


Custo em Partes - Recálculo do Custo Médio

 Tempo aproximado para leitura: superior a 15 minutos

Custo em Partes - Recálculo do Custo Médio

Descrição dos procedimentos para utilização do custo em partes com e sem *stored procedures*.

O recálculo do custo médio possibilita dividir o custo de produtos fabricados em mais de uma parte, facilitando a visualização da composição de custos dos produtos acabados.

O sistema permite dividir o custo de produtos fabricados em até 99 partes diferentes, cada parte nas 5 moedas padrão do sistema.

O usuário deve através do ponto de entrada "MA330CP" definir as regras que irão classificar cada matéria-prima em uma parte do custo. O número de partes é sempre acrescido de mais uma parte, a qual contempla os materiais que não se encontram em nenhuma regra.

Importante!

O Custo em partes é basicamente o desmenbramento da composição do custo médio de um produto produzido, facilitando a visualização dos custos de cada parte utilizada na produção e não esta relacionado com o custo de reposição de um produto, que envolve o custo de compra ou aquisição para reposição do estoque.

Utilização do Custo em Partes somente em ADVPL

Para definir a regra de utilização do custo em partes, deve-se utilizar o ponto de entrada MA330CP. Abaixo segue um exemplo do ponto de entrada.

Ponto de entrada MA330CP (desenvolvido em linguagem AdvPL):

MA330CP

```
#include "RWMAKE.CH"

User Function MA330CP ()

LOCAL aRegraCP:={}

AADD(aRegraCP,"SB1->B1_TIPO == 'MP'")
AADD(aRegraCP,"SB1->B1_TIPO == 'MC'")

Return aRegraCP
```

Utilizando o exemplo descrito, é possível verificar o custo dividido em três partes:

- A parte 1 composta dos valores de produtos que tenham o campo B1_TIPO = "MP"
- A parte 2 composta dos valores de produtos que tenham o campo B1_TIPO = "MC"
- A parte 3 composta dos valores de produtos que não se enquadram nas partes citadas anteriormente

Para utilizar o custo em partes devem ser criados alguns campos no sistema seguindo as regras e nomes do quadro abaixo. Se o ponto de entrada que define as regras do custo existir e os campos não forem criados, a rotina de recálculo do custo processará o recálculo do custo **desconsiderando** o custo em partes.

Os campos devem ser definidos utilizando a seguinte regra:

CC – Numero do Custo Em Partes

MM – Numero da Moeda (1 a 5)


TABELA	CAMPOS
--------	--------

SB9 – Saldos Iniciais	B9_CP + CC + MM (Custo em partes) – B9_VINI1 B9_CPM + CC + MM (Custo em partes Unitário) – B9_CM1
SB2/TR2XXSP – Saldos em Estoque	B2_CP + CC + MM (Custo em partes Unitário) – B2_CM1 B2_CPF + CC + MM (Custo em partes Final) – B2_VFIM1
SB6 – Saldos em poder de terceiros	B6_CP + CC + MM (Custo em partes) – B6_CUSTO1
SC2 – Ordens de Produção	C2_CPF + CC + MM (Custo em partes Final) – C2_VFIM1 C2_CPI + CC + MM (Custo em partes Inicial) – C2_VINI1 C2_API + CC + MM (Apropriação Inicial do Custo em partes) – C2_APRINI1 C2_APF + CC + MM (Apropriação Final do Custo em partes) – C2_APRFIM1
SD1 – Itens das NFs de Entrada	D1_CP + CC + MM (Custo em partes) – D1_CUSTO
SD2 – Itens das NFs de Saída	D2_CP + CC + MM (Custo em partes) – D2_CUSTO1
SD3 – Movimentos Internos	D3_CP+ CC + MM (Custo em partes) – D2_CUSTO1

Para processar o custo em partes em ADVPL, as *procedures* padrões não devem estar instaladas. Exemplo: se o processo 17 (Virada de Saldos) estiver instalado e as *procedures* do Custo em Partes não existirem no banco de dados, o sistema não atualizará as informações do custo em partes nas tabelas envolvidas.

Utilização do Custo em Partes com Stored Procedures

A utilização do custo em partes com *stored procedures* é mais complexa que a utilização em ADVPL, pois, envolve a linguagem de programação SQL. Para definir a regra do custo em partes, é necessário utilizar o ponto de entrada MA330CP em ADVPL. Existindo o ponto de entrada MA330CP em ADVPL, a instalação de procedures irá instalar algumas procedures adicionais básicas essenciais ao processo, porém sem tratamentos lógicos devido ao número de partes não ser fixo, podendo variar para cada cliente. Estas procedures adicionais são M330INB2CP, M330INC2CP para o recálculo do custo médio e MA280INB9CP, MA280INC2CP para o fechamento e seus nomes devem ser mantidos. Por exemplo M330INB2CP_xx, onde xx é o código da empresa. Veja um **exemplo** de todo o procedimento a ser implementado para o correto funcionamento do custo em partes com *stored procedures*.


Importante!
Caso o ambiente possua o parâmetro MV_A330SB2 configurado como .T., ou release igual ou superior à 12.1.2310, é necessário que as queries onde é feita leitura/atualização da SB2, nas procedures **M330INB2CP**, **M330INC2CP** e **MA330CP**, sejam alteradas para acessar a tabela TR2XXSP, onde XX é o código do grupo de empresas.

