



UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO



Especificação de um Mecanismo de Integração entre o Moodle e uma Aplicação de TV Digital Interativa

Paula Marin Leite

Juiz de Fora
Dezembro de 2009

Especificação de um Mecanismo de Integração entre o Moodle e uma Aplicação de TV Digital Interativa

Paula Marin Leite

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Orientador: Eduardo Barrére

Juiz de Fora
Dezembro de 2009

ESPECIFICAÇÃO DE UM MECANISMO DE INTEGRAÇÃO ENTRE O MOODLE E
UMA APLICAÇÃO DE TV DIGITAL INTERATIVA

PAULA MARIN LEITE

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA COMO PARTE
INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Eduardo Barrére, DSc (orientador)

Daves Márcio Silva Martins, MSc

Fernanda Cláudia Alves Campos, DSc

Juiz de Fora,
Dezembro de 2009

Sumário

1	Introdução	1
2	Educação a Distância	2
2.1	Breve histórico da Educação a Distância no Brasil	2
2.2	O AVA Moodle	3
3	TV Digital	6
3.1	Arquitetura da TV Digital	7
3.2	Breve histórico da TV Digital no Brasil	9
4	O modelo de integração	11
4.1	A linguagem XML	13
5	O processo de comunicação	15
5.1	<i>Login</i> do usuário	15
5.2	Escolha do curso	18
5.3	Escolha da atividade Chats	20
5.4	Escolha da atividade Fóruns	24
5.5	Escolha da atividade Glossários	29
5.6	Escolha da atividade Recursos Materiais	33
5.7	Escolha da atividade Wikis	38
5.8	Escolha da função Perfil	41
5.9	Escolha da função Participantes	42
5.10	Escolha da função Mensagens	46
5.11	Escolha da função Nota	47
6	Conclusão	50
	REFERÊNCIAS BIBLIOGRÁFICAS	52

Índice de Figuras

Figura 1: Arquitetura da TV Digital. Fonte: (CPqD, 2006) -----	7
Figura 2: Previsão de implantação da TV Digital. Fonte: (DTV, 2009) -----	10
Figura 3: Ambiente de integração entre o Moodle e a TVDI (BARRÉRE-a, 2009) -----	11
Figura 4: XML para requisição de <i>login</i> -----	15
Figura 5: XML de resposta de <i>login</i> -----	16
Figura 6: <i>Query</i> para validação de <i>login</i> -----	17
Figura 7: <i>Query</i> para relação de cursos -----	17
Figura 8: XML para requisição de curso -----	18
Figura 9: XML de resposta com as funcionalidades disponíveis para o curso -----	19
Figura 10: <i>Query</i> para listagem das atividades disponibilizadas para o curso -----	20
Figura 11: XML para requisição da atividade chats -----	20
Figura 12: XML de resposta com os chats disponíveis no curso -----	21
Figura 13: XML para requisição de um chat -----	22
Figura 14: XML de resposta contendo histórico do chat escolhido (Parte I) -----	22
Figura 15: XML de resposta contendo histórico do chat escolhido (Parte II) -----	23
Figura 16: <i>Query</i> para listagem dos chats de um curso -----	23
Figura 17: <i>Query</i> para exibição do histórico do chat escolhido -----	23
Figura 18: XML para requisição da atividade fóruns -----	24
Figura 19: XML de resposta com os fóruns disponíveis no curso -----	25
Figura 20: XML para requisição de um fórum -----	26
Figura 21: XML de resposta contendo as discussões do fórum escolhido (Parte I) -----	26
Figura 22: XML de resposta contendo as discussões do fórum escolhido (Parte II) -----	27
Figura 23: XML para requisição de uma discussão -----	27
Figura 24: XML de resposta contendo as postagens da discussão escolhida -----	28
Figura 25: <i>Query</i> para listagem dos fóruns de um curso -----	29
Figura 26: <i>Query</i> para exibição das discussões do fórum escolhido -----	29
Figura 27: <i>Query</i> para exibição das postagens da discussão escolhida -----	29
Figura 28: XML para requisição da atividade glossários -----	30
Figura 29: XML de resposta com os glossários disponíveis no curso (Parte I) -----	30
Figura 30: XML de resposta com os glossários disponíveis no curso (Parte II) -----	31

Figura 31: XML para requisição de um glossário -----	31
Figura 32: XML de resposta contendo os termos do glossário escolhido -----	32
Figura 33: <i>Query</i> para listagem dos glossários de um curso -----	32
Figura 34: <i>Query</i> para exibição dos itens do glossário escolhido -----	33
Figura 35: XML para requisição da atividade recursos -----	33
Figura 36: XML de resposta com os recursos disponíveis no curso -----	34
Figura 37: XML para requisição de um recurso -----	35
Figura 38: XML de resposta contendo o recurso escolhido (texto simples ou página web) --	36
Figura 39: XML de resposta contendo o recurso escolhido (arquivo ou <i>link</i> para web) -----	37
Figura 40: <i>Query</i> para listagem dos recursos de um curso -----	37
Figura 41: <i>Query</i> para seleção de recurso (texto simples ou página web) -----	37
Figura 42: <i>Query</i> para seleção de recurso (arquivo ou <i>link</i> para web) -----	38
Figura 43: XML para requisição da atividade wikis -----	38
Figura 44: XML de resposta com as wikis disponíveis no curso -----	39
Figura 45: XML para requisição de uma wiki -----	40
Figura 46: XML de resposta contendo a wiki escolhida (Parte I) -----	40
Figura 47: XML de resposta contendo a wiki escolhida (Parte II) -----	41
Figura 48: <i>Query</i> para listagem das wikis de um curso -----	41
Figura 49: <i>Query</i> para seleção da wiki -----	41
Figura 50: XML para requisição da função perfil -----	42
Figura 51: XML de resposta com a função perfil (Parte I) -----	42
Figura 52: XML de resposta com a função perfil (Parte II) -----	43
Figura 53: <i>Query</i> para função perfil -----	44
Figura 54: <i>Query</i> para exibição das funções -----	44
Figura 55: <i>Query</i> para relação de cursos (adaptada) -----	44
Figura 56: XML para requisição da função participantes -----	44
Figura 57: XML de resposta com os participantes do curso -----	45
Figura 58: <i>Query</i> para listagem dos participantes -----	46
Figura 59: XML para requisição da função mensagens -----	46
Figura 60: XML de resposta com as mensagens -----	47
Figura 61: <i>Query</i> para listagem das mensagens -----	47
Figura 62: XML para requisição da função nota -----	48

Figura 63: XML de resposta com as notas (Parte I) -----	48
Figura 64: XML de resposta com as notas (Parte II) -----	49
Figura 65: <i>Query</i> para listagem das notas -----	49

Índice de Tabelas

Tabela 1 – Atividades do Moodle -----	4
Tabela 2 – Funcionalidades do Moodle integradas à TVDI -----	12

RESUMO

Este trabalho propõe a especificação de um mecanismo de integração entre uma aplicação web para Educação a Distância, o Moodle, a outra de TV Digital. De início foram estudadas as funções do Moodle e identificadas aquelas que seriam exibidas na TV Digital, levando em consideração as restrições de interatividade das *set-top boxes*. Posteriormente, teve início a criação das *queries* de consulta ao banco de dados do Moodle e a definição dos arquivos XML com seus respectivos DTD. Os arquivos XML são adotados na comunicação entre os aplicativos.

Palavras-chave:

TV Digital. Moodle. XML. Integração.

1 Introdução

Segundo dados disponibilizados em 2008 pelo AbraEAD (Anuário Brasileiro Estatístico de Educação a Distância), um em cada 73 brasileiros estudava a distância (ABRAEAD, 2008), sendo que a maioria absoluta utiliza a Internet para seus estudos. Porém, há empecilhos em adotar ferramentas baseadas na Internet. De acordo com dados da Pnad (Pesquisa Nacional por Amostra de Domicílios), referentes a 2006 e divulgados em 2007 pelo IBGE, 27% das residências brasileiras tinham computador, um número bem inferior se comparado aos 94,5% dos domicílios que possuíam televisão. Do total de residências com computadores, apenas 20,4% estavam conectadas à Internet (IBGE, 2006).

Para muitas pessoas a TV é a única fonte de informação (BECKER, 2004). O Brasil está passando pelo processo de transição da TV analógica para a TV Digital que, dentre outros recursos, permite ao usuário solicitar dados a um provedor de serviços utilizando para tal o canal de retorno.

Visando a integração entre um aplicativo web voltado para Educação a Distância (EAD) e a TV Digital, foi estabelecida uma parceria entre o Laboratório de Aplicações e Inovação em Computação (LApIC) da UFJF e o Núcleo de Educação a Distância (NEAD) da mesma instituição. Neste contexto, é proposto no presente trabalho um modelo de integração entre o Moodle, plataforma web de EAD adotada pelo NEAD, e uma aplicação para a TV Digital de EAD. Assim, um estudante matriculado nos cursos oferecidos pelo NEAD poderá visualizar as informações disponibilizadas em seu curso através de um aparelho de televisão. No entanto, para a integração do Moodle com o aplicativo da TV Digital é necessário possuir um canal de retorno (KNIGHT, 2007).

Este trabalho está em conformidade com uma das metas da comissão formadora do Sistema Brasileiro de TV Digital Terrestre (SBTVD), que é promover a inclusão social e também digital, através da educação.

O trabalho está dividido da seguinte forma: nos Capítulos 2 e 3 é realizada a revisão bibliográfica de EAD e TV Digital, respectivamente; no Capítulo 4 é apresentado o ambiente de integração proposto pelo projeto; o capítulo 5 traz as etapas de desenvolvimento e os resultados obtidos e no capítulo 6 é apresentada a conclusão e sugestões para trabalhos futuros.

2 Educação a Distância

Educação a Distância (EAD) é a modalidade de ensino onde aluno e professor não se encontram no mesmo ambiente (espacial ou temporal). Durante certo tempo foi rejeitada por professores e estudantes sendo considerada de baixa qualidade, mas agora se mostra como um caminho incontornável (BELLONI, 2008).

A EAD permite que todos tenham acesso à educação, incluindo aqueles com dificuldades em frequentar o ensino presencial devido, por exemplo, à distância aos centros de ensino, dificuldade de locomoção ou por impossibilidade de assistir aulas nos horários tradicionais. Mas a EAD não deve ser considerada “apenas como um meio de superar problemas emergenciais (como parece ser o caso na LDB brasileira), ou de consertar alguns fracassos dos sistemas educacionais em dado momento de sua história [...]” (BELLONI, 2008, p.4), pois com a necessidade do mercado de profissionais qualificados e multidisciplinares, é grande a procura por capacitação nas instituições de ensino e a EAD deve ajudar a suprir essa demanda.

A interação aluno-professor e aluno-aluno é favorecida por diversos meios de comunicação tais como rádio, televisão, materiais impressos enviados pelos Correios ou Internet. A essa forma de ensino baseada na Internet denomina-se *e-learning*.

Para apoiar a prática de *e-learning* tem-se os Ambientes Virtuais de Aprendizagem (AVA). Um AVA é “um espaço fecundo de significação onde seres humanos e objetos técnicos interagem potencializando assim, a construção de conhecimentos, logo a aprendizagem [...]” (SANTOS, 2003). Sendo assim, um AVA não precisa, necessariamente, ser apoiado por tecnologias digitais, mas pode ser favorecido pelos recursos de multimídia e pela possibilidade de comunicação muitos-para-muitos permitida pela Internet.

2.1 Breve histórico da Educação a Distância no Brasil

No Brasil, de acordo com Telebrasil (2009), as primeiras práticas de EAD tiveram início por volta de 1904, com escolas internacionais oferecendo cursos pagos de idiomas por correspondência.

Posteriormente, surgiram várias iniciativas para levar conhecimento à população e utilizando diversos meios para tal. Algumas dessas iniciativas merecem destaque, como a

fundação da Rádio Sociedade do Rio de Janeiro, em 1923, que buscava, através dos serviços de radiodifusão, educar e divulgar temas relacionados à ciência e cultura instruindo mesmo aqueles que não sabiam ler.

Em 1942, é fundado o Instituto Universal Brasileiro que adotava os serviços de correspondência da antiga Empresa Brasileira de Correios e Telégrafos, atual Correios, para oferecer cursos profissionalizantes.

Já em 1978, empregando os meios de teledifusão, é lançado o Telecurso de 2º Grau, uma ação da Fundação Padre Anchieta e da Fundação Roberto Marinho.

Com a expansão da internet, aproximadamente a partir de 1994, as universidades passaram a oferecer cursos de graduação a distância. E em dezembro de 1996, a EAD é regulamentada com sua inclusão na Lei de Diretrizes e Bases da Educação, a lei nº 9394/96 (BRASIL, 1996).

2.2 O AVA Moodle

O Moodle (MOODLE, 2009) é o AVA idealizado por Martin Dougiamas no início da década de 1990. A palavra Moodle vem do acrônimo *Modular Object-Oriented Dynamic Learning Environment* (Ambiente Dinâmico de Aprendizagem Modular Orientado a Objetos).

Esta plataforma foi criada baseada na proposta pedagógica social construcionista, a qual sugere que a informação recebida é realmente apreendida pelo indivíduo se ele tiver que retransmiti-la a outros. Por isso é aconselhado ao aluno fazer anotações de aula, mesmo que nunca volte a lê-las. Sendo assim, o Moodle permite a postagem de materiais em diversos formatos, estimula a interação entre os participantes da disciplina com trocas de mensagens de forma síncrona e assíncrona e permite a criação de tarefas e envio das resoluções.

Por ser um software livre, segundo os termos da Licença Pública GNU, conta com a participação de profissionais de diversas áreas, não apenas desenvolvedores, resultando em sua constante evolução ganhando novas ferramentas e melhorias de interface, usabilidade e desempenho.

Segundo Santos (2003), as características esperadas em um AVA são: possibilitar acesso a documentos; permitir o relacionamento de um documento a outros disponíveis dentro ou fora do ambiente; garantir comunicação síncrona e assíncrona; favorecer as atividades de pesquisa; possibilitar recebimento de *feedback*; possuir interface agradável e que favoreça a

usabilidade. Todas essas características estão presentes no ambiente Moodle e seus módulos de atividades são listados na Tabela 1 (MOODLE, 2007).

