



**UNIVERSIDADE DO SAGRADO CORAÇÃO – USC**

**CCESA – CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS**

Fabício Elias Jeremias  
Juliana Rubira de Amorim

**MANUAL DO SISTEMA**

**SGPF**  
**SISTEMA DE GERENCIAMENTO DE PRODUTOS**  
**FARMACÊUTICOS**

**Bauru - 2007**



**UNIVERSIDADE DO SAGRADO CORAÇÃO – USC**

**CCESA – CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS**

**Fabrcio Elias Jeremias  
Juliana Rubira de Amorim**

**SGPF  
SISTEMA DE GERENCIAMENTO DE PRODUTOS  
FARMACÊUTICOS**

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas como parte dos requisitos para obtenção do título de bacharel em Sistemas de Informação, sob orientação do Prof. Ms. Élvio Gilberto da Silva.

J55s

Jeremias, Fabricio Elias

SGPF Sistema de Gerenciamento de Produtos  
Farmacêuticos / Fabrício Elias Jeremias, Juliana Rubira de  
Amorim – 2007.  
82f.

Orientador Prof. Ms. Elvio Gilberto da Silva  
Trabalho de Conclusão de Curso (Bacharelado em  
Sistema de Informação) - Universidade do Sagrado Coração  
– Bauru - São Paulo.

1. Farmácia 2. Produtos 3. Controle de estoque I.  
Amorim, Juliana Rubira de II. Silva, Elvio Gilberto da  
III. Título.

## **AGRADECIMENTOS**

Aos meus pais, Emanuel Jeremias e Marli Aparecida Elias Jeremias sempre presentes tanto nos momentos alegres e difíceis da minha vida, transmitindo-me valores e princípios. Vocês acreditaram que eu venceria todas as barreiras que surgissem em meu caminho, desde que houvesse dedicação.

Essa lição, com certeza, nunca será esquecida.

Ao meu Prof. Ms. Elvino Gilberto da Silva, pela sua orientação objetiva, incentivo, dedicação e apoio, em todos os momentos, para que este trabalho fosse desenvolvido corretamente.

O meu agradecimento, também, à Profa. Luciene Rodrigues Rinaldi.

Fabício

Aos meus pais, Alonzo Passos de Amorim e Mara Lucia Norinho Rubira de Amorim, que, com dedicação e carinho, ensinaram-me a mestra arte de lapidar as dificuldades da vida com as ferramentas da verdade, aprendidas no lar. Grata pela oportunidade de vida!

Ao meu Prof. Ms. Elvino Gilberto da Silva, pela sua orientação objetiva, incentivo, dedicação e apoio, em todos os momentos, para que este trabalho fosse desenvolvido corretamente.

O meu agradecimento, também, à Profa. Luciene Rodrigues Rinaldi.

Juliana

## SUMÁRIO

1.	ESTUDO DE CASO .....	1
1.1.	INTRODUÇÃO .....	1
1.2.	AMBIENTE .....	1
1.3.	A EMPRESA .....	1
1.4.	AMBIENTE TECNOLÓGICO .....	2
2.	PROJETO .....	3
2.1	IDENTIFICAÇÃO DO SOFTWARE .....	3
2.2	MÓDULOS DO SOFTWARE .....	3
2.2.1	<i>Módulo de Cadastro</i> .....	3
2.2.2	<i>Módulo de Consulta</i> .....	4
2.2.3	<i>Módulo de Relatório</i> .....	4
2.2.4	<i>Módulo Administrativo</i> .....	4
2.2.5	<i>Módulo Financeiro</i> .....	5
2.3	JUSTIFICATIVA DO SISTEMA .....	5
2.4	ESPECIFICAÇÕES TÉCNICAS .....	5
2.5	FERRAMENTAS TECNOLÓGICAS UTILIZADAS .....	7
2.6	FUNÇÕES DE SOFTWARE IMPORTANTES .....	8
2.7	LIMITAÇÕES E RESTRIÇÕES IMPORTANTES EXISTENTES .....	8
1.	DIAGRAMA DE FLUXO DE DADOS (DFD) .....	9
2.	MODELO DE ENTIDADE E RELACIONAMENTO (MER) .....	25
3.	DICIONÁRIO DE DADOS DE DEPÓSITOS .....	27
4.	DICIONÁRIO DE DADOS DOS FLUXOS .....	36
5.	MINI-ESPECIFICAÇÃO .....	47

## **1. ESTUDO DE CASO**

### **1.1. INTRODUÇÃO**

Este manual contém as informações referentes a apresentação da empresa, projeto proposto, justificativa, especificações técnicas.

### **1.2. AMBIENTE**

Farmácia da Clínica de Educação para Saúde (CEPS) da Universidade do Sagrado Coração (USC).

### **1.3. A EMPRESA**

A Clínica de Educação para Saúde (CEPS) situa-se à Rua Irmã Arminda, nº 10-50, dentro da Universidade do Sagrado Coração, na cidade de Bauru, estado de São Paulo.

A CEPS tem como objetivo a realização de diversos exames e o fornecimento de remédios a pessoas atendidas pelo Sistema Único de Saúde (SUS). A clínica funciona de segunda à sexta - feira das 7h30min às 17h.

A estrutura física da CEPS é composta por 2 recepções e 22 salas, onde são realizadas avaliações clínicas e exames específicos, sendo uma das salas o almoxarifado, onde são armazenados os medicamentos recebidos pela farmácia. Atualmente conta com 25 médicos em 17 especialidades, 6 recepcionistas, 2 auxiliares de enfermagem, 1 técnica em EEG (Eletroencefalograma).

#### **1.4. AMBIENTE TECNOLÓGICO**

A Farmácia da CEPS não possui nenhum sistema que forneça informações sobre funcionários, fornecedores, produtos, controle de validade, emissão de relatórios, controle de entrada e saída de produtos. Todo e qualquer tipo de contagem é feita manualmente, e os remédios são organizados nas prateleiras por data de vencimento.

## **2. PROJETO**

### **2.1 IDENTIFICAÇÃO DO SOFTWARE**

O *software* proposto solucionará as falhas e dificuldades da farmácia da USC ou de farmácias que possuem movimentação financeira, trazendo soluções e visando, de uma maneira simples e eficiente, o retorno das informações, além de aprimorar o gerenciamento de estoque e do setor financeiro.

### **2.2 MÓDULOS DO SOFTWARE**

#### ***2.2.1 Módulo de Cadastro***

Neste módulo serão cadastradas todas as informações com as quais a empresa trabalha, como por exemplo:

Cadastros:

- Fornecedores: Permite ao usuário cadastrar os dados pessoais dos fornecedores.
- Funcionários: Permite ao usuário cadastrar os dados pessoais dos funcionários de sua empresa
- Produtos: Permite ao usuário cadastrar os dados dos produtos.
- Lote: Permite controlar por lote e prazo de validade.
- Usuários e Permissões: Permite ao administrador do sistema cadastrar usuários e suas permissões no sistema.
- Grupos e Permissões: Permite ao administrador do sistema cadastrar determinado grupo e suas permissões no sistema



### **2.2.2 Módulo de Consulta**

Este módulo permitirá a consulta de informações como consulta de fornecedores (Ativos e Inativos), funcionários cadastrados e logados no sistema, grupos de acesso, produtos disponíveis, contas a pagar e contas a receber, estoque.

### **2.2.3 Módulo de Relatório**

Este módulo conta com os seguintes relatórios:

- Clientes;
- Fornecedores;
- Usuários cadastrados e logados;
- Grupo de usuários;
- Produtos mais vendidos;
- Gráfico da Curva ABC;
- Controle de estoque (entrada de mercadorias no estoque e vendas de mercadorias);
- Notas fiscais de saída;
- Contas a pagar e a receber.

### **2.2.4 Módulo Administrativo**

Este módulo apresenta as seguintes funções:

- *Backup* e restauração do banco;
- Atualização do sistema;
- Relatórios dinâmicos;
- Alteração do nome e senha de usuário, grupo, permissões no sistema;
- Usuários bloqueados;
- Registro de atividades no sistema (LOG);

### **2.2.5 Módulo Financeiro**

Este módulo possuirá as seguintes telas:

- Receitas e despesas;
- Controle do pagamento dos produtos aos fornecedores;
- Pedido de venda, pedido de compra,
- Faturamento e solicitação de nota fiscal;
- Recebimento do cliente.

## **2.3 JUSTIFICATIVA DO SISTEMA**

Com base nas informações obtidas dentro da Farmácia da USC, foi constatado que algumas mudanças precisavam ser feitas. Portanto, o desenvolvimento de um novo sistema tem como objetivo sanar as falhas e dificuldades da farmácia da USC ou de farmácias que possuem movimentação financeira, trazendo soluções e visando, de uma maneira simples e mais eficiente, o retorno das informações, melhorando assim o gerenciamento do estoque. O sistema também possui o módulo financeiro, facilitando o seu gerenciamento, caso uma farmácia possua movimentação financeira.

## **2.4 ESPECIFICAÇÕES TÉCNICAS**

A farmácia deverá possuir um servidor para armazenar o banco de dados e o sistema. Os computadores deverão estar ligados em rede para que interajam com a base de dados. O software deverá ser instalado em cada máquina que for utilizar os recursos disponíveis do mesmo. A resolução mínima exigida deverá ser de 1024 x 768.

Segue abaixo a configuração mínima para o servidor:

- Pentium IV 2.0 GHz ou similar
- 512 MB RAM
- 80 GB de disco rígido
- Microsoft SQL Server 2000
- Crystal Reports 8.5

- Windos 2000 Professional ou superior
- Gravador de CD-ROM

Segue abaixo a configuração mínima para as estações de trabalho:

- Pentium III 700 MHz ou similar
- 256 MB RAM
- 20 GB de disco rígido
- Windows 98, 2000 ou XP
- Monitor de 15 ou 17"
- Leitor de CD-ROM

## **2.5 FERRAMENTAS TECNOLÓGICAS UTILIZADAS**

As tecnologias usadas para o desenvolvimento do software foram:

- Microsoft Visual Basic 6.0
- Microsoft SQL Server 2000
- BPwin 4.0
- ERwin 4.1
- Crystal Reports 8.5

## 2.6 FUNÇÕES DE SOFTWARE IMPORTANTES

O Sistema apresenta as seguintes funções:

- Multiusuário: essa função permite que mais de um usuário tenha acesso ao sistema ao mesmo tempo;
- *Log* do sistema: essa função permite ao administrador visualizar informações sobre os acessos ao sistema.
- *Backup* e restauração do banco: Permite ao administrador ou usuário executar um backup de todas as atividades que foram executadas até o momento.
- Geração de relatórios: Permite ao usuário gerar relatórios desde os mais simples até os mais detalhados.
- Criptografia de senha;
- Tratamentos de erros;
- Permissão de usuário e grupo por nível de acesso.

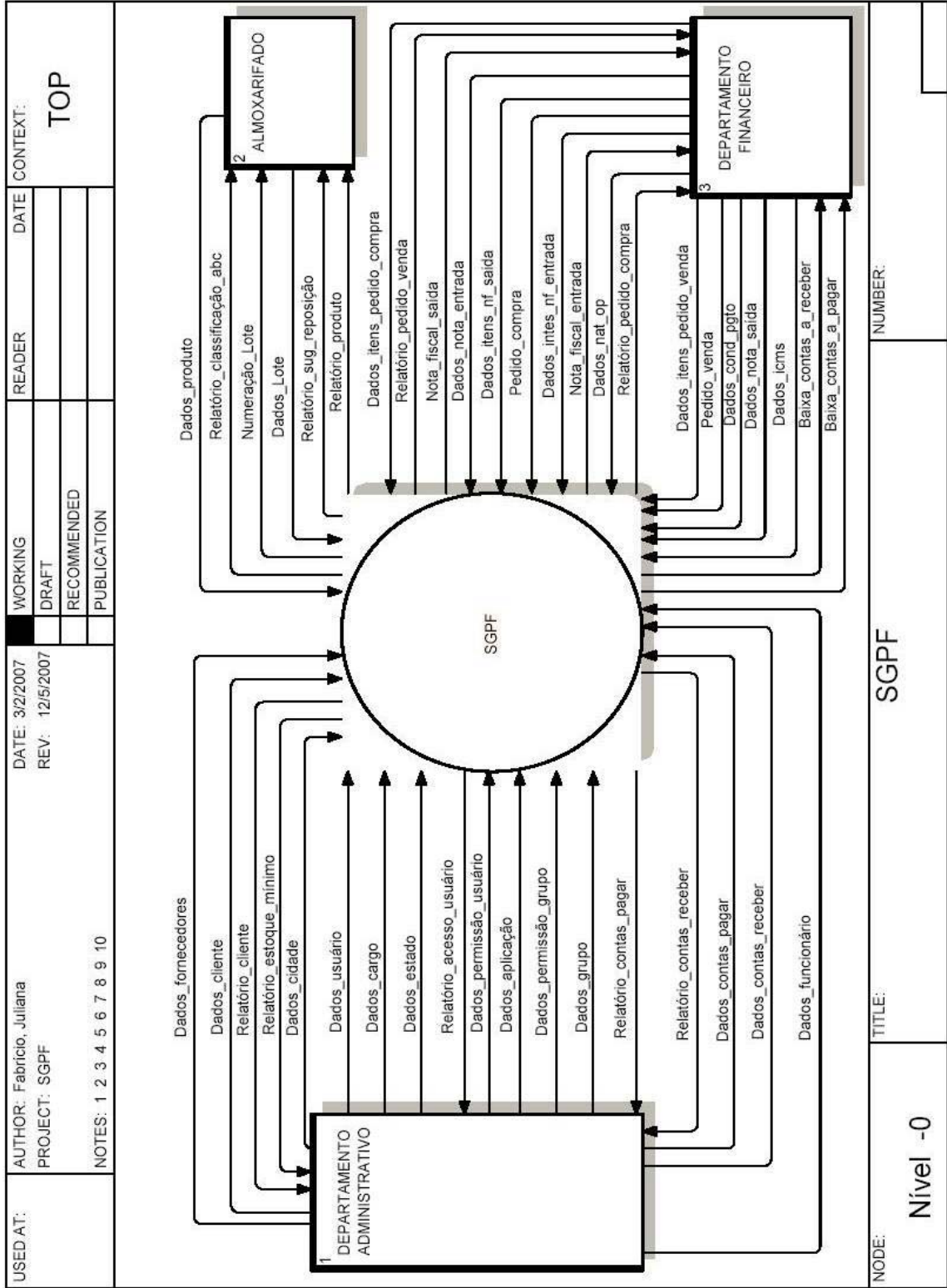
O administrador terá acesso total ao sistema. Quanto ao estoque de produtos, avisa quando o nível estiver baixo; controle de fornecedores, usuários ativos e inativos. Durante a confirmação de um pedido, verifica se há produtos suficientes. Análise dos produtos mais vendidos e menos vendidos e as informações visualizadas através de gráficos e tabelas, checando o gerenciamento nas decisões de quais produtos tiveram mais saída.

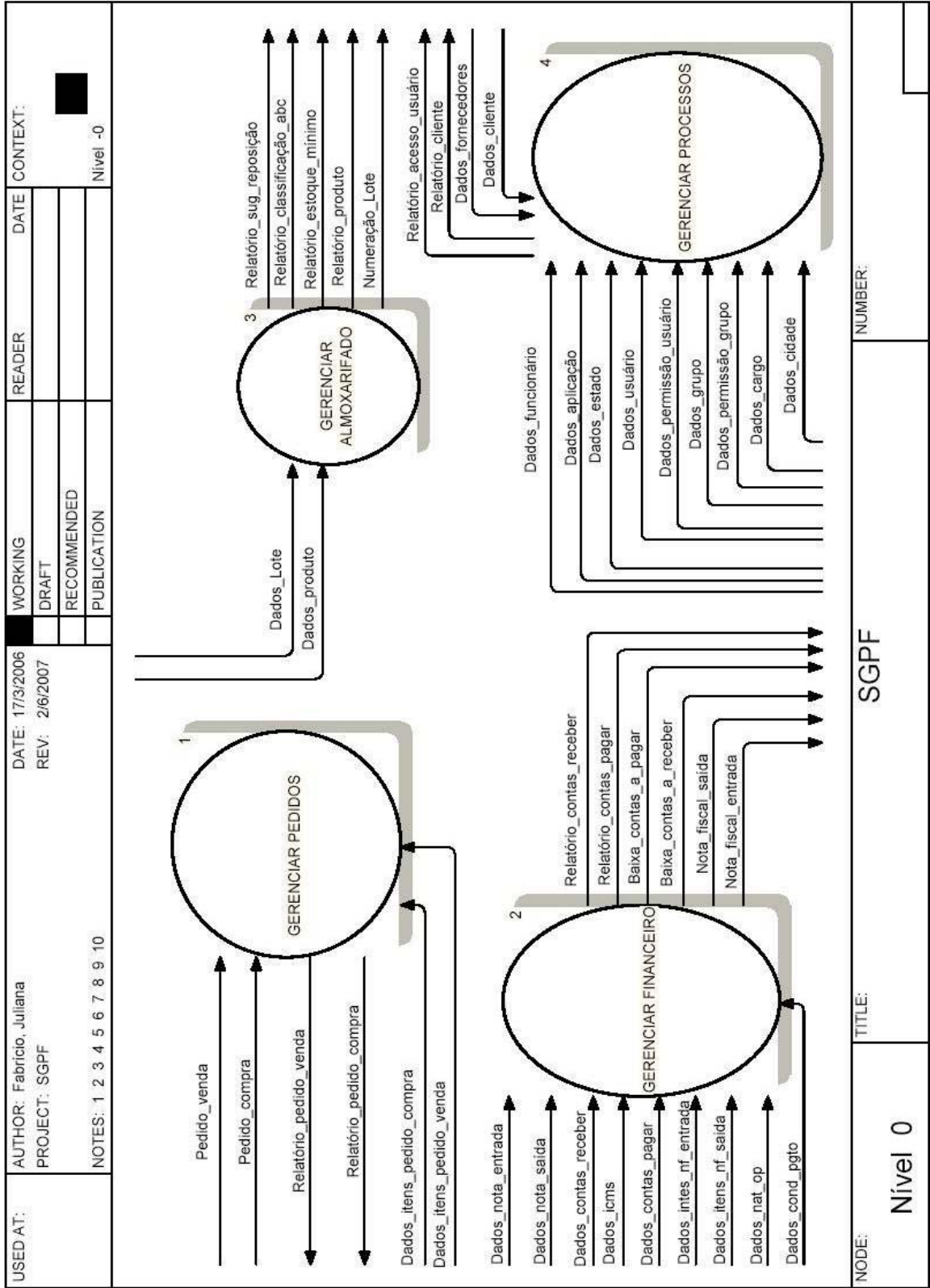
## 2.7 LIMITAÇÕES E RESTRIÇÕES IMPORTANTES EXISTENTES

Caso o *software* seja utilizado em farmácias com movimentação financeira, alguns campos contidos na tabela lote, tais como número do lote, quantidade, valor do lote, prazo de validade deverão estar desabilitados, para que o usuário ou administrador de entrada dos produtos no estoque através do lançamento dos valores relativos a nota fiscal de entrada.

