

Série Ensino, Aprendizagem e Tecnologias

# Cognição e Aprendizagem em Mundo Virtual Imersivo

ORGANIZADORES:

Liane Margarida Rockenbach Tarouco

Patrícia Fernanda da Silva

Fabício Herpich

2ª edição

  
**UFRGS**  
EDITORA

 **SEAD**  
**UFRGS**  
EDUCAÇÃO A DISTÂNCIA



UNIVERSIDADE  
FEDERAL DO RIO  
GRANDE DO SUL

---

Reitor

**Rui Vicente Oppermann**

Vice-Reitora e Pró-Reitora  
de Coordenação Acadêmica

**Jane Fraga Tutikian**

---

EDITORA DA UFRGS

Diretor

**Alex Niche Teixeira**

Conselho Editorial

**Álvaro R. C. Merlo**

**Augusto Jaeger Junior**

**Enio Passiani**

**José Rivair Macedo**

**Lia Levy**

**Márcia Ivana de Lima e Silva**

**Naira Maria Balzaretto**

**Paulo César Ribeiro Gomes**

**Rafael Brunhara**

**Tania D. M. Salgado**

**Alex Niche Teixeira, presidente**



Série Ensino, Aprendizagem e Tecnologias

# Cognição e Aprendizagem em Mundo Virtual Imersivo

ORGANIZADORES:

Liane Margarida Rockenbach Tarouco

Patrícia Fernanda da Silva

Fabrício Herpich

2ª edição

  
**UFRGS**  
EDITORA

  
**SEAD**  
**UFRGS**  
EDUCAÇÃO A DISTÂNCIA

© dos autores

1.ª edição: 2019

Direitos reservados desta edição:

Universidade Federal do Rio Grande do Sul

Coordenação da Série:

Laura Wunsch, Cíntia Kulpa, Tanara Forte Furtado e Marcello Ferreira

Coordenação da Editoração: Cíntia Kulpa e Ely Petry

Revisão: Equipe de Revisão da SEAD

Capa: Bruno Assis e Tábata Costa

Editoração eletrônica: Bruno Assis e Tábata Costa

A grafia desta obra foi atualizada conforme o Acordo Ortográfico da Língua Portuguesa, de 1990, que entrou em vigor no Brasil em 1º de janeiro de 2009.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.



---

C659 Cognição e aprendizagem em mundo virtual imersivo [recurso eletrônico] / organizadores Liane Margarida Rockenbach Tarouco, Patrícia Fernanda da Silva [e] Fabrício Herpich ; coordenado pela SEAD/UFRGS. – 2. ed. – Porto Alegre: Editora da UFRGS, 2020.  
355 p. : pdf

(Série Ensino, Aprendizagem e Tecnologias)

Inclui índice remissivo.

1. Educação. 2. Ensino e aprendizagem. 3. Laboratórios virtuais. 4. Ambientes imersivos. 5. Cognição I. Tarouco, Liane Margarida Rockenbach. II. Silva, Patrícia Fernanda da. III. Herpich, Fabrício. IV. Universidade Federal do Rio Grande do Sul. Secretaria de Educação a Distância. V. Série.

CDU 37:681.3

---

CIP-Brasil. Dados Internacionais de Catalogação na Publicação.  
(Jaqueline Trombin – Bibliotecária responsável CRB10/979)

ISBN 978-65-5725-001-3

# Sumário

LISTA DE ABREVIATURAS E SIGLAS	11
<b>Apresentação</b>	<b>13</b>
<b>1 Introdução aos Mundos Virtuais Imersivos</b>	<b>17</b>
1.1 MUNDOS VIRTUAIS IMERSIVOS	18
1.1.1 Histórico e evolução dos Mundos Virtuais Imersivos	22
1.1.2 Mundos Virtuais Imersivos na educação	26
1.1.3 Possibilidades e vantagens dos laboratórios virtuais imersivos para o ensino de Ciências	38
1.1.4 Diferenças entre real e virtual no processo de aprendizagem	45
1.1.5 Motivação e MVI	58
REFERÊNCIAS	61
<b>2 As ferramentas de autoria para criar um Mundo Virtual Imersivo</b>	<b>69</b>
2.1 CRIAÇÃO DO MUNDO VIRTUAL	70

2.1.1 Versão <i>StandAlone</i>	72
2.1.2 Versão Servidor	79
2.2 CUSTOMIZAÇÃO DO AVATAR	103
2.3 AS FERRAMENTAS DE AUTORIA	107
2.3.1 Importação de regiões prontas obtidas em repositórios	107
2.4 AGREGAR COMPORTAMENTOS AOS OBJETOS ( <i>SCRIPTS</i> )	111
2.5 NPC - <i>Non-Player Character</i>	118
2.5.1 Conexão do NPC no <i>OpenSim</i> com o ambiente <i>Pandorabots</i> ao <i>OpenSim</i>	120
2.5.2 Conexão do NPC no <i>OpenSim</i> com o agente conversacional ATENA	126
2.5.3 Criação de ações e gestos	133
REFERÊNCIAS	136
<b>3 Agentes conversacionais no Mundo Virtual Imersivo</b>	<b>139</b>
3.1 AGENTE CONVERSACIONAL METIS	145
3.2 O FastAIML	149
3.3 AGATA	153

3.4 APRENDIZAGEM COLABORATIVA COM ENTIDADES VIRTUAIS	158
3.5 AMBIENTE, PÚBLICO E ANÁLISE DA REAÇÃO DE USUÁRIOS AO USO DO METIS	162
3.5.1 Ferramentas e coleta dos dados	164
3.5.2 Análise e discussão de resultados	168
3.6 CONSIDERAÇÕES FINAIS	171
REFERÊNCIAS	175
<b>4 Enriquecendo o ambiente com recursos multimídia</b>	<b>179</b>
4.1 CONSTRUINDO RECURSOS MULTIMÍDIA NO AMBIENTE DO MUNDO VIRTUAL IMERSIVO	180
4.2 ENRIQUECENDO O AMBIENTE COM RECURSOS MULTIMÍDIA	184
4.3 INCORPORANDO E USANDO MULTIMÍDIA NO MUNDO VIRTUAL IMERSIVO	185
4.4 MULTIMÍDIA INTERATIVA COMO RECURSO DE COGNIÇÃO E APRENDIZAGEM NO MUNDO VIRTUAL IMERSIVO	199
REFERÊNCIAS	204
<b>5 Rastreamento em Mundos Virtuais Imersivos</b>	<b>207</b>

5.1 USO E CONFIGURAÇÃO DE SENSORES EM MVI	208
5.2 COLETA DE DADOS SOBRE O USO DOS RECURSOS MULTIMÍDIA DO MVI	216
5.2.1 Monitoramento das ações do avatar do estudante	217
5.2.2 Estrutura dos Servidores <i>OpenSim</i> , WWW e de banco de dados	222
5.2.3 Estrutura das tabelas de dados	226
5.3 ANÁLISE DOS DADOS	231
REFERÊNCIAS	235
<b>6 Estratégias pedagógicas para usar o MVI</b>	<b>239</b>
6.1 EXPERIÊNCIA DO ALUNO NO MVI	240
6.1.1 Experiência do aluno no MVI e a orientação de agentes virtuais	246
6.2 VERIFICAÇÃO DAS TRAJETÓRIAS DO ALUNO OU IDENTIFICAÇÃO DO PERCURSO EDUCACIONAL DO ALUNO	250
6.2.1 Verificação de aprendizagem	250
6.3 ESTRATÉGIA PEDAGÓGICA ENVOLVENDO A APRENDIZAGEM EXPERIENCIAL	257
6.3.1 Experiências em MVI	258
6.4 MOTIVAÇÃO DOS ALUNOS EM MVI	261



<b>ANEXOS</b>	<b>333</b>
ANEXO 1 - VARIÁVEIS	333
ANEXO 2 - OPERADORES	334
ANEXO 3 - COMANDOS LSL	335
<b>INFORMAÇÕES SOBRE OS AUTORES</b>	<b>347</b>
<b>APOIO</b>	<b>352</b>
<b>ÍNDICE REMISSIVO</b>	<b>353</b>



## LISTA DE ABREVIATURAS E SIGLAS

AGATA	<i>Automatic Generation of AIML from Text Acquisition</i>
AIML	<i>Artificial Intelligence Markup Language</i>
ALICE	<i>Artificial Linguistic Internet Computer Entity</i>
API	<i>Application Programming Interface</i>
ATENA	Agente Tutor para Ensino e Navegação no Ambiente
AVATAR	Ambiente Virtual de Aprendizagem e Trabalho Acadêmico Remoto
CSS	<i>Cascading Style Sheets</i>
FAQ	<i>Frequently Asked Questions</i>
HIGIA	Habitat Individual e Guia Interativo de Atitudes
HTTP	<i>Hypertext Transfer Protocol</i> (em português Protocolo de Transferência de Hipertexto)
LSL	<i>Linden Scripting Language</i>
METIS	Mediadora de Educação em Tecnologia Informática e Socializadora
MVI	Mundo Virtual Imersivo
NPC	<i>Non-Player Character</i>
OSSL	<i>OpenSim Scripting Language</i>
PHP	<i>Hypertext Preprocessor</i>
PPGIE	Programa de Pós-Graduação em Informática na Educação
TIC	Tecnologia de Informação e Comunicação
UFRGS	Universidade Federal do Rio Grande do Sul

XML	<i>eXtensible Markup Language</i>
ZDP	Zona de Desenvolvimento Proximal

# Apresentação

O livro *Cognição e Aprendizagem em Mundo Virtual Imersivo*<sup>1</sup> teve como meta apresentar e discutir os princípios e teorias que dão suporte ao modo de aprender neste contexto. Para que essa aprendizagem possa ocorrer, é preciso, naturalmente, que seja criado um Mundo Virtual Imersivo capaz de ensinar atividades e de levar à cognição e aprendizagem. O objetivo a ser atingido demanda uma aprendizagem significativa que seja capaz de ensinar a formação do pensamento complexo, o que implica exercitar o uso do pensamento de alto nível.

Com base na proposta de David Kolb, a teoria da aprendizagem experiencial (ELT – Experiencial Learning Theory) que inclui o Ciclo de Kolb, conforme

---

<sup>1</sup> Usaremos a denominação Mundo Virtual Imersivo ou Mundos Virtuais Imersivos para designar o ambiente imersivo 3D proporcionado pela plataforma *OpenSim*, na qual o projeto AVATAR foi desenvolvido e que é objeto de apresentação e discussão neste livro.

descrito no Capítulo 1 deste livro, contempla o processo de aprendizagem que deve envolver basicamente a experiencição seguida de observação reflexiva, para, então, acontecer a formação de conceitos abstratos relacionados ao contexto da experiencição e, finalmente, a transferência para outro contexto.

O material reunido neste livro discute teoricamente a relação da cognição e da aprendizagem com os aspectos tecnológicos envolvidos na criação de um Mundo Virtual Imersivo. Com isso, espera-se oferecer suporte e permitir a realização de experiências.

O ambiente selecionado para o desenvolvimento dos experimentos virtuais foi a plataforma *Open Simulator (OpenSim)*. O livro apresenta e exemplifica como deve ser feita a configuração do Mundo Virtual Imersivo, ferramenta de apoio tanto para a capacitação de docentes na criação de experimentos virtuais quanto para a formação dos próprios alunos, que poderão, também, planejar atividades e experimentos virtuais nesse ambiente. Ainda são discutidos neste livro conceitos e estratégias para aumentar a motivação dos estudantes que irão desenvolver atividades de aprendizagem nesse contexto.

Contribuíram para a elaboração do livro diversos autores do grupo de pesquisas do Programa de Pós-Graduação em Informática na Educação da UFRGS, os quais têm em comum o fato de terem pesquisado e efetivamente trabalhado com o Mundo Virtual Imersivo, baseado no *OpenSim*, e nele terem desenvolvido experimentos virtuais testados com estudantes de Ensino Médio em algumas escolas do estado.

O Capítulo 1 aborda os conceitos básicos sobre Mundos Virtuais Imersivos, seu surgimento, e a plataforma *OpenSim*, usada para o desenvolvimento dos laboratórios virtuais construídos pela equipe para o ensino de Ciências. Os aspectos pedagógicos inerentes à cognição e à aprendizagem nesse contexto também são abordados nesse Capítulo, o qual foi resultado da colaboração de Fabiana Santiago Sgobbi, Fabrício Herpich, Felipe Becker Nunes, Liane Margarida Rockenbach Tarouco e Patrícia Fernanda da Silva.

O Capítulo 2 inicia a descrição do processo de criação de Mundos Virtuais Imersivos. São apresentadas ferramentas de autoria para criar um Mundo Virtual Imersivo, desde a customização do avatar que representa o usuário (aspectos físicos, vestimenta etc...) até a configuração do Mundo Virtual e a inclusão de objetos no cenário, além de associar comportamentos aos objetos. Um aspecto relevante do Mundo Virtual é o elemento avatar não controlado por usuário. Esse avatar, denominado NPC (*Non-Player Character*), no contexto do *OpenSim*, faz o papel de agente pedagógico, acompanhando o usuário ao longo de suas atividades no Mundo Virtual, apresentando sugestões e orientações e, ainda, respondendo a perguntas eventualmente formuladas pelo usuário relacionadas com o campo contextual envolvido no experimento. Esse Capítulo foi resultado da colaboração de Fabiana Santiago Sgobbi, Fabrício Herpich e Felipe Becker Nunes.

O Capítulo 3 apresenta o processo de criação e construção de um *chatbot* que, interligado ao agente pedagógico, enseja o diálogo com o usuário, simulando a participação de um tutor humano. São autores desse Capítulo: Clóvis da Silveira, Anita Raquel da Silva e Liane Margarida Rockenbach Tarouco.

O Capítulo 4 mostra e discute o uso de diversos recursos de multimídia no Ambiente Virtual Imersivo baseado no *OpenSim*. O uso de imagens estáticas ou animadas, vídeos e recursos interativos (como testes e outros tipos de exercícios) é descrito e exemplificado. Esse Capítulo foi desenvolvido por Liane Margarida Rockenbach Tarouco.

O Capítulo 5 explica como rastrear as atividades dos usuários, no Mundo Virtual Imersivo, por meio do uso de sensores que capturam automaticamente a passagem do usuário por determinados locais ou sua ação sobre objetos contidos no cenário. Os autores desse Capítulo são Felipe Becker Nunes e Leandro Rosniak Tibola.

O Capítulo 6 discute estratégias pedagógicas para o uso de Mundos Virtuais Imersivos que busquem melhorar a experiência do aluno no ambiente. São autores desse Capítulo: Bárbara Gorziza Avila, Fabiana Santiago Sgobbi, Felipe Becker Nunes, Liane Margarida Rockenbach Tarouco e Leandro Rosniak Tibola.

O Capítulo 7 mostra como construir roteiros associados aos objetos incluídos no Mundo Virtual. Esses roteiros são incorporados aos objetos e permitem tornar estes objetos interativos e capazes de reagir a ações dos usuários (cliques) sobre eles, exibindo comportamentos que permitem simular experiências realizadas em um laboratório de ciências. Participaram da autoria desse Capítulo: Bárbara Gorziza Avila, Érico Amaral, Fabiana Santiago Sgobbi, Fabrício Herpich, Liane Margarida Rockenbach Tarouco e Leandro Rosniak Tibola.

# 1

## Introdução aos Mundos Virtuais Imersivos

**Fabiana Santiago Sgobbi - PPGIE - UFRGS**  
fabianasgobbi@gmail.com

**Fabrcio Herpich - PPGIE - UFRGS**  
fabricio\_herpich@hotmail.com

**Felipe Becker Nunes - AMF**  
nunesfb@gmail.com

**Liane Margarida Rockenbach Tarouco**  
**PPGIE - UFRGS - liane@penta.ufrgs.br**

**Patrcia Fernanda da Silva - PPGIE - UFRGS**  
patriciasilvaufrgs@gmail.com

## 1.1 MUNDOS VIRTUAIS IMERSIVOS

No decorrer dos últimos anos, uma expansão significativa pôde ser constatada acerca do uso das TICs no contexto educacional do país, assegurando novas possibilidades de aplicação dos recursos computacionais como elementos de apoio e motivação aos processos de ensino e aprendizagem. Ao oferecer aos professores novas alternativas a serem exploradas, o uso dos Ambientes Virtuais de Aprendizagem, jogos educacionais, recursos de Realidade Aumentada e aplicativos para dispositivos móveis tem auxiliado a modificar a forma de ensino e aprendizagem. A implementação dos Mundos Virtuais pode ser considerada uma alternativa a ser empreendida nesse contexto, envolvendo a modalidade semipresencial, com a realização de atividades complementares no período inverso ao presencial ou em atividades extras nos laboratórios de informática da instituição de ensino. As características presentes nesse ambiente, como imersão, colaboração, comunicação e interação, podem criar novas possibilidades, em que os estudantes, no momento da realização das atividades educacionais, tornam-se mais ativos e exploram novas oportunidades de aprendizado no Mundo Virtual.

Tais constatações são corroboradas por Fernández-Gallego *et al.* (2013) e Rafalski *et al.* (2014), ao entenderem que os estudantes passam de meros espectadores para protagonistas do processo de aprendizagem de forma mais autônoma. Em complemento a essa asserção, Chow (2016) explica que, em um ambiente virtual 3D, os estudantes estão livres para explorá-lo, de modo que a aprendizagem se caracteriza por ser mais ativa e participativa, em vez de estar centralizada em ouvir e absorver informações. Nesse contexto, torna-se essencial destacar,



inicialmente, ao leitor desta obra que não existe consenso acerca da nomenclatura que se faz presente entre os pesquisadores sobre o uso desse tipo de ambiente. No meio acadêmico, podem ser encontradas diferentes maneiras de referir-se a esses ambientes, como pode ser visto em detalhes a seguir:

a) *3D Virtual Worlds (3DVW)* ou *Mundos Virtuais 3D* – Antonio (2016) e Xenos *et al.* (2017).

b) *Digital Virtual World 3D (DVW3D)* ou *Mundos Digitais Virtuais em 3D* – Moretti e Schlemmer (2012) e Reinhard (2012).

c) *Metaverse* ou *Metaverso* – Griol *et al.* (2014) e Amaral (2015).

d) *Multi-User Virtual Environment (MUVE)* ou *Ambientes Virtuais* – Reinhard (2012) e Khan e Safaan (2017).

e) *Persistent Online World* ou *Ambientes On-line Persistentes* – Ivory (2012).

f) *Three-dimensional Collaborative Virtual Environments (3D CVE)* ou *Ambientes Virtuais Tridimensionais Colaborativos* – Schmeil (2012) e Poppe *et al.* (2017).

Esses são alguns termos cunhados por diferentes pesquisadores que têm trabalhado com esse tipo de abordagem. Dessa forma, levando em consideração a expertise dos autores desta obra e com base nos experimentos realizados e em levantamentos de literatura conduzidos nos últimos anos, foi estabelecida a seguinte definição formal para este livro: *Mundos Virtuais Imersivos*. Essa dificuldade de unificação de um termo pode ser estendida ao processo de definição formal de um conceito referente aos *Mundos Virtuais*. Diferentes definições têm sido atribuídas a esses ambientes, conforme apresentado a seguir.

Soto (2013) fornece uma definição mais ampla, na qual os Mundos Virtuais são simulações computadorizadas que oferecem um espaço gráfico tridimensional. Esse representa um ambiente físico em que os usuários podem interagir entre si e manipular o ambiente com a criação e modificação de objetos. Griol *et al.* (2014) têm o entendimento de que os Mundos Virtuais podem ser considerados ambientes gráficos simulados por computador no qual os seres humanos convivem com outros usuários por meio de seus avatares.

Em uma definição complementar, Xenos *et al.* (2017) consideram os Mundos Virtuais ambientes *on-line* gráficos e Interativos Tridimensionais Imersivos, que podem ser uma réplica de um lugar físico existente ou um lugar imaginário, ou mesmo lugares que são impossíveis de visitar na vida real pelas restrições, como o alto custo e/ou questões de segurança. Descritas as definições gerais de Mundos Virtuais, uma análise dos recursos presentes nesse tipo de ambiente torna-se necessária para fornecer uma visão mais ampla da engenharia de funcionamento e dos elementos que o compõem.

Normalmente os usuários podem navegar em todo o cenário disposto no ambiente, interagir com objetos (tocar, guardar, empurrar itens etc.) ou conversar com outros usuários do Mundo Virtual (RICO *et al.*, 2017). Conforme descrito por Allison *et al.* (2010), os usuários criam suas representações virtuais por meio de avatares, que possuem um inventário pessoal associado a eles, no qual podem estar contidos objetos como vestimentas, carros, prédios, entre outros tipos de elementos.

O processo de comunicação nesse tipo de ambiente é constituído por um ou mais usuários conectados em um espaço virtual tridimensional, sendo possível trocar informações pelo canal de *chat* disponibilizado no ambiente (SCHETTINO, 2015). Os avatares podem se movimentar de diferentes formas, seja caminhando, correndo, voando ou se teletransportando de uma região para outra no Mundo Virtual.

Voss (2014) explica o significado do termo região, que pode ser criada, no formato de pequeno terreno (256m x 256m), e estar tanto interligada quanto separada geograficamente no Mundo Virtual. Dessa forma, a distribuição desses terrenos obedece a uma matriz bidimensional (X e Y), para que possam ser distribuídos sem que ocorram conflitos de posicionamento. Os Mundos Virtuais são baseados em um sistema de coordenadas cartesianas, contendo três eixos: Eixo X, Eixo Y e Eixo Z. Cada região no Mundo Virtual tem um conjunto único de coordenadas cartesianas.

Nelson e Erlandson (2012) esclarecem que cada objeto está localizado por meio desses três eixos de coordenadas, visto que nenhum objeto ou avatar pode ser inserido ou se movimentar para um local que não possua uma coordenada previamente especificada, por exemplo, em uma região inexistente no Mundo Virtual. Portanto, todos os objetos devem estar posicionados dentro das limitações de uma região e sempre irão conter uma posição geográfica definida pelos três eixos cartesianos.

Para a visualização e interação com o Mundo Virtual, é necessária a instalação de uma aplicação conhecida como *viewer*, na qual o usuário poderá interagir e realizar suas atividades. Nunes *et al.* (2013) expõem que esta aplicação fornece suporte para a visualização, importação e exportação de objetos e demais recursos relacionados aos Mundos Virtuais.

Segundo os autores, a escolha dessa ferramenta é um fator determinante, pois deve estar de acordo com aquilo que o usuário pretende exercer dentro do ambiente, visto que a escolha errada pode impedir, limitar ou dificultar a realização das atividades propostas. Como exemplos de *viewers*, temos o *Singularity*, *Firestorm* e *Imprudence*. Outros detalhes sobre os *viewers*, seus recursos e melhores opções para a escolha de utilização podem ser visualizados nas pesquisas executadas por Nunes *et al.* (2013), Voss *et al.* (2013) e Falcade *et al.* (2014).

### **1.1.1 Histórico e evolução dos Mundos Virtuais Imersivos**

Bainbridge (2010) entende que os Mundos Virtuais são ambientes *on-line* persistentes (ou seja, continuam existindo mesmo depois que os usuários saem dele) e que as mudanças realizadas por eles são, de certa forma, permanentes. Neles, as pessoas podem interagir, seja para o trabalho ou lazer, de forma comparável ao mundo real (BAINBRIDGE, 2010, p. 1). Como exemplo de Mundos Virtuais que têm sido utilizados por diferentes pesquisadores estão *Active Worlds*, *Open Simulator (Open-Sim)*, *Second Life (SL)*, *Sansar* e *Open Wonderland*.

*Active Worlds* pode ser considerado um dos pioneiros na exploração dos Mundos Virtuais. Sua origem data de meados de 1995, mantendo uma estrutura robusta e com um grande número de usuários até hoje. Conforme Kotsilieris e Dimopoulou (2013), os usuários podem acessar o ambiente com um nome único para seu avatar, por meio de um navegador (*browser*), interagindo em mundos 3D que foram construídos pelos demais usuários. No contexto educacional, um exemplo de espaço virtual criado é o *Active Worlds Education Universe (AWEDU)*, cujo foco está no fortalecimento e desenvolvimento de atividades utilizando diferentes teorias educacionais (KOTSILIERIS; DIMOPOULOU, 2013). Apesar dos pontos positivos apresentados, a ausência de um controle maior pelo desenvolvedor, personalização dos recursos do ambiente e ausência de código aberto acabam por estreitar suas vantagens no que concerne à sua aplicação no âmbito educacional.

Já o *Second Life* é um Mundo Virtual 3D *on-line* de licença proprietária que conta com uma base extensa de usuários, tendo como características a escalabilidade de operação, troca de mensagens e manipulação das regiões e objetos. Esse ambiente teve o seu início, em 2007, com o objetivo de proporcionar uma plataforma flexível e modular a criação de ambientes virtuais personalizados (GOMES, 2016). Conforme Konstantinidis (2010), o *Second Life* pode ser considerado um ambiente ideal à aprendizagem experiencial, visto que, embora não tenha sido criado com propósitos educacionais, foi adotado por diversas instituições de ensino como ferramenta de suporte para a realização de atividades educacionais, principalmente a distância. Pelo fato de múlti-

plos recursos serem privados e pagos, a sua utilização no meio educacional perdeu intensidade e mercado para soluções gratuitas, como o *Open Wonderland* e *OpenSim*.

A mesma empresa criadora do *Second Life* (*Linden Research*) tem avançado significativamente na construção de uma nova plataforma geradora de Mundos Virtuais, conhecida como *Sansar*. Essa nova alternativa surge como uma evolução de todas as plataformas criadas anteriormente, com resolução gráfica em alta definição, maior suporte à criação de *scripts* na linguagem C# e ao uso de recursos de realidade aumentada com óculos especiais, em que o ambiente modelado é apresentado, e as ações do usuário, que está utilizando os óculos no espaço real, são replicadas para dentro do Mundo Virtual.

Com relação ao *Open Wonderland*, ele é constituído por um conjunto de ferramentas de código aberto, para a criação de Mundos Virtuais 3D colaborativos, em que os usuários podem se comunicar, utilizando o canal do *chat*, e interagir com aplicações compartilhadas e cooperativas em áreas como a educacional, social ou de negócios (KOTSILIERIS; DIMOPOULOU, 2013). É um *software* em desenvolvimento ativo que surgiu como uma alternativa aos Mundos Virtuais *OpenSim* e *Second Life*, que já estão plenamente estabelecidos no meio acadêmico.

Quanto ao *OpenSim*, ele é uma aplicação *open source*, multiplataforma e com acesso compartilhado dos usuários que permite aos indivíduos e empresas do mundo todo personalizarem os seus Mundos Virtuais baseados em suas preferências de tecnologia (KICKMEIER-RUST; ALBERT, 2013). Tal como o *Second Life*, o *OpenSim* possui diversas funcionalidades, como a comunicação entre usuários, modelagem

gráfica de objetos 3D e possibilidade de interação com o ambiente virtual (GOMES; FIGUEIREDO, 2014). Na Tabela 1.1, elaborada por Gomes (2016) em seu trabalho de Tese, especificam-se algumas das principais diferenças e similaridades existentes entre esses dois ambientes.

Tabela 1.1 - Comparativo de características entre *Second Life* e *OpenSim*.

Características	Mundos Virtuais	
	<i>Second Life</i>	<i>OpenSim</i>
Licença	Proprietária	Gratuita
Linguagem de Programação do Servidor	C#	C#
Linguagem de programação de <i>Scripts</i> Suportadas	LSL	LSL; OSSL; C#; Java Script e VB.NET
Criação de Objetos	Limitado	Livre
Ocupação de Terrenos	Limitado	Livre
<i>Chats</i> e Mensagens	Sim	Sim
Audioconferência	Sim	Sim

Fonte: Gomes (2016).

Apesar de existirem algumas características similares entre os ambientes, torna-se importante ressaltar as diferenças existentes entre eles. O *OpenSim* apresenta maior diversidade de opções e quantidade de recursos gratuitos, sendo possível destacar o fato de o código do ambiente ser aberto e passível a modificações, além da criação ilimitada de regiões, objetos 3D e suporte a um número maior de linguagens de programação na criação de *scripts*.

Quanto ao *Open Wonderland*, ele apresenta alguns diferenciais, como interação com planilhas e documentos dentro do ambiente, provendo maior grau de imersão aos usuários. Porém, o nível de detalhamento gráfico e robustez ainda necessita ser aprimorado, quando comparado com os ambientes *OpenSim* e *Second Life*.

Com relação ao *Sansar*, pelo fato de o ambiente estar em processo de desenvolvimento, não se torna adequado identificar vantagens e desvantagens de forma consistente no presente momento.

Banerjee *et al.* (2013) destacam que o *OpenSim* pode ser considerado uma ferramenta melhor para o desenvolvimento de conteúdos educacionais, uma vez que está disponível gratuitamente e todos os arquivos podem ser recuperados no formato “OAR”, podendo ser compartilhados e utilizados pelo usuário em qualquer espaço dentro deste escopo.

### **1.1.2 Mundos Virtuais Imersivos na educação**

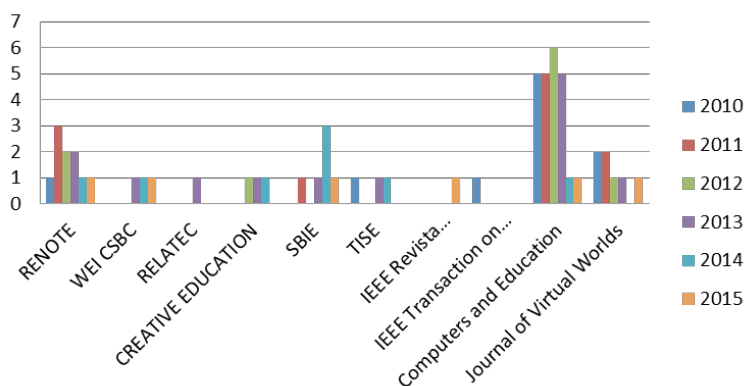
No que concerne à aplicação dos Mundos Virtuais no âmbito educacional, esses espaços tridimensionais são representações do mundo real, capazes de impulsionar experiências diversificadas aos usuários, pois fazem com que o estudante fique imerso no ambiente, estimulando os diferentes tipos de aquisição de conhecimento (NUNES *et al.*, 2016). Também se consideram tecnologias de baixo custo que podem representar salas de aulas, laboratórios, simuladores, jogos, permitindo um maior envolvimento do usuário com conteúdos que necessitam de reflexão por análise de uma representação visual para melhor compreensão (OLIVEIRA, 2015).



No cenário mundial, os Mundos Virtuais Imersivos também apresentam desenvolvimentos importantes na educação. Nessa perspectiva, ao longo desta seção, serão discutidos os resultados de uma revisão sistemática da literatura sobre o uso educacional de Mundos Virtuais Imersivos no Brasil e no exterior, realizada por Nunes *et al.* (2016), a fim de evidenciar as particularidades intrínsecas a essas ferramentas educacionais.

Foram encontrados 58 artigos científicos. Para demonstrar a difusão dos Mundos Virtuais Imersivos, a Figura 1.1 demonstra que a maior quantidade de artigos científicos sobre o uso de Mundos Virtuais Imersivos na educação concentrou-se no ano de 2013, representando 23% do total de artigos que foram descobertos.

Figura 1.1 - Distribuição das publicações por ano e fonte de busca.



Fonte: Nunes *et al.* (2016).

O ano que obteve a menor quantidade foi 2015, representando apenas 10% dos artigos, e 60% deles foram socializados em bases de dados brasileiras. Observa-se que o número de publicações diminuiu nos últimos anos e teve seu pico entre os anos de 2010 e 2013, possivel-

mente pela expansão do uso dos Mundos Virtuais por meio do *Second Life* e pelo surgimento de soluções para a construção de Mundos Virtuais gratuitos (e.g. *OpenSim*, *Open Wonderland*).

Na Figura 1.1, também se percebe que 40% dos artigos foram publicados na revista *Computers and Education*. Esse *journal* juntamente com a RENOTE obteve publicações durante todos os anos considerados nesta revisão sistemática, característica que não está presente nas demais opções existentes. O SBIE foi a única base de dados que ultrapassou o número de publicações do *Computers and Education*, fato ocorrido unicamente em 2014. O WEI manteve seu número durante os três últimos anos, e as seguintes bases de dados não obtiveram números durante o ano de 2015: *RELATEC*, *Creative Education*, *TISE* e *IEEE Transaction on Learning Technologies*.

No que diz respeito ao levantamento dos tipos de Mundos Virtuais que vêm sendo utilizados no decorrer dos últimos cinco anos, a análise dos resultados demonstrou um cenário logicamente esperado pelos autores deste trabalho. Tal asserção se explica pelo fato de o Mundo Virtual *Second Life* ter sido o mais utilizado (30 artigos dos 58), visto que, dentre as opções existentes atualmente, tal aplicação pode ser considerada a mais consolidada e difundida, tanto no meio acadêmico quanto no profissional e pessoal.

Os demais Mundos Virtuais foram o *OpenSim* (17), *Open Wonderland* (4) e um pequeno grupo de diferentes *softwares* (6) que somente foram utilizados em um artigo cada um. É importante ressaltar que, em um artigo, não foi citado o tipo de Mundo Virtual utilizado, visto que o

*OpenSim* se trata de uma aplicação ainda em desenvolvimento e oriunda do próprio *Second Life*. O menor número se justifica e, apesar dessas particularidades, ele não deixa de estar concorrendo fortemente com o SL.

Pensando a partir do ponto de vista temporal, a distribuição apresentada pela análise mostra uma predominância do uso do *Second Life* no intervalo dos anos de 2010 a 2013, dado que se trata de um período em que a aplicação estava em destaque no meio acadêmico. Após esse intervalo, a análise mostra que o número de usos começou a diminuir, de modo concomitante à expansão do *OpenSim*, assim como do *Open Wonderland*, que tomaram espaço no meio acadêmico. Atualmente, foi possível perceber uma distribuição mais equilibrada entre SL e *OpenSim* com o *Open Wonderland*, e demais aplicações menores tendo uma pequena parcela de utilização.

Nesse contexto, conforme explicitado na Seção 1 deste trabalho, os Mundos Virtuais são visualizados por meio de uma aplicação denominada *viewer*. O levantamento realizado demonstrou que 46 artigos dos 58 não apresentaram de forma explícita o nome do visualizador utilizado. Dentro desse escopo, a maioria dos artigos utilizou como Mundo Virtual o *Second Life*, que possui uma *viewer* oficial, o que pode ser considerado um dos motivos para não haver nenhuma menção à aplicação nesses artigos, visto que para os autores, possivelmente, não é necessário mencionar o tipo de *viewer* utilizado.

Quanto aos artigos que citaram o tipo de *viewer* empregado, o que obteve o maior número de menções foi o *Firestorm* (6), seguido por *Singularity* (2), *Imprudence* (2) e *ActiveWorlds 3D Browser* (1). Também foram citados em um artigo os visualizados para dispositivos móveis *Lumyia*

e *Pixie Viewer*. Tais percentuais demonstram uma predominância do *Firestorm*, visto que, conforme dados da pesquisa realizada por Voss et al. (2013), essa aplicação possui diversos recursos necessários à manipulação de objetos 3D no Mundo Virtual, como a possibilidade de importação de arquivos no formato XML e DAE, criação de objetos 3D, importação de arquivos multimídia e suporte à criação de *scripts*.

Com relação aos diferentes tipos de artefatos e recursos que podem ser empregados nos mundos virtuais, é importante citar os *scripts* de programação, na linguagem *OpenSim Script Language* (OSSL), nativa do ambiente, que podem ser utilizados em consonância com os objetos 3D. A análise efetuada mostrou que, em 54 artigos de um total de 58, usaram-se tais recursos. Efetuando uma interconexão com as características presentes nesse tipo de ambiente, a elevada utilização dos objetos 3D com *scripts* se justifica pelo fato de ser um ambiente interativo e imersivo, no qual a interatividade e a autoria dos usuários devem ser exploradas constantemente.

Dentre os demais recursos utilizados, 12 artigos também acrescentaram materiais multimídia, como vídeos, *slides*, ferramentas externas, textos e questões. Isso demonstra o foco educacional presente nas pesquisas realizadas, já que tais tipos de recursos foram inseridos como suporte para o processo de aprendizagem dos estudantes. Torna-se importante ressaltar que, embora o número não seja elevado em comparação com o total de artigos (56), ele pode ser considerado positivo, pois a inserção de tais recursos envolve conhecimentos mais avançados nesse tipo de ambiente, o que aumenta a complexidade de programação do ambiente para usuários que não estão ambientados.

Também foi observada a utilização do *software* Scratch em quatro artigos. Esse *software* é constituído por uma ferramenta para auxiliar na programação de *scripts* para os objetos 3D do Mundo Virtual. O foco do uso dessa ferramenta esteve voltado a experiências envolvendo programação com alunos, uma vez que ela pode ser considerada uma alternativa válida para auxiliar no processo de aprendizagem desse tópico.

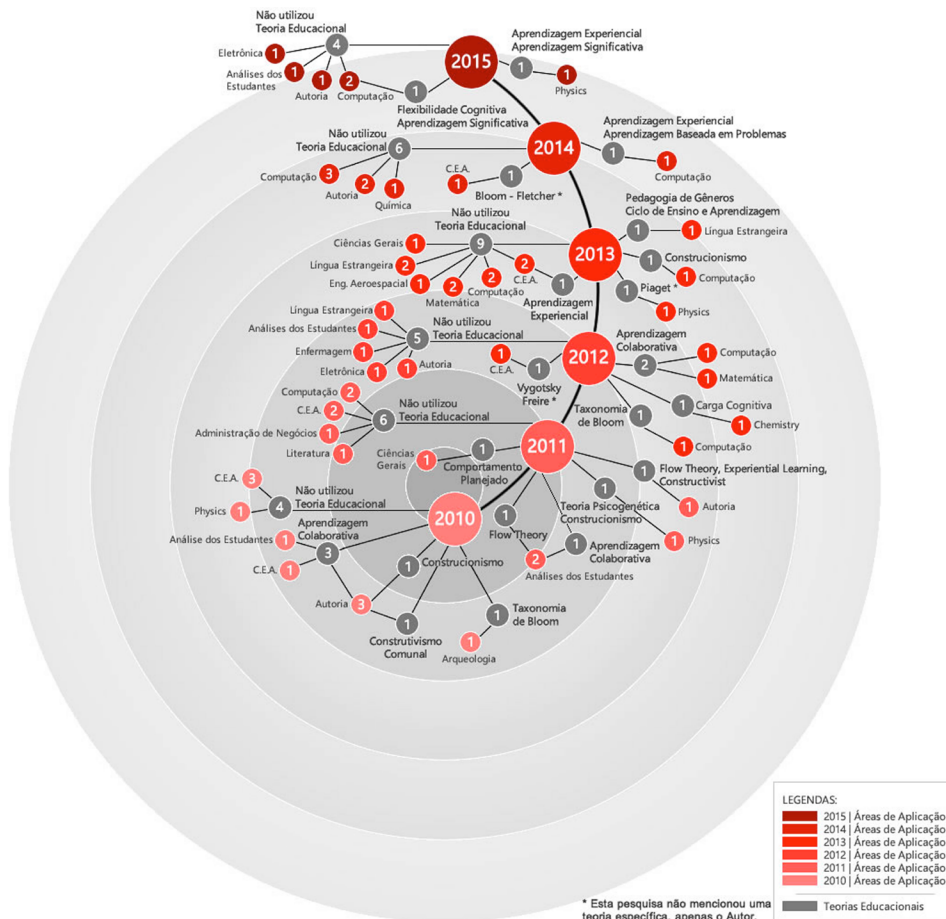
Os agentes *Non-Player Characters* (NPC), que são entidades programáveis por meio da linguagem OSSL para executar ações previamente definidas, foram referidos em cinco artigos. Seu objetivo do ponto de vista educacional esteve centrado no fato de agregar a esses agentes um maior nível de interação e conversação com os usuários, para que eles possam ser utilizados como agentes tutores, instruindo o estudante durante a realização das atividades educacionais no Mundo Virtual. O menor número de artigos que utilizou NPCs também se deve ao fato citado anteriormente, de que o trabalho de programação exige maior nível de conhecimento do desenvolvedor.

Nesse contexto, também está inserida a utilização da ferramenta *Sloodle*, que realiza a integração do Mundo Virtual (*Second Life* ou *Open-Sim*) com o ambiente *Moodle*. Seu objetivo está centrado em facilitar o processo de inserção de materiais educacionais nos Mundos Virtuais (*slides*, vídeos, textos e questões), além de registrar a presença e realização de atividades no Mundo Virtual, como questionários, diretamente no ambiente *Moodle*, para posterior verificação pelo professor. Um total de nove artigos utilizou esta ferramenta nas pesquisas realizadas, obtendo resultados positivos em relação à aplicação dessa ferramenta para auxiliar no processo de aprendizagem.

Descritos os aspectos voltados ao ponto de vista tecnológico, faz-se necessário esclarecer as particularidades relacionadas às teorias educacionais e linhas de pensamento desempenhadas nesses 58 artigos, a fim de evidenciar quais embasamentos os autores usaram para construir e/ou aplicar esses ambientes no âmbito educacional. Ao efetuar a análise das investigações selecionadas, constatou-se a utilização de diferentes fundamentos. Inclusive, alguns realizaram a associação de duas ou mais teorias no mesmo trabalho, seguindo os princípios de aprendizagem colaborativa (6), aprendizagem baseada em problemas (1), aprendizagem experiencial (4), aprendizagem significativa (2), construcionismo (4), construtivismo (1), carga cognitiva (1), flexibilidade cognitiva (1), comportamento planejado (1), teoria do *flow* (2), teoria psicogenética (1), pedagogia dos gêneros (1), ciclo de ensino e aprendizagem (1) e taxonomia de Bloom (2). Alguns autores não mencionaram uma teoria específica em que embasaram suas pesquisas. Por outro lado, em cada uma dessas investigações, foi citado um pesquisador reconhecido, entre eles: Vygotsky, Bloom, Piaget, Freire e Fletcher. Além disso, outros autores não empregaram nenhum fundamento educacional, perfazendo o total de 34 pesquisas.

Quanto às áreas de aplicação desses 58 artigos, foram identificados artigos desenvolvidos na área de Computação (13), Física (4), Língua Estrangeira (4), Matemática (3), Química (2), Eletrônica (2), Ciências Gerais (2), Enfermagem (1), Arqueologia (1), Literatura (1), Administração de Negócios (1), Engenharia Aeroespacial (1). Os demais artigos, que não apresentaram uma área específica, foram classificados como: Autoria (8), Centro de Ensino e/ou Administração (10) e Análises dos Estudantes (5).

Figura 1.2 - Análise Temporal dos Artigos, Teorias e Áreas de Aplicação.



Fonte: Nunes et al. (2016).

Com o intuito de melhor representar as informações derivadas da revisão sistemática, optou-se pela representação gráfica por meio de uma série temporal (Figura 1.2), em que foram demonstrados os dados obtidos no intervalo dos anos analisados, reproduzindo também quais foram as teorias educacionais empregadas e as áreas de aplicação dos Mundos Virtuais educacionais durante o período. Em vista disso, buscou-se a discussão das cinco teorias mais utilizadas, a fim de evidenciar suas características e apresentar os elementos do Mundo Virtual que foram explorados para a sua execução.

As pesquisas que utilizaram a teoria educacional que enseja uma aprendizagem colaborativa tinham como objetivo oportunizar a cooperação entre os estudantes para a realização de determinada atividade, aspecto que é ressaltado nos mundos virtuais pela disposição de um espaço compartilhado: quando o estudante efetua alguma mudança nele, os demais colegas conseguem verificar em tempo real. Aliado a isso, os alunos também possuem à sua disposição a ferramenta de *chat*, a qual possibilita a troca de mensagens instantâneas para a solução de dúvidas e discussão generalizada de uma problemática. Essa teoria foi abordada nas áreas de Computação, Matemática, análises dos estudantes e Centros de Ensino e/ou Administração. Essas áreas propiciam a oportunidade de uma aprendizagem baseada na colaboratividade, fator que foi constatado nesses artigos por meio dos experimentos realizados, que ensejam uma maior troca de conhecimentos e primam pelo uso do espaço de forma compartilhada.



Já os autores que conduziram investigações com base na taxonomia dos objetos educacionais, mais conhecida como Taxonomia de Bloom, buscaram viabilizar aos usuários do Mundo Virtual as três possibilidades de aprendizagem que Bloom defende, por meio da classificação do cognitivo, afetivo e psicomotor. Nas áreas de Arqueologia e Computação, abordadas por essa teoria, foram trabalhados conteúdos para o desenvolvimento intelectual, com objetos 3D e demais elementos disponíveis no ambiente, abrangendo aspectos afetivos entre os usuários, assim como realizando a execução prática de atividades quando imersos no ambiente, favorecendo o desenvolvimento de habilidades que envolvem o aparelho motor do aluno.

As atividades desenvolvidas com base em experimentação buscam oportunizar aos alunos criar seus conhecimentos a partir da transformação de uma experiência. Assim, a partir do desenvolvimento de uma “experiência concreta”, os alunos aproveitam para observar, refletir e buscar compreender acontecimentos decorrentes da transformação. A característica em comum dessa estratégia consiste em oportunizar aos estudantes envolvimento ativo nas quatro etapas do Ciclo de Kolb (1984): Experiência Concreta (CE), Conceituação Abstrata (AC), Observação Reflexiva (RO) e Experimentação Ativa (AE). A experiência concreta imediata é a base da observação e reflexão. Nesse processo, são envolvidos os quatro tipos diferentes de habilidades inerentes ao Ciclo de Kolb: capacidade de se envolver totalmente, abertamente e sem preconceitos em novas experiências (CE), capacidade de refletir e observar suas experiências a partir de muitas perspectivas (RO), capacidade de

criar conceitos que integrem suas observações em teorias lógicas (AC) e capacidade de usar essas teorias para tomar decisões e resolver problemas (AE).

Com relação aos Mundos Virtuais implementados e fundamentados na teoria construcionista (PAPERT, 2008), os autores empenharam-se no desenvolvimento de atividades que favorecessem a construção do conhecimento baseado na realização de alguma ação concreta, que poderia resultar em algo palpável e do interesse dos estudantes imersos no ambiente. Essa teoria foi aplicada nas áreas de Aúria, Computação e Física, o que facilita o desempenho de atividades pelos estudantes imersos no Mundo Virtual, visto que os autores desses estudos incentivaram os estudantes a interagirem com os objetos disponíveis e até mesmo construírem seus próprios objetos 3D, fazendo modificações ou complementações que resultaram em alguma ação, as quais foram desempenhadas por programação de *scripts* que compõem um determinado objeto, pela manipulação de objetos e pelo preenchimento de valores para a execução de um experimento de maneira correta.

Quanto aos autores que desempenharam a teoria da aprendizagem significativa no desenvolvimento e aplicação de Mundos Virtuais educacionais, estes propuseram a utilização dos princípios abordados por Ausubel (2000) para o emprego dos conceitos subsunçores. Dessa forma, promoveram a aprendizagem dos usuários imersos em conteúdos já existentes na estrutura cognitiva do aluno. Portanto, primeiramente, era oferecida uma base para que um tópico mais complexo pudesse ser trabalhado. Nessa perspectiva, concebeu-se ao estudante um crescimento pela modificação do conceito subsunçor, fazendo com

que ele conseguisse evoluir. Essa teoria foi associada à tese da aprendizagem experiencial e da flexibilidade cognitiva, sendo trabalhada nas áreas de Física e Computação.

Discorridas as particularidades presentes juntamente ao gráfico analisado, outro aspecto importante a ser verificado está relacionado à forma de avaliação do estudo realizado e ao público-alvo. Foi possível mensurar que, dos 58 artigos analisados, um total de 14 não continha nenhum tipo de avaliação com usuários: somente apresentava o ambiente desenvolvido. Tal constatação se deve ao fato de que, em grande parte desses artigos, foi efetuada uma apresentação geral do ambiente e dos objetivos a que visavam contemplar, geralmente associados a uma teoria educacional que serviu de embasamento para a construção do ambiente, postergando para trabalhos futuros a realização de avaliações com usuários.

Nos demais artigos analisados, realizou-se uma divisão por quatro tipos de níveis educacionais, em que o maior número de avaliações realizadas envolveu estudantes de graduação (23) de diferentes áreas de ensino, visto que se trata de uma fase que proporciona realizar testes em diferentes disciplinas de um curso com a mesma turma durante um período maior de tempo. Já nos cursos de pós-graduação (7), uma fase em que os estudantes estão mais focados em suas pesquisas e têm menor número de disciplinas, torna-se mais complicada a realização de testes. Por fim, os cursos de licenciatura também obtiveram diversas pesquisas realizadas (8), nas quais o escopo desses estudos esteve centralizado na avaliação por parte dos futuros docentes sobre os ambientes desenvolvidos para uma determinada área de ensino. Em menor número, estão

alunos do ensino fundamental (6), visto que se trata de estudantes menores de idade que necessitam da autorização de seus pais em testes realizados e também podem vir a ter mais dificuldades em entender o funcionamento do ambiente e seguir, restritamente, uma determinada metodologia aplicada.

### **1.1.3 Possibilidades e vantagens dos laboratórios virtuais imersivos para o ensino de Ciências**

O avanço das tecnologias digitais vem sendo impulsionado diariamente, influenciando as pessoas no modo de agir, comunicar-se, relacionar-se, solucionar problemas e também de aprender.

A *Internet* móvel, os dispositivos móveis com tela sensível ao toque e as redes sem fio têm oportunizado o acesso cada vez mais rápido, fácil e intuitivo, levando crianças, jovens e adultos a estarem constantemente conectados, buscando por informações e conhecimento de forma autônoma.

A variedade de ferramentas que possibilita o acesso às informações torna-se mais ampla em quantidade de modelos e armazenamento, porém oferece dispositivos menores em tamanho e espessura, incentivando seu uso em qualquer momento e lugar, sem a necessidade de fios e cabos para conexão.

Ao mesmo tempo em que avanços das tecnologias facilitam as atividades diárias, experiências também são relatadas sobre o uso de *mobile learning* no contexto educacional, transformando os modos de aprender e ensinar (PRENSKY, 2001; WANKEL; BLESSINGER, 2012).

Atualmente, a tecnologia de informática e comunicação permite criar material educacional digital, usando multimídia com interatividade, que torna mais efetivos os ambientes de ensino-aprendizagem apoiados na Tecnologia de Informação e Comunicação (TIC).

No entanto, a disponibilidade crescente de laboratórios de informática nas escolas contrasta com a carência de laboratórios para apoiar as atividades de ensino-aprendizagem em Ciências nas escolas.

Laboratórios virtuais usando ambientes imersivos e dispositivos móveis começam a surgir, bem como soluções prontas ou ferramentas de autoria para a criação de laboratórios virtuais, tanto de *software* comercial como de *software* livre disponíveis no mercado internacional e nacional.

No desenvolvimento de soluções em termos de laboratórios virtuais voltadas ao contexto da educação no país, é preciso aproveitar as estratégias já existentes, empregando-as como elemento alavancador para o delineamento e a construção de novas soluções, que permitam disponibilizar não apenas um novo elenco de soluções em termos de laboratórios virtuais, como também promover capacitação para o desenvolvimento de laboratórios virtuais voltados ao ensino de Ciências.

Os laboratórios reais são os laboratórios de Ciências, Biologia e Química que estão disponíveis na maioria das escolas. Compostos por bancadas, reagentes, vidrarias e diversos outros equipamentos, são utilizados para desenvolver atividades em que os professores ou os próprios alunos (quando há conjuntos de experimentação suficientes) exploram e manipulam os materiais.

Conhecidos também pelo termo *Hands-on* (KLAHR *et al.*, 2007), essa prática tem seu mérito defendido por diferentes autores (KLAHR *et al.*, 2007; SCALISE *et al.*, 2011), que consideram importante que os alunos possam tomar suas decisões baseadas no uso de materiais, na exploração de situações. Afirmam, ainda, que trabalhar com materiais palpáveis promove uma aprendizagem que vai ao encontro da teoria construtivista, pois disponibiliza fontes de ativação cerebral que proporcionam motivação e engajamento nos alunos (KLAHR *et al.*, 2007).

No entanto, os próprios autores apontam alguns aspectos que dificultam ou impedem o uso dos laboratórios reais, como a falta de espaço físico nas escolas, o perigo e a responsabilidade ao utilizar produtos químicos com os alunos, tempo de aulas disponíveis a fim de preparar os experimentos, coleta de dados, fazer os registros necessários, bem como os relatórios previstos e, subsequentemente, organizar e limpar os materiais dos laboratórios. Isso tudo deve ser considerado tendo em vista que os laboratórios das escolas públicas não possuem monitores e que o tempo de aula disponível para realizar os experimentos com os alunos é limitado.

Adicionalmente, ressalta-se que um grande fator impeditivo é relacionado com o custo para equipar, manter, fazer a manutenção e compra de materiais (insumos) para um laboratório real, considerando que atualmente as verbas para as escolas são cada vez menores.

Peffer *et al.* (2015) ressaltam que uma das barreiras para o trabalho com materiais reais é a viabilidade. As escolas estão limitadas por tempo, dinheiro, espaço, segurança e, também, pelo fato de que os experimentos são entregues para os alunos como “receitas de bolo”: eles

apenas seguem as instruções e, algumas vezes, sem sucesso, não conseguem chegar a uma conclusão efetiva sobre o resultado atingido durante a atividade.

Em contrapartida, os laboratórios virtuais podem ser uma opção mais atrativa para escolas que não possuem espaço físico e recursos financeiros para a implementação e uso de um laboratório real, mas que tenham condições de proporcionar aos seus alunos ambiente de aprendizagem baseado na tecnologia de informação e comunicação, incluindo tanto os laboratórios de informática existentes nas escolas como os dispositivos móveis de propriedade dos próprios alunos.

De acordo com os Parâmetros Curriculares do Ensino Médio (BRASIL, 2000), é relevante priorizar o desenvolvimento da autonomia intelectual e do pensamento crítico. O pensamento lógico é parte integral desse contexto. Além disso, os recursos atualmente disponíveis em tecnologia digital, em especial os recursos de *mobile learning* e laboratórios digitais, apresentam-se como uma estratégia com boa possibilidade de desenvolvimento dessas capacidades.

Os laboratórios virtuais permitem, por meio da realidade virtual, representar situações fidedignas da realidade concreta. Oportunizam aos alunos pensar e manusear uma situação que não seria possível no ambiente real, mas que *softwares*, aplicativos, simulações ou programações permitem experimentar, manipular e reproduzir quantas vezes forem necessárias. Além de viabilizar a experimentação em Ciências e Matemática, os laboratórios virtuais permitem ensinar a visualização de aspectos ou fenômenos difíceis, ou mesmo impossíveis de observar no

contexto e em tempo real, como, por exemplo: ondas, aceleração, velocidade, erosão, enfim, ocorrências que não são possíveis de observar a olho nu em um espaço de tempo limitado (SCALISE *et al.*, 2011).

Conseguir atingir o resultado científico esperado nem sempre é garantia de que o aluno conseguiu entender a atividade prática, pois, de acordo com alguns críticos, os *feedbacks* podem ser confusos e inconsistentes (KLAHR *et al.*, 2007), com características abstratas e difíceis de serem conceituadas (PEFFER *et al.*, 2015), levando a dúvidas em vez de oportunizar a construção de aprendizagens.

Situações que necessitam ser repetidas diversas vezes, também, podem ser melhor exploradas em laboratórios virtuais, uma vez que o recurso, além de oportunizar a repetição, permite que parâmetros do experimento sejam variados, o que seria difícil ou praticamente impossível de fazer em um laboratório real, como, por exemplo, a variação de massa, de velocidade, de gravidade.

Conforme destacado por Maraschin (2000), o laboratório virtual é um ambiente que permite o uso intencional da reversibilidade às situações de controle de instrumentos que simulem alterações nos objetos das experiências propostas, problematizando, desta maneira, as diferenças de cada usuário e/ou grupo com vista à compreensão dos conteúdos desenvolvidos.

