

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

Sumario:

Este artigo descreve a manipulação de dados do tipo de dados DATA e HORA, este, artigo descreve o tipo de dado D e T, descrevendo algumas das diversas funções disponibilizadas no R/3 para manipular os dados no sistema.

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

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

Os tipos de dados para representação de DATAS e HORAS no SAP são:

- D Tipo de dado para representação de data no formato YYYYMMDD

Onde:

YYYY representa um numeral inteiro representando ano. EX.: 2000.

MM representa um numeral inteiro representando o mês. EX.: 01 a 12

DD representa um numeral inteiro representando o dia 01 a 31.

Tamanho da variável: 8

Valor Inicial : 00000000.

- T Tipo de dado para representação de hora no formato HHMMSS

Onde:

HH representa um numeral inteiro representando a Hora: 00 até 24.

MM representa um numeral inteiro representado os minutos: 00 até 59.

SS representa um numeral inteiro representado os segundos: 00 até 59.

Tamanho da variável: 6

Valor Inicial : 000000.

Declaração:

```
DATA: {NOME_DA_VARIAVEL} TYPE D,  
      {NOME_DA_VARIAVEL} TYPE T.
```

Exe.:

```
DATA: DATA_ATUAL TYPE D,  
      HORA_ATUAL TYPE T.
```

O sistema R/3 contém diversas variáveis de sistemas para controle, entre elas as variáveis para Hora e Data:

- SY-DATUM

Variável do sistema R/3 que armazena a data atual.

- SY-UZEIT

Variável do sistema R/3 que armazena a data atual.

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

Exe.:

```
WRITE: 'DATA ATUAL: ', SY-DATUM, ' HORA ATUAL:', SY-UZEIT.
```

Cálculos

Para trabalhar com cálculos de datas o abap é uma linguagem Forte, pois ao realizar uma calculo adicionando ou subtraindo um valor numérico em uma variável do tipo data, o próprio abap trata os dados.

Exe.:

```
DATA DIA_DA_SEMANA(10) TYPE C .  
DATA: DATA_ATUAL TYPE D.
```

```
DATA_ATUAL = SY-DATUM.
```

```
WRITE DATA_ATUAL.
```

```
DATA_ATUAL = DATA_ATUAL + 20.
```

```
WRITE DATA_ATUAL.
```

```
DATA_ATUAL = DATA_ATUAL - 1.
```

```
WRITE DATA_ATUAL.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

Funções para trabalhar com datas:

- DATE_COMPUTE_DAY

Está retorna o dia da semana em formato numérico 1 à 7, a entrada é uma data no formato DDMMYYYY e a saída é uma caractere contendo o dia de 1 à 7.

Exemplo:

```
DATA_ATUAL = SY-DATUM.

CALL FUNCTION 'DATE_COMPUTE_DAY'
  EXPORTING
    DATE      = DATA_ATUAL
  IMPORTING
    DAY       = DIA_DA_SEMANA.

WRITE: / DIA_DA_SEMANA.

CASE DIA_DA_SEMANA.

  WHEN 1.
    WRITE: / 'SEGUNDA-FEIRA'.
  WHEN 2.
    WRITE: / 'TERÇA-FEIRA'.
  WHEN 3.
    WRITE: / 'QUARTA-FEIRA'.
  WHEN 4.
    WRITE: / 'QUINTA-FEIRA'.
  WHEN 5.
    WRITE: / 'SEXTA-FEIRA'.
  WHEN 6.
    WRITE: / 'SABADO'.
  WHEN 7.
    WRITE: / 'DOMINGO'.
ENDCASE.
```

Está função contém a seguinte lógica para calcular uma data:

DATA no formato numérico anomesdia MOD 7.

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

Numero = DATA mod 7

Se numero > 1

 Numero = numero -1

Senão

 Numero = numero + 6

Fim

- DATE_TO_DAY

Está função é equivalente a função DATE_COMPUTE_DAY, porém esta função retorna o dia da semana em formato TEXTO.

Entrada: Data n formato DDMMYYYY

Saída: Texto de tamanho 10.

Exe.:

```
DATA DIA_DA_SEMANA(10) TYPE C .  
DATA: DATA_ATUAL TYPE D.
```

```
DATA_ATUAL = SY-DATUM.
```

```
CALL FUNCTION 'DATE_TO_DAY'  
  EXPORTING  
    DATE      = DATA_ATUAL  
  IMPORTING  
    WEEKDAY   = DIA_DA_SEMANA.
```

```
WRITE: DIA_DA_SEMANA.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- MONTH_NAMES_GET

Está função retorna os dados:

- Numero do mês
- Nome abreviação do Mês
- Nome completo do mês

Entrada:

- Idioma

Saída:

Estrutura T247 contendo os seguintes campos;

- SPRAS = Idioma
- MNR = Numero do mês
- KTX = Nome do mês abreviado
- LTX = Nome do mês complete.

Exemplo:

```
DATA IDIOMA TYPE SY-LANGU.
```

```
IDIOMA = 'PT'.
```

```
DATA: BEGIN OF MESES OCCURS 0.
```

```
INCLUDE STRUCTURE T247.
```

```
DATA: END OF MESES .
```

```
CALL FUNCTION 'MONTH_NAMES_GET'
```

```
EXPORTING
```

```
LANGUAGE = IDIOMA
```

```
TABLES
```

```
MONTH_NAMES = MESES
```

```
EXCEPTIONS
```

```
MONTH_NAMES_NOT_FOUND = 1
```

```
OTHERS = 2.
```

```
IF SY-SUBRC = 0.
```

```
LOOP AT MESES.
```

```
WRITE: / MESES-MNR, ' ', MESES-LTX.
```

```
ENDLOOP.
```

```
ENDIF.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- FKK_LAST_DAY_OF_MONTH

Esta função retorna o último dia de um mês, este retorno é baseado sobre a data de entrada, por exemplo qual o último dia do mês de janeiro. A entrada é uma data no formato DDMMYYYY e a saída é DDMMYYYY. Esta função pode ser utilizada para realizar um cálculo por exemplo, quero descobrir quantos dias falta para terminar o mês a partir de hoje.

Exemplo:

```
DATA: HOJE TYPE SY-DATUM,  
      ULTIMO TYPE SY-DATUM,  
      DIAS TYPE I.  
  
HOJE = SY-DATUM.  
  
CALL FUNCTION 'FKK_LAST_DAY_OF_MONTH'  
  EXPORTING  
    DAY_IN      = HOJE  
  IMPORTING  
    LAST_DAY_OF_MONTH = ULTIMO  
  EXCEPTIONS  
    DAY_IN_NO_DATE    = 1  
    OTHERS             = 2.  
  
IF SY-SUBRC = 0.  
  
  DIAS = ULTIMO - HOJE.  
  
  WRITE: / 'HOJE:   ', HOJE,  
         / 'ULTIMO DIA: ', ULTIMO,  
         / 'FALTAM:   ', DIAS.  
  
ENDIF.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- HR_AUPBS_MONTH_DAY

Esta função é muito interessante, pois ela realiza cálculos entre duas datas, os cálculos de retornos são dados de quantidade de dias entre data inicial, e final, quantidade de mês(es) entre a data inicial e final e quantidade de anos entre a data inicial e a data final, e a quantidade total de dias entre a data total e a data final.

Exemplo:

DATA INICIAL = 12022008
DATA FINAL = 18022008

Diferenças entre dia(s):	18 - 12	=	7
Diferenças entre mês(es):	02 - 02	=	0
Diferenças entre ano(s):	2008 - 2008	=	0
Total de dias:	7		

- Entrada

Data inicial no formato DDMMYYYY
Data final no formato DDMMYYYY

- Saída

NO_DAY	=	Numero de dia(s)
NO_MONTH	=	Numero de mês(es)
NO_YEAR	=	Numero de anos
NO_CAL_DAY	=	Numero total de dias.

