

FRANCISCO ADRIVAGNER DANTAS DE FIGUEIREDO

Relatório Estágio

Universidade Federal da Paraíba - UFPB  
Centro de Ciências e Tecnologia - CCT  
Departamento de Sistemas e Computação - DSC

Estagiário: Francisco Adrivagner Dantas de Figueiredo

Matrícula: 881-1350-6

Curso: Processamento de Dados

Empresa: Springer Panasonic da Amazônia S. A.

Local: Manaus - AM

Departamento: Centro de Processamento de Dados

Período: 90.2

Orientador: Marcus Costa Sampaio

Supervisor: Hitoshi Takano

Manaus - Fevereiro de 1991



Biblioteca Setorial do CDSA. Março de 2021.

Sumé - PB

Universidade Federal da Paraíba  
Centro de Ciências e Tecnologia  
Departamento de Sistemas e Computação

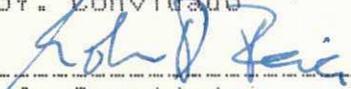
Aluno: Francisco Adrivagner Dantas de Figueiredo

Empresa: Springer Panasonic da Amazônia S. A.

Área de Estágio: Análise de Sistemas

Período de Estágio: 29/08/90 a 31/01/91

A Coordenação de Estágio do DSC, adotando os procedimentos de praxe para a avaliação da disciplina Estágio Supervisionado, apresenta seu parecer final:

Comissão Examinadora	Relat.	Defesa
 Prof. Orientador	A	A
 Prof. Convidado	A	A
 Prof. Convidado	A	A
Nota Final:	A	( A )

Tendo em vista o conceito atribuído ao aluno pelo estágio realizado conforme autorização deste Departamento de Sistemas e Computação e tendo em vista as normas que regulam o Estágio Supervisionado o aluno tem direito a \_\_\_\_ créditos.

Campina Grande, \_\_\_\_ de \_\_\_\_\_ de 1991.

## Agradecimentos

---

Agradeço a meus pais, irmãos e noiva pela compreensão nas ausências e pela força e estímulo nas ocasiões necessárias.

Agradeço também aos meus mestres e colegas que nunca mediram esforços para cooperar no decorrer das minhas atividades acadêmicas.

Finalmente agradeço aos colegas da Springer Panasonic da Amazônia, que muito me apoiaram e colaboraram na realização deste meu trabalho.

1. Introdução
2. Springer Panasonic, Um Breve Histórico
3. O Centro de Processamento de Dados
4. O Ambiente de Desenvolvimento
  - 4.1. O Net-pass
  - 4.2. O Com-plete
  - 4.3. O Adabas
  - 4.4. Normas, Metodologias e Padrões
    - 4.4.1. Normas Para Desenvolvimento Natural
    - 4.4.2. Metodologia Para Análise de Sistemas
    - 4.4.3. Metodologia Para Desenvolvimento de Sistemas
    - 4.4.4. Padrões - Panasonic, Divisão de Informática
5. O Sistema de Faturamento
  - 5.1. O Sistema Atual
  - 5.2. O Invoice
6. Conclusão
7. Bibliografia

ANEXOS

Capítulo 1

Introdução

---

Este relatório destina-se à apresentação da defesa do Estágio Supervisionado do aluno Francisco Adrivagner Dantas de Figueiredo realizado na Springer Panasonic da Amazônia S.A., no período de 29 de Agosto de 1990 a 31 de Janeiro de 1991 no Centro de Processamento de Dados desenvolvendo a função de Analista de Sistemas, tendo como orientador o Prof. Marcus Costa Sampaio, Professor do Departamento de Sistemas e Computação, e como supervisor Hitoshi Takano, gerente do Centro de Processamento de Dados da Springer Panasonic da Amazônia S. A.

No Capítulo 2 temos um histórico da empresa desde o seu surgimento até hoje. No Capítulo 3 temos a estrutura do Centro de Processamento de Dados com suas áreas. No Capítulo 4 temos um resumo das ferramentas utilizadas no ambiente de desenvolvimento de sistemas: o Net-pass, o ADABAS, o Com-plete e também as normas, metodologias e padrões utilizados. No Capítulo 5 temos um resumo sobre o sistema de faturamento existente e também sobre o sistema no qual trabalhei, o INVOICE.

Capítulo 2

Springer Panasonic, Um  
Breve Histórico

A Springer Panasonic da Amazônia S. A. foi fundada em 29 de Maio de 1970, sob a razão social Springer da Amazônia S. A., tendo como sócios:

- Springer Refrigeração S. A.
- TeleSpringer S. A.
- Paulo D'Arrigo Vellinho
- Darcy Alves
- Frederico Luiz Behrends
- Ronald Schramm Ely
- Joel Alberto Lank

Inicialmente instalada na Rua Tamandaré, no centro da cidade, num prédio alugado com 365 m<sup>2</sup> de área e usando tecnologia própria e da Admiral - U.S.A., deu-se início a produção de rádios para automóveis e radiolas, utilizando para isso 53 funcionários.

Em maio de 1972, mudou-se para o Bairro de Educandos ainda em prédio alugado, desta feita com uma área de 1.200 m<sup>2</sup>, produzindo rádios portáteis. Em 11 de Dezembro de 1972 passou a denominar-se Springer Amazônia S. A. - Indústria e Comércio.

Finalmente em 01 de Setembro de 1973, mudou-se para as suas próprias instalações no Distrito Industrial da Suframa, Rua Matrinxá No. 1155, com área construída de 4.405 m<sup>2</sup>, terreno de 34.750 m<sup>2</sup>. A Atual área construída é de 24.016 m<sup>2</sup>, e terreno de 121.924 m<sup>2</sup>.

Em 1980 o número de funcionários elevou-se para mais de 900 (novecentos), ocupando uma área de 84.219 m<sup>2</sup>, sendo 19.256 m<sup>2</sup> de área construída.

Usando tecnologia própria e da Matsushita Electric Industries, Co. Ltd., do Japão, passou a produzir Rádio Relógio, Rádio Gravador, Gravadores, Televisores a cores, Toca Disco, Receivers, Tape Decks da marca Technics e Aparelho de Ar condicionado para veículos. Em março de 1981, foi assinado no Japão o Contrato de Associação entre Springer da Amazônia S. A. e National do Brasil Ltda. Juntas, elas deram origem à Springer National da Amazônia S.A. Uma empresa 50% brasileira e 50% japonesa, mas que nasceu amazonense voltada para a consolidação da Zona Franca de Manaus.