Para mais informações, veja: [Recálculo do Custo Médio sem concorrência com movimentações de Estoque](#)

Ponto de entrada MA330CP (desenvolvido em linguagem AdvPL):

MA330CP
<pre>#include "RWMAKE.CH" User Function MA330CP () LOCAL aRegraCP:={} AADD(aRegraCP,"SB1->B1_TIPO == 'MP'") AADD(aRegraCP,"SB1->B1_TIPO == 'MC'") Return aRegraCP</pre>

Stored Procedure do ponto de entrada MA330CP:

[MSSQL](#)
[ORACLE](#)

```
-- Criacao de procedure
-- ** Esta procedure não é instalada pela instalação de procedures e seu nome não necessariamente deve ser MA330CP_99
-- ** aqui foi usado o mesmo nome do ponto de entrada em ADVPL para melhor entendimento, mas deve ter o nome ch
-- ** logo abaixo no exemplo M330INB2CP_99, Caso a procedure necessite de pesquisar a regra das partes.

CREATE PROCEDURE MA330CP_99 (
    @IN_FILIALCOR Char( 02 ) ,
    @IN_COD Char( 15) ,
    @OUT_RESULTADO Integer output ) AS

-- Declaracoes de variaveis
DECLARE @cFil_SB1 Char( 2)
DECLARE @cB1_TIPO Char( 2)
BEGIN
    EXEC XFILIAL_99 'SB1' , @IN_FILIALCOR , @cFil_SB1 output

    SELECT @cB1_TIPO = B1_TIPO
    FROM SB1990
    WHERE B1_FILIAL = @cFil_SB1 and B1_COD = @IN_COD and D_E_L_E_T_ = ' '

    SET @OUT_RESULTADO = 3

    IF @cB1_TIPO = 'MP'
    BEGIN
        SET @OUT_RESULTADO = 1
    END
    IF @cB1_TIPO = 'MC'
    BEGIN
        SET @OUT_RESULTADO = 2
    END
END
```

Atenção aos tópicos abaixo:

1. O ponto de entrada MA330CP foi escrito em linguagem SQL e Oracle. Caso esteja utilizando outro gerenciador de banco de dados, é necessário fazer a compatibilização da linguagem escrita.
2. O nome da *stored procedure* deverá ser compatibilizada de acordo com o ambiente do cliente. Exemplo:

Caso o cliente esteja utilizando a Empresa 01, deve alterar a chamada da *procedure* MA330CP_99 para MA330CP_01 e, também, alterar a chamada da *procedure* XFILIAL_99 para XFILIAL_01.

Caso não seja feita essa alteração, e seja mantida a chamada XFILIAL_99, ao salvar a *procedure* no banco de dados será apresentada a seguinte mensagem:

"O módulo 'MA330CP_01' depende do objeto ausente 'XFILIAL_99'. O módulo ainda será criado; entretanto, não poderá ser executado com êxito até o objeto existir."

Caso o cliente esteja utilizando a Empresa 01, é necessário alterar o nome físico da tabela SB1990 para SB1010.
3. A implementação do custo em partes exige um grau elevado de conhecimentos em *stored procedures*. Desta forma, é recomendado que o procedimento descrito seja efetuado por um profissional qualificado TOTVS.

Utilizando o exemplo, é possível verificar o custo dividido em três partes:

- A parte 1 composta dos valores de produtos que tenham o campo B1_TIPO = "MP"
- A parte 2 composta dos valores de produtos que tenham o campo B1_TIPO = "MC"
- A parte 3 composta dos valores de produtos que não se enquadram nas partes citadas

Para utilizar o custo em partes devem ser criados alguns campos no sistema seguindo as regras e nomes, conforme quadro abaixo. Se o ponto de entrada que define as regras do custo existir e os campos não forem criados, a rotina de recálculo do custo processará o recálculo do custo **desconsiderando** o custo em partes.

Os campos devem ser definidos utilizando a seguinte regra:

CC – Numero do Custo Em Partes

MM – Numero da Moeda (1 a 5)

TABELA	CAMPOS
SB9 – Saldos Iniciais	B9_CP + CC + MM (Custo em partes) – B9_VINI1
	B9_CPM + CC + MM (Custo em partes Unitário) – B9_CM1
SB2/TR2XXSP – Saldos em Estoque	B2_CP + CC + MM (Custo em partes Unitário) – B2_CM1
	B2_CPF + CC + MM (Custo em partes Final) – B2_VFIM1
SB6 – Saldos em poder de terceiros	B6_CP + CC + MM (Custo em partes) – B6_CUSTO1
SC2 – Ordens de Produção	C2_CPF + CC + MM (Custo em partes Final) – C2_VFIM1
	C2_CPI + CC + MM (Custo em partes Inicial) – C2_VINI1
	C2_API + CC + MM (Apropriação Inicial do Custo em partes) – C2_APRINI1
	C2_APF + CC + MM (Apropriação Final do Custo em partes) – C2_APRFIM1
SD1 – Itens das NFs de Entrada	D1_CP + CC + MM (Custo em partes) – D1_CUSTO
SD2 – Itens das NFs de Saída	D2_CP + CC + MM (Custo em partes) – D2_CUSTO1
SD3 – Movimentos Internos	D3_CP+ CC + MM (Custo em partes) – D2_CUSTO1

Além dos campos descritos, também devem ser ajustadas as *stored procedures* M330INB2CP, M330INC2CP, MA280INB9CP e MA280INC2CP. Este ajuste é necessário devido à implementação dos novos campos. Exemplos de como realizar o ajuste:

Stored Procedure M330INB2CP

[MSSQL](#)
[ORACLE](#)

```
ALTER PROCEDURE M330INB2CP_99 (
    @IN_FILIALCOR Char( 02 ) ,
    @IN_DINICIO   Char( 08 ) ,
    @IN_CUSUNIF   Char( 01 ) ,
    @IN_COD       Char( 15 ) ,
    @IN_LOCAL     Char( 2  ) ,
    @IN_RECNOSB2  Integer ) AS
```

```
-- Declaracoes de variaveis
DECLARE @nB2_VFIM1  Float
DECLARE @nB2_VFIM2  Float
DECLARE @nB2_VFIM3  Float
DECLARE @nB2_VFIM4  Float
DECLARE @nB2_VFIM5  Float
DECLARE @nB9_CP0101 Float
DECLARE @nB9_CP0102 Float
DECLARE @nB9_CP0103 Float
DECLARE @nB9_CP0104 Float
DECLARE @nB9_CP0105 Float
DECLARE @nB9_CP0201 Float
DECLARE @nB9_CP0202 Float
DECLARE @nB9_CP0203 Float
DECLARE @nB9_CP0204 Float
DECLARE @nB9_CP0205 Float
DECLARE @nB9_CP0301 Float
DECLARE @nB9_CP0302 Float
DECLARE @nB9_CP0303 Float
DECLARE @nB9_CP0304 Float
DECLARE @nB9_CP0305 Float
DECLARE @nB9_CPM0101 Float
DECLARE @nB9_CPM0102 Float
DECLARE @nB9_CPM0103 Float
```

```

DECLARE @nB9_CPM0104 Float
DECLARE @nB9_CPM0105 Float
DECLARE @nB9_CPM0201 Float
DECLARE @nB9_CPM0202 Float
DECLARE @nB9_CPM0203 Float
DECLARE @nB9_CPM0204 Float
DECLARE @nB9_CPM0205 Float
DECLARE @nB9_CPM0301 Float
DECLARE @nB9_CPM0302 Float
DECLARE @nB9_CPM0303 Float
DECLARE @nB9_CPM0304 Float
DECLARE @nB9_CPM0305 Float
DECLARE @nParte Integer
DECLARE @nQtd Float
DECLARE @cFil_SB9 Char(02)
DECLARE @iRecnoTRT Integer
DECLARE @cFilAux Char(02)

BEGIN

EXEC XFILIAL_99 'SB9' , @IN_FILIALCOR , @cFil_SB9 output
UPDATE TR299SP
WITH (ROWLOCK)
SET B2_CPF0101= 0, B2_CPF0102 = 0, B2_CPF0103 = 0, B2_CPF0104 = 0, B2_CPF0105= 0,
    B2_CPF0201= 0, B2_CPF0202 = 0, B2_CPF0203 = 0, B2_CPF0204 = 0, B2_CPF0205= 0,
    B2_CPF0301= 0, B2_CPF0302 = 0, B2_CPF0303 = 0, B2_CPF0304 = 0, B2_CPF0305= 0,
    B2_CP0101 = 0, B2_CP0102 = 0, B2_CP0103 = 0, B2_CP0104 = 0, B2_CP0105 = 0,
    B2_CP0201 = 0, B2_CP0202 = 0, B2_CP0203 = 0, B2_CP0204 = 0, B2_CP0205 = 0,
    B2_CP0301 = 0, B2_CP0302 = 0, B2_CP0303 = 0, B2_CP0304 = 0, B2_CP0305 = 0
WHERE R_E_C_N_O_ = @IN_RECNOB2

SELECT @nB9_CP0101 = ISNULL(B9_CP0101,0), @nB9_CP0102 = ISNULL(B9_CP0102,0),
    @nB9_CP0103 = ISNULL(B9_CP0103,0), @nB9_CP0104 = ISNULL(B9_CP0104,0),
    @nB9_CP0105 = ISNULL(B9_CP0105,0), @nB9_CP0201 = ISNULL(B9_CP0201,0),
    @nB9_CP0202 = ISNULL(B9_CP0202,0), @nB9_CP0203 = ISNULL(B9_CP0203,0),
    @nB9_CP0204 = ISNULL(B9_CP0204,0), @nB9_CP0205 = ISNULL(B9_CP0205,0),
    @nB9_CP0301 = ISNULL(B9_CP0301,0), @nB9_CP0302 = ISNULL(B9_CP0302,0),
    @nB9_CP0303 = ISNULL(B9_CP0303,0), @nB9_CP0304 = ISNULL(B9_CP0304,0),
    @nB9_CP0305 = ISNULL(B9_CP0305,0),
    @nB9_CPM0101 = ISNULL(B9_CPM0101, 0), @nB9_CPM0102 = ISNULL(B9_CPM0102, 0),
    @nB9_CPM0103 = ISNULL(B9_CPM0103, 0), @nB9_CPM0104 = ISNULL(B9_CPM0104, 0),
    @nB9_CPM0105 = ISNULL(B9_CPM0105, 0), @nB9_CPM0201 = ISNULL(B9_CPM0201, 0),
    @nB9_CPM0202 = ISNULL(B9_CPM0202, 0), @nB9_CPM0203 = ISNULL(B9_CPM0203, 0),
    @nB9_CPM0204 = ISNULL(B9_CPM0204, 0), @nB9_CPM0205 = ISNULL(B9_CPM0205, 0),
    @nB9_CPM0301 = ISNULL(B9_CPM0301, 0), @nB9_CPM0302 = ISNULL(B9_CPM0302, 0),
    @nB9_CPM0303 = ISNULL(B9_CPM0303, 0), @nB9_CPM0304 = ISNULL(B9_CPM0304, 0),
    @nB9_CPM0305 = ISNULL(B9_CPM0305, 0)

FROM SB9990
WHERE B9_FILIAL = @cFil_SB9 AND B9_COD = @IN_COD AND B9_LOCAL = @IN_LOCAL AND
    B9_DATA = (
        SELECT MAX(B9_DATA)
        FROM SB9990
        WHERE B9_FILIAL = @cFil_SB9 AND
            B9_COD = @IN_COD AND
            B9_LOCAL = @IN_LOCAL AND
            B9_DATA <= @IN_DINICIO AND
            D_E_L_E_T_ = ' '
    ) AND D_E_L_E_T_ = ' '

SET @nB9_CP0101 = ISNULL(@nB9_CP0101, 0)
SET @nB9_CP0102 = ISNULL(@nB9_CP0102, 0)
SET @nB9_CP0103 = ISNULL(@nB9_CP0103, 0)
SET @nB9_CP0104 = ISNULL(@nB9_CP0104, 0)
SET @nB9_CP0105 = ISNULL(@nB9_CP0105, 0)
SET @nB9_CP0201 = ISNULL(@nB9_CP0201, 0)
SET @nB9_CP0202 = ISNULL(@nB9_CP0202, 0)

