

UNIVERSIDADE DE CAXIAS DO SUL
Centro de Ciências Exatas e Tecnologia
Curso de Bacharelado em Ciência da Computação

Rafael Buseti

MUNDUS – Um ambiente de aprendizagem instigando o conhecimento

Caxias do Sul

2009

Rafael Buseti

MUNDUS – Um ambiente de aprendizagem instigando o conhecimento

Trabalho de Conclusão de Curso
para obtenção do Grau de
Bacharel em Ciência da
Computação da Universidade de
Caxias do Sul.

Profa. Dra. Elisa Boff
Orientador

Caxias do Sul
2009

AGRADECIMENTOS

É difícil saber quem agradecer primeiro, porque não foi por uma única pessoa que cheguei aqui hoje, então eu gostaria de agradecer a todas as pessoas que me ajudaram ao longo do meu caminho.

Dedico ainda uma ressalva aos meus pais que sempre me apoiaram e me ajudaram quando eu precisei, aos meus amigos e minha namorada que me ajudaram a não perder a cabeça e sempre seguir em frente.

Agradeço ainda aos que me ajudaram a conseguir este assunto e me ajudaram na construção deste trabalho, que tanto me fascina e me envolveu. Tentando seguir uma linha lógica de acontecimentos, começo pelo João que teve que ouvir minhas ideias inquietas em busca de um assunto. Após isto aos meus amigos Diego, Matheus e Tiago que estavam lá naquele dia, nas aulas iniciais para orientação do TCC, onde toda a ideia se originou a partir de nossas conversas. Ao Caio que me ajudou a viajar nas ideias para o desenvolvimento do jogo. O Gustavo, que me ajudou na implementação indiretamente. A Gabriela Oliveira que desenvolveu toda parte visual do jogo. Por fim a Elisa que me orientou e me ajudou em todo o trabalho. De fato cada pessoa teve seu papel importante para minha chegada até aqui.

RESUMO

A utilização de jogos para a área de educação cada vez cresce mais. Porém muitos dos jogos dão muita atenção ao conteúdo e estratégias pedagógicas em detrimento a diversão que envolve um jogo. Questões lúdicas, de envolvimento e interatividade não podem ser deixadas de lado quando estamos desenvolvendo um jogo educativo.

Este trabalho buscou entender os ambientes de aprendizagem e sua relação com os jogos educativos. Através de questionários e do levantamento bibliográfico foi estruturado os requisitos do sistema, protótipos e alguns diagramas para utilização na implementação do sistema. Através destas análises foi possível traçar um ambiente educativo e interativo para desenvolvimento de um MMOG.

A implementação é baseada em diversos estudos e técnicas, para ser robusto e eficiente, visando atender a três níveis propostos, a saber, o painel gerencial, o cliente para o jogador jogar e o servidor para receber as requisições do cliente.

Palavras-chave: Jogos, MMOG, ambiente de aprendizagem, conhecimento, diversão.

ABSTRACT

The use of games to the area of education grows ever more. But many of them have many teaching approaches and forget it is still a game. Recreational issues of engagement and interactivity can not be left aside when we are developing an educational game.

This study aimed to understand the environments for learning and its relationship with educational games. Through questionnaires and the literature review was structured the system requirements, prototyping and some diagrams for use in implementing the system. Through this analysis could provide a learning environment and interactive development of a MMOG.

The implementation is based on several studies and techniques to be robust and efficient, to meet the three levels proposed are the panel management, the client for playing the player and the server to receive client requests.

Keywords: Games, MMOG, the learning environment, knowledge, fun.

SUMÁRIO

1	INTRODUÇÃO	13
1.1	QUESTÃO DE PESQUISA.....	14
1.2	ORGANIZAÇÃO DO TRABALHO.....	15
2	AMBIENTES INTERATIVOS PARA APRENDIZAGEM	17
2.1	COMPUTADOR NA EDUCAÇÃO	17
2.2	AMBIENTES DE APRENDIZAGEM	19
2.3	JOGOS EDUCATIVOS	23
2.4	JOGOS EDUCATIVOS RELACIONADOS.....	28
2.4.1	<i>Jeomm</i>	28
2.4.2	<i>Crossword Compiler</i>	28
2.4.3	<i>O que é o que é?</i>	29
2.4.4	<i>Ensino aprendizagem</i>	30
2.4.5	<i>Subindo e escorregando: Jogo para introdução do conceito de adição de números inteiros</i>	30
2.4.6	<i>L.I.S. – Learning in the Space</i>	31
2.4.7	<i>Protetores das Tartarugas</i>	32
2.4.8	<i>Diversos</i>	33
2.5	COMPARATIVO ENTRE OS JOGOS EDUCATIVOS ABORDADOS	33
3	O JOGO: MUNDUS	34
3.1	QUESTIONÁRIO PARA LEVANTAMENTO DOS REQUISITOS.....	34
3.1.1	<i>Análise: Questionários dos Alunos</i>	35
3.1.2	<i>Análise: Questionários dos Professores</i>	36
3.1.3	<i>Conclusão do Levantamento</i>	37
3.2	REQUISITOS	37
3.3	CASOS DE USO	42
3.3.1	<i>Painel: Visão Geral</i>	42
3.3.2	<i>Painel: Expansão do Caso de Uso – Gerenciar</i>	45
3.3.3	<i>Painel: Expansão do Caso de Uso – Criar Desafios</i>	48
3.3.4	<i>Jogo: Visão Geral</i>	50
3.3.5	<i>Jogo: Expansão do Casos de Uso – Mundo</i>	52
3.3.6	<i>Jogo: Expansão do Casos de Uso – Desafio</i>	55
3.4	DIAGRAMA ER	59
3.5	DIAGRAMA DE CLASSES.....	60
3.5.1	<i>Jogo</i>	61
3.5.2	<i>Painel</i>	62
3.6	USABILIDADE E ERGONOMIA	64
3.7	PROTOTIPAGEM	66
3.7.1	<i>Painel</i>	66
3.7.2	<i>Jogo</i>	74
3.8	ARQUITETURA	79
3.8.1	<i>Linguagem e Ferramenta</i>	80
3.8.2	<i>Banco de Dados</i>	82
3.8.3	<i>Comunicação Flex-Php e Flash-Php</i>	82
3.8.4	<i>Fluxo Entre Camadas</i>	83
3.8.5	<i>Tratamento de Erros</i>	84
3.8.6	<i>Arquivos</i>	84
3.8.7	<i>Organização de Pastas</i>	85
3.8.8	<i>Logs</i>	85
3.8.9	<i>Gráfico do Jogo</i>	86
3.8.10	<i>Multiplayer</i>	88

3.8.11	<i>Análise de Middlewares para o Servidor</i>	89
3.8.12	<i>Conclusão dos Middlewares</i>	95
4	DESENVOLVIMENTO	97
4.1	PAINEL ADMINISTRATIVO	97
4.1.1	<i>Usabilidade</i>	97
4.1.2	<i>Comunicação AMF</i>	98
4.1.3	<i>Requisições</i>	98
4.1.4	<i>Relatórios</i>	99
4.1.5	<i>Respostas do Professor</i>	100
4.1.6	<i>Segurança</i>	100
4.1.7	<i>Skins</i>	101
4.1.8	<i>Limitações e Problemas Enfrentados</i>	101
4.2	SERVIDOR	101
4.2.1	<i>Escolha do middleware</i>	103
4.2.2	<i>Desafios</i>	104
4.2.3	<i>Mundo</i>	108
4.2.4	<i>Segurança</i>	112
4.2.5	<i>Painel de Gerenciamento</i>	112
4.2.6	<i>Limitações e Problemas Enfrentados</i>	114
4.3	CLIENTE	115
5	CONFIGURANDO O AMBIENTE	128
5.1	CONFIGURANDO UM SERVIDOR WEB COM BANCO DE DADOS DEFAULT	128
5.2	PAINEL E BANCO DE DADOS.....	128
5.3	CLIENTE	129
5.4	SERVIDOR.....	130
6	CONCLUSÃO	132
6.1	TRABALHOS FUTUROS	133
7	REFERÊNCIAS	135

LISTA DE ILUSTRAÇÕES

FIGURA 1: APRENDIZAGEM SIGNIFICATIVA.....	22
FIGURA 2: CROSSWORD COMPILER.....	29
FIGURA 3: O QUE É O QUE É?.....	29
FIGURA 4: JOGO PERGUNTAS E RESPOSTAS E ENIGMA.....	31
FIGURA 5: L.I.S.....	32
FIGURA 6: PROTETORES DE TARTARUGAS.....	32
FIGURA 7: PALAVRAS RELACIONADAS AO JOGO.....	38
FIGURA 8: FLUXO DA DIVERSÃO.....	41
FIGURA 9: CASO DE USO – PAINEL – VISÃO GERAL.....	42
FIGURA 10: CASO DE USO – PAINEL – EXPANSÃO – GERENCIAR.....	45
FIGURA 11: CASO DE USO – PAINEL – EXPANSÃO – CRIAR DESAFIOS.....	48
FIGURA 12: CASO DE USO – JOGO – VISÃO GERAL.....	51
FIGURA 13: CASO DE USO – JOGO – EXPANSÃO – MUNDO.....	52
FIGURA 14: CASO DE USO – PAINEL – EXPANSÃO – DESAFIO.....	55
FIGURA 15: DIAGRAMA ER.....	60
FIGURA 16: DIAGRAMA DE CLASSES – JOGO.....	62
FIGURA 17: DIAGRAMA DE CLASSES – PAINEL.....	63
FIGURA 18: PROTÓTIPO PAINEL – LOGIN.....	67
FIGURA 19: PROTÓTIPO PAINEL – ESTRUTURA BASE.....	67
FIGURA 20: PROTÓTIPO PAINEL – CRIAR DESAFIO – INFORMAÇÕES.....	68
FIGURA 21: PROTÓTIPO PAINEL – CRIAR DESAFIO – PERGUNTAS.....	68
FIGURA 22: PROTÓTIPO PAINEL – PERGUNTAS – CADASTRO.....	69
FIGURA 23: PROTÓTIPO PAINEL – CRIAR DESAFIO – CONTEÚDOS RELACIONADOS.....	70
FIGURA 24: PROTÓTIPO PAINEL – CONTEÚDO RELACIONADO – CADASTRO.....	71

FIGURA 25: PROTÓTIPO PAINEL – RESPONDER DESAFIO - LISTAGEM.	71
FIGURA 26: PROTÓTIPO PAINEL – RESPONDER DESAFIO - CADASTRO.	72
FIGURA 27: PROTÓTIPO PAINEL – RELATÓRIO – LISTAGEM PROFESSOR.	73
FIGURA 28: PROTÓTIPO PAINEL – RELATÓRIO – INFORMAÇÕES.	73
FIGURA 29: PROTÓTIPO JOGO – LOGIN.	74
FIGURA 30: PROTÓTIPO JOGO – CRIAÇÃO DO PERSONAGEM.	74
FIGURA 31: PROTÓTIPO JOGO – ESTRUTURA.	75
FIGURA 32: PROTÓTIPO JOGO – MAPA.	76
FIGURA 33: PROTÓTIPO JOGO – SESSÃO.	76
FIGURA 34: PROTÓTIPO JOGO – DESAFIO.	77
FIGURA 35: PROTÓTIPO JOGO – CONTEÚDO RELACIONADO.	78
FIGURA 36: PROTÓTIPO JOGO – FEEDBACK.	78
FIGURA 37: PROTÓTIPO JOGO – RANKING	79
FIGURA 38: ESTRUTURA BÁSICA DA APLICAÇÃO.	80
FIGURA 39: COMPARATIVO DE TECNOLOGIAS.	81
FIGURA 40: EXEMPLO DE REPRESENTAÇÃO ISOMÉTRICA NO JOGO DIABLO I.	87
FIGURA 41: REPRESENTAÇÃO DO MAPA DIAMOND.	87
FIGURA 42: FUNCIONAMENTO DO SMARTFOXPRO.	91
FIGURA 43: ARQUITETURA SMARTFOXPRO.	92
FIGURA 44 – 9 BLOCOS CARREGADOS, 4 BLOCOS CARREGADOS	109
FIGURA 45 – TELA INICIAL DO PAINEL DE GERENCIAMENTO DO SMARTFOXSERVER.	113
FIGURA 46 – TELA DAS ZONAS NO PAINEL	114
FIGURA 47 – REPRESENTAÇÃO DE UM DOCUMENT, PAGE E CONTENT	116
FIGURA 48 – REPRESENTAÇÃO DAS CAMADAS PRESENTES NO FRAMEWORK.	117
FIGURA 49 – EXEMPLO DE UM SPRITE SHEET, RETIRADO DO LIVRO ACTIONSCRIPT FOR MULTIPLAYER GAMES AND VIRTUAL WORLDS (MAKAR, 2009)	118
FIGURA 50 – REPRESENTAÇÃO DO GRID COM SUAS RESPECTIVAS POSIÇÕES	124
FIGURA 51 – EXEMPLO REPRESENTANDO UMA IMAGEM COM UM GRID SEM SORTING E OUTRO GRID COM SORTING.	124

LISTA DE TABELAS

TABELA 1: RELACIONAMENTO ENTRE OS JOGOS	33
TABELA 2: TEMPO DE ENVIO DOS DADOS	82
TABELA 3: PAINEL	85
TABELA 4: JOGO	85
TABELA 5: LOGS	86
TABELA 6: COMPARATIVO DAS ANÁLISES DOS <i>MIDDLEWARES</i>	95

LISTA DE ABREVIATURAS E SIGLAS

Sigla	Significado em Português	Significado em Inglês
AMF3	Formato de Mensagem Actionscript	Actionscript Message Format
API	Interface de Programação de Aplicativos	Application Programming Interface
AS2	Actionscript 2	Actionscript 2
AS3	Actionscript 3	Actionscript 3
CAI	Instrução Com Auxílio do Computador	Computer Assisted Instruction System
ECMA-262	Associação Europeia de Fabricantes de Computadores	European Computer Manufacturers Association 262
ES4 - Electro Server 4	Electro Server 4	Electro Server 4
ER	Entidade Relacionamento	Entity Relationship
GPL 2	Licença Pública Geral	GNU General Public License
HTTP	Protocolo de Transferência de Hipertexto	Hypertext Transfer Protocol
IA	inteligência artificial	Artificial Intelligence
ISO	Organização Internacional para Padronização	International Organization for Standardization
JSON	Notação do Objeto Javascript	Javascript Object Notation
MMOG	Jogos multiusuários online de uso em massa	Massively Multiplayer Online Game
NPC	Personagem não jogador	Non-player character
PDO	Objeto de Dados PHP	PHP Data Object
PHP	PHP: Hypertext Preprocessor	PHP: Hypertext Preprocessor
RIA	Aplicativos Ricos para Internet	Rich Internet applications
RPC	Chamada Remota de	Remote Procedure Call

Procedimento

RPG	jogo de interpretação de Personagens	Role-playing game
RSS	Acesso Fácil a Notícias	Really Simple Syndication
STI	Sistema Tutor Inteligente	Tutoring Systems
TCP	Protocolo de Controle de Transmissão	Transmission Control Protocol
UDP	Protocolo de Datagrama de Usuário	User Datagram Protocol
WEB	Rede de alcance mundial	World Wide Web
XML	Linguagem De Marcação Extensível	Extensible Markup Language
ZDP	Zona desenvolvimento proximal	Zone of proximal development

1 INTRODUÇÃO

Podemos perceber facilmente que cada vez mais os computadores estão inseridos no nosso dia a dia. A sociedade vive hoje a era da informação, que faz com que as pessoas tenham um canal direto com diferentes tipos de informação e abrindo assim novos horizontes para a educação.

Segundo Fellers (1996) existem mudanças significativas na educação. Entramos em um novo paradigma educacional, onde o conhecimento é construído de uma forma coletiva. A aprendizagem, segundo Fellers (1996), é vista como cooperativa e não mais como individual. O aluno é instigado a buscar o conhecimento, não sendo apenas mais um processo de ordem do professor. O conceito de cooperação, segundo Barros (1994), é a ação coletiva visando um mesmo objetivo, onde a colaboração está incluída na cooperação.

Percebendo esta quebra na educação, novas ferramentas devem ser abordadas para suprir este novo mundo. Visando este aspecto, através da união do conceito de ambiente de aprendizagem com a utilização de jogos temos uma forte ferramenta de ensino-aprendizagem. Os jogos podem ir além da diversão e do entretenimento, quando trabalhados junto à educação. Eles se tornam uma ferramenta poderosa capaz de envolver o jogador, desenvolvendo então suas habilidades e seus conhecimentos, independente da idade das pessoas que jogam.

Bonjiolo (1998) já dizia em seu trabalho, “a fórmula computador mais jogo se torna perfeita, pois associa a riqueza dos jogos educativos com o poder de atração dos computadores. E, como consequência desta associação, teremos os jogos educativos computadorizados, onde o computador será usado de forma lúdica e prazerosa, para explorar um determinado ramo de conhecimento, além de trabalhar com algumas habilidades, como, por exemplo, destreza, associação de ideias e raciocínio lógico e indutivo, entre outras”.

Porém para conseguirmos abstrair ao máximo o poder de um ambiente de aprendizagem, diversas características devem ser levadas em consideração e ainda quando falamos que será através de um jogo, muitas outras abordagens devem ser discutidas para conseguir traçar um caminho onde os jogadores consigam utilizar bem os recursos do sistema e que eles sejam convertidos em conhecimento.

Com base nas ideias acima expostas, este trabalho tem o intuito de desenvolver um ambiente de aprendizagem, modelado e representado através de um jogo. Diversos jogos já

foram desenvolvidos com o objetivo de ensinar o aluno, porém diversos fatores dificultam a utilização desses sistemas educativos, alguns sendo executados apenas localmente, dificultado o seu acesso. Outros, jogados por apenas uma pessoa, entre outros diversos fatores.

No momento em que queremos atender o maior número de jogadores e com uma ferramenta capaz de instigar o aluno a jogar mais, trazendo vários jogadores para um mesmo espaço, que possam interagir entre si, conversando, trocando ideias e experiências, começamos a configurar os cenários que um jogo deve contemplar. O tipo de jogo que será construído é o MMOG, que sua abreviação significa *Massively Multiplayer Online Game*, que em português significa Jogo Multiusuário Online de uso em massa.

Este tipo de jogo tem como principal característica um mundo virtual persistente onde os jogadores podem interagir entre si. Este mundo existe independente de usuários estarem conectados nele ou não. Os jogadores criam um avatar, acessam o mundo e podem vagar pelo mapa, conversar com outras pessoas, interagir com objetos, entre outras coisas. Uma referencia interessante para este tipo de jogo é o Second Life da Linden Lab, ele é um MMOG com gráficos 3D, diferente do jogo que será desenvolvido com gráficos 2D do tipo isométrico.

Assim, um dos principais objetivos do trabalho é desenvolver um jogo MMOG web que o professor tenha total liberdade para criar novos desafios aos jogadores e ainda poder consultar o andamento através de relatórios através de um painel.

Outro objetivo deste trabalho é fornecer diferentes abordagens para que o jogador seja estimulado, tanto na questão individual, como na cooperativa. O jogo deverá fornecer um avatar para diferenciação de cada jogador e deverão ser estudadas técnicas e estratégias para instigá-lo a jogar e aprender mais, motivar, desenvolver a cognição.

Também faz parte do processo de desenvolvimento do jogo um aprofundamento das bases teóricas relacionadas e a definição de formas de coleta de requisitos. As bases teóricas se referem tanto a questões pedagógicas, psicológicas, como usabilidade de interfaces. Considerando estes aspectos, o ambiente modelado poderá dar suporte a aprendizagem, onde os alunos possam aprender de forma lúdica e os professores consigam orientar a aprendizagem do aluno definindo diferentes caminhos.

1.1 Questão de Pesquisa

Visando os objetivos, foi definida a seguinte questão de pesquisa:

Quais são as características que devem ser incorporadas em um MMOG para que ele possa ser usado como uma ferramenta de aprendizagem tanto para o professor quanto para o aluno?

Relacionado a esta questão, foi levantada a seguinte hipótese:

1. Para que um MMOG possa ser considerado educativo, o mundo deve fornecer ao aluno jogador diferentes formas de construção do conhecimento. Estas formas ou caminhos serão definidos pelo professor, pois consideramos que ele assume o papel de gestor do conhecimento. Além disso, o ambiente deve fornecer painéis de cadastro de conteúdos para os professores e *feedbacks* das interações com os desafios.

1.2 Organização do Trabalho

Este documento está organizado da seguinte forma: o capítulo 2 apresenta as questões pertinentes para o desenvolvimento de um jogo educativo. O início do capítulo aborda a educação voltada a informática, após esta abordagem é introduzido os conceitos de ambientes de aprendizagem e suas características. O próximo assunto fala sobre os jogos educativos, caracterizando como eles são e seus tipos e então finalmente o capítulo é concluído com jogos educativos relacionados.

O capítulo 3 aborda a construção do jogo onde, inicialmente é apresentado o resultado dos questionários e então desenvolvido todos os requisitos principais do jogo com base do questionário e levantamento teórico abordado no capítulo 2.

Através dos requisitos, a sequencia do capítulo é constituída dos diagramas de casos de uso, ER e diagrama de classes. Após isto é feita uma breve introdução sobre usabilidade e ergonomia e então são apresentados os protótipos do painel administrativo e do jogo.

Ainda no capítulo 3 é apresentado toda a estrutura da arquitetura do jogo, descrevendo o uso das tecnologias, ferramentas, banco de dados, comunicações entre as camadas, arquivos, organização de pastas, logs, gráfico do jogo. No final desta sessão é apresentado um estudo sobre *middlewares* para construir um jogo *multiplayer* (com vários jogadores).

No capítulo 4 é abordado sobre o desenvolvimento do jogo, explicando fluxos de funcionamento de cada sessão (painel, cliente e servidor), como foi implementado as principais características e técnicas utilizadas. É listada toda a requisição feita entre as camadas de persistências e consultas dos dados. São abordadas as dificuldades e limitações no momento da implementação do projeto e como foram solucionadas.

O capítulo 5 aborda sobre a configuração de um ambiente num todo para rodar a aplicação e para editar o código implementado.

O capítulo 6 apresenta as conclusões baseadas em todo desenvolvimento do trabalho e possibilidade de trabalhos futuros.

2 AMBIENTES INTERATIVOS PARA APRENDIZAGEM

Para a construção de um ambiente interativo visando a aprendizagem precisamos entender como a educação está inserida na computação e traçar as características necessárias que um ambiente deve ter para atender tanto a questão interativa tendendo aos jogos quanto as questões de aprendizado através de ambientes de aprendizagem.

O início deste capítulo introduz o computador na educação, após isto é apresentado um estudo acerca dos ambientes de aprendizagem e por fim são abordados os jogos e exemplos de jogos relacionados.

2.1 Computador na educação

Na busca de um pouco da história da união da educação com a computação pode-se perceber que no início a máquina não passava de uma forma de ensinar, sem que o aluno pudesse construir o seu próprio caminho para o conhecimento.

Inicialmente em 1924, S. Pressey inventou uma máquina para corrigir testes de escolha múltipla, porém isto não era suficiente. Trazendo mais perto da realidade, em 1950, B. F. Skinner propôs uma máquina para ensinar usando o conceito de instrução programada, também conhecida como CAI (instrução com auxílio do computador). A instrução programada se refere a forma de apresentação do conteúdo em diversas questões estruturadas em uma sequência lógica para o aluno responder. Estas questões são respondidas pelo aluno através da marcação das respostas corretas. Podemos caracterizar um jogo computacional de *Quiz* como um exemplo de instrução programada.

Com o passar dos anos os sistemas CAIs precisavam de um sucessor, já que eram impostas diversas limitações. Então surgiu o Sistema Tutor Inteligente (STI), que através da Inteligência Artificial (IA), consegue manter um modelo de aluno e ampliar a interação com o aluno.

Os sistemas geralmente conhecidos dentro do STI são os tutores, que auxiliam o aluno durante o desenvolvimento de alguma ação, medindo se é necessário ou não sua intervenção.

Os auxílios podem ser de diversos tipos, podem indicar quais são os problemas para o aluno tentar resolver, podem fazer alguma explicação mais ampla sobre o assunto, sempre com o objetivo de auxiliar e instruir o aluno.

Com a evolução da educação junto à computação, a utilização do computador vem provocando uma mudança no paradigma pedagógico, alguns consideram a computação como um catalisador para uma mudança no paradigma educacional.

O computador pode ser visto como uma máquina de ensino ou como uma ferramenta. Quando falamos que o computador está “ensinando” uma pessoa, estamos falando sobre o paradigma instrucionista.

Este paradigma foi fortemente influenciado por Skinner, através de sua teoria psicológica behaviorista. Todo o conhecimento pronto e hierarquizado poderia ser passado para o aluno.

Quando estamos falando da informática junto a este modelo, ele pode ser tanto no paradigma instrucionista como construtivista. O paradigma define a forma como o software será projetado e como serão apresentadas as atividades no mesmo.

Segundo Papert (1986) o construtivismo é a abordagem que o aluno constrói o seu próprio conhecimento através do computador. Nesta construção do conhecimento o aluno se envolve afetivamente, pois é de seu interesse, tornando o aprendizado mais rico.

No trabalho visto de Valente (1993) ele consegue descrever claramente como o computador pode ser utilizado como uma ferramenta, através da utilização do programa Logo. Nesta ferramenta o aluno tem o poder, através de comandos, de digitar qual são as ações que ele deseja que uma tartaruga faça. Conforme os códigos são digitadas pelos alunos, as ações são replicadas visualmente na tela do computador. No momento que as ações são visualizadas na tela, o aluno entra num processo de reflexão sobre o que ele fez e assim provocando alterações na estrutura mental de acordo com Piaget (Piaget, 1977 e Manton, 1991).

Essa alteração na estrutura mental é o momento pelo qual o aluno reorganiza todo o conhecimento adquirido com a interação, transformando em conhecimentos prévios, onde são suas próprias ideias. O construtivismo defende a ideia de que os erros ajudam a entender as ações.

Percebemos então que o controle do ensino é colocado na mão do aluno, onde a educação já não é mais apenas uma transferência de conhecimento, mas sim um processo de construção de conhecimento do próprio aluno.

Jonassen (1996) descreve bem a diferença entre o instrucionismo e o construtivismo. “O construtivismo é uma filosofia de aprendizagem que descreve o que significa saber alguma coisa e o que é a realidade. As concepções tradicionais de aprendizagem admitem que o conhecimento é um objeto, algo que pode ser transmitido do professor para o aluno. Esta

concepção presume que o conhecimento é algo que pode ser adquirido, como suprimentos comprados num supermercado. Os construtivistas, por outro lado, acreditam que o conhecimento é uma construção humana de significados que procura fazer sentido do seu mundo. Os seres humanos são observadores e intérpretes naturais do mundo físico. A fim de realizar isto, eles explicam ideias e fenômenos novos nos termos do conhecimento existente”.

No momento que é dada a liberdade para uma pessoa fazer suas escolhas e buscar o próprio conhecimento, pode-se perceber que o aprendizado se torna muito mais rico, pois ele está envolvido. Quando falamos em jogos, a primeira ideia que passa para as pessoas é a diversão que eles podem trazer.

A diversão pode ser vista como uma forte aliada da educação, já que em momentos que as pessoas estão se divertindo, elas querem mais e mais. Levando isto em consideração podemos dizer que se o jogo é divertido e está ensinando, provavelmente a pessoa vai ser instigada a jogar cada vez mais a jogar e, conseqüentemente, desenvolver cada vez mais seu intelecto.

2.2 Ambientes de Aprendizagem

Para falarmos sobre os ambientes de aprendizagem precisamos definir o que é aprendizagem. Segundo Piaget (1983), “o conhecimento não procede, em suas origens, nem de um sujeito consciente de si mesmo nem de objetos já constituídos que a ele se imporiam. O conhecimento resultaria de interações que se produzem a meio caminho entre os dois”.

Para Baranauskas (1999) “o aprendizado é entendido como a construção individual do conhecimento a partir de atividades de exploração, investigação e descoberta...”.

Logo, a ação que existe sobre o objeto é o que constrói o conhecimento (Jonassen, 1996). Esta interação está ligada diretamente à afetividade e a motivação. Segundo Piaget (1983) a afetividade exerce influência direta no desenvolvimento intelectual, então um ambiente de aprendizagem deve possuir motivação e um nível de afetividade com o aluno.

Piaget (1983) definiu que o conhecimento é fundamentado na ideia de equilíbrio e desequilíbrio. Quando uma pessoa entra em contato com um novo conhecimento, surge naquele momento o desequilíbrio e então a pessoa começa assimilar este novo conhecimento, através da interação. Chega um momento que a pessoa volta novamente ao equilíbrio.

Se a pessoa não consegue chegar a um equilíbrio pode ser por três motivos:

O primeiro motivo é quando a pessoa não tem estrutura suficiente para conhecer determinado assunto e então o ambiente que a pessoa está deve ser melhorado proporcionando a aprendizagem.

Em segundo se a pessoa possui estruturas que estão ainda em andamento, o professor deve dar o auxílio necessário para o entendimento do conteúdo. E o terceiro motivo é quando a pessoa possui já uma estrutura formada, porém não está conseguindo aprender, neste momento o professor deve intervir para mostrar o erro a pessoa e muitas vezes alterar a metodologia.

No momento em que referenciamos a forma pelo qual o conhecimento é formado para Piaget, não podemos deixar de abordar os trabalhos construídos por Vygotsky. Segundo Vygotsky (2003) todas as pessoas possuem uma zona de desenvolvimento real que é o conhecimento que a pessoa detém. Todo o resto que não conhecemos constitui a nossa Zona de Desenvolvimento Proximal (ZDP), definido como “a distância entre o nível de desenvolvimento real, que se costuma determinar através da solução independente de problemas, e o nível de desenvolvimento potencial, determinado através da solução de problemas sob a orientação de um adulto ou em colaboração com companheiros mais capazes (Vygotsky, 2003)”.

Um ambiente de aprendizagem pode tomar como base os princípios de Piaget e Vygotsky, não esquecendo que segundo as ideias de Vygotsky, o ambiente deve ser cooperativo, já que a colaboração entre mais de uma pessoa ajuda a desenvolver estratégias e habilidades gerais de solução de problemas pelo processo cognitivo, evoluindo então a ZDP.

Antes de abordar os fundamentos principais de um ambiente de aprendizagem devemos entender o que significa a palavra cooperação e colaboração. Apesar de muitos acharem que é a mesma coisa, Barros (1994) defende a distinção significativa entre as duas palavras. Segundo Barros (1994) colaboração está relacionada diretamente a contribuição. Já cooperação engloba colaboração, a interação e o trabalho coletivo para atingir um mesmo objetivo.

Mantovani (1996) e Jonassen (1996) abordam que no momento que existe um ambiente cooperativo, onde os alunos podem interagir entre si, existe uma troca de ideias, socialização e podem até compartilhar conhecimento. No momento ainda que temos esta socialização entre alunos, existem diferentes pontos de vista, estabelecendo diferenças em suas perspectivas. Logo, o ambiente favorece o confronto e desenvolve a geração dos conflitos sócio-cognitivos (desequilíbrio e equilíbrio segundo Piaget, desenvolvimento da ZDP para Vygotsky).

O aluno passa do papel que uma vez era imposto para ele, onde o aprendizado não é passado mais para ele diretamente, mas sim ele mesmo o constrói. Assim é criado todo um ambiente, onde através dos recursos que ele fornece, o aluno poderá seguir seus passos e construir seu conhecimento.

Porém para que tudo isso seja possível existem fundamentos cruciais para o funcionamento de um ambiente de aprendizagem. Barros diz que no momento em que se deseja o desenvolvimento cognitivo em ambientes, criar situações que estimulem o aluno é um dos grandes fatores de sucesso. Outro fator crucial é a ênfase na superação dos conflitos ou desequilíbrios, conforme a teoria piagetista.

Os ambientes de aprendizagem possuem diversas características, aos quais Barros (1994), Baranauskas (1999) e Freitas (2006) conseguem descrever bem:

- O conteúdo é estruturado em atividades que possam ser mensurados, onde os alunos podem ter um controle parcial da interação com o ambiente. O aluno controla o processo de aprendizagem.
- Deve existir uma transmissão de informação.
- Existe um responsável, que controlaria o ambiente para relacionar os processos internos e externos e vice-versa.
- Feedback é importante, deve ser gerado pela interação do aluno com o ambiente.
- Cobrir cada fase do aprendizado significativo (Figura 1), conforme Jonassen (1996).
- Enfatiza a assimilação das diversas informações e habilidades específicas.
- Individualização é determinada pelo aluno, não pela ferramenta. Isto para ambientes interativos.
- Organização do conteúdo para serem objetivos e conseguirem um resultado melhor no aprendizado dos alunos.
- Os alunos devem seguir alguns critérios para que haja harmonia dentro do ambiente.
- Os conteúdos abordados neste ambiente não devem ser fornecidos diretamente ao aluno, ele deve por si só buscar, formular suas hipóteses e validar.

Existem mais algumas características ainda abordadas por Barros que instigam os alunos nos ambientes de aprendizagem como o fato de criar desafios, novas descobertas, proporcionando um mundo maleável aos caminhos seguidos pelo aluno.

Conforme dito nas características dos ambientes de aprendizagem, Jonassen (1996) aborda a aprendizagem significativa, que de certa forma está ligada com as características que

um ambiente de aprendizagem deve seguir, como a pessoa pode se comportar e que soluções podem ser tomadas para tornar o ambiente mais interessante (Figura 1).

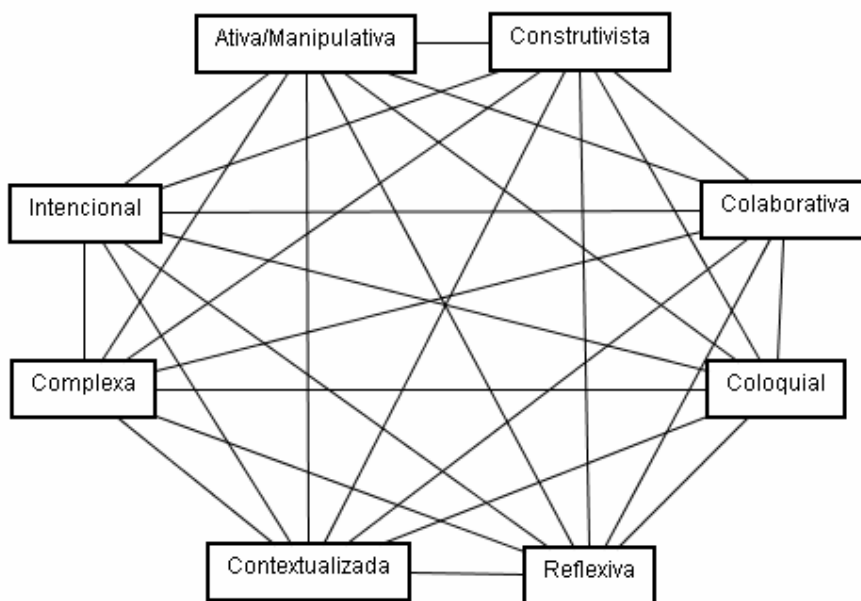


Figura 1: Aprendizagem significativa.

Os ambientes de aprendizagem podem ser:

- **Ativa:** Os comprometimentos em atividades essenciais garantem o envolvimento do aluno no ambiente, fornecendo a estrutura sólida necessária para o aprendizado.
- **Construtivista:** Os alunos conseguem formar novas ideias para suas experiências ao longo de sua caminhada.
- **Reflexiva:** Através da reflexão de suas próprias ações e de suas análises o aluno consegue entender mais e demonstrar melhor o conhecimento que adquiriu, como por exemplo, em momentos que ele está respondendo alguma questão e já possui um feedback imediato para entender o motivo do acerto ou erro.
- **Colaborativa:** Construção do conhecimento de forma colaborativa, onde grupos são formados, para que exista uma interação entre as pessoas do grupo.
- **Intencional:** As ações que são desenvolvidas possuem um objetivo. No momento em que esses objetivos são de intenção do aluno e ele realmente quer aquilo, o aluno pensa mais e aprende mais, logo o ambiente deve propiciar este cenário ao

aluno.

- **Complexa:** Transformar a realidade em algo simples muitas vezes não traz grandes benefícios ao aluno, as resoluções simples de algumas ações muitas vezes trazem uma ideia falsa para o aluno.
- **Contextual:** No momento em que o mundo real começa a ser empregado dentro do ambiente, retirando as formas abstratas que são decoradas pelas pessoas, o conhecimento ganha um impulso maior.
- **Coloquial:** As resoluções de problemas geralmente cabem a uma atividade social, fornecer uma forma que ajude a discussão entre as pessoas é crucial.

Existem diversas variáveis cognitivas que devem ser promovidas no ambiente. O aluno deve ter um conhecimento prévio do objeto de estudo, ou a tarefa. Os conteúdos devem criar conflitos cognitivos para o aluno se equilibrar novamente e absorver o conhecimento.

Um ambiente onde seja possível testar hipóteses e que na vida real é difícil replicar torna-se um meio importante para novos pensamentos. As tarefas propostas pelo ambiente devem mensurar a capacidade do aluno, para que ele não desista facilmente da tarefa. Quanto mais criatividade e formas de abordar os desafios, mais envolvidos os alunos ficarão.

2.3 Jogos Educativos

Os jogos educativos hoje vem se caracterizando pela palavra *edutainment* (*Education + Entertainment*) ou ludoinformação. Significa associar a educação com o entretenimento, criar experiências educativas que sejam divertidas.

Para Battaiola (2000) os jogos educativos nada mais são que uma modalidade dos jogos computadorizados e estes tipos de jogos apresentam uma proposta pedagógica explícita, onde tem como principal objetivo ensinar de forma prazerosa.

Quando falamos em jogos educativos, conseqüentemente estamos abordando sobre ambientes de aprendizagem, onde o jogo em si é apenas um dos meios pelo qual pode ser construído um ambiente educativo para o aluno vivenciar experiências e evoluir seu conhecimento.

Campos (1995) diz que os jogos educativos conseguem despertar no aluno diversos aspectos: motivação, curiosidade, estímulo, envolvimento no fato de querer aprender e cabe ao professor utilizar esses aspectos para criar estratégias pedagógicas. Assim o aluno consegue enriquecer seu conhecimento de forma divertida.

Piaget defende a ideia que o jogo pode ajudar no desenvolvimento do intelecto da criança, pois possui uma forte influência como instrumento incentivador e motivador no ensino.

Para Rizzo (1988) e Amory (2001), os jogos são um poderoso recurso que estimulam o desenvolvimento do aluno. Desenvolvem não apenas o cognitivo, mas a atenção, autocontrole, disciplina, habilidades motoras dependendo dos recursos do jogo. Os jogos ainda podem ser jogados de forma individual ou coletiva e os professores são os encarregados de propiciar todo o desenvolvimento, observar e avaliar os alunos e se as metodologias empregadas através do jogo estão efetivamente funcionando.

O jogo em si traz consigo o significado de diversão, no momento em que conseguimos unir o jogo com a educação, percebemos que não estamos falando apenas da formação do conhecimento de um aluno, mas sim instigar o aluno a buscar o conhecimento, a querer mais, a se socializar com outros colegas, ser motivado a desenvolver continuamente as tarefas propostas e no momento em que esses objetivos são alcançados, podemos ter certeza que um bom profissional está se formando. Amory (2001) aborda que para os jogos educativos conseguirem uma aceitação dos seus jogadores, eles devem apresentar enredos atraentes.

Segundo Silveira (1998) “os jogos educativos são ambientes nos quais o aluno aprende através da descoberta. Estes ambientes de aprendizagem possuem um enfoque heurístico, não fornecendo o conteúdo diretamente ao aluno. O que o aluno vai aprender, as respostas para alcançar o objetivo, não são pré-determinados. Os jogos educativos utilizam-se do cognitivismo, o aluno aprende através da busca, da descoberta e do raciocínio”.

