

## A P R E S E N T A Ç Ã O

# UM MUNDO FEITO (QUASE COMPLETAMENTE) DE SOFTWARE

Silvio Lemos Meira

**E**ste Núcleo Temático de *Ciência e Cultura* é sobre software, o mais novo meio de armazenamento de conhecimento da curta história da humanidade. Parte de uma linhagem que começa com DNA, passa por cérebros, ferramentas e livros (e outros registros estáticos), software tem vantagens competitivas fundamentais sobre todos os outros: é abstrato, muito maleável (em relação aos outros meios) e executável. Codificado em *bis*, pode ser transferido, na Era da Informação, em frações de segundo para qualquer lugar do planeta e para fora dele, e pode “rodar” em variados tipos de plataforma, desde telefones celulares a sistemas de controle de automóveis e elevadores até gigantes computadores capazes de simular ecologias inteiras e, claro, o PC sobre a mesa de trabalho do leitor.

Esta abertura está dividida em quatro partes, interligadas, que vão da nossa relação com tecnologias até trabalho e software na sociedade da informação. O objetivo destes textos, que estão disponíveis com todos os seus links em [www.meira.com](http://www.meira.com), é estabelecer o contexto para o Núcleo Temático: onde, por que e como estamos trilhando os caminhos de tecnologias da informação e para onde eles nos levam —ou poderiam levar, se o mundo ao redor, principalmente aqui no Brasil, estivesse mais disposto a mudar, mais rápido.

A primeira parte, “O futuro era melhor no passado”, é sobre a relação da humanidade com as tecnologias da informação, e de como sempre esperamos algo muito melhor do que a tecnologia pode nos dar, até porque as previsões de futuro são sempre muito mais auspiciosas do que a realidade futura a provará. A segunda, “No caos, a receita do sucesso”, discute o problema tecnológico e econômico de desenvolvimento de sistemas de informação, as aplicações que fazem a sociedade moderna funcionar. O problema, aí, é que só 28% dos projetos de software terminam bem, o que dá uma idéia do que ainda precisa ser resolvido para que o desenvolvimento de sistemas seja eficiente e eficaz, apesar dos significativos avanços, em qualidade e produtividade, da última década.

A terceira parte, “Santa Rita de Cássia, padroeira dos sistemas de informação”, continua o tema anterior, tratando o caso do software do Space Shuttle, sem dúvida um dos melhores (e mais caros) sistemas do mundo, ainda assim com os problemas sobre os quais discorreremos no texto. Antes de apresentar os artigos restantes do núcleo, a quarta parte trata de “Trabalho, emprego, software e economia da informação”, onde discutimos um pouco do que é (e poderá vir a ser) a sociedade da informação, qual o papel de software na mesma e como o Brasil está deixando passar uma das maiores oportunidades da história, porque os novos empreendimentos, principalmente de software, estão amarrados a um contexto legal completamente anacrônico e cuja revisão parece muito difícil hoje. Finalmente, segue-se uma pequena apresentação de cada um dos artigos e seus autores.

**O FUTURO ERA MELHOR NO PASSADO** Daqui a cinquenta anos os computadores serão invisíveis. O que não significa que serão feitos de alguma substância transparente, mas que estarão de tal forma embutidos nos objetos que nos cercam, e no meio ambiente, que não mais conseguiremos nos preocupar com os “computadores”... Em muito menos tempo, haverá centenas de bilhões, talvez dezenas de trilhões de dispositivos *on-line*, em tempo real: desde a porta da sua casa e a casa como um todo, até sua camisa e seus óculos. Falando nisso, deficientes visuais que tenham um olho muito melhor do que o outro poderão decidir trocar o olho “ruim” por algo mais sofisticado, como um sistema ótico adaptativo ou mesmo uma visão infravermelha... robótica, claro. Em cem anos, não haverá nem como pensar que os robôs da época tentariam tomar-nos o mundo, porque não haverá mais “nós”, pelo menos como nos entendemos hoje. A cada (r)evolução da robótica, da biologia, da medicina, dos implantes, não haverá como não querer-mos ter um pouco das novas possibilidades fazendo parte das nossas vidas, de sua melhoria de qualidade, parte dos nossos corpos. O que tornará nossos corpos físicos cada vez mais lógicos, mais simbólicos, mais tecnológicos. Para todos os que se assustaram com o parágrafo anterior, quantos carregam tecnologia como óculos (sistemas passivos para melhoria de qualidade de visão), não têm um dente obturado, usam relógio (seu outro “clock”...), têm um pino de titânio no joelho ou, melhor ainda, falam, tecnologia desenvolvida de forma única pela nossa espécie? Pois... o futuro faz uso das mesmas licenças do passado, poéticas ou não (...só uso porque é necessário, senão...) para nos abrir novas e infinitas possibilidades de interação e magnificação da presença humana no universo. Incluindo tudo o que for preciso para nossa saída, organizada, da Via Láctea, galáxia que se encontra em rota de colisão com Andrômada... apesar do impacto não ser esperado pelos próximos sete bilhões de anos. Mas que vai acontecer, vai. Estamos vivendo num curioso instantâneo do espaço-tempo, como todos antes de nós viveram. Mas a percepção da crise de sobrevivência que nosso planeta enfrenta tem criado situações interessantes, magníficas e, também, ridículas. Nossa relação com o ambiente tem sido de amor ou ódio. Uns, destroem sem pensar. Outros, preservam sem pensar. Dia desses, estive numa conversa sobre a “preservação” de um conjunto de dunas, o que era para ser entendido como estabilização das ditas. Ora, se dunas são coisas móveis por excelência, paralisá-las, por conseguinte, é interferir destrutivamente no meio. Mas o que se queria, de fato, era “preservar” uma beleza natural criada por um conjunto de dunas que havia chegado até o lugar onde estavam... algo como achar que a foto ou o filme são mais desejáveis, definitivas e melhores do que a realidade, como se o mapa de Borges fosse o mundo nele representado...

