

A arte milenar de comentar códigos ABAP

04/09/2012 14:28

Na minhas andanças por códigos alheios e códigos DMM ("*de mim mesmo*") antigos, sempre acho curioso a quantidade de comentários sem sentido que encontro. Impulsionado por alguns artigos da interwebs, resolvi analisar **a arte milenar de comentar códigos ABAP**. Comentar... será mesmo que é **sempre** preciso/útil? Início este post com uma fábula sobre comentários, e depois apresento algumas novas idéias para você poder refletir sobre esta arte milenar. Coloque o seu kimono, e vamos para luta!

ABAP-San, o seu Comentutsu não é bom o bastante! *FU-PITCH* (hashizada na cabeça)

1 – “A treta” ou “Mauricio apanhando do próprio código”.

Quando eu comecei no mundo da programação, me disseram que **“comentar códigos é uma forma de conversar com alguém que irá dar manutenção no seu programa, ou até mesmo com o seu eu do futuro”**. Fiquei maravilhado com essa possibilidade de conversar através do ABAP, e, desde então, comecei a comentar freneticamente meus códigos, imaginando que aquele monte de palavras cinzas (ou azuis se você estiver no editor velho) iriam ajudar alguém algum dia.

O problema é que me falaram que eu deveria *“comentar”*, mas **ninguém** me disse **quais informações** eu deveria transmitir pelos comentários. Obviamente, começaram a nascer comentários infames, daquele tipo que não presta pra nada, e só enchem linguiça:

```
START-OF-SELECT
```

```
* Validações iniciais
```

```
PERFORM f_valida_dados_tela_seleção.
```

```
* Atualiza dados
```

```
PERFORM f_atualiza_tabela_casdatro.
```

```
* Exibe ALV na Tela
```

```
PERFORM f_exibe_alv_final.
```

```
* Análise de todas as ordens de vendas
```

```
LOOP AT t_ordens ASSIGNING <fs_ordens> .
```

```
* Verifica se o Material existe no cadastro de materiais
```

```

READ TABLE t_material WITH KEY
  mantr = <fs_ordens>-matnr
BINARY SEARCH.

ENDLOOP.

```

Esse tipo de comentário me lembra de provas dissertativas e aquele grande técnica de repetir a pergunta que o professor fez, para a resposta parecer maior. 😊

Com comentários inúteis como os acima, eu acreditava que o meu código estava muito bem comentado, e que isso ajudaria muito quem fosse dar manutenção no futuro. Ah e, obviamente, comentar o código era um pré-requisito da revisão de pares, então, quem fosse revisar meu código não poderia reclamar que ele não estava “bem” comentado.

Essa foi a técnica que eu utilizei por mais de 1 ano e meio... até que eu tive que alterar um código DMM antigo, daqueles que eu havia feito nos primeiros meses da minha carreira como ABAPer. O código era grande, continha diversos problemas estruturais e a lógica era complicada o suficiente para eu ter me esquecido o que diabos aquele pedaço de código fazia. Foi aí que eu percebi que os comentários que eu havia feito não prestavam para nada. Um pequeno exemplo para mostrar mais ou menos o tipo de comentário que eu encontrei:

```

* Começo da lógica sem sentido e que eu inventei agora para exemplificar o mau uso de comentários

* Análise das Ordens
LOOP AT t_ordens ASSIGNING <fs_ordens> .

* Busca o cliente
READ TABLE t_cliente WITH KEY
  kunnr = <fs_ordens>-kunnr
BINARY SEARCH.

IF sy-subrc <> 0.
* Continua para o próximo registro - não soma
  CONTINUE.
ENDIF.

wa_soma-vbeln = <fs_ordens>-vbeln.

* Busca o Material
READ TABLE t_materiais WITH KEY
  matnr = <fs_ordens>-matnr.
BINARY SEARCH.

* Efetua o cálculo
IF sy-subrc = 0.
  wa_soma-total = v_total + <fs_ordens>-valor1 + <fs_ordens>-valor2 + <fs_ordens>-valor3.
ELSE.
  wa_soma-total = v_total - ( <fs_ordens>-valor5 + <fs_ordens>-valor6 ).
ENDIF.

* Sumariza os valores
COLLECT wa_soma INTO t_soma.

...

ENDLOOP.

```

Esse código aí em cima está comentado, mas nem de longe está **bem comentado**. Para que serve essa soma? E porque aquelas validações existem? Como assim uma ordem tem um cliente que não existe na “tabela de clientes”? Porque, um material é inválido para a soma?

Pessoas mais experientes conseguirão pseudo-facilmente achar as respostas para essas perguntas, se fizessem uma análise depurando o código, ou até mesmo lendo a Especificação Técnica / Funcional. Mas eu era um ABAP Júnior, no

meio de um código de 5 mil linhas... e vou te falar: sofri **MUITO** para entender porque a soma estava errada ali no código.

