          33.141 introdução_indiec.docx
07/04/2009 23:06      148.570 linguagem_e_trabalho.pdf
12/01/2010 10:04 <DIR>      livros_de_hakhtin
17/12/2009 21:32 <DIR>      meu_desktop
25/04/2009 20:34 <DIR>      meus_arquivos_recebidos
27/04/2009 00:30 <DIR>      ms-office-2007
25/04/2009 20:11 <DIR>      musica_soulseek
29/11/2009 10:56 <DIR>      meu_folder
29/11/2009 10:55 <DIR>      nova_pasta
20/04/2009 17:43          174.776 novo_regimento.pdf
11/02/2009 01:17      534.098.021 offent2007-halxebr_2.zip
23/04/2009 12:42          277.464 eo.jpg
07/01/2010 15:43 <DIR>      pendrive
09/04/2009 01:46          1.512.051 pbpc.jk.jp.pdf
23/04/2009 21:34          393.333 pict0485.jpg
10/04/2009 15:32          128.250 projeto_aguinaldo_reescritura.docx
26/03/2009 12:19          320.512 projeto_aguinaldo_revisandonaço2009.doc
14/04/2009 23:20          318.704 projeto_revisado.doc
26/04/2009 14:14          318.464 projeto_revisado.sampaio_25.4.09_(2).doc
14/04/2009 23:32          311.296 projeto_revisado_notasexplicativas.doc
20/04/2009 17:44          101.411 regimento_psletras.pdf
23/04/2009 23:03          1.188.713 uuu.jpg
12/12/2009 15:10 <DIR>      uuu
          27 File(s)      756.341.477 bytes
          17 Dir(s)    52.272.050.176 bytes free

D:\> aguinaldo
'aguinaldo' is not recognized as an internal or external command,
operable program or batch file.
    
```

Figura 1 – interface baseada em linhas de comando

Embora tenha sido a equipe da Xerox que elaborou a primeira interface baseada em janelas, é atribuído ao presidente da Apple, Steven Jobs, o mérito de ter libertado o conceito de janelas dos laboratórios da Xerox PARC. O desktop inicialmente foi vendido em conjunto com o Lisa (um computador desenvolvido pela Apple) e logo em seguida com o Macintosh.

Esse último, com seu desktop, mudou radicalmente o conceito de experiência de usuário. Antes dele o computador era uma máquina inerte, um objeto misterioso entre o usuário e o microprocessador (JOHNSON, 2001 p.55).

Tudo que veio depois do desenvolvimento da primeira janela pode ser considerado como evolução natural, não havendo praticamente mudanças nos padrões utilizados, ou seja, os programas de computador posteriores a primeira interface utilizam sempre os mesmos elementos (ícones, janelas, menus etc). Com as interfaces gráficas, o usuário do computador começou a se libertar de vários paradigmas, como por exemplo quando o usuário desejava excluir um arquivo ou um programa, teria que digitar um comando em texto para que o computador pudesse interpretar e excluir o arquivo. Com o desenvolvimento das interfaces ele apenas arrasta o arquivo para a lixeira. A interface gráfica baseada em janelas facilita o uso por meio da manipulação direta com o uso do mouse. Ver figura 2.

