

# CPAPI014 - Transferência de Materiais para Produção

 Tempo aproximado para leitura: 9 minutos

## Documento de API

<b>Produto:</b>	Datasul
<b>Ocorrência:</b>	Documentação de API
<b>Nome Físico</b>	cpp/cpapi014.p

## Considerações Gerais

### Objetivo

Efetuar a transferência dos materiais dos depósitos de armazenagem para o depósito de processo. Permite também realizar a devolução total ou parcial dessas transferências. O programa atua exclusivamente com as requisições sumarizadas, geradas pelo programa CP0304, o qual agrupa as necessidades de um item para várias ordens em uma única requisição. As alocações também são tratadas por esse programa, realizando a transferência dos saldos alocados.

### Funcionamento

A API pode ser chamada de duas formas: A padrão (batch) e a persistente.

Na forma padrão, realiza-se a chamada através de um comando RUN normal, passando-se as temp-tables carregadas com as informações necessárias ao processamento de uma ou mais transferências.

A sintaxe para a chamada é:

```
run cpp/cpapi014.p (input table tt-transf,  
                    input-output table tt-erro,  
                    input l-deleta-erros)
```

A execução na forma persistente, permite apenas o “carregamento” na memória do código da API, sem a execução de nenhuma transferência, devendo a temp-table ser passada sem nenhum registro. Uma vez carregada, a API permite a execução de diversos “métodos”, que poderão ser chamados de forma interativa, englobando todo o processo de transferência. Esses “métodos” são procedures que são acessadas através de um “handle” que é retornado quando o programa é executado de forma persistente.

A sintaxe para executar a API em modo persistente é:

```
run cpp/cpapi014.p persistent set h-handle (input      table tt-transf,  
                                             input-output table tt-erro,  
                                             input      l-deleta-erros)
```

A variável denominada h-handle irá receber o “handle” do programa, permitindo o acesso aos “métodos” internos da API.

**Parâmetros:** Os parâmetros utilizados pela API são:

- tt-rep-transf – Temp-table contendo as informações principais da transferência, tais como o percentual a ser transferido, o depósito de destino e o número da requisição sumarizada.

- tt-erro – Temp-table que retorna os erros e “avisos” retornados pela API durante o processamento das transferências.

- l-deleta-erros – Informa à API que o conteúdo anterior, eventualmente existente na temp-table de erros será eliminado logo após o início do processamento da API. Normalmente utiliza-se YES, sendo parametrizado como NO apenas se a chamada para a API fizer parte de uma transação maior, que eventualmente já tenha gerado registros na temp-table de erros e que devam permanecer após a chamada da API.

Além das temp-tables acima, se a API for executada em modo persistente, são utilizadas outras temp-tables a fim de passar os parâmetros para os “métodos” da API. Essas temp-tables são:

- tt-aloca – Contém as informações relativas à alocação das reservas das ordens de produção englobadas pela requisição sumarizada.

tt-req-sum – Contém as informações relativas às requisições sumarizadas de cada item, as quais irão gerar a movimentação de transferência

**Retorno:** A API irá setar o return-value de forma a informar ao programa “chamador” se a execução foi realizada com sucesso ou não. Os valores possíveis para o return-value são: OK (execução correta) e NOK (execução com problemas). Os motivos para a rejeição do movimento são especificados na temp-table tt-erro.

**Principais Campos:** Os principais campos, utilizados pela API são:

#### - TT-TRANSF:

**Tipo-trans** – Determina o tipo de transação a ser realizada pela API (1 – transferência, 2 – devolução).

**Nr-req-sum** – Contém o número da requisição sumarizada que será processada.

**Percentual** – Contém o percentual a ser transferido/devolvido.

**Data** – Data da transação de transferência.

**Cod-estabel** – Código do estabelecimento (opcional).

**Cod-depos-dest** – Código de depósito de destino da transferência ou origem da devolução (opcional).

**Cod-localiz-dest** – Código de localização de destino da transferência ou origem da devolução.

**Conta-contabil** – Conta utilizada para a transferência (opcional).

**Ct-codigo** – Conta utilizada para transferência

**Sc-codigo** – Sub-Conta utilizada para transferência

**Item-ini, Item-fim, Deposito-ini, Deposito-fim** – Seleção de itens e depósitos que farão parte da transferência.

**Loc-ent-única** – Determina se o código de localização única de destino será utilizado.

**Carrega-req-sum** – Informa se a API irá carregar automaticamente uma temp-table interna com as requisições a serem processadas, ou se irá receber através do método pi-recebe-tt-req-sum. Esse parâmetro deve ser YES para execução batch (padrão) e NO para execução persistente.

