

**INSTITUTO SUPERIOR DE PÓS-GRADUAÇÃO**  
**CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIA DA INFORMAÇÃO**  
**DISCIPLINA PROJETO DE SISTEMAS**

**GLOBAL EDI – PROPOSTA DE UM SISTEMA  
QUE GERENCIE TROCAS DE  
INFORMAÇÕES ENTRE ORGANIZAÇÕES, DE  
FORMA RÁPIDA, SEGURA E BARATA**

**MARINGÁ**  
**JULHO 2001**

**ANA MARIA LEMOS  
LEANDRO MAGNO CORREA DA SILVA**

**GLOBAL EDI – PROPOSTA DE UM SISTEMA  
QUE GERENCIE TROCAS DE  
INFORMAÇÕES ENTRE ORGANIZAÇÕES, DE  
FORMA RÁPIDA, SEGURA E BARATA**

**Trabalho de Pós-Graduação apresentado  
à disciplina de Projeto de Sistemas, do  
Curso de Tecnologia da Informação, do  
Instituto Superior de Pós-Graduação.**

**Orientador: Professor Luís Pedro Zambon**

**MARINGÁ  
JULHO 2001**

## *Agradecimentos*

Agradecemos primeiramente a Deus que, concedeu-nos o dom de aprender e produzir, inseriu-nos num contexto de mundo onde foi possível chegar até o término não só deste trabalho mas também ao final do curso de pós-graduação, e principalmente, ilumina-nos dia-a-dia em tudo o que fazemos.

Eu, Ana Maria, agradeço a minha mãe, que me incentivou durante todo do desenvolvimento do projeto e a todos que de alguma forma nos ajudaram para a conclusão deste.

Eu, Leandro, agradeço à minha esposa Rejane que, além de dar um sentido a minha vida, mesmo tantas vezes privada de minhas atenções, incentivou-me durante toda esta fase de desenvolvimento, acreditando sempre em minhas capacidades. E a minha filha Laura, que mesmo antes de nascer já é motivação para a perseverança e determinação.

Agradecemos aos nossos pais pela confiança, dedicação e carinho com que criaram seus filhos, não medindo esforços para que houvesse todo o apoio psicológico necessário que possibilitasse o desenvolvimento deste trabalho.

Agradecemos a excelente atuação do professor orientador Luís Pedro Zambon. Além de nos dar atenção e apoio, sua presença foi fundamental na realização deste trabalho. Mais do que um colega de trabalho, um ser humano formidável.

## *Resumo*

Com o crescimento da Internet e o aparecimento de um mundo globalizado veio também a necessidade das empresas (de todos os portes) se capacitarem para esta economia mundial, no sentido de estarem aptas a fazerem negócios com quaisquer outras empresas parceiras ao redor do mundo, usando a mesma agilidade e segurança com que esta outra está condicionada. A não adaptação a este novo mundo lhes impõe a pena de não serem competitivas o suficiente para sobreviver neste novo contexto.

Também neste contexto extremamente heterogêneo, existe a necessidade de pequenas e médias empresas manterem as bases de dados de suas unidades (filiais, franquias ou parceiras) atualizadas. Ao mesmo tempo que a telefonia direta encarece o processo de atualização de dados e que linhas de dados dedicadas o inviabilizam, a Internet têm se mostrado um meio útil e relativamente barato de troca de informações.

Estes dois problemas criam um cenário inóspito a empresas de menor porte, algumas delas mal conseguiram se informatizar e já estão sujeitas a um mundo que tem como principal característica a *“tolerância zero”*.

Entretanto, a tecnologia da informação avançando sobre os novos recursos, agora disponíveis a, literalmente, todo o mundo, nos mostra alguns conceitos (a troca eletrônica de dados, os negócios voltados aos negócios) e tecnologias (consagradas como o FTP, o SMTP, e o POP3; e também outras de mais alto nível como o XML, o PHP, e o DELPHI) aplicáveis nestes problemas de organização da comunicação. Neste sentido este trabalho traz uma proposta de implementação de um sistema que gerencie trocas de informações entre organizações (ou unidades), de forma rápida, eficiente e barata – tornando-se um produto de fácil operação e acessível a pequenas e médias empresas.

## *Abstract*

Besides the Internet growing and the globalized world appearing, the companies (from all sizes) need to be ready to make business with any other partners companies all over the world, using the other companies same conditioned agility and safety

The non adaptation to this new world impose them a punishment: not to be competitive enough to survive in this new context.

Also in this extremely heterogeneous context, there is also the mid-small companies needing to keep their units databases brought up to date. At the same time the direct phone calls increase the data updating costs and the dedicated data communication lines becomes inviable, the Internet have been showed itself a useful and relatively cheap media to the information interchange.

These two problems create a hostile scenery to the less sizes companies, some of them had even got the information technology and are already exposed to a world whose have as the main attribute the “zero tolerancy”.

By other side, the information technology comes over new resources, now available to, literally, all the world, show us some concepts (electronic data interchange, business to business) and technologies (consagrated ones like the FTP, the SMTP, and the POP3; and also another high level ones like the XML, the PHP, and the DELPHI) applicable in these communication organization problems.

So, in this way, this work brings a implementation propose of a system whose task is manage information interchanges between companies (or units); in a speed up, safety and cheap way – becoming a low cost and easy operation product available for small companies.

## **Índice**

<b>INTRODUÇÃO.....</b>	<b>1</b>
<b>PEQUENAS EMPRESAS COM GRANDES PROBLEMAS.....</b>	<b>4</b>
<b>NOVOS E IMPORTANTES CONCEITOS.....</b>	<b>7</b>
3.1.1 EDI.....	7
3.1.2 B2B.....	8
<b>TECNOLOGIAS CONSAGRADAS E INOVADORAS.....</b>	<b>12</b>
4.1 SMTP.....	12
4.2 POP3.....	14
4.3 XML.....	18
4.4 PHP.....	19
<b>ESCOLHA DAS TECNOLOGIAS.....</b>	<b>22</b>
5.1 LINGUAGEM DE PROGRAMAÇÃO.....	22
5.2 BANCO DE DADOS.....	24
<b>DESENVOLVIMENTO.....</b>	<b>28</b>
6.1 ESCOLHA DO MÉTODO.....	28
6.2 DESCRIÇÃO DO GLOBAL EDI.....	29
6.3 REQUISITOS DO SISTEMA.....	30
6.4. DIAGRAMA DE CLASSES.....	32
6.5 DICIONÁRIO DE DADOS.....	33
<b>CONCLUSÕES.....</b>	<b>40</b>
7.1 SOBRE OS RUMOS TRAÇADOS.....	40
7.2 O FUTURO.....	40
7.3 CONSIDERAÇÕES FINAIS.....	40
<b>GLOSSÁRIO.....</b>	<b>44</b>

## **Lista de Ilustrações**

FIGURA 4.1: MODELO DE USO DO SMTP.....	13
QUADRO 4.4: BANCOS DE DADOS SUPOSTADOS PELO PHP.....	21
FIGURA 6.2: ESQUEMA GERAL DE FUNCIONAMENTO DO GLOBAL EDI.....	29
FIGURA 6.4: DIAGRAMA DE CLASSES.....	32
FIGURA A: ESQUEMA FUNCIONAL DO FTP.....	1

## Capítulo 1

### Introdução

Temos visto ao longo dos anos inúmeras empresas crescerem e, de repente, precisarem se informatizar. Ao crescerem um pouco mais, suas filiais também passam a depender da informatização de seus sistemas para um bom andamento da unidade. Depois de algum tempo o malote com disquete usado para a atualização dos dados das filias na matriz também tornou-se um processo demorado e caro, fazendo-se indispensável uma outra forma de troca de dados entre a matriz e as filiais.

Houve uma época, que ainda não acabou em muitas empresas, que foi marcada pela troca de dados através de linhas discadas, onde era freqüente operadores humanos dizendo: "Agora digite aí 'ATX3D' e nos falamos pelo CHAT", ou automaticamente com softwares de BBS em execução. Existiram, neste tempo, algumas empresas cujo principal negócio era comunicar unidades de empresas, mas que, ainda assim, tinham muito do seu faturamento reduzido pelo custo da telefonia interurbana.

Com o surgimento da Internet, as empresas comunicadoras que não passaram a usar este meio como principal sucumbiram aos "usuários avançados" das empresas clientes – já que desde então o E-MAIL (correio eletrônico) e o FTP (protocolo de transferência de arquivos) passaram a ser relativamente de fácil utilização, mas ainda deixou um grande espaço para erros de dados e, conseqüentemente, despesas adicionais.

A globalização que a Internet possibilitou e catalisou a troca de informações e trouxe uma nova realidade para as empresas, que agora tem como um de seus maiores problemas a troca de informação segura e organizada com outras empresas e/ou filiais.

Este processo de troca de informações assemelha-se muito ao EDI (troca eletrônica de dados) que cujo conceito e prática tem em suas bases problemas congênitos que geram dificuldades que oneram o processo de comunicação.

Neste ("novo") cenário é que propõe-se um utilitário que economize tempo e dinheiro, sendo acessível a pequenas e médias empresas que efetuam negócios com quaisquer outras empresas que exigem tecnologia, flexibilidade e agilidade nas transações.