Os alunos, ao trabalharem em laboratórios virtuais com a exploração de *softwares* e simuladores, necessitam tomar decisões e exercitar a crítica a todo instante. Ao executar esses recursos, é necessário que pensem por si mesmos, avaliem evidências, tenham calma e também opinião própria. Diversas hipóteses necessitam ser efetuadas. Além



disso, o aluno deve buscar meios de aprender e reconhecer dados científicos, a fim de utilizá-los diante das diferentes situações que encontrar (KLAHR *et al.*, 2007).

O controle de parâmetros pode ser manipulado pelo aluno, representando situações e possibilitando a aprendizagem por meio de tentativas e erros, sendo adaptável a qualquer disciplina e/ou faixa etária.

Trabalhar com simulações em laboratórios virtuais permite que os alunos façam descobertas, o que é imprescindível para a idade e também para a construção de conhecimentos, além de descobrir por si próprios, elaborar as suas hipóteses e reflexões com base no conhecimento científico estudado.

Mudanças de variáveis, repetições, parâmetros, pausas, coletas de dados, enfim, uma infinidade de possibilidades para usar em uma mesma situação ocasiona testar e tirar suas próprias conclusões, que, necessariamente, não precisam ser corretas, pois as ferramentas de um laboratório virtual oferecem ao aluno a vantagem de reproduzir a situação, refletir e reelaborar hipóteses e testá-las a qualquer momento.

As funções de um simulador podem ser ferramentas de cognição, visto que, diante das inquietações, os alunos sofrem perturbações, fazem validações, assimilam e acomodam as informações coletadas.

Além dos elementos multimídia, recursos de *mobile learning* e laboratórios virtuais permitem que experiências multimídia interativas, envolventes e interessantes possam ser compartilhadas, apoiando o processo de construção de aprendizagem e oportunizando um traba-

lho colaborativo e cooperativo, em que o sujeito constrói conhecimento baseado em suas vivências e experiências com o meio, conforme preconizado na teoria construtivista de Piaget (1993).

Ao criar situações virtuais, os alunos têm a oportunidade de trabalhar com materiais de forma muito mais rápida, pois, ao contrário dos laboratórios reais, a simulação permite montar, testar e realizar adequações de forma muito mais ágil.

Com alguns cliques e em segundos, é possível efetuar modificações que, em tempo real, levariam muito mais tempo e iriam requerer maior destreza dos alunos. Desse modo, no trabalho em laboratórios virtuais, o tempo consegue ser mais bem aproveitado e as atividades, desenvolvidas de acordo com a aula prevista.

O uso de laboratórios virtuais propicia às crianças e adolescentes novas maneiras de interagir, tendo em vista o potencial oferecido pelos computadores e a facilidade de utilizar *softwares*, simuladores, objetos digitais, dentre outros recursos que favorecem uma aprendizagem baseada em problemas que incentivam a exploração e teste de hipóteses, as quais constituem degraus na formação do raciocínio lógico sobre o qual deve ser construído o conhecimento em Ciências em especial.

Ao criar um ambiente de ensino inovador, aprimorado com o uso de tecnologias *mobile learning* e laboratórios digitais, novas estratégias pedagógicas e instruções poderão ser delineadas juntamente com os professores, apoiadas em estudos de referencial teórico relacionado à construção de aprendizagens e formação do pensamento lógico no adolescente para identificar ferramentas que contribuam diretamente com este desenvolvimento cognitivo.

Pela conectividade, os alunos conseguem promover a sua aprendizagem, por meio do acesso imediato a conteúdos, comunicação com demais colegas, pesquisas e diferentes fontes, o que permite que o conhecimento seja manipulado e também avaliado (GIKAS; GRANT, 2013). Dessa maneira, será possível incentivar o uso da tecnologia *mobile learning* e os laboratórios digitais como instrumentos de transformação pedagógica (DOMINGO; GARGANTÉ, 2016).

Conforme Scalise *et al.* (2011), ao escolher trabalhar com laboratórios virtuais ou reais, sempre existirão perdas e ganhos, sendo necessário considerar, em qualquer análise, aspectos referentes à realidade da escola, aos custos de materiais, à flexibilidade de espaço, às ferramentas disponíveis e ao tempo disponível para as atividades experimentais em sala de aula e como atividade extraclasse.

#### **1.1.4 Diferenças entre real e virtual no processo de aprendizagem**

Assim como alunos adeptos ao uso de laboratórios virtuais, verificam-se, também, aqueles que se sentem instigados a observar a manipulação de vidraria, cheiros, cores e possíveis resultados de uma experimentação, isto é, quando possuem materiais em número suficiente e os alunos podem realizar os experimentos e não apenas assistem a uma demonstração feita pelo professor.

Considerando a realidade da maior parte das escolas brasileiras, a presença de laboratórios para o ensino de Ciências é insuficiente. Laboratórios digitais oferecem uma alternativa para combinar a alfa-

betização digital, inerente aos estudantes do ensino médio, com as possibilidades que a tecnologia de realidade virtual, realidade aumentada e *mobile learning* combinadas viabilizam.

Adicionalmente, é importante lembrar que alunos estudantes de ensino médio, considerados “nativos digitais”, termo cunhado por Prensky (2001a), possuem muitas habilidades e competências *on-line*, sendo assim, em sua maioria, adeptos das tecnologias emergentes. Preferem trabalhar com sugestões, apontamentos de *feedbacks* imediatos, possibilidades de repetições, pausas, *zoom* e avanços que podem ser realizados rapidamente.

Scalise *et al.* (2011), ao se referirem ao uso de laboratórios virtuais, salientam que, para essa geração de estudantes, as aulas consideradas inesquecíveis ao aprendizado estão nos laboratórios de ciências, sendo que muitos alunos experimentam parte do seu estudo científico pelos laboratórios virtuais, experimentos de simulações e demonstrações de fenômenos via *software* de computador.

Silva (2017), em sua pesquisa de doutorado, explora a utilização de tecnologias digitais por crianças na primeira infância. É apontado como resultado que as crianças na primeira infância percebem as tecnologias digitais de modo diferente do que os materiais reais (concretos), conseguindo realizar as atividades que lhes foram oferecidas para exploração com uso de recursos digitais com maior desenvoltura.

De acordo com os aspectos relatados até então, ambas as experiências de laboratórios (reais e virtuais) possuem seus adeptos, defensores, vantagens e também limitações. No entanto, cabe ressaltar

algumas características desses laboratórios, no tocante à aprendizagem e à construção do conhecimento por parte dos alunos, bem como os objetivos a serem alcançados pela legislação vigente.

Conforme preconizado nos Parâmetros Curriculares Nacionais (BRASIL, 2000, p. 91), “a construção da Base Nacional Comum passa pela constituição dos saberes integrados à ciência e à tecnologia, criados pela inteligência humana”. Já o documento PCN, também, dispõe sobre a necessidade de investigar o uso do próprio estudo da tecnologia, ultrapassando, assim, o “discurso sobre as tecnologias”, identificando nas matemáticas, nas ciências naturais, nas ciências humanas, na comunicação e nas artes os elementos de tecnologia que lhes são essenciais e desenvolvê-los como conteúdos vivos, como objetivos da educação e, ao mesmo tempo, meio para tanto (MENEZES, 1998).

Zervas *et al.* (2014) comentam que soluções que facilitam o desenvolvimento de aplicações para os dispositivos móveis capazes de oferecer plataformas com simulações de laboratórios virtuais começam a se proliferar. No entanto, ainda cabe investigar de forma mais detalhada o impacto dos recursos digitais no processo de cognição e aprendizagem, tendo em vista que grande parte dos estudos disponíveis na base de dados Scopus (2018) apresenta relatos de atividades desenvolvidas em disciplinas no Ensino Médio, principalmente de Química e Física.

A aprendizagem usando dispositivos móveis (*mobile learning*) tornou-se um componente importante em tecnologia educacional, pois promove o aprender, colabora e compartilha ideias com a ajuda da *Internet* e também o desenvolvimento das tecnologias. O uso apropriado desse componente de aprendizagem precisa considerar o modo como

os alunos se comunicam, interagem e se comportam ao utilizá-lo, pensando que a aprendizagem poderá ocorrer em múltiplos contextos (AL-EMRAN *et al.*, 2016). Os laboratórios virtuais permitem a exploração tanto no laboratório de informática da escola como nos dispositivos móveis de propriedade dos próprios alunos, o que facilita o trabalho do professor e faz com que o tempo seja otimizado. De acordo com os Parâmetros Curriculares do Ensino Médio (BRASIL, 2000), é relevante priorizar o desenvolvimento da autonomia intelectual e do pensamento crítico. O pensamento lógico é parte integral desse contexto. Usar os recursos atualmente disponíveis da tecnologia digital, em especial, os recursos *mobile learning* e laboratórios digitais, apresenta-se como uma estratégia com boa possibilidade de desenvolvimento dessas capacidades.

Os laboratórios virtuais permitem, por meio da realidade virtual, representar situações fidedignas da realidade concreta, oportunizando aos alunos pensar e manusear uma situação que no ambiente real não seria possível, mas que *softwares*, aplicativos, simulações ou programações disponibilizam experimentar, manipular e reproduzir quantas vezes forem necessárias.

Além de viabilizar a experimentação em Ciências, os laboratórios virtuais ensinam a visualização de aspectos ou fenômenos difíceis ou mesmo impossíveis de observar no contexto real ou até em tempo real, como, por exemplo, ondas, aceleração, velocidade, erosão, enfim, ocorrências que não são possíveis de observar a olho nu em um espaço de tempo limitado (SCALISE *et al.*, 2011). Outro fato importante da experimentação em laboratórios virtuais é a possibilidade de repetição das

situações com modificações de variáveis, o que oportuniza aos alunos verificar o que aconteceria, em dado evento, caso valores de grandezas fossem aumentados ou diminuídos.

Al-Emran *et al.* (2016), em pesquisa realizada com alunos universitários e pós-graduandos, evidenciaram que eles percebem e utilizam seus dispositivos móveis para estudar e gerenciar suas aprendizagens. Também relatam que os alunos se motivam a utilizar as suas tecnologias para estudos, acarretando, assim, que sejam inseridas em qualquer nível de ensino.

Os mesmos autores ressaltam que as tecnologias *M-learning* facilitam o compartilhamento de conhecimento entre alunos e educadores, promovendo aprendizagem mediada por diferentes contextos e também o desenvolvimento do pensamento crítico, pois os alunos compartilham, colaboram e discutem as ideias entre si diante da resolução de uma situação.

No que tange ao processo de construção do conhecimento e desenvolvimento da aprendizagem, várias ideias de pensadores influentes na área da educação podem ser apontadas. Porém, neste Capítulo, serão enfatizadas as abordagens da teoria da Epistemologia Genética (PIAGET, 1976) e a Teoria Experiencial (KOLB, 2015), que têm por base as obras de Dewey, Lewin e também Jean Piaget. A escolha destes autores aconteceu por pertinência e por afinidade das ideias.

Durante o uso de laboratórios virtuais, evidencia-se o pressuposto de que os alunos, ao explorarem os recursos, estão em contato com um objeto (PIAGET, 1976) e ainda que necessitam experimentar e resolver uma situação-problema (KOLB, 2015).

Mediante trabalhos de grupo, discussões de ideias e questionamentos, permite-se que os alunos sofram desequilíbrios em seus sistemas conceituais (PIAGET, 1976), levando-os a buscar respostas para suas indagações.

As atividades desenvolvidas em laboratórios virtuais com os alunos oferecem a perspectiva de uma experiência, da resolução de um problema, ou seja, uma visão construtivista, em que é priorizada a experiência, o aprender por perguntas, por meio de informações, explicações, confirmações experimentais e por análises críticas, diferente de treinar e praticar para a reprodução de respostas. Juntamente com o fato de aprender por experimentação e perguntas, encontra-se a ocasião de realizar trabalhos coletivos e cooperativos, tendo em vista que as atividades podem ser desenvolvidas aos pares.

Diante das atividades a serem desenvolvidas em um laboratório virtual, o aluno, ao descobrir o que acontece com o objeto de experimentação e verificar as transformações ocorridas, sofre modificações em seu sistema cognitivo. As transformações fundamentais, que ocorrem em todo o sistema cognitivo, são definidas por Piaget (1976) como assimilação e acomodação.

A assimilação também é conhecida pela incorporação de um elemento exterior em um esquema sensório-motor ou conceitual, mesmo sendo ele um objeto ou até mesmo um acontecimento, e ocorre sempre que estes esquemas se relacionam. Quando se assimila algo, as características dos elementos assimilados são consideradas, chegando-se, assim, ao processo de acomodação, que acontece de forma diferente, pois são observadas as particularidades de cada objeto.



Conforme Piaget (1976), a acomodação é subordinada à assimilação, pois sempre é necessário acomodar um esquema da assimilação:

[...] esta subordinação é mais estreita e sobretudo mais previsível no caso destas acomodações recíprocas que no das adaptações aos objetos exteriores A', B', C', etc., quando novos dados observáveis surgem de maneira inesperada sob a pressão da experiência. (PIAGET, 1976, p. 14).

Para tanto, não basta apenas que o sujeito incorpore elementos externos, mas sim que ele exerça atividades, pois, caso contrário, poderia assimilar diferentes informações sobre tudo o que vê, sem ao menos compreendê-las. Em síntese, pode-se dizer que o sujeito realiza assimilações, as quais, posteriormente, são acomodadas e necessitam estar em equilíbrio. Durante o processo de equilíbrio, tateamentos sucessivos acontecem e são eliminados os resultados em que não se obteve êxito. Aqueles com êxito são retidos à medida que o sujeito busca por estabilidade (equilíbrio). Mas como fazer com que os sujeitos obtenham assimilações, acomodações e cheguem ao equilíbrio? Segundo Piaget (1976), uma das fontes para que o desenvolvimento do conhecimento progrida é o desequilíbrio, que faz com que o sujeito ultrapasse seu estado atual e busque por novas direções.

[...] os desequilíbrios não representam senão um papel de desencadeamento, pois que sua fecundidade se mede pela possibilidade de superá-los – quer dizer, sair deles. É pois evidente que a fonte real do progresso deve ser procurada na reequilibração, naturalmente, no sentido não de um retorno à forma anterior de equilíbrio, cuja insuficiência é responsá-

vel pelo conflito ao qual esta equilibração provisória chegou, mas de um melhoramento desta forma precedente. (PIAGET, 1976, p. 19).

A todo momento, o sujeito interage com o objeto, ocorrendo desequilíbrios, assimilações, acomodações e equilíbrios. Desses processos, Piaget (1976) ressalta a importância para a equilibração, que, de acordo com o autor, é “o fator fundamental do desenvolvimento cognitivo” (PIAGET, 1976, p. 23), não apenas um dos aspectos, vindo, sim, em segunda ordem das construções que são características de cada estágio.

É importante salientar que nos períodos iniciais existe um motivo ordenado de desequilíbrio, que se modifica de estágio em estágio, tendendo sempre para um sentido de melhor equilíbrio, o qual acontece de diferentes formas, e uma das formas do seu mecanismo são as regulações, as quais se mostram importantes pela possibilidade de *feedbacks*, sejam eles positivos ou negativos.

Fala-se de regulação, de modo geral, quando a retomada  $A'$  de uma ação  $A$  é modificada pelos resultados desta, logo quando de um efeito contrário dos resultados de  $A$  sobre seu novo desenvolvimento  $A'$ . A regulação pode, então manifestar-se por uma correção de  $A$  (*feedback* negativo) ou por seu reforçamento (*feedback* positivo), mas neste caso com possibilidade de um crescimento do erro [...] ou de sucesso (formação dos hábitos, etc.) [...]. (PIAGET, 1976, p. 24).

O sujeito, quando recebe um *feedback* negativo e se torna consciente dele, percebe a existência de uma lacuna e, por isso, tenta buscar alternativas para a sua resolução, pois percebe a necessidade de correção. Já ao receber um *feedback* positivo, o sujeito está recebendo um reforço, existe satisfação ao recebê-lo.

É importante salientar que o sujeito, ao receber um *feedback* positivo, tem um grande valor e este valor é atribuído por ele mesmo. Existe uma necessidade de satisfação ao recebê-lo.

Para Piaget (1976), essa necessidade pode ser considerada um desequilíbrio momentâneo, e a satisfação, uma reequilibração.

Ao se observar como acontece o processo de construção do conhecimento, percebe-se que o uso dos laboratórios virtuais apresenta elementos que são essenciais ao desenvolvimento do sistema cognitivo do sujeito, como, por exemplo, situações que geram desequilíbrios, possibilidade de testar, verificar como se desenvolve o processo, mudanças de variáveis, bem como a observação dos acontecimentos.

Como adicional, também se obtêm *feedbacks*, sejam eles positivos ou negativos, os quais são ressaltados por Piaget (1976) como elementos fundamentais para que ocorram os processos de desequilíbrio e reequilibração.

Durante o trabalho com os laboratórios virtuais, ressalta-se a importância do trabalho colaborativo e cooperativo. Os alunos possuem a oportunidade de trabalhar aos pares, favorecendo a discussão de ideias, levantar hipóteses e questionamentos, ações essas que possibilitam aos alunos sofrer desequilíbrios em seus sistemas conceituais (PIAGET, 1976), levando-os a buscar respostas para suas indagações.

O trabalho cooperativo se mostra importante no que tange à psicologia do pensamento, pois a criança não é passiva e, quando em contato com as demais crianças, vai elaborando a razão. Sendo assim, as atividades aos pares contribuem para a formação do pensamento racional.

À medida que grupos se formam e trabalham em um assunto comum, existe a colaboração e a troca de ideias, pois a tomada de consciência é estimulada pela cooperação (PIAGET, 1993).

Outro fato relevante no trabalho cooperativo é que ele contribui para a objetividade, pois o sujeito, quando trabalha sozinho, fica limitado somente à sua perspectiva. Contudo, ao trabalhar cooperativamente, observa também o ponto de vista do outro.

Vê-se, assim sendo, que a cooperação não age somente sobre a tomada de consciência do indivíduo e sobre o seu senso de objetividade, mas termina, afinal, por constituir toda uma estrutura normativa que remata sem dúvida o funcionamento da inteligência individual, mas completando-a no sentido da reciprocidade – essa norma fundamental que é a única a conduzir o pensamento racional. Pode-se, pois, dizer, parecidos que a cooperação é verdadeiramente criadora, ou, o que vem a ser o mesmo, constitui a condição indispensável para a completa formação da razão. (PIAGET, 1993, p. 8).

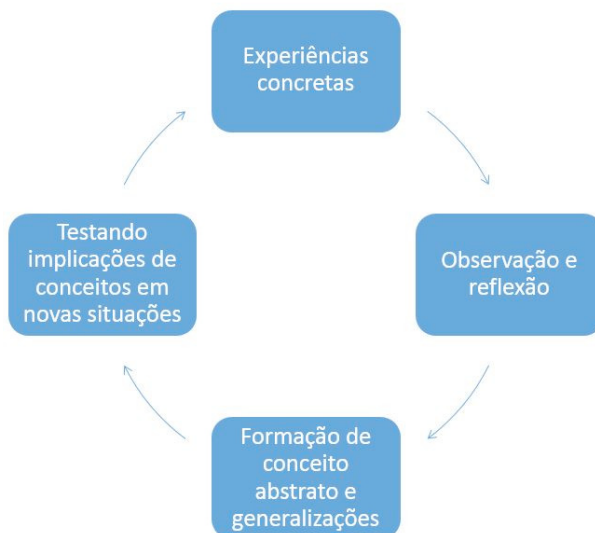
Adicionalmente, pode-se destacar também que a equipe contribui para o desenvolvimento da independência intelectual de seus membros e, ainda, que “[...] Os frutos específicos do método são, assim, o espírito experimental, de um lado, e de outro, a objetividade e o progresso do raciocínio” (PIAGET, 1993, p. 15).

Na visão de Kolb (2015), o papel central da aprendizagem é a experiência. Por meio dela o aluno pode explorar, manipular, observar, coletar dados e também analisar fatos que acontecem durante o processo. A partir da análise e reflexão sobre os dados, o aluno chega a conclusões e consegue estabelecer conceitos que integrem as conclusões em teorias lógicas, que podem servir como subsídio para que novas experiências sejam realizadas.

Esse processo é evidenciado no diagrama do Ciclo de Kolb, conforme apresentado na Figura 1.3.

Com base na experiência concreta, na conceituação abstrata, na observação reflexiva e na experimentação ativa, o autor caracteriza um ciclo, conforme apresentado na Figura 1.3.

Figura 1.3 - O modelo de aprendizagem experiencial.



Fonte: Adaptado de Kolb (1984, p. 21).

Nesse modelo, o autor explica que a aprendizagem se dá por meio de um ciclo que acontece em quatro etapas: “experiência concreta”, base para que o aluno possa realizar a experimentação; “observação reflexiva”, observações e reflexões que são assimiladas e analisadas; “conceituação abstrata”, tendo em vista as assimilações e reflexões que aconteceram, conceitos abstratos surgem, podendo ser utilizados para impulsionar novas ações e também para impulsionar e servir como guia para a criação de novas experiências levando à “experimentação ativa”.

Por meio dessas quatro etapas, o estudante tem uma participação ativa, podendo acionar também suas experiências pessoais para resolver problemas a partir da compreensão intelectual de uma situação, o que lhe irá exigir um maior nível de abstração, pois demandará uma abordagem lógica e sistemática dos conceitos envolvidos.

Ao exercer a experimentação ativa, terá a oportunidade de experimentar e mudar variáveis, podendo testá-las, levantar hipóteses e posteriormente realizar a verificação.

É importante salientar que as experiências concretas ou imediatas propiciam a base para a realização de observações e reflexões, que são assimiladas e organizadas em conceitos abstratos. Esses conceitos, por sua vez, produzem novas implicações, oportunizando que novas ações sejam delineadas e testadas, proporcionando novas experiências.

Em sua teoria, Kolb (2015) também faz referência a um conjunto de habilidades (estilos de aprendizagem) que são utilizadas pelo aluno para cada situação de aprendizagem, tendo em vista que para alguns

é necessário observar e para outros, abstrair ou ainda interagir com o concreto para posteriormente realizar suas reflexões sobre o que foi percebido.

Diante desse modelo de aprendizagem experimental, observa-se que, ao explorar experiências concretas, observar, refletir, formular novos conceitos e, posteriormente, testá-los em novas situações, o aluno/sujeito utiliza também as suas vivências, características hereditárias, experiências já vividas e as exigências do contexto em que vive, para, a partir daí, fazer as suas escolhas e utilizar um modelo específico para aprender. Esse modelo específico, que traz algumas características padronizadas, é utilizado no momento de aprender. Kolb (2015) definiu-os como “estilos de aprendizagem”. No entanto, por se tratar de características muito peculiares e individuais de cada aluno/sujeito, cada um irá fazer uso do estilo de aprendizagem que achar mais pertinente.

Com base nas ideias defendidas por Piaget (1976) e Kolb (2015), percebe-se que os laboratórios virtuais apresentam recursos que fazem com que o aluno/sujeito possa construir seus conhecimentos a partir de mudanças em seu sistema conceitual, passando por desequilíbrios, assimilações, acomodações e equilíbrios, baseados em situações em que necessita solucionar um problema a partir de uma experimentação. Para tanto, tem a possibilidade de testar hipóteses, testar experiências, observar, refletir e obter condições de criar novos conceitos que sejam utilizados em situações diversas, fazendo uso de ferramentas que são próprias do seu contexto e que facilitam a percepção de conceitos que podem ser muito abstratos ou de difícil compreensão.

### 1.1.5 Motivação e MVI

O ensino de Ciências é considerado difícil, pois a compreensão dos conceitos abordados necessita de experimentos com problemas concretos ou de simulações para mostrar os efeitos ou causas das noções físicas. Muitas instituições de ensino não possuem recursos financeiros ou espaços adequados para representar diversos fenômenos que, ao serem simulados, qualificam o processo de ensino, aumentando o interesse e a motivação. Simulações computacionais podem ser utilizadas como recurso para reproduzir fenômenos, testar hipóteses, controlar e observar variáveis e situações-problema (ALDRICH, 2009).

Falar sobre motivação em Mundos Virtuais Imersivos requer primeiramente um olhar mais detalhado sobre o que é motivação e como ela pode ser trabalhada em Mundos Virtuais Imersivos. Por isso, foi escolhido um embasamento teórico para nortear esta seção.

A motivação é uma ideia central na Teoria da Autodeterminação (TAD). O estudo da motivação implica explorar a energia e direção do comportamento e, somente quando esses dois aspectos são contemplados, pode-se falar, verdadeiramente, de teorias da motivação em psicologia (DECI; RYAN, 1985).

Diz-se que a pessoa está motivada quando ela se sente ativa e com energia para atingir um fim e não está motivada quando não se sente impelida ou inspirada para a ação (RYAN; DECI, 2000a). A motivação pode ser definida como um estado energético interno que dirige o comportamento ou a ação e se relaciona com direção e persistência (RYAN;



DECI, 2000b, 2002). Esses dois aspectos podem ser trabalhados em mundos virtuais, os quais oportunizam ao aluno novas experiências, um novo olhar sobre o conteúdo ministrado.

A direção abrange os processos e as estruturas que dão significado aos estímulos internos e externos, direcionando a ação para a satisfação das necessidades. No estudo da motivação, a distinção mais básica e clássica ocorre entre a motivação intrínseca e a motivação extrínseca (DECI; RYAN, 1985).

Na **motivação intrínseca**, também conhecida como motivação interna, a pessoa propõe-se a realizar algo ou envolve-se numa tarefa porque esta é inerentemente agradável ou interessante. Corresponde a um desejo genuíno, a uma tendência inata do ser humano para explorar o mundo. A ação é vista como um fim em si mesma, e esse conceito está relacionado à força interior, que é capaz de se manter ativa mesmo diante de adversidades. Esse tipo de combustível se relaciona aos interesses individuais e que podem ser alterados apenas por escolha pessoal.

Ao contrário, na **motivação extrínseca**, também conhecida como motivação externa, o indivíduo realiza algo para alcançar determinado resultado. A ação responde a pressões externas, ou porque é útil para atingir determinado objetivo, independentemente da ação em si (MINER, 2015). O termo está conectado ao ambiente, às situações e aos fatores externos, como atingir uma nota para ser promovido no ano letivo ou estudar para uma prova para evitar represália.

Para trabalhar a motivação no ensino de Ciências, é necessário que haja um deslocamento de energia. Segundo Deci e Ryan (1985), a energia é, fundamentalmente, uma questão de necessidades: as neces-

sidades inatas e as necessidades que se adquirem nas interações com o meio. Quando é oferecido um novo estímulo ao aluno, como é o caso dos Mundos Virtuais Imersivos, ele responde com grau de curiosidade, que pode ser considerado um nível de motivação intrínseca.

A motivação é um conceito dinâmico, o qual varia ao longo de um contínuo (desmotivação, motivação controlada e motivação autônoma), comportamento que pode assumir diferentes níveis de autonomia ou autodeterminação, persistência e envolvimento. A TAD oferece as bases conceituais para a compreensão das razões que levam as pessoas a adotarem e manterem determinados comportamentos. É uma “meta-teoria”, com uma visão positiva do ser humano (RYAN; DECI, 2002).

Embasados nesse “*continuum* de autodeterminação” dessa teoria, afirma-se que é acreditável trabalhar níveis de motivação que podem ser elevados com uso de estímulos constantes e diversificados, uma das características básicas do Mundo Virtual Imersivo, uma vez que este pode oferecer interação com objetos, troca entre participantes, além de agentes virtuais conversacionais capazes de instigar novas possibilidades de dúvidas no mesmo experimento. Os Agentes Conversacionais são capazes que aumentar significativamente a motivação do aluno.

Essa teoria analisa a associação entre fatores contextuais e os diferentes tipos de motivação (RYAN; DECI, 2002). Na prática, um dos pontos principais é o efeito das variáveis do contexto social nos comportamentos intrinsecamente motivados (RYAN; DECI, 2000b). Essa contextualização se torna possível com utilização de Mundos Virtuais Imersivos, que têm surgido como alternativa para a realização de vários tipos de experiências, porque oferecem ao usuário a sensação de

realidade, permitindo, inclusive, a interação com objetos 3D. Os MVIs estimulam a participação ativa dos usuários e incentivam a resolução de problemas (AMARAL; AVILA; TAROUCO, 2012), o que pode contribuir para o aumento da motivação dos estudantes.

A ideia de envolvimento, por sua vez, está ligada com o grau de motivação para o engajamento de uma pessoa com determinada atividade. O envolvimento pode ser passivo, como ler um livro ou assistir a uma aula, ou ativo, ao participar de uma experiência virtual com uso de sensores, Agentes Conversacionais ou apenas colegas de classe. O Mundo Virtual Imersivo tem potencial para os dois tipos de envolvimento, ao permitir a exploração de um ambiente virtual e ao propiciar a interação do usuário com um Mundo Virtual Dinâmico.

## REFERÊNCIAS

ALDRICH, Clark. **Learning online with games, simulations, and virtual worlds**. San Francisco, CA: Jossey-Bass, 2009.

AL-EMRAN, Mostafa; ELSHERIF, Hatem M.; SHAALAN, Khaled F. Investigating attitudes towards the use of mobile learning in higher education. **Computers in human behavior**, v. 56, p. 93-102, Mar. 2016. Disponível em: <<https://doi.org/10.1016/j.chb.2015.11.033>>. Acesso em: 22 out. 2019.

ALLISON, Colin; MILLER, Alan; STURGEON, Thomas; NICOLL, James Ross; PERERA, Indika. Educationally enhanced virtual worlds. *In: ASEE/IEEE FRONTIERS IN EDUCATION CONFERENCE*, 40., 2010, Washington, D.C. **Proceedings** [...]. USA: Institute of Electrical and Electronics Engineers – IEEE, 2010.

AMARAL, Érico; AVILA, Bárbara Gorziza; TAROUÇO, Liane Margarida Rockenbach. Aspectos teóricos e práticos da implantação de um laboratório virtual no OpenSim. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE, 23., 2012, Rio de Janeiro. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2012. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/1696/1457>>. Acesso em: 3 ago. 2019.

AMARAL, Érico Marcelo Hoff do. **Processo de ensino e aprendizagem de algoritmos integrando ambientes imersivos e o paradigma de blocos de programação visual**. 2015. 255 f. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, Centro Interdisciplinar de Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/131053>>. Acesso em: 22 out. 2019.

ANTONIO, Caroline Porto. **Mundos virtuais 3D integrados à experimentação remota: aplicação no ensino de ciências**. 2016. 162 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Tecnologias da Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, 2016. Disponível em: <<https://repositorio.ufsc.br/xmlui/handle/123456789/172166>>. Acesso em: 22 out. 2019.

AUSUBEL, David P. **The acquisition and retention of knowledge: a cognitive view**. Dordrecht: Kluwer Academic Publishers, 2000.

BAINBRIDGE, William Sims (ed.). **Online worlds: convergence of the real and the virtual**. London: Springer-Verlag, 2010. (Human-Computer Interaction Series). Disponível em: <<http://link.springer.com/10.1007/978-1-84882-825-4>>. Acesso em: 25 ago. 2014.

BANERJEE, Indushree; PERERA, Indika; CHOUDHURY, Joydeep. Introducing immersive technologies for learning. In: INTERNATIONAL CONFERENCE ON CONTEMPORARY COMPUTING, 6., 2013, Noida, Índia. **Proceedings [...]**. USA: Institute of Electrical and Electronics Engineers – IEEE, 2013.

BRASIL. Ministério da Educação. **Parâmetros curriculares nacionais**. Brasília, DF: MEC, 2000.

CHOW, Meyrick. Determinants of presence in 3D virtual worlds: a structural equation modelling analysis. **Australasian journal of educational technology**, v. 32, n. 1, 2016. Disponível em: <<https://ajet.org.au/index.php/AJET/article/view/1939>>. Acesso em: 22 out. 2019.

DECI, Edward L.; RYAN, Richard M. **Intrinsic motivation and self-determination in human behavior**. New York: Plenum, 1985.

DOMINGO, Marta Gómez; GARGANTÉ, Antoni Badia. Exploring the use of educational technology in primary education: teachers' perception of mobile technology learning impacts and applications use in the classroom. **Computers in human behavior**, v. 56, p. 21-28, Mar. 2016.

FALCADE, Andressa; HERPICH, Fabrício; KRASSMANN, Aliane Loureiro; SILVA, Luiz Evandro Garcia da; MEDINA, Roseclea Duarte; JARDIM, Rafaela Ribeiro. Web viewers para ambientes imersivos: uma proposta de framework avaliativo. **Nuevas ideas en informática educativa – Memorias del XIX Congreso Internacional Informática Educativa (TISE 2014)**, Fortaleza, Brasil, 2014.

FERNÁNDEZ-GALLEGO, Beatriz; LAMA, Manuel; VIDAL, Juan C.; MUCIENTES, Manuel. Learning analytics framework for educational virtual worlds. **Procedia computer science**, v. 25, p. 443-447, 2013. (Part of special issue: 2013 International Conference on Virtual and Augmented Reality in Education. Edited by Jorge Martin-Gutierrez, Egils Ginters). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050913012611?via%3Dihub>>. Acesso em: 22 out. 2019.

GIKAS, Joanne; GRANT, Michael M. Mobile computing devices in higher education: student perspectives on learning with cellphones, smartphones & social media. **Internet and higher education**, v. 19, p. 18-26, 2013.

GOMES, José Duarte Cardoso. **Mundos virtuais e realidade aumentada: desenvolvimento e implementação de artefactos de média-arte digital para o ensino de educação musical no ensino básico**. 2016. 375 f. Tese (Doutorado em Média-Arte Digital), Universidade Aberta de Portugal, 2016.

GOMES, José Duarte Cardoso; FIGUEIREDO, Mauro Jorge Guerreiro. Desenvolvimento de recursos educativos em mundos virtuais. In: WCCA 2014 – WORLD CONGRESS ON COMMUNICATION AND ARTS, 7., 2014, Vila Real, Portugal. **Proceedings [...]. [S.l.]: COPEC – Science and Education Research Organization, 2014.**

GRIOU, David; MOLINA, José Manuel; CALLEJAS, Zoraida. An approach to develop intelligent learning environments by means of immersive virtual worlds. **Journal of ambient intelligence and smart environments**, v. 6, n. 2, p. 237-255, 2014.

IVORY, James D. **Virtual lives: a reference handbook**. Santa Barbara: ABC-CLIO, 2012. 269 p.

KHAN, Muhammad Saleem; SAFAAN, Samy Abdelwahab. Frequency of attendance and interaction of students in 3D multi-user virtual environments (MUVES): the effect on learning English language. **Advances in social sciences research journal**, v. 4, n. 6, p. 91-102, 2017. Disponível em: <<https://journals.scholarpublishing.org/index.php/ASSRJ/article/view/2891/1726>>. Acesso em: 22 out. 2019.

KICKMEIER-RUST, Michael D.; ALBERT, Dietrich. Learning analytics to support the use of virtual worlds in the classroom. *In*: HOLZINGER, Andreas; PASI, Gabriella (ed.). **Human-computer interaction and knowledge discovery in complex, unstructured, big data**. HCI-KDD 2013. Berlin: Springer, 2013. (Lecture notes in computer science, v. 7947). p. 358-365.

KLAHR, David; TRIONA, Lara M.; WILLIAMS, Cameron T. Hands on what? The relative effectiveness of physical versus virtual materials in an engineering design project by middle school children. **Journal of research in science teaching**, v. 44, n. 1, p. 183-203, 2007.

KOLB, David. **Experiential learning: experience as the source of learning and development**. Englewood Cliffs, NJ: Prentice Hall, 1984.

KOLB, David A. **Experimental learning: experience as the source of learning and development**. 2nd ed. New Jersey: Pearson Education, 2015.

KONSTANTINIDIS, Andreas; TSIATSOS, Thrasyvoulos; DEMETRIADIS, Stavros; POMPORTSIS, Andreas. Collaborative learning in OpenSim by utilizing Sloodle. *In*: ADVANCED INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS, 6., 2010, Barcelona. **Proceedings [...]**. USA: Institute of Electrical and Electronics Engineers – IEEE, 2010. p. 90-95.

KOTSILIERIS, Theodore; DIMOPOULOU, Nikoletta. The evolution of e-learning in the context of 3D virtual worlds. **Electronic journal of e-learning**, v. 11, n. 2, p. 147-167, 2013.

LUN, Erwin van. Conversational agent: a program that acts on behalf of humans and chats. **Blog Chatbots.org**. Nederland, 2011. Disponível em: <[https://www.chatbots.org/conversational\\_agent/](https://www.chatbots.org/conversational_agent/)>. Acesso em: 18 fev. 2015.

MARASCHIN, Cleci. Avaliação (da ou na) aprendizagem. *In*: CONGRESSO INTERNACIONAL DE EDUCAÇÃO DO COLÉGIO CORAÇÃO DE JESUS, 2., 2000, Florianópolis. **Anais [...]**. Florianópolis: Palotti, 2000. p. 36-39.

MENEZES, Luiz Carlos. **A tecnologia no currículo do ensino médio**. [S.l.: s.n.], 1998.

MINER, John B. **Organizational behavior 1: essential theories of motivation and leadership**. New York: Routledge, 2015.

MORETTI, Gaia; SCHLEMMER, Eliane. Virtual learning communities of practice in metaverse. *In*: ZAGALO, Nelson; MORGADO, Leonel; BOA-VENTURA, Ana (org.). **Virtual worlds and metaverse platforms: new communication and identity paradigms**. Hershey, PA: IGI Global, 2012. p. 149-165.

NELSON, Brian C.; ERLANDSON, Benjamin E. **Design for learning in virtual worlds**. New York: Routledge, 2012.

NUNES, Eunice Pereira dos Santos; ROQUE, Licínio G.; MARQUES, Fátima de Lourdes dos Santos Nunes. Measuring knowledge acquisition in 3D virtual learning environments. **IEEE Computer graphics and applications**, v. 36, n. 2, p. 58-67, Mar./Apr. 2016.

NUNES, Felipe Becker *et al.* Viewers para ambientes virtuais imersivos: uma análise comparativa teórico-prática. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 11, n. 1, jul. 2013.

NUNES, Felipe Becker; HERPICH, Fabrício; PASCHOAL, Leo Natan; LIMA, José Valdeni de; TAROUCO, Liane Margarida Rockenbach. Systematic review of virtual worlds applied in education. *In*: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – CBIE, 5.; SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE, 27., 2016, Uberlândia. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2016.

OLIVEIRA, Leander Cordeiro de. **Artefato metodológico de autoria aplicado aos mundos virtuais para educação**. 2015. 181 f. Dissertação (Mestrado em Engenharia de Computação) – Centro de Ciências Computacionais, Universidade Federal do Rio Grande do Norte, Natal, 2015.

PAPERT, Seymour. **A máquina das crianças: repensando a escola na era da informática**. Tradução: Sandra Costa. ed. rev. Porto Alegre: Artmed, 2008.

PEFFER, Melanie E.; BECKLER, Matthew L.; SCHUNN, Christian; RENKEN, Maggie; REVAK, Amanda. Science classroom inquiry (SCI) simulations: a novel method to scaffold science learning. **PLoS ONE**, v. 10, n. 3, e0120638, 2015. Disponível em: <<https://doi.org/10.1371/journal.pone.0120638>>. Acesso em: 3 ago. 2019.

PIAGET, Jean. **A equilibração das estruturas cognitivas**: problema central do desenvolvimento. Rio de Janeiro: Zahar, 1976.

PIAGET, Jean. O trabalho por equipes na escola. Tradução: Luiz G. Fieury. **Revista de educação**, Diretoria do Ensino do Estado de São Paulo, São Paulo, set./dez. 1936. Adaptação para o português moderno (maio 1993): Andrea A. Botelho.

POPPE, Erik, BROWN, Ross; RECKER, Jan; JOHNSON, Daniel; VANDERFEESTEN, Irene. Design and evaluation of virtual environments mechanisms to support remote collaboration on complex process diagrams. **Information systems**, v. 66, n. C, p. 59-81, June 2017.

PRENSKY, Marc. Digital natives, digital immigrants. **Blog Marcprensky.com**, USA, 2001a. Disponível em: <<https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part1.pdf>>. Acesso em: 1 nov. 2018.

PRENSKY, Marc. Do they really think differently? **Blog Marcprensky.com**, USA, 2001b. Disponível em: <<https://www.marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part2.pdf>>. Acesso em: 1 nov. 2018.

RAFALSKI, Jadson do Prado; VIEIRA JÚNIOR, Ramon Rosa Maia; SILVA, Carlos Alexandre Siqueira da. Mundos virtuais como suporte à aprendizagem: uma avaliação na implementação de projetos de aprendizagem. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 12, n. 1, 2014.

REINHARD, CarriLynn D. Virtual worlds and reception studies: comparing engagings. In: ZAGALO, Nelson; MORGADO, Leonel; BOA-VENTURA, Ana (org.). **Virtual worlds and metaverse platforms**: new communication and identity paradigms. Hershey, PA: IGI Global, 2012. p. 117-136.

RICO-ALMODOVAR, Mariano; RAMÍREZ-RODRÍGUEZ, Jaime; RIOFRÍO-LUZCANDO, Diego; BERROCAL-LOBO, Marta. A cost-effective approach for procedural training in virtual worlds. **Journal of universal computer science**, v. 23, n. 2, p. 208-232, 2017.

RYAN, Richard M.; DECI, Edward L. Intrinsic and extrinsic motivations: classic definitions and new directions. **Contemporary educational psychology**, v. 25, n. 1, p. 54-67, 2000a. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0361476X99910202>>. Acesso em: 5 jan. 2018.



- RYAN, Richard M.; DECI, Edward L. Overview of self-determination theory: an organismic-dialectical perspective. *In*: RYAN, Richard M.; DECI, Edward L. (ed.). **Handbook of self-determination research**. Rochester: University of Rochester Press, 2002. p. 3-33.
- RYAN, Richard M.; DECI, Edward L. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. **American psychologist**, v. 55, n. 1, p. 68, 2000b.
- SCALISE, Kathleen; TIMMS, Michael; MOORJANI, Anita; CLARK, Lakisha; HOLTERMANN, Karen; IRVIN, P. Shawn. Student learning in science simulations: design features that promote learning gains. **Journal of research in science teaching**, v. 48, n. 9, p. 1050-1078, 2011.
- SCHETTINO, Patrizia. Re-defining the concept of immersion in digital immersive environments. Published in 2015 Digital Heritage. *In*: SEGALLER, Stephen. **Nerds 2.0.1**. New York: TV Books, LLC, 2015.
- SCHMEIL, Andreas. **Designing collaboration experiences for 3D virtual worlds**. 2012. 226 f. Dissertation (Ph.D. in Communication Sciences) – Faculty of Communication Sciences, Università della Svizzera italiana, Lugano, 2012.
- SCOPUS. Base de dados bibliográfica. Elsevier B.V., Amsterdam, 2018. Disponível em: <<https://www2.scopus.com/search/form.uri?display=basic>>. Acesso em: 23 out. 2019.
- SGOBBI, Fabiana Santiago. **Explorando autodeterminação, utilizando novas tecnologias para ensinar autocuidado em obesos**. 2017. 206 f. Tese (Doutorado) – Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- SILVA, Patrícia Fernanda da; MENEZES, Crediné Silva de; FAGUNDES, Léa da Cruz. O processo de avaliação no desenvolvimento de projetos de aprendizagem em ambientes digitais. *In*: CICLO DE PALESTRAS SOBRE NOVAS TECNOLOGIAS NA EDUCAÇÃO, 24., 2016, Porto Alegre. **Anais [...]**. Porto Alegre: UFRGS, 2016.
- SOTO, Victor J. Which instructional design models are educators using to design virtual world instruction? **MERLOT Journal of online learning and teaching**, Long Beach, CA, v. 9, n. 3, Sept. 2013.

VOSS, Gleizer Bierhalz. **TCN – Teaching computer networks in a immersive virtual environment**. 2014. 159 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Informática, Centro de Tecnologia, Universidade Federal de Santa Maria, Santa Maria, 2014. il., 30 cm.

VOSS, Gleizer Bierhalz; NUNES, Felipe Becker; HERPICH, Fabrício; MEDINA, Roseclea Duarte. Ambientes virtuais de aprendizagem e ambientes imersivos: um estudo de caso utilizando tecnologias de computação móvel. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE, 24., 2013, Campinas. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2013. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/2479>>. Acesso em: 23 out. 2019.

WANKEL, Charles; BLESSINGER, Patrick. **Increasing student engagement and retention using immersive interfaces: virtual worlds, gaming, and simulation**. Bingley, UK: Emerald, 2012.

XENOS, Michalis Nik; MARATOU, Vicky; NTOKAS, Ioannis; METTOURIS, Christos; PAPADOPOULOS, George Angelos. Game-based learning using a 3D virtual world in computer engineering education. *In*: IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE – EDUCON, 2017, Athens, Greece. **Proceedings** [...]. USA: Institute of Electrical and Electronics Engineers – IEEE, 2017. p. 1078-1083.

ZERVAS, Panagiotis; KALIMERIS, Ioannis; SAMPSON, Demetrios G. A method for developing mobile virtual laboratories. *In*: IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES – ICALT, 14., 2014, Athens, Greece. **Proceedings** [...]. USA: Institute of Electrical and Electronics Engineers – IEEE, 2014.

# 2

## As ferramentas de autoria para criar um Mundo Virtual Imersivo

**Fabiana Santiago Sgobbi - PPGIE - UFRGS**

fabianasgobbi@gmail.com

**Fabrício Herpich - PPGIE - UFRGS**

fabricio\_herpich@hotmail.com

**Felipe Becker Nunes - AMF**

nunesfb@gmail.com

## 2.1 CRIAÇÃO DO MUNDO VIRTUAL

Do ponto de vista arquitetural, considerando o modo de operação do *OpenSim*, em termos de distribuição de recursos e acesso compartilhado, Kotsilieris e Dimopoulou (2013) explanam acerca de três alternativas, denominadas *StandAlone*, *Grid* e *Hypergrid*.

No modo *StandAlone*, cada usuário possui autorização para interagir, sendo possível criar o maior número de regiões desejado, mas apenas na mesma máquina. O modo de funcionamento está centralizado, basicamente, na execução dos serviços localmente na máquina do usuário, em que um banco de dados local é criado para salvar todas as informações e dados que são gerados no Mundo Virtual instalado.

O usuário terá a possibilidade de acessar o ambiente com suas credenciais, realizar as operações normalmente, como criação de regiões, objetos e demais elementos relacionados, sendo todas estas alterações salvas localmente no banco de dados hospedado na máquina do usuário. Outra vantagem que pode ser destacada é a portabilidade do Mundo Virtual criado na máquina de um usuário para outras máquinas, sendo possível outro usuário acessar o ambiente criado. Entretanto, torna-se importante ressaltar que esse tipo de portabilidade se refere ao uso do Mundo Virtual criado em uma máquina por outros usuários em suas próprias máquinas, seja com a utilização do diretório completo do *OpenSim* ou com a exportação/importação de uma região criada no Mundo Virtual. Como desvantagem, tem-se que não é possível o acesso compartilhado em uma mesma máquina no modo *StandAlone*, ou seja, haver vários avatares usando ao mesmo tempo o mesmo Mundo Virtual não é concebível nesta modalidade.

Já o modo **Grid** reúne um conjunto de serviços geralmente referidos como *User, Grid, Asset, Inventory e Messaging* (UGAIM), que compreendem os serviços de dados. Cada região pode ser executada em máquinas diferentes e fazer uso dos serviços UGAIM que estão hospedados em um servidor separado. Nesse tipo de modalidade, o acesso simultâneo de vários usuários a um mesmo Mundo Virtual é possibilitado, e estes usuários poderão ver os demais participantes, trocar mensagens via *chat* ou áudio, receber e enviar presentes, assim como outros elementos do ambiente, por exemplo, objetos, vestuários, entre outros. Esse fato permite que qualquer usuário com conexão à *Internet*, em qualquer local do mundo, possa acessar o visualizador e inserir suas credenciais para entrar no respectivo Mundo Virtual.

Demais vantagens dessa modalidade podem ser citadas, como a possibilidade de separação de grupos de usuários para cada região, limitando ou permitindo-lhes seu acesso a ela. Nesse contexto, cada grupo de usuários também pode ter uma série de políticas de permissão, estabelecidas de acordo com a necessidade do administrador da região no Mundo Virtual, sendo citada, como exemplo, a proibição de voar, criar objetos e desabilitar *scripts*, entre outros.

No modo **Hypergrid**, é implementada a ideia de uma *web* de Mundos Virtuais, permitindo a sua interligação por meio da *Internet*. Nesse modo, os administradores da região podem colocar *hyperlinks* dentro dos seus mapas para regiões conservadas e manipuladas por outros (KOTSILIERIS; DIMOPOULOU, 2013, p. 6). A ideia geral é a criação de

uma grande rede interligada de Mundos Virtuais na *web*, em que as credenciais são compartilhadas, para acesso liberado a essa rede, sendo uma implementação mais complexa do ponto de vista computacional.

Essa modalidade permite a oportunidade de melhorar o desempenho dos Mundos Virtuais interligados com o compartilhamento de recursos de *hardware* entre diferentes máquinas alocadas como nós. Dessa forma, tem-se uma máquina que hospeda o servidor de Mundos Virtuais centralmente e permite que sejam criadas diversas instâncias de Mundos Virtuais em diferentes máquinas interligadas na rede.

Efetuada as descrições introdutórias essenciais a este Capítulo, é importante detalhar aos leitores os aspectos referentes ao processo de criação de Mundos Virtuais em diferentes tipos de modalidades, conforme são demonstradas, a seguir, as implementações no modo *StandAlone* e Servidor.

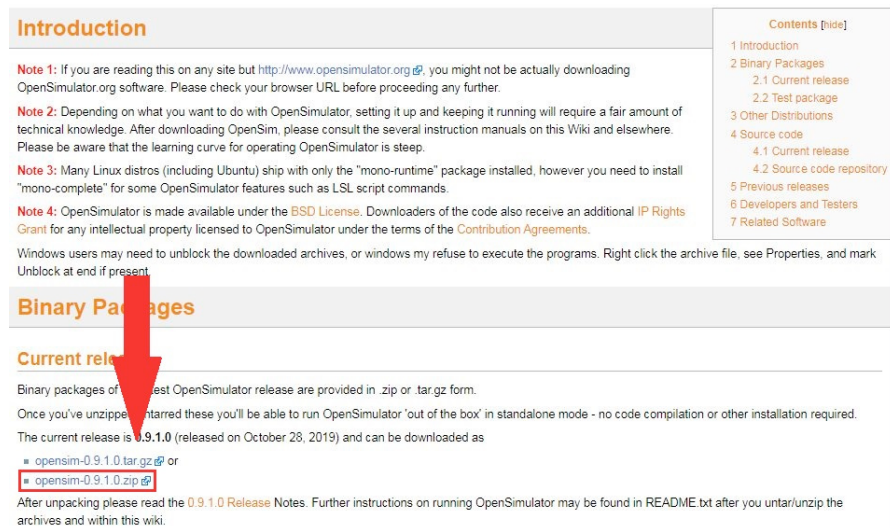
### **2.1.1 Versão *StandAlone***

Para demonstrar a instalação do Mundo Virtual *OpenSimulator* neste Capítulo, será utilizada a versão 0.9.0.1. Para realizar o *download* do *OpenSim*, acesse:

<<http://opensimulator.org/wiki/Download>>

Role a tela e selecione a opção Binary Packages. Faça *download da opção* <<http://OpenSimulator.org/dist/OpenSim-0.9.0.1.zip>>. A Figura 2.1 ilustra essa opção.

Figura 2.1 - Página oficial do *OpenSim* para *download*.



**Introduction**

**Note 1:** If you are reading this on any site but <http://www.opensimulator.org>, you might not be actually downloading OpenSimulator.org software. Please check your browser URL before proceeding any further.

**Note 2:** Depending on what you want to do with OpenSimulator, setting it up and keeping it running will require a fair amount of technical knowledge. After downloading OpenSim, please consult the several instruction manuals on this Wiki and elsewhere. Please be aware that the learning curve for operating OpenSimulator is steep.

**Note 3:** Many Linux distros (including Ubuntu) ship with only the "mono-runtime" package installed, however you need to install "mono-complete" for some OpenSimulator features such as LSL script commands.

**Note 4:** OpenSimulator is made available under the [BSD License](#). Downloaders of the code also receive an additional [IP Rights Grant](#) for any intellectual property licensed to OpenSimulator under the terms of the [Contribution Agreements](#).

Windows users may need to unblock the downloaded archives, or windows may refuse to execute the programs. Right click the archive file, see Properties, and mark Unblock at end if present.

**Contents [hide]**

- 1 Introduction
- 2 Binary Packages
  - 2.1 Current release
  - 2.2 Test package
- 3 Other Distributions
- 4 Source code
  - 4.1 Current release
  - 4.2 Source code repository
- 5 Previous releases
- 6 Developers and Testers
- 7 Related Software

**Binary Packages**

**Current release**

Binary packages of the latest OpenSimulator release are provided in .zip or .tar.gz form.

Once you've unzipped/untarred these you'll be able to run OpenSimulator 'out of the box' in standalone mode - no code compilation or other installation required.

The current release is **0.9.1.0** (released on October 28, 2019) and can be downloaded as

- [opensim-0.9.1.0.tar.gz](#) or
- [opensim-0.9.1.0.zip](#)

After unpacking please read the [0.9.1.0 Release Notes](#). Further instructions on running OpenSimulator may be found in README.txt after you untar/unzip the archives and within this wiki.

Fonte: <<http://OpenSimulator.org/wiki/Download>>.

A seguir, basta descompactá-la em seu computador. Recomenda-se que, após descompactar, seja feito o recorte dessa pasta para o seguinte caminho: "C:\OpenSim". Abra a pasta OpenSim-0.9.0.1/bin e execute o arquivo Prebuild. A Figura 2.2 apresenta essa tela.

Figura 2.2 - Execução dos arquivos de instalação do *OpenSim*.

Computador > OS (C:) > opensim > opensim-0.9.0.1 > bin

Nome	Data de modific...	Tipo	Tamanho
inventory	29/06/2018 22:07	Pasta de arquivos	
j2kDecodeCache	31/08/2018 15:19	Pasta de arquivos	
lib	29/06/2018 22:07	Pasta de arquivos	
lib32	29/06/2018 22:07	Pasta de arquivos	
lib64	29/06/2018 22:07	Pasta de arquivos	
Library	29/06/2018 22:07	Pasta de arquivos	
maptiles	31/08/2018 15:23	Pasta de arquivos	
openmetaverse_data	29/06/2018 22:07	Pasta de arquivos	
Regions	31/08/2018 15:14	Pasta de arquivos	
ScriptEngines	31/08/2018 16:59	Pasta de arquivos	
mautil	29/06/2018 22:07	Aplicativo	8 KB
OpenSim.ConsoleClient	29/06/2018 22:07	Aplicativo	10 KB
OpenSim	29/06/2018 22:07	Aplicativo	78 KB
OpenSim.Tests.Clients.AssetClient	29/06/2018 22:07	Aplicativo	7 KB
OpenSim.Tools.Configger	29/06/2018 22:07	Aplicativo	12 KB
OpenSim.Tools.IsIc	29/06/2018 22:07	Aplicativo	11 KB
OpenSim32	29/06/2018 22:07	Aplicativo	6 KB
pCampBot	29/06/2018 22:07	Aplicativo	58 KB
Prebuild	29/06/2018 22:07	Aplicativo	229 KB
Robust	29/06/2018 22:07	Aplicativo	7 KB
Robust32	29/06/2018 22:07	Aplicativo	6 KB

2

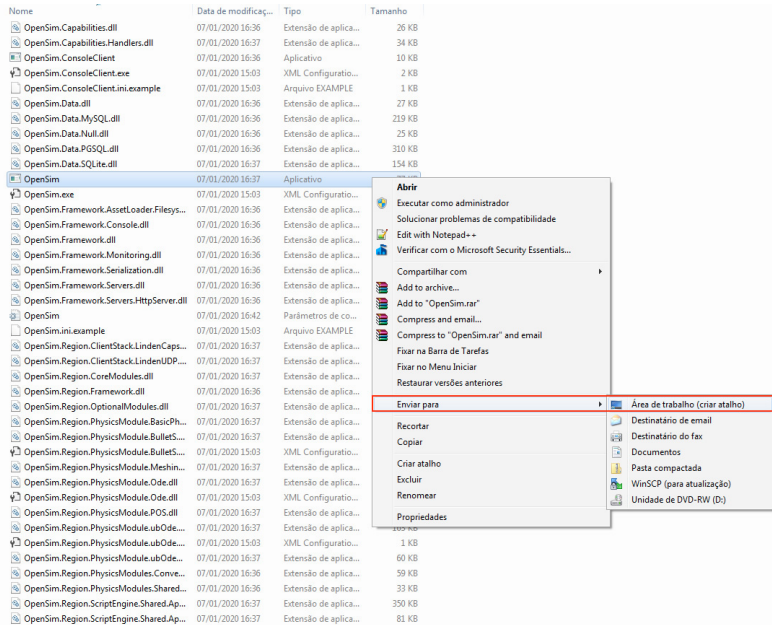
1

Uma tela será aberta e algumas ações serão realizadas. Depois ela irá se fechar automaticamente. Essa ação é necessária somente uma vez. Assim que a janela se fechar, já se pode executar o arquivo *OpenSim.exe*. Ao executar o *OpenSim* pela primeira vez, uma mensagem irá surgir, solicitando-lhe acesso à rede. Habilite a opção “redes privadas” e clique em “permitir acesso”.

