A INTERFACE GRÁFICA COMO LUGAR DE CONSTITUIÇÃO SUBJETIVA

Benedito Fernando Pereira*

Resumo:
Criadas a partir de uma concepção matemática e lógica de linguagem, as linguagens de programação constituem a ferramenta pela qual o universo informático é construído. Tidas como unívocas e perfeitas, com elas se escrevem os códigos fontes e se desenham as interfaces gráficas dos programas massivamente usados atualmente. As linguagens de programação, contudo, não deixam de ser linguagens que se abrem aos múltiplos sentidos e determinam posições sujeitos. Neste trabalho, tendo como base o referencial teórico da Análise de Discurso francesa, entendemos as interfaces gráficas como objetos construídos pelas linguagens de programação e, como tais, objetos de linguagem que funcionam como um lugar de constituição subjetiva e de deriva dos sentidos a partir da análise de uma interface tomada como corpus.

Palavras-chave: linguagens de programação; constituição subjetiva; interfaces e sentido.

Abstract:
The programming languages have been developed taking into account a concept of logical and mathematical language which could read the reality and reproduce it in electronic systems. So, these languages constitute the tool by which the cyberspace is built. Taken as unequivocal and perfect, they are used to write source codes and to design the GUI of massively currently used programs. However, the programming languages are open to multiple meanings as well as other languages, and they even determine subject positions. In this paper, based on the theoretical framework of French Discourse Analysis, we understand graphical interfaces as objects constructed by the programming languages and, as such, objects of language that function as a place of subjective constitution and of drift of the senses. For that, we take a software interface as a corpus for our analysis.

Keywords: programming languages; constitution of the subject; interfaces and meaning.

* Licenciado em Letras (Univás), Bacharel em Filosofia (FACAPA), Especialista em Ensino de Filosofia (FACAPA), Mestre em Ciências da Linguagem (Univás). Contato: bferpereira2@hotmail.com.

Introdução

Tendo como origem a concepção mecanicista da natureza, entendemos aqui as linguagens de programação como concretização de um discurso sobre a língua que materializa a ideologia da língua lógica e semanticamente perfeita e estável. Essa linguagem seria usada para descrever o mundo e reconstruí-lo no mundo comunicacional proposto pela Cibernética de Norbert Wiener.

Mas, as linguagens de programação, embora tenham sido elaboradas como línguas lógicas segundo o imaginário de clareza da linguagem e materializem a ideologia cibernética do controle, são também formas de significar e, portanto, passíveis de interpretação. Em outras palavras, são também opacas e incompletas, mas provocam o efeito de completude pela rigidez lógico-matemática. Assim, tomando como *corpus* de análise a interface gráfica de um software de cálculo de médias, neste trabalho buscamos entender o seu funcionamento como objeto de linguagem que se abre à interpretação e à deriva dos sentidos. Também nessas linguagens e nos objetos por elas construídos estão presentes os gestos de interpretação, como em todo objeto de linguagem. São, pois, lugares de constituição do sujeito.

1. Linguagens de programação e sentidos

Tendo como um dos seus pilares ideológicos o projeto informacional proposto pela Cibernética formulada pelo matemático Norbert Wiener, áreas como a Inteligência Artificial e a Robótica começaram a se desenvolver em meados do século XX com o intuito de criar máquinas inteligentes. Foi o início do que se passou a chamar de Tecnologias de Comunicação e Informação (TICs), destinadas a provocar profundas transformações nos modos de vida contemporâneos em pouco tempo.

A Informática foi uma das áreas que mais se desenvolveu desde então, e já em 1946, o projeto de computadores potentes tornou-se realidade com o lançamento do ENIAC, primeiro computador eletrônico (VELLOSO, 1999). O computador é uma máquina física (hardware) que opera por meio de sistemas operacionais e aplicativos nele instalados, denominados softwares. Softwares “são programas preparados pelo fabricante do computador (alguns) e pela equipe que o utiliza diretamente (outros), que permitem a obtenção de resultados buscados” (VELLOSO, 1999, p. 59). Um programa é um conjunto de instruções que dizem à máquina o que fazer para

alcançar tais resultados. Estas instruções são textos escritos em uma linguagem específica, chamada linguagem de programação e que, uma vez em funcionamento, realizam tarefas nas e por meio das máquinas, criam e possibilitam a manipulação de arquivos de qualquer tipo (textos, imagens, áudios, vídeos etc) e, enfim, criam o ciberespaço. Um software é, então, um objeto de linguagem e, como tal, simbólico. Desse modo, a realidade virtual, o ciberespaço e todo o universo das TICs são em sua constituição, basicamente, construções textuais. Uma linguagem de programação é "um conjunto de termos (vocabulário) e de regras (sintaxe) que permite a formulação de instruções a um computador" (VELLOSO, 1999, p. 60), ou seja, é uma linguagem logicamente estruturada, escrita na forma de algoritmos estruturados que devem ser lidos e interpretados pela máquina.

Do nosso ponto de vista, essas linguagens são uma forma particular de dar sentido e de significar o sujeito e o mundo, partindo de uma leitura lógico-matemática. Se a matemática em geral é uma linguagem que procura descrever o mundo, os algoritmos são uma subárea dela que, mais do que descrever, procura interpretar logicamente eventos e processos naturais de modo a fundamentar a construção de projetos de sistemas executáveis pelas máquinas que os simulem e/ou interfiram neles.

> O algoritmo constitui uma codificação do raciocínio necessário à resolução [de um] problema. É por meio desta capacidade de captar e transferir inteligência mediante os algoritmos que são construídas máquinas com comportamento inteligente. [...] Uma vez descoberto um algoritmo para solucionar um problema, o passo seguinte consiste em representá-lo de forma apropriada para que seja transmitido para alguma máquina, ou para que seja lido por outros seres humanos. Isto significa que se torna necessário transformar o algoritmo conceitual em um conjunto facilmente compreensível de comandos que representem, sem ambiguidade, essas instruções (BROOKSHEAR, 2000, p. 16-19).

