

## **Workshop Abap**

Atenção! Esta lista de exercícios faz parte do Workshop de Abap, não é um material de auto estudo. Deve ser feita sob a orientação de um instrutor.

Use o seu número de aluno para nomear os seus objetos, assim:

**Zxx\_YYYYYYYYYYYYYYYYYYYYYYYY**

Onde:

**Z** : obrigatório na maioria dos objetos criados pelo cliente SAP (Z ou Y, usaremos Z neste curso);

**xx** : o seu número de aluno;

**\_** (travessão) : não é obrigatório, mas vai ajudar na busca de programas do curso;

**YYYYYYYY** : livre, mas não pode conter espaços;

Exemplos: Z05\_EXERCICIO\_3\_1, Z12\_LISTA\_CLIENTES

Observação: para nomes de programas e identificadores dentro do código (palavras reservadas e nomes de variáveis), o abap não diferencia minúsculas de maiúsculas.

## **Tópico 1 – Introdução, programas Report e telas de seleção**

Observação: em geral os programas do tipo Report seguem a seguinte divisão:

- ✓ Declarações globais (tipos, variáveis e classes)
- ✓ Tela de seleção e seus eventos
- ✓ START-OF-SELECTION
- ✓ END-OF-SELECTION.
- ✓ Definições de rotinas (FORM)

### **1.1 - Tela de seleção, declaração de variáveis:**

- Montar tela de seleção com 2 números inteiros (PARAMETERS);
- Declarar 4 variáveis inteiras (DATA). Cada uma armazenará o resultado de uma operação com os parâmetros de entrada: soma, diferença, produto e quociente (resultado da divisão do primeiro número pelo segundo número);
- Neste exercício vamos trabalhar somente com variáveis inteiras;

- Lembre que não existe divisão por zero. Caso ocorra este cenário, imprima o literal "INEXISTENTE". Será necessário uso do comando IF ou tratar a exceção com TRY...CATCH:

```

TRY.
    <expressão aritmética>
CATCH cx_sy_zerodivide.
    <tratamento do erro>
ENDTRY.

```

- Imprimir (WRITE) os dois números digitados e o resultado das operações, conforme o lay-out abaixo:

```

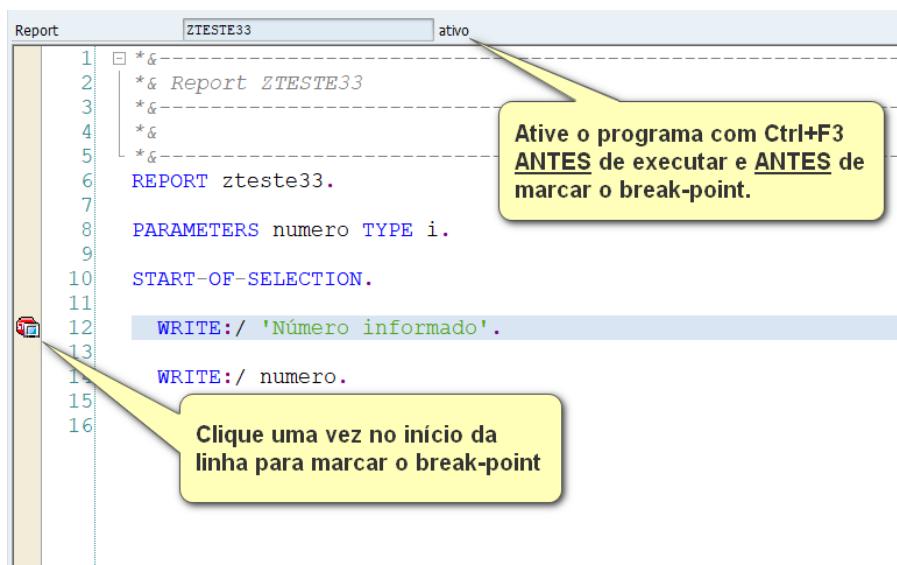
Número 1: 999
Número 2: 999
Soma: 999999
Diferença: 999999
Produto: 999999
Divisão: 999999

```

- Atenção para os procedimentos de testes abaixo. Ele será necessário em quase todos os exercícios deste curso:**

