

## Códigos pequenos... e úteis! – Parte 1 – TVARV

20/01/2011 18:16

É galera! Eu e o Mauro estamos um pouco ausentes porque estamos preparando uma sequência de posts **bem** interessantes 😊 .

Códigos simples e úteis que todo mundo precisa no dia-a-dia. Chega de ficar escrevendo aquele **mesmo** pedaço de código toda semana.

Eu terminei o primeiro hoje, e acredito que ainda terão mais ou menos umas 6 ou 7 partes, antes de somarmos todos os códigos e gerarmos algo parecido com a [ABAP Zombie Screen Examples](#).

Então vamos lá!

**TVARV!** Aff, cansei de fazer aquele mesmo select, aquela mesma lógica para pegar os dados do range... Então eu fiz uma classe simplezinha para buscar valores da TVARV.

O que? Você não sabe o que é TVARV?

*TVARV: Tabela de parâmetros do SAP, para ser utilizada pelos projetos. A idéia é ter a maioria dos parametrôns que podem ser alterados durante o funcionamento do programa num só local, evitando assim, retrabalho desnecessário em programas custom.*

O código fica assim:

```
REPORT z_tvarv.

* Essa é a include que contém a classe de acesso à TVARV
INCLUDE zaz_tvarv.

* Parametro e Range de Exemplo
DATA: p_param TYPE char10.
DATA: r_range TYPE RANGE OF char10.

* Objeto da TVARV
DATA: o_tvarv TYPE REF TO lcl_tvarv.

*-----*
* START-OF-SELECTION
*-----*
START-OF-SELECTION.

* Aqui estamos criando o Objeto e acessando a TVARV.
* PREFIX: valor que vem antes do separador. Exemplo: em GAP1-VARIAVEL1
*         o valor de PREFIX seria GAP1, pois todas as variáveis devem
*         iniciar com GAP1 para o programa GAP1.
* SEPARATOR: O Separador. Não precisa de exemplo né? :P
*-----*

CREATE OBJECT o_tvarv
EXPORTING
    prefix    = 'AUDI' "Exemplo de Prefixo
    separator = '_'.

* Busca Valor de um paraâmetro simples.
o_tvarv->get_parameter(
EXPORTING
```

```

    suffix = 'KONTENPLAN' "Sufixo qualquer de Parâmetro
IMPORTING
    value = p_param ).

* Busca os Valor de um SelOption da TVARV, e ainda monta o range.
* Agora ficou fácil :D
o_tvarv->get_seloption(
    EXPORTING
        suffix = 'KONTENPLAN' "Sufixo qualquer de Select Option
    IMPORTING
        value = r_range ).

```

E a include? Relax, aqui está o código: [ABAP\\_ZOMBIE\\_GEAR-TVARV.txt](#)

A idéia é bem simples, e funciona para qualquer tipo de acesso que você precise para a TVARV. Eu pretendo, no futuro, expandir essa lógica para gravar valores na TVARV, e também para retornar valores de mais de um prefixo. Mas o que está feito, já ajuda 😊

Precisou de tvarv nos próximos programas? É só colocar INCLUDE zaz\_tvarv.

\* Aliás! O ZAZ não é de zueira, é de Z ABAP ZOMBIE!



Abraços!

## Comentários

**Adriano de Oliveira Colenghi Baptista** — 17/03/2023 10:23

Olá.

Dúvida:

\* Essa é a include que contém a classe de acesso à TVARV

INCLUDE zaz\_tvarv.

Onde temos o código desse include?

**Marcelo** — 25/02/2016 15:20

Gostei muito desse Post.

Já coloquei nas minhas anotações para usa-la..!!!

abraço

**Vanessa** — 11/06/2014 10:04

Parabens pelo blog! Muito interessante e produtivo. Linguagem simples até pra quem não entende nada de programação.

Posso isugerir um post? Como fazer uma boa especificação funcional para um ABAP?

Sou consultora junior e acredito que isso iria ajudar bastante tanto consultores quanto abaps.

Obrigada,

**Mauricio Cruz** — 11/06/2014 10:29

Sugestão anotada Vanessa.

Vlw!

**Nuno Godinho** — 26/12/2011 08:43

Olá zombies!!

Concordo que deve haver um lugar centralizado para guardar constantes mas não sou apologista de usar a TVARV para isso. Por uma única razão: a tabela, tendo apenas um identificador (NAME), não permite fazer uma organização decente das constantes.

Para mim a solução ideal passa por uma tabela (Z) cuja chave permita identificar:

1. **contexto** – RICEF ou nome do programa ou desenvolvimento que usa as constantes. que pode bem ser um nome genérico nos casos em que serão usadas em vários sítios);
2. **campo** – nome do campo que vai usar estes valores (ex: BUKRS);
3. **ocorrência** – permite distinguir mais do que uma ocorrência do mesmo campo no mesmo contexto. Por exemplo, dentro de um programa podemos fazer uma coisa se o BUKRS for 1111 e outra coisa se o BUKRS for 2222 ou 3333. Que para simplificar pode ficar vazio nos casos de ser apenas uma;
4. **linha** – para poder guardar RANGES, permitir múltiplas linhas para cada ocorrência.

E depois é só definir uma classe que faça para esta tabela Z o mesmo que a que vocês definiram faz para a TVARV. Mas assim permite que a coisa fique mais organizada... e a longo prazo dará frutos :))

Abraços,  
Nuno

**maurocesarsl** — 26/01/2011 04:43

=D usei deste código hoje... testado é funcionando.. pq reinventar a roda?! Dei logo um GET XD