Nossa percepção do presente parece partir do princípio de que “no passado, o futuro era melhor”. Tomamos as promessas do passado, comparamos com

o presente e nos forçamos a achar que tudo poderia ser muito melhor. Em informática, este é certamente o caso, apesar do gigantesco progresso que experimentamos nas cinco décadas da computação eletrônica. Mas, nas últimas semanas, vários pedaços de informática que eu costumo usar entraram em pane — e alguns deles me deixaram em pânico. O carro de “Kakinhas” se “desprogramou” (segundo a concessionária) e passou a limitar a força do motor a uns cinco pôneis. Quase não andava. O roteador da rede banda larga do meu prédio se “desconfigurou” (segundo o provedor) e fiquei mais de dois dias sem acesso, além de ter que testar (com auxílio do *call center*) todo o serviço e seus *links*, coisa que dificilmente seria feita por alguém sem conhecimentos de computação, mesmo com ajuda remota de um técnico. Um dos elevadores do prédio “pipocou” (segundo o porteiro...) porque a umidade “derrotou” a “placa”, ou seja, porque entrou água no computador de bordo do bicho. E, claro, meu celular foi clonado — o que significa que alguém “programou” outro aparelho com seus dados e fez milhares de reais em ligações para a África: perdi meu velho número e tive que mandar *e-mail* para um monte de gente, avisando... tudo porque estamos usando coisas sobre as quais ainda não temos, nem nós nem os que nos fornecem os produtos e serviços, controle efetivo. E isso, claro, sem mencionar quantas vezes meu *laptop* e seus programas falham, todo dia.

O cômico alemão Karl Valentin, autor da frase que dá título a este texto (*Die zukunft war früher auch besser*) estava certo. E a gente até pode explicar porque. Muito pouco do que usamos, hoje, em informática (computação, comunicação e controle), está no estado de maturidade tecnológica e social equivalente aos bujões de gás que, simplíssimos, ainda “pipocam” de verdade aqui e ali. O mundo da informática ainda está em acelerada construção, mudando de uma geração para outra no máximo a cada dois anos, e nos sujeitando a todos os caprichos e demandas de servidores que não conseguem apenas nos servir: nos obrigam a muito mais, a participar intensamente de seu processo de desenvolvimento. Sim, pois quase tudo o que nos cai às mãos, hoje, é digital e “beta”, ou seja, algo melhor do que um protótipo, mas que os fabricantes orgulhosamente chamam de primeira geração. Dez gerações depois, sempre nos perguntamos como é que usávamos aquelas “gambiarras”... quando deveríamos nos lembrar de como a vida teria sido tão pior sem a maioria delas.

Nós, usuários, reclamamos do presente, porque ouvimos promessas de um futuro, no passado, muito melhores do que o que temos hoje. Mas construímos, com fabricantes e provedores, um imaginário do qual queremos participar desde o começo. A visão do futuro é nossa também, até porque somos sujeitos inteligentes e ativos do que usamos. Muitos poucos querem continuar com seus celulares analógicos, nem que seja porque as baterias duram menos. Ninguém quer um carro a carburador. Ou não quer acesso mais rápido, mais cores na tela, mais espaço no disco, tudo o que possa vir a ser associado a qualidade, apesar de quase tudo ser só quantidade. E maiores dificuldades na utilização de tecnologias que não estão prontas. A pergunta do dia, então, é: seriam tais dificuldades maiores mesmo, ou o são apenas na aparência?