Destaca-se uma dica importante: crie um atalho para o *OpenSim* em sua área de trabalho. Assim, não precisará navegar por todas essas pastas novamente para executar o Mundo Virtual. Clique com o botão direito no *OpenSim*, vá em “Enviar para” e, em seguida, “Área de trabalho (criar atalho)”, como mostra a Figura 2.3.



Figura 2.3 - Criação de um atalho para a aplicação *OpenSim*.



Após esse procedimento, inicia-se o aplicativo *OpenSim.exe*. Uma janela será aberta e será solicitado o preenchimento das configurações. Dê um nome à nova região e tecla *enter* para os demais campos. Essa tela deverá permanecer aberta enquanto o Mundo Virtual estiver em uso. Caso contrário, o Mundo Virtual será fechado. A Figura 2.4 mostra a tela.

Figura 2.4 - Tela contendo as configurações do *OpenSim*.

```
C:\Windows\system32\cmd.exe - OpenSim.exe
ertain migration commands attempt to fetch data out of old tables. However, if
you're using an existing database and you see database related errors while run
ing OpenSim then you will need to fix these problems manually. Continuing.
09:59:10 - [MIGRATIONS]: Updating FriendsStore to version 2
09:59:10 - [FRIENDS MODULE]: FriendsModule enabled
09:59:10 - [FLOTSAM ASSET CACHE]: FlotsamAssetCache enabled
09:59:10 - [FLOTSAM ASSET CACHE]: Cache Directory ./assetcache
09:59:11 - [LLLOGIN IN CONNECTOR]: Starting..
09:59:11 - [LOAD REGIONS PLUGIN]: Loading region configurations from filesystem

=====
We are now going to ask a couple of questions about your region.
You can press 'enter' without typing anything to use the default
the default is displayed between [ ] brackets.
=====

New region name []: teste
RegionUUID [2eb1d90-0b3c-4cba-9bf6-eff149281055]:
Region Location [1000,1000]:
Internal IP address [0.0.0.0]:
Internal port [9000]:
Resolve hostname to IP on start (for running inside Docker) [False]:
External host name [SYSTEMIP]:
10:00:03 - [REGIONINFO]: Resolving SYSTEMIP to 192.168.56.1 for external hostnam
e of region teste
10:00:03 - [REGION LOADER FILE SYSTEM]: Loading config files from .\Regions
10:00:03 - [REGION LOADER FILE SYSTEM]: Loading config file .\Regions\Regions.in
i
```

Em seguida, será solicitado o *login* de administrador. Entre com seu nome (*first name*), sobrenome (*last name*) e uma senha. Para os campos *e-mail* e *user ID*, apenas tecle *enter*, pois eles não são obrigatórios. Com isso, o Mundo Virtual na modalidade *StandAlone* estará pronto para ser utilizado, restando apenas realizar os procedimentos de instalação do visualizador (*viewer*). A Figura 2.5 mostra a tela.

Figura 2.5 - Finalização do processo de configuração do *OpenSim*.

```
C:\Windows\system32\cmd.exe - opensim.exe
10:43:58 - [MODULE COMMANDE]: Script engine found, module active
10:43:58 - [LLUDPSERVER]: Starting inbound packet processing for the LLUDP server
10:43:58 - [UDPBASE]: Starting inbound UDP loop
10:43:58 - [UDPBASE]: SIO_UDP_CONNRESET flag set
10:43:58 - [WATCHDOG]: Started tracking thread Incoming Packets (teste), ID 19
10:43:58 - [LLUDPSERVER]: Starting outbound packet processing for the LLUDP server
10:43:58 - [UDPBASE]: Starting outbound UDP loop
10:43:58 - [WATCHDOG]: Started tracking thread Outgoing Packets (teste), ID 20
10:43:58 - [WATCHDOG]: Started tracking thread GetCapsAssetWorker0, ID 21
10:43:58 - [WATCHDOG]: Started tracking thread GetCapsAssetWorker1, ID 22
10:43:58 - [WATCHDOG]: Started tracking thread GetCapsAssetWorker2, ID 23
10:43:58 - [WATCHDOG]: Started tracking thread InventoryWorkerThread0, ID 24
10:43:58 - [WATCHDOG]: Started tracking thread InventoryWorkerThread1, ID 25
10:43:58 - [SCENE]: Secure permissions loading enabled, modules loaded: Default
PermissionsModule
Estate admin has no owner set.
Estate owner first name [Test]: admin
Estate owner last name [User]: admin
Password:
User ID [a2b9b9f3-5a03-40c3-96f0-dc5de7725532]:
10:44:21 - [AUTHENTICATION DB]: Set password for principalID a2b9b9f3-5a03-40c3-
96f0-dc5de7725532
10:44:21 - [GRID SERVICE]: GetDefaultRegions returning 0 regions
10:44:21 - [USER ACCOUNT SERVICE]: Unable to set home for groupID admin admin.
10:44:24 - [USER ACCOUNT SERVICE]: Created user inventory for admin admin.
10:44:24 - [USER ACCOUNT SERVICE]: Creating default appearance items for a2b9b9f3-
5a03-40c3-96f0-dc5de7725532
```

Para esta explicação, foi escolhido o uso do visualizador *Singularity*, cujo *download* deve ser realizado em: <<http://www.singularityviewer.org/downloads>>.

Realize o *download* do arquivo *Windows 32-bit* e o instale. A Figura 2.6, a seguir, mostra a tela de *download*.

Figura 2.6 - Página oficial de *download* do visualizador *Singularity*.



Fonte: <<http://www.singularityviewer.org/downloads>>.

Selecione “sim”, caso uma mensagem de segurança do *Windows* apareça. Após a instalação, clique no ícone para abrir o *Singularity*, que deve ter sido criado em sua Área de Trabalho.

Atenção: se aparecer uma mensagem de que está faltando MS-VCP100.dll no seu computador, baixe e instale o programa Microsoft Visual C++ 2010 Redistributable Package (x86):

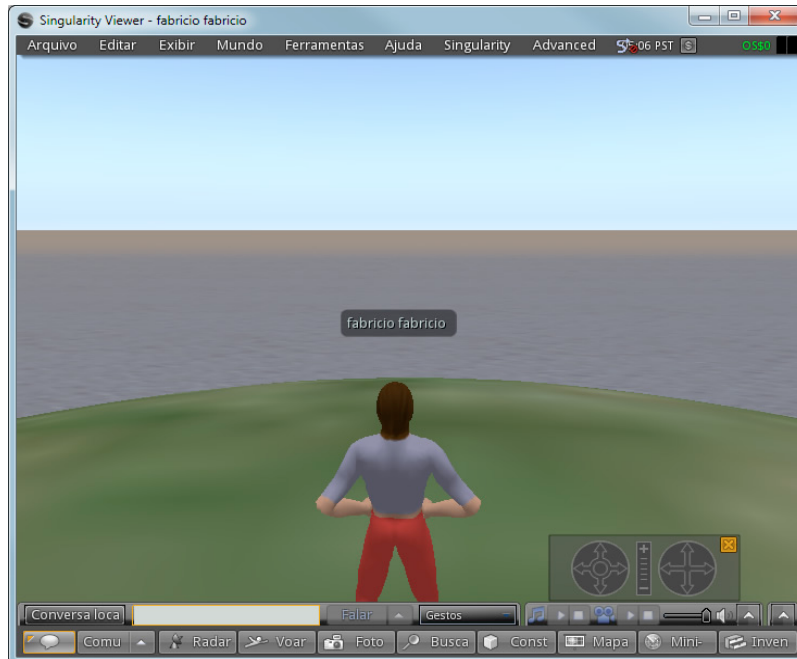
<<https://www.microsoft.com/en-in/download/details.aspx?id=5555>>

Depois de instalar esse *software*, tente novamente instalar o *Singularity*. Na tela inicial do *Singularity*, digite:

- Nome: nome [espaço] sobrenome com que se cadastrou.
- Senha: a senha que foi cadastrada para esse nome e sobrenome.
- Selecione a opção: Lembre-se de manter essas informações em seu computador.
- No campo Grelha, escolha a opção *Local Host*.
- Clique em Entrar.

Pronto! Está no Mundo Virtual. A Figura 2.7, a seguir, mostra o Mundo Virtual funcionando.

Figura 2.7 - Tela inicial do Mundo Virtual no *OpenSim*.



Pode-se criar objetos e realizar as alterações desejadas. Pode-se fazer o *download* de uma região já construída (arquivo.oar), para que não se precise iniciar do zero.

## 2.1.2 Versão Servidor

Para demonstrar a instalação do Mundo Virtual *OpenSimulator* neste Capítulo, será utilizada a versão 0.9.0.1. Para realizar o *download*, acesse:

<<http://OpenSimulator.org/wiki/Download>>

Selecione a opção “OpenSim-0.9.0.1-source.zip”.

O *download* da versão correta (“OpenSim-0.9.0.1-source.zip”) se encontra quase no final da página, em “SOURCE CODE”! (Figuras 2.8 e 2.9). Posteriormente, basta escolher a pasta de destino, em seu computador, para salvar e descompactar o arquivo. Recomenda-se que essa pasta seja copiada para o caminho: “C:”

Figura 2.8 - Escolha da versão e *download* do *OpenSim*.

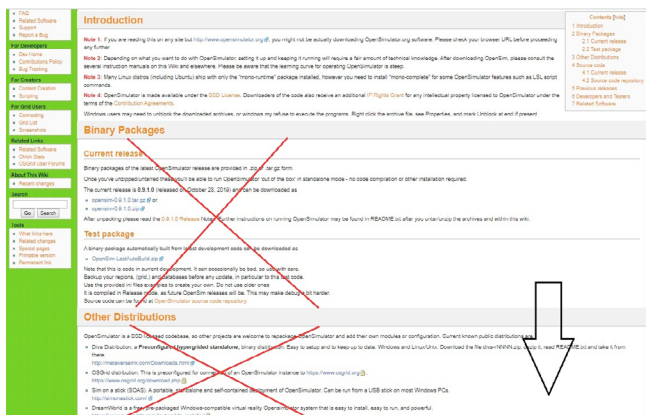
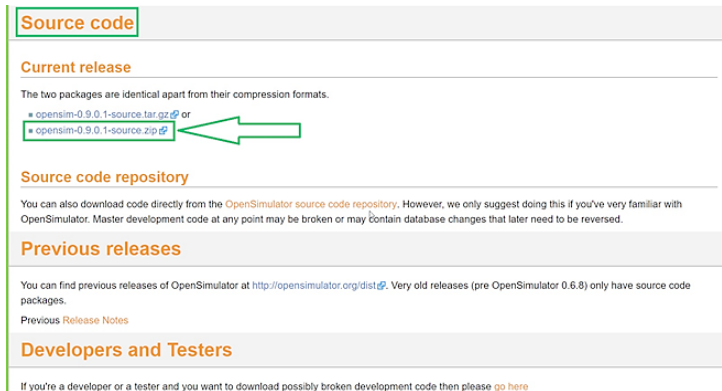
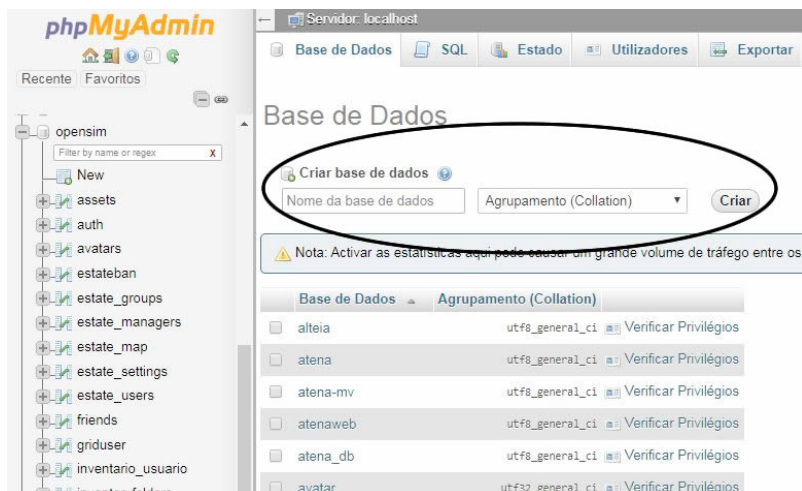


Figura 2.9 - Indicação da versão do *OpenSim* para *download*.



Antes de realizar a configuração do *OpenSim*, a primeira etapa a ser realizada envolve a criação do banco de dados para o Mundo Virtual. Dessa forma, é necessário acessar o seguinte endereço no seu navegador: <http://localhost:8080/phpmyadmin> ou somente <http://localhost/phpmyadmin> (para acessar os endereços acima citados, o *WampServer* deve ser inicializado e os serviços Apache e MySQL executados).

Figura 2.10 - Criando o Banco de Dados.



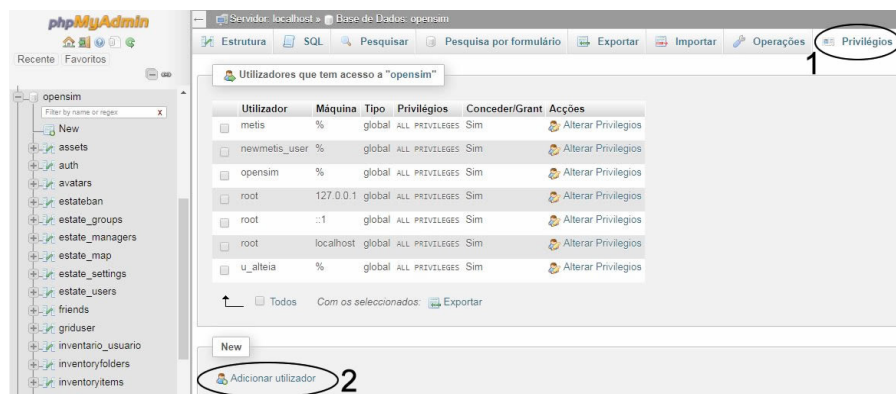
Será aberta a página do *phpMyAdmin*, na qual se deve criar o banco de dados para o *OpenSim*. Nessa página inicial, em seu centro, está a opção para criar um novo banco de dados (Figura 2.10). Na caixa de texto, digite o nome do banco de dados a ser criado: “*OpenSim*”. Altere a opção ao lado, de “Collation” para “*utf8\_general\_ci*”. Clique em “Criar”. Uma mensagem aparecerá: “Banco de dados *moodle* foi criado” (Figura 2.11).

Figura 2.11 - Criando o Banco de Dados.



Nessa mesma página, clique na aba “Privilégios” e clique em “Adicionar novo usuário” (Figura 2.12).

Figura 2.12 - Adicionando novo usuário.





Preencha as informações, nos campos solicitados, conforme orientações (Figuras 2.13, 2.14 e 2.15):

- “Informações de login”
- Nome do usuário: [Usar campo texto]: *OpenSim*
- Servidor: [Local]: localhost
- Senha: 12345678
- Redigite: 12345678

Figura 2.13 - Informações de login.

Adicionar utilizador

Informação de Login

User name: Usar campo de texto: opensin

Host: Local localhost

Palavra-passe: Usar campo de texto: .....

Confirma: .....

Gerar palavra-passe: Gerar

Na opção “Banco de Dados para usuário”, selecione a opção: Conceder todos os privilégios no banco de dados “*OpenSim*” (Figura 2.14).

Figura 2.14 - Banco de dados para usuário.

Database for user

Create database with same name and grant all privileges.

Grant all privileges on wildcard name (username\\_%).

Grant all privileges on database "opensim".



Acerca dos “Privilégios Globais”, selecione a opção: “Marcar todos” (Figura 2.15). Para finalizar a criação do usuário, clique em “Executar”. Uma mensagem aparecerá com a informação “Adicionado usuário” (Figura 2.16).

Figura 2.15 - Privilégios globais.

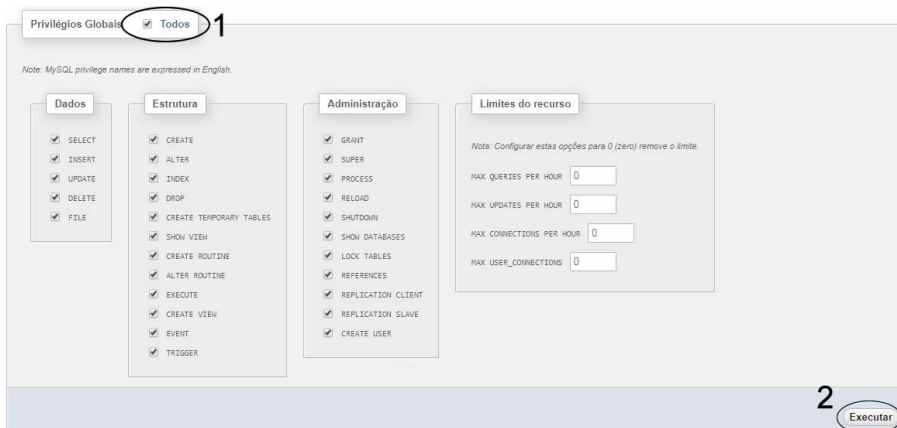
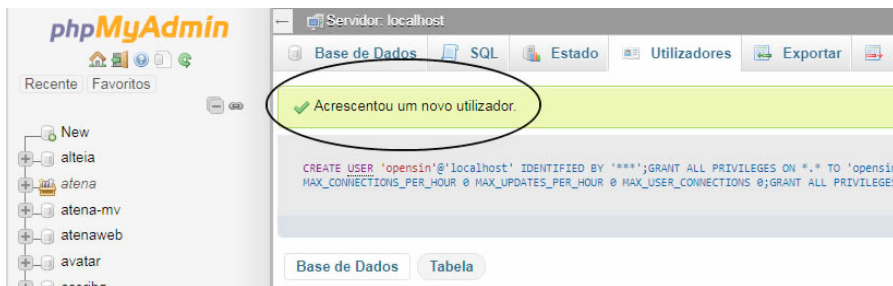
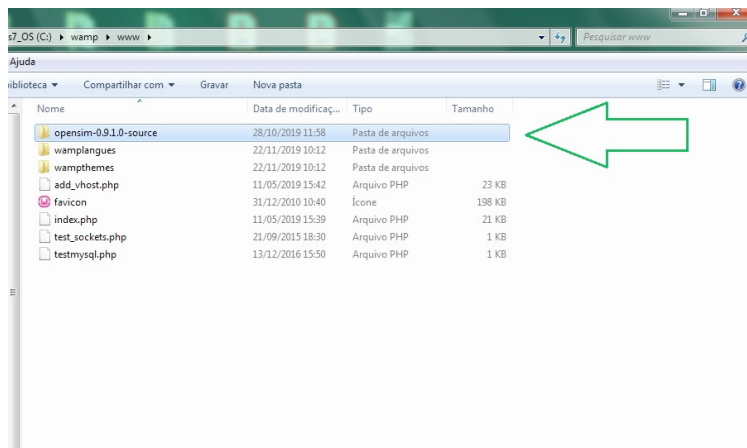


Figura 2.16 - Mensagem de confirmação.



Depois disso, caso ainda não o tenha feito, localize a pasta do OpenSim e copie-a para o caminho: “C:\wamp\www\OpenSim-0.9.0.1-source” (Figura 2.17).

Figura 2.17 - Copiando o *OpenSim* para o WampServer.



Abra a pasta copiada e execute o arquivo *runprebuild* (Figura 2.18). Será aberta uma tela em que algumas ações serão realizadas e depois ela se fechará automaticamente (Figura 2.19).

Figura 2.18 - Executando o arquivo “runprebuild”.

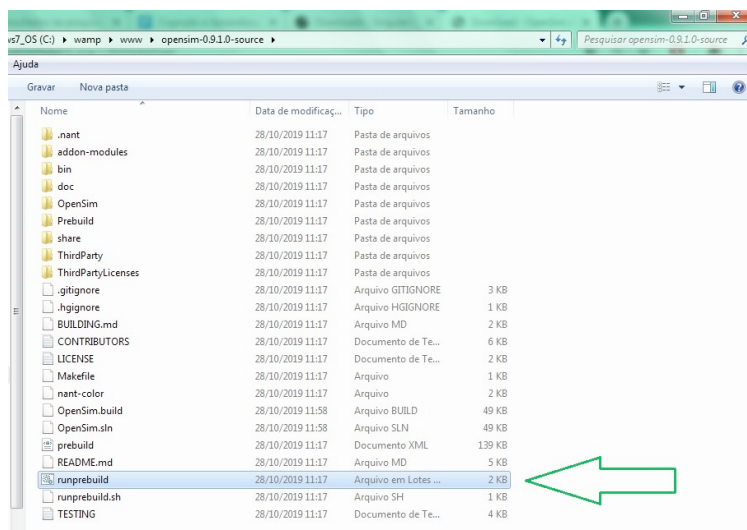
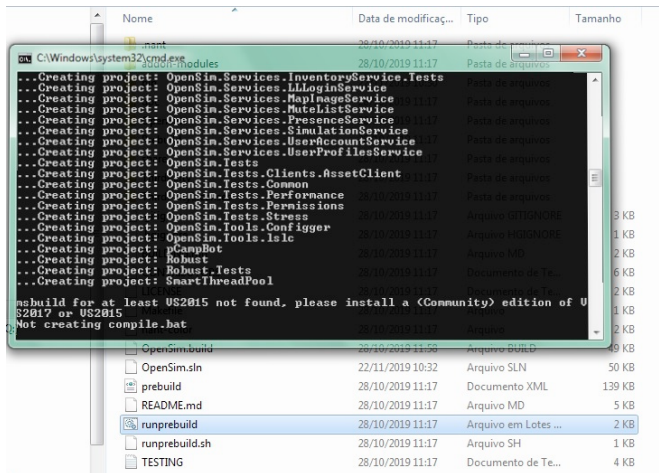


Figura 2.19 - Executando o arquivo “runprebuild”.



Nota-se que alguns novos arquivos foram adicionados na pasta, sendo um deles o “compile”. Portanto, execute esse arquivo também (Figura 2.20). Uma tela será aberta, algumas ações serão realizadas, e ela irá se fechar automaticamente (Figura 2.21).

Figura 2.20 - Novos arquivos adicionados.

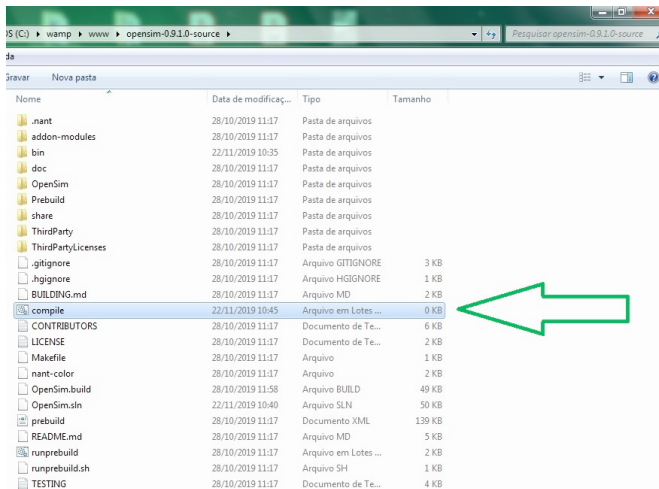


Figura 2.21 - Executando o arquivo “compile”.

```
C:\WINDOWS\system32\cmd.exe
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Microsoft.Common.targets(1605,5): warning MSB3270: There was a mismatch
between the processor architecture of the project being built "MSIL" and the processor architecture of the reference "
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorlib.dll", "AMD64". This mismatch may cause runtime failures. Pleas
e consider changing the targeted processor architecture of your project through the Configuration Manager so as to align
the processor architectures between your project and references, or take a dependency on references with a processor
architecture that matches the targeted processor architecture of your project. [C:\wamp64\www\opensim-0.9.0.1-source\Op
enSim\Framework\Console\OpenSim.Framework.Console.csproj]
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Microsoft.Common.targets(1605,5): warning MSB3270: There was a mismatch
between the processor architecture of the project being built "MSIL" and the processor architecture of the reference "
System.Web", "AMD64". This mismatch may cause runtime failures. Please consider changing the targeted processor archite
cture of your project through the Configuration Manager so as to align the processor architectures between your project
and references, or take a dependency on references with a processor architecture that matches the targeted processor archi
tecture of your project. [C:\wamp64\www\opensim-0.9.0.1-source\OpenSim\Framework\Console\OpenSim.Framework.Console.
csproj]
GenerateTargetFrameworkMonikerAttribute:
Skipping target "GenerateTargetFrameworkMonikerAttribute" because all output files are up-to-date with respect to the i
nput files.
CoreCompile:
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Csc.exe /noconfig /unsafe /checked- /nowarn:1701,1702 /nostdlib+ /p1
atform:AnyCPU /warn:4 /baseaddress:285212672 /define:TRACE;DEBUG /highentropyva- /reference:C:\wamp64\www\opensim-0.9
.0.1-source\bin\log4net.dll /reference:C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscorlib.dll /reference:C:\wam
p64\www\opensim-0.9.0.1-source\bin\Mini.dll /reference:C:\wamp64\www\opensim-0.9.0.1-source\bin\OpenMetaverseTypes.dl
1 /reference:C:\wamp64\www\opensim-0.9.0.1-source\bin\OpenSim.Framework.dll /reference:C:\wamp64\www\opensim-0.9.0.1-
source\bin\OpenSim.Framework.Servers.HttpServer.dll /reference:C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Core
V4.0.4.0_0_0_77735C561934E089\System.Core.dll /reference:C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.V4.0_4_0_0_
_b77a5c561934e089\System.dll /reference:C:\WINDOWS\Microsoft.Net\assembly\GAC_64\System.Web.V4.0_4_0_0_0_b69f5f
741d58a3a\System.Web.dll /reference:C:\WINDOWS\Microsoft.Net\assembly\GAC_MSIL\System.Xml.V4.0_4_0_0_0_b77a5c561934
e089\System.Xml.dll /debug+ /filealign:4096 /optimize- /out:obj\Debug\OpenSim.Framework.Console.dll /target:library /
warnaserror- /utf8output assemblyInfo.cs CommandConsole.cs ConsoleBase.cs ConsoleDisplayList.cs ConsoleDisplayTable.cs
s ConsoleDisplayUtil.cs ConsoleLogCommand.cs ConsoleUtil.cs LocalConsole.cs MockConsole.cs OpenSimAppender.cs Remo
```

Após o arquivo terminar de executar, acesse a pasta “bin”, para abrir o arquivo “OpenSim.ini”. O tipo desse arquivo é chamado “Parâmetros de Configuração” (Figura 2.22). Então, utilize o Notepad++ para editar o arquivo (Figura 2.23).

- C:\OpenSim-0.9.0.1-source\bin

Figura 2.22 - Arquivo de configuração “OpenSim.ini”.

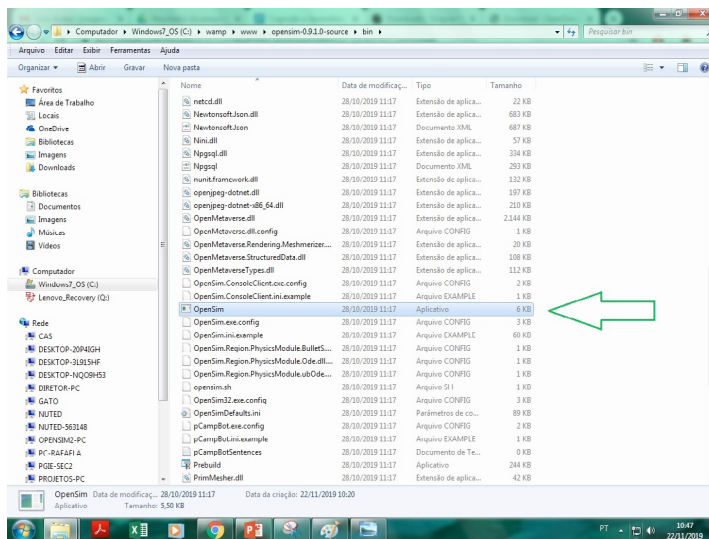
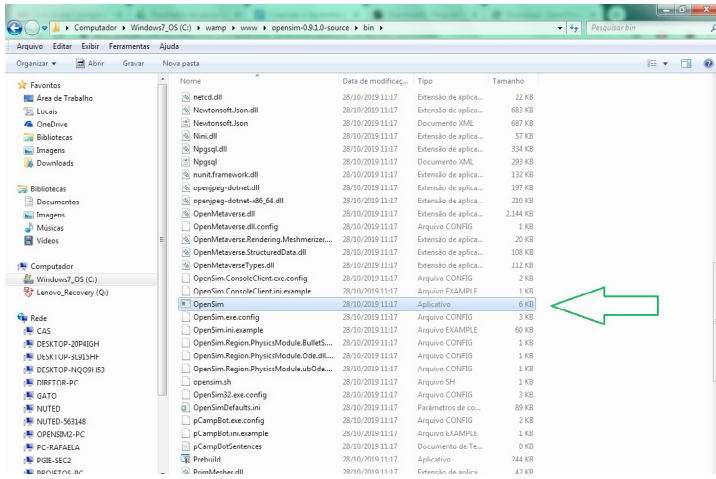


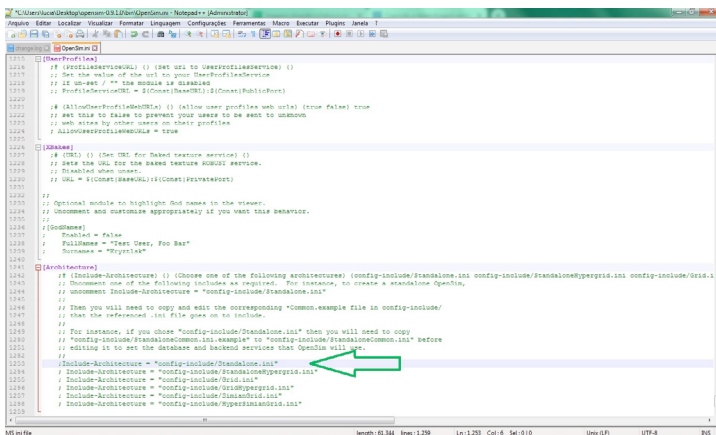
Figura 2.23 - Editando o arquivo de configuração “OpenSim.ini”.



Com o arquivo aberto, localize a linha 1253 (essa numeração pode variar conforme a versão utilizada):

- Include-Architecture = “config-include/Standalone.ini” e comente esta linha (inserir um ponto e vírgula antes).
- ; Include-Architecture = “config-include/Standalone.ini” (Figura 2.24).

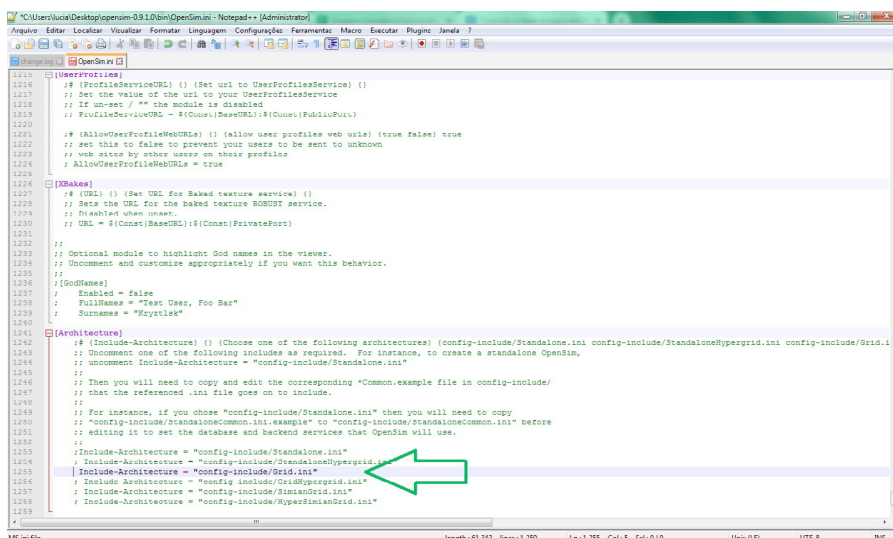
Figura 2.24 - Comentando a linha 1253.



Agora localize a linha 1255 (essa numeração pode variar conforme a versão utilizada):

- ; Include-Architecture = "config-include/Grid.ini" e retire o comentário (ponto e vírgula existente no início da linha):
- Include-Architecture = "config-include/Grid.ini" (Figura 2.25).

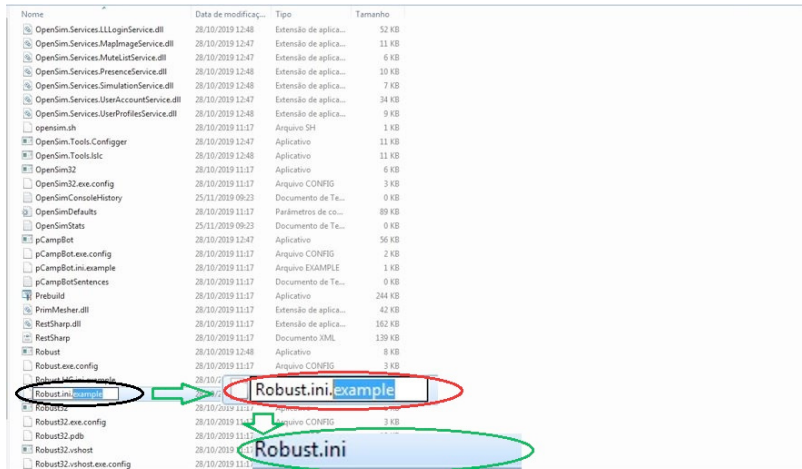
Figura 2.25 - Retirando comentário da linha 1255.



```
1216 [UserProfiles]
1217 {
1218   [ProfileServiceURL] {} (Set url to UserProfilesService) {}
1219   // Set the value of the url to your UserProfilesService
1220   // If un-set /* the module is disabled
1221   // ProfileServiceURL = ${Const\BaseURL}:${Const\PublicPort}
1222   // AllowUserProfileWebURLs {} (allow user profile web url) (true false) true
1223   // set this to false to prevent your users to be sent to unknown
1224   // web site by other users on those profiles
1225   // AllowUserProfileWebURLs = true
1226 }
1227
1228 [Robust]
1229 {
1230   [URL] {} (Set URL for Baked texture service) {}
1231   // Set the URL for the baked texture ROBUST service.
1232   // Disabled when unset.
1233   // URL = ${Const\BaseURL}:${Const\PrivatePort}
1234 }
1235
1236 //
1237 // Optional module to highlight God names in the viewer.
1238 // Uncomment and customise appropriately, if you want this behavior.
1239 {
1240   [GodNames]
1241   {
1242     Enabled = false
1243     FullNames = "Test User, Foo Bar"
1244     Surname = "Mystique"
1245   }
1246 }
1247
1248 [Architecture]
1249 {
1250   [Include-Architecture] {} (Choose one of the following architectures) (config-include/Standalone.ini config-include/StandaloneHypergrid.ini config-include/Grid.i
1251   // Uncomment one of the following includes as required. For instance, to create a standalone OpenSim,
1252   // uncomment Include-Architecture = "config-include/Standalone.ini"
1253   //
1254   // Then you will need to copy and edit the corresponding "Common.example file in config-include/
1255   // that the referenced .ini file goes on to include.
1256   //
1257   // For instance, if you chose "config-include/Standalone.ini" then you will need to copy
1258   // "config-include/StandaloneCommon.ini.example" to "config-include/StandaloneCommon.ini" before
1259   // editing it to set the database and backend services that OpenSim will use.
1260   //
1261   //
1262   // Include-Architecture = "config-include/Standalone.ini"
1263   // Include-Architecture = "config-include/StandaloneHypergrid.ini"
1264   // Include-Architecture = "config-include/Grid.ini"
1265   // Include-Architecture = "config-include/GridHypergrid.ini"
1266   // Include-Architecture = "config-include/StandGrid.ini"
1267   // Include-Architecture = "config-include/HyperSimandGrid.ini"
1268 }
1269 }
```

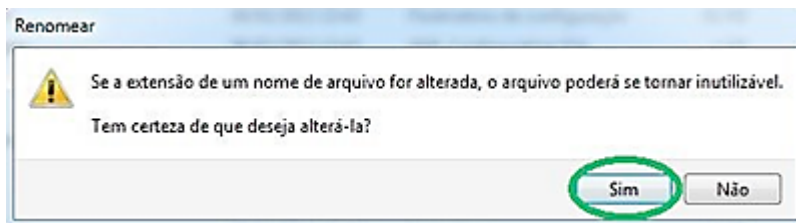
Salve o arquivo e feche-o. Localize, agora, dentro da pasta "bin", o arquivo denominado Robust.ini.example e mude o seu nome, retirando o ponto e a palavra *example*, ficando o seu nome assim: Robust.ini (Figura 2.26).

Figura 2.26 - Renomeando o arquivo de configuração “Robust.ini”.



Uma mensagem será exibida, perguntando se tem certeza de que deseja alterar o nome do arquivo; clique em Sim. Agora a extensão deste arquivo é do mesmo tipo do arquivo OpenSim.ini, que foi modificado anteriormente (Figura 2.27).

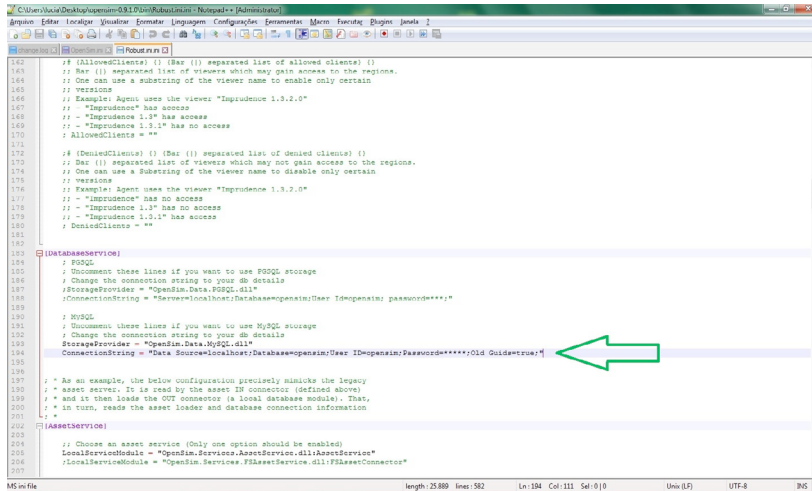
Figura 2.27 - Mensagem de confirmação.



Abra este arquivo (“Robust.ini”) pelo Notepad++ e localize a linha 194, que, por padrão, deverá estar assim:

- **ConnectionString = "Data Source = localhost; Database=OpenSim; User ID=OpenSim; Password=\*\*\*\*; Old Guids=true;"**

**Figura 2.28 - Configuração do Robust.ini.**



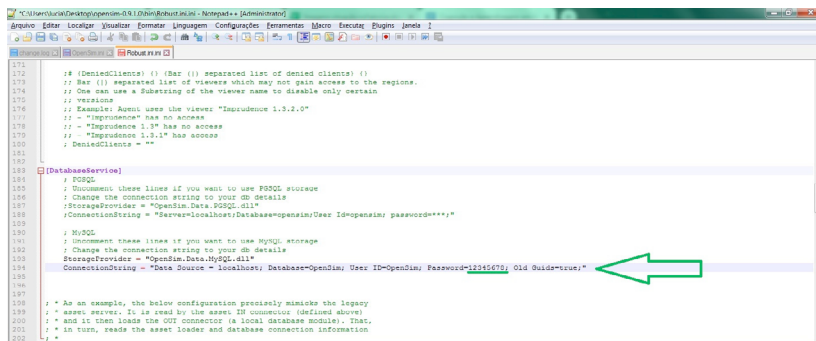
```
152 // (AllowedClients) {} (See {} separated list of allowed clients) {}
153 // See {} separated list of viewers which may gain access to the regions.
154 // One can use a substring of the viewer name to enable only certain
155 // viewers
156 // Example: Agent uses the viewer "Imprudence 1.3.2.0"
157 // "Imprudence" has access
158 // - "Imprudence 1.3*" has access
159 // - "Imprudence 1.3.1*" has no access
160 // AllowedClients = ""
161
162 // (DeniedClients) {} (See {} separated list of denied clients) {}
163 // See {} separated list of viewers which may not gain access to the regions.
164 // One can use a substring of the viewer name to disable only certain
165 // viewers
166 // Example: Agent uses the viewer "Imprudence 1.3.2.0"
167 // "Imprudence" has no access
168 // - "Imprudence 1.3*" has no access
169 // - "Imprudence 1.3.1*" has access
170 // DeniedClients = ""
171
172 [DatabaseService]
173 // PGSQL
174 // Uncomment these lines if you want to use PGSQL storage
175 // Change the connection string to your db details
176 //StorageProvider = "OpenSim.Data.PGSQL.dll"
177 //ConnectionString = "Server=localhost;Database=openSim;User ID=openSim;Password=****;"
178
179 // MySQL
180 // Uncomment these lines if you want to use MySQL storage
181 // Change the connection string to your db details
182 //StorageProvider = "OpenSim.Data.MySQL.dll"
183 //ConnectionString = "Data Source=localhost;Database=openSim;User ID=openSim;Password=****;Old Guids=true;"
184
185 // * As an example, the below configuration precisely mimicks the legacy
186 // * asset server. It is read by the asset IF connector (defined above)
187 // * and it then loads the OUI connector (a local database module). That,
188 // * in turn, reads the asset loader and database connection information
189 // *
190
191 [AssetService]
192 // Choose an asset service (Only one option should be enabled)
193 LocalServiceModule = "OpenSim.Services.AssetService.dll;AssetService"
194 //LocalServiceModule = "OpenSim.Services.FSAssetService.dll;FSAssetConnector"
```

Modifique essa linha, inserindo as informações do banco de dados que foi criado anteriormente, ou seja, insira a senha:

- **ConnectionString = "Data Source = localhost; Database=OpenSim; User ID=OpenSim; Password=12345678; Old Guids=true;"**



Figura 2.29 - Configuração do Robust para conexão com o banco de dados.



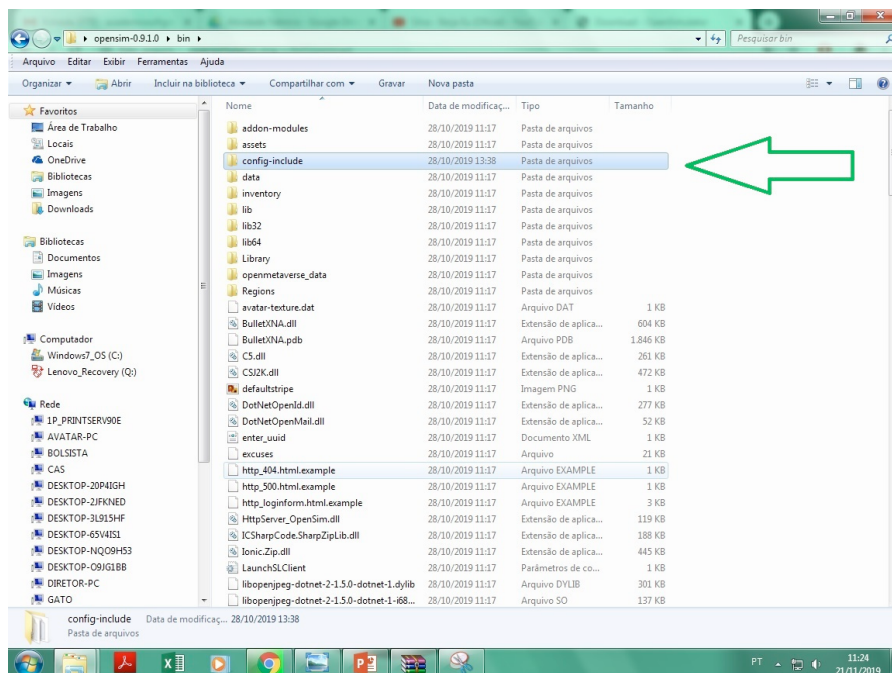
```
272  <!-- (DeniedClients) --> (Max ()) separated list of denied clients) (
273  // Max () separated list of viewers which may not gain access to the engine.
274  // One can use a substring of the viewer name to disable only certain
275  // viewers.
276  // Example: Agent uses the viewer "Impudence 1.3.2.0"
277  // - "Impudence" has no access
278  // - "Impudence 1.3" has no access
279  // - "Impudence 1.3.1" has access
280  <!-- DeniedClients = ""
281
282
283  <!-- [DatabaseService]
284  <!-- PostgreSQL
285  <!-- Uncomment these lines if you want to use PostgreSQL storage
286  <!-- Change the connection string to your db details
287  <!--StorageProvider = "OpenSim.Data.Postgres"
288  <!--ConnectionString = "Server=localhost;Database=opensim;User ID=opensim; password=***"
289
290  <!-- MySQL
291  <!-- Uncomment these lines if you want to use MySQL storage
292  <!-- Change the connection string to your db details
293  <!--StorageProvider = "OpenSim.Data.MySQL"
294  <!--ConnectionString = "Data Source = localhost; Database=OpenSim; User ID=OpenSim; Password=12345678; Old Guide=true"
295
296
297
298
299  * As an example, the below configuration precisely mimicks the legacy
300  * asset server. It is read by the asset ZIP connector (defined above)
301  * and it then loads the ODF connector (a local database module). That,
302  * in turn, reads the asset loader and database connection information
303  *
```

Salve e feche o arquivo.

Depois disso realizado, dentro da pasta “bin”, está localizada outra pasta, denominada “config-include”.

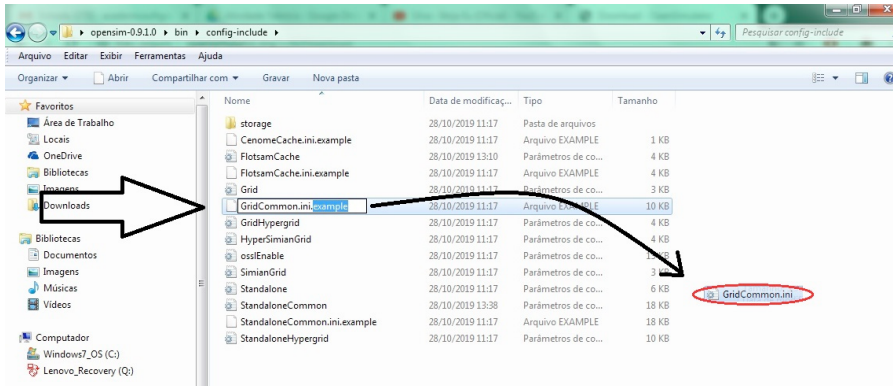
- C:\OpenSim-0.9.0.1-source\bin\config-include

Figura 2.30 - Pasta “config-include”.



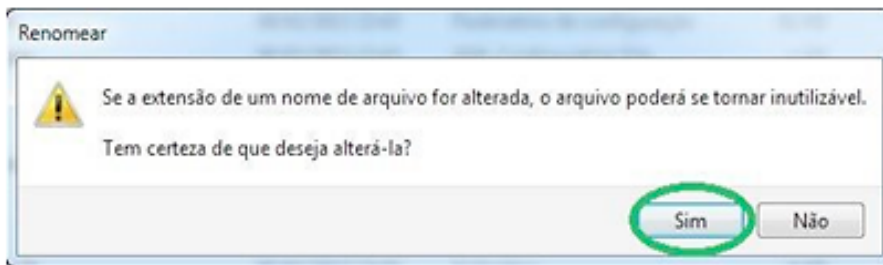
Abra essa pasta e selecione o arquivo chamado “GridCommon.ini.example”, realizando o mesmo processo feito anteriormente para o arquivo Robust.ini, ou seja, mude o nome deste arquivo, retirando o ponto e a palavra example, ficando o seu nome assim: GridCommon.ini (Figura 2.31).

Figura 2.31 - Renomeando o arquivo de configuração “GridCommon.ini”.



Uma mensagem será exibida, perguntando se tem certeza de que deseja alterar o nome do arquivo; clique em “Sim”.

Figura 2.32 - Mensagem de confirmação.



Utilizando o Notepad++, abra o arquivo denominado GridCommon.ini e localize a linha 9:

- Include-Storage = “config-include/storage/SQLiteStandalone.ini”;

A seguir, comente essa linha (inserindo um ponto e vírgula):

- ;Include-Storage = “config-include/storage/SQLiteStandalone.ini”;

Figura 2.33 – Comentando a linha 9.

```
8 ; SQLite
9 ;Include-Storage = "config-include/storage/SQLiteStandalone.ini";
10
```

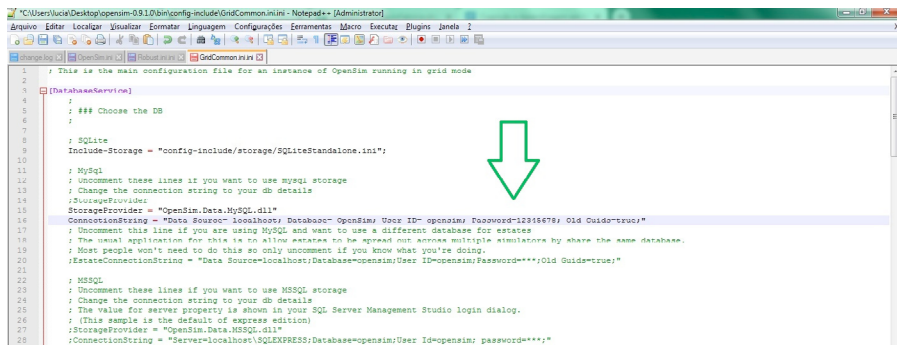
Localize as linhas 15 e 16:

- ;StorageProvider = "OpenSim.Data.MySQL.dll"
- ;ConnectionString = "Data Source= localhost; Database= OpenSim; User ID= opensim; Password= \*\*\*; Old Guids=true;"

Retire o comentário dessas linhas (removendo o ponto e vírgula no início de cada sentença). Além disso, na linha 16, retire os asteriscos (\*\*\*) existentes depois da palavra "Password", inserindo a senha:

- StorageProvider = "OpenSim.Data.MySQL.dll"
- ConnectionString = "Data Source= localhost; Database= OpenSim; User ID= opensim; Password=12345678; Old Guids=true;"

Figura 2.34 - Retirando o comentário da linha 15 e 16.



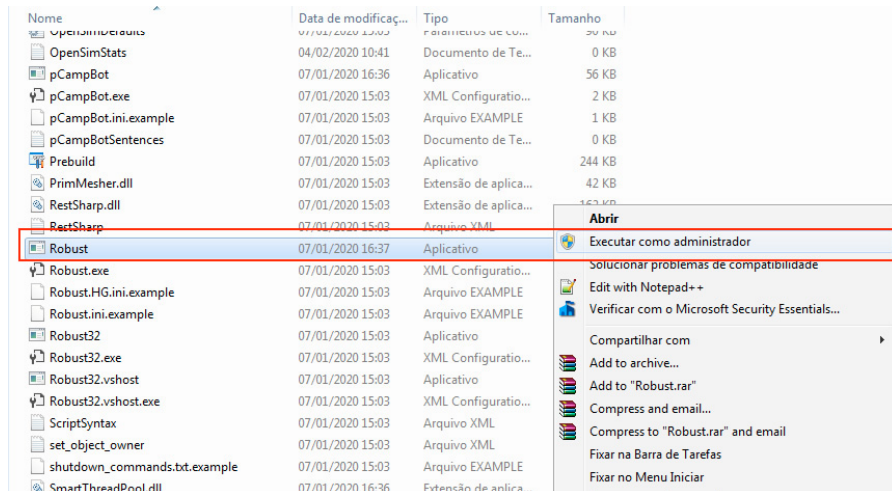
- Salve o arquivo e feche o Notepad++.

A configuração agora está quase finalizada. Volte para a pasta "bin".

- C:\OpenSim-0.9.0.1-source\bin

Clique com o botão direito do *mouse* no arquivo “Robust.exe”, cujo tipo é denominado “Aplicativo”, e selecione a opção “Executar como administrador” (Figura 2.35). Uma mensagem poderá ser exibida. Caso seja, clique na opção “Sim”.

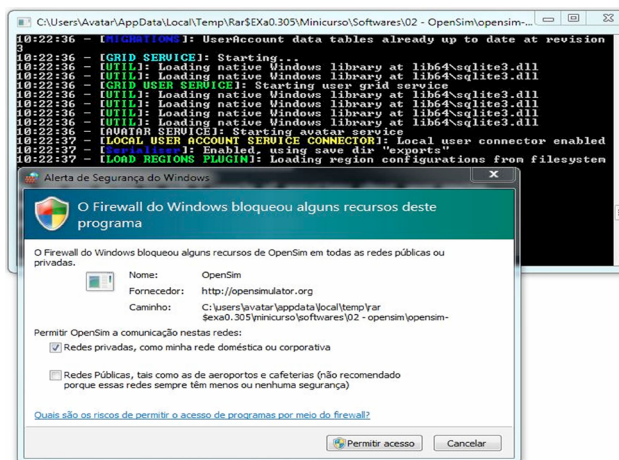
Figura 2.35 - Executando o arquivo “Robust.exe”.



Uma janela será aberta e o programa começará a rodar. Uma mensagem irá surgir solicitando acesso à rede. Marque a opção “redes privadas” e clique em “permitir acesso”. O seu antivírus também poderá realizar uma verificação (Figura 2.36).

Figura 2.36 - Verificação durante a execução do “Robust.exe”.

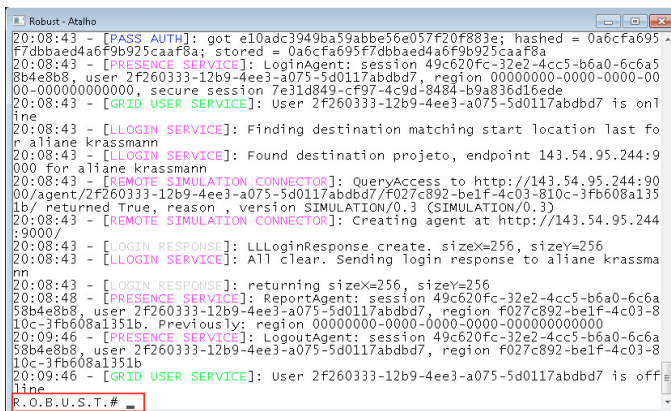
Após o programa terminar de rodar os seus processos, ele ficará



com o seguinte texto:

- R.O.B.U.S.T.#

Figura 2.37 - Término da execução do “Robust.exe”.

