

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**  
(Bacharelado)

**PROTÓTIPO DE UM AMBIENTE DE APOIO AO ENSINO DA  
ARQUITETURA TCP/IP**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE  
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA  
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA  
COMPUTAÇÃO — BACHARELADO

**ROBERTO FABIANO FERNANDES**

BLUMENAU, NOVEMBRO/2001

2001/2-43

# **PROTÓTIPO DE UM AMBIENTE DE APOIO AO ENSINO DA ARQUITETURA TCP/IP**

**ROBERTO FABIANO FERNANDES**

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Francisco Adell Péricas — Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

## **BANCA EXAMINADORA**

---

Prof. Francisco Adell Péricas

---

Prof. Sérgio Stringari

---

Prof. Carlos Eduardo Negrão Bizzotto

# AGRADECIMENTOS

Agradeço ao meu orientador Prof. Francisco Adell Péricas pela dedicação, paciência, encorajamento e orientação para a realização deste trabalho. Aos professores Carlos Eduardo Negrão Bizzotto, Sérgio Stringari e Everaldo A. Ghral, pelos ensinamentos e experiências que ajudaram-me a construir a base de conhecimentos para realização deste trabalho e a todos os professores do Curso de Ciências da Computação pela ajuda e pelas oportunidades oferecidas durante todo curso.

Aos meus amigos Eduardo Kohler e Gilvani Kisner com quem compartilhei vitórias e derrotas nestes anos de graduação.

E por fim, à minha noiva Catia Silene Possamai pelo auxílio em todos os momentos difíceis que passei até a conclusão deste curso de graduação.

# SUMÁRIO

AGRADECIMENTOS .....	III
SUMÁRIO.....	IV
LISTA DE FIGURAS .....	VI
LISTA DE TABELAS .....	VIII
RESUMO.....	IX
ABSTRACT .....	X
1 INTRODUÇÃO .....	1
1.1 OBJETIVOS .....	2
1.2 ESTRUTURA DO TRABALHO.....	3
2 FUNDAMENTAÇÃO TEÓRICA.....	4
2.1 A INTERATIVIDADE NO PROCESSO ENSINO-APRENDIZAGEM .....	4
2.2 UTILIZAÇÃO DO COMPUTADOR NO PROCESSO ENSINO-APRENDIZAGEM .....	5
3 REDES DE COMPUTADORES .....	9
3.1 MODELOS DE REFERÊNCIA .....	10
3.1.1 MODELO DE REFERÊNCIA OSI.....	10
3.1.2 ARQUITETURA TCP/IP .....	12
4. TÉCNICAS E FERRAMENTAS UTILIZADAS .....	29
4.1 OOHDM (OBJECT ORIENTED HIPERMEDIA DESIGN METHOD).....	29
4.1.1 DESIGN CONCEITUAL.....	31
4.1.2 DESIGN NAVEGACIONAL .....	32
4.1.3 DESIGN ABSTRATO .....	34

4.1.4 ABSTRACT DATA VIEW (ADV).....	34
4.2 MACROMEDIA DIRECTOR.....	35
4.2.1 APRESENTAÇÃO DO DIRECTOR.....	36
4.3 LINGO.....	43
4.3.1 OS SCRIPTS.....	44
4.3.2 EVENTOS.....	45
4.3.3 LISTAS.....	45
4.4 PERFORMANCE.....	46
5 DESENVOLVIMENTO E ESPECIFICAÇÃO DO TUTORIAL.....	47
5.1 DIAGRAMAS DA OOADM.....	48
5.1.1 DIAGRAMA DO PROJETO CONCEITUAL.....	48
5.1.2 MODELO NAVEGACIONAL.....	49
5.2 IMPLEMENTAÇÃO E FUNCIONAMENTO DO TUTORIAL.....	50
6 CONCLUSÃO.....	57
6.1 SUGESTÕES.....	57
7 REFERÊNCIAS BIBLIOGRÁFICAS.....	59

## LISTA DE FIGURAS

FIGURA 01 - As sete camadas do modelo de referência OSI.....	11
FIGURA 02 - As quatro camadas da Arquitetura TCP/IP.....	13
FIGURA 03 – Colisão em redes de banda Base.....	18
FIGURA 04 – Fluxograma de funcionamento do CSMA/CD.....	19
FIGURA 05 – Protocolos da Camada de Rede.....	21
FIGURA 06 – Formato do Datagrama IP.....	22
FIGURA 07 – Exemplo de Roteamento.....	23
FIGURA 08 – Ponteiros de uma Janela TCP.....	26
FIGURA 09 – Modelo de Janela Deslizante.....	26
FIGURA 10 – Transferência de Arquivos entre Sistemas de Arquivos Remotos e Local.....	27
FIGURA 11 – Modelo conceitual com representação de duas classes relacionando-se com um sub-sistema.....	32
FIGURA 12 – Representação de processos, com as visões sobre modelo conceitual, e o modelo de interface, interagindo neste conjunto.....	32
FIGURA 13 – Composição do ADV.....	35
FIGURA 14 – Tela de abertura do Director 8.0.....	36
FIGURA 15 – Stage.....	36
FIGURA 16 – Propriedades do Stage.....	37
FIGURA 17 – Membros do Cast.....	37
FIGURA 18 – Canais do Score.....	40
FIGURA 19 – Control Panel.....	41
FIGURA 20 – Módulos do Tutorial .....	48
FIGURA 21 – Projeto Conceitual.....	49

FIGURA 22 – Modelo Navegacional.....	50
FIGURA 23 – Tela de Entrada do Tutorial.....	51
FIGURA 24 – Tela da Arquitetura TCP/IP.....	52
FIGURA 25 – Mudança do formato do mouse.....	52
FIGURA 26 – Tela da Camada de Aplicação.....	53
FIGURA 27 – Comando PUT.....	54
FIGURA 28 – Conexão de Controle.....	54
FIGURA 29 – Conexão de Dados.....	55
FIGURA 30 – Funcionamento do CSMA/CD.....	55
FIGURA 31 – Detecção de Colisão no CSMA/CD.....	56

## LISTA DE TABELAS

TABELA 1 – Tipos de Elementos do Cast.....	38
TABELA 2 – Barra de Ferramentas do Director 8.0.....	41
TABELA 3 – Termos Lingo.....	44
TABELA 4 – Eventos e Tipos de Script.....	45
TABELA 5 – Sintaxe Para a Criação de Listas.....	46



## **RESUMO**

Este trabalho visa destacar a importância do uso de ambientes interativos multimídia no processo ensino-aprendizagem através da especificação e implementação de um protótipo de um ambiente de apoio ao ensino da Arquitetura TCP/IP. No tutorial desenvolvido há demonstrações de alguns protocolos pertencentes a cada uma das quatro camadas da arquitetura TCP/IP, onde o usuário terá uma apresentação multimídia interativa sobre o seu funcionamento.

## **ABSTRACT**

This work intends to highlight the importance of using multimedia interactive environments in the teaching-learning process through the specification and implementation of a software for supporting computer networking education based on the TCP/IP reference model. In the developed tutorial there are simulation of some protocols belonging to each of the four layers of the TCP/IP architecture, where the user will have a multimedia interactive presentation about its operation.

# 1 INTRODUÇÃO

Lida-se hoje com um elevado volume de informações, que chegam a todo instante, através dos mais diferentes meios. Algumas dessas informações têm relevância no momento em que se apresentam. Outras, vão fazer sentido somente quando combinadas e integradas a outras informações. A forma como estes conhecimentos se apresentam pode permitir fácil assimilação do assunto ou torná-lo obscuro por muito tempo, dependendo das ferramentas utilizadas e da combinação adequada dos meios empregados.

Todo este cenário, composto por pessoas, informações e ferramentas, reflete o ambiente educacional disponível atualmente, onde o importante é a visão global da área de estudo ou pesquisa, seu aprofundamento em sub-áreas e a compreensão de seus mecanismos e processos, e principalmente, o desenvolvimento da capacidade de lidar com informações dinâmicas, que se enquadram e se ajustam conforme a necessidade e os surgimento de novos elementos.

O mais importante no processo de aprendizagem não é a detenção de um conjunto de teorias, mas sim, os conhecimentos básicos, o saber buscar informações necessárias no momento em que se aplicam a forma, como utilizar as teorias e os fundamentos nas soluções práticas e como relacionar os resultados da pesquisa com os estudos realizados por outros.

Na pedagogia o aprender-fazendo tem tido sua eficácia comprovada, sobretudo em se tratando de ciências exatas e tecnológicas. As teorias Piagetiana e Skinneriana, enquanto distantes no trato da questão ensino-aprendizagem, são uníssonas quanto a importância da atividade prática neste processo (Skinner, 1981).

Para enriquecer atividades pedagógicas, fazendo com que alunos e pesquisadores participem ativamente do processo, as ferramentas utilizadas e a maneira como são aplicadas são fundamentais para uma percepção mais ampla, rica não só em conhecimentos, mas em experiências. O aluno não se limita a sentar e ouvir o detentor do conteúdo, mas expõe suas dúvidas e exercita.

As ferramentas existentes são as mais diversas possíveis e podem ser utilizadas para tornar mais atrativo e interativo o aprendizado. A tecnologia tem sido bastante utilizada e aplicada em favor da educação, principalmente os computadores, que se tornaram

praticamente indispensáveis no meio acadêmico, como ferramenta de auxílio a trabalhos e pesquisas. Software educativos e tutoriais são largamente utilizados no ensino das diferentes ciências, propiciando um ambiente onde é possível demonstrar virtualmente o que acontece na realidade.

Tutoriais e softwares educativos poderiam ser mais amplamente explorados por mestres e instrutores, em seu uso como ferramenta, e por programadores e analistas, pela sua importância em ambientes de estudo e pesquisa, visto que sabemos ser possível simular qualquer coisa no computador e, se bem projetado, adicionar, modificar e relacionar informações com facilidade, fator que, como colocado a princípio, é de fundamental importância no aprendizado.

Juntos, experiência do profissional da educação, recursos didáticos tradicionais e recursos de multimídia e hipermídia, poderão ilustrar clara e eficazmente um assunto estudado, não importa a qual área pertença. A pesquisa realizada visa salientar a importância do uso de um ambiente interativo de ensino, através do desenvolvimento de um tutorial que possa ser utilizado como ferramenta de auxílio ao aprendizado das tecnologias envolvidas nas redes de computadores, em especial as da arquitetura TCP/IP.

## **1.1 OBJETIVOS**

O objetivo do trabalho é destacar e caracterizar a importância do uso de ambientes interativos multimídia no processo ensino-aprendizagem através da especificação e implementação de um software para o apoio ao ensino de redes de computadores. Neste trabalho é demonstrado o funcionamento de alguns dos protocolos da arquitetura TCP/IP, especificamente um protocolo de cada uma das quatro camadas, e que possa ser utilizado como ferramenta no estudo dos princípios gerais de transmissão de dados entre computadores;

O objetivo específico é desenvolver interfaces didáticas e propícias para a simulação de alguns procedimentos dos protocolos FTP, TCP, IP, IEEE 802.3, das camadas de aplicação, transporte, rede e enlace respectivamente, demonstrando o quanto um tutorial utilizado como ferramenta de ensino pode auxiliar o aprendizado.

## **1.2 ESTRUTURA DO TRABALHO**

O trabalho está organizado da seguinte forma:

O capítulo 1, inicia com uma breve introdução sobre os assuntos que vão ser vistos no decorrer deste trabalho, seguido de seus objetivos.

No capítulo 2, o presente trabalho aborda conceitos sobre a interatividade no processo educativo e o uso de computadores no processo de ensino-aprendizagem.

O capítulo 3, trata do assunto Redes de Computadores, abordando os seus conceitos básicos, o modelo de referência OSI e a arquitetura TCP/IP. Na arquitetura TCP/IP descrevemos sucintamente as camadas e os protocolos escolhidos para o referido trabalho.

O capítulo 4 apresenta as técnicas e ferramentas utilizadas para a implementação do tutorial. Além disso, faz uma breve descrição dos ambientes de desenvolvimento e as metodologias utilizadas na especificação do protótipo.

O capítulo 5 traz a especificação do tutorial implementado e os tópicos a ele relacionado.

No capítulo 6 serão apresentados as conclusões do trabalho e sugestões para trabalhos futuros.

## **2 FUNDAMENTAÇÃO TEÓRICA**

Está claro que se vive em uma sociedade em que a informação é o ingrediente básico e as tecnologias que nos ajudam a lidar com a informação são essenciais. Não só no mercado de trabalho, mas em ambientes acadêmicos, o desenvolvimento de estudos e pesquisas pode ser mais eficaz com o emprego destas tecnologias como forma de filtrar e classificar as informações e como ferramenta de interação da pesquisa (Gadotti, 2000).

### **2.1 A INTERATIVIDADE NO PROCESSO ENSINO-APRENDIZAGEM**

Existem linhas de pesquisadores que condenam o uso de computadores no processo de aprendizagem, com a alegação que os mesmos limitam a capacidade criativa do aluno. No entanto, observa-se a variedade de ferramentas que o computador disponibiliza para os mais diversos tipos de trabalhos, a interação que alguns softwares permitem e a possibilidade de aplicar o computador para fins bem específicos, pode-se observar que deixar de utilizá-lo poderia sim ser uma forma de limitar o aluno, não permitindo que ele usufrua de toda tecnologia disponível (Fischer, 2000) .

Outro ponto importante é a utilização da imagem, do movimento, do som, da música, do jogo. A sala de aula sem recursos audiovisuais se torna cada vez mais distante dos ambientes de animação computadorizados. O computador quebrou preconceitos e abriu espaços, que podem ser ocupados pelo vídeo, pelo jogo corporal, pelas bandas musicais, pelas oficinas de artes plásticas, por passeios no campo, pelo teatro, etc. (Fischer, 2000).

O computador trabalha com representações virtuais de forma coerente e flexível, possibilitando, assim, a descoberta e a criação de novas relações. Na utilização do computador como auxiliar na tarefa ensino-aprendizagem existe pontos importantes facilitados pelo computador, segundo Fischer (2000) destaca a seguir:

- a) dispõe suas informações de forma clara, objetiva e lógica, facilitando a autonomia do usuário;
- b) exige que o usuário tenha consciência do que quer, se organize e informe de modo ordenado o que deseja fazer;
- c) dá um retorno extremamente rápido e objetivo do processo em construção, favorecendo a auto-correção;
- d) trabalha com uma disposição espacial das informações, que pode ser controlada continuamente pelo usuário através de seu campo perceptivo visual, apoiando o raciocínio lógico;
- e) trabalha com imagens e textos de forma combinada, ativando os dois hemisférios cerebrais;
- f) através de recursos de multimídia, pode combinar imagens pictóricas ou gráficas, em uma infinidade de cores e formas, com sons verbais e/ou musicais, com movimentos, criando uma verdadeira trama de combinações possíveis, integrando a percepção, em suas múltiplas formas, ao raciocínio e à imaginação, de forma fluente, pessoal e cheia de vida.

Portanto, a inclusão de recursos didáticos multimídia no ambiente de ensino é uma importante alternativa com abordagem intermediária entre os livros e o prática podendo constituir em um forte instrumento facilitador de superação de dificuldades didático pedagógicas, seja pela motivação na utilização de inovações tecnológicas, seja pelo software apresentar situações muitas vezes difíceis de serem explicadas e demonstradas em sala de aula utilizando os recursos tradicionais de ensino.

## **2.2 UTILIZAÇÃO DO COMPUTADOR NO PROCESSO ENSINO-APRENDIZAGEM**

A comunicação é uma das maiores necessidades da sociedade humana. Conforme os povos foram se espalhando em áreas geograficamente dispersas, as dificuldades de comunicação foram tornando-se desafios.

Durante séculos, os viajantes analisaram o atlas para terem uma idéia das terras que queriam visitar, que rotas tomar e que curiosidades as diferentes regiões poderiam conter.

Vive-se no meio de um “oceano” de mudanças que os primeiros usuários de rádio e de telefone devem ter vivido.

O computador durante muito tempo foi percebido somente como uma máquina, perante o grande público. Atualmente, é de conhecimento geral que o universo da informática comporta dimensões múltiplas: os investimentos industriais, o complexo campo da programação e das linguagens, a multiplicidade de diferentes usos.

Aos poucos o acesso as tecnologias tornou o computador um elemento da nossa cultura. O computador tem feito uma trajetória na vida brasileira semelhante à da televisão, ou seja, muitas famílias optam por ter um microcomputador em casa, em lugar de outros bens.

Atualmente não questiona-se mais a introdução do computador em sala de aula, mas sim de que forma utilizá-lo e como usar suas potencialidades no desenvolvimento e na construção do conhecimento (Martins, 2000).

Com a revolução tecnológica e o desenvolvimento da informática, a exigência por ambientes de aprendizagem está cada vez maior, pois está ligada aos novos cenários que a sociedade apresenta (Zandomenghi, 2000).

Pode-se dizer que com a invenção do telégrafo por Samuel F. B. Morse em 1838 a comunicação atravessou uma grande evolução, dando origem a maior parte dos grandes sistemas de comunicação que temos hoje em dia.

As redes de computadores não emergiram de uma hora para outra como uma tecnologia única e independente. De fato, as redes modernas têm raízes nos primeiros sistemas de telefones e telégrafos.

A fusão dos computadores e das comunicações teve uma profunda influência na forma como os sistemas computacionais eram organizados. O modelo de um computador atendendo todas as necessidades computacionais de uma organização foi substituído pelas redes de computadores, onde os trabalhos são realizados por uma série de computadores interconectados. Não há dúvidas das vantagens que as redes trazem aos usuários, principalmente no que diz respeito ao compartilhamento de recursos e possibilidade de



comunicação de forma rápida e eficaz. Mas, para técnicos, desenvolvedores e cientistas da computação, o assunto redes de computadores é vasto e complexo, com diferentes tipos de usuários que precisam ser atendidos em suas necessidades. Por isso, as redes são, antes de mais nada, tecnologias que precisam ser compreendidas e estudadas, para que se possa usufruir de todas as suas vantagens.

