

## 50 tons de estimativa: software empresarial e o tesão do inventor

05/07/2016 10:30

Antes de mais nada: nada nesse *post* é lei, é apenas minha opinião. (E não tem nada de estimativa também, foi mal.)

Algum tempo atrás, quando estávamos indo buscar ingressos prum *show* do [ÁIRO MEIDI](#), o Maurício e eu estávamos conversando sobre desenvolvimento de *software*. O Maurício então soltou a seguinte dúvida:

*Porque será que o pessoal que desenvolve ABAP não gosta de verdade de desenvolvimento de software?*

Isso é uma pergunta difícil de responder, porque não existe um motivo único. Mas a premissa é verdadeira. Tem muita gente que trabalha com ABAP que não gosta realmente de desenvolvimento de *software*.

Permitam-me voltar brevemente no tempo pra contar uma historinha.



YOU'RE ABOUT TO HACK TIME, ARE YOU SURE? >YES NO

No Ano de Nosso Senhor de 2007, eu ainda estava na faculdade. Nessa época, a galera se juntava e ia em massa em vários eventos de empresas de tecnologia, com o óbvio objetivo principal de comparecer aos *coffee breaks* pra comer MUITO e DE GRAÇA, e talvez assistir uma palestra ou outra.

Nesse ano, a Microsoft estava lançando a versão nova do Office, cheia de coisas novas como o padrão aberto de XML dos documentos, o *design* reformulado, aquele esquema de colaboração com o Sharepoint, etc. E uma das palestras do evento era exatamente sobre isso, e foi muito bacana. Só que logo depois dessa palestra, vinha uma outra de um cara falando sobre o [XNA](#), o *framework* gratuito para desenvolvimento de jogos da Microsoft. O palestrante (cujo nome eu infelizmente não consigo lembrar) começou sua apresentação dizendo mais ou menos o seguinte:

*Ó, quem gosta de ficar fazendo contas a pagar, a receber, balanço, fluxo de caixa, pode ir embora, que agora a gente vai falar de coisas divertidas.*

E seguiu a palestra mostrando as novidades do *framework*, inclusive com um joguinho onde uma nave na forma do X do Xbox destruía inimigos que eram pinguins do Linux. Na hora foi tudo bem engraçado e tal, mas refletindo sobre isso um tempo depois eu percebi que o apresentador falou indiretamente como brincadeira uma coisa que é séria e REAL:

## Desenvolvimento de *software* empresarial não é divertido

Não é mesmo. Na maior parte do tempo, não vai ser prazeroso nem recompensador. Esse negócio de “*enterprise geek*” pra mim é papinho de currículo. Mas antes de dizer o porquê, eu preciso fazer uma distinção clara entre desenvolvimento de *software* empresarial e desenvolvimento de produtos de *software*:

- O que eu estou chamando aqui de **desenvolvimento de *software* empresarial** é o conjunto de atividades que compreende a implementação, ampliação e manutenção de sistemas distribuídos de *software* que suportam as atividades de uma empresa, como por exemplo um sistema ERP como o SAP;
- e o que eu chamo de **desenvolvimento de produtos de *software*** é o conjunto de atividades que têm por fim a entrega de um ou mais produto(s) de *software* específico(s), junto com manutenção e suporte respectivos, contínuos durante o seu ciclo de vida – pense em um aplicativo Android ou iOS.

É claro que existem alguns pontos onde essas áreas se interceptam. É possível que dentro de uma empresa exista uma pessoa ou um time exclusivamente responsável por uma determinada aplicação, *framework* ou API, ou que um produto seja tão grande ou tão importante que o seu ciclo de desenvolvimento se torne uma atividade rotineira ou até mesmo crítica pra empresa.

Com essa definição, dá pra ver que o foco ou fim de cada área é diferente também: no desenvolvimento de *software* empresarial, o foco é **manter a empresa funcionando**. Já quando você desenvolve um produto, a ideia principal é **entregar o melhor produto possível**.