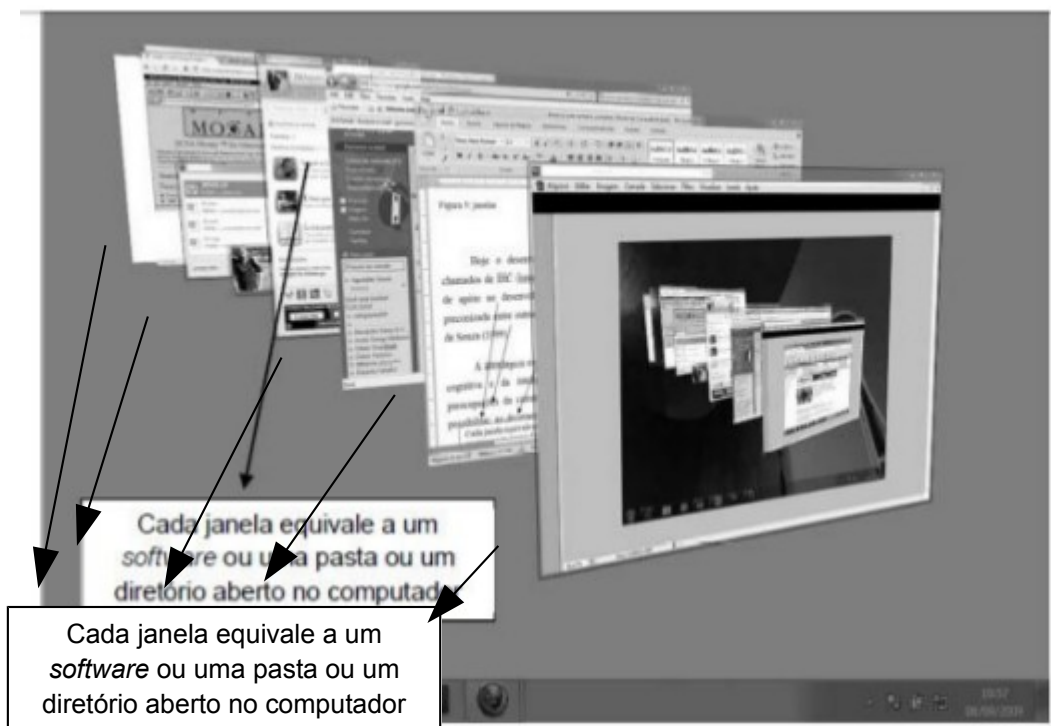


Figura 2 – janelas

Hoje o desenvolvimento de interfaces de software encontra-se no campo dos estudos chamados de IHC (Interação Humano-Computador). Existem duas correntes que servem de apoio ao desenvolvimento de interface baseada em janelas: a cognitivista (ou engenharia cognitiva) preconizada entre outros, por Norman (1991) e a de base semiótica (ou engenharia semiótica), apregoada por De Souza (1993; 2005).

A abordagem cognitivista advém diretamente da psicologia cognitiva, da ciência cognitiva e da inteligência artificial as quais estudam a cognição. Uma das preocupações da corrente cognitivista, quando do desenvolvimento das interfaces, é possibilitar, ao designer, o entendimento dos processos cognitivos humanos.

A ideia básica parte do uso de modelos cognitivos que descrevem os processos e estruturas mentais (e.g. recordação, interpretação, planejamento e aprendizado). Tais modelos podem indicar para pesquisadores e projetistas de interfaces quais propriedades devem ter os modelos de interação para que a interação se realize mais facilmente pelos usuários. Como estas abordagens adotam uma perspectiva centrada nos aspectos cognitivos do usuário, o design feito com base nelas é chamado de design de sistemas centrado no usuário (User Centered System Design – UCSD) (DE SOUZA et al., 1999, p. 13).

O designer de artefatos digitais deve ser capaz de construir um sistema (uma interface) que permita ao usuário, ao interagir com esse artefato, criar um modelo mental consistente com o modelo projetado pelo designer. Para ajudar os designers a entenderem como os usuários leigos utilizam uma interface, Norman (1991) argumenta que os designers precisam entender os processos mentais pelos quais os humanos interagem com a interface, propondo uma teoria da ação, segundo a qual a interação usuário-sistema é desempenhada num ciclo-de-ação ou sequência de ações.

Assim, os estudos voltados para interação homem-máquina, de base cognitivista, veem a interação com a interface, como uma travessia de dois golfos: o de execução e o de avaliação. A travessia do primeiro golfo está correlacionada à sequência de ações que o usuário terá que estabelecer para interagir com o sistema computacional; a segunda está correlacionada com a percepção da interface, a interpretação e a avaliação da meta a ser seguida para interagir com a máquina.