Não será abordado o controle de pagamento de funcionários.

## 1. DIAGRAMA DE FLUXO DE DADOS (DFD)





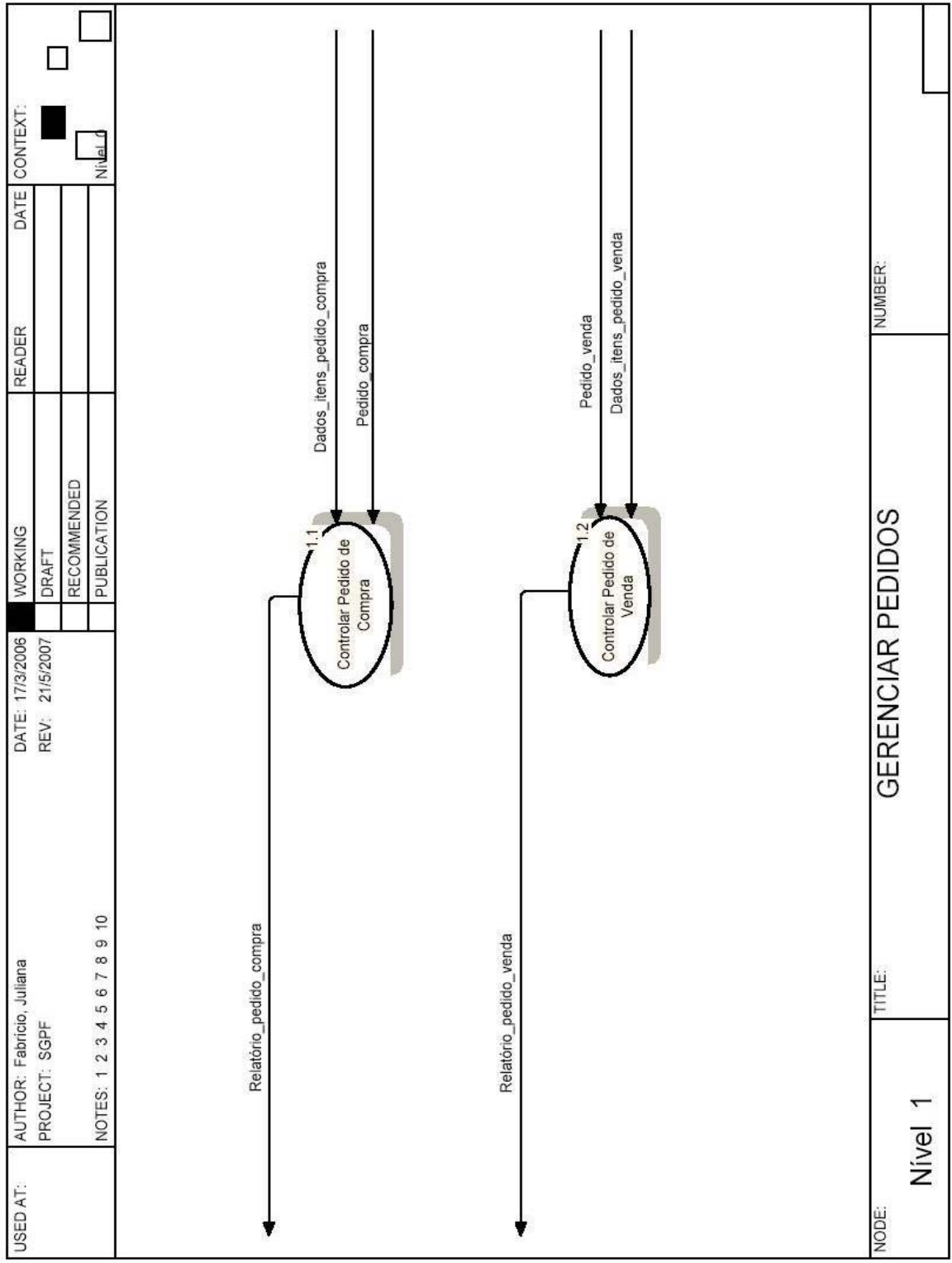
NUMBER:

SGPF

TITLE:

Nível 0

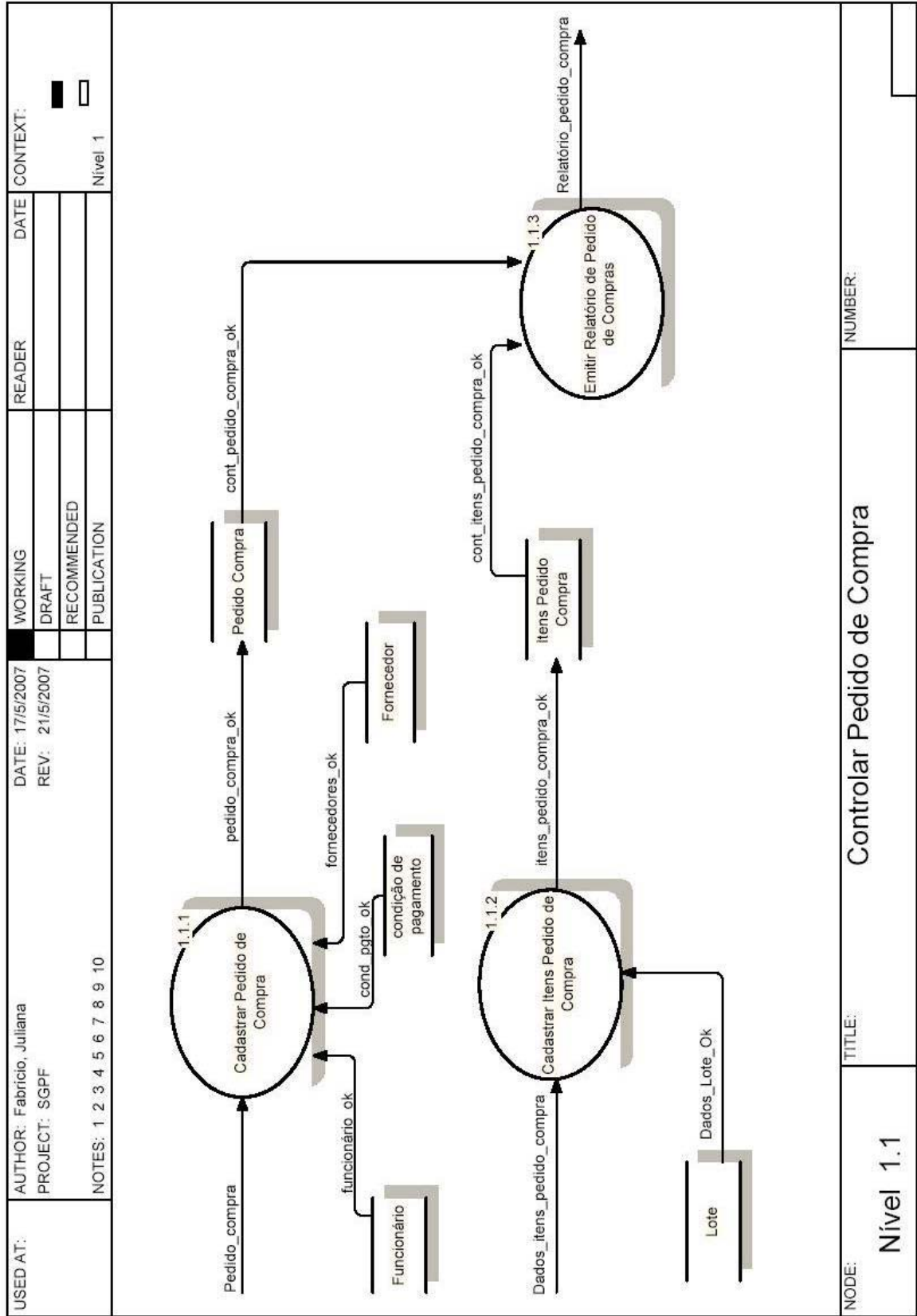




NODE: **Nível 1**

TITLE: **GERENCIAR PEDIDOS**

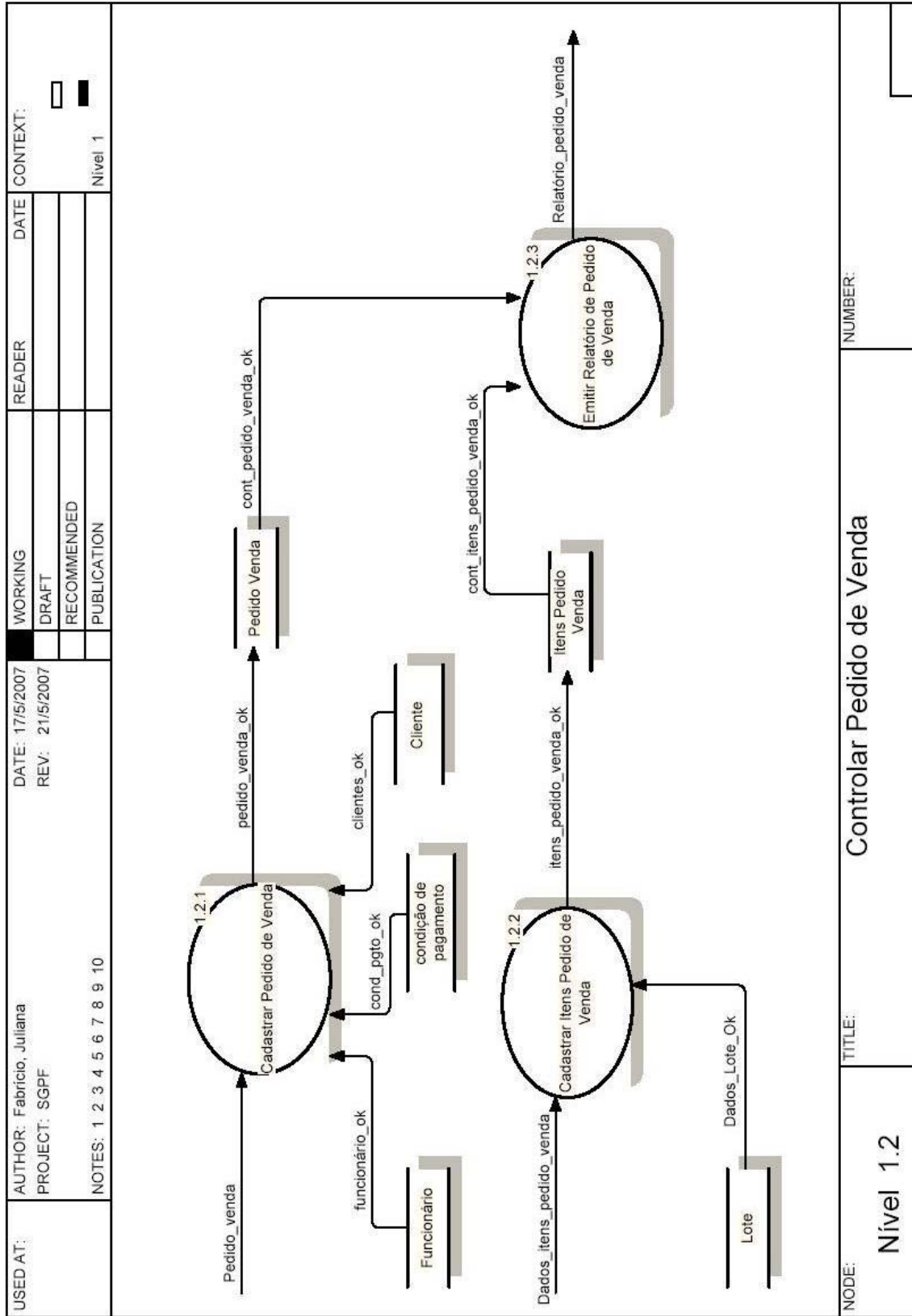
NUMBER:



Controlar Pedido de Compra

Nível 1.1

NUMBER:

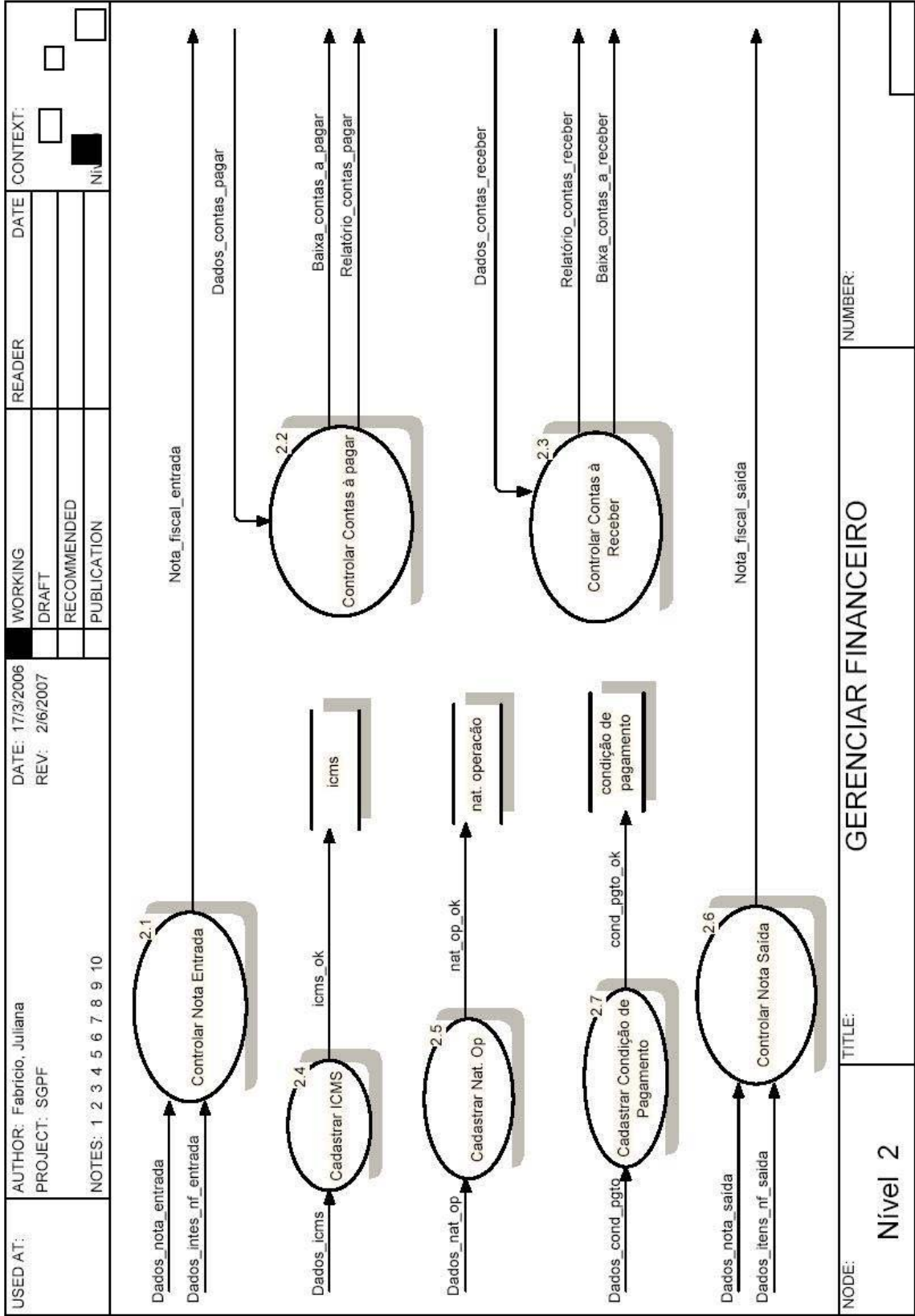


NUMBER:

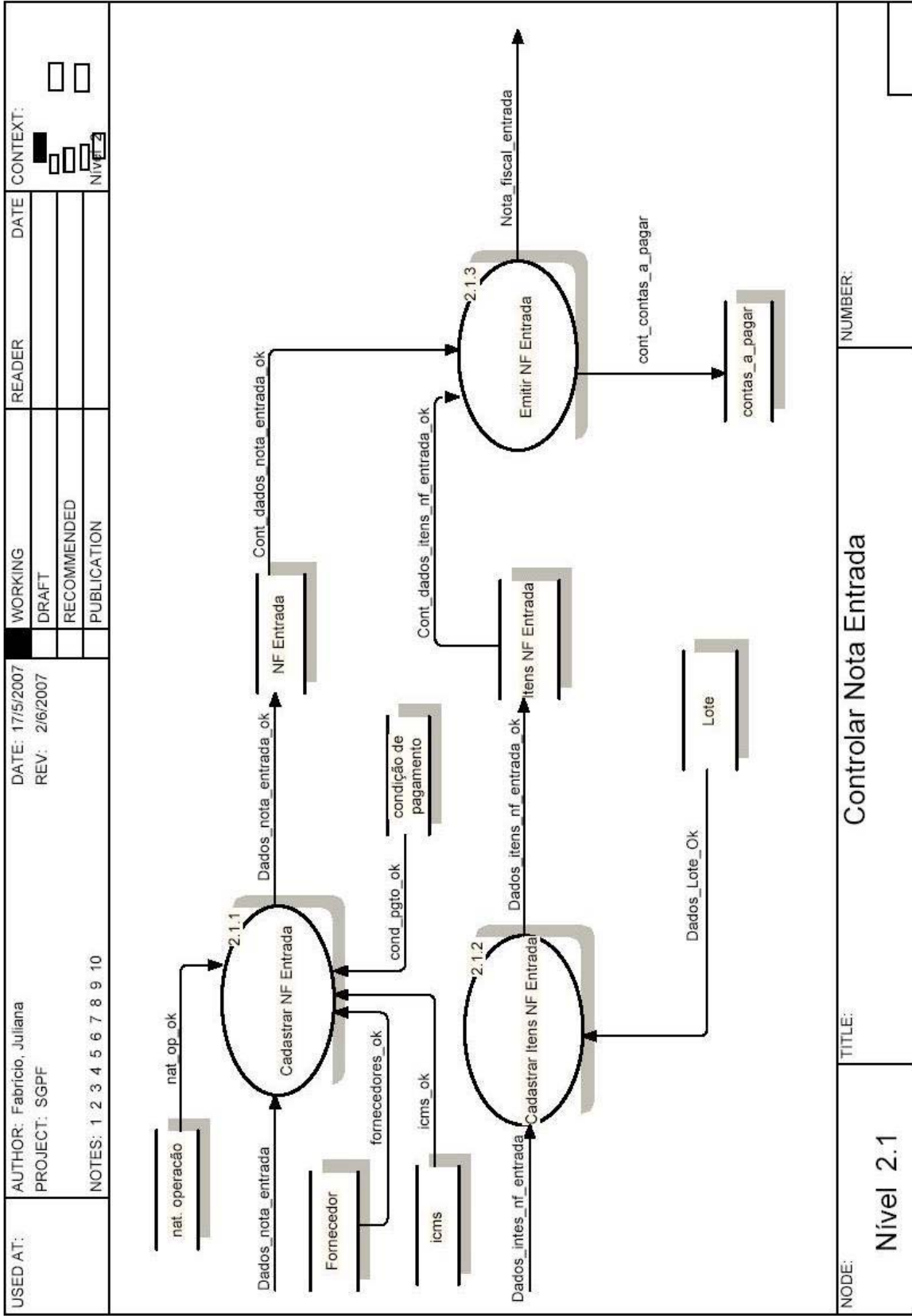
TITLE: **Controlar Pedido de Venda**

NOTE:

**Nível 1.2**



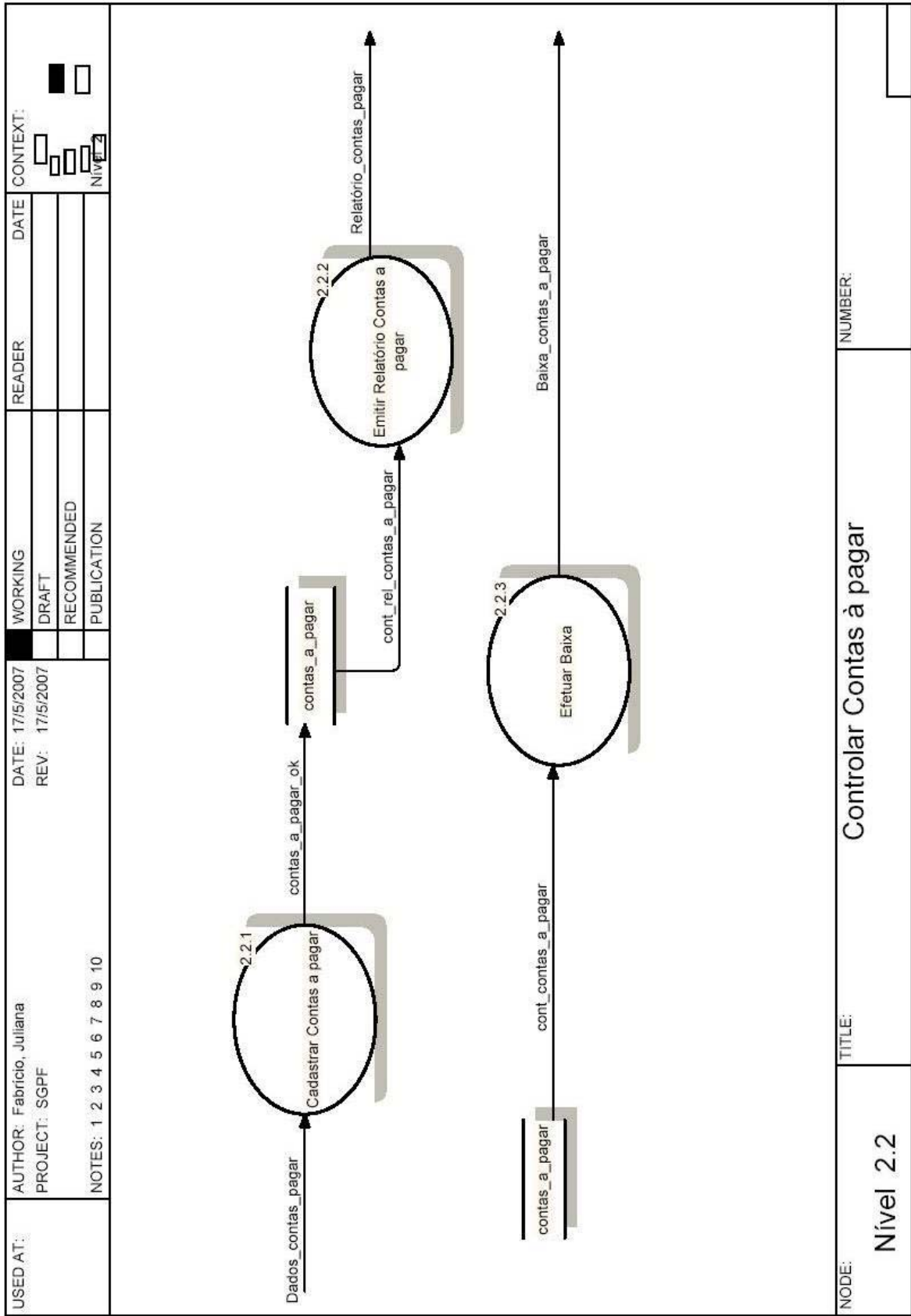
<p>NOTE: <b>Nível 2</b></p>	<p>TITLE: <b>GERENCIAR FINANCEIRO</b></p>	<p>NUMBER:</p>
-----------------------------	---	----------------



Controlar Nota Entrada

Nível 2.1

NUMBER:



NODE:

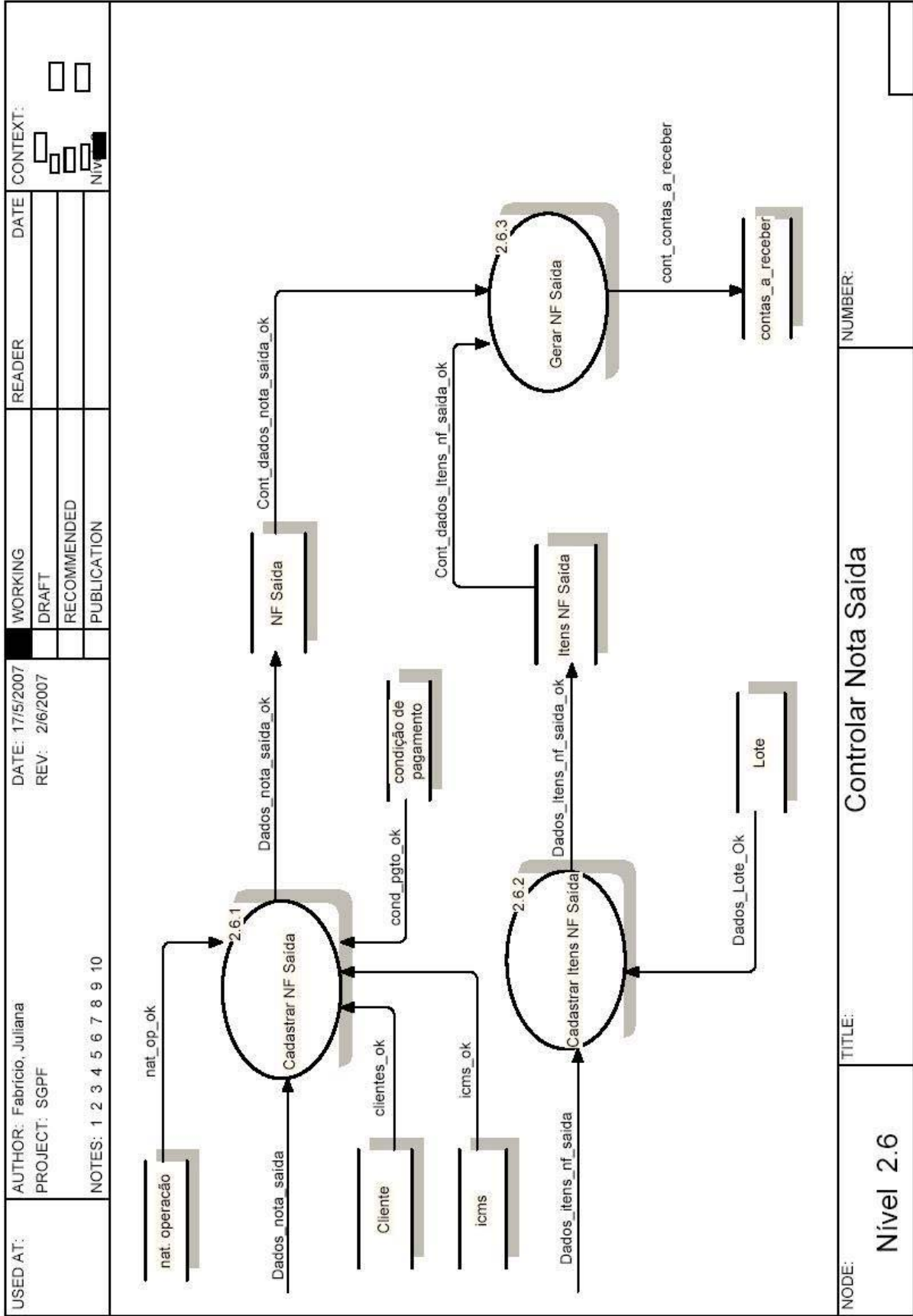
Nível 2.2

TITLE:

Controlar Contas à pagar

NUMBER:

USED AT:	AUTHOR: Fabrício, Juliana PROJECT: SGPF	DATE: 17/5/2007 REV: 17/5/2007	WORKING <input type="checkbox"/> DRAFT <input type="checkbox"/> RECOMMENDED <input type="checkbox"/> PUBLICATION	READER	DATE	CONTEXT:
NOTES: 1 2 3 4 5 6 7 8 9 10						
<pre>                     usecaseDiagram                         actor Actor1 as [Dados_contas_receber]                         actor Actor2 as [contas_a_receber]                         actor Actor3 as [contas_a_receber]                         actor Actor4 as [cont_rel_contas_a_receber]                         actor Actor5 as [contas_a_receber]                         usecase UC231 as 2.3.1 Cadastrar Contas a receber                         usecase UC232 as 2.3.2 Emitir Relatório Contas a Receber                         usecase UC233 as 2.3.3 Efetuar Baixa Ct Receber                         Actor1 --&gt; UC231                         Actor2 --&gt; UC231                         Actor2 --&gt; UC232                         Actor3 --&gt; UC232                         Actor4 --&gt; UC232                         Actor5 --&gt; UC233                         UC231 --&gt; UC232                         UC232 --&gt; UC233                     </pre>						
NODE:	Título: Controlar Contas à Receber					NUMBER:
Nível 2.3						



NODE: **Nível 2.6**      TITLE: **Controlar Nota Saída**      NUMBER: \_\_\_\_\_



USED AT:	AUTHOR: Fabricio, Juliana PROJECT: SGPF	DATE: 17/3/2006 REV: 16/5/2007	WORKING DRAFT RECOMMENDED PUBLICATION	READER	DATE	CONTEXT:
	NOTES: 1 2 3 4 5 6 7 8 9 10					Nível 3

Gerenciar_estoque	Relatório_estoque_minimo
3 1	Relatório_sug_reposicao
	Relatório_classificacao_abc
	Numeracao_Lote
	Relatório_produto

Dados\_produto

Dados\_Lote

Relatório\_estoque\_minimo

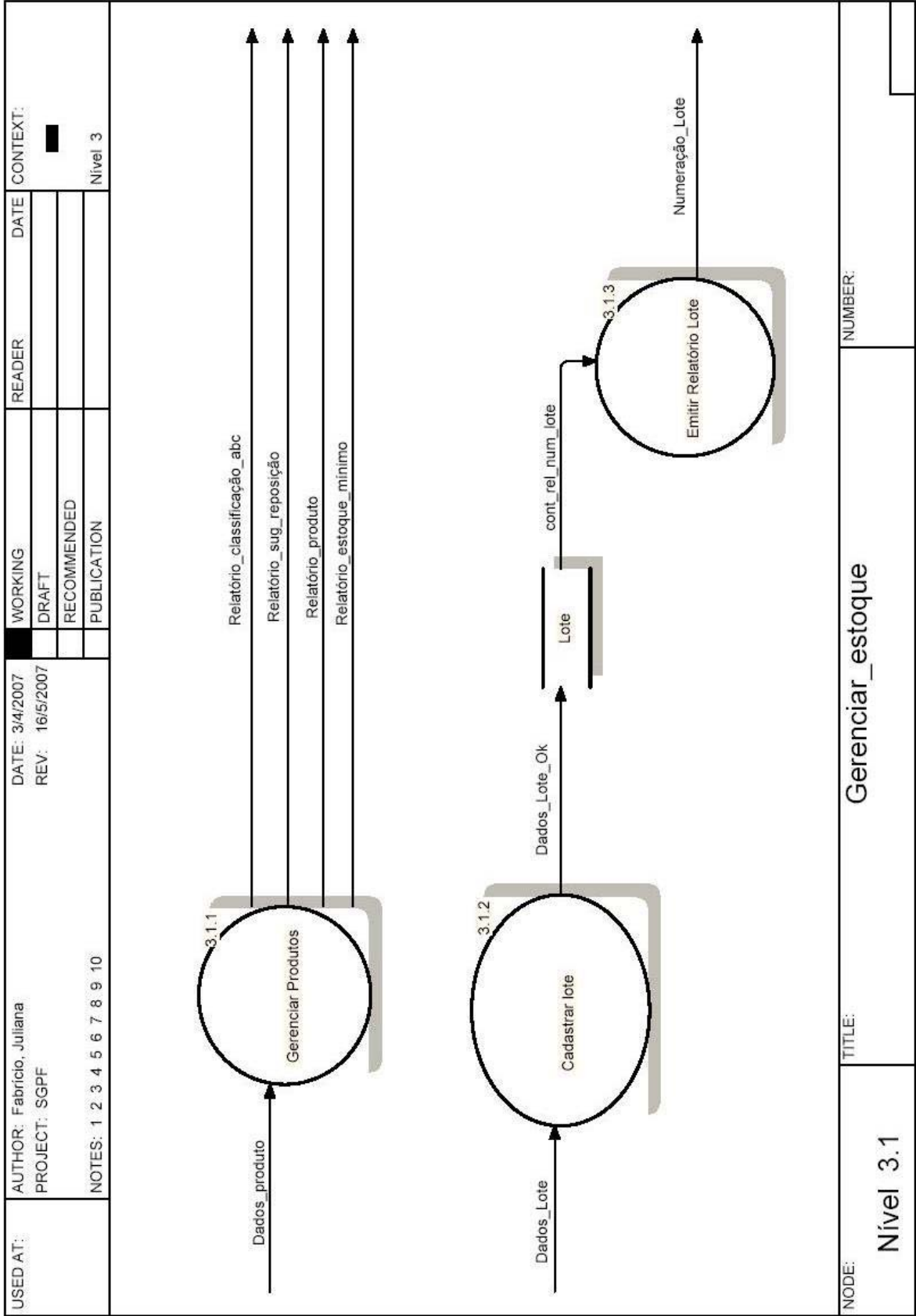
Relatório\_sug\_reposicao

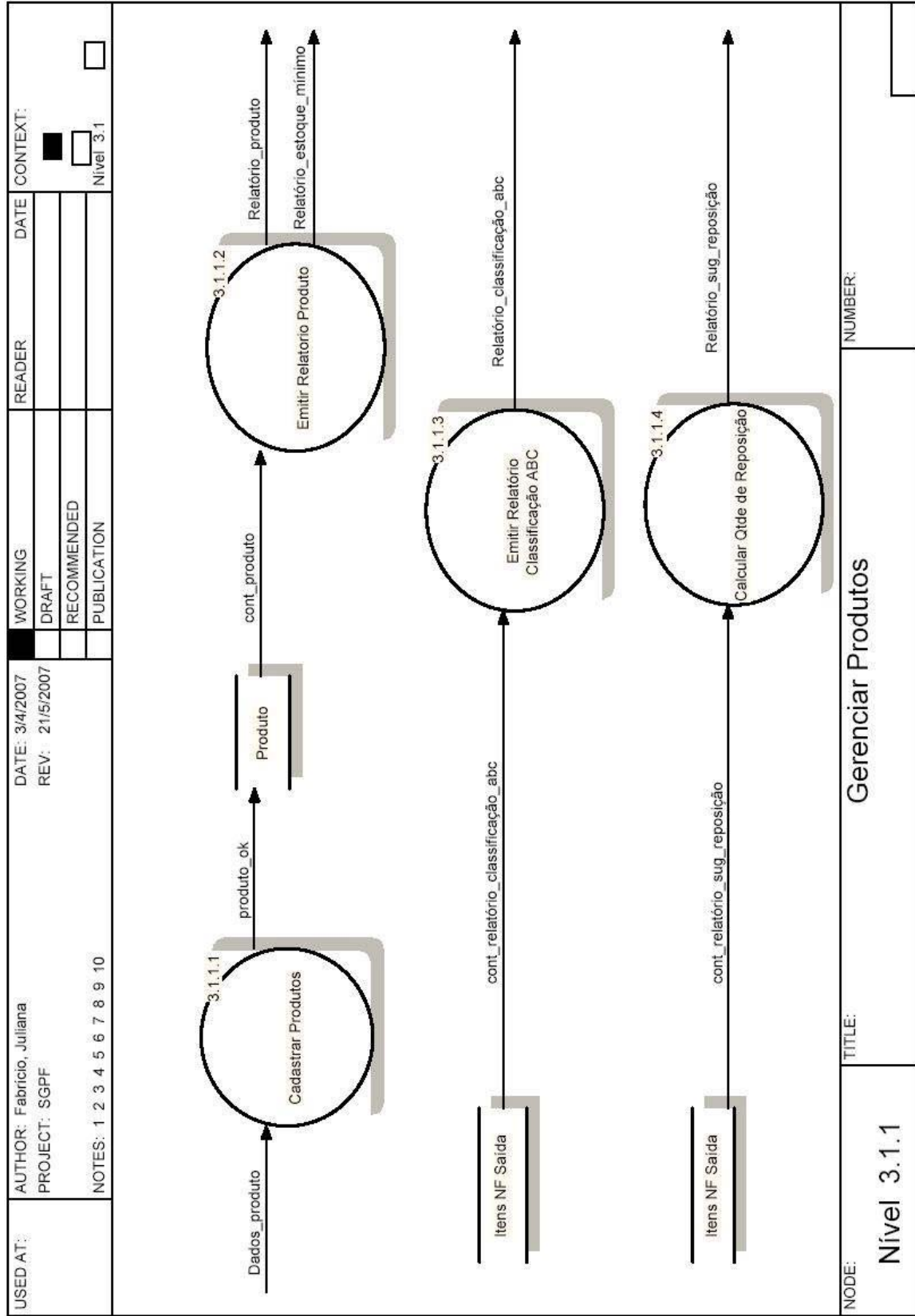
Relatório\_classificacao\_abc

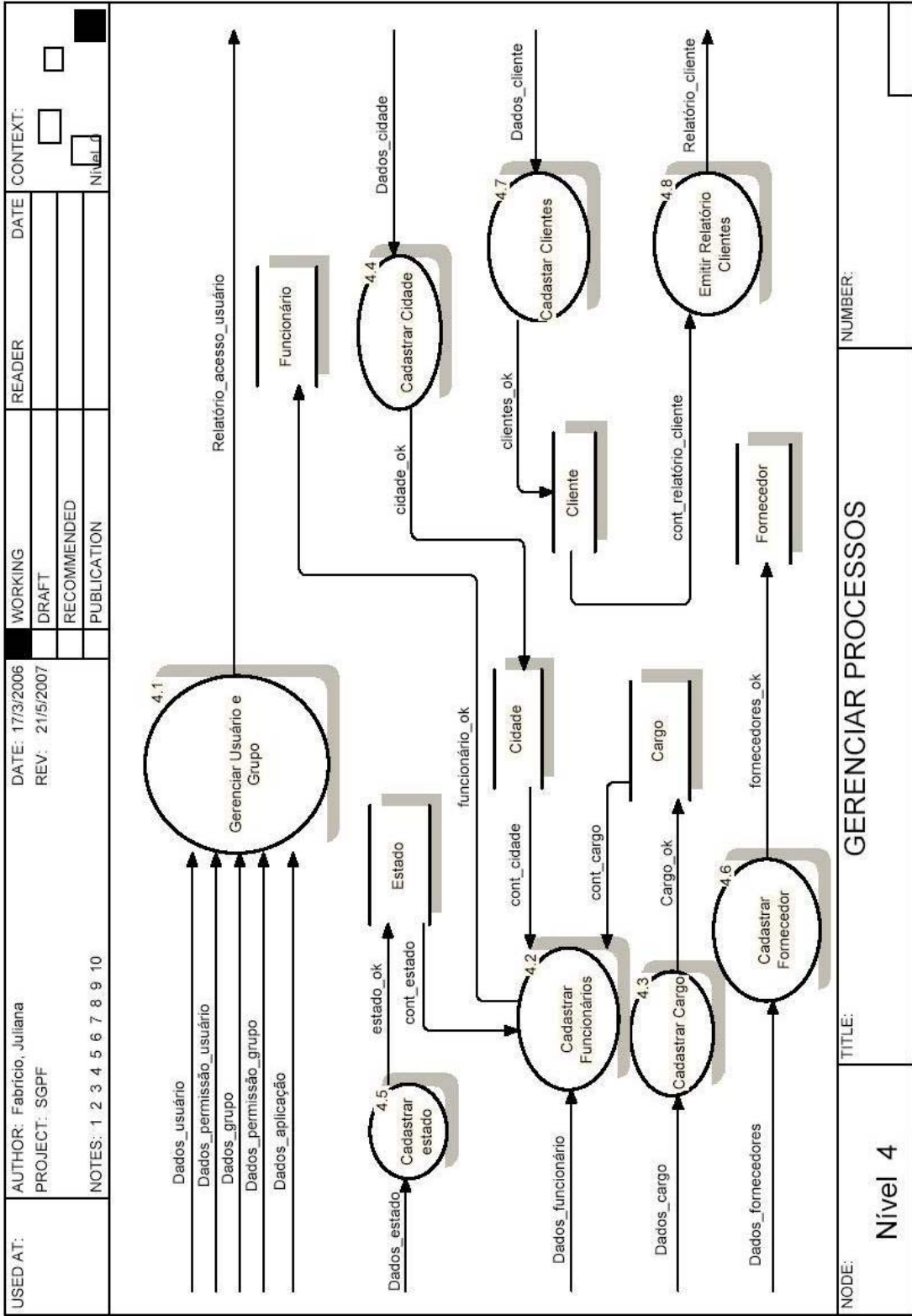
Numeracao\_Lote

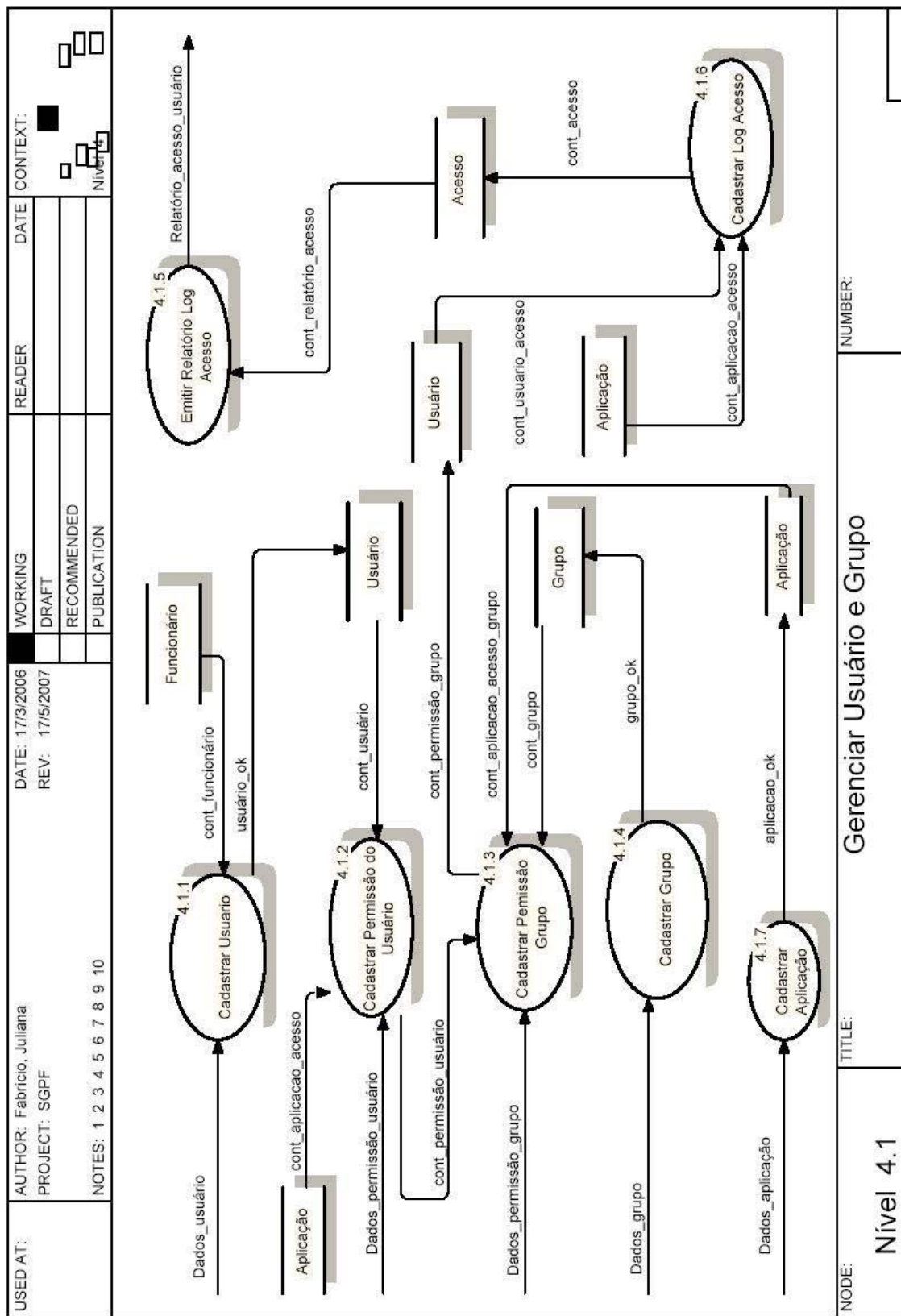
Relatório\_produto

Nível 3	GERENCIAR ALMOXARIFADO	NUMBER:
NODE:	TITLE:	

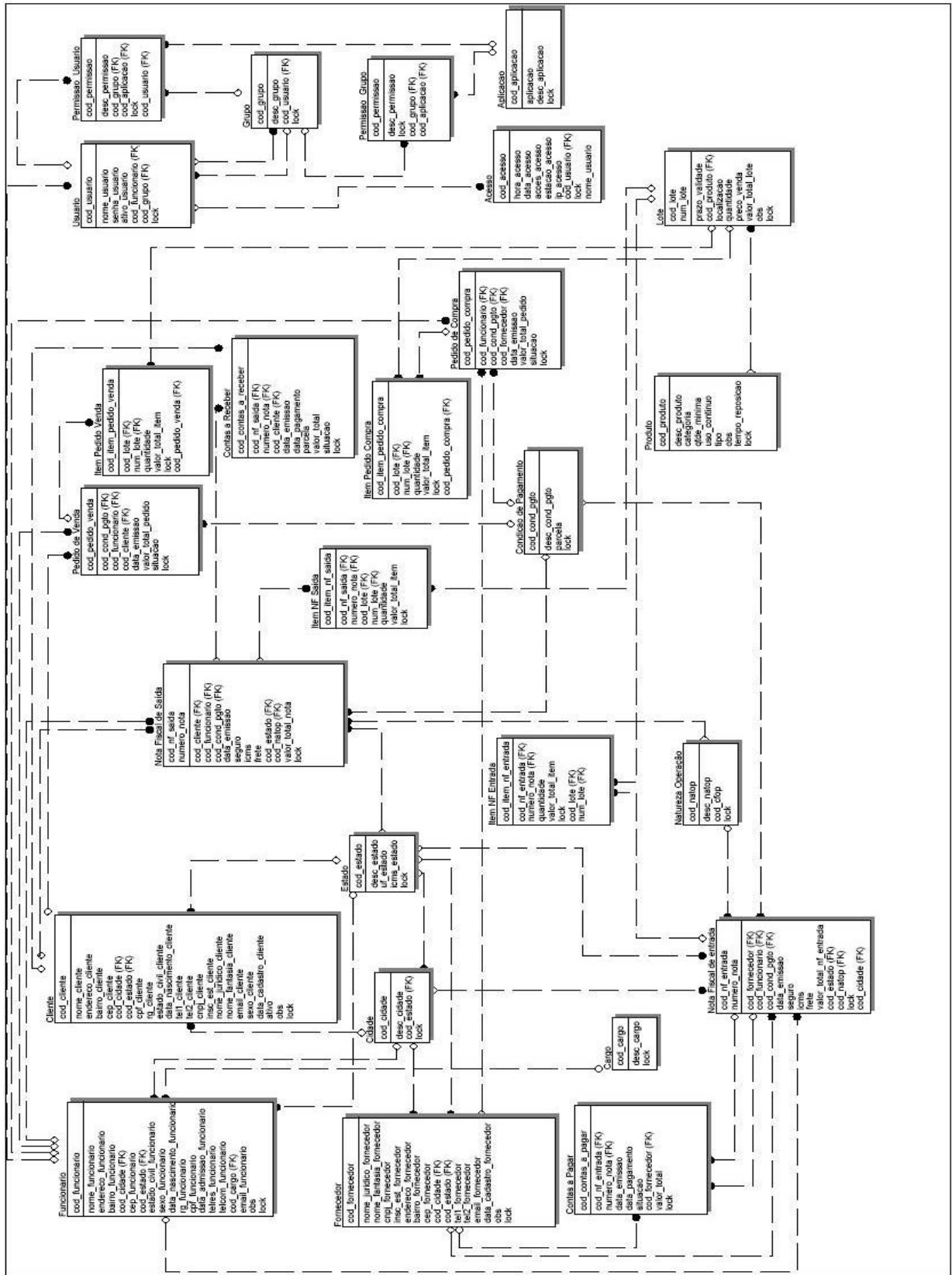








## **2. MODELO DE ENTIDADE E RELACIONAMENTO (MER)**



### **3. DICIONÁRIO DE DADOS DE DEPÓSITOS**



## A

**Acesso** = @cod\_acesso + cod\_usuario + hora\_acesso + data\_acesso + acoes\_acesso + estacao\_acesso + ip\_acesso

@cod\_acesso = [1-9999999999]  
 cod\_usuario = [1-9999999999]  
 hora\_acesso = [1 - 99:99:99]  
 data\_acesso = [1 - 99/99/9999]  
 acoes\_acesso = 1{[A-Z | 0-9 | ]}50  
 estacao\_acesso = 1{[A-Z | 0-9 | ]}50  
 ip\_acesso = 1{[A-Z | 0-9 | "." | "." ]}12

**Aplicação** = @ cod\_aplicacao + desc\_aplicacao

@cod\_aplicacao: [1-9999999999]  
 desc\_aplicacao:1{[A-Z | 0-9 | ]}30

## C

**Categoria** = @cod\_categoria + desc\_categoria

@cod\_categoria: [1-9999999999]  
 desc\_categoria:1{[A-Z | 0-9 | ]}30

**Cargo** = @cod\_cargo + desc\_cargo

@cod\_cargo = [1-9999999999]  
 desc\_cargo = 1{[A-Z | 0-9 | ]}40

**Cidade** = @cod\_cidade + desc\_cidade + cod\_estado

@cod\_cidade = [1-9999999999]  
 desc\_cidade = 1{[A-Z | 0-9 | ]}60  
 cod\_estado = [1-9999999999]

**Cliente** = @cod\_cliente + nome\_cliente + endereco\_cliente + bairro\_cliente + cep\_cliente + cod\_cidade + cod\_estado + cpf\_cliente + rg\_cliente + estado\_civil\_cliente + data\_nascimento\_cliente + tel1\_cliente + tel2\_cliente + cnpj\_cliente + inscr\_est\_cliente + nome\_juridico\_cliente + nome\_fantasia\_cliente + e-mail\_cliente + sexo\_cliente + data\_cadastro\_cliente + ativo

@cod\_cliente = [1-9999999999]

nome\_cliente = 1{A-Z | 0-9 | }50  
 endereco\_cliente = 1{[A-Z | 0-9 | ]} 40  
 bairro\_cliente = 1{[A-Z | 0-9 | ]} 40  
 cep\_cliente = 1{[A-Z | 0-9 | ]}10  
 cod\_cidade = [1-9999999999]

cod\_estado = [1-9999999999]  
 cpf\_cliente = 1{[A-Z| 0-9 | "." | "." ]}12  
 rg\_cliente = 1{[A-Z | 0-9 | "." | "." ]} 10  
 estado\_civil\_cliente =1{A-Z | 0-9 | }15  
 data\_nascimento\_cliente = [1 – 99/99/9999]  
 tel1\_cliente = 1{[0-9 | - | “ ( | “) ”]}15  
 tel2\_cliente = 1{[0-9 | - | “ ( | “) ”]}15  
 cnpj\_cliente = 1{[A-Z| 0-9 | "." | "." ]}20  
 inscr\_estadual = 1{[A-Z | 0-9 | "." | "." ]} 18  
 nome\_juridico\_cliente=1{[A-Z | 0-9 | "." | "." ]} 50  
 nome\_fantasia\_cliente=1{[A-Z | 0-9 | "." | "." ]} 50  
 email\_cliente = 1{[a-z | A-Z | 0-9 | ”-“ | “@” | “\_” | “. ”]}30  
 sexo\_cliente = 1{[A-Z | 0-9 | ]}1  
 data\_cadastro\_cliente = [1 – 99/99/9999]  
 ativo = 1{[A-Z | 0-9 | ]}1

**Condicao de Pagamento = @Cod\_condpagamento + desc\_cond\_pgto + parcela**

@cod\_cond\_pgto = [1-9999999999]  
 desc\_cond\_pgto = 1{[A-Z | 0-9 | ]}50  
 parcela = [1-9999999999]

**Contas a pagar = @cod\_contas\_a\_pagar + cod\_nf\_entrada + cod\_fornecedor + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao**

@ cod\_contas\_a\_pagar = [1-9999999999]  
 cod\_nf\_entrada = [1-9999999999]  
 cod\_fornecedor = [1-9999999999]  
 numero\_nota = [1-9999999999]  
 data\_emissao = [1 – 99/99/9999]  
 data\_pagamento = [1 – 99/99/9999]

valor\_total = [0,01-9999999999,99]  
 situacão = 1{[A-Z | 0-9 | ]}50

**Contas a receber = @cod\_contas\_a\_receber + cod\_nf\_saida + cod\_cliente + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao**

@ cod\_contas\_receber = [1-9999999999]  
 cod\_nf\_saida = [1-9999999999]  
 cod\_cliente = [1-9999999999]  
 numero\_nota = [1-9999999999]  
 data\_emissao = [1 - 99/99/9999]  
 data\_pagamento = [1 - 99/99/9999]  
 valor\_total = [0,01-9999999999,99]  
 situacao = 1{[A-Z | 0-9 | ]}50

### E

**Estado** = @cod\_estado + desc\_estado + uf\_estado + icms\_estado

@ cod\_estado = [1-9999999999]  
 desc\_estado = 1{[A-Z | 0-9 | ]}50  
 uf\_estado = 1{[A-Z | 0-9 | ]}2  
 icms\_estado = [1-9999999999]

### F

**Fornecedor** = @cod\_fornecedor + nome\_juridico\_fornecedor +  
 nome\_fantasia\_fornecedor + cnpj\_fornecedor + insc\_est\_fornecedor +  
 endereco\_fornecedor + bairro\_fornecedor + cep\_fornecedor + cod\_cidade\_fornecedor +  
 cod\_estado\_fornecedor + tel1\_fornecedor + tel2\_fornecedor + email\_fornecedor +  
 data\_cadastro\_fornecedor

@cod\_fornecedor = [1-9999999999]  
 nome\_juridico\_fornecedor = 1{[A-Z | 0-9 | ]}50  
 nome\_fantasia\_fornecedor = 1{[A-Z | 0-9 | ]}50  
 cnpj\_fornecedor = 1{[A-Z | 0-9 | "." | "-" ]} 20  
 insc\_est\_fornecedor = 1{[A-Z | 0-9 | "." | "-" ]} 18  
 endereco\_fornecedor = 1{[A-Z | 0-9 | ]}50  
 bairro\_fornecedor = 1{[A-Z | 0-9 | ]}50  
 cep\_fornecedor = 1{[A-Z | 0-9 | ]}50  
 cod\_cidade\_fornecedor = [1-9999999999]  
 cod\_estado\_fornecedor = [1-9999999999]  
 tel1\_fornecedor = 1{[0-9 | - | " (" | ")"]}15  
 tel2\_fornecedor = 1{[0-9 | - | " (" | ")"]}15  
 email\_fornecedor = 1{[a-z | A-Z | 0-9 | "." | "@" | "-" | "." ]}50  
 data\_cadatro\_fornecedor = [1 - 99/99/9999]

**Funcionário** = @cod\_funcionario + nome\_funcionario + endereço\_funcionario + bairro\_funcionario + cod\_cidade + cep\_funcionario + cod\_estado + estado\_civil\_funcionario + sexo\_funcionario + data\_nascimento\_funcionario + rg\_funcionario + cpf\_funcionario + data\_admissao\_funcionario + telres\_funcionario + telcom\_funcionario + cod\_cargo + e-mail\_funcionario

@cod\_funcionario = [1-9999999999]  
 nome\_funcionario = 1{[A-Z | 0-9 | ]}50  
 endereço\_funcionario = 1{[A-Z | 0-9 | ]} 40  
 bairro\_funcionario = 1{[A-Z | 0-9 | ]} 40  
 cod\_cidade = [1-9999999999]  
 cep\_funcionario = 1{[A-Z | 0-9 | ]}10  
 cod\_estado = = [1-9999999999]  
 estado\_civil\_funcionario = 1{[A-Z | 0-9 | ]}50  
 sexo\_funcionario = 1{[A-Z | 0-9 | ]}9  
 data\_nascimento\_funcionario= 1 – 99/99/9999]  
 rg\_cliente = 1{[A-Z | 0-9 | "." | "." ]} 10  
 cpf\_cliente = 1{[A-Z| 0-9 | "." | "." ]}12  
 data\_admissao\_funcionario = [1 – 99/99/9999]  
 telres\_funcionario = 1{[0-9 | - | " (" | ")"] }15  
 telcom\_funcionario = 1{[0-9 | - | " (" | ")"] }15  
 cod\_cargo = [1-9999999999]  
 e-mail\_funcionario = 1{[a-z | A-Z | 0-9 | "." | "@" | "\_" | "." ]}50

## G

**Grupo** = @cod\_grupo+ cod\_usuario + desc\_grupo

@cod\_grupo = [1-9999999999]  
 cod\_usuario = [1-9999999999]  
 desc\_grupo = 1{[A-Z | 0-9 | ]}30

## I

**Itens nf entrada** = @cod\_item\_nf\_entrada + cod\_nf\_entrada + numero\_nota + cod\_lote + num\_lote + quantidade + valor\_total\_item

@cod\_item\_nf\_entrada = [1 – 9999999999]  
 cod\_nf\_entrada = [1 – 9999999999]  
 numero\_nota = [1 – 9999999999]

cod\_lote= [1 – 9999999999]  
 num\_lote = [1 – 9999999999]  
 quantidade = [1 – 9999999999]  
 valor\_total\_item= [0 – 9.999.999,99]

**Itens nf saída** = @cod\_item\_nf\_saida + cod\_nf\_saida + numero\_nota + cod\_lote + num\_lote + quantidade + valor\_total\_item

@cod\_item\_nf\_saida = [1 – 9999999999]  
 cod\_nf\_saida = [1 – 9999999999]  
 numero\_nota = [1 – 9999999999]  
 cod\_lote= [1 – 9999999999]  
 num\_lote = [1 – 9999999999]  
 quantidade = [1 – 9999999999]  
 valor\_total\_item= [0 – 9.999.999,99]

**Itens pedido compra** = @cod\_item\_pedido\_compra + cod\_pedido\_compra + @cod\_lote + num\_lote + quantidade + valor\_total\_item

@cod\_item\_pedido\_compra = [1 – 9999999999]  
 cod\_pedido\_compra = [1 – 9999999999]  
 cod\_lote= [1 – 9999999999]  
 num\_lote = [1 – 9999999999]  
 quantidade = [1 – 9999999999]  
 valor\_total\_item= [0 – 9.999.999,99]

**Itens pedido venda**= @cod\_item\_pedido\_venda + cod\_pedido\_venda + @cod\_lote + num\_lote + quantidade + valor\_total\_item

@cod\_item\_pedido\_venda = [1 – 9999999999]  
 cod\_pedido\_venda = [1 – 9999999999]  
 cod\_lote= [1 – 9999999999]  
 num\_lote = [1 – 9999999999]  
 quantidade = [1 – 9999999999]  
 valor\_total\_item= [0 – 9.999.999,99]

L

**Lote** = @cod\_lote + num\_lote + prazo\_validade + cod\_produto + localizacao + quantidade + preco\_venda + valor\_total\_lote

cod\_lote = [1 – 9999999999]  
 num\_lote = [1 – 9999999999]  
 prazo\_validade = [1 – 99/99/9999]  
 cod\_produto = [1 – 9999999999]  
 localizacao = 1{[A – Z | a – z | . | , | "" | 0 – 9]} 40

quantidade = [1 – 9999999999]  
 preco\_custo = [0 – 9.999.999,99]  
 valor\_total\_lote = [0 – 9.999.999,99]

## N

**Nat\_ operação** = @cod\_natop + desc\_natop + cod\_cfop

cod\_natop = [1 – 9999999999]  
 cod\_cfop = [1 – 9999999999]  
 desc\_natop = 1{[A – Z | a – z | . | , | "" | 0 – 9]} 40

**Nf entrada** = @cod\_nf\_entrada + @numero\_nota + cod\_natop + cod\_cfop +  
 cod\_fornecedor + cnpj\_fornecedor + cod\_funcionario + cod\_cond\_pgto + data\_emissao  
 + data\_saida\_entrada + hora\_saida + base\_calculo\_icms + valor\_icms + valor\_frete +  
 valor\_seguro + valor\_total\_produto + valor\_total\_nf\_entrada

@cod\_nf\_entrada = [1 – 9999999999]  
 numero\_nota = [1 – 9999999999]  
 cod\_natop = [1 – 9999999999]  
 cod\_cfop = [1 – 9999999999]  
 cod\_fornecedor = [1 – 9999999999]  
 cnpj\_fornecedor = cnpj\_fornecedor = 1{[A-Z | 0-9 | "." | "." ]} 20  
 cod\_funcionario = [1 – 9999999999]  
 cod\_cond\_pgto = [1 – 9999999999]  
 data\_emissao = [1 – 99/99/9999]  
 data\_saida\_entrada = [1 – 99/99/9999]  
 hora\_saida = [1 – 99:99:99]  
 base\_calculo\_icms = [0 – 9.999.999,99]  
 valor\_icms = [0 – 9.999.999,99]  
 valor\_frete = [0 – 9.999.999,99]  
 valor\_seguro = [0 – 9.999.999,99]  
 valor\_total\_produto = [0 – 9.999.999,99]  
 valor\_total\_nf\_entrada = [0 – 9.999.999,99]

**Nf saída** = @cod\_nf\_saida + @numero\_nota + cod\_natop + cod\_cfop + cod\_cliente +  
 cnpj\_fornecedor + cpf\_cliente + cod\_funcionario + cod\_cond\_pgto + data\_emissao +  
 data\_saida\_entrada + hora\_saida + base\_calculo\_icms + valor\_icms + valor\_frete +  
 valor\_seguro + valor\_total\_produto + valor\_total\_nf\_saida

@cod\_nf\_saida = [1 – 9999999999]  
 numero\_nota = [1 – 9999999999]

cod\_natop = [1 – 9999999999]  
cod\_cfop = [1 – 9999999999]  
cod\_cliente = [1 – 9999999999]  
cnpj\_fornecedor = 1{[A-Z | 0-9 | “.” | “.” ]} 20  
cpf\_cliente = 1{[A-Z| 0-9 | “.” | “.” ]}12  
cod\_funcionario = [1 – 9999999999]  
cod\_cond\_pgto = [1 – 9999999999]  
data\_emissao = [1 – 99/99/9999]  
data\_saida\_entrada = [1 – 99/99/9999]  
hora\_saida = [1 – 99:99:99]  
base\_calculo\_icms = [0 – 9.999.999,99]  
valor\_icms = [0 – 9.999.999,99]  
valor\_frete = [0 – 9.999.999,99]  
valor\_seguro = [0 – 9.999.999,99]  
valor\_total\_produto = [0 – 9.999.999,99]  
valor\_total\_nf\_saida = [0 – 9.999.999,99]

## P

**Pedido Compra** = @cod\_pedido\_compra + cod\_funcionario + cod\_cond\_pgto +  
cod\_fornecedor + data\_emissao + valor\_total\_pedido + situacao

@cod\_pedido\_compra = [1 – 9999999999]  
cod\_funcionario = [1 – 9999999999]  
cod\_cond\_pgto = [1 – 9999999999]  
cod\_fornecedor = [1 – 9999999999]  
data\_emissao = [1 – 99/99/9999]  
valor\_total\_pedido = [0 – 9.999.999,99]  
situação = 1{[A-Z | 0-9 | ]}9

**Pedido Venda** = @cod\_pedido\_venda + cod\_funcionario + cod\_cond\_pgto +  
cod\_fornecedor + data\_emissao + valor\_total\_pedido + situacao

@cod\_pedido\_venda = [1 – 9999999999]  
cod\_funcionario = [1 – 9999999999]  
cod\_cond\_pgto = [1 – 9999999999]  
cod\_fornecedor = [1 – 9999999999]  
data\_emissao = [1 – 99/99/9999]  
valor\_total\_pedido = [0 – 9.999.999,99]  
situação = 1{[A-Z | 0-9 | ]}9

---

**Produto** = @cod\_produto+desc\_ produto +cod\_categoria + qtde\_minima +  
uso\_continuo + tipo + tempo\_reposicao

@cod\_prod: [1-9999999999]  
desc\_produto: 1{[A-Z | 0-9 | ]}30  
cod\_categoria: [1-9999999999]  
qtde\_minima: [1-9999999999]  
uso\_continuo: 1{[A - Z | a - z | . | , | "" | 0 - 9]} 40  
tipo; 1{[A - Z | a - z | . | , | "" | 0 - 9]} 40  
tempo\_reposicao: [1-9999999999]

## U

**Usuário** = @cod\_usuario + nome\_usuario + senha\_usuario + ativo + cod\_grupo +  
cod\_funcionario

@cod\_usuario = [1-9999999999]  
nome\_usuario = 1{[A-Z | 0-9 | ]}10  
senha\_usuario = 1{[A-Z | 0-9 | ]}20  
ativo = [1-9999999999]  
cod\_grupo = [1-9999999999]  
cod\_funcionario = [1-9999999999]



#### **4. DICIONÁRIO DE DADOS DOS FLUXOS**

---

**A**

**Aplicação\_ok** = @cod\_aplicacao + desc\_aplicacao + obs

**B**

**Baixa\_contas\_a\_pagar** = @cod\_contas\_a\_pagar + cod\_nf\_saida+ cod\_fornecedor + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Baixa\_contas\_a\_receber** = @@cod\_contas\_a\_receber + cod\_nf\_entrada + cod\_cliente + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**C**

**Cargo\_ok** = @cód\_cargo+descr\_cargo

**Cidade\_ok** = @cod\_cidade+ desc\_cidade+ cod\_estado

**Clientes\_ok** = @cod\_cliente + nome\_cliente + endereco\_cliente + bairro\_cliente + cep\_cliente + cod\_cidade + cod\_estado + cpf\_cliente + rg\_cliente + estado\_civil\_cliente + data\_nascimento\_cliente + tel1\_cliente + tel2\_cliente + cnpj\_cliente + inscr\_est\_cliente + nome\_juridico\_cliente + nome\_fantasia\_cliente + e-mail\_cliente + sexo\_cliente + data\_cadastro\_cliente + ativo

**Cond\_pgto\_ok** = @Cod\_cond\_pgto + desc\_cond\_pgto + parcela

**Cont\_acesso** = @cod\_acesso + cod\_usuario + hora\_acesso + data\_acesso + acoes\_acesso + estacao\_acesso + ip\_acesso

**Cont\_aplicação\_acesso** = @cod\_acesso + cod\_aplicacao + cod\_usuario

**Cont\_aplicação\_acesso\_grupo** = @cod\_acesso + cod\_aplicacao + cod\_grupo

---

**Cont\_cargo** = @cod\_cargo + desc\_cargo

**Cont\_cidade** = @cod\_cidade + desc\_cidade + cod\_estado

**Cont\_contas\_a\_pagar** = @cod\_contas\_a\_pagar + cod\_nf\_entrada + cod\_fornecedor + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Cont\_contas\_a\_receber** = @cod\_contas\_a\_receber + cod\_nf\_saida + cod\_cliente + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Cont\_dados\_itens\_nf\_entrada\_ok** = @cod\_item\_nf\_entrada + cod\_nf\_entrada + numero\_nota + cod\_lote + num\_lote + quantidade + valor\_total\_item

**Cont\_dados\_itens\_nf\_saida\_ok** = @cod\_item\_nf\_saida + cod\_nf\_saida + numero\_nota + cod\_lote + num\_lote + quantidade + valor\_total\_item

**Cont\_dados\_nota\_entrada\_ok** = @cod\_nf\_entrada + @numero\_nota + cod\_natop + cod\_cfop + cod\_fornecedor + cnpj\_fornecedor + cod\_funcionario + cod\_cond\_pgto + data\_emissao + data\_saida\_entrada + hora\_saida + base\_calculo\_icms + valor\_icms + valor\_frete + valor\_seguro + valor\_total\_produto + valor\_total\_nf\_entrada

**Cont\_dados\_nota\_saida\_ok** = @cod\_nf\_saida + @numero\_nota + cod\_natop + cod\_cfop + cod\_fornecedor + cnpj\_fornecedor + cod\_funcionario + cod\_cond\_pgto + data\_emissao + data\_saida\_entrada + hora\_saida + base\_calculo\_icms + valor\_icms + valor\_frete + valor\_seguro + valor\_total\_produto + valor\_total\_nf\_saida

**Cont\_estado** = @cod\_estado + uf\_estado + icms\_estado

**Cont\_relatorio\_fornecedor** = @cod\_fornecedor + nome\_juridico\_fornecedor + nome\_fantasia\_fornecedor + cnpj\_fornecedor + insc\_est\_fornecedor + endereco\_fornecedor + bairro\_fornecedor + cep\_fornecedor + cod\_cidade\_fornecedor + cod\_estado\_fornecedor + tel1\_fornecedor + tel2\_fornecedor + email\_fornecedor + data\_cadastro\_fornecedor

---

**Cont\_grupo** = @cod\_grupo + desc\_grupo + cod\_usuario

**Cont\_itens\_pedido\_compra\_ok** = @cod\_item\_pedido\_compra + cod\_pedido\_compra + @cod\_lote + num\_lote + quantidade + valor\_total\_item

**Cont\_itens\_pedido\_venda\_ok** = @cod\_item\_pedido\_venda + cod\_pedido\_venda + @cod\_lote + num\_lote + quantidade + valor\_total\_item

**Cont\_pedido\_compra\_ok** = @cod\_pedido\_compra + cod\_funcionario + cod\_cond\_pgto + cod\_fornecedor + data\_emissao + valor\_total\_pedido + situacao

**Cont\_pedido\_venda\_ok** = @cod\_pedido\_venda + cod\_funcionario + cod\_cond\_pgto + cod\_cliente + data\_emissao + valor\_total\_pedido + situacao

**Cont\_permissão\_grupo** = @cod\_grupo + cod\_permissao + desc\_permissao + cod\_aplicacao + cod\_grupo

**Cont\_permissão\_usuario** = @cod\_usuario + cod\_permissao + desc\_permissao + cod\_aplicacao + cod\_usuario + cod\_grupo

**Cont\_produto** = @cod\_produto+desc\_produto + cod\_categoria + qtde\_minima + uso\_continuo + tipo + tempo\_reposicao

**Cont\_relatório\_aceso** = @cod\_aceso + cod\_usuario + hora\_aceso + data\_aceso + acoes\_aceso + estacao\_aceso + ip\_aceso

**Cont\_rel\_contas\_a\_pagar** = @cod\_contas\_a\_pagar + cod\_nf\_entrada + cod\_fornecedor + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Cont\_rel\_contas\_a\_receber** = @cod\_contas\_a\_receber + cod\_nf\_saida+ cod\_cliente + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Cont\_rel\_num\_lote** = @cod\_lote + num\_lote + prazo\_validade + cod\_produto + localizacao + quantidade + preco\_custo + valor\_total\_lote

**Cont\_usuário** = @cod\_usuario + nome\_usuario + senha\_usuario + ativo + cod\_grupo + cod\_funcionario

**Cont\_usuário\_acesso** = @cod\_usuario + nome\_usuario + senha\_usuario + ativo + cod\_grupo + cod\_funcionario

**Contas\_a\_pagar\_ok** = @cod\_contas\_a\_pagar + cod\_nf\_entrada + cod\_fornecedor + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Contas\_a\_receber\_ok** = @cod\_contas\_a\_receber + cod\_nf\_saida + cod\_cliente + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

## D

**Dados\_aplicação** = @cod\_aplicacao + desc\_aplicacao + obs

**Dados\_cargo** = @cod\_cargo + desc\_cargo

**Dados\_cidade** = @cod\_cidade + desc\_cidade + cod\_estado

**Dados\_cliente** = @cod\_cliente + nome\_cliente + endereco\_cliente + bairro\_cliente + cep\_cliente + cod\_cidade + cod\_estado + cpf\_cliente + rg\_cliente + estado\_civil\_cliente + data\_nascimento\_cliente + tel1\_cliente + tel2\_cliente + cnpj\_cliente + inscr\_est\_cliente + nome\_juridico\_cliente + nome\_fantasia\_cliente + e-mail\_cliente + sexo\_cliente + data\_cadastro\_cliente + ativo

**Dados\_cond\_pgto** = @cod\_cond\_pgto + desc\_cond\_pgto + parcela

**Dados\_contas\_pagar** = @cod\_contas\_a\_pagar + cod\_nf\_entrada + cod\_fornecedor + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

---

**Dados\_contas\_receber** = @cod\_contas\_a\_receber + cod\_nf\_saida + cod\_cliente + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Dados\_estado** = @cod\_estado+ desc\_estado + uf\_estado + icms\_estado

**Dados\_fornecedores** = @cod\_fornecedor + nome\_juridico\_fornecedor + nome\_fantasia\_fornecedor + cnpj\_fornecedor + insc\_est\_fornecedor + endereco\_fornecedor + bairro\_fornecedor + cep\_fornecedor + cod\_cidade\_fornecedor + cod\_estado\_fornecedor + tel1\_fornecedor + tel2\_fornecedor + email\_fornecedor + data\_cadastro\_fornecedor

**Dados\_funcionário** = @cod\_funcionario + nome\_funcionario + endereço\_funcionario + bairro\_funcionario + cod\_cidade + cep\_funcionario + cod\_estado + estado\_civil\_funcionario + sexo\_funcionario + data\_nascimento\_funcionario + rg\_funcionario + cpf\_funcionario + data\_admissao\_funcionario + telres\_funcionario + telcom\_funcionario + cod\_cargo + e-mail\_funcionario

**Dados\_grupo** = @cod\_grupo + cod\_usuario + desc\_grupo

**Dados\_itens\_nf\_entrada** = @cod\_item\_nf\_entrada + cod\_nf\_entrada + numero\_nota + cod\_lote + num\_lote + quantidade + valor\_total\_item

**Dados\_itens\_nf\_entrada\_ok** = @cod\_item\_nf\_entrada + cod\_nf\_entrada + numero\_nota + cod\_lote + num\_lote + quantidade + valor\_total\_item

**Dados\_itens\_nf\_saida** = @cod\_item\_nf\_saida + cod\_nf\_saida + numero\_nota + cod\_lote + num\_lote + quantidade + valor\_total\_item

**Dados\_itens\_nf\_saida\_ok** = @cod\_item\_nf\_saida + cod\_nf\_saida + numero\_nota + cod\_lote + num\_lote + quantidade + valor\_total\_item

**Dados\_itens\_pedido\_compra** = @cod\_item\_pedido\_compra + cod\_pedido\_compra + @cod\_lote + num\_lote + quantidade + valor\_total\_item

**Dados\_itens\_pedido\_venda** = @cod\_item\_pedido\_venda + cod\_pedido\_venda + @cod\_lote + num\_lote + quantidade + valor\_total\_item

**Dados\_Lote** = @cod\_lote + num\_lote + prazo\_validade + cod\_produto + localizacao + quantidade + preco\_custo + valor\_total\_lote

**Dados\_Lote\_ok** = @cod\_lote + num\_lote + prazo\_validade + cod\_produto + localizacao + quantidade + preco\_custo + valor\_total\_lote

**Dados\_nat\_op** = @cod\_natop + cod\_cfop

**Dados\_nota\_entrada** = @cod\_nf\_entrada + @numero\_nota + cod\_natop + cod\_cfop + cod\_fornecedor + cnpj\_fornecedor + cod\_funcionario + cod\_cond\_pgto + data\_emissao + data\_saida\_entrada + hora\_saida + base\_calculo\_icms + valor\_icms + valor\_frete + valor\_seguro + valor\_total\_produto + valor\_total\_nf\_entrada

**Dados\_nota\_saida** = @cod\_nf\_saida + @numero\_nota + cod\_natop + cod\_cfop + cod\_cliente + cnpj\_fornecedor + cpf\_cliente + cod\_funcionario + cod\_cond\_pgto + data\_emissao + data\_saida\_entrada + hora\_saida + base\_calculo\_icms + valor\_icms + valor\_frete + valor\_seguro + valor\_total\_produto + valor\_total\_nf\_saida

**Dados\_permissão\_grupo** = @cod\_permissao + desc\_permissao + cod\_grupo + cod\_aplicacao

**Dados\_permissão\_usuario** = @cod\_permissao + desc\_permissao + cod\_grupo + cod\_usuario + cod\_aplicacao

**Dados\_produto** = @cod\_produto + desc\_produto + cod\_categoria + qtde\_minima + uso\_continuo + tipo + tempo\_reposicao

**Dados\_usuario** = @cod\_usuario + cod\_funcionario + cod\_grupo + cod\_permissao + nome\_usuario + senha\_usuario + ativo\_usuario

---

**E**

**Estado\_ok** = @cod\_estado + nome\_estado + uf\_estado + icms\_estado

**F**

**Fornecedores\_ok** = @cod\_fornecedor + nome\_juridico\_fornecedor +  
nome\_fantasia\_fornecedor + cnpj\_fornecedor + insc\_est\_fornecedor +  
endereco\_fornecedor + bairro\_fornecedor + cep\_fornecedor + cod\_cidade\_fornecedor +  
cod\_estado\_fornecedor + tel1\_fornecedor + tel2\_fornecedor + email\_fornecedor +  
data\_cadastro\_fornecedor

**Funcionário\_ok** = @cod\_funcionario + nome\_funcionario + endereco\_funcionario +  
bairro\_funcionario + cod\_cidade + cep\_funcionario + cod\_estado +  
estado\_civil\_funcionario + sexo\_funcionario + data\_nascimento\_funcionario +  
rg\_funcionario + cpf\_funcionario + data\_admissao\_funcionario + telres\_funcionario +  
telcom\_funcionario + cod\_cargo + e-mail\_funcionario

**G**

**Grupo\_ok** = @cod\_grupo + @cod\_usuario + desc\_grupo

**I**

**Itens\_pedido\_compra\_ok** = @cod\_item\_pedido\_compra + cod\_pedido\_compra +  
@cod\_lote + num\_lote + quantidade + valor\_total\_item

**Itens\_pedido\_venda\_ok** = @cod\_item\_pedido\_venda + cod\_pedido\_venda +  
@cod\_lote + num\_lote + quantidade + valor\_total\_item

**N**



---

**Nat\_op\_ok** = @cod\_natop + cod\_cfop

**Nota\_fiscal\_entrada** = @cod\_nf\_entrada + @numero\_nota + cod\_natop + cod\_cfop + cod\_fornecedor + cnpj\_fornecedor + cod\_funcionario + cod\_cond\_pgto + data\_emissao + data\_saida\_entrada + hora\_saida + base\_calculo\_icms + valor\_icms + valor\_frete + valor\_seguro + valor\_total\_produto + valor\_total\_nf\_entrada

**Nota\_fiscal\_saida** = @cod\_nf\_saida + @numero\_nota + cod\_natop + cod\_cfop + cod\_cliente + cnpj\_fornecedor + cpf\_cliente + cod\_funcionario + cod\_cond\_pgto + data\_emissao + data\_saida\_entrada + hora\_saida + base\_calculo\_icms + valor\_icms + valor\_frete + valor\_seguro + valor\_total\_produto + valor\_total\_nf\_saida

**Numeração\_Lote** = @cod\_lote + num\_lote + prazo\_validade + cod\_produto + localizacao + quantidade + preco\_custo + valor\_total\_lote

## P

**Pedido\_compra** = @cod\_pedido\_compra + cod\_funcionario + cod\_cond\_pgto + cod\_fornecedor + data\_emissao + valor\_total\_pedido + situação

**Pedido\_compra\_ok** = @cod\_pedido\_compra + cod\_funcionario + cod\_cond\_pgto + cod\_fornecedor + data\_emissao + valor\_total\_pedido + situacao

**Pedido\_venda** = @cod\_pedido\_venda + cod\_funcionario + cod\_cond\_pgto + cod\_cliente + data\_emissao + valor\_total\_pedido + situacao

**Pedido\_venda\_ok** = @cod\_pedido\_venda + cod\_funcionario + cod\_cond\_pgto + cod\_cliente + data\_emissao + valor\_total\_pedido + situacao

**Produto\_ok** = @cod\_produto + desc\_produto + cod\_categoria + qtde\_minima + uso\_continuo + tipo + tempo\_reposicao

---

**R**

**Relatório\_ acesso\_usuario** = @cod\_acesso + cod\_usuario + hora\_acesso + data\_acesso + acoes\_acesso + estacao\_acesso + ip\_acesso

**Relatório\_classificação\_abc** = @ cod\_nf\_saida + @cod\_item\_nf\_saida + @cod\_produto + desc\_produto + qtde + valor\_total\_item\_nf\_saida + valor\_total\_nf\_saida

**Relatório\_cliente** = @cod\_cliente + nome\_cliente + endereco\_cliente + bairro\_cliente + cep\_cliente + cod\_cidade + cod\_estado + cpf\_cliente + rg\_cliente + estado\_civil\_cliente + data\_nascimento\_cliente + tel1\_cliente + tel2\_cliente + cnpj\_cliente + inscr\_est\_cliente + nome\_juridico\_cliente + nome\_fantasia\_cliente + e-mail\_cliente + sexo\_cliente + data\_cadastro\_cliente + ativo

**Relatório\_contas\_pagar** = @cod\_contas\_a\_pagar + cod\_nf\_entrada + cod\_fornecedor + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Relatório\_contas\_receber** = @cod\_contas\_a\_receber + cod\_nf\_saida + cod\_cliente + numero\_nota + data\_emissao + data\_pagamento + valor\_total + situacao

**Relatório\_fornecedor** = @cod\_fornecedor + nome\_juridico\_fornecedor + nome\_fantasia\_fornecedor + cnpj\_fornecedor + insc\_est\_fornecedor + endereco\_fornecedor + bairro\_fornecedor + cep\_fornecedor + cod\_cidade\_fornecedor + cod\_estado\_fornecedor + tel1\_fornecedor + tel2\_fornecedor + email\_fornecedor + data\_cadastro\_fornecedor

**Relatório\_estoque\_minimo** = @cod\_lote + num\_lote + cod\_produto + desc\_produto + cod\_categoria + qtde\_minima + uso\_continuo + tipo + tempo\_reposicao

**Relatório\_pedido\_compra** = @cod\_pedido\_compra + cod\_funcionario + cod\_cond\_pgto + cod\_fornecedor + data\_emissao + valor\_total\_pedido + situação

**Relatório\_pedido\_venda** = @cod\_pedido\_venda + cod\_funcionario + cod\_cond\_pgto + cod\_cliente + data\_emissao + valor\_total\_pedido + situacao

**Relatório\_produto** = @cod\_produto + desc\_produto + preço\_prod + qtde

**Relatório\_produto\_vencimentos** = @cod\_lote + num\_lote + prazo\_validade + cod\_produto + localizacao + quantidade + preco\_venda + valor\_total\_lote

**Relatório\_sug\_reposição** = @cod\_prod + desc\_prod + qtde + meses

## U

**Usuário\_ok** = @cod\_usuario + cod\_funcionario + cod\_grupo + cod\_permissao  
nome\_usuario + senha\_usuario + ativo\_usuario

## **5. MINI-ESPECIFICAÇÃO**

## NÍVEL 1

### Processo 1.1 - Controlar Pedido de Compra

```

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_pedido_compra
PROCURE cod_pedido_compra em Pedido de Compra
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_fornecedor em Fornecedor
    PROCURE cod_cond_pgto em Condicao de Pagamento
    PROCURE cod_funcionario em Funcionário
    PROCURE cod_item_pedido_compra em Item Pedido Compra
    EXIBA Pedido_Compra
  CASO Opcao = 'Alterar'
    ACESSE Pedido_Compra
    ACESSE Dados_Itens_pedido_compra
    REGRAVE Pedido_Compras_ok em Pedido de Compra
  CASO Opcao = 'Excluir'
    APAGUE Pedido de Compra
    APAGUE Dados_Itens_pedido_compra
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Pedido_Compra
  GERE cod_pedido_compra = cod_pedido_compra + 1
  ACESSE Pedido_Compra
  GRAVE compras_ok em Pedido_Compra
FIM SE

```

### Processo 1.1.1 - Cadastrar Pedido de Compra

```

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_pedido_compra
PROCURE cod_pedido_compra em Pedido de Compra
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_fornecedor em Fornecedor
    PROCURE cod_cond_pgto em Condicao de Pagamento
    PROCURE cod_funcionario em Funcionário
    EXIBA Pedido_Compra
  CASO Opcao = 'Alterar'
    ACESSE Pedido_Compra

```

```

                REGRAVE Pedido_Compras_ok em Pedido de Compra
            CASO Opcao = 'Excluir'
                APAGUE Pedido de Compra
            FIM CASO
        CASO CONTRARIO
            POSICIONE no fim de Pedido_Compra
            GERE cod_pedido_compra = cod_pedido_compra + 1
            ACESSE Pedido_Compra
            GRAVE compras_ok em Pedido_Compra
        FIM SE

```

### **Processo 1.1.2 - Cadastrar Itens Pedido de Compra**

```

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_item_pedido_compra
PROCURE cod_item_pedido_compra em Pedido de Compra
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_item_pedido_compra em Item Pedido Compra
            EXIBA Pedido_Compra
        CASO Opcao = 'Alterar'
            ACESSE Dados_Itens_pedido_compra
            REGRAVE Pedido_Compras_ok em Pedido de Compra
        CASO Opcao = 'Excluir'
            APAGUE Dados_Itens_pedido_compra
        FIM CASO
    CASO CONTRARIO
        POSICIONE no fim de Pedido_Compra
        GERE cod_item_pedido_compra = cod_item_pedido_compra + 1
        ACESSE Pedido_Compra
        GRAVE compras_ok em Pedido_Compra
    FIM SE

```

### **Processo 1.1.3 – Emitir Relatório de Pedido de Compra**

```

DETERMINE relatorio = ['Todos', 'Cod']
POCISONE inicio de Pedido_de_Compra
ACESSE relatorio
SE relatorio = 'Cod'
    ENQUANTO nao atingir fim de Pedido_de_Compra
        PROCURE cod_fornecedor em Fornecedor
        PROCURE cod_cond_pgto em Condicao de Pagamento
        PROCURE cod_funcionario em Funcionário
        PROCURE cod_item_pedido_compra em Item Pedido Compra
    FIM ENQUANTO

```

```

                IMPRIMA cont_relatorio_pedido_compra
            POSICIONE proximo Pedido de Compras
        FIM ENQUANTO
    FIM SE
    SE relatorio = 'Todos'
        ENQUANTO nao atingir fim de Pedido de Compras
            PROCURE cod_fornecedor em Fornecedor
            PROCURE cod_cond_pgto em Condição de Pagamento
            PROCURE cod_funcionario em Funcionário
            PROCURE cod_item_pedido_compra em Item Pedido Compra
                IMPRIMA cont_relatorio_pedido_compra
            POSICIONE proximo Pedido_de_Compras
        FIM ENQUANTO
    SENA0
        ACESSE cod_pedido_compra
        PROCURE cod_pedido_compra em Pedido de Compras
        SE ENCONTROU
            IMPRIMA cont_relatorio_pedido_compra
        FIM SE
    FIM SE

```

### **Processo 1.2 – Controlar Pedido de Venda**

```

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_pedido_vendas
PROCURE cod_pedido_vendas em Pedido de Venda
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_cliente em Cliente
            PROCURE cod_cond_pgto em Condição de Pagamento
            PROCURE cod_funcionario em Funcionário
            PROCURE cod_item_pedido_venda em Item Pedido Venda
            EXIBA Pedido_venda
        CASO Opcao = 'Alterar'
            ACESSE Pedido_venda
            ACESSE Dados_Itens_pedido_venda
            REGRAVE Pedido_venda_ok em Pedido de Venda
        CASO Opcao = 'Excluir'
            APAGUE Pedido_venda
            APAGUE Dados_Itens_pedido_venda
        FIM CASO
    CASO CONTRARIO
        POSICIONE no fim de Pedido_venda
        GERE cod_pedido_venda = cod_pedido_venda + 1
        ACESSE Pedido_Venda

```

GRAVE Pedido\_venda\_ok em Pedido de Venda  
 FIM SE

### **Processo 1.1.2 – Cadastrar Pedido de Venda**

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']  
 ACESSE cod\_pedido\_vendas  
 PROCURE cod\_pedido\_vendas em Pedido de Venda  
 SE ENCONTROU  
   ACESSE Opcao  
     CASO Opcao = 'Consultar'  
       PROCURE cod\_cliente em Cliente  
       PROCURE cod\_cond\_pgto em Condicao de Pagamento  
       PROCURE cod\_funcionario em Funcionário  
       EXIBA Pedido\_venda  
     CASO Opcao = 'Alterar'  
       ACESSE Pedido\_venda  
       REGRAVE Pedido\_venda\_ok em Pedido de Venda  
     CASO Opcao = 'Excluir'  
       APAGUE Pedido\_venda  
   FIM CASO  
 CASO CONTRARIO  
   POSICIONE no fim de Pedido\_venda  
   GERE cod\_pedido\_venda = cod\_pedido\_venda + 1  
   ACESSE Pedido\_Venda  
   GRAVE Pedido\_venda\_ok em Pedido de Venda  
 FIM SE

### **Processo 1.1.2 - Cadastrar Itens Pedido de Venda**

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']  
 ACESSE cod\_item\_pedido\_venda  
 PROCURE cod\_item\_pedido\_venda em Pedido de Venda  
 SE ENCONTROU  
   ACESSE Opcao  
     CASO Opcao = 'Consultar'  
       PROCURE cod\_item\_pedido\_compra em Item Pedido Venda  
       EXIBA Pedido\_Venda  
     CASO Opcao = 'Alterar'  
       ACESSE Dados\_Itens\_pedido\_venda  
       REGRAVE Pedido\_Venda\_ok em Pedido de Venda  
     CASO Opcao = 'Excluir'  
       APAGUE Dados\_Itens\_pedido\_venda  
   FIM CASO  
 CASO CONTRARIO  
   POSICIONE no fim de Pedido\_Venda  
   GERE cod\_item\_pedido\_venda = cod\_item\_pedido\_venda + 1  
   ACESSE Pedido\_Venda



---

GRAVE compras\_ok em Pedido\_Venda  
FIM SE  
**Processo 1.2.3 – Emitir Relatório de Pedido de Venda**

DETERMINE relatorio = ['Todos', 'Cod']  
POCISONE inicio de Pedido\_de\_Venda  
ACESSE relatorio  
SE relatorio = 'Cod'  
    ENQUANTO nao atingir fim de Pedido\_de\_Venda  
        PROCURE cod\_cliente em Cliente  
        PROCURE cod\_cond\_pgto em Condicao de Pagamento  
        PROCURE cod\_funcionario em Funcionário  
        PROCURE cod\_item\_pedido\_compra em Item Pedido Venda  
            IMPRIMA cont\_relatorio\_pedido\_venda  
        POSICIONE proximo Pedido de Venda  
    FIM ENQUANTO  
FIM SE  
SE relatorio = 'Todos'  
    ENQUANTO nao atingir fim de Pedido de Venda  
        PROCURE cod\_fornecedor em Fornecedor  
        PROCURE cod\_cond\_pgto em Condicao de Pagamento  
        PROCURE cod\_funcionario em Funcionário  
        PROCURE cod\_item\_pedido\_venda em Item Pedido Venda  
            IMPRIMA cont\_relatorio\_pedido\_venda  
        POSICIONE proximo Pedido de Venda  
    FIM ENQUANTO  
SENAO  
    ACESSE cod\_pedido\_venda  
    PROCURE cod\_pedido\_venda em Pedido de Venda  
    SE ENCONTROU  
        IMPRIMA cont\_relatorio\_pedido\_venda  
    FIM SE  
FIM SE

## NÍVEL 2

### Processo 2.1 – Controlar NF Entrada

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_nf_entrada
PROCURE cod_nf_entrada em Nota Fiscal de Entrada
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_nf_entrada em Nota Fiscal de Entrada
    EXIBA Dados_nota_entrada
  CASO Opcao = 'Alterar'
    ACESSE Dados_nota_entrada
    REGRAVE cod_nf_entrada em Nota Fiscal de Entrada
  CASO Opcao = 'Excluir'
    APAGUE Nota Fiscal de Entrada
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Nota Fiscal de Entrada
  GERE cod_nf_entrada = cod_nf_entrada + 1
  ACESSE Dados_nota_entrada
  GRAVE dados_nota_entrada_ok em Nota Fiscal de Entrada
FIM SE
```

### Processo 2.2 - Controlar Contas à Pagar

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_contas_a_pagar
PROCURE cod_contas_a_pagar em Contas a Pagar
PROCURE
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_contas_a_pagar em Contas a Pagar
    EXIBA Dados_contas_pagar
  CASO Opcao = 'Alterar'
    ACESSE Dados_contas_pagar
    REGRAVE cod_contas_a_pagar em Contas a Pagar
  CASO Opcao = 'Excluir'
    APAGUE Contas a Pagar
  FIM CASO
CASO CONTRARIO
```

---

```
    POSICIONE no fim de Contas a Pagar
    GERE cod_contas_a_pagar = cod_contas_a_pagar + 1
    ACESSE Dados_contas_pagar
    GRAVE contas_a_pagar_ok em Contas a Pagar
FIM SE
```

### **Processo 2.3 - Controlar Contas à receber**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_contas_a_receber
PROCURE cod_contas_a_receber em Contas a Receber
PROCURE
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_contas_a_receber em Contas a Receber
            EXIBA Dados_contas_receber
        CASO Opcao = 'Alterar'
            ACESSE Dados_contas_receber
            REGRAVE cod_contas_a_receber em Contas a Receber
        CASO Opcao = 'Excluir'
            APAGUE Contas a receber
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Contas a receber
    GERE cod_contas_a_receber = cod_contas_a_receber + 1
    ACESSE Dados_contas_receber
    GRAVE contas_a_receber_ok em Contas a Receber
FIM SE
```

### **Processo 2.4 – Cadastrar ICMS**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_estado
PROCURE cod_estado em Estado
PROCURE
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_estado em Estado
            EXIBA Dados_icms
        CASO Opcao = 'Alterar'
            ACESSE Dados_icms
            REGRAVE cod_estado em Estado
        CASO Opcao = 'Excluir'
            APAGUE Estado
```

```
FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Estado
  GERE cod_estado = cod_estado + 1
  ACESSE Dados_icms
  GRAVE icms_ok em Estado
FIM SE
```

### **Processo 2.5 – Cadastrar Nat. Op**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_natop
PROCURE cod_natop em natureza_operacao
PROCURE
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_natop em natureza_operacao
    EXIBA Dados_nat_op
  CASO Opcao = 'Alterar'
    ACESSE Dados_nat_op
    REGRAVE cod_natop em natureza_operacao
  CASO Opcao = 'Excluir'
    APAGUE natureza_operacao
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de natureza_operacao
  GERE cod_natop = cod_natop + 1
  ACESSE Dados_nat_op
  GRAVE nat_op_ok em natureza_operacao
FIM SE
```

### **Processo 2.6 – Controlar Nota Saída**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_nf_saida
PROCURE cod_nf_saida em Nota Fiscal de Saida
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_nf_saida em Nota Fiscal de Saida
    EXIBA Dados_nota_saida
  CASO Opcao = 'Alterar'
    ACESSE Dados_nota_saida
    REGRAVE cod_nf_saida em Nota Fiscal de Saida
```

---

```
CASO Opcao = 'Excluir'
    APAGUE Nota Fiscal de Saida
FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Nota Fiscal de Saida
    GERE cod_nf_saida = cod_nf_saida + 1
    ACESSE Dados_nota_saida
    GRAVE dados_nota_saida_ok em Nota Fiscal de Saida
FIM SE
```

### **Processo 2.7 – Cadastrar Condição de Pagamento**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_cond_pgto
PROCURE cod_cond_pgto
PROCURE
SE ENCONTROU
    ACESSE Opcao
    CASO Opcao = 'Consultar'
        PROCURE cod_cond_pgto em Condição de Pagamento
        EXIBA Dados_cond_pgto
    CASO Opcao = 'Alterar'
        ACESSE Dados_cond_pgto
        REGRAVE cod_cond_pgto em Condição de Pagamento
    CASO Opcao = 'Excluir'
        APAGUE Condição de Pagamento
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Condição de Pagamento
    GERE cod_cond_pgto_top = cod_cond_pgto + 1
    ACESSE Dados_cond_pgto
    GRAVE cond_pgto_ok em Condição de Pagamento
FIM SE
```

### **Processo 2.1.1 – Cadastrar NF Entrada**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_nf_entrada
PROCURE cod_nf_entrada em Nota Fiscal de Entrada
SE ENCONTROU
    ACESSE Opcao
    CASO Opcao = 'Consultar'
        PROCURE cod_nf_entrada em Nota Fiscal de Entrada
        EXIBA Dados_nota_entrada
    CASO Opcao = 'Alterar'
```

---

```
        ACESSE Dados_nota_entrada
        REGRAVE cod_nf_entrada em Nota Fiscal de Entrada
    CASO Opcao = 'Excluir'
        APAGUE Nota Fiscal de Entrada
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Nota Fiscal de Entrada
    GERE cod_nf_entrada = cod_nf_entrada + 1
    ACESSE Dados_nota_entrada
    GRAVE dados_nota_entrada_ok em Nota Fiscal de Entrada
FIM SE
```

### **Processo 2.1.2 – Cadastrar Itens NF Entrada**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_itens_nf_compra
PROCURE cod_itens_nf_compra em Itens Nota Fiscal de Entrada
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_itens_nf_compra em Itens Nota Fiscal de Entrada
            EXIBA Dados_itens_nf_entrada
        CASO Opcao = 'Alterar'
            ACESSE Dados_itens_nf_entrada
            REGRAVE cod_itens_nf_compra em Itens Nota Fiscal de Entrada
        CASO Opcao = 'Excluir'
            APAGUE Itens Nota Fiscal de Entrada
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Itens Nota Fiscal de Entrada
    GERE cod_itens_nf_compra = cod_itens_nf_compra + 1
    ACESSE Dados_itens_nf_entrada
    GRAVE dados_itens_nota_entrada_ok em Itens Nota Fiscal de Entrada
FIM SE
```

### **Processo 2.1.3 – Emitir NF Entrada**

```
DETERMINE relatorio = ['Cod', 'Todos']
POCISONE inicio de Nota Fiscal de Entrada
ACESSE relatorio
SE relatorio = 'Cod'
    ENQUANTO nao atingir fim de Nota Fiscal de Entrada
        PROCURE cod_nf_compra em Nota Fiscal de Entrada
        IMPRIMA Dados_nota_entrada
        POSICIONE proximo Nota Fiscal de Entrada
```

```
FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
    ENQUANTO nao atingir fim de Nota Fiscal de Entrada
        PROCURE cod_nf_compra em Nota Fiscal de Entrada
            IMPRIMA Dados_nota_entrada
            POSICIONE proximo
        FIM ENQUANTO Nota Fiscal de Entrada
    SENAO
        ACESSE cod_nf_compra
        PROCURE cod_nf_compra em Nota Fiscal de Entrada
        SE ENCONTROU
            IMPRIMA Dados_nota_entrada
        FIM SE
    FIM SE
```

### **Processo 2.2.1 - Cadastrar Contas à Pagar**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_contas_a_pagar
PROCURE cod_contas_a_pagar em Contas a Pagar
PROCURE
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_contas_a_pagar em Contas a Pagar
            EXIBA Dados_contas_pagar
        CASO Opcao = 'Alterar'
            ACESSE Dados_contas_pagar
            REGRAVE cod_contas_a_pagar em Contas a Pagar
        CASO Opcao = 'Excluir'
            APAGUE Contas a Pagar
        FIM CASO
    CASO CONTRARIO
        POSICIONE no fim de Contas a Pagar
        GERE cod_contas_a_pagar = cod_contas_a_pagar + 1
        ACESSE Dados_contas_pagar
        GRAVE contas_a_pagar_ok em Contas a Pagar
    FIM SE
```

### **Processo 2.2.2 - Emitir Relatório Contas à Pagar**

```
DETERMINE relatorio = ['Cod', 'Todos']
POCISIONE inicio de Contas a Pagar
ACESSE relatorio
```

```
SE relatorio = 'Cod'
  ENQUANTO nao atingir fim de Contas a Pagar
    PROCURE cod_contas_a_receber em Contas a Pagar
    IMPRIMA Dados_contas_pagar
    POSICIONE proximo Contas a Pagar
  FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
  ENQUANTO nao atingir fim de Contas a Pagar
    PROCURE cod_contas_a_pagar em Contas a Pagar
    IMPRIMA cod_contas_a_pagar
    POSICIONE proximo Contas a Pagar
  FIM ENQUANTO
SENAO
  ACESSE cod_contas_a_pagar
  PROCURE cod_contas_a_pagar em Contas a Pagar
  SE ENCONTROU
    IMPRIMA Dados_contas_pagar
  FIM SE
FIM SE
```

### **Processo 2.2.3 - Efetuar Baixa**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_contas_a_pagar
PROCURE cod_contas_a_pagar em Contas a Pagar
PROCURE
SE ENCONTROU
  ACESSE Opcao
  PROCURE cod_contas_a_pagar em Contas a Pagar
  SE Opcao = 'SIM'
    EXIBA Dados_contas_pagar
    REGRAVE Dados_contas_pagar
  SENÃO
    EXIBA Dados_contas_pagar
  FIM SE
CASO CONTRARIO
  POSICIONE no fim de Contas a Pagar
  GERE cod_contas_a_pagar = cod_contas_a_pagar + 1
  ACESSE Dados_contas_pagar
  GRAVE contas_a_pagar_ok em Contas a Pagar
FIM SE
```



### Processo 2.3.1 - Cadastrar Contas à receber

```

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_contas_a_receber
PROCURE cod_contas_a_receber em Contas a Receber
PROCURE
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_contas_a_receber em Contas a Receber
    EXIBA Dados_contas_receber
  CASO Opcao = 'Alterar'
    ACESSE Dados_contas_receber
    REGRAVE cod_contas_a_receber em Contas a Receber
  CASO Opcao = 'Excluir'
    APAGUE Contas a receber
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Contas a receber
  GERE cod_contas_a_receber = cod_contas_a_receber + 1
  ACESSE Dados_contas_receber
  GRAVE contas_a_receber_ok em Contas a Receber
FIM SE

```

### Processo 2.3.2 - Emitir Relatório Contas a Receber

```

DETERMINE relatorio = ['Cod', 'Todos']
POCISIONE inicio de Contas a Receber
ACESSE relatorio
SE relatorio = 'Cod'
  ENQUANTO nao atingir fim de Contas a Receber
    PROCURE cod_contas_a_receber em Contas a Receber
    IMPRIMA Dados_contas_receber
    POSICIONE proximo Contas a Receber
  FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
  ENQUANTO nao atingir fim de Contas a Receber
    PROCURE cod_contas_a_receber em Contas a Receber
    IMPRIMA cod_contas_a_receber
    POSICIONE proximo Contas a Receber
  FIM ENQUANTO
SENAO
  ACESSE cod_contas_a_receber
  PROCURE cod_contas_a_receber em Contas a Receber
  SE ENCONTROU

```

---

```
        IMPRIMA Dados_contas_receber
    FIM SE
FIM SE
```

### **Processo 2.3.3 - Efetuar Baixa Ct. Receber**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_contas_a_receber
PROCURE cod_contas_a_receber em Contas a Receber
PROCURE
SE ENCONTROU
    ACESSE Opcao
        PROCURE cod_contas_a_receber em Contas a Receber
            SE Opcao = 'SIM'
                EXIBA Dados_contas_receber
                REGRAVE Dados_contas_receber
            SENÃO
                EXIBA Dados_contas_receber
        FIM SE
CASO CONTRARIO
    POSICIONE no fim de Contas a Receber
    GERE cod_contas_a_receber = cod_contas_a_receber + 1
    ACESSE Dados_contas_receber
    GRAVE contas_a_receber_ok em Contas a Receber
FIM SE
```

### **Processo 2.6.1 – Cadastrar NF Saída**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_nf_saida
PROCURE cod_nf_saida em Nota Fiscal de Saida
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_nf_saida em Nota Fiscal de Saida
            EXIBA Dados_nota_saida
        CASO Opcao = 'Alterar'
            ACESSE Dados_nota_saida
            REGRAVE cod_nf_saida em Nota Fiscal de Saida
        CASO Opcao = 'Excluir'
            APAGUE Nota Fiscal de Saida
        FIM CASO
CASO CONTRARIO
```

---

```
    POSICIONE no fim de Nota Fiscal de Saida
    GERE cod_nf_saida = cod_nf_saida + 1
    ACESSE Dados_not_saida
    GRAVE dados_not_saida_ok em Nota Fiscal de Saida
FIM SE
```

### **Processo 2.6.2 – Cadastrar Itens NF Saída**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_itens_nf_saida
PROCURE cod_itens_nf_saida em Itens Nota Fiscal de Saida
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_itens_nf_saida em Itens Nota Fiscal de Saida
            EXIBA Dados_itens_nf_saida
        CASO Opcao = 'Alterar'
            ACESSE Dados_itens_nf_saida
            REGRAVE cod_nf_saida em Itens Nota Fiscal de Saida
        CASO Opcao = 'Excluir'
            APAGUE Itens Nota Fiscal de Saida
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Itens Nota Fiscal de Saida
    GERE cod_itens_nf_saida = cod_itens_nf_saida + 1
    ACESSE Dados_itens_nf_saida
    GRAVE dados_itens_nf_saida_ok em Itens Nota Fiscal de Saida
FIM SE
```

### **Processo 2.6.3 – Gerar NF Saída**

```
DETERMINE relatorio = ['Cod', 'Todos']
POCISONE inicio de Nota Fiscal de Saida
ACESSE relatorio
SE relatorio = 'Cod'
    ENQUANTO nao atingir fim de Nota Fiscal de Saida
        PROCURE cod_nf_venda em Nota Fiscal de Saida
        IMPRIMA Dados_not_saida
        POSICIONE proximo Nota Fiscal de Saida
    FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
    ENQUANTO nao atingir fim de Nota Fiscal de Saida
        PROCURE cod_nf_venda em Nota Fiscal de Saida
        IMPRIMA Dados_not_saida
```

```
        POSICIONE proximo
    FIM ENQUANTO Nota Fiscal de Saida
SENAO
    ACESSE cod_nf_venda
    PROCURE cod_nf_venda em Nota Fiscal de Saida
    SE ENCONTROU
        IMPRIMA Dados_nota_saida
    FIM SE
FIM SE
```

### NÍVEL 3

#### Processo 3.1 – Gerenciar Estoque

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_lote
PROCURE cod_lote em Lote
PROCURE cod_produto
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_lote em Lote
            EXIBA Dados_lote
        CASO Opcao = 'Alterar'
            ACESSE Dados_lote
            REGRAVE cod_lote em Lote
        CASO Opcao = 'Excluir'
            APAGUE Lote
        FIM CASO
    CASO CONTRARIO
        POSICIONE no fim de Lote
        GERE cod_lote = cod_lote + 1
        ACESSE Dados_lote
        GRAVE dados_lote_ok em Lote
    FIM SE
```

#### Processo 3.1.1 – Gerenciar Produtos

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_produto
PROCURE cod_produto em Produto
PROCURE
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
```

```
        PROCURE cod_produto em Produto
        EXIBA Dados_produto
    CASO Opcao = 'Alterar'
        ACESSE Dados_produto
        REGRAVE cod_produto em Produto
    CASO Opcao = 'Excluir'
        APAGUE Produto
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Produto
    GERE cod_produto = cod_produto + 1
    ACESSE Dados_produto
    GRAVE cod_produto_ok em Produto
FIM SE
```

### **Processo 3.1.1.1 – Cadastrar Produtos**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_produto
PROCURE cod_produto em Produto
PROCURE
SE ENCONTROU
    ACESSE Opcao
    CASO Opcao = 'Consultar'
        PROCURE cod_produto em Produto
        EXIBA Dados_produto
    CASO Opcao = 'Alterar'
        ACESSE Dados_produto
        REGRAVE cod_produto em Produto
    CASO Opcao = 'Excluir'
        APAGUE Produto
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Produto
    GERE cod_produto = cod_produto + 1
    ACESSE Dados_produto
    GRAVE cod_produto_ok em Produto
FIM SE
```

### **Processo 3.1.1.2 – Emitir Relatório Produto**

```
DETERMINE relatorio = ['Cod', 'Todos']
POCISIONE inicio de Produto
ACESSE relatorio
```

```

SE relatorio = 'Cod'
  ENQUANTO nao atingir fim de Produto
    PROCURE cod_produto em Produto
    IMPRIMA Dados_produto
    POSICIONE proximo Produto
  FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
  ENQUANTO nao atingir fim de Produto
    PROCURE cod_produto em Produto
    IMPRIMA Dados_produto
    POSICIONE proximo
  FIM ENQUANTO Produto
SENAO
  ACESSE cod_produto
  PROCURE cod_produto em Produto
  SE ENCONTROU
    IMPRIMA Dados_ Produto
  FIM SE
FIM SE

```

### **Processo 3.1.1.3 – Emitir Relatório Classificação ABC**

```

DETERMINE relatorio = ['Cod', 'Todos']
POCISIONE inicio de Nota Fiscal de Saída
ACESSE relatorio
SE relatorio = 'Cod'
  ENQUANTO nao atingir fim de Itens Nota Fiscal de Saida
    PROCURE cod_produto, cod_nota_fiscal_saida, cod_itens_nf_saida em Nota
Fiscal de Saída, Itens Nota Fiscal de Saida
    SE Dados_Itens_nf_saida = 20% E 65 %
      IMPRIMA relatorio_classificacao_abc = A
    SENAO Dados_Itens_nf_saida = 30% E 25%
      IMPRIMA relatorio_classificacao_abc = B
    SENAO Dados_Itens_nf_saida = 50% E 10%
      IMPRIMA relatorio_classificacao_abc = C
    FIM SE
  POSICIONE proximo Itens Nota Fiscal Saida
  FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
  ENQUANTO nao atingir fim de Itens Nota Fiscal Saida
    PROCURE cod_produto, cod_nota_fiscal_saida, cod_itens_nf_saida em Nota
Fiscal de Saída, Itens Nota Fiscal de Saida
    SE Dados_Itens_nf_saida = 20% E 65 %
      IMPRIMA relatorio_classificacao_abc = A

```

```

        SENAO Dados_Itens_nf_saida = 30% E 25 %
            IMPRIMA relatorio_classificacao_abc = B
        SENAO Dados_Itens_nf_saida = 50% E 10 %
            IMPRIMA relatorio_classificacao_abc = C
        FIM SE
        POSICIONE proximo Itens Nota Fiscal Saida
    FIM ENQUANTO Itens Nota Fiscal Saida
SENAO
    ACESSE cod_itens_nf_saida
    PROCURE cod_produto, cod_nota_fiscal_saida, cod_itens_nf_saida em Nota
Fiscal de Saída, Itens Nota Fiscal de Saida
    SE ENCONTROU
        IMPRIMA relatório_classificacao_abc
    FIM SE
FIM SE

```

#### **Processo 3.1.1.4 – Emitir Relatório Sugestão de Reposição**

```

DETERMINE relatorio = ['Cod', 'Todos']
POCISIONE inicio de Produto
ACESSE relatorio
SE relatorio = 'Cod'
    ENQUANTO nao atingir fim de Produto
        PROCURE cod_produto em Produto
        IMPRIMA Dados_produto
        POSICIONE proximo Produto
    FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
    ENQUANTO nao atingir fim de Produto
        PROCURE cod_produto em Produto
        IMPRIMA Dados_produto
        POSICIONE proximo
    FIM ENQUANTO Produto
SENAO
    ACESSE cod_produto
    PROCURE cod_produto em Produto
    SE ENCONTROU
        IMPRIMA Dados_ Produto
    FIM SE
FIM SE
FIM SE

```

**Processo 3.1.2 – Cadastrar Lote**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_lote
PROCURE cod_lote em Lote
PROCURE cod_produto em produto
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_lote em Lote
            EXIBA Dados_lote
        CASO Opcao = 'Alterar'
            ACESSE Dados_lote
            REGRAVE cod_lote em Lote
        CASO Opcao = 'Excluir'
            APAGUE Lote
        FIM CASO
    CASO CONTRARIO
        POSICIONE no fim de Lote
        GERE cod_lote = cod_lote + 1
        ACESSE Dados_lote
        GRAVE dados_lote_ok em Lote
    FIM SE
```

**Processo 3.1.3 – Emitir Relatório Lote**

```
DETERMINE relatorio = ['Cod', 'Todos']
POCISIONE inicio de Lote
ACESSE relatorio
SE relatorio = 'Cod'
    ENQUANTO nao atingir fim de Lote
        PROCURE cod_lote em Lote
        IMPRIMA Dados_lote
        POSICIONE proximo Lote
    FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
    ENQUANTO nao atingir fim de Lote
        PROCURE cod_acesso em Lote
        IMPRIMA Dados_lote
        POSICIONE proximo
    FIM ENQUANTO Lote
SENAO
    ACESSE cod_lote
    PROCURE cod_lote em Lote
    SE ENCONTROU
        IMPRIMA Dados_lote
```



FIM SE  
FIM SE

## NÍVEL 4

### Processo 4.1 – Gerenciar Usuário e Grupo

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_func
PROCURE cod_func em Funcionarios
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_cidade em Cidade
    PROCURE cod_area em Area
    EXIBA Dados_funcionario
  CASO Opcao = 'Alterar'
    ACESSE Dados_funcionario
    REGRAVE funcionario_ok
  CASO Opcao = 'Excluir'
    APAGUE Funcionarios
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Funcionarios
  GERE cod_func = cod_func + 1
  ACESSE Dados_funcionario
  GRAVE funcionario_ok em Funcionarios
FIM SE
```

#### Processo 4.1.1 - Cadastrar Usuário

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_usuario
PROCURE cod_usuario em Usuario
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_usuario em Usuario
    EXIBA Dados_usuario
  CASO Opcao = 'Alterar'
    ACESSE Dados_usuario
    REGRAVE usuario_ok em Usuario
  CASO Opcao = 'Excluir'
    APAGUE Usuario
  FIM CASO
```

```

CASO CONTRARIO
    POSICIONE no fim de Usuario
    GERE cod_usuario = cod_usuario + 1
    ACESSE Dados_usuario
    GRAVE usuario_ok em Usuario
FIM SE

```

#### **Processo 4.1.2 – Cadastrar Permissão do Usuário**

```

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_permissao
PROCURE cod_usuario A, cod_grupo B, cod_aplicacao C em A Usuario, B Grupo, C
Aplicacao
SE ENCONTROU
    ACESSE Opcao
    CASO Opcao = 'Consultar'
        PROCURE cod_usuario A, cod_grupo B, cod_aplicacao C em A Usuário,
B Grupo, C Aplicacao
        EXIBA Dados_permissao_usuario
    CASO Opcao = 'Alterar'
        ACESSE Dados_permissao_usuario
        REGRAVE usuario_ok, grupo_ok, aplicação_ok em Permissao_Usuario
    CASO Opcao = 'Excluir'
        APAGUE Permissao_Usuario
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Permissao_Usuario
    GERE cod_permissao = cod_permissao + 1
    ACESSE Dados_permissao_usuario
    GRAVE usuario_ok, grupo_ok, aplicação_ok em Permissao_Usuario
FIM SE

```

#### **Processo 4.1.3 – Cadastrar Permissão do Grupo**

```

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_permissao
PROCURE cod_usuario A, cod_grupo B, cod_aplicacao C em A Usuario, B Grupo, C
Aplicacao
SE ENCONTROU
    ACESSE Opcao
    CASO Opcao = 'Consultar'
        PROCURE cod_usuario A, cod_grupo B, cod_aplicacao C em A
        usuario, B Grupo, C Aplicacao
        EXIBA Dados_permissao_grupo
    CASO Opcao = 'Alterar'

```

```
        ACESSE Dados_ permissao_grupo
        REGRAVE usuario_ok em Permissao_Grupo
    CASO Opcao = 'Excluir'
        APAGUE Usuario
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de permissao_grupo
    GERE cod_permissao = cod_permissao + 1
    ACESSE Dados_ permissao_grupo
    GRAVE grupo_ok em Permissao_Grupo
FIM SE
```

#### **Processo 4.1.4 - Cadastrar Grupo**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_grupo em Grupo
PROCURE cod_grupo em Grupo
SE ENCONTROU
    ACESSE Opcao
        CASO Opcao = 'Consultar'
            PROCURE cod_grupo em Grupo
            EXIBA Dados_grupo
        CASO Opcao = 'Alterar'
            ACESSE Dados_grupo
            REGRAVE grupo_ok em Grupo
        CASO Opcao = 'Excluir'
            APAGUE Grupo
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de grupo
    GERE cod_grupo = cod_grupo + 1
    ACESSE Dados_grupo
    GRAVE grupo_ok em Grupo
FIM SE
```

#### **Processo 4.1.5 – Emitir Relatório Log Acesso**

```
DETERMINE relatorio = ['Cod', 'Todos']
POCISONE inicio de Acesso
ACESSE relatorio
SE relatorio = 'Cod'
    ENQUANTO nao atingir fim de Acesso
        PROCURE cod_acesso em Acesso
        IMPRIMA Cont_acesso
    POSICIONE proximo Acesso
```

```
FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
    ENQUANTO nao atingir fim de Acesso
        PROCURE cod_acesso em Acesso
        IMPRIMA Cont_acesso
        POSICIONE proximo
    FIM ENQUANTO Acesso
SENAO
    ACESSE cod_acesso
    PROCURE cod_acesso em Acesso
    SE ENCONTROU
        IMPRIMA Dados_acesso
    FIM SE
FIM SE
```

#### **Processo 4.1.6 – Cadastrar Log Acesso**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_acesso
PROCURE cod_acesso em Acesso
SE ENCONTROU
    ACESSE Opcao
    CASO Opcao = 'Consultar'
        PROCURE cod_acesso em Acesso
        EXIBA Dados_acesso
    CASO Opcao = 'Alterar'
        ACESSE Dados_acesso
        REGRAVE cont_acesso em Acesso
    CASO Opcao = 'Excluir'
        APAGUE Acesso
    FIM CASO
CASO CONTRARIO
    POSICIONE no fim de Acesso
    GERE cod_acesso = cod_acesso + 1
    ACESSE Dados_acesso
    GRAVE cont_acesso em Acesso
FIM SE
```

#### **Processo 4.1.7 – Cadastrar Aplicação**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_aplicacao
PROCURE cod_aplicacao em Aplicacao
SE ENCONTROU
```

```
ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_aplicacao em Aplicacao
    EXIBA Dados_aplicacao
  CASO Opcao = 'Alterar'
    ACESSE Dados_aplicacao
    REGRAVE cont_acesso em Aplicacao
  CASO Opcao = 'Excluir'
    APAGUE Aplicacao
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Aplicacao
  GERE cod_aplicacao = cod_aplicacao + 1
  ACESSE Dados_aplicacao
  GRAVE aplicacao_ok em Aplicacao
FIM SE
```

#### **Processo 4.2 - Cadastrar Funcionários**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_func
PROCURE cod_func em Funcionarios
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_cidade em Cidade
    PROCURE cod_area em Area
    EXIBA Dados_funcionario
  CASO Opcao = 'Alterar'
    ACESSE Dados_funcionario
    REGRAVE funcionario_ok
  CASO Opcao = 'Excluir'
    APAGUE Funcionarios
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Funcionarios
  GERE cod_func = cod_func + 1
  ACESSE Dados_funcionario
  GRAVE funcionario_ok em Funcionarios
FIM SE
```

### Processo 4.3 - Cadastrar Cargo

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_cargo
PROCURE cod_cargo em Cargo
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_cargo em Cidade
    EXIBA Dados_cidade
  CASO Opcao = 'Alterar'
    ACESSE Dados_cargo
    REGRAVE cargo_ok em Cidade
  CASO Opcao = 'Excluir'
    APAGUE Cargo
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Cargo
  GERE cod_cargo = cod_cargo + 1
  ACESSE Dados_cargo
  GRAVE cargo_ok em Cargo
FIM SE
```

### Processo 4.4 - Cadastrar Cidade

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_cidade
PROCURE cod_cidade em Cidade
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_cidade em Cidade
    EXIBA Dados_cidade
  CASO Opcao = 'Alterar'
    ACESSE Dados_cidade
    REGRAVE cidade_ok em Cidade
  CASO Opcao = 'Excluir'
    APAGUE Cidade
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Cidade
  GERE cod_cidade = cod_cidade + 1
  ACESSE Dados_cidade
  GRAVE cidade_ok em Cidade
FIM SE
```

**Processo 4.5 - Cadastrar Estado**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_estado
PROCURE cod_estado em Estado
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_estado em Estado
    EXIBA Dados_estado
  CASO Opcao = 'Alterar'
    ACESSE Dados_estado
    REGRAVE estado_ok em Estado
  CASO Opcao = 'Excluir'
    APAGUE Estado
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Estado
  GERE cod_estado = cod_estado + 1
  ACESSE Dados_estado
  GRAVE estado_ok em Estado
FIM SE
```

**Processo 4.6 - Cadastrar Fornecedor**

```
DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_fornecedores
PROCURE cod_fornecedores em Fornecedores
SE ENCONTROU
  ACESSE Opcao
  CASO Opcao = 'Consultar'
    PROCURE cod_fornecedores em Fornecedores
    EXIBA Dados_fornecedores
  CASO Opcao = 'Alterar'
    ACESSE Dados_fornecedores
    REGRAVE fornecedores_ok em Fornecedores
  CASO Opcao = 'Excluir'
    APAGUE Fornecedores
  FIM CASO
CASO CONTRARIO
  POSICIONE no fim de Fornecedores
  GERE cod_fornecedores = cod_fornecedores + 1
  ACESSE Dados_fornecedores
  GRAVE fornecedores_ok em Fornecedores
FIM SE
```



### Processo 4.7 - Cadastrar Cliente

```

DETERMINE Opcao = ['Incluir', 'Excluir', 'Alterar', 'Consultar']
ACESSE cod_cliente
PROCURE cod_cliente em Clientes
SE ENCONTROU
  ACESSE Opcao
    CASO Opcao = 'Consultar'
      PROCURE cod_cliente em Clientes
      EXIBA Dados_clientes
    CASO Opcao = 'Alterar'
      ACESSE Dados_clientes
      REGRAVE clientes_ok em Clientes

    CASO Opcao = 'Excluir'
      APAGUE Clientes
    FIM CASO
  CASO CONTRARIO
    POSICIONE no fim de Clientes
    GERE cod_cliente = cod_cliente + 1
    ACESSE Dados_clientes
    GRAVE clientes_ok em Clientes
  FIM SE

```

### Processo 4.8 – Emitir Relatório Cliente

```

DETERMINE relatorio = ['Cod', 'Todos']
POCISIONE inicio de Cliente
ACESSE relatorio
SE relatorio = 'Cod'
  ENQUANTO nao atingir fim de Cliente
    PROCURE cod_cliente em Cliente
    PROCURE cod_cidade em Cidade
    IMPRIMA Dados_cliente
    POSICIONE proximo cliente
  FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
  ENQUANTO nao atingir fim de Cliente
    PROCURE cod_cliente em Cliente
    PROCURE cod_cidade em Cidade
    IMPRIMA Dados_cliente
    POSICIONE proximo cliente
  FIM ENQUANTO
SENAO
  ACESSE cod_cliente

```

```
    PROCURE cod_cliente em Cliente
    SE ENCONTROU
        IMPRIMA Dados_cliente
    FIM SE
FIM SE
```

#### **Processo 4.9 – Emitir Relatório Fornecedor**

```
DETERMINE relatorio = ['Cod', 'Todos']
POCISIONE inicio de Fornecedor
ACESSE relatorio
SE relatorio = 'Cod'
    ENQUANTO nao atingir fim de Fornecedor
        PROCURE cod_fornecedor em Fornecedor
        PROCURE cod_fornecedor em Fornecedor
        IMPRIMA Dados_fornecedor
        POSICIONE proximo Fornecedor
    FIM ENQUANTO
FIM SE
SE relatorio = 'Todos'
    ENQUANTO nao atingir fim de Fornecedor
        PROCURE cod_fornecedor em Fornecedor
        PROCURE cod_fornecedor em Fornecedor
        IMPRIMA Dados_fornecedor
        POSICIONE proximo Fornecedor
    FIM ENQUANTO
SENAO
    ACESSE cod_fornecedor
    PROCURE cod_fornecedor em Fornecedor
    SE ENCONTROU
        IMPRIMA Dados_fornecedor
    FIM SE
FIM SE
```