

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Sumario:

Este artigo tem como finalidade exemplificar o Maximo o entendimento e criação de uma tabela z. Este artigo é voltado a abap iniciantes e abap que algum tempo não faz a criação de tabelas.

Sobre o Autor:

Uderson Luis Fermino, formado em Ciências da Computação pela Faculdade de Pesquisa e Ensino IPEP, atua no mercado a 2 anos como desenvolvedor Java nas plataformas: (J2SE, J2EE e J2ME), com participação em grandes projetos envolvendo estas tecnologias. É consultor ABAP com experiências em REPORT, ALV (GRID, LIST, BLOCK, OO, TREE, HIERARQUICK), IDOC, ALE, ONLINE, SAPSCRIPT, SMARTFORM, NETWEAVER (JCO, BSP, WebDynpro).

Email:

Uderson@gmail.com

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Introdução

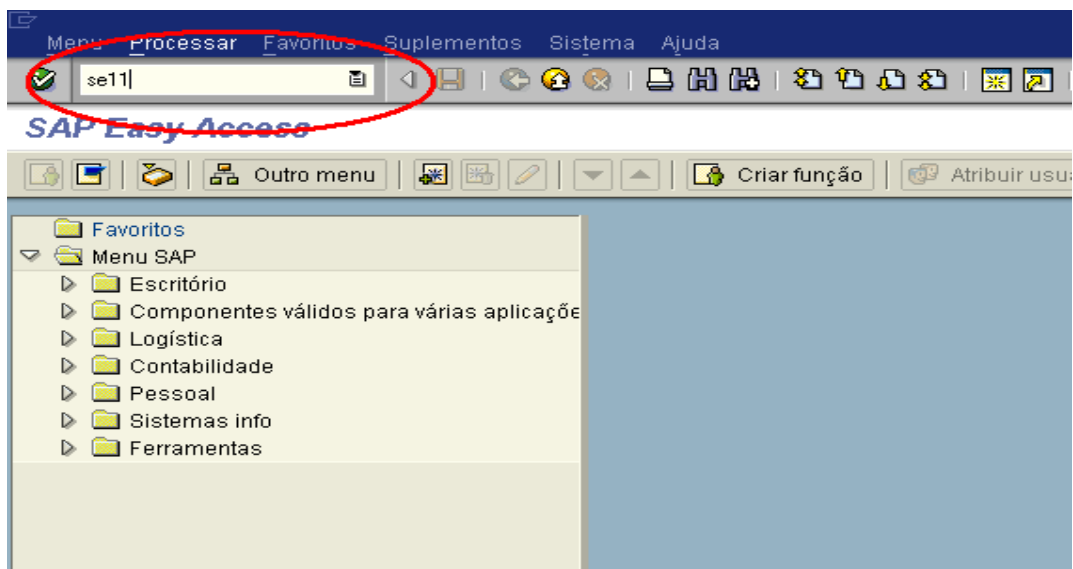
Uma tabela é uma estrutura do tipo matriz multidimensional, que pode armazenar valores relativos algum tipo de dado pré-definido. Em abap, uma tabela é uma estrutura de dado, que faz referencias a tipos de dado pré-construidos, estes tipos de dados são chamados de **elementos de dados**,

Exemplo de um **elemento de dado**:

Char20 = **ZNOME** onde ZNOME é um elemento de dados que faz referencia a um tipo de dado primitivo, char20, uma vez definido este elemento de dados, ele poderá ser usado em qualquer programa dentro do SAP. **Char20** é chamado dentro do SAP de Domínios de dado, pois é um tipo de dado, que faz referencia a um tipo de dados definido em uma estrutura de dado, CHAR, onde a tipo char é um tipo e de dado de capacidade de 1 BYTE, podendo armazenar dentro dele apenas um caractere, para que este tipo de dado possa armazenar uma palavra de 20 caracteres é necessário criar um vetor de caractere de 20 posições, em abap, para criar este vetor, é necessário utilizar transações. Está etapa dentro do SAP é chamada de definição de Domínios. Pode-se definir qualquer tipo de Domínio, para ser utilizado dentro de um elemento de dado, pois na construção de tabelas é necessário colocar elementos de dados que fazem referencias a domínios validos.

1. Criando uma tabela Z com Elementos de Dados e Domínios.

Primeiro passo é entrar na transação, SE11, através ta tela inicial.

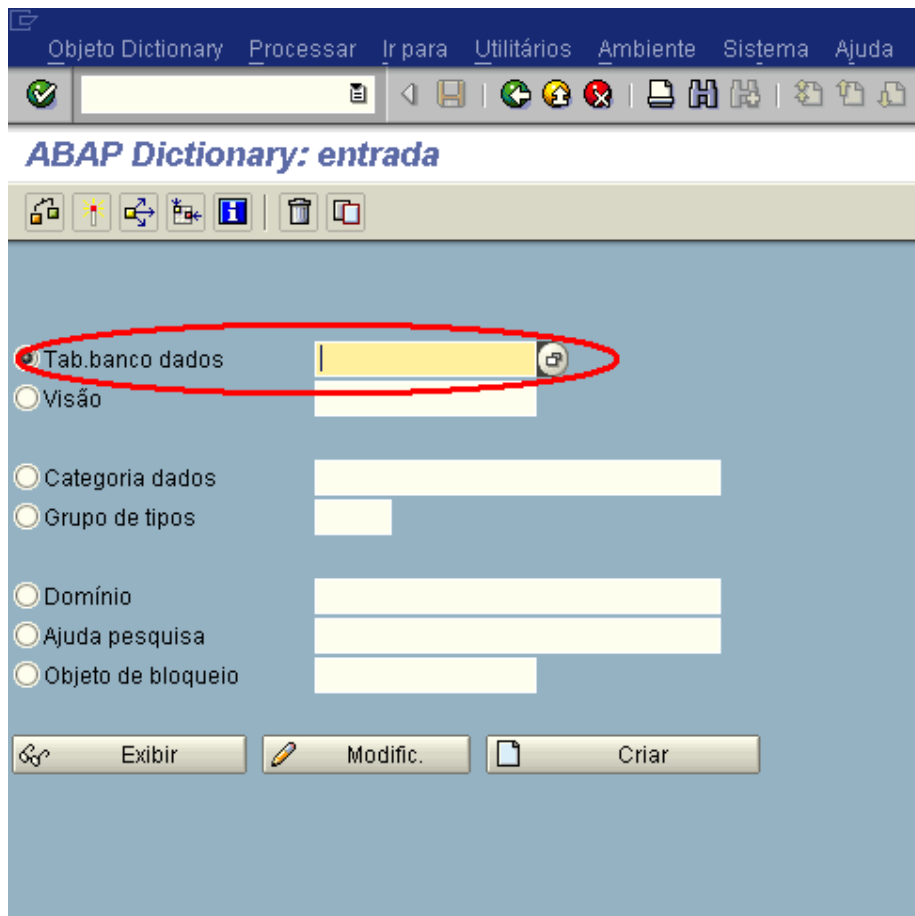


"tudo posso naquele que me fortalece" (Filipenses 4:13).

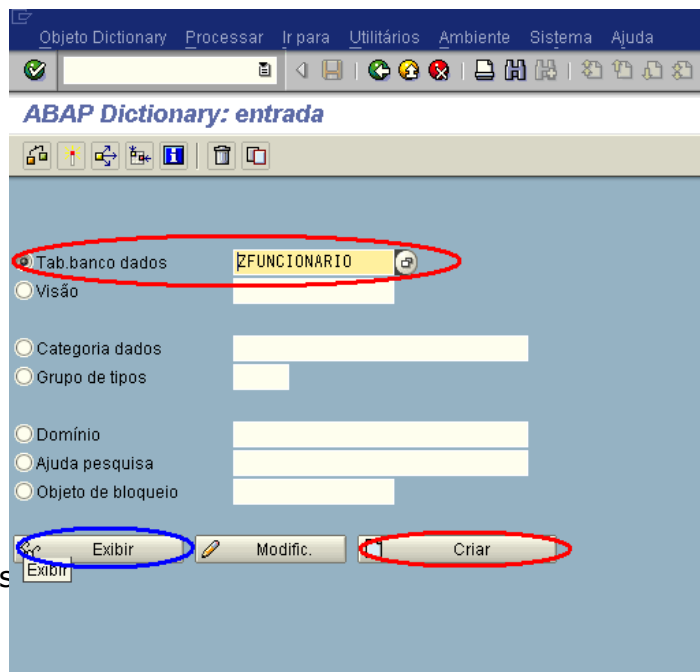
Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Este será a tela, relativa a transação SE11



No campo indicado coloque o nome da tabela existente, ou o nome a ser criado, neste caso, o nome que será colocado, será um nome, de uma tabela, que necessita, ser criada, colando o nome desta de ZFUNCIONARIO.

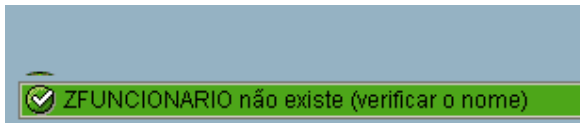


"tudo pos

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Clique em **exibir**, para verificar se existe esta tabela, no sistema, e verifique a barra de status.



Caso a mensagem seja que já exista, troque para outro nome, exemplo:

ZFUNCIONARIO1, caso a mensagem seja que não exista, e pede para verificar o nome, clique em **Criar**.

A seguinte tela deverá, aparecer:

Descrição breve: Descrição que será referência de sua tabela, coloque textos relativos as informações que a tabela irá armazenar.

Aba **Entrega e atualização:** Como os dados irão ser manipulados.

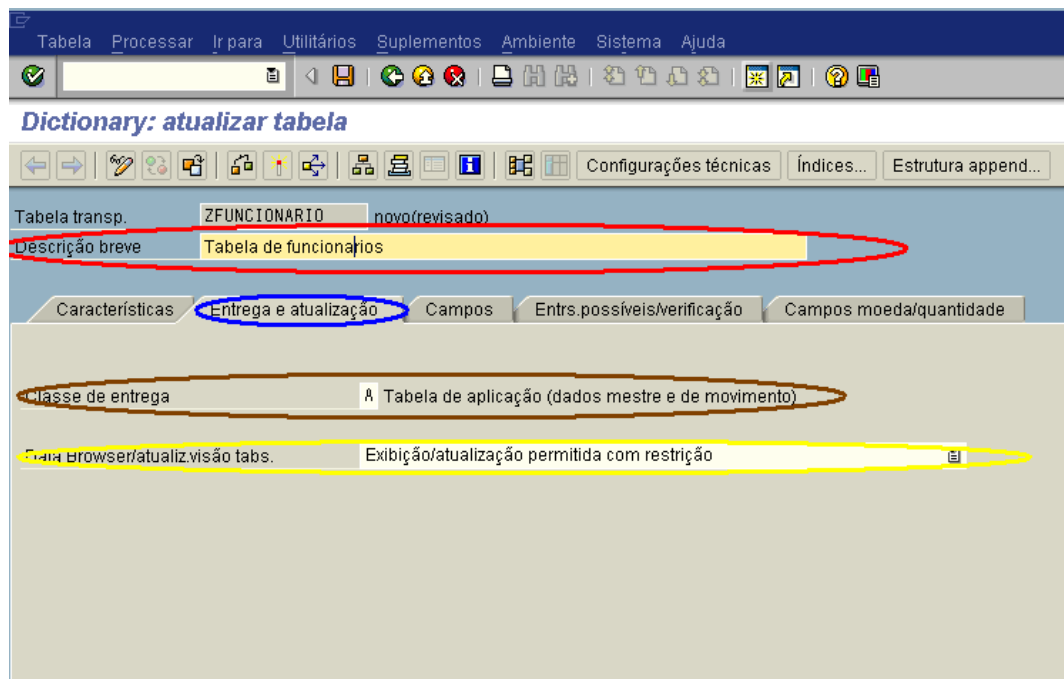
Classe de entrega: tipo de dados a se manipulados, geralmente é escolhido o tipo A.

Data browser/atualização visão da tabela: como as exi

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Após definir a manipulação dos dados da tabela, devemos criar os campos desta tabela, seguiremos a seguinte tabela:

Matricula	Nome	DataNascimento	RG	CPF

Descrição

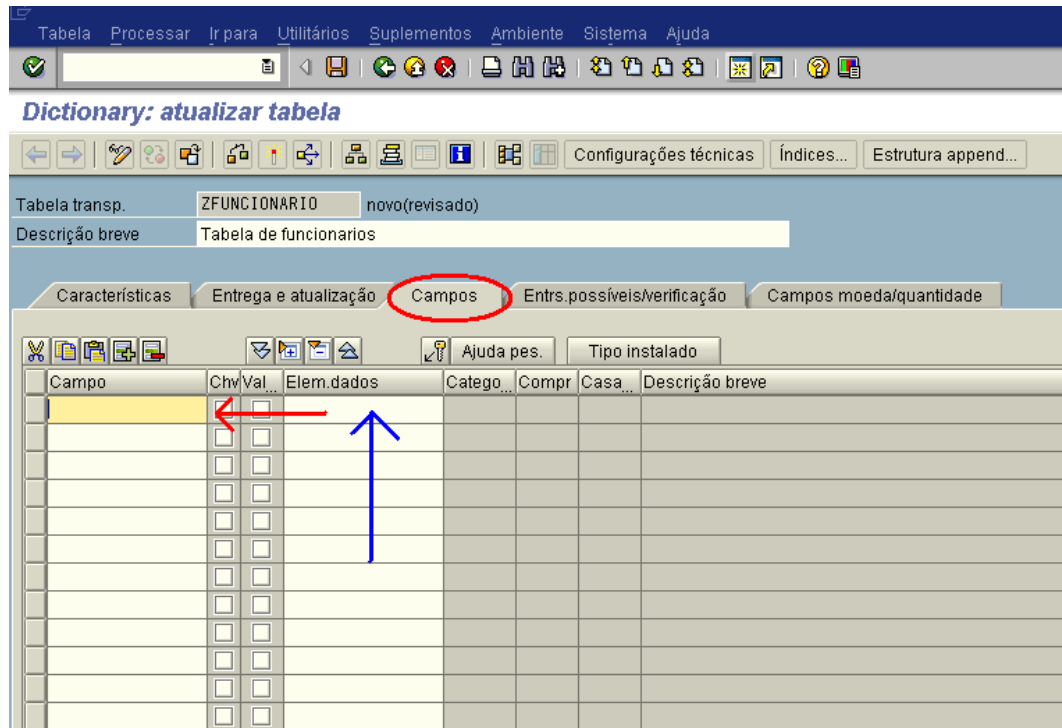
Campo	Tipo de dado	Elemento de Dado
Matricula	Int	ZMATF
Nome	Char(30)	ZNOMEF
DataNascimento	Date	ZDATNASF
RG	Char(15)	ZRGF
CPF	Char(15)	ZCPFF

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

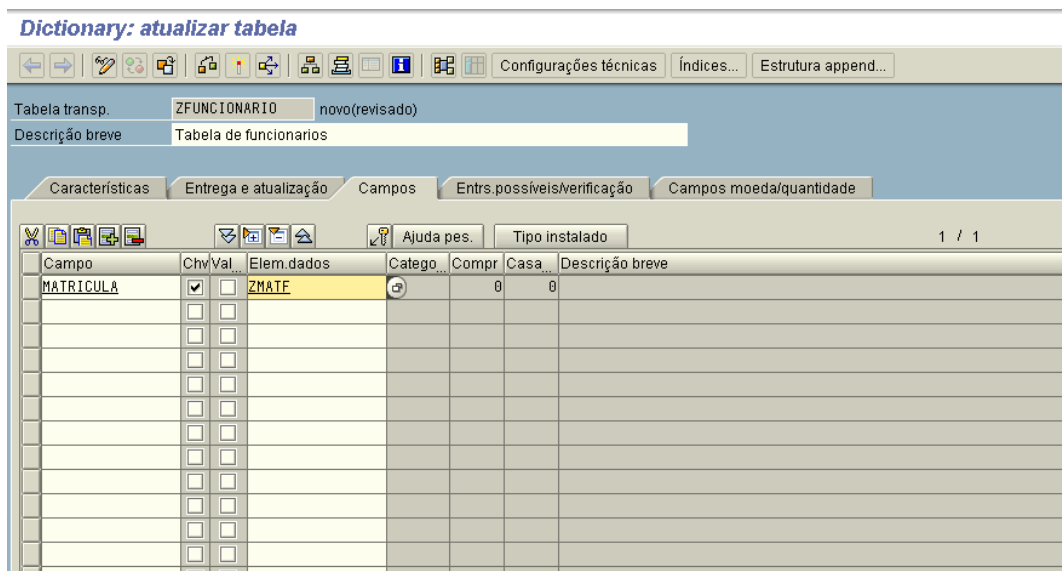
Clique na aba Campos:



Na seta **VERMELHA**, são os campos para inserção de **CAMPO** da tabela, na seta **AZUL**, será a inserção cós elementos de dados.