Esses dois focos nascem de jeitos diferentes. Se a ideia do desenvolvimento é suportar a atividade da empresa, o *software* que vai fazer isso acaba sendo orientado por **requisitos** do negócio. É aquele tipo de coisa que chega na forma de um fiscal da empresa dizendo num e-mail algo do tipo: “*P\*\*\* QUE PARIU eu preciso URGENTEMENTE que o XML da Nota Fiscal tenha esses 20 campos novos PRA ONTEM. A cada dia que eu fico sem mandar essas informações pra Sefaz eu estou tomando uma multa de TROCENTOS MIL REAIS. ME AJUDA POR TUDO QUE VOCÊ CONSIDERA DE MAIS SAGRADO. Atenciosamente, Fiscal*”. É a necessidade do negócio ali, traduzida às vezes em desespero, talvez em especificação, que motiva esse tipo de desenvolvimento. Um trabalho de formiga, que não tem nem pretende ter fim em vista. Recebeu o chamado, desenvolveu, testou, usuário aprovou, próximo chamado, repete, segue o jogo.

Mas se o resultado buscado é um produto, a coisa muda de figura. O produto de *software* geralmente nasce de uma **ideia**. É aquilo que acontece num *brainstorm*, onde se diz coisas do tipo “*Eu tava pensando aqui e CARAMBA CARA imagina se no celular do vendedor ali no ponto de venda tivesse um aplicativo que consultasse a base de produtos e calculasse ali na hora um preço ótimo baseado no histórico de pagamentos e de variação de preços? Não seria bacana? Aí já dispara um e-mail que...*” e assim vai. Isso evolui, vira um projeto, monta-se um time, decide-se por uma arquitetura específica, escolhe-se as melhores ferramentas, o trabalho braçal acontece, testes, implementação, manutenção, *feedback*, *features* novas, versões novas, etc. Até que venha a próxima ideia.

Por que *software* empresarial não é divertido então?

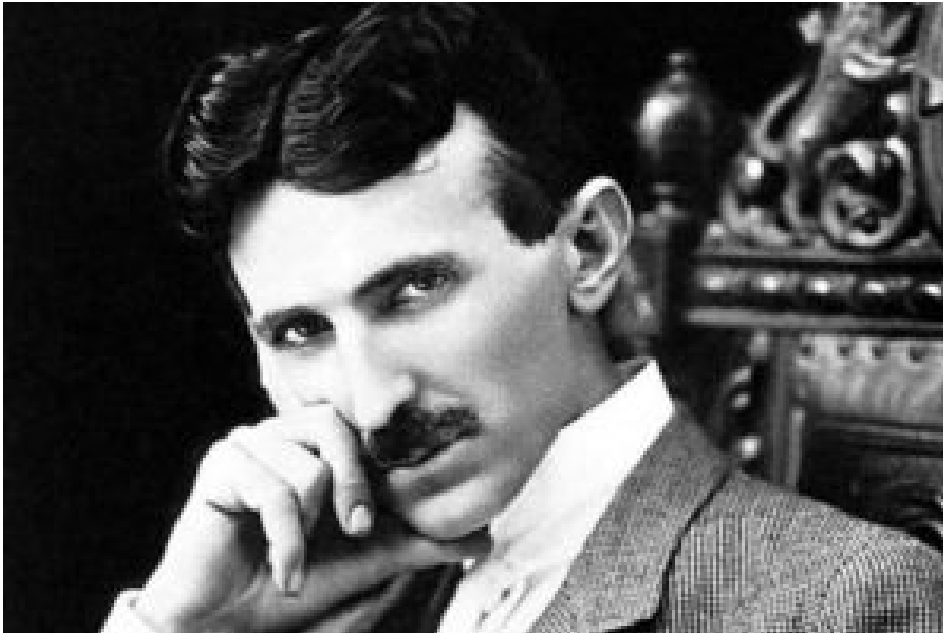
Porque falta algo simples: o **envolvimento**. No exemplo do fiscal, o desenvolvedor que vai atender o chamado normalmente não tem relação NENHUMA com a necessidade da empresa. Pra esse desenvolvedor, pessoalmente, pouco interessa se a Sefaz rejeita uma, duas a cada dez, ou todas as notas do cliente. Não foi ele quem criou as necessidades, não foi ele quem introduziu os erros, ele só vai ter que corrigir essa falta. Uma vez corrigida, sobra um e-mail de agradecimento e um tapinha nas costas, ambos perdidos em mais uma montanha de chamados novos. Até que venha outra nota técnica e tudo se repita.

No mundo dos produtos de *software*, o envolvimento é muito maior com cada entrega. Existem muito mais decisões a serem tomadas, concernentes a arquitetura, paradigmas, segurança, bibliotecas, documentação, etc., fazendo com que o desenvolvedor se sinta parte de um processo e tenha uma visão mais geral do que está sendo feito. Com esse

panorama, cada alteração é sentida pelos seus efeitos nos testes, e cria-se uma motivação para entregar código cada vez mais bem feito.