Exemplo:

```
DATA: DATA_INICIAL TYPE D,  
      DATA_FINAL TYPE D,  
      NR_DIA TYPE I,  
      NR_MES TYPE I,  
      NR_ANO TYPE I,  
      NR_DAT_TOT TYPE I.
```

```
DATA_INICIAL = SY-DATUM.  
DATA_FINAL = SY-DATUM + 500.
```


TRABALHANDO COM DATAS ABAP UDERSON LUIS FERMINO

```
CALL FUNCTION 'HR_AUPBS_MONTH_DAY'  
EXPORTING  
  BEG_DA      = DATA_INICIAL  
  END_DA      = DATA_FINAL  
IMPORTING  
  NO_DAY      = NR_DIA  
  NO_MONTH    = NR_MES  
  NO_YEAR     = NR_ANO  
  NO_CAL_DAY  = NR_DAT_TOT.  
  
WRITE: 'DATA INICIAL: ', DATA_INICIAL, ' DATA FINAL: ',  
DATA_FINAL.  
  
WRITE: /'DIA(S): ', NR_DIA,  
       /'MÊS(SES): ', NR_MES,  
       /'ANO(S): ', NR_ANO,  
       /'DIA TOTAL: ', NR_DAT_TOT.
```

- DAY_IN_WEEK

Está função retorna o dia da semana, em formato numérico de 1 a até 7,
onde:

1	=	Domingo
2	=	Segunda-Feira
3	=	Terça-Feira
4	=	Quarta-Feira
5	=	Quinta-Feira
6	=	Sexta-Feira
7	=	Sabado

Entrada:

Data no formato DD.MM.YYYY

Exemplo:

```
DATA: DATA TYPE SY-DATUM,  
      DIA TYPE P.  
  
CALL FUNCTION 'DAY_IN_WEEK'  
EXPORTING  
  DATUM      = DATA  
IMPORTING  
  WOTNR      = DIA.  
  
WRITE: 'DIA DA SEMANA: ', DIA.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- DATE_GET_WEEK

Está função é muito interessante ela retorna o numero da semana atual de uma determinada data, o formato da saída é:

YYYYXX onde YYYY é equivalente o ano da data e XX é o numero da semana.

- Entrada:

Data no formato DDMMYYYY.

- Saída:

YYYYXX onde YYYY é equivalente o ano da data e XX é o numero da semana.

Exemplo:

```
DATA: DATA TYPE SY-DATUM,  
      SEMANA TYPE SCAL-WEEK.
```

```
DATA = SY-DATUM.
```

```
CALL FUNCTION 'DATE_GET_WEEK'  
  EXPORTING  
    DATE      = DATA  
  IMPORTING  
    WEEK      = SEMANA  
  EXCEPTIONS  
    DATE_INVALID = 1  
    OTHERS      = 2.
```

```
WRITE: 'SEMANA', SEMANA+4.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- HRIQ_WEEK_GET_FIRST_DAY

Está função é parceira da função DATE_GET_WEEK, pois com uma determinada semana esta função retorna o primeiro dia em formato data relativo àquela semana.

- Entrada:

Data no formato DDMMYYYY.

- Saída:

Exemplo:

```
DATA: DATA TYPE SY-DATUM,  
      SEMANA TYPE SCAL-WEEK.
```

```
DATA = SY-DATUM.
```

```
CALL FUNCTION 'DATE_GET_WEEK'
```

```
EXPORTING
```

```
  DATE      = DATA
```

```
IMPORTING
```

```
  WEEK      = SEMANA
```

```
EXCEPTIONS
```

```
  DATE_INVALID    = 1
```

```
  OTHERS          = 2.
```

```
WRITE: 'SEMANA', SEMANA+4.
```

```
CALL FUNCTION 'HRIQ_WEEK_GET_FIRST_DAY'
```

```
EXPORTING
```

```
  WEEK      = SEMANA
```

```
IMPORTING
```

```
  DATE      = DATA
```

```
EXCEPTIONS
```

```
  WEEK_INVALID    = 1
```

```
  OTHERS          = 2.
```

```
WRITE: / 'PRIMEIRO DIA DA SEMANA', DATA+6(2).
```

Está função é idêntica a WEEK_GET_FIRST_DAY, o ideal é utilizar esta ao invés da HRIQ_WEEK_GET_FIRST_DAY.

