

## ABAPZombie Guide to ABAP – Parte 17 – BREAK-POINT

02/03/2011 08:30

E lá vamos nós com esse comandinho criador do caos: o **BREAK-POINT**.

Bom, este comando “para” a aplicação e abre o Debug. Se for em background, ele loga que o Debugger foi acionado no log da aplicação mas continua o processamento (ou seja, em back não atrapalha a execução).

Como no comando [ASSERT](#), você pode ainda definir um ID para o BREAK-POINT, e controlar sua ativação pela transação SAAB.

### E porque esse comando cria o caos?

Porque sempre tem *aquele* ABAP que esquece esse comando no meio do Report, da Exit, do Online... e daí o funcional vai testar e.. BUM, para no Debugger.

Portanto, só use este comando entre um IF testando se é o seu usuário executando, ou utilizar a macro BREAK (nome do usuário), que eu mostrei [neste post](#). **O ideal é não esquecer e remove-lo antes de liberar o programa para testes**, mas pelo menos usando isso você não atrapalha ninguém, só você mesmo 😊

Abraços a todos aqueles que já ouviram um funcional/usuário falando “ué, porque abriu essa tela estranha quando eu executei?” 😊

---

## Comentários

Luiz Henrique — 18/02/2013 08:07

Um ABAP que se preza antes de liberar seus programas para testes sempre os verifica com transação SLIN...

Mauricio Cruz — 18/02/2013 11:02

E um ABAP LIKE A BOSS, sempre encontra zero erros em tudo nessa transação 😊