A crítica maior, feita à teoria cognitiva, é que ela focaliza apenas a interação usuário-sistema, desconsiderando o processo de design bem como o produto final do processo de design. Com o argumento de que a teoria de base cognitivista nega o rastro da subjetividade deixada pelos produtores dos softwares, De Souza (1999; 2005) propõe outra teoria para o campo da IHC: a da engenharia semiótica, que toma como base a semiótica de Peirce ([1839-1914] 2003).

Assim, como salienta De Souza et al., (1999, p.16).

nestas abordagens toda aplicação computacional é concebida como um ato de comunicação que inclui o designer no papel de emissor de uma mensagem para os usuários dos sistemas por ele criados.

Desse modo, a engenharia semiótica torna-se complementar a corrente cognitiva, uma vez que enfatiza que o design não está só buscando construir a imagem do sistema, mas buscando comunicar ao usuário sua própria visão do sistema.

Esta mensagem tem como objetivo comunicar ao usuário a resposta a duas perguntas fundamentais: (1) Qual a interpretação do designer sobre o(s) problema(s) do usuário?, e (2)

Como o usuário pode interagir com a aplicação para resolver este(s) problema(s)? O usuário concebe a resposta a estas perguntas à medida que interage com a aplicação. Assim, esta mensagem é unilateral, uma vez que o usuário recebe a mensagem concluída e não pode dar continuidade ao processo de comunicação [de Souza, 1993] naquele mesmo contexto de interação. Além disso, como esta mensagem (a interface) é ela mesma capaz de trocar mensagens com o usuário, ela é um artefato de comunicação sobre comunicação, ou meta-comunicação. (DE SOUZA et al., 1999 p.443),

Com a engenharia semiótica, a interação entre o usuário-sistema deixa de ser o foco principal, dando lugar para a expressão do designer e ao processo de design. Dessa forma, para a engenharia semiótica, o que se deve atentar é a mensagem do designer, enviada de forma unidirecional ao usuário do software.

## UM OLHAR DA METALINGUÍSTICA SOBRE SOFTWARES E INTERFACES

Em relação à linguagem, reconhecemos que tanto a engenharia cognitiva quanto a engenharia semiótica não respondem satisfatoriamente as diversas perguntas que podem ser colocadas, em relação ao nosso objeto de estudo, no âmbito da Ciência Linguística (como os sujeitos interagem com softwares e interfaces? Como as enunciações concretas presentes em interfaces de softwares colaboram para o uso do software? Como os ícones, imagens e menus auxiliam o usuário a estabelecer uma macrocompreensão entre camadas de informações e como isto colabora com o embate entre dois sujeitos que utilizam os softwares?). Nesse sentido, buscamos, na teoria dialógica bakhtiniana, subsídios para um estudo das relações de alteridade que se estabelecem também na relação homem-máquina (SOUZA, 2009).

Assim, iremos identificar todas as formas como os softwares e as interfaces se apresentam ancorados na metalinguística, disciplina proposta por Bakhtin (1981). O que entendemos por isso é que, tanto nos softwares quanto nas interfaces desses softwares, existem formas marcadas e não marcadas de presença do outro, deixadas por desenvolvedores, que perpassam desde o processo de fabricação, até o produto em uso por outros sujeitos em interação por meio de softwares.

Tomando por base esse princípio, Peres (2007 p.50) propôs uma nova forma de olhar os estudos sobre a IHC, partindo do arcabouço conceitual desenvolvido pelo Círculo de Bakhtin, envolvendo conceitos como autor, autoria, e dialogismo. Assim, enquanto a engenharia semiótica defende a ideia da 'mensagem concluída e unilateral' enviada do designer para o usuário do software, olhando por um prisma dialógico, a autora (PERES, 2007, p.15) argumenta que essas mensagens são vozes que se materializam em interfaces. Desta feita, quem interage com um software<sup>1</sup>.

