

Academia ABAP With

Hana & Fiori



### Objetos ABAP

o Variáveis

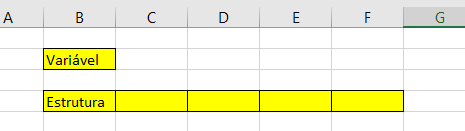
| **Tipo** | **Função** |
| --- | --- |
| **ANY** | Qualquer tipo de dados |
| **C** | Campo de texto com um comprimento genérico |
| **CLIKE** | Estruturas semelhantes a caracteres (c, d, n, t, sequência e estruturas planas semelhantes a caracteres); em programas não-Unicode também x, xstring e qualquer estrutura plana |
| **CSEQUENCE** | Tipo de texto (c, string) |
| **DATA** | Qualquer tipo de dados |
| **DECFLOAT** | Número decimal de ponto flutuante |
| **N** | Texto numérico com comprimento genérico |
| **NUMERIC** | Numérico (i (b, s), p, decfloat16, decfloat34, f) |
| **OBJECT** | Qualquer tipo de objeto (classe raiz da hierarquia de herança) |
| **P** | Campo de valor com comprimento genérico e número genérico de casas decimais |
| **SIMPLE** | Tipo de dados elementar, incluindo tipos estruturados com componentes planos exclusivamente para caracteres |
| **X** | Campo de bytes com comprimento genérico |
| **XSEQUENCE** | Campo de bytes (x, xstring) |

**DATA**: v\_variavel\_1 **TYPE** c, v\_variavel\_2(10) **TYPE** c, v\_variavel\_3 **TYPE** mara-matnr, v\_variavel\_4 **TYPE** n, v\_variavel\_5 **TYPE** p **DECIMALS** 2, v\_variavel\_6 **TYPE** d, v\_variavel\_7 **TYPE** t,

v\_variavel\_8 **TYPE** float, v\_variavel\_9 **TYPE** i, v\_variavel\_10 **TYPE** string.

o Estruturas

Estruturas são um conjunto de variáveis, ou seja, ao invés de podermos armazenar apenas uma informação por vez no objeto, na estrutura podemos inserir várias informações limitadas a uma linha, conforme o exemplo do no Excel abaixo:

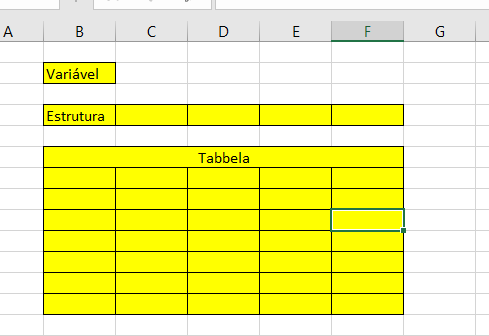


**DATA**: e\_estrutura **TYPE** mara.

Conforme podemos notar, a estrutura tem várias colunas, mas contém apenas uma linha, esse tipo de objeto é ideal na hora de selecionar valores únicos em tabelas, ou seja, apenas uma linha da tabela, ou para armazenar as informações durante a execução do código ABAP, ela pode ser criada com base em dados de tabelas ou até mesmo criada internamente no programa ABAP.

o Tabelas internas / Tabelas Transparentes

Diferente da variável e da estrutura, uma tabela interna pode ter várias colunas e também várias linhas, conforme o exemplo no Excel apresentado abaixo:



DATA: t\_tabela **TYPE TABLE OF** mara.

Conforme vemos no modelo acima, uma tabela é a soma de várias estruturas, podemos então entender que a sequência lógica dos objetos primários do ABAP é: “Váriavel, estrutura e depois tabela interna”, conforme também o desenho acima.

o Constantes

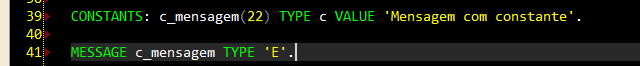
Com a mesma configuração da variável, podendo ser de todos os tipos, tamanhos, porém ao usar constantes, nós estamos definindo textos fixos dentro de um código o que são chamados de “literais”, uma literal sem constante é considerada uma má prática de programação, pois a utilização de constantes facilita na hora de dar manutenção ao programa de forma geral.