No SAP, não é possível colocar tipos de dados primitivos diretamente na tabela, é necessário criar Elementos de Dados que fazem referências para estes tipos, primeiramente, varemos a inserção do campo MATRICULA, e o elemento de dado ZMATF.

Observação: Matricula é o campo chave primaria, observe que ao lado do CAMPO a a Chave primaria, marque está opção com um clique.

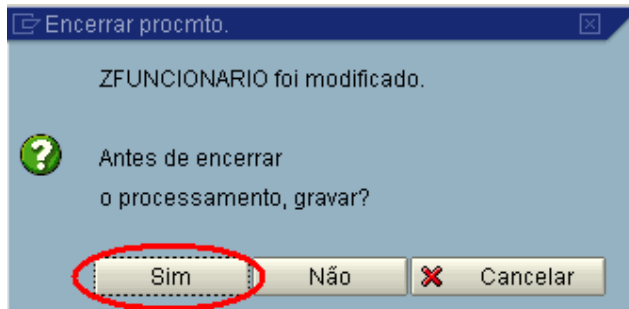


Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

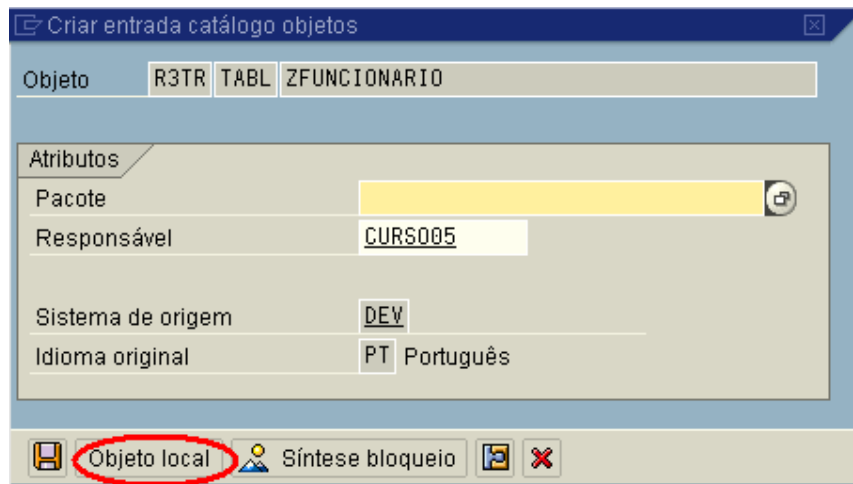
Após a inserção destes, de um enter, e veja que os campos Categoria, Comprimento, Casa e Descrição, estão em branco, isso quer dizer que o elemento de dado ZMATF, não existe, para que estes campos sejam preenchidos é necessário, criar este elemento de dados, para criar, basta dar “2 cliques em cima dele ZMATF”.

Ao clicar aparecerá a seguinte mensagem:

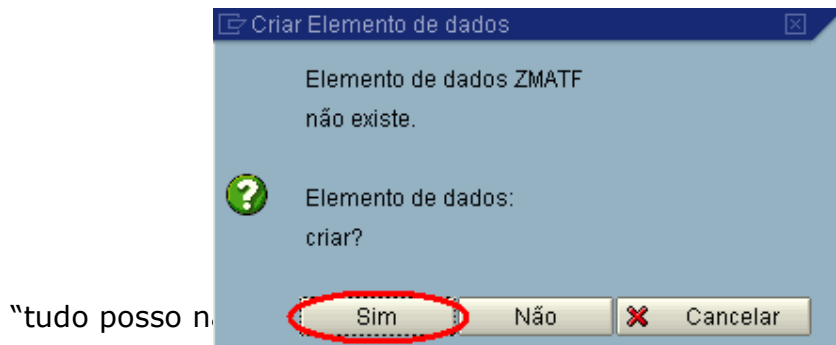


Clique em **sim**.

Após irá aparecer a seguinte tela, pedindo para especificar, em qual pacote o requeste deseja salvar este elemento de dados, escolha “Salvar Local”.



Novamente a mensagem dizendo que o elemento de dado, não existe, e se deseja criar. Novamente escolha **SIM**



Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

A seguinte tela, será apresentada:

Descrição breve: Descrição que será referencia de sue elemento de dado, coloque textos relativos as informações que o elemento de dado se refere.

Categoria de dados: No sap um tipo primitivo é vinculado ao um domínio, um domínio é um tipo de dado com determinadas características.

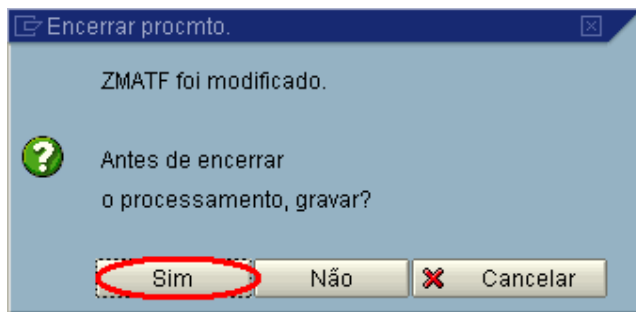
The screenshot shows the SAP Dictionary 'Atualizar elemento de dados' (Update data element) screen. The data element 'ZMATF' is selected, and the 'Descrição breve' (Short description) field is highlighted with a red oval. Below the description field, the 'Ctg.dds.' (Data Element Category) tab is selected, and the 'Domínio' (Domain) radio button is chosen. The 'Compr.' (Length) and 'Casas decimais' (Decimal places) fields are set to 0. The 'Tipo de referência' (Reference type) section is also visible, with the 'Referência a tipo instalado' (Reference to installed type) radio button selected.

Preencha a descrição relativo ao elemento de dado atual, no exemplo o elemento de dado ZMATF é o “Código da matrícula do Funcionário”, está pode ser uma boa descrição para este elemento de dados. No campos Categoria Elementar Domínio escolheremos o tipo de domínio que o elemento de dado terá, será usando um Domínio referente ao tipo primitivo NUMC, porem um numc puro tem 5 casa decimais e no nosso exemplo queremos um ZNUMC de 19 casa decimais, para isso devemos criar este domino de dado chamado de ZNUMC19. Após inserir zNUMC19 no campo, basta dar dois clique, e seremos levado a seguinte mensagem:

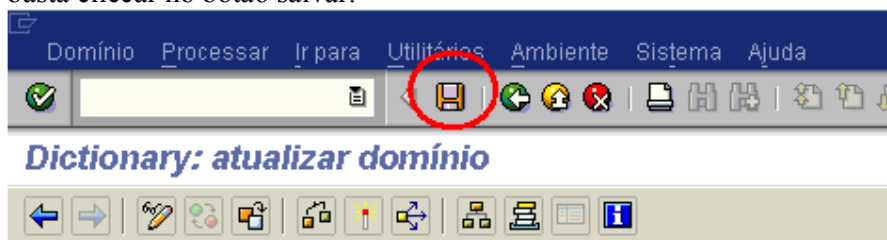
“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

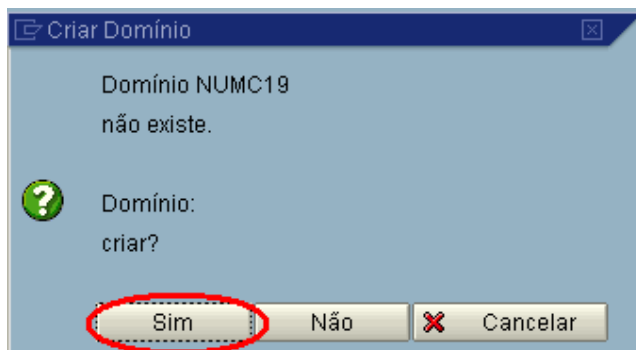
UDERSON LUIS



Clique em **SIM**, está mensagem é somente para salvar as ultimas alterações, para não ficar apresentando está mensagem sempre quando tiver que criar uma novo elemento basta cliecar no botão salvar:



Após salvar, será apresentada a mensagem dizendo se deseja cria este objeto realmente, escolha **SIM**



“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

A seguinte tela será apresentada:

Dictionary: atualizar domínio

Domínio: ZNUMC19 novo(revisado)

Descrição breve: Domínio numerico

Caracts. Definição Interv.vals.

Formato

Categoria dados: NUMC Cadeia de caracteres só com algarismos

Núm. posições: 19

Decimais: 0

Características da saída

Comprim. saída: 19

Rotina conversão:

☐ Sinal +/-

☐ Letr.minúsculas

Descrição breve: Descrição que será referencia de seu domínio, coloque textos relativos as informações que a tabela irá armazenar.

Categoria de Dados: tipo de dado primitivo do SAP.

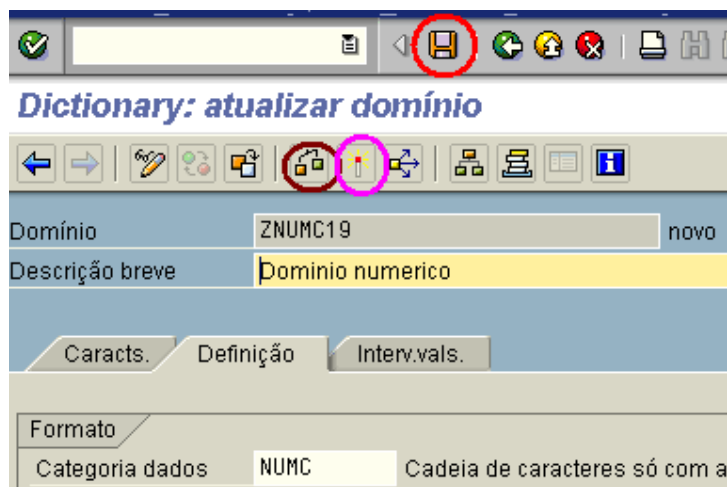
Características de Saídas: Tamanho do domínio e do tipo de dados que vocês está criando.

Após o preenchimento dos campos relativamente igual acima, será necessário seguir estas três retinas: SALVAR, CHECAR e ATIVAR. Para que um objeto seja enxergado em qualquer lugar dentro do SAP, é necessário Ativar este elemento, para ativado é necessário Salvar as alterações, e Checar se não existe erros. Após, verificar estás rotinas devemos Ativar o objeto. Para realizar estas rotinas segue o menu:

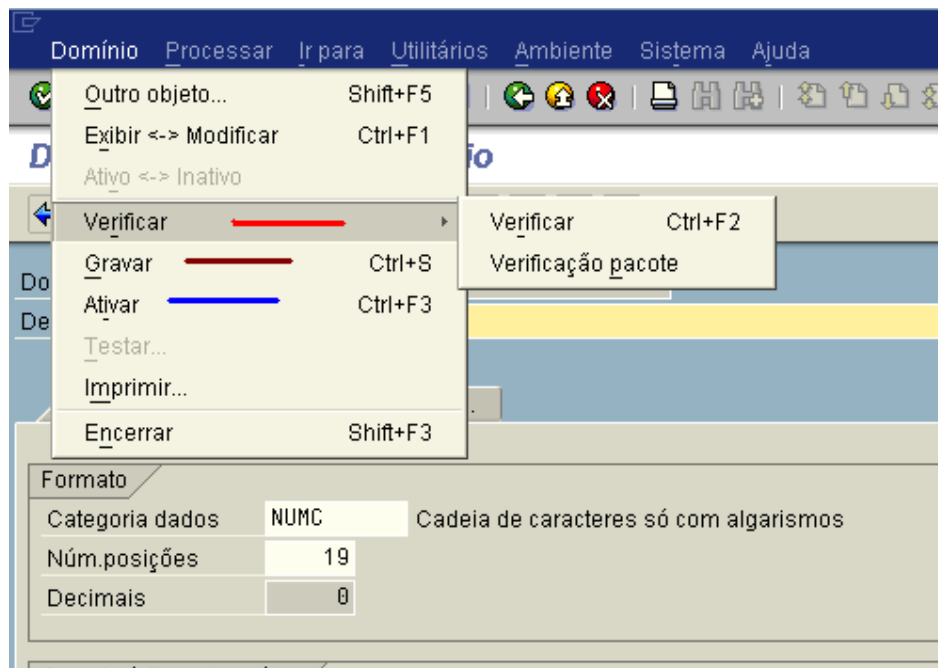
“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Ou

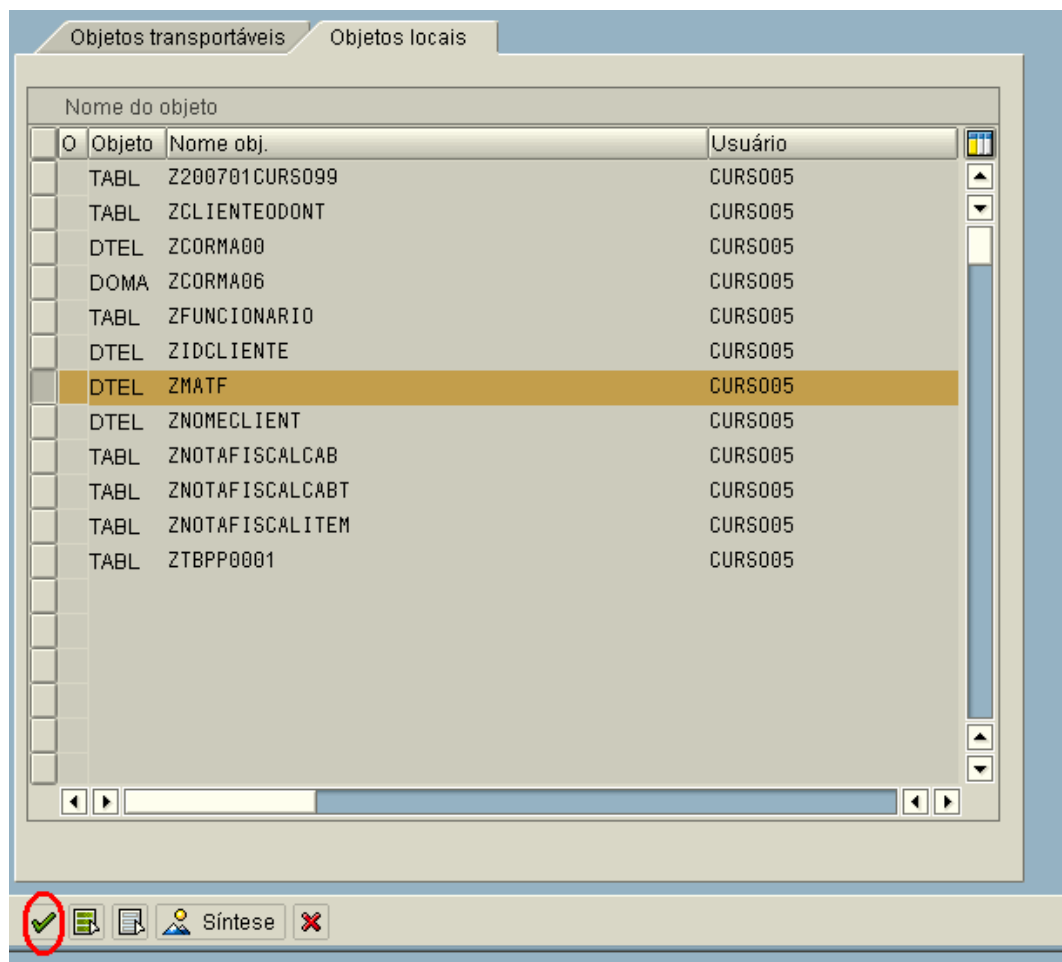


Ao ativar será mostrado a seguinte janela, que é apenas uma confirmação, basta clicar em ok e segui.