---

## O tédio do inventor



"gata vamo lá em casa pra eu te mostrar minha energia ilimitada"

Pra mim, quem gosta MESMO de desenvolver *software* tem um tipo de motivação que eu chamo de **tesão do inventor** (deve ter algum nome mais apropriado pra isso, mas eu não conheço). É aquele sentimento que surge depois de você ter quebrado a cabeça usando todo tipo de técnicas, desde as mais simples até as mais complicadas, pra no final alcançar um resultado que você pode apontar felicidade da vida e dizer pra alguém *"MALUCO OLHA ISSO AQUI. SÉRIO. OLHA ISSO. VÊ SE FUNCIONA. FUNCIONOU? P\*\*\* QUE PARIU. SÉRIO. NÃO TO ACREDITANDO. FUNCIONA MESMO"*.

E isso só existe de verdade quando você tem a chance de ter um envolvimento grande com o que você faz. Você **vê o seu trabalho realizado** ali, funcionando, fazendo o que deveria.

Mais de uma vez eu me vi pensando o seguinte: eu queria muito poder, de vez em quando, pegar alguma demanda que eu desenvolvi do zero em ABAP – seja um relatório ALV, um Smartform, um Workflow, sei lá, qualquer uma – e sentar do lado do usuário que vai usar aquilo e dizer: *"mostra pra mim em produção como era o seu trabalho antes disso e como você vai usar isso pra facilitar o seu trabalho daqui em diante"*. Eu acho que eu NUNCA consegui fazer isso. E isso é uma das decepções do meu trabalho como ABAP. E também ajuda a responder a pergunta do Maurício.

Imagina alguém chegando do lado de um desenvolvedor que, por exemplo, tá descobrindo JavaScript, em estado de graça vendo coisas incríveis que o pessoal consegue fazer como o [BabylonJS](#), e diz *"Putz, que legal, você mexe com JavaScript! Bem loco isso aí, parabéns! Agora faz um favor, pega o UI5 aqui e cria um relatório de 10 colunas pra semana que vem, pode ser? Pessoal tá esperando. Mande um e-mail com os detalhes. Valeu!"* Talvez esse primeiro projeto seja muito divertido, mas conforme as coisas vão surgindo no dia-a-dia, a atividade burocrática do desenvolvimento para a empresa vai corroendo lentamente o interesse com aquela ferramenta.

Ainda assim, o JavaScript tem um universo muito amplo de aplicações, e pode ser que aquela vontade de criar continue nos projetos pessoais/paralelos. Mas o ABAP não é tão flexível assim. A motivação inicial logo se perde no mar de relatórios ALV que são entregues todos os dias.

Eu não estou dizendo aqui que é impossível extrair algum prazer do trabalho que se faz com ABAP e/ou com SAP. Os projetos, em certa medida, te dão um pouco disso em cada “entregável” (um termo ridículo, mas consegue ser melhor do que um outro que eu já vi em anúncios de vaga, *delivrável*), principalmente se você está envolvido no planejamento. Mas no fim do dia, o que sobra de verdade é mais uma passageira sensação de alívio pelo dever cumprido em vez do duradouro orgulho de ter criado alguma coisa nova. E se você trabalha como funcionário de uma empresa, ou mesmo alocado há muito tempo no mesmo cliente, é possível que você seja pessoalmente responsável por alguma ou mesmo todas as suas entregas e conheça de cabo a rabo cada implementação ou mesmo toda a arquitetura, e talvez seja capaz de sugerir melhorias ou mudanças. Mas, infelizmente, essas são exceções.

Some à essa conjuntura toda o fato de que a gente trabalha com uma linguagem de quarta geração que ficou estagnada por vários anos e você terá o ambiente perfeito pra deixar o pessoal *dev* totalmente entediado (“broxado”, talvez).

---

## Mas não é o fim do mundo

Eu vou tentar acabar um *post* pela primeira vez aqui dando uma imagem positiva. Mas, ainda assim, não se engane: *software* empresarial vai continuar sendo chato de desenvolver.

Faz algum tempo que a SAP está mudando. Eu sei que é difícil de perceber isso no dia-a-dia (inclusive pra gente aqui do ABAP Zombie, como a gente já disse inclusive no nosso [podcast](#)), mas é verdade. E não importa qual tipo de desenvolvimento você faz: a outra coisa que motiva quem gosta de desenvolvimento de *software* é **tecnologia nova**. Não que essas novidades da SAP estejam usando tecnologias lançadas ONTEM, mas estão as usando em um contexto novo, que é o contexto das aplicações SAP.