Abaixo segue um exemplo de um código usando literais sem constantes, os

chamados “Hard Codes”:



MESSAGE 'Mensagem Literal' TYPE 'E'. Exemplo com constante:



CONSTANTS: c\_mensagem(22) TYPE c VALUE 'Mensagem com constante'. MESSAGE c\_mensagem TYPE 'E'.

Conforme o exemplo acima, usamos uma constante para criar a mensagem, especificamos exatamente o tamanho da mensagem e inserimos os tipos, exatamente como uma variável, porém, essa de valor fixo e imutável.

* Ranges

Diferente de valores fixos únicos como variáveis, os ranges podem determinar um intervalo “de/até” de informações a serem usadas internamente, por exemplo.

Podemos buscar uma data de: 01/01/2019 até 31/08/2019, o “de” seria a primeira data e o “para” seria a segunda, vejamos na aula mais detalhes de como funcionam os ranges.

* Field Symbols

Embora sejam um pouco complicados para esta parte do curso, vamos já nos ambientar com os “ponteiros”. Através de um ponteiro é possível transformar a informação de um objeto para essa ser usada em diversos locais, como esse conceito é um pouco complicado, vamos entender melhor durante a explicação no vídeo, pois voltaremos a usar esse conceito um pouco mais tarde.

* Parameter

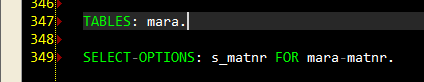
Para possibilitar a entrada de dados de usuários em programas, podemos usar o comando PARAMETER, nele criamos uma entrada personalizada de acordo com suas configurações técnicas, também igual as variáveis em tipos, tamanhos, etc. Esses campos só podem aceitar um valor de cada vez, respeitando sua limitação e tipo associado.



PARAMETERS: p\_campo1(10) TYPE c.

* Select Options

Ao contrário do PARAMETER, o SELECT-OPTIONS é usado para mais de um valor por vez em uma entrada de usuário, ou seja, ele permite que o usuário informe vários dados na entrada além de ranges como vimos acima, durante a aula veremos detalhado como esse comando funciona.



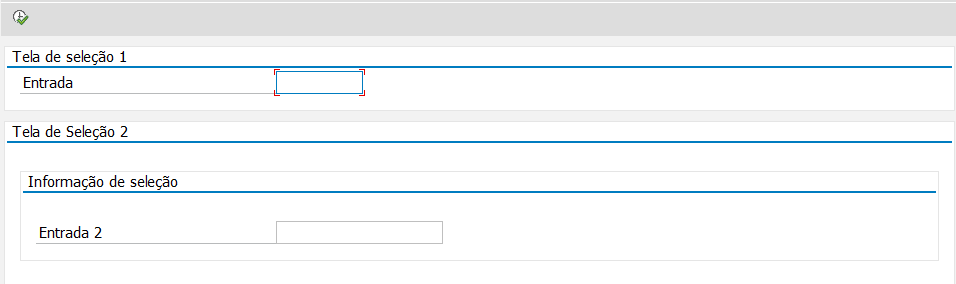
**TABLES**: MARA.

**SELECT-OPTIONS**: s\_matnr **FOR** mara-matnr.

### Telas de seleção

Para um usuário interagir com o programa, é necessário que sejam criados parâmetros de entrada, para que os dados possam ser digitados ou até mesmo selecionados para a execução parcial ou completa do programa.

Existem diversos tipos de tela de seleção, pois cada processo terá uma definição diferente e assim, a tela também será criada de acordo com cada necessidade.