Será que dava para eu ter sofrido menos, só mudando um pouco os comentários?

2 – Como o meu eu do passado poderia ter ajudado meu eu do futuro

Todo mundo já fez comentários idiotas no código algum dia. Isso é normal. O que não deve ser *normal*, é você cair na [síndrome do robô](#) e continuar fazendo comentários idiotas para sempre. Muitas vezes somos orientados desde o início à errar, pois aprendemos algumas coisas de maneira errada. Mas a situação ali de cima me fez pensar um pouco em como comentar melhor meu código.

Separei algumas reflexões sobre o uso de comentários que poderiam ter me ajudado, a partir de artigos que li em diversos lugares da web. Para o compilado deste post, usei de base artigos bem bacanas do [CodingHorror](#) sobre comentários, e também a página de comentários do tutorial de C++ do [Learncpp.com](#). Na época li outros sites, mas a mensagem é, basicamente, a mesma.

E vamos começar nossa jornada de reflexões jogando no lixo todas as redundâncias:

“Se o seu comentário explica exatamente a mesma coisa que alguém entende ao ler o comando, ele não serve para NADA”

Bom, só isso já mata praticamente todos os comentários do código da parte 1 deste post. Pra que explicar num comentário que eu vou entrar no bloco que faz validações, se o nome do FORM já me indica isso? Isso nos leva à próxima reflexão:

“Comentários não devem explicar O QUE o código está fazendo, eles devem mostrar PORQUE aquela lógica precisa existir”.

Este conceito é uma extensão do primeiro. Se levarmos em conta aquele trecho de código ali em cima, podemos dizer que aquele comentário “Efetua o cálculo” é um comentário **bem** inútil. Aliás, em momento algum os comentários me explicam o porque eu tive que fazer aquelas várias operações: somar os valores de forma diferente de acordo com a busca do material, ou ignorar a soma da linha caso o cliente não exista.. Eles só me dizem o óbvio, tornando-se comentários descartáveis.

Mas então quer dizer que eu posso dar CTRL+C, CTRL+V da EF no código? Diz a lenda que...

“...os melhores tipos de comentários são aqueles dos quais você não precisa.”

Um bom indicador para analisar se o seu código está “limpo” e tem fácil manutenção, é verificar quantos comentários estão espalhados pelo fonte. Normalmente quando você teve que usar muitos comentários para conseguir explicar o que o seu código está fazendo, pode ser a hora certa para re-escrever aquela lógica.

A arte de comentar está diretamente ligada à outra arte muito ignorada por programadores: A arte milenar de nomear variáveis e blocos do programa. Com nomes melhores de variáveis e de FORMS/Métodos/Funções, o seu código fica muito mais claro e direto, com relação às suas maléficas intenções.

Para vocês terem idéia do quanto essa questão de “saber dar nomes aos bois” é importante, tem até livros falando sobre como manter o seu código bem escrito, [como este aqui](#). Neste livro, o autor defende a tese de que **comentários são necessários somente quando há uma falha do programador** em se expressar através do código fonte. É uma idéia bem extrema, mas muitos programadores concordam com ele.

No nosso caso, quase sempre trabalhamos com desenvolvimentos cheios de regras de negócios complicadas, e com tabelas/campos/funções/classes/métodos com nomes em alemão. Logo, nós **precisamos sim** de comentários nos códigos ABAP. Um dica legal para você começar a escrever códigos mais auto-explicativos é a seguinte:

“Programe como se o recurso de comentar códigos simplesmente não existisse”

Imagine um mundo onde o compilador não reconhecesse o ‘*’ nem o ‘’’ como caracteres de início de comentários. Um mundo onde você ainda tem que se comunicar com outros programadores e até com o “você-mesmo-do-futuro” através do código. Neste mundo, a tarefa de criar um código fica muito mais interessante! A não ser que você seja um troglodita, obrigatoriamente, você tem que se preocupar **fortemente** com a estrutura do seu código, com os nomes de variáveis e de todos os outros objetos.

Entrar neste mundo da próxima vez que você começar a programar, deve ajudar o seu código a ficar muito mais auto-explicativo. A tarefa vai ficar bem mais difícil, mas a regra aqui é a boa e velha “no pain, no gain”!

3. Para ler mais

Existem quilos, litros e toneladas de artigos sobre comentários na internet. Uma pequena busca pelo assunto no StackOverflow e você vai acessar muitas boas discussões sobre o assunto.

Neste post utilizei como base [esta página](#) de um tutorial que segui recentemente, quando estava estudando C++. Foi ela que me lembrou que eu ainda tinha que falar deste assunto aqui no site. Recomendo também [este](#) e [este outro](#) artigo do Coding Horror. Em ambos os sites, leia também os comentários, para entender como as pessoas reagem à reflexões desse tipo.

