



Universidade Federal da Paraíba – **UFPB**
Centro de Ciência e Tecnologia – **CCT**
Departamento de Sistemas e Computação – **DSC**

Disciplina :

➤ Estágio Integrado

Tutor Acadêmico:

➤ Joseluze Cunha

Aluno:

➤ Lysianne do Couto Andrade

Relatório de Estágio

24/09/2001



Biblioteca Setorial do CDSA. Maio de 2021.

Sumé - PB

Agradecimentos

Os agradecimentos são direcionados à professora titular do DSC/UFPB, Joselúce Cunha, pela orientação dada no decorrer da disciplina Estágio Integrado. Além da sua disponibilidade em esclarecer dúvidas referentes à área de Banco de Dados e em revisar constantemente o relatório final do estágio.

Os agradecimentos também são extensivos aos demais estagiários desse projeto que, comigo formaram, a equipe de trabalho.

Índice

Índice.....	2
Apresentação.....	3
Introdução.....	4
1. Ambiente de Estágio.....	5
1.1. Aspectos positivos.....	5
1.2. Aspectos negativos.....	5
2. Fundamentação Teórica.....	6
2.1. Metodologia de Análise.....	6
2.2. Ferramenta de Desenvolvimento.....	6
2.3. Ferramenta de Armazenamento.....	7
3. Descrição do Problema.....	8
4. Proposta de Solução.....	10
4.1. O Sistema de Automação de Postos de Gasolina (SAP).....	12
4.2. Análise do Sistema e Projeto do Banco de Dados.....	26
4.2.1. Análise do Sistema.....	26
4.2.2. Projeto do Banco de Dados.....	26
4.2.2.1. Migração dos Dados.....	26
4.2.2.2. Garantia de Performance: Uso de Procedures.....	32
5. Atividades Desenvolvidas.....	36
5.1. Estudo das Tecnologias.....	36
5.2. Definição dos Requisitos.....	36
5.3. Criação do Modelo Conceitual.....	36
5.4. Implementação e Testes.....	36
5.5. Edição do Relatório de Estágio.....	37
6. Conclusão.....	38
7. Referência Bibliográfica.....	39
8. Anexos.....	40
Anexo 1.....	40
Plano de Estágio.....	40
Anexo 2.....	44
Metodologia de Análise.....	44
Análise Essencial.....	44
Anexo 3.....	46
Ferramenta de Desenvolvimento.....	46
Borland Delphi 5.....	46
Anexo 4.....	51
Ferramenta de Armazenamento.....	51
Interbase 6.0.....	51
Anexo 5.....	52
Modelagem do Sistema SAP usando Análise Essencial.....	52
Requisitos do Sistema.....	52
Diagrama de Entidade/Relacionamento (DER).....	54
Dicionário de Dados do DER.....	55
Diagramas de Fluxo de Dados (DFD) e Mini-Especificações.....	65

Apresentação

O presente documento tem por objetivo relatar os principais detalhes das atividades realizadas no decorrer da disciplina de Estágio Integrado pelo aluna de Ciência da Computação Lysianne do Couto Andrade, matrícula 29711141, da Universidade Federal da Paraíba – Campus II.

Primeiramente a importância da disciplina em questão deve ser citada pelo fato de integrar o aluno no campo de atuação. Uma vez que toda teoria assimilada durante o curso começa a ser posta em prática numa situação real de mercado. Possibilitando também o enriquecimento em termos de experiência para sua formação. Além de que ajuda o graduando a se familiarizar com necessidades reais de clientes, bem como um ambiente de trabalho em equipe, uma vez que o estágio em questão contou com uma equipe de três estagiários para desenvolvimento do Sistema que será apresentado.

Os conhecimentos empregados durante o estágio tiveram como base as disciplinas de programação, engenharia de software, análise de sistemas e banco de dados. Além da iniciativa individual da estagiário em aprender a noções básicas da linguagem Delphi 5 e a manipulação de dados no SGBD(Sistema de Gerenciamento de Banco de Dados) Interbase 6.

É necessário salientar a participação dos dois estagiários que fizeram parte do desenvolvimento do Sistema objeto de estudo do estágio. Como já citado, o processo de desenvolvimento foi acompanhado por três estagiários. *Leidjane Souto*, que deu início ao seu estágio quando já havia sido iniciado o desenvolvimento do projeto. Esta teve como sua principal função a implementação da impressão dos relatórios necessários para o funcionamento do Sistema. *Vlademir Batista*, que teve como principal função o desenvolvimento da aplicação. *Lysianne do Couto Andrade*, presente redatora deste relatório, cuja principal função foi a análise do Sistema e implementação do banco de dados. Vale salientar que qualquer modificação na análise era discutida em conjunto entre os três estagiários. Isso era possível, pois todos tinham uma visão geral do sistema.

As funcionalidades particulares da estagiária em questão serão descritas em detalhes no decorrer do relatório.

Introdução

O estágio aconteceu na empresa Microsistemas Informática, atuante no ramo de desenvolvimento de softwares. A Empresa dispõe de vários clientes que utilizam softwares de controle produzidos pela mesma há algum tempo. Este desenvolvimento havia sido baseado na linguagem de programação Clipper utilizando arquivos de extensão DBF como tabelas de base de dados. Isso acarretou em uma baixa performance quando foi preciso utilizar um servidor de base de dados a uma distância considerável entre duas cidades.

O sistema desenvolvido no estágio fornece um controle geral para uma rede de postos de gasolina, Postos BR Cruzeiro (um dos clientes da Empresa Microsistemas). A base de dados, localizada em um servidor na cidade de Pesqueira PE, precisa ser acessada pela aplicação instalada na cidade de Arco Verde PE, através de um link da empresa de Telecomunicações Embratel. Com a queda de performance devido o acesso via Embratel, a Microsistemas resolveu melhorá-la, fazendo sua antiga aplicação em uma nova versão utilizando o Delphi 5 e um SGBD Interbase 6.

O Sistema anterior não possui uma documentação formal de requisitos, modelo de entidade-relacionamento, bem como outras documentações. É sabido que é imprescindível para o desenvolvimento de um sistema de informação que existam todos estes artefatos de desenvolvimento. Então, além do sistema em si, a proposta de estágio também reside em uma documentação formal e completa de toda a sua análise.

Uma descrição mais precisa do problema será feita no decorrer do relatório, bem como a solução empregada e os resultados finais do estágio. Inicialmente será relatado o ambiente físico onde a estagiária trabalhou, as tecnologias utilizadas, dificuldades encontradas e, como citado acima, uma descrição das atividades realizadas pelo estagiário no desenvolvimento do sistema em questão, que foi a base de estudo da disciplina de Estágio Integrado.

1. Ambiente de Estágio

O estágio ocorreu na Empresa *Microsistemas Informática LTDA*, localizada na Rua Vereador Benedito Motta, 875, Alto Branco– CEP.: 58.102-520. Fonefax (0**83) 342 4452 / 9972 0727 – Campina Grande – PB. E-mail: microsis@cgnet.com.br.

A empresa disponibiliza três máquinas com as seguintes configurações:

- 1) Pentium II 300, 128 MB de RAM, 20 GB de HD
- 2) Pentium III 800, 128 MB de RAM, 20 GB de HD
- 3) Athlon 900, 128 MB de RAM, 30 GB de HD

O público alvo da empresa é a indústria e comércio, tendo como principais clientes:

- 1) Café Aurora
- 2) Tubos Nogueira
- 3) Pneumax LTDA
- 4) Postos BR Cruzeiro
- 5) CIPAN LTDA (Comércio e Indústria de Produtos Alimentícios do Nordeste)
- 6) SAMIC (Serviço de Assistência Médica em Campina Grande)
- 7) Indústria Alimentícia 3 de Maio S.A.

A Empresa possui um analista de sistemas e programador (dono da empresa), um programador, um técnico em hardware e quatro estagiários.

1.1. Aspectos positivos

- O ambiente físico é bastante agradável, amplo e iluminado;
- Os companheiros de trabalho são bastante dinâmicos e prestativos o que facilitou a troca de informações necessárias;
- São disponibilizados materiais didáticos, como livros e apostilas para consulta e máquinas de porte suficiente para o trabalho em questão, todas conectadas em rede e com acesso à Internet;
- Há liberdade para um horário flexível de acordo com as disciplinas ainda em curso pelo estagiário.

1.2. Aspectos negativos

- A empresa é localizada em um bairro distante da universidade e de onde mora a estagiária;
- O cliente mora em outro estado, dificultando o acesso ao mesmo.

2. Fundamentação Teórica

2.1. Metodologia de Análise

Como Metodologia de Análise foi escolhida a Análise Essencial. Como o Sistema em questão deveria ser desenvolvido em uma abordagem Estruturada, era necessária uma análise também estruturada. Logo, como a estagiária tinha uma maior familiaridade com a Análise Essencial, bem como os demais componentes da equipe de desenvolvimento, composta de três estagiários, a escolha se mostrou a mais sensata para o caso presente.

Além de que, a Análise Essencial é uma ferramenta bastante detalhada, no que diz respeito aos seus artefatos gerados. Seus diagramas são bastante simples de se projetar e entender e envolvem tanto o banco de dados (DER), quanto a aplicação (DFD) de forma completa. Para maiores detalhes a respeito da metodologia de análise abordada ver Anexo 2.

2.2. Ferramenta de Desenvolvimento

Desde que a primeira versão do Delphi foi lançada, em 1995, esta ferramenta tem se mostrado como a melhor escolha no desenvolvimento para Windows. Numa relação com outros ambientes de programação, podemos dizer que o Delphi tem o poder do C++, e a facilidade do Visual Basic.

A principal vantagem do Delphi está na linguagem usada, Object Pascal, que é uma evolução do Pascal padrão. O Pascal surgiu no final dos anos 60 e, até hoje, é usada como uma das primeiras linguagens de programação para estudantes de computação. Em 1984, a Borland lançou o Turbo Pascal, que se firmou como o melhor compilador de Pascal do mercado e, a partir de então, passou a incluir novos recursos nesta linguagem, como Units e Objetos, até a ascensão do Windows, quando foi lançado o Turbo Pascal for Windows e, depois, o Borland Pascal, cuja linguagem é considerada a primeira versão da Object Pascal. Na sua atual versão, usada pelo Delphi, a Object Pascal é uma linguagem que permite a implementação dos mais variados tipos de aplicação (comercial ou acadêmico), é uma linguagem que já está há muito tempo no mercado sem perder sua peculiar facilidade de entendimento e implementação.

No Delphi, a criação de aplicativos começa com a montagem de componentes em janelas, como se fosse um programa gráfico, o usuário também pode utilizar componentes desenvolvidos por terceiros ou criar seus próprios componentes.

O Delphi vem com todas as ferramentas necessárias para a criação de bancos de dados dBase e Paradox, além de uma versão do Interbase, permitindo a criação de aplicativos com banco de dados sem a necessidade de aquisição de outro programa. O Delphi também tem

acesso a bases de dados como Foxpro, Access, InFormix, SYBASE, Oracle, SQL Server e DB2, além de qualquer outro banco de dados para Windows compatível com ODBC.

Com todo esse poder de desenvolvimento oferecido pela ferramenta, não haveria dificuldades para projetar uma interface com um bom *design* e agradável aos olhos do cliente. Para maiores detalhes sobre o Delphi 5 da Borland ver Anexo 3.

2.3. Ferramenta de Armazenamento

Como ferramenta de armazenamento foi escolhido o Interbase em sua versão 6.0. Uma vez que seria feita uma atualização no Sistema da rede de postos, por motivo principal de baixa performance, era preciso um SGBD para manipulação dos dados. O Interbase adequava-se principalmente por ser um software que garante integridade e consistência dos dados. Dá suporte *Triggers*¹, *Stored Procedures*², *Generatos*³ de auto-incremento, dentre outras vantagens que um bom SGBD deve possuir. Além de tudo ele é *freeware* (grátis), assim o cliente não teria um grande impacto financeiro com as atualizações feitas no seu sistema de controle. Para maiores detalhes sobre o Interbase ver Anexo 4.

¹ Trigger – Ação do banco de dados que é tomada em determinadas situações previamente definidas.

² Stored Procedures - Procedimentos armazenados no banco de dados que podem ser acessados da aplicação, executando na própria máquina onde está o banco de dados.

³ Generator – Gera números em seqüência quando acionados por um Trigger.

3. Descrição do Problema

A Empresa Microsistemas Informática está no ramo de desenvolvimento de Software há 10 anos. Nesse tempo, foram desenvolvidos vários sistemas de gerenciamento para seus clientes, dentre os quais está a rede de postos de gasolina, Posto BR Cruzeiro.

Para esse caso, foi desenvolvido um sistema de controle de estoque, que dá suporte às compras e vendas feitas por toda a rede de postos. Tal sistema controla o que é vendido, no Hotel, Bar, Lanchonete, Restaurante, bem como o que é vendido na pista, pelos frentistas. Dá suporte também às compras feitas aos seus fornecedores.

Como se trata de mais de um posto para gerenciar, o sistema cobre todas as suas filiais, inclusive as que se encontram em cidades distintas, mantendo apenas uma base de dados única e consistente, a qual se encontra na cidade de Pesqueira em Pernambuco.

Para comunicação entre o sistema e a base de dados, existe um link alugado à empresa de telecomunicações Embratel, por onde trafegam as informações sempre que requisitadas pelo sistema. Esse link fica entre a cidade de Pesqueira e Arco Verde também no estado de Pernambuco, onde a primeira possui um servidor de banco de dados que utiliza o sistema operacional Linux. As informações da base de dados desse servidor são constantemente requisitadas pela aplicação instalada no Posto BR Cruzeiro IV, localizado em Arco Verde.

Todo esse trabalho foi desenvolvido na linguagem de programação Clipper com uma base de dados xbase, na qual todas as tabelas são arquivos individuais de extensão dbf.

Além do sistema não possuir uma interface moderna, atual e prática, pelo fato de ser uma interface DOS, o sistema não possui um SGBD para gerenciamento da base de dados que utiliza. Outro problema, que talvez seja um dos mais importantes a destacar, é o fato de não haver nenhuma documentação formal inerente ao processo de desenvolvimento do sistema, como um documento de requisitos, diagramas relacionados à base de dados ou à aplicação (DER, DFD, etc).

A principal preocupação da Empresa é a seguinte:

Com a integração do link Embratel no acesso ao banco de dados, a aplicação perdeu muito de sua performance. Isso acarretou uma lentidão considerável na manipulação dos dados entre as cidades supracitadas. Essa lentidão se deu pelo fato de que em arquivos xbase, não é utilizada a linguagem SQL, logo os dados que trafegam do servidor para o cliente e vice-versa são em sua totalidade, o que na maioria das vezes não é necessário. Além de que, numa simples inserção, todos os arquivos de índice são trazidos para a máquina cliente e são atualizados um a um. Logo, com esse tráfego intenso, claramente a performance do sistema é comprometida. Provocando um acesso demásiadamente lento e conseqüentemente uma aceitação negativa por parte dos usuários, melhor dizendo, do cliente.

Dessa forma, foram apresentadas as propostas para resolução dos problemas acima, que serão apresentadas no tópico seguinte.

4. Proposta de Solução

As propostas de solução explicadas abaixo foram tomadas em equipe. O analista da empresa onde ocorreu o estágio, juntamente com os três estagiários que participaram do desenvolvimento, discutiram cada escolha que viria a ser feita para que fosse obtido sucesso no desenvolvimento do Sistema que deu origem ao estágio.

Para resolver o problema da interface, precisaríamos de uma ferramenta que fornecesse uma interface agradável, em termos visuais, e eficiente ao usuário do Sistema. Para isso, foi escolhido o Delphi 5 da Borland (Ver anexo 3), por ser uma ferramenta bastante poderosa em termos de usabilidade e pela sua facilidade de integração com vários Sistemas de Gerenciamento de Banco de Dados, dentre os quais está o Interbase 6.0.

Como SGBD, foi escolhido o Interbase 6.0. O principal motivo dessa escolha, além de ser um bom gerenciador de Banco de Dados, foi o fato de ser um software grátis. Assim, o cliente não teria um impacto tão grande, em termos financeiros, com as mudanças que seu Sistema atual viesse a sofrer. Diminuindo assim os gastos tanto para o cliente (Posto BR Cruzeiro), quanto para a Empresa Microsistemas, que também precisaria dispor do software escolhido como SGDB. Outro aspecto importante é a facilidade de uso do Interbase, além de sua robustez em termos de integridade e consistência de dados (Ver anexo 4), tal como um bom SGBD deve ser.

A ausência de documentação inerente ao processo de desenvolvimento também foi citada como problema. Para solucioná-lo, foi aplicada toda técnica de análise de sistemas estudada nas principais disciplinas do curso. A metodologia de análise escolhida foi a Análise Essencial, estudada com detalhes na disciplina de Sistemas de Informação.

Com a metodologia escolhida, a proposta foi criar os diagramas essenciais, tais como Diagrama de Fluxo de Dados (DFD) e suas Mini-especificações, Diagrama de Entidade/Relacionamento (DER) e o Dicionário de Dados. Antes destes, foi criado um documento formal com toda a descrição detalhada dos Requisitos do Sistema. Esses Requisitos foram baseados nas funcionalidades do Sistema a ser substituído, no conhecimento do analista e dono da Empresa Microsistemas, desenvolvedor do Sistema que estava sendo utilizado pelo cliente e, obviamente, na interação com o cliente antes e durante o desenvolvimento do novo Sistema.

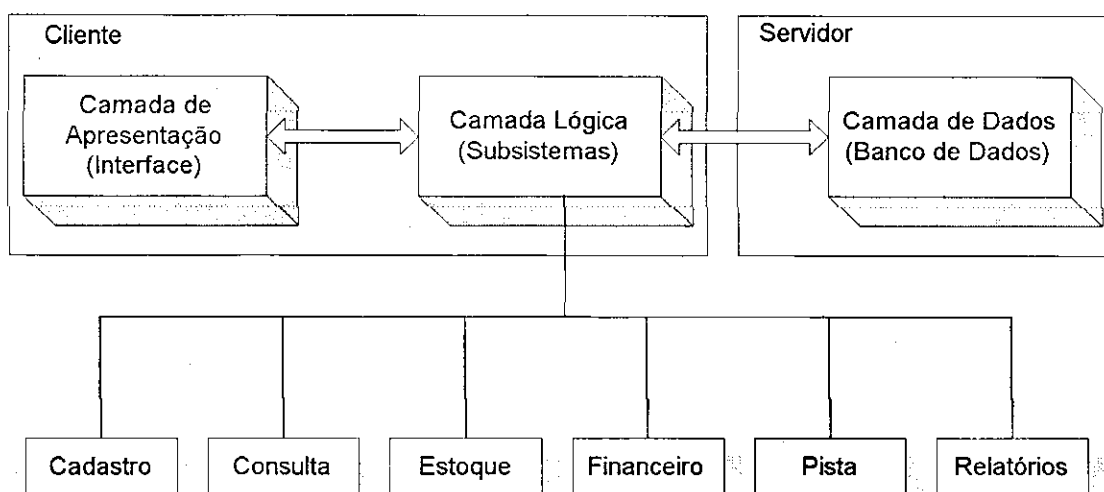
E por fim, solucionar o principal problema, o motivo que levou à necessidade de uma atualização, ou seja, o melhoramento da performance na manipulação dos dados. Com a eliminação da necessidade de lidar com arquivos, usando um SGBD, foi possível melhorar satisfatoriamente o desempenho do Sistema. Onde foi possível utilizar as facilidades da linguagem de manipulação de dados SQL, lidando com um Banco de Dados relacional.

