

Processamento em Paralelo utilizando RFC Assíncrona

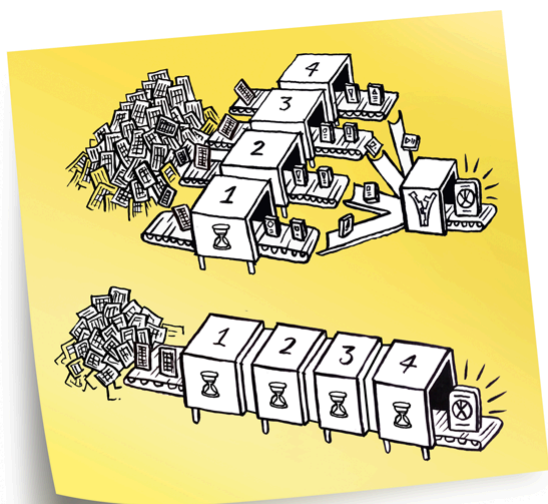
12/03/2012 15:39

Demorou, mas chegou! Meu post inaugural no ABAP Zombie! Vou começar falando um pouco sobre processamento paralelo com RFC Assíncrona. Hein???

O processamento em paralelo (ou paralelismo) não é um conceito novo, mas é pouco utilizado para melhoria de performance. Por quê?

Em alguns casos podemos utilizar mais de uma sessão (ou task) disponível, mas geralmente utilizamos apenas uma delas. É como se uma transportadora contasse apenas com um caminhão para levar seus produtos, quando há uma frota inteira de caminhões que podem ser utilizados.

Não é só isso, a maioria dos sistemas conta com mais de um servidor de aplicação com uma série de sessões disponíveis. Então, o que estamos esperando?



Utilizar processamento paralelo ou não, eis a questão!

Neste post vou mostrar um exemplo prático utilizando RFC Assíncrona. Vamos tentar equilibrar o trabalho do programa com base no número de dados informados na tela de seleção e nas sessões disponíveis nos servidores de aplicação. Por último, vamos analisar o tempo de execução do programa e comparar a execução utilizando uma única sessão. Então, vamos lá!

```
REPORT zppbap_rfc_assincrona.
*****
* VARIÁVEIS GLOBAIS (V_...) *
*****

DATA: v_tasks      TYPE i,
      v_task_id    TYPE numc2,
      v_task_count TYPE i,
      v_task_ativa TYPE i.

*****
* TABELA INTERNA (T_...) *
*****

DATA: t_bseg      TYPE TABLE OF bseg,
      t_bseg_aux  TYPE TABLE OF bseg.
```

```

*****
* TABELA                                                                 *
*****
TABLES: bseg.

*****
* PARÂMETROS DE TELA:                                                  *
*   SELECT OPTIONS (S_...)                                             *
*   PARAMETERS     (P_...)                                             *
*****
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME.

SELECT-OPTIONS: s_gjahr FOR bseg-gjahr OBLIGATORY NO INTERVALS. "Ano
* Obs. Estou utilizando apenas como chave o campo GJAHR, pois o ambiente
* que estou fazendo testes não possui muitos registros

SELECTION-SCREEN END OF BLOCK b1.

*****
* START-OF-SELECTION                                                  *
*****
START-OF-SELECTION.
    DATA: wa_gjahr LIKE LINE OF s_gjahr.

* Obter o número de sessões disponíveis
CALL FUNCTION 'SPBT_INITIALIZE'
    IMPORTING
        free_pbt_wps                = v_tasks
    EXCEPTIONS
        invalid_group_name          = 1
        internal_error              = 2
        pbt_env_already_initialized = 3
        currently_no_resources_avail = 4
        no_pbt_resources_found      = 5
        cant_init_different_pbt_groups = 6
    OTHERS                          = 7.

    IF sy-subrc <> 0.
*** Mensagem de erro
        LEAVE LIST-PROCESSING.
    ENDIF.

* Para cada ano informado na tela de seleção, será chamada a RFC
LOOP AT s_gjahr INTO wa_gjahr.

* Como não sabemos a quantidade de registros, devemos controlar quantas sessões serão geradas
DO.

* Incrementar ao contador de sessões ativas
    IF sy-subrc IS INITIAL.
        ADD 1 TO v_task_ativa.
    ENDIF.

* Verificar se o número de sessões ativas está dentro do limite
    IF v_task_ativa <= v_tasks.

* Cada sessão deverá ter um ID único
        ADD 1 TO v_task_id.

* Chamar a RFC em uma nova sessão
        CALL FUNCTION 'Z_RFC_ASSINC'
            STARTING NEW TASK v_task_id
            DESTINATION IN GROUP DEFAULT
            PERFORMING update_order ON END OF TASK
        EXPORTING
            i_gjahr = wa_gjahr-low