4. E viva o debate!

Eu, particularmente, tento entrar naquele *mundinho* onde comentários não existem, mas recorro aos comentários para explicar partes complicadas da regra de negócio, ou para explicar alguma lógica mirabolante que eu precisei fazer. O ideal pra mim é que a pessoa consiga dar manutenção no código sem precisar olhar um documento externo.

Outro coisa que eu gosto muito de fazer, é explicar de forma geral o que determinado bloco de código faz, como se eu estivesse preparando o leitor antes de chegar na lógica pesada. Isso, aliás, já me ajudou quando eu estava dando manutenção em programas antigos que eu mesmo tinha programado.

Mas e **você**, segue alguma dessas coisas? Ou acha que é tudo baboseira, e comenta **freneticamente** coisas sem sentido? Ou talvez você escreva **letras de música** nos comentários, para deixar o código mais “cool”? Deixe a sua opinião e vamos todos discutir juntos!

*** Finaliza o post de hoje mandando um abraço. Sim, eu sou um comentário inútil.*

Abraços e até a próxima!

Comentários

Renan — 28/10/2021 11:56

Li esse post há muitos anos e parei de fazer comentários inúteis por causa dele. De vez em quando ainda volto pra ler novamente pra garantir que eu não volte a fazer isso.

Li sobre o ninho de IF's e ainda vivo me perdendo neles, geralmente gosto de colocar comentário nos ELSE's, pra saber de qual IF ele se refere, pois quando clicamos no IF o ELSE não fica marcado, dificultando a visualização e encadeamento deles.

Renan — 15/06/2016 09:45

"Outro coisa que eu gosto muito de fazer, é explicar de forma geral o que determinado bloco de código faz".
Estou no abap a uma semana e meia exatamente, e gosto de fazer isso para lembrar como utiliza os comando, por que estou aprendendo.
mas gosto muito de focar em boas maneiras, vc tem um exemplo de como vc faz isso ?
Abraço

Mauricio Cruz — 15/06/2016 09:56

Renan, tudo bem?

Esse comentário é só para lógicas com calculos ou considerações bem bizarras... vou inventar algo por exemplo

* Cálcula a % de componentes importados em cada item explodindo as listas técnicas de forma recursiva. A tabela "component_cache" previne que a função de explosão seja disparada 2x para o mesmo material.

Comentários que lembram comandos não é uma coisa muito boa... já que o F1 foi feito exatamente para isso 😊

Abs!

Renan — 15/06/2016 19:01

Perdoa ai =)

é só ate eu familiarizar com a linguagem, venho do PHP então pensa na diferença =)

vlw pela experiencia, e já to acabando seu livro, faz um técnico agora ensinando do 0 abap =) hahaha.

Abraço.

Mauricio Cruz — 16/06/2016 08:09

Valeu! 😊

E não me vejo fazendo um livro técnico de como aprender ABAP do 0... muito trabalho para ficar defasado em pouco tempo.

Abs!

Fawcs — 16/10/2012 13:06

Continuando no tema "comentarios de alterações", estou vendo um programa que foi ativado a sintese de modificação para você ajustar utilizando os botõezinhos em cima, isso também cria um monte de comentário e é standard da SAP. A sua visão sobre comentarios continua a mesma, mesmo em relação a essa ferramenta?

Mauricio Cruz — 15/06/2016 09:53

Mil anos depois, devo dizer que minha visão continua a mesma. Pobre da IDE que enfia coisa inútil...

Elton — 03/10/2012 11:03

Acabei de perceber alguns códigos com ótimos comentários assim como destes posts!! 😊

Mauricio Cruz — 03/10/2012 14:41

Não se foi irônico ou não. Se foi irônico, normal, todo mundo já passou por isso. Se não foi irônico, solta um rojão em homenagem ao cara que comentou direito. Em meio a tantos comentários zuados, o cara merece! 😊

Abs!

Elton Donadoni — 03/10/2012 15:18

Na verdade eram os comentários em códigos "DMM" esqueci de dizer!!rsrs

Mauricio Cruz — 03/10/2012 15:23

😞 Sem rojões hoje #chatiado

Amanda — 18/09/2012 08:35

UAU é beem issoo, rs

É tão triste qdo pegamos um código onde se tem coisas do tipo:

* Dá um loop na tabela de clientes

LOOP AT t_clientes INTO w_clientes.

kkkkk

Comentários são feitos para serem explicativos, mostrando a lógica que cada trecho do programa tem. Sei que todos tem uma boa intenção por trás, mas comentários redundantes além de deixar o código gigante, até atrapalha uma manutenção futura ou até mesmo a sua depois de um tempo, rs

post muito fera, falou tudo 😊

Amanda — 18/09/2012 08:30

Quando será que vamos perceber que esse tipo de comentário além de não ajudar, ATRAPALHA quem for dar manutenção ou até mesmo a própria pessoa que o desenvolveu? rs

Comentários devem ser explicativos, mostrando a lógica de como as coisas foram feitas, e não redundantes e sem propósitos.