No Sistema anterior havia muitos arquivos de índice que precisavam ser atualizados a cada inserção e remoção de dados. Além do fato de que estes arquivos trafegavam via link Embratel do servidor de banco de dados para a máquina cliente retornando para o servidor após sua atualização. Usando um SGBD esse tráfego é evitado, pois os índices são atualizados no próprio servidor pelo próprio gerenciador do banco de dados, no caso, o Interbase. E com o uso da linguagem SQL, foi possível diminuir consideravelmente o tráfego intenso de informações que havia anteriormente, pois nas requisições de dados feitas pela aplicação houve uma diminuição nessa carga através da filtragem do que realmente seria necessário ou não obter dos dados armazenados. Evitando que todos os registros de uma tabela, por exemplo, fosse carregados para a máquina cliente sem necessidade.

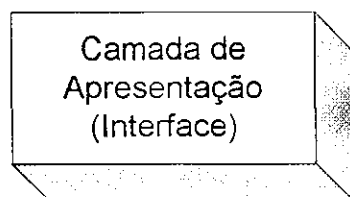
Assim concluem-se as propostas de solução para os problemas citados. E no próximo tópico serão detalhadas as funcionalidades do Sistema desenvolvido durante o estágio.

4.1. O Sistema de Automação de Postos de Gasolina (SAP)

O Sistema desenvolvido em Delphi controla toda a movimentação de mercadorias, compra e venda de produtos da rede de postos BR Cruzeiro em Pernambuco. Dá suporte às vendas realizadas na pista pelos frentistas, dentre os mais variados produtos, desde o combustível no abastecimento de veículos, até picolés, água, e outros produtos que geralmente são encontrados em postos de gasolina. Os postos também dispõem de bar, restaurante, hotel e lanchonete. Dessa maneira, o Sistema SAP também controla a movimentação dos garçons, gerentes de hotel e dos demais vendedores, nas suas mais variadas funções. Abaixo está um esboço da arquitetura do Sistema SAP.



Não apenas as movimentações de estoque, como também é controlada pelo Sistema toda a parte financeira dos postos de gasolina. Essa é também uma característica crucial do Projeto, pois aí é que entra a movimentação de dinheiro, lucros, despesas, além de controlar as ações dos funcionários e suas prestações de conta nos finais de turno. Em seguida, serão detalhados os subsistemas e camadas que foram desenvolvidos pelo estagiário Vladimir, os quais são: Interface, Cadastro, Consulta, Estoque, Financeiro e Pista.



A interface do Sistema, bem como todas as suas telas de controle foram desenvolvidas na linguagem de programação Delphi. Esta foi projetada de forma que sua manipulação pelo usuário fosse intuitiva. Seguindo o padrão Windows, foram

colocados Menus, barra de ferramentas, barra de status e botões de acesso, onde o usuário pode acessar rapidamente as funcionalidades desejadas.

No aspecto de usabilidade foi seguido o padrão do Sistema antigo na maneira de preenchimento e ativações de funcionalidades. O uso de metáforas⁴ nos ícones dos botões foi feito com bastante cautela. De forma que o usuário saiba o que vai acionar apenas conhecendo a figura. O acionamento das consultas foi otimizado permitindo que o usuário apenas teclando "Enter" realize a consulta desejada. Mesmo existindo botões e menus, foram colocadas teclas de atalho para agilizar a utilização do usuário mais experiente. Não exigindo do usuário o uso do teclado e do mouse ao mesmo tempo, o que se torna cansativo. Assim, o usuário poderá optar entre mouse e teclado para operar o Sistema. Essas e outras características tiveram fundamento no conhecimento em Interface Homem-Máquina⁵. Para consulta bibliográfica (Ref. [5])

No aspecto gráfico, foram utilizadas algumas figuras encontradas no próprio Delphi. Das quais algumas necessitaram de alterações de tamanho, brilho e outros ajustes. Também foi utilizado um pacote para Delphi 5, encontrado gratuitamente na Internet, para edição de textos em três dimensões.

Das formas citadas acima o design do projeto foi desenvolvido, seguindo as exigências do cliente.



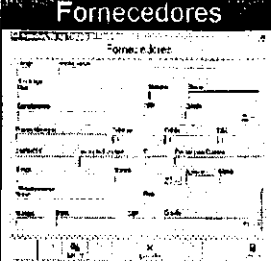

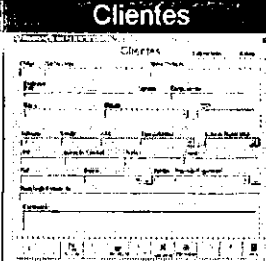
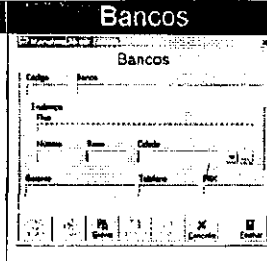
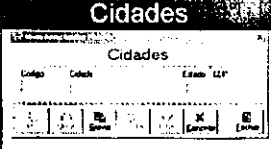
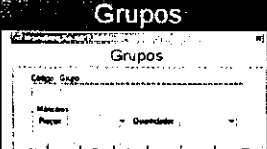

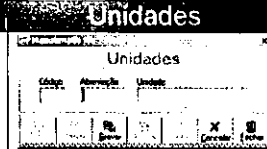

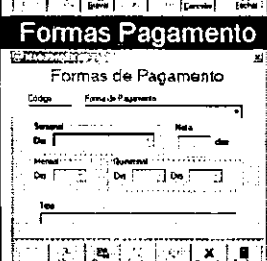
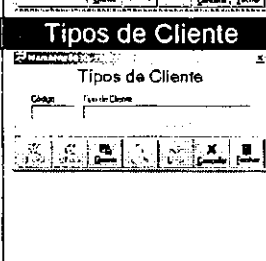
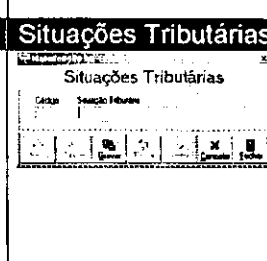
Tela principal do Sistema SAP

⁴ Metáfora – Recurso utilizado para representar a mesma semântica, porém com sintaxe diferente.

⁵ Interface Homem-Máquina – Curso voltado para usabilidade de sistemas de informação visto no PCT Motorola.

Cadastro




Várias são as entidades que precisam ser cadastradas para o funcionamento do Sistema. Dentre todas as mais importantes são Clientes, Produtos e Fornecedores. Foram projetadas telas de cadastro para cada uma das Entidades do Sistema, as quais estão abaixo.

Fornecedores	Produtos	Clientes	Bancos
			
Cidades	Grupos	Fabricantes	Unidades
			
Frentistas	Formas Pagamento	Tipos de Cliente	Situações Tributárias
			

As telas acima seguem um padrão nas suas funcionalidades. Possuem os seguintes botões de controle: *Novo*, com o qual o usuário insere um novo registro no banco de dados, ou seja, cadastra uma nova entidade; *Alterar*, onde o usuário pode alterar os campos de uma entidade já cadastrada e anteriormente selecionada; *Gravar*, que realiza a gravação dos dados informados pelo usuário nos campos de edição, tanto no modo de inserção como de alteração; *Excluir*, como o qual a entidade previamente selecionada pode ser excluída com a confirmação do usuário; *Consultar*, sendo de grande utilidade para consulta de algum registro que o usuário queira alterar, excluir ou até mesmo apenas conferir os dados armazenados; *Cancelar*, com o qual o usuário pode interromper alguma atividade em curso e finalmente o botão *Fechar*, para fechar o formulário de cadastro.

Algumas telas possuem características específicas e por isso possuem botões exclusivos. Como é o caso do formulário de Clientes que possui um botão que bloqueia e desbloqueia o Cliente, permitindo ou não que seja realizada alguma venda ao mesmo. Como também é o caso do formulário de Produtos, onde existe um botão que abre uma nova tela com as quantidades em estoque do produto selecionado.

Consulta

Todas as telas de cadastro possuem uma opção de consulta através do botão . Na mesma tela também existem dois botões   que são habilitados sempre que a consulta resulta em mais de um registro, possibilitando o usuário adiantar ou retornar nos registros de sua consulta.

Algumas consultas precisaram de telas diferentes para a sua realização, outras tiveram sua consulta na própria tela de cadastro, as quais precisavam de maior agilidade.


Em se tratando de uma tabela com um volume pequeno e de difícil aumento, como é o caso das *unidades* de venda e compra, as informações são mostradas por completo em uma grade. Ao clicar no botão de consulta o usuário tem acesso a todas as informações contidas na tabela que aparecem em uma outra janela.

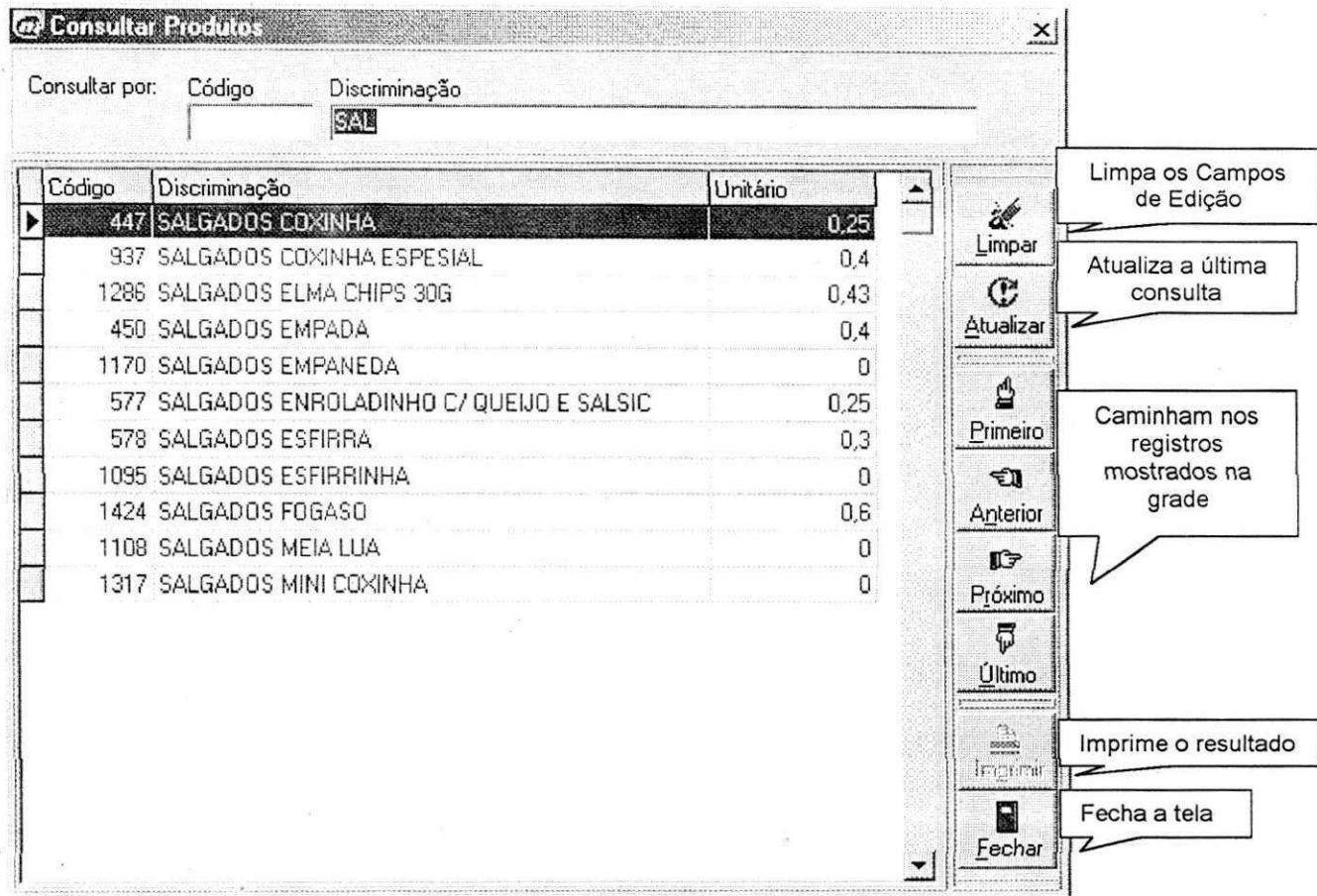


Código	Abreviação	Unidade
5	TON	Tonelada
6	MT	Metro
7	LT	Litro
8	LAT	Lata
9	MIL	Milheiro
10	DZ	Duzia
11	KG	Kilo Grama
12	BAL	Balde

As consultas mais otimizadas e que requeriam mais detalhes foram feitas também em uma tela separada, mas com os campos de edição para preenchimento dos valores a serem consultados. Dessa forma, o usuário terá apenas o resultado da consulta desejada e não todas as informações. Esse tipo de consulta foi organizada de duas formas.

Uma delas está disponível na tela principal para simples conferência, sem alteração ou exclusão. Também são disponíveis nas telas de controle de estoque (Entradas, Vendas, Pedidos, Transferências, Reposições – Ver Estoque), pois se o usuário não sabe o código do produto, por exemplo, ele abre essa tela de consulta para

consultar pelo nome do produto (discriminação), podendo também listar todos através do botão . Essa é feita apenas para Clientes, Produtos e Fornecedores. Geralmente o usuário precisará desse modo de consulta para impressão dos resultados. A tela descrita será mostrada abaixo.



The screenshot shows a window titled "Consultar Produtos" with a search bar and a table of products. The search bar is set to "Discriminação" with the value "SAL". The table lists products with their codes, descriptions, and unit prices. The first row is highlighted.

Código	Discriminação	Unitário
447	SALGADOS COXINHA	0,25
937	SALGADOS COXINHA ESPECIAL	0,4
1286	SALGADOS ELMA CHIPS 30G	0,43
450	SALGADOS EMPADA	0,4
1170	SALGADOS EMPANEDA	0
577	SALGADOS ENROLADINHO C/ QUEJO E SALSIC	0,25
578	SALGADOS ESFIRRA	0,3
1095	SALGADOS ESFIRRINHA	0
1424	SALGADOS FOGASO	0,6
1108	SALGADOS MEIA LUA	0
1317	SALGADOS MINI COXINHA	0

The sidebar on the right contains the following buttons and callouts:

- Limpar**: Limpa os Campos de Edição
- Atualizar**: Atualiza a última consulta
- Primeiro**: Caminham nos registros mostrados na grade
- Anterior**: Caminham nos registros mostrados na grade
- Próximo**: Caminham nos registros mostrados na grade
- Último**: Caminham nos registros mostrados na grade
- Imprimir**: Imprime o resultado
- Fechar**: Fecha a tela

E por fim a consulta que geralmente se resume a apenas um registro como resultado. Essa consulta foi usada na parte financeira onde é necessário o status das contas. Por isso, na consulta, o usuário informa se a conta a ser consultada deve ser paga, não paga ou ambas. A tela é mostrada abaixo.

Consultar Contas a Pagar

Status do Título
 Pago Não Pago

Título

Cód. Fornecedor

Período Início

Término

Em Atraso Até

Estoque

No subsistema Estoque, foram desenvolvidos os seguintes controles: Pedidos, Reposições, Transferências, Entradas e Vendas.

Pedidos dá suporte às encomendas feitas pelo posto aos seus fornecedores. Dessa forma, os pedidos são registrados de acordo com a data de pedido, fornecedor e seus itens que podem ser mais de um cadastrado no mesmo pedido. Todos os pedidos feitos na empresa são registrados e o usuário tem acesso ao valor total dos itens de pedido na hora do cadastro.

Manutenção na Tabela de Pedidos

Pedidos

Código	Data do Pedido	Data da Entrega	Fornecedor	Total
	22/09/2001	19 / / 19		

Ítems Pedidos				
Código	Discriminação	Quantidade	Preço	Total
*				

Código	Discriminação	Quantidade	Preço	Total
*				

Reposições trata das mudanças de produtos de um estoque para outro. Existem os estoques Depósito, Pista, Bar e Hotel. Logo, se um produto sai do depósito para ser vendido na pista ele deve ser cadastrado em reposições juntamente com a quantidade repostada. As quantidades estão registradas em cada produto na tabela de produtos, então, somente poderá ser repostada uma quantidade menor ou igual a que existe no depósito.

Reposições

Código Data Turno Estoque

 22/09/2001 19 [v]

Ítems Pedidos

Código	Discriminação	Quantidade
*		

Novo Alterar Gravar Excluir Consultar Cancelar Anterior Próximo Fechar

Transferências controla os produtos que saem de um posto para outro dentro da mesma rede de postos, cliente do Sistema. Assim, cada produto que sai de um dos postos da rede para outro posto, geralmente em outra cidade, é cadastrado na tela de transferências.

Manutenção na Tabela de Transferências

Transferências

Código Data Destino

 22/09/2001 15

Ítems Tranferidos

Código	Discriminação	Quantidade	Preço	Total
* *				

Novo Alterar Gravar Excluir Consultar Cancelar Inicializar Filtros Fechar

Entradas trata da compra de produtos aos fornecedores do posto. Controla tudo que entra no posto, tratando da geração de fatura ou não. Ou seja, existe a opção *Gera Financeiro* que controla os títulos que ainda não foram pagos. Essa opção habilita ou não o preenchimento das informações dos títulos a pagar. Se a compra não está sendo paga no momento, cada título deve ser registrado em *Contas a Pagar* (Ver *Financeiro*), para posterior pagamento. Os títulos em débito têm seu valor e sua referência armazenados na tabela de *Contas a Pagar*, dessa forma, o gerente pode pagá-la posteriormente, pois estará registrada como uma conta não paga.

Entrada de Produtos X

Entradas

Código Fornecedor Estoque Gera Financeiro Sim Não

Pedido	Fatura	Emissão	Entrada	Modelo	Série	Class. Fiscal
		//	22/09/2001			
Valor Contábil	Base de Cálculo	Alíquota	ICM Normal	Base/Substituição	ICM Substituição	
ICM por Entrada	ICM na Fonte	Valor do IPI	Valor do Produto	Isentas	Outras	
Valor do Frete	Outros Valores	Contábil	Observação			

Títulos	Número	Vencimento	Valor	Observação
		//		

Número	Vencimento	Valor Total	Observação
*			

Ítems de Entrada						
Código	Discriminação	Quantidade	Etq.	Unitário	ICM(%)	IPI(%)

Produto	Quantidade	Etiqueta	Unitário	Total	ICM(%)	IPI(%)
*						

Vendas controla as vendas ocorridas na pista de toda a rede de postos. O controle de vendas precisa ser muito robusto, de forma que evite inconsistências entre as vendas e as prestações de contas dos frentistas. Para isso, é verificada a escala (Ver Pista) do frentista que está sendo cadastrado com vendedor. Com a escala feita semanalmente, o frentista só tem permissão de vender determinado combustível de determinado bico. Não permitindo a venda se o frentista não estiver escalado devidamente. O controle também é feito de acordo com a quantidade em estoque, ou seja, se a quantidade vendida não condiz com a quantidade em estoque, a venda também não é confirmada.