Em 02 de julho de 1984, fundamentada na política da Suframa foi inaugurada a Fábrica de Componentes, com a razão social de Springer National Componentes da Amazônia S. A.

Em 01 de Dezembro de 1990, houve uma mudança na razão social das empresas, passando a chamarem-se:

- Panasonic do Brasil Ltda.
- Panasonic Componentes do Brasil Ltda.
- Springer Panasonic da Amazônia S. A.
- Springer Panasonic Componentes da Amazônia S. A.

Hoje as duas empresas, Springer Panasonic da Amazônia S. A.

e Springer Panasonic Componentes da Amazônia S. A., contam com um quadro de mais de 1200 funcionários, fornecendo componentes para muitas empresas do Distrito Industrial da Zona Franca de Manaus e fabricando televisores a cores, vídeos cassetes (inclusive o modelo Philips), micro-systems, toca-discos, receivers e tape deck da marca Technics e fornos micro-ondas (inclusive os modelos Continental 2001 e Brastemp).

Capítulo 3

O Centro de Processamento  
de Dados

O CPD da SPAM é um CPD de médio porte, subdividindo-se em duas áreas: micro-informática e sistemas main-frame.

Na área de sistemas main-frame o primeiro sistema desenvolvido foi o PA (Production Administration) que está em sua fase final de implantação. Agora o IV (Invoice) está sendo desenvolvido parte em Manaus (70%) e parte em São Paulo (30%).

Contamos com muitos outros sistemas. O PD (Product Design) desenvolvido para auxiliar o pessoal da engenharia no projeto de modelos novos e de modelos já existentes. O PM (Panasonic Eletronic Mail System) desenvolvido para reduzir o número de comunicações internas (CI's) a circular pela empresa e também o custo com telefonia, já que mensagens podem ser enviadas para qualquer usuário cadastrado em qualquer parte do país, e também para envio de telex para qualquer empresa do grupo Matsushita espalhadas pelo mundo inteiro. O PC (Product Cost) desenvolvido para acompanhamento dos custos de produção dos modelos constantes do PD, simulação de custos, criação de modelos fictícios (alterações no modelo original) para comparação de custos reais de fabricação. O OE (Order Entry), o FS (Financial Statement), o MI (Management Information), o IS (Infra Structure), todos desenvolvidos em linguagem Natural, e também o NATOS (National Total On-Line System) com suas duas divisões: Materiais e Produtos Acabados, que estão sendo progressivamente substituídos por novos sistemas, já que são sistemas muito antigos e de difícil manutenção.

Quando da implantação dos sistemas desenvolvidos em São Paulo os analistas responsáveis pelo sistema em Manaus recebem treinamento para que possam prestar assistência aos usuários.

Contamos com uma equipe de 2 analistas para micro-informática, 5 analistas para os sistemas main-frame, 3 operadores, 1 coordenador de operação e 1 digitadora para as entradas de dados mais pesadas dos sistemas do COBRA. Não contamos com nenhum programador, o que eu acho uma grande falha, já que o analista perde muito tempo programando, quando poderia estar desenvolvendo tarefas mais específicas de sua função. Outra grande falha é não haver reciclagem entre os analistas. Cada analista é responsável pelos seus sistemas. Não há um treinamento, mesmo que básico dos analistas em outros sistemas que não sejam os seus. E isso acarreta muitos problemas quando um analista é afastado da empresa ou mesmo quando sai de férias, já que ninguém tem conhecimento suficiente para resolver os problemas daquele sistema, causando, muitas vezes insatisfação dos usuários. Outro problema bastante sério é que a equipe de Manaus não tem autonomia para tomar certas decisões, dependendo sempre de uma posição do pessoal de São Paulo. Tudo que for projetado para um sistema desenvolvido pelo pessoal de Manaus tem que passar pela análise e depender da aprovação do pessoal de São Paulo, mesmo que tudo esteja dentro dos padrões estabelecidos. Muitas vezes as questões levantadas pelo pessoal de Manaus ficam em segundo plano, esperando por uma resposta, que pode levar até mais que um mês, atrasando, dessa forma, todo o cronograma que foi estabelecido para o desenvolvimento daquele sistema.

A área de micro-informática funciona como sendo uma estrutura de C. I. Tem serviços de atendimento a usuários, cursos e mais a tarefa extra de desenvolver diversos sistemas para auxiliar os trabalhos desenvolvidos por outros departamentos.

Existem diversos micros espalhados por todas as fábricas (TV, Vídeo, Forno Micro-Ondas, Áudio, Placa Montada, Capacitor Cerâmico, Micro Motor, Soquete, etc) e por outros departamentos (Administração, Suprimentos, Manutenção, etc) com as mais diversas aplicações: controle de produção por funcionário, planilhas de produção de modelos, controle de frequência, etc., além de existirem vários micros com modem para transmissão de telex para o Japão via Nova York.

Está sendo implantada uma rede local utilizando o software de rede Novell. Esta rede servirá para minimizar as despesas com compra de software e também para centralizar as informações com a finalidade de melhorar o controle sobre as mesmas.

Dispomos de dois sistemas IBM 4381 interligados, situados em São Paulo. Mantemos comunicação com São Paulo através de 4 linhas TRANSDATA (tres para terminais e impressoras e uma para transmissão de RJE.) e 1 linha RENPAC de reserva e mais uma linha discada (veja ANEXO, vol. II, item 01).

Temos também um sistema COBRA - 480 com 8 terminais e 3 impressoras. Este computador é utilizado para transmissão de RJE, além de possuir alguns sistemas nele desenvolvidos (Importação, Faturamento). Este sistema será desativado, ficando seu uso

restrito a transmissão de RJE.

O Net-pass (Network Parallel Application Session System) possibilita aos usuários de uma rede VTAM a conexão simultânea a múltiplas aplicações, tais como Com-plete, CICS, TSO, ROSCOE, IMS/DC, etc. Isto permite a passagem de uma sessão a outra de forma extremamente simples - basta pressionar uma tecla - não importando se as sessões estão no mesmo computador ou em diferentes pontos da rede. A retomada de uma sessão suspensa é feita a partir de seu ponto de interrupção. Essas facilidades concorrem para um significativo aumento da produtividade do ambiente, pois praticamente elimina o tempo gasto na navegação entre aplicações.