**Procura-saldos** – Informa se a API irá realizar a busca dos saldos (FIFO) automaticamente no depósito de saída informado (o default para execução batch é YES).

**Prog-seg** – Recebe nome do programa de origem, ou seja, do programa que está “chamando” a API de transferência

**Execução Persistente:** Quando a API for executada em modo persistente, os seguintes métodos estarão disponíveis para execução:

f-adiciona-lista (input c-lista as char  
input c-item as char)

Retorna uma lista separada por vírgulas, contendo a lista original passada como parâmetro mais o item informado (c-item).

pi-recebe-tt-transf (input table tt-transf)  
pi-recebe-tt-req-sum (input table tt-req-sum)  
pi-recebe-tt-aloca (input table tt-aloca)

Recebem um input parameter table contendo a temp-table correspondente. Tem por objetivo enviar para a API uma temp-table, que vai ser utilizada na execução de outros métodos.

pi-retorna-tt-transf (output table tt-transf)  
pi-retorna-tt-req-sum (output table tt-req-sum)  
pi-retorna-tt-aloca (output table tt-aloca)

Retornam (por meio de um output parameter table) da API a temp-table correspondente.

pi-processa-transf (input-output table tt-transf,  
input-output table tt-erro,  
input l-deleta-erros)

Procedure principal da API de transferência, responsável pelo processamento e geração de todos os movimentos relativos à transferência. Recebe os mesmos parâmetros que a API de transferência quando executada em modo batch, processando os movimentos de estoque relativos à transferência. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, os movimentos não foram realizados.

pi-valida-transf (input l-imediato,  
input i-nr-req-sum,  
output c-erro,  
output c-texto)

Realiza todas as validações relativas ao movimento de transferência. O retorno da procedure ocorre através dos output parameters c-erro (que contém uma lista dos códigos de erros encontrados) e c-texto (que contém uma lista de textos eventualmente associados a cada erro). Também é utilizado o return-value (“OK” ou “NOK”). Para executar essa procedure, é necessário que a API tenha recebido a temp-table de transferência (pi-recebe-tt-transf).

pi-valida-req-sum (input i-nr-ord-prod,  
input i-seq,  
input i-tipo-trans,  
input l-imediato,  
output c-erro,  
output c-texto)

Valida um registro da temp-table de req-sum, localizado através dos parâmetros i-nr-req-sum e i-seq, que contém respectivamente o número da requisição sumarizada e a sequência do registro, sendo que esses campos constituem a chave primária e única da temp-table tt-req-sum. Os parâmetros l-imediato, c-erro, c-texto e também o return-value funcionam como nas procedures anteriores. É necessário que a temp-table de req-sum já esteja carregada na instância da API. O parâmetro i-tipo-trans informa se a transação é uma transferência (1) ou devolução (2).

pi-valida-alocacao (input i-nr-req-sum,  
input i-seq,  
input i-tipo-trans  
input l-imediato,  
output c-erro,  
output c-texto)

Realiza a validação de um registro da temp-table de alocações. Os demais parâmetros e o return-value seguem o funcionamento das procedures anteriores.

pi-finalizar

Finaliza a execução persistente da API, eliminando o código da memória e tornando o handle inválido. Deve ser executada após o término de todos os movimentos, de preferência na saída do programa “chamador”.

pi-verifica-saldo (input i-transf-dev,  
input c-cod-estabel,  
input c-cod-depos-dest)

Verifica se as requisições existentes na temp-table tt-req-sum possuem ou não saldo disponível para serem atendidas. O campo c-cod-estabel indica o estabelecimento onde será realizada a busca de saldos. O campo i-transf-dev deve conter o valor 1 (transferência) ou 2 (devolução). O parâmetro c-cod-depos-dest contém o depósito de destino da transferência ou de origem da devolução. Para que essa procedure seja executada, é necessário que a temp-table tt-req-sum tenha sido previamente carregada.

pi-carrega-tt-req-sum

Cria internamente as temp-tables de requisições (tt-req-sum) e de alocações (tt-aloca), podendo as mesmas serem obtidas através da execução dos métodos pi-retorna-tt-req-sum e pi-retorna-tt-aloca. Para que a execução dessa procedure tenha êxito, a temp-table de transferência (tt-transf) deve ter sido previamente carregada.

 Macro desconhecida: 'rate'

[base\\_de\\_conhecimento](#) [mcp](#) [cpapi014](#)



[Política de  
privacidade](#)

[Termos  
de uso](#)