Um aspecto importante na parte de vendas é a Forma de Pagamento. No cadastro do cliente, ele informa qual forma de pagamento ele irá adotar como cliente do posto. Essas formas podem ser: *Mensal*, onde o cliente informa qual dia do mês ele irá pagar; *Quinzenal*, onde o cliente escolhe duas datas (nas duas quinzenas) para pagar suas contas; *Semanal*, na qual o cliente diz qual dia da semana ele efetuará seu pagamento; *Nota*, onde o cliente informa o número de dias ele pagará após o preenchimento da nota.

No Sistema anterior existiam datas fixas para pagamento. Uma inovação do Sistema novo, o SAP, é que o cliente do posto pode

escolher a data que queira pagar mensalmente, quinzenalmente, semanalmente ou por nota. Dessa forma, aumentou a flexibilidade para os clientes que querem pagar em uma determinada data mais conveniente para o mesmo.

Vendas

Código: [] Frentista: [] Pagamento: [] Cliente: []

Placa: [] Km Atual: [] Data da Venda: [22/03/2001] [16] Data da Fatura: [22/03/2001] [16] Tumo: [TARDE]

Ítems de Venda

Código	Discriminação	Quantidade	Preço	Total
*				

Novo Atualizar Gravar Excluir Consultar Cancelar Anterior Próximo Total Fechar

Financeiro

A parte financeira, talvez a parte mais importante de todo o Sistema, controla tudo referente às contas a pagar, receber, prestação de contas dos frentistas e a geração de fatura.

Para gerar uma fatura, existem duas opções: por data, ou por cliente. A geração de fatura nada mais é que cadastrar uma nova conta a receber para que haja controle sobre os débitos dos clientes do posto.

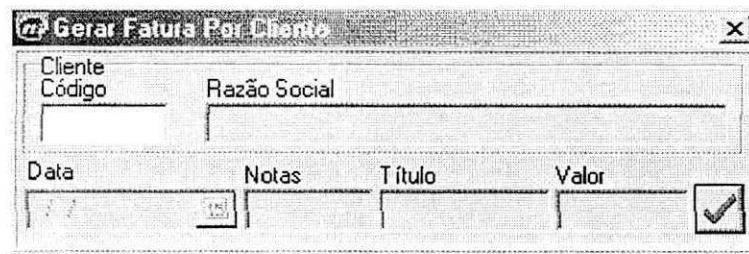
Para quem paga a vista, existe um tipo de cliente cadastrado chamado *consumidor*. Para os clientes que tem uma forma de pagamento fixa (mensal, quinzenal, semanal ou por notas) são geradas faturas de acordo com sua forma de pagamento. Dessa forma, obtendo a data de fatura do cliente em débito, sua conta é cadastrada e fica armazenada como não paga, até que o pagamento seja efetuado.

Para gerar fatura por data, o usuário emite a data que quer que gere. E as vendas que têm sua data de fatura na data informada pelo usuário vão para as contas como não pagas. Esse tipo de atividade de controle é feito diariamente para saber o que precisará ser pago no dia.



A screenshot of a software dialog box titled "Gerar Fatura". It contains a "Data" field with a date input showing "11" and "15", a checkmark icon, and two empty text input fields labeled "Faturas" and "Valor".

A geração de fatura por cliente se aplica aos clientes que querem pagar antes de sua data de fatura, ou seja, a sua conta ainda não foi cadastrada, pois ainda não chegou seu dia de pagamento. Neste caso, gera-se a fatura informando qual o cliente que está querendo pagar, efetuando o pagamento em seguida.



A screenshot of a software dialog box titled "Gerar Fatura Por Cliente". It features a "Cliente" section with "Código" and "Razão Social" fields. Below is a table with columns "Data", "Notas", "Título", and "Valor", and a checkmark icon in the bottom right corner.

Data	Notas	Título	Valor
11			

A prestação de contas ocorre quando um frentista vai entregar ao gerente o apurado durante o turno. Como suas vendas foram cadastradas com o tipo de pagamento que foi escolhido pelo cliente, o frentista terá que prestar conta do que tem em mãos de acordo com o que está registrado.

Manutenção da Tabela de Contas X

Contas

Código: [] Frentista: [ELISANGELA] Data: [19/09/2001] Turno: [15] [TARDE] []

Débitos		Créditos	
Dinheiro	[]	Dinheiro	[]
Cheque	[]	Cheque	[]
Cheque Pré.	[]	Cheque Pré.	[]
Cartão	[]	Cartão	[]
Vales Fiado	[]	Vales Fiado	[]
Totais	[R\$ 0,00]	Totais	[R\$ 0,00]
		Diferença	[R\$ 0,00]

[Novo] [Alterar] [Gravar] [Excluir] [Consultar] [Cancelar] [Anterior] [Próximo] [Fechar]

O pagamento e o recebimento são feitos nas seguintes telas abaixo.

Pagamento

Consultar por: Título [] Cod. Fornecedor [] "Enter" para consultar

Código	Título	Cód. Fornecedor	Vencimento	Status
[]	[]	[]	[]	[]

[Selecionar]
[Limpar]
[Atualizar]
[Printar]
[Anterior]
[Próximo]
[Último]
[AD-PU]
[Fechar]

Pagamento: [] Valor a Pagar: [] Valor Pago: [] Dinheiro: [] Cheque: [] [Pagar]

Recebimento

Consultar por: Título [] Cod. Cliente [] "Enter" para consultar

Conta	Código	Classe	Total
[]	[]	[]	[]

[Selecionar]
[Limpar]
[Atualizar]
[Printar]
[Anterior]
[Próximo]
[Último]
[AD-PU]
[Fechar]

Pagamento: [] Valor a Pagar: [] Valor Pago: [] Dinheiro: [] Cheque: [] Cheque Pré.: [] [Pagar]

Pista

Os Frentistas são escalados semanalmente pelo gerente local, de forma que em determinado *turno* (manhã – 06:01/14:00, tarde – 14:01/22:00 e noite – 22:01/06:00) ele terá acesso à determinada bomba de combustível chamada, no escopo do Sistema, de *Bicos*, uma vez que se trata dos bicos das máquinas de abastecimento. Cada *bico* está cadastrado no Sistema com a informação de qual produto ele irá fornecer. Sendo assim, as escalas dos frentistas são feitas de domingo a domingo, de forma que o mesmo seja escalado para trabalhar em um ou mais *bicos*, vendendo seus produtos condizentes. Então, uma venda só é confirmada na *pista* (como é chamado o terminal de vendas dos frentistas) se o vendedor (frentista) estiver escalado naquele *turno* para vender aquele *produto* (combustível) daquele *bico*. Em caso contrário a venda não é confirmada. Evitando que alguém perca sua venda. A tela para Escalas está apresentada abaixo.

Manutenção da Tabela de Escalas

Escalas

Código	Início	Final	Turno
	22	22	
Bico	Produto do Bico	Frentista	
6	GASOLINA COMUN		

Bico	Combustível	Frentista
1	ALCOOL HIDRATADO	ELISANGELA
2	GASOLINA SUPRA	ELISANGELA
3	GASOLINA COMUN	SILVIO
4	GASOLINA SUPRA	SILIANO
5	ALCOOL HIDRATADO	CARLOS
*		

Novo Alterar Gravar Excluir Consultar Cancelar Fechar

Uma outra característica importante do Sistema é a chamada de *Encerrantes*. Todo início de turno o frentista abre uma *encerrante* informando qual a marcação inicial da bomba de combustível a qual ele está escalado, ou seja, quantos litros de combustível estavam presentes no marcador da máquina. No final do turno, o frentista fecha a *encerrante*, informando quantos litros havia saído da mesma bomba, informado pelo marcador. Dessa forma, a prestação de contas de suas vendas se torna mais eficaz. Existe uma verificação, geralmente diária, para conferir se as bombas de combustíveis estão marcando os valores corretamente. Esta verificação é chamada de *Aferição*. A *Aferição* nada mais é que a retirada de uma quantidade determinada de combustível, colocada em um recipiente de volume conhecido. Dessa forma, compara-se a quantidade no recipiente com a quantidade registrada pela bomba. Portanto, a *aferição* deve ser levada em conta no fechamento de *Encerrantes*, pois é uma quantidade registrada como saída da bomba, mas que não foi vendida. Logo, é preciso que o frentista informe a quantidade de combustível retirada para *aferição*.

Existem três telas para as *Encerrantes*: Abertura de Encerrantes, Fechamento de Encerrantes e Manutenção de Encerrantes.

Abertura				Fechamento				Manutenção					
Abertura de Encerrantes				Fechamento de Encerrantes				Manutenção de Encerrantes					
Código	Data	Turno	Valor Inicial	Código	Data	Turno	Valor Final	Código	Data	Turno	Valor Inicial	Valor Final	Valor
06	02/01/04			01	02/01/04			01	02/01/04				
07	02/01/04			02	02/01/04			02	02/01/04				
08	02/01/04			03	02/01/04			03	02/01/04				
09	02/01/04			04	02/01/04			04	02/01/04				
10	02/01/04			05	02/01/04			05	02/01/04				
11	02/01/04			06	02/01/04			06	02/01/04				
12	02/01/04			07	02/01/04			07	02/01/04				
13	02/01/04			08	02/01/04			08	02/01/04				
14	02/01/04			09	02/01/04			09	02/01/04				
15	02/01/04			10	02/01/04			10	02/01/04				
16	02/01/04			11	02/01/04			11	02/01/04				
17	02/01/04			12	02/01/04			12	02/01/04				

Relatórios

A Impressão de relatórios foi desenvolvida pela estagiária *Leidjane Souto*.

4.2. ANÁLISE DO SISTEMA E PROJETO DO BANCO DE DADO

4.2.1. – ANÁLISE DO SISTEMA

Um projeto de Banco de Dados exige, sem dúvida alguma, uma análise minuciosa do sistema.

A fase de análise foi uma atividade desempenhada pelo supervisor técnico e também proprietário da empresa e pela estagiária, junto à equipe, durante todo o processo de desenvolvimento.

Dentre os problemas enumerados, foi citado a ausência de uma documentação detalhada do sistema. Tal fato originou-se do desconhecimento do supervisor técnico e antes analista, de uma metodologia que auxiliasse essa etapa

A estagiária propôs então o uso da análise Essencial . Seus componentes (DER, Dicionário de Dados, DFD e Mini-especificação) podem ser vistos nos anexos.

Apesar do sistema já existir, como já foi abordado, a análise teve que ser refeita para que se agregasse um controle mais amplo no que se refere a parte financeira.

Usou-se o Processo Unificado que por ser iterativo e incremental, permitiu que essa fase fosse revista a cada iteração, mediante modificações dos requisitos propostos pelo cliente.

4.2.2. PROJETO DO BANCO DE DADOS

Uma vez redefinida a análise, a estagiária propôs a criação do Banco de Dados da rede de postos, detendo-se na solução dos problemas referentes ao a migração dos dados dos arquivos Clipper (.dbf) para a novo banco de dados e no comprometimento da performance do sistema em uso.

4.2.2.1. MIGRAÇÃO DOS DADOS

Após inúmeras conversas com o cliente, concluiu-se que seria indispensável o reaproveitamento e manutenção dos dados do sistema anterior.

O maior problema ao qual a estagiária deparou-se foi o fato de migrar os dados de arquivos, sem consistência alguma, para uma base de dados consistente e segura.

Conforme abordado no problema, o sistema anterior armazenava seus dados em arquivos ".dbf" (Clipper) e, como todo arquivo, caracterizava-se por:

- possuir inúmeras redundâncias de informações de dados;
- falta de restrições de integridade entre os dados

Migrar esses dados para uma base consistente e segura seria a solução. Daí a estagiária, junto à equipe, propor o uso do Interbase 6.0 como SGBD (Sistema Gerenciador de Banco de Dados).

Partiu-se então para criação das tabelas do Banco de Dados, removendo-se as redundâncias presentes e inevitáveis quando se manipula arquivos.

As questões relacionadas às restrições de integridade, segurança, tolerância à falhas e o controle de concorrência a (***) passariam a ser executadas através das facilidades disponibilizadas pelo próprio SGDD, no caso específico, o Interbase 6.0.

Uma vez criado banco, restava migrar os dados do sistema anterior para a nova base de dados.

A partir daí, foram necessários algumas reuniões com o supervisor técnico para definição da ferramenta de migração a ser utilizada.

As ferramentas disponíveis e conhecidas até o presente momento migravam dados apenas de um banco para um outro. Programas de transferência de informações de arquivos ".dbf" para um banco não era de nosso conhecimento.

Era preciso adequar algum programa já existente e com funcionamento similar, para que assim, se convertesse dados de arquivos para colunas da tabela de um determinado banco.

Um outro desenvolvedor da empresa Microsistemas implementou um programa que foi adaptado pela estagiária. Alterações no código fonte do programa foram então introduzidas pela estagiária adequando às suas necessidades e possibilitando a migração dos dados com sucesso.

A estagiária utilizou portanto, como ferramenta de migração, o programa então denominado Transf, desenvolvido por um programador da empresa em Delphi 5.0 e adaptado pela própria estagiária.

Em virtude da inconsistência e falta de regras de integridade dos arquivos, tornou-se impraticável a transferência dos dados em totalidade.

Após novas reuniões com o cliente, concluiu-se migrar apenas os dados mais relevantes para o posto (cadastro de clientes, produtos, frentistas, fornecedores, etc) e aqueles relacionados à parte financeira (contas a pagar e a receber). As tabelas cujo conteúdo era referente apenas aos históricos dispensaria a migração. Apenas os dados mais recentes destas tabelas poderiam ser, eventualmente, digitados novamente.

O consenso alcançado visou também evitar a inconsistência de algumas informações no novo sistema.

A seguir será exibido um exemplo de código do Transf adaptado pela estagiária, com as alterações feitas para que se fossem removidas as inconsistências.

MIGRAÇÃO DA TABELA CLIENTES:

```
TbDestino.Append;  
    TbDestino.FieldByName('codcliente').AsInteger      :=  
StrToInt(TbOrigem.FieldByName('codcli').AsString);  
    TbDestino.FieldByName('razaosocial').AsString     :=  
TbOrigem.FieldByName('razsoc').AsString;  
    TbDestino.FieldByName('fantasia').AsString       :=  
TbOrigem.FieldByName('fantas').AsString;
```

```

        TbDestino.FieldbyName('rua').AsString                :=
TbOrigem.FieldbyName('endcli').AsString;
        TbDestino.FieldbyName('bairro').AsString            :=
TbOrigem.FieldbyName('bairro').AsString;
        TbDestino.FieldbyName('cidade').AsInteger           :=
StrToInt(TbOrigem.FieldbyName('codcid').AsString);
        TbDestino.FieldbyName('CEP').AsString               :=
TbOrigem.FieldbyName('C_E_P_').AsString;
        TbDestino.FieldbyName('contato').AsString           :=
TbOrigem.FieldbyName('contat').AsString;
        TbDestino.FieldbyName('CGC').AsString               :=
TbOrigem.FieldbyName('C_G_C_').AsString;
        TbDestino.FieldbyName('inscri').AsString            :=
TbOrigem.FieldbyName('inscri').AsString;
        TbDestino.FieldbyName('atividade').AsInteger        :=
StrToInt(TbOrigem.FieldbyName('codati').AsString);
        TbDestino.FieldbyName('CAE').AsString               :=
TbOrigem.FieldbyName('C_A_E_').AsString;
        TbDestino.FieldbyName('email').AsString            :=
TbOrigem.FieldbyName('E_MAIL').AsString;
        TbDestino.FieldbyName('pagamento').AsInteger        :=
StrToInt(TbOrigem.FieldbyName('codpgt').AsString);
        If (TbOrigem.FieldbyName('codbco').AsString)='999'// O arquivo DBF
dos bancos tinha como códigos strings, como na tabela bancos do novo
sistema o código passou a ser um contador, tivemos que colocar todas
as condições de códigos de bancos antigos recebendo o novo código.
Essas condições foram inseridas em todas as tabelas que tinham
relacionamento com a tabela bancos p/ garantir a integridade do sistema.
        then TbDestino.FieldbyName('banco').AsInteger :=1;

        If (TbOrigem.FieldbyName('codbco').AsString)='02'
        then TbDestino.FieldbyName('banco').AsInteger :=2;

        If (TbOrigem.FieldbyName('codbco').AsString)='03'
        then TbDestino.FieldbyName('banco').AsInteger :=3;

        If (TbOrigem.FieldbyName('codbco').AsString)='04'
        then TbDestino.FieldbyName('banco').AsInteger :=4;
        If (TbOrigem.FieldbyName('codbco').AsString)='05'
        then TbDestino.FieldbyName('banco').AsInteger :=5;
        If (TbOrigem.FieldbyName('codbco').AsString)='BB'
        then TbDestino.FieldbyName('banco').AsInteger :=6;
        If (TbOrigem.FieldbyName('codbco').AsString)='CF'
        then TbDestino.FieldbyName('banco').AsInteger :=7;

        TbDestino.FieldbyName('data_nasc').AsDateTime      :=
TbOrigem.FieldbyName('d_nasc').AsDateTime;
        TbDestino.FieldbyName('credito').AsFloat           :=
TbOrigem.FieldbyName('credit').AsFloat;

```

```

        TbDestino.FieldbyName('debito').AsFloat           :=
TbOrigem.FieldbyName('debito').AsFloat;
        TbDestino.FieldbyName('pos_referencia').AsString :=
TbOrigem.FieldbyName('p_refe').AsString;
        TbDestino.FieldbyName('observacao').AsString     :=
TbOrigem.FieldbyName('obser1').AsString;
        TbDestino.FieldbyName('desconto').AsFloat        :=
TbOrigem.FieldbyName('descon').AsFloat;
        TbDestino.FieldbyName('status').AsString         :=
TbOrigem.FieldbyName('status').AsString;

```

O arquivo "clientes.dbf" apresentava inúmeras inconsistências. Uma delas era o fato de possuir vários códigos de cidade que não estavam na tabela de cidades.

O problema foi solucionado da seguinte forma:

1. O programador Clipper do sistema anterior alterou o programa inserindo uma condição que verificava todos os códigos de cidades, em clientes, que não estavam presentes na tabela de "cidades.dbf" e, quando isso era confirmado, setava o atributo cidade em clientes com o valor 99999, arbitrariamente escolhido;
2. No banco, foi inserido na tabela de cidades esse valor 99999, declarando a cidade como indefinida;
3. A migração pode ser feita então com sucesso.