O Net-pass além de proporcionar rápida instalação e operação, garante total segurança ao ambiente através de sua interface com RACF, ACF2 ou TOPSECRET (sistemas de segurança).

O Net-pass pode direcionar o usuário para uma área específica de uma operação, o que é mais conveniente e também mais rápido, pois os "pseudo-I/O" são executados na CPU e não têm que ser enviados pela linha de comunicação.

O Net-pass permite o recebimento de mensagens pelos usuários durante o logon ou a qualquer instante (comando NEWS), possibilitando assim, a centralização desta importante função em um único componente do ambiente.

Com o Net-pass os usuários podem enviar mensagens (comando

Capítulo 4

O Ambiente de Desenvolvi-  
mento

SEND) a outros usuários ou a um grupo deles ou ainda a todos eles, garantindo o seu recebimento, com diferentes graus de urgência, independente das aplicações às quais eles estejam conectados.

O Net-pass permite a transmissão de qualquer imagem da tela (comandos SHOWUID ou SHOWTID) para outro terminal ou impressora. O comando LOG possibilita o envio de uma sequência de telas para uma impressora.

O usuário pode liberar o seu terminal físico, suspendendo todas as sessões às quais esteja conectado para posterior continuação do seu trabalho, podendo ainda conectar-se a outro terminal. Através do comando LOCK, o Net-pass aciona um mecanismo que impede qualquer operação de I/O naquele terminal, até que o usuário forneça novamente sua password.

Para o usuário de única sessão o Net-pass Easy-log funciona como alternativa para o VTAM USS TABLE, pois permite o recebimento de qualquer informação genérica. E com o uso de pf-keys direciona os usuários para a aplicação designada, liberando todos os recursos envolvidos com troca de aplicações.

Qualquer usuário de um terminal IBM-3270 ou compatíveis pode visualizar simultaneamente várias sessões com as quais esteja conectado. Assim, cada sessão é mostrada numa janela, podendo ser controlada a partir daí. Um grande número de comandos permite controlar as características das janelas.

O Net-pass proporciona três diferentes opções, selecionáveis a nível de aplicação, que objetivam reduzir sensivelmente a quantidade de dados transmitidos. São elas: Data Compression, Field Net Change e Character Net Change. Com isto, pode-se diminuir de 25% a 40% o volume de dados transmitidos, reduzindo muito o tempo de resposta dos usuários remotos VTAM.

O Net-pass é capaz de fornecer múltiplas informações sobre os usuários conectados, permitindo o completo acompanhamento da atividade existente na rede.

---

Com-plete é um ambiente de teleprocessamento com funções de gerenciamento de informações avançado e um poderoso conjunto de ferramentas interativas em um único pacote de software. Sua arquitetura única é bem apropriada para todo o conjunto de tarefas de processamento interativo, e destina-se a tornar os recursos o mais disponível quanto a consistência permita.

O Com-plete executa muitas funções para os usuários do sistema de teleprocessamento:

- como um monitor de teleprocessamento, suporta redes de terminais de qualquer porte, consistindo de dispositivos locais ou remotos.

- como um processador de transações, supervisiona a execução interativa de programas e utilitários e gerencia o acesso a recursos compartilhados.

- como um complemento interativo ao ADABAS e Natural, provê acesso as facilidades do Banco de Dados e ao conjunto de funções disponíveis para usuários e programadores Natural.

- como facilidade de desenvolvimento de programas, oferece ferramentas para desenvolvimento e testes de programas escritos em linguagens tradicionais, como COBOL, PL/I e Assembler.

- como facilidade de gerenciamento de sistemas, inclui utilitários para gerenciamento de arquivos, bibliotecas, monitoramento de Sistemas Operacionais e controle do ambiente de

teleprocessamento.

As facilidades oferecidas pelo Com-plete inclui:

- suporte a teleprocessamento
- serviços de execução de programas
- suporte ao Natural
- utilitários interativos
- serviços de sessões interativas
- extensões de comunicação de dados
- serviços de sistema de segurança

As características do suporte a teleprocessamento do Com-plete incluem:

- suporte a dispositivos SNA/SLDC, sob o VTAM
- suporte a dispositivos BSC e TTY, sob o CTAM e VTAM
- suporte a terminais gráficos coloridos IBM - 3279 e a impressoras laser IBM - 3284
- sistema operacional - suporte a dispositivos independentes
- alocação e liberação dinâmica de terminais ou grupos de terminais
- comandos de operador para controlar 'spool', I/O e processamento de interrupções de terminais
- configuração de rede, em tempo de inicialização do Com-plete.

O Com-plete provê uma faixa completa de serviços de execução

de programas. Estes serviços incluem:

- I/O de terminal e funções de mapeamento
- acesso aos arquivos do Banco de Dados ADABAS
- funções de I/O em arquivos VSAM, BDAM e ISAM
- acesso a arquivos sequencias diretos (SD) do próprio Com-plete
- funções de trocas de mensagens
- funções de "spool" de impressão
- funções de paginação de terminais
- serviços do supervisor (obtenção e liberação de memória, serialização de recursos, etc.)
- entrada de job remota (RJE)
- funções de gerenciamento de tarefas

As facilidades de suporte ao ADABAS e ao Natural provêm:

- monitoramento da performance
- funções de impressão "on-line"
- suporte a gráficos
- arquivos de trabalho
- paginação de terminais
- profiles do usuário
- sessões Natural paralelas

Os utilitários interativos do Com-plete são um poderoso recurso que provê ferramentas vitais para o pessoal de processamento de dados e acesso conveniente aos serviços do

sistema para usuários finais. Os utilitários abrangem uma grande faixa das tarefas interativas, incluindo:

- desenvolvimento de programas
- gerenciamento de arquivos e bibliotecas
- controle e operações "batch"
- controle de sessões
- controle e monitoramento do sistema

#### Desenvolvimento de Programas

UEDIT - é um editor avançado disponível em ambientes "main-frame" IBM. Ele provê sessões de edição recuperáveis, profiles de usuário, definição pelo usuário das teclas de funções programadas (PF's), edição hexadecimal, suporte a "spool" de impressão, entrada de job remota (RJE) em adição as características padrões dos editores.

