UNIVERSIDADE EDUARDO MONDLANE

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE MATEMÁTICA E INFORMÁTICA

TRABALHO DE LICENCIATURA

Uso de XML
para
Gestão de Redes de Telecomunicações

ERNESTO ALBERTO MANDLATE

TRABALHO DE LICENCIATURA

Uso de XML
para
Gestão de Redes de Telecomunicações

Autor: Ernesto Alberto Mandlate
Supervisor: Engº José Pedro Fernandes (PT-Inovação)
Co-Supervisor: Engº Américo Muchanga (UEM)

Realizado em Portugal, na PT – Inovação

Inovação Rua Eng José Ferreira P. Bastos
3800 Aveiro
Tel. + 351-234-403497
Fax: + 351-234-424160
URL: http://www.ptinovacao.pt

Fevereiro - Junho 2000

À minha mãe e

à memória do meu pai [1948, 1989]

Agradecimentos

O trabalho aqui apresentado resulta da união de vários esforços humanos, culturais, técnico-científicos e materiais.
Gostaria, assim, de prestar o meu agradecimento às pessoas e instituições cujo apoio e colaboração determinaram para a sua prossecução.

Em primeiro lugar, Cooperação Portugal Telecom - Universidade Eduardo Mondlane pela oportunidade oferecida.
Ao Engº José Pedro Fernandes, meu supervisor, que apesar de muito trabalho que tem, dedicou parte do seu tempo dando apoio e estímulos necessários bem como pela disponibilidade permanente que sempre me concedeu.

Mensagem similar vai ao Engº Américo Muchanga, meu Co-Supervisor, pela força e apoio na preparação e desenvolvimento do trabalho.

Ao Engº Vítor Ramos e Drª Maria Antónia, pela coordenação e acompanhamento de todos os momentos que o estágio atravessou, para eles o meu muito obrigado.

Ao Engº Nuno Santos, pela paciência com que me apoiou no desenvolvimento deste trabalho.

Aos meus colegas Gastão Gome, Míriam Germano e Sílvia Remtula pelo apoio moral, críticas e sugestões, no decorrer do estágio.

Ao Mário Soto e João Figueiredo bem como a todo o pessoal de apoio e de secretaria da PT-Inovação em me atender e disponibilizar todo o material que foi requisitado para este trabalho.

A todos os meus amigos pelo apoio e encorajamento em todo o processo da minha aprendizagem.

Finalmente, aos meus familiares, os melhores amigos de sempre.

Declaração de Honra

" Declaro que este trabalho é resultado das minhas próprias investigações, e o mesmo foi realizado apenas para ser submetido como trabalho de **Licenciatura em Informática** na Universidade Eduardo Mondlane ".

Junho de 2000

Ernesto A. Mandlate

(Ernesto Alberto Mandlate)

RESUMO

Como o título deste trabalho sugere, o processo de integração de aplicações de gestão de serviços de telecomunicações, usando a tecnologia XML, que é protótipo desde trabalho, foi feito através de um estudo prévio sobre a arte de XML, desde a origem até aos aspectos relacionados com as suas funcionalidades.

Existem ferramentas que permitem que aplicações, como bases de dados, forneçam seus resultados, tais como relatórios na Web. Contudo, ainda não estavam disponíveis funções que permitissem a partilha de dados entre diferentes aplicações.

Com o surgimento da tecnologia XML, a partilha atrás referida torna-se uma realidade, tanto numa IntraNet como na Internet (para o caso de aplicações remotas).

As facilidades fornecidas pela tecnologia XML permitirão aos gestores de informação uma maior economia de tempo e eficiência, no desempenho das suas funções, o que implicará maior dinamismo no fluxo de informação dentro da pirâmide de Gestão Organizacional, desde o nível Estratégico até ao Operacional.

Índice

INTRODUÇÃO

"*A comunicação é fundamental, seja por motivos profissionais ou particulares. Ter possibilidade de se comunicar de diferentes formas é imprescindível.* [URL - 1]

A rápida evolução e popularização das tecnologias de informação tem sido fundamental para garantir que as sociedades se comuniquem de forma ágil e desembaraçosa.

Não existe nada mais simbólico nesse processo da evolução das tecnologias de informação do que as redes mundiais de computadores, designadas por **Internet**, que completam a sensação de que a distância e o tempo deixaram de ser factores decisivos para a realização de investimentos, circulação de informações e para a tomada de decisões em qualquer nível e âmbito em que se organizem as sociedades e os indivíduos. A explosão da Internet teve mais impacto devido ao surgimento da *World Wide Web* (WWW), pelo facto de nela poder se usar multimédia[1]. A *Web* é um universo de páginas interligadas que se podem percorrer.

Uma página[2] típica da *Web* é composta por hipertexto, podendo este ser um texto regular, contendo ligações dentro do texto para outras páginas, ou ainda possuir *scripts* que carregam parte da sua informação em base de dados. É importante lembrar que tanto a tecnologia Internet como a WWW surgem após o surgimento de tecnologias como a de sistemas de informação(SI), usada para garantir a qualidade dos processos de produção através da disponibilização de informação, em tempo útil, aos requerentes.

O facto de um SI ser desenhado para servir a uma e única Organização, faz com que a partilha de informação entre sistemas, que não tenham sido previamente desenhados para esse propósito, seja uma ilusão e, consequentemente surgem barreiras na interacção entre dois ou mais processos de produção, mesmo que estes pertençam à mesma Organização.

[1] tecnologia informática de comunicação que combina texto, som, imagem, vídeo e animação
[2] um simples documento HTML na *Web*

Actualmente, diversos gestores de informação e grupos de pesquisa e desenvolvimento de *Software* têm centralizado suas atenções e esforços no sentido de se implementar soluções que permitam a integração de SI's na tecnologia Internet. Isto significa que aspectos relacionados com a construção de componentes reutilizáveis e de tecnologias apropriadas constituem-se como críticos na área de desenvolvimento de novas tecnologias. Neste sentido, tem-se assistido a um crescimento no aparecimento de ferramentas para facilitar essa integração, tais como *Aplication Server,* que consegue integrar sistemas baseados em *Web*, suportados por aplicações de *back end* de forma simples, eficiente e escalável.

Como estrutura de suporte a esta tecnologia, encontram-se normalmente arquitecturas baseadas em tecnologias como **XML** - ***eXtensible Markup Language,*** por disponibilizar facilidades aos programadores quanto aos formatos de dados e à sua partilha por diversas aplicações através da *Web*. Um documento xml pode ser compartilhado por várias aplicações, sejam elas compatíveis como não.

A Tecnologia XML na Gestão de Redes de Telecomunicações, cujas funcionalidades serão descritas mais adiante, é parte integrante de um projecto de criação de aplicações de gestão de redes de telecomunicações na Portugal Telecom(PT).

Objectivo Principal

Partindo de soluções existentes tais como:

- ARCO e Dim-SDH, que são produtos da PT-Inovação;
- Oracle8i da Oracle;
- Java da Sun MicroSystem;
- A tecnologia XML da W3C e outras.

O objectivo deste trabalho é desenhar e desenvolver uma solução de partilha de dados entre ARCO e Dim-SDH usando àquelas ferramentas.

Objectivos Específicos

Para o alcance dos objectivos principais serão considerados os seguintes objectivos específicos:

- Analisar as funcionalidades e potencialidades da tecnologia XML;
- Definir o modelo de interligação entre os dois sistemas (ARCO e Dim-SDH);
- Desenvolver formas de representar os dados que deverão ser partilhados;
- Considerar mecanismos adequados da transferência de dados de uma aplicação para outra.

Metodologias Usadas

Para o desenvolvimento do trabalho foram usadas as seguintes metodologias:

- Recolha de dados, usando as seguintes técnicas:
  Leitura da documentação relativa ao projecto de Interligação do ARCO e Dim-SDH;
  Entrevista ao pessoal do departamento de sistemas de informação da PT-Inovação;
- Definição do problema
- Revisão bibliográfica através da navegação na Internet, consultas a livros e brochuras fornecidas pela PT-Inovação;
- Consultas permanentes ao Supervisor , Co-Supervisor e outros especialistas na área de sistemas de informação e programação;

Prefácio

Para um melhor entendimento, o relatório do trabalho foi particionado em capítulos com vista a alcançar os objectivos preconizados e que passarei a descrevê-los resumidamente.

Capítulo I

Neste capítulo dá-se uma visão geral do estado da tecnologia XML, sua origem, conceitos gerais, tecnologias relacionadas e aborda a forma como os dados podem ser partilhados por diversas aplicações através da combinação da XML com outras tecnologias como as da Oracle e da Sun Microsystem, através duma revisão bibliográfica.

Capítulo II

Mostra a área em que o presente trabalho está inserido, descreve as aplicações que constituem o caso do nosso estudo, DIM-SDH e ARCO, bem como mostra a arquitectura da interligação.

Capítulo III

Descrição das linguagens de programação usadas para desenvolvimento do projecto, focando aspectos relacionados com a sua origem, objectivos, características e a forma como contribuem para o alcance da solução do estudo em causa.

Capítulo IV

Apresentação da estrutura funcional através de um cenário, modelo de representação de dados e são explicadas as técnicas aplicadas.

Capítulo V

Apresentação das conclusões e recomendações tendo em vista que a tecnologia XML possui especificações ainda em desenvolvimento.

Capítulo VI

Bibliografia referenciada e não referenciada, que foi a base da realização deste trabalho.

Como anexos temos o glossário, programas e conceitos referentes a telecomunicações concretamente a tecnologia PDH e SDH

Notações e Convenções usadas

A fonte normal utilizada é *Times New Roman* e o tamanho normal é 12.

Os títulos dos capítulos têm tamanho 16 e os sub-capítulos tamanho 14 e estão em negrito.

Os termos em Inglês que aparecem em itálico, são palavras cujo significado em português não foi possível obter ou são frequentemente usadas com tal nome.

Os nomes de *Softwares,* Organizações ou Empresas estão apresentados normalmente, mesmo que sejam escritos em Inglês.

Capítulo I
Estudo do Estado de XML

1.1 Origem e Objectivos

A WWW, devido as facilidades que oferece de estruturar e enviar diversos documentos para um número "ilimitado" de utilizadores de computadores, tem tido uma evolução destacada. Ela atrai companhias e desenhadores de páginas Web, dando origem ao desenvolvimento de um número cada vez elevado de aplicações, surgimento de estruturas e padrões que apoiem esses desenvolvimentos.

A XML é um subconjunto do *Standard Generalized Markup Language*(SGML). SGML é uma forma de expressar estrutura e conteúdo em diferentes tipos de documentos electrónicos.

A XML foi desenvolvida por um grupo de trabalho XML (originalmente conhecido como "*SGML editior review Board*"), formado sob o auspício da **W3C** em 1996. Esse grupo foi presidido por Jon Bosac, da Sun Microsystem, com a participação activa do grupo SGML, também da W3C. A tecnologia XML trouxe muitas expectativas aos criadores de páginas: "*A Web foi foi construída com base no HTML mas poderá ser reconstruída com base no XML*" [URL – 2], por várias razões:

1. O objectivo de HTML era que os seus elementos fossem utilizados para marcação da informação, de acordo com o seu significado, sem ter em conta à forma como ela seria apresentada no *Browser*. Citando como exemplo, título, cabeçalho principal e texto em realce deveriam ser colocados dentro dos elementos *TITLE, H1,* e *EM,* respectivamente. A utilização de elementos como *FONT* e outros que permitem uma boa apresentação torna-se difícil de forma independente ao utilizador. O mesmo acontece com o processamento de informação, que se torna difícil ou mesmo impossível. Deixar a decisão de como apresentar o título ou o cabeçalho para o *Browser* torna-se vantajoso uma vez que ele conhece melhor o ambiente e preferência do utilizador, podendo, deste modo, tomar decisões com base nesses conhecimentos. Assim, os desenhadores de *Browsers* decidiram ignorar esse facto,

e passaram a introduzir seus elementos, com o objectivo de especificar apresentação como a *font, center, bgcolor*, etc;

2. Para apresentar informação de acordo com o seu significado, são necessários elementos que não foram definidos no HTML, por exemplo, se um físico precisar de elementos específicos para suas fórmulas ou seus dados, significará uma enorme quantidade de elementos ou mesmo pode não ser possível apresentar tais elementos;

3. A utilização de *sites* para realização de encomendas, ou qualquer operação interactiva, torna-se também problemática, por exemplo, quando se alteram algumas especificações tais como quantidade de produto a encomendar, método de envio da encomenda e se pretende visualizar os dígitos alterados, tem que se pedir a um servidor que esteja talvez muito longe, ou sobrecarregado, que envie a página inteira, o que nos pode levar muito tempo.

Já existe um padrão internacional, SGML, proposto para permitir que documentos electrónicos possam ser definidos conforme seu conteúdo e estrutura, independentemente de sua forma de apresentação. Mais exactamente, SGML é uma metalinguagem, ou seja, uma maneira de descrever formalmente uma linguagem, neste caso uma linguagem para marcação de textos electrónicos.
Uma propriedade importante de SGML é sua capacidade de descrever a estrutura lógica de um documento através de marcas padronizadas (*markup*). O termo *markup* refere-se à sequência de caracteres ou de outros símbolos que são colocados em certos locais num texto para indicar como deve ser apresentado ou como está estruturado o texto. No entanto, SGML é bastante complexo de implementar e contém uma série de funcionalidades que raramente são usadas.

A ideia por detrás de XML, consiste em usar os benefícios do SGML, retirar as partes complexas, mantê-lo leve e fazê-lo trabalhar na Web. Isto não significa que HTML e SGML serão descartados, pois continuarão a ser utilizados onde se julgar conveniente. Se os dados forem de curto prazo, HTML é a forma mais simples de publicar estes documentos. Se os dados tiverem que ser usados durante muito tempo e necessitam de um pouco mais de estrutura, a recomendação é usar XML, enquanto o SGML nunca terá aceitação generalizada na Internet[3], por não ter sido desenvolvido ou optimizado

[3] *Neologismo*, palavra inglesa que designa a rede mundial de comunicação por computadores

para as necessidades dos protocolos[4] de rede, contudo, será o mais adequado para publicações que estejam altamente estruturadas.

1.2 Potencialidades e Requisitos

Simplicidade - XML foi cuidadosamente desenhado para que a sua utilização seja tão fácil como HTML;

Um padrão aberto - XML é SGML, isto é, existem múltiplas ferramentas comerciais para que se possa construir uma página sem que se saiba o código;

Baseado na experiência – A XML foi desenhado por pessoas com muitos anos de experiência, incluindo governadores da HTML e SGML, e pessoas que construíram negócios de sucesso baseados nas duas linguagens;

Extensível – Podemos criar as nossas próprias *tags*, e podemos partilhá-las com os outros utilizadores;

Neutro - Ninguém "possui" a XML, na medida em que não é uma tecnologia desenvolvida por uma empresa em particular, por outro lado, ninguém poderá vir a possui-la, dado que a XML é SGML e este último é um padrão definido internacionalmente;

Internacional – A XML foi construída de forma a suportar todos os alfabetos do mundo;

Estruturante - Esta linguagem tem poder e flexibilidade para suportar estruturas de dados;

Segurança – Devido ao alto grau de estruturação de dados, em documentos XML tornar-se-á mais fácil acrescentar assinaturas digitais ou encriptar partes ou a totalidade do documento. A **Microsoft**

[4] Conjunto de regras e formatos utilizados para que se estabeleça comunicação entre vários pontos

está a trabalhar com o W3C para definir um padrão na assinatura digital. Isto acrescentará segurança e autenticação aos ficheiros e dados XML;

Maleável - A XML inclui métodos para declarar e forçar estruturas nos documentos (através de DTD - *Document Type Definitions* - ou *XML Data Schemas*), tal como uma base de dados, assim certificamo-nos que os documentos em XML estão a ser criados correctamente;

Múltipla Interface - já que num documento XML os dados estão separados da interface, o mesmo suporte de dados pode ter várias apresentações no ecrã. Por exemplo, para um agente vendedor podemos ter uma interface muito detalhada e para o consumidor uma interface muito mais simples, estando baseadas na mesma base de dados.

Compatibilidade com a HTML - XML disponibiliza meios para embutir dados em HTML, uma vez inseridos, para os alterar não é necessário alterar a página inteira. O que torna as páginas HTML mais eficientes e mais dinâmicas.