MIGRAÇÃO DA TABELA CONTASRECEBER:

```

var k:integer;
begin
    k := 0;
    TbOrigem.First;
    While not TbOrigem.eof do begin
        TbDestino.Append;
        k := k + 1;
    TbDestino.FieldbyName('codcontasrece').AsInteger := k; // tabela destino do
banco recebe um contador inserido no código, pois foi acrescentado um
surgate na tabela do banco que antes não tinha no arquivo . DBF

TbDestino.FieldbyName('fatura').Asstring           :=
TbOrigem.FieldbyName('fatura').Asstring;

TbDestino.FieldbyName('cliente').Asinteger        :=
TbOrigem.FieldbyName('codcli').Asinteger;
    If (TbOrigem.FieldbyName('codbco').Asstring)='999' // Condições
inseridas mediante o relacionamento com a tabela bancos.
        then TbDestino.FieldbyName('banco').AsInteger :=1;

```

```

If (TbOrigem.FieldbyName('codbco').Asstring)='02'
    then TbDestino.FieldbyName('banco').AsInteger :=2;

If (TbOrigem.FieldbyName('codbco').Asstring)='03'
    then TbDestino.FieldbyName('banco').AsInteger :=3;

If (TbOrigem.FieldbyName('codbco').Asstring)='04'
    then TbDestino.FieldbyName('banco').AsInteger :=4;
If (TbOrigem.FieldbyName('codbco').Asstring)='05'
    then TbDestino.FieldbyName('banco').AsInteger :=5;
If (TbOrigem.FieldbyName('codbco').Asstring)='BB'
    then TbDestino.FieldbyName('banco').AsInteger :=6;
If (TbOrigem.FieldbyName('codbco').Asstring)='CF'
    then TbDestino.FieldbyName('banco').AsInteger :=7;

```

```

TbDestino.FieldbyName('cartei').Asinteger           :=
1;//TbOrigem.FieldbyName('bico__').Asinteger;
TbDestino.FieldbyName('dt_fat').Asdatetime          :=
TbOrigem.FieldbyName('dt_fat').Asdatetime;
TbDestino.FieldbyName('dt_emi').Asdatetime          :=
TbOrigem.FieldbyName('dt_emi').Asdatetime;
TbDestino.FieldbyName('dt_ven').Asdatetime          :=
TbOrigem.FieldbyName('dt_ven').Asdatetime;
TbDestino.FieldbyName('dt_pgt').Asdatetime          :=
TbOrigem.FieldbyName('dt_pgt').Asdatetime;
TbDestino.FieldbyName('juro_p').Asfloat             :=
TbOrigem.FieldbyName('juro_p').Asfloat;
    TbDestino.FieldbyName('mult_p').Asfloat          :=
TbOrigem.FieldbyName('mult_p').Asfloat;
    TbDestino.FieldbyName('tolerancia').Asinteger    :=
TbOrigem.FieldbyName('tolera').Asinteger;
    TbDestino.FieldbyName('valort').Asfloat          :=
TbOrigem.FieldbyName('valort').Asfloat;
    TbDestino.FieldbyName('valorf').Asfloat          :=
TbOrigem.FieldbyName('valorf').Asfloat;
    TbDestino.FieldbyName('vjuros').Asfloat          :=
TbOrigem.FieldbyName('vjuros').Asfloat;
    TbDestino.FieldbyName('vmulta').Asfloat          :=
TbOrigem.FieldbyName('vmulta').Asfloat;
    TbDestino.FieldbyName('vdesconto').Asfloat      :=
TbOrigem.FieldbyName('vdesco').Asfloat;
    TbDestino.FieldbyName('valorp').Asfloat          :=
TbOrigem.FieldbyName('valorp').Asfloat;
    TbDestino.FieldbyName('valorr').Asfloat          :=
TbOrigem.FieldbyName('valorr').Asfloat;
    TbDestino.FieldbyName('observ').Asstring        :=
TbOrigem.FieldbyName('observ').Asstring;

```

```

    TbDestino.FieldbyName('pg_din').Asfloat           :=
TbOrigem.FieldbyName('pg_din').Asfloat;
    TbDestino.FieldbyName('pg_chq').Asfloat           :=
TbOrigem.FieldbyName('pg_che').Asfloat;
    TbDestino.FieldbyName('pg_chp').Asfloat           :=
TbOrigem.FieldbyName('pg_chp').Asfloat;
    TbDestino.FieldbyName('titulo').Asstring          :=
TbOrigem.FieldbyName('titulo').Asstring;
    TbDestino.Post;
    TbOrigem.Next;

```

Apesar de ser mostrado apenas a migração destas duas tabelas, a estagiária migrou os dados das seguintes tabelas: Bancos, Bicos, CFOPENTRADA, CFOPSAIDA, ContasPagar, Cidades, Encerrantes, Escalas, Fabricantes, Fornecedores, Frentista, Grupos, Produtos, Situação e Unidades (devido os relacionamentos).

4.2.2.2. GARANTIA DE PERFORMANCE: USO DE PROCEDURES

O subsistema referente à parte financeira trouxe inovações que poderiam acarretar um impacto na performance do atual sistema. O controle financeiro já foi devidamente explicado no tópico 4.1 – subsistema financeiro do presente relatório

Além das formas de pagamento por fatura e as formas convencionais (cartão, cheque, ...) já mencionadas, existe o pagamento por nota para pagar com 30 dias. Assim, esta forma de pagamento difere da fatura, pois todas as compras do cliente por fatura geram uma única conta. Já por nota, a cada compra feita, gera-se uma conta individualmente com data de vencimento 30 dias após a compra.

No sistema anterior não havia como recuperar os itens associados a cada compra. O cliente recebia apenas o valor total da fatura ou da nota.

O novo sistema deveria garantir essa funcionalidade. Após sucessivas reuniões, decidiu-se que a parte financeira seria implementada no banco.

Vejamos o porquê através de um exemplo prático.

Supomos que, em média, haja 100 clientes com a mesma data de vencimento da fatura. A rotina na aplicação que gera fatura por data é executada diariamente. Ao ser "chamada" ela teria que:

- fazer uma consulta SQL na tabela de vendas;
- somar o total das vendas feitas aquele cliente;
- retornar à aplicação este valor;
- com um novo comando SQL, inserir este valor na tabela de Contas a Receber;
- atualizar o status da venda;
- fazer uma nova consulta SQL na tabela de vendas;
- finalmente inserir em **cada** item de **cada** venda o código da conta a receber. Porque somente assim, se poderia garantir a recuperação de cada item associado ao valor total da fatura do cliente.

Isto 100 vezes !

Num link Embratel isso poderia comprometer consideravelmente a performance do nosso sistema.

Concluiu-se então, que a parte financeira seria tratada no banco através de Procedures e Triggers.

As vantagens que justificam o uso das procedures pela estagiária podem ser descritas a seguir:

Melhoramento de Performance

Os procedimentos armazenados são executados no servidor e não na máquina cliente. Dependendo do trabalho a ser realizado, o fato de o processamento ser feito no servidor e não na estação pode reduzir muito o tráfego de informações na rede.

Além disso, como normalmente a máquina em que está instalado o servidor é mais potente, isto também pode gerar um ganho de performance.

Compartilhamento de Código

Uma vez escrito, um mesmo procedimento armazenado pode ser utilizado por varias aplicações.

Reduzir os Custos de Manutenção

Quando um procedimento armazenado é atualizado, essas modificações aplicam-se automaticamente a todas as aplicações que as usam, sem que tenham que ser compiladas novamente.

Existem basicamente dois tipos de Procedimento Armazenado:

O primeiro deles são procedimentos de seleção (Select Procedures) que, ao serem invocados a partir de uma aplicação, normalmente resultam em uma tabela ou visão. Devem ser definidos tendo em vista basicamente retornar um ou mais valores (parâmetros de saída).

São utilizados dentro das aplicações no lugar do table ou view em uma instrução SQL do tipo SELECT.

O segundo tipo são procedimentos executáveis (Executable Procedures) que podem ou não retornar valores para a aplicação que os ativou.

Este tipo de procedimento é ativado dentro de uma aplicação por meio da instrução EXECUTE PROCEDURE.

PROCEDURE QUE GERA FATURA POR DATA:

SET TERM ^ ;

/* Stored procedures */

CREATE PROCEDURE faturapordata

```
(  
PARDT TIMESTAMP  
)
```

AS

```
declare variable  
soma integer;  
declare variable  
recebecli integer;  
declare variable  
recebevenda integer;  
declare variable  
ven integer;  
declare variable  
prod integer;  
begin  
for  
select cliente from vendas  
where dtfat =:pardt and ctrl f ='n'  
into :recebecli  
do  
begin  
insert into contasreceber ( valort )  
select sum (totalv)  
from vendas  
where cliente= :recebecli and dtfat =:pardt and ctrl f ='n';  
soma = gen id (gen codcontasrece,0);  
for  
select numero from vendas  
where cliente= :recebecli and dtfat =:pardt and ctrl f ='n'  
into :recebevenda  
do  
begin  
for  
select venda , produto from item_venda  
where venda= :recebevenda  
into :ven , :prod  
do  
begin  
update item_venda  
set contarecebe=:soma  
where venda= :ven and produto= : prod;  
end  
end  
update contasreceber set cliente = :recebecli  
where codcontasrece =:soma;  
update contasreceber set dt fat = :pardt  
where codcontasrece =:soma;  
update contasreceber set titulo = :soma
```

```

where codcontasrece =:soma;
update vendas set ctrl f ='s'
where cliente =:recebecli;
end
SUSPEND;
END

```

PROCEDURE QUE GERA FATURA POR CLIENTE E DATA:

SET TERM ^ ;

/* Stored procedures */

CREATE PROCEDURE FATURACLIEDATA

```

(
  PARDT TIMESTAMP,
  PARCLI INTEGER
)

```

AS

```

declare variable
soma integer;
declare variable
recebevenda integer;
declare variable
ven integer;
declare variable
prod integer;
begin
  insert into contasregeber ( valor )
  select sum (totalv)
  from vendas
  where cliente= :parcli and dtfat =:pardt and ctrl f ='n';
  soma = gen_id (gen_codcontasrece,0);
  for
  select numero from vendas
  where cliente= :parcli and dtfat =:pardt and ctrl f ='n'
  into :recebevenda
  do
  begin
    for
    select venda , produto from item venda
    where venda= :recebevenda
    into :ven , :prod
    do
    begin
      update item venda

```



```

set contarecebe=:soma
where venda= :ven and produto= : prod;
end
end
update contasreceber set cliente = :parcli
where codcontasrece =:soma;
update contasreceber set dt fat = :pardt
where codcontasrece =:soma;
update contasreceber set titulo = :soma
where codcontasrece =:soma;
update vendas set ctrl f ='s'
where cliente =:parcli and ctrl f ='n';
SUSPEND;
END

```

PROCEDURE QUE GERA FATURA POR NOTA:

SET TERM ^;

/* Stored procedures */

CREATE PROCEDURE INSERIRPORNOTA

```

(
  PARDT TIMESTAMP,
  PARCLI INTEGER,
  TOTAL NUMERIC(15, 2),
  PARVENDA INTEGER
)

```

AS

```

declare variable
soma integer;
declare variable
ven integer;
declare variable
prod integer;
begin
  insert into contasreceber ( valor, cliente, dt fat) values (:total, :parcli, :pardt);
  soma = gen id (gen codcontasrece,0);
  for
    select venda , produto from item venda
    where venda= :parvenda
    into :ven , :prod
  do
    begin
      update item venda
      set contarecebe=:soma

```

```
where venda= :ven and produto= : prod;  
end  
update contasreceber set titulo = :soma  
where codcontasrece =:soma;  
SUSPEND;  
END
```

Mediante o uso de procedures no subsistema financeiro, a performance desejada satisfaz o cliente.

5. Atividades Desenvolvidas

5.1. Estudo das Tecnologias

Após a escolha das tecnologias que seriam utilizadas no processo de desenvolvimento do Sistema, foi necessário um estudo da linguagem de programação Delphi, mais precisamente o Delphi 5 (Ver Anexo 3). Essa atividade foi realizada durante as duas primeiras semanas do estágio. Também foi preciso uma familiarização com o SGBD Interbase 6.0, pois não havia nenhum conhecimento de tal sistema anteriormente.

5.2. Definição dos Requisitos

A definição dos requisitos foi baseada no sistema anteriormente utilizado pelo cliente, sistema esse, como já mencionado, desenvolvido em Clipper. O analista da Empresa, desenvolvedor do antigo Sistema, passou todas as informações inerentes ao funcionamento e controle da rede de postos. Como havia sido o analista do sistema anterior, não houve dúvidas quanto aos requisitos. Durante o desenvolvimento o cliente também foi consultado, dando *feedback* para melhoramentos. Tal atividade foi concluída na segunda semana de estágio, juntamente com a definição do escopo do problema. (Ver Anexo 5)

5.3. Criação do Modelo Conceitual

O modelo conceitual foi criado logo após o levantamento de requisitos, pelos estagiários envolvidos no desenvolvimento do projeto. Utilizando Análise Essencial (Ver Anexo 2), foi criado um diagrama de Entidade/Relacionamento (Ver Anexo 4) descrevendo todas as entidades pertencentes ao sistema, bem como quais os relacionamentos entre as mesmas. Essa atividade foi realizada na terceira semana de estágio.

5.4. Implementação e Testes

Nesta etapa foi dado início à implementação do Sistema, começando pela parte de cadastro (Clientes, Fornecedores, Produtos, ...).

Os testes foram realizados tanto na Microsistemas quanto no Cliente. Na Microsistemas foi colocada uma conexão via linha telefônica para simular o link utilizado pelo cliente para acesso aos dados. A base de dados foi instalada num servidor na casa de um amigo do dono da empresa. Essa base era acessada da empresa pela aplicação para testes de velocidade e funcionalidades. No cliente os testes foram realizados na presença do mesmo, onde o próprio cliente pôde utilizar o software que estava sendo desenvolvido e dar suas opiniões e críticas. Dessa forma, com a interação com o cliente, foi possível alcançar sua satisfação, chegando ao resultado esperado.

5.5. Edição do Relatório de Estágio

O Relatório final começou a ser desenvolvido antes do término da implementação do Sistema. No último mês de estágio foi dado início na confecção do Relatório.

6. Conclusão

O programa de estágio integrado é de fundamental importância para um curso que busca a qualidade na formação de pessoal, pois é através do estágio que o aluno tem a oportunidade de aplicar na prática os conhecimentos teóricos adquiridos ao longo do curso.

O estágio integrado traz vantagens tanto para alunos quanto para as empresas. Para o aluno estagiário, essa oportunidade de treinamento no mercado de trabalho o prepara para a vida profissional e é, muitas vezes, a única forma de se adquirir experiência enquanto estudante universitário. Como a experiência é um requisito fundamental exigido pelas empresas, o estágio, se torna uma ferramenta indispensável à qualificação profissional mínima. Até mesmo no sentido financeiro o estágio se torna importante para o estagiário, pois o mesmo começa a ter idéia de quão gratificante se torna ser recompensado por um trabalho bem feito. Para as empresas, o estágio é importante, pois é possível criar e manter a qualidade de seus futuros funcionários desde cedo, e até mesmo descobrir talentos promissores. Além disso, as empresas podem usar o programa de estágio integrado como forma de reciclagem de informação, contratando estagiários treinados em novas tecnologias.

7. Referência Bibliográfica

- [1] MCMENAMIM, Stephen M. e PALMER, John F. Análise Essencial de Sistemas, São Paulo Editora Mcgraw Hill, 1991.
- [2] SILBERSCHATZ, Abraham; KORTH, Henry F. e SUDARSHAN, S. Sistema de Banco de Dados, São Paulo, Editora Makron Books, 1999.
- [3] CORNELL, Gary e STRAIN Troy. Delphi: Segredos e Soluções, São Paulo, Editora Makron Books, 1995.
- [4] CANTÙ, Marco, Dominando o Delphi 5 "A Bíblia" Editora Makron Books, 1999.
- [5] SHNEIDERMAN, Ben, WESLEY, Addison: Designing the User Interface - Strategies for Effective Human-Computer Interaction, 1998

Anexo 1

Plano de Estágio

1. Ambiente do Estágio

A empresa: *Microsistemas Informática LTDA*. Rua Floriano Peixoto, 229, Centro – CEP.: 58.100-001. Fonefax (0**83) 342 4452 / 972 0727 – Campina Grande – PB. E-mail: microsis@cgnet.com.br.

A empresa disponibiliza três máquinas com as seguintes configurações:

- 4) Pentium II 300, 128 MB de RAM, 20 GB de HD
- 5) Pentium III 800, 128 MB de RAM, 20 GB de HD
- 6) Athlon 900, 128 MB de RAM, 30 GB de HD

O público alvo da empresa é a indústria e comércio, tendo como principais clientes:

- 1) Café Aurora
- 2) Tubos Nogueira
- 3) Pneumax LTDA
- 4) Postos BR Cruzeiro
- 5) CIPAN LTDA (Comércio e Indústria de Produtos Alimentícios do Nordeste)
- 6) SAMIC (Serviço de Assistência Médica em Campina Grande)
- 7) Indústria Alimentícia 3 de Maio S.A.

2. Supervisão

2.1. Supervisor Técnico: João Bosco de Macedo

Endereço : Rua Henrrimar C. de Oliveira nº 87 apto 203, Alto Branco Campina Grande - PB

2.2. Supervisor Acadêmico: Joseluze Cunha

3. Resumo do Problema Objeto do Estágio

3.1. Reengenharia de um sistema de gerência de estoque e venda da rede de postos de gasolina BR Cruzeiro, de porte médio e que incorpora um bar e restaurante, hotel e lanchonete. O sistema caracteriza-se como um software de controle de estoque e venda não apenas de combustíveis, mas também dos produtos vendidos na lanchonete, no bar e restaurante, no hotel e aqueles expostos na pista e vendidos pelos próprios frentistas (água, refrigerante, sucos, salgados, sorvetes, etc.).

O sistema existente encontra-se implementado em Clipper. Tal linguagem manipula apenas arquivos. Com a expansão do sistema de informação da rede de postos, tornou-se inviável a utilização de arquivos por questão de performance. Principalmente por existir um link via Embratel entre os dois principais postos localizados nas cidades de Pesqueira e Arco Verde, ambas em Pernambuco. Por esse link a aplicação em Arco Verde acessa o banco de dados localizado na cidade de Pesqueira em um servidor Linux.

3.2. Ausência de uma documentação formal do processo de desenvolvimento do sistema.

4. Proposta de Solução

4.1. Migrar o sistema para Delphi visando o melhoramento das interfaces com o usuário.

4.2. Agilizando o desempenho do sistema torna-se evidente a migração das informações dos arquivos para um banco de dados.

4.3. Modelagem do sistema utilizando Análise Essencial

4.4. Mudança de tecnologia, utilizando uma abordagem Orientada a Objetos. Será imprescindível então o uso de pelo menos ferramentas tais como: Rational Rose 2000, Borland Delphi 5 e Interbase 6.0.

5. Atividades a serem desenvolvidas

5.1 Cronograma:

- Análise de Requisitos juntamente com a Equipe como a qual irei trabalhar. Esta será feita baseada no aplicativo existente em Clipper, atualmente em funcionamento na Empresa cliente. Bem como baseada nas novas funcionalidades exigidas pelo cliente. Um exemplo desta é o uso do aplicativo em redes utilizando um link da Embratel;
- Projeto do Modelo de Entidade Relacionamento. Este será baseado nas tabelas já existentes em arquivos Clipper, após uma melhor análise da mesma, com exclusão de possíveis redundâncias e outros problemas que possam existir por falta de um projeto relacional anterior;
- Implementação do banco de dados no que se refere do sistema em si, cadastro, financeiro e as rotinas da pista, usando como SGBD Interbase 6.0;
- Migração dos dados e implementação da parte financeira mediante o uso de procedures
- Testes do Aplicativo diretamente com o cliente em Arco Verde.