```

```

EXCEPTIONS
  OTHERS = 1.

IF sy-subrc <> 0.
* Se a RFC falhar, vamos tentar novamente e diminuir o número de sessões ativas. Cuidado para não
* em loop infinito!!
  SUBTRACT 1 FROM v_task_ativa.
ELSE.
  EXIT.
ENDIF.
ELSE.
  SUBTRACT 1 FROM v_task_ativa.
ENDIF.
ENDDO.
ENDLOOP.

IF sy-subrc IS INITIAL.
* Esperar até que todas as sessões sejam finalizadas. O número é decrementado na subrotina abaixo
* e incrementado quando a RFC é chamada.
  WAIT UNTIL v_task_ativa = 0.

*** Neste ponto você poderá utilizar os resultados retornados da RFC para continuar a lógica do pr

ENDIF.

*****
* SUBROTINA
*****
FORM update_order USING name.

* Obter os resultados da RFC
RECEIVE RESULTS FROM FUNCTION 'Z_RFC_ASSINC'
  TABLES
    t_bseg = t_bseg_aux.

APPEND LINES OF t_bseg_aux TO t_bseg.

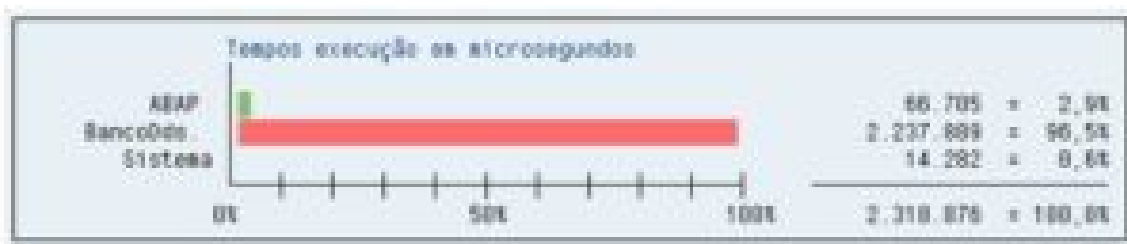
SUBTRACT 1 FROM v_task_ativa.

ENDFORM.                " UPDATE_ORDER

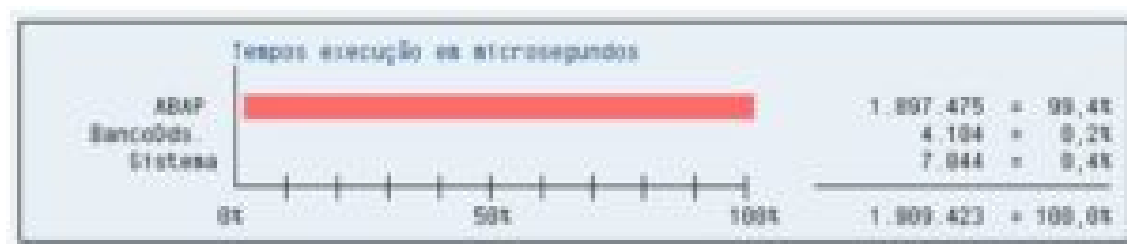
```

Análise do Tempo de Execução

Única sessão:



Utilizando Processamento em Paralelo:



Através da transação SE30, podemos analisar o tempo de execução de cada programa. No meu exemplo, eu consegui uma melhoria de 18% do tempo que levou para executá-lo em uma única sessão.

Mas atenção!!!

- Antes de começar a implementar, devemos nos certificar que o processamento em paralelo deve ser logicamente independente. Como assim? Explico: cada processo não pode depender de outros dados que estão sendo processados paralelamente, pois não temos como garantir que os dados serão processados em uma ordem específica.
- O programa principal não pode mudar de sessão depois de chamar a RFC Assíncrona, ou seja, você não deve usar SUBMIT ou CALL TRANSACTION depois de usar CALL FUNCTION STARTING NEW TASK.
- Não é recomendado utilizar COMMIT dentro da RFC Assíncrona, pois poderia entrar em conflito com a sessão em que o programa principal está sendo executado.

