

Embora imprescindível, a TI, Tecnologia da Informação, não é tudo!



GLÁUCIO BRANDÃO

Nessa **Aula Condensada de nº 39** (obrigado Prof. Severo pelo termo!), convido os leitores a uma reflexão sobre uma das principais ferramentas que trouxeram o Mundo ao patamar evolutivo em que se encontra hoje: a Tecnologia da Informação e das Comunicações, ou simplesmente TIC! A principal é o **Algoritmo!** Como assim? Explico ao longo deste texto *longo!*

Em nosso contexto, e de modo amplo, a TIC será utilizada para indicar toda a tecnologia que envolva hardware e software, massa ou energia (coloquei estas duas últimas palavras pra dar uma ar mais científico, mais charmoso) usados para processar, armazenar e transportar informação em forma digital! Utilizarei, ao invés de TIC, seu apelido mais popular, TI.

Dou o pontapé inicial à reflexão sobre TI com a seguinte, e não muito simpática, sentença:

“Traga-me uma ideia, e encontro um(a) desenvolvedor(a) pra fazer a ideia andar por conta própria. Traga-me um(a) desenvolvedora(a), e não saberei o que fazer com ele(a)”.
(Gláucio-San, MMXIX)

Vou cortar os “(a)” de todo o texto para não ficar enfadonho. Daqui pra frente, assumam que tudo se aplica a todos os gêneros.

Meu “testemunho” vai ancorado em uma experiência de mais de 30 anos programando em *assembly*. Para quem não entende o que essa palavra significa, o Mundo foi construído em linguagem *assembly*; depois veio a **C** e sua irmã vitaminada, a linguagem C++, chegada a objetos. Minha capacidade de programação parou aqui. Na sequência, a cosmopolita tomadora de café que anda em qualquer terra, a Java. A coisa emborcou para a WEB e fez nascer a dinâmica PHP e a Ruby, seguidora de scripts, finalizando na linguagem universal atualmente mais aceita: o *Facebook!*

Olhando para trás, penso hoje na TI como a extensão de uma ideia, como seu complemento. Pode ser enquadrada como uma grande ferramenta de presente e futuro, mas não o próprio futuro em si... e nem o presente. Aqui que a confusão começa.

Hoje TI não é mais inovação, não é fim, porém um meio de se chegar a ela. Uma *Commodity* - explico mais adiante. Tenha uma ideia, tenha um tema, e conseguiremos “arranjar” recursos em um ecossistema para dar corpo a ela. Tenha apenas desenvolvedores, os famosos e raros DEV, e tenha somente mão de obra custosa e qualificada. Necessária, claro, mas se não tiver um rumo orientado à inovação, torna-se-ão desenvolvedores de serviço, centros de TI sem temas, sem estratégias, apenas locais especializados em ferramentas. Sem propósito definido que guie estes centros, tenderão a se transformar em estruturas a-temáticas (Esse termo existe?). E, como qualquer empreendedor aqui sabe, todo planejamento estratégico é guiado pela missão do negócio. Sem ela, a sustentabilidade passa a ser a coisa mais precíval.

Ideias, Algoritmos, Linhas de Código e Programação

Diferentemente do que se pensa, todos sabemos desenvolver algoritmos. Pegando o conceito na WEB, conceito o qual atesto, dada à minha cátedra inicial na carreira do magistério e pelos quatro anos seguintes, pode-se confiar:

“Conjunto das regras e procedimentos lógicos perfeitamente definidos que levam à solução de um problema em um número finito de etapas”
(Dicionário online)

Assim, todos nós temos condições de “algoritmizar”, de criar receitas. Dado um problema qualquer, o algoritmo é a resposta natural a ele. Exemplo: crie o algoritmo [Escovar os Dentes]!

Algoritmo [Escovar os Dentes]
<ol style="list-style-type: none">1. Início (as condições de início incluem, por exemplo, estar acordado, ter pasta e escova)2. Pegar escova3. Pegar pasta4. Colocar pasta na escova5. Molhar conjunto pasta-escova6. Abrir a boca7. Levar conjunto pasta-escova à boca8. Fazer movimentos ascendentes, descendentes e laterais no entorno do dente9. Passou por todos? Se sim, vá ao passo 10. Se não, vá ao passo 8.10. Retire o conjunto pasta-escova da boca11. Enxague a boca12. Fim (aqui as condições de finalização como, por exemplo, lavar e guardar a escova)

Fonte: eu mesmo, todos os dias, por volta das 5h00.

