

# **Integração Loja Virtual**

**entre dados do MicrovixERP e sistemas de terceiros**

**Layout e formato de envio de dados**

**- Manual de Integração -**

## 1. Índice

1. Índice .....	2
2. Introdução .....	4
3. Resumo geral da integração.....	4
4. Fornecimento das informações.....	5
5. Layout das tabelas.....	6
5.1. b2c_empresas .....	6
5.2. formas_pgto .....	6
5.3. planos .....	6
5.4. planos_parcelas.....	7
5.5. produtos_setores .....	7
5.6. produtos_linhas.....	7
5.7. produtos_marcas .....	8
5.8. produtos_colecoes .....	8
5.9. produtos_espessuras .....	8
5.10. produtos_grade1.....	8
5.11. produtos_grade2.....	8
5.12. produtos_unidade.....	8
5.13. produtos_classificacoes .....	8
5.14. produtos .....	9
5.15. produtos_detalhes .....	10
5.16. produtos_custos.....	10
5.17. produtos_campanhas.....	11
5.18. produtos_promocao.....	12
5.19. produtos_codebar .....	12
5.20. produtos_informacoes .....	13
5.21. produtos_dimensoes.....	13
5.22. imagens .....	13
5.23. produtos_imagens.....	13
5.24. produtos_associados.....	14
5.25. pedido.....	14

5.26.	pedido_item .....	15
5.27.	clientes .....	15
5.28.	clientes_enderecos_entrega .....	17
5.29.	com_config_encomenda .....	18
5.30.	produtos_campos_adicionais_nomes.....	18
5.31.	produtos_campos_adicionais_detalhes.....	19
5.32.	produtos_campos_adicionais_valores.....	19
5.33.	data_ultima_atualizacao .....	19
5.34.	status .....	19
5.35.	pedido_status.....	20
5.36.	com_config_encomenda .....	20
5.37.	fornecedores .....	20
5.38.	produtos_tabelas .....	21
5.39.	produtos_tabelas_precos.....	21
5.40.	transportadores.....	21
5.41.	tipos_cobranca_frete .....	22
5.42.	clientes_saldo.....	22
5.43.	produtos_depositos .....	22
5.44.	produtos_detalhes_depositos .....	23
5.45.	clientes_creditos .....	23
5.46.	vendedores.....	23
5.47.	nfe.....	24
5.48.	nfe_situacao .....	24
6.	Descrição das tabelas .....	25
7.	Script de criação da estrutura .....	28

## 2. Introdução

Este manual tem por objetivo detalhar o layout das tabelas de banco de dados para a integração entre o MicrovixERP e lojas virtuais de terceiros, bem como esclarecer as funcionalidades compatíveis com a plataforma.

Os dados cadastrais necessários para a realização de um pedido serão fornecidos pelo ERP, já os dados referentes aos pedidos e clientes serão fornecidos pela loja virtual.

Todas as informações serão disponibilizadas em uma base de dados no SQL SERVER 2008 R2, que poderão ser acessadas com um usuário específico para a função.

## 3. Resumo geral da integração

Os dados envolvidos na integração serão:

1. Dados das empresas;
2. Planos de pagamento;
3. Clientes;
4. Produtos;
5. Cadastros auxiliares de produtos;
6. Saldos;
7. Promoções;
8. Pedidos;
9. Fornecedores.

Toda a integração ocorrerá em uma base intermediária entre a loja virtual do terceiro e a base do MicrovixERP. Esta base ficará em um servidor próprio da Linx – Microvix e o fornecedor da loja virtual terá acesso para inserir dados em determinadas tabelas e permissão de leitura em todas as tabelas. O modelo e estrutura desta base permanecerão inalterados por padrão, e caso ocorra algum ajuste necessário que influencie no funcionamento geral (que só poderá ser feito pela Linx – Microvix), o mesmo será reportado para a empresa fornecedora da loja virtual.

A interação entre esta base e o MicrovixERP será de total responsabilidade da Linx – Microvix, já a interação entre esta base e a loja virtual, será de total responsabilidade do fornecedor do produto.

A disponibilidade desta base de dados intermediária é de inteira responsabilidade da Linx – Microvix. Um usuário de acesso será fornecido com todas as permissões necessárias para a realização da integração e deverá ser usado em cada interação.

#### 4. Fornecimento das informações

Cada uma das partes será responsável por enviar determinadas informações que, ao serem unidas, gerem uma comunicação uniforme e transparente entre dois sistemas distintos.

Será responsabilidade da **Linx – Microvix** a inserção e atualização dos dados referentes à:

1. Dados da empresa;
2. Planos de pagamento;
3. Produtos;
4. Cadastros auxiliares de produtos;
5. Saldos;
6. Promoções;
7. Fornecedores.

Lembrando que todas estas informações serão extraídas diretamente do MicrovixERP, onde qualquer alteração feita no mesmo em qualquer um destes itens, impactará diretamente na base intermediária.

As regras de atualização bem como a periodicidade dos envios também serão definidas pela Linx – Microvix no que se diz respeito aos itens citados.

Será de responsabilidade do **fornecedor da loja virtual** a inserção e atualização dos dados referentes à:

1. Clientes;
2. Pedidos.

O envio das informações seguirá as regras do fornecedor, porém essas informações deverão adaptar-se à estrutura já criada na base intermediária de recebimento. Caso não seja possível atender os requisitos que a estrutura fornece, a integração não poderá ser concluída.

A periodicidade de envio das informações para a base intermediária também ficará a critério do fornecedor, porém a atualização do MicrovixERP com as informações em questão obedecerá uma regra criada pela Linx – Microvix.

## 5. Layout das tabelas

### 5.1. b2c\_empresas

Nome	Obrigatório	Tipo	Descrição	Exemplo
empresa	Sim	INT	Código da empresa no MicrovixERP	1
nome_emp	Não	VARCHAR(50)	Nome da empresa no MicrovixERP	'01 – LOJA SHOPPING FANTASIA'
cnpj_emp	Não	VARCHAR(14)	CNPJ da empresa no MicrovixERP	'12345678912345'
end_unidade	Não	VARCHAR(250)	Endereço da empresa no MicrovixERP	'Avenida das alamedas nº 467'
complemento_end_unidade	Não	VARCHAR(60)	Complemento do endereço da empresa no MicrovixERP	'Fundos'
nr_rua_unidade	Não	VARCHAR(20)	Número da rua da empresa no MicrovixERP	'467'
bairro_unidade	Não	VARCHAR(60)	Bairro da empresa no MicrovixERP	'Boa Vista'
cep_unidade	Não	VARCHAR(9)	CEP da empresa no MicrovixERP	'89234-091'
cidade_unidade	Não	VARCHAR(50)	Cidade da empresa no MicrovixERP	'Joinville'
uf_unidade	Não	CHAR(2)	Estado da empresa no MicrovixERP	'SC'
fone_unidade	Não	VARCHAR(70)	Telefone da empresa no MicrovixERP	'3422-7554'
email_unidade	Não	VARCHAR(50)	E-mail da empresa no MicrovixERP	'empresa@empresa.com.br'

### 5.2. formas\_pgto

Nome	Obrigatório	Tipo	Descrição	Exemplo
cod_forma_pgto	Sim	INT	Código da forma de pagamento	1
forma_pgto	Sim	VARCHAR(50)	Descrição da forma de pagamento	'Dinheiro'

### 5.3. planos

Nome	Obrigatório	Tipo	Descrição	Exemplo
plano	Sim	INT	Código do plano de pagamento	1

nome_plano	Sim	VARCHAR(30)	Nome do plano de pagamento	'30 vezes'
forma_pagamento	Sim	INT	Forma de pagamento padrão do plano de pagamento	1
qtde_parcelas	Sim	INT	Quantidade de parcelas para pagamento	5
valor_minimo_parcela	Não	MONEY	Valor mínimo de cada parcela do plano de pagamento	149.99
indice	Não	MONEY	Índice de acréscimo ou desconto para calcular o valor de venda dos produtos do pedido ao utilizar o plano	0,95

#### 5.4. planos\_parcelas

Nome	Obrigatório	Tipo	Descrição	Exemplo
plano	Sim	INT	Código do plano de pagamento	1
ordem_parcela	Sim	INT	Ordem de pagamento de cada parcela	5
prazo_parc	Sim	INT	Prazo de pagamento entre as parcelas em dias	30
id_planos_parcelas	Sim	INT	Código da parcela do plano de pagamento	18

#### 5.5. produtos\_setores

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_setor	Sim	INT	Código do setor	3
nome_setor	Sim	VARCHAR(100)	Descrição do setor	'BLUSAS'

#### 5.6. produtos\_linhas

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_linha	Sim	INT	Código da linha	12
nome_linha	Sim	VARCHAR(30)	Descrição da linha	'Infantil'

### 5.7. produtos\_marcas

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_marca	Sim	INT	Código do marca	1
nome_marca	Sim	VARCHAR(100)	Descrição da marca	'FAKE'

### 5.8. produtos\_colecoes

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_colecao	Sim	INT	Código da coleção	8
nome_colecao	Sim	VARCHAR(100)	Descrição da coleção	'INVERNO 2012'

### 5.9. produtos\_espessuras

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_espessura	Sim	INT	Código da espessura	22
nome_espessura	Sim	VARCHAR(100)	Descrição da espessura	'FIO'

### 5.10. produtos\_grade1

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_grade1	Sim	INT	Código do tamanho	7
nome_grade1	Sim	VARCHAR(100)	Descrição do tamanho	'P'

### 5.11. produtos\_grade2

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_grade2	Sim	INT	Código da cor	3
nome_grade2	Sim	VARCHAR(100)	Descrição da cor	'AZUL'

### 5.12. produtos\_unidade

Nome	Obrigatório	Tipo	Descrição	Exemplo
unidade	Sim	VARCHAR(50)	Descrição da unidade	'PC'

### 5.13. produtos\_classificacoes

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_classificacao	Sim	INT	Código da classificação	2
nome_classificacao	Sim	VARCHAR(50)	Nome da classificação	'LEGAL'