Para que tudo isto seja possível, existem características cruciais para o desenvolvimento de um bom jogo educativo. Malone (1981), Mantovani (1996), Amory (2001) e Clua (2002) descreveram algumas dessas características:

- Estimular a imaginação e condutas cognitivas (raciocínio, associação de ideias).
- Desenvolvimento de habilidades funcionais: aplicação de regras, destreza, força, rapidez e a concentração.
- Desenvolver um ambiente onde possa existir cooperação entre os jogadores, estimulando atividades sociais.
- Ajudar no desenvolvimento de condutas afetivas: auto-estima, autonomia e iniciativa;
- Os jogos devem ser objetivos.
- Os resultados podem ser diversos, onde não existe uma linha lógica para seguir no jogo, mas sim que os jogadores possam escolher a forma que ele quer seguir.

- Oferecer formas para que possa ser medido o sucesso dos jogadores, até mesmo para eles saberem se eles estão evoluindo ou não dentro do jogo.
- Feedback claro na performance do jogador.
- Os jogos devem ser divertidos, prazerosos e motivadores, deve ter ênfase no lúdico;
- O *design* do jogo deve ser bem trabalhado para fazer com que o jogo educativo, apesar de um forte apelo pedagógico e seja ainda um jogo e se ele não apresentar um gráfico bem desenvolvido, ninguém irá imergir no jogo.
- Apesar de muitos jogos de sucesso não terem enredos, ele não prejudica em nada o desenvolvimento do jogo, é importante que o jogo tenha uma história onde possa envolver os jogadores.
- Personagens bem elaborados no contexto do jogo.

Foi realizada uma pesquisa entre jovens de 10 e 17 anos, de classe média na cidade do Rio de Janeiro, onde foi verificado o que torna atrativo os jogos e o que pensam sobre jogos educativos. Para 85% das pessoas o desafio é o que torna mais atrativos os jogos, histórias ricas, com qualidade visual e bastante IA. Dentre estes jovens 68% consideram jogos educativos ruins, onde o principal problema é que apresenta um baixo nível de desafios motivadores, imersão no jogo e gráficos ruins. (Clua apud Bittencourt, 2002).

Os jogos devem fornecer ainda um ambiente pelo qual o jogador consiga se interessar. A construção desse ambiente esta ligada diretamente ao visual, no momento em que definimos uma interface limpa e prazerosa, o jogador não ficará cansado de estar visualizando diferentes partes do jogo. A questão de usabilidade é importante e será abordada mais para frente do trabalho.

A utilização de recursos multimídias é outro fator crucial nos tempos de hoje, um jogo caracterizado apenas por textos pode não envolver o jogador, no momento em que oferecemos vídeos e áudio, novos recursos para estimular o aprendizado são engajados.

"Se o ensino for lúdico e desafiador, a aprendizagem prolonga-se fora da sala de aula, fora da escola, pelo cotidiano, até as férias, num crescendo muito mais rico do que algumas informações que o aluno decora porque vão cair na prova" (Neto, 1992).

O contato com uma simples ferramenta de educação pode parecer sem graça muitas vezes, fria, até desumana por ser uma máquina racional sem sentimentos, levando isto em conta a utilização de jogos como um canal de comunicação com a pessoa se torna muito atrativo, mascarando para muitas pessoas toda a insegurança que elas podem ter consigo.

Valente diz que “A interação aluno-computador precisa ser mediada por um profissional que conhece Logo, tanto do ponto de vista computacional, quanto do pedagógico e do psicológico. Esse é o papel do mediador no ambiente Logo”. Esta frase tem como objetivo dizer que independente da ferramenta utilizada, um profissional capacitado deve conduzir os alunos, visando isto à utilização de um professor no jogo de desafios é crucial. O professor terá poder para alimentar toda a base de conhecimentos do jogo e interagir com os demais jogadores para o bom andamento do grupo.

Percebemos que não há mais o ensino direto do professor, mas sim a criação de um ambiente de aprendizagem que propicia ao aluno aprender sobre diferentes assuntos, facilitando o desenvolvimento intelectual do aluno.

Há uma classificação de tipos de jogos, vale ressaltar que muitas vezes um jogo pode estar engajado em mais de um tema, segundo Battaiola (2001) os tipos são:

- **Estratégia:** O jogador tem como principal objetivo a tomada de decisões. A partir das decisões tomadas o ambiente e os outros jogadores poderão se comportar de diferentes maneiras. Exemplos destes tipos de jogo é o Simcity, Warcraft, Starcraft e Age of Empires.
- **Simuladores:** A maioria dos simuladores é em primeira pessoa e envolvem conceitos de física do ambiente. Dentro desta categoria estão os simuladores de avião como o *Flight Simulator*, carros como *Nascar*, vida como *The Sims* e *Second Life*.
- **Aventura:** Estes jogos têm como objetivo passar de fase, para isto muitas vezes é preciso solucionar enigmas, confrontar inimigos e etc. Exemplos deste jogo são Pitfall, Tomb Raider, Sonic, Mario.
- **Infantil:** Estes jogos são caracterizados assim porque o público alvo deles são as crianças, que tem como objetivo ensinar as crianças e diverti-las.
- **Passatempo:** São jogos simples, como jogos de carta e tabuleiro.
- **RPG:** São jogos adaptados do RPG (role-playing game) de mesa. O jogador possui um personagem, fazendo parte de uma aventura imaginária.
- **Esporte:** São jogos esportivos.
- **Educativos:** São jogos que tem um apelo para o lado pedagógico.

O estilo do jogo pode influenciar diretamente na forma em que o conhecimento é passado ao aluno. No momento em que temos um mundo virtual persistente, onde diversas pessoas podem acessar e interagir entre si, logo abrimos um universo de resultados.

Neste mundo virtual, o jogador possui um avatar, uma espécie de personagem virtual. Dependendo de como o mundo é construído, o jogador tem a possibilidade de fazer diversas

ações, evoluir suas habilidades, adquirir itens, como por exemplo, roupas e até mesmo alterar suas características, competir com outros jogadores.

Vale ressaltar que um jogo do estilo MMOG estimula uma série de habilidades cognitivas do jogador, inicia-se no conhecimento geral dos objetos do jogo, indo até o domínio dos desafios apresentados pelo jogo. Além da cognição, o psicomotor também é envolvido, a coordenação motora e a percepção visual no ambiente são fundamentais para o desenvolvimento do jogador no jogo.

Logo este estilo de jogo consegue englobar diversas características, como a socialização entre os jogadores, cooperação, criatividade diante de suas ações e imersão no jogo, interatividade com outros jogadores, com os desafios e a interdisciplinaridade no momento em que ele pode acessar diferentes tipos de desafios. Assim podemos ressaltar que um jogo do estilo MMOG é uma das formas de um ambiente de aprendizagem ser moldado para ser apresentado a seus usuários.

A interatividade nesses ambientes não se resumem apenas ao fato de poder andar, mas sim por manipular objetos, fazer ações diversas, conversar com outras pessoas e no momento em que podemos fazer tudo com gráficos mais realísticos, conseguimos imergir ainda mais o jogador no ambiente.

Ainda assim, quando falamos em jogos, que tem um estilo que permite a interação entre os jogadores, estamos falando sobre formas de abrir um canal de comunicação para ambas as partes. Rodrigues (2008) em seu trabalho apresenta algumas pesquisas relacionadas à forma pelo qual o jogador pode se comunicar com outros jogadores. As duas formas presentes eram via texto através do *chat* e a outra forma era por comunicação de voz.

Alguns autores apontam que o recurso de voz é superior ao texto, porém outros defendem que o uso de voz dificulta a imersão do jogador no ambiente. A grande vantagem apontada para a utilização de voz é a facilidade na hora de coordenação de uma equipe de forma rápida no andamento do jogo. Outro fator interessante apontado é que o jogador fica com as mãos livres para fazer outras ações no jogo enquanto se comunica com os outros jogadores.

Contudo a voz pode apresentar diversas limitações no momento em que o usuário está jogando o jogo, como ruídos que possam prejudicar o entendimento da mensagem enviada para os outros jogadores e como dito antes, prejudica na imersão do jogador no jogo.

Percebemos que no momento em que queremos desenvolver um jogo educativo, muito mais aspectos estão relacionados, além do que um jogo de outro estilo por si só já demanda.

Além de necessitar áreas de usabilidade, computação gráfica, física, IA, sistemas distribuídos, redes, psicologia entre outras, devemos focalizar ainda mais a área pedagógica.

2.4 Jogos Educativos Relacionados

Os jogos listados são jogos que foram desenvolvidos com o intuito de oferecerem um suporte à educação e com embasamentos para que isto seja possível, alguns deles são MMOGs.

2.4.1 Jeomm¹

É um jogo online 3D do estilo MMOG voltado para a educação utilizando arquiteturas organizacionais que existem nas escolas como disciplinas, notas, temas, avaliações, textos de explicação, entre outras coisas.

O mundo virtual consiste em replicar locais históricos, salas de aula, zoológicos. A ideia do jogo é replicar qualquer parte do mundo onde seja interessante para que o aluno possa adquirir experiências e aprender.

O interessante deste projeto é que eles têm como objetivo criar um ambiente onde o mundo virtual possa ser construído por qualquer pessoa que deseja ajudar no projeto, onde qualquer parte do mundo pode ser constantemente atualizada.

2.4.2 Crossword Compiler²

Este jogo permite criar palavras cruzadas, o interessante é que é possível que alguma pessoa (por exemplo, o professor) pode configurar as palavras que devem ser apresentadas na horizontal e na vertical.

As palavras podem ser selecionadas de uma lista disponível no jogo ou a pessoa pode digitar ainda se preferir. As palavras estão vinculadas a uma frase que direciona a resposta.

¹ <http://jeomm.unb.br/portal/>

² <http://www.crossword-compiler.com/?lang=en>

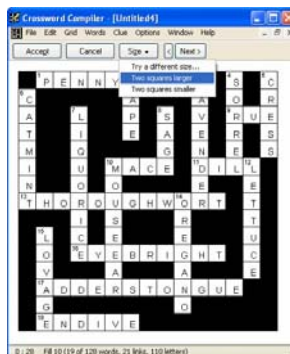


Figura 2: Crossword Compiler.

2.4.3 O que é o que é? ³

É um jogo construído a partir da ideia construcionista parcial, onde é o conhecimento constituído pela interação da pessoa com o meio físico e social. Ele foi desenvolvido via web e utiliza o conhecimento do senso comum coletado em outro projeto.

O jogo é de adivinhação baseado em cartas com conteúdo, logo o jogador precisa descobrir qual é a palavra secreta para acertar. Durante este processo o jogo apresenta algumas dicas para ajudar.

Todo conteúdo pode ser cadastrado pelo professor e ele ainda pode ter um feedback das respostas dos alunos, podendo assim avaliá-los. O jogo foi desenvolvido na linguagem Python.

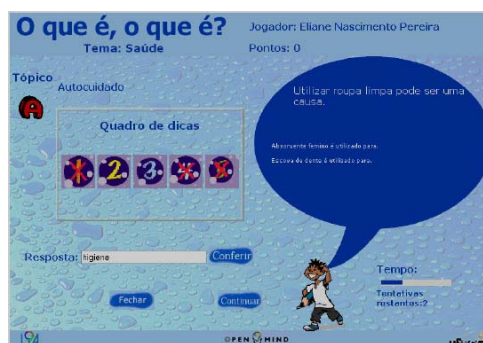


Figura 3: O que é o que é?.

³ <http://lia.dc.ufscar.br/game>

2.4.4 Ensino aprendizagem

Este jogo de Albuquerque (2008) está voltado ao assunto do meio ambiente, da disciplina de ciências da segunda série do ensino fundamental. O primeiro jogo desenvolvido é sobre perguntas e respostas. A pergunta é lançada na tela, a criança tem a opção de escolher entre três letras que indicam a resposta que ele acha que é a correta. Logo após a resposta o sistema lança um feedback ao aluno dizendo se ele acertou ou não.

O segundo jogo é chamado de “Limpe o Rio”, onde a criança precisa arrastar o lixo mostrado dentro do rio e colocar na lixeira, incentivando a criança a limpar os rios. Ao final do jogo é lançada uma mensagem dando os parabéns para a criança pela ajuda na despoluição do rio.

O terceiro é “Decifre o Enigma” onde o aluno precisa decifrar uma frase trocando o sinal de interrogação pela letra correspondente. É mostrado um painel com símbolos e qual a letra correspondente para o aluno saber qual letra colocar no sinal de interrogação. Neste jogo é apresentado um conteúdo multidisciplinar (Ciências, Português e Lógica para Matemática).

O último jogo apresentado pelo trabalho é um quebra-cabeça sobre um cenário específico que é o rio Muriaé.

2.4.5 Subindo e escorregando: Jogo para introdução do conceito de adição de números inteiros

É um jogo educativo da Bongioiolo (1998) motivado pela ideia de estimular e gerar um ambiente de fascínio e sedução em torno de atividades que desafiem o aluno a pensar e criar soluções para os problemas apresentados.

Um ambiente visual é constituído por uma trilha e umas montanhas, onde o jogador irá jogar os dados para andar nesta trilha. O jogo possui dois dados, um azul e o outro vermelho. O vermelho caracteriza quantas casas ele deve avançar e o azul representa quantas casas ele deve recuar.

Quando o jogador jogar os dados, ele deve fazer a operação entre um dado e o outro para saber em que casa ele deve ir. O jogo termina quando chegar na casa 10.

Durante o jogo o aluno ainda pode pedir ajuda. Onde irá para uma outra tela mostrando a operação e o resultado, quando o aluno voltar, o peão avança para a casa correta.

Porém o jogo não se caracteriza apenas por isso, existe ainda uma outra modalidade dentro dele que é a de desafios, que são perguntas e respostas exatas. Os desafios podem ser cadastrados pelo professor e são questões de múltipla escolha com resposta única.



Figura 4: Jogo Perguntas e Respostas e Enigma.

2.4.6 L.I.S. – Learning in the Space

O L.I.S. de Montovani (1996) é um jogo educativo cooperativo que promove a resolução de problemas, tomada de decisões e cooperação entre pares. Ele é destinado a alunos da 5^a e 6^a séries do ensino fundamental.

O jogo é constituído por três missões: quiz, caça palavras e quebra-cabeça. O professor cria um jogo e então ele é designado como comandante, após isto, os alunos poderão acessar o jogo criado e serão os tripulantes.

O comandante irá dividir os alunos em equipes, quando então é habilitado um *Chat* para os alunos poderem conversar. Quando a equipe estiver formada é escolhido então um capitão para fazer a interação com o software.

O capitão então irá escolher entre três missões para começarem a responder. Quando concluídas as missões do jogo, é lançado um feedback para a equipe. O professor terá acesso a um editor, onde poderá fazer cadastro de diversos tipos de conteúdo para os jogadores jogarem.



Figura 5: L.I.S..

2.4.7 Protetores das Tartarugas

Silva (2008) desenvolveu um jogo que transmite a sensação de que o jogador está dentro de um submarino (Figura 6), fazendo uma viagem. Quando o jogador conecta-se ao jogo ele encontra um mapa-mundi com pontos onde existem submarinos esperando o embarque.

Cada submarino pode possuir no máximo três jogadores, que podem se comunicar através do Chat. Dentro do submarino existe um radar que irá informar para um jogador que algo foi encontrado naquela região. O outro jogador, o mergulhador, terá que ir até o local e tirar fotos e trazer para o submarino e o terceiro jogador ficará com a responsabilidade de identificar qual a espécie encontrada nas fotos.

Existem ainda predadores, onde os tripulantes do submarino devem usar um sonar para expulsá-los, neste momento existe uma interação entre elas para haver decisão em grupo. É possível ainda que o professor entre em cada submarino como “chefe de pesquisas” para avaliar o andamento do grupo e até mesmo direcioná-los.



Figura 6: Protetores de Tartarugas.

2.4.8 Diversos

Através da revisão bibliográfica de *Learning in Immersive Worlds* de Freitas (2006) podemos encontrar diversos jogos que servem como base para a educação, apesar de muitos não terem sido desenvolvidos para este fim. Os principais jogos são *The River City Project*, *Racing Academy* e *Second Life*.

2.5 Comparativo entre os jogos educativos abordados

Com o levantamento de diversos jogos relacionados à educação foi traçada uma tabela com pontos relevantes, informando suas características. É possível constatar que muitos jogos apresentam uma boa estrutura para jogos educativos, porém é difícil não deixar a desejar em algum ponto.

Alguns dos jogos utilizaram a mesma tecnologia proposta neste trabalho como o jogo Protetor das Tartarugas que foi desenvolvido na tecnologia Flash. A Tabela 1 é ilustrado através das métricas de comparação 0, 1, 2 e 3 indicando o significado semântico dos pontos.

Tabela 1: Relacionamento entre os jogos

Jogo	Conteúdo Interdisciplinar	Para todas as idades	Ambiente Colaborativo	Conteúdo Manipulável	Roda na Web	Apresenta conteúdo multimídia (vídeo / imagens)	Freeware	Linguagem Programação
JEOMM	3	1	1	1	1	1	1	Não informado
Crossword Compile	1	1	0	1	0	0	0	Não informado
O que é o que é?	1	1	0	1	1	3	1	Python
Ensino aprendizagem – Educação Ambiental	3	0	0	0	1	1	1	Flash
Subindo e Escorregando	0	0	0	2	0	2	1	Não informado
Learning in the Space	1	2	1	1	0	1	1	Não informado
Protetores das tartarugas	0	3	1	0	1	1	1	Flash
The River City Project	0	2	1	1	0	1	1	Não informado
Racing Academy	0	1	0	0	0	0	1	Não informado
Second Life	1	1	1	1	0	1	1	Linden Scripting Language

Legenda: 0 – Não 1 – Sim / Rico 2 – Médio 3 – Pobre / Difícil

3 O JOGO: MUNDUS

A origem do nome surgiu do título “mundo dos desafios” da proposta deste trabalho, a partir do amadurecimento das ideias. No momento em que diversos jogadores estão respondendo as questões do jogo estamos vivenciando desafios escolares e culturais. O jogo se passa dentro de um mundo virtual, logo a palavra *Mundus* transcreve o significado necessário.

Este capítulo trata das etapas cumpridas para definição preliminar do modelo do ambiente interativo de aprendizagem. Serão apresentadas as etapas para o levantamento dos requisitos com sua devida conclusão. A partir dos requisitos levantados, foram diagramados os casos de uso, classes e o modelo relacional que servirão de suporte para a implementação do trabalho.

As definições de usabilidade e ergonomia foram abordadas neste capítulo para embasar os protótipos do jogo e o painel administrativo. E por fim apresentamos o modelo arquitetural do programa, informando tecnologias, ferramentas, APIs de desenvolvimento, gráfico do jogo, estrutura de dados, logs e um estudo sobre *middlewares* para atender a necessidade do jogo em ser *multiplayer*.

3.1 Questionário para levantamento dos requisitos

Apesar de todo levantamento teórico realizados para embasar as principais abordagens do jogo e como ele deve funcionar, devemos considerar que os protagonistas são a peça principal para o funcionamento do jogo.

Se o jogador entrar em um mundo que não interessa, em que os desafios não fazem ele interagir e nem participar, temos um enorme problema, pois um jogo que era para ser educativo e acima de tudo divertido, se torna chato e de material muito complicado para educá-lo.

Outro problema chave é a forma em que o professor pode interagir com os outros jogadores, precisamos dos *feedbacks* deste público, pois é através dele que o enriquecimento do mundo será possível e então assim fazendo o primeiro passo para o jogo ser educativo e eficaz.

Visando estes dois aspectos principais foi necessário buscar uma forma para conseguir integrar tanto o aluno quanto o professor para ajudar na construção desta ferramenta.

Através de dois modelos de questionário foram abordadas diversas perguntas buscando uma informação consistente para sua utilização, um questionário é voltado para os alunos (Anexo A) e o outro questionário é voltado para os professores (Anexo B).

A ferramenta utilizada para montar o questionário foi o *Google Form*, uma ferramenta muito prática que facilita a análise dos resultados. Após sua montagem foi enviado através do *e-mail* aonde conforme iam respondendo as informações eram salvas pelo *Google Form*.

3.1.1 Análise: Questionários dos Alunos

Através da análise de resultados das respostas do questionário apresentado para os alunos (Anexo C) foram computadas 74 respostas. Iniciou-se no dia 18/05/09 até o dia 13/06/09. A maioria das pessoas que responderam o questionário é do sexo masculino com cerca de 74%.

Pelos resultados analisados, verificou-se que 26% não costumam jogar jogos eletrônicos, sendo que apenas 3% realmente não se interessam por jogos que possam apresentar desafios. Percebemos então que a grande maioria se interessaria por um jogo educativo. Eles abordaram que o jogo deve apresentar desafios, que devem ficar mais difíceis e ao mesmo tempo o jogo deve ser divertido. A educação deve estar em segundo plano no jogo, onde ela deverá ser uma consequência do andamento do jogador no mundo. Muitos abordam que o jogo é uma forma de relaxar e escapar do dia a dia, se o jogo se torna maçante e repetitivo, eles não vão se interessar.

Outro dado relevante é que 74% acham importante a formação de equipes durante o jogo para enfrentar os desafios, sendo que a principal forma de comunicação é através de *chats*. Mas esta não é a única forma em que o pessoal acha interessante, o áudio também, segundo eles, é um bom recurso para comunicação no momento em que estiverem imersos nos desafios. Poucos acharam importante o vídeo para comunicação, alguns até justificando que o vídeo pode tirar a concentração do jogador e o seu foco no desafio.

Foi perguntando qual o cenário que seria mais interessante para que o jogo acontecesse. Os mais citados foram ambientes medievais, seguido de ambientes urbanos e florestas. Segundo algumas respostas, o ambiente deve apresentar um cenário bem desenvolvido e que permita ao jogador explorar o mundo.

No momento de criação do personagem, as principais características que foram ditas são cor da pele, tipo de cabelo, sexo e roupas. Muitos acham que quanto mais customizáveis, melhor, para poder transcrever suas características para seu personagem.

A última questão foi uma pergunta deixando a pessoa livre para suas considerações. Alguns abordaram que seria interessante que o jogo tivesse uma moeda de troca. Conforme o andamento do jogador ele vai adquirindo uma espécie de dinheiro onde pudesse comprar novos itens, como roupas. Outro sugeriu um ranking para que os jogadores possam saber como vai o andamento de outros jogadores.

Outro questionário respondido disse que seria interessante ter outras formas de enfrentar desafios, não sendo apenas de perguntas e respostas. O jogo deve apresentar varias formas de se ganhar, não tendo apenas uma. Ele também deve ter uma historia atrativa e com cenários bem desenvolvidos.

Outro questionário apontou que no momento em que existe uma moeda de troca, evolução no ambiente e até mesmo ranking, o jogo pode se tornar mais atrativo para ser jogado.

3.1.2 Análise: Questionários dos Professores

O questionário dos professores (Anexo D) iniciou no dia 15/05/09 e foi até o dia 13/06/09. Todos acharam interessante que o conteúdo tivesse materiais relacionados, devido à complexidade de algumas questões e até mesmo para não frustrar o aluno. Em geral eles preferem links que direcionem o aluno para algum material, imagens e vídeos.

Para apresentar o feedback ao aluno a maioria das respostas sugeriu o uso de texto. Uma outra forma abordada foi por através de símbolos, onde cada símbolo poderia representar o acerto ou erro do aprendiz.

Na hora do cadastro do desafio e suas respectivas perguntas, as informações devem ser diretas, além da classificação mais ampla do tipo da questão como Matemática, Historia, deve ter um nível baixo onde o professor possa classificar em qual disciplina a pergunta pode ser relacionada como Álgebra linear, ou Teoria da Ciência ou o que ele deseja classificar de forma escrita.

Apresentar um nível de dificuldade para a questão. Quanto à questão de relacionar os desafios com senhas de acesso, muitos acharam desnecessários e qualquer um poderia ter o acesso a questão. Outros acharam que seria interessante para fazer uma tarefa para um

determinado grupo ou até mesmo quando a turma apresenta diversos alunos de cursos diferentes, podendo então dar tarefas específicas a estes grupos.

Para a construção do relatório, os que responderam acham interessante o trajeto em que o aluno fez para as tarefas, como quais foram acessadas, quando os conteúdos relacionados foram visitados, as respostas corretas e incorretas. Outra questão levantada foi o tempo de resposta de cada pergunta e da seção em que o aluno entrou.

Em geral a geração do Log no sistema deve apresentar os desafios acessados, as perguntas acertadas e erradas. A outra pergunta foi se seria interessante criar um mundo para os jogadores acessarem e acessar as seção para enfrentar os desafios. Por volta de duas pessoas abordaram que não seria interessante criar um mundo. Em geral as pessoas que responderam acham interessante.

A formação de equipe entre jogadores para responder os desafios seguindo o questionário é importante sim e deve ser liberado. Para o jogo instigar mais os alunos as sugestões foram de ter obstáculos para os alunos passarem para a próxima etapa, interfaces adequadas, atrativas e fáceis de usar.

3.1.3 Conclusão do Levantamento

Foi muito importante este levantamento para se definir os requisitos para a modelagem do ambiente a fim de estruturar o jogo.

Os alunos estão mais preocupados se o jogo realmente vai conseguir ser divertido ou não e os professores querem saber como estes alunos estão se comportando diante aos desafios apresentados.

Através das respostas dos alunos, foi possível constatar que eles se interessaram bastante pelo tipo do jogo, logo se o jogo educativo for bem desenhado, com aspectos relevantes e divertidos, eles irão jogar.

O questionário apresentou algumas questões em que as pessoas apresentaram um pouco de dificuldade nas respostas ou até mesmo não souberam responder.

3.2 Requisitos

Visando a análise entre a educação e a computação, é possível traçar um contexto para um jogo educacional com um ambiente cooperativo em formato de jogo. Em busca da

resposta da questão de pesquisa e de toda teoria levantada foi possível chegar em um desenho que engloba todos os aspectos principais. Ao longo desta seção as palavras ficaram mais claras.



Figura 7: Palavras relacionadas ao jogo.

O jogo tem como ideia inicial apresentar desafios aos jogadores e conforme o seu andamento o jogador vai adquirindo mais experiência para enfrentar outros novos desafios. No momento em que o jogador vai respondendo estes desafios, percebemos que não é apenas um jogo, mas também uma ferramenta que pode auxiliar o professor.

No seu primeiro acesso o jogador poderá criar o seu personagem definindo características como sua aparência. Neste momento da criação ele poderá escolher o sexo masculino ou feminino, algumas características físicas como cor do cabelo, tipo do cabelo e cor da pele. Será neste momento também que o usuário deverá escolher o nome do seu personagem. Vale ressaltar que deve existir uma validação na hora do cadastro dos nomes, para não permitir nomes iguais dentro do jogo.

Depois de criado seu personagem o usuário terá o primeiro contato com o mundo, ele poderá interagir com os outros usuários neste ambiente através de um *chat*, fazendo com que o ambiente permita uma socialização entre os jogadores, um dos aspectos principais de um ambiente colaborativo conforme Mantovani e Jonassen (1996).

No momento em que o jogador está dentro do mundo, ele precisa explorar o local, investigar onde podem estar os outros jogadores, buscar onde estão os desafios para resolver. Jonassen (1996) e Baranauskas (1999) falam que a investigação e a descoberta estimulam ainda mais o jogador dentro do ambiente, motivando-o a continuar e então se envolvendo ainda mais efetivamente.

Até mesmo para fornecer um suporte maior a esta investigação, exploração, curiosidade e motivação, seria interessante fornecer um mapa do mundo onde o jogador poderia ver em uma visão macro os locais que poderiam ser acessados e até mesmo uma lista de usuários logados.

O jogador terá autonomia própria, ele pode escolher os locais que ele deseja ir, quais desafios ele deseja responder, ele tem total controle sobre suas ações.

Inserindo no contexto do jogo o modelo ZPD definido por Vygotsky (1978), o professor deve possuir uma ferramenta capaz de atingir diretamente o nível potencial do jogador e não o desenvolvimento atual do aluno.

Logo os desafios presentes no jogo não são pré-fixados, toda a base de desafios pode ser ampliada constantemente através de um gerenciador acessado pela web onde existirá um painel para cadastro. O responsável pela base de conhecimento dentro do jogo são os professores.

No momento em que o professor está criando os desafios para serem inseridos dentro do jogo, será possível determinar diversas características nas perguntas. Uma seção é composta por diversos desafios. Um desafio é composto por diversas perguntas. As perguntas poderão ser de diferentes tipos como múltipla escolha, única escolha e dissertativa. O professor já deve fornecer a resposta, caso a questão seja dissertativa, no momento em que o aluno responder o professor receberá uma solicitação de correção da questão respondida pelo aluno. Esta questão é um bom diferencial do jogo, fazendo com que perguntas com respostas mais elaboradas sejam suportadas pelo sistema.

Uma das características do cadastro de perguntas é que poderá ser adicionado conteúdo relacionado às questões do desafio. Estes conteúdos relacionados poderão ser vídeos, imagens e links para textos, que servirão como subsidio para os alunos responderem o desafio caso o professor ache necessário.

Através dos desafios que podem ser visto como exercícios o aluno pode fixar um conhecimento já abordado antes e mesmo que ele não tenha aprendido, os conteúdos relacionados poderão ser fontes de informação para o aprendizado do aluno.

Ainda será possível adicionar uma senha de acesso na sala de desafio, assim o professor poderá criar grupos mais específicos permitindo o acesso apenas de alguns alunos, ou, por exemplo, apenas de uma turma.

É possível definir o número máximo de alunos que podem formar um grupo para responder esta pergunta, no momento em que o usuário acessa a pergunta para responder, ele pode convidar outros alunos para a mesma sala de desafio. Logo a individualização é determinada parcialmente pelo aluno que tem o poder de convidar ou não e pelo professor que define quantos alunos podem fazer parte do grupo ou se nenhum pode fazer.

No momento em que é formulado um grupo, tem-se como objetivo a cooperação entre eles e o equilíbrio e desequilíbrio de suas ideias para a formação de novas, onde a discussão entre os alunos ajudará no processo de construção do conhecimento.

Esta transmissão de informação será possível através de um *chat* interno apenas naquela sala de desafio, onde apenas os alunos do grupo poderão conversar entre si, caso exista um grupo.

Visando a ideia construcionista, foi desenvolvido um sistema de experiência global no mundo. Quando o professor cadastra um desafio ele determina quantos pontos de experiência é necessário ter para entrar no desafio. Caso o aluno não tenha os pontos necessários o jogo irá disparar um alerta avisando o jogador, pedindo se mesmo assim ele deseja ou não responder as perguntas do desafio.

A cada pergunta que o usuário acerta, ele ganha pontos de experiência. Caso erre, o usuário ganhará menos pontos de experiência. Evidenciando que é importante estimular o aluno a continuar tentando, pois se o jogador ficar desestimulado com o jogo, provavelmente ele quebrará o vínculo com ele e então abandonará o jogo, parando então o seu desenvolvimento. O jogo quer sempre estimular o aluno a continuar a desenvolver sua jornada. É importante ressaltar que a cada pergunta respondida, o sistema irá retornar um feedback para o aluno, fazendo com que o aluno entre num processo de reflexão sobre sua resposta e até mesmo para a formação de uma nova ideia conforme Piaget (1967) descreve sobre a construção do conhecimento.

Valente (1993) abordou que “O processo de achar e corrigir o erro constitui uma oportunidade única para o aluno aprender sobre um determinado conceito envolvido na solução do problema ou sobre estratégias de resolução de problemas”. Levando isto em consideração podemos perceber que é crucial que nos desafios respondidos pelo jogador, as respostas devem ser sempre mostradas, permitindo ao aluno identificar qual seria a questão certa.

Voltando aos aspectos do painel, será possível emitir relatórios, os professores poderão emitir para poder acompanhar a formação do conhecimento dos jogadores e ver como anda o andamento do jogador. O jogador terá direito também de emitir um relatório de seu andamento.

O jogo ainda fornecerá uma sessão onde será possível ver o ranking dos melhores jogadores do jogo, este ranking estará classificado pelos pontos de experiência ganhos nos desafios respondidos.

Os gráficos do jogo serão leves e divertidos, tornando o ambiente agradável. O mundo ainda apresentará sons ambientes para que o clima num todo seja agradável para o usuário evoluir seu conhecimento e experiência.

Visando o desafio e a diversão, podemos referenciar o trabalho de Braga (2007), ele aponta que o psicólogo Mihaly Csikszentmihalyi identifica numa pesquisa o que diverte uma pessoa na realização de um desafio. Segundo sua teoria, a pessoa só se sentirá desafiado quando a carga de dificuldade de uma determinada tarefa aumenta e na medida que ela vai adquirindo habilidade, garante que a pessoa fique na linha do fluxo e ainda se divirta.

O fato em que a pessoa consegue resolver os problemas propostos e a possibilidade de concentração em uma tarefa e somente isto, permite a pessoa entrar em um outro mundo, esquecendo de seus problemas diários, tornando-se uma alienação saudável. Braga ainda representou através de uma imagem que é interessante ressaltar, da Figura 8.

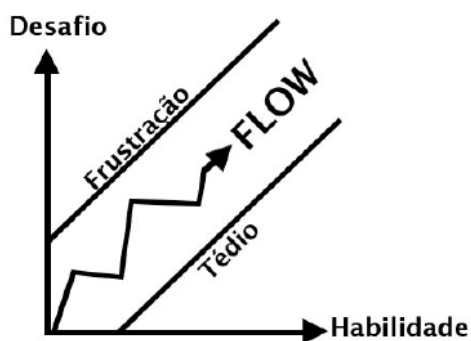


Figura 8: Fluxo da Diversão.

No momento em que a pessoa entra num desafio para resolução de um problema, ela está na linha do fluxo, onde esta linha exige do jogador o conhecimento para resolver o desafio. Conforme ela vai realizando a tarefa se não ficar mais complicado, ela vai ficando entediada e caso fique muito complicado ela começa a ficar frustrada. Note que existe uma

linha tênue entre a frustração e o tédio, onde é nesta linha que o sistema deve se comportar para abstrair ao máximo a pessoa.

3.3 Casos de Uso

Esta seção apresenta os diagramas de casos de uso para descrever como diferentes tipos de usuários interagem com o sistema para fazer determinadas ações. Ele servirá de base para a modelagem de outros diagramas, protótipos e implementação do sistema.

3.3.1 Painel: Visão Geral

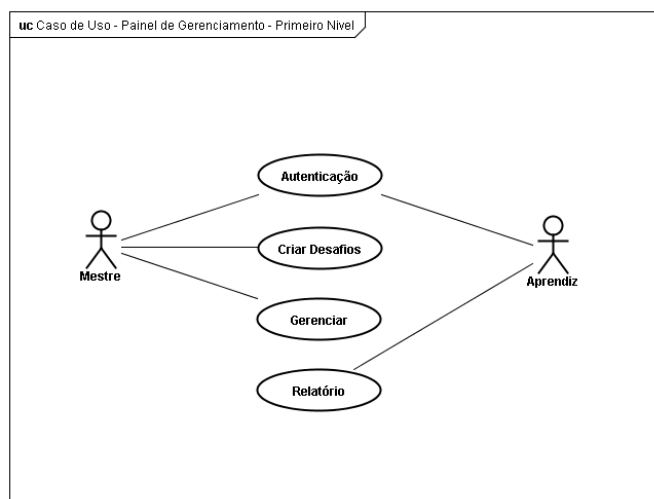


Figura 9: Caso de uso – Painel – Visão geral

1. Autenticação

Atores: Mestre e Aprendiz

Finalidade: Autenticar usuário para utilização do painel do jogo.

Visão Geral: No momento em que é aberto o painel, o sistema irá solicitar o e-mail e a senha para fazer a autenticação. Caso as entradas dos dados sejam válidas o sistema irá mostrar a tela principal do painel. A senha deverá ser criptografada em MD5.

Fluxo de Atividades:

- 1) O usuário entrará com seus dados de e-mail e senha;

- 2) O sistema validará os dados e retornará uma resposta;
- 3) O usuário será direcionado para a tela principal do sistema caso o *login* tenha sido aceito.

Fluxo de Exceção (3):

- 1) O usuário não é autenticado pelo sistema e então permanecerá na tela de autenticação do sistema;
- 2) Retorna para ao item 1 (atividades).

Fluxo de Testes:

- 1) Verificar se a senha está sendo criptografada com MD5;
- 2) Verificar se a autenticação está funcionando.

2. Criar Desafio

Atores: Mestre

Finalidade: Criar um novo desafio para ser utilizado dentro do jogo.

Visão geral: O mestre irá criar um desafio informando alguns parâmetros para a definição de sua estrutura, criar tarefas relacionadas e caso deseje alocar conteúdos relacionados. O desafio é composto por diversas tarefas, que são as perguntas que os jogadores terão que responder. Os conteúdos relacionados são materiais que o professor acha importante para os alunos pesquisarem no momento da interação com o desafio. Através deles o aluno poderá buscar o conhecimento para responder as questões propostas.

Fluxo de atividades:

- 1) O mestre selecionará no menu principal do programa a ação de criar um novo desafio;
- 2) O sistema retorna a tela de cadastro de desafios;
- 3) O mestre entra com os parâmetros do desafio, cadastra as tarefas respectivas e caso deseje, conteúdos relacionados;
- 4) O sistema salva as informações e retorna um aviso ao mestre.

Fluxo de Exceção – Inconsistência dos dados (4):

- 1) Se algum campo não estiver preenchido o sistema deve retornar uma mensagem de alerta e não salva as informações;

- 2) Retorna para o item 3 do fluxo de atividades.

Fluxo de Teste:

- 1) Verificar se o mestre está logado para fazer o cadastro;
- 2) Verificar integridade dos dados.

3. Gerenciar

Atores: Mestre

Finalidade: Gerenciar informações geradas pelo jogo como as perguntas que necessitam de respostas e relatórios em geral.

Visão geral: O gerenciamento das informações é crucial para o desenvolvimento do aluno e o bom andamento do jogo. Através do gerenciamento será possível responder as questões dissertativas das tarefas respondidas pelo aluno que necessitam de uma moderação. Outra ação de gerenciamento são os relatórios dos alunos onde o professor poderá saber como está o andamento no aluno no desenrolar do jogo.

Fluxo de Atividades:

- 1) O mestre irá acessar no menu principal a ação desejada;
- 2) O sistema irá retornar a tela desejada ao mestre;
- 3) O mestre irá fazer as ações desejadas;
- 4) O sistema irá salvar as informações quando necessário.

4. Relatório

Atores: Aprendiz

Finalidade: Analisar como anda sua performance relacionada às atividades desenvolvidas no jogo.

Visão Geral: O aluno poderá acessar o painel e ver um relatório detalhado com todos os desafios participados, quais perguntas foram acertadas e erradas, quanto ele tem de experiência e a porcentagem de acerto.

Fluxo de Atividades:

- 1) Aprendiz entra na sessão de relatório;

2) Sistema verifica qual é o aprendiz e retorna um relatório de seu andamento no jogo.

Fluxo de Exceção (2):

- 1) Caso o aprendiz não tenha nenhuma tarefa realizada, lançar um alerta na tela;
- 2) Não abrir a sessão do relatório.

Fluxo de Teste:

- 1) Verificar se o relatório está trazendo as informações consistentes e completas.