Tabela 1 – Atividades do Moodle

Item	Descrição
Atividade Hot Potatoes	Trata-se de um quiz, onde o tutor disponibiliza as perguntas para os alunos responderem. Ao tutor é fornecido um relatório mostrando como cada aluno respondeu às perguntas e também exibe dados estatísticos.
Base de dados	É uma base de dados construída pelo tutor e/ou aluno a fim de permitir a consulta de diversos assuntos relacionados aos tópicos da matéria.
Chat	Comunicação síncrona entre os participantes. O histórico é salvo e pode ser disponibilizado para acesso dos alunos.
Diário	O tutor faz uma pergunta sobre um tópico para o aluno refletir. A resposta pode ser alterada e refinada pelo aluno durante o número de dias estipulado pelo tutor.
Escolha	O tutor disponibiliza uma pergunta com resposta de múltipla escolha.
Exercício	O aluno desenvolve um trabalho prático proposto pelo tutor e permite a auto-avaliação e a avaliação feita pelo tutor.
Fórum	Comunicação assíncrona entre os participantes. Arquivos podem ser enviados anexados às mensagens. Há diversos modelos de fóruns.
Glossário	Local onde são incluídos termos ligados aos assuntos tratados no curso. Além de textos, podem ser incluídos links ou criar uma galeria de fotos.
Laboratório de avaliação	Permite a avaliação de documentos por parceiros e o tutor pode atribuir nota a avaliação feita pelos estudantes.
LAMS	LAMS (<i>Learning Activity Management System</i>) é usado para <i>design</i> , administração e entrega de atividades de aprendizagem colaborativa <i>online</i> .
Lição	Espaço no qual o tutor pode definir atividades, enviar apresentações, arquivos de texto ou podem ser propostas tarefas sequenciais. O prosseguimento do aluno dependerá de suas respostas.
Pesquisa de avaliação	Pode ser utilizada para votação sobre determinado assunto ou para o tutor receber um <i>feedback</i> dos alunos, por exemplo.
Questionário	O tutor mantém uma base de dados com questões que deverão ser respondidas pelos alunos. As questões podem ser exibidas de forma aleatória, para evitar trapaças, e são avaliadas automaticamente. Podem ser questões de múltipla escolha, verdadeiro ou falso, respostas numéricas, associação, repostas curtas ou repostas embutidas dentro de textos.
Rótulo	Permite adicionar um breve texto em alguma área para dar instruções ao aluno como, por exemplo, instruí-lo a clicar em determinado botão.
Recurso	Local onde são disponibilizados materiais aos alunos. Pode ser disponibilizado um <i>link</i> para um arquivo ou site, uma página de texto simples ou uma página web.
SCORM/AICC	Permite que o tutor inclua qualquer pacote SCORM ou AICC em seu curso. SCORM (<i>Sharable Content Object Reference Model</i>) é uma coleção de especificações que permitem a interoperabilidade, acessibilidade e reutilização de aprendizagem baseado no conteúdo web.
Tarefa	Local onde o tutor disponibiliza uma atividade que deverá ser desenvolvida pelo aluno e encaminhada de volta ao tutor.
Wiki	Espaço para edição de texto colaborativo.

Além dos módulos de atividades, o ambiente possui mais funcionalidades, tais como: perfil dos participantes, lista com os inscritos no curso, mensagem enviada a um participante ou a um grupo, relatório de notas, blog dentre outras.

O Moodle é a plataforma adotada pelo NEAD/UFJF (www.nead.ufjf.br) para práticas de educação a distância, e também para auxílio nas disciplinas presenciais, e por isso foi escolhido para este trabalho de integração com a TV Digital.

3 TV Digital

A TV Digital, de forma simplificada, pode ser caracterizada pela substituição do sinal de transmissão de analógico para digital. No Brasil, as transmissões digitais tiveram início em 2007 e a previsão para a total migração é em 2016, com o desligamento das transmissões analógicas (DTV, 2009). Além da forma de transmissão do sinal, outras características da TV Digital são:

- **Alta definição de som (MPEG-2) e imagem (MPEG-4 ou H-264):** o áudio e vídeo recebidos pelo usuário apresentam a mesma qualidade dos sinais originais.
- **Multiprogramação:** permite o envio de dados (imagem, texto, aplicativos...) junto ao áudio e vídeo principal. Com isso, o conceito de canal precisa ser revisto, pois na TV analógica o canal de programação pode ser confundido com canal de frequência e na TV Digital pelo mesmo canal de frequência é possível transmitir até quatro programas de qualidade inferior (SDTV) ou um programa de qualidade mais alta (HDTV).
- **Interatividade:** permite que o usuário interaja com a programação e solicite dados à emissora ou a um provedor de serviços, caso possua um canal de retorno.
- **Multidispositivos:** o conteúdo pode ser apresentado em diversos dispositivos como, aparelho de televisão, celular ou qualquer outro que suporte um middleware.
- **Maior resolução:** na TV Digital as telas apresentam resolução 16:9 e transmissão de 30 quadros por segundos, com até 1080 linhas em cada quadro e 1920 pixels por linha, no padrão HDTV. Em contrapartida, a TV analógica apresenta resolução 4:3 com transmissão de 29,97 quadros por segundo, 480 linhas por quadro e 704 pixels em cada linha (SOARES, 2008).
- **Sinal:** tanto a TV analógica quanto a digital sofrem ruídos ocasionados pela distância em relação à fonte, múltiplos percursos, interferência por motores elétricos, dentre outros. Na TV analógica os ruídos resultam em "chuviscos" ou "fantasmas" na tela e na TV Digital, causam erros nos bits. Caso a taxa de erros seja aceitável, serão corrigidos pelos códigos corretores e o sinal chegará perfeito ao aparelho receptor do usuário. Caso esteja acima do permitido, não sendo possível sua correção, o sinal não será apresentado (SOARES, 2008).

3.1 Arquitetura da TV Digital

A arquitetura proposta pelo SBTVD representa um modelo simétrico de comunicação fim-a-fim caracterizando um ambiente cliente-servidor. Do lado servidor estão as estações de radiodifusão (representadas pelo lado direito da Figura 1) e do lado cliente estão os receptores dos usuários (representados pelo lado esquerdo da mesma Figura).

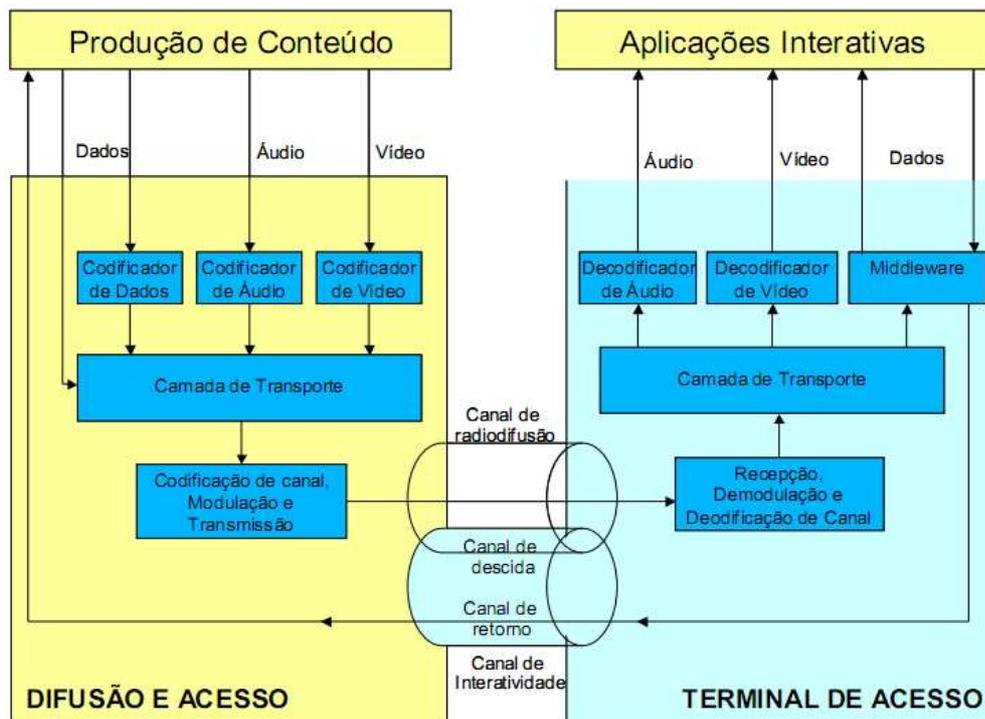


Figura 1: Arquitetura da TV Digital. Fonte: (CPqD, 2006)

A parte de Difusão e acesso é responsável pela codificação e empacotamento das informações produzidas e que deverão ser transmitidas aos usuários. Para que os sinais de áudio, vídeo e dados sejam transmitidos, precisam primeiro ser codificados por seus respectivos codificadores de áudio, vídeo e dados a fim de serem compactados e acrescidos de informações que permitam sua adequada recuperação do lado receptor. Uma vez codificados, os sinais são enviados para a camada de transporte que os empacota em um único fluxo de transporte e acrescenta informações auxiliares de controle. O fluxo de transporte segue da camada de transporte para o módulo de Codificação de canal, Modulação e Transmissão onde passará por um processamento adicional e será transmitido.

A parte do Terminal de acesso deve realizar as operações reversas para recuperação dos sinais e exibição ao usuário. O sinal é captado por antenas receptoras e passa por um processo de demodulação e decodificação sendo extraído o fluxo de transporte que será

encaminhado ao módulo da camada de transporte. A camada de transporte é responsável pela demultiplexação, ou seja, ela separa os sinais codificados de áudio, vídeo e dados e os entrega ao decodificador de áudio, decodificador de vídeo e ao middleware, respectivamente. Os decodificadores de áudio e de vídeo fazem a reconstituição do sinal original e os exibe ao usuário. O middleware, além da decodificação dos dados, é responsável por tratar as instruções e garantir a programação interativa.

De acordo com o CPqD (2006), a arquitetura da TV Digital é composta pelos seguintes subsistemas:

- **Transmissão e recepção:** No lado emissor é composto pelos módulos de Codificação de canal, Modulação e Transmissão, responsáveis por receber informações da camada de transporte e transmiti-las pelo canal de radiofrequência. No lado receptor é composto pelos módulos de Recepção, Demodulação e Decodificação, sendo os módulos responsáveis por sintonizar o canal, receber as informações, garantir a recuperação de erros e entregá-las à camada de transporte.
- **Codificação e decodificação de sinais:** No lado emissor é composto pelos codificadores de áudio, vídeo e dados que recebem os sinais originais, realizam a compressão e encaminham para a camada de transporte. No lado receptor, a camada de transporte entrega o áudio, o vídeo e os dados codificados aos decodificadores de áudio, vídeo e ao middleware, respectivamente, que realizam a decodificação e os exibe ao usuário.
- **Camada de transporte:** No lado emissor, recebe os sinais de áudio, vídeo e dados codificados, realiza a multiplexação, ou seja, encapsula estes sinais em um único feixe de transporte, acrescenta informações temporais para posterior sincronização no lado receptor, e entrega o fluxo de transporte ao subsistema de modulação. No lado receptor, recebe o fluxo do subsistema de demodulação e realiza a demultiplexação entregando os sinais codificados aos decodificadores de vídeo e áudio e ao middleware. A camada de transporte comporta o carrossel de dados, responsável pelo transporte dos sinais de dados, de forma cíclica, ao receptor. Sendo assim, somente quando o receptor possuir todos os dados requisitados por certo aplicativo, tal aplicativo será inicializado. Por isso, o carrossel de dados deve ser otimizado para reduzir o tempo de espera e evitar que o usuário troque de canal sem antes ver o aplicativo.
- **Middleware:** Está inserido na *set-top box* (STB) sendo definido como uma camada de software situada entre o hardware e as aplicações. É responsável por gerenciar as

atividades do receptor, manipular os sinais de dados, garantir a interatividade e permitir a correta exibição do conteúdo nos diversos meios, tais como, aparelho de televisão, celular ou PDA. O middleware brasileiro é o Ginga (GINGA, 2006), composto por uma parte declarativa, o Ginga-NCL, desenvolvido pelo Laboratório Telemídia da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), e uma parte procedural, o Ginga-J, desenvolvido pelo LAVID da Universidade Federal da Paraíba (UFPB).

- **Canal de interatividade:** Representa toda a comunicação feita entre as emissoras e os usuários, sendo composto pelo canal de descida (o sinal é enviado pela emissora em *broadcast* aos receptores) e pelo canal de retorno (comunicação *unicast* entre o receptor e a emissora ou a um provedor de serviços).
- **Terminal de acesso:** Localizado no lado receptor, é representado pela STB sendo responsável por capturar e processar os sinais de radiofrequência vindos das emissoras, decodificar as informações de áudio, vídeo e dados e exibi-las ao usuário, e suportar interatividade local ou completa, caso possua canal de retorno. As STB serão disponibilizadas comercialmente com dois perfis, onde o Perfil 1 deverá conter recursos básicos de interatividade e o Perfil 2 terá tecnologia mais avançada, conforme divulgado no site oficial da TV Digital brasileira (DTV, 2009).

3.2 Breve histórico da TV Digital no Brasil

Segundo informações disponibilizadas no site oficial da TV Digital brasileira (DTV, 2009), em 1999 a Agência Nacional de Telecomunicações (Anatel) iniciou os estudos para definir qual padrão de transmissão digital seria adotado no Brasil e foram importados e testados equipamentos dos três padrões vigentes, o americano, o europeu e o japonês.

Em julho de 2000, os testes foram finalizados e a intenção era adotar o padrão japonês, mas o pronunciamento oficial foi adiado até a posse do novo governo.

O novo governo, de caráter mais voltado ao social, buscava o desenvolvimento com tecnologias nacionais e as discussões foram retomadas. Assim, no dia 23 de novembro de 2003, é instituído o SBTVD através do Decreto Presidencial nº 4.901 (BRASIL, 2003).

No dia 29 de junho de 2006, teve início o processo de implantação do SBTVD com a publicação do Decreto Presidencial nº 5.820 (BRASIL, 2006), ficando decidido que o padrão brasileiro seria baseado no japonês, o ISDB-T, mas incorporaria tecnologias nacionais. A

tecnologia adotada do Japão refere-se ao modelo de transmissão, transporte e codificação de áudio. Para a codificação de vídeo, o Brasil adotou o H.264 (ou MPEG-4), mais moderno que o MPEG-2 empregado no Japão, e o middleware, o Ginga (GINGA, 2006), é brasileiro, desenvolvido pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), responsável pelo Ginga-NCL, e pela Universidade Federal da Paraíba (UFPB), que desenvolve o Ginga-J. Através deste decreto também foi decidido que as emissoras receberiam um novo canal de radiofrequência para não precisar interromper a transmissão analógica até a completa migração para o sinal digital.

As transmissões digitais iniciaram em 2 de dezembro de 2007, em São Paulo, mas ainda não incluem interatividade. O encerramento das transmissões analógicas está previsto para 29 de junho de 2016, conforme mostrado na Figura abaixo.



Figura 2: Previsão de implantação da TV Digital. Fonte: (DTV, 2009)

Em 2007, o Brasil iniciou seus esforços em expandir o padrão aos demais países da América Latina. Atualmente já é adotado por Argentina, Peru, Chile e Venezuela, sendo aguardada a decisão do Equador e Cuba.

Em outubro de 2009, recebe o aval da União Internacional de Telecomunicações (UIT), órgão de padronização e regulamentação em telecomunicações ligado à ONU, o que significa que qualquer país que queira produzir baseado no padrão brasileiro terá acesso às normas e com recomendação de um órgão de expressividade global.

O SBTVD conseguiu realizar, pela primeira vez no mundo, um trabalho de desenvolvimento e implementação de um sistema de TV Digital por meio de parceria entre emissoras de TV, fabricantes de transmissores, de receptores, a indústria de software, universidades, centros de pesquisa e o governo federal.

4 O modelo de integração

Para a integração entre o Moodle e a aplicação de EAD na TV Digital Interativa (TVDI) foi proposto o ambiente indicado na Figura 3.