Milestones	Data Inicial	Data Final
Requisitos	04/06/2001	08/06/2001
Análise	11/06/2001	15/06/2001
MER	18/06/2001	20/06/2001
MR	20/06/2001	22/06/2001
Impl. BD do posto no que se refere a parte do sistema em si, cadastro, financeiro e as rotinas da pista	25/06/2001	27/07/2001
Migração dos dados e implementação da parte financeira mediante o uso de procedures	25/06/2001	30/08/2001
Testes	30/08/2001	30/09/2001

5.2 Subsistemas a serem desenvolvidos:

Subsistemas	Impl. da Aplicação	Impl. Do banco	Migração dos dados
Sistema		X	X
Cadastros		X	X
Estoques			X
Financeiro		X	X
Rotinas da Pista		X	*
Rotinas do Restaurante			*
Rotinas do Hotel			*
Relatórios	X		*

(*) Dispensa a migração dos dados.

6. Aprovação

João Bosco de Macedo
(Supervisor Técnico)

Joseluze Cunha
(Supervisora Acadêmica)

Camilo de Lelis Gondim
(Coordenador do Estágio Integrado)

Vlademir Lenin Donato Batista
(Estagiário)

Análise Essencial

A análise Essencial tem como enfoque principal os eventos e preocupa-se, principalmente, em encontrar e definir os requerimentos verdadeiramente essenciais à construção do sistema sem preocupar-se com requerimentos tecnológicos.

Esta ferramenta é baseada em um esboço conceitual composto por: descrição das características do sistema, classificação dos componentes da essência do sistema e classificação das restrições de implementação.

O processo de desenvolvimento de um sistema deve ter os seguintes passos: identificar a essência do sistema, selecionar a encarnação da essência e construir o sistema.

Identificação do Sistema

Quando há interação entre o sistema e o ambiente através da implementação de respostas aos eventos do sistema, diz-se que o sistema está informatizado. Um evento é um acontecimento do mundo exterior que requer do sistema uma resposta e uma resposta é o conjunto de ações executadas pelo sistema sempre que ocorre um determinado evento.

As respostas podem ser:

Adhoc - são formadas espontaneamente pelo sistema em resposta a um evento não previsto.

Planejada - é a resposta que é determinada antes da ocorrência do evento.

Componentes da essência de um sistema

Essência de um sistema são todas as características de um sistema de resposta planejadas que existiriam se o sistema fosse implementado com tecnologia perfeita. Consiste de algumas atividades e da memória essencial. As atividades fundamentais são aquelas que executam uma tarefa que é parte da finalidade declarada do sistema. É composta de uma resposta planejada e uma definição do estímulo da atividade. A memória essencial serve para armazenar os itens de dados desde o tempo em que eles se tornam disponíveis até o momento em que são utilizados pelas atividades fundamentais.

As atividades custodiais estabelecem e mantêm a memória essencial do sistema pela obtenção e armazenamento da informação

necessária às atividades fundamentais. Elas também atualizam a informação armazenada de modo que esta permaneça correta.

Modelo de Implementação

Diz respeito à implementação do sistema. Neste modelo são colocadas todas as características tecnológicas.

O modelo é subdividido em dois outros modelos: o modelo ambiental e o modelo comportamental.

O modelo ambiental é voltado para fora do sistema, para o ambiente em que está inserido. Este modelo representa a interface do sistema com o mundo exterior, mostrando a interação do sistema com os elementos externos a ele.

Os componentes do modelo ambiental são:

Lista dos eventos que afetam o sistema : O primeiro passo na especificação de um sistema é identificar a quais eventos do mundo exterior ele deverá responder. A lista de eventos do sistema está ligada à sua finalidade, uma vez que esta finalidade é atender a determinadas necessidades e estas são decorrentes de eventos que acontecem no mundo exterior.

Diagrama de contexto : Representa o sistema por um único processo e suas interligações com as entidades externas, mostrando apenas as interfaces do sistema com o ambiente em que ele está inserido.

Declaração dos objetivos do sistema : Procura responder questões como: finalidade do sistema, problemas a serem resolvidos com a implementação do sistema, requisitos que devem ser atendidos. A declaração deve ser elaborada em poucas frases e com uma linguagem simples, para que ela seja entendida pelos usuários do sistema e pela administração da empresa.

O modelo comportamental é o modelo do interior do sistema. Sua finalidade é mostrar as ações que o sistema deve executar para responder adequadamente aos eventos previstos no modelo ambiental.

Os componentes do modelo comportamental são:

Diagrama de Entidade-Relacionamento (DER) : Ferramenta gráfica para representar classes de entidades, relacionamentos e atributos.

Dicionário de Dados : é um repositório de informações sobre os componentes do sistema.

Miniespecificação: descrição das funções primitivas do modelo funcional. As principais técnicas de especificação são: português estruturado, pseudocódigo, tabela de decisão e árvore de decisão.

Ferramenta de Desenvolvimento

Borland Delphi 5

Principais Características do Delphi

- Compilador/otimizador de código mais rápido do mercado, gerando executáveis rápidos e puros, sem run-time
- Totalmente orientado a objetos e com suporte a threads e OLE Automation
- Baseado em componentes, com facilidade de criação de componentes nativos, além de controles ActiveX, inclusive com disponibilidade do código fonte dos componentes padrão
- Programação two-way, utilização de métodos visuais ou diretamente sobre o código
- Suporte a manipulação de exceções, que permite criar aplicações mais robustas e com maior segurança
- Acesso rápido e seguro a bancos de dados através do Borland Database Engine (**BDE**), com facilidades de manipulação. *Utilizado no Projeto em pauta.*
- Criação de relatórios no próprio executável, com utilização de componentes nativos
- Facilidade de upsizing para bancos de dados cliente/servidor
- Capacidade de criação de aplicações multi-tier, com objetos distribuídos
- Suporte a código in-line, em assembly
- Capacidade de criação de outros tipos de utilitários, como DLL's, Screen Saver's e aplicações CGI..
- Literatura diversificada
- Fluxo de programação baseado em eventos

Versões Disponíveis

- A versão básica (a edição "Standard") é para ser usado por programadores iniciantes e casuais do Delphi.
- O segundo nível (a edição "Professional") é dirigida à projetistas profissionais. Ela inclui todos os recursos básicos, além de suporte à programação de banco de dados, amplo suporte a servidores web (webBroker) e algumas das ferramentas externas.
- O Delphi completo (a edição " Enterprise", anteriormente chamada de "Client/Server Suíte") tem por objetivo os projetistas que estão construindo aplicativos empresariais. Ele inclui SQL Links para conexão Cliente/Servidor BDE nativas, componentes ADO e InterBase Express, suporte a aplicativos

multiusuários, internacionalização e arquitetura de três camadas, e muitas outras ferramentas, incluindo SQL Monitor.

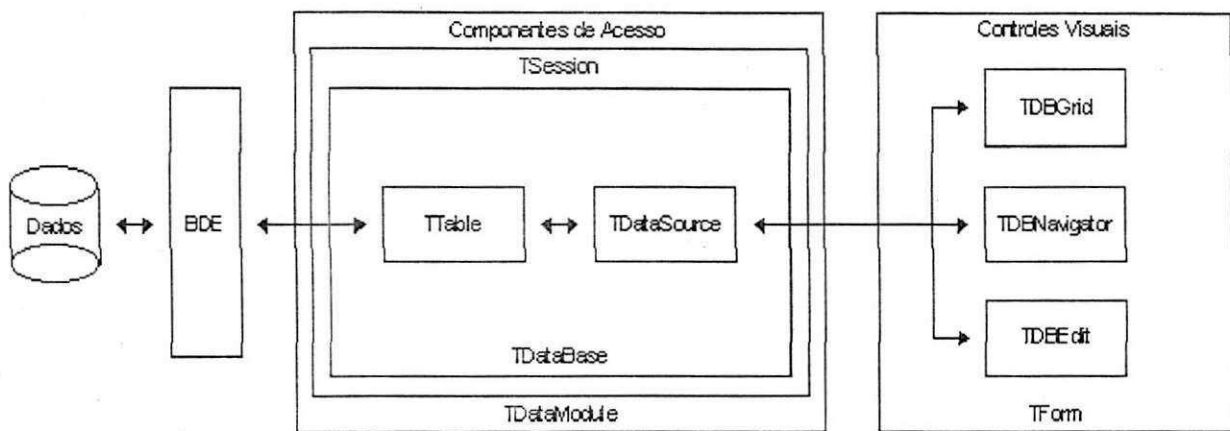
Arquitetura de Acesso

O acesso e manipulação de um banco de dados por um programa Delphi é realizado como mostrado abaixo, note que a aplicação não acessa os dados diretamente, mas usa sempre o BDE. O Borland Database Engine(BDE) foi utilizado no Sistema SAP (Sistema de Automação de Postos de Gasolina) para conexão direta com o banco de dados Interbase.

O BDE fornece a capacidade de acesso padronizado a banco de dados para Delphi, C++ Builder e outros ambientes de programação da Borland, oferecendo um grande conjunto de funções para auxiliar no desenvolvimento de aplicações Desktop e Cliente/Servidor.

Os controladores da BDE podem ser usados para acessar bases de dados dBase, Paradox, Access, FoxPro, Interbase, Oracle, Sybase e MS-SQL Server, DB2, Informix, além de um controlador de acesso a arquivos texto. Você também pode utilizar fontes de dados ODBC, podendo acessar qualquer base de dados compatível.

As funções que compõe uma API da BDE são usadas internamente pelos componentes de acesso a dados do Delphi e muito raramente você teria que usá-las diretamente, mas isso é totalmente possível. A referência completa das funções da BDE, com exemplos em Delphi, está no BDE API Help na pasta do Delphi no Menu Iniciar.

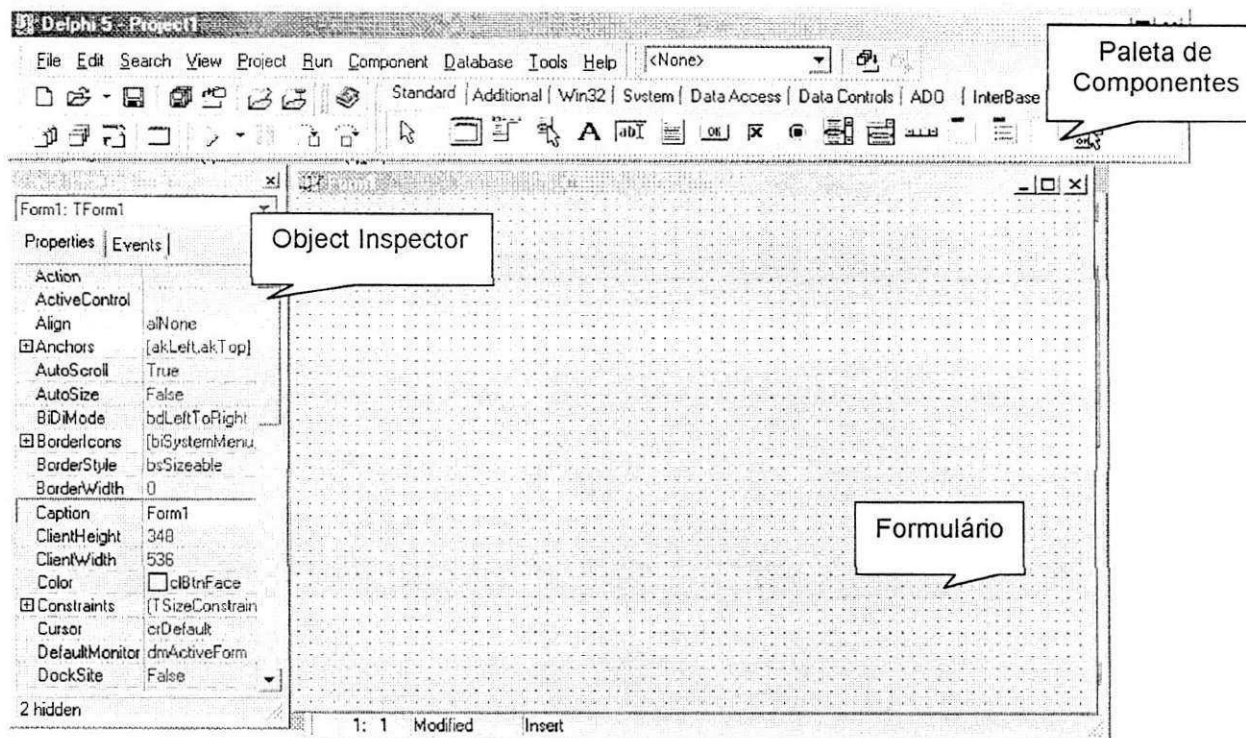


O IDE (Integrated Development Environment) do Delphi 5

Quando ativamos o Delphi, a tela inicial é parecida com a figura abaixo. Na janela superior, temos a barra do menu principal do Delphi, à esquerda a SpeedBar, com as opções mais comuns e à direita a paleta de componentes. Estes componentes são a base da programação visual e é onde o *designer* vai buscar recursos para sua aplicação.

Abaixo da SpeedBar, está a janela do *Object Inspector*, que permite visualizar e modificar as propriedades e eventos de todos os

componentes. É também largamente utilizado pelo *designer*. Abaixo da paleta ficam a janela de código-fonte e as janelas que estão sendo construídas



Formulário : É a janela com características do padrão Windows da Microsoft, onde o programador constrói sua janela utilizando os componentes que o Delphi disponibiliza.

Paleta de Componentes : Possui todos os componentes que o Delphi oferece organizados em paletas de acordo com o estilo, além daqueles componentes criados pelo próprio programador. Simples de usar, o usuário pode clicar em cima do componente e clicar no formulário adicionando-o.

Object Inspector : Com esse utilitário, o programador pode modificar as propriedades e eventos inerentes aos componentes colocados no formulário.

O IDE do Delphi 5 inclui algumas das mudanças mais amplas que a Borland introduziu, desde que migrou do Delphi 1 para o Delphi 2. Dentre os novos recursos estão um Object Inspector de projeto novo, um novo Project Manager, a capacidade de salvar a posição das janelas da área de trabalho, etc.

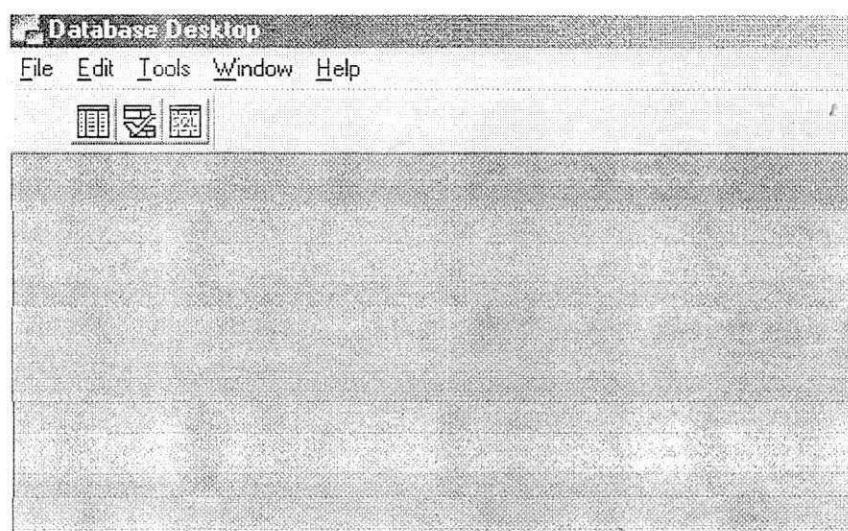
O Delphi oferece dois níveis de programação distintos. Existe o nível do que o manual chama de *designer*, que se utiliza dos recursos de

programação visual e aproveita componentes prontos, e o nível do *component writer*, que escreve os componentes para o *designer* utilizar nas aplicações. Podemos dizer que o *component writer* programa em um nível mais baixo e o *designer* em um nível mais alto.

Ferramentas que acompanham o Delphi 5

DataBase Desktop

O *Database Desktop* possibilita a manipulação de tabelas DB e DBF, que são utilizadas em bancos de dados Paradox. A versão anterior do sistema SAP, projetada em Clipper, utilizava esses tipos de tabelas, o que tornou essa ferramenta de grande importância para a manipulação dos dados do sistema antigo.



SQL Explorer

O *SQL Explorer* permite que sejam testadas consultas SQL em um determinado Banco de Dados. Assim sendo, foi de grande ajuda nos testes de consulta SQL durante o desenvolvimento do Sistema, em tempo de projeto. Facilitando na inserção de queries nos componentes de acesso a banco de dados do Delphi.

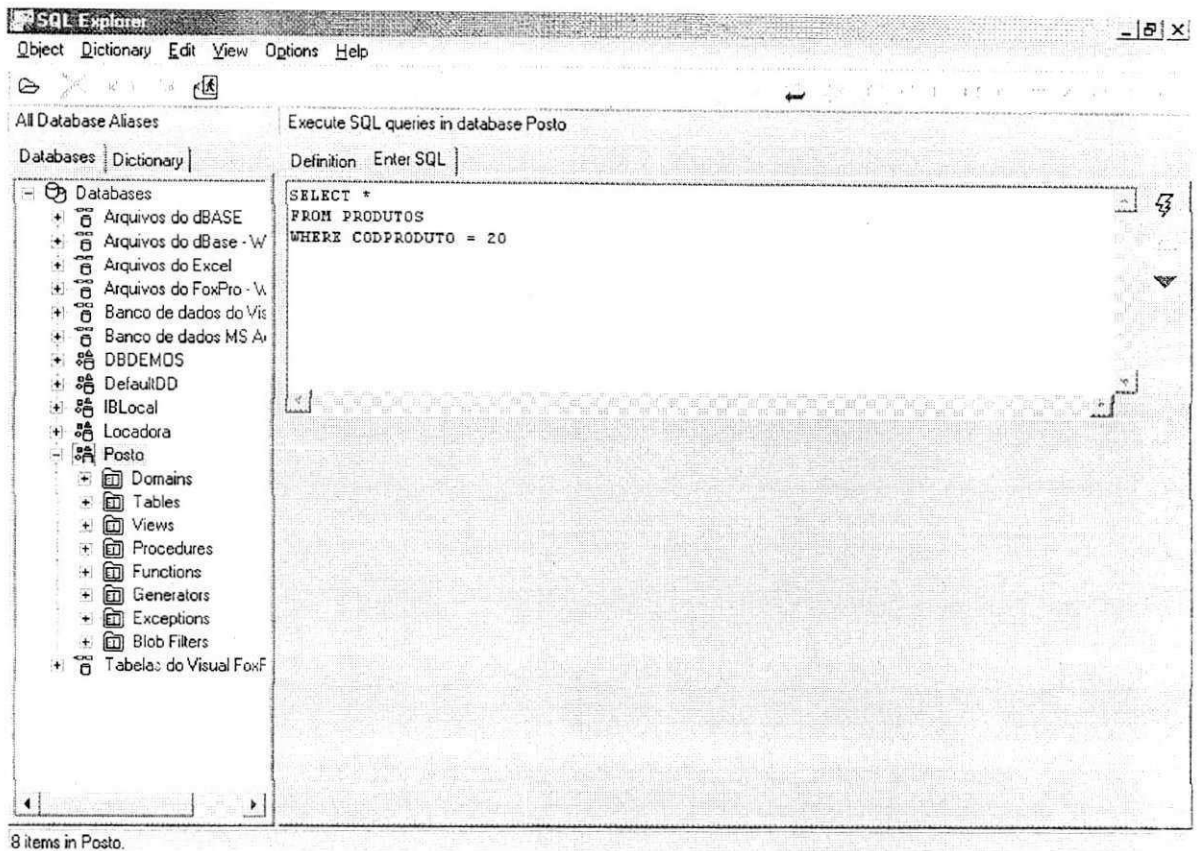
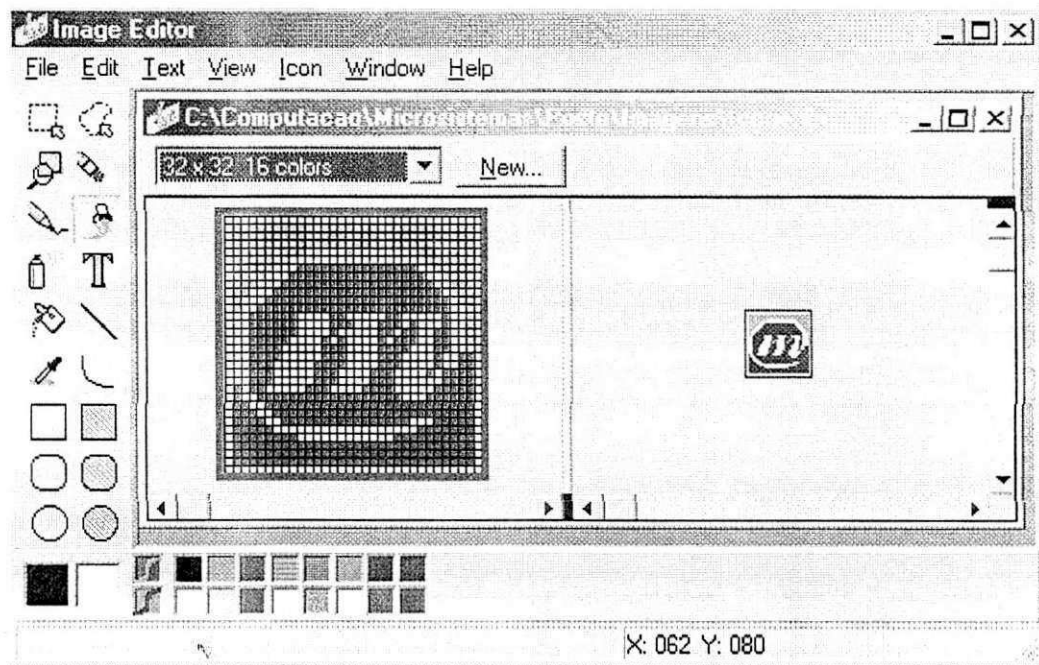


Image Editor

O *Image Editor* ajudou na construção do ícone que representa a logomarca da empresa Microsistemas Informática.