Se por um lado a realidade hoje é bastante heterogênea no sentido de diversidade de soluções de hardware para troca de dados, por outro lado existem também uma infinidade de soluções de software, e é aí que esconde-se um "perigo": sabemos que não há mágica no mundo da informática e que se algo tem de ser feito, existem algumas poucas maneiras de fazê-lo de forma segura, e não milhares delas, ou seja, por traz de muitos softwares, escondem-se soluções iguais entre si e/ou promessas enganosas de soluções.

É indiscutível que com o crescimento constante dos montantes de dados trafegados, a cada vez mais, precise-se de linhas de comunicações mais rápidas e seguras. Neste sentido, deixemos para a história as linhas discadas, já que a foram projetadas para trafegar, no máximo, apenas 9600k bytes por segundo (ou menos), o suficiente para o tráfego de voz.

Paralelamente temos as tecnologias usadas nos recursos disponíveis na Internet cada vez mais se aperfeiçoando, se não por suas concepção e funcionalidades, por sua compatibilidade, criando um novo universo de soluções interdependentes mas extremamente eficientes.

Sobre este universo surgem conceitos e aplicações se propondo a solucionar novos e velhos problema, aqueles mesmos velhos problemas de comunicação entre matriz e filiais e entre empresas que negociam entre si.

No próximo capítulo trouxemos um relato algumas de nossas experiências em pequenas e médias empresas com estes mesmos problemas de comunicação, e algumas soluções implementadas.

No capítulo 3 faremos a introdução dos conceitos inovadores de B2B e EDI que estão além das tecnologias com as quais podem ser implementados.

No capítulo 4 veremos uma breve descrição do funcionamento de alguns recursos usados (e consagrados) na Internet, como: FTP, SMTP, POP3; e também de tecnologias de nível um pouco mais alto como XML e PHP.

No capítulo 5 serão mostradas as tecnologias utilizadas ressaltando os pontos pelos quais foram escolhidas para a implementação do produto.

No capítulo 6 será mostrada a modelagem e detalhes de implementação suficientes para se produzir uma versão experimental.

Finalmente no capítulo 7 trouxemos uma visão geral do trabalho em seus três momentos de desenvolvimento: passado, presente e futuro.

Alguns aprofundamentos e curiosidades que tangem à “espinha dorsal” do trabalho encontram-se nos apêndices.

## Capítulo 2

### Pequenas empresas com grandes problemas

#### Uma breve história

Em 1994, dois jovens estudantes, (Ruiz, André e Silva, Leandro), da Universidade Estadual de Maringá, num experimento acadêmico com modems (na época eram de 2.400 kbps) encontraram um novo universo já bastante difundido em São Paulo, Rio de Janeiro, Belo Horizonte e Curitiba, os *Bulletin Board Systems* (ou BBS) - definido em [SILVA].

Depois de alguns contatos montaram seu próprio BBS que começou a operar dentro de um guarda-roupas, no apartamento do André, em 20 de setembro do mesmo ano, de forma experimental e gratuita, sem compromisso algum.

Em meados de agosto do ano seguinte um de seus usuários (Leme, Marco A. S. M.) propôs uma sociedade ([LEME]) que concretizou-se em outubro de 1995 e que terminou em dezembro de 1997 com a falência da empresa.

Este tempo foi o suficiente para que o Instant BBS (como era assim chamado) se transformasse em uma referência em serviços online em Maringá (mesmo porque era a única no ramo até que fosse "atropelada" pelo primeiro provedor de Internet da cidade - a World Net Internet - em setembro de 1997), porque os dois jovens não estavam preparados para aceitar a proposta de sociedade dos fundadores da World Net.

Dentre os vários serviços do Instant BBS, um deles foi, durante quase todo o tempo de sua existência, trafegar os dados de uma empresa cliente: a Transportadora Interpreiss.

O serviço era simples e consistia em um software de BBS para atender as chamadas e separar os arquivos (de conhecimento de frete) transmitidos entre as diversas unidades da Interpreiss em suas respectivas caixas postais. Na Interpreiss, rodava um script produzido no terminal de comunicação *Telnet* para DOS.

Neste meio tempo em que o Instant esteve em operação, também uma outra empresa de Palotina, (oeste do PR), a I. Riedi Comercial de Cereais Ltda. precisou de seus serviços. Para este problema, em novembro de 1996, foi montado um outro BBS na matriz e criados scripts nas filiais. Este serviço funcionou bem até 24 de dezembro de 1997, quando a I. Riedi precisou ampliar o número de linhas de atendimento de uma para duas linhas, mas os custos para isto foram inviáveis e a empresa migrou para a internet.

Em janeiro de 1997 foi implementado uma solução parecida de troca de informações entre a Drugovich Auto Peças de Maringá e suas 4 filiais espalhadas pelo país que viria a substituir o velho e caríssimo STM-400 contratado da Telepar. A solução era baseada em FTP e contava linhas discadas para os provedores locais com scripts *Telnet* e utilitários compactadores de dados. Ironicamente o provedor a contratado pela Drugovich foi a World Net (apesar já haverem outros em Maringá)

Quando a Interpreiss migrou para a Internet, em março de 1998, o Instant BBS fechou suas portas.

### **A realidade mudou nestes poucos anos... mas nem tanto!**

Hoje a Cocari - Cooperativa dos Cafeicultores de Mandaguari Ltda., com o mesmo problema de comunicação de dados entre seus entrepostos, com apenas uma diferença que ela já possui rotinas de replicação de dados em seus aplicativos desenvolvidos em Progress. No entanto, desta vez a solução não foi a Internet, mas

caras linhas dedicadas de dados entre as unidades, um privilégio de uma empresa de médio para grande porte!

E se fosse uma empresa menor? Pois bem, em Maringá a Sicredi cotou com a GVT o custo de uma linha do tipo DDG (0800) para comunicar-se via linha discada com seus entrepostos, mas ainda não se decidiram acerca deste meio, de linhas dedicadas ou internet.

Em Maringá, a Elotech, uma software house que desenvolve software para instituições públicas com prefeituras, câmaras e autarquias também se viu recentemente obrigada a incluir recursos de EDI para que as prefeitura pudessem fazer convênios com os bancos.

Em Curitiba, outra software house concorrente da Elotech, a Meta Informática também desenvolveu seus sistemas com recursos de EDI para que seus clientes pudessem fechar convênios com bancos e outras empresas de grande porte.

Há algo de comum em todas estas experiências: todas tendem à internet e possuem os mesmos problemas que o EDI resolve.

No próximo capítulo conheceremos melhor este e outros conceitos envolvidos em soluções desta natureza.

## Capítulo 3

### Novos e Importantes Conceitos

#### 3.1.1 EDI

Segundo [ARTHA], “EDI é uma abreviatura de Eletronic Data Interchange (Intercâmbio Eletrônico de Dados), é a troca de informações eletrônicas entre os computadores de uma empresa e seus parceiros. Essa troca de informações tem como base os padrões de mensagens definidas pela EAN, garantindo, assim, que todos os parceiros, a nível mundial, utilizem documentos de mesmo ‘formato’ e criando uma linguagem única entendida por pessoas de qualquer cultura. Além disso, o EDI possibilita formas de negócios mais dinâmicas e confiáveis, reforça parceiros e requisita um baixo investimento com alto retorno”.

A utilidade e funcionalidade do EDI podem ser descritas como em [SINTEL]:

- Intercâmbio Eletrônico de Documentos (EDI) surgiu para simplificar, automatizar e agilizar todos os processos que envolvem a comunicação entre as organizações.
- Os benefícios de sua implantação incluem economia de papel, fax e telefone, melhorando as relações entre os parceiros comerciais. Ao disponibilizar de maneira rápida e precisa as informações, facilita ainda o planejamento das atividades diárias das empresas.
- Acessível a pessoas com conhecimentos mínimos de informática, essa tecnologia pode ser utilizada por qualquer organização, requerendo apenas um microcomputador, acoplado a modem e linha telefônica, e um *software* específico para comunicação e tradução dos documentos eletrônicos.

- Contar com o apoio de uma empresa especializada em EDI ajuda bastante, pois simplifica todo o processo que envolve, dentre outras etapas, as seguintes:
  - Seleção dos parceiros iniciais e dos padrões de EDI a serem adotados;
  - Escolha dos tipos de mensagens e das RVAs (Redes de Valor Agregado);
  - Integração das aplicações;
  - Instalação e configuração dos sistemas necessários;
  - Testes preliminares;
  - Treinamento dos usuários;
  - Implantação dos parceiros;
  - Acompanhamento e avaliação global do projeto”.

Note que tantas tarefas executadas por uma “empresa especializada” (como se propõe a Sintel) de fato torna-se fácil, porém tem um preço. Preço este que, em muitos casos (principalmente vindos de pequenos e médios empreendimentos), inviabiliza o processo de EDI.

A inviabilização do EDI por razões do seu preço de implantação é a principal causa do encarecimento do EDI.

É no sentido de diminuir este encarecimento do EDI que será proposto o Global EDI.

### **3.1.2 B2B**

O conceito de B2B [SML] é muito mais do que usar as tecnologias de que dispomos em uma parte do processo. B2B é a necessidade de estreitar mais e mais o relacionamento entre empresas fazendo que seus negócios fluam de maneira eficiente e segura, utilizando tecnologia.