Esse procedimento, como se vê, exige o emprego de uma 'linguagem perfeita' e teoricamente isenta de ambiguidade, capaz de dizer tudo e tudo resolver, expressão unívoca das relações entre coisas concretas do mundo físico. Trata-se de um discurso sobre a linguagem e da eleição da lógica matemática como única linguagem capaz de ler o mundo corretamente e expressá-lo, transformando-o em informação e comunicação. É um discurso fundado no imaginário de que pela lógica tem-se acesso ao real. As linguagens de programação encarnam esse ideal de linguagem formal projetada matematicamente com base no imaginário de precisão e clareza, a

evidência do sentido unívoco e evidente. Ideal, aliás, perseguido há séculos, como colocado por (ORLANDI, 2013, p. 9):

> [...] sobre a linguagem perfeita, a que não falha e que asseguraria as certezas do cálculo (aritmético), podemos ir, por exemplo, ao século XVII e a Leibniz e aos filósofos, lógicos e matemáticos e, particularmente, a Frege (1882), com a proposta da primeira linguagem formal. Não nos esqueçamos das várias tentativas de matematização, mesmo bem atuais, das Ciências Humanas: eis um esforço de evitar o corpo das palavras, suas ambiguidades, equívocos e contradições. Evitar a materialidade dos gestos de interpretação, a historicidade que aí se inscreve, e as teorias que sustentam as disciplinas de interpretação. A busca do exato, não só na relação linguagem-pensamento-mundo, mas também na do sujeito-sentido. E juntos vêm o cálculo, a precisão. A máquina e/ou o programa.

Contudo, é uma linguagem que lê a realidade e a reconstrói nos textos dos códigos num processo interpretativo, sendo, portanto, uma forma de dar sentido a ela. Se considerarmos com Orlandi (2007) que os processos de significação são determinados historicamente, podemos dizer que todo ato de leitura demanda gestos de interpretação e não se faz de forma isenta: o sujeito – que é assujeitado à língua – produz sentidos ao interpretar, e o faz de um lugar, de uma condição social e histórica determinada. Quando o sujeito lê o mundo, estabelece uma relação de compreensão com ele, de natureza sócio-histórica, que determina a produção de sentidos (ORLANDI, 2007). Mas nem o real da história, nem o da linguagem é acessível e transparente ao sujeito (*idem*, *ibidem*), de modo que a naturalidade dos sentidos é aparente e fruto dos esquecimentos de que o que se diz já foi dito antes e também de que o sentido sempre pode ser outro (PÊCHEUX, 1975). Esses esquecimentos produzem a impressão da realidade do pensamento (ORLANDI, 2007, p. 35) e, acreditamos, influem na (ou até determinam a) própria estruturação lógica deste. Ora, um código-fonte é um texto escrito em linguagem de programação que faz sentido para o sujeito e para a máquina: isso garante o seu funcionamento, não só do ponto de vista técnico (executar tarefas, resolver problemas), mas enquanto texto, enquanto discurso. Ele marca posições discursivas (a do programador, a do cientista da computação, a do usuário) e se insere no repetível, no já-dito, condição básica de significação e de perpetuidade. Estabelece-se, então, uma relação com a memória, necessária para que a ideologia se materialize no discurso e este nas formulações possíveis de linguagem (ORLANDI, 2007). Aí os objetos do discurso adquirem sua estabilidade referencial. Desse modo, o repetível é uma sistematicidade não abstrata,

mas real e histórica. Assim, uma linha de código é um lugar do dizer, é um enunciado; ele faz parte do interdiscurso (do repetível), o qual está no intradiscurso (uma determinada sequência linguística).

Desse modo, as linguagens de programação surgem como representantes máximas da tentativa de controle (e mesmo, de eliminação) da deriva dos sentidos na leitura do mundo e do sujeito e, assim, produzem um discurso sobre a linguagem e sobre o próprio sujeito e o mundo. Tentativa de controle da entropia na e pela linguagem. Mas elas caem – como qualquer outra linguagem – na ilusão referencial que "nos faz acreditar que há uma relação direta entre o pensamento, a linguagem e o mundo, de tal modo que pensamos que o que dizemos só pode ser dito com aquelas palavras e não outras, que só pode ser assim" (ORLANDI, 2007, p. 35). Então, se as linguagens de programação fazem sentido, estes também são construídos, e estão permeados pelas relações de poder presentes na sociedade e que se materializam na linguagem, com seus imaginários, produzindo discursos e interpelando sujeitos. E é nesse processo que se constituem a posição-sujeito do programador e a do usuário.

Perseguindo o ideal de língua lógica e fechada, cristalizada na evidência do dizer, o programador procura regular tanto quanto possível as ações do usuário do programa, ações essas que também são linguagem na medida em que dependem da relação dialógica entre ele e a interface gráfica do programa (dependem da leitura e interpretação dos dados exibidos na tela, de caixas de diálogo etc.) para a qual o usuário também deve dar respostas. A regulagem se dá por meio de tratamentos de erros, via código, que limitam as ações dos usuários, direcionando em certa medida a sua apropriação do programa, e caracterizam, do lado do programador, um movimento de "dizer x para não (deixar) dizer y" (NUNES, 2013, p. 117). Mas, como qualquer outra linguagem, a escrita do texto do código é um ritual também sujeito a falhas (PÊCHEUX, 1990), e essas falhas são interpretadas como erro lógico, segundo a formação discursiva que constitui o sujeito programador. O "erro" é, também neste caso, o lugar do inesperado, da criatividade, do sentido outro, do deslocamento, da entropia, enfim, do virtual no sentido específico de tendência à mudança e que mostra a deriva do sentido também nessa forma de linguagem.

Considerando "o eletrônico enquanto processo discursivo a partir do qual sentidos são produzidos" (DIAS, 2011, p. 21) e a materialidade da linguagem como o "processo de significação a partir do qual um discurso se textualiza numa forma e não em outra" (*idem*, *ibidem*), podemos dizer que a forma como o sujeito programador constrói os algoritmos e os códigos a partir deles, ou seja, o modo como textualiza em

linguagem de programação a sua leitura do mundo, é efeito dos processos de significação que se inserem na história e ganham sentido nessa materialidade do eletrônico.

