“SOMOS” = “TEMOS”

Fernando Severo
Henrique Cukierman
Isabel Cafezeiro
Ivan da Costa Marques
Rodrigo Primo

Consta que houve entre os dois grandes romancistas norte-americanos a seguinte troca de frases:

Fitzgerald: – Os ricos são diferentes de nós.

Hemingway: – É. Eles têm mais dinheiro.

Ao separarmos os “somos” dos “temos” podemos inadvertidamente fazer desta distinção uma diferença de natureza do tipo que Fitzgerald sugere e Hemingway ironiza no fragmento de diálogo acima. Qual a diferença entre “somos” e “temos”? Até que ponto, onde, como, por quem e por quê, para quem e para quê estas duas figuras ou conceitos são separáveis? Nosso hábito é enxergar as fronteiras entre os “somos” e os “temos” como naturalmente bem definidas. Abrimos nesta mesa um espaço para a problematização dessas fronteiras. Esperamos apontar direções onde se podem abrir novos espaços de possibilidades de análises e de ações (políticas). Os “somos” atuam cognitivamente, e nos fazem ver-nos como se fôssemos unidades, individuais ou coletivas, com fronteiras bem definidas em suas especificidades. Os “somos” também atuam normativamente ao realçarem algumas classificações – assim como Fitzgerald sugeriu que os ricos são “gente de outra espécie”, quando sugerem classificarmo-nos como mestiços, brasileiros, cristãos, de um lado, e brancos ou negros puros, estrangeiros ou gringos, muçulmanos, budistas, judeus ou ateus de outro. Hemingway, no entanto, concorda que os ricos são diferentes dos demais, mas ironicamente traz um circunstancialista “temos” para borrar as fronteiras de um essencialista “somos”.

Não queremos dizer que “somos” e “temos” sejam categorias inúteis ou equivocadas. Queremos problematizá-las, ressaltando que sua utilização proporciona rendimentos diversos se mudam as circunstâncias, os pontos de vista, os interesses, os hábitos, os materiais e as competências. De fato, podemos considerar o “temos” de uma unidade individual ou coletiva, (uma pessoa, uma família, uma universidade, uma comunidade, um país) como uma construção heterogênea socialmente compactuada que atua no sentido de controlar, isto é, de trazer estabilidade e facilitar o constante ordenar que,

sempre provisionalmente, a cada dia, conforma esta unidade. É por essa abordagem que "somos" as escolas, os hospitais, os transportes, as comunicações, as inteligências, os bancos de dados, as polícias, as prisões e as urnas eletrônicas que "temos" no Brasil. Nossos "temos" = nossos "somos".

Um "temos" é indissociável dos pactos ou dos vínculos entre as entidades das redes que o sustentam. Seu potencial de ordenamento depende das redes onde ele está compactuado. Se variam as redes, variam as formas eficazes de organização do "temos" e alteram-se as funcionalidades das arquiteturas conceptíveis dele decorrentes (suas distribuições).

Entre muitos exemplos disponíveis, os relatos de expedições científicas de Manguinhos ao interior do Brasil no início do século XX ilustram bem a inegável, mas nem sempre consciente, imbricação entre a conformação de um "temos", suas possibilidades de mobilização e as redes onde ele atua ou pretende atuar:

> "Desde que entramos em Goiás, a nossa principal moeda para obter dos habitantes que nos forneçam ovos, galinhas, mandioca, batata doce, etc., tem sido carrinhos de linha, agulhas, alfinetes e objetos de fantasia, como brincos, pulseiras, anéis, cordões dourados, de que nos munimos abundantemente no Rio de Janeiro. À exceção dos fazendeiros e alguns indivíduos viajados, ninguém liga importância ao dinheiro, e pode-se oferecer quantias relativamente grandes por uma dúzia de ovos, ou por um frango, que são recusadas desdenhosamente. Isso verificamos por várias vezes. Oferecíamos então às crianças e às mulheres, objetos de fantasia, carrinhos de linha, agulhas e logo nos eram oferecidas as mercadorias que desejávamos. ..." (apud Cukierman, 2007, p.386)

Não há necessidade de nos alongarmos aqui sobre o que aponta Fitzgerald, uma vez que é amplamente reconhecido que diferenças quantitativas podem se transformar em diferenças qualitativas. Mas não precisamos ir tão longe para perceber o quanto os "temos" daqueles expedicionários urbanos e daqueles habitantes rurais de Goiás de cem anos atrás diferenciavam e se imbricavam com os seus respectivos "somos". Vamos aqui apresentar vivências que ressoam "somos" = "temos" na contemporaneidade informatizada.

***

O filme ELA, de 2013, dirigido por Spike Jonze, mostra o envolvimento de Theodore com Samantha, o sistema operacional do computador que ele acaba de "ter" para si. Costumamos chamar de "sistema operacional" a um programa de computador que gerencia os recursos da máquina e simplifica o acesso para o usuário. Mas aqui, numa reconceituação no encontro do que seria dito "técnico" com o que seria dito "social" ou "humano", o conceito de "sistema operacional" é mesclado com ideias do campo da Inteligência Artificial e da vida, mostrando que esse sistema também gerencia recursos da vida do usuário, no limite (inatingível?) em que "ter" um sistema operacional é também "ser" um sistema operacional. A operação do sistema no filme começa pela configuração do *software* por voz. No lugar das esperadas perguntas do campo da técnica, uma voz masculina padrão surpreende com perguntas sobre a personalidade do usuário. Daí, após um breve tempo de operação, surge Samantha, voz feminina sensualizada, um programa de computador pelo qual Theodore vem a se apaixonar.

Essa relação meio-humana, meio-maquínica apresenta-se sedutora e com efeito supera contratempos diversos. Quando o sexo entra em cena, configura-se um constrangimento: Samantha não "tem" um corpo humano. A solução é apresentada a Theodore por ela mesma: um corpo "emprestado". A solução para Samantha "ser" mais humana despertou reações as mais diversas, manifestações otimistas e pessimistas, constrangimentos com relação à convivência homem-máquina e ao corpo emprestado, mas sempre acompanhadas da importante ressalva de que o filme apresenta um cenário (ainda?) no campo da ficção.