Imaging, ERM, Workflow, XML, WEB, EDI, etc. - são tecnologias que isoladas organizam e melhoram parte do negócio. Utilizadas em conjunto permitem que empresas e clientes se relacionem de maneira estreita e segura, permitindo a integração de interesses entre empresas.

O grande avanço que queremos estabelecer é que a amplitude de visão do negócio mudou e que sem dúvida, o meio de transporte é a Web.

Uso da Tecnologia voltada para o Negócio - A necessidade de preparar os negócios para a globalização faz com que o uso da tecnologia permita que as empresas se relacionem com outras empresas que fazem parte da sua cadeia da qual ela necessita para fazer negócios

O G.E.D. - Gerenciamento Eletrônico de Documentos vem sendo largamente utilizado pelas Empresas para organizar, ganhar qualidade, agilidade e automação em seus negócios. As tecnologias de Gerenciamento de Imagens e Documentos, Workflow, ERM, continuarão a serem utilizadas mas não como propósito final de um projeto, mas sim como parte integrante de um negócio.

### **Alguns pontos importantes**

- Os negócios são complexos - As transações chegam em muitos formatos e necessitam ser manipuladas e gerenciadas por muitos sistemas.
- As Empresas estão procurando maneiras de simplificar e melhorar o gerenciamento de seus processos de negócios, sejam eles eletrônicos ou baseados em papel.
- As Empresas necessitam:
  - Acessar a informação onde estão baseados seus negócios
  - Melhorar o atendimento a clientes

- Acelerar o ciclo do negócio
- Compartilhar as informações do negócio entre clientes, fornecedores e parceiros

[EXTOL] Instituições financeiras utilizam suas redes para fazer negócios B2B nos mercados corporativos.

Os sites de serviços corporativos não estão abertos ao público, muitas vezes seus endereços nem são divulgados, as instituições estão automatizando e migrando seus processos operacionais tradicionais para a Internet/Intranet. Relatórios, fax, arquivos de integração e mesmo telefonemas que são utilizados para consolidar e confirmar negócios ou processos operacionais estão sendo substituídos por tecnologia mais eficiente, segura e barata.

Instituições Financeiras estão investindo em tecnologia de Internet para baixar os custos operacionais e melhorar a comunicação entre o cliente e a empresa. Mas onde ficam os clientes privados, ou atendimento a negócios corporativos e privilégios de informações?

As informações privilegiadas trocadas entre os funcionários das empresas terão sempre prioridade corporativa e para isto as empresas estão utilizando o conceito de B2B onde a conexão das informações tem prioridade e privilégios distintos, fazendo com que somente os clientes especiais ou corporações tenham acesso a estes dados antes de torná-los públicos.

Através da Internet, o cliente pode acompanhar posições consolidadas dos negócios realizados, executar aplicações, resgates, simular taxas e performance dos negócios. As empresas estão disponibilizando aos usuários uma conexão via Internet para auxiliar no trabalho operacional, onde cada funcionário com um micro computador tem acesso a estas informações de maneira simples e ágil, os antigos modems e linhas telefônicas já estão rapidamente sendo substituídos por fibra óptica,

TV a cabo/Internet, linhas especiais de telefone (speed), Internet Condomínio (Radio), estas facilidades mudam o cotidiano das empresas e trazem agilidade aos negócios.

## Capítulo 4

### Tecnologias Consagradas e Inovadoras

#### 4.1 SMTP

[SMTP] O objetivo do SMTP (Simple Mail Transfer Protocol – protocolo de transferência de mail simples) é transferir mail segura e eficientemente.

O SMTP é independente do subsistema de transmissão particular e requer somente um canal data stream ordenado e seguro.

Um recurso importante SMTP é sua capacidade de retransmitir mail através dos ambientes de serviço de transporte. Um serviço de transporte provê um ambiente de comunicação interprocessos. (IPCE). Um IPCE pode cobrir uma rede, diversas redes, ou um subconjunto de uma rede. É importante notar que os IPCE's não são um para um com as redes. Um processo pode comunicar diretamente com outro processo através de qualquer IPCE mutuamente conhecido. Mail é uma aplicação da comunicação interprocesso. Mail pode ser comunicado entre processos em diferentes IPCE's retransmitindo através de um processo conectado a dois (ou mais) IPCE's. Mais especificamente, mail pode ser retransmitido entre servidores (hosts) de diferente sistemas de transportes por um host em ambos sistemas de transporte.

#### ***O Modelo SMTP***

O desenho do SMTP é baseado no seguinte modelo de comunicação: como o resultado de um pedido de mail de um usuário, o emissor-SMTP estabelece um canal de transmissão de duas mãos com o receptor-SMTP. O receptor pode ser tanto o destino final ou um intermediário. Comandos SMTP são gerador pelo emissor e

enviados para o receptor. Respostas SMTP são enviadas do receptor para o emissor em resposta aos comandos.

Uma vez que o canal de transmissão é estabelecido, o emissor envia um comando MAIL indicando o envio de um mail. Se o receptor pode aceitar mail para este destinatário ele responde com um OK; senão ele responde rejeitando este destinatário (mas não a transação inteira de mail). Quando o destinatário foi negociado, o emissor envia os dados do mail, terminando com uma seqüência especial. Se o receptor processar os dados do mail ele responde com um OK. O diálogo é propositalmente de passos fechados e um por vez .

#### FIGURA 4.1: MODELO DE USO DO SMTP

O SMTP provê mecanismos para a transmissão de mail; diretamente do host do usuário remetente para o host do usuário destinatário quando os 2 hosts estão conectados ao mesmo serviço de transporte, ou via um ou mais servidores de SMTP de retransmissão quando os hosts origem e destino não estão conectados ao mesmo serviço de transporte.

Para estar apto a prover a capacidade de retransmissão o servidor SMTP precisa suprido com o nome do host destino final tal como o nome da caixa postal do destino.

O argumento para o comando MAIL é um caminho reverso, o qual especifica quem é a origem do mail. O argumento para o comando RCPT é um caminho de forward, que especifica para quem é o mail. O caminho de forward é a origem da rota, enquanto o caminho reverso é a rota de retorno (que pode ser usada para retornar a mensagem para o remetente quando um erro ocorrer com a mensagem retransmitida).

Quando a mesma mensagem é enviada para múltiplos destinatários o SMTP tenta a transmissão de somente uma cópia dos dados para todos os destinatários no mesmo host destino.

Os comandos mail e respostas possuem uma sintaxe rígida. Respostas também têm códigos numéricos.

Comandos e respostas não são sensíveis a diferenças entre maiúsculas e minúsculas (case sensitive). Isto é, a palavra de comando ou resposta pode ser maiúscula, minúscula, ou misturada com maiúsculas e minúsculas. Note que isto não acontece com os nomes das caixas postais. Para alguns hosts o nome do usuário é case sensitive, e implementações do SMTP precisam cuidar para preservar os nomes dos usuários como eles aparecem nos argumentos da caixa postal. Nomes de hosts não são case sensitive.

Comandos e respostas são compostos de caracteres do conjunto de caracteres ASCII [1]. Quando o serviço de transporte provê um canal de transmissão de byte de 8 bits (octeto), cada caracter de 7 bits é transmitido justificado à direita em um octeto com o bit de mais alta ordem limpo para zero.

## **4.2 POP3**

[POP3] Alguns tipos de pequenos nodes (hosts) na Internet são freqüentemente na prática para manter um sistema de transporte de mensagens (MTS). Por exemplo, uma estação de trabalho pode não ter recursos suficientes

(ciclos/segundo, espaço em disco) em ordem para permitir um servidor SMTP [RFC82] e um sistema de distribuição de mail local associado para ser mantido residente e continuamente rodando. Analogamente, pode ser caro (ou impossível) manter um computador pessoal interconectado a uma rede com IP por longos períodos de tempo (o node está perdendo o recurso conhecido como “conectividade”).

A despeito disso, é muito bom estar apto a gerenciar mail nestes pequenos nodes, e eles freqüentemente suportam um agente usuário (UA) para ajudar as tarefas de manipulação de mail. Para resolver este problema, o node que pode suportar uma entidade MTS oferece um serviço de caixa postal para estes nodes menos dotados. O Protocolo de Correio – Versão 3 (POP3) pretende permitir que uma estação acesse dinamicamente uma caixa postal em um servidor (host) de uma boa maneira. Normalmente, isto significa que o protocolo POP3 é usado para permitir que uma estação de trabalho receba mail que o servidor está guardando para ela.

O POP3 não pretende prover operações de manipulação de mail extensivas no servidor; normalmente, o mail é baixado e então apagado. Um protocolo mais avançado (e complexo) é o IMAP4, discutido no [IMAP].

Para o redator deste RFC, o termo “host cliente” refere-se a um host fazendo uso do serviço de POP3, enquanto o termo “host servidor” refere-se a um host que oferece o serviço de POP3.

### **Uma pequena divagação**

Esta RFC não especifica como um host cliente insere mails no sistema de transporte, no entanto um processo compatível com a filosofia desta RFC é apresentada a seguir:

Quando o agente usuário em um host cliente deseja inserir uma mensagem no sistema de transporte, ele estabelece uma conexão SMTP com o host transmissor e envia todo o mail para ele. Este host transmissor pode ser, não necessariamente, o host servidor de POP3 para o host cliente. Claro que, o host transmissor precisa aceitar mail para distribuir para endereços de destino arbitrários, que funcionalmente não são requeridos para todos os servidores SMTP.