É isso aí zombies, espero que tenham gostado, até uma próxima! 😊

Comentários

Gilson — 20/01/2015 09:14

Olá pessoal!

Utilizei recentemente este exemplo no meu programa e tenho uma dúvida aqui... Quando o programa atinge todas as sessões ele cai no ELSE e decrementa 1 na variável v_task_ativa. Porém, volta para o início do Do e incrementa novamente e fica nesse Loop infinito por que só irá chamar o perform UPDATE_ORDER para subtrair a variável v_task_ativa quando sair do Loop. Isso é normal? Ou estou fazendo algo errado? :S

Abs.

Daiane Zigiotto — 05/02/2015 10:26

Oi Gilson! Desculpa a demora 😊

Muito bem observado, tinha um erro no código 😊

Coloquei uma validação de sy-subrc, assim nos casos de erro, não incrementa o número de tasks, senão vira um loop infinito. Acho que fazendo isso já soluciona o problema.

Vou arrumar o código no post, valeu pelo comentário!

Robson — 24/09/2017 15:46

Segundo meu teste, se tiver menos tasks disponíveis que o número de incrementos do LOOP vai ficar em loop infinito, pois não vai ocorrer a chamada da FORM update_order para decrementar o números de tasks ativas. Pelo que estou observando, só vai chamar este FORM se encontrar um WAIT, faz sentido?

Ex.:

Número de tasks disponíveis: 2

Número de anos a serem processados: 3

Obrigado.

Daniel Jesus — 17/02/2014 11:55

Mauricio, as funções ficam mesmo dialogo usando esta técnica. =/

http://help.sap.com/saphelp_nw04/helpdata/EN/22/0425c6488911d189490000e829fbbd/content.htm

Tatiana — 13/06/2013 09:29

Olá Daiane, obrigada pelo retorno!

Bom, já utilizamos este recurso.. 😞

De fato ajuda, mas não resolve o problema. A nossa questão aqui é o acesso a tabelas muito grandes, sem arquivamento.

A outra idéia que surgiu foi disparar vários JOBS (pequenos JOBS, assim como é feita a chamada da CALL FUNCTION). O que estou tentando ver agora é como encadear as execuções posteriores. Tenho 5 processamentos encadeados, dentro do mesmo programa. O primeiro é que é problemático, e foi neste que trabalhamos o paralelismo. Os próximos 4 dependem da conclusão deste primeiro. Na solução com paralelismo, o tempode execução caiu de 2 horas para 20 min. Gostaria muito de continuar com este tempo... rrsrrs...

Tatiana — 13/06/2013 08:31

Oi Mauricio, obrigada pelo retorno.

Bom, hoje meu processamento está sendo executado em background sem cancelar. Porém tive que aumentar um pouco o número de seções a serem abertas. Pois com um número menor está dando TIME OUT.

Daiane Zigiutto — 13/06/2013 09:18

Olá Tatiana, obrigada pelo seu comentário! Talvez o que pode te ajudar a definir a número de tasks é utilizando a função SPBT_INITIALIZE onde retorna a quantidade de tasks livres (FREE_PBT_WPS) e a quantidade máxima de tasks (MAX_PBT_WPS). Abs!

Tatiana — 12/06/2013 16:56

Olá, boa tarde!

Tenho uma dúvida: é possível este processamento ser executado em background?

Pelo que percebi, as seções são abertas "online" mesmo quando executo em background. Com isso, se for um processamento muito pesado, posso ter o risco de TIME OUT em alguma destas seções.

Mauricio Cruz — 13/06/2013 08:19

Olha, até onde eu sei essas novas tasks também são executadas em background... Particularmente, sempre otimizei programas massivos que são executados em background com paralelismo (starting new task) e nunca tive problemas.

Abs!

Navier Stokes — 26/09/2012 22:24

achei um lance no link http://help.sap.com/abapdocu_70/en/ABAPCALL_FUNCTION_STARTING.htm

da um search em "...must not exceed certain threshold values. "

tem limite de diálogo nisso ?? Esse problema é disso ???