Validação de dados - Usando o *Schema,* o autor da página pode definir exactamente quais os nomes dos elementos, seus atributos e relações que são permitidos na página. Podemos importar fragmentos de outros *Schemas*, e implementar classes de heranças. Assim é validada automaticamente a estrutura do documento. [URL-3]

1.3 Definições Gerais

A XML define um formato padrão e universal para expressar estruturas em dados. Colocar estruturas em dados significa estruturar o documento de acordo com o conteúdo, sentido ou utilização dos dados. Os documentos xml contêm dados e *markup*. Os caracteres de dados são, por vezes, referenciados simplesmente como conteúdo.

A estrutura do xml é baseada em *markups tags* que, ao contrário do HTML, não têm sentido predefinido e, para além disso, o documento xml é extensível, estruturado e pode ser validado,

separando claramente os dados da sua apresentação, isto é, o mesmo documento xml pode ser mostrado de várias formas, sem alteração à estrutura de dados subjacentes.
Os exemplos seguintes mostram claramente as diferenças entre os documentos HTML e XML

Exemplo 1.1 Registo de uma Estrutura – SDH em HTML

```
<H1>Estrutura_SDH</H1>
<H2>ID_BD_Estr: 1999</H2>
<H2>Sigla_Estr: A001</H2>
<H2>Topologia: A</H2>
<H2>Designacao: Anel da Beira Litoral</H2>
<H2>Capacidade: 16</H2>
<H2>Proteccao: 1</H2>
<H2>Designacao_Estr: ANEL DE LISBOA</H2>
```

Todas as *tags* dizem algo sobre como os dados devem ser visualizados. A informação sobre a estrutura ou relação existente entre os dados não estão especificados.

Exemplo 1.2 Registo de uma estrutura SDH em XML

```
<ESTRUTURA>
        <ID_BD_ESTR>1999</ID_BD_ESTR>
        <SIGLA_ESTR>A003</SIGLA_ESTR>
        <TOPOLOGIA>A</TOPOLOGIA>
        <CAPACIDADE>4</CAPACIDADE>
        <PROTECCAO>1</PROTECCAO>
        <DESIGNACAO_ESTR>ANEL DE LISBOA</DESIGNACAO_ESTR>
</ESTRUTURA>
```

Nenhuma das *tags* nos diz como é que os dados serão visualizados. As *tags* nos mostram a estrutura e o conteúdo do documento. A apresentação do documento depende de aplicação que irá ler o documento. A estrutura formal do documento é constituído pelas *markups tags*, podendo estas serem escolhidas pelo autor do documento, contudo, cada *tag* deve descrever os dados guardados no seu interior.

1.3.1 Elementos

Aos componentes de uma estrutura em xml, nomeadamente conjunto de *markups tags* e caracteres de dados, designamos por elementos. Um elemento pode ser uma *tag* apenas com os caracteres que essa *tag* descreve ou uma *tag* contendo outros elementos.

os caracteres descritos pelos *tags* podem ser de três tipos :

- *Parsed Character Data* (PCData), que guarda markup que será avaliado pelo *Parser*;
- *Unparsed Character Data* (CData), guarda *markup-text* de forma a que a *markup* não seja avaliada pelo *Parser;*
- *Processing Instruction*(PI), guarda directrizes de processamento e informações a passar aos programas e *parser*. Os PIs começam sempre por <? e terminam por ?>

Exemplo 1.3

```
<? xml version="1.0"?>
<! ELEMENT Estrutura_SDH( Estacao, Equipamento)>
...
<!ELEMENT Estacao(PCData)>
...
```

1.3.2 Atributos

Dentro de um elemento podemos encontrar informação que descreve suas características, que chamamos de atributos.

Exemplo 1.4

```
<!ELEMENT Estrutura_SDH (PCData)>
<!ATTList Estrutura_SdH
        ID_BD_Estr CData # Required
        Sigla_Estr Cdata  Required
        Topologia ( A | C) #Implied
>
```

Os atributos da Estrutura_SDH são o ID_BD_Estr, Sigla_Estr e Topologia.

1.3.3 Entidades

Para além de possuir caracteres definidos de uma forma linear, o documento xml pode fazer referência a outros objectos. O xml possui mecanismos que permitem ao texto e objectos, serem organizados de forma não linear, cabendo a função de reorganizá-lo de forma linear ao *Parser*. Os mecanismos que tornam o descrito anteriormente possível são chamados de **entidades**.

Dependendo do seu tamanho, um documento xml pode ser partido em vários ficheiros num disco, ou em vários objectos numa base de dados, ou espalhados na Internet e a cada partição designamos por entidade. Uma entidade é composta por um nome e seu conteúdo. O conteúdo corresponde aos dados realmente guardados e o nome referencia esses dados.

Existem dois tipos de entidades:

a) Entidades Internas

São as que não necessitam de ficheiro para armazenamento separado e o seu conteúdo é dado na sua declaração. As entidades internas são processadas como qualquer outro texto, isto é, são *parsed entities*.

b) Entidades Externas

Estas recebem seus dados de outros sistemas. A utilização do seu conteúdo é dada pelo uso da palavra *SYSTEM* seguida por URI.

Exemplo 1.5

<! Entity front-page SYSTEM "http:\\www.book.com/frontpage.gif" NDATA Gif>

A entidade front-page contem a imagem *gig* "frontpage.gif." , "NDATA" mostra que se trata de uma entidade *Unparsed* com notação Gif. (Figura 1.1)

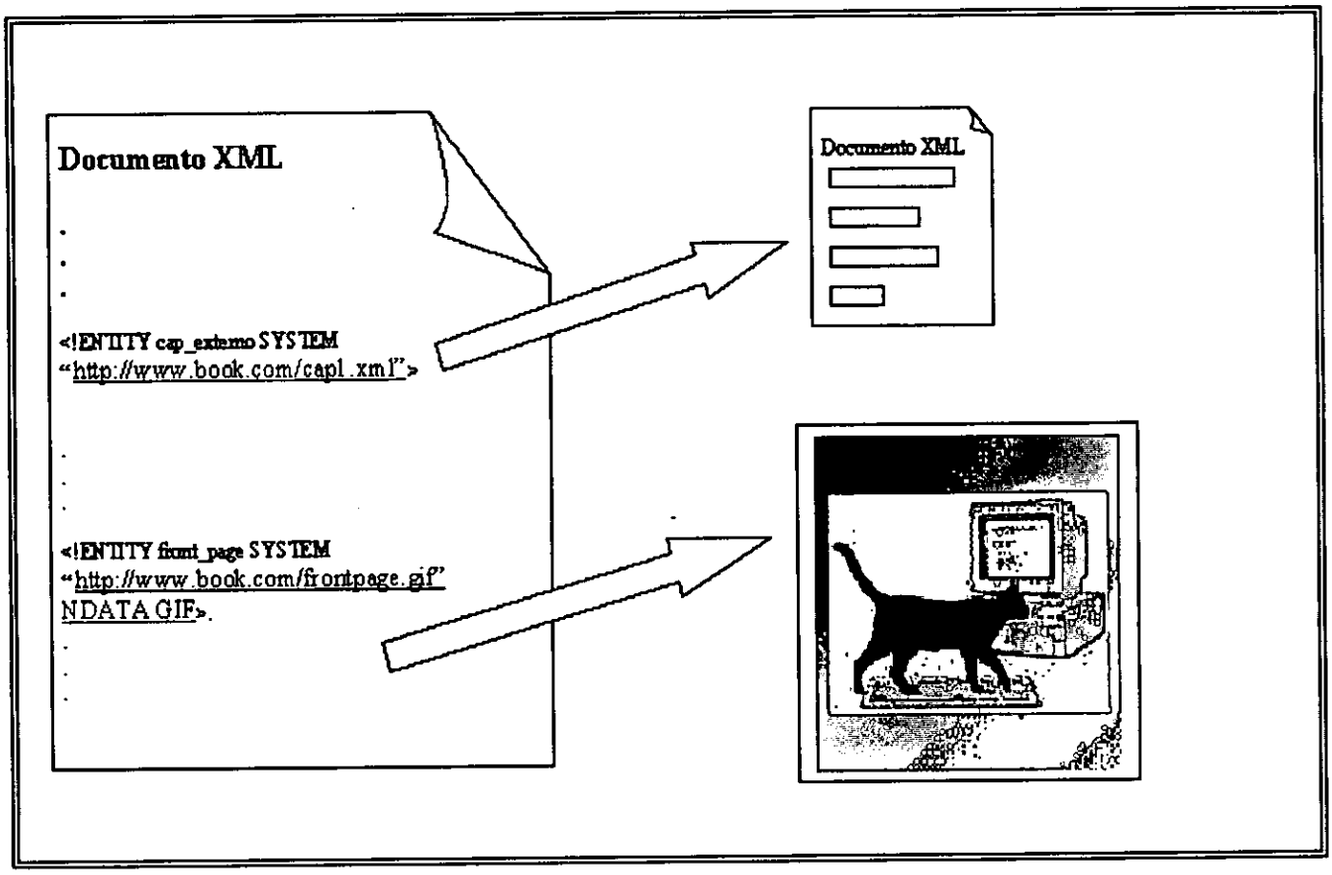


Figura 1.1 Entidades Externas

1.3.4 *Document Type Definition* (DTD)

É um conjunto de regras de sintaxe que ilustra como estão definidas as *tags* hierarquicamente, quais os atributos de cada *tag* e quais são as propriedades destes.

O DTD permite as aplicações terem informação de que nome e estruturas podem ser utilizadas num determinado tipo de documento. Todos os documentos pertencentes a um certo DTD são construídos do mesmo modo.

O DTD pode ou não fazer parte de um documento xml. Se fizer parte deste, deve ser definido como cabeçalho do mesmo, caso contrário, guardá-lo no disco com extensão .dtd e referenciá-lo no documento.

Um documento xml não tem um dtd universal, como acontece com HTML. Cada Organização, Indústria ou Programador que pretende usar um dtd para troca de informação pode definir seu próprio DTD.

O exemplo 1.4 ilustra um DTD que tem um elemento que é Estrutura_SDH e os seus atributos.

1.3.5 Documentos *Well-Formed* e Válidos

Um documento xml baseado na estrutura e regras de sintaxe de XML é considerado "*well-formad*" Um documento XML *well-formed* não tem que ter ou referenciar um DTD, mas antes, pode implicitamente definir seus elementos de dados e relações. Um documento *well-formed* deve seguir as seguintes regras:

- O documento deve começar por uma declaração XML <?xml version="1.0">;
- Todos elementos devem pertencer a um elemento chamado raiz;
- Elementos devem ser listados numa estrutura de árvore sem sobreposição;
- Todos elementos não vazios devem ter *tags* de início e fim.

Os documentos *well-formed* que também contêm ou referenciam um DTD são considerados **válidos.** Quando um documento referencia ou está conforme um DTD é *parsed.* A plicação de *parsing* verifica se o documento obedece a um DTD e é também válido, o qual permite à aplicação de *parsing* processá-lo com a certeza de que todos os dados seguem as regras definidas no DTD. Quando se guardam dados a partir de documento XML na base de dados , o DTD pode ser usado para validar sua estrutura garantindo que seus elementos de dados serão mapeados para a coluna correspondente na tabela da base de dados. Se um documento xml for gerado com base na leitura de dados numa tabela da base de dados, então será um documento válido.

1.3.6 XML *Schema*

Um *schema* define os tipos de dados que podemos inserir no local definido por #PCData no DTD. Se não existir um DTD associado ao documento xml, o esquema valida os dados inseridos entre as abertura e o fecho das *tags.*

O seu principal objectivo é definir e construir uma classe de documentos xml, utilizando construtores para restringir e documentar o sentido, utilização e relação das suas partes constituintes: tipos de dados, elementos e seu conteúdo, atributos e os seus valores, entidades e os seus conteúdos e notações. Para a especificação da informação explícita, tal como valores por defeito., os construtores de esquema também podem ser usados.

Qualquer que seja documento xml, em princípio, pode usar as facilidades do *Schema* para fazer restrições sintácticas, estruturais e de valores aplicáveis às instâncias do documento. Estas facilidades permitem a descrição e validação de restrições impostas numa aplicação desenvolvida em XML.

1.3.7 XML e os Tipos de Dados

Sempre que nos confrontamos com uma nova linguagem de programação temos nos preocupados, em primeiro lugar, pelos tipos de dados que esta suporta. Em relação aos tipos de dados, XML apresenta uma grande vantagem, pois suporta todos os tipos de dados, quer dizer, os tipos de dados que um documento xml pode ter, dependem da aplicação que fará o *parsing* do ficheiro, uma vez que XML em si não faz verificação do dados guardados, apenas verifica a estrutura do documento(*valid e Well-formed*).

Porém, existe uma necessidade de descrever os tipos contidos em um documento xml de uma forma padronizada e consistente, foi nesse âmbito que se propôs uma extensão da XML, que se designou de XML-Data.

Na especificação XML-Data, o conteúdo dos elementos *tagged* é sempre interpretado como *String*. Contudo, muitas aplicações precisam de especificar restrições rígidas para os dados que manipulam. Por exemplo, precisam de saber se um determinado dado é um inteiro, ou real ou *string*. Essa especificação ainda está em desenvolvimento e resultará em definição de tipos de dados e extensão do DTD.

1.3.8 *Document Object Model* (DOM)

É uma API(*Application Programming Interface*) para HTML e XML. O DOM define a estrutura lógica do documento e a forma de acesso e manipulação dos dados. Na especificação DOM, o termo documento é usado num sentido claro. Em xml, o termo documento é usado como um meio de representar diferentes tipos de informações, que podem ser armazenadas em diferentes sistemas, e estas informações poderão ser vistas como dados e não como documentos. Todavia, XML apresenta seus dados como documentos, e o DOM será usado para manipular seus dados.

Com o DOM, os programadores podem construir documentos, navegar, aumentar elementos, modificar ou apagar elementos ou conteúdos. Qualquer dado num documento XML ou HTML pode ser acedido, mudado, apagado ou aumentado usando o DOM, com uma pequena excepção, a interface DOM do XML ainda não foi especificada.
Como uma especificação W3C, um dos principais objectivos de DOM é providenciar um API que pode ser usado em extensas variedades de ambientes e aplicações. O DOM é desenhado para ser usado com uma linguagem de programação.

O DOM é uma API para documentos. É baseado num documento que efectivamente assemelha-se à estrutura do modelo do documento
Suponhamos que temos o seguinte documento:

```
<? Xml version ="1.0" ?>
< Estacao ( Equipamento )*>
    <Equipamento id = "0010">
        <Fornecedor_Equip> Nec</Fornecedor_Equip>
        < Config_Equip> ADM-4 </Config_Equip>
    </Equipamento>
    <Equipamento id = "0011">
        <Fornecedor_Equip> Alcatel</Fornecedor_Equip>
        < Config_Equip> ADM-16 </Config_Equip>
    </Equipamento>
```

Depois de estar criado o Object *Model* , o documento é disposto em forma de árvore, como mostra a figura 1.2.