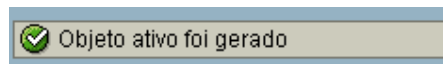
“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

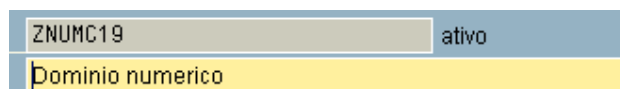
UDERSON LUIS



Sempre observe a barra de status. Após ativar a barra de status está assim:



E ao lado do objeto está o status.

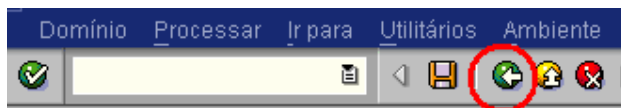


Após o objeto está ativo, basta voltar.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Dictionary: atualizar domínio

Observe estas mudanças:

A screenshot of a configuration window with tabs: 'Caracts.', 'Ctg.dds.', 'Características adicion.', and 'Denomin.campo'. The 'Domínio' tab is active. It shows a radio button selection for 'Categoria elementar' with 'Domínio' selected. The domain name is 'ZNUMC19' and the type is 'domínio numerico'. Below, there are fields for 'Ctg.dds.' (NUMC), 'Compr' (19), 'Cadeia de caracteres só com algaris' (empty), 'Casas decimais' (0), 'Tipo instalado' (empty), and 'Tipo de referência' (empty).

Ponto o Domínio já está criado.

Devemos continuar a criar o elemento de dado. Clique na aba **DENOMIN CAMPO**

A screenshot of the same configuration window, but with the 'Denomin.campo' tab selected and circled in red. The tab contains a table with columns 'Compr' and 'Denominador de campo'. The table has four rows: 'Breve', 'Médio', 'longa', and 'Título', each with an input field for 'Compr' and a longer input field for 'Denominador de campo'.

Nesta tela deverá ser colocado os textos do elemento de dado definidos, por tamanho, Onde BREVE, somente as iniciais da descrição, MÉDIO, duas ou três Silabas de cada para lavra da descrição, Logo, a descrição inteira, fncado para o exemplo:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Breve	Compr	Denominador de campo
Breve		MF
Médio		MT FUNC
longa		Macricula do Funcionario
Título		

Pronto, o elemento de dado já está pronto, basta seguir as rotinas descritas no Domínio.

Breve	Compr	Denominador de campo
Breve		MF
Médio		MT FUNC
longa		Macricula do Funcionario
Título		

Após seguir as rotinas e tudo estiver de acordo, Ativado, basta voltar.

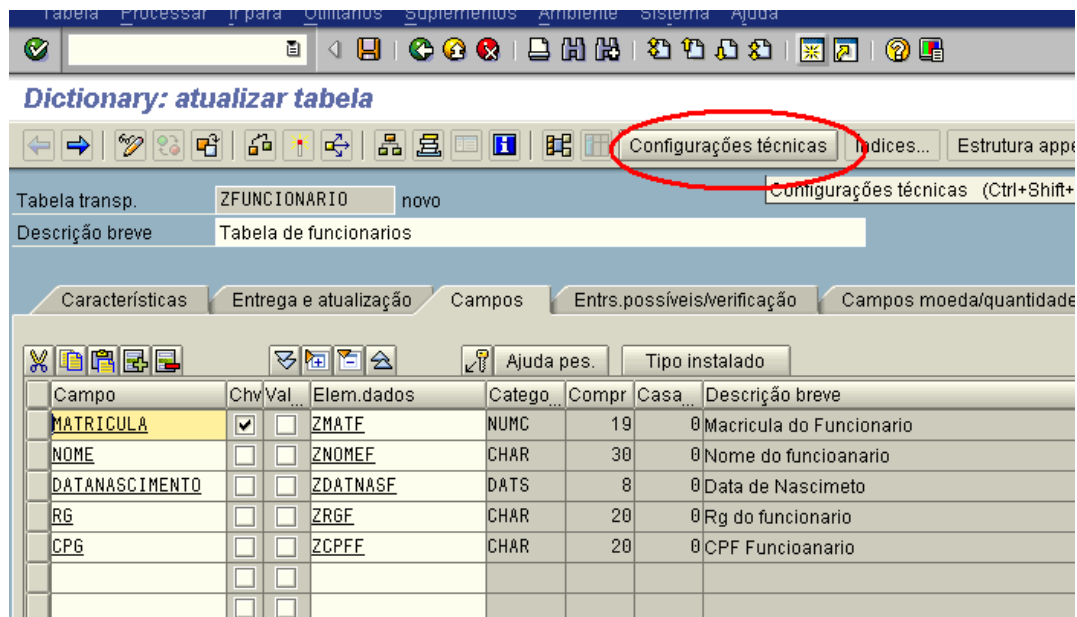
Breve	Compr	Denominador de campo
Breve		MF
Médio		MT FUNC
longa		Macricula do Funcionario
Título		

Siga os passo anteriores para criar os elementos de dados relativos a cada campo, de acordo a tabela que foi criada lá no inicio do texto, vale observar que pode acontecer que já exista elementos de dados dom os nomes que se irá usar, caso o domínios do mesmo seja equivalente ao que é requisitado não é necessário criar um novo basta usar este que já está criado, e vise-verso, para o domínio, geralmente domínios dos tipos:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Escolha os seguinte dados nesta tela, e depois salve e volta para anterior



Nome ZFUNCIONARIO Transparente Tabela

Descrição breve Tabela de funcionarios

Última modif. CURS005 21.06.2007

Status novo Gravado

Parâmetros de memória lógicos

Tipo de dados APPL0 Dados mestre, tabelas transparentes

Categor.tamanho 0 Registros dados esperados: 0 até 6.300

Armazenamento em buffer

☒ Ar.buffer não permitido

☐ Arm.buffer permitido, mas desativado

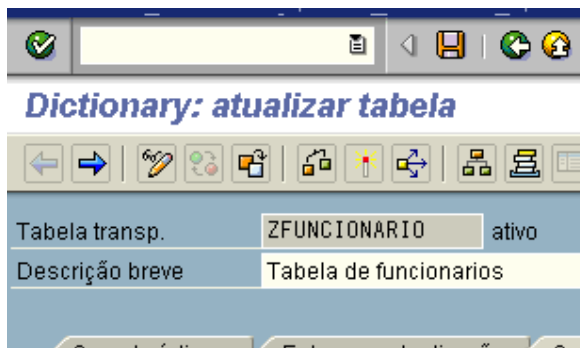
☐ Armaz.em buffer ativado

Tipo de armazenamento em buffer

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

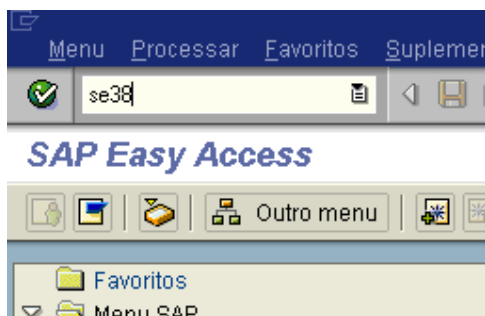


Para terminar sigar as rotinas de **salvar, checar e ativar...**

Pronto sua tabela está pronta para receberem dados.

Criando um programa para manipular está tabela.

Os programas dentro do SAP são chamados de **REPORTS**.
Para criar um REPORT é necessário utilizar a transação **SE38**.



Será apresentada a seguinte tela:

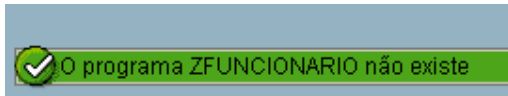
“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

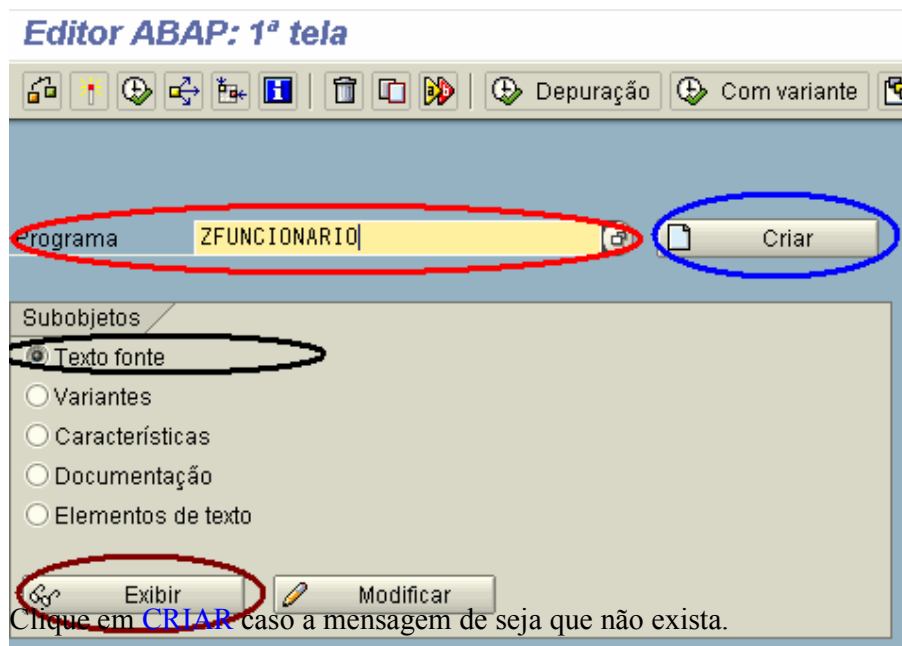
UDERSON LUIS

Programa: Nome que o programa terá, preencha com **ZFUNCIONARIO**, para seguir o exemplo, lembrando que todo programa que não seja STANDARD (programa já construídos pela SAP, devem começar com a letra Z).

Texto Fonte: Opção para criar, exibir ou modificar um programa escrito, esta opção será usada para a criação de exemplo FUNCIONARIO, antes de criar é necessário, verificar se este programa já existe dentro do SAP, para isso clique em **EXIBIR**.
Observe a para de status:



Caso a mensagem seja que já exista este programa, coloque outro nome, mais não se esqueça de marcar, pois usaremos, no exemplo o nome ZFUNCIONARIO.



A tela seguinte a parecerá:

Título: Título referente a programa, coloque texto relativo ao programa.

Tipo: Tipo de programa que será feito escolha o tipo "**Programa Executável**", pois será um programa que deverá ser executado.

Status: status do tipo do programa, no nosso exemplo "**Programa Teste**".

Aplicação: tipo da aplicação, modulo, lembrando que o SAP é um ERP, que tem diversos tipos de módulos, exemplos são:
Financeiros, Compras etc.

"tudo posso naquele que me fortalece" (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

ABAP: características do programa ZFUNCIONARIO modificar

Título: Cadastro de cliente

Idioma original: PT Português

Criado: 22.06.2007 CURS005

Última modific.:

Status: novo(revisado)

Atributos

Tipo: Programa executável

Status: Programa de teste

Aplicação: Aplicação desconhecida

Grupo autorizações:

Banco de dados lógico:

Versão tela seleção:

☐ Bloqueio do editor ☒ Aritmética em ponto fixo

☒ Verifs.unicode ativas ☐ Início via variante

Gravar

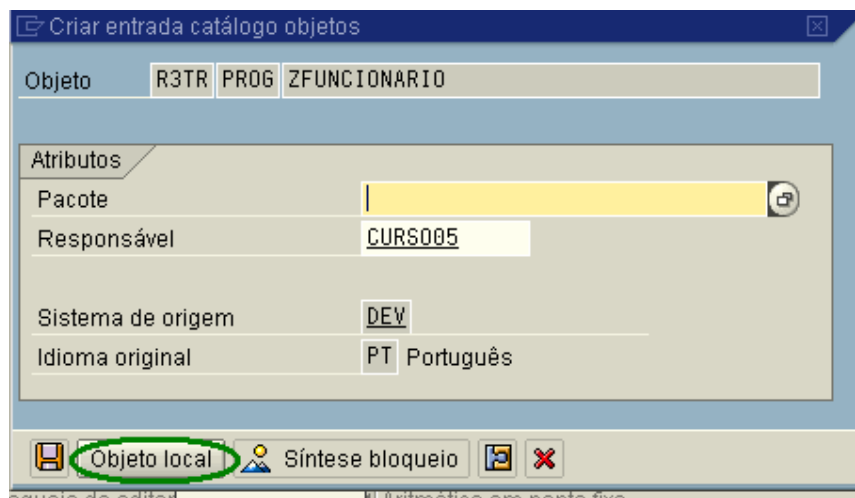
Após preenchido clique em **GRAVAR**.

Será pedido para gravar o program , clique em **OBJETO LOCAL**.

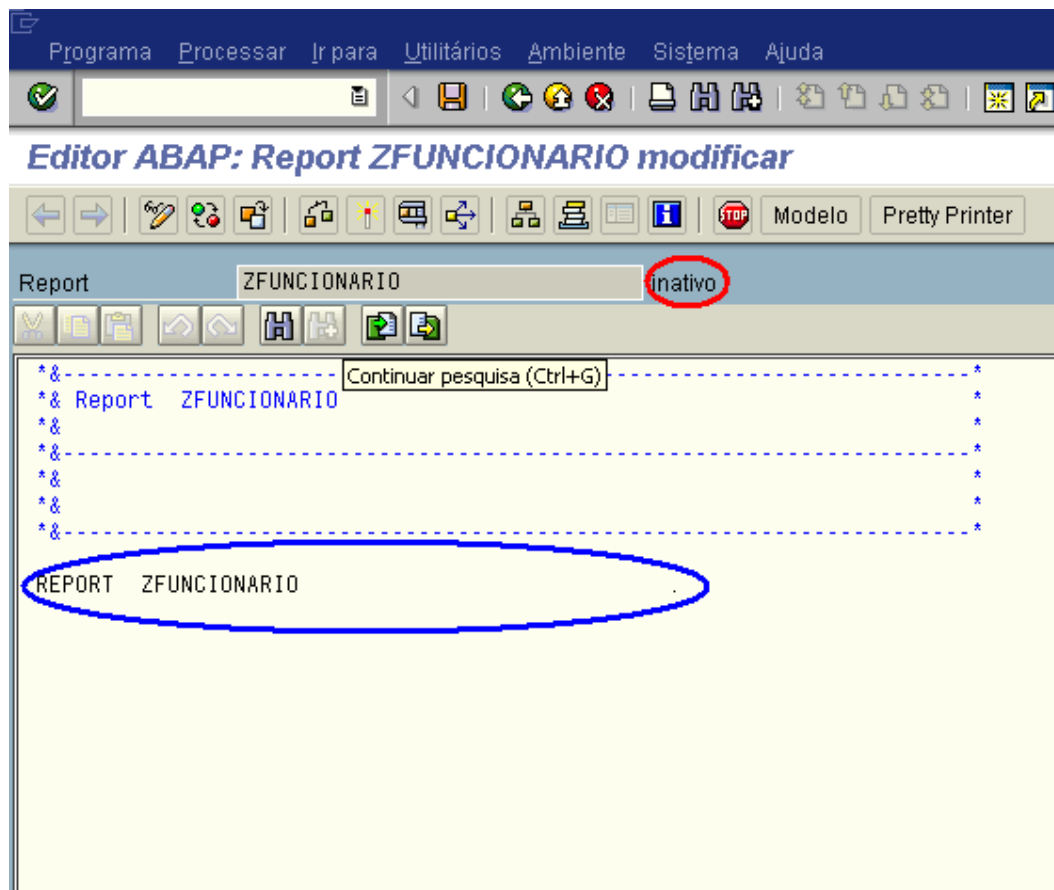
“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Após salvar, será aberto o editor de códigos ABAP.



