

Códigos de Zumbis para Zumbis

22/03/2012 22:47

Nesta seção estão listados os códigos fontes reutilizáveis desenvolvidos pelos membros do ABAPZombie, e, possivelmente, pelas pessoas que visitam o site.

Os códigos estão armazenados no Github, no formato de arquivos ".NUGG", do SAPLink. Se você não entendeu nada, clique [aqui](#).

Se você tiver algum código que queria compartilhar, você pode fazer um "pull request" no repositório do github, para que possamos revisar e adicionar seu código ao repositório (por revisar leia-se "se ele ativa e executa". Cada um que cuide da qualidade do seu código" :P). Os únicos requisitos são

- Seu código **deve** estar num formato que o SAPLink possa ler. Se não for para mandar no formato do SAPLink, nem envie, pois não vamos olhar.
- Você também precisa enviar um arquivo de texto explicando como usar o seu código.
- Ah, e seu nome vai ficar como criador do código, fique tranquilo 😊

Repositório e Envio de Códigos

Repositório ABAP ToolBox no Github, contendo todos os códigos do ABAPZombie: [mauriciorcruz/ABAP-Toolbox](#)

E-mail para enviar seus projetos, caso não tenha e não queria criar conta no Github (o que eu acho uma pena, sinceramente): contato@abapzombie.com

Projetos Publicados

- [APO BAPI Class](#), por Mauricio Cruz

Classe para simplificar o acesso em sequência de BAPIs no APO, um ambiente onde grande partes dos acessos à informações é feito através das BAPIs ao invés de SELECTs diretos na tabela.

- [Class for BDC Mapping](#), por Mauricio Cruz

Como eu estava cansado daquele FORM de sempre que aceitava 5 parâmetros de entrada, e de códigos com um monte de valores e nomes de campo que não faziam sentido, estou tentando criar uma classe que deixe o código mais amigável. Ex.:

```
CREATE o_bdc EXPORTING transaction = 'XX01'
                        dismode   = 'N'.

o_bdc->add_main_scren( program = 'PROGSTANDARD'
                      dynpro  = '1000' ).
```

```

o_bdc->add_fcomm( fnam = 'PARAMETER'
                  fval = 'X' ).

o_bdc->add_fcomm( fnam = 'RADIOBUTTON'
                  fval = 'X' ).

* Muda o radiobutton na Tela
o_bdc->add_radiobutton( ).

* Não precisa falar quem é a main screen de novo
o_bdc->add_main_screen( ).

* Abre lista de Valores num Range
o_bdc->add_okcode( '=%010' ).

* Tela nova
o_bdc->add_screen( program = 'PROGSTD2'
                  dynpro  = '1234' ).

o_bdc->add_fcomm( fval = 'BLABLA-LOW(01)'
                  fnam = 'VALOR' ).

* Envia OKCode do Pagedown - para listas ou Table Controls
o_bdc->add_pagedown( ).

* Se fosse preciso chamar a mesma tela de novo que não a main
o_bdc->add_last_screen( ).

* Aceita os valores entrados no pop-up de Ranges
o_bdc->add_accept( )☐

o_bdc->add_main_screen( ).
o_bdc->add_execute( ).
o_bdc->run

```

Esse é só um exemplo, mas deu pra entender a idéia.. 😊 O negócio é ficar fácil de ler, e aproveitar das técnicas de OO para executar várias transações em sequência. Ainda não funciona para pastas, fiz somente para CALL TRANSACTIONS.

- [Performance Examples](#), por Mauricio Cruz

Programa com os exemplos de performance deste [post aqui](#).

- [ABAPZombie Screen Examples](#), por ABAPZombie

O primeiro código que compartilhamos pelo ABAPZombie, [nesta série de posts aqui](#).