

Manias de Código

05/02/2015 09:00

Todo programador tem as suas “manias”: **SEU** jeito de fazer um READ TABLE, o **SEU** jeito de escrever um SELECT... Muitos desses costumes dependem do seu aprendizado (herdados do seu mentor ABAPer), outros são desenvolvidos com o tempo, conforme você vai descobrindo o que funciona e o que não funciona para suas necessidades.

O fato é um só: todo programador tem as suas “manias”. Aqui eu compartilho as minhas e espero que você também compartilhe as suas.

Para começar, eu confesso que demorou um bom tempo para que eu admirasse as diferentes formas de estruturar um código. No começo eu sempre pensava que o MEU jeito era, obviamente, o jeito mais claro para que EU entendesse o meu código (o que faz sentido), mas também achava que o MEU jeito era o melhor para que os OUTROS entendessem meus códigos (o que demonstra uma prepotência desgraçada). Mas um dia a gente cresce e larga a mão de besteira, sabe como é 😊 .

Para não virar zona, vou organizar por tópicos.

SELECTs

Eu tenho uma forma bem particular de organizar meus SELECTs, tanto que só de bater o olho eu já sei se fui eu ou não que fiz aquilo. Basicamente eu sigo um padrão bem idiota de alinhamento do código, que não faz sentido algum. Vamos fazer um SELECT com FOR ALL ENTRIES:

```
SELECT  campo1
        campo2
        campo3
FROM    tabela_standard
INTO TABLE tabela_interna
FOR ALL ENTRIES IN fae_table
WHERE   campo1 = 'bla'
        AND campo2 = 'bla'.
```

- Eu nunca coloco mais de um campo de seleção na mesma linha. **Nunca.**
- O alinhamento é bizarro: eu alinho pelas primeiras palavras reservadas de cada pedaço do comando: SELECT, FROM, INTO, FOR, WHERE e AND. Se tiver mais de uma não importa, alinho só pela primeira (como o INTO de INTO TABLE).
- Todos os operadores lógicos (AND/OR) vão abaixo do WHERE, abaixo um dos outros.
- Agora a melhor parte: se tiver um INNER JOIN eu não ligo que ele quebre a formatação, mas o resto precisa ficar alinhado como eu já expliquei. Sabe-se lá o porquê.

Dá para dizer também que eu nunca uso EQ, LT, GT e etc... mas isso vale para todo o código, não só para o SELECT.

READ TABLE

O Mauro gosta muito desta mania. Um chefe que eu tive e que me ensinou muito sobre SAP e ABAP me passou essa mania como um legado e eu sigo isso até hoje. Pensando bem, acho que ele nem sabe que eu faço isso por causa dele, então fica como se fosse uma cópia.

Esse daqui você vai achar estranho, certeza.

Meu READ TABLE é igual um **hamburguer**. Sério.

```

READ TABLE tabela INTO workarea WITH KEY
    campo1 = 'bla'
    campo2 = 'bla'
BINARY SEARCH.

```

Não entendeu? Se liga:

```

PAO COM GERGELIM
dois hamburgheres
alface
queijo
molho especial
cebola
picles
PAO COM GERGELIM

```

Eu simplesmente NÃO consigo fazer READ TABLEs de outra forma. As palavras reservadas abraçam as condições.

Não me zoa, sério.

Declarações de Variáveis

Eu tenho uma preguiça monstruosa de ter que declarar tudo daquele jeito enorme do ABAP. Acho muito chato... **muito**.

Se pelos dois tópicos anteriores você pensou “pqp que cara crica com a organização”, acalme-se comandante. Eu não ligo a MÍNIMA para como as variáveis são declaradas. Tanto que é bem comum ver coisas desse tipo nos meus códigos:

```

DATA: var1 TYPE bla.
DATA var2 TYPE oi.
DATA: xyz TYPE 1234,
      hjs TYPE char200.