```

```

SET @nB9_CP0203 = ISNULL(@nB9_CP0203, 0)
SET @nB9_CP0204 = ISNULL(@nB9_CP0204, 0)
SET @nB9_CP0205 = ISNULL(@nB9_CP0205, 0)
SET @nB9_CP0301 = ISNULL(@nB9_CP0301, 0)
SET @nB9_CP0302 = ISNULL(@nB9_CP0302, 0)
SET @nB9_CP0303 = ISNULL(@nB9_CP0303, 0)
SET @nB9_CP0304 = ISNULL(@nB9_CP0304, 0)
SET @nB9_CP0305 = ISNULL(@nB9_CP0305, 0)
SET @nB9_CPM0101 = ISNULL(@nB9_CPM0101, 0)
SET @nB9_CPM0102 = ISNULL(@nB9_CPM0102, 0)
SET @nB9_CPM0103 = ISNULL(@nB9_CPM0103, 0)
SET @nB9_CPM0104 = ISNULL(@nB9_CPM0104, 0)
SET @nB9_CPM0105 = ISNULL(@nB9_CPM0105, 0)
SET @nB9_CPM0201 = ISNULL(@nB9_CPM0201, 0)
SET @nB9_CPM0202 = ISNULL(@nB9_CPM0202, 0)
SET @nB9_CPM0203 = ISNULL(@nB9_CPM0203, 0)
SET @nB9_CPM0204 = ISNULL(@nB9_CPM0204, 0)
SET @nB9_CPM0205 = ISNULL(@nB9_CPM0205, 0)
SET @nB9_CPM0301 = ISNULL(@nB9_CPM0301, 0)
SET @nB9_CPM0302 = ISNULL(@nB9_CPM0302, 0)
SET @nB9_CPM0303 = ISNULL(@nB9_CPM0303, 0)
SET @nB9_CPM0304 = ISNULL(@nB9_CPM0304, 0)
SET @nB9_CPM0305 = ISNULL(@nB9_CPM0305, 0)

IF @IN_CUSUNIF = '1' OR @IN_CUSUNIF = '2'
    BEGIN
        IF @IN_CUSUNIF = '1' --Custo unificado por filial
            BEGIN
                SET @cFilAux = @IN_FILIALCOR
            END
        ELSE --Custo unificado por empresa
            BEGIN
                SET @cFilAux = ' ' --Número de espaços em branco igual ao tamanho da f
            END

        BEGIN
            SELECT @iRecnoTRT = ISNULL(MAX(R_E_C_N_O_), 0)
            FROM TRT99SP
            WHERE TRB_FILIAL = @cFilAux
            AND TRB_COD = @IN_COD

            IF @iRecnoTRT = 0
                BEGIN
                    SELECT @iRecnoTRT = ISNULL(MAX(R_E_C_N_O_), 0)
                    FROM TRT99SP
                    SET @iRecnoTRT = @iRecnoTRT + 1
                    INSERT INTO TRT99SP(TRB_FILIAL, TRB_COD, R_E_C_N_O_)
                    VALUES(@cFilAux, @IN_COD, @iRecnoTRT)
                END

            SELECT @nQtd = ISNULL(TRB_QFIM, 0)
            FROM TRT99SP
            WHERE R_E_C_N_O_ = @iRecnoTRT

            IF @nQtd = 0
                BEGIN
                    SELECT @nQtd = 1
                END

            UPDATE TRT99SP
            WITH (ROWLOCK)
            SET TRB_VF0101 = TRB_VF0101 + @nB9_CP0101,
                TRB_VF0102 = TRB_VF0102 + @nB9_CP0102,
                TRB_VF0103 = TRB_VF0103 + @nB9_CP0103,
                TRB_VF0104 = TRB_VF0104 + @nB9_CP0104,

```

```

TRB_VF0105 = TRB_VF0105 + @nB9_CP0105,
TRB_VF0201 = TRB_VF0201 + @nB9_CP0201,
TRB_VF0202 = TRB_VF0202 + @nB9_CP0202,
TRB_VF0203 = TRB_VF0203 + @nB9_CP0203,
TRB_VF0204 = TRB_VF0204 + @nB9_CP0204,
TRB_VF0205 = TRB_VF0205 + @nB9_CP0205,
TRB_VF0301 = TRB_VF0301 + @nB9_CP0301,
TRB_VF0302 = TRB_VF0302 + @nB9_CP0302,
TRB_VF0303 = TRB_VF0303 + @nB9_CP0303,
TRB_VF0304 = TRB_VF0304 + @nB9_CP0304,
TRB_VF0305 = TRB_VF0305 + @nB9_CP0305,
TRB_CP0101 = (TRB_VF0101 + @nB9_CP0101) / @nQtd,
TRB_CP0102 = (TRB_VF0102 + @nB9_CP0102) / @nQtd,
TRB_CP0103 = (TRB_VF0103 + @nB9_CP0103) / @nQtd,
TRB_CP0104 = (TRB_VF0104 + @nB9_CP0104) / @nQtd,
TRB_CP0105 = (TRB_VF0105 + @nB9_CP0105) / @nQtd,
TRB_CP0201 = (TRB_VF0201 + @nB9_CP0201) / @nQtd,
TRB_CP0202 = (TRB_VF0202 + @nB9_CP0202) / @nQtd,
TRB_CP0203 = (TRB_VF0203 + @nB9_CP0203) / @nQtd,
TRB_CP0204 = (TRB_VF0204 + @nB9_CP0204) / @nQtd,
TRB_CP0205 = (TRB_VF0205 + @nB9_CP0205) / @nQtd,
TRB_CP0301 = (TRB_VF0301 + @nB9_CP0301) / @nQtd,
TRB_CP0302 = (TRB_VF0302 + @nB9_CP0302) / @nQtd,
TRB_CP0303 = (TRB_VF0303 + @nB9_CP0303) / @nQtd,
TRB_CP0304 = (TRB_VF0304 + @nB9_CP0304) / @nQtd,
TRB_CP0305 = (TRB_VF0305 + @nB9_CP0305) / @nQtd
WHERE R_E_C_N_O_ = @iRecnoTRT

```

END

END

UPDATE TR299SP

WITH (ROWLOCK)

```

SET B2_CPF0101 = @nB9_CP0101, B2_CPF0102 = @nB9_CP0102,
    B2_CPF0103 = @nB9_CP0103, B2_CPF0104 = @nB9_CP0104,
    B2_CPF0105 = @nB9_CP0105,
    B2_CPF0201 = @nB9_CP0201, B2_CPF0202 = @nB9_CP0202,
    B2_CPF0203 = @nB9_CP0203, B2_CPF0204 = @nB9_CP0204,
    B2_CPF0205 = @nB9_CP0205,
    B2_CPF0301 = @nB9_CP0301, B2_CPF0302 = @nB9_CP0302,
    B2_CPF0303 = @nB9_CP0303, B2_CPF0304 = @nB9_CP0304,
    B2_CPF0305 = @nB9_CP0305