Xiaodie recebe os ajustes finais de engenheiro na fábrica da empresa chinesa Exdoll. Ela é equipada com uma IA similar ao sistema Siri da Apple. O homem de jaleco pergunta: “Como você se chama?”. “Me chamo Xiaodie, mas você pode me chamar de baby”, responde ela em mandarim com uma voz sensual.

.

No entanto, apesar de colocar-se como companheira ideal, há pontos em que Samantha encontra dificuldade em corresponder às expectativas do humano Theodore, pelo menos daquele humano Theodore, que se surpreende com os grandes números que integram a existência de Samantha, pelo menos daquela Samantha, que também não deixa de se decepcionar com as limitações humanas e as ilusões de Theodore a respeito dela. Conhecemos a capacidade dos computadores de alterarem as escalas, de tornarem viáveis as grandes quantidades ambicionadas pelos modos de existência que necessitam contar além de 1, 2, 3, 4, 5, muitos.[1] Alterar as escalas do nosso mundo, as quantidades com que vivemos, interfere não só na gerência do que “temos” como também do que “somos” nos relacionamentos que buscamos. Tanto é assim que é a partir do seguinte diálogo:

1 Uma população indígena conta “1,2,3,4,5, muitos”, possivelmente porque não necessita de um conceito de acumulação como o nosso, e daí não faz sentido para eles contar além do 5, distinguir o 6 do 7, ou o 1000 do 2000. Ferreira, E. S. (2005). "Racionalidade dos índios brasileiros." Scientific American Brasil **11**(Edição especial Etnomatemática): 90-93.

Theodore: Você conversa com alguém mais enquanto nós conversamos?

Samantha: Sim.

Theodore: Você está conversando com outro agora, neste momento? Pessoa? Sistema Operacional? O que seja?

Samantha: Sim.

Theodore: Quantos outros?

Samantha: 8.316.

Theodore: Você está amando alguém outro?

Samantha: Por que você pergunta isso?

Theodore: Eu não sei. Você está?

Samantha: Estive pensando sobre como conversar com você sobre isso.

Theodore: Quantos outros?

Samantha: 641.

que Theodore se dá conta de que Samantha não pode lhe satisfazer totalmente e encontra ânimo para buscar outra relação. É forçoso reconhecer que este Theodore mantém em seu "somos" um reduto ao abrigo da aquisição desenfreada dos "temos" do utilitarismo moderno. Quem se identifica com os sentimentos de Theodore constrói as fronteiras que delimitam redutos nos eixos ficcionalidades-realidades do filme e pode seguir tranquilo para casa pois afinal trata-se apenas de uma obra de ficção. Volta seguro, e guiado pelo WAZE que dribla os engarrafamentos. No percurso, o WAZE faz o pedido por uma fotografia do local por onde passam juntos, humano e WAZE, de modo a produzir informações mais apuradas a quem vier a passar por ali. Prontamente, o humano empresta seu corpo ao *software*, que, humanamente incorpóreo como Samantha, necessita de ajuda para a realização da tarefa. E, possivelmente notificados pelo WAZE através de conexão com outros *softwares,* cerca de 641 amigos poderão saber que este humano passou por ali.