---

<sup>1</sup> por exemplo, o software cliente de e-mail recebe mensagens em formato digital (gênero digital e-mail), além de interagir com outros sujeitos de forma diacrônica, encontra também respostas de outros sujeitos situados, uma vez que as vozes representadas nas enunciações que aparecem nas interfaces dos softwares.

Vistas pelo enfoque do princípio dialógico inerente à linguagem, as interfaces dos softwares possuem, em si, marcas linguísticas das formas de presença do outro, podendo revelar aspectos de interesse como: as relações de alteridade, os signos, os processos de intersubjetividade, as marcas estilistas nos enunciados presentes nas interfaces de softwares etc. De certa forma, essas formas de presença do outro são materializadas por vozes – no sentido bakhtiniano do termo, ou seja, a marca ideológica de um eu que se vincula a um lugar social –, encontra aí a sua maior ressonância. As relações dialógicas que perpassam os artefatos computacionais, são, pois, relações de alteridade entre um eu e um outro, situado em um lugar social. Daí o olhar do analista para a interface, não é mais uma mensagem já concluída – conforme apregoa os engenheiros semióticos –, mas são vozes e formas de presença do eu e do outro que guiam ações de uso, que conectam desenvolvedores e usuários, situados em polos distintos.

É mais especificamente no trabalho da equipe de programadores que irá desenvolver a interface do software – Graphical User Interface (GUI) – que as questões relacionadas a subjetividade tornam-se mais expressivas, considerando que esses profissionais de design têm, por missão, comunicar ao usuário do software, o que ele poderá fazer com o software. Assim, a pressuposição de um outro, a quem o software irá servir, pode interferir significativamente no acabamento do produto. Dessa forma, questões como: a quem a mensagem se destina? Que problemas a interface se propõe a resolver? Como interagir com sistemas para resolvê-los? são pensadas pelos protagonistas da atividade, enquanto estão desenvolvendo as interfaces dos softwares.

A ideia de comunicar algo a alguém e fazer com que esse outro, em potencial, assuma uma posição de destaque frente a um chamado manifestado por um artefato computacional (ver na figura 3 as enunciações dos desenvolvedores dos softwares para o usuário interagir com o software), é um dos trabalhos mais vulneráveis a interferência de terceiros dentro de uma fábrica de software.





Figura 3 – o software de e-mail acoplado no software de navegação Mozilla

Essas formas de enunicações apresentadas em uma interface de software, constituem-se como material de base sobre a qual o usuário do software interage. Um bom

exemplo dessas marcas de subjetividade que os protagonistas da atividade deixam, durante o processo de fabricação de software, é as ‘caixas de diálogo’ que aparecem quando o usuário está interagindo com um software.

Poderíamos então falar que uma interface de software carrega em si um ‘mínimo dialógico’ e que esse mínimo é marcado por uma série de formas de representação do outro que antecedem a criação de um artefato tecnológico. Esse ‘mínimo’ pode ser observado também nos embates dialógicos travados entre dois sujeitos situados, através de softwares aplicativos, como por exemplo, uma conversa que ocorre através de um comunicador instantâneo como o Messenger. Conforme apontam Meira;Peres (2004), este tipo de comunicação, mediada por um artefato tecnológico, é sempre co-dependente da organização da informação da interface.

Para esses autores, o software participa desses diálogos, uma vez que os enunciados, proferidos pelos interlocutores, são de certa forma, enunciações dos desenvolvedores e estão submetidas ao funcionamento do próprio software, ou mesmo estão misturadas com a própria interface, através de sugestões que aparecem na interface (enunciações dos desenvolvedores).

Dessa forma, o processo de uso de um software está diretamente relacionado com o processo de desenvolvimento do software. É nessa direção – processo de desenvolvimento/processo de uso –, que a instância humana, a presença de um sujeito historicamente situado, emerge.