```

WHERE R_E_C_N_O_ = @IN_RECNO5B2

SELECT @nQtd = ISNULL(B2_QFIM, 0)

FROM TR299SP

WHERE R_E_C_N_O_ = @IN_RECNO5B2

IF @nQtd = 0

BEGIN

UPDATE TR299SP

WITH (ROWLOCK)

```

SET B2_CP0101 = @nB9_CPM0101,
    B2_CP0102 = @nB9_CPM0102,
    B2_CP0103 = @nB9_CPM0103,
    B2_CP0104 = @nB9_CPM0104,
    B2_CP0105 = @nB9_CPM0105,
    B2_CP0201 = @nB9_CPM0201,
    B2_CP0202 = @nB9_CPM0202,
    B2_CP0203 = @nB9_CPM0203,
    B2_CP0204 = @nB9_CPM0204,
    B2_CP0205 = @nB9_CPM0205,
    B2_CP0301 = @nB9_CPM0301,
    B2_CP0302 = @nB9_CPM0302,
    B2_CP0303 = @nB9_CPM0303,

```

```

        B2_CP0304 = @nB9_CPM0304,
        B2_CP0305 = @nB9_CPM0305
    WHERE R_E_C_N_O_ = @IN_RECNOB2

END

ELSE

    BEGIN

        UPDATE TR299SP
        WITH (ROWLOCK)
        SET B2_CP0101 = @nB9_CP0101 / @nQtd,
            B2_CP0102 = @nB9_CP0102 / @nQtd,
            B2_CP0103 = @nB9_CP0103 / @nQtd,
            B2_CP0104 = @nB9_CP0104 / @nQtd,
            B2_CP0105 = @nB9_CP0105 / @nQtd,
            B2_CP0201 = @nB9_CP0201 / @nQtd,
            B2_CP0202 = @nB9_CP0202 / @nQtd,
            B2_CP0203 = @nB9_CP0203 / @nQtd,
            B2_CP0204 = @nB9_CP0204 / @nQtd,
            B2_CP0205 = @nB9_CP0205 / @nQtd,
            B2_CP0301 = @nB9_CP0301 / @nQtd,
            B2_CP0302 = @nB9_CP0302 / @nQtd,
            B2_CP0303 = @nB9_CP0303 / @nQtd,
            B2_CP0304 = @nB9_CP0304 / @nQtd,
            B2_CP0305 = @nB9_CP0305 / @nQtd

        WHERE R_E_C_N_O_ = @IN_RECNOB2

    END

END

```

Atenção aos tópicos abaixo:

1. O ponto de entrada M330INB2CP foi escrito em linguagem SQL e Oracle. Caso esteja utilizando outro gerenciador de banco de dados, é necessário fazer a compatibilização da linguagem escrita.
2. O nome da *stored procedure* deve ser compatibilizada de acordo com o ambiente do cliente. Exemplo:

Se for utilizada a Empresa 01, é necessário alterar a chamada das *procedures*:

 - M330INB2CP_99 para M330INB2CP_01
 - XFILIAL_99 para XFILIAL_01
 - MA330CP_99 para MA330CP_01

Se não for feita essa alteração e for mantida a chamada XFILIAL_99, ao salvar a *procedure* no banco de dados será apresentada a seguinte mensagem: "O módulo 'MA330CP_01' depende do objeto ausente 'XFILIAL_99'. O módulo ainda será criado; entretanto, não poderá ser executado com êxito até o objeto existir."

Se for utilizada a Empresa 01, é necessário alterar o nome físico das seguintes tabelas:

- SB2990 para SB2010 ou TR201SP, dependendo do parâmetro MV_A330SB2 ou release (ver: [Recálculo do Custo Médio sem concorrência com movimentações de Estoque](#))
- SB9990 para SB9010
- TRT99SP para TRT01SP

3. A implementação do custo em partes exige um grau elevado de conhecimentos em *stored procedures* e Protheus. Desta forma, é recomendado que o procedimento seja efetuado por um profissional qualificado TOTVS.

Stored Procedure M330INC2CP

MSSQL ORACLE

```

-- Criacao de procedure
CREATE PROCEDURE M330INC2CP_99 (
    @IN_FILIALCOR Char( 02 ) ) AS

-- Declaracoes de variaveis
DECLARE @cFil_SC2      Char( 02 )
DECLARE @nMaxRecnoSC2 Integer

```



```

DECLARE @nRec Integer
BEGIN
    EXEC XFILIAL_99 'SC2' , @IN_FILIALCOR , @cFil_SC2 output

    SELECT @nMaxRecnoSC2 = ISNULL ( MAX ( R_E_C_N_O_ ), 0 )
    FROM SC2990
    WHERE C2_FILIAL = @cFil_SC2 and D_E_L_E_T_ <> '*'

    SELECT @nRec = ISNULL ( MIN ( R_E_C_N_O_ ), 0 )
    FROM SC2990
    WHERE C2_FILIAL = @cFil_SC2 and D_E_L_E_T_ <> '*'

    WHILE (@nRec <= @nMaxRecnoSC2 )
    BEGIN
        UPDATE SC2010
        WITH (ROWLOCK)
        SET C2_CPF0101 = C2_CPI0101 ,
            C2_CPF0102 = C2_CPI0102 ,
            C2_CPF0103 = C2_CPI0103 ,
            C2_CPF0104 = C2_CPI0104 ,
            C2_CPF0105 = C2_CPI0105 ,
            C2_CPF0201 = C2_CPI0201 ,
            C2_CPF0202 = C2_CPI0202 ,
            C2_CPF0203 = C2_CPI0203 ,
            C2_CPF0204 = C2_CPI0204 ,
            C2_CPF0205 = C2_CPI0205 ,
            C2_CPF0301 = C2_CPI0301 ,
            C2_CPF0302 = C2_CPI0302 ,
            C2_CPF0303 = C2_CPI0303 ,
            C2_CPF0304 = C2_CPI0304 ,
            C2_CPF0305 = C2_CPI0305 ,
            C2_APF0101 = C2_API0101 ,
            C2_APF0102 = C2_API0102 ,
            C2_APF0103 = C2_API0103 ,
            C2_APF0104 = C2_API0104 ,
            C2_APF0105 = C2_API0105 ,
            C2_APF0201 = C2_API0201 ,
            C2_APF0202 = C2_API0202 ,
            C2_APF0203 = C2_API0203 ,
            C2_APF0204 = C2_API0204 ,
            C2_APF0205 = C2_API0205 ,
            C2_APF0301 = C2_API0301 ,
            C2_APF0302 = C2_API0302 ,
            C2_APF0303 = C2_API0303 ,
            C2_APF0304 = C2_API0304 ,
            C2_APF0305 = C2_API0305 ,
        WHERE R_E_C_N_O_ >= @nRec and R_E_C_N_O_ < @nRec + 1024 and
            C2_FILIAL = @cFil_SC2 and D_E_L_E_T_ <> '*'
        SET @nRec = @nRec + 1024
    END
END