3.3.2 Painel: Expansão do Caso de Uso – Gerenciar

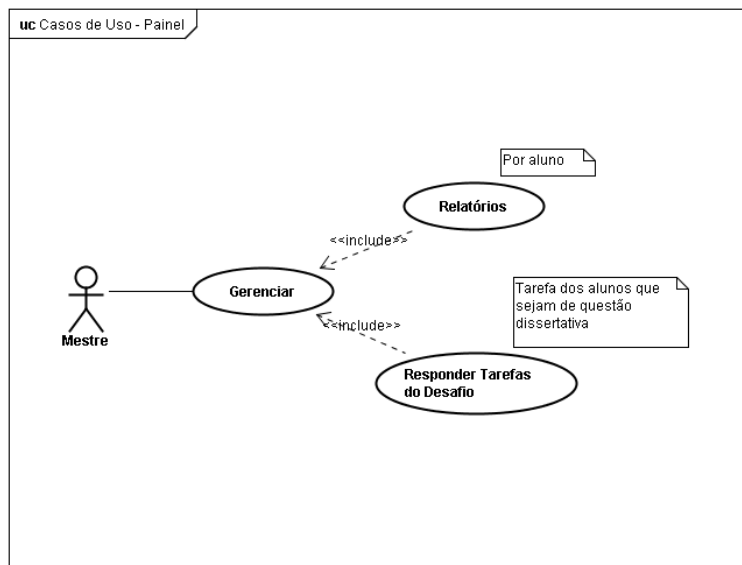


Figura 10: Caso de uso – Painel – Expansão – Gerenciar

1. Relatório

Atores: Mestre

Finalidade: Analisar o andamento dos jogadores em relação aos desafios que foram enfrentados.

Visão Geral: O relatório serve como uma fonte de informação para o professor conseguir verificar se os jogadores estão conseguindo acertar ou não os desafios propostos por ele, se o jogador está conseguindo evoluir entre outras coisas. É a ligação mais direta que o professor poderia ter para fazer uma análise mais focada no jogador. No relatório irá constar as mesmas

informações emitidas no relatório que o aprendiz pode visualizar. A única diferença é que o mestre consegue escolher de quem quer visualizar o relatório enquanto o aprendiz só pode ver o dele.

Fluxo de Atividades:

- 1) Mestre acessa a sessão de relatórios;
- 2) Sistema retorna uma listagem de todos os alunos dando a possibilidade do mestre fazer alguns filtros para encontrar o jogador desejado;
- 3) Mestre escolhe o jogador;
- 4) Sistema retorna os dados referentes ao jogador.

Fluxo de Exceção – Sem dados (1):

- 1) Retorna um alerta caso não exista jogadores cadastrados no sistema;
- 2) Não abre a sessão de relatórios.

Fluxo de Exceção – Sem jogadas (3):

- 1) Jogador não tem nenhum desafio enfrentado;
- 2) Sistema retorna um alerta.

Fluxo de Teste:

- 1) Verificar se o jogador selecionado e os dados retornados conferem;
- 2) Verificar se está vindo todas as informações para o mestre referente ao jogador.

2. Responder Tarefas do Desafio

Atores: Mestre

Finalidade: Responder tarefas dos desafios que apresentam questões dissertativas e que o desafio está vinculado ao professor.

Visão Geral: No momento em que o professor acessa esta seção, ele poderá ver uma listagem de todas as tarefas que estão pendentes para verificar se o aprendiz respondeu certo ou não a tarefa. Nesta listagem ele poderá escolher qual tarefa ele deseja responder e então abrirá uma janela na frente das outras, onde ele poderá informar se está certo ou não o que o aprendiz respondeu e caso não esteja ele poderá colocar a resposta correta.

Fluxo de Atividades:

- 1) Mestre acessa a seção para responder as tarefas dos desafios;
- 2) O sistema retorna uma listagem de tarefas;
- 3) Mestre escolhe alguma tarefa para responder;
- 4) Sistema retorna a tela de resposta;
- 5) Mestre responde e clica para salvar os dados;
- 6) Sistema salva as informações.

Fluxo de Exceção – Listagem (2):

- 1) Caso não exista nenhum item para responder, mostrar aviso ao usuário.

Fluxo de Exceção – Inconsistência dos dados (5):

- 1) Sistema verifica e encontra algum campo obrigatório vazio;
- 2) Lança um alerta na tela para avisar o mestre e retorna ao item 4.

Fluxo de Teste:

- 1) Verificar se as tarefas estão relacionadas certas ao professor que cadastrou o desafio;
- 2) Verificar se a tarefa está sendo respondida e o usuário possa acessar a resposta do professor.

3.3.3 Painel: Expansão do Caso de Uso – Criar Desafios

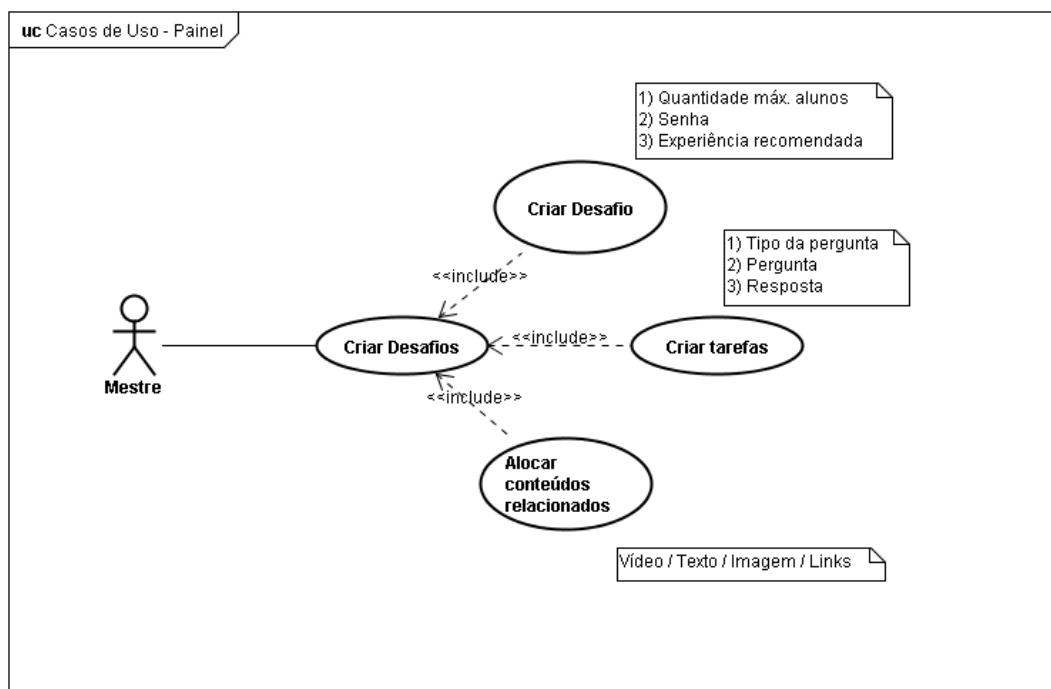


Figura 11: Caso de uso – Painel – Expansão – Criar Desafios

1. Criar desafio

Atores: Mestre

Finalidade: Iniciar a configuração para criar o desafio, cadastrar conteúdos relacionados e tarefas.

Visão Geral: Neste momento o mestre irá informar o nome do desafio, se os aprendizes podem fazer grupos para responderem junto o desafio proposto e qual a quantidade máxima do grupo e informará uma senha de acesso caso ele deseje que o desafio fique específico para um grupo de alunos. Após a entrada dos dados ele irá alocar os conteúdos relacionados, as tarefas e então salvar.

Fluxo de Atividades:

- 1) O mestre irá lançar as informações principais do desafio no sistema;
- 2) O mestre irá criar as tarefas;
- 3) O mestre irá alocar os conteúdos relacionados;
- 4) O sistema irá validar a informação e salvará a informação.

Fluxo de Exceção Inconsistência dos Dados (4):

1) Sistema verifica se foi deixado o campo nome ou quantidade de grupos em branco e se existem tarefas alocadas ao desafio, retorna uma mensagem de alerta caso alguma desses requisitos não estejam corretos;

2) Retorna ao item 1.

Fluxo de Teste:

1) Verificar se as informações estão sendo gravadas e de forma correta.

2. Alocar conteúdos relacionados

Atores: Mestre

Finalidade: Conteúdos complementares no desafio para quando os aprendizes forem interagir com o desafio, terem subsídios para buscar as informações caso necessitem.

Visão Geral: Os conteúdos relacionados poderão ser em vídeo, texto e imagem. Poderá existir nenhum ou vários conteúdos relacionados em um mesmo desafio. No momento em que o aprendiz acessar o desafio ele terá a lista de conteúdos relacionados referente ao desafio.

Fluxo de atividades:

- 1) Clicar em adicionar um novo conteúdo relacionado dentro da tela de criar desafios;
- 2) Sistema retornará um *modal* para cadastro do conteúdo relacionado;
- 3) Mestre escolherá o tipo da informação e fará *upload* do arquivo ou irá digitar o texto ou *link* referente ao conteúdo;
- 4) O mestre clicará em incluir a informação no desafio;
- 5) Sistema irá validar as informações e inserir o conteúdo no desafio.

Fluxo de Exceção Inconsistência dos Dados (3):

1) Tipo de arquivo inválido ou campos não preenchidos, lança um alerta na tela informando o usuário;

2) Retorna ao item 2.

Fluxo de Teste:

- 1) Verificar se o arquivo está sendo salvo no servidor;
- 2) Verificar se o conteúdo relacionado está vinculado ao desafio correto.

3. Criar Tarefas

Atores: Mestre

Finalidade: Criar as tarefas referentes ao desafio que está sendo criado.

Visão Geral: Neste momento o mestre irá cadastrar as tarefas, ele poderá escolher se a tarefa é do tipo múltipla escolha, única escolha ou dissertativa. Ele cadastrará a pergunta e também qual será a resposta. Caso o tipo da pergunta seja dissertativo ele não precisará colocar a resposta.

Fluxo de Atividades:

- 1) Clicar em adicionar uma nova tarefa;
- 2) O sistema irá retornar uma janela na frente das outras para cadastro da tarefa;
- 3) O mestre irá informar os dados para o cadastro da tarefa e clicará em incluir;
- 4) O sistema validará as informações e incluirá o conteúdo no desafio.

Fluxo de Exceção Inconsistência dos Dados (3):

- 1) Dados não preenchidos, então retorna um alerta na tela;
- 2) Volta para o item 2;

3.3.4 Jogo: Visão Geral

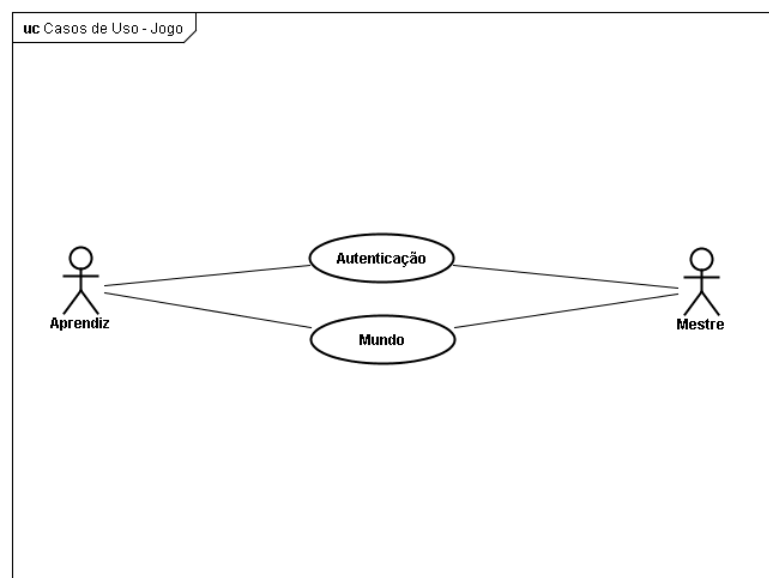


Figura 12: Caso de uso – Jogo – Visão Geral

1. Autenticação

Atores: Aprendiz e Mestre

Finalidade: Autenticar jogador para acessar o mundo virtual.

Visão Geral: Neste momento o sistema irá validar o usuário permitindo então o acesso ou não ao jogo.

Fluxo de Atividades:

- 1) Jogador informa seus dados no *login*;
- 2) Sistema verifica os dados e retorna a resposta;
- 3) Jogador acessa o mundo virtual caso o acesso tenha sido liberado.

Fluxo de Exceção (3):

- 1) O sistema não libera o usuário para o acesso;
- 2) O jogador recebe um aviso na tela;
- 3) Retorna ao item 1.

Fluxo de Exceção – Servidor (3):

- 1) O sistema não consegue acessar o servidor;
- 2) Sistema retorna uma mensagem de servidor fora do ar.

2. Mundo

Atores: Mestre e Aprendiz

Finalidade: Mundo virtual para os jogadores acessarem, interagirem com outros jogadores e enfrentarem desafios.

Visão Geral: No mundo o jogador irá ter seu próprio avatar que caracteriza sua identidade. Os jogadores poderão acessar diversas seções e enfrentar desafios, para eles conseguirem acessar a seção o jogador deverá explorar o mapa e achar os pontos de sessão.

Fluxo de Atividades:

- 1) O jogador entrará no mundo persistente;
- 2) O sistema e o jogador começam a interagir.

3.3.5 Jogo: Expansão do Casos de Uso – Mundo

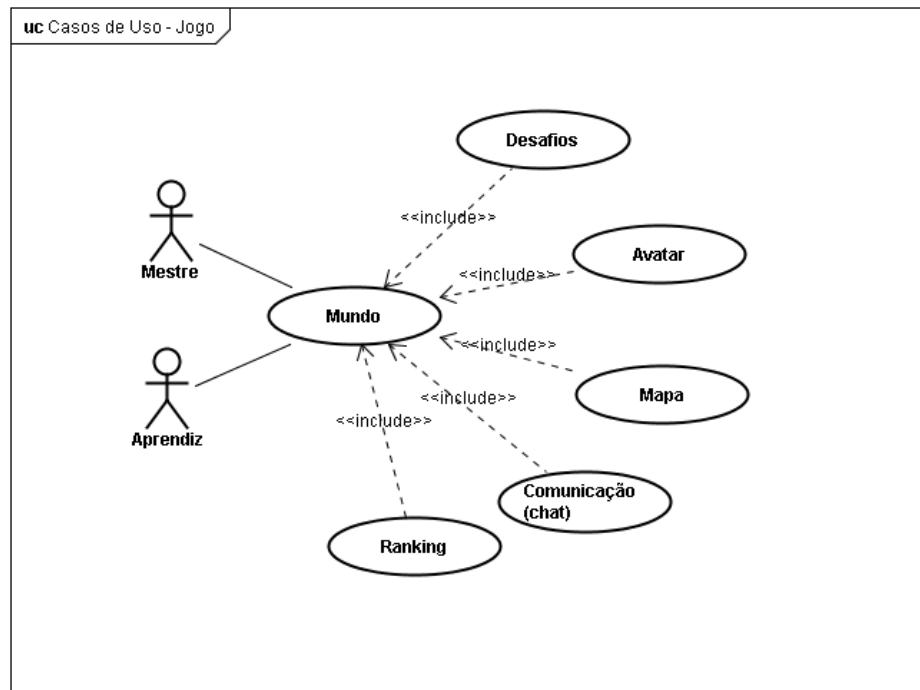


Figura 13: Caso de uso – Jogo – Expansão – Mundo

1. Desafios

Atores: Mestre e Aprendiz

Finalidade: Apresentar desafios aos jogadores

Visão geral: Os desafios estão divididos em diferentes seções, no momento em que o jogador acessa uma seção ele pode escolher algum desafio para responder. Os desafios são compostos por diversas tarefas. No momento em que o jogador diz que quer enfrentar o desafio ele poderá ver conteúdos relacionados, responder as tarefas ou pular alguma, convidar outros jogadores para enfrentar as tarefas e se comunicar com eles através de um *chat* para trocarem ideias. No final, o desafio apresentará um *feedback* para o jogador ou jogadores.

Fluxo de Atividades:

- 1) Jogador explora o mundo e entra numa seção;
- 2) Mundo retorna uma listagem de desafios, tanto os respondidos quanto os não respondidos;

- 3) O aluno escolhe algum desafio que não foi respondido ainda por ele;
- 4) Sistema retorna o desafio para o aluno responder;
- 5) O aluno responde o desafio;
- 6) Sistema salva as informações.

Fluxo de Testes:

- 1) Desafios com marcações de enfrentados ou não.

2. Avatar

Atores: Mestre e Aprendiz

Finalidade: Permite que o usuário tenha uma identidade dentro do mundo virtual.

Visão Geral: O jogador poderá criar seu avatar, escolhendo o nome, sexo entre outras características. As características principais são o sexo, cor da pele, tipo de cabelo e cor do cabelo. Cada *login* pode criar apenas um avatar dentro do jogo.

Fluxo de Atividades:

- 1) O jogador acessa o mundo no momento da autenticação;
- 2) O mundo verifica se o jogador já possui um avatar criado e retorna ele;
- 3) O jogador visualizará seu personagem e de outras pessoas;

Fluxo Alternativo (2):

- 1) O mundo verifica que o jogador não possui um avatar;
- 2) O sistema abrirá uma tela de cadastro de avatar;
- 3) O jogador entrará com suas informações e prosseguirá;
- 4) O sistema irá validar as informações e salvar o avatar do jogador.

Fluxo de Exceção (3 - fluxo alternativo):

- 1) O sistema irá verificar se o nome já existe;
- 2) Se o nome existir retornar para o item 3 do fluxo alternativo;
- 3) Sistema mostra um alerta ao jogador.

Fluxo de Teste:

- 1) Verificar se a validação do nome está correta;
- 2) Verificar se a criação do avatar está replicando no visual do mundo.

3. Mapa

Atores: Mestre e Aprendiz

Finalidade: Permite que o jogador consiga ter uma noção de todo o ambiente e ainda ver em que lugar ele está localizado.

Visão geral: O mapa será uma imagem marcando apenas o ponto em que o jogador está naquele momento. O mapa poderá ser acessado através da tela principal do jogo.

Fluxo de Atividades:

- 1) O jogador irá acessar o mapa;
- 2) O sistema irá retornar o mapa com a posição do jogador.

Fluxo de Teste:

- 1) Verificar se a posição retornada no mapa equivale a posição do jogador no mundo.

4. Comunicação

Atores: Mestre e Aprendiz

Finalidade: Permitir uma forma de comunicação com os outros jogadores que estão online dentro do mundo.

Visão Geral: O jogador deverá possuir uma forma de comunicação com os outros jogadores através do texto, quando o jogador acessar o mundo ele já poderá se comunicar com os outros que estão aparecendo na tela.

Fluxo de Atividades:

- 1) Jogador escreverá uma mensagem e enviará;
- 2) Sistema irá receber a mensagem e replicará para os jogadores visíveis na tela;
- 3) Os outros jogadores receberam a mensagem

Fluxo de Teste:

- 1) Validar se a mensagem está sendo mostrada apenas para os outros usuários visíveis na tela.

5. Ranking

Atores: Mestre e Aprendiz

Finalidade: Permitir que os jogadores consigam ver o nível dos outros jogadores.

Visão Geral: Através de um local no mundo o jogador poderá acessar uma listagem com os melhores jogadores do mundo. O critério para definir os melhores será através da experiência (pontos) adquiridos por eles por responderem os desafios.

Fluxo de Atividades:

- 1) O jogador deve encontrar local para acesso ao ranking no mundo;
- 2) O jogador irá clicar para abrir a listagem;
- 3) O sistema irá retornar o ranking.

3.3.6 Jogo: Expansão do Casos de Uso – Desafio

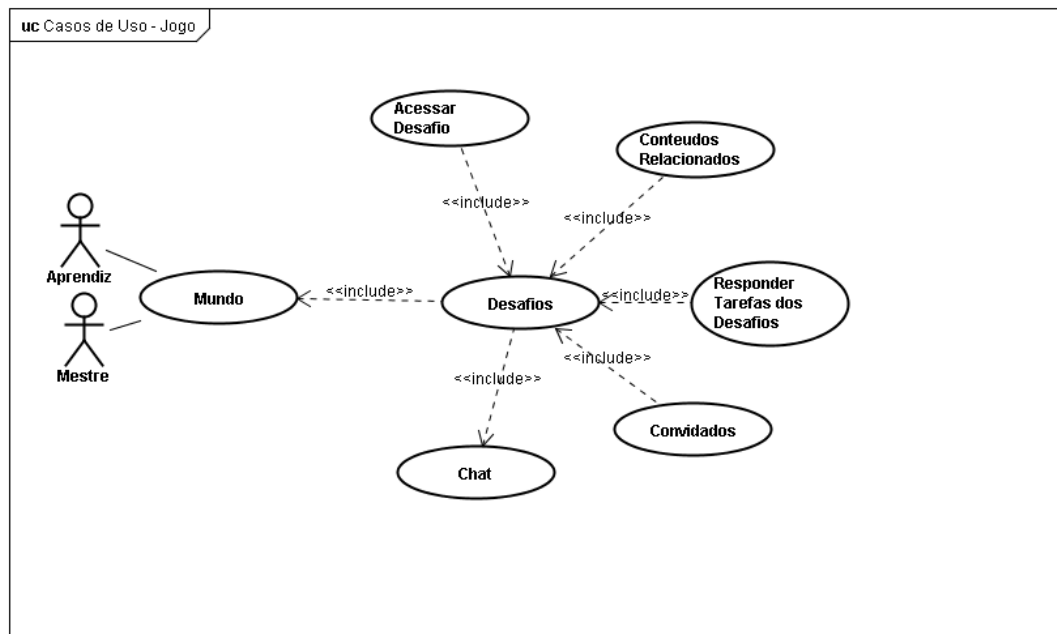


Figura 14: Caso de uso – Painel – Expansão – Desafio

1. Acessar Desafio

Atores: Mestre e Aprendiz

Finalidade: Acessar os desafios presentes em diferentes sessões espalhadas pelo mapa

Visão Geral: Neste momento o jogador irá explorar o mundo encontrando alguma seção. Nas seções ele poderá ver uma lista de todos os desafios relacionados a ela. Esta lista é formada através do cadastro dos professores. Através dessa listagem ele poderá escolher o desafio que ele deseja responder, caso ele não tenha respondido ainda. No momento de escolha abrirá uma janela para começar a interação.

Fluxo de Atividades:

- 1) Explorar o mundo, encontrar uma seção e abrir a listagem;
- 2) Sistema retornará a listagem de desafios;
- 3) Escolher um desafio;
- 4) Sistema irá retornar o desafio para o jogador interagir.

Fluxo de Exceção (3):

- 1) Sistema verifica que o desafio já foi respondido;
- 2) Sistema retorna um alerta ao jogador;
- 3) Retorna ao item 2.

Fluxo de Teste:

- 1) Verificar se a listagem de desafios é correspondente à seção.

2. Conteúdos Relacionados

Atores: Mestre e Aprendiz

Finalidade: Permitir que os jogadores consigam ver os materiais relacionados ao desafio caso o mestre tenha cadastrado.

Visão Geral: No momento em que o mestre está cadastrando o desafio ele poderá cadastrar conteúdos relacionados que ajudará o aluno no desenvolvimento dos desafios. O aluno poderá acessar os conteúdos relacionados que podem ser texto, imagem ou vídeo. Vale ressaltar que poderá ser mais de um item.

Fluxo de Atividades:

- 1) Jogador acessa o desafio;

- 2) Sistema busca os conteúdos relacionados do desafio;
- 3) Sistema retorna os conteúdos relacionados quando solicitado;
- 4) Jogador vê os conteúdos relacionados.

Fluxo de Exceção (2):

- 1) Sistema não encontre nenhum conteúdo relacionado cadastrado;
- 2) Sistema retorna um alerta para o usuário avisando que não existem itens cadastrados.
- 3) Retorna para o item 1.

3. Convidados

Atores: Mestre e Aprendiz

Finalidade: Permitir que o jogador convide outros jogadores para ajudarem a responder o desafio.

Visão geral: No momento em que o jogador acessar o desafio ele terá um botão que permitirá a ele convidar outros jogadores para ajudar a responder o desafio. Esta opção só estará habilitada caso o mestre tenha habilitado a permissão de formar grupos no desafio. Depois de convidar os outros jogadores para entrarem na mesma seção, os jogadores poderão utilizar o *chat* para se comunicar.

Fluxo de Atividades:

- 1) Jogador clica em incluir novo integrante ao grupo;
- 2) Sistema retorna uma lista de usuários online;
- 3) Jogador escolhe um dos usuários;
- 4) Sistema envia uma requisição assíncrona ao jogador escolhido;
- 5) Jogador aguarda o outro jogador aceitar ou rejeitar o convite.

4. Chat

Atores: Mestre e Aprendiz

Finalidade: Permitir um canal de comunicação entre os jogadores dentro do desafio.

Visão geral: No momento em que existem mais de um jogador dentro do desafio é habilitado

o *chat*. O *chat* servirá como a principal ferramenta de discussão entre os jogadores para a formulação das respostas para responder as tarefas do desafio.

Fluxo de atividades:

- 1) Sistema verifica se existe mais de um usuário para habilitar o *chat*;
- 2) Caso habilitado, o jogador envia mensagens;
- 3) O sistema publica as mensagens no mini *chat*.

Fluxo de Exceção (1):

- 1) Se não existir mais de um jogador na sessão o *chat* fica desabilitado.

5. Responder Tarefas dos Desafios

Atores: Mestre e Aprendiz

Finalidade: Responder o desafio

Visão geral: Neste momento o jogador ou os jogadores terão acesso às perguntas do desafio, eles poderão ainda consultar os conteúdos relacionados. No momento em que ele vê a pergunta abrirá um campo para que ele possa preencher a resposta ou marcar a resposta, dependendo do tipo da tarefa. Após a última pergunta respondida um *feedback* é retornado ao aluno com a quantidade de acertos, erros e pendentes que aguardam a resposta do professor. Neste momento ele ganha pontos de experiência pelas respostas acertadas e por participação.

Fluxo de atividades:

- 1) Jogador inicia o desafio;
- 2) Sistema mostra a pergunta ao jogador;
- 3) O jogador responde;
- 4) O sistema armazena a resposta;
- 5) Retorna ao item 3 até terminar as perguntas, quando terminar vai para o item 6;
- 6) O sistema dá um *feedback* ao aluno;
- 7) O sistema adiciona pontos de experiência ao jogador;
- 8) Sistema encerra o desafio e fecha a tela.

Fluxo de Exceção (3):

- 1) Caso não tenha a resposta o sistema não valida a resposta;
- 2) Retorna ao item 2.

Fluxo de Teste:

- 1) Verificar se todas as perguntas estão sendo mostradas na tela;
- 2) Verificar se o *feedback* ao aluno está correto;
- 3) Verificar se os pontos de experiências estão sendo salvo tanto para perguntas acertadas, quanto para erradas.

3.4 Diagrama ER

O diagrama da Figura 15 representa o modelo de dados do sistema e suas devidas relações. Podemos partir da entidade seção, que representa os tipos do desafio, como por exemplo: matemática, história e etc. A entidade desafio é composta por uma única seção e é composto por diversos conteúdos relacionados, representado na entidade *conteudo_relacionado*.

Um conteúdo relacionado apresenta o atributo tipo, que pode ser texto, link, vídeo ou imagem, o atributo dados contém os dados referente ao tipo do conteúdo. Caso seja link apresentará o link, caso seja um arquivo, representará o caminho onde está localizado este arquivo.

Voltando ao desafio, ele é composto por diversas perguntas representado pela entidade pergunta. A pergunta é composta pelo atributo pergunta que contém um texto e o atributo tipo que representa o tipo da pergunta, que pode ser múltipla escolha, única escolha ou dissertativa. Cada pergunta contém um conjunto de respostas representado pela entidade *item_pergunta*.

A resposta é composta pelo atributo *item*, que representa a resposta e o atributo *fl_resposta* que indica se está resposta é correta ou não. A entidade *jogador_resposta* representa as respostas dos jogadores, contendo o atributo *resposta* que é a resposta que o jogador deu no momento do jogo, o atributo *fl_acertou* indica se está certo ou não e o atributo *observacao* é quando o professor corrige a resposta do aluno colocando então suas considerações ou resposta certa.

A entidade *jogador* caracteriza tanto o jogador na hora do jogo como o usuário na hora de acessar o painel administrativo. O jogador contém atributos que caracterizam seu personagem como também a senha para acesso. O atributo *fl_mestre* indica se ele é considerado um mestre para poder cadastrar novos desafios no jogo ou não.

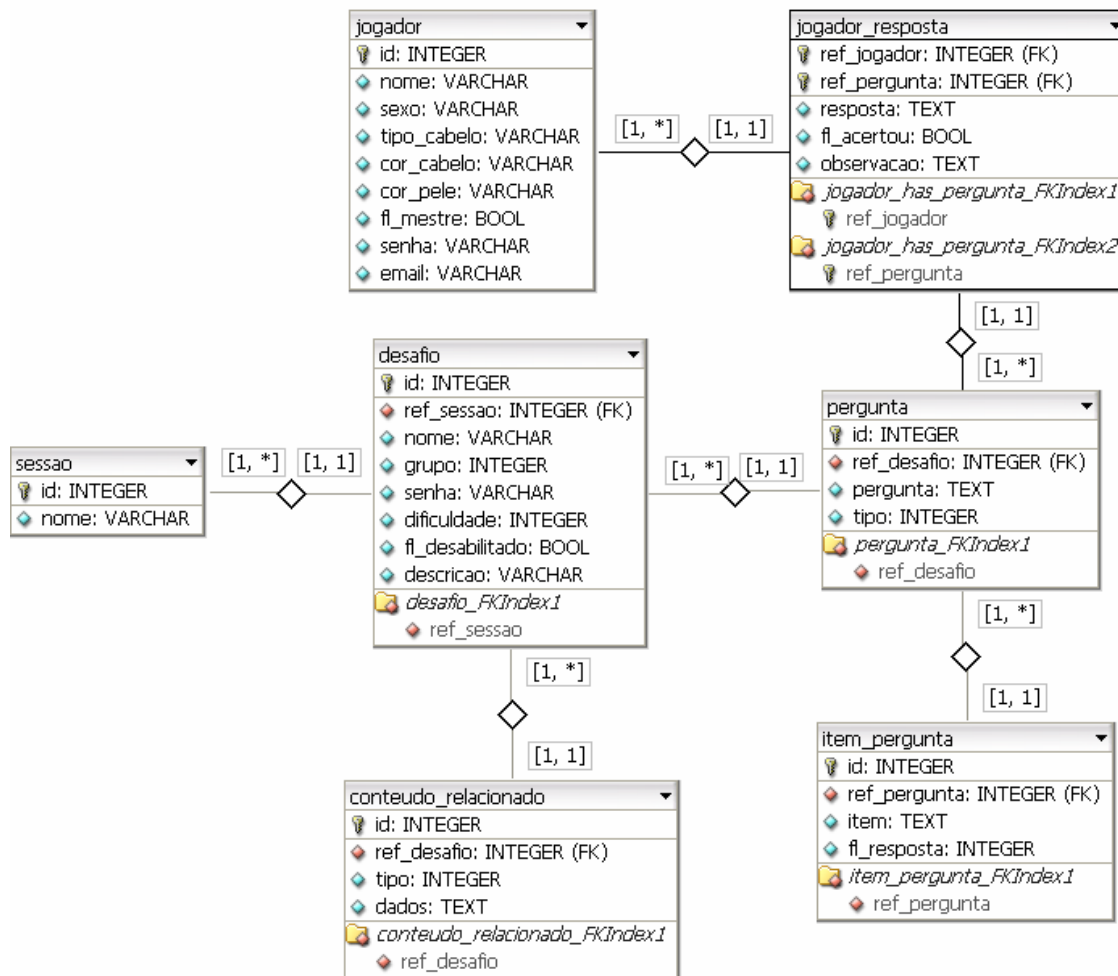


Figura 15: Diagrama ER.

3.5 Diagrama de Classes

Esta seção está constituída por dois diagramas que tem como objetivo representar as classes principais para implementação do sistema. Cada diagrama será explicado para entendimento geral da futura implementação do sistema, tanto o jogo quanto o painel administrativo.

3.5.1 Jogo

Para fazermos uma explicação do diagrama de classes do jogo (Figura 16) iniciamos pela classe *Principal* que representa o núcleo inicial do jogo. No momento em que o jogador acessar o jogo, a classe *Principal* irá carregar a estrutura base do jogo e mostrará a tela de autenticação representada pela classe *LoginView* que cuidará da autenticação do usuário. (Figura 18).

A classe *Principal* iniciará o controle de autenticação, carregamento inicial da aplicação, iniciar o mundo quando o usuário está autenticado e verificar se existe a necessidade de criação do avatar caso o usuário ainda não possua.

CriarAvatar apresentará as funções para a criação do avatar como personalização de algumas características e o nome do personagem (Figura 30). A classe *Menu* possuirá as chamadas das ações do menu assim como o conteúdo visual.

O mapa (Figura 31) está representado pela classe *Mapa* que conterà todas as regras relacionadas ao mundo. *MiniMapa* é a classe que controla o minimapa do jogo. *MapaSave* terá como objetivo persistir o mundo virtual no servidor e a classe *MapaLoader* serve para carregamento do mundo virtual quando solicitado pela classe *Mapa*. A classe *Jogador* está ligada diretamente ao mapa pois o mapa conterà toda a estrutura visual do mundo virtual, e o jogador em si tem uma representação visual no mundo.

A *SessaoView* representa a tela visual da listagem das seções (Figura 33) e o *DesafioView* é a tela visual do desafio acessado (Figura 34). O *DesafioView* está ligado com a classe *Chat* que tem como finalidade controlar toda as regras do *chat*.

PerguntasView e *FeedBackView* são os dois modos possíveis dos desafios (veja capítulo 3, Figura 34 e Figura 35). *PerguntasView* terá o controle das respostas dadas do jogador assim como solicitar a próxima pergunta do desafio. Quando concluído está etapa a classe *FeedBackView* apresentará os resultados obtidos pelo jogador.

A classe *ContedoView* representa a tela de conteúdos relacionados de um desafio acessado (Figura 36). Esta classe tem como objetivo organizar os dados cadastrados pelo professor e mostrar visivelmente na tela. Caso ele tenha cadastrado um *link*, um link deve aparecer, caso o mestre tenha cadastrado uma imagem ou um vídeo neste desafio, o conteúdo deve mostrar de forma visual.

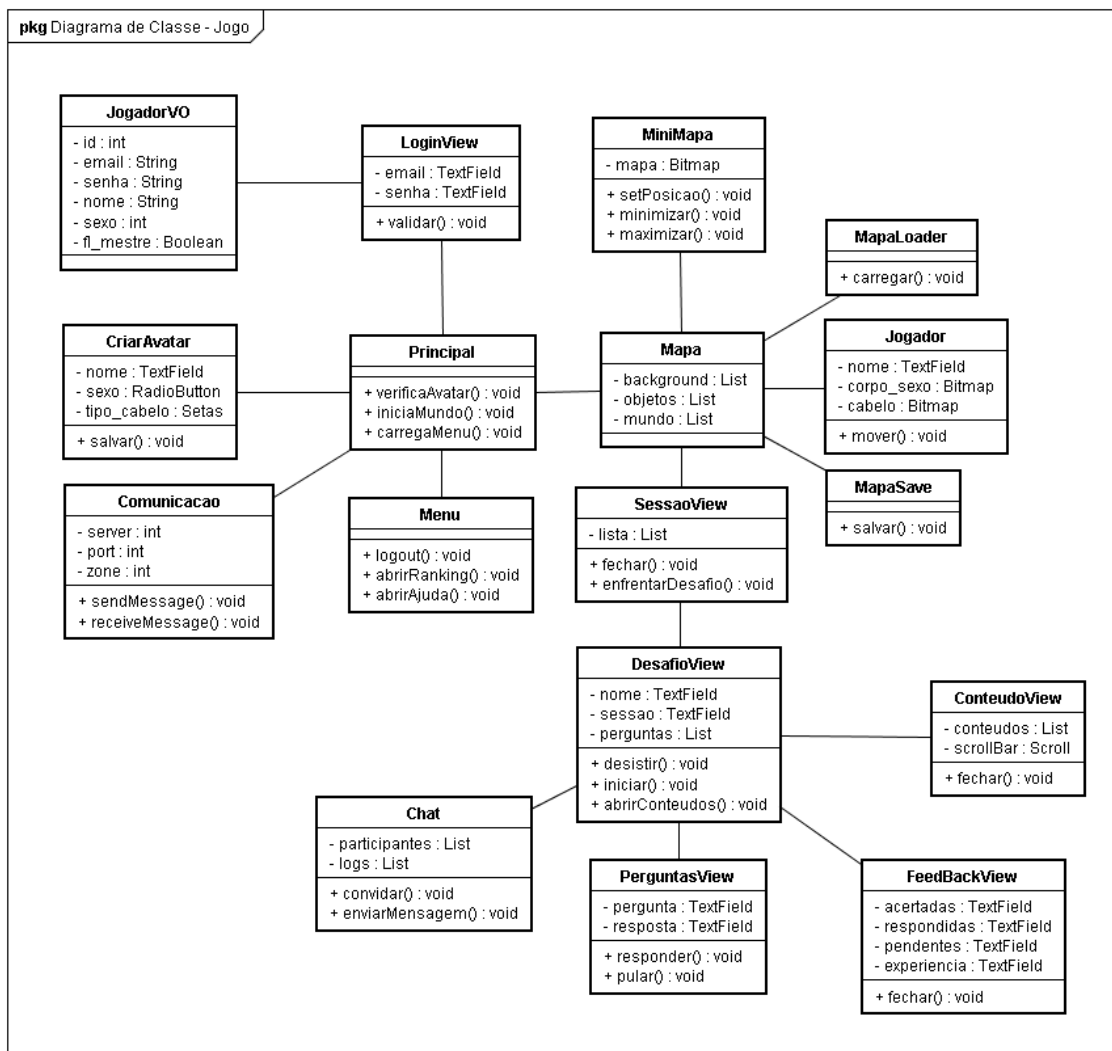


Figura 16: Diagrama de Classes – Jogo.

3.5.2 Painel

A análise do diagrama de classes do Painel (Figura 17) começa pela classe *PrincipalView*, que é o início da aplicação. Quando carregado a aplicação esta classe começa a desempenhar seu papel, solicitando a autenticação do usuário (Figura 18) e permitindo o acesso ao menu principal quando autenticado (Figura 19).

A classe *MenuView* apresenta a lógica necessária para carregar devidamente cada seção do painel como criar desafios, responder desafios e relatórios. A classe *DesafioView* apresenta toda a lógica necessária para o cadastro de um novo desafio no jogo e persistir no banco de dados (Figura 20, Figura 21 e Figura 23).

No momento em que se está cadastrando um desafio, existe o cadastro de conteúdos relacionados que suas ações estão na classe *ConteudoAddView* (Figura 24) e o cadastro de perguntas que estão relacionadas a classe *PerguntaAddView* (Figura 22).

A classe *ResponderView* (Figura 27) possui a listagem de perguntas que necessitam análise do professor. É nesta classe que ele faz a chamada para a classe *ResponderModalView* (Figura 26) que tem como objetivo controlar a resposta do professor a uma determinada pergunta e persistir as informações.

Todas as classes *View* apresentam seus devidos controladores que servem para manipular o negócio e destinar as informações para o *backend*, onde serão persistidos os dados.