No meio social a discursividade do eletrônico ganha outros sentidos. Se a linguagem lógica da programação lê o mundo e o reconstrói digitalmente de um lado da tela, do outro lado está o sujeito usuário, que vai se apropriar dos programas como elementos de linguagem e atribuir-lhes sentidos, constituindo-se juntamente com eles e com os discursos que o interpelam. O seu funcionamento será outro, para além do pretendido controle tão caro à ideologia cibernética que embasa as TICs.

E entre o sujeito programador e o sujeito usuário está a interface dos softwares, aqui tomados como objetos de linguagem que fazem a intermediação entre o sujeito e o mundo digital e como dispositivos de constituição do sujeito.

2. As interfaces e o sujeito

Segundo Caiçara Jr. e Paris (2007, p. 66), a interface é o "elemento responsável pela comunicação entre o usuário e o computador". Ela diz respeito tanto ao aspecto físico da máquina (hardware) quanto aos softwares: a interface de um programa é o que efetivamente aparece na tela do usuário, as janelas com suas caixas para digitação de texto, botões e demais ferramentas para entrada e saída de dados. E não se trata de interface de computadores e softwares apenas, mas das TICs de modo geral, de qualquer equipamento que faça uso de tecnologia eletrônica.

As linguagens lógicas e rígidas de programação, embora estejam imbuídas do imaginário de univocidade e clareza do sentido, não são, contudo, de fácil compreensão, sobretudo, para o sujeito leigo dos seus processos de escrita e funcionamento. De fato, por muito tempo, os programas de computador foram projetados tendo em vista apenas a resolução de problemas práticos por meio da transmissão e processamento de informação, e seriam utilizados por pessoal especializado no assunto, e isso não supunha a necessidade de as interfaces serem "amigáveis" aos usuários:

> Dedicar tempo e dinheiro à interface com o usuário era considerado um frívolo desperdício, pois, sendo os ciclos do computador tão preciosos, eles tinham de ser empregados no problema, e não na pessoa. [...] Aqueles dentre nós que trabalhavam no desenvolvimento da interface homem-computador ao final da década de 1960 e ao

> longo da de 1970 eram considerados os maricas da computação e vistos com franco desprezo. [...] A GUI [sigla em inglês para Graphical User Interface – Interface gráfica do usuário] desenvolveu-se muito desde que surgiu por volta de 1971 com um trabalho da Xerox, prosseguindo pouco depois no MIT e em alguns outros lugares e culminando uma década mais tarde num produto de verdade, quando Steve Jobs teve a sabedoria e a perseverança necessárias para criar o Macintosh (NEGROPONTE, 1996, p. 90).

Foi a partir daí e, posteriormente, com o lançamento do Windows pela Microsoft, acompanhado pelo barateamento das máquinas, que os computadores pessoais se popularizaram e a preocupação com a interface do usuário tornou-se uma questão levada a sério. Popularizar as TICs, em especial os computadores e softwares, significa também tornar sua apropriação mais fácil, rápida e eficaz pelo sujeito usuário. Desse modo, desenvolver interfaces é mais do que um problema estético ou funcional, como afirma o autor:

> [...] A interface dos computadores pessoais tem sido tratada como um problema de desenho físico. Contudo, ela não diz respeito apenas à aparência e ao manuseio do computador [e programas]. Trata-se, na verdade, da criação de uma personalidade, do *design* da inteligência e da construção de máquinas capazes de reconhecer a expressão humana. [...] O segredo do projeto de uma interface [reside em] fazê-la desaparecer. Quando somos apresentados a alguém, podemos prestar grande atenção em sua aparência, em suas palavras e em seus gestos. Logo, porém, o conteúdo da comunicação passa a predominar, ainda que ele seja expresso pelo tom de voz ou com auxílio da linguagem das expressões faciais. Uma boa interface de computador deveria comportar-se de modo semelhante. Trata-se menos de desenhar um painel de instrumentos do que de desenhar um ser humano (NEGROPONTE, 1996, p. 91-93).

As interfaces são construções de linguagem que podem ser elaboradas via código ou utilizando ferramentas disponíveis no ambiente de programação. Em todo caso, são elementos gráficos pelos quais o sujeito interage com a máquina e também com outros usuários por meio de textos e de imagens digitais. Conforme o autor, uma boa interface deve "desaparecer", ou seja, deve ser elaborada de maneira tal que ofereça ao usuário um ambiente ao mesmo tempo funcional e "intuitivo" de operação, de modo que a apropriação desse dispositivo se faça o mais integral e rapidamente possível, tendo-se em perspectiva um sujeito usuário universal. Assim, uma caixa de diálogo como o *Inbox* da rede social *Facebook*, ou uma sala de bate-papo, ou

qualquer tela de programa, são interfaces de usuário que, para fazer sentido e cumprir bem a sua função, devem ser "transparentes" ao usuário (ilusão de transparência da linguagem), de modo a não lembrar ao sujeito a existência de toda a complexidade técnica, tanto eletrônica quanto de programação existente por trás daquele objeto e, assim, dar-lhe a sensação de falta de limite e de instantaneidade de acesso e comunicação, bem como de objetividade da linguagem. Aí está tecnicamente colocado o princípio ideológico da comunicação sem fronteiras conforme proposto pelos cibernéticos, a informatividade. E esse princípio é a base do que se chama em informática de *design* universal:

> O design universal é o processo de criar produtos, comercialmente viáveis, que possam ser usados por pessoas com as mais variadas habilidades, operando em situações (ambiente, condições e circunstâncias), as mais amplas possíveis. Alguns princípios que suportam o design universal: uso equitativo, flexibilidade no uso, uso simples e intuitivo, informação perceptível, tolerância a falhas, baixo esforço físico, tamanho e espaço para aproximação e uso (DIAS, 2007, p. 103).