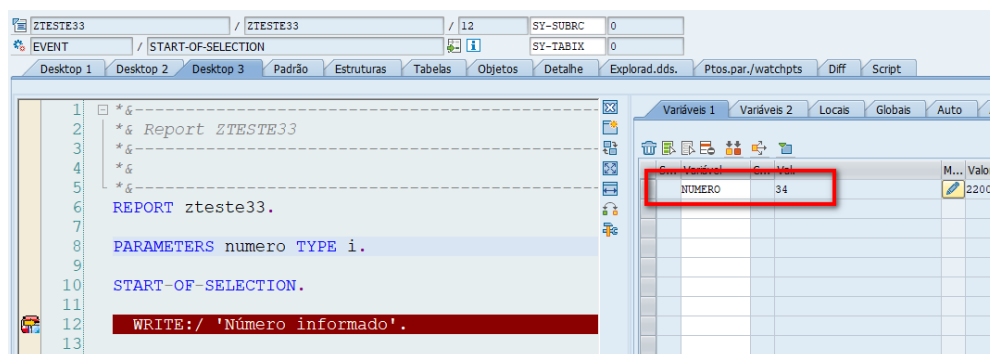
### **Procedimentos de testes:**

- Ativar o programa (Ctrl+F3) e elimine os erros de ativação. Não ative programas com erros!
- Antes de executá-lo** (F8), coloque um break-point na primeira linha executável (primeira linha do evento START-OF-SELECTION), conforme o exemplo abaixo:



- Tecele F8 para executar o programa;
- Tecele F5 na janela de debug para executar cada linha do programa;

e) Mostre na janela da direita todos os valores das variáveis de tela e do programa.



Cenário de testes:

1º número	2º Número	Soma	Diferença	Produto	Quociente
2	3	5	-1	6	1
3	2	5	1	6	2
0	17	17	-17	0	0
1	3	4	-2	3	0
6	2	8	4	12	3
673	201	874	472	135.273	3
45	0	45	45	0	INEXISTENTE
-17	-3	-20	-14	51	6
-4	8	4	-12	-32	-1
2	-2	0	4	-4	-1

Obs.:

- apesar de não ser obrigatório, é uma boa prática colocar o nome do evento START-OF-SELECTION;
- o comando WRITE está em desuso para exibição de valores em tela para o usuário, mas como ferramenta de aprendizado é mais prático do que outros recursos de exibição (janelas popup e ALV, vistos mais adiante neste curso).

## 1.2 - Tela de seleção, loop, IF

- Tela de seleção: 2 números inteiros e obrigatórios;
- Fazer as verificações abaixo. Em caso de erro, encerrar o programa (comando STOP):
  - os dois números devem ser positivos e não podem ser zero; caso contrário → Erro;
  - o primeiro número deve ser menor do que o segundo número; caso contrário → Erro;

- a diferença entre o segundo número e o primeiro número deve ser pelo menos 5 (cinco). Se for menor do que 5 → Erro;
- Se não ocorrer nenhum erro na etapa anterior, use o comando WHILE ... ENDWHILE para imprimir as duas listas abaixo:
  - Lista 1: imprimir uma sequência de números inteiros começando do primeiro número até o segundo número. Por exemplo, se o primeiro número for 5 e o segundo número for 13, imprimir: 5 6 7 8 9 10 11 12 13;
  - Lista 2: imprimir a mesma sequência anterior, mas em ordem inversa, do segundo número até o primeiro número. Por exemplo, se o primeiro número for 5 e o segundo número for 13, imprimir: 13 12 11 10 9 8 7 6 5;
- Calcular e imprimir a soma da lista de números;
- Calcular e imprimir a média aritmética simples da lista de números;
- Calcular e imprimir a multiplicação do terceiro item da lista 1 pelo quinto item da lista 2. Por exemplo, se os números informados forem 5 e 13, temos:
  - Lista 1: 5 6 7 8 9 10 11 12 13 → terceiro item é 7;
  - Lista 2: 13 12 11 10 9 8 7 6 5 → quinto item é 9;
  - Para este exemplo o resultado da multiplicação seria 63 (7 x 9).
- Não utilize variáveis estruturadas neste exercício: listas, vetores, matrizes, estruturas, arrays, tabelas internas.

