

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

O-Learning TVDi :UM *FRAMEWORK* DE TV DIGITAL INTERATIVA PARA
MANIPULAR OBJETOS DIGITAIS DE APRENDIZAGEM RELACIONADOS AO
ENSINO DA TRIGONOMETRIA

WANDERLAN CARVALHO DE ALBUQUERQUE

MANAUS

2012

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

WANDERLAN CARVALHO DE ALBUQUERQUE

O-Learning TVDi: UM *FRAMEWORK* DE TV DIGITAL INTERATIVA PARA
MANIPULAR OBJETOS DIGITAIS DE APRENDIZAGEM RELACIONADOS AO
ENSINO DA TRIGONOMETRIA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica, área de concentração: Controle e Automação de Sistemas.

Orientador:

Prof. Dr. José Pinheiro de Queiroz Neto

Co-orientador:

Prof. Dr. Antonio da Fonseca de Lira

MANAUS

2012

WANDERLAN CARVALHO DE ALBUQUERQUE

O-Learning TVDi :UM FRAMEWORK DE TV DIGITAL INTERATIVA PARA
MANIPULAR OBJETOS DIGITAIS DE APRENDIZAGEM RELACIONADOS AO
ENSINO DA TRIGONOMETRIA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica, área de concentração: Controle e Automação de Sistemas.

Aprovado em 29 de Junho de 2012.

BANCA EXAMINADORA

Prof. Dr. José Pinheiro de Queiroz Neto

Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - IFAM

Prof. Dr. Antônio da Fonseca de Lira

Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - IFAM

Prof. Dr – Ing. Vicente Ferreira de Lucena Júnior

Universidade Federal do Amazonas – UFAM

Profa. Dra. Andréa Pereira Mendonça

Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - IFAM

Ficha catalográfica elaborada pela Biblioteca Central da UFAM

A345o Albuquerque, Wanderlan Carvalho de

O-Learning TVDi: um Framework de TV digital interativa para manipular objetos digitais de aprendizagem relacionados ao ensino da trigonometria / Wanderlan Carvalho de Albuquerque. - Manaus, AM: UFAM, 2012.

131 f.: il. color. ; 30 cm

Inclui referências.

Dissertação (Mestre em Engenharia Elétrica. Área de concentração: Controle e automação de sistemas). Universidade Federal do Amazonas.

Orientador: Prof. Dr. José Pinheiro de Queiroz Neto. Co-Orientador: Prof. Dr. Antonio da Fonseca de Lira.

1. O-Learning TVDi 2. Framework (Programa de computador) 3. Programação orientada a objetos (Computação) 4. Televisão digital 5. Televisão interativa 6. Sistemas de computação interativos 7. Trigonometria – Estudo e ensino I. Queiroz Neto, José Pinheiro de (Orient.) II. Lira, Antonio da Fonseca de (Co-orient.) III. Título

CDU (2007): 621.397:004.921 (043.3)

Dedico este trabalho à minha família, a meus amigos de luta, a meus amigos do grupo de TV Digital da UFAM e do UNINORTE, ao Prof. José Pinheiro e Antônio Lira e a todos aqueles que, como eu, são fascinados pela educação e aprendizado através de novas tecnologias.

AGRADECIMENTOS

Este trabalho é de grande importância para minha carreira, pois representa o fim de um ciclo de muito aprendizado determinando a próxima direção da minha vida. Portanto quero agradecer de coração primeiramente a Deus pela sabedoria e perseverança para superar os momentos de dificuldades que passei e a todos aqueles que compartilharam comigo em especial:

A meus amigos do grupo de TV digital Antonio Santos, Marcelo Prado, Lady Daiana, Luciano Pinto, Nairon Viana, Orlewilson Bentes, Ricardo Rosa, Vandermi Silva e Walter Simões.

A meus amigos do Centro Universitário do Norte Anderson Esteves, Edson Lira, Emmerson Santa Rita, Luiz Nagata, Ricardo Barbosa, Mário Jorge, Leandro Costa, Robson Costa, Rui Souto e Wollace Picanço.

A todos os professores das disciplinas que cursei.

Aos professores: orientador José Pinheiro e co-orientador Antônio Lira por todo apoio e incentivo.

À toda minha família: e em especial a minha mãe Odaisa, meu pai Waldemir , minha esposa Neila e a meu primo Thiago.

À minha filha Naylê pela fonte de inspiração e motivação surgida durante esta trajetória de muita luta.

Numa caminhada só existem duas alternativas; seguir adiante ou parar. Aqueles que param nunca descobrirão a beleza que se revela ao desbravar novos horizontes. Aqueles que continuam sempre renovarão as suas forças e ainda poderão abrir caminhos desconhecidos por onde muitos passarão, ainda que eles mesmos não vejam o que muitos verão.

Valdecir Alves Nogueira.

RESUMO

O desenvolvimento e a reutilização de arquitetura de softwares voltados para educação no Brasil continuam carentes de pesquisas para reduzir o prazo, o custo e melhor aproveitamento das funcionalidades que obedeçam às características pedagógicas em ambiente real de aprendizagem. Esta dissertação explora os recursos da orientação a objetos para definir uma arquitetura de *framework* para TV Digital interativa e reutilizar as funcionalidades dos módulos de objetos digitais de aprendizagem. Com a necessidade de focalizar sobre um tipo específico de problema, identifica que alunos do ensino médio e superior ainda possuem dificuldades em aprender matemática, por isso é investigado o campo da trigonometria com propósito de promover entretenimento e educação sobre as formas de recursos didáticos utilizados pelos professores. Faz parte desta dissertação também investigar aplicações e *frameworks* desenvolvidos através do *middleware* de TV Digital no padrão europeu e brasileiro, destacando dificuldades como definição de padrões de interfaces, manipulações gráficas, recursos de áudio e vídeo. Para qualificação do sistema gerado são apresentados resultados avaliativos em três turmas utilizando os recursos de objetos de aprendizagem para aulas de trigonometria em um sistema de televisão digital. Os resultados obtidos na experiência demonstram que o ambiente contribui de forma positiva como mais um apoio de recursos tecnológicos que podem ser utilizados na educação e fomenta para necessidade de utilizar aplicações gratuitas para TV Digital interativa, deixando uma arquitetura que possa ser explorada para outro campo da matemática ou outra disciplina utilizando a mesma metodologia.

Palavras-chave: Objetos digitais de aprendizagem. *Middleware*. TV Digital interativa. Educação. *Framework*.

ABSTRACT

The development and reuse of software architecture oriented education in Brazil are still in need of research to reduce the time, cost and better utilization of the features that meet the pedagogical features in a real environment for learning. This dissertation explores the features of object orientation to define an architecture framework for interactive Digital TV and reuse the functionality of the modules digital learning objects. With the need to focus on a specific type of problem, which identifies students in high school and college still have difficulties in learning mathematics, therefore investigates the field of trigonometry in order to promote entertainment and education on ways of teaching resources used by teachers. It is part of this dissertation also investigate applications and middleware frameworks developed by the Digital TV standard in Europe and Brazil, highlighting difficulties such as definition of standard interfaces, handling graphics, audio and video resources. To qualify the system evaluation results generated are presented in three classes using the resources of learning objects for teaching trigonometry in a digital television system. The results obtained in the experiment show that the environment contributes positively as a further support of technological resources that can be used in education and promotes the need to use free applications for interactive digital TV, leaving an architecture that can be exploited for other field of or other mathematical discipline using the same methodology.

Key-words: Digital learning objects. Middleware. interactive Digital TV. Education. Framework.

LISTA DE FIGURAS

Figura 1 - Fluxograma com fases e interações.	24
Figura 2 - Cobertura da população entre 2008 a 2011.	28
Figura 3 - Sistema de TV Digital.	28
Figura 4 - Constituição de um <i>middleware</i>	29
Figura 5 - Arquitetura de TVD em camadas.	30
Figura 6 - Padrão DVB em camadas.	32
Figura 7 - Padrão ISDB em camadas.....	34
Figura 8 - Padrão ISDB-Tb em camadas.....	35
Figura 9 - Modelo de interação por canal de retorno.	36
Figura 10 - Arquitetura JavaTV.....	39
Figura 11 - Ciclo de vida de um <i>Xlet</i>	39
Figura 12 - Divisão de tela em TVD em três camadas no MHP.	41
Figura 13 - Arquitetura Ginga com padrão GEM 1.1.....	43
Figura 14 - Arquitetura Ginga- J com JavaDTV.....	43
Figura 15 - Convergências de tecnologias.....	45
Figura 16 - Visão de <i>T-Learning</i> que associa educação e entretenimento.	46
Figura 17 - <i>T- Learning</i> sobre o modelo de aplicação do tipo <i>Quiz</i>	48
Figura 18 - Módulo de classes do jogo de perguntas e respostas.....	58
Figura 19 - Interface de perguntas e respostas.	58
Figura 20 - Interface inicial da composição do jogo <i>Pac-Man</i>	59
Figura 21 - Tela principal do aplicativo Digital Math.....	60
Figura 22 - Tela de <i>menu</i> com áreas temáticas.....	60
Figura 23 - Estrutura de classes do MoonDo.	61
Figura 24 - Tela com componentes gráficos do MoonDo.....	62
Figura 25 - Pontos comuns entre aplicações de mesmo domínio.....	65
Figura 26 - Contextualização de desenvolvimento de aplicações para <i>framework</i>	66
Figura 27- Atividades realizadas no desenvolvimento do <i>framework</i>	68
Figura 28- Integrantes do desenvolvimento do <i>framework</i> e uso da aplicação.....	70
Figura 29- Caso de uso das funcionalidades dos usuários.....	72

Figura 30- Fluxograma com execução dos OA em uma aula de trigonometria.	76
Figura 31 - Diagrama de arquitetura classe do <i>framework</i>	77
Figura 32 - Caracterização da arquitetura do <i>framework</i>	80
Figura 33 - Pacotes do <i>framework</i>	84
Figura 34 - Interface principal do <i>framework</i>	85
Figura 35 - Interface principal de História da trigonometria.	86
Figura 36 - Interface final de História da trigonometria.	86
Figura 37 - Interface principal de Mapa conceitual.	87
Figura 38 - Mapa conceitual de Teorema de Pitágoras.	88
Figura 39 - Mapa conceitual de funções trigonométricas.	88
Figura 40 - Tela principal de canais de vídeos.	89
Figura 41 - Tela de execução do canal 01.	90
Figura 42 - Funções trigonométricas com o <i>software</i> Trigonometria 1.1.	91
Figura 43 - Círculo trigonométrico.	92
Figura 44 - Interface do nível <i>Expert</i>	93
Figura 45 - Técnica de animação limitada.	93
Figura 46 - Interface de perguntas um (01) do nível <i>Expert</i>	94
Figura 47 - Interface de respostas da perguntas um (01) do nível <i>Expert</i>	94
Figura 48 - Interface de pergunta dois (02) do nível <i>Expert</i>	95
Figura 49 - Interface de representação de erro do nível <i>Expert</i>	95
Figura 50 - Interface de pergunta três (03) do nível <i>Expert</i>	96
Figura 51 - Interface de pergunta quatro (04) do nível <i>Expert</i>	96
Figura 52 - Interface de erros e acertos do nível <i>Expert</i>	97
Figura 53 - Interface de total de acertos do jogo.	97
Figura 54 - Primeira interface de Desafios.	98
Figura 55 - Segunda interface de Desafios.	98
Figura 56 - Terceira interface de Desafios.	99
Figura 57 - Interface de respostas dos Desafios.	99
Figura 58 - Ambiente de máquina virtual VMware <i>Player</i>	102
Figura 59 - Interface de ambiente interativo.	102
Figura 60 - Interface de <i>menu</i> na plataforma Ginga-NCL.	103
Figura 61 - Interface de <i>menu</i> de vídeos na plataforma Ginga-NCL.	103

Figura 62 - Interface de execução de vídeos na plataforma Ginga-NCL.	104
Figura 63 - Interface de Círculo trigonométrico na plataforma Ginga-NCL.	104
Figura 64 - Estrutura física de comunicação dos equipamentos.	106
Figura 65 - Execução da aplicação via TVDi.....	107
Figura 66 - Interface de chamada do módulo de Mapa conceitual.....	108
Figura 67- A) Apresentação da Interface Principal; B) Acesso ao Mc geral.	108
Figura 68 - A) Interface do Mc de Teorema de Pitágoras; B) Interface do Mc de Funções Trigonômicas.....	109
Figura 69 - Gráfico sobre os conceitos aplicados em sala de aula.	111
Figura 70 - Gráfico sobre a fácil navegação e operacionalidade.....	111
Figura 71 - Gráfico sobre o uso de objetos de aprendizagem digitais pela ferramenta.....	112
Figura 72 - Gráfico sobre a aplicabilidade da ferramenta para o ensino médio e superior.	112
Figura 73 - Gráfico sobre as variedades de suporte de conteúdo da ferramenta.	113
Figura 74 - Gráfico sobre o uso da ferramenta em ambiente de TVD.	113
Figura 75 - Gráfico por turma sobre a prática de conceitos aplicados em sala de aula.....	114
Figura 76 - Gráfico sobre a fácil navegação e operacionalidade da ferramenta.....	115
Figura 77 - Gráfico por turma sobre o uso de OA pela ferramenta.....	115
Figura 78 - Gráfico por turma sobre aplicabilidade da ferramenta	116
Figura 79 - Gráfico por turma sobre as variedades de suporte de conteúdos.....	117
Figura 80 - Gráfico sobre uso da ferramenta em um ambiente de TVD.	117

LISTA DE QUADROS

Quadro 1 - Trabalhos relacionados.....	64
Quadro 2 - Descrição do caso de uso "Visualizar Mapas Conceituais".	72
Quadro 3 - Descrição do caso de uso “Visualizar Histórico”	73
Quadro 4 - Descrição do caso de uso “Manipular Vídeos de Trigonometria”.....	73
Quadro 5 - Descrição do caso de uso “Manipular Funções Trigonométricas e Gráficas”.	74
Quadro 6 - Descrição dos casos de uso “Visualizar Desafios”.	74
Quadro 7 - Descrição do caso de uso “Manipular Jogo de Perguntas e Respostas”.	75

LISTA DE SIGLAS

AAC	<i>Advanced Audio Coding</i>
API	<i>Application Programming Interfaces</i>
ARIB	<i>Association of Radio Industries and Businesses</i>
ATSC	<i>Advanced Television System Comitee</i>
AWT	<i>Abstract Window Toolkit</i>
BML	<i>Broadcast Markup Language</i>
CC	<i>Common-Core</i>
CRPG	<i>Computer- Role –Playing Games</i>
COFDM	<i>Coded Orthogonal Frequency Division Multiplexing</i>
DASE	<i>DTV Application Software Environment</i>
DiBEG	<i>Digital Broadcasting Experts Group</i>
DAVIC	<i>Digital Audio-Visual Council</i>
DVB	<i>Digital Vídeo Broadcasting</i>
DVB-C	<i>Digital Vídeo Broadcasting – Cable</i>
DVB-J	<i>Digital Vídeo Broadcasting – Java</i>
DVB-H	<i>Digital Vídeo Broadcasting – Handheld</i>
DVB-T	<i>Digital Vídeo Broadcasting –Terrestrial</i>
DVB-S	<i>Digital Vídeo Broadcasting – Satellite</i>
DVB-SI	<i>Digital Vídeo Broadcasting – Service Information</i>
GEM	<i>Globally Executable MHP</i>
GPL	<i>GNU Public Licence</i>
HAVI	<i>Home Audio Video Interoperability</i>
HDMI	<i>High-Definition Multimedia Interface</i>
HDTV	<i>High-Definition Television</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hyper Text Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
ISDB	<i>Integrated Services Digital Broadcasting</i>
ISDB-T	<i>Integrated Services Digital Broadcasting – Terrestrial</i>
ISDB-Tb	<i>Integrated Services Digital Broadcasting – Terrestrial Brazilian</i>

IEEE	<i>Institute of Electrical and Electronic Engineers</i>
IP	<i>Internet Protocol</i>
IPTV	<i>Internet Protocol-Television</i>
JCIC	<i>Joint Committee on InterSociety Coordination</i>
JMF	<i>Java Media Framework</i>
JVM	<i>Java Virtual Machine</i>
LTSC	<i>Learning Technology Standards Committee</i>
LWUIT	<i>Lightweight User Interface Toolkit</i>
LUA	<i>A Programing Language defined at www.lua.org</i>
MHP	<i>Multimedia Home Plataform</i>
MPEG	<i>Moving Pictures Expert Group</i>
NAB	<i>National Association of Broadcasters</i>
NCL	<i>Nexted Context Language</i>
NCTA	<i>National Cable Television Association</i>
ORB	<i>Object Request Broker</i>
QAM	<i>Quadrature Amplitude Modulation</i>
QPSK	<i>Quadrature Phase Shift Keying</i>
RAM	<i>Random Acesss Memory</i>
RTP	<i>Real-time Transport Protocol</i>
SBTVD	<i>Sistema Brasileiro de TV Digital</i>
SBTVD-T	<i>Sistema Brasileiro de TV Digital -Terrestre</i>
SMPTE	<i>Society of Motion Picture and Television Engineer</i>
STB	<i>Set-Top-Box</i>
TCP/IP	<i>Transport Control Protocol/Internet Protocol</i>
TVD	<i>TV Digital</i>
TVDi	<i>TV Digital interativa</i>
TS	<i>Transport Stream</i>
UI	<i>User Interface</i>
XML	<i>Extensible Markup Language</i>
WBT	<i>Web Based Training</i>

SUMÁRIO

1 INTRODUÇÃO	19
1.1 MOTIVAÇÃO/JUSTIFICATIVA.....	21
1.2 PROBLEMA.....	22
1.3 OBJETIVOS	22
1.3.1 Objetivo geral.....	23
1.3.2 Objetivos específicos	23
1.4 METODOLOGIA.....	23
1.5 ORGANIZAÇÃO DO TRABALHO	25
2 REFERENCIAL TEÓRICO	27
2.1 PERSPECTIVA DA TV DIGITAL NO BRASIL.....	27
2.2 SISTEMA DE TV DIGITAL	28
2.3 <i>MIDDLEWARE</i> DO RECEPTOR DE TV DIGITAL	29
2.4 PADRÕES PARA TV DIGITAL.....	30
2.4.1 Padrão DVB	31
2.4.2 Padrão ISDB.....	33
2.4.3 Padrão ISDB-Tb.....	34
2.5 INTERATIVIDADE E CANAL DE RETORNO EM TV DIGITAL.....	35
2.6 APIs PARA TV DIGITAL	37
2.6.1 API JavaTV	37
2.6.2 API HAVI	40
2.6.3 API DAVIC.....	41
2.6.4 API JMF	42
2.6.5 APIs definidas pelo DVB.....	42
2.6.6 APIs para o Ginga	42
2.7 <i>T-LEARNING</i>	45
2.7.1 Classificação de processos educativos em <i>T-Learning</i>	46
2.7.2 Definição e justificativa do <i>Quiz</i>	47
2.7.3 Vantagens e desvantagens de <i>T-Learning</i> para o Brasil	48
2.8 OBJETOS DE APRENDIZAGEM DIGITAIS	49
2.8.1 Características de objetos de aprendizagem.....	50

2.8.2 Aspectos pedagógicos para objetos de aprendizagem	51
2.9 <i>FRAMEWORK</i>	53
2.9.1 Classificação de <i>frameworks</i>	55
2.9.2 Vantagens e desvantagens de <i>frameworks</i>	56
2.10 TRABALHOS RELACIONADOS	57
2.11 CONCLUSÃO	64
3 MODELAGEM DO <i>FRAMEWORK O-LEARNING TVDi</i>	65
3.1 MODELAGEM CONCEITUAL DO <i>FRAMEWORK</i>	65
3.2 MODELO DE PROCESSO DE DESENVOLVIMENTO DO <i>FRAMEWORK</i>	67
3.3 ESPECIFICAÇÃO DE REQUISITOS	68
3.4 ANÁLISE DE COMPONENTES	69
3.5 MODIFICAÇÃO DE REQUISITOS	69
3.6 PROJETO DE SISTEMA COM REUSO	70
3.6.1 Integrantes do <i>Framework</i> e Uso da Aplicação	70
3.6.2 Caso de uso	71
3.6.3 Estudo de caso de aplicações criadas pelo <i>framework</i>	75
3.6.4 Diagrama de classes	76
3.6.5 Levantamento da arquitetura de execução do <i>framework</i>	79
3.7 DESENVOLVIMENTO E INTEGRAÇÃO	80
3.8 VALIDAÇÃO DE SISTEMAS	81
3.9 CONCLUSÃO	81
4 DESENVOLVIMENTO DO <i>FRAMEWORK O-LEARNING TVDi</i>	82
4.1 FERRAMENTAS DE DESENVOLVIMENTO	82
4.1.1 Sistema operacional	82
4.1.2 Plataforma de desenvolvimento	82
4.1.3 Emulador Exletview	83
4.2 ARQUITETURA MODULARIZADA DO <i>FRAMEWORK</i>	83
4.3 CENÁRIO PRINCIPAL DO <i>FRAMEWORK</i>	84
4.4 ESTUDO DE CASOS IMPLEMENTADOS	85
4.4.1 História da trigonometria	85
4.4.2 Mapa conceitual	87
4.4.3 Vídeos	89

4.4.4 Gráficos	90
4.4.5 Jogo de Quiz para trigonometria	92
4.4.6 Desafios	97
4.5 CONCLUSÃO	100
5 AVALIAÇÕES DO <i>FRAMEWORK O-LEARNING TVDi</i>	101
5.1 AMBIENTE GINGA	101
5.2 AMBIENTE REAL DE TVDi	105
5.2.1 Estrutura física de comunicação dos equipamentos	105
5.2.2 Execução do módulo do <i>framework</i> em ambiente de TVDi	106
5.3 AVALIAÇÕES EM AMBIENTE REAL DE APRENDIZAGEM	109
5.3.1 Avaliação de resultados	110
5.3.2 Avaliação dos recursos do <i>framework</i> pelas turmas	110
5.3.3 Avaliação dos recursos do <i>framework</i> por cada turma	113
5.3.4 Análise dos resultados das pesquisas	118
5.4 CONCLUSÃO	118
6 CONSIDERAÇÕES FINAIS	120
6.1 DIFICULDADES ENCONTRADAS	121
6.2 TRABALHOS FUTUROS	121
REFERÊNCIAS BIBLIOGRÁFICAS	123
ANEXO A – FORMULÁRIO DE PESQUISA DE OPINIÃO	131

1 INTRODUÇÃO

O processo de ensino/aprendizagem, seja a distância ou presencial, alcançou mudanças significativas no novo cenário mundial, graças à evolução de novas tecnologias da informação e da comunicação como a TV Digital interativa (TVDi) entre outras. Segundo Zancanaro, Santos e Todesco (2011) o surgimento da TVDi é um importante instrumento para apoiar o ensino e a aprendizagem disponível para o uso de educadores. Por meio dela é possível transmitir áudio, vídeo e dados, levando o conhecimento a um número maior de pessoas, promovendo assim a inclusão social. Ele afirma também que aprendizagem através da TVDi é conhecida como *T-Learning* e como na *web* necessita de um ambiente de aprendizagem onde os estudantes podem entre outras coisas: interagir com o conteúdo pedagógico e trocar informações com o professor.

Aliado a este cenário cresce ainda mais o uso de televisores digitais. De acordo com o Site Oficial da TV Digital Brasileira - DTV (2011), até 2016 o governo federal quer que o sistema de TV Digital seja implantado em todo o Brasil, período em que ocorrerão eventos esportivos da copa do mundo e das olimpíadas. Para Maia (2010), o objetivo da utilização da TVDi, além de democratizar o acesso à informação para a população nos próximos anos é possibilitar que um telespectador aprendiz possa interagir com a programação em tempo real, escolhendo ângulos de câmera, acessando aplicativos interativos, tendo uma comunicação bidirecional entre a emissora e o telespectador, que passa a ter uma participação efetiva no conteúdo.

Mesmo nesta perspectiva ainda há carência da utilização de recursos que ajudam a promover a aprendizagem. Destacam-se, por exemplo, o emprego de objetos digitais de aprendizagem conceituados pela *Learning Technology Standards Committee* (LTSC) do *Institute of Electrical and Electronics Engineers* (IEEE, 2005), como uma entidade digital ou não digital, que pode ser usada e reutilizada ou referenciada durante um processo de suporte tecnológico ao ensino e aprendizagem. Autores como Audino e Nascimento (2010) colaboram afirmando que quando se refere a objetos de aprendizagem (OA) na sua maioria, associa-se ao uso de computador e a utilização da Internet, pouco se fala para o uso desses objetos para fins didáticos em arquitetura de TV Digital no Brasil.

Apesar de existirem poucos trabalhos nesta linha de pesquisa no Brasil, Hashim et al. (2009) realizaram pesquisas sobre os benefícios dos OA. Eles comprovaram que os mesmos, quando bem explorados, têm o potencial de mudar significativamente as formas de trocar informação entre os discentes e docentes, potencializando a facilidade de

interação e dinamizando o processo didático em sala de aula. Isso se explica pelas possibilidades dos OA poderem servir como um repositório de acesso às informações para serem compartilhados em qualquer hora e lugar, além de permitir a reusabilidade em múltiplos contextos, como faculdades, escolas e ambiente virtuais de aprendizagem.

Os mesmos destacaram também que os OA ajudam a organizar melhor o conteúdo preparado para ser ministrado de acordo com a expertise do docente, pois as utilizações deles só fazem sentido, quando bem planejados para alcançar os resultados de aprendizagem esperados. Outro fator a ser observado por esses pesquisadores é o aumento da participação dos alunos em um processo de ensino e aprendizagem, através dos usos desses suportes tecnológicos que estimulam a auto-avaliação e a colaboração.

Entretanto, apesar dessas qualidades, existem dois requisitos importantes a considerar: as arquiteturas das aplicações que permitem o uso de objetos digitais e os recursos disponíveis via TVDi para atender as necessidades de um cenário de aprendizagem. Sobre essas exigências surge a importância de aplicações com características de *frameworks*. Existem na literatura várias definições sobre *frameworks*, contudo, a maioria dos autores argumentam que os *frameworks* permitem a reutilização através de implementações de aplicações e que, principalmente atendem ao domínio do problema.

Por exemplo, segundo Mattsson e Bosh (1997) o *framework* é como um conjunto de classes que cooperam entre si e fornecem um projeto reusável para uma categoria específica de aplicações. Desta forma, é possível usufruir dos benefícios da orientação a objetos, um paradigma bastante utilizado em diversas aplicações que precisam de manipulação de recursos gráficos e de interações com usuários como jogos digitais entre outras (DAMSOM, 2004).

Neste contexto, este trabalho apresenta como dissertação de mestrado o desenvolvimento de um *framework* de TVDi para manipular objetos digitais de aprendizagem relacionados ao ensino da trigonometria. Serão levadas em consideração as especificações do *middleware* MHP e Ginga, a fim de propor uma arquitetura que possa ser executada e adaptada em mais de um ambiente de TVD, bem como servir de fonte para trabalhos futuros.

1.1 MOTIVAÇÃO/JUSTIFICATIVA

As pesquisas realizadas sobre o desenvolvimento de aplicações para TVDi no Brasil continuam em fase de expansão. Entretanto, existem diversos trabalhos realizados em outros países com utilização do *middleware* MHP, onde a tecnologia já é adotada há mais tempo e o desenvolvimento de *software* através de APIs (*Application Programming Interfaces*) possibilitam a reusabilidade de arquitetura de *softwares* para TVDi em diversos padrões de manipulações de interface gráficas.

As limitações de desenvolvimento de aplicações para TVDi percebidas no cenário brasileiro são devidas a alguns recursos que estão em fase de testes e validação previstos para serem empregados no padrão de sistema ISDB-Tb - Serviço Integrado de Radiodifusão Digital Terrestre brasileiro, que utilizam *Set-Top-Boxes* (STBs). Entre esses recursos estão as funcionalidades de APIs, interatividade, multiprogramação e portabilidade, com exceção da alta definição (BREUNING, 2009). O que se espera com a implantação definitiva da TVD no Brasil é uma evolução mais rápida de todos esses benefícios, permitido uma melhor contribuição dos serviços oferecidos e o aumento da demanda de desenvolvimento de *software* para TVDi.

As produções de projetos e pesquisas para a construção de conteúdos digitais educacionais como instrumentos didáticos para instituições de ensino através da TV Digital ainda são limitadas. Isso se comprova pelas poucas orientações de usabilidade de objetos digitais e por poucos trabalhos realizados com a utilização dos mesmos (GOMES; LIMA; NEVADO, 2007). Outros requisitos fundamentais que continuam ainda carentes de mais iniciativas, segundo Kemczinski et al. (2011), são os de ponto de vista pedagógicos que permitem a reusabilidade, a organização de dados em diferentes ambientes e o fácil acesso de informação pelos os usuários, contribuindo para melhorar a metodologia utilizada em ambiente de *T-Learning*.

Justifica-se através deste contexto a utilidade desse trabalho, pois verifica-se a oportunidade de investigar e de promover uma arquitetura de *software* para fins de pesquisas na área de educação. Além disso, pretende-se disponibilizar uma infraestrutura de *software* no padrão de *middleware* Europeu MHP, visando poder ser adequado para outro padrão de *middleware* Ginga-J, a fim de incentivar a comunidade científica a despertar ainda mais para as necessidades de analisar as diferenças e as dificuldades de elaboração de projetos nesta área.

1.2 PROBLEMA

A produção de uma arquitetura de *software* através de OA para TVDi envolve várias etapas, como definição de um contexto específico, requisitos de interfaces gráficas padronizadas e implementação em uma linguagem de programação. Pesquisadores como Gama (2007) informam que o desafio é integrar a parte pedagógica e a parte ergonômica, e principalmente construir e avaliar uma ferramenta que poderá levar a um aprendizado do aluno ou usuário.

Tal desafio é imposto também a *designers* instrucionais, programadores e outros profissionais que desenvolvem estes produtos. Estes precisam tomar conhecimento de qual é o público alvo e como construir um produto que explore eficazmente os objetivos do OA, no que tange à qualidade da informação e que atenda requisitos para serem empregados ao ambiente proposto.

Outro aspecto a considerar é que apesar da TVD no Brasil ser uma oportunidade de promover serviços e desenvolvimentos de aplicação para setores da educação, indústria e comércio eletrônico, o fato é que existem entraves políticos, econômicos e tecnológicos que contribuem para uma transição lenta e fazem da TVD uma plataforma com recursos similares a da TV analógica. Surgem com isso dificuldades no desenvolvimento de aplicações através do *middleware* Ginga e seu ambiente de teste (MINHO et al. 2009).

Dadas as dificuldades para implementar um cenário de aprendizagem através de uma abordagem pedagógica visando atender uma área específica sobre ambiente de TVDi, é possível apontar uma preocupação predominante quanto à elaboração de uma estrutura de *software* que facilite a reusabilidade, interação e integração de suportes tecnológicos voltados para educação.

Portanto, o problema a ser tratado é como reduzir estas dificuldades associadas às aplicações de TVDi para auxiliar o desenvolvimento de uma arquitetura que possa manipular OA e melhorar as formas de comunicação dos discentes através dos conteúdos de aprendizagem.

1.3 OBJETIVOS

Os objetivos estão divididos em objetivo geral e específicos e são apresentados nas próximas subseções.

1.3.1 Objetivo geral

Desenvolver um *framework no modelo de middleware* MHP que permita criar interfaces gráficas de manipulação de objetos de aprendizagem digitais para TVDi no contexto do ensino da trigonometria.

1.3.2 Objetivos específicos

- a) Desenvolver um conjunto de classes orientadas a objetos que permitam a construção e organização de módulos de OA;
- b) Implementar funcionalidades para promover a navegação e tratamento de eventos no ambiente de TVDi;
- c) Desenvolver estudos de casos para verificar as funcionalidades dos módulos de OA para TVDi no emprego da matemática através da trigonometria; e
- d) Indicar limitações e novos benefícios a serem oferecidos pelo *framework* através da plataforma do MHP, levando em consideração requisitos para extensão da arquitetura para o padrão brasileiro Ginga.

1.4 METODOLOGIA

A metodologia usada para o desenvolvimento deste trabalho de pesquisa consistiu em cumprir etapas visando atingir cada um dos objetivos específicos descritos na seção 1.3.2. Na Figura 1 é apresentado o fluxograma com as fases e interações e em seguida as descrições de como ocorreram cada fase.

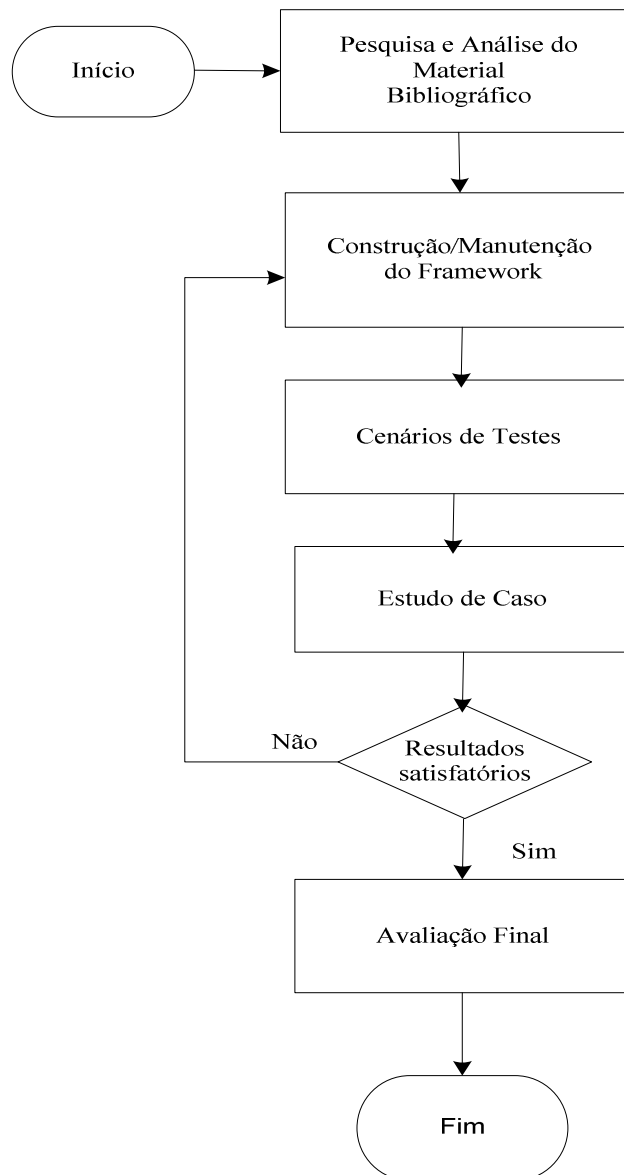


Figura 1 - Fluxograma com fases e interações.

A fase de pesquisa e análise do material bibliográfico foi realizada em diversas fontes de literatura científica para desenvolvimento deste trabalho. Nessa fase ocorreu um estudo detalhado sobre os seguintes tópicos: conceitos e características de OA, como explorá-los e as tecnologias e definições necessárias para o desenvolvimento de aplicações em ambiente de TVDi. Esta fase se estendeu para verificar os conceitos de *frameworks*, *middleware*, recursos didáticos e pedagógicos visando definir um cenário de aprendizagem. O objetivo foi selecionar quais tecnologias ofereciam os suportes necessários para desenvolvimento do *framework*.

A fase de construção/manutenção do *framework* e cenário de testes ocorreram no laboratório de TVD do Centro de Pesquisa e Desenvolvimento em Tecnologia Eletrônica

da Informação (CETELI) da Universidade Federal do Amazonas (UFAM). A proposta da fase de construção/manutenção do *framework* foi testar e avaliar, primeiramente, os recursos do *framework* para auxiliar a manipulação de OA via ambiente emulado, de máquina virtual e definir seus módulos de estudo de caso que serão construídos em sua arquitetura. Em seguida, na fase de cenário de teste foi avaliado a manipulação de OA no ambiente real de TVDi.