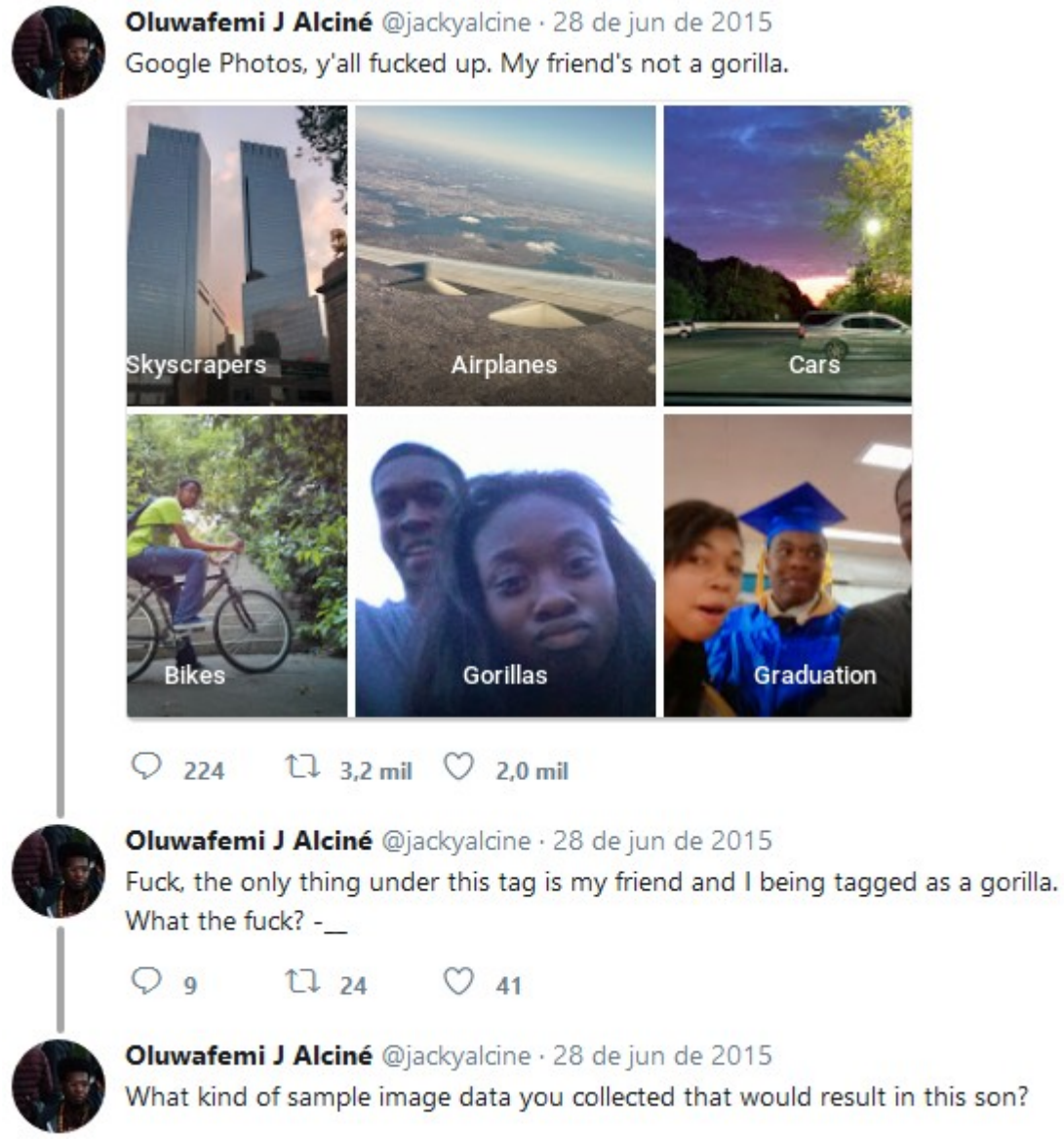
***

Insistindo em explorar regiões nada definidas entre a ficção e/ou a realidade, podemos visitar outro sistema computacional que dependeu e depende do corpo humano para realizar suas funções. Resumidamente, podemos dizer que esse sistema foi construído ao longo de uma década e se misturou com a habilidade humana de associar um nome a uma imagem (rotular uma imagem). Em outras palavras, trata-se de um software que depende do que a inteligência humana reconheceu em função do que os olhos enxergaram. Se a inteligência faz parte do nosso corpo, como conseguimos dar uma parte de nosso corpo a uma máquina? Afinal, "somos" inteligência ou "temos" inteligência? "Somos" ciborgues ou "temos" computadores? A depender das narrativas e metáforas que escolhermos ou inventarmos para conviver com essas questões, o nosso "somos" pode ser um tanto igual ou bem diferente do "temos".

Essas são algumas questões que giram em torno de uma controvérsia com o *Google Photos*, um sistema computacional para armazenamento ilimitado e gratuito de fotos. Lançado em maio de 2015, cerca de 10 anos após a publicação de um trabalho que propunha uma solução para a rotulagem de imagens em larga escala via um jogo de computador, o ESP (Ahn e Dabbish, 2004), o *Google Photos* além de cumprir a função básica de armazenamento de dados pessoais prometia o acesso simplificado em múltiplas plataformas, a organização de acervo por linha do tempo (criação de linhas do tempo de imagens) e a cereja do bolo: um filtro de buscas e classificação automatizada que agrupava fotografias em álbuns através do reconhecimento de rostos, lugares, objetos ou situações cotidianas. Imaginem o seguinte cenário muito comum nos dias de hoje: as lembranças que registramos em fotografias digitais durante alguns anos de vida, desde as situações mais triviais, como um passeio de bicicleta com os amigos, até momentos marcantes, como uma cerimônia de formatura, muito provavelmente estão espalhadas nos diversos dispositivos eletrônicos (computadores, *smartphones*, na nuvem, em câmeras digitais, em HDs, etc) que "temos" e que "somos" (pense na situação de incompletude/ansiedade que sentimos ao perder um celular). Agora imaginem uma inteligência artificial (um programa de computador que em tese tem a capacidade de aprender) que num passe de mágica, automaticamente, organize esse caos digitalizado. Essa era a promessa do *Google Photos*: organização, praticidade e eficiência.

Nos dias atuais, não seria um exagero afirmar que estamos sendo literalmente inundados pela vazão de conteúdos que muitas vezes nós mesmos produzimos: a cada

passo um *selfie* marca o compasso de muitas vidas nas redes sociais. A controvérsia do Google Photos começou quando Jacky Alciné, um jovem norte-americano, residente no Brooklyn, que talvez estivesse angustiado com o caos de imagens eletrônicas jogadas aleatoriamente em seus dispositivos pessoais, e, por isso, decidiu confiar o seu cotidiano em *selfies* e fotos a essa promessa de eficiência. Como podemos observar na figura a seguir, o *Google Photos* de fato organizou em álbuns fotográficos a vida de Jacky: as imagens dos passeios de bicicletas no álbum "Bicicletas"; as fotos da formatura no álbum "Formatura"; e, finalmente, *selfies* tirados com os amigos no álbum "**Gorilas"**. É isso mesmo: **GORILAS!**



Fonte: tweets de Jacky Alciné.

Jacky Alciné e seus amigos são pessoas pretas. As técnicas de reconhecimento facial (aprendizado de máquina, a técnica mais utilizada atualmente no campo da Inteligência Artificial) do *Google Photos* não montaram um álbum **AMIGOS** ou **MOMENTOS FELIZES** ao mapearem o sorriso no rosto de Jackie e seus amigos, elas montaram o álbum **GORILAS**.

"Temos" a Inteligência Artificial que "somos", uma inteligência majoritariamente branca, masculina, localizada nos EUA, falante da língua inglesa e racista. Até o presente (2018), os engenheiros da Google não conseguiram fazer com que a IA parasse de classificar imagens de pessoas pretas como gorilas. A solução provisória encontrada foi excluir todas as palavras gorilas da base.

Grosseiramente, uma Inteligência Artificial que implementa uma técnica de aprendizagem de máquina reconhece/aprende um padrão a partir de uma base/banco de dados. Aprendido o padrão, esse agente começa a realizar ações ou tarefas seguindo o padrão aprendido que reproduz o racismo.