```

Não segue padrão nenhum, vou adicionando conforme vou criando e fico com um preguiça imensa de ajustar para ficar tudo em um só “DATA”. Tento manter os vários DATA, TYPES e FIELD-SYMBOLs agrupados, mas nem sempre sigo esse padrão.

Basicamente, não estou nem aí. #shameonme

Eu tenho tanta preguiça que transformei isso em uma prática “positiva”: penso muito antes de criar uma variável. Quanto mais variáveis eu crio, mais me dá a sensação de que eu estou fazendo besteira e poderia ter resolvido com menos áreas de memória. Tudo para não precisar declarar.

(Tá bom... na verdade, esse último parágrafo sou eu me convencendo de que não sou um preguiçoso do kct).

IFs encadeados

Quando eu comecei com ABAP eu fazia muito isso aqui:

```

IF bla1 = bla1.
    IF bla2 = bla2.
        IF bla3 = bla3.
            IF bla4 = bla4.

```

uma cacetada de código

```
ENDIF. "bla1
ENDIF. "bla2
ENDIF. " bla3
ENDIF. "bla4
```

Eu pensava que isso ia me ajudar no código, mas depois de um tempo eu saquei a tremenda da besteira que é fazer isso. Se eu quiser ver qual é o IF, é só clicar duas vezes... Tudo bem que isso foi na época da 4.6 e a navegação dentro do debug antigo era muito zuada... mas não justifica. Isso é zoado.

Desde então, declarei guerra aos IFs encadeados. Fujo deles como diabo foge da cruz. Eu faço um esquema que eu gosto de chamar de "IF de exclusão". Ao invés de usar o IF para dizer "faça a lógica maluca se o IF der certo", eu digo "CAIA FORA se o IF falhar".

Código acima reescrito:

```
IF bla1 <> bla1.
  EXIT.
ENDIF.

IF bla2 <> bla2.
  EXIT.
ENDIF.

IF bla3 <> bla3.
  EXIT.
ENDIF.

IF bla4 <> bla4.
  EXIT.
ENDIF.
```

Agora tenho CERTEZA que se o programa chegar aqui é pq ele tinha que chegar mesmo.
E o melhor: sem aquela zona de **IF** dentro de **IF**.

Isso vale não só para o EXIT, mas para CONTINUEs também dentro de SELECTs. Eu sempre coloco tudo que for IF de exclusão ante de qualquer lógica que eu vá fazer, tratando as mensagens dentro dele. E assim eu também evito aqueles programas bizarros com lógicas dentro de 3849348 IFs.

Aliás, os IFs de exclusão estão muito presentes no começo dos meus métodos e forms.

Aposto que não sou o único maluco

Mostrei para vocês minhas 4 principais manias quando estou abapando. Tenho CERTEZA que você também tem as suas. Quais são elas? Coloque ae nos comentários!

Abs a todos que roem unhas.

Comentários

Rogério Marques — 12/01/2019 10:59

Eita faço tudo diferente disso aff kkkkkk

Guilherme Dellagustin — 19/05/2015 15:35

Eu geralmente sigo uma regra de ir incrementando a indentação statements multi linha:

```
SELECT a FROM b INTO TABLE c
WHERE
    field1 = f1 AND
    field2 = f2.
```

Mas a minha mania fixa que já está bem definida é chamadas de métodos:

```
lv_value = lo_instance->get_something(
    EXPORTING
        iv_attr = lv_attr
    CHANGING
        cv_attr = lv_other_attr
).
```

Sempre fechando o Parêntese alinhado com o a indentação onde a chamada foi feita.

Guilherme Dellagustin — 19/05/2015 15:36

Não saiu minha indentação.
Como crio blocos de código?

Mauricio Cruz — 20/05/2015 10:44

Arrumei para você. Valeu por compartilhar! 😊

O Léo aqui do site faz chamadas de métodos exatamente do mesmo jeito nos códigos dele. Eu não gosto pq a linha do parênteses fica vazio, mas eu entendo que numa suposta alteração fica mais fácil.