Quando evocamos a ideia de que nas interfaces dos softwares existem enunciações de outrem dirigidas para alguém (SOUZA, 2009), estamos afirmando que essa pressuposição de um outro interfere significativamente no trabalho prescrito e no acabamento do produto (software). O acabamento pode ser estritamente linguístico – ou seja, quais enunciados devem vir na interface para que o sujeito, que irá utilizar o software, não tenha problemas de interpretação? – a um acabamento puramente estilístico –, considerado não só como um acabamento do design do produto, mas também em relação às marcas linguísticas deixadas nesse produto.

Nesse sentido, o estilo está, de tal forma, relacionado ao conceito de autor/autoria, que para uma análise substancial desse par seria necessário rever as relações constitutivas entre os diversos protagonistas da atividade (os desenvolvedores de software) e observar como essa sequência de relações entra em relação dialógica com o outro para quem o software é pensado. Assim, as marcas da gênese estilísticas em uma interface de software são marcas deixadas ao longo do processo de fabricação do software. Elas representam a consequência da interação entre os protagonistas desenvolvedores e da imagem que esses fazem daqueles a quem o software irá servir. Muito embora esse outro só exista, em um primeiro momento, no campo das ideias, é um outro idealizado. Essa pressuposição de um outro existe desde o momento de produção ao projeto de trabalho.

Vejamos a figura 04: nela temos um software aplicativo de código fechado, chamado de Internet Explorer, esse software de navegação na Internet, ao ser aberto, envia para o sujeito usuário a pergunta ‘what do you want to do next?’. Essa enunciação que não é propriamente uma enunciação do software, mas uma enunciação deixada pelos profissionais



que o fabricaram (mensagens na interface, sinais de alerta etc), já está pré-formatada, ou seja, a forma como se apresenta, a tipologia, as cores, a organização das opções a serem escolhidas tudo isso que em conjunto marcam o que chamamos de estilo, é dado.

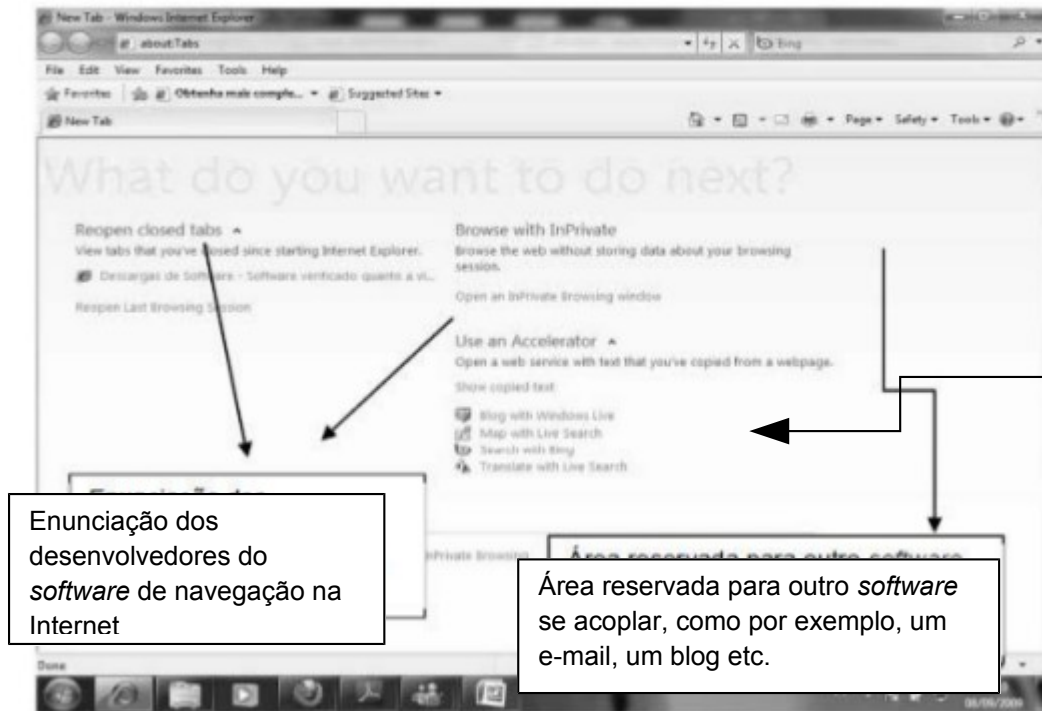


Figura 4 - software Internet Explorer

No trabalho prescrito, na produção de interfaces, a marca pessoal, dos desenvolvedores de software, aparece na escolha das cores, na escolha dos botões ou mesmo na escolha de uma tecnologia para fazer um tipo de acabamento aos enunciados: se serão sublinhados, se estarão em negrito, qual o tamanho da fonte etc. Assim, gêneros como o bate-papo e o curriculum on-line possuem formas estilísticas já pré-concebidas pelos desenvolvedores de software. Essa determinação estilística acaba possibilitando, e ao mesmo tempo limitando, a margem de ação de outros desenvolvedores de software e dos usuários dos softwares.