Observe que o program está **INATIVO**, e que começa apenas com as Instrução de Início, **REPORT ZFUNCIONARIO**.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Embaixo de **REPORT ZFUNCIONARIO**, será onde codificaremos todas as instruções abap para o programa **ZFUNCIONARIO**, lembrando o abap usa o caracterer **(.)** **PONTO FINAL**, como instrução token, que diz que é o fim de uma instrução. Cada instrução deverá terminar com esté caracter.

Inicio do codigo.

Primeiro passo é usar o comando **TABLES**, que serve para indicar as tabelas Standard ou Z que existe no SAP, que iremos usar no programa.

Sisntaxe do comando:

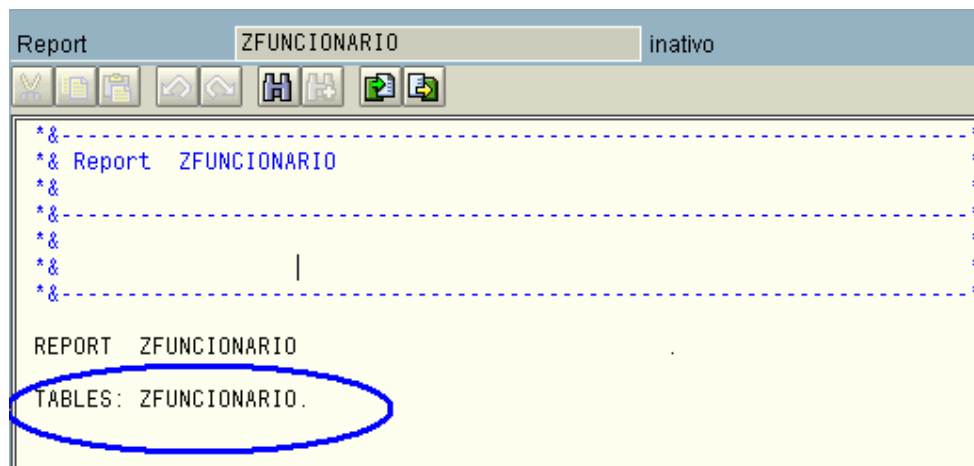
TABLE: nome_da_tabela .

ou

TABLE: nome_da_tabelaX, nome_da_tabelaY, ..., nome_da_tabelaN.

No codigo do nosso exemplo:

TABLE: ZFUNCIONARIO.



Não à problema que o nome da tabela seja igual ao do programa, pois as instruções são diferentes uma refere ao programa REPORT e a outra a TABLES.

2º Passo: criar uma interface para coletar os dados do usuario.

Criaremos um frame na tela, usando os comandos

```
SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.  
....Botões  
....Radios Buttons  
....Caixa de textos  
....etc.  
SELECTION-SCREEN END OF BLOCK b1001.
```

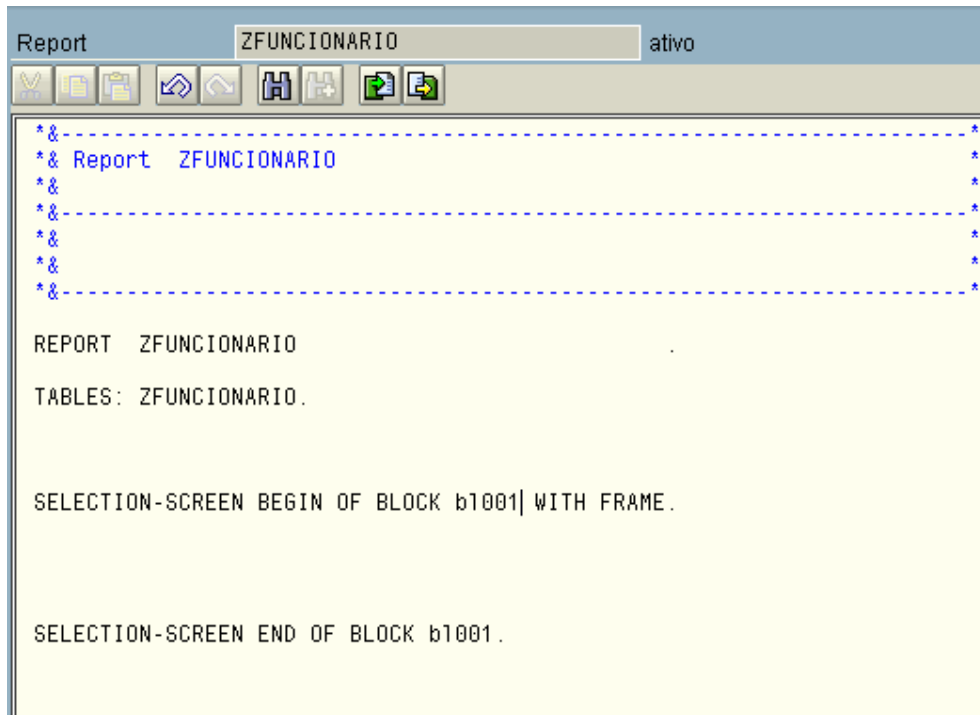
“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

O comando SELECTION-SCREEN BEGIN OF BLOCK bl001 WITH FRAME, está abrindo um bloco com o nome bl001, este bloco terá um frame.

O comando SELECTION-SCREEN END OF BLOCK bl001, está fechando o frame. É dentro do bloco que deverá colocar os “botões, caixas de textos, radio butons”.



Criaremos as caixas de textos, para coletar as informações, o comando PARAMETER, é o comando para criar caixas de textos em abap.

Sintaxe

PARAMETER: nome_do_objeto_caixa_de_texto **TYPE** tipo_de_dado.

PARAMETER: nome_do_objeto_caixa_de_texto **LIKE** elemento_de_dado.

Para declarar a caixa de texto, devemos declara o tipo de dado que está receberá e o tamanho, exemplo, para declara uma caixa de texto que recebe tipos de dados do tipo char de 20 é necessário usar a sintaxe:

PARAMETER: p_char(20) **TYPE** c.

Em abap é recomendado usar caixas de textos com elementos de dados, referentes aos campos da tabela que irá coletar ou inserir dados, para que os mesmo não venham dar erro de incompatibilidade de dados.

A sintaxe para o nosso exemplo, usando os tipos de dados já construídos para a tabela ZFUNCIONARIO ficará assim:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
PARAMETER: P_NOME LIKE ZFUNCIONARIO-NOME OBLIGATORY.
PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
PARAMETER: P_RG LIKE ZFUNCIONARIO-RG OBLIGATORY.
PARAMETER: P_CPF LIKE ZFUNCIONARIO-CPF OBLIGATORY.

Observe que os objetos **P_Alguna_Coisa**, fazem referencia ao nome dos campos da tabela e não aos elementos de dados, isso é porque os nomes dos campos já fazem referencias aos elementos de dados, não sendo necessário fazer ligação direta ao elementos de dados.

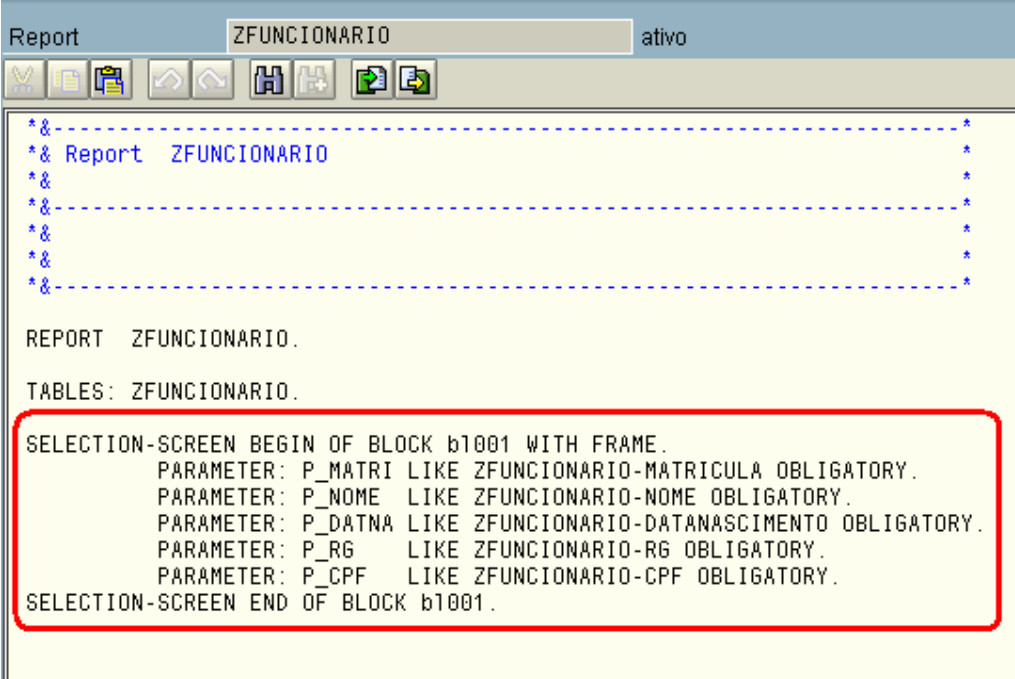
P_MATRI LIKE ZFUNCIONARIO-MATRICULA é equivalente:

P_MATRI = ZFUNCIONARIO-MATRICULA = ZMATF

O comando **OBLIGATORY** server para deixar o campo obrigatório, onde será necessário preencher o campo de qualquer forma.

É um bom padrão usar o **P_** para facilitar a legibilidade do componente, facilitando identificar no decorrer do programa qual componente é.

No código ficará:



```
Report      ZFUNCIONARIO      ativo

*&-----*
*& Report  ZFUNCIONARIO      *
*&-----*
*&
*&
*&
*&-----*

REPORT  ZFUNCIONARIO.

TABLES: ZFUNCIONARIO.

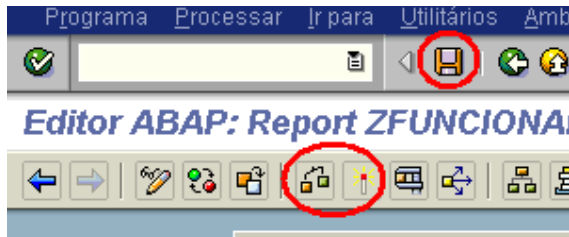
SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
    PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
    PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
    PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
    PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
    PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.
```

“tudo posso naquele que me fortalece” (Filipenses 4:13).

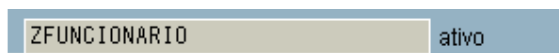
Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

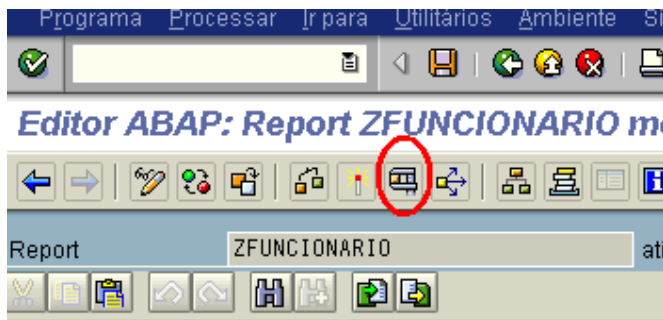
Para ver a telinha, **salvar, check e ativo**.



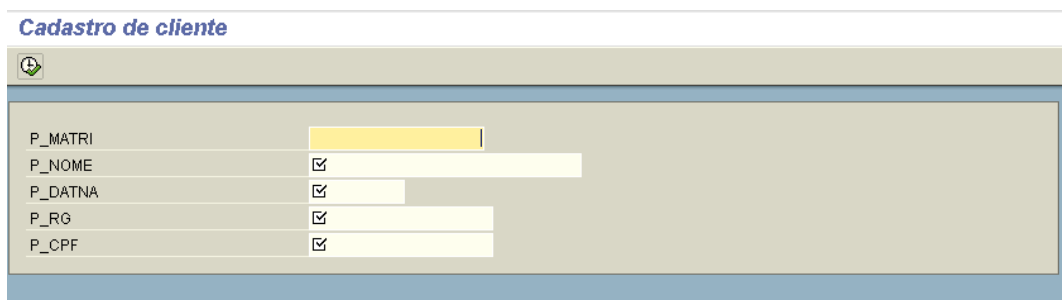
E veja se está tudo ok, status estar ativo.



Para ver a telinha clique em **Direto** ou tecle **F8**:



A tela é está:



Cadastro de cliente	
P_MATRI	<input type="text"/>
P_NOME	<input checked="" type="checkbox"/> <input type="text"/>
P_DATNA	<input checked="" type="checkbox"/> <input type="text"/>
P_RG	<input checked="" type="checkbox"/> <input type="text"/>
P_CPF	<input checked="" type="checkbox"/> <input type="text"/>

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Conhecendo o ambiente:

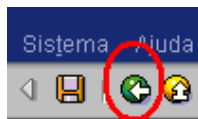
Cadastro de Cliente: Título que foi inserido na criação do programa

P_Alguna_Coisa: Títulos dos componentes **PARAMETER**.

Caixas_de_Inserção: Componentes gerado através do comando **PARAMETER**.

The screenshot shows a software window titled "Cadastro de cliente" with a green checkmark icon in the top-left corner. The window is divided into two main sections. On the left, there is a list of parameters: P_MATRI, P_NOME, P_DATNA, P_RG, and P_CPF, each enclosed in a blue rectangular box. On the right, there is a form area enclosed in a red rectangular box. This form contains a yellow rectangular input field at the top, followed by four rows, each consisting of a checked checkbox and a white rectangular input field.

Observer que **P_Alguna_Coisa**, estão com títulos difícil entender o que se refere a **Caixas_de_Inserção**. Então iremos criar títulos relativos as **Caixas_de_Inserção**. Volte ao código.