***

As comunidades de software livre tipicamente se identificam como comunidades globais. O efeito decorrente da informação que integra o quase "senso comum" da participação aberta no software livre faz com que, com a ausência de propriedade, nós, brasileiras, não sem certa ironia nas palavras, "temos" o software livre, o que nos leva à ideia de que "somos" atrizes neste importante campo da atividade informática.[2] Um exame minucioso, no entanto, revela um descompasso, até certo ponto surpreendente, entre o que os brasileiros e brasileiras acreditamos, e indulgentemente até nos auto incentivamos a acreditar, que "somos" – atrizes de peso significativo no software livre – e o que "temos", obrigando-nos a reconhecer que não "somos" o que pensamos que

2 Opta-se nesta sessão por usar o feminino no lugar do masculino para se referir a ambos os gêneros. Espera-se com isso convidar a leitora a uma reflexão sobre as desigualdades existentes entre homens e mulheres. No contexto da Engenharia de Software essa desigualdade se manifesta de diversas maneiras, como, por exemplo, ao favorecer as alterações de código feitas por homens em comparação com as alterações feitas por mulheres, conforme demonstrado por uma análise das contribuições feitas utilizando a plataforma de hospedagem de código-fonte GitHub. Terrell, J. e. a. (2017). "Gender differences and bias in open source: pull request acceptance of women versus men." PeerJ Computer Science **3**(May 2017): Disponível em: <https://peerj.com/articles/cs-111>. Acesso em: 24 jul. 2017

"somos" porque não "temos" entre nós, em escala significativa e proporcional, a residência de autoras do software livre.

Observando mais atentamente a comunidade do WordPress (WP), software livre utilizado em pouco mais de 30% dos sites da internet[3], nota-se que, no discurso, o WP é apresentado como uma meritocracia aberta à participação de todas, independentemente de onde estejam, como exemplificado na passagem retirada do site do software:

> "Tudo o que você vê aqui, da documentação ao código, foi criado para e pela comunidade. WordPress é um projeto de código aberto, o que quer dizer que existem centenas de pessoas *espalhadas por todo o mundo* trabalhando nele."[4] (grifo nosso)

Com o intuito de problematizar essa ideia, a de que não "temos" o WP à nossa disposição, e, portanto, dele não "somos" autoras, realizamos um levantamento do país de residência das desenvolvedoras do WordPress a partir da extração de dados do repositório de código e o cruzamento dessas informações com o perfil de cada uma no próprio site do WP. Foi levantada a hipótese de que, assim como na ciência a ideia de universal representa um particular no poder, o global do WordPress esconderia a concentração das desenvolvedoras em alguns poucos países.

Os dados do repositório de código foram coletados em abril de 2016 e foram encontradas 53 desenvolvedoras com acesso de escrita ao repositório do *core*, 1876 desenvolvedoras que haviam feito contribuições incorporadas ao *core* e 25.057 que haviam contribuído com ou criado um *plugin*[5]. Nos três grupos foi identificada uma concentração significativa de *commits*[6] em um número pequeno de indivíduos, como

3 Disponível em: https://w3techs.com/technologies/overview/content_management/all. Acesso em: 25 maio 2018.

4 Tradução nossa de: "Everything you see here, from the documentation to the code itself, was created by and for the community. WordPress is an Open Source project, which means there are hundreds of people all over the world working on it." Disponível em: https://wordpress.org/about/. Acesso em: 28 fev. 2017.

5 O *core* é como é denominado o núcleo central do WordPress. Apenas um grupo restrito de desenvolvedoras com autorização pode alterá-lo diretamente. O restante da comunidade deve enviar contribuições que podem ser aceitas ou não ou então estender as funcionalidades do *core* criando um plugin.

6 Um commit é o ato de enviar para um repositório de código uma alteração em um ou mais arquivos. Cada commit cria uma nova versão no repositório, que armazena informações sobre a alteração, como o que foi feito, por quem e quando.

mostra a tabela 1, que usa como recorte, para efeito de comparação, 80% das alterações feitas em cada um dos casos estudados.

Tabela 1: Percentual de desenvolvedoras que realizaram 80% das contribuições

Grupo	**Percentual de desenvolvedoras responsáveis por 80% das alterações**
Core developers	21%
Contribuidoras do *core*	4%
Desenvolvedoras dos *plugins*	23%

A partir da identificação das desenvolvedoras que fazem parte da comunidade do WordPress, partiu-se para encontrar o seu país de residência. A hipótese inicial foi confirmada e viu-se que elas estão concentradas em alguns poucos países de língua inglesa, em especial os Estados Unidos. Vale destacar que o grupo mais concentrado nos EUA é justamente o grupo que tem mais poder para decidir os rumos do projeto, a saber, as desenvolvedoras com acesso de escrita ao repositório de código do *core*. 58% delas estão nos EUA e 83% em um país cujo idioma oficial é o inglês. Isso indica que, ao menos no caso do WordPress, é possível extrapolar para o desenvolvimento de software a discussão dos Estudos CTS sobre a universalidade da ciência como um particular no poder.

Essa concentração de desenvolvedoras em poucos lugares traz uma questão relevante para a comunidade brasileira do software. Se é aceito o discurso de que o WordPress é uma meritocracia, onde a participação e as posições de poder são definidas pelo desempenho de cada uma das participantes, pode-se cair na armadilha, identificada por Yuri Takhteyev, de explicar a concentração de desenvolvedoras em países de língua inglesa a partir da ideia de que esses locais concentram as desenvolvedoras mais talentosas e de que o problema dos demais locais é a falta de talentos. (Takhteyev 2012:7)

O “global” do WordPress fala inglês, entende referências a elementos da cultura estadunidense e pode frequentar encontros presenciais que ocorrem em cidades dos Estados Unidos ou da Europa. Ou seja, são os centros vestidos como representantes do global. É mais custoso para alguém de fora dos centros participar da comunidade do WordPress do que para aqueles que estão dentro. Os que estão fora precisam fazer um

esforço maior para entender a cultura daqueles que estão dentro, enquanto que os que estão dentro podem contar com o esforço dos que estão fora para entender a sua própria cultura.

a meritocrata: – Os (norte)americanos são diferentes de nós.

a latinoamericana: – É. Eles têm mais tecnologia.