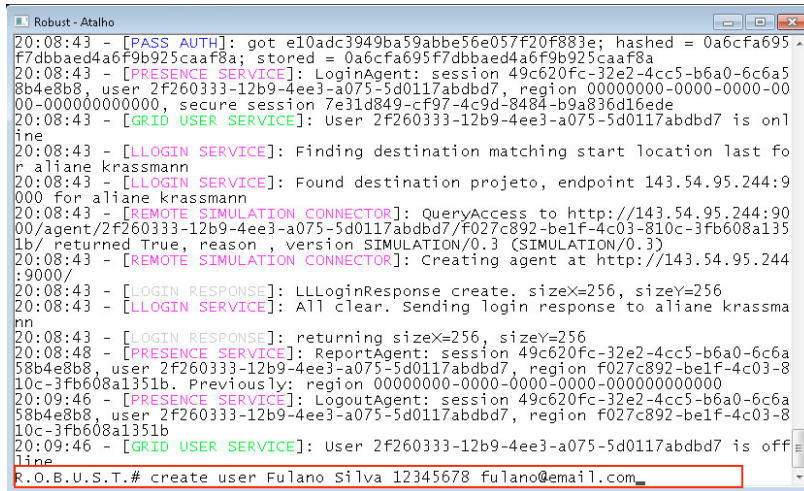


Digite dentro da janela o comando para criar um usuário no Mundo Virtual. Siga o formato:

- create user primeironome ultimônimo senha email

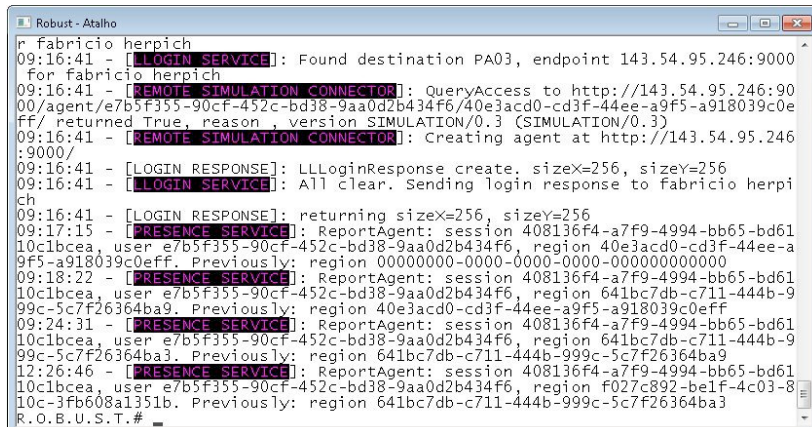
Exemplo: create user seuNome seuSobrenome suaSenha seuEmail@provedor.com (Figura 2.38).

Figura 2.38 - Criando um usuário.



Dê um **enter**, outro campo aparecerá, dê mais um **enter**. O usuário, então, será criado, exibindo a mensagem de confirmação.

Figura 2.39 - Usuário criado com sucesso.

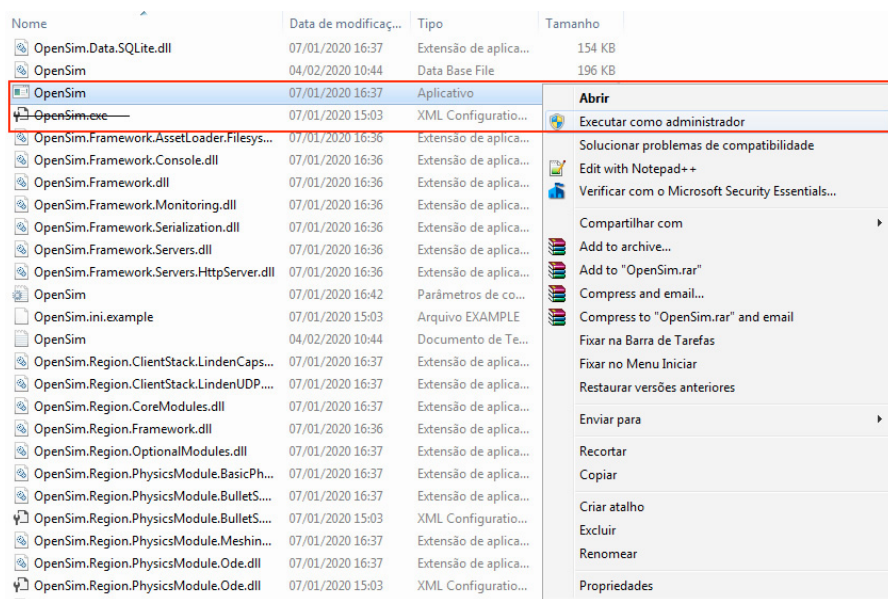




Agora, deixe a janela do Robust.exe aberta e volte para a pasta C:\OpenSim-0.9.0.1-source\bin.

Dentro da pasta, clique com o botão direito do mouse no arquivo “OpenSim.exe”, cujo tipo é denominado “Aplicativo”, e selecione a opção “Executar como administrador” (Figura 2.40). Uma mensagem poderá ser exibida. Clique na opção “Sim”. O seu antivírus também poderá solicitar autorização para continuar a execução.

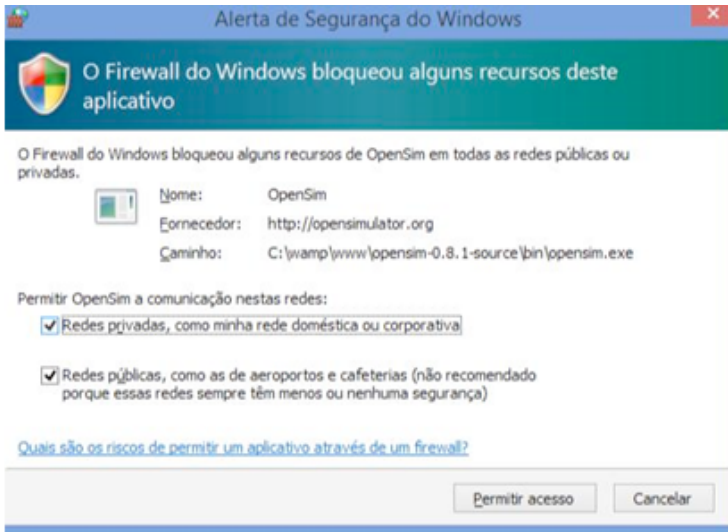
Figura 2.40 - Executando o aplicativo OpenSim.



Uma janela será aberta e o programa “OpenSim.exe” começará a rodar. Uma mensagem irá surgir solicitando acesso à rede; marque a opção “Redes privadas...” e clique em “Permitir acesso” (Figura 2.41).

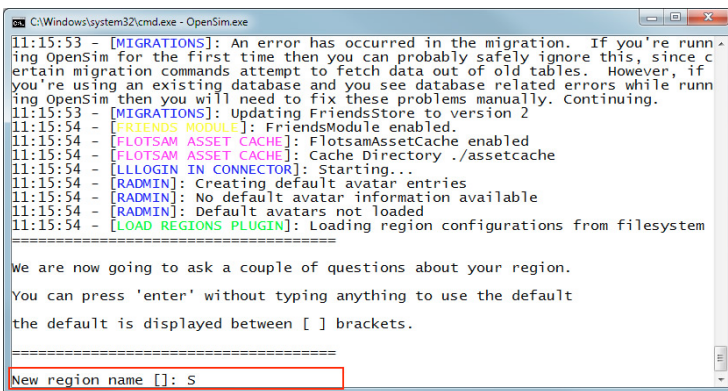


Figura 2.41 - Mensagem do *firewall* do Windows.



Quando ele terminar, dentro da janela pedirá para preencher uma lista de campos de configuração (Figura 2.42).

Figura 2.42 - Instalação do *OpenSim*.



Configurações demonstradas na Figura 2.43:

- New region name []: test environment [Enter]

- RegionUUID: apenas tecle [Enter]
- Region Location [1000,1000]: apenas tecle [Enter]
- Internal IP address [0.0.0.0]: 127.0.0.1 [Enter]
- Internal Port [9000]: apenas tecle [Enter]
- Resolve hostname to IP on start (for running inside Docker) [False]: apenas tecle [Enter]
- External host name [SYSTEMIP]: 127.0.0.1 [Enter]
- New estate name [My Estate]: room [Enter]

Figura 2.43 - Configurando o OpenSim.

```

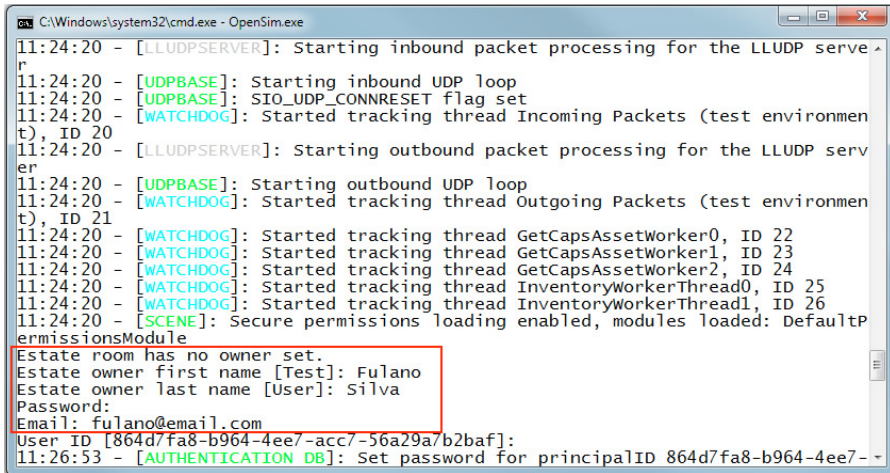
C:\Windows\system32\cmd.exe - OpenSim.exe
You can press 'enter' without typing anything to use the default
the default is displayed between [ ] brackets.
=====
New region name []: test environment
RegionUUID [ae42a53e-224b-4f6a-9d76-671a2fc4ee0f]:
Region Location [1000,1000]:
Internal IP address [0.0.0.0]: 127.0.0.1
Internal port [9000]:
Resolve hostname to IP on start (for running inside Docker) [False]:
External host name [SYSTEMIP]: 127.0.0.1
11:22:06 - [REGION LOADER FILE SYSTEM]: Loading config files from .\Regions
11:22:06 - [REGION LOADER FILE SYSTEM]: Loading config file .\Regions\Regions.in
i
11:22:06 - [REGION LOADER FILE SYSTEM]: Loaded config for region test environmen
t
11:22:06 - [LOAD REGIONS PLUGIN]: Loading specific shared modules...
11:22:06 - [LOAD REGIONS PLUGIN]: Done.
11:22:06 - [LOAD REGIONS PLUGIN]: Creating Region: test environment (ThreadID: 1
)
11:22:06 - [ESTATE]: Region test environment is not part of an estate.
11:22:06 - [ESTATE]: No existing estates found. You must create a new one.
New estate name [My Estate]: room
  
```

Configurações demonstradas na Figura 2.44:

- Estate owner first name [Test]: seu Nome [Enter]
- Estate owner last name [User]: seu Sobrenome [Enter]
- Password: informe a sua senha [Enter]

- Email: informe seu Email [Enter]

Figura 2.44 - Finalizando configuração do *OpenSim*.

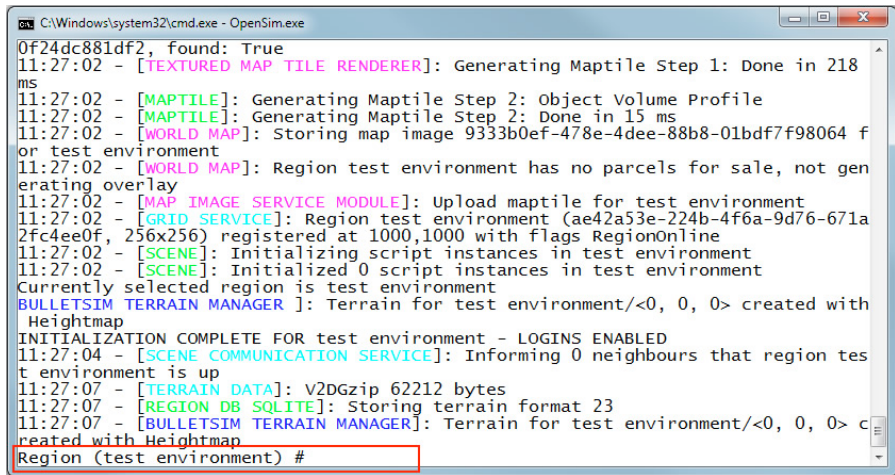


```
C:\Windows\system32\cmd.exe - OpenSim.exe
11:24:20 - [LLUDPSERVER]: Starting inbound packet processing for the LLUDP server
11:24:20 - [UDPBASE]: Starting inbound UDP loop
11:24:20 - [UDPBASE]: SIO_UDP_CONNRESET flag set
11:24:20 - [WATCHDOG]: Started tracking thread Incoming Packets (test environment), ID 20
11:24:20 - [LLUDPSERVER]: Starting outbound packet processing for the LLUDP server
11:24:20 - [UDPBASE]: Starting outbound UDP loop
11:24:20 - [WATCHDOG]: Started tracking thread Outgoing Packets (test environment), ID 21
11:24:20 - [WATCHDOG]: Started tracking thread GetCapsAssetWorker0, ID 22
11:24:20 - [WATCHDOG]: Started tracking thread GetCapsAssetWorker1, ID 23
11:24:20 - [WATCHDOG]: Started tracking thread GetCapsAssetWorker2, ID 24
11:24:20 - [WATCHDOG]: Started tracking thread InventoryWorkerThread0, ID 25
11:24:20 - [WATCHDOG]: Started tracking thread InventoryWorkerThread1, ID 26
11:24:20 - [SCENE]: Secure permissions loading enabled, modules loaded: DefaultPermissionsModule
Estate room has no owner set.
Estate owner first name [Test]: Fulano
Estate owner last name [User]: Silva
Password:
Email: fulano@email.com
User ID [864d7fa8-b964-4ee7-acc7-56a29a/b2baf]:
11:26:53 - [AUTHENTICATION DB]: Set password for principalID 864d7fa8-b964-4ee7-
```

O programa terminará de carregar todas as configurações; então, escreva a sentença (Figura 2.45):

- “terrain fill 25”

Figura 2.45 - OpenSim configurado.



```
C:\Windows\system32\cmd.exe - OpenSim.exe
0f24dc881df2, found: True
11:27:02 - [TEXTURED MAP TILE RENDERER]: Generating Maptile Step 1: Done in 218
ms
11:27:02 - [MAPTILE]: Generating Maptile Step 2: Object Volume Profile
11:27:02 - [MAPTILE]: Generating Maptile Step 2: Done in 15 ms
11:27:02 - [WORLD MAP]: Storing map image 9333b0ef-478e-4dee-88b8-01bdf7f98064 f
or test environment
11:27:02 - [WORLD MAP]: Region test environment has no parcels for sale, not gen
erating overlay
11:27:02 - [MAP IMAGE SERVICE MODULE]: Upload maptile for test environment
11:27:02 - [GRID SERVICE]: Region test environment (ae42a53e-224b-4f6a-9d76-671a
2fc4ee0f, 256x256) registered at 1000,1000 with flags RegionOnline
11:27:02 - [SCENE]: Initializing script instances in test environment
11:27:02 - [SCENE]: Initialized 0 script instances in test environment
Currently selected region is test environment
BULLETSIM TERRAIN MANAGER ]: Terrain for test environment/<0, 0, 0> created with
Heightmap
INITIALIZATION COMPLETE FOR test environment - LOGINS ENABLED
11:27:04 - [SCENE COMMUNICATION SERVICE]: Informing 0 neighbours that region tes
t environment is up
11:27:07 - [TERRAIN DATA]: v2DGzip 62212 bytes
11:27:07 - [REGION DB SQLITE]: Storing terrain format 23
11:27:07 - [BULLETSIM TERRAIN MANAGER]: Terrain for test environment/<0, 0, 0> c
reated with Heightmap
Region (test environment) #
```

Pronto, o *OpenSim* está instalado e rodando em sua máquina.

**LEMBRE-SE DE QUE OS PROGRAMAS DEVEM SEMPRE SER INICIADOS NA ORDEM:**

- 1) WAMP SERVER.
- 2) ROBUST.EXE.
- 3) OPENSIM.EXE.

## 2.2 CUSTOMIZAÇÃO DO AVATAR

Conforme Quintas-Mendes (2009), o avatar é a representação gráfica de um utilizador. O termo “avatar” provém do Hindu e do Sânscrito, significando a transfiguração do ser divino no corpo humano, a reencarnação de um deus e a sua descida do céu à terra; alguns avatares são cópias perfeitas dos seus donos (ZIMMER; VEZZANI, 2017).

A sensação de imersão mais comum obtida num ambiente tridimensional ocorre ao uso de avatares (SCHAF *et al.*, 2012). Numa aprendizagem colaborativa em mundos virtuais, a personificação por meio de avatares permite que os estudantes disponham de um processo mais fácil de comunicação baseado em rede, contribuindo para melhorar o processo de relacionamento social, influenciar positivamente o processo do grupo e criar níveis mais elevados de confiança interpessoal (BENTE *et al.*, 2004).

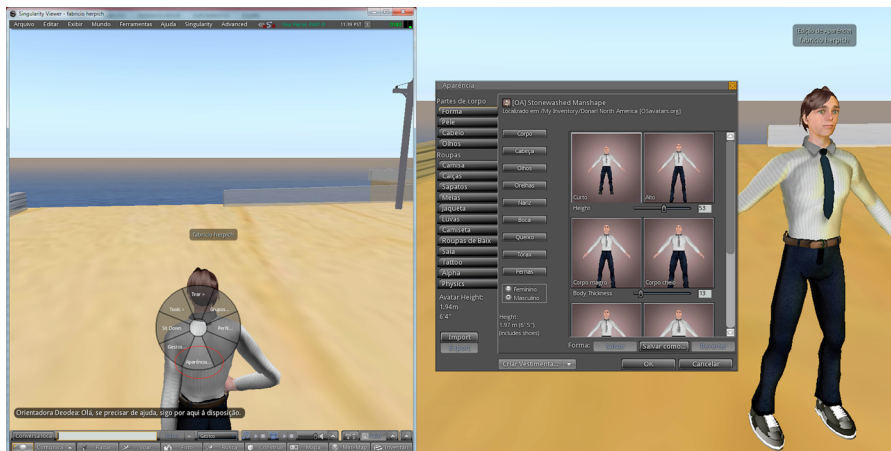
O avatar pode se comunicar com outros avatares, usando áudio, vídeo, textos e/ou gestos (CRUZ *et al.*, 2014). Para movimentar seu avatar no ambiente, podem-se utilizar as setas do teclado. Para correr, tecle duas vezes a seta para a frente. Utilize o *scroll* do *mouse* para afastar ou aproximar a câmera. No *Singularity*, também, há um botão com a opção de Voar, conforme pode ser visto na Figura 2.46.

Figura 2.46 - Ilustração sobre como voar com o avatar.



Após conseguir realizar a movimentação do personagem, torna-se importante aprender a customizar o avatar. Edita-se a aparência do avatar clicando com o botão direito do mouse sobre ele e, em seguida, em Aparência, conforme pode ser visto na Figura 2.47.

Figura 2.47 - Ilustração sobre como personalizar a aparência do avatar.



Essa é uma forma de realizar a customização do personagem de forma manual. Há também a possibilidade de efetuar a personalização do avatar de forma automática, com a utilização de uma aparência pronta. Algumas regiões previamente criadas possuem aparências predefinidas que podem ser incorporadas ao avatar; deve-se escolher a que desejar e clicar na imagem. Irá aparecer uma janela como visto na Figura 2.48, em que se deve clicar em Segure.

Figura 2.48 - Ilustração sobre como trocar de vestuário no avatar.



Observe que, na figura anterior, o nome do arquivo era MY AVATAR. Portanto, é por esse arquivo que se deve procurar no Inventário. O Inventário é um menu do *Singularity* e contém todos os objetos do Mundo



*Virtual.* Clique nele, role a tela até o final e encontre o arquivo MY AVATAR. Clique, segure e arraste o arquivo para o avatar, conforme pode ser visto na Figura 2.49.

Figura 2.49 - Ilustração sobre como selecionar o vestuário do avatar.



Dessa forma, tem-se a personalização do avatar com a inclusão automática de roupas, sendo possível, posteriormente, realizar novas modificações no perfil, de acordo com a preferência de cada usuário. A partir deste momento, podemos realizar a importação de uma diversidade de elementos para o MVI, utilizando os recursos providos pelo visualizador *Singularity*.



## 2.3 AS FERRAMENTAS DE AUTORIA

### 2.3.1 Importação de regiões prontas obtidas em repositórios

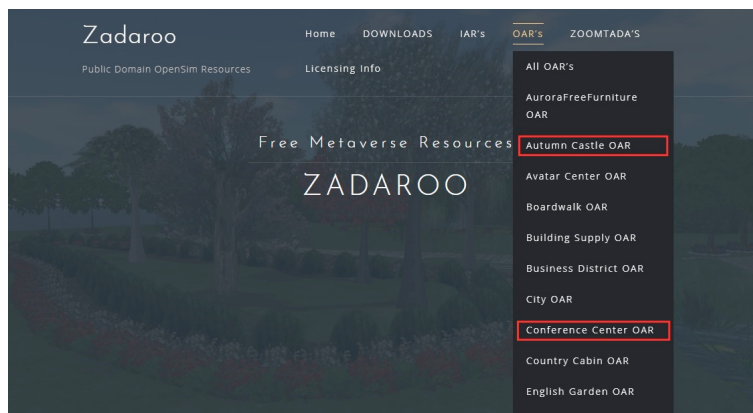
Neste momento, destaca-se que IARs são objetos encapsulados que podem ser importados, ou seja, são objetos como roupas, caixas, elementos de cenários, entre outros, que podem ser importados para dentro do inventário que o usuário possui no MVI.

Com relação aos OARs, são regiões inteiras encapsuladas que podem ser importadas, ou seja, todos os elementos e *scripts* contidos em uma região do Mundo Virtual são inseridos neste arquivo e podem ser importados para qualquer outra região, em diferentes Mundos Virtuais, de forma completa.

Para realizar a importação de regiões previamente criadas, uma das opções é realizar o *download* dos arquivos em diferentes tipos de repositórios *on-line*. Com o propósito de ilustrar um exemplo de uso de um OAR, acesse o site Zadaroo, que é o maior repositório de elementos gratuitos para *OpenSim*.

Faça o *download* das regiões *AutumnCastle* e *ConferenceCenter*, conforme visto na Figura 2.50, com o intuito de testá-las à ação de importação de uma região.

Figura 2.50 - Página inicial do repositório Zadaroo.

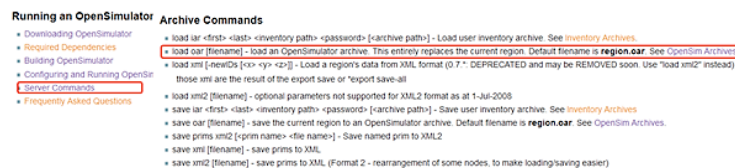


A maioria dos comandos está disponível na própria página do *OpenSim* ([OpenSimulator.org](http://OpenSimulator.org)), na opção *Server Commands* do menu. Dentro da página *Server Commands*, há uma seção *Archive Commands*, em que uma grande diversidade de comandos pode ser vista.

Recomenda-se que os comandos disponibilizados nesta página sejam inseridos no painel de comandos do *OpenSim.exe*, que é a tela com fundo preto ou branco em que foram digitados os comandos de configuração de instalação.

Para importar uma região, usa-se o comando *load nome\_do\_arquivo.oar*, conforme pode ser visto na Figura 2.51.

Figura 2.51 - Página do *OpenSim* com os comandos de configuração da região.



Fonte: <[http://OpenSimulator.org/wiki/Main\\_Page](http://OpenSimulator.org/wiki/Main_Page)>.

Observe o formato do arquivo de região em que foi feito o *download*. No caso das regiões *ConferenceCenter* e *AutumnCastle*, o formato é *tgz*. Assim, ao realizar o comando de importação, o arquivo deve ser chamado nessa extensão.

Recomenda-se que copie seu arquivo para a pasta C:, assim só precisará digitar:

- *load oar* C:\ConferenceCenter.tgz

O visualizador *Singularity* pode permanecer aberto neste processo; o resultado é automático, e a região será carregada em pequenas partes até atingir a sua totalidade. Dessa forma, tem-se uma região completamente importada com todos os objetos e *scripts* em pleno funcionamento.

Reitera-se que o arquivo anterior foi importado para a mesma região inicial em que havia sido criada. Caso se importe outro arquivo, sem que mude a região, ele sobrescreverá esta. Cria-se, por isso, uma nova região e importa-se um arquivo OAR diretamente para ela.

Na tela do OpenSim.exe, digita-se:

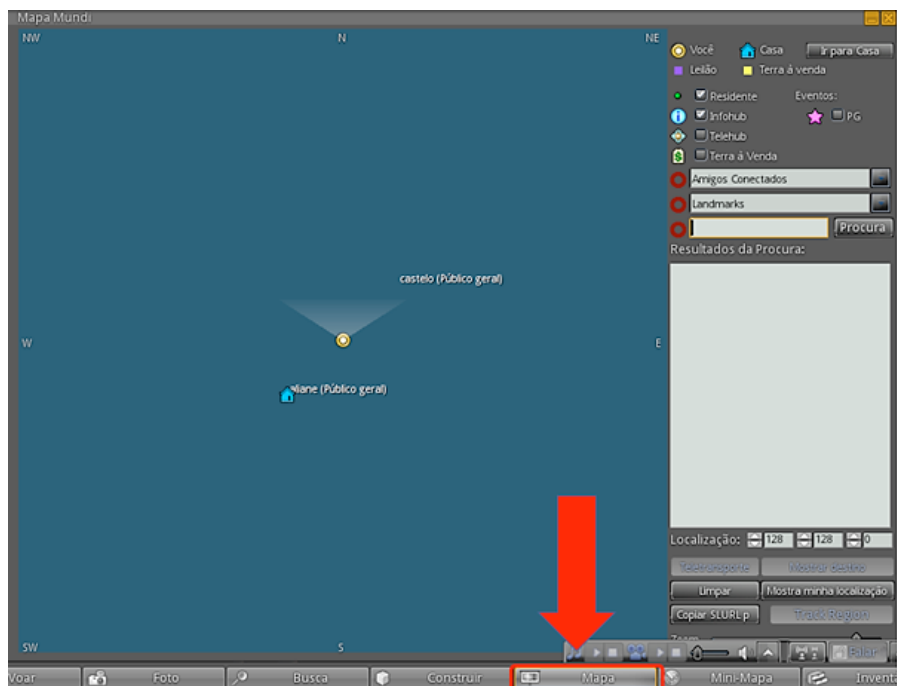
- *create region* [nome] [caminho do arquivo de regiões].

Nesse caso, para exemplificar, digita-se: *create region castelo* C:\OpenSim\OpenSim-0.9.0.1\bin\Regions\Regions.ini

Nas opções de preenchimento, é importante apenas diferenciar o local (*Location*) e o número da porta (*InternalPort*). Por padrão, a região tem o lugar 1000,1000 e a porta, 9000. Portanto, sugere-se a escolha do lugar 1001,1001 e da porta 9001.

Voltando ao *Singularity*, clique no menu Mapa para ver a região recém-criada. Para chegar a ela, basta clicar duas vezes na localização desejada (próximo ao nome “castelo”). O ponto amarelo é a localização atual, conforme pode ser visto na Figura 2.52.

Figura 2.52 - Ilustração sobre como acessar o mapa.



Dessa forma, tem-se uma região vazia criada e pronta para ser utilizada. É necessário, no entanto, popular esta região com objetos e *scripts*, de acordo com a necessidade e preferência.

## 2.4 AGREGAR COMPORTAMENTOS AOS OBJETOS (SCRIPTS)

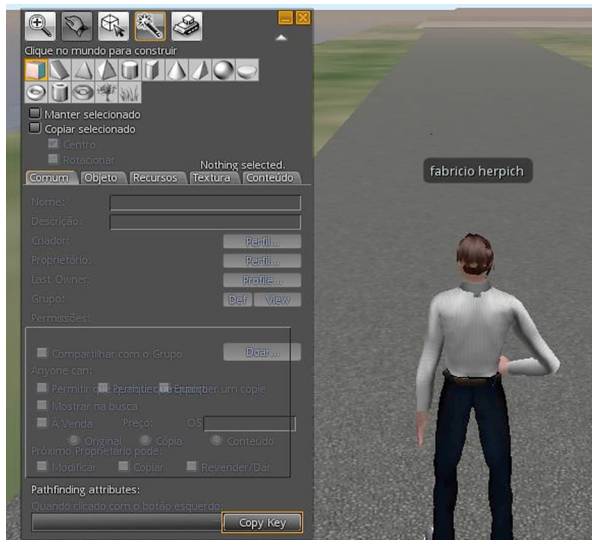
A partir do momento em que se tem uma região completamente preenchida e funcional, precisa-se ressaltar a forma de interagir com os objetos. Para importar uma imagem, com o visualizador iniciado e dentro do MV, clique no menu a opção “Arquivo”, depois clique em “Imagem”, clique em “Escolha o arquivo”, selecione o arquivo de imagem que deseja (respeitando os formatos aceitos) e clique em realizar “Upload”. O resultado pode ser visto na Figura 2.53.

Figura 2.53 - Ilustração sobre como importar uma imagem.



Para importar uma animação, com o visualizador iniciado e dentro do MV, clique no menu a opção “Arquivo”, depois clique em “Animação”, realize a escolha do arquivo (respeitando os formatos aceitos), clique em “Importe”, depois em “Carregar”. A Figura 2.54 apresenta o resultado final.

Figura 2.54 - Ilustração sobre como importar uma animação.



Para importar um arquivo no formato DAE, que são modelos 3D, com o visualizador iniciado e dentro do MV, clique no menu "Arquivo" a opção "Upload Mesh", realize a escolha do arquivo (respeitando os formatos aceitos), clique em "Importe", depois em "Calcular weights & fee" e, por fim, clique em "Carregar". A Figura 2.55 apresenta o resultado final.

Figura 2.55 - Ilustração sobre como importar um objeto 3D.



Para importar um som, com o visualizador iniciado e dentro do MV, clique no menu a opção “Arquivo”, depois clique em “Som”, realize a escolha do arquivo (respeitando os formatos aceitos), clique em “Importe”, depois em “Upload”. A Figura 2.56 apresenta o resultado final.

Figura 2.56 - Ilustração sobre como importar um som.



Para importar um arquivo no formato XML, que renderiza um objeto, com o visualizador iniciado e dentro do MV, clique no menu a opção “Arquivo”, depois clique em “Object XML”, realize a escolha do arquivo (respeitando os formatos aceitos), clique em “Importe”, depois em “Carregar”. A Figura 2.57 apresenta o resultado final.

Figura 2.57 - Ilustração sobre como importar um arquivo XML.



Para criar um objeto 3D, há no menu inferior a opção chamada “Construir”, em que, ao clicá-la, é apresentada uma tela para a escolha do formato de objeto 3D desejado. A Figura 2.58 apresenta o resultado final.

Figura 2.58 - Ilustração sobre como criar um objeto 3D com os recursos do ambiente.





Ao escolher o objeto desejado, que, neste caso, é um cubo, pode-se dimensioná-lo de acordo com os eixos X, Y e Z. Além disso, pode-se movimentá-lo para qualquer local válido dentro do ambiente, de acordo com a necessidade ou preferência. A Figura 2.59 apresenta o resultado final.

Figura 2.59 - Ilustração sobre como manipular o objeto 3D criado.



Os Mundos Virtuais são baseados em um sistema de coordenadas cartesianas, contendo três eixos: Eixo X, Eixo Y e Eixo Z. Cada região no Mundo Virtual tem um conjunto único de coordenadas cartesianas, conforme pode ser visto na ilustração anterior, em que o objeto representado por uma caixa está localizado em uma posição das coordenadas. Há uma grande gama de opções a serem exploradas para cada objeto 3D no ambiente, como a personalização do seu tamanho, posição, rotação, aplicação de texturas em suas faces, dentre outras possibilidades.

Também é possível atribuir e criar *scripts* com blocos de programação previamente definidos nos objetos 3D. Precisa-se selecionar a opção de “Conteúdo” entre as abas disponíveis, posteriormente clicando em “Novo Script”. Dessa forma, um bloco de código-padrão em um arquivo de *script* é criado e associado a este objeto. As Figuras 2.60 e 2.61 apresentam, respectivamente, o resultado final do *script* criado e de seu bloco de programação padrão criado.

Figura 2.60 - Ilustração sobre como associar um *script* a um objeto.

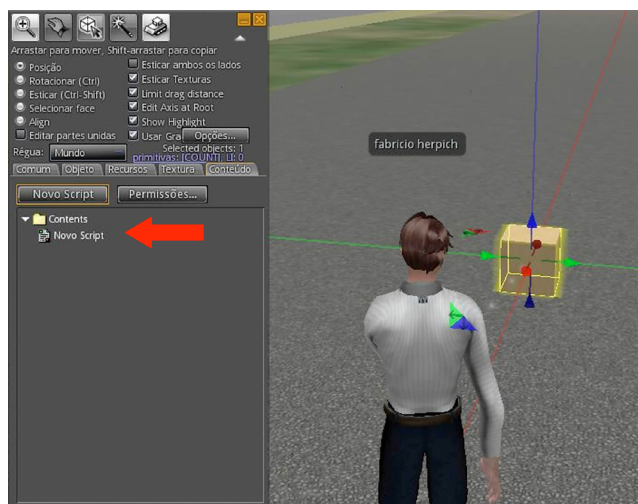
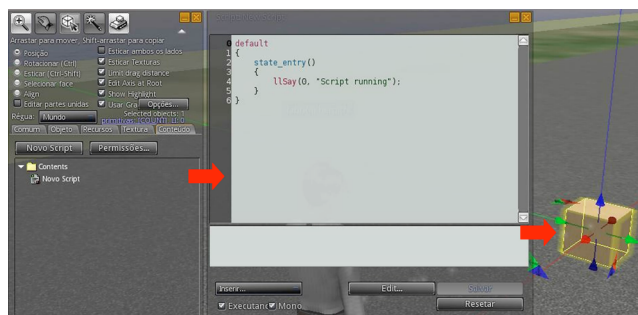
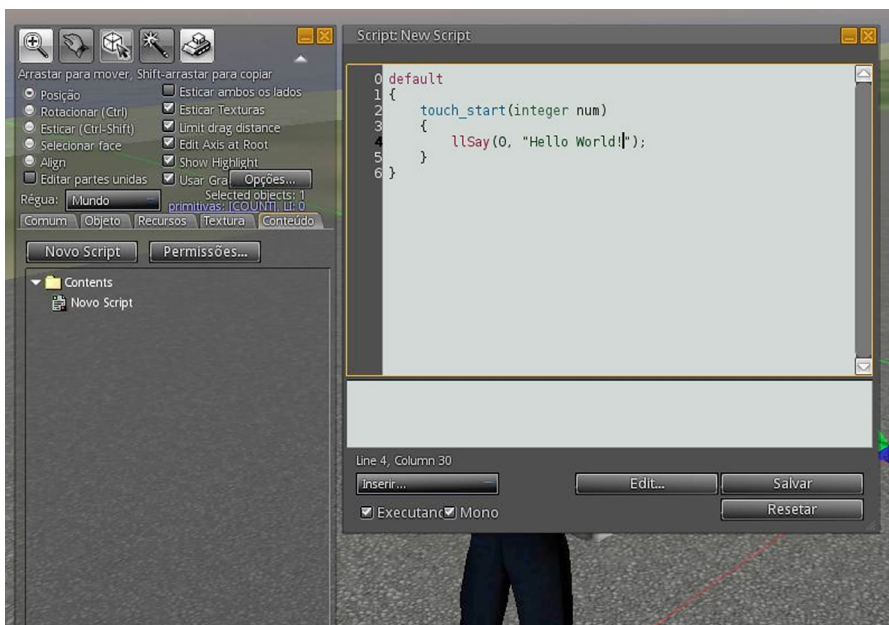


Figura 2.61 - Ilustração sobre como escrever e salvar o *script*.



É importante ressaltar que a programação tem como base a *OpenSim Script Language* (OSSL), sendo adaptada para a utilização no ambiente. No caso da ilustração, o objeto, ao ser inicializado, emite uma mensagem global no *chat* (comando `llSay`), por meio do canal 0, que transmite a todos os usuários com o conteúdo de “*Script Running*”. A Figura 2.62 apresenta um exemplo de uso do *script* associado ao objeto.

Figura 2.62 - Ilustração com um exemplo de *script*.



Neste exemplo, a função previamente definida no ambiente denominada “*touch\_start*” habilita o clique em um objeto e permite que determinada ação seja realizada após o usuário ter interagido com o clique no objeto. Neste caso, o usuário, ao clicar no cubo, verá uma mensagem geral no *chat* com o conteúdo de “*Hello World*”. A partir desse momento, pode-se ter uma infinita variedade de possibilidades a serem construí-

das com os *scripts*, que permitem movimentar objetos e personagens, assim como fazer com que realizem ações previamente definidas, exibam diferentes comportamentos, falem determinadas palavras, entre outras possibilidades.

## **2.5 NPC - NON-PLAYER CHARACTER**

Tendo em vista os potenciais benefícios derivados do uso de sistemas de tutoria inteligentes, foi investigada, no Projeto AVATAR, a viabilidade tecnológica para a implantação de agentes com algum grau de inteligência a fim de apoiar a atividade dos participantes de experimentos e atividades de aprendizagem em mundos virtuais. A investigação utilizou um ambiente implementado mediante o uso do *software* livre *OpenSim*, e os agentes foram construídos usando a estratégia de *Non-Player Character* (NPC), tal como descrito em *OpenSim* (2013). Para ampliar a flexibilidade das interações entre o estudante, por meio de seu avatar, e o NPC, foi estabelecida sua interconexão com um *chatterbot* cuja base de conhecimento foi preparada para oferecer respostas relevantes ao contexto do experimento realizado.

O controle de um NPC, no ambiente *OpenSim*, é feito por meio de *scripts* acoplados a objetos de controle, executando comandos *Linden Scripting Language* (LSL) e *OpenSim Scripting Language* (OSSL) (OPEN-SIM, 2013). A ativação de um NPC pode ocorrer por meio de um toque do avatar em algum objeto especialmente criado no Mundo Virtual.

Também pode ser ativado por meio de um comando registrado no *chat*, assim como por meio de sensores que detectam a presença do avatar e realizam automaticamente a ativação do NPC para interagir com ele.

Por exemplo, é possível definir que um NPC irá realizar um conjunto de etapas previamente definidas, por meio da linguagem de *script* OSSL, sendo que, no momento de sua ativação, este conjunto de ações será desempenhado pelo *bot*. Esse conjunto de ações pode englobar a sua interação com o avatar por meio do *chat*, explicando algum tipo de experimento existente no Mundo Virtual Imersivo ou respondendo a questionamentos realizados por algum avatar, ou ao tocar algum tipo de objeto, dentre outras atividades.

Além da função de interação com o usuário (via seu próprio avatar), o NPC também pode efetuar um registro da interatividade ocorrida entre ele e os avatares no Mundo Virtual Imersivo, seja por meio de *logs* do *chat*, seja por algum outro tipo de recurso, com o objetivo de proporcionar elementos adicionais para a avaliação das ações do estudante durante o período de atividade no Mundo Virtual Imersivo.

Nesta seção, será visualizado o funcionamento, sua configuração e programação de ações, gestual, movimentação e mensagens produzidas (por geração interna do texto ou como reação a perguntas enviadas via *chat* para o NPC, que pode também ser integrado a um *chatbot* com vista a ampliar as possibilidades de produzir respostas às questões dos usuários).

Na sequência, serão apresentadas duas possibilidades de conexões com base de dados AIML, utilizando *Pandorabots* e o *Program-O*.

## 2.5.1 Conexão do NPC no *OpenSim* com o ambiente *Pandorabots* ao *OpenSim*

A primeira providência para estabelecer a interligação entre o NPC e o *Pandorabots* é naturalmente a criação de uma conta no *Pandorabots*. O nome e dados de acesso ao bot criado devem ser obtidos, pois serão necessários para a configuração do NPC. O *Pandorabots* criado deve se tornar público.

A segunda parte da tarefa envolve a criação de um NPC, ou seja, um personagem programável e não manipulável por um avatar, em Mundos Virtuais Imersivos. Dessa forma, o usuário terá a sensação de que está interagindo com outro avatar, o que lhe proporcionará uma sensação de tutoria constante. No ambiente *OpenSim*, é necessário fazer algumas edições no arquivo *OpenSim.ini*, que está na pasta */BIN*. Comandos de configuração deverão ser ajustados, tal como ilustrado a seguir.

Algumas atribuições devem ser alteradas:

- Enabled = true na sessão [NPC].
- Enabled = true na sessão [XEngine].
- AllowOSFunctions = true na sessão [XEngine].
- OSFunctionThreatLevel = VeryHigh na sessão [XEngine].

Será alterado o [XEngine] na seção [GridInfo] e serão adicionadas as linhas específicas de funções NPC:

- Allow\_osNpcCreate = true.
- Allow\_osNpcMoveTo = true.

- Allow\_osNpcRemove = true.
- Allow\_osNpcSay = true.
- Allow\_osAvatarPlayAnimation = true.
- Allow\_osAvatarStopAnimation = true.

Assim feito, o arquivo OpenSim.ini deverá ser salvo. Cabe destacar que tais alterações afetarão todas as regiões. A seguir, devem ser editados os *scripts* em um *prim* (objeto do Mundo Virtual) disparador do NPC. Quando este objeto for tocado pelo avatar, será instanciado um NPC. A programação do NPC para a comunicação com o *chatterbot* do *Pandorabots* envolve o uso de comandos tais como no exemplo:

```
{ state_entry() { gOwner = IIGetOwner();
  cust="";botid="a77ebb73ce343648"; IIListen(0,"",NULL_KEY,""); }
```

O código que estabelece o *link* com o banco de dados do *Pandorabots* usa comandos como nos exemplos:

```
link_message(integer sender_num, integer num, string msg, key id) {re-
  questid = IHTTPRequest("http://www.pandorabots.com/pandora/
  talk-xml?botid="+botid+"&input="+IIEscapeURL(msg)+"&custid="+cust,[HTTP_ME-
  THOD,"POST"], ""); } http_response(key request_id, integer status, list metadata,
  string body) { if (request_id == requestid) if (msg == "/create"){ listen_to(id);
```

A criação do NPC com número UUID do avatar é feita usando o código:

```
npc = osNpcCreate("Test", "NPC", <128, 128, 24>, "b95bf240-e386-4f1d-bc57-e-5b313ff3d58"); osNpcSay (npc, "Eu estou criado"); return;
```

Para criar um NPC em seu Mundo Virtual, basta adicionar a seguinte programação a um *prim*:

```
integer criado = FALSE;
key npc;
Default {
state_entry()
{ llSensorRepeat("", "", AGENT_BY_LEGACY_NAME, 30.0, PI, 2.0); } //final do
entry
sensor(integer num_detected) { //inicio do sensor
if (criado==FALSE) {
npc = osNpcCreate("Jump", "João", llGetPos() - <0.5, 0, 0>, "joao");
osNpcPlayAnimation(npc, "pula2"); criado=TRUE; } //fim do if
} //final do sensor
no_sensor()
{ osNpcSay(npc, "Volte sempre!");
osNpcRemove(npc);
npc = NULL_KEY;
criado=FALSE; } //final do não sensor
} //final
```



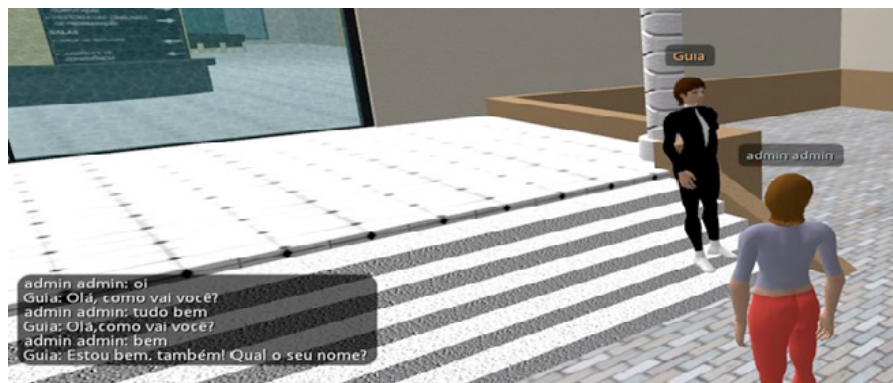
Pode-se adicionar a movimentação ao NPC, com códigos tais como os abaixo indicados:

```
osAvatarPlayAnimation (npc, "avatar_backflip");
{
  llSleep(2);
  osAvatarStopAnimation (npc, "avatar_backflip");
  llSleep(1);
  osAvatarPlayAnimation (npc, "bouncy_ball_walk");
  llSleep(9); osAvatarStopAnimation (npc, "bouncy_ball_walk");
  osNpcSay (npc, "Seja bem-vindo ao Mundo Virtual...");
}
```

Com essas codificações, instancia-se um NPC conectado à base de dados do *Pandorabots*. A base de dados do *chatterbot* pode ser personalizada para o experimento em que o *chatterbot* vai intervir. A edição da base de conhecimento do *chatterbot* é feita editando um arquivo construído com a linguagem de marcação AIML. As principais *tags* do AIML são:

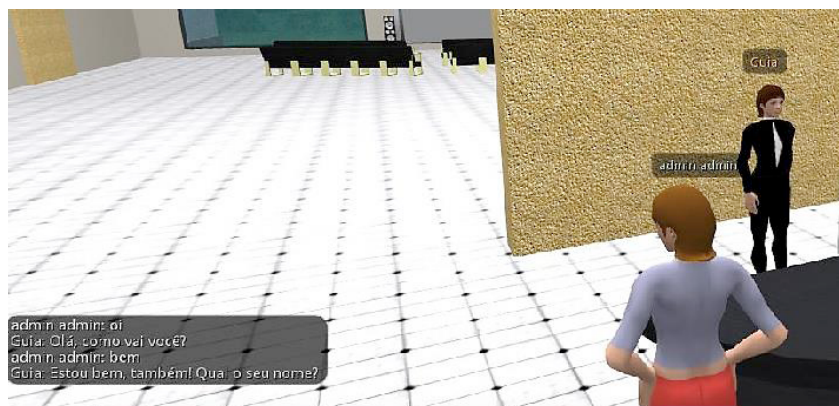
- <aiml> inicia e termina um bloco programado em AIML;
- <category> identifica uma “unidade de conhecimento” na base de conhecimento;
- <pattern> identifica um padrão de mensagem simples frequentemente utilizado por usuários;
- <template> contém a resposta para uma mensagem do usuário.

Figura 2.63 - Chatterbot do Pandora integrado com o OpenSim.



No exemplo visto na Figura 2.63, o padrão de entrada “Oi”, seguido de qualquer informação, terá como resposta ou possível resposta: “Olá, como vai você?”; “Tudo bem”; “Estou bem, também, qual é seu nome?”. Com a utilização de AIML, pode-se definir mais de uma resposta para um padrão e ainda se podem especificar critérios de escolha de cada uma das respostas que a base de conhecimento pode conter.

Figura 2.64 - Guia tutor interagindo com o aluno.



Na Figura 2.64, o exemplo mostra o guia tutor interagindo, usando as interações sociais (conhecimento genérico usável em qualquer diálogo usuário-*chatterbot*), antes de responder sobre a sua base de conhecimento específico do contexto em que esteja sendo usada.

Figura 2.65 - Página do site do *Chatterbot Pandora*.



**Human:** java

**Projeto:** Java é uma linguagem de programação orientada a objeto desenvolvida na década de 90 por uma equipe de programadores chefiada por James Gosling, na empresa Sun Microsystems. Diferentemente das linguagens convencionais, que são compiladas para código nativo, a linguagem Java é compilada para um bytecode que é executado por uma máquina virtual. A linguagem de programação Java é a linguagem convencional da Plataforma Java, mas não sua única linguagem.

Na Figura 2.65, demonstra-se como o *chatterbot* apresenta a sua base de conhecimento específica, relatando sobre os objetos e artefatos encontrados no Mundo Virtual. O avatar aluno é representado como *human*, e o *chatterbot* se apresenta como projeto respondendo à pergunta buscada em sua base de conhecimento, criada para atender a esta demanda; ou seja, ao digitar “Java”, o aluno buscou sobre a linguagem de programação Java.

A base de conhecimento pode ser criada em um serviço de hospedagem de um *software* simulador de robô (também conhecido como *bot*). De qualquer navegador (*browser*), criam-se e publicam-se seus próprios robôs via *web*. Exemplo: *Pandorabots*.

A programação à base de dados é via *Artificial Intelligence Markup Language*, uma linguagem baseada em XML desenvolvida para criar diálogos semelhantes à linguagem natural, por meio de *softwares*, simulando, assim, a inteligência humana.

## **2.5.2 Conexão do NPC no *OpenSim* com o agente conversacional ATENA**

No âmbito do Projeto AVATAR, não foi utilizado o servidor *Pandorobot*, em função de práticas comerciais daquele serviço. Uma versão do *software* ALICEbot foi instalada em um servidor *web* local. O programa utilizado foi o *Program-O AI Chatterbot*.

O *Program-O*, a partir de sua versão 2.6.5, permite a interpretação de *JavaScript*, o que é viabilizado inserindo dentro de categorias as *tags* `<script>`, abrindo diversas possibilidades. Por exemplo, em uma resposta pode ser aberta uma janela de notificação, inserindo o código *alert* (“exemplo”), cujo texto “exemplo” será exibido na janela.

Para estabelecer a comunicação entre o NPC e esse agente, é necessária a programação das funcionalidades que serão executadas pelo *script* utilizado para a criação de um NPC no MVI, funcionalidades estas que podem ser do tipo seguir o aluno dentro do ambiente e responder aos questionamentos sobre Física.

O primeiro procedimento foi criar variáveis para serem utilizadas no processo de conexão NPC ao Atena:

```

string programaURL = "http://avatar.cinted.ufrgs.br/atena/chatbot/conversation_
start.php?say=";

string avatarName = "";
string conversationID = "";
key conversationRequest = NULL_KEY;
key idNPC;
key idAvatar;
integer escutaCanal = 7;
integer criado = FALSE;

```

Essa função é muito importante, porque algumas respostas são extensas e, sem esta programação, seriam apresentadas truncadas.

```

string QuebraDeLinha(string texto_retorno, string separar, integer espaco){
    integer compensar_espaco = espaco;
    integer retornoTam = IlStringLength(texto_retorno);
    integer separatorLen = IlStringLength(separar);

    while (retornoTam > espaco)
    {
        integer dividir = IlSubStringIndex(IlGetSubString(texto_retorno, compensar_espaco,
-1), separar);
        if (dividir != -1)
        {
            texto_retorno = IlInsertString(texto_retorno, compensar_espaco + dividir + separa-
torLen, "\n");
            retornoTam = IlStringLength(texto_retorno) - compensar_espaco;
            compensar_espaco += dividir + espaco + 2;
        }
        else retornoTam = -1;
    }
    return texto_retorno;
}

```

Na sequência, tem-se o código que estabelece a conexão com o banco de dados do *Program-O*, que utiliza comandos como nos exemplos:

```
default {
state_entry(){
    llSetText("Escutando o canal "+escutaCanal, <200,100,50>, 1);
    llListen(escutaCanal,"",NULL_KEY,"");

    llSensorRepeat("", "",AGENT, 15.0, PI, 2.0); // 2.0 segundos
}

listen(integer channel, string name, key id, string message){
    if(id == idAvatar){ //Só conversa com quem chegou primeiro.
        if (criado == TRUE){ //Só responder quando existir um NPC criado.
            llSetText("Aguardando resposta...", <1.0, 1.0, 0.0>, 1.0);
            if (conversationID == "")
            {
                conversationRequest = llHTTPRequest(programoURL
                +llEscapeURL(message) +"&bot_id=2&format=json",
                [HTTP_METHOD,"GET", HTTP_MIMETYPE,"plain/text;charset=utf-8"], "");
            } else {
                conversationRequest = llHTTPRequest(programoURL+
                llEscapeURL(message)+"&convo_id="+conversationID+
                "&bot_id=1&format=json", [HTTP_METHOD,"GET",
                HTTP_MIMETYPE,"plain/text;charset=utf-8"], "");
            } if (message != "") {
                if (message=="matar" || message=="quit" || message=="bye" || message=="tchau"){
                    llInstantMessage(idAvatar, "Adeus!");
                    osNpcRemove(idNPC);
                    idNPC = NULL_KEY;
                    criado = FALSE;
                }
            }
        }
    }
}
}
```

Detalhando o código acima, na seção `state_entry()`, a função `llListen` abre um canal de comunicação para a conversação entre o estudante e a Tutora Atena e executa o evento `listen()`, enquanto a função `llSensorRepeat` verifica repetidamente a presença do avatar do aluno, em um raio de alcance de 15 metros, executando o evento `sensor()`.

O evento *listen* (*integer channel, string name, key id, string message*) estabelece ou não a comunicação entre o aluno e a Tutora Atena. Primeiro ele verifica a presença do avatar e se a Tutora Atena (NPC) foi criada. Se ambos forem verdadeiros, uma conexão com o *Program-O* será estabelecida.

Se for a primeira conversa do avatar com a Tutora Atena (`conversationID == ""`), uma identificação para a conversa será estabelecida, e as conversas seguintes seguirão esta identificação por meio da variável `"convo_id"`.

Por fim, se o canal de comunicação 7 receber as *strings* `"matar"`, `"quit"`, `"bye"` ou `"tchau"`, a Tutora Atena será removida do MVI e o canal de comunicação será invalidado.

A próxima parte de código tem o objetivo de responder à função `llHTTPRequest`, que faz uma chamada HTTP à página PHP que processa a conversação. Essa página está hospedada no domínio em que o *chat-terbot* está disponível:

```

http_response(key request_id, integer status, list metadata, string body){
  if (request_id == conversationRequest){
    if (status != 200) {
      llSetText("Erro, tente dizer algo novamente!", <1.0, 0.0, 0.0>, 1.0);
      llOwnerSay("HTTP Error " + (string)status + ": " + body);
    } else {
      llSetText("Conversando com " + avatarName + "\nconvo_id: " +
conversationID, <0.0, 0.0, 0.0>, 1.0);
      if (conversationID == "") {
        conversationID = llGetSubString(body, 13, 34);
      }
      string botReply = llGetSubString(body, llSubStringIndex(body, "\botsay\":" +
+ 10, -3);

      osNpcPlayAnimation(idNPC, "Type");
      llSleep(3);
      osNpcStopAnimation(idNPC, "Type");

      llInstantMessage(idAvatar, QuebraDeLinha(botReply, " ", 70));
    }
  }
}

```

Observando o código acima, percebe-se, inicialmente, que verificar a resposta do *Program-O* é para a requisição HTTP enviada. Uma vez que o código de *status* do HTTP com valor 200 indica sucesso, todos os valores diferentes são tratados como erro. Se o código HTTP é 200, indica sucesso e o conteúdo de “*body*” (a resposta do *Program-O*) tem uma parte recortada (“\botsay\":"), logo a Tutora Atena é animada, simulando estar digitando, e a resposta do *chatterbot* é apresentada ao avatar que a solicitou por meio da função “*llInstantMessage*”.