Com o uso de redes locais em escritórios (automação), indústrias (controles de processos), escolas (ensino e administração), tem-se constatado a carência de recursos humanos nesta tecnologia.

A partir de 1990 as redes de computadores começaram a oferecer serviços para pessoas físicas em suas respectivas casas. Através da Internet muitas pessoas pagam suas despesas, administram contas bancárias e gerenciam investimentos eletronicamente. Para tanto, cursos, palestras e artigos técnicos sobre o tema têm acontecido com maior frequência, além da oferta de disciplinas afins em currículos de engenharia e computação. Os recursos didáticos utilizados nestes cursos e palestras são recursos tradicionais de ensino: giz, transparência, “*slides*”.

Esta última década vêm se caracterizando por grandes mudanças na sociedade como um todo, destacando a alteração das relações dos meios para o ensino e treinamento e a completa transformação na maneira de se obter informações. Todas essas mudanças têm como uma das causas a exponencial evolução da informática e em função disto um número cada vez maior de pessoas tem tido acesso a computadores.

O crescimento da informática gerou uma evolução na procura por softwares e serviços onde os aspectos marcantes são o uso intensivo de multimídia e posteriormente da hipermídia. Entende-se por multimídia softwares que utilizam, em conjunto, diversas mídias, como som, imagem, texto, onde há uma característica marcante, a interatividade. Já a hipermídia é a revolução da mídia, onde o usuário passa a ser um participante ativo do ambiente criado pelo software (Bizzotto, 2000).

Juntos, recursos didáticos, multimídia e a hipermídia podem atuar de forma eficiente no desenvolvimento de aplicações que auxiliem o ensino de diversas tecnologias, que pelo seu grau de complexidade tornaria-se difícil o aprendizado e entendimento.

Não pode-se esquecer que, tratando-se de uma ferramenta de ensino, a necessidade de se trabalhar em equipes interdisciplinares é de suma importância para saber o que fazer, o que desenvolver na ferramenta mais adequada.

Quanto aos programas de autoria mais usados nas produções multimídia profissionais no Brasil podemos citar (VALENTE, 1999):

- a) o Assymetix Toolbook, indicado para produções de hipermídia convencionais, ou seja, aquelas em que a navegação se faz por seleção de palavras-chaves ou ícones que se abrem em informações mais específicas ou aprofundadas.
- b) o Macromedia Director, indicado para produções em que a velocidade de exibição da sequência seja mais alta, favorecendo a criação de animações

Segundo Bizzotto (2000), a vantagem de utilizar-se estas ferramentas é que a pessoa que irá desenvolver o software não precisa ser um especialista em computação para desenvolver produtos de qualidade: é o caso de um professor que deseja desenvolver uma hipermídia para auxiliá-lo em sua disciplina.

Esse trabalho mostra como é útil o uso de ferramentas multimídia para auxiliar o ensino de situações complexas de serem apresentadas em sala de aula utilizando-se somente recursos como giz, quadro, transparências, etc, através da implementação de um ambiente de ensino para redes de computadores baseadas em TCP/IP, especialmente algumas funcionalidades de suas quatro camadas.

### 3 REDES DE COMPUTADORES

A seguir serão apresentados alguns conceitos básicos sobre redes de computadores como modelo de referência OSI e a Arquitetura TCP/IP. Para a Arquitetura TCP/IP e suas camadas visa-se dar maior ênfase no desenvolver do trabalho.

Uma rede de computadores é formada por um conjunto de dispositivos capazes de se comunicar através do sistema de comunicação por troca de mensagens e compartilhar recursos, interligados por um sistema de comunicação.

O sistema de comunicação constitui-se de um arranjo topológico interligando os vários módulos processadores através de enlaces físicos (*meios de transmissão*) e de um conjunto de regras com o fim de organizar a comunicação (*protocolos*).

Existem alguns tipos de redes de computadores, as quais estão descritas abaixo:

- a) Redes Locais (Local Area Networks – LANs) caracterizam-se como sendo redes que permitem a interconexão de equipamentos de comunicação de dados numa pequena região. Costuma-se considerar “pequena região” distâncias entre 100 m e 25 Km. Algumas características encontradas são as altas taxas de transmissão e baixas taxas de erros, porém estes termos estão ligados à tecnologia atual. Geralmente são de propriedade privada.
- b) Redes Metropolitanas (Metropolitan Area Networks – MANs): este termo surge com o aparecimento do padrão IEEE 802.6, ou DQDB (Distributed Queue Dual Bus), que nada mais é do que um padrão especial. Apresenta características semelhantes às das redes locais, sendo que as MANs, em geral, cobrem distâncias maiores do que as LANs operando em velocidades maiores.
- c) Redes Geograficamente Distribuídas (Wide Area Networks – WANs): surgiram da necessidade de se compartilhar recursos especializados por uma maior comunidade de usuários geograficamente dispersos. Tais redes são em geral públicas.(SOARES, 1995)

Protocolos e serviços são conceitos que confundem, porém a importância dos mesmos é muito grande. Serviço é um conjunto de primitivas (operações que uma entidade servidora

oferece a entidade cliente acima dela. O serviço define as operações que a camada que está preparada para executar e satisfazer a seus usuários.

O protocolo é um conjunto de regras que controla o formato e o significado dos quadros, pacotes ou mensagens trocadas pelas entidades pares. O protocolo diz respeito a implementação do serviço, e portanto, não é visto pelo usuário de serviço.

Os protocolos possibilitam a comunicação através de uma rede, onde os dados são transferidos entre as máquinas em unidades chamadas de pacotes, que contém informações de controle (cabeçalho) e dados. São agrupados em famílias organizadas em camadas, que por sua vez formam uma pilha. Cada camada tem uma função específica na pilha de protocolos (Tanenbaum, 1997).

### **3.1 MODELOS DE REFERÊNCIA**

O modelo de referência serve como uma diretriz útil para a divisão das tarefas de comunicação. Antes cada fabricante tinha seu próprio protocolo que muitas vezes não era compatível, impossibilitando a interoperabilidade.

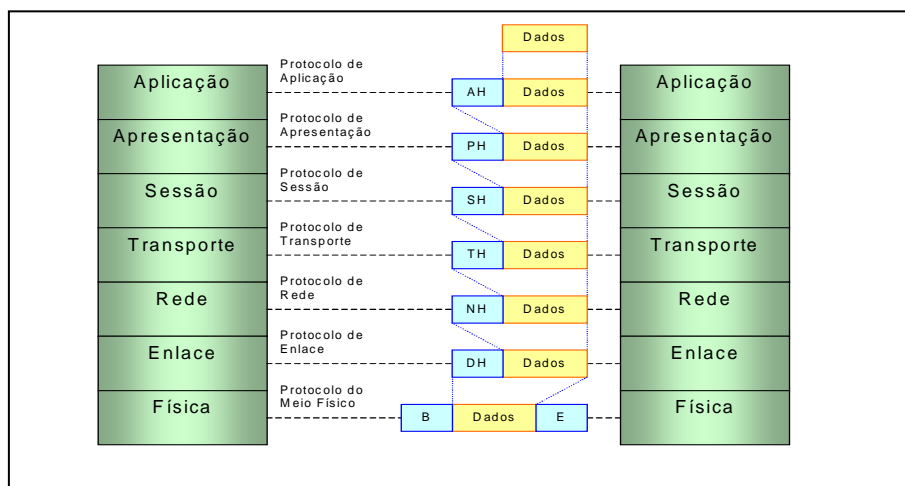
Para descrever a arquitetura do conjunto de protocolos da Arquitetura TCP/IP geralmente é utilizado um modelo conceitual de camadas.

Existem dois modelos: o modelo OSI de sete camadas da ISO (*International Standards Organization*) e o modelo DARPA de quatro camadas.

#### **3.1.1 MODELO DE REFERÊNCIA OSI**

O modelo de referência OSI é baseado em uma proposta desenvolvida pelo ISO (International Standards Organization) organismo sediado em Genebra, como um primeiro passo em direção da padronização internacional dos protocolos usados nas diversas camadas (Tanenbaum, 1997). Este modelo se baseia em um conjunto de sete camadas, conforme figura 1:

FIGURA 1 AS SETE CAMADAS DO MODELO DE REFERÊNCIA OSI



Fonte: Tanenbaum (1997)

- a) **Camada Física:** trata da transmissão de bits através de um canal de comunicação. Questões de projeto devem levar em conta interfaces mecânicas, elétricas, procedurais e o meio físico;
- b) **Camada de Enlace:** transforma a transmissão bruta em uma linha de transmissão de quadros que pareça livre de erros. Cabe à camada de enlace reconhecer os limites de quadros através da inclusão de padrões de bits no seu início e no seu fim. Também deve resolver problemas de quadros repetidos, perdidos ou danificados, assim como controlar o seu fluxo;
- c) **Camada de Rede:** especifica o modo com que os pacotes são roteados da origem ao destino. As rotas podem ser estáticas ou dinâmicas. Esta camada é responsável pela interconexão de redes heterogêneas;
- d) **Camada de Transporte:** a função principal é tratar da divisão dos dados em pacotes de tamanhos compatíveis com a camada de rede a ser utilizada e reagrupá-los na outra extremidade. É também responsável pelo estabelecimento das conexões e pelo controle de fluxo. Ao estabelecer uma conexão, deve-se definir o tipo de serviço requerido: ponto-a-ponto livre de erros, mensagens isoladas sem garantia de entrega, difusão de mensagens. A camada de transporte é a verdadeira camada fim a fim que liga a origem ao destino por intermédio da resolução dos seus endereços e nomes;
- e) **Camada de Sessão:** permite que usuários de diferentes máquinas estabeleçam sessões entre elas. Além disso é responsável pela sincronização do fluxo de dados;

- f) **Camada de Apresentação:** se preocupa com a sintaxe e a semântica das informações transmitidas, tais como conversão de dados e criptografia;
- g) **Camada de Aplicação:** contém aplicações padrão específicas para o protocolo: transferência de arquivos, gerência de rede, pesquisa de diretório, etc;

### 3.1.2 ARQUITETURA TCP/IP

O desenvolvimento do protocolo TCP/IP começou em 1969, com o projeto ARPANET, da Agência de Projetos de Pesquisas Avançadas do Departamento de Defesa dos EUA (DARPA – *Department of Defense Advanced Research Projects Agency*), com o objetivo de desenvolver uma rede que interligasse os computadores do governo americano, de diferentes fabricantes e utilizando diferentes sistemas operacionais. Essa rede deveria ser descentralizada e mesmo que um dos computadores dessa rede fosse destruído num eventual ataque militar, os demais continuariam a funcionar normalmente, graças a um mecanismo de rotas alternativas.

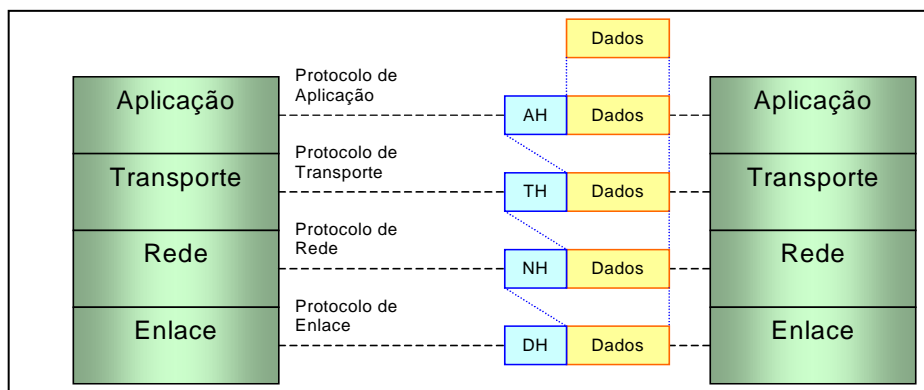
Algum tempo depois desse início com finalidade militar, a *National Science Foundation* criou uma rede semelhante para interconectar instituições de pesquisa e universidade, utilizando os mesmos protocolos da rede ARPANET.

Desses projetos surgiu o protocolo TCP/IP que serviu como alicerce para a construção da rede que hoje se conhece como *Internet*.

A partir de 1993 a *Internet* ficou disponível para uso comercial e se popularizou de tal forma que hoje a maioria das pessoas a utiliza com familiaridade (Palma, 1999).

Segundo Kurose (2001), a arquitetura Internet TCP/IP é organizada em quatro camadas conceituais, conforme figura 2. TCP/IP é um acrônimo para o termo *Transmission Control Protocol/Internet Protocol* ou seja, é um conjunto de protocolos, onde dois dos mais importantes (o TCP e o IP) deram seus nomes a arquitetura.

FIGURA 2 AS QUATRO CAMADAS DA ARQUITETURA TCP/IP



Fonte: Palma (1999)

A Camada de Enlace é responsável pelo envio de datagramas construídos pela camada de rede. Realiza também o mapeamento entre um endereço de identificação de nível rede para um endereço físico ou lógico do nível de enlace. Alguns protocolos existentes nesta camada estão citados abaixo, segundo Comer (1998):

1. protocolos com estruturas de rede própria ( X25, Frame-Relay, ATM);
2. protocolo de Nível Físico (X.21);
3. protocolos de barramento de alta velocidade (SCSI, HIPPI);
4. protocolos de Mapeamento de Endereços (ARP, podendo também ser considerado como parte da camada de Rede).

Os protocolos deste nível possuem um esquema de identificação das máquinas intelígadas por este protocolo. Por exemplo, cada máquina situada em uma rede possui um identificador único chamado endereço MAC (Media Access Control) ou endereço físico que permite distinguir uma máquina de outra, possibilitando o envio de mensagens específicas para cada uma delas. Tais redes são chamadas redes locais de computadores.

A Camada de Rede integra toda a arquitetura. Esta camada realiza a comunicação entre máquinas vizinhas através do protocolo IP. Para identificar cada máquina e a própria rede onde estas estão situadas, é definido um identificador, chamado endereço IP. Os protocolos existentes nesta camada são definidos por Tanenbaum (1997) da seguinte forma:

1. protocolo de transporte de dados: IP – *Internet Protocol*;
2. protocolo de controle e erro: ICMP – *Internet Control Message Protocol*;

3. protocolo de controle de grupos de endereços: IGMP – *Internet Group Management Protocol*;

4. protocolos de controle de informações de roteamento.

Define um formato de pacote e um protocolo chamado IP (*Internet Protocol*). É responsável pelo roteamento e pelo controle de congestionamentos.

A Camada de Transporte permite que as entidades par da origem e do destino mantenham um canal de comunicação. Dois protocolos fim a fim foram definidos: TCP (*Transmission Control Protocol*), orientado a conexão e confiável; e UDP (*User Datagram Protocol*), sem conexão e não confiável.

O protocolo UDP realiza a multiplexação para que várias aplicações possam acessar o sistema de comunicação de forma coerente.

O protocolo TCP realiza, além da multiplexação, uma série de funções para tornar a comunicação entre origem e destino confiável. São responsabilidades do protocolo TCP: o controle de fluxo, o controle de erro, o sequenciamento e a multiplexação de mensagens.

A Camada de Aplicação reúne os protocolos que fornecem serviços de comunicação ao sistema ou ao usuário. Segundo Comer (1998), pode-se separar os protocolos de aplicação em protocolos de serviços básicos ou protocolos de serviços para o usuário:

1. protocolos de serviços básicos, que fornecem serviços para atender as próprias necessidades do sistema de comunicação TCP/IP: DNS, BOOTP, DHCP.
2. protocolos de serviços para o usuário: Telnet, FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), SNMP (*Simple Network management Protocol*), NNTP (*Network News Transfer Protocol*), HTTP (*Hypertext Transport Protocol*), RNP (*Real Time Protocol*). Estes protocolos dão suporte às aplicações padrão comuns.

### **3.1.2.1 CAMADA DE ENLACE**

Um dos protocolos de maior interesse nesta camada é o CSMA (Carrier Sense Multiple Access), mais especificamente o CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*).



O IEEE (*Institute of Electrical and Electronics Engineers*) desenvolve padrões para as áreas de engenharia elétrica e informática. Um dos padrões é o 802 para redes locais, sendo o mais importante adotado pelas LANs o padrão 802.3 - CSMA/CD (*Carrier Sense Multiple Access*), em cuja especificação se baseia o Ethernet.

Esta técnica sincroniza os quadros de colisão tentando ao máximo evitar a colisão e algumas de suas variantes, detectar quadros colididos em tempo de transmissão abortando-a, fazendo com que os quadros colidam durante o menor tempo possível, aumentando assim a eficiência na utilização da capacidade do canal (Tanenbaum, 1997).

Os protocolos nos quais as estações escutam uma portadora (transmissão) e funcionam de acordo com elas são denominados protocolos com detecção de portadora. Duas estratégias foram desenvolvidas para os protocolos com detecção de portadora:

1. na estratégia CSMA persistente, ao efetuar uma transmissão, a estação continua a escutar o meio até que ele fique livre. Transmitindo com uma probabilidade  $p$  ou espera por um intervalo de tempo fixo e então transmite seguindo assim até transmitir ou até que outra estação ganhe acesso ao canal, quando o procedimento de transmissão recomeça;
2. na estratégia CSMA não-persistente, se, ao escutar o meio de transmissão, um nó sentir que está havendo uma transmissão ele fica esperando por um intervalo de tempo aleatório antes de tentar novamente o acesso.

Nas duas estratégias acima há prós e contras. O algoritmo não-persistente é eficiente para evitar colisões, pois quando duas estações quiserem transmitir esperarão por intervalos diferentes (aleatórios). Porém pode haver um tempo perdido. No algoritmo persistente não há tempo perdido, porém quando mais de uma estação esperar para transmitir haverá colisão (Soares, 1995).

Para a detecção de colisão ser realizada durante a transmissão surge o método CSMA/CD. Inicialmente havia o sistema ALOHA no Hawaii. Depois o conceito de detecção ou percepção da portadora (*carrier sensing*) foi incluído e a Xerox construiu um sistema denominado CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) para

conectar cerca de 100 estações em um cabo de 1 km. O sistema foi chamado de "*ether after the luminiferous ether*" (relacionado ao fato da crença que havia quanto a forma pela qual as radiações eletromagnéticas se propagavam) sendo depois conhecido como Ethernet. Atualmente, o termo *ether* refere-se a qualquer meio através do qual se deseja transmitir dados (Tanenbaum, 1997).

A Ethernet da Xerox obteve tanto sucesso que a Xerox, DEC e Intel projetaram um padrão para 10 Mbps Ethernet. Este padrão formou a base para o padrão IEEE 802.3. As implementações do 802.3, incluindo a Ethernet, usam a codificação Manchester. A presença de uma transição no meio de cada bit torna possível a sincronização entre transmissor e receptor. A qualquer instante de tempo a Ethernet deve estar em um dos três estados: transmitindo um bit 0, transmitindo um bit 1 ou ociosa (0 volts). Um sinal alto é +0,85V e uma transição para baixo é -0,85V. A diferença de voltagem, é claro deveria ser 1,7V.

Dessa forma, um sistema empregando CSMA/CD deveria encontrar-se em um desses estados: transmissão, contenção ou ociosidade. Um pequeno montante de tempo ocioso é necessário antes e após cada estado de transmissão e contenção para que a percepção da portadora possa ocorrer. Entretanto, em uma rede congestionada pode acontecer que várias interfaces possa querer tentar transmitir quando percebem que a rede está ociosa.

O CSMA/CD, ao contrário das outras técnicas das outras relacionadas ao CSMA puro, suspende a transmissão no momento em que detecta uma colisão. A estação aguarda então por um determinado tempo e tenta transmitir novamente.

Eficiência (E), segundo Soares (1995), depende do tamanho do quadro (M), do tempo de propagação ( $t_p$ ) e da taxa de transmissão (C) na seguinte relações que seguem.

#### **Relação entre o tamanho dos pacotes a taxa de transmissão e o tempo de propagação**

$$M \geq 2 C t_p$$

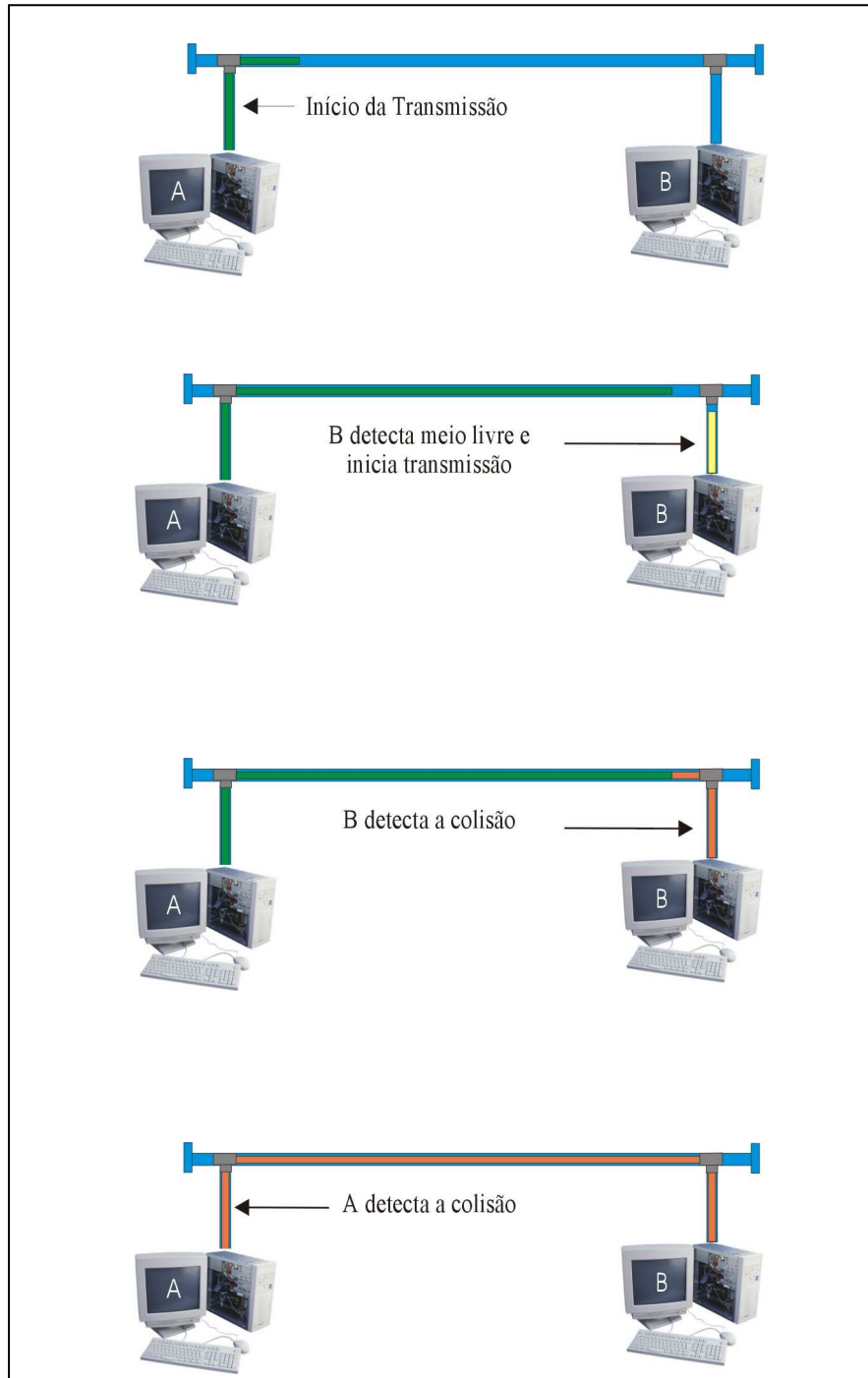
**A Eficiência dessa técnica é dada em função do tamanho dos pacotes, da taxa de transmissão e do tempo de propagação**

$$E = \frac{1}{1 + (3,4 * t_p * \frac{C}{M})}$$

Em redes reais, entretanto, as coisas não são tão perfeitas como em estudos estatísticos. Os quadros variam em tamanho e o tráfego tendem a vir em rajadas sendo proporcionalmente e uniformemente propagados. A contenção vem somente no início do quadro, então a mais simples medição da rede considera o quanto a sua relação com as colisões é a taxa de pacotes. A Ethernet tem um limite inferior no tamanho de um quadro o qual limita a taxa máxima teórica de pacotes a cerca de 14.000 pacotes por segundo, mas com pacotes mais próximos do tamanho máximo do quadro poder-se-ia esperar umas poucas centenas de pacotes por segundo em uma rede congestionada e cerca de mil pacotes não tão congestionada.

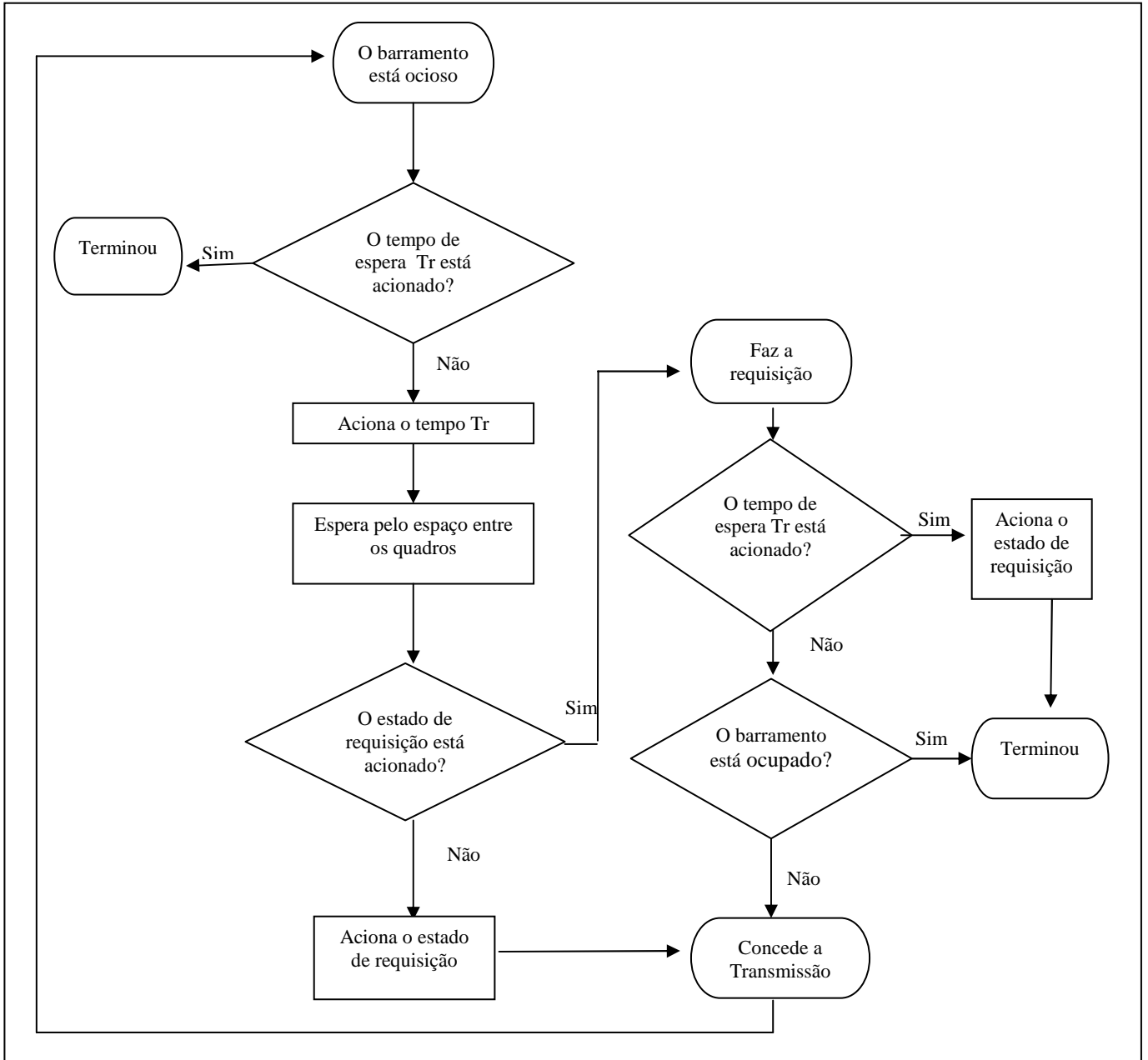
Para que possa haver detecção de colisão conforme a figura 3, é necessário que sejam observadas as relações entre M, C e  $t_p$  vistas anteriormente.

FIGURA 3 COLISÃO EM REDES EM BANDA BASE



Fonte: Tanenbaum (1997)

FIGURA 4 FLUXOGRAMA DE FUNCIONAMENTO DO CSMA/CD



A figura 04, apresentada acima demonstra o fluxograma de funcionamento do CSMA/CD, sobre o qual teremos algumas observações a serem feitas:

- a) é eficiente para redes de pequenas distâncias (até 2 km) e pouco tráfego podendo atingir 98% de utilização do meio;
- b) não há garantia quanto à limitação do retardo de transferência;

- c) a adoção dessa estratégia pode ser um pouco mais onerosa em função do custo de interfaces mais caras;
- d) o CSMA/CD com espera aleatória tornou-se um padrão internacional (ISO 8802-3/IEEE 802.3)
- e) não exige reconhecimento de mensagens para retransmissão. Entretanto algumas implementações fazem a confirmação nesse nível a fim de obter maior eficiência.

### 3.1.2.2 CAMADA DE REDE

Na camada de rede da arquitetura TCP/IP, o protocolo IP realiza a função mais importante que é a própria comunicação inter-redes. Para isso ele realiza a função de roteamento que consiste no transporte de mensagens entre redes e na decisão de qual rota uma mensagem deve seguir através da estrutura de rede para chegar ao destino.

Seu objetivo é que pacotes emitidos sejam entregues ao seu destino independentemente da(s) tecnologia(s) de transmissão utilizada(s). Esta camada define um formato de pacote e um protocolo chamado IP (*Internet Protocol*). É responsável pelo roteamento e pelo controle de congestionamentos (Tanenbaum, 1997).

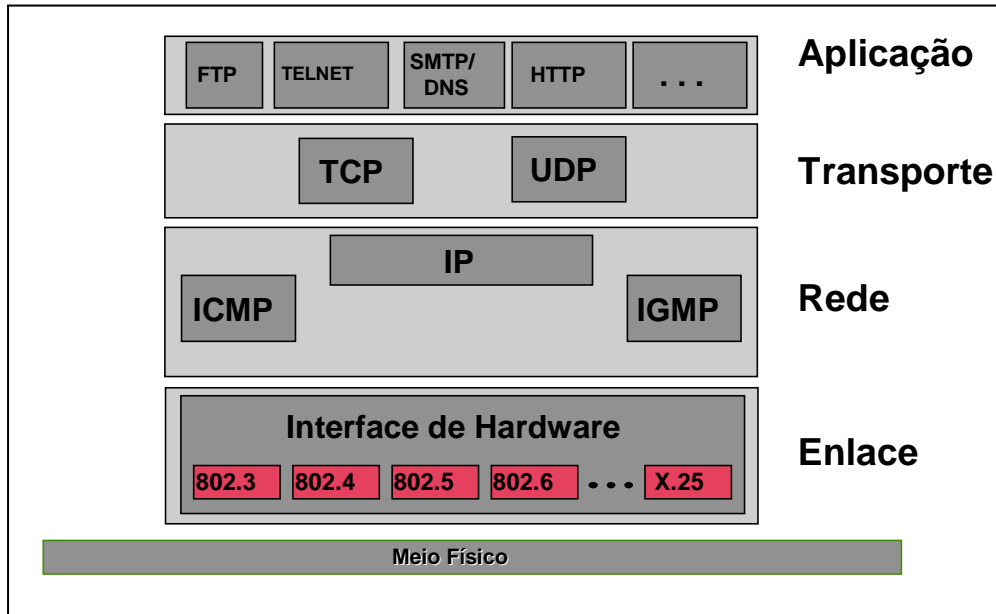
Na arquitetura TCP/IP, os elementos responsáveis por interligar duas ou mais redes distintas são chamados de **roteadores**. As redes interligadas podem ser tanto redes locais, redes geograficamente distribuídas, redes de longa distância com chaveamento de pacotes ou ligações ponto-a-ponto seriais. Um roteador tem como característica principal a existência de mais de uma interface de rede, cada uma com seu próprio endereço específico. Um roteador pode ser um equipamento específico ou um computador de uso geral com mais de uma interface de rede.

Por outro lado, um componente da arquitetura TCP/IP que é apenas a origem ou destino de um datagrama IP (não realiza a função de roteamento) é chamado de **host**.

Os protocolos da camada de rede recebem datagramas da camada de transporte e analisam-nos para definir a rota que será utilizada. Na camada de rede residem quatro protocolos: **IP**, que define mecanismos de endereçamento e roteamento de pacotes na rede, **ICMP**, que fornece informações sobre as condições de transmissão de datagramas na rede ou

sobre erros e **IGMP**, utilizado para especificar quais computadores pertencem a um grupo *multicast* (Palma, 1999).

FIGURA 5 PROTOCOLOS DA CAMADA DE REDE



Fonte: Palma (1999)

O IP define mecanismo de endereçamento e roteamento dos datagramas em redes TCP/IP. O IP foi projetado para permitir a interconexão de redes de computadores que utiliza a tecnologia de comutação de pacotes. O ambiente inter-redes consiste em *hosts* conectados a redes, que por sua vez são interligadas através de *gateways*, que na Internet também são chamados roteadores.

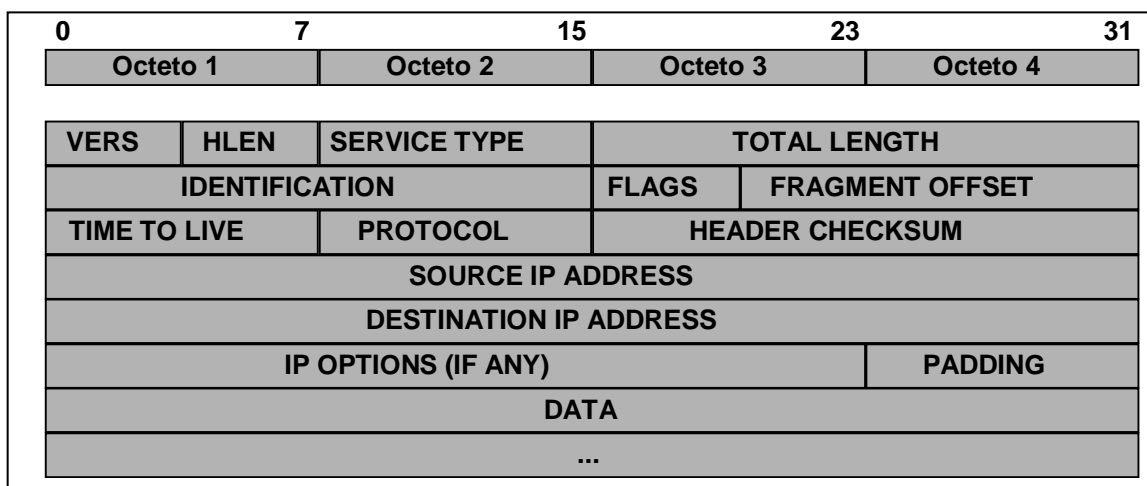
O protocolo IP não é orientado a conexão. Ele apenas recebe os datagramas da camada de transporte, analisa-os para definir a rota a ser utilizada e os envia para a camada inferior (interface de rede), deixando a cargo da camada de transporte a garantia de entrega dos dados. O IP não confirma a recepção dos dados nem retransmite os dados em caso de erro, é um protocolo não confiável, esse controle é deixado pelo protocolo TCP na camada de transporte (Palma, 1999).

Os endereços IP (versão 4) são números com 32 bits, normalmente escritos com quatro octetos (em decimal). A primeira parte do endereço identifica uma rede específica na inter-rede, a segunda parte identifica um *host* dentro da rede. Um endereço IP não identifica uma máquina individual, mas uma conexão à inter-rede (Soares, 1995).

Um endereço IP pode ser usado para referenciar uma rede ou um *host* individual. Um endereço de rede tem o campo identificador de *host* com todos os bits iguais a 0. se todos os bits do campo identificador de *host* forem iguais a 1, por convenção, o endereço está se referenciando a todos os *hosts* de uma rede (Soares, 1995).

A figura 6 mostra o formato de um datagrama IP:

FIGURA 6 FORMATO DO DATAGRAMA IP



Fonte: Soares (1995)

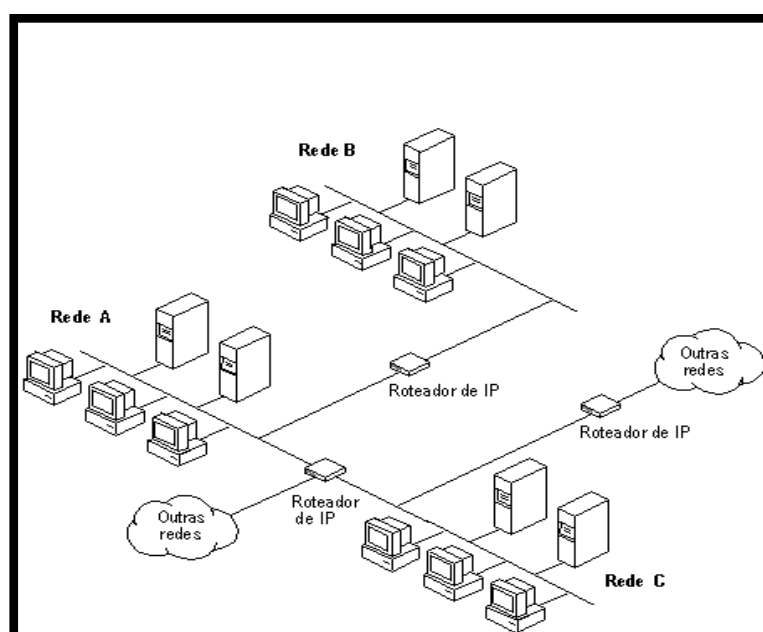
O campo VERS traz a versão do protocolo que montou o quadro, o campo HLEN mantém o comprimento do cabeçalho do datagrama IP, em palavras de 32 bits. SERVICE TYPE armazena parâmetros que indicam a qualidade do serviço que deve ser prestado pelas redes onde passa o datagrama. TOTAL LENGTH é o tamanho do datagrama (cabeçalho + dados). IDENTIFICATION, FLAGS e FRAGMENT OFFSET são usados no procedimento de fragmentação e remontagem de datagramas. TIME TO LIVE limita o tempo de transmissão dos datagramas, recebendo um valor inicial quando o datagrama é criado e, sempre que um gateway retransmite ele decrementa o valor do campo. Quando o valor Time to Live atinge o valor zero, o datagrama é descartado, pois indica que o datagrama estava em *loop*. O campo PROTOCOL identifica o protocolo de transporte que gerou e que irá receber o datagrama no destino. HEADER CHECKSUM é o campo usado pelos *gateways* para verificar se as informações do cabeçalho de um datagrama IP estão corretos. SOURCE IP ADDRESS e DESTINATION IP ADDRESS carregam os endereços dos *hosts* origem e destino. IP OPTIONS é o campo usado para informações de segurança, roteamento na origem, relatório



de erros, depuração, fixação de hora, etc; e o campo DATA carrega os dados do datagrama IP (Soares, 1995).

Os roteadores permitem interligar redes distintas. Um roteador é um dispositivo com diversas conexões (portas) e a cada uma destas portas pode-se conectar uma rede. Sua função é analisar o endereço destino de uma mensagem, definir a qual rede este endereço pertence e encaminhar a mensagem para esta rede, transmitindo-a para a porta correspondente (figura 7). Em redes TCP/IP, cada porta do roteador recebe um endereço IP (Palma, 1999).

FIGURA 7: EXEMPLO DE ROTEAMENTO



Fonte: Palma (1999)

O roteamento no IP baseia-se no identificador de rede do endereço de destino. Cada computador possui uma tabela, denominada tabela de roteamento IP, cujas entradas são pares: endereço de rede/endereço do roteador. Quando um módulo IP, por exemplo, executando em um roteador, tem que encaminhar um datagrama, ele inicialmente verifica se o destino é um *host* conectado à mesma rede. Se for o caso, o datagrama é entregue à interface de rede que se encarrega de mapear o endereço IP no endereço físico do *host*, encapsular o datagrama IP em um quadro da rede e transmiti-lo ao destinatário.

Se a rede identificada no endereço de destino do datagrama for diferente da rede onde está o módulo IP, ele procura em sua tabela de roteamento uma entrada com o endereço de

rede igual ao do endereço destino do datagrama, recuperando o endereço do roteador que deve ser usado para alcançar a rede que possui o *host* destinatário. O roteador recuperado pode não estar diretamente ligado à rede de destino, mas ele deve fazer parte do caminho percorrido para alcançá-la.

Quando uma inter-rede interliga um número muito grande de redes individuais (como na Internet), o esquema de roteamento apresentado traz a desvantagem do elevado tamanho das tabelas de roteamento. Para amenizar o problema, uma das estratégias é a utilização de roteadores *default*. Por exemplo, em redes que são ligadas à inter-rede por um único roteador, não é necessário ter uma entrada separada na tabela de rotas para cada uma das redes distintas da inter-rede. Simplesmente define-se o roteador como caminho *default*.

Quando a rede possui mais de um roteador, define-se o roteador *default* como caminho quando não é encontrada uma rota específica para um datagrama. Através de um protocolo específico ICMP (*Internet Control Message Protocol*) o roteador informará ao módulo IP se ele é a melhor escolha para alcançar uma determinada rede. Essa mensagem, chamada *redirect*, carrega como parâmetro o endereço do roteador que é a escolha correta. O módulo IP, ao receber uma mensagem *redirect*, adiciona uma entrada em sua tabela de roteamento, associando a rede de destino ao endereço do roteador recebido na mensagem (Soares, 1995).

### **3.1.2.3 CAMADA DE TRANSPORTE**

A camada de transporte, segundo Tanenbaum (1997), é o coração de toda hierarquia de protocolos, tendo como função promover uma transferência de dados confiável e econômica entre a máquina de origem e a máquina de destino.

A Internet tem dois protocolos principais na camada de transporte, uma operando no modo orientado a conexão (TCP – *Transmission Control Protocol*) e outro sem conexão (UDP – *User Datagram Protocol*).

A camada IP não oferece qualquer garantia de que os datagramas serão entregues de forma apropriada, portanto, cabe ao TCP administrar os temporizadores, retransmití-los sempre que necessário e reorganizá-los na seqüência correta. Resumindo, o TCP deve fornecer a confiabilidade que a maioria dos usuários quer mas que o IP não fornece.

As entidades TCP transmissoras e receptoras trocam dados na forma de segmentos. Um **segmento** consiste em um cabeçalho fixo de 20 bytes, seguido de zero ou mais bytes de dados. O tamanho do segmento é decidido pelo software TCP. Dois fatores restringem o tamanho do segmento. Primeiro, cada segmento incluindo o cabeçalho do TCP, deve caber na carga útil do IP, que é de 65.535 bytes. Segundo Palma (1999), cada rede tem uma MTU (*Maximum Transfer Unit*), e cada segmento deve caber na MTU.

Se um segmento atravessar uma seqüência de redes sem ser fragmentado e depois chegar a uma rede cuja MTU não o comporta, o roteador localizado na fronteira fragmentará o segmento em dois ou mais segmentos menores. A criação de novos segmentos pelos roteadores aumentam o overhead total pois adiciona mais 40 bytes de informação de cabeçalho.

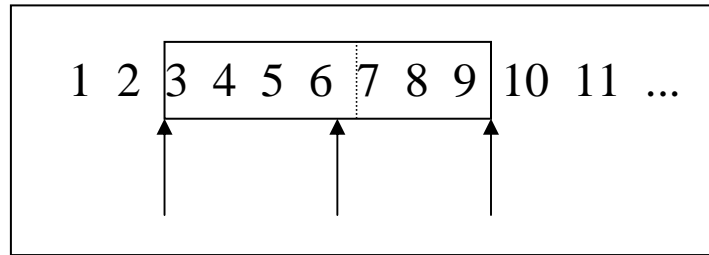
O protocolo básico utilizado pelas entidades TCP é o protocolo de janelas deslizantes. Quando envia um segmento, o transmissor também dispara um temporizador. Quando o segmento chega ao destino, a entidade TCP receptora retorna um segmento com um número de confirmação igual ao próximo número de seqüências que espera receber. Se o temporizador expirar antes da confirmação a ser recebida, o segmento será retransmitido (Tanenbaum, 1997).

Os protocolos de janelas deslizantes utilizam melhor a largura de banda da rede, porque permitem que o transmissor transmita pacotes múltiplos antes de esperar uma confirmação.

O mecanismo de janelas deslizantes TCP opera em nível de octetos, e não de pacotes ou segmentos. Os octetos do stream de dados são enumerados sequencialmente, e um transmissor mantém três ponteiros associados a cada conexão, conforme é apresentado na figura 8.

O primeiro ponteiro marca a esquerda da janela deslizante, separando os octetos que tenham sido enviados e confirmados, dos octetos ainda por enviar. Um segundo ponteiro à direita da janela deslizante define o maior octeto da seqüência que pode ser enviado antes que mais confirmações sejam recebidas. O terceiro ponteiro marca o limite interno da janela que separa os octetos já enviados daqueles que ainda não o foram.

FIGURA 8 PONTEIROS EM UMA JANELA TCP

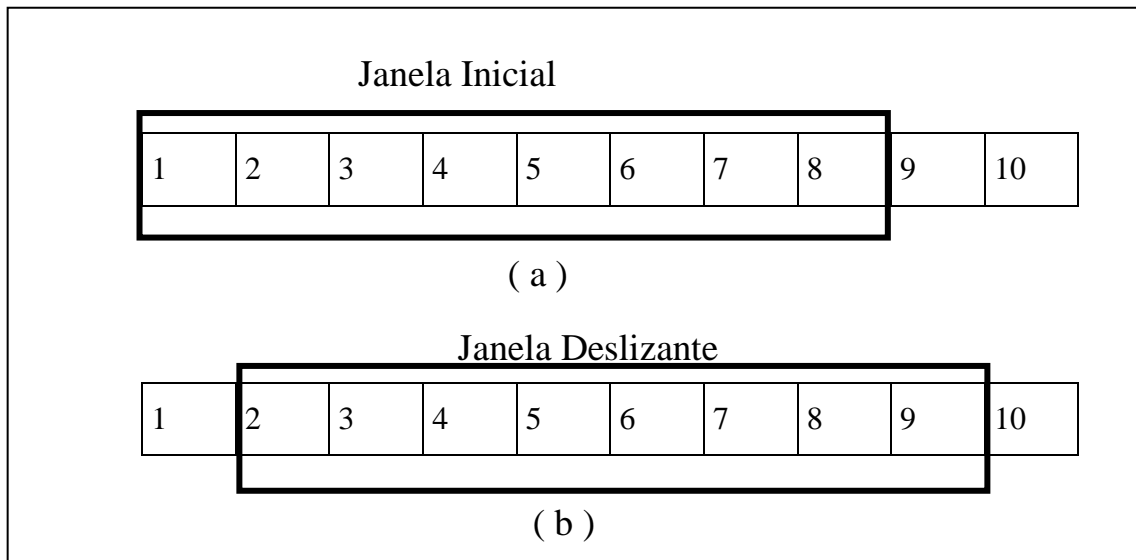


Fonte: Comer (1998)

A maneira mais simples de avaliar uma operação de janela deslizante é imaginar uma seqüência de pacotes a serem transmitidos, conforme mostra a figura 9, onde um protocolo de janela deslizante com oito pacotes na janela ( a ) e ( b ) a janela deslizando de modo que o pacote nove possa ser enviado quando for recebida uma confirmação para o pacote um, somente os pacotes que forem confirmados serão retransmitidos.

O desempenho dos protocolos de janelas deslizantes depende do tamanho da janela e da velocidade com que a rede aceita os pacotes (Comer, 1998).

FIGURA 9 MODELO DE JANELA DESLIZANTE



Fonte: Comer (1998)

Em tese, um protocolo de janela deslizante sempre se lembra de quais pacotes foram confirmados e mantém um temporizador separado para cada pacote confirmado. Se um pacote for perdido, o temporizador termina e o transmissor retransmite esse pacote. Quando o transmissor desliza sua janela, ele move para trás todos os pacotes confirmados. Na

extremidade do receptor, o protocolo mantém uma janela análoga que aceita e confirma os pacotes à medida que chegam.

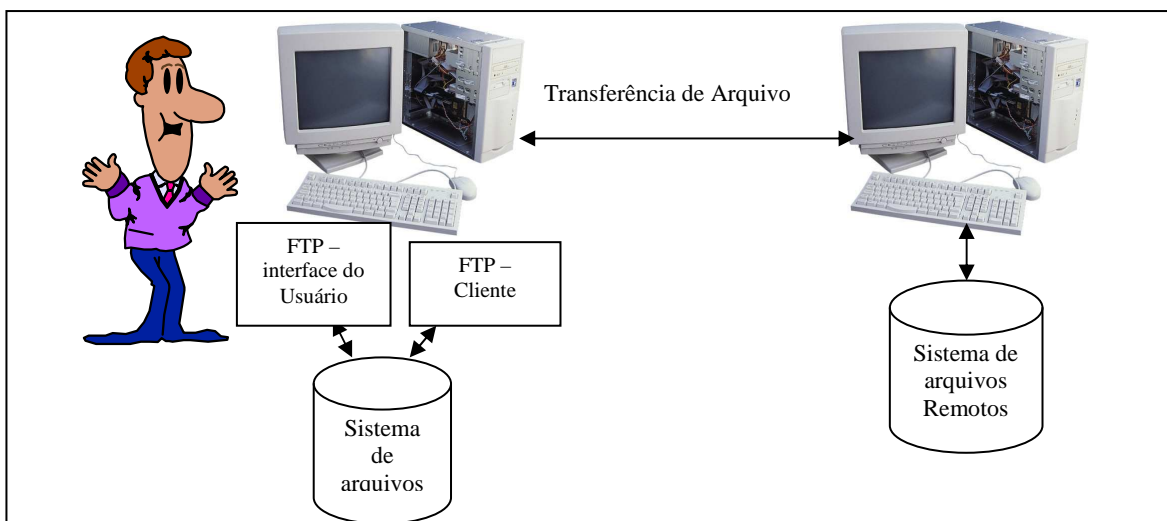
Assim, a janela particiona a seqüência de pacotes em três conjuntos: os pacotes à esquerda da janela que foram transmitidos com êxito, recebidos e confirmados; aqueles à direita ainda não transmitidos; e os pacotes que permanecem na janela e que estão sendo transmitidos.

### 3.1.2.4 CAMADA DE APLICAÇÃO

As camadas abaixo da camada de aplicação têm a função de oferecer um serviço de transporte confiável. Segundo Soares (1995), não existe um padrão que defina como deve ser estruturada uma aplicação. As aplicações trocam dados utilizando diretamente a camada de transporte através de chamadas padronizadas, como APIs.

Além de compartilhar o mesmo conjunto de primitivas de transporte, muitas aplicações Internet TCP/IP adotam o modelo cliente/servidor, quando servidor é a aplicação que ofereça um serviço a outra aplicação e cliente é a aplicação que envia de volta uma solicitação a um servidor e espera por uma resposta. A aplicação FTP é uma das principais aplicações disponíveis na arquitetura Internet TCP/IP. Ele permite que o usuário transfira ou remova arquivos ou diretórios remotos (figura 10). Este protocolo é anterior a 1971, quando a Internet ainda era uma experiência, e está descrito na RFC 959. A transferência de arquivos é um dos aplicativos TCP/IP usados com mais frequência e responde por grande parte do tráfego de rede (Comer, 1998).

FIGURA 10 – TRANSFERÊNCIA DE ARQUIVOS ENTRE SISTEMAS DE ARQUIVOS REMOTO E LOCAL



A operação de FTP baseia-se no estabelecimento de duas conexões entre o cliente (módulo FTP que está solicitando o acesso a arquivos remotos) e o servidor (módulo FTP que fornece o acesso a seus arquivos locais).

O processo de controle do cliente conecta-se ao processo de controle do servidor usando uma conexão TCP, enquanto os processos de transferência de dados relacionados usam sua própria conexão TCP. Para cada transferência de arquivos o FTP estabelece uma nova conexão de transferência de dados (Kurose, 2001).

Comer (1998) resume da seguinte forma:

“As conexões de transferência de dados e os processos de transferência de dados que as utilizam podem ser criados dinamicamente, quando necessário, mas a conexão de controle persiste através de uma sessão. Uma vez que a conexão de controle desaparece a sessão é finalizada e o software de ambas as extremidades encerra todos os processos de transferência de dados”.

## 4 TÉCNICAS E FERRAMENTAS UTILIZADAS

A seguir serão apresentados os métodos de especificação OOHDM e o ambiente de autoria Director 8.0 para implementação do tutorial proposto neste trabalho.

### 4.1 OOHDM (OBJECT ORIENTED HIPERMEDIA DESIGN METHOD)

O grau de sucesso de uma aplicação multimídia ou hipermídia está diretamente associado com a habilidade do autor em capturar e organizar a estrutura de um complexo material de forma a seqüenciá-lo de maneira clara e acessível a uma ampla audiência. Para controlar a explosão potencial do número de *elos*, uma aplicação deste tipo não deve estar totalmente conectada, ou seja, o autor deve somente interconectar diretamente apenas as partes mais importantes e significativas da informação, o essencial de um modo mais natural (Coelho, 2001).

Alguns conceitos básicos sobre os componentes que formam a metodologia de desenvolvimento de software baseado à objetos são apresentados para melhor entender a OOHDM, segundo a abordagem de Furlan (1998).

Um objeto é uma representação de algo que existe no mundo real: carro, casa, homem, etc. É uma entidade capaz de reter um estado e que fornece uma série de operações ou para alterar ou para manter esse estado. Atributos são valores e propriedades dados a objetos do mundo real. Como por exemplo, o objeto aluno possui como atributo seu nome, idade, sexo. São os valores das propriedades de um objeto que indicam seu estado atual. Muitos objetos do mundo real possuem características comuns e podem se agrupados de acordo com elas. Uma classe representa um gabarito para muitos objetos e descreve como esses objetos estão estruturados internamente. Uma instância é a ocorrência de um objeto da classe. Herança é o mecanismo para compartilhar automaticamente atributos e operações entre as classes e

objetos. Este é um poderoso recurso, não encontrado em linguagens tradicionais (Furlan, 1998).

É necessário que um modelo seja usado para a implementação de forma a maximizar a independência dos diálogos e o reuso amplo dos componentes da interface. As decisões tomadas durante um projeto de interface, devidamente documentadas, poderão ser usadas para testar a validade da implementação durante a manutenção da aplicação.

As principais vantagens de se ter um modelo para o desenvolvimento de aplicações multimídia são as seguintes:

- a) melhoria da comunicação – um modelo de projeto define uma linguagem que pode ser usada por um analista de aplicação para especificar uma dada aplicação. Assim ele facilita a comunicação entre o analista e o projetista do sistema; e entre o projetista e o implementador;
- b) desenvolvimento das metodologias de projeto e estilos de retórica – modelos de projeto definem um esqueleto no qual os autores de aplicações multimídia podem desenvolver, analisar e comparar metodologias e estilos de retórica da hiper-autoria, em um alto nível de abstração, sem se preocuparem com os detalhes contidos nas unidades de informação ou suas visualizações particulares;
- c) reusabilidade – a acessibilidade de um modelo de linguagem prepara a o terreno para o reuso da espinha dorsal da estrutura de aplicações desde que as especificações do modelo básico capturem a semântica essencial das aplicações, e permitindo portanto o reuso dessa estrutura básica quando o contexto das duas aplicações forem suficientemente similares;
- d) definição de ambientes de leitura consistentes e eficientes – as ferramentas específicas para estruturas de aplicações multimídia ajudam a evitar inconsistências estruturais e erros por parte dos autores. Porém se as aplicações forem bem estruturadas de acordo com um modelo de especificação, terão como resultado final da implementação aplicações com ambientes de navegação consistentes e eficientes, ajudando os leitores em documentos mais complexos, diminuindo o problema de desorientação;



- e) uso por ferramentas de projeto – modelos de projeto são a base para construção de ferramentas de projeto que suportam um sistemático e estruturado processo de desenvolvimento, permitindo que o projetista trabalhe em um nível de abstração próximo ao do domínio da aplicação e permitindo uma passagem sistemática para o nível de implementação (Coelho, 2001).

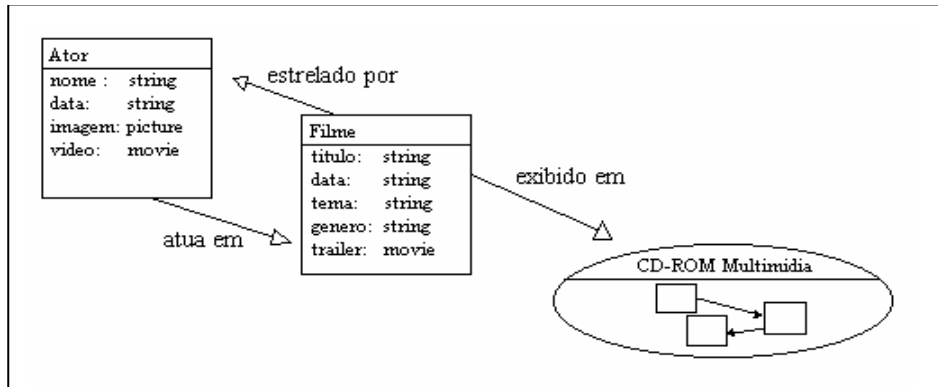
Segundo Schwabe (1999) o OOHDM é subdividido em quatro etapas distintas, sendo uma mistura de estilos de desenvolvimento com base interativa, de adição, e em protótipos. A cada passo o modelo é construído ou enriquecido e após a última etapa já se tem informação suficiente para se implementar a aplicação multimídia. As quatro fases do processo de modelagem OOHDM são: *design* conceitual; *design* da navegação; *design* da interface abstrata; implementação.

#### **4.1.1 DESIGN CONCEITUAL**

O Design Conceitual se constitui basicamente por classes, relações e subsistemas. O modelo conceitual é concebido utilizando as técnicas de modelagem já utilizadas para construção de aplicações orientadas à objetos, com a adição de outros elementos, como perspectivas de atributos. As classes conceituais podem ser montadas utilizando as hierarquias de agregação e generalização/especificação; cada classe pode se relacionar com um subsistema (abstrações de um sistema conceitual completo), desde que o mesmo possua um ou mais pontos de entrada (Schwabe, 1996). A principal preocupação neste ponto é de observar e representar a semântica do domínio da aplicação, sem muita preocupação com usuários ou tarefas.

Classes conceituais podem ser construídas usando-se de agregação e generalização. O principal objetivo desse passo é se formar um domínio semântico o mais geral possível. No final dessa etapa tem-se uma classe e o esquema instanciado, baseado em sub-sistemas, classes e relacionamentos. Cada classe é representada por um cartão classe, que é desenhado como uma caixa contendo os atributos da classe definidos em seu interior (Coelho, 2001).

FIGURA 11 MODELO CONCEITUAL COM A REPRESENTAÇÃO DE DUAS CLASSES RELACIONANDO-SE COM UM SUBSISTEMA.

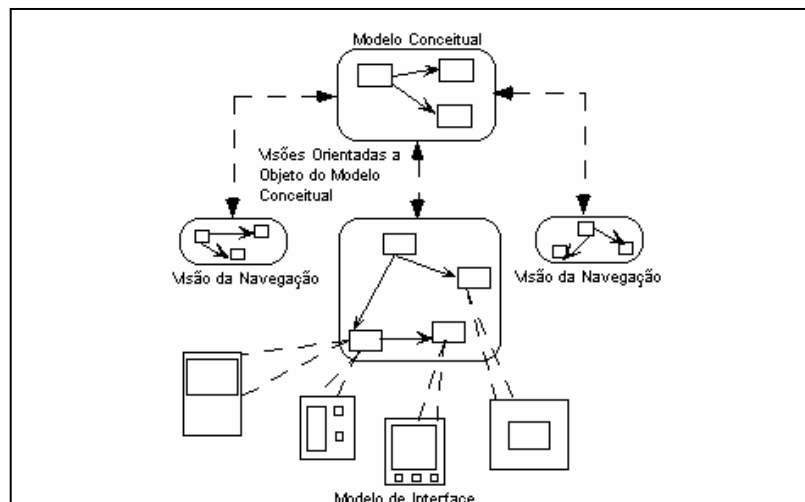


Fonte: Schwabe (1999)

## 4.1.2 DESIGN NAVEGACIONAL

Segundo Schwabe (1995) no método OOADM, a aplicação é vista como uma visão navegacional sobre o domínio conceitual. Isto demonstra que uma das principais características desta metodologia que difere das outras é a noção de navegação. Neste ponto, o desenvolvedor deve levar em conta os tipos de usuários e as tarefas que os mesmos irão realizar no uso da aplicação conforme é apresentado na figura 12:

FIGURA 12 REPRESENTAÇÃO DO PROCESSO, COM AS VISÕES DE NAVEGAÇÃO SOBRE O MODELO CONCEITUAL, E O MODELO DE INTERFACE, INTERAGINDO NESTE CONJUNTO.



Fonte: Schwabe (1999)

Para expressar diferentes visões (ou aplicações), modelos de navegação diferentes podem ser construídos sobre o mesmo esquema conceitual do domínio. A estrutura de

navegação de uma aplicação hipermídia é definida por um esquema especificando classes de navegação que refletem a visão escolhida sobre o domínio da aplicação. Existe um conjunto de tipos pré-definidos de classes de navegação: os nodos, links e estruturas de acesso, que são organizados em contextos de navegação, ou seja, um conjunto de nodos, links, classes de contexto e outros contextos de navegação.; estes tipos serão explicados com mais detalhes em seguida. A semântica dos nodos e links são comuns em aplicações hipermídia, enquanto as estrutura de acesso podem representar uma maneira alternativa de acessar nodos como índices e *tours*.

Nodos são as estruturas mais básicas utilizadas nas aplicações multimídia como armazenadores de informação. São visões orientadas à objetos de classes conceituais definidas durante o *design* conceitual, usando uma linguagem de consulta, permitindo assim que o nodo seja definido por uma combinação de atributos de diferentes classes relacionadas no esquema conceitual. Eles contém atributos unitários e *links*.

*Links* refletem os relacionamentos que serão utilizados pelo usuário final; são realização navegacional dos relacionamentos. Classes de *links* são definidas especificando-se atributos, comportamentos, objetos fontes, alvo e cardinalidade. Atributos do *link* expressam propriedades do mesmo podem ser úteis para definir *links* com cardinalidade maior do que um.

Estrutura de acesso agem como índices ou dicionários e são úteis para auxiliar o usuário final encontrar a informação desejada. Menus e índices podem ser citados como exemplos de estruturas de acesso. Essas estruturas são também modeladas dentro das classes e mais tarde caracterizadas como um conjunto de selecionadores, um conjunto de objetos de alvo. Em aplicações multimídia deve-se ter um cuidado especial aos índices, principalmente os visíveis ao usuário. O autor deve tomar algumas precauções na maneira como o usuário pode explorar o ambiente criado na aplicação, de forma a evitar a redundância de informações, e também que o usuário fique perdido na aplicação.

### 4.1.3 DESIGN ABSTRATO

Nesta etapa do processo de modelagem, com a estrutura de navegação já definida, devemos definir quais objetos irão interagir com o usuário, e principalmente a maneira como objetos de navegação diferentes vão ser visualizados, quais objetos irão ativar a navegação, a maneira como os objetos de interface multimídia serão sincronizados e quais transformações de interface irão ocorrer. Uma clara separação entre as duas etapas, a interface de navegação e a abstrata, permite a construção de diferentes interfaces para o mesmo modelo de navegação, mantendo assim um alto grau de independência da tecnologia utilizada para criar a interface para com o usuário, e também permitindo a conformidade com várias necessidades ou preferências do usuário. Na metodologia OOHDM é utilizado o conceito de *Abstract Data View* (ADV), ou Visão de Dados Abstrata, para descrever a interface da aplicação hipermídia.

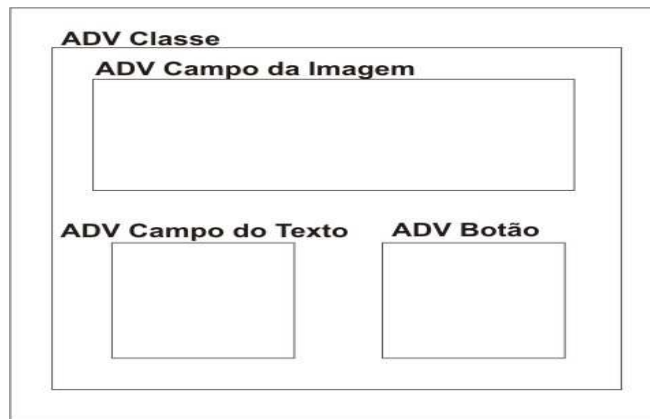
### 4.1.4 ABSTRACT DATA VIEW (ADV)

*Abstract Data View* são objetos que possuem um estado e uma interface, onde a interface pode ser utilizada através de chamadas de funções ou procedimentos regulares, ou eventos de entrada e saída. ADVs são abstratas pois representam apenas a interface e o estado, não sua implementação.

No contexto da OOHDM objetos de navegação como nodos, *links* ou estruturas de acesso agem como objetos da aplicação e a ADV é utilizada para especificar a sua aparência para o usuário. Quando o ADV usado no design de aplicações multimídia ou hipermídia, o mesmo pode ser visualizado como um objeto de interface englobando um conjunto de atributos que define suas propriedades de percepção, e um conjunto de eventos que trata a interação do usuário.

O modelo de projeto ADV foi desenvolvido para especificar clara e formalmente a separação entre a interface do usuário e os componentes de um sistema de *software*, oferecendo um método de projeto independente de implementação ou ferramenta utilizada, criando novas alternativas para reusabilidade dos componentes do projeto e interface. O exemplo da representação de um ADV pode ser visto na figura 13.

FIGURA13 COMPOSIÇÃO DO ADV



Fonte: Schwabe (1999)

A última fase do desenvolvimento da OOADM é a implementação. Nela o desenvolvedor precisa mapear os modelos de interface abstrata e de navegação em objetos concretos disponíveis no ambiente de implementação escolhido. O modelo gerado depois que todas as etapas anteriores foram concluídas, pode ser implementado de uma maneira direta, utilizando as várias aplicações para este objetivo disponíveis no mercado. O uso de um conjunto de construtores de modelagem (objetos e classes), nesta metodologia, permite uma transição suave da modelagem de domínio para o *design* da navegação e interface. Este passo da implementação não precisa de um ambiente de desenvolvimento orientado a objetos, apesar de que o mesmo possa tornar a tarefa mais fácil de ser realizada. Para manter o conceito de reusabilidade, e seqüenciá-lo quando necessário, é importante documentar, além de todos os passos das etapas anteriores, as decisões de implementação.

Como o constante crescimento do fenômeno Internet e dos próprios produtos multimídia, esta metodologia se torna um fator importante para auxiliar no desenvolvimento e na manutenção dos projetos e aplicações para os nichos de mercado acima. Devido ao grande montante de informação entregue diretamente aos usuários destas aplicações, se a mesma não for bem desenvolvida, pode acabar gerando um resultado inverso ao esperado.

## 4.2 MACROMEDIA DIRECTOR

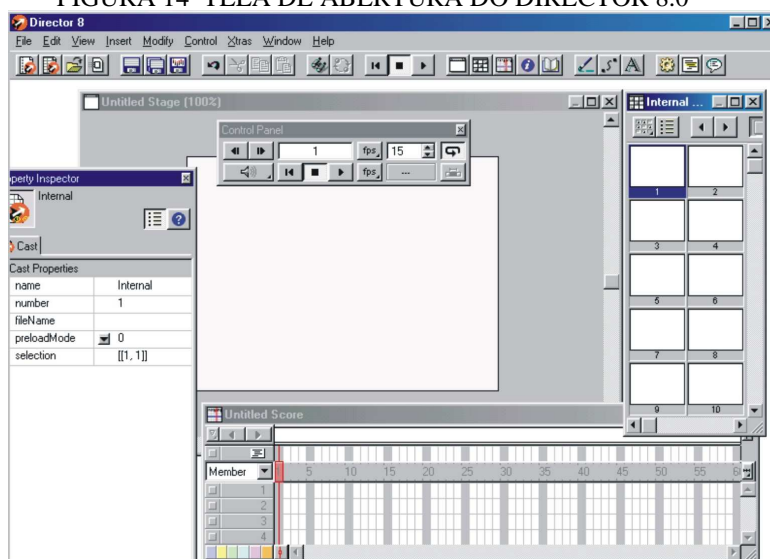
No desenvolvimento do ambiente para ensino de Redes de Computadores baseadas em TCP/IP optou-se pela ferramenta de desenvolvimento de softwares hipermídia Macromedia Director 8.0. Esta escolha deve-se às facilidades oferecidas por este software no

desenvolvimento de aplicações multimídia e hipermídia. Esta característica do Director, permite que este software seja utilizado por programadores ou leigos. Ele utiliza a metáfora de filmes: assim como na produção de um filme, definem-se os atores (castmembers), cenário (stage), a seqüência de cenas (score) e o papel de cada ator (script).

## 4.2.1 APRESENTAÇÃO DO DIRECTOR

A tela de abertura do Director, mostrada na figura 14, apresenta alguns dos elementos citados anteriormente, a área de *CastMembers*, *Stage*, *Score* e *Script*. Estes elementos serão posteriormente detalhados.

FIGURA 14 TELA DE ABERTURA DO DIRECTOR 8.0

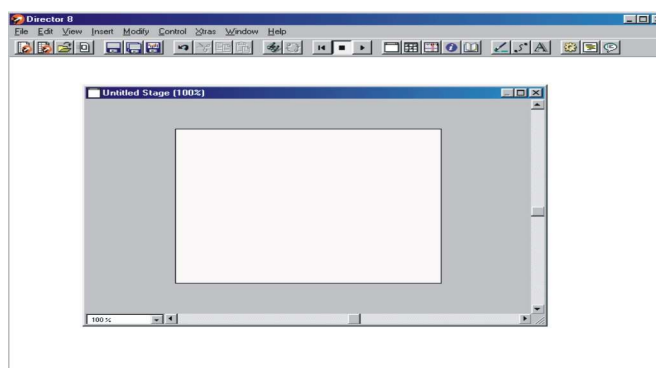


Fonte: Bizzotto (2000)

### 4.2.1.1 STAGE

A janela *Stage*, figura 15, é o palco onde serão projetados os filmes. O *Stage*, é o “micromundo” onde o software multimídia irá existir.

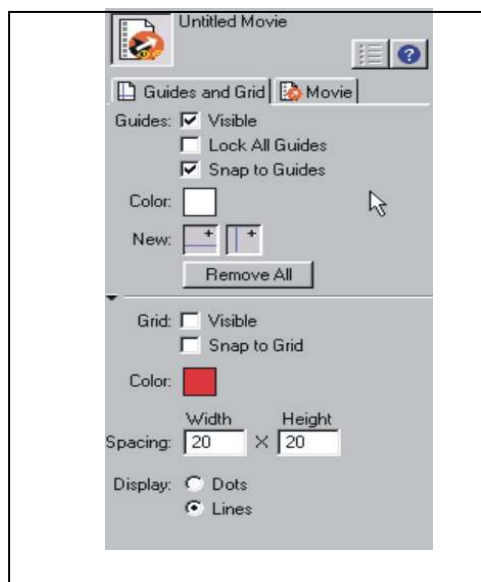
FIGURA 15 STAGE



Fonte: Bizzotto (2000)

No *Stage*, podem ser definidas as propriedades (figura 16) do filme atual (*Modify – Movie-Properties*), tais como, tamanho do *Stage* de apresentação, a localização, a paleta *Default*, entre outros.

FIGURA 16 PROPRIEDADES DO STAGE

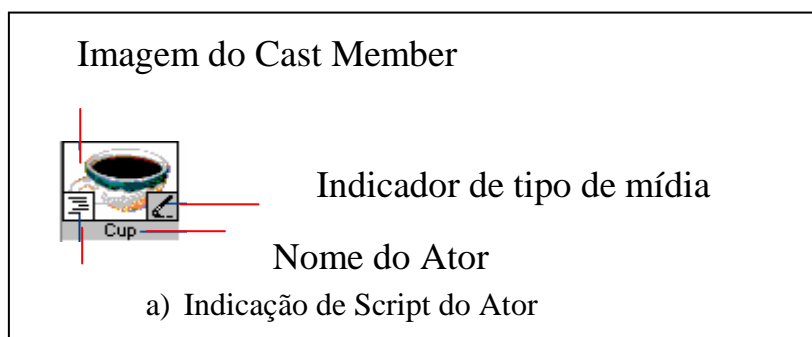


Fonte: Bizzotto (2000)

#### 4.2.1.2 JANELA CAST

A janela *Cast*, contém os *cast members*, que são os atores que atuarão sobre o *Stage*. Esse elenco de atores pode ser formado por imagens, textos, sons, etc. Um membro no *Cast Member*, possui informações conforme as indicadas na figura 17, a seguir:

FIGURA 17 MEMBRO DO CAST













Fonte: Bizzotto (2000)














Um mesmo membro do *Cast member*, pode ser posicionado em locais diferentes do *Stage*, sendo chamado de “Instância de *Cast Member*”, ou seja, uma cópia do original. Logo, as alterações feitas em um original, afetarão suas equivalentes cópias. Por outro lado, uma alteração feita em uma cópia do original, não afetará o *Cast* original. Embora possam existir várias cópias no *Stage* de um mesmo elemento do *Cast*, estes elementos podem assumir papéis diferentes no *Stage*.

De acordo com o tipo de mídia existente no *Cast*, temos representados na tabela 1 , a origem destes tipos de mídia (criados ou importados):

TABELA 1 TIPOS DE ELEMENTOS DO CAST

<b>Tipos</b>	<b>Ícones</b>	<b>Descrição</b>	<b>Criado no Director</b>	<b>Importado</b>
Animated Gif		Gif Animado		X
Bitmap		Imagem	X	X
Check Box		Caixa de verificação	X	
Digital Vídeo		Vídeo Digital	X	X
Film Loop		Seqüência de Animação	X	
Flash Movie		Arquivo Flash		X
Behavior		Componente de Biblioteca	X	X
Button		Botão	X	
Custon Cursor		Cursor Personalizado	X	
Field		Campo	X	



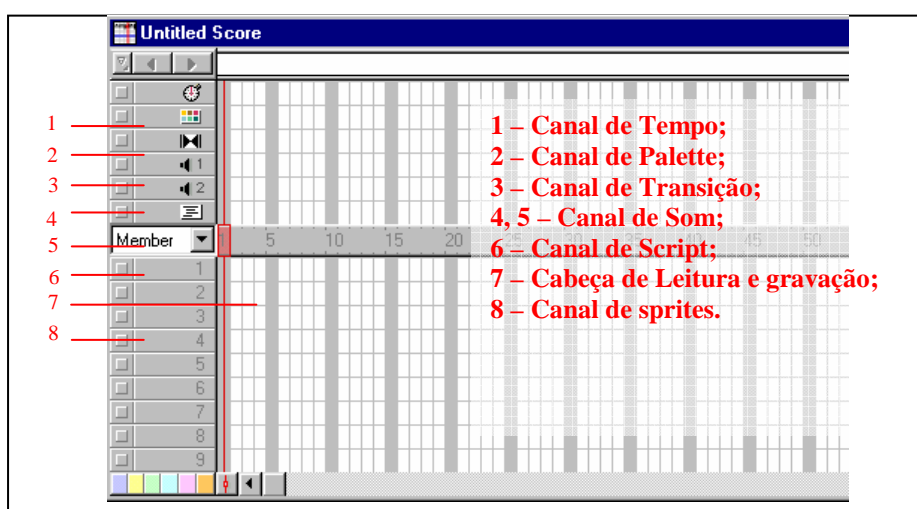
Font		Fonte		X
Linked Bitmap		Link de um arquivo bitmap		X
OLE		Objeto OLE		X
PICT		Imagem		X
Radio Button		Botão de Rádio	X	
Shape		Imagem do tipo Shape	X	
Sound		Arquivo de Som		X
Transition		Efeitos de transição de sprites	X	X
Xtra		Plugin	X	X
Palette		Paleta	X	X
Quick Time Video		Vídeo Digital		X
Script		Script Lingo	X	
Shockwave Audio		Som compactado para internet		X
Text		Texto	X	X
Vector Shape		Imagem Vetorial	X	X

### 4.2.1.3 SCORE

Conforme mencionado anteriormente, os *Cast Members* contém os atores do *Stage*. A janela *Score* controla quando e como as ações vão ocorrer.

O *Score* é dividido em linhas (canais) e colunas (frame). A intersecção de uma coluna com uma linha, denominamos célula. Além dos canais usuais para o gerenciamento dos *Cast Members*, ele possui seis canais especiais para efeitos, conforme mostra a figura 18.

FIGURA 18 CANAIS DO SCORE



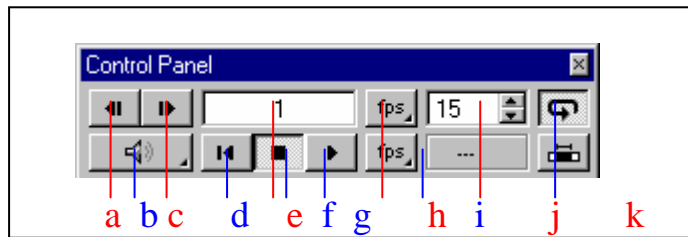
Fonte: Bizzotto (2000)

Os itens dispostos na Figura 18, são responsáveis: (1) pelo gerenciamento do tempo da apresentação e acontecimentos no *Stage*, (2) a possibilidade de utilização de um certo grupo de cores (palette) em um dado filme, (3) gerar um efeito de transição de uma cena para outra, (4,5) utilização de sons ou narrações, (6) permitir a programação em *Lingo*, (7) acompanhar a execução dos *frames*, (8) mostrar todos os *Cast Members* que irão aparecer no *Stage*, respectivamente.

### 4.2.1.4 CONTROL PANEL

A janela *Control Panel*, é comumente utilizada durante o desenvolvimento, ao executar, parar, avançar ou retroceder um filme. Permite ainda o controle de repetições, no caso de optar-se por uma execução contínua do filme, e também a habilitação ou não dos sons, conforme figura 19.

FIGURA 19 CONTROL PANEL



Fonte: Bizzotto (2000)






Os componentes que compõem o Control Panel são:



- a) botão step backward: retrocede um frame;
- b) botão sound: controla a intensidade de som no filme;
- c) botão step forward: avança um frame;
- d) botão rewind: retorna para o frame 1;
- e) frame counter: indica onde se encontra a cabeça de leitura e gravação;
- f) botão stop: interrompe a exibição do filme;
- g) botão play: inicia a exibição do filme;
- h) botão de modo de tempo: permite o andamento do filme em frames por segundo;
- i) botão modo de tempo real: velocidade real da apresentação do filme;
- j) tempo display: indica o andamento do filme;
- k) loop playback: permite que o filme fique em loop, executando.

#### 4.2.1.5 BARRA DE FERRAMENTAS

A barra de ferramentas do Director 8.0 compreende os seguintes itens apresentados conforme a tabela 2:

TABELA 2 BARRA DE FERRAMENTAS DO DIRECTOR 8.0

Ferramentas	Descrição	Atalhos
	New Movie – cria novo filme	Ctrl + N
	New Cast – cria novo Cast	Ctrl + Alt + N
	Open – abre um filme ou cast externo	Ctrl + O
	Import – importa cast members para a janela ativa	Ctrl + R
	Save – salva o filme e os cast ligados a este	Ctrl + S

	Save All – salvar tudo	
	Save As Shockwave Movie – salva como filme shockwave	
	Undo – retorna uma ação	Ctrl + Z
	Cut – Recortar	Ctrl + X
	Copy – Copiar	⌘ + C
	Paste – Colar	Ctrl + V
	Find Cast Member – encontrar Cast Member	Ctrl + ;
	Exchange Cast Members -	Ctrl + E
	Rewind – posiciona a cabeça do playback(leitura e gravação) no frame 1	Ctrl + Alt + R
	Stop – para a execução do filme	Ctrl + Alt + .
	Play – inicia a execução de um filme	Ctrl + Alt + P
	Stage – aciona o Stage	Ctrl + 1
	Cast Window – aciona a janela Cast	Ctrl + 3
	Score Window – aciona a janela Score	Ctrl + 4
	Sprite Inspector – aciona a janela Sprite Inspector	Ctrl + Alt + S
	Library Palette – aciona a janela Library Palette	
	Paint Window – aciona a janela de Paint	Ctrl + 5
	Vector Shape Window – aciona a janela de Vector Shape	Ctrl + Shift + V
	Text Window – aciona a janela Text	Ctrl + 6
	Behavior Inspector – aciona a janela Behavior Inspector	Ctrl + Alt + ;
	Script Window – aciona a janela Movie Script	Ctrl + O
	Message Window – aciona a janela Message	Ctrl + M

## 4.3 LINGO

Lingo é o nome da linguagem utilizada pelo Director, para a produção mais efetiva junto ao usuário. O Lingo é uma linguagem de “script”, cuja sintaxe e construção se aproxima da língua inglesa em sua forma usual, ou seja, falada. Os scripts podem ser compostos de um único comando, assim como de uma seqüência deles, dispostos em conjuntos, similares a parágrafos. A complexidade varia de acordo com aplicação.

Como nas demais linguagens de programação, o Lingo segue algumas regras (Epstein, 1998):

- a) todo *script* é formado por handlers, que em outras linguagens normalmente são definidos como sub-rotinas ou funções. Um handler inicia-se com *on* e é encerrado por *end*;
- b) variáveis locais: utilizadas somente dentro do *handler* onde foi criada;
- c) variáveis globais: utilizadas por diferentes *frames* de um filme, ou de diferentes filmes;
- d) ordem de precedência de *handlers*:
  - a) *scripts* associados a eventos primários tais como:
    1. *keydownscript*;
    2. *mousedownscript*;
    3. *mouseupscript*;
    4. *timeoutscript*;
    5. script associado a um *sprite script*;
    6. script associado a um *cast member script*;
    7. script associado a um *frame script*;
    8. script associado ao filme (*movie script*);
    9. Orientação a objetos.

Segundo Bizzotto (2000), o *Lingo* permite a programação orientada a objetos, embora não seja considerado uma linguagem orientada a objetos. Através das variáveis globais

privadas, cada objeto possui características próprias. A terminologia utilizada na orientação objeto difere do *Lingo*, sendo comparadas na tabela 3:

TABELA 3 TERMOS LINGO

<b>Termo no Lingo</b>	<b>Equivalente em Orientação Objeto</b>
Parent Script	Classe (Class)
Child object	Instância de classe (class instance)
Property variable	Variável de instância (instance variable)
Handler	Método
Ancestor script	Super classe

Fonte: Adaptado de (Epstein, 1998)

Características que possibilitam a orientação a objetos:

- a) a função “NEW” permite cópias idênticas de um determinado objeto através de um script denominado Parent Script;
- b) através das propriedades, Property, os objetos mantêm suas próprias variáveis globais privadas;
- c) os objetos podem dividir propriedades ou handlers através do Ancestor.

### 4.3.1 OS SCRIPTS

Os *scripts* de Lingo podem variar quanto a sua localização e tipo (Epstein, 1998):

- a) *sprite script*: método mais utilizado em *sprites*. Clicando sobre o *sprite* com o botão direito do mouse, gera um *behavior script* (procedimento do ator). Podem ser aplicados múltiplos *Behaviors* (comportamentos) a um *sprite* (objeto). Este efeito, determina que quando algo ocorrer no *sprite*, este efeito se restringirá apenas àquele *sprite*.;
- b) *cast members script*: definido ao colocar-se um objeto no *cast* e selecionar-se o botão de *cast member script*. O comportamento definido a este *cast*, o afetará durante toda a apresentação do filme;
- c) *frame script*: é definido na barra de *script* do *score*, determinando o comportamento daquele frame, definindo o fluxo de apresentação;
- d) *movie script*: posicionando o cursor em um *cast* vazio, e selecionando a opção de menu *Window – Script*, é possível definir um *movie script*. Nele são criadas

funções, definidas variáveis, comportamentos iniciais, *frame-a-frame*, e finalizadores do programa.

### 4.3.2 EVENTOS

Alguns eventos são restritos ao tipo de script correspondente, sendo assim, executados somente a partir do movie script a que pertencem. Conforme tabela 4, (Epstein, 1998):

TABELA 4 EVENTOS E TIPOS DE SCRIPT

Movie script	Frame script	Cast member script e sprite script
On startMovie	On enterFrame	On keyDown
On stopMovie	On exitFrame	On keyUp
On timeOut	On keyDown	On mouseDown
On idle	On keyUp	On mouseUp
On enter Frame	on mouseDown	On mouseEnter
On exitFrame	On mouseUp	On mouseLeave
On keyDown	On mouseEnter	On mouseWithin
On keyUp	On mouseLeave	On rightMouseDown
On mouseDown	On mouseWithin	On rightMouseUp
On mouseUp	On prepareFrame	
On mouseEnter	On cuePassed	
On mouseLeave	On rightMouseDown	
On mouseWithin	On rightMouseUp	
On prepareFrame		
On cuePassed		
On rightMouseDown		
On rightMouseUp		

### 4.3.3 LISTAS

As listas em Director, ou arrays em outras linguagens, são “variáveis” que podem conter diversos elementos. A alocação dos elementos na memória, é feita segundo a necessidade desta, não sendo portanto necessárias as informações quanto ao tamanho dos elementos da lista. O primeiro elemento da lista no Director é contado como “1”, diferentemente das demais linguagens que consideram o primeiro elemento de uma lista “0”. A tabela 5 mostra a sintaxe para a criação de listas:

TABELA 5 SINTAXE PARA A CRIAÇÃO DE LISTAS

Sintaxe	Criando uma lista
Set the list to [ ]	Para criar uma lista linear vazia
Set the list to [:]	Para criar um propriedade de lista vazia, você pode usar um operador de lista para especificar valores na lista.
Especifique a lista de elementos com os seqüência da função list(). List(value1, value2, value3...)	Para criar uma lista linear usando a função list() Exemplo: set designers = list("Gee", "Kayne", "Ohashi")

## 4.4 PERFORMANCE

Segundo Bizzotto (2000), a performance dos filmes poderá ser otimizada verificando-se os seguintes requisitos:

- reduzir as animações: o movimento simultâneo de sprites pode acarretar uma performance indesejada para o filme. Sugere-se a criação de cast members menores, que juntos compõe o todo;
- os efeitos de tinta da propriedade "Ink" do Score, também exercem influência diminuindo a velocidade de uma animação. Preferencialmente, usa-se o efeito copy, a menos que sejam indispensáveis os demais efeitos;
- evitar aumentar o tamanho dos *sprites*;
- áreas brancas, espaços em torno dos *cast members* devem ser reduzidos a fim de que os cast's também reduzam o seu tamanho e venham a desenvolver uma performance melhor durante a execução;
- evitar o uso de sons diversos simultaneamente;
- evitar transições em *frames* onde o filme está em "loop".

Após uma breve visão das ferramentas e do funcionamento do Macromedia Director 8.0 será abordado o desenvolvimento e especificação do protótipo de um ambiente de apoio ao ensino da arquitetura TCP/IP.



## 5 DESENVOLVIMENTO E ESPECIFICAÇÃO DO TUTORIAL

O presente trabalho visa demonstrar o potencial didático de um ambiente de ensino multimídia através do desenvolvimento de um ambiente de auxílio ao aprendizado das tecnologias envolvidas nas redes de computadores, em especial as do modelo de referência TCP/IP. O ambiente constituirá de um tutorial que facilite o entendimento do usuário aos princípios gerais de transmissão de dados entre computadores que utilizam o protocolo TCP/IP, demonstrando como se comportam alguns dos protocolos envolvidos, para que possa ser apoio aos recursos didáticos tradicionais.

Neste processo de informatização da educação, o professor tornar-se-á um mediador, facilitador da educação que juntamente com o tutorial proposto fará parte de estratégias pedagógicas determinadas para tornar o estudo uma ação eficiente e prazerosa de construção do conhecimento.

Neste contexto, juntamente com o conteúdo assimilado em sala de aula, questionamentos e dúvidas esclarecidas, o tutorial será mais uma ferramenta de grande auxílio no processo de ensino-aprendizagem, onde as idéias estarão organizadas e divididas por etapas e estas por sua vez mais aprofundadas no assunto.

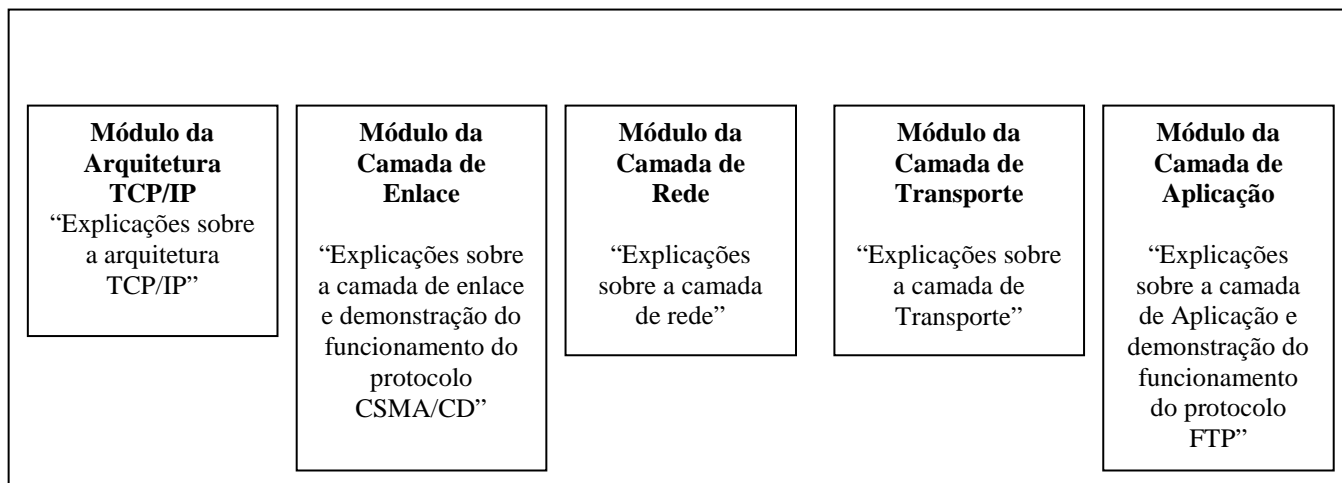
Diferentes graus de conhecimento surgirão proporcionados pelos recursos de sons, textos, imagens, vídeos e animações: desta forma, determinado assunto pode ser visto de diversas maneiras.

Como este tutorial é específico ao ensino de redes de computadores baseadas em TCP/IP, o aluno terá como tela central a lista de camadas da arquitetura TCP/IP, onde serão apresentados conceitos sobre esta arquitetura e suas camadas.

Em algumas camadas há uma demonstração sobre um protocolo específico, explicando através de recursos multimídia o seu funcionamento.

Em termos de conteúdo, o tutorial desenvolvido foi dividido em 5 módulos, conforme apresentado na figura 20.

FIGURA 20 - MÓDULOS DO TUTORIAL



## 5.1 DIAGRAMAS DA OOHDM

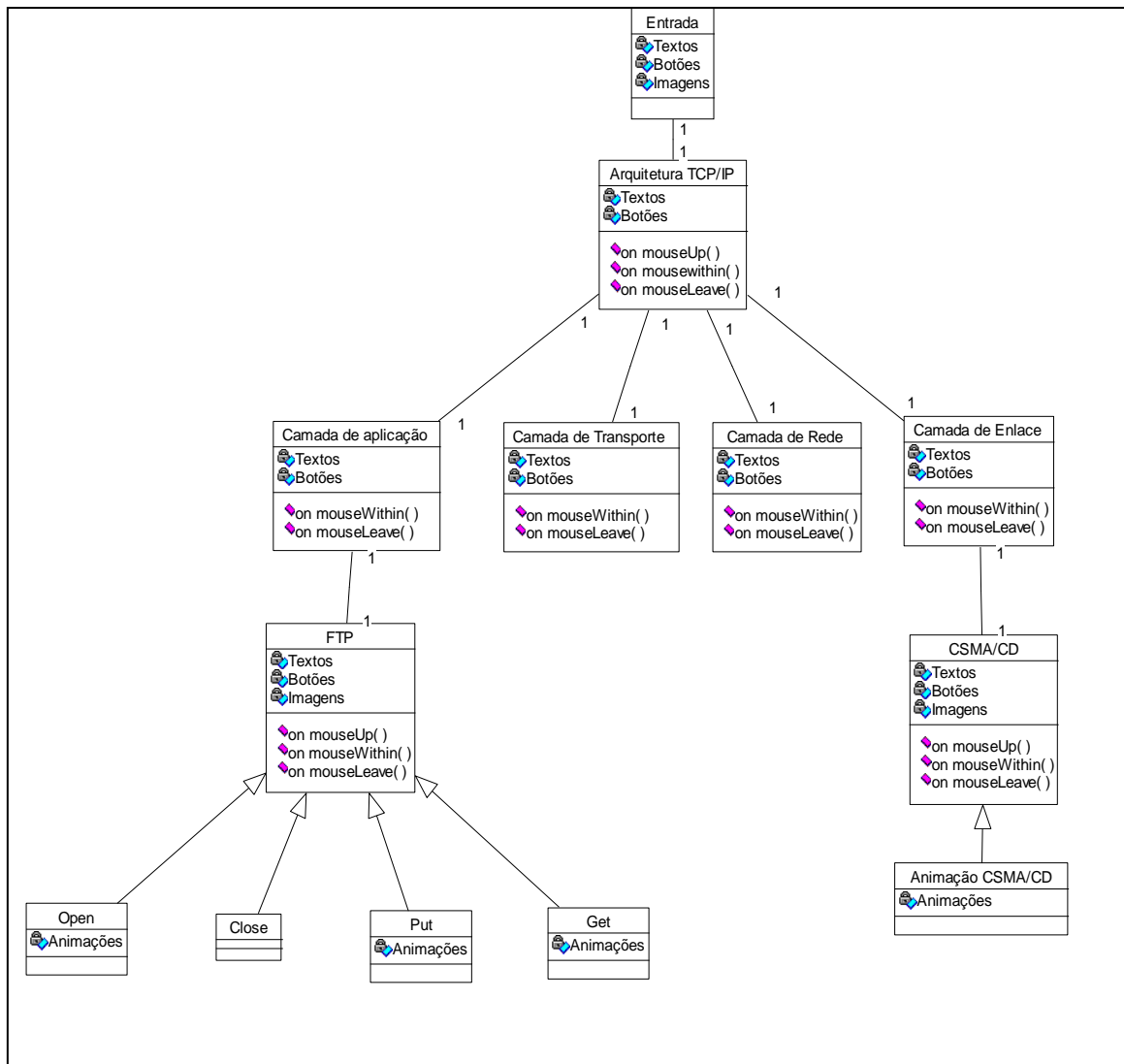
Para especificar o tutorial proposto, foram utilizados os diagramas da OOHDM (*Object Oriented Hipemidia Design Method*) por ser o método que mais atende a necessidade de especificação, planejamento, modelagem para construir aplicações específicas para softwares multimídia, como as desenvolvidas no Director.

### 5.1.1 DIAGRAMA DO PROJETO CONCEITUAL

Na figura 21 está especificado o projeto conceitual do tutorial desenvolvido no Director 8.0.

Apesar de não existir uma ferramenta específica para desenvolver a especificação do OOHDM, utilizou-se a ferramenta Rational Rose por conter a maioria dos elementos existentes para o desenvolvimento do projeto conceitual. As classes representam os diferentes módulos propostos para o ensino e aprendizado, em que cada atributo representa um ou mais elementos existentes no módulo. As operações atribuídas a cada classe especificada se referem às ações associados às animações desenvolvidos no Director.

FIGURA 21 PROJETO CONCEITUAL



### 5.1.2 MODELO NAVEGACIONAL

Na figura 22 é apresentado o modelo navegacional do tutorial proposto, onde uma das suas principais características é a navegação. O modelo navegacional tende a deixar claro os *links*, que refletem os relacionamentos que serão utilizados pelo usuário final levando em conta os tipos de usuários e as tarefas que os mesmos irão realizar no uso da aplicação.

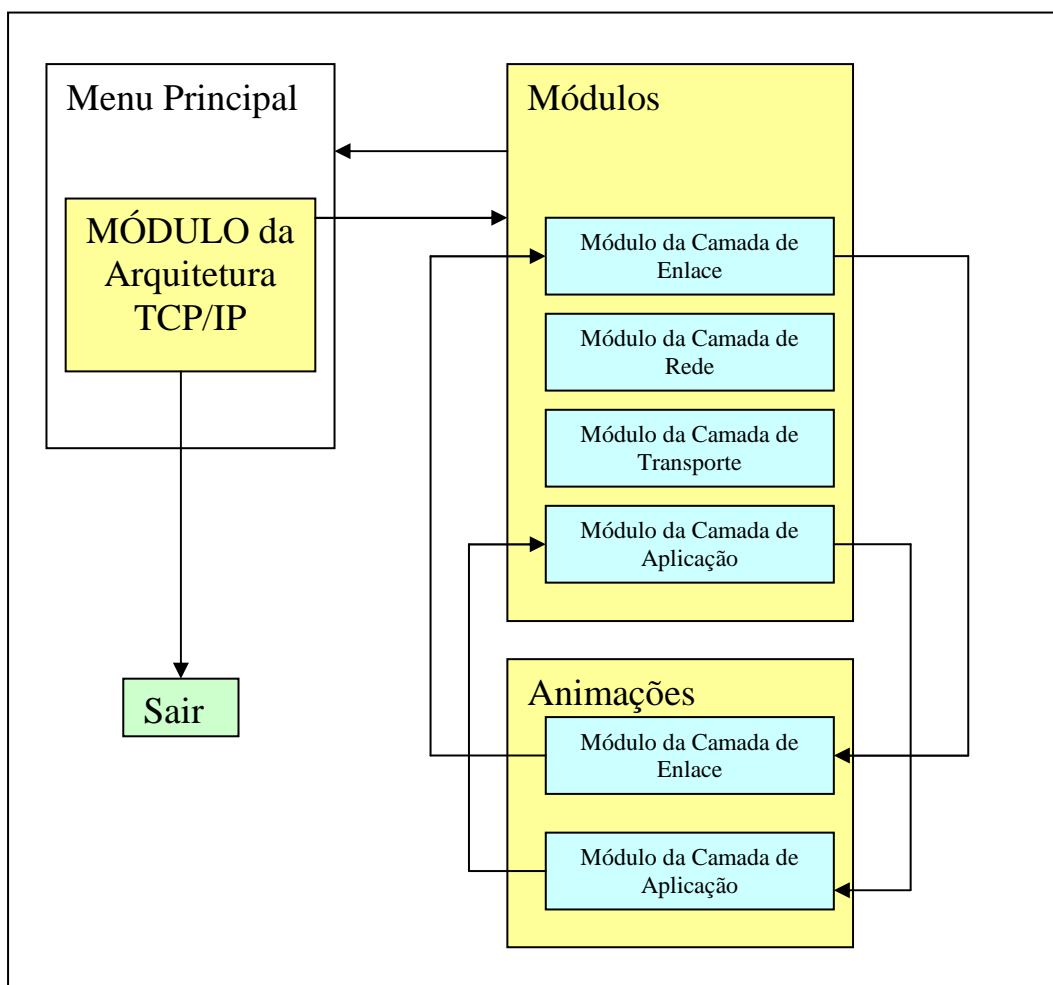


FIGURA 22 MODELO NAVEGACIONAL

## 5.2 IMPLEMENTAÇÃO E FUNCIONAMENTO DO TUTORIAL

Inicialmente o tutorial apresenta uma breve animação como introdução, conforme mostrado na figura 23.

Todas as figuras, animações, textos têm a capacidade de conduzir o usuário a um ambiente de estudo de conceitos e técnicas já visto ou lidos. Como o tutorial é específico a uma tecnologia, o prévio conhecimento é necessário. Lembrando mais uma vez que o tutorial serve para complemento de conceitos estudados. Em hipótese alguma o educador será indispensável, pelo contrário, ele é que será o estimulante para o uso de tecnologias que

auxiliem o aprendizado.

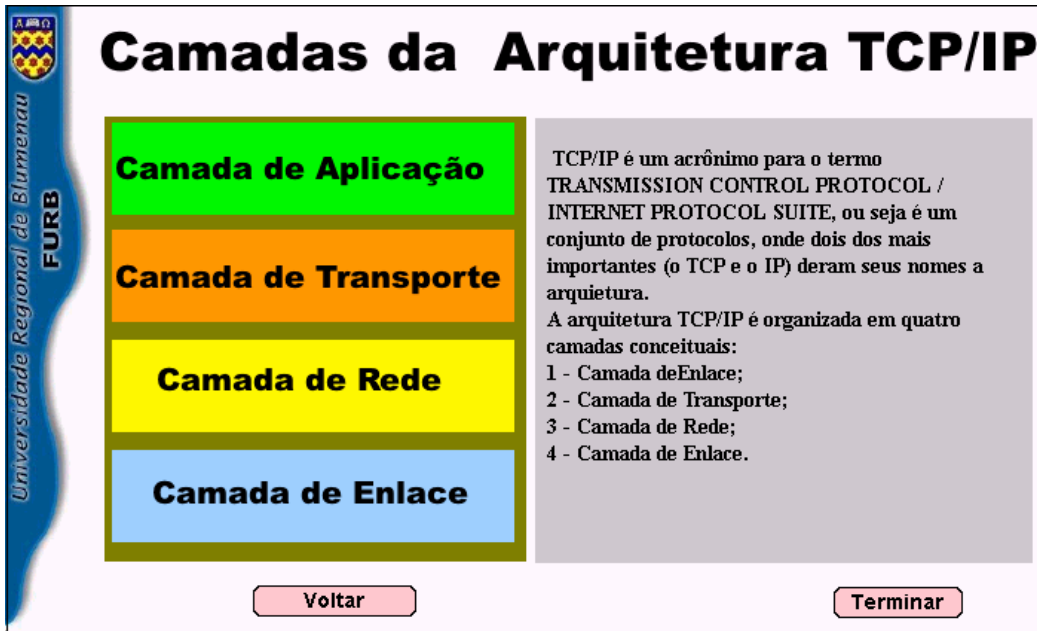
FIGURA 23 TELA DE ENTRADA DO TUTORIAL



Na tela de entrada há o botão “Iniciar Tutorial” que, quando clicado, leva o usuário ao Módulo de apresentação da arquitetura TCP/IP.

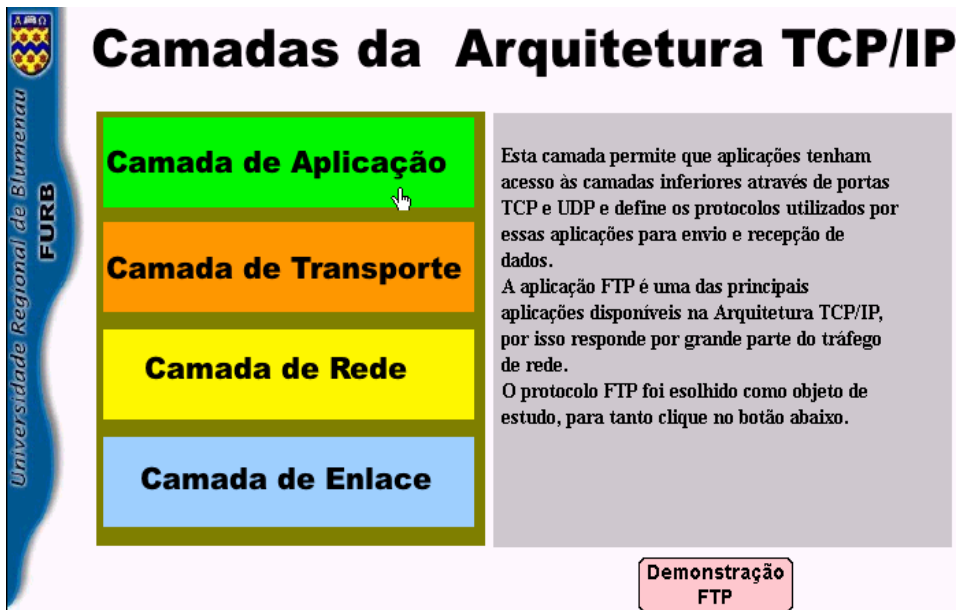
Neste módulo há explicações sobre a Arquitetura TCP/IP e também a opção para terminar o tutorial clicando no botão “Terminar”, conforme está apresentado na figura 24 abaixo.

FIGURA 24 TELA DA ARQUITETURA TCT/IP



Quando o mouse entra na área de cada camada é alterada a sua forma, indicando que em cada camada há um *link* para a explicação e exibição de demonstração conforme é apresentado na figura 25 abaixo:

FIGURA 25 MUDANÇA DO FORMATO DO MOUSE

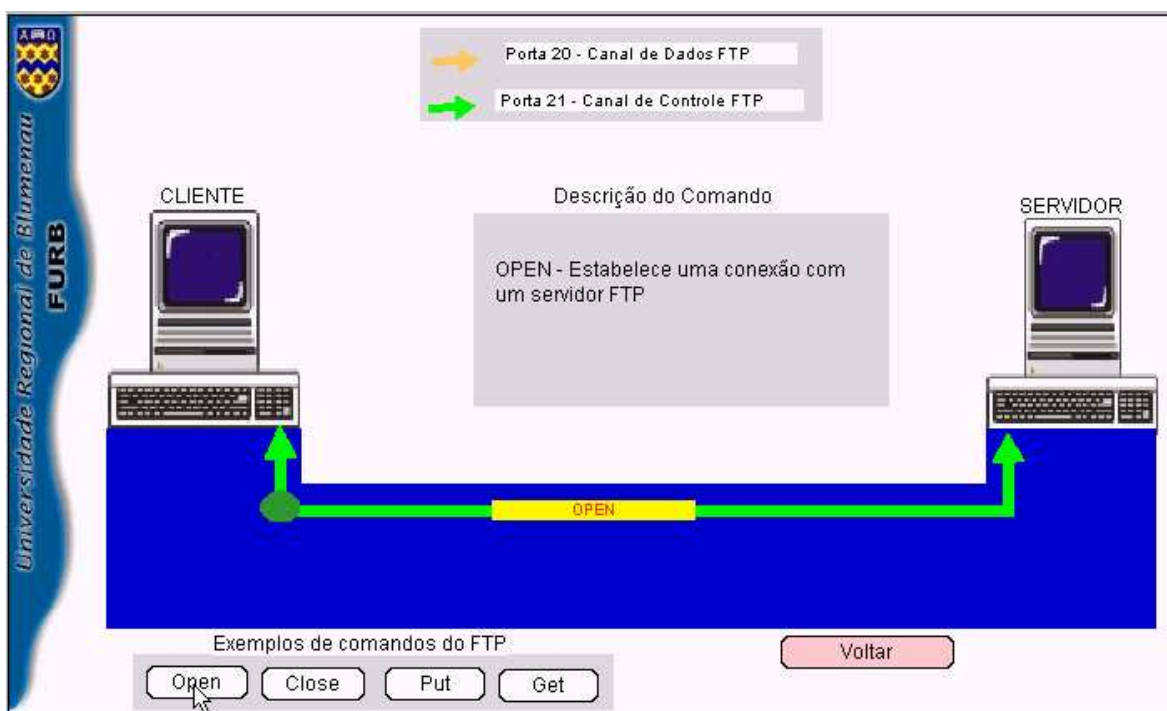


Na camada de Aplicação há um botão chamado de "Demonstração FTP" que faz um *link* com uma animação específica sobre este protocolo.

Há um menu para explicar a função das setas, diferenciadas por cores e um campo texto explicando o comando que foi clicado. Alguns comandos do FTP são apresentados em forma de animação multimídia, conforme a figura 26 mostra abaixo, ativados pelo botões correspondentes aos comandos. Como exemplo, utilizou-se os seguintes comandos:

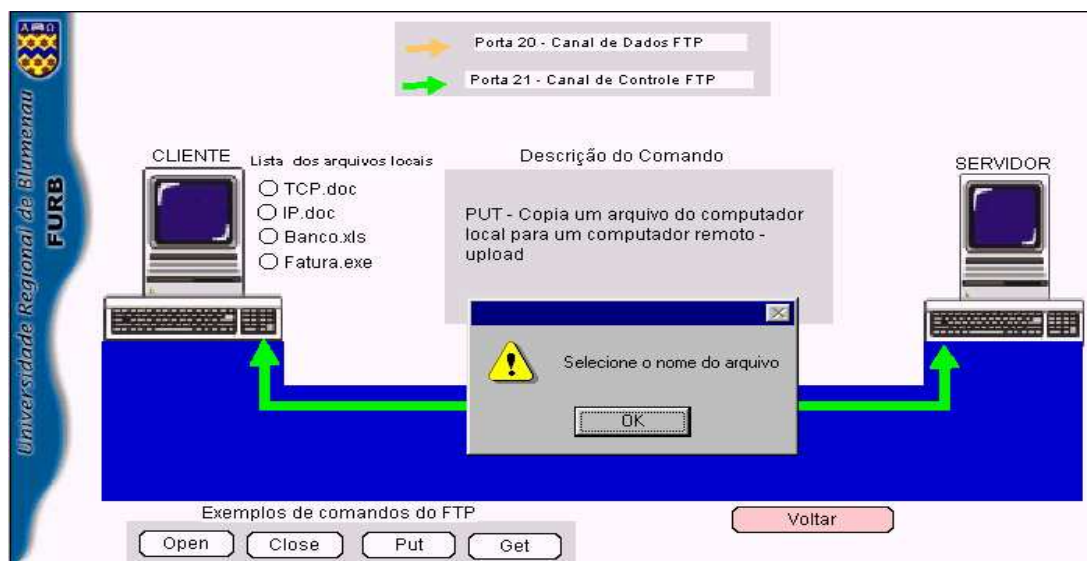
1. OPEN: que estabelece uma conexão com um servidor FTP;
2. CLOSE: que encerra uma conexão;
3. PUT: que copia um arquivo do computador local para um computador remoto;
4. GET: que copia um arquivo do computador remoto para um computador local.

FIGURA 26 TELA DA CAMADA DE APLICAÇÃO - FTP



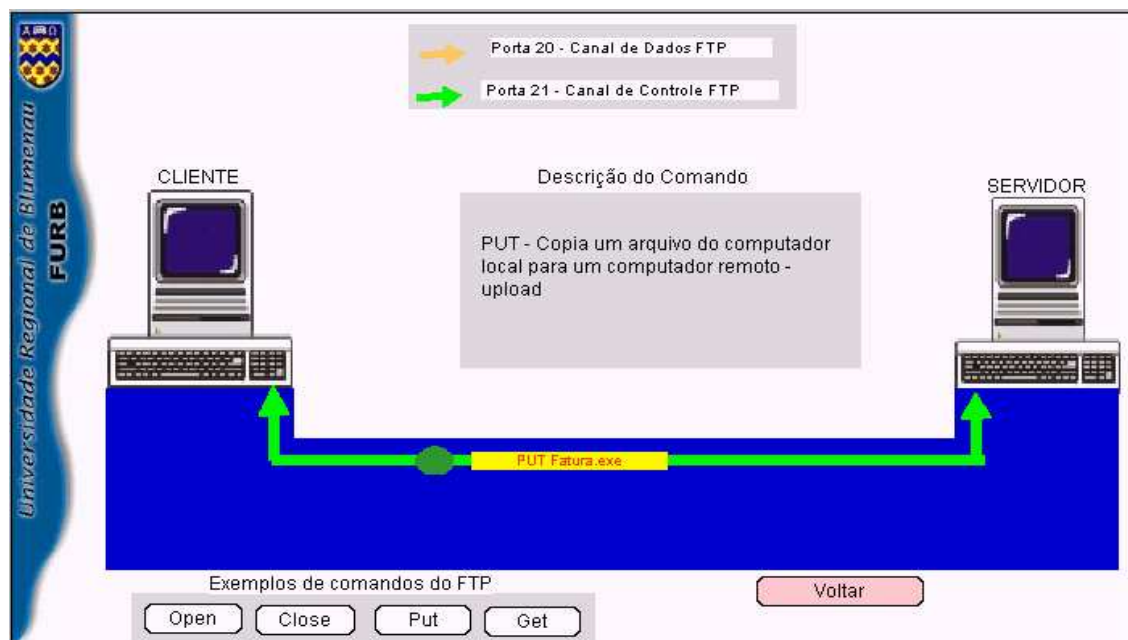
Ao clicar no botão PUT o usuário terá que escolher um arquivo para ser copiado para o computador remoto, para ilustrar a necessidade de selecionar-se um arquivo ativamos uma caixa de alerta, para que o usuário seja conduzido a ação de escolher um arquivo no computador local – Cliente, conforme apresentamos na figura 27 abaixo:

FIGURA 27 COMANDO PUT



Ao ser selecionado uma das quatro possibilidades de arquivos, para melhor exemplificarmos o tutorial, o canal de controle aparece em forma de uma seta de cor verde e em seguida aparece o canal de dados em forma de uma seta de cor laranja, demonstrando o envio de um ou mais arquivos selecionados, conforme é apresentado na figura 28 abaixo:

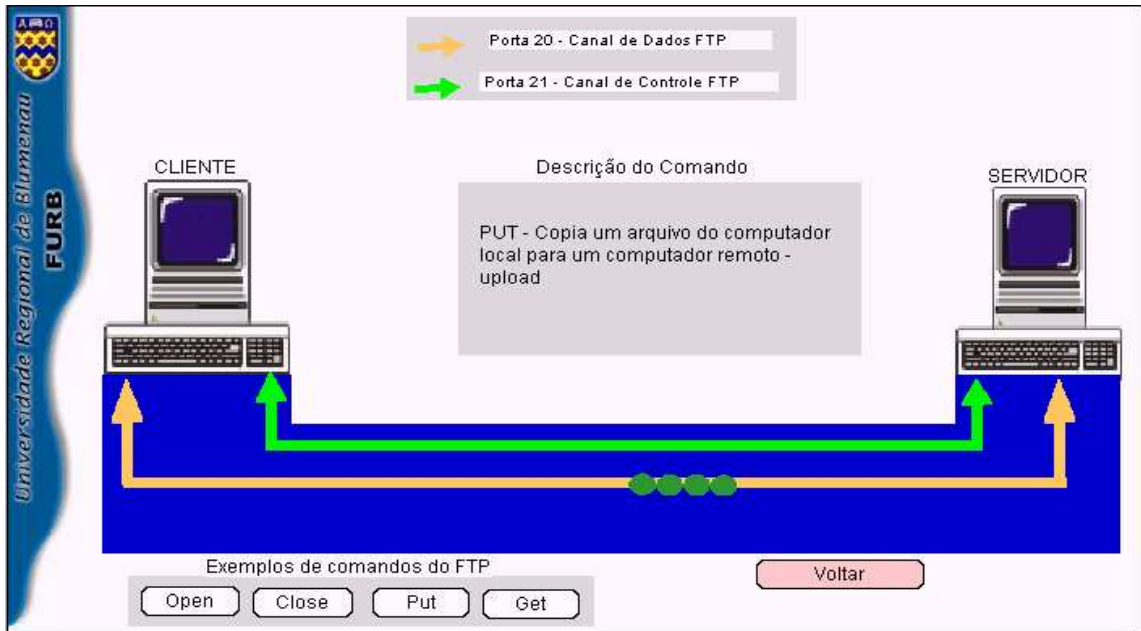
FIGURA 28 FTP – CONEXÃO DE CONTROLE



A figura 29 apresenta a sequência do comando PUT com a transferência do arquivo.

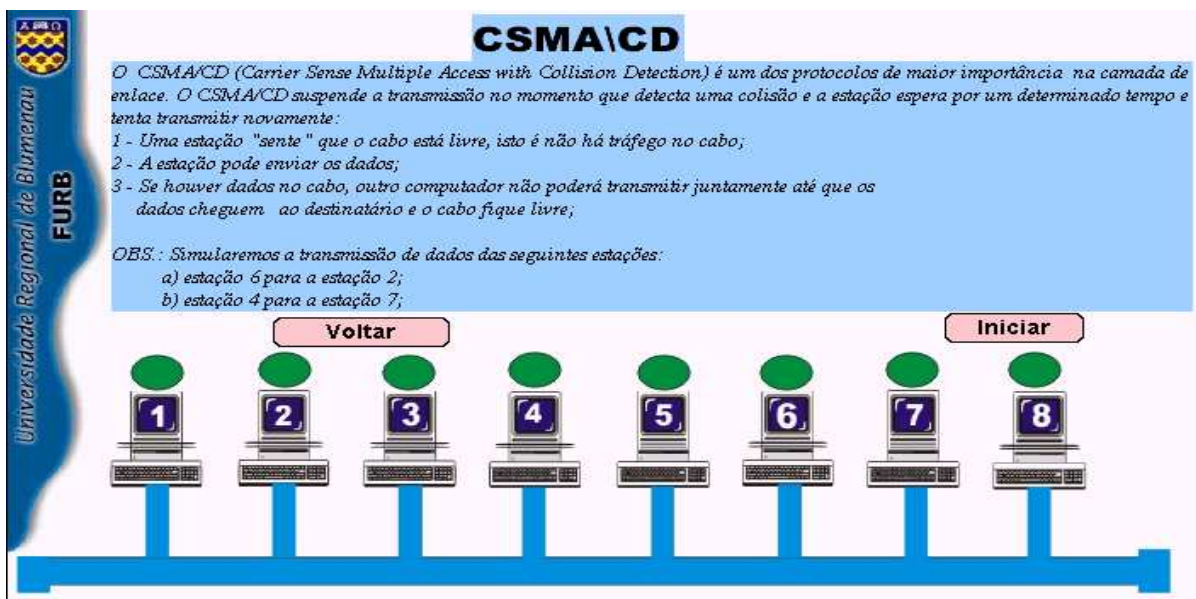


FIGURA 29 FTP – CONEXÃO DE DADOS



Na camada de Enlace há um botão chamado de “Demonstração CSMA/CD” que faz um *link* com uma animação específica sobre este protocolo. Esta tela traz uma breve explicação sobre o que o protocolo faz e a simulação de transmissão de dados, conforme a figura 30 mostra abaixo.

FIGURA 30 FUNCIONAMENTO DO CSMA/CD



Além da simulação de transmissão de dados há também a colisão e retransmissão, que nos é apresentada na figura 31 abaixo.

FIGURA 31 DETECÇÃO DE COLISÃO NO CSMA/CD



Conforme foi apresentado através das figuras acima, toda a implementação foi desenvolvida por eventos simples, como por exemplo, o clique no botão “iniciar” da tela CSMA/CD dispara o seguinte evento:

```
On mouseUp
Go to frame 11
End mouseUp
```

Assim também pode-se associar procedimentos (Behaviors) que o próprio Director possui, como por exemplo, *JUMP TO MARKER BUTTON*, que move a cabeça de gravação para um marcador.

Por ser uma ferramenta de fácil desenvolvimento, o usuário que possui noções de programação pode divertir-se desenvolvendo aplicações multimídia e ao mesmo tempo tornando o ambiente de ensino mais proveitoso.

## 6 CONCLUSÃO

O tutorial alcançou o seu objetivo de desenvolver interfaces didáticas propícias para interatividade, sendo uma ferramenta que juntamente com as experiências obtidas em sala de aula ajudarão no entendimento de assuntos muitas vezes difíceis de serem absorvidos.

Com o uso de um software multimídia ou hipermídia pode-se apresentar idéias organizadas e dividí-las em etapas, e estas por sua vez ainda mais aprofundadas no assunto, auxiliando o raciocínio e podendo atingir diferentes graus de conhecimento.

A animação através do Director tornou-se, portanto, um facilitador de superação das dificuldades didático-pedagógicas, evidenciado pela implementação dos protocolos estudados, visto que imagens, sons, textos e linguagem de programação ajudam a incrementar o processo de aprendizagem através de animações.

A linguagem de programação LINGO faz a distinção de que o Director não é um simples aplicativo de criação de apresentação do tipo Microsoft *PowerPoint* melhorado, pois oferece total controle ao usuário de sistemas interativos.

Assim como pode-se desenvolver softwares simples com o Director, a complexidade pode ser resolvida pela presença da linguagem de programação, deixando nas mãos do usuário o que irá acontecer, como e quando.

Portanto, o tutorial demonstrou que a utilização de softwares multimídia enriquecem o conteúdo de um estudo, pesquisa ou aula, pois permitem tornar visíveis os vários elementos que meramente escritos tornam-se abstratos, porém quando ilustrados passam a ter características como movimento, cor, forma, sons, comportamento, aspectos estes que facilitam a compreensão de qualquer área de conhecimento.

### 6.1 SUGESTÕES

Como sugestão para trabalhos futuros, indicamos a adição de interatividade com novas funcionalidades utilizando todo o potencial que o software Director possibilita.

Especificamente em relação ao tutorial proposto neste trabalho, sugerimos continuar a implementação da arquitetura TCP/IP, simulando outros protocolos de cada camada.

Outra sugestão é, baseado na sequência da especificação desenvolvida neste trabalho, criar tutoriais que possam ser utilizados como ferramentas para outras áreas do conhecimento.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

BIZZOTTO, Carlos Eduardo Negrão. **Director 8: rápido e fácil**. São Paulo: Makron Books, 2000.

COELHO, Marcelo de Miranda. **O uso de estruturas de navegacionais e vistas abstratas de dados no OOHMD e conceitos de objetos multimídia para a construção de uma aplicação**. Disponível em: <<http://tathy.comp.ita.cta.br/~coelho/tgonline/cap1&2.htm/>>. Página acessada em: 30/10/2001.

COMER, Douglas E. **Interligação em rede com TCP/IP**. – 2ª. ed. - Rio de Janeiro : Campus, 1998.

EPSTEIN, Bruce A. **Lingo in a nutshell**. Sebastopol: O'Reilly & Associates Inc., 1998.

FISCHER, Julianne. Sugestões para o desenvolvimento do trabalho pedagógico. In: ii Congresso Internacional de Educação, 2000, Florianópolis. **Anais...Florianópolis: Colégio Coração de Jesus, 2000, p. 228-230.**

FURLAN, José David. **Modelagem de objetos através da UML – the unified modeling language**. São Paulo: Makron Books, 1998.

GADOTTI, Moacir. Perspectivas atuais da Educação. In: ii Congresso Internacional de Educação, 2000, Florianópolis. **Anais...Florianópolis: Colégio Coração de Jesus, 2000, p. 76-80.**

KUROSE, James F. e Keith W Ross. **Computer networking: a top down approach featuring the Internet**. Ed. Addison Wesley Longman, 2001.

MARTINS, Maria de Lourdes Zanettini. Hiperemídia para o ensino de História. In: ii Congresso Internacional de Educação, 2000, Florianópolis. **Anais...Florianópolis: Colégio Coração de Jesus, 2000, p. 228-230.**

PALMA, Luciano. **TCP/IP: guia de consulta rápida**. São Paulo: Ed. Novatec, 1999

SCHWABE, Daniel. **Systematic Hypermedia Application Design with OOHDM**, jan [1995]. Disponível em <http://www.cs.unc.edu/~barman/HT96/P52/section1.html>> Página acessada em 25/09/2001.

SCHWABE, Daniel. **Abstraction, Composition and Lay-Out Definition Mechanisms in OOHDM**, mar [1996]. Disponível em: <<http://www.cs.tufts.edu/~isabel/schwabe/MainPage.html>> Página acessada em 25/09/2001.

SCHWABE, Daniel. **The object oriented hypermidia design model (OOHDM)**. jan [1999] Disponível em: <<[http://www-di.inf.puc-rio.br/~schwabe//schwabe\\_por.html](http://www-di.inf.puc-rio.br/~schwabe//schwabe_por.html). Página Acessada em: 03/11/2001.

SKINNER, Burrhus. Frederic. **Ciência e comportamento Humano**, 5ª ed. São Paulo: Martins Fontes, 1981

SOARES, Luis Fernando. **Redes de computadores : das LANs MANs e WANs as redes ATM**. Rio de Janeiro : Campus, 1995..

SOUZA, Lindeberg Barros de. **Redes de computadores: dados, voz, imagem**. São Paulo: Érica, 1999.

TANENBAUM, Andrew S. **Redes de computadores**. Rio de Janeiro: Prentice-Hall do Brasil, 1997.

VALENTE, José Armando. **Análise dos diferentes tipos de software utilizados na educação**. Campinas: NIED, 1999.

ZANDOMENEGHI, Ana Lúcia A. de Oliveira. **A informatização na Educação**. In: ii Congresso Internacional de Educação, 2000, Florianópolis. **Anais...**Florianópolis: Colégio Coração de Jesus, 2000, p. 250-252.