

SAP ABAP

2.

Criar primeira tabela

1. Fazer o logon 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.visão 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)
finalizar

Descrição breve:* CODIGO

Caracts.	Ctg.dds.	Caracts.adicion.	Denomin.campo
<input checked="" type="radio"/> Categoria elemental <input checked="" type="radio"/> Domínio	ZDM01_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"/> Tipo de referência <input type="radio"/> Nome tipo refer.			
<input type="radio"/> Referência a tipo instalado	Tp.dados: Compr: 0		

Criar entrada catálogo objetos

Objeto: R3TR DTEL ZEL01_11

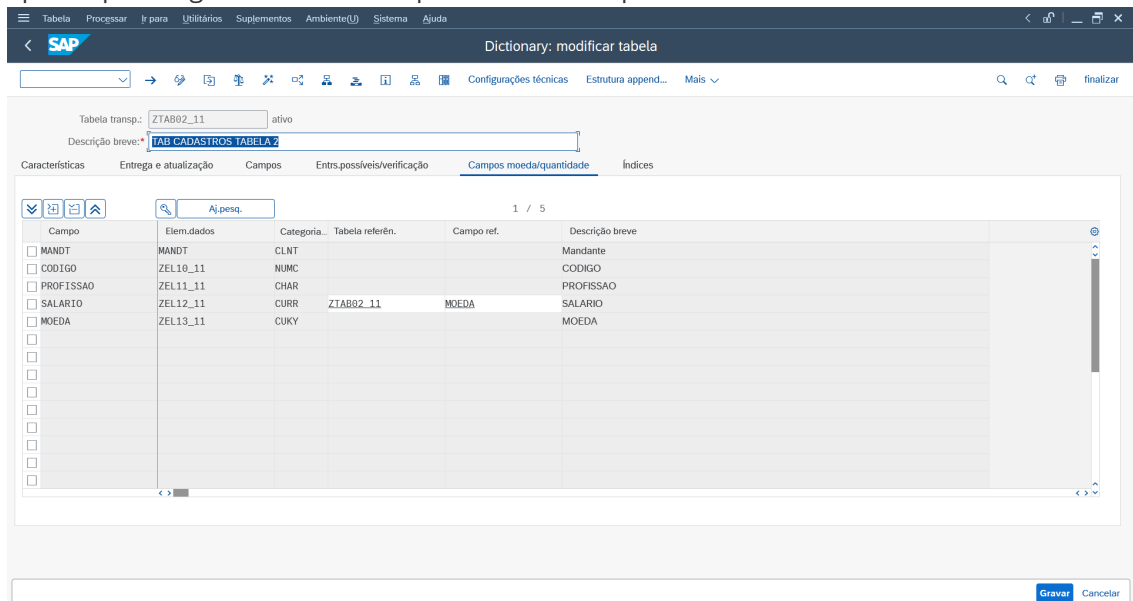
Atributos

- Pacote: ZTREINAMENTO
- Responsável: CURSOR2
- Sistema de origem: S4H
- Idioma original: PT Português
- Data de criação:

[Objeto local](#)
[Síntese bloqueio](#)

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. Após isso vamos voltar para a tela dos campos da tabela e vamos chamar esse elemento na mesma o resultado será:

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:

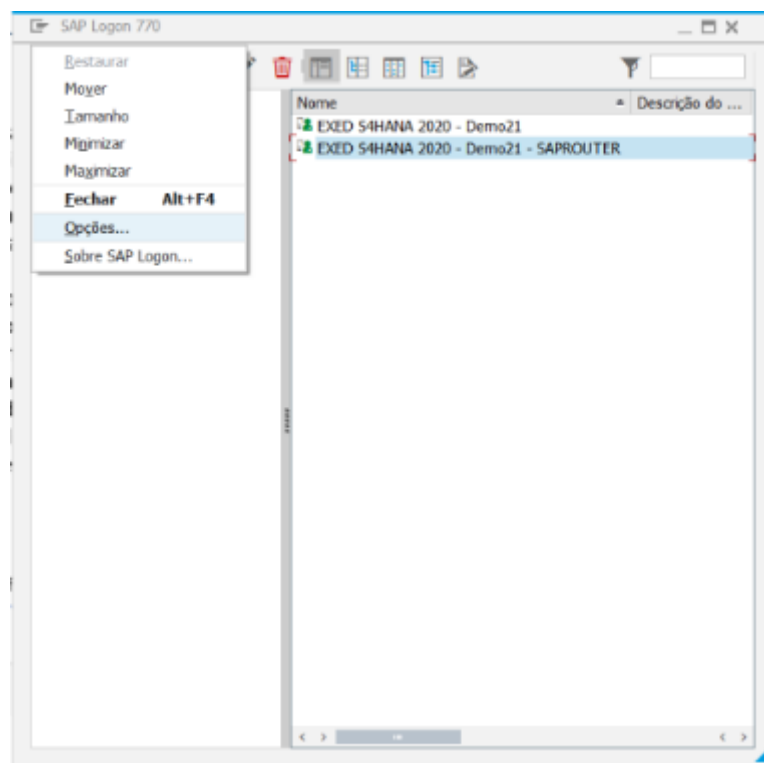


16. note na imagem acima que o campo salario esta relacionado com o campo moeda onde poderemos informar o Símbolo da moeda para o dinheiro

