

# CPAPI017 - Requisição Avulsa de Materiais para Produção

 Tempo aproximado para leitura: 8 minutos

## Documento de API

<b>Produto:</b>	Datasul
<b>Ocorrência:</b>	Documentação de API
<b>Nome Físico</b>	cpp/cpapi017.p

**Objetivo:** Efetuar requisição de materiais para a produção, item a item, sendo responsável pela baixa na matéria-prima utilizada para a produção, gerando o movimento de material, atualizando as reservas da ordem e eventuais alocações.

**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 requisições. A sintaxe para a chamada é:

```
run cpp/cpapi017.p (input      table tt-requis,
                    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 requisição, devendo a temp-tables 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 requisição de materiais. 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/cpapi017.p persistent set h-handle (input      table tt-requis,
                                           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-requis – Temp-table contendo todas as informações necessárias para a requisição, como a quantidade, a data da transação e o número da ordem de produção sendo requisitada.
- tt-erro – Temp-table que retorna os erros e “avisos” retornados pela API durante o processamento dos movimentos.
- 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.

**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 de requisição são especificados na temp-table tt-erro.

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

- TT-REQUIS:

**Tipo-trans** – Define o tipo de transação que será executada. Pode valer 1 (requisição) ou 2 (devolução).

**Nr-ord-produ** – Deve conter o número da ordem de produção sendo requisitada.

**Quantidade** – Contém a quantidade que será requisitada.

**Data** – Data da transação da requisição/devolução.

**Procura-saldos** – Informa se a API irá realizar a busca dos saldos (FIFO) automaticamente (o default para execução batch é YES).

**It-codigo** – Informa o item que será requisitado.

**Item-pai** – Informa o item do nível acima do item a ser requisitado.

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

Existem alguns campos que não são obrigatórios, sendo preenchidos com valores default pela API:

**Tipo-trans** – O tipo de transação padrão é 1 (requisição).

**Data** – A data de transação padrão é TODAY.

**Cod-depos** – Se não for informado um depósito, e a procura automática de saldos estiver ativa, será feita a busca em todos os depósitos. Se a procura automática não for ativada, será utilizado o depósito da reserva ou, caso não exista reserva, o depósito padrão do item.

**Cod-localiz** – Se não for informada uma localização, e a procura automática de saldos estiver ativa, será feita a busca em todas as localizações. Se a procura automática de saldos não for ativada, será utilizada a localização da reserva, ou caso não exista, a localização branca (“”). Obs.: Se o item possuir localização única, será utilizada a localização do item, independentemente do parâmetro de busca automática de saldos.

**Quantidade** – Se não for informada uma quantidade, será utilizado o saldo da reserva, ou seja, a quantidade original da reserva menos a quantidade atendida.

**Item-pai** – Se o item pai da reserva não for informado, será assumido o item da ordem. Obs.: Para a requisição de reservas de itens “fantasmas” é necessário informar o item pai.

**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-requis (input table tt-requis)

Recebe 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-requis (output table tt-requis)

Retorna (através de um output parameter table) da API a temp-table correspondente.

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

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

pi-valida-ordem (input l-imediato,  
input i-nr-ord-prod,  
output c-erro,  
output c-texto)


Realiza validações relativas à ordem de produção a ser requisitada. O parâmetro l-imediato informa à procedure se ela deve retornar imediatamente após detectar o primeiro erro, ou se deve realizar todas as validações antes de retornar. O parâmetro i-nr-ord-prod informa o número da ordem de produção que será validada. 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”).

pi-valida-requis (input l-imediato,  
input i-nr-ord-prod,  
input i-seq,  
output c-erro,  
output c-texto)

Realiza todas as validações relativas ao movimento de requisição. Os parâmetros e o return-value funcionam da mesma forma que a procedure anterior. O parâmetro i-seq indica qual a sequência da temp-table tt-requis que será validada. Para executar essa procedure, é necessário que a API tenha recebido a temp-table de requisição (pi-recebe-tt-requis). Obs.: Essa procedure executa internamente a pi-valida-ordem (se o parâmetro l-imediato for NO).

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”.

 Macro desconhecida: 'rate'

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

[cpapi017](#)

[mcp](#)



[Política de  
privacidade](#)

[Termos  
de uso](#)