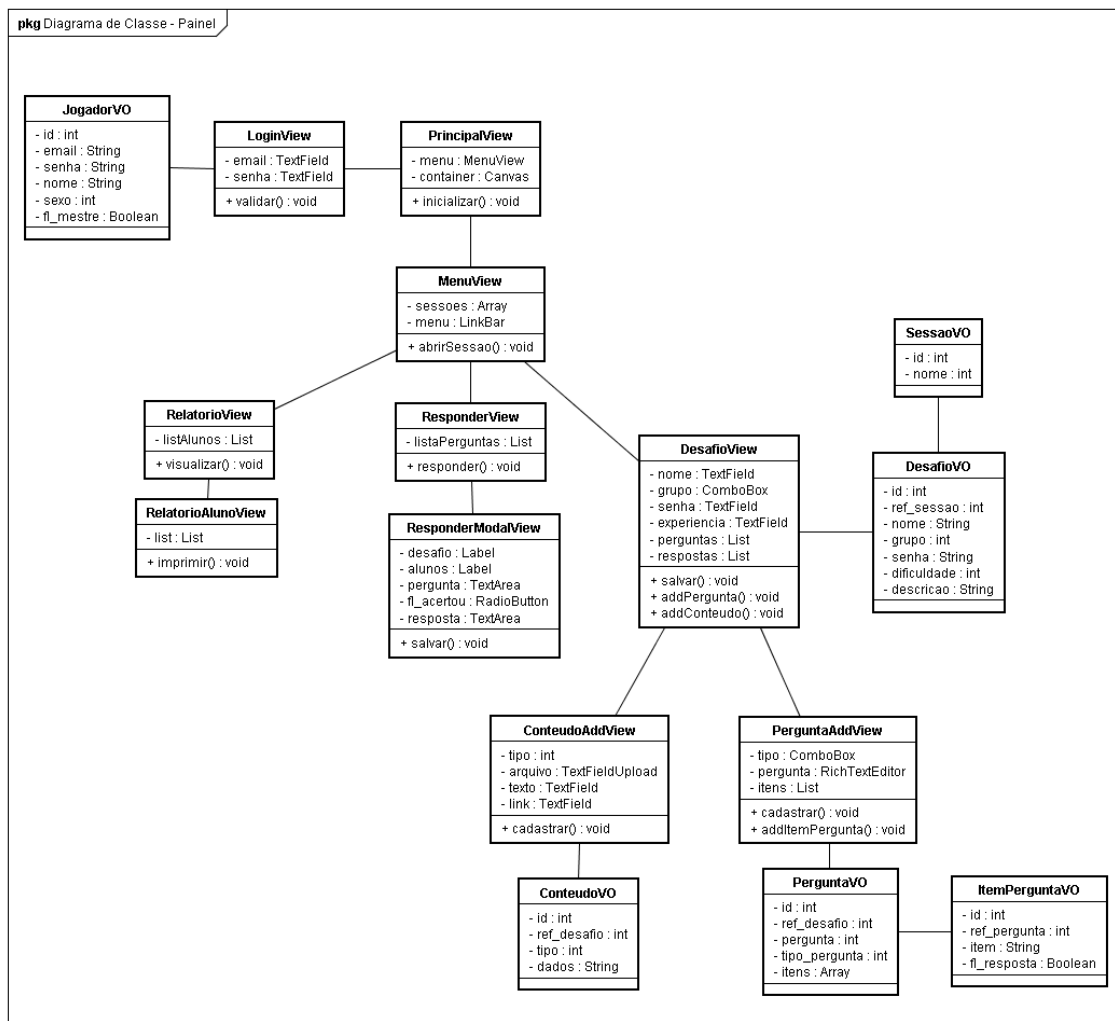


Figura 17: Diagrama de Classes – Painel.

3.6 Usabilidade e Ergonomia

Quando pensamos em desenvolver um jogo, que tem como característica servir como suporte para o aprendizado, existem diversos fatores que devem ser levadas em consideração.

A comunicação entre homem e máquina se torna indispensável e de grande importância, pois as mensagens trocadas entre eles, espelharão em ações dentro do jogo ou no painel de gerenciamento, logo estamos conectados diretamente a uma via de mão única que exige que a usabilidade seja empregada e bem avaliada para que o software possa apresentar qualidade e ser bem utilizado.

Usabilidade é a capacidade que um sistema interativo oferece a seu usuário, para a realização de tarefas, onde seja eficaz, agradável e eficiente (ISO 9241, 1993). É a qualidade de uso.

A ideia da explicação sobre usabilidade é levantar o conhecimento para fornecer subsídios em relação à construção do jogo e do painel gerencial, tendo como finalidade atender diferentes tipos de usuários e da melhor forma possível.

Cada usuário que acessa o sistema terá uma visão de como ele funciona e o que é preciso fazer para interagir, isto varia de acordo com a experiência passada, com todo seu conhecimento semântico (conceitos) e procedimental (procedimentos) e evoluem conforme ele vai trabalhando nessas ferramentas.

A interface do sistema deve ser flexível o suficiente para adequar-se aos diferentes tipos de usuário. Além do nível de conhecimento do usuário, deve ser levada em conta a percepção que o sistema oferece. Existem tipos de percepção que devem ser levadas em conta no desenvolvimento: visual, auditiva, fala, vigilância e atenção, memória, raciocínio e aprendizado.

O jogo deve conter um visual leve e que possa atender a diferença de idades. Na percepção auditiva, o som deve fazer sentido com o que está sendo mostrada visualmente e que eles tenham algum significado, a interação na hora de responder e acertar os desafios podem ser um exemplo.

Quando falamos em atenção e vigilância, o sistema deve produzir meios pelo qual o usuário possa saber que ocorreu algum erro no sistema, ou que possa saber como está o andamento do processo.

Para o tipo de percepção memória é interessante fornecer um sistema com padrões de interface, erros, interações, para que as ações possam ser facilmente percebidas.

O raciocínio produz novas informações a partir das existentes, logo o sistema deve fornecer formas que facilite o entendimento do usuário e até mesmo formas de buscar as informações.

Quando falamos em usabilidade não podemos deixar de abordar o tema ergonomia, que é uma disciplina voltada ao entendimento das interações entre homem e máquina.

Existem oito critérios bases desenvolvidos por Scapin e Bastien (1993) que tem como objetivo minimizar a ambiguidade na identificação e classificação dos problemas ergonômicos do sistema interativo. Alguns dos itens listados abaixo são ainda reafirmados por Dias (2003) e Winckler (2001).

- **Condução:** O software informa e conduz o usuário na interação. Existem alguns aspectos como a prestação que é o acesso rápido a ações que devem ser desenvolvidas no software, facilitando a navegação e ações em geral. Temos o feedback imediato, que é a resposta do sistema quando o usuário faz determinadas ações. Outro aspecto é a legibilidade apresentada para o usuário. Por último, o agrupamento/distinção de itens, que é o entendimento da interface, ela depende de vários fatores como posicionamento, distinção de seções, ordenação, é importante que siga um padrão.
- **Carga de trabalho:** Quanto maior for o tempo que ele deve interagir para fazer determinada ação, mais chance de cometer algum erro. As entradas de informação para o usuário devem ser sucintas e breves, facilitando os passos do usuário para uma determinada ação. Outro item importante a ser levado em consideração é a quantidade de informação que o usuário deve armazenar para fazer diversas ações, quanto mais complexas elas forem, pior para o usuário final.
- **Controle explícito:** as ações devem ser processadas quando o usuário solicitar e a cada ação do usuário, o sistema deve fornecer os próximos passos, não forçando um erro para voltar e refazer o processo.
- **Adaptabilidade:** Forma pelo qual o sistema reage com o ambiente. Uma interface deve ser flexível para atender diferentes tipos de usuário (básicos e avançados, por exemplo) e ações que ele pode fazer com o sistema, dando diversos caminhos para um mesmo fim.
- **Gestão de Erros:** Gerenciar, evitando erros ou tratando antes de mostrar ao usuário.
- **Coerência / Homogeneidade:** As seções quando equivalentes, devem fornecer os mesmos recursos, códigos, posicionamentos, formatos, caso a seção seja diferente,

então ela pode ser apresentar diferente, facilitando o entendimento e manuseio do usuário.

- **Significado dos códigos e denominações:** A relação entre objetos e suas informações de referencia devem fazer sentido no contexto.
- **Compatibilidade:** Refere-se a compatibilidade do usuário com o sistema apresentado, tanto na questão de idade, hábitos, similaridades entre os ambientes, como em questões cognitivas. O desempenho é maior quando a informação apresentada está nos padrões do usuário.

3.7 Prototipagem

Está sessão tem como objetivo apresentar a prototipagem do painel administrativo e do jogo. A prototipagem do painel é mais avançada, já que foi utilizado a mesma ferramenta em que ele será programado. O protótipo do jogo é apenas um rascunho visando as suas principais características.

3.7.1 Painel

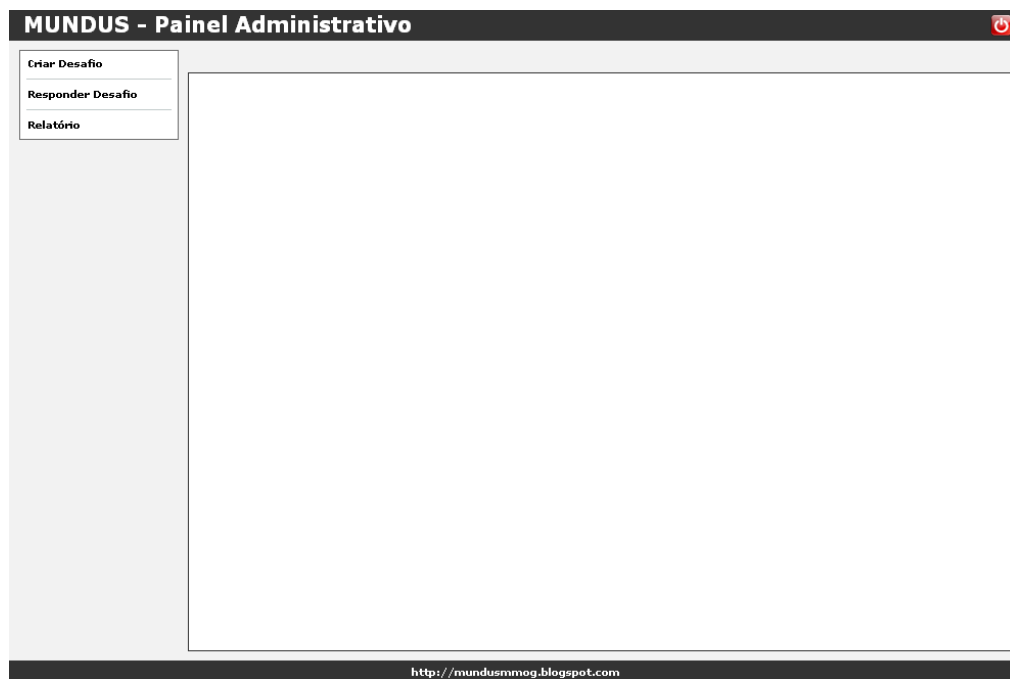
Quando acessado o painel administrativo, a primeira tela que o usuário terá contato é a tela do *login* (Figura 18). Neste momento ele informará seu email e sua senha para poder logar. Quando autenticado ele poderá navegar dentro do sistema. A navegação ocorre através do menu esquerdo, composto pelas opções: criar desafio, responder desafio e relatório (Figura 19).

Quando o usuário acessar a opção “criar desafio”, aparecerá o conteúdo para ser cadastrado. As informações do desafio são divididas em 3 abas: informação, perguntas, conteúdos relacionados.



Protótipo de uma tela de login. O formulário tem um cabeçalho "Login" em um fundo escuro. Abaixo dele, há dois campos de entrada: "E-mail" com o texto "admin" e "Senha" com "*****". Cada campo tem um asterisco vermelho à esquerda. Abaixo dos campos, há um botão "Logar" com um efeito de sombra.

Figura 18: Protótipo Painel – Login.



Protótipo de uma tela de painel administrativo. O cabeçalho é "MUNDUS - Painel Administrativo" com um ícone de power no canto superior direito. À esquerda, há um menu com três itens: "Criar Desafio", "Responder Desafio" e "Relatório". O restante da tela é uma grande área vazia. Na base, há uma barra de endereço com o URL "http://mundusmmog.blogspot.com".

Figura 19: Protótipo Painel – Estrutura Base.

A primeira aba é composta por nome, seção, grupo, dificuldade, descrição e senha (Figura 20). A senha é o único campo não obrigatório. Quando o usuário preencher os dados ele poderá clicar no botão prosseguir ou selecionar a segunda aba.

MUNDUS - Painel Administrativo

Desafio

1) Informações 2) Perguntas 3) Conteúdos

1 Passo - Informações sobre o Desafio

Nome *

Sessão * História

Grupo * Nenhum

Dificuldade * Fácil

Descrição (Curta) *

Senha

2 Passo - Perguntas

<http://mundusmmog.blogspot.com>

Figura 20: Protótipo Painel – Criar Desafio – Informações.

O segundo item da aba é perguntas (Figura 21). Nesta aba ele cadastra novas perguntas clicando no botão adicionar nova pergunta e também ver a listagem de todas as perguntas já cadastradas para o desafio. Caso ele deixe o mouse sobre um item da listagem (uma pergunta), um pequeno retângulo aparecerá contendo as informações referentes a pergunta.

MUNDUS - Painel Administrativo

Desafio

1) Informações 2) Perguntas 3) Conteúdos

1 Passo - Cadastrar Perguntas

Adicionar Nova Pergunta

Pergunta:	Tipo da Pergunta:	Quantidade de Respostas:
Pergunta Aqui	Múltipla Escolha	3

Pergunta: Pergunta Aqui
 Tipo: Múltipla Escolha
 Respostas: Resposta: teste Correta? Sim
 Resposta: não Correta? Sim
 Resposta: talvez Correta? Sim

2 Passo - Conteúdos

<http://mundusmmog.blogspot.com>

Figura 21: Protótipo Painel – Criar Desafio – Perguntas.

No momento em que o usuário clicar na opção “cadastrar perguntas”, abre uma nova tela (Figura 22). Deve-se então escolher o tipo de pergunta, escrevê-lo e adicionar as respostas, caso seja de múltipla escolha ou de única escolha. A tela se fecha e um novo item aparecerá na listagem de perguntas do desafio.

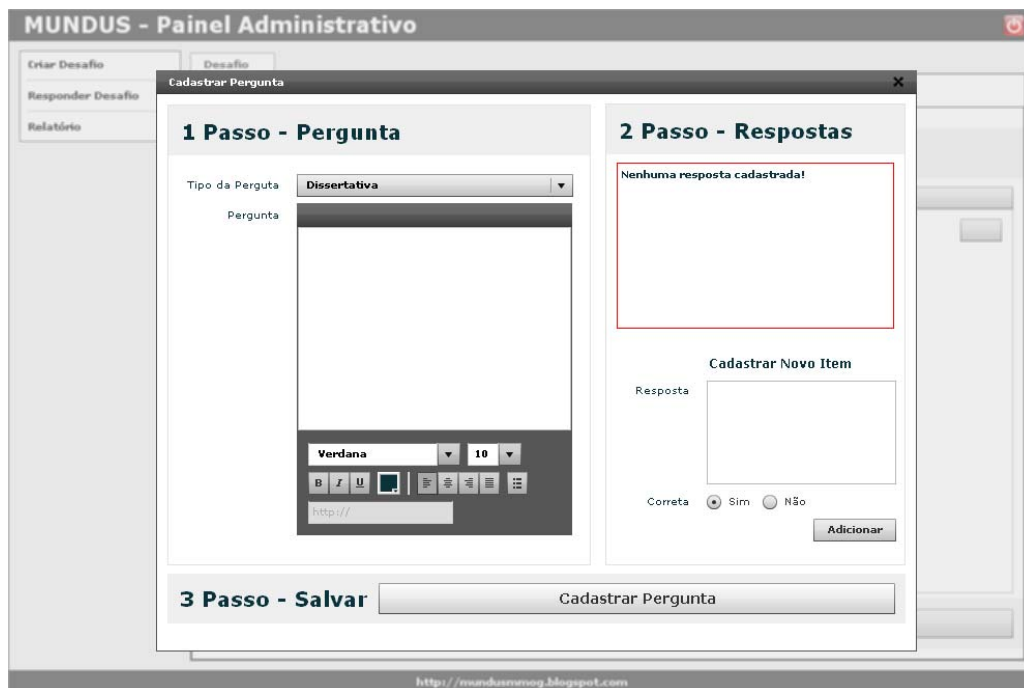



Figura 22: Protótipo Painel – Perguntas – Cadastro.

Após concluído esta etapa, ele pode selecionar a 3ª aba ou clicar em prosseguir. Nesta tela (Figura 23) ele pode clicar no botão “adicionar novo conteúdo relacionado”, ver a listagem de conteúdos já cadastrados para este desafio ou salvar o desafio.



MUNDUS - Painel Administrativo

Criar Desafio

Responder Desafio

Relatório

Desafio

1) Informações 2) Perguntas 3) Conteúdos

1 Passo - Conteúdos

Adicionar Novo Conteúdo Relacionado

Tipo Conteúdo:	Dados:
Link	Ucs
Texto	Texto sobre o assunto relacionado.

Ucs

2 Passo (Último) - Salvar Salvar

<http://mundusmmog.blogspot.com>

Figura 23: Protótipo Painel – Criar Desafio – Conteúdos Relacionados.

Quando acessada a opção de cadastrar um novo conteúdo relacionado (Figura 24), o usuário pode escolher o tipo do conteúdo relacionado e então cadastrar os dados referentes ao tipo, que podem ser de 3 tipos: link, texto ou arquivo (vídeo/imagem). Quando o usuário mestre clicar em cadastrar conteúdo, o item será adicionado na listagem de conteúdos.

Quando o mestre clicar em salvar desafio o sistema validará as informações e então as salvará. Caso os dados não estejam corretos o sistema mostrará um aviso na tela direcionado o usuário.

MUNDUS - Painel Administrativo

Criar Desafio
Responder Desafio
Relatório

Desafio

1) Informações 2) Perguntas 3) Conteúdos

1 Passo - Conteúdos

Conteúdo Relacionado

Tipo
Link
Tipo
Text

1 Passo - Pergunta

Tipo do Conteúdo: Link

Link:

Titulo:

2 Passo - Salvar Cadastrar Conteúdo

2 Passo (Último) - Salvar Salvar

http://mundusmmog.blogspot.com

Figura 24: Protótipo Painel – Conteúdo Relacionado – Cadastro.

A segunda opção do menu é “responder desafio” (Figura 25). Quando acessada esta opção, o usuário terá uma relação de todas as perguntas dissertativas respondida pelos jogadores que ainda não foram gerenciadas pelo mestre. Neste momento, o usuário poderá escolher uma das questões e clicar no botão responder.

MUNDUS - Painel Administrativo

Criar Desafio
Responder Desafio
Relatório

Responder

Escolha a pergunta e clique em Responder

Desafio	Alunos	Pergunta	Responder
desafio	alunos	pergunta	Responder

http://mundusmmog.blogspot.com

Figura 25: Protótipo Painel – Responder Desafio - Listagem.

Depois de clicado no botão responder será aberto uma nova tela (Figura 26) que mostrará o desafio, o aluno que respondeu a questão e a pergunta. Abaixo terá um campo onde o mestre deverá digitar a resposta ou alguma observação e selecionar se o jogador acertou ou não a pergunta. Depois de seguidos esses passos ele salvará a questão e este item sairá da listagem.

The image shows a web application interface titled "MUNDUS - Painel Administrativo". On the left, there is a sidebar menu with options: "Criar Desafio", "Responder Desafio", and "Relatório". The main content area is titled "Responder" and contains a table with columns "Desafio" and "Alunos". A modal dialog box titled "Responder Desafio" is open, showing the "1 Passo - Resposta" screen. The dialog contains the following fields and controls:

- Desafio: desafio aqui
- Alunos: alunos aqui
- Pergunta: [Empty text area]
- Resposta: [Radio button selected] Acertou [Radio button] Errou
- [Empty text area for the answer]
- 2 Passo - Salvar: [Salvar Resposta button]

At the bottom of the dialog, there is a URL: <http://mundusmang.blogspot.com>

Figura 26: Protótipo Painel – Responder Desafio - Cadastro.

O último item do menu principal é o relatório (Figura 27). Neste item o mestre terá poder de selecionar um jogador e então clicar no botão visualizar para ver o relatório do jogador. No momento em que ele clicar abrirá uma nova tela (Figura 28) que mostrará um resumo das informações e ainda dar a opção de impressão.



Figura 27: Protótipo Painel – Relatório – Listagem Professor.

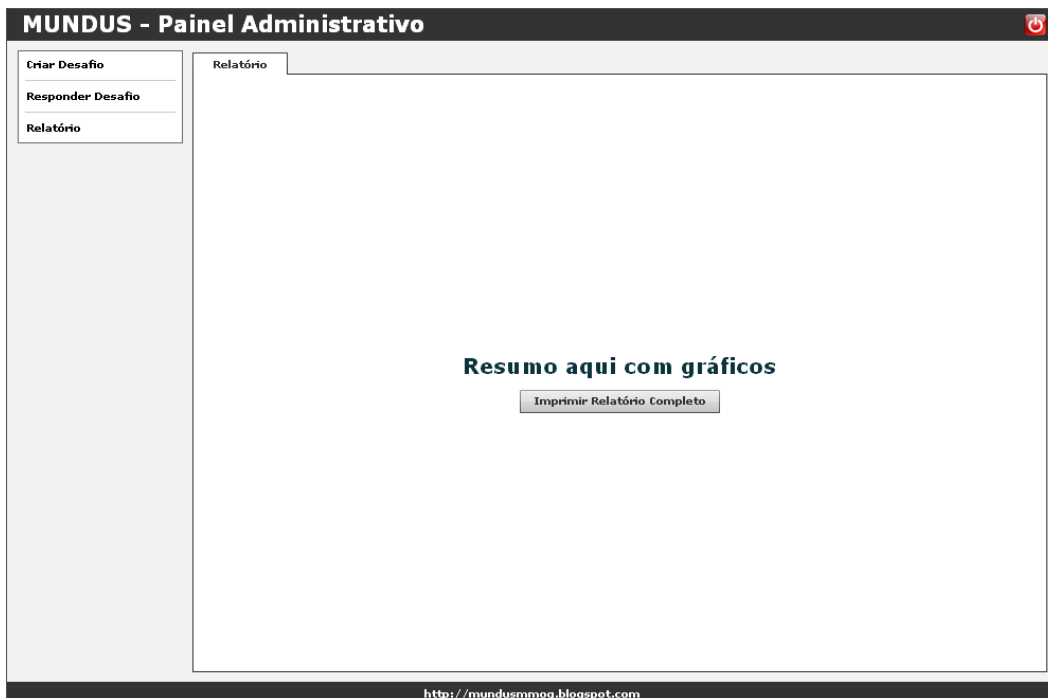


Figura 28: Protótipo Painel – Relatório – Informações.

3.7.2 Jogo

A primeira tela que o jogador terá contato será a de autenticação (Figura 29), que pedirá para o jogador informar seu *login* e senha. Vale ressaltar que o jogador pode ser tanto o professor quanto o aluno. Quando acessado, o jogo verificará se o jogador já possui um avatar (características do personagem) criado, caso não possua, uma tela se abrirá (Figura 30) solicitando nome e seleção de algumas características do personagem, quando concluído o jogador clicará em entrar.

800x600

Imagem de Fundo

Login

Senha

Entrar

Detailed description: This is a wireframe for a login screen. It features a large rectangular area labeled 'Imagem de Fundo' (Background Image) in the center. At the bottom left, there are two input fields: one for 'Login' and one for 'Senha' (Password). Below these fields is a button labeled 'Entrar' (Enter).

Figura 29: Protótipo Jogo – Login.

Personalização do Personagem

Nome

Masculino Feminino

Outras Propriedades:
Cores / Tipos

Entrar

Detailed description: This is a wireframe for a character creation screen. At the top, it says 'Personalização do Personagem' (Character Customization). Below this is a text input field for 'Nome' (Name). Underneath are two buttons: 'Masculino' (Male) and 'Feminino' (Female). In the center is a stick figure with four triangular arrows pointing outwards (up, down, left, right) for customization. At the bottom left, it says 'Outras Propriedades: Cores / Tipos' (Other Properties: Colors / Types). At the bottom center is an 'Entrar' (Enter) button.

Figura 30: Protótipo Jogo – Criação do Personagem.

Neste momento o usuário estará dentro do jogo, visualizando um pedaço do mundo virtual (Figura 31). Conforme ele vai andando com o seu personagem, ele vai visualizando o resto do mapa. Neste momento ele poderá interagir com os outros jogadores conversando com eles.

A tela principal é dividida em três partes principais. No canto superior direito estará mostrando um mini-mapa do mundo, no centro mostrará o mundo virtual e na parte inferior da janela, os seus dados mais relevantes e as ações principais.

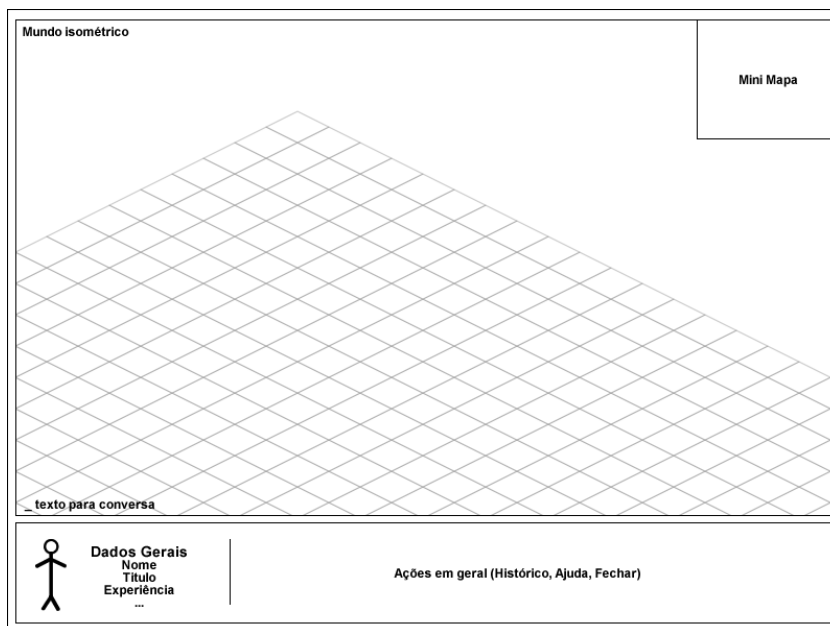


Figura 31: Protótipo Jogo – Estrutura.

O jogador, mesmo tendo acesso ao mini-mapa, poderá clicar nele e ver a representação de todo o mapa do jogo (Figura 32). Conforme o jogador vai andando pelo mundo, ele poderá acessar diferentes seções que estarão em diferentes locais. Quando ele acessar uma seção (Figura 33) aparecerá uma listagem de desafios. Esta listagem conterá todos os desafios daquela seção, mesmo os que já foram respondidos. Os que já foram respondidos, que terão uma sinalização diferente.

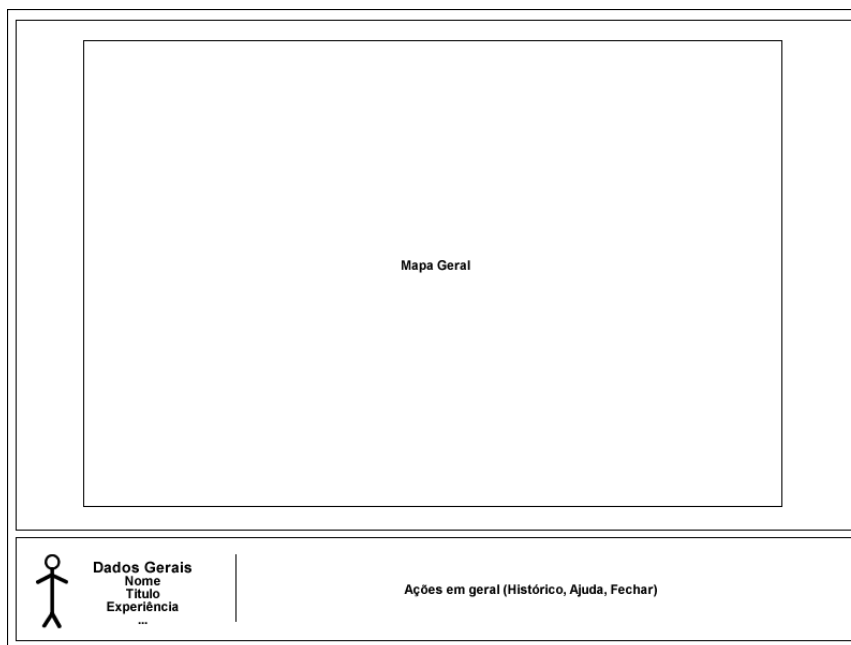


Figura 32: Protótipo Jogo – Mapa.

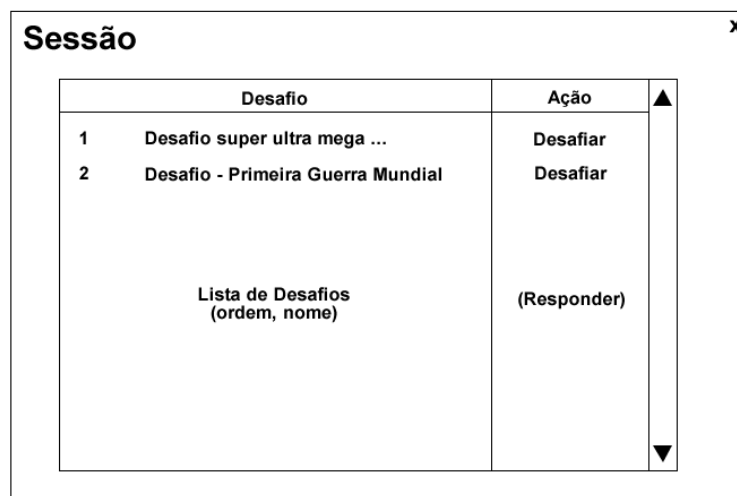


Figura 33: Protótipo Jogo – Sessão.

A partir desta tela (Figura 33) o jogador poderá escolher qual desafio ele deseja enfrentar e então clicar em desafiar. Depois de escolhido o desafio, uma nova tela se abrirá (Figura 34). Esta tela conterá um botão para acesso aos conteúdos relacionados que abrirá

uma nova tela (Figura 35) mostrando todos os conteúdos relacionados a este desafio. Os conteúdos poderão ser vídeo, textos, imagens e links de acesso.

Voltando a tela das respostas do desafio (Figura 34), podemos ver um minichat na parte direita, que apresenta a lista dos participantes, a interação entre eles e um botão para convidar novas pessoas para aquele desafio. Vale lembrar que ele só poderá convidar outras pessoas desde que o mestre que tenha cadastrado este desafio tenha atribuído que pode haver grupos no desafio e quantas pessoas podem participar.

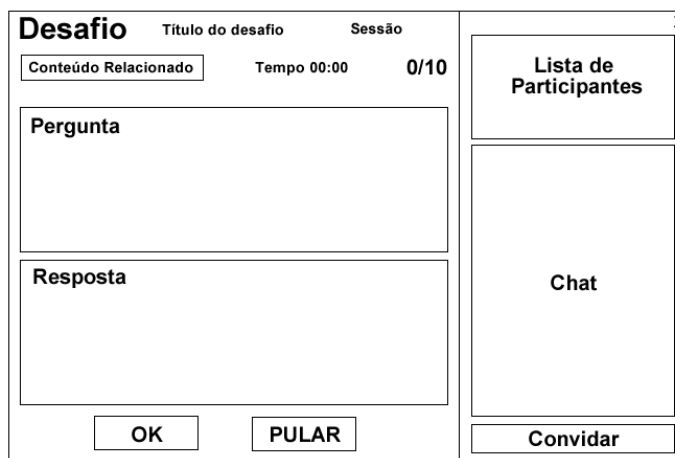


Figura 34: Protótipo Jogo – Desafio

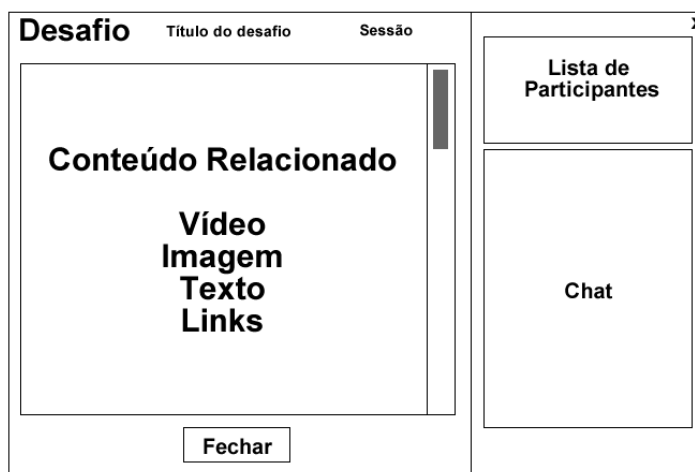


Figura 35: Protótipo Jogo – Conteúdo Relacionado.

No centro da tela se apresenta a pergunta corrente com um campo para o jogador selecionar ou escrever a resposta. Conforme vai respondendo pode-se clicar em *ok* ou pular a questão. Se ainda existir outra nova questão, então ela será mostrada na tela, caso não exista mais, é mostrado um feedback ao jogador (Figura 36) com as questões que ele acertou, errou e que necessita uma análise de um mestre, para os casos das questões dissertativas. Neste momento ele ganha pontos de experiência.

Desafio	Título do desafio	Sessão
Você acertou		8 + 000 exp
Errou		2 + 000 exp
Necessita Análise <small>Aguarde resposta do professor.</small>		0

OK

Lista de Participantes

Chat

Figura 36: Protótipo Jogo – Feedback.

Por fim, dentro do jogo o usuário ainda poderá ver o *ranking* de todos os jogadores (Figura 37), o qual só poderá ser acessado em uma determinada parte do mundo.

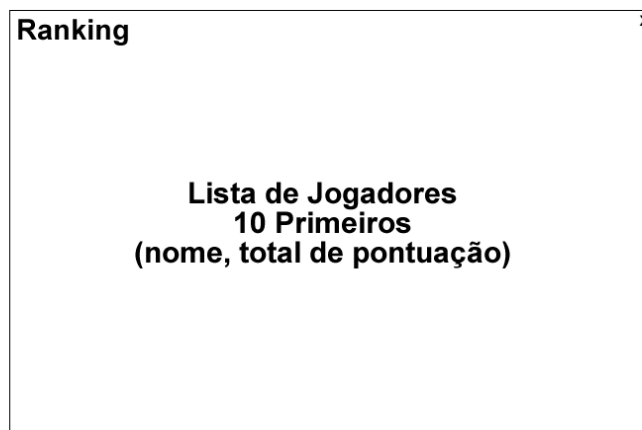


Figura 37: Protótipo Jogo – Ranking

3.8 Arquitetura

Visando a criação de um ambiente de aprendizagem cooperativo, através de um jogo do estilo MMOG é necessário diagramar uma arquitetura que consiga contemplar toda a necessidade da aplicação.

Algumas APIs serão utilizadas para facilitar o desenvolvimento do jogo, e também um *middleware* para construção do servidor do jogo e para a comunicação entre os jogadores.

O painel de gerenciamento será desenvolvido sobre uma arquitetura em três camadas. A camada mais alta, de visualização, é onde existe a interação do sistema com o usuário e vice-versa. Nesta camada não existem regras de negócio e nem acesso direto aos dados.

A camada intermediária apresenta todas as regras de negócio, ela receberá as requisições da camada de visualização, processará as informações e quando necessário acessará os dados.

A última camada servirá apenas para acesso aos dados, onde será utilizado um banco de dados para armazenamento das informações, como usuários do sistema, professores, os desafios. Esta última camada receberá as requisições apenas da camada de negócio.

Quanto ao jogo, será utilizado o mesmo princípio abordado no painel de gerenciamento e cadastro de conteúdo, camada de visualização, negócio e persistência.

A Figura 38 ilustra a arquitetura do sistema proposto. Os clientes através do navegador irão conectar na aplicação cliente do jogo ou no painel administrativo que irão transmitir e requisitar informações para diferentes destinos como o servidor do jogo, o banco de dados e arquivos em geral.

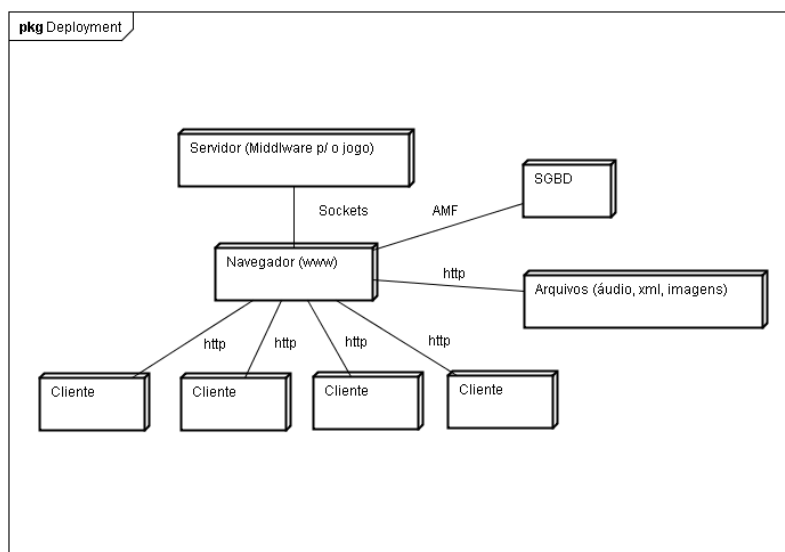


Figura 38: Estrutura básica da aplicação.

3.8.1 Linguagem e Ferramenta

A linguagem empregada para desenvolvimento do jogo e para a construção do painel é o Actionscript 3. Ela é uma linguagem de programação que está especificada no padrão internacional ECMA-262, além de possuir suporte a outras tecnologias como XML, Webservices e Javascript.

O Flash Player roda nos browsers (navegador web) e ele consegue rodar em sistemas operacionais como Windows, Linux, outros derivados de Unix e MAC.

O Flash Player é largamente utilizada hoje em dia nos computadores. Na Figura 39 mostramos uma comparação do player da Adobe em relação a outras empresas. Hoje ele está presente em mais de 99% de computadores conectados na Internet segundo o site da Adobe⁴.

⁴ http://www.adobe.com/products/player_census/flashplayer/ - comparativo acessado em 20/06/09.

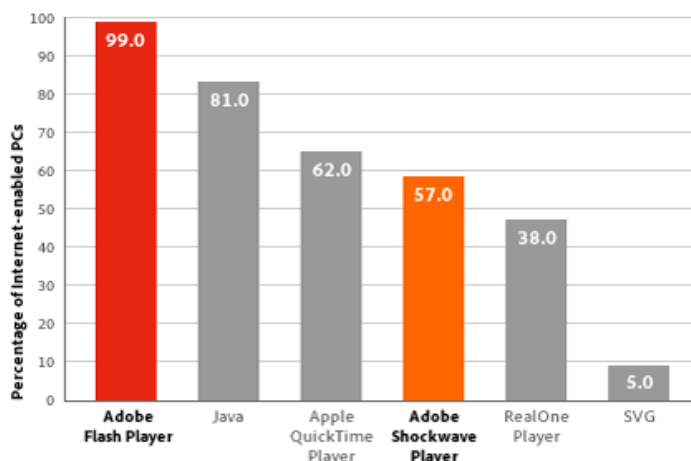


Figura 39: Comparativo de tecnologias.

Visando a construção de um jogo que deve ser graficamente rico, deve garantir a portabilidade em diferentes sistemas operacionais e que seu acesso seja facilitado através da WEB, a tecnologia Flash apresenta uma boa alternativa para atender essas necessidades, pois fornece suporte multiplataforma e multibrowser.

Atualmente, diversos jogos de complexidade baixa são desenvolvidos nesta tecnologia, pois ela fornece um grande número de funcionalidades na manipulação de gráficos e sons, facilitando o desenvolvimento.

No início do ano de 2009, o *Flash Player* trouxe diversos recursos consigo, porém o que mais chamou atenção foi a adição do eixo Z nativamente em seus objetos visuais, facilitando ainda mais o desenvolvimento de ambientes que simulam o 3D.

A camada de negócio do jogo será construída na linguagem Actionscript e para persistência de dados será utilizada a linguagem PHP com a extensão PDO (Php Data Object).

Ainda para o desenvolvimento do trabalho será utilizada a tecnologia Flex, que serve para o desenvolvimento de aplicações RIA. Note que o Flex utiliza também a linguagem Actionscript 3.