### ***Operação básica***

Inicialmente, o host servidor inicia o serviço de POP3 escutando a porta TCP 110. Quando um host cliente deseja usar o serviço, ele estabelece uma conexão TCP com o host servidor. Quando a conexão é estabelecida, o servidor POP3 envia uma saudação. O cliente e o servidor POP3 então trocam comandos e respostas (respectivamente) até que a conexão é fechada ou cancelada.

Os comandos no POP3 consistem em palavras chave case sensitive, podendo ser seguidos de um ou mais argumentos. Todos os comandos são terminados por um par de CRLF. Palavras chave e argumentos consistem em caracteres ASCII imprimíveis e são, cada um, separados por um caracter espaço simples. Estas palavras chave possuem 3 ou 4 caracteres de tamanho. Cada argumento pode ter até 40 caracteres.

Respostas no POP3 consistem de um indicador de estado e uma palavra chave possivelmente seguida de uma informação adicional. Todas as respostas são terminadas por um par de CRLF. Estas respostas podem ter até 512 caracteres de tamanho, incluindo o terminados CRLF. Existem correntemente dois indicadores de estado: positivo (“+OK”) e negativo (“-ERR”). Os servidores precisam enviar “+OK” e “-ERR” em maiúsculo.

As resposta para certos comandos são multilinha. Neste caso, depois de enviar a primeira linha de uma resposta e um par de CRLF, qualquer adicional é enviada, cada uma terminada por um par de CRLF. Quando todas as respostas forem enviadas, uma linha final é enviada, consistindo de um octeto terminador (código decimal 046, “.”) e um par de CRLF. Se alguma linha de uma resposta multi linha inicia com um octeto terminador, a linha é preenchida por um octeto terminador pré estabelecido para esta linha de resposta.

Portanto uma resposta multilinha é terminada com cinco octetos de “CRLF.CRLF”. Quando examina uma resposta multilinha, o cliente checa para ver se a linha inicia com o octeto terminador. Se sim, e se outros octetos que seguem o CRLF, o primeiro octeto de uma linha (o octeto terminador) está destacado. Se sim, então se o CRLF segue imediatamente o caracter terminador, então a resposta para o servidor POP é terminada e a linha contendo “.CRLF” não é considerada parte da resposta multilinha.

Uma sessão POP3 progride através de um número de estados durante seu tempo de vida. Uma vez que a conexão TCP foi aberta e o servidor POP3 enviou as saudações, a sessão entra no estado de Autorização. Neste estados, o cliente precisa se identificar para o servidor POP3. Uma vez que o cliente fez isto com sucesso, o servidor aloca os recursos associados com a caixa postal do cliente, e a sessão entra em estado de transação. Neste estado, o cliente requisita ações da parte do servidor POP3. Quando o cliente tem concluído o comando QUIT, a sessão entra em estado de UPDATE. Neste estado, o servidor POP3 libera qualquer recurso disponível no estado de transação e diz “tchau” . Então a conexão TCP é fechada.

Um servidor deve responder para um comando desconhecido, não implementado, ou sintaticamente inválido com um indicador de estado negativo. Um servidor deve responder para um comando de conclusão quando a sessão está num estado incorreto com um indicador estado negativo. Não existem métodos gerais para

um cliente distinguir entre um servidor que não implementa um comando especial e um servidor que é incapaz de processar o comando.

Um servidor POP3 pode ter um temporizador para desconectar-se automaticamente por inatividade. Tal temporizador deve ser de, pelo menos, 10 minutos de duração. A recepção de qualquer comando vindo do cliente durante este intervalo deveria reiniciar o temporizador. Quando o temporizador vence, a sessão não entra no estado de UPDATE – o servidor poderia fechar a conexão TCP sem remover nenhuma mensagem ou enviar qualquer resposta para o cliente.

### **4.3 XML**

[XML] XML foi desenvolvida pelo XML Working Group (originalmente conhecida como SGML Editorial Review Board) formado sob o patrocínio do W3C em 1996. Foi presidido por Jon Bosak da Sun Microsystems com uma participação ativa num Grupo de Interesse Especial (previamente conhecido como o SGML Working Group) também organizado pelo W3C.

#### **Os objetivos do XML:**

- Deve ser diretamente usável sobre a Internet
- Deve suportar a grande variedade de aplicações
- Deve ser compatível com SGML
- Deve ser fácil de escrever programas que processem documentos XML
- número de características opcionais no XML é para ser mantido no absolutamente no mínimo, idealmente zero.
- Documentos XML deveriam ser legível por humanos e razoavelmente claros.
- desenho do XML poderia ser preparado rapidamente.
- desenho do XML deve ser formal e conciso.
- Documentos XML devem ser fácil de serem criados.

- Falta de coesão em XML é de mínima importância.

Esta especificação, junto com os padrões associados (Unicode e ISO/IEC 10646 para caracteres, RFC 1766 da Internet para tags de identificação de linguagem, ISO 639 para códigos de nome de linguagem, e ISO 3166 para os códigos de nome de país, provêm toda a informação necessária para entender XML versão 1.0 e construir programas de computadores para processá-la.

### **Vantagens e desvantagens do XML**

Vantagens:

- Extensível
- Não há grupos fixos de tags
- Bom para processamento de informações
- Adequado para rede
- Processamento pode ser feito ao lado do cliente

Desvantagens:

- Difícil de ser programada
- Falta de Browsers que suportem esta tecnologia

### **4.4 PHP**

[PHP] PHP, que é sigla para "PHP: Hypertext Preprocessor", é uma linguagem de macros embutida em HTML, multiplataforma e executada do lado do servidor. Muito de sua sintaxe é baseada em C, Java e Perl com algumas propriedades específicas da linguagem PHP. O objetivo da linguagem é permitir a desenvolvedores WEB escrever páginas dinâmicas geradas rapidamente.

O que isso significa ? Um exemplo:

Exemplo 1-1. Um exemplo introdutório

```
<html>
<head>
<title>Example</title>
</head>
<body>
<?php echo "Hi, I'm a PHP script!"; ?>
</body>
</html>
```

Note como isso é diferente de scripts CGI escritos em outras linguagens como Perl ou C --- ao invés de escrever um programa com um vários comandos para imprimir HTML, escreve-se um script HTML com algum código inserido para fazer alguma coisa (nesse caso, imprimir algum texto). O código PHP é delimitado por tags iniciais e finais que permitem pular para dentro e para fora do modo PHP.

O que distingue o PHP de algo como Javascript do lado do cliente é que o código é executado no servidor. Se você tivesse um script similar ao acima em seu servidor, o cliente receberia os resultados da execução desse script, sem nenhum modo de determinar qual é o código fonte. Você pode inclusive configurar seu servidor para processar todos os seus arquivos HTML como PHP, e então não haverá nenhum modo dos usuários descobrirem que você usa código.

No nível mais básico, o PHP pode fazer qualquer coisa que qualquer outro programa CGI pode fazer, como coletar dados de formulário, gerar conteúdo de página dinâmico, ou enviar e receber cookies.

Talvez a mais forte e mais significativa característica do PHP é seu suporte a uma ampla variedade de banco de dados. Escrever uma página que consulte um banco de dados é incrivelmente simples.

• Adabas D	• FrontBase	• Sybase
• Ingres	• PostgreSQL	• IBM DB2
• Oracle (OCI7 and OCI8)	• FilePro (read-only)	• MySQL
• DBase	• MSQL	• Velocis
• InterBase	• Solid	• Informix
• Ovrimos	• Hyperwave	• ODBC
• Empress	• Direct MS-SQL	• Unix dbm

**QUADRO 4.4: BANCOS DE DADOS SUPORTADOS PELO PHP**

O PHP também tem suporte para conversar com outros serviços usando protocolos como IMAP, SNMP, NNTP, POP3, HTTP e outros inúmeros. Pode-se também abrir cockets raw de rede e interagir usando outros protocolos.

## Capítulo 5

### Escolha das Tecnologias

#### 5.1 Linguagem de Programação

A linguagem escolhida para implementação do projeto foi Delphi 6. As units do Delphi, nada mais são de que uma interface amigável para as chamadas as API's nativas de Socket do Windows. Desta forma conseguimos trabalhar nativamente com os Sockets do windows através de uma linguagem de alto nível e poderosa como o Pascal, manipulando facilmente POP3 e SMPT. POP3 e SMTP são dois protocolos simples de envio e recebimento de mensagens eletrônicas, definidas na Internet. Utilizando Delphi podemos implementar o cliente e o servidor , usando os componentes TserverSocket e TclientSocket , encontrados na paleta de Internet. Estes componentes permitem codificar em poucas linhas códigos complexos.