A esses últimos cabe interagir com esses softwares e aplicar – quando possível –, suas próprias marcas estilísticas. Em geral essas marcas são vistas em formas típicas de enunciados, em gêneros digitais: as abreviações dos chats, os emoticons, etc, são exemplos disso.

## **DAS MARCAS DA INTERSUBJETIVIDADE NO DESENVOLVIMENTO DE SOFTWARES**

As marcas da intersubjetividade do desenvolvedor nos softwares aparecem mais claramente durante o desenvolvimento das interfaces. A marca ideológica de um eu que se vincula a um lugar social encontra nas interfaces sua maior ressonância. Daí o motivo de alguns olharem as interfaces dos softwares e enxergarem nelas enunciações de outrem, por exemplo nos chats temos discursos para guiar ações de uso entredesenvolvedores e usuários.<<nesses discursos o uso das letras, formatos, cores etc aparecem as marcas da intersubjetividade.>>

É mais especificamente no trabalho da equipe que irá desenvolver a interface do software – Graphical User Interface (GUI) –, que as questões relacionadas ao estilo, a intersubjetividades, as posições valorativas, a alteridade, se tornam mais expressivas. Esses profissionais de design têm por missão comunicar, ao usuário do software, o que ele poderá fazer com um software. A pressuposição de um outro a quem o software irá servir interfere significativamente no acabamento do produto. Dessa forma, questões como: a quem a mensagem se destina? Que problemas a interface se propõe a resolver? Como interagir com sistemas para resolvê-los? São pensadas pelos designers enquanto estão desenvolvendo as interfaces dos sistemas. A ideia de comunicar algo a alguém e fazer com que esse outro, em potencial, assuma uma posição de destaque frente a um chamado que se manifesta em um artefato computacional, constitui um dos trabalhos mais suscetível a interferência de terceiros dentro da fábrica de software.

O trabalho da equipe de design é um trabalho que é constantemente alvo de juízos de valor – estéticos e ideológicos –, por parte de terceiros, como os gerentes de produto ou mesmo os clientes. Essas avaliações ou coerções externas não estão diretamente ligadas ao domínio da técnica. O saber-fazer, por parte dos protagonistas [designer e desenvolvedor], que qualquer atividade evoca, mas que se situa no plano do estilo e da instância ideológica, requer, do protagonista da atividade – o designer –, a capacidade de se adaptar às coerções sociais impostas pelo ambiente. Integra, dessa forma, a coesão e o conflito de opiniões na situação de trabalho.