Na fase de estudo de caso foram realizados testes desses módulos para o apoio em ambiente de sala de aula em três (03) instituições de ensino em Manaus-AM. O objetivo foi analisar o ambiente de aprendizagem desenvolvido com apoio do *framework*, suas contribuições, suas limitações e ajustes necessários que ainda precisavam ser realizados baseados na organização e funcionalidades dos OA utilizadas durante uma aula de trigonometria. Ao identificar os ajustes, esses foram corrigidos na fase de construção/manutenção e as possíveis melhorias que ainda necessitassem ser realizadas foram realocadas para trabalhos futuros.

Ao término do projeto na fase da avaliação final ocorreram análises da dissertação de mestrado no contexto educacional para TVDi, indicando as contribuições e as dificuldades encontradas para produzir o *framework*. Nesta fase também foi possível definir trabalhos futuros para fonte de pesquisas através da estrutura disponível.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho é organizado em seis capítulos, onde no primeiro capítulo é apresentada a introdução do trabalho, contextualizando o cenário a qual o mesmo se insere, além da justificativa/motivação, problema a ser tratado, objetivos do trabalho e o detalhamento da metodologia.

O capítulo 2 descreve os conceitos necessários para elaboração deste trabalho. São detalhadas as fundamentações teóricas das tecnologias utilizadas para o ambiente de TVDi, características pedagógicas a serem adotadas e os trabalhos relacionados com o tema da dissertação.

O capítulo 3 apresenta a modelagem da construção da arquitetura. Nesta etapa são descritos os levantamentos de requisitos, análise, os principais diagramas e as ferramentas de desenvolvimento, levando em consideração os ambientes com estrutura de *frameworks* e as características de OA estudados no capítulo anterior.

O capítulo 4 descreve o *framework* desenvolvido, os estudos de casos implementados e suas funcionalidades.

No capítulo 5 são apresentados os testes realizados para validar a construção do *framework* e as avaliações das funcionalidades de seus módulos através de OA, detalhando os resultados obtidos.

E, por fim, no capítulo 6 são apresentadas as considerações finais, contendo as contribuições desta dissertação, as dificuldades encontradas e os trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os fundamentos necessários para o entendimento das tecnologias envolvidas em sistemas de TVD e *frameworks*. Com esta finalidade estão formalizados conceitos de tecnologias referentes a alguns padrões de TVD, incluindo as formas de interatividade, ambiente de suporte à aprendizagem “*T-Learning*” e objetos de aprendizagem digitais, relacionando suas características e as necessidades pedagógicas.

E, por fim, são apresentados os de conceitos de *frameworks*, suas classificações, vantagens, desvantagens e os trabalhos relacionados para indicar as semelhanças, as diferenças e novos recursos a serem disponibilizados pelo *framework* desta dissertação.

2.1 PERSPECTIVA DA TV DIGITAL NO BRASIL

Conforme o Site Oficial da TV Digital Brasileira – DTV (2011), a TV Digital estreou no Brasil em 2 de dezembro de 2007 em São Paulo e foi desenvolvida como uma variação inovadora do sistema japonês de TV Digital. A implementação do ISDB-Tb busca combinar os sólidos fundamentos estabelecidos na experiência japonesa com avanços tecnológicos posteriores. O que se pretende com a sua implantação de forma definitiva é a melhoria na transmissão de conteúdo para os telespectadores, permitindo ao mesmo tempo recepção móvel e portátil dos sinais de TVD.

As vantagens técnicas e o potencial de mercado do sistema no Brasil se evidenciam pela rapidez de sua adoção em escala nacional. Desde as primeiras transmissões, o sinal de TVD se expandiu por 43 áreas metropolitanas com estações geradoras de sinal digital em mais de 19 municípios contendo estações reforçadoras de sinal, chegando a mais de 82 milhões de pessoas. A cobertura do sinal de TV Digital cresce a ritmo acelerado atingindo, até março de 2011, 43% da população brasileira. A Figura 2 justifica a apresentação desses dados através da cobertura da população entre os anos de 2008 a 2011.



Figura 2 - Cobertura da população entre 2008 a 2011.

Fonte: Adaptado do Site Oficial da TV Digital Brasileira (DTV, 2011).

2.2 SISTEMA DE TV DIGITAL

De acordo com Becker e Montez (2005), um sistema de TV Digital pode ser decomposto em três partes principais: (i) um difusor, responsável por prover o conteúdo a ser transmitido e dar suporte às interações dos telespectadores; (ii) um receptor, que recebe o conteúdo e oferece a possibilidade do telespectador reagir ou interagir com o difusor; e (iii) um meio de difusão, que habilita a comunicação entre o difusor e o receptor. A Figura 3 apresenta como acontecem as interações entre as partes do sistema.

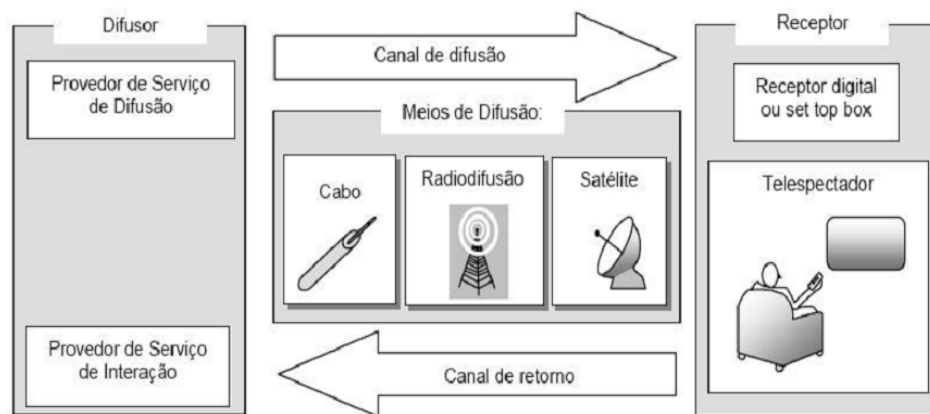


Figura 3 - Sistema de TV Digital.

Fonte: Adaptado de Becker e Montez (2005).

O provedor de serviço de difusão é o que envia o sinal digital que pode ser um vídeo, áudio e dados, enquanto que o provedor de serviço de interação é responsável pela coleta e manipulação da interatividade oriunda do canal de retorno. O meio de difusão é

responsável por possibilitar formas de transporte do sinal digital, até ser decodificado pelo *Set-Top-Box* conhecido atualmente como receptor digital. Os meios de difusão mais utilizados são via Cabo, Radiodifusão e Satélite. Ainda segundo Becker e Montez (2005) quando os dados retornam para o difusor o provedor de serviço de interação processa estas informações e envia uma resposta novamente, através do canal de difusão até o receptor que fez a interação. A repetição desse processo é chamada de “Carrossel de dados”.

A definição de *receptor digital* utilizada neste trabalho é a empregada por Vêras et al. (2009) que considera o receptor digital é um equipamento que se conecta ao televisor e uma fonte externa de sinal, e transforma o sinal em um conteúdo no formato ao qual possa ser apresentado em uma tela.

2.3 MIDDLEWARE DO RECEPTOR DE TV DIGITAL

Segundo Emmerich, Aoyama e Seventek (2007 apud COSTA, 2008), o *middleware* é uma camada de *software* existente entre as aplicações e o sistema operacional, que proporciona aos programadores a abstração do funcionamento do *hardware*, permitindo a realização de operações independente da solução do fabricante, o que facilita consideravelmente o desenvolvimento de sistemas. Eles afirmam que o *middleware* pode ser empregado em diversos dispositivos de tecnologia inclusive como parte integrante de um receptor.

Esta camada de *software* é constituída de várias bibliotecas de funções, apresentadas na Figura 4. As bibliotecas por sua vez possuem interfaces, ou *Application Programming Interfaces* (API) bem definidas que, ao serem implementadas pelos fabricantes de *hardware*, estendem ao *middleware* suas funcionalidades.

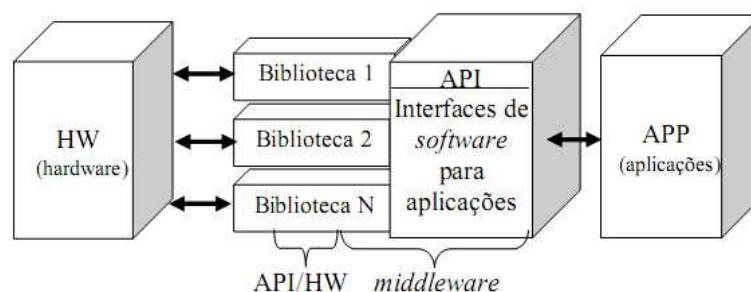


Figura 4 - Constituição de um *middleware*.
Fonte: Adaptado de Costa (2008).

2.4 PADRÕES PARA TV DIGITAL

A arquitetura de um sistema de TVD é integrada pelas camadas de Modulação (Transmissão), Transporte, Compressão, *Middleware* e Aplicações. Cada uma contém uma função bem definida que utiliza recursos da camada inferior e oferece suporte a camada superior (MARQUES, 2008). A Figura 5 exibe uma arquitetura de um sistema TVD com divisão em camadas.

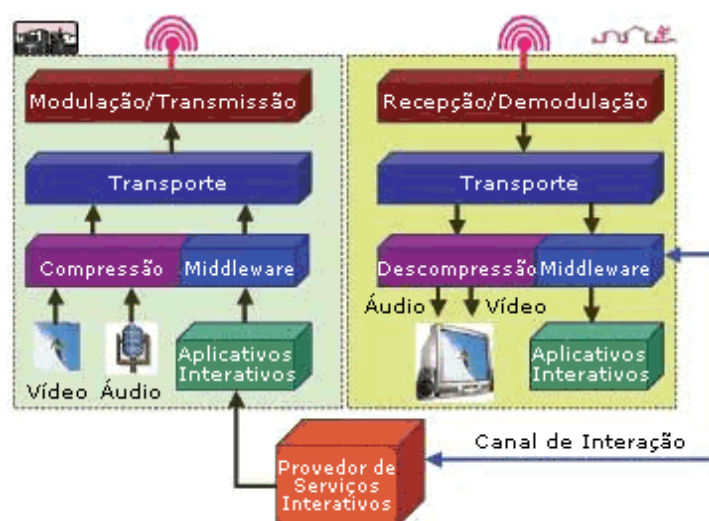


Figura 5 - Arquitetura de TVD em camadas.
 Fonte: Adaptado de Oliveira (2005).

A camada de transmissão é responsável pela transmissão e recepção do sinal entre o difusor e receptor, adicionando serviço de modulação, demodulação, codificação e decodificação. Esse serviços são esclarecidos de acordo com Andreato (2006) que explica que o sinal a ser transmitido na saída do multiplexador é formado por 0's e 1's, é modulado, isto é, o sinal é acoplado a uma portadora, onde são acrescentados estados para facilitar a detecção de erros e aumentar a taxa de transmissão. A demodulação corresponde à reconstituição do sinal que foi modulado.

O referido autor também define que um sinal de vídeo em uma transmissão de TVD é codificado na maioria dos casos no formato padrão MPEG (*Moving Pictures Expert Group*). O processo de recuperação do sinal através da utilização de um código de recuperação para garantir a qualidade de imagem e som ao telespectador é chamado de decodificação.

A camada de transporte, no lado do provedor de serviços (emissora), é encarregada de efetuar a multiplexação de vários programas em um único fluxo de transporte. No lado do receptor, ela faz a demultiplexação do fluxo de transporte de acordo com o programa selecionado pelo usuário.

A camada de compressão é responsável pela compressão e descompressão dos sinais de áudio e vídeo, no lado da emissora e no lado do usuário. A compressão corresponde à técnica de diminuição do tamanho de um vídeo formado por várias imagens projetadas. A título de exemplo, a técnica funciona da seguinte forma: quando um vídeo que tem uma pessoa falando e que esta esteja parada, no primeiro momento a imagem é projetada completa e no segundo momento fragmentos das imagens que são idênticos à imagem anterior são removidos. A descompressão corresponde à reconstituição do vídeo comprimido evitando o máximo de perdas em relação ao original.

A camada de *middleware* é responsável por padronizar o serviço oferecido para a camada de aplicação, escondendo as peculiaridades das camadas de compressão, transporte e transmissão.

A camada de aplicações é a responsável pela execução dos aplicativos.

Os padrões de TVD mais conhecidos que estão relacionados a este trabalho são: DVB (Europa), ISDB (Japão) e o ISDB-Tb (Brasil). Esses padrões foram criados baseados na experiência e necessidades específicas de uma região, como tomadas de eletricidade, voltagem, frequência, experiência com sistemas analógicos, sistemas de banda larga, codificadores, *set-top-boxes*, *middlewares*, emissoras, entre outros (GOULART, 2009).

Nas seções seguintes são apresentados de forma detalhada esses padrões, abordando suas características, *middlewares*, funcionalidades e tecnologias envolvidas.

2.4.1 Padrão DVB

Conforme DVB (2009), o padrão europeu DVB (*Digital Video Broadcasting*) foi fundado em setembro de 1993 e faz parte de um consórcio liderado pelo mercado de organizações do setor público e privado da indústria de televisão. Ele compreende mais de 200 organizações em mais de 25 países ao redor do mundo, e atualmente, existem várias formas de transmissão de dados e as mais conhecidas são o DVB-T (*Terrestrial*), DVB-S (*Satellite*), DVB-C (*Cable*) e o DVB-H (*Handheld*) para sistemas móveis. A transmissão terrestre por rádiofusão utiliza a modulação COFDM (*Coded Orthogonal Frequency*

Division Multiplexing), com taxas de transmissão que variam de 5 a 31,7 Mbit/s, dependendo dos parâmetros utilizados na codificação e modulação do sinal.

Na transmissão via satélite, esse sistema utiliza a modulação QPSK (*Quadrature Phase Shift Keying*). Na transmissão via cabo é recomendada a modulação 64-QAM (*Quadrature Amplitude Modulation*). Para transmissão via microondas utilizando frequências abaixo de 10 GHz é usada a modulação 16, 32 ou 64-QAM e para transmissão via microondas com frequências acima de 10 GHz utiliza a modulação QPSK.

O sinal de vídeo e a multiplexação são baseados na compressão no formato MPEG-2 vídeo, e a compressão de áudio é feita com MPEG-2 BC. Há um sistema de acesso condicional para controlar o acesso a material proprietário e, na camada de transporte, há uma tabela chamada de DVB-SI (*Service Information*) que conecta os vários fluxos elementares em programas e fornece descritores para a construção de guias eletrônicos de programação. O formato para o transporte de dados adotado é o MPEG2 TS.

O *middleware* utilizado pelo DVB é o MHP (*Multimedia Home Platform*). Este *middleware* busca oferecer um ambiente de TVD, independente de *hardware* e *software* específicos, aberto e inter-operável, para receptores de TVD. Seu ambiente de execução é baseado no uso de uma máquina virtual Java (JVM) e em um conjunto de APIs que possibilitam aos programas escritos em Java, o acesso a recursos e facilidades do receptor digital de forma padronizada.

Uma aplicação DVB usando API Java é chamada de aplicação DVB-J. Além do uso da API Java, o MHP introduziu também a possibilidade de usar uma linguagem de programação semelhante ao HTML (*HyperText Markup Language*), denominada DVB-HTML. A Figura 6 exibe a arquitetura do padrão DVB em Camadas.

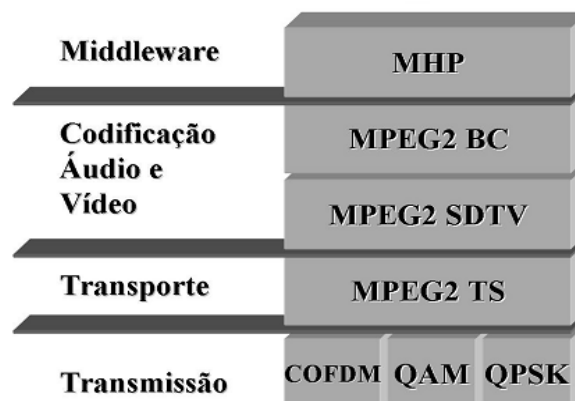


Figura 6 - Padrão DVB em camadas.
Fonte: Adaptado de Teleco (2010).

Este padrão é usado para o desenvolvimento da arquitetura do *framework* desta dissertação como base de testes em ambiente real de TVD. A escolha deste padrão é pela melhor compreensão das APIs exigidas durante a atividade de implementação.

2.4.2 Padrão ISDB

Segundo Lemos, Fernandes e Silveira (2004), o padrão Japonês ISDB (*Integrated Services Digital Broadcasting*) foi definido em 1999 no Japão pelo grupo DiBEG (*Digital Broadcasting Experts Group*) compondo várias empresas e operadoras de televisão. O objetivo do grupo DiBEG era promover e especificar o sistema de difusão terrestre de televisão digital japonês. Este padrão foi adotado apenas no Japão, porém é amplamente divulgado e reúne recursos como: alta definição – HDTV (*High Definition Television*), transmissão de dados e recepção móvel e portátil.

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

Na camada de codificação, o sinal de áudio é codificado usando a recomendação MPEG-2 AAC (*Advanced Audio Coding*) e o sinal de vídeo é codificado usando a recomendação MPEG-2 Vídeo com qualidade HDTV.

Na camada de transporte o padrão ISDB multiplexa e desmultiplexa os fluxos elementares de áudio, vídeo e dados (aplicações) usando a recomendação MPEG-2 TS. Na camada de *middleware* o padrão ISDB utiliza o ARIB (*Association of Radio Industries and Businesses*). Ele é formado por alguns padrões, como o ARIB STD-B24 - *Data Coding and Transmission Specification for Digital Broadcasting*, que define uma linguagem declarativa (BML – *Broadcast Markup Language*) baseada em XML (*Extensible Markup Language*), e usada para especificação de serviços multimídia para TVD. Outra especificação do *middleware* é o ARIB-STD-B23 baseada no padrão europeu (MHP), que permite a execução de aplicações interativas baseadas em Java. A Figura 7 exibe o padrão ISDB através da estrutura em camadas de TVD.

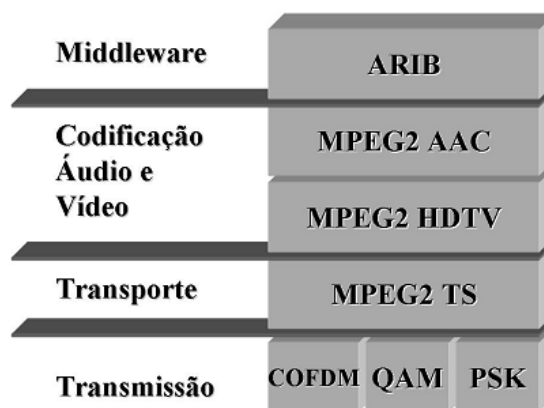


Figura 7 - Padrão ISDB em camadas.
Fonte: Adaptado de Teleco (2010).

2.4.3 Padrão ISDB-Tb

Segundo o DTV (2012), o Sistema Brasileiro de TV Digital Terrestre atualmente conhecido como ISDB-Tb foi desenvolvido com base no sistema japonês *Integrated Services Digital Broadcasting Terrestrial* (ISDB-T). Este sistema oferece uma série de diferenciais em relação aos sistemas de TVD em funcionamento no mundo. Esses diferenciais estão justamente no “casamento” entre a base técnica de transmissão do sistema japonês com os padrões de compressão digital de áudio e vídeo introduzidos pelo Brasil.

Na versão brasileira foram acrescentadas tecnologias desenvolvidas pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) e pela Universidade Federal da Paraíba (UFPB). O ISDB-Tb possibilita a transmissão de conteúdo de altíssima qualidade, tanto em termos de imagem e de som, permitindo ao mesmo tempo a recepção móvel e portátil dos sinais de TVD. A camada de aplicação é por onde acontecem as entradas de informações de áudios, vídeos e serviços interativos. Para oferecer esses recursos, foi adotado o padrão MPEG-4, também conhecido como H.264, para codificação de vídeo e o HE-AAC v2 para o áudio. Na camada de transporte o padrão utilizado é o MPEG-2 para transmissão terrestre de TV e Protocolo RTP (*Real-time Transport Protocol*) para Internet (IPTV). Na camada de transmissão é utilizada a modulação COFDM.

O *middleware* utilizado é o Ginga que tem a perspectiva de oferecer suporte a recursos de interatividade, possibilitando o desenvolvimento de aplicações que permitirão vários serviços. Ele representará a interface com o mundo da Internet, servindo de suporte para a interação entre o IP (*Internet Protocol*) com *broadcast* que é a transmissão da TV

aberta. O sistema formado por este *middleware* é dividido em três subsistemas principais interligados: o Ginga-CC, Ginga-NCL e o Ginga-J, que permitem o desenvolvimento de aplicações seguindo paradigmas de programação diferentes. Dependendo das funcionalidades requeridas no projeto de cada aplicação, um paradigma será mais adequado do que o outro.

O Ginga-CC (*Ginga Common-Core*) oferece o suporte básico para os ambientes Ginga-NCL e Ginga-J, de maneira que suas principais funções sejam para tratar da exibição de vários objetos de mídia, como JPEG, MPEG-4, MP3, GIF, entre outros formatos. O Ginga-CC fornece também o controle do plano gráfico para o modelo especificado para o ISDB-Tb e controla o acesso ao canal de retorno, módulo responsável por controlar o acesso à camada de rede.

Segundo Carvalho (2011), o Ginga-NCL (*Nested Context Language*) é voltado para aplicações declarativas e o Ginga-J voltado para aplicações procedurais Java que serão utilizados de acordo com as funcionalidades requeridas de cada aplicação. Através desta especificação é possível utiliza-se da linguagem Lua como *script* que é uma linguagem desenvolvida no Brasil por uma equipe da Pontifícia Universidade Católica do Rio de Janeiro (PUC-RIO). A Figura 8 apresenta a arquitetura do ISDB-Tb em camadas.

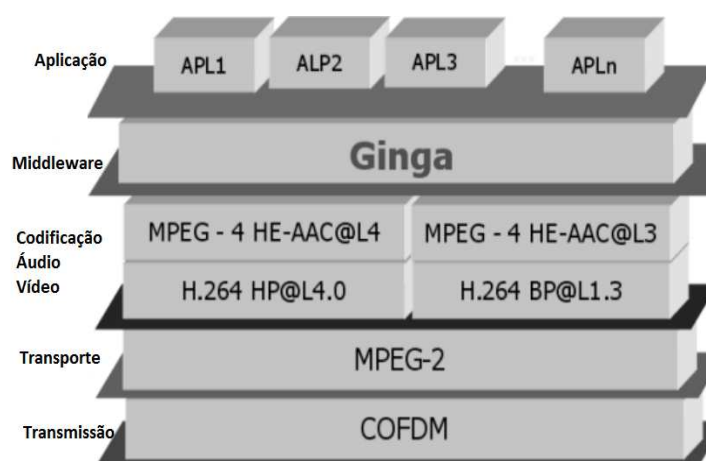


Figura 8 - Padrão ISDB-Tb em camadas.
Fonte: Adaptado de Lucas (2009).

2.5 INTERATIVIDADE E CANAL DE RETORNO EM TV DIGITAL

A TVD tem como uma das principais diferenças da TV analógica permitir que aplicações similares a que são executadas em computadores pessoais sejam transmitidas

em conjunto com o áudio e vídeo. As emissoras e agências de publicidade podem fazer programas que sejam mais atraentes ao telespectador, permitindo que ele possa interagir com aplicações associadas a um determinado comercial ou programa de TV, fazendo com que o usuário de TVD não seja apenas um elemento passivo, mas sim um usuário ativo com a programação, interagindo através da TV na hora que lhe convier (HEMMLEPP, 2006).

O canal de retorno (interação) se refere a um meio que o receptor possa se comunicar até um servidor de geração de conteúdo, podendo ser um modem ou uma placa de rede ethernet, rede TCP/IP ou uma rede sem fio. A partir dessa estrutura torna possível ao usuário o uso de aplicações como: comércio eletrônico (*t-commerce*), acesso bancário (*t-banking*), jogos multi-usuário (*Game Console*) entre outros.

De acordo com Monteiro et al. (2008), a forma de compreender a interação através do canal de retorno é que o fluxo recebido pelo usuário é composto por três subfluxos: áudio, vídeo e dados. Estes subfluxos são multiplexados pelo servidor de geração de conteúdo, que então monta o fluxo principal que é enviado ao receptor, a partir desse momento os dados retornam através do canal de retorno até o servidor de aplicação, que repassa os mesmos ao servidor de geração de conteúdo, conforme a Figura 9.

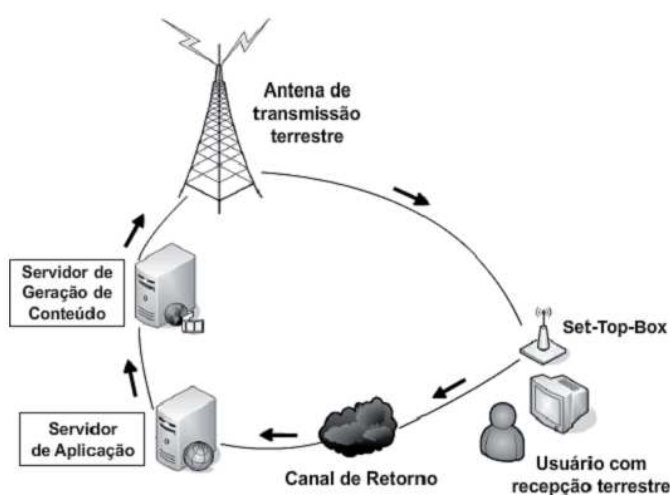


Figura 9 - Modelo de interação por canal de retorno.
Fonte: Adaptado de Monteiro et al. (2008).

Outro requisito a considerar é a classificação de interação que acontece em um determinado sistema de TVD através do uso do canal de retorno ou não. Segundo Perozzo (2011), a interatividade que acontece, por exemplo, no ISDB-Tb é classificada de duas formas: a interatividade parcial e a interatividade plena.

Este autor afirma ainda que a interatividade parcial é realizada quando um receptor de TVD possui um *middleware* que executa aplicações interativas, e por algum motivo não utiliza uma rede de comunicação para troca de dados, seja pelo fato da aplicação não utilizar um canal de retorno ou receptor não ter esta funcionalidade disponível, por exemplo: a emissora de TV transmite o aplicativo juntamente com o sinal de áudio e vídeo, e o telespectador apenas interage com a aplicação localmente, utilizando os recursos que já estão na aplicação.

Já a interatividade plena acontece quando o receptor possui um *middleware*, executa a aplicação e troca de dados com outro equipamento, em uma rede local ou Internet, por exemplo, servidores de aplicação utilizados para receber votação de uma enquete esportiva. Neste cenário a emissora de TV transmite a aplicação pelo ar e o telespectador retorna seu voto pela Internet, caracterizando o uso de um canal de retorno.

A forma de interatividade utilizada neste trabalho é baseada na definição deste autor. Neste contexto foi utilizada a interatividade plena, pois para executar os módulos de aplicações do *framework* através do receptor digital com *middleware* MHP, foi utilizada uma rede TCP/IP para acessar estes módulos localizados em um servidor de aplicação, havendo assim a troca de dados pelo canal de retorno.

2.6 APIs PARA TV DIGITAL

Nas próximas subseções serão apresentadas as APIs para o desenvolvimento de aplicações para TVD. O objetivo foi verificar os recursos possíveis de funcionalidades destas APIs que podem ser empregadas no desenvolvimento de um *framework* para TVD, observando os padrões de TVD brasileiro e europeu.

2.6.1 API JavaTV

A API JavaTV é uma extensão da linguagem Java que tem por objetivo possibilitar o desenvolvimento de aplicações em ambiente de TVD através de STBs, obedecendo as normas existentes em padrões de sistemas para TVD (ORACLE, 2012). Primeiramente o DVB baseou seu *middleware* (MHP) para TVD Européia nessa API. Isso foi expandido no GEM (*Globally Executable MHP*) para outros padrões mundiais americano (DASE), japonês (ARIBE) e brasileiro (Ginga). Esta API consiste de várias bibliotecas de código reutilizáveis para TVD dispostas da seguinte forma:

- `javax.tv.graphics` : Oferece mecanismo para os *Xlets* poderem encontrar o *container* raiz para seus componentes gráficos e disponibilizar mecanismo de controle sobre níveis de transparência;
- `javax.tv.locator` : Fornece um meio para referenciar dados e recursos acessíveis pela API JavaTV;
- `javax.tv.media` : Disponibiliza controles e eventos para o gerenciamento em tempo real de mídias em ambiente de TV;
- `javax.tv.net` : Fornece acesso a datagramas IP na *stream* de transporte;
- `javax.tv.service` : Disponibiliza mecanismo para acesso à informação do serviço (SI) e classes que representam seus elementos;
- `javax.tv.service.guide` : Oferece suporte para guias eletrônicos de programação (EPG);
- `javax.tv.service.navigation` : Fornece suporte para navegação entre serviços e informações hierárquicas de serviços;
- `javax.tv.service.selection` : Disponibiliza mecanismo para selecionar um serviço para apresentação;
- `javax.tv.service.transport` : Oferece informações adicionais sobre os mecanismos de transporte que transmitem os serviços
- `javax.tv.util` : Fornece suporte à criação de eventos e gerenciamento de eventos no tempo; e
- `javax.tv.xlet` : Disponibiliza a interface para os aplicativos e o gerenciador de aplicações se comunicarem.

As principais funcionalidades da API JavaTV de acordo com suas bibliotecas para Sedrez (2008) são:

- Fluxo de áudio e vídeo: Além do fluxo de vídeo e áudio vindos da emissora, é possível gerar na aplicação outros fluxos;
- Acesso a dados no canal de transmissão: no JavaTV pode-se receber dados para as aplicações;
- Interatividade com aplicações: Os aplicativos que usam esta API podem processar dados e retorná-los através de um canal de retorno; e
- Gerenciamento do ciclo de vida das aplicações: permite que aplicações coexistam com o conteúdo convencional de TV e possibilitando a troca de canal sem que a aplicação deixe de existir.

Um dos recursos mais explorados pelo gerenciamento do ciclo de vida das aplicações é o controle do *Xlet*. De acordo com Lemos, Fernandes e Silveira (2004), uma aplicação JavaTV é denominada *Xlet* e não precisa estar previamente armazenada no STB, pois pode ser enviada pelo canal de difusão quando necessária. Ou seja, o modelo *Xlet* é baseado na transferência de código executável pelo canal de difusão para o STB com posterior carga e execução do mesmo sendo realizada de forma automática ou manual.

Para controlar esse processo nos *Xlets*, cada STB possui um gerente de aplicações instalado. A Figura 10 ilustra a API JavaTV em camadas. A primeira camada destacada é do *Xlet*, e do gerente de aplicações. Logo abaixo da camada da API JavaTV está a camada de máquina virtual Java (JVM). Em seguida a camada de sistema operacional que através de seus *drivers* acessa os dispositivos dos receptores (camada de STB).

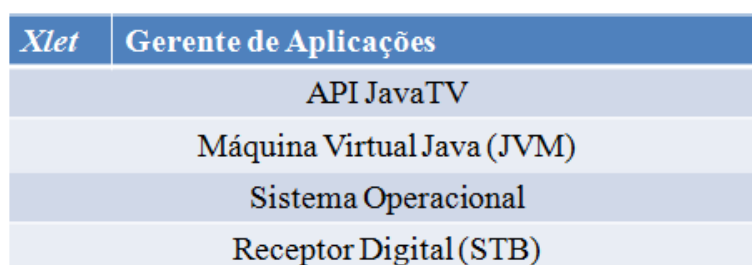


Figura 10 - Arquitetura JavaTV.
Fonte: Adaptado de Lemos, Fernandes e Silveira (2004).

Um *Xlet* em um ambiente de TVD equivale a um *applet* Java em um *browser* para PC. O seu ciclo de vida é composto por 4 estados: *loaded*, *paused*, *active* e *destroyed*. Todo *Xlet* deve implementar a interface `javax.tv.xlet.Xlet`, cujos métodos são ativados para sinalizar mudanças de estado da aplicação. A Figura 11 mostra o ciclo de vida de um *Xlet*, identificando os estados e os métodos suportados por sua interface.

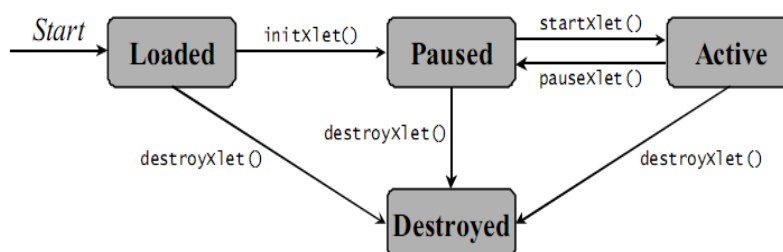


Figura 11 - Ciclo de vida de um *Xlet*.
Fonte: Adaptado de Lemos, Fernandes e Silveira (2004).

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

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

2.6.2 API HAVI

Conforme Guilherme e Gomarim (2009), a HAVI (*Home Audio Video Interoperability*) é uma organização criada em 1998 por um grupo de fabricantes representados pelas empresas *Grudig AG*, *Hitachi*, *Philips*, *Thomson*, *Panasonic*, *Sharp*, *Sony* e *Toshiba*, com intuito de oferecer uma arquitetura de rede de modo que todos os dispositivos de áudio e vídeo que compõe a rede possam interagir entre si.

A API HAVI foi originalmente criada para fornecer um padrão de interoperabilidade entre dispositivos por disponibilizar funcionalidades específicas para TV, existentes em componentes de botões, campos de textos e dentre outros, que facilitam o desenvolvimento das interfaces interativas das aplicações (NEVES, 2010).

Parte desta API é herdada do pacote Java AWT (*Abstract Window Toolkit*). A AWT vem com interfaces voltadas mais para computadores com uso de programação para *mouse* e texto. Por outro lado, a API HAVI provê UI (*User Interface*) baseado em imagens, reusa os aspectos da AWT que não são orientados a computadores e também tem acesso ao sistema gráfico e resolução do vídeo. Estes recursos são organizados por componentes de alto nível *container* e por componentes necessários para o gerenciamento de objetos da UI.

Neste contexto é possível compreender como é a divisão de uma tela TVD em três camadas através do *middleware* MHP apresentada na Figura 12. A camada de Fundo

mostra uma imagem ou cor fixa; a camada de Vídeo é a área em que o vídeo é executado; a camada de Gráfico mostra a interface da aplicação onde são adicionados todos os recursos de UI.



Figura 12 - Divisão de tela em TVD em três camadas no MHP.
Fonte: Adaptado de Santos (2007).

A API HAVI possui ainda uma classe chamada *HScreen* que é responsável pela configuração de tela. Esta classe possui três subclasses (*HVideoDevice*, *HGraphicsDevice* e *HBackgroundDevice*) que são capazes de configurar cada camada da TVD separadamente. As aplicações *Xlets* compartilham a tela de uma TVD pela classe *HScene*, cada *Xlet* pode possuir apenas um *HScene*, que substitui a classe *Frame* de AWT. A *HScene* é responsável pelo controle de foco para o controle remoto, sendo que é no objeto do tipo *HScene* que são adicionados os objetos e *containers* responsáveis pela UI, e controle quando a aplicação fica visível.

2.6.3 API DAVIC

Brackmann (2008) e Maia (2010) afirmaram que API DAVIC (*Digital Audio-Visual Council*) foi criada pela associação DAVIC. Esta API especifica formatos de conteúdo para objetos como áudio, vídeo, textos e hipertexto e ainda controla o acesso ao aplicativo e a língua adotada (áudio e legenda).

Para Guilherme e Gomarim (2009), as implementações com o uso da API DAVIC aparecem mais freqüentes na Europa, onde trabalha junto com o padrão DVB, e normalmente é necessária a integração com outras APIS como a HAVI. Algumas funcionalidades desta API são:

- Define que funcionalidade um sistema digital de áudio e vídeo devem fornecer;
- Arquiteturas de referência do provedor, da rede de distribuição e também do sistema presente na casa do usuário;
- Arquitetura de gerenciamento;
- Protocolos de camadas baixas e definição de interfaces físicas;
- Representação de informação;
- Ferramentas básicas de segurança;
- Ferramentas para verificação de conformidade e interoperabilidade;
- Arquitetura para armazenamento de conteúdo; e
- Arquitetura que disponibilize interatividade ao usuário final.