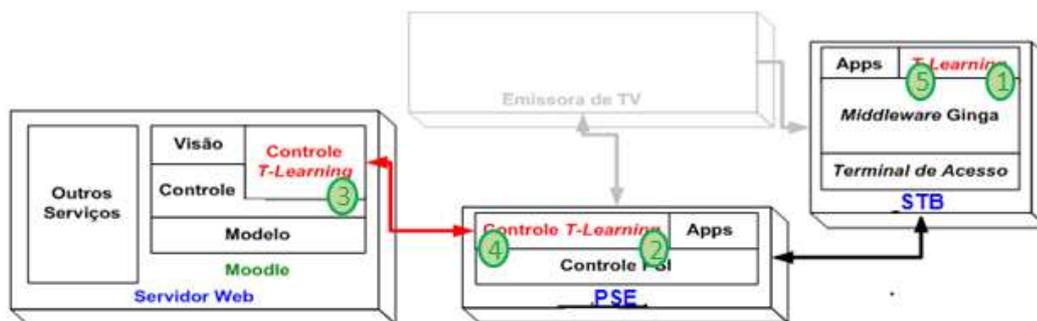


Figura 3: Ambiente de integração entre o Moodle e a TVDI (BARRÉRE-a, 2009)

Este ambiente é composto por um servidor web, um receptor e um Provedor de Serviços Interativos (PSI) que, neste caso, será chamado de Provedor de Serviços Educacionais (PSE). No servidor web está presente o Moodle e um *plug-in* responsável por receber as solicitações do aplicativo da TVDI, efetuar as consultas na base de dados do Moodle e enviar o resultado ao aplicativo. A escolha por desenvolver um *plug-in* ao invés de efetuar as adaptações direto no código, permite que o Moodle permaneça inalterado na web e faz maior distinção entre o aplicativo web e o da TVDI.

O receptor é representado pela STB que armazena o aplicativo para EAD na TVDI sendo responsável por exibir as informações ao usuário e também por receber as solicitações e enviá-las ao PSE. Este é responsável pela comunicação entre servidor e receptor. Ele recebe as solicitações vindas da STB, encaminha ao *plug-in* do Moodle no servidor web e efetua o caminho reverso, ou seja, recebe as respostas do *plug-in* e encaminha ao aplicativo da STB.

A comunicação entre os aplicativos é feita através de documentos XML (BARRÉRE-b, 2009). O aplicativo da TVDI gera um XML de requisição e envia ao PSE (Figura 3, fluxo de comunicação indicado pela numeração 1 e 2). O PSE analisa o documento XML recebido e encaminha ao servidor onde está hospedado o Moodle (Figura 3, fluxo de 2 para 3). O *plug-in* no servidor recebe o XML, analisa e gera um XML de resposta com os dados vindos do banco de dados do Moodle, enviando-o ao PSE (Figura 3, fluxo de 3 para 4). O PSE recebe o XML de resposta, analisa e entrega à aplicação solicitante (Figura 3, fluxo de 4 para 5). Optou-se por adotar o XML para a comunicação por ser uma linguagem que permite a criação das

próprias *tags* (adaptando ao ambiente de TVDI) e também por permitir a integração entre diferentes linguagens.

Este trabalho é parte integrante do projeto de integração da plataforma Moodle à TVDI desenvolvido pelo LApIC/UFJF em parceria com o NEAD/UFJF e tem como objetivo a implementação de um mecanismo de integração entre os dois ambientes.

O primeiro passo para o desenvolvimento do presente trabalho foi estudar o Moodle a fim de identificar suas funções e analisar quais dessas seriam integradas à TV Digital. Para a escolha das funções foi considerada a questão da interatividade, dependente da STB que comporta o canal de retorno. Como o objetivo do trabalho é promover a inclusão social, e também a inclusão digital, através da educação, buscou-se uma maneira de abranger o maior número possível de usuários. Já que as STB de Perfil 1 terão custo menor, por permitir menor interatividade, serão mais acessíveis à população. Sendo assim, as funções selecionadas devem poder ser exibidas em dispositivos com interação limitada. Dessa forma, as funcionalidades escolhidas não devem requerer inserção de dados, sendo selecionadas as funções que permitam visualização, tais como (BARRÉRE, 2008): perfil do usuário, mensagem recebida, participantes do curso, fórum, chat, wiki, glossário, recursos materiais e nota.

Após a definição dessas funções, teve início o processo de criação das *queries* para consulta ao banco de dados do Moodle. Como não será feita nenhuma inserção nem alteração, as *queries* utilizadas são de seleção e visando otimizar a consulta reduzindo o tempo de resposta, apenas os relacionamentos necessários foram empregados. Sendo assim, para projetos posteriores onde haja inserção ou alteração, os relacionamentos deverão ser revistos.

Com todas as *queries* identificadas, foi possível iniciar o processo de construção dos arquivos XML de requisição e de resposta. A função de cada arquivo XML é demonstrada na Tabela 2 e detalhada no capítulo seguinte.

Tabela 2 – Funcionalidades do Moodle integradas à TVDI

Funcionalidade	XML de requisição	XML de resposta
<i>Login</i>	Contém <i>login</i> e senha do usuário, bem como informações sobre a STB e o servidor para estabelecimento de conexão.	Contém a chave de entrada na tabela de <i>login</i> do <i>plug-in</i> do Moodle; informações sobre a STB e o servidor e os cursos matriculados.
Escolha do curso e listagem das funcionalidades	Armazena o identificador do curso escolhido.	Exibe a listagem das funcionalidades presentes no curso.
Atividade Chats	Contém o identificador 2 (definido na tabela <i>modules</i>).	Listagem de todos os chats visíveis no curso escolhido.
	Armazena o identificador do chat escolhido.	Exibe data, usuário e mensagem para cada interação no chat, finalizando a atividade.

Funcionalidade	XML de requisição	XML de resposta
Atividade Fóruns	Contém o identificador 6 referente à atividade fóruns (definido na tabela <i>modules</i>).	Exibe a listagem de todos os fóruns visíveis no curso escolhido.
	Armazena o identificador do fórum escolhido.	Exibe a listagem de todas as discussões do fórum selecionado.
	Armazena o identificador da discussão escolhida.	Exibe todas as postagens da discussão, encerrando a atividade.
Atividade Glossários	Contém o identificador 7 referente à atividade glossários (definido na tabela <i>modules</i>).	Exibe a listagem de todos os glossários visíveis no curso.
	Armazena o identificador do glossário escolhido.	Exibe cada conceito, definição e usuário, finalizando a atividade.
Atividade Recursos Materiais	Contém o identificador 14 referente à atividade recursos (definido na tabela <i>modules</i>).	Exibe a listagem de todos os recursos visíveis no curso selecionado.
	Armazena o identificador do recurso escolhido.	Tipo <i>text</i> ou <i>html</i> : exibe o conteúdo do atributo <i>alltext</i> da tabela <i>resource</i> e finaliza a atividade. Tipo <i>file</i> iniciando por <i>http://</i> : exibe o conteúdo do atributo <i>reference</i> da tabela <i>resource</i> . Ao receber o XML, a STB fará uma solicitação de <i>download</i> , finalizando a atividade. Tipo <i>file</i> : exibe o diretório de armazenamento de arquivos do Moodle, ou seja, <i>..\moodledata\id_curso\atributo_reference</i> . Ao receber o XML, a STB fará uma solicitação de <i>download</i> , finalizando a atividade.
Atividade Wikis	Contém o identificador 17 referente à atividade wikis (definido na tabela <i>modules</i>).	Exibe a listagem de todas as wikis visíveis no curso.
	Armazena o identificador da wiki escolhida.	Exibe a versão mais recente da wiki selecionada, finalizando a atividade.
Função Perfil	Contém o identificador 20 referente à função perfil (definido para o trabalho).	Exibe informações cadastradas pelo usuário na plataforma Moodle. Para exibição da foto, é indicado o diretório de armazenamento no Moodle, sendo <i>..\moodledata\user\id_grupo\id_usuario\</i> para os usuários que cadastraram uma imagem e <i>..\user\default\</i> para aqueles que não cadastraram. Ao receber o XML, a STB faz uma requisição de <i>download</i> , finalizando a função.
Função Participantes	Contém o identificador 21 referente à função participantes (definido para o trabalho).	Exibe nome, cidade, país e imagem dos inscritos no curso, finalizando a função. Para exibição da imagem, são seguidos os mesmos procedimentos descritos na função perfil.
Função Mensagem	Contém o identificador 22 referente à função mensagem (definido para o trabalho).	Exibe nome do remetente, data e a mensagem ainda não lida pelo usuário <i>logado</i> .
Função Nota	Contém o identificador 23 referente à função nota (definido para o trabalho).	Exibe o nome da tarefa avaliada, a pontuação máxima e mínima, a pontuação obtida, a pontuação total e pode exibir um <i>feedback</i> .

Deve-se ressaltar que o aplicativo da TVDI não pretende substituir o uso do Moodle na web. O objetivo é permitir a visualização do ambiente Moodle por pessoas que não possuam conexão com a Internet, visto que para possuir um canal de retorno não é necessária conexão com Internet, podendo ser, por exemplo, um telefone fixo ou um telefone celular (KNIGHT, 2007). Sendo assim, as atividades típicas de administradores e tutores tais como, criação de cursos, matrícula dos alunos nas turmas, inserção de recursos materiais e lançamento de notas, continuarão sendo feitas pela web.

4.1 A linguagem XML

A *Extensible Markup Language* (XML – Linguagem de Marcação Extensível) foi criada em 1996 com o objetivo de se tornar uma linguagem de marcação de ampla distribuição pela Internet, integrada às demais linguagens de marcação, como a HTML, e facilmente compreendida tanto por programas quanto por seres humanos (TITTEL, 2003).

Enquanto a HTML é uma linguagem de apresentação, a XML é mais adotada no processamento dos dados e por permitir ao programador a definição de suas *tags* (elementos e atributos) se adapta a diversas classes de aplicações.

Para a validação dos elementos aceitos, de seus respectivos atributos e da estrutura lógica dos arquivos XML, é necessário referenciar um *Document Type Definition* (DTD - Definição de Tipo de Documento) ou um *XML Schema Definition* (XSD – Definição de Esquema de XML).

Para este trabalho foram adotados arquivos DTD inseridos nos próprios XML por atender às necessidades atuais do projeto.

5 O processo de comunicação

O aplicativo de EAD é disponibilizado ao usuário na TVDI pelo canal de retorno. Uma vez disponível, o usuário poderá interagir com a aplicação solicitando visualização das funcionalidades de um curso através de comandos do controle remoto. A primeira ação a ser desempenhada pelo usuário é efetuar *login*. Esta e outras ações são descritas a seguir.

5.1 Login do usuário

Para ter acesso ao Moodle o usuário deve primeiro efetuar o *login* informando seu nome de usuário e senha. O aplicativo da TVDI gera um arquivo XML de requisição de *login*, como apresentado na Figura 4.

Todos os arquivos XML contêm a declaração do DTD que define as funcionalidades apresentadas bem como os elementos e atributos aceitos (LEITE, 2009). Estas *tags* recebem, sempre que possível, a mesma nomenclatura adotada nos atributos das tabelas do Moodle. Para a determinação da obrigatoriedade dos elementos foram observados os atributos de preenchimento obrigatório na criação de atividades e funções.

```
1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3 <ELEMENT stbmoodle (head)>
4 <ELEMENT head (device, location?, moodleserver, loginuser)>
5 <ELEMENT device EMPTY>
6 <ATTLIST device serialnumber CDATA #REQUIRED
7 type CDATA #REQUIRED>
8 <ELEMENT location EMPTY>
9 <ATTLIST location zip CDATA #IMPLIED
10 lat CDATA #IMPLIED
11 long CDATA #IMPLIED>
12 <ELEMENT moodleserver EMPTY>
13 <ATTLIST moodleserver ip CDATA #REQUIRED>
14 <ELEMENT loginuser EMPTY>
15 <ATTLIST loginuser usr CDATA #REQUIRED
16 psw CDATA #REQUIRED>
17 ]>
18 <stbmoodle>
19 <head>
20 <device serialnumber="200910xs" type="set-top-box"/>
21 <location zip="36035130" lat="22.017500" long="47.890831"/>
22 <moodleserver ip="192.168.2.100"/>
23 <loginuser usr="paulamarin" psw="084e0343a0486ff05530df6c705c8bb4"/>
24 </head>
25 </stbmoodle>
```

Figura 4: XML para requisição de *login*

O cabeçalho deste arquivo (Figura 4, linhas 19 a 24) permite a identificação do receptor e do servidor, necessária para o estabelecimento da conexão, e também armazena os

dados para *login* do usuário. O receptor é identificado pelo elemento *device* que contém os atributos de tipo (*type*) de receptor e número serial (*serialnumber*). O servidor é identificado pelo elemento *moodleserver* que contém o atributo *ip* responsável pelo armazenamento de seu endereço IP. O usuário é identificado pelo elemento *loginuser* que contém os atributos para armazenamento de seu nome de usuário (atributo *usr*) e senha (atributo *psw* – utilizando o algoritmo de criptografia MD5). O elemento *location* com seus atributos para CEP (*zip*), latitude (*lat*) e longitude (*long*) é adotado apenas para coleta de dados estatísticos e, embora conste neste exemplo, não é obrigatório. A falta da obrigatoriedade pode ser percebida no DTD do documento representada por *location?* (Figura 4, linha 4).