Ser "intuitivo" aqui aparece como algo dado, natural; não se considera que o intuitivo também é algo construído historicamente no sujeito: algo é intuitivo porque aciona mecanismos inconscientes do sujeito, além de suas memórias (que são históricas) e, por isso faz sentido para ele. Um sujeito cuja formação se processa pela ideologia e pelo inconsciente (ORLANDI, 2012). Desse modo, o sujeito programador deve escrever o código fonte dos seus programas pensando, não só na sua interpretação pela máquina, mas também levando em consideração a implementação dele numa interface que atenda a esses requisitos de forma satisfatória.

O que nos interessa destacar é que estas especificações só têm sentido em função da imagem que o sujeito programador faz do sujeito usuário. E "não há sentido que não tenha sido produzido em condições específicas, em uma relação com a exterioridade, com uma direção histórico-social que se produz em relações imaginárias que derivam de um trabalho simbólico" (ORLANDI, 1998, p. 75). O que chamamos de sujeito usuário e sujeito programador são posições, "lugares de significação historicamente constituídos" (*idem*, *ibidem*) que surgem como efeito dos processos discursivos em questão e correspondem às imagens que esses sujeitos fazem de si e dos outros (PÊCHEUX, 1969, p. 82). Assim, qualquer software (ou *site* da internet) é um objeto de linguagem, não só porque é um texto escrito em uma linguagem de programação, mas porque também apresenta uma interface de usuário

que por si mesma já é um objeto simbólico de espessura semântica. Ela será lida e interpretada por um sujeito usuário, o qual pode ter conhecimentos técnicos ou não, mas que, de qualquer forma, deve se apropriar dela e passar a (se) significar por seu intermédio. E é através das interfaces que todas as interações[1] no ambiente digital ocorrem. São formas de escrita que influenciam nos modos como o sujeito se constitui, conforme coloca Dias (2009, p. 12) ao afirmar que "as formas da escrita contemporânea que utilizam a tecnologia digital, especificamente a escrita no computador, colocam o sujeito em relação com distintos imaginários".

E, assim como no caso da escrita do código fonte, a elaboração da interface parte do princípio da objetividade e clareza da linguagem, da univocidade do sentido:

> Um dos principais requisitos para o desenvolvimento de produtos, considerando a perspectiva do design universal, é garantir que todas as informações apresentadas possam ser percebidas, mesmo sem a visão, audição, habilidade normal de leitura e aprendizado, percepção de cores, e sem causar distúrbios mentais. [...] Apresentar textos em forma oral ou compatível com dispositivos que transformem o texto escrito em texto falado [...]; utilizar linguagem simples, compatível com o tipo de produto, público e situação de uso, e ainda facilitar a identificação de termos em outros idiomas (para que sejam traduzidos por ferramentas auxiliares) são boas medidas para facilitar a compreensão geral do produto. [...] Outros fatores [também] são importantes para um bom *design*, tais como estética, custo, segurança, adequação cultural e de gênero (DIAS, 2007, p. 106-109).

A tela apresentada na figura a seguir foi criada na linguagem de programação Visual Basic e corresponde a uma interface gráfica para servir de diálogo com o sujeito usuário. A tela mostrada na imagem abaixo é a interface gráfica de um programa concebido para calcular as médias dos alunos em uma turma:

[1] O termo "interação" é frequentemente usado para se referir às relações que se estabeleceram, sobretudo, após o surgimento da Web 2.0, que tornou possível a saída do sujeito usuário de um estado de passividade frente ao conteúdo da rede ao dar-lhe a possibilidade de acrescentar, remover ou alterar dados/conteúdo da rede e de se relacionar com outros usuários de modo dinâmico por meio dela, com vídeos e áudio, além dos textos. Conferir em: <http://pt.wikipedia.org/wiki/Web_2.0>.

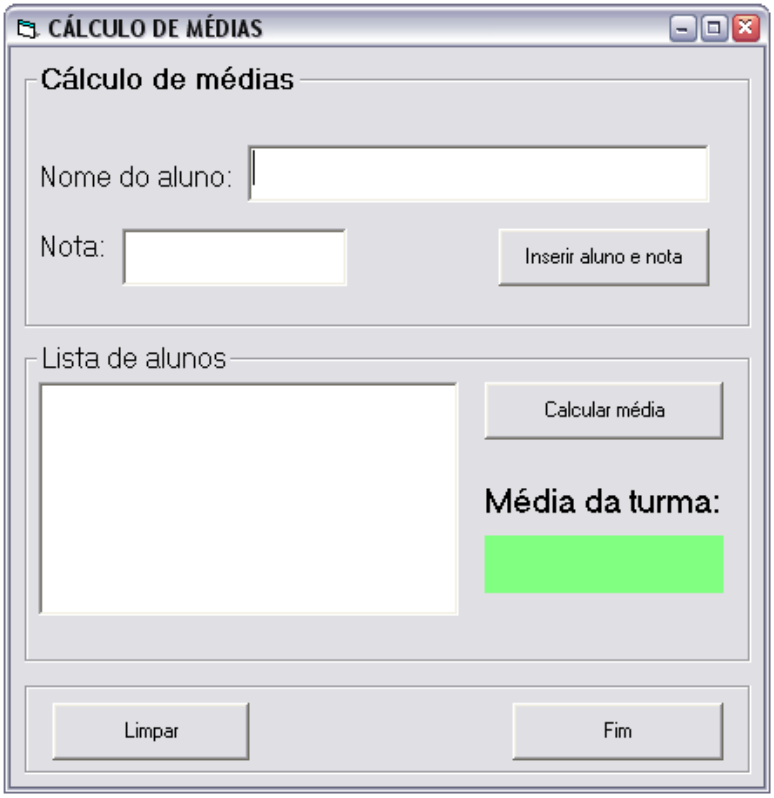

Figura 1: Interface de usuário do programa "Cálculo de Médias" desenvolvida em Visual Basic.