Eu fico chateada qdo vejo coisas do tipo:

*Dá um loop na tabela de clientes

LOOP AT t_clientes INTO w_clientes

kkkkkkkkkkkkk

Agr sério, ÓTIMO POST, parabéns! 😊

Raul Romano — 14/09/2012 14:22

Cara, adorei esse POST...

Muito interessante mesmo, são detalhes que não damos tanta importância, mais quando entramos em algum projeto e nos deparamos com códigos com uma lógica meia complicada e pra piorar sem comentários...ai sim, lembramos da importância de um bom comentário.

Valeu mesmo cara...

Estou em um projeto atualmente e pensando bem vou apagar o "LIXO" que deixei lá e deixar os comentários com o "PORQUE" e não "O QUE".

Abraço!

Mauricio Cruz — 14/09/2012 15:10

Olá Raul! Bom saber que você curtiu! 😊

Abraços!

Arthur — 10/09/2012 17:57

Também comentava assim até perceber, durante a codificação mesmo, que aquilo ficava muito confuso e não falava o que o programa fazia como um todo, e sim somente aquela linha. Descobri isso fazendo uma função de mais de 5mil linhas e diversos ninhos de IF. Parecia um labirinto.

Mauricio Cruz — 11/09/2012 10:13

Nem me fale de labirintos de IFs. O mais divertido é quando alguém cria vários forms dentro de outros forms, achando que isso faz o programa ter "menos" ifs, haha.

Abraços!

Haroldo — 06/09/2012 20:10

Confesso que já escrevi muito "lixo" e muitas vezes deixei de comentar também.

Esses dias mesmo peguei um código DMM e por falta de um comentário, tive que me colocar na posição de "chico xavier" e lembrar o que aquilo fazia.

Consegui me lembrar e comentei na mesma hora.

Ótimo post...

Mauricio Cruz — 11/09/2012 10:12

Valeu Haroldo!

Fawcs — 04/09/2012 23:22

Bom, eu gosto de ler posts assim.. não programo a muito tempo então estou constantemente mudando o modo de programar, qualquer coisa sobre "boas práticas" de programação em geral ganha uma atenção especial pra mim.

Particularmente e pode parecer uma inutilidade, mas eu gosto de escrever que raios que é a variável do lado da sua declaração pra não ter que ficar viajando pra SE11(malditas conexões lentas e vpns ingratas!), não pratico

tanto, mas em um cliente que eu demoro 5 segundos entre uma tela e outra é uma ajuda e tanto.

Eu costumava comentar muito mais na primeira semana como abap, agora eu só comento lugares que eu acho complicado de compreender.

Outra coisa, isso de não comentar o que o código faz quando o comentário é exatamente o comando é IMO igual usar constantes em vez de hardcode, pra que raios você vai fazer uma constante `c_x = 'X'` (um `c_true` ainda vai, mas já é forçar a barra `=P`) se não há ganho?

Mauricio Cruz — 05/09/2012 10:35

Então.. mas esses comentários todos que você utiliza para não ter que ir para a SE11, são relevantes somente em tempo de desenvolvimento. Não é todo mundo que vai acessar pela VPN, não é pra todo mundo que a VPN é lenta, pode ser que a VPN fique melhor no futuro, etc, etc,etc.. a chance desses seus comentários se tornarem inúteis é beeem grande 😊

E é verdade, um `c_x` ou " *Lê a tabela" antes de um `READ TABLE` é tudo farinha do mesmo saco 😊

Abraços e valeu pelo comentário!

Fawcs — 05/09/2012 16:41

O dia que a VPN dessa galera melhorar vai ser o dia que todos eles saírem do 4.6C XD

E qual a sua opinião sobre comentário de alteração? eu acho que depois de um tempo polui o código, mas também acho necessário pra quem trabalha em suporte!

Mauricio Cruz — 05/09/2012 22:56

Atrapalha mais do que ajuda. Depois de um tempo sempre vira aquela salada de "comentário de alteração em cima de comentário de alteração", e fica ridículo de ler o código.

E o pior é que são poucos os devs que eu conheço que usam a ferramenta de comparação de versões do SAP (o que eu acho bem bizarro).

Fawcs — 06/09/2012 08:27

Eu sempre acabo esquecendo da ferramenta mesmo..

Sobre os comentários, eu acabo fazendo igual nota da SAP quando tenho oportunidade, não acho que um `"EKSD89797687786899809809"` deixe muito "zuado" o código `=P`