# Customização de operandos de faturamento de serviço

# TOTVS Logística WMS – Linha Logix

09/07/2020

# Sumário




Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

# CUSTOMIZAÇÃO DE OPERANDOS DE FATURAMENTO DE SERVIÇO

## 1. Objetivo

Desenvolver uma documentação complementar do Cadastro de Operando de Faturamento (WMS6401), detalhando principalmente os campos que compõem a aba **2- Customizações.**

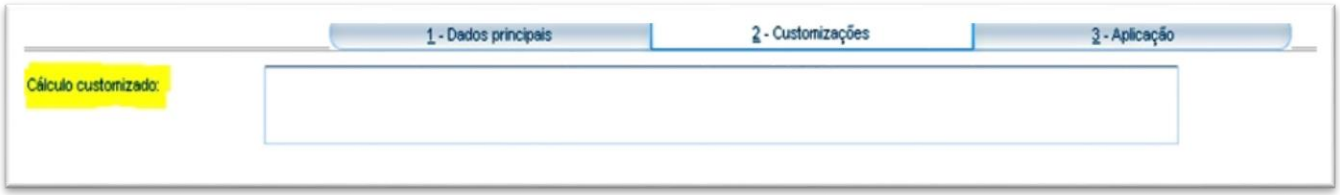

Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

# 2. Aba 2 – Customizações

## 2.1. Cálculo customizado

Neste campo é possível efetuar a customização do cálculo, sendo necessário informar o SQL com o comando SELECT.

O comando do tipo **SQL** deve retornar obrigatoriamente um único valor, pois o Sistema está preparado para trabalhar apenas com o resultado final a ser retornado por este comando, sendo que atualmente este campo comporta até **255 caracteres.**




**Exemplo**

```
SELECT SUM(minha_tabela.meu_campo)
SELECT COUNT(m.campo_nao_padrao)
SELECT MAX(d.campo_nao_padrao)
SELECT AVG(m.qtd_volumes) #Média de volumes expedição por dia. m -> wms_exp_mestre_?
```

O campo **Cálculo Customizado** somente tem utilidade quando na aba **1 - Dados Principais**, o campo **Tipo Cálculo** estiver configurado como **Customizado pelo usuário,** conforme especificado a seguir:


## 2.2. Comando customizado

Neste campo é possível cadastrar uma função que pode manipular os comandos SQL (**SELECT, FROM** e **WHERE**) padrões antes da execução na função padrão de apuração, sendo que esta função será executada antes do **PREPARE** do SQL utilizado para efetuar a leitura do valor calculado do operando no período, atuando como um ***before_prepare.***

Esta função vai receber por **SET** as variáveis cmd_select , cmd_from e cmd_where que estarão com os valores padrões como, por exemplo:

```
cmd_select = "SELECT SUM(m.qtd_palete_fisico)"
cmd_from = " FROM wms_armaz_end_08345690901 m" onde 08345690901 é código do depositante.
cmd_where = " WHERE m.empresa = '01' AND m.sit_registro <> 'C' AND m.data_posicao_estoque >= ?
AND m.data_posicao_estoque <= ?"
```

A função de apuração chamará a função cadastrada no campo **Comando Customizado**, podendo editar e retornar estes comandos que serão recebidos por funções de **GET** conforme abaixo:

LET l_select_stmt = LOG_getVar("cmd_select") Máximo 255 caracteres
LET l_from_stmt = LOG_getVar("cmd_from") Máximo 1000 caracteres
LET l_where_stmt = LOG_getVar("cmd_where") Máximo 5000 caracteres

Deve obrigatoriamente retornar **TRUE** ou **FALSE.** Nesta função é possível, por exemplo, alterar o comando **FROM** que a função padrão executará, informando uma tabela específica ou adicionar um filtro no comando **WHERE.**


**Exemplo básico**

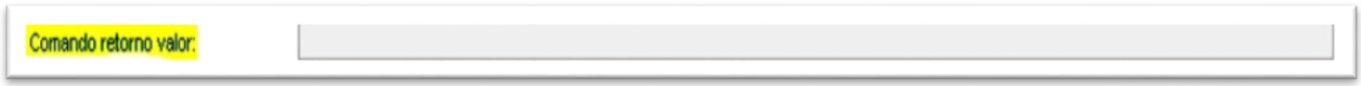
```
FUNCTION wmsyXXXX_customiza_operando()
DEFINE l_cmd_from CHAR(1000)

LET l_cmd_from = LOG_getVar("cmd_from")

# 'm' representa o alias da tabela mestre do processo
LET l_cmd_from = l_cmd_from CLIPPED,
" INNER JOIN minha_tabela",
" ON minha_tabela.empresa = m.empresa",
" AND minha_tabela.ctr_ent_sai_veic_docum = m.ctr_ent_sai_veic_docum"

CALL LOG_setVar("cmd_from", l_cmd_from)

RETURN TRUE
END FUNCTION
```