O sujeito usuário que interage com essa janela digitando os nomes dos alunos e as respectivas notas e, posteriormente, apenas clica no botão que apresenta o rótulo "Calcular média" para conhecer o resultado dos cálculos efetuados pelo programa, normalmente não tem idéia dos processos internos que ocorrem por trás dessa tela e que possibilitam o funcionamento do software. De um lado da tela está o sujeito usuário, de outro, o texto escrito pelo programador: a interação ocorre, então, não entre sujeitos diretamente, mas entre sujeito e texto. No caso de uma conversação on-line entre duas pessoas, como numa sala de bate-papo, a interação se dá entre sujeitos usuários por meio de outros dois textos: o texto do código fonte da página ou programa e o texto da interface. Estes, porém, desaparecem ao olhar do sujeito, que se apropria deles e os incorpora à própria subjetividade ao dar-lhes sentido e se significarem por seu meio. Eles fazem parte da forma como o sujeito (se) textualiza nas condições de produção do digital, uma textualização proporcionada por uma textualização anterior: aquela do código escrito em linguagem de programação que subjaz à interface. Processos de escrita e de constituição de subjetividades. Para Orlandi (1988), a noção de sujeito é indissociável da linguagem como trabalho constitutivo: é no espaço discursivo que se constituem os sentidos e o sujeito em sua incompletude. Linguagem e sujeito são necessariamente incompletos e

> é na escritura que o sujeito habita seu nome próprio ou sua identidade, e através da escritura o sujeito coincide consigo mesmo e encontra o fantasma da unidade. Porque o ato de escrever é a tentativa de suturar a perda, embora seja movido por essa impossibilidade que o sujeito escreva. É aí que Pêcheux põe em relação o real da língua com a história e aponta para o fato de que é possível contornar o impossível. [...] Com isso, a ciência linguística tem na língua universal, lógico-matemática, a esperança de reconstruir uma língua ideal. E não só linguistas, mas também filósofos como Descartes e Leibniz perseguiram esse ideal (DIAS, 2004, p. 39-40).

Escritura do código fonte e construção da interface de um lado (sujeito programador) e interação com a interface de outro (sujeito usuário), a qual ocorre num processo interpretativo, de leitura e escrita. Assim, trata-se de uma linguagem em funcionamento e, portanto, de discursos, de ideologias, de visões e representações de mundo que são acionados pelo que Dias (2009) chama de gesto do clique, o qual, entre outras coisas, instaura novas formas de se relacionar socialmente e com os saberes. O ato de programar, de um lado, e o ato de usar um programa, de outro, são práticas de linguagem distintas, porém, correlacionadas e interdependentes. A figura a seguir mostra a mesma interface em funcionamento:

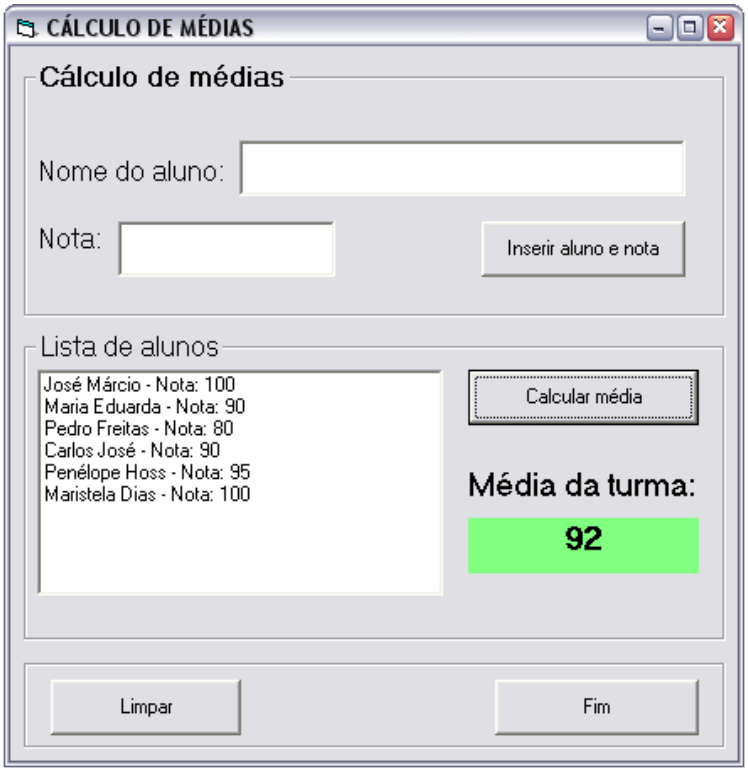

Figura 2: Programa "Cálculo de Médias" em funcionamento.

Quando o sujeito usuário clica nos botões “Inserir aluno e nota” e “Calcular média”, sem que ele se dê conta, ele está, na verdade, trabalhando com o código escrito para esses dois botões específicos. Cada clique que se dá em um botão, o texto correspondente a ele é relido e interpretado pela máquina. O sujeito usuário lê a interface, interpreta-a, reage a ela com o gesto de clicar; a máquina responde ao clique lendo e interpretando o código conforme os dados inseridos pelo usuário e responde a este conforme esses dados: instaura-se um diálogo em que leituras, interpretações, ações e respostas a ações ocorrem. A figura abaixo mostra a interface de usuário com os códigos para cada botão presente nele:

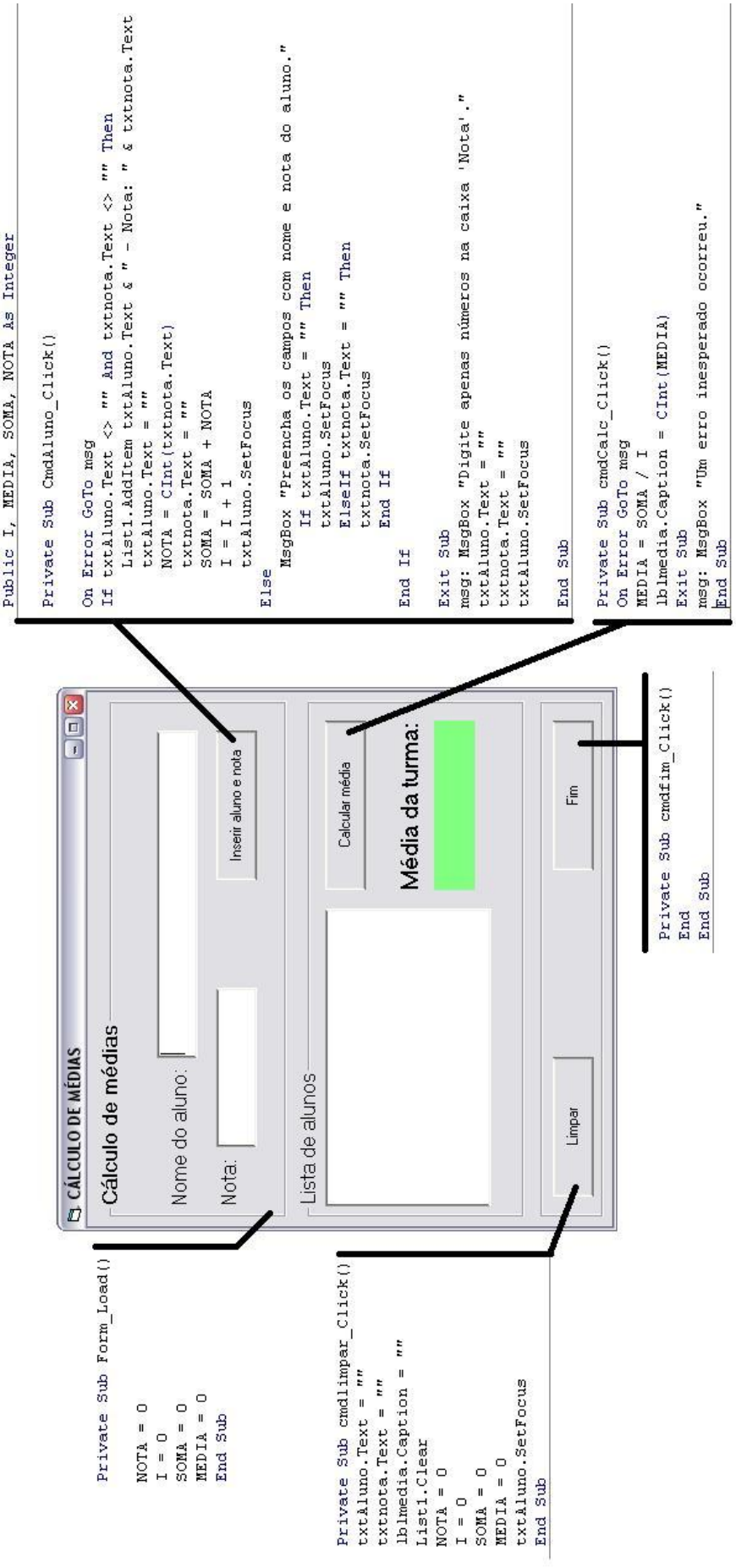

Figura 3: interface de usuário e códigos fontes dos botões que a compõem.

Note-se que é necessário que se coloquem tratamentos de erros nos códigos de botões que realizam operações de entrada de dados e cálculos como antecipação às possíveis ações (equivocadas) do usuário. As linhas de tratamento de erro são as seguintes:

On Error go to msg
(...)
msg: Msgbox “Digite apenas números na caixa 'Nota'.”
txtAluno.Text = “”
txtnota.Text = “”
txtAluno.SetFocus

Que, se acionadas em caso de o usuário digitar letras no lugar de números no campo “Nota” na interface, geram a seguinte caixa de mensagem:

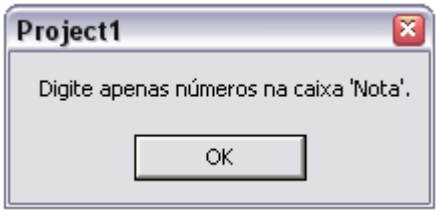

Figura 4: mensagem de erro.

O código diz à máquina que se um erro ocorrer, o programa deve exibir ao usuário a mensagem “Digite apenas números na caixa ‘Nota’.”, e em seguida, os dados digitados devem ser apagados. O sujeito usuário não vê o código, mas apenas a mensagem que o programador deixou ali por antecipação. Isso significa que, da perspectiva do programador (e da linguagem lógica de programação), há interpretações “erradas” possíveis do texto ali colocado. Ora, essa interface de usuário é um objeto de linguagem que deve ser lido e interpretado por um sujeito usuário para que faça sentido para ele. Mas o sentido nem sempre é o esperado pelo sujeito programador em função do código que escreveu.

Desse modo, a interface pode ser tomada em duas perspectivas: a do programador e a do usuário. Pelo viés do programador, ela é um texto cujo objetivo é tornar acessível, atraente e manipulável de forma intuitiva o código do programa pelo sujeito usuário. Ela deve ser elaborada de maneira a moldar-se o mais possível a ele para alcançar esse fim (deve “desaparecer”, como vimos). Pelo viés do usuário, a interface é uma construção que diz e permite dizer algo e é, assim, um lugar de constituição subjetiva. A interface é um objeto de linguagem e, como tal, opaco, de espessura semântica frente ao qual o sujeito deve interpretar sempre (ORLANDI,

1999). Por isso ela significa de diferentes formas, e parte de formações imaginárias já postas, como vimos. O gesto de interpretação ocorre juntamente com o gesto de leitura em que sentidos são produzidos. Ao se moldar a essa estrutura, o sujeito usuário se conforma à lógica do programa, passando a significar-se e a agir nele e por meio dele. O apagamento da interface é fundamental para a imersão do sujeito no ciberespaço e para criar o efeito de "fusão" dele com a máquina. Então, o sujeito usuário tem a sensação de que a interação acontece com um outro sem que entre eles haja algo que se interponha.