E a incorporação desses elementos novos está abrindo espaço para que os problemas sejam resolvidos de formas diferentes, usando jeitos diferentes de pensar. Aquelas interfaces de PI que alguma aplicação usa podem virar um serviço no Gateway. Aqueles BSPs e WebDynpros antigos podem virar aplicações UI5. Aquele desenvolvimento de Workflow que manda um monte de *e-mail* pode virar uma aplicação com *websockets* e ABAP Channel. E por aí vai.

Não é como se ninguém mais estivesse pedindo por ALVs e *module pools* novos hoje em dia (como já vi gente mais entusiasmada dizendo por aí) – e pode ter certeza que esses pedidos ainda vão vir por um BOM tempo – mas as coisas novas estão batendo na porta. É uma boa hora pra saber como usá-las.

Abraços pra quem está implantando BSP *standard* com HTMLB em 2016.

E aprendam JavaScript.

---

## Comentários

Lucila — 22/06/2018 14:21

Perfeito, resumiu meu sentimento nesses 11 anos de desenv. ABAP que tenho. Já tive que ouvir muita gente falando para eu largar o “tequiquês” que em SAP você tem que se envolver com o negócio, que um bom profissional não é aquele que se busca novas tecnologias e sim o cara que se envolve com o negócio. E quando você vai atrás de um treinamento novo ou coisa assim, seus argumentos ficam desmoralizados, pois para que você quer aprender coisa nova se as demandas para a consultoria são só NF-e e o de sempre, tem sempre aquela conversa “ah isso aí vai demorar para as empresas do Brasil solicitarem, não perde tempo”. Sim, eu concordo que seja muito importante se envolver com o negócio, mas daí a não ter interesse por

tecnologias novas e tudo mais... isso cansa um pouco no mundo SAP, mas como você disse no fim da publicação, as coisas estão mudando.

**Leo Schmidt** — 22/06/2018 15:15

Valeu Lucila!

Eu sempre acho engraçado quando alguém tenta segregar o desenvolvimento ABAP do “tequiquês”, como se o ABAP fosse uma ferramenta mágica de outra dimensão que existisse somente pra satisfazer as necessidades de negócio, ou como se nunca fossem surgir demandas por soluções com tecnologias novas. Esse tipo de pensamento fica cada dia mais errado, porque além de todas as inovações da SAP o ABAP (impressionantemente) ainda continua evoluindo. Mas no fim das contas é o que o pessoal comentou aqui, isso é coisa de quem gosta de ficar encostado.

**willi** — 22/07/2016 14:21

Ae cara concordo com tdo que vc disse... Eu mesmo fui desenvolvedor Java, C e Python no passado e agora sou um mero Funcional que as vezes da uns paranaue... eu mesmo já vi muitos desenvolvedores SAP “Abaper” falando que se dentem limitados com o ABAP e caramba 4, porem estes mesmos desenvolvedores não sabem nem programar orientado objeto ainda... não procuram evoluir... hoje a SAP tem uma grande gama de produtos ta ai a família Hybris com tdo, Java Spring, Java, Abap melhorado, Fiori, varios SDK bacanas e mesmo assim eles não procuram evoluir.....

Ufa desabafeiiiiii kkkk

Ah seu top ficou muito topppp Parabéns

**Leo Schmidt** — 22/07/2016 15:11

Valeu Willi!

O Maurício [já comentou aqui no blog](#) sobre esse lance do pessoal estar despreparado, e eu acho que o que ele escreveu ali ainda é válido. Nesse post eu tentei enxergar esse problema de outro ângulo, vendo mais o perfil da galera que trabalha com ABAP do que propriamente a motivação que existe nesse pessoal pra continuar aprendendo.

Mas é verdade, pra reclamar e continuar no lugar se gasta menos energia do que pra aprender e evoluir.

**Fabio Marques** — 12/07/2016 13:53

Parabéns pelo post, é a pura verdade.

**Leo Schmidt** — 12/07/2016 14:04

Valeu Fabio!

**Ricardo** — 06/07/2016 21:09

"É a necessidade do negócio ali, traduzida às vezes em desespero, talvez em especificação, que motiva esse tipo de desenvolvimento. "

É um poeta...