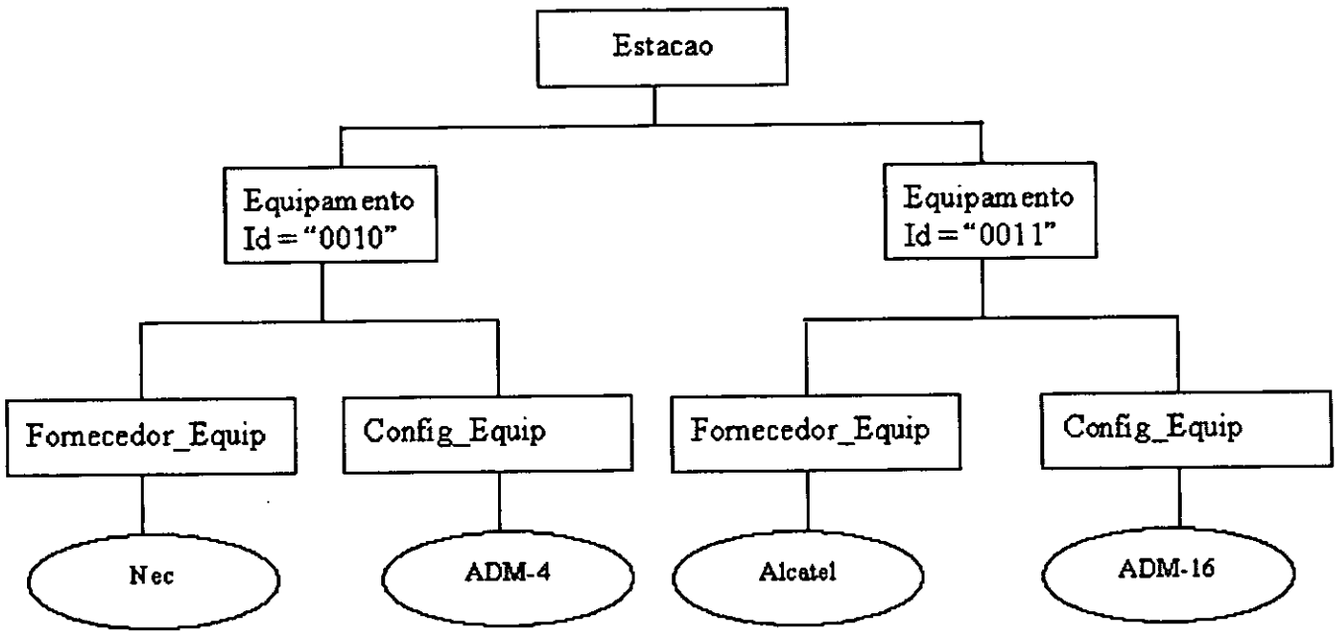


Figura 1.2 Representação de um objecto DOM

O elemento **Estacao** possui dois elementos filhos (dois Equipamentos) que são qualificados pelos seus atributos, que são fornecedor de Equipamento e Configuração de Equipamento. Os valores em elipses são folhas ou filhos de Fornecedor e Configuração ou netos da Estação.

1.4 Especificações relacionadas com XML

A tecnologia XML é definida pelas seguintes especificações:

- **Extensible Markup Language (XML) 1.0**

Define a sintaxe da XML e já foi detalhada nos capítulos anteriores;

- **XML *Linking*[5]**

Define um padrão para representar as ligações (*Links*) entre os recursos. Para além das ligações simples, como as de HTML, XML possui mecanismos para ligar recursos múltiplos e diferentes . A Xpointer (*XML Pointer*[6]) descreve como endereçar um recurso, e a XLink (*XML Link*) descreve como associar dois ou mais recursos ;

- **XSL - *Extensible Style Language***

Define a linguagem de folhas de estilo padrão para a XML.

1.4.1 XML *Linking*

Formalmente conhecido por XLink ou XLL (eXtensible Linking Language) é um trabalho em desenvolvimento na W3C e está estritamente relacionado com a *XML Recommendation.*
A especificação inicial foi agora dividida em duas : **XLink** e **Xpointer**.

XLink providencia capacidade de *linking* tais como *links* multidirecionais e externos, enquanto que XPointer providencia um caminho conveniente e fácil de descrever a sua localização em documentos XML. XPointer é agora uma camada de XPath que providencia um sistema comum para a especificação da localização que está entre nodos ou lista de nodos; a camada XPointer providencia outros dados para a localização, tais como selecções de uso geral.

[5] Ligação, elo de união
[6] Apontador

Tanto XPointer como XLink pode ser usado separadamente. Um exemplo claro, é que XLinks pode conectar muitos tipos de dados além de XML (audio, video, código de programa, etc); igualmente XPointer pode ser usado para especificar qual destas localizações são *link-ends* formais.

XLink acrescenta estes tipos de *hypertext linking funcionality* para Web:

- ***Links* multidirecionais**

  Notar que "*go-back*" não é a mesma coisa. Links multidireccionais podem ser atravessados em qualquer sentido, independentemente de se ter atravessado no outro sentido primeiro.

- ***Links Database***

  Oferece possibilidades para filtragem, ordenação, análise e processamento de colecções de links. Quando libertadas, estas bases de dados de destino libertam os autores da preocupação de gerir *links* para destinos que mudem frequentemente;

- Permitem aos utilizadores acrescentarem *links* a uma página que não seja sua. Isto implica um processo de definição de quais os *links* que se pretende mostrar, mas também torna possível as construções de infra-estruturas de anotações, comentários e avaliações comuns na Web;

- ***Links* Bi-Direcionais**

  Links que anotam documentos somente de leitura, isto é, pode se criar um *link* que mostrará o documento, ainda que o autor não seja dono do mesmo. Com certeza, isto envolve o processo de decisão de qual dos *link* será ou não visualizado deste modo;

XPointer

Especifica quais os elementos que permitem o endereçamento de estruturas internas a documentos xml. Permitem a referência específica a elementos, cadeias de caracteres, e outras partes de documentos XML, quer existe ou não um atributo ID explícito. Enquanto que o HTML providencia *links* que apontam para elementos num documentos que lhes tiver sido associado uma âncora, o XPoniter permite uma melhor especificação de destinos que o HTML:

- *Links* que apontam para locais específicos dentro do documento, mesmo quando o autor do documento não tenha introduzido um ID nesse local;

- Endereçamentos mais finos de elementos, cadeias de caracteres e extensões dentro de documentos;
- Uma sintaxe clara para indicar destinos e relações em hierarquias, de forma que os destinos sejam perfeitamente legíveis;
- O XPointer é uma sequência de *location term* que, quando interpretados, especificam uma localização em termos da estrutura em árvore expressa pela *XML markup.*

1.4.2 XSL

Um documento XSL, na verdade é um documento xml que transforma um dado ficheiro .xml em outros formatos de documento, como HTML. A linguagem XSL pode ser utilizada para acrescentar aspectos de apresentação aos elementos de um documento xml. Deste modo, é possível múltiplas representações da mesma informação a partir de vários documentos XSL diferentes aplicados a um único documento xml.

Um documento XSL pode conter uma série de regras denominadas *templates*. Os *templates* são aplicados ao documento xml e o resultado obtido é o conteúdo do documento xml com o estilo de apresentação aplicado e organizado de forma como o documento xsl especifica.

O xsl combina os padrões, DSSSL[7] e CSS[8], e incorpora todas as capacidades de uma linguagem de programação de forma a permitir funcionalidades avançadas tanto de estilo como de interactividade. Esta linguagem foi desenhada para trabalhar com linguagens de *scripting* como VBScript, JavaScript.

Objectivos de XSL

- Deve suportar *browsing*, impressão, edição interactiva e ferramentas de desenho;
- Deve ser capaz de especificar apresentação para ambientes tradicionais e Web;
- Deve suportar a interacção com informação estruturada, bem como a sua apresentação;
- Deve suportar apresentações visuais e não visuais
- Ser uma linguagem declarativa;
- Ser optimizada para disponibilizar especificações simples para tarefas comuns de formatação e não introduzir tarefas que tornem a formatação mais complexa;

[7] *Document Style Semantic and Specification Language*

[8] *Cascading Style Sheets*

- Deve disponibilizar um mecanismo que lhe torne extensível;
- As capacidades opcionais de XSL devem ser mínimas
- Deve levar em consideração outras recomendações e padrões, tais como o XML, XLL, DOM e HTML;
- Deve estar expresso na sintaxe XML;
- Os *Stylesheet* XSL devem ser claros e de fácil leitura.

Como Funciona

O XSL é uma linguagem que foi concebida para expressar *stylesheet* . Para visualizar informação que esteja armazenada, o processador XSL utiliza um *stylesheet* XSL para fazer o *parse* do documento fonte XML e tem como resultado um *output* reutilizável. Uma vez que alguns componentes desta linguagem ainda estão em desenvolvimento, até agora, os processadores XSL existentes produzem *output* em HTML contudo, teoricamente esses ficheiros podem ser tudo o que se pretender pois o padrão XSL não tem limitações no respeitante ao *output* .(ver figura 1.3)

Em cada documento XML deve existir um API que define de forma explícita o ficheiro XSL que define o *stylesheet* que o processador de XSL deverá utilizar, como no exemplo:

```
<?xml-stylesheet type="text/xsl" href="caninos.xsl"?>
```

O *stylesheet* contêm um conjunto de regras que lhe facultam a apresentação de uma classe de documentos fontes XML. Essas regras são constituídas por duas partes:

- Um padrão (*pattern*) que identifica os elementos do documento fonte XML a que esta regra se aplica;
- Uma acção (*action*) que define o que deve ser feito com o elemento.

Esses padrões e acções são depois usadas para mapear a árvore fonte(árvore XML construída pelo DOM) para uma outra árvore composta por objectos de fluxo (*folw objects*) e estes descrevem o interface utilizador.(ver figura 1.4)

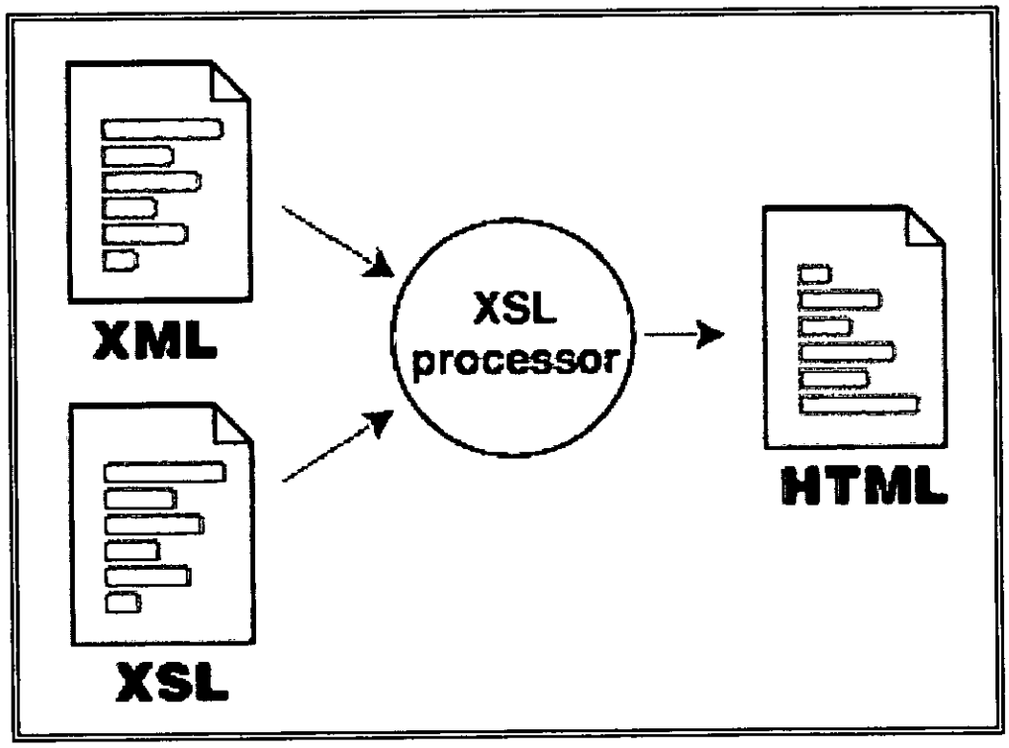


Figura 1.3 - Modelo de funcionamento de XSL.

O *output* visual é responsabilizado para os *flow object*. Vamos considerar cada item de interface utilizador como *flow object*, então poderá ser imagem, texto, tabela, lista, etc. Esses objectos formam, de forma conceptual, uma árvore, em que:

- Os parágrafos, tabelas, sequências e outros contentores de objectos formam os **Ramos**;
- Caracteres, imagens outros objectos atómicos são **Folhas**;
- O documento de *output* é a raiz da árvore. Esta árvore é designada por *flow object tree.*(figura 1.4)

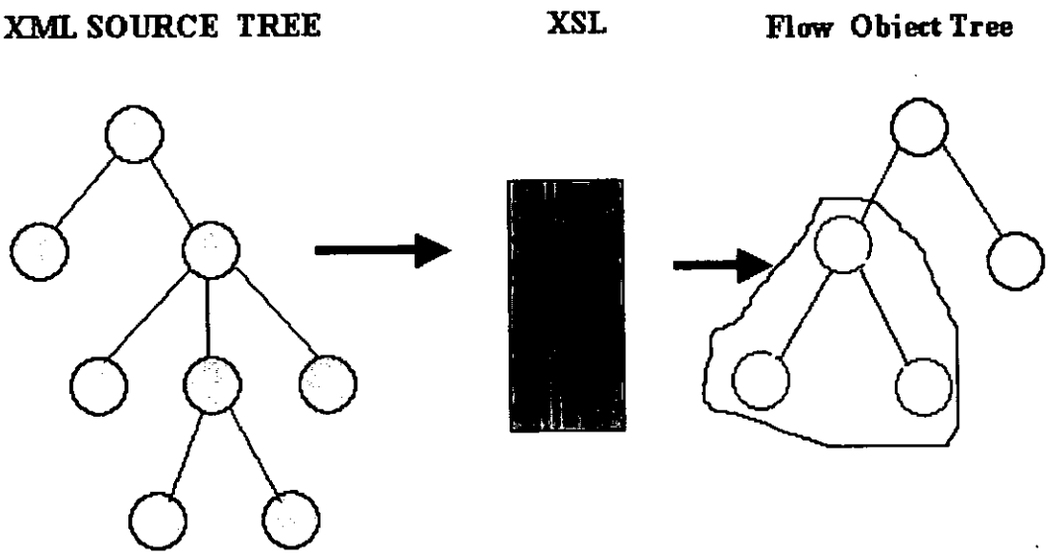


Figura 1.4 Árvore Lógica definida por *flow Objects*

1.5 Troca de dados entre aplicações usando um documento xml

O documento XML pode ser usado como um formato comum para troca de informação entre aplicações. Como o XML descreve os dados, aplicações diferentes podem partilhar dados sem que necessariamente sejam homogéneas. As aplicações podem partilhar os dados com ou sem um DTD comum. A troca de dados entre aplicações que partilham um DTD comum faz com que a troca de dados seja eficiente e fácil de gerir. Sem um DTD comum, as aplicações podem trocar dados XML, mas existe uma necessidade de um processamento adicional para restaurar, e em alguns casos, interpretar como os dados são passados de uma aplicação para outra.

Em geral, a troca de dados entre aplicações que não partilham um DTD comum tem o seguinte processo:

1. A aplicação transmissora gera um documento xml baseado num DTD;
2. Transmite o documento xml para aplicação receptora;
3. A aplicação receptora faz o *parser* dos dados xml, executa seus procedimentos específicos, e escreve os dados na sua base de dados;

4. A aplicação receptora envia o documento original (ou mesmo um novo documento gerado) para a outra aplicação que estiver em cadeia.

1.5.1 Troca de dados usando um DTD comum

O DTD especifica o tipo de elementos de dados xml e a estrutura relacional entre eles. A figura 1.5 descreve o processo da troca de dados entre duas aplicações que compartilham o mesmo dtd.

Neste caso os dados xml são gerados por um XSQL Servlet baseado em "*queries*" introduzidos por um formulário *web*. O documento resultante é estruturado e baseado num DTD comun.