A parte do *script* que verifica a presença e a ausência do avatar é apresentada a seguir:

```
sensor(integer num){
  if (criado == FALSE){
    idAvatar = IIDetectedKey(0);
    avatarName = IIDetectedName(0);
    vector posAvatar = IIDetectedPos(0);
    vector posNPC = IIGetPos() + <1,1,0>;
    idNPC = osNpcCreate("Tutora", "Atena", posNPC, " ");

    criado = TRUE;

    vector userPos = IIDetectedPos(0);
    vector npcPos = osNpcGetPos(idNPC);
    vector diff = userPos - npcPos;
    if(IIVecMag(diff) > 3.0) {
      vector diffNorm = IIVecNorm(diff);
      vector targetPos = <userPos.x-diffNorm.x*2.0, userPos.y-diffNorm.y*2.0,
userPos.z>;
      osNpcMoveToTarget(idNPC, targetPos, OS_NPC_NO_FLY);
    }

    IInstantMessage(idAvatar, "Olá "+avatarName+", seja bem-vindo ao Projeto AVA-
TAR.");
    IISleep(1);
    IInstantMessage(idAvatar, "Eu sou a Atena e estou aqui para o auxiliar com os experi-
mentos de física.");
    IISleep(2);
    IInstantMessage(idAvatar, "Converse comigo no canal " + escutaCanal + " (ao escre-
ver uma mensagem utilize o prefixo '/' + escutaCanal + "mensagem' sem as aspas).");
  } else {
    vector userPos = IIDetectedPos(0);
    vector npcPos = osNpcGetPos(idNPC);
    vector diff = userPos - npcPos;
    diff.z = 0;
    vector initial = <1.0,0.0,0.0>;
    osNpcSetRot(idNPC, IIRotBetween(initial,diff));
  }
}
```

```

userPos = lIDetectedPos(0);
npcPos = osNpcGetPos(idNPC);
diff = userPos - npcPos;

if(lIVecMag(diff) > 3.0){
    vector diffNorm = lIVecNorm(diff);
vector targetPos = <userPos.x-diffNorm.x*2.0, userPos.y-diffNorm.y*2.0, userPos.
z>;
    osNpcMoveToTarget(idNPC, targetPos, OS_NPC_NO_FLY);
}
}
}

no_sensor(){
    osNpcRemove(idNPC);
    idNPC = NULL_KEY;
    criado = FALSE;
}
}

```

Lembre-se de que o *script* acima é executado a cada 2 segundos e verifica-se a presença ou ausência de um avatar em um raio de 15 metros no Mundo Virtual. Pormenorizando, se a Tutora Atena (NPC) não existir (`criado==FALSE`), ela será criada, a posição do avatar será registrada e a Tutora Atena será comandada para caminhar até o avatar, ficando à sua frente e se colocando à disposição do avatar.

Caso a Tutora Atena já tenha sido criada, quando o avatar for detectado, ela caminhará até ele e ficará à sua frente. Essa parte do código causa o efeito de a Tutora Atena “caminhar” com o avatar do aluno e sempre estar de frente para ele. Essa estratégia foi pensada para que o aluno tenha a impressão de que a Tutora Atena está sempre acompa-

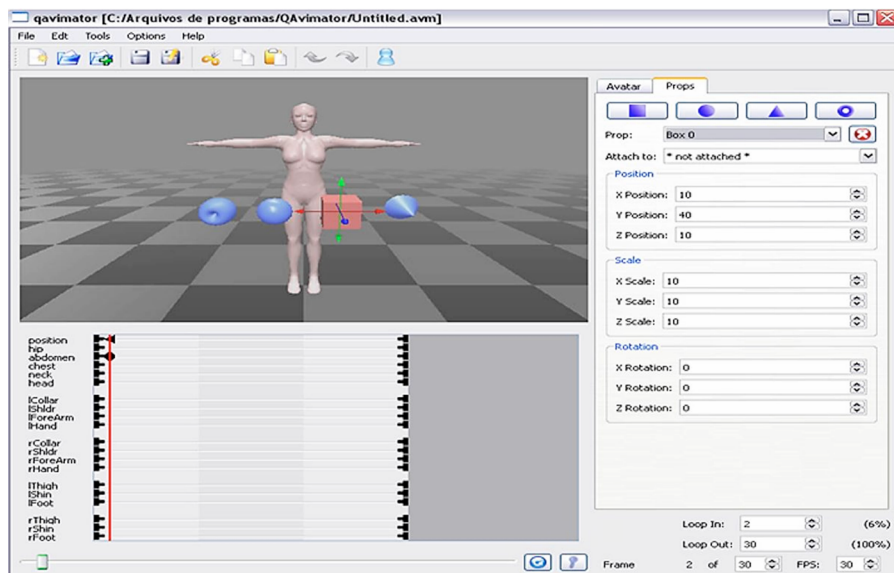
nhando o avatar e sempre prestando atenção ao que ele “fala”. Isso fornece um comportamento de atenção e cuidado à Tutora Atena, levando o aluno a sentir confiança na conversação.

Caso nenhum avatar seja detectado, o evento no sensor() será executado, a Tutora Atena será removida e as variáveis recebem seus valores iniciais. Dessa forma, a conversação fica em estado de espera novamente.

### **2.5.3 Criação de ações e gestos**

O NPC pode realizar qualquer ação. Desde que programado, ele poderá dançar, exercitar-se, andar etc. A programação pode ser editada por vários aplicativos do tipo Daz Studio, QAnimator, Poser, ou qualquer outro editor de movimento capaz de gerar um arquivo do tipo BVH, que será utilizado no Mundo Virtual para gerar animações no NPC, em um *prim* ou mesmo no próprio avatar do participante.

Figura 2.66 - Tela do Qanimator.



Esse tipo de aplicativo gera um BVH que será utilizado no Mundo Virtual. O formato de arquivo BVH foi originalmente desenvolvido pela Biovision, uma empresa de serviços de captura de movimento, como uma forma de fornecer dados de captura de movimento a seus clientes. O nome BVH significa “dados hierárquicos de Biovision”.

Existem diversos repositórios desse tipo de arquivo que já estão prontos para serem utilizados e muitos podem ser editados. A instalação é muito simples. Primeiro, deve-se importar para o *OpenSim* o seu arquivo BVH, bastando clicar em “import files” do menu principal. Nessa tela, serão encontradas várias configurações que poderão ajudar em sua utilização – por exemplo, se é clicável, se irá aguardar o evento tempo.

Para adicionar a animação, basta soltar o *script* e um cartão em branco no objeto, clicar no objeto e dar um nome à sua animação. Mova todos os *prims* e clique em Gravar. Quando terminar, clique no Nome

da Animação e assista a cada movimento! Esse *script* usa um comando novo e muito rápido chamado *IISetPrimitiveParamsFast*, que significa que dezenas de *prims* podem se mover quase instantaneamente.

Os *prims* possuem algumas animações simples previamente programadas:

```
void osAvatarPlayAnimation (key avatar, string animation)
```

Essa função faz com que uma animação seja reproduzida no avatar especificado.

A **animação** variável deve ser o nome de uma animação dentro do inventário de tarefas. Por motivos de segurança, os UUIDs não são permitidos aqui.

Em vez do nome de uma animação no inventário do *prim*, também podem-se usar os nomes das animações internas do visualizador.

- **osAvatarPlayAnimation** não executam nenhuma verificação de segurança ou solicitam permissões de animação do avatar de destino.
- **Nível de ameaça:** Muito alto. **Permissões:** o uso desta função está sempre desativado por padrão **Demora** 0 segundo. **Exemplos).**

```
// Exemplo de uso:  
padrão { touch_start ( integer num )  
{  
    string anim = IIGetInventoryName ( INVENTORY_ANIMATION, 0 );  
    osAvatarPlayAnimation ( IIDetectedKey ( 0 ), anim );  
}  
}
```

**Notas:** Ao usar essa função em um objeto que requer que o usuário se sente no objeto (como uma cadeira ou um *poseball*), precisa-se interromper a animação *sit*, incluindo o *snippet*:

```
alterado ( mudança de inteiro )
{
  if ( alterar e CHANGED_LINK )
  {
    chave user = IAvatarOnSitTarget ( );
    osAvatarStopAnimation ( usuário, "sentar" );
    osAvatarPlayAnimation ( usuário, anim );
  }
}
```

## REFERÊNCIAS

BENTE, Gary; KRÄMER, Nicole C. Virtual gestures: analyzing social presence effects of computer-mediated and computer-generated nonverbal behaviour. *In: PRESENCE 2002 – ANNUAL INTERNATIONAL WORKSHOP ON PRESENCE*, 5., 2002, Porto, Portugal. **Proceedings** [...]. Porto: ISPR, 2002.

CRUZ, Armando; PAREDES, Hugo; FONSECA, Benjamim; MORGADO, Leonel; MARTINS, Paulo. Can presence improve collaboration in 3D virtual worlds? *In: SLACTIONS 2013: RESEARCH CONFERENCE ON VIRTUAL WORLDS – LEARNING WITH SIMULATIONS*, 2013, Vila Real, Portugal. **Proceedings** [...]. [S.l.]: Elsevier Procedia Technology, 2014. v. 13, p. 47-55.

KOTSILIERIS, Theodore; DIMOPOULOU, Nikoletta. The evolution of e-learning in the context of 3D virtual worlds. **Electronic journal of e-learning**, v. 11, n. 2, p. 147-167, 2013.

OPENSIM. **Non-player characters**. USA, 2013. Disponível em: <<http://OpenSimulator.org/wiki/NPC>>. Acesso em: 15 dez. 2018.

QUINTAS-MENDES, António. **Comunicação mediada por computador e educação online: da distância à proximidade**. Disciplina de Comunicação e Educação, Mestrado em Comunicação Educacional, UAb de Portugal, 2009.

SCHAF, Frederico Menine; PALADINI, Suenoni; PEREIRA, Carlos Eduardo. 3D Autoslab prototype: a social, immersive and mixed reality approach for collaborative learning environments. **iJEP – International journal of engineering pedagogy**, v. 2, n. 2, p. 15-22, 2012.

SGOBBI, Fabiana Santiago *et al.* Interação com artefatos e personagens artificiais em mundos virtuais. *In*: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – CBIE, 3.; SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE, 25., 2014, Dourados. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2014. p. 642-651.

TAROUÇO, Liane Margarida Rockenbach *et al.* Virtual laboratory for promoting engagement and complex learning. *In*: E-LEARN 2014 – WORLD CONFERENCE ON E-LEARNING, 2014, New Orleans, USA. **Proceedings [...]**. Waynesville, NC: The Association for the Advancement of Computing in Education – AACE, 2014. p. 1933-1938.

ZIMMER, Josete Maria; VEZZANI, Marco Antônio. Second Life para educação à distância: uma experiência entre estudantes brasileiros e portugueses. *In*: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – WCBIE, 6., 2017, Recife. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2017.





# 3

## **Agentes conversacionais no Mundo Virtual Imersivo**

**Anita Raquel da Silva - UFRGS**

**anita@cinted.ufrgs.br**

**Clóvis da Silveira - PPGIE - UFRGS**

**csclovis@gmail.com**

**Liane Margarida Rockenbach Tarouco**

**PPGIE - UFRGS - liane@penta.ufrgs.br**

O avanço das tecnologias de informação e comunicação tem apresentado consideráveis contribuições aos processos de ensino e aprendizagem, tais como maiores possibilidades de compreensão dos conteúdos e esclarecimentos de dúvidas mediados por agentes computacionais.

Agentes Conversacionais (também chamados de *chatbots*, *chatterbots*, agentes de conversação, sistemas de diálogo) são aplicações que simulam conversa humana por meio de uma interação textual entre um usuário humano que fornece a entrada e o agente que responde a ele (fornecendo respostas ou formulando perguntas). Conforme Wallace (1995), que implementou a mais popular solução para apoiar a implementação do agente conversacional Alicebot, os Agentes Conversacionais são definidos como sistemas que buscam simular uma conversa por meio do intercâmbio de mensagens em linguagem natural.

O *software* do agente conversacional interpreta a consulta enviada, busca em uma base de conhecimento uma resposta que contenha elementos da consulta e seleciona, de acordo com alguns critérios, uma resposta para enviar ao usuário. A base de conhecimento que fornece o apoio ao funcionamento do agente conversacional, no caso do Alicebot, é construída usando uma linguagem de marcação denominada AIML (*Artificial Intelligence Markup Language*). Essa linguagem permite configurar e programar as respostas do agente conversacional usando recursos de mais alto padrão, sem necessidade de conhecer uma linguagem de programação. O livro técnico *The elements of AIML style*, escrito por Wal-

lace (2003), apresenta os primeiros passos para criar seu próprio agente conversacional utilizando a linguagem AIML, com vista às possibilidades de perguntas e respostas, apresentando as *tags* da linguagem.

Estudos de Satu *et al.* (2015) apontam alguns Agentes Conversacionais na linguagem AIML com finalidades diferentes, dentre as quais estão as aplicações relacionadas a *e-learning*, governo eletrônico, modelo de diálogo, humorista *Expert*, gerenciamento de rede, arquitetura modular adaptativa, entre outros, em que os *chatbots* não estão apenas oferecendo serviços úteis, mas interagem com os clientes e fornecem-lhes solução aos seus problemas por meio de AIML *chatbot*, substituindo a ação humana nos casos de perguntas que são frequentemente formuladas.

O uso de agente conversacional em lugar de FAQ (*Frequently Asked Questions*) é um dos mais usuais, o que torna possível obter respostas a partir de conversas com a máquina, de forma dialógica, usando linguagem natural (DALE, 2016). Para Leonhardt *et al.* (2003), os Agentes Conversacionais permitem que alunos se relacionem de forma mais humana com o computador, permanecendo 24 horas por dia à sua disposição. Conforme Dale (2016), o uso de Agentes Conversacionais constitui uma das grandes tendências da tecnologia na atualidade, sendo disponibilizado para interação com usuários de grandes corporações, como Apple (Siri), Amazon (Alexa), Microsoft (Cortana) e Google (Now).

Com o objetivo de auxiliar na implementação de Agentes Conversacionais, surgiram algumas plataformas de desenvolvimento, as quais disponibilizam o modelo *Software as a Service (SaaS)*<sup>1</sup>, como, por exemplo, *Pandorabots*, *Chatfuel*, *Botsify*, *Watson Conversation (IBM)*, *Rebot.me*, *Imperson*, *Wit.ai (Facebook)*, *Api.ai (Google)*, dividindo assim a responsabilidade entre o provedor de serviços e o cliente (RADZIWILL; BENTON, 2017). Por meio de cadastro e *login* de acesso, o usuário pode criar e personalizar seu próprio *chatbot on-line*.

Outros serviços que ajudam na implementação de Agentes Conversacionais são as soluções *PaaS (Plataform as a Service)*<sup>2</sup>, disponíveis para *download*, como *RiveScript*<sup>3</sup> e *Superbot*, os quais disponibilizam *frameworks* para a programação do *chatbot* em linguagem de *script* ou AIML. O *Superbot* é um *kit* de desenvolvimento disponibilizado pela própria comunidade A.L.I.C.E. (*Artificial Linguistic Internet Computer Entity*), fornecendo um caminho para a construção rápida da base de conhecimento por meio de planilhas.

---

1 No modelo SaaS, o fornecedor do *software* se responsabiliza por toda a estrutura necessária à disponibilização do sistema (servidores, conectividade, cuidados com segurança da informação), e o cliente utiliza o *software* via *Internet*, pagando um valor pelo serviço. O usuário não administra as características individuais da aplicação, exceto configurações específicas. Os desenvolvedores se concentram em atualização, e não na infraestrutura.

2 PaaS (Plataforma como serviço) é um ambiente de desenvolvimento e implantação completo na nuvem, com recursos que permitem fornecer tudo, de aplicativos simples, baseados na nuvem, a sofisticados aplicativos empresariais habilitados para a nuvem. Por meio de um provedor de serviços de nuvem, os recursos necessários são adquiridos em uma base pré-paga e acessados por uma conexão segura com a *Internet*.

3 Mais informações em <<https://www.rivescript.com/>>.

Estudos propostos por Fryer *et al.* (2017) e Crown *et al.* (2011) apontam que os Agentes Conversacionais oportunizam inúmeros benefícios à educação, essencialmente, com o oferecimento de informações, em forma de diálogo, tornando-se uma fonte potencial de motivação para uma comunicação sustentada para o aprendizado.

O atendimento interativo tem bastante valor como recurso educacional, conforme demonstrado por Bloom (1984), em seu estudo conhecido como *The 2 sigma problem*, no qual demonstra que a diferença na *performance* de alunos que tiveram atendimento individual em relação aos que receberam atendimento em sala de aula é de 2 desvios-padrão. Mas, como o atendimento individualizado tem alto custo associado, Bloom e outros pesquisadores investigaram soluções para tornar a atividade de aprendizagem mais interativa.

Bloom desafiou pesquisadores a encontrar métodos de instrução para grupos tão eficazes quanto as aulas individuais. Com o desenvolvimento da tecnologia da informação, tornou-se possível contar com novos recursos para simular efeitos de tutoria, sem o alto custo de fornecer um tutor presencial para cada aluno, e o agente conversacional é um recurso que pode contribuir com essa tutoria.

Nesse sentido, o agente conversacional constitui uma solução na categoria dos tutores inteligentes, embora um tanto mais limitada, e tem potencial para melhorar o desempenho acadêmico dos estudantes, fornecendo-lhes um atendimento que simula o que seria fornecido por um tutor individual.

O agente conversacional também pode ser utilizado nos Mundos Virtuais Imersivos. Para Greis e Reategui (2010), o conceito de Mundos Virtuais pressupõe características tanto de imersão como também de interação e navegação e, além disso, caracteriza-se por possibilitar: espaços partilhados, com vários usuários congregados num mesmo espaço/tempo; interface gráfica, disponibilizando um ambiente virtual tridimensional; imediaticidade, viabilizando uma interação em tempo real; interatividade, oportunizando as opções de interação e criação de objetos e conteúdos; socialização e comunidade, acarretando a criação de grupos e comunidades com eventuais interesses em comum.

Pode-se ampliar as funcionalidades dos Mundos Virtuais Imersivos com o apoio de um agente conversacional, potencializando ainda mais os recursos e tecnologias do Mundo Virtual Imersivo e de um agente conversacional, o que pode ser amplamente explorado, para diálogos, no processo de ensino-aprendizagem de qualquer conteúdo, abrangendo o ensino fundamental, médio, técnico, superior, bem como a formação continuada.

No âmbito do Projeto AVATAR, foram utilizados NPCs (*Non-Player Characters*) que podem ser instanciados, em qualquer ponto do Mundo Virtual e em diferentes condições, quando o avatar do usuário se aproximar de determinada localização ou quando alguma ação do usuário ocorrer – por exemplo, se ele terminou de realizar um experimento. Se toda a programação necessária para a capacidade de reação do agente tivesse que ser criada a partir do nada, usando a linguagem de *scripts*,

a tarefa seria imensa. Foi então investigada a ligação entre o NPC presente no Mundo Virtual com um *chatbot* externo, baseada na máquina de inferência ALICEbot.

Com base nesta ligação, foram desenvolvidos diferentes Agentes Conversacionais para uso no Mundo Virtual Imersivo e, mesmo em páginas *web*, que poderiam ser acessadas diretamente pelos estudantes ou em ambientes virtuais de aprendizagem como o *Moodle*. ATENA<sup>4</sup> (Agente Tutora ao Ensino e Navegação no Ambiente de Física) foi a agente conversacional desenvolvida para o Projeto AVATAR e pode ser também acessada diretamente pela interface *web*. Duas outras Agentes Conversacionais foram adicionalmente desenvolvidas: HIGIA (*Individual Habitat Individual* e Guia Interativo de Atitudes), usada em um projeto com o objetivo de motivação de indivíduos para a prática de atividade física (SGOBBI; TAROUCO; REATEGUI, 2017), e a Agente METIS (Mediadora de Educação em Tecnologia Informática e Socializadora), usada via interface *web* e no ambiente virtual de aprendizagem *Moodle* (LUCESI; SILVA; TAROUCO, 2018). Como a agente conversacional METIS passou por um processo de avaliação por usuários, será descrita em primeiro lugar.

### 3.1 AGENTE CONVERSACIONAL METIS

A agente conversacional METIS foi projetada para conversar com os alunos por meio de uma interface que simula uma conversação *on-line (chat)* entre pessoas. A implementação desta agente utiliza o

---

<sup>4</sup> Acesso para a agente ATENA em: <<http://avatar.cinted.ufrgs.br/atenaweb/gui/jquery/?>>.

sistema A.L.I.C.E. desenvolvido por Wallace (2003), no qual se organiza a base de conhecimento, notação baseada em XML (*eXtensible Markup Language*), AIML. A base de conhecimento do sistema foi alimentada por especialistas em mídias e informática na educação, facultando um conhecimento mínimo dos conteúdos e retorno positivo às dúvidas apresentadas pelos alunos.

Embora já tenha sido testada com estudantes, a agente encontra-se em constante desenvolvimento e aperfeiçoamento. A METIS é capaz de interagir usando o idioma Português; possui em sua base de conhecimento um conjunto de informações relacionadas com tecnologias educacionais e pode ser acessada em: <<http://avatar.cinted.ufrgs.br/metis/>>.

Conforme a Figura 3.1, constata-se que o estudante envia uma mensagem (“você poderia me explicar o que é licença de *software*”) ao agente, que responde com uma mensagem de texto que contém um conceito para o conteúdo questionado, além de um recurso de vídeo relacionado. Essa estrutura está organizada no formato de um *frame*, inserido na própria janela de *chat* com tamanho predefinido e, quando as respostas da agente são muito longas ou possuem muitos recursos, a janela de resposta apresenta uma barra de rolagem em sua lateral direita, permitindo, assim, que o usuário tenha acesso ao conteúdo apresentado pela agente METIS. Com relação aos conteúdos de vídeo, todos têm a opção de serem vistos em tamanho miniatura, o que é exibido como padrão dentro da área de resposta da agente, como podem ser visualizados em formato de tela cheia clicando na opção “Tela inteira” que se encontra disponível no vídeo.



Figura 3.1 - Interface web da agente conversacional METIS.



Fonte: <<http://avatar.cinted.ufrgs.br/metis/>>.

Vale salientar que outros recursos multimídia podem ser adicionados às respostas, por meio de *links* para repositórios digitais, bibliotecas eletrônicas, consultas na *web*, áudio e imagens. Sendo assim, a partir da possibilidade de vincular diversos tipos de recursos a serem compartilhados com os usuários, acredita-se que este modelo de entidade virtual que está sendo proposto venha incentivar o aluno a buscar mais informações, atuando como um companheiro virtual. A agente foi desenvolvida utilizando o *software Open Source Program-O* e possui uma interface em *Cascading Style Sheets (CSS)*, o que permite que a tela seja ajustada de acordo com o dispositivo, facilitando que estudantes interajam até mesmo por um aparelho celular.

A base de dados conversacionais da METIS foi construída com base em perguntas e respostas elaboradas por especialistas na área da educação, mídias e tecnologias. Posteriormente, quando os alunos estabeleceram “conversas” com METIS, iniciou-se uma inspeção dos registros das interações (*logs*), visando a identificar perguntas não respondidas

pelo sistema. Todas as perguntas dos usuários foram avaliadas por humanos especialistas e, conforme sua relevância, adicionadas à base de conhecimento do agente, ampliando sua capacidade de respostas.

Durante essa nova etapa de alimentação e realimentação da base de conhecimento da agente METIS, foram introduzidos novos recursos multimídia:

- adição de imagens nas respostas;
- inclusão de *links* para textos complementares;
- *links* para outros materiais, como vídeos e objetos de aprendizagem externos;
- múltiplas respostas a uma mesma pergunta (selecionadas de forma randômica, para que o comportamento do agente não fosse demasiado determinístico).

Como complemento à interação entre a agente METIS e os alunos, foi utilizada uma interface com jQuery, que consiste em uma biblioteca de funções *JavaScript*, interpretada pelos navegadores. Essa biblioteca possui funções que permitem a inserção de diversos recursos, como botões para alternância de imagens, busca de informações em banco de dados (para uso nas respostas fornecidas) e busca em outros mecanismos de pesquisa.

Porém, para a elaboração de uma base de conteúdos razoável, a fim de que um agente conversacional consiga interagir com o usuário e ele consiga responder à maioria de seus questionamentos e sinta-se amparado pela agente, profissionais conteudistas indicam que esta base de conhecimento deva conter uma massa crítica de conteúdo.

Pelo fato de as ferramentas de autoria para construir e editar a base de conhecimento existente serem complexas, demandando um padrão de conhecimento por parte do usuário tanto para se cadastrar quanto para utilizá-las, na construção de conteúdos (diálogos) para o Agente Conversacional, optou-se por construir uma ferramenta que subsidiasse essa carência percebida pelos pesquisadores e, assim, facilitasse a adaptação da agente a outros contextos. Dessa forma, surgiu o FastAIML<sup>5</sup>, uma ferramenta para apoiar a geração de base de conhecimento para *chatbots* educacionais (KRASSMANN *et al.*, 2017). A ferramenta FastAIML será apresentada na Seção 3.2.

## 3.2 O FASTAIML

Um dos desafios dos Agentes Conversacionais é ampliar a base de conhecimento para um melhor diálogo entre o aluno e o agente conversacional. No caso da Agente METIS, a linguagem de programação utilizada para programar e ampliar a base de conhecimento é a linguagem AIML, a qual tem um conjunto de *tags* e parâmetros relativamente complexos, com funções para memória temporal, variáveis e tópicos (WALLACE, 2003). O interpretador AIML funciona tentando combinar palavra com palavra, para se obter a correspondência-padrão mais longa e, portanto, tentar descobrir qual é a melhor (SATU *et al.*, 2015).

A concepção e o desenvolvimento do FastAIML surgiram em decorrência da dificuldade em encontrar ferramentas de autoria de auxílio a desenvolvedores, professores e pesquisadores, da área educacional,

---

5 <<http://avatar.cinted.ufrgs.br/fastaiml/>>.

na realização de atividades que envolvessem a construção de arquivos em AIML de forma intuitiva. Nesse sentido, a ferramenta FastAIML foi desenvolvida com o objetivo de auxiliar na composição da base de conhecimento de *chatbots* educacionais. O FastAIML permite que sejam gerados arquivos, na linguagem de marcação AIML, contendo categorias simples e/ou compostas, com a possibilidade de inclusão de recursos multimídia de forma intuitiva. Na categoria simples do FastAIML, tem-se a opção de uma pergunta e uma resposta e, na categoria composta, há a opção para uma pergunta e várias opções de respostas que podem ser exibidas randomicamente. A ferramenta encontra-se disponível para uso em: <<http://avatar.cinted.ufrgs.br/fastaiml/>>.

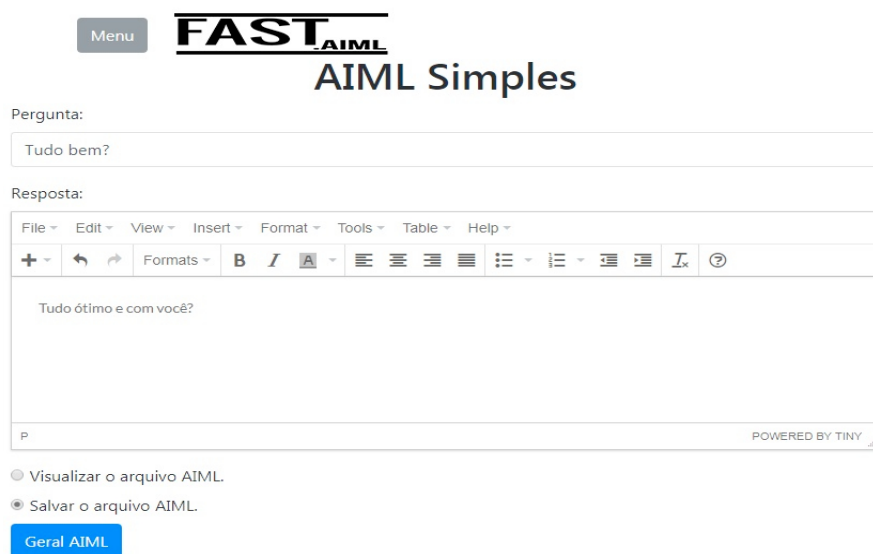
Destaca-se a função do FastAIML que propicia incluir, automaticamente, um rol predefinido de questões associadas a uma palavra-chave, ou seja, quando o aluno pesquisar utilizando o termo “mídias”, por exemplo, pode ter diversas opções de respostas relacionadas a mídias que aparecerão randomicamente, facilitando, assim, o processo de ampliação das bases de conhecimento.

A ferramenta é desenvolvida em PHP (*Hypertext Preprocessor*), que promove uma interface *web* totalmente leve e com *layout* responsivo. As funcionalidades para gerar AIML são escritas empregando a biblioteca DomDocument do PHP, que possui funções *built-in* a fim de escrever documentos XML, já colocando as informações dentro das *tags* AIML, como *category*, *template*, *pattern*, entre outras. Com essa biblioteca, é necessário apenas informar o que deve ir em cada *tag*, simplificando a inserção dos dados para o AIML.

O FastAIML conta com três menus principais contendo funcionalidades variadas. Em cada um deles, o usuário dispõe de opções para formatar a resposta a ser apresentada, podendo adicionar recursos multimídia como *hyperlinks*, imagens, vídeos, bem como realizar a formatação textual (e.g., negrito, itálico e tabelas), entre outras opções possibilitadas pela API (*Application Programming Interface*) TinyMCE (editor de textos *on-line*).

Os três menus principais do FastAIML são apresentados: 1 - AIML Simples; 2 - AIML Composto; 3 - Geração predefinida.

Figura 3.2 - Menus AIML Simples do FastAIML.

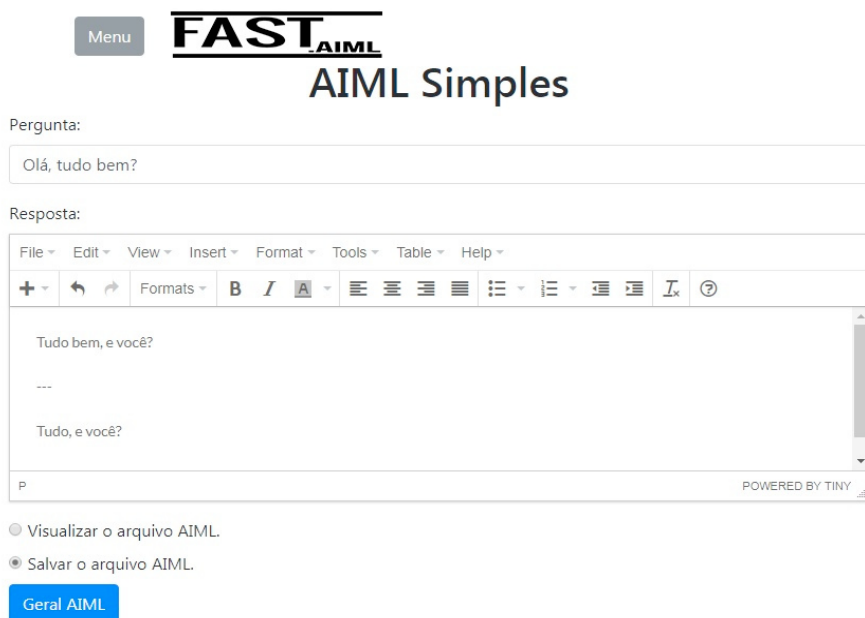


Fonte: <<http://avatar.cinted.ufrgs.br/fastaiml/>>.

Conforme a Figura 3.2, pode-se digitar uma pergunta e uma opção de resposta. Observam-se, ainda, as opções de formatação e os botões que permitem gerar o código AIML, ou seja, que preenchem os dados

nas lacunas (perguntas e respostas). Logo, os códigos são gerados automaticamente, assegurando grande agilidade na geração dos programas em AIML, os quais formarão as bases de conhecimento.

Figura 3.3 - Menus AIML Composto do FastAIML.



Fonte: <<http://avatar.cinted.ufrgs.br/fastaiml/>>.

A Figura 3.3 apresenta o *layout* ao qual se pode atribuir várias respostas relacionadas a uma questão. Por exemplo, na Figura 3.3, a pergunta que o aluno pode fazer para a agente METIS é “Olá, tudo bem?” e, nas opções de respostas, pode-se ter em um momento “Tudo bem, e você?” ou, ainda, em outro momento “Tudo, e você”? Observa-se que até 10 opções de respostas podem ser organizadas randomicamente.

### 3.3 AGATA

Ainda com o objetivo de auxiliar os usuários a construírem de forma mais ágil os conteúdos para a base de conhecimento dos Agentes Conversacionais, foi desenvolvido o AGATA (*Automatic Generation of AIML from Text Acquisition*).

O AGATA é um sistema de geração de base de conhecimento em AIML a partir da análise e tratamento de *corpus* linguístico<sup>6</sup>. Possui a funcionalidade de criar automaticamente arquivos na linguagem de marcação AIML, para a base de conhecimento de Agentes Conversacionais, baseada em um *corpus* linguístico. Foi desenvolvido por Krassmann e Flach (2018).

O sistema realiza a busca de palavras-chave que podem conter termos simples ou compostos em um *corpus* também indicado pelo usuário e, ainda, seleciona todas as frases que contêm esta palavra, entregando-a a uma nova tela. A partir desse resultado, o usuário seleciona as frases mais representativas das palavras-chave, clicando em caixas de seleção. Em uma nova tela, é possível editar cada uma das frases selecionadas, ajustando a formatação ou complementando com informações adicionais ou recursos multimídia. Após esse procedimento, um arquivo AIML contendo todas as *tags* da linguagem é gerado, pronto para ser enviado à base de conhecimento do *chatterbot*.

A Figura 3.4 apresenta um exemplo de resultado gerado pelo AGATA.

---

<sup>6</sup> *Corpus* linguístico é o conjunto de textos escritos e registros orais em determinada língua e que serve como base de análise. O estudo de *corpora* (plural de *corpus*) apresenta muitas vantagens.

Figura 3.4 - Exemplo de código gerado pelo AGATA.

```
<category>
  <pattern>MÍDIA ELETRÔNICA</pattern>
  <template>
    <random>
      <li>Podemos considerar como mídias eletrônicas os meios de comunicação que precisam de recursos eletrônicos para transmitir seu conteúdo até o usuário. </li>
      <li>As mídias eletrônicas são compostas por equipamentos de comunicação unilateral, onde o usuário é apenas receptor de uma determinada informação, como rádio, cinema, TV analógica, mídias graváveis. </li>
    </random>
  </template>
</category>
```

Fonte: <<http://www.ufrgs.br/avatar/agata>>.

Ainda, conforme a Figura 3.4, observa-se que dentro da *tag pattern* está a palavra-chave inserida pelo usuário e, dentro da *tag template*, há uma função de resposta aleatória (*tag random*) e *tags li* para cada uma das frases que foram selecionadas pelo usuário, após o sistema tê-las localizado e separado.

A funcionalidade do AGATA utiliza as tecnologias Python, Django, PHP e atua da seguinte forma:

1. O usuário seleciona um arquivo no seu computador e faz o *upload*, enviando-o ao sistema. Esse arquivo pode ser um livro, uma apostila, um artigo, uma revista etc., desde que possua frases que explicam um conceito e esteja no formato txt. Nesse momento, sugere-se que sejam retiradas as figuras, tabelas, quadros e gráficos do arquivo, realizando nele uma “limpeza”, para que permaneça somente o conteúdo textual.
2. O usuário insere de uma a três palavras-chave para busca e clica em enviar.
3. Em uma nova tela, as frases que contêm cada uma dessas palavras-chave são exibidas, ordenadas por termo buscado e ordem em que aparecem no *corpus*, conforme a figura abaixo:



Figura 3.5 - Resultados de frases selecionadas pelo usuário.

### Palavra-Chave: áudio

- O potencial de combinação entre as diferentes mídias, como texto, imagem, áudio e vídeo, acrescenta notório valor ao conceito de multimídia, no enriquecimento da comunicação e no compartilhamento de conhecimentos.
- Quanto aos recursos tecnológicos, a representação das mídias interativas pode ser composta de multimídias (áudio, vídeos, imagens..).
- Nas interfaces “computador-humano”, os sensores eletrônicos (captação de dados) como os de captação de áudio, captação de movimento, sensibilidade ao toque, estão cada vez mais presentes nos aparelhos eletrônicos, principalmente nos dispositivos pessoais como TV, relógio e celulares.
- Na educação, essas mudanças permitem transformações no modo de ensinar e aprender, conduzindo novas formas de interações, sendo na Educação a Distância (EAD), modalidade em contínuo crescimento, que ao longo dos anos tem sido a grande impulsionadora para o avanço da Tecnologia da Informação e Comunicação (TIC), possibilitando recursos que permitem criar materiais em áudio, vídeo, texto, hipertexto e animações, como também a possibilidade de integrar esses diversos recursos em um Ambiente Virtual de Aprendizagem (AVA).

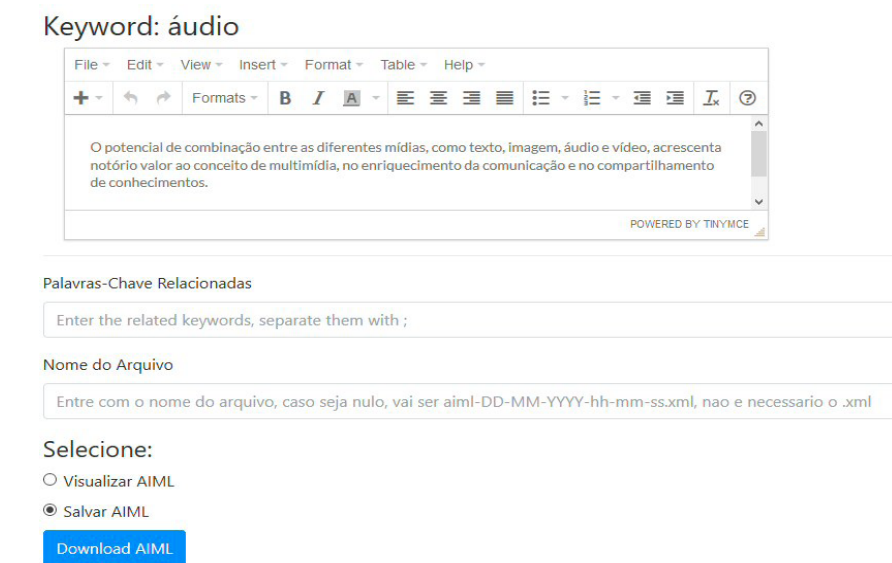
Enviar

Fonte: <<http://www.ufrgs.br/avatar/agata>>.

4. O usuário seleciona as frases que deseja utilizar, clicando em caixas de seleção. Nesse momento, o especialista de domínio, a pessoa que possui o conhecimento acerca do *corpus* em questão, utiliza sua *expertise* para indicar as frases mais significativas e que sejam condizentes com uma “resposta” ao usuário em uma conversa com o *chatbot*.

5. Em uma nova tela, as frases selecionadas pelo usuário agora aparecem cada uma delas com um editor, para que possam ser realizados ajustes no texto, alterações, formatações e inserção de recursos multimídia (vídeo, imagens, *hyperlinks*), conforme a Figura 3.6.

Figura 3.6 - Tela do AGATA com o resultado da mineração de frases no corpus com a expressão “áudio”.



Fonte: <<http://www.ufrgs.br/avatar/agata>>.

6. O usuário pode adicionar outras palavras-chave relacionadas que julgar serem pertinentes e que sejam direcionadas a essas respostas. Por exemplo, junto à lista de frases referentes à palavra-chave “áudio”, o especialista pode acrescentar o termo “som”, se as frases estiverem relacionadas a essa temática. Dessa forma, quando o termo “som” for “pronunciado” ou estiver contido no meio de uma frase emitida pelo estudante, uma das frases referentes ao termo “áudio” poderá ser enviada como resposta.

7. Após o usuário realizar os ajustes necessários e clicar em enviar, a nova tela mostra o arquivo AIML contendo todas as *tags*, permitindo que o usuário tanto visualize essa saída como faça o *download* do arquivo. Abaixo, a Figura 3.7 apresenta a captura de tela que ilustra um exemplo de saída do sistema.

Figura 3.7 - Tela do AGATA com exemplo de saída do sistema.

```
-<aiml encoding="UTF-8" version="3.0">
- </--
  Copyright ©2017 - FAST.aiml
  URL: http://avatar.cinted.ufrgs.br/fastaiml/
  Powered by Ailane Loureiro Krassmann <alkrassmann@gmail.com >
  Alvaro S. P. Silva <alvarosilva@ig.com >
  Project coordinator: Liane Margarida Rockenbach Tarouco <liane at penta.ufrgs.br >
-->
-<category>
  <pattern> audio </pattern>
-<template>
  -<random>
    -<li>
      <p>A inclusão da tecnologia extrapola o espaço da sala de recursos multifuncionais, adentra na sala de aula e nos demais espaços sociais, principalmente na vida do sujeito sem restringir-se apenas à perspectiva educacional, como veremos no estudo que detalhamos sobre o uso da audiodescrição num espaço educativo.</p>
    </li>
    -<li>
      <p>O potencial de combinação entre as diferentes mídias, como texto, imagem, áudio e vídeo, acrescenta notório valor ao conceito de multimídia, no enriquecimento da comunicação e no compartilhamento de conhecimentos.</p>
    </li>
  </random>
-</template>
</category>
```

Fonte: <<http://www.ufrgs.br/avatar/agata>>.

Constata-se que o sistema AGATA tem o grande potencial de auxiliar professores e pesquisadores no desenvolvimento da base de conhecimento de Agentes Conversacionais, de forma simples e rápida, pois os códigos são gerados automaticamente na linguagem AIML, a partir de um *corpus* linguístico. Esse fator aumenta a facilidade para gerar a base de conhecimento de um agente conversacional para outros contextos, de forma simples e rápida, a partir de um *corpus* linguístico. Sendo assim, o agente conversacional pode ter conhecimento sobre o que é abordado em um livro de História, Biologia, em algum artigo científico ou qualquer outro documento que contenha informações específicas sobre determinado nível de conhecimento para futuros diálogos.

O sistema AGATA foi usado para construir a base de conhecimento da agente conversacional ATENA na área de Física.

### 3.4 APRENDIZAGEM COLABORATIVA COM ENTIDADES VIRTUAIS

Conforme apresentado no início deste capítulo, é possível ampliar as funcionalidades dos Mundos Virtuais Imersivos com o apoio de um agente conversacional. A construção dos agentes conversacionais pode ser facilitada usando ferramentas de autoria tais como FastAIML e AGATA, que ajudam na ampliação da base de conhecimento do agente conversacional.

Nesta seção, serão abordadas as possibilidades de aprendizagem colaborativa no Mundo Virtual Imersivo. O Mundo Virtual Imersivo (MVI) oportuniza o uso de entidades que impersonam agentes sintéticos, implementados mediante o uso de *Non-Player Character* (NPC), ou em português “Personagem não jogável”. Portanto, o NPC é o personagem, dentro do MVI, que propicia explorar as possibilidades e recursos desenvolvidos no respectivo MVI. Logo, o valor do recurso viabilizado pela implementação de NPC no MVI, especialmente com as funcionalidades ampliadas de respostas, quando esta entidade é conectada a um agente conversacional externo, suscita mais e mais flexíveis respostas.

O NPC elicitou a oportunidade de enriquecer a experiência de aprendizagem do usuário neste ambiente, mas, ao mesmo tempo, fez emergir a dificuldade de implantar a programação dos *scripts* que permitiam ao NPC prover respostas mais flexíveis ao aluno, considerando que há algumas restrições na plataforma MVI. Uma solução para superar esta dificuldade foi encontrada com o uso de *chatbots* já disponíveis em *software* e que possibilitaram a implantação de um agente conversacional sem necessidade de programação extra, visto que o MVI tem

ferramentas e recursos tais como a inserção de *links* que possibilitam o acesso a partir do MVI para o agente conversacional. Isso é importante pois, embora o MVI promova o acesso por mais de um usuário simultaneamente, o que ensejaria uma colaboração e troca de ideias, em muitas ocasiões, o aluno está sozinho no ambiente e pode se sentir inseguro e desorientado. Assim, o agente conversacional pode aparecer (disparado pela sinalização de que o usuário se aproximou de determinada área) e apresentar orientações ou responder a perguntas do usuário, originando um companheiro virtual que acompanha o avatar do usuário enquanto ele se movimenta dentro de determinado contexto.

A Figura 3.8 apresenta o cenário de um Mundo Virtual. Na imagem, observa-se que é um cenário para o ensino de Eletricidade.

Figura 3.8 - Cenário com avatar do usuário e agente implementado por NPC.



Fonte: <<http://www.ufrgs.br/avator>>.

Essa colaboração encontra subsídio teórico na teoria sociointeracionista de Vygotsky (1978), que propõe que a aprendizagem é um processo social e que o desenvolvimento do estudante é influenciado pelos seus pares nesse processo de crescimento cognitivo.

A Zona de Desenvolvimento Proximal (ZDP) é um conceito básico desta teoria, definida como a distância entre o nível de desenvolvimento real, que se costuma determinar por meio da solução independente de problemas, e o nível de desenvolvimento potencial, determinado pela solução de problemas sob a orientação de um adulto ou em colaboração com companheiros mais capazes. Com base nesse conceito, as interações sociais passam a ser entendidas como relevantes para a produção de conhecimento por parte dos alunos, particularmente aquelas que admitem o diálogo, a cooperação e a troca de informações mútuas e o confronto de pontos de vista divergentes.

Assim, um agente conversacional com uma boa base de conhecimento e, também, quando conectado a uma base de pesquisa externa, a partir de um MVI, tem o potencial para atuar como um companheiro virtual para o estudante, oportunizando diálogos e respostas flexíveis e realizando intervenções que afetam a ZDP com o fim de atuar de diversas maneiras, tais como:

- Orientar sobre as atividades a serem realizadas (especialmente relevantes em MVI cujos estudantes podem sentir-se inseguros e desorientados em um ambiente em que há muito a ser explorado).
- Responder a perguntas relacionadas ao campo conceitual em ação.

- Formular questionamentos ao estudante com vista a instigar reflexão e metachecimento.
- Sugerir *sites* e vídeos com informações adicionais relacionadas ao contexto abordado.

Acredita-se que com essas estratégias seja possível criar um ambiente de aprendizagem apoiado na teoria sociointeracionista de Vygotsky, capaz de ensejar aprendizagem colaborativa em MVI com o apoio de Agentes Conversacionais como substitutos de companheiros ou tutores reais. Desse modo, as interações sociais passam a constituir elementos relevantes para a produção de conhecimento de forma colaborativa, por meio do diálogo, da troca de informações e do confronto de pontos de vista divergentes.

Os pares não aprendem porque são dois, mas porque realizam atividades que desencadeiam mecanismos de aprendizagem específicos. Isso inclui atividades ou mecanismos realizados individualmente, uma vez que a cognição individual não é suprimida em interação entre pares. Além disso, a interação entre os indivíduos gera atividades extras (explicação, desacordo, regulação mútua) que acarretam mecanismos cognitivos adicionais, tais como elicitación do conhecimento, internalização, carga cognitiva reduzida etc. (DILLENBOURG, 1999).

Ao usar o MVI e o agente conversacional de forma simultânea e com o mesmo propósito, potencializam-se tais mecanismos de aprendizagem específicos, propostos por Dillenbourg (1999), visto que pelo MVI o aluno poderá entrar em modo de imersão e, com o apoio do agente conversacional, gerar um universo de possibilidades para a interação, ocasionando o desenvolvimento de mecanismos cognitivos adicionais.

Embora o uso de um agente conversacional atrelado ao MVI possa ajudar em um processo de ensino-aprendizagem, há limitações nos diálogos entre os estudantes e o respectivo agente. Essas limitações relacionadas à implementação impedem um diálogo mais rico e flexível, mas promovem uma estratégia de reação do agente a questões apresentadas pelo usuário. Mesmo assim, seu valor como recurso educacional foi percebido pelos usuários tal como comentado na seção a seguir.

### **3.5 AMBIENTE, PÚBLICO E ANÁLISE DA REAÇÃO DE USUÁRIOS AO USO DO METIS**

Esta seção apresenta o estudo que foi realizado em uma disciplina na modalidade de ensino a distância. Tal disciplina é ofertada, como eletiva para diversos cursos de licenciatura e, normalmente, é cursada por alunos de áreas distintas, como Matemática, História, Geografia, entre outras.

Especificamente para este estudo, foi investigada uma turma do primeiro semestre de 2017, composta por onze alunos, que tiveram acesso ao Agente Conversacional METIS. Com o objetivo de analisar com mais precisão a relação entre METIS e a interação dos alunos na disciplina, os dados foram comparados com outros coletados em uma segunda turma. A segunda turma, composta por dez alunos, cursou a mesma disciplina no semestre anterior, ou seja, segundo semestre de 2016, período no qual a agente conversacional ainda não estava disponível à interação com os alunos.



As análises foram realizadas com uma versão de um agente conversacional METIS acessado diretamente, via interface *web*, mas com funcionalidades equivalentes ao que pode ser disponibilizado no âmbito do MVI. A partir do momento em que os alunos estabeleceram “conversas” com METIS, iniciou-se a inspeção dos registros das interações (*logs*), visando a identificar perguntas não respondidas pelo sistema. Tais perguntas foram avaliadas e, conforme sua relevância, adicionadas à base de conhecimento do agente, ampliando sua capacidade de respostas.

Durante a alimentação e realimentação da base de conhecimento da agente METIS, foram introduzidos novos recursos multimídia: (i) adição de imagens nas respostas; (ii) inclusão de *links* para textos complementares; (iii) *links* para outros materiais, como vídeos e objetos de aprendizagem externos; (iv) múltiplas respostas para uma mesma pergunta (selecionadas de forma randômica, a fim de que o comportamento do agente não fosse demasiado determinístico).

As conversas registradas pelo agente conversacional ocorreram no período de 8 semanas. Nesse período, METIS apresentava, aproximadamente, 2.800 categorias (atualmente, conta com 4.300 categorias em sua base de conhecimento). Foram analisadas em detalhe 11 conversas, originárias de 11 endereços IP distintos, confirmando usuários diferentes. Os *logs* registrados resultaram em um total de 117 linhas, com média de 10 linhas por conversa.

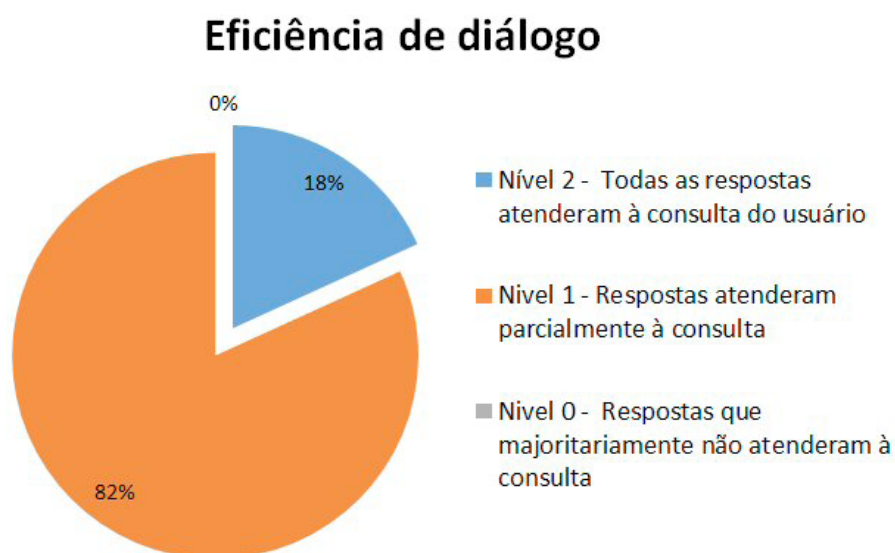
### 3.5.1 Ferramentas e coleta dos dados

A coleta de dados foi realizada por meio de três instrumentos, conforme descrito:

- Análise dos *logs* de conversa registrada pelo agente conversacional: refere-se aos *logs* de conversa registrada entre METIS e os alunos, tendo como objetivo avaliar a eficiência dos diálogos estabelecidos. Assim sendo, 11 conversas foram analisadas e cada uma foi enquadrada em um nível de qualidade, concernente à capacidade de responder ou não ao usuário satisfatoriamente.
- Análise de acesso aos materiais de apoio da disciplina: estes dados foram coletados a partir do Ambiente Virtual de Aprendizagem – Moodle – no qual foi desenvolvida a disciplina, para a análise comparativa entre as duas turmas. Para levantamento e comparação dos acessos dos alunos aos materiais de apoio da disciplina, foi empregado um *plugin* disponibilizado no Moodle. Esse recurso permite a extração do número de visualizações nos recursos disponíveis no ambiente, como tutoriais, apresentações, *links* externos, entre outros.
- Questionário de avaliação do agente: neste sentido, foi proposto aos alunos da turma investigada um questionário *on-line* composto por 26 questões objetivas, organizadas em 5 categorias: aprendizagem, confiabilidade, relações, engajamento e visão geral, com opção de resposta tipo escala Likert (LIKERT, 1932). O questionário em questão foi adaptado de Wechsung *et al.* (2013).

Tais logs, inicialmente, foram classificados e, depois, cada diálogo (correspondente a um usuário diferente) foi avaliado quanto à sua eficiência. Por isso, foi adaptada por De Gasperis *et al.* (2013) uma métrica para aferir a exatidão das respostas oferecidas pela agente conversacional, apresentando os seguintes níveis de qualidade:

Figura 3.9 - Gráfico da Eficiência de diálogo a partir dos registros de logs.



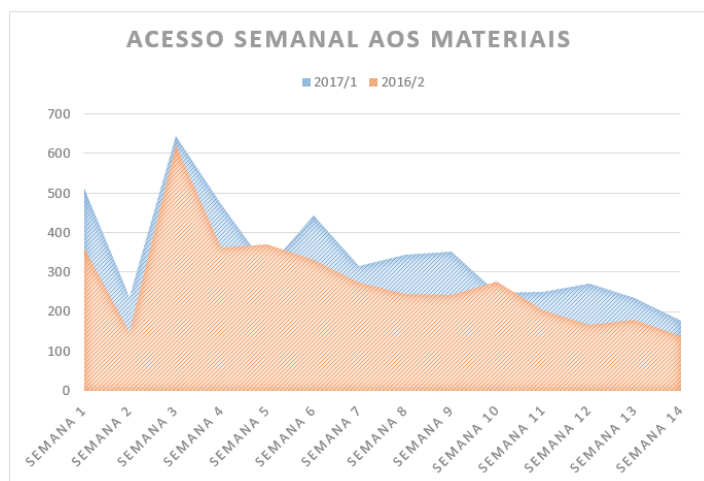
Fonte: Lucchesi *et al.* (2018).

Quanto mais baixo o nível de qualidade, menos o agente entendeu e respondeu satisfatoriamente ao aluno durante o diálogo. A qualidade aumenta à medida que o aluno tenha recebido majoritária ou exclusivamente respostas que atenderam a suas questões.

A fim de se obter a relação detalhada sobre o acesso dos alunos aos materiais de apoio da disciplina, empregou-se um *plugin* do Moodle denominado *Analytics Graphs*<sup>7</sup>. Então, foi possível extrair o número total de visualizações em cada tópico (semanal) estudado, comparando-o nas duas turmas.

Dos períodos analisados e comparados entre as duas turmas, em somente dois deles os acessos ocorreram em maior quantidade, na turma do segundo semestre de 2016, correspondente aos alunos sem acesso à METIS, conforme a Figura 3.10. Dessa forma, apresentando 12 tópicos com maior número de acessos, a turma que cursou a disciplina no segundo semestre de 2017 (com acesso à METIS) teve um aumento de, aproximadamente, 22% no total de acessos aos materiais de apoio.

Figura 3.10 - Gráfico dos dados de acesso aos materiais de apoio da disciplina.



Fonte: Lucchesi *et al.* (2018).

7 Tal recurso oferece o número de visualizações efetuadas, em cada repetição da palavra recurso “material de apoio”, disponibilizado na disciplina, que inclui apresentações, vídeos, textos de apoio, tutoriais, *links* externos etc. (<[https://moodle.org/plugins/block\\_analytics\\_graphs](https://moodle.org/plugins/block_analytics_graphs)>).