Mauricio Cruz — 13/06/2013 08:21

Sim, por isso que você tem que usar a função SPBT_INITIALIZE para descobrir quantas tasks o seu sistema tem aberto, antes de implementar o paralelismo.

Navier Stokes — 26/09/2012 22:21

o ponto de cancelamento foi na função

AC_FLUSH_CALL_INTERNAL".

Navier Stokes — 26/09/2012 22:19

seguinte tive um dump depois de passar algumas vezes : segue o dump

RPERF_ILLEGAL_STATEMENT

TEXTO BREVE

STATEMENT "CALL FUNCTION .. DESTINATION/STARTING NEW TASK /IN BACKGROUND TASK"

SAPLOLEA

The program was probably... Function modules called CONVERSION_EXIT_XXXXX_INPUT/OUTPUT or USER_EXIT_XXXXX_INPUT.

etc etc etc .. alguém por favor me ajuda ?

abracos

Deocleciano Maranhão Neto — 29/08/2012 11:41

O código que enviei não ficou formatado ... acho que não inseri a tag corretamente !!!!

Será que ainda dá pra editar?

(Deletar Comentário)

Deocleciano Maranhão Neto — 27/08/2012 16:07

Gostei muito desse post, pois paralelismo é um assunto muito interessante em qualquer linguagem !!!

Só fiquei triste pois o exemplo não mostra várias entradas para ilustrar melhor o paralelismo. Eu fiz alguns ajustes no código para tentar testar ao meu gosto 😊

(A RFC deve ser feita em outro ambiente!)

```
REPORT  znto_assincrona_sbook.
```

```
*****
* VARIÁVEIS GLOBAIS (V_...)                                *
*****
```

```
DATA: v_tasks_sist      TYPE i,
      v_max_tasks      TYPE i,
      v_task_id        TYPE numc2,
      v_task_count     TYPE i,
      v_task_ativa     TYPE i.
```

```

*****
* TABELA INTERNA (T_...)
*****
DATA: t_sbook      TYPE TABLE OF sbook,
      t_sbook_aux  TYPE TABLE OF sbook,
      t_carriid    TYPE TABLE OF zes_carriid.

*****
* TABELA
*****
TABLES: sbook.

*****
* PARÂMETROS DE TELA:
*   SELECT OPTIONS (S_...)
*   PARAMETERS     (P_...)
*****
SELECTION-SCREEN BEGIN OF BLOCK b1 WITH FRAME.

SELECT-OPTIONS: s_carriid FOR sbook-carriid OBLIGATORY NO INTERVALS. "Ano
*SELECT-OPTIONS: s_carriid FOR sbook-carriid OBLIGATORY NO INTERVALS. "Ano
* Obs. Estou utilizando apenas como chave o campo carriid, pois o ambiente
* que estou fazendo testes não possui muitos registros

SELECTION-SCREEN END OF BLOCK b1.

*****
* START-OF-SELECTION
*****
START-OF-SELECTION.
  DATA: wa_carriid LIKE LINE OF s_carriid,
        wa_carriid TYPE zes_carriid,
        vl_num_registros TYPE i ,
        vl_num_linhas TYPE i ,
        vl_dados_task TYPE i ,
        vl_lin_carriid TYPE i ,
        vl_num TYPE char10 ,
        vl_mensagem TYPE char80.

* Obter o número de sessões disponíveis
CALL FUNCTION 'SPBT_INITIALIZE'
  IMPORTING
    free_pbt_wps          = v_tasks_sist
    max_pbt_wps           = v_max_tasks
  EXCEPTIONS
    invalid_group_name    = 1
    internal_error        = 2
    pbt_env_already_initialized = 3
    currently_no_resources_avail = 4
    no_pbt_resources_found = 5
    cant_init_different_pbt_groups = 6
    OTHERS                 = 7.

  IF sy-subrc = 0.
*** Mensagem de erro
    LEAVE LIST-PROCESSING.
  ENDIF.

* v_tasks_sist = 2.

  DESCRIBE TABLE s_carriid LINES vl_num_linhas.

  vl_dados_task = vl_num_linhas / v_tasks_sist.

  LOOP AT s_carriid INTO wa_carriid.