O XML de requisição é enviado ao PSE que o encaminha ao servidor. O *plug-in* no servidor recebe o XML, analisa, efetua as consultas na base de dados do Moodle e gera o XML de resposta, conforme Figura 5:

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course*)>
4      <!ELEMENT head (moodleserver, location?, device, userlogged)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED
7            version CDATA #REQUIRED>
8        <!ELEMENT location EMPTY>
9          <!ATTLIST location zip CDATA #IMPLIED
10             lat CDATA #IMPLIED
11             long CDATA #IMPLIED>
12        <!ELEMENT device EMPTY>
13          <!ATTLIST device serialnumber CDATA #REQUIRED
14             type CDATA #REQUIRED>
15        <!ELEMENT userlogged EMPTY>
16          <!ATTLIST userlogged id CDATA #REQUIRED>
17        <!ELEMENT course (fullname, shortname, summary?)>
18          <!ATTLIST course id CDATA #REQUIRED>
19          <!ELEMENT fullname (#PCDATA)>
20          <!ELEMENT shortname (#PCDATA)>
21          <!ELEMENT summary (#PCDATA)>
22    ]>
23  <moodlestb>
24    <head>
25      <moodleserver ip="192.168.2.100" version="1.9.5+"/>
26      <location zip="36680000" lat="22.017501" long="47.890832"/>
27      <device serialnumber="200910xs" type="set-top-box"/>
28      <userlogged id="101"/>
29    </head>
30    <course id="2">
31      <fullname>Engenharia de Software</fullname>
32      <shortname>ES</shortname>
33      <summary>Curso destinado à disciplina de ES da UFJF.</summary>
34    </course>
35    <course id="3">
36      <fullname>Bando de Dados</fullname>
37      <shortname>BD</shortname>
38      <summary>Curso destinado à disciplina de BD da UFJF.</summary>
39    </course>
40  </moodlestb>

```

Figura 5: XML de resposta de *login*

O cabeçalho (Figura 5, linhas 24 a 29) é semelhante ao apresentado na Figura 4 e permite a identificação do Moodle, do receptor e do usuário. Para identificar o Moodle, é informada sua versão (atributo *version* do elemento *moodleserver*), além do endereço IP e o usuário passa a ser identificado pelo atributo *id* do elemento *userlogged*. Para validação do *login*, o *plug-in* verifica na base de dados do Moodle se existe o usuário identificado por *usr* e *psw* (Figura 4, linha 23) e, caso exista, verifica se não foi excluído. Ao verificar nesta fase se o usuário está ativo, nas demais *queries* a verificação de exclusão não será mais necessária.

Concluída a validação, o *plug-in* insere em uma tabela própria o identificador do usuário, consultado na base de dados do Moodle, e adiciona a chave de entrada nessa tabela no atributo *id* do elemento *userlogged* (Figura 5, linha 28). Dessa forma, nos próximos XML de requisição não será mais necessário enviar nome de *login* e senha para validação, bastando disponibilizar a chave de entrada da tabela de *login*.

No corpo do XML de resposta (Figura 5, linhas 30 a 39) são listados os cursos aos quais o usuário está matriculado. Para cada curso é fornecido seu identificador (atributo *id*), nome completo (elemento *fullname*), nome abreviado (elemento *shortname*) e descrição (elemento *summary*), sendo este um elemento opcional, conforme verificado em *summary?* (Figura 5, linha 17), pois o Moodle não exige que seja obrigatório seu preenchimento ao se criar um novo curso.

O usuário pode estar cadastrado no Moodle, mas não estar matriculado em nenhum curso, assim como pode estar matriculado em vários cursos. Essa situação é demonstrada na Figura 5, linha 3 ao indicar *course**. O XML de resposta é então enviado ao PSE e encaminhado ao receptor. O aplicativo para EAD localizado no receptor exibe na tela a relação de cursos. A Figura 6 mostra a *query* para validação do usuário e a Figura 7 mostra a relação dos cursos matriculados.

```
1 SELECT id FROM user
2 WHERE username LIKE "paulamarin" AND
3     password = "084e0343a0486ff05530df6c705c8bb4" AND
4     deleted = 0;
```

Figura 6: *Query* para validação de *login*

```
1 SELECT co.id, co.fullname, co.shortname, co.summary
2 FROM user u
3 JOIN role_assignments ra ON ra.userid = u.id
4 JOIN context c           ON c.id = ra.contextid
5 JOIN course co          ON co.id = c.instanceid
6 WHERE c.contextlevel = 50 AND u.id = 234
```

Figura 7: *Query* para relação de cursos

Nesta parte, a conexão entre o aplicativo da TVDI e o servidor web, onde o Moodle está hospedado, foi estabelecida. Nos demais arquivos de comunicação as informações de tipo (*type*) do receptor, versão (*version*) do Moodle e localização (*location*) não farão parte do cabeçalho.

5.2 Escolha do curso

Nesta fase, o usuário deverá selecionar, dentre as opções exibidas, o curso que deseja acessar.

Após a seleção do curso desejado, o aplicativo da TVDI gera um novo XML de requisição com o identificador do curso, conforme Figura 8.

```
1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE stbmoodle[
3    <!ELEMENT stbmoodle (head, course)>
4      <!ELEMENT head (device, moodleserver, userlogged)>
5        <!ELEMENT device EMPTY>
6          <!ATTLIST device serialnumber CDATA #REQUIRED>
7        <!ELEMENT moodleserver EMPTY>
8          <!ATTLIST moodleserver ip CDATA #REQUIRED>
9        <!ELEMENT userlogged EMPTY>
10         <!ATTLIST userlogged id CDATA #REQUIRED>
11      <!ELEMENT course EMPTY>
12        <!ATTLIST course id CDATA #REQUIRED>
13    ]>
14  <stbmoodle>
15    <head>
16      <device serialnumber="123456xx"/>
17      <moodleserver ip="192.168.2.100"/>
18      <userlogged id="101"/>
19    </head>
20    <course id="2"/>
21  </stbmoodle>
```

Figura 8: XML para requisição de curso

O atributo *id* do elemento *course* armazena o identificador do curso escolhido. O *plugin* recebe o XML e gera outro arquivo de resposta contendo as funcionalidades disponibilizadas para o curso selecionado, conforme Figura 9.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE moodlestb[
3   <!ELEMENT moodlestb (head, course)>
4     <!ELEMENT head (moodleserver, device)>
5       <!ELEMENT moodleserver EMPTY>
6         <!ATTLIST moodleserver ip CDATA #REQUIRED>
7       <!ELEMENT device EMPTY>
8         <!ATTLIST device serialnumber CDATA #REQUIRED>
9     <!ELEMENT course (chats?, forums?, glossaries?, resources?, wikis?, profile, assignments, messages, grade)>
10       <!ATTLIST course id CDATA #REQUIRED>
11       <!ELEMENT chats EMPTY>
12         <!ATTLIST chats id CDATA #REQUIRED>
13       <!ELEMENT forums EMPTY>
14         <!ATTLIST forums id CDATA #REQUIRED>
15       <!ELEMENT glossaries EMPTY>
16         <!ATTLIST glossaries id CDATA #REQUIRED>
17       <!ELEMENT resources EMPTY>
18         <!ATTLIST resources id CDATA #REQUIRED>
19       <!ELEMENT wikis EMPTY>
20         <!ATTLIST wikis id CDATA #REQUIRED>
21       <!ELEMENT profile EMPTY>
22         <!ATTLIST profile id CDATA #REQUIRED>
23       <!ELEMENT assignments EMPTY>
24         <!ATTLIST assignments id CDATA #REQUIRED>
25       <!ELEMENT messages EMPTY>
26         <!ATTLIST messages id CDATA #REQUIRED>
27       <!ELEMENT grade EMPTY>
28         <!ATTLIST grade id CDATA #REQUIRED>
29   ]>
30 <moodlestb>
31   <head>
32     <moodleserver ip="192.168.2.100"/>
33     <device serialnumber="200910xs"/>
34   </head>
35   <course id="2">
36     <!-- Atividades inseridas no curso 2 -->
37     <chats id="2"/>
38     <forums id="6"/>
39     <glossaries id="7"/>
40     <resources id="14"/>
41     <wikis id="17"/>
42     <!-- Atividades que estão presentes em todos os cursos -->
43     <profile id="20"/>
44     <assignments id="21"/>
45     <messages id="22"/>
46     <grade id="23"/>
47   </course>
48 </moodlestb>

```

Figura 9: XML de resposta com as funcionalidades disponíveis para o curso

O Moodle mantém em sua tabela *modules* o identificador para cada módulo das atividades mostradas na Tabela 1 deste documento. Este identificador é sequencial variando de 1 a 18 na versão utilizada no trabalho, ou seja, a versão 1.9.5+, sendo o número 2 indicativo de chat, o número 6 atribuído para indicar fórum, o 7 para glossário, o 14 para recursos materiais e o 17 para wiki. Estes valores foram adotados no trabalho a fim de manter o mesmo padrão de identificação e foram acrescentados os identificadores 20 para a função de exibição do perfil, 21 para lista dos participantes, 22 para visualização das mensagens recebidas e 23 para visualização da nota. As funções para exibição de perfil, participantes, mensagem e nota são disponibilizadas em todos os cursos, mas os módulos de atividades (chat, fórum, glossário, recurso e wiki) variam de curso para curso, podendo ser exibidos em certos cursos e não exibidos em outros. Este fato é verificado na linha 9 da Figura 9. A *query* para exibir os módulos de atividades presentes em cada curso é mostrada na Figura 10.

```

1 SELECT distinct cm.module
2 FROM modules cm
3 JOIN course c ON c.id = cm.course
4 WHERE c.id = 2 AND cm.visible = 1;

```

Figura 10: *Query* para listagem das atividades disponibilizadas para o curso

5.3 Escolha da atividade Chats

O aplicativo da TVDI exibe as funções disponíveis para escolha do usuário. Feita a seleção, o aplicativo gera um XML de requisição informando o identificador da funcionalidade escolhida. Como primeiro exemplo, é demonstrada a seleção da atividade chats e o XML é exibido na Figura 11.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
2 <!DOCTYPE stbmoodle[
3   <!ELEMENT stbmoodle (head, course)>
4     <!ELEMENT head (device, moodleserver, userlogged)>
5       <!ELEMENT device EMPTY>
6         <!ATTLIST device serialnumber CDATA #REQUIRED>
7       <!ELEMENT moodleserver EMPTY>
8         <!ATTLIST moodleserver ip CDATA #REQUIRED>
9       <!ELEMENT userlogged EMPTY>
10        <!ATTLIST userlogged id CDATA #REQUIRED>
11     <!ELEMENT course (request)>
12       <!ATTLIST course id CDATA #REQUIRED>
13     <!ELEMENT request EMPTY>
14       <!ATTLIST request id CDATA #REQUIRED>
15 ]>
16 <stbmoodle>
17   <head>
18     <device serialnumber="123456xx"/>
19     <moodleserver ip="192.168.2.100"/>
20     <userlogged id="101"/>
21   </head>
22   <course id="2">
23     <request id="2"/>
24   </course>
25 </stbmoodle>

```

Figura 11: XML para requisição da atividade chats

O elemento *request* é responsável por armazenar em seu atributo *id* o identificador da função selecionada, onde 2 identifica a atividade chats. O *plug-in* recebe o XML de requisição e gera um arquivo de resposta com os chats, conforme Figura 12.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4      <!ELEMENT head (moodleserver, device)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED>
7        <!ELEMENT device EMPTY>
8          <!ATTLIST device serialnumber CDATA #REQUIRED>
9      <!ELEMENT course (request)>
10        <!ATTLIST course id CDATA #REQUIRED>
11        <!ELEMENT request (chat+)>
12          <!ATTLIST request id CDATA #REQUIRED>
13          <!ELEMENT chat (name, intro)>
14            <!ATTLIST chat id CDATA #REQUIRED>
15            <!ELEMENT name (#PCDATA)>
16            <!ELEMENT intro (#PCDATA)>
17    ]>
18  <moodlestb>
19    <head>
20      <moodleserver ip="192.168.2.100"/>
21      <device serialnumber="200910xs"/>
22    </head>
23    <course id="2">
24      <request id="2">
25        <chat id="1">
26          <name>Trabalho 1</name>
27          <intro>Discussões sobre o trabalho 1.</intro>
28        </chat>
29        <chat id="2">
30          <name>Trabalho 2</name>
31          <intro>Discussões sobre o trabalho 2.</intro>
32        </chat>
33      </request>
34    </course>
35  </moodlestb>

```

Figura 12: XML de resposta com os chats disponíveis no curso

Cada elemento *chat* possui um identificador (atributo *id*), um nome (elemento *name*) e uma descrição (elemento *intro*), sendo o identificador atribuído pelo Moodle e nome e descrição de preenchimento obrigatório para a criação de um novo chat. A partir do momento em que na Figura 9 foi constatada a presença da atividade chats (linha 37) para o curso escolhido, deve haver no mínimo um ou vários chats, sendo este fato representado por *chat+* na linha 11 da Figura 12.

Após a exibição de todos os chats disponíveis, o usuário deverá selecionar um deles para visualizar seu histórico, conforme Figura 13.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE stbmoodle[
3    <!ELEMENT stbmoodle (head, course)>
4      <!ELEMENT head (device, moodleserver, userlogged)>
5        <!ELEMENT device EMPTY>
6          <!ATTLIST device serialnumber CDATA #REQUIRED>
7        <!ELEMENT moodleserver EMPTY>
8          <!ATTLIST moodleserver ip CDATA #REQUIRED>
9        <!ELEMENT userlogged EMPTY>
10         <!ATTLIST userlogged id CDATA #REQUIRED>
11      <!ELEMENT course (request)>
12        <!ATTLIST course id CDATA #REQUIRED>
13        <!ELEMENT request (chat)>
14          <!ATTLIST request id CDATA #REQUIRED>
15          <!ELEMENT chat EMPTY>
16            <!ATTLIST chat id CDATA #REQUIRED>
17    ]>
18  <stbmoodle>
19    <head>
20      <device serialnumber="123456xx"/>
21      <moodleserver ip="192.168.2.100"/>
22      <userlogged id="101"/>
23    </head>
24    <course id="2">
25      <request id="2">
26        <chat id="1"/>
27      </request>
28    </course>
29  </stbmoodle>

```

Figura 13: XML para requisição de um chat

O atributo *id* do elemento *chat* armazena o identificador do chat escolhido. O *plug-in*, ao receber este XML, realiza uma consulta e gera um XML de resposta com o histórico do chat selecionado, conforme Figuras 14 e 15.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4      <!ELEMENT head (moodleserver, device)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED>
7        <!ELEMENT device EMPTY>
8          <!ATTLIST device serialnumber CDATA #REQUIRED>
9      <!ELEMENT course (request)>
10        <!ATTLIST course id CDATA #REQUIRED>
11        <!ELEMENT request (chat)>
12          <!ATTLIST request id CDATA #REQUIRED>
13          <!ELEMENT chat (interaction*)>
14            <!ATTLIST chat id CDATA #REQUIRED>
15            <!ELEMENT interaction (date, user, message)>
16              <!ELEMENT date (#PCDATA)>
17              <!ELEMENT user (#PCDATA)>
18              <!ELEMENT message (#PCDATA)>
19    ]>

```

Figura 14: XML de resposta contendo histórico do chat escolhido (Parte I)

```

20 <moodlestb>
21 <head>
22   <moodleserver ip="192.168.2.100"/>
23   <device serialnumber="200910xs"/>
24 </head>
25 <course id="2">
26   <request id="2">
27     <chat id="1">
28       <iteration>
29         <date>1257524296</date>
30         <user>Paula</user>
31         <message>enter</message>
32       </iteration>
33       <iteration>
34         <date>1257524307</date>
35         <user>Paula</user>
36         <message>Olá a todos</message>
37       </iteration>
38       <iteration>
39         <date>1257524344</date>
40         <user>Paula</user>
41         <message>exit</message>
42       </iteration>
43     </chat>
44   </request>
45 </course>
46 </moodlestb>

```

Figura 15: XML de resposta contendo histórico do chat escolhido (Parte II)

A query para exibição dos chats de um curso é exibida na Figura 16.

```

1 SELECT ch.id, ch.name, ch.intro
2 FROM chat ch,
3     course_modules cm,
4     mdl_modules m
5 WHERE m.name = 'chat' AND
6     cm.module = m.id AND
7     cm.visible = 1 AND
8     ch.id = cm.instance AND
9     cm.course = 2

```

Figura 16: Query para listagem dos chats de um curso

Como é permitido ao tutor criar um chat, mas mantê-lo invisível, a condição mostrada na linha 7 da Figura 16 garante que apenas os chats do curso selecionado e que estejam visíveis sejam mostrados. A consulta ao banco de dados do Moodle para recuperação do histórico do chat é exibida na Figura 17. Assim, é encerrada a atividade de verificação do histórico de um chat.

```

1 SELECT m.timestamp, m.message, u.firstname
2 FROM chat_messages m
3 JOIN user u ON u.id = m.userid
4 WHERE m.chatid = 1;