UMAP - é um utilitário de mapeamento interativo. Ele permite ao usuário criar, deletar, copiar e alterar mapas interativamente; gera "copy code" de definição de mapas para programas COBOL, PL/I e Assembler, e simula operações de "read" e "write" de mapas para testes.

UQ - é um utilitário de monitoramento de sistema batch. As funções de desenvolvimento de programas do UQ habilitam o usuário a monitorar a execução de jobs batch, mostrar as filas de entrada e saída do sistema, percorrer os dados de entrada e saída

---

do job ("entrar" no job) e direcionar a saída para impressoras conectadas na rede.

ULIB - com o ULIB, utilitário para bibliotecas de programas, o usuário pode adicionar ou deletar programas interativamente, sem a necessidade de alterar ou montar tabelas.

UDUMP - é um utilitário interativo de "dump". Ele habilita programadores a ver, interativamente, "dumps" no conjunto de dados e "dumps" do Com-plete com um poderoso conjunto de comandos.

#### Gerenciamento de Bibliotecas e Arquivos

UPDS - utilitário de gerenciamento do conjunto de dados particionados, habilita o usuário a listar diretórios de bibliotecas e mostrar, imprimir, deletar ou renomear membros (arquivos).

USERV - contém todas as funções do utilitário UPDS aplicáveis para ambientes DOS - VSE.

#### Operação e Controle Batch

UQ - utilitário de sistemas batch. As funções deste utilitário permitem ao usuário mostrar jobs e módulos ativos, paginação e atividades de I/O do sistema, mensagens da console do operador e informações de status de dispositivos de discos e

fitas. Com o UQ, usuários autorizados podem, também, entrar qualquer comando da console do operador e suspender, liberar e cancelar jobs e saídas de jobs.

USPOOL - utilitário para gerenciamento de spool de impressão, provê funções de gerenciamento de impressoras da rede similares às funções da console do operador.

#### Controle de Sessão

USTACK - utilitário para acesso a programas paralelos. Ele provê um menu adaptável para o Com-plete, que permite seleção de facilidades e uso da facilidade de acesso a programas paralelos. Há também um modo estendido do USTACK, destinado a programadores.

UPF - utilitário para definição de teclas de função programada (PF keys). Com o UPF o usuário pode definir comandos fixos ou variáveis a serem atribuídos com até 24 teclas PF.

UDEF - utilitário para definição de defaults do usuário. Permite aos usuários definir muitos dos seus próprios defaults para uso automático pelos utilitários Com-plete.

ULIBID - utilitário para definição de código de bibliotecas, habilita usuários a manter suas próprias tabelas de códigos de dois caracteres das bibliotecas, útil para facilitar a rápida identificação de bibliotecas quando utilizar os utilitários de gerenciamento de bibliotecas e de arquivos.

Estes utilitários também estão disponíveis no utilitário geral de profiles do Com-plete, o UPROF, que melhora e complementa estes utilitários.

#### Controle e Monitoramento do Sistema

ULOGM - utilitário de definição de profile do usuário, habilita o administrador do sistema a adicionar, modificar e excluir User Id's, passwords e opções de profiles do usuário.

UM - utilitário de gerenciamento de troca de mensagens. Ele permite ao administrador do sistema a restartar, re-enfileirar, excluir, bloquear e liberar mensagens, mostrar informações de status do terminal e rever mensagens e impressões enviadas dentro das últimas 24 horas.

ULIB - utilitário de gerenciamento de bibliotecas de programas. Com ele o administrador do sistema pode mostrar o diretório de bibliotecas de programas, bloquear e liberar arquivos e mostrar o diretório de arquivos SD (Com-plete) e informações sobre o espaço disponível em disco.

UDD - utilitário para mostrar registros em disco, permite ao programador a ver os conteúdos que qualquer registro em qualquer dispositivo de disco disponível para o sistema.

USTOR - utilitário para mostrar a memória virtual, habilita o programador a mostrar e varrer a memória virtual e acessar o

sistema de controle de blocos por nome.

Os serviços de sessão interativos do Com-plete são representativos pela sua poderosa, além de fácil de usar, interface com o usuário.

O Com-plete provê ao administrador do sistema a facilidade de estabelecer profiles para cada usuário do sistema. "Setando" as teclas de funções, opções do menu principal do Com-plete e os parâmetros defaults para os utilitários Com-plete que podem ser executados pelo usuário. Para usuários não conhecedores do sistema, pode-se especificar uma transação inicial para receber o controle automaticamente depois que o usuário se conectar (após o "logon"), simplificando o acesso às funções interativas.

A facilidade de acesso a programas paralelos (execução concorrente) é extremamente simples. Ela é bastante poderosa e fácil de usar, habilitando o usuário a invocar, suspender e reativar múltiplos programas (o limite é especificado no instante de criação do perfil do usuário) logicamente concorrentes. Esta capacidade é especialmente útil para programadores, que podem editar programas, listar definições de arquivos, listar e recuperar saídas batch, com a facilidade de, em qualquer tempo, suspender uma atividade e re-assumir uma outra do ponto de sua última suspensão.

O uso desta poderosa facilidade quase não custa nada ao processador (CPU). Ela usa a característica "rollin/rollout" do Com-plete para armazenar vários programas suspensos pelo usuário

nos arquivos de "rollin/rollout" ao invés de usar a característica usual.

O Com-plete provê serviços de impressão on-line para situações em que a informação é necessária em forma de "hard copy" (listagens). O suporte do spool de impressão é integrado com o Natural e muitos utilitários Com-plete, habilitando o usuário a obter listagens sem prender seu terminal. Em adição, o Com-plete provê um suporte tipo "print screen" que habilita imagens da tela serem copiadas em qualquer dispositivo de "hard copy" da rede.

As características avançadas do Com-plete são o resultado de uma arquitetura que provê novas capacidades enquanto evita problemas existentes em esquemas tradicionais.

A organização multi-tarefa do Com-plete, técnicas de gerenciamento de programas e o esquema de proteção da memória virtual aumenta a performance, a eficiência e a capacidade para novos níveis no ambiente de teleprocessamento.