```

Atenção aos tópicos abaixo:

1. O ponto de entrada M330INC2CP foi escrito em linguagem SQL e Oracle. Se for utilizado outro gerenciador de banco de dados, é necessário compatibilizar a linguagem escrita
2. O nome da *stored procedure* deve ser compatibilizada de acordo com o ambiente do cliente. Exemplo:

Se for utilizada a Empresa 01, é necessário alterar a chamada das *procedures*:

- M330INC2CP_99 para M330INC2CP_01
- XFILIAL_99 para XFILIAL_01

Se não for feita essa alteração e for mantida a chamada XFILIAL_99, ao salvar a *procedure* no banco de dados será apresentada a seguinte mensagem:

"O módulo 'MA330CP_01' depende do objeto ausente 'XFILIAL_99'. O módulo ainda será criado; entretanto, não poderá ser executado com êxito até o objeto existir."

Se for utilizada a Empresa 01, é necessário alterar o nome físico da tabela:

- SC2990 para SC2010

3. A implementação do custo em partes exige um grau elevado de conhecimentos em *stored procedures* e Protheus. Desta forma, é recomendado que o procedimento seja efetuado por um profissional qualificado TOTVS.

Stored Procedure MA280INB9CP

MSSQL ORACLE

```
MA280INB9CP

ALTER PROCEDURE MA280INB9CP_99 (
    @IN_FILIALCOR Char( 02 ) ,
    @IN_COD Char( 15 ) ,
    @IN_MV_CUSZERO Char( 01 ) ,
    @IN_RECNOSB9 Integer ,
    @IN_RECNOSB2 Integer ,
    @IN_NDIVISOR Float ,
    @IN_B9_VINI1 Float ,
    @IN_B9_VINI2 Float ,
    @IN_B9_VINI3 Float ,
    @IN_B9_VINI4 Float ,
    @IN_B9_VINI5 Float ) AS

-- Declaracoes de variaveis
-- Variaveis custo em partes (total)
DECLARE @nCP0101 Float
DECLARE @nCP0102 Float
DECLARE @nCP0103 Float
DECLARE @nCP0104 Float
DECLARE @nCP0105 Float
DECLARE @nCP0201 Float
DECLARE @nCP0202 Float
DECLARE @nCP0203 Float
DECLARE @nCP0204 Float
DECLARE @nCP0205 Float
DECLARE @nCP0301 Float
DECLARE @nCP0302 Float
DECLARE @nCP0303 Float
DECLARE @nCP0304 Float
DECLARE @nCP0305 Float
-- Variaveis custo em partes (unitario)
DECLARE @nCPM0101 Float
DECLARE @nCPM0102 Float
DECLARE @nCPM0103 Float
DECLARE @nCPM0104 Float
DECLARE @nCPM0105 Float
DECLARE @nCPM0201 Float
DECLARE @nCPM0202 Float
DECLARE @nCPM0203 Float
DECLARE @nCPM0204 Float
DECLARE @nCPM0205 Float
DECLARE @nCPM0301 Float
DECLARE @nCPM0302 Float
DECLARE @nCPM0303 Float
DECLARE @nCPM0304 Float
DECLARE @nCPM0305 Float
DECLARE @nB9_QINI Float
--Variaveis para evitar diferenca de arredondamento
DECLARE @nREST01 Float
```

```

DECLARE @nREST02 Float
DECLARE @nREST03 Float
DECLARE @nREST04 Float
DECLARE @nREST05 Float

BEGIN

SELECT  @nB9_QINI = ISNULL(B9_QINI,0),
        @nREST01 = ISNULL(B9_VINI1,0),
        @nREST02 = ISNULL(B9_VINI2,0),
        @nREST03 = ISNULL(B9_VINI3,0),
        @nREST04 = ISNULL(B9_VINI4,0),
        @nREST05 = ISNULL(B9_VINI5,0)

FROM SB9990
WHERE R_E_C_N_O_ = @IN_RECNO SB9

SELECT  @nCP0101 = ISNULL(B2_CPF0101,0), @nCP0102 = ISNULL(B2_CPF0102,0),
        @nCP0103 = ISNULL(B2_CPF0103,0), @nCP0104 = ISNULL(B2_CPF0104,0),
        @nCP0105 = ISNULL(B2_CPF0105,0), @nCP0201 = ISNULL(B2_CPF0201,0),
        @nCP0202 = ISNULL(B2_CPF0202,0), @nCP0203 = ISNULL(B2_CPF0203,0),
        @nCP0204 = ISNULL(B2_CPF0204,0), @nCP0205 = ISNULL(B2_CPF0205,0),
        @nCP0301 = ISNULL(B2_CPF0301,0), @nCP0302 = ISNULL(B2_CPF0302,0),
        @nCP0303 = ISNULL(B2_CPF0303,0), @nCP0304 = ISNULL(B2_CPF0304,0),
        @nCP0305 = ISNULL(B2_CPF0305,0),
        @nCPM0101 = ISNULL(B2_CP0101,0), @nCPM0102 = ISNULL(B2_CP0102,0),
        @nCPM0103 = ISNULL(B2_CP0103,0), @nCPM0104 = ISNULL(B2_CP0104,0),
        @nCPM0105 = ISNULL(B2_CP0105,0), @nCPM0201 = ISNULL(B2_CP0201,0),
        @nCPM0202 = ISNULL(B2_CP0202,0), @nCPM0203 = ISNULL(B2_CP0203,0),
        @nCPM0204 = ISNULL(B2_CP0204,0), @nCPM0205 = ISNULL(B2_CP0205,0),
        @nCPM0301 = ISNULL(B2_CP0301,0), @nCPM0302 = ISNULL(B2_CP0302,0),
        @nCPM0303 = ISNULL(B2_CP0303,0), @nCPM0304 = ISNULL(B2_CP0304,0),
        @nCPM0305 = ISNULL(B2_CP0305,0)

FROM SB2990 -- Sempre SB2990, independente da release maior ou igual a 12.1.2310, ou parâmetro MV_A3305
WHERE R_E_C_N_O_ = @IN_RECNO SB2

-- Obtem custo em partes (total)
IF @nB9_QINI <= 0 OR @nCP0101 < 0
    BEGIN
        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0101 = 0 END
        ELSE BEGIN SET @nCP0101 = @nCPM0101 END
    END
ELSE
    BEGIN SET @nCP0101 = @nCP0101 / @nB9_QINI END
BEGIN
    SET @nCP0101 = @nCP0101 * @nB9_QINI
    SET @nREST01 = @nREST01 - @nCP0101 -- tratamento para evitar problemas de arredondamento
END

IF @nB9_QINI <= 0 OR @nCP0102 < 0
    BEGIN
        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0102 = 0 END
        ELSE BEGIN SET @nCP0102 = @nCPM0102 END
    END
ELSE
    BEGIN SET @nCP0102 = @nCP0102 / @nB9_QINI END
BEGIN
    SET @nCP0102 = @nCP0102 * @nB9_QINI
    SET @nREST02 = @nREST02 - @nCP0102
END

IF @nB9_QINI <= 0 OR @nCP0103 < 0
    BEGIN