```

Figura 17: Query para exibição do histórico do chat escolhido

5.4 Escolha da atividade Fóruns

O usuário também pode optar por acessar a atividade fóruns, conforme Figura 18.

```
1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE stbmoodle[
3      <!ELEMENT stbmoodle (head, course)>
4          <!ELEMENT head (device, moodleserver, userlogged)>
5              <!ELEMENT device EMPTY>
6                  <!ATTLIST device serialnumber CDATA #REQUIRED>
7              <!ELEMENT moodleserver EMPTY>
8                  <!ATTLIST moodleserver ip CDATA #REQUIRED>
9              <!ELEMENT userlogged EMPTY>
10                 <!ATTLIST userlogged id CDATA #REQUIRED>
11          <!ELEMENT course (request)>
12              <!ATTLIST course id CDATA #REQUIRED>
13              <!ELEMENT request EMPTY>
14                  <!ATTLIST request id CDATA #REQUIRED>
15      ]>
16  <stbmoodle>
17  <head>
18      <device serialnumber="123456xx"/>
19      <moodleserver ip="192.168.2.100"/>
20      <userlogged id="101"/>
21  </head>
22  <course id="2">
23      <request id="6"/>
24  </course>
25  </stbmoodle>
```

Figura 18: XML para requisição da atividade fóruns

Neste caso, o atributo *id* do elemento *request* recebe o valor 6. O XML de requisição é enviado ao *plug-in* que gera um XML de resposta contendo os fóruns visíveis para o curso selecionado, conforme Figura 19.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3      <!ELEMENT moodlestb (head, course)>
4          <!ELEMENT head (moodleserver, device)>
5              <!ELEMENT moodleserver EMPTY>
6                  <!ATTLIST moodleserver ip CDATA #REQUIRED>
7              <!ELEMENT device EMPTY>
8                  <!ATTLIST device serialnumber CDATA #REQUIRED>
9          <!ELEMENT course (request)>
10              <!ATTLIST course id CDATA #REQUIRED>
11              <!ELEMENT request (forum+)>
12                  <!ATTLIST request id CDATA #REQUIRED>
13                  <!ELEMENT forum (name, intro, countdiscussions)>
14                      <!ATTLIST forum id CDATA #REQUIRED>
15                      <!ELEMENT name (#PCDATA)>
16                      <!ELEMENT intro (#PCDATA)>
17                      <!ELEMENT countdiscussions (#PCDATA)>
18          ]>
19  <moodlestb>
20  <head>
21      <moodleserver ip="192.168.2.100"/>
22      <device serialnumber="123456xx"/>
23  </head>
24  <course id="2">
25      <request id="6">
26          <forum id="1">
27              <name>Fórum de notícias</name>
28              <intro>Notícias e avisos</intro>
29              <countdiscussions>1</countdiscussions>
30          </forum>
31          <forum id="2">
32              <name>Trabalho em grupo</name>
33              <intro>Informar os grupos</intro>
34              <countdiscussions>3</countdiscussions>
35          </forum>
36      </request>
37  </course>
38 </moodlestb>

```

Figura 19: XML de resposta com os fóruns disponíveis no curso

Cada elemento *forum* possui um identificador (atributo *id*), um nome (elemento *name*), uma descrição (elemento *intro*) e o total de discussões (elemento *countdiscussions*), onde identificador é atribuído pelo Moodle, total de discussões é um valor calculado e nome e descrição de preenchimento obrigatório para a criação de um novo fórum, conforme Figura 19, linha 13. Assim como na atividade chats, deve haver no mínimo um fórum para o curso escolhido e este fato é representado por *forum+* na linha 11 da Figura 19.

O usuário receberá em sua tela a listagem de todos os fóruns e deverá selecionar um, gerando um XML de requisição, conforme Figura 20.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE stbmoodle[
3      <!ELEMENT stbmoodle (head, course)>
4          <!ELEMENT head (device, moodleserver, userlogged)>
5              <!ELEMENT device EMPTY>
6                  <!ATTLIST device serialnumber CDATA #REQUIRED>
7              <!ELEMENT moodleserver EMPTY>
8                  <!ATTLIST moodleserver ip CDATA #REQUIRED>
9              <!ELEMENT userlogged EMPTY>
10                 <!ATTLIST userlogged id CDATA #REQUIRED>
11          <!ELEMENT course (request)>
12              <!ATTLIST course id CDATA #REQUIRED>
13              <!ELEMENT request (forum)>
14                  <!ATTLIST request id CDATA #REQUIRED>
15                  <!ELEMENT forum EMPTY>
16                      <!ATTLIST forum id CDATA #REQUIRED>
17          ]>
18  <stbmoodle>
19  <head>
20      <device serialnumber="123456xx"/>
21      <moodleserver ip="192.168.2.100"/>
22      <userlogged id="101"/>
23  </head>
24  <course id="2">
25      <request id="6">
26          <forum id="1"/>
27      </request>
28  </course>
29  </stbmoodle>

```

Figura 20: XML para requisição de um fórum

O atributo *id* do elemento *forum* armazena o identificador do fórum escolhido. O *plugin* receberá esse XML e irá gerar um arquivo de resposta com a relação das discussões para o fórum selecionado, conforme Figuras 21 e 22.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3      <!ELEMENT moodlestb (head, course)>
4          <!ELEMENT head (moodleserver, device)>
5              <!ELEMENT moodleserver EMPTY>
6                  <!ATTLIST moodleserver ip CDATA #REQUIRED>
7              <!ELEMENT device EMPTY>
8                  <!ATTLIST device serialnumber CDATA #REQUIRED>
9          <!ELEMENT course (request)>
10              <!ATTLIST course id CDATA #REQUIRED>
11              <!ELEMENT request (forum)>
12                  <!ATTLIST request id CDATA #REQUIRED>
13                  <!ELEMENT forum (discussion*)>
14                      <!ATTLIST forum id CDATA #REQUIRED>
15                      <!ELEMENT discussion (name, user, countposts)>
16                          <!ATTLIST discussion id CDATA #REQUIRED>
17                          <!ELEMENT name (#PCDATA)>
18                          <!ELEMENT user (#PCDATA)>
19                          <!ELEMENT countposts (#PCDATA)>
20          ]>

```

Figura 21: XML de resposta contendo as discussões do fórum escolhido (Parte I)

```

21 <moodlestb>
22 <head>
23   <moodleserver ip="192.168.2.100"/>
24   <device serialnumber="123456xx"/>
25 </head>
26 <course id="2">
27   <request id="6">
28     <forum id="1">
29       <discussion id="1">
30         <name>Último dia de aula</name>
31         <user>José Silva</user>
32         <countposts>2</countposts>
33       </discussion>
34     </forum>
35   </request>
36 </course>
37 </moodlestb>

```

Figura 22: XML de resposta contendo as discussões do fórum escolhido (Parte II)

Cada discussão é representada por um elemento *discussion* e possui identificador (atributo *id*), assunto (elemento *name*), usuário que a criou (elemento *user*) e o total de postagens (elemento *countposts*), sendo identificador atribuído pelo Moodle, usuário atributo consultado, postagens atributo calculado e assunto atributo de preenchimento obrigatório para criação de uma discussão. Um fórum pode ter sido criado, mas não possuir nenhuma discussão, assim como pode ter várias. Este fato é representado na linha 13 da figura 21 por *discussion**.

Este XML é encaminhado à STB e exibido ao usuário que deverá selecionar uma dentre as discussões disponíveis. O aplicativo gera um XML de requisição com o identificador da discussão, conforme Figura 23.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3   <!ELEMENT stbmoodle (head, course)>
4     <!ELEMENT head (device, moodleserver, userlogged)>
5       <!ELEMENT device EMPTY>
6         <!-->
7         <!-->
8         <!-->
9         <!-->
10        <!-->
11        <!-->
12        <!-->
13        <!-->
14        <!-->
15        <!-->
16        <!-->
17        <!-->
18        <!-->
19      ]>
20 <stbmoodle>
21 <head>
22   <device serialnumber="123456xx"/>
23   <moodleserver ip="192.168.2.100"/>
24   <userlogged id="uid1"/>
25 </head>
26 <course id="2">
27   <request id="6">
28     <forum id="1">
29       <discussion id="1"/>
30     </forum>
31   </request>
32 </course>
33 </stbmoodle>

```

Figura 23: XML para requisição de uma discussão

O atributo *id* do elemento *discussion* armazena o identificador da discussão. O *plug-in* receberá o XML e irá gerar outro contendo todas as postagens da discussão selecionada, conforme Figura 24.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4      <!ELEMENT head (moodleserver, device)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED>
7        <!ELEMENT device EMPTY>
8          <!ATTLIST device serialnumber CDATA #REQUIRED>
9      <!ELEMENT course (request)>
10        <!ATTLIST course id CDATA #REQUIRED>
11        <!ELEMENT request (forum)>
12          <!ATTLIST request id CDATA #REQUIRED>
13          <!ELEMENT forum (discussion)>
14            <!ATTLIST forum id CDATA #REQUIRED>
15            <!ELEMENT discussion (post+)>
16              <!ATTLIST discussion id CDATA #REQUIRED>
17              <!ELEMENT post (subject, message, user, date)>
18                <!ELEMENT subject (#PCDATA)>
19                <!ELEMENT message (#PCDATA)>
20                <!ELEMENT user (#PCDATA)>
21                <!ELEMENT date (#PCDATA)>
22      ]>
23 <moodlestb>
24 <head>
25   <moodleserver ip="192.168.2.100"/>
26   <device serialnumber="123456xx"/>
27 </head>
28 <course id="2">
29   <request id="6">
30     <forum id="1">
31       <discussion id="2">
32         <post>
33           <subject>Último dia para apresentação</subject>
34           <message>Último dia: 14 de dezembro.</message>
35           <user>João Costa</user>
36           <date>1253898695</date>
37         </post>
38         <post>
39           <subject>Re: Último dia para apresentação</subject>
40           <message>Não dá pra adiar?</message>
41           <user>Carla Silva</user>
42           <date>1253898732</date>
43         </post>
44       </discussion>
45     </forum>
46   </request>
47 </course>
48 </moodlestb>

```

Figura 24: XML de resposta contendo as postagens da discussão escolhida

O elemento *post* representa uma postagem que contém assunto (elemento *subject*), mensagem (elemento *message*), usuário (elemento *user*) e data (elemento *date*), sendo que assunto representa o nome da discussão, usuário e data são atributos recuperados e mensagem é de preenchimento obrigatório. Ao se criar uma discussão, é obrigatório informar uma mensagem, ou seja, uma postagem. Por isso, cada discussão deverá ter no mínimo uma postagem representado por *post+* na Figura 24, linha 15.

A *query* para exibição dos fóruns de um curso é exibida na Figura 25.

```
1 SELECT f.id, f.name, f.intro, count(d.id)
2 FROM forum f,
3      forum_discussions d,
4      course_modules cm,
5      modules m
6 WHERE f.id = d.forum AND
7        m.name = 'forum' AND
8        cm.module = m.id AND
9        cm.visible = 1 AND
10       f.id = cm.instance AND
11       cm.course = 2
12 GROUP BY f.id, f.name, f.intro;
```

Figura 25: *Query* para listagem dos fóruns de um curso

Assim como no exemplo da atividade chat, a linha 9 da Figura 25 delimita a exibição apenas dos fóruns visíveis. A *query* para a seleção de todas as discussões de um fórum é mostrada na Figura 26.

```
1 SELECT d.id, d.name, u.firstname, u.lastname, (count(p.id)-1)
2 FROM forum_discussions d
3 JOIN forum_posts p ON p.discussion = d.id
4 JOIN user u ON d.userid = u.id
5 WHERE d.forum = 1
6 GROUP BY d.id, d.name, u.firstname, u.lastname;
```

Figura 26: *Query* para exibição das discussões do fórum escolhido

A Figura 27 mostra a *query* para exibição das postagens de uma discussão. Dessa forma, é encerrada a atividade de visualização de um fórum.

```
1 SELECT p.subject, p.message, u.firstname, u.lastname, p.modified
2 FROM forum_posts p
3 JOIN user u ON p.userid = u.id
4 WHERE p.discussion = 2;
```

Figura 27: *Query* para exibição das postagens da discussão escolhida

5.5 Escolha da atividade Glossários

Caso o usuário opte pela atividade glossários, o aplicativo gera um XML de requisição conforme Figura 28.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE stbmoodle[
3    <!ELEMENT stbmoodle (head, course)>
4      <!ELEMENT head (device, moodleserver, userlogged)>
5        <!ELEMENT device EMPTY>
6          <!ATTLIST device serialnumber CDATA #REQUIRED>
7        <!ELEMENT moodleserver EMPTY>
8          <!ATTLIST moodleserver ip CDATA #REQUIRED>
9        <!ELEMENT userlogged EMPTY>
10         <!ATTLIST userlogged id CDATA #REQUIRED>
11      <!ELEMENT course (request)>
12        <!ATTLIST course id CDATA #REQUIRED>
13        <!ELEMENT request EMPTY>
14          <!ATTLIST request id CDATA #REQUIRED>
15    ]>
16  <stbmoodle>
17    <head>
18      <device serialnumber="123456xx"/>
19      <moodleserver ip="192.168.2.100"/>
20      <userlogged id="101"/>
21    </head>
22    <course id="2">
23      <request id="7"/>
24    </course>
25  </stbmoodle>

```

Figura 28: XML para requisição da atividade glossários

O elemento *request* recebe o valor 7 em seu atributo *id*. O XML é recebido pelo *plugin* que gera um arquivo de resposta com a relação dos glossários do curso, conforme Figuras 29 e 30.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4      <!ELEMENT head (moodleserver, device)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED>
7        <!ELEMENT device EMPTY>
8          <!ATTLIST device serialnumber CDATA #REQUIRED>
9      <!ELEMENT course (request)>
10        <!ATTLIST course id CDATA #REQUIRED>
11        <!ELEMENT request (glossary+)>
12          <!ATTLIST request id CDATA #REQUIRED>
13          <!ELEMENT glossary (name, intro, countitems)>
14            <!ATTLIST glossary id CDATA #REQUIRED>
15            <!ELEMENT name (#PCDATA)>
16            <!ELEMENT intro (#PCDATA)>
17            <!ELEMENT countitems (#PCDATA)>
18    ]>

```

Figura 29: XML de resposta com os glossários disponíveis no curso (Parte I)