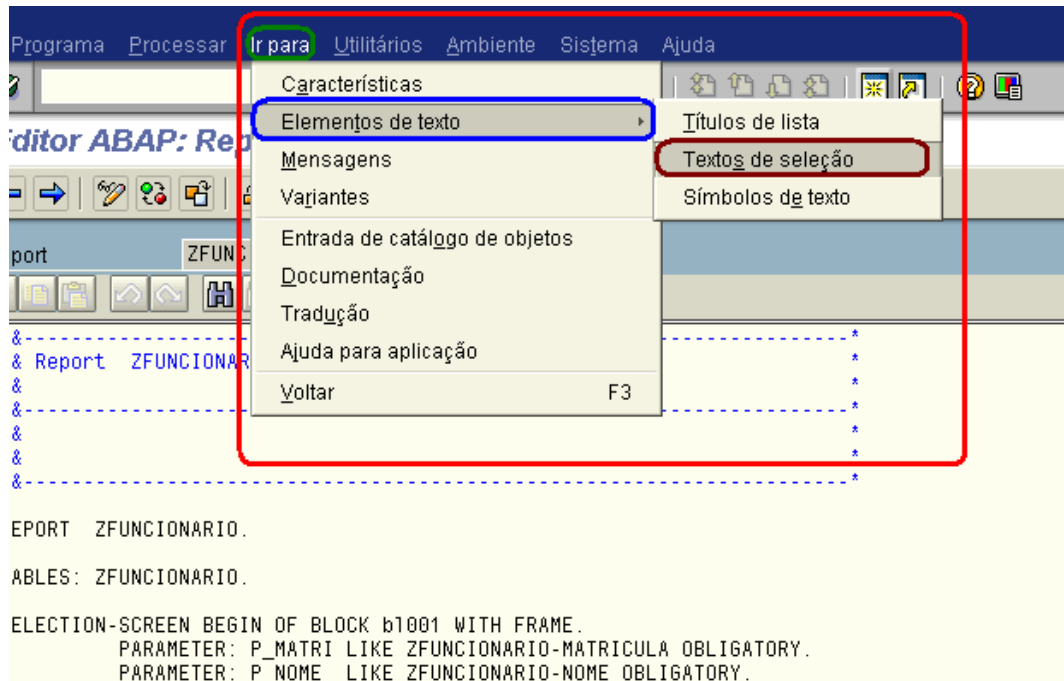
"tudo posso naquele que me fortalece" (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Siga o menu:

Ir para → Elemento de Texto → Texto de Seleção

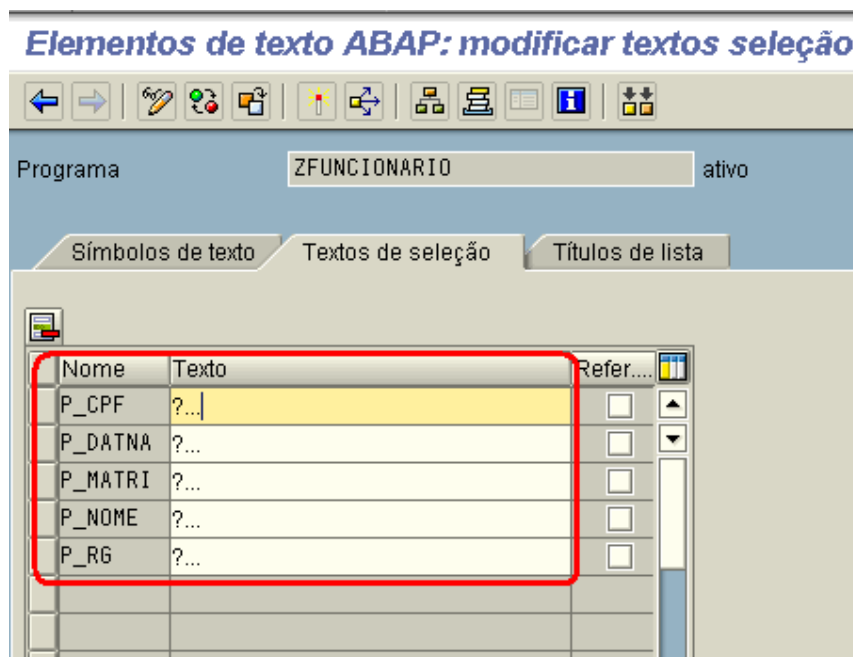


Será apresentado a seguinte tela:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Observe que os P_Alguema_Coisa estão com um ponto de interrogação, por isso que apareceu o próprio nome do componete na tela de apresentação.

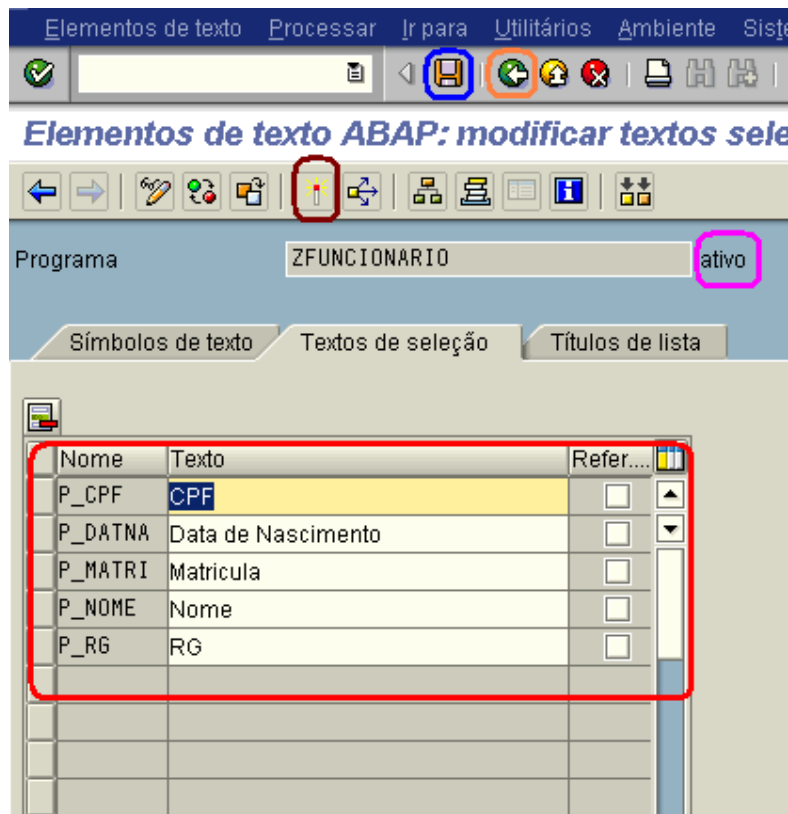
Troque os “?...”, pelo texto que deverá aparecer.

No exemplo será trocados por

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



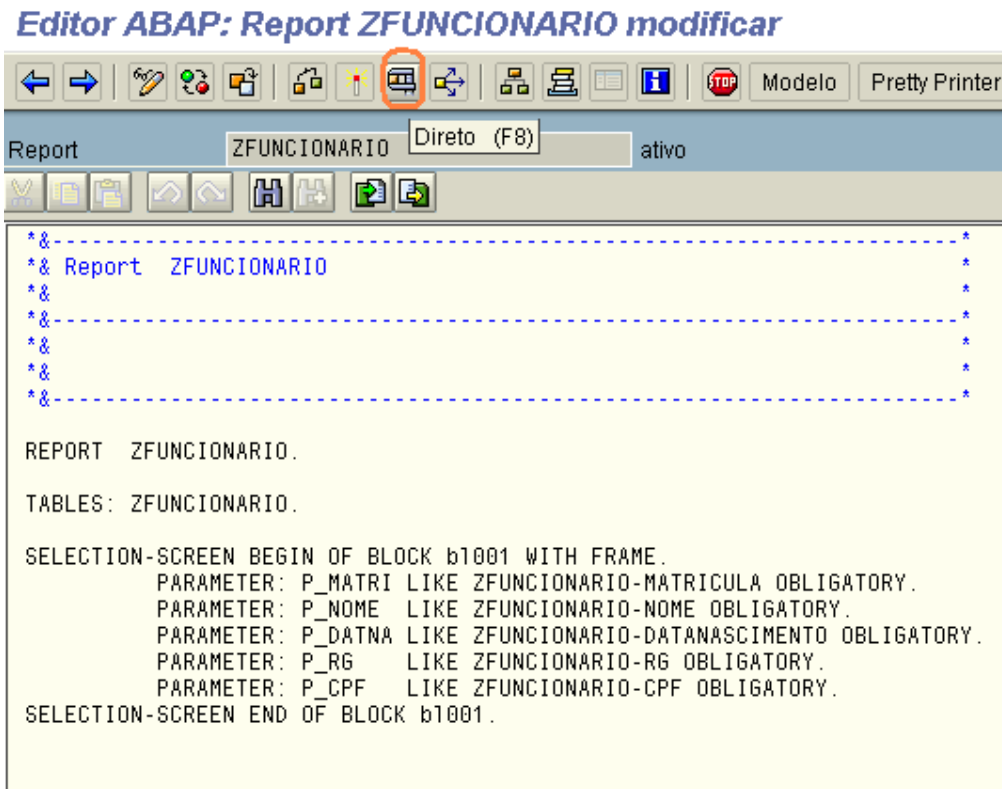
Após trocar siga os passos:

1. SALVAR
2. ATIVAR
3. VERIFICAR STATUS
4. VOLTAR

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Após seguir os passos, e voltar a tela de código, clique **Direto**, o tecla **F8** para ver as mudanças.

Veja:

The screenshot shows a SAP screen titled "Cadastro de cliente". It features a table with five rows for data entry. The first row, "Matricula", has a yellow input field. The subsequent rows, "Nome", "Data de Nascimento", "RG", and "CPF", each have a checkbox in the first column and a yellow input field in the second column. All checkboxes are checked. A blue rectangle highlights the input fields, and a red rectangle highlights the entire table area.

Cadastro de cliente	
Matricula	<input type="text"/>
Nome	<input checked="" type="checkbox"/> <input type="text"/>
Data de Nascimento	<input checked="" type="checkbox"/> <input type="text"/>
RG	<input checked="" type="checkbox"/> <input type="text"/>
CPF	<input checked="" type="checkbox"/> <input type="text"/>

O que fizemos a te agora foi a tela de, para coletar as informações agora, iremos criar a lógica do programa.
Volte ao código.

"tudo posso naquele que me fortalece" (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Em baixo da declaração de tabela, será criada uma tabela interna que terá a mesma estrutura da tabela ZFUNCIONARIO, uma tabela interna serve para auxiliar no desenvolvimento, inserção e remoção de dados sem precisar trabalhar diretamente na tabela real, a tabela interna é uma copia dos da estrutura da tabela real, pôr ela é armazenada em memória em tempo de execução, a grande vantagem de não trabalhar diretamente na tabela, é que o servidor onde os dados estão não fica sobrecarregado. Para declara uma tabela interna a sintaxe pode-se encontrar diversa sintaxe, usaremos a seguinte sintaxe:

```
DATA: BEGIN OF t_nome_da_tabela OCCURS 0,  
      campo referencia(TYPE/LIKE) TIPO_PRIMITIVO/TABELA-CAMPO,  
      .  
      .  
      END OF t_nome_da_tabela.
```

A tabela interna que usaremos ficará com a seguinte estrutura:

```
DATA: BEGIN OF T_ZFUNCIONARIO OCCURS 0,  
      MATRICULÁ      LIKE ZFUNCIONARIO-MATRICULA,  
      NOME            LIKE ZFUNCIONARIO-NOME,  
      DATANASCIMENTO LIKE ZFUNCIONARIO-DATANASCIMENTO,  
      RG              LIKE ZFUNCIONARIO-RG,  
      CPF             LIKE ZFUNCIONARIO-CPF,  
      END OF T_ZFUNCIONARIO.
```

O nome da tabela interna é T_ZFUNCIONARIO, que faz referencias as campos da tabela ZFUNCIONARIO, está duas tabelas agora são idêntica, o que defere as duas, é que T_ZFUNCIONARIO é usada em tempo de execução e armazenada em memória.

No código deverá ficar assim:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

```
REPORT ZFUNCIONARIO.

TABLES: ZFUNCIONARIO.

DATA: BEGIN OF T_ZFUNCIONARIO OCCURS 0,
      MATRICULA      LIKE ZFUNCIONARIO-MATRICULA,
      NOME           LIKE ZFUNCIONARIO-NOME,
      DATANASCIMENTO LIKE ZFUNCIONARIO-DATANASCIMENTO,
      RG             LIKE ZFUNCIONARIO-RG,
      CPF            LIKE ZFUNCIONARIO-CPF,
END OF T_ZFUNCIONARIO.

SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
  PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
  PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
  PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
  PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
  PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.
```

Agora que a tabela interna já está declarada, a interface, pronta para receber dados, iremos, criar a lógica de transferir os dados inseridos nas CAIXA_DE_INSERTÃO, para a tabela interna, lembrando que os campos devem ser respeitados de acordo cada tipo. O código deverá ser inserido abaixo do comando: SELECTION-SCREEN END OF BLOCK b1001, ficando assim:

```
T_ZFUNCIONARIO-MATRICULA      = P_MATRI.
T_ZFUNCIONARIO-NOME          = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG            = P_RG .
T_ZFUNCIONARIO-CPF           = P_CPF.
```

Onde estamos colocando dentro de cada campo da tabela interna, os campos relativos pos cada PARAMETER (CAIXA_DE_INSERTÃO) corresponde aos campos da tabela internas respeitando os tipos de dados.

No Código deverá estar desta forma:

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

```
REPORT ZFUNCIONARIO.

TABLES: ZFUNCIONARIO.

DATA: BEGIN OF T_ZFUNCIONARIO OCCURS 0,
      MATRICULA      LIKE ZFUNCIONARIO-MATRICULA,
      NOME           LIKE ZFUNCIONARIO-NOME,
      DATANASCIMENTO LIKE ZFUNCIONARIO-DATANASCIMENTO,
      RG             LIKE ZFUNCIONARIO-RG,
      CPF            LIKE ZFUNCIONARIO-CPF,
END OF T_ZFUNCIONARIO.

SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
  PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
  PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
  PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
  PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
  PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.

T_ZFUNCIONARIO-MATRICULA = P_MATRI.
T_ZFUNCIONARIO-NOME      = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG        = P_RG.
T_ZFUNCIONARIO-CPF       = P_CPF.
```

Para que os dados sejam inseridos dentro da tabela é necessário usar o comando APPEND.

Sua sintaxe é;

APEEND nome_da_tabela.

O código deverá estar assim:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

```
REPORT ZFUNCIONARIO.

TABLES: ZFUNCIONARIO.

DATA: BEGIN OF T_ZFUNCIONARIO OCCURS 0,
      MATRICULA      LIKE ZFUNCIONARIO-MATRICULA,
      NOME           LIKE ZFUNCIONARIO-NOME,
      DATANASCIMENTO LIKE ZFUNCIONARIO-DATANASCIMENTO,
      RG            LIKE ZFUNCIONARIO-RG,
      CPF           LIKE ZFUNCIONARIO-CPF,
END OF T_ZFUNCIONARIO.

SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
  PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
  PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
  PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
  PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
  PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.

T_ZFUNCIONARIO-MATRICULA = P_MATRI.
T_ZFUNCIONARIO-NOME      = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG        = P_RG .
T_ZFUNCIONARIO-CPF       = P_CPF.
|
APPEND T_ZFUNCIONARIO.
```

A tabela interna já está preenchida, agora iremos passa os dados da tabela interna para a tabela real, usaremos o comando MOVE, para mover os dados da tabela interna, para a tabela real.

Sintaxe do comando MOVE:

MOVE valor1 **TO** valor2.

Onde valor1 e valor2 podem ser trocados por variáveis ou campos de tabelas.