O instrumento (iii) é composto de 26 questões objetivas, organizadas em 5 categorias: aprendizagem, contendo 13 questões; confiabilidade, com 5 questões; relações e engajamento, com três questões cada; e visão geral, apresentando 2 questões. Todas as questões apresentaram opção de resposta tipo escala Likert (LIKERT, 1932) de cinco pontos, indicando como extremos: discordo totalmente (1) e concordo totalmente (5). A utilização do questionário visou a complementar os dados coletados quanto à usabilidade do agente e seu efeito no engajamento dos estudantes.

Dentro do escopo da pesquisa, oito (8) dos onze (11) alunos da disciplina responderam ao questionário *on-line*, no período de 4 semanas. A Tabela 3.1 apresenta uma síntese das pontuações obtidas com relação às respostas dos alunos, mostrando a porcentagem de respostas a cada categoria abordada no questionário. Optou-se por agrupar o resultado das respostas por categoria para tornar a análise mais sucinta.

Tabela 3.1 - Síntese das pontuações obtidas em relação às respostas dos alunos.

Categoria	Porcentagem por categoria				
	1	2	3	4	5
Aprendizagem	7,69%	37,36%	30,77%	14,29%	9,89%
Confiabilidade	11,43%	11,43%	14,29%	37,14%	25,71%
Relações	19,05%	9,52%	19,05%	38,10%	14,29%
Engajamento	14,29%	14,29%	23,81%	33,33%	14,29%
Visão geral	7,14%	14,29%	42,86%	21,43%	14,29%

Fonte: Lucchesi *et al.* (2018).

## 3.5.2 Análise e discussão de resultados

A fim de potencializar a explanação dos resultados obtidos e a análise de cada instrumento utilizado para a coleta de dados, estas serão apresentadas na subseções seguintes. A primeira subseção apresentará os dados do instrumento (i) análise dos *logs* de conversa registrada pelo *chatbot*. A subseção seguinte tratará do instrumento (ii) análise de acesso aos materiais de apoio da disciplina e, por último, serão expostos os dados do instrumento (iii) questionário de avaliação do agente.

### 3.5.2.1 Análise do desempenho da Agente Conversacional

A proposta era verificar se, com essa quantidade de categorias inseridas no agente, ele conseguiria manter um “diálogo” de qualidade com o aluno, sem que este perdesse, em seguida, o interesse em questioná-lo sobre os conteúdos relacionados à disciplina do curso.

A partir da verificação dos *logs*, constatou-se que a maior parte dos diálogos enquadrou-se no nível 1 de qualidade, no qual a agente conversacional forneceu respostas que atenderam, parcialmente, à consulta dos alunos, como pode ser observado na Figura 3.9. Destaca-se que, em parte, os diálogos encontrados no nível 1 registraram questionamentos sobre conceitos não incluídos pela agente conversacional, ou por tratarem de assuntos que não são o foco trabalhado, ou por ainda não terem sido inseridos na base de dados, que se encontra em construção. Um resultado bastante significativo neste estudo foram os questionamentos enquadrados no nível 2 de qualidade, no qual todas as respostas recebi-

das durante o diálogo foram satisfatórias, conforme apontado na Figura 3.9. Isso significa que a agente conseguiu responder aos questionamentos dos alunos de forma satisfatória.

Ressalta-se que as respostas oferecidas pela agente METIS, concernentes a conteúdos específicos, conceitos e *softwares* relacionados à disciplina e ao foco da agente conversacional, incluem, além de definições, indicações de materiais disponíveis na disciplina e/ou fontes de pesquisa externas, incentivando os alunos a aprofundarem os assuntos questionados. A indicação de materiais é uma das vantagens destacadas por Leonhardt *et al.* (2003) sobre o uso de *chatbot* na educação. Neste contexto, a subseção seguinte analisa a influência da agente METIS quanto à atuação dos alunos na disciplina.

Outro ponto a ressaltar trata do aumento, ainda que moderado, da média geral alcançada pelas duas turmas na disciplina. Enquanto a turma de 2016 concluiu o semestre com média 8,2, a turma de 2017 obteve a média 8,6. Infere-se que este aumento esteja relacionado à maior inspeção dos materiais de apoio da disciplina, pois, quanto mais aprofundados os recursos, melhor se torna o embasamento à realização das atividades propostas.

Nesse contexto, ponderando que a agente METIS indica recursos da disciplina ao responder dúvidas dos alunos, considerou-se que possa ter contribuído nestes resultados. Ao encontro de Da Silva *et al.* (2014), no que diz respeito ao potencial dos Agentes Conversacionais, para além da habilidade de conversar, observou-se a possibilidade de engajar os alunos na pesquisa dos materiais disponibilizados, considerando que a METIS atuou no papel de um tutor inteligente, disponível vinte e qua-

tro horas por dia, para o aluno conversar e esclarecer suas dúvidas, especificamente, sobre o assunto abordado no curso, obtendo respostas em forma de texto e sugestões de vídeos sobre o tema abordado.

Com relação ao nível de conhecimento da agente, os alunos consideraram que a agente METIS possui um bom nível de conhecimento, não apenas sobre os conteúdos relacionados à disciplina, mas também sobre outros assuntos relacionados, e que as trocas realizadas contribuíram à sua aprendizagem. Eles também consideraram que as respostas, informações e materiais indicados pelo *chatbot* em resposta aos seus questionamentos foram muito confiáveis como respostas e material de consulta.

No que se refere à desenvoltura da agente para desenvolver uma conversa muito próxima à linguagem natural dos participantes, demonstrou-se amistosa aos questionamentos realizados; foi considerada muito amigável e atenciosa, assim como preocupada em entender as perguntas, para respondê-las de acordo com o conteúdo de sua base de conhecimento. Quanto ao fato de a agente METIS ter mantido a sua atenção, demonstrando interesse sobre o conteúdo apresentado, como respostas às suas indagações, ela promoveu, desta forma, novos questionamentos por parte dos alunos.

Mesmo que o *chatbot* tenha se mostrado amistoso com os participantes durante as trocas e indicações de materiais, ele precisa ser aperfeiçoado, a fim de envolver o participante, para que as trocas de informações sejam mais abrangentes e contínuas. Esse tipo de engaja-



mento dar-se-á a partir do tipo de respostas fornecidas pela agente conversacional, nas quais se apresentam indagações ou questões reflexivas sobre o tema abordado pelo participante.

No tocante à experiência dos alunos como um todo, relativamente a utilizar a agente como uma ferramenta de apoio à aprendizagem dentro da disciplina de graduação, no sentido de realizarem trocas sobre os conteúdos estudados e correlacionados, os alunos consideraram que a agente METIS teve um papel relevante no processo de promover e incentivar a aprendizagem. Eles informaram que esperam ter uma nova oportunidade de utilizá-la em outras disciplinas durante o curso, para auxiliá-los a diminuir suas dúvidas por meio de materiais e recursos apresentados por ela, além de auxiliá-los com suas dicas e sugestões.

Cabe lembrar que, no período no qual foi realizada a aplicação do questionário, a base de conhecimento da agente METIS possuía, aproximadamente, 2.800 categorias, ou seja, encontrava-se em fase bastante inicial. Periodicamente, novas categorias foram implementadas e categorias já existentes foram aperfeiçoadas para melhor desempenho.

### **3.6 CONSIDERAÇÕES FINAIS**

Neste capítulo, foram abordados conceitos de Agentes Conversacionais, os quais foram descritos com os principais recursos e linguagens de programação por eles utilizados, bem como as ferramentas FastAIML e AGATA, desenvolvidas, especificamente, para otimizar e automatizar o processo de geração da base de conhecimento. O agente conversacional constitui uma solução na categoria dos tutores inteligentes, embora

com algumas limitações, e tem potencial para melhorar o desempenho acadêmico dos estudantes, fornecendo-lhes um atendimento que simula o que seria o atendimento prestado por um tutor individual.

Os Mundos Virtuais Imersivos são ambientes que permitem a simulação de diversas situações do mundo real em um ambiente virtual, com a oportunidade de vários acessos simultâneos, ao viabilizar aos estudantes a exploração do Mundo Virtual e a interação com diversos recursos e ferramentas, tais como animações e objetos 3D, havendo, ainda, a probabilidade de se relacionarem com os demais usuários conectados.

A possibilidade de ampliar as funcionalidades de um Mundo Virtual Imersivo atribuindo os recursos, funcionalidades e potencialidades de um agente conversacional também foi demonstrada neste capítulo, considerando que a integração com *chatbots* externos potencializa ainda mais os recursos e tecnologias dos Mundos Virtuais Imersivos.

Para tanto, percebeu-se que ampliar as funcionalidades foi importante, porque, embora o Mundo Virtual Imersivo permita o acesso por mais de um usuário simultaneamente, o que ensejaria uma colaboração e troca de ideias, em muitas ocasiões, o aluno está só no ambiente e pode se sentir inseguro e desorientado; assim, o apoio fornecido por um agente conversacional pode orientar e esclarecer dúvidas.

O experimento realizado em uma disciplina do curso de Licenciatura em Pedagogia, na modalidade de ensino a distância, oportunizou aplicar os testes e coletar resultados, a fim de proporcionar uma boa reflexão, envolvendo a teoria e a prática sobre Mundos Virtuais Imersivos

e Agentes Conversacionais, associada a uma metodologia qualitativa e quantitativa, a partir da análise dos *logs* coletados durante a conversa com a METIS e os alunos.

Na análise dos *logs*, constatou-se que a maior parte dos diálogos, ou seja, 82% teve pontuação nível 1, ou seja, as respostas a partir dos *logs* atenderam parcialmente à consulta do usuário. Salienta-se que aqui não houve nenhuma resposta em que o agente conversacional não atendesse à consulta. Conclui-se que o diálogo entre o agente conversacional e os estudantes foi satisfatório neste experimento, não havendo nenhum *feedback* negativo. Logo, infere-se que o diálogo do agente METIS teve boa eficiência.

Na Análise de Acesso aos materiais de Apoio da disciplina, o *Plugin Analytics Graphs* trouxe o número de visualizações efetuadas nos respectivos recursos disponibilizados, sendo possível extrair o número de visualizações de cada tópico por cada aluno.

Nesta análise, merece atenção, em nossas considerações finais, a comparação realizada entre uma turma que utilizou o agente conversacional e outra que não o utilizou. Assim, registrou-se que a turma que utilizou o agente conversacional obteve um aumento no total de acessos aos materiais disponibilizados no *Moodle*.

Além disso, outro ponto que se destaca é a média geral alcançada pelas duas turmas na disciplina: a turma que utilizou o Agente Conversacional obteve nota superior em relação aos alunos que não o utilizaram. Dessa forma, tanto o aumento do acesso aos materiais disponibilizados quanto a média geral da turma permite-nos concluir que utilizar um agente conversacional, juntamente a um Ambiente Virtual de Aprendi-

zagem como uma ferramenta de apoio à conversação entre os alunos, pode ajudar a despertar o seu interesse em visualizar os recursos disponíveis em um ambiente, como também tornar a aprendizagem mais significativa, visto que há um agente conversacional 24 horas por dia à disposição do aluno para esclarecer suas dúvidas. O agente conversacional, além disso, propiciou o engajamento dos alunos concernente aos materiais disponibilizados, conforme descrito por Da Silva *et al.* (2014).

O terceiro instrumento de análise de dados foi um Questionário de Avaliação do Agente, composto de 26 questões objetivas, organizadas em cinco categorias: **Aprendizagem, Confiabilidade, Relações, Engajamento e Visão Geral.**

Na categoria **Aprendizagem**, pontuou-se que os alunos não perceberam o impacto na Aprendizagem; por outro lado, houve um aumento no acesso do material e na média final, o que, indiretamente, colaborou para a aprendizagem dos alunos. A categoria **Confiabilidade** demonstrou que os alunos, realmente, confiaram nas informações dos materiais disponibilizados. Na categoria **Relações**, constatou-se um diálogo amigável entre o agente METIS e os respectivos alunos. Na categoria **Engajamento**, também, mostra-se que a METIS obteve um bom engajamento nos diálogos com os alunos durante o curso. Pela categoria **Visão Geral**, estimou-se que o maior percentual apresentado nas respostas foi o de neutralidade.

Por fim, pode-se deduzir que os resultados obtidos no respectivo estudo, a partir do experimento realizado, foi muito positivo, considerando os três instrumentos de análise, cujos resultados foram satisfa-

tórios, e não se encontraram resultados negativos em relação ao uso da agente METIS. Vale ressaltar que a agente METIS está em constante desenvolvimento e aperfeiçoamento.

Finalmente, mesmo em fase inicial, durante este estudo, foi possível apurar que a agente METIS alcançou resultados satisfatórios quanto à eficiência de diálogo e influência no engajamento dos alunos na turma investigada.

## REFERÊNCIAS

BLOOM, Benjamin S. The 2 sigma problem: the research for methods of group instruction as effective as one-to-one tutoring. **Educational researcher**, v. 13, n. 6, p. 4-16, 1984.

CROWN, Stephen; FUENTES, Arturo; JONES, Robert; NAMBIAR, Rajiv; CROWN, Deborah. Anne G. Neering: interactive chatbot to engage and motivate engineering students. **Computers in education journal**, v. 21, n. 2, p. 24-34, 2011.

DA SILVA, Alexandre Sawczuk; GAO, Xiaoying; ANDREAE, Peter. Wallace: incorporating search into chatting. In: PHAM, Duc-Nghia e PARK, Seong-Bae (ed.). **LECTURE notes in computer science. Including subseries Lecture notes in artificial intelligence and Lecture notes in bioinformatics**. New York: Springer Nature, 2014. v. 8862, p. 842-848.

DALE, Robert. The return of the chatbots. **Natural language engineering**, v. 22, n. 5, p. 811-817, Sept. 2016.

DE GASPERIS Giovanni; CHIARI, Isabella; FLORIO, Niva. AIML knowledge base construction from text corpora. In: YANG, Xin-She (ed.). **Artificial intelligence, evolutionary computing and metaheuristic**. Berlin: Springer, 2013. (Studies in computational intelligence, 427). p. 287-318.

DILLENBOURG, Pierre. What do you mean by collaborative learning? In: DILLENBOURG, Pierre. **Collaborative-learning: cognitive and computational approaches**. Oxford: Elsevier, 1999. p. 1-19.

FRYER, Luke K.; AINLEY, Mary; THOMPSON, Andrew; GIBSON, Aaron; SHERLOCK, Zelinda. Stimulating and sustaining interest in a language course: an experimental comparison of chatbot and human task partners. **Computers in human behavior**, v. 75, n. C, p. 461-468, Oct. 2017.

GREIS, Luciano Kercher; REATEGUI, Eliseo. Um simulador educacional para disciplina de física em mundos virtuais. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 8, n. 2, jul. 2010.

KRASSMANN, Aliane Loureiro; FLACH, João Marcos. **AGATA – Automatic Generation of AIML from Text Acquisition**. Disponível em: <<http://agata.pgie.ufrgs.br/>>. Acesso em: 10 out. 2018.

KRASSMANN, Aliane Loureiro; HERPICH, Fabrício; SILVA, Álvaro Souza Pereira da; SILVA, Anita Raquel da; ABREU, Cristiane de Souza; SCHMITT, Marcelo Augusto Rauh; BERCHT, Magda; TAROUCO, Liane Margarida Rockenbach. FastAIML: uma ferramenta para apoiar a geração de base de conhecimento para *chatbots* educacionais. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 15, n. 2, 2017.

LEONHARDT, Michelle Denise; CASTRO, Daiane Dorneles de; DUTRA, Renato Luís de Souza; TAROUCO, Liane Margarida Rockenbach. Elektra: um *chatbot* para uso em ambiente educacional. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 1, n. 2, 2003.

LÉVY, Pierre. **A inteligência coletiva**. São Paulo: Loyola, 1997.

LIKERT, Rensis. A technique for measurement of attitudes. **Archives of psychology**, p. 5-55, 1932.

LUCCHESI, Ivana Lima; SILVA, Anita Raquel da; ABREU, Cristiane; TAROUCO, Liane Margarida Rockenbach. Avaliação de um *chatbot* no contexto educacional: um relato de experiência com Metis. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 16, n. 1, 2018. Disponível em: <<https://seer.ufrgs.br/renote/article/view/85903/49294>>. Acesso em: 7 out. 2018.

OPEN WONDERLAND. **About Open Wonderland**. Open Wonderland Foundation, Eden Prairie, MN, 2018. Disponível em: <<http://openwonderland.org/index.php/about/about-project-wonderland>>. Acesso em: 5 nov. 2018.

PRIMO, Alex Fernando Teixeira; COELHO, Luciano Roth; PAIM, Marcos Flávio Rodrigues; REICHEL, Dagmar. Júnior, um *chatbot* para educação a distância. In: CONGRESSO IBEROAMERICANO DE INFORMÁTICA EDUCATIVA, 5.,

2000, Viña del Mar, Chile. **Anais** [...]. Viña del Mar: Rede Iberoamericana de Informática Educativa, 2000. p. 2006-2016.

RADZIWILL, Nicole M.; BENTON, Morgan C. Evaluating quality of chatbots and intelligent conversational agents. **arXiv**, Cornell University, Ithaca, arXiv:1704.04579, 2017.

SATU, Shahriare; PARVEZ, Hasnat; MAMUM, Shamim Al. Review of integrated applications with AIML based chatbot. *In: INTERNATIONAL CONFERENCE ON COMPUTER & INFORMATION ENGINEERING*, 1., 2015, Rajshahi, Bangladesh. **Proceedings** [...]. USA: Institute of Electrical and Electronics Engineers – IEEE, 2015. Disponível em: <<https://goo.gl/M2QFop>>. Acesso em: 25 ago. 2017.

SECOND LIFE. **What is Second Life?** Linden Research, Boulder, CO, USA, 2018. Disponível em: <<http://secondlife.com/whatis/?lang=en-US>>. Acesso em: 4 ago. 2018.

SGOBBI, Fabiana Santiago; TAROUÇO, Liane Margarida Rockenbach; REATEGUI, Eliseo. The pedagogical use of the Internet of Things in virtual worlds to encourage a behavior change in obese individuals. *In: IEEE INTERNATIONAL CONFERENCE ON INTERNET OF THINGS*, 2017, Exeter, UK. **Proceedings** [...]. USA: Institute of Electrical and Electronics Engineers – IEEE, 2017. v. 10, p. 676-682.

VYGOTSKY, Lev Semyonovich. **Mind in society: the development of higher psychological processes**. Cambridge, MA: Harvard University Press, 1978.

WALLACE, Richard S. **ALICE: Artificial Linguistic Internet Computer Entity**. ALICE A. I. Foundation, 1995.

WALLACE, Richard S. **The elements of AIML style**. ALICE A. I. Foundation, 2003.

WECHSUNG, Ina; WEISS, Benjamin; KÜHNEL, Christine; EHRENBRINK, Patrick; MÖLLER, Sebastian. Development and validation of the conversational agents scale (CAS). *In: INTERSPEECH 2013 – ANNUAL CONFERENCE OF THE INTERNATIONAL SPEECH COMMUNICATION ASSOCIATION*, 14., 2013, Lyon, France. **Proceedings** [...]. Baixas, France: ISCA, 2013. p. 1106-1110.





# 4

## Enriquecendo o ambiente com recursos multimídia

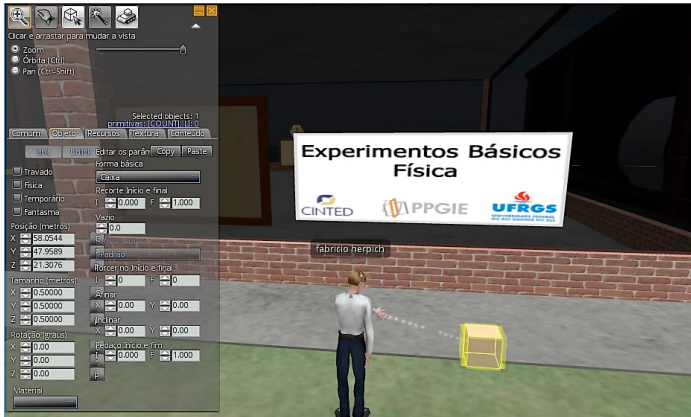
Liane Margarida Rockenbach Tarouco  
PPGIE - UFRGS - liane@penta.ufrgs.br

## 4.1 CONSTRUINDO RECURSOS MULTIMÍDIA NO AMBIENTE DO MUNDO VIRTUAL IMERSIVO

Um ambiente de Mundo Virtual Imersivo (MVI) como o *OpenSim* oferece inúmeras possibilidades para adicionar recursos multimídia, tais como imagens estáticas e animadas, vídeos e áudio. Os recursos são usualmente associados a algum objeto no MVI. Para inclusão de um recurso multimídia, o processo deve iniciar com a adição de um objeto tal como ilustrado na Figura 4.1. Clicando com o botão direito do *mouse* em algum ponto do mundo, é aberta a janela para especificar o objeto que está sendo criado. Existem diversos tipos de objetos que fazem parte do elenco de funcionalidades disponíveis: objetos geométricos, plantas etc... Outros objetos podem ser importados de repositórios em formatos diversificados, tais como imagens (em formato jpg, png, entre outros), imagens 3D (criadas com editores 3D externos, convertidas ao formato .dae para importação para o Mundo Virtual), animações (em formato .bhv e .anim) ou ainda objetos em formato .xml.

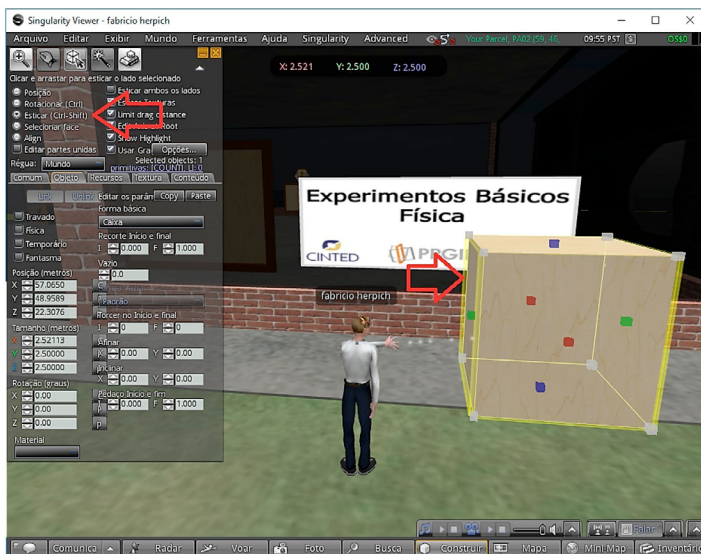
A Figura 4.1 ilustra o primeiro passo da criação de um objeto básico (*prim*). Após escolher o tipo do objeto e clicar em algum ponto do terreno no Mundo Virtual, o objeto será exibido e será possível fazer algumas configurações que irão afetar sua localização e aparência. O painel que é aberto após o processo de criação permite definir as coordenadas para a posição do objeto, bem como configurar algumas características.

Figura 4.1 - Adicionando um novo objeto ao MVI.



Para adicionar uma imagem, animação ou vídeo, deve-se ajustar as dimensões do objeto, posicioná-lo no local desejado e, logo depois, usar a opção de selecionar uma face (*Select Face*), para indicar onde a imagem será afixada, passando a substituir a textura original do objeto. A Figura 4.2 ilustra este passo.

Figura 4.2 - Alterando as dimensões e selecionando uma face do objeto.



A imagem que se deseja “colar” a uma face do objeto deve ser previamente importada, usando a opção no *Viewer* de *File -> Upload Image* ou *Import Object*. A Figura 4.3 mostra o resultado obtido com a importação de um objeto primitivo inerente ao ambiente (a árvore), o *upload* de um objeto em formato .xml (o banco) e a criação de um objeto a partir de um objeto primitivo inerente ao ambiente (um cubo que foi redimensionado). Tal procedimento pode ser utilizado para adicionar uma imagem relacionada a área da Física, por exemplo, associada a uma de suas faces, sendo que as etapas necessárias para alcançar tal resultado podem ser visualizadas na Figura 4.4.

Figura 4.3 - Resultado da importação de imagens para o Mundo Virtual.

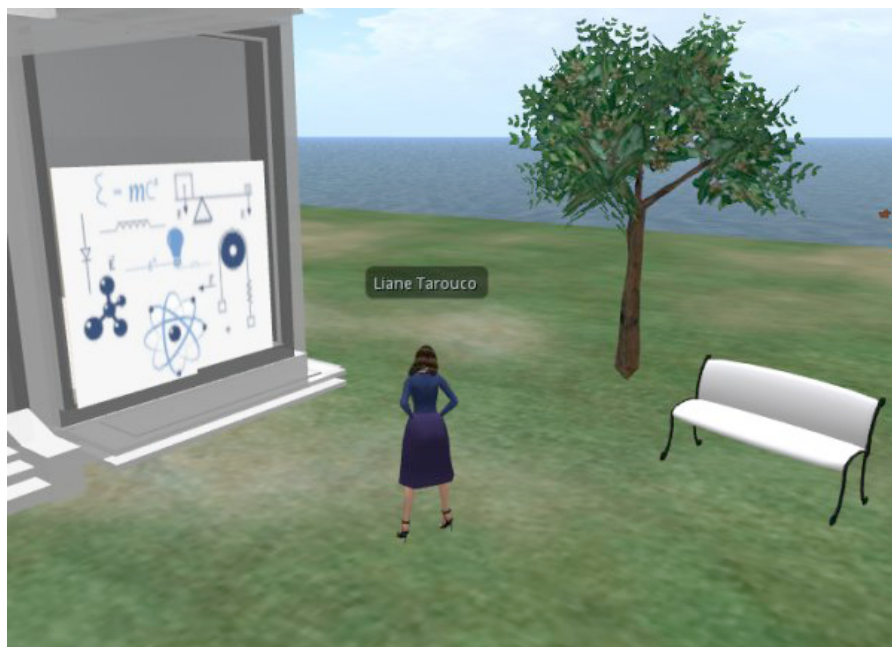
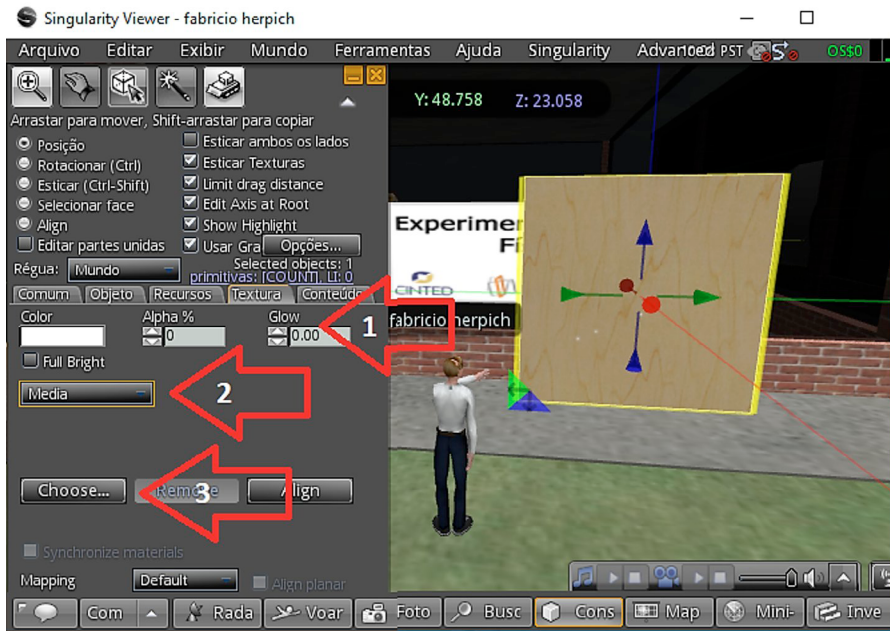
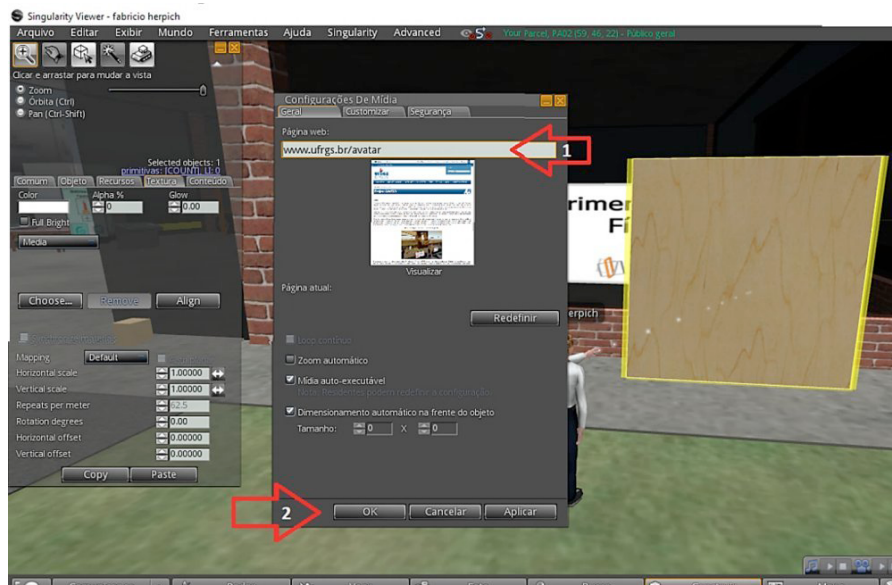


Figura 4.4 - Etapas para associar uma imagem ou mídia na face do prim.



Seguindo os mesmos passos demonstrados anteriormente, a Figura 4.5 demonstra o processo para associar uma mídia *on-line*, que pode ser incorporada ao objeto por meio de um endereço *web*. Nas configurações do objeto, mais especificamente na aba *textura* (passos 1, 2 e 3 da Figura 4.4), existe a opção *textura* do objeto, em que o usuário pode selecionar a opção *mídia* e clicar no botão “*Choose*” para abrir a tela de configuração demonstrada na Figura 4.5.

Figura 4.5 - Adicionando um endereço web.



## 4.2 ENRIQUECENDO O AMBIENTE COM RECURSOS MULTIMÍDIA

O desenvolvimento de material educacional digital de qualidade não prescinde do uso de multimídia, pois sua utilização tem sido amplamente reconhecida pelo potencial aprimoramento na forma como as pessoas aprendem (MAYER, 2005). Numa sociedade com as possibilidades tecnológicas da atual, a mediação textual da aprendizagem e da construção do conhecimento não pode limitar-se apenas ao texto como livro, já que a tecnologia evoluiu e admite maior variedade de possibilidades, tal como será descrito a seguir.

## 4.3 INCORPORANDO E USANDO MULTIMÍDIA NO MUNDO VIRTUAL IMERSIVO

Com base na constatação de que a multimídia é relevante como recurso educacional, muita investigação foi realizada para determinar de que formas ela poderia ser incorporada aos recursos educacionais digitais. Um trabalho interessante foi um estudo de Myller, Bednarik, Sutinen e Ben-Ari (2009) em que eles usaram o embasamento de um trabalho efetuado por NAPS (2005), buscando identificar como ferramentas para a visualização de algoritmos poderiam contribuir para o processo de aprendizagem colaborativo. Sua investigação resultou em uma tabela de engajamento.

Com base nos resultados alcançados por Myller, Bednarik, Sutinen e Ben-Ari (2009), Ávila (2016) propôs uma nova adaptação deste estudo, com vista a analisar as atividades educacionais ensejadas nos Mundos Virtuais (Tabela 4.1).

Tabela 4.1 - Taxonomia do Engajamento para os Mundos Virtuais.

Nível	Elemento	Descrição
T1	Visualização	Visualização sem interação
T2	Controle sobre a visualização	O estudante pode controlar a visualização
T3	Entrada de dados e alteração programada	O estudante realiza alterações de parâmetros na visualização
T4	Questionamentos	A visualização é acompanhada de perguntas sobre o seu conteúdo
T5	Modificações	Alterações não programadas podem ser realizadas na visualização oferecida
T6	Construção	A visualização é criada interativamente pelo estudante
T7	Apresentação e revisão	A visualização é apresentada para obtenção de <i>feedback</i> e discussão

Fonte: Ávila (2016).

O nível mais baixo da Taxonomia oferecido pelos Mundos Virtuais encontra-se na categoria **T1 (Visualização)**. Embora a categoria T1 não tenha previsão de qualquer forma de interação, é um elemento muito comum nos ambientes imersivos que propicia enriquecer visualmente o contexto com paisagens, cenários e objetos, para criar ao usuário a ilusão de que se encontra em um ambiente contextualizado e realístico. Esse nível pode ser atingido a partir de objetos adicionados ao MVI, mas bloqueados para a interferência de terceiros, tornando-se possível apenas a sua visualização passiva. Nesse nível de engajamento, o professor pode se valer de imagens, animações ou de apresentações



que não podem ser controladas pelo estudante, para a apresentação de um determinado conteúdo. Em um laboratório virtual, o professor poderia disponibilizar, por exemplo, apresentações iniciais para cada sala, ou experimento, às quais o estudante deveria assistir antes de realizar a atividade prática. Assim, mesmo uma visita a um MVI que apenas mostrasse um cenário com laboratório, mesmo sem a possibilidade de realizar experimentos, possibilitaria aos estudantes uma atividade preparatória para uma futura atividade em laboratório real, fazendo com que se sentissem mais confortáveis a partir desse prévio contato com a reprodução virtual do ambiente no qual, posteriormente, eles viessem a realizar seus experimentos.

O Mundo Virtual Imersivo permite associar a objetos 3D tanto imagens estáticas como animações, vídeos ou páginas *web* externas. Para tanto, basta criar um objeto 3D (*prim* ou objeto externo importado, tal como um quadro ou um monitor de TV) e aplicar sobre sua face frontal a imagem ou outro recurso multimídia. A Figura 4.6 ilustra o uso de diferentes tipos de multimídia.

Figura 4.6 - Multimídia no Mundo Virtual.



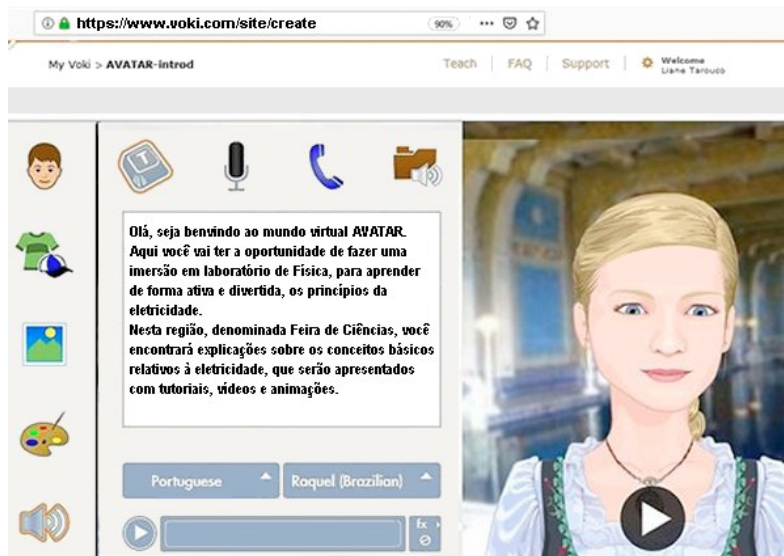
Ao fundo, em uma tela branca, é exibida uma página *web* que contém uma imagem animada de um átomo (GIF animado), usada com o propósito de explicar e ilustrar conceitos relacionados com a atividade a ser realizada no laboratório. Essa é a forma mais apropriada para a exibição de imagens animadas, como GIF animado ou mesmo uma animação criada com Flash.

Ao lado, outro painel exibe uma página que contém um personagem narrando de forma audível uma explicação ou orientação. Esta geração de personagem com narrativa pode ser gerada usando, por exemplo, um *site* como o Voki<sup>1</sup>, que permite escolher um personagem (dentro um conjunto limitado na opção de uso gratuito). A Figura 4.7 ilustra a criação de um personagem falante no Voki. Nesse ambiente, informa-se o texto a ser verbalizado pelo personagem, a língua e o acento (português europeu ou brasileiro, voz masculina ou feminina). Na versão grátis, a narrativa produzida é limitada a 60 segundos. A Figura 4.6 mostra a fer-

1 Disponível em: <<https://www.voki.com/>>.

ramenta de autoria, para criar um personagem animado que apresenta a narração gravada ou sintetizada a partir do texto. O personagem movimentava a cabeça quando fala e seus olhos seguem a direção do *mouse*.

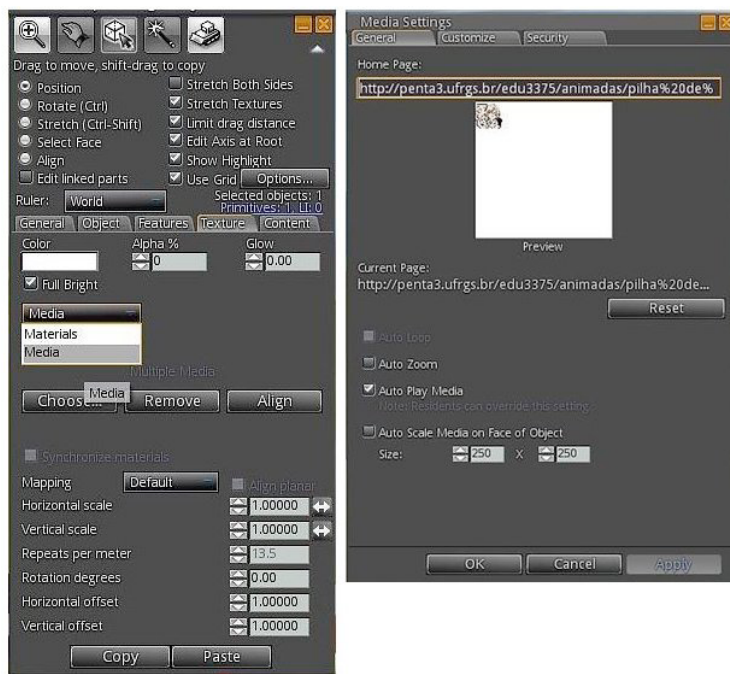
Figura 4.7 - Criando personagem falante no Voki.



Uma vez criado o personagem e sua fala, o endereço (URL) pode ser associado a uma textura a ser aplicada na face do *prim* em que vai ser exibida.

A aplicação de uma página *web* como textura em uma face de um *prim* ou em outro objeto 3D (importado) é feita usando um recurso presente na opção de criação de *prims* no ambiente *OpenSim*. Em vez de selecionar a opção *Materials*, na escolha da Textura, deve-se selecionar *Mídia (Media)*, tal como ilustrado na Figura 4.8. Depois, abre-se uma janela para especificar o URL da página a ser associada com aquela face do *prim* e com o seu tamanho, tal como ilustrado na figura da direita.

Figura 4.8 - Configurando a página web a ser associada com uma face do *prim*.



Vídeos também podem ser exibidos no Mundo Virtual e pode-se usar duas estratégias: associar o vídeo com uma textura de um *prim* ou exibi-lo em uma página *web* externa ao ambiente. Quando é necessário acessar uma página externa, é possível também usar dispositivos móveis (celular) para exibir a página contendo a multimídia. Assim, o dispositivo móvel funcionaria como complemento ao Mundo Virtual. Em tais casos, pode ser informado o URL ou pode-se exibir um QR Code no Mundo Virtual, para ser lido com um programa de leitura de QR Code no celular do usuário. A partir disso, pode-se exibir a página apontada pelo QR Code. A Figura 4.9 mostra uma situação na qual o avatar do usuário está à frente de um painel contendo um QR Code.

Figura 4.9 - Cenário do laboratório com QR Code.

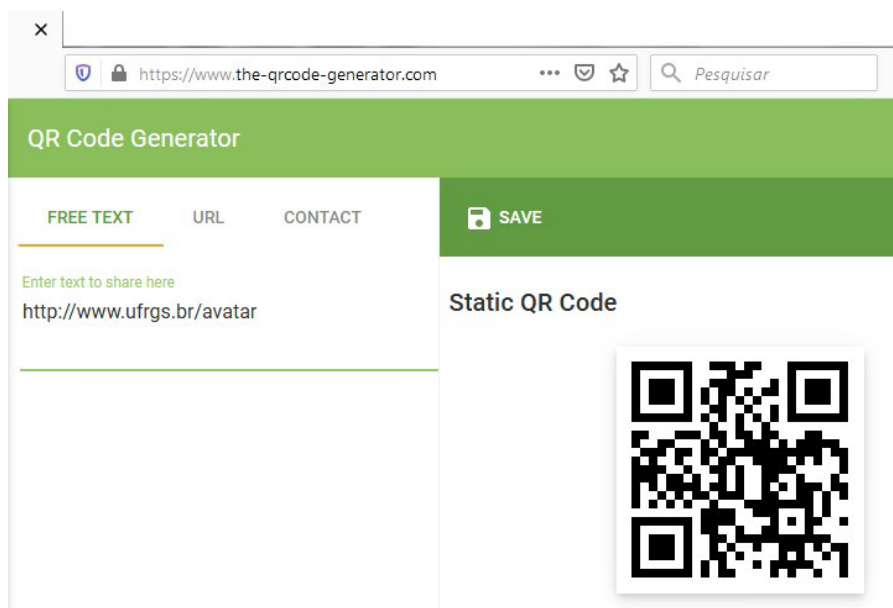


A inclusão de um QR Code no Mundo Virtual requer, em primeiro lugar, a inclusão de um objeto (*prim*) sobre o qual vai ser aplicada uma textura (imagem) contendo o QR Code. Assim, pode-se usar o mecanismo de inserção de um *prim* e escolher uma das formas disponíveis no ambiente (como descrito no Capítulo 2), tal como um cubo, ajustar suas dimensões e posição de modo a que se assemelhe a um painel na parede ou em um suporte. Depois, é preciso gerar o QR Code e, para isso, pode-se usar algum dentre os inúmeros programas *on-line* que geram imagem com QR Code correspondente a endereços *web*. Um deles, por exemplo, é o sistema *on-line* QR Code Generator<sup>2</sup>. Para obter o QR Code do ende-

2 <<https://www.the-qrcode-generator.com/>>

reço do Projeto AVATAR, por exemplo, deve ser informado o endereço (URL) do *site* do projeto e a imagem com o QR Code é imediatamente exibida, tal como ilustrado na Figura 4.10.

Figura 4.10 - Gerando QR Code.



A imagem com o QR Code deve então ser copiada e colada como uma textura no painel, para ser lida pelos estudantes que desejarem acessar aquele endereço com seus dispositivos móveis. A Figura 4.8 mostra um exemplo de painel com QR Code.

A exibição de vídeos em dispositivos móveis oferece melhor qualidade de imagem e de áudio quando comparada com a exibição no Mundo Virtual e, no Projeto AVATAR, preferiu-se esta forma complementar para exibição de recursos multimídia sem interatividade.

A categoria **Controle sobre a Visualização (T2)** é facilmente alcançada nos mundos virtuais, oferecendo diversas possibilidades para a visualização e interação com objetos. Nessa categoria, estudantes podem manipular objetos de modo a observá-los a partir de diversos ângulos e posições no espaço. Operações de *zoom* podem permitir uma inspeção de detalhes de um objeto 3D e podem, inclusive, incluir uma visão interna deles, mostrando detalhes de sua estrutura (se o objeto tiver sido construído usando composição de objetos primitivos encapsulados). No exemplo ilustrado na Figura 4.11, a seguir, o usuário pode comandar a movimentação do avatar que o representa e aproximar-se dos objetos no ambiente, ou observá-los por diferentes ângulos.

Figura 4.11 - Exemplo de controle sobre a visualização.



Quando um objeto composto, como a construção ilustrada na Figura 4.12 a seguir, é incluído no Mundo Virtual, é possível configurar suas propriedades de transparência (usando valores diferentes de zero para o parâmetro Alfa na janela de editar, tal como demonstrado na figura 4.13) de modo que se possa “ver” através das paredes. Assim, o usuário pode ingressar na casa e, uma vez dentro dela, visualizar os objetos ali colocados. As paredes da casa, configuradas com um parâmetro Alfa de 50%, conferem uma visibilidade através das paredes, tanto de



fora para dentro como na direção oposta. Isso facilita a autolocalização do usuário. Mas, apesar dessa transparência, o usuário não pode atravessar paredes e precisa usar movimentar seu avatar através da porta para poder ingressar na casa.

Figura 4.12 - Objeto composto (casa contendo mobiliário em seu interior).

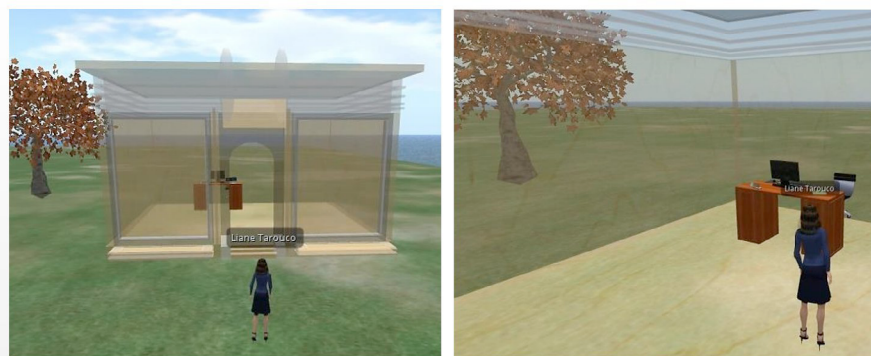
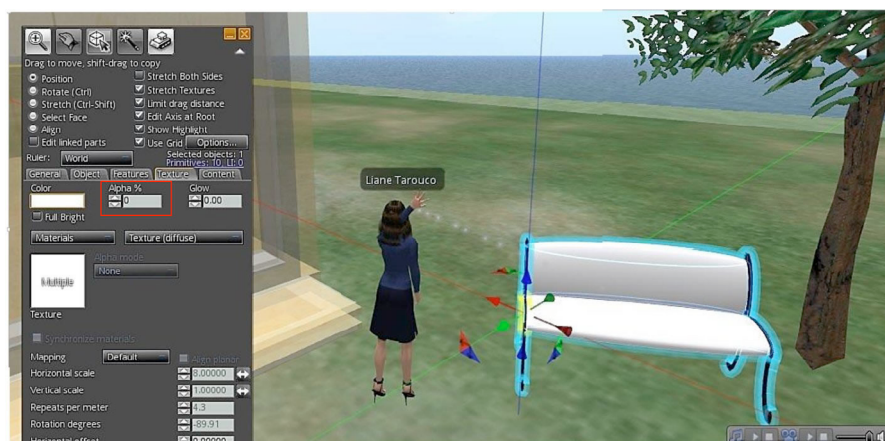


Figura 4.13 - Editando a transparência de um objeto.





No que concerne à categoria **Entrada de Dados e Alteração Programada (T3)**, trata-se da possibilidade de se realizar a alteração de parâmetros que regem o comportamento e a aparência do objeto de estudo. O estudante pode, por exemplo, alterar valores, como a temperatura ou a velocidade, em uma determinada simulação, com vista a observar seus diferentes resultados. Em decorrência, a aparência do objeto pode variar para representar as novas condições. Estudantes podem fazer, por exemplo, alterações de valores nos campos de massa e velocidade de carros em um experimento, no intuito de desenvolver conceitos relativos à colisão de corpos. Ainda, podem-se ter objetos programados para responderem de acordo com as ações realizadas pelo estudante, tais como movimentar-se ou aumentar/diminuir suas dimensões a partir de um clique do usuário.

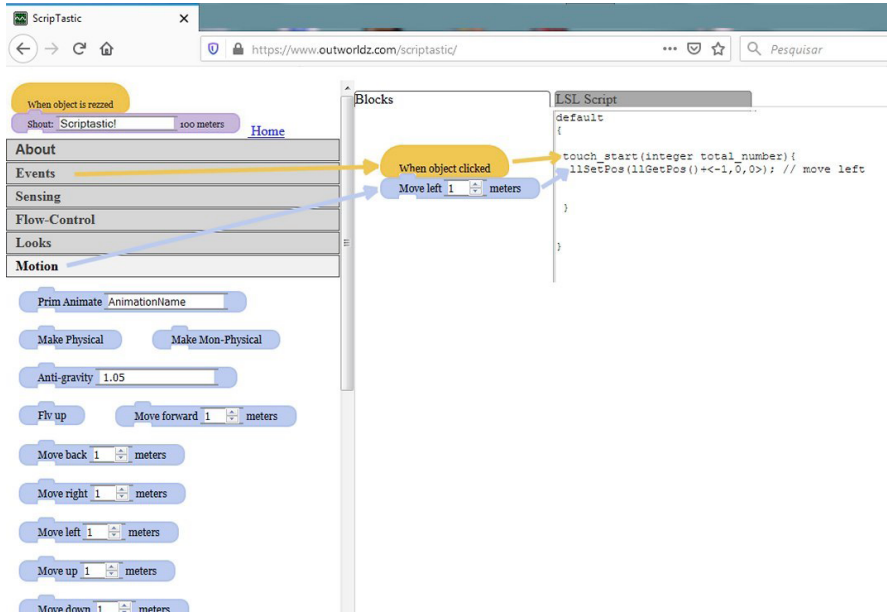
Quanto à categoria **Questionamentos (T4)**, o Mundo Virtual Imer-sivo pode instigar o estudante à reflexão a partir de questionamentos apresentados ao longo da interação com os objetos no Mundo Virtual Imer-sivo. O MVI pode também dispor de uma série de agentes (*chatbots*) programados (NPC - *Non-Player Character*) para interagir com o estudante, oferecendo-lhe dicas, discutindo aspectos do conteúdo abordado, conforme discutido no Capítulo 3. Objetos também podem ser programados para interagir com o estudante a partir de diferentes eventos, como o toque, comandos na janela de *chat*, interação com um terceiro objeto e assim por diante. Numa perspectiva mais simples, o professor, ainda, poderia se valer de exercícios interativos externos ao Mundo Virtual, como os testes apresentados em painéis no próprio Mundo Virtual ou em dispositivos móveis. Esses recursos de interati-vidade externos podem estar na *web*, e o respectivo URL pode ser in-

dicado no Mundo Virtual Imersivo diretamente ou por meio de um QR Code. O estudante então aponta o seu *smartphone* para o QR Code no Mundo Virtual, e a página indicada é carregada no seu dispositivo. A partir daí, ele pode trabalhar de forma independente do Mundo Virtual.

No que se refere à categoria **Modificações (T5)**, o estudante dispõe da possibilidade de realizar determinadas modificações sobre os objetos do mundo, alterando aspectos como o seu tamanho ou a sua forma. Nesse nível, os estudantes podem ser convidados a realizar alterações sobre objetos disponibilizados pelo docente no Mundo Virtual. Para que tal circunstância seja viabilizada, basta que o docente compartilhe com os alunos os poderes de edição sobre seus objetos. O estudante pode importar novos objetos para o Mundo Virtual Imersivo e associar-lhes comportamentos. Essa associação implica o uso de linguagem de programação de *scripts* a ser incorporada aos objetos. Esses *scripts* podem ser desenvolvidos com o apoio de geradores de código, baseados em linguagem de programação de blocos visuais, tais como *ScripTastic* e FS2LSL (descritas no Capítulo 7).

A Figura 4.14 mostra um exemplo de uso do *ScripTastic* para gerar o código em LSL necessário para mover um objeto no Mundo Virtual quando ele for tocado (clicado).

Figura 4.14 - Gerando código com o *ScriptTastic*.



A categoria **Construção (T6)** prevê a possibilidade de construção e edição de objetos diretamente pelo estudante por meio de ferramentas de construção do Mundo Virtual. Ferramentas para a construção de *prims* (Componentes Primários do Mundo Virtual), disponíveis nos visualizadores, podem ser exploradas no intuito de se desenvolver o próprio conteúdo dentro do Mundo Virtual. Os *prims* podem ser construídos e alterados de modo a constituírem diferentes objetos a partir da sua integração e manipulação. Essa categoria comporta também a adição de interatividade aos objetos do Mundo Virtual. Nesse nível de engajamento, o estudante é responsável por alterações significativas sobre o seu próprio objeto de estudo. Atividades podem demandar a importação e edição de objetos externos ao Mundo Virtual, a construção de objetos pela integração de diferentes *prims*, ou ainda a programação

de ações a serem executadas no Mundo Virtual. Existem diversos repositórios que oferecem uma grande quantidade de objetos que podem ser importados ao Mundo Virtual e comportamentos regidos por *scripts* em LSL e OSSL podem ser a eles associados.

Quanto à **Apresentação e revisão (T7)**, os mundos virtuais, por oferecerem um ambiente compartilhado, com diferentes canais de comunicação, propiciam aos estudantes a oportunidade de compartilhar suas realizações com o grupo, no qual suas produções podem ser analisadas e discutidas juntamente com professores e colegas, gerando uma série de dicas e realimentações para o aprimoramento da atividade. Nessa perspectiva, enquadra-se a proposta de uso dos mundos virtuais para a realização de sessões de pôsteres, apresentada por Okutsu *et al.* (2013). A promoção desse espaço para a discussão e reflexão permite pôr em evidência não somente os aspectos positivos alcançados durante o percurso da aprendizagem, mas também lacunas de compreensão que tenham ficado pendentes. Como sugerem McGowen e Tall (2013), o estudante deve ser conduzido a debates e reflexões sobre sua aprendizagem, a fim de que ele próprio se torne capaz de perceber eventuais fragilidades conceituais que possua.

A elaboração de situações de aprendizagem significativas no Mundo Virtual demanda o desenvolvimento de conhecimentos técnicos que permitam ao docente viabilizar a construção de atividades diversas. É importante que o docente disponha de um domínio básico sobre construção, edição e importação de conteúdo multimídia para o MVI. O docente deve ter domínio sobre ferramentas de construção do Mundo Virtual e bloqueio e desbloqueio para a interferência de terceiros em

seus objetos. Se o docente tiver conhecimento de programação, pode desenvolver atividades com maior riqueza em termos de interatividade. Em caso contrário, o conhecimento básico sobre o campo conceitual em foco vai ajudar a especificar melhor estratégias de interação dos avatares com objetos no Mundo Virtual Imersivo.

#### **4.4 MULTIMÍDIA INTERATIVA COMO RECURSO DE COGNIÇÃO E APRENDIZAGEM NO MUNDO VIRTUAL IMERSIVO**

Em sua investigação sobre o uso de multimídia como recurso de cognição e aprendizagem, Moreno e Mayer (2007) propõem alguns tipos de interatividade em ambientes educacionais multimídia: dialogar, controlar, manipular, pesquisar e navegar. A Tabela 4.2 caracteriza as cinco formas de interatividade consideradas.

Tabela 4.2 - Tipos de interatividade em ambientes educacionais multimídia.

Tipo de interatividade	Descrição
Dialogar	O estudante recebe questões e respostas ou realimentação para suas reações
Controlar	O estudante determina o ritmo e/ou ordem da apresentação
Manipular	O estudante estabelece parâmetros para uma simulação, define o foco e a aproximação (zoom) ou move objetos no cenário
Pesquisar	O estudante encontra novo material, fazendo uma pergunta, recebendo opções e selecionando as alternativas desejadas
Navegar	O estudante move-se por diferentes áreas do conteúdo, selecionando diversas fontes de informação disponíveis

Fonte: Moreno e Mayer (2007).

A possibilidade de interatividade constitui uma estratégia para promover uma aprendizagem significativa, envolvendo o estudante em processamento ativo do material educacional. A aprendizagem significativa ocorre quando o estudante dedica esforço consciente ao processo de cognição, por meio de atividades tais como selecionar, organizar, integrar nova informação no conhecimento existente. Fatores motivacionais afetam a aprendizagem, aumentando ou reduzindo o engajamento cognitivo. Da mesma forma, diferenças no conhecimento prévio e habilidades do estudante afetam o quanto é aprendido no manuseio de determinada mídia. Esses princípios podem ser usados no projeto de um laboratório usando Mundo Virtual Imersivo.