## 2.3. Comando retorno valor

Neste campo é possível cadastrar uma função que pode utilizar o valor calculado para um determinado operando para, por exemplo, gravar uma tabela auxiliar específica.

Esta função não manipula o valor calculado, recebendo o valor calculado do operando para cada dia do período pela função **setVar:**

CALL LOG_setVar("valor_operando",l_valor_operando)





Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

**Exemplo**

```
# Comando para ser executado após o calculo do valor do operando
# e a gravação da tabela padrão wms_geracao_dados_operando
FUNCTION wmsyXXXX_comando_retorno_valor()

DEFINE l_depositante        LIKE wms_geracao_dados_operando.depositante,
l_operando_faturamento   LIKE wms_geracao_dados_operando.operando_faturamento,
l_data_geracao              LIKE wms_geracao_dados_operando.dat_geracao,
l_val_parametro             LIKE wms_geracao_dados_operando.val_parametro

LET l_depositante = wmsr48_get_depositante()
LET l_operando_faturamento = wmsr48_get_operando_faturamento()
LET l_data_geracao = wmsr48_get_data_atual_processamento()
LET l_val_parametro = LOG_getVar("valor_operando")

INSERT INTO minha_tabela (depositante, operando, data_geracao, valor_operando)
VALUES (l_depositante, l_operando_faturamento, l_data_geracao, l_val_parametro)

RETURN TRUE
END FUNCTION
```

## 2.4. Filtro customizado


Com esse campo é possível gerar filtros adicionais nas tabelas padrão de dados base como, por exemplo:

- Cobrar recebimento de um veículo de determinada placa de forma diferenciada:


- Cobrar a armazenagem de forma diferenciada de endereços com determinada função de endereço:


Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

# CUSTOMIZAÇÃO DE OPERANDOS DE FATURAMENTO DE SERVIÇO


Existem algumas funções padrões que retornam valores que podem ser usadas para configurar parâmetros customizados para aplicar filtros em um operando de faturamento:

| Função | Descrição | Tipo |
|---|---|---|
| **wmsr48_get_empresa** | Retorna a empresa corrente. | CHAR(02) |
| **wmsr48_get_seq_apuracao_faturamento** | Retorna um sequencial que corresponde ao processo de faturamento cadastrado na proposta de faturamento do depositante, ao qual o operando processado está associado. | INTEGER |
| **wmsr48_get_seq_processo_faturamento** | Retorna um sequencial que corresponde ao processo de faturamento cadastrado na proposta de faturamento do depositante, ao qual o operando processado está associado. | INTEGER |
| **wmsr48_get_depositante** | Retorna o depositante atual para o qual está sendo processada a apuração de faturamento. | CHAR(15) |
| **wmsr48_get_operando_faturamento** | Retorna o operando de faturamento atual que está sendo processado. | CHAR(20) |
| **wmsr48_get_data_ini_processamento** | Retorna a data inicial do período de apuração que está sendo processada.<br><br>**Exemplo**<br>Se o processo é mensal retornará o dia 01/MM/AAAA. | DATE |
| **wmsr48_get_data_fim_processamento** | Retorna a data final do período de apuração que está sendo processado.<br><br>**Exemplo**<br>Se o processo é mensal retornará o último dia do mês DD/MM/AAAA. | DATE |
| **wmsr48_get_data_hora_ini_processamento** | Igual a data inicial, porém, retorna a data/hora.<br><br>**Exemplo**<br>01/MM/AAAA 00:00:00 | DATETIME YEAR TO SECOND |
| **wmsr48_get_data_hora_fim_processamento** | Igual a data final, porém, retorna a data/hora. | DATETIME YEAR TO SECOND |