No início dos anos 80, na Idade da Pedra da Rede, os *modems* eram imensos e externos aos computadores, cada provedor (BBS) resolvia se configurar de um jeito, e o simples ato de “entrar na rede” (usando Kermit!) era precedido de malabarismos impensáveis hoje, coisa de viciados em rede, inatingíveis para mortais comuns. Isso, lá no começo, para usar 1200 *bits* por segundo e transferir texto em ASCII. Em comparação, qual a importância relativa, hoje, de um roteador que, vez em quando, se “desconfigura”, mas tem uma per-

formance mais de 100 vezes superior por um décimo do preço? Melhorando 1.000 vezes a relação entre o custo e o benefício da minha experiência como usuário?

A relação da humanidade com a tecnologia data do começo da vida inteligente no planeta e nunca nos deixamos dominar por ela; e continuaremos a controlar as próximas gerações de qualquer tipo de tecnologia que venhamos a desenvolver. Mas sempre quisemos mais, sempre estivemos insatisfeitos e sempre tivemos aqueles, dentre nós, mais aptos e alertas aos perigos de termos tecnologia demais e humanidade de menos. É por isso que é possível acreditar que a informática, nesses 50 anos de sua efetiva utilização pela sociedade, vem trazendo, e a cada vez mais gente, desenvolvimento e não, puramente, progresso. E tudo indica que o futuro será, pelo menos, melhor do que o passado...

**NO CAOS, A RECEITA DO SUCESSO** Ao entrar no restaurante, a *hostess* lhe informa, gentil e sorridente, dos resultados de uma pesquisa sobre os trinta mil últimos pedidos realizados na casa. Em números redondos, 28% dos pratos servidos eram o que os fregueses haviam pedido, chegaram no tempo previsto e pelo preço do cardápio. Em 49% dos casos, alguma coisa deu errado no prato, no preço, no prazo ou numa combinação de todos. Teve gente que pediu bacalhau e recebeu tapioca, pelo mesmo preço, tempos depois. Em muitos casos, as discussões quase chegaram às vias de fato; em pelo menos um, o cliente sacou uma 9000S e queria mandar o *chef* direto para o banquete celestial. Enfim, nos outros 23% dos pedidos, os pratos e os clientes não se encontraram nunca; às vezes porque os primeiros demoraram tanto que os clientes desistiram; noutras, porque o *chef* veio à mesa, mais de uma vez, para aumentar o preço; noutras, porque o cliente descobriu que o prato ia chegar, mas seria muito diferente do pedido. Em suma, sua chance de sair dali realmente satisfeito é de 28%; a pergunta que a *hostess* lhe faz é... “Posso levar-lhe à mesa, senhor?” Você iria ou não?

Nunca vi nenhuma recepção assim, em restaurante nenhum. Até porque a vasta maioria deles tem um recorde melhor do que este. Mas não acho que entraria num buraco, por mais chique que fosse, onde minha chance de ser bem servido, comer bem e pagar o combinado fosse uma em quatro. E não acho que haja tantos *gastro-masoquistas* por aí, a ponto de manter restaurantes do tipo acima abertos. Só que os números dizem respeito a projetos de software e são do Standish Group, empresa que realiza, desde 1994, um levantamento bastante sofisticado sobre projetos de software realizados em empresas norte-americanas. O campo é fértil, pois o gasto anual americano em projetos de aplicações de software chega a US\$275 bilhões, com mais de 200 mil projetos em andamento. O Standish Group levantou a história de 30 mil desses projetos desde 1994 e, para alguns deles, realizou estudos detalhados. Os dados e as conclusões revelam o tamanho dos problemas que ainda existem em desenvolvimento de software, bem como o tamanho da oportunidade para quem tentar e conseguir resolvê-los.