A utilização do Flex será para a construção do painel de gerenciamento e cadastro de conteúdos, ele servirá como a primeira camada (camada visual) da aplicação, fazendo então a comunicação com as outras camadas para validar as informações e persistir os dados.

Quando o programa é compilado, ele gera um arquivo *swf* que o *Flash Player* poderá ler e então estruturar internamente e mostrar na tela para o usuário o arquivo que ele acessou.

3.8.2 Banco de Dados

Para persistência dos dados será utilizado o banco de dados MySQL, pois é um banco freeware, sem limitações de tamanho. Ele existe à um tempo no mercado, tem uma marca consolidada e atende todas as necessidades previstas na utilização do projeto.

3.8.3 Comunicação Flex-Php e Flash-Php

Tanto para o jogo, quanto para o painel, há momentos em que será necessário acessar a base dos dados, logo a tecnologia de comunicação empregada será a mesma. A tecnologia escolhida para fazer esta ponte de comunicação é a utilização do AMFPHP que usa o protocolo AMF3, que provê a comunicação binária entre as tecnologias e possui também compactação de dados para realizar a transferência dos dados de forma mais rápida.

Outro fator importante a ser considerado é a possibilidade de fazer o mapeamento de objetos entre a linguagem AS3 e PHP, facilitando e agilizando ainda mais o desenvolvimento.

O protocolo AMF é um RPC que é implementado no Flash e prove uma comunicação de dados mais eficiente. Poderiam ser utilizados para a comunicação JSON ou até mesmo XML.

Existem vários aspectos que podem influenciar a passagem dos dados de uma camada para outra. Segundo o teste de Mike Potter⁵ o XML apresentou um bom tempo de envio enquanto JSON apresentou comparado aos outros resultados o mais demorado, porém o cenário era local, não enfrentando a maior realidade das aplicações.

A tabela abaixo mostra testes desenvolvidos por Leonardo França (2006) sobre o tempo de transição de dados em XML, AMFPHP e JSON. Ele não especifica qual o tamanho de cada dado trafegado em bytes, porém o cenário é remoto e segundo ele é o mesmo teste realizado por Miker (2006).

Tabela 2: Tempo de envio dos dados

Quantidade de dados (item)	XML*	AMFPHP*	JSON*
2500	4406	3015	3855
5000	6820	5378	6500
7500	9653	7281	9564

⁵ http://blogs.adobe.com/mikepotter/2006/07/php_and_flex_js.html

10000	12428	9544	12458
15000	20399	13709	error
20000	25967	18977	error
25000	33708	24045	error

* Tempo em ms.

O autor do blog [5 1/2](#) ⁶, defendeu a tecnologia JSON e o protocolo AMF, abordando todos os aspectos, tanto em questão de dados trafegados, tempo de desenvolvimento, portabilidade para outras aplicações, cenários mais realísticos e segurança, tendo como conclusão que se for desenvolver em Javascript utilize JSON, se for desenvolver em Actionscript utilize AMF e se for apenas para receber informações como RSS utilize então XML.

3.8.4 Fluxo Entre Camadas

O fluxo de dados entre as camadas do jogo acontece assim:

1. O Flash chama uma ação para a camada de negócio;
2. A camada de negócio (em linguagem AS3) processa a informação, faz a validação das regras e retorna um erro ou procede para o servidor multiplayer;
3. Neste momento, é acessada a lógica de negócio do servidor, que irá processar a informação e retornar ao usuário ou fazer uma persistência;
4. Até o item 3 as chamadas eram todas síncronas, mas se há necessidade de uma persistência, então lançado uma chamada assíncrona através do protocolo AMF3 para a camada de negócios do PHP ou enviado por POST pelo protocolo HTTP;
5. Os dados na camada de destino serão validados e encaminhados para a camada de dados;
6. Retorna o estado para a camada de visualização, com alguma mensagem caso enviado pela camada de dados ou regra;

O fluxo de dados entre as camadas do painel acontece de uma forma diferente, da seguinte forma:

1. O usuário faz alguma ação no painel administrativo que necessita a requisição de alguma informação ou envio para o banco de dados.
2. O Flex chama um serviço do PHP através do protocolo AMF3;

⁶ <http://www.5etdemi.com/blog/archives/2006/12/clearing-the-fud-on-amfphps-speed-versus-json-and-xml>

3. O método chamado no Flex já está acessando a camada de negócio, que fará as validações e regras necessárias;
4. Caso exista algum erro no processo, este é reportado à camada de visualização;
5. Caso não exista nenhum erro, a camada de negócio prossegue verificando a necessidade de fazer a persistência de algum dado;
6. Se houver a necessidade de persistência, a camada de negócio dispara uma solicitação para a camada de dados e então processa a informação;
7. Após a conclusão ou erro, o estado é retornado para a camada de negócio que conseqüentemente retorna a camada de visualização;

3.8.5 Tratamento de Erros

O sistema em geral, tanto o jogo quanto o painel deve tratar todos os erros que possam ocorrer, sendo eles críticos ou não. Quando encontrado o erro, caso não consiga tratar, deve ser lançado uma mensagem ao usuário informando a causa.

3.8.6 Arquivos

A persistência do mundo virtual e alguns parâmetros do sistema serão salvos no banco de dados e na hora de leitura serão convertidos para XML. A escolha desta tecnologia visa a rápida e forte integração nativa com o Flash.

O XML apresentará os caminhos das imagens que devem ser carregados para montar o cenário, facilitando assim a alteração de qualquer imagem do jogo quando necessário.

Outras extensões de arquivos que serão utilizados:

- **.mp3:** som para serem usados como som ambiente, sons do personagem, objetos e etc. Para a utilização do som *mp3* no Flash, o Flash será compilado na versão 10.
- **.js:** São arquivos Javascript que servirão para testar se o cliente possui o *Flash Player* necessário para rodar o jogo ou acessar o painel.
- **.html:** Arquivo base para carregamento da aplicação.
- **.swf:** Arquivo gerado que apresenta todo o núcleo da aplicação.
- **.jpg, .png, .gif:** Arquivos de imagens presentes no painel e no jogo.
- **.as:** Arquivos fontes da aplicação, visando a parte visual e de negócio.

- **.php:** Arquivos fontes da aplicação, para interligar o Flash com o *backend* e possibilitar a persistência de dados.
- **.swc:** Bibliotecas compiladas para utilização no Flash.

3.8.7 Organização de Pastas

Nesta seção apresenta duas tabelas (Tabela 3 e 4) que descrevem a organização de pastas da implementação do trabalho, ajudando a entender a estrutura das pastas.

Tabela 3: Painel

Pasta/Arquivo	Descrição
\bin	projeto compilado
\bin\amf	classes php para ligação do flex com o backend
\bin\css	css do projeto com as imagens
\bin\index.html	arquivo para abertura do painel
\src	código fonte do projeto
\lib	bibliotecas

Tabela 4: Jogo

Pasta	Descrição
\bin	projeto compilado
\bin\js	arquivos javascript
\bin\images	imagens do projeto em geral
\bin\swf	arquivos compilados dos .fla da pasta \fla
\bin\xml	XML do projeto
\bin\index.html	abertura do jogo
\src	código fonte
\fla	arquivos visuais para serem compilados
\lib	bibliotecas

3.8.8 Logs

O *log* (arquivo de registro) do jogo consistirá em apresentar o trajeto do aluno durante o jogo. Ele irá apresentar quais os desafios acessados, quantas questões corretas e incorretas,

quantas vezes foram acessados os conteúdos relacionados em um determinado desafio, se havia uma formação de equipe no momento em que o desafio estava sendo respondido.

Todas as interações serão salvas no banco de dados, conforme a tabela 5.

Tabela 5: Logs

Tipo	Identificação do Jogador	Dados (Id do desafio, conteúdo relacionado, pergunta ...)	Data
Acesso no desafio	01	5 (identificação)	10/10/2000 10:10:00
Acesso no conteúdo relacionado	01	3 (identificação)	10/10/2000 10:11:00
Acesso em qual item do conteúdo relacionado	01	10 (identificação)	10/10/2000 10:12:00
Tempo no desafio	01	2040 (em segundos)	10/10/2000 10:13:00
Tempo na pergunta	01	10 (em segundos)	10/10/2000 10:14:00
Acertou pergunta	01	10 (identificação)	10/10/2000 10:15:00
Errou pergunta	01	11 (identificação)	10/10/2000 10:16:00
Equipe no desafio	01	01#01/02/03 (identificação do desafio e das pessoas que formam a equipe)	10/10/2000 10:17:00

3.8.9 Gráfico do Jogo

A projeção isométrica é uma técnica utilizada no início dos anos 80, onde através dela é possível criar uma simulação tridimensional, que dá a ilusão de profundidade. A vantagem desta técnica é a necessidade de poucos cálculos tornando a execução eficiente. A projeção isométrica também é conhecida como falso 3D.

Vale ressaltar que a projeção isométrica que será utilizada é uma representação axonométrica, onde mesmo à distância da câmera sendo alterada, a escala dos objetos permanece a mesma, não possuindo um ponto de fuga.

Diversos jogos contêm as características de um jogo isométrico, como, por exemplo, o Diablo I, lançado pela Blizzard que tem como representação isométrica 2:1 (Figura 40).



Figura 40: Exemplo de representação isométrica no jogo Diablo I.

Os mapas do jogo são baseados em *tiles*, que é uma pequena porção de uma imagem. Estes *tiles* são colocados um ao lado do outro para a construção do cenário. Dos diversos tipos de mapa existentes, nossa proposta utilizará o *Diamond* (Figura 41 – Peters,2009). Note que só é possível uma representação isométrica de um mundo porque é possível utilizar cor transparente no fundo do *tile*, caso não fosse possível, os *tiles* iriam se sobrepor desorganizando todo o padrão visual do mapa.

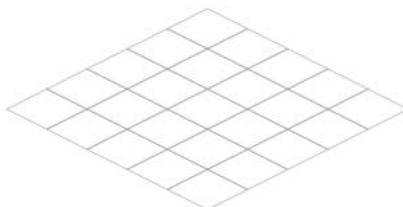


Figura 41: Representação do Mapa Diamond.

Nos mundos isométricos existem diversas técnicas para desenvolvimento da parte visual. Sem considerar modelagens 3D temos duas técnicas principais que são utilizadas por diversos jogos. A primeira técnica é a baseada na arte que tem como característica desenhar todas as partes do cenário e os objetos que o compõe. Geralmente não se aproveita a arte em um todo que já foi feita. Este estilo enriquece muito visualmente o jogo, porém existem duas desvantagens consideráveis que é a questão da implementação que fica restrita aos desenhos desenvolvidos e o tempo para desenvolvimento das artes.

A outra técnica é baseada em *tiles*. Na Figura 41, por exemplo, podemos ver um *tile*, que é representado pelo quadrado menor, sendo assim na figura existem 25 *tiles*. A arte é desenvolvida dentro de cada *tile* e assim pode ser reaproveitada por todo o mundo. Ex: Se

quisermos desenhar um muro, ele pode ser composto por três *tiles*. O primeiro é o canto direito, o segundo é o canto esquerdo e o terceiro é o meio. Sendo assim podemos construir um muro com três *tiles* ou com dez *tiles*, já que o *tile* do meio pode ser reaproveitado diversas vezes.

A técnica utilizada neste jogo será baseada pelas duas formas. A primeira forma (arte) será para os backgrounds que vão compor o mundo. A segunda técnica (*tile*) vai representar os objetos, chão, paredes entre outros itens do jogo.

3.8.10 Multiplayer

Visando a necessidade de criar um jogo multiplayer, será necessária a utilização de um sistema que consiga fazer a comunicação cliente-servidor para disseminar as mensagens devidas aos demais usuários.

O servidor tem uma visão total e constantemente atualizada de todo o mundo virtual que os jogadores poderão fazer acesso e tem como principal função repassar a todos os clientes como o ambiente esta (estado do ambiente). O mundo é persistente, objetos, posições, personagens, tudo é gravado.

A comunicação que será utilizada entre cliente e servidor será através de *sockets*, onde é possível abrir um canal e então trafegar informações constantemente para que seja possível interagir com o jogo proposto.

Outra característica que deve ser considerada é o possível suporte a *streaming* de vídeo e áudio, esta necessidade surge caso seja necessário um meio de comunicação por webcam e até mesmo microfone entre os jogadores no momento em que estão respondendo as perguntas.

Alguns conceitos são importantes para o entendimento do funcionamento do servidor e dos clientes, para que possa ser desenvolvido um jogo:

- **Cliente:** É um aplicativo controlado por uma pessoa, que serve para o usuário conectar ao servidor, receber e enviar mensagens para que as ações possam ocorrer. Esta aplicação oferece uma interface para o usuário interagir com as outras pessoas e com o mundo.
- **Servidor:** É a peça central, onde todos os clientes farão acesso. A partir do acesso, serão trocadas diversas mensagens para o andamento do jogo. O servidor tipicamente é desenvolvido em *socket* sobre o protocolo TCP ou UDP. Os clientes não se

comunicam entre eles, as mensagens são sempre enviadas para o servidor e então ele propaga as informações para onde elas devem ir.

- **Mensagem:** Elas são uma simples instrução, que apresentam um nome, argumentos, a fonte que enviou e para onde é destinada. Elas servem para enviar a ação que deve ser executada, como atualizar o posicionamento de personagens, ações no ambiente, mensagens a outros usuários.
- **Room:** A sala serve para criar um ambiente em que clientes compartilham o mesmo canal de comunicação. Estas salas ajudam bastante no desenvolvimento, geralmente é necessário criar grupos de usuários que devem ficar em ambientes separados, pois cada ambiente contém ações restritas a ele. Logo, cada seção pode fazer suas ações específicas sem que afetem os usuários que não fazem parte da seção. Um exemplo simples é um *Chat*: tenho duas salas, os usuários de uma sala não enxergam o que os usuários da outra sala escrevem.
- **Atributos:** Eles são as variáveis utilizadas pelas mensagens, são os dados que trafegam de um lado para o outro como, por exemplo, o posicionamento de personagens, ou o envio de uma mensagem com o texto “olá”.

Levando em conta os conceitos, a necessidade do que deve ser desenvolvido, foram avaliados os seguintes *middlewares*:

1. SmartFoxPro Server;
2. ElectroServer 4;
3. Gfs Server;
4. Palabre;
5. Photon;

3.8.11 Análise de *Middlewares* para o Servidor

SmartFoxPro Server⁷

É um servidor *socket* multi-plataforma desenvolvido para ser utilizado no Flash, porém ele ainda pode ser utilizado no .Net, Unity3D, iPhone entre outras. Ele foi escrito na linguagem Java, logo garante a portabilidade para outras plataformas.

⁷ <http://www.smartfoxserver.com>

Segundo as especificações do servidor, ele é baseado em uma alta performance e escalável, podendo rodar com mais de mil pessoas conectadas em apenas uma máquina. O SmartFoxPro tem sido escolhido por muitas empresas e programadores para desenvolvimento de jogos multiplayer.

Alguns cases de grande porte que utilizam este servidor é o Club Penguin da Walt Disney Company⁸. Este jogo apresenta mais de 2 milhões de registros em menos de 2 anos e é considerado a maior aplicação desenvolvida com o SmartFoxPro.

Outras aplicações como, por exemplo, YoVille⁹, são bem conceituadas. Desenvolvida pela Social MMO, o YoVille é uma aplicação para o Facebook¹⁰, onde teve mais de 150 mil acessos em um único dia.

A arquitetura do SmartFoxPro permite o desenvolvimento de games e aplicações multiusuários, ele permite a criação de extensões onde são programadas as lógicas do jogo, novos comportamentos do servidor, conexão com banco de dados, etc. As linguagens suportadas são Actionscript, Javascript, Java e Python.

Existe uma API rica em Flash, que ajuda no desenvolvimento mais rápido das aplicações, a documentação é bem rica, contendo diversos exemplos de aplicações, demonstrando o funcionamento do servidor.

A Figura 20 descreve basicamente como funciona o servidor. O cliente através da aplicação carregada no browser, inicia uma conexão com o servidor. Assim é criada uma conexão entre o cliente e o servidor onde trafegam as informações que são acessadas pelas extensões que guardam as regras de negócio e quando necessário é acessado um servidor de banco de dados, ou qualquer outro servidor necessário.

⁸ <http://www.clubpenguin.com>

⁹ <http://www.yoville.com>

¹⁰ <http://www.facebook.com>

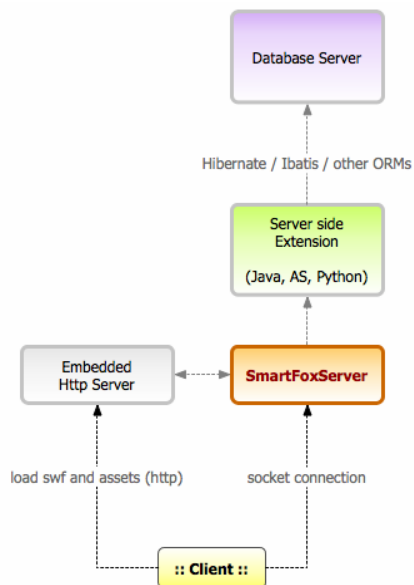


Figura 42: Funcionamento do SmartFoxPro.

O SmartFoxPro pode ser considerado uma aplicação autônoma, já que além do suporte a extensões ele ainda traz consigo um servidor web embutido, abrangendo ainda mais o seu poder.

Outra característica peculiar é a integração com o Red5¹¹ chamado de Red Box no SmartFoxPro, o Red Five é um servidor de streaming de áudio e vídeo. Sendo assim, o SmartFoxPro simplifica ainda mais o desenvolvimento de aplicações deste gênero.

Na Figura 43 podemos entender um pouco mais como é a arquitetura.

¹¹ <http://osflash.org/red5>

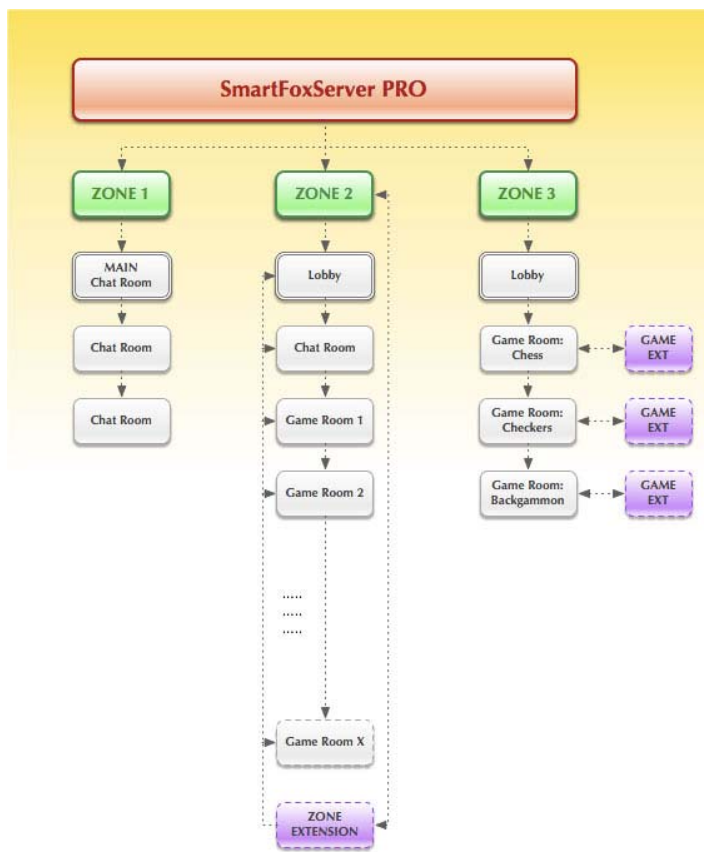


Figura 43: Arquitetura SmartFoxPro.

Quando iniciado o servidor, ele carrega diversas zonas, cada zona corresponde a uma aplicação, é possível no mesmo servidor rodar diversas aplicações e jogos simultaneamente.

Podemos perceber ainda que cada zona possui suas respectivas extensões (lógicas de jogo, lógica de acesso a base de dados e etc.), garantindo que as ações são vistas apenas pela zona permitida.

Cada zona pode ter diversas Rooms (conceito de salas), cada Room tem um grupo de usuários que acessam ela, o usuário pode participar de mais de uma sala ao mesmo tempo ou participar de apenas uma.

Esta tecnologia (Pro Edition) pode ser utilizada sem custo até 20 usuários logados simultaneamente, após isto os valores variam conforme o limite de usuários desejados.

ElectroServer 4¹²

¹² <http://www.electro-server.com>

O ElectroServer (ES4) é outro servidor *socket* multiplataforma voltado ao desenvolvimento de aplicações multi-usuário e jogos multiplayer na tecnologia Flash/Flex/Air. Sua arquitetura é amplamente escalável podendo suportar mais de 200 mil usuários simultâneos, onde é possível distribuir a carga entre diversos servidores.

É possível instalar o ES4 em diversas máquinas e criar uma conexão entre eles, facilitando a ampliação e a distribuição da informação em mais de um servidor.

Possui suporte a criação de extensões no lado do servidor tanto na linguagem Actionscript como Java, permitindo a implementação da regra de negócio do jogo, ou da aplicação.

Outra característica peculiar do ES4 é a possibilidade de fazer streaming de vídeo e áudio, agilizando assim o desenvolvimento de aplicações que necessitam desta tecnologia.

Existem diversos jogos desenvolvidos nesta tecnologia que está no mercado desde junho 2001, como Stick Cricket¹³ criado em 2004 que já teve mais de 2 milhões de usuários cadastrados nos primeiros anos. Existem outros jogos desenvolvidos pela Disney, Nickelodeon e Hotwheels.

As APIs desenvolvidas apresentam-se maduras e com uma boa documentação, segundo o site do próprio ES4, é possível encontrar mais de 1000 páginas de ajuda, tanto na parte do Actionscript quanto na parte de Java. Existem diversos exemplos desenvolvidos que podem ser usados para auxílio.

Esta tecnologia pode ser utilizada até 25 usuários simultâneos sem custo. A arquitetura do ES4 é semelhante ao do SmartFoxPro, ele contém zonas e rooms, possuem extensões e é possível rodar varias zonas simultaneamente. As funcionalidades das Rooms se assemelham também.

Gfs Server¹⁴

Desenvolvido em janeiro de 2008 e liberado sobre a licença GPL 2, com base no projeto Apache Mina, tem como objetivo criar um servidor *socket* voltado para a tecnologia Flash/Flex/Air.

É um projeto relativamente pequeno comparado ao SmartFoxPro e ao ElectroServer 5, sua API para Flash desenvolvida na linguagem AS3 suporta as funcionalidades mais básicas de conexão, *login* e troca de mensagens. Mesmo sendo apenas funcionalidades básicas, o servidor pode contemplar o jogo perfeitamente.

¹³ <http://www.stickcricket.com>

¹⁴ <http://code.google.com/p/gfs-server>

Outros pontos que deixam a desejar é a documentação pobre, mostrando exemplos muito básicos. A instalação e a configuração do servidor também não é simplificada. O servidor é construído totalmente na linguagem Java e toda a regra de negócio deve ser implementada em Java. O server tem um limite de 1000 usuários simultâneos devido a questões técnicas segundo o autor do projeto.

Palabre¹⁵

Palabre é um servidor *socket open source* desenvolvido na linguagem Python para utilizar na tecnologia Flash/Flex/Air, segundo o site do fabricante ele é limitado, porém possui as funções básicas para a ligação com o Flash, criado em meados de 2005.

Analisando sua documentação é possível constatar que ele trabalha com o conceito de Rooms, abordado na tecnologia SmartFoxPro, onde usuários acessam nenhuma, uma ou mais salas, nestas salas são criadas diferentes ações que replicarão apenas para os usuários que estão no ambiente. É possível ainda enviar mensagens para um grupo de usuários ou mensagens privadas. A API desenvolvida para a tecnologia Flash, para a aplicação cliente fazer a ligação e trocar informação está na linguagem Actionscript 2.

A regra de negócio que fica no servidor deve ser implementada na linguagem Python. A documentação e os exemplos se assemelham ao GFS Server, não sendo muito extensa. A instalação é simplificada, apenas necessitando extrair os arquivos e alterar algumas propriedades do arquivo de configuração do servidor.

Photon¹⁶

Desenvolvido na linguagem C/C++, o Photon é um servidor *socket* que tem como principal característica dar suporte à comunicação de dados via protocolo UDP, tornando assim a transferência de informações entre cliente-servidor mais rápida.

O servidor só pode ser rodado numa máquina Windows, não sendo multiplataforma. A inicialização dele é extremamente fácil e a instalação requer apenas a extração dos arquivos e a configuração é através de um arquivo XML.

Este servidor dá suporte a diversas plataformas para o cliente como Flash, Unity, iPhone, Android, .NET, J2ME, BREW. O funcionamento do Photon se assemelha ao padrão dos outros servidores já abordados, apresentam Rooms e troca de mensagens privadas e

¹⁵ <http://palabre.gavroche.net/>

¹⁶ <http://www.exitgames.com/product/photon>

publicas. Existem poucos exemplos sobre como trabalhar com ele e apesar de possuírem diversas APIs para diferentes linguagens, a documentação deixa a desejar, porém é possível que eles forneçam um material rico após a compra do produto.

A utilização deste servidor é trial por 30 dias e sua melhor licença custa \$3500¹⁷, com acesso ilimitado a usuários. Segundo o site do servidor, ele serve para milhares de clientes simultâneos, sem qualquer limitação.

Alguns exemplos de aplicações de grande porte que utilizaram o Photon é o World Golf Tour¹⁸, desenvolvido na tecnologia Flash. Em 06/2008 foi registrado mais de um milhão de registros no jogo. Diversas empresas já desenvolveram algum trabalho com o servidor Photon, como Konami, Taito, Glu, Fugumobili, Yahoo, Brasiltelecom.

3.8.12 Conclusão dos *Middlewares*

Todos os servidores oferecem suporte para que possa ser desenvolvido o jogo, porém alguns se destacam, podendo facilitar o desenvolvimento.

Tabela 6: Comparativo das Análises dos *Middlewares*

Servidores	Painel Administrativo	Multiplataforma	Servidor maduro, documentação, exemplos	Streaming de vídeo e áudio	Licença	Servidor Distribuído	Linguagem da API Flash dos servidores	Linguagem da regra de negócio
SmartFoxPro	1	1	3	1	20*	0	As1, As2, As3	Java, Javascript, AS, Python
ElectroServer	1	1	3	1	25*	1	As2, As3	Java, Javascript, AS, Python
GFS Server	0	1	1	0	Free	0	As3	Java
Palabre	0	1	1	0	Free	0	As2	Phyton
Photon	0	0	2	0	Trial-30 dias	0	As3	.NET

* Usuários simultâneos.

0 – Não possui

1 – Possui, porém é pouco/escasso

2 – Possui, é médio/básico

3 – Possui, é bom

Algumas características devem ser levadas em conta no momento da escolha:

- **Linguagem de programação para o cliente:** A linguagem que será programada o cliente é em Actionscript 3. Levando a tabela 6 em consideração, apenas um não atende esta necessidade que é o Palabre.

¹⁷ Acessado em 10/04/2009

¹⁸ <http://www.wgt.com/>

- **Multiplataforma:** O servidor deve fornecer suporte para rodar em sistemas operacionais Linux/Unix para que não seja necessário o custo da licença do servidor, sendo assim apenas o Photon não contempla este requisito.
- **Custo:** O custo deve ser baixo ou não deve haver nenhum custo de licença para utilização. Percebendo a tabela levantada, apenas dois conseguem fornecer o suporte sem custo nenhum para longo prazo que é o GFS e o Palabre, o SmartFoxPro e o ES4 atendem a demanda num primeiro momento, já que a ideia é que o servidor atenda inicialmente uma demanda máxima de 20 jogadores simultâneos e apenas um é totalmente pago, o Photon.
- **Documentação:** A necessidade de uma documentação bem redigida auxilia na hora do desenvolvimento, logo é importante que a tecnologia escolhida ofereça uma boa fonte de exemplos ou uma documentação clara para que possa ser desenvolvido o trabalho. Apenas duas possuem uma documentação razoável, o resto apresenta uma boa documentação, mesmo sendo razoável elas atendem as necessidades.
- **Streaming de Vídeo e Áudio:** Apenas duas fornecem o auxílio direto para criação de streaming de vídeo e áudio. Os outros middlewares precisam de um servidor só para atender o streaming de vídeo e áudio.

Levando estes aspectos, duas tecnologias foram previamente escolhidas para o desenvolvimento do trabalho. A ElectroServer e a SmartFoxPro, pois atendem melhor os requisitos necessários para a implementação do trabalho.

4 DESENVOLVIMENTO

Este capítulo tem como objetivo descrever de forma detalhada o desenvolvimento das ideias e implementações propostas para a criação do Mundus. O primeiro assunto trata sobre o painel administrativo, abordando as principais características, requisições e limitações presentes.

A segunda parte deste capítulo aborda sobre o servidor, especialmente o fluxo principal de seu funcionamento e o middleware escolhido. Esta parte é dividida em duas, o desafio e o mundo. A parte relacionada aos desafios desenvolvida em AS1 explica as requisições feitas entre cliente e servidor e seu funcionamento geral.

A segunda parte busca explicar o mundo, detalhando a implementação feita em Java, como as requisições feitas pelo *client*, carregamento do mundo, tipos de mensagens que trafegam, segurança e problemas enfrentados no seu desenvolvimento.

A última parte desta seção descreve sobre o cliente, seu fluxo de funcionamento, requisições, principais características e técnicas utilizadas para resolver a implementação. As telas do jogo são mostradas no anexo F.

4.1 Painel Administrativo

O painel é a principal ligação entre professores e jogadores. Seu funcionamento é explicado no capítulo 3 na seção de prototipagem (item 3.7.1). Visando uma explicação da implementação, serão pontuadas as principais características e as requisições utilizadas para comunicação com a camada de dados.

4.1.1 Usabilidade

As telas foram desenvolvidas com o mesmo padrão, abrindo o conteúdo necessário na mesma área e um menu lateral para acesso às seções. Cada seção corresponde a uma aba da área principal, permitindo abrir mais de uma seção ao mesmo tempo, facilitando consultas e cadastros.

Nas partes do sistema em que apresentam uma lista de informações, é possível ver

mais detalhes colocando o mouse exatamente sobre o item. Após um curto período de tempo uma caixa de dica (*tooltip*) irá abrir sobre o item com informações mais detalhadas.

Para facilitar o andamento dos processos dentro de cada seção foi destacado com números e textos qual é o passo em que ele está e os passos seguintes. Por exemplo, na tela de cadastro de um conteúdo relacionado, o primeiro passo é adicionar um conteúdo, o segundo passo é salvar.

4.1.2 Comunicação AMF

Como descrito na arquitetura, seção 3.8.3, a comunicação com a camada de dados é feita através do protocolo AMF. Foi desenvolvida uma classe chamada **Service** para facilitar a chamada das funções na camada de dados.

Nas propriedades do projeto é configurado um arquivo XML de configuração para a utilização do *remoting* (serviço AMF), apontando o endereço de ponte entre o Flex e o Php (comando de compilação: `-services service.xml`).

As chamadas de função são feitas através do método *setService*, passando por parâmetro o nome da classe, o método, a função de retorno dos dados e os argumentos que deseja ser enviado (separados por vírgula), conforme exemplificado no código abaixo:

```
Service.setService('JogadorService', 'logar', logarHandler, "meulogin", "minhasenha");
```

4.1.3 Requisições

Arquivo	Requisição (Classe, Método, Parâmetros)	Descrição
DesafioConteudo.mxml	DesafioService, salvar, DesafioVO	Salvar o desafio contendo suas características, conteúdos relacionados e perguntas/respostas.
DesafioParam.mxml	DesafioService, selectSessao	Recuperar a lista de seções (Ex. de sessões: Português, História, Matemática ...).
DesafioParam.mxml	DesafioService, selectSerie	Recuperar a lista de séries (Ex: terceira série, quarta

		série, primeiro semestre ...).
ListarDesafio.mxml	ListaDesafioRecuperar, selectAll	Recupera uma lista de todos os desafios cadastrados no sistema.
Login.mxml	JogadorService, logar, login, senha	Autentica o usuário.
Relatorio.mxml	RelatorioService, selectJogador, identificação do jogador	Recupera um XML com dados estatísticos do jogador solicitado.
RelatorioProfessor.mxml	RelatorioService, selectAll	Recupera uma lista de todos os jogadores cadastrados.
Responder.mxml	ResponderService, selectAll, id do moderador	Recupera uma lista de todas as respostas pendentes dos desafios cadastrados pelo moderador (usuário que está logado no sistema).
ResponderAluno.mxml	ResponderService, selectItem, identificação da pergunta, resposta	Recupera uma pergunta (do tipo dissertativa) com a resposta junto com uma lista dos jogadores que responderam esta questão (grupo).
ResponderAluno.mxml	ResponderService, salvar, objeto	Salvar a resposta do professor para os alunos que responderam aquela questão.

4.1.4 Relatórios

Os relatórios são gerados em uma página HTML através de uma classe desenvolvida em PHP. O motivo foi para facilitar o retorno de grande quantidade de informações, facilitar a impressão e agilizar o desenvolvimento e manutenção das informações retornadas aos professores e jogadoras.

Na emissão do relatório o painel mostra um resumo com dados estatísticos do andamento do jogador. Existe um botão específico para emitir o relatório completo de todos os desafios junto com as respostas do jogador (abre uma nova página web).

4.1.5 Respostas do Professor

No momento das respostas no cliente o jogador pode estar com um grupo de outros jogadores, ou jogando individualmente. Independente da forma, as respostas são salvas individuais para cada jogador. No momento em que o professor acessa a lista de perguntas pendentes, as perguntas são agrupadas por resposta. Quando o professor responder, ele irá responder para um grupo de alunos que contém a mesma resposta.

4.1.6 Segurança

A segurança é um ponto importante no desenvolvimento da aplicação. No momento do envio de dados para autenticação (*login*) a camada lógica criptografa a senha no formato MD5. Na camada de dados é validada a informação e então criada uma sessão de autenticação caso as informações sejam iguais as armazenadas no banco de dados.

No momento em que é chamada uma função da camada de dados (Ex: acesso a listas, cadastro de um desafio), é validado se o usuário está autenticado. Caso não esteja, é bloqueado o acesso e retornado um erro. Esta validação inibe parte dos problemas de segurança que poderia ocorrer.

Outra questão de segurança está na forma em que o Flash cria uma ponte de conexão entre a camada de dados através do protocolo AMF. Caso o painel esteja em um servidor diferente do servidor que contém a camada de dados, o servidor de dados deve liberar o endereço IP do servidor que apresenta o painel de gerenciamento para acessar.

Esta liberação é dada através do arquivo *crossdomain.xml* que apresenta as políticas de permissão de acesso para aquele servidor, este arquivo sempre fica na raiz do servidor web. Sendo assim apenas a aplicação rodando no servidor X poderá acessar os dados do servidor Y.

Outra questão de segurança que é muito comum é o SQL Injection, que explora as consultas SQL para acessar ou remover dados do banco de dados. Para acesso aos dados foi utilizada a biblioteca PDO no PHP para unificar conexões com a base de dados, sendo assim o sistema fica seguro contra este tipo de problema.

4.1.7 Skins

O visual da tela pode ser alterado a qualquer momento em aplicações desenvolvidas com a tecnologia Flex. Para conseguir aplicar um novo visual é programado um arquivo CSS (similar ao arquivo .css de páginas de Internet) que apresenta definições de como os componentes visuais da aplicação serão, podendo alterar cores, formatos, tamanhos e etc.

4.1.8 Limitações e Problemas Enfrentados

As seções que apresentam listas (*datagrid*) como a listas de respostas enviadas pelos jogadores, listas de alunos e listas dos desafios possui uma limitação de 5 mil registros.

Para solucionar este problema, as listas de registro deveriam possuir paginação, limitando o número de registros e otimizando o sistema. Esta solução não foi implementada por questão de demanda de usuários que utilizarão o sistema e pelo tempo para desenvolvimento.

Esta aplicação funciona apenas nas versões iguais ou superiores ao Flash Player 10 devido a utilização do SDK Flex 4.0. A escolha deste SDK foi pela questão de agilidade de desenvolvimento e otimização da ferramenta.

Não houve problemas significativos no desenvolvimento do painel.

4.2 Servidor

No capítulo 3 foi abordado alguns *middlewares* que pudessem suprir a necessidade de um servidor socket que conseguisse fornecer subsídios para trabalhar junto a tecnologia Flash, levando em consideração segurança, robustez, preço de licença e maturidade. No próximo item, dentro desta categoria, será detalhado qual foi escolhido.

A implementação da camada do servidor se deteve em duas partes, a primeira voltada aos desafios e a segunda voltada ao mundo virtual. Cada parte foi implementada independentemente, sendo assim é possível utilizar os desafios, sem precisar do mundo virtual. A independência acontece desde o início da implementação, onde o *SmartfoxServer* exige um cadastro de *extensions*, que são classes independentes que possuem a lógica da aplicação. A *extension* do desafio chama-se *Rpg*, e do mundo é chamada de *Server*. Quando informamos a *extension*, dizemos também que linguagem ela é programada. Estes dois itens

serão aprofundados nas seções 4.2.2 e 4.2.3.

Para conseguir entender melhor como funciona e até as características implementadas será detalhado o fluxo de funcionamento do servidor.

1. No início do servidor, é carregado toda a parte lógica do mundo e então libera o login aos usuários.
2. Inicia uma tarefa (*task*) que fica sempre rodando, ela tem o objetivo de persistir os dados do mundo. No caso do Mundus a principal função desta tarefa é salvar o posicionamento dos jogadores que estão logados no mundo.
3. O servidor fica aguardando os usuários entrarem no mundo.
4. Jogadores enviam solicitação de autenticação. O servidor verifica se os dados conferem, caso seja autenticado é verificado se possui um avatar criado ou não (no momento de registro da conta o usuário não cria o avatar, apenas os dados de acesso ao mundo). Caso não possua um avatar criado é enviada uma flag junto a resposta de “ok”, sinalizando a necessidade de criar um avatar.
5. O cliente recebe uma mensagem do servidor informando se ele foi autenticado ou não. Se o usuário foi autenticado e não necessita criar um avatar ele envia uma mensagem solicitando o acesso ao mundo virtual. Caso ele não possua um avatar, ele é direcionado para a parte de criação de avatar.
6. O servidor recebe a mensagem para o jogador entrar no mundo e então aloca para seus respectivos setores e retorna ao jogador a lista de setores que ele faz parte junto com a lista de jogadores que já estão naquela parte do mundo.
7. Neste momento o jogador estará escutando os eventos dos setores que ele faz parte, recebendo ações diversas (Ex: movimento, acesso ao setor, entre outros) de outros jogadores para tratar visualmente as informações.
8. Todas as interações do jogador são replicadas para os outros jogadores que pertencem ao mesmo setor (*room*). Como, por exemplo, o jogador andar pelo cenário, sair do jogo, andar para fora de um setor que outros jogadores estão contidos, entrar em um setor visível por outro jogador.
9. O jogador pode acessar a parte dos desafios nos NPCs. No momento da abertura de um desafio é enviada uma mensagem ao servidor solicitando uma determinada sessão do desafio e, então, os dados são retornados, abrindo os desafios para o jogador enfrentar. Os desafios apresentam um fluxo secundário de funcionamento:
 - a. Lista os desafios com suas características para o jogador escolher qual

- deseja enfrentar.
- b. Jogador escolhe o desafio e então é disparada uma requisição ao servidor para retornar a lista das salas existentes que o jogador pode acessar.
 - c. O jogador escolhe uma sala e então envia uma requisição informando qual a sala escolhida. Ele tem o poder de criar uma nova sala. Para criação, são enviados os dados principais como nome, quantidade de usuários e a senha de acesso a sala para o servidor, que irão criar a sala e retornar a informação ao jogador.
 - d. Neste momento do acesso ao desafio (criando ou entrando numa sala já existente) o jogador recebe as informações gerais do desafio.
 - e. No momento em que iniciar a partida (apenas o dono da sala pode começar a partida) os jogadores presentes na sala recebem do servidor as perguntas e as opções de resposta da questão.
 - f. Jogadores têm o poder de aprovar respostas e o dono da sala de enviar a resposta para o servidor. O servidor exige a aprovação de todos os jogadores da sala para aceitar resposta do moderador. Neste momento são trocadas várias informações com o servidor para aprovação até chegar ao envio da resposta.
 - g. O servidor recebe a resposta do grupo, processa a informação e envia a nova pergunta, caso já tenha terminado as perguntas, o servidor envia dados estatísticos e uma mensagem de fim do desafio para apresentar o feedback aos jogadores. Neste momento é gravada a experiência ganha pelos jogadores.
10. Sair do jogo e então enviar uma mensagem avisando todos os outros jogadores para removerem o jogador.

