

SAP ABAP

2.

Criar primeira tabela

1. Fazer o login no ambiente correto, informado pelo cliente ou treinamento;
2. digitar na caixa de comando o código **SE11**;
3. outra informação importante é que todo objeto customizado deve ter seu nome iniciado com Z ou Y, por isso o elemento abaixo tem nome ZEL...

Tela de dicionário de dados:

ABAP Dictionary: entrada

Tab.banco dados: ZTAB01_11

Visão:

Categoria dados: ZEL01_11

Grupo de tipos:

Domínio: ZDM01_11

Ajuda de pesquisa:

Objeto de bloqueio:

Exibir Modif. Criar

3. com a caixa do Tab. banco dados = digitar o nome da sua tabela de dados;
4. clicar no botão [criar];
5. Na tela Dicionário: modificar tabela, inserir a breve descrição da tabela;
6. Ainda na tela Dicionário: modificar tabela aba Entrega atualização preencher o campo Classe de entrega com a categoria [A], Data Browser: opção Exibição/ Atualização permitida

Dictionary: modificar tabela

Tabela transp.: ZTAB01_11 novo(Revisado)

Descrição breve: Primeira tabela atividade

Características Entrega e atualização Campos Entrs.possíveis/verificação Campos moeda/quantidade Índices

Classe de entrega: A Tabela de aplicação (dados mestre e de movimento)

Data Browser/atualiz.vision tabs: Exibição/atualização permitida

Gravar Cancelar

7. Agora vamos clicar na aba campos
8. primeiro campo para essa aba será MANDT(mandante) e esse terá Ch e val marcados, lembrando que o campo caso ainda não exista vc pode digitar seu nome ai, clicar duas vezes no nome que vai ficar sublinhado que o programa te direciona para uma tela onde poderá criar esse campo.

[illegible]

9. após o campo mandante sim vamos adicionar o campo código.

SAP

Dictionary: modificar elemento de dados

Elemento de dados: ZEL01_11 novo(Revisado)

Descrição breve:* CODIGO

Caracts.	Ctg.dds.	Caracts.adicionis.	Denomin.campos
<input checked="" type="radio"/> Categoria elementar <input checked="" type="radio"/> Domínio	ZDMG1_11 código		
	Ctg.dds.: INT1 Nº inteiro de 1 byte, 0		
	Compr: 3		
<input type="radio"/> Tipo incorporado	Ctg.dds.:		
	Compr: 0		
<input type="radio"/> Nome tipo refer.			
<input type="radio"/> Referência a tipo instalado			
	Tp.dados:		
	Compr: 0		

X

Criar entrada catálogo objetos

Objeto: RSTR DTEL ZEL01_11

Atributos

Pacote: ZTREINAMENTO

Responsável: CURSOR2

Sistema de origem: S4H

Idioma original: PT Português

Data de criação:

Objeto local
 Síntese bloqueio
 X

10. após clicar duas vezes no campo ele vai pedir informações do campo, como tamanho do mesmo, tipo de dados e pacote, acima temos um exemplo de preenchimento.
11. preencher tbm o Denomcampo com o nome do campo novamente.
12. Após isso vamos voltar para a tela dos campos da tabela e vamos chamar esse elemento na mesmo o resultado será:

SAP Dictionary: modificar tabela

Tabela transp.: ZTAB01_11 novo(Revisado)

Descrição breve: Primeira tabela atividade

Características Entrega e atualização Campos Entrs.possíveis/verificação Campos moeda/quantidade Índices

Campo	Chv	Valor	Elemento dados	CtgDados	Compr	Casas d.	SistCoords.	Descrição breve	Grupo
<input type="checkbox"/> MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0		0 Mandante	
<input type="checkbox"/> CODIGO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL01_11	INT1	3	0		0 CODIGO	

Gravar Cancelar

12. os passos anteriores devem se seguir para os demais campos da sua tabela sempre criando Domínio e Elemento

SAP Dictionary: modificar domínio

Domínio: ZDM02_11 novo(Revisado)

Descrição breve: NOME CAMPO NOME

Caracts. Definição Intervals.

Formato

CtgDados: CHAR Cadeia de caracteres

Nº de posições: 38

Casas decimais: 0

Características da saída

Comprim.saída: 38

Rotina conversão

☐ Sinal +/-

☐ Letr.minúsculas

Cursor mal posicionado ou objeto não incluído na navegação Visualizar detalhes

Gravar Cancelar

13. após informar todos os campos necessários, podemos ativar a tabela, para isso, caso não haja inconsistências nos campos da tabela, você vai ser direcionado para a seguinte tela:

ABAP Dictionary: atualizar configurações técnicas

Nome: ZTAB01_11 Transparente Tabela

Descrição breve: Primeira tabela atividade

Última modif.: CURS002 19.06.2023

Status: nv. não gravado

Características gerais Caracts.específicas BD

Parâmetros de memória lógicos

Categoria dados: APPL0

Ctg.tamanho: 1

Compartilhamento de tabelas

Tp.compartmt.: Não classificado

Exibir

Armazenamento em buffer

☒ Armazenamento em buffer não permitido

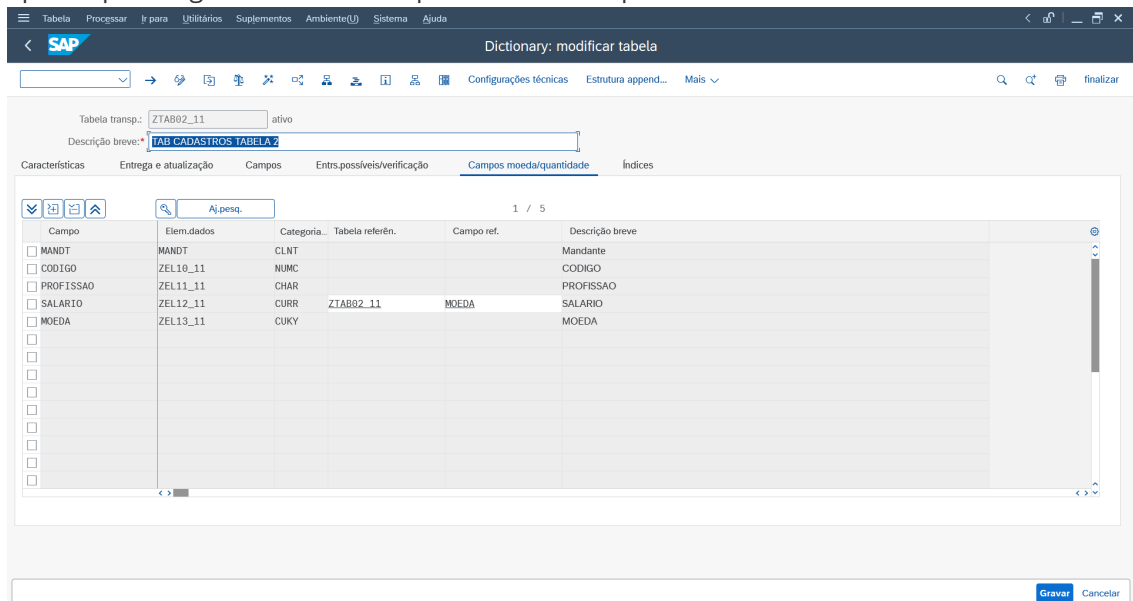
☐ Arm.buffer permitido, mas desativado

☐ Armazenamento em buffer ativado

Tipo de armazenamento em buffer

Gravar Avançar Cancelar

14. no campo categoria escolher a opção APPL0 no campo ctg. tamanho podemos deixar a opção 1
15. apenas para registro de um exemplo de dado do tipo salario em tabela:

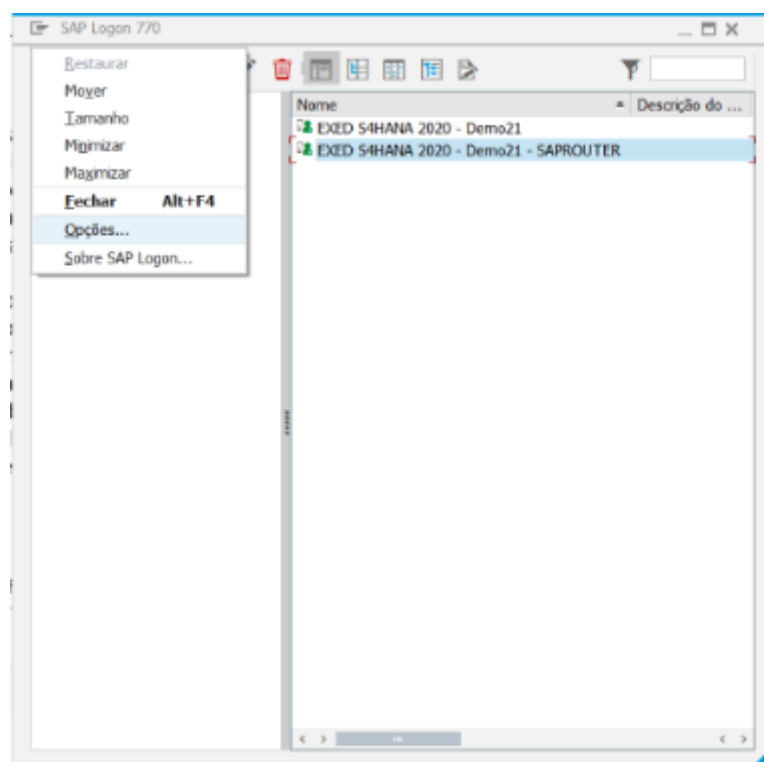


16. note na imagem acima que o campo salario esta relacionado com o campo moeda onde poderemos informar o Símbolo da moeda para o dinheiro

Trocando o tema do SAP LOGON:

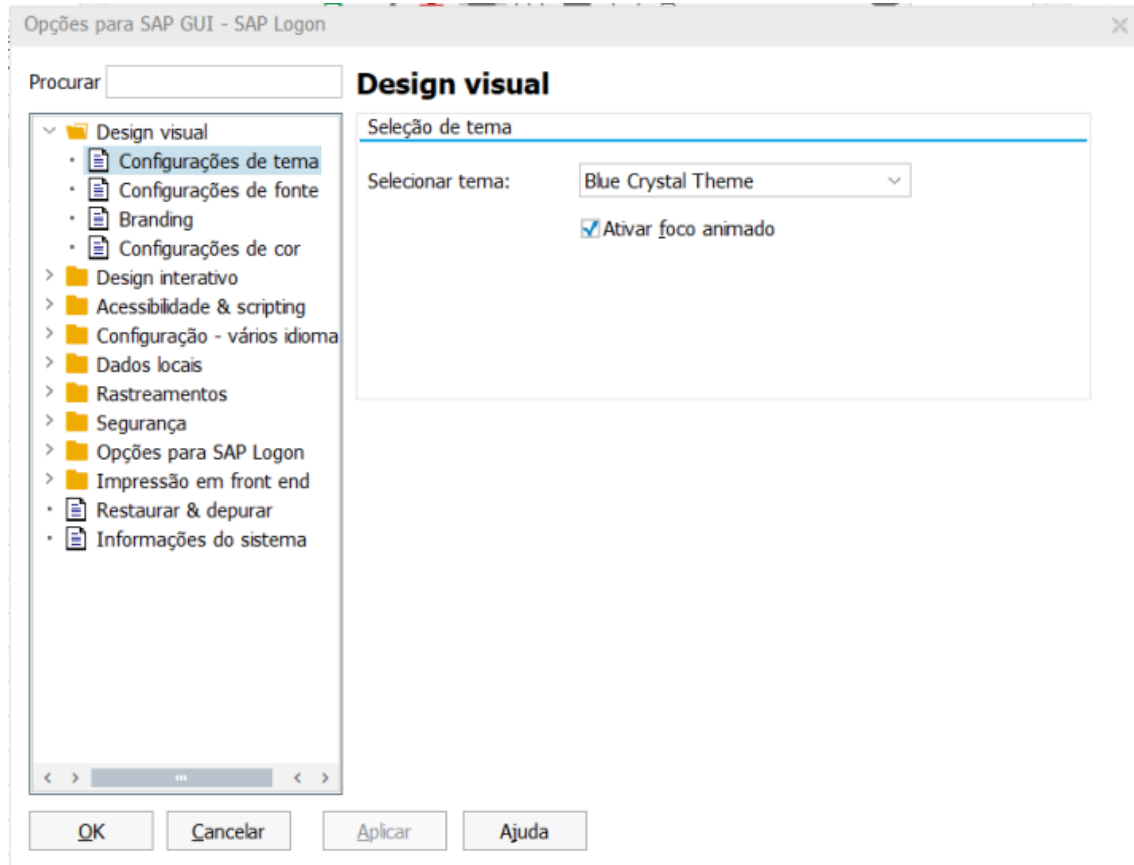
A tela do SAP LOGON dispõem de alguma opções de temas para trabalho, para alterar o tema faça o seguinte passo a passo:

1. na tela inicial do SAP Logon, antes de realizar o login você vai clicar no primeiro botão no canto superior esquerdo da tela:



2. clique na opção opções

3. expandir a opção Designer visual e clicar na opção Configurações de tema



4. selecionar no campo selecionar tema o tema desejado, estamos utilizando nesse documento o tema: Blue Crystal Theme

5. clicar no botão **Aplicar** e depois em **OK**

6. caso ainda não esteja vendo as alterações, feche o programa e abra novamente

ABAP Open SQL

O ABAP faz uso de SQL para suas consultas ao banco de dados e aqui a lógica é a mesma para quem já fez uso de SQL

- **Selecionar dados** - Comando `SELECT campo From Tabela` também é permitido utilizar critérios WHERE e relação entre tabelas com JOIN;
- **Inserir dados** no banco - comando `INSERT campo = 'valor' INTO tabela`
- **Alterar dados** no banco - comando `UPDATE tabela SET campo = "informação" WHERE codigo = 1` ou `MODIFY tabela FROM campo = "informação"`, diferença entre esses dois comandos é que o Modify cria um novo registro caso não encontrar o mesmo para atualizar no banco.
- **Apagar dados** no banco = comando `DELETE * FROM tabela WHERE codigo = 1`, uma vez apagado o registro não é possível recuperar o mesmo.