```

```

        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0103 = 0 END
        ELSE BEGIN SET @nCP0103 = @nCPM0103 END
    END
ELSE
    BEGIN SET @nCP0103 = @nCP0103 / @nB9_QINI END
BEGIN
    SET @nCP0103 = @nCP0103 * @nB9_QINI
    SET @nREST03 = @nREST03 - @nCP0103
END

IF @nB9_QINI <= 0 OR @nCP0104 < 0
    BEGIN
        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0104 = 0 END
        ELSE BEGIN SET @nCP0104 = @nCPM0104 END
    END
ELSE
    BEGIN SET @nCP0104 = @nCP0104 / @nB9_QINI END
BEGIN
    SET @nCP0104 = @nCP0104 * @nB9_QINI
    SET @nREST04 = @nREST04 - @nCP0104
END

IF @nB9_QINI <= 0 OR @nCP0105 < 0
    BEGIN
        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0105 = 0 END
        ELSE BEGIN SET @nCP0105 = @nCPM0105 END
    END
ELSE
    BEGIN SET @nCP0105 = @nCP0105 / @nB9_QINI END
BEGIN
    SET @nCP0105 = @nCP0105 * @nB9_QINI
    SET @nREST05 = @nREST05 - @nCP0105
END

IF @nB9_QINI <= 0 OR @nCP0201 < 0
    BEGIN
        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0201 = 0 END
        ELSE BEGIN SET @nCP0201 = @nCPM0201 END
    END
ELSE
    BEGIN SET @nCP0201 = @nCP0201 / @nB9_QINI END
BEGIN
    SET @nCP0201 = @nCP0201 * @nB9_QINI
    SET @nREST01 = @nREST01 - @nCP0201
END

IF @nB9_QINI <= 0 OR @nCP0202 < 0
    BEGIN
        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0202 = 0 END
        ELSE BEGIN SET @nCP0202 = @nCPM0202 END
    END
ELSE
    BEGIN SET @nCP0202 = @nCP0202 / @nB9_QINI END
BEGIN
    SET @nCP0202 = @nCP0202 * @nB9_QINI
    SET @nREST02 = @nREST02 - @nCP0202
END

IF @nB9_QINI <= 0 OR @nCP0203 < 0