```

19 <moodlestb>
20 <head>
21   <moodleserver ip="192.168.2.100"/>
22   <device serialnumber="200910xs"/>
23 </head>
24 <course id="2">
25   <request id="7">
26     <glossary id="1">
27       <name>Termos_1</name>
28       <intro>Termos para o trabalho 1.</intro>
29       <countitems>3</countitems>
30     </glossary>
31     <glossary id="2">
32       <name>Termos_2</name>
33       <intro>Termos para o trabalho 2</intro>
34       <countitems>2</countitems>
35     </glossary>
36   </request>
37 </course>
38 </moodlestb>

```

Figura 30: XML de resposta com os glossários disponíveis no curso (Parte II)

Cada glossário é representado pelo elemento *glossary* que possui identificador (atributo *id*), nome (elemento *name*), descrição (elemento *intro*) e total de itens (elemento *countitems*), sendo identificador atribuído pelo Moodle, nome e descrição de preenchimento obrigatório para criação de um glossário e total de itens um atributo calculado. Neste caso, assim como nos anteriores, é obrigatória a presença de no mínimo um glossário indicado por *glossary+* na linha 11 da Figura 29.

O usuário deverá selecionar um glossário e um XML de requisição é enviado ao servidor contendo o identificador do glossário, conforme Figura 31.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3   <!ELEMENT stbmoodle (head, course)>
4   <!ELEMENT head (device, moodleserver, userlogged)>
5     <!ELEMENT device EMPTY>
6     <!-->
7     <!-->
8     <!-->
9     <!-->
10    <!-->
11    <!ELEMENT course (request)>
12    <!-->
13    <!-->
14    <!-->
15    <!-->
16    <!-->
17  ]>
18 <stbmoodle>
19 <head>
20   <device serialnumber="123456xx"/>
21   <moodleserver ip="192.168.2.100"/>
22   <userlogged id="101"/>
23 </head>
24 <course id="2">
25   <request id="7">
26     <glossary id="2"/>
27   </request>
28 </course>
29 </stbmoodle>

```

Figura 31: XML para requisição de um glossário

O *plug-in* recebe o XML e gera um arquivo de resposta com os termos do glossário selecionado, conforme Figura 32.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4    <!ELEMENT head (moodleserver, device)>
5    <!ELEMENT moodleserver EMPTY>
6    <!ATTLIST moodleserver ip CDATA #REQUIRED>
7    <!ELEMENT device EMPTY>
8    <!ATTLIST device serialnumber CDATA #REQUIRED>
9  ]>
10 <course id="2">
11   <request id="7">
12     <glossary id="2">
13       <item*>
14         <!ATTLIST glossary id CDATA #REQUIRED>
15         <ELEMENT item (concept, definition, user)>
16           <ELEMENT concept (#PCDATA)>
17           <ELEMENT definition (#PCDATA)>
18           <ELEMENT user (#PCDATA)>
19       </item*>
20     </glossary>
21   </request>
22 </course>
23 </moodlestb>

```

Figura 32: XML de resposta contendo os termos do glossário escolhido

Cada termo é representado pelo elemento *item* que contém os elementos para conceito (*concept*), definição (*definition*) e usuário que o criou (*user*), sendo conceito e definição de preenchimento obrigatório para a criação de um item e o usuário é identificado pelo *login* armazenado na base de dados. É importante ressaltar que um glossário pode ser criado pelo tutor, mas ainda não conter nenhum termo, assim como pode conter vários. Este fato é representado por *item** na linha 13 da Figura 32. A *query* para seleção dos glossários é apresentada na Figura 33.

```

1  SELECT g.id, g.name, g.intro, count(e.id) AS itens
2  FROM glossary g,
3       glossary_entries e,
4       course_modules cm,
5       modules m
6  WHERE g.id = e.glossaryid AND
7         m.name = 'glossary' AND
8         cm.module = m.id AND
9         cm.visible = 1 AND
10        g.id = cm.instance AND
11        cm.course = 2
12 GROUP BY g.id, g.name, g.intro;

```

Figura 33: Query para listagem dos glossários de um curso

A linha 9 da Figura 33 faz com apenas os glossários marcados como visíveis sejam exibidos aos usuários. Para seleção dos termos de um glossário é adotada a *query* da Figura 34. Assim é encerrada a atividade de consulta ao glossário.

```
1 SELECT e.id, e.concept, e.definition, u.firstname
2 FROM glossary_entries e
3 JOIN user u ON u.id = e.userid
4 WHERE e.glossaryid = 2;
```

Figura 34: *Query* para exibição dos itens do glossário escolhido

5.6 Escolha da atividade Recursos Materiais

Ao optar pela seleção da atividade recursos, é gerado um XML de requisição conforme mostrado na Figura 35.

```
1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3   <!ELEMENT stbmoodle (head, course)>
4     <!ELEMENT head (device, moodleserver, userlogged)>
5       <!ELEMENT device EMPTY>
6         <!ATTLIST device serialnumber CDATA #REQUIRED>
7       <!ELEMENT moodleserver EMPTY>
8         <!ATTLIST moodleserver ip CDATA #REQUIRED>
9       <!ELEMENT userlogged EMPTY>
10        <!ATTLIST userlogged id CDATA #REQUIRED>
11     <!ELEMENT course (request)>
12       <!ATTLIST course id CDATA #REQUIRED>
13     <!ELEMENT request EMPTY>
14       <!ATTLIST request id CDATA #REQUIRED>
15 ]>
16 <stbmoodle>
17   <head>
18     <device serialnumber="123456xx"/>
19     <moodleserver ip="192.168.2.100"/>
20     <userlogged id="101"/>
21   </head>
22   <course id="2">
23     <request id="14"/>
24   </course>
25 </stbmoodle>
```

Figura 35: XML para requisição da atividade recursos

Neste exemplo de seleção, o elemento *request* armazena em seu atributo *id* o identificador 14, referente a atividade recursos. O *plug-in* no servidor irá gerar um XML de resposta com os recursos disponíveis para o curso indicado, conforme Figura 36.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4      <!ELEMENT head (moodleserver, device)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED>
7        <!ELEMENT device EMPTY>
8          <!ATTLIST device serialnumber CDATA #REQUIRED>
9      <!ELEMENT course (request)>
10        <!ATTLIST course id CDATA #REQUIRED>
11        <!ELEMENT request (resource+)>
12          <!ATTLIST request id CDATA #REQUIRED>
13          <!ELEMENT resource (name, summary?)>
14            <!ATTLIST resource id CDATA #REQUIRED>
15            <!ELEMENT name (#PCDATA)>
16            <!ELEMENT summary (#PCDATA)>
17    ]>
18  <moodlestb>
19    <head>
20      <moodleserver ip="192.168.2.100"/>
21      <device serialnumber="123456xx"/>
22    </head>
23    <course id="2">
24      <request id="14">
25        <resource id="1">
26          <name>Aula 1</name>
27          <summary>Introdução</summary>
28        </resource>
29        <resource id="2">
30          <name>Aula 2</name>
31          <summary>Capítulo 1</summary>
32        </resource>
33      </request>
34    </course>
35  </moodlestb>

```

Figura 36: XML de resposta com os recursos disponíveis no curso

Cada recurso material é representado pelo elemento *resource* que contém identificador (atributo *id*), nome (elemento *name*) e sumário (elemento *summary*). O identificador é atribuído pelo Moodle, o nome é de preenchimento obrigatório ao criar um novo recurso e sumário é opcional, sendo representado por *summary?* na linha 13 da Figura 36. Por ser obrigatória a presença de no mínimo um recurso, a linha 11 da Figura 36 contém a representação *resource+*. O usuário deverá selecionar o recurso desejado e o aplicativo irá gerar um XML para requisição, conforme Figura 37.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3   <!ELEMENT stbmoodle (head, course)>
4     <!ELEMENT head (device, moodleserver, userlogged)>
5       <!ELEMENT device EMPTY>
6         <!ATTLIST device serialnumber CDATA #REQUIRED>
7       <!ELEMENT moodleserver EMPTY>
8         <!ATTLIST moodleserver ip CDATA #REQUIRED>
9       <!ELEMENT userlogged EMPTY>
10        <!ATTLIST userlogged id CDATA #REQUIRED>
11     <!ELEMENT course (request)>
12       <!ATTLIST course id CDATA #REQUIRED>
13       <!ELEMENT request (resource)>
14         <!ATTLIST request id CDATA #REQUIRED>
15         <!ELEMENT resource EMPTY>
16         <!ATTLIST resource id CDATA #REQUIRED>
17   ]>
18 <stbmoodle>
19 <head>
20   <device serialnumber="123456xx"/>
21   <moodleserver ip="192.168.2.100"/>
22   <userlogged id="101"/>
23 </head>
24 <course id="2">
25   <request id="14">
26     <resource id="2"/>
27   </request>
28 </course>
29 </stbmoodle>

```

Figura 37: XML para requisição de um recurso

O XML é recebido pelo *plug-in* no servidor. Como se trata de requisição de recurso material é necessário verificar qual o seu tipo, pois conforme o tipo do recurso solicitado será utilizada uma *query*. Os tipos de recurso aceitos pelo Moodle são: arquivos, textos e HTML.

O tipo arquivo exhibe um *link* para o diretório do arquivo e os tipos texto e HTML exibem as páginas de texto ou HTML criadas pelo tutor. A informação do tipo do recurso é armazenada no atributo *type* da tabela *resource* e pode ser *file* (para arquivos e *links* para *sites*), *text* (para páginas de texto simples) e *html* (para páginas web).

Caso o recurso seja uma página de texto simples ou uma página web, seu conteúdo fica armazenado no atributo *alltext* da tabela *resource* e o XML de resposta é exibido na Figura 38.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3      <!ELEMENT moodlestb (head, course)>
4          <!ELEMENT head (moodleserver, device)>
5              <!ELEMENT moodleserver EMPTY>
6                  <!ATTLIST moodleserver ip CDATA #REQUIRED>
7              <!ELEMENT device EMPTY>
8                  <!ATTLIST device serialnumber CDATA #REQUIRED>
9          <!ELEMENT course (request)>
10              <!ATTLIST course id CDATA #REQUIRED>
11              <!ELEMENT request (resource)>
12                  <!ATTLIST request id CDATA #REQUIRED>
13                  <!ELEMENT resource (name, summary?, alltext)>
14                      <!ATTLIST resource id CDATA #REQUIRED>
15                      <!ELEMENT name (#PCDATA)>
16                      <!ELEMENT summary (#PCDATA)>
17                      <!ELEMENT alltext (#PCDATA)>
18      ]>
19  <moodlestb>
20  <head>
21      <moodleserver ip="192.168.2.100"/>
22      <device serialnumber="200910xs"/>
23  </head>
24  <course id="2">
25      <request id="14">
26          <resource id="2">
27              <name>Aula 2</name>
28              <summary>Capítulo 2</summary>
29              <alltext>A engenharia de requisitos (...)</alltext>
30          </resource>
31      </request>
32  </course>
33 </moodlestb>

```

Figura 38: XML de resposta contendo o recurso escolhido (texto simples ou página web)

Para recursos do tipo *file*, é necessário localizar seu diretório e essa informação é armazenada no atributo *reference* da tabela *resource*. Caso o dado deste atributo comece com `http://` indicando se tratar de um *link* para uma página web, basta informar no XML de resposta este valor, do contrário, é necessário fornecer o diretório no Moodle para armazenamento de arquivos.

Os arquivos são armazenados no diretório *moodledata* onde é criada uma pasta para cada curso que disponibilize a atividade recursos. Para nomear esta pasta é utilizado o identificador do curso. Os arquivos são armazenados na pasta de seu curso no diretório indicado pelo atributo *reference* da tabela *resource*. Por exemplo, o tutor do curso Engenharia de Software (cujo identificador é 2), disponibilizou um arquivo de nome `material_Aula1.pdf`. No atributo *reference* da tabela *resource* constará o dado `material_Aula1.pdf`. Sendo assim, o diretório de armazenamento deste recurso no Moodle é `..\moodledata\2\material_Aula1.pdf`. O XML de resposta é mostrado na Figura 39.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4      <!ELEMENT head (moodleserver, device)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED>
7        <!ELEMENT device EMPTY>
8          <!ATTLIST device serialnumber CDATA #REQUIRED>
9      <!ELEMENT course (request)>
10        <!ATTLIST course id CDATA #REQUIRED>
11        <!ELEMENT request (resource)>
12          <!ATTLIST request id CDATA #REQUIRED>
13          <!ELEMENT resource (directory)>
14            <!ATTLIST resource id CDATA #REQUIRED>
15            <!ELEMENT directory (#PCDATA)>
16        ]>
17  <moodlestb>
18  <head>
19    <moodleserver ip="192.168.2.100"/>
20    <device serialnumber="200910xs"/>
21  </head>
22  <course id="2">
23    <request id="14">
24      <resource id="2">
25        <directory>..\moodledata\2\material_Aula1.pdf</directory>
26      </resource>
27    </request>
28  </course>
29 </moodlestb>

```

Figura 39: XML de resposta contendo o recurso escolhido (arquivo ou *link* para web)

O aplicativo da STB ao receber este XML, fará uma requisição de *download* exibindo o recurso em seguida. A *query* para seleção dos recursos é mostrada na Figura 40.

```

1  SELECT r.id, r.name, r.summary
2  FROM resource r,
3       course_modules cm,
4       modules m
5  WHERE m.name = 'resource' AND
6         cm.module = m.id AND
7         cm.visible = 1 AND
8         r.id = cm.instance AND
9         cm.course = 2

```

Figura 40: *Query* para listagem dos recursos de um curso

A condição mostrada na linha 7 da Figura 40 faz com que apenas os recursos visíveis sejam exibidos. Para a seleção de um recurso do tipo texto ou HTML é empregada a *query* da Figura 41.

```

1  SELECT r.name, r.summary, r.alltext
2  FROM resource r,
3       course_modules cm,
4       modules m
5  WHERE m.name = 'resource' AND
6         cm.module = m.id AND
7         r.id = cm.instance AND
8         cm.course = 2 AND
9         r.id = 2

```

Figura 41: *Query* para seleção de recurso (texto simples ou página web)

Para um recurso do tipo arquivo é usada a *query* da Figura 42. Dessa forma se encerra a atividade de visualização de um recurso material.

```
1 SELECT r.reference
2 FROM resource r,
3     course_modules cm,
4     modules m
5 WHERE m.name = 'resource' AND
6     cm.module = m.id AND
7     r.id = cm.instance AND
8     cm.course = 2 AND
9     r.id = 2
```

Figura 42: *Query* para seleção de recurso (arquivo ou link para web)

5.7 Escolha da atividade Wikis

Caso o usuário escolha acessar a atividade wikis, o XML de requisição gerado será como mostrado na Figura 43.