Cenário de testes:

1º número	2º Número	Resultado
2	3	Erro: diferença menor do que 5
3	2	Erro: primeiro número maior do que o segundo
0	17	Erro: número inválido
5	13	Imprimir conforme exemplo do enunciado e produto = 63
45	0	Erro: número inválido
-17	-3	Erro: número inválido
-4	8	Erro: número inválido
2	-2	Erro: número inválido

Desafio: imprima novamente uma lista crescente, do menor número até o maior número, mas somente números pares. Dica: como obter o resto de uma divisão por 2?

### 1.3 - Tela de seleção, substrings, operações com datas

- Tela de seleção (2 campos - PARAMETERS)
  - Nome com 30 posições caracter (obrigatório);

- Campo no formato data (obrigatório);
- Use o evento `INITIALIZATION` para inicializar a data da tela com o dia de amanhã;
- Imprimir:
  - Data do dia anterior à data informada;
  - Data com o primeiro dia do mês seguinte. Exemplos:
    - se a data informada for 17.08.2026, imprimir 01.09.2026;
    - se a data informada for 08.12.2021, imprimir 01.01.2022;
  - Número de dias entre a data informada e a data atual (somente se a data informada for maior que a data atual);
  - Tamanho (em caracteres) do nome (função `STRLEN`);
  - 3 primeiros caracteres do nome informado na tela de seleção;
  - 2 últimos caracteres do nome informado na tela de seleção;
  - Imprimir a data informada por extenso, por exemplo, 25 de janeiro de 2025. Para encontrar o mês utilize o comando `IF` ou o comando `CASE`.

#### 1.4 - Tela de seleção, evento `AT SELECTION-SCREEN`, variáveis de ambiente

- Fazer uma tela para cadastro de usuário, com 5 parâmetros:
  - Nome (char40) – obrigatório --> JOSÉ
  - Sobrenome (char30) – obrigatório --> ANTUNES: ANTUN0624
  - Data inicial – obrigatório – inicializar com a data atual;
  - Data final
  - Checkbox "Aplicar data no código", com o valor default ' ' (desmarcado).
- Consistir no evento `AT SELECTION-SCREEN`, com mensagens do tipo "E" em caso de erro:
  - a data inicial deve ser maior ou igual a data atual. Se não for, mostrar mensagem de erro "A data inicial deve ser maior ou igual a data atual";
  - se a data final for informada, ela deve ser maior ou igual a data inicial. Caso contrário, mostrar a mensagem de erro: "Data final deve ser maior ou igual a data inicial";
  - para as mensagens, criar uma classe de mensagens `Zxx` na transação `SE91`, onde "xx" é o seu número de usuário;
- Montar e imprimir um código de usuário:
  - Se o checkbox estiver marcado, o código de usuário deve ser formado pelos 5 primeiros caracteres do sobrenome + mês + ano da data inicial informada na tela (mês e ano com dois dígitos cada um). Não é necessário verificar o tamanho do sobrenome;

- Se o checkbox estiver desmarcado, o código de usuário deve ser formado pelo primeiro caracter do nome + o sobrenome;
- Em ambos os casos, eliminar os espaços em branco do código formado, usando o comando CONDENSE;
- Imprimir a quantidade de dias entre a data inicial e a data final (se a data final for informada);
- Imprimir o código do usuário (usuário de logon do sap - SY-UNAME) que está fazendo o cadastro, a data atual (SY-DATUM), a hora atual (SY-UZEIT) o nome do programa (SY-REPID) e a transação em execução (SY-TCODE).

## 1.5 - Tela de seleção, evento AT SELECTION-SCREEN, radio-button, checkbox

- A data inicial é obrigatória;
- A data final é opcional, mas se for informada:
  - deve ser menor ou igual à data atual (emitir mensagem de erro caso contrário);
  - deve ser maior que a data inicial (emitir mensagem de erro caso contrário);
- Dos três checkboxes da tela, pelo menos um dos dois primeiros DEVE estar marcado (emitir mensagem de erro caso contrário);
- Se o checkbox "Imprimir data por extenso" estiver marcado, imprimir as duas datas conforme o formato indicado no radio-button selecionado. Atenção: na primeira e terceira opções, o nome do mês deve vir com 3 letras;
- Se o checkbox "Imprimir calendário do ano" estiver marcado, imprimir todo o calendário do ano da data inicial (a partir de 01 de janeiro) um dia por linha, saltando uma linha a cada mês;

- Se o checkbox "Verificar ano bissexto" estiver marcado, verificar se o ano da data inicial é bissexto. Se sim, imprimir "XXXX é ano bissexto", onde XXXX é o ano da data inicial.