Trabalhando informações na tabela

Report - é um recurso que te permite processar os dados, como por exemplo exibir os dados. esse recurso é feito através da **transação SE38**. Esse recurso é utilizado para gerar relatórios com os dados da sua base de dados. Esse pode ser executado on-line ou em background, a diferença entre os dois modos é que o modelo background permite executar processamentos de grandes volumes de dados sem travar a tela para o usuário.

É importante manter a estrutura do programa organizada e isso pode ser feito dividindo o Reporte com linhas do cabeçalho e comentários indicando partes do código. Exemplo, podemos indicar através do comentário/ cabeçalho que uma parte do código é responsável pelos dados do relatório com dados dos funcionários da base de dados.

OBS: veja se seu cliente já possui um modelo de cabeçalho e siga sempre o padrão já utilizado.

Outra dica sobre organização do código é separar as declarações por blocos e não criar tudo junto e misturado = tabelas, constantes...

- Tabelas internas - iniciamos o nome com t, exemplo: t_table
- variáveis globais - iniciamos o nome com v, exemplo: v_codigo
- constantes - iniciamos o nome com c, exemplo: c_dado

Tela de seleção:

Inicia-se com Selection-Screen Begin e encerra em Selection-Screen end

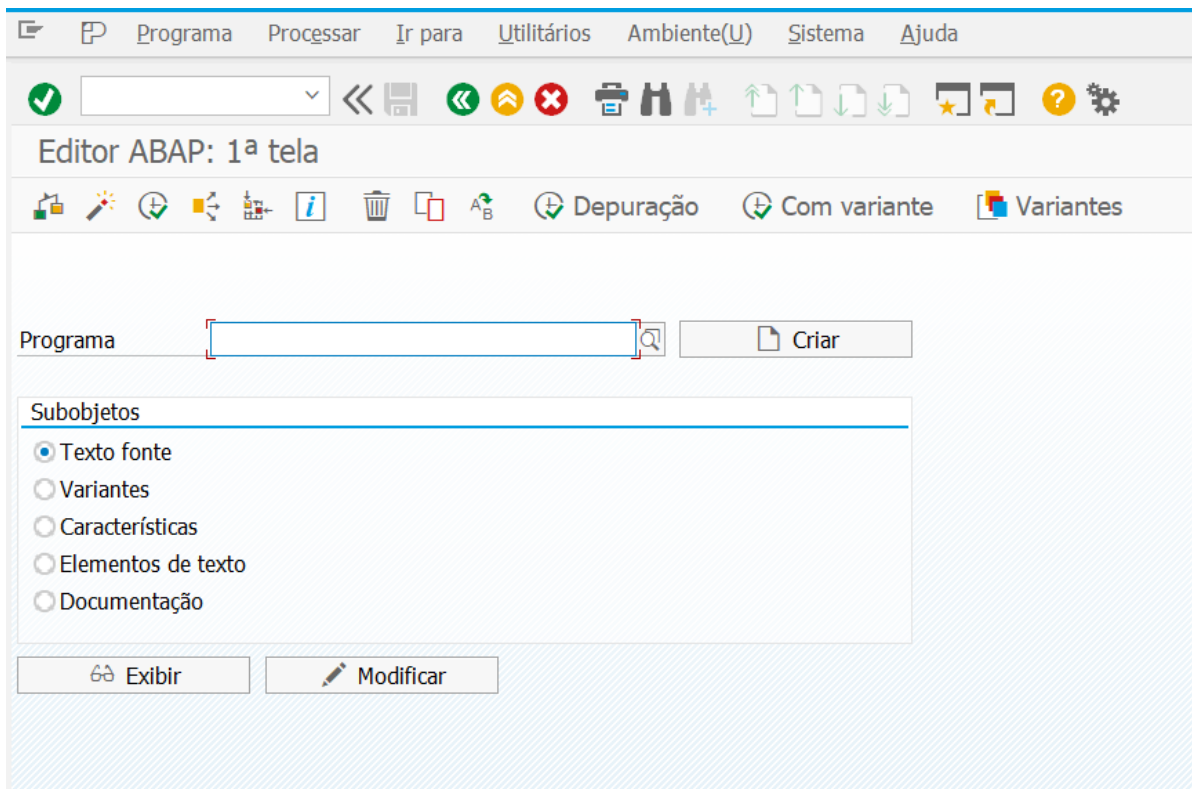
Condicional IF:

```
IF V_X EQ 0.  
    WRITE 'TEXTO DE SAIDA'.  
ELSE.  
    WRITE 'TEXTO DE SAIDA'.  
ENDIF.
```

ou podemos ainda usar **CASE**:

```
CASE v_dado,  
    WHEN 'valoresperado-A'.  
        WRITE 'TEXTO DE SAIDA'.  
  
    WHEN 'valoresperado-B'.  
        WRITE 'TEXTO DE SAIDA'.  
  
    WHEN others.  
ENDCASE.
```

Tela inicial do SE38:



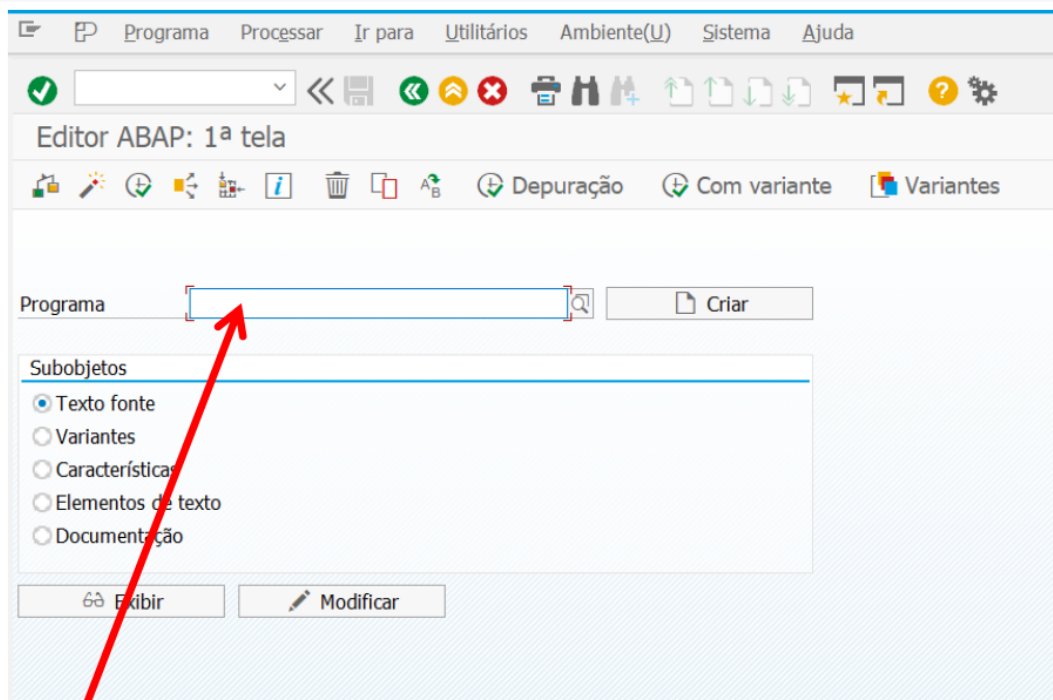
Criar Report

Para criar um Report vamos seguir a padronização a seguir: o nome deve ser ZREPX_XX onde o X é o número de report criado: primeiro, segundo... e XX é o número de aluno(a).

Devemos retornar o nome e país registrado na tabela ZTAB01_xx criada na tabela 1 criada.

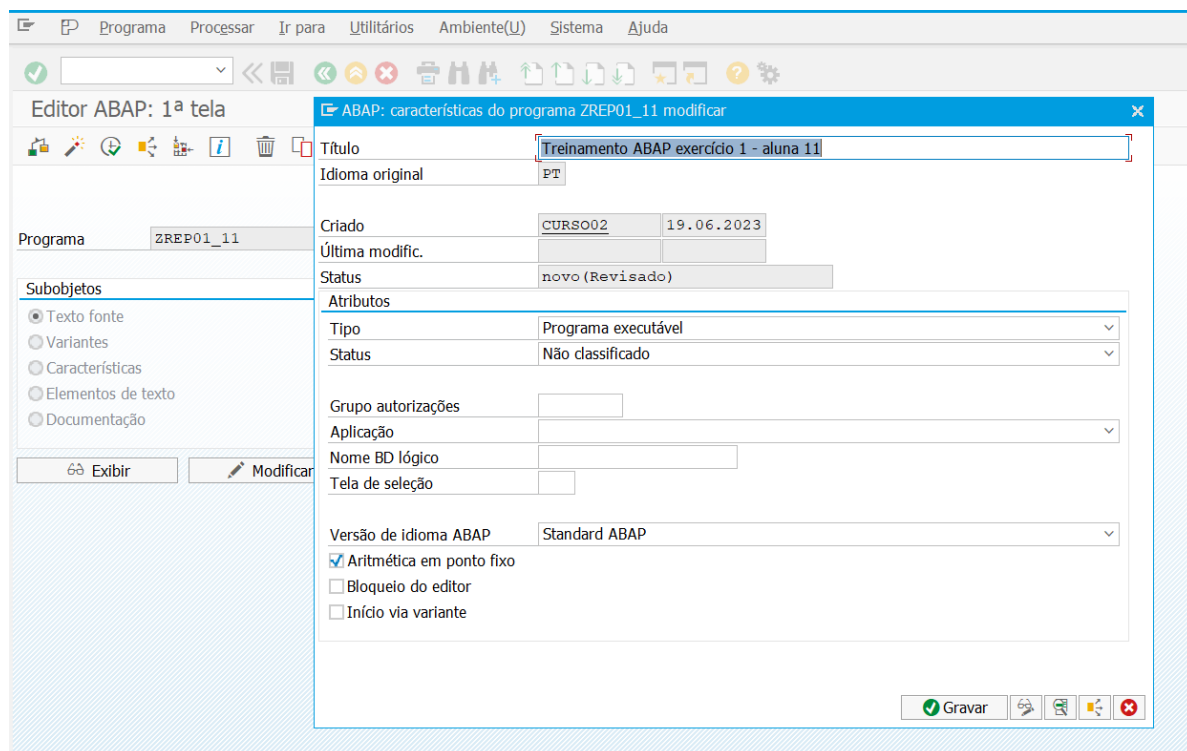
Passo a passo:

1. Preencher o nome do programa ZREP01_XX e clicar no botão Criar

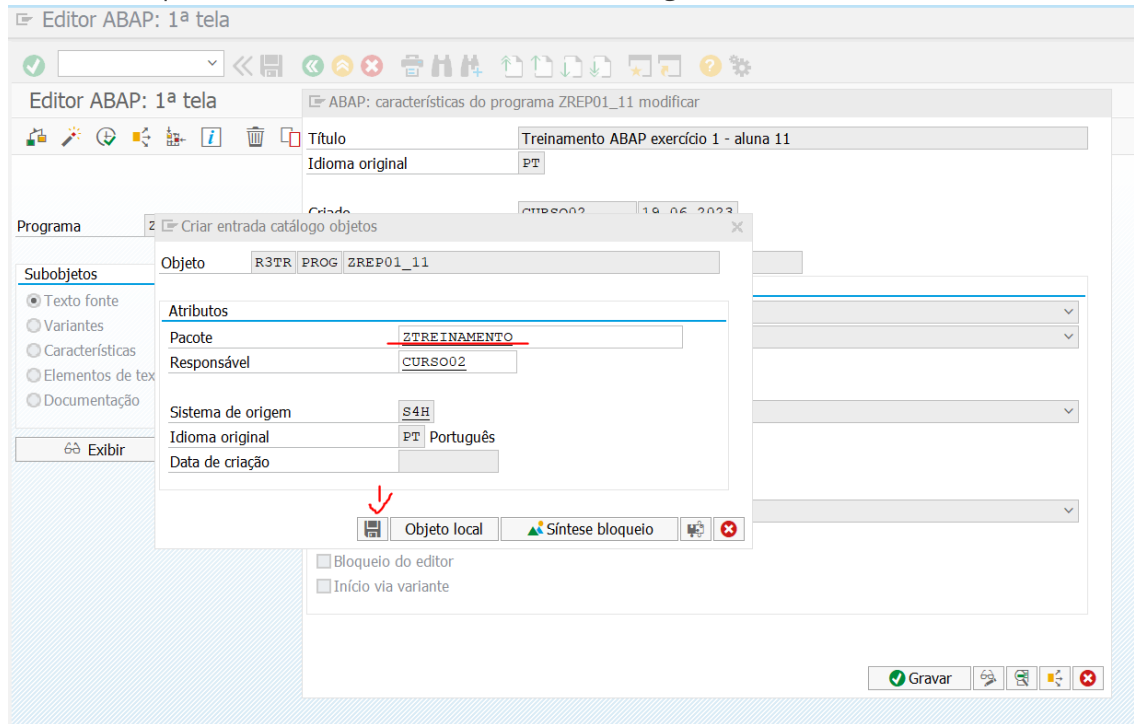


Informar nome do seu programa

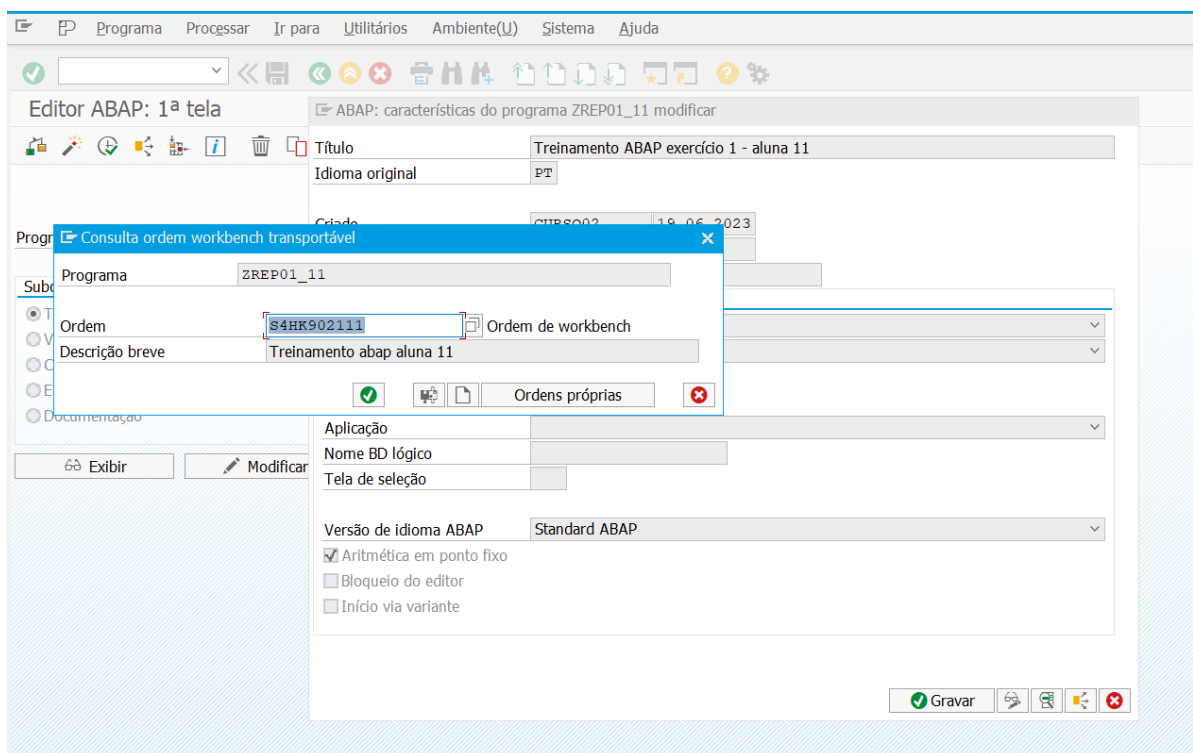
2. Em título informar um título para seu programa;
3. Em tipo selecionar a opção treinamento executavel;
4. Clicar no botão [Gravar]



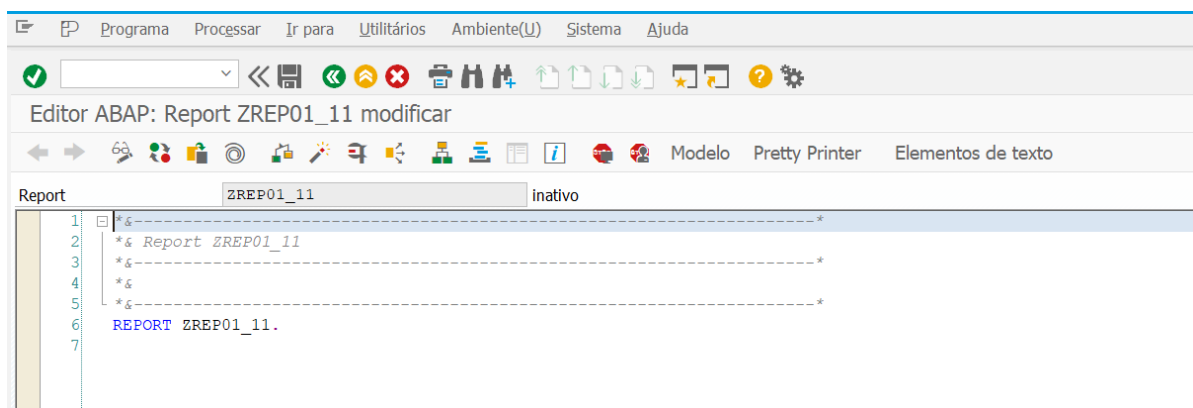
5. Preencher o pacote com ZTREINAMENTOS e clicar em gravar