| | | |
|---|---|---|
| | **Exemplo**<br>DD/MM/AAAA 23:59:59 | |
| **wmsr48_get_data_atual_processamento** | Retorna a data atual do processamento no período de apuração.<br><br>**Exemplo**<br>Se o processamento for **mensal** e estiver processando o dia 5 do mês 05/MM/AAAA. | DATE |
| **wmsr48_get_data_hora_atual_ini_processamento** | Retorna a data/hora atual do processamento no período de apuração. A hora sempre considerará o dia completo começando às 00:00:00h e terminando às 23:59:59h.<br><br>**Exemplo**<br>Se o processamento for **mensal** e estiver processando o dia 5 do mês 05/MM/AAAA 00:00:00. | DATETIME YEAR TO SECOND |
| **wmsr48_get_data_hora_atual_fim_processamento** | Retorna a data/hora atual do processamento no período de apuração, sendo que a hora sempre considererará o dia completo começando às 00:00:00h e terminando às 23:59:59h.<br><br>**Exemplo**<br>Se o processamento for **mensal** e estiver processando o dia 5 do mês 05/MM/AAAA 23:59:59. | DATETIME YEAR TO SECOND |

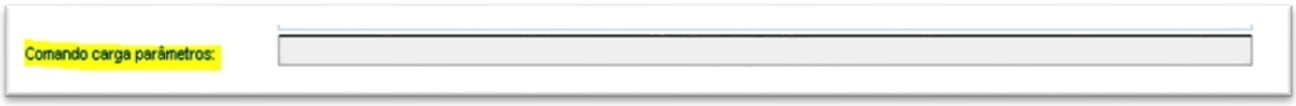
**Exemplo de filtro utilizando parametrização customizada**

```
AND minha_tabela.meu_depositante = ?
AND minha_tabela.meu_campo_data_hora >= ?
AND minha_tabela.meu_campo_data_hora <= ?
AND minha_tabela.meu_campo_customizado = ?
```

| Sequência | Função | Tipo Retorno | Tamanho | Precisão |
|---|---|---|---|---|
| 1 | wmsr48_get_depositante | CHAR | 15 | 0 |
| | wmsr48_get_data_hora_atual_ini_processamento | DATETIME | 1 | 6 |
| 3 | wmsr48_get_data_hora_atual_fim_processamento | DATETIME | 1 | 6 |
| 4 | wmsyXXXX_minha_funcao_customizada | INTEGER | 10 | 0 |


Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

## 2.5. Comando caga parâmetros




É especificada uma função customizada, que será executa antes do **EXECUTE** do SQL usado para fazer a leitura do valor calculado do operando no período, atuando como um ***before_execute*** do operando.

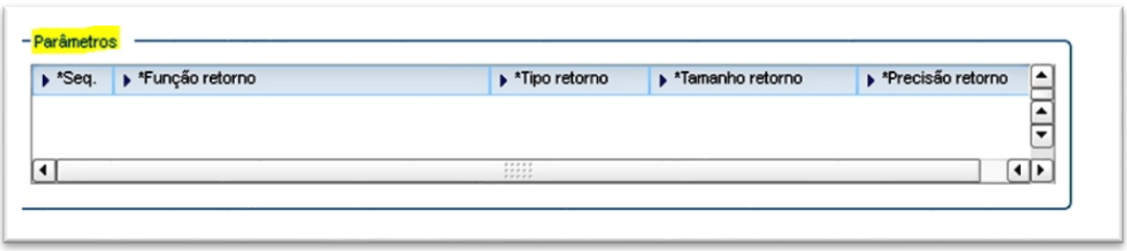
- Não possui parâmetros que podem ser recuperados pela função **LOG_getVar().**
- Pode ser utilizada para carregar os valores retornados pelas funções dos parâmetros customizados
- Deve obrigatoriamente retornar **TRUE** ou **FALSE.**

a. Esta função não recebe nenhum retorno específico via **LOG_setVar().**

b. Pode ser utilizada para fazer a carga de variáveis customizadas para o operando que serão utilizadas de forma dinâmica, de acordo com o cadastro de parâmetros customizados.