Enfim, como dizia o meu pai: cada qual com seu cada qual (?).

Abs!

Leo Schmidt — 20/05/2015 18:53

Eu uso essa formatação aí pra chamada de métodos até hoje porque é a mesma do Code Wizard do WebDynpro. Mas eu gosto dela na verdade porque isso de deixar parênteses/chaves abertos(as) é costume de outras linguagens tipo C# e JavaScript. Aliás, chamadas encadeadas também são, e eu também uso sempre que possível.

Outra coisa: nesse post aqui a única pessoa que escreveu um SELECT nota 9,5 de formatação foi a Daiane. #ChoraHaters

Mauricio Cruz — 27/05/2015 08:54

Chaves ok, parênteses me irritam. E o SELECT da Dai é desalinhado, blergh.

Alexandre Azambuja — 27/04/2015 17:06

Ótimos posts.

Flavio — 20/03/2015 12:30

Isso não é mania não, é cuidado com o código. Quem não tem mania é porque não gosta de programar (mamilos!).

Maurício, seu READ TABLE é bizarro ahahahahah!

Vagnão, eu também acho legal usar EQ, NE etc. e "=" para atribuição. Ainda vou mais longe, prefiro usar ADD 1 to X ao invés de $x = x + 1$. A mesma coisa para o SUBTRACT.

FORM, o que é isso mesmo? (mamilos2)

Para mim, #ABAP740SP8 ainda é um grau de nirvana ainda a ser alcançado. Já fico feliz de ter acesso ao 7.2 e poder usar ADT aqui na empresa. Ok, temos 7.2, mas ainda não tenho autorização para ADT, mas uma coisa de cada vez né....

Abraços!

Custodio — 24/02/2015 23:12

Quem usa READ hoje em dia?

`data(estrutura) = tabela[campo1 = 'bla'].`

#ABAP740SP8



Felipe Simionatto — 25/02/2015 05:58

Pra mim 7.40 por enquanto é igual nota R\$100, só na AWS mesmo. :/

Felipe Simionatto — 24/02/2015 22:58

Se usar o READ em uma sorted table vai ficar faltando um 'teco' do hamburger. =)

Vagnão — 19/02/2015 12:42

Algumas manias do Vagnão:

`Select campo1 campo2 campo3 campo4 campo5 campoN`

`campoN+1... (eu alinho no primeiro campo)`

`into table (2 espaços)`

`where campo1 in s_so (2 espaços)`

`and campo2 eq algo (alinhado à direita do where)`

=====

Utiizo GT LT EQ e tudo o que for possível. Símbolo = só uso para atribuição de valor.

=====

Read table itab into wa

with key campo1 = algo (dois espaços... pena que não dá pra usar EQ...)

=====

Declaração de dados:

Se possível, um DATA só, um TYPES só. Alinhamento? Já cansei disso. Agora vai como estiver e ainda mais: itabs e wa juntas, se possível. Exemplo (real):

```
DATA: v_fikey TYPE fikey_kk,
v_hkont TYPE hkont_kk,
wa_addons TYPE zccsaccst0119,
wa_basics TYPE fkk_mad_basics,
t_t0165 TYPE TABLE OF zccsacct0165,
wa_t0165 TYPE zccsacct0165.
```

=====

Macros. Uso sem dó sempre que a rotina é besta. Exemplo:

```
DEFINE check_intervals.
```

```
loop at &1.
```

```
if &1-high is not initial.
```

```
message e208(00) with 'Não incluir valores De – Até'.
```

```
endif.
```

```
endloop.
```

```
END-OF-DEFINITION.
```

```
AT SELECTION-SCREEN ON s_fikey.
```

```
check_intervals s_fikey.
```

=====

Linhas em branco no código. O mínimo possível, até onde não fica ruim de visualizar. Prefiro ver um monte de código na tela do que um monte de linhas em branco. Duas linhas em branco contínuas na tela e eu já tenho convulsões.