A situação vem melhorando, e de forma muito significativa em alguns aspectos. Em 1994, o aumento médio de preço nos projetos chegava a 189%, tendo caído para 45% em 2000. Uma senhora queda, diga-se não tão de passagem assim, que reflete o aumento da preocupação, atenção e da competência mundial em relação ao desenvolvimento de software, que está passando a ser a ferramenta mais básica e fundamental dos processos econômicos e sociais. É quase impossível se passar um dia sem sofrer o efeito de algum tipo de sistema de informação; para nosso bem, portanto, é bom que eles fun-

cionem, a contento, no prazo e no preço que, por exemplo, o supermercado combinou. Senão, sempre pagaremos uma parte do prejuízo, seja em aumento de preços e/ou piores serviços, ou em diminuição de competição, pois software de má qualidade, fora do prazo ou muito caro (ou os três, vez por outra) pode ser falência certa. Um dos casos mais rumorosos pode ter destruído, em 1996, um dos maiores distribuidores de medicamentos dos EUA, a FoxMeyer Drug Co., que faturava US\$5 bilhões por ano. Os liquidantes processaram a SAP e a Andersen Consulting (hoje Accenture), pedindo uma compensação de US\$ 1 bilhão, processo que deverá ser julgado este ano, correndo o risco de criar jurisprudência nos EUA.

Alguns críticos acham que as companhias “não aprendem com fracassos anteriores”, mas o fato é que o estado da prática de construção de software vem mudando, e para muito melhor, nos últimos 15 anos. Segundo dados do mesmo estudo do Standish Group, o atraso médio dos projetos de software caiu de 222% do tempo inicialmente previsto, em 1994, para 63% em 2000; em 1994, apenas 61% da funcionalidade prevista era entregue, em média, número que subiu para 67%. Pode não ser essas coisas todas saber que sua encomenda de software, se estiver na média, custará 45% mais caro do que o contratado, terá um atraso de 63% sobre o prazo previsto para entrega e não terá 33% das funcionalidades encomendadas. Mas há de se convir (compare acima) que houve um progresso gigantesco.

Construir software não é uma atividade simples; o problema que se trata é normalmente o da replicação virtual de processos, sistemas e, na maioria das vezes, instituições e redes delas. Fazer com que o item cujo estoque caiu abaixo do mínimo seja encomendado e reposto na prateleira do supermercado é muito fácil... desde que todos os processos, sistemas e instituições envolvidos sejam individualmente entendidos e se entendam entre si. Na maioria dos casos, é muito difícil, em projetos complexos, de porte e de longo prazo, manter a instituição estática, esperando que o projeto se complete. Na prática, os requisitos que um grande sistema tem que atender vão mudando à medida em que ele vai sendo construído, como se seu pedido para a cozinha do restaurante fosse sendo mudado enquanto está sendo preparado: o *chef* começa fazendo seu sirigado com arroz de frutos do mar e, na hora em que o prato chega, seu pedido já era javali ao barolo. Não dá.

Grandes sistemas são como o que talvez tenha falido a FoxMeyer: a encomenda custou US\$35 milhões. Segundo o Standish Group, a taxa de sucesso para projetos acima de US\$10 milhões, que tipicamente envolvem quinhentas ou mais pessoas por períodos de três anos ou mais, é estatisticamente zero! Para projetos de até US\$750 mil, que tipicamente envolvem seis pessoas por seis meses, a taxa de sucesso é 55%. Para projetos pequenos (“micro-projetos”), que custam até US\$250 mil e duram de três a quatro semanas, taxas de sucesso de 70% podem ser esperadas depois de alguma prática e há razões para se pensar que, havendo um processo tão afinado quanto a cozinha dos melhores restaurantes, as taxas de sucesso possam chegar perto dos 100%.

A receita do Standish Group -Chaos: a recipe for success – (e de outros, como Donald Reifer) para o sucesso inclui envolvimento e apoio dos executivos e usuários, gerentes de projeto experientes, objetivos claros e escopo reduzido. Estes poucos itens, bem cuidados, seriam responsáveis por 70% das chances de sucesso de projetos de software. A receita é mais simples do que parece: gasta muito esforço, envolvimento de todas as pessoas-chave das instituições, educação contínua e implantação de métodos e processos claros e efetivos. A mudança é de longo prazo, mas é para sempre e começa por quem

contrata. Se for você, comece por perguntar, da próxima vez, pelos métodos e processos de quem você vai contratar. Ou então apele para filosofia...