De uma forma muito parecida, a jovem cientista e engenheira elétrica [Katie Bouman](#), desenvolveu o algoritmo (repetindo, a-l-g-o-r-i-t-m-o) capaz de "tirar" uma foto de um buraco negro, 100 anos depois do cientista Albert Einstein, *que não programava nem sabia o que era TI*, ter previsto sua aparência. (Por questões de espaço, e somente por isto, não vou fazer aqui o algoritmo de minha colega!). As linguagens de programação que a Cientista Katie Bouman utilizou não diferem das que utilizamos em problemas físico-matemático aqui

na UFRN. Podem perguntar ao professor Marcelo Nogueira (ECT-UFRN), que tem uma formação muito parecida com a dela!

Traduzir o algoritmo [Escovar os Dentes] em linha de código é o que chamamos de codificação ou programação. Acreditem em mim: não é difícil! Pode ser feita em qualquer linguagem das que citei no começo do texto. Já existem ambientes que aceitam codificação por voz e até conversam com você. O [Cloudia](#) é um exemplo. Em breve, os programadores serão uma classe extinta. Ou seja: quem conseguir pensar vai conseguir programar. Simples assim!



Katie Bouman: aos 29 anos, ela liderou uma equipe de 200 cientistas (MIT/Divulgação)

A sequência é então essa:

Problema - Ideia - Algoritmo - Codificação - Programa (ou App)

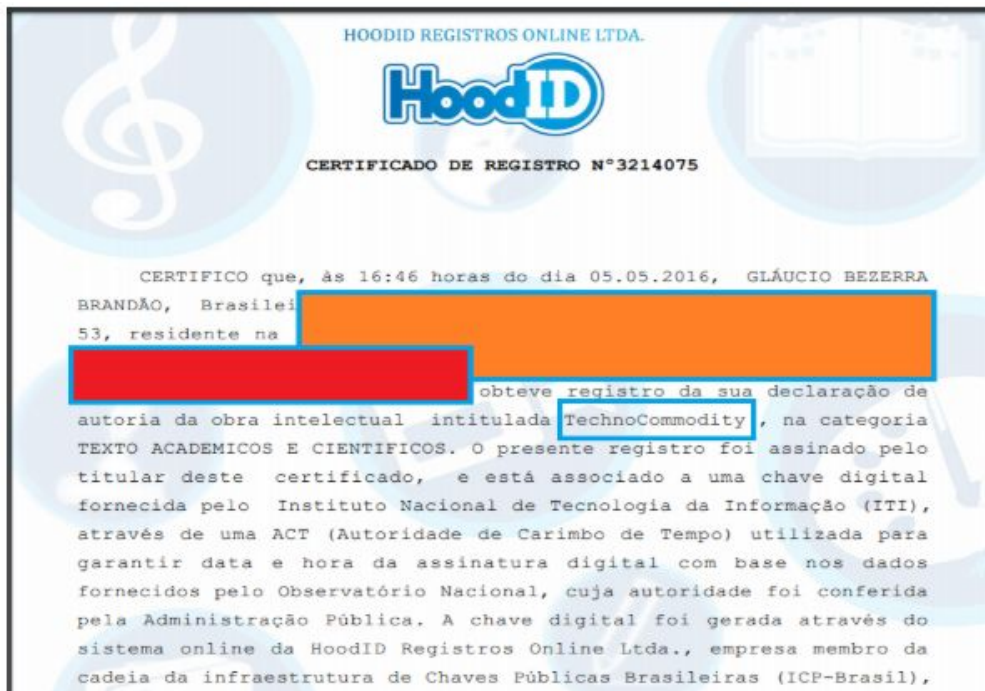
E aí faço o link com o título: a codificação vem depois do algoritmo! Portanto, quem liga a ferramenta à ideia, ao tema, é o algoritmo. Não faz sentido programar se não há um algoritmo, produto de um pensamento, de uma ideia, de um propósito. Um programa (App) sem algoritmo é a-temático (não achei esse termo ainda na Internet!). Levado esse pensamento à grande estrutura, desenvolver centros especializados puramente em TI não garantirá sustentabilidade. Exemplos não nos faltam. Vamos pegar um vizinho. O [Porto Digital \(Recife\)](#) começou como um parque tecnológico em TIC e ampliou para a Economia Criativa (EC). Fazia sentido no começo, pois havia uma demanda reprimida. Assim que começou a perder “sustância” implementou o [Portomídia](#) que, ao ser “exportado” também para Caruaru e Petrolina mudaram em menos de cinco anos ambas realidades locais. Em Caruaru ganhou o nome de [Armazém da Criatividade](#) transformando minha cidadizinha (emoção!) no 2º maior polo de EC de Pernambuco, perdendo apenas para o próprio imenarrável Recife.

O Vale do Silício já começou criativo: as soluções em TI vieram na esteira de problemas de hardware e portabilidade (dos PCs), educação, comunicação (Internet), robótica, medicina, questões bélicas etc. A TI foi a ferramenta da “consequência”, não da causa! Geralmente os

problemas estão no algoritmo, não na codificação. Sendo transversal, poderia ser vista como um commodity. Por que não?

