

# CPAPI013 - Alocação de Materiais para Produção

 Tempo aproximado para leitura: superior a 15 minutos

## Documento de API

<b>Produto:</b>	Datasul
<b>Ocorrência:</b>	Documentação de API
<b>Nome Físico</b>	cpp/cpapi013.p

**Objetivo:** Efetuar a alocação de materiais no sistema, direcionando o saldo de estoque das matérias primas diretamente para uma ordem de produção, impedindo que esse saldo seja utilizado por qualquer outro processo.

**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 alocações. A sintaxe para a chamada é:

```
run cpp/cpapi013.p (input      table tt-alocacao,  
                    input      table tt-deposito,  
                    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 alocação, 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 alocação. 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/cpapi013.p persistent set h-handle (input      table tt-alocacao,  
                                           input      table tt-deposito,  
                                           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-alocacao – Temp-table contendo as informações principais da alocação, tais como as ordens a serem alocadas, o tipo de transação (alocação/estorno), a forma de alocação que será utilizada etc.
- tt-deposito – Temp-table utilizada para conter os depósitos que deverão ser considerados no processo de alocação. Caso se parametrize para alocar todos os depósitos, essa temp-table pode estar vazia.
- tt-erro – Temp-table que retorna os erros e “avisos” retornados pela API durante o processamento das alocações.
- 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, é utilizada outra temp-table a fim de passar os parâmetros para os “métodos” da API. Essa temp-tables é:

- tt-reservas – Contém as informações relativas à alocação das reservas das ordens de produção envolvida no processo.

**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-ALOCACAO:**

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

**Sit-Aloc** – Determina o tipo de alocação que será utilizado (1 – Total, 2 – Parcial, 3 – Proporcional).

**Nr-ord-produ** – Contém o número da ordem de produção que será processada.

**Prioridade** - Determina quais os depósitos que terão prioridade durante o processo de alocação (1 – Depósitos de Processo, 2 – Depósito da Reserva, 3 – Aleatório). **Informa-Dep** – Determina se a temp-table tt-deposito será utilizada. Se for “NO”, serão utilizados todos os depósitos que permitem alocação. Se for “YES”, serão utilizados apenas os depósitos passados através da tt-deposito.

**Reaproveita** – Informa se após o processo de estorno da alocação, a quantidade disponibilizada será utilizada para complementar a alocação de outras ordens de produção.

**Aloca-Altern** – Determina se serão utilizados componentes alternativos para a alocação, caso a matéria-prima principal não possua saldo suficiente. Se for passado “?”, assume o existente nos parâmetros da produção (cp0106).

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

**Perc-Proporcional** – Retorna qual o percentual utilizado para alocação proporcional.

- **ttAlocaReserva:**

**nr-ord-produ** – Contém o número da ordem de produção que será processada.

**item-pai** - Código do Item pai.

**cod-roteiro** - Contém o código do roteiro.

**op-codigo** - Contém o código da operação.

**it-codigo** - Código do item.

**cod-refer** - Contém a referência utilizada.

**cod-estabel** - Estabelecimento.

**cod-depos** - Depósito

**cod-localiz** - Localização.

**lote-serie** - Lote

**quant-aloc** - Contém a quantidade alocada.

**qt-aloc-lote** - Usado somente quando ativo PPM.

**estado** - (0- Display, 1- Inclui, 2- Modifica, 3- Elimina).

**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-alocacao (input table tt-alocacao)

pi-recebe-tt-reservas (input table tt-reservas)

pi-recebe-tt-deposito (input table tt-deposito)

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-alocacao (output table tt-alocacao)

pi-retorna-tt-reservas (output table tt-reservas)

pi-retorna-tt-deposito (output table tt-deposito)

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

run pi-processa-alocacoes (input table tt-alocacao,

```
input      table tt-deposito,  
input-output table tt-erro,  
input      l-deleta-erros).
```

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

```
pi-valida-alocacao (input l-imediato,  
input i-nr-ord-prod,  
input i-tipo-trans,  
input c-origem,  
output c-erro,  
output c-texto)
```

Realiza todas as validações relativas ao processo de alocação. 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 alocação (pi-recebe-tt-alocacao). O parâmetro l-imediato determina se a procedure retornará após o primeiro erro ou se irá acumular todos os erros antes de retornar. O campo c-origem indica qual o módulo chamador da procedure, podendo ser “CP” ou “MI”, representando respectivamente o módulo de produção e o de manutenção industrial. O campo i-tipo-trans determina se é uma alocação (1) ou um estorno (2).

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

Realiza a validação das informações da ordem de produção utilizada para a alocação. Internamente é chamada pela rotina pi-valida-alocacao. Os parâmetros funcionam de forma idêntica aos utilizado na procedure anterior.

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-carrega-tt-reservas (input i-sit-aloc,  
input i-nr-ord-prod,  
input i-prioridade,  
input l-lock,  
input l-aloca-altern,  
output de-perc,  
output c-erro,  
output c-texto)
```