2.6.4 API JMF

Em Santos (2007), a API JavaTV utiliza a API JMF (*Java Media Framework*) para o gerenciamento da transmissão de mídia como áudio e vídeo, fornecendo para a JavaTV um *framework* para exibição de mídia dependente de tempo *time-based media*, e independente do mecanismo de execução e do protocolo de transporte, além do conteúdo de mídia.

2.6.5 APIs definidas pelo DVB

Segundo DVB (2012) são APIs que fazem parte do pacote DVB, necessárias para segurança, acesso a dados e dispositivos de entrada e saída (I/O). Elas foram definidas como padrão de utilização do GEM.

2.6.6 APIs para o Ginga

A utilização de partes dessas APIs citadas nos itens anteriormente não são livres e foram adotadas de formas definitiva apenas no padrão GEM conforme a Figura 13 apresentada no padrão brasileiro Ginga como uma proposta inicial. Para que as mesmas fossem utilizadas no padrão brasileiro de *middleware* Ginga era necessário a cobrança de *royalties* (SARAIVA JR., 2010).

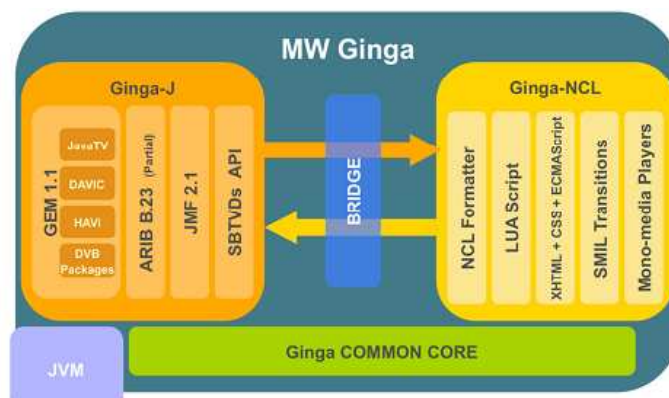


Figura 13 - Arquitetura Ginga com padrão GEM 1.1.

Fonte: Adaptado de Saraiva Jr. (2010).

Diante deste cenário o ISDB-Tb e a *Sun Microsystems* trabalharam em conjunto para especificar APIs equivalentes ao GEM livres de *royalties* e logo foi aprovada uma proposta de desenvolvimento da arquitetura GINGA apresentada na Figura 14. Nesta arquitetura é substituído parte das APIs incorporadas no GEM tais como DAVIC e HAVI e são inseridas APIs referentes ao modelo JavaDTV.

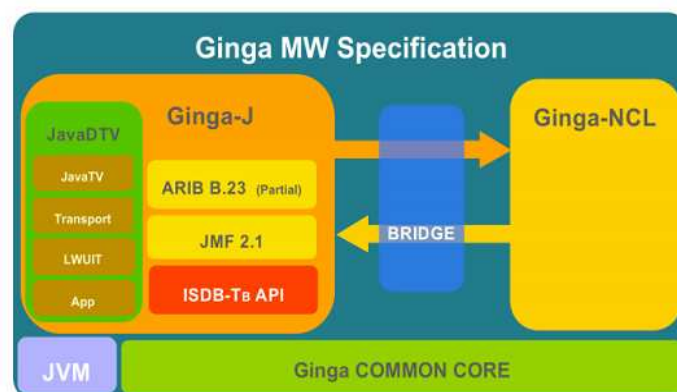


Figura 14 - Arquitetura Ginga- J com JavaDTV.

Fonte: Adaptado de Saraiva Jr. (2010).

Neste novo formato de arquitetura o Ginga-J é composto por um conjunto de APIs definidas para atender todas as funcionalidades necessárias para a implementação de aplicativos para TVD, desde a manipulação de dados multimídia até protocolos de acesso à Internet. Sua especificação é formada por uma adaptação da API de acesso à informação de serviço do padrão japonês (ISDB ARIB-B23), pela especificação JavaDTV (que inclui a API JavaTV), além de um conjunto de APIs adicionais de extensão ou inovação.

Estas APIs incluem o conjunto de classes disponíveis para a ponte entre os aplicativos escritos nas linguagens NCL e Java, funcionalidades adicionais para sintonia de

canais, envio de mensagens assíncronas, acesso pelo canal de interatividade e um conjunto de comandos para a integração de dispositivos externos ao *middleware*, viabilizando o suporte a recursos multimídia e interação simultânea de múltiplos usuários em aplicações de TVDi.

Entre as principais diferenças do JavaDTV em relação ao GEM, no que se refere ao desenvolvimento de aplicações, pode-se citar a API LWUIT (*Lightweight User Interface Toolkit*), responsável por definir elementos gráficos para TVD, gerenciadores de *layout* e eventos do usuário.

Visando a melhor compreensão das funcionalidades das APIs incluídas no GINGA, a norma ABNT NBR 15606-4 (2010) definiu uma plataforma para execução de aplicações Java para TV Digital através do GINGA-J, disponibilizando uma série de APIs que são identificadas conforme a seguir:

- **Pacotes da plataforma JavaME (*Java Micro Edition*)** : contem funcionalidades da plataforma JavaME providas pela configuração CDC 1.1 (*Connected Device Configuration*) e os perfis FP 1.1 (*Foundation Profile*) e PBP 1.1 (*Personal Basis Profile*);
- **Pacotes da especificação JSSE (*Java Secure Socket Extension*) 1.0.1** : oferece funcionalidades opcionais de segurança para a plataforma básica de Java para TV Digital, como os protocolos de transporte seguro;
- **Pacotes da especificação JCE (*Java Cryptography Extension*) 1.0** : implementa especificamente funcionalidades de segurança relacionadas à criptografia;
- **Pacotes da especificação SATSA (*Security and Trust Services API*) 1.0.1** : permite a comunicação com dispositivos externos, como smartcards, utilizando o protocolo APDU (do inglês, *Application Protocol Data Unit*);
- **Pacotes da especificação JavaTV** : atua como modelo de gerenciamento de aplicações, provendo funcionalidades específicas para TVD, além de incluir a API JMF (*Java Media Framework*);
- **Pacotes da especificação JavaDTV** : determina funcionalidades específicas de TVDi como tratamento de eventos do controle remoto e interface gráfica com LWUIT. Os principais pacotes são: `com.sun.dtv.application`, `com.sun.dtv.broadcast`, `com.sun.dtv.filtering`, `com.sun.dtv.io`, `com.sun.dtv.locator`, `com.sun.dtv.media`, `com.sun.dtv.net`, `com.sun.dtv.platform`, `com.sun.dtv.reso`

urces, com.sun.dtv.service, com.sun.dtv.smartcard, com.sun.dtv.transport, com.sun.dtv.tuner, com.sun.dtv.tuner. com.sun.dtv.ui e com.sun.dtv.lwuit; e

- **Pacotes do Ginga-J:** contém funcionalidades exclusivas do SBTVD contidas nos pacotes br.org.sbtvd.net, br.org.sbtvd.net.si (ARIB B.23), br.org.sbtvd.net.tuning (Tuner Control API), br.org.sbtvd.bridge, br.org.sbtvd.net.rc (Return Channel API), br.org.sbtvd.ui (Graphics Planes API) e br.org.sbtvd.interactiondevices (Device Integration API).

2.7 T-LEARNING

Existem várias referências para o significado do *T-Learning* destacadas conformes os autores abaixo:

Bertoti et al. (2004) afirma que o termo *T-Learning* é utilizado para caracterizar qualquer tipo de aplicação para TVD voltada para fornecer serviços interativos educacionais.

Jokipelto (2005) complementa afirmando que *T-Learning* é a convergência entre as tecnologias de TVD e *E-Learning*, que se refere ao uso da tecnologia computacional para prover treinamentos ou outras atividades educacionais através da Internet ou não. Neste contexto encontram-se as tecnologias IP, TVD e dispositivos móveis. A Figura 15 apresenta o esquema entre essas tecnologias.

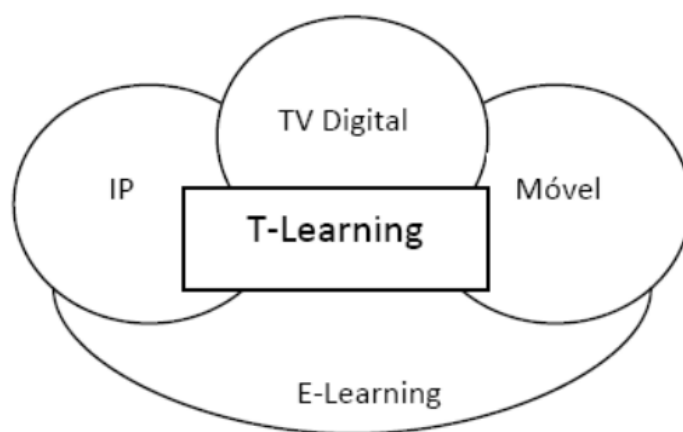


Figura 15 - Convergências de tecnologias.
Fonte: Adaptado de Jokipelto (2005).

Em Bates (2003), *T-Learning* significa o acesso a materiais de aprendizagens ricos em vídeos, através de uma TV ou um dispositivo mais parecido com a TV do que um computador.

Sancin (2009) afirma que o termo *T-Learning* pode ser relacionado ao uso de materiais de aprendizagem interativos e ferramentas que utilizam um receptor digital.

Pazos-Arias (2006) caracteriza o *T-Learning* como uma combinação de educação e entretenimento, introduzindo um novo termo: “*edutainment*” (*education* + *entertainment*). Esse termo ainda não tem um consenso de tradução para a língua portuguesa. Entretanto, se uma regra similar de formação for aplicada, tem-se “edutretenimento” (educação + entretenimento). Assim, edutretenimento, ou *edutainment*, envolve as características atrativas dos programas de TV e as características educacionais dos sistemas virtuais integrados (*software*) de *E-Learning*, de acordo com a Figura 16.

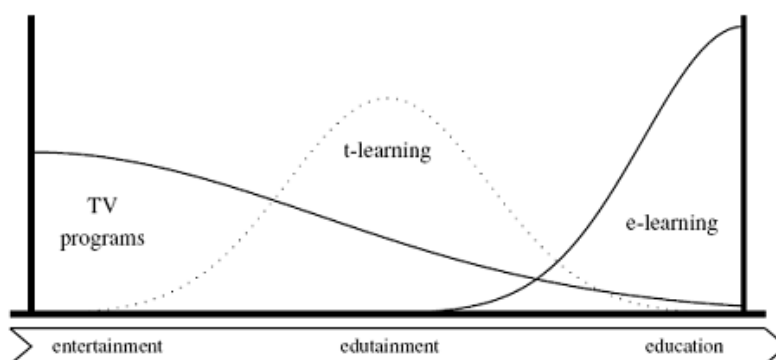


Figura 16 - Visão de *T-Learning* que associa educação e entretenimento.
Fonte: Adaptado de Franco (2009).

O conceito empregado nesta dissertação sobre o *T-Learning* foi o definido por Sancin (2009). Assim, foi feito o uso de materiais de aprendizagem interativos como jogos, vídeos, relações gráficas, textos e imagens, que foram combinados para formar aplicações para TVD e poderem ser executadas através de um receptor digital.

2.7.1 Classificação de processos educativos em *T-Learning*

Bates (2003) justifica que a importância da televisão como instrumento educativo está na popularidade do aparelho, que está presente na maioria absoluta dos lares tanto de países desenvolvidos como em desenvolvimento como é o caso do Brasil. Neste cenário

segundo Walldén e Soronen (2004), a classificação de processos educativos é importante para especificar as formas de reconhecimento de aprendizagem e uma metodologia para um público alvo de acordo com o contexto e os propósitos organizados. Estas autoras classificam os processos educativos em:

- Processos educativos formais: Aqueles fornecidos por instituições de ensino e treinamento formais e que fornecem diplomas e certificados reconhecidos pelos órgãos oficiais do país;
- Processos educativos não formais: Referem-se aos processos educativos fornecidos por instituições de ensino e treinamento, mas que não fornecem tipicamente diplomas ou certificados para o conhecimento difundido. Normalmente é obtido através de atividades em grupo de organizações da sociedade civil;
- Processos educativos informais: Referem-se aos processos educativos em que o indivíduo adquire valores, habilidades ou conhecimentos gerais a partir de experiências diárias em casa, no trabalho ou em qualquer ambiente propício; e
- Processos educativos acidentais: São aqueles em que o indivíduo adquire conhecimentos involuntariamente ou sem a percepção do processo educativo em andamento. Isso pode acontecer durante a realização de processos recreativos como participação em jogos, assistir a certos filmes ou programas de competições. Este interessante caso é bastante difundido na literatura sob o termo “*Edutainment*” em referência à junção das palavras *Education* e *Entertainment*.

Esta última Classificação definida por Walldén e Soronen (2004) é importante para esta dissertação, pois fornece a oportunidade de aplicar conceitos de acordo com um processo educativo, que explore o contexto de jogos, através de aplicações de TVDi. Entre os possíveis tipos de aplicações está a modalidade chamada de *Quiz*, que pela aplicabilidade neste trabalho será abordada no tópico seguinte.

2.7.2 Definição e justificativa do *Quiz*

Costa (2008) definiu *Quiz* como um termo da língua inglesa que se refere a uma prova, exame ou teste em que o usuário responde de acordo com seus conhecimentos

específicos ou opiniões pessoais às perguntas que sejam apresentadas. Na prática, aplicações interativas do tipo *Quiz* são aquelas que apresentam ao usuário uma ou mais perguntas, acompanhadas de múltiplas alternativas e tendo uma ou várias respostas corretas que podem ser escolhidas pelo usuário.

A resposta do usuário para cada pergunta é armazenada e, com a finalização do processo, devido à ocorrência de um evento disparado pelo usuário, o programa que controla o *Quiz* apresenta as respostas ao usuário. Havendo a possibilidade e disponibilidade do canal de retorno às respostas são enviadas para algum outro servidor que cuida de processar e armazenar os resultados fornecidos.

Com esse princípio, de coletar opinião ou conhecimento do usuário, aplicações de tipo *Quiz* podem fornecer importantes informações para as provedoras de conteúdo, sejam em programas de entretenimento e concursos, mas também para direcionamento de programação de acordo com o desejo do público alvo e classificação de processos educativos existentes em *T-Learning* apresentados na Figura 17.

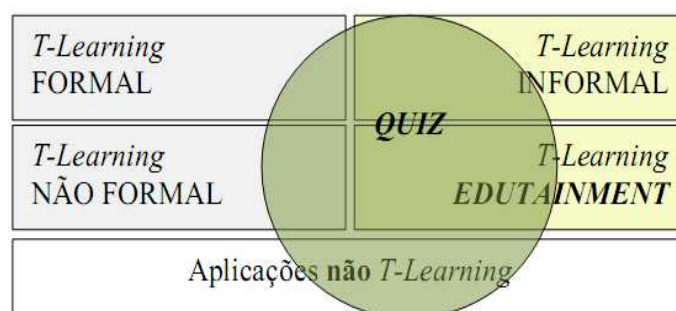


Figura 17 - *T-Learning* sobre o modelo de aplicação do tipo *Quiz*.
Fonte: Adaptado de Costa (2008).

2.7.3 Vantagens e desvantagens de *T-Learning* para o Brasil

Para Monteiro (2009) entre as principais vantagens deste paradigma estão:

- TV já é utilizada como forma de adquirir informação;
- Nem toda casa possui acesso a Internet;
- Qualidade na transmissão;
- Oportunidade para prática informal e colaborativa através de grupos de estudos;
- Grande quantidade de aparelhos de TV no Brasil;

- Familiaridade que grande parte da população tem com a TV; e
- Oportunidade de prática de aprendizagem através de objetos digitais.

As desvantagens também são identificadas por Monteiro (2009) as quais se podem citar:

- Ainda existem poucos investimentos na área;
- Carência de divulgação de conhecimento para sociedade;
- O canal de retorno pode trazer custos relacionados à estrutura de ambiente físico, atrapalhando adoção de serviços, fazendo com que algumas aplicações sejam impraticáveis; e
- A grande parte das funcionalidades oferecidas pela TV necessita do uso do controle remoto que limita a interação entre o telespectador e a TV no momento em que precisa usar uma aplicação interativa, e atender requisitos de usabilidade.

Estas limitações servem como um incentivo para pesquisas em desenvolvimento de aplicações para *T-Learning* visando contribuir para cenários que possam utilizar esta metodologia no Brasil.

2.8 OBJETOS DE APRENDIZAGEM DIGITAIS

Assim como o conceito de *T-Learning*, o conceito de objetos digitais de aprendizagem é muito vasto na literatura. Alguns desses conceitos são apresentados para contextualizar neste trabalho qual foi o conceito empregado, de acordo com os autores apresentados a seguir:

Wiley (2000) define objeto de aprendizagem (OA) como recursos digitais criados para suporte ao ensino e que permitem também a reutilização. Este conceito é bastante abrangente e faz com que *softwares*, imagens, textos, apresentações em *PowerPoint* ou animações produzidas em *Flash* sejam considerados OA.

Segundo Weller et al. (2003), um OA é uma parte digital do material da aprendizagem que se dirige a um tópico claramente identificável ou resultado da aprendizagem e tem o potencial de reutilização em contextos diferentes podendo ser utilizado via computador ou TV.

Gutierrez (2004) afirma que um OA é um objeto utilizado como meio de ensino/aprendizagem através de cartazes, maquetes, canções, peças teatrais, apostilas, filmes, livros, jornais, páginas da *web*, jogos e mapas conceituais. Esta autora acrescenta que a maioria desses OA podem ser reutilizados, modificados ou servir para outros objetos que não são determinados e que para isso é preciso saber explorá-los.

Spinelli (2007) afirma que um OA pode ser considerado objeto virtual de aprendizagem que auxilie na aprendizagem de algum conceito e, ao mesmo tempo, estimule o desenvolvimento de capacidades pessoais, como por exemplo, imaginação e criatividade. Dessa forma, um objeto virtual de aprendizagem pode contemplar um único conceito que engloba todo um corpo de uma teoria ou pode compor um percurso didático, envolvendo um conjunto de atividades, focalizando apenas o determinado aspecto do conteúdo envolvido ou formando, com exclusividade, uma metodologia para um determinado trabalho.

Para Sá Filho e Machado (2004), um OA pode ser uma montagem de recursos disponíveis e criados por desenvolvedores, como imagens, textos e algum outro elemento. O objetivo do OA é estimular a reflexão por parte dos alunos em um determinado assunto que esteja sendo ministrado contribuindo assim para o aprendizado. Eles afirmam que os OA em ambientes de aprendizagem podem torná-los ricos e flexíveis pelas possibilidades de serem utilizados como recursos simples ou combinados, formando uma unidade maior e podem ainda serem utilizados em um determinado contexto e reaproveitados posteriormente em contextos similares.

Após esta explanação de conceitos sobre OA, neste trabalho decidiu-se por utilizar o conceito definido por Sá Filho e Macahado (2004). Para desenvolver os componentes gráficos que iriam compor as interfaces de TVDi foram realizadas combinações de textos, figuras e imagens para forma um ou vários OA, ou aproveitando recursos já existentes nos próprios OA, que pudessem ser explorados em um determinado ambiente de aprendizagem através do ensino da trigonometria.

2.8.1 Características de objetos de aprendizagem

Existem nas fontes de pesquisas como publicações em artigos, livros e revistas características que favorecem o uso de OA na área educacional. Elas são consideradas importantes, pois através delas são apresentadas opções que podem ser exploradas na aplicabilidade no *framework* desta dissertação como:

- Reusabilidade: um OA deve ser projetado em uma estrutura modular obedecendo às regras de orientações a objetos com objetivo educacional explícito (LIRA, 2008);
- Interoperabilidade: Desenvolvidos para um ambiente ou plataforma, eles podem ser utilizados em outros ambientes, sem a necessidade de modificações ou adequações (AUDINO; NASCIMENTO, 2010);
- Flexibilidade: os OA são construídos de forma simples apresentado inicio, meio e fim, que podem ser reutilizáveis sem nenhum custo com manutenção (LONGIMIRE, 2001; SÁ FILHO; MACHADO, 2004);
- Metadados: Permitem descrever as características importantes dos OA. Essas informações serão úteis para a sua catalogação no repositório de objetos e, posteriormente, poderão ser localizados por um sistema de busca e reutilizados por qualquer membro de uma instituição de ensino. A definição desses metadados deve permitir uma categorização por diferentes critérios, facilitando a busca dos recursos (TAROUCO et al., 2003);
- Customização: Sendo objetos de aprendizagem independentes, a idéia de utilização em um curso, especialização ou qualquer tipo de aprendizagem torna-se viável, sendo que cada recurso educacional pode utilizar-se dos objetos e arranjá-los da forma que mais convir (MIRANDA, 2004);
- Durabilidade: capacidade de manter sua funcionalidade por um longo período de tempo sem a necessidade de se criar um novo projeto ou uma nova configuração (FRANCO, 2009); e
- Interação: relação entre o indivíduo e o sistema computacional por meio de manipulação de interfaces gráficas. Ela define o processo de comunicação entre o usuário e o sistema durante a realização de tarefas (PADOVANI; MOURA, 2008).

2.8.2 Aspectos pedagógicos para objetos de aprendizagem

Para identificar como os recursos de OA podem ser mais bem aproveitados em ambiente de aprendizagem foram realizadas pesquisas sobre algumas abordagens educacionais existentes, afim de alinhar seu modo de utilização às práticas pedagógicas que se buscar implantar.

Sobre a visão de Piaget (2001, Apud Ribeiro et al., 2011), a criatividade é avaliada através do campo da epistemologia genética e todo sujeito que realiza um trabalho e tem idéias novas, ele as cria conforme seus esforços. Piaget também afirma que existem três condições que auxiliam a criatividade, sendo elas: a possibilidade de se trabalhar sozinho e fazer seu próprio caminho intelectual; ter acesso a conteúdos de diversas áreas; e ter a presença de um adversário intelectual, como um interlocutor que confronte as hipóteses iniciais. Ele afirma que o conhecimento é uma estruturação da realidade, constituindo-se como uma construção por parte do sujeito. Logo, o desenvolvimento da inteligência é uma criação contínua, caracterizado pelo aparecimento de estruturas totalmente novas ou, pelo menos, refeitas em função de novidades, através do processo de abstração.

A abstração é agir sobre objetos, ou sobre as próprias ações, e retirar dessas ações alguma coisa (abstração empírica). Após agir sobre o meio (objetos, relações sociais, etc.), o sujeito passa a retirar qualidades da própria coordenação de ações. Assim a gênese da criação das novidades está na capacidade de retirar as qualidades da coordenação de suas ações, através do processo de abstração. Dessa forma, acredita-se que ao criar um OA para ser manipulado é possível fazer uso das possibilidades em um ambiente de aprendizagem para estruturar o conteúdo e explorá-los, aproveitando a inteligência coletiva.

Contribuindo ainda mais para campo da epistemologia genética de Jean Piaget, Menezes et al. (2006) abordam características onde os OA podem atender necessidades pedagógicas em:

- Criação de diversos níveis de interação para objetos, tal como ocorre em videogames, onde os jogos possuem vários graus de dificuldades;
- Incorporação de agentes inteligentes que desempenham, de alguma forma, o papel de um assistente que possibilitar investigar a exploração de conteúdos por parte do aluno;
- Utilização de diferentes formas, possibilitando a criação de ferramentas de apoio à análise do professor e reflexão do aluno;
- Facilitação de utilização (disponíveis na rede), dando acesso ao grupo de usuário em interação; e
- Reconfiguração, extensão e adaptação para ampliação de suas possibilidades de uso.

A abordagem comportamentalista de Skinner (1982) permite verificar que a medida que o objeto de aprendizagem é avaliado pode dar suporte aos processos de aprendizagem desta forma:

- Apresenta informações em seções breves;
- Testa o aluno após cada seção;
- Fornece recompensa para respostas corretas;
- Só permite seguir para outro nível se obtiver resposta esperada do aprendiz;
- Propõe questões que incentivam a memorização; e
- Obriga o aluno, no caso de erros, a retornar ao ponto anterior.

2.9 FRAMEWORK

Visando construir estruturas de classes para serem reutilizadas em aplicações de TVDi, esta dissertação baseou-se no paradigma de orientação a objetos. Conforme Sommerville (2011), o reuso da orientação a objetos é melhor suportado em processo de desenvolvimento orientado a objetos por meio das abstrações de alta granularidade que permite que um sistema possa ser dividido em partes, chamadas *de frameworks*. Segundo Schimidt et al. (2004), o *framework* é uma estrutura genérica estendida para se criar uma aplicação ou subsistema mais específico como : um conjunto integrados de artefatos de *softwares* definidos em classes, objetos e componentes que colaboram para fornecer uma arquitetura reusável para uma família de aplicações relacionadas.

Como se trata de um paradigma orientado a objetos é composto de classes abstratas, concretas e provê um modelo de interação ou colaboração entre as instâncias das classes. Neste contexto um *framework* é utilizado através de configuração ou conexão de classes concretas e derivação de novas classes a partir de classes abstratas que podem fazer parte do próprio *framework* ou de um conjunto de APIs, que estejam sendo utilizadas para o suporte a uma aplicação.

Para Gamma et al. (2000 apud PEROZZO, 2011), um *framework* é um conjunto de classes cooperantes para construção de projetos reutilizáveis. Perozzo (2011) afirma também que a utilização de um *framework* se justifica por ele implementar uma arquitetura que pode ser utilizada de modo a tornar o desenvolvimento de sistemas computacionais mais rápidos e produtivos, facilitando a captura de decisões de projetos que são comuns ao domínio de uma aplicação, permitindo a redução do tempo utilizado com o

desenvolvimento de novas aplicações. Dessa forma a principal contribuição de um *framework* é a definição de uma arquitetura, projetada para suportar todas as aplicações de um domínio de maneira flexível e extensível (GAMMA et al., 2000).

De acordo com Fayad et al. (1999 apud CARNEIRO, 2003) os *frameworks* orientados a objetos podem ser vistos como uma categoria de abstrações arquiteturais que suportam o projeto e a construção de uma arquitetura lógica. Este autor afirma que o uso das abstrações no desenvolvimento de *software* é considerado uma vantagem, pois torna o desenvolvimento mais fácil, já que permite o desenvolvedor trabalhar com menos elementos, além disso, a abstração facilita a reutilização. Este conceito de aplica quando os projetos reutilizáveis e as implementações são abstraídas de contextos específicos e representadas as relações generalizadas.

Para obter benefícios em utilizar uma arquitetura orientada a objetos no formato de *framework para TVDi* nesta dissertação foi preciso seguir requisitos necessários nas características:

- Usabilidade: o *framework* deve oferecer estrutura de classes bem organizada e documentada, além de classes que permita a fácil navegação de interfaces gráficas ou componentes existentes no *framework*;
- Reusabilidade: os requisitos existentes no *framework* devem possibilitar recorrer a conceitos de interfaces, herança ou extensibilidade;
- Modularidade: permite ao desenvolver que aplicações sejam distribuídas em formato de pacotes de maneira que organize diversos subsistemas ou conjuntos de aplicações. Um dos grandes benefícios a serem adquiridos é a facilidade de tratamentos de erro e lidar com escopo menor relacionado ao tamanho da aplicação; e
- Domínio específico: o *framework* é desenvolvido para tratar o problema proposto em um campo específico, no caso deste trabalho e voltado para definir módulos de classes que utilizam APIs para tratamento de eventos e manipulação de interfaces gráficas para TVDi, com o propósito de definir um ambiente de ensino de apoio à matemática através da trigonometria.

2.9.1 Classificação de *frameworks*

Segundo Assis e Suzano (2003), a classificação de *framework* pode se dá a partir do modo e da necessidade de como ele é empregado, que pode ser voltado para dados ou para arquitetura. Caso o *framework* seja voltado para dados o desenvolvimento de aplicações é realizada pelas várias maneiras de combinar a instâncias das classes já existentes. Já os *frameworks* voltados para a arquitetura possuem subclasses desenvolvidas tendo como base as classes pré-definidas do *framework*.

Sobre esses requisitos para os autores Fayad e Schmidt (1997) os *frameworks* são classificados nas categorias a seguir:

- *Framework* de Infraestrutura do Sistema: simplificam o desenvolvimento da infra-estrutura de sistemas portáteis e eficientes, como por exemplo, os sistemas operacionais, sistemas de comunicação, interfaces com o usuário e ferramentas de processamento de linguagem. Em geral são usados internamente em uma organização de *software* e não são vendidos a clientes diretamente;
- *Framework* de integração de *middleware* : são usados, em geral, para integrar aplicações e componentes distribuídos. Eles são projetados para melhorar a habilidade dos desenvolvedores em modularizar, reutilizar e estender sua infra-estrutura de *software* para funcionar em um ambiente distribuído. Exemplos dessa classe de *framework* são o “*Object Request Broker*” (ORB), “*middleware*” orientado a mensagens e bases de dados transacionais; e
- *Framework* de Aplicações Corporativas: são empregados no desenvolvimento de aplicações comerciais para usuários finais e são os mais caros para serem desenvolvidos quando comparados com outros tipos de *framework*. Como exemplo, tem-se os *frameworks* para aplicações financeiras ou sistemas contábeis;

A outra classificação dos *frameworks* é quanto à maneira que acontece a extensão. Esta classificação se divide em três partes:

- *Framework* de Caixa Branca: baseia-se nas características da orientação a objetos, como herança, métodos abstratos que são implementados na subclasse, métodos com uma implementação padrão que pode ser definida na

subclasse, ligação dinâmica e faz-se necessário aos desenvolvedores um bom conhecimento interno do *framework*;

- *Framework* de Caixa Preta: baseia-se na composição de objetos, ou seja, ele define as interfaces com um contrato definido, de modo que elas sejam conectadas ao *framework* por meio de composição de objetos, tendo maior facilidade de uso e extensão que os de caixa branca; e
- *Framework* de Caixa Cinza (Híbrido): utiliza as características do *framework* de caixa branca e caixa preta provendo tanto a flexibilidade quanto a capacidade de extensão.

A classificação do *framework* desta dissertação é a mesma classificação do *framework* de caixa cinza, por se tratar do emprego da orientação a objetos e o reuso constante de métodos abstratos de interfaces para manipulação de eventos de controle remoto e construção do ciclo de interfaces gráficas para TVDi.

2.9.2 Vantagens e desvantagens de *frameworks*

Para Oliveira (2007), um *framework* ao possuir as características das linguagens de programação orientadas a objetos: herança, polimorfismo, abstração de dados, modularidade entre outras. Essas características facilitam a reusabilidade e, com certeza, esta é uma das vantagens que chama mais a atenção dos pesquisadores. Um *framework* pronto, há a maximização do reuso, ou seja, reutiliza-se, em geral, análise, projeto, código e testes e, com isso, acontece uma redução significativa do tempo de desenvolvimento de novas aplicações. As outras vantagens apontadas são:

- Diminuição das linhas de código em um módulo do *framework*;
- Possibilidade de utilizar outras técnicas de reuso em conjunto, por exemplo, padrões de projetos (*design patterns*) e componentes;
- Diminuição dos erros no código já que ele é usado em várias aplicações; e
- Aumento da qualidade de *software*, facilidade na manutenção, pois quando um erro é corrigido no *framework*, automaticamente, ele é corrigido nas aplicações desenvolvidas a partir deste *framework*.

Algumas desvantagens são apontadas por Mattson (1996):

- Dificuldade em desenvolver um *framework*, considerando, por exemplo, para qual plataforma o *framework* vai ser construído e quais suas limitações;

- Caso o *framework* não possuir uma documentação apropriada, certamente ele não será bem utilizado; e
- O processo de depuração de erro pode ser complicado, porque é difícil distinguir quando o erro é do *framework* ou da aplicação que o utiliza. Caso o erro esteja no *framework* pode ser impossível o usuário conseguir corrigi-lo.

Fayad e Schmidt (1997) destacaram também que as outras desvantagens para utilizar um *framework* são: a dificuldade para entendê-lo e consequentemente usufruir de seus benefícios; o esforço demandado para desenvolvê-lo; e a falta de fluxo de controle explícita pode dificultar a depuração de alguns erros.

2.10 TRABALHOS RELACIONADOS

Desde o início da implantação do ISDB-Tb no Brasil vem ocorrendo pesquisas e desenvolvimento de aplicações para este padrão, baseados em outros sistemas de televisão digital em utilização no mundo. Neste cenário tem-se a oportunidade de explorar duas linhas de pesquisas. A primeira linha de pesquisa trata da possibilidade de desenvolver aplicações de TVD que possam servir de arquitetura para outros padrões de TVD e serem integradas ou adaptadas ao sistema brasileiro.

A segunda linha de pesquisa trata a possibilidade de desenvolver arquiteturas baseadas no próprio sistema brasileiro considerando as limitações ainda existentes. Sobre essas perspectivas foi realizado um levantamento de materiais bibliográficos relacionados a alguns trabalhos sobre *frameworks* para plataforma de TVDi, que venha a contribuir com o tema de investigação apresentados a seguir :

O *framework* Ginga Game desenvolvido por Barbosa e Clua (2009), abordou a construção de um módulo de classes para elaboração de jogos de perguntas e repostas para TV Digital, conforme a Figura 18.

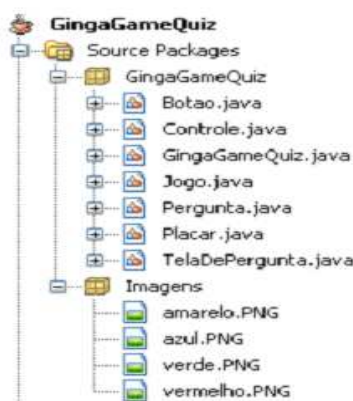


Figura 18 - Módulo de classes do jogo de perguntas e respostas.
Fonte: Adaptado de Barbosa e Clua (2009).

Na Figura 19, é apresentada a aplicação gerada por este *framework* com apresentação da interface de perguntas e respostas contendo uma pergunta e quatro respostas a serem apresentadas ao jogador. As respostas aparecem através das cores vermelho, amarelo, verde e azul, usando o botão das cores do controle remoto para tornar o processo mais intuitivo. Para cada pergunta o jogador deve pressionar o botão referente a cor que indica a resposta que o mesmo supõe ser correta. Em seguida são mostradas as quantidades de acerto e erro de cada jogador.

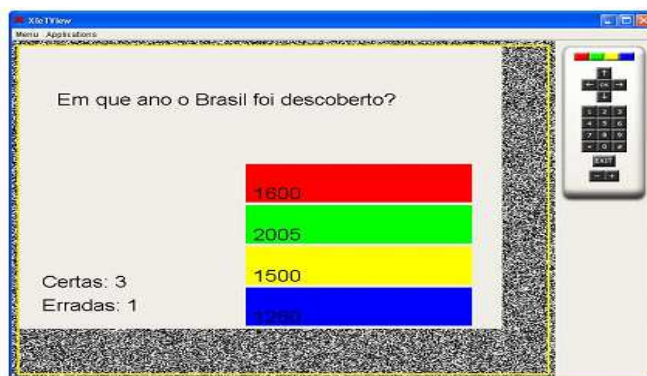


Figura 19 - Interface de perguntas e respostas.
Fonte: Adaptado de Barbosa e Clua (2009).

Este trabalho mostrou ser importante para esta dissertação porque através dele foi possível identificar o módulo das classes do jogo implementado para uma aplicação do tipo Quiz simples em um ambiente de TVD utilizando o Ginga-J. O que é comum com o *framework* desta dissertação é que existe uma arquitetura de interface para manipulação de interfaces gráficas de perguntas e respostas exibindo resultado. A diferença é que o *framework* desta dissertação disponibiliza uma arquitetura desenvolvida através das APIs

do *middleware* MHP que pode ser comparada e adaptada para os padrões do Ginga-J de acordo com os seus recursos disponíveis através de suas APIs, além de possibilitar desenvolvimento de outros módulos de aplicações que não seja somente um jogo.

Em Maia (2010) foi desenvolvido o *framework* GameTVD para geração de jogos, que consistiu em um conjunto de classes para permitir a criação de novos jogos, apenas sendo necessário ao usuário adicionar objetos e cenários para criação dos jogos, através de uma interface gráfica. Na Figura 20 é apresentada a interface inicial da composição do jogo *Pac-Man* que contém botões de início para iniciar um jogo, de ajuda com as instruções para jogar, sair ou finalizar o jogo.