Ferramenta de Armazenamento

Interbase 6.0

O INTERBASE é um Sistema Gerenciador de Banco de Dados (SGBD) relacional que provê controle de transações e compartilhamento de dados em um ambiente mono-usuário ou multi-usuário.

É oferecido de duas formas, Interbase local (local Interbase, para usuário local) e Servidor Interbase (Interbase Server, para acesso remoto em arquitetura multi-usuário).

Algumas características oferecidas pelo Interbase estão listadas abaixo e demonstram que ele é um poderoso SGBD relacional.

- Compatível com SQL-92;
- Acesso simultâneo e múltiplos bancos de dados;
- Otimização automática, pelo servidor, de consultas (também podem ser efetuadas pelo usuário);
- Integridade Referencial Declarativa (através de chaves primárias e estrangeiras);
- Gatilhos (Triggers) e Visões;
- Windows ISQL (Interactive SQL) ferramenta para definição e consulta a dados;

Server manager: ferramenta para manutenção, restauração, backup e controle de segurança dos dados.

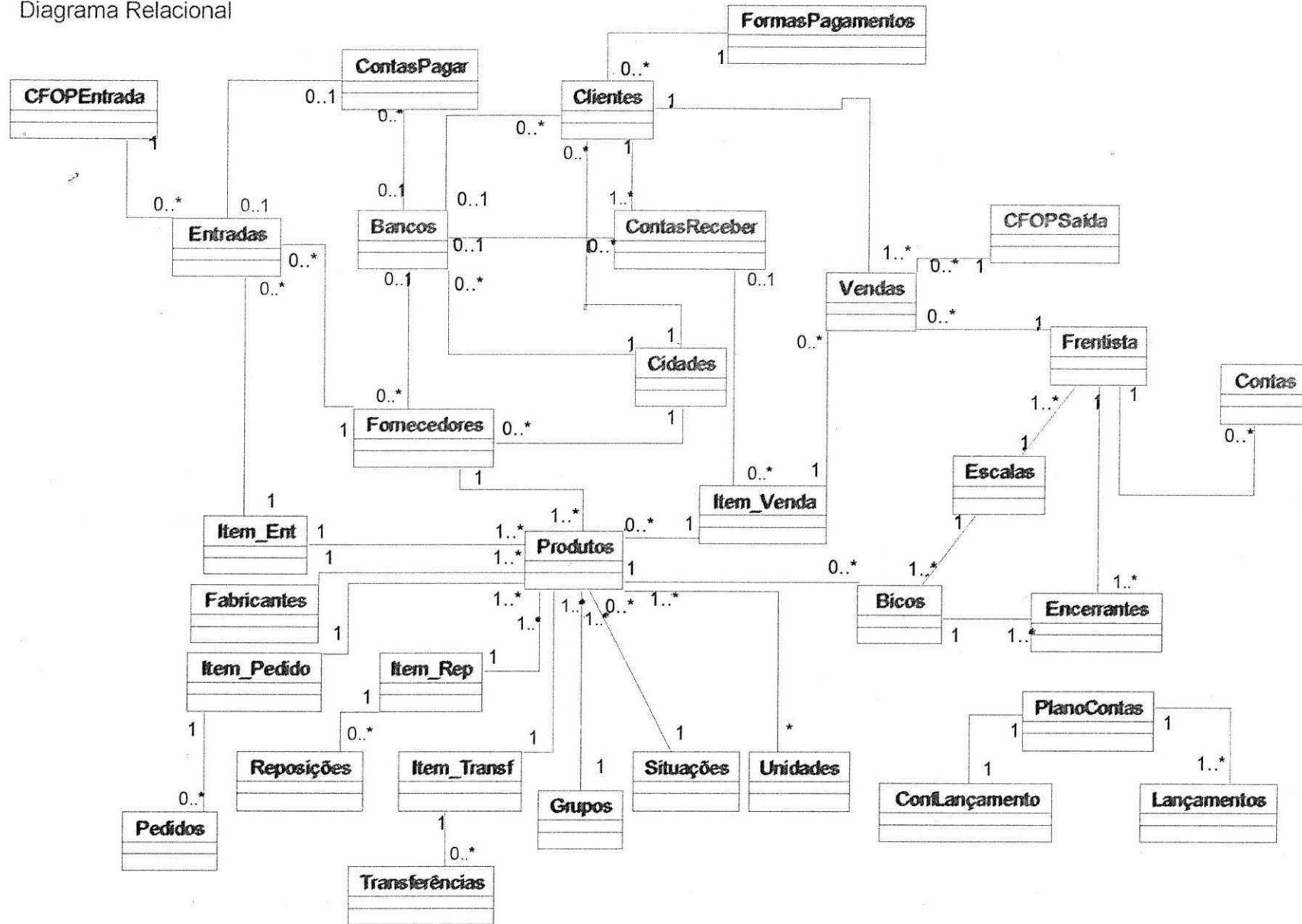
Modelagem do Sistema SAP usando Análise Essencial

Requisitos do Sistema

Funcionalidades	Descrição
Cadastrar Entidades	Cada entidade deve ser cadastrada em uma tela individual, onde deve haver as seguintes opções: Inclusão, Alteração, Exclusão e Consulta.
Consultar Entidades	A consulta deve ser realizada de forma rápida e deve ser feita pelos campos mais importantes de cada uma.
Efetuar pagamento	Cada pagamento deve ser feito através da consulta por título. Devendo existir o status da conta afirmando se está paga ou não.
Efetuar recebimento	Para receber do cliente deve haver a opção de consultar todas as contas do cliente, bem como uma conta individual.
Gerar fatura	A geração de fatura deve ser feita diariamente por data, acionada pelo usuário. E também pode ser feita por cliente, para aqueles que querem pagar antes de sua data de fatura..
Vender	A venda deve ter a opção de incluir vários e diferentes produtos na mesma tela. Não deve permitir venda de frentistas não escalados.
Comprar	A compra deve ser feita com a inclusão de diferentes produtos na mesma tela. Deve haver a geração de fatura no momento da inclusão de uma nova entrada com as informações dos títulos.
Escalar frentistas	A escala deve ser feita para cada bico cadastrado. Será feita semanalmente de domingo a domingo. Devendo haver a escolha do turno e do frentista que será escalado para o bico.
Encerrar turno	Deve ser feito para cada turno individualmente. Havendo as seguintes opções: Abertura (onde cada bico será aberto na mesma tela, baseado na escala dos frentistas), Fechamento (idem a abertura, fornecendo o valor final marcado pela bomba de gasolina) e Manutenção para possíveis alterações.
Prestação de contas	As vendas dos frentistas devem ser consultadas na data e no turno informados. Devem ser mostrados os valores das vendas (somatórios) de cada forma de pagamento, para que o frentista preste conta dos valores que possui em mãos.
Repor produto	O estoque do depósito deve ser atualizado com as quantidades que forem colocadas em outro estoque, seja pista, restaurante, hotel ou bar.

Transferir produto	O estoque do depósito deve ser atualizado quando um produto sai de um posto para outro. Devendo ser informado o local de destino do ou dos produtos que deve ser registrados na mesma tela.
Emitir pedido	Todos os pedidos aos fornecedores devem ser cadastrados juntamente com sua data de pedido e uma possível data de entrega. Todos os itens devem ser incluídos na mesma tela.
Cadastro de usuário	Devem ser dadas permissões de utilização do sistema aos usuário de acordo com sua função na empresa.
Cadastro de clientes	Deve haver a opção de bloquear e desbloquear cliente, permitindo ou não a venda ao mesmo. Deve haver também a opção de ativar e desativar um cliente que deixou de comprar no posto com a possibilidade de voltar a comprar. Evitando o recadastramento do mesmo.
Cadastro de produtos	Deve haver uma opção para visualização de sua quantidade nos diferentes estoques.
Login	O usuário deve se logar no sistema com um login e uma senha previamente cadastrados.

Diagrama Relacional



Tabelas Relacionais Geradas a Partir do DER

- **Tabela: Bancos**
- **Descrição:** Cadastro dos bancos, nos quais os clientes e o posto têm contas.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODBANCO	Código do banco auto-incrementado	INTEIRO		NAO NULO
NOMEBANCO	Nome do Banco	CHARACTER	30	
RUA	Endereço	CHARACTER	40	
NUMERO	Endereço	INTEIRO		
BAIRRO	Endereço	CHARACTER	10	
GERENTE	Gerente do Banco	CHARACTER	20	
CIDADE	Endereço	INTEIRO		
FONE_R	Telefone	CHARACTER	13	
F_A_X	Fax	CHARACTER	13	

Chave Primária: CODBANCO

Chaves Estrangeiras: CIDADE

- A tabela BANCOS se relaciona com a tabela CIDADES através de CIDADE

- **Tabela: Bicos**
- **Descrição:** Cadastro das bombas (bicos) que contêm algum tipo de combustível como produto.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODBICO	Código do bico auto-incrementado	INTEIRO		NÃO NULO
PRODUTO	Combustível que sai do bico da bomba	INTEIRO		NÃO NULO
LEITURA	Leitura inicial da bomba	REAL	15:2	

Chave Primária: CODBICO

Chaves Estrangeiras: PRODUTO

- A tabela BICOS se relaciona com a tabela PRODUTOS através de PRODUTO

- **Tabela: ConfLancamento**
- **Descrição:** Cadastro dos lançamentos financeiros feitos no posto. É uma tabela fundamental p/ questões de contabilidade do posto, pois armazena dados relacionados à todos os locais de onde o patrimônio do posto está investido. Como por exemplo, dinheiro, contas correntes, fiado, estoque, e também as principais despesas, como água, fornecedores e até mesmo despesas com softwares.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODCONF	Campo com auto-incremento que indica o código de configuração	INTEGER		NAO NULO
DESCRICA0	Nome do plano de contas (dinheiro, contas correntes, estoque, fiado, etc)	CHARACTER	30	NÃO NULO
CONTA	Código do plano de contas (patrimônio, como dinheiro, conta corrente, etc) do posto.	CHARACTER	4	

Chave Primária: CODCONF

- **Tabela: CFOPEntrada**

- **Descrição:** Cadastro dos códigos fiscais de entradas de produto. Esses códigos são fixados por lei.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CFOPE	Código da classificação fiscal determinado por lei	CHARACTER	3	NÃO NULO
NATUREZA	Classificação fiscal da Compra	CHARACTER	70	NÃO NULO

Chave Primária: CFOPE

- **Tabela: CFOPSaída**
- **Descrição:** Cadastro dos códigos fiscais de saída de produto. Esses códigos são fixados por lei.
-

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CFOPS	Código da classificação fiscal determinado por lei	CHARACTER	3	NÃO NULO
NATUREZA	Classificação fiscal da Compra	CHARACTER	70	NÃO NULO

Chave Primária: CFOPS

- **Tabela: Cidades**
- **Descrição:** Cadastro das principais cidades do país.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODCIDADE	Código da cidade auto-incrementado	INTEIRO		NÃO NULO
CIDADE	Nome da Cidade	CHARACTER	40	NÃO NULO
ESTADO	Estado onde está a cidade	CHARACTER	2	NÃO NULO
CEP	CEP da cidade	CHARACTER	8	

Chave Primária: CODCIDADE

- **Tabela: Clientes**
- **Descrição:** Cadastro dos clientes do posto. Visa manter o controle, sobretudo, daqueles clientes com alguma das formas de pagamento que não são à vista. Clientes que pagam à vista são cadastrados como consumidor.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODCLIENTE	Código do clienetauto-incrementado	INTEIRO		NÃO NULO
RAZAOSOCIAL	Razão social ou nome do cliente pessoa física	CHARACTER	40	NÃO NULO
FANTASIA	Nome fantasia do cliente pessoa jurídica	CHARACTER	2	NÃO NULO
RUA	Endereço	CHARACTER	8	
NUMERO	Endereço	INTEIRO		
COMPLEMENTO	Endereço	CHARACTER	20	
BAIRRO	Endereço	CHARACTER	20	
CIDADE	Endereço	INTEIRO		
CEP	Endereço	CHARACTER	9	
CONTATO	Pessoa para contato	CHARACTER	20	
CGC	CGC do cliente	CHARACTER	14	
INSCRI	Número de inscrição do cliente pessoa jurídica	CHARACTER	20	
CAE		CHARACTER	10	
EMAIL	E-mail do cliente	CHARACTER	30	
PAGAMENTO	Forma de pagamento escolhida pelo cliente	INTEIRO		NÃO NULO
DATA_CADASTRO	Data do cadastro	DATA		
DATA_NASC	Data de nascimento	DATA		
CREDITO	Valor de um possível crédito do cliente	REAL	15:2	
DEBITO	Valor de um possível débito do cliente	REAL	15:2	
POS_REFERENCIA	Local para contato	CHARACTER	60	
DESCONTO	Valor de um possível desconto dado ao cliente	REAL	9:2	
STATUS	Ativo ou Inativo (evita excluir clientes)	CHARACTER	1	
BANCO	Banco dos cheques do cliente	INTEIRO		
FONE_R	Telefone Residencial	CHARACTER	13	
FONE_C	Telefone Comercial	CHARACTER	13	

F_A_X	FAX	CARACTER	13	
OBSERVAÇÃO	Alguma observação necessária	CARACTER	120	
CLIENTESTATUS	Bloqueado ou liberado para comprar	CARACTER	1	

Chave Primária: CODCLIENTE

Chaves Estrangeiras: CIDADE, BANCO

- A tabela CLIENTES se relaciona com a tabela CIDADES através de CIDADE
- A tabela CLIENTES se relaciona com a tabela BANCOS através de BANCO

- **Tabela: Contas**
- **Descrição:** Cadastro das quantias em cheques, dinheiro, cartão ou fiado entregues por cada frentista a cada final de escala (trabalho).

Atributo	Descrição	Tipo	Tam.	Restrição
(*)NUMEROCONTA	Código da conta auto-incrementado	INTEIRO		NÃO-NULO
DT_CON	Data da prestação de contas pelo frentista	DATA		NÃO NULO
FRENTISTA	Frentista que prestou contas de suas vendas	INTEIRO		NÃO NULO
TURNO	Turno da prestação de contas {M, N, T}	CARACTER	1	
DEB_DIN	Valor debitado em dinheiro	REAL	15:2	
DEB_CHE	Valor debitado em cheque	REAL	15:2	
DEB_C_P	Valor debitado em cheque pré-datado	REAL	15:2	
DEB_CAR	Valor debitado em cartão	REAL	15:2	
DEB_FIA	Valor debitado a prazo	REAL	15:2	
CRE_DIN	Valor creditado em dinheiro	REAL	15:2	
CRE_CHE	Valor creditado em cheque	REAL	15:2	
CRE_C_P	Valor creditado em cheque pré-datado	REAL	15:2	
CRE_CAR	Valor creditado em cartão	REAL	15:2	
CRE_FIA	Valor creditado a prazo	REAL	15:2	
TOT_CRE	Valor total creditado	REAL	15:2	
TOT_DEB	Valor total debitado	REAL	15:2	
DIFEREN	Valor total debitado menos valor total creditado	REAL	15:2	

Chave Primária: NUMEROCONTA

Chaves Estrangeiras: FRENTISTA

- A tabela CONTAS se relaciona com a tabela FRENTISTA através de FRENTISTA

- **Tabela: ContasPagar**
- **Descrição:** Cadastro de todas as contas a pagar do posto, quer sejam aos fornecedores, como também despesas com água, luz, alguma compra extra, etc

Atributo	Descrição	Tipo	Tam.	Restrição
(*)CODCONTAS	Código das contas a pagar auto-incrementado	INTEIRO		NÃO NULO
BANCO	Cód. Banco	INTEIRO		
DT_FAT	Data de fatura	DATA		
DT_EMI	Data de emissão	DATA		
DT_VEN	Data da venda	DATA		
DT_PGT	Data do pagamento	DATA		
VALORT	Valor do título	REAL	15:2	
VALORF	Valor da fatura	REAL	15:2	
VJUROS	Juros no pagamento	REAL	15:2	
VMULTA	Multa no pagamento	REAL	15:2	

VDESCONTO	Desconto no pagamento	REAL	15:2	
VALORP	Valor pago	REAL	15:2	
OBSERV	Observação sobre o título	CHARACTER	59	
PG_DIN	Valor pago em dinheiro	REAL	15:2	
PG_CHQ	Valor pago em cheque	REAL	15:2	
ENTRADA	Código da entrada	INTEIRO		
TITULO	Número do título	CHARACTER	12	NÃO NULO
STATUS	Paga ou não paga	CHARACTER	1	

Chave Primária: CODCONTAS

Chaves Estrangeiras: BANCO, ENTRADA, FORNECEDOR

- A tabela CONTASPAGAR se relaciona com a tabela BANCOS através de BANCO
- A tabela CONTASPAGAR se relaciona com a tabela ENTRADAS através de ENTRADA

- **Tabela: ContasReceber**
- **Descrição:** Cadastro de todas as contas a receber do posto. Essas contas podem ser provenientes de vendas à prazo aos clientes, bem como, alguma venda extra de algum bem do dono do posto.