#### ***Características e Benefícios do Delphi 6***

O Borland Delphi 6 [DELPHI] é o primeiro ambiente RAD (Desenvolvimento Rápido de Aplicações) para Windows que suporta totalmente futuros *Web Services*. Os *Web Services* são o próximo passo em desenvolvimento de aplicações na Internet e os desenvolvedores estão sofrendo pressão para converter as aplicações corporativas aos padrões abertos para que, mesmo tendo infra-estruturas de TI diferentes, duas empresas possam conduzir atividades de *e-business* em conjunto. Os desenvolvedores individuais ou corporativos podem criar aplicações de *e-business* com rapidez e facilidade. Os principais recursos do produto incluem *BizSnap*, *WebSnap* e *DataSnap*, que permitem aos desenvolvedores criar aplicações que simplifiquem a integração B2B com suporte total a XML, SOAP e WSDL.

O suporte a *Web Services* padrão da indústria do Delphi 6 possibilita integração imediata e contínua com novos *Web Services* baseados em plataformas como *.Net* e *BizTalk* da *Microsoft*, e *ONE* da *Sun Microsystems*. Além disso, quando associado ao *Borland Kylix*, o primeiro ambiente RAD nativo para o sistema operacional Linux, os usuários do Delphi 6 podem criar aplicações de fonte única tanto para Windows quanto para Linux.

A nova estrutura do Delphi inclui *BizSnap*, *WebSnap* e *DataSnap*, que ajudarão os usuários a criar aplicações do lado do cliente e do servidor que suportam as especificações de *Web Services* através de um conjunto totalmente integrado de ferramentas visuais, tecnologia avançada de compilador e componentes de software reutilizáveis. O Delphi 6 foi projetado para acelerar o desenvolvimento e a disponibilidade de soluções no mercado, simplificando a integração B2B.

### **Os novos recursos**

- *BizSnap*: uma plataforma RAD para desenvolvimento de *WebServices*, que simplifica a integração B2B criando conexões e *Web Services* baseados em XML/SOAP;
- *WebSnap*: uma estrutura de desenvolvimento de aplicações Web baseada em componentes que suporta os principais Servidores de Aplicações Web, inclusive *Apache*, *Netscape* e *Microsoft Internet Information Services (IIS)*;
- *DataSnap*: que permite aos clientes criar *middleware* de alto desempenho capazes de trabalhar com *Web Services*, possibilitando fácil conexão de qualquer serviço ou aplicação de cliente com os principais bancos de dados, por exemplo, Oracle, MS-SQL Server, Informix, IBM, DB2, Sybase e InterBase, através de Serviços Web padrão da indústria e XML, DCOM ou CORBA.

- Novas palhetas foram adicionadas, como as palhetas indy, para a construção de aplicações Delphi com Internet, novos componentes foram adicionados na palheta additional.

## **5.2 Banco de Dados**

Foram avaliados vários bancos de dados, através de versões de demonstração, testes e conversas com os usuários dos mesmos. Procuramos levar em conta performance, preço e praticidade. Dentre eles Oracle, SqlServer e InterBase. Analisando os benefícios de cada um, concluímos que a melhor opção seria o InterBase, uma vez que este possui facilidades para trabalhar com a Internet e oferece versões gratuitas - um dos requisitos primordiais para o sucesso de nosso produto, uma vez que ele se propõe a ser acessível a pequena e médias empresas. Outro determinante foi o desempenho: uma vez que o Oracle e o SqlServer são mais indicados para grandes aplicações (que não é nosso caso) também necessitam de mais memória RAM, mais disco e maior poder de processamento.

### ***Características e Benefícios do Interbase***

InterBase [IB] é feito idealmente para desenvolvedores de aplicações que querem um banco de dados realmente encaixado. A pouca necessidade de manutenção (O DBA não é requerido) do InterBase te dão a tranqüilidade de que seus clientes mal notarão que ele está lá.

O Interbase possui uma alta performance tanto em aplicações críticas como em aplicações menores enquanto se mantém com preço acessível (IB6 será free) e fácil de usar para desenvolvedores de aplicações menores.

O Servidor InterBase é construído em uma arquitetura multi-geracional(MGA). O MGA provê um dispositivo versionig único que assegura alta disponibilidade para usuários que suportam decisões (support-decision). Servidores de

banco de dados suportam tradicionalmente o modelo de interação de banco de dados On-Line Transaction Processing (OLTP), caracterizado por um grande volume de transações simples e curtas. Enquanto o dispositivo versioning do InterBase suporta essas transações curtas, do tipo OLTP, o InterBase supera a competição em aplicações reais porque também gerencia ao mesmo tempo transações de longa duração do tipo support-decision.

A arquitetura SuperServer aumenta a performance e otimiza o uso de recursos de sistemas, especialmente por um grande número de usuários. Isto te permite ter mais clientes no seu servidor, e ainda aumentar a velocidade de suas repostas. Este é um servidor multi-threaded compartilhado, que provê grande performance.

InterBase é instalado facilmente com um único comando. Ele tem uma pequena footprint, então você não precisa se preocupar em ter bastante espaço livre em disco e não precisará estabelecer centenas de parâmetros de sintonia. O InterBase foi preparado para requerer pouca manutenção, portanto ele otimiza suas transações pra você.

Sinalizadores de eventos tornam possível a existência de um banco de dados ativo, avisando automaticamente as "partes interessadas" quando certas mudanças acontecem. Tudo isso é feito sem polling constante no banco de dados, portanto isto não limita os recursos do sistema.

Funções Definidas pelo Usuário (UDFs) provêm meios de aumentar as capacidades analíticas do InterBase através da criação de funções habituais de negócios. UDFs são um código reutilizável e asseguram a integridade e confiabilidade dos dados. Da mesma maneira, UDFs podem ser usadas para chamar aplicações externas ao banco de dados InterBase estabeleceu o padrão industrial com o lançamento de seu primeiro produto em 1986, quando armazenou som, imagem,

gráficos e informações binárias diretamente no banco de dados usando os seus tipos de dados BLOB.

InterBase também suporta os *c* multidimensionais usados extensivamente em aplicações financeiras e científicas. Armazenando arrays multidimensionais com até 16 dimensões em um único campo no banco de dados o InterBase simplifica o design da aplicação e aumenta a performance..

Banco de dados distribuídos para flexibilidade da aplicação. Quando você precisar mover a sua solução desktop de banco de dados para uma configuração client/server ou aumentar as suas aplicações de grupo de trabalho para servir a um ou mais departamentos, o InterBase é ideal, pois é feito para ambientes de banco de dados distribuídos.

InterBase é um verdadeiro servidor de bancos de dados distribuídos SQL que permite que cada query do sistema de banco de dados retorne a informação para qualquer outro servidor InterBase

InterBase também gerencia transações com servidores múltiplos de maneira rápida e fácil.

Ele inclui o processamento de transações com commit em duas fases, o que assegura que as atualizações sejam feitas sem intervenção da aplicação. Toda vez que uma transação abrange dois ou mais servidores de banco de dados o InterBase primeiro investiga os servidores participantes para assegurar de que eles estejam em atividade e rodando, então envia o comando commit para completar a transação.

Para soluções em aplicações de missão crítica, o InterBase oferece total compatibilidade com SQL-92.

Interbase utiliza tecnologia de travamento otimista para proporcionar grande taxa de uso de operações de banco de dados para clientes. O Interbase implementa travamentos a nível de linha reais para restringir mudanças somente nos registros do banco de dados que um cliente modifica. Diferente de travamentos a nível de página, que restringe qualquer dado arbitrário que estiver armazenado fisicamente próximo no banco de dados. Travamentos a nível de linha permitem múltiplos clientes atualizarem dados em uma mesma tabela sem conflito, resultando em menor serialização das operações de bancos de dados.

Além do Interbase possuir a flexibilidade de plataformas (Windows, Linux, Unix, Solaris, NetWare), também é de Distribuição livre – partir da versão 6 o InterBase será open-source e free, o que significa que o custo de seu projeto cairá! Consequentemente seus clientes ficarão bem satisfeitos.

## Capítulo 6

### Desenvolvimento

#### 6.1 Escolha do método

Para o projeto Global EDI, foi escolhido o método de análise orientado a objetos, por ser este o método com o qual os autores mais se familiarizam, e por outros motivos (em especial o primeiro) a seguir relacionados:

- A linguagem escolhida foi Delphi que é uma linguagem que, não somente suporta implementações OO, como também as torna muito mais fácil.
- OO é um método que melhor representa o mundo real e a Internet é um mundo bastante real e funcional.
- Concentra-se nos aspectos essenciais do objeto sem detalhamento, focando em suas características e o que ele faz;
- Impede que um sistema se torne tão interdependente que uma pequena alteração ou implementação resulte em grandes alterações em sua estrutura;
- Combina estrutura e comportamento em um único objeto;
- Compartilha elementos estruturais e de comportamento com objetos de níveis inferiores;
- Enfatiza a estrutura de objetos ao invés da estrutura de procedimentos, ou seja, o que o objeto é e não como ele é utilizado.

## 6.2 Descrição do GLOBAL EDI

GLOBAL EDI é uma proposta para uma ferramenta que tem como objetivos simplificar, automatizar, baratear e popularizar a tecnologia do EDI.

Uma das grandes dificuldades numa implantação de EDI é a negociação do lay-out.

A transmissão e recepção dos dados muitas vezes é manual: via disquetes, etc.

O funcionamento da tecnologia torna-se caro (e mais sujeito a erros) dado o número de processos com interação humana.