4.2.1 Escolha do middleware

Para atender todos os requisitos levantados na fase de análise foi escolhido dentre a lista de *middlewares* apresentados no capítulo 3 - Análise de Middlewares para o Servidor, o *SmartfoxServer*.

A escolha foi motivada pelo fato do *middleware* apresentar uma API que facilite a

programação do *client*, tendo suporte a linguagem AS3. Também por causa da programação da camada do servidor suportar Java e AS1. Outra influência foi o *middleware* ser multiplataforma.

4.2.2 Desafios

O desafio é representado pela *extension* RPG cadastrada no arquivo de configurações do servidor. A extensão do desafio é do tipo script, isto significa que foi programado na linguagem AS1.

A escolha por programar na linguagem AS1 foi motivada pelo fato de trazer agilidade no desenvolvimento, porém houve alguns problemas que serão apresentados no final desta seção.

Esta extensão é utilizada pelo jogador no momento em que ele interage com um NPC do mundo dentro do jogo, isto é, quando ele clica em um personagem do mundo. Nesta interação é aberta uma nova janela com informações iniciais sobre uma lista de desafios para uma determinada seção.

O acesso aos dados se concentra em um único arquivo PHP, que recebe o tipo e os parâmetros de entrada para fazer a consulta e persistência na base de dados, caso necessário. O arquivo é o *desafio.php*. Para o acesso aos dados, o servidor envia pelo método POST via protocolo HTTP as informações necessárias e então a camada de dados retorna a solicitação para o servidor. Note que todos os acessos aos dados são entre o servidor e a camada de dados.

O acesso a este arquivo poderia ter sido desenvolvido direto pela camada do cliente em alguns casos, sem passar então pelo servidor, porém, por questões de organização e principalmente por questões de segurança, todas requisições do cliente são feitas para o servidor e este então acessa os arquivos necessários para fazer a ação.

A fim de detalhar melhor o funcionamento da extensão que trata apenas da parte dos desafios, serão listados os nomes das requisições que o servidor recebe do cliente, sua descrição, a resposta que o servidor envia para o cliente e uma descrição.

Requisição: ListDesafio
Descrição: Recebe o <i>id</i> da sessão e série para acesso. Com os dados acessa a camada de dados que retorna uma consulta.

Resposta: ListDesafio
Descrição da resposta: Retorna um arquivo XML com os desafios presentes para o usuário que solicitou.

Requisição: InformacaoDesafio
Descrição: Consulta de um desafio. Lista informações do desafio para o usuário que solicitou.
Resposta: InformacaoDesafio
Descrição da resposta: Retorna um arquivo XML com as informações do desafio solicitado.

Requisição: ListSala
Descrição: Lista todas as salas criadas referente ao desafio selecionado pelo jogador. (Estas salas são criadas por outros jogadores para a formação de grupos).
Resposta: ListSala
Descrição da Resposta: Retorna um arquivo XML com a lista de salas abertas e fechadas para o jogador que solicitou.

Requisição: JoinRoom
Descrição: Requisição solicitada no momento em que o jogador vai entrar em uma sala de um determinado desafio criado por outro jogador. Ela tem como objetivo verificar se a sala já esta fechada ou não. Caso fechada ela não deixa o jogador entrar, caso esteja aberta o jogador pode entrar na sala e compor o grupo que irá responder o desafio.
Resposta: JoinRoom
Descrição da resposta: Retorna um XML para o jogador que solicitou se ele pode ou não entrar naquela sala.

Requisição: InitDesafio:
Descrição: Recebe o <i>id</i> do desafio para configurar os parâmetros da sala. Neste momento é criada uma nova room e então adicionado o XML do desafio, quem é o moderador e a identificação do desafio da sala. Estes parâmetros serão lidos no momento em que iniciar a partida.
Resposta: InitializedDesafio
Descrição da resposta: Envia o <i>id</i> da sala para o jogador quando o desafio foi inicializado.

Requisição: StartGame
Descrição: Recebe o <i>id</i> da sala e configura os parâmetros para iniciar o desafio.
Resposta: StartGame
Descrição da resposta: Envia o comando <i>StartGame</i> para uma lista de usuários que já estão presentes na sala para iniciar o jogo e visualizar as questões.

Requisição: NextPergunta
Descrição: Recebe a posição da pergunta do desafio que os jogadores estão enfrentando. Através da posição é verificado se vai para a próxima pergunta ou se finaliza a partida do desafio.
Resposta: NextPergunta
Descrição da resposta: Envia um XML com a próxima questão e os itens para resposta.

Requisição: SendRespostaPergunta
Descrição: Persiste a resposta dos jogadores e chama a requisição <i>NextPergunta</i> para ir para a próxima pergunta. Esta requisição só pode ser chamada pelo moderador da sala. Outro detalhe importante é que se os outros jogadores da sala não aprovarem a resposta do moderador, o moderador não pode enviar a resposta para o servidor.
Resposta: Chama a requisição <i>NextPergunta</i> caso todos os jogadores da sala tenham aprovado a pergunta, se não a resposta é <i>NaoAprovada</i> .
Descrição da resposta (Resposta <i>NaoAprovada</i>): Envia uma mensagem para a lista de usuários da sala informando que a resposta do moderador ainda não pode ser persistida porque algum usuário da sala não aprovou.

Requisição: ChangeResposta
Descrição: Esta requisição tem como objetivo “resetar” a aprovação de respostas da sala para forçar que todos os jogadores do desafio aprovem novamente a resposta. O outro objetivo é a atualização da resposta na tela dos jogadores.
Resposta: ChangeResposta
Descrição da resposta: Envia um XML com a nova resposta da questão para os jogadores presentes na sala, atualizando a resposta na tela. Exceto para o moderador, que foi o jogador que respondeu a questão. Esta requisição é chamada para perguntas de múltipla escolha e

única escolha.

Requisição: ChangeRespostaTexto

Descrição: Tem como objetivo “resetar” a aprovação da resposta da sala e forçar a atualização da resposta do moderador na tela dos outros jogadores. Esta requisição é chamada para perguntas dissertativas.

Resposta: ChangeRespostaTexto

Descrição da resposta: Envia um XML com a nova resposta da questão que o moderador escreveu para os jogadores que estão na sala.

Requisição: AprovarResposta

Descrição: Esta requisição é acessada apenas pelos jogadores que estão presentes na sala (com exceção do moderador). Ela tem como objetivo aprovar a resposta que o moderador deu para a resposta de uma pergunta do desafio.

Resposta: UpdateAprovado

Descrição da resposta: Envia para a lista de usuários um XML que contém a informação de quem já aprovou a resposta do moderador.

Requisição: FinishGame

Descrição: Recebe o *id* dos jogadores e a sala para persistir os dados de acerto e erro do desafio. Neste momento é persistido também a experiência ganha pelo jogador.

Resposta: FinishGame

Descrição da resposta: Envia para a lista de usuários o comando *FinishGame* sinalizando que o jogo terminou.

Requisição: GetFeedback

Descrição: Recupera um XML com as informações necessárias para dar um feedback ao jogador de uma determinada partida jogada.

Resposta: GetFeedback

Descrição da resposta: Retorna um XML para o jogador com informações de seu andamento na partida.

Requisição: GetConteudoRelacionado

Descrição: Recupera um XML com o conteúdo relacionado de um determinado desafio.
Resposta: GetConteudoRelacionado
Descrição da resposta: Envia um XML com o conteúdo relacionado de um determinado desafio para o jogador que solicitou.

Requisição: ChangeModerador
Descrição: Esta requisição é acionada no momento em que o moderador da sala sai por algum motivo, por exemplo, falha de conexão da Internet. Outro jogador presente na sala é titulado de novo moderador (caso não exista outro jogador, a partida é finalizada).
Resposta: ChangeModerador
Descrição da resposta: Envia para o usuário que será o novo moderador um comando informando que ele foi o escolhido.

Requisição: VerifyStatus
Descrição: Requisição utilizada para dar continuidade a um desafio incompleto. Esta requisição é verificada no momento do login do jogador.
Resposta: VerifyStatus
Descrição da resposta: Retorna para o jogador o desafio e a posição da pergunta em que o jogador parou.

4.2.3 Mundo

O mundo foi programado em Java, em função de fornecer um suporte mais confiável para solucionar problemas mais complexos. A classe principal é a *Server*, que recebe todas as requisições feitas pelos clientes e então propaga a informação para a sua devida área. Existem dois métodos principais que são o *handleRequest* e o *handleInternalRequest*.

O *handleInternalRequest* recebe eventos internos, do próprio servidor que gerenciam os usuários e o *server*. Nesta parte são tratados o login do usuário, o logout e o carregamento do mundo.

O *handleRequest* é responsável por receber as requisições dos clientes (os jogadores) e então executar a ação solicitada. Esta questão será detalhada na seção 4.2.3.4 Requisições de Interação. As classes *Config*, *Field* e *Command* servem para auxiliar o recebimento e envio de requisições.

4.2.3.1 Lógica do Mundo

O mundo é separado em pequenos blocos ou setores. Cada setor contém seus limites e seus vizinhos (outros setores). Um setor na camada do servidor corresponde a uma *Room* (explicado na sessão da arquitetura do sistema, item 3.8.10).

As *rooms* contêm uma lista de usuários. Cada usuário pode estar em mais de uma *room* ao mesmo tempo. No momento em que o usuário está em uma *room*, todos os outros usuários que estão nesta conseguem visualizar todas as ações deste usuário. Ex: Se o usuário A está na sala 1, o usuário B está na sala 1 e o usuário A anda, o usuário B vai ver o usuário A andando.

Cada usuário estará sempre em no máximo 9 *rooms* e no mínimo em 4, devido à lógica da aplicação. Através desta lógica é possível otimizar o processamento do servidor e o recebimento de informações dos clientes. As informações que irão trafegar para um grupo de usuários não serão processadas pelo grupo que está no outro lado do mundo.

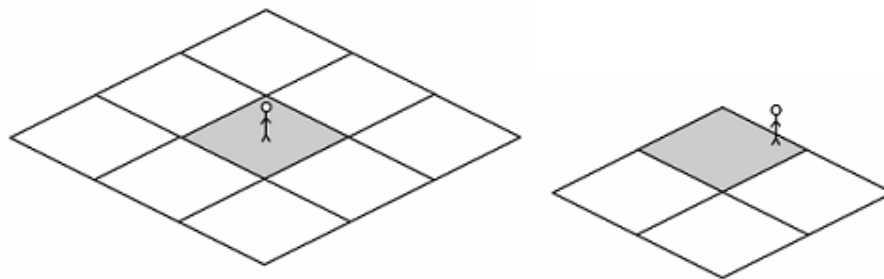


Figura 44 – 9 blocos carregados, 4 blocos carregados

4.2.3.2 Carregamento do Mundo

O carregamento do mundo representado pela classe *WorldLoader* é chamado quando é iniciado a aplicação *SmartfoxServer*. Nesta etapa é carregado um arquivo XML inicial com as definições e todos os setores necessários para carregamento do mundo.

Para o carregamento de arquivos foi desenvolvida a classe *Request* que implementa a classe *Runnable* e estende a classe *Observer*. No momento em que é solicitado o carregamento de um arquivo, é necessária uma *Thread* para não trancar o servidor. O motivo do *Observer* é conseguir tratar uma ação de retorno quando o carregamento estivesse concluído.

Após o carregamento dos setores, é criada a referência com os vizinhos. Esta associação serve para otimizar o processamento na hora do acesso a esse tipo de informação.

No final é carregado todos os NPCs do mundo e então posicionados em suas posições. O NPC é tratado como um jogador normal que faz login no servidor, vou explicar melhor detalhando em 3.0.3.5.

Depois de concluído o carregamento do mundo, o acesso é liberado aos jogadores.

4.2.3.3 Acesso ao Mundo

O acesso ao mundo é feito através do método *loginUser* dentro da classe *Server*. No *login* é verificado se o usuário já possui um avatar criado ou não. Caso ele não tenha criado um avatar ainda, uma *flag* é sinalizada no objeto de retorno para o cliente saber que o jogador deve ser direcionado para a criação do avatar.

Outra propriedade importante é o tempo corrente em milisegundos que é armazenado no jogador para ser utilizado na base de cálculos de animações e posicionamentos. Isto será explicado na seção 4.3.

4.2.3.4 Requisições de Interação

As implementações das requisições estão representadas pelo *package request*. As requisições implementam a interface *IRequestHandler* que tem como objetivo descentralizar todo código da classe *Server*, sendo que cada classe representa uma requisição.

1. **Walk (Classe WalkRequestHandler):** Esta requisição é utilizada para os movimentos dos jogadores feitos através do teclado. Ela recebe o *tile* que o jogador está e o *tile* que ele deve ir. Através dessas informações é verificado se o jogador permanece no setor (*room*) que ele está. Caso o setor mude, o jogador é realocado para os novos setores e retirados dos setores que ele não está mais presente. Após este processo a informação de movimento é replicada para todos os usuários que estão no mesmo setor (*room*) que ele. Para as *rooms* que o jogador saiu é enviada uma mensagem informando a saída do jogador da *room*.
2. **JoinWorld (Classe JoinWorldRequestHandler):** Logo após o login esta ação

é chamada se o jogador possuir um avatar criado. Ela tem como objetivo alocar o jogador nos seus respectivos setores (*rooms*) para começar a receber os eventos dos outros jogadores.

3. **RegisterAvatar (Classe RegisterAvatarRequestHandler):** Requisição utilizada no momento de guardar as informações do avatar criado pelo jogador no cliente. Ela recebe uma série de parâmetros e então direciona para a camada de dados onde é persistido na base de dados.
4. **SaveWorld (SaveTaskHandler):** Esta requisição é na verdade uma tarefa. Ela é executada de tanto em tanto tempo e tem como objetivo persistir as informações de posicionamento de todos os jogadores do mundo.
5. **SendMessage (Dentro da classe Server):** Responsável por receber mensagem de um usuário e repassar ela para todos os usuários da sala (chat do mundo virtual).

4.2.3.5 Player e NPC

É a representação do personagem no mundo. O player é um jogador humano e NPC é a representação do computador. Eles possuem as mesmas regras dentro do mundo, pertencem aos setores, podem se movimentar e possuem características.

A diferença entre eles é que o NPC possui uma tarefa (*task*) que de tanto em tanto tempo envia uma mensagem para o servidor. Esta mensagem serve para avisar o servidor que o jogador está respondendo para o servidor não desconectá-lo do mundo.

4.2.3.6 Mensagens

As mensagens que o servidor recebe e envia podem ser de três formas diferentes: ActionscriptObject, Json e Raw.

1. ActionscriptObject: A mais utilizada no decorrer do projeto. Todo o objeto é serializado em formato XML que contém as informações para trafegar de um lado para o outro. Para usar este formato basta criar um objeto no Flash, no momento do envio da mensagem do cliente para o servidor o objeto é convertido em XML. Exemplo:

```
<msg t='sys'><body action='login' r='0'><login z='Rpg'> <nick>
```

```
<![CDATA[admin]]></nick><pwd><![CDATA[21232f297a57a5a743894a0e4a801fcA]]></pwd></login></body></msg>
```

1. **Json:** Este formato utiliza o protocolo Json para converter os objetos e trafegar as informações. Este tipo de mensagem é melhor que o formato enviado em forma de XML. Para enviar no formato Json basta criar um Object no flash.

Exemplo:

```
{_cmd:login, nick:admin, pwd: 21232f297a57a5a743894a0e4a801fcA}
```

2. **Raw:** Protocolo que envia dados em pacotes pequenos. A maior diferença é que neste formado não são enviados objetos, mas apenas os valores finais (strings). O formato Raw que só passa strings é o formato mais rápido. Para passar as strings via flash é criado um *array* com valores. Exemplo:

```
%login%0%admin%21232f297a57a5a743894a0e4a801fcA
```

4.2.4 Segurança

Alguns pontos na parte de segurança seguem a mesma lógica de segurança explicada no painel do professor e aluno. O Flash possui um Security Box que protege os usuários de possíveis códigos maliciosos que estão fora do arquivo SWF.

Sendo assim, o Flash executa apenas o que está no mesmo domínio do servidor ou quando é configurado um arquivo de política de acesso no servidor de acesso aos dados (*crossdomain.xml*).

Porém, os pacotes não foram analisados a fim de saber como as mensagens são encapsuladas para envio do servidor, ou se as mensagens possuem algum tipo de criptografia.

4.2.5 Painel de Gerenciamento

O painel de gerenciamento serve para o monitorar o servidor, contendo informações de seu status, carga utilizada, largura da banda entre outros dados que ajudam na administração do servidor. Como mostra a Figura 45.



Figura 45 – Tela inicial do painel de gerenciamento do SmartFoxServer

É possível também ver todas as zonas, rooms e usuários que estão logados no servidor e todas as propriedades presentes nele. O painel ajudou muito no desenvolvimento do servidor e do cliente, pois muitas informações podiam ser confirmadas visualizando o painel (Figura 46).

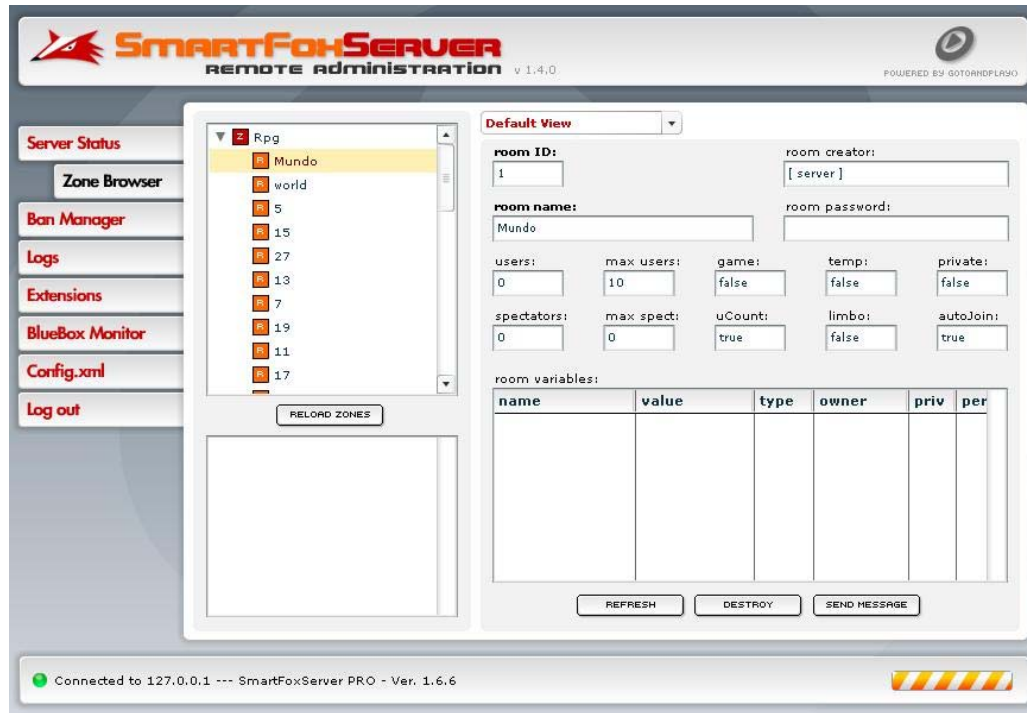


Figura 46 – Tela das zonas no painel

4.2.6 Limitações e Problemas Enfrentados

A principal limitação enfrentada foi o limite de 20 usuários no servidor por à versão gratuita. Assim não foi possível escalar o servidor para testar com muitos usuários e NPCs no jogo. (Como foi comentado na sessão Player e Npc, o NPC ocupa um slot como se fosse um usuário comum acessando o servidor).

Outra limitação foi o recarregamento do código inserido no servidor, com as mudanças do código era necessário reiniciar o servidor para carregar as novas alterações. Apesar do servidor se propor a recarregar o código digitado, ele funciona apenas para o nível principal (classe *Server*).

O maior problema enfrentado foi na parte dos desafios (programado em AS1). A manipulação de XML se apresentou falha, pois não foi possível utilizar o CDATA dentro dos nodos dos XML. Exemplo: <nodo><![CDATA[Valores]]></nodo>. Conseqüentemente, alguns arquivos XML do desafio não conseguiam ser convertidos e lançavam um erro no servidor comprometendo todo o andamento da aplicação. Um *workaround* foi limitar o uso de alguns caracteres no momento do cadastro do desafio. Porém, a solução ideal seria

implementar um conversor de XML para AS1 ou migrar todo código para a linguagem Java.

4.3 Cliente

Ao longo desta sessão será detalhado como tudo foi planejado e implementado. O fluxo geral do jogo compreende as seguintes etapas:

1. Acessar o cliente do jogo pelo navegador de Internet.
2. Criar um novo cadastro ou caso já possua, logar no jogo.
3. Criar avatar caso seja um novo cadastro.
4. Acessar o mundo isométrico com o seu avatar.
5. Andar pelo mundo, conversar com outros jogadores e acessar os desafios através dos NPCs.
6. No acesso ao desafio abrimos um sub fluxo:
 - a. Acessar uma listagem de desafios correspondentes ao NPC que foi interagido.
 - b. Ver as informações do desafio e escolher qual deseja enfrentar.
 - c. Criar uma nova sala ou entrar numa sala já criada.
 - d. Caso tenha criado a sala, aguardar novos jogadores entrar. Caso você não seja dono da sala, aguardar o moderador iniciar o jogo.
 - e. Responder os desafios interagindo como os outros usuários da mesma sala (seu grupo).
 - f. Ao final do desafio, um feedback é mostrado para o usuário saber como foi.
 - g. Voltar para o mundo.

Para que todas essas etapas serem consistentes uma série de definições e técnicas foram adotadas no momento da implementação do jogo. Abaixo serão explicadas as principais partes da implementação do cliente.

4.3.1.1 Estrutura

A estrutura foi implementada por um framework em AS3 desenvolvido por Gustavo Bremm e Rafael Buseti¹⁹ que tem como objetivo fornecer suporte ao carregamento de configurações e páginas (arquivos SWF), fluxo das sessões, transição de animação, validações, entre outras características.

De forma geral cada sessão é representada por um documento (Document), cada documento possui uma ou mais páginas (Page) e cada página possui seu conteúdo (Content). Os documentos e páginas são registrados na classe principal do sistema, pois é a partir do registro que é possível acessar o conteúdo das páginas.

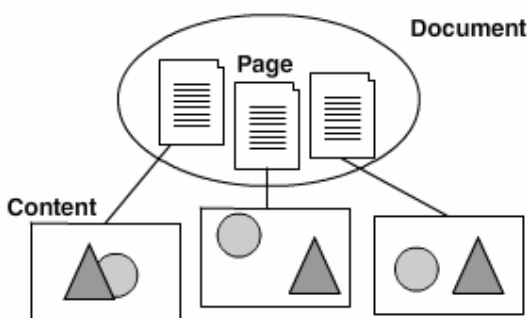
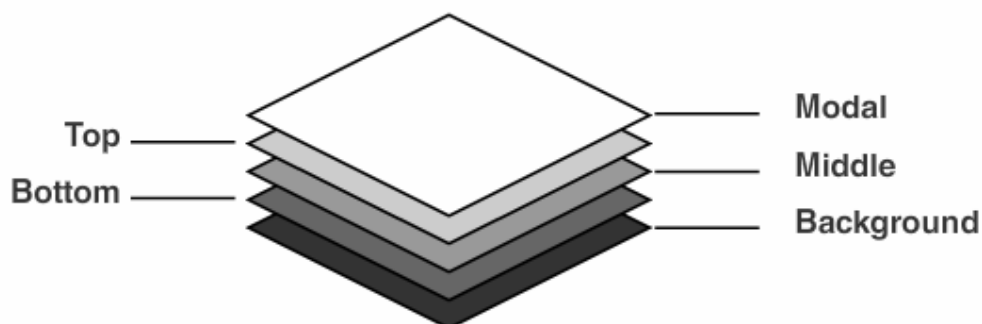


Figura 47 – Representação de um document, page e content

As páginas são registradas em camadas (*layers*), as layers podem ser Modal, Top, Middle, Bottom, Background fazendo com que elas mostrem os conteúdos em diferentes níveis (Figura 48). Toda a navegação no site é controlada por um fluxo, esses fluxos podem variar dependendo da aplicação.



¹⁹ [http:// www.auxteam.com.br](http://www.auxteam.com.br)

Figura 48 – Representação das camadas presentes no framework.

Para acessar os conteúdos, é aberto um documento ou apresentada uma página. No momento em que é solicitada a abertura é iniciada a transição de saída do conteúdo que está visível, depois carregado a nova sessão e, então, se inicia a transição de entrada do que foi carregado.

Outras classes foram utilizadas deste framework, as principais são validação de formulários (Validator), carregamento de arquivos em sequência (Queue), geração de dados (Dataset), scroll de conteúdo (SimpleScrollerV), grupo lógico (Check, CheckItem), e as classes gerais representado pelo package *utils*.

4.3.1.2 Registro do Avatar

Representado pela classe RegisterAvatarContent o jogador seleciona suas principais características como sexo, cabelo, roupas e escolhe um nome. Conforme ele vai configurando, um *preview* é mostrado ao jogador.

Vale ressaltar que o nome escolhido pelo jogador deve ser único no mundo, sendo assim é feita uma verificação, analisando se outro jogador já possui o nome informado pelo jogador.

Depois de concluído a customização do avatar o jogador envia as informações para o servidor armazenar os dados e então inserir o novo jogador no mundo.

4.3.1.3 Customização e Animação de Avatares

Existem técnicas para visualizar, animar e customizar os personagens (avatares), cada uma tem suas vantagens e desvantagens. Dentre as técnicas analisadas que são Puppet, Layered Animation, Sprite Sheets e 3D a escolhida foi Sprite Sheets.

Esta técnica consiste em uma imagem grande que contém pedaços de informações gráficas de um personagem, isto é, um *grid* de colunas que representam os frames e as linhas que representam o estado do personagem (Figura 49).



Figura 49 – Exemplo de um Sprite Sheet, retirado do livro ActionScript for Multiplayer Games and Virtual Worlds (Makar, 2009)

O interessante desta técnica é que quando se trata da customização, comparado a outras técnicas, esta possui maior rapidez para implementar a solução já que a lógica principal é sobrepor diversas imagens uma sobre a outra para ter um personagem customizado. Segundo Makar (2009) o Sprite Sheets é a melhor técnica para mundos virtuais construídos em Flash.

Makar comenta ainda que em questão de performance esta técnica possui um impacto muito baixo nos FPS (frames por segundo da aplicação) do jogo, pois no momento em que é carregado o Bitmap é feito *cache* (armazenado a imagem uma única vez na memória mesmo existindo várias instâncias), assim ele mostra rapidamente na tela quando necessário.

A grande desvantagem da utilização desta técnica é o alto consumo de memória, já que os *bitmaps* carregados são armazenados, logo fica presente na memória o tempo todo para um futuro acesso rápido.

O consumo de memória de qualquer bitmap carregado é dado pela seguinte operação:

$$\boxed{\text{Memória (em bytes)} = \text{largura} * \text{altura} * 4}$$

Sendo assim é muito importante analisar qual será o tamanho do *sprite sheet* e suas possíveis combinações para a aplicação não rodar lenta ou travar em alguns momentos. Esta

técnica esta representada pela classe *SpriteAnimation*.

Ela consiste em receber um ou mais *BitmapData* e gerar um *grid* com as *layers* sobrepostas, gerando o personagem customizado. Cada linha do *grid* possui então a ação como, por exemplo, o ângulo em que o Avatar está. Cada coluna do *grid* representa o frame de animação, se rodarmos uma linha em sequência poderá ver uma animação, por exemplo, um personagem andando. Esta classe ainda possui os métodos para chamar o frame anterior e o próximo, tornando possível a animação do *sprite*.

4.3.1.4 Chat

O chat pode ser visto em dois momentos no jogo.

- **Mundo:** No momento em que o jogador consegue acessar o mundo ele já pode interagir com outros jogadores. Através da classe *Console*, o jogador pode escrever e quando pressionar a tecla *Enter* é disparada uma mensagem para o servidor que irá receber a mensagem e replicar para os jogadores que estão no mesmo setor (*room*) que ele. Mostrando, assim, a mensagem.
- **Resposta dos desafios:** Quando os jogadores estiverem respondendo um desafio, no lado direito possui um chat. Ele funciona exatamente igual ao chat do mundo, todas as mensagens são enviadas para o servidor que então replica a mensagem para os jogadores da mesma sala. A principal diferença deste chat é que é possível alterar a cor da mensagem enviada para os outros. Este chat é representado pela classe *Chat*.

4.3.1.5 Carregamento do Mundo

O carregamento do mundo consiste em carregar as definições dos itens visuais, como *tiles*, objetos e carregar os setores que compõem o mundo e propriedades em geral. A linha lógica será apresentada a seguir, juntamente com descrições para facilitar o entendimento. Esta sessão irá explicar em linhas gerais as principais propriedades dos arquivos XML presentes no cliente.

O fluxo é representado basicamente pela classe *WorldLoader*.

1. Quando a aplicação cliente é rodada a classe *WorldLoader* é instanciada e então começa-se o carregamento do mundo através do método *load*.

2. Quando o método *load* é chamado, ele empilha o carregamento de cinco arquivos principais que são:
- Skins (skins.swf)**: Compreende um arquivo SWF que contém os objetos visuais do mundo como uma cadeira, espelho, grama, muros e etc. Cada imagem dentro deste arquivo está linkado a uma classe.
 - Background (background.swf)**: Contém os backgrounds do jogo. Cada background está linkado a uma classe.
 - Avatar (avatar.swf)**: Contém os *sprite sheets* dos avatares femininos e masculinos. Cada *sprite sheet* está linkado com uma classe.
 - Mapa (map.xml)**: Contém os parâmetros de configuração do mundo. Este XML possui as definições dos objetos representada pelo nodo `<itens>` que contém as propriedades de identificação, classe utilizada nos arquivos SWF, quantas linhas e colunas ocupam o objeto e se é permitido andar sobre ele. Exemplo:

```
<item>
  <id>5</id>
  <className>skins.Muro </className>
  <walkable>>false</walkable>
  <rows>6</rows>
  <cols>1</cols>
</item>
```

O XML do mapa ainda possui as definições dos backgrounds representada pelo nodo `<backgrounds>` e que contém uma identificação e classe utilizada nos SWF para fazer a ligação do id ao objeto visual no mundo. Exemplo:

```
<background>
  <id>0</id>
  <className>skins.Background</className>
</background>
```

O próximo nodo importante do XML é o `<world>` que contém o tamanho de cada setor e cada *tile*, este tamanho serve para calcular o posicionamento dos objetos no mundo e contém uma lista de setores (**sectorXXX.xml**) (representado pelo nodo `<sector>`), que representa o caminho físico dos XML de configuração de

cada setor do mundo.

- e. **Definição dos avatares:** Compreende um XML de configuração para fazer a ligação do visual com a parte lógica do mundo (mesma lógica do background e objetos do mundo). Neste XML são listadas as principais propriedades visuais de um avatar junto com uma lista de todas as classes visuais possíveis que um avatar pode usar.
3. Após o carregamento desses arquivos de configuração é carregada a lista de setores <sectors> presentes no XML do mapa (Anexo E). Cada setor possui um XML que contém suas características. As características são
 - Identificação.
 - Backgrounds.
 - Uma lista de tiles que são utilizados apenas para tratar colisões ou disparar ações.
 - Uma lista de objetos que podem representar objetos visuais do mundo, informar objetos de colisão e disparar ações como os tiles.
 4. Após o carregamento de todos os arquivos XML e SWF é iniciado a transição para a parte lógica do sistema, onde são criados diversos objetos com diversas propriedades presentes no XML para facilitar a captura das informações que são utilizadas durante o jogo. Neste momento é instanciado a classe WorldVO que contém toda a parte lógica do mundo.
 5. É disparado um evento para indicar que finalizou o carregamento inicial do mundo. Inicial porque durante o jogo, o cliente recebe diversas mensagens que adicionam e removem avatares do mundo que constitui também o carregamento constante do mundo.

4.3.1.6 Desafios

Os desafios são representados pelas classes que estão dentro dos packages *assets.desafio*, *assets.desafioescolha*, *assets.sala* e pelas classes *DesafioEscolhaContent*, *DesafioContent* e *SalaContent*.

Quando acessado um desafio num NPC a primeira classe chamada é a *DesafioEscolhaContent* (junto as classes do package *assets.desafioescolha*) que tem como

objetivo mostrar uma lista de desafios de uma determinada seção para o jogador escolher.

Logo após a escolha a classe **SalaContent** (junto as classes do package *assets.sala*) é acionada tendo como objetivo mostrar uma lista de salas criadas para aquele desafio e fornecer a ação de criar uma nova sala. Após entrar numa sala ou criar uma nova sala a classe **DesafioContent** (junto as classes do package *assets.desafio*) é chamada.

A classe **DesafioContent** é a classe principal dos desafios, é nela que existe todo o gerenciamento de perguntas e respostas, controle do início e fim da partida (*feedback*), chat e através dela é possível o acesso aos conteúdos relacionados de um determinado desafio.

4.3.1.7 Actions

Visando a necessidade de criar diferentes ações em diferentes locais do mundo foi implementado um sistema que dispara ações pré-registradas nos arquivos XML de cada setor. Essas ações podem ser do tipo:

- **TILE_CLICKED**: Momento em que o jogador clica com o mouse em um *tile*.
- **TILE_STEP**: Momento em que o avatar passar por cima de algum *tile*. Ele é disparado na entrada do jogador no *tile*.

No momento do registro é informado o tipo, qual a classe que representa a ação (esta classe implementa a interface *IAction*) e dentro do nodo *params* possui todos os parâmetros de entrada para execução.

Sendo assim é possível criar diferentes ações para diversos tipos de itens do jogo. Em um momento posso criar um item que abre uma imagem quando ele é clicado. Em outro item é possível abrir um vídeo quando o avatar passa por cima de um *tile* específico. Neste contexto, seria possível montar um sistema de batalhas individuais voltadas à educação. Sendo assim fica fácil criar novas ações que possam auxiliar no desenvolvimento do conhecimento dos alunos conseguindo apresentar o conteúdo de diferentes formas.

Exemplo de um item cadastrado com um evento, em destaque a parte que define a Action:

```
<item>
  <type>10</type>
  <col>4</col>
  <row>6</row>
  <events>
    <event>
      <type>TILE_CLICKED</type>
```

```

        <action>game.action.TesteAction</action>
        <params>
            <teste>Teste parâmetro 1 de entrada</teste>
            <teste2>Teste parâmetro 2 de entrada</teste2>
        </params>
    </event>
</events>
</item>

```

4.3.1.8 Mini Mapa

O mini-mapa é uma representação do mundo em miniatura. Apesar da representação do mundo virtual ser isométrica, a representação do mini-mapa é plana, com marcações representando objetos do mundo e um ponto central caracterizando o personagem que está logado.

A base de cálculo para tornar o posicionamento correto no mini-mapa (ponto centralizado) foi o seguinte:

X = Posição Linha do Avatar * (Tamanho do Setor no MiniMapa * Quantidade de Tiles em um Setor)

Y = Posição Coluna do Avatar * (Tamanho do Setor no MiniMapa * Quantidade de Tiles em um Setor)

Note que o tamanho do setor no mini-mapa significa um quadrado de, por exemplo, 100x100 que representa um setor do mundo isométrico. Para atualizar constantemente a posição do jogador foi adicionado um evento de *EnterFrame* e aplicado um *smoothing*²⁰ para o personagem não se mover de bloco em bloco no mini-mapa.

4.3.1.9 Mundo Isométrico

Toda a parte isométrica do mundo é calculada através da classe *Isometric* que contém os cálculos para converter um ponto de coordenadas (x, y) em um ponto isométrico (x,y,z) e um ponto isométrico em posição x, y de um plano.

O grid pode ser visto na Figura 50. Observe que o ponto inicial (0,0) é centralizado e está no topo do grid. Existem diversas bases de cálculos isométricos que posicionam o ponto

²⁰ Suaviza as bordas de um bitmap, tirando o serrilhado e embassando a imagem

inicial de diferentes maneiras.

Para o bom funcionamento do mundo isométrico ainda existem outras variáveis como o *sorting* de objetos e avatares, colisão, movimento de avatares, *scrolling*. Todas estas variáveis serão detalhadas nas próximas seções.

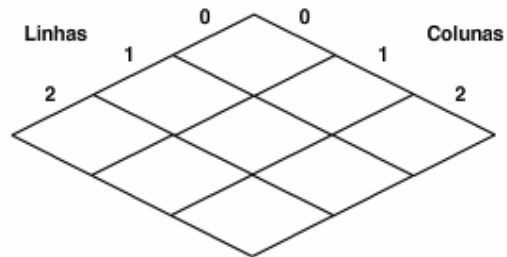


Figura 50 – Representação do grid com suas respectivas posições

4.3.1.9.1 Sorting

O *sorting* é feito dentro da classe World e é necessária para posicionar os objetos na frente ou atrás de outros objetos. Para ficar mais fácil o entendimento veja a Figura 51.

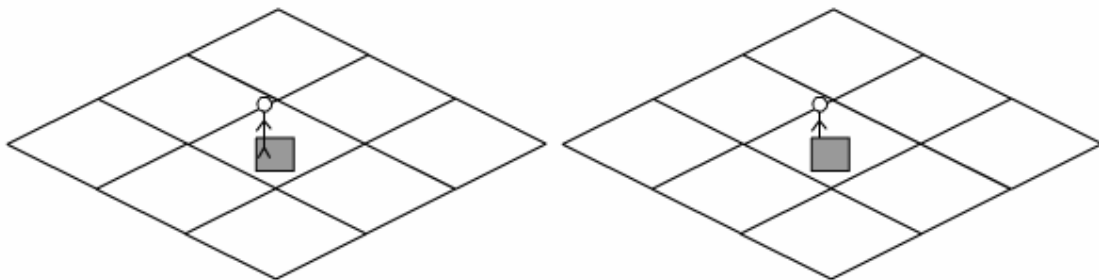


Figura 51 – Exemplo representando uma imagem com um grid sem *sorting* e outro grid com *sorting*.

Na primeira imagem não foi aplicado um algoritmo de *sorting*, na segunda foi. Veja que na primeira o boneco fica sobreposto a um objeto que deveria estar na frente dele.

O algoritmo funciona da seguinte forma:

1. Primeiro é preciso de uma lista de todos os objetos que se deseja fazer o *sorting*.

2. Cria-se uma lista vazia que irá possuir a lista final do *sorting*, denominada, a partir deste momento, de lista final.
3. Agora são feitas as comparações de todos os itens desta nova lista com a lista final (que inicialmente não tem nenhum valor).
4. Se um objeto da nova lista tiver uma coluna menor ou igual e uma linha menor ou igual ao objeto comparado da lista final, é adicionado antes dos outros objetos.
5. Se o teste não encontrar nenhum resultado é adicionado no final da lista.
6. Quando acabar a ordenação dos objetos é criado um novo laço que então adiciona novamente os objetos visuais no flash para conseguir fazer o *sorting* corretamente.