No exemplo o comando ficará:

```
T_ZFUNCIONARIO-MATRICULA = P_MATRI.
T_ZFUNCIONARIO-NOME      = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG        = P_RG .
T_ZFUNCIONARIO-CPF       = P_CPF.
APPEND T_ZFUNCIONARIO.

MOVE: T_ZFUNCIONARIO-MATRICULA TO ZFUNCIONARIO-MATRICULA,
      T_ZFUNCIONARIO-NOME      TO ZFUNCIONARIO-NOME,
      T_ZFUNCIONARIO-DATANASCIMENTO TO ZFUNCIONARIO-DATANASCIMENTO,
      T_ZFUNCIONARIO-RG        TO ZFUNCIONARIO-RG,
      T_ZFUNCIONARIO-CPF       TO ZFUNCIONARIO-CPF.
```

Após mover os dados para a tabela real, usaremos o comando INSERT, para inserir realmente dentro da tabela.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Observação::: sempre antes do APPEND ou do INSERT, o que fazemos é inserir nas HEADER LINE da tabela, quando usamos estes comandos realmente tiramos os valores que estão nas HEADER LINE e inserimos nas tabelas.

A sintaxe do comando INSERT é:

INSERT nome_da_tabela_real

O código deverá ficar:

```
SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
  PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
  PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
  PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
  PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
  PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.

T_ZFUNCIONARIO-MATRICULA = P_MATRI.
T_ZFUNCIONARIO-NOME      = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG        = P_RG .
T_ZFUNCIONARIO-CPF       = P_CPF.
APPEND T_ZFUNCIONARIO.

MOVE: T_ZFUNCIONARIO-MATRICULA TO ZFUNCIONARIO-MATRICULA,
      T_ZFUNCIONARIO-NOME      TO ZFUNCIONARIO-NOME,
      T_ZFUNCIONARIO-DATANASCIMENTO TO ZFUNCIONARIO-DATANASCIMENTO,
      T_ZFUNCIONARIO-RG        TO ZFUNCIONARIO-RG,
      T_ZFUNCIONARIO-CPF       TO ZFUNCIONARIO-CPF.

INSERT ZFUNCIONARIO.
```

Após usar o comando **INSERT** é necessário dar um **COMIT**, para que os dados sejam transportados, para a tabela real que está no servidor.

A sintaxe do comando **COMIT** é:

COMMIT WORK.

Antes de dar um **COMIT** é necessário verificar se o comando **INSERT** teve sucesso, a variável, que guarda códigos de retornos, de comando sucedidos ou não é a variável **SY-SUBRC**

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

O código deverá ficar:

```
RG          LIKE ZFUNCIONARIO-RG,
CPF         LIKE ZFUNCIONARIO-CPF,
END OF T_ZFUNCIONARIO.

SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
  PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
  PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
  PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
  PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
  PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.

T_ZFUNCIONARIO-MATRICULA      = P_MATRI.
T_ZFUNCIONARIO-NOME          = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG            = P_RG .
T_ZFUNCIONARIO-CPF           = P_CPF.
APPEND T_ZFUNCIONARIO.

MOVE: T_ZFUNCIONARIO-MATRICULA      TO ZFUNCIONARIO-MATRICULA,
      T_ZFUNCIONARIO-NOME          TO ZFUNCIONARIO-NOME,
      T_ZFUNCIONARIO-DATANASCIMENTO TO ZFUNCIONARIO-DATANASCIMENTO,
      T_ZFUNCIONARIO-RG            TO ZFUNCIONARIO-RG,
      T_ZFUNCIONARIO-CPF           TO ZFUNCIONARIO-CPF.

INSERT ZFUNCIONARIO.

IF SY-SUBRC = '0'.
  COMMIT WORK.
ENDIF.
```

Onde SY-SUBRC = 0 é porque o comando INSER teve sucesso, podendo dar o comit, na tabela.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

O código Final do programa ZFUNCIONARIO, deverá ficar assim:

```
*&-----*
*& Report  ZFUNCIONARIO
*&
*&-----*
*&
*&
*&-----*

REPORT  ZFUNCIONARIO.

TABLES: ZFUNCIONARIO.

DATA: BEGIN OF T_ZFUNCIONARIO OCCURS 0,
      MATRICULA      LIKE ZFUNCIONARIO-MATRICULA,
      NOME           LIKE ZFUNCIONARIO-NOME,
      DATANASCIMENTO LIKE ZFUNCIONARIO-DATANASCIMENTO,
      RG             LIKE ZFUNCIONARIO-RG,
      CPF            LIKE ZFUNCIONARIO-CPF,
END OF T_ZFUNCIONARIO.

SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
      PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
      PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
      PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
      PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
      PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.

T_ZFUNCIONARIO-MATRICULA = P_MATRI.
T_ZFUNCIONARIO-NOME      = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG        = P_RG .
T_ZFUNCIONARIO-CPF       = P_CPF.
APPEND T_ZFUNCIONARIO.

MOVE: T_ZFUNCIONARIO-MATRICULA TO ZFUNCIONARIO-MATRICULA,
      T_ZFUNCIONARIO-NOME      TO ZFUNCIONARIO-NOME,
      T_ZFUNCIONARIO-DATANASCIMENTO TO ZFUNCIONARIO-DATANASCIMENTO,
      T_ZFUNCIONARIO-RG        TO ZFUNCIONARIO-RG,
      T_ZFUNCIONARIO-CPF       TO ZFUNCIONARIO-CPF.

INSERT ZFUNCIONARIO.

IF SY-SUBRC = '0'.
      COMMIT WORK.
ENDIF.
```

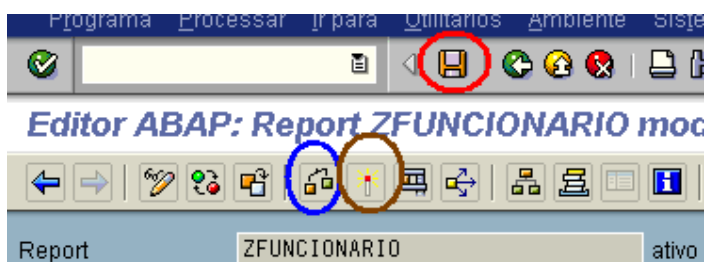
Vamos testar,

Primeiro siga as regras de **SALVAR, CHECAR e ATIVAR**

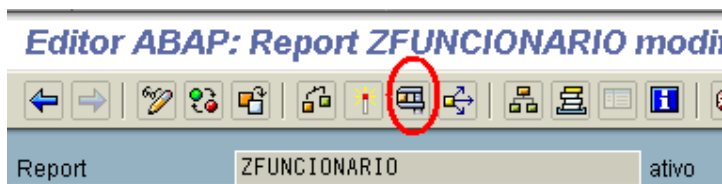
“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS





Após o status estar ativo, clique **DIRETO**, ou tecla **F8**.



Preencha os dados das **CAIXAS_DE_INSERTÃO**.

Cadastro de cliente

Matricula	9900
Nome	Uderson Luis
Data de Nascimento	1233
RG	1234
CPF	1222

Após preencher clique em **EXECUTAR** ou tecla **F8**.

No ícone com a seta **AZUL**.


É necessário preencher a **DATA DE NASCIMENTO**, de acordo o exemplo com os pontos, para não dar nenhum problema.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Cadastro de cliente

 ←

Matricula	1234
Nome	Uderson Luis
Data de Nascimento	22.06.2007 ←
RG	432644
CPF	34444





Após executar, não aparecerá nada, será deixado esta implementação de uma mensagem de retorno dizendo se teve sucesso ou não, para exercícios.


Verificando os dados inseridos.

Entre na transação SE16.

Nome da tabela: coloque o nome da tabela ZFUNCIONARIO na caixa de inserção.

Data Browser: 1ª tela

Nome da tabela 

Clique em conteúdo da tabela:

Data Browser: 1ª tela





Abrirá a seguinte tela:






“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Data Browser: tabela ZFUNCIONARIO: tela de seleção





 N° de entradas

MATRICULA	<input type="text"/>	até	<input type="text"/>	
NOME	<input type="text"/>	até	<input type="text"/>	
DATANASCIMENTO	<input type="text"/>	até	<input type="text"/>	
RG	<input type="text"/>	até	<input type="text"/>	
CPF	<input type="text"/>	até	<input type="text"/>	

Tecla **F8**, e abrirá a seguinte tela:

Data Browser: Tabela ZFUNCIONARIO 2 acertos

Tabela: ZFUNCIONARIO

Campos exibidos: 5 de 5 Colunas chave fixas: 1 Larg.lista 0250

	MATRICULA	NOME	DATANASCIMENTO	RG	CPF
<input type="checkbox"/>	0000000000000000123	UDERSON	08.05.2007	1234	223344
<input type="checkbox"/>	00000000000000001234	UDERSON LUIS	22.06.2007	432644	34444

A tabela que foi criada ZFUNCIONARIO, já possui dados dentro dela.

Exercícios.

Colocar uma mensagem de Erro ou Sucesso caso os dados foi inserido ou não na tabela, o comando para gerar mensagens pode ser:

MESSAGE I368(00) WITH 'mensagem'.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Manutenção de tabelas:

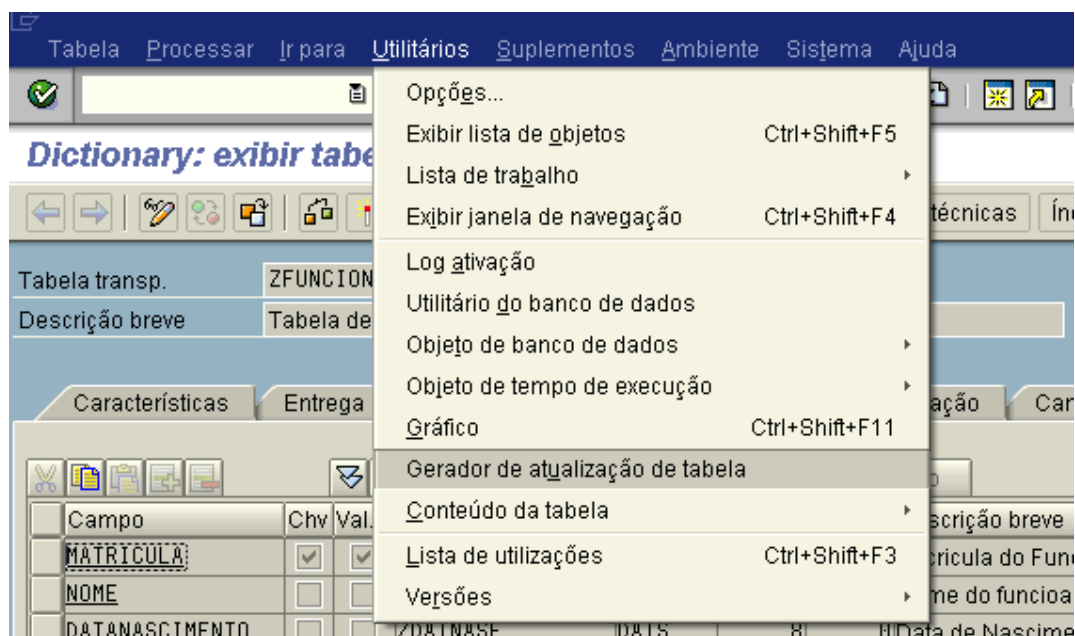
A manutenção de tabelas é uma transação de população de dados de uma determinada tabela, para popular uma tabela, não é necessário criar um programa report ou module pool ou qualquer outro tipo de programa ABAP, basta apenas criar uma manutenção de tabela.

Após ter criado a tabela, cliente no menu Utilitario >> Gerador de atualização de tabela.

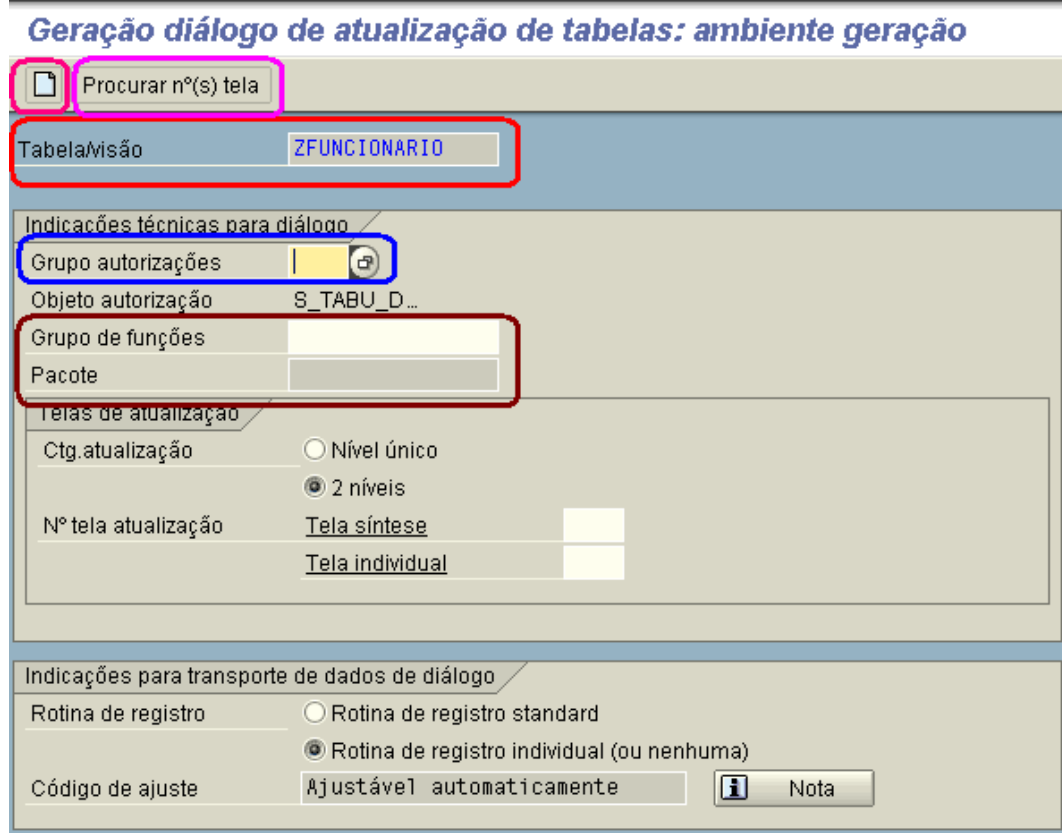
“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Será apresentada a seguinte tela:



Para entender:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

TABELA/VISÃO: Nome da tabela que será feita a manutenção

GRUPO DE AUTORIZAÇÃO: Tipo de permissão para popular os dados

GRUPO DE FUNÇÃO: Nome do Grupo de Função

PROCURAR NUMERO DE TELA: NUMETO TE TELAS DA TRANSAÇÃO

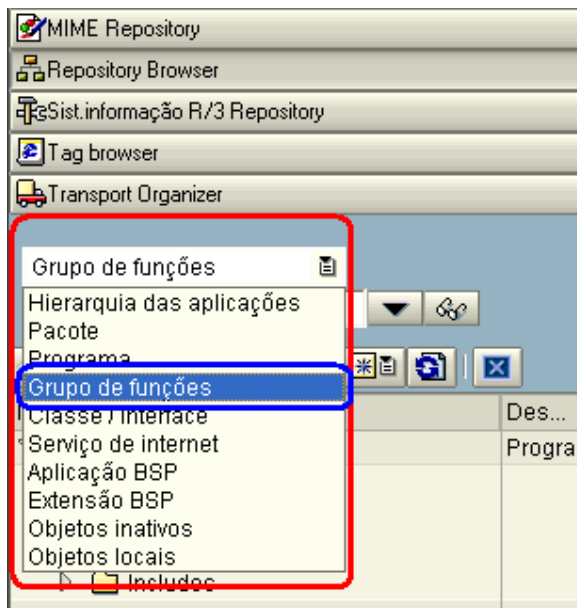
CRIAR: CRIAR MANUTENÇÃO

O grupo de autorização pode o mais adequado a cada caso, para iniciar iremos escolher a opção “Sem Grupo de Autorização” = &NC&.