O GLOBAL EDI se propõe a:

- Diminuir a dificuldade na negociação do lay-out disponibilizando a ambos os lados, uma Interface de Configuração de lay-out (ICL).
- Transmitir tanto os lay-outs como os dados através da internet
- Economizar tempo (e dinheiro!)
- Assegurar a consistência e a segurança dos dados

**FIGURA 6.2: ESQUEMA GERAL DE FUNCIONAMENTO DO GLOBAL EDI**

Cada instância do GLOBAL EDI é um cliente de FTP ou de E-MAIL especializado que pode ser configurado para qualquer servidor destes serviços da internet de forma que ambos acessem o mesmo servidor e verifiquem sua “caixa postal”, baixe seus dados e em seguida transmita os seus dados para a caixa postal do seu(s) destinatário(s).

Uma vez instalado em um microcomputador com Microsoft Windows, inicia-se sua configuração. Cria-se pelo menos uma caixa postal a qual esta instância do sistema irá receber seus mails. Cria-se ao menos um parceiro com uma caixa postal para ele. Cria-se pelo menos um lay-out para esta caixa postal composto de ao menos um campo. Cria-se o arquivo que será gerado (ou de onde será lido) os dados do sistema local. Este arquivo terá o formato texto comma file.

No momento de transmissão, o GLOBAL EDI é chamado recebendo como parâmetro um identificador de lay-out e uma operação (que pode ser enviar ou receber), conecta-se à internet e realiza esta operação retornando um status.

### **6.3 Requisitos do Sistema**

Analisando as necessidades das empresas citadas no capítulo 2, podemos determinar as características que o sistema deve possuir para supri-las.

O meio de transmissão de dados, precisa suportar o protocolo TCP/IP, sobre o qual o as tecnologias SMTP e POP3 funcionam.

O sistema deve guardar as informações das outras empresas (ou unidades) com a qual irá trocar informações, incluindo os dados de sua(s) caixa(s) postal(is), mantendo também os lay-outs que serão usados com cada uma delas.

Os dados que serão transmitidos e os que forem recebido também precisam ser guardados até que sejam transmitidos ou sejam decodificados e exportados para a aplicação local, respectivamente.

O sistema precisa saber transmitir dados e verificar se há dados a serem recebidos, e recebe-los; assim como converte-los da aplicação local para o formato do lay-out (codificar) fazer o processo inverso (decodificar).

Precisa também possuir todas as interfaces de configuração e operação, permitindo que os recursos (incluindo monitoramento das transações de dados/meta-dados) sejam utilizados de forma mais amigável possível.

## 6.4. Diagrama de Classes

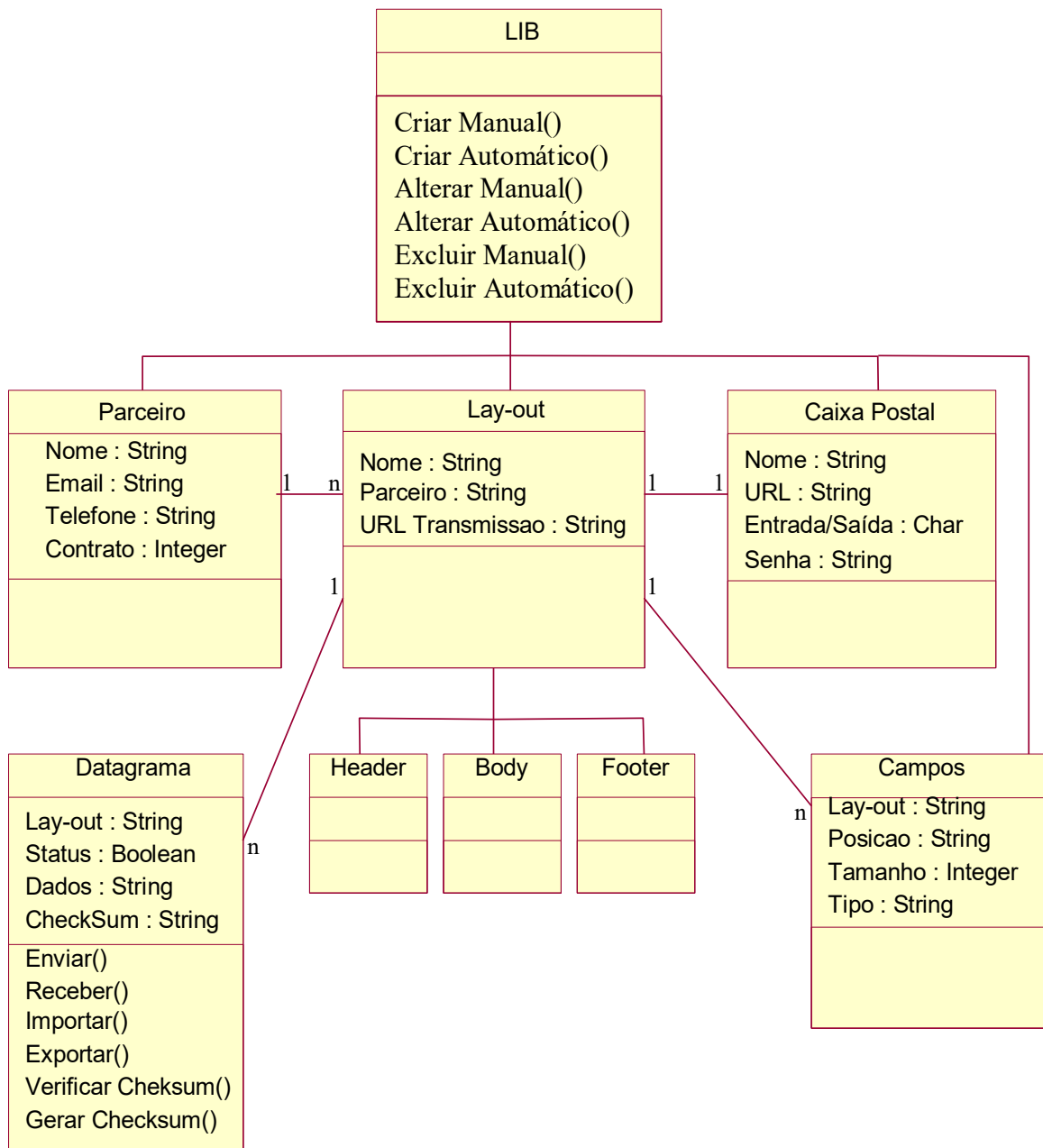


FIGURA 6.4: DIAGRAMA DE CLASSES

## 6.5 Dicionário de Dados

### 6.5.1 Lib

Biblioteca de métodos para ser herdada pelas outras classes.

#### Atributos

Lib não possui atributos

#### Regras

- As classes de cadastramento (parceiro, layout, caixa postal e campo) usarão (por herança) os métodos desta classe.

#### Operações

Todas as operações desta classe recebem uma mensagem de entrada chamada "classe indicada" que conterà a classe do objeto a ser manipulado. Outras mensagens são descritas em cada operação.

***Incluir Manual:*** Através de uma interface com o usuário no formato de uma janela com os campos e nomes o sistema receberá os dados que o operador digitar e criará uma instância da classe indicada. No banco de dados será executado:

```
INSERT INTO <classe_indicada> VALUES
(<valor_1o._campo>, <valor_2o._campo>,
<valor_3o._campo>, ..., <valor_no._campo>);
```

***Incluir Automático:*** quando o sistema receber em sua caixa postal um e-mail formatado com o padrão do GLOBAL EDI (contendo os atributos e dados para a criação de uma instância da classe indicada) em que uma outra instância do sistema está se identificando como parceiro desta instância local, então esta instância criará uma instância indicada procedendo assim como no Incluir Manual. Na primeira oportunidade de interação com o usuário, o GLOBAL EDI confirmará a inclusão de

um novo objeto da classe indicada, permitindo que esta operação possa ser rejeitada localmente.

**Alterar Manual:** Usando uma interface com o usuário no formato de uma janela com os campos e nomes o sistema disponibilizará os atributos de uma instância para que o operador e possa alterá-los. No banco de dados será executado:

```
UPDATE <classe_indicada>
    SET <campo_1> = <novo_valor_campo_1>,
    SET <campo_2> = <novo_valor_campo_2>,
    SET <campo_3> = <novo_valor_campo_3>,
    ...
    SET <campo_n> = <novo_valor_campo_n>
WHERE <campo_identificador> = <identificação_objeto>
```

**Alterar Automático:** quando o sistema receber em sua caixa postal um e-mail formatado com o padrão do GLOBAL EDI (contendo os atributos e dados para a alteração de uma instância de parceiros) em que uma outra instância do sistema está se identificando como parceiro desta instância local, então esta instância do sistema alterará uma instância da classe indicada procedendo assim como no Alterar Manual. Na primeira oportunidade de interação com o usuário, o GLOBAL EDI confirmará a alteração do objeto da classe indicada, permitindo que esta operação possa ser rejeitada localmente.

**Excluir Manual:** Usando uma interface com o usuário no formato de uma janela com os campos e nomes o sistema disponibilizará os atributos de uma instância para que o operador e possa destruir esta instância. No banco de dados será executado:

```
DELETE FROM <classe_indicada>
    WHERE <campo_identificador> = <identificação_objeto>
```

**Excluir Automático:** quando o sistema receber em sua caixa postal um e-mail formatado com o padrão do GLOBAL EDI (contendo a identificação de um objeto da classe indicada) em que uma outra instância do sistema está se identificando como parceiro desta instância local, então esta instância local do sistema destruirá o objeto da classe indicada procedendo assim como no Excluir Manual. Na primeira oportunidade de interação com o usuário, o GLOBAL EDI confirmará a destruição do parceiro, permitindo que esta operação possa ser rejeitada localmente.