4.3.1.9.2 Colisão

Como o movimento dos personagens é baseado por *tile* o teste de colisão é caracterizado pela previsão dos movimentos do personagem. Quando um avatar se movimenta para algum lado é testado o *tile* para onde ele irá e então é verificado se é permitido caminhar ou não, caso não seja ele não lança a ação de movimento para o servidor, fazendo com que o personagem fique parado. É como se fosse um grid mapeado em quais quadrados podem e não podem andar.

4.3.1.9.3 Movimento

O modelo de movimento adotado no trabalho é através do teclado. Seu funcionamento lógico será explicado, mas inicialmente vale ressaltar um fator crucial na movimentação do personagem que é o tempo em que foi solicitado o movimento. Quando os jogadores recebem a ação de movimento de um jogador (através do servidor) é montada uma lista de *tiles* que o jogador irá percorrer e então através do tempo inicial é calculado o tempo de todos os passos que o jogador deseja fazer.

Este tempo é utilizado para conseguir mostrar o jogador andando na tela igualmente para todos que estão naquele setor visualizando o jogador, fazendo com que mesmo se um dos jogadores tiver *lag* (atraso por causa dos pacotes trafegados entre o jogador e o servidor), não comprometa a real posição na tela.

A lógica funciona da seguinte forma

1. Jogador clica em alguma seta direcional do teclado e então verificado se o *tile* que ele deseja ir é permitido andar ou não.
2. Caso não seja permitido ele fica parado, caso seja permitido é disparado uma mensagem para o servidor com o *tile* atual, o próximo *tile* e o tempo atual do jogador.
3. O servidor recebe esta informação e replica para todos os jogadores da sala, junto com o jogador que fez o movimento.
4. O cliente pega estas informações monta um caminho que o jogador irá percorrer, calculando ainda o tempo para cada movimento através do tempo inicial passado pelo servidor. Este cálculo de tempo é feito através da distância de um *tile* ao outro.
5. Este caminho então é fixado no jogador que fez a ação e então a cada atualização de tela o jogador começará a se mover para todos os jogadores que estão na mesma sala que este jogador. Isso é possível devido existir um caminho nele e o tempo do caminho não ter atingido o final.
6. No momento em que o índice de movimento bater com o tamanho do caminho para percorrer o jogador para.

Apesar desta ação ter sido usada através do teclado, os métodos de movimento foram preparados para aceitar caminhos longos (utilização do Mouse para movimento), isto é, uma lista de caminhos. Para a implementação seria necessário programar um algoritmo de *Pathing* (Makar, 2009, pg. 157) em que seria retornado uma lista de movimentos válidos através do clique em um *tile* do mundo.

4.3.1.10 Requisições do Cliente

Todas as requisições são feitas através classe *SmartFoxClient* da API fornecida pelo próprio *SmartfoxServer*. Para centralizar as chamadas foi criado um *singleton* desta classe (representada pela classe *Client*) facilitando as chamadas e tratamentos de retorno. As requisições do cliente são todas as requisições apresentadas anteriormente na sessão Mundo.

4.3.1.11 Limitações e Problemas Enfrentados

Na implementação do cliente foi necessário tomar cuidados com a questão de memória e velocidade da aplicação. A primeira limitação foi na imagem de background do mundo, onde o limite imposto pelo flash é próximo de 4000x4000 (16000000 pixels, um pouco mais) por imagem.

Em questão de performance, os setores não podiam ser muito grandes, já que quanto maior, mais itens teriam no mundo para o jogador renderizar, deixando, assim, o jogo lento. Mas a arquitetura do sistema preveu este possível problema e então o mundo foi separado em blocos, carregando e mostrando apenas o necessário.

Nos movimentos do jogador eram testados se ele tinha mudado de setor, quando ele mudava de setor o cliente removia os setores antigos e adicionava os novos. Neste momento a aplicação dava uma leve travada mesmo se os setores não eram muito grandes (muitos tiles).

A solução para isto foi tornar o carregamento e a remoção de forma assíncrona, utilizando um tempo de *delay* para carregar e descarregar os setores da tela e aumentar os FPS (frames por segundo) da aplicação para 60.

O carregamento de imagens para evitar o grande consumo de memória também foi testado, em testes gerais era mais custoso carregar uma imagem por uma e alocar numa lista do que deixar todas as imagens dentro de um SWF e carregá-lo no início da aplicação. Isto porque no primeiro modo ele alocava o bitmap inteiro na memória, no segundo método ele só alocava o bitmap inteiro quando fosse realmente utilizado (adicionava a imagem no palco da aplicação), logo a quantidade de memória inicial foi bem menor que do primeiro modo.

Um problema enfrentando foi dar vários cliques no mesmo botão disparando várias requisições iguais ao servidor. A solução foi bloquear o botão por alguns segundos.

5 CONFIGURANDO O AMBIENTE

O processo de configuração do ambiente para executar os aplicativos não é uma tarefa fácil. Serão necessários os seguintes programas:

- Servidor Web em PHP (Versão do Php ≥ 5)
- Banco de dados MySQL
- SmartFoxServer (O jogo foi desenvolvido na versão 1.6.6)
- Flash (Para editar os fontes do Cliente e compilar)
- Flex (Para editar os fontes do Painel e compilar)
- Processador de Texto (Para editar o código em Java – ele é compilado no momento em que rodamos o servidor SmarFoxServer).
- O sistema operacional pode ser Unix/Linux ou Windows.

Será necessário o arquivo compactado do projeto que contém todos os fontes e arquivos de configuração para iniciar o processo de configuração do ambiente.

Inicialmente será detalhada a forma de configuração do servidor web, o próximo passo será a configuração do painel de gerenciamento e o banco de dados do projeto. Com esta etapa configurada, será feita a configuração do cliente e logo em seguida a do servidor.

5.1 Configurando um Servidor Web com Banco de Dados Default

Existem ferramentas que facilitam a instalação e já vem pré-configurados para utilização. Exemplos desta aplicação são o EasyPHP e Wamp Server. Com eles vem configurado o servidor web Apache, banco de dados MySQL, phpMyAdmin para gerenciamento das informações no banco de dados entre outros aplicativos.

5.2 Painel e Banco de Dados

Arquivo: painel.rar

O painel foi desenvolvido em FLEX. Assim, qualquer modificação de código fonte deve ser feita através da importação do projeto neste ambiente. Os arquivos principais

(compilados) ficam na pasta *bin-debug* (são estes que devem ser publicados em seu servidor para acesso do painel). Para edição, depois de importado o projeto, é necessário configurar a compilação com o seguinte parâmetro `-services services.xml`.

O `services.xml` deve ser editado no nodo `<endpoint>` (Ex: `<endpoint uri=http://localhost/mundus/flex/amfphp/gateway.php ...>`). É preciso colocar o endereço de onde apresenta o arquivo de ponte do AMFPHP.

Para configurar o AMFPHP, a pasta *amfphp* no *.rar* deve estar dentro de um servidor Web PHP. A pasta pode ser colocada dentro do painel, desde que o local esteja num servidor web.

Após, é necessário abrir o arquivo *globals.php* na raiz da pasta *amfphp* e no final dele configurar o acesso ao banco de dados. Após feito isto, o arquivo *painel.sql* deve ser aberto e importado em seu banco de dados MySQL. Existe um arquivo no projeto do flex chamado *config.xml*, nele são configurados os endereços do relatório, que estão dentro da pasta *amfphp/relatorio/*.php*. Com isto feito, o painel já estará configurado para utilização.

Para testar se a conexão do banco foi efetuada com sucesso e se o flex irá conseguir acessar as informações, é preciso acessar o seguinte endereço `http://SEUENDereco/painel/amfphp/browser` ou `http://SEUENDereco/mundus/amfphp/browser` (tudo depende de como foi configurado) que abrirá um painel de gerenciamento onde se pode executar os métodos do PHP para verificar os retornos.

É importante configurar bem a pasta *amfphp* pois dentro dela estão os métodos principais que o servidor do jogo chama para persistir e consultar os dados.

5.3 Cliente

Arquivo: cliente.rar

O cliente é mais simples de configurar. Descompacte a pasta com os arquivos cliente em algum local que já seja o servidor Web Php, pois vai facilitar o acesso local e os testes. Existem dois arquivos que precisam ser configurados. O *config.xml* é o primeiro, o nodo `<serverIp>` deve apontar para o endereço que esta instalado o servidor SmartfoxServer e os nodos `<feedback>` e `<register>` para o endereço onde temos o *amfphp* configurado.

Para compilar o projeto foi utilizado o *FlashDevelop*, uma excelente ferramenta para editar códigos em Flash e Flex. No arquivo compactado do cliente já existe o arquivo para

abrir o projeto nesta ferramenta. Ela pode ser encontrada no endereço <http://www.flashdevelop.org>.

A pasta compilada do projeto é a *cliente/bin/*, que pode configurar os arquivos de XML e publicá-la diretamente. Não é necessário compilar o projeto para que funcione.

5.4 Servidor

Arquivo: servidor.rar

Primeiro instale o servidor *SmartFoxServer*²¹. Após instalado o servidor, configure o arquivo *pasta_do_servidor/Server/config.xml* os nodos `<AdminLogin>` e `<AdminPassword>` para liberar o acesso ao administrador do server.

No final do arquivo e dentro do nodo `<zones>` é preciso adicionar o seguinte código (exemplo do *config.xml* na pasta *server*)

```
<Zone name="Rpg" uCountUpdate="false" maxUsers="20" customLogin="true">
  <AutoReloadExtensions>true</AutoReloadExtensions>
  <Rooms>
    <Room name="Mundo" maxUsers="10" isPrivate="false" isTemp="false" autoJoin="true">
    </Room>
  </Rooms>
  <Extensions>
    <extension name="rpg" className="rpg.as" type="script" />
    <extension name="world" className="world.as" type="script" />
    <extension name="server" className="br.mundus.Server" type="java"/>
  </Extensions>
  <Moderators status="on">
    <Mod name="NOMEGERAL" pwd="SENHA" />
  </Moderators>
</Zone>
```

Agora é necessário copiar o arquivo *world.as* para a pasta *pasta_smarfoxserver \Server\sfsExtensions*. Após, deve-se copiar o pacote do Java (pasta com nome *Br*) para a pasta *pasta_smarfoxserver \Server\javaExtensions*.

Após configurado, o servidor deve ser executado e então o recurso de *admin* deve ser aberto. O *admin* fica localizado em *pasta_smarfoxserver/admin/AdminTool.swf*. Com ele aberto, deve-se acessar a aba *Zone Browser* e verificar se a zona RPG foi carregada. Dentro

²¹ <http://www.smartfoxserver.com>

desta zona deve ter as Rooms Mundo, World e números de 1 até 35. Neste cenário, o servidor estará sendo executado perfeitamente.

Para editar o código, é preciso editar os arquivos diretamente da pasta *pasta_smarfoxserver\Server\javaExtensions\br* e reiniciar o servidor. Neste trabalho foi utilizado o Eclipse como ambiente de programação e o projeto foi criado dentro desta pasta. No pacote de arquivos do servidor, está o projeto para importação no Eclipse. Vale ressaltar que no momento da importação os arquivos não podem ser movidos do local do servidor.

No arquivo *pasta_smartfoxserver/server/javaExtensions/ Br/mundus/Server.java*, as variáveis *urlServer* e *urlLogin* devem ser atualizadas para apontarem para os novos endereços. Os arquivos estão na pasta *amfphp*, cuja configuração foi detalhada no início desta sessão.

Por fim, o servidor deve ser reiniciado novamente.

6 CONCLUSÃO

O desenvolvimento de um MMOG que possibilite sua utilização como ferramenta educativa tanto para os mestres quanto para os aprendizes é possível, mas para isto é necessário criar um ambiente propício à educação.

Para a sua construção, precisamos unir todos os requisitos básicos no desenvolvimento de um ambiente de aprendizagem com as características de um jogo. No momento em que conseguimos englobar as características e que elas não se anulem, conseguiremos desenvolver um jogo educativo que instigue a aprendizagem e ao mesmo tempo seja divertido.

No momento que fornecemos um conteúdo estruturado, de forma clara e organizada, estamos facilitando os seus caminhos em busca do conhecimento. Toda a base do conhecimento do jogo é formulada e estruturada por pessoas de nível qualificado, que através de um painel administrativo podem criar os desafios do jogo.

Quando o aprendiz estiver interagindo com o jogo, ele estará formulando suas hipóteses com base em seu conhecimento ou no que o ambiente está fornecendo com feedbacks, conteúdos ou interações com outros jogadores. Neste cenário de equilíbrio e desequilíbrio remete a teoria descrita por Piaget e fornece a chave principal para a aprendizagem do jogador.

O feedback está presente no final de cada desafio enfrentado. Os conteúdos relacionados ao desafio são cadastrados pelos mestres e podem ser acessados pelo jogador no momento da resolução dos problemas.

Quando o jogador está enfrentando os desafios, ele próprio decide sua individualização perante os outros jogadores. O ambiente é *multiplayer*, permitindo a interação direta com outros jogadores dentro do mundo virtual e nos momentos em que ele está solucionando os desafios através do *chat*.

Um ambiente de aprendizagem modelado como um jogo, deve possuir um roteiro bem elaborado, um visual atrativo e personagens bem formulados. Sendo assim temos um jogo educativo que a diversão não é anulada, mas sim mais estimulada.

Se o jogador se sentir motivado neste ambiente lúdico, ele vai desenvolver continuamente suas ações que levarão para o desenvolvimento de seu conhecimento. O jogo ainda pode trazer o desenvolvimento de habilidades como concentração, resolução de problemas, colaboração com os outros jogadores e tomada de decisões.

Contudo, a união das características de um ambiente de aprendizagem e as características dos jogos por si só não são suficientes. O desenvolvimento de um MMOG exige ainda toda uma estrutura para seu desenvolvimento.

Através do levantamento de requisitos do jogo, conseguimos subsídios para começar a traçar uma arquitetura do sistema, definição de tecnologias que deverão ser utilizadas, construção de protótipos e diagramas para implementação e testes do sistema.

Como o jogo é *multiplayer*, foi necessário abordar e testar diferentes *middlewares* que fornecessem suporte a construção de um jogo *multiplayer* na tecnologia Flash para ser utilizado na implementação.

As linguagens de programação utilizadas e o *middleware* escolhido para o servidor se apresentaram maduras, fornecendo todos os subsídios necessários para a construção de um jogo *multiplayer*.

A implementação e a ligação entre todas as camadas se tornaram complexas ao longo do desenvolvimento do trabalho. Muitas vezes era preciso implementar um grande número de testes, para validar todo o desenvolvimento.

É importante ressaltar que o Flash como ferramenta visual é extremamente adequada, permitindo desenvolver o aspecto visual rapidamente. Para desenvolvimento de determinados tipos de jogos ele é uma poderosa ferramenta. O maior problema é a performance de processamento, que às vezes deixa a desejar.

Foi possível constatar que desenvolver um jogo educativo, sobretudo *multiplayer*, é realmente complexo em função da sua natureza interdisciplinar, pois envolvem questões de implementação, técnicas de programação, estratégias para aprendizagem, direção de arte, conceitos, enfim, estudos nas mais diversas áreas.

6.1 Trabalhos futuros

Até o momento, foram identificadas algumas possibilidades de trabalhos futuros, a saber:

- Com a geração de *logs* (arquivos de registro) de cada interação do jogador com os desafios propostos, seria possível analisar como está sendo o andamento do jogador no ambiente;
- Como o jogador é imerso em um mundo virtual, seria possível colocar personagens fictícios com seu raciocínio baseado em técnicas de IA (Inteligência Artificial) para

auxiliar e ou divertir o jogador durante sua jornada;

- No momento em que o jogador está respondendo perguntas dissertativas, o jogo poderia fornecer um feedback se a resposta está correta ou não através do uso de ontologias;
- Aprimorar as formas de interação com o mundo, fornecendo novos recursos como a utilização de uma *webcam*;
- Aprimorar a parte de perguntas e respostas, dando suporte a vídeos e imagens tanto para as perguntas, como para as respostas;
- Utilização do jogo para construir outros novos jogos educativos. Como as extensões são separadas, é possível utilizar apenas os desafios ou apenas o mundo virtual. Dentro do mundo virtual as ações dos *tiles* fornecem suporte a desenvolvimento de qualquer ferramenta por fora para utilizar dentro do mundo.

7 REFERÊNCIAS

ALBUQUERQUE, R. C., **Promovendo o Ensino-Aprendizagem de Educação Ambiental no Ensino Fundamental com Jogos Baseados em Ferramentas Computacionais**, PGIE/UFRGS, V. 6 N° 1, Julho, 2008.

AMORY, A. Building an Educational Adventure Game: Theory, Design and Lessons. In: **Journal of Interactive Learning Research**, v.12 num. 23, 2001.

BARANAUSKAS, M. C. C., ROCHA, H. V., MARTINS, M. C., D'ABREU, J. V. V. **Uma taxonomia para usos do computador em educação**. In: VALENTE, J. A. O Computador na Sociedade do Conhecimento. Brasília: Proinfo-SED Ministério da Educação, Governo Federal, 1999.

BARROS, L. A. **Suporte a ambientes distribuídos para aprendizagem cooperativa**. Rio de Janeiro, UFRJ, 1994.

BATTAIOLA, A. L. Jogos por Computador – Histórico, Relevância Tecnológica e Mercadológica, Tendências e Técnicas de Implementação. In: **XIX Jornada de Atualização em Informática**. Curitiba: SBC, v. 2, Julho, 2000.

BATTAIOLA, A. L., ELIAS, N. C. e DOMINGUES, R. G. et al Desenvolvimento de um Software Educacional com Base em Conceitos de Jogos de Computador In: **XIII Simpósio Brasileiro de Informática na Educação**. São Leopoldo: SBC, 2002.

BITTENCOURT, J. R. G., LUCIA M. Modelando Ambientes de Aprendizagem Virtuais utilizando Role-Playing Games In: **XIV Simpósio Brasileiro de Informática na Educação**. Rio de Janeiro: SBC, 2003.

BONGIOLO, C. E. F. et al. Subindo e Escorregando: jogo para introdução do conceito de adição de números inteiros. In: **Congresso da rede iberoamericana de informática na educação**, 1998.

BRAGA, A. **Jogos como sistemas: análise dos elementos do jogo que divertem o jogador**. Universidade Anhembi Morumbi, Design de Games, Brasil Pontifícia Universidade Católica de São Paulo (PUC-SP), Departamento de Ciência da Computação, 2007.

CAMPOS, M de B. Hiperhistórias na Educação: Um meio de educação/reeducação psicomotora. In: **VI Simpósio Brasileiro de Informática na Educação**. Anais. Florianópolis: SBC-UFSC, 1995.

CLUA, E. W. G., JUNIOR, C. L. de L e NABAIS, R. J. de M. Importância e Impacto dos Jogos Educativos na Sociedade. In: **I Workshop Brasileiro de Jogos e Entretenimento Digital**. SBC: Fortaleza, 2002.

CYBIS, W. **Engenharia de Usabilidade: Uma Abordagem Ergonômica**. Florianópolis: Laboratório de Utilizabilidade de Informática, 2003.

DIAS, C. A. **Usabilidade na web: criando portais mais acessíveis**. Rio de Janeiro: Alta Books, 2003.

FELLERS, J. W. **People Skills: Using the Cooperative Learning Model to Teach Students People Skills**. Interfaces, v. 26, 1996.

ISO 9241 Part 1. **Ergonomic requirements for office work with visual display terminals, Part 1 General Introduction**. International Standard ISO 9241-1, 1993.

JONASSEN, D. **O Uso das Novas Tecnologias na Educação a Distância e a Aprendizagem Construtivista**. Brasília: Em Aberto, ano 16, n. 70, abr/jun, 1996.

LÉVY. **A inteligência coletiva Por uma antropologia do ciberespaço**. São Paulo: Loyola, 1998.

MAKAR, J. **ActionScript for Multiplayer Games and Virtual Worlds**. Berkeley, United States of America, New Riders, 2009.

MALONE, T. **Toward a theory of intrinsically motivating instruction**. Cognitive Science, 1981.

MANTOAN, M. T. E.. **O processo de conhecimento - tipos de abstração e tomada de consciência**. Campinas: NIED-Memo, 1991.

MANTOVANI, A. M. et al. L.I.S. – Learning in the Space: Ambiente de Aprendizagem Computacional Cooperativo. In: **Congresso Iberoamericano de Informática Educativa, 5, 2000**, Viña del Mar, Chile: Ribie, 2000.

MELLO, H. **Ambientes computacionais y desarrollo cognitivo: perspectiva na psicologia**. Boletim de Informatica Educativa. n.2, v.2, 1989.

NETO, E. R.. Laboratório de matemática. In: **Didática da Matemática**. São Paulo: Ática, 1992.

PAPERT, S. **Constructionism: A New Opportunity for Elementary Science Education**. A proposal to the National Science Foundation, Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, Cambridge, Massachusetts, 1986.

PIAGET, J. **A Psicologia da Inteligência**. Lisboa: Fundo de Cultura, 1967.

RIZZO, G. O Método Natural de Alfabetização. In: **Alfabetização Natural**. Rio de Janeiro: Francisco Alvez, 1988.

ROCHA, E. J. T. S., RAMALHO G. L. e SANTOS A. L. M. Forge 16V: **Um Framework para Desenvolvimento de Jogos Isométricos**. Dissertação de Mestrado em Ciência da Computação. Centro de Informática, Universidade Federal de Pernambuco, 2003.

RODRIGUES, L. C., **Um estudo das formas de comunicação e interação em redes sociais de jogos on-line**. Universidade Presbiteriana Mackenzie, 2008.

FREITAS, S. **Learning in Immersive Worlds**, 2006 – Disponível na Internet em http://www.jisc.ac.uk/media/documents/programmes/elearninginnovation/gamingreport_v3.pdf acessado em 14/04/2009.

SILVA, F. M. Protetores das Tartarugas - um ambiente virtual colaborativo para a educação ambiental. In: **IV Seminário Jogos eletrônicos, educação e comunicação**. Salvador, 2008.

SILVEIRA, S. e BARONE, D. **Estudo e construção de uma ferramenta de autoria multimídia para elaboração de jogos educativos**. Porto Alegre: UFRGS., 1998.

SOELTL, F. A.. **Como está o e-learning no Brasil?** E-learning Brasil. Revista Vencer, n. 53, 2004.

VALENTE, J.A. **Por Quê o Computador na Educação**. Campinas: Gráfica da UNICAMP, 1993.

VALENTE, J.A. **Diferentes Usos do Computador na Educação**. Campinas: Gráfica da UNICAMP, 1993.

VALENTE J.A. **Computador e Conhecimento: repensando a educação**. Campinas: Gráfica UNICAMP, 1993.

VALENTE, J.A. **O Computador na Sociedade do Conhecimento**. Campinas: 1. ed., Nied/Unicamp, 1999.

VYGOTSKY, L. V. **A formação social da mente: desenvolvimento dos processos mentais superiores**. São Paulo: Martins Fontes Editora Ltda, 1994.

WINCKLER, M. A. A., FARENC, C., PALANQUE, P. e PIMENTA, M. S.. Avaliação da Navegação de Interfaces Web a partir de Modelos. In: **Workshop sobre fatores humanos em sistemas computacionais**. Florianópolis, 2001.

ANEXO A

Questionário - Aprendiz

Estou desenvolvendo um ambiente de aprendizagem no modelo de um jogo. O jogo consiste basicamente em apresentar desafios para os jogadores, esses desafios são apresentados em forma de pergunta / resposta.

No momento em que o aluno acessa o jogo, se for a primeira vez, ele poderá criar um avatar. Após a criação do avatar ele irá entrar no mundo e poderá se comunicar com outros personagens.

Conforme ele vai explorando o mundo ele poderá acessar sessões que terão diversos desafios, esses desafios serão um conjunto de perguntas e respostas adicionadas pelos mestres do jogo. Neste jogo, o professor assume o papel de mestre. É ele quem administra o sistema, cadastrando perguntas e visualizando relatórios dos alunos.

Vale ressaltar que o ambiente será todo WEB.

Conto com sua ajuda =), um forte abraço!

Para informações do andamento do projeto você pode consultar o endereço:
<http://mundusmmog.blogspot.com/>

* Required

1) Quantos anos você tem? *

2) Sexo *

- Masculino
- Feminino

3) Você costuma jogar jogos eletrônicos? *

- sim
- não

4) Quantas vezes por semana? *

5) Você se interessa por jogos que apresentam desafios? *

- sim
- não

6) Você se interessa por jogos em que os jogadores participam de equipes? *

- sim

- não

7) Qual a melhor forma de se comunicar com os membros da sua equipe? Conversa com áudio, texto (chat), vídeo? *

8) Qual cenário (ambiente) de jogo você acha mais interessante (floresta com cabanas, deserto, castelo medieval, ou qualquer outro tipo que você goste)? *

9) Quando você tem a possibilidade de escolher o seu personagem em um jogo, quais características gostaria de dispor para personalizá-lo (cabelo, cor da pele, corpo, vestimenta)? *

10) Você se sente motivado, inicialmente, a participar de um jogo que tenha uma finalidade educativa, ou seja, se o conteúdo dos desafios visa a aprendizagem de um conteúdo? *

11) Deixe aqui suas sugestões, caso deseje.

ANEXO B

Questionário - Mestre

Estou desenvolvendo um ambiente de aprendizagem no modelo de um jogo. O jogo consiste basicamente em apresentar desafios para os jogadores, esses desafios são apresentados em forma de pergunta / resposta.

No momento em que o aluno acessa o jogo, se for a primeira vez, ele poderá criar um avatar. Após a criação do avatar ele irá entrar no mundo e poderá se comunicar com outros personagens.

Conforme ele vai explorando o mundo ele poderá acessar sessões que terão diversos desafios, esses desafios serão um conjunto de perguntas e respostas adicionadas pelos mestres do jogo. Neste jogo, o professor assume o papel de mestre. É ele quem administra o sistema, cadastrando perguntas e visualizando relatórios dos alunos.

Vale ressaltar que o ambiente será todo WEB.

Conto com sua ajuda, um forte abraço!

Para informações do andamento do projeto você pode consultar o endereço:
<http://mundusmmog.blogspot.com/>

* Required

1) Sexo *

- Masculino
- Feminino

2) Você é professor? Se sim, de que?

3) No momento em que o aluno acessa o desafio, deveria ter conteúdos relacionados para ele poder buscar uma fonte de informações ou até mesmo dicas p/ as respostas? De que tipo (links para material, vídeos, tutoriais, ...)? *

4) Como deveria ser o feedback para o aluno do que ele já respondeu e até mesmo no momento em que ele responde? Uma mensagem de texto? Um áudio? *

5) Quais informações são relevantes na hora do cadastro da pergunta? *

6) As perguntas além da classificação do tema (matemática, história...) deveriam ter algum outro tipo de classificação (ex: estimula raciocínio, interdisciplinaridade, senso comum, trabalho em equipe, criatividade ...)? *

7) No momento que o professor cadastra um novo desafio (conjunto de perguntas), ele poderia colocar uma senha para acesso restrito para alguns alunos? Esta característica é importante? *

8) Quais informações deveriam aparecer no relatório de atividades do aluno? *

9) A cada resposta do aluno será gerado um LOG, quais são as informações relevantes para serem armazenadas? * Log é um arquivo que armazena registros (ex: ações do jogador)

10) Criar um mundo para o jogador explorar e acessar as sessões específicas é interessante, há outra abordagem que você acha mais interessante? *

11) Quando o aluno estiver respondendo perguntas, ele pode convidar outros amigos para participar do mesmo desafio desde que esteja no início? *

- Sim
- Não

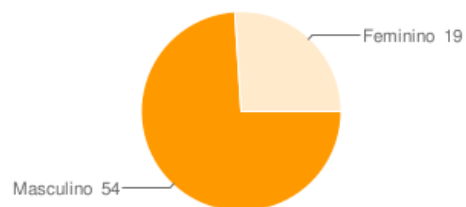
12) Como o jogo poderia instigar o aluno a querer jogar mais?

13) Que outros tipos de desafios poderiam ser abordados dentro do jogo, levando em conta que o jogo não faz distinção de idades?

ANEXO C

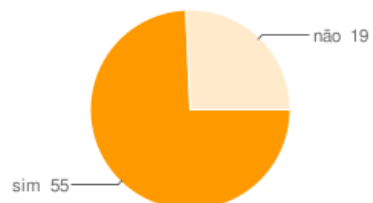
Dados estatísticos – Questionário do Aprendiz

Sexo



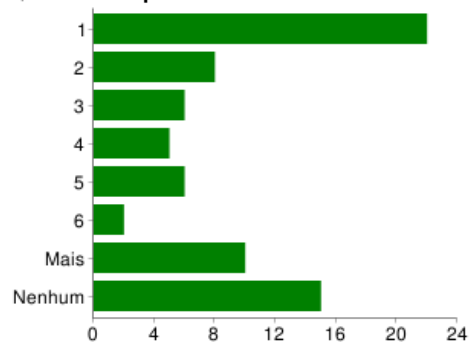
Masculino	54	74%
Feminino	19	26%

Você costuma jogar jogos eletrônicos?



sim	55	74%
não	19	26%

Quantas vezes por semana?

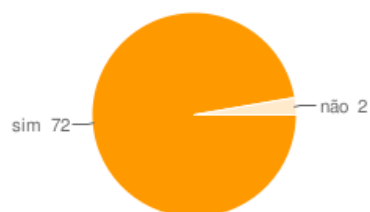


1	22	30%
2	8	11%
3	6	8%
4	5	7%
5	6	8%
6	2	3%
Mais	10	14%
Nenhum	15	20%

Você se interessa por jogos que apresentam desafios?

sim 72 97%

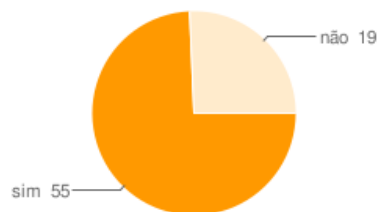
não 2 3%



Você se interessa por jogos em que os jogadores participam de equipes?

sim 55 74%

não 19 26%



Respostas – Questionário do Aprendiz

1	23
2	sim
3	3
4	sim
5	sim
6	Chat
7	Hmmm, difícil. Gosto da idéia do castelo medieval...
8	Tudoo que for possível! Hehehe. Incluindo criação de roupas diferentes. =D
9	Acredito que isso vai depender do formato do jogo. Se for divertido, instigante, e não monótono, sim.
10	
11	Feminino

1	22
2	sim
3	Mais
4	sim
5	sim
6	Chat
7	Todos exceto os espaciais e de ficcao cientifica
8	Nao acho importante
9	Se depois da parte intelectual vier a parte de ação.. talvez
10	
11	Masculino

1	25
---	----

2 não
 3 1
 4 sim
 5 não
 6 Para mim a melhor forma é através do áudio. Acho que não é necessário o vídeo.
 7 Eu prefiro ambientes mais fechados como florestas. Não gosto de ambientes com muito espaço livre.
 8 estilo e cor do cabelo, cor da pele, roupas e armas.
 9 acho que depende muito do jogo, ele não pode ficar muito repetitivo. Sempre deve possuir novos desafios e níveis de dificuldade.
 10
 11 Masculino

1 20
 2 não
 3 1
 4 sim
 5 sim
 6 texto
 7 medieval
 8 cor da pele e do cabelo, e opções de roupas
 9 +- depende do conteúdo
 10
 11 Feminino

1 23
 2 sim
 3 1
 4 sim
 5 sim
 6 chat
 7 castelo, ruínas de guerra
 8 cabelo preto, cor da pele morena, vestimenta de guerra
 9 sim
 10
 11 Masculino

1 23
 2 sim
 3 2
 4 sim
 5 sim
 6 Chat
 7 era medieval em geral...com castelos...bosques...camponeses....colonias....cidades
 8 cabelo cor de pele barba vestimenta raça (nao precisa ser humano oras)
 9 Sim, desde que o conteúdo seja trabalhado de uma forma que integre diversão e aprendizado e que nao fique algo monotono.

10 A ideia é muito boa. Seria interessante se a medida que o jogador for evoluindo no jogo ele ganhe algum incentivo tipo dinheiro pra comprar itens, ou algum acesso a lugares restritos.

11 Masculino

1 22
 2 sim
 3 2
 4 sim
 5 sim
 6 texto
 7 floresta
 8 todas características possíveis
 9 não
 10
 11 Masculino

1 22
 2 não
 3 Nenhum
 4 sim
 5 sim
 6 texto(chat) e audio/video
 7 autódramo de formula 1..
 8 cabelo
 9 sim
 10
 11 Masculino

1 25
 2 sim
 3 3
 4 sim
 5 não
 6 chat
 7 futurista
 8 todos estes =)
 9 se o conteudo for de meu interesse, sim!
 10
 11 Masculino

1 23
 2 sim
 3 2
 4 sim
 5 sim
 6 áudio

7	castelo medieval
8	nenhuma
9	as vezes
10	bom trabalho
11	Masculino

1	28
2	sim
3	4
4	sim
5	não
6	Conversa com áudio
7	O cenário mais interessante é aquele que é bem construído e que se identifique com o tema do jogo.
8	Estes citados dão uma boa gama de opções. Não precisa mais.
9	Sim.
10	Uma ótima idéia! Gostaria de apreciar a aplicação quando do término do projeto.
11	Masculino

1	28
2	não
3	Nenhum
4	sim
5	não
6	audio
7	idades
8	corpo
9	nao
10	
11	Masculino

1	33
2	sim
3	1
4	sim
5	sim
6	audio e video
7	floresta
8	personagem Robusto
9	visaria testar conhecimentos
10	Crie uma maneira de os usuarios poderem compararem seus resultados com os de outros jogadores
11	Masculino

1	28
2	sim
3	1

4	sim
5	sim
6	audio, texto
7	floresta, castelo medieval
8	preto - metaleiro cabelo arrepiado e longo
9	com certeza (estamos precisando disso)
10	
11	Feminino

1	23
2	não
3	Nenhum
4	sim
5	sim
6	Áudio e texto(chat).
7	Qualquer tipo.
8	Todos os possíveis.
9	Sim
10	
11	Masculino

1	25
2	sim
3	3
4	sim
5	sim
6	audio
7	Medieval
8	Vestimenta
9	A principio nao.
10	
11	Masculino

1	21
2	sim
3	Mais
4	sim
5	sim
6	Depende muito do objetivo. Se tratando de um jogo de ação, eu prefiro audio pela velocidade de comunicação, mas no geral, conversa em texto está de bom tamanho. Vídeo eu não acho ideal uma vez que desvia o foco de atenção do jogador.
7	Ambientes mais urbanos (Cidade, ruas, prédios...)
8	Cabelo e vestimenta já está bom. Nada que demore demais pra fazer.
9	Não sei responder essa pergunta!
10	
11	Masculino

1	17
2	sim
3	Mais
4	sim
5	sim
6	Áudio e chat me parecem mais do que suficientes.
7	Gosto muito de cenários com água, poderia ser uma praia ou algo parecido. Floresta com cabanas parece interessante e misterioso.
8	Os já citados são os mais importantes.
9	Sinceramente, não.
10	
11	Masculino

1	21
2	sim
3	Mais
4	sim
5	sim
6	Áudio, ou texto (no caso de conexões lentas).
7	Qualquer ambiente medieval.
8	Todas as citadas acima, mais faces diversas (com bigode ou sem p.ex.).
9	Sim.
10	
11	Masculino

1	27
2	sim
3	1
4	sim
5	sim
6	chat e video.
7	floresta, deserto, cidade a noite...
8	todas
9	sim.
10	
11	Feminino

1	19
2	sim
3	2
4	sim
5	sim
6	audio + chat
7	floresta
8	vestimenta
9	não
10	

11	Masculino
----	-----------

1	20
2	sim
3	2
4	sim
5	sim
6	Áudio e texto.
7	Floresta com cabanas.
8	Cabelo, roupa e aparência física(gordo, magro, forte)
9	não.
10	
11	Masculino

1	26
2	sim
3	5
4	sim
5	sim
6	áudio e texto
7	floresta, deserto, castelo medieval, cidades fantasma, futurísticos, etc.
8	todos possíveis, inclusive aspecto físico e mental (assassino, terrorista, herói, etc.)
9	depende o conteúdo... Se você for capaz de misturar arquitetura de computadores com um desafio que realmente ensine o usuário, com certeza será bom jogar.
10	
11	Masculino

1	25
2	sim
3	2
4	sim
5	não
6	N
7	Cidade
8	vestimenta
9	sim
10	
11	Masculino

1	27
2	não
3	Nenhum
4	sim
5	sim
6	áudio, vídeo
7	floresta, desertos e cidades
8	corpo, vestimenta

9	com certeza
10	
11	Masculino

1	23
2	sim
3	6
4	sim
5	sim
6	Estou acostumado com texto, mas áudio é mais interessante quando possível.
7	Prefiro um ambiente variado.
8	Cabelo.
9	Se for divertido... =D
10	Boa sorte!
11	Masculino

1	22
2	sim
3	1
4	sim
5	sim
6	texto e áudio (dependendo do jogo - se precisar de respostas rápidas com as mãos, melhor deixá-las ocupadas com isso, usando áudio para comunicação)
7	Em jogos de tiro, por exemplo, a variação dos cenários torna o jogo diferente cada vez que se joga. Depende do tipo de jogo e qual o objetivo. O mais interessante seria uma "evolução", com os cenários tornando-se mais complexos e difíceis à medida que o j
8	interessante seria se cada personagem tivesse uma forma diferente de se jogar (um facilita algumas coisas e dificulta outras, outro faz o contrário, como "habilidades" de um personagem). outras características, em ordem: corpo, vestimenta, cabelo, cor da
9	depende do conteúdo. O "desafio sebrae" e o "em ação" (http://folhainvest.folha.com.br/) são interessantes
10	o que mais motiva num jogo é o desafio a ser superado
11	Masculino

1	24
2	não
3	Nenhum
4	sim
5	não
6	audio
7	qqr
8	cabelo, barba, bigode, cor da pele, camiseta, calças
9	não
10	
11	Masculino

1	25
---	----

- | | |
|----|--|
| 2 | não |
| 3 | Nenhum |
| 4 | sim |
| 5 | sim |
| 6 | Quando jogo, geralmente é single player, mas acho que a melhor forma de comunicação é com áudio. |
| 7 | Acho maneiro florestas e castelos. |
| 8 | Corpo e vestimenta. Os demias não são importantes. |
| 9 | Sim. Sempre é interessante quando pode se aprender com um entretenimento. |
| 10 | Achei bem interessante seu projeto de TCC. Parabéns e sucesso. |
| 11 | Feminino |

- | | |
|----|---|
| 1 | 18 |
| 2 | sim |
| 3 | 4 |
| 4 | sim |
| 5 | não |
| 6 | audio, vc acaba se distraindo pois não tem q digitar |
| 7 | não tenho preferência eu acho desde q esteja bem detalhado |
| 8 | loiro, branco, corpo tanto faz, vestimenta de acordo com o que o personagem irá fazer |
| 9 | Sim, não sendo massante e repetitivo |
| 10 | |
| 11 | Masculino |

- | | |
|----|--|
| 1 | 20 |
| 2 | sim |
| 3 | 1 |
| 4 | sim |
| 5 | sim |
| 6 | áudio e/ou texto |
| 7 | floresta com cabanas |
| 8 | características físicas |
| 9 | sim, mas que não seja o único objetivo |
| 10 | |
| 11 | Masculino |

- | | |
|----|---|
| 1 | 30 |
| 2 | sim |
| 3 | Mais |
| 4 | sim |
| 5 | sim |
| 6 | audio e texto |
| 7 | depende do jogo, um fps é interessante que tenha áres em florestas, ja um rpg de ação se tiver masmorras fica muito bom |
| 8 | o máximo de customização possível |
| 9 | sim |
| 10 | |

11	Masculino
----	-----------

1	22
2	sim
3	4
4	sim
5	sim
6	Depende da velocidade do gameplay do jogo. Caso o jogo requira muitos comandos do teclado e atenção às imagens da tela, como um jogo de ação, o melhor tipo de comunicação é por áudio. Caso o jogo seja mais lento a comunicação por texto é melhor.
7	Medieval ou fantasia medieval.
8	A maior quantidade possível de características.
9	Não muito.
10	Um ranking de classificação dos usuários sempre ajuda a acirrar a disputa e estimular os participantes.
11	Masculino

1	17
2	sim
3	2
4	sim
5	sim
6	Texto ou áudio
7	ambientes futurísticos
8	rosto, corpo, equipamento, etc.
9	Não muito
10	
11	Masculino

1	28
2	sim
3	6
4	sim
5	sim
6	audio
7	castelo medieval
8	nenhuma
9	com certeza
10	boa sorte no tcc
11	Masculino

1	19
2	sim
3	Mais
4	sim
5	sim
6	audio eh provavelmente a melhor ideia (ts), no caso do video tornaria o jogo muito

lento por causa da sobrecarga no computador, mas o chat tbm eh muito util

7 atualmente nos jogos online(mmorpg) todos esses cenarios existem num mesmo mundo, a utilização d um unico ambiente pode tornar o jogo xato. eu prefiro jogos com o ambiente medieval com castelos e tudo +.