Atributo	Descrição	Tipo	Tam.	Restrição
(*)CODCONTASRECE	Código das contas a receber auto-incrementado	INTEIRO		NÃO NULO
CLIENTE	Cód. Cliente	INTEIRO		
BANCO	Cód. Banco	INTEIRO		
DT_FAT	Data de fatura	DATA		
DT_EMI	Data de emissão	DATA		
DT_VEN	Data da venda	DATA		
DT_PGT	Data do pagamento	DATA		
VALORT	Valor do título	REAL	15:3	
VALORF	Valor da fatura	REAL	15:3	
VJUROS	Valor dos juros	REAL	15:3	
VMULTA	Valor da multa	REAL	15:3	
VDESCONTO	Valor do desconto	REAL	15:3	
VALORP	Valor pago	REAL	15:3	
VALORR	Valor recebido	REAL	15:3	
OBSERV	Observação	CHARACTER	59	
PG_DIN	Valor pago em dinheiro	REAL	15:3	
PG_CHQ	Valor pago em cheque	REAL	15:3	
PG_CHP	Valor pago em cheque predatado	REAL	15:3	
TITULO	Número do título	INTEIRO		NÃO-NULO
STATUS	Paga ou não paga	CHARACTER	1	

Chave Primária: CODCONTASRECE

Chaves Estrangeiras: BANCO, CLIENTE

- A tabela CONTASRECEBER se relaciona com a tabela BANCOS através de BANCO
- A tabela CONTASRECEBER se relaciona com a tabela CLIENTES através de CLIENTE

- **Tabela: Encerrantes**
- **Descrição:** Cadastro dos frentistas que encerram cada turno diariamente e em cada bico (bomba de combustível). É uma espécie de histórico por turno.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODENCE	Código dos encerrantes (frentista que finaliza um turno) auto-incrementado	INTEIRO		NÃO NULO
DATAENCE	Data da encerrante	DATA		NÃO NULO
TURNO	Turno da encerrante {M, T, N}	CHARACTER	1	NÃO NULO
FRENTISTA	Frentista que encerrou turno	INTEIRO		NÃO NULO
BICO	Bico da encerrante	INTEIRO		NÃO NULO
INICIO	Valor inicial registrado na bomba	REAL	15:1	
AFERICAO	Valor da aferição	INTEIRO		
TERMINO	Valor final registrado na bomba	REAL	15:1	
VENDAS	Quantia em vendas	REAL	9:1	
UNITAR	Valor unitário	REAL	15:4	
TOTALV	Total das vendas	REAL	15:2	

Chave Primária: CODENCE

Chaves Estrangeiras: BICO, FRENTISTA

- A tabela ENCERRANTES se relaciona com a tabela BICOS através de BICO
- A tabela ENCERRANTES se relaciona com a tabela FRENTISTA através de FRENTISTA

- **Tabela: Entradas**
- **Descrição:** Cadastro das entradas de produtos no posto.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODENTRADA	Código da entrada auto-incrementado Chave primária auto-incrementada	INTEIRO		NÃO NULO
PEDIDO	Número do pedido	INTEIRO		
FATURA	Número da fatura	INTEIRO		NÃO NULO
FORNECEDOR	Cód. fornecedor	INTEIRO		NÃO NULO
DT_EMI	Data de emissão	DATA		
DT_ENT	Data da entrada	DATA		
SERIE	Número de série	CHARACTER	3	
VCONTA	Valor contábil	REAL	15:2	
CCONTA	Custo contábil	INTEIRO		
OBSERV	Observação	CHARACTER	10	
V_BASE	Valor base	REAL	15:2	
ALIUO	Valor da alíquota	REAL	9:2	
ICMS_N	ICM normal	REAL	15:2	
ICMS_E	ICM por entrada	REAL	15:2	
ICMS_F	ICM na fonte	REAL	15:2	
BASE_S	Base/Substituição	REAL	15:2	
ICMS_S	ICM Substituição	REAL	15:2	
ISENTA	Valor Isentas	REAL	15:2	
OUTRAS	Outras isentas	REAL	15:2	
I_P_I	IPI	REAL	15:2	
VALORP	Valor do produto	REAL	15:2	
VALORF	Valor da fatura	REAL	15:2	
OUTROS	Outros valores	REAL	15:2	
GERAR_FIN	Gera ou não financeiro (contas a pagar) {S, N}	CHARACTER	1	
CFOP	Classificação fiscal	CHARACTER	3	NÃO NULO
MODELO	Modelo	CHARACTER	2	
LOCALDESPACHO	Estoque de despacho do produto {P,H,B,L,R,D}	CHARACTER	1	

Chave Primária: CODENTRADA

Chaves Estrangeiras: CFOP, FORNECEDOR

- A tabela ENTRADAS se relaciona com a tabela CFOPENTRADA através de CFOP
- A tabela ENTRADAS se relaciona com a tabela FORNECEDORES através de FORNECEDOR

- **Tabela: Escalas**
- **Descrição:** Cadastro das escalas de frentistas que vão trabalhar nos respectivos bicos (bombas), em um dado turno e durante um determinado período de dias.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODESCALA	Código da escala auto-	INTEIRO		NÃO NULO
DATAINI	Data inicial	DATA		NÃO NULO
DATAFIM	Data final	DATA		NÃO NULO
TURNO	Turno da escala {M, T, N}	CARACTER	1	
FRENTISTA	Cód. Frentista	INTEIRO		NÃO NULO
BICO	Cód. Bicos	INTEIRO		NÃO NULO

Chave Primária: CODESCALA

Chaves Estrangeiras: BICO, FRENTISTA

- A tabela ESCALAS se relaciona com a tabela BICOS através de BICO
- A tabela ESCALAS se relaciona com a tabela FRENTISTA através de FRENTISTA

- **Tabela: Fabricante**
- **Descrição:** Cadastro dos fabricantes de produtos do posto.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODFABRICANTE	Chave primária auto-incrementada	INTEIRO		NÃO NULO
NOMEFABRICANTE	Nome do fabricante	CARACTER	30	NÃO NULO
ABREVIACAO	Como é conhecido o fabricante	CARACTER	15	NÃO NULO

Chave Primária: CODFABRICANTE

- **Tabela: FormasPagamento**
- **Descrição:** Cadastro das formas de pagamento permitidas aos clientes do posto. Além das formas convencionais (cheques, cartão), existe facilidades de pagamento por nota (fiado com o número de dias que o cliente desejar), mensal e quinzenal.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODFORMA	Código da forma de pagamento auto-incrementado	INTEIRO		NÃO NULO
FORMA	Forma de pagamento {mensal, semanal, quinzenal, nota}	CARACTER	10	NÃO NULO
DATA1DOMES	Primeira quinzena {1..15}	CARACTER	2	
DATA2DOMES	Segunda quinzena {16..28}	CARACTER	2	
DATA	Data do mês {1..28}	CARACTER	2	
DIADASEMANA	Dia da semana {1..7}	INTEIRO		
DÍAS	Número de dias {1..99}	CARACTER	2	
TIPO	FORMA + {DATA, DIADASEMANA, DATA1DOMES,..}	CARACTER	30	

Chave Primária: CODFORMA

- **Tabela: Fornecedores**
- **Descrição:** Cadastro dos fornecedores de produtos.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODFORNECEDOR	Código do fornecedor auto-incrementado	INTEIRO		NÃO NULO
RAZAOSOCIAL	Razão Social do Fornecedor	CHARACTER	50	NÃO NULO
ABREVIACAO	Como é conhecido o fornecedor	CHARACTER	15	NÃO NULO
RUA	Endereço	CHARACTER	40	
NUMERO	Endereço	INTEIRO		
COMPLEMENTO	Endereço	CHARACTER	20	
BAIRRO	Endereço	CHARACTER	20	
CEP	Endereço	CHARACTER	9	
CONTATO	Pessoa para contato	CHARACTER	15	
CGC	CGC do fornecedor	CHARACTER	14	
INSCRICAO	Número de inscrição	CHARACTER	14	
EMAIL	E-mail do fornecedor	CHARACTER	30	
CONTABIL	Valor contábil	CHARACTER	4	
AGENCIA	Número da agência	CHARACTER	5	
CONTA	Numero da conta	CHARACTER	10	
REPRESENTANTE	Nome do representante	CHARACTER	40	
RUA_REP	Endereço do representante	CHARACTER	40	
NUMERO_REP	Endereço do representante	INTEIRO		
BAIRRO_REP	Endereço do representante	CHARACTER	20	
CIDADE_REP	Endereço do representante	INTEIRO		
CEP_REP	Endereço do representante	CHARACTER	9	
CIDADE	Endereço	INTEIRO		
BANCO	Cód. Banco	INTEIRO		
FONE_R	Telefone residencial do representante	CHARACTER	13	
FONE_C	Telefone comercial do representante	CHARACTER	13	
F_A_X	Número de FAX do representante	CHARACTER	13	

Chave Primária: CODFORNECEDOR

Chaves Estrangeiras: BANCO, CIDADE

- A tabela FORNECEDORES se relaciona com a tabela BANCOS através de BANCO
- A tabela FORNECEDORES se relaciona com a tabela CIDADES através de CIDADE

- **Tabela: Frentista**
- **Descrição:** Cadastro dos frentistas que estão empregados até o presente momento ou que já trabalharam no posto.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODFRENTISTA	Código do frentista auto-incrementado	INTEIRO		NÃO NULO
NOMEABREVIADO	Nome abreviado do frentista	CHARACTER	15	
NOME	Nome completo do frentista	CHARACTER	30	NÃO NULO
STATUS	Indica se o frentista está disponível ou não.	CHARACTER	1	

	(manter o controle dos frentistas que estão empregados no posto no momento)			
--	---	--	--	--

Chave Primária: CODFRENTISTA

- **Tabela: Grupos**
- **Descrição:** Cadastro dos grupos de produtos do posto. Exigência do cliente devido a diversidade de produtos do posto. Existem várias categorias de produtos, tendo em vista que também há a parte do bar, restaurante e hotel.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODGRUPO	Código do grupo auto-incrementado	INTEIRO		NÃO NULO
NOMEGRUPO	Nome do grupo	CHARACTER	25	NÃO NULO
PRECO	Máscara de preço	CHARACTER	10	NÃO NULO
QUANTIDADE	Máscara de quantidade	CHARACTER	8	NÃO NULO

Chave Primária: CODGRUPO

- **Tabela: Item_Ent**
- **Descrição:** Cadastro dos itens de produto feitos a cada entrada.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) ENTRADA	Cód. Entrada	INTEIRO		NÃO NULO
(*) PRODUTO	Cód. Produto	INTEIRO		NÃO NULO
QUANTIDADE	Quantidade de produtos	REAL	15:3	NÃO NULO
ETIQUETA	Etiqueta do produto	REAL	15:2	
CUSTOU	Custo unitário do produto	REAL	15:3	NÃO NULO
VALORT	Valor total	REAL	15:3	NÃO NULO
VALORC	Valor contábil	REAL	15:3	NÃO NULO
PC_IPI	IPI	REAL	9:2	
PC_ICM	ICM	REAL	9:2	
PC_SUB	Substituição	REAL	9:2	

Chave Primária: ENTRADA, PRODUTO

Chaves Estrangeiras: ENTRADA, PRODUTO

- A tabela ITEM_ENT se relaciona com a tabela ENTRADAS através de ENTRADA
- A tabela ITEM_ENT se relaciona com a tabela PRODUTOS através de PRODUTO

- **Tabela: Item_Venda**
- **Descrição:** Cadastro dos itens de produto feitos a cada venda.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) VENDA	Cód. Venda	INTEIRO		NÃO NULO
(*) PRODUTO	Cód. Produto	INTEIRO		NÃO NULO
QUANTIDADE	Quantidade do produto	REAL	8:2	NÃO NULO
PRECOV	Preço de venda do produto	REAL	15:4	NÃO NULO
TOTALV	Valor total da venda	REAL	15:2	NÃO NULO
CONTARECEBE	Cód. Conta a Receber	INTEIRO		

Chave Primária: PRODUTO, VENDA

Chaves Estrangeiras: PRODUTO, VENDA, CONTARECEBE

- A tabela ITEM_VENDA se relaciona com a tabela VENDAS através de VENDA
- A tabela ITEM_VENDA se relaciona com a tabela PRODUTOS através de PRODUTO
- A tabela ITEM_VENDA se relaciona com a tabela CONTASRECEBER através de CONTARECEBE

- **Tabela: Lançamentos**
- **Descrição:** Cadastro dos lançamentos financeiros (crédito ou débito) feitos no posto.

Atributo	Descrição	Tipo	Tam	Restrição
(*) CODLANCA	Campo auto-incrementado que indica o código de lançamento	INTEIRO		NÃO NULO
CONTA	Código do plano de contas de onde será debitado ou creditado uma quantia.	CHARACTER	40	NÃO NULO
DATA	Data do saque ou depósito	DATA		
TIPO	C (crédito) ou D (débito)	CHARACTER	1	NÃO NULO
HISTORICO	Destino da quantia	CHARACTER	60	
SALDO	Quantia disponível no plano de contas	NUMERICO	15:2	

Chave Primária: CODLANCA

Chaves Estrangeiras: CONTA

A tabela Lançamentos se relaciona com a tabela PlanoContas através de CONTA.

- **Tabela: Produtos**
- **Descrição:** Cadastro dos produtos do posto.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODPRODUTO	Código do produto auto-incrementado	INTEIRO		NÃO NULO
GRUPO	Cód. Grupo	INTEIRO		NÃO NULO
FORNECEDOR	Cód. Fornecedor	INTEIRO		
FABRICANTE	Cód. Fabricante	INTEIRO		
CODEAN	Código de barras	INTEIRO		
DISCRIMINACAO	Discriminação do produto	CHARACTER	40	NÃO NULO
UNIDENT	Cód, unidade de entrada	INTEIRO		NÃO NULO
DISCABREVIADA	Discriminação abreviada do produto	CHARACTER	29	NÃO NULO
QUANTENT	Quantidade de entrada	REAL	9:3	NÃO NULO
UNIDVENDA	Cód, unidade de venda	INTEIRO		NÃO NULO
QUANTVENDA	Quantidade de venda	REAL	9:3	NÃO NULO
CUSTO_E	Custo da embalagem	REAL	15:4	
CUSTO_U	Custo unitário	REAL	15:4	
CUSTO_C	Custo contábil	REAL	15:4	
CUSTO_L	Custo líquido	REAL	15:4	
CUSTO_M	Custo médio	REAL	15:4	
PRECO_V	Preço de venda	REAL	15:4	NÃO NULO
PRECO_P	Preço promocional	REAL	15:4	NÃO NULO
SITTRIB	Cód. Situação tributária	INTEIRO		NÃO NULO
ICMINT	ICM interno	REAL	9:2	
ICMEXT	ICM externo	REAL	9:2	

ICMSUB	ICM de substituição	REAL	9:2	
REDINT	Redução interna	REAL	9:2	
REDEXT	Redução externa	REAL	9:2	
IPI	IPI	REAL	9:2	
PESO_BR	Peso bruto	REAL	9:3	
PESO_LQ	Peso líquido	REAL	9:3	
DATAPRECO	Data da última mudança de preço	DATA		
DATACADASTRO	Data de cadastro	DATA		
MAXIMOP	Quantidade máxima do produto exposto na pista	INTEIRO		
MAXIMOL	Quantidade máxima na lanchonete	INTEIRO		
MAXIMOH	Quantidade máxima no hotel	INTEIRO		
MAXIMOB	Quantidade máxima no bar	INTEIRO		
ESTOQUEL	Quantidade na lanchonete	REAL	15:3	
ESTOQUEP	Quantidade na pista	REAL	15:3	
ESTOQUEH	Quantidade no hotel			
ESTOQUED	Quantidade no depósito	REAL	15:3	
ESTOQUEB	Quantidade no bar	REAL	15:3	
TOTAL	Total em estoque	REAL	15:2	
DATA_ULT_S	Data da última saída	DATA		
QT_ULT_S	Quantidade da última saída	REAL	15:3	
DATA_ULT_E	Data da última entrada	DATA		
QT_ULT_E	Quantidade da última entrada	REAL	15:3	
CONTROLAESTOQUE	Venda altera estoque {S, N}	CARACTER	1	NÃO NULO

Chave Primária: CODPRODUTO

Chaves Estrangeiras: FABRICANTE, FORNECEDOR, GRUPO, UNIDENT, UNIDVENDA, SITTRIB

- A tabela PRODUTOS se relaciona com a tabela FABRICAMNTES através de FABRICANTE
- A tabela PRODUTOS se relaciona com a tabela FORNECEDORES através de FORNECEDOR
- A tabela PRODUTOS se relaciona com a tabela GRUPOS através de GRUPO
- A tabela PRODUTOS se relaciona com a tabela SITUACOES através de SITTRIB
- A tabela PRODUTOS se relaciona com a tabela UNIDADES através de UNIDENT, UNIDVENDA

- **Tabela: PlanoContas**
- **Descrição:** Cadastro dos planos de contas (patrimônios) do posto, com seus respectivos saldos disponíveis. Esses patrimônios podem ser em dinheiro, contas correntes, cheques, estoque e até mesmo as dívidas.

Atributo	Descrição	Tipo	Tam.	Restrição
(*)CONTACOD	Código do plano de contas do posto	CHARACTER	4	NÃO NULO
DESCRICAO	Nome do patrimônio, como dinheiro, conata, fiado	CHARACTER	30	
SALDOANT	O saldo anterior disponível no patrimônio	NUMERICO	15:2	
SALDOATUAL	O saldo atual disponível no patrimônio	NUMERICO	15:2	
QUANTIA	Valor do último crédito ou débito	NUMERICO	15:2	
TIPO	C (crédito) ou D (débito)	CHARACTER	1	

Chave Primária: CONTACOD

Chaves Estrangeiras: CONTACOD

A tabela PlanoContas se relaciona com a tabela ConfLancamento através de CONTACOD.

- **Tabela: Situações**
- **Descrição:** Cadastro das situações tributárias (ICM normal, isento, etc).

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODSITUACAO	Código das situações auto-incrementado	INTEIRO		NÃO NULO
SITUACAO	Nome da situação tributária	CHARACTER	16	NÃO NULO

Chave Primária: CODSITUACAO

- **Tabela: Unidades**
- **Descrição:** Cadastro das unidades de compra e venda de produtos.

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODUNIDADE	Código da unidade auto-incrementado	INTEIRO		NÃO NULO
ABREVIACAO	Abreviação da unidade	CHARACTER	3	NÃO NULO
NOMEUNIDADE	Nome da unidade	CHARACTER	10	NÃO NULO

Chave Primária: CODUNIDADE

- **Tabela: Vendas**
- **Descrição:** Cadastro das vendas realizadas no posto (parte da pista).