Eu trabalho com tecnologia, com infra e sei bem qual é este sentimento: no final de um projeto temos algo: uma nova VLAN, uma rede WI-FI, etc algo palpável, infra por outro lado é o aspira não deixando a viatura do comandante parar: não há satisfação porque não há "produto", só chamado, um atrás do outro.

Enfim, tamo junto e mais uma vez um ótimo post.

**Leo Schmidt** — 08/07/2016 18:17

Valeu Ricardo!

Cara, quase fui pra área de redes, acho muito bacana. Mas imagino que realmente sofra do mesmo problema.

**Felipe** — 06/07/2016 13:36

Muito bacana o post! Parabéns e compartilho da sua visão, infelizmente em alguns pontos mas felizmente em outros.

"Abraços pra quem está implantando BSP standard com HTMLB em 2016." – kkkkk

Eu tb #aindaFaçoALV ... Abs!

**Leo Schmidt** — 06/07/2016 18:39

Valeu Felipe!

HTMLB = How To Make Lots of Bullsh\*\*

**Erick Carvalho** — 05/07/2016 19:10

Muito bom o post Leo,

Aqui temos a velha e importante discussão Codificador x Desenvolvedor, é muito complicado e difícil de acreditar que ainda existem muitos profissionais reforçando a teoria X onde profissionais possuem aversão ao trabalho e encaram como um mal necessário para ganhar dinheiro.

Já desenvolvi em XNA inclusive participando do Imagine Cup e hoje sou apenas mais um ABAPer criando produtos, mas posso garantir que ainda me divirto tanto quanto naquela época, minha opinião sobre o que ainda faz existir tantos profissionais reforçando a teoria X é o simples fato da cultura organizacional existente, que ao contrário do que muitos pensam, a cultura da sua empresa é feito por você que trabalha lá.

Acredito de verdade que se começarmos a mudar a nossa cultura e sermos mais pragmáticos todo mundo ganha.

Parabéns pelo Post novamente e sim #aindaFaçoALV

**Felipe Oliveira** — 06/07/2016 18:20



Concordo com você Eric, e entendo quando você fala de cultura organizacional, por todas as consultorias que passei raramente e agora não me lembro qual foi a última vez que um companheiro abaper me indicou um evento, palestra ou um simples livro da área pra ler, isso não deveria depender da empresa exigir para acontecer,

parabéns pelo post e #aindaFacoAlv

**Leo Schmidt** — 06/07/2016 09:33

Valeu Erick! Aliás, muito bom o [Bar8](#)!

Confesso que nunca tinha ouvido falar de teoria X / teoria Y antes, então posso estar falando groselha. Mas, realmente, se a única motivação é receber no quinto dia útil, o resultado é ruim tanto pro profissional quanto pro negócio. Porém, o lance da cultura ser mais aberta não é perfeito também, porque se a empresa abraça a teoria Y com força, o problema é diferente: o funcionário acaba se autoescravizando, como a gente vê acontecendo em agência de publicidade e nessas empresas que publicam anúncio de vaga pedindo desenvolvedor "ninja", "jedi" ou outras babaquices do gênero, onde se paga pouco e se exige o mundo.

Eu também gostaria que a gente melhorasse a nossa cultura dentro das empresas... mas no fim do dia eu ainda acho que sempre fica no "contas a pagar, a receber, balanço e fluxo de caixa" 😞

(apoiadíssima a tag #aindaFaçoALV)

**Eric Carvalho** — 06/07/2016 22:32

Leo para uma empresa adotar a teoria Y primeiro ela precisa ter uma gestão 3.0 e te garanto sem medo de errar que esse cenário que vc descreveu não irá acontecer... mas aí amigão já é difícil encontrar empresa com tal maturidade, imagina se for trazer para o mundo de consultorias SAP...

Lendo novamente o post, partes do seu texto remete a problemas citados nos frameworks corporativos que são os terrores de desenvolvedores JAVA, .NET, Ruby e etc...

Caso interesse existe uma palestra do Uncle Bob na Norwegian Developers Conference 2009 falando sobre frameworks corporativos que ressaltam problemas como a desmotivação por não permitir que o desenvolvedor crie e faça a diferença, apenas codifique.

Agradeço pela citação do Bar8!!!! flw

**Leo Schmidt** — 08/07/2016 18:21

É, maturidade parece que é uma coisa difícil de encontrar mesmo.

E quando eu escrevi eu não pensei só em ABAP.

Vou dar uma olhada nessas palestras, parecem bem interessantes.

Valeu!