A aplicação receptora recebe os dados, faz o *parser* e processa o xml usando um *xml parser for java,* e ultimamente escreve os dados na base de dados, usando o "*xml sql utility*"

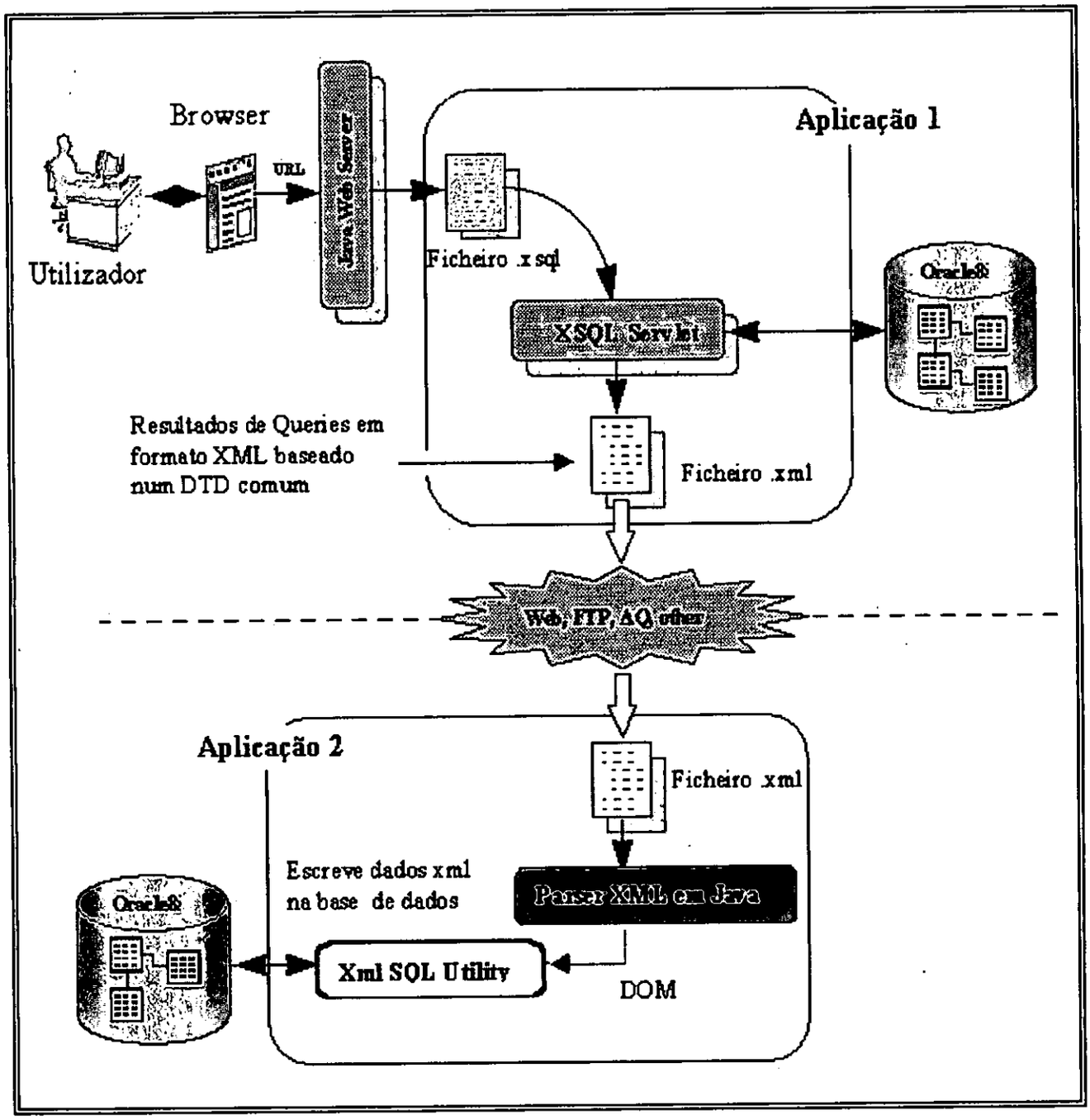


Figura 1.5 - Troca de dados usando um dtd comum

1.5.2 Troca de dados sem um DTD comum

Trocar dados xml entre aplicações que não compartilham um DTD comum requer um processamento intermediário e transformação do xml.

Se quisermos escrever um xml para um objecto ou tabela de uma base de dados enquanto a estrutura dos dados xml não é compatível com a estrutura do objecto ou tabela , será necessário transformar os dados xml antes de escrevê-los na Base de Dados. O XSL *Stylesheet* é uma abordagem que pode ser usada para transformar o documento xml para um novo documento xml, que esteja com estrutura compatível à base de dados. (ver figura 1.6)

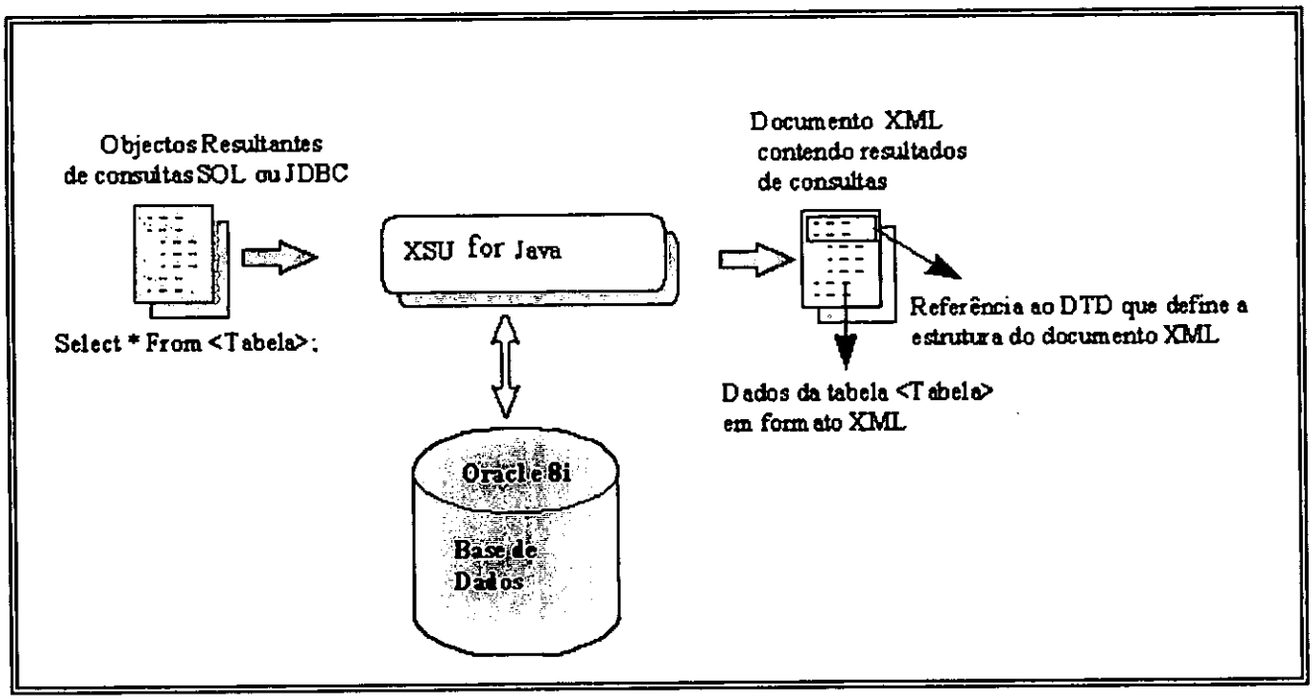


Figura 1.6 Troca de dados sem um DTD comum

O XSU gera o DTD como um ficheiro separado ou acrescido, para o xml gerado dentro da *tag* DOCTYPE. O dtd pode ser usado para desenhar um xsl *stylesheet* que transforma o documento xml original antes de inserir seus dado na base de dados.

Os dtd's gerados são limitados uma vez que não providenciam informação sobre o tipo de dados . Todos os dados no dtd gerado são do tipo *String*. Assim, quando se gera um dtd a partir da base de dados, a informações sobre o tipo de dados é perdida pois, não existe modo de representar mesmo um simples tipo de dado como *boolean, data, integer*, etc. Deste modo, as aplicações que usam dtd devem definir tipos de dados baseados no contexto de dados ou nas suas *tags*.

1.5.3 Dados de um formulário *Web* para uma base de dados

Uma forma de enfatizar que os dados obtidos via formulário serão mapeados para um esquema da base de dados é desenhar o formulário seguindo a estrutura dos dados da base de dados, deste modo, os dados xml gerados serão compatíveis com os da base de dados.

O cenário seguinte descreve o uso de ***xsu*** e o ***xml parser for java***

1. Uma aplicação java usa o xsu para gerar um dtd que está de acordo com o formato da tabela ou objecto em causa;
2. A aplicação sustenta este dtd para dentro de *xml class generator for java*, que irá construir classes que podem ser usadas para carregar o formulário web apresentado ao utilizador;
3. Usando as classes geradas, o formulário web é dinamicamente gerado por *java Serve page*, *java servlet* ou outro componente;
4. Quando o utilizador tira o formulário e substitui-o, o servlet mapea os dados para os dados xml e xsu escreve-os na base de dados.

Pode-se usar o *DTD-Generation Capability de XSU* para determinar que formato xml é esperado pelo objecto ou pela tabela.
O dtd resultante pode ser usado com *input* para o *xml class generator* para um conjunto de classes baseadas nos elementos do dtd. A seguir pode se escrever um código java que usa estas classes para gerar uma infra-estrutura por detrás do formulário *web*. O resultado é a conversão dos dados submetidos via *Web* para um documento xml que pode ser escrito numa base de dados.

1.5.4 Resumo do processo de envio do documento

Na descrição que se segue, a aplicação emissora transmite o documento xml e a aplicação receptora recebe-o.

File transfer
A aplicação receptora pede o documento xml da aplicação emissora via FTP ou outros protocolos de transferência de ficheiros. O documento é copiado para o sistema de ficheiro da aplicação receptora que lê e processa o ficheiro.

HTTP

A aplicação receptora faz um pedido HTTP para o Servlet, o qual retorna o documento xml para aplicação receptora que irá ler e processar o documento.

Formulário Web

A aplicação emissora entrega o formulário Web. O utilizador submete a informação a partir de *applet* java ou javascript que corre no browser. O *applet* ou javascript transmite o formulário do utilizador em formato xml para a aplicação receptora. A aplicação receptora pode escrever os dados na sua base de dados, para tal efeito, a aplicação emissora deve enviar um documento xml compatível com o formato das tabelas ou objectos alvos na base de dados.

1.6 Segurança

"*Várias metodologia e ferramentas são utilizadas para garantir o máximo de segurança dos sistemas críticos da rede e dos dados que trafegam pelos quatro cantos do mundo*" [URL – 9]

A utilização de *firewall* é sempre uma solução indicada pelos especialistas no assunto, porém, não se deve parar por aí. A segurança em tecnologia de informação baseia-se em três grandes tópicos:
Políticas;
Ferramentas;
comportamentos.

A implementação da política de segurança por meio de normas e procedimentos é um processo bastante longo, burocrático e em certas vezes traumático.

A cada fase de implementação devemos dar máxima importância aos factores comportamentais. É preciso trabalhar na conscientização dos utilizadores e promover treinamento específico de aderência às normas e procedimentos que estão sendo incorporados. De nada vale uma regra se ela não é seguida ou sequer entendida. [Pereira, 1998]
Várias ferramentas possibilitam auditoria dos dispositivos, equipamentos, configurações, *softwares* operacionais, base de dados e aplicativos instalados na rede.

Em função da integridade, confidencialidade e disponibilidade desejada pela organização nos seus sistemas de informação, várias outras soluções como criptografia, certificados e assinaturas digitais podem ser usadas.

1.6.1 Criptografia

O principal objectivo da criptografia é o de garantir a privacidade. A criptografia permite uma comunicação segura mesmo em um canal de comunicação inseguro. Criptografar um mensagem significa codificá-la através de uma cifra (algoritmo, outros alfabetos, etc.) de forma a tornar a sua descodificação possível somente a quem possui a cifra com a qual a mensagem foi codificada. Esta

cifra é comumente chamada de chave. O processo de "disfarçar" a mensagem é **cifragem** e transforma-a em criptograma. O processo de recuperar a mensagem original a partir do criptograma denomina-se decifragem. A criptonálise é a ciência de quebrar criptogramas, ou seja descobrir como fazer a decifragem de um criptograma, sem saber, à partida, como ele foi cifrado.

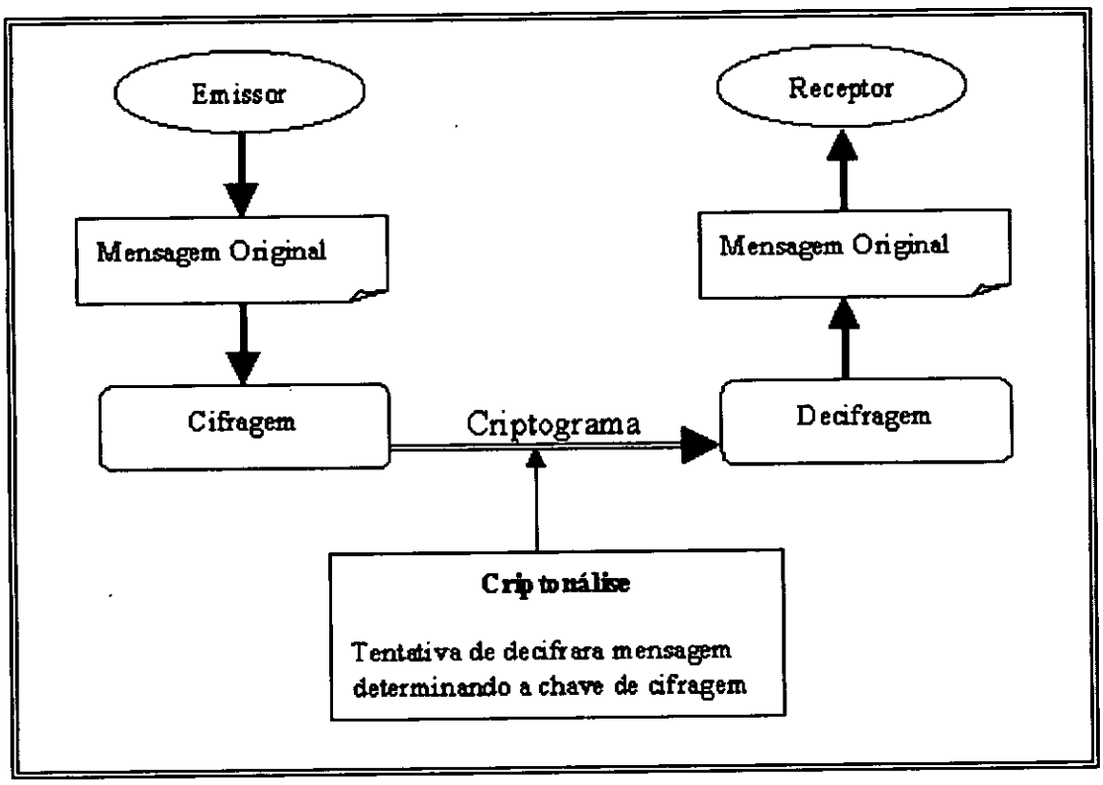


Figura 1.7 Esquema de transmissão

Basicamente existem dois sistemas de criptografia: o sistema simétrico e o sistema assimétrico, e que a seguir passo a descrevê-los.

Sistema Simétrico

Este sistema baseia-se num algoritmo de chave secreta. O sistema é baseado num emissor e num receptor da mensagem, onde ambos conhecem e usam a mesma chave secreta. O emissor usa esta chave secreta para encriptar a mensagem enquanto para o receptor é usada para decriptografá-la. Neste caso, é necessário que o emissor e o receptor acordem numa chave, antes de poderem usar o sistema.

O principal problema deste sistema é fazer com que tanto o emissor como o receptor utilizem a mesma chave secreta sem que ninguém descubra. Seria necessário utilizar um canal de comunicação realmente seguro para transmitir a chave secreta para o receptor para depois transmitir o criptograma pelo canal inseguro.

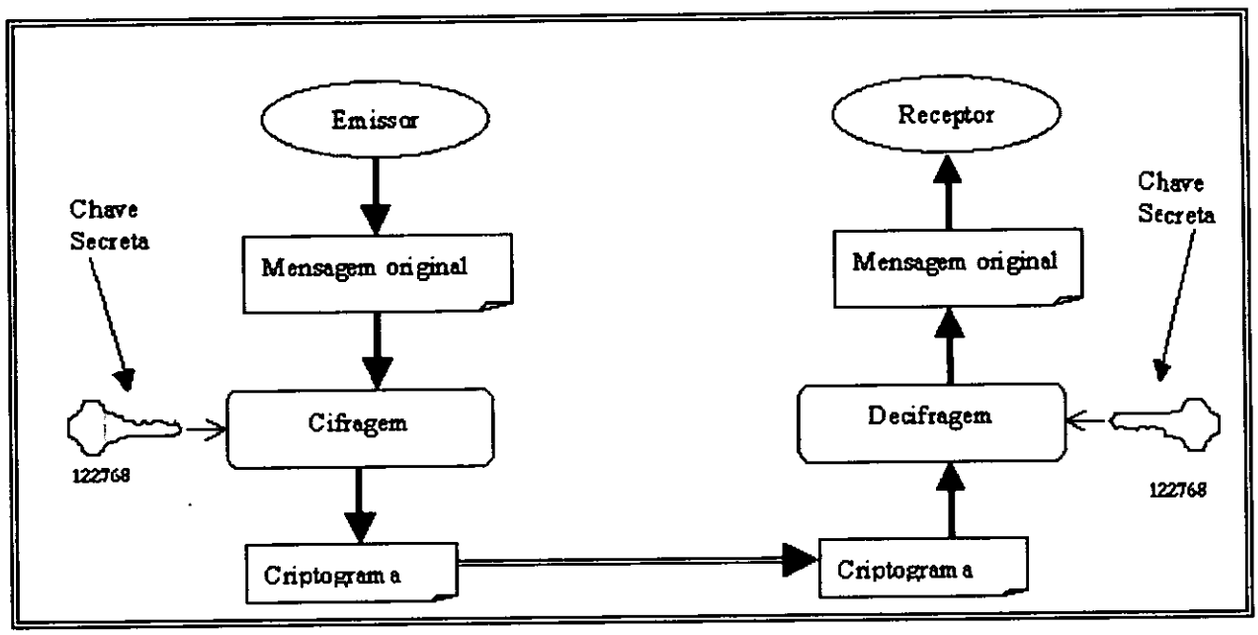


Figura 1.8 Sistema simétrico

Exemplos deste sistema de criptografia podem ser DES (*Data Encription Standard*), triple-DES, IDEA (*international Data Encription Algorithm*), etc.