=====

Rotinas form (Sim, eu uso muito Form): Todas com nome em inglês e iniciando com verbo. Exemplo da chamada e do form:

```
perform get_header_data using wa_basics.
```

```
form get_header_data using basics TYPE fkk_mad_basics.
```

Detalhe (mania bizarra do Vagnão): tudo que eu declaro no form "perde" o prefixo. Exemplo: t_itab vira itab, wa_area vira area, v_myvar vira myvar.

Abraços!!!

fawcs — 06/02/2015 09:21

Dúvida para os loucos de plantão, como vocês se acertam com um IF com vários 'AND'

```
IF 1 = 2
AND 2 = 3
AND 3 = 4.

ENDIF.
```

vocês:

1) alinham o primeiro if com o resto:

```
IF 1 = 2
AND 2 = 3
AND 3 = 4.

ENDIF.
```

2) alinham só a clausula:

```
IF      1 = 2
AND 2 = 3
AND 3 = 4.

ENDIF.
```

3) ignoram

4) ficam frustrados(eu).

Vinicius Silva — 06/02/2015 10:09

Eu alinho todos os "=" e todos os "AND" e "OR" rs.

ps: eu sempre coloco os "AND" e "OR" no fim da linha

```
IF aaaa = bbbb AND
ccc = ddd AND
e = f AND
GG = HH.
```

Vinicius Silva — 06/02/2015 10:13

ops:

```
IF aaaa = bbbb AND
  ccc  = ddd  AND
  e    = f    AND
  GG   = HH.
```

Mauricio Cruz — 09/02/2015 09:42

Eu faço igualzinho!

fawcs — 06/02/2015 09:26

nao consigo alinhar esses codigos:(pqp

daniel jesus — 06/02/2015 10:03

Alinho o IF com AND e as clausulas também ahahahhahahaha

Daniel Jesus — 05/02/2015 20:06

Sasinhora!
Bloco dos Abaps Loucos!

Custodio — 05/02/2015 18:17

Dica pra voces:

Clínica Prisma – Excelência em Saúde Mental

<http://www.clinicaprisma.com.br/>

Especializada no tratamento de transtornos psiquiátricos e dependência química.

Abracos,
Custodio

Bruno Lucattelli — 09/02/2015 09:50

Obrigado Custódio. Finalmente largarei minhas manias ABAP e o vício de café.

Mauricio Cruz — 09/02/2015 10:05

Nem preciso disso, eu já trabalho em uma empresa de psicologia:

<http://www.institutonumen.com.br/index.php/clinica-numen>

Os loucos são vocês, normais.

Vinicius Silva — 05/02/2015 13:40

Eu tenho manias para praticamente todos os comandos e declarações, mas a que mais me irrita é: se eu vou chamar um perform mais de uma vez seguida eu PRECISO alinhar as chamadas, mesmo que seja um FCAT de duzentos campos!

fawcs — 05/02/2015 11:56

meu read table é muito semelhante, adotei ele quando estava numa idéia de tentar deixar meu código padrão e fácil de achar as os pontos importantes.

O select depende do humor, de vez em quando eu alinho com a primeira clausula, de vez em quando eu alinho com o FROM alinhado com a primeira clausula.. tpo:

```
SELECT matrnr mmenge meins FROM ekpo  
INTO CORRESPONDING FIELDS OF t_ekpo  
FOR ALL ENTRIES IN T_wow
```



```
WHERE ebeln = T_wow-ebeln
AND ebelp = T_wow-ebelp.
```

Eu tento deixar agrupado pontos lógicos que trabalham em conjunto... por exemplo

```
fae_table = itab.
```

```
sort fae_table by keys.
```

```
delete adjacent duplicates from fae_table comparing keys.
```

```
if fae_table is not initial.
```

```
select fae
```

```
endif.
```

Mas é engraçado ver como os abapers se atrapalham com os settings do pretty printer. de vez em quando eu faço um experimento social aqui e marreto a tabela de opções da pessoa pra ver quanto tempo ela demora pra perceber que as maiúsculas estão diferentes.. 😊

Mauricio Cruz — 05/02/2015 13:51

Putz esses experimentos são os melhores, também gosto de fazer. Boa!