### 5.14. produtos

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigoproduto	Sim	BIGINT	Código do produto no MicrovixERP (sku).	12837
referencia	Não	VARCHAR(20)	Código de referência do produto agrupado.	'1403EF'
codauxiliar1	Não	VARCHAR(40)	Código auxiliar do produto.	'ant.0012'
descricao_basica	Não	VARCHAR(100)	Descrição resumida do produto.	'Bota MX'
nome_produto	Não	VARCHAR(250)	Descrição completa do produto.	'Bota MX 1403EF verde 37'
peso_liquido	Não	FLOAT	Peso líquido do produto.	0.35
codigo_setor	Não	INT	Código do setor em que o produto está cadastrado.	5
codigo_linha	Não	INT	Código da linha em que o produto está cadastrado.	3
codigo_marca	Não	INT	Código da marca em que o produto está cadastrado.	2
codigo_colecao	Não	INT	Código da coleção em que o produto está cadastrado.	10
codigo_espessura	Não	INT	Código da espessura em que o produto está cadastrado.	1
codigo_grade1	Não	INT	Código do tamanho do produto.	7
codigo_grade2	Não	INT	Código da cor do produto.	9
unidade	Não	VARCHAR(50)	Código da unidade do produto está cadastrado.	'CX'
Ativo	Não	BIT	Indica se o produto deve ou não ser	1

			mostrado na loja virtual	
codigo_classificacao	Não	INT	Código da classificação do produto	3
dt_cadastro	Não	SMALLDATETIME	Data de cadastro do produto no ERP	2014-01-01
observacao	Não	VARCHAR(MAX)	Observações dos produtos	“Produto para demonstração”
cod_fornecedor	Sim	INT	Código do fornecedor do produto	3
dt_update	Não	SMALLDATETIME	Data de atualização do cadastro do produto no ERP	‘2014-01-01’

### 5.15. produtos\_detalhes

Nome	Obrigatório	Tipo	Descrição	Exemplo
Id_prod_det	Sim	BIGINT	Código único de controle	56234
codigoproduto	Sim	BIGINT	Código do produto	12093
empresa	Sim	INT	Empresa relacionada ao produto	2
saldo	Sim	FLOAT	Quantidade em estoque do produto	12.3
controla_lote	Sim	BIT	Exibe se o produto controla ou não lotes	0
nomeproduto_alternativo	Não	VARCHAR(250)	Exibe o nome alternativo configurado no ERP	“Bota Militar”
dt_update	Não	SMALLDATETIME	Exibe a data de atualização do registro na tabela	‘2016-05-25 16:08:00’

### 5.16. produtos\_custos

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_produtos_custos	Sim	INT	Código único de controle	981273
codigoproduto	Sim	BIGINT	Código do produto	12938
empresa	Sim	INT	Empresa relacionada ao produto	1

custoicms1	Sim	FLOAT	Custo de compra do produto	12.48
ipi1	Sim	FLOAT	Percentual do ipi	0
markup	Sim	FLOAT	Percentual de lucro do produto	100
customedio	Sim	FLOAT	Custo médio do produto	12.48
frete1	Sim	FLOAT	Valor de frete	5
precisao	Sim	INT	Número de casas decimais para os valores do produto	2
precominimo	Sim	FLOAT	Preço mínimo para venda	24
dt_update	Não	SMALLDATETIME	Data de atualização do registro	'2012-06-11'
custoliquido	Não	FLOAT	Custo líquido do produto	12.48
precovenda	Não	FLOAT	Preço de venda do produto	24.96
custototal	Não	FLOAT	Custo total do produto	12.48
precocompra	Não	FLOAT	Preço de compra do produto	12.48

### 5.17. produtos\_campanhas

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_campanha	Sim	INT	Código interno da campanha de promoção	12
nome_campanha	Sim	VARCHAR(60)	Nome da campanha de promoção	'QUEIMA DE ESTOQUE'
vigencia_inicio	Sim	SMALLDATETIME	Data de início da campanha de promoção	'2012-06-11'
vigencia_fim	Sim	SMALLDATETIME	Data de fim da campanha de promoção	'2012-06-30'
observacao	Não	VARCHAR(MAX)	Observações da campanha de promoção	'Avisar fulano'
ativo	Sim	BIT	Indica se a campanha está ativa ou desativa	1

### 5.18. produtos\_promocao

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_promocao	Sim	BIGINT	Código interno da promoção	92387
empresa	Sim	INT	Empresa de vigência da promoção	1
codigoproduto	Sim	BIGINT	Código do produto	1283
preco	Sim	MONEY	Preço do produto em promoção	23.89
data_inicio	Sim	SMALLDATETIME	Data de início da promoção	'2012-06-11'
data_termino	Sim	SMALLDATETIME	Data de termino da promoção	'2012-06-30'
data_cadastro	Sim	DATETIME	Data de cadastro da promoção	'2012-06-11'
ativa	Sim	CHAR(1)	Indica se a promoção está ativa ou desativa	'S'
codigo_campanha	Não	INT	Código da campanha vinculada à promoção	12
promocao_opcional	Sim	BIT	Indica se o preço normal do produto pode ser praticado durante o período de vigência de promoção.	1

### 5.19. produtos\_codebar

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigoproduto	Sim	BIGINT	Código do produto	9283
codebar	Sim	VARCHAR(20)	Código de barras do produto	'17283947263902'
id_produtos_codebar	Não	INT	Código do registro da tabela interno	9283
principal	Sim	BIT	Indica se o código de barras é o principal para o produto caso exista mais de um cadastrado para o mesmo produto	1

empresa	Sim	INT	Código da empresa do código de barras.	9999
---------	-----	-----	--	------

### 5.20. produtos\_informacoes

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_produtos_informacoes	Sim	INT	Código do registro da tabela interno	235
codigoproduto	Sim	BIGINT	Código do produto	23094
informacoes_produto	Sim	VARCHAR(MAX)	Informações adicionais dos produtos adicionadas em formato de texto	'<P>lápis</P>'

### 5.21. produtos\_dimensoes

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigoproduto	Sim	BIGINT	Código do produto	23094
altura	Sim	Decimal(18,2)	Informação da altura do produto	293.03
comprimento	Sim	Decimal(18,2)	Informação do comprimento do produto	123.01

### 5.22. imagens

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_imagem	Sim	INT	Código do registro da tabela interno	12
imagem	Não	IMAGE	Imagem convertida em binário	'0xFD8FFE0001...'
md5	Não	CHAR(32)	Hash de identificação da imagem	E1A6B1B1750E25179FC1E714C2155594

### 5.23. produtos\_imagens

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_imagem_produto	Sim	INT	Código interno da tabela	983
id_imagem	Sim	INT	Código da imagem	439

codigoproduto	Sim	BIGINT	Código do produto	90298
---------------	-----	--------	-------------------	-------

#### 5.24. produtos\_associados

Nome	Obrigatório	Tipo	Descrição	Exemplo
id	Sim	INT	Código interno da tabela	123
codigoproduto	Sim	BIGINT	Código do produto que receberá a associação	90298
codigoproduto_associado	Sim	BIGINT	Código do produto que foi associado	90299
coeficiente_desconto	Sim	FLOAT	Percentual de desconto dado aos produtos associados	10

#### 5.25. pedido

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_pedido	Sim	INT	Código do pedido	583
dt_pedido	Sim	SMALLDATETIME	Data do pedido	'2012-06-11'
id_status_pedido	Sim	INT	Status do pedido	1
cod_cliente	Sim	INT	Código do cliente	70
vl_frete	Não	FLOAT	Valor do frete	8.5
vl_total	Sim	FLOAT	Valor total do pedido	44.5
forma_pgto	Sim	INT	Forma de pagamento	3
plano_pagamento	Sim	INT	Plano de pagamento	12
anotacao	Não	VARCHAR(400)	Observação do pedido	'primeira venda'
taxa_impressao	Não	FLOAT	Taxa de impressão do pedido	0.85
finalizado	Sim	BIT	Indica se o pedido foi finalizado	1
valor_frete_gratis	Não	FLOAT	Valor do frete grátis	0
tipo_frete	Não	INT	Identificador do tipo do frete	99
id_status	Não	INT	Id do status do pedido	0
cod_transportador	Não	INT	Código do transportador do pedido	704

tipo_cobranca_frete	Não	INT	Indica o tipo de cobrança do frete do pedido	1
ativo	Sim	BIT	Indica se o pedido está ativo ou cancelado	1
empresa	Não	INT	Indica em qual empresa o pedido será integrado. Para o recurso funcionar, é necessário entrar em contato com a equipe de banco de dados para ativar o parâmetro	1
dt_disponivel_faturamento	Não	SMALLDATETIME	Indica a data e hora que o pedido ficou disponível para faturamento no Microvix.	'2015-06-11 10:00:03'
valor_credito	Não	MONEY	Valor de crédito utilizado como meio de pagamento	10.00
cod_vendedor	Não	INT	Código do vendedor	2
finalizado_erp	Não	BIT	Indica se o pedido foi finalizado pelo MicrovixERP	0

### 5.26. pedido\_item

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_pedido_item	Sim	INT	Código interno do item do pedido	87354
id_pedido	Sim	INT	Código do pedido	583
codigoproduto	Sim	BIGINT	Código do produto	23453
quantidade	Sim	INT	Quantidade pedida do item	2
vl_unitario	Sim	FLOAT	Valor unitário do item	23.30
txt_observacao	Não	VARCHAR(MAX)	Observação do item	'presente'

### 5.27. clientes

Nome	Obrigatório	Tipo	Descrição	Exemplo
------	-------------	------	-----------	---------

cod_cliente	Sim	INT	Código do cliente	10934
doc_cliente	Não	VARCHAR(14)	CPF do cliente	77303835431
nm_cliente	Não	VARCHAR(50)	Nome do cliente	'João da Silva'
apelido_cliente	Não	VARCHAR(30)	Apelido do cliente utilizado na loja	'Jão'
nm_mae	Não	VARCHAR(50)	Nome da mãe do cliente	'Maria da Silva'
nm_pai	Não	VARCHAR(50)	Nome do pai do cliente	'José da Silva'
nm_conjuge	Não	VARCHAR(50)	Nome do cônjuge do cliente	'Marta da Silva'
dt_cadastro	Não	SMALLDATETIME	Data em que o cliente foi cadastrado na loja	'2012-06-13'
dt_nasc_cliente	Não	SMALLDATETIME	Data de nascimento do cliente	'1973-04-21'
end_cliente	Não	VARCHAR(250)	Endereço de cadastro do cliente	'Avenida das alamedas'
complemento_ed_cliente	Não	VARCHAR(50)	Complemento do endereço de cadastro do cliente	'fundos'
nr_rua_cliente	Não	VARCHAR(20)	Número da casa do cliente	467
bairro_cliente	Não	VARCHAR(60)	Bairro do cliente	'Centro'
cep_cliente	Não	VARCHAR(9)	CEP do cliente	'89339238'
cidade_cliente	Não	VARCHAR(40)	Cidade do cliente	'Santana do Parnaíba'
uf_cliente	Não	CHAR(2)	Estado do cliente	'SP'
fone_cliente	Não	VARCHAR(20)	Telefone para contato com o cliente	'1134586732'
fone_comercial	Não	VARCHAR(20)	Telefone comercial para contato com o cliente	'1134586732'
cel_cliente	Não	VARCHAR(20)	Celular para contato com o cliente	'1199399483'
email_cliente	Não	VARCHAR(50)	E-mail do cliente	'jao30@mail.com'
rg_cliente	Não	VARCHAR(20)	RG do cliente	83949309
rg_orgao_emissor	Não	VARCHAR(7)	Órgão emissor do RG do cliente	'SSPSC'

estado_civil_cliente	Não	TINYINT	Estado civil do cliente	2
empresa_cliente	Não	VARCHAR(30)	Empresa em que o cliente trabalha	'JAUM REPRESENTAÇÕES'
cargo_cliente	Não	VARCHAR(30)	Cargo do cliente na empresa em que trabalha	'Dono'
sexo_cliente	Não	CHAR(1)	Sexo do cliente	'M'
dt_update	Não	SMALLDATETIME	Data da última atualização do cliente	'2012-06-13'
ativo	Sim	BIT	Indica se o cliente está ativo na loja ou não	1
receber_email	Sim	BIT	Indica se o cliente deseja receber e-mails ou não	1
dt_expedicao_rg	Não	SMALLDATETIME	Data de expedição do RG do cliente	'1983-04-21'
cartao_proprio	Não	BIT	Indica se utiliza ou não um cartão de crédito próprio	1
naturalidade	Não	VARCHAR(40)	Naturalidade do cliente	'Joinville'
tempo_residencia	Não	TINYINT	Tempo em que o cliente reside na empresa	3
dt_admissao	Não	SMALLDATETIME	Data de admissão na empresa	'1989-01-13'
renda	Não	MONEY	Renda mensal do cliente	1923.00
numero_compl_rua_cliente	Não	VARCHAR(10)	Complemento do número da rua do cliente	'Fnd 302'

### 5.28. clientes\_enderecos\_entrega

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_endereco_entrega	Sim	INT	Código interno do endereço de entrega do cliente	832094
cod_cliente	Sim	INT	Código do cliente	982

endereco_cliente	Sim	VARCHAR(250)	Nome da rua do cliente	'R. dos Capinzais'
numero_rua_cliente	Não	VARCHAR(20)	Número da residência do cliente na rua	'23'
complemento_end_cli	Não	VARCHAR(60)	Complemento do endereço do cliente	'fundos'
cep_cliente	Não	CHAR(9)	CEP do cliente	89773745
bairro_cliente	Não	VARCHAR(60)	Bairro do cliente	'centro'
cidade_cliente	Não	VARCHAR(40)	Cidade do cliente	'Jaborá'
uf_cliente	Não	CHAR(2)	Estado do cliente	'SC'
descricao	Não	VARCHAR(250)	Descrição do endereço de entrega do cliente	''
principal	Sim	BIT	Indica se o endereço do cliente é o principal utilizado para as entregas da loja	1
id_cidade	Não	INT	Id da cidade no IBGE	4208609

### 5.29. com\_config\_encomenda

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_tipo_encomenda	Sim	INT	Código interno do tipo da encomenda	99
nm_tipo_encomenda	Sim	VARCHAR(100)	Nome do tipo da encomenda	'Transportadora'

### 5.30. produtos\_campos\_adicionais\_nomes

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_campo	Sim	INT	Código interno do cadastro auxiliar adicional	2
ordem	Sim	TINYINT	Ordem em que o campo aparece no ERP. Indica a importância.	3
legenda	Sim	VARCHAR(30)	Descrição do campo adicional	"GRUPO"
tipo	Sim	CHAR(1)	Determina o tipo de informação que o campo	"c"

			adicional deverá validar no input, se texto[c], inteiro [i], numérico[n] ou lógico [b].	
ativo	Sim	BIT	Indica se o registro está ativo ou inativo	1

### 5.31. produtos\_campos\_adicionais\_detalhes

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_campo_detalle	Sim	INT	Código interno do detalhamento de cadastro auxiliar adicional	15
ordem	Não	INT	Ordem em que o campo aparece no ERP. Indica a importância.	3
descricao	Sim	VARCHAR(30)	Descrição do detalhamento do campo adicional	"MEIA IDADE"
id_campo	Sim	INT	Relacionamento com a tabela produtos_campos_adicionais_nomes	2
ativo	Sim	BIT	Determina se o registro está ativo ou não	1

### 5.32. produtos\_campos\_adicionais\_valores

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_campo_valor	Sim	INT	Código interno do valor do cadastro auxiliar adicional para determinado produto	1585
codigoproduto	Sim	BIGINT	Relacionamento com a tabela produtos. Determina o código do produto (SKU)	583
id_campo_detalle	Sim	INT	Relacionamento com a tabela produtos_campos_adicionais_detalhes. Determina qual é o valor do cadastro adicional que o produto está relacionado	15

### 5.33. data\_ultima\_atualizacao

Obrigatório	Tipo	Descrição	Exemplo
Sim	SMALLDATETIME	Determina a data e hora da última atualização da base intermediária	'2014-04-30 14:03:00'

### 5.34. status

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_status	Sim	TINYINT	Código interno do valor do cadastro de status efetuado no ERP	2
descricao_status	Sim	VARCHAR(30)	Descrição do status cadastrado no ERP	"EM ABERTO"

### 5.35. pedido\_status

Nome	Obrigatório	Tipo	Descrição	Exemplo
id	Sim	INT	Código interno do valor do cadastro de status efetuado no ERP	135
id_status	Sim	TINYINT	Chave estrangeira com a tabela status	2
id_pedido	Sim	INT	Chave estrangeira com a tabela de pedidos	44
data_hora	Sim	DATETIME	Data e hora em que o status do pedido foi alterado no ERP	'2014-04-20 14:30'
anotacao	Sim	VARCHAR(80)	Anotação anexada à troca de status no ERP	"manipulação indevida"

### 5.36. com\_config\_encomenda

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_tipo_encomenda	Sim	INT	Código interno do tipo da encomenda	99
nm_tipo_encomenda	Sim	VARCHAR(100)	Descrição do tipo da encomenda	"Transportadora"

### 5.37. fornecedores

Nome	Obrigatório	Tipo	Descrição	Exemplo
cod_fornecedor	Sim	INT	Código interno do fornecedor	3
nome	Sim	VARCHAR(60)	Nome do fornecedor	"Comércio LTDA"
nome_fantasia	Não	VARCHAR(60)	Nome fantasia do fornecedor	"Comércio LTDA"
tipo_pessoa	Sim	CHAR(1)	Indica se o fornecedor é pessoa física ou jurídica	"F"
tipo_fornecedor	Sim	CHAR(1)	Indica se o fornecedor é cadastrado no MicrovixERP como cliente, fornecedor ou transportador	"F"
Endereço	Não	VARCHAR(250)	Endereço do fornecedor	"Avenida das alamedas"
numero_rua	Não	VARCHAR(20)	Número do endereço do fornecedor	"467"
bairro	Não	VARCHAR(60)	Bairro do fornecedor	"Centro"
cep	Não	CHAR(9)	CEP do fornecedor	"892227-65"
cidade	Sim	VARCHAR(40)	Cidade do fornecedor	"Joinville"
uf	Sim	VARCHAR(200)	Estado do fornecedor	"SC"
documento	Sim	VARCHAR(14)	Número do documento fornecedor (CPF, CNPJ...)	"05746237784"
fone	Não	VARCHAR(20)	Telefone do fornecedor	"34223456"

email	Não	VARCHAR(50)	E-mail do fornecedor	"sac@comerciolt da.com.br"
pais	Não	VARCHAR(80)	País do fornecedor	"Brasil"
obs	Não	VARCHAR(MAX)	Observação atribuída ao cadastro do fornecedor	"Rápido"

### 5.38. produtos\_tabelas

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_tabela	Sim	INT	Código interno da tabela de preço	2
nome_tabela	Sim	VARCHAR(50)	Nome da tabela de preço	"Fim de Mês"
ativa	Sim	CHAR(1)	Indica se a tabela está ativa ou inativa	"S"

### 5.39. produtos\_tabelas\_precos

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_prod_tab_preco	Sim	INT	Código interno do preço da tabela de preço	203
id_tabela	Sim	INT	Código da tabela de preço a qual esse item está relacionado.	2
codigoproduto	Sim	BIGINT	Código do produto	2398
precovenda	Sim	MONEY	Valor do item na tabela	3,40

### 5.40. transportadores

Nome	Obrigatório	Tipo	Descrição	Exemplo
cod_transportador	Sim	INT	Código do transportador (o mesmo do ERP)	714
nome	Sim	VARCHAR(60)	Nome do transportador	"Transp. XV novembro"
nome_fantasia	Não	VARCHAR(60)	Nome fantasia do transportador	"Transp. XV novembro"
tipo_pessoa	Sim	CHAR(1)	Indica se é pessoa física (F) ou jurídica (J)	"F"
tipo_transportador	Sim	CHAR(1)	Indica o tipo do transportador (T)	"T"
endereço	Sim	VARCHAR(250)	Endereço do transportador	"Avenida das alamedas"
numero_rua	Não	VARCHAR(20)	Número da rua do transportador	"467"
bairro	Não	VARCHAR(60)	Bairro do transportador	"Centro"
cep	Não	CHAR(9)	CEP do transportador	"05859642284"

cidade	Sim	VARCHAR(40)	Cidade do transportador	"Joinville"
uf	Sim	VARCHAR(20)	Estado do transportador	"SC"
documento	Sim	VARCHAR(14)	CPF ou CNPJ do transportador	"05745841154"
fone	Não	VARCHAR(20)	Telefone do transportador	"4734554673"
email	Não	VARCHAR(50)	E-mail do transportador	"mail@mail.com"
pais	Não	VARCHAR(80)	País do transportador	"Brasil"
obs	Não	VARCHAR(MAX)	Observações gerais sobre o transportador	"Após às 16"

#### 5.41. tipos\_cobranca\_frete

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigo_tipo_cobranca_frete	Sim	INT	Código do tipo da cobrança	1
nome_tipo_cobranca_frete	Sim	Varchar(60)	Descrição do tipo da cobrança	"CIF"

#### 5.42. clientes\_saldo

Nome	Obrigatório	Tipo	Descrição	Exemplo
saldo	Sim	MONEY	Saldo disponível para o cliente no momento da compra	193.40
codigo_cliente	Sim	INT	Código do cliente	293
empresa	Sim	INT	Código da empresa	1

#### 5.43. produtos\_depositos

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_deposito	Sim	INT	Código do depósito	1
nome_deposito	Sim	VARCHAR(50)	Nome do depósito	Devolvidos
disponível	Sim	CHAR(1)	Indica se o depósito está disponível para consulta. Se ele não estiver disponível, produtos relacionados não constam em estoque	'S'
disponivel_transferencia	Sim	BIT	Indica se o depósito pode ou não receber transferência de produtos	1
disponível_franquias	Sim	BIT	Indica se o depósito é disponível para as franquias	1

**5.44. produtos\_detalhes\_depositos**

Nome	Obrigatório	Tipo	Descrição	Exemplo
codigoproduto	Sim	BIGINT	Código do produto	9234
empresa	Sim	INT	Código da empresa do saldo do produto	1
id_deposito	Sim	INT	Código do depósito	1
saldo	Sim	FLOAT	Saldo do produto no depósito para a empresa	8

**5.45. clientes\_creditos**

Nome	Obrigatório	Tipo	Descrição	Exemplo
cod_cliente_erp	Sim	INT	Código do cliente no Microvix ERP	3344
cod_cliente_b2c	Sim	INT	Código do cliente na base B2C	13
empresa	Sim	INT	Código da empresa em que o crédito está disponível	1
valor	Sim	MONEY	Valor de crédito disponível no ERP para ser faturado	50.00
valor_ao_finalizado	Sim	MONEY	Valor de crédito utilizado nos pedidos válidos ainda não faturados	10.00

**5.46. vendedores**

Nome	Obrigatório	Tipo	Descrição	Exemplo
cod_vendedor	Sim	INT	Código do vendedor no Microvix ERP	12
nome_vendedor	Sim	VARCHAR(50)	Nome do vendedor no Microvix ERP	'Zé vendedor'
comissao_produtos	Não	MONEY	Percentual de valor de comissão recebida por produtos	1
comissao_servicos	Não	MONEY	Percentual de valor de comissão recebida por serviços	1
tipo	Não	CHAR(1)	Indica se é comprador ou vendedor	'C'
ativo	Não	BIT	Indica se o vendedor está ativo	0
comissionado	Não	Bit	Indica se o vendedor é comissionado	1

### 5.47. nfe

Nome	Obrigatório	Tipo	Descrição	Exemplo
id_nfe	Sim	INT	Código interno da nfe no Microvix ERP	9
id_pedido	Sim	INT	Código do pedido do B2C	7
documento	Sim	INT	Número da nota fiscal	8
data_emissao	SIM	SMALLDATETIME	Data em que o documento foi emitido	'2016-12-31'
chave	Não	CHAR(44)	Chave de 44 dígitos da NFE	'35130614439371001231550020000000051074042931'
situação	SIM	TINYINT	Indica a situação da NFE	3
xml	NÃO	VARCHAR(MAX)	Exibe os dados do arquivo XML da NFE	</>
excluído	SIM	BIT	Indica se a NFE está excluída ou não	1
identificador_microvix	NÃO	UNIQUEIDENTIFIER	Identificador único da NFE. Código interno do MicrovixERP	7639AE3A-AF95-4C2B-B6EF-872B159C9240
dt_insert	SIM	SMALLDATETIME	Data em que o registro foi inserido	'2017-01-01'
dt_update	SIM	SMALLDATETIME	Data em que o registro foi atualizado	'2017-01-02'
valor_nota	NÃO	MONEY	Valor total final da nota	1114.30
serie	NÃO	VARCHAR(10)	Série da nota	'2'
frete	NÃO	MONEY	Valor total do frete	8.90

### 5.48. nfe\_situacao

Nome	Obrigatório	Tipo	Descrição	Exemplo
if_nfe_situacao	Sim	TINYINT	Código único da situação da NFE	1
descricao	Sim	VARCHAR(30)	Descrição da situação da NFE	Autorizada

## 6. Descrição das tabelas

Cada uma das tabelas tem por objetivo armazenar dados pertinentes à loja virtual do cliente. Algumas destas tabelas são meramente utilizadas para consultas e relacionamentos, já outras são tabelas onde são guardados os cadastros propriamente ditos.

A seguir, tem-se uma lista das tabelas com sua descrição de funcionamento e características. As tabelas em VERDE permitem a inserção e/ou atualização dos dados por parte do e-commerce contratado pelo cliente, já as tabelas em VERMELHO, poderão somente ser consultadas.

- **empresas**: Dados da empresa relacionada no MicrovixERP. Esta tabela é utilizada somente para o relacionamento com outras tabelas ou bem como consulta para a integração.
- **formas\_pgto**: Formas de pagamento fixos do ERP. Os registros desta tabela não serão alterados e servem apenas para consulta e relacionamento. A tabela que faz menção à tabela *formas\_pgto* é: *pedido*.
- **planos**: Cabeçalho dos planos de pagamento cadastrados no ERP. Utilizado para relacionar com os pedidos (tabela *pedido*).
- **planos\_parcelas**: Detalhamento dos planos aberto por parcelas.
- **clientes**: Informações dos clientes que se cadastraram na loja virtual. Para que a integração possa ocorrer, é necessário que o cliente exista caso o mesmo tenha feito um pedido. As tabelas que fazem menção à tabela *clientes* são: *clientes\_enderecos\_entrega*, *pedido*.
- **clientes\_estado\_civil**: Tabela fixa e meramente informativa para o cadastro dos clientes, utilizada para relacionar com o ERP o estado civil do cliente em questão.
- **clientes\_enderecos\_entrega**: Tabela complementar ao cadastro dos clientes. Cada cliente pode possuir mais de um endereço de entrega para o pedido realizado.
- **pedido**: Cabeçalho do pedido realizado na loja virtual. É detalhada pela tabela *pedido\_item*.
- **pedido\_item**: Itens do pedido (tabela *pedido*) realizado na loja virtual, relacionado diretamente com a tabela *produtos*.
- **produtos\_setores**<sup>1</sup>: Cadastro auxiliar de produtos, conforme preenchido no MicrovixERP.
- **produtos\_linhas**<sup>1</sup>: Cadastro auxiliar de produtos, conforme preenchido no MicrovixERP.
- **produtos\_marcas**<sup>1</sup>: Cadastro auxiliar de produtos, conforme preenchido no MicrovixERP.
- **produtos\_colecoes**<sup>1</sup>: Cadastro auxiliar de produtos, conforme preenchido no MicrovixERP.
- **produtos\_espessuras**<sup>1</sup>: Cadastro auxiliar de produtos, conforme preenchido no MicrovixERP.

- **produtos\_grade1**<sup>1</sup>: Cadastro auxiliar de produtos (tamanhos), conforme preenchido no MicrovixERP.
- **produtos\_grade2**<sup>1</sup>: Cadastro auxiliar de produtos (cores), conforme preenchido no MicrovixERP.
- **produto\_unidade**<sup>1</sup>: Cadastro auxiliar de produtos, conforme preenchido no MicrovixERP.
- **produtos\_classificacoes**<sup>1</sup>: Cadastro auxiliar de produtos, conforme preenchido no MicrovixERP.
- **produtos**: Cadastro resumido dos produtos marcados no ERP para fazerem parte da loja virtual. Os produtos estão no último nível (SKU ou codebar), sendo assim, o código do produto é sempre único. As tabelas que fazem menção à tabela *produtos* são: *pedido\_item*, *produtos\_associados*, *produtos\_associados*, *produtos\_codebar*, *produtos\_custos*, *produtos\_detalhes*, *produtos\_imagens*, *produtos\_informacoes* e *produtos\_promocao*.
- **produtos\_detalhes**: Detalhamento dos produtos exibido por empresa e seu respectivo saldo (posição em estoque).
- **produtos\_custos**: Informações sobre os valores (preço de venda, custo, mark-up, etc.) dos produtos. É considerada a tabela padrão no MicrovixERP.
- **produtos\_codebar**: Armazena os códigos de barras dos produtos. O MicrovixERP permite que seja cadastrado mais de um código de barras para o mesmo produto sem a necessidade de duplicá-lo.
- **produtos\_associados**: Utilizada para realizar “venda casada”. Caso um produto seja comprado com outro, é aplicado um percentual de desconto no pedido.
- **produtos\_campanhas**: Campanhas relacionadas diretamente com as promoções.
- **produtos\_promocao**: Promoções vinculadas ao produto. A informação de preço desta tabela prevalece sobre a tabela *produtos\_custos*, caso a data atual esteja entre o período de vigência.
- **produtos\_dimensoes**: Informações referentes às dimensões do produto (comprimento e altura).
- **imagens**: Cadastro das imagens dos *produtos*. A imagem é salva diretamente no banco de dados de forma binária (campo do tipo *image*).
- **produtos\_imagens**: Relacionamento de produtos com imagens. Vários produtos podem possuir a mesma imagem, bem como uma imagem pode possuir apenas um produto (imagem por SKU).
- **produtos\_campos\_adicionais\_nomes**: Determina o nome do cadastro auxiliar adicional (campo auxiliar adicional) como se fosse no nome de uma tabela de cadastro auxiliar de produtos.
- **produtos\_campos\_adicionais\_detalhes**: Detalhamento dos registros da tabela *produtos\_campos\_adicionais\_nomes*. Comporta-se como se fossem as colunas das tabelas dos cadastros auxiliares.

---

<sup>1</sup> Os cadastros auxiliares no MicrovixERP podem mudar de nome, pois utilizam uma label dinâmica. O nome destas tabelas são os originais do ERP.

- **produtos\_campos\_adicionais\_valores**: É o relacionamento entre os produtos e os campos adicionais. Relaciona o valor ao produto.
- **data\_ultima\_atualizacao**: Exibe a data e hora da última atualização da base intermediária.
- **status**: Cadastro de status para os pedidos.
- **pedido\_status**: Histórico de status dos pedidos por data.
- **com\_config\_encomenda**: Cadastro de tipos de encomenda (frete) vindo do ERP.
- **fornecedores**: Cadastro de fornecedores dos produtos.
- **produtos\_tabelas**: Cadastro de tabelas de preços adicionais.
- **produtos\_tabelas\_precos**: Preços dos produtos relacionados às tabelas de preços adicionais.
- **transportadores**: Cadastro de transportadores.
- **tipos\_cobranca\_frete**: Cadastro de tipos de cobrança (CIF ou FOB).
- **clientes\_saldo**: Exibe o saldo em dinheiro que o cliente possui como crédito para gastar na loja. O valor pode ser dividido por empresa ou acumulado entre todas as empresas.
- **produtos\_depositos**: Cadastros de depósitos.
- **produtos\_detalhes\_depositos**: Saldo dos produtos por empresa e por depósito.
- **clientes\_creditos**: Exibe o valor que o cliente tem em créditos disponíveis para compra. O valor a ser considerado pelo E-commerce deve ser sempre *valor – valor\_nao\_finalizado* para evitar que o cliente utilize mais de uma vez o crédito antes de ser baixado.
- **vendedores**: Cadastro de vendedores.
- **nfe**: Cadastro das NFE's geradas através dos pedidos do B2C faturados no MicrovixERP.
- **nfe\_situacao**: Cadastro de situações (status) das NFE's.

## 7. Script de criação da estrutura

Tudo o que foi documentado nos tópicos anteriores, está refletido no script de criação das tabelas a seguir. Este script deve ser utilizado para a criação de um possível ambiente de testes e caso seja necessário. Para a real integração de dados, esta estrutura já estará totalmente montada no servidor da **Linux – Microvix**.

```

USE MASTER
GO

IF NOT EXISTS (SELECT NULL FROM sysdatabases WHERE name = 'b2c_XX')
CREATE DATABASE b2c_XX
GO

ALTER DATABASE b2c_XX SET RECOVERY SIMPLE

USE b2c_XX
GO

-----
-- DADOS GERAIS E CONFIGURACOES B2C
-----

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'b2c_empresas')
CREATE TABLE dbo.b2c_empresas (
    empresa int NOT NULL,
    nome_emp varchar(50) NULL,
    cnpj_emp varchar(14) NULL,
    end_unidade varchar(250) NULL,
    complemento_end_unidade varchar(60) NULL,
    nr_rua_unidade varchar(20) NULL,
    bairro_unidade varchar(60) NULL,
    cep_unidade varchar(9) NULL,
    cidade_unidade varchar(50) NULL,
    uf_unidade char(2) NULL,
    fone_unidade varchar(70) NULL,
    email_unidade varchar(50) NULL,
    CONSTRAINT PK_b2c_empresas PRIMARY KEY (empresa)
)
go

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'formas_pgto')
CREATE TABLE dbo.formas_pgto (
    cod_forma_pgto int NOT NULL,
    forma_pgto varchar(50) NOT NULL,
    CONSTRAINT PK_formas_pgto PRIMARY KEY (cod_forma_pgto)
)
go

IF NOT EXISTS (SELECT NULL FROM formas_pgto WHERE cod_forma_pgto = 1)
INSERT INTO formas_pgto (cod_forma_pgto,forma_pgto) VALUES (1, 'Dinheiro')
GO

IF NOT EXISTS (SELECT NULL FROM formas_pgto WHERE cod_forma_pgto = 2)
INSERT INTO formas_pgto (cod_forma_pgto,forma_pgto) VALUES (2, 'Chq.Vista')
GO

IF NOT EXISTS (SELECT NULL FROM formas_pgto WHERE cod_forma_pgto = 3)
INSERT INTO formas_pgto (cod_forma_pgto,forma_pgto) VALUES (3, 'Chq.Prazo')
GO

IF NOT EXISTS (SELECT NULL FROM formas_pgto WHERE cod_forma_pgto = 4)
INSERT INTO formas_pgto (cod_forma_pgto,forma_pgto) VALUES (4, 'Crediário')
GO

IF NOT EXISTS (SELECT NULL FROM formas_pgto WHERE cod_forma_pgto = 5)
INSERT INTO formas_pgto (cod_forma_pgto,forma_pgto) VALUES (5, 'Cartão')

```

```

GO

IF NOT EXISTS (SELECT NULL FROM formas_pgto WHERE cod_forma_pgto = 6)
INSERT INTO formas_pgto (cod_forma_pgto,forma_pgto) VALUES (6, 'Convênio')
GO

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'planos')
CREATE TABLE dbo.planos (
    plano int NOT NULL,
    nome_plano varchar(30) NOT NULL CONSTRAINT DF_plano_pagamento_nm_plano DEFAULT (''),
    forma_pagamento int NOT NULL CONSTRAINT DF_plano_pagamento_forma_pagamento DEFAULT
('0'),
    qtde_parcelas int NOT NULL CONSTRAINT DF_plano_pagamento_qtde_parcelas DEFAULT
('0'),
    valor_minimo_parcela money NULL,
    indice money NULL,
    CONSTRAINT PK_plano_pagamento PRIMARY KEY (plano)
)
go

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'planos_parcelas')
CREATE TABLE dbo.planos_parcelas (
    plano int NOT NULL,
    ordem_parcela int NOT NULL,
    prazo_parc int NOT NULL,
    id_planos_parcelas int NOT NULL,
    CONSTRAINT FK_planos_parcelas_planos FOREIGN KEY (plano) REFERENCES planos(plano),
    CONSTRAINT PK_planos_parcelas PRIMARY KEY (ordem_parcela,plano)
)
go

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'com_config_encomenda')
CREATE TABLE dbo.com_config_encomenda (
    id_tipo_encomenda int NOT NULL,
    nm_tipo_encomenda varchar(100) NOT NULL,
    CONSTRAINT PK_com_config_encomenda PRIMARY KEY (id_tipo_encomenda)
)
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '99')
INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('99', 'Transportadora')
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '2500')
INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('2500', 'Retirada na Loja')
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '3000')
INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('3000', 'Entregue pela Loja')
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '40010')
INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('40010', 'Sedex')
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '40045')
INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('40045', 'Sedex a Cobrar')
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '40215')
INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('40215', 'Sedex 10')
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '40290')
INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('40290', 'SEDEX Hoje')
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '41017')

```

```

INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('41017', 'Encomenda Normal')
GO

IF NOT EXISTS (SELECT NULL FROM com_config_encomenda WHERE id_tipo_encomenda = '41106')
INSERT INTO com_config_encomenda (id_tipo_encomenda, nm_tipo_encomenda) VALUES
('41106', 'PAC')
GO

IF NOT EXISTS (SELECT NULL FROM SYSOBJECTS WHERE NAME='data_ultima_atualizacao')
CREATE TABLE data_ultima_atualizacao(
    data SMALLDATETIME NOT NULL CONSTRAINT DF_data_ultima_atualizacao_data
default(getdate())
)
GO

IF NOT EXISTS (SELECT NULL FROM data_ultima_atualizacao)
INSERT INTO data_ultima_atualizacao (data) VALUES (GETDATE())
GO

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'tipos_cobranca_frete')
CREATE TABLE dbo.tipos_cobranca_frete (
    codigo_tipo_cobranca_frete int NOT NULL,
    nome_tipo_cobranca_frete varchar(60) NOT NULL,
    CONSTRAINT PK_tipos_cobranca_frete PRIMARY KEY (codigo_tipo_cobranca_frete)
)
GO

IF NOT EXISTS (SELECT NULL FROM tipos_cobranca_frete)
BEGIN
    INSERT INTO tipos_cobranca_frete
(codigo_tipo_cobranca_frete,nome_tipo_cobranca_frete) VALUES (0, '')
    INSERT INTO tipos_cobranca_frete
(codigo_tipo_cobranca_frete,nome_tipo_cobranca_frete) VALUES (1, 'CIF')
    INSERT INTO tipos_cobranca_frete
(codigo_tipo_cobranca_frete,nome_tipo_cobranca_frete) VALUES (2, 'FOB')
END
GO

-----
-----
-----
-----
-- CLIENTES (recebimento no Microvix)
-----
-----

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'clientes_estado_civil')
CREATE TABLE dbo.clientes_estado_civil (
    id_estado_civil tinyint NOT NULL,
    estado_civil VARCHAR(20) NOT NULL
    CONSTRAINT PK_clientes_estado_civil PRIMARY KEY (id_estado_civil)
)
GO
IF NOT EXISTS (SELECT NULL FROM clientes_estado_civil WHERE id_estado_civil = 1)
INSERT INTO clientes_estado_civil (id_estado_civil,estado_civil) VALUES (1, 'Não
Informado')
GO

IF NOT EXISTS (SELECT NULL FROM clientes_estado_civil WHERE id_estado_civil = 2)
INSERT INTO clientes_estado_civil (id_estado_civil,estado_civil) VALUES (2, 'Casado')
GO

IF NOT EXISTS (SELECT NULL FROM clientes_estado_civil WHERE id_estado_civil = 3)
INSERT INTO clientes_estado_civil (id_estado_civil,estado_civil) VALUES (3, 'Solteiro')
GO

IF NOT EXISTS (SELECT NULL FROM clientes_estado_civil WHERE id_estado_civil = 4)
INSERT INTO clientes_estado_civil (id_estado_civil,estado_civil) VALUES (4,
'Divorciado')
GO

IF NOT EXISTS (SELECT NULL FROM clientes_estado_civil WHERE id_estado_civil = 5)
INSERT INTO clientes_estado_civil (id_estado_civil,estado_civil) VALUES (5, 'Viuvo')

```

GO

```
IF NOT EXISTS (SELECT NULL FROM clientes_estado_civil WHERE id_estado_civil = 6)
INSERT INTO clientes_estado_civil (id_estado_civil,estado_civil) VALUES (6, 'Outros')
GO
```

```
IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'clientes')
CREATE TABLE dbo.clientes (
cod_cliente int NOT NULL,
doc_cliente varchar(14) NOT NULL,
nm_cliente varchar(50) NULL,
apelido_cliente varchar(30) NULL,
nm_mae varchar(50) NULL,
nm_pai varchar(50) NULL,
nm_conjuge varchar(50) NULL,
dt_cadastro smalldatetime NULL,
dt_nasc_cliente smalldatetime NULL,
end_cliente varchar(250) NULL,
complemento_end_cliente varchar(50) NULL,
nr_rua_cliente varchar(20) NULL,
bairro_cliente varchar(60) NULL,
cep_cliente varchar(9) NULL,
cidade_cliente varchar(40) NULL,
uf_cliente char(2) NULL,
fone_cliente varchar(20) NULL,
fone_comercial varchar(20) NULL,
cel_cliente varchar(20) NULL,
email_cliente varchar(50) NULL,
rg_cliente varchar(20) NULL,
rg_orgao_emissor varchar(7) NULL,
estado_civil_cliente tinyint NOT NULL CONSTRAINT DF_clientes_estado_civil_cliente
DEFAULT (1),
empresa_cliente varchar(30) NULL,
cargo_cliente varchar(30) NULL,
sexo_cliente char(1) NULL,
dt_update smalldatetime NULL,
ativo bit NOT NULL CONSTRAINT DF_clientes_ativo DEFAULT (1),
receber_email bit NOT NULL CONSTRAINT DF_clientes_receber_email DEFAULT (0),
dt_expedicao_rg smalldatetime NULL,
cartao_proprio bit NULL,
naturalidade varchar(40) NULL,
tempo_residencia tinyint NULL,
dt_admissao smalldatetime NULL,
renda money NULL,
numero_compl_rua_cliente varchar(10) NULL,

CONSTRAINT FK_clientes_clientes_estado_civil FOREIGN KEY
(estado_civil_cliente) REFERENCES clientes_estado_civil(id_estado_civil),
CONSTRAINT CHK_clientes_doc_cliente CHECK ([doc_cliente]<>' ' AND
[doc_cliente]<>' '),
CONSTRAINT PK_clientes PRIMARY KEY (cod_cliente)
)
go
```

```
IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'clientes_enderecos_entrega')
CREATE TABLE dbo.clientes_enderecos_entrega (
id_endereco_entrega int NOT NULL IDENTITY(1,1),
cod_cliente int NOT NULL,
endereco_cliente varchar(250) NOT NULL,
numero_rua_cliente varchar(20) NULL,
complemento_end_cli varchar(60) NULL,
cep_cliente char(9) NULL,
bairro_cliente varchar(60) NULL,
cidade_cliente varchar(40) NULL,
uf_cliente char(2) NULL,
descricao varchar(250) NULL,
principal bit NOT NULL CONSTRAINT DF_clientes_enderecos_entrega_principal DEFAULT
(0),
id_cidade int NULL,
CONSTRAINT FK_clientes_enderecos_entrega_clientes FOREIGN KEY
(cod_cliente) REFERENCES clientes(cod_cliente),
CONSTRAINT PK_clientes_enderecos_entrega PRIMARY KEY
(id_endereco_entrega)
)
go
```

```

IF NOT EXISTS (SELECT NULL FROM SYSOBJECTS WHERE NAME='clientes_saldo')
CREATE TABLE clientes_saldo (
    saldo money not null,
    codigo_cliente int not null,
    empresa int not null,

    CONSTRAINT PK_clientes_saldo PRIMARY KEY (codigo_cliente,empresa)
)
go

-----
-----
-----
-- FORNECEDORES (envio do Microvix para terceiros)
-----

IF NOT EXISTS (SELECT NULL FROM sysobjects WHERE xtype = 'U' AND NAME = 'fornecedores')
CREATE TABLE fornecedores (
    cod_fornecedor INT NOT NULL,
    nome VARCHAR(60) NOT NULL,
    nome_fantasia VARCHAR(60) NULL,
    tipo_pessoa CHAR(1) NOT NULL,
    tipo_fornecedor CHAR(1) NOT NULL,
    endereco VARCHAR(250) NOT NULL,
    numero_ua VARCHAR(20) NULL,
    bairro VARCHAR(60) NULL,
    cep CHAR(9) NULL,
    cidade VARCHAR(40) NOT NULL,
    uf VARCHAR(200) NOT NULL,
    documento VARCHAR(14) NOT NULL,
    fone VARCHAR(20) NULL,
    email VARCHAR(50) NULL,
    pais VARCHAR(80) NULL,
    obs VARCHAR(MAX) NULL,
    CONSTRAINT PK_fornecedores PRIMARY KEY (cod_fornecedor)
)
GO

-----
-----
-----
-- TRANSPORTADORES (envio do Microvix para terceiros)
-----

IF NOT EXISTS (SELECT NULL FROM sysobjects WHERE name = 'transportadores')
CREATE TABLE dbo.transportadores (
    cod_transportador int NOT NULL,
    nome varchar(60) NOT NULL,
    nome_fantasia varchar(60) NULL,
    tipo_pessoa char(1) NOT NULL,
    tipo_transportador char(1) NOT NULL,
    endereco varchar(250) NOT NULL,
    numero_ua varchar(20) NULL,
    bairro varchar(60) NULL,
    cep char(9) NULL,
    cidade varchar(40) NOT NULL,
    uf varchar(20) NOT NULL,
    documento varchar(14) NOT NULL,
    fone varchar(20) NULL,
    email varchar(50) NULL,
    pais varchar(80) NULL,
    obs varchar(MAX) NULL,
    CONSTRAINT PK_transportadores PRIMARY KEY (cod_transportador)
)
GO

-----
-----
-----
-- PRODUTOS (envio do Microvix para terceiros)
-----

```

```

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_setores')
    CREATE TABLE dbo.produtos_setores (
        codigo_setor int NOT NULL,
        nome_setor varchar(100) NOT NULL CONSTRAINT DF_produtos_setores_nome_setor DEFAULT
(''),
        CONSTRAINT PK_produtos_setores PRIMARY KEY (codigo_setor)
    )
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_linhas')
    CREATE TABLE dbo.produtos_linhas (
        codigo_linha int NOT NULL,
        nome_linha varchar(30) NOT NULL CONSTRAINT DF_produtos_linhas_nome_linha DEFAULT
('')
        CONSTRAINT PK_produtos_linhas PRIMARY KEY (codigo_linha)
    )
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_marcas')
    CREATE TABLE dbo.produtos_marcas (
        codigo_marca int NOT NULL,
        nome_marca varchar(100) NOT NULL CONSTRAINT DF_produtos_marcas_nome_marca DEFAULT
('')
        CONSTRAINT PK_produtos_marcas PRIMARY KEY (codigo_marca)
    )
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_colecoes')
    CREATE TABLE dbo.produtos_colecoes (
        codigo_colecao int NOT NULL,
        nome_colecao varchar(100) NOT NULL CONSTRAINT DF_produtos_colecoes_nome_colecao
DEFAULT ('')
        CONSTRAINT PK_produtos_colecoes PRIMARY KEY (codigo_colecao)
    )
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_espessuras')
    CREATE TABLE dbo.produtos_espessuras (
        codigo_espessura int NOT NULL,
        nome_espessura varchar(100) NOT NULL CONSTRAINT DF_produtos_materiais_nome_espessura
DEFAULT ('')
        CONSTRAINT PK_produtos_espessuras PRIMARY KEY (codigo_espessura)
    )
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_gradel')
    CREATE TABLE dbo.produtos_gradel (
        codigo_gradel int NOT NULL,
        nome_gradel varchar(100) NOT NULL CONSTRAINT DF_produtos_gradel_nome_gradel DEFAULT
(''),
        indice int NULL CONSTRAINT DF_produtos_gradel_indice DEFAULT ('0')
        CONSTRAINT PK_produtos_gradel PRIMARY KEY (codigo_gradel)
    )
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_grade2')
    CREATE TABLE dbo.produtos_grade2 (
        codigo_grade2 int NOT NULL,
        nome_grade2 varchar(100) NOT NULL CONSTRAINT DF_produtos_grade2_nome_grade2 DEFAULT
('')
        CONSTRAINT PK_produtos_grade2 PRIMARY KEY (codigo_grade2)
    )
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'unidade')
    CREATE TABLE dbo.unidade (
        unidade varchar(50) NOT NULL,
        CONSTRAINT PK_unidade PRIMARY KEY (unidade)
    )
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_classificacoes')
    CREATE TABLE dbo.produtos_classificacoes (
        codigo_classificacao INT NOT NULL,
        nome_classificacao VARCHAR(50) NOT NULL
        CONSTRAINT PK_produtos_classificacoes PRIMARY KEY (codigo_classificacao)
    )

```

go

```

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'produtos')
CREATE TABLE dbo.produtos (
  codigoproduto bigint NOT NULL,
  referencia varchar(20) NULL,
  codauxiliar1 varchar(40) NULL,
  descricao_basica varchar(100) NULL,
  nome_produto varchar(250) NULL,
  peso_liquido float NULL,
  codigo_setor int NULL,
  codigo_linha int NULL,
  codigo_marca int NULL,
  codigo_colecao int NULL,
  codigo_espessura int NULL,
  codigo_grade1 int NULL,
  codigo_grade2 int NULL,
  codigo_classificacao int NULL,
  unidade varchar(50) NULL,
  ativo BIT NOT NULL CONSTRAINT DF_produtos_ativo DEFAULT(1),
  dt_cadastro smalldatetime NULL,
  observacao VARCHAR(MAX) NULL,
  cod_fornecedor INT NULL,
  dt_update SMALLDATETIME NULL,
  CONSTRAINT FK_produtos_setores_produtos FOREIGN KEY (codigo_setor)
REFERENCES produtos_setores(codigo_setor),
  CONSTRAINT FK_produtos_linhas_produtos FOREIGN KEY (codigo_linha)
REFERENCES produtos_linhas(codigo_linha),
  CONSTRAINT FK_produtos_marcas_produtos FOREIGN KEY (codigo_marca)
REFERENCES produtos_marcas(codigo_marca),
  CONSTRAINT FK_produtos_colecoes_produtos FOREIGN KEY (codigo_colecao)
REFERENCES produtos_colecoes(codigo_colecao),
  CONSTRAINT FK_produtos_espessuras_produtos FOREIGN KEY (codigo_espessura)
REFERENCES produtos_espessuras(codigo_espessura),
  CONSTRAINT FK_produtos_gradel_produtos FOREIGN KEY (codigo_grade1)
REFERENCES produtos_gradel(codigo_grade1),
  CONSTRAINT FK_produtos_grade2_produtos FOREIGN KEY (codigo_grade2)
REFERENCES produtos_grade2(codigo_grade2),
  CONSTRAINT FK_unidade_produtos FOREIGN KEY (unidade) REFERENCES
unidade(unidade),
  CONSTRAINT FK_produtos_classificacoes_produtos FOREIGN KEY
(codigo_classificacao) REFERENCES produtos_classificacoes(codigo_classificacao),
  CONSTRAINT FK_produtos_fornecedores FOREIGN KEY (cod_fornecedor) REFERENCES
fornecedores(cod_fornecedor),
  CONSTRAINT PK_produtos PRIMARY KEY (codigoproduto)
)
go

```

```

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'produtos_detalhes')
CREATE TABLE dbo.produtos_detalhes (
  id_prod_det bigint NOT NULL,
  codigoproduto bigint NOT NULL,
  empresa int NOT NULL,
  saldo float NOT NULL CONSTRAINT DF_produtos_detalhes_saldo DEFAULT (0),
  controla_lote bit NOT NULL CONSTRAINT DF_produtos_detalhes_controla_lote DEFAULT
(0),
  nomeproduto_alternativo VARCHAR(250) NULL,
  CONSTRAINT FK_produtos_detalhes_produtos FOREIGN KEY (codigoproduto) REFERENCES
produtos(codigoproduto),
  CONSTRAINT FK_produtos_detalhes_b2c_empresa FOREIGN KEY (empresa) REFERENCES
b2c_empresas(empresa),
  dt_update SMALLDATETIME NULL,
  CONSTRAINT PK_produtos_detalhes PRIMARY KEY (codigoproduto, empresa)
)
go

```

```

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'produtos_custos')
CREATE TABLE dbo.produtos_custos (
  id_produtos_custos int NOT NULL,
  codigoproduto bigint NOT NULL,
  empresa int NOT NULL,
  custoicms1 float NOT NULL CONSTRAINT DF_produtos_custos_custoicms1 DEFAULT (0.01),
  ipil float NOT NULL CONSTRAINT DF_produtos_custos_ipil DEFAULT (0),
  markup float NOT NULL CONSTRAINT DF_produtos_custos_markup DEFAULT (0),

```

```

    descontomaximo float NOT NULL CONSTRAINT DF_produtos_custos_descontomaximo DEFAULT
(0),
    customedio float NOT NULL CONSTRAINT DF_produtos_custos_customedio DEFAULT (0),
    fretel float NOT NULL CONSTRAINT DF_produtos_custos_fretel DEFAULT (0),
    precisao int NOT NULL CONSTRAINT DF_produtos_custos_precisao DEFAULT (2),
    precominimo float NOT NULL CONSTRAINT DF_produtos_custos_precominimo DEFAULT (0),
    dt_update smalldatetime NOT NULL CONSTRAINT DF_produtos_custos_dt_update DEFAULT
(getdate()),
    custoliquido FLOAT NULL CONSTRAINT DF_produtos_custos_custoliquido DEFAULT (0),
    precovenda FLOAT NULL CONSTRAINT DF_produtos_custos_precovenda DEFAULT (0),
    custototal FLOAT NULL CONSTRAINT DF_produtos_custos_custototal DEFAULT (0),
    precocompra FLOAT NULL CONSTRAINT DF_produtos_custos_precocompra DEFAULT (0),
    CONSTRAINT FK_produtos_custos_produtos FOREIGN KEY (codigoproduto)
REFERENCES produtos(codigoproduto),
    CONSTRAINT PK_produtos_custos PRIMARY KEY (codigoproduto,empresa)
)
GO

```

```

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_campanhas')
CREATE TABLE dbo.produtos_campanhas (
    codigo_campanha int NOT NULL,
    nome_campanha varchar(60) NOT NULL,
    vigencia_inicio smalldatetime NOT NULL,
    vigencia_fim smalldatetime NOT NULL,
    observacao VARCHAR(MAX) NULL,
    ativo bit NOT NULL CONSTRAINT DF_produtos_campanhas_ativo DEFAULT (1),
    CONSTRAINT PK_produtos_campanhas PRIMARY KEY (codigo_campanha)
)
go

```

```

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_promocao')
CREATE TABLE dbo.produtos_promocao (
    codigo_promocao bigint NOT NULL,
    empresa int NOT NULL,
    codigoproduto bigint NOT NULL,
    preco money NOT NULL,
    data_inicio smalldatetime NOT NULL,
    data_termino smalldatetime NOT NULL,
    data_cadastro datetime NOT NULL CONSTRAINT
DF_produtos_promocao_data_cadatro_promocao DEFAULT (getdate()),
    ativa char(1) NOT NULL CONSTRAINT DF_produtos_promocao_promocao_ativa DEFAULT ('S'),
    codigo_campanha int NULL,
    promocao_opcional bit NOT NULL CONSTRAINT DF_produtos_promocao_promocao_opcional
DEFAULT (0),
    CONSTRAINT FK_produtos_promocao_b2c_empresas FOREIGN KEY (empresa) REFERENCES
b2c_empresas(empresa),
    CONSTRAINT FK_produtos_promocao_produtos_campanhas FOREIGN KEY (codigo_campanha)
REFERENCES produtos_campanhas(codigo_campanha),
    CONSTRAINT FK_produtos_promocao_produtos FOREIGN KEY (codigoproduto) REFERENCES
produtos(codigoproduto),
    CONSTRAINT PK_produtos_promocao PRIMARY KEY (codigo_promocao)
)
go

```

```

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_codebar')
CREATE TABLE dbo.produtos_codebar (
    codigoproduto bigint NOT NULL,
    codebar varchar(20) NOT NULL,
    id_produtos_codebar int NULL,
    principal bit NOT NULL CONSTRAINT DF_produtos_codebar_principal DEFAULT (0),
    empresa int NOT NULL CONSTRAINT DF_produtos_codebar_empresa DEFAULT (9999),
    CONSTRAINT FK_produtos_codebar_produtos FOREIGN KEY (codigoproduto) REFERENCES
produtos(codigoproduto),
    CONSTRAINT PK_produtos_codebar PRIMARY KEY (codebar,empresa)
)
go

```

```

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_informacoes')
CREATE TABLE dbo.produtos_informacoes (
    id_produtos_informacoes int NOT NULL,
    codigoproduto bigint NOT NULL,
    informacoes_produto VARCHAR(MAX) NOT NULL CONSTRAINT
DF_produtos_informacoes_informacoes_produto DEFAULT (''),

```

```

        CONSTRAINT FK_produtos_informacoes_produtos FOREIGN KEY (codigoproduto)
REFERENCES produtos(codigoproduto),
        CONSTRAINT PK_produtos_informacoes PRIMARY KEY (id_produtos_informacoes)
)
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'imagens')
CREATE TABLE dbo.imagens (
    id_imagem int NOT NULL,
    imagem image NULL,
    md5 char(32) NULL,
    CONSTRAINT PK_imagens PRIMARY KEY (id_imagem)
)
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_imagens')
CREATE TABLE dbo.produtos_imagens (
    id_imagem_produto int NOT NULL,
    id_imagem int NOT NULL,
    codigoproduto bigint NOT NULL,
    CONSTRAINT FK_produtos_imagens_imagens FOREIGN KEY (id_imagem) REFERENCES
imagens(id_imagem),
    CONSTRAINT PK_produtos_imagens PRIMARY KEY (id_imagem_produto)
)
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_associados')
CREATE TABLE dbo.produtos_associados (
    id int NOT NULL,
    codigoproduto bigint NOT NULL,
    codigoproduto_associado bigint NOT NULL CONSTRAINT
DF_produtos_associados_id_produto_associado DEFAULT ('0'),
    coeficiente_desconto float NOT NULL CONSTRAINT
DF_produtos_associados_coeficiente_desconto DEFAULT ('0')
    CONSTRAINT FK_produtos_associados_produtos FOREIGN KEY (codigoproduto) REFERENCES
produtos(codigoproduto),
    CONSTRAINT FK_produtos_associados_produtos2 FOREIGN KEY (codigoproduto_associado)
REFERENCES produtos(codigoproduto),
    CONSTRAINT PK_produtos_associados PRIMARY KEY (id)
)
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_dimensoes')
CREATE TABLE produtos_dimensoes (
    codigoproduto BIGINT NOT NULL,
    altura DECIMAL(18,2) NOT NULL,
    comprimento DECIMAL(18,2) NOT NULL
)
GO

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_campos_adicionais_nomes')
CREATE TABLE dbo.produtos_campos_adicionais_nomes (
    id_campo int NOT NULL,
    ordem tinyint NOT NULL,
    legenda varchar(30) NOT NULL,
    tipo char(1) NULL,
    ativo bit NOT NULL CONSTRAINT DF_produtos_campos_adicionais_nomes_ativo DEFAULT(1),
    CONSTRAINT PK_produtos_campos_adicionais_nomes PRIMARY KEY (id_campo)
)
GO

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name =
'produtos_campos_adicionais_detalhes')
CREATE TABLE dbo.produtos_campos_adicionais_detalhes (
    id_campo_detalhe int NOT NULL,
    descricao varchar(30) NOT NULL,
    ordem int NULL,
    id_campo int NULL,
    chave int NULL,
    ativo bit NOT NULL CONSTRAINT DF_produtos_campos_adicionais_detalhes_ativo DEFAULT
(1),
    CONSTRAINT FK_produtos_campos_adicionais_detalhes_produtos_campos_adicionais_nomes
FOREIGN KEY (id_campo) REFERENCES produtos_campos_adicionais_nomes(id_campo),
    CONSTRAINT PK_produtos_campos_adicionais_detalhes PRIMARY KEY
(id_campo_detalhe)
)
GO

```

```

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name =
'produtos_campos_adicionais_valores')
    CREATE TABLE dbo.produtos_campos_adicionais_valores (
        id_campo_valor int NOT NULL,
        codigoproduto bigint NOT NULL,
        id_campo_detalhe int NOT NULL,
        CONSTRAINT FK_produtos_campos_adicionais_valores_produtos FOREIGN KEY
(codigoproduto) REFERENCES produtos(codigoproduto),
        CONSTRAINT FK_produtos_campos_adicionais_valores_produtos_campos_adicionais_detalhes
FOREIGN KEY (id_campo_detalhe) REFERENCES
produtos_campos_adicionais_detalhes(id_campo_detalhe),
        CONSTRAINT PK_produtos_campos_adicionais_valores PRIMARY KEY
(codigoproduto,id_campo_detalhe)
    )
GO

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'produtos_tabelas')
    CREATE TABLE dbo.produtos_tabelas (
        id_tabela INT NOT NULL CONSTRAINT PK_produtos_tabelas PRIMARY KEY,
        nome_tabela VARCHAR(50) NOT NULL,
        ativa CHAR(1) NOT NULL
    )
GO

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'produtos_tabelas_precos')
    CREATE TABLE dbo.produtos_tabelas_precos (
        id_prod_tab_preco INT NOT NULL CONSTRAINT PK_produtos_tabelas_precos PRIMARY
KEY,
        id_tabela INT NOT NULL CONSTRAINT
FK_produtos_tabelas_precos_produtos_produtos_tabelas FOREIGN KEY REFERENCES
produtos_tabelas(id_tabela),
        codigoproduto BIGINT NOT NULL CONSTRAINT FK_produtos_tabelas_precos_produtos
FOREIGN KEY REFERENCES produtos(codigoproduto),
        precovenda MONEY NOT NULL
    )
GO

-----
-----
-----
-- PEDIDOS (recebimento no Microvix)
-----
-----

IF NOT EXISTS (SELECT 1 FROM sysobjects WHERE name = 'pedido')
    CREATE TABLE dbo.pedido (
        id_pedido int NOT NULL,
        dt_pedido smalldatetime NOT NULL,
        id_status_pedido int NOT NULL CONSTRAINT DF_pedido_id_status_pedido DEFAULT ('0'),
        cod_cliente int NOT NULL,
        vl_frete float NULL,
        vl_total float NOT NULL CONSTRAINT DF_pedido_vl_total DEFAULT ('0'),
        forma_pgto int NOT NULL,
        plano_pagamento int NOT NULL CONSTRAINT DF_pedido_plano_pagamento DEFAULT (''),
        anotacao varchar(400) NULL,
        taxa_impressao float NULL,
        finalizado bit NOT NULL CONSTRAINT DF_pedido_finalizado DEFAULT (1),
        tipo_frete INT NOT NULL CONSTRAINT DF_pedido_tipo_frete DEFAULT('99'),
        id_status INT NOT NULL CONSTRAINT DF_pedido_id_status DEFAULT(0),
        CONSTRAINT FK_pedido_clientes FOREIGN KEY (cod_cliente) REFERENCES
clientes(cod_cliente),
        CONSTRAINT FK_pedido_planos FOREIGN KEY (plano_pagamento) REFERENCES
planos(plano),
        CONSTRAINT FK_pedido_formas_pgto FOREIGN KEY (forma_pgto) REFERENCES
formas_pgto(cod_forma_pgto),
        CONSTRAINT FK_pedido_com_config_encomenda FOREIGN KEY (tipo_frete) REFERENCES
com_config_encomenda(id_tipo_encomenda),
        CONSTRAINT PK_pedido PRIMARY KEY (id_pedido)
    )
GO

IF NOT EXISTS (SELECT NULL from syscolumns where name ='tipo_frete' and
id=object_id('pedido'))

```

```

ALTER TABLE pedido ADD tipo_frete INT NOT NULL CONSTRAINT DF_pedido_tipo_frete
DEFAULT(99)
GO

IF NOT EXISTS (SELECT NULL from syscolumns where name ='cod_transportador' and
id=object_id('pedido'))
ALTER TABLE pedido ADD cod_transportador INT NULL CONSTRAINT FK_pedido_transportadores
FOREIGN KEY REFERENCES transportadores(cod_transportador)
GO

IF NOT EXISTS (SELECT NULL from syscolumns where name ='valor_frete' and
id=object_id('pedido'))
ALTER TABLE pedido ADD valor_frete MONEY NOT NULL CONSTRAINT DF_pedido_valor_frete
DEFAULT(0)
GO

IF NOT EXISTS (SELECT NULL from syscolumns where name ='tipo_cobranca_frete' and
id=object_id('pedido'))
ALTER TABLE pedido ADD tipo_cobranca_frete INT NULL CONSTRAINT
FK_pedido_tipos_cobranca_frete FOREIGN KEY REFERENCES
tipos_cobranca_frete(codigo_tipo_cobranca_frete)
GO

IF NOT EXISTS (SELECT NULL from syscolumns where name ='ativo' and
id=object_id('pedido'))
ALTER TABLE pedido ADD ativo BIT NOT NULL CONSTRAINT DF_pedido_ativo DEFAULT(1)
GO

IF NOT EXISTS (SELECT NULL from syscolumns where name ='empresa' and
id=object_id('pedido'))
ALTER TABLE pedido ADD empresa INT NULL
GO

IF NOT EXISTS (SELECT NULL from syscolumns where name ='dt_disponivel_faturamento' and
id=object_id('pedido'))
ALTER TABLE pedido ADD dt_disponivel_faturamento SMALLDATETIME NULL
GO

IF NOT EXISTS (SELECT NULL FROM syscolumns WHERE name ='valor_credito' AND
id=OBJECT_ID('pedido'))
ALTER TABLE pedido ADD valor_credito MONEY NULL
GO

IF EXISTS (SELECT NULL FROM sysobjects WHERE NAME='fnCalculaCreditosNaoFinalizados')
BEGIN
EXEC ('ALTER TABLE clientes_creditos DROP COLUMN valor_ao_finalizado')
EXEC ('DROP FUNCTION fnCalculaCreditosNaoFinalizados')
END
GO

IF NOT EXISTS (SELECT NULL FROM sysobjects WHERE NAME='fnCalculaCreditosNaoFinalizados')
EXEC ('CREATE FUNCTION fnCalculaCreditosNaoFinalizados (@cod_cliente_b2c INT)
RETURNS MONEY
AS
BEGIN
RETURN (
SELECT ISNULL(SUM(ISNULL(valor_credito, 0)),0) AS valor
FROM pedido
WHERE ativo = 1
AND finalizado = 0
AND cod_cliente = @cod_cliente_b2c
)
END')
GO

IF NOT EXISTS (SELECT NULL FROM syscolumns WHERE name ='valor_ao_finalizado' AND
id=OBJECT_ID('clientes_creditos'))
ALTER TABLE clientes_creditos ADD valor_ao_finalizado AS
dbo.fnCalculaCreditosNaoFinalizados(cod_cliente_b2c)
GO

IF NOT EXISTS (SELECT NULL FROM sysobjects WHERE NAME='clientes_creditos')
CREATE TABLE clientes_creditos (

```

```

        cod_cliente_erp INT NOT NULL,
        cod_cliente_b2c INT NOT NULL CONSTRAINT FK_clientes_creditos_clientes FOREIGN KEY
REFERENCES clientes,
        empresa INT NOT NULL CONSTRAINT FK_clientes_creditos_b2c_empresas FOREIGN KEY
REFERENCES b2c_empresas,
        valor MONEY NOT NULL,
        valor_ao_finalizado AS dbo.fnCalculaCreditosNaoFinalizados(cod_cliente_b2c)
    )
GO

IF NOT EXISTS (SELECT NULL FROM sysobjects WHERE NAME='PK_clientes_creditos')
ALTER TABLE clientes_creditos ADD CONSTRAINT PK_clientes_creditos PRIMARY KEY
(cod_cliente_erp,empresa)
GO

IF NOT EXISTS (SELECT NULL FROM syscolumns WHERE name ='finalizado_erp' AND
id=OBJECT_ID('pedido'))
ALTER TABLE pedido ADD finalizado_erp BIT
GO

UPDATE pedido SET finalizado_erp = 0
GO

IF NOT EXISTS (SELECT NULL FROM sysobjects WHERE NAME='DF_pedido_finalizado_erp')
ALTER TABLE pedido ADD CONSTRAINT DF_pedido_finalizado_erp DEFAULT 0 FOR finalizado_erp
GO

IF NOT EXISTS (SELECT NULL FROM syscolumns WHERE name ='valor_credito' AND
id=OBJECT_ID('pedido'))
ALTER TABLE pedido ADD valor_credito MONEY NULL
GO

IF NOT EXISTS (SELECT NULL FROM sysobjects WHERE NAME='vendedores')
CREATE TABLE vendedores (
    cod_vendedor INT NOT NULL CONSTRAINT PK_vendedores PRIMARY KEY,
    nome_vendedor VARCHAR(50) NOT NULL,
    comissao_produtos MONEY NULL,
    comissao_servicos MONEY NULL,
    tipo CHAR(1) NULL,
    ativo BIT NULL,
    comissionado BIT NULL
)
GO

IF NOT EXISTS (SELECT NULL FROM syscolumns WHERE name ='cod_vendedor' AND
id=OBJECT_ID('pedido'))
ALTER TABLE pedido ADD cod_vendedor INT NULL CONSTRAINT FK_pedido_tabela_vendedores
FOREIGN KEY REFERENCES vendedores
GO

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'pedido_item')
CREATE TABLE dbo.pedido_item (
    id_pedido_item int NOT NULL,
    id_pedido_int NOT NULL,
    codigoproduto bigint NOT NULL CONSTRAINT DF_pedido_item_id_produto DEFAULT (0),
    quantidade int NOT NULL CONSTRAINT DF_pedido_item_quantidade DEFAULT ('1'),
    vl_unitario float NOT NULL CONSTRAINT DF_pedido_item_vl_unitario DEFAULT (0.01),
    txt_observacao VARCHAR(MAX) NULL,
    CONSTRAINT FK_pedido_item_pedido FOREIGN KEY (id_pedido) REFERENCES
pedido(id_pedido),
    CONSTRAINT FK_pedido_item_produtos FOREIGN KEY (codigoproduto) REFERENCES
produtos(codigoproduto),
    CONSTRAINT PK_pedido_item PRIMARY KEY (id_pedido_item)
)
go

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'status')
CREATE TABLE dbo.[status] (
    id_status tinyint NOT NULL,
    descricao_status varchar(30) NOT NULL CONSTRAINT DF_status_descricao_status DEFAULT
(''),
    CONSTRAINT PK_status PRIMARY KEY (id_status)
)
go

```

```

IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'pedido_status')
    CREATE TABLE dbo.pedido_status (
        id int NOT NULL,
        id_status tinyint NOT NULL,
        id_pedido int NOT NULL,
        data_hora datetime NOT NULL,
        anotacao varchar(80) NOT NULL,
        CONSTRAINT FK_pedido_status_status FOREIGN KEY (id_status) REFERENCES
        [status](id_status)
    )
go

GO
IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'produtos_depositos')
    CREATE TABLE dbo.produtos_depositos (
        id_deposito INT NOT NULL ,
        nome_deposito varchar(50) NOT NULL,
        disponivel CHAR(1) NOT NULL,
        disponivel_transferencia BIT NOT NULL,
        disponivel_franquias BIT NOT NULL,
        CONSTRAINT PK_produtos_depositos PRIMARY KEY (id_deposito)
    )
go

-----
GO
IF NOT EXISTS (SELECT NULL FROM sysobjects WHERE NAME='produtos_detalhes_depositos')
    CREATE TABLE produtos_detalhes_depositos(
        codigoproduto BIGINT NOT NULL CONSTRAINT FK_produtos FOREIGN KEY REFERENCES produtos
        (codigoproduto),
        empresa INT NOT NULL CONSTRAINT FK_empresa FOREIGN KEY REFERENCES b2c_empresas
        (empresa),
        id_deposito INT NOT NULL CONSTRAINT DF_produtos_depositos FOREIGN KEY REFERENCES
        produtos_depositos (id_deposito),
        saldo FLOAT NOT NULL
    )
GO

-----
IF NOT EXISTS(SELECT 1 FROM sysobjects WHERE name = 'nfe_situacao')
    CREATE TABLE dbo.nfe_situacao (
        id_nfe_situacao TINYINT NOT NULL,
        descricao VARCHAR(30) NOT NULL,
        CONSTRAINT PK_nfe_situacao PRIMARY KEY (id_nfe_situacao)
    )
GO

-----
IF NOT EXISTS(SELECT NULL FROM sysobjects WHERE name = 'nfe')
    CREATE TABLE dbo.nfe (
        id_nfe INT NOT NULL,
        id_pedido INT NOT NULL,
        documento INT NOT NULL,
        data_emissao SMALLDATETIME NOT NULL,
        chave CHAR(44) NULL,
        situacao TINYINT NULL,
        [xml] VARCHAR(MAX) NULL,
        excluido BIT NOT NULL CONSTRAINT DF_nfe_excluido DEFAULT 0,
        identificador_microvix UNIQUEIDENTIFIER,
        dt_insert SMALLDATETIME NOT NULL CONSTRAINT DF_nfe_dt_insert DEFAULT GETDATE(),
        dt_update SMALLDATETIME NOT NULL CONSTRAINT DF_nfe_dt_update DEFAULT GETDATE(),
        valor_nota MONEY NULL,
        serie VARCHAR(10) NULL,
        frete MONEY NULL,
        CONSTRAINT PK_nfe PRIMARY KEY (id_nfe),
        CONSTRAINT FK_nfe_pedido FOREIGN KEY (id_pedido) REFERENCES pedido(id_pedido),
        CONSTRAINT FK_nfe_nfe_situacao FOREIGN KEY (situacao) REFERENCES
        nfe_situacao(id_nfe_situacao)
    )
GO

```