Sistema Assimétrico

Os algoritmos simétricos requerem uma chave comum e secreta que é usada tanto para a cifragem como decifragem. Como forma de solucionar os problemas resultantes do uso da mesma chave os dois momentos na criptografia, surgiram os **algoritmos assimétricos,** em que as chaves são obrigatoriamente diferentes.

Os sistemas assimétricos possuem duas vantagens importantes:

a) A derivação da chave de decifragem é computacionalmente inviável, conhecendo a chave de cifragem (chave pública). Neste contexto cada utilizador tem duas chaves: a privada, conhecida apenas por ele e a pública, conhecida por todos.

b) Qualquer da duas chaves pode ser escolhida para ser usada na cifragem e a outra para a decifragem. Um dos algoritmos com estas características é o **RSA**-Rivest, Shamir e Adleman (1978). Este algoritmo usa na sua composição dois problemas numéricos complexos, logaritmo discreto e factorização. A sua segurança está patente na dificuldade de fazer a factorização de números muitos grandes. Por exemplo, para factorizar um número de com mais de 100 dígitos requer cerca de 2 biliões de anos. RSA é um dos algoritmos mais importantes no vasto mundo da criptografia e o seu uso é muito variado. Serve de base para esquemas de assinaturas digitais , mecanismos de segurança de *E-mail,* etc.

Para além dos algoritmos de criptografia acima descritos. existem muitas outras formas para assegurar que mensagens enviadas em meios não muito seguros, mesmo que sejam interceptadas não sejam descodificadas:

Assinaturas Digitais

São assinaturas electrónicas que dificilmente podem ser falsificadas. A assinatura é uma parte de texto computado que é encriptado e enviado com a mensagem. O receptor decripta a mensagem e recompõe o pedaço do texto recebido.

As assinaturas digitais podem se r baseadas em chave secreta ou mesmo pública.[URL-10]

Certificado Digital

O certificado digital associa a identidade de um titular um par de chaves electrónicas, das quais uma pública e outra secreta, que comprovam a identificação, quando usadas em conjunto.

1.6.2 Segurança em XML

Tal como em outras aplicações da Intenet, a segurança em XML também será baseada em algoritmo ou sistemas atrás descritas: criptografia, assinaturas digitais, certificados digitais, etc.
XML pode também providenciar formas de autenticação que possam ser usadas em outras aplicações da *Web*. A *Internet Engineering Task Force* (IETF) e a W3C juntaram esforços para estabelecer um padrão de assinaturas digitais na Web baseado na linguagem XML. As duas entidades tomaram a decisão no 45º encontro da IETF na cidade Norueguesa Oslo a julho de 1999.

A decisão foi tomada porque a XML possui chaves de criptografia pública e privadas que identificam os usuários e que possam ter aceitação universal.
Esta solução foi tomada devido à existência de desconfiança de pessoas na verificação de assinaturas digitais nas transações do comércio electrónico.[URL –11]

Para a segurança em XML já foram definidos requisitos que deverão ser levados em consideração no desenho dos algoritmos.

Modelo de dados e sintaxe de Assinatura

1. A estrutura de dados de assinatura XML devem ser baseados no modelo RDF[9] ;
2. As assinaturas XML são aplicadas para qualquer recurso endereçável por um localizador (URIs ou fragmentos) dentro da demonstração que refere a um recurso externo ou interno;
3. Uma assinatura XML deve ser aplicável para uma parte ou totalidade de um documento xml;
4. Múltiplas assinaturas XML devem ser capazes de existir num conteúdo estático de um recurso Web dadas várias chaves;
5. Assinaturas em XML são em primeiro lugar classes de objectos por si e, podem ser referenciados e assinados;
6. A especificação deve permitir o uso de várias assinaturas digitais e codificação de mensagens de autenticação, como os esquemas de autenticação simétrica;

[9] *Resource Description Framework*

Formato

1. Uma assinatura XML deve ser um elemento xml (como o definido na especificação XML 1.0);
2. Quando assinaturas XML são postas dentro de um documento as operações devem preservar (1) o elemento raiz do documento e (2) os elementos descendentes em seus devidos lugares excepto pela adição da assinatura;
3. Um importante uso de assinaturas XML será separado das assinaturas Web. Contudo, as assinaturas devem ser embutidas dentro ou encapsular XML ou codificar o conteúdo.

Capítulo II
Enquadramento do Trabalho

O recurso a novas tecnologias para a solução de problemas relacionados com a partilha ou gestão de dados é uma realidade que deve ser aderida e explorada, de modo a acompanhar as evoluções mundiais com rapidez e tornar o factor tempo desprezível na execução de tarefas dependentes da disponibilização imediata de informação. É nesse âmbito que surge a ideia de estudar e implementar uma solução para a troca de dados entre duas aplicações de gestão de redes de telecomunicações da Portugal Telecom, onde se poderá explorar as potencialidades da tecnologia XML.

Para o melhor entendimento da arquitectura de interligação, em primeiro lugar, passo a descrever o âmbito em que aplicações em estudo se inserem.

2.1 DIM-SDH .- Dimensionamento da rede SDH

É uma aplicação que foi desenvolvida com o objectivo de dimensionar os anéis e cordas de redes SDH, tendo em conta as restrições e características dos fornecedores (NEC e ALCATEL) de equipamentos SDH existente na PT. Falar sobre dimensionamento de rede significa referir-se à ocupação da largura de banda dos "*span*" (ligação entre nós SDH) de uma forma optimizada, e ao dimensionamento dos nós SDH utilizados, tendo em vista a descrição das cartas que compõem o nó.

Os dados são introduzidos pelo utilizador, através da técnica de navegação entre menús. (figura 2.1)
Exemplo de dados :

- Tipo da topologia;
- Nome do fornecedor;
- Tipo de hierarquia;
- Tipo de equipamento;

- Tipo de protecção;
- Número de nós e sua identificação;
- Matriz de tráfego;
- Matriz óptica.

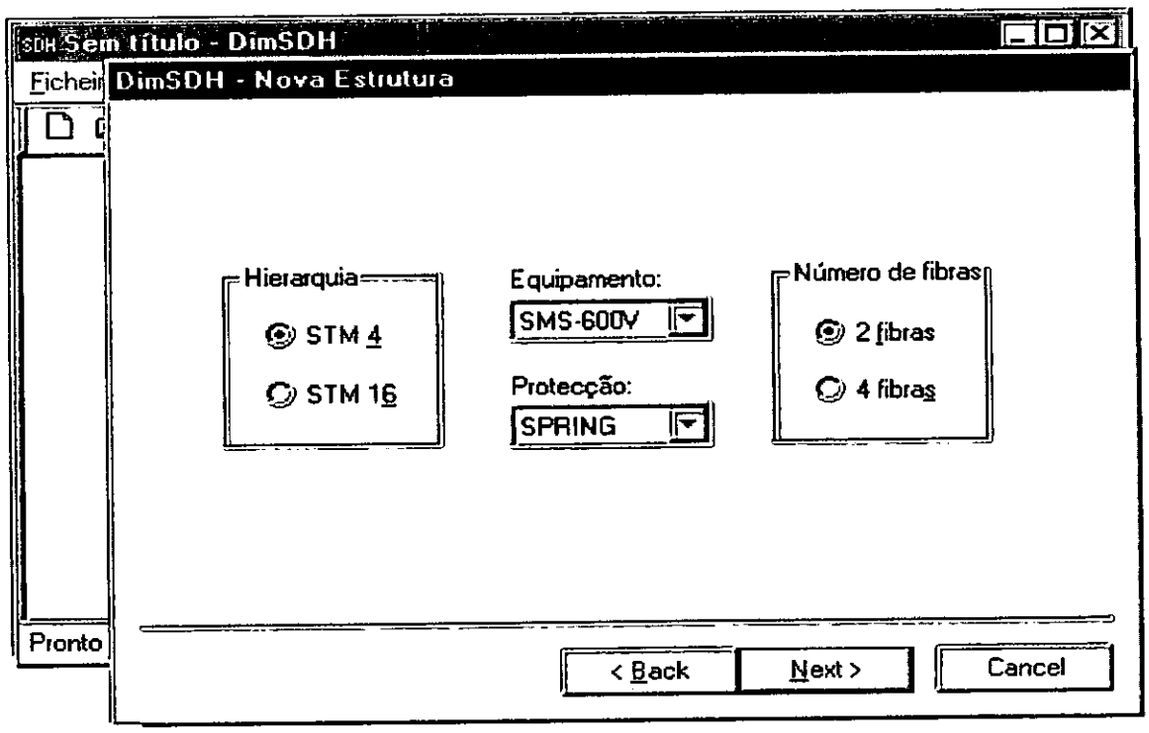


Figura 2.1 Janela de entrada de dados no DIM-SDH

Após a sua introdução, os dados são processado e produz-se um *output* em ficheiros Excel.

Exemplo de um relatório de equipamento.

E15							
Hierarquia do Anel: STM-4			Equipamento SMS-600V				
Código Sistel	**Código NEC**	**Descrição**	**Preço Unitário**	**N1**	**N2**	**Quant. Total**	**Preço Total**
		Bastidor ETSI					
50104000		Bastidor ETSI V2 (2.2m)	140.195,00	1	1	2	280.390,00
50104001		PDP-P. Distribuição Alim. V2	58.414,00	1	1	2	116.828,00

		Rack SMS-600					
50104011	E32-146-F2724-BD00	Sub-B. Básico tipo 2.4 (75ohm)	679.756,00	1	1	2	1.359.512,00
50104024	E32-146-F2724-BD00	Sub-B. Extensão 2M (75ohm)	519.615,00	0	0	0	0,00
		Interface PDH					
...	...	...	...	...	...	...	...
50104041	E32-484-Y6771-0A00	U. 140/STM-1e	169.601,00	2	2	4	678.404,00
50104043	E32-484-Y6361-0A00	U. 34M	177.710,00	0	0	0	0,00
50104044	E32-484-Y6362-0B00	U. 2M (75ohm)	154.060,00	2	2	4	616.240,00
		TOTAL					**11.735.495,00**

Os resultados[10] do dimensionamento de redes SDH são usados para a realização de tarefas tais como gestão de projectos de redes, diagrama de mux, modelo para anel ou corda que sejam NEC ou ALCATEL.

2.2 ARCO – Arquitectura da rede Core[11] da PT

É um sistema de informação responsável pela gestão de provisão de recursos da rede, nas componentes de transmissão e de comutação, suportado por cadastros que permitam atribuir e gerir a capacidade instalada na rede. Até este momento foi especificada e desenvolvida a versão 1.0, centrada apenas em componentes tecnológica SDH e o projecto do desenvolvimento do ARCO ainda está em avante, no sentido de se implementar outras componentes que sejam prioritárias, como a PDH.

Tal como na aplicação DIM-SDH, no ARCO os dados são introduzidos a partir de menús de navegação. Os dados são relativos à Estrutura_SDH, Equipamento, Portos, Interligação de Equipamentos, estabelecimentos de caminhos entres os Portos, etc.

Os dados são processados e armazenados em uma base de dados Oracle 8i.

[10] Em geral os resultados podem ser:
lista de equipamentos, material de instalação,
ocupação física dos portos, etc.

[11] core significa central ou núcleo

2.3 Arquitectura do Modelo de Interligação

Uma vez que o ARCO é uma aplicação que processa e armazena a informação em uma base de dados e o DIM-SDH processa a informação e guarda em ficheiros .xls, o objectivo da interligação é que o *output* do processamento feito na DIM-SDH possa ser armazenado na ARCo e que o ARCO forneça parte do seu *input* ou *output* ao DIM-SDH.

A figura 2.3, mostra o mecanismo de troca de dados entre o ARCO e o DIM-SDH.
O utilizador do DIM, através de uma chave que lhe dá permissão de estabelecer a ligação com o ARCo e usando o XSU, cria e envia, um ficheiro .xsql (1), contendo especificações de dados que ele necessita. A aplicação receptora, que é ARCO, recebe e testa se o ficheiro é válido ou *well formed* (*parser*) e verifica a compatibilidade dos dados requeridos com os da base de dados. Se os dados forem compatíveis, através de XSU selecciona-se os dados na base de dados e produz um ficheiro com extensão .xml (2), baseado num DTD comum. (2) é enviado ao DIM-SDH. Se este for válido, pode ser usado como *input* conforme o pedido.
Os *outputs* gerados pelo DIM-SDH podem serem usados como dados de entrada no ARCO ou então guardados na base de dados ARCO.

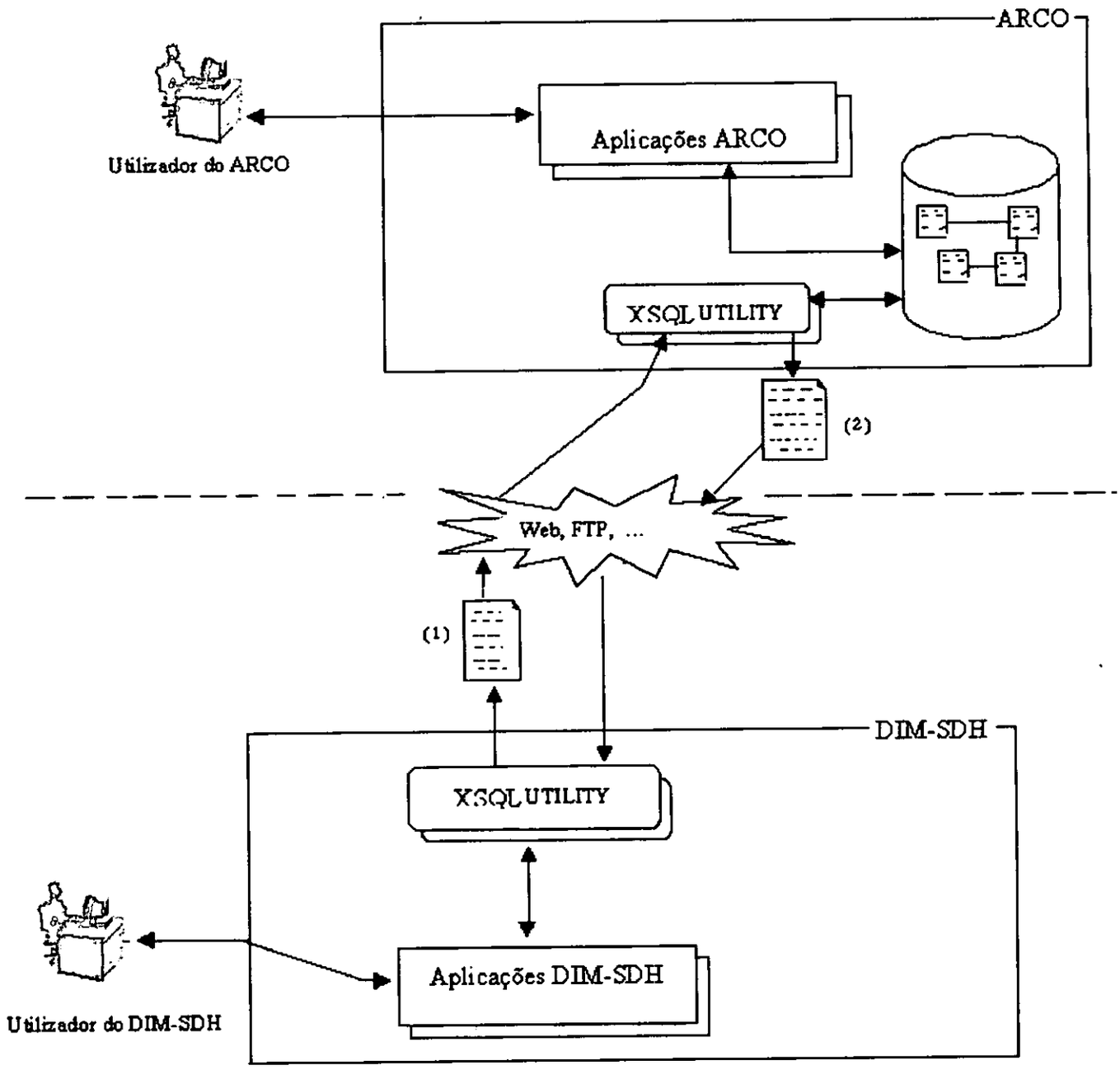


Figura 2.3 Arquitectura da Interligação

Capítulo III
Linguagens de Programação

3.1 Linguagem de Programação Java

3.1.1 Introdução

A Internet e a WWW são, sem dúvida, a grande explosão nos últimos tempos nas áreas de informática e telecomunicações e, por serem tão revolucionários, passam a fazer parte do nosso quotidiano. O crescimento explosivo da Intenet deve-se, em grande parte, ao surgimento da WWW. A web surgiu pela necessidade de transferência de mensagens numa IntraNet, pouco mais tarde foi generalizado para uso na Internet, trazendo assim um pouco de ordem ao caos que era a Internet, e à possibilidade de navegar pelo mundo inteiro usando uma interface gráfica, o que atraiu grande massa de utilizadores de computadores pessoais para a rede, que antes era apenas dominada pelo meio académico.

A www consegue distribuir informação organizada para milhões de utilizadores de computadores espalhados pelo mundo inteiro, contudo, " *a Web era só para ver* ", [URL –5], pois essas informações eram apenas estáticas e passivas. Com o surgimento de Java, a www passa a ter algo a mais – animações, apresentações, multimédia, jogos em tempo real com vários participantes, etc.

Em poucas palavras, podemos dizer que Java é uma linguagem de programação Orientada a Objectos, com o objectivo de proporcionar capacidade de realizar várias tarefas em paralelo e portabilidade entre várias plataformas e sistemas operacionais.

3.1.2 Características de Java

A linguagem de programação Java foi criada para resolver um leque de problemas na prática da programação moderna. A linguagem java é simples, orientada ao Objecto, Distribuída, robusta, Segura, Interpretada, *Multithreaded* e dinâmica.

Simples

- Os criadores de java, tiveram como um dos objectivos principais, criar uma linguagem que pudesse ser fácil de implementar, sem necessidade de muito treinamento. Por outro lado, para manter a linguagem familiar, mas ao mesmo tempo pequena, os projectistas de java removeram muitos recursos em C e C++, para habilitar a projecção de software que possa rodar sozinha em máquinas pequenas.

- **Orientado ao Objecto**
- As facilidades da orientação ao objecto da linguagem java são essencialmente as de C++. A orientação ao objecto é muito poderoso pois facilita a clara definição de interface e permite a reutilização dos softwares.

Distribuída

A existência de uma biblioteca de rotinas que possam ser copiadas facilmente com protocolos TCP/IP como HTTP e FTP, assim, as aplicações java podem abrir e aceder objectos através da rede via URL's.

Robusta

A linguagem java destina-se a escrever programas que possam ser efectivamente confiáveis. Coloca ênfase no princípio de "*checar*" possíveis problemas e eliminar situações que possam originar erros contudo, um programador incompetente ainda consegue escrever *softwares* cheio de *Bugs*.

O facto de Java ser uma linguagem fortemente tipada (como C++) viabiliza uma abrangente pesquisa em tempo de compilação, logo *bugs* podem ser encontrados rapidamente.
A grande diferença entre Java e C/C++ é que Java possui um modelo de ponteiros que elimina a possibilidade de sobreescrição de memórias e corrupção de dados. Em lugar de um ponteiro, java possui vectores.

Segura

Java é destinada a ser usado em redes/meios distribuídos. Muita ênfase tem sido colocada na questão de segurança. Java possibilita a construção de sistemas livres de vírus e de adulterações. As técnicas de autenticação são baseadas na encriptação da chave pública (usa concretamente o algoritmo RSA). As transferências da java incluem verificação d e código de *byte*, evitando deste modo acréscimo de vírus ou cavalo de Tróia (se o tamanho for alterado no caminho, será abordado).

Arquitectura Neutra

" *Java nasceu na rede e foi feito para a rede* " [URL –5]. Em geral as redes são compostas de uma variedades de sistemas cada um com variedade de CPU's e arquitectura de sistemas operacionais. Para que a aplicação java execute em qualquer lugar na rede, o seu compilador gera um formato de arquivo objecto de arquitectura neutra. O código compilado pode ser usado em muitos processadores, trazendo à presença do sistema em tempo de execução java.

Interpretada

O interpretador java pode executar códigos de *byte* java directamente em qualquer máquina para qual o interpretador tenha sido portado.

Multithreaded

Multithreaded é o caminho de fazer aplicações com várias linhas de execução .Java possui um leque de primitivas de sincronização baseadas amplamente no uso de monitor e na variável da condição.

Dinâmica

Isto significa que qualquer classe java pode ser carregada em um interpretador java a qualquer momento, mesmo quando ele já está rodando.

A linguagem Java associada ao produto XSU da Oracle, permitiu-me fazer códigos para a introdução de dados de um ficheiro .xml para tabelas de base de dados bem como fazer consultas à base de dados para depois fazer a criação do correspondente documento .xml. Para além de usar as bibliotecas da classe **jdk1.2.1** da Sun Microsystem, foi necessário usar as seguintes classes:

- xmlparserv2.jar
- oraclexmlsql.jar
- classes111.zip
- classes12_01.zip

3.2 SQL em Oracle

Oracle SQL (Structured Query Language) é uma aplicação componente do sistema de gestão de base de dados Oracle da ORACLE. Suas funções estão relacionadas com a definição e manipulação de dados em base de dados: Definir ou modificar a estrutura de uma base de dados, definir usuários, aceder aos dados, inserir novos dados e mais.

O SQL é composto por dois elementos: a linguagem de definição de dados (DDL – *Data Definition Language*) e a linguagem de manipulação de dados (DML – *Data Manipulation Language*) que funcionam segundo a mesma estrutura e sintaxe de tal maneira que a sua distinção não é aparente.

DDL é usada para definir a estrutura dos dados: especifica quais as colunas que compõem uma tabela, define índices, direitos de acesso dos usuários, etc.

DML lida com consultas, inserção, actualização e eliminação de dados e com várias outras tarefas associadas, como por exemplo a protecção dos dados de actualizações simultâneas por parte de usuários, etc. [Campos, 1999]

No presente trabalho, esta ferramenta foi de utilidade para realizar as consultas na base de dados ARCO, que foram muito úteis para a definição da estrutura do modelo de interligação entre o ARCO e o DIM-SDH.

Capítulo IV

Desenvolvimento Implementação

Uma vez que a partilha de informação é feita a partir de um DTD comum, este capítulo dedica-se a mostrar os modelos que serviram de apoio para a concepção bem como mostrar alguns módulos desenvolvidos no âmbito deste projecto. O projecto culminou com a definição do modelo de dados a serem partilhados e implementação dos módulos pertinentes, nomeadamente introdução de dados na base de dados a partir de um ficheiro .xml e leitura dos dados da tabelas da base de dados para criar um feiro .xml capaz partilhável.

4.1 Modelo de Dados

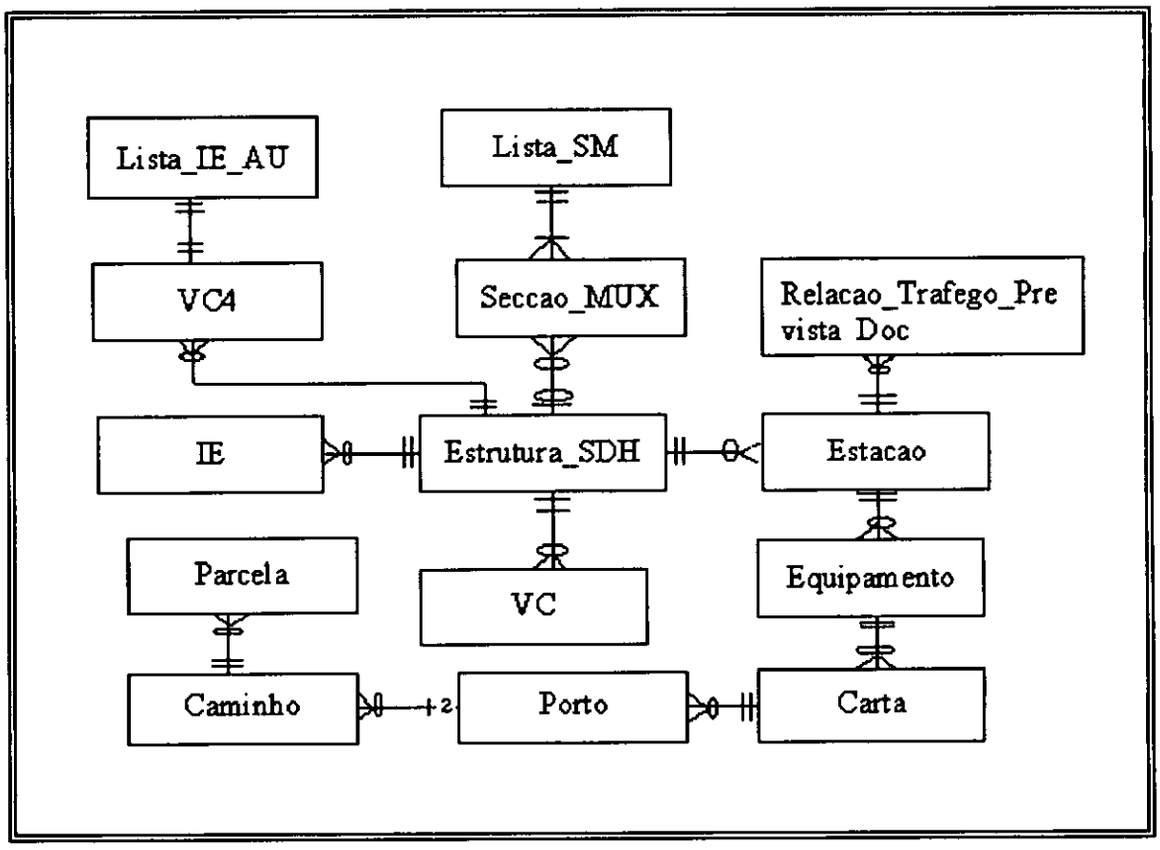
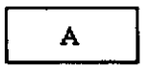
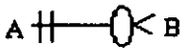


Figura 4.1 Modelo de dados

Legenda



Entidade A



Relação entre Entidades A eB. B ocorre muitas vezes para a A mas não obrigatória B ocorre uma vez para A e é obrigatória

Descrição das Entidades

Estrutura_SDH

È uma partição de rede de telecomunicações, com determinada topologia, função do tipo de conexão de um conjunto de secções MUX (estrutura em anel ou corda).

A estrutura é identificada por uma chave única e uma sigla estrutura.

Estação

Local em que se encontra um conjunto de equipamentos afectos a Estrutura_SDH. A área de estação representa uma área geográfica na qual se pode incluir diferentes estações de telecomunicações.

A Estação está afecta a uma Estrutura e é identificada por uma chave e sigla.

Equipamento

O equipamento aqui considerado é o SDH. È um aparelho de tecnologia SDH de um dado fornecedor. Este, é identificado pela estação a que pertence, pelo endereço e tipo do elemento de rede correspondente.

Carta

É uma partição de um equipamento SDH que é identificada pelo equipamento a que pertence e respectivo grupo.

Porto

Identificado através do equipamento e respectivas cartas tributárias ou agregadas.

Seccção_MUX

É um conjunto de meios cuja função final é o transporte entre dois repartidores digitais consecutivos de um sinal a um ritmo específico. Estes meios incluem processos de modulação em amplitude, fase ou frequência de adaptação ao meio físico de suporte de transmissão. Corresponde também à ligação entre duas estações adjacentes de uma estrutura. Esta ligação é efectuada através dos portos de dois equipamentos. É identificada através das estações terminais, capacidade e nº de ordem.

IE – Interligação de Equipamentos

Representa a interligação de equipamentos, entre cartas agregadas/tributárias ou cartas tributárias/ tributárias. Relativamente aos equipamentos afectos a uma mesma estrutura, esta informação é relevante já que implica actualização dos portos físicos que não estarão disponíveis para ocupação.

VC4 – Virtual *Conteiner* 4

Conceito genérico que corresponde a um débito/capacidade VC4 de transporte existente entre dois pontos da rede.

Lista_IE_AU

Listas das Interligações e os seu respectivos AUs (Unidade Administrativa)

Lista_SM

Lista das Secções-MUX que foram alocados à Estrutura-SDH.

VC – Virtual Conteiner

O *virtual Conteiner* é composto pelos respectivos TUGs e TUs. Se o caminho VC4 estiver estruturado em VC4 os TUG3, TUG2 eTU12 não são considerados. Se o caminho estiver estruturado em VC3 os TUG2 e TU2 não serão considerados. Se o caminho VC4 estiver estruturado em VC12 os TUG3, TUG12 e TU12 serão todos tomados em consideração.

Relação_Tráfego_Prevista_Doc

Lista de relação de tráfego previsto contendo a especificação da Estação de Origem e de Destino bem como os débito.

Caminho

Entidade lógica que corresponde à existência de um débito de transporte existente entre dois Portos de rede sem que exista inserção de sinal em portos intermediários a estes dois Portos.
Identificado através das estações terminais e nº de ordem.

Parcela

Modela a construção do caminho através de parcelas elementares que são interligações de equipamentos através de cartas tributárias.
As Entidades com os seus respectivos atributos estão especificados em ANEXO B.