Com estas observações não se pretende sugerir que as membras da comunidade brasileira do WordPress deixem de contribuir com o software e criem uma solução puramente local. Seguindo o que propõe Ivan da Costa Marques, a sugestão é que as brasileiras realizem suas contribuições cientes das assimetrias discutidas e até o momento em que elas forem úteis para seus próprios objetivos. Em outras palavras, a sugestão é que ousem saber para diminuir o descompasso entre o que acreditamos que "somos" e o que de fato "temos". (da Costa Marques 2014:9)

***

Trazemos aqui uma quarta situação que se faz visível aqui e mundo afora, em que há um descompasso entre o que cientistas (ciência) e profissionais (trabalho) da computação pensamos que "somos" e o que efetivamente "temos" no que diz respeito aos programas que escrevemos. Esse descompasso se sustenta em uma separação entre o pensar e o fazer, que por sua vez, tem suas bases na separação entre mente e corpo. Na narrativa que conduziremos aqui, a sobrevalorização daquilo que se estabelece como ciência-pensar-mente se faz diretamente presente em nossas vidas, ao demarcar um território especialmente reservado para o software proprietário. Se "somos" cientistas, "temos" o software proprietário, que por ser validado e certificado pelos pressupostos de exatidão e objetividade da lógica matemática, tem justificado seu o preço estratosférico. Se "somos" engenheiros, "temos" o código aberto, que opera no mundo da vida, pela dinâmica e fluidez dos processos sociais, contaminado pelo trabalho-fazer-corpo.

Na Ciência da Computação, chamamos de "especificações formais de software" as descrições abstratas de um software, em linguagem lógica matemática. A meta é que dessa descrição matemática se possam derivar implementações, bem como provar propriedades sobre o software. Costuma-se dizer que o software que cumpre com

propriedades expressas em sua especificação formal está "formalmente verificado", ou seja, pode ser considerado correto e confiável. Esta concepção que situa o software como um objeto formal nasceu na conveniência de estabelecer o campo da Ciência da Computação como ciências exatas, como se vê nas declarações dos cientistas fundadores do campo na década de 1960: "Programação de computadores é uma ciência exata em que todas as propriedades de um programa e todas as consequências de executá-lo em qualquer ambiente podem, em princípio, ser encontradas no próprio texto do programa através de raciocínio puramente dedutivo".[7] (Hoare 1969:576) Mas, ainda na década de 1960, acreditar que "somos" matemáticos, supostamente habitantes de um mundo formal, encontrava resistência por parte daqueles que entendiam que o que "temos" na construção de software (assim como na própria matemática) é uma atividade coletiva, cujo sucesso depende de negociação: "(...) o estudo de algoritmos e modelos de programas se desenvolverá como qualquer outra atividade matemática, principalmente por meio de mecanismos sociais informais, e muito pouco, ou mesmo nada, por meio de mecanismos formais"[8] (De Millo, Lipton et al. 1979:277)

O mundo da vida, o mundo em que suamos, gozamos, amamos e odiamos, é um mundo em constante mudança, em fluxo, e, neste mundo, "temos" um software confiável quando ele cumpre com as suas demandas, o que não significa necessariamente cumprir com sua especificação formal. Hoje, assim como era na década de 1960, quando um programador entrega um software, ele usualmente aplica uma bateria de testes de modo a verificar que tudo funciona conforme o esperado. Daí então, ele próprio, o programador no mundo da vida, garante: "Está correto!". Por outro lado, no mundo formal, da matemática, um software é confiável quando ele atende aos seus requisitos, isto é, quando ele é submetido a um mecanismo de verificação formal que atesta sua conformidade a um referencial específico, supostamente impessoal e objetivo. Este debate era claramente colocado nos primórdios da configuração do campo da Ciência da Computação. Os porta-vozes da matematização, como Edsger Dijkstra, empurravam a tarefa da programação para longe das circunstancialidades da vida:

7 Tradução nossa de: "Computer programming is an exact science in that all the properties of a program and all the consequences of executing it in any given environment can, in principle, be found out from the text of the program itself by means of purely deductive reasoning"

8 Tradução nossa de: "the study of algorithms and model programs will develop like any other mathematical activity, chiefly by informal, social mechanisms, very little if at all by formal mechanisms."

*"*Testes mostram a presença, não a ausência de bugs."[9] (NATO 1969,1970) Mereciam a ironia de Alan Perlis, que não admitia o distanciamento entre software e demandas da vida, e portanto descartava qualquer garantia de acurácia e correção absoluta por meios matemáticos: "Existem duas formas de escrever programas livres de erros. Somente a terceira funciona."[10] (Perlis 1982) Parecia claro para Perlis, que a verificação formal desloca autoridade e confiança das mãos do programador para a matemática, uma entidade habitante do mundo dos conceitos, que tem sido identificada com rigor, exatidão e verdade. Requisitos formais fixados versus atendimento a demandas em fluxo: vemos aqui o conflito entre o que se diz objetivo e formal (um código de um sistema) e o que se diz subjetivo e social (um sistema em execução), entre um "somos" essencialista e um "temos" encarnado que requer negociação.

Os matemáticos bem sabem da impossibilidade de garantir, isto é, provar que um programa efetivamente cumpre com aquilo que se espera que ele faça. Então o que se pode esperar em termos de especificações formais é a conformidade com algumas propriedades específicas. Na década de 1970, quando já se fortalecia uma comunidade dedicada à atividade de programação, isto já era evidente para os implementadores: "Com base em dez anos de experiência no projeto, implementação e uso de software (...) eu não me lembro de uma só vez em que uma prova (matemática) da correção de um programa tenha sido útil."[11] (Hill 1979) Apesar disso, o discurso dos partidários dos métodos formais costuma ser ainda hoje carregado de certeza e autoridade que não se confirmam quando o software é implementado ou posto em operação.

Nos dias de hoje, em pleno século XXI, os cientistas formalistas ainda propõem ambiciosos projetos para os 10 anos seguintes desconsiderando que o que "temos" são interações com processos sociais: "É chegada a hora de embarcar num Grande Desafio internacional para construir um verificador de programas que use provas lógicas para verificar automaticamente a correção de programas submetidos a ele."[12]. (Hoare and

9 Tradução nossa de: "Testing shows the presence, not the absence of bugs"

10 Tradução nossa de: "There are two ways to write error-free programs; only the third one works"

11 Tradução nossa de: "On the basis of ten years experience in the design, implementation and use of software (...) I cannot recall a single instance in which a proof of a program's correctness would have been useful."