#### SANTA RITA DE CÁSSIA, PADROEIRA DOS SISTEMAS DE INFORMAÇÃO

Mais hora menos hora, é capaz de alguém descobrir que quem derrubou o Space Shuttle Columbia foi o software embarcado na nave. Não que o software, em si, tenha arrancado placas de isolamento térmico ou tocado fogo no Columbia. Mas o “sistema” pode muito bem ter embicado a espaçonave de tal forma, contra a atmosfera terrestre, que a estrutura não resistiu e a coisa deu no que deu. Ou podem descobrir que foi a falta de software que causou o desastre: se (por exemplo) caíram placas cerâmicas, porque tal fato não foi denunciado pelos sistemas de supervisão da nave? Cada placa, se é tão crítica como parece, deveria ser individualmente monitorada e sua ausência ou disfunção deveria, imediatamente, arrepiar cabelos em muitos, da tripulação ao controle de voo, passando pelos fabricantes. Claro que o Columbia pode ter sido atingido por um raio cósmico e o software, afinal, não teria ajudado a derrubar a nave... a menos que... estivesse faltando, no Shuttle um radar para detectar tais fenômenos, que certamente seria em boa parte software, caso em que...

Nas últimas décadas, começamos a informatizar o mundo. Quase todos os sistemas com os quais lidamos no nosso dia-a-dia dependem, fundamentalmente, de hardware e software para seu funcionamento apropriado. Mais ainda, o software que “roda” nesses objetos agrega muito pouco aos mesmos, quando considerado isoladamente. Informatizar uma geladeira, tornando-a capaz de avisar (num monitor, na porta) que o compressor está com problemas, é muito pouco, pois só cria mais um problema na vida do dono, que tem que se preocupar em chamar a assistência, etc, etc. Uma geladeira informatizada de verdade teria que fazer o diagnóstico, entrar em contato com a fábrica, descobrir o que precisa ser feito e por que e, aí, o sistema de informação da fábrica (na verdade, da assistência técnica) entraria em contato com o feliz proprietário de tão gentil objeto, para discutir o melhor horário para uma visita que iria consertar... o que ele nem sabe, ainda, que está quebrado ou para quebrar. Quem detecta o problema, na geladeira, é software e hardware, nela. Quem resolve o problema, para o usuário, é a operação, coordenada, de vários sistemas de informação.

À medida em que tornamo-nos cada vez mais dependentes de serviços muito sofisticados, (que têm que ser muito simples, também) mais crítica é a presença de sistemas de informação nas nossas vidas. Se fôssemos completamente dependentes de serviços informatizados, certamente haveria um mundo virtual, paralelo a este nosso, real, que simularia, através dos seus sistemas de informação, tudo o que precisamos fazer aqui na Terra (e, depois, pelo mundo afora). Nossas necessidades, detectadas lá, resultariam em ações, aqui, resolvendo nossos menores e maiores problemas. Arretado, não? Agora pense na complexidade de “virtualizar” tudo o que fazemos e precisamos ter para fazê-lo. Para começar a discussão, por que não partir do Columbia? Se este tipo de nave, na realidade se o conjunto de processos dos quais o correto funcionamento da nave depende fosse realmente informatizado, até mesmo o desastre da Challenger, que explodiu na decolagem, poderia ter sido evitado. Para quem não se lembra, o primeiro desastre de um Shuttle, em 1986, se deveu a problemas com anéis de vedação nos *boosters*, os foguetes auxiliares do lançador do Shuttle. A NASA tinha dados sobre a erosão dos chamados *O-rings* dos foguetes, que poderiam ter levado à conclusão de que, mais voo menos voo, um *booster* iria explodir. Mas os dados sobre os anéis foram descartados pelo “sistema de informação” da NASA, que não era, para este caso, software, mas os analistas que decidiam se era ou não seguro continuar voando com aquele tipo de problema.

Se houvesse um modelo apropriado da nave, se os dados tivessem sido corretamente inseridos neste modelo, possivelmente teria sido possível descobrir que o risco era muito maior do que o publicado pela NASA antes do acidente.

Enquanto a administração acreditava que a possibilidade de falha era de uma em cem mil, a engenharia tinha razões para crer que entre um e dois, a cada cem lançamentos, poderiam falhar. O famoso adendo ao relatório do comitê que investigou a tragédia, escrito pelo físico Robert Feynman termina por dizer que “...para uma tecnologia de sucesso, a realidade deve ter precedência sobre relações públicas, porque a natureza não pode ser enganada.” Em 1995, a melhor estimativa para falha no Shuttle era de uma a cada 145 missões, o que é um risco muito grande. No adendo ao relatório do primeiro desastre, Feynman comenta a seriedade com a qual o processo de desenvolvimento do software do Shuttle é tratado – o que é verdade desde o início do projeto. No Challenger, no entanto, o software era muito pequeno, cerca de 250 mil linhas de código. Windows XP, que faz funcionar os PCs sobre nossas mesas, tem 50 milhões de linhas de código, 200 vezes mais do que o Shuttle de 1986 investigado por Feynman.