### **6.5.2 Parceiro**

Armazena os dados das empresas participantes da troca de informações.

#### **Atributos**

- **Nome:** nome que identificará a empresa parceira
- **E-mail:** e-mail da empresa
- **Telefone:** telefone da empresa
- **Contrato:** número do contrato da empresa

#### **Regras**

- Todas as empresas que participarão da comunicação de dados devem estar cadastradas e possuir um número de contrato
- Cada parceiro pode possuir n lay-outs.
- A própria empresa onde esta instância do Global EDI está instalada deve ser cadastrada como um parceiro e com o atributo Contrato 0 (zero), deve haver ao menos 1 lay-out e uma caixa postal de entrada para este parceiro.

#### **Operações**

Apenas os métodos herdados de Lib.

### 6.5.3 Caixa Postal

Armazena os dados das caixas postais – informação necessária para o envio e a recepção de dados.

#### Atributos

- **Nome**: denominação da Caixa Postal
- **URL**: endereço da caixa postal (ex: <mailto:edi2@empresa.com>)
- **Senha**: senha da caixa postal - usada apenas se a caixa for de entrada.
- **Entrada/Saída**: um caracter que sinaliza se a caixa postal é de entrada ou saída. As caixas que forem de entrada devem ser verificadas, então a usar-se-á a informação no atributo senha. As caixas de saída serão destinatários para onde os datagramas serão enviados.

#### Regras

- Um parceiro pode ter uma ou mais Caixas Postais
- Cada Caixa Postal deve possuir um lay-out com o mesmo atributo URL

#### Operações

Apenas os métodos herdados de Lib.

### 6.5.4 Lay-Out

Armazena o formato dos dados que serão codificados e decodificados para a troca de informações.

#### Atributos

**Nome**: denominação com o qual o lay-out será identificado

**Parceiro**: Nome do Parceiro do qual este lay-out será usado para se comunicar

**URL**: Transmissão: endereço da caixa postal (ex: <mailto:edi3@empresa.com>)

## Regras

- Vários Datagramas serão criados usando-se um lay-out.
- Cada lay-out pode possuir até 3 instâncias com o mesmo Nome, cada uma de um tipo: *HEADER*, *BODY*, *FOOTER*
- Um lay-out pode ter de 1 a n campos
- Deve haver uma Caixa Postal para cada Lay-out
- Deve haver um Parceiro para cada Lay-out

## Operações

Apenas os métodos herdados de Lib.

### 6.5.5 Datagrama

Armazena os dados codificados para a transmissão ou recém recebidos para a decodificação.

## Atributos

- **Lay-out**: Identificação do Lay-Out que foi usado para codificar este datagrama
- **Status**: Sinaliza se este datagrama foi recebido ou será transmitido
- **Dados**: Uma seqüência de bytes de tamanho máximo de 500kb.
- **Checksum**: Um inteiro de 4 bytes que é a soma de cada um dos caracteres ASCII que compõe o campo Dados.

## Regras

- Os campos compõe cada um dos Datagramas.
- Serão criados de 0 a 1 datagrama usando os lay-outs dos tipos Header e Footer; e ao menos um datagrama usando o lay-out do tipo body.

## Operações

**Enviar:** recebe o identificador do datagrama a ser transmitido. Estabelece conexão com a internet, lê o datagrama, envia o datagrama via smtp e desfaz a conexão. Retorna um status que pode ser “OK” ou nulo.

**Receber:** Estabelece conexão com a internet, verifica se há alguma mensagem na(s) caixa(s) postal(is) de recepção usando o pop3. Se não houver retorna um status de “VAZIO”. Caso contrário recebe a(s) mensagem(ns) e grava o(s) datagrama(s), retornando um status “OK”. Retorna nulo se houver mensagens, mas não conseguir baixá-las.

**Importar(Codificar):** codifica os dados mapeando os campos do arquivo da aplicação para o datagrama, gera um checksum deixando-o pronto para ser enviado.

**Exportar(Decodificar):** valida o checksum, se for inválido retorna um status de nulo, caso contrário decodifica os dados do datagrama gerando um arquivo no formato “comma file” (ou arquivo de vírgulas).

**Validar Checksum:** verifica se a soma dos códigos ASCII dos caracteres que compõe os dados do datagrama é igual ao valor inteiro armazenado nos últimos 4 bytes do datagrama.

**Gerar Checksum:** soma dos códigos ASCII dos caracteres que compõe os dados do datagrama e os adiciona em forma de um número inteiro de 4 bytes ao final do datagrama.

### 6.5.6 Campos

Armazena a informação de cada campo de dados contidos no lay-out dos dados.

**Atributos**

- **Lay-out**: Identificação do Lay-Out que foi usado para codificar este datagrama
- **Posição**: Posição ordinal em que o campo ocupa na composição do lay-out.
- **Tamanho**: Tamanho em bytes do campo.
- **Tipo**: Tipo de dados do campo (inteiro, string, caracter, decimal, lógico)

**Regras**

- Será sugerido que os campos sejam numerados na posição de 10 em 10 números, para facilitar possíveis inserções de campos trocas de posições.
- Podem haver de 1 a n campos para cada lay-out.

**Operações**

Apenas os métodos herdados de Lib.

## Capítulo 7

### Conclusões

#### 7.1 Sobre os rumos traçados

Apesar da proposta ser relativamente modesta em comparação a tudo o que vislumbram o EDI e o B2B, pode-se reconhecer que as experiências resgatadas à realidade de hoje, confirmam a aplicabilidade das idéias aqui apresentadas.

Um ponto interessante foi levantar as necessidades provenientes de um ambiente prático, onde qualquer falta necessariamente gera um prejuízo.

A reunião das informações sobre novas e consagradas tecnologias possibilitou validarmos a aplicabilidade dos conceitos e tecnologias abordados.

#### 7.2 O Futuro

A implementação da aplicação modelada neste projeto é uma realidade totalmente tangível, considerando que, as tecnologias e as ferramentas necessárias são relativamente de fácil acesso.

Pode ser acrescentado aos meios de comunicação utilizados para a transmissão e recepção de dados, (além do e-mail), a tecnologia do FTP, o que aumentaria a usabilidade do sistema.

#### 7.3 Considerações Finais

Ao verificar-se a simplicidade da proposta perto do que se espera de um produto final e dos conceitos abordados, pode-se pensar que o passo dado é pequeno

demais. Porém, tendo em vista o contexto didático imediato do trabalho desenvolvido, reconhecemos que o conteúdo obtido deste projeto não só pode ser utilizado comercialmente, como também abre um novo horizonte em soluções para problemas constantes nas empresas.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [ARTHA] ARTHA INFORMÁTICA LTDA.. Site oficial. Disponível na Internet.  
<http://www.artha.com.br>. Abril 2001
- [DELPHI] – CLUBE DELPHI. Delphi 6 – Novos Recursos. Disponível na Internet.  
<http://www.clubedelphi.com.br/principart.htm>. Maio 2001.
- [EXTOL] MONTENEGRO E., EXTOL Informática. Disponível na Internet.  
<http://www.extol.com.br>. Abril 2001.
- [FTP] Postel, Johnathan. B.; Reynolds Jane., File Transfer Protocol Specification, RFC 959.  
Disponível na Internet. <http://www.ietf.org/rfc/rfc0959.txt>. Outubro 1985.
- [IB] Cantu, Carlos. Características do Interbase. Disponível na Internet.  
<http://www.warmboot.com.br/ib/>. Abril 2001.
- [IMAP] Crispin, M. Internet Message Access Protocol – Version 4, RFC 1730. University of Washington. Disponível na Internet. <http://www.ietf.org/rfc/rfc1730.txt>. Dezembro 1994.
- [LEME] Leme, M. A. S. M. Contrato Social do Instant BBS - Contato - Serviços de Informações por Computador Ltda. ME, Maringá, Maio 1996.
- [PHP] The PHP Group – Apache Software Foundation. Disponível na Internet.  
<http://www.php.net>. Abril 2001.
- [PHPB] PHP Builder. The PHP's Manual. Disponível na Internet.  
<http://www.phpbuilder.com>. Abril 2001.
- [PHPW] PHP Wizard. Disponível na Internet. <http://www.phpwizard.com>. Abril 2001.
- [POP3] Myers, J.G.; Marshall, T. R., Post Office Protocol - Version 3”, RFC 1939.  
Disponível na Internet. <http://www.ietf.org/rfc/rfc1939.txt>. Maio/1996.