4.2 DTD Comum

Para a solução do problema de partilha, foi desenhado um DTD comum para as duas aplicações. Qualquer documento xml deverá ser validado com base no DTD comum, de modo a ser interpretado por ambos sistemas. O DTD mantém a relação entre Entidades usando uma relação pai-filho. Se no modelo conceptual a relação entre duas entidade é um para muitos, no DTD o de um fica pai ao que ocorre muitas vezes.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Estrutura_SDH (Estacao, IE, Seccao_Mux, VC4, Relacao_Trafego_Prevista_Doc, Caminho)*>
<!-- Atributos do Elemento Estrutura_SDH-->
<!ATTLIST Estrutura_SDH Sigla_Estr CDATA #REQUIRED
    Designacao CDATA #REQUIRED
    Topologia (A | C) #IMPLIED
    Capacidade (1 | 4 | 16) #IMPLIED
    Protecao (1 | 2 | 3 | 4 | 5 | 6) #IMPLIED>
<!--                    ESTACAO                    -->
<!ELEMENT Estacao (Equipamento)+>
<!ATTLIST Estacao ID_BD_Est ID #REQUIRED
    Sigla_Est CDATA #REQUIRED>
<!--                    Equipamento                -->
<!ELEMENT Equipamento (Carta)+>
<!ATTLIST Equipamento ID_BD_Equip ID #REQUIRED
    Endereco_NE CDATA #REQUIRED
```

```
    Config_Equip (ADM-1 | ADM-4 | ADM-16 | TM-1 | TM-4 | TM-16) #REQUIRED
    Cod_Equip CDATA #REQUIRED
    Fornecedor_Equip (Nec | Alcatel) #REQUIRED>
<!--                    Carta                    -->
<!ELEMENT Carta (Porto)+>
<!ATTLIST Carta ID_BD_Carta ID #REQUIRED
    Posicao CDATA #REQUIRED
    Tipo_Carta (A | T | I | P | I1 | I2) #REQUIRED
    Debito (2Mb | 34Mb | 140Mb | Stm-1 | Stm-4 | Stm-16) #REQUIRED
    Grupo (A | B | C | D | E | W) #REQUIRED
    Tipo_Bastidor CDATA #REQUIRED
    Codificacao (E1 | E2 | E3 | E4 | W1 | W2 | W3 | W4) #REQUIRED>
<!--                    Porto                    -->
<!ELEMENT Porto (#PCDATA)>
<!ATTLIST Porto ID_BD_Porto ID #REQUIRED
    ID_Porto CDATA #REQUIRED
    Tipo_Porto (A | T | I | PA | PT) #REQUIRED
    Estado_Porto (O | V) #REQUIRED>
<!--          Seccao Mux          -->
<!ELEMENT Seccao_Mux (#PCDATA)>
<!ATTLIST Seccao_Mux ID_BD_Sec ID #REQUIRED
    Capacidade (1 | 4 | 16) #REQUIRED
    N_Ordem CDATA #REQUIRED
    ID_BD_EstO IDREF #REQUIRED
    ID_BD_EstD IDREF #REQUIRED>
<!--                IE          -->
<!ELEMENT IE (#PCDATA)>
<!ATTLIST IE ID_BD_IE ID #REQUIRED
    Utilizacao (MEME | MEED) #REQUIRED
    Func_IE (W | P | WP) #REQUIRED
    ID_BD_PortoO IDREF #REQUIRED
    ID_BD_PortoD IDREF #REQUIRED
    ID_Sec_Mux IDREF #IMPLIED>
<!--          VC4          -->
<!ELEMENT VC4 (Lista_IE_AU, Lista_SM, VC)>
<!ATTLIST VC4 ID_BD_VC4 ID #REQUIRED
    N_Ordem CDATA #REQUIRED
    Tipo_Cam CDATA #REQUIRED>
<!ELEMENT Lista_IE_AU (#PCDATA)>
<!ATTLIST Lista_IE_AU ID_BD_IE_AU ID #REQUIRED
    ID_BD_IEO IDREF #IMPLIED
    AU CDATA #REQUIRED
    ID_BD_IED IDREF #IMPLIED
    AU CDATA #REQUIRED>
```

```
<!ELEMENT Lista_SM (#PCDATA)>
<!ATTLIST Lista_SM ID_BD_Lista_SM ID #REQUIRED
    ID_BD_Sec IDREF #REQUIRED
    AU CDATA #REQUIRED>
<!ELEMENT VC (#PCDATA)>
<!ATTLIST VC ID_BD_VC ID #REQUIRED
    TUG3 CDATA #IMPLIED
    TUG2 CDATA #IMPLIED
    TU12 CDATA #IMPLIED>
<!--                Relação de tráfego prevista Doc          -->
<!ELEMENT Relacao_Trafego_Prevista_Doc (#PCDATA)>
<!ATTLIST Relacao_Trafego_Prevista_Doc ID-BD_Rel_Traf ID #REQUIRED
    ID_BD_EstO IDREF #REQUIRED
    ID_BD_EstD IDREF #REQUIRED
    Debito (2Mb | 34Mb | 140Mb | Stm-1 | Stm-4 | Stm-16) #REQUIRED
    Quantidade CDATA #REQUIRED>
<!--          Caminho          -->
<!ELEMENT Caminho (Parcela)+>
<!ATTLIST Caminho ID_BD-Cam ID #REQUIRED
    ID_BD_PortoO IDREF #REQUIRED
    ID_BD_PortoD IDREF #REQUIRED
    Debito (2Mb | 34Mb | 140Mb | Stm-1 | Stm-4 | Stm-16) #REQUIRED>
<!--              Parcela                    -->
<!ELEMENT Parcela (#PCDATA)>
<!ATTLIST Parcela N_Parcela ID #REQUIRED
    Id_VC IDREF #REQUIRED>
```

Para cada documento xml baseado neste DTD deverá ter no cabeçalho a seguinte linha.

```
<!DOCTYPE Estrutura_SDH SYSTEM "Modelo_InterligacaoF.DTD">
```

Onde **Modelo_InterligacaoF** é o nome do ficheiro com extensão DTD gravado no **XML Spy**.

Esta fase consistiu em conceber e testar os programa, em java, que permitem a introdução e selecção de dados na base de dados, usando um documento xml.

4.3 Introdução dos dados do documento xml na base de dados

Os dados na base de dados ARCO podem ser introduzidos pelo utilizador ARCO como também a partir de documento xml contendo dados vindos do DIM-SDH. Neste caso, o sistema recebe o documento e

através de programas em Java introduz os dados do documento nas tabelas da base de dados. A seguir é apresentado um documento xml específico e o respectivo programa que permite a inserção dos seus dados na tabela.

```
<?xml version="1.0"?>
<?xml version="1.0"?>
<!DOCTYPE Estrutura_SDH SYSTEM "Modelo_InterligacaoF.DTD">
<ESTRUTURA>
        <ROW num="1">
                <ID_BD_ESTR>9111</ID_BD_ESTR>
                <SIGLA_ESTR>A990</SIGLA_ESTR>
                <TOPOLOGIA>A</TOPOLOGIA>
                <CAPACIDADE>4</CAPACIDADE>
                <PROTECCAO>3</PROTECCAO>
<DESIGNACAO_ESTR>Anel de cidade de aveiro</DESIGNACAO_ESTR>
        </ROW>
</ESTRUTURA>
```

Documento xml correspondente aos dados da Estrutura_SDH

Os dados podem ser vistos no ambiente SQL da Oracle, ou fazer a ligação da base de dados com o Access através de ODBC para poder ver as tabelas e os respectivos dados em Access.

a) Para além de guardar os dados no ARCO, DIM-SDH pode necessitar de dados provenientes de ARCO. Estes dados podem ser propriamente do ARCO ou os guardados em ARCO enquanto provenientes do DIM. Para um e outro caso, para podermos ler e enviar os dados da base de dados para o DIM-SDH precisamos de seleccionar os dados da base de dados e produzir um documento xml que possa ser enviado e lido pelo DIM-SDH.

A seguir, um ficheiro .xml como resultado de pedido de todos dados referentes a duas primeiras ESTRUTURA_SDH.

```
C:\JDK1.2.1\BIN\java.exe   xml_from_bd
Working Directory - C:\Dim-SDH\
Class Path - .;C:\Program
Files\kawaclasses.zip;c:\jdk1.2.1\lib\tools.jar;c:\jdk1.2.1\jre\lib\rt.ja
r;c:\jdk1.2.1\jre\lib\i18n.jar;C:\Dim-SDH\classes12_01.zip;C:\Dim-
SDH\classes111.zip;C:\Dim-SDH\oraclexmlsql.jar;C:\Dim-SDH\xmlparserv2.jar
OUTPUT IS:
<?xml version = '1.0'?>
<?xml-stylesheet href="estrutura_sdh.xsl" type="text/xsl"?>
<estrutura_sdh>
   <ROW num="1">
      <ID_BD_ESTR>2345</ID_BD_ESTR>
      <SIGLA_ESTR>A001</SIGLA_ESTR>
      <TOPOLOGIA>A</TOPOLOGIA>
      <CAPACIDADE>4</CAPACIDADE>
      <PROTECCAO>2</PROTECCAO>

 <DESIGNACAO_ESTR>ANEL DA BEIRA LITORAL</DESIGNACAO_ESTR>
   </ROW>
   <ROW num="2">
<ID_BD_ESTR>2343</ID_BD_ESTR>
      <SIGLA_ESTR>A002</SIGLA_ESTR>
      <TOPOLOGIA>C</TOPOLOGIA>
      <CAPACIDADE>1</CAPACIDADE>
      <PROTECCAO>3</PROTECCAO>
      <DESIGNACAO_ESTR>CORDA DA BEIRA LITORAL</DESIGNACAO_ESTR>
   </ROW>
</estrutura_sdh>