## 2.1 - Recuperando dados de uma tabela – registro único

- Fazer uma tela de seleção com o ID do cliente (tabela SCUSTOM) – usar PARAMETERS
- Imprimir o nome, cidade e idioma do cliente selecionado
- Usar SELECT...SINGLE
- Observar rotina de conversão no idioma: consultar o registro na transação SE16 e ver o conteúdo do campo LANGU. Por que o valor listado pelo seu programa é diferente daquele consultado na tabela SCUSTOM (transação SE16)? Use agora a transação SE16N e faça a mesma consulta.

## 2.2 - Recuperando dados de várias tabelas:

- Fazer uma tela de seleção com os seguintes campos, todos eles não obrigatórios:
  - Código da companhia aérea (tabela SCARR - "Denominação breve da companhia aérea");
  - Número do voo (tabela SPFLI - "Nº da conexão de voo");
  - Data do voo (tabela SFLIGHT - "Data do voo");
  - Número da reserva (tabela SBOOK - "Nº de marcação");
- Os campos acima também são chaves primárias das tabelas citadas;
- Faça a leitura da tabela correspondente **SOMENTE** se todos os campos da chave primária da tabela forem informados (SELECT SINGLE);
- Se a leitura for feita e não encontrar registro, emitir mensagem (comando WRITE):
  - "Cia aérea não encontrada" (SCARR)
  - "Horário de voo não encontrado" (SPFLI)
  - "Voo não encontrado" (SFLIGHT)
  - "Reserva não encontrada" (SBOOK)
- Imprimir, se encontrar, para cada tabela:
  - SCARR: código da cia aérea, nome e site;
  - SPFLI: código da cia aérea, número do voo, cidade e país de origem, cidade e país de destino, duração do voo, nome do aeroporto de origem e nome do aeroporto de destino (tabela SAIRPORT - fazer 2 outros SELECT...SINGLE);
  - SFLIGHT: código da cia aérea, número do voo, data do voo, preço e moeda do voo, tipo de avião, velocidade de cruzeiro e fabricante do avião. Atenção: os dois últimos campos estão na tabela SAPLANE, fazer outro SELECT...SINGLE;
  - SBOOK: código da cia aérea, número do voo, data do voo, número da reserva, número do passageiro, peso da bagagem.

IMPORTANTE!! Verifique o conteúdo de cada tabela na transação SE16 ou SE16N, antes de fazer a codificação de cada leitura.

## 2.3 - Alterando dados de um registro

- Tela de seleção:
  - Código do passageiro (SCUSTOM-ID), obrigatório;
  - Novo e-mail do passageiro (SCUSTOM-EMAIL), obrigatório;
- No evento AT SELECTION-SCREEN:
  - ⊖ Verificar a existência do código do passageiro. Se não existir, emitir a mensagem de erro "Passageiro não cadastrado".
  - Consistir o novo e-mail, usando a rotina VALIDA\_EMAIL, logo abaixo;
- START-OF-SELECTION:
  - Atualizar (comando UPDATE ou MODIFY) o registro, alterando o e-mail.
- Observação importante: como regra geral, não atualizamos tabelas standard diretamente com comandos UPDATE, MODIFY, INSERT ou DELETE. Para este propósito, o SAP fornece outras ferramentas (vistas mais adiante).

```
*&-----*
*&   Para informações e tutorial de expressões regulares acesse:
*&   http://www.regular-expressions.info
*&-----*
FORM valida_email USING p_email
                     CHANGING email_ok.

DATA: go_regex TYPE REF TO cl_abap_regex,
      go_matcher TYPE REF TO cl_abap_matcher.

CREATE OBJECT go_regex
EXPORTING
  pattern      = '\w+(\.\w+)*@(\w+\.)+(\w{2,4})'
  ignore_case  = abap_true.
go_matcher = go_regex->create_matcher( text = p_email ).
IF go_matcher->match( ) IS INITIAL.
  CLEAR email_ok.
ELSE.
  email_ok = 'X'.
ENDIF.

ENDFORM.                "valida_email
```