O Com-plete suporta várias sub-tarefas do sistema operacional e mais uma sub-tarefa adicional para cada usuário definido. Este uso real de multi-tarefa habilita o Com-plete a trabalhar perfeitamente no ambiente de memória virtual.

#### - Comunicação de terminal

Esta sub-tarefa é responsável pela inicialização das operações de I/O em terminal sob o VTAM e CTAM. Sob o CTAM ele

também executa um conjunto de operações de 'read' e 'write' e detecção e recuperação de erros.

- Comunicação com o operador

A atribuição de uma sub-tarefa separada para as funções de comunicação com o operador habilita o operador da rede a interagir com o núcleo do Com-plete em situações onde este tipo de comunicação em outros tipos de sistemas de teleprocessamento seria impossível.

- Troca de mensagens e 'spool' de impressão

Uma sub-tarefa do sistema é dedicada a transmissão de impressões, eliminando a necessidade do usuário esperar o término da impressão para liberar o seu terminal.

- Paginação de terminal

Os comandos de paginação são interceptados e processados pela sub-tarefa de paginação de terminal para que aplicações que criem os arquivos de paginação não necessitem conter código para acessar o conteúdo destes arquivos.

O Com-plete suporta redes de teleprocessamento de qualquer tamanho, em processadores IBM das famílias 370, 30xx e 43xx, e compatíveis.

O suporte a terminal do Com-plete inclui a família inteira de terminais IBM 3270 e impressoras, e compatíveis, em adição a dispositivos de teletipo (TTY) e vários outros.

---

O Com-plete suporta os seguintes sistemas operacionais IBM:

- XA
- MVS
- MVS/XA
- DOS/VSE SP1 e posteriores (inclusive o SSX)
- VS1

Vários produtos destes sistemas também são suportados, como o MVS/SP e VS1/BPE.

O Banco de Dados ADABAS (Adaptable Data Base System) é composto de alguns arquivos internos:

- 1 - Associator
- 2 - Data Storage
- 3 - Work Storage (veja ANEXO, vol.II, item 02)

#### Associator

Este arquivo contém informações de controle para administrar os dados dos usuários. Possui basicamente:

- . listas invertidas;
- . conversor de endereços;
- . demais dados de controle (espaço disponível, arquivos carregados, controle de ISN).

#### Data Storage

Neste arquivo é que são armazenados os dados propriamente ditos. Estes dados são armazenados de forma comprimida, possibilitando assim, um aumento no índice de utilização de memória em disco.

#### Work Storage

Este é um arquivo utilizado pelo núcleo do ADABAS como

arquivo intermediário na execução de tarefas como:

- proteção e recuperação de arquivos do Banco de Dados
- armazenamento temporário de listas de ISN's provenientes de pesquisas
- área intermediária para operação dos algoritmos de pesquisas.

A terminologia a seguir esclarece os termos utilizados pelo ADABAS:

- Campo: é o nível elementar de informação manipulado pelo ADABAS. Seu conceito é o mesmo de um ambiente de Processamento de Dados convencional.

- Campo Grupo: equivalente ao item de grupo do COBOL ou ao "struct" do "C".

- Campo Múltiplo Valor: nada mais que uma tabela de um único campo, ou seja um array.

- Registro: um conjunto de campos constitui um registro ADABAS. Os tipos de campos podem ser todos os anteriormente mencionados. Cada registro ADABAS tem um número sequencial interno assinalado e administrado pelo ADABAS. Os registros são armazenados de forma arbitrária na memória, através de uma tabela especial de controle de áreas livres no disco.

- Bloco: um bloco ADABAS constitui um conjunto físico de registros e possui tamanho fixo em função do tipo de disco utilizado. O ADABAS sempre trabalha a nível de bloco, ou seja,

-----

cada vez que há transferência de dados entre disco e memória ou vice-versa, é transferido um bloco inteiro e não apenas um registro.

- ISN (Internal Sequence Number): consiste de um número sequencial interno associado a cada registro que exista em Banco de Dados ADABAS. Toda manipulação de registros é feita com base no ISN, que pode ser gerado tanto pelo ADABAS como pelo próprio programa que estiver gerando o arquivo.

- RABN (Relative ADABAS Block Number): é o número relativo do bloco dentro do arquivo ADABAS.

#### Compressão de Dados

O ADABAS comprime os dados para uma melhor utilização do espaço físico do disco. Existem tres tipos de compressão de dados:

- Default ou Standard
- Null Supression (Supressão de Campos Nulos)
- Fixed (Tamanho Fixo)

#### Compressão Default

Neste tipo de compressão o ADABAS elimina os zeros a esquerda dos campos numéricos e os brancos a direita dos campos alfanuméricos (no-leading, no-trailing). Os campos numéricos são representados em Decimal Compactado. O primeiro byte de cada

campo indica o seu tamanho mais 1 (o byte de tamanho).

Exemplo: Temos o seguinte registro:

15 bytes para NOME

06 bytes para MATRIC

15 bytes para CARGO

06 bytes para SALARIO, que totaliza 42 bytes.

Este registro, com os seguintes valores:

NOME = Adrivagner

MATRIC = 3205

CARGO = Anal. Sistemas

SALARIO = 150000

Após a compressão, ocuparia 36 bytes.

#### Supressão Nula

Os campos nulos simplesmente são eliminados, restando apenas o byte de tamanho.

Exemplo: No registro do exemplo anterior o campo CARGO tem seu conteúdo como sendo brancos e o campo MATRIC tem zeros. Usando a compressão default o registro ocuparia 21 bytes, ocupando-se 4 bytes para representar os dois campos vazios. Se usarmos a supressão nula ao invés de ocuparmos 4 bytes, ocuparíamos apenas 1, com valor 'C2' (decimal compactado) que indica a presença de 2 campos com valores nulos. Quando os últimos campos de um registro forem nulos, não será gerado nenhum

indicador de campo nulo.

#### Tamanho Fixo

Campos com compressão do tipo tamanho fixo não sofrem nenhum tipo de compressão. Utilizando o mesmo registro dos exemplos anteriores, ele ocuparia os 42 bytes (seu tamanho) se todos os campos estivessem indicados como sendo de tamanho fixo. Logicamente isso não é bom, pois o ideal é economizar espaço em disco.