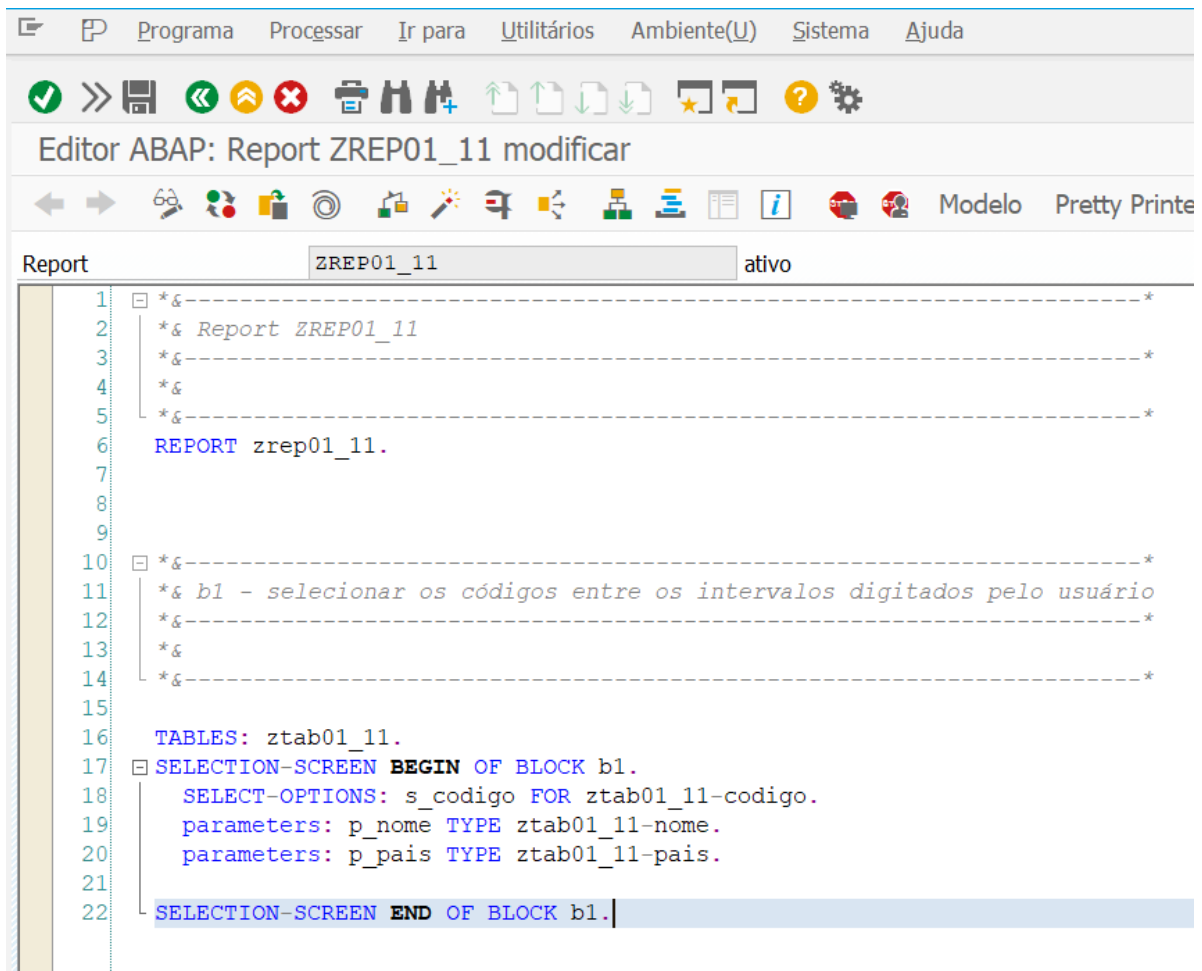
6. escolher sua request e salvar



7. o resultado esperado é:



8. Feito isso vamos preencher com o código que faz a tela desejada para o usuário



```
1  *-----*
2  *& Report ZREP01_11
3  *-----*
4  *&
5  *-----*
6  REPORT zrep01_11.
7
8
9
10 *-----*
11 *& b1 - selecionar os códigos entre os intervalos digitados pelo usuário
12 *-----*
13 *&
14 *-----*
15
16 TABLES: ztab01_11.
17 * SELECTION-SCREEN BEGIN OF BLOCK b1.
18   SELECT-OPTIONS: s_codigo FOR ztab01_11-codigo.
19   parameters: p_nome TYPE ztab01_11-nome.
20   parameters: p_pais TYPE ztab01_11-pais.
21
22 * SELECTION-SCREEN END OF BLOCK b1.
```

linha 16 - comando `Tables: zrep01_11.`: define a tabela que será utilizada para esse programa

linha 17 - `SELECTION-SCREEN BEGIN OF BLOCK b1.` esta iniciando a ação de seleção do report e nomeando o mesmo como **b1**

linha 18 - `SELECT-OPTIONS: s_codigo FOR ztab01_11-codigo.` essa linha indica que teremos um elemento do tipo random - intervalo de seleção, estamos nomeando esse elemento de `s_codigo` e esse vai fazer sua busca na tabela `ztab01_11` campo código.

linha 19 - `parameters: p_nome TYPE ztab01_11-nome.` estamos passando um parametro de retorno de dados que retornará o campo nome da tabela

linha 20 - `parameters: p_pais TYPE ztab01_11-pais.` estamos passando um parametro de retorno de dados que retornará o campo pais da tabela

linha 22 - `SELECTION-SCREEN END OF BLOCK b1.` esta finalizando a ação do SELECTION de nome **b1**

Note que todas as linhas de comando terminam com (.) ponto, essa regra deve ser seguida para não ter erros durante o desenvolvimento. Você ainda poderá usar (,) virgula, caso for chamar um elemento do mesmo tipo na linha abaixo.

Finalizada a digitação do código podemos ativar o programa, clicando no botão ativar e caso não de nenhum erro, podemos usar o botão Direto ou Executar, assim teremos o seguinte resultado:

Programa Processar Ir para Sistema Ajuda

Treinamento ABAP exercício 1 - aluna 11

S_CODIGO até

P_NOME

P_PAIS

Vamos agora tratar os rótulos do formulário

1. Voltar para tela de código e clicar no menu opção IR PARA -> opção Elementos de texto

Programa Processar Ir para Utilitários Ambiente(U) Sistema Ajuda

Editor ABAP: Report ZREP01

Report ZREP01

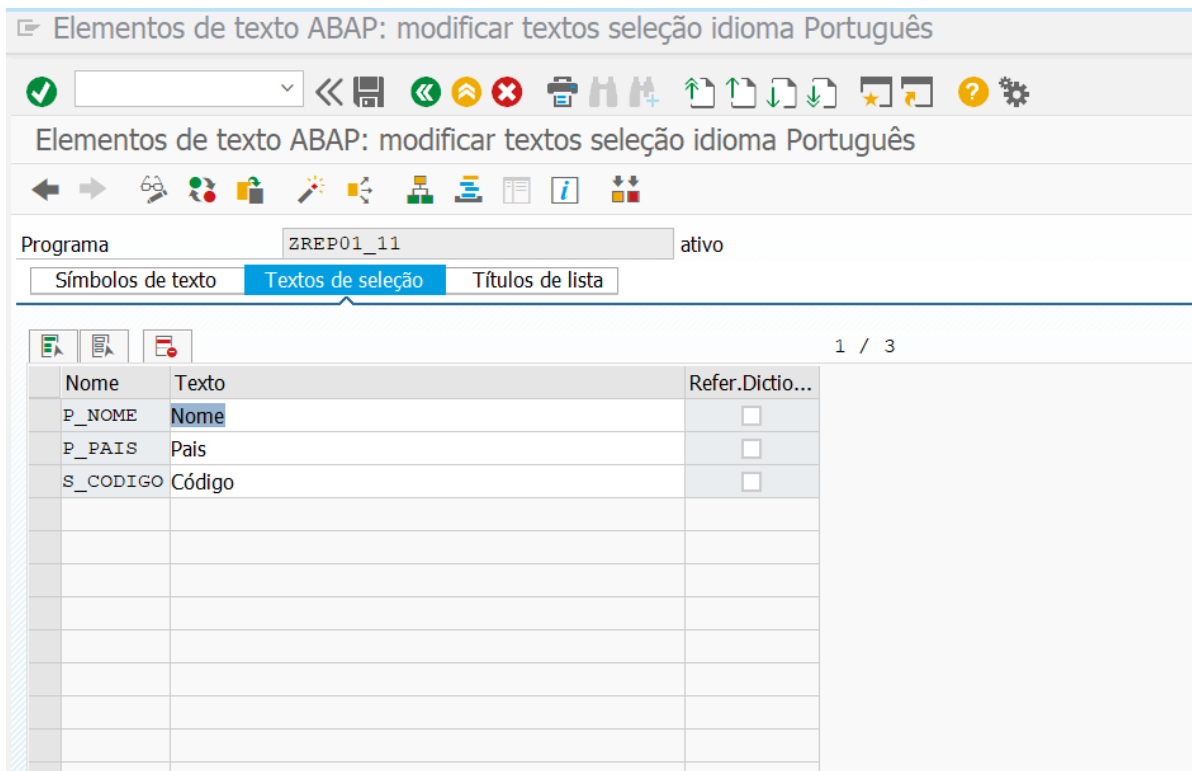
```

1  *-----*
2  *  Report ZREP01  *
3  *-----*
4  *  *
5  *-----*
6  REPORT zrep01_11.
7
8
9
10 *-----*
11 *  b1 - selecionar os códigos entre os intervalos digitados pelo usuário
12 *-----*
13 *  *
14 *-----*
15
16 TABLES: ztab01_11.
17 *-----*
18 *  SELECTION-SCREEN BEGIN OF BLOCK b1.
19 *    SELECT-OPTIONS: s_codigo FOR ztab01_11-codigo.
20 *    parameters: p_nome TYPE ztab01_11-nome.
21 *    parameters: p_pais TYPE ztab01_11-pais.
22 *  SELECTION-SCREEN END OF BLOCK b1.

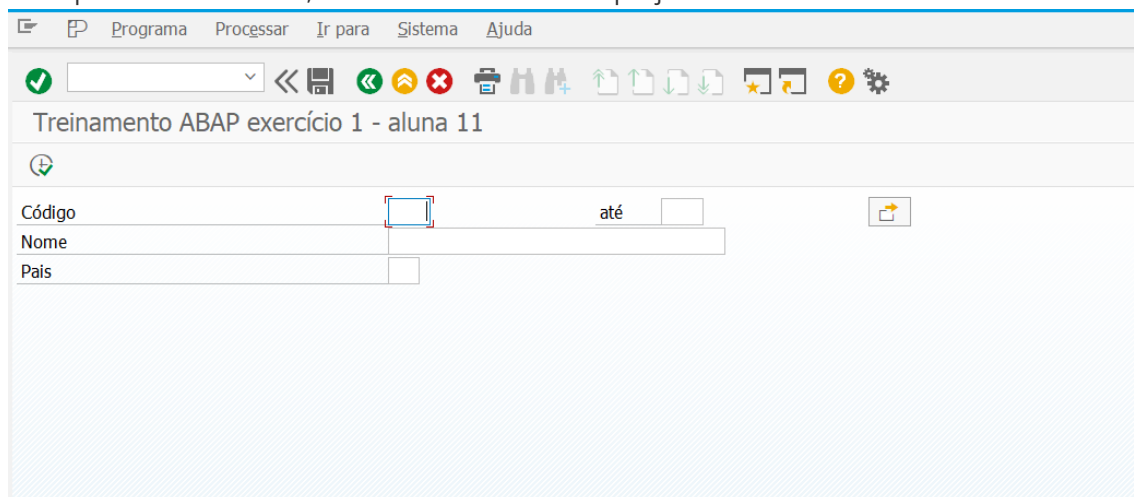
```

Modelo Pretty Printer Elementos de texto

2. Preencher o texto que deseja ter de rótulo para o nosso formulário



3. volte para a tela anterior, resultado ao executar o projeto novamente



Manutenção dos dados da tabela

Para inserir, alterar ou deletar dados da tabela customizada, ou seja, tabelas que nos criamos, vamos utilizar a transação de código SM30.

Para inserir dados na nossa tabela vamos seguir o seguinte passo a passo:

1. vamos abrir a tabela criada na operação SE11
2. vamos informar a tabela e clicar no botão modificar

Tabela Processar Ir para Utilitários Suplementos Ambiente(U) Sistema Ajuda

Dictionary: modificar tabela

Tabela transp. ZTAB01_11 ativo

Descrição breve Primeira tabela atividade

Características Entrega e atualização Campos Entr.possíveis/verificação Campos moeda/quantidade Índices

Campo	Chv	Val...	Elemento dados	CtgDados	Compr	Casas...	SistCoords.	Descrição breve
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3	0		0 Mandante
CODIGO	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL01_11	INT1	3	0		0 CODIGO
NOME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL02_11	CHAR	30	0		0 NOME
IDADE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL03_11	INT1	3	0		0 IDADE
RUA	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL04_11	CHAR	30	0		0 RUA CAMPO
NUMERO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL05_11	NUMC	5	0		0 RUA
BAIRRO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL06_11	CHAR	20	0		0 BAIRRO
CIDADE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL07_11	CHAR	30	0		0 CIDADE
ESTADO	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL08_11	CHAR	2	0		0 ESTADO
PAIS	<input type="checkbox"/>	<input checked="" type="checkbox"/>	ZEL09_11	CHAR	2	0		0 PAIS

- utilizando o menu superior vamos clicar na opção utilitários -> gerador de atualização de tabela
- na tela seguinte vamos informar em Grupo de autorizações a opção &NC& - caso o cliente já tenha um outro grupo informe conforme documentação, caso não pode utilizar esse grupo
- em grupo de funções informar o nome da tabela que esta sendo atualizada
- em tipo de atualização informar nível único

Objetos gerados Processar Ir para Ambiente(U) Utilitários Sistema Ajuda

Geração diálogo de atualização de tabelas: ambiente geração

Procurar nº(s) tela

Tabela/visão ZTAB01_11

Indicações técnicas para diálogo

Grupo autorizações &NC&

Objeto autorização S TABU NAM S TABU DIS

Grupo de funções ZTAB01_11

Pacote

Telas de atualização

Tipo de atualização ☒ Nível único ☐ 2 níveis

Nº tela atualização Tela de síntese

Tela individual

Indicações para transporte de dados de diálogo

Rotina de registro ☐ Rotina de registro standard ☒ Rotina de registro individual (ou nenhuma)

Código de ajuste Ajustável automaticamente

Nota

- agora vamos clicar no botão Procurar nº de tela e ele vai preencher o campo tela de síntese
- então vamos clicar no botão [criar] a folha em branco lado esquerdo da tela ao lado do procurar nº tela

9. Aqui teremos o nome do nosso pacote você vai salvar clicando no disquete por duas vezes
10. agora vamos salvar em nossa request