Trocando o tema do SAP LOGON:

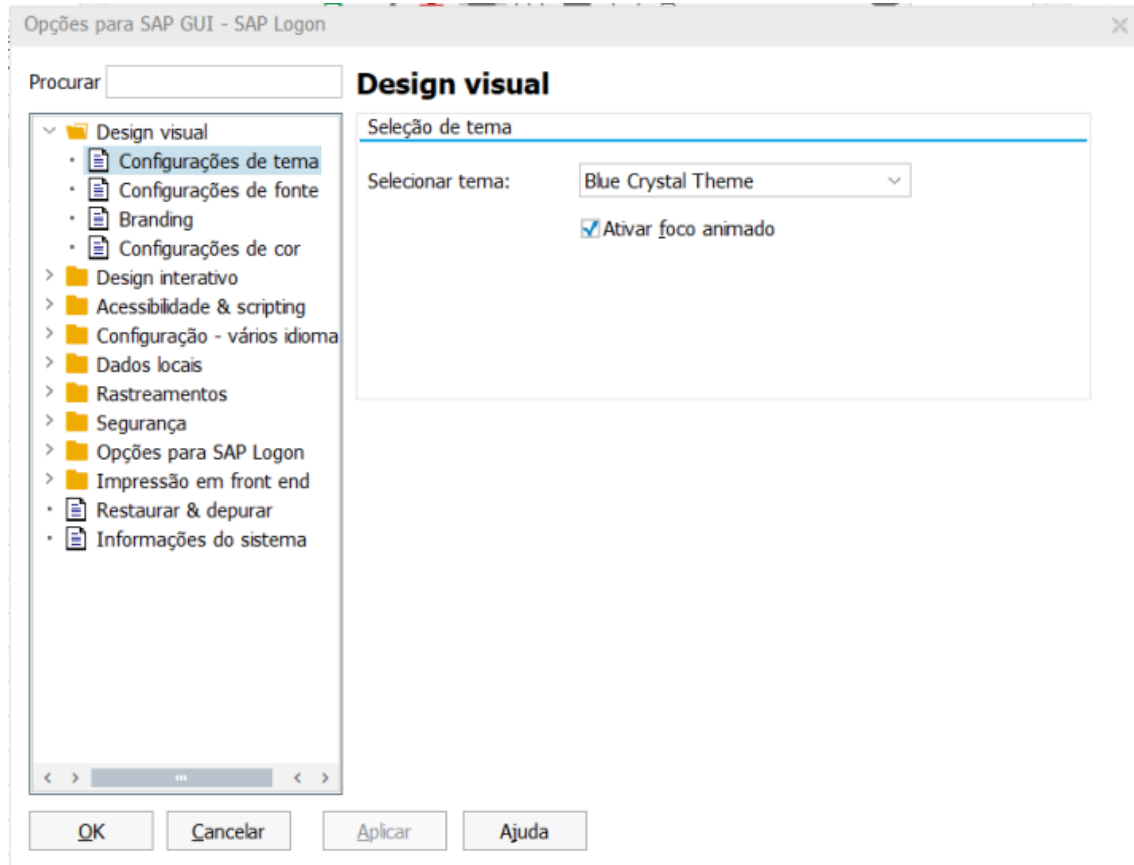
A tela do SAP LOGON dispõem de alguma opções de temas para trabalho, para alterar o tema faça o seguinte passo a passo:

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.