O Grupo de função deve-ser o grupo específico para cada funcionalidades do sistemas criado, para criar um grupo de função, entre na transação SE80:



No Workbench, vá até Transport Organizer e escolha Grupo de função.

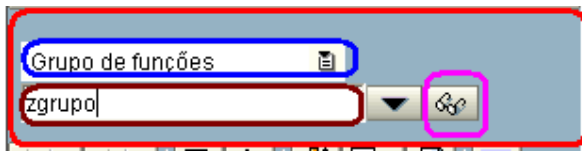


Após ter escolhida **Grupo de Função**, de o nome para o grupo de função: **ZGRUPO**, e Clique em **exibir**.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

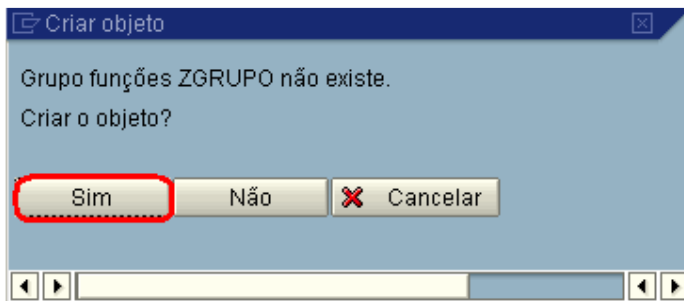


Grupo de funções

zgrupo

Gravar

A mensagem que deverá aparecer, é que ainda não existe este objeto e se deseja criar um novo (caso já exista este objeto troque o nome), escolha sim.

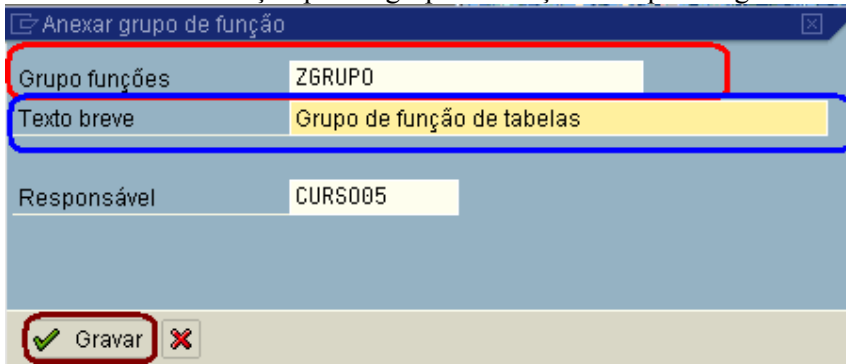


Criar objeto

Grupo funções ZGRUPO não existe.
Criar o objeto?

Sim Não Cancelar

Preencha uma descrição para o grupo de função e clique em gravar:



Anexar grupo de função

Grupo funções ZGRUPO

Texto breve Grupo de função de tabelas

Responsável CURS005

Gravar

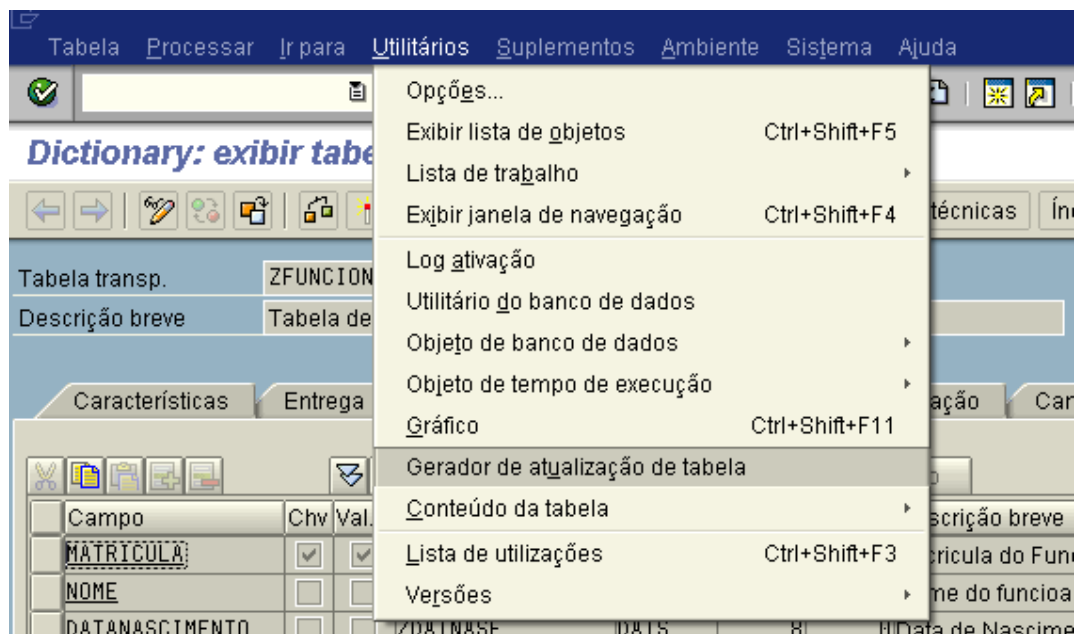
Voltando na manutenção:::

Após ter criado a tabela, cliente no menu Utilitario >> Gerador de atualização de tabela.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Após ter o grupo de função já criado basta preencher os campos :

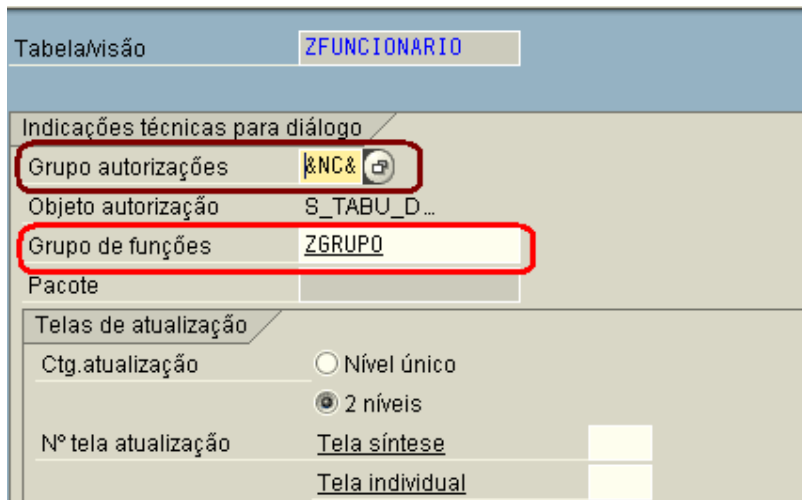
Grupo de autorização : “Sem Grupo de Autorização”

Grupo de função: “ZGRUPO”, ou o nome do grupo criado

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS



Tabela/visão ZFUNCIONARIO

Indicações técnicas para diálogo

Grupo autorizações &NC&

Objeto autorização S_TABU_D...

Grupo de funções ZGRUP0

Pacote

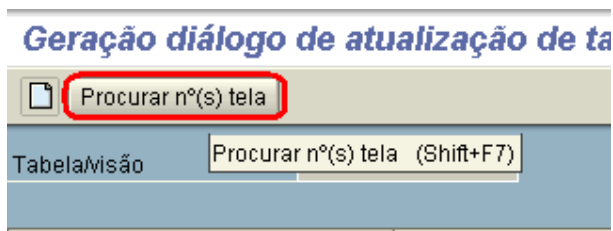
Telas de atualização

Ctg.atualização ☐ Nível único ☒ 2 níveis

Nº tela atualização Tela síntese

Tela individual

O terceiro passo é preencher o numero das telas, esta transação faz isto automático basta clicar em: Procurar números de tela (SHIFT + F7)

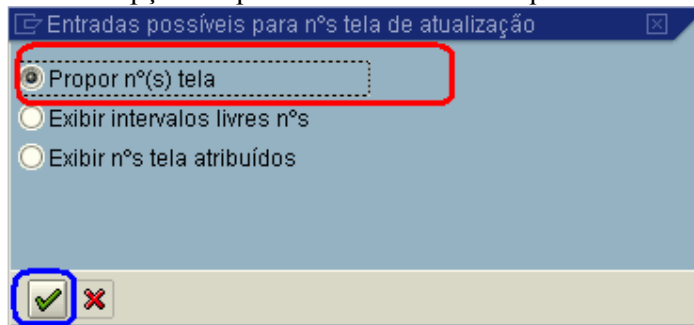


Geração diálogo de atualização de ta

Procurar nº(s) tela

Tabela/visão Procurar nº(s) tela (Shift+F7)

Deixe a opção Propor numero de tela e clique em OK



Entradas possíveis para nºs tela de atualização

☒ Propor nº(s) tela

☐ Exibir intervalos livres nºs

☐ Exibir nºs tela atribuídos

OK

Será preenchido os campo:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Indicações técnicas para diálogo

Grupo autorizações S/grp.autorizações

Objeto autorização

Grupo de funções

Pacote Objetos temporários, nunca são transporta

Telas de atualização

Ctg atualização ☐ Nível único

☒ 2 níveis

Nº tela atualização

Tela síntese	1
Tela individual	2

Indicações para transporte de dados de diálogo

Rotina de registro ☐ Rotina de registro standard

☒ Rotina de registro individual (ou nenhuma)

Código de ajuste

Após estes preenchimentos, clique em criar:

Tabela/visão

Indicações técnicas para diálogo

Grupo autorizações S/grp.a

Depois de criado volte a tabela.

Lembre-se de : SALVAR, CHECAR e ATIVAR, antes de sai da tabela

Para entrar na Manutenção de tabela, entre na transação sm30:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Atualização de visão de tabelas: 1ª tela

Procurar diálogo de atualização

Tabela/visão ZFUNCIONARIO

Restringir área de dados

☒ Sem restrições
☐ Entrar condições
☐ Variante

Exibir Atualizar Transporte Customizing

Coloque o nome da tabela que a qual será feita a manutenção e clique em exibir:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Para Entrar em modo de edição clique sobre o ícone (Modificar)

Modificar visão "Tabela de funcionarios": síntese

Entradas novas

Modificar -> exibir (Ctrl+F4)

Macricula do Funciona...	Nome do funcionario	Rg do funcionario	CPF Fur
65	654	3333	233
66	65	5435	534
123	UDERSON	1234	223344
234	EWR	FES	FSDF
432	43242	423	432
1234	UDERSON LUIS	432644	34444
3242	EWRW	REWR	RWER
3333	TESTES	6654	332
4324	FSDFSDF	6654	32131
5565	654	3333	233
12345	FERNANDO LINO	2233444	22323
12345678	TESTE DE DEBUG	2233445	2117788
123456789	TESTE DE DEBUG	2233445	2117788

Para entrar com um novo valor clique em : Entradas novas

Modificar visão "Tabela de funcionarios": síntese

Entradas novas

Seguindo o exemplo observe que o campo MT FUNC (Matricula do Funcionário), encontra-se sozinho, pois este campo é a chave primaria desta tabela.

"tudo posso naquele que me fortalece" (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

MT FUNC	NM FUNC	DT NASC	RG FUNC	CP FUNC

Para o usuário final não é legal deixar ele popular a tabela através da transação sm30, para isso deve-se criar uma transação para a população de dados desta tabela.

Para criar uma transação basta entrar na transação standard SE93.

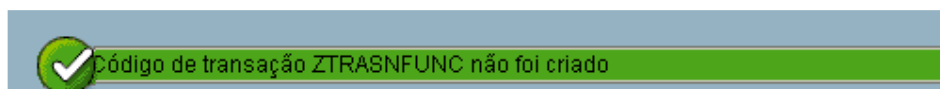
Código transação: ZTRASNFUNC

Exibir Modific. Criar

Entre com o nome da transação: ZTRASNFUNC

E clique em exibir.

Observe o status, para saber se ainda não existe esta transação.



Caso exista uma transação com este nome troque o nome para outro, senão clique em exibir:

Código transação: ZTRASNFUNC

Exibir Modific. Criar

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Preencha uma descrição para a transação
Escolha a opção “Transação com parâmetro”
Clique em Criar

Criar transação

Código de transação: ZTRASNFUNC

Atributos de transação

Texto breve: Transação da tabela ZFUNCIONARIO

Objeto de execução

- ☐ Programa e tela (transação de diálogo)
- ☐ Programa e tela de seleção (transação de report)
- ☐ Método de uma classe (transação OO)
- ☐ Transação com variante (transação de variantes)
- ☒ Transação com parâmetros (transação de parâmetros)

✓ ✗

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Na transação SE93, entre com a descrição da transação
“Texto da Transação”

Na transação, preencha com o código da transação SM30, pois é está a transação que manipula os dados, para que entre diretamente na tela de manutenção de dados da tabela, ative o campo “Ignorar a primeira tela”, para ir diretamente a tabela, deixe a opção “Transação Professional User”, habilitada, marque todas as opções de GUI, (“HTML”, “Java”, “Windows”).

A parte que se deve tomar muito cuidado é em “Valores Propostos”, pois e neste campo que será feito a ligação da transação SM30 para a tabela ZFUNCIONARIO (ou qualquer outra tabela), preencha conforme a figura.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Texto da transação Transação da tabela ZFUNCTIONARIO

Valor proposto para

Transação SM30

☐ Ignorar 1ª tela

Obsoleto: utilizar valores propostos para transação

Tela 0

Do pool módulos

Classificação

☐ Herdar características GUI

Classificação de transação

☒ Transação Profissional User

☐ Transação EasyWeb

☐ Ativo globalmente

Serviço

Capacidade GUI

☒ SAP GUI para HTML

☒ SAP GUI para Java

☒ SAP GUI para Windows

Vals.propostos

Nome da área de tela	Valor
VIEWNAME	ZFUNCTIONARIO
UPDATE	X

Após ter preenchido os campos conforme descrito salve.



A transação para a manutenção de tabelas já está pronta, entre com o código da transação e automaticamente será apresentada a tela de manutenção de tabela da tabela descrita (ZFUNCTIONARIO).

ZTRASNFUNC



“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Resposta do exercício:

```
*&-----*
*& Report  ZFUNCIONARIO *
*& *
*&-----*
*& *
*& *
*&-----*
```

REPORT ZFUNCIONARIO.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

TABLES: ZFUNCIONARIO.

```
DATA: BEGIN OF T_ZFUNCIONARIO OCCURS 0,
      MATRICULA      LIKE ZFUNCIONARIO-MATRICULA,
      NOME           LIKE ZFUNCIONARIO-NOME,
      DATANASCIMENTO LIKE ZFUNCIONARIO-DATANASCIMENTO,
      RG             LIKE ZFUNCIONARIO-RG,
      CPF            LIKE ZFUNCIONARIO-CPF,
END OF T_ZFUNCIONARIO.

SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
  PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
  PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
  PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
  PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
  PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.

T_ZFUNCIONARIO-MATRICULA = P_MATRI.
T_ZFUNCIONARIO-NOME      = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG        = P_RG .
T_ZFUNCIONARIO-CPF       = P_CPF.
APPEND T_ZFUNCIONARIO.

MOVE: T_ZFUNCIONARIO-MATRICULA TO ZFUNCIONARIO-MATRICULA,
      T_ZFUNCIONARIO-NOME      TO ZFUNCIONARIO-NOME,
      T_ZFUNCIONARIO-DATANASCIMENTO TO ZFUNCIONARIO-DATANASCIMENTO,
      T_ZFUNCIONARIO-RG        TO ZFUNCIONARIO-RG,
      T_ZFUNCIONARIO-CPF       TO ZFUNCIONARIO-CPF.

INSERT ZFUNCIONARIO.

IF SY-SUBRC = '0'.
  COMMIT WORK.
  MESSAGE I368(00) WITH 'DADOS SALVOS COM SUCESSO'.
ELSE.
  ROLLBACK WORK.
  MESSAGE I368(00) WITH 'ERRO AO SALVAR DADOS'.
ENDIF.
```

Usando WRITE e SELECT.