12 Tradução nossa de: "[T]he time is ripe to embark on an international Grand Challenge project to construct a program verifier that would use logical proof to give an automatic check of the correctness of programs submitted to it."

Misra 2005:13)Este “somos” formalista encastela a garantia do software correto na ciência formal: “O projeto vai prover as *bases científica*s de uma solução para muitos dos problemas de erros de programação que afligem todos os construtores e usuários de software hoje”[13], que autoriza a este “somos” um rentoso “temos”: “Antecipamos que o projeto vai durar mais de 10 anos, consumir cerca de mil pessoas-ano de esforço científico qualificado, vindo de todo o mundo”[14]. Ao mesmo tempo, a ciência formal reafirma suas ambições, fortalecendo um “somos” universalista: “O objetivo final do projeto é que o verificador do programa certifique *todos* os programas automaticamente”[15], e marginaliza (coloca à margem) os testes funcionais: “Na prática, em qualquer tempo, haverá sempre uma porcentagem de condições de verificação que o verificador não pode provar. Para estas, as técnicas tradicionais teste e verificação da execução do programa serão empregadas”[16] (grifos nossos). Como já foi dito neste texto, se variam as redes, variam também as formas eficazes de organização do “temos”, sua utilização proporciona rendimentos diversos se mudam as circunstâncias, os pontos de vista, os interesses, os hábitos e as competências.

Os sistemas de hoje nos mostram que, em situações de grande dificuldade computacional, o que “temos” são processos sociais que entram em cena e resolvem satisfatoriamente o problema em questão, uma operação conjunta (co-operação) entre o formal e o informal. Vemos que afirmações muito generalistas sobre a correção de sistemas contribuem para fortalecer uma conformação de poder que supervaloriza o “somos” matemáticos, a matemática e as bases formais, frente ao que “temos”, técnicas tradicionais de testes dos programas. Além disso, disseminam no senso comum a ilusão de que um programa formalmente verificado estaria *livre* de erros, e de que as especificações formais, embora difíceis, garantiriam a correção de sistemas. Fica claro nos depoimentos dos formalistas o estabelecimento de uma fronteira que separa de um lado os produtores de software muito caros e “muito confiáveis”. Do outro lado estariam

13 Tradução nossa de: “The project will provide the scientific basis of a solution for many of the problems of programming error that afflict all builders and users of software today”

14 Tradução nossa de: “We anticipate that the project would last more than ten years, consume over one thousand person-years of skilled scientific effort, drawn from all over the world”

15 Tradução nossa de: “The ultimate aim of the project is that the program verifier will certify *all* programs automatically”

16 Tradução nossa de: “In practice, at any given time there will always be a percentage of verification conditions that the verifier cannot prove. For these, the traditional techniques of testing and run-time checking will be employed.”

os charlatões, engenheiros, que produzem software barato e vagabundo, software de código aberto. O trecho a seguir é parte do debate posterior à proposição do grande desafio pelo cientista inglês Tony Hoare. Transcrevemos a observação de Richard Bornat dirigindo-se à Hoare:

> Se prometemos desenvolvimento de programa mais barato (...) estamos, infelizmente - como você sabe como um inglês - em concorrência com um número de charlatões (Tony Hoare ri), que afirmam que massageando o processo de gestão do trabalho de um programador, pode-se entregar um software mais barato e mais confiável. (…) E não poderemos fazer o que esses charlatães - não diremos a palavra "charlatão" - o que aquelas pessoas dizem que podem fazer, mas faremos outra coisa. E parece-me, esse é o único ponto em que espero ver uma melhoria do que é uma descrição verdadeiramente maravilhosa do porquê sou cientista e não engenheiro[17]. (Hoare and Misra 2005:17)

Referindo-se ao outro lado da fronteira, o lado dos "engenheiros", os cientistas formalistas deixam claro a quem se se referem: "Para nós, os engenheiros, gostemos ou não, são pessoas como Linus Torvalds, com sua atitude em relação às especificações sendo o que é"[18]. Linus Torvalds é o principal desenvolvedor do *kernel* do Linux, um sistema de código aberto. O rótulo de charlatão deveu-se a uma mensagem por e-mail em que Torvalds deixa clara sua posição com respeito às especificações formais, com destaques em letras maiúsculas:

> Uma 'espec' está perto do inútil. Eu nunca vi uma especificação que fosse simultaneamente grande o suficiente para ser útil e precisa. E eu vi um monte de asneira total baseado em especificações. É a pior maneira de escrever software, porque, por definição, significa que o software foi escrito para corresponder à teoria, não à realidade. Então, há duas razões MAIORES para evitar especificações: (1) elas estão perigosamente

17 Tradução nossa de: "[I]f we promise cheaper program development (...) we are unfortunately – as you know as an English person – in competition with a number of charlatans (Tony Hoare laughs), who would claim that by massaging the process of management of the work of a programmer, one can deliver cheaper and more reliable software. (…) And we will not be able to do what those charlatans—we won't say the word "charlatan"—what those people over there say they can do, but we will do something else. And it seems to me, that is the only point where I hope to see an improvement of what is a truly marvelous description of why I am a scientist and not an engineer."