A função especificada customizada, fará o processamento e tratamento de datas de período início e fim, de forma que possam ser recuperadas por variáveis modulares ou tabelas temporárias, através de outra função que deverá ser informada na grade **Parâmetros.**

## 2.6. Grade parâmetros




| Identificação | Descrição |
|---|---|
| Função Retorno | Função 4GL que retornará o valor do parâmetro. |
| Tipo Retorno | Tipo de retorno da função retorno: **Char, Smallint, Integer, Decimal, Date, Datatime, Varchar.** |
| Tamanho Retorno | Tamanho a ser determinado para o retorno, para parâmetros com tipo de retorno **Char, Decimal e Varchar.** |
| Precisão Retorno | Número de casas decimais que determinará a precisão no retorno para parâmetros com tipo retorno **Decimal.** |


Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

# 3. Informações adicionais sobre os comandos customizados

Gerar funções customizadas que podem ser utilizadas para o Comando customizado, Comando retorno valor ou Comando carga parâmetro requer alguns padrões a serem seguidos, para que a execução da função ocorra da forma desejada e apresente o resultado esperado.

Desta forma, segue um detalhamento sobre o modo trabalho e comportamento de cada tipo de função para cada comando.

## 3.1. Campo customizado

Campo para informar o nome da função na qual deverá ser efetuada a manipulação do SQL que deverá determinar o valor do operando.

Esta função será executada antes do **PREPARE** do SQL utilizado para efetuar a leitura do valor calculado do operando no período, atuando como um ***before_prepare.***

- Possui parâmetros que podem ser recuperados pela função **LOG_getVar().**

| Comando | Descrição |
|---|---|
| cmd_select | Comando SQL do tipo **SELECT,** utilizado para determinar o valor do operando. |
| cmd_from | Comando SQL do tipo **FROM**, que indica qual tabela será utilizada para executar o **SELECT** |
| cmd_where | Comando SQL do tipo WHERE, que indica quais filtros serão utilizados para determinar o valor do operando. |

- Recebe como parâmetros que podem ser setados pela função **LOG_setVar().**

| Comando | Descrição |
|---|---|
| cmd_select | Comando SQL do tipo **SELECT,** utilizado para determinar o valor do operando. |
| cmd_from | Comando SQL do tipo **FROM**, que indica qual tabela será utilizada para executar o **SELECT**. |
| cmd_where | Comando SQL do tipo **WHERE**, que indica quais filtros serão utilizados para determinar o valor do operando. |

- Deve obrigatoriamente retornar **TRUE** ou **FALSE**


Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

## 3.2. Gerando uma função customizada

Para gerar uma função customizada que poderá ser utilizada no **Comando Customizado** devem ser observados alguns padrões, pois dentro da rotina o comando SQL deve ser quebrado em três partes: **SELECT, FROM** e **WHERE**, conforme descrito a seguir:

- Definir as variáveis **l_select_stmt, l_from_stmt ,l_where_stmt** com os respectivos tamanhos a seguir:

```
DEFINE l_select_stmt    VARCHAR(255)
       l_from_stmt      VARCHAR(1000)
       l_where_stmt     VARCHAR(5000)
```

- Definir **Record** para receber as informações do operando:

```
DEFINE lr_operand_fatur    RECORD
                    tip_processo            CHAR(20)            , tip_operando_faturamento SMALLINT
                    ,dat_base_operando      SMALLINT
                    ,filtro_customizado     LIKE wms_operando_faturamento.filtro_customizado
                    ,forma_cobranca         LIKE wms_operando_faturamento.forma_cobranca
                   END RECORD

DEFINE l_depositante CHAR(15)
LET l_depositante = wmsr48_get_depositante()
```

- Utilizar o código do depositante para montar os nomes das tabelas no **SELECT,** pois as informações estão divididas em tabelas com prefixo padrão, porém, com código do depositante compondo o nome. Mais detalhes podem ser obtidos a seguir, em **Tabelas e Alias.**

- Utilizar a função abaixo para obter as informações do operando de faturamento:

```
LET l_operando_faturamento = wmsr48_get_operando_faturamento()
```

- Utilizar a função **LOG_getVar(" ")** para carregar os comandos de **SELECT, FROM** e **WHERE**:

```
LET l_select_stmt = LOG_getVar("cmd_select")
LET l_from_stmt = LOG_getVar("cmd_from")
LET l_where_stmt = LOG_getVar("cmd_where")
```

- Inicializar as variáveis **l_select_stmt, l_from_stmt ,l_where_stmt:**

```
INITIALIZE l_select_stmt
,l_from_stmt
,l_where_stmt TO NULL
```

Para montar a estrutura da sua função, a variável **l_select_stmt** deve conter um comando SQL que retorne um valor único. Neste caso, podem ser utilizadas as funções **SUM(), MAX()** e **AVG(),** por exemplo.


Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

# CUSTOMIZAÇÃO DE OPERANDOS DE FATURAMENTO DE SERVIÇO

Também pode ser efetuada a leitura do campo **Forma Cobrança,** para que seja realizado um tratamento pelo Cadastro do Operando, sendo que para isso deve utilizado como por exemplo:

```
IF lr_operand_fatur.tip_processo = "ARMAZENAGEM" THEN
   IF lr_operand_fatur.tip_operando_faturamento = 99 THEN #Customizado pelo Usuário
      CASE lr_operand_fatur.forma_cobranca
         WHEN "S"
            LET l_select_stmt = "SELECT SUM(palete.qtd_saldo)"
         WHEN "P"
            LET l_select_stmt = "SELECT MAX(palete.qtd_saldo)"
         WHEN "M"
            LET l_select_stmt = "SELECT AVG(palete.qtd_saldo)"
       END CASE
```

- Montar cada estrutura separadamente, em sua respectiva variável:

**Select – l_select_stmt**

```
LET l_select_stmt = "SELECT SUM(b.qtd_saldo/palete.qtd_saldo)"
```

**From – l_from_stmt**

```
LET l_from_stmt =    " FROM wms_armaz_pal_", l_depositante CLIPPED, " b",
                     "          ,(SELECT DISTINCT a.empresa ",
                     "                           ,a.data_posicao_estoque ",
                     "                           ,a.palete ",
                     "                           ,SUM(a.qtd_saldo) qtd_saldo ",
                     "              FROM wms_armaz_pal_", l_depositante CLIPPED, " a",
                     "             WHERE a.empresa = '",l_empresa ,"'",
                     "               AND a.data_posicao_estoque >= ? ",
                     "               AND a.data_posicao_estoque <= ? ",
                     "             GROUP BY a.empresa ",
                     "                      ,a.data_posicao_estoque ",
                     "                      ,a.palete) palete "
```

**Where = l_where_stmt**

```
LET l_where_stmt =  " WHERE b.empresa = palete.empresa ",
                    " AND b.data_posicao_estoque = palete.data_posicao_estoque ",
                    " AND b.palete = palete.palete ",
                    " AND EXISTS(SELECT DISTINCT 1 ",
                    "              FROM wms_armaz_item_", l_depositante CLIPPED, " c",
                    "             WHERE c.empresa = b.empresa ",
                    "               AND c.data_posicao_estoque = b.data_posicao_estoque ",
                    "               AND c.item = b.item ",
                    "               AND c.sit_registro <> 'C' "
```

- E chamar a função **Log_setVar(),** passando os parâmetros, como **cmd_select, cmd_from** e **cmd_where**, respectivamente com suas variáveis, para que seja efetuado devido carregamento, e então o **Return True** indicará para a aplicação que esta poderá seguir com o **prepare execute** do comando SQL.


Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

```
CALL LOG_setVar("cmd_select",l_select_stmt)
CALL LOG_setVar("cmd_from", l_from_stmt)
CALL LOG_setVar("cmd_where", l_where_stmt)

RETURN TRUE

END FUNCTION
```

## 3.3. Tabelas e alias padrões

Para facilitar o entendimento do SQL foi assumido por padrão que o **ALIAS "m"** indica a tabela mestre e **"d"** a tabela detalhe. Este padrão deverá ser seguido.

**Montagem do SELECT**

- Processo: **RECEBIMENTO**
  LET l_from_stmt = " FROM wms_rec_mestre_", l_depositante CLIPPED, " m"
- Processo: **EXPEDICAO**
  LET l_from_stmt = " FROM wms_exp_mestre_", l_depositante CLIPPED, " m"
- Processo: **ARMAZENAGEM**
  LET l_from_stmt = " FROM wms_armaz_end_", l_depositante CLIPPED, " m"
- Processo: **SEGURO**
  LET l_from_stmt = " FROM wms_seg_detail_", l_depositante CLIPPED, " d"
- Processo: **ATRIBUTO**
  LET l_from_stmt = " FROM wms_atr_mestre_", l_depositante CLIPPED, " m"


Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©

# 4. Assuntos relacionados

**Documento de Referência**

- Operando - WMS6401


Este documento é propriedade da TOTVS. Todos os direitos reservados. ©
Este documento é propriedade da TOTVS. Todos os direitos reservados. ©