Os grupos que fazem o software do Shuttle estão entre os melhores do mundo e boa parte do seu trabalho envolve negociar demandas e especificações do que o software embarcado no Shuttle tem que fazer e manter a maior parte da miríade de solicitações... fora da espaçonave. Não que vá rodar noutro lugar, na Terra. Fora no sentido de nem ser feita, de jeito nenhum. Para que se garanta a qualidade do que é feito, de forma a minimizar o risco do software, literalmente, derrubar a nave. Desde o princípio do programa, o lançamento e a reentrada na atmosfera terrestre são feitos por software. Mas o que significaria informatizar a nave a ponto de ter sensores e atuadores em todos seus pontos críticos, talvez aliados a software embarcado que pudesse controlar tudo isso e a um simulador de voo virtual, capaz de replicar, em Terra, o voo que o Shuttle estivesse fazendo, lá em cima? Para “informatizar a nave” em condições próximas da realidade de cada voo, a NASA talvez tivesse que usar um Earth Simulator, o mega computador japonês capaz de 35 trilhões de operações por segundo, rodando um modelo computacional do Space Shuttle, e não de qualquer um. Cada um é diferente – os dados do modelo teriam que representar o estado do Columbia, se dele quiséssemos ter uma simulação.

Aí é onde entra Santa Rita de Cássia, que compartilha com São Judas Tadeu o patrocínio das causas impossíveis. O software do Shuttle, desenvolvido sobre um hardware bem entendido, que não muda há 20 anos, tem cerca de 500 mil linhas de código (1/100 do tamanho de Windows XP), escritas por um time de menos de três centenas de pessoas, a um custo de US\$ 750 milhões. Astronômico, se considerarmos as razões entre o tamanho, o tempo e o custo. Razoável, quando se considera a complexidade do projeto e a necessidade de segurança e correção do sistema. Ainda assim, estima-se que, com as ferramentas atuais, se o software do Shuttle tivesse 10 milhões de linhas, haveria cerca de 100 erros críticos, capazes de comprometer o desempenho e a segurança do sistema. O software do Shuttle é um dos mais bem projetados, escritos e controlados do mundo. E, ainda assim, o sistema de informação do qual ele faz parte precisa, a cada segundo, da intercessão de Santa Rita de Cássia, pois o software é muito menos do que o sistema que muitos pensam que o software, sozinho, implementa... Por isso que deveríamos pedir ao Vaticano a nomeação imediata da “Santa das causas impossíveis” para “Padroeira dos sistemas de informação”...

**O SOFTWARE DO SHUTTLE É UM DOS MAIS BEM PROJETADOS, ESCRITOS E CONTROLADOS DO MUNDO**

#### TRABALHO, EMPREGO, SOFTWARE E ECONOMIA DA INFORMAÇÃO

... “Os baixos salários nacionais devem-se à baixa produtividade do país. E não por os patrões serem mais canalhas em Portugal do que noutro sítio qualquer... Os americanos acham que o trabalho é uma riqueza que deve ser multiplicada. Os europeus preferem olhar para o emprego, preservando-o com elevados subsídios aos desempregados. Os portugueses sempre protegeram os que têm emprego mas não querem trabalhar, desprezando os que procuram trabalho e não arranjam emprego”. O trecho acima é parte de um artigo de Sérgio Figueiredo, no *Correio da Manhã*, de Lisboa, em 18/11/2002. Poderia ser sobre o Brasil, a menos das citações a Portugal e aos portugueses.

E tais conversas são ordem do dia, lá, porque Portugal está discutindo uma revisão de sua legislação trabalhista, numa tentativa de criar mais flexibilidade nas contratações, para aquecer o mercado de trabalho nacional. Mas há greves gerais no horizonte: uns parecem não entender que o mercado de trabalho precisa ter liquidez, pessoas indo de um emprego para outro, de forma eficaz e eficiente do ponto de vista dos custos para os empregadores, mantido um conjunto razoável de garantias para os trabalhadores. Os sindicatos não querem nada com isso; e olha que Portugal tem um dos piores cenários de emprego e renda da Europa e poucas chances de melhorar enquanto sua legislação trabalhista for tão senil quanto a brasileira. Nós, obviamente, estamos em pior situação, à falta de uma Europa a nos ajudar, cobrindo custos educacionais e subvencionando nossa improdutividade.