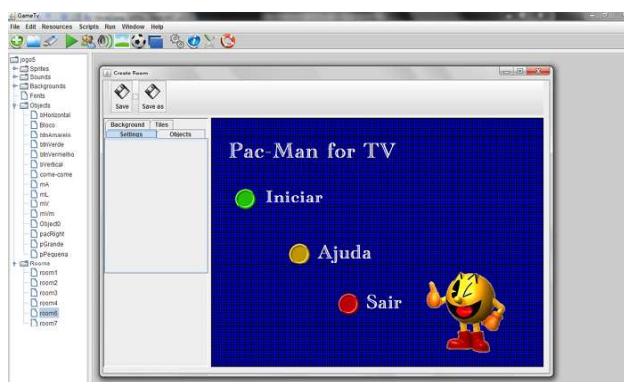


Figura 20 - Interface inicial da composição do jogo *Pac-Man*.
Fonte: Adaptado de Maia (2010).

Este trabalho contribui para analisar interfaces gráficas desenvolvidas sobre as bibliotecas do padrão europeu utilizando a API HAVI e as bibliotecas do padrão brasileiro utilizando a API LWUIT. Além do mais foi possível definir como estruturar a aplicação que pode ser executada em padrão de TVD. A diferença do trabalho apresentado com o desta dissertação, é que na dissertação existe também a possibilidade de definir módulos de estrutura no *framework* não só para manipular um jogo, mas para outras aplicações que explorem outras definições de objetos digitais de aprendizagem, relacionados a vídeos, mapa conceitual, análise gráfica ou um desafio com postagens de perguntas.

O sistema desenvolvido por Oliveira et al. (2009) disponibilizou uma arquitetura semelhante ao desta dissertação através do *middleware* MHP, utilizando as APIs JavaTV e HAVI para o ensino da matemática via TV Digital interativa e para o perfil do ensino fundamental com foco em funções do primeiro grau. Foi possível disponibilizar através deste ambiente manipulações com vídeos, questionários, programação agendada para uma

aula e enquete para verificação dos serviços oferecidos pela ferramenta classificados como: ruim, bom e ótimo.

A Figura 21 exibe a tela principal do aplicativo com as opções correspondentes ao ensino da matemática via TV Digital interativa.



Figura 21 - Tela principal do aplicativo Digital Math.
Fonte: Adaptado de Oliveira et al. (2009).

Embora este trabalho não tenha sido considerado um *framework*, pois houve uma implementação e uma documentação baseadas nos serviços a serem oferecidos pela TVD, e não em uma estrutura de classes que permitissem ser reutilizadas por outras classes, abordando características de *framework*. Considerando esta diferença para o *framework* da dissertação a outra seria que os módulos dos OA desenvolvidos nesta dissertação são mais variados que os deste sistema, permitindo a maior disponibilidade dos recursos oferecidos.

O trabalho de Silva et al. (2004) desenvolveram a aplicação TV Escola interativa. Esta aplicação foi útil para esta dissertação para compreender as manipulações de interfaces gráficas para vídeos através das APIs utilizadas no *middleware* MHP. A Figura 22 mostra a tela do *menu* com áreas temáticas para cada vídeo.



Figura 22 - Tela de menu com áreas temáticas.
Fonte: Adaptado de Silva et al. (2004).

A outra característica percebida neste trabalho foi a possibilidade de estruturar vídeos como recursos de apoio de professores específicos à cada disciplina através da TV Digital promovendo aplicações com características de *T-Learning*.

No trabalho de Prota (2010) foi desenvolvido o *framework* MoonDo para o desenvolvimento de aplicações declarativas no ISDB-Tb. O objetivo deste *framework* é fornecer uma arquitetura modularizada para seus componentes gráficos e funcionais, que possam ser reutilizados de acordo com as características da aplicação que está sendo desenvolvida. Destacam-se neste *framework* as funcionalidades para manipulação de componentes de interfaces gráficas para permitir que o desenvolvedor modifique ou crie novos componentes gráficos e as funcionalidades já existentes possam ser reutilizadas.

Este *framework* foi desenvolvido para base de arquitetura em aplicações NCL e Lua, sendo que a parte de apresentação foi feita através de documentos NCL e as partes funcionais foram construídas através de *script* Lua. Visando melhorar e facilitar a manipulação de interface gráfica em relação ao tempo demandado para compreender como executar o que era responsabilidade da linguagem NCL e do *script* Lua o Moondo formalizou uma arquitetura para manipulação de componentes gráficos definidos em uma Scene. Na Figura 23 é possível perceber a organização da estrutura de classes do MoonDo.

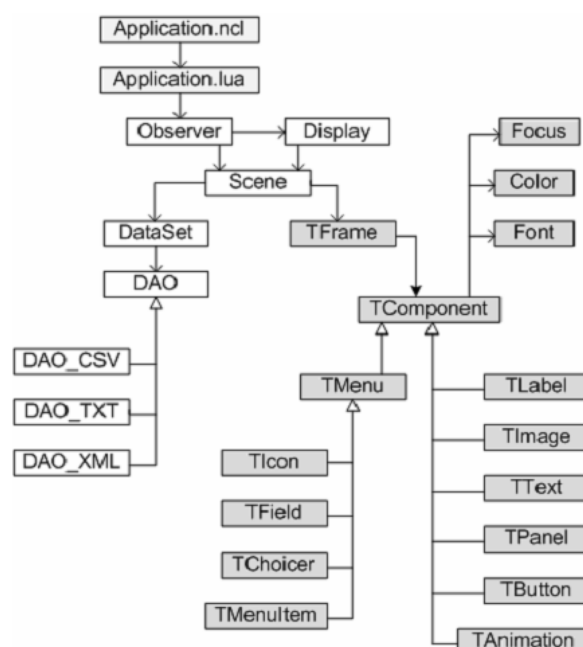


Figura 23 - Estrutura de classes do MoonDo.
Fonte: Adaptado de Monteiro et al. (2010).

Nesta arquitetura os *scripts* Lua chamam os componentes gráficos definidos em uma *Scene*. Na Figura 24 é possível visualizar um exemplo de componente gráfico definido neste *framework*.



Figura 24 - Tela com componentes gráficos do MoonDo.
Fonte: Adaptado de Prota (2010).

A diferença do MoonDo para o *framework* deste trabalho é que MoonDo foi desenvolvido para plataforma NCL-Lua, mas especificou uma estrutura de componentes definida como *Scene* para manipulação de interface gráfica semelhante ao deste trabalho definidas na APIs do *middleware* MHP. O *framework* desenvolvido neste trabalho não precisou definir estruturas extras como definidas em documentos NCL e na linguagem Lua, embora tenha ocorrido testes nesta plataforma. A própria *Scene* do *framework* desta dissertação permite desenvolver de forma direta os componentes gráficos que serão manipulados.

O trabalho de Belloti et al. (2008) desenvolveu o curso para prática do *T-Learning* através de objetos de aprendizagem e sua apresentação em uma arquitetura de *framework*, descrevendo a produção do conteúdo de aprendizagem, a transmissão pelo canal de retorno e o acesso à TV possibilitado pelo STB. Neste *framework* foi possível identificar que a estrutura dos OA da aplicação foi feita utilizando APIs do *middleware* MHP. Os professores realizavam a descrição de conteúdo em forma de jogo, imagens e textos em uma ferramenta personalizada para construção de interfaces para TVDi. Esses formatos eram convertidos em aplicativos e adaptados à estrutura do *framework* e disponibilizados em um ou mais STB aos alunos ou telespectadores cadastrados com acesso. Esse trabalho mostrou ser importante porque foi possível explorar como organizar aplicações com característica de *T-Learning* e os recursos através de OA.

A diferença do trabalho desta dissertação para este trabalho estar na construção de uma arquitetura que possa ser adaptada para outros padrões de TVD e não utilizar ferramentas para a criação de objetos de aprendizagem, que sejam adaptadas apenas no padrão europeu através do *middleware* MHP.

Arias et al. (2008) descreveram o *framework* Atlas para propor práticas de ensino baseada nas características do *T-learning*. Esse *framework* organizava serviços de acordo com as especialidades de cada conteúdo ministrado, preocupando-se com a lógica da sequência de atividades, regras aplicadas e a organização e apresentação de diversas fontes de aprendizagem como texto, imagens, sons e vídeos. Todos esses recursos eram disponíveis de acordo com o perfil do estudante e a estrutura deste *framework* também foi desenvolvida utilizando as APIs do *middleware* MHP. A diferença do *framework* desta pesquisa é que os conteúdos da aplicação não são apresentados de acordo com perfil de alunos, mas de forma geral para alunos com interesse à prática de trigonometria.

Já Lytras et al. (2002) descreveram três fases para definir e avaliar uma aplicação no cenário de aprendizagem para TV Digital interativa. A primeira fase definida de planejamento descreveu o problema abordado, as metas de aprendizagem como níveis cognitivos a serem alcançados e como promover interação para a solução do problema. A segunda fase especificou a execução do planejamento. Nessa fase foi utilizada a TV Digital interativa para promover interações entre alunos. Os autores perceberam que aumentou o grau de motivação dos alunos pelo fato de agregar entretenimento e comodidade ao acesso de recursos como imagens, vídeos, textos, jogos e outros serviços interativos.

Na terceira fase definida como avaliação foi feito um estudo identificando os recursos que já estavam dispostos de forma satisfatória e os que ainda precisavam ser mais bem organizados, buscando melhorar a interação dos alunos com a aplicação de TVDi em cenários diferentes para mesma aplicação. Embora este trabalho não tenha abordado a estrutura de um *framework*, ele foi importante, pois foi possível verificar e analisar como empregar objetos de aprendizagem para promover uma aplicação com características de ambiente de *T-Learning*.

No Quadro 1 é mostrada a comparação entre os trabalhos relacionados com o presente trabalho, levando em consideração os *frameworks* e aplicações para TVDi que foram desenvolvidos para empregos de objetos digitais. Os parâmetros de diferenciação de cada trabalho são: aplicação de TVDi, *framework*, tratamentos de eventos, variedades de módulos de OA, aplicação organizada por perfil e avaliação da ferramenta.

Trabalho	Aplicação de TVDi	<i>Framework</i>	Tratamento de eventos	Variedades de OA	Aplicação organizada por perfil	Avaliação da ferramenta
Barbosa e Clua (2009)	Sim	Sim	Sim	Não	Não	Não
Maia (2010)	Sim	Sim	Sim	Não	Não	Não
Oliveira et al. (2009)	Sim	Não	Sim	Não	Não	Não
Silva et al. (2004)	Sim	Não	Sim	Não	Sim	Não
Prota et al. (2010)	Sim	Sim	Sim	Não	Não	Não
Belloti et al. (2008)	Sim	Sim	Sim	Não	Não	Não
Arias et al. (2008)	Sim	Sim	Sim	Não	Sim	Não
Lytras et al. (2002)	Sim	Não	Sim	Não	Não	Sim
Trabalho da Dissertação	Sim	Sim	Sim	Sim	Não	Sim

Quadro 1 - Trabalhos relacionados.

2.11 CONCLUSÃO

Neste capítulo foram apresentados os conceitos explorados e trabalhos relacionados para desenvolver o *framework* deste trabalho. As pesquisas sobre as APIs para TVD, OA e *middlewares* foram importantes para identificar os principais recursos disponíveis para construção do *framework* deste trabalho, já o estudo detalhado dos modelos de *frameworks* permitiu identificar a sua categoria. Como será visto no próximo capítulo estes pontos destacados foram de fundamental importância para elaborações de visões de funcionalidades de interfaces gráficas para TVD a serem exploradas na formação do *framework* deste trabalho, e como estas serão organizadas para permitir a reusabilidade do projeto através de classes que utilizam recursos da orientação a objetos.

3 MODELAGEM DO *FRAMEWORK O-LEARNING TVDi*

Este capítulo apresenta a modelagem conceitual do *framework* desta dissertação atendendo a um domínio específico. Em seguida para construção deste *framework* é descrita a modelagem das atividades utilizadas em um modelo de processo de *software*, que especifica como foi o processo para abstrair os requisitos essenciais e definir funcionalidades e componentes a serem integrados, bem como possíveis adaptações para plataforma de *middleware* MHP e GINGA.

3.1 MODELAGEM CONCEITUAL DO *FRAMEWORK*

A principal contribuição desta dissertação está na especificação de um *framework* que possibilite definir um cenário de aprendizagem para trigonometria através de manipulação de objetos digitais de aprendizagem. Neste contexto se faz necessário identificar como será a formação deste *framework* para promover a construção de subsistemas que serão identificados como módulos de aplicações de OA. Estes módulos são comparados à visão de Gomes (2002), apresentada como exemplo na Figura 25, que descreve um *framework* dividida em duas partes, um núcleo (*Kernel*) e os pontos flexíveis (*hot-pots*).

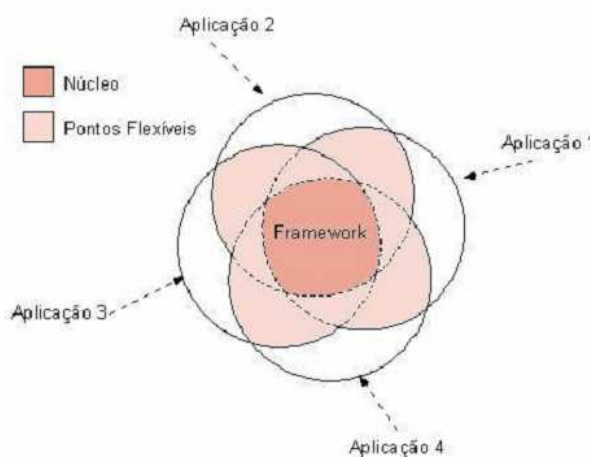


Figura 25 - Pontos comuns entre aplicações de mesmo domínio.
Fonte: Adaptado de Gomes (2002).

Conforme este mesmo autor, o núcleo envolve o que há de comum a todas as aplicações às quais o *framework* pode gerar e o conjunto de *hot-spots* descrevem os

diferentes componentes que cada instânciação do *framework* pode ter. Os *hot-spots* são classes que podem possuir uma implementação diferente para cada instância do *framework*, portanto eles se constituem incompletos até o momento de sua instânciação, além de seus próprios métodos eles podem usar os métodos e informações disponibilizadas pelo *kernel*, e em sua implementação podem estendê-los.

A partir desta contextualização de como é organizada as aplicações ou subsistemas de *framework*, o processo de desenvolvimento de cada aplicação é apresentado nas atividades de análise de domínio, projeto de *framework* e instânciação de *framework* definidos por Mattson (2000), conforme a Figura 26.

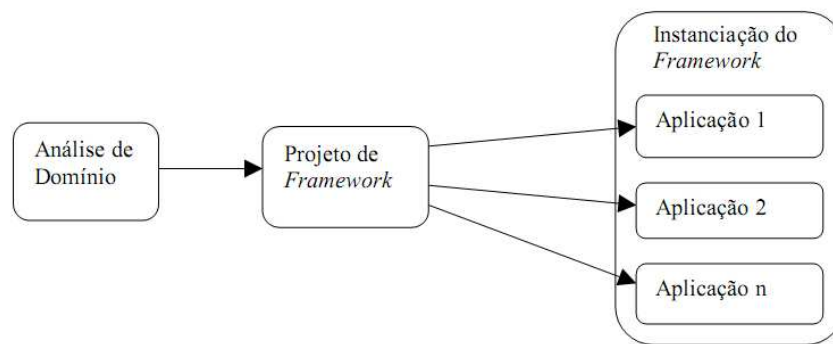


Figura 26 - Contextualização de desenvolvimento de aplicações para *framework*.
Fonte: Adaptado de Carneiro (2003).

A atividade de análise de domínio corresponde ao processo de organizar o conhecimento sobre alguma classe de problemas. Segundo Taligent (1994), um *framework* que atende um domínio específico cobre funcionalidades particulares a cada aplicação, por exemplo: *framework* para manufaturas, controle de produção, multimídia, engenharia financeira, entre outros. Nesta dissertação o *framework* atende ao domínio de classes de problemas para o ensino da matemática, possibilitando a reusabilidade de funcionalidades para navegação de interfaces de TVD, manipulação de eventos, manipulações de funções trigonométricas e manipulações de funções de jogo de perguntas e repostas na modalidade Quiz.

Ainda em Mattsson (2000), a atividade de projeto de *framework* tem como o objetivo construir um *framework* para ser flexível para aumentar a extensibilidade e poder ser usado em várias aplicações com o mesmo domínio. Este mesmo autor afirma que na fase de instânciação do *framework*, um *framework* pode ser instanciado uma ou mais vezes dentro de uma aplicação ou instanciado em diferentes aplicações.

Na seção seguinte será apresentado a modelo de processo de desenvolvimento definido para atender a este modelo conceitual.

3.2 MODELO DE PROCESSO DE DESENVOLVIMENTO DO *FRAMEWORK*

Conforme Sommerville (2011), um processo de software é um conjunto de atividades relacionadas que levam à produção de um *software*. Este mesmo autor relata que existem muitos processos de *softwares* diferentes, mas todos devem incluir as seguintes atividades: especificação, projeto, implementação, validação e evolução de *software*.

Na atividade de especificação são definidas as funcionalidades do *software* e as restrições sobre suas operações. Nas atividades de projeto e implementação é necessário produzir um *layout* da arquitetura e um *software* que atenda à especificação do domínio do problema. Na atividade de validação, o *software* deve ser avaliado para garantir que ele faça o que o cliente deseja. E na atividade de evolução deve ser permitido que o *software* atenda as necessidades mutáveis como, por exemplo, mudanças de arquitetura e adaptações de novos componentes a serem inseridos.

Para compreensão destas atividades executadas durante o desenvolvimento de um *software* com qualidade que satisfaça às necessidades de um projeto específico de acordo com o prazo, custo de desenvolvimento e domínio específico, se faz necessário definir um modelo de processo de *software*. Segundo Pressman (2006), um modelo de processo de *software* ou modelo prescritivo de processo descreve um conjunto distinto de atividades, tarefas, marcos e produtos de trabalhos (*software* produzidos), que são necessários para fazer a engenharia de *software* com alta qualidade.

Neste contexto este autor afirmar que um modelo de processo de *software* determina uma ação (ou ações) em termos de um conjunto de seqüências de tarefas que devem ser executadas em cada atividade para representação do trabalho, considerando a especificação de cada projeto que levam a produção do *software*.

O modelo de processo de *software* adotado para o desenvolvimento do *framework* desta dissertação é o modelo de engenharia de *software* orientada a reuso. De acordo com Sommerville (2011), neste modelo de processo de *software* contem as seguintes atividades (estágios): Especificação de Requisitos, Análise de Componentes, Modificação de Requisitos, Projeto de Sistema com Reuso, Desenvolvimento e Integração e Validação de sistema. A sequencia lógica da realização destas atividades é apresentada na Figura 27.

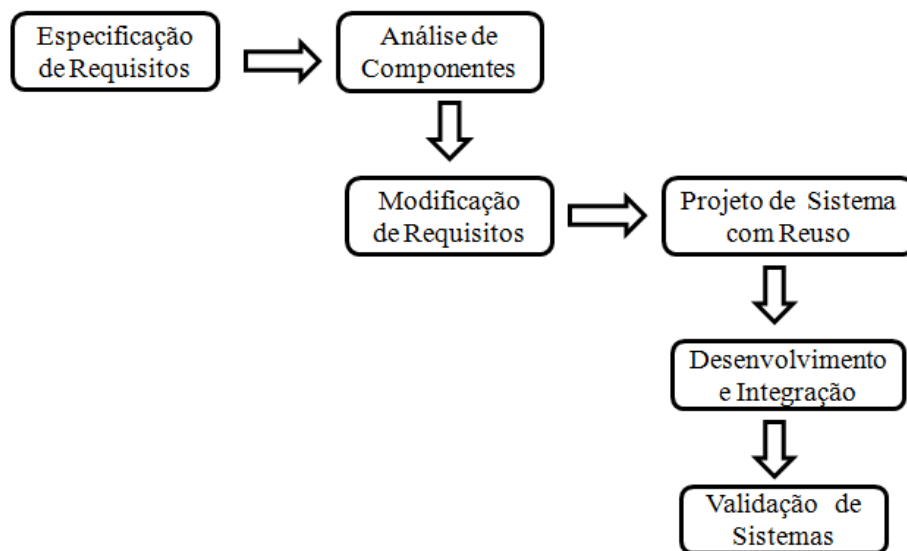


Figura 27- Atividades realizadas no desenvolvimento do *framework*.

Fonte: Adaptado de Sommerville (2011).

Com objetivo de detalhar o emprego destas atividades, nas próximas seções estão descritas estas atividades adaptadas a especificação de desenvolvimento do *framework* deste trabalho.

3.3 ESPECIFICAÇÃO DE REQUISITOS

Considerando que o objetivo deste *framework* é atender o ambiente de TVDi são considerados alguns requisitos:

- Prover suporte à implementação de módulos de aplicações de OA para serem executados em TVDs com auxílio de STBs;
- O *framework* ser desenvolvido sobre o paradigma de orientação a objetos, explorando recursos de reusabilidade como herança, e outros;
- O *framework* funcionar inicialmente em sistema operacional Windows com auxílio do emulador *Xletview* e IDE para desenvolvimento Eclipse, mas poder funcionar sem restrições em ambiente de sistema operacional Linux;
- O *framework* ser desenvolvido sobre o *middleware* MHP para teste e poder ser adaptado para funcionar em ambiente de *middleware* Ginga-J.
- implementar o *framework* na linguagem Java devido ao suporte de APIs para *middleware* MHP e Ginga-J, além de ser orientada a objetos;
- Verificar a viabilidade de construir o *framework* utilizando somente a plataforma Ginga-NCL e Ginga-J;

- Construir módulos de OA para facilitar testes em ambiente diferentes para TVD, permitindo o melhor controle, adaptação e organização dos módulos desenvolvidos; e
- Testar o uso dos módulos de OA em ambiente real de aprendizagem para verificar as funcionalidades e o nível da aceitação da ferramenta construída.

3.4 ANÁLISE DE COMPONENTES

Dada a especificação de requisitos, realizou-se uma busca de componentes considerados OA digitais para integrar uma aula de suporte ao ensino da trigonometria. Eles foram definidos em:

- Divulgação de textos em formatos de imagens sobre o surgimento da trigonometria;
- Exposição de imagens de mapas conceituais explorando as definições de trigonometria;
- Vídeos de trigonometria para o estudo específico de uma aula;
- Manipulação de funções básicas de seno e cosseno que permitem projetar seus resultados gráficos em uma circunferência como estudo de caso;
- Jogo de perguntas e respostas no modelo *Quiz*;
- Postagem de perguntas para resolução de desafios; e
- Gráficos de funções trigonométricas.

Depois de identificados esses objetos, em seguida, foi elaborado uma análise sobre as correspondências de suas funções para formar estruturas de módulos do *framework*. Nesta atividade foram verificados quais destes OA eram conteúdos apenas de apresentação e os que eram ao mesmo tempo de apresentação, mas que exigiam manipulação de procedimentos mais trabalhados nos eventos de controle remoto da TVDi, propiciando navegações e interações de interfaces gráficas entre o usuário e o ambiente.

3.5 MODIFICAÇÃO DE REQUISITOS

Durante esta fase foi elaborada uma revisão dos requisitos da atividade “análise de componentes”. Os componentes foram adaptados para atender o prazo de execução do trabalho. Neste caso, não foram implementados os módulos de gráficos de funções

trigonométricas que avaliassem como plotar e analisar as funções trigonométricas: Seno, Cosseno, Tangente, Cotangente, Secante e Cossecante.

3.6 PROJETO DE SISTEMA COM REUSO

Neste estágio de atividade o *framework* foi projetado, não para ser definitivo, mas para disponibilizar uma arquitetura reutilizável, visando melhorar os recursos oferecidos. Foi levado em consideração como desenvolver os componentes e organizá-los para prover o reuso das funcionalidades e classes do *framework* em ambiente de TVDi.

Através desta atividade foram construídas as visões dos integrantes envolvidos com o desenvolvimento do *framework* e uso da aplicação, de caso de uso, estudo de caso de aplicações criadas pelo *framework*, diagrama de classes e levantamento da arquitetura de execução do *framework*, apresentadas nos tópicos seguintes.

3.6.1 Integrantes do *Framework* e Uso da Aplicação

Nesta dissertação foram definidos dois (02) integrantes conforme a Figura 28. O primeiro integrante definido foi desenvolvedor que cria o *framework* e o usa para criar outras aplicações. O segundo integrante definido foi o usuário que usa a aplicação criada pelo desenvolvedor interagindo com suas interfaces e manipulando os recursos disponíveis.

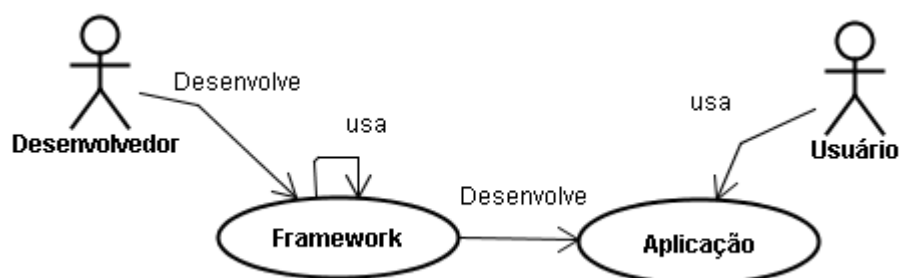


Figura 28- Integrantes do desenvolvimento do *framework* e uso da aplicação.

Diante deste cenário o desenvolvedor criou o *framework* explorando primeiramente as características de caixa branca e foram definidos em sua arquitetura interna os recursos de herança, superclasses, subclasses, métodos padronizados e instâncias de classes. Segundo Deitel (2010), estes recursos são definidos da seguinte forma:

- Superclasses são classes genéricas que determinam métodos e atributos comuns que podem ser reutilizados nas subclasses;
- Subclasses são classes especializadas que reutilizam métodos padronizados e atributos das superclasses, além de implementações específicas de seus métodos;
- Herança é uma forma de reutilização de *software* na qual uma nova classe (subclasse) é criada, observando membro de uma classe existente (superclasse) e aprimoradas com capacidades novas ou modificadas;
- Métodos padronizados são tarefas realizadas nas classes de modo que possam ser aproveitadas em outras classes; e
- Instâncias de classes são objetos criados durante a execução de um programa.

Para características de caixa preta foram definidas e exploradas os recursos de interfaces com métodos abstratos para manipulações de eventos para o controle remoto e controle do ciclo de vida de aplicações em ambiente de TVD. Conforme Deitel (2010), as interfaces permitem várias formas de aplicar a reusabilidade e definir polimorfismos de reusabilidade de métodos. Assim a interfaces definem um conjunto de métodos abstratos que podem ser reconstruídos de acordo com as suas necessidades.

Com o objetivo de contextualizar a aplicação gerada pelo *framework* desta dissertação na próxima seção serão apresentadas as funcionalidades e interações executadas pelo usuário.

3.6.2 Caso de uso

De acordo com Fowler (2005), o diagrama de caso de uso descreve um cenário através de sequencia de passos com interação entre o usuário e o sistema. No contexto desta dissertação o caso de uso apresentado na Figura 29 mostra as funcionalidades dos usuários que poderão ser executadas nos módulos do *framework* desta dissertação via TVDi.

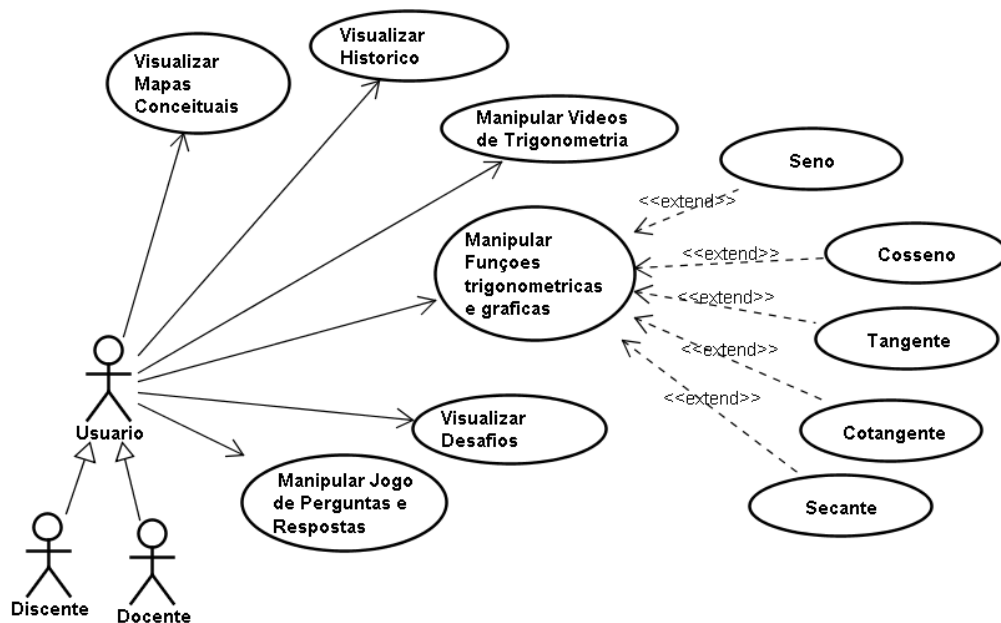


Figura 29- Caso de uso das funcionalidades dos usuários.

Em seguida serão apresentadas as descrições dos fluxos principais dos casos de usos individuais, contextualizados como estudo de caso nas funcionalidades executadas pelo discente, conforme os quadros abaixo:

Caso de Uso: Visualizar Mapas Conceituais (UC01)
Objetivo: Representar conceitos de Trigonometria
Ator: Discente
Fluxo Principal: 1. Este Caso de uso se inicia quando o discente na tela de <i>menu</i> principal escolhe a opção de Mapas Conceituais; 2. É exibida em seguida uma tela com o Mapa Conceitual geral de Trigonometria; 3. O discente visualiza o mapa geral e pode navegar pelas as outras telas de mapas através das opções continuar, voltar e sair.

Quadro 2 - Descrição do caso de uso "Visualizar Mapas Conceituais".

Caso de Uso: Visualizar Histórico (UC02)
Objetivo: Disponibilizar informações sobre a trigonometria.
Ator: Discente
Fluxo Principal: 1. Este Caso de uso se inicia quando o discente na tela de <i>menu</i> principal escolhe a opção de histórico; 2. É exibida em seguida uma tela com informações sobre o assunto de trigonometria, bem como informações de seu surgimento; 3. O discente navega pelas telas de histórico através das opções continuar, voltar e sair.

Quadro 3 - Descrição do caso de uso “Visualizar Histórico”.

Caso de Uso: Manipular Vídeos de Trigonometria (UC03)
Objetivo: Assistir vídeos de trigonometria e ter opção de pausar, continuar e parar o vídeo.
Ator: Discente
Fluxo Principal: 1. Este Caso de uso se inicia quando o discente na tela de menu principal escolhe a opção de vídeos; 2. É exibida em seguida uma tela de canal com vídeos disponíveis de trigonometria; 3. O discente ao escolher um canal poderá assistir um vídeo; 4. O mesmo poderá em seguida assistir um vídeo todo e manipular as opções de pausar, parar e continuar; 5. Terminando de assistir um vídeo em um determinado canal o discente poderá escolher outros canais com mais vídeos disponíveis sobre outros assuntos.

Quadro 4 - Descrição do caso de uso “Manipular Vídeos de Trigonometria”.

Caso de Uso: Manipular Funções Trigonométricas e Gráficas (UC04)
Objetivo: Disponibilizar telas para analisar graficamente através do círculo trigonométrico as funções de Seno, Cosseno.
Ator: Discente
Fluxo Principal: 1. Este Caso de uso se inicia quando o discente na Tela de <i>menu</i> principal escolhe a opção de gráficos; 2. É exibida em seguida uma tela com números ordenados no padrão de controle remoto para TV Digital e as opções de estudo de caso para o ângulo, seno, cosseno correspondentes e análise gráfica de seu cálculo via círculo trigonométrico; 3. O discente navega sobre um número e o seleciona de acordo com o número correspondente do controle remoto; 4. Em seguida são mostradas as opções com o ângulo digitado, seno e cosseno calculados, bem com o resultado das funções seno e cosseno em um círculo trigonométrico; 5. O discente em seguida poderá apagar o ângulo e selecionar outros números para analisar outros ângulos através destas funções em um círculo trigonométrico.

Quadro 5 - Descrição do caso de uso “Manipular Funções Trigonométricas e Gráficas”.

Caso de Uso: Visualizar Desafios (UC05)
Objetivo: Disponibilizar telas de questões discursivas que estimulem o discente resolverem situações que apliquem o conceito de trigonometria.
Ator: Discente
Fluxo Principal: 1. Este Caso de uso se inicia quando o discente na Tela de <i>menu</i> principal escolhe a opção de desafio; 2. É exibida em seguida a primeira tela com questões discursivas; 3. O discente visualiza as questões discursivas disponíveis para resolver; 4. O discente pode escolher a opção enter para ir à próxima tela ou escolher as opções voltar para retornar à tela anterior ou sair para retornar ao menu principal; 5. Caso o discente escolha a opção enter sucessivamente navegará pelas telas de perguntas de até chegar ao final encontrando a tela com respostas das questões; 6. Em seguida o discente verifica as respostas das questões e escolhe entre as opções para navegar sobre as telas: voltar ou sair

Quadro 6 - Descrição dos casos de uso “Visualizar Desafios”.

Caso de Uso: Manipular jogo de perguntas e respostas (UC06)
Objetivo: Disponibilizar um jogo que permita adicionar um nível de perguntas e respostas associadas às questões objetivas a cada interação do jogo, tendo opção para verificar porcentagem de acertos e erros durante a análise de resultados.
Ator: Discente
Fluxo principal: 1. Este Caso de uso se inicia quando o discente na Tela de menu principal escolhe a opção de Jogo; 2. É exibida em seguida uma tela de jogo com um cenário para o discente passar por quatro questões; 3. O discente ao passar por uma questão na tela seguinte é disponibilizada perguntas objetivas; 4. O discente resolve e marca a opção e passa para próxima questão com a opção enter do controle remoto; 5. O discente acompanha o resultado e resolver as quatro questões; 6. Chega ao fim do nível; 7. Verifica o resultado e só é passa para o próximo nível o discente que alcançar no mínimo 75% de acerto do total de quatro questões do nível atual; 8. Caso o discente não passe de nível e queira jogar novamente o mesmo nível, ele terá que voltar ao início do jogo pela opção sair indicada como botão vermelho do controle remoto; 9. Alcançado o resultado de 75% de acerto o discente joga o nível seguinte com grau de dificuldade superior de questões, sendo um total de quatro níveis; 10. Ao término de todos os níveis é mostrada a porcentagem que o discente alcançou quando superados todos os níveis do jogo.

Quadro 7 - Descrição do caso de uso “Manipular Jogo de Perguntas e Respostas”.

3.6.3 Estudo de caso de aplicações criadas pelo *framework*

Foi modelado um exemplo de ambiente construído nesta dissertação através da simbologia do fluxograma da Figura 30, visando organizar o raciocínio lógico a ser seguido e indicar os passos durante a execução da ferramenta para apenas os OA *menu*, vídeos, gráficos e jogo. De acordo com a figura, uma típica aula que utiliza uma aplicação de TVD gerada pelo *framework* deste trabalho é composta dos seguintes OA: um *menu* com uma interface gráfica para a seleção de mídias (áudio, vídeos, gráficos e jogo) relacionadas ao conteúdo de sala de aula.

Uma vez escolhida uma opção, o aluno/professor terá a possibilidade de usar um recurso gráfico para reforçar o aprendizado sobre determinado tema. Como exemplo, um gráfico de uma função em um círculo trigonométrico ou um vídeo auxiliar explicativo.

Outra opção mostrada na figura é a escolha do jogo, onde o discente poderá testar os conhecimentos apresentados em um vídeo sobre um referente assunto, seguindo um *Quiz* de perguntas e respostas.