```

```

wa_carrid-i_carrid = wa_carrid-low.

APPEND wa_carrid TO t_carrid.

DESCRIBE TABLE t_carrid LINES vl_lin_carrid.
* Verifica se tabela esta com x dados para chamar nova RFC
  IF vl_dados_task = vl_lin_carrid .

      PERFORM znova_task.

      FREE t_carrid.
*
      ELSEIF ( vl_dados_task > vl_lin_carrid AND
              sy-tabix = vl_num_linhas ).
          PERFORM znova_task.

      ENDIF.
ENDLOOP.

IF sy-subrc IS INITIAL.
* Esperar até que todas as sessões sejam finalizadas. O número é decrementado na subrtina abai
* e incrementado quando a RFC é chamada.
  WAIT UNTIL v_task_ativa = 0.

*** Neste ponto você poderá utilizar os resultados retornados da RFC para continuar a lógica do
  DESCRIBE TABLE t_sbook LINES vl_num_registros.

  WRITE: vl_num_registros TO vl_num.

  CONCATENATE 'Tabela com ' vl_num ' registros' INTO vl_mensagem.

  WRITE :/ vl_mensagem.
ENDIF.

*****
* SUBROTINA
*****
FORM update_order USING name.

* Obter os resultados da RFC
  RECEIVE RESULTS FROM FUNCTION 'Z_RFC_ASSINC_SBOOK'
*   TABLES
*     t_sbook      = t_sbook_aux.
  IMPORTING
    t_sbook      = t_sbook_aux.

  APPEND LINES OF t_sbook_aux TO t_sbook.

  SUBTRACT 1 FROM v_task_ativa.

ENDFORM.          " UPDATE_ORDER

*&-----*
*&      Form  ZNOVA_TASK
*&-----*
*      text
*-----*
* --> p1      text
* <-- p2      text
*-----*
FORM znova_task .
*      Como não sabemos a quantidade de registros, devemos controlar quantas sessões serão gerad
*
*      Incrementar ao contador de sessões ativas

```



```

ADD 1 TO v_task_ativa.

*   Verificar se o número de sessões ativas está dentro do limite
IF v_task_ativa <= v_tasks_sist.

*   Cada sessão deverá ter um ID único
ADD 1 TO v_task_id.

*   Chamar a RFC em uma nova sessão
CALL FUNCTION 'Z_RFC_ASSINC_SBOOK'
  STARTING NEW TASK v_task_id
*   DESTINATION IN GROUP DEFAULT
  DESTINATION 'nome_do_destino'
  PERFORMING update_order ON END OF TASK
  EXPORTING
    t_carriid = t_carriid
  EXCEPTIONS
    OTHERS = 1.

IF sy-subrc = 0.
*   Se a RFC falhar, vamos tentar novamente e diminuir o número de sessões ativas. Cuidado pa
*   em loop infinito!!
  SUBTRACT 1 FROM v_task_ativa.
ELSE.
  EXIT.
ENDIF.

FREE t_carriid.
CLEAR vl_lin_carriid.
ELSE.
  SUBTRACT 1 FROM v_task_ativa.
ENDIF.
*   ENDDO.
ENDFORM.                " ZNOVA_TASK

```

Jesus — 04/06/2012 09:35

Pô eu gostei. Tá simples e objetivo. O resto é só pesquisar, mas a base tá aqui.

Bruno — 22/05/2012 10:26

A ideia é muito boa, mas o exemplo foi muito mauzinho... As melhores situações para usar paralelismo são processos que usem principalmente capacidade de processamento, e não base de dados. Este exemplo feito com um programa de computação matemática tipo um algoritmo para calcular uma raiz quadrada ou algo desse gênero os resultados seriam muito melhores. 18% quase não justifica a perda de tempo a implementar o paralelismo.

Daiane Zigiotta — 22/05/2012 19:41

Olá Bruno,

A ideia do Paralelismo Assíncrono é dividir a carga do programa em pacotes e que os dados executados em paralelo sejam independentes entre si, e isso pode se aplicar a uma seleção de dados que usa uma tabela no FOR ALL ENTRIES com MUITOS registros, por exemplo, pois dessa forma você poderá melhorar (e muito) a performance do seu programa, e é justamente sobre isso que eu queria falar neste post...

O seu exemplo pode ser aplicado no conceito de RFC Transacional (usando CALL FUNCTION... IN BACKGROUND TASK), pois é necessário chamar apenas um único processo separadamente. Infelizmente, no ambiente de testes que eu usei, existiam poucos registros então eu tive que me virar com o que eu tinha! 18%

não parece ser muita coisa, mas isso varia de acordo com cada ambiente e com a quantidade de dados.
Abs,

Antonio — 28/03/2012 16:56

Cade a estrutura da Function Z_RFC_ASSINC?