OBS: A compressão do tipo tamanho fixo é a nível de campo e não de registro.

#### Data Lock

O esquema de "Data Lock" do ADABAS também é bastante interessante. Todo registro acessado fica em estado de "hold" (bloqueado) até que seja liberado pelo programa que o acessou ou pelo próprio ADABAS. Existe um parâmetro de instalação que define o número máximo de registros que podem ficar em "hold" para o mesmo programa.

No caso de atualizações em registros, o mesmo fica em "hold" até que o programa o libere efetivando a atualização (end transaction), ou cancelando a atualização (backout transaction) ou ainda se for excedido o tempo máximo que um registro pode

---

ficar em "hold", o próprio ADABAS executa um "backout transaction", liberando o registro. Um detalhe importante é que todas as atualizações em arquivos da base de dados são realizadas em "buffers" de trabalho (na memória) só sendo efetivadas (no disco) com execução de um "end transaction". O tempo máximo que um registro pode ficar em "hold" é especificado por um parâmetro de instalação.

#### Descritores de Arquivos

Descritores, subdescritores, superdescritores e descritores fonéticos são campos que podem ser utilizados como chaves de acesso a arquivos ADABAS.

#### Descritores Simples

Podemos ter vários descritores dentro do mesmo arquivo. Com isso, podemos acessar um registro deste arquivo por várias chaves. Consideremos, como exemplo, um arquivo de um sistema de consultas On-Line a saldos de contas bancárias.

Normalmente utiliza-se o código da agência e o número da conta como chaves de acesso. Porém se o usuário (cliente) não sabe ou o código da agência ou o número da conta, e o nome do cliente também é um descritor, usa-se o nome para acessar os dados referentes ao saldo de sua conta bancária.

#### Subdescritores

São descritores formados por parte de um campo.

Exemplo: Selecionar todos os modelos, cujas posições de 5 a 8 de seus códigos sejam iguais a "A093".

Para que isto seja possível deverá existir um subdescriptor com posições de 5 a 8 do campo COD-MODELO.

#### Superdescritores

São descritores formados por partes ou pela totalidade de vários campos.

Exemplo: Selecionar todos os fornecedores cujos CGC iniciem com "080" e forneçam material "COLA".

Para que isto seja possível deve existir um superdescriptor que tenha as posições de 1 a 3 do campo CGC-FORNEC e o campo MAT-FORNECIDO.

#### Descritores Fonéticos

São descritores utilizados para acessar registros através de palavras chave. Para isso, é utilizada uma rotina especial que gera os códigos fonéticos da palavra desejada.

Exemplo: Selecionar todos os clientes cujo nome contenha a composição "SONY". Para isto deve existir um descriptor fonético para o campo NOME-CLIENTE.

OBS: Os descritores simples, subdescritores e superdescritores são quem dão origem às listas invertidas.

#### Listas Invertidas

Listas invertidas são arquivos gerados automaticamente pelo ADABAS, dentro do associator, com as seguintes características:

- são classificadas ascendentemente pelo valor do descritor;
- contém o número de registros que satisfazem o conteúdo do descritor;
- contém a lista de ISN's desses registros.

Vamos utilizar como exemplo um arquivo com dados dos automóveis de alguma revendedora. Os registros são formados pelos campos: COR, MODELO, ANO-FABRIC, TIPO (gas., álcool, diesel), PLACA, PRECO. Os campos COR, MODELO E PLACA são descritores. Todas essas informações bem como tamanho, formato, nome dos campos são geradas pelo Administrador de Banco de Dados e dão origem a uma FDT (File Description Table). Se resolvermos consultar os dados de todos os carros de cor azul que estão no arquivo, o ADABAS seguirá os seguintes passos:

1. pesquisa a lista invertida para aquele descritor (algoritmo não especificado) para localizar o valor do descritor (azul).
2. obtém o próximo ISN da lista de ISN's.
3. obtém o RABN para aquele ISN através do Conversor de

Endereços.

4. Acessado o bloco, pesquisa-o sequencialmente para encontrar o registro pretendido.

5. retorna ao passo 2 (gera um loop para que se varra toda a lista de ISN's, veja ANEXO, vol. II, item 03).

OBS: o acesso a arquivos ADABAS pode ser:

- . sequencial lógico;
- . sequencial físico;
- . sequencial por ISN;
- . por descritores (veja ANEXO, vol. II, item 04).

#### FDT - File Description Table

A FDT é uma tabela que descreve os campos dos arquivos ADABAS, e contém basicamente:

1 - tipo do campo

- . múltiplo
- . periódico ou
- . grupo

2 - nível hierárquico do campo dentro do registro

3 - nome ADABAS (nome do campo com 2 letras gerado pelo ADABAS)

4 - nome (nome do campo gerado pelo DBA)

## 5 - formato

- . numérico zonado
- . alfanumérico
- . numérico compactado
- . binário

## 6 - tamanho do campo

## 7 - opção de compressão

- . default
- . null supression
- . fixed

## 8 - se campo descritor

- . D - descritor simples
- . S - superdescritor
- . B - subdescritor
- . P - descritor fonético,

além de informar se é descritor único (sem duplicidade).

A tabela a seguir é um exemplo de uma FDT construída pelo ADABAS com opções escolhidas pelo DBA.

## FDT - File Description Table

TYPE	LEVEL	ADABAS NAME	NAME	FORMAT LENGTH	COMPRESSION OPTION	DESCRIPTOR
	1	AA	NUMERO-COD	U 7		DE U0
	1	AB	NOME	A 40	NU	
MU	1	AC	PROFISSAO	A 10	NU	DE
	1	AD	SEXO	A 1	FI	
PE	1	AE	CARTAO-CRED			
	2	AF	NOME-CARTAO	A 15	NU	DE
	2	AG	LIMITE	P 7	NU	
GR	1	AH	ENDERECO			
	2	AI	RUA	A 40	NU	
	2	AJ	NUMERO-END	U 5	NU	
	2	BA	CEP	P 3		

---

### Restrições do Banco de Dados ADABAS

O banco de dados ADABAS tem algumas restrições numéricas que seguem:

- 3000 bytes como tamanho máximo de registro comprimido.
- 99 ocorrências de grupos periódicos por registro.
- 191 campos múltiplos por tabela.
- 16,7 milhões de registros por arquivo.
- 255 arquivos por banco de dados.
- 500 campos múltiplos por arquivo.
- 200 descritores por arquivo.

Esses números são números máximos, logo, podem ser usados, mas com isso, a performance na utilização do arquivo não será das melhores.

Existem várias bibliotecas no ambiente de desenvolvimento NATURAL. Cada uma delas tem um propósito bem específico.

São elas:

- . SLIBss -> utilizada para desenvolvimento dos protótipos do sistema.

- . SISTEMAS -> utilizada para desenvolvimento e testes isolados de programas.

- . TLIBss -> utilizada para testes mais complexos do sistema como um todo.

- . PLIBss -> utilizada no ambiente de produção.

- . Duserid -> biblioteca particular de cada usuário.

onde:

ss -> código do sistema;

userid -> identificação do usuário.

Todo sistema tem seus protótipos (mapas, etc.) desenvolvidos na biblioteca SLIB. Os programas são todos desenvolvidos na biblioteca SISTEMAS.

Depois de preliminarmente testados os programas são transferidos para a TLIB (read only) e são testados em conjunto sob o sistema de segurança. Sendo aprovados eles são transferidos para a biblioteca PLIB onde entram em produção

também sob o sistema de segurança.

No caso de alterações em programas, eles devem ser copiados para a biblioteca SISTEMAS e seguem todos os passos descritos anteriormente.

A metodologia para análise de sistemas utilizada é o "JAD" (Joint Application Design). Este método de análise de sistemas tem alguns propósitos básicos. São eles:

- . minimizar o tempo de desenvolvimento de aplicações
- . minimizar os custos
- . consolidar o conhecimento da aplicação entre todas as partes envolvidas
- . evitar grandes mudanças durante o desenvolvimento de aplicações
- . melhorar a qualidade da aplicação
- . obter o comprometimento dos usuários com as soluções adotadas
- . obter aplicações com alto índice de participação dos usuários.

O "JAD" se divide em tres partes:

- . Reunião Inicial;
- . Reunião de Revisão;
- . Reunião de Design.

Nestas reuniões participam os usuários de todos os setores envolvidos com o sistema que está sendo analisado. Cada participante tem uma tarefa específica a executar.

Antes dessas reuniões serem realizadas são feitos levantamentos preliminares sobre o sistema atual junto aos usuários, utilizando um formulário específico para isso (veja ANEXO, vol. II, item 05).

Com esses documentos, o setor de informática toma conhecimento adicional sobre o sistema existente e as propostas dos usuários, podendo discutir, em bases mais concretas, as soluções adotadas.

A reunião inicial segue os seguintes passos:

- . apresentação do organograma dos usuários
- . objetivos e funções básicas das áreas envolvidas
- . construção do fluxo atual em dfd's
- . problemas, defeitos e dificuldades do sistema atual

Na reunião de revisão são feitas as:

- . definições dos objetivos do novo sistema
- . racionalização do fluxo atual.

Na reunião de design:

- . definição dos principais relatórios desejados
- . definição das funções a serem automatizadas
- . definição dos "releases" a serem desenvolvidos.

No encerramento de cada "JAD" há uma avaliação do mesmo, apresentando-se pontos positivos e negativos e sugestões para os próximos "JAD's".

Após a fase de análise ser concluída inicia-se a fase de desenvolvimento com os seguintes passos:

- . definição do projeto lógico da base de dados para o sistema (MER)
- . definição do projeto físico da base de dados para o sistema (modelo relacional)
- . definição dos mapas, relatórios, lógica de processos, etc.
- . criação de protótipos
- . codificação dos programas
- . testes isolados
- . integração dos módulos
- . criação de massa de dados para testes
- . teste do sistema integrado
- . treinamento dos usuários
- . testes com os usuários
- . implantação
- . acompanhamento
- . manutenção

Todos esses passos são totalmente documentados (no próprio computador)

1. para uma posterior análise do desenvolvimento do sistema
2. para se confrontar o projeto com a análise e desenvolvimento

3. para facilitar possíveis manutenções no sistema.

Para uma melhor organização e também para facilitar a manipulação dos nomes de arquivos, programas, variáveis, etc foi estabelecido um padrão para ser utilizado por todas as empresas do grupo PANASONIC do continente americano que utilizem linguagem Natural e banco de dados ADABAS. A seguir serão explicados e exemplificados alguns componentes mais importantes deste padrão.

- Sistema: todo sistema tem seu título em inglês e um código de duas posições alfanuméricas. Este código é utilizado em vários locais, tais como: nomes de programas, view's, etc.

exemplo: Invoice - IV, Production Administration - PA, Data Dictionary - DD, Panasonic Electronic Mail System - PM, etc.

- Programas: os nomes de programas são formados por oito posições alfanuméricas com os seguintes significados:

SSXXXT99

SS - código do sistema

XXX - código da opção no menu do sistema, para o caso de programas on-line, ou número sequencial de 000 a 999 no caso de programas batch.

T - tipo do programa

X - consulta a base de dados

A - atualiza a base de dados

99 - número sequencial de 00 a 99 que especifica a número do programa para aquela opção.

exemplo: IUGDIA01, PACEMX01, DD034X01, PA555A01, etc

Arquivos - os arquivos físicos ADABAS não possuem nomes, mas

sim números de 000 a 255.

View - os nomes de views são compostos por até 32 posições alfanuméricas com os seguintes significados:

SS999-XX...XX

SS - código do sistema que gera o arquivo

999 - número do arquivos físico ADABAS

XX...XX - nome da view

exemplo: IV167-NOTA-FISCAL-SAIDA, PA151-ESTOQUE-MAT

No caso de serem usados arquivos VSAM-KSDS são criadas views utilizando o seguinte padrão:

VS000-nome-arq-vsam

Campos da view - os nomes dos campos das views são formados por até 32 posições alfanuméricas, de composição livre, desde que expresse, tão claro quanto possível, o que o dado que está associado àquele campo.

exemplo: NUM-NF, COD-MAT, MODELO, etc.