Porque não pensar a TI como uma espécie de commodity tecnológica, uma TechnoCommodity?

Em maio de 2016, sem nada pra fazer, comecei a “serendipar” em torno da tríplice hélice: “Por que não roda aqui? Será que posso desenvolver uma ferramenta de medição das forças que deveriam unir as três esferas da inovação prescritas por Henry Etzkowitz?”. Certo de que essas forças deveriam ser medidas em termos de artefatos tecnológicos, e de que o “sangue” que alimenta a inovação é, indiscutivelmente, a TI, pensei: “Se a TI é o catalisador da inovação e ela, a TI, está tão consolidada no globo, deve ser tratada como uma espécie de *commodity*. Sendo a TI tecnologia e *commodity*, deve haver um termo que reúna essas palavras. Vou “Googlar”? Passei os dedos no teclado em busca dessa palavra híbrida e nada encontrei. Pensei: “Será possível que o Google não consegue achar essa palavra?”.



Documento de registro da TechnoCommodity, ou TC.

Busquei no Yahoo, Bing, DuckDuckGo, StartPage, Qwant, SearX, Peekier, Firefox e nada! Decidi: “É minha. Ninguém tasca!”. Não acreditava que poderia ter inventado uma palavra que fazia algum sentido (Acho que vou registrar *a-temático*?!). Muito agitado, passei a descrevê-la:

“Definimos este elemento como TechnoCommodity ou TC, formado pela justaposição das palavras provenientes da língua inglesa Techno, contração de Technology, e Commodity, cujos conceitos individuais são conhecidos. A TC tem por atributos:

1. Atomicidade: é indivisível e pontual.
2. Portabilidade: uma mesma TC deve ser de fácil aplicação a diferentes contextos.

3. Escalabilidade: quando houver necessidade de replicação, esta será de baixo custo.
4. Inflexibilidade: cada TC é única!
5. Percibilidade: a tecnologia é altamente mutável, o que faz com que a TC herde esta característica.
6. Proposta de Valor Contextual. Cada TC deve ser negociada conforme a aplicação.
7. Ser de base tecnológica, desde baixa a alta intensidade.
8. Resolutiva: tem de resolver um problema ou parte dele.
9. Toda TC tem de ser do bem.”

Tem mais coisa: variantes, instâncias, modo de operação etc., mas isto é papo para outro dia!

Entrei na plataforma de meu “cumpadi” [Valdecir “Hoodld” Fontes](#) e fiz o registro com um bônus que ele havia me dado (Foi mal, brother. Achou que eu não usaria, né?). Aproveitei para depositar minha ideia “estupenda” na Wikipedia, registro que caiu por falta de manutenção. Mas a TC voltará com força total na plataforma [IncaaS](#), pois servirá de criptomoeda para o circuito de jogos. Aguardem!

Assim, de hoje em diante, trato a TI como uma espécie de commodity (não ousou falar oficialmente em TechnoCommodity, pois carece de aprovação de pares e certificadores), já que hardware, linguagens, a codificação, o exercício de programação, são métodos padrão, diferindo em poucos aspectos. O que muda são os algoritmos, as ideias, o que reforça meu pensamento dela, a TI, como ferramenta e não fim.

Sendo a TI uma commodity, o que têm-se que ser feito então?

Investir em gente; em educação criativa, empreendedora; em **educação algorítmica!!!** Do mesmo jeito que a formação em TI deveria ser obrigatória em todos os cursos de todas as áreas do conhecimento, a formação em algoritmos também. Algoritmos e TI deveriam ser oferecidos em disciplinas separadas. O Algoritmo cola a ideia às linhas de código! Algoritmo é criatividade, planejamento. TI é execução. A TI não é tudo: ela automatiza o algoritmo! Quem você acha que vem antes de quem?

As instituições de capacitação têm de flexibilizar suas grades (são prisões mesmo) disciplinares, de modo a permitirem a entrada do ar da inovação, coisa que já bati aqui em [O que falta aos cursos das Sociais Aplicadas e Engenharias?](#) Quer mudar a Educação, pai, mãe, gestor, autoridade competente ou incompetente? Quer dar um tema à sua software house? Então comecem pelos algoritmos. Depois falamos em TI! (emoji com olho piscando aqui, rsrs). Sua avó já fazia algoritmos de bolo antes de você nascer! Refazendo a frase lá de cima, do “filósofo” medieval mais alto do vale da inPACTA:

*“Traga-me um algoritmo, e encontro um desenvolvedor pra convertê-lo em linhas de código.
Traga-me um desenvolvedor, e não saberei o que fazer com ele”.*
(Gláucio-San, MMXIX, Versão 2.0)