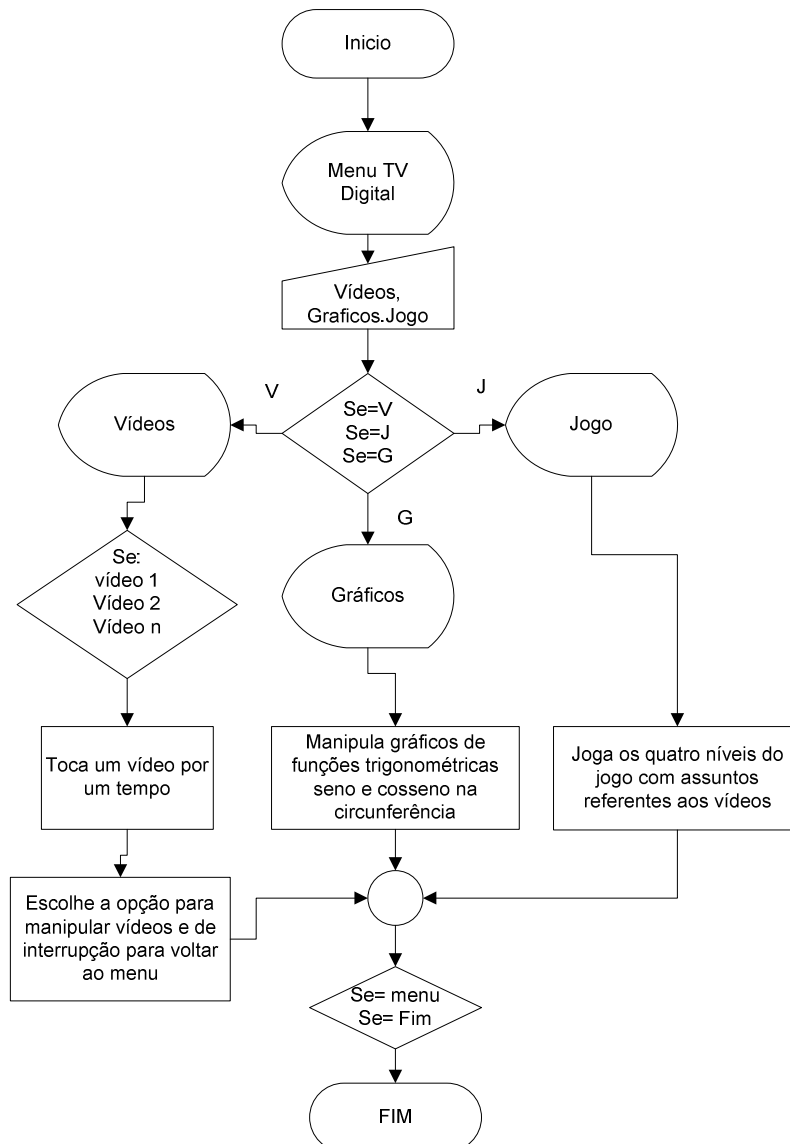


Figura 30- Fluxograma com execução dos OA em uma aula de trigonometria.

3.6.4 Diagrama de classes

Um diagrama de classe é uma estrutura estática do modelo de um projeto ou sistema, em que os elementos são representados por classes ou instância de classes (objetos) com sua estrutura interna e seus relacionamentos. A estrutura interna de uma classe é formada por atributos que especificam identidades de objetos e suas operações definidas como métodos. As relações (associações) indicam as conexões entre classes e

dentre outras relações encontram-se as classificadas em generalização, agregação e composição. No relacionamento de generalização é contextualizado o emprego da superclasse (mãe) e a subclasse (filha) que permite a reusabilidade de *softwares* através de herança entre mãe e filha. O relacionamento de agregação é usado para mostrar o fato de um todo (classe) ser composto por partes, e o todo sobrevive sem as partes e as partes sem o todo. O relacionamento de composição é um relacionamento de agregação mais forte em que a classe ou objeto parte pode pertencer somente a um todo. Normalmente qualquer remoção em um todo é considerada respectivamente a remoção das partes (LIMA, 2008).

Sobre esta exposição de conceitos neste trabalho é apresentado na Figura 31 um diagrama de arquitetura de classes simplificado, com intuito de exemplificar as principais classes implementadas através do uso do *framework*, bem como as classes que indicam sua composição.

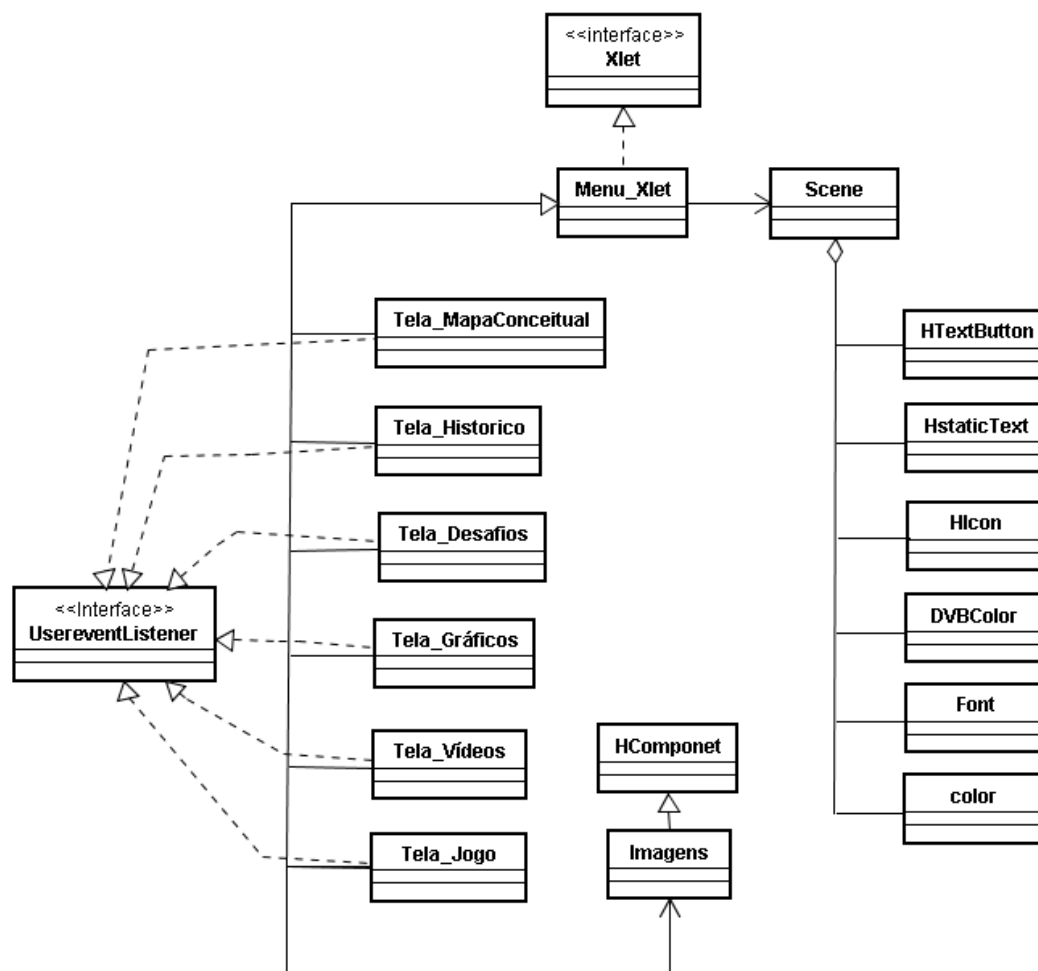


Figura 31 - Diagrama de arquitetura classe do *framework*.

Abaixo, segue a descrição dos relacionamentos e funcionalidade das classes apresentadas:

- A classe (interface) *Xlet* é uma entidade caixa preta que disponibiliza métodos abstratos para serem implementados por outras classes presentes na hierarquia. O desenvolvedor faz uso deste recurso para permitir manipular métodos (operações) para o controle do ciclo de vida das aplicações *Xlet* em ambiente de TVD;
- A Classe *MenuXlet* (mãe) é uma entidade caixa branca, que implementa os métodos da Interface *Xlet*. O desenvolvedor faz uso deste recurso para inicializar a aplicação e definir variáveis e objetos manipulados por todo ciclo de vida, bem como permitir repassar estas funcionalidades para classe filhas;
- A Classe *Scene* é uma entidade caixa branca responsável por permitir instanciar objetos (componentes gráficos) em uma determinada interface gráfica de TVD. O desenvolvedor faz uso deste recurso definido por agregação entre objetos para decidir e manipular os tipos de componentes que irão compor as interfaces gráficas de TVD em cada aplicação. Estes objetos são classificados em:
 - *HtextButton*, *HStaticText*, *Font*: objetos que permitem manipulação de textos;
 - *DVBColor* e *Color*: objetos que permite manipular a cor de fundo de tela e cores de textos; e
 - *HIcon*: objeto que permite indicar as posições das imagens que irão compor as telas de TVD.
- As subclasses (filhas) são entidades caixas brancas e fazem parte dos módulos de aplicações construídas pelo *framework*. O desenvolvedor utiliza esses recursos para estender as funcionalidades da classe mãe através de herança e implementar de forma diferente as funcionalidades existentes nas interfaces. Estas classes são: *Tela_MapaConceitual*, *Tela_Histórico*, *Tela_Desafios*, *Tela_Gráficos*, *Tela_Vídeos* e *Tela_Jogo*;
- A Classe Abstrata *HComponet* (mãe) é uma entidade caixa preta que disponibiliza a reutilização de métodos abstratos. O desenvolvedor utiliza

esse recurso para implementar o método abstrato que localiza a imagem em um diretório ou pasta;

- A classe Imagem é uma entidade caixa branca que estende a classe abstrata HComponet. O desenvolvedor utiliza esse recurso para carregar as imagens em cada módulo da aplicação desenvolvida pelo *framework* através do objeto HIcon; e
- A interface `UserEventListener` é uma classe de entidade caixa preta que determina os métodos abstratos para as manipulações de eventos do controle remoto em interfaces de TVD. O desenvolvedor utiliza esse recurso para implementar as operações desta interface nas classes dos módulos de aplicações.

3.6.5 Levantamento da arquitetura de execução do *framework*

O *framework O-Learning TVDi* é um ambiente que consiste em um conjunto de classes Java implementadas no padrão de *middleware* MHP. Essas classes disponibilizam estruturas para manipular OA como imagens, vídeos, gráficos em circunferência, jogos e questionários, sendo necessário o usuário apenas navegar sobre as interfaces no padrão de TVD e utilizar os recursos desta aplicação em uma determinada aula de trigonometria.

Este ambiente é preparado para ser executado tanto em ambiente emulado pelo computador como poderá ser estendido para um ambiente real de TVD com auxílio de um STB que permita executar esse tipo de aplicação. Apesar de testado todo ambiente no padrão de *middleware* MHP, essa arquitetura disponibiliza módulos para serem testados no padrão do *middleware* Ginga-J.

A arquitetura do ambiente de aprendizagem descrita é representada por dois blocos que ilustram como a mesma pode ser executada: o primeiro bloco consiste de classes Java que implementam um *Xlet* utilizando um conjunto de APIs do padrão MHP e emulado em um ambiente de computador. Já o segundo bloco apresenta o padrão real de TV Digital, com utilização de um STB para receber uma aplicação que possa ser utilizada em padrão de *middleware* MHP ou Ginga-J. Para execução do ambiente é necessário um sistema operacional com suporte para uma JVM executar aplicações Java sobre um *hardware* específico. Na Figura 32 é caracterizada a arquitetura deste *framework*.

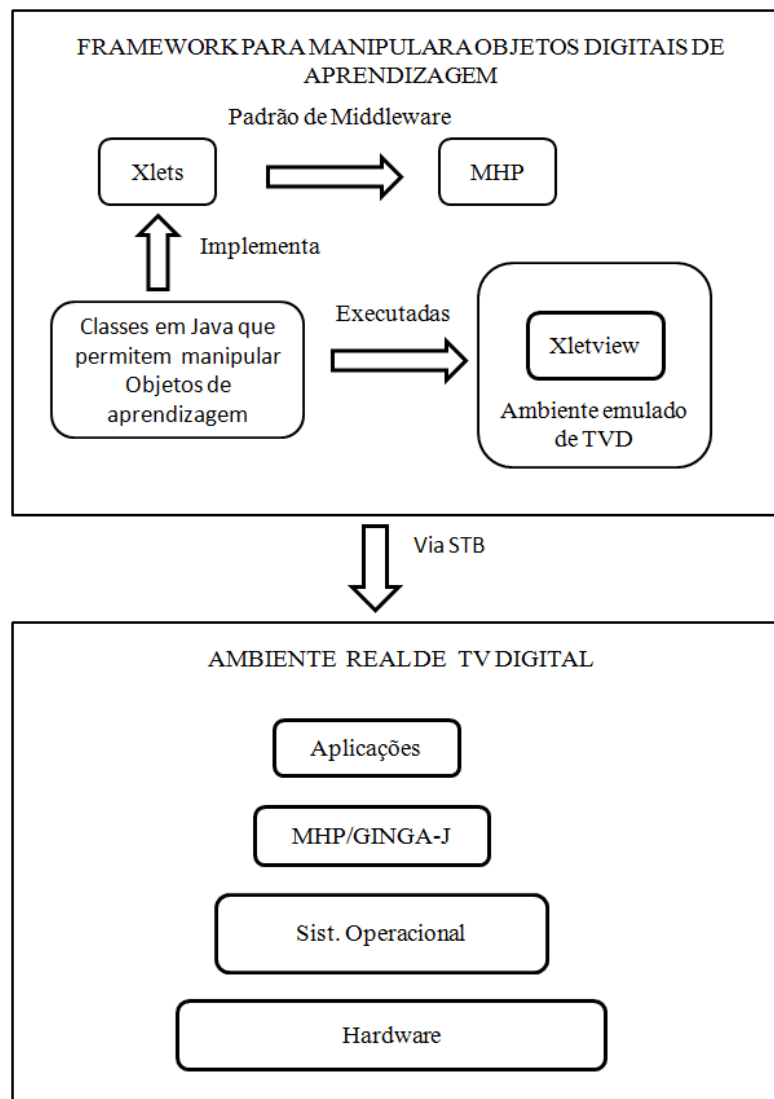


Figura 32 - Caracterização da arquitetura do *framework*.

3.7 DESENVOLVIMENTO E INTEGRAÇÃO

Como o *framework* deste trabalho foi desenvolvido para atender as especificações de um ambiente emulado para testes e um ambiente real de TVD. Nesta estrutura, a integração dos módulos de OA no ambiente emulado foi realizada de uma só vez, pois não se tem a restrição de controle de recursos de memória ainda existente em aplicações que são executadas nos STBs de TVD.

Tendo em vista esta restrição, a arquitetura deste *framework* foi projetada e desenvolvida para testar seus módulos de Histórico, Mapas Conceituais, Vídeos, Gráficos, Jogo e Desafios em partes, verificando quais destes módulos poderão ser adaptados em STBs dos padrões de *middlewares* MHP e GINGA-J. A outra possibilidade verificada para garantir o desenvolvimento da aplicação foi a de testar quais das APIs existentes em cada

um destes *middlewares* permitem construir as interfaces gráficas, explorando os recursos da orientação a objetos para garantir a reusabilidade dos componentes do *framework*.

3.8 VALIDAÇÃO DE SISTEMAS

Para atender as especificações de requisitos foram realizados dois modelos de testes com o objetivo de verificar estudos de casos para construir e utilizar a ferramenta. O primeiro modelo de teste foi sobre as tecnologias ou APIs dos padrões brasileiro e europeu descrito no Capítulo 5. O segundo modelo de teste também descrito neste capítulo foi a verificação da utilização da ferramenta como suporte às aulas de trigonometria para duas turmas de alunos de ensino médio e uma turma de alunos do ensino superior.

O objetivo desta atividade foi identificar qual a melhor adaptação deste *framework* para atender as necessidades dos discentes e docentes durante uma aula de trigonometria, investigando pontos positivos e críticos das funcionalidades desenvolvidas através do *framework*.

3.9 CONCLUSÃO

As seções que descrevem os requisitos para modelar a estrutura do *framework* desta dissertação resultam em um projeto de *software* que possa ser adaptado para padrões de TVD europeu e brasileiro, visando a reusabilidade de recursos tecnológicos descritos pelas APIs do capítulo anterior. Entretanto, empregar todos estes recursos é uma tarefa complexa, tanto para construir interfaces gráficas interativas como para oferecer manutenção e administrá-las. Neste contexto, é descrito no próximo capítulo o desenvolvimento do *framework* através do uso das ferramentas e APIs utilizadas.

4 DESENVOLVIMENTO DO *FRAMEWORK O-LEARNING TVDi*

O *framework* implementado neste trabalho tem como finalidade disponibilizar uma arquitetura que possa ser testada, melhorada e readaptada para pesquisas relacionadas ao desenvolvimento de aplicações para TVD, levando em consideração recursos disponíveis neste tipo de ambiente para criar e manipular interfaces gráficas, que possibilitam aos usuários interagir com uma determinada aplicação. Neste capítulo estão descritas as ferramentas de suporte utilizadas para desenvolver o ambiente e testá-lo, APIs do padrão MHP, que possam ser adaptadas para o padrão Ginga-J e os estudos de casos desenvolvidos.

4.1 FERRAMENTAS DE DESENVOLVIMENTO

Nas próximas subseções são apresentadas as ferramentas de desenvolvimento utilizadas para construir e testar o *framework*, destacando recursos para edição e execução de código fonte.

4.1.1 Sistema operacional

O *framework* foi desenvolvido em um PC de 64 bits, sistema operacional Windows 7 Home Premium, sendo utilizado um processador AMD Turion II, Dual Core Mobile M500 2.2 Ghz e 4 Gb de memória RAM.

4.1.2 Plataforma de desenvolvimento

O ambiente de desenvolvimento do *framework* foi a plataforma do Eclipse, que possibilitou a edição de códigos fontes utilizando a linguagem Java. Esta linguagem foi definida como linguagem padrão de desenvolvimento do *framework*, por ser orientada a objetos e permitir criar classes, heranças, polimorfismos, instanciar objetos e aplicar os recursos de reusabilidade necessários em cada módulo do *framework*.

Através desta linguagem foi possível interpretar os *Bytes Codes* do editor de código do Eclipse utilizando o *Kit* de Desenvolvimento Java (JDK). Segundo Campos (2011), o JDK constitui um conjunto de programas que engloba um compilador, interpretador e utilitários, fornecendo um pacote de ferramentas básicas para a construção de aplicações

Java. A plataforma Eclipse também ofereceu vantagens para adaptar *plugins* necessários para o desenvolvimento, testar e depurar erros de execução na linguagem Java com recursos de APIs correspondentes ao padrão de *middleware* europeu MHP e brasileiro Ginga.

4.1.3 Emulador Exletview

Devido a complexidade da pesquisa em investigar qual padrão de *middleware* para TVD fosse mais adequado para ambiente real de teste em um receptor digital, o uso de um emulador tornou possível realizar testes da arquitetura do *framework* em ambiente de PC mais próximo do ambiente de TVD. Neste contexto, após pesquisas realizadas sobre emuladores para simular um ambiente de TVD foi escolhido o Exletview. Este emulador possui o código aberto sob a licença GPL (*GNU Public Licence*), além de uma implementação de referência da API JavaTV, traz consigo implementações de outras APIs especificadas do padrão MHP, como a HAVI, DAVIC, além das bibliotecas do *PersonalJava* que o mesmo padrão faz uso (XLETVIEW, 2012).

Como este emulador possibilita programação em Java, suas aplicações podem ser executadas tanto em plataforma de sistema operacional Windows ou Linux, bastando para isso utilizar o JDK para compilar *Xlets* e executá-los no Xletview. Em relação ao *framework* pode verificar que apesar de utilizar o *middleware* MHP para testar as APIs como JavaTV que controlam aplicações em STB, as mesmas podem ser utilizadas também no padrão Ginga, como a API AWT para gerenciar componentes de interfaces gráficas. Para execução de vídeo e áudio é utilizada a API JMF 2.1.1 com incapacidade de execução de vídeos no padrão MPEG. Neste *framework* o padrão vídeo testado foi do formato AVI.

A outra possibilidade identificada no desenvolvimento de interface gráfica através da API HAVI neste ambiente é de tratar seus componentes com o suporte da API LWUIT, quando terminado o processo de validação das APIs no Ginga, visto até o momento como uma limitação para quem pretende desenvolver aplicações no padrão Ginga-J.

4.2 ARQUITETURA MODULARIZADA DO *FRAMEWORK*

A Figura 33 apresentada na IDE do Eclipse exibe a organização dos pacotes implementados do *framework*. O pacote principal corresponde ao início da aplicação é o do Menu.Xlet. Neste pacote constam as classes principais que são instanciadas e permitem

a comunicação com outras classes dos outros módulos, como exemplo é mostrada a classe instanciada TelaHistórico. Os módulos são identificados da seguinte forma: desafios, gráficos, histórico, os módulos de jogos para cada nível do jogo, módulo de mapa para implementação de navegação de mapa conceituais e por fim o módulo para vídeos.

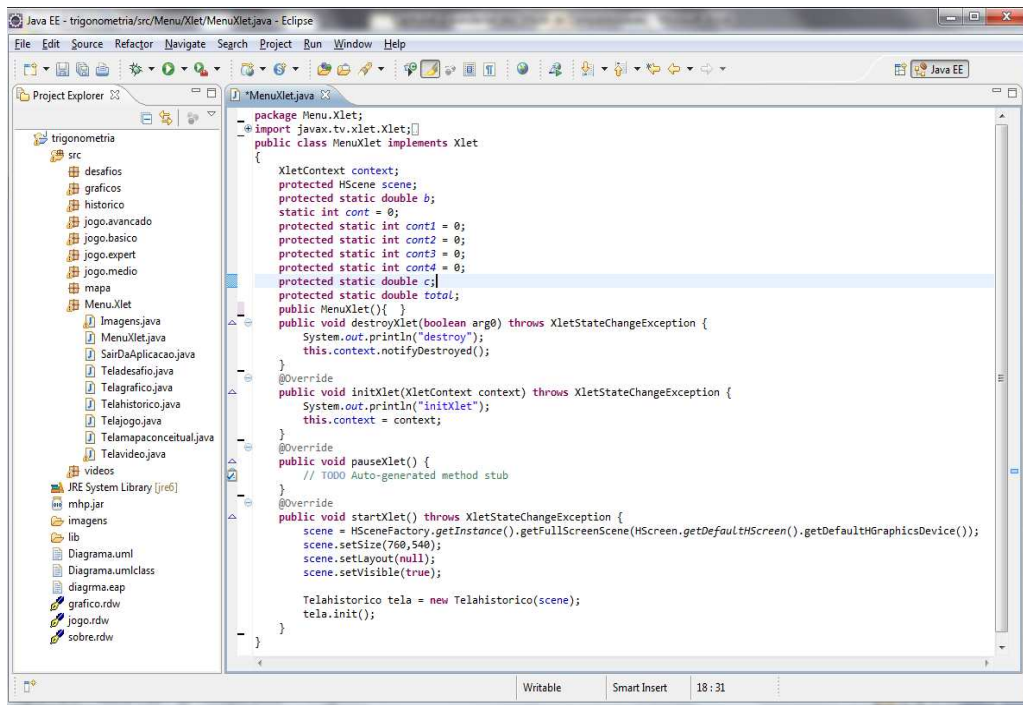


Figura 33 - Pacotes do *framework*.

4.3 CENÁRIO PRINCIPAL DO *FRAMEWORK*

Foram definidos seis (06) estudos de casos que são organizados de acordo com as características dos objetos digitais de aprendizagem apresentados na Figura 34 e descritos na próxima seção. O que se buscou foi organizar esses recursos para compor um cenário de suporte de TVDi para uma aula de trigonometria.

As interfaces implementadas em cada módulo estão dispostas para o professor já manipulá-las, não tendo como característica montar uma aula em tempo real de execução, preocupando-se apenas com a interação com os alunos, pois os docentes não precisariam organizar os componentes de OA em uma aula e sim escolher quais dos módulos do *framework* gostariam que fossem utilizados.

Figura 34 - Interface principal do *framework*.

4.4 ESTUDO DE CASOS IMPLEMENTADOS

Nas subseções seguintes estão as descrições de cada módulo desenvolvido através do *framework*, especificando as funcionalidades e formas de navegações em ambiente de TVDi para manipular OA.

4.4.1 História da trigonometria

A História da trigonometria é abordada por Moreira (2005) e está disposta na Figura 35. Nesta interface o docente pode, durante a aula, debater sobre o surgimento da trigonometria e ter a opção de manipular eventos através da tecla *enter* ou *ok* do controle remoto para ir à próxima interface de história. Foram definidas outras opções de navegação entre as páginas como Voltar e Sair. Para todas as interfaces do *framework*, Voltar significa voltar a interface anterior e Sair, voltar a interface principal.



Figura 35 - Interface principal de História da trigonometria.

A Figura 36 consta a segunda interface de história da trigonometria. Nesta interface o professor ao chegar ao término do assunto poderá acessar a interface anterior para realizar revisões ou sair deste módulo, voltando à interface principal do *framework*.

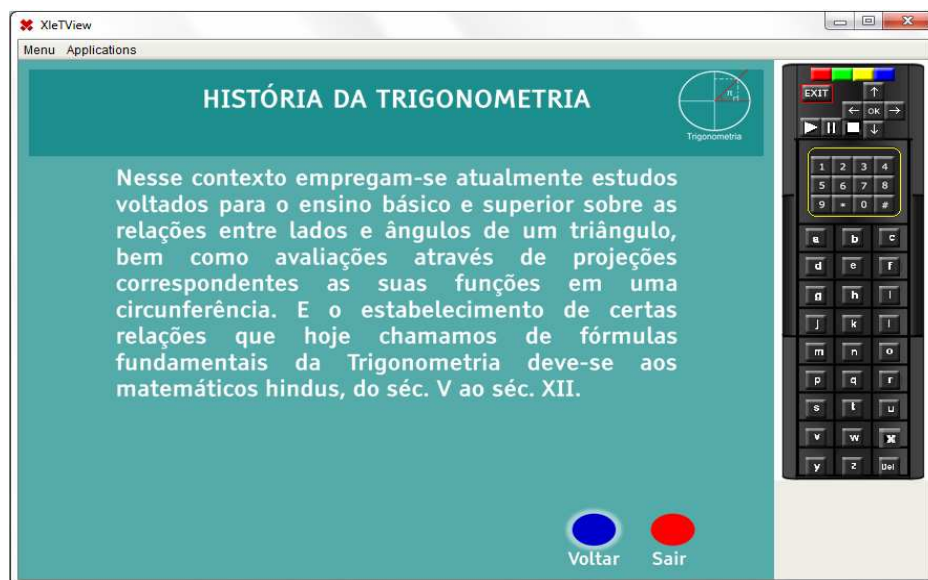


Figura 36 - Interface final de História da trigonometria.

4.4.2 Mapa conceitual

As interfaces de Mapas conceituais (Mcs) foram criadas para mostrar a organização dos assuntos em uma aula de trigonometria. Segundo Cabral e Giraffa (2002), os Mcs auxiliam na organização e estruturação do conteúdo, permitindo ao projetista (instrutor/professor) organizar o conhecimento observando os pré-requisitos, necessidades e objetivos dos alunos que trazem consigo experiências e aprendizagem prévias. A Figura 37 disponibiliza de forma geral os tópicos que podem ser explorados.

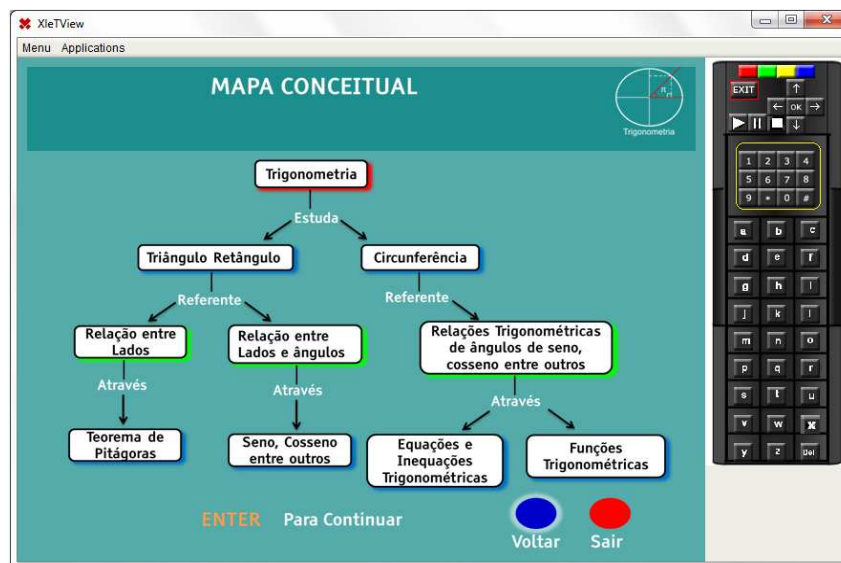


Figura 37 - Interface principal de Mapa conceitual.

Os alunos/professores por sua vez podem visualizar o assunto de forma geral e compreender como investigar seus conceitos e hierarquias. De posse desta estrutura eles têm a opção de decidir qual a melhor forma de inicializar seu estudo. Assim como nas interfaces de histórico os padrões de navegações para essas interfaces são os mesmos. Ao apertar a tecla *OK* ou *Enter* do controle remoto será mostrado outro Mc sobre o estudo específico do Teorema de Pitágoras, disponibilizado da Figura 38.

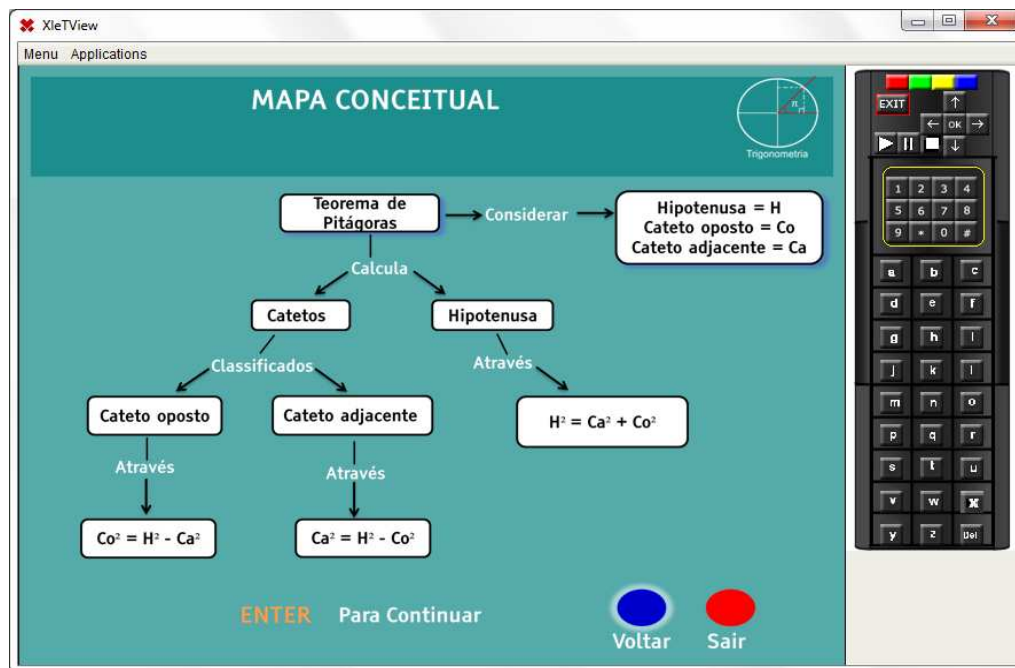


Figura 38 - Mapa conceitual de Teorema de Pitágoras.

Na Figura 39 o docente/discente, após apertar a opção *Enter*, visualiza a interface de Mc de classificação de funções trigonométricas. Além disso, é possível identificar que através das funções trigonométricas o mesmo poderá verificar as definições e avaliações de suas projeções em uma circunferência.

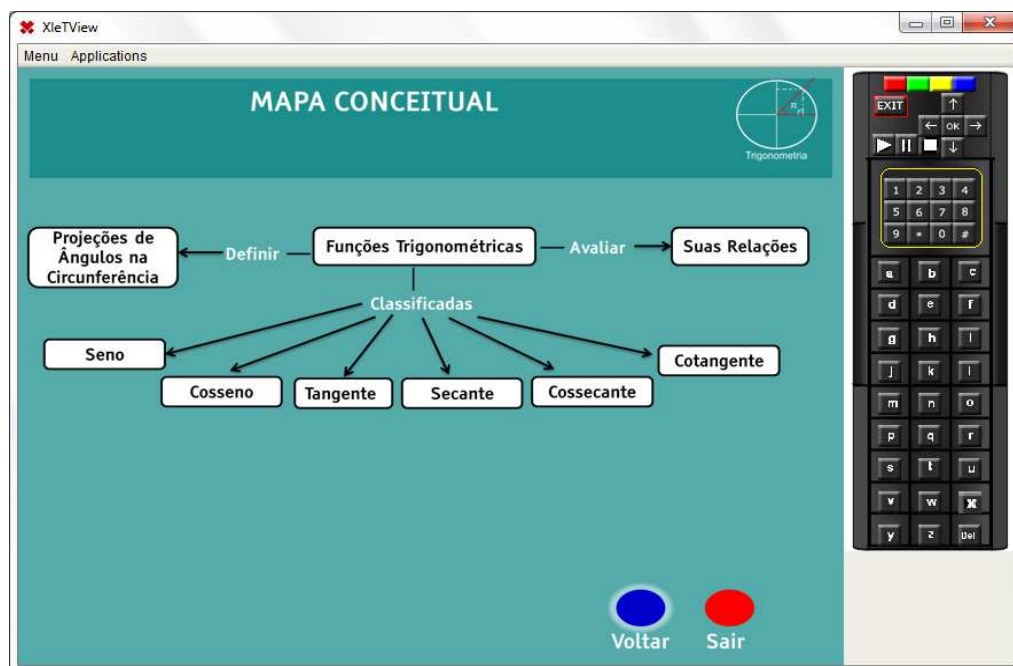


Figura 39 - Mapa conceitual de funções trigonométricas.

4.4.3 Vídeos

O módulo de vídeos desenvolvido oferece ao aluno a oportunidade de acessar um canal relacionado ao vídeo sobre um assunto da trigonometria (HOFMANN, 2012). A Figura 40 apresenta a interface com estudo de caso para três (03) canais.



Figura 40 - Tela principal de canais de vídeos.

O aluno pode navegar por cada um destes canais e, em cada canal, é apresentado um foco com o guia de execução do canal selecionado. Neste caso, se o aluno pressionar a opção 1 do controle remoto será disparado um evento para executar um vídeo do canal 01 e será criada um *Scene* da biblioteca HAVI para manipular a tela que executa o vídeo. A Figura 41 apresenta a interface de execução do canal um (01) com execução de um vídeo sobre funções trigonométricas.

O aluno poderá, através desta interface, executar um vídeo, pausar um vídeo ou parar e voltar à tela principal para escolher outro canal relacionado a outro assunto se assim preferir. Para isso foram definidas as opções um (1) para execução, quatro (4) para pausar e cinco (5) para parar. A proposta dessa definição foi utilizar recursos diferentes do controle remoto de TVD para garantir independência de funcionalidades e evitar conflitos de funções para os casos de testes em ambiente de STB.



Figura 41 - Tela de execução do canal 01.

Para interfaces de canal dois (02) e canal três (03) seguem o mesmo modelo de navegação de interface e execução de vídeos implementados no canal um (01).

4.4.4 Gráficos

Este estudo de caso seguiu a referência do aplicativo Trigonometria 1.1 conforme a Figura 42, como modelo visual de construção de gráficos para o ensino de trigonometria. Segundo Silva et al. (2000) este modelo foi criado para atender as necessidades dos alunos de ensino médio, onde o mesmo oferece a oportunidade de inserir os dados dos ângulos e verificar os valores do Seno, Cosseno, Tangente, Co-tangente, Secante e Co-secante.

O objetivo foi identificar os padrões de funcionalidade e organização visual de uma interface para servir de base para um protótipo no qual poderão ser manipuladas as funções trigonométricas e feitas adaptações aos padrões de interface gráfica para TVDi.