No que tange aos aspectos de interatividade, Mayer (2005) sugere cinco princípios empíricos para o projeto de ambientes de aprendizagem multimídia interativos:

a) **Atividade guiada** – ensina a interação dos estudantes com um agente pedagógico que guia (provendo orientações) o processo de cognição durante a aprendizagem, ensejando o engajamento dos estudantes na seleção, organização e integração de nova informação.

b) **Reflexão** – estudantes aprendem melhor quando são solicitados a refletir sobre as respostas e sua correção durante o processo de “fazer sentido”. A reflexão pode ser estimulada a partir de perguntas que instiguem o pensamento crítico.

c) **Realimentação** – deve ser explanatória e não corretiva (mal-entendidos são dirimidos com a explanação).

d) **Ritmo** – cada um deve poder controlar o ritmo à medida que o trabalho progride.

e) **Pré-treinamento** (ou nivelamento) – os estudantes aprendem melhor quando recebem conhecimento específico preparatório, que lhe proporciona ou ativa conhecimento relevante prévio.

No Projeto AVATAR, foi usado o recurso do agente conversacional que foi integrado à arquitetura pedagógica do ambiente, cumprindo diferentes funções.

a) Agente com função 1 é ativado quando o usuário (por meio de seu avatar) se aproxima da entrada de uma região em que serão desenvolvidas atividades experienciais. Esse agente apresenta boas-vindas e explica o que o estudante vai encontrar naquela re-

ção e o que ele deve fazer. Esse agente tem a função de motivar o estudante para o tema que será trabalhado no laboratório virtual. Ainda pode sugerir ao estudante interagir com recursos de animação, vídeos, imagens estáticas, colocados no ambiente para contextualizar o cenário no qual a atividade de laboratório vai se realizar.

b) Agente com função 2 tem a responsabilidade de chamar a atenção do estudante para os conceitos relacionados e relevantes ao experimento que vai ser realizado no laboratório virtual. Sempre que o avatar do estudante se aproxima de algum recurso multimídia incluído no ambiente, o agente pode aparecer e explicar a importância da inspeção daquele recurso. Ele buscará incentivar o estudante a inspecionar demonstrações, animações, vídeos, cartazes e *links* para *sites* que ofereçam informações complementares sobre o campo conceitual envolvido. O objetivo dessa atividade é o de assegurar que o estudante detenha os subsunçores necessários para servir de andaime a uma aprendizagem significativa, conforme proposto por Ausubel (1978) e detalhado por Moreira (2008).

c) Agente com função 3 deve orientar e incentivar uma postura ativa por parte do estudante. Ele buscará isso incentivando e orientando o estudante na realização do experimento, bem como assegurando que o estudante tenha atentado e observado os aspectos importantes e relevantes do experimento. Adicionalmente, esse agente deve procurar promover desequilíbrios e auxiliar no processo de assimilação e acomodação, segundo a teoria de Piaget (1973). Perguntas instigadoras de observação serão formuladas pelo agente em função do andamento (fase) do experimento.



d) Agente com função 4 tem a missão de promover a reflexão sobre a experiência proporcionada pelo laboratório virtual com vista a alcançar metacognição.

O Ciclo de Kolb (1984) foi usado como estratégia de organização da atividade do estudante no Projeto AVATAR. A proposta de Kolb é a de que a aprendizagem é um processo contínuo, baseado na experiência, passando por quatro fases, tal como ilustrado na Figura 4.15. Essa proposta considera que ideias não são elementos de pensamento fixos e imutáveis, mas são formados e reformados pela experiência.

Figura 4.15 - Ciclo de Kolb.



Na primeira fase, o estudante tem a oportunidade de realizar experiência com o fim de testar hipóteses. Na fase seguinte, é preciso que ocorra uma reflexão a partir da observação dos resultados dos experimentos realizados. Essa reflexão prepara o terreno para a construção da abstração inerente a um processo de conceitualização. O que foi ob-

servado é passível de replicação em qualquer outro contexto? Sob que condições? Quais regras gerais regem o comportamento do processo observado? Uma vez que estas perguntas consigam ser respondidas, o estudante está em condições de transladar o conhecimento adquirido naquela experiência para outros contextos.

Segundo esta estratégia, o uso do laboratório construído no Mundo Imersivo Virtual, com todos os recursos multimídia que o ambiente *OpenSim* possibilita, constitui um recurso educacional valioso a fim de promover uma aprendizagem rica e significativa, conforme discutido em Tarouco (2014).

## REFERÊNCIAS

AUSUBEL, David P.; NOVAK, Joseph D.; HANESIAN, Helen. **Psicologia educacional**. 2. ed. Rio de Janeiro: Interamericana, 1978. 625 p.

ÁVILA, Barbara Gorziza. **Formação docente para a autoria nos mundos virtuais: uma aproximação do professor às novas demandas tecnológicas**. 2016. 231 f. Tese (Doutorado em Informática na Educação) – Centro de Estudos Interdisciplinares em Novas Tecnologias, Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.

KOLB, David. **Experiential learning: experience as the source of learning and development**. Englewood Cliffs, NJ: Prentice Hall, 1984.

MAYER, Richard E. (ed.). **The Cambridge handbook of multimedia learning**. 2. ed. Cambridge, UK: Cambridge University Press, 2005.

MAYER, Richard E.; MORENO, Roxana. Animation as an aid to multimedia learning. **Educational psychology review**, v. 14, n. 1, Mar. 2002.

MCGOWEN, Mercedes A.; TALL, David O. Flexible thinking and met-before: impact on learning mathematics. **Journal of mathematical behavior**, v. 32, p. 527-537, 2013.

MOREIRA, Marco Antônio. A teoria da aprendizagem significativa segundo Ausubel. *In: MASINI, Elcie Aparecida Fortes Salzano; MOREIRA, Marco Antônio (org.). **Aprendizagem significativa: condições para ocorrência e lacunas que levam a comprometimentos.** São Paulo: Vetor, 2008. p. 15-44.*

MORENO, Roxana; MAYER, Richard. Interactive multimodal learning environments. **Educational psychology review**, v. 19, n. 3, p. 309-326, Sept. 2007.

MYLLER, Niko; BEDNARIK, Roman; SUTINEN, Erkki; BEN-ARI, Mordechai. Extending the engagement taxonomy: software visualization and collaborative learning. **Transactions on computing education**, v. 9, n. 1, art. 7, Mar. 2009.

NAPS, Thomas L. Jhavé: addressing the need to support algorithm visualization with tools for active engagement. **IEEE Computer graphics and applications**, v. 25, n. 5, p. 49-55, Sept./Oct. 2005.

OKUTSU, Masataka; DELAURENTIS, Daniel; BROPHY, Sean; LAMBERT, Jason. Teaching an aerospace engineering design course via virtual worlds: a comparative assessment of learning outcomes. **Computers & education**, v. 60, p. 288-298, 2013.

PIAGET, Jean. **Biologia e conhecimento: ensaio sobre as relações entre as regulações orgânicas e os processos cognoscitivos.** Petrópolis: Vozes, 1973.

TAROUCO, Liane Margarida Rockenbach *et al.* Virtual laboratory for promoting engagement and complex learning. *In: E-LEARN 2014 - WORLD CONFERENCE ON E-LEARNING, 2014, New Orleans, USA. **Proceedings** [...].* Waynesville, NC: The Association for the Advancement of Computing in Education - AACE, 2014. p. 1933-1938.



# 5

## Rastreamento em Mundos Virtuais Imersivos

**Felipe Becker Nunes - AMF**  
nunesfb@gmail.com

**Leandro Rosniak Tibola - PPGIE - UFRGS**  
lrtibola@gmail.com

## 5.1 USO E CONFIGURAÇÃO DE SENSORES EM MVI

Os professores e os alunos podem construir os seus ambientes de aprendizagem a fim de que possam vivenciar uma “viagem em busca do conhecimento” (ZIMMER; VEZZANI, 2017). Conforme Girvan e Savage (2010) e Maceu e Sousa (2018), os principais aspectos que contribuem para a imersão do usuário em ambientes 3D são a interação e as possibilidades de comunicação. Na educação, essas ferramentas são sustentáculos da aprendizagem construtivista, contribuindo, assim, para que o aluno dê significância ao conhecimento.

A proposição de laboratórios virtuais, estruturados com recursos multimídia e simulações dentro do Mundo Virtual, também carece de um importante aspecto a ser considerado. O monitoramento das atividades dos usuários, durante o período em que estão interagindo com os recursos no Mundo Virtual, caracteriza-se como um importante indicador de assiduidade e permite melhorar a avaliação de cada participante no ambiente.

Em ambientes virtuais mais consolidados no meio acadêmico, como o *Moodle*, esse tipo de monitoramento já é implementado automaticamente. Dessa forma, informações pertinentes sobre as interações dos alunos nos cursos são armazenadas, como o momento em que acessou, com quais materiais interagiu e que atividades realizou. Esse tipo de informação pode fornecer ao professor indícios relativos sobre a dedicação do usuário em sua disciplina e auxilia no momento da avaliação individual dos alunos.

O'Brien e Toms (2008) consideram que a dedicação do usuário depende da profundidade de participação que os utilizadores são capazes de alcançar em relação a cada atributo experimental disponível no sistema. Ao desempenhar tal análise no contexto dos Mundos Virtuais, torna-se necessário efetuar a coleta de dados resultantes das interações do usuário no ambiente, como a assiduidade de acesso, locais que visitou e quanto tempo permaneceu, bem como quais ações executou ao interagir com os materiais didáticos e demais usuários no ambiente (HERPICH *et al.*, 2016). Originalmente, o Mundo Virtual não fornece dados das interações realizadas pelos usuários no ambiente, assim como o docente não consegue monitorar o tempo inteiro e em todos lugares as interações realizadas pelos usuários. O Mundo Virtual contém diversas limitações e dificuldades no processo de coletar dados resultantes das interações dos usuários, como o momento em que acessou e saiu do ambiente, quais locais visitou e que atividades desempenhou (NUNES *et al.*, 2016). Dessa forma, a criação de um processo automatizado de monitoramento nesse ambiente se configura como essencial, já que os dados coletados poderão fornecer ao professor uma visão mais detalhada, seja de cada indivíduo seja do grupo em geral.

A identificação dessa necessidade resultou na publicação de um “recorte” desta pesquisa (NUNES *et al.*, 2016), no qual é ressaltado que esse tipo de ambiente não foi desenvolvido, exclusivamente, para o uso educacional, como pode ser constatado em outros tipos de *softwares*, como o *Moodle*, que é um ambiente desenvolvido com foco no âmbito educacional, preparado para realizar o monitoramento e avaliação das atividades educacionais realizadas pelos estudantes. Dessa forma, no

Mundo Virtual, existem limitações e dificuldades no processo de coletar dados resultantes das interações dos usuários no ambiente, como o momento em que acessou e saiu, quais locais visitou e que atividades desempenhou, assim como a realização de avaliações. Tais exemplos de informações coletadas podem ser consideradas essenciais para que o professor possa realizar uma análise consistente de uma intervenção didática ocorrida neste tipo de ambiente (NUNES *et al.*, 2016).

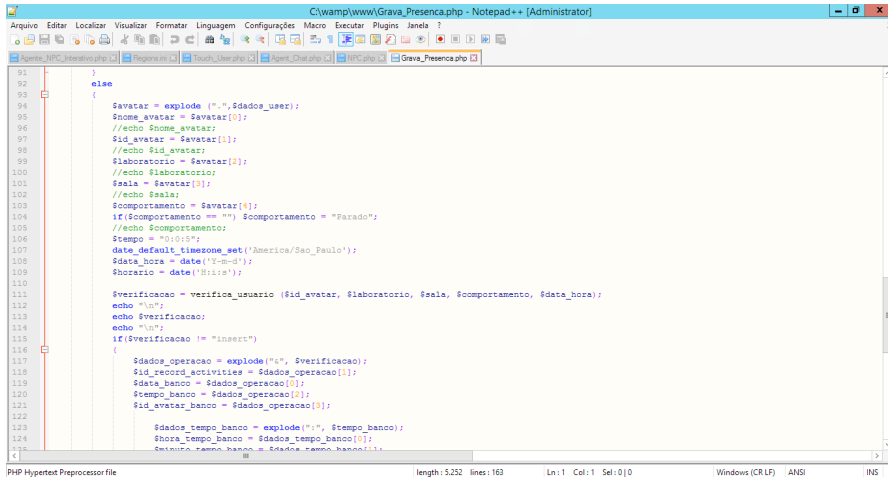
Os sensores no ambiente têm como configuração-padrão a identificação, por um intervalo de tempo previamente definido, da presença de algum usuário em seu raio de cobertura, ou seja, é definida uma área de abrangência em que o sensor irá ficar monitorando em um intervalo de segundos, previamente definido, se alguém entrou neste espaço. Caso alguém tenha entrado, diversas ações podem ser executadas, como a identificação do usuário, realização de alguma ação por parte de um objeto ou agente virtual, etc.

Pode-se citar como exemplo o sistema de monitoramento criado no trabalho de Nunes (2017), em que os sensores foram criados no Mundo Virtual utilizando a linguagem OSSL para efetuar a programação necessária, em que foram adicionadas pequenas esferas na cor vermelha, em diferentes locais do ambiente, cada uma com sua função específica, esquemmatizando, desta forma, o processo de monitoramento no Mundo Virtual. Os *scripts* OSSL foram interligados a arquivos externos, desenvolvidos utilizando a linguagem de programação PHP e hospedados no servidor *web* (Figura 5.1), sendo que a comunicação entre eles era realizada por meio dos comandos: HTTP Request, que envia os dados capturados no ambiente pelos *scripts* OSSL para os respectivos *scripts*



PHP no servidor *web*; e HTTP Response, por meio do qual, após ter sido processada a respectiva ação definida para estes dados no *script* PHP, uma resposta é gerada e enviada ao *script* OSSL correspondente.

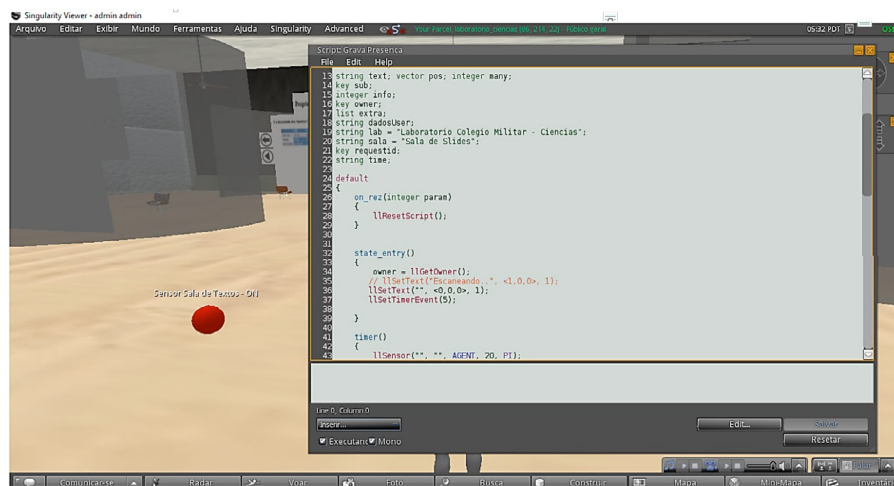
Figura 5.1 - Exemplo de código em PHP para envio e recebimento de dados no servidor.



```
91 }
92 else
93 {
94     $avatar = explode("-", $dados_user);
95     $nome_avatar = $avatar[0];
96     //echo $nome_avatar;
97     $id_avatar = $avatar[1];
98     //echo $id_avatar;
99     $laboratorio = $avatar[2];
100    //echo $laboratorio;
101    $sala = $avatar[3];
102    //echo $sala;
103    $comportamento = $avatar[4];
104    if($comportamento == "") $comportamento = "Parado";
105    //echo $comportamento;
106    $tempo = "00:00";
107    date_default_timezone_set('America/Sao_Paulo');
108    $data_hora = date('Y-m-d');
109    $horario = date('H:i:s');
110
111    $verificacao = verifica_usuario($id_avatar, $laboratorio, $sala, $comportamento, $data_hora);
112    echo "\n";
113    echo $verificacao;
114    echo "\n";
115    if($verificacao != "insert")
116    {
117        $dados_operacao = explode(";", $verificacao);
118        $id_record_activities = $dados_operacao[1];
119        $data_banco = $dados_operacao[0];
120        $tempo_banco = $dados_operacao[2];
121        $id_avatar_banco = $dados_operacao[3];
122
123        $dados_tempo_banco = explode(":", $tempo_banco);
124        $hora_tempo_banco = $dados_tempo_banco[0];
125        $minuto_tempo_banco = $dados_tempo_banco[1];
126    }
127 }
```

Portanto, o modo de funcionamento ocorre desta maneira: os *scripts* no *OpenSim*, que estão alocados em diversas esferas vermelhas dispostas no laboratório, captam os dados especificados anteriormente e enviam-nos para os arquivos PHP, que realizam o seu tratamento e armazenamento no banco de dados do *OpenSim*. A Figura 5.2 apresenta um exemplo de código no MVI que envia e recebe dados dos *scripts* construídos em PHP e hospedados no servidor.

Figura 5.2 - Exemplo de código no MVI para envio e recebimento de dados.



Tais dados podem ser atualizados conforme as informações vão sendo coletadas dentro do Mundo Virtual, fornecendo assim um monitoramento em tempo real. Como exemplo de tipos de sensores para a captura de dados das interações dos usuários no ambiente, é possível citar:

a) **Controle de Assiduidade:** pela implementação de um sensor para averiguar se o usuário acessou ou não o ambiente para realizar uma determinada tarefa ou visualizar os materiais didáticos disponibilizados, permite estabelecer a assiduidade de cada encontro virtual previamente programado.

b) **Controle de Ação:** pela implementação de um sensor para armazenar dados referentes às interações dos alunos dentro do ambiente, pode-se captar, por exemplo, se o usuário está escrevendo, ausente, voando, andando, parado, correndo, entre outros estados do avatar.

c) **Controle de Conversação:** pela implementação de um sensor para armazenar todas as conversas do usuário no ambiente, separado pela identificação do avatar, texto da mensagem, data e hora da escrita, pode-se auxiliar a identificação da interação textual dos participantes, assim como até exportar estes dados para serem analisados com técnicas de mineração de texto.

d) **Controle de Interação:** pela implementação de um sensor para armazenar dados referentes às interações dos alunos com os objetos 3D, também, pode-se ter uma boa alternativa a ser adotada, pela qual se torna possível identificar em quais objetos os usuários tocaram, para interagir com eles, e quantas vezes isso ocorreu em cada vez que ele acessou o ambiente.

Os autores ressaltam que tais informações podem ser consideradas de grande valia ao professor, visto que ele tem a sua disposição um conjunto de dados que podem ser manipulados e analisados para se obter maior controle do desempenho de cada aluno durante o decorrer do semestre, assim como identificar possíveis estudantes que não tenham realizado as atividades e/ou obtido um desempenho considerado insatisfatório na disciplina, para que possam ser aplicadas ações corretivas em tempo hábil (NUNES *et al.*, 2016, p. 9).

Assim, o professor cria condições de saber qual objeto específico foi tocado e qual conteúdo estava sendo abordado nele, além das simulações, em que é possível identificar qual experimento foi tocado e quantas vezes, buscando aprimorar o nível de detalhamento das trajetórias conduzidas pelos alunos e sua avaliação. Carvalho Júnior (2005) explica que o método de mensuração do percurso pedagógico percorrido pelo aluno, durante uma intervenção didática, para estimar a tra-

jetória de aprendizagem desenvolvida em ambientes como os Mundos Virtuais, é uma prática encarada como a rota de aprendizagem que o aluno trilha com o objetivo de atingir o conhecimento necessário acerca de um ou vários conceitos interligados por meio de um tópico geral.

Saint-Georges e Fillietaz (2008) definem esse processo com a expressão “Trajetória de Aprendizagem Situacional”, para se referirem à trajetória efetivamente realizada, sendo que o interesse em monitorá-la é com vista a um planejamento em longo prazo. Os autores explicam que o uso desse termo está ligado a duas proposições, as quais envolvem a perspectiva situacional, que diz respeito ao foco nas ações que ocorrem em tempo real, realizadas pelos indivíduos, e envolve a ideia de trajetória, que leva em consideração ações em longo prazo por parte do professor (SAINT-GEORGES; FILLIETTAZ, 2008).

O conhecimento que o professor detém de cada indivíduo que compõe o grupo de alunos sob sua orientação poderá fornecer o embasamento necessário ao entendimento do caminho genérico a ser proposto pelo docente, assim como os motivos pelos quais determinado trajeto foi percorrido pelos estudantes durante a realização de um conjunto de atividades didáticas. Simon (1995) pontua que o docente está constantemente percebendo a extensão das modificações e transformações que podem ser construídas por algum ou todos os componentes de uma trajetória de aprendizagem: o método, as atividades e o processamento hipotético da aprendizagem. Essa postura agrega novos conhecimentos ao professor, que poderá, com base nas informações que já detinha, refletir sobre os acontecimentos, tanto de forma individual, quanto do

ponto de vista geral do grupo, para que possam ser executados ajustes na trajetória de aprendizagem proposta, sendo este um processo constante e repetitivo.

É importante ressaltar que as rotas de construção de conhecimento dos alunos não são padronizadas, o que acarreta na interpretação de que cada estudante percorre um caminho pessoal e distinto ao longo de uma atividade de intervenção didática. Portanto, é necessário compreender que o mapeamento das trajetórias deve ser considerado um processo contínuo de aprendizagem dos alunos durante determinado período de tempo, não o tornando um julgamento dicotômico entre certo e errado. Wilson *et al.* (2013) enfatizam que diferentes níveis de aprendizagem e particularidades são esperados, não devendo ser tratados como algo estático e equiparados a um único caminho correto a ser cumprido. Destaca-se, ainda, que a construção de um laboratório virtual e a implementação de recursos para monitorar as ações dos usuários e auxiliar no processo de ensino e aprendizagem dos estudantes não podem ser consideradas suficientes. A interação ocorrida nos Mundos Virtuais fornece um aspecto motivador para estudantes e, geralmente, atrai seu interesse e entusiasmo. No entanto, é preciso notar que esses ambientes não podem efetivamente substituir todas as outras abordagens de aprendizado existentes (ZAHARIAS *et al.*, 2010). O planejamento didático efetuado pelo professor na disciplina, tendo como base uma abordagem educacional bem definida, irá nortear os objetivos das atividades e auxiliar os alunos a desempenharem suas tarefas de forma mais adequada e consistente nesse tipo de ambiente.

## 5.2 COLETA DE DADOS SOBRE O USO DOS RECURSOS MULTIMÍDIA DO MVI

Os MVIs são Mundos Virtuais em 3D baseados em computador. Segundo Bell (2008), esses Mundos Virtuais podem ser definidos como uma rede síncrona e persistente de pessoas representadas como avatares.

Os Mundos Virtuais são formados a partir de primitivos e objetos. Os primitivos ou *prims* são as unidades básicas que possibilitam a criação de objetos complexos no MVI. Os *prims* são as formas geométricas únicas, tais como um cubo, uma esfera, uma pirâmide, um cilindro e um cone. Nos MVIs, objetos como prédios, móveis, carros e até mesmo o cabelo e a roupa de um avatar são construídos com um ou mais *prims* (OPENSIM, 2018).

Os *prims* ou os objetos podem ter finalidades simples, como compor o ambiente tridimensional, fazendo parte do cenário, ou cumprir funções complexas, comportando-se de acordo com determinada ação provocada pelo avatar ou por outro objeto. A inserção de comportamento nos *prims* ou objetos é possível com a programação de *scripts* LSL e/ou OSSL (OPENSIM, 2012).

Considerando os recursos disponíveis na programação LSL/OSSL, as necessidades de monitoramento das ações dos alunos nos MVIs e seu armazenamento para posterior análise, ampliou-se o entendimento dos comportamentos obtidos com *scripts* LSL/OSSL descritos em Tibola e Tarouco (2015). Assim, os *scripts* LSL/OSSL permitem: (I) comportamentos ativos: quando o avatar propositalmente exerce uma ação sobre o objeto, consciente de que o objeto executa um procedimento,

como, por exemplo, quando o objeto é “tocado” pelo avatar; (II) comportamentos reativos: quando o objeto possui um *script* que executa um procedimento, quando o avatar casualmente exerce uma ação sobre este objeto, mesmo que a presença de um *script* embutido no objeto não seja visível ao aluno, como, por exemplo, no evento *collision*, que é acionado quando o avatar colide com o objeto e, em consequência, um *script* é executado.

A clara percepção de que um objeto possui comportamentos ativos é a visualização de uma “mão” quando o ponteiro do *mouse* passa sobre ele, identificando a presença de um *script* com eventos do tipo *touch*. Já os comportamentos reativos não apresentam uma identificação semelhante, sendo a presença de *scripts* transparente ao aluno, assim, a percepção da presença de *scripts* acontece quando a ação entre o avatar e o objeto ocorre. Os objetos com comportamentos reativos podem conter *scripts* com eventos do tipo *collision* e *sensor*.

Enquanto este Capítulo foca especificamente nos *scripts* relacionados com a coleta de dados das ações do avatar no MVI, o Capítulo 6 apresenta informações mais detalhadas sobre as linguagens *scripts* LSL e OSSL.

### **5.2.1 Monitoramento das ações do avatar do estudante**

A investigação, a validação, a promoção e a capacitação de desenvolvedores e usuários para o uso de laboratórios virtuais em ambientes imersivos é o objetivo do Projeto AVATAR. A concepção e o desenvolvimento dos recursos presentes nos laboratórios, disponibilizados no

Projeto AVATAR, mesmo considerando o uso de modernas ferramentas de autoria, demandam um complexo arranjo de habilidades multidisciplinares. Assim, no âmbito deste projeto, são desenvolvidos objetos completos, que liberam a realização de experimentos e objetos básicos que possam ser usados na construção de novos experimentos – componentes com comportamentos que respondam a ações de interatividade do usuário (PROJETO AVATAR, 2018).

Dentro do Projeto AVATAR, algumas pesquisas enfocam as possibilidades criadas pela programação *script* LSL/OSSL para o monitoramento das ações dos avatares dos alunos nos MVIs, bem como o registro dessas ações em banco de dados para análises posteriores, tal como nas investigações pioneiras de Tibola e Tarouco (2015), nas aplicações destas descobertas, em pesquisas associadas dentro do Projeto AVATAR (PROJETO AVATAR, 2018) e, mais especificamente, envolvendo o processo completo de uso do MVI, monitoração das ações dos avatares dos estudantes, armazenamento em banco de dados e análise destes dados, como proposto por Tibola (2018).

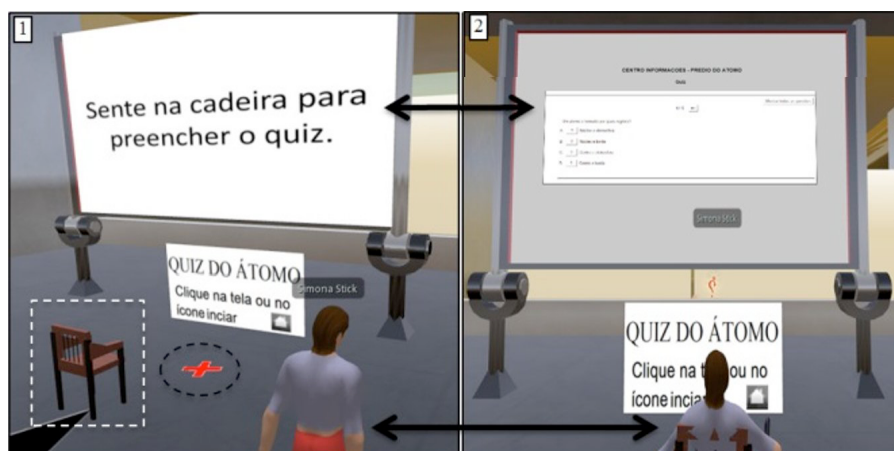
Assim, para identificar se o aluno cumpriu com o que estava previsto na estratégia educacional utilizada, se apresenta indícios de engajamento e motivação, se seguiu o trajeto pedagógico esperado ou qualquer que seja o objetivo de ensino-aprendizagem proposto, é necessário registrar todas as informações pertinentes às tarefas que o aluno realizou ou deixou de realizar e os locais por onde ele transitou no MVI. Os objetos sensores podem ser utilizados para executar estas funções.



Segundo Tibola e Tarouco (2015), os objetos sensores possuem código LSL/OSSL embutido, o que permite identificar a presença do avatar e monitorar quais foram as ações realizadas por ele durante a interação com o objeto ou se não houve ações pela falta de interação. Os sensores podem ser de dois tipos: (I) sensores geográficos e (II) sensores de interação.

Os sensores geográficos possuem um funcionamento transparente para o aluno, pois ele não percebe a atuação dos sensores, uma vez que é executado em segundo plano. Ainda, os sensores geográficos permitem verificar o posicionamento do aluno dentro do MVI, se ele entrou e/ou saiu de um prédio ou sala, se passou por determinado lugar, se andou ou voou, por quanto tempo, conforme a necessidade de registro (TIBOLA; TAROUCO, 2015). A Figura 5.3 apresenta um objeto sensor geográfico, no formato de “X”, envolvido por um círculo tracejado.

Figura 5.3 - Objetos sensores de localização e de interação.



No caso da Figura 5.3, o sensor geográfico coleta dados tais como o caminho sugerido a ser seguido pelo aluno, o local visitado, o código de identificação do avatar, o nome do avatar e se o avatar está chegando à área do sensor ou saindo. Logo a função “lIHTTPRequest” envia estas informações para um servidor WWW pelo protocolo HTTP, transferindo-as para uma página php (grava\_trajeto\_local.php), que se comunica com uma base de dados e grava as informações na respectiva tabela. A Figura 5.4 descreve a parte do *script* que realiza o envio das informações para o banco de dados.

Figura 5.4 - Envio das informações do sensor geográfico para o servidor WWW.

```
----- LOCAL HOST -----  
chamadaHTTP =  
  lIHTTPRequest("http://127.0.0.1/opensim/grava_trajeto_local.php?workflow="+workflow+"&local="+local+  
  "&idavatar="+id_avatar+"&nomeavatar="+nome_avatar+"&presenca="+presenca+"&sensorid="+sensorID,  
  [HTTP_METHOD,"GET"],"");
```

Por sua vez, a Figura 5.5 mostra a parte do código php, presente no servidor WWW. A estrutura de servidores *OpenSim*, WWW e de banco de dados será melhor detalhada na Seção 5.2.2.

Figura 5.5 - Gravação das informações do sensor geográfico no banco de dados.

```
$aSQL = "insert into $tabela values (0, '$workflow', '$local', '$data', '$hora', '$idavatar', '$nomeavatar', '$presenca',  
 '$sensorid');";
```

Por sua vez, os sensores de interação são objetos com o objetivo específico de identificar o que ocorreu entre o avatar e o objeto (TIBOLA; TAROUCO, 2015). A Figura 5.3 mostra um objeto sensor de interação envolvido por um retângulo tracejado.

O sensor de interação da Figura 5.3 possui programação *script* LSL/OSSL para registrar a ação “sentar-se” do avatar, indicando que o aluno esteve presente e iniciou este questionário (*quiz*). As informações como nome e identificação do avatar, data, hora, nome do questionário, se o avatar se sentou ou se levantou da cadeira e nome da sala são agrupadas e enviadas para uma página php (*grava\_interacao\_local.php*), que se conecta com o banco de dados e armazena essas informações na tabela adequada. A Figura 5.6 apresenta a parte do *script* que envia as informações para a página php.

Figura 5.6 - Envio das informações do sensor de interação para o servidor WWW.

```
----- LOCAL HOST -----
chamadaHTTP =
  llHTTPRequest("http://127.0.0.1/opensim/grava_interacao_local.php?trilha="+trilha+"&workflow="+
+workflow+"&local="+local+"&nome_apresentacao="+nome_apresentacao+"&controle_acionado="
+controle_acionado+"&idavatar="+id_avatar+"&nomeavatar="+nome_avatar+ "&presenca="+presenca,
[HTTP_METHOD,"GET"], "");
```

Tal como ocorre com o sensor geográfico, a página php acionada pelo *script* do sensor de interação conecta o banco de dados e envia as informações para a respectiva tabela. A Figura 5.7 mostra o fragmento do código php que executa esse procedimento.

Figura 5.7 - Gravação das informações do sensor de interação no banco de dados.

```
$aSQL = "insert into $tabela values (0, '$workflow', '$local', '$data', '$hora', '$idavatar', '$nomeavatar', '$presenca', '$sensorid');";
```

Os sensores geográficos e de interação coletam os dados do avatar dos alunos e os enviam para o servidor WWW e este para o banco de dados. Algumas estruturas possíveis aos servidores *OpenSim*, WWW e de banco de dados são apresentadas na próxima seção.

## 5.2.2 Estrutura dos Servidores *OpenSim*, WWW e de banco de dados

Em seu formato-padrão, o *OpenSim* é acessado em um servidor disponível na *Internet*. Ele possui recursos para trabalhar em rede e, desta forma, comunicar-se com outros servidores conectados à WWW, que podem ser *OpenSim* ou não. Aqui, trata-se da comunicação entre os servidores pelo protocolo HTTP.

Os projetos que envolvem a utilização do *OpenSim* como ferramenta educacional estão atrelados à disponibilidade de recursos tecnológicos e humanos para sua implantação, execução e manutenção. Dessa forma, as configurações de um servidor *OpenSimulator* podem variar para cada projeto. Os servidores WWW e de banco de dados vão ser configurados dentro da mesma disponibilidade. Serão mostradas algumas sugestões, entendendo que as necessidades de recursos tecnológicos para os projetos usando o *OpenSim* são as mais distintas possíveis, bem como as estruturas para atendê-las.

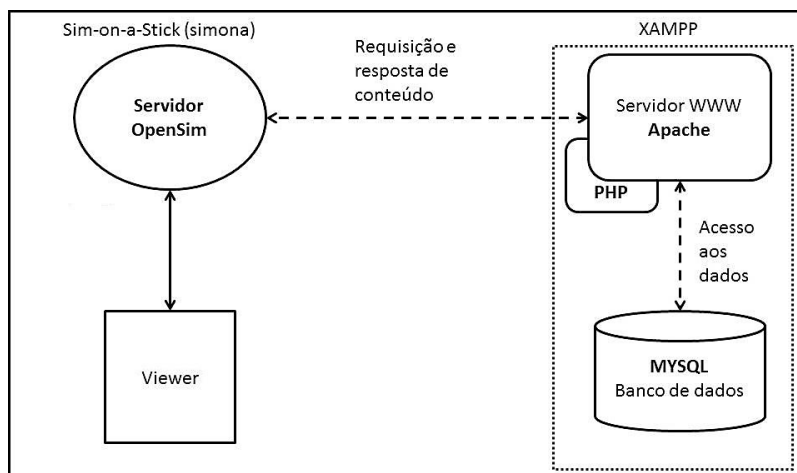
### 1) Estrutura *stand-alone* ou local

As escolas e universidades com recursos limitados para os projetos educacionais *OpenSim* ou que preferam manter os servidores dentro de seu controle podem optar por uma instalação local. Nesse caso, tanto o servidor *OpenSim* quanto os servidores WWW e de banco de dados são instalados em um único computador.

Para essa estrutura, pode ser utilizada uma versão do *OpenSim* chamada de *Sim-on-a-Stick* ou *simona* (SIM-ON-A-STICK, 2018). O *simona* é uma versão portátil, de instalação simples, uma vez que basta descompactar o arquivo de instalação em uma pasta no computador, executar o configurador com os parâmetros básicos e o servidor *OpenSim* estará pronto para uso. Uma vez que um arquivo do *Sim-on-a-Stick* tenha sido descompactado e o servidor configurado, a pasta onde ele foi instalado pode ser copiada para um *pen drive* ou para outro computador sem a necessidade de configurações adicionais (TIBOLA, 2018). Como os servidores WWW e de banco de dados não estão presentes no *simona*, é necessária a utilização de outro fornecedor para a sua instalação, como descrito a seguir.

Uma busca na *Internet* por servidores WWW e banco de dados portáteis resulta em várias opções. Aqui será usado o pacote XAMPP (PORTABLEAPPS.COM, 2018). O XAMPP é um pacote que disponibiliza os servidores Apache e MySQL, além de interpretar os *scripts* PHP, executáveis em mídias removíveis e que são pré-configurados e prontos para usar, bastando apenas descompactar o arquivo do XAMPP em uma pasta no computador. Com esse pacote é possível gravar as informações do aluno em um banco de dados MySQL, criar páginas em PHP para acessá-los e executar requisições HTTP a partir do *OpenSim*, sem a necessidade de permissões de administrador do *Windows*, configurações e permissões na rede de computadores local e remota. A Figura 5.8 mostra a estrutura de servidores no modo *stand-alone* proposta por Tibola (2018).

Figura 5.8 - Estrutura de servidores no modo *stand-alone*.



Uma estrutura *stand-alone*, como apresentada acima, pode ser duplicada e, assim, disponibilizar que um número maior de alunos possa se beneficiar do uso do projeto educacional implementado no *OpenSim*, de forma que os recursos computacionais fiquem sob o controle da instituição que o implementou.

## II) Estrutura remota

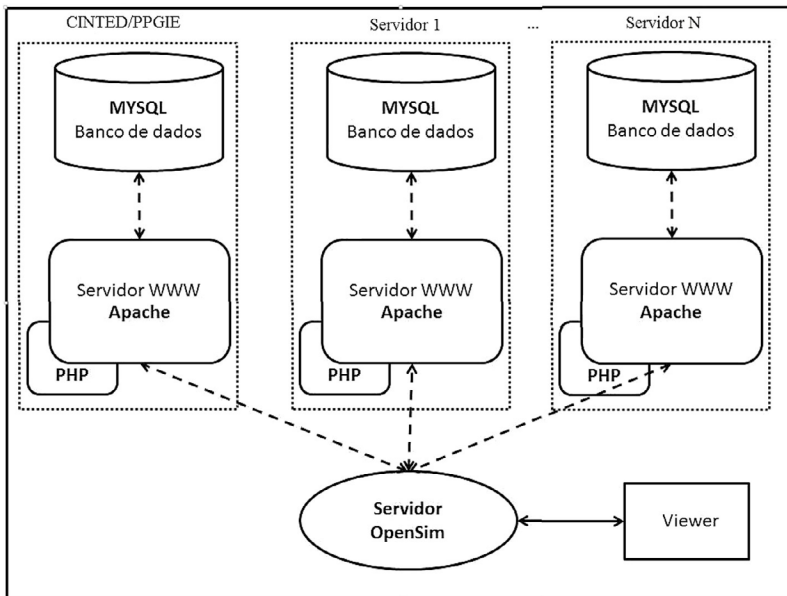
Quando a instituição de ensino dispõe de mais recursos tecnológicos ou o projeto educacional do *OpenSim* é mais amplo, a estrutura tende a ser mais complexa que a estrutura *stand-alone* descrita acima.

Uma estrutura com servidores independentes, interconectados e configurados para trabalhar em conjunto pode ser adequada quando o número de usuários e a demanda por acesso ao servidor *OpenSim* forem mais altos. Esse é o caso de instituições de ensino como a UFRGS, que, pelo CINTED/PPGIE, disponibiliza o Projeto AVATAR (AVATAR, 2018),

o qual possui uma estrutura de computadores em *grid*, em que o *OpenSim* (servidor de MVI), o Apache (servidor WWW) e o MySQL (servidor de banco de dados) estão instalados em computadores separados, mas interligados e com comunicação entre os três.

A versão do *OpenSim* instalada em uma estrutura semelhante à descrita anteriormente é a distribuição completa do *OpenSim* com uma configuração que permite usar todos os recursos disponíveis. A escolha do servidor Apache (APACHE, 2018), para os serviços de WWW, e do servidor MySQL (MYSQL, 2018), para o banco de dados, pode ser atribuída à popularidade, farta documentação, gratuidade, livre distribuição e conhecimento prévio destes servidores. A Figura 5.9 apresenta a estrutura de múltiplos servidores remotos descrita por Tibola (2018).

Figura 5.9 - Estrutura de servidores remotos.



A estrutura de servidores remotos é uma alternativa mais dispendiosa que a estrutura *stand-alone*, mas é a estrutura mais apropriada para projetos mais abrangentes, que demandem maior capacidade computacional e quantidade de acessos. Em ambas as estruturas de servidores, *stand-alone* e remotos, as informações são armazenadas em arranjos de dados. Um arranjo de dados possível é apresentado na sequência.

### 5.2.3 Estrutura das tabelas de dados

Esta seção apresenta as tabelas “interacao”, “sensores”, “hotpotatoes” e “pontos\_tarefas”, as quais podem ser usadas para armazenar informações relativas aos sensores geográficos e aos sensores de interação. A construção da base de dados e das tabelas depende de cada projeto e dos objetivos da utilização dos dados. Aqui será descrita a estrutura adaptada de Tibola (2018).

A tabela “interacao” é utilizada para armazenar as informações relativas às interações do avatar do aluno com os objetos, sejam eles primitivos (*prims*) ou objetos complexos. A tabela “interacao” é formada pelos campos: *codigo* (identificação única do registro), *local* (local no Mundo Virtual em que o experimento está posicionado), *nome\_experimento* (texto que identifica o experimento), *controle\_acionado* (para *prims* é o nome do próprio *prim*, para objetos complexos pode ser, por exemplo, botão ligar, botão pausar, botão desligar), *data* (dia, mês e ano de interação), *hora* (hora da interação no formato hora, minuto e segundo), *id\_avatar* (identificação numérica única do avatar), *nome\_avatar* (texto com o nome do avatar), *presenca* (identifica se o avatar interagiu com o



objeto ou não), *turma* (amostra à qual o aluno pertence – turma, grupo, etc...), *computador* (equipamento que o aluno utilizou – relacionado com o *login/logout*). A Figura 5.10 apresenta a tabela “interacao”.

Figura 5.10 - Tabela “interacao”.

#	Nome	Tipo
1	codigo 🗝️	int(11)
2	local	varchar(100)
3	nome_experimento	varchar(100)
4	controle_acionado	varchar(100)
5	data	date
6	hora	time
7	id_avatar	time
8	nome_avatar	varchar(100)
9	presenca	text
10	turma	varchar(50)
11	computador	varchar(50)

Fonte: Adaptado de Tibola (2018).

A tabela “sensores” guarda as informações relativas aos sensores geográficos acionados pelo avatar do aluno. Ela possui estes campos: *codigo* (identificação única do registro), *codigo\_workflow* (identificação da sequência de atividades recomendada para o aluno realizar), *local* (identificação do local onde o sensor está posicionado), *id\_avatar* (identificação numérica única do avatar), *nome\_avatar* (texto com o nome do avatar), *entrada\_saida* (identifica se o avatar entrou ou saiu do raio de alcance de detecção do sensor), *sensorID* (identificação do sensor que efe-

tuou o registro), *turma* (amostra à qual o aluno pertence – turma, grupo, etc...), *computador* (equipamento que o aluno utilizou – relacionado com o *login/logout*). A Figura 5.11 mostra a tabela “sensores”.

A integração do *OpenSim* com ferramentas externas é possível e amplia a possibilidade de uso de material instrucional. Uma integração aceitável e viável é a de material produzido no *HotPotatoes* (HOT POTATOES, 2018), dentro do *OpenSim*, por meio do servidor WWW. Desta forma, a tabela “hotpotatoes” reúne as informações dos questionários (*quizzes*) aplicados ao final da apresentação do conteúdo e da respectiva prática nos laboratórios educacionais virtuais, conforme relatado por Tibola (2018). A tabela “hotpotatoes” original foi levemente alterada para conter informações necessárias a análises futuras. Os campos desta tabela são: *codigo* (identificação única do registro), *realname* (nome do avatar que preencheu o questionário), *Exercise* (nome do questionário), *Score* (percentual de acertos alcançado), *Start\_Time* (hora de início de preenchimento do questionário), *End\_Time* (hora do fim do preenchimento do questionário), *Start\_Date* (data de início do preenchimento do questionário), *End\_Date* (data do fim do preenchimento do questionário), *turma* (amostra à qual o aluno pertence – turma, grupo, etc...), *computador* (equipamento que o aluno utilizou – relacionado com o *login/logout*). A Figura 5.12 mostra a tabela “hotpotatoes”.

Figura 5.11 - Tabela “sensores”.

#	Nome	Tipo
1	<u>codigo</u> 🗝️	int(11)
2	<u>codigo_workflow</u>	int(11)
3	<u>local</u>	int(11)
4	<u>data_local</u>	date
5	<u>hora_local</u>	time
6	<u>id_avatar</u>	varchar(100)
7	<u>nome_avatar</u>	varchar(100)
8	<u>entrada-saida</u>	text
9	<u>sensorID</u>	decimal(10,5)
10	<u>turma</u>	varchar(50)
11	<u>computador</u>	varchar(50)

Fonte: Adaptado de Tibola (2018).

Figura 5.12 - Tabela “hotpotatoes”.

#	Nome	Tipo
1	<u>codigo</u> 🗝️	int(11)
2	<u>realname</u>	varchar(50)
3	<u>Exercise</u>	varchar(50)
4	<u>Score</u>	int(5)
5	<u>Start_Time</u>	time
6	<u>End_Time</u>	time
7	<u>Start_Date</u>	date
8	<u>End_Date</u>	date
9	<u>turma</u>	varchar(50)
10	<u>computador</u>	varchar(50)

Fonte: Adaptado de Tibola (2018).

Uma forma de tornar os ambientes educacionais mais desafiadores para os alunos e aproximá-los de seus hábitos de lazer é utilizar técnicas de gamificação na construção das atividades disponibilizadas a eles. Dentro do contexto projetado por Tibola (2018), a tabela “pontos\_tarefas” mantém os dados relativos à pontuação correspondente ao sucesso na realização das atividades realizadas pelo aluno. Os campos da tabela “pontos\_tarefas” são: *codigo* (identificação única do registro), *nome\_avatar* (nome do avatar que realizou a atividade), *tarefas* (identificação da atividade realizada: visualização de material instrucional – texto, vídeo, página WWW, *slides*; realização de práticas no laboratório educacional virtual; preenchimento de questionários), *pontos* (quantidade de pontos alcançados pelo aluno em relação à pontuação total possível para a atividade), *turma* (amostra à qual o aluno pertence – turma, grupo, etc...), *computador* (equipamento que o aluno utilizou – relacionado com o *login/logout*). A Figura 5.13 mostra a tabela “pontos\_tarefas”.

Figura 5.13 - Tabela “pontos\_tarefas”.

#	Nome	Tipo
1	<u>codigo</u> 🗝️	int(11)
2	<u>nome_avatar</u>	varchar(50)
3	<u>tarefas</u>	varchar(500)
4	<u>pontos</u>	int(11)
5	<u>turma</u>	varchar(50)
6	<u>computador</u>	varchar(50)

Fonte: Adaptado de Tibola (2018).

Uma vez que os dados estejam armazenados, realiza-se a sua análise. A seção seguinte descreve algumas ferramentas que podem ser usadas para tratamento de dados.

## 5.3 ANÁLISE DOS DADOS

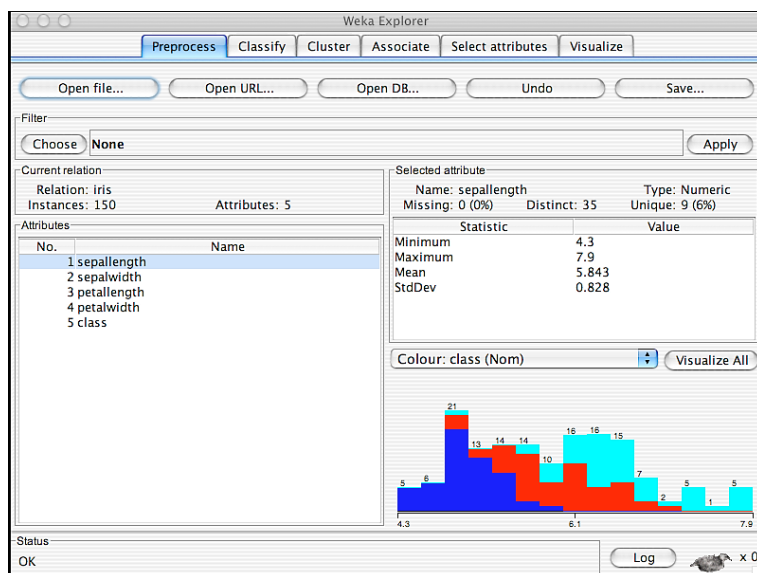
A coleta de dados das ações do aluno no MVI é importante para que, por meio de ferramentas adequadas, possam ser descobertas informações relevantes para que o ambiente tridimensional atinja o objetivo educacional pretendido. Uma vez que os MVIs podem gerar grandes volumes de dados, uma análise mais apurada se faz necessária. Uma alternativa para a obtenção de informações importantes sobre o uso do MVI é a aplicação de técnicas de mineração de dados educacionais.

Segundo a IEDMS (2018), na Mineração de Dados Educacionais – MDE (ou *Educational Data Mining* – EDM), os dados educacionais são retirados do uso que o aluno faz de ambientes interativos de aprendizagem, ambientes de aprendizagem colaborativa suportada por computador ou de dados administrativos de escolas e universidades. Eles, frequentemente, possuem vários níveis de hierarquia, as quais, muitas vezes, precisam ser determinadas pelas propriedades dos dados em si antes que o processo ocorra. Questões de tempo, sequência e contexto também desempenham papéis importantes no estudo dos dados educacionais. Em virtude da complexidade deste tema e da restrição de es-

paço neste Capítulo, são apresentadas algumas ferramentas, deixando a discussão das técnicas de Mineração de Dados Educacionais para trabalhos futuros.

Uma ferramenta popular para a mineração de dados educacionais é o Weka, uma coleção de algoritmos de aprendizagem de máquina para tarefas de mineração de dados. Os algoritmos podem ser aplicados diretamente a um conjunto de dados ou chamados a partir de seu próprio código Java. O Weka contém ferramentas para pré-processamento de dados, classificação, regressão, agrupamento, regras de associação e visualização. Também é adequado para o desenvolvimento de novos esquemas de aprendizado de máquinas (WEKA, 2018). A Figura 5.14 apresenta a interface do Weka.

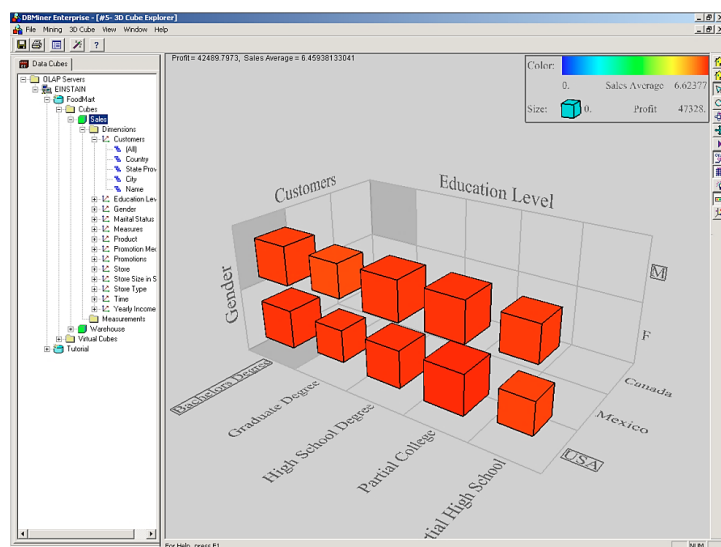
Figura 5.14 – WEKA.



Fonte: Weka (2018).

Outra ferramenta que pode ser usada para a descoberta de conhecimento em grandes volumes de dados é o DBMiner. O DBMiner é um sistema de mineração de dados desenvolvido para a mineração interativa de conhecimento, em múltiplos níveis, em grandes bancos de dados relacionais e *data warehouses*. O sistema implementa um amplo espectro de funções de mineração de dados, incluindo caracterização, comparação, associação, classificação, previsão e clusterização. Ao incorporar várias técnicas de mineração de dados, incluindo OLAP e indução orientada a atributos, análise estatística, aprofundamento progressivo para mineração de conhecimento em vários níveis e mineração orientada por metarregra, o sistema oferece um ambiente de mineração de dados interativo e fácil de usar (HAN *et al.*, 1999). A Figura 5.15 demonstra o DBMiner.

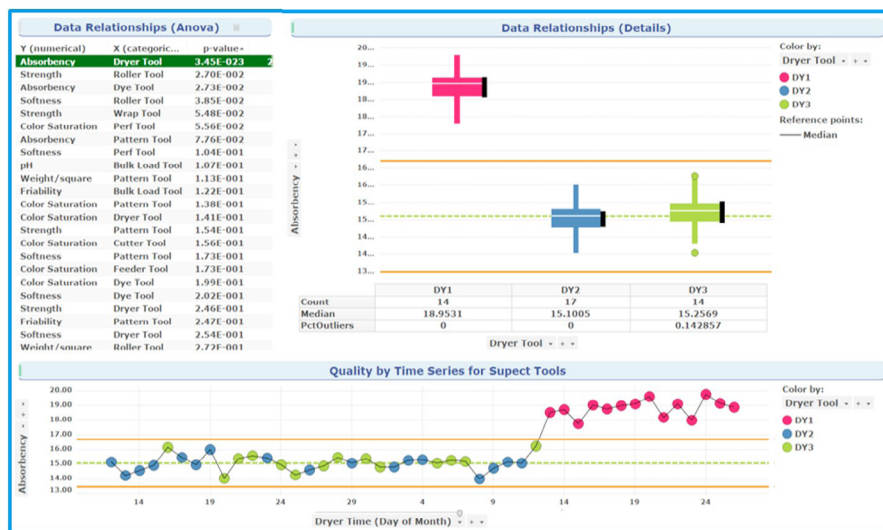
Figura 5.15 - DBMiner.



Fonte: GTU (2017).

Já o *software Statistica* (TIBCO, 2018) é um programa integrado para o gerenciamento de análises estatísticas e bases de dados, caracterizando uma ampla seleção do processo analítico, do básico ao avançado, para as mais diversas áreas – Ciências Biomédicas, Biológicas, Engenharias, Ciências Sociais, Agrárias – permitindo a realização de Mineração de Dados (*Data Mining*). O sistema não inclui somente procedimentos estatísticos e gráficos gerais, mas também módulos especializados: Análise de Regressão, Análise de Sobrevivência, Séries Temporais, Análise Fatorial, Análise Discriminante e diversos outros módulos (OGLIARI; PACHECO, 2011). Uma das interfaces do *Statistica* pode ser visualizada na Figura 5.16.

Figura 5.16 – *Statistica*.



Fonte: TIBCO (2018).



Além das ferramentas relacionadas acima, muitas outras estão disponíveis, cada uma dispondo de um conjunto de recursos estatísticos e de conexão com fontes de dados que as tornam mais ou menos adequadas às necessidades do projeto educacional pretendido.

## REFERÊNCIAS

APACHE. **The Apache HTTP server project**. The Apache Software Foundation, Forest Hill, MD, USA, 2018. Disponível em: <<https://httpd.apache.org/>>. Acesso em: 5 dez. 2018.

BELL, Mark W. Toward a definition of “Virtual Worlds”. **Journal of virtual worlds research**, v. 18, n. 1, Jan. 2008.

CARVALHO JÚNIOR, Gabriel Dias de. **Trajetórias de aprendizagem de alunos de ensino médio**: produção de significados em um curso introdutório de física térmica. 2005. 265 f. Tese (Doutorado) – Faculdade de Educação, Universidade Federal de Minas Gerais, Belo Horizonte, 2005.

GIRVAN, Carina; SAVAGE, Timothy. Identifying an appropriate pedagogy for virtual worlds: a communal constructivism case study. **Computers & education**, v. 55, n. 1, p. 342-349, 2010.

GTU. **Study of DBMiner tool**. 2017. Disponível em: <<http://gtubeit7.blogspot.com/2017/09/study-of-db-miner-tool.html>>. Acesso em: 4 ago. 2019.