```

```

BEGIN
    IF @IN_MV_CUSZERO = 'S'
        BEGIN SET @nCP0203 = 0 END
    ELSE BEGIN SET @nCP0203 = @nCPM0203 END
END
ELSE
    BEGIN SET @nCP0203 = @nCP0203 / @nB9_QINI END
BEGIN
    SET @nCP0203 = @nCP0203 * @nB9_QINI
    SET @nREST03 = @nREST03 - @nCP0203
END

IF @nB9_QINI <= 0 OR @nCP0204 < 0
    BEGIN
        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0204 = 0 END
        ELSE BEGIN SET @nCP0204 = @nCPM0204 END
    END
ELSE
    BEGIN SET @nCP0204 = @nCP0204 / @nB9_QINI END
BEGIN
    SET @nCP0204 = @nCP0204 * @nB9_QINI
    SET @nREST04 = @nREST04 - @nCP0204
END

IF @nB9_QINI <= 0 OR @nCP0205 < 0
    BEGIN
        IF @IN_MV_CUSZERO = 'S'
            BEGIN SET @nCP0205 = 0 END
        ELSE BEGIN SET @nCP0205 = @nCPM0205 END
    END
ELSE
    BEGIN SET @nCP0205 = @nCP0205 / @nB9_QINI END
BEGIN
    SET @nCP0205 = @nCP0205 * @nB9_QINI
    SET @nREST05 = @nREST05 - @nCP0205
END

-- Na ultima parte processa valor da diferenca para evitar problemas de arredondamento
BEGIN SET @nCP0301 = @nREST01 END
BEGIN SET @nCP0302 = @nREST02 END
BEGIN SET @nCP0303 = @nREST03 END
BEGIN SET @nCP0304 = @nREST04 END
BEGIN SET @nCP0305 = @nREST05 END

UPDATE SB9990
WITH (ROWLOCK)
SET B9_CP0101 = @nCP0101, B9_CP0102 = @nCP0102, B9_CP0103 = @nCP0103,
    B9_CP0104 = @nCP0104, B9_CP0105 = @nCP0105, B9_CP0201 = @nCP0201,
    B9_CP0202 = @nCP0202, B9_CP0203 = @nCP0203, B9_CP0204 = @nCP0204,
    B9_CP0205 = @nCP0205, B9_CP0301 = @nCP0301, B9_CP0302 = @nCP0302,
    B9_CP0303 = @nCP0303, B9_CP0304 = @nCP0304, B9_CP0305 = @nCP0305,
    B9_CPM0101 = @nCPM0101, B9_CPM0102 = @nCPM0102, B9_CPM0103 = @nCPM0103,
    B9_CPM0104 = @nCPM0104, B9_CPM0105 = @nCPM0105, B9_CPM0201 = @nCPM0201,
    B9_CPM0202 = @nCPM0202, B9_CPM0203 = @nCPM0203, B9_CPM0204 = @nCPM0204,
    B9_CPM0205 = @nCPM0205, B9_CPM0301 = @nCPM0301, B9_CPM0302 = @nCPM0302,
    B9_CPM0303 = @nCPM0303, B9_CPM0304 = @nCPM0304, B9_CPM0305 = @nCPM0305
WHERE R_E_C_N_O_ = @IN_RECNO SB9
END

```

Atenção aos tópicos abaixo:

1. O ponto de entrada MA280INB9CP escrito em linguagem SQL e Oracle. Se for utilizado outro gerenciador de banco de dados, é necessário compatibilizar a linguagem escrita.

2. O nome da *stored procedure* deve ser compatibilizada de acordo com o ambiente do cliente. Exemplo:
- Se for utilizada a Empresa 01, é necessário alterar a chamada das *procedures*:
 - MA280INB9CP_99 para MA280INB9CP_01
 - XFILIAL_99 para XFILIAL_01
 - Se for utilizada a Empresa 01, é necessário alterar o nome físico das tabelas:
 - SB2990 para SB2010
 - SB9990 para SB9010
3. A implementação do custo em partes exige um grau elevado de conhecimentos em *stored procedures* e Protheus. Desta forma, é recomendado que o procedimento seja realizado por um profissional qualificado TOTVS.

Stored Procedure MA280INC2CP

[MSSQL](#) [ORACLE](#)

MA280INC2CP

```
-- Criacao de procedure
CREATE PROCEDURE MA280INC2CP_99 (
    @IN_RECNO5C2 Integer ) AS

BEGIN
    UPDATE SC2990
    WITH (ROWLOCK)
    SET C2_CPI0101 = C2_CPF0101 ,
        C2_CPI0102 = C2_CPF0102 ,
        C2_CPI0103 = C2_CPF0103 ,
        C2_CPI0104 = C2_CPF0104 ,
        C2_CPI0105 = C2_CPF0105 ,
        C2_CPI0201 = C2_CPF0201 ,
        C2_CPI0202 = C2_CPF0202 ,
        C2_CPI0203 = C2_CPF0203 ,
        C2_CPI0204 = C2_CPF0204 ,
        C2_CPI0205 = C2_CPF0205 ,
        C2_CPI0301 = C2_CPF0301 ,
        C2_CPI0302 = C2_CPF0302 ,
        C2_CPI0303 = C2_CPF0303 ,
        C2_CPI0304 = C2_CPF0304 ,
        C2_CPI0305 = C2_CPF0305 ,
        C2_API0101 = C2_APF0101 ,
        C2_API0102 = C2_APF0102 ,
        C2_API0103 = C2_APF0103 ,
        C2_API0104 = C2_APF0104 ,
        C2_API0105 = C2_APF0105 ,
        C2_API0201 = C2_APF0201 ,
        C2_API0202 = C2_APF0202 ,
        C2_API0203 = C2_APF0203 ,
        C2_API0204 = C2_APF0204 ,
        C2_API0205 = C2_APF0205 ,
        C2_API0301 = C2_APF0301 ,
        C2_API0302 = C2_APF0302 ,
        C2_API0303 = C2_APF0303 ,
        C2_API0304 = C2_APF0304 ,
        C2_API0305 = C2_APF0305
    WHERE R_E_C_N_O_ = @IN_RECNO5C2

END
```

Atenção aos tópicos abaixo:

1. O ponto de entrada MA280INC2CP acima foi escrito em linguagem SQL e Oracle. Se for utilizado outro gerenciador de banco de dados, é necessário compatibilizar a linguagem escrita.

2. O nome da *stored procedure* deve ser compatibilizada de acordo com o ambiente do cliente. Exemplo:

Se for utilizada a Empresa 01, é necessário alterar a chamada da *procedure*:

- MA280INC2CP_99 para MA280INC2CP_01

Se for utilizada a Empresa 01, é necessário alterar o nome físico da tabela:

- SC2990 para SC2010

3. A implementação do custo em partes exige um grau elevado de conhecimentos em *stored procedures* e Protheus. Desta forma, é recomendado que o procedimento seja efetuado por um profissional qualificado TOTVS.
4. A quantidade limite de campos em cada tabela deve ser observada, considerando os campos padrões e os campos criados para o custo em partes, que não deve ultrapassar o limite de 359 campos.

❗ Importante:

Para mais informações sobre o limite de campos, acesse [Limite de campos por tabela](#).

Utilização do parâmetro MV_MUDATRT

Quando o parâmetro MV_MUDATRT estiver configurado como .T. , a referência do grupo de tabelas TR* deve ser alterada para TR*_SP, conforme documentação disponível na URL abaixo:

[SIGAMNT com Stored Procedures de Materiais - MV_MUDATRT](#)

Sem rótulos



[Política de
privacidade](#)

[Termos
de uso](#)