```
1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3     <!ELEMENT stbmoodle (head, course)>
4         <!ELEMENT head (device, moodleserver, userlogged)>
5             <!ELEMENT device EMPTY>
6                 <!ATTLIST device serialnumber CDATA #REQUIRED>
7             <!ELEMENT moodleserver EMPTY>
8                 <!ATTLIST moodleserver ip CDATA #REQUIRED>
9             <!ELEMENT userlogged EMPTY>
10                 <!ATTLIST userlogged id CDATA #REQUIRED>
11         <!ELEMENT course (request)>
12             <!ATTLIST course id CDATA #REQUIRED>
13             <!ELEMENT request EMPTY>
14                 <!ATTLIST request id CDATA #REQUIRED>
15     ]>
16 <stbmoodle>
17 <head>
18     <device serialnumber="123456xx"/>
19     <moodleserver ip="192.168.2.100"/>
20     <userlogged id="101"/>
21 </head>
22 <course id="2">
23     <request id="17"/>
24 </course>
25 </stbmoodle>
```

Figura 43: XML para requisição da atividade wikis

O elemento *request* irá armazenar em seu atributo *id* o identificador referente à atividade wikis. O XML de requisição será recebido pelo *plug-in* do Moodle que irá gerar o XML de resposta com a listagem das wikis do curso, conforme Figura 44.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3      <!ELEMENT moodlestb (head, course)>
4          <!ELEMENT head (moodleserver, device)>
5              <!ELEMENT moodleserver EMPTY>
6                  <!ATTLIST moodleserver ip CDATA #REQUIRED>
7              <!ELEMENT device EMPTY>
8                  <!ATTLIST device serialnumber CDATA #REQUIRED>
9          <!ELEMENT course (request)>
10              <!ATTLIST course id CDATA #REQUIRED>
11              <!ELEMENT request (wiki+)>
12                  <!ATTLIST request id CDATA #REQUIRED>
13                  <!ELEMENT wiki (name, summary)>
14                      <!ATTLIST wiki id CDATA #REQUIRED>
15                      <!ELEMENT name (#PCDATA)>
16                      <!ELEMENT summary (#PCDATA)>
17      ]>
18  <moodlestb>
19  <head>
20      <moodleserver ip="192.168.2.100"/>
21      <device serialnumber="200910xs"/>
22  </head>
23  <course id="2">
24      <request id="17">
25          <wiki id="1">
26              <name>Wiki 1</name>
27              <summary>Wiki da matéria para a primeira prova.</summary>
28          </wiki>
29          <wiki id="2">
30              <name>Wiki 2</name>
31              <summary>Wiki da matéria para a segunda prova.</summary>
32          </wiki>
33      </request>
34  </course>
35  </moodlestb>

```

Figura 44: XML de resposta com as wikis disponíveis no curso

As wikis são representadas pelos elementos *wiki* que contêm identificador (atributo *id*), nome (elemento *name*) e sumário (elemento *summary*). O identificador é atribuído pelo Moodle e nome e sumário são de preenchimento obrigatório para criação de uma wiki. Para indicar a obrigatoriedade de pelo menos uma wiki, na linha 11 da Figura 44 é representado *wiki+*. Após seleção de uma das wikis, o aplicativo da TVDI irá gerar um XML de requisição, conforme Figura 45.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE stbmoodle[
3    <!ELEMENT stbmoodle (head, course)>
4      <!ELEMENT head (device, moodleserver, userlogged)>
5        <!ELEMENT device EMPTY>
6          <!ATTLIST device serialnumber CDATA #REQUIRED>
7        <!ELEMENT moodleserver EMPTY>
8          <!ATTLIST moodleserver ip CDATA #REQUIRED>
9        <!ELEMENT userlogged EMPTY>
10         <!ATTLIST userlogged id CDATA #REQUIRED>
11      <!ELEMENT course (request)>
12        <!ATTLIST course id CDATA #REQUIRED>
13        <!ELEMENT request (wiki)>
14          <!ATTLIST request id CDATA #REQUIRED>
15          <!ELEMENT wiki EMPTY>
16            <!ATTLIST wiki id CDATA #REQUIRED>
17    ]>
18  <stbmoodle>
19    <head>
20      <device serialnumber="123456xx"/>
21      <moodleserver ip="192.168.2.100"/>
22      <userlogged id="101"/>
23    </head>
24    <course id="2">
25      <request id="17">
26        <wiki id="1"/>
27      </request>
28    </course>
29  </stbmoodle>

```

Figura 45: XML para requisição de uma wiki

O atributo *id* do elemento *wiki* armazena o identificador da wiki escolhida. O XML é recebido pelo *plug-in* no servidor que irá gerar um arquivo de resposta, conforme Figuras 46 e 47.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4      <!ELEMENT head (moodleserver, device)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED>
7        <!ELEMENT device EMPTY>
8          <!ATTLIST device serialnumber CDATA #REQUIRED>
9      <!ELEMENT course (request)>
10        <!ATTLIST course id CDATA #REQUIRED>
11        <!ELEMENT request (wiki)>
12          <!ATTLIST request id CDATA #REQUIRED>
13          <!ELEMENT wiki (pagename, content)>
14            <!ATTLIST wiki id CDATA #REQUIRED>
15            <!ELEMENT pagename (#PCDATA)>
16            <!ELEMENT content (#PCDATA)>
17    ]>

```

Figura 46: XML de resposta contendo a wiki escolhida (Parte I)

```

18 <moodlestb>
19 <head>
20   <moodleserver ip="192.168.2.100"/>
21   <device serialnumber="200910xs"/>
22 </head>
23 <course id="2">
24   <request id="17">
25     <wiki id="1">
26       <pagename>Wiki 1</pagename>
27       <content>0 levantamento de requisitos (...)</content>
28     </wiki>
29   </request>
30 </course>
31 </moodlestb>

```

Figura 47: XML de resposta contendo a wiki escolhida (Parte II)

Neste arquivo é informado o nome da wiki escolhida (elemento *pagename*) e seu conteúdo (elemento *content*). A *query* para mostrar a relação de todas as wikis visíveis de um curso é mostrada na Figura 48.

```

1 SELECT w.id, w.name, w.summary
2 FROM wiki w,
3     course_modules cm,
4     modules m
5 WHERE m.name = 'wiki' AND
6       cm.module = m.id AND
7       cm.visible = 1 AND
8       w.id = cm.instance AND
9       cm.course = 2

```

Figura 48: Query para listagem das wikis de um curso

A Figura 49 contém a *query* para exibição da wiki escolhida. Dessa forma é encerrada a atividade de visualização de uma wiki.

```

1 SELECT p.pagename, p.content
2 FROM wiki_pages p
3 JOIN wiki w ON p.pagename = w.pagename
4 WHERE w.id = 2 AND p.version >= ALL (SELECT version
5                                     FROM wiki_pages p
6                                     JOIN wiki w ON p.pagename = w.pagename
7                                     WHERE w.id = 2);

```

Figura 49: Query para seleção da wiki

5.8 Escolha da função Perfil

Para visualização do perfil, o usuário deve acessar a função perfil e um XML de requisição é enviado, conforme Figura 50.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3   <!ELEMENT stbmoodle (head, course)>
4     <!ELEMENT head (device, moodleserver, userlogged)>
5       <!ELEMENT device EMPTY>
6         <!ATTLIST device serialnumber CDATA #REQUIRED>
7       <!ELEMENT moodleserver EMPTY>
8         <!ATTLIST moodleserver ip CDATA #REQUIRED>
9       <!ELEMENT userlogged EMPTY>
10        <!ATTLIST userlogged id CDATA #REQUIRED>
11     <!ELEMENT course (request)>
12       <!ATTLIST course id CDATA #REQUIRED>
13     <!ELEMENT request EMPTY>
14       <!ATTLIST request id CDATA #REQUIRED>
15 ]>
16 <stbmoodle>
17 <head>
18   <device serialnumber="123456xx"/>
19   <moodleserver ip="192.168.2.100"/>
20   <userlogged id="101"/>
21 </head>
22 <course id="2">
23   <request id="20"/>
24 </course>
25 </stbmoodle>

```

Figura 50: XML para requisição da função perfil

O atributo *id* do elemento *request* recebe o valor 20 que representa a função perfil.

O *plug-in* gera um XML de resposta como mostrado nas Figuras 51 e 52.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE moodlestb[
3   <!ELEMENT moodlestb (head, course)>
4     <!ELEMENT head (moodleserver, device)>
5       <!ELEMENT moodleserver EMPTY>
6         <!ATTLIST moodleserver ip CDATA #REQUIRED>
7       <!ELEMENT device EMPTY>
8         <!ATTLIST device serialnumber CDATA #REQUIRED>
9     <!ELEMENT course (request)>
10       <!ATTLIST course id CDATA #REQUIRED>
11     <!ELEMENT request (name, description?, country, city, address?,
12       phone1?, phone2?, email, url?, icq?, skype?,
13       yahoo?, aim?, msn?, courses, functions,
14       picture, imagealt)>
15       <!ATTLIST request id CDATA #REQUIRED>
16       <!ELEMENT name (#PCDATA)>
17       <!ELEMENT description (#PCDATA)>
18       <!ELEMENT country (#PCDATA)>
19       <!ELEMENT city (#PCDATA)>
20       <!ELEMENT address (#PCDATA)>
21       <!ELEMENT phone1 (#PCDATA)>
22       <!ELEMENT phone2 (#PCDATA)>
23       <!ELEMENT email (#PCDATA)>
24       <!ELEMENT url (#PCDATA)>
25       <!ELEMENT icq (#PCDATA)>
26       <!ELEMENT skype (#PCDATA)>
27       <!ELEMENT yahoo (#PCDATA)>
28       <!ELEMENT aim (#PCDATA)>
29       <!ELEMENT msn (#PCDATA)>
30       <!ELEMENT courses (#PCDATA)>
31       <!ELEMENT functions (#PCDATA)>
32       <!ELEMENT picture (#PCDATA)>
33       <!ELEMENT imagealt (#PCDATA)>
34 ]>

```

Figura 51: XML de resposta com a função perfil (Parte I)

```

35 <moodlestb>
36 <head>
37   <moodleserver ip="192.168.2.100"/>
38   <device serialnumber="200910xs"/>
39 </head>
40 <course id="2">
41   <request id="20">
42     <name>Paula Marin Leite</name>
43     <description>Estudante</description>
44     <country>BR</country>
45     <city>Juiz de Fora</city>
46     <address>Rua Carlota Malta</address>
47     <phone1>(32) 8845-1366</phone1>
48     <email>paula.marinleite@gmail.com</email>
49     <skype>paula-marin</skype>
50     <msn>paula.marin@hotmail.com</msn>
51     <courses>Engenharia de Software, Banco de Dados</courses>
52     <functions>Estudante</functions>
53     <picture>1</picture>
54     <imagealt>..\moodledata\user\0\234\</imagealt>
55   </request>
56 </course>
57 </moodlestb>

```

Figura 52: XML de resposta com a função perfil (Parte II)

As informações nome (elemento *name*), e-mail (elemento *email*), cidade (elemento *city*) e país (elemento *country*) são de preenchimento obrigatório, conforme Figura 51, linhas de 11 a 14. Caso o usuário tenha inserido uma foto, o elemento *picture* armazenará valor 1. O Moodle salva a imagem com nome f1.jpg (em tamanho maior) e f2.jpg (em tamanho menor) no diretório `..\moodledata\user`. Neste diretório são criadas pastas conforme o total de alunos cadastrados. Os alunos cujo identificador (atributo *id* da tabela *user*) vai de 0 a 999 são armazenados na pasta 0, para identificadores de 1000 a 1999 é criada a pasta 1000, a pasta 2000 armazena alunos de identificador de 2000 a 2999 e assim por diante. Dentro dessas pastas há uma outra cujo nome é o identificador do usuário. Por exemplo, a usuária Paula Marin Leite (cujo identificador é 234) inseriu uma imagem. Logo, o diretório de armazenamento desta imagem é `..\moodledata\user\0\234\`. Para os usuários que não inseriram foto é adotada a imagem padrão do Moodle que está armazenada no diretório `..\user\default\` e, neste caso, o elemento *picture* conterá valor 0.

O aplicativo da STB, após receber este XML, fará uma requisição de *download* da imagem exibindo-a ao usuário e encerrando a função de visualização de perfil.

A *query* para exibição do perfil é mostrada na figura 53.

```

1 SELECT firstname, lastname, description, country,
2         city, address, phone1, phone2, email, url,
3         icq, skype, yahoo, aim, msn, picture
4 FROM user
5 WHERE id=234;

```

Figura 53: Query para função perfil

Para exibir a relação das funções do usuário (elemento *functions*) é adotada a query da Figura 54.

```

1 SELECT r.name
2 FROM role r
3 JOIN role_assignments a ON a.roleid=r.id
4 WHERE a.userid = 234
5 GROUP BY r.name;

```

Figura 54: Query para exibição das funções

A query com relação dos cursos (elemento *courses*) é uma adaptação da Figura 7, sendo exibida na Figura 55.

```

1 SELECT co.fullname
2 FROM user u
3 JOIN role_assignments ra ON ra.userid = u.id
4 JOIN context c ON c.id = ra.contextid
5 JOIN course co ON co.id = c.instanceid
6 WHERE c.contextlevel = 50 AND u.id = 234

```

Figura 55: Query para relação de cursos (adaptada)

5.9 Escolha da função Participantes

Para exibir a lista dos inscritos em um curso, o usuário deverá escolher a função participantes e o XML de requisição é mostrado na Figura 56.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3     <!ELEMENT stbmoodle (head, course)>
4         <!ELEMENT head (device, moodleserver, userlogged)>
5             <!ELEMENT device EMPTY>
6                 <!ATTLIST device serialnumber CDATA #REQUIRED>
7             <!ELEMENT moodleserver EMPTY>
8                 <!ATTLIST moodleserver ip CDATA #REQUIRED>
9             <!ELEMENT userlogged EMPTY>
10                 <!ATTLIST userlogged id CDATA #REQUIRED>
11         <!ELEMENT course (request)>
12             <!ATTLIST course id CDATA #REQUIRED>
13             <!ELEMENT request EMPTY>
14                 <!ATTLIST request id CDATA #REQUIRED>
15     ]>
16 <stbmoodle>
17 <head>
18     <device serialnumber="123456xx"/>
19     <moodleserver ip="192.168.2.100"/>
20     <userlogged id="101"/>
21 </head>
22 <course id="2">
23     <request id="21"/>
24 </course>
25 </stbmoodle>

```

Figura 56: XML para requisição da função participantes

Neste caso, o elemento *request* armazena em seu atributo *id* o identificador 21 que representa a função participantes.

O *plug-in* recebe o XML e gera um arquivo de resposta com a relação de todos os inscritos, conforme Figura 57.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3    <!ELEMENT moodlestb (head, course)>
4      <!ELEMENT head (moodleserver, device)>
5        <!ELEMENT moodleserver EMPTY>
6          <!ATTLIST moodleserver ip CDATA #REQUIRED>
7        <!ELEMENT device EMPTY>
8          <!ATTLIST device serialnumber CDATA #REQUIRED>
9      <!ELEMENT course (request)>
10        <!ATTLIST course id CDATA #REQUIRED>
11        <!ELEMENT request (assignment+)>
12          <!ATTLIST request id CDATA #REQUIRED>
13          <!ELEMENT assignment (name, city, country, picture, imagealt)>
14            <!ELEMENT name (#PCDATA)>
15            <!ELEMENT city (#PCDATA)>
16            <!ELEMENT country (#PCDATA)>
17            <!ELEMENT picture (#PCDATA)>
18            <!ELEMENT imagealt (#PCDATA)>
19    ]>
20
21 <moodlestb>
22 <head>
23   <moodleserver ip="192.168.2.100"/>
24   <device serialnumber="200910xs"/>
25 </head>
26 <course id="2">
27   <request id="21">
28     <assignment>
29       <name>Paula Marin Leite</name>
30       <city>Juiz de Fora</city>
31       <country>BR</country>
32       <picture>1</picture>
33       <imagealt>..\moodledata\user\0\234</imagealt>
34     </assignment>
35     <assignment>
36       <name>José Silva</name>
37       <city>Juiz de Fora</city>
38       <country>BR</country>
39       <picture>0</picture>
40       <imagealt>..\user\default</imagealt>
41     </assignment>
42   </request>
43 </course>
44 </moodlestb>

```

Figura 57: XML de resposta com os participantes do curso

O aplicativo da STB exibe as informações recebidas e faz uma requisição de *download* da imagem, conforme feito na função perfil, encerrando a função participantes.