Procurar nº(s) tela

Visão

Consulta ordem workbench transportável

Grupo de funções

Ordem Ordem de workbench

Descrição breve

☒
☐
☐

☐

2 níveis

Atualização

Tela de síntese	1
Tela individual	

Opções para transporte de dados de diálogo

de registro ☐ Rotina de registro standard

☒ Rotina de registro individual (ou nenhuma)

de ajuste

Geração diálogo de atualização de tabelas: ambiente geração

Procurar nº(s) tela

Tabela/visão:

Indicações técnicas

Grupo autorizações

Objeto autorização

Grupo de funções

Pacote

Telas de atualização

Tipo de atualização

Nº tela atualização

Indicações para transação

Rotina de registro

Código de ajuste

Criar entrada catálogo objetos

Objeto:

Atributos

Pacote:

Responsável:

Sistema de origem:

Idioma original: Português

Data de criação:

☒ Rotina de registro standard
☐ Rotina de registro individual (ou nenhuma)

11. Confirme mais uma vez em salvar

12. nesse momento, já podemos abrir uma nova janela e informar a transação sm30

Declarar a tabela interna passando um nome a ela e a tabela a qual se refere, aqui teremos todos os campos da tabela Ztab01_11 copiados para a tabela interna it_ztab01_11:

```
Data: it_ztab01_11 type table of ztab01_11.
```

A tabela interna pode ter todos os campos das tabelas que ela espelha, ou podemos ainda definir quais são os campos de cada tabela que vamos utilizar aqui.

Para isso vamos declarar os TYPES, dessa forma teremos a seguinte estrutura:

```
TYPES: BEGIN OF ty_table_01,  
      codigo TYPE ztab01_11-codigo,  
      nome   TYPE ztab01_11-nome,  
      idade  TYPE ztab01_11-idade,  
END OF ty_table_01.  
  
Data: it_ztab01_11 type table of ty_table_01.
```

Work area

Também podemos criar a Work area, que é uma tabela interna que retorna um único registro, ou seja, uma única linha do banco.

Primeiro precisamos entender que um Work area é um objeto de dados estruturados de um tipo e estrutura simples, ela contém um único registro por vez. Precisamos dessa estrutura porque não é possível ler diretamente da tabela, quando executamos um SELECT os dados passam da tabela para o WORK AREA e depois para a tabela interna, podemos dizer então que é um "espaço temporário na memória do SAP e dura o tempo de execução do seu programa".

Estrutura:

```
*& --- DECLARAR WORKAREA PARA AS TABELAS *  
DATA: wa_ztab01_11 type ztab01_11,  
      wa_ztab02_11 TYPE ztab02_11.
```

Entendendo o código: waztab01_11 é o nome desse espaço em memória e type indica a tabela que vai ocupar esse espaço, no caso o ztab01_11, como temos duas tabelas em uso, podemos usar vírgula(,) para já declarar a baixo a próxima tabela, ou escrever outra linha DATA:. importante lembrar que os dois casos funciona e que escrever por blocos pode facilitar a manutenção futura do seu programa.

Nesse momento o arquivo ficou assim, note que passaremos as linhas de código da tela de seleção para baixo da tabela, tabela interna e work area:

```
&-----*  
*& Report ZREP01_11  
*&-----*  
*&  
*&-----*  
REPORT zrep01_11.  
  
*&-----*  
*& b1 - selecionar os códigos entre os intervalos digitados pelo usuário  
*&-----*
```



```

*&
*&-----*

TABLES: ztab01_11,
        ztab02_11.

*&-----*
*& TABELA INTERNA - É UMA TABELA QUE VAI ESPELHAR ALGO QUE ESTA EM TABELA - ELÁ
NÃO É UMA TABELA ASSIM COMO A ZTAB01_11*
*& A TABELA INTERNA VAI SE UTILIZADA PARA ESPERAR DADOS JUNTANDO DUAS OU MAIS
TABELAS *
*& -----*

*& TABELA INTERNA -----*
*& nome da tabela interna ----- nota da tabela que ela vai espelhar
Data: it_ztab01_11 type table of ztab01_11,
      it_ztab02_11 TYPE TABLE OF ztab02_11.

*& --- DECLARAR WORKAREA PARA AS TABELAS *
DATA: wa_ztab01_11 type ztab01_11,
      wa_ztab02_11 TYPE ztab02_11.

*& TELA DE SELEÇÃO DOS DADOS *
SELECTION-SCREEN BEGIN OF BLOCK b1.
  SELECT-OPTIONS: s_codigo FOR ztab01_11-codigo.
  parameters: p_nome TYPE ztab01_11-nome.
  parameters: p_pais TYPE ztab01_11-pais.

SELECTION-SCREEN END OF BLOCK b1.

```

ALV - ABAP List Viewer

O ABAP List Viewer (ALV) é uma estrutura fornecida pelo SAP para facilitar a exibição e manipulação de dados em formato de lista. Ele oferece recursos avançados de formatação, classificação, filtragem e interação com o usuário, tornando a exibição de dados mais eficiente e amigável.

O ALV é amplamente utilizado para criar relatórios e exibir resultados de consulta de maneira tabular. Ele permite que você personalize a aparência da lista, organize colunas, aplique cores e formate os dados de acordo com as necessidades específicas. Além disso, é possível adicionar recursos interativos, como ordenação dinâmica, cálculos automáticos, filtros personalizados, totalizações e agrupamentos.

Existem diferentes tipos de ALV no ABAP, incluindo:

- **ALV Grid:** É o tipo mais comum de ALV e permite exibir os dados em uma grade tabular. É altamente personalizável, com opções de formatação, filtragem e interação.
- **ALV Hierarchical:** Permite exibir dados hierárquicos em uma exibição semelhante a uma estrutura de árvore, mas com recursos adicionais, como expansão e contração de nós e funcionalidades de drill-down.
- **ALV List:** É um tipo mais simples de ALV que exibe os dados em formato de lista simples. É útil para listas simples de dados, sem a necessidade de recursos avançados de interação.

O ALV pode ser criado usando a biblioteca de classes do ALV, que fornece métodos e eventos para configurar e manipular a exibição dos dados. Também é possível criar ALVs usando a ferramenta de layout do SAP, o que facilita a configuração visual e a personalização da aparência do ALV.

Em resumo, o ABAP List Viewer (ALV) é uma estrutura poderosa no ABAP que facilita a exibição e a manipulação de dados em formato de lista. Ele fornece recursos avançados de formatação, filtragem e interação, permitindo a criação de relatórios e exibições de dados mais eficientes e amigáveis ao usuário.

É uma ferramenta para criar relatórios e telas para monitor.

Abap List Viewer (ALV)

Exercícios

- 1 – Utilizando as tabelas criadas nos exercícios anteriores, crie um ALV Grid para exibir os dados das tabelas.
- 2 – Faça uma cópia do programa do exercício 1 e altere para possibilitar a edição da coluna salário.
- 3 - Faça uma cópia do programa do exercício 1 e altere para colorir a coluna país.