Mas a interface é um lugar onde também se dá a erupção dos sentidos, posto que se abre ao inesperado, uma vez que não está imune à entropia natural entendida, na linguagem, como a possibilidade do sentido outro, das resistências que geram deslocamentos e rupturas, a transgressão, o que justifica a necessidade de se colocarem tratamentos de erro que controlem em certa medida o dizer do sujeito que a usa. Controle que vem justificado pelos limites operacionais da máquina e pela estrutura interna do código fonte do programa escrito em uma linguagem lógica, mas que, em todo caso, configura um gesto de dizer x para não (deixar) dizer y: "a significância do contexto é delimitada pelo já-dito que con-forma o conjunto da situação que intervém no dizer. É só o que conta para o sentido "x" (efeito de pré-construído) que faz parte das condições de produção imediatas" (ORLANDI, 1998). O programador cumpre, desse modo, a função-autor que produz um efeito-leitor, como afirma (ORLANDI, 2001, p. 65-66): "[...] se temos, de um lado, a função-autor como unidade de sentido formulado, em função de uma imagem de leitor virtual, temos, de outro, o efeito-leitor como unidade (imaginária) de um sentido lido. [...] O efeito-leitor é uma função do sujeito como a função-autor". Toda essa problemática se coloca, dessa forma, com a relação estabelecida pelas interfaces gráficas. A interface ao ser lida (interpretada) pelo sujeito usuário funciona pela "ilusão de conteúdo" produzida na superfície da linguagem como sentido único, um já-lá literal, de modo que "pelo trabalho da ideologia, o conteúdo se substitui à forma material" (ORLANDI, 1998).

Mas nenhum tratamento de erro é impecável e nenhum programa é imune a falhas e, por essa razão, com frequência ocorrem os chamados *bugs* dos sistemas. O que falha evidencia a incompletude dessa linguagem e coloca em xeque o imaginário de exatidão, clareza e de fechamento que constitui os discursos que circulam sobre ela. Damo-nos conta, então, de que a linguagem lógica também é um ritual com falhas, é língua de bits que se abre ao equívoco. E o que escapa nesse processo, isto é,

o que fica à margem, silenciado como "erro" também reclama sentidos, significa. Como afirma Pêcheux (1990), não há ritual sem falhas, algo sempre escapa.

Teclar e clicar são práticas sócio-históricas que reorganizam textualmente o dizer, demandam gestos de leitura, interpretação e escrita, que mobilizam sentidos e discursos, e ressignificam a máquina, a linguagem, o sujeito e os imaginários envolvidos. Desse modo, são atos políticos historicamente localizáveis e abertos ao devir numa nova relação de tempo-espaço por eles mesmos instituída.

Programas de computador são uma forma peculiar de texto, diferentes dos textos "comuns" que encontramos em livros, por exemplo. Trata-se de um texto que se "esconde" por trás de uma tela, de uma interface de programa. O código que está "por trás" dela, na verdade, só é lido pelo programador no momento da sua escrita e pela máquina, que deve interpretá-lo e executá-lo. Mas a interface, em si mesma, já é um texto será lido pelo usuário e que já será outro texto a partir de cada ato de leitura.

A interface gráfica funciona como um lugar de virtualização (desprendimento de um aqui e agora particular, passagem ao público e heterogênese (LÉVY, 1996)) e atualização do sujeito, porta de entrada para o ciberespaço que entendemos como uma realidade construída pela linguagem. Vejamos o que diz Lévy (1996, p. 39-40):

> O leitor de um livro ou de um artigo no papel se confronta com um objeto físico sobre o qual uma certa versão do texto está integralmente manifesta. Certamente ele pode anotar nas margens, fotocopiar, recortar, colar, proceder a montagens, mas o texto inicial está lá, preto no branco, já realizado integralmente. Na leitura em tela, essa presença extensiva e preliminar à leitura desaparece. O suporte digital (disquete, disco rígido, disco ótico) não contém um texto legível por humanos, mas uma série de códigos informáticos que serão eventualmente traduzidos por um computador em sinais alfabéticos para um dispositivo de apresentação. A tela apresenta-se então como uma pequena janela a partir da qual o leitor explora uma reserva potencial. Potencial e não virtual, pois a entalhe digital e o programa de leitura predeterminam um conjunto de possíveis que, mesmo podendo ser imenso, ainda assim é numericamente finito e logicamente fechado. Aliás, não é tanto a quantidade que distingue o possível do virtual, o essencial está em outro lugar: considerando-se apenas o suporte mecânico (hardware e software), a informática não oferece senão uma combinatória, ainda que infinita, e jamais um campo problemático. O armazenamento em memória digital é uma potencialização, a exibição é uma realização.
> Um hipertexto é uma matriz de textos potenciais, sendo que alguns deles vão se realizar sob o efeito da interação com um usuário. Nenhuma diferença se introduz entre um texto possível da combinatória e um texto real que será lido na tela. A maior parte dos programas são máquinas de exibir (realizar) mensagens (textos,

> imagens, etc) a partir de um dispositivo computacional que determina um universo de possíveis. Esse universo pode ser imenso, ou fazer intervir procedimentos aleatórios, mas ainda assim é inteiramente pré-contido, calculável. Deste modo, seguindo estritamente o vocabulário filosófico, não se deveria falar de imagens virtuais para qualificar as imagens digitais, mas de imagens possíveis sendo exibidas.
> O virtual só eclode com a entrada da subjetividade humana no circuito, quando num mesmo movimento surgem a indeterminação do sentido e a propensão do texto a significar, tensão que uma atualização, ou seja, uma interpretação resolverá na leitura. Uma vez claramente distinguidos esses dois planos, o do par potencial-real e o do par virtual-atual, convém imediatamente sublinhar seu envolvimento recíproco: a digitalização e as novas formas de apresentação do texto só nos interessam porque dão acesso a outras maneiras de ler e de compreender.

É o sujeito que dá sentido ao texto, seja no código fonte, seja interagindo com as interfaces e às TICs de modo geral, uma vez que são objetos criados pela linguagem como tudo que constitui o mundo para o sujeito. Sem ele, não há significado: por isso dizermos que as interfaces são o lugar da virtualização por excelência do sujeito e do texto, e a leitura é um processo de atualização de um dentre os vários sentidos virtualmente possíveis. Lugar de constituição. E isso só ocorre por ser a linguagem incompleta e opaca, como afirma Orlandi (2007), não abarcando jamais a totalidade do real. E ainda:

> A língua significa porque a história intervém, o que resulta em pensar que o sentido é uma relação determinada do sujeito com a história. Assim, o gesto de interpretação é o lugar em que se tem a relação do sujeito com a língua. Essa é a marca da 'subjetivação', o traço da relação da língua com a exterioridade (ORLANDI, 1996, p. 46).