A Figura 58 mostra a *query* com a relação dos participantes.

```
1 SELECT u.firstname, u.lastname, u.city, u.country, u.picture
2 FROM user u
3 JOIN role_assignments ra ON ra.userid = u.id
4 JOIN context c          ON c.id = ra.contextid
5 WHERE c.contextlevel = 50 AND c.instanceid = 2
```

Figura 58: *Query* para listagem dos participantes

5.10 Escolha da função Mensagens

Para visualizar as mensagens recebidas e ainda não lidas, o usuário deve escolher a função mensagens e o XML de requisição é exibido na Figura 59.

```
1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3   <!ELEMENT stbmoodle (head, course)>
4     <!ELEMENT head (device, moodleserver, userlogged)>
5       <!ELEMENT device EMPTY>
6         <!ATTLIST device serialnumber CDATA #REQUIRED>
7       <!ELEMENT moodleserver EMPTY>
8         <!ATTLIST moodleserver ip CDATA #REQUIRED>
9       <!ELEMENT userlogged EMPTY>
10        <!ATTLIST userlogged id CDATA #REQUIRED>
11     <!ELEMENT course (request)>
12       <!ATTLIST course id CDATA #REQUIRED>
13     <!ELEMENT request EMPTY>
14       <!ATTLIST request id CDATA #REQUIRED>
15 ]>
16 <stbmoodle>
17   <head>
18     <device serialnumber="123456xx"/>
19     <moodleserver ip="192.168.2.100"/>
20     <userlogged id="101"/>
21   </head>
22   <course id="2">
23     <request id="22"/>
24   </course>
25 </stbmoodle>
```

Figura 59: XML para requisição da função mensagens

O atributo *id* do elemento *request* armazenará o valor 22, referente a função mensagens.

O *plug-in* recebe o XML e gera um arquivo de resposta, conforme Figura 60, encerrando a função.

```

1  <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2  <!DOCTYPE moodlestb[
3      <!ELEMENT moodlestb (head, course)>
4          <!ELEMENT head (moodleserver, device)>
5              <!ELEMENT moodleserver EMPTY>
6                  <!ATTLIST moodleserver ip CDATA #REQUIRED>
7              <!ELEMENT device EMPTY>
8                  <!ATTLIST device serialnumber CDATA #REQUIRED>
9          <!ELEMENT course (request)>
10              <!ATTLIST course id CDATA #REQUIRED>
11              <!ELEMENT request (name, date, message)>
12                  <!ATTLIST request id CDATA #REQUIRED>
13                  <!ELEMENT name (#PCDATA)>
14                  <!ELEMENT date (#PCDATA)>
15                  <!ELEMENT message (#PCDATA)>
16      ]>
17  <moodlestb>
18  <head>
19      <moodleserver ip="192.168.2.100"/>
20      <device serialnumber="200910xs"/>
21  </head>
22  <course id="2">
23      <request id="22">
24          <name>José Silva</name>
25          <date>1259759760</date>
26          <message>Oi Paula! Você pode fazer o trabalho conosco hoje?</message>
27      </request>
28  </course>
29  </moodlestb>

```

Figura 60: XML de resposta com as mensagens

A query para exibição das mensagens é mostrada na Figura 61.

```

1  SELECT u.firstname, u.lastname, m.timecreated, m.message
2  FROM message m
3  JOIN user u ON m.useridfrom=u.id
4  WHERE m.useridto=234;

```

Figura 61: Query para listagem das mensagens

5.11 Escolha da função Nota

Caso o usuário queira visualizar suas notas, deverá escolher a funcionalidade nota e será gerado um XML de requisição conforme Figura 62.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE stbmoodle[
3   <!ELEMENT stbmoodle (head, course)>
4     <!ELEMENT head (device, moodleserver, userlogged)>
5       <!ELEMENT device EMPTY>
6         <!ATTLIST device serialnumber CDATA #REQUIRED>
7       <!ELEMENT moodleserver EMPTY>
8         <!ATTLIST moodleserver ip CDATA #REQUIRED>
9       <!ELEMENT userlogged EMPTY>
10        <!ATTLIST userlogged id CDATA #REQUIRED>
11     <!ELEMENT course (request)>
12       <!ATTLIST course id CDATA #REQUIRED>
13       <!ELEMENT request EMPTY>
14         <!ATTLIST request id CDATA #REQUIRED>
15   ]>
16 <stbmoodle>
17 <head>
18   <device serialnumber="123456xx"/>
19   <moodleserver ip="192.168.2.100"/>
20   <userlogged id="101"/>
21 </head>
22 <course id="2">
23   <request id="23"/>
24 </course>
25 </stbmoodle>

```

Figura 62: XML para requisição da função nota

O atributo *id* do elemento *request* deverá conter o valor 23, identificador da funcionalidade nota.

O XML será recebido pelo *plug-in* que irá gerar um arquivo de resposta, conforme Figuras 63 e 64, encerrando a função.

```

1 <?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
2 <!DOCTYPE moodlestb[
3   <!ELEMENT moodlestb (head, course)>
4     <!ELEMENT head (moodleserver, device)>
5       <!ELEMENT moodleserver EMPTY>
6         <!ATTLIST moodleserver ip CDATA #REQUIRED>
7       <!ELEMENT device EMPTY>
8         <!ATTLIST device serialnumber CDATA #REQUIRED>
9     <!ELEMENT course (request)>
10       <!ATTLIST course id CDATA #REQUIRED>
11       <!ELEMENT request (item*)>
12         <!ATTLIST request id CDATA #REQUIRED>
13         <!ELEMENT item (exercise, grademax, grademin, rawgrade, finalgrade, feedback?)>
14           <!ELEMENT exercise (#PCDATA)>
15           <!ELEMENT grademax (#PCDATA)>
16           <!ELEMENT grademin (#PCDATA)>
17           <!ELEMENT rawgrade (#PCDATA)>
18           <!ELEMENT finalgrade (#PCDATA)>
19           <!ELEMENT feedback (#PCDATA)>
20 ]>

```

Figura 63: XML de resposta com as notas (Parte I)

```

21 <moodlestb>
22 <head>
23     <moodleserver ip="192.168.2.100"/>
24     <device serialnumber="200910xs"/>
25 </head>
26 <course id="2">
27     <request id="22">
28         <item>
29             <exercise>Tarefa 1</exercise>
30             <grademax>100</grademax>
31             <grademin>0</grademin>
32             <rawgrade>75</rawgrade>
33             <finalgrade>75</finalgrade>
34         </item>
35     </request>
36 </course>
37 </moodlestb>

```

Figura 64: XML de resposta com as notas (Parte II)

A função nota pode não exibir nenhum item (caso não haja lançamento ou esteja invisível para o usuário) ou exibir vários, sendo representado por *item** na linha 11 da Figura 63. Cada item armazena o nome da tarefa avaliada (elemento *exercise*), a nota máxima (elemento *grademax*), a nota mínima (elemento *grademin*), a nota obtida (elemento *rawgrade*), a média final (elemento *finalgrade* - calculado) e pode apresentar um *feedback* (elemento *feedback*). A não obrigatoriedade do *feedback* é representada por *feedback?* na linha 13 da Figura 63.

A *query* para exibição das notas visíveis ao aluno é mostrada na Figura 65.

```

1 SELECT i.itemname, i.grademax, i.grademin, g.rawgrade, g.finalgrade, g.feedback
2 FROM grade_grades g
3 JOIN grade_items i ON i.id = g.itemid
4 WHERE i.courseid = 2 AND g.userid = 234 AND g.hidden = 0;

```

Figura 65: *Query* para listagem das notas

6 Conclusão

Neste trabalho foi proposto um modelo de integração entre um aplicativo do ambiente web a outro desenvolvido para a TV Digital. Visando contribuir com a inclusão social, um dos objetivos da comissão formadora do SBTVD, o aplicativo escolhido se destina a práticas de EAD, sendo adotado o Moodle, plataforma utilizada pelo NEAD/UFJF.

As funções do Moodle escolhidas permitem apenas visualização de dados, não sendo disponibilizadas aquelas que exijam atualização ou inserção, a fim de que possam ser exibidas em STB com interatividade restrita.

Para a criação das *queries* de consulta ao banco de dados foram adotados os relacionamentos necessários, garantindo otimização no tempo de resposta. Uma dificuldade encontrada foi a identificação do relacionamento entre as tabelas do Moodle, pois não possuem chave estrangeira. Desta forma, as *queries* para inserção e atualização devem manipular outras tabelas, além das utilizadas para cada consulta.

Certas funcionalidades requerem acesso a arquivos, como a função perfil e participantes que exibem a imagem do usuário e a atividade recursos do tipo *file*. Sendo assim, a recuperação do caminho absoluto destes arquivos ainda é um desafio, devido a possibilidade do administrador da plataforma alterar o local do diretório de arquivos do Moodle, o *moodledata*.

Este trabalho foi desenvolvido de forma a permitir integração de qualquer aplicativo em qualquer ambiente à TV Digital, sendo necessário para tal, desenvolver um aplicativo para a STB, outro para o PSI e um *plug-in* para o ambiente escolhido. A comunicação entre os módulos é feita em XML, por permitir integração entre aplicações de diversas linguagens.

Os arquivos XML apresentados estão sendo validados por aplicações que serão inseridas no servidor do Moodle, no PSE e na STB. Tais aplicações encontram-se em fase de desenvolvimento no LApIC/UFJF e o módulo de *login* já foi validado, demonstrando a viabilidade do trabalho.

Como trabalhos futuros são sugeridas:

- A integração das demais funções da plataforma Moodle ao aplicativo de EAD da TV Digital;
- A especificação para acesso a objetos de aprendizagem (IEEE/LTSC, 2001). Objeto de aprendizagem pode ser definido como qualquer recurso pedagógico utilizado na

educação. O ideal é possuir um repositório para armazenamento desses objetos, seguindo um padrão de identificação, a fim de permitir seu reaproveitamento em outros contextos. Tal especificação não foi incluída no trabalho devido ao fato de o NEAD não adotar, atualmente, um padrão de armazenamento.

- A implementação de um mecanismo de segurança para a troca dos arquivos XML. A questão da segurança não foi abordada no trabalho, pois ainda é aguardada a decisão do SBTVD sobre os recursos a serem adotados.

REFERÊNCIAS BIBLIOGRÁFICAS

ABRAEAD (2008). **Um em cada 73 brasileiros estuda a distância**. Disponível em: <<http://www.abraead.com.br/noticias.cod=x1.asp>>. Acesso em: 25 ago. 2009.

BARRÉRE, E., SANTOS Jr.; J. B.; BUENO, L. M. (2008). **Adaptação das Funcionalidades de um AVA para um Ambiente de TVDi**. In: V Seminário Municipal de Informática Educativa, SMIE 2008. Fortaleza, Brasil.

BARRÉRE, E., LEITE, P.M. (2009). **Metodologia de Integração entre Aplicações Web e Aplicações para TV Digital**. In: VI Simpósio de Excelência em Gestão e Tecnologia, SEGeT, 2009. Resende, Brasil.

BARRÉRE, E., LEITE, P.M. (2009). **Especificação de um Mecanismo de Integração entre o Moodle e uma Aplicação de TV Digital Interativa**. In: XX Simpósio Brasileiro de Informática na Educação, SBIE 2009. Florianópolis, Brasil.

BECKER, V. et al. (2004) **Júri Virtual I2TV: Uma Aplicação para TV Digital Interativa baseada em JavaTV e HyperProp**. In webmídia 2004, 2004, Ribeirão Preto. Proceedings WebMedia & LA-Web 2004, 2004. v. 2. p. 12-19.

BELLONI, M.L. (2008). **Educação a distância**. 5. ed. Campinas: Editora Autores Associados, 2008.

BRASIL (1996). **Lei nº. 9.394, de 20 de dezembro de 1996**. Disponível em: <<http://www.planalto.gov.br/ccivil/LEIS/L9394.htm>>. Acesso em: 16 jun. 2009.

BRASIL (2003). **Decreto nº. 4.901, de 26 de novembro de 2003**. Disponível em: <<http://www.planalto.gov.br/CCIVIL/decreto/2003/D4901.htm>>. Acesso em: 25 nov. 2009.

BRASIL (2006). **Decreto nº. 5.820, de 19 de junho de 2006**. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2004-2006/2006/Decreto/D5820.htm>. Acesso em: 25 nov. 2009.

CPqD (2006). **Arquitetura de referência - Sistema Brasileiro de Televisão Digital Terrestre**. PD.30.12.34A.0001A/RT-13/AA

DTV (2009). **Entenda a TV Digital**. Disponível em: <<http://www.dtv.org.br>>. Acesso em: 07 out. 2009.

GINGA (2006). **Ginga Digital TV Middleware Specification**. Disponível em: <<http://www.ginga.org.br>>. Acesso em: 06 ago. 2009.

IBGE (2006). **Acesso a Transferências de Renda de Programas Sociais**. 2006. Disponível em: <<http://oglobo.globo.com/pais/arquivos/pesquisaibge.pdf>>. Acesso em: 15 set. 2009.

IEEE Learning Technology Standards Committee - IEEE/LTSC (2001). **“IEEE Standard for Learning Object Metadata”**, 2001.

KNIGHT, P.T. (2007). **TV Digital interativa: o canal de retorno que falta**. In: Banco Hoje, setembro de 2007, p.42. Disponível em: <<http://www.tedbr.com/publicacoes/bancohoje2007/bancohoje09-07.htm>>. Acesso em: 20 ago. 2009.

MOODLE (2009). **Sobre o Moodle**. Disponível em: <<http://www.moodle.org>>. Acesso em: 23 set. 2009.

MOODLE, versão 1.9.5+ (2007). Disponível em: <<http://download.moodle.org>>. Acesso em: jun. 2009.

LEITE, P.M., BARRÉRE, E. (2009). **Especificação de Esquemas XML para um Mecanismo de Integração entre o Moodle e uma Aplicação de TV Digital Interativa**. I Simpósio de Televisão Digital, SIMTVD, 2009. Bauru, Brasil.

SANTOS, E.; OKADA, A. (2003). **A construção de ambientes virtuais de aprendizagem: por autorias plurais e gratuitas no ciberespaço**. In: 26ª Reunião Anual da Associação Nacional de Pós-Graduação e Pesquisa em Educação, ANPEd 2003. Poços de Caldas, Brasil.

SOARES, L. F., BARBOSA, S. D. (2008). **TV Digital interativa no Brasil se faz com Ginga: Fundamentos, Padrões, Autoria Declarativa e Usabilidade**. In T. Kowaltowski, & K. Breitman, Atualizações em Informática (pp. 105-174). Rio de Janeiro, RJ: Editora PUC-Rio.

TELEBRASIL (2009). **Breve histórico da EAD no Brasil**. Disponível em: <<http://www.telebrasil.org.br/EAD.pdf>>. Acesso em: 23 set. 2009.

TITTEL, Ed. **Teoria e Problemas de XML**. Traduzido por Ralph Miller Jr.. Porto Alegre: Bookman, 2003. Tradução de: Schaum's Outline of Theory and Problems of XML.