Bruno Lucattelli — 05/02/2015 13:58

Aqui não é preciso marretar nada. TODO MUNDO JÁ USA O MESMO USUÁRIO ABAP.

LOL

ABS

Miller Nunes — 05/02/2015 11:14

Meu select também é exatamente igual ao seu. Mas eu vou um pouco além. As condições dos meus IFs também tem que ficar alinhadas. rsrsrs

Bruno Lucattelli — 05/02/2015 14:03

IFs!! Vamos falar de IFs!!!

Eu tenho uma coisa com os parêntesis no IF. Pra mim não há meio termo: ou usa parêntesis, ou não usa. Esta regra não vale para um programa inteiro, mas vale para cada IF. Ou seja, isto é aceitável:

```
IF 1 EQ 2.
  EXIT.
ENDIF.
```

Isto também é aceitável:

```
IF ( 1 EQ 2 ).
  EXIT.
ENDIF.
```

Porém, isso aqui **não** é:

```
IF 1 EQ 2 AND ( 2 EQ 3 OR 2 EQ 4 ).
  EXIT.
ENDIF.
```

Pra mim, se você colocou UM PARÊNTESES, coloque TODOS!

```
IF ( 1 EQ 2 ) AND ( 2 EQ 3 OR 2 EQ 4 ).
  EXIT.
ENDIF.
```

Mauricio Cruz — 05/02/2015 14:08

Sente essa vibe, você que está lendo isso.

Essa é a hora que um comentário faz o autor sentir-se bem, pq PQP COMO TEM DOIDO

Bruno Lucattelli — 05/02/2015 14:13

HAHAHAHAHA

Bruno Lucattelli — 05/02/2015 10:36

Engraçado. Me identifiquei com vários destes itens aí.

Uma coisa que eu costumo fazer é separar o código em blocos, deixando linhas em branco em todo o lugar. Exemplo:

```
// BLOCO 1
IF 1 EQ 2.
  EXIT.
ENDIF.

IF 1 EQ 3.

// BLOCO 2.
MESSAGE 'X' TYPE 'I'.
PERFORM z.
PERFORM a.
PERFORM p.
SET PARAMETER ID 'ZAP' FIELD 'ZOMBIE'.

ENDIF.
```

Aliás, eu peguei uma mania com o Guilherme do ABAP.ninja de usar comentários sempre com aspas (mesmo quando começa no início da linha) seguidos por duas barras.

Outra coisa (considerada polêmica) é agrupar informações globais no TOP da seguinte maneira:

```
DATA : BEGIN OF global,
        t_header TYPE TABLE OF type_header WITH DEFAULT KEY,
        t_items TYPE TABLE OF type_items WITH DEFAULT KEY,
      END OF global.

DATA : BEGIN OF dynpro_0100,
```

```

        alv TYPE REF TO cl_salv_table,
    END OF dynpro_0100.

DATA : BEGIN OF dynpro_0200,
        alv TYPE REF TO cl_salv_table,
    END OF dynpro_0200.

CLEAR : global, dynpro_0100, dynpro_0200.

```

Eu concordo que é um pouco polêmico, mas desta forma eu garanto algumas coisas como:

- Limpar todas as variáveis globais é fácil e eu não preciso me preocupar em sair incluindo a limpeza de novas variáveis aqui e ali. Um bom **CLEAR global.** funciona.
- Variáveis de tela que precisam ser globais como ALV, CONTAINER, ETC não precisam ter nomes bizarros para diferenciar um do outro.
- As variáveis globais que são usadas no programa todo são facilmente separadas das globais usadas em um escopo de tela ou FM.
- Não preciso separar locais de globais usando **LT_DADOS[] = GT_DADOS[]**, e além disso **T_DADOS[] = GLOBAL-T_DADOS[]**. é mais evidente.