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- /OSP/GET_DAYS_IN_MONTH

A partir de uma data esta função retorna o numero de dias total relativo aquele mês, esta função é muito importante para descobrir os dias totais de fevereiro, pois este mês tem uma variação de 28 e 29 dependendo d ano.

- Entrada:

Data no formato DDMMYYYY.

- Saída:

Numérico: 28 ou 29, ou 30 ou 31.

Exemplo:

```
DATA: HOJE TYPE SY-DATUM,  
      DIA TYPE INT1.
```

```
HOJE = SY-DATUM.
```

```
CALL FUNCTION '/OSP/GET_DAYS_IN_MONTH'  
  EXPORTING  
    IV_DATE = HOJE  
  IMPORTING  
    EV_DAYS = DIA.
```

```
WRITE: 'ULTIMO DIA: ', DIA.
```

- CACS_DATE_GET_YEAR_MONTH

Esta função é apenas uma função que divide uma data em mês e anos, a entrada é a data no formato DD.MM.YYYY é o retorno é as quebra em suas variáveis contendo os dados de mês e anos.

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- SD_DATETIME_DIFFERENCE

Está função retorna a diferença de dias entre duas datas, entre a data inicial e a data final.

- Entrada:

Data inicial no formato DDMMYYYY.
Data final no formato DDMMYYYY.

Hora inicial no formato HH:MM:SS.
Hora final no formato HH:MM:SS.

- Saída:

Diferença entre dias do tipo P
Diferença entre horas do tipo P

Exemplo:

```
DATA: DATA1 TYPE SY-DATUM,  
      DATA2 TYPE SY-DATUM,  
      HORA1 TYPE SY-UZEIT,  
      HORA2 TYPE SY-UZEIT,  
      DIFD TYPE P,  
      DIFH TYPE P.
```

```
DATA1 = SY-DATUM.  
DATA2 = SY-DATUM + 10.  
HORA1 = SY-UZEIT.  
HORA2 = SY-UZEIT + 2.
```

```
CALL FUNCTION 'SD_DATETIME_DIFFERENCE'  
EXPORTING  
  DATE1          = DATA1  
  TIME1          = HORA1  
  DATE2          = DATA2  
  TIME2          = HORA2  
IMPORTING  
  DATEDIFF      = DIFD  
  TIMEDIFF      = DIFH  
EXCEPTIONS  
  INVALID_DATETIME = 1  
  OTHERS         = 2.
```

```
WRITE: 'DIFERENÇA DIAS: ', DIFD, ' DIFERENÇA HORAS: ', DIFH.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- RP_LAST_DAY_OF_MONTHS

Está função é muito importante, pois ela é comparada com a função /OSP/GET_DAYS_IN_MONTH, porém além de retornar o ultimo dia ele retorna a data completa, a entrada é uma data no formato DD.MM.YYYY e a saída é DD.MM.YYYY, contendo o ultimo dia do mês de uma determinada data.

Exemplo:

```
DATA: DATA1  TYPE SY-DATUM,  
      DATA2  TYPE SY-DATUM.
```

```
DATA1 = SY-DATUM.
```

```
CALL FUNCTION 'RP_LAST_DAY_OF_MONTHS'  
EXPORTING  
  DAY_IN          = DATA1  
IMPORTING  
  LAST_DAY_OF_MONTH = DATA2  
EXCEPTIONS  
  DAY_IN_NO_DATE    = 1  
  OTHERS            = 2.
```

```
WRITE: 'DATA: ', DATA1, ' ULTIMO DIA: ', DATA2.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- RP_CALC_DATE_IN_INTERVAL

Está função é uma das funções mais importantes para cálculos com datas, os cálculos serão realizados por esta função, ela é responsável por realizar os cálculos sobre uma data ficando transparente ao programador ABAP como os tipos de dado serão transportados, por exemplo, para realizar um calculo da data de hoje até uma data determinada, por exemplo.