Mapas - os nomes dos mapas são formados por oito posições alfanuméricas com os seguintes significados:

SSXXM99

SS - código do sistema

XXX - código da opção no menu do sistema

M - fixo

99 - número sequencial de 00 a 99 que indica o número do mapa para aquela opção.

exemplo: IVGDIM01, IVGDIM02, PACEMM01, etc.

Relatórios - os nomes dos relatórios são formados por oito

posições alfanuméricas com os seguintes significados:

SSXXR99

SS - código do sistema

XXX - número do programa batch que gera o relatório

R - fixo

99 - número sequencial de 00 a 99 que indica o número do relatório para o mesmo programa.

exemplo: PA028R01, PC005R01, etc.

Bibliotecas Natural - os nomes das bibliotecas Natural são formados por até oito posições alfanuméricas, com os seguintes significados:

bibliotecas de sistemas

TLIBSS

T - tipo da biblioteca

T - teste

P - produção

S - protótipos

LIB - fixo

SS - código do sistema

Exemplo: TLIBIV, PLIBPA, SLIBPA, PLIBPM, etc.

bibliotecas de usuários

DUSERID

D - fixo

USERID - identificação do usuário

Exemplo: DADRIV, DFAB, DRINALD, etc.

Além desses padrões há uma longa lista de padrões para nomes

de variáveis, dependendo de vários fatores, tais como: tipo da variável, tipo de utilização, etc.

Capítulo 5

O Sistema de Faturamento

O sistema de faturamento atual foi totalmente desenvolvido em linguagem COBOL e possui alguns programas LTD para entrada de dados (notas fiscais, duplicatas, etc.)

Este sistema já está implementado há bastante tempo e já sofreu muitas modificações (mudança na legislação, correções de erros, adaptação a uma nova necessidade, etc.) e está totalmente desorganizado. Constantemente se tem problemas com o sistema. Os arquivos do sistema foram mal projetados e tem-se muitas dificuldades em mantê-lo funcionando, mesmo que precariamente.

O sistema não tem opções de consultas on-line, e também não tem módulos específicos para alterações em registros (apenas inclusão e exclusão). Se algum campo for digitado errado, o registro tem que ser excluído e incluído novamente com as devidas correções. Este sistema é um mero transcritor de dados. Os dados são "inputados" e o sistema gera relatórios com esses dados quase ou mesmo sem nenhum processamento.

Existem arquivos que não são mais usados, ou quase nunca foram usados, relatórios que nunca foram emitidos, programas que pouco foram executados. Por outro lado, os usuários reclamam porque não existe um determinado relatório que lhes seria muito importante, outro reclama porque está faltando um determinado campo em um relatório (este campo não existe no arquivo utilizado para gerar o relatório). É um caos, principalmente no fechamento (virada do mês), porque existem vários fechamentos (contábil, fiscal, em dólar, etc.). A partir de um estudo feito, constatou-se que não seria recompensável alterar o sistema existente e

assim decidiu-se pelo projeto de um novo sistema de faturamento, o INVOICE (IV), para que sejam atendidas todas as necessidades dos usuários.

---

O Invoice foi projetado com a finalidade de substituir o antigo sistema de faturamento existente no COBRA. Ele foi esquematizado em duas etapas: carteira de pedidos e o faturamento propriamente dito.

A etapa da carteira de pedidos já estava na fase de testes com os usuários quando foi encontrada uma falha gravíssima; usuários estavam tendo acesso a informações que nada se relacionavam com o seu departamento. Isso foi devido ao controle de acesso ter sido planejado apenas por companhia. Mas com isso teve que ser alterado, e passou a ser feito por companhia e departamento. Devido a esse problema todos os programas da carteira de pedidos tiveram que ser alterados. Foi também necessária a alteração de algumas views.

Depois de resolvido esse problema deu-se início a segunda etapa do Invoice.

Foi feito um levantamento minucioso do sistema que estava sendo utilizado, principalmente dos procedimentos manuais, para que pudessem ser feitas as melhorias necessárias.

Um dos maiores problemas encontrados foi a falta de controle sobre mercadorias em trânsito, o controle de frete por modelo e controle de produção, vendas e estoque.

Para resolver estes problemas foi preciso criar uma integração entre vários sistemas: o PA (Production Administration), o Natos-P (Produtos Acabados) e o OE (Order Entry), isto para não haver necessidade de um sistema estar

alimentando outro com seus dados. Ao invés disso, apenas alguns arquivos de outros sistemas são acessados.

Com isso conseguimos cobrir grande parte das exigências dos usuários, e além disso foi oferecido a eles uma grande quantidade de consultas aos dados da faturamento da empresa.

O volume II, itens de 06 a 12 contém a parte da documentação do sistema que não é sigilosa: DFD, DER, mensagens de erros, relação de opções do sistema, views, mapas e o dicionário de dados com todas as entidades com seus respectivos atributos.

Capítulo 6

Conclusão

Fundamentalmente, o papel da Universidade é edificar uma base de apoio que permita ao aluno, durante a vida profissional, superar obstáculos e desafios. Diante disso, a disciplina Estágio Supervisionado se apresenta como complemento dos estudos, dando oportunidade ao aluno de por em prática seus conhecimentos teóricos.

Partindo desse princípio é importante que se dê mais atenção ao estagiário, sendo necessário maior assistência por parte da Coordenação e da empresa. É preciso também que o estagiário seja mais bem preparado, tendo uma idéia de como funciona uma empresa e qual seu papel dentro dela, pois um aluno bem preparado tem maiores chances de obter um melhor aproveitamento.

Capítulo 7

Bibliografía

Com-plete Introduction Manual

Software AG, 1987

Adabas Fundamentals

Software AG, 1987

Banco de Dados e Linguagem de Quarta Geração

Panasonic, Divisão de Informática, 1988

Natural 2 - Reference Manual

Software AG, 1987

Natural 2 - Programmers Guide

Software AG, 1987

Natural 2 - Reference Data

Software AG, 1987

Adabas, Concepts and Facilities

Software AG, 1987

Natural Programming I

Software AG, 1987

Natural Programming II

Software AG, 1987

Normas e Padrões Para Desenvolvimento Natural

Panasonic, Divisão de Informática, 1988

Padronização do Desenvolvimento

Panasonic, Divisão de Informática, 1988

JAD - Joint Application Design

Panasonic, Divisão de Informática, 1988