8 a principio todas as caracteristicas deveriam poder ser mudadas. um bom exemplo seria o jogo The Sims 2 onde vc pode modificar totalmente suas personagens. outro jogo eh o Perfect World, onde vc pode modificar todo seu corpo: vc pode escolher o tipo d cab

9 se o jogo possuir desafios ja eh uma motivação inicial, caso aja uma aprendizagem juntamente com os desafios soh acredito q os jogadores sairao ganhando. o jogo n deve ser exclusivamente educacional pq issu o tornara xato.

10 sou aluno do curso d computação da ucs, apesar de n ter experiencia em desenvolvimento web, tenho amplo conhecimento em jogos, alem de criação d mundos virtuais e desenvolvimento 3D.caso precisem meu email eh pablo081989@hotmail.com ou pragross1@ucs.br bo

11 Masculino

1 18
2 não
3 Nenhum
4 não
5 não
6 NENHUM
7 NENHUM
8 NENHUM
9 NENHUM
10 NADA
11 Masculino

1 27
2 sim
3 1
4 sim
5 não
6 Como não prefiro jogos em equipes, não há comunicação.
7 Medieval/Espacial
8 Indiferente
9 Sim existem jogos com esse fim que são muito interessantes.
10 Bom trabalho!
11 Masculino

1 29
2 sim
3 1
4 sim
5 não
6 chat
7 floresta
8 cabelo, cor da pele, corpo, vestimenta, roupas

9	nao
10	
11	Masculino

1	21
2	não
3	Nenhum
4	sim
5	sim
6	Texto e áudio
7	Floresta, Montanhas, Gelo
8	Geralmente prefiro escolher um personagem já existente ou postar uma imagem de minha preferência.
9	Sim
10	Sugiro que você não se detenha na criação do avatar. Seria interessante criar alguma outra forma de desafio além de perguntas e respostas. Como por exemplo encontrar padrões ou decifrar códigos em um texto ou sequência numérica. Obviamente, para descobrir
11	Masculino

1	22
2	sim
3	5
4	sim
5	sim
6	chat texto, ou se caso todos tiverem uma boa conexao adsl via audio penso que video atrapalharia o escopo do jogo
7	castelo medieval ou florestas
8	vestimentas
9	porque nao?
10	
11	Masculino

1	20
2	sim
3	2
4	sim
5	sim
6	vídeo
7	castelo mediaeval
8	características físicas em geral
9	dependendo do jogo...
10	
11	Masculino

1	18
2	sim
3	1

4	sim
5	sim
6	Texto
7	Castelo medieval
8	Guerreiro medieval
9	Sim
10	
11	Masculino

1	22
2	sim
3	3
4	sim
5	não
6	texto
7	castelo medieval
8	vestimenta
9	sim
10	
11	Masculino

1	30
2	sim
3	1
4	não
5	não
6	Chat
7	Ruas de uma Cidade.
8	cabelo, cor da pele, corpo, vestimenta, acessórios
9	Sim.
10	
11	Masculino

1	22
2	sim
3	Mais
4	sim
5	sim
6	Áudio e chat.
7	Florestas e Cidades
8	Todos esses itens.
9	Educativo ?
10	
11	Masculino

1	27
2	sim

3 Mais
 4 sim
 5 sim
 6 audio - uso o software Mumble
 7 Sou meio suspeito a falar por que sou muito viciado em jogos, gosto de muitos. Gosto de todos estes, mas acho q o q mais divertido ainda é o na cidade, estilo GTA. Pra mim é um divisor de aguas nos jogos, onde tu tem uma cidade, e pode ir livremente para
 8 Nao importa muito, mas prefiro que seja realista e de acordo com o jogo. Mas lendo a explicação la de cima, acho assim, se for um jogo de guerra, tem q ter vestimentas militares, medieval, armaduras, na cidade, roupas casuais. A cara do personagem nao imp
 9 Depende muito. Tem jogo educativo que é um saco!!! Agora por exemplo, um Age of Empires ensina um monte sobre historia. Acho mais efetivo quando o conteudo está como plano de fundo, e o aprendizado ocorre naturalmente.
 10 Da uma olhada nesta ferramenta q eu te falei, o Mumble. Ele é open source e serve para comunicação por voz entre jogadores. Pode te ser util. Boa sorte no teu trabalho! Espero ter ajudado.
 11 Masculino

1 17
 2 sim
 3 Mais
 4 sim
 5 sim
 6 Texto
 7 Coisas antigas, como 2ª guerra mundial.
 8 tipo de arma^^
 9 se o jogo for legal sim
 10
 11 Masculino

1 24
 2 sim
 3 1
 4 sim
 5 não
 6 Acho melhor na forma de texto
 7 Gostaria de uma mistura de ambientes: floresta, deserto, diferentes prédios,...
 8 Gosto de personalizar cabelo e vestimenta
 9 Sim, especialmente se forem apresentados de forma criativa
 10 Se um jogo tivesse o função educativa, seria interessante se apresentasse uma breve explicação dos assuntos de serem apresentados em perguntas ou desafios, o jogador poderia escolher entre acessá-las ou não.
 11 Feminino

1 25
 2 não
 3 Nenhum

4	sim
5	sim
6	Texto
7	Floresta
8	Tanto faz (de preferência que já venha pronto)
9	Com certeza
10	Acho interessante jogos estratégicos, que desenvolvam a lógica.
11	Feminino

1	10
2	sim
3	4
4	sim
5	não
6	CHAT
7	castelo medieval, gelo, navio pirata
8	cabelo, cor da pele, corpo e vestimenta e bichinhos de estimação
9	sim
10	
11	Feminino

1	28
2	sim
3	4
4	sim
5	sim
6	chat
7	Um cenário legal dos citados, seria floresta com cabanas
8	cabelo, corpo e a vestimenta
9	Sim, eu acho interessante jogos educativos
10	Interessante o desenvolvimento do jogo. Continue. Parabéns
11	Masculino

1	24
2	não
3	Nenhum
4	sim
5	sim
6	vídeo
7	castelo
8	vestimenta
9	Com certeza.
10	
11	Feminino

1	33
2	sim

3 Mais
 4 sim
 5 sim
 6 Áudio
 7 Espaço (com naves).
 8 Cabelo, pele, formato do rosto, roupas, adereços (óculos, brincos, etc).
 9 Não. Para mim, jogar é o momento em que posso fugir do trabalho e fazer coisas sem pensar muito.
 10 Acho tua iniciativa excelente. Por favor, não desanime pela minha resposta negativa acima. Como sugestão, recomendo o artigo <http://rpgvault.ign.com/articles/986/986323p1.html> , que discute os efeitos das premiações por progresso em jogos e traça paralel
 11 Masculino

1 22
 2 sim
 3 1
 4 sim
 5 sim
 6 Texto, audio
 7 Urbano moderno
 8 Todos. Desde o corpo até as roupas e acessórios.
 9 Sim.
 10
 11 Masculino

1 32
 2 sim
 3 3
 4 sim
 5 sim
 6 audio, chat
 7 urbano
 8 indio
 9 desafio na aprendizagem
 10
 11 Masculino

1 25
 2 sim
 3 5
 4 sim
 5 sim
 6 texto
 7 Estilo medieval ou ambientes que lembrem os samurais
 8 Todas as características que o deixem o mais parecido comigo possível
 9 Se o conteúdo for de meu interesse, sim

10	Divulgue o jogo quando estiver pronto
11	Masculino

1	23
2	sim
3	1
4	sim
5	sim
6	audio
7	floresta castelo
8	vestimenta rosto
9	sim
10	
11	Masculino

1	23
2	não
3	Nenhum
4	sim
5	sim
6	Penso que conversas com audio sejam mais dinamicas e interativas.
7	Gosto muito de cenários apocalipticos (ex Mad Max), pois me deixa muito mais livre psiquicamente. Porém os cenários medievais são otimos (mesmo sendo cliches).
8	Quanto mais características eu puder manipular, mais vou me identificar!
9	Sim, pois ja li que essa seria a tendencia para o futuro, e parece muito mais divertido que apenas ler e estudar convencionalmente. Porém a trama por tras do jogo educativo deve ser consistente e interessante também
10	Parabéns pela iniciativa. Espero experimentar o jogo!
11	Masculino

1	24
2	sim
3	3
4	sim
5	não
6	Tanto faz.
7	Floresta, cidades, shows (rock band)...
8	Cabelos, rosto, olhos, corpo, vestimenta, carro.
9	Sim.
10	
11	Feminino

1	22
2	não
3	Nenhum
4	sim
5	sim

- | | |
|----|---|
| 6 | Audio |
| 7 | Um ambiente urbano |
| 8 | Um personagem que se pareça comigo (mulher, morena, magra) ou um personagem "bonitinho". |
| 9 | Acho muito interessante, algo que instigue o pensamento, a pesquisa e que promova a aprendizagem. |
| 10 | Mas acredito que deva ser algo retroalimentado, que não seja algo estanque, que ao passar de alguns meses se saiba de cor o jogo. |
| 11 | Feminino |

- | | |
|----|---|
| 1 | 24 |
| 2 | sim |
| 3 | 1 |
| 4 | sim |
| 5 | sim |
| 6 | Eu penso que seja o audio, porque você não precisa deixar a tela do jogo, é só escutar. |
| 7 | Seria legal uma cidade, com mapas, onde vc se locomove, vários becos, endereços... |
| 8 | sexo, cabelo, roupas, estilos, nome, cor da pele, tipo físico, rosto. |
| 9 | Sim, acho interessante unir as duas coisas. |
| 10 | Criar um tipo de pontuação, seria legal um jogo que não tivesse só 1 forma para ganhar, mas que fosse rizomática e aleatória. |
| 11 | Masculino |

- | | |
|----|---|
| 1 | 22 |
| 2 | sim |
| 3 | 5 |
| 4 | sim |
| 5 | sim |
| 6 | ts, ventrilo ou chat padrão da guilda/clã/party |
| 7 | castelo medieval ou qualquer outro que possa promover interatividade com a história do local |
| 8 | feições do rosto, cabelo, tamanho do corpo, raça, cor dos olhos |
| 9 | sim |
| 10 | seria interessante um jogo que ao mesmo tempo traz desafios mentais, esteja sendo interpretado através de uma história atrativa e que apresente grande interação. |
| 11 | Masculino |

- | | |
|----|--|
| 1 | 30 |
| 2 | não |
| 3 | 1 |
| 4 | sim |
| 5 | não |
| 6 | Gosto de jogos onde cada um joga por si, sem equipes. |
| 7 | Castelo Medieval e floresta também é legal! |
| 8 | Depende do tipo de jogo. |
| 9 | Depende do jogo, mas acho interessante jogos com finalidade educativa. |
| 11 | Principalmente para se usar com crianças e adolescentes. |

10

11 Feminino

1 23

2 não

3 1

4 sim

5 sim

6 texto

7 Locais similares ao ambiente em que vivemos.

8 Todas essas mencionadas acima.

9 Sim

10

11 Feminino

1 33

2 não

3 Nenhum

4 sim

5 sim

6 chat

7 castelo medieval

8 características de corpo e de personalidade

9 sim

10

11 Feminino

1 20

2 sim

3 1

4 sim

5 não

6 áudio e chat

7 depende o jogo, atualmente estou jogando um que é de cenário medieval, com floresta.

8 o interessante é poder deixa-lo o mais parecido possível comigo, por isso cor de pele, olhos, cabelo, feições do rosto, tudo isso é de suma importância. acessórios para colocar na cabeça, no corpo (como tatuagens) e na roupa são muito bons também.

9 olha, até hoje os jogos desse cunho sempre eram chatos! então faça uma forma que não fique chata. crie um jogo legal, com cenários legais, avatares legais e no meio coloque as questões educativas, senão fica muito chato e não me interessa nem um pouco. a

10

11 Feminino

1 21

2 sim

3 5

4 sim

5	sim
6	Chat
7	Florestas sem cabanas, com árvores e animais
8	Cabelo, corpo, roupas, acessórios, calçados
9	Nunca participei, mas a proposta é bem interessante. Creio que me motivaria sim!
10	
11	Feminino

1	19
2	sim
3	5
4	sim
5	sim
6	Áudio, vídeo
7	Floresta com cabanas
8	Cor da pele, vestimenta, arma, personalidade
9	Sim, procuro aprender o máximo
10	
11	Masculino

1	20
2	não
3	Nenhum
4	sim
5	sim
6	Acredito que a melhor forma seria por vídeo e chat.
7	Floresta com cabanas
8	As características que mais se assemelham com a minha fisionomia.
9	COM CERTEZA.
10	
11	Feminino

1	20
2	sim
3	1
4	sim
5	não
6	Depende do tipo de jogo! Se for estilo MMORPG o melhor mesmo é através de salas de comunicação através de audio. Video pouco adianta, se vc tiver que estar de olho no jogo. Mas se for um jogo de perguntas, repostas, como parece ser a proposta desse, conve
7	Acho que quanto mais variado, melhor. Digo, qnto mais cenários, mais desafios diferenciados vc pode propor.
8	Bom, quanto mais possibilidades, melhor também! Temos como exemplo o aplicativo do buddypoke do orkut, onde vc pode criar uma caricatura fiel vc msmo, ou uma fantasiada (o que tmb é legal, poder inventar). Também tem a questão de vc poder adquirir itens a
9	Sim! Eu acho muito interessante! Sem dúvida, as vezes é mais gostoso aprender assim. O máximo de jogo com finalidade educativa que eu já joguei, foram jogos em flash com

exercícios de inglês, e um que fala sobre economia de energia em casa, no site da WWF

10 Olha, eu achei uma idéia legal. Só que, sei que não vai ser fácil! Pode ser que os alunos estejam bem preparados para lidar com ambientes virtuais, mas mesmo aqui na região da serra (RS), não é todo mundo que tem facilidades. E o mesmo se equivale para pr

11 Feminino

1 25

2 sim

3 1

4 sim

5 não

6 Audio e video

7 Cenário Urbano

8 Roupas normais do cotidiano

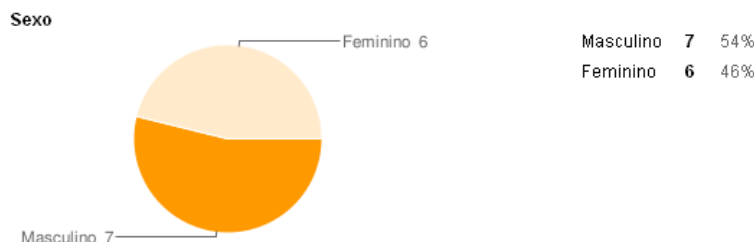
9 Sim, acredito que é uma forma interessante de se aprender

10

11 Masculino

ANEXO D

Dados estatísticos – Questionário do Mestre



Respostas – Questionário do Mestre

- 1 Feminino
- 2 sim, videos e tutoriais animados
- 3 Pode ser uma mensagem de texto, o importante é o retorno a esse aluno.
- 4 "saber o q perguntar é base. entao tem q saber qual é o objetivo do jogo. tipo a pergunta q q Alice faz apra o coelho: como faço para sair daqui. E o coelho responde: Depende onde vc quer ir??
"
- 5 sim, pensando em estímulos conforme o objetivo do jogo, mas pode ter a interdisciplinaridade e o trabalho em equipe.
- 6 nao é necessario.
- 7 todo a caminhada dele, todos os acessos, as tarefas efetuadas, qtas vzes enviou a tarefa.
- 8 o historico deste aluno.
- 9 sim acho interessante.
- 10 sim
- 11 tendo q ultrapassar obstaculos, para avançar apra o proxima tarefa....
- 12 desafios de raciocinio logico, estrategias etc

- 1 Feminino
- 2 Sim, todo o material é válido, desde que seja realmente significativo para o problema do aluno. Não adianta ser material muito extenso, tem que ser bem do foco do aluno. Conforme o tipo de material disponível, o aluno pode se desmotivar, em vez de se motivar buscando novas fontes de informação. Cuidado com os materiais multimídias, pois nem todos os alunos terão recursos para exibir, ouvir, etc. este tipo de material. Conforme o ambiente que queres atingir, os recursos tecnológicos são pobres.
- 3 O quanto antes o aluno tiver o feedback melhor e mais detalhado também. Deveria ter sempre uma mensagem de texto acompanhado de áudio, pelas questões levantadas acima.
- 4 Acho que seria importante ter bem claro qual o objetivo da pergunta.
- 5 Por área estaria bom, pois de acordo com a pergunta realizada é que o raciocínio, interdisciplinaridade, etc... vão estar sendo consideradas. O ideal é que em qualquer área de conhecimento, o aluno tenha que pensar, realizar relações para responder as questões. O

sucesso estará no tipo de pergunta elaborada e os subsídios que tu darás para eles responderem.

6 Seria para formar grupos específicos? Não entendi.

7 Quais foram as questões corretas e as incorretas, Nas incorretas o que ele respondeu e principalmente uma resposta bastante correta que o leve a entender onde foi seu erro.

8 Quais foram as ações executadas, o caminho que ele percorreu, as interações que ele fez com outros jogadores (se for o caso), o uma análise de suas respostas de acordo com o que era esperado dele.

9 O caminho do jogador deve ser guiado para que ele passe dos níveis de dificuldades mais fáceis para os mais difíceis.

10 sim

11 Se fornecer perguntas que instigue a sua curiosidade e que tenha uma interface adequada.

12

1 Masculino

2 Digamos que sejam apresentados conteúdos através dos quais ele desenvolva o conhecimento e possa resolver o problema. Após ele resolver um requisito mínimo do assunto, ele poderá questionar o professor ou outros personagens no ambiente para ajudá-lo a resolver o problema. Porém, se o outro informar a resposta, ambos devem voltar ao nível anterior sendo exigido uma pontuação maior para a mudança de nível.

3 Talvez através de um símbolo, veja semiologia gráfica. Algo que ele possa identificar como resposta correta, incompleta, ou errada.

4 Palavras-chaves são fundamentais. Ajudam a definir o assunto, ou pelo menos chega bem próximo.

5 Essa informação é interessante para o avaliador, que pode depois orientar o aluno a fim de melhorar determinados tipos de classificação.

6 Essa característica vai de encontro a resposta anterior. Ou seja, se ele verifica que determinados alunos precisam estimular o raciocínio, desenvolve-se atividades para esse grupo de alunos. Atividade essa só acessada por eles através de determinada senha.

7 O nível no qual se encontra, as disciplinas que estão sendo cursadas, a pontuação atual e a pontuação para mudança de nível.

8 "Disciplinas, respostas certas e número de tentativas para acertá-las, os erros, pontuação, etc..."

Outras informações irão depender do perfil da disciplina.

Disciplinas lógicas são mais simples nesse aspecto."

9 Talvez haja, mas como não parei para pensar no assunto fica difícil opinar.

10 não

11 Isso é difícil. Alguns jovens quando usam determinado jogo, não o fazem seguindo as regras. Isso devido a própria dificuldade existente. Portanto, tem de haver uma cobrança, algo que ele realmente ganhe para jogar.

12 Não havendo distinção de idades, o melhor é pensar em um jogo de estratégia com desafio intelectual.

1 Masculino

2 Sim, isto evitaria alguma frustração quando a pergunta for complexa demais. Pode ser qualquer formato de arquivo.

3 Bah... tua criatividade é que manda. Pode ser até o flash de uma espada laser passando

raspando pela sua cabeça...

4 Hum... isto pode ser complexo. Além da pergunta e resposta(s)? Talvez pistas para a resposta, além dos links para infos adicionais. Se o jogo for dinâmico, bem que o Mestre poderia parametrizar uma ação sobre a pergunta-resposta e também possibilitasse o "encadeamento" de perguntas relacionadas. Por exemplo, algumas perguntas podem depender de responder anteriores, criando uma árvore de decisões parametrizável, levando o aluno por labirintos de conhecimento... (putz, viajei agora)

5 hum... o cadastro da pergunta está ficando complexo. Acho que deveria ter sim, mas deve ser genérica o suficiente para o Mestre não ficar muito tempo "navegando" pelas classificações no cadastro da pergunta.

6 Bah, acho que não. A princípio eu gostaria que todos participassem. Se estiver dentro do contexto do UCSVirtual, já temos a garantia que apenas os alunos da UCS usam.

7 Tem um relatório de atividades???? Primeiro eu gostaria de saber pra que ele serve.... (:D

8 Acho que depende do tipo do desafio e dos objetivos do Mestre. Não sei quando o tempo de resposta é relevante, ou quantas dicas tiveram que ser usadas, ou quantas "vidas" o aluno gastou até acertar. Por enquanto eu gostaria de ver o "rastros" do aluno até alcançar o objetivo.

9 Colocar personagens neste mundo, modeláveis pelo Mestre de maneira que o aluno pudesse encontrar companheiros e adversários neste mundo. Modelar o comportamento dos personagens em função do desafio. Assim, por exemplo, se as perguntas são muito fáceis, eu poderia criar um pequeno "elfo" que iria apenas tentar confundir o aluno com pistas falsas. Também seria interessante criar uma trama em torno destes personagens, mas acho que agora isto extrapola teu trabalho.

10 sim

11 Como eu disse antes, se fosse possível modelar personagens e uma trama. Poderíamos criar desafios relacionados na forma de capítulos ou fases com complexidade crescente. Mas também, orientar o aluno a "aplicar" os conhecimentos adquiridos. Aí eu não sei muito bem como. Se o assunto é física, poderíamos ter uma fase teórica e a próxima prática em que ele trabalharia em um "laboratório" para testar os conhecimentos anteriores. Ou então, à medida que ele avança nas fases, premiá-lo com permissões para ele mesmo fazer "ajustes" no jogo e reapplicá-lo a seus colegas. Ele poderia ser um co-orientador do Mestre. Por exemplo, criando categoria de Mestres em que o próprio aluno pode ir avançando. Claro que o aluno teria que trabalhar num "clone" da página do jogo. (:D

12 "Putz... acho que escrevi tudo antes... Eheheh"

1 Feminino

2 Textos, material didático.

3 Um feedback simples, uma msg curta.

4 Enunciado da questão e respostas possíveis.

5 Seria interessante. Esta classificação seria apresentada ao aluno?

6 Isso seria importante se o desafio não deve estar disponível a todos.

7 Os desafios que o aluno acessou, seu desempenho na realização das tarefas solicitadas.

8 Resposta do aluno, tempo necessário para a realização da tarefa, número de tentativas,...

9 Acho interessante se o mundo for dinâmico, foi se for estático o aluno rapidamente perderá seu interesse.

10 sim

- 11 Ele poderia ter um enredo... envolver o aluno na compreensão ou investigação de um fato misterioso... instigar a curiosidade talvez.
- 12 Os desafios variam ao longo das idades. Este tema deveria ser abordado com professores de ensino fundamental e médio, que tem mais experiência e preocupação em despertar nos alunos o gosto da leitura, da aprendizagem...
- 13 Programação e IA.

- 1 Masculino
- 2 Links para materiais textuais e links para applets para simulação computacional de algoritmos
- 3 Acho que um email seria mais produtivo se estamos falando de atividades não sincronas. Também acho que seria útil um vídeo com o aluno utilizando um recurso computacional, por exemplo uma applet.
- 4 "A possibilidade de utilizar texto é claro, mas também que colocar hiperlinks, postar arquivos de qualquer natureza.
- Não sei se entendi bem esta pergunta."
- 5 Trabalho em equipe seria muito útil.
- 6 "Eu acho isto interessante se em uma turma houver alunos de diferentes cursos e para disciplinas que possuem pré-requisitos diferentes para diferentes currículos.
- Se for para uma turma com alunos de um mesmo curso, isto será um problema."
- 7 Tempo utilizado para responder cada questão, o tempo total da seção, a possibilidade do aluno colocar possíveis dúvidas, a possibilidade de o aluno postar arquivos para auxiliar na resolução de uma pergunta.
- 8 Tempo total da seção, tempo por pergunta, se for uma questão de múltipla escolha saber se ele mudou a resposta e quais foram as escolhas anteriores.
- 9 não vejo no momento, até porque é preciso conhecer o "mundo".
- 10 sim
- 11 "A resposta tradicional seria um plus por nota, certo?"
- A melhor alternativa seria colocar perguntas que instigassem a sua curiosidade em se aprofundar no assunto."
- 12 deve ser levado em consideração o perfil do aluno e do curso a que se destina o jogo.
- 13 Organização de Arquivos

- 1 Feminino
- 2 "links para o material tutoriais"
- 3 Uma mensagem de texto é mais acessível.
- 4 Não entendi a pergunta pois não sei como funciona o programa.
- 5 Seria interessante.
- 6 Acho que deve estar disponível para todos os alunos da disciplina e/ou turma pela qual o professor é responsável.
- 7 Novamente não entendi a pergunta, pois não visualizo o programa.
- 8 A etapa do jogo em que ele está e as habilidades que ele demonstrou apresentar ao responder as questões que o levarão a tal etapa.
- 9 Não sei.
- 10 não

- | | |
|----|----------------------|
| 11 | Através de reforços. |
| 12 | Sim. Psicologia. |

- | | |
|----|---|
| 1 | Masculino |
| 2 | Acredito que links para materiais e tutorias seriam de grande valor.. |
| 3 | Acho que uma mensagem texto é suficiente. |
| 4 | Acho que seria interessante classificar as perguntas por área, nível e caso haja necessidade de algum outro conteúdo. |
| 5 | Acredito que deveria haver uma outra classificação mas não nesse nível. Por exemplo, matemática: algebra, trigonometria, etc. |
| 6 | Não acho necessário mas é uma característica a mais. Mas acredito que o ambiente e o acesso deva ser a todos. |
| 7 | Acho que seria interessante o tempo que ele demorou para responder, se possível, cada pergunta e histórico de acertos. |
| 8 | Se as perguntas forem de múltipla escolha seria interessante armazenar possíveis alterações nas escolhas. Algo do tipo, caso haja uma alteração na resposta que isso fosse armazenado também. |
| 9 | Não |
| 10 | sim |
| 11 | Em um primeiro momento créditos, um ranking, etc. |
| 12 | Para idades menores talvez jogos a níveis de "revistas" infantis. Mas não estou convicto disso.. |
| 13 | Sim |

- | | |
|----|---|
| 1 | Masculino |
| 2 | Sim. |
| 3 | Depende da estrutura do ambiente. |
| 4 | Seria importante que ela pudesse ser relacionada ao perfil do aluno. |
| 5 | "Grau de dificuldade.
As demais classificações dependem de como a resposta poderia ser utilizada." |
| 6 | No contexto do jogo me parece mais adequado que todas as perguntas fossem sendo apresentadas em função da evolução do aluno no ambiente/jogo, e não por ser o aluno A ou B (ambiente estático). |
| 7 | O roteiro percorrido, com as perguntas e respostas. |
| 8 | Tudo que aconteceu no contexto daquela questão. |
| 9 | Sem sugestões. |
| 10 | sim |
| 11 | Conheço pouco do jogo em si para opinar sobre isso. |
| 12 | Conheço pouco do jogo em si para opinar sobre isso. |
| 13 | Sim, de programação e empreendedorismo |

- | | |
|---|--|
| 1 | Feminino |
| 2 | Links diversos, para que o estudo dos conteúdos fique mais dinâmico |
| 3 | Uma msg de texto é mo ideal, já que é mais prática para ser redigida |
| 4 | Não compreendi essa quest~~ao! |
| 5 | Sim, pois isso auxilia o aluno na hora de responder (ele saberá o que está sendo testaodo) |

6	Creio que não
7	Horário e tempo que ele demorou para realizar a atividade (tempo de login)
8	As mesmas citadas acima
9	É interessante sim
10	sim
11	Criando sempre novos desafios
12	Sim
1	Masculino
2	Sim, acho que devem ser indicadas fontes para pesquisa, de vários formatos, dependendo do material disponível.
3	Acho que depende da interface do mundo.
4	Não consegui contextualizar a pergunta.
5	Possivelmente sim, mas não consegui visualizar o uso dessa classificação.
6	Pode ser uma funcionalidade interessante, de maneira que o professor possa controlar a liberação dos desafios.
7	Acho que um acompanhamento dos desafios executados pelo aluno, assim como algum indicador de respostas adequadas. Talvez, um painel de acompanhamento seja mais amigável.
8	Acho que as informações necessárias para o acompanhamento do progresso do aluno, com as quais o professor possa ter um panorama das dificuldades.
9	Não consegui contextualizar a pergunta.
10	sim
11	Acho que uma interface amigável e facilidade de uso é um ponto importante a ser considerado.

1	Feminino
2	Acho que deveria ter links para materiais relacionados, a internet é muito rica em informações, mas tem que ser selecionado um bom material.
3	Sou mais pela mensagem de texto.
4	Contextualizar a pergunta, dar um cenário envolvendo o desafio para motivar.
5	Acho que seria interessante sim. Hoje para fins de formação fala-se em habilidades e competências a serem desenvolvidas, me parece que enriqueceria o ambiente colocar essas informações. Mas não pode ser algo muito extenso.
6	Dependendo do desafio creio que sim, talvez para aqueles que necessitem de alguma discussão intra-equipe para trabalhar a resolução do desafio.
7	O que mais motivou, quais as etapas para realização, quais as principais dificuldades, que recursos poderiam ter ajudado.
8	Não me ocorre outro tipo de informação, considerando que as ações do jogador significam todas as interações com o ambiente do jogo desde o login.
9	Dar opções de customização.
10	sim
11	Se o desafio foi motivador ele vai querer jogar mais, se houver opções durante o desafio ele pode querer jogar de novo para testar outras opções.
12	Além de um conjunto de perguntas? Parece-me que outras opções envolvem utilização de imagens e figuras, animações.

1	Masculino
---	-----------

- 2 acho que sim, se o aluno não conseguir resolver o desafio sem apoio, o ambiente deve fornecer condições para ele conseguir resolvê-lo. O tipo de material depende do conteúdo relacionado ao desafio.
- 3 o uso de áudio pode limitar o uso do ambiente em algumas situações. P.ex., se o usuário compartilha sala com outras pessoas o uso de áudio pode perturbar as pessoas em volta. E não acho que seja mais eficaz do que o uso de texto.
- 4 Informações sobre o conteúdo que está sendo trabalhado com a pergunta
- 5 se fosse um ambiente para o ensino de algoritmos, as perguntas poderiam ser classificadas pela estrutura necessária, repetição, vetores, ou uso de variáveis como contadores, acumuladores, flags, recursão...
- 6 não vejo a necessidade para isso
- 7 as perguntas respondidas
- 8 eu teria que saber mais do ambiente para responder, mas além da resposta do aluno, não sei que outras informações estão envolvidas
- 9 não
- 10 sim

ANEXO E

```

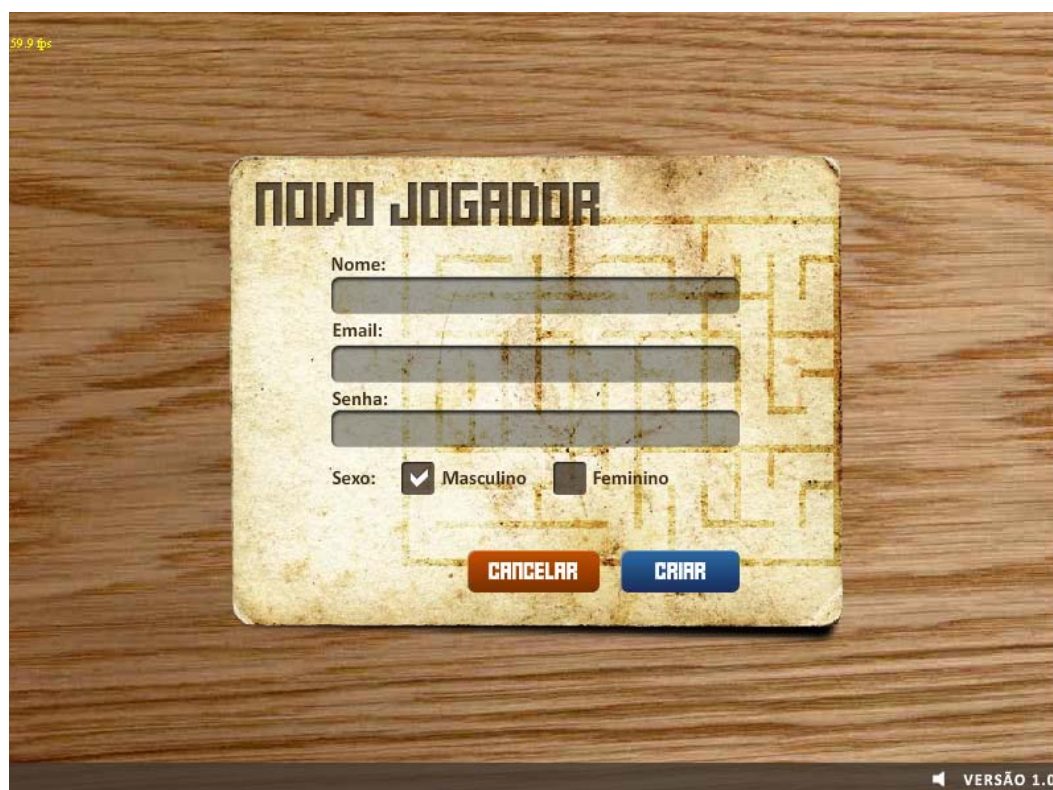
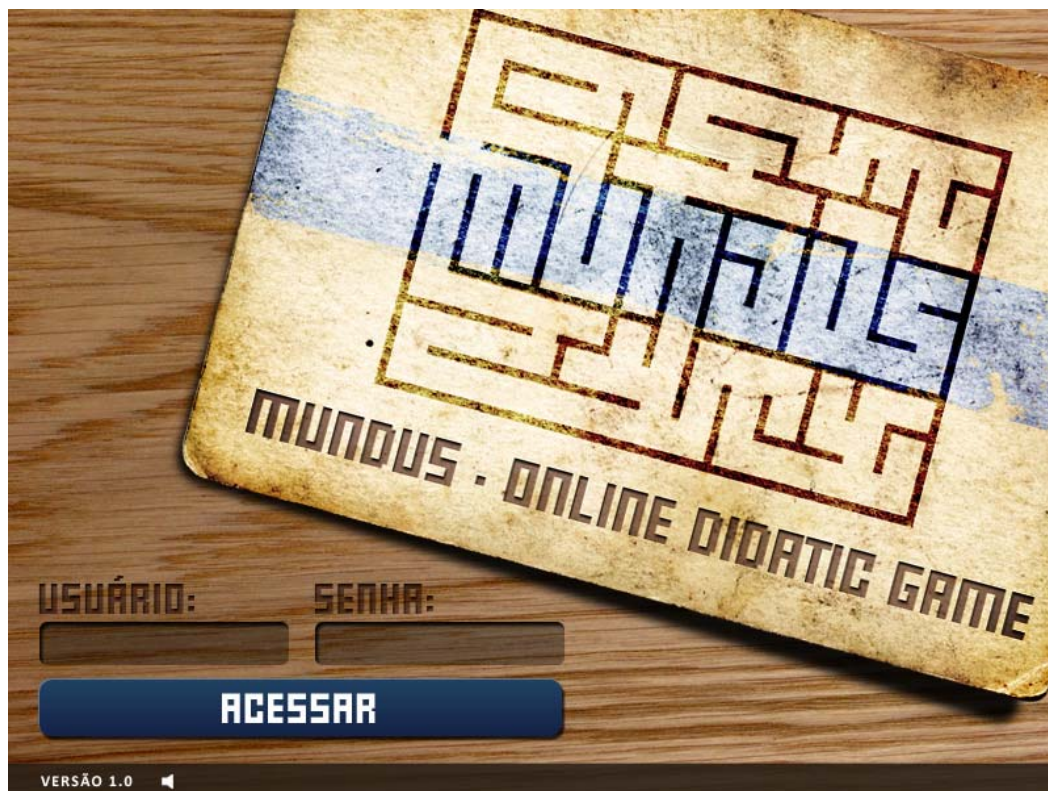
<data>
  <id>1</id>
  <x>0</x>
  <y>0</y>

  <backgrounds>
    <background>
      <type>0</type>
      <x>0</x>
      <y>-15</y>
    </background>
  </backgrounds>
  <tiles>
    <tile>
      <type>1</type>
      <col>0</col>
      <row>0</row>
    </tile>
  </tiles>
  <itens>
    <item>
      <type>10</type>
      <col>4</col>
      <row>6</row>

      <events>
        <event>
          <type>TILE_CLICKED</type>
          <action>game.action.TesteAction</action>
          <params>
            <teste>Teste ehehehehhe</teste>
            <teste2>Teste ehehehehhe</teste2>
          </params>
        </event>
      </events>
    </item>
    <item>
      <type>1</type>
      <col>0</col>
      <row>5</row>
    </item>
    <item>
      <type>4</type>
      <col>4</col>
      <row>4</row>
    </item>
  </itens>
</data>

```

ANEXO F





MATEMÁTICA NÍVEL 3º 4º 5º

ID	DIFICULDADE	TÍTULO	SÉRIE	
2	☆☆☆	LOREM IPSUM	4ª	INFORMAÇÕES JOGAR
3	☆☆☆	LOREM IPSUM TESTE	3ª	INFORMAÇÕES JOGAR
4	☆☆☆	LOREM IPSUM IS SIMPLY DUMMY TEXT OF THE PRINTING	5ª	INFORMAÇÕES JOGAR

MUNDUS . ONLINE DIDATIC GAME | VERSÃO 1.0 |

INFORMAÇÕES

DESAFIO: LOREM IPSUM.
 SESSÃO: MATEMÁTICA
 SÉRIE: 3ª
 DIFICULDADE: ☆☆☆
 GRUPO: 2 JOGADORES
 POSSUE SENHA: NÃO
 DESCRIÇÃO:
 LOREM IPSUM IS SIMPLY DUMMY TEXT OF THE PRINTING AND TYPESETTING INDUSTRY. LOREM IPSUM HAS BEEN THE INDUSTRY'S STANDARD DUMMY TEXT EVER SINCE THE 1500S, WHEN AN UNKNOWN PRINTER TOOK A GALLEY OF TYPE AND SCRAMBLED IT TO MAKE A TYPE SPECIMEN BOOK. IT HAS SURVIVED NOT ONLY FIVE CENTURIES, BUT ALSO THE LEAP INTO ELECTRONIC TYPESETTING, REMAINING ESSENTIALLY UNCHANGED. IT WAS POPULARISED IN THE 1960S WITH THE RELEASE OF LETRASET SHEETS CONTAINING LOREM IPSUM PASSAGES, AND MORE RECENTLY WITH DESKTOP PUBLISHING SOFTWARE

FECHAR

MUNDUS . ONLINE DIDATIC GAME | VERSÃO 1.0 |