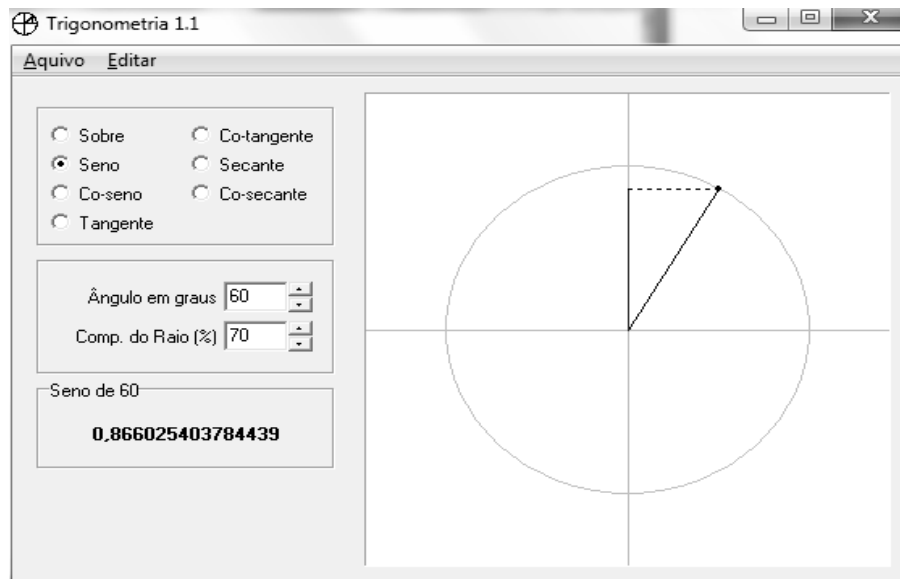


Figura 42 - Funções trigonométricas com o *software* Trigonometria 1.1.
Fonte: Adaptado de Silva et al. (2000)

Após a verificação desta viabilidade foi desenvolvida a interface de Círculo trigonométrico para TVD exposta na Figura 43, que possibilita a manipulação de ângulos referentes ao estudo de caso para as funções trigonométricas de Seno e Cosseno. O usuário pode navegar pelo padrão de teclas do controle remoto de TVD e inserir um ângulo e verificar sua projeção gráfica em um círculo trigonométrico. Ao clicar no botão Limpar as projeções são retiradas do círculo trigonométrico e o usuário poderá inserir outros ângulos, calcular novos valores e projeções das funções Seno e Cosseno.

O botão Iniciar está presente nesta interface para permitir que ao selecionar um valor numérico bastar apertar esta opção que a mesma também possibilitará inserir projeções no círculo e calcular valores para as referidas funções. Ao pressionar o botão Sair, o usuário voltará à tela principal do *framework* e poderá escolher outros módulos. O botão Voltar presente segue o padrão de navegação para os casos que estiverem mais de uma tela para manipular outras funções trigonométricas com ângulos e suas projeções gráficas. Observa-se nesta interface que as cores marrom, cinza, verde e vermelho correspondem respectivamente às seguintes projeções:

- Marrom indica a interseção do ângulo entre as funções de seno e cosseno;
- Verde indica a projeção de seno de acordo com seu valor assumido pelo usuário;
- Vermelho indica a projeção do cosseno conforme seu valor assumido pelo usuário, comparando mesmo valor assumido pela função seno; e
- Cinza indica a projeção máxima do ângulo no círculo trigonométrico.

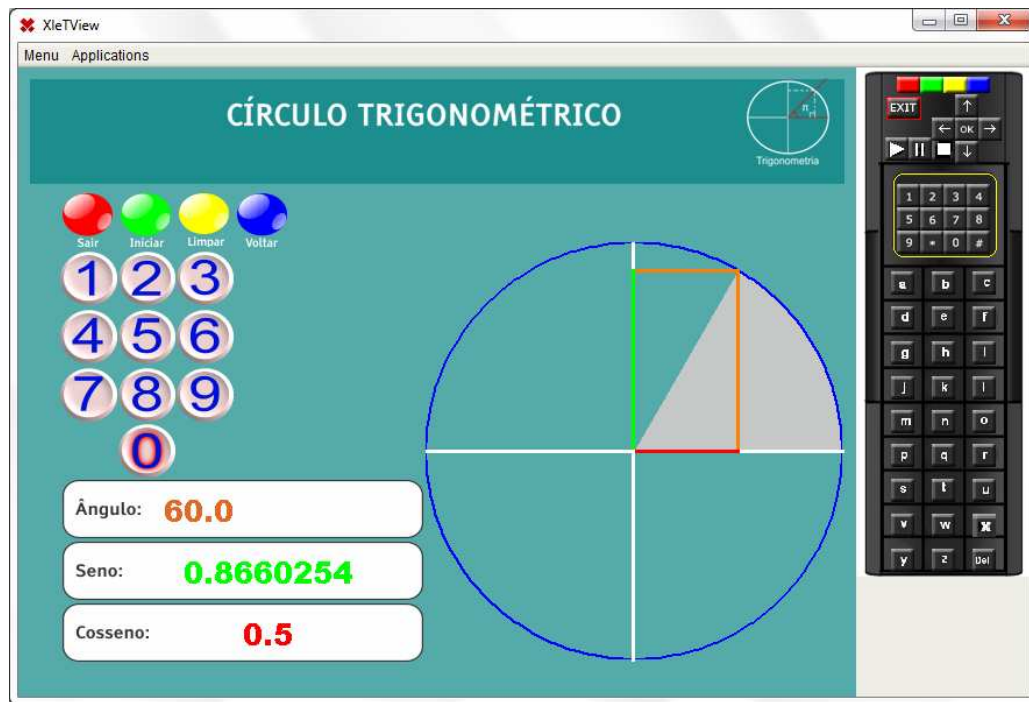


Figura 43 - Círculo trigonométrico.

4.4.5 Jogo de Quiz para trigonometria

Para o desenvolvimento do jogo de perguntas e respostas foi criado um cenário onde o jogador para responder as perguntas no modelo de *Quiz* tem que percorrê-lo. Sua elaboração foi definida sobre o conceito de jogos para CRPG - *Computer-Role-Playing-Games*. Segundo Silva (2006), o CRPG consiste de um tipo de jogo que utiliza o conceito do *Role-Playing-Game* conhecido como RPG de mesa, adaptado para o suporte digital. Nesse tipo de jogo, o jogador assume uma personagem, enfrenta desafios e ganha experiência baseada, normalmente, em enredo de fantasia, envolvendo tramas e cenários. Envolve, principalmente, o raciocínio lógico para solução dos problemas.

O jogo deste estudo de caso é composto pelos níveis básico, médio, avançado e *Expert* contendo quatro (04) perguntas. A Figura 44 apresenta apenas o nível *Expert* e o jogo segue a idéia de pirâmides onde o jogador ao iniciar um percurso tem que responder as perguntas localizadas nas pirâmides até chegar ao final.



Figura 44 - Interface do nível *Expert*.

Para navegar entre os itens das questões foi utilizada a técnica da animação limitada ou planejada por Bill Hanna e Joe Barbera (dos desenhos animados Hanna-Barbera). De acordo com Blair (1994), esta técnica divide um personagem animado em diferentes partes separadas em camadas que podem ou não serem animadas de forma independente uma das outras. Permite assim a criação de diversas possibilidades através da combinação entre os elementos e o reaproveitamento destas partes em diferentes cenas. A Figura 45 exibe a imagem com várias projeções que correspondem à possibilidade de reutilização das posições da imagem em diferentes cenas.

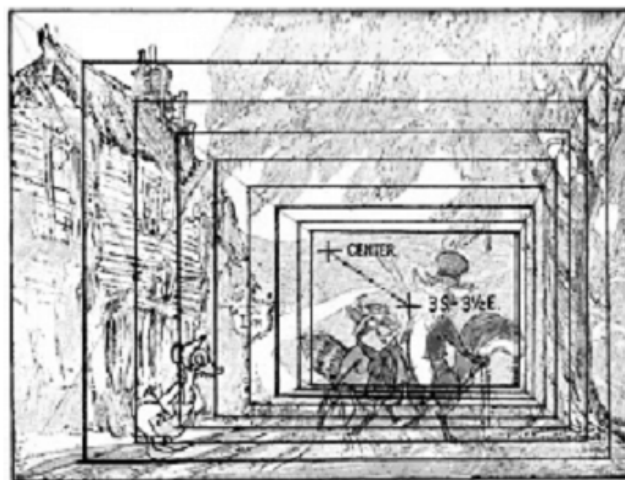


Figura 45 - Técnica de animação limitada.
Fonte: Adaptado de Picanço et al. (2011).

Esta metodologia é empregada para carregar as interfaces que indicam as posições das respostas utilizando a classe *Imagens* com o método *carregarImagem* passando como parâmetro o caminho onde se encontra a imagem. O procedimento acontece da seguinte forma: a classe *Tela4_Pergunta1* é instanciada para manipular os itens das perguntas através do método *userEventReceived* pertencente a interface *UserEventListener*, conforme a Figura 46. Este método permite instanciar a classe *Tela Final* contendo o método construtor *public TelaFinal (HScene scene, int opção)*, para referenciar a *Scene* utilizada e alternar entre as telas de opção de resposta da pergunta.

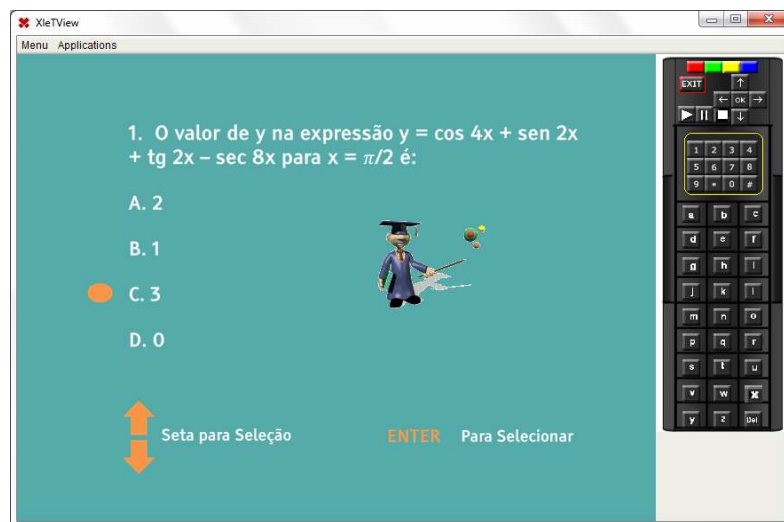


Figura 46 - Interface de perguntas um (01) do nível *Expert*.

O método o tipo *void init* instancia o tipo *Hicon* da API *HAVI* que através do objeto *imagens* carrega a interface com resposta certa. A Figura 47 exibe a resposta da questão um (01).

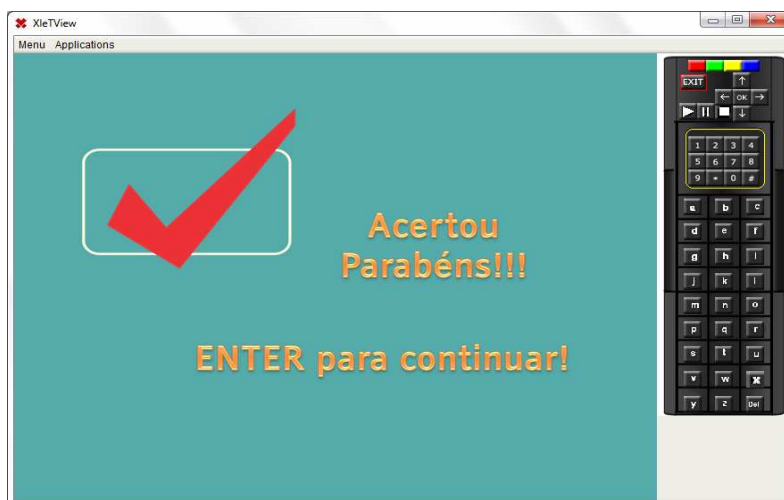


Figura 47 - Interface de respostas da perguntas um (01) do nível *Expert*.

Este procedimento é repetido para interface de dois (02), três (03) e quatro (04) de perguntas deste nível conforme a Figura 48 que exhibe a interface de pergunta dois (02).

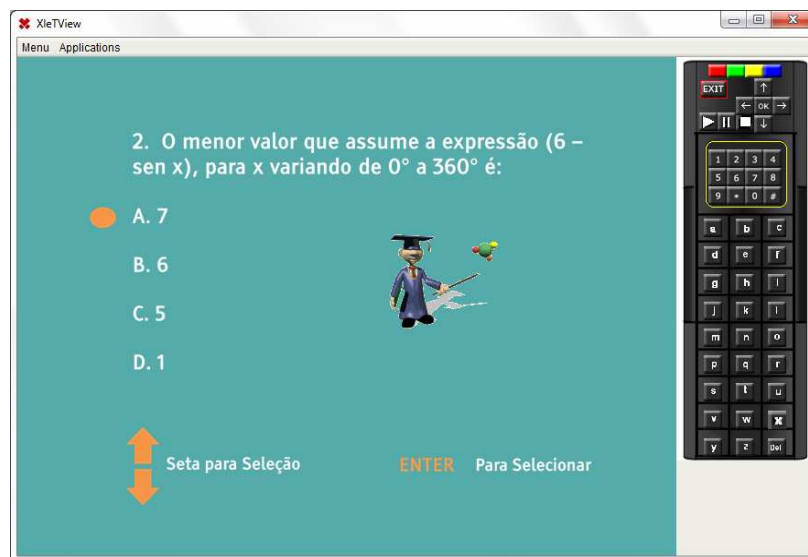


Figura 48 - Interface de pergunta dois (02) do nível *Expert*.

A Figura 49 apresenta a opção que depois de respondida a pergunta dois (02) é apresentada a tela de erro de uma questão. No caso em que o jogador ao errar poderá continuar ao pressionar *Enter*, pois no total de quatro (04) questões é permitido errar somente uma questão e poderá através da opção Sair do controle remoto ir para o início do jogo e começar novamente.



Figura 49 - Interface de representação de erro do nível *Expert*.

Em seguida ao pressionar *Enter* é exibida na Figura 50, a questão três (03) deste nível para o jogador responder e avaliar se acertou ou errou.



Figura 50 - Interface de pergunta três (03) do nível Expert.

Ao responder a questão três (03) e ter acertado esta questão é apresentada a questão quatro (04) na Figura 51.

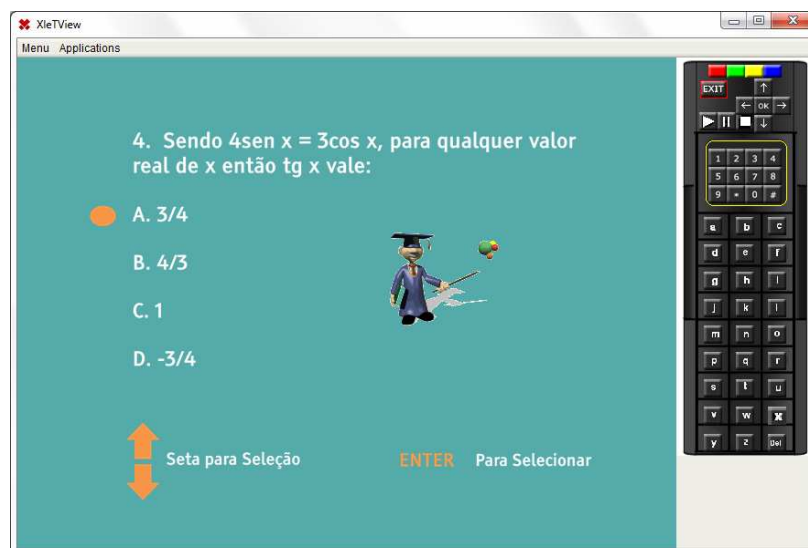


Figura 51 - Interface de pergunta quatro (04) do nível Expert.

E respondida a questão, a Figura 52 exibe os acertos e erros neste nível e o jogador pode pressionar para Sair do nível presente ou continuar ao pressionar a opção *Enter*.



Figura 52 - Interface de erros e acertos do nível *Expert*.

Caso o jogador escolha a opção *Enter* é apresentada a interface de Fim de jogo na Figura 53 informando a porcentagem total de acertos correspondentes aos quatros níveis, pois para jogar este nível tem que ser percorrido os níveis anteriores.



Figura 53 - Interface de total de acertos do jogo.

4.4.6 Desafios

As interfaces de Desafios disponibilizam questões discursivas obtidas de Moreira (2005) sobre a aplicabilidade da trigonometria. A Figura 54 expõe a primeira interface com 02 (duas) questões.

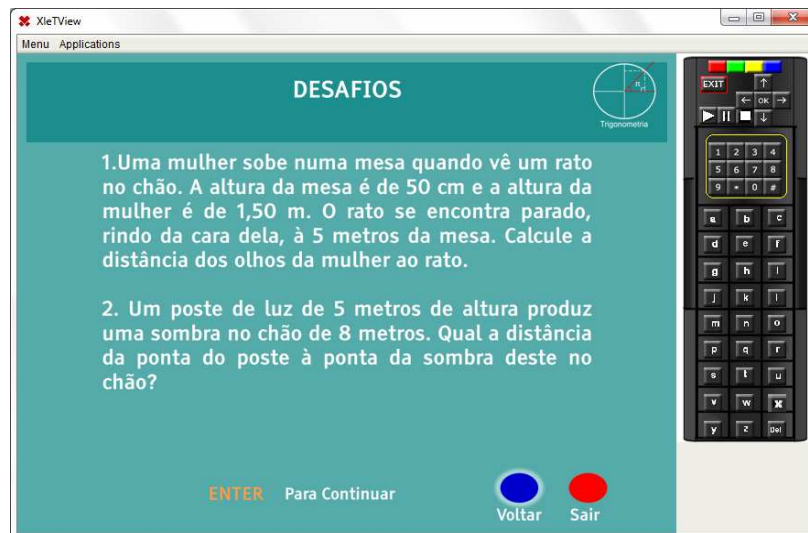


Figura 54 - Primeira interface de Desafios.

Na Figura 55 é apresentada a segunda interface com mais 02 (duas) questões seguindo também o mesmo padrão de navegação das interfaces dos módulos anteriores.

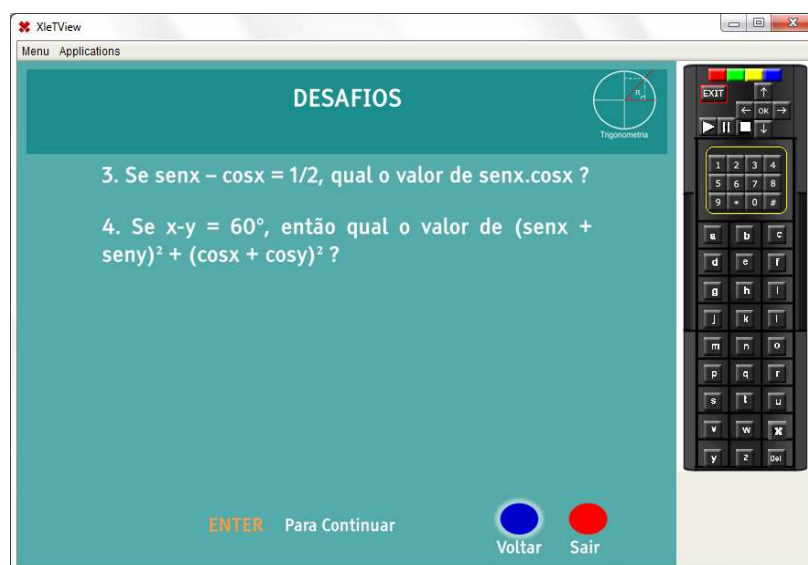


Figura 55 - Segunda interface de Desafios.

A Figura 56 exibe a última questão definida. O objetivo através deste cenário foi apresentar as situações onde o discente possa exercitar e desenvolver a sua própria solução.



Figura 56 - Terceira interface de Desafios.

Ao chegar ao termino de questões discursivas a Figura 57 disponibiliza os resultados para a possível conferência para os alunos. Um aluno específico pode conferir seus resultados e optar por navegar pelas interfaces das questões anteriores ou voltar à interface principal e escolher outros módulos através das opções Voltar e Sair que possibilitam acionar eventos do controle remoto da TVD.



Figura 57 - Interface de respostas dos Desafios.

4.5 CONCLUSÃO

Neste capítulo foram apresentadas as ferramentas utilizadas para o desenvolvimento do *framework* deste trabalho e os módulos de estudos de casos construídos a partir dele. Estes estudos de casos definiram um cenário para manipulação de OA em ambiente real de TVDi. As funcionalidades de manipulação de interfaces gráficas pelo controle remoto foram reaproveitadas em cada módulo desenvolvido, o que facilitou a organização e a produção mais rápida destes módulos e tratamento de erros, pois cada módulo foi tratado de forma independente.

No próximo capítulo serão apresentadas as avaliações destes módulos desenvolvidos com objetivo de identificar as dificuldades para o desenvolvimento na plataforma de *middleware* Ginga, além de verificar pontos positivos através das funcionalidades oferecidas na plataforma de *middleware* MHP, e suas limitações que precisam ser melhoradas ou resolvidas.

5 AVALIAÇÕES DO *FRAMEWORK O-LEARNING TVDi*

As avaliações foram realizadas desde o início do desenvolvimento do *framework* dividindo-se em quatro (04) etapas de testes avaliativos. Na primeira etapa realizaram-se testes avaliativos na plataforma de *middleware* Ginga. Foi avaliado o módulo Ginga-NCL para manipulação de aplicações declarativas e linguagem Lua para manipulação de aplicações procedurais, além das tentativas iniciais com Ginga-J.

Na segunda etapa foi permitido definir a estrutura deste *framework*, utilizando o padrão de *middleware* MHP. Na terceira etapa foi realizado um teste avaliativo em um dos módulos do *framework* para execução em um ambiente real de TVD. E na quarta etapa foram realizados testes avaliativos sobre a opinião do uso da ferramenta em duas (02) turmas do ensino médio na Escola Estadual prof^a Jacira Caboclo do Centro de Educação de Jovens e Adultos e uma (01) turma do ensino superior do Centro Universitário do Norte - Uninorte-Laureate, ambos na cidade de Manaus-AM. Os próximos itens descrevem os testes avaliativos da primeira, terceira e quarta etapa, visto que os testes avaliativos da segunda etapa correspondem ao desenvolvimento da base estrutural deste *framework* já descritos no capítulo 4.

5.1 AMBIENTE GINGA

Os testes avaliativos realizados na plataforma Ginga-NCL foram executados no ambiente de máquina virtual VMWare. Para Mattos (2008), o VMWare é uma infraestrutura que fornece *softwares* para virtualização desde ambientes desktop até ambientes de *data centers*. Este autor afirma que o VMWare é como se fosse um programa no espaço da aplicação, dentro de um sistema operacional hospedeiro que oferece as seguintes categorias de produtos: VMWare *Workstation*, *Server*, *Fusion* e *Player*.

A Figura 58 mostra o ambiente empregado para o teste da máquina virtual VMware *Player* onde a conexão com ambiente de virtualização é realizada através do número IP: 192.200.128. Este ambiente mostra também qual o procedimento para executar a aplicação, informando usuário (*root*) e senha (*password*), além do modelo de diretório que deve estar localizada a aplicação. A outra informação obtida foi a manipulação de teclas de controle remoto necessárias para executar a aplicação. Neste teste foi utilizada a tecla *Enter* (OK) do controle remoto para acionar o ambiente interativo da aplicação. A opção F1 para acionar o botão vermelho, F2 para eventos do botão verde, F3 para disparar

eventos do botão amarelo e F4 para tratar eventos do botão azul. Foram utilizadas as teclas F5 para voltar ao *menu* (tecla azul clara) e F6 (tecla verde clara) como tecla que pode parar a execução de um vídeo por exemplo.

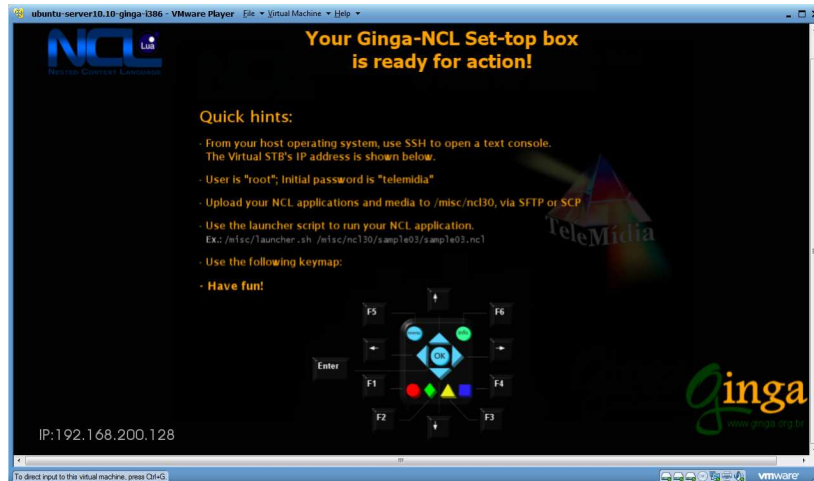


Figura 58 - Ambiente de máquina virtual VMware Player.

A conexão com a máquina virtual ocorreu via ambiente de edição de código fonte do Eclipse. Este ambiente foi configurado para editar código de apresentação NCL. A Figura 59 exibe a primeira interface inicial após a conexão com a máquina virtual. Esta interface informa ao usuário que ele está em um ambiente interativo e para acessar a interface principal da aplicação basta pressionar *Enter* (Ok) do controle remoto.

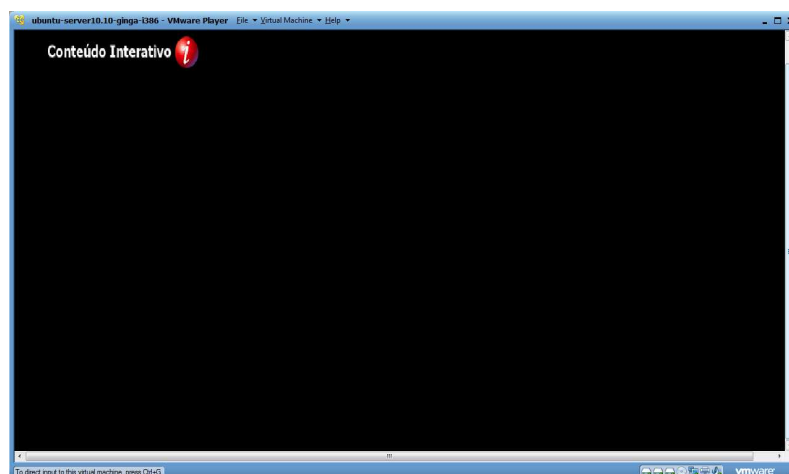


Figura 59 - Interface de ambiente interativo.

Em seguida, é exibida a interface de *menu* na Figura 60, para acessar OA através de vídeos, gráficos e Jogos como estudo de caso. Tendo ainda a opção *Exit* correspondente

a tecla F1 do controle remoto. Neste teste o padrão de interface do *framework* ainda não estava definido, mas possibilitava verificar os recursos possíveis através do Ginga-NCL.

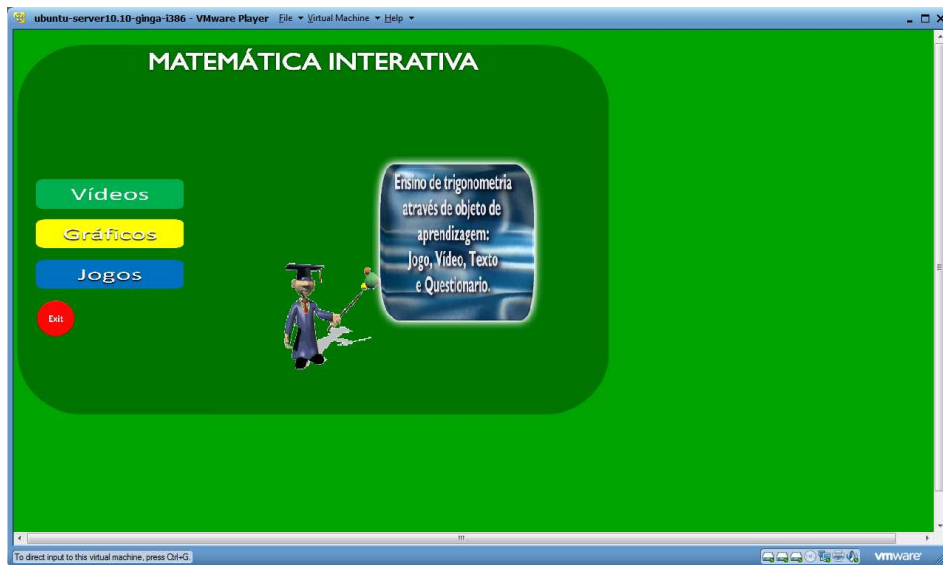


Figura 60 - Interface de *menu* na plataforma Ginga-NCL.

Neste estudo de caso só foram testados os módulos de vídeo e gráficos. Ao pressionar F2 é exibida a interface de *menu* de vídeos conforme a Figura 61.

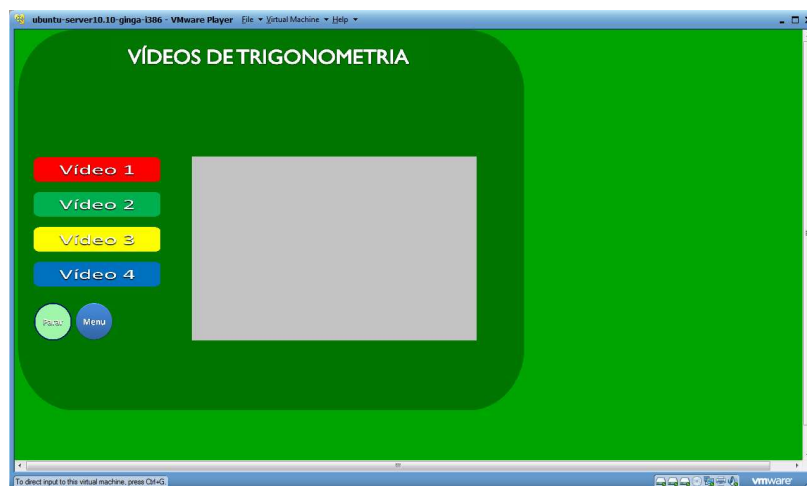


Figura 61 - Interface de *menu* de vídeos na plataforma Ginga-NCL.

Em seguida o usuário pode navegar entre as execuções de vídeos. No caso, ao pressionar a tecla F1 é executado o vídeo sobre triângulo retângulo conforme a Figura 62. Ao pressionar a tecla F2 é exibido outro vídeo e assim por diante ao pressionar F3 e F4. O

usuário pode ainda parar um vídeo ao pressionar F6 e voltar ao *menu* principal ao pressionar F5.

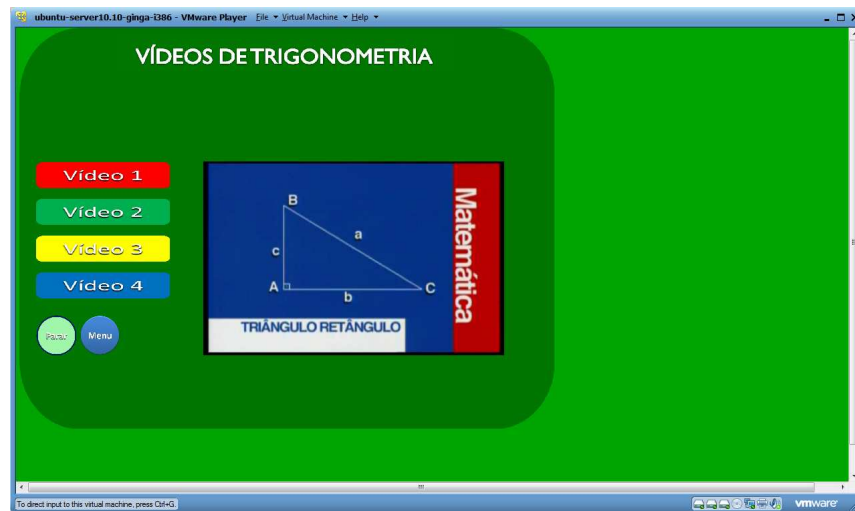


Figura 62 - Interface de execução de vídeos na plataforma Ginga-NCL.

Após voltar ao *menu* principal o usuário pode acessar a opção de gráficos através do pressionamento da tecla F3 e será apresentada a interface de Círculo trigonométrico conforme a Figura 63. Esta interface foi projetada para manipular funções trigonométricas correspondentes ao Seno e Cosseno.

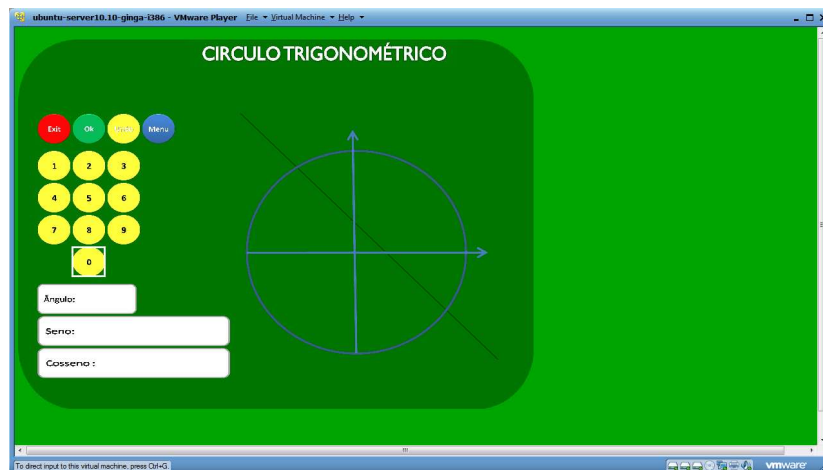


Figura 63 - Interface de Círculo trigonométrico na plataforma Ginga-NCL.

Neste estudo de caso foram realizados testes avaliativos utilizando a linguagem NCL para apresentação e *Script* Lua como tentativa de manipular funções matemáticas e realizar projeções no círculo trigonométrico. A maior dificuldade avaliada nesses testes foram as poucas possibilidades de reusabilidade para implementar funções matemáticas e

disponibilizar uma estrutura de *software* para arquitetura do *framework*. Houve uma tentativa inicial de integrar a plataforma Ginga-NCL e Ginga-J. Mas, nas pesquisas realizadas nas diversas fontes de literatura com referência para o Ginga foram encontradas muita documentação que ainda se encontra em fase de validação, principalmente para desenvolver interfaces gráficas através da API LWUIT. Neste contexto optou-se por desenvolver primeiramente a estrutura do *framework* na Plataforma MHP. O intento foi disponibilizar uma estrutura de interfaces gráficas construídas com o suporte da API HAVI que pudesse ser adaptada para os padrões da API LWUIT, à medida que os recursos estiverem disponíveis para atender as especificações definidas em sua documentação.

Os testes avaliativos realizados através do *middleware* MHP apresentaram resultados mais satisfatórios. Isto ocorreu pelo fato de já possuir APIs padronizadas há mais tempo que possibilitou desenvolver a arquitetura do *framework* de forma mais rápida. Foram elaboradas classes para manipular imagens da história da trigonometria, de mapas conceituais e desafios de questionários. Foram criados os módulos para manipular vídeos, um jogo educativo para trigonometria, e o módulo de gráficos contendo classes que permitissem manipular funções trigonométricas e visualizar projeções em um círculo trigonométrico. Apesar destes recursos para construir interfaces gráficas foram realizadas várias adaptações de interfaces gráficas para atender critérios de navegações e usabilidade em ambiente de TVDi, mesmo assim podendo ainda não ser o projeto de interface definitivo, pois as atualizações de modelos de interfaces para TVDi continuam ocorrendo de forma constante.

5.2 AMBIENTE REAL DE TVDi

Nas próximas subseções serão apresentadas a descrição do ambiente físico e a apresentação visual de como são executadas as aplicações geradas pelo *framework*.

5.2.1 Estrutura física de comunicação dos equipamentos

Para este teste avaliativo foram utilizados os seguintes equipamentos apresentados seguir (ver Figura 64) com objetivo primeiramente de identificar como foi organizada a estrutura e a comunicação física destes equipamentos em um ambiente real de TVDi:

1. Computador Pessoal (Servidor) com as seguintes configurações:

- Processador Intel Pentium Dual-Core 3,40 GHZ;

- HD de 45,8 GB;
 - Memória RAM de 2 GB; e
 - Sistema Operacional Ubuntu 7.10.
2. Televisor LG;
 3. Switch gerenciador 3com – modelo 2024;
 4. STB do fabricante ADB com conector RJ-45;
 5. Cabo de Rede par trançado categoria 5E no padrão EIA/TIA 568-B (Cabo UTP -5E);
 6. Ponto Lógico de rede; e
 7. Cabo HDMI.