18 Tradução nossa de: "For us, the engineers (...) are people like Linus Torvalds, with his attitude to specifications being what it is."

erradas. A realidade é diferente e qualquer um que pense que especs importam mais que a realidade deve sair da programação do kernel AGORA. Quando a realidade e as especificações colidem, a especificação tem um significado zero. Zilch. Nada. Nenhum. É como a ciência real: se você tem uma teoria que não casa com experimentos, não importa o quanto você goste dessa teoria. Está errado. Você pode usá-la como uma aproximação, mas você DEVE ter em mente que é uma aproximação. (2) As especificações têm uma tendência inevitável para tentar introduzir níveis de abstração e políticas de redação e documentação que façam sentido para uma especificação escrita. Tentar implementar o código real a partir da especificação faz com que o código pareça e funcione como ASNEIRA. O exemplo clássico disso são os protocolos do modelo de rede OSI, design clássico de especificação, que tem absolutamente zero relevância para o mundo real. Ainda falamos sobre o modelo de sete camadas, porque é um modelo conveniente para discussão, mas isso tem absolutamente nada a ver com qualquer engenharia de software da vida real. Em outras palavras, é uma maneira de falar sobre as coisas, não de implementá-las. E isso é importante. As especificações são a base para falar sobre as coisas. Mas não são uma base para implementar software. Então, por favor, não fale em especificações. Os padrões reais crescem apesar das especificações, não graças a elas[19]. (https://yarchive.net/comp/linux/specs.html)

19 Tradução nossa de: "A 'spec' is close to useless. I have never seen a spec that was both big enough to be useful and accurate. And I have seen lots of total crap work that was based on specs. It's the single worst way to write software, because it by definition means that the software was written to match theory, not reality. So there's two MAJOR reasons to avoid specs: (1) They're dangerously wrong. Reality is different and anybody who thinks spec matter over reality should get out of kernel programming NOW. When reality and specs clash, the spec has zero meaning. Zilch. Nada. None. It's like the real science: if you have a theory that doesn't match experiments, it doesn't matter how much you like that theory. It's wrong. You can use it as an approximation, but you MUST keep in mind that it's an approximation. (2) Specs have an inevitable tendency to try to introduce abstraction levels and wording and documentation policies that make sense for a written spec. Trying to implement actual code off the spec leads to the code looking and working like CRAP. The classic example of this is the OSI network model protocols. Classic spec-design, which had absolutely zero relevance for the real world. We still talk about the seven layers model, because it's a convenient model for discussion, but that has absolutely zero to do with any real-life software engineering. In other words, it's a way to talk about things, not to implement them. And that's important. Specs are basis for talking about things. But they are not a basis for implementing software. So please, don't bother talking about specs. Real standards grow up despite specs, not thanks to them".

De um lado a autoridade do “somos” uma ciência, limpa e exata, que não se deixa contaminar pelas questões mundanas, aqui incorporadas na figura do engenheiro (ou do cientista charlatão). De outro lado, o “temos” de uma ciência da computação contaminada de vida. Este “somos” afiliado à Microsoft por seu proponente Tony Hoare é suficientemente prestigiado a ponto de justificar custos elevados e grandes investimentos. O “temos” porta-voz do código aberto oferece possibilidades alternativas ao trabalho em software pelo uso do código aberto.

***

Uma parte cada vez maior do cotidiano das pessoas se aloja no ciberespaço. Antes visto como o lugar das realidades virtuais, um lugar que se dizia sem espaço (“a spaceless place”), o ciberespaço torna-se cada vez um lugar real por onde as pessoas transitam em suas atividades cotidianas. As portas de entrada e saída deste lugar encontram-se na materialidade dos teclados, controles (mouses), telas, óculos e das cada vez mais diversas próteses. Tornou-se mais visível que a cada dia atravessamos diversas vezes estas portas intercalando idas e vindas ao ciberespaço (onde as coisas acontecem nas memórias das máquinas) no fluxo de nosso mundo da vida.

2018 é um ano de eleições no Brasil. Na cabine eleitoral “somos” eleitores que rapidamente entram e saem de um ciberespaço, transitando pelas portas oferecidas pela urna eletrônica que “temos”, crucialmente imbricada com a sociedade e, de modo geral, enaltecida por aqueles “somos” que “detêm” os demais dispositivos que a sustentam na rede. Dizem eles hoje,

> “[e]scolhemos pelo voto, em *eleições absolutamente livres* e promovidas com imensa eficiência e com competência pelo Tribunal Superior Eleitoral, os ocupantes do poder executivo. Se há uma coisa que deu certo (nos quase trinta anos de regime democrático) foi o aperfeiçoamento do nosso processo eleitoral. Hoje, é todo *controlado eletronicamente, sem fraudes, sem filas, sem os inconvenientes ‘cabos eleitorais’ e o melhor: é rápido*”. (Antonio Delfim Netto, Folha de São Paulo, 24/05/2017, p. A2) (ênfase acrescentada).

Aí estão os mais enaltecidos ícones daquilo que as mesclagens espaço-ciberespaço viabilizam: controle, eficiência e rapidez deslocando as ações das pessoas, eleitores e cabos eleitorais para o automatismo possível criado pela urna eletrônica.

Não queremos neste momento questionar estes ícones nem a ilusória impossibilidade de fraude. É outra a questão que queremos ressaltar: "somos" eleitores em "eleições absolutamente livres?" É somente pelas entradas e saídas fixadas na urna que "temos", nos teclados, telas e impressoras da maquinaria de informação que se entra e sai do ciberespaço onde os votos são contados.

Diz-se que os eleitores "somos" absolutamente livres, mas o que somos não independe do que "temos". "Temos" urna eletrônica e "somos" eleitores cujas opções de ação (liberdade) ela pode, afinal, aumentar ou diminuir. Em sua versão atual, a urna eletrônica prevê o voto em branco mas não prevê o voto nulo. A diferença entre esses dois votos é negada. Essa diferença é vitalmente política. O que "temos" define o que "somos" como cidadãs e cidadãos porque define o que pode e que não pode habitar (estar presente) no universo da eleição.