Data atual 16022008 + 20+ 2 meses = 07052008.

Está função pode incrementar dias, meses e anos sobre uma data, também pode decrementar dias, meses e anos.

- DATE = Data que será realizado os cálculos
- DAYS = Total de dias a ser acrescentado ou decrementado.
- MONTHS = Total de mês(es) dias a ser acrescentado ou decrementado.
- SIGNUM = Operador de de calculo

+ Incremento
- Decremetno

- YEARS = Total de anos a ser acrescentado ou decrementado.

Saída:

Data no formato DD.MM.YYYYYY.

Exemplo:

```
DATA: DATAC TYPE SY-DATUM,  
      DATAF TYPE SY-DATUM,  
      DAYS TYPE N VALUE 20 LENGTH 2,  
      MONTHS TYPE N VALUE 2 LENGTH 2,  
      YEARS TYPE N VALUE 1 LENGTH 2.
```

```
      DATAC = SY-DATUM.
```

```
CALL FUNCTION 'RP_CALC_DATE_IN_INTERVAL'
```

```
EXPORTING
```

```
  DATE      = DATAC
```

```
  DAYS      = DAYS
```

```
  MONTHS    = MONTHS
```

```
  SIGNUM    = '+'
```

```
  YEARS     = YEARS
```

```
IMPORTING
```

```
  CALC_DATE = DATAF.
```

```
WRITE: DATAF.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- L_MC_TIME_DIFFERENCE

Está função retorna os minutos de ente duas datas, estas datas podem ser passadas juntamente de horas e é retornado os minutos de entre a data inicial e a data final.

Exemplo um dia tem 24 horas então um dia tem 1.440 minutos pos cada hora contém 60 minutos e um dia tem 24 horas então $24 * 60 = 1.440$.

Entrada:

- Data Inicial DD.MM.YYYY
- Data Final DD.MM.YYYY
- Hora Inicial DD.MM.YYYY
- Hora Final DD.MM.YYYY

Exemplo:

```
DATA: DATAFROM TYPE SY-DATUM,  
      DATATO TYPE SY-DATUM,  
      TIMEFROM TYPE SY-UZEIT,  
      TIMETO TYPE SY-UZEIT,  
      DELTATIME TYPE P LENGTH 8.
```

```
DATAFROM = SY-DATUM.
```

```
DATATO = SY-DATUM + 1.
```

```
CALL FUNCTION 'L_MC_TIME_DIFFERENCE'  
  EXPORTING  
    DATE_FROM      = DATAFROM  
    DATE_TO        = DATATO  
    TIME_FROM      = TIMEFROM  
    TIME_TO        = TIMETO  
  IMPORTING  
    DELTA_TIME     = DELTATIME  
  EXCEPTIONS  
    FROM_GREATER_TO = 1  
    OTHERS         = 2.
```

```
WRITE: DELTATIME.
```


TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- DATE_CONVERT_TO_FACTORYDATE

Está função converte uma determinada data em um factorydate, este factorydate é iniciado de 0 valendo 01.01.1996, e assim por diante, cada data apartir de 01.01.1996 será gerando um factorydate para cada data,

- Entrada:

- CORRECT_OPTION = Signos
- DATE = Data de geração
- FACTORY_CALENDAR_ID = Tipo do calendário

- Saída:

- FACTORYDATE = Numeração da data

Exemplo:

```
PARAMETERS: DATE_1 LIKE SY-DATUM DEFAULT SY-DATUM,  
             BIZ_DAYS TYPE I DEFAULT 2,  
             FACCALID LIKE SCAL-FCALID DEFAULT 'US'.
```

```
DATA:   FAC_DATE_1 LIKE SCAL-FACDATE,  
        DATE_2 LIKE SY-DATUM,  
        FAC_DATE_2 LIKE SCAL-FACDATE.
```

```
CALL FUNCTION 'DATE_CONVERT_TO_FACTORYDATE'  
  EXPORTING  
    DATE           = DATE_1  
    FACTORY_CALENDAR_ID = FACCALID  
  IMPORTING  
    FACTORYDATE = FAC_DATE_1 .  
  
FAC_DATE_2 = ABS( FAC_DATE_1 + BIZ_DAYS ).
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- FACTORYDATE_CONVERT_TO_DATE

Está função converte um determinado numero factorydate em uma data. É o inverso da função DATE_CONVERT_TO_FACTORYDATE.

FACTORYDATE	=	Numero Factory
FACTORY_CALENDAR_ID	=	Tipo do calendario.

Exemplo.:

```
CALL FUNCTION 'FACTORYDATE_CONVERT_TO_DATE'
EXPORTING
  FACTORYDATE           = FAC_DATE_2
  FACTORY_CALENDAR_ID   = FACCALID
IMPORTING
  DATE                  = DATE_2

WRITE: / DATE_2.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- DATE_CHECK_WORKINGDAY

Esta função retorna se a data é um dia de trabalho “SEGUNDA-FEIRA” até “SEXTA-FEIRA”, caso a data seja SABADO ou DOMINGO será retornado uma erro.

Entrada:

- DATA = Para verificação
- ID do calendário = O id do calendário definirá os dias de trabalho e dias de descanso.
- Tipo de Mensagem
 - E = erro

Exemplo:

```
DATA DATA TYPE SY-DATUM.  
DATA = SY-DATUM - 4.
```

```
CALL FUNCTION 'DATE_CHECK_WORKINGDAY'  
EXPORTING  
  DATE = DATA  
  FACTORY_CALENDAR_ID = '01'  
  MESSAGE_TYPE = 'E'  
EXCEPTIONS  
  DATE_AFTER_RANGE = 1  
  DATE_BEFORE_RANGE = 2  
  DATE_INVALID = 3  
  DATE_NO_WORKINGDAY = 4  
  FACTORY_CALENDAR_NOT_FOUND = 5  
  MESSAGE_TYPE_INVALID = 6  
  OTHERS = 7.
```

```
IF SY-SUBRC = 0.
```

```
  WRITE: DATA, 'ESTE DIA É UM DIA DE TRABALHO'.
```

```
ELSE.
```

```
  WRITE: DATA, 'ESTE DIA NÃO É UM DIA DE TRABALHO'.
```

```
ENDIF.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- FIMA_DAYS_AND_MONTHS_AND_YEARS

Está função, retorna a diferença entre duas datas, a diferença retornada em:

- Quantidade de dias
 - Quantidade de mês
 - Quantidade de anos
- Entrada:
 - Data inicial
 - Data final
 - Saída
 - Dias
 - Mês
 - Anos

Exemplo:

```
DATA: DATAFROM TYPE SY-DATUM,  
      DATATO TYPE SY-DATUM,  
      DIA TYPE I,  
      MES TYPE I,  
      ANO TYPE I.
```

```
DATAFROM = SY-DATUM.  
DATATO = SY-DATUM + 100.
```

```
CALL FUNCTION 'FIMA_DAYS_AND_MONTHS_AND_YEARS'  
  EXPORTING  
    I_DATE_FROM      = DATAFROM  
    I_KEY_DAY_FROM   = '00'  
    I_DATE_TO        = DATATO  
    I_KEY_DAY_TO     = '00'  
    I_FLG_SEPARATE   = ''  
  IMPORTING  
    E_DAYS           = DIA  
    E_MONTHS         = MES  
    E_YEARS          = ANO.  
  
WRITE: / 'DIA: ',  
      DIA,  
      / 'MES: ',  
      MES,  
      / 'ANO: ',  
      ANO.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

- SLS_MISC_GET_LAST_DAY_OF_MONTH

Está função é mais uma das funções que retornam o ultimo dia de um determinado mês, a entrada é uma data no formato DD.MM.YYYY, pois é necessário informar a data completa pois cada data pode e variar o ultimo dia.

- Entrada:
 - Data no formato DD.MM.YYY
- Saída:
 - Data no formato DD.MM.YYYY

Exemplo:

```
DATA: DATAI TYPE SY-DATUM,  
      DATAF TYPE SY-DATUM.
```

```
DATAI = SY-DATUM.
```

```
CALL FUNCTION 'SLS_MISC_GET_LAST_DAY_OF_MONTH'  
  EXPORTING  
    DAY_IN          = DATAI  
  IMPORTING  
    LAST_DAY_OF_MONTH = DATAF  
  EXCEPTIONS  
    DAY_IN_NOT_VALID  = 1  
    OTHERS             = 2.
```

```
WRITE: DATAF.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

DATE_CREATE

Está função criar uma determinada data a partir de outra data, podemos dizer que está função calcula uma determinada data acrescentando dias, meses e anos.

- Entrada:
 - ANZAHL_JAHRE = Quantidades de Anos
 - ANZAHL_KALTAGE = Quantidades de Dias
 - ANZAHL_MONATE = Quantidades de Meses
 - DATUM_EIN = Data atual
- Saída
 - DATUM_AUS = Data criada (calculada)

Exemplo:

```
DATA: DATAI TYPE SY-DATUM,  
      DATAF TYPE SY-DATUM.
```

```
DATAI = SY-DATUM.
```

```
CALL FUNCTION 'DATE_CREATE'  
EXPORTING  
  ANZAHL_JAHRE = 12  
  ANZAHL_KALTAGE = 1  
  ANZAHL_MONATE = 1  
  DATUM_EIN = DATAI  
IMPORTING  
  DATUM_AUS = DATAF.
```

```
WRITE: 'DATA ATUAL: ', DATAI, ' DATA CALCULADA: ', DATAF.
```

TRABALHANDO COM DATAS ABAP

UDERSON LUIS FERMINO

Parameter com tipo DATA.

Quando trabalhamos com parameter do tipo data, o próprio sistema R/3 cria um objeto MATCH CODE automaticamente, este objeto criar um calendário para escolha de dado.

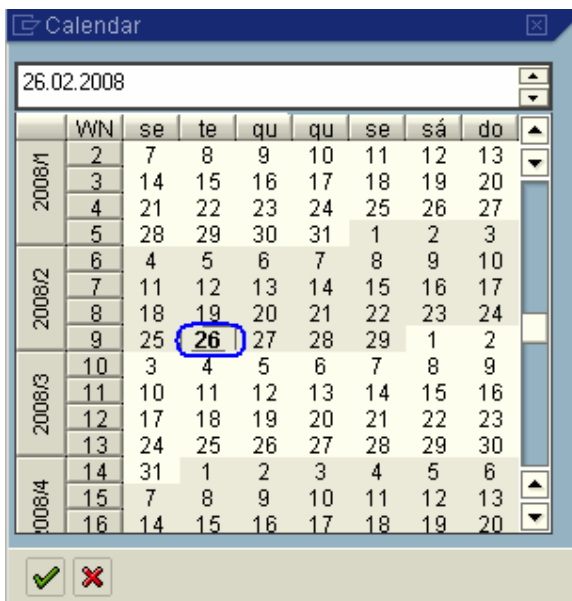
Exemplo.:

PARAMETERS: P_START TYPE SY-DATUM
P_END TYPE SY-DATUM.

Saída:



O calendário



O calendário é carregado na data atual podendo ser alterado a escolha do usuário,