HAN, Jiawei *et al.* **DBMiner**: a system for data mining in relational databases and data warehouses. Simon Fraser University, British Columbia, Canada, 1999. Disponível em: <<http://hanj.cs.illinois.edu/pdf/cascon97.pdf>>. Acesso em: 5 dez. 2018.

HERPICH, Fabrício; NUNES, Felipe Becker; CAZELLA, Silvio Cesar; TAROUCO, Liane Margarida Rockenbach. Mineração de dados educacionais: uma análise sobre o engajamento de usuários em mundos virtuais. *In: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – WCBIE*, 5., 2016, Uberlândia. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2016.

HOT POTATOES. **Hot Potatoes home page**. Half-Baked Software, Victoria, Canada, 2018. Disponível em: <<https://hotpot.uvic.ca/>>. Acesso em: 5 dez. 2018.

IEDMS – INTERNATIONAL EDUCATIONAL DATA MINING SOCIETY. **Homepage**. IEDMS, Worcester, MA, USA, 2018. Disponível em: <<http://www.educationaldatamining.org/>>. Acesso em: 5 dez. 2018.

LINDEN RESEARCH. **Second Life wiki**. Getting started with LSL. Linden Research, Boulder, CO, USA, 2015. Disponível em: <[http://wiki.secondlife.com/wiki/Help:Getting\\_started\\_with\\_LSL](http://wiki.secondlife.com/wiki/Help:Getting_started_with_LSL)>. Acesso em: 5 dez. 2018.

LSL WIKI. **Homepage**. Linden Research, Boulder, CO, USA, 2012. Disponível em: <<http://lslwiki.digiworldz.com/>>. Acesso em: 5 dez. 2018.

MACEU, Joselice Siebra; SOUSA, Verônica Lima Pimentel de. Um ambiente virtual imersivo para o ensino de sistemas operacionais: concepção e primeiras análises. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 16, n. 1, jul. 2018.

MYSQL. **Homepage**. Oracle Corporation, Redwood City, MA, USA, 2018. Disponível em: <<https://www.mysql.com/>>. Acesso em: 5 dez. 2018.

NUNES, Felipe Becker. **Um método de ensino pautado na aprendizagem integrada aos mundos virtuais e princípios do *mastery learning***. 2017. 225 f. Tese (Doutorado) – Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

NUNES, Felipe Becker; HERPICH, Fabrício; TAROUCO, Liane Margarida Rockenbach; LIMA, José Valdeni de. Monitoramento e avaliação de estudantes em mundos virtuais. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 14, n. 1, 2016.

O'BRIEN, Heather L.; TOMS, Elaine G. What is user engagement? A conceptual framework for defining user engagement with technology. **Journal of the American Society for Information Science & Technology**, v. 59, n. 6, p. 938-955, 2008.

OGLIARI, Paulo José; PACHECO, Juliano Anderson. **Análise estatística usando o Statística 6.0**. Centro Tecnológico, Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2004. Disponível em: <[https://moodle.ufsc.br/pluginfile.php/1087168/mod\\_resource/content/0/statistica.pdf](https://moodle.ufsc.br/pluginfile.php/1087168/mod_resource/content/0/statistica.pdf)>. Acesso em: 5 nov. 2018.

- OPENSIM. **Scripting languages**. USA, 2012. Disponível em: <[http://OpenSimulator.org/wiki/Scripting\\_Languages](http://OpenSimulator.org/wiki/Scripting_Languages)>. Acesso em: 5 dez. 2018.
- OPENSIM. **What is OpenSimulator?** USA, 2018. Disponível em: <[http://OpenSimulator.org/wiki/Main\\_Page](http://OpenSimulator.org/wiki/Main_Page)>. Acesso em: 5 dez. 2018.
- PORTABLEAPPS.COM. **XAMPP launcher**: complete portable server. New York, 2018. Disponível em: <<https://portableapps.com/apps/development/xampp>>. Acesso em: 5 dez. 2018.
- PROJETO AVATAR. **Homepage**. Universidade Federal do Rio Grande do Sul, Porto Alegre, 2018. Disponível em: <<http://www.ufrgs.br/avatar>>. Acesso em: 5 nov. 2018.
- SAINT-GEORGES, Ingrid de; FILLIETTAZ, Laurent. Situated trajectories of learning in vocational training interactions. **European journal of psychology of education**, v. 23, n. 2, p. 213-233, 2008.
- SGOBBI, Fabiana Santiago; TAROUÇO, Liane Margarida Rockenbach; REATEGUI, Eliseo. The pedagogical use of the Internet of Things in virtual worlds to encourage a behavior change in obese individuals. *In: IEEE INTERNATIONAL CONFERENCE ON INTERNET OF THINGS, 2017, Exeter, UK. Proceedings [...]*. USA: Institute of Electrical and Electronics Engineers – IEEE, 2017. v. 10, p. 676-682.
- SIM-ON-A-STICK. **Sim-on-a-Stick 0.8**. USA, 2018. Disponível em: <<http://simonastick.com/>>. Acesso em: 5 nov. 2018.
- SIMON, Martin A. Reconstructing mathematics pedagogy from a constructivist perspective. **Journal for research in mathematics education**, v. 26, n. 2, p. 114-145, 1995.
- TIBCO. **TIBCO Statistica**. TIBCO Software, Palo Alto, CA, USA, 2018. Disponível em: <<https://www.tibco.com/products/tibco-statistica>>. Acesso em: 5 nov. 2018.
- TIBOLA, Leandro Rosniak. **Fatores ensejadores de engajamento em ambientes de mundos virtuais**. 2018. 235 f. Tese (Doutorado em Informática na Educação) – Centro de Estudos Interdisciplinares em Novas Tecnologias, Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2018.

TIBOLA, Leandro Rosniak; TAROUCO, Liane Margarida Rockenbach. Rastreamento de interações em laboratórios educacionais nos mundos virtuais 3D para identificação de engajamento. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 13, n. 2, p. 10, 2015. Disponível em: <<https://seer.ufrgs.br/renote/article/view/61440/36327>>. Acesso em: 5 nov. 2018.

WEKA. **Weka 3**: data mining with open source machine learning software in Java. University of Waikato, New Zealand, 2018. Disponível em: <<http://www.cs.waikato.ac.nz/ml/weka/>>. Acesso em: 5 nov. 2018.

WILSON, P. Holt; MOJICA, Gemma F.; CONFREY, Jere. Learning trajectories in teacher education: supporting teacher's understandings of student's mathematical thinking. **Journal of mathematical behavior**, v. 32, p. 103-121, 2013.

ZAHARIAS, Panagiotis; ANDREOU, Ioannis; VOSINAKIS, Spyros. Educational virtual worlds, learning styles and learning effectiveness: an empirical investigation. *In*: PAN-HELLENIC CONFERENCE WITH INTERNATIONAL PARTICIPATION – INFORMATION AND COMMUNICATION TECHNOLOGIES IN EDUCATION, 7., 2010, Korinthos, Greece. **Proceedings** [...]. [S.l.]: Hellenic Association of ICT in Education (HAICTE), 2010.

ZIMMER, Josete Maria; VEZZANI, Marco Antônio. Second Life para educação à distância: uma experiência entre estudantes brasileiros e portugueses. *In*: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – WCBIE, 6., 2017, Recife. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2017.

# 6

## Estratégias pedagógicas para usar o MVI

**Bárbara Gorziza Avila - IFRGS**  
bapadoin@gmail.com

**Fabiana Santiago Sgobbi - PPGIE - UFRGS**  
fabianasgobbi@gmail.com

**Felipe Becker Nunes - AMF**  
nunesfb@gmail.com

**Liane Margarida Rockenbach Tarouco**  
**PPGIE - UFRGS - liane@penta.ufrgs.br**

**Leandro Rosniak Tibola - PPGIE - UFRGS**  
lrtibola@gmail.com

## 6.1 EXPERIÊNCIA DO ALUNO NO MVI

As plataformas de Mundos Virtuais são uma tecnologia inovadora, no âmbito das TICs. Fornecem ferramentas para a criação de ambientes gráficos tridimensionais interativos e altamente imersivos. Esses podem ser réplicas de espaços físicos existentes, de espaços imaginários ou de espaços impossíveis de aceder, por questões de segurança ou de custo (MARATOU; XENOS, 2014). O aprendizado deve ser proposto de forma a propiciar aos alunos o desenvolvimento de uma compreensão do mundo que lhes dê condições de continuamente colher e processar informações, desenvolver sua comunicação, avaliar situações, tomar decisões, ter atuação positiva e crítica em seu meio social (BRASIL, 1998). Nesse contexto, o uso das Tecnologias da Informação e Comunicação representa um importante papel no processo de mudança de postura e compreensão do aprendizado que está sendo construído continuamente no ambiente escolar e fora dele.

Em virtude das inerentes possibilidades que têm surgido no decorrer das últimas décadas, como a aplicação de recursos de realidade virtual, realidade aumentada, *Internet das Coisas*, agentes virtuais, entre outros, sua aplicabilidade no contexto educacional sofreu um grande impacto positivo e inovador. No contexto da realidade virtual, de meados de 2005 em diante, houve uma expansão significativa no uso de ambientes virtuais conhecidos como Mundos Virtuais, cujos conceitos formadores foram explicados detalhadamente nos Capítulos iniciais desta obra. O grau de imersão e envolvimento dos alunos com o

conteúdo do curso, com os colegas e com os professores é totalmente diferente daquele dos ambientes tradicionais de ensino, tornando impossível sua reprodução em tais ambientes (ZIMMER; VEZZANI, 2017).

A intensificação do uso desse tipo de ambiente tridimensional em diferentes campos de aplicação possibilitou a sua inserção no âmbito educacional, do qual emergem novas perspectivas de uso desses recursos computacionais. Johnson *et al.* (2009) entendem que esses ambientes podem oferecer um novo tipo de abordagem, na qual os usuários passam de um estado considerado passivo e observador para se tornarem sujeitos mais ativos dentro do Mundo Virtual.

Oliveira *et al.* (2016) entendem que, em um Mundo Virtual planejado com esses fins educacionais, os multiusuários têm a possibilidade de circular livremente e desenvolver seus processos de ensino e aprendizagem, conforme sua própria demanda. Essa linha de pensamento é corroborada por Simsek e Can (2016), os quais afirmam que, ao proporcionar aos alunos a liberdade de escolher o tipo de material de aprendizagem a explorar, o Mundo Virtual os torna indivíduos ativos em seu processo de aprendizagem, desenvolvendo assim a impressão de autoria durante esse processo.

Em um ambiente virtual tridimensional voltado ao contexto educacional, viabilizam-se variadas possibilidades a serem exploradas pelos docentes e pesquisadores, as quais podem ser diversificadas quanto ao uso de recursos multimídia, como vídeos, imagens, textos, animações, entre outros, assim como podem ser resultantes da interação dos alunos

com experimentos estáticos ou simulações dinâmicas, disponibilizadas no ambiente virtual para complementar o ensino de um determinado tópico de aprendizagem.

Também é possível citar o uso de *chat*, via texto ou voz, a navegação pelos cenários dispostos e a interação com os elementos presentes no MVI, o que pode gerar um cenário propício para que esse tipo de transformação de atitude conduza a um estado considerado ativo/crítico. O ambiente motiva a colaboração por meio da exploração e descoberta, e a interação social permite o desenvolvimento de atividades de aprendizagem formais e informais, em paralelo ou não, com as atividades de aprendizagem do mundo real (ZIMMER; VEZZANI, 2017).

Distinguem-se, nesse tipo de ambiente, as oportunidades inerentes ao desenvolvimento de atividades de cunho prático e simulações, que demonstram experimentos corriqueiramente difíceis de serem visualizados em laboratórios reais, por custo e/ou perigos intrínsecos à sua execução junto aos alunos. Pellas (2014) destaca que as simulações interativas fornecem uma ilusão plausível, que permite aos usuários ter uma experiência que reflita situações realistas usando esse tipo de ambiente.

A construção de um simulador tridimensional multiutilizador fornece um conjunto de novas oportunidades que podem ser usadas na formação tanto por formadores como por formandos (PINHEIRO *et al.*, 2012). As simulações em ambiente virtual vêm sendo utilizadas nos mais diferentes cenários de educação e formação, tendo sido defendido entre a comunidade científica que a utilização desse tipo de simulações é, em geral, benéfica para a aprendizagem de procedimentos (MOR-



GADO, 2009). Dessa forma, os MVIs podem fornecer aos alunos a prática experimental de um domínio particular de conhecimento, que pode ser realizada com laboratórios virtuais e objetos 3D, agentes inteligentes, dentre outros fatores (HERPICH *et al.*, 2016). Chang e Law (2008) destacam, também, que o uso de simulações nos Mundos Virtuais tem um número de características que são de especial ajuda no ensino de Ciências, Física, Química e Biologia.

Chiu (2015) explica que experiências de laboratório físicas, em áreas de ensino como as citadas anteriormente, capacitam os alunos a interagir com fenômenos científicos observáveis, mas os estudantes, muitas vezes, não conseguem fazer conexões com os comportamentos de nível molecular subjacente. Experiências laboratoriais virtuais e visualizações baseadas em computador capacitam os alunos a interagir com conceitos científicos não observáveis, mas os estudantes podem ter dificuldades para se conectarem a instâncias reais do fenômeno observado (GUILLERMO, 2016).

Nesse contexto, os laboratórios virtuais construídos em Mundos Virtuais apresentam vantagens em relação aos laboratórios convencionais: entre elas, a possibilidade de repetição de experimentos com ou sem alteração de variáveis; a simulação de atividades experimentais que necessitam de reagentes que trariam risco à sua manipulação, por serem tóxicos ou inflamáveis; a não produção de resíduos químicos, entre outros. Nas circunstâncias em que a atenção do estudante esteja mobilizada por uma simulação de um fenômeno, o controle sobre variáveis no Mundo Virtual pode ser exercido com o intuito de observar regularidades, estabelecer premissas, fazer previsões, ou, ainda, a própria

representação visual do fenômeno simulado pode servir de referência para o estudante, elaborações, explicações relacionadas ao fenômeno observado (GIORDAN, 2008).

As simulações podem proporcionar mediações que relacionem os fenômenos macroscópicos e microscópicos em uma construção teórica que nem sempre encontra sustentação empírica para medições. É no ambiente de simulação que se pode formar cenários estimuladores para a criação de representações mentais por parte do sujeito, que passa a reconhecer nos modelos então simulados uma instância intermediária entre suas representações internas e as representações externas do fenômeno (GIORDAN, 2008). Além dessas características, laboratórios virtuais nesse tipo de ambiente constituem espaços de aprendizagem que proporcionam o prolongamento das atividades educacionais para além das paredes da escola, uma vez que possibilitam aos estudantes realizar experimentos a partir de qualquer local, em qualquer horário e com custo reduzido (NUNES *et al.*, 2017a).

Destaca-se que, apesar de esses indícios positivos serem identificados na realização de atividades educacionais nesse tipo de ambiente, os Mundos Virtuais possuem diferentes tipos de limitações em seu modo de aplicação, diversas comprovadas durante o período de testes realizados nesta pesquisa. Potkonjaka *et al.* (2016) explicam que esse tipo de ambiente não foi criado para fins educacionais, sendo necessária a realização de treinamentos com os usuários, ressaltando, também, a complexidade existente em criar objetos 3D, o que exige o uso de

*softwares* específicos, como *SketchUp* e *Blender*, que fornecem o suporte adequado para a modelagem e exportação desses elementos para o Mundo Virtual Imersivo.

Problemas envolvendo a dificuldade de acesso ao ambiente, por instabilidade na velocidade de conexão da *Internet* e recursos limitados de *hardware*, também podem ser considerados empecilhos para sua utilização. Pesquisas que abordam tais dificuldades podem ser vistas em Young (2010), Smith-Robbins (2011) e Nunes *et al.* (2017a). Além disso, a ausência de suporte para acesso utilizando dispositivos móveis pode ser considerada um dos principais problemas enfrentados, atualmente, nas pesquisas desenvolvidas com Mundos Virtuais, conforme pode ser visto na pesquisa de Voss *et al.* (2013). Moran *et al.* (2013) destacam que as tecnologias móveis trazem grandes desafios, pois descentralizam os processos de gestão do conhecimento e permitem o aprendizado em qualquer hora e em qualquer lugar.

Em virtude da eminente evolução dos dispositivos móveis e de sua crescente utilização, torna-se uma tendência que os participantes queiram acessar esse tipo de ambiente em seus dispositivos móveis. Tal suporte já pode ser considerado um dos principais requisitos a serem explorados futuramente, uma vez que somente uma solução mais robusta, conhecida como *Lumiya*, está disponível atualmente para acessar um Mundo Virtual Imersivo. Voss *et al.* (2013) asseveram que se trata de um *software* proprietário, que demanda de recursos de *hardware* mais robustos para que o ambiente possa ser acessado, o que dificulta o processo.

O pensamento compartilhado por Jacka (2015) reflete a situação atual. A autora acredita que exista um senso comum de que os Mundos Virtuais estão em processo de construção, ainda havendo muito a ser feito antes que os professores, estudantes e gerentes adotem plenamente essa solução como um espaço de aprendizado. Apesar das desvantagens descritas, os benefícios providos nesse tipo de abordagem têm se mostrado instigadores para prosseguir no desenvolvimento de cenários e simulações com caráter educacional.

Tüzün e Özdiñç (2016) explicam que os MVIs têm o potencial de permitir que os indivíduos obtenham dados mais significativos em longo prazo em comparação com os ambientes multimídia interativos e tradicionais. Yilmaz *et al.* (2015) destacam que esses componentes multimídia podem ajudar na interatividade e nas diversidades de opções no ambiente, observando que estudantes com diferentes necessidades e preferências podem se sentir bem situados. Adjunto aos componentes multimídia, que podem trazer benefícios ao processo de ensino/aprendizagem, também é possível a criação e interação com agentes virtuais dentro do Mundo Virtual.

### **6.1.1 Experiência do aluno no MVI e a orientação de agentes virtuais**

Os Mundos Virtuais tridimensionais oferecem a oportunidade de o utilizador se envolver num ambiente digital que imita ou simula o ambiente físico, sendo a sua exploração e desenvolvimento cada vez maior (DREHER *et al.*, 2009). Podem também gerar oportunidades para facili-

tar a comunicação e a colaboração com outros tipos de agentes (PELET *et al.*, 2011). Os agentes *Non-Player Characters* (NPCs) são entidades programáveis com o uso de *scripts* que desempenham ações predeterminadas. Além disso, integra este recurso uma base de conhecimento de um *chatbot*, que permite simular conversações entre o agente e o usuário em alto padrão, utilizando uma base de conhecimento personalizável.

No âmbito educacional, os NPCs podem desempenhar uma ação semelhante à de um tutor, por meio de uma ligação com uma base de conhecimento de um *chatbot*. Esse tipo de entidade representa programas de computador que tentam simular conversações com os usuários, com o objetivo de, pelo menos temporariamente, levar um ser humano a pensar que está conversando com outra pessoa (PILASTRI; BREGA, 2009).

Também é possível o uso de agentes virtuais inteligentes no contexto dos MVIs. Os dados provenientes desse tipo de ambiente, somados à sua interligação com uma base de conhecimento adaptada ao conteúdo trabalhado na disciplina abordada, poderiam fornecer a capacidade de esses agentes se tornarem tutores virtuais sensíveis ao contexto do usuário e, ao mesmo tempo, sanar dúvidas referentes aos tópicos trabalhados no ambiente.

A interação dos alunos com os recursos do Mundo Virtual e com NPCs também possibilita a coleta de diversos tipos de dados, conforme pode ser visto no trabalho de Nunes *et al.* (2016). Esses tipos de dados estão relacionados aos locais que um aluno visitou no ambiente, com que tipos de recursos multimídia ele interagiu, quanto tempo e quantas vezes esse tipo de interação ocorreu, quais dúvidas e impressões

ele mencionou no *chat* do ambiente, dentre outros tipos de dados que podem ser coletados, a partir de uma interação do aluno no MVI, e processados de acordo com a necessidade ou preferência do docente e/ou pesquisador.

Dentre as possibilidades inerentes aos dados armazenados, está o mapeamento dos trajetos percorridos pelos usuários durante cada interação no ambiente. Saint-Georges e Filliettaz (2008) entendem que esse tipo de mapeamento permite inferir o trajeto efetivamente realizado por cada estudante e a possibilidade de monitoramento com vista a um planejamento em longo prazo por parte do professor. Os autores explicam que o uso desse termo está ligado a duas proposições: a perspectiva situacional e a ideia de trajetória. A primeira diz respeito ao foco nas ações que ocorrem em tempo real e são realizadas pelos indivíduos, enquanto a segunda envolve a ideia de trajetória, que implica ter uma visão mais voltada para o futuro, considerando ações em longo prazo. Portanto, tanto na parte de monitoramento do aluno (para verificar quais ações ele tem desempenhado nas interações no Mundo Virtual) quanto no planejamento das atividades pelo professor, a análise das trajetórias dos alunos poderia ser aplicada. Dessa forma, os agentes virtuais poderiam ser criados no MVI com o intuito de formular e apresentar sugestões de trajetos baseados nos materiais de aprendizagem presentes no ambiente. Tal proposição visa a auxiliar os alunos durante sua interação, propondo um percurso pedagógico para ser percorrido no ambiente, como também armazenar os dados para futuras análises relacionadas à avaliação do aluno e ao planejamento por parte do professor.

Outra possibilidade presente nesse tipo de interação é a análise textual por meio de algoritmos de mineração de dados, mediante os quais os registros armazenados referentes às conversações realizadas no MVI poderiam ser analisados com o intuito de averiguar quais tipos de tópicos estão sendo mencionados pelos alunos, quais dificuldades ou questionamentos estão sendo realizados, entre outras possibilidades. Tais análises poderiam resultar na formulação de respostas, sugestões ou até questionamentos por parte dos agentes virtuais disponibilizados no MVI, com o intuito de estimular os alunos a interagirem com ele e construir seu aprendizado com seu auxílio.

No momento em que o agente conversacional NPC detém o conhecimento que está sendo debatido, ele pode processar a requisição e expressar sua resposta de acordo com aquilo que sua pesquisa retornou (NUNES *et al.*, 2017b). Os autores ressaltam que, nas circunstâncias em que o *chatbot* não dispõe de determinada informação em sua base de dados, irá empenhar-se em conduzir a conversação até que as expressões utilizadas pelo estudante sejam encontradas na base de conhecimento do agente conversacional. Conforme mencionado anteriormente, as possibilidades inerentes ao uso de agentes virtuais são diversificadas. Em especial, os seus trajetos nesse ambiente são relevantes para a análise do progresso do aluno e da interação com o MVI. Dessa forma, a subseção a seguir explora de forma detalhada os aspectos inerentes a esse tipo de abordagem e quais seus impactos na interação do aluno e seu aprendizado.

## **6.2 VERIFICAÇÃO DAS TRAJETÓRIAS DO ALUNO OU IDENTIFICAÇÃO DO PERCURSO EDUCACIONAL DO ALUNO**

### **6.2.1 Verificação de aprendizagem**

Considerando a estratégia educacional utilizada, a avaliação do desempenho do aluno em relação aos tópicos abordados em um MVI pode ser verificada de várias maneiras. Uma alternativa é o monitoramento da movimentação do avatar do aluno pelo MVI e a inserção de questionários eletrônicos em locais onde há a troca do assunto abordado.

#### **6.2.1.1 Monitoramento da movimentação do avatar do aluno**

O monitoramento da movimentação do avatar pelo Mundo Virtual pode ser realizado por meio da utilização dos sensores, como descrito por Tibola e Tarouco (2015). De modo geral, os sensores são os objetos do MVI que possuem código LSL/OSSL embutido. Possibilitam reconhecer e registrar a presença do avatar, bem como registrar suas ações durante a interação com um objeto. Os sensores podem ser de dois tipos: (I) sensores geográficos e (II) sensores de interação. Aqui serão tratados os sensores geográficos, uma vez que estão dentro do escopo deste capítulo.

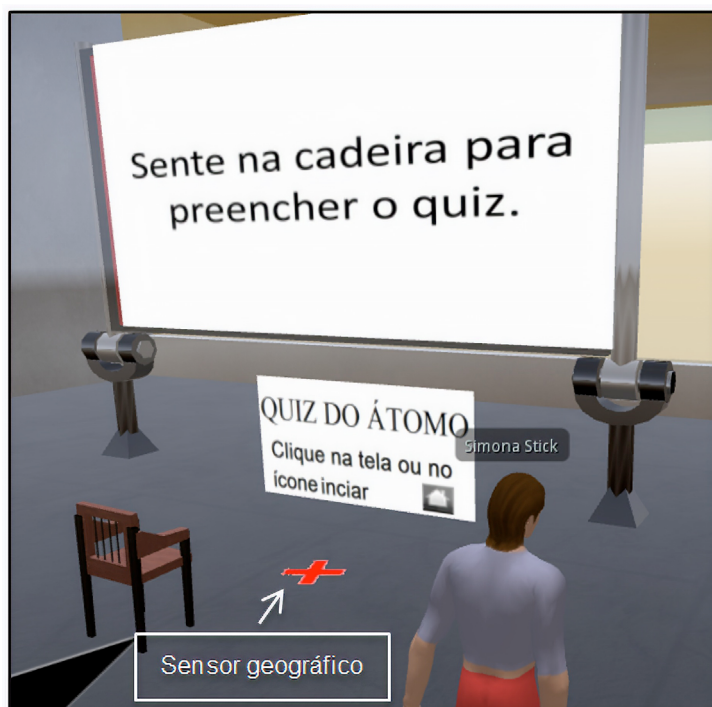
Enquanto o aluno está imerso no MVI, a presença dos sensores geográficos não é percebida por ele, uma vez que o código LSL/OSSL, que faz a detecção e o registro da passagem do aluno por aquele local,



é executado sem a apresentação de mensagens que o identifiquem no ambiente virtual. Ainda, os sensores geográficos permitem verificar o posicionamento do aluno dentro do MVI, se ele entrou e saiu de um prédio ou sala, se passou por determinado lugar, se andou ou voou, conforme a necessidade de registro (TIBOLA; TAROUCO, 2015).

A Figura 6.1 mostra um objeto, no formato de um “X”, o qual é um sensor geográfico. Nesse caso, o objeto “X” tem o objetivo de indicar uma posição para onde o avatar deve se dirigir e, ao mesmo tempo, ser um sensor geográfico. Se for do interesse do projeto do ambiente 3D, as propriedades do objeto podem ser alteradas para torná-lo invisível e mantê-lo como um sensor (TIBOLA, 2018).

Figura 6.1 - Objeto visível com a função de sensor geográfico.



O monitoramento da movimentação do avatar do aluno em um MVI é útil quando se almeja fazer a comparação de uma trajetória educacional recomendada com a trajetória efetivamente realizada pelo aluno, permitindo que possam ser feitas análises dessa relação e dos reflexos das escolhas dos alunos. Além da movimentação do avatar do aluno, a avaliação é completada com a inserção de questionários eletrônicos no MVI, descrita a seguir.

### **6.2.1.2 Questionários eletrônicos com o *HotPotatoes***

O *HotPotatoes* é uma ferramenta que permite a criação de exercícios interativos para a WWW, sendo compatível com todos os navegadores e possuindo versões para *Windows*, *Linux* e *Mac*. Ele é composto por cinco módulos de criação e um módulo de compilação: (1) *JCross*, para a elaboração de palavras cruzadas; (2) *JMix*, para o desenvolvimento de formulários para a ordenação de palavras de uma frase; (3) *JCloze*, para a criação de exercícios de texto com lacunas em branco para preencher; (4) *JQuiz*, para a construção de questionários de escolha múltipla, de seleção múltipla, verdadeiro/falso, ou de resposta curta; (5) *JMatch*, para criar páginas de associação de pares ou ordenação de frases; e o módulo de compilação (6) *The Masher*, que agrupa os arquivos de diversos exercícios e cria uma página com os *links* para cada um dos exercícios (HOTPOTATOES, 2018).

Tendo o exercício criado no *HotPotatoes*, basta exportar os arquivos para um servidor WWW. Com as páginas postadas no servidor WWW, é necessário fazer algumas alterações, já que, por padrão, o *HotPotatoes* não grava as informações dos formulários em banco de dados.

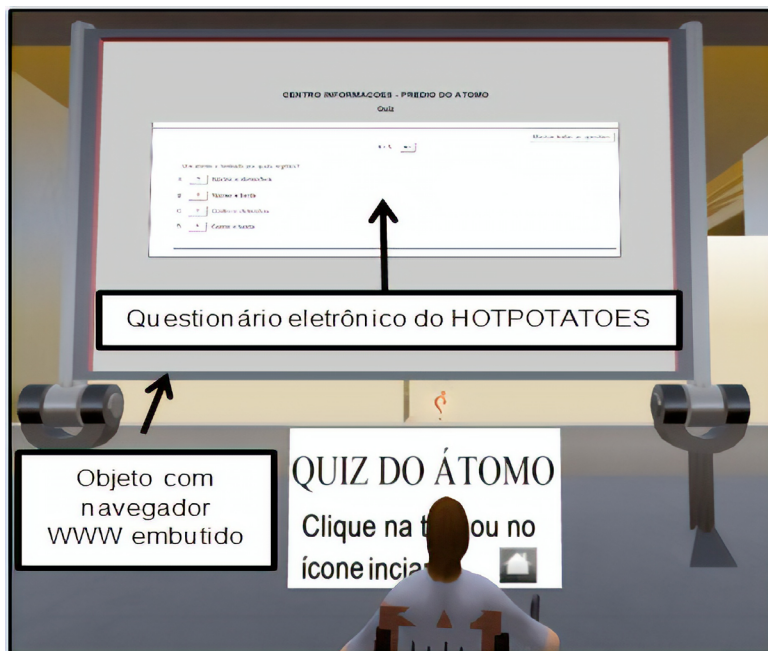
Alterando o arquivo do exercício gerado, por exemplo, “atomo.htm” ou “atomo.php”, é necessário armazenar as informações enviadas pelo formulário *HotPotatoes*: *realname* (nome do avatar), *Exercise* (exercício preenchido), *Score* (percentual de acertos), *Start\_Time* (hora de início do preenchimento), *End\_Time* (hora de fim do preenchimento), *Start\_Date* (data de início do preenchimento), *End\_Date* (data do fim do preenchimento) em variáveis PHP. Então, as variáveis PHP que contêm essas informações podem ser gravadas em um banco de dados previamente configurado. A Figura 6.2 mostra o comando SQL que armazena essas informações em um banco de dados.

Figura 6.2 - SQL para gravação dos dados do *HotPotatoes* em banco de dados.

```
//-----  
// insere dados tabelas HOTPOTATOES  
//-----  
$query = "INSERT INTO $tabelaHot (codigo, realname, Exercise, Score, Start_Time, End_Time, Start_Date, End_Date) VALUES  
(0, '$realname', '$Exercise', '$Score', '$nhora1', '$nhora2', '$ndata1', '$ndata2' );";
```

Como os formulários do *HotPotatoes* são disponibilizados na WWW, o MVI deve fornecer um meio para seu preenchimento. No *OpenSim*, criam-se objetos que possuem um navegador WWW embutido, o qual permite acesso à *Internet* (OPENSIM, 2018). Dessa forma, são acessados os exercícios do *HotPotatoes* com o auxílio de uma URL. A Figura 6.3 demonstra um objeto com acesso à WWW e um questionário do *HotPotatoes* sendo acessado.

Figura 6.3 - Acesso aos exercícios do *HotPotatoes* no MVI.



Ao terminar o preenchimento do exercício do *HotPotatoes*, o aluno é direcionado a uma página que mostra seu desempenho naquele exercício, e as informações correspondentes são gravadas na base de dados. A Figura 6.4 apresenta uma página PHP que realiza a consulta de um aluno específico na tabela de “hotpotatoes”.

Figura 6.4 - Consulta ao desempenho de um aluno no *HotPotatoes*.

CONSULTA DESEMPENHO NOS QUIZES - BUSCA POR ALUNO

Aluno:  Consultar

Nome do aluno	Exercício	Desempenho	Data inicio	Hora inicio	Data fim	Hora fim	Duração	Turma	Computador
aluno68 avatar68	QUIZ DO ATOMO - PREDIO DO ATOMO	100	2017-10-20	08:53:53	2017-10-20	08:56:00	00:02:07	1aEM	opensim68
aluno68 avatar68	CONDUTORES - TERRA	93	2017-10-20	09:24:51	2017-10-20	09:27:26	00:02:35	1aEM	opensim68
aluno68 avatar68	MODELOS ATOMICOS - PREDIO DO ATOMO	80	2017-10-20	09:11:24	2017-10-20	09:14:15	00:02:51	1aEM	opensim68
aluno68 avatar68	CENTRO INFORMACOES - PREDIO DO ATOMO	80	2017-10-20	08:34:58	2017-10-20	08:37:53	00:02:55	1aEM	opensim68
aluno68 avatar68	ISOLANTES E CONDUTORES	77	2017-10-20	09:17:22	2017-10-20	09:20:52	00:03:30	1aEM	opensim68
aluno68 avatar68	ELETRIZACAO POR CONTATO E POR INDUCAO	66	2017-10-20	09:34:00	2017-10-20	09:36:54	00:02:54	1aEM	opensim68
Total: 6	Média desempenho	82,67	-	-	-	-	-	-	-

### 6.2.1.3 Questionários eletrônicos com o SLOODLE

Uma alternativa para a criação de exercícios dentro do MVI é o SLOODLE. O SLOODLE é um projeto gratuito e de código aberto que integra os ambientes virtuais multiusuários do *Second Life* (LINDEN RESEARCH, 2018) e/ou *OpenSim* (OPENSIM, 2018) com o sistema de gerenciamento de aprendizado *Moodle* (MOODLE, 2018). O SLOODLE fornece uma variedade de ferramentas para apoiar o aprendizado e o ensino no Mundo Virtual Imersivo, ferramentas totalmente integradas a um sistema de gerenciamento de aprendizado baseado na *web*, tais como o *Presenter*, o *WebIntercom*, o *Toolbar*, o *Quiz Chair*, o *Pile On Quiz*, o *Prim Drop*, o *MetaGloss*, o *Choice*, o *Vending Machine* e o *Picture* (SLOODLE, 2018).

Depois de instalar o módulo SLOODLE no *Moodle* e habilitá-lo no *OpenSim*, é possível criar exercícios no *Moodle* e interligá-los com um objeto no Mundo Virtual. A Figura 6.5 exibe um objeto que possui conexão com um destes exercícios, enquanto, na Figura 6.6, é possível visualizar as opções de criação de exercícios para o SLOODLE no *Moodle*.

Figura 6.5 - Objeto que interliga MVI e Moodle.

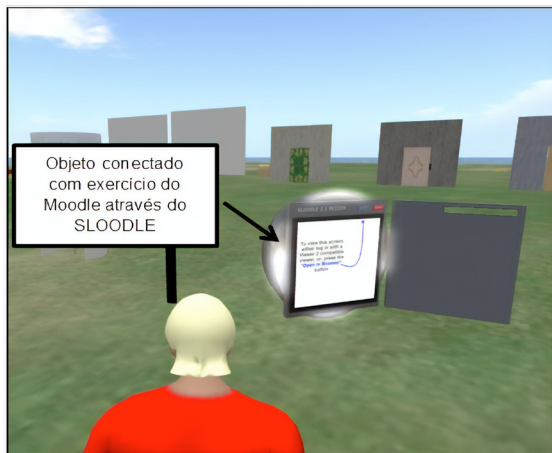
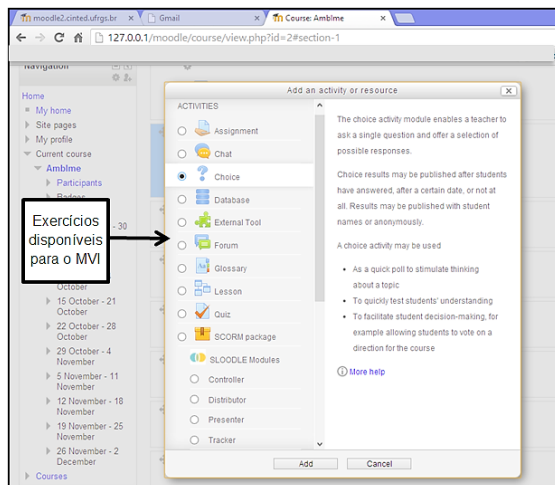


Figura 6.6 - Exercícios disponíveis no módulo SLOODLE do Moodle para o MVI.



Os dados dos exercícios disponibilizados pelo SLOODLE e preenchidos no MVI serão gravados em uma tabela específica na base de dados do Moodle. Para consultar essa tabela ou ter acesso aos relatórios do SLOODLE, pode ser necessário consultar o administrador do Moodle.

Uma vez que as informações da movimentação do avatar do aluno pelo MVI e os dados dos questionários eletrônicos estejam armazenados em um banco de dados, é possível analisá-los de forma a obter informações aplicáveis em novas estratégias educacionais.

## **6.3 ESTRATÉGIA PEDAGÓGICA ENVOLVENDO A APRENDIZAGEM EXPERIENCIAL**

David A. Kolb (1984) propôs a experiência como fonte de aprendizado e desenvolvimento. Seu trabalho expôs o princípio de que uma pessoa aprenderia pela descoberta e pela experiência. Sua teoria da aprendizagem tornou-se amplamente usada e é conhecida como o ciclo de aprendizagem experiencial de Kolb, no qual sugere que há quatro estágios de aprendizagem: a experiência concreta seguida pela reflexão sobre essa experiência em uma base pessoal.

Isso pode então ser seguido pela derivação de regras gerais que descrevem a experiência, ou a aplicação de teorias conhecidas a ela (Conceitualização Abstrata) e, portanto, a construção de modos de modificar a próxima ocorrência da experiência (Experimentação Ativa), levando de volta para a próxima Experiência Concreta. Tudo isso pode acontecer em um curto período de tempo ou durante dias, semanas ou meses.

### 6.3.1 Experiências em MVI

Os Mundos Virtuais são caracterizados pelo uso e navegação no espaço 3D e pela representação de seus usuários dentro dos MVIs pelos personagens conhecidos como avatares (CHILDS; PEACHEY, 2013). A presença do aprendiz em um Mundo Virtual Imersivo é operacionalizada com o avatar que o representa e realiza as ações possíveis no contexto delineado: corrida, caminhada, corrida ou até mesmo voar, interagir com objetos no Mundo Virtual por mera abordagem ou tocá-los intencionalmente para obter respostas e reações.

Uma pergunta feita foi se essas ações tomadas no Mundo Virtual Imersivo afetam o aluno. Elas serão capazes de motivar e engajar o usuário aprendiz, promovendo aprendizado ou impacto, conforme previsto pelas teorias da aprendizagem construtivista ou experiencial, mesmo como uma experiência vicária? Marsh, Yang e Shahabi (2005) estudaram a experiência vicária e empática em ambientes tridimensionais mediados interativos, e afirmam que a aprendizagem é um processo ativo, por meio do qual o aprendiz manipula o objeto de conhecimento, desenhando suas reações e respostas. Esse processo permite não apenas que ele se torne consciente do que é visível, aparente à primeira vista, mas também que descubra o comportamento oculto, quando certas ações são executadas no objeto. Em termos de promover a experiência com o objeto de conhecimento, Occhioni (2013) mostra um Mundo Virtual chamado Techland e a Ilha Mathland, em que o aluno pode tocar, mover, girar e selecionar os objetos para completar e organizar as for-



mas em experimentos de geometria. Cada experiência tem diferentes configurações e graus de interatividade entre alunos e objetos, exigindo mais conhecimento de um aluno.

Mundos Virtuais como o *OpenSim*, usado para suportar os laboratórios, permitem algum manuseio de objetos. A visualização 3D com *zoom*, por exemplo, é uma possibilidade básica nesse ambiente. O avatar pode se mover em torno do objeto, tornando-se mais ou menos próximo. Mas, quando se trata de possibilitar reações e respostas dos objetos, dependerá dos comportamentos previamente definidos para qualquer objeto e *script* relacionado ao objeto. Se um objeto não possuir um *script* que permita seu movimento, quando tocado, por exemplo, ele não se moverá! Esse é o trabalho intensivo exigido na construção de um laboratório virtual.

A partir do momento em que o avatar pode ser considerado uma extensão no ambiente virtual de seu criador no mundo real, é razoável aceitar que as situações pelas quais passa são sentidas de alguma forma por seu usuário. Assim, mesmo que não seja o mundo real de fato, o ambiente virtual pode contribuir para o desenvolvimento de habilidades que se aproximam do real por uma experiência substituta ou vicária, ou seja, não precisa ser submetido a um choque de alta tensão para imaginar suas consequências.

Por isso, é possível usar os Mundos Virtuais para expandir e qualificar alguns aspectos cognitivos dos alunos. Por exemplo, um experimento no laboratório virtual explora os conceitos básicos de circuitos elétricos que podem ajudar o aluno a desenvolver a percepção sobre os

efeitos da especificação incorreta de um componente elétrico. O aluno verá a lâmpada do circuito queimar, caso a corrente esteja acima de um certo limite (Figura 6.7).

Figura 6.7 - Exemplo de um experimento com circuitos elétricos.



Assim, os Mundos Virtuais podem preencher o vazio que antes era preenchido com filmes e outros artefatos audiovisuais no desenvolvimento de habilidades para as quais a imaginação é capaz de minimizar a ausência de uma prática efetiva.

Existem diferenças entre fazer no mundo real e no virtual, diferenças que podem ser consideradas eficientes e/ou efetivas ou não, dependendo dos objetivos a serem alcançados. Embora o ambiente virtual tenha menos realismo que o mundo real, por outro lado oferece bene-

fícios em termos de redução de custos e risco de acidentes. Além disso, é possível modificar o grau de liberdade do aluno em um experimento virtual (inclusive em tempo real), o que nem sempre é possível em um experimento físico.

Vale a pena, em vez de tentar ver o que é melhor, investigar quais são as peculiaridades de cada um e como eles podem se complementar. Muitas vezes, o esforço e o investimento em um experimento físico, para desenvolver certas habilidades, podem ser reduzidos oferecendo o mesmo experimento no ambiente virtual, aproximando-se da competência necessária.

## **6.4 MOTIVAÇÃO DOS ALUNOS EM MVI**

É observado no cotidiano escolar que muitas variáveis interferem no processo ensino-aprendizagem dos alunos, tais como: falta de perspectivas e metas a atingir, falta de orientação e acompanhamento por parte dos responsáveis, problemas emocionais e a falta de motivação.

Segundo Rapkiewicz (2007), é frequente a queixa dos professores com a desmotivação dos alunos, demonstrando em comportamentos que não condizem com o fazer pedagógico.

No ambiente virtual, existe uma predisposição à motivação, mas a curiosidade por si só não mantém os altos índices de motivação se essa não for estimulada. Sendo assim, no cenário educacional mediado por Mundos Virtuais, devem-se priorizar as estratégias de ensino, ações que busquem a interlocução entre professores, alunos e conteúdo/informações disponíveis no ambiente, o fortalecimento da percepção de

vínculo e o desenvolvimento de habilidades direcionadas ao monitoramento da aprendizagem, à gestão do tempo de estudo e à autonomia (BADIA; MONEREO, 2010; MAURI; ONRUBIA, 2010).

Na década de 1970, Edward Deci e Richard Ryan elaboraram a teoria da autodeterminação, com o objetivo de investigar os elementos constituintes da motivação intrínseca e da extrínseca e os fatores que determinavam sua promoção. Essa teoria compreende o ser humano como indivíduo ativo, que tende naturalmente ao crescimento saudável e à autorregulação (RYAN; DECI, 2017). Na teoria da autodeterminação, o envolvimento do indivíduo em atividades de aprendizagem busca atender a três necessidades psicológicas básicas e universais: a autonomia, a competência e o pertencer/estabelecer vínculos (BORUCHOVITCH *et al.*, 2010).

Tradicionalmente, a literatura científica categoriza a motivação em duas vertentes: a motivação intrínseca e a extrínseca. A motivação intrínseca é aquela inerentemente agradável ou interessante, corresponde a um desejo genuíno, a uma tendência inata do ser humano para explorar o mundo, a ação é vista como um fim em si mesma. Mas não se pode esquecer que, muitas vezes, mesmo com tantos recursos computacionais, essa motivação desejável se torna muito distante. Nesse caso, pode-se recorrer à motivação extrínseca, ou seja, a pessoa realiza algo para alcançar determinado resultado, a ação vem responder a pressões externas, ou porque é útil para atingir determinado objetivo, independentemente da ação em si (SGOBBI *et al.*, 2014).

Habitualmente, a educação vem trabalhando com a motivação extrínseca há anos: o aluno resolve uma expressão algébrica porque vai ter uma nota e/ou não quer ser punido. Esse processo é útil quando o índice de motivação é variável, de forma a aumentar a motivação controlada.

No Mundo Virtual, existe uma gama de vantagens e gatilhos motivacionais, segundo Sgobbi (2015):

- a) Possibilitar a criação de cenários desafiadores com a presença de sensores (dentro do Mundo Virtual) ou *tracking system* (um sistema de rastreamento é usado para a observação de pessoas ou objetos em movimento e o fornecimento de uma sequência ordenada e oportuna de dados de localização para processamento adicional), oferecendo aos professores informações acerca da experiência virtual de cada aluno, o que facilita a programação de novas intervenções.
- b) Oportunizar a utilização de agentes virtuais (NPCs) que ofereçam questões e direcionamentos intrigantes.
- c) Utilizar os *chatbots* que ofereçam uma conversação direcionada e tenham uma base de dados totalmente programável e flexível.
- d) Outra vantagem do Mundo Virtual é oportunizar novas vivências. Muitas escolas não possuem laboratórios para aulas práticas (de Física, Química, Matemática etc.) e essa ferramenta oportuniza ao aluno a possibilidade de tocar os objetos, criar e/ou participar de experimentos, observando e tendo *feedback* em casos positivos e negativos (afinal, o erro também é uma motivação para a aprendizagem).

e) Oferecer interconexão com *smartphones*, sensores de movimentação, entre outros, que oportunizam inúmeras outras vantagens. Além de lembrar ao aluno suas atividades, o Mundo Virtual pode enviar vídeos ao aluno, que pode ainda receber *links* de questionários interativos e muitas outras possibilidades.

De modo geral, acredita-se que os resultados alcançados sejam relevantes com o uso adequado dos recursos em Mundos Virtuais, pois ampliam o rol de informações necessárias à investigação de questões relacionadas às estratégias de ensino e de monitoramento de aprendizagem e à motivação dos estudantes para aprender.

## REFERÊNCIAS

BADIA, Antoni; MONEREO, Carles. Ensino e aprendizado de estratégias de aprendizagem em ambientes virtuais. *In*: BORUCHOVITCH, Evely; BZUNECK, José Aloyseo; GUIMARÃES, Sueli Edi Rufini (org.). **Motivação para aprender: aplicações no contexto educativo**. Petrópolis: Vozes, 2010. p. 43-70.

BORUCHOVITCH, Evely; BZUNECK, José Aloyseo; GUIMARÃES, Sueli Edi Rufini (org.). **Motivação para aprender: aplicações no contexto educativo**. Petrópolis: Vozes, 2010.

BRASIL. Ministério da Educação. Secretaria de Educação Fundamental. **Parâmetros curriculares nacionais: Ciências naturais**. Brasília, DF: MEC, 1998.

CHANG, Man Kit; LAW, Sally Pui Man. Factor structure for young's internet addiction test: a confirmatory study. **Computer in human behavior**, v. 24, n. 6, p. 2597-2619, 2008.

CHILDS, Mark; PEACHEY, Anna (ed.). **Understanding learning in virtual worlds**. 1st ed. London: Springer-Verlag London, 2013. 179 p.

CHIU, Jennifer L.; DEJAEGHER, Crystal J.; CHAO, Jie. The effects of augmented virtual science laboratories on middle school students' understanding of gas properties. **Computers & education**, v. 85, p. 59-73, 2015.

COLL, César; MONEREO, Carles (org.). **Psicologia da educação virtual: aprender e ensinar com as tecnologias da informação e da comunicação**. Porto Alegre: Artmed, 2010.

DREHER, Carl; REINERS, Torsten; DREHER, Naomi; DREHER, Heinz. 3D virtual worlds enriching innovation and collaboration in information systems research, development, and commercialisation. *In: IEEE INTERNATIONAL CONFERENCE ON DIGITAL ECOSYSTEMS AND TECHNOLOGIES*, 3., 2009, Istanbul, Turkey. **Proceedings** [...]. USA: Institute of Electrical and Electronics Engineers – IEEE, 2009. p. 168-173.

GIORDAN, Marcelo. **Computadores e linguagens nas aulas de ciências: uma perspectiva sociocultural para compreender a construção de significados**. Ijuí: Ed. Unijuí, 2008.

GUILLERMO, Oscar Eduardo Patrón. **Uso de laboratórios virtuais de aprendizagem em mecânica dos fluidos e hidráulica na engenharia**. 2016. 162 f. Tese (Doutorado) – Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.

HERPICH, Fabrício. **ELAI: intelligent agent adaptive to the level of expertise of students**. 2015. 196 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria, Santa Maria, 2015.

HERPICH, Fabrício; NUNES, Felipe Becker; CAZELLA, Silvio Cesar; TAROUCO, Liane Margarida Rockenbach. Mineração de dados educacionais: uma análise sobre o engajamento de usuários em mundos virtuais. *In: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – WCBIE*, 5., 2016, Uberlândia. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2016.

HERPICH, Fabrício; TAROUCO, Liane Margarida Rockenbach. Engajamento de usuários em mundos virtuais: uma análise teórico-prática. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 14, n. 1, 2016. Disponível em: <<http://seer.ufrgs.br/index.php/renote/index>>. Acesso em: 12 nov. 2018.

HOT POTATOES. **Hot Potatoes home page**. Half-Baked Software, Victoria, Canada, 2018. Disponível em: <<https://hotpot.uvic.ca/>>. Acesso em: 12 nov. 2018.

JACKA, Lisa. **Virtual worlds in pre-service teacher education: the introduction of virtual worlds in preservice teacher education to foster innovative teaching-learning processes.** 2015. 304 f. Tese (Doutorado) – Southern Cross University, Lismore, Australia, 2015.

JOHNSON, Constance M.; VORDERSTRASSE, Allison A.; SHAW, Ryan. Virtual worlds in health care higher education. **Journal of virtual worlds research**, v. 2, n. 2, p. 1-12, 2009.

KOLB, David A. **Experiential learning: experience as the source of learning and development.** Englewood Cliffs, NJ: Prentice Hall, 1984.

LINDEN RESEARCH. **Second Life.** Homepage. Linden Research, Boulder, CO, USA, 2018. Disponível em: <<https://secondlife.com/?lang=pt-BR>>. Acesso em: 12 nov. 2018.

MARATOU, Vicky; XENOS, Michalis. **Report on 3D virtual worlds platforms and technologies.** V-ALERT project 543224-LLP-1-2013-1-GR-KA3-KA3MP. Athens, Greece, 2014. Disponível em: <<https://www.enter-network.eu/project/v-alert/>>. Acesso em: 12 nov. 2018.

MARSH, Tim; YANG, Kiyoung; SHAHABI, Cyrus. Vicariously there: connected with and through our own and other characters. In: AISB'05 CONVENTION: SOCIAL INTELLIGENCE AND INTERACTION IN ANIMALS, ROBOTS AND AGENTS, 2005, Hertfordshire, UK. **Proceedings [...]**. London: Society for the Study of Artificial Intelligence and Simulation for Behaviour – AISB, 2005. p. 115-121.

MAURI, Teresa; ONRUBIA, Javier. O professor em ambientes virtuais: perfil, condições e competências. In: COLL, César; MONEREO, Carles (org.). **Psicologia da educação virtual: aprender e ensinar com as tecnologias da informação e da comunicação.** Porto Alegre: Artmed, 2010. p. 118-135.

MOODLE. **Homepage.** Open-source learning platform. 2019. Disponível em: <<https://moodle.org/>>. Acesso em: 4 ago. 2019.

MORAN, José Manuel; MASETTO, Marcos Tarciso; BEHRENS, Marilda Aparecida. **Novas tecnologias e mediação pedagógica.** 21. ed. Campinas: Papirus, 2013. 176 p.

MORGADO, Leonel. Os mundos virtuais e o ensino-aprendizagem de procedimentos. **Educação & cultura contemporânea**, v. 13, n. 6, p. 35-48, 2009.



NUNES, Felipe Becker; HERPICH, Fabrício; TAROUCO, Liane Margarida Rockenbach; LIMA, José Valdeni de. Monitoramento e avaliação de estudantes em mundos virtuais. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 14, n. 1, 2016.

NUNES, Felipe Becker; HERPICH, Fabrício; ZUNGUZE, Manuel Constantino; NICHELE, Aline Grunewald; ANTUNES, Fabiano Ferreira; TAROUCO, Liane Margarida Rockenbach; LIMA, José Valdeni de. A virtual world for the teaching and learning of natural sciences. *In: EDULEARN17 – ANNUAL INTERNATIONAL CONFERENCE ON EDUCATION AND NEW LEARNING TECHNOLOGIES*, 9., 2017, Barcelona, Spain. **Proceedings** [...]. Valencia, Spain: International Academy of Technology, Education and Development – IATED, 2017a. v. 1, p. 6-13.

NUNES, Felipe Becker; ZUNGUZE, Manuel Constantino; HERPICH, Fabrício; TAROUCO, Liane Margarida Rockenbach; LIMA, José Valdeni de; ANTUNES, Fabiano Ferreira; NICHELE, Aline Grunewald. Perceptions of pre-service teachers about a science lab developed in OpenSim. **International journal for innovation education and research**, v. 5, n. 5, p. 71-94, 2017b. Disponível em: <<http://www.ijer.net/index.php/ijer/article/view/675/573>>. Acesso em: 12 nov. 2018.

NUNES, Felipe Becker; ZUNGUZE, Manuel Constantino; HERPICH, Fabrício; VOSS, Gleizer Bierhalz; TAROUCO, Liane Margarida Rockenbach; LIMA, José Valdeni de. Teaching sciences in virtual worlds with mastery learning: a case of study in elementary school. **IJAERS – International journal of advanced engineering research and science**, v. 5, n. 7, p. 191-211, 2018.

OCCHIONI, Michelina. Techland, a virtual world for maths and science. *In: EUROPEAN IMMERSIVE EDUCATION SUMMIT*, 3., 2013, London, UK. **Proceedings** [...]. [S.l.]: Immersive Education Initiative, 2013. p. 94-99.

OLIVEIRA, Leander Cordeiro de; AMARAL, Marília Abrahão; BOTELHO, Sílvia Silva da Costa; ESPÍNDOLA, Danúbia Bueno; BARWALDT, Regina. Artefato metodológico de autoria aplicado aos mundos virtuais para educação. **Revista brasileira de informática na educação**, v. 24, n. 3, p. 97-108, 2016.

OPENSIM. **What is OpenSimulator?** USA, 2018. Disponível em: <[http://OpenSimulator.org/wiki/Main\\_Page](http://OpenSimulator.org/wiki/Main_Page)>. Acesso em: 12 nov. 2018.

PELET, Jean-Éric; LECAT, Benoît; PAPADOPOULOU, Panagiota. Enhancing learning and cooperation through digital virtual worlds. *In: INTERNATIONAL CONFERENCE ON EDUCATION, RESEARCH AND INNOVATION*, 2011,

Madrid, Spain. **Proceedings** [...]. Singapore: International Association of Computer Science and Information Technology – IACSIT, 2011. v. 18, p. 24-28.

PELLAS, Nikolaos. The development of a virtual learning platform for teaching concurrent programming languages in secondary education: the use of open sim and scratch4OS. **Journal of e-learning and knowledge society**, v. 10, n. 1, p. 1-15, 2014.

PIAGET, Jean. **The principles of genetic epistemology**. London: Routledge & Kegan Paul, 1972.

PILASTRI, André Luiz; BREGA, José Remo Ferreira. *Chatterbot* com interatividade ao avatar encapsulado no ambiente virtual Second Life usando a base de conhecimento em AIML. *In: WORKSHOP DE REALIDADE VIRTUAL