Exemplo de telas de seleção: (Programa: **Z\_APASSARELLI\_004\_1**)

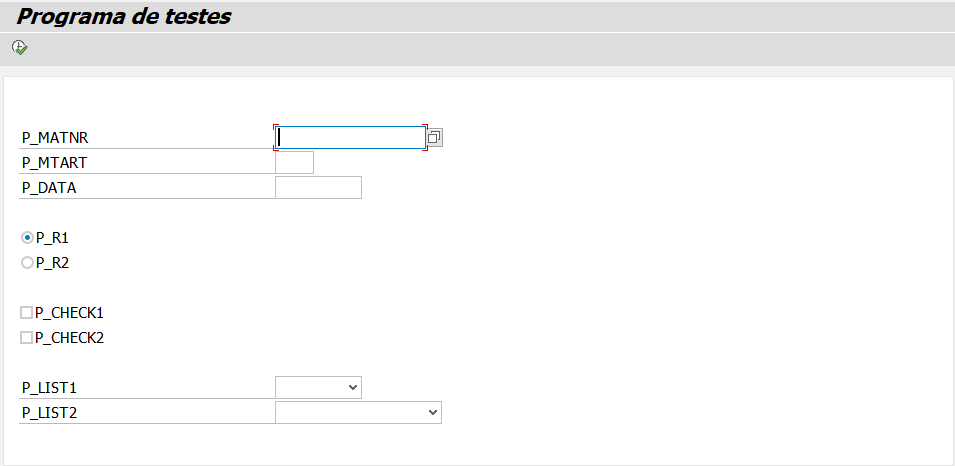
### o Opções da tela de seleção

Para tornar a tela de delação mais completa e personalizada de acordo com o solicitado, podemos alterar diversas opções no comando para a chamada de tela de seleção: “**SELECTION-SCREEN**”.

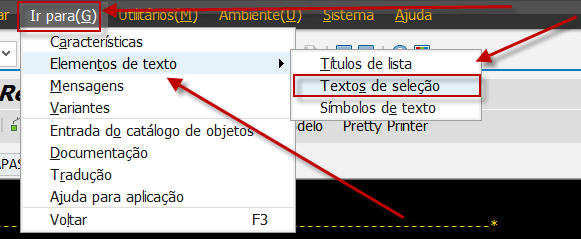
Dentro de uma tela de seleção, além de poder personalizar como a tela será exibida, como foi dito anteriormente a tela de seleção pode conter objetos de entrada para o usuário, e para isso usamos os dois mais comuns para uma tela de seleção:

PARAMETER SELECT-OPTIONS

Um PARAMETER é um parâmetro de entradas únicas, ou seja, é possível inserir apenas um registro de cada vez, se precisarmos de uma nova informação a anterior deverá ser apagada para que o novo valor seja aceito, exemplos de PARAMETERS:

Programa: **Z\_APASSARELLI\_004\_6**

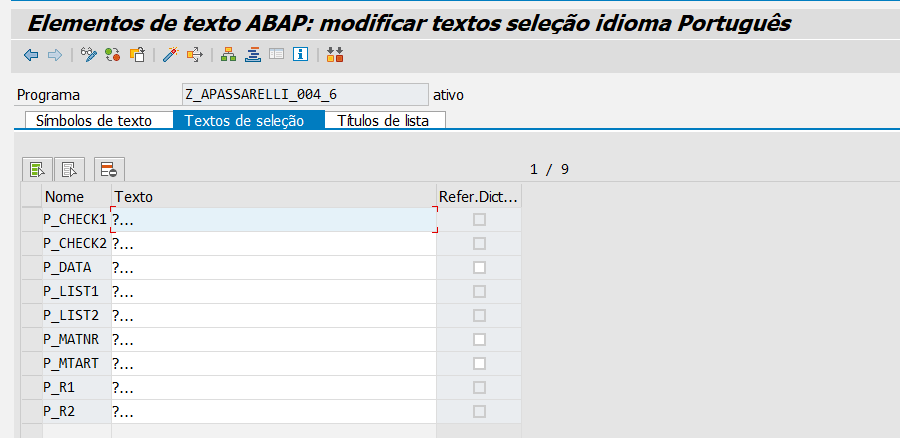
Acima vemos os parâmetros criados exibindo no programa seus nomes técnicos, pois ainda não definimos qual será o texto de exibição de cada um, para mudarmos o nome técnico para um texto personalizado, usamos o seguinte menu dentro da transação SE38:



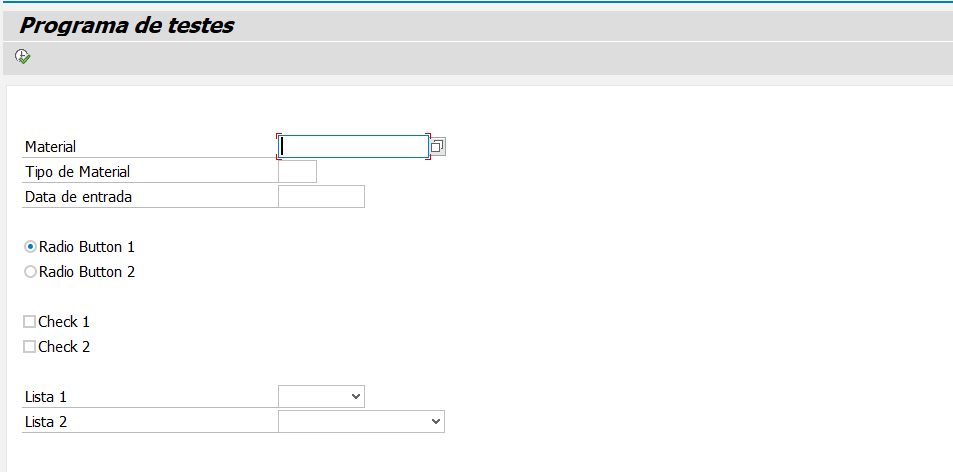
Podemos inserir o nome que desejarmos para cada um dos campos, bastando

substituir o “ ?... ” pelo nome que você deseja, mas lembre-se sempre de deixar

um nome bastante informativo, pois o usuário muitas vezes não tem qualquer conhecimento avançado sobre SAP, precisando que tudo seja muito informativo na hora que for descrito.

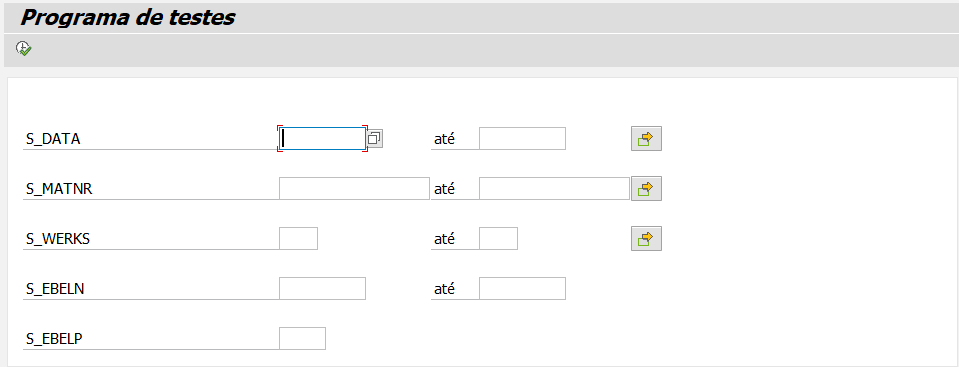


Após informar os nomes de cada campo e ativar, os textos serão exibidos no lugar das informações técnicas do campo.



Ao contrário de um PARAMETER o SELECT-OPTIONS pode aceitar mais de um valor de uma vez, além de permitir que sejam definidos ranges

(de/até) que podem auxiliar muito na hora de pesquisar uma informação, vejamos abaixo alguns exemplos de SELECT-OPTIONS em telas de seleção:

Programa: **Z\_APASSARELLI\_004\_7**

### Após mudarmos os elementos de texto:

