

# Introdução à Televisão Digital Interativa: Arquitetura, Protocolos, Padrões e Práticas

Jorge Fernandes<sup>1,2</sup>, Guido Lemos<sup>3</sup> e Gledson Silveira<sup>3</sup>

<sup>1</sup>Departamento de Ciência da Computação – Universidade de Brasília (jhcf@cic.unb.br).

<sup>2</sup>Afastado do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte.

<sup>3</sup>Departamento de Informática da Universidade Federal da Paraíba (guido@di.ufpb.br, gledson@dimap.ufrn.br).

## **Abstract**

*This text presents an overview of Interactive Digital Television Systems (IDTV) aimed for introducing computing students and professionals in the field. The architecture, protocols, and middleware standards for IDTV systems comprise the theoretical view. Practical aspects are approached through the demonstration of steps necessary for construction and use of a simplified test platform to be employed for initial development of Interactive Digital Television applications.*

## **Resumo**

*Este texto apresenta uma introdução teórico-prática à Televisão Digital Interativa (TVDI), para profissionais e estudantes de computação. Aspectos teóricos são abordados através da apresentação de arquiteturas, protocolos e padrões de middleware para sistemas de TVDI. Aspectos práticos são introduzidos através da descrição e demonstração de como montar e utilizar uma plataforma de testes simplificada, capaz de permitir o desenvolvimento básico de aplicações para TVDI.*

# Introdução à Televisão Digital Interativa: Arquitetura, Protocolos, Padrões e Práticas

## Tabela de Conteúdos

1. Introdução	6
2. Conceitos Básicos e Histórico	8
2.1 A Imagem em Movimento e o Olho Humano	8
2.2 Imagem Televisiva e Padrões de TV Analógica	8
2.3 A Televisiva de Alta Definição	9
2.4 O Surgimento da TV Digital	9
3. Arquitetura de Sistemas de TVDI	10
3.1. Arquitetura de Software e Sistemas	10
3.2. Detalhes Arquiteturais de um Sistema de TV Atual	11
3.3. Arquitetura Básica de um Sistema de Televisão Digital	13
3.4. Novos Conceitos Introduzidos do Modelo de TV Digital	15
3.5. Arquitetura de Sistemas de TV Digital Pseudo-Interativa	16
3.5. Arquitetura do STB Interativo	19
3.5. Arquitetura do Gerador de Carrossel	20
4. Padrões para TVDI	21
4.1. Padrões Mundiais de TVDI	22
4.2.1. DVB	23
4.2.2. ATSC	25
4.2.3. ISDB	26
4.3. Padrões para Modulação e Transmissão	28
4.4. Padrão para Multiplexação e Transporte	30
4.5. Padrões para Codificação e Compressão	34
5. Padrões de Middleware	35
5.1. Blocos Fundamentais	36
5.1.1. DAVIC	37
5.1.2. HAVi	38
5.1.3. Java TV	39
5.2. Padrões de Midl्लware para TVDI	41
5.2.1. DVB	41

5.2.2. DASE/ATSC	41
5.2.3. ARIB/ISDB	42
6. Prática em Desenvolvimento de Aplicações para TVDI	42
6.1. Uma plataforma de desenvolvimento de Software para TVDI	42
6.2. Uma plataforma pessoal para desenvolvimento DVB-J/MHP.	43
6.3. Uso de Emuladores	44
6.4. O Emulador de Xlets da Espial	44
6.4. Um Exemplo de Xlet: QuizXlet.	47
7. Conclusões e Agradecimentos	53
Referências	54

## Sobre os Autores

Glêdson Elias recebeu o título de Doutor em Ciência da Computação pelo Centro de Informática (Cin) da UFPE, em 2002. A sua tese apresenta um framework para distribuição, gerenciamento e evolução de sistemas de software baseados em componentes, explorando conceitos de engenharia de software e sistemas distribuídos. É Professor Adjunto no Departamento de Informática da UFPB. No período de 1993 a 1996, foi Coordenador Técnico dos Projetos: A Rede de Pesquisa da UFRN; Implantação do Ponto de Presença da RNP no RN; e Internet/RN – Integração da Comunidade Científica do RN, onde atuou na implantação da Internet no Estado do Rio Grande do Norte e na UFRN. Foi participante ativo na sede do Projeto RNP em Campinas de 1991 a 1993. Ocupou cargos de Direção e Assessoria na UFRN. Leciona na graduação e pós-graduação em cursos como redes de computadores, redes de alta velocidade, gerência de redes, redes móveis e sem fio, arquitetura TCP/IP, middleware, e sistemas operacionais. Pesquisa nas áreas de redes de computadores, sistemas distribuídos e engenharia de software. Nestas áreas está avaliando protocolos de roteamento para redes móveis ad hoc utilizando o simulador NS2; concebendo uma arquitetura de mobilidade para a UFRN; desenvolvendo aplicações para dispositivos móveis e televisão interativa usando J2ME e JavaTV; desenvolvendo uma plataforma de distribuição de vídeo sob demanda; desenvolvendo um ambiente de execução para componentes JavaBeans; e desenvolvendo um ambiente para busca e recuperação de componentes em repositórios distribuídos. Participa como pesquisador colaborador de projetos de pesquisa com outras instituições como: InfraVida (UFPE) – desenvolvendo um sistema de videoconferência com aplicação em telemedicina; e I2TV/HiTV (UFPB) - desenvolvendo um middleware para televisão digital interativa. Atua ainda no Grupo de Trabalho RNP de Vídeo Digital 2003/2004, onde realiza a coordenação local da equipe de Natal.

Guido Lemos de Souza Filho é Doutor em Informática pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO) e atualmente trabalha como professor do Departamento de Informática da UFPB. É um dos autores do livro "Redes de Computadores: das LANs, MANs e WANs às Redes ATM", best seller da Editora Campus com mais de 35.000 exemplares vendidos. Possui mais de 70 publicações em diversas conferências nacionais e internacionais nas áreas de redes de computadores e sistemas multimídia. Coordena os projetos ICSpace e I2TV financiados pelo CNPq, e HiTV financiado pela FINEP. Nestes projetos estão sendo realizadas pesquisas nas áreas de Televisão Digital, Museus Virtuais, Vídeo sob Demanda, Videoconferência e Redes de Alta Velocidade. É diretor executivo do LARC (Laboratório Nacional de Redes de Computadores) e coordenador da Comissão Especial de Sistemas Multimídia e Hiperfídia da SBC (Sociedade Brasileira de Computação). É também coordenador do Grupo de Trabalho de Vídeo Digital da RNP, onde atua na organização dos experimentos com vídeo digital no projeto Internet2 Brasil. Contribui para o projeto do Sistema Brasileiro de Televisão Digital (SBTVD) participando do Grupo de Trabalho sobre Serviços, Aplicações e Conteúdo (GTSAC).

Jorge Fernandes é Doutor e Mestre em Informática pelo Centro de Informática da UFPE. Está afastado do Departamento de Informática e Matemática Aplicada da UFRN,

atualmente atuando no Departamento de Ciência da Computação da Universidade de Brasília. Possui mais de 18 anos de experiência em educação, pesquisa, desenvolvimento e consultoria em engenharia de software e sistemas, tendo desenvolvido vários softwares e atuado em consultoria de projetos de tecnologia da informação e desenvolvimento de software de pequeno, médio e grande porte. Leciona em cursos de graduação e pós-graduação em engenharia de software, sistemas de informação, reengenharia de processos, arquitetura de software, técnicas de programação e programação para dispositivos móveis. Foi autor e apresentador de vários cursos, mini-cursos e treinamentos em eventos locais, regionais e nacionais, entre eles alguns promovidos pela SBC, como Jornada de Atualização em Informática da SBC 1998 (Ciberespaço), Jornada de Atualização em Informática da SBC 1996 (Introdução à Java), Simpósio Brasileiro de Banco de Dados 2000 (Integração com Java e Banco de Dados) e Simpósio Brasileiro de Sistemas Multimídia e Hiperímia 2000 (Sistemas Multimídia com JMF – Java Media Framework). Possui extensa experiência de educação e treinamento em outros temas como programação para dispositivos móveis, reuso de software, métricas de qualidade de código, padrões de design, metodologias e processos iterativos de desenvolvimento de software. Atua em projetos de pesquisa e desenvolvimento nas áreas de Engenharia de Software, Vídeo e Televisão Digital, como: BLNET – Desenvolvimento de Protótipo de Serviço de Acesso em Banda Larga à Internet usando Plataforma de TDVI; HiTV – Software e Hardware para Televisão Digital de Alta Definição e I2TV – Infra-estrutura Internet2 para Sistemas de Televisão Digital Interativa. Possui ainda interesse em áreas como Educação em Informática, Divulgação Científica e Modelos de Transferência de Tecnologia.

## 1. Introdução

A televisão, como os outros meios de comunicação de massa, segue a tendência mundial do movimento de convergência digital, que é a fusão entre os mercados de mídias e Tecnologias da Informação e Comunicação (TICs), através de um acelerado processo de substituição de suas plataformas analógicas por plataformas e tecnologias digitais interoperáveis. Onde e quando ocorre, a convergência digital provoca grandes mudanças na cadeia de produção e consumo de mídias. A título de exemplo, podemos dizer que a produção de filmes, jornais e software, que antes empregava processos produtivos diversos, tende hoje a empregar a mesma gama de profissionais e atividades. Transformações similares ocorrerão também com a transição da TV analógica para a TV digital (BroadcastPapers, 2004a), que já está ocorrendo, em três ondas de impacto, que ocorrerão inclusive no Brasil.

A primeira onda de impacto, já sentida internamente por várias redes de TV brasileiras, é a necessária substituição dos equipamentos de captura, edição e transmissão interna de áudio e vídeo analógicos, por similares digitais, visando melhoria da imagem e som.

A segunda onda de impacto, a ser sentida pelo conjunto da sociedade, é a necessária adoção de um padrão uniforme para codificação, transmissão, modulação, difusão e recepção digital de programas. No Brasil, este impacto é maior nos sistemas de TV Digital Terrestre (DTT – *Digital Terrestrial Television*), comumente usado nos centros urbanos, onde o maior desafio é a escolha técnica-econômica-social-política do formato de modulação de sinais (BroadcastPapers, 2004c). Está em discussão no país a definição do padrão a ser adotado para DTT, sendo as opções os padrões ATSC (americano), DVB (europeu), ISDB (japonês), além do padrão chinês que ainda está em desenvolvimento. Todos os padrões para o sistema de transmissão que estão sendo avaliados convergem com relação ao formato adotado para a codificação digital dos dados a serem transmitidos: o padrão ISO MPEG-2. Este padrão, que na realidade define não uma, mas um conjunto de normas, divide-se em MPEG Vídeo, MPEG Áudio e MPEG Sistema. O MPEG Vídeo define métodos para compressão e codificação de vídeo digital com diferentes expectativas de qualidade, dentre elas, SDTV (*Standard Digital Television*) e HDTV (*High Definition Digital Television*). O MPEG Sistema define formatos e protocolos para montagem de pacotes para transmissão de fluxos elementares multiplexados. Os fluxos elementares transportam *streams* de vídeo, áudio ou dados. Porém, a escolha deste padrão, apesar de importante, representa apenas a ponta de um *iceberg*, pois, segundo a ABERT/SET, o impacto da escolha representa menos de 4% do custo dos aparelhos receptores, e se dá apenas sobre os equipamentos moduladores e demoduladores no conjunto de equipamentos das estações transmissoras e retransmissoras, onde segundo (CPqD, 2001), o país conta com pouco mais de 15 mil estações emissoras e retransmissoras, incluindo-se tanto as efetivamente em operação quanto as planejadas.

A terceira onda de impacto da TVD, a ser sentida após a adoção do padrão de DTT, é a necessidade de desenvolvimento de novos modelos de negócios que (i) estimulem a população a investir em equipamentos de TV de nova geração, e (ii) permitam às redes obter retorno sobre os investimentos efetuados. No Brasil existem

atualmente cerca de 56 milhões de aparelhos de televisão, cobrindo mais de 90% dos domicílios brasileiros. Mesmo os brasileiros mais humildes, que não têm poder aquisitivo para adquirir um aparelho, assistem televisão através de aparelhos disponíveis em locais de acesso público. Como consequência, a rede de distribuição de conteúdo televisivo tem a característica singular de atingir a quase totalidade da população brasileira sendo, portanto, um dos principais fatores de integração nacional. Segundo esta visão, os recursos aplicados pelas cadeias de TV e pela audiência televisiva devem ser encarados como investimentos que permitem a exploração e geração de novos negócios. A maior expectativa de retorno reside na interação. O canal de dados do MPEG-2 torna possível a agregação de elementos de interação aos programas de televisão tornando-os “interativos”. A possibilidade de interação coloca como requisitos para o aparelho receptor de televisão capacidade para processar o código que define os elementos de interação, o “código do programa de tv interativo”, e enviar o resultado da interação através de um “canal de retorno” para a estação emissora do programa interativo. A estação, conseqüentemente, também precisa estar equipada com hardware e software adequados para dar suporte aos “programas interativos”. Espera-se também que a rede difusora de TV se associe a outras redes de transporte de informações dando aos telespectadores capacidade para interagir influenciando nos programas que irá assistir. Em uma etapa mais avançada, espera-se que a rede de difusão de vídeo digital se torne uma das inúmeras redes que formam a imensa inter-rede que é a Internet. Em suma, quando a TV se tornar interativa (TVDI), espera-se que a mesma venha a associar imenso apelo e penetração com capacidade de interação instantânea com milhões de telespectadores e com uma vasta cadeia de produtores de conteúdo.

Portanto, é preciso entender a TVDI como um novo meio a ser explorado que, no mundo e no Brasil, só obterá sucesso através do desenvolvimento de novas aplicações, reorganização das cadeias de produção televisiva, geração de negócios e de transformações sociais neste nosso imenso país. Porém, a transição da TV analógica para a TV digital se dará no curto, médio e longo prazo, porque não é razoável esperar que os usuários troquem seus aparelhos receptores analógicos por digitais quando o sistema digital se tornar disponível. Prevê-se um período de transição de cerca de 15 anos onde as difusoras deverão transmitir sua programação simultaneamente nas formas analógica e digital (simulcast). Outra solução usada também para suavizar a transição é o uso de uma unidade conversora de sinais transportando vídeo digital (que serão recebidos da estação) para sinais com vídeo codificado na forma analógica, que é compatível com o sinal esperado pelos aparelhos receptores analógicos. Este conversor está sendo usualmente denominado Set-top Box (STB ou URD – Unidade de Recepção e Decodificação). O nome set-top box vem do inglês “caixa que fica sobre a TV”. Cabe ressaltar que o hardware e o software deste equipamento será totalmente incorporado ao aparelho receptor de televisão digital.

Este capítulo apresenta, do ponto de vista de profissionais de computação, uma introdução teórico-prática à TV Digital Interativa. A parte teórica é composta pelas seções 2 - Conceitos Básicos e Histórico, 3 - Arquitetura de Sistemas de TVDI, 4 - Protocolos de TVDI e 5 - Padrões de Middleware. A parte prática é apresentada na Seção 6 - Prática em Desenvolvimento de Aplicações para TVDI. A Seção 7 apresenta as conclusões do capítulo e indica próximos passos a serem seguidos por profissionais e estudantes que desejam se aprofundar por um tema atual, relevante e cheio de possibilidades.

## 2. Conceitos Básicos e Histórico

Nesta seção serão apresentados conceitos básicos sobre televisão analógica e digital. Estes conceitos serão introduzidos tomando por base uma apresentação sucinta dos principais sistemas de televisão analógicos para que se possa entender onde os melhoramentos historicamente introduzidos pelos sistemas digitais.

### 2.1 A Imagem em Movimento e o Olho Humano

Fisicamente, uma imagem é definida por ondas eletromagnéticas refletidas, refratadas ou geradas por objetos localizados em uma determinada região do espaço que são percebidas por células especiais (bastonetes e cones) localizados no olho humano. As células funcionam como pequenos aparelhos receptores de ondas eletromagnéticas com frequência variando entre  $4,3 \times 10^{14}$ Hz (vermelho) e  $7,5 \times 10^{14}$ Hz (violeta). O processo de recepção de imagens pelo olho é extremamente complexo, mas pode ser modelado por um parâmetro denominado *persistência da visão*, que é uma propriedade que o sistema de recepção visual possui de reter por um certo tempo a impressão de uma imagem que já passou. Com isto o olho humano age como um integrador e permite que uma sucessão de imagens paradas possa ser compreendida pelo cérebro como uma imagem contínua. Para se produzir as imagens em movimento é necessário se justapor uma sequência de imagens separadas por um determinado intervalo de tempo que é função das propriedades de persistência da visão. Na produção de um filme, por exemplo, a filmadora nada mais é que uma câmara fotográfica que tira fotografias sucessivas de uma cena, em uma frequência conhecida como frequência de amostragem, armazenando estas fotografias em um filme.

Para que o olho humano perceba a sensação de movimento nos objetos contidos em uma cena são necessárias pelo menos 15 imagens por segundo, porém a percepção só atinge uma qualidade ideal com 30 imagens (ou quadros) por segundo.

### 2.2 Imagem Televisiva e Padrões de TV Analógica

Nos monitores de televisão, as imagens são produzidas por um processo que se baseia no bombeamento de energia para átomos de fósforo que, ao serem excitados, emitem fótons (luz) e voltam ao estado de equilíbrio. Deste modo, as imagens são acesas na tela do monitor, que em seguidas apagam quando o fósforo volta ao estado equilíbrio. Sendo acesas novamente em um processo contínuo. Percebe-se então que a tela do monitor cintila (acende e apaga) com uma determinada frequência. Para que o olho não perceba a cintilação é necessária uma frequência de pelo menos 50 refrescamentos por segundo.

Existem vários padrões de sistemas de televisão, como o M, N, BGH, entre outros. Esses padrões especificam as características da imagem, como número de pontos por linha, número de linhas por quadro, número de quadros por segundo, entre outras informações. No início dos anos 50, as cores passaram a ser aproveitadas, com os Estados Unidos criando o padrão conhecido na prática como NTSC (*National Television System Committee*), que na verdade representa o nome do comitê que foi criado pela indústria para elaborar a proposta de televisão a cores ao FCC. Esse padrão de cores foi aceito pelo Japão e rejeitado pela Europa, que em 1966 adotou o SECAM (*Séquentiel Couleur Avec Memoire*) como padrão. Os europeus possuem ainda outro padrão de cores, que foi o adotado pelo Brasil, o PAL (*Phase Alternation Line*).



No sistema NTSC, o vídeo é formado por 30 imagens (ou quadros) por segundo com 525 linhas. Este padrão é utilizado nos Estados Unidos, Canadá, Groenlândia, México, Cuba, Panamá, Japão, Filipinas, Porto Rico, e partes da América do Sul. Os quadros são divididos em campo ímpar (conjunto das linhas ímpares do quadro) e campo par. Os campos dos pares são transmitidos em seqüência. Assim, no NTSC para que se atinja uma taxa de 30 quadros de vídeo são transmitidos e acesos na tela das TVs, de forma alternada e entrelaçada, 60 campos por segundo de vídeo, evitando a percepção da cintilação.

Nos sistemas SECAM e PAL as imagens são formadas por 625 linhas e são transmitidos 25 quadros por segundo (50 campos alternados e entrelaçados). O sistema SECAM é usado na França e na maior parte dos países vizinhos ou dentro da antiga União Soviética. O sistema PAL é usado por quase toda a Europa Ocidental (com exceção da França). As 100 linhas a mais dos sistemas PAL e SECAM acrescentam significativamente, em termos de detalhe e clareza, à imagem de vídeo, mas quando comparamos os 50 campos por segundo com os 60 campos do sistema do NTSC, podemos notar uma ligeira trepidação (flicker) na imagem daqueles sistemas. A televisão analógica que assistimos nos dias de hoje é bastante semelhante a que foi inventada há quase cinco décadas.

### **2.3 A Televisiva de Alta Definição**

A evolução na direção da televisão de alta definição se iniciou com uma pesquisa realizada pelos japoneses, que desenvolveram um sistema analógico que oferecia um número de linhas maior que o dobro (de 525 para 1125) e permitia a transmissão de som de alta qualidade. Os resultados da pesquisa japonesa foram apresentados no início da década de 80. Os japoneses, na verdade, direcionaram sua pesquisa buscando primeiro um padrão de produção em estúdio, para daí derivar um padrão para transmissão, tanto via satélite, como em sistema de cabo e difusão terrestre. Assim, o primeiro sistema de televisão de alta definição a entrar em operação em escala comercial foi o sistema japonês MUSE (Multiple Sub-Nyquist Sampling Encoding). No MUSE, o sinal de alta definição, com mais de 1 Gbit/s de informação, é codificado em um canal com 27 MHz de largura de faixa, compatível com os canais disponíveis em satélites.

Os europeus também iniciaram suas pesquisas na direção da melhoria da qualidade da televisão usando tecnologia analógica. Em 1986, se iniciou o projeto da Comunidade Européia “Eureka”, financiando o desenvolvimento do sistema MAC – Multiplexed Analog Components. O padrão MAC se baseava na digitalização e compressão independente de cada componente de croma e utilizava algumas técnicas analógicas para a composição final do sinal. Uma versão com maior resolução do sistema básico foi denominada HD-MAC (*High Definition - MAC*). O sistema, entretanto, não foi bem sucedido comercialmente e, em 1993, a comunidade européia voltou seus esforços de pesquisa na direção de um padrão totalmente digital.

### **2.4 O Surgimento da TV Digital**

Nos Estados Unidos, em 1987, foram iniciados estudos com o objetivo de desenvolver novos conceitos no serviço de televisão. Foi então criado o ACATS (Advisory Committee on Advanced Television). No início de seus trabalhos, o comitê decidiu

desenvolver um sistema totalmente digital, que foi denominado DTV - Digital Television. Foi então criado um laboratório, o ATTC - Advanced Television Test Center, que, entre 1990 e 1992, testou seis propostas. Nos testes realizados, nenhuma das propostas satisfaz a todos os requisitos. Em 1993, sete empresas e instituições participantes dos testes (AT&T, GI, MIT, Phillips, Sarnoff, Thomson e Zenith) se uniram formando a “Grande Aliança” para desenvolver um padrão juntas. Numa decisão arrojada foi adotado como padrão para compressão do vídeo o padrão MPEG-2.

No final de 1993, os europeus também decidiram desenvolver um padrão totalmente digital e adotaram o padrão MPEG. Criou-se então o consórcio DVB - *Digital Video Broadcasting*. A versão DVB para a radiodifusão terrestre (DVB-T) entrou em operação em 1998, na Inglaterra. Em 1995, o ATSC - Advanced Television System Committee recomenda à FCC adotar o sistema da Grande Aliança como o padrão para a DTV norte-americana. O padrão americano, que ficou conhecido como ATSC, entrou em operação também em 1995. Só em 1997 os Japoneses decidiram desenvolver um padrão totalmente digital. O sistema Japonês denominado ISDB - Integrated Services Digital Broadcasting assemelha-se ao europeu e entrou em operação com transmissão via satélite em 2000.

A título de ilustração, a Figura 1 mostra o aspecto de uma imagem NTSC com 525 linhas de resolução e razão de aspecto de 4:3 ao lado do aspecto de uma imagem HDTV com 1080 linhas e razão 16:9 (semelhante a usada nas telas de cinema).

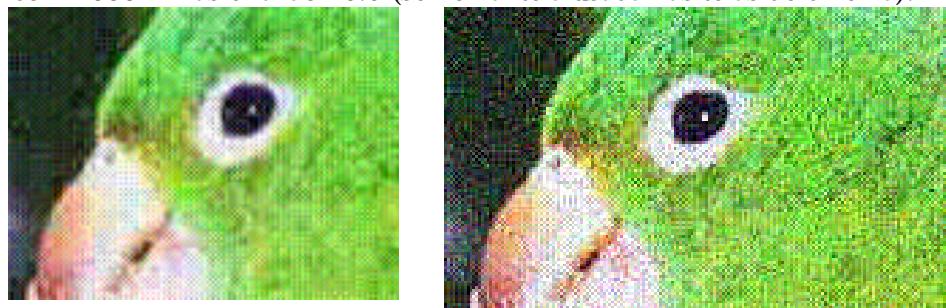


Figura 1. Comparação entre Aspectos de imagem NTSC e HDTV

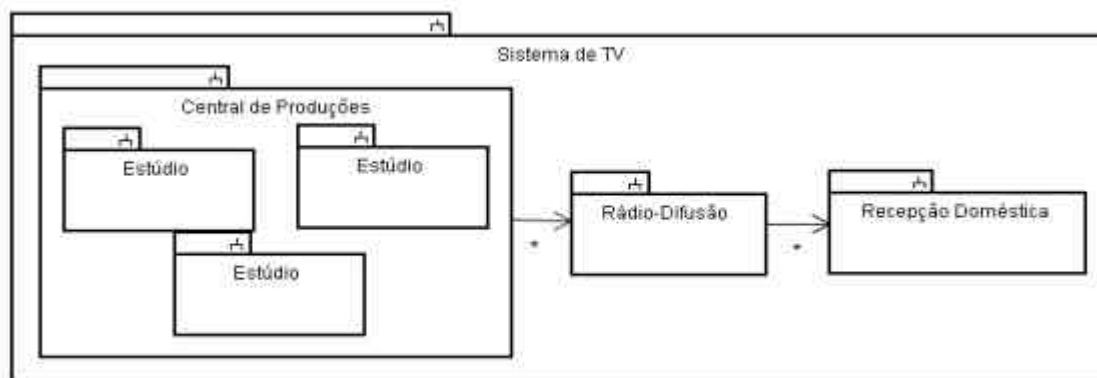
### 3. Arquitetura de Sistemas de TVDI

Esta seção apresenta os diversos componentes que fazem parte da arquitetura de sistemas de TVDI. Visando situar historicamente o caráter evolucionário dos sistemas, a arquitetura será apresentada através do refinamento de uma série de diagramas arquiteturais, que se inicia com a apresentação do modelo atual, que contém um misto de componentes analógicos e digitais, e conclui com uma definição arquitetural geral de sistemas de TV Digital Interativa e seus diversos componentes.

#### 3.1. Arquitetura de Software e Sistemas

Arquitetura de um software, ou de um sistema qualquer, é a estrutura descritiva do elemento sob análise, composta pelos componentes do sistema, pelas propriedades externamente visíveis destes componentes, e pelo relacionamento entre estes componentes. Os componentes podem ser objetos, processos, bases de dados, subsistemas, etc, o que depende do ponto de vista sob o qual estão sendo descritos. Numa

arquitetura são omitidas, as informações que não são pertinentes às interações entre os componentes. Adotaremos neste capítulo uma visão simplificada de descrição arquitetural, baseada no uso de diagramas UML para representar subsistemas e classes que compõem a arquitetura de sistemas de TVDI, do ponto de vista de profissionais de computação. A Figura 2 apresenta um modelo de análise arquitetural de alto nível de um sistema de TV, composto por vários subsistemas.



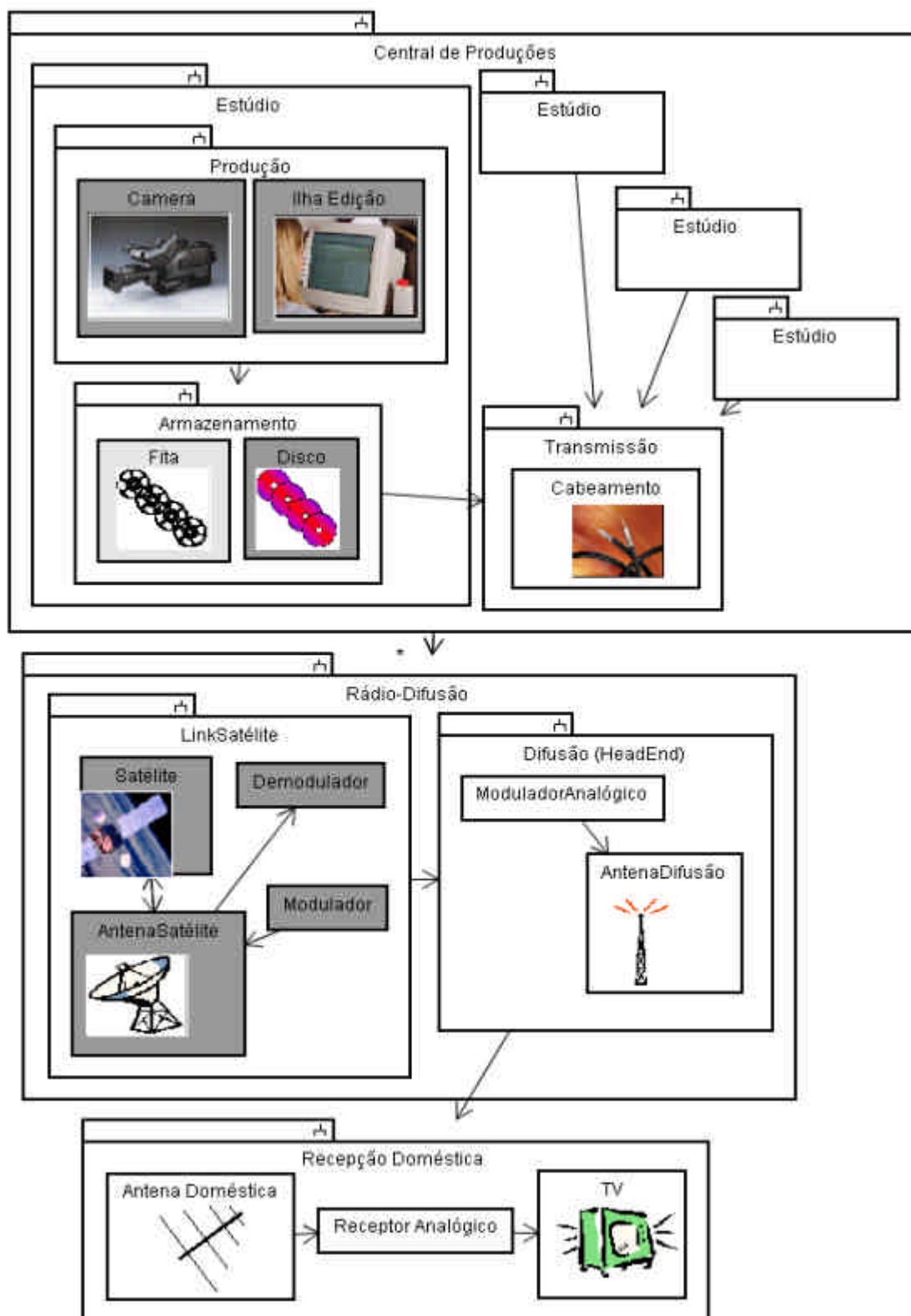
**Figura 2. Arquitetura de Alto Nível de um Sistema de TV.**

A Figura 2 descreve a composição de um sistema de TV, como formado por três subsistemas principais: Central de Produções, Rádio-Difusão e Recepção Doméstica. A Central de Produções é responsável por gerar programas de TV que serão veiculados através do sistema de Rádio-Difusão, e é composta por vários Estúdios. Existe uma relação unidirecional entre múltiplas Centrais de Produção e subsistemas de Rádio-Difusão. O subsistema de Rádio-Difusão é responsável por receber o sinal a ser difundido, realizar a devida modulação conforme o meio usado e transmitir o sinal para a Recepção Doméstica (audiência). Existe também uma relação unidirecional entre múltiplos sistemas de Rádio-Difusão e múltiplos sistemas de Recepção Doméstica.

### **3.2. Detalhes Arquiteturais de um Sistema de TV Atual**

A Figura 3 refina a análise dos elementos típicos que formam um sistema de TV atual, mostrando que estes são formados por uma mistura de componentes digitais e analógicos. Para simplificar a arquitetura serão omitidos aspectos como iluminação, cenários, etc, que também variam na transição do analógico para o digital, mas que estão fora do escopo nesta visão.

Cada Estúdio da Central de Produção é composto pelos subsistemas de Produção/Pós-produção e Armazenamento. A Produção é responsável pela gravação de cenas, edição e criação dos programas. O Armazenamento é responsável por armazenar e recuperar o acervo de cenas e programas criados pelo Estúdio. A Transmissão é responsável pela transferência de cenas e programas intra-Estúdio e inter-Estúdio, bem como a transmissão para a Rádio-Difusão.



**Figura 3. Arquitetura de Sistemas de TV Atuais.**

A transmissão dos programas do Estúdio para Rádio-difusão pode se dar de vários meios, sendo o mais comum o uso de links de satélites e micro-ondas. A difusão para Recepção Doméstica pode ser dar através de três meios: terrestre, satélite e cabo, sendo a transmissão terrestre aquela realizada em áreas urbanas através do posicionamento de antenas em regiões elevadas da cidade.

O subsistema de Recepção Doméstica recebe o sinal difundido através do conjunto Antena + Receptor Analógico (em geral embutido no aparelho de TV) e envia este sinal para o Monitor de TV, que exhibe o programa para a audiência.

Atualmente várias partes dos subsistemas de TV atual já empregam codificação digital. Os elementos da Figura 3 que possuem cor mais escura são em geral os módulos que já empregam alguma forma de digitalização. Por exemplo, formatos de vídeo como o MiniDV, usados em produção de TV, empregam codificação digital de sinais. Alguns aparelhos de TV comuns empregam algum tipo de tratamento digital de imagens, como o PIP (picture in picture). Por fim, praticamente todo sistema de transmissão via satélite, inclusive a recepção doméstica de sinais de satélite, usa modulação digital.

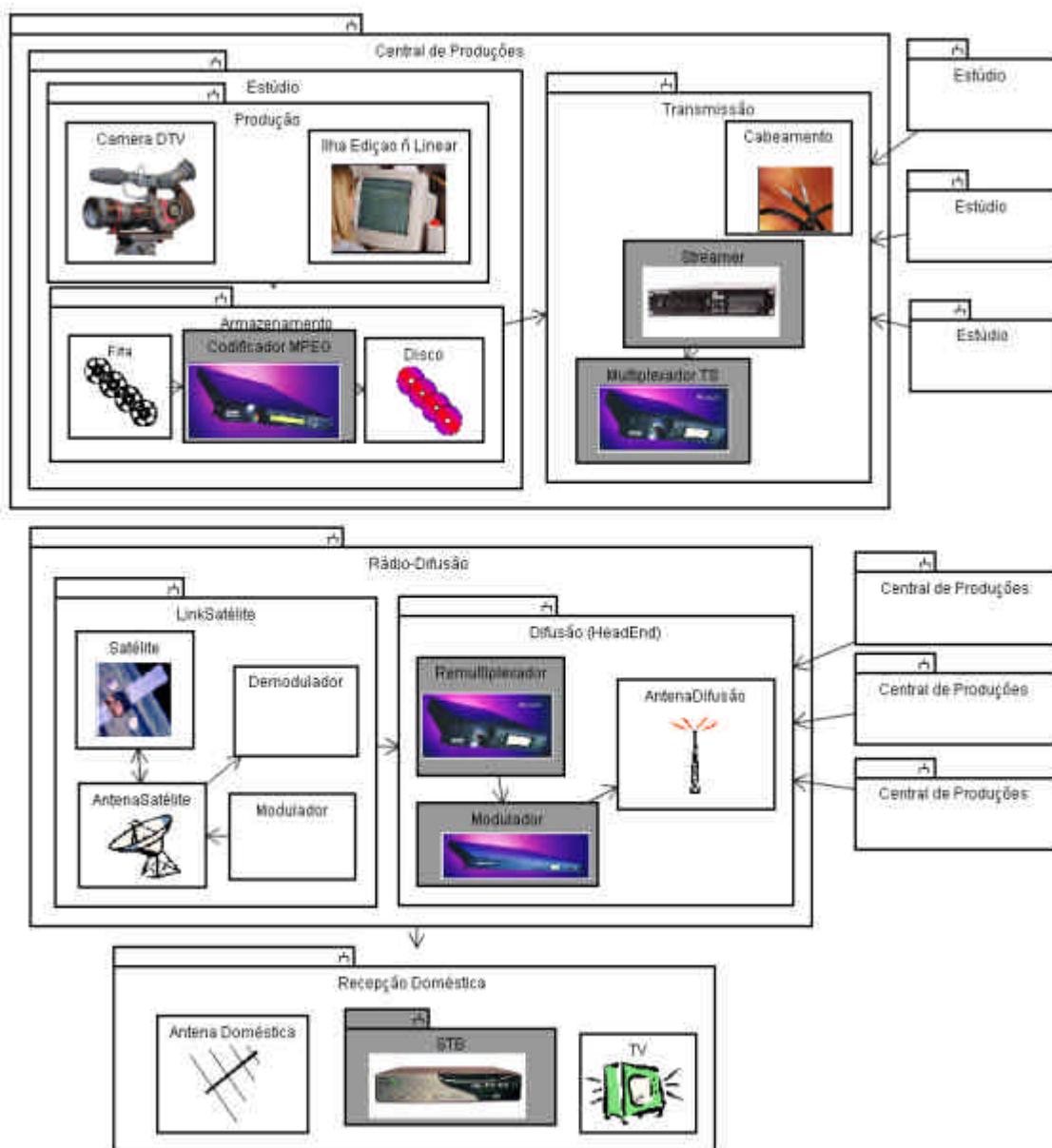
### **3.3. Arquitetura Básica de um Sistema de Televisão Digital**

O único fator considerado fixo dentre as opções atuais de plataformas de TV digital é a adoção do padrão MPEG-2. Sistemas, o que traz em seu bojo um imenso conjunto de soluções que definem a arquitetura básica de sistemas de TV Digital, cujos elementos são apresentados na Figura 4. A adoção do padrão MPEG-2 confere grande capacidade de interoperabilidade entre os diversos subsistemas de uma plataforma de TV digital. Os elementos com cores escuras na Figura 4 representam os principais módulos adicionais acrescentados pela arquitetura MPEG-2 ao cenário de plataformas de TV digital.

**Impacto da TV Digital sobre o Estúdio.** No Estúdio de TV Digital a Câmera tem uma maior resolução de linhas e colunas. Destaca-se também a presença do Codificador MPEG (Tektronix, 2002), responsável principalmente por aplicar técnicas de compressão temporal e espacial de imagens a um sinal de vídeo digital, originalmente em formato quadro a quadro, a fim de produzir um fluxo de streams elementares de A/V, que no caso de codificação digital para o vídeo obtém taxas de compressão que chegam a 1 bit comprimido para cada 70 bits da codificação sem compressão. Streams de A/V comprimido são facilmente armazenáveis em um meio permanente como um DVD (Digital Vídeo Disk). No estúdio destaca-se também que a Ilha de Edição passa a ser Não Linear, pois as cenas não são mais armazenadas em fitas (acesso linear), mas sim em dispositivos de acesso direto (não linear), como discos rígidos e DVDs. Para distribuição dos programas para a Central os arquivos podem ser transmitidos através de uma rede local de computadores, reduzindo a quantidade de cabeamento e o transporte de fitas ou mesmo DVDs. Para transmissões em espaço urbano ou à longa distância o estúdio pode também dispor de um Streamer, detalhado mais abaixo.

**Impacto da TV Digital sobre a Central de Produções.** Na Central de Produções (agora chamada de Provedora de Serviços) são introduzidos dois elementos: Streamer e Multiplexador. O Streamer (também chamado de Empacotador TS) é responsável por transmitir e receber streams (fluxos) de transporte MPEG-2 (MPEG-2-TS), a partir da segmentação de streams elementares de A/V. O Streamer facilita a geração de fluxos que podem ser transmitidos através de redes de computadores de média

e longa distância com grande qualidade e custo reduzido (Broadcast Papers, 2004c). O uso de Streamers reduz a necessidade de links de satélite pela Central de Produções. Combinado com o uso do Multiplexador, o Streamer aumenta fortemente a capacidade de integração da Central com uma maior quantidade de estúdios, inclusive externos, o que permite uma maior oferta de programas.



**Figura 4. Arquitetura de Sistemas de TV Digital.**

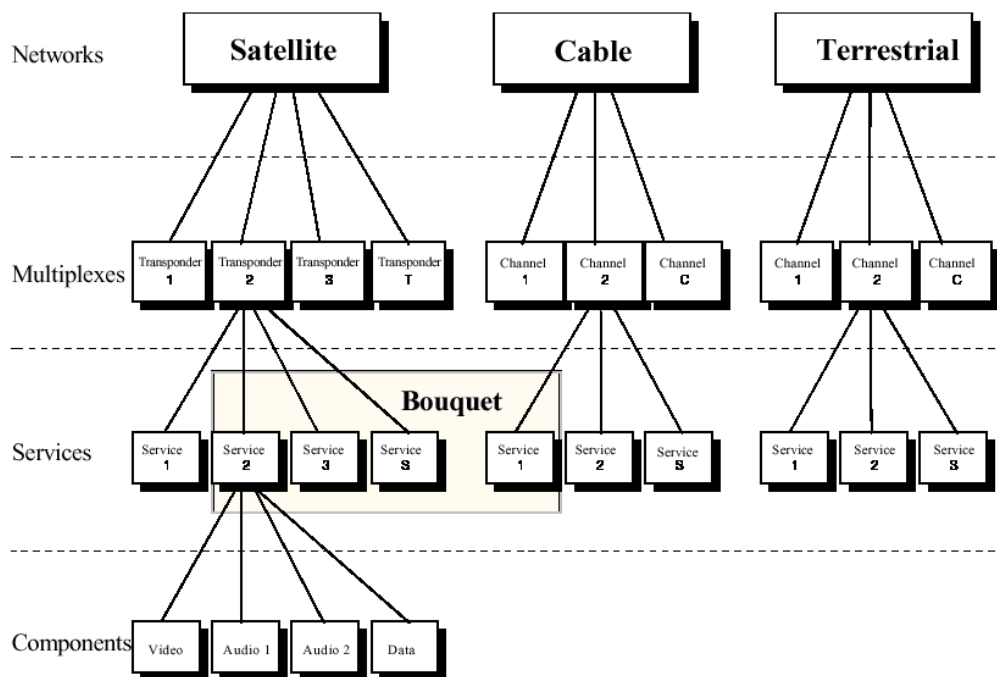
**Impacto da TV Digital sobre a Rádio-Difusão.** A aplicação de técnicas de compressão aos sinais televisivos permite que, no mesmo espaço de banda passante terrestre (canal UHF/VHF) por onde hoje trafega um sinal analógico, seja possível transmitir pelo menos 4 programas com qualidade superior. Este espaço adicional pode ser usado para transmitir programas e dados adicionais. Deste modo, no subsistema de

Rádio-difusão (ou Rede de Difusão) é introduzido um módulo de multiplexação, normalmente chamado de Remultiplexador (Broadcast Papers, 2004c). O Remultiplexador é responsável por fazer a multiplexação entre os vários TS gerados por uma ou mais centrais de produção. Além de permitir a transmissão de mais programas em um único espaço de banda (canal), a Remultiplexação pode ter outras funções, dentre as quais se destacam: (i) renomear os identificadores dos programas e fluxos elementares enviados pelas centrais para evitar colisão de identificadores, (ii) eliminar, substituir ou inserir programas e fluxos de dados que serão veiculados, (iii) inserir informações gerais sobre a programação dos vários canais veiculados e (iv) proteger programas cujo conteúdo é consumido através de pagamento (pay-per-view). A introdução do remultiplexador permite ao difusor operar uma maior quantidade de Centrais de Produção (Provedores de Serviços), o que aumenta a oferta de conteúdo e canais.

**Impacto da TV Digital sobre a Recepção Doméstica.** O subsistema de recepção doméstica de TV Digital necessita de um STB que seja capaz de receber, demodular, decodificar e remodular o sinal televisivo que será apresentado pela TV. O STB é um equipamento digital com capacidade de processamento de sinais de áudio e vídeo, e eventualmente capacidade de execução de programas. O sinal de A/V gerado na saída do STB pode ser compatível com televisores analógicos. Detalhes sobre a arquitetura do STB são apresentados na Seção 3.5.

### 3.4. Novos Conceitos Introduzidos do Modelo de TV Digital

A arquitetura geral de TVD apresentada acima introduz uma série de conceitos que apóiam novos modelos de negócios, fundamentais para o sucesso da mudança tecnológica para a TV Digital. A Figura 5 apresenta as principais relações entre estes conceitos, conforme definidos no modelo DVB – Digital Vídeo Broadcasting (DVB, 2004).



**Figura 5. Modelo de Entrega de Serviços na Plataforma DVB (DVB, 2004).**



No modelo de consumo de serviços de TV digital o elemento atômico de produção de mídia é chamado de evento. Um evento é um agrupamento de streams elementares (A/V/D) com um tempo definido de início e fim, como, por exemplo, a primeira parte de uma novela ou o primeiro tempo de uma partida de futebol. Um programa é uma concatenação de um ou mais eventos produzidos por um estúdio, como um capítulo de novela ou um show. Um serviço é uma sequência de programas (programação) controlada por um difusor, que tem por objetivo atingir uma determinada audiência, e que é veiculado em uma determinada faixa de horários. O serviço é a principal **unidade de produção e consumo** na TV digital. Uma Central de Produções pode compor um conjunto de serviços (programação) produzidos por vários estúdios, formando o que se chama de Bouquet. O Bouquet é, portanto, a **unidade de distribuição das programações** de uma Central de Produções. Os bouquets produzidos pelas centrais são remultiplexados pelas redes de rádio-difusão, que canalizam um ou mais serviços através da alocação dos sinais em uma faixa do espectro eletromagnético.

### 3.5. Arquitetura de Sistemas de TV Digital Pseudo-Interativa

A capacidade de interatividade da TV digital se deve à presença de três elementos: Gerador de Carrossel, Multiplexador e STB Interativo (Broadcast Papers, 2004d). O Gerador de Carrossel é capaz de transformar um conjunto de arquivos de dados em um fluxo elementar, empregado um esquema de transmissão cíclica de dados. O Multiplexador é capaz de fundir um ou mais fluxos de dados aos fluxos de áudio e vídeo que compõem os eventos e programas, os quais por sua vez compõem os serviços consumidos pela audiência. O STB Interativo possui capacidade de processamento computacional, sendo capaz de interpretar computacionalmente os fluxos de dados multiplexados. Deste modo o STB executa uma aplicação que exhibe na TV uma interface com o usuário. Isto permite à audiência interagir com o programa de TV através do teclado ou controle remoto. Ao entregar à audiência um fluxo de dados localmente computável, a TV Digital se torna interativa.

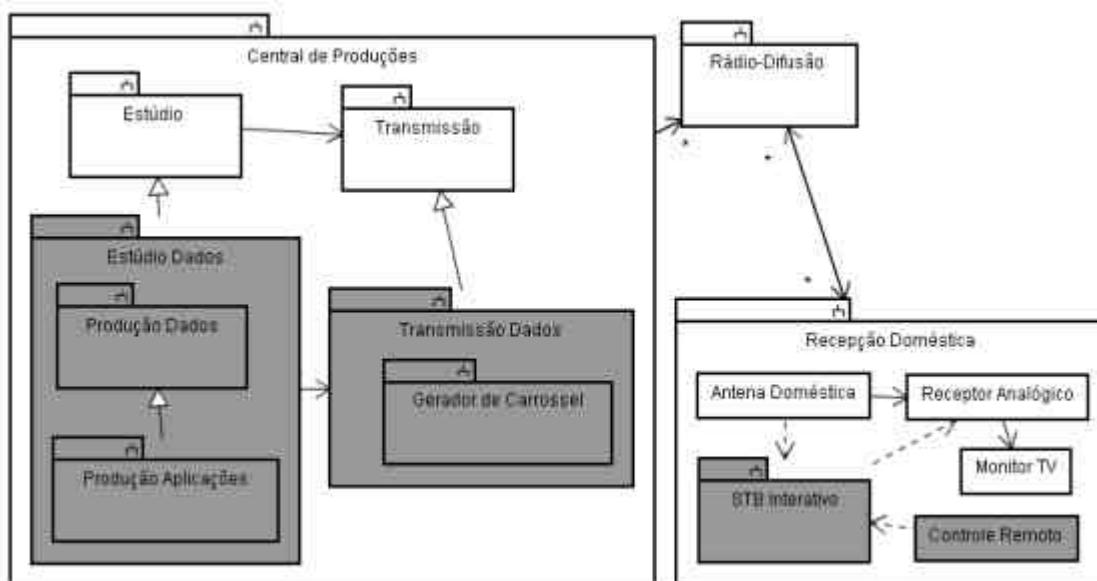
Caso o resultado da interação entre o usuário e o STB fique restrito ao subsistema de Recepção Doméstica, o modelo é chamado de TV Digital Pseudo-Interativa (Enhanced DTV).

Uma expansão do modelo de pseudo-interatividade permite que o STB envie e receba dados adicionais através de um canal de interação (retorno), estabelecido via modem, por exemplo. Neste caso o resultado da interação com o usuário pode ser avaliado em tempo quase real por um provedor de serviços vinculado à rede de TVD, resultado no modelo chamado de TV Digital Interativa (Interactive DTV).

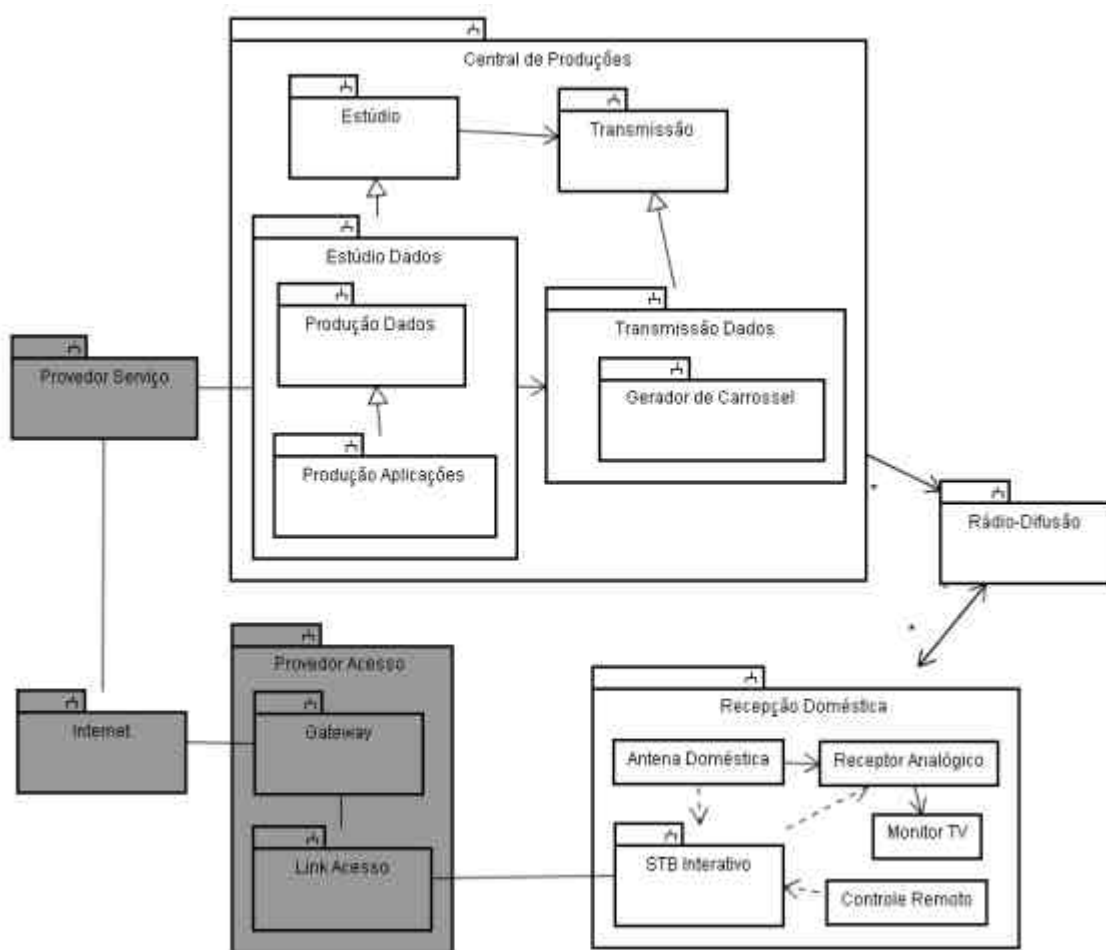
A Figura 6 apresenta os elementos que definem a arquitetura geral de sistemas de TV Digital Pseudo-Interativa (Enhanced DTV). Na EDTV existe um estúdio especializado, chamado Estúdio Dados, que realiza dois novos processos: (1) Produção Dados, o que envolve geração de vídeo-texto e páginas HTML e (2) Produção Aplicações, que envolve desenvolvimento de software. O Estúdio Dados disponibiliza seus resultados para o subsistema de Transmissão de Dados, que contém um Gerador de Carrossel, responsável por transmitir um fluxo elementar de dados (e aplicações) a partir dos dados do estúdio. Este fluxo D (Dados) é sincronizado com os fluxos A/V. Os fluxos A/V/D são recebidos pelo STB (Pseudo-)Interativo. Os dados e programas são



interpretados e produzem interação local, que se dá principalmente através do Controle Remoto.



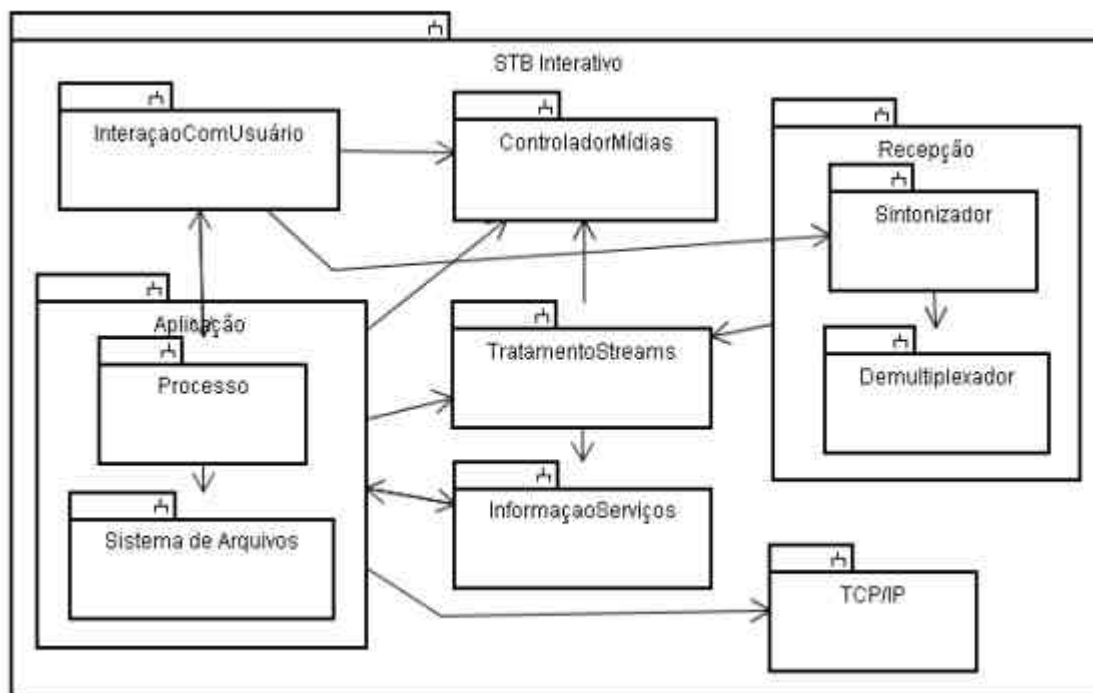
**Figura 6. Arquitetura Geral de Sistemas de TV Digital Pseudo-Interativa.**



**Figura 7. Arquitetura Geral de Sistemas de TV Digital Interativa.**

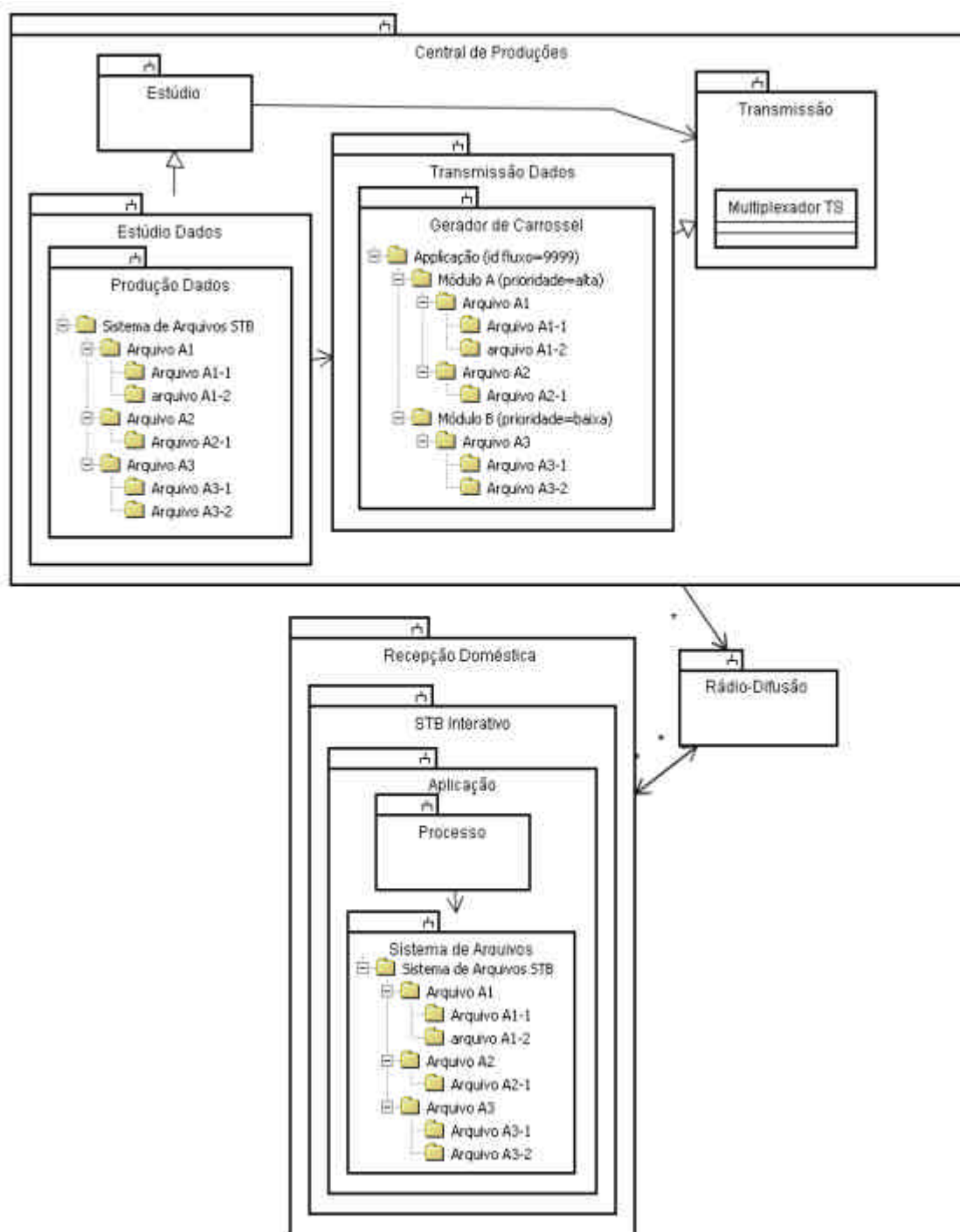
A Figura 7 apresenta a arquitetura geral de sistemas de TV Digital Interativa, com indicação dos elementos adicionais relativos à TV Pseudo-Interativa. Na arquitetura de TVDI destaca-se a presença de um Provedor de Acesso, ao qual o STB Interativo conecta-se para enviar e/ou receber dados resultados do processo de interação local. Esta conexão pode ser feita através de um modem ou outro meio alternativo. O Provedor de Acesso contém um Gateway para acesso à Internet, e desta forma o STB Interativo pode ter acesso a dados e serviços da Internet. A fim desenvolver um modelo de negócios que produza resultados econômicos mais relevantes na TVDI, o STB é normalmente direcionado a interagir com um Provedor de Serviço específico, que oferta um produto ou serviço fortemente relacionado ao conteúdo de A/V/D do evento televisivo produzido na Central. As seções que se seguem apresentam maiores detalhes da arquitetura dos seguintes subsistemas: (i) STB Interativo e (ii) Gerador de Carrossel. Os componentes Multiplexador e Demultiplexador apóiam praticamente toda a flexibilidade de operação da TV Digital e são fundamentais para compreensão da arquitetura de sistemas de TVDI. O funcionamento e a arquitetura destes componentes já foi descrito na Seção 3.2.

### 3.5. Arquitetura do STB Interativo



**Figura Arquitetura de um STB Interativo.**

O STB Interativo é o elemento que fica na extremidade da arquitetura de TVDI, e sua arquitetura é esboçada na Figura. O STB Interativo é um pequeno computador dedicado à tarefa de processar fluxos de áudio, vídeo e dados, através da sintonização e demultiplexação do sinal de TVDI. O STB seleciona uma série de streams relacionadas a um serviço. As streams pertencem a duas categorias: (i) relativas a áudio e vídeo - A/V e (ii) dados sobre serviços. As streams A/V são diretamente enviadas para o Controlador de Mídias, que as exibirá conforme controles ajustados pelo usuário ou pelas aplicações executadas no STB. Os dados sobre serviço são remetidos ao subsistema de Informação Serviços (SI - Service Information). Estes fluxos contêm informações detalhadas sobre todas as outras streams A/V/D disponíveis para o STB. A partir das Informações sobre Serviços é possível exibir os dados e executar as aplicações que foram produzidas nos estúdios. A execução de aplicações segue um modelo computacional padronizado, que contém áreas de Processo e Sistema de Arquivos.



**Figura 9. Arquitetura do Carrossel.**

### 3.5. Arquitetura do Gerador de Carrossel

A Figura 9 apresenta um esboço arquitetural de como funciona o Carrossel. A intenção primária do modelo de Carrossel é permitir a instalação dinâmica, no STB, de uma cópia de um sistema de arquivos produzido no Estúdio de Dados. Este Sistema de Arquivos

persiste no STB apenas enquanto o serviço está sintonizado. O Gerador de Carrossel é o elemento responsável por gerar um stream elementar de dados que, recebida pelo STB, produz este efeito. O stream Carrossel segue o protocolo DSM-CC, sub-protocolo DSM-CC Object, do padrão MPEG-2. Outras variações do protocolo DSM-CC permitem a transmissão de outros tipos de dados, como fluxos IP e atualização de firmware, que estão descritas em detalhes em (ETSI, 2003b). O nome carrossel se deve ao fato de que os fluxos de dados que geram o sistema de arquivos precisam ser re-transmitidos ciclicamente, a fim de que seja possível a um STB que sintonizou o serviço receber este sistema de arquivos, mesmo após o início da difusão.

Os arquivos a serem transmitidos pela Central de Produções - chamada de Server User no protocolo DSM-CC - fazem parte de uma aplicação, que estará sendo transmitida em um fluxo individualmente identificado. Os arquivos são agrupados em módulos, aos quais estão associadas prioridades de re-transmissão. O Gerador de Carrossel gera continuamente um stream contendo os módulos a transmitir, sendo que os módulos de maior prioridade são transmitidos com maior frequência. A Geração do Carrossel pode ser feita em tempo real ou off-line. Se feita off-line um arquivo é produzido no Gerador de Carrossel, e continuamente enviado. Qualquer que seja a alternativa, o fluxo DSM-CC Object é sincronizado e multiplexado com os outros fluxos que fazem parte do programa a transmitir. O subsistema de Rádio Difusão - chamado de Network, no protocolo DSM-CC - não sofre alterações. Ao serem recebidos pelo STB interativo - chamado de Client User - os arquivos do carrossel podem ter várias finalidades, como apresentar dados específicos para serem apresentados por um programa de EPG - Guia de Programação Eletrônica, conter informações adicionais sobre uma determinada propaganda veiculada, apresentar um teletexto, bem como enviar uma aplicação a ser executada no STB, como um Xlet.

#### **4. Padrões para TVDI**

Uma questão fundamental para o sucesso de um sistema de televisão digital é a adoção e aceitação de padrões abertos para os vários componentes do sistema. É fato que o negócio televisão envolve o trabalho de inúmeros profissionais de diferentes organizações na produção do conteúdo televisivo. Este conteúdo deve ser transmitido, no caso do Brasil, por milhares de estações transmissoras e retransmissoras para milhões de aparelhos receptores. Os aparelhos receptores serão produzidos por inúmeros fabricantes no Brasil ou no mundo. Como já apresentado, os aparelhos receptores digitais incluem em seu hardware, de forma embarcada, o STB. Como os dados podem ser códigos executáveis, o STB inclui também um conjunto de componentes de software que constituem seu sistema operacional e o ambiente de execução dos programas de televisão interativos. Os componentes de software que formam o sistema operacional são normalmente dependentes do hardware, com vários de seus componentes desenvolvidos sob medida para um determinado hardware. Assim, é esperado que o conjunto dos aparelhos receptores inclua equipamentos elaborados por fabricantes distintos executando sistemas operacionais distintos. Porém, neste cenário não tem sentido esperar que os produtores de conteúdo televisivo, no caso programas de televisão interativos, codifiquem versões de seus programas para os diferentes sistemas operacionais e hardwares dos aparelhos receptores. Torna-se então fundamental então a padronização também de uma camada de adaptação

de software que ofereça para os desenvolvedores um ambiente de programação padronizado.

Feitas estas considerações, a Figura 10 apresenta uma série de escolhas tecnológicas que precisam ser consideradas quando da definição de um sistema de televisão digital.

As escolhas tecnológicas para os componentes do sistema são questões estratégicas definidas por órgãos de padronização de países ou blocos econômicos, levando em consideração aspectos socioeconômicos.

#### **4.1. Padrões Mundiais de TVDI**

Um sistema de televisão digital interativa deve adotar e integrar um conjunto de diferentes tecnologias de hardware e software para implementar suas funcionalidades. Conjuntamente, estas tecnologias permitem que um sinal eletromagnético, que transporta fluxos elementares de áudio, vídeo, dados e aplicações, possa ser transmitido para o STB e, então, que estes fluxos sejam recebidos, processados e apresentados aos usuários.

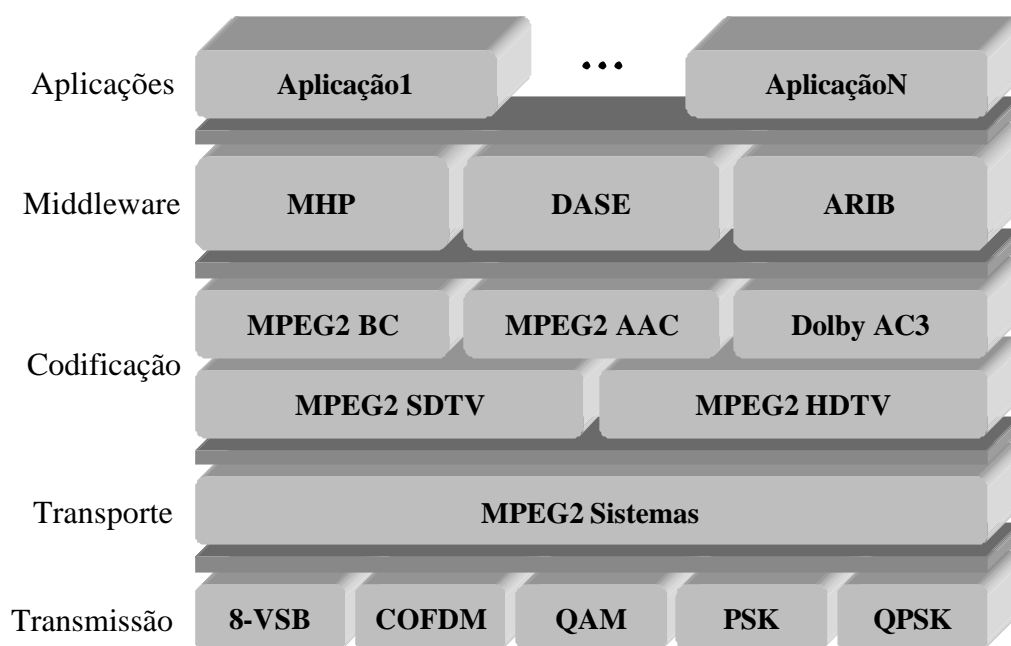
Considerando a diversidade de soluções tecnológicas que podem ser adotadas para implementar um sistema de televisão digital interativa, diversos órgãos de padronização concentraram esforços na especificação de padrões. Como resultado destes esforços, atualmente, existem três padrões mundiais de sistema de televisão digital interativa reconhecidos:

?? DVB - *Digital Video Broadcasting*

?? ATSC - *Advanced Television Systems Committee*

?? ISDB - *Integrated Services Digital Broadcasting*

Estes sistemas (DVB, ATSC e ISDB) adotam diferentes padrões para modulação do sinal de difusão; transporte de fluxos elementares de áudio, vídeo, dados e aplicações; codificação e qualidade de áudio e vídeo; e serviços de *middleware*. A Figura 10 apresenta uma visão arquitetural em camadas de um sistema de televisão digital, identificando as diversas opções de padrões de transmissão, transporte, codificação e *middleware* que podem ser adotados em cada camada.



**Figura 10. Opções de padrões para um sistema de televisão digital interativa.**

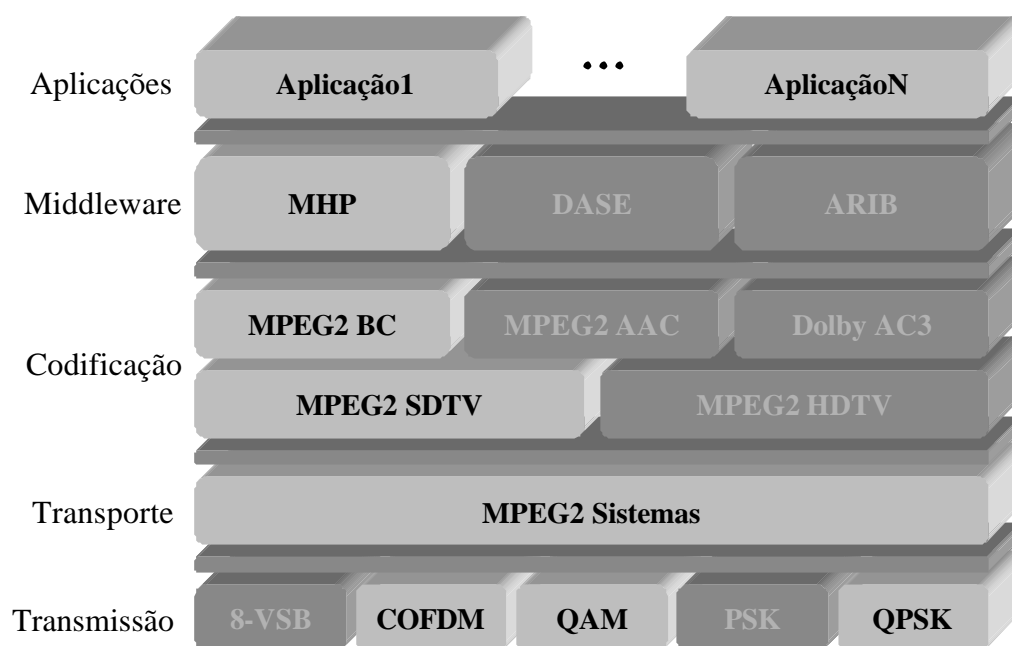
A seguir apresentamos uma breve descrição da arquitetura dos principais padrões de sistema de televisão digital interativa, identificando os componentes básicos adotados nestes padrões. A organização do Middleware dos padrões de TVDI será apresentada na seção 5.

#### **4.2.1. DVB**

O projeto DVB - *Digital Video Broadcasting* (DVB, 2004) é um consórcio iniciado em setembro de 1993 e composto por mais de 300 membros, incluindo fabricantes de equipamentos, operadoras de redes, desenvolvedores de software e órgãos de regulamentação de 35 países. O objetivo do projeto DVB é especificar uma família de padrões mundiais para sistemas de televisão digital interativa, incluindo a transmissão do sinal e serviços de dados associados.

A família de padrões especificada pelo projeto DVB caracteriza o padrão de sistema de televisão digital também denominado DVB, que é conhecido como o padrão europeu de televisão digital. O padrão DVB é adotado nos países da União Européia e em outros países como Austrália, Nova Zelândia, Malásia, Hong Kong, Singapura, Índia e África do Sul. A Inglaterra é o país onde a adoção do padrão DVB está mais consolidada, pois já possui mais de um milhão de receptores digitais instalados.

O padrão DVB é formado por um conjunto de documentos que definem os diversos padrões adotados, incluindo aqueles relacionados à transmissão, transporte, codificação e *middleware*. A Figura 12 ilustra esquematicamente a arquitetura do sistema DVB, identificando os principais componentes padronizados nas diversas camadas.



**Figura 11. Arquitetura DVB**

O padrão DVB permite diversas configurações para a camada de transmissão, cada configuração apresentando uma diferente relação capacidade/robustez. Os principais padrões de transmissão adotados pelo DVB são: DVB-T: (transmissão terrestre por radiodifusão); DVB-C (transmissão via cabo); DVB-S (transmissão via satélite); DVB-MC (transmissão via microondas operando em frequências de até 10GHz); e DVB-MS (transmissão via microondas operando em frequências acima de 10GHz).

Os padrões DVB-T, DVB-C, DVB-S, DVB-MC e DVB-MS adotam diferentes esquemas de modulação. O DVB-T (ETSI, 2001) pode operar em canais de 6, 7 ou 8 MHz e adota a modulação COFDM (*Coded Orthogonal Frequency Division Multiplexing*), cuja taxa de transmissão pode variar entre 5 e 31,7 Mbps, dependendo dos parâmetros utilizados na codificação e modulação do sinal. O COFDM será detalhado na Seção 4.2.

O DVB-T suporta seis modos de transmissão com resoluções que variam de 1080 à 240 linhas, podendo ser usado para sistemas de alta definição (HDTV – *High Definition Television*) e sistemas móveis de baixa definição (LDTV – *Low Definition Television*). No entanto, alguns estudos apontam que o funcionamento não é satisfatório quando ocorrem transmissões simultâneas para sistemas de alta definição e sistemas móveis.

Na Europa, em um primeiro momento, está sendo utilizada a resolução padrão (SDTV - *Standard Definition Television*), inicialmente em formato de tela 4:3. Considerando a largura de banda do canal, a transmissão SDTV permite a difusão de até seis programas simultaneamente. Portanto, o modelo de negócios dos países europeus privilegiou a oferta diversificada de programas e serviços na transmissão terrestre. Por outro lado, a Austrália optou por combinar programas em alta definição (HDTV) e programas em definição padrão (SDTV).



Para as redes de televisão a cabo, o DVB-C (ETSI, 1998) adota a modulação 64-QAM. O termo 64 é a constelação do esquema de modulação, que representa o número de símbolos possíveis. O número de símbolos determina diretamente o número de bits associado a cada símbolo transmitido. Por exemplo, a modulação 64-QAM transporta 6 bits por símbolo, pois com 64 símbolos é possível representar as 64 combinações possíveis de 6 bits. Embora o DVB-C sugira a modulação 64-QAM, em função das características da rede e do serviço desejado, a modulação QAM também pode ser usada com outras constelações. Por exemplo, pode-se utilizar constelações de 16, 32, 64, 128 e 256 símbolos.

Para a difusão via satélite, o DVB-S (ETSI, 1997a) recomenda a modulação QPSK. Para a radiodifusão terrestre utilizando microondas, são previstos dois tipos de modulação. Para frequências abaixo de 10 GHz (MMDS), o DVB-MC (ETSI, 1997c) recomenda a utilização da modulação 16, 32 ou 64-QAM. Para frequências acima de 10 GHz (LMDS), o DVB-MS (ETSI, 1997b) recomenda o mesmo mecanismo de modulação que o DVB-S, ou seja, QPSK.

Nas camadas de transporte e codificação, o padrão DVB é um sistema fundamentalmente baseado no MPEG-2. Portanto, os padrões de transporte e codificação adotados pelo DVB são baseados nas recomendações MPEG-2.

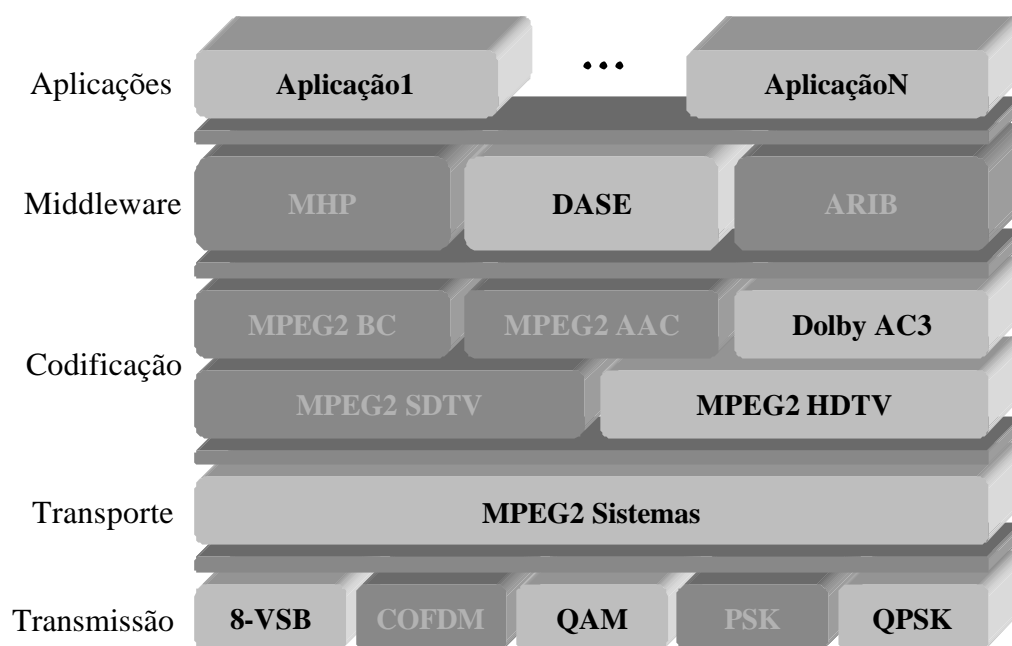
Na camada de codificação, o sinal de áudio é codificado usando a recomendação MPEG2-BC e o sinal de vídeo é codificado usando a recomendação MPEG-2 Vídeo (ISO, 1996b) com qualidade SDTV. Na camada de transporte, os fluxos elementares de áudio, vídeo e dados (aplicações) são multiplexados no produtor de conteúdo e demultiplexados nos *set-top boxes* dos usuários usando a recomendação MPEG-2 Sistemas (ISO, 1996a).

#### **4.2.2. ATSC**

O comitê ATSC (*Advanced Television Systems Committee*) (ATSC, 2004) é uma organização de padronização americana iniciada em 1982 e composta por cerca de 170 membros, incluindo fabricantes de equipamentos, operadores de redes, desenvolvedores de software e órgãos de regulamentação. O objetivo do comitê ATSC é especificar padrões para televisão digital.

O conjunto de padrões especificado pelo comitê ATSC caracteriza o padrão de sistema de televisão digital também denominado ATSC, que é conhecido como o padrão americano de televisão digital. Em funcionamento nos Estados Unidos desde novembro de 1998, o padrão ATSC também já foi adotado pelo Canadá, Coreia do Sul, Taiwan e Argentina. Nestes dois últimos países, existe uma sinalização que deverá ocorrer uma revisão do padrão de sistema de televisão digital a ser adotado.

Da mesma forma que o padrão DVB, o padrão ATSC é formado por um conjunto de documentos que definem os diversos padrões adotados, incluindo aqueles relacionados à transmissão, transporte, codificação e *middleware*. A Figura 12 ilustra esquematicamente a arquitetura do sistema ATSC, identificando os principais componentes padronizados nas diversas camadas.



**Figura 12. Arquitetura ATSC.**

O padrão ATSC permite diversas configurações para a camada de transmissão, definindo diferentes esquemas de modulação para transmissão terrestre, via cabo e via satélite. Na radiodifusão terrestre, o padrão ATSC pode operar com canais de 6, 7 ou 8 MHz e utiliza a modulação 8-VSB, cuja taxa de transmissão é de 19,8 Mbps. Em função deste esquema de modulação, um sistema ATSC apresenta problemas na recepção por antenas internas e não permite a recepção móvel. Para as redes de televisão a cabo e as transmissões via satélite, da mesma forma que o padrão DVB, o padrão ATSC adota as modulações 64-QAM e QPSK, respectivamente.

O padrão ATSC prevê diversos modos de transmissão com diferentes níveis de resolução da imagem e formatos de tela. No entanto, o modelo de negócios americano foi direcionado para a televisão de alta definição (HDTV). Em função do alto custo dos aparelhos de televisão de alta definição, o sistema americano de televisão digital ainda possui uma baixa adesão dos usuários.

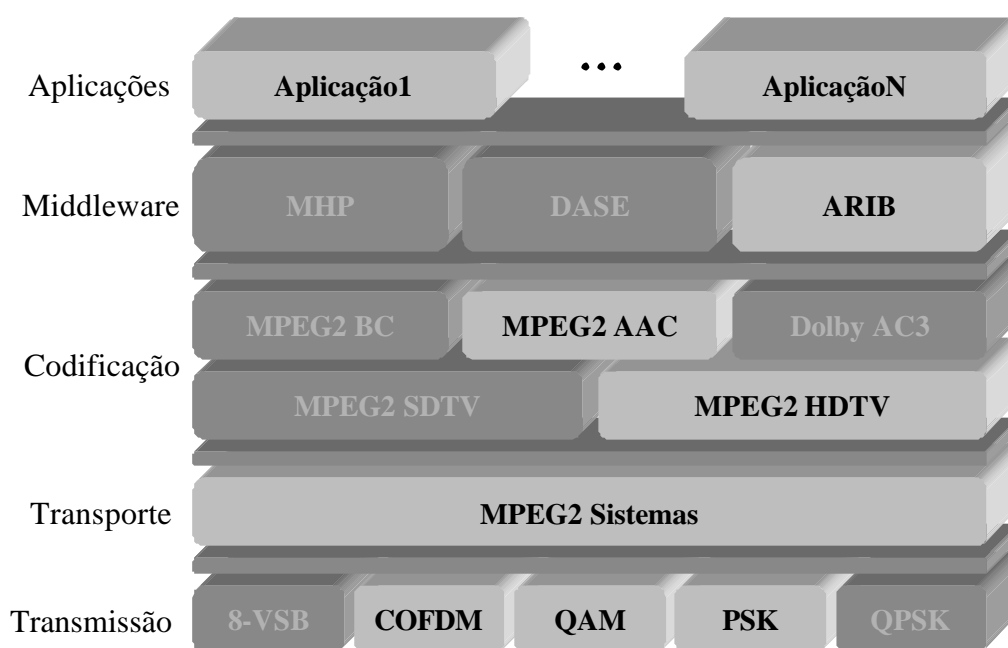
Na camada de codificação, o sinal de áudio é codificado usando o padrão Dolby AC-3 (ATSC, 2001) e o sinal de vídeo é codificado usando a recomendação MPEG-2 Vídeo (ISO, 1996b) com qualidade HDTV. Na camada de transporte, da mesma forma que o padrão DVB, o padrão ATSC multiplexa e demultiplexa os fluxos elementares de áudio, vídeo e dados (aplicações) usando a recomendação MPEG-2 Sistemas (ISO, 1996a).

#### **4.2.3. ISDB**

O padrão ISDB foi especificado em 1999 no Japão pelo grupo DiBEG - *Digital Broadcasting Experts Group* (DiBEG, 2004), criado em 1997 e composto por várias empresas e operadoras de televisão. O objetivo do grupo DiBEG é promover e especificar o sistema de difusão terrestre de televisão digital japonês.

O padrão ISDB é também conhecido como o padrão japonês de televisão digital. Até o momento, o padrão ISDB foi adotado apenas no Japão, porém é amplamente divulgado que o mesmo é um sistema que reúne o maior conjunto de facilidades: alta definição - HDTV, transmissão de dados e recepção móvel e portátil.

Da mesma forma que os padrões DVB e ATSC, o padrão ISDB é formado por um conjunto de documentos que definem os diversos padrões adotados, incluindo aqueles relacionados à transmissão, transporte, codificação e *middleware*. A Figura 13 ilustra esquematicamente a arquitetura do sistema ISDB, identificando os principais componentes padronizados nas diversas camadas.



**Figura 13. Arquitetura ISDB**

Da mesma forma que os padrões DVB e ATSC, o padrão ISDB permite diversas configurações para a camada de transmissão, definindo diferentes esquemas de modulação para transmissão terrestre, via cabo e via satélite. Na radiodifusão terrestre, a especificação ISDB-T (*Integrated Services Digital Broadcasting – Terrestrial*) (ISDB, 1998) pode operar com canais de 6, 7 ou 8 MHz e, da mesma forma que o padrão DVB, utiliza a modulação COFDM (*Coded Orthogonal Frequency Division Multiplexing*), mas com algumas variações, alcançando uma taxa de transmissão que varia entre 3,65 e 23,23 Mbps. Para as redes de televisão a cabo e as transmissões via satélite, o padrão ISDB adota as modulações 64-QAM e 8-PSK, respectivamente.

O ISDB é projetado para suportar sistemas hierárquicos com múltiplos níveis, podendo ser usado, por exemplo, para prover simultaneamente recepção de baixa taxa de dados sob condições móveis excepcionalmente difíceis, taxa de dados intermediária (SDTV) para recepção estática e alta taxa de dados (HDTV) para boas condições de recepção.

Embora seja baseado no sistema de transmissão europeu, o ISDB-T é superior ao DVB-T quanto à imunidade a interferências, permitindo a convivência da televisão de alta definição com a recepção móvel.

Na camada de codificação, o sinal de áudio é codificado usando a recomendação MPEG-2 AAC (*Advanced Audio Coding*) (ISO, 1997) e o sinal de vídeo é codificado usando a recomendação MPEG-2 Vídeo (ISO, 1996b) com qualidade HDTV. Vale ressaltar que, em função da flexibilidade do sistema, o sinal de vídeo pode ser codificado usando a recomendação MPEG-2 Vídeo em diferentes níveis de resolução.

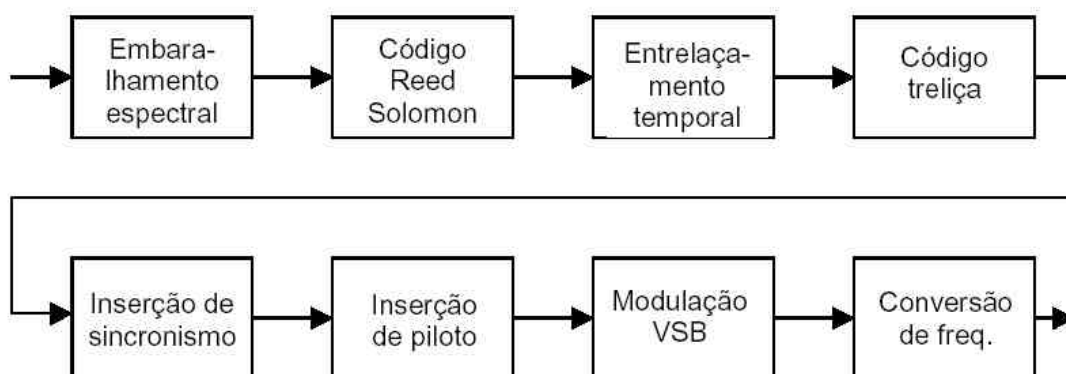
Na camada de transporte, da mesma forma que os padrões DVB e ATSC, o padrão ISDB multiplexa e demultiplexa os fluxos elementares de áudio, vídeo e dados (aplicações) usando a recomendação MPEG-2 Sistemas (ISO, 1996a).

Segue uma descrição resumida dos principais padrões de modulação, transmissão, codificação, compressão, multiplexação e transporte, utilizados nos padrões de televisão digital.

### 4.3. Padrões para Modulação e Transmissão

Os sistemas de televisão digital já definidos adotam dois esquemas de modulação – o 8-VSB e o COFDM – que são descritos a seguir.

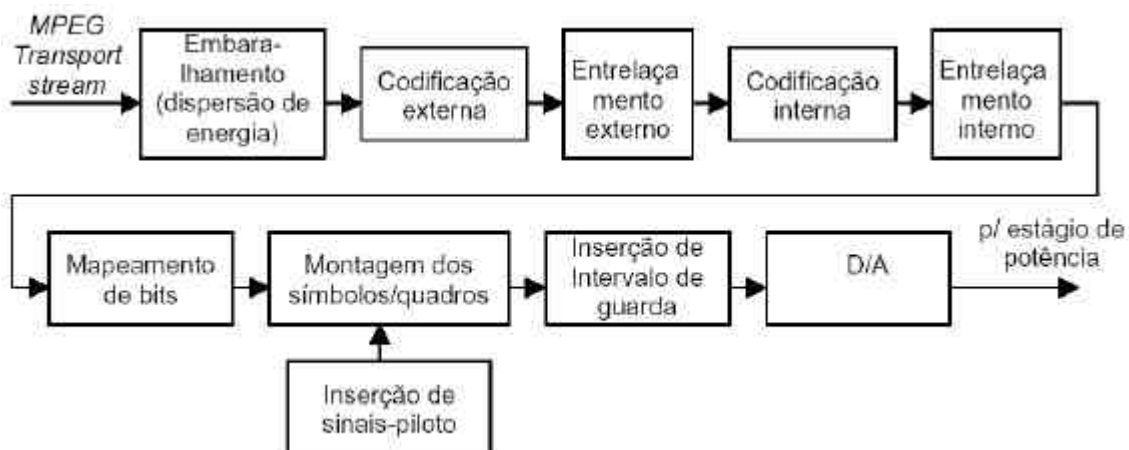
**Modulação 8VSB (Vestigial Sideband).** O ATSC definiu como esquema de modulação para transmissão terrestre o 8-VSB [Sparano]. A Figura 14 ilustra o processo de modulação 8-VSB [13]<sup>16</sup>. O fluxo de bits MPEG-2 transporte é embaralhado, para suavizar o espectro, evitando a concentração de energia em alguns pontos. Em seguida, o sinal passa por um gerador de código corretor de erros (Reed Solomon) que opera em nível de blocos, inserindo 20 bytes de paridade para cada bloco de 187 bytes. Os 207 bytes formam um segmento. Depois da codificação é realizado o entrelaçamento temporal, onde os bytes são espalhados em 52 segmentos com o objetivo de evitar que um ruído impulsivo danifique um ou vários segmentos inteiros. O espalhamento distribui os erros provocados pelo ruído em bytes de vários segmentos, o que em conjunto com o código corretor de erros, garante uma boa imunidade do sistema a ruídos impulsivos. É inserido então um segundo código corretor de erros (treliça ou convolucional), onde cada 2 bits originais são convertidos para 3 bits. O terceiro bit melhora a redundância da informação. Os 3 bits são convertidos para um símbolo de 8 níveis ( $2^3$ ). A carga útil de cada segmento é formada por 828 símbolos de 8 níveis.



**Figura 14. Modulação 8-VSB.**

Os segmentos recebem então alguns símbolos adicionais de sincronismo. 312 segmentos, mais um de sincronismo, formam um quadro. Esse conjunto (que é um sinal AC), recebe um pequeno nível DC, o qual, ao ser modulado, aparecerá como um ressaltado no espectro, formando o sinal piloto do canal. Esse conjunto é colocado num modulador VSB, que pode ser analógico ou um circuito que sintetize digitalmente a forma de onda já em rádio-freqüência (mais precisamente, em FI – freqüência intermediária). O sinal resultante está pronto para ser adaptado para a freqüência de operação da emissora, amplificado e transmitido.

**Modulação COFDM (*Coded Orthogonal Frequency-Division Multiplexing*).** O COFDM é baseado na utilização de diversas portadoras, onde cada portadora transporta uma parte do sinal em subcanais FDM (Frequency Division Multiplexing) em um canal de 6, 7 ou 8 MHz. No DVB são usadas 1705 (modo 2K) ou 6817 (modo 8K) portadoras. A interferência entre as portadoras é evitada por condições de ortogonalidade entre as mesmas, que ocorre quando o espaçamento entre as portadoras é o inverso do período sobre o qual o receptor fará a operação de demodulação do sinal. Para melhorar a imunidade a interferências externas, é utilizada uma série de técnicas de codificação, que inclui uma permuta pseudo-aleatória da carga útil entre as diversas portadoras. A Figura 15 ilustra o processo de codificação de um sinal COFDM.



**Figura 15. Modulação COFDM.**

O feixe de sinal recebido do multiplexador MPEG é embaralhado, para distribuir de maneira uniforme a energia. Em seguida o sinal passa por um primeiro processo de codificação externa, que utiliza o Reed-Solomon para criar bits redundantes utilizados para recuperação de erros. Os bytes de cada 12 blocos são entrelaçados para que, caso algum bloco não seja recebido sejam perdidos poucos bits por bloco em vez de um bloco completo. Na codificação interna é usado um código convolucional FEC (Forward Error Correction) que gera bits adicionais para melhorar a redundância. Alguns dos bits adicionais são omitidos em intervalos regulares para desbalancear a energia dos símbolos. Desta forma alguns símbolos (os que tiveram bits omitidos) ficam com a energia reduzida, enquanto outros ganham um reforço de potência. Os símbolos com melhor relação sinal/ruído (SNR) são utilizados para transportar as informações de controle e sincronismo

do canal. Após o entrelaçamento interno os bits são mapeados para compor os símbolos e quadros da transmissão. A montagem é parametrizável podendo ser definidos o tipo de modulação (QPSK, 16-QAM ou 64-QAM), número de portadoras e intervalo de guarda. Para uma dada configuração dos parâmetros supracitados, os bits são agrupados para formar uma palavra. Cada palavra irá modular uma portadora, durante um tempo  $T_U$ . O conjunto de palavras de todas as portadoras num dado intervalo  $T_U$  é chamado de símbolo COFDM. Cada conjunto de 68 símbolos COFDM forma um quadro COFDM.

Algumas portadoras são utilizadas como sinal piloto, sendo utilizadas para sincronismo e controle de fase. As duas portadoras extremas do canal (as de número 0 e 1704 no modo 2K e 0 e 6816 no modo 8K) têm essa finalidade. Outras 43 portadoras são utilizadas como piloto contínuo no modo 2K e 175 no modo 8K. Nas demais portadoras, algumas palavras são utilizadas dentro de uma seqüência predefinida para atuar como sinais pilotos dispersos, que são usados para estimar as características de transmissão da portadora e de portadoras adjacentes. Algumas portadoras são utilizadas para transportar um sinal de controle chamado TPS (Transmission Parameter Signalling) que identifica os parâmetros de transmissão do canal, como o tipo de modulação, número de portadoras, etc.

O ISDB apresenta três modos de operação COFDM, existe um modo intermediário 4K. Existem também diferenças nos outros modos. Por exemplo, o número de portadoras no modo 2K ISDB é 1405 e 1705 no modo 2k do DVB.

#### **4.4. Padrão para Multiplexação e Transporte**

A função do subsistema de multiplexação e transporte é receber as seqüências elementares de bits geradas pelos codificadores de aplicações dos diferentes subsistemas (vídeo, áudio, dados auxiliares, etc) e, através da multiplexação, gerar em sua saída uma seqüência única de pacotes, cujo formato é definido pelo padrão *MPEG-2 Systems*. A codificação dos pacotes pode ser realizada de duas formas: fluxo de transporte ou fluxo de programa. No fluxo programa os pacotes gerados possuem tamanho variável e usualmente grande. Este tipo de fluxo é usado para sistemas de transmissão com baixa probabilidade de ocorrência de erros, o que não é o caso de sistemas de televisão. No fluxo transporte os pacotes possuem tamanho fixo de 188 bytes, sendo mais adequados para tratamento de erros, além de simplificar a implementação de circuitos eletrônicos e algoritmos, e aumentar a velocidade de processamento.

A seqüência de pacotes de transporte resultante da multiplexação pode ser novamente multiplexada com outras seqüências do mesmo tipo antes do envio para o subsistema de transmissão. No receptor, essa seqüência de pacotes será demultiplexada e as seqüências elementares de bits serão reconstruídas e entregues aos seus respectivos decodificadores. Utilizando informações contidas no cabeçalho dos pacotes de transporte, é possível a realização de operações como sincronização do aparelho receptor, detecção e sinalização de erros.

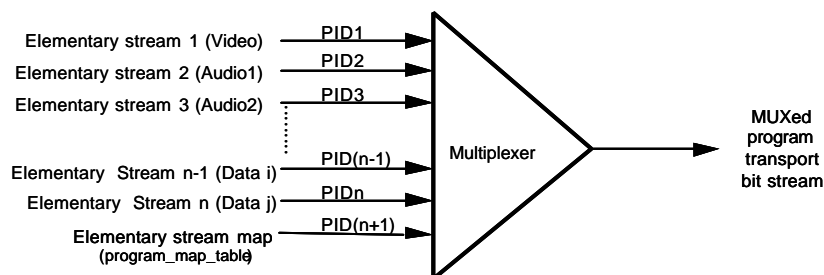
As seqüências elementares de bits podem ou não, antes da multiplexação e formatação em pacotes de transporte, passar por um processo de organização em segmentos PES (Packetized Elementary Stream) de tamanho variável. As principais finalidades da segmentação PES são viabilizar a sincronização das seqüências elementares de bits de um mesmo programa. As seqüências de áudio e vídeo passam obrigatoriamente

por essa etapa. O processo de geração de segmentos PES pode ser realizado diretamente pelo subsistema de multiplexação e transporte ou pelo próprio codificador da aplicação geradora da sequência elementar de bits.

A multiplexação das seqüências de dados auxiliares, áudio e vídeo, nos pacotes de transporte é realizada através de um simples campo identificador em seu cabeçalho. Esse campo é denominado o PID (Packet Identifier), e sua utilização permite, por exemplo, que a capacidade do canal seja alocada de forma dinâmica para rajadas de determinado subsistema gerador de seqüências elementares.

A arquitetura empregada na multiplexação e no transporte permite a extensibilidade dos serviços oferecidos, ao mesmo tempo que garante compatibilidade futura com o parque de equipamentos já instalados. Novos serviços serão implementados através do emprego de novos PIDs, sem que seja alterada a estrutura do pacote. Dessa maneira, os equipamentos que não estejam preparados para recebê-los, simplesmente filtrarão os pacotes de PIDs desconhecidos, decodificando apenas as seqüências de pacotes cujo tratamento é possível ser realizado.

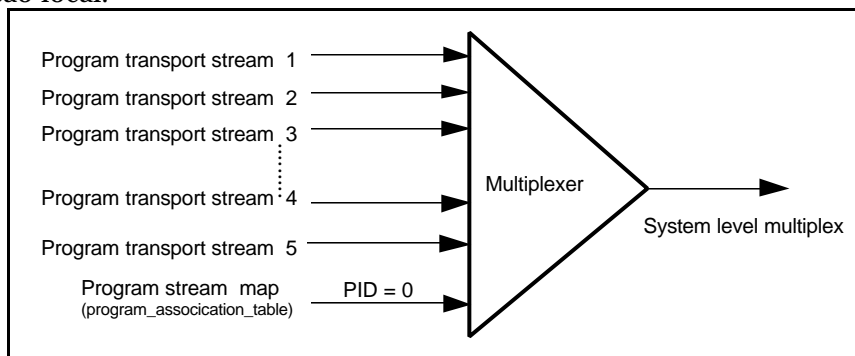
A multiplexação no subsistema de multiplexação e transporte é realizada em dois níveis distintos. Inicialmente, as seqüências elementares de bits (em formato PES ou não), que compartilham uma mesma base de tempo, são multiplexadas entre si e com uma seqüência de controle, chamada de *Elementary Stream Map* para formar um programa. Programa é o termo utilizado na televisão digital para a denominação do que vem a ser um “canal” na TV tradicional. Nesse primeiro nível de multiplexação, cada seqüência elementar possui seu próprio identificador, chamado *Stream\_ID*. Não há restrições sobre o número e tipo de seqüências elementares presentes em um programa. A seqüência de controle possui uma tabela, a *Program Map Table*, que inclui informações sobre os identificadores de cada uma das seqüências que compõem o programa e sobre o relacionamento entre as mesmas. A Figura 16 ilustra a multiplexação nesse nível, supondo que as seqüências elementares já estão na forma de pacotes de transporte, após passarem pela etapa PES.



**Figura 16. Multiplexação de fluxos elementares**

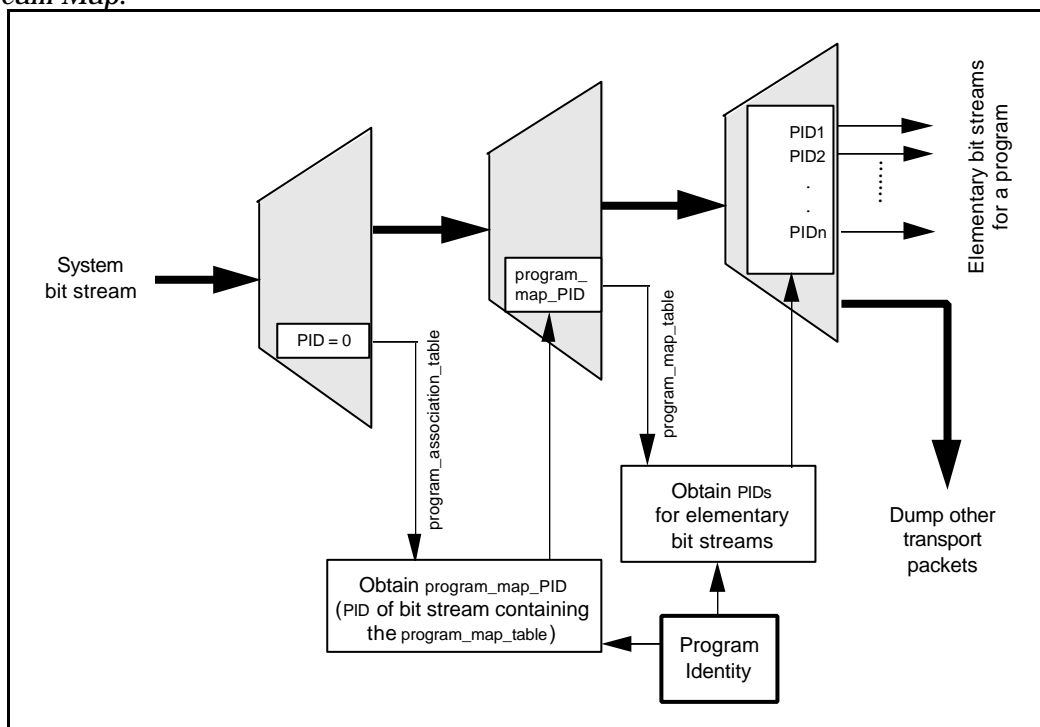
Os programas, por sua vez, são multiplexados assincronamente entre si e com uma seqüência de controle de mais alto nível, a *Program Stream Map*, para formar a seqüência de transporte do sistema. Essa seqüência de controle contém, de forma análoga à *Program Map Table* de um programa, uma tabela de mapeamento entre os programas e suas seqüências de transporte, a *Program Association Table*.

Esse segundo nível de multiplexação, mostrado na Figura 17, oferece uma funcionalidade importante, que é a definição de programas com uma combinação qualquer de seqüências PES, incluindo repetições e seleções de seqüências específicas. Podemos citar como exemplo uma mesma seqüência de áudio que deve ser sincronizada com duas seqüências de vídeo para a composição de dois programas diferentes. Além dessa funcionalidade, a multiplexação a nível de programas permite a inserção de programação local.



**Figura 17. Multiplexação de programas**

Esse segundo nível de multiplexação pode ocorrer recursivamente, ou seja, podem haver sucessivas multiplexações de várias seqüências de sistema em uma única seqüência de maior largura de banda. Esse procedimento recursivo exige, no entanto, a recriação da *Program Association Table*, gerando, conseqüentemente, uma nova seqüência *Program Stream Map*.

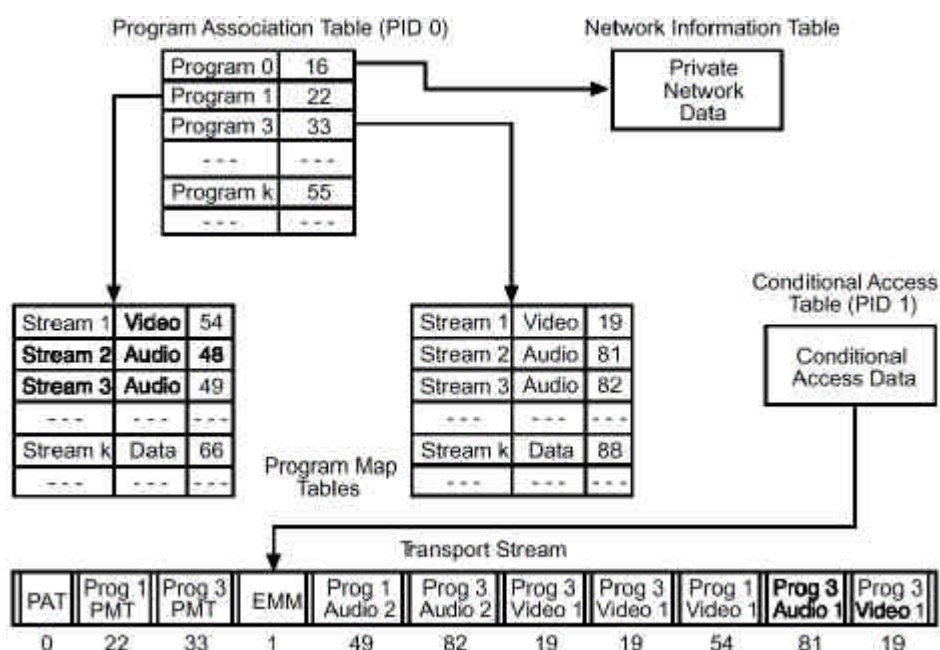




**Figura 1 Demultiplexação de programas e fluxos elementares.**

A política de multiplexação e o funcionamento do multiplexador não são objetos de padronização, nem mesmo sendo necessária sua implementação em dois níveis distintos. Apenas o formato das seqüências deve ser obedecido, de forma a serem possíveis suas decodificações. A Figura 18 apresenta a estrutura da demultiplexação de uma seqüência de pacotes de transporte em um aparelho receptor de TV digital.

A Figura 19 apresenta um exemplo sumário da estrutura básica de fluxos que compõem um MPEG-2-TS. Cada fluxo usa um identificador único, chamado de PID (packet id). No padrão MPEG-2, três identificadores de fluxo são reservados para usos especiais, que são: PAT (Program Association Table) – PID=0, CAT (Conditional Access Table) – PID = 1 e TSDT (Transport Stream Description Table) – PID = 2. O fluxo PAT indica quais são todos os programas que são veiculados no TS. O PAT, de fato, indica apenas os PIDs dos fluxos que contém as tabelas dos programas, chamadas PMT (Program Map Table). Cada PMT indica os fluxos que compõem o programa. Para cada fluxo é especificado o tipo (vídeo, áudio ou dados) e o PID dos pacotes que podem ser usados para gerar os streams elementares de vídeo, áudio ou dados. O primeiro programa da tabela PAT contém informações específicas da rede difusora, como dados sobre outros serviços que podem estar disponíveis em outros canais ou frequências. A parte inferior da Figura 19 apresenta como estes pacotes podem ser multiplexados gerando um MPEG-2-TS.



**Figura 19. Estrutura de tabelas do MPEG-2-TS (Tektronix, 2002).**

Através do acesso às tabelas que estão nos fluxos PAT, CAT e PMT o multiplexador pode ser programado para inserir, remover e renomear programas e fluxos de vídeo, áudio e dados, tanto no subsistema de Estúdio quanto no subsistema de Rádio-

Difusão, tornando possível, a mistura de centenas de fluxos de conteúdos produzidas por diversos estúdios, o que potencializa enormemente a aquisição e veiculação de conteúdos em sistemas de TVDI.

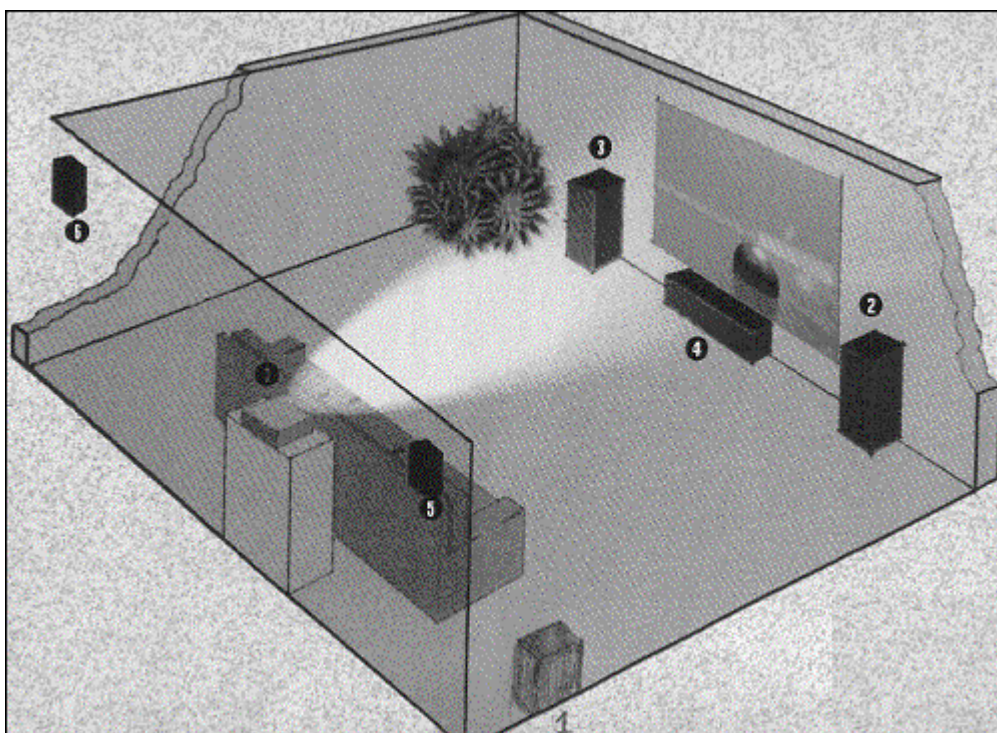
#### 4.5. Padrões para Codificação e Compressão

Todos os sistemas já definidos adotaram para codificação e compressão de vídeo o padrão MPEG-2. Este padrão na realidade faz parte de uma família de padrões (MPEG-1, MPEG-4, MPEG-7 etc.) de compressão de áudio, vídeo, codificação de objetos multimídia, multiplexação de sinais e descrição de objetos de mídia. O MPEG-2 por sua vez é composto por diversos padrões para vídeo (ISO, 1996b), para áudio com compatibilidade regressiva (ISO, 1998a), etc.

O método de compressão do MPEG-2 Vídeo baseia-se em algoritmos assimétricos, onde o custo da codificação é muito maior que o da decodificação. Esta é uma característica interessante para a televisão, pois o alto custo do codificador é assimilado pelo rádio-difusor, enquanto que o receptor do telespectador requer um decodificador de baixo custo. Os algoritmos são bastante flexíveis, possibilitando a codificação de imagens com diferentes níveis de resolução (qualidade).

A parte do MPEG-2 que trata da codificação de vídeo é um padrão genérico, contendo muitos algoritmos e ferramentas. O uso de diferentes subconjuntos do MPEG de uma forma desordenada poderia inviabilizar a interoperabilidade dos sistemas. Por tal motivo, foi criada uma estrutura hierarquizada de perfis e níveis, de forma a garantir a interoperabilidade de sistemas mesmo que estes estejam operando em níveis diferentes. Existem cinco perfis definidos, sendo que cada perfil contempla um conjunto de facilidades, ou seja, de algoritmos e ferramentas, sendo orientado a determinados tipos de aplicações. Os níveis definem as restrições sobre os parâmetros, o que restringe o escopo das aplicações. Dentro de cada perfil, um nível mais alto engloba todas as funcionalidades do nível inferior.

Com relação a codificação de áudio, o sistema Europeu prevê o uso do MPEG-2 BS e o Japonês do MPEG-2 AAC. Os americanos optaram por usar o padrão Dolby AC-3. Todos os sistemas prevêem a transmissão de 6 canais distribuídos conforme mostra a Figura 20. Onde uma caixa de som deve ser instalada exatamente à frente da audiência, acima ou abaixo do aparelho de televisão, principalmente para reprodução dos diálogos. Nas laterais, à frente, deve ser instalado um par de caixas de som, para a reprodução da trilha sonora do programa sendo assistido, de forma similar ao efeito estéreo já conhecido. Atrás da audiência, lateralmente, deve ser posicionado mais um par de caixas de som, para a reprodução do som *surround*, cuja principal função é proporcionar a terceira dimensão da trilha sonora. Por fim, uma sexta e última caixa de som, especial para a reprodução de sons de baixa frequência (conhecida como *subwoofer*), deverá ser posicionada, preferencialmente, próxima a uma das extremidades do ambiente. O ambiente estabelecido por um aparelho de televisão (normalmente de tela grande, maior do que 29 polegadas) e a distribuição das caixas de som acima apresentada ficou conhecido nos últimos anos como *home theater*.



**Figura 20. Distribuição das caixas de som em ambientes de TV Digital.**

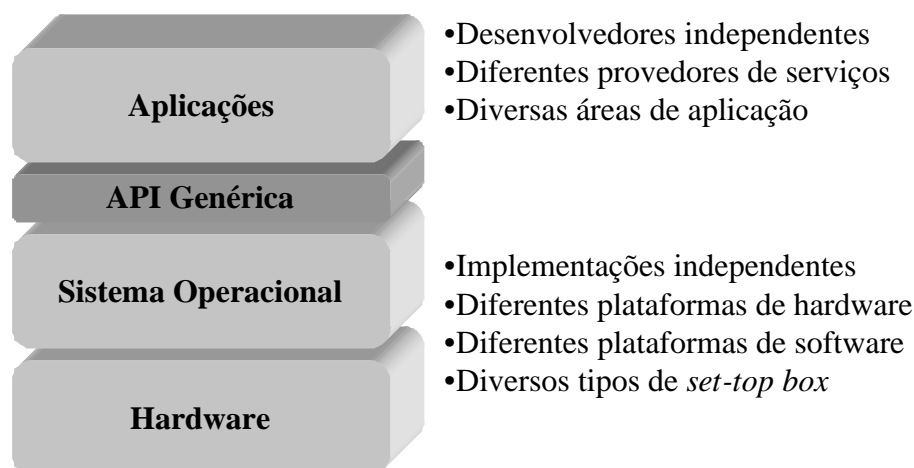
## **5. Padrões de Middleware**

As tecnologias de TVDI permitem a fabricação de STBs com diferentes arquiteturas de hardware, cujas capacidades de processamento, armazenamento e comunicação são bastante variáveis. Além disso, estes diversos dispositivos também podem adotar diferentes sistemas operacionais.

Neste cenário de hardware e software heterogêneos, os desenvolvedores de aplicações devem escrever diferentes versões dos programas para cada combinação de hardware e sistema operacional dos diversos tipos de STBs. Conseqüentemente, a heterogeneidade das plataformas torna o desenvolvimento de aplicações para TVDI uma atividade ineficiente e de custo elevado, que pode inviabilizar sua adoção em larga escala.

Para tornar mais eficiente o desenvolvimento de aplicações, bem como reduzir os custos associados, favorecendo assim a consolidação da TVDI, os fabricantes e provedores de conteúdo perceberam que a solução é adotar mecanismos que tornem portáteis as aplicações e os serviços nos diversos tipos de STBs.

Neste sentido, para atender ao requisito de portabilidade, os STBs devem prover às aplicações uma API (*Application Programming Interface*) genérica, padronizada e bem definida (Figura 21). Esta API deve abstrair as especificidades e heterogeneidades de hardware e software dos diversos tipos de dispositivos receptores.



**Figura 21. Portabilidade baseada em API genérica.**

Para prover esta API genérica, uma camada de software adicional, denominada *middleware*, deve ser incluída entre o sistema operacional e as aplicações. O objetivo do *middleware* é oferecer um serviço padronizado às aplicações, escondendo as especificidades e heterogeneidades das camadas de hardware e sistema operacional, que dão suporte as facilidades básicas de codificação, transporte e modulação de um sistema de televisão digital.

Oferecendo uma API padronizada, o *middleware* incrementa a portabilidade das aplicações. Desta forma, as aplicações não acessam diretamente as facilidades providas pelo sistema operacional e o hardware do dispositivo, mas apenas os serviços oferecidos pela camada de *middleware*. Conseqüentemente, sem qualquer tipo de modificação no código, as aplicações podem ser diretamente executadas em qualquer STB que suporte o *middleware* adotado no desenvolvimento das mesmas.

Em função dos benefícios da adoção de uma camada de *middleware*, diversos órgãos de padronização concentraram esforços na especificação de padrões de *middleware*. Como resultado destes esforços, atualmente, existem três padrões de *middleware* para TVDI: MHP - *Multimedia Home Platform* (ETSI, 2003c), DASE - *DTV Application Software Environment* (ATSC, 2003) e ARIB - *Association of Radio Industries and Businesses* (ARIB, 2002).

Apesar da existência de padrões, para ser suportado nos diversos tipos de STBs, um determinado padrão de *middleware* deve ser implementado para cada plataforma de hardware e sistema operacional. Vale ressaltar que, embora as aplicações se tornem portáveis entre diferentes plataformas de hardware e sistema operacional, elas se tornam dependentes do *middleware* adotado. Ou seja, uma aplicação desenvolvida para o *middleware* MHP não é diretamente portátil para o DASE e o ARIB.

## **5.1. Blocos Fundamentais**

Embora os diversos padrões de *middleware* não sejam compatíveis entre si, estes padrões adotam versões modificadas, reduzidas ou estendidas de determinados componentes. Dentre estes blocos fundamentais comuns, podemos destacar os seguintes componentes: DAVIC - *Digital Audio-Visual Council* (DAVIC, 1999); HAVi - *Home Audio Video*

*Interoperability* (HAVi, 2001) e Java TV (Sun, 2000). Em função da ampla adoção destes blocos fundamentais, antes de descrever os padrões de televisão digital, apresentamos uma breve descrição das principais funcionalidades do DAVIC, HAVi e Java TV.

### 5.1.1. DAVIC

DAVIC é uma associação que representa diversos setores da indústria audiovisual, que foi iniciada em 1994, mas extinta após 5 anos de atividade, conforme já previsto no seu estatuto. O principal objetivo da associação DAVIC foi especificar um padrão da indústria para interoperabilidade fim-a-fim de informações audiovisual digital interativa e por difusão.

Para obter esta interoperabilidade, as especificações DAVIC (DAVIC, 1999) definem interfaces em diversos pontos de referência, onde cada interface é definida por um conjunto de fluxos de informações e protocolos. As especificações DAVIC especificam formatos de conteúdos para diversos tipos de objetos (fonte, texto, hipertexto, áudio e vídeo) e também incluem facilidades para controlar a língua adotada no áudio e na legenda.

Além disso, as especificações DAVIC definem diversas APIs relacionadas a informações de serviços, filtragem de informações, notificação de modificações nos recursos, sintonização de canais de transporte (*tuning*) e controle de acesso:

- ?? *Service Information API*: provê às aplicações uma interface de alto nível para acessar informações de serviços presentes em fluxos MPEG-2. Esta API define métodos para acessar todas as informações presentes nas tabelas de serviços (*SI Tables*), permitindo, por exemplo, que um guia de programação eletrônico (*EPG – Electronic Program Guide*) possa identificar o escalonamento dos programas de cada serviço.
- ?? *MPEG-2 Section Filter API*: permite as aplicações identificarem a ocorrência de determinados padrões nos dados mantidos em seções privadas MPEG-2.
- ?? *Resource Notification API*: define um mecanismo padrão para aplicações registrarem interesse em determinados recursos e serem notificadas de mudanças nestes recursos.
- ?? *Tuning API*: especifica uma interface de alto nível para fisicamente sintonizar os diferentes fluxos de transporte.
- ?? *Conditional Access API*: provê uma interface básica para o sistema de controle de acesso. Por exemplo, esta API permite a aplicação verificar se o usuário possui direito de acesso a um dado serviço ou evento.
- ?? *DSM-CC User-to-Network API*: define mecanismos para que as aplicações possam controlar as sessões DSM-CC.

Para apresentação de saída gráfica, as especificações DAVIC adotam um subconjunto do pacote AWT de interface com o usuário da API Java. Para apresentar fluxos de áudio e vídeo, as especificações DAVIC adotam o JMF - *Java Media Framework* (Sun, 1999) e definem algumas extensões para características específicas de

televisão digital. Por exemplo, as especificações definem APIs para sincronizar aplicações em um instante específico de tempo de um determinado conteúdo e gerenciar eventos incluídos no conteúdo ou início da apresentação de uma determinada mídia.

### 5.1.2. HAVi

HAVi é uma iniciativa das oito maiores companhias de produtos eletrônicos que especifica um padrão para interconexão e interoperação de dispositivos de áudio e vídeo digital. A especificação (HAVi, 2001) permite que os dispositivos de áudio e vídeo da rede possam interagir entre si, como também define mecanismos que permitem que as funcionalidades de um dispositivo possam ser remotamente usadas e controladas a partir de outro dispositivo.

A especificação HAVi é independente de plataforma e linguagem de programação, podendo ser implementada em qualquer linguagem para qualquer processador e sistema operacional. Desta forma, a especificação HAVi permite que os fabricantes projetem dispositivos interoperáveis e os desenvolvedores de aplicações possam escrever aplicações Java para estes dispositivos usando a API provida pelo HAVi.

No contexto de televisão digital interativa, o STB pode ser conectado em uma rede HAVi, podendo compartilhar seus recursos com outros dispositivos e usar os recursos de outros dispositivos para compor aplicações mais sofisticadas. Por exemplo, um STB pode gerar um menu completo que permite ao usuário acessar funcionalidades de qualquer dispositivo ou uma combinação de dispositivos HAVi, usando somente o controle remoto da televisão e apresentando o sistema de forma consistente para o usuário. Como outro exemplo, um STB pode automaticamente programar o aparelho de vídeo cassete a partir das informações obtidas do guia de programação eletrônico (*EPG – Electronic Program Guide*).

HAVi adota o padrão de rede IEEE-1394 - *Firewire* (IEEE, 1995) que atualmente suporta uma taxa de transmissão de até 400Mbps e é capaz de suportar comunicação isócrona, tornando-o adequado para tratamento simultâneo de múltiplos fluxos de áudio e vídeo digital em tempo real.

A especificação HAVi define uma arquitetura de software distribuída cujos elementos de software asseguram a interoperabilidade e implementam serviços básicos tais como: gerenciamento da rede, comunicação entre dispositivos e gerenciamento da interface com os usuários. HAVi define um conjunto de serviços distribuídos que suportam APIs Java padronizadas, permitindo que aplicações distribuídas possam transparentemente acessar os serviços através da rede. Para assegurar a interoperabilidade, todos os elementos de software se comunicam usando um mecanismo de passagem de mensagem que adota formatos de mensagens e protocolos padronizados pelo HAVi. Os elementos de software da arquitetura HAVi são:

?? *1394 Communication Media Manager*: coordena a comunicação assíncrona e isócrona em uma rede IEEE-1394.

?? *Messaging System*: responsável pela passagem de mensagens entre os diversos elementos de software.

- ?? *Registry*: define um serviço de diretório que permite localizar os diversos elementos de software na rede e identificar suas funcionalidades e propriedades.
- ?? *Event Manager*: implementa um serviço de notificação de eventos, que sinaliza mudanças no estado dos elementos de software ou na configuração da rede HAVi.
- ?? *Stream Manager*: gerencia a transferência em tempo real de fluxos de áudio e vídeo entre elementos de software.
- ?? *Resource Manager*: controla o compartilhamento de recursos e realiza o escalonamento de ações.
- ?? *Device Control Module (DCM)*: representa um dispositivo da rede HAVi e expõe as APIs deste dispositivo. Cada dispositivo da rede HAVi possui um DCM associado.
- ?? *DCM Manager*: coordena a instalação e remoção de DCMs.

### 5.1.3. Java TV

Java TV (Sun, 2000) é uma extensão da plataforma Java que permite a produção de conteúdo para televisão interativa. O principal objetivo de Java TV é viabilizar o desenvolvimento de aplicações interativas portáteis, que são independentes da tecnologia de rede de difusão.

Java TV consiste de uma máquina virtual Java JVM – Java Virtual Machine (Lindholm, T. and Yellin, F., 1999) e várias bibliotecas de códigos reusáveis e específicos para televisão digital interativa. A JVM é hospedada e executada no próprio STB. Java TV foi desenvolvido sobre o ambiente J2ME (Sun, 2002), que foi projetado para operar em dispositivos com restrições de recursos. Neste contexto, determinadas características da plataforma Java foram excluídas, pois são consideradas desnecessárias ou inadequadas para tais ambientes. No entanto, o J2ME não define funcionalidades específicas de televisão. Tais funcionalidades são incluídas no Java TV.

Java TV permite níveis avançados de interatividade, gráficos de qualidade e processamento local no próprio STB. Estas facilidades oferecem um amplo espectro de possibilidades para os desenvolvedores de conteúdo, mesmo na ausência de um canal de retorno. Por exemplo, EPGs podem oferecer uma visão geral da programação disponível, permitindo a mudança para o canal desejado pelo usuário. Através de mecanismos de sincronização, aplicações específicas podem ser associadas a um determinado programa de televisão. Além disso, aplicações isoladas podem executar de forma independente do programa de televisão.

Em Java TV, programas de televisão tradicionais e interativos são caracterizados como um conjunto de serviços individuais. Um serviço é uma coleção de conteúdo para apresentação em um STB. Por exemplo, um serviço pode representar um programa de televisão convencional, com áudio e vídeo sincronizados, ou um programa de televisão interativa, que contém áudio, vídeo, dados e aplicações associadas.

Cada serviço Java TV é caracterizado por um conjunto de informações que descrevem o conteúdo do serviço (*SI - Service Information*). As informações sobre os serviços disponíveis são armazenadas em uma base de dados de informações de serviços (*SI database*).



A API Java TV provê uma abstração que permite aplicações obterem informações sobre os diversos serviços disponíveis de forma independente do hardware e dos protocolos adotados. Desta forma, uma aplicação pode ser reusada em uma variedade de ambientes.

Java TV define vários pacotes que suportam um conjunto de facilidades para selecionar serviços, obter informações dos serviços, filtrar informações de serviços, controlar a apresentação dos serviços, acessar informações que são entregues através do canal de difusão e gerenciar o ciclo de vida das aplicações. As informações dos serviços podem ser acessadas através de filtros que encontram apenas os serviços de interesse da aplicação. A API Java TV usa o JMF (Sun, 1999) para tratar os fluxos digitais que são recebidos pelo STB, definindo fontes de dados e manipuladores de conteúdo.

Uma aplicação Java TV é denominada *Xlet*. *Xlets* não precisam estar previamente armazenados no STB, pois podem ser enviados pelo canal de difusão quando necessários. Ou seja, o modelo *Xlet* é baseado na transferência de código executável pelo canal de difusão para o STB e posterior carga e execução do mesmo, de forma automática ou manual. Um *Xlet* é bastante similar a um *Applet* na *Web* ou *MIDlet* em celulares e outros dispositivos móveis.

O ciclo de vida de um *Xlet* é composto por 4 estados: *loaded*, *paused*, *active* e *destroyed*. Todo *Xlet* deve implementar a interface *javax.tv.xlet.Xlet*, cujos métodos são ativados para sinalizar mudanças de estado da aplicação. A Figura 22 ilustra o ciclo de vida de um *Xlet*, identificando os estados e os métodos suportados por sua interface.

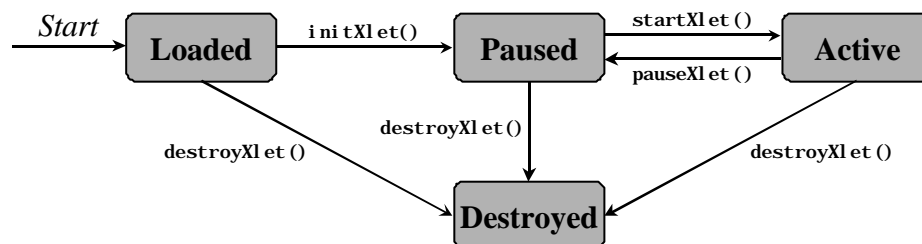


Figura 22. Ciclo de vida de um *Xlet*.

Para gerenciar o ciclo de vida das aplicações (*Xlets*), Java TV define o conceito de um gerente de aplicação (*application manager*). O estado de um *Xlet* pode ser mudado pelo gerente de aplicação ou pelo próprio *Xlet*. Para tal, métodos da interface *Xlet* devem ser ativados pelo gerente de aplicação ou pelo próprio *Xlet*. Neste último caso, o próprio *Xlet* notifica o gerente de aplicação sobre a transição de estado via um mecanismo de *callback*, que é configurado durante o processo de inicialização do *Xlet*. Desta forma, o estado de um *Xlet* é sempre conhecido pelo gerente de aplicação.

Inicialmente, o *Xlet* é instanciado pelo gerente de aplicação usando o método *new*. Após a instanciação, o *Xlet* encontra-se no estado *loaded*. Em seguida, o *Xlet* pode ser inicializado pelo gerente de aplicação usando o método *initXlet*. No processo de inicialização, o gerente de aplicação passa para o *Xlet* um objeto *XletContext* que define o contexto de execução do *Xlet*. Através deste objeto, o *Xlet* pode obter propriedades do ambiente de execução e notificar o gerente de aplicação sobre mudanças de estados via o mecanismo de *callback*.



Após a inicialização, o *Xlet* encontra-se no estado *paused*. Neste estado, o *Xlet* não pode manter ou usar nenhum recurso compartilhado. O *Xlet* no estado *paused* pode ser ativado usando o método *startXlet*. Após a ativação, o *Xlet* encontra-se no estado *active*. Neste estado, o *Xlet* ativa suas funcionalidades e provê seus serviços. O *Xlet* no estado *active* pode voltar ao estado *paused* usando o método *pauseXlet*. Em qualquer estado, um *Xlet* pode ser destruído usando o método *destroyXlet*. Após ser destruído, o *Xlet* libera todos os recursos e finaliza a execução.

Java TV tem sido amplamente adotado por organizações de padronização, tornando-o um forte candidato a padrão mundial para conteúdo de televisão digital interativa. Por exemplo, diversas implementações de *middleware* adotam o modelo Java TV, com ligeiras diferenças entre si.

## **5.2. Padrões de Middleware para TVDI**

### **5.2.1. DVB**

Na camada de *middleware*, o padrão DVB adota o MHP, cuja especificação é denominada DVB-MHP (*Digital Video Broadcasting – Multimedia Home Platform*) (ETSI, 2003c). A plataforma MHP começou a ser especificada pelo projeto DVB em 1997. No entanto, a primeira versão (MHP 1.0) foi oficialmente lançada em junho de 2000. Após um ano do lançamento da primeira versão, em junho de 2001, foi lançada uma nova especificação (MHP 1.1). Em junho de 2003, foi lançada a versão 1.1.1 do MHP.

O MHP define uma interface genérica entre as aplicações e o *set-top box* (hardware e sistema operacional), no qual as aplicações são executadas. Além disso, o MHP define o modelo e o ciclo de vida das aplicações, como também os protocolos e os mecanismos distribuição de dados em ambientes de televisão pseudo-interativa e interativa.

Nas versões 1.1 e 1.1.1, o MHP provê funcionalidades adicionais em relação à versão inicial, incluindo, por exemplo, a possibilidade de carregar programas interativos através do canal de retorno e o suporte opcional a aplicações desenvolvidas usando uma linguagem declarativa.

A partir da versão 1.1, o MHP adota modelos de aplicações baseados em linguagens procedural e declarativa. No modelo procedural, o MHP suporta a execução de aplicações Java TV, denominadas DVB-J. No modelo declarativo, opcionalmente, o MHP suporta a execução de aplicações desenvolvidas com tecnologias relacionadas à linguagem HTML, denominadas DVB-HTML.

### **5.2.2. DASE/ATSC**

Na camada de *middleware*, o padrão ATSC adota o DASE (*DTV Application Software Environment*) (ATSC, 2003), definindo uma camada de software que permite a programação de conteúdo e aplicações. O DASE adota modelos de aplicações baseados em linguagens procedural e declarativa. No modelo procedural, o DASE suporta a execução de aplicações Java TV. No modelo declarativo, o DASE suporta a execução de aplicações desenvolvidas em uma versão estendida da linguagem HTML.

### 5.2.3. ARIB/ISDB

Na camada de *middleware*, o padrão ISDB adota a plataforma padronizada pelo ARIB (*Association of Radio Industries and Businesses*) (ARIB, 2002), definindo uma camada de software que permite a programação de conteúdo e aplicações. O ARIB adota um modelo de aplicação baseado na linguagem declarativa denominada BML (*Broadcast Markup Language*), que é baseada na linguagem XML (*Extensible Markup Language*).

## 6. Prática em Desenvolvimento de Aplicações para TVDI

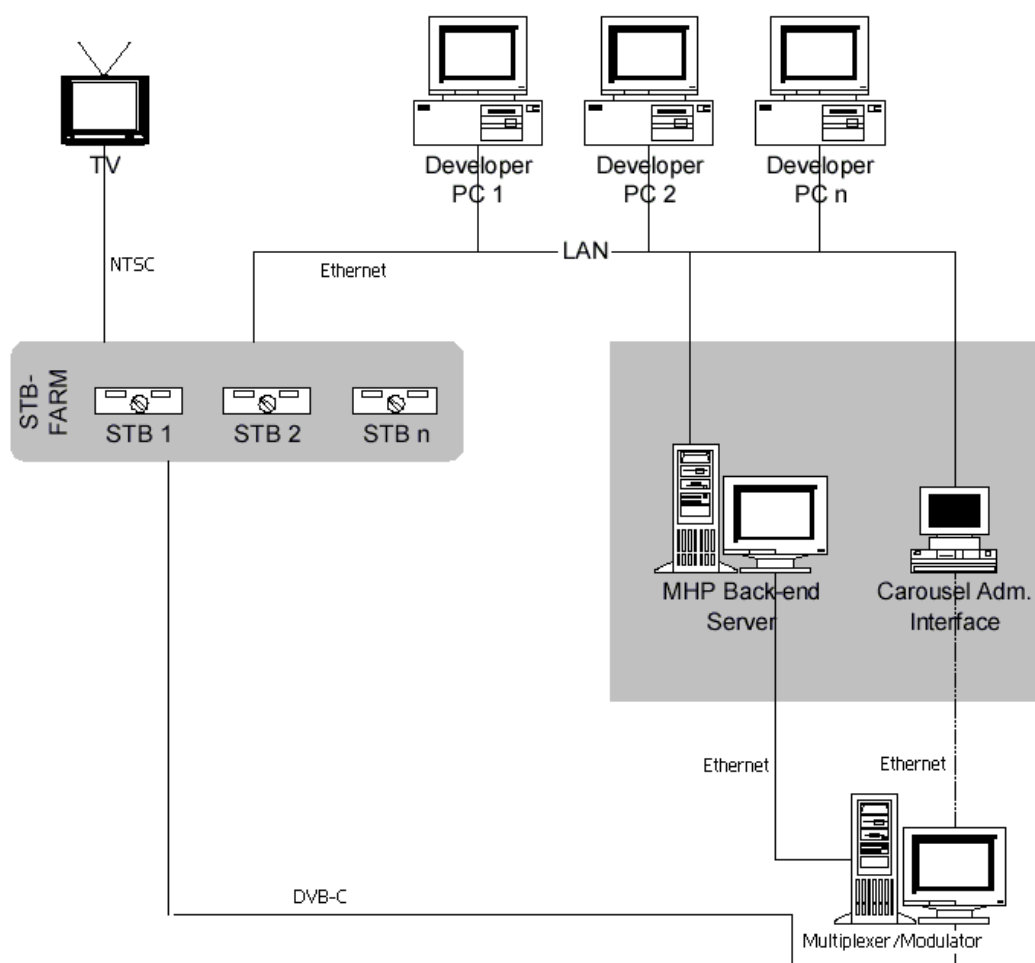
Esta seção apresenta um breve tutorial sobre desenvolvimento de aplicações para sistemas de TVDI, com ênfase nas características do modelo DVB-J/MHP. São apresentados elementos fundamentais para montagem de plataformas para desenvolvimento e teste de Xlets. O ciclo de desenvolvimento de aplicações será apresentado, com identificação dos passos e competências necessárias. Por fim, uma pequena aplicação será desenvolvida, integrada e executada usando a uma plataforma de testes baseada em emuladores MHP. Serão explorados o uso de componentes visuais (widgets), o tratamento de eventos do controle remoto, e o acesso aos arquivos do carrossel.

### 6.1. Uma plataforma de desenvolvimento de Software para TVDI

A Figura 23 apresenta um ambiente para desenvolvimento de software na plataforma MHP, que pode ser aplicada ao desenvolvimento em equipe (Frolich, 2002). Na plataforma apresentada destaca-se a presença de computadores PC 1, PC2,..., PCn, que são estações de trabalho contendo o ambiente usado para edição, compilação e emulação dos Xlets. Os STBs são conectáveis a um ou mais monitores de TV e servem para testes do funcionamento dos Xlets em condições próximas às reais. O computador Carousel Adm Interface é usado para configurar os arquivos do Carrossel, entre os quais estão as aplicações DVB-J que são produzidas nos PCs. O computador Multiplexer/Modulator realiza as funções de: (1) multiplexador de streams (incluindo o stream carrossel) e (2) modulador do sinal que será difundido para os STBs. O esquema de modulação usado depende do tipo de interface com o STB, onde o mais simples é fazer modulação DVB-C (cabo). Para que o A/V possa ser usado durante os testes é necessário enviar ao multiplexador um stream com o que deve ser apresentado, através da instalação de um streamer, que pode estar na mesma máquina onde está o Gerador de Carrossel. O canal de retorno é fornecido através de uma rede Ethernet, onde o MHP Back-End Server atua como o Provedor de Serviço.

O ciclo de desenvolvimento nesta plataforma é composto por seis passos:

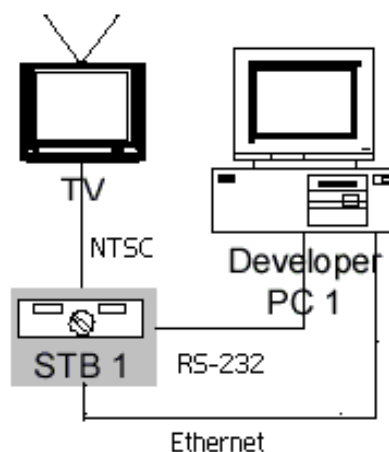
1. Edição e compilação dos Xlets no PC;
2. Emulação e teste inicial do Xlets no PC;
3. Carga do Xlet no Gerador de Carrossel;
4. Sintonização do STB no canal onde está sendo transmitido o Carrossel;
5. Carga manual ou automática (auto-start) do Xlet no STB;
6. Teste e coleta de dados sobre funcionamento do Xlet executando no STB.



**Figura 23. Ambiente para Desenvolvimento MHP em Equipe (Frolich, 2002).**

## **6.2. Uma plataforma pessoal para desenvolvimento DVB-J/MHP.**

A Figura 24 apresenta uma plataforma simplificada para desenvolvimento DVB-J/MHP, composta basicamente por três elementos de hardware: (1) TV, (2) PC e (3) STB. O PC é usado para codificação e emulação de Xlets. A carga de Xlets é feita manualmente, com o uso de um software loader que carrega o Xlet no STB através de um cabo serial. O canal de retorno do STB é feito através da saída Ethernet. Alguns STBs para desenvolvimento são comercializados pela ADB Global (ADB, 2004).



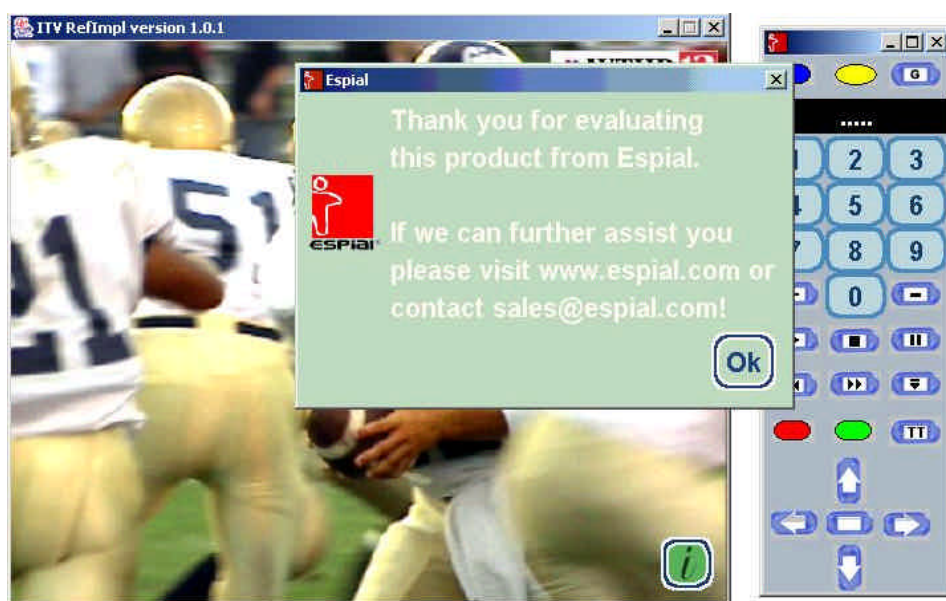
**Figura 24. Plataforma individual para Desenvolvimento DVB-J/MHP.**

### **6.3. Uso de Emuladores**

Caso não se tenha acesso a um STB ainda assim é possível realizar-se a prototipação de aplicações DVB-J, através do uso de emuladores. Emuladores permitem montar um ambiente para aprendizagem introdutória ao desenvolvimento de aplicações DVB-J. É importante ressaltar que o funcionamento de aplicações DVB-J em emuladores apresenta diferenças significativas no que se refere a desempenho e interface com o usuário. Tendo estas limitações em mente, são apresentados os passos necessários para iniciar o desenvolvimento de aplicações para TVDI.

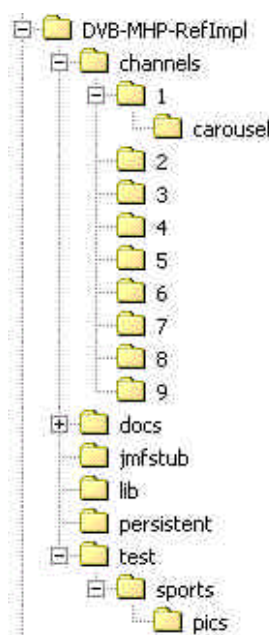
### **6.4. O Emulador de Xlets da Espial**

Para emular os programas aqui apresentados será utilizado o Espial's iTV reference platform, disponível em (Espial, 2002). O emulador Espial contém um subconjunto mínimo do MHP, suficiente para a construção de Xlets simples. Uma versão de avaliação do emulador é fornecida sem custo. Para que o emulador funcione é preciso instalar o J2SDK (Java Development Kit). Após o download e descompactação dos arquivos do emulador, pode-se executar o arquivo runit.bat, que está na pasta onde foi descompactado o emulador. Aparecerá na tela um conjunto de três janelas como na Figura 25.



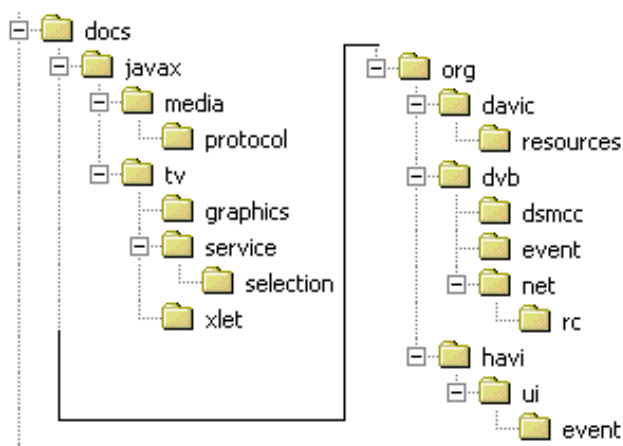
**Figura 25. Aspecto do Emulador Espial's iTV reference platform.**

A tela de título Espial, que aparece na frente das demais, é executada sempre que o emulador é iniciado, e pode ser fechada ao pressionar do mouse no botão Ok. A tela que aparece à direita da figura emula o funcionamento do controle remoto. Observe que não existe teclado, e que em geral o usuário espera usar apenas as setas de navegação na região inferior do controle remoto para interagir com a aplicação. A tela à esquerda emula o funcionamento da tela de TV, e será chamada de Monitor. Apenas imagens estáticas são apresentadas no Monitor, pois o objetivo é emular apenas o comportamento das aplicações DVB-J, cuja interface em geral será sobreposta ao 'vídeo' emulado. No canto inferior direito do monitor existe um pequeno botão com um símbolo 'i', que emula o botão de ativação manual da interação, que aparece quando há um Xlet disponível no carrossel.