## 2.4 - Inclusão, exclusão e alteração de registros

- Fazer um programa para manter a tabela SCARR. Campos: código, nome e site;
- Deixar somente o código obrigatório (SCARR-CARRID);
- Criar 3 radiobuttons: Criar, Alterar, Excluir. Colocar os radio-buttons dentro de uma moldura com o título "Operação". A opção "Alterar" deve ser o valor inicialmente marcado;
- Ao teclar F8 executar a ação selecionada no radiobutton;



- Observe que na alteração e exclusão é obrigatório que a cia aérea exista. Em caso de erro emitir a mensagem "Companhia aérea inexistente";
- Na inclusão a cia aérea não deve existir. Na tentativa de incluir uma cia aérea já existente, emitir a mensagem de erro "Companhia aérea já cadastrada";
- Apenas o código da cia e o nome são campos obrigatórios para inclusão e alteração. Na exclusão, somente o código é obrigatório;
- Use rotinas separadas (PERFORM) para cada operação,
- Observação importante: como regra geral, não atualizamos tabelas standard diretamente com comandos UPDATE, MODIFY, INSERT ou DELETE. Para este propósito, o SAP fornece outras ferramentas (vistas mais adiante).

### 3.1 - Recuperando dados em tabelas internas

- Tela de seleção: seleção múltipla do código do cliente (SELECT-OPTIONS);
- Fazer três rotinas para recuperar registros da tabela SCUSTOM em uma tabela interna. A primeira rotina usará tabela interna sem header line; a segunda usará com header line e a terceira rotina usará field-symbols.
- Em cada rotina imprimir o código, nome e cidade dos clientes a partir da tabela interna criada, usando uma work-area (primeira rotina), a linha de cabeçalho (segunda rotina) ou field-symbol (terceira rotina);
- Lembre: field-symbol é mais indicado. Tabela interna com linha de cabeçalho está obsoleta!

### 3.2 - Atualização de tabela transparente (do banco de dados) a partir de tabela interna

a) Consulte a tabela SFLIGHT na transação SE16 ou SE16N e anote em papel, bloco de notas, MS Word ou outro local:

- Selecione 3 ou mais registros com o valor de SEATSOCC superior a 60% do valor de SEATSMAX;
- Selecione 3 ou mais registros com o valor de SEATSOCC igual ou inferior a 60% do valor de SEATSMAX;

b) Tela de seleção:

- Código da companhia aérea, seleção múltipla, obrigatório;
- Data do voo, seleção múltipla, obrigatório. Usar o evento INITIALIZATION para inicializar a faixa de datas com o mês atual: coloque na data inicial o primeiro dia do mês e na data final o último dia do mês;

c) Aumentar em 15% o preço da passagem (campo SFLIGHT-PRICE) dos registros selecionados cuja ocupação na classe econômica seja inferior ou igual a 60%;

d) Sugestão:

- carregar para uma tabela interna os registros de acordo com a tela de seleção;

- eliminar dela os voos com ocupação superior a 60%. Usar os campos da classe econômica: SEATSMAX – número máximo de assentos; SEATSOCC – número de assentos ocupados;
- Em seguida, aplicar o aumento NA TABELA INTERNA;
- NÃO EXECUTAR MODIFY na tabela transparente!!!
- Teste o seu programa usando o debug, para verificar na tabela interna de resultado os registros levantados no item a). Somente avance para o item e) se sua tabela interna de resultado tiver SOMENTE as linhas que deveriam ser alteradas na tabela transparente SFLIGHT;

e) Coloque o MODIFY SFLIGHT from tabela\_interna no seu programa, FORA do loop;

f) Após a execução do seu programa, consulte novamente na transação SE16/SE16N, os registros selecionados no item a);

- Observação importante: como regra geral, não atualizamos tabelas standard diretamente com comandos UPDATE, MODIFY, INSERT ou DELETE. Para este propósito, o SAP fornece outras ferramentas (vistas mais adiante).