Contém o algoritmo principal do processo de alocação. Gera a temp-table tt-reservas, contendo as informações relativas à alocação de cada matéria-prima da ordem, com quantidade alocada, lote, depósito, localização etc. O parâmetro i-sit-aloc determina o tipo de alocação (1 – Total, 2 – Parcial, 3 – Proporcional). O campo i-prioridade determina qual a prioridade que será aplicada aos depósitos (1 – Depósitos de Processo, 2 – Depósitos das Reservas, 3 – Aleatório). O parâmetro l-lock determina se o saldo de estoque será “travado” após a execução dessa rotina ou se as leituras do saldo serão feitas em “NO-LOCK”, sem travamento de registros. O parâmetro l-aloca-altern, indica se a rotina irá procurar por componentes alternativos caso os principais não possuam saldo. O campo de-perc retorna o percentual que a rotina utilizou para a alocação proporcional. Finalmente, os campos c-erro e c-texto retornam os erros eventualmente ocorridos durante o processo. Caso a rotina não tenha sido bem sucedida, o return-value conterá “NOK”.

pi-cria-tt-deposito

Realiza a geração dos registros da temp-table tt-deposito.

```
pi-valida-reserva (input i-nr-ord-prod,  
input i-seq,  
input l-imediato,  
input de-perc,
```

output c-erro,

output c-texto)

Valida um registro da temp-table de reservas, localizado através dos parâmetros i-nr-ord-prod e i-seq, que contém respectivamente o número da ordem e a sequência da reserva da ordem, sendo que esses campos constituem a chave primária e única da temp-table de reservas. O campo de-perc contém o percentual que a rotina utiliza para a alocação proporcional. 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 reservas já esteja carregada na instância da API.

pi-aloca-manual (input table ttAlocaReserva,  
input-output table tt-erro)

Procedure de alocação manual, responsável pelo processamento de todos os movimentos relativos à alocação. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, a alocação não foi realizada.

pi-mod-aloca-manual (input p-aloca-reserva,  
input p-quant)

Procedure de alteração de alocação manual, responsável pela modificação na quantidade de alocação manual. O campo p-aloca-reserva contém o rowid do registro a ser alterado na tabela aloca-reserva. O campo p-quant indicará a nova quantidade. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, a alocação não foi alterada.

pi-transfere-aloc (input p-aloca-reserva,  
input p-transf,  
input p-new-cod-depos,  
input p-new-cod-localiz,  
input-output table tt-erro)

Procedure de transferência de alocação, responsável pela transferência de quantidade entre depósitos e localizações de alocação. O campo p-aloca-reserva contém o rowid do registro a ser alterado na tabela aloca-reserva. O campo p-transf indicará a quantidade a ser transferida. Os campos p-new-cod-depos e p-new-cod-localiz contém respectivamente o depósito e localização de destino deste saldo no estoque. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, a alocação não foi alterada.

pi-recalcula-aloc (input p-rowid)

Procedure de recálculo de alocação, responsável por recalcular as quantidades das alocações. O campo p-rowid indica o rowid do registro na tabela reservas. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, o processo não foi concluído.

pi-elimina-aloca-manual (input p-aloca-reserva)

Procedure de Eliminação alocação, responsável por recalcular as quantidades das alocações. O campo p-aloca-reserva indica o rowid do registro na tabela aloca-reserva. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, o processo não foi concluído.

pi-atualiza-estado-ordem (input p-nr-ord-produ)

Procedure de atualização de estado de ordens, responsável por ajustar o campo estado da ordem de produção baseando-se pelos movimentos de estoque, alocações, ordens de manutenção e operações. O campo p-nr-ord-produ indica o rowid do registro na tabela ord-prod. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, o processo não foi concluído.

pi-desaloca-reserva (input p-reserva)

Procedure de atualização de estado de ordens. O campo p-reserva indica o rowid do registro na tabela reservas. Através do return-value verifica-se a execução da procedure. Se for “OK”, o processo foi realizado com sucesso. Se for “NOK”, o processo não foi concluído.

## Considerações Gerais:

**Tipos de Alocação** – Os tipos de alocação que podem ser realizados pela API são:

Total – Apenas aloca a ordem de produção caso todos os componentes possuam saldo suficiente. Se apenas um dos componentes não possuir saldo, a alocação não será realizada.

Parcial – Realiza a alocação dos componentes que possuam saldo. Caso alguma matéria-prima não possua saldo em estoque suficiente, ele será alocado até o limite do saldo disponível e o estado da ordem será alterado para “Alocada”.

Proporcional – Realiza uma verificação inicial nos saldos dos componentes, determinando qual o menor saldo disponível. Uma vez determinado esse saldo, é calculada a proporcionalidade entre a necessidade da matéria-prima e o saldo disponível. Esse percentual é utilizado para alocar todos os demais componentes da ordem.

**Alocação de Alternativos** – No módulo de engenharia, é possível o cadastramento de itens alternativos para uma relação na estrutura. Caso esse alternativo tenha sido setado para permitir alocação e o parâmetro de alocação de alternativos na API for passado como “YES”, esse item será alocado no lugar do componente principal da estrutura caso esse componente não possua saldo.

É possível o cadastramento de diversos alternativos para um componente. A API irá utilizar todos eles, seguindo a ordem determinada na engenharia, até que a quantidade necessária para a alocação seja atingida. A API irá respeitar a relação entre a quantidade do componente original e a do alternativo cadastrado na engenharia e irá permitir a utilização de alternativos nos três tipos disponíveis de alocação. Uma vez confirmada a alocação com alternativos, serão criadas reservas na ordem de produção para esses componentes e a quantidade da reserva do item principal será reduzida proporcionalmente.

 Macro desconhecida: 'rate'

base\_de\_conhecimento

mcp

cpapi013



Política de  
privacidade

Termos  
de uso