**Figura 26. Sistema de Arquivos do Emulador Espial.**

A plataforma Espial permite a emulação da carga e execução de Xlets e bem como dos arquivos no Carrossel. A estrutura de arquivos do emulador é apresentada na Figura 26. A pasta channels emula as informações associadas aos canais (serviços) que são sintonizados pelo usuário, em número de 1 a 9. Na pasta de cada serviço existe um arquivo channel.properties, em formato texto, no qual estão definidos os valores de dois atributos: TVImage=<IMAGE>.jpg e Xlet=<XLET>. <IMAGE> é o nome do arquivo que contém a imagem a ser apresentada quando o serviço for selecionado, e <XLET> é o nome do Xlet a ser executado automaticamente quando o usuário selecionar o serviço. Em um STB real o Xlet só é executado automaticamente se o Xlet e STB estiverem configurados no modo autostart. A pasta carousel, contida na pasta de cada um dos serviços, emula os arquivos que são distribuídos no carrossel. A pasta docs contém javadocs da biblioteca MHP, apresentados na Figura 27.



**Figura 27. Packages do subconjunto MHP disponíveis no Emulador Espial.**

Na Figura 27, as pastas jmfstub e lib contêm as bibliotecas de apoio à compilação e emulação MHP. A pasta persistent emula o sistema de arquivos persistente do STB, possivelmente armazenado numa memória flash ou em um HD do STB real. Em um STB real, os Xlets também são distribuídos através da pasta carrossel, mas neste emulador eles são montados a partir da área raiz do sistema de arquivos do emulador. Na pasta test/sports existe um Xlet de nome test.sports.SportsXlet pode ser associado a qualquer serviço.

#### 6.4. Um Exemplo de Xlet: QuizXlet.

A Figura 28 apresenta o aspecto visual do QuizXlet, um protótipo de aplicação DVB-J que faz perguntas e apresenta respostas ao usuário, estimulando sua interação com a TV.



Figura 2 Aspecto Visual do QuizXlet executado no emulador Espial.

O QuizXlet é formado por duas classes: QuizXlet e QuizPanel, e importa várias outras classes, apresentadas no diagrama da Figura 29. O código completo das classes que forma o QuizXlet é apresentado a seguir, e para facilitar o entendimento do mesmo, as linhas que realizam as funções descritas no texto são indicadas entre chaves.

QuizXlet implementa a interface Xlet {28}, e deste modo se obriga a implementar os métodos initXlet() {51}, startXlet() {75}, pauseXlet() {89} e destroyXlet() {93}. Adicionalmente, o QuizXlet é responsável por obter as perguntas e respostas que serão mostradas ao usuário através do object carrossel. Para tal, QuizXlet implementa a interface Asynchronous Loading Event Listener {28}, e deste modo pode ser notificado da carga do arquivo no carrossel através do método receiveEvent() {97}. Este último método recebe como argumento um Asynchronous Loading Event, que contém referência a um DSMCCObject {99}, que é um arquivo (File). Um FileInputStream {104} pode ser usado para ler o DSMCCObject. QuizXlet também é responsável por obter espaço na tela de TV, para apresentar a GUI. Este espaço é definido por um objeto HAVi do tipo HScene. Um Xlet obtém uma HScene {68} através da chamada de um método de HSceneFactory.

A fábrica de cenas **{66}** precisa de um conjunto de parâmetros para formatação para a cena, que estão definidos em HSceneTemplate **{55}**, este último contendo principalmente a posição (HScreenPoint **{56}**) e tamanho (HScreenDimension **{57}**) da cena na tela de TV. A HScene obtida herda de java.awt.Container, que por sua vez herda de java.awt.Component. Deste modo, a uma cena podem ser adicionados **{127}** outros componentes AWT, como o QuizPanel **{125}**. Para organizar o QuizPanel no centro da HScene, QuizXlet utiliza um BorderLayout **{70}**.

QuizPanel é um HAVi Container, e apresenta dois componentes visuais (widgets) que desempenham funções de Label e Button, e que são ELabel **{54}** e EButton **{57}**, respectivamente. Estes widgets, produzidos pela Espial, não pertencem ao padrão MHP. Qualquer STB DVB/MHP real contém uma versão completa do middleware MHP, que contém componentes padronizados HAVi.





```

1  /*
2  * QuizXlet - v 1.0 - 25/05/2004
3  * Xlet de perguntas e respostas que funciona em plataforma de TVDI
4  * Copyright (c) 2004, por Jorge Henrique Cabral Fernandes (jhcf@cic.unb.br)
5  * Uso livre para fins educacionais e não lucrativos, desde que
6  * sejam mantidas intactas as informações originais de autoria
7  */
8  package br.unb.cic.introtvdi;
9  // Para leitura dos arquivos do carrossel.
10 import java.io.FileInputStream;
11 import java.io.IOException;
12 // gerenciador de layout awt
13 import java.awt.BorderLayout;
14 // xlet
15 import javax.tv.xlet.Xlet;
16 import javax.tv.xlet.XletContext;
17 // para obter arquivos do carrossel
18 import org.dvb.dsmcc.AsynchronousLoadingEvent;
19 import org.dvb.dsmcc.AsynchronousLoadingEventListener;
20 import org.dvb.dsmcc.DSMCCObject;
21 import org.dvb.dsmcc.InvalidPathNameException;
22 // para obter espaço na tela de TV
23 import org.havi.ui.HScene;
24 import org.havi.ui.HSceneFactory;
25 import org.havi.ui.HSceneTemplate;
26 import org.havi.ui.HScreenDimension;
27 import org.havi.ui.HScreenPoint;
28 public class QuizXlet implements Xlet, AsynchronousLoadingEventListener {
29     // questões e respostas padrão do QuizXlet
30     private String defaultQ = "Desafio";
31     private String defaultA = "Aguarde...";
32     private String answer = null;
33     private String question = null;
34     //Cena sobre a qual será apresentada o Xlet.
35     private HScene hScene;
36     //Painel que contém a GUI do Xlet
37     private QuizPanel panel;
38     //Context que é passado para o Xlet na inicialização
39     private XletContext xletContext;
40     // Fábrica de cenas
41     HSceneFactory factory;
42     // Template para definir o formato da cena desejada pelo Xlet
43     HSceneTemplate hSceneTemplate;
44     // Localização da cena na tela
45     HScreenPoint hScreenPoint;
46     // Dimensão da cena solicitada
47     HScreenDimension hScreenDimension;
48     // Layout manager
49     BorderLayout layout;
50     //Chamado para inicializar o Xlet, uma única vez
51     public void initXlet(XletContext context) {
52         System.out.println("init QuizXlet (" + context + ")");
53         xletContext = context;
54         //Define um template que será usado para pedir a cena à fábrica
55         hSceneTemplate = new HSceneTemplate();

```

```

56     hScreenPoint = new HScreenPoint((float) 0.0, (float) 0.0);
57     hScreenDimension = new HScreenDimension((float) 1.0, (float) 1.0);
58     hSceneTemplate.setPreference(
59         HSceneTemplate.SCENE_SCREEN_DIMENSION,
60         hScreenDimension,
61         HSceneTemplate.REQUIRED);
62     hSceneTemplate.setPreference(
63         HSceneTemplate.SCENE_SCREEN_LOCATION,
64         hScreenPoint,
65         HSceneTemplate.REQUIRED);
66     factory = HSceneFactory.getInstance();
67     // obtém uma cena
68     hScene = factory.getBestScene(hSceneTemplate);
69     // Configura o layout manager para a cena
70     hScene.setLayout(new BorderLayout());
71     // insere o painel quiz na cena
72     changePanel(defaultQ, defaultA);
73 }
74 // Coloca o Xlet em estado ativo
75 public void startXlet() {
76     System.out.println("start QuizXlet");
77     // busca informações no carrossel
78     try {
79         System.out.println("Requesting file");
80         DSMCCObject carouselFile = new DSMCCObject("msgfile");
81         // solicita notificação assíncrona quando o arquivo estiver disponível
82         carouselFile.asynchronousLoad(this);
83     }
84     catch (InvalidPathNameException e) {
85         System.out.println("Invalid path name!");
86     }
87 }
88 // Chamado pelo app manager quando o Xlet precisa parar
89 public void pauseXlet() {
90     System.out.println("pause QuizXlet");
91 }
92 // Chamado pelo app manager quando o Xlet vai ser removido da memória
93 public void destroyXlet(boolean unconditional) {
94     System.out.println("destroy QuizXlet (" + unconditional + ")");
95 }
96 // recebe evento do DSMCC controller
97 public void receiveEvent(AsynchronousLoadingEvent ale) {
98     // verifica se é um arquivo que foi encontrado no carrossel
99     DSMCCObject file = (DSMCCObject) ale.getSource();
100     try {
101         // abre o arquivo e lê as perguntas e respostas
102         FileInputStream fis = new FileInputStream(file);
103         byte[] buffer = new byte[100];
104         fis.read(buffer);
105         String quiz = new String(buffer);
106         int idx1 = quiz.indexOf("-", 0);
107         int idx2 = quiz.indexOf("-", idx1 + 1);
108         question = new String(buffer, 0, idx1);
109         answer = new String(buffer, idx1 + 1, idx2 - idx1 - 1);
110         // muda o painel na GUI
111         changePanel(question, answer);

```

```

112     }
113     catch (IOException ioe) {
114         System.out.println("Error reading file in carrossel: " + ioe);
115     }
116 }
117 // modifica o painel apresentado
118 private void changePanel(String q, String a) {
119     // remove todos os componentes de GUI da cena atual
120     if (panel != null) {
121         panel.release();
122         hScene.remove(panel);
123     }
124     // Cria um QuizPanel, com perguntas e respostas específicas
125     panel = new QuizPanel(q, a, 10, 10, 500, 200);
126     // insere o novo componente de GUI na cena
127     hScene.add(panel);
128     // Prepara e apresenta a cena no monitor de TV
129     hScene.validate();
130     hScene.repaint();
131     hScene.setVisible(true);
132 }
133 }

```

```

1  /*
2   * QuizPanel.java - V 1.0 - 25/05/2004
3   * Painel que apresenta uma pergunta e mostra a resposta correspondente,
4   * caso o usuário pressione o botão <enter> do controle remoto.
5   * Copyright (c) 2004, por Jorge Henrique Cabral Fernandes (jhcf@cic.unb.br)
6   * Uso livre para fins educacionais e não lucrativos, desde que
7   * sejam mantidas intactas as informações originais de autoria.
8   */
9  package br.unb.cic.introtvdi;
10 // gerenciador de layout awt
11 import java.awt.BorderLayout;
12 // interação do usuário com o Xlet através do controle remoto
13 import org.havi.ui.event.HRcEvent;
14 import org.dvb.event.EventManager;
15 import org.dvb.event.UserEvent;
16 import org.dvb.event.UserEventListener;
17 import org.dvb.event.UserEventRepository;
18 // o QuizPanel é um container
19 import org.havi.ui.HContainer;
20 // os widgets são da Espial
21 import espial.awt.EButton;
22 import espial.awt.ELabel;
23 import espial.awt.EPanel;
24 public class QuizPanel extends HContainer implements UserEventListener {
25     // Contém o conjunto de eventos que serão tratados pelo Xlet.
26     private UserEventRepository userEventRepository;
27     private EventManager eventManager; // Gerenciador de eventos
28     private EPanel showPanel; // sub-painel
29     private EButton showButton; // botão
30     private ELabel showLabel; // rótulo
31     private BorderLayout layout1, layout2; // gerenciadores de layout
32     private String question; // pergunta apresentada ao usuário

```

```

33     private String answer; // resposta
34     // cria um painel de perguntas e respostas
35     public QuizPanel(String q, String a, int x, int y, int w, int h) {
36         super(x, y, w, h);
37         this.question = q;
38         this.answer = a;
39         //Define inputs do usuário que serão tratados
40         userEventRepository = new UserEventRepository("EventsOfInterest");
41         userEventRepository.addAllArrowKeys();
42         userEventRepository.addKey(HRcEvent.VK_ENTER);
43         eventManager = EventManager.getInstance();
44         eventManager.addUserEventListener(this, userEventRepository);
45         // cria um layout manager associado ao painel
46         layout1 = new BorderLayout();
47         this.setLayout(layout1);
48         // Cria um sub-painel que conterá os componentes apresentados
49         showPanel = new EPanel();
50         layout2 = new BorderLayout();
51         showPanel.setLayout(layout2);
52         this.add(showPanel, BorderLayout.CENTER);
53         //Cria um botão, que ao ser ativado mostra a mensagem
54         showButton = new EButton(question);
55         showPanel.add(showButton, BorderLayout.NORTH);
56         // Cria o label que vai conter a resposta.
57         showLabel = new ELabel("?");
58         showLabel.setCentered(true);
59         showPanel.add(showLabel, BorderLayout.CENTER);
60         // inicia com foco no botão
61         showButton.requestFocus();
62     } //end constructor
63     // remove tratamento de eventos a partir deste componente
64     public void release() {
65         eventManager = EventManager.getInstance();
66         eventManager.removeUserEventListener(this);
67     }
68     // Trata eventos do controle remoto
69     public void userEventReceived(UserEvent e) {
70         if (e.getCode() == HRcEvent.VK_ENTER) {
71             showLabel.setLabel(answer);
72             repaint();
73         }
74     }
75 }

```

## 7. Conclusões e Agradecimentos

Este capítulo apresentou uma introdução à Televisão Digital Interativa (TVDI), que esperamos tenha servido para introduzir estudantes e profissionais de computação na área. Muito ainda há que ser aprendido nesta área na qual a ciência da computação e o desenvolvimento de software foram inseridos recentemente.

O trabalho aqui apresentado foi resultado da vivência prática e teórica pela qual os autores passaram durante a realização de vários projetos de pesquisa e desenvolvimento da área de Vídeo e TV Digital, nos Laboratórios LAVID-DI-UFPB e Natalnet-DIMap-UFRN. Gostaríamos de externar nossos agradecimentos a todos os professores, alunos e

técnicos que contribuíram para que o conhecimento aqui apresentado fosse compreendido, especialmente as contribuições de Luiz Eduardo Leite, Milano Gadelha e Jorge Washington de Sousa. Todos os erros e omissões são de inteira responsabilidade dos autores.

Finalmente os autores gostariam de indicar o apoio financeiro e logístico das seguintes instituições e projetos, sem o qual não teria sido possível a realização deste trabalho: Universidade Federal da Paraíba – Departamento de Informática; Universidade Federal do Rio Grande do Norte – Departamento de Informática e Matemática Aplicada; Universidade de Brasília – Departamento de Ciência da Computação; Projeto HITV para Desenvolvimento de Hardware e Software para Sistemas de Televisão Digital de alta Definição, financiado pela FINEp; Projeto BLNet – de Acesso em Banda Larga à Internet em Plataforma de TV Digital, financiado pelo CPqD; Projeto I2TV - Infra-estrutura Internet2 para Desenvolvimento e Teste de Programas e Ferramentas para TV Interativa, financiado pelo CNPq; e Projeto GT de Vídeo Digital, financiado pela RNP.

## **Referências**

- ADB (2004). ADB Advanced Digital Broadcast. 2004. <http://www.adbglobal.com>.
- ARIB. (2002) “ARIB STD B24 - ARIB Standard: Data Coding and Transmission Specification for Digital Broadcasting”, Revision 3.2. 2002.
- ATSC. (2001). “Standard A/52A - Digital Audio Compression (AC-3) Standard”, Revision A.
- ATSC. (2003) “Standard A/100 - DTV Application Software Environment – Level 1 (DASE-1)”.
- ATSC. (2004) “Advanced Television Systems Committee”, [www.atsc.org](http://www.atsc.org), acessado em junho de 2004.
- Broadcast Papers. (2004a) “Asset Management Papers”. Disponível URL: <http://www.broadcastpapers.com/asset/asset.htm>. Acessado em junho de 2004.
- Broadcast Papers. (2004b) “Editing and Post-Production Management Papers”. Disponível URL: <http://www.broadcastpapers.com/editing/editing.htm>. Acessado em junho de 2004.
- Broadcast Papers. (2004c) “TV Transmission Papers”. Disponível URL: <http://www.broadcastpapers.com/tvtran/tvtran.htm>. Acessado em junho de 2004.
- Broadcast Papers. (2004d) “Interactive TV and Datacasting Papers”. Disponível URL: <http://www.broadcastpapers.com/data/data.htm>. Acessado em junho de 2004.
- CPqD. (2001). “Relatório Integrador dos Aspectos Técnicos e Mercadológicos da Televisão Digital”. Anatel. Março de 2001.
- DAVIC. (1999) “DAVIC 1.4.1 Specification Part 9: Information Representation”.
- DiBEG. (2004) “Digital Broadcasting Experts Group”, [www.dibeg.org](http://www.dibeg.org), acessado em junho de 2004.
- DVB. (2004) “Digital Video Broadcasting Project”, [www.dvb.org](http://www.dvb.org), acessado em junho de 2004.

- Espial. (2002) "iTV Development Platform". <http://www.devicetop.com/dt/tutorials/DVB-MHP-RefImpl.ZIP>. Acessado em junho de 2004.
- ETSI. (1997a) "ETSI EN 300 421 - Digital Video Broadcasting (DVB): Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services", Edition 1.1.2, August 1997.
- ETSI (1997b) "ETSI EN 300 748, "Digital Video Broadcasting (DVB): Multipoint Video Distribution Systems (MVDS) at 10 GHz and above", Edition 1.1.2, August 1997.
- ETSI. (1997c). "ETSI EN 300 749 - Digital Video Broadcasting (DVB): Microwave Multipoint Distribution Systems (MMDS) below 10 GHz", Edition 1.1.2, August 1997.
- ETSI. (1998) "ETSI EN 300 429 - Digital Video Broadcasting (DVB): Framing Structure, Channel Coding and Modulation for Cable Systems", Edition 1.2.1, April 1999
- ETSI. (2001) "ETSI EN 300 744 - Digital Video Broadcasting (DVB): Framing Structure, Channel Coding and Modulation for Digital Terrestrial Television", Edition 1.4.1, January 2001.
- ETSI (2003b) "ETSI EN 301 192 - Specification for Data Broadcast". Edition 1.3.1. May 2003.
- ETSI (2003c) "ETSI TS 102 812 - Digital Video Broadcasting (DVB) – Multimedia Home Platform (MHP) Specification 1.1.1", June 2003.
- Frolich. (2002). "Establishment of MHP application development and play-out seen from a public broadcasters point of view". IBC 2002 Conference.
- HAVi (2001). "HAVi v1.1 - Home Audio Video Interoperability Version 1.1", [www.havi.org](http://www.havi.org), acessado em junho de 2004.
- IEEE (1995). "IEEE1394 - Standard for a High Performance Serial Bus", IEEE, 1995.
- ISDB. (1998) "ISDB-T - Terrestrial Integrated Services Digital Broadcasting (ISDB-T): Specification of Channel Coding, Framing Structure and Modulation", September 1998
- ISO. (1996a). "ISO/IEC 13818-1 - Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 1: Systems (MPEG-2 Systems)", 1996.
- ISO. (1996b). "ISO/IEC 13818-2 - Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video (MPEG-2 Video)", 1996.
- ISO. (1997). "ISO/IEC 13818-7 - Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Advanced Audio Coding", 1997.
- ISO. (1998a). "ISO/IEC 13818-3 - Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 3: Audio (MPEG-2 Audio)", Second Edition, 1998

- ISO. (1998b). "ISO/IEC 13818-6 - Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Extensions for Digital Storage Media Command and Control". 199
- Sparano, D. "What Exactly is 8-VSB Anyway?", The Guide to Digital Television, 3<sup>rd</sup> edition, Silbergleid & Pescatore org, Miller Freeman Inc, March 2000.
- Sun. (1999) "JMF 2.0 - Java Media Framework 2.0 API Guide", Sun Microsystems November 1999.
- Sun. (2000). "Java TV 1.0 - Java TV API Technical Overview: The Java TV API White Paper", Version 1.0, Sun Microsystems. November 2000.
- Sun. (2002) "JSR62 - J2ME Personal Profile Specification 1.0", Final Release, September 2002.
- Lindholm, T. and Yellin, F. (1999), "The Java Virtual Machine Specification", Second Edition, Addison-Wesley, 1999.
- Tektronix. (2002) "A Guide to MPEG Fundamentals and Protocol Analysis (Including DVB and ATSC)". Disponível URL: <http://www.tektronix.com>.