E a interpretação é integrante da relação entre a língua e a história (PÊCHEUX, 2012), uma relação que constitui o sujeito e que escapa ao seu controle (ORLANDI, 1996, p. 47) porque intervêm o inconsciente e a ideologia. Desse modo, a interface não se reduz à tela: "o leitor estabelece uma relação muito mais intensa com um programa de leitura e de navegação que com uma tela" (LÉVY, 1996, p. 42). São as características dos programas usados, suas vias de comunicação com o usurário, que estabelecerão as relações possíveis de interação e de subjetivação no processo de virtualização.

3. Considerações finais

O modo de funcionamento das interfaces gráficas como objetos de linguagem, bem como o seu papel na constituição subjetiva, conforme vimos, leva-nos a entender as linguagens de programação como linguagens estruturadas, mas não completas. Em outras palavras, o imaginário de completude, de fechamento comumente associado à linguagem matemática, é efeito de sentido de um discurso sobre ela.

Assim, a linguagem e suas tecnologias são mais do que artes miméticas do real, ou seja, artes dedicadas à produção de simulacros, imagens das aparências das coisas, mas como própria constituinte do sujeito e das realidades nas quais ele habita.

Referências

BROOKSHEAR, J. G. **Ciência da computação**: uma visão abrangente. Trad. Cheng Mei Lee. 5. Ed. Porto Alegre: Bookman, 2000.

CAIÇARA JÚNIOR, C.; PARIS, W. S. **Informática, Internet e aplicativos**. Curitiba: Ibpex, 2007.

DIAS, Cristiane. **A discursividade da rede (de sentidos)**: a sala de bate-papo hiv. 2004. Tese (doutorado). IEL – Instituto de Estudos da Linguagem, Universidade Estadual de Campinas – UNICAMP. Campinas, SP. 2004. Disponível em: <http://www.bibliotecadigital.unicamp.br/document/?code=vtls000341547&opt=4>. Acesso em: 15/12/14.

_____. **A escrita como tecnologia de linguagem**. Coleção Hiper S@beres. Tecnologias de linguagem e produção do conhecimento. Universidade Federal de Santa Maria, Santa Maria, Vol. II, Dez/2009. Disponível em: <http://w3.ufsm.br/hipersaberes/volumeII/textos_pdf/TXTS_PDF/cristiane_dias.pdf>. Acesso em: 15/01/2015.

_____. **e-Urbano**: a forma material do eletrônico no urbano. In. DIAS, Cristiane. E-urbano: Sentidos do espaço urbano/digital [Online]. 2011. Disponível em: <http://www.labeurb.unicamp.br/livroEurbano/>. Laboratório de Estudos Urbanos – LABEURB/Núcleo de Desenvolvimento da Criatividade – NUDECRI, Universidade Estadual de Campinas – UNICAMP. Acesso em: 04/06/14.

DIAS, Cláudia. **Usabilidade na web**: criando portais mais acessíveis. 2. Ed. Rio de Janeiro: Alta Books, 2007.

LÉVY, P. **O que é o virtual?** 1. ed. São Paulo: Editora 34, 1996.

NEGROPONTE, N. **A vida digital**. Trad. Sérgio Tellaroli. São Paulo: Companhia das Letras, 1996.

NUNES, S. R. Práticas de leitura no infográfico eletrônico: trajetos, tropeços e movimentos. In: DIAS, Cristiane. **Formas de mobilidade no espaço e-urbano**: sentido e materialidade digital [online]. Série e-urbano. Vol. 2, 2013, Disponível em: <http://www.labeurb.unicamp.br/livroEurbano>, Laboratório de estudos Urbanos –

LABEURB/Núcleo de Desenvolvimento da Criatividade – NUDECRI, Universidade Estadual de Campinas – UNICAMP.

ORLANDI, E. P. A materialidade do gesto de interpretação e o discurso eletrônico. In. DIAS, Cristiane. **Formas de mobilidade no espaço e-urbano**: sentido e materialidade digital [online]. Série e-urbano. Vol. 2, 2013. Disponível em: <http://www.labeurb.unicamp.br/livroEurbano/volumeII/arquivos/pdf/eurbanoVol2_EniOrlandi.pdf>. Laboratório de Estudos Urbanos – LABEURB/Núcleo de Desenvolvimento da Criatividade – NUDECRI, Universidade Estadual de Campinas – UNICAMP. Acesso em: 15/12/2014.

_____. A incompletude do sujeito: e quando o outro somos nós? In: ORLANDI, Eni. (et alii). **Sujeito e texto**. São Paulo: EDUC, 1988.

_____. **Análise de Discurso**: princípios & procedimentos. 7. ed. Campinas: Pontes, 2007.

_____. **Contextos epistemológicos da análise do discurso**. Escritos n. 4. Campinas: Labeurb, 1999.

_____. **Discurso e argumentação**: um observatório do político. In: Fórum Linguístico, Fpolis, n. 1, p. 73-81, jul-dez, 1998.

_____. **Discurso e texto**: formulação e circulação dos sentidos. Campinas: Pontes, 2001.

_____. **Discurso em Análise**: Sujeito, Sentido, Ideologia. Campinas: Pontes, 2012.

_____. **Interpretação**: autoria, leitura e efeito do trabalho simbólico. Petrópolis: Vozes, 1996.

PÊCHEUX, M. Análise Automática do Discurso (AAD-69). In: GADET, Françoise; HAK, Tony (Orgs.). **Por uma Análise Automática do Discurso**: uma introdução à obra de Michel Pêcheux. 3. ed. Tradução Bethania S. C. Mariani e outros. Editora da Unicamp, 1997.

_____. **Delimitações, inversões, deslocamentos**. Trad. de José Horta Nunes. Caderno de Estudos Linguísticos, n. 19, Campinas: Unicamp, jul./dez. 1990.

_____. **O discurso**: estrutura ou acontecimento. Trad. Eni P. Orlandi. 6. ed. Campinas: Pontes Editores, 2012.

VELLOSO, F. C. **Informática**: conceitos básicos. 4. Ed. Rio de Janeiro: Campus, 1999.