A transação sel6, pode ser simulada dentro do exemplo, para fazer esta simulação é necessário, utilizar dois comando WRITE e SELECT.

Write: este comando escreve dados em formato relatório, pode-se passas como argumento para este comando qualquer tipo de dado, pois todos serão convertidos em formato char, e serão impressos em um relatório, dentro do SAP.

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Sisntaxe:

WRITE 'valor'

WRITE: / 'valor', onde '/', faz pular uma linha.

DATA: v_valor type i value '2'.

WRITE 'O valor é ', v_valor. , Utilizando o write com um valor char e uma variavel do tipo i.

E assim por diante o write.

SELECT: select é uma intrução do SQL (Structure Query Languagem), que faz a seleção de dados de uma ou mais tabelas de um banco de dado, este comando pode ser combinado, com diversos outros comando das instruções SQL, no exemplo usaremos a seguinte sintaxe:

SELECT campos **FROM** tabela_real **INTO TABLE** tabela_interna.

Onde:

Campos = os campos da tabela real que construímos.

Tabela_real = tabela ZFUNCIONARIO.

Tabela_interna = tabela Interna usada dentro do código

Exemplo:

```
SELECT matricula nome datanascimento rg cpf  
      FROM zfuncionario  
      INTO TABLE t_funcionario.
```

Observação: Na sintaxe do select tem o comando INTO TABLE, que é um comando exclusivo do dicionário de dados OPEN SQL da linguagem ABAP, este comando faz com que os dados selecionados da tabela real sejam transferidos a uma tabela interna que tenha a mesma **ESTRUTURA** e **ORDEM** dos campos selecionados. A tabela interna **t_funcionario** tem a mesma ordem e estrutura dos campos **matricula nome datanascimento rg cpf**.

No código do exemplo ficará:

```
REPORT ZFUNCIONARIO.
```

```
TABLES: ZFUNCIONARIO.
```

```
BREAK-POINT.
```

```
DATA: BEGIN OF T_ZFUNCIONARIO OCCURS 0,
```

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

```
MATRICULA      LIKE ZFUNCIONARIO-MATRICULA,
NOME            LIKE ZFUNCIONARIO-NOME,
DATANASCIMENTO LIKE ZFUNCIONARIO-DATANASCIMENTO,
RG              LIKE ZFUNCIONARIO-RG,
CPF             LIKE ZFUNCIONARIO-CPF,
END OF T_ZFUNCIONARIO.

SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
  PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
  PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
  PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
  PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
  PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.

T_ZFUNCIONARIO-MATRICULA = P_MATRI.
T_ZFUNCIONARIO-NOME     = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG       = P_RG.
T_ZFUNCIONARIO-CPF      = P_CPF.
APPEND T_ZFUNCIONARIO.

MOVE: T_ZFUNCIONARIO-MATRICULA TO ZFUNCIONARIO-MATRICULA,
      T_ZFUNCIONARIO-NOME      TO ZFUNCIONARIO-NOME,
      T_ZFUNCIONARIO-DATANASCIMENTO TO ZFUNCIONARIO-DATANASCIMENTO,
      T_ZFUNCIONARIO-RG        TO ZFUNCIONARIO-RG,
      T_ZFUNCIONARIO-CPF       TO ZFUNCIONARIO-CPF.

INSERT ZFUNCIONARIO.

IF SY-SUBRC = '0'.
  COMMIT WORK.
  MESSAGE I368(00) WITH 'DADOS SALVOS COM SUCESSO'.
ELSE.
  ROLLBACK WORK.
  MESSAGE I368(00) WITH 'ERRO AO SALVAR DADOS'.
ENDIF.

CLEAR: T_ZFUNCIONARIO, T_ZFUNCIONARIO[].

      SELECT MATRICULA NOME DATANASCIMENTO RG CPF
      FROM ZFUNCIONARIO
      INTO TABLE T_ZFUNCIONARIO.
```

Observe que tem um novo comando antes do SELECT, que é CLEAR, o comando, clear, faz a limpeza da **tabela e da headerline**, como a tabela e a headerline estava com dados, pois antes deste comando tínhamos usado a tabela interna t_funcionario para coletar as informações dos campos de inserção. Para preenchê-las novamente é necessário fazer a limpeza, para que não fique com sujeira de dados, em memória.

No comando SELECT estamos selecionando os campos “**MATRICULA NOME DATANASCIMENTO RG CPF**” da tabela **ZFUNCIONARIO**, e estamos inseridos dos dados correspondentes dentro da tabela interna **T_ZFUNCIONARIO**.

Exemplificando:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

tabela **ZFUNCIONARIO**:

A tabela ZFUNCIONARIO contem os seguintes registros dados:

MATRICULA	NOME	DATANASCIMENTO	RG	CPF
00012	Uderson Luis	19900220	4455778899	2233556644
00013	Luis Soares	19500114	5566221100	2233889977

Tabela interna **T_ZFUNCIONARIO**.

MATRICULA	NOME	DATANASCIMENTO	RG	CPF

```
SELECT MATRICULA NOME DATANASCIMENTO RG CPF
FROM ZFUNCIONARIO
INTO TABLE T_ZFUNCIONARIO.
```

Tabela interna **T_ZFUNCIONARIO**.

Agora a tabela interna **T_ZFUNCIONARIO**, não está mais vazia, pois o comando select preencheu a tabela, observe que a tabela interna tem a mesma estrutura que a tabela real, não é necessário seguir esta ordem, porem o que tem que ser seguido é ordem dos campo no select ser idêntico a estrutura da tabela interna.

MATRICULA	NOME	DATANASCIMENTO	RG	CPF
00012	Uderson Luis	19900220	4455778899	2233556644
00013	Luis Soares	19500114	5566221100	2233889977

Supondo que ao invés de termos a tabela interna idêntica com a estrutura da tabela real, temos a estrutura seguinte da tabela interna, os campos não estão na mesma ordem e não esta com todos os campos.

Tabela interna **T_ZFUNCIONARIO**.

RG	MATRICULA	DATANASCIMENTO	CPF

O select para preencher esta tabela interna seria:

```
SELECT RG MATRICULA DATANASCIMENTO CPF
FROM ZFUNCIONARIO
INTO TABLE T_ZFUNCIONARIO.
```

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

No código já está quase pronto, pois já inserimos, valores na tabela real, verificamos os valores inseridos na transação SE16, limpamos as tabelas internas, voltamos a preencher a tabela interna, com o comando SELECT, agora iremos imprimir os dados em formato relatório.

Usaremos o comando LOOP, para andar em cada registro da tabela interna, sua sintaxe é a seguinte:

LOOP AT tabela_interna.

ENDLOOP.

Ou loop condicional

LOOP AT tabela_interna **WHERE** algum_campo_da_tabela_interna = ‘’.

ENDLOOP.

A tabela interna que temos é T_ZFUNCIONARIO, então a sintaxe do comando ficará:

LOOP AT T_ZFUNCIONARIO.

ENDLOOP.

Com este comando conseguimos verificar cada registro da tabela interna, basta agora imprimir estes registros com o comando Write. Lembre-se que este trecho de código deverá ficar embaixo do SELECT.

LOOP AT T_ZFUNCIONARIO.

WRITE: / T_ZFUNCIONARIO-MATRICULA, T_ZFUNCIONARIO-NOME,
T_ZFUNCIONARIO-DATANASCIMENTO,
T_ZFUNCIONARIO-RG, T_ZFUNCIONARIO-CPF.

ENDLOOP.

Ao executar o programa será necessário criar um novo registro, pois este é o início do programa e após inserir um novo registro, será impresso todos os registros. Deverá aparecer um relatório parecido com este:

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Cadastro de cliente

Cadastro de cliente				
000000000000000001234	UDERSON LUIS	22.06.2007	432644	34444
000000000000000003333	TESTES	12.12.2007	6654	332
000000000000000000234	EWR	22.05.2007	FES	F5DF
0000000000000000012345	FERNANDO LINO	27.06.2007	2233444	22323
000000000000000003242	EWRW	30.05.2007	REWR	RWR
000000000000000000123	UDERSON	08.05.2007	1234	223344
000000000000000004324	F5DF5DF	04.07.2007	6654	32131

Para ficar melhor usaremos os comandos ULINE e o caracter “|”.

ULINE, Cria uma linha Horizontal, e ‘|’ uma linha Vertical

```

LOOP AT T_ZFUNCIONARIO.
  WRITE: /  ' |', T_ZFUNCIONARIO-MATRICULA,
            ' |', T_ZFUNCIONARIO-NOME,
            ' |', T_ZFUNCIONARIO-DATANASCIMENTO,
            ' |', T_ZFUNCIONARIO-RG,
            ' |', T_ZFUNCIONARIO-CPF, ' |'.
  ULINE.
ENDLOOP.

```

Ao executar o programa será necessário criar um novo registro, pois este é o início do programa e após inserir um novo registro, será impresso todos os registros. Deverá aparecer um relatório parecido com este:

Cadastro de cliente

Cadastro de cliente				
000000000000000000001234	UDERSON LUIS	22.06.2007	432644	34444
000000000000000000003333	TESTES	12.12.2007	6654	332
00000000000000000000234	EUR	22.05.2007	FES	F5DF
0000000000000000000012345	FERNANDO LINO	27.06.2007	2233444	22323
000000000000000000003242	EURW	30.05.2007	REWR	RWER
000000000000000000000034	43424	05.06.2007	4324234	4324234
000000000000000000000123	UDERSON	08.05.2007	1234	223344
000000000000000000004324	F5DF5DF	04.07.2007	6654	32131

Código Final.

```
*&-----*
*& Report  ZFUNCIONARIO
*&-----*
```

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

```
*&-----*
*&
*&
*&-----*

REPORT ZFUNCIONARIO.

TABLES: ZFUNCIONARIO.
BREAK-POINT.
DATA: BEGIN OF T_ZFUNCIONARIO OCCURS 0,
        MATRICULA      LIKE ZFUNCIONARIO-MATRICULA,
        NOME           LIKE ZFUNCIONARIO-NOME,
        DATANASCIMENTO LIKE ZFUNCIONARIO-DATANASCIMENTO,
        RG             LIKE ZFUNCIONARIO-RG,
        CPF            LIKE ZFUNCIONARIO-CPF,
    END OF T_ZFUNCIONARIO.

SELECTION-SCREEN BEGIN OF BLOCK b1001 WITH FRAME.
    PARAMETER: P_MATRI LIKE ZFUNCIONARIO-MATRICULA OBLIGATORY.
    PARAMETER: P_NOME  LIKE ZFUNCIONARIO-NOME OBLIGATORY.
    PARAMETER: P_DATNA LIKE ZFUNCIONARIO-DATANASCIMENTO OBLIGATORY.
    PARAMETER: P_RG    LIKE ZFUNCIONARIO-RG OBLIGATORY.
    PARAMETER: P_CPF   LIKE ZFUNCIONARIO-CPF OBLIGATORY.
SELECTION-SCREEN END OF BLOCK b1001.

T_ZFUNCIONARIO-MATRICULA = P_MATRI.
T_ZFUNCIONARIO-NOME      = P_NOME.
T_ZFUNCIONARIO-DATANASCIMENTO = P_DATNA.
T_ZFUNCIONARIO-RG        = P_RG .
T_ZFUNCIONARIO-CPF       = P_CPF.
APPEND T_ZFUNCIONARIO.

MOVE: T_ZFUNCIONARIO-MATRICULA TO ZFUNCIONARIO-MATRICULA,
      T_ZFUNCIONARIO-NOME      TO ZFUNCIONARIO-NOME,
      T_ZFUNCIONARIO-DATANASCIMENTO TO ZFUNCIONARIO-DATANASCIMENTO,
      T_ZFUNCIONARIO-RG        TO ZFUNCIONARIO-RG,
      T_ZFUNCIONARIO-CPF       TO ZFUNCIONARIO-CPF.

INSERT ZFUNCIONARIO.

IF SY-SUBRC = '0'.
    COMMIT WORK.
    MESSAGE I368(00) WITH 'DADOS SALVOS COM SUCESSO'.
ELSE.
    ROLLBACK WORK.
    MESSAGE I368(00) WITH 'ERRO AO SALVAR DADOS'.
ENDIF.

CLEAR: T_ZFUNCIONARIO, T_ZFUNCIONARIO[].

        SELECT MATRICULA NOME DATANASCIMENTO RG CPF
        FROM ZFUNCIONARIO
        INTO TABLE T_ZFUNCIONARIO.

LOOP AT T_ZFUNCIONARIO.

WRITE: /   '|', T_ZFUNCIONARIO-MATRICULA,
        '|', T_ZFUNCIONARIO-NOME,
        '|', T_ZFUNCIONARIO-DATANASCIMENTO,
        '|', T_ZFUNCIONARIO-RG,
        '|', T_ZFUNCIONARIO-CPF, '|'.

        ULINE.
```

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Existem tabelas que não conhecemos suas transações para suas devidas manutenções, para encontrar as manutenções de tabelas existe um macete, que é fazer a procura em uma tabela STANDARD, lembrando que tudo no R/3 está praticamente em tabelas. A tabela TSTCP armazena os dados de Parâmetro para transações a sua estrutura é a seguinte:

TCODE	TCODE	CHAR 20	0	Código de transação
PARAM	TCDPARAM	CHAR 254	0	Parâmetro p/transações

Esta tabela armazena o código da transação e os parâmetros da manutenção aqueles que são inseridos na transação SE93.

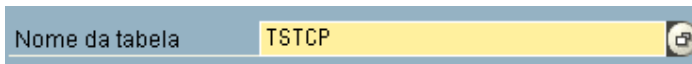
Para encontrar a manutenção de uma tabela basta fazer um filtro relativo a uma tabela ou qualquer tabela.

Exemplo prático, encontrar a transação que faz a manutenção da tabela ZFUNCIONARIO.

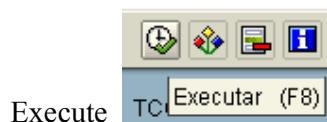
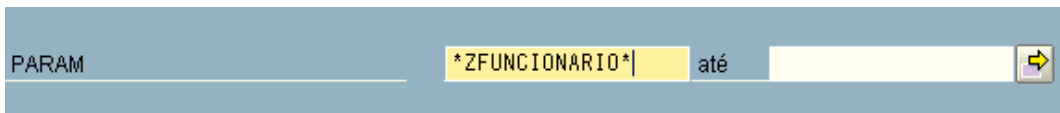
Entre na transação SE16 (pode-se qualquer forma de se filtrar um determinado dado), usaremos esta transação para exemplificar:



Entre como o nome da tabela que armazena os dados (TSTCP)



Entre com o nome da tabela a ser procurada:



TCODE	PARAM
ZTRASNFUNC	/*SM30 VIEWNAME=ZFUNCIONARIO;UPDATE=X;

“tudo posso naquele que me fortalece” (Filipenses 4:13).

Criando uma tabela Z com Elementos de Dados e Domínios

UDERSON LUIS

Observe que usamos a tabela criada neste tutorial, para exemplificar, e foi retornada a função que foi criada neste tutorial, lembrando que estes passos servem para achar informações de qualquer tabela.

	TCODE	PARAM
<input type="checkbox"/>	ZTRASNFUNC	/*SM30 VIEWNAME=ZFUNCIONARIO;UPDATE=X;