- [SEARLE] Searle, S. J., A Brief History of Character Codes in North America, Europe, and East Asia. Sakamura Laboratory. University Museum. University of Tokyo. Disponível na Internet, <http://tronweb.super.co.jp/characcodehist.html>. 1999.
- [SILVA] L. M. C.; Santos, N. M. Utilização de Técnicas de IA para Implementação de um Agente Operador de Sistema de Acesso Remoto, Trabalho de Graduação, Universidade Estadual de Maringá, Paraná, Brasil, Dezembro 1996).
- [SINTEL] Sintel Informática Ltda. Disponível na Internet <http://www.sintel.com.br>. Abril 2001.
- [SML] SML Informática. Disponível na Internet. <http://www.smlinfo.com.br>. Abril 2001.
- [SMTP] Postel, J. B., Simple Mail Transfer Protocol, RFC 821, University of Southern California. Disponível na Internet. <http://www.ietf.org/rfc/rfc0821.txt>. Agosto 1982.
- [XML] Bray, T.; Paoli J.; Sperberg-McQueen, C. M.; Maler E., Extensible Markup Language (XML) 1.0 (Second Edition)” - W3C Recommendation. Disponível na Internet. <http://www.w3.org/TR/REC-xml>. Outubro 2000.

## GLOSSÁRIO

***RFC*** – “Request For Comments” – Pedido de Comentários – documentação oficial dos padrões e das tecnologias utilizadas na implementação da internet. Traz conceitos, descrições de funcionamento e de implementação.

***ASCII*** - American Standard Code for Information Interchange - Código Padrão Americano para Troca de Informações.

***Comma File*** - Arquivo de Vírgulas: arquivo do tipo texto em que cada linha (separada por CR+LF) contém campos separados por vírgulas(.). Os campos dos tipos Strings e Char devem ser cercados de aspas (").

***Imaging*** - Gerenciamento Eletrônico de Documentos, digitalização e indexação de Documentos, Plantas, Fotos, Slides, Negativos, Fitas de Áudio, Fitas de Vídeo, Microfilmes, Microfichas e Radiografias

***ERM*** - (enterprise resource management) Alguns analistas preferem este termo e fazem uma distinção sutil entre ERM e ERP. ERM abrange contabilidade, RH e gestão de materiais; ERP é ERM mais aplicativos.

***ERP*** - O software ERP (enterprise resource planning - gestão empresarial) é um conjunto de aplicativos que automatiza os departamentos de finanças e recursos humanos e ajuda os fabricantes a lidarem com o processamento de pedidos e o cronograma de produção, entre outros aspectos. Sistemas ERP como o R/3 da SAP AG são notoriamente complexos e a instalação do software, com frequência, obriga os usuários a mudarem seus processos internos. Os fornecedores de ERP estão se diversificando e oferecendo aplicativos como gestão de supply chain e automação de força de vendas, mas ainda se discute se estas funções se encaixam na categoria ERP.

**Supply Chain** - conjunto de empresas que atendem a um determinado mercado com seus produtos, produzindo e transportando desde as matérias-primas básicas até os produtos para os clientes finais. A tradução do termo supply chain para o português é cadeia logística. O termo em inglês reforça a imagem de integração dos diversos componentes da cadeia, dos elos da corrente, que são as diversas empresas consideradas individualmente (chain = corrente). Em outras palavras, é a cadeia de abastecimento constituída pelo relacionamento entre todos os pontos da produção, desde os fornecedores de matéria-prima até o consumidor final.

**Workflow** - Automatiza o roteamento de documentos e dados na seqüência de usuários apropriada.

**WEB** – Referência usual de WWW (World Wide Web) – serviço na internet de páginas servido através da porta 80 do servidor e que permite ao usuário acessar texto, imagens, sons e outros protocolos e também permitindo que as informações sejam publicadas e vistas em arquivos em documentos interativos e que podem ser **hiperlinkados**, assim como um hipertexto.

**W3C** - World Wide Web Consortium – Consórcio WWW. Um hold de instituições de naturezas, diversos que se uniram para um interesse em comum: padronizar e desenvolver a internet

**Unicode** - Apêndice A.

**Script** – Pseudo linguagem de programação interpretada e com número reduzido de mnemônicos, porém de fácil utilização.

**Cookies** – Informações depositadas no disco do usuário que guarda meta dados do que o usuário acessou, para serem utilizadas posteriormente pelo servidor.

**Lay-Out** – Padrão, (conjunto de) formato de dados.

**Units** - unidades de programas utilizadas pelo Delphi.

**Free** – Gratuito, de livre distribuição

**Arrays** – Matriz. Estrutura de dados composta de várias ocorrências de um determinado tipo ou (outra) estrutura de dados. Uma combinação de elementos de dados. Quando a matriz tem somente uma dimensão, é chamada de vetor.

**Query** – Componente utilizado pelo Delphi para acessar a base de dados através de instruções SQL.

**Commit** – Comando sql que confirma uma operação que modifica o banco de dados.

**Open-source** – Fonte aberto, software cujo fonte é distribuído livremente

**Versionig** – controle de versões

**Footprint** – Base, A quantidade de espaço geográfico ocupada por um objeto. Uma base de computador é a mesa ou o espaço no chão que ele ocupa. Uma base de satélite é a área da Terra coberta por seu *downlink*.

**Socket** – Soquete, uma interface de comunicações que permite que um aplicativo acesse um protocolo de rede, "abrindo um soquete" e declarando um destino. Os soquetes são muito conhecidos porque fornecem uma maneira simples de direcionar um aplicativo na rede (protocolo TCP/IP).

**Services** – programa de serviço, programa útil que é usado para atividades de rotina, processo servidor

**E-business** – (Electronic Business) – negócios eletrônicos – comumente usado para se referir a negócios feitos via internet.

**Polling** - Uma técnica de comunicações que determina quando um terminal está pronto para enviar dados. O computador interroga continuamente seus terminais conectados num esquema de rodízio. Se um terminal tem dados para enviar, ele envia de volta uma confirmação e a transferência começa.

**Serialização** – forma de executar as tarefas de forma serial, uma após a outra. Frequentemente costuma-se evitá-la para ganhar tempo com processos que podem ser paralelos (e não seriais).

## Apêndice A

### FTP

[FTP] Os objetivos do FTP são:

- Promover o compartilhamento de arquivos (dados e/ou programas de computador)
- Incentivar indireta ou implicitamente (via programas) o uso de computadores remotos
- Proteger o usuário de variações nos sistemas de armazenamento de arquivos nos hosts
- Transferir dados segura e eficientemente.
- 

Apesar do FTP ser utilizável diretamente pelo usuário no terminal, foi concebido principalmente para ser usado por programas.

## Anexo A

### UNICODE

Com a globalização mundial, a idéia de unir economias, houve uma maior importância a comunicação e ao alfabeto de outras nações, que muitas vezes não era entendido pelos computadores mundiais. Com a idéia de unir os códigos do alfabeto, foi criado o Unicode, que deverá se tornar um padrão mundial na área de codificação de caracteres. Unicode foi criado por um grupo de engenheiros internacionais e a sua criação visava acabar com as limitações impostas pelos padrões de codificação incompatíveis dos computadores: ASCII, JIS (japonês), GB-code (Chinês), BIG-S (Taiwan), KS-code (Coreana), ISCII (India), entre outros.

Com suporte para diversas línguas estrangeiras, o unicode deverá substituir no futuro, o ASCII e o EBCDIC, pois apresenta grandes vantagens em relação a estes.

O que é Unicode ?

O Unicode faz parte de um novo padrão chamado ISO 10646. O Unicode foi projetado com base no código ASCII, sem contudo, ficar limitado ao alfabeto latino: O Unicode é capaz de codificar todos os caracteres presentes em qualquer uma das linguagens escritas existentes no mundo atualmente. Para tanto ele trabalha com um código de 16-bit ao invés dos 7 bits do código ASCII. Esta pequena diferença no número de bits faz com que o Unicode consiga reconhecer mais de 65.000 caracteres, enquanto que o ASCII só é capaz de reconhecer 128. A ausência de códigos diferenciados para caracteres especiais faz do Unicode um código mais simples e ao mesmo tempo mais eficiente.

Os sistemas operacionais estão gradualmente adotando o Unicode. Atualmente o domínio ainda é do ASCII, mas é uma questão de tempo para o unicode se tornar o

novo padrão mundial em codificação de caracteres. Já existem sistemas operando com o unicode, como é o caso do Windows NT, e espera-se que a Microsoft, adote o unicode em todos os seus produtos, já que participa do consórcio criador do unicode, este consórcio ditará as normas do padrão Unicode no mundo, assim como as páginas de códigos serão organizados na tabela de valores. Também neste consórcio, a Apple e a IBM também deverão utilizar o Unicode em seus futuros sistemas.

A unificação de todos os caracteres mundiais resultará em grandes vantagens para empresários, que poderão, assim que obtiverem um sistema com o padrão unicode, escrever textos em suas máquinas utilizando todos os caracteres mundiais.

#### Características do Padrão Unicode:

- O padrão Unicode é simples e consistente: o tamanho de seus códigos de caracter é sempre 16-bit, e por isso não precisa usar códigos diferenciados para codificar caracteres especiais.
- Incorporou códigos usados em outros padrões tais como o Latin-1 com seus 256 caracteres. E também, de muitos outros padrões, sejam eles nacionais ou internacionais.
- Inclui vários ideogramas Japoneses, Koreanos e Chineses.
- Como o ASCII, permite a combinação de alguns caracteres a fim de formar caracteres especiais.
- Como o ASCII e o EBCDIC (código muito usado nos computadores IBM), o Unicode codifica seus caracteres utilizando-se do código binário.