Outra coisa é o alinhamento dos dois pontos nas declarações. Eu costumo separá-los do token, em especial os DATA (com dois espaços) para alinhar TYPES e DATAs, conforme abaixo:

```

TYPES : type_maracutaia TYPE matnr.
DATA : t_maracutaia TYPE TABLE OF type_maracutaia WITH DEFAULT KEY.

```

fawcs — 05/02/2015 11:50

Gostei dos detalhes das variáveis globais, começou a virar costume na empresa encapsular as globais numa classe e matar ela quando precisa.. mas essa das telas é muito boa

Bruno Lucattelli — 05/02/2015 12:57

A classe global é legal também porque ela não armazena só variável, como rotinas de código que podem dar FLUSH na tela, etc.

Eu comecei a usar isso pra facilitar a limpeza de uma tela. É muito fácil fazer algo como:

```

CLEAR dynpro_0100.
CALL SCREEN 0100.

```

Assim você não se preocupa né?

Mauricio Cruz — 05/02/2015 13:50

Putz eu faço tudo diferente. Eu não crio globais, fim.

haha

Bruno Lucattelli — 05/02/2015 13:56

Então, eu já fui radical assim. Só que em alguns casos eu me via precisando adicionar um parâmetro em um FORM pq precisava dele em um FORM que era chamado por outro FORM que era chamado

pelo primeiro FORM.

Este é um sintoma de que você está localizando dados que na verdade deveriam ser globais. Eu ainda uso esta abordagem, mas quando percebo o sintoma aparecendo, mudo pra uma global.

E sim, pra OO a parada é diferente, mas eu ainda sou obrigado a trabalhar com FORMS aqui 😊

Daiane Zigiotto — 05/02/2015 10:35

Ah muito bom!

Eu fico doida no meio de código bagunçado, a ponto de gastar tempo só para deixar arrumado... No geral as mulheres percebem mais detalhes visuais que os homens (é a ciência que diz isso), então é por isso que pra mim é fundamental que todo código tem que ser bem organizado (senão eu surto). Eu também tenho minhas manias, é claro. Por exemplo:

```
* Selects
SELECT campo1
      campo2
      campo3
FROM tabela_standard
INTO TABLE tabela_interna
FOR ALL ENTRIES IN fae_table
WHERE campo1 = 'bla'
      AND campo2 = 'bla'.

* Read table
READ TABLE tabela INTO workarea WITH KEY campo1 = 'bla'
                                         campo2 = 'bla'
                                         BINARY SEARCH.
```

É isso ae mano! Abs!

PS: Seu read table é horroroso! hahaha

Mauricio Cruz — 05/02/2015 10:58

Haters gonna hate 😊

READ TABLE de HAMBURGUER #FTW \o/

Mauro Laranjeira — 06/03/2015 16:04

Eu aprendi assim "tem que parecer um hamburger"

Vlw Mauricio por me encimar a fazer hamburger

Fred Pereira — 05/02/2015 10:08

Meu select é muito parecido em indentação com o seu, mas a minha pior mania (a mais inútil) é...indentar pontos e virgulas!

Denis Vieira Andrade — 05/02/2015 09:32

Meu select é EXATAMENTE igual... rrsrrsrrs... E eu ainda tenho mania de mudar qualquer um que eu veja pela frente se estiver fazendo alguma manutenção....

Rodney Amancio — 02/03/2015 17:35

O meu SELECT é identico porém sou extremamente metódico nas declarações, datas, type, etc... tudo em um único data, com os types alinhados, flores e violinos...meu read table é alinhado todo a direita...porém eu estou para o IF assim como vc está para as declarações de variáveis....rsrs

abs

Mauricio Cruz — 05/02/2015 13:52

Putz, disso eu não sofro sabia? Não ajusto o resto do código que eu estiver trabalhando, até para não sujar o log de modificações.