Assim, o grupo social – os companheiros da jornada –, passam a fazer parte da atividade do profissional que projeta a interface. Essas vozes, essas posições avaliativas, que ao mesmo tempo integram e reorganizam o processo de acabamento de um artefato tecnológico, refletirá no próprio artefato em desenvolvimento, uma vez que o trabalho da equipe é moldado pelas diversas sugestões [de como fazer, o que fazer, o que colocar no artefato tecnológico] que surgem durante o desenvolvimento.

A interface – que vista por um prisma dialógico –, é um enunciado dirigido a alguém, carrega em si muito da intersubjetividade de quem a cria. Veja, por exemplo, a figura 05, quando a interface do software Firefox envia uma enunciação para o sujeito utilizador

informando o motivo pelo qual não conseguiu restaurar algumas abas e pede para que este tome uma atitude responsiva, escolhendo entre restaurar ou iniciar nova sessão.

A interface, assim, está no nível enunciativo enquanto diálogo reflexivo que o designer (ou desenvolvedor do software) estabelece com o usuário do *software*. Esse diálogo é materializado através de signos – e.g. o botão restaurar, iniciar nova sessão, ou ainda o botão colar, copiar, os menus arquivos e ajuda vistos em um *software* como o MS Word –, e também por todos os enunciados que se acoplam a uma interface para guiar ações de uso como chats.

Essas formas de enunciações, que se materializam em uma interface de *software*, constituem-se como material de base sobre o qual o usuário do *software* estabelecerá relação.

Os enunciados dos desenvolvedores, como tomamos as interfaces, têm por objetivo mobilizar o usuário do *software*, de orientar suas ações. Essa orientação primária – em primeiro momento facilita o uso de um *software* –, se intercambia a um só tempo com a orientação para o outro – a comunicação mediada entre dois sujeitos sociais por meio de *softwares* dão origem aos gêneros digitais, como o bate-papo, por exemplo.

## REFERÊNCIAS BIBLIOGRÁFICAS

BAKHTIN, M. *Problemas da poética de Dostoiévski*. Forense Universitária. Rio de Janeiro, 1981.

DE SOUZA, C.S. “The Semiotic Engineering of User Interface Languages”. *International Journal of Man-Machine Studies* 39. Academic Press. pp. 753-773. 1993.

DE SOUZA, C.S. *The Semiotic Engineering of Human-Computer Interaction*. Cambridge, Mass. The MIT Press. 307 p. 2005.

DE SOUZA, C. S., Leite, J. C., Prates, R. O., Barbosa, S. D. J. Interação Humano-Computador: Perspectivas Cognitivas e Semióticas In: *Anais das Jornadas de Atualização em Informática I* ed. Rio de Janeiro : Edições Entre Lugar, 1999.

JOHNSON, Steven. *Cultura Da Interface: como o computador transforma nossa maneira de criar e comunicar*. Jorge Zahar Editor. Rio de Janeiro, 1ª Edição – 2001.

NORMAN, D. “Cognitive Artifacts”. In Carroll (ed.) *Designing Interaction: Psychology at the Human-Computer Interface*, 1991.

PEIRCE, Charles Sanders. *Semiótica*. Trad. José Teixeira Coelho Neto. 3. ed. São Paulo: Perspectiva, 2003. Tradução de: *The Collected Papers of Charles Sanders Peirce*. Charles Hartsforne e Paul Weiss, Org.1939-1914.

PERES, Flávia. *Diálogo e autoria: do desenvolvimento ao uso de sistemas de informação*. 204 f, Tese de Doutorado, Curso de Pós-graduação em Psicologia Cognitiva, Universidade Federal de Pernambuco. 2007.

PERES, Flávia e Meira, Luciano. O diálogo como metáfora dos processos de desenvolvimento de software. In.: *Anais do VII Simpósio sobre fatores humanos em sistemas computacionais*: Natal, RN. 2006.

SOUZA A.G. O software, lugar de inscrição da escrita em ambiente digital. In. *Anais do VI Congresso internacional da ABRALIN*. UFPB João Pessoa, Paraíba. 2009.