Process Exit...
```

Em Anexo C podemos ver os programas que permitem a manipulação dos ficheiros .xml.

Para a programação foram usadas as seguintes linguagens de programação:

- XML – para definição da estrutura do documento e do DTD;
- Java – para ligação entre os ficheiros XML e a base de dados Oracle;
- SQL – para fazer as selecções na base de dados.

Para além das linguagens de programação, foram utilizadas aplicações para desenvolvimento de várias componentes deste projecto:

- Internet Explore 4.0 – para a navegação na Web;
- XML Spy – para a criação de programas com extensão .xml e .dtd;
- Kawa – ambiente para redigir, compilar e executar programas Java.
- Oracle 8i – Sistema de gestão de base de dados;
- Microsoft Oficce 97 – Ferramenta usada para desenvolvimento do relatório.

Capítulo V

Conclusões e Recomendações

O presente trabalho esteve apenas centrado no estudo de XML e na definição da plataforma para suporte de interacção entre os sistemas de provisão da rede - ARCO, e de planeamento de rede - SDH. Este ainda não é um trabalho acabado, carece de desenvolvimento e implementação.

Muitas são as funcionalidades existentes nesta nova tecnologia de gestão de dados. A tecnologia XML pretende fazer dos dados o que a linguagem Java fez pela programação, que é tornar os dados independentes da plataforma e do vendedor.

Depois da W3C ter começado a trabalhar na XML, à cerca de dois (2) anos atrás, muitas linguagens baseados em XML, tais como as de Transações Financeiras, Gráficos Vectoriais e Multimédia, Avaliação de locais Web, tomaram lugar, apesar de algumas ainda serem "*Work Draft*".

O Comércio Electrónico bem como as aplicações necessárias para o implementar, crescerão à grande velocidade pois existem muitas perspectivas de XML vir a fornecer uma autenticação de assinaturas digitais mais segura.

O uso de XML nos documentos electrónicos traz também muitas vantagens, uma vez que XML define a natureza dos dados e a decisão de como apresentar o documento na Web cabe ao *Browser*.

Se se pretender publicar uma página, e os seus dados precisarem de uma boa estrutura, o melhor é usar XML ao invés de HTML;

A integração de aplicações remotos bem como a publicação de seus serviços na Web, pode ser facilitada pelo uso da tecnologia XML, mesmo que as aplicações emergentes não sejam compatíveis;

Capítulo VI
BIBLIOGRAFIA

6.1 Bibliografia Referenciada

[URL –1] http://www.tijuca.com.br/rodolfo/telecom.htm

[URL – 2] ZDNet Portugal – Semana Informática - Actualidades
http://194.65.81.1/Zdnet/htdocs/pcmagazine/analise/9904/premiuos6.shtml.

[URL – 3] http://www.dsi.uminho.pt/~vjs/assuntos/dhtml/poten.htm

[URL – 4] Benefits of using XML
http://developerlife.com/xmlbenefits/default.htm

[URL – 5] http://www.dsc.ufpb.br/~helder//java/cap-1.html

[URL – 6] About Oracle XML Products
http://technet.oracle.com/tech/xml/info/htdocs/otnwp/about_oracle_xml_products.htm

[URL – 7] Using XML in Oracle Database Applications
http://technet.oracle.com/tech/xml/info/htdocs/otnwp/xml_data_exchange.htm

[URL – 8] Using XML and Relational DataBase for Internet Aplications
http://technet.oracle.com/tech/xml/info/htdocs/relational/index.htm

[URL – 9] Segurança: o firewall é suficiente? http://i2000.intermol.com.br/virus-segurança/vs-15101999-1.htm

[URL –10] http://www.w3c.org/tr/xmldsig-core/

[URL – 11] http://www.uol.com.br/idgnow/busca/130799b28.htm

[Campos, 1999] Campos, L.M.(Setembro 1999). " Oracle 8I- Curso Completo", 1ª edição. 940 pag., Lisboa - FCA

[Pereira, 1998] Pereira, J.L. (Outubro, 1998). "Tecnologia de Base de Dados", 2ª edição. 494 pag., Lisboa-FCA

6.2 Bibliografia não Referenciada

Cortesão L. Miguel (1999). Estudo Sobre XML, PT Inovação

Oracle Announces Industry's First Deveolpment Tool with End-To-End XML Suprt
http://www.oracle.com/corporate/press/14356.html

XML and the Oracle Internet Platform
http://technet.oracle.com/tech/xml/

Simplified XML Development
http://technet.oracle.com/tech/xml/parser_java/listing.htm

XML Parser for java
http://technet.oracle.com/tech/xml/parser_java/htdocs/relnotes.htm

The XML Toolkit
http://csmctmto.interpoint.net/didx/xml.html

Architag Events: XML and E-Cvopmmerce Seminar
http://architag.com/events/xmlecommerce/

Parser XML
http://www.iscm.sc.usp.br/.../icmc-usp-disciplina-hm-seminario-www-paerser-xml.htm

XSL Transformations (XSLT)
http://www.w3.org/1999/08/ED-xslt-19990813.html

Oracle Technology Network | XML
http://technet.oracle.com/tech/xml

BellFone Telecomunicação
http://www.bellfone.com.br/HistTelefone.htm

http://www.icon-is.com/e/prod/sw/xml_main.asp

Modeling Relational Data in XML
http://apps.xmlschema.com/white_papers/modeling.htm

Wiennr J. (agosto 1993)
http://www.dei.isep.pt/~andre/normas/wiener.htm

A.I.S.A. – Transferência de Arquivos
http://www.aisa.com.br/tranfe.html

Anexos

Anexo A: Tecnologia PDH e SDH

A.1 Breves considerações sobre as tecnologias PDH e SDH

A.1.1 Introdução

Com o desenvolvimento da informática, em particular a área de Internet, os usuários de transmissão de dados tornaram-se mais dependentes de comunicações eficientes e, desta forma houve uma explosão na exigência para serviços de telecomunicações mais avançados.

Serviços tais como videoconferência, acesso remoto a dados, transferência de arquivos multimédia requerem uma rede de telecomunicações com disponibilidade de faixa maior.

A complexidade da rede baseada em transmissão de plesiócrono, impossibilita os usuários desta satisfazerem essas exigências. A hierarquia correspondente à transmissão de pesiócronos, PDH (*Plesiochronous Digital Hierarchy*) tinha sido desenvolvido para responder as exigências da simples telefopnia e, não é ideal para as necessidades actuais, serviços de faixa larga.

Devido às facilidades da utilização de largura de banda de faixa maior e de baixo custo, fornecimento dos serviços de comunicações mais eficiente aos clientes, surge então, como resposta aos problemas de PDH, a SDH(*Sinchronous Digital Hierarchy*).

A SDH , com as suas poderosas capacidades de gestão de uma rede fornecerá:

- Melhoramento no controlo das redes de transmissão;
- Melhoramento no restauro, e capacidade de reconfiguração das redes que irão resultar em maiores disponibilidade dos serviços existentes bem como numa enorme rapidez no fornecimento de novos serviços;
- Quebrar a barreira existente entre os equipamentos multiplexados e os equipamentos da linha para a fibra óptica, englobando na gestão de multiplexagem o processamento de informação necessária ao encaminhamento e supervisão de um canal de transmissão digital por fibra óptica.

Desta forma a SDH pode adequar-se às necessidades actuais na área de telecomunicações e problemas relacionados com a transmissão de *plesiócrona*, tais como diferenças hierarquias de multiplegem , pouca flexibilidade da rede, falta de capacidade para monitorização de performance de suas ligações, etc, deixam de atormentar os fornecedores de serviços de telecomunicações.

Contudo, a SDH não é uma solução tecnológica isolada, mas sim um " *up-grad*" das soluções já existentes da PDH. Desta forma, a instalação de um sistema síncrono de transmissão é argumentada pela sua capacidade de interacção com outros sistemas. A SDH define uma estrutura que capacita sinais *plesiócronos* a serem combinados dentro de um quadro normal da SDH o que protege os investimentos em equipamentos síncronos que existem ou poderão ser instalados segundo as necessidades.

A.1.2 Padrões Universais:

a) Os sistemas de multiplexagem SDH permitem que os sinais digitais atravessem fronteiras sem conversão para outro padrão;
b) A velocidade de transmissão básica é 155.520 Mbps e, as restantes velocidades são múltiplos inteiros deste (622.080 Mbps, 2488.320 Mbps, 9953.280 Mbps) correspondentes a STM-1, STM-4, STM-16 e STM-64, respectivamente.

Um STM-1 (Módulo Síncrono de Transporte de 155.520 Mbps), pode ainda carregar um número de sinais de índices mais baixos como carga útil, assim podem existir cargas PDH carregados sobre uma rede síncrona.

A.1.3 Tramas SDH

É possível efectuar multiplexagem de sinais STM-n num STM-m (m>n). Este processo é feito por "*byte interleaving*" , multiplexagem *byte a byte* de cada um dos sinais tributários STM-n.

Blocos Constituintes

- ***Conteiner*-n (Cn , n=1 . . 4)**

É uma estrutura de, que contêm informação de "*PayLoad* " para o VC (*Virtual Conteiner*). Para cada VC existe um C . O contentor possui informação de controle adicional –POH- *Path OverHead.*

Assim temos:

C_3 e C_4 Contentores de alta ordem e C_{12} de baixa ordem

HO-POH POH de alta ordem
LO-POH POH de baixa ordem

- ***Virtual Conteiner*-n (VC-n , n=1 . . 4)**

O contentor e o seu respectivo POH formam um VC. O VC é uma estrutura usada para estabelecer ligações de caminho no SDH. Os VCs são organizados num bloco cuja estrutura de trama se repete em cada 125μs ou 50μs

- **Unidades Tributárias-n (TU-n, n=1, 2, 3)**

É constituído por um VC-n e um *TU Pointer*. O TU providencia adaptação entre o estado de caminho de baixa ordem e o de alta ordem.
Consiste em "*PayLoad* " (Contentor virtual de baixa ordem) e um " *TU-Pointer* " que indica o deslocamento desse "*Payload*" relativamente ao VC de alta ordem.

- **Grupo de Unidade Tributária (TUG)**

É uma estrutura formada por TUs idênticos

Exemplo:

Um TUG-2 é a multiplexação de 3 TU-12 que ocupam posições fixas num VC-3 ou VC-4

Um TUG-3 é a multiplexação de 7 TU-2 ou de 1 único TU-3

- **Unidade Administrativa (AU)**

É responsável pela adaptação entre camadas de via de ordem superior e camada de multiplexação .
Consiste de um VC mais um ponteiro de Unidade Administrativa (AUPointer).
AUPointer indica a posição do VC com relação ao quadro STM-1 e têm posição fixa dentro do quadro.
O SDH define o número de contentores e cada um corresponde a uma taxa existente de *plesiócrono*.
A informação de um sinal *plesiócrono* é mapeada no *conteiner*.

Em suma, a SDH foi projectada para dar mais eficiência e confiabilidade na gerência da rede e apoiar serviços tais como Redes Metropolitana de Área (MANs) e Rede Digital de Serviços Integrados.

Anexo B: Descrição das Entidades do Sistema

Estrutura_SDH

Nome do atributo	Tipo de dados	Descrição
ID_BD_Estr	Number(4)	Identificador único da tabela Estrutura
Sigla_Estr	VarChar2(4)	Identificador único da estrututra
Designacao_Estr	Varchar2(30)	Designação da estrutura
Topologia	VarChar2(1)	Anel(A)/ Corda(C)
Capacidade	Number(2)	1, 4, 16: Stm1(1)/ Stm4(4)/Stm16(16)
Proteccao	Number (2)	1, 2, 3, 4 ou 5: SNCP- 2 fibras(1) / MS-SPRING- 2 fibras (2)/ MS-SPRING 4 fibras(3)/ MSP 1+1(4) /MSP 1+0(5)/ SNCP- 4 fibras(6)
Estacao		Entidade Filha
Seccao_Mux		Entidade Filha
VC		Entidade Filha
VC4		Entidade Filha

Estação

Nome do atributo	Tipo de dados	Descrição
ID_BD_Est	Number(6)	Identificador único da estação
Sigla_Est	Varchar2(7)	Sigla da Estação
Equipamento		Entidade filha

Equipamento

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_Equip	Number(4)	Identificador único do equipamento
Endereço_NE	Varchar2(6)	Endereço do elemento de rede no centro de gestão
Config_Equip	Varchar2(6)	Tipo de configuração ADM-4/ADM-16/TM-1/TM-16/ ADM-1 / TM-4
Cod_Equip	VarChar2(10)	Designação abreviada pela qual é conhecida o Equipamento
Fornecedor_Equip	Varchar2(10)	Forncedor do equipamento Nec/Alcatel
Carta		*Entidade filha*

Carta

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_Carta	Number(6)	Identificador único da carta
Posição	Number(2)	Posição da carta no bastidor
Tipo_Carta	Varchar2(2)	Designação abreviada que identifica a carta: **A**gregada, **T**ributária, **I**nterligacao, **P**roteccao, **I1**-I. ao Bast. EXT1, **I2**-I. ao Bast. EXT2
Débito	Varchar2(6)	2Mb, 34Mb, 140Mb, Stm-1, Stm-4, Stm-16
Grupo	Varchar2(4)	Grupo em que uma dada carta se insere: A | B | C | D | E | W
Tipo_Bastidor	Varchar2(4)	Tipo de Bastidor
Codificacao	Varchar2(6)	Codificação utilizada pelos portos da carta agregada: E1 | E2 | E3 | E4 | W1 | W2| W3| W4
Porto		Entidade filha

Porto

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_Porto	Number(8)	Identificador único do Porto
ID_Porto	Number(2)	N° do Porto na Carta
Tipo_Porto	Char(2)	**A**gregado | **T**ributário | **I**nterligação| **PA** | **PT**
Estado_Porto	VarChar2(4)	**O**- em caminho ou reservado | **V**- Vago
Caminho		Entidade filha

Seccao_Mux

Representa a existência de uma dada capacidade SDH entre duas estações adjacentes

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_Sec	Number(6)	Identificador único da Seccao_Mux
Capacidade	Number(2)	**1**-155 MB, **4**-622MB, **16**- 2.5GB
N_Ordem	Number(2)	Nº de ordem da seccao mux entre um dado par de extremos
ID_BD_EstO	Number(4)	*IDRef* referência à elemento Estacao
ID_BD_EstD	Number (4)	*IDRef* referência à elemento Estacao

* O tipo de suporte que se utiliza é sempre a Fibra Óptica (FO).

IE **Modeliza uma ligação física entre dois Portos**

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_IE	Number(6)	Identificador único de IE
Utilizacao	Varchar2(4)	MEME, MEED
Func_IE	Varchar2(2)	**Working, Protecting, WP-Work/Protecting**
ID_BD_PortoO	Number(8)	*IDRef* referência à elemento Porto
ID_BD_PortoD	Number(8)	*IDRef* referência à elemento Porto
ID_BD_Secc ?	Number(6)	*IDRef* referência à elemento Secção Mux *

* O ID_BD_Secc não existe para **MEME** e é mandatário para **MEED**

MEME – Representa a interligação de equipamentos afectos à Mesma Estrutura e dentro da Mesma Estação.
MEED – Representa a interligação de equipamentos afectos à Mesma Estrutura e dentro de Estações Diferentes.

VC4

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_VC4	Number(6)	Identificador único de VC4
N_Ordem	Number(2)	Nº de ordem do caminho VC4 de acordo com a codificação
Tipo_Cam	Varchar2(1)	SDH(S) | PDH(P), por defeito o valor é " S"
Lista_IE_AU		Entidade filha de VC4

Lista_IE_AU

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_IE_AU	Number(6)	Identificador único de VC4
ID_BD_IEO	Number(6)	Lista de IE de Origem
AU	Number(2)	AU correspondente à IE de Origem
ID_BD_IED	Number(6)	Lista de IE de Destino
AU	Number(2)	AU correspondente à IE de Destino

Lista_SM

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_SM	Number(6)	Identificador único de VC4
AU	Number(2)	AU correspondente à lista de Secção Mux

VC

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_VC	Number(8)	Identificador único de VC
TUG3	Number(1)	Posição de contentor da trama SDH, correspondente a um débito de 34 Mb/s. Válido entre 1 e 3
TUG2	Number(1)	Posição de contentor da trama SDH, correspondente a um grupo de unidades tributárias de 2Mb/s. Válido entre 1 e 7
TUG12	Number(1)	Posição de contentor da trama SDH, correspondente a uma unidades tributárias de 2Mb/s. Válido entre 1 e 3

- Se o caminho VC4 estiver estruturado em VC4 os campos TUG3, TUG2 e Tu12 estarão vazios;
- Se o VC4 estiver estruturado em VC3 os TUG2 ew TU12 respectivos estão vazios;
- Se o caminho VC4 estiver estruturado em VC12 os campos TUG3, TUG2 e Tu12 estarão todos preenchidos e a tabela conterá, por VC4, 63 registos.

Relaccao_Trafego_Prevista_Doc

Nome do atributo	**Tipo de dados**	**Descrição**
Id_Rel_Traf	Number(4)	Identificador único da relaccao de tráfego
ID_BD_EstO	Varchar2(4)	*IDRef* referência à elemento Estacao
ID_BD_EstD	Varchar2(4)	*IDRef* referência à elemento Estacao
Débito	Varchar2(6)	2Mb, 34Mb, 140Mb, Stm-1, Stm-4, Stm-16

Caminho

Nome do atributo	**Tipo de dados**	**Descrição**
ID_BD_Cam	Number(4)	Identificador único do caminho
ID_BD_PortoO	Number(8)	*IDRef* referência à elemento Estacao
ID_BD_PortoD	Number(8)	*IDRef* referência à elemento Estacao
Debito	Varchar2(6)	

Parcela

Nome do atributo	**Tipo de dados**	**Descrição**
N_Parcela	Number(4)	Identificador único da parcela
ID_BD_VC	Number(4)	Lista de VC's

Anexo C

C.1 O código que introduz os dados provenientes de um documento xml para a tabela da base de dados

----------Ficheiro arcoxmller.java ----------

```
import java.io.*;
import oracle.xml.sql.dml.*;
import java.sql.*;
import oracle.jdbc.driver.*;
import oracle.jdbc.*;
import java.net.*;

public class arcoxmller
{
        public static void main(String args[]) throws SQLException
        {


                String re;
                try
                  {


                        //ler dados do teclado


                   byte bArray[]=new byte[128];

                   System.out.print(" Escreva a Tabela: ");

                        //a instrução seguinte lê para uma matriz de bytes
                        System.in.read(bArray);
```

```
                byte bArray1[]=new byte[128];

            System.out.print(" Escreva o ficheiro que deseja inserir na tabela : ");

                //a instrução seguinte lê para uma matriz de bytes
                System.in.read(bArray1);

                // Converter a matriz numa String
                // antes de tentar apresentá-la

                String s=new String(bArray,0);
            String r=new String(bArray1,0);
            re=s;


        String tabName=re;
            String fileName=r;

            DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());

            //inicilizar a coneccao

            Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:@192.168.89.20:1521:arco2","arco_v11T","arco");

            //inserir os dados a partir do ficheiro xml para a tabela da base de dados

            OracleXMLSave save=new OracleXMLSave(conn,re/*tabName*/);
            URL url=save.createURL(fileName);
            int rowCount=save.insertXML(url);

        System.out.println(" Successo:" +rowCount+
                " coluna inserida na nabela  "+ tabName);

            conn.close();
            }
```

```
        catch(IOException ioe)
                        {
                                System.out.println(ioe.toString());
                                ioe.printStackTrace();
            }
                }
        }
```

2. O código que cria um documento XML a partir de dados da base de dados

---------- ficheiro xml_from_bd.java----------

```
import java.sql.*;
import java.math.*;
import oracle.xml.sql.query.*;
import oracle.jdbc.*;
import oracle.jdbc.driver.*;

public class xml_from_bd
{
public static void main(String args[]) throws SQLException
{
    String tabName="estrutura_sdh";
    String user="ARCO_V11T/ARCO";

DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
// Inicializar aconeccao
Connection conn=
DriverManager.getConnection("jdbc:oracle:thin:@192.168.89.20:1521:arco2","arco_v11T","arco");

//inicializar a OracleXMLQuery

   OracleXMLQuery qry=new OracleXMLQuery(conn,"select * from "+tabName);
```

```
//EStruturar o documento XML gerado
qry.setMaxRows(2);
qry.setRowsetTag("estrutura_sdh");
qry.setRowTag("ROW");
qry.setStyleSheet("estrutura_sdh.xsl");

//get Document in string format
String xmlString=qry.getXMLString();

// imprimir o documento
System.out.println("OUTPUT IS:\n"+xmlString);
}
}
```

Anexo D: Glossário

Assinatura Digital - É um mecanismo de criptografia que pode ser aplicado a um ficheiro ou qualquer documento. Ela é uma parte de um texto computado que é encriptado e enviado com a mensagem do texto. O receptor decripta a assinatura e recompõe o texto recebido.

API's - Aplication Program Interfaces – Um formato de linguagem e mensagem usado por um programa de aplicação para comunicar-se com um sistema operativo ou outro programa de sistema tal como um sistema de gestão de bases de dados.

Browser ou Navegador Internet é o programa cliente utilizado para acessar informações em Servidores Web. Em outras palavras é o programa que possibilita a visualização de páginas com informações distribuídas por meio de texto, sons e imagens (multimidia), que são disponibilizadas na Internet

CSS-Cascading Style Sheets-É uma simples linguagem declarativa que permite aos autores e utilizadores implementar informações de estilo (fonte, espaçamento, cor,...) em documentos HTML ou XML.

FAQs ou Frenquently Asked Questions: São questões frequentemente feitas. É um recurso muito útil no atendimento aos clientes pela Intenet, já que antecipa as perguntas dos clientes e as respostas sob a forma de págiona Web

FTP-File Transfer Protocol - É o protocolo responsável de enviar arquivos entre computadores pela Internet

Firewall - Parede de Fogo. Medida de segurança que pode ser implementada para limitar o acesso de terceiros a um determinada rede ligada à Internet. Os mecanismos de implementação são variados, percorrendo variados tipos de controlo por software ou hardware. Num caso limite, a única coisa que uma firewall poderia deixar passar de um lado (rede local) para o outro (resto da Internet) era o correio electrónico (podendo mesmo filtrar correio de/para determinado sítio).

Hyperlinks - São palavras ou ilustrações pré-estabelecidas como pontos de saltos. Quando clicadas, originam a transferência para op outro assunto ou página Web. São comumente chamados **links;**

HTML ou HyperText Markup Language - É a linguagem baseada em texto usada para construir as páginas da web e os websites.

HTTP ou HyperText Transfer Protocol - É o protocolo padrão que permite a transferência de dados na Web entre os servidores e os browsers. É este protocolo que permite os saltos de uma página para outra através dos links do hipertexto.

Internet - É uma rede mundial de computadores ligados entre sí, que se servem de protocolos comuns para comunicarem entre si.

IP - Internet Protocol. Um dos protocolos mais importantes do conjunto de protocolos da Internet. Responsável pela identificação das máquinas e redes e encaminhamento correcto das mensagens entre elas.

JavaScript - é uma linguagem compacta para desenvolvimento de aplicações para a Internet.

Netscape - Um programa (browser) para o WWW. Sucessor do Mosaic e desenvolvido pela mesma equipa de programadores, o Netscape evolui mais rapidamente e está-se a tornar no browser de WWW mais usado.

Parser – Um programa em Java usado para analisar se um documento é *Well-Formed* ou Válido.

Protocolo - Um protocolo é para os computadores o que uma linguagem (língua) é para os humanos. Dois computadores para poderem transferir informações entre si devem utilizar o mesmo protocolo (ou ter um terceiro que perceba os dois protocolos e faca a tradução).

TCP/IP - Conjunto de protocolos da Internet, definindo como se processam as comunicações entre os vários computadores. Pode ser implementado em virtualmente qualquer tipo de computador, pois é independente do hardware. Geralmente, para alem dos protocolos TCP e IP (porventura os 2 mais importantes), o nome TCP/IP designa também o conjunto dos restantes protocolos Internet: UDP, ICMP, etc.

URL - Uniform Resource Locator. Localizador Uniformizado de Recursos. Método de especificação de um determinado recurso na Internet, seja ele obtido por FTP, News, Gopher, Mail, HTTP, etc.

Pretende uniformizar o maneira de designar a localização de um determinado tipo de informação na Internet.

Web - Em português, teia. Abreviatura para designar o World-Wide-Web.

WWW -World Wide Web - É o ambiente multimédia da Internet, a reunião de texto, imagem, som, vídeo e movimento na Internet

W3C-WWW Consortium - É uma organização composta por cerca de 275 membros incluindo companhias, organizações sem fins lucrativos, grupos da indústria e agências governamentais de todo o mundo. Podemos considerâ-la o mais próximo de uma entidade reguladora da Web que existe no momento e a entidade em melhores condições de determinar a estrutura da Web para o século XXI