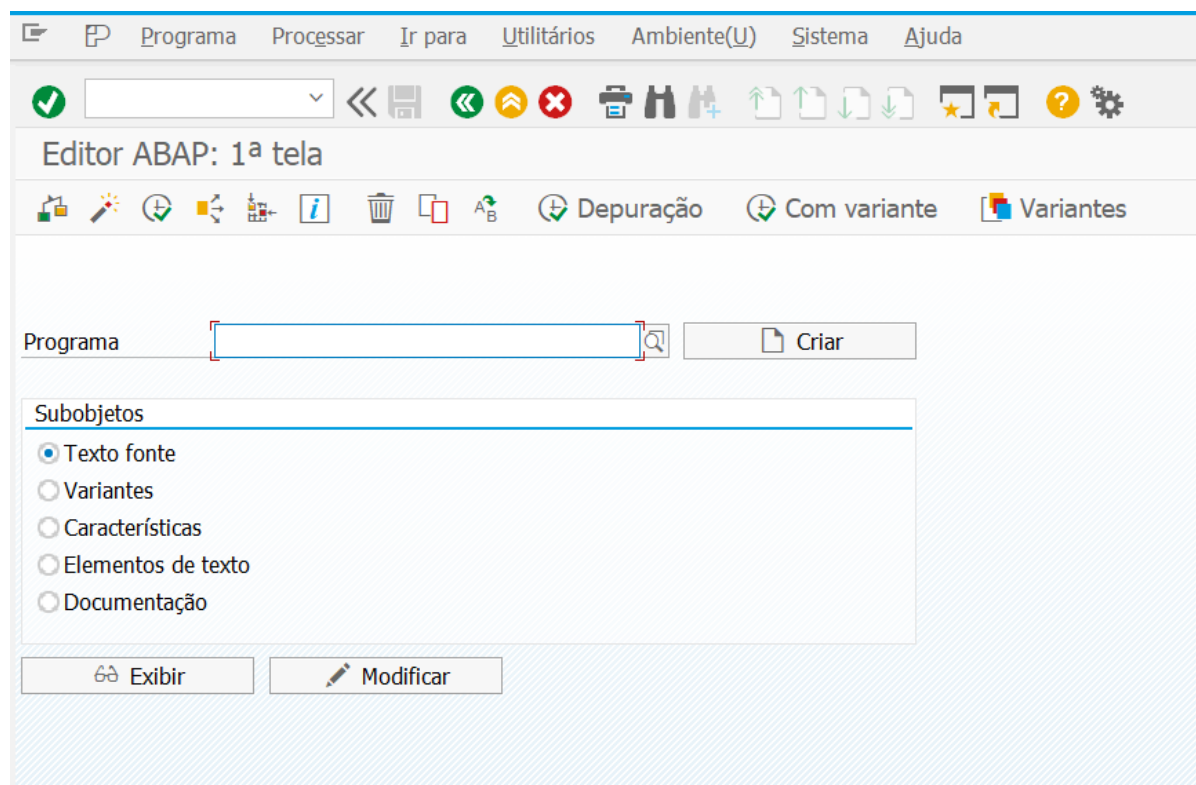
Outra dica sobre organização do código é separar as declarações por blocos e não criar tudo junto e misturado = tabelas, constantes...

- 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

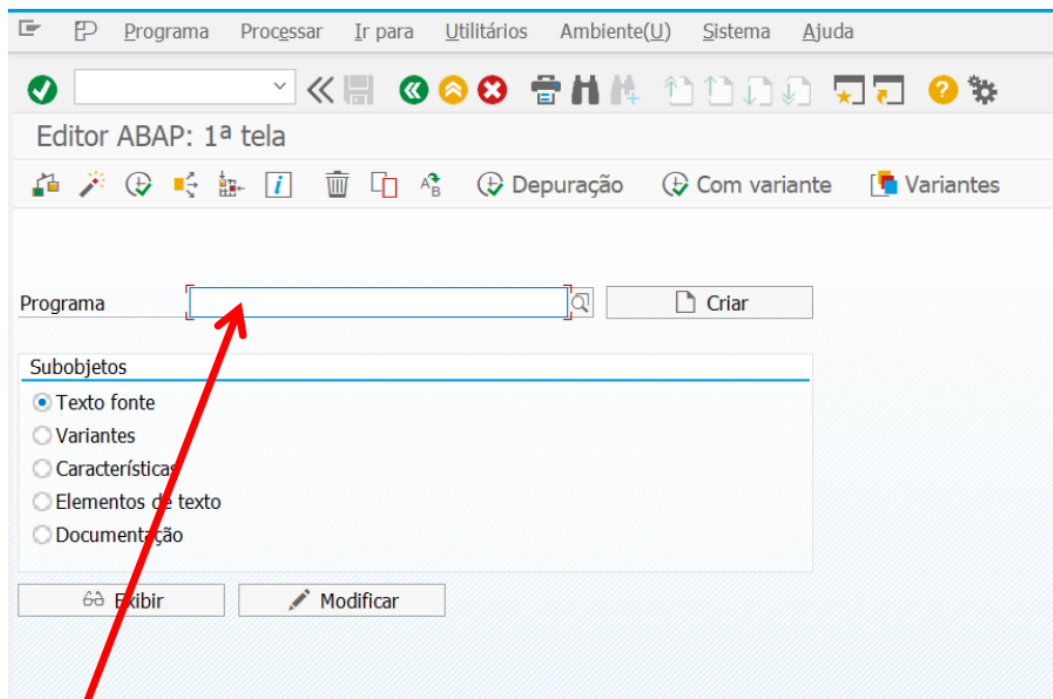
Tela de seleção:

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

Tela inicial do SE38:



Criar programa



Informar nome do seu programa

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.

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.