Daiane Zigiotta — 28/03/2012 17:10

Oi Antonio!

A Z_RFC_ASSINC faz uma seleção na tabela BSEG utilizando como condição o valor do parâmetro de importação I_GJAHR e depois retorna os registros na tabela T_BSEG.

Era essa sua dúvida?

Abs,

Mauricio Cruz — 29/03/2012 09:06

Ahm, a Daiane explicou e tal, mas a estrutura da RFC não importa muito para o exemplo. Tanto faz quais são os parâmetros, o funcionamento do paralelismo, que é o importante do post, é o mesmo.

Ah, e para quem não sabe: quando falamos "RFC", estamos falando de uma função criada pela SE37/SE80 (e derivados do leite), mas que na primeira aba de características deve ter a opção "módulo de função remota" marcada.

Gabriel Tognoli — 14/03/2012 14:18

Prezados,

Gostei do post e o incrível é que estou estudando o uso de RFC para um possível desenvolvimento, mas acho que vocês não estão cumprindo com o prometido! Lembro de um post em que vocês comentam que estão abolindo o uso de códigos procedurais.

Cade o ABAP Objects? cadê as classes, os objetos, etc...??

Daiane Zigiotta — 14/03/2012 14:46

Oi Gabriel! Obrigada pelo comentário! A ideia do post era somente mostrar o conceito de RFC Assíncrona, mas se você preferir, ao invés de usar CALL FUNCTION... PERFORMING (subrotina) você pode usar CALL FUNCTION... CALLING (método).

Abraços!

Mauro Laranjeira — 15/03/2012 12:19

=O essa eu não sabia.

Vivendo a aprendendo, rs

Mauricio Cruz — 14/03/2012 14:20

Olá Gabriel!

Na verdade, quem falou que ia deixar de programar de forma procedural fui eu, nesse post [aqui](#)! A idéia do ABAPZombie não é trocar todas as dicas que damos aqui para OO, cada um dos membros trabalha da maneira que achar melhor. Mas o Mauro e eu estamos preparando algumas coisas divertidas para as próximas semanas. Acho que você vai gostar 😊

Abraços!

Rodrigo — 27/04/2012 12:06

Aee Mauricio... vc sabe minha opinião! heheh

Eu odeio OO mas tudo bem... não programo mais mesmo!! heheh

Gabriel Tognoli — 14/03/2012 14:59

Maurico/Daiane,

Meus amigos, obrigado pelo retorno!

Comentei no intuito de desenvolvermos uma comunidade cada vez mais forte, deixando de lado velhos hábitos e rotinas, que fazíamos e que fazemos até hoje.

Eu mesmo, sempre que penso em uma solução, me vem na cabeça o maldito procedural.

Farei uma comparação com o inglês... Estudar no Brasil ou no exterior talvez não seja tão diferente, mas ao estar em outro país, nos vemos obrigados a entender e estudar, pq TUDO é inglês.

Também estou adotando esta postura, mudar os meus códigos procedurais para OO. Confesso que ainda carrego comigo as velhas manias, acho até que vou recorrer a uma lobotomia! rs

Mauro Laranjeira — 14/03/2012 09:16

aEEE Dai,

Parabéns, post Legal Estadual e Bonito (Acho que só a Dai vai entender)

Curti muito, muito mesmo.

Abs.

Daiane Zigiutto — 14/03/2012 17:16

Valeu Mauro! 😊

Para quem quiser entender também, assista esse vídeo -> <http://www.youtube.com/watch?v=IZCRuy7hkpA>

uhsaushuahsuhauhsua

Custodio — 12/03/2012 19:50

Ou eu estou bebado, ou esta faltando um monte de codigo nesse post, nao?

Daiane Zigiotto — 12/03/2012 22:13

Quando fui passar o código-fonte acabou cortando uma parte, mas já está corrigido!!!

Miguel — 12/03/2012 21:51

Hahaha!

Estava me perguntando a mesma coisa!

Quando corrigido será um post muito útil!

Daiane Zigiotto — 12/03/2012 22:25

Já corrigi o código-fonte, obrigada por avisar!