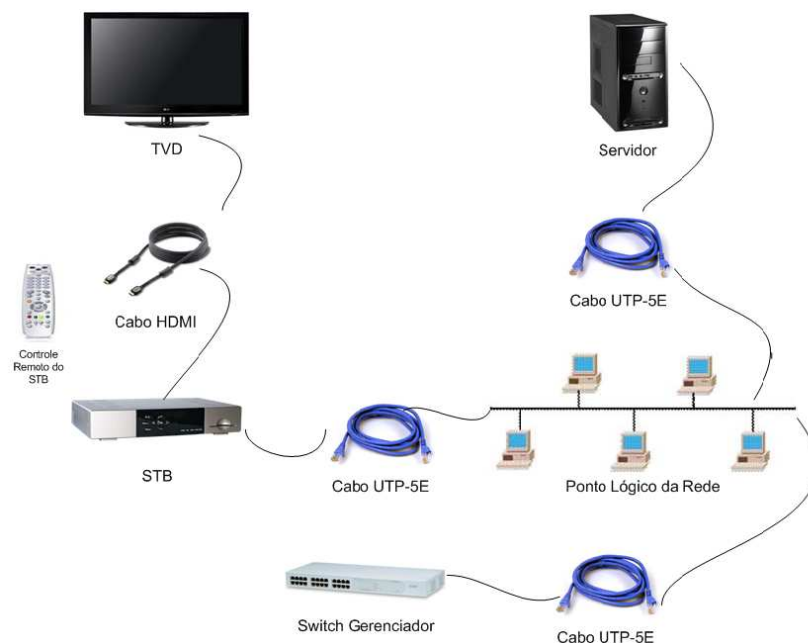


Figura 64 - Estrutura física de comunicação dos equipamentos.

5.2.2 Execução do módulo do *framework* em ambiente de TVDi

Na Figura 65 consta como foi realizado a execução da aplicação via TVDi. Nesta foi utilizada uma TVD para exibição da execução de um módulo do *framework* e um STB provido do *middleware* MHP para execução de aplicações interativas. Através deste STB é possível executar os *Xlets* com a utilização do canal de retorno utilizado em uma Rede TCP/IP (IP: 10.210.10.111) permanente e local. A conexão e o processo de execução restante são apresentados a seguir:

- Na mesma rede TCP/IP (IP: 10.210.10.225) está um computador com servidor de aplicações Apache versão 2.2.4, encontrado no site www.apache.org, que disponibiliza as aplicações para serem executadas através dos arquivos com extensão .ait, com informações necessárias providas pelo fabricante do referido STB;
- Em seguida o usuário requisita do servidor de aplicações via STB as aplicações que correspondem às classes *Xlets* com extensão *.Class* do *framework*; e
- O usuário interage com a TVD e o STB via controle remoto e executa a aplicação gerada pelo *framework*.

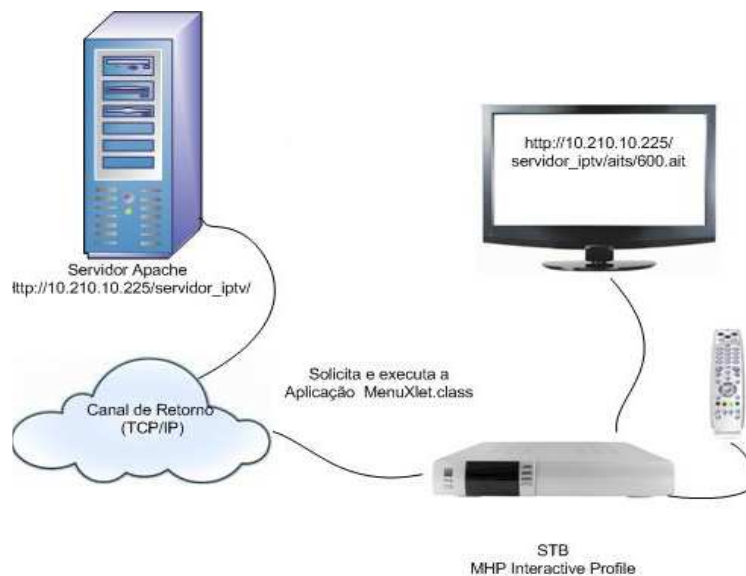


Figura 65 - Execução da aplicação via TVDi.

A seguir são mostradas as fotografias tiradas durante o teste do módulo de Mapa conceitual do *framework*. Na Figura 66 é apresentada a chamada deste módulo via interface de TVD pelo STB.

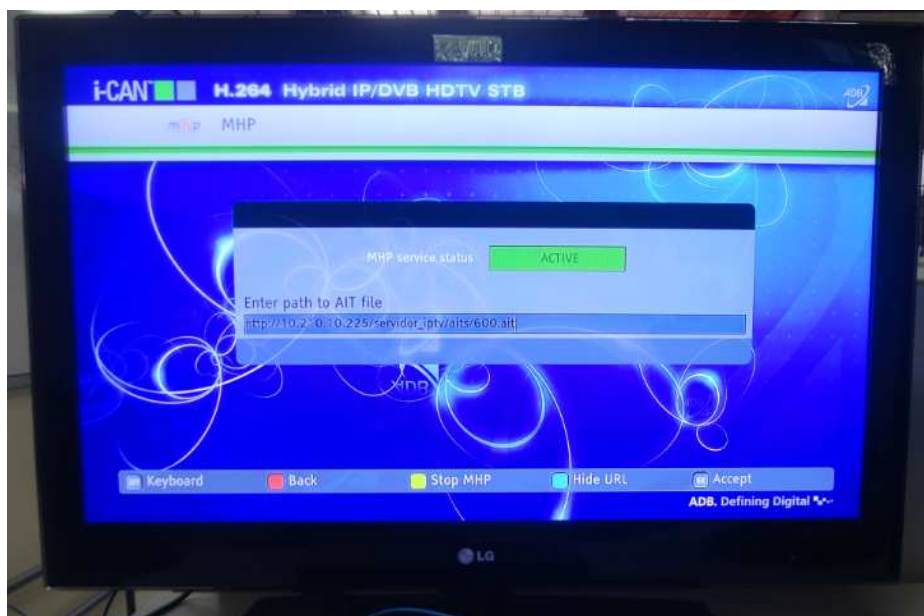


Figura 66 - Interface de chamada do módulo de Mapa conceitual.

Após a requisição de chamada ao servidor de aplicação será disponibilizada a interface principal para manipulação dos módulos do *framework* com a seleção do módulo de Mc e apresentação do Mc geral na TVD, conforme a Figura 67.

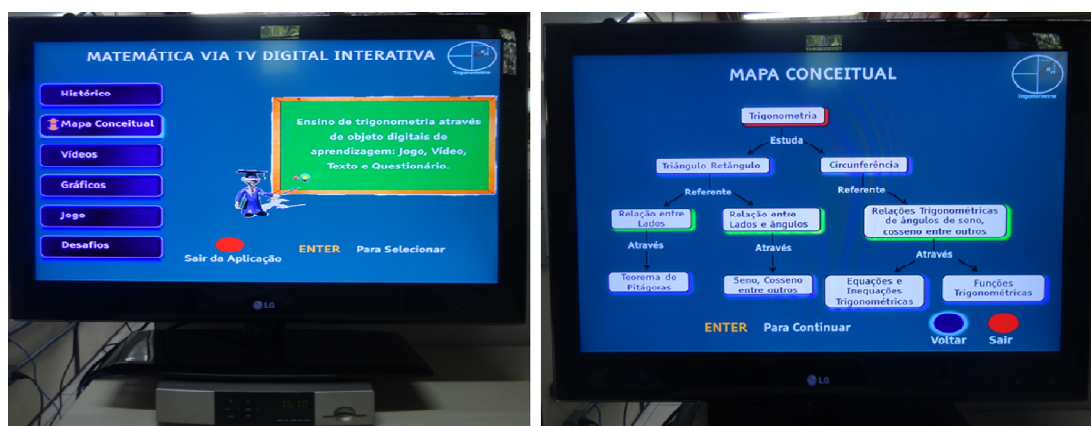


Figura 67- A) Apresentação da Interface Principal; B) Acesso ao Mc geral.

Na Figura 68 é apresentada as outras 02(duas) interfaces de módulo do Mc implementadas e já definidas suas navegações no Capítulo 4 (Seção 4.4.2).

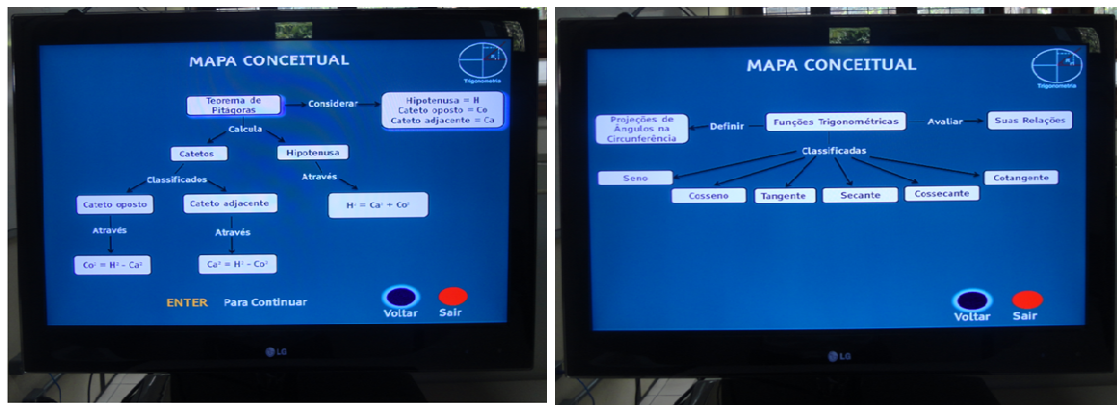


Figura 68 - A) Interface do Mc de Teorema de Pitágoras; B) Interface do Mc de Funções Trigonômétricas.

5.3 AVALIAÇÕES EM AMBIENTE REAL DE APRENDIZAGEM

Visando verificar a eficácia deste *framework* e melhorias futuras, foram realizadas três (03) avaliações de opiniões sobre o ambiente de sala de aula, baseadas nas análises qualitativas de Pereira (2001). Nessas avaliações foram abordadas as seguintes assertivas:

- A ferramenta possibilita a prática de conceitos aplicados em sala de aula;
- A ferramenta oferece fácil navegação e operacionalidade;
- O uso de recursos de OA pela ferramenta facilita a organização e melhor compreensão de conteúdos em uma aula de trigonometria;
- A ferramenta é útil em instituições de Ensino Médio e Superior;
- A ferramenta oferece variedade de suporte de conteúdo durante uma aula de trigonometria; e
- O uso desta ferramenta em um ambiente de TV Digital facilitaria a prática de estudo sobre a trigonometria.

Essas avaliações foram aplicadas logo após o término das aulas com seis (06) assertivas de múltipla escolha, com as opções variando entre as respectivas opiniões: concordo plenamente, concordo, indiferente, discordo e discordo plenamente.

Para os alunos foram explicados somente os significados dos OA e explicações gerais sobre o funcionamento da ferramenta. Neste caso, o professor tinha uma lousa com recursos de pincel para explicação da aula em geral e o mesmo utilizou do projetor para exibir a aplicação em ambiente de PC com o auxílio do Emulador para simular a aplicação como se fosse um ambiente de TVDi.

Na primeira e segunda avaliação participaram duas (02) turmas C e D do ensino médio de 40 alunos, mas estiveram presentes na turma C, vinte e oito (28) alunos e na turma D, vinte e sete (27) alunos.

Na terceira avaliação participou uma turma de ensino superior do curso de Engenharia Ambiental (EN01S1), correspondente a uma revisão de apoio sobre o assunto de trigonometria. A turma formada por trinta e seis (36) alunos, destes estiveram presentes, apenas 26 alunos.

5.3.1 Avaliação de resultados

Por meio da descrição da avaliação anterior tornou-se possível coletar um conjunto de sugestões de melhoria para o protótipo desenvolvido. Foi identificado que durante os testes realizados em sala de aula os estudantes interagiam mais através da utilização da ferramenta. Após finalizada a avaliação, com base nos dados obtidos, foram produzidos gráficos com relação aos recursos disponibilizados pelo *framework* com o objetivo de aferir o grau de satisfação da ferramenta. Os principais são mostrados nos próximos itens desta seção.

5.3.2 Avaliação dos recursos do *framework* pelas turmas

Os gráficos demonstram os resultados das avaliações aplicadas em 03 (três) turmas, de acordo com as assertivas estabelecidas anteriormente.

A coleta de opinião sobre a ferramenta possibilita prática de conceitos aplicados em sala de aula, dos entrevistados entre as três turmas, 46 (56,8%) concordam plenamente com a assertiva, 34 (42,0%) concordam, 1 (1,2%) é indiferente, 0 (0%) discorda e 0 (0%) discorda plenamente com a assertiva. A Figura 69 disponibiliza esta representação.

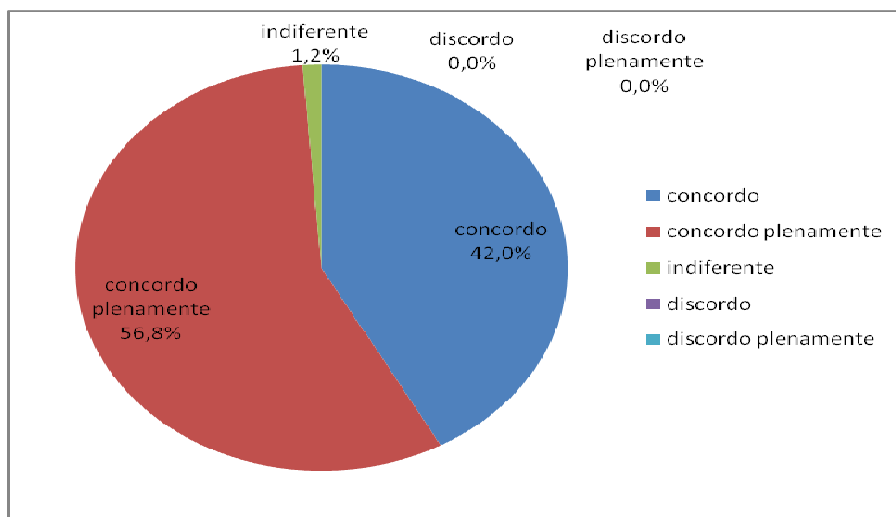


Figura 69 - Gráfico sobre os conceitos aplicados em sala de aula.

A coleta de opinião sobre a ferramenta oferece fácil navegação e operacionalidade, dos entrevistados entre as três turmas, 52 (64,2%) concordam plenamente com a assertiva, 25 (30,9%) concordam, 4 (4,9%) são indiferentes, 0 (0%) discorda e 0 (0%) discorda plenamente da assertiva. A Figura 70 disponibiliza a representação desta porcentagem.

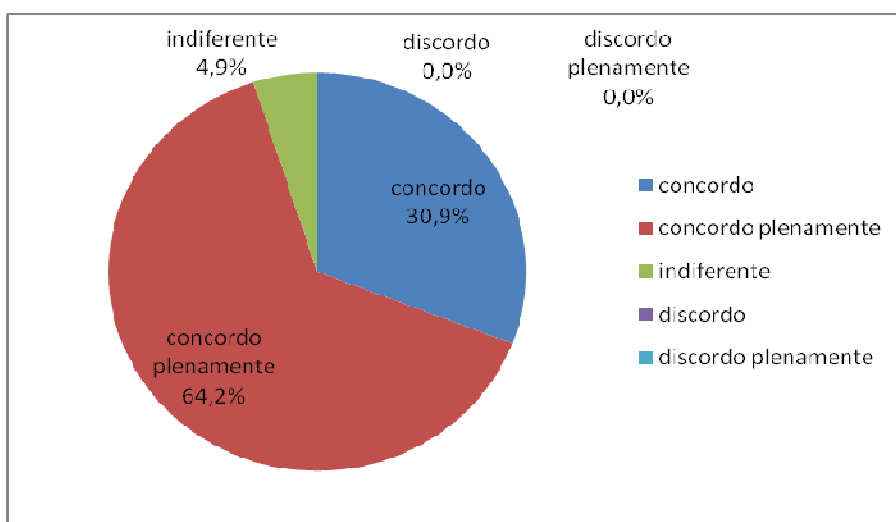


Figura 70 - Gráfico sobre a fácil navegação e operacionalidade.

A coleta de opinião sobre o uso de recursos de OA digitais pela ferramenta, dos entrevistados entre as três turmas, 53 (65,4%) concordam plenamente com a assertiva, 23 (28,4%) concordam, 5 (6,2%) são indiferentes, 0 (0%) discorda e 0 (0%) discorda plenamente da assertiva. Através da Figura 71 é disponibilizada graficamente esta representação.

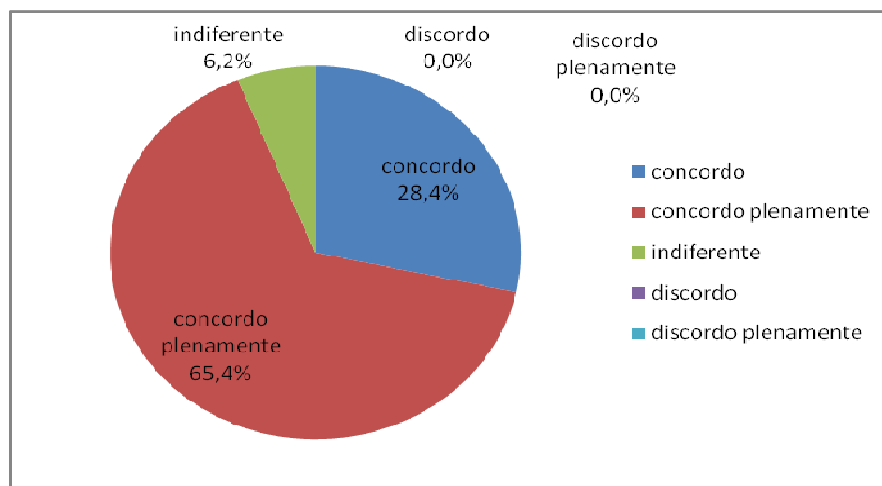


Figura 71 - Gráfico sobre o uso de objetos de aprendizagem digitais pela ferramenta.

A coleta de opinião sobre a aplicabilidade da ferramenta em instituições de ensino médio e superior, dos entrevistados entre as três turmas, 56 (69,1%) concordam plenamente com a assertiva, 21 (25,9%) concordam, 4 (4,9%) são indiferentes, 0 (0%) discorda e 0 (0%) discorda plenamente da assertiva. A Figura 72 disponibiliza graficamente esta representação.

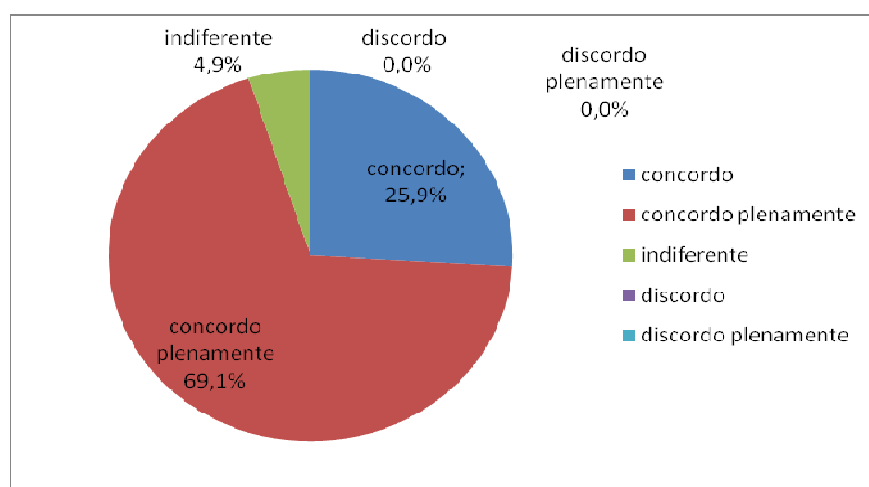


Figura 72 - Gráfico sobre a aplicabilidade da ferramenta para o ensino médio e superior.

A coleta de opinião sobre a variedade do suporte de conteúdos, dos entrevistados entre as três turmas, 51 (63%) concordam plenamente com a assertiva, 28 (34,6%) concordam, 2 (2,5%) são indiferentes, 0 (0%) discorda e 0 (0%) discorda plenamente da assertiva. Através da Figura 73 é disponibilizada graficamente a representação.

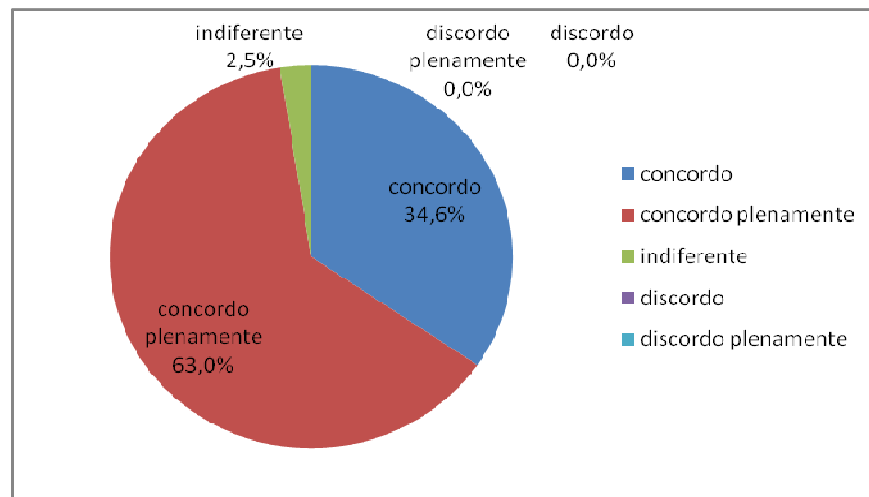


Figura 73 - Gráfico sobre as variedades de suporte de conteúdo da ferramenta.

A coleta de opinião sobre o uso da ferramenta em ambiente de TVD facilitaria a prática de estudos sobre a trigonometria, dos entrevistados entre as três turmas, 64 (79%) concordam plenamente com a assertiva, 16 (19,8%) concordam, 1 (1,2%) é indiferente, 0 (0%) discorda e 0 (0%) discorda plenamente da assertiva. Através da Figura 74 é disponibilizada graficamente a representação.

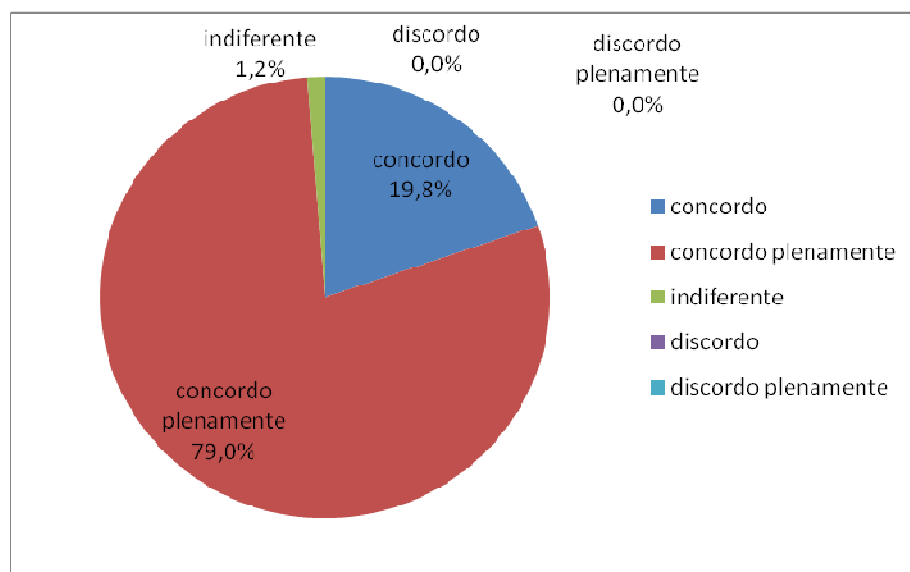


Figura 74 - Gráfico sobre o uso da ferramenta em ambiente de TVD.

5.3.3 Avaliação dos recursos do *framework* por cada turma

Foram observados os recursos empregados por turma, apresentando a descrição das avaliações e descartando as opiniões não indicadas, conforme a seguir:

Os entrevistados da turma C do total de 28 alunos, 20 (71,4%) concordaram plenamente com a assertiva, 7 (25%) concordaram e 3 (3,6%) foram indiferentes sobre a pesquisa de opinião que investigava se ferramenta possibilita prática de conceitos aplicados em sala de aula. A turma D do total de 27 alunos, 17 (63%) concordaram plenamente com a assertiva, 10 (37%) concordaram e 0 (0%) foi indiferente sobre a pesquisa de opinião que investigava se ferramenta possibilita prática de conceitos aplicados em sala de aula. Para turma EN01S1 sobre essa mesma pesquisa, 19 (69,2%) concordaram plenamente, 7 (30,8%) concordaram e 0 (0%) foi indiferente. Na Figura 75 é disponibilizada a análise desta avaliação.

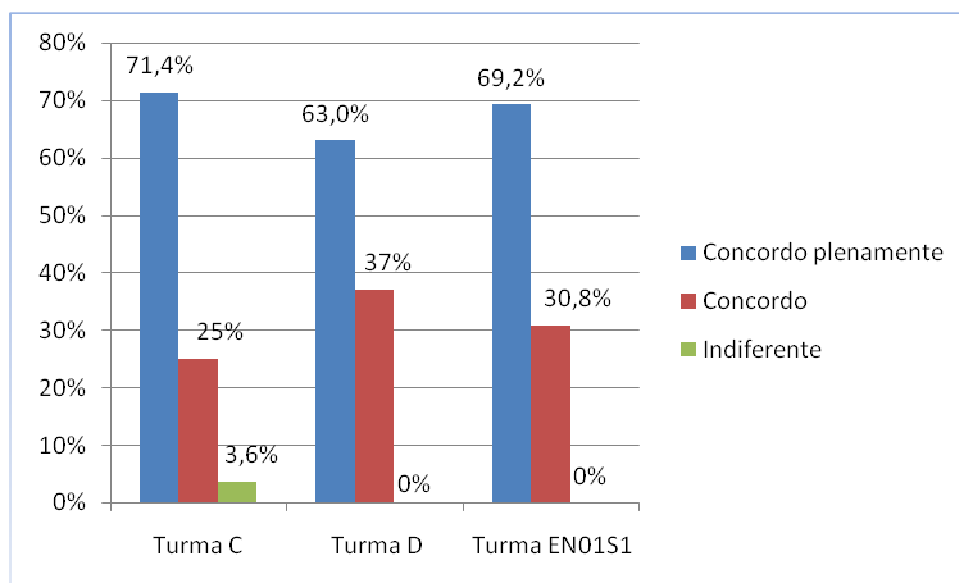


Figura 75 - Gráfico por turma sobre a prática de conceitos aplicados em sala de aula.

Para pesquisa de opinião sobre a ferramenta oferece fácil navegação e operacionalidade, os entrevistados da turma C do total de 28 alunos, 22 (78,6%) concordaram plenamente com a assertiva, 5 (17,8%) concordaram e 1 (3,6%) foi indiferente. A turma D do total de 27 alunos, 17 (70,4%) concordaram plenamente com a assertiva, 10 (29,6%) concordaram e 0 (0%) foi indiferente. Para turma EN01S1, 14 (69,2%) concordaram plenamente, 12 (30,8%) concordaram e 0 (0%) foi indiferente. Na Figura 76 é disponibilizada a análise desta avaliação.

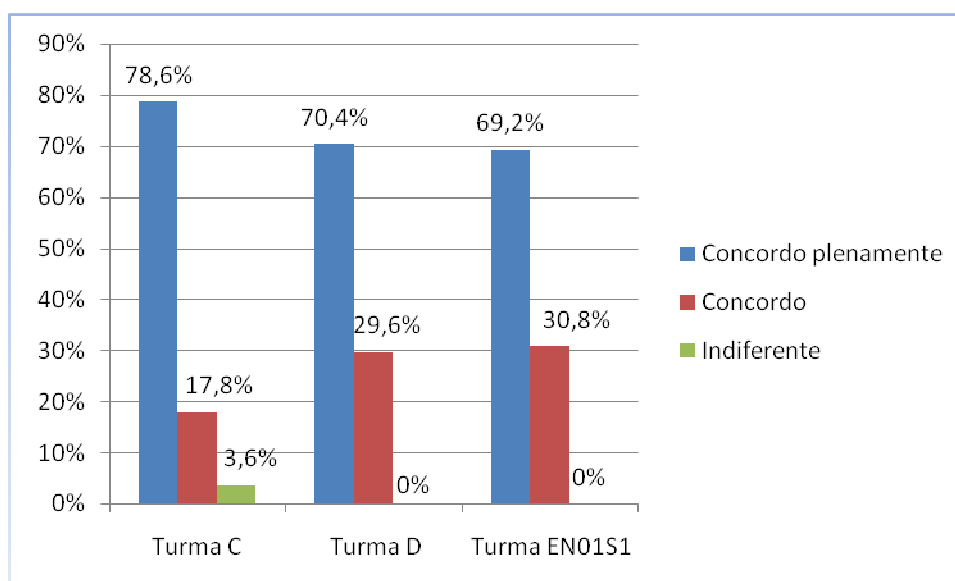


Figura 76 - Gráfico sobre a fácil navegação e operacionalidade da ferramenta.

A pesquisa de opinião sobre o uso de OA pela ferramenta facilita a organização e melhor compreensão de conteúdos em uma aula de trigonometria, os entrevistados da turma C do total de 28 alunos, 21 (75%) concordaram plenamente com a assertiva, 6 (21,4%) concordaram e 1 (3,6%) foi indiferente. A turma D do total de 27 alunos, 18 (66,7%) concordaram plenamente com a assertiva, 7 (25,9%) concordaram e 2 (7,4%) foram indiferentes. Para turma EN01S1, 15 (57,7%) concordaram plenamente, 11 (42,3%) concordaram e 0 (0%) foi indiferente. A Figura 77 disponibiliza a análise desta avaliação.

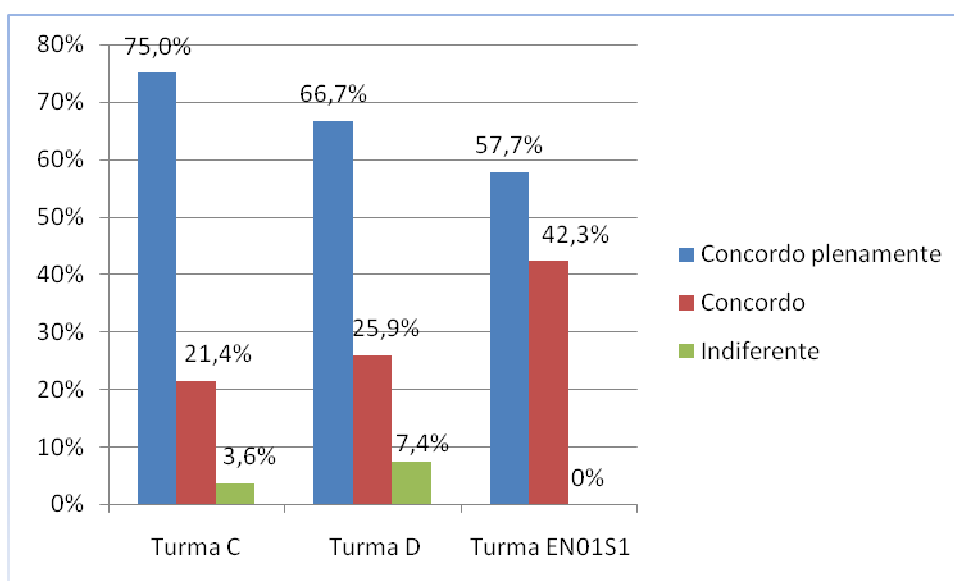


Figura 77 - Gráfico por turma sobre o uso de OA pela ferramenta.

Em relação à pesquisa de opinião sobre a ferramenta é interessante usar durante o ensino médio e superior, os entrevistados da turma C do total de 28 alunos, 20 (71,4%) concordaram plenamente, 5 (17,8%) concordaram e 3 (10,8%) foram indiferentes. Para turma D do total de 27 alunos, 23 (85,2%) concordaram plenamente, 2 (7,4%) concordaram e 2 (7,4%) foram indiferentes. Na turma EN01S1 do total de 26 alunos, 15 (57,7%) concordaram plenamente, 11 (42,3%) concordaram e 0 (0%) foi diferente. Na Figura 78 é visualizado graficamente este resultado.

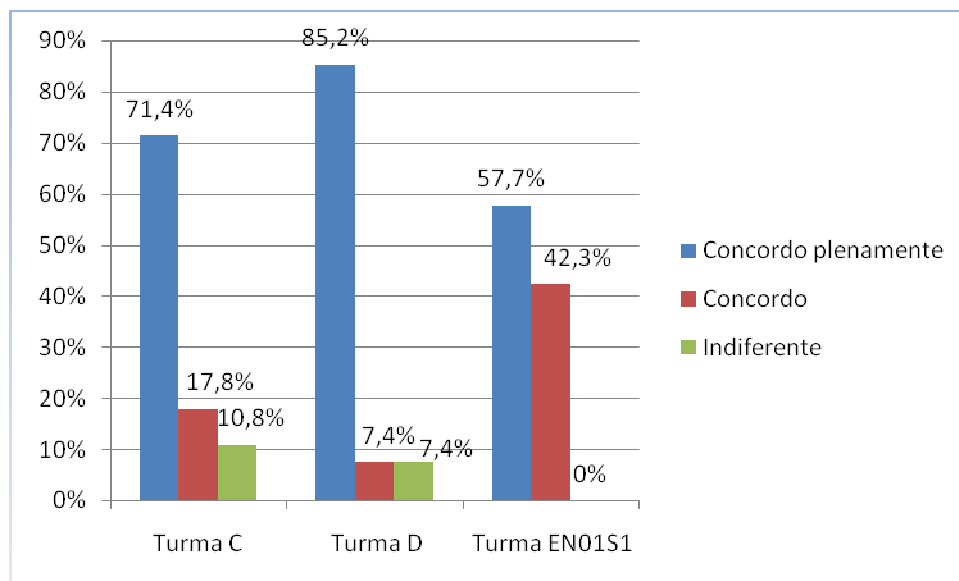


Figura 78 - Gráfico por turma sobre aplicabilidade da ferramenta

Para pesquisa de opinião sobre a ferramenta oferece variedades de suporte de conteúdo durante uma aula de trigonometria. Do total de 28 alunos da turma C, 20 (71,4%) concordaram plenamente, 7 (25%) concordaram e 1 (3,6%) foi indiferente. Na turma D do total de 27 alunos, 21 (77,8%) concordaram plenamente, 6 (22,2%) concordaram e 0 (0%) foi indiferente. Os alunos da turma EN01S1, 10 (38,4%) concordaram plenamente, 16 (61,6%) concordaram e 0 (0%) foi indiferente. A Figura 79 exibe o resultado desta porcentagem graficamente.