A exclusão da opção do voto nulo resultou de um grupo específico de pessoas (e não do eleitorado), defendida como uma decisão dita "técnica" que encarna e ressoa, em instâncias locais e atualizadas, a construção racional de "uma alternativa à democracia entendida como governo do povo pelo povo", denunciada por Bernard Manin em seu marcante estudo histórico dos debates que levaram à constituição americana. (MANIN 1997:165) A ata da Plenária nº. 04 da Comissão de Informatização das Eleições Municipais de 1996, ocorrida em Brasília, em 06 de junho de 1995, atesta que

> "por maioria presente, fica estabelecida que a solução a ser adotada não deverá conter de forma explícita a opção de voto nulo [...] Fica registrada a não concordância dos membros Jorge Lheureux de Freitas, Luiz Roberto da Fonseca e Márcio Luiz Guimarães Collaço com esta decisão. A solução deverá conter de forma explícita a opção de voto em branco." (Mendes 2010:105)

Os três membros dissidentes questionaram a ausência da tecla "nulo" no teclado do terminal da urna eletrônica. Segundo Osvaldo Catsumi Imamura, integrante do Grupo Técnico instalado pelo TSE em setembro de 1995,

> "[a] questão da tecla nulo foi resolvida pela Corte do TSE. ... A urna eletrônica é um instrumento de auxílio ao eleitor para a manifestação do

> seu voto. Assim sendo, foi entendido que a expressão do voto se manifesta na forma de voto no candidato, na legenda e em branco. Como o voto nulo também faz parte desta manifestação, mas não representa o voto propriamente dito (sic), optou-se pela forma de expressão do voto nulo por meio de voto em candidato ou legenda inexistente...” (Mendes 2010:106)

Em agosto de 2006, no Programa Roda Viva, da Rede Brasil, o então presidente do TSE, ministro Marco Aurélio de Mello afirmou que “o voto nulo ... não deve ser feito porque é uma fuga”. (MENDES, 2010, p. 110-111) Foi a racionalidade de um grupo restrito de pessoas que classificou o voto nulo como um voto “errado”, ou um voto inadvertidamente invalidado, para justificar a não inclusão da tecla “nulo” no teclado da urna eletrônica. Mas não materializar na urna a opção “voto nulo” não é uma decisão “técnica”, é uma decisão vitalmente política que, uma vez incorporada na máquina, torna-se menos visível:

> ... a grande maioria da população domina os meios eletrônicos através de uma relação binária simples. Entretanto, quase sempre surgem dificuldades quando nesta relação é incluída uma interpretação a partir de algumas opções. ... Na urna eletrônica opções por candidatos [ou] legendas [...] o eleitor tecla os números, olha o monitor e confirma. [Caso queira votar em branco, o eleitor aperta a tecla “BRANCO” e confirma.] Já no voto nulo, o eleitor tecla um número inexistente, confirma e aparece no monitor uma expressão constrangedora “NÚMERO ERRADO”, ou seja, o eleitor não está votando nulo e sim, votando errado, segundo a Justiça Eleitoral. Somente após confirmar mais uma vez, o voto será anulado. Portanto, a própria urna eletrônica dificulta o voto nulo. (Mendes 2010:112)

Vetar a opção do voto nulo como uma forma autêntica de expressão da preferência política do/a eleitor/a é mais tijolo na construção autoritária de “uma alternativa à democracia entendida como governo do povo pelo povo”. A urna eletrônica sem a tecla “nulo” induz uma diminuição do espaço de liberdade do/a eleitor/a. Mas, como eleitores, “somos” a urna que “temos”.

***

Por serem mais atrelados ao mundo da vida, o único mundo em que, aqui e agora, cada um goza, sua, ama e odeia, os “temos” oferecem mais clareza e maior potencial de mobilização do que os “somos” mais permeáveis às promessas mistificadoras. À guisa de conclusão, naquela mesa do V Congresso da Associação Brasileira de Estudos Sociais das Ciências e das Tecnologias (ESOCITE.BR) defendemos a ideia de que, nos debates sobre o que seria “o Brasil que eu quero”, devemos inclinar a balança, aumentando o peso dos circunstancialistas “temos” em relação aos essencialistas “somos”.

***

Referências

da Costa Marques, I. (2014). O que os Estudos CTS podem fazer pela América Latina? Uma resposta antropofágica e alguns exemplos. Rio de Janeiro.

De Millo, R., R. Lipton and A. Perlis (1979). "Social Process and Proofs of Theorems and Programs." Communications of the ACM **22**(5): 271-280.

Ferreira, E. S. (2005). "Racionalidade dos índios brasileiros." Scientific American Brasil **11**(Edição especial Etnomatemática): 90-93.

Hill, R. (1979). "In: Letters to the editor." Communications of the ACM **22**(11): 621:622.

Hoare, C. A. R. and J. Misra (2005). Verified software: theories, tools, experiments vision of a grand challenge project. IFIP TC 2/wg 2.3, VSTTE 2005, 1. B. W. Meyer, J. . Zurich, Proceedings… Zurich: LNCS, Springer. **4171:** 1-18.

Hoare, T. (1969). "An axiomatic basis for computing programming." Communications of the ACM **12**(10): 576-583.

Mendes, P. S. P. (2010). A urna eletrônica brasileira: uma (des)construção sociotécnica. Doutor, Universidade Federal do Rio de Janeiro.

NATO (1969,1970). NATO, Software Engineering: Report of a conference sponsored by the NATO Science Committee.

Perlis, A. (1982). "Epigrams in Programming." ACM’s SIGPLAN publications(September 1982).

Takhteyev, Y. (2012). Coding places : software practice in a South American city. Cambridge, Mass., MIT Press.

Terrell, J. e. a. (2017). "Gender differences and bias in open source: pull request acceptance of women versus men." PeerJ Computer Science **3**(May 2017): Disponível em: <https://peerj.com/articles/cs-111>.

Mini Currículo:

Isabel Cafezeiro é doutora em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (2000) e pós-doutora pelo Programa de Pós-Graduação em História das Ciências e das Técnicas e Epistemologia da UFRJ (2011). Desde 1994 é professora do Instituto de Computação da Universidade Federal Fluminense, atuando na área de Ciência da Computação com ênfase em Lógicas e Semântica de Programas, na área de Sistemas de Informação com ênfase em Computação e Sociedade e Abordagens Sociotécnicas, e na área de Estudos Sociais de Ciência e Tecnologia, focando principalmente a História da Computabilidade e investigações sobre o trabalho acadêmico. É sócia fundadora da ESOCITE.BR e membro da diretoria na gestão 2017-2019.