Isso num mundo em que o trabalho braçal, repetitivo, de baixa qualidade, está se extinguindo. Estudos da George Washington University apontam para 10% dos postos de trabalho na indústria em 2015 e para fazendas automáticas em 2020. Pouca surpresa, pois os serviços saíram, no século passado, de um para três quartos das principais economias. E parte significativa dos novos serviços não resulta da clássica ligação entre um capitalista detentor dos meios e um trabalhador explorado por um patrão desalmado, aquele precisando da tutela do Estado e este perseguido quase como se fosse um criminoso.

O trabalho da economia da informação já é muito diferente do que pensam os sindicatos e os governos de países periféricos e suas políticas isoladas, retrógradas e compensatórias. Sistemas opressivos de “proteção” aos empregados (que têm emprego, claro) servem apenas para esconder a incompetência dos estados em assuntos de educação e fomento ao mercado, ao empreendedorismo e à inovação. A falta dos quatro últimos, combinada com a força do primeiro, é exatamente o que se precisa para que continuemos como sempre estivemos: além de altas taxas de desemprego e de poucos empregos de qualidade, manter-nos-ão à beira das revoluções pelas quais vêm passando o mundo moderno, do qual acabamos comprando quase tudo que realmente importa, desde processos triviais até a “carga” tecnológica (expressão usada por Jared Diamond) que não conseguimos, por pura e simples incompetência, desenvolver.

O trabalho da economia da informação é baseado em redes de agregação de valor onde uns se especializam em análise de mercado, outros em marketing e vendas, ainda outros em distribuição, uns em métodos e processos, ou em marcas e reputações e, finalmente, nos que efetivamente realizam um serviço, que pode terminar na construção de objetos físicos em algum lugar do mundo. Como as pessoas não são especialistas em tudo e nenhum negócio tem demanda perene pelas mesmas habilidades, nada seria mais interessante, para to-

dos lados, que houvesse condições para que fossem criados mercados de trabalho reais e dinâmicos. Pode não desejável (ou possível) “testar” redes dinâmicas de trabalho numa economia nacional qualquer, como um todo, mas certamente é viável fazê-lo em desenvolvimento de software e sistemas de informação, uma das áreas onde os trabalhadores não são pobres desprotegidos e onde o Brasil precisa de uma competitividade internacional muito maior.

Fábricas de software vivem de projetos sazonais, que podem necessitar dos mais variados tipos de competências, projeto a projeto. O desenvolvimento de sistemas de informação, por sua vez, depende de métodos, processos, instrumentos e ferramentas usados pelos engenheiros de software que, por sua vez, estão num ou noutro projeto em função de suas competências pessoais. Ninguém encontra especialistas em sistemas embarcados em aviões trabalhando em automação de vendas. As empresas que sobrevivem, na economia de software, são as que combinam infra-estrutura, métodos, processos e pessoas com vendas e serviços, parte de uma complexa receita onde se deve equilibrar qualidade, produtividade, custo e preço. E isto é cada vez mais difícil no Brasil, principalmente face à competição internacional de países como Índia, China e Bangladesh, mas não só. Rússia e Irlanda, entre outros, estão na primeira divisão do campeonato.

Ou começamos a testar, e com urgência, novas redes pessoais e institucionais de produção de software e sistemas de informação em geral, com incentivo e apoio daqueles que hoje trabalham justamente para “enquadrar” as empresas de software como se fossem simples montadoras de produtos físicos ou começaremos, muito em breve, a exportar, em grande escala, trabalho sofisticado e potencialmente bem remunerado. Desenvolvimento de software depende de educação e prática, é um tipo de serviço moderno que pode, com relativa facilidade, ser contratado em qualquer lugar do mundo, desde que se administre, razoavelmente, os processos envolvidos.

A boa notícia é que um conjunto de padrões internacionais para processo de software está começando a ser usado por muitas companhias em muitos países. Regiões onde, por sinal, a disponibilidade de engenheiros de software é significativamente maior do que no Brasil, e a um custo muito, muito menor. A má notícia é que a “globalização” nacional, conduzida no mais das vezes pela exportação do controle de companhias locais, está começando a exportar, também, trabalho intelectual que gera benefícios muito maiores do que empregos e salários. Fábricas de software são criadouros naturais de práticas e processos que formam, sempre, novas empresas ao seu redor e o fazem por muito tempo, mesmo depois que, por uma razão ou por outra, encerram suas atividades. Ou seja, estamos exportando, para a Índia e Rússia, parte da nossa própria capacidade de competir na economia da informação. Ao congelar relações ultrapassadas de emprego cujo trabalho pode ser exportado de forma eficaz e eficiente, como é o caso de desenvolvimento de software, estamos tentando parar o tempo e nos isolando, cada vez mais remotamente, da competição. Talvez fosse hora de experimentar, em software, para ver se conseguimos transformar um rombo (crescente) de um bilhão de dólares na balança comercial em superávit e esperança de dias melhores. Antes que, também em software, sejamos apenas o país do futuro cada vez mais distante, como muitos portugueses também estão começando a se sentir.

**OS ARTIGOS E AUTORES DESTA NÚCLEO TEMÁTICO** O primeiro artigo, de Jorge Fernandes, trata da prática de software, desde sua concepção até o uso. O status epistemológico de software, claro, não é um dos aspectos já resolvidos pelo conjunto de teorias que trata do problema; os leitores familiares com software e com seu ciclo de vida poderão se surpreender com o tipo de abordagem usa-

do no artigo, muito mais para holístico (ou sistêmico) do que para tecnológico. Fernandes, doutor em informática pela UFPE, é graduado em biologia pela UFRN (onde é professor) e uma das mais coerentes e criativas das vozes dissonantes do *status quo* de pesquisa e desenvolvimento de software no país.

O segundo texto, de Ana Cavalcanti, trata de refinamento, um dos poucos aspectos do que poderia chamar uma “teoria geral para o desenvolvimento de software” que tem seu rigor e formalismo mais ou menos resolvidos e utilizáveis na prática -isso porque boa parte das “teorias” para desenvolvimento de software não consegue passar no teste prático de agregar qualquer valor ao desenvolvimento de sistemas reais. Refinamento é o correspondente teórico metodológico ao “bom senso” e às “regras do polegar” que são usadas por programadores no seu dia-a-dia e é exatamente seu sucesso, como teoria, que nos permite imaginar que, com o tempo, um número cada vez maior de facetas do ciclo de vida de software possa ser contemplado com fundamentos teóricos-metodológicos práticos. Ana Cavalcanti é PhD pela University of Oxford e leciona, hoje, na University of Kent at Canterbury. José Palazzo Oliveira é doutor em informática pelo Institut National Polytechnique de Grenoble e professor titular da UFRGS e, em seu artigo, trata do papel dos sistemas de informação na sociedade, incluindo uma discussão sobre os “incluídos e excluídos” do processo de informatização da economia e, conseqüentemente, do todo social. As possibilidades e os desafios da nova era, habilitada por software em quase todo canto, são o tema central do discurso de Palazzo, que trata do tema no sentido amplo, onde sistemas de informação têm software na sua base mas são muito mais que software no seu todo.

Eratóstenes Araújo, por sua vez, discute as oportunidades para o Brasil no setor de software, tanto do ponto de vista da capacitação e do mercado locais como da demanda e das exigências do mundo. Araújo, mestre em informática pelo IME e coordenador da área de capacitação empresarial e empreendedorismo da sociedade Sofrex, trata há mais de uma década dos problemas de competitividade das empresas nacionais de software e serviços de informática, e nos dá, em seu texto, uma contribuição fundamental para o entendimento dos problemas do setor de software no país, hoje.

Finalizando, o artigo de Vanderlei Perez Canhos, diretor do Centro de Referência em Informação Ambiental (CRIA), de Campinas, SP, oferece uma visão geral da evolução da informática para biodiversidade, enquanto o texto de Renato Toi, diretor da empresa VentureLabs e consultor do Softex, trata do empreendedorismo em software.

Nossa intenção, neste núcleo temático, foi a de aliar visões sobre a prática e a teoria associadas ao ciclo de vida de software, sobre os sistemas resultantes na economia e sobre as possibilidades, para software e para o Brasil, na economia mundial de software. O futuro pode não ser melhor do que era no passado, mas uma coisa é certa: quase nada, nele, existirá sem software. O novo sistema operacional do mundo, baseado em redes, na ubiqüidade da comunicação e na pervasividade das interações mediadas por sistemas de informação, vem sendo escrito há cinquenta anos e levará o restante da história da humanidade para ficar pronto. À medida em que digitalizamos não só nossas transações comerciais mas quase todo o patrimônio histórico, artístico e cultural do planeta, pode ser que o legado final da Terra não seja, apenas, a diáspora definitiva, mas a codificação, executável, de quase tudo o que aconteceu desde que computadores e software apareceram para mudar, de uma vez por todas, o que nos somos e como nos entendemos.

*Silvio Lemos Meira é professor titular de engenharia de software da Universidade Federal de Pernambuco e cientista-chefe do Centro de Estudos e Sistemas Avançados do Recife (Cesar).*