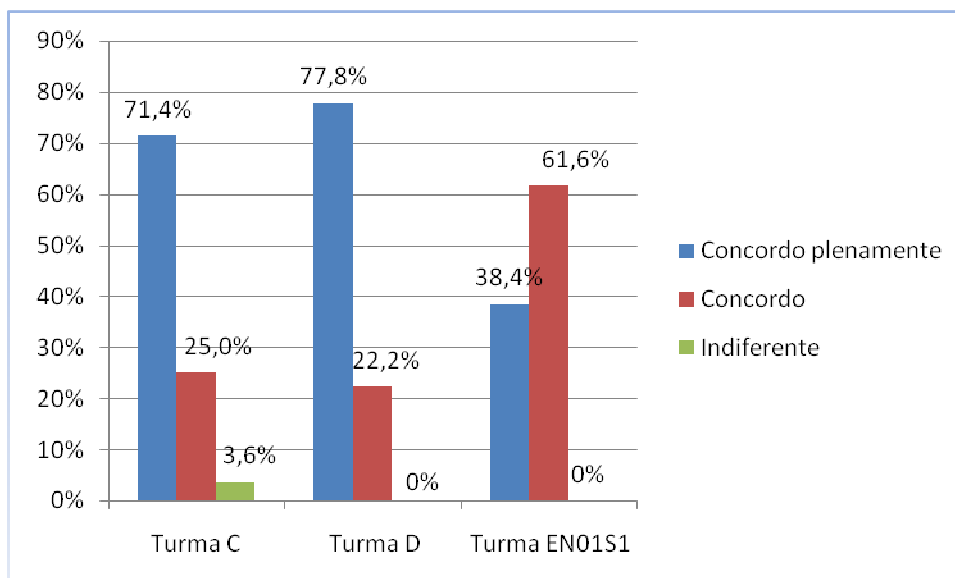


Figura 79 - Gráfico por turma sobre as variedades de suporte de conteúdos.

A pesquisa de opinião realizada sobre o uso desta ferramenta em ambiente de TVD facilitaria a prática do estudo sobre a trigonometria. Do total de 28 alunos da turma C, 24 (85,7%) concordaram plenamente, 3 (10,7%) concordaram e 1 (3,6%) foi indiferente. Na pesquisa para turma D em um total de 27 alunos, 25 (92,6%) concordaram plenamente, 2 (7,4%) concordaram e 0 (0%) foi indiferente. A pesquisa elaborada na turma EN01S1 do total de 26 alunos, 12 (46,2%) concordaram plenamente, 14 (53,8%) concordaram e 0 (0%) foi indiferente. Esta pesquisa é comprovada conforme a Figura 80 abaixo.

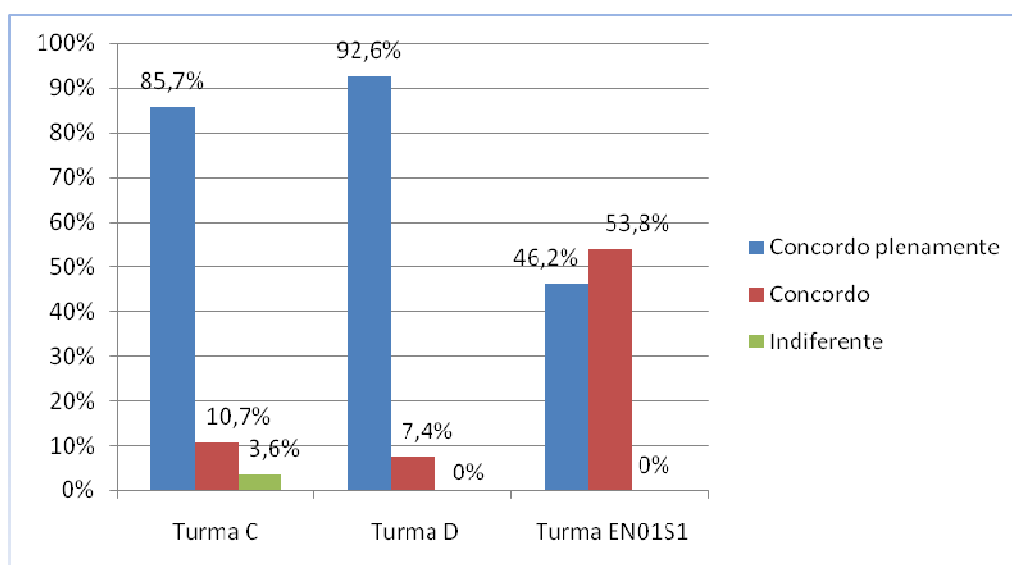


Figura 80 - Gráfico sobre uso da ferramenta em um ambiente de TVD.

5.3.4 Análise dos resultados das pesquisas

É importante destacar que embora as avaliações detalhadas nas seções anteriores terem uma boa aceitação ainda possuem algumas limitações, por exemplo, não foi avaliado o aprendizado sem a utilização da ferramenta e após a utilização da ferramenta aplicando provas para verificar se houve melhor rendimento entre as turmas. Foi diagnosticado também entre os alunos que os módulos de vídeos, gráficos e jogos poderiam ser melhorados. A turma C e D sugeriram que os módulos de vídeos fossem compostos pelo próprio professor ministrando as aulas, pois o mesmo saberia compreender melhor a realidade da turma.

O módulo de gráficos foi considerado interessante, mas teve suas limitações, pois essas turmas propuseram melhorias para as projeções das funções trigonométricas na circunferência, indicando também a apresentação do ângulo na circunferência, e não apenas projeções de acordo com seus eixos. Já a turma EN01S1 recomendou que para conceitos básicos de revisão de trigonometria para o ensino superior essa ferramenta seria ideal. Mas, os mesmos indicaram que esta ferramenta poderia se estender para explicar projeções de funções de forma geral, não apenas de funções de Seno e Cosseno no módulo de gráficos. Foi sugerido também que o módulo de jogo fosse melhorado no aspecto de não apresentar só um cenário para cada nível, apresentando cenários diferentes e que fossem readaptados para cada nível.

5. 4 CONCLUSÃO

Neste capítulo foi descrito como foi o processo para escolha da arquitetura de desenvolvimento do *framework O-Learning* TVDi e as avaliações de suas funcionalidades. Primeiramente foram realizadas as avaliações na plataforma de desenvolvimento brasileira utilizando o *middleware* Ginga. Nesta plataforma foram diagnosticadas dificuldades em utilizar APIs para tratamento de interfaces gráficas e manipulações de funções matemáticas para TVD, principalmente para utilização do Ginga-J, por falta ainda de exemplos e testes validados em receptores digitais. Foram apresentados os resultados apenas em ambiente declarativo, fazendo uso de uma máquina virtual para executar aplicações referentes ao Ginga-NCL. Por este motivo a arquitetura deste *framework* foi desenvolvida primeiramente na plataforma européia através de um receptor digital com *middleware*

MHP. Como se trata de uma ferramenta para atender requisitos pedagógicos foram realizadas avaliações em ambientes real de aprendizagem, visando identificar melhorias em seus módulos desenvolvidos.

Estas avaliações indicaram limitações de funcionalidades que contribuíram para informar os pontos positivos e os que precisam ser melhores empregados pela ferramenta, de modo que novas versões possam ser aprimoradas e testadas de forma mais criteriosa. Através destas avaliações no próximo capítulo serão descritas não só as dificuldades para o desenvolvimento do *framework*, mas também contribuições e trabalhos futuros que possam dar continuidade a esta pesquisa.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma arquitetura de *software* dividida em módulos de objetos digitais de aprendizagem para o apoio didático em aulas de trigonometria, com o foco em aplicações para TVDi devido à forte tendência cultural de grande parte da população brasileira utilizar a TV como meio de acesso à informação e entretenimento. Através desta arquitetura pode-se acrescentar e adaptar recursos de interatividade pela utilização do canal de retorno com o apoio da Internet. Este trabalho contribui com testes de desenvolvimento de plataforma para os *middleware* Ginga e MHP. Foi possível verificar APIs existentes nesses padrões que pudessem ser substituídas ou adaptados novos procedimentos que atendessem essas duas plataformas na construção de interfaces gráficas e manipulações de eventos em TVDi.

Observou-se o emprego massivo de objetos digitais para aplicações em ambiente de *E-Learning*, mas uma carência de maiores contribuições para fins educacionais através do *T-Learning*, principalmente para plataforma brasileira. Foram identificadas várias oportunidades de pesquisas que melhoram a aprendizagem e interação entre o que transmite e o que recebe a informação através destes conceitos, pois nos estudos de casos desenvolvidos e testados houve uma boa aceitação da manipulação desses recursos no *framework*, porém com melhorias a serem acrescentadas e desenvolvidas.

Esta arquitetura contribui ainda como modelo de aplicações que podem ser exploradas pedagogicamente para organizar conteúdos não só para o estudo da matemática no contexto da trigonometria, servindo como base para outros assuntos da matemática ou até mesmo em disciplinas diferentes. Isso se comprova pelas variedades dos OA que foram desenvolvidos com funcionalidades diferentes em um mesmo domínio específico do problema, mas que podem ser reutilizadas, devido à aplicabilidade do paradigma orientado a objetos empregado nesta arquitetura.

Por fim, a elaboração desta arquitetura para manipulação de objetos digitais em TVDi seguiu detalhadamente os conceitos definidos e analisados ao longo do texto dessa dissertação. O resultado foi um ambiente testado na plataforma emulada e em plataforma real de TVDi com utilização de um STB para avaliar o seu comportamento e limitações encontradas que podem ser superadas em trabalhos futuros. Assim essa arquitetura pode ser disponibilizada gratuitamente para fins de realizações de novos trabalhos na comunidade acadêmica.

6.1 DIFICULDADES ENCONTRADAS

As principais dificuldades encontradas tratam-se ainda da falta da documentação finalizada e disponível por completa na plataforma Ginga-J, visto que na plataforma de Ginga-NCL testada inicialmente neste trabalho, já se tem muitos trabalhos elaborados e bibliografias, como: livros, trabalhos de finalização de cursos e artigos. Em diversas pesquisas elaboradas foram encontradas várias bibliografias para o Ginga-J que tratava da especificação da API LWUIT de forma geral, mas de forma limitada para construir interfaces gráficas, pois funciona apenas parte dos recursos disponíveis o que causou outra limitação no momento para este trabalho.

A partir desta investigação não foi possível decidir em testar em ambiente Real de TVDi para o ambiente na plataforma Ginga. Apesar dessas dificuldades para o desenvolvimento de trabalhos para o Ginga-J em suas comunidades, já houve um grande avanço de pesquisas e cada vez mais são disponibilizados materiais que contribuem para o desenvolvimento de novas aplicações. A outra perspectiva para este trabalho são recursos existentes através da API JavaTV e a facilidade de manipulação de interface gráfica pela API HAVI e AWT que servem de apoio para possível adaptação para plataforma Ginga, não se transformando em uma tarefa complexa e onerosa.

Tratou-se de outra dificuldade a padronização de interface gráfica, pois o modelo de interface gráfica para TVD no mercado continua em constante mudança que levou este trabalho ainda não definir o padrão final de interface, apenas para testes, apesar de realizadas várias modificações nas interfaces deste *framework*. Foram identificados problemas também no tamanho da aplicação que cresce bastante chegando alcançar 58 MB quando se formava novas imagens que seriam integradas ao *framework*, visto como outra limitação de desempenho que pode ainda ser melhorado com novas adaptações de busca nos repositórios de imagens, principalmente para testes em STBs.

6.2 TRABALHOS FUTUROS

Os trabalhos futuros apresentados nesta seção poderão dar continuidade do aperfeiçoamento desta pesquisa, através de melhorias dos módulos do *framework O-Learning* TVDi e oportunidades para acrescentar outros módulos que explorem os conceitos de objetos digitais de aprendizagem. A outra contribuição é a realização de novas pesquisas para a plataforma de desenvolvimento de *middleware* de TVDi brasileira que

possibilite investigar e realizar testes entre a plataforma do *middleware* MHP e Ginga-J, além de desenvolvimento específicos para as camada do *middleware* Ginga através da ponte entre Ginga-NCL e Ginga-J. Dentre essas possibilidades de trabalhos apresentam-se:

- Estender o módulo de mapa conceitual para o módulo de navegação dinâmica. Neste, cada item do mapa poderá servir de acesso ao novo mapa permitindo a interação através das teclas do controle de TVD;
- Migrar a aplicação para plataforma Ginga-NCL tratando toda parte de apresentação e interligando com a plataforma Ginga-J para tratar de manipulações procedurais orientada a objetos dos módulos de interfaces gráficas e funções matemáticas;
- Adaptar a arquitetura para atender a aplicações de educação a distância através da utilização do canal de retorno e de receptor digital em ambiente real de TVDi;
- Construir um repositório de OA para cadastrar e manipular questões para os módulos de desafios e jogo educativo;
- Construir um módulo gráfico para analisar e gerar todos os gráficos de funções trigonométricas;
- Adaptar a aplicação para o professor personalizar e construir sua própria aula e disponibilizar aos seus alunos para prática do aprendizado; e
- Realizar um aprofundamento teórico sobre uma teoria pedagógica a ser utilizada como referencial na construção de objetos de aprendizagem para TVDi. Possivelmente essa teoria seria a de Jean Piaget.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDREATA, Jomar Alberto. **InteraTV: um portal de aplicações colaborativas em TV Digital Interativa utilizando a plataforma MHP**. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal de Santa Catarina, Florianópolis, 2006.

ARIAS, J. J.; NORES, M. L.; DUQUE, J. G.; REDONDO, R. P. D.; FERNÁNDEZ, Y. B; CABRER, M. R; SOLLA, A. G; VILLAS, A. F. **Provision of distance learning services over Interactive Digital TV with MHP**. Ed. Elsevier. pp. 927-949. 2008.

ASSIS, Semíramis, Ribeiro; SUZANO, Rita. **Frameworks: Conceitos e Aplicações**. In: CienteFico. Ano III, v. II, Salvador, jul-dez, 2003.

ABNT. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **Parte 4: Ginga-J - Ambiente para a execução de aplicações procedurais**: NBR-15606-4: 2010.

AUDINO, Daniel Fagundes; NASCIMENTO, Rosemy da Silva. **Objetos de aprendizagem – diálogos entre conceitos e uma nova proposição aplicada à educação**. In: Revista Contemporânea de Educação, vol. 5, n. 10, jul./dez. 2010.

BARBOZA, D. C.; CLUA, E. W. G. **Ginga Game: A Framework for Game sevelopment for the interactive Digital Television**. Simpósio Brasileiro de Games – SBGAMES, Rio de Janeiro, 2009.

BATES, P. J. **A study into TV-based interactive learning to the home**. Pjb Associates, UK, 2003.

BECKER, V.; MONTEZ, C. **TV Digital Interativa: Conceitos, desafios e perspectivas para o Brasil**. Florianópolis: UFSC, 2005.

BELLOTTI, F.; VROCHIDIS, S.; PARISSI, E.; LHOAS, P.; MATHEVON, D.; PELLEGRINO, M.; BO, G.; KOMPATSIARIS, I. **T-learning Courses Development and Presentation Framework**. In: IEEE Multidisciplinary Engineering Education Magazine, vol. 3, n. 3, set. 2008, p. 69-76.

BERTOTTI, Giuliano A.; ALMEIDA, Felipe A. D.; BACCAN, Davi D. **Educação à distância Mediada pela Televisão Interativa: Panorama Atual e Perspectivas para o Brasil**, 2004. Disponível em: http://sbie2004.ufam.edu.br/anais_cd/extras/anaisvoli/vdigital/artigos/5544.pdf.

BLAIR, Preston. **Cartoon Animation**. Laguna Hills: Walter Foster, 1994.

BRACKMANN, Christian Puhmann. **Sistema Brasileiro de TV Digital**. Dissertação (Mestrado em Informática), Universidade Católica de Pelotas, Pelotas, 2008.

BREUNING, Mônica Caroline. **Expectativas pela implantação da TV Digital e as ameaças à TV analógica**. In: XXXII CONGRESSO BRASILEIRO DE CIÊNCIAS DA COMUNICAÇÃO: Intercom – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação, Curitiba, 2009. p. 1-13.

CABRAL, Anderson Ricardo Yanzer; GIRAFFA, Lucia Maria Martins. **Avaliação de cursos WBT utilizando Mapas Conceituais**. In: XIII Simpósio Brasileiro de Informática na Educação – SBIE Unisinos, São Leopoldo, 2002.

CAMPOS, Sylvia; BORGES, Henrique; BORBA, Paulo. **Introdução ao JDK**. Universidade Federal de Pernambuco, 2011. Disponível em: <http://www.cin.ufpe.br/~phmb/ip/MateriaDeEnsino/IntroducaoAoJDK/IntroducaoAoJDK.htm>. Acesso em 11 de junho de 2011.

CARNEIRO, Cristiane Marise Pérez da Silva. **Framework de Aplicações Orientadas a Objetos- Uma Abordagem Interativa e Incremental**. Dissertação (Mestrado em Redes de Computadores). Universidade da Bahia, Salvador, 2003.

CARVALHO, Emílio Souto de. **A TV digital no Brasil e as tendências futuras**. Monografia (Sistemas de Informação). Faculdade Anhanguera, Belo Horizonte, 2011.

COSTA, Juliano Rodrigues. **EMTV - Extensão de Middleware para TV Digital brasileira baseada em componentes de softwares**. Dissertação (Mestrado em Engenharia Elétrica). Universidade Federal do Amazonas, Manaus, 2008.

DAMSOM, M. **Beginning C++ Game Programming**. Boston: Premier Press, 2004. 325 p.

DEITEL, Paul; DEITEL, Harvey. **Java: como programar**. 8ª ed. São Paulo: Pearson Prentice Hall, 2010.

DTV. SITE OFICIAL DA TV DIGITAL BRASILEIRA. **Onde já tem TV Digital**. 2011. Disponível em: <http://www.dtv.org.br/index.php/onde-ja-tem-tv-digital>. Acesso em: dez. 2011.

____. **Quando a TV Digital vai chegar na minha cidade**. Disponível em: <http://www.dtv.org.br/index.php/duvidas-frequentes/quando-a-tv-digital-vai-chegar-na-minha-cidade/>. Acesso em: dez. 2011.

____. **História da TV Digital no Brasil**. Disponível em: <http://www.dtv.org.br/index.php/informacoes-tecnicas/historia-da-tv-digital-no-brasil>. Acesso em: jan. 2012.

DVB. DIGITAL VIDEO BROADCASTING. **Specification for the use of Video and Audio Coding in DVB services delivered directly over IP protocols**, July 2009. Disponível em: <http://www.dvb.org/>. Acesso em: fev. 2012.

____. **Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, New Gathering and other broadband satellite applications (DVB-S2)**. Disponível em: <http://www.dvb.org/>. Acesso em: fev. 2012.

EMMERICH, Wolfgang; AOYMA, Mikio; SEVENTEK, Joe. **The impact of Research on Middleware Technology**. In: ACM SIGSOFT Software Engineering Notes, v. 32, n. 1, p. 21, 2007. apud COSTA, Juliano Rodrigues. EMTV - Extensão de Middleware para Tv Digital brasileira baseada em componentes de softwares. Dissertação (Mestrado em Engenharia Elétrica). Universidade Federal do Amazonas, Manaus, 2008.

FAYAD, M. E.; SCHMIDT, D. C. **Object-oriented application frameworks**. In: Communications of the ACM, vol. 40, p. 10, 1997.

_____. **Building Application Frameworks: Object Oriented Foundations of Framework Design**. 1.ed. Wiley, Canada, 1999.

FOWLER, Martin. **UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos**. trad. João Tortello. 3. ed. Porto Alegre: Bookman, 2005.

FRANCO, Bárbara Bernadini de. **Convergência digital de sistemas de aprendizado colaborativo, considerando ambientes da Web e da TV Digital no Brasil**. Dissertação (Mestrado Ciência da Computação). Universidade Estadual Paulista “Júlio de Mesquita Filho” - UNESP, São José do Rio Preto, 2009.

GAMA, Carmem Lúcia Graboski da. **Método de construção de objetos de aprendizagem com aplicação em métodos numéricos**. Tese (Doutorado em Métodos Numéricos em Engenharia). Universidade Federal do Paraná, Curitiba, 2007.

GAMMA, E; Helm, R; Johnson, R. **Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos**. Editora Bookman, Porto Alegre, RS, 2000.

GOMES, André Luiz Pereira. **Flexframe: Framework para Desenvolvimento de Sistemas Flexíveis, que parametrizem regras de negócios**. Dissertação (Mestrado em informática), Universidade do Ceará, Fortaleza, 2002.

GOMES, Fábio de Jesus Lima; LIMA, José Valdeni de; NEVADO, Rosane Aragón. **Definindo orientações de usabilidade para o desenvolvimento de objetos de aprendizagem para TV Digital**. In: Revista Novas Tecnologias na Educação, v. 5, p.1-10, 2007.

GOULART, Leandro Jekimim. **Estudo de Caso de uma extensão de Middlewares de TV Digital Interativa para suporte a aplicações residentes não-nativas**. Dissertação (Mestrado em Ciências da Computação), Universidade Estadual Paulista “Júlio de Mesquita Filho”, São José do Rio Preto, 2009.

GUILHERME, Josimar Ferreira; GOMARIN, Wellington César. **Desenvolvimento de um aplicativo para Tv Digital usando as bibliotecas de Java Tv**. Monografia (Ciências da Computação), Universidade de Franca, Franca, 2009.

GUTIERREZ, Suzana de Souza. **Distribuição de conteúdos e aprendizagem on-line**. TRENOE – Revista Novas Tecnologias na Educação, v. 2, p. 1-14, 2004.

HASHIM, Umami Raba'ah bt; KADIR, Aina Fazliana Abdul; ALIAS, Azrita; HASSAN, Elia Erwani. **Development of Learning Object for Engineering Courses in UTeM**. In: International Conference on Engineering Education (ICEED), Kuala Lumpur, Malaysia, dez. 2009.

HEMMLEPP, Paulo Martinelli. **Aplicações de TV Digital sensíveis a contexto para dispositivos móveis**. Monografia (Curso Ciências da Computação), Universidade Federal de Pernambuco, 2006.

HOFMANN, Paulo. **Vestibulando Digital Matemática Aula 28 Relações Trigonométricas nos Triângulos e Retângulos.** Disponível em: <http://www.youtube.com/watch?v=zmB1qZKoQvQ>. Acesso em: fev. 2012.

_____. **Vestibulando Digital Matemática Aula 29 Funções Trigonométricas no ciclo Trigonométrico I.** Disponível em: <http://www.youtube.com/watch?v=zmB1qZKoQvQ>. Acesso em: fev. 2012.

IEEE. **LEARNING TECHNOLOGY STANDARD COMMITTEE (LTSC).** (2005) In: WG12 – Learning Object Metadata. Disponível em: <http://ltsc.ieee.org/wg12>. Acesso em maio. 2011.

JOKIPELTO, P. A. **T-learning model for learning via digital TV.** In: 16th EAEEIE Conference, Lappeeranta, Finlândia, 2005. Disponível em: <http://it.lut.fi/eaeeie05/index.html>. Acesso em: out. 2011.

KEMCZINSKI, Avanilde; HOUNSELL, Marcelo da Silva; GASPARINI, Isabela; GEHRKE FILHO, Ricardo; SILVA, Thiago Corrêa da. **Repositório de objetos de aprendizagem para área de computação e informática – ROAI.** Anais do XXII SBIE – XVII WIE, Aracaju, p. 234-243, nov. 2011.

LEMONS, G.; FERNANDES, J.; SILVEIRA, G. **Introdução à Televisão Digital Interativa: Arquitetura, Protocolos, Padrões e Práticas.** In: Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Comunicação, Anais... Salvador, 2004.

LIMA, Adilson da Silva. **UML 2.0: do requisito à solução.** 3. ed. São Paulo: Érica, 2008.

LIRA, Antonio da Fonseca. **O processo da construção do conceito matemático de limite pelo aprendiz com utilização de objetos digitais.** Tese (Doutorado em Informática na Educação), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

LONGMIRE, W. **A printer on learning objects.** American Society for Training & Development. Virginia, USA, 2001.

LUCAS, Adriano dos Santos. **Personalização para Televisão Digital utilizando a estratégia de Sistema de Recomendação para ambientes multiusuário.** Dissertação (Mestrado em Ciências da Computação), Universidade Federal de São Carlos, São Carlos-SP, 2009.

LYTRAS, M.; LOUGOS, C.; CHIZOS, P.; POULOU, A. **Interactive Television and e-Learning Convergence: Examining the Potential of t-Learning.** European Conference on e-Learning. p. 1-11, 2002.

MAIA, Lady Daiana de Oliveira. **GameTVD: uma proposta de arquitetura para framework de jogos 2D para TV digital.** Dissertação (Mestrado em Informática), Universidade Federal do Amazonas, Manaus, 2010.

MARQUES, Vinicius Carneiro. **Desenvolvimento de uma aplicação para TV Digital utilizando Middleware nacional Ginga-NCL voltado ao aprendizado de crianças.** Projeto de Graduação em Ciências da Computação, Universidade Católica de Pelotas, Pelotas, 2008.

MATTOS, D. M. F. **Virtualização: VMWare e Xen**. 2008. Disponível em: http://www.gta.ufrj.br/grad/08_1/virtual/artigo.pdf. Acesso em: Agosto. 2011.

MATTSSON, M. **Object-Oriented Frameworks**. Licentiate Thesis, Lund University, Department of Computer Science, 1996.

_____. **Evolution and Composition of Object- Oriented Frameworks**, 150 f. Thesis (Doctor of Philosophy in Engineering). Faculty of Tecnology, University of Karlskrona, Ronneby, 2000.

_____.; BOSCH, L. **Framework Composition: Problems, Causes and Solutions**. In: Tecnology of Object-Oriented Languages and Systems, Santa Bárbara, CA. p. 203-214, jul/ago. 1997.

MENEZES, Crediné S.; LIRA, Antonio F.; FERRETI, Cláudio. **ODAI – Objetos Digitais para aprendizagem Interacionista**. SBIE – Simpósio Brasileiro de Informática na Educação, 2006.

MINHO, Marcelo; SANTOS, Alexandre; BRIDI, Larissa; ALVES, Lynn; MORET, Marcelo. Jogos eletrônicos para TV Digital: relato de uma experiência. In: **VIII Brazilian Symposium on Games and Digital Entertainment**, Rio de Janeiro, p. 1-6, out. 2009.

MIRANDA, Raquel Mello. **GROA: um gerenciamento de repositórios de objetos de aprendizagem**. Dissertação (Mestrado em Ciências da Computação), Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004. 80 f.

MONTEIRO, Bruno de Sousa. **AMADEUS-TV: Portal Educacional na TV Digital Integrado a um Sistema de Gestão de Aprendizado**. Dissertação. Universidade Federal de Pernambuco, Recife, 2009.

_____.; PROTA, Thiago Monteiro; SOUZA, Fernando da Fonseca; GOMES, Alex Sandro. **Desenvolvimento de objetos de aprendizagem para TVDi**. In: XIX Simpósio Brasileiro de Informática na Educação (SBIE 2008). Pernambuco, p. 198-207, 2008.

_____.; _____.; _____.; _____. **Amadeus TV: Portal Educacional na TV Digital Integrado a um Sistema de Gestão de Aprendizado**. In: Revista Brasileira de Informática na Educação (SBIE 2010), vol.18, n. 1, 2010. p. 5-16.

MOREIRA, Filipe Rodrigues de S. **Trigonometria**. 2005. Disponível em: http://www.rumoaioita.com/site/index.php?option=com_content&view=article&id=66:apostila-de-trigonometria&catid=12:matematica&Itemid=86. Acesso em: fev. 2012.

NEVES, Rogério Baldini das. **Explorando o canal de retorno em sistema de Televisão Digital Interativa: uma abordagem centrada no suporte à comunicação entre aplicações e provedores de serviços**. Dissertação (Mestrado em Informática), Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, 2010.

OLIVEIRA, Ana Cláudia Tiessi Gomes de. **ViMeT – Projeto e implementação de um framework para aplicações de treinamento médico usando realidade virtual**. Dissertação (Mestrado em Ciência da Computação), Centro Universitário Eurípedes de Marília, Marília, SP, 2007.

OLIVEIRA, Carina Teixeira de. **Um estudo sobre os padrões de middleware para televisão digital interativa**. Monografia (Tecnologia Telemática). Centro Federal de Educação Tecnológica do Ceará – CEFETCE, Fortaleza, 2005.

OLIVEIRA, Gabriel Paes Barreto do Nascimento; CAVALCANTE, José Henrique Santos; MORAIS, Henrique Simch; GONZAGA, Michela dos Santos. **Sistemas de Ensino da Matemática via TV Digital Interativa**. Monografia (Tecnologia em desenvolvimento de Software), Centro Universitário do Norte, Manaus, 2009.

ORACLE. Oracle Technology Network. **Java TV**. Disponível em: <http://www.oracle.com/technetwork/java/javame/javatv/overview/getstarted/index.html>. Acesso em fev. 2012.

PADOVANI, Stephania; MOURA, Dinara. **Navegação em hipermídia: uma abordagem centrada no usuário**. Rio de Janeiro: Ciência Moderna, 2008.

PAZOS-ARIAS, J. J. et al. ATLAS: a framework to provide multiuser and distributed t-learning services over MHP. *Software: Practice and Experience*. John Wiley & Sons, 2006, p. 845-869.

PEREIRA, R. César Júlio, **Análise de Dados Qualitativo**. São Paulo: Edusp, 2001.

PEROZZO, Reiner Franchesco. **Framework para Integração entre Ambientes Inteligentes e o Sistema Brasileiro de TV Digital**. Tese (Doutorado em Engenharia Elétrica). Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

PIAGET, Jean. **Criatividade**. In: *Criatividade, psicologia, educação e conhecimento do novo*. Jean Piaget et. al. (Org.) Mario Sergio Vasconcelos. São Paulo: Editora Moderna, 2001, p. 11-20.

PICANÇO, Wollace de Souza; ALBUQUERQUE, Wanderlan; ESTEVES, Anderson F.; BARBOZA, Ricardo da Silva; JESUS, Tatiane da Silva. **Jogo de Raciocínio Lógico Matemático Via TV Digital Emulado no Computador como Ferramenta Educacional**. In: I Simpósio de Informática e Geotecnologia de Santarém, Santarém, 2011.

PRESSMAN, Roger S. **Engenharia de Software**. 6. Ed. São Paulo: Mc Graw Hill, 2006.

PROTA, Thiago Monteiro; **MoonDo: um framework para desenvolvimento de aplicações declarativas no SBTVD**. Monografia (Ciência da Computação), Universidade Federal de Pernambuco, Recife, 2010.

RIBEIRO, Ana Carolina; LONGARAY, Ariane N.C.; BEHAR, Patricia A. **Práticas Criativas na Web 2.0: a construção de um objeto de aprendizagem**. In: XXII Simpósio Brasileiro de Informática na Educação (SBIE 2011). Aracaju, p. 313-320, 2011.

SÁ FILHO, C. S.; MACHADO, E. C. **O computador como agente transformador da educação e o papel do objeto de aprendizagem**, 2004. Disponível em: <http://www.universia.com.br/materia/materia.jps?materia=5939>. Acesso em: jan. 2012.

SANCIN, Chiara et. al. **T-learning for social inclusion**. *Elearning Papers*, Dida, Italy, n.12, p. 1-12. Fev. 2009.

SANTOS, Davi Trindade. **Estudo de aplicativos de TVDi para Educação a Distância**. Dissertação (Mestrado em Engenharia Elétrica). Universidade de. Campinas, Campinas, 2007.

SARAIVA JR., Erisvaldo Gadelha. **Ginga-J Emulator: Uma ferramenta de Execução de Aplicações Imperativas para o Middleware Ginga**. Monografia (Ciências da Computação). Universidade Federal da Paraíba, João Pessoa, 2010.

SEDREZ, Fernando de Moraes. **Desenvolvimento de um aplicativo para TV Digital Interativa utilizando a tecnologia Java TV**. Monografia (Ciências da Computação), Universidade Católica de Pelotas, Pelotas, 2008.

SCHMIDT, D.C.; GOKHALE, A.; NATARAJAN, B. **Leveraging Application Framework**. ACM Queue, v. 2, 5 jul/ago. 2004, p. 66-75.

SILVA, A. A.; GUIMARÃES, D. P.; CARREIRA, F. R. F.; MOREIRA. **Aplicativo Trigonometria 1.1.**: Trabalho Desenvolvido pelos Acadêmicos do Curso de Matemática da Universidade Federal de Rondônia – UNIR, Rondônia, 2000. Disponível em: <http://ultradownloads.uol.com.br/download/Trigonometria/baixar-gratis.html>. Acesso em: ago. 2011.

SILVA, Fernanda Félix da Silva. **O uso do roleplaying game de computador (CRPG) no ensino à distância via internet: um estudo de caso**. Dissertação (Mestrado em Educação). Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.

SILVA, F. P. R.; MOURA, T. J. M.; SANTOS, A. H. S.; RIBEIRO, M. A. B.; FILHO, GUIDO. L. S.; BRENNAND, E. G. G. **TV Escola Interativa: Uma Proposta Educativa para TV Digital**. In: VII CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA, 2004, Monterrey. RIBIE, 2004.

SKINNER, B. F. **Sobre o behaviorismo**. São Paulo: Cultrix/Edusp, 1982.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Addison Wesley, 2011.

SPINELLI, Walter. **Os objetos de aprendizagem: ação, criação e conhecimento**. 2007. Disponível em: <http://www.lapef.fe.usp.br/rived/textoscomplementares/textoImodulo5.pdf>. Acesso em: nov. 2011.

TALIGENT. **Building object-oriented frameworks**. A Taligent White Paper. 1994.

TAROUCO, Liane; FABRE, E. Marie; TAMUSIUNAS, Fabrício. **Reusabilidade de Objetos Educacionais**. RENOTE – Revista Novas Tecnologias na Educação, v.1, n.1, CINTED/UFRGS. Porto Alegre, 2003.

TELECO. Inteligência em Telecomunicações. **TV Digital I: padrões existentes**. 2010. Disponível em: www.teleco.com.br/tutoriais/tutorialtvdconsis1/pagina_3.asp. Acesso em: mar. 2012.

XLETVIEW. Free MHP Emulator. **Navigation Bar: Home e Xlet Emulator**. Disponível em: <http://www.xletview.org>. Acesso em: mar. 2012.

VÉRAS, Douglas; SILVA, Marlos; BISPO, Pedro; BRAZ, Lucas M.; BITTENCOURT, Ig Ibert; COSTA, Evandro. **Uma arquitetura para integrar ambientes educacionais na Web com Sistemas em T-Learning**. In: Simpósio Brasileiro de Informática na Educação, Florianópolis, 2009.

WALLDÉN, Sari; SORONEN, Ane. **Edutainment from television and computers to digital tv**, 2004. Disponível em: http://sbie2004.ufam.edu.br/anais_cd/extras/anaisvolI/vDigital/artigos/5544.pdf.

WELLER, M.; PEGLER, C.; MASON, R. **Putting the pieces together: what working with learning objects means for the educator**, 2003.

WILEY, David. **Learning Object Design and Sequencing Theory**. Tese (Doutorado em Filosofia). Brigham Young University, Provo, 2000, 35 f.

ZANCANARO, Airton; SANTOS, Paloma Maria; TODESCO, José Leomar. **Requisitos de um ambiente virtual de aprendizagem para TV Digital Interativa**. In: CINTED-UFRGS Novas Tecnologias na Educação. Porto Alegre, v.9, n.1, jul. 2011.

ANEXO A – FORMULÁRIO DE PESQUISA DE OPINIÃO

Formulário de pesquisa de opinião sobre avaliação dos módulos de aplicações desenvolvidas pelo *Framework Olearning TVDi*, aplicado em uma turma de vinte e seis alunos do Centro Universitário do Norte - Uninorte - Laureate do curso de Engenharia Ambiental, como ferramenta de apoio à disciplina Álgebra Linear.

Indique sua opinião para as assertivas abaixo:

1) A Ferramenta possibilita prática de conceitos aplicados em sala de aula.

() Concordo plenamente () Concordo () Indiferente () Discordo () Discordo plenamente

2) A Ferramenta oferece fácil navegação e operacionalidade.

() Concordo plenamente () Concordo () Indiferente () Discordo () Discordo plenamente

3) O uso de recursos de Objetos de Aprendizagem pela ferramenta facilita a organização e melhor compreensão de conteúdos em uma aula de trigonometria.

() Concordo plenamente () Concordo () Indiferente () Discordo () Discordo plenamente

4) A ferramenta tem aplicabilidade e é interessante usar durante o ensino médio/superior.

() Concordo plenamente () Concordo () Indiferente () Discordo () Discordo plenamente

5) A ferramenta oferece variedades de suporte de conteúdo durante uma aula de trigonometria.

() Concordo plenamente () Concordo () Indiferente () Discordo () Discordo plenamente

6) O uso desta ferramenta em um ambiente de TV Digital facilitaria a prática de estudo sobre a trigonometria.

() Concordo plenamente () Concordo () Indiferente () Discordo () Discordo plenamente

Sugestões:

Manaus, 13 de Abril de 2012.