Atributo	Descrição	Tipo	Tam.	Restrição
(*) NUMERO	Código da venda auto-incrementado	INTEIRO		NÃO NULO
FRENTISTA	Cód. Frentista	INTEIRO		NÃO NULO
CLIENTE	Cód. Cliente	INTEIRO		NÃO NULO
DTVEN	Data da venda	DATA		NÃO NULO
DTFAT	Data da fatura	DATA		NÃO NULO
TURNO	Turno da venda {M, T, N}	CHARACTER	1	
TOTALV	Valor total da venda	REAL	15:2	
DESCP	Desconto	REAL	9:2	

LIQUID	Valor líquido	REAL	15:2	
PLACA	Placa do automóvel	CARACTER	10	
KM_ATU	Quilometragem atual do automóvel	INTEIRO		
CTRL_F	Gerou ou não a conta a receber {S, N}	CARACTER	1	
PAGAMENTO	Forma de pagamento {cheque, dinheiro, cartão, a prazo}	CARACTER	20	

Chave Primária: NUMERO

Chaves Estrangeiras: FRENTISTA, CLIENTE

- A tabela VENDAS se relaciona com a tabela FRENTISTAS através de FRENTISTA
- A tabela VENDAS se relaciona com a tabela CLIENTES através de CLIENTE

- Tabela: Item_Ped
- Descrição : Itens pedidos pelo posto aos fornecedores

Atributo	Descrição	Tipo	Tam.	Restrição
(*) PEDIDO	Cód. Pedido	INTEIRO		NÃO NULO
(*) PRODUTO	Cód. Produto	INTEIRO		NÃO NULO
QUANTIDADE	Quantidade de produtos	REAL	8:2	NÃO NULO
PRECO	Preço do produto	REAL	15:2	NÃO NULO
TOTAL	Valor total pedido	REAL	15:2	NÃO NULO

Chave Primária: PEDIDO+PRODUTO

Chaves Estrangeiras: PEDIDO, PRODUTO

- A tabela Item_Ped se relaciona com a tabela PEDIDOS através de PEDIDO
- A tabela Item_Ped se relaciona com a tabela PRODUTOS através de PRODUTO

- Tabela: Item_Rep
- Descrição : Itens que saem do depósito e são repostos nos demais estoques(pista, bar,...)

Atributo	Descrição	Tipo	Tam.	Restrição
(*) REPOSICAO	Cód. Reposição	INTEIRO		NÃO NULO
(*) PRODUTO	Cód. Produto	INTEIRO		NÃO NULO
QUANTIDADE	Quantidade do produto	REAL	8:2	NÃO NULO

Chave Primária: REPOSICAO+PRODUTO

Chaves Estrangeiras: REPOSICAO, PRODUTO

- A tabela Item_Rep se relaciona com a tabela REPOSICOES através de REPOSICAO
- A tabela Item_Rep se relaciona com a tabela PRODUTOS através de PRODUTO

- Tabela: Item_Transf
- Descrição : Itens transferidos de um posto para outro na rede de postos

Atributo	Descrição	Tipo	Tam.	Restrição
(*) TRANSF	Cód. Transferência	INTEIRO		NÃO NULO
(*) PRODUTO	Cód. Produto	INTEIRO		NÃO NULO
QUANTIDADE	Quantidade do produto	REAL	8:2	NÃO NULO

PRECO	Preço do produto	REAL	15:3	NÃO NULO
TOTAL	Valor total da transferência	REAL	15:2	NÃO NULO

Chave Primária: TRANSF+PRODUTO

Chaves Estrangeiras: TRANSF, PRODUTO

- A tabela Item_Transf se relaciona com a tabela TRANSFERENCIAS através de TRANSF
- A tabela Item_Transf se relaciona com a tabela PRODUTOS através de PRODUTO

- Tabela: Pedidos
- Descrição: Pedidos feitos pelo posto a seus fornecedores

Atributo	Descrição	Tipo	Tam.	Restrição
(*) CODPEDIDO	Código do pedido auto-incrementado	INTEIRO		NÃO NULO
FORNECEDOR	Cód. Fornecedor	INTEIRO		NÃO NULO
DATAPEDIDO	Data do pedido	DATA		
DATAENTREGA	Data de entrega	DATA		
TOTAL	Valor total do pedido	REAL	15:2	

Chave Primária: CODPEDIDO

Chaves Estrangeiras: FORNECEDOR

- A tabela PEDIDOS se relaciona com a tabela FORNECEDORES através de FORNECEDOR

Tabela: Reposicoes

Descrição: Reposições feitas do depósito para outro estoque (pista, bar,...)

Atributo	Descrição	Tipo	Tam.	Restrição
CODREPOSICAO	Código da reposição auto-incrementada	INTEIRO		NÃO NULO
DATA	Data da reposição	DATA		
TURN0	Turno da reposição	CARACTER	1	NÃO NULO
ESTOQUE	Estoque destino	CARACTER	1	NÃO NULO

Chave Primária: CODREPOSICAO

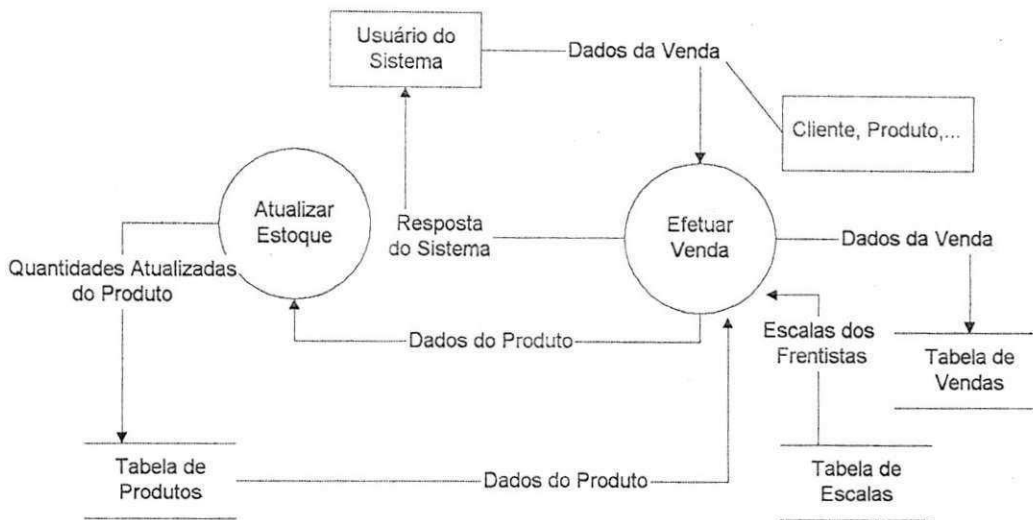
- Tabela: Transferências
- Descrição: Transferências feitas de um posto para outro na rede de postos

Atributo	Descrição	Tipo	Tam.	Restrição
CODTRANSF	Código da transferência auto-incrementado	INTEIRO		NÃO NULO
DATA	Data da transferência	DATA		NÃO NULO
DESTINO	Destino da transferência	CARACTER	20	NÃO NULO

Chave Primária: CODTRANSF

Diagramas de Fluxo de Dados (DFD) e Mini-Especificações

DFD Efetuar Venda



Mini-Especificação Efetuar Venda

Vender (Dados da Venda)

Recupera (quantidade em estoque, Tabela de Produtos)

Se quantidade em estoque > quantidade vendida Então

Retorna Venda não Autorizada

Senão

Recupera (Escala, Tabela de Escalas)

Se Frentista não Escalado Então

Retorna Venda não Autorizada

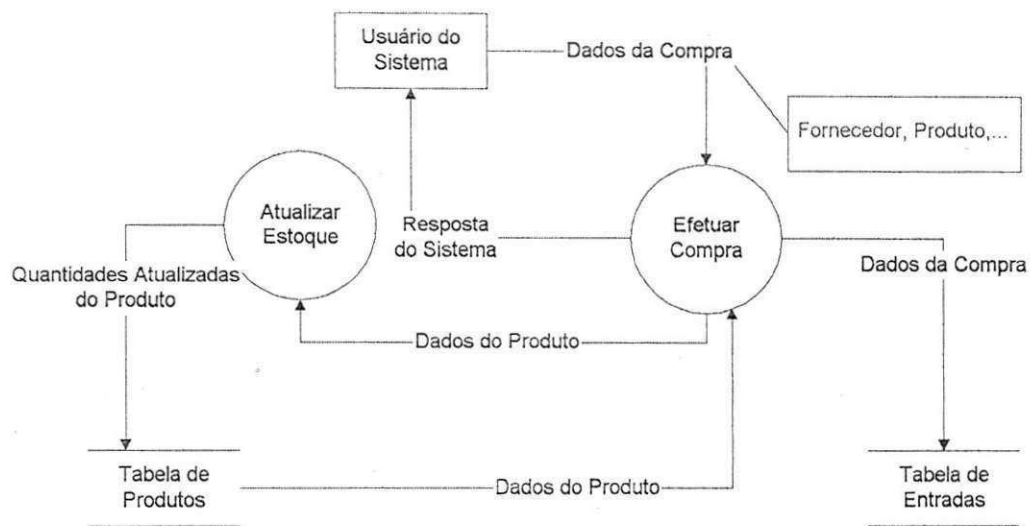
Senão

Atualizar Estoque (Dados do Produto)

Armazena (Dados da Venda, Tabela Vendas)

Retorna Venda Confirmada

DFD Efetuar Compra



Mini-Especificação Efetuar Compra

Comprar (Dados da Compra)

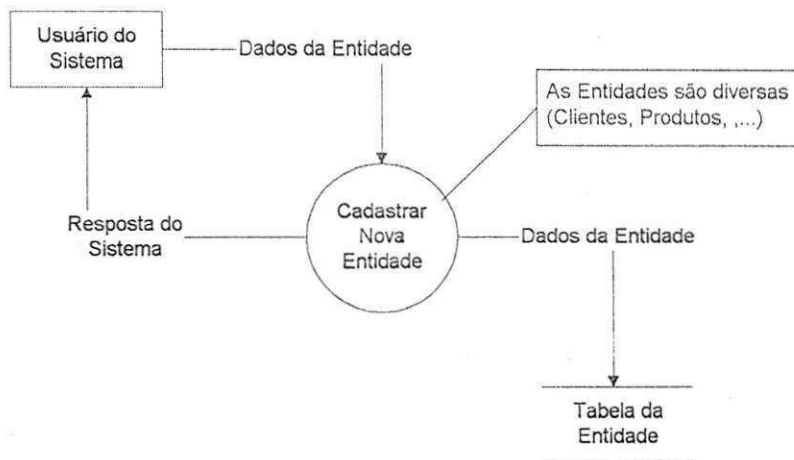
Recupera (quantidade em estoque, Tabela de Produtos)

Atualizar Estoque (Dados do Produto)

Armazena (Dados da Compra, Tabela Entradas)

Retorna Compra Confirmada

DFD Cadastrar Nova Entidade



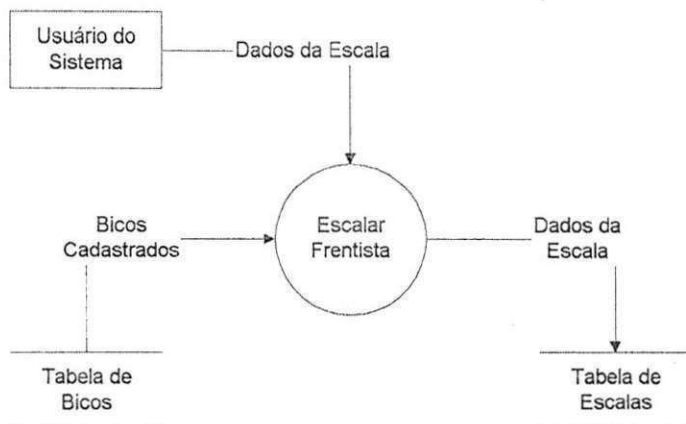
Mini-Especificação Cadastrar Nova Entidade

Cadastrar (Dados da Entidade)

Armazena (Dados da Entidade, Tabela da Entidade)

Retorna Cadastro Confirmado

DFD Escalar Frentista



Mini-Especificação Escalar Frentista

Escalar (Dados da Escala)

Turno = {manhã, tarde, noite}

Recupera (bicos, Tabela de Bicos)

Para cada Turno Faça

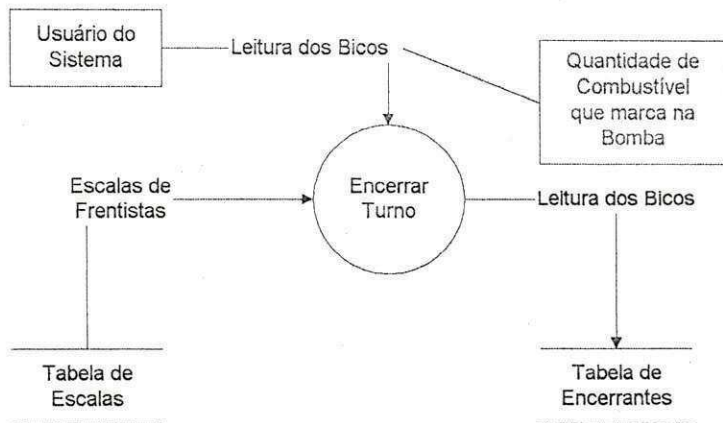
 Para cada bico Faça

 Escala Frentista para bico em Turno

 Armazena (Dados da Escala, Tabela de Escalas)

Retorna Escala Confirmada

DFD Encerrar Turno



Mini-Especificação Encerrar Turno

Encerrar (Leitura dos Bicos)

Turno = {manhã, tarde, noite}

Recupera (escalas, Tabela de Escalas)

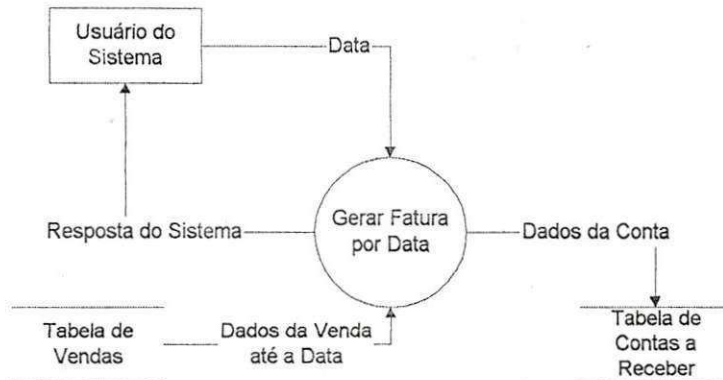
Para cada escala Faça

 Registra Leitura do Bico

Armazena (Leitura dos Bicos, Tabela de Encerrantes)

Retorna Turno Encerrado

DFD Gerar Fatura por Data



Mini-Especificação Gerar Fatura por Data

Gerar Fatura (Data)

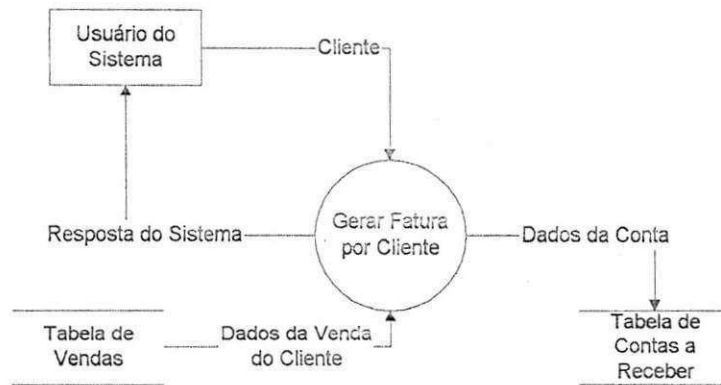
Recupera (Dados da Venda até Data, Tabela de Vendas)

Para cada Conta "não paga" faça

 Armazena (Dados da Conta, Tabela de Contas a Receber)

Retorna Fatura Gerada

DFD Gerar Fatura por Cliente



Mini-Especificação Gerar Fatura por Cliente

Gerar Fatura (Cliente)

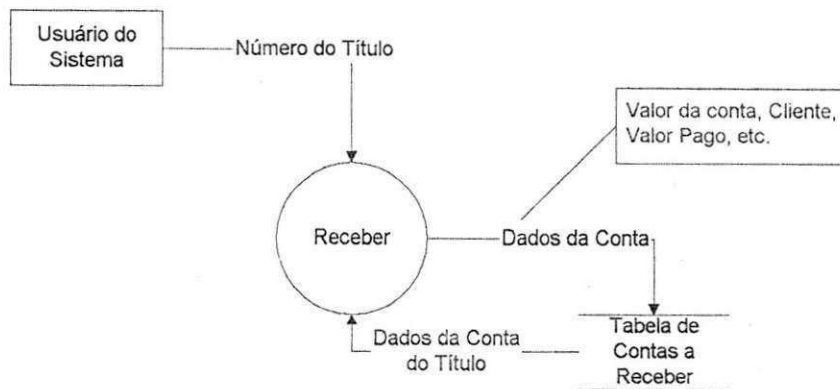
Recupera (Dados da Venda do Cliente, Tabela de Vendas)

Para cada Conta "não paga" faça

 Armazena (Dados da Conta, Tabela de Contas a Receber)

Retorna Fatura Gerada

DFD Receber



Mini-Especificação Receber

Receber (Título)

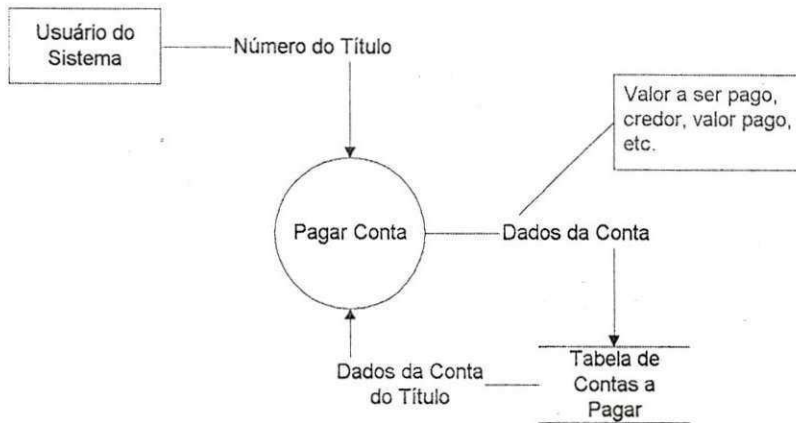
Recupera (Dados da Conta do Título, Tabela de Contas a

Receber)

Armazena (Dados do Recebimento, Tabela de Contas a Receber)

Retorna Recebimento Efetuado

DFD Pagar Conta



Mini-Especificação Pagar Conta

Pagar (Título)

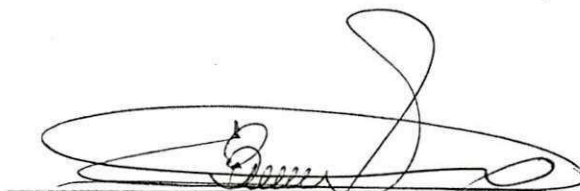
Recupera (Dados da Conta do Título, Tabela de Contas a Pagar)

Armazena (Dados do Pagamento, Tabela de Contas a Pagar)

Retorna Pagamento Efetuado

Declaro para os devidos fins que Lysianne do Couto Andrade estagiou na Empresa Microsistemas Informática LTDA, cumprindo as atividades planejadas.

Campina Grande / /



João Bosco de Macêdo

*Av. Floriano Peixoto, 229
Centro – CEP.: 58.100 -001
Campina Grande - Paraíba
Fone/Fax: (0xx83)342 4452
microsis@cgnet.com.br
CNPJ.: 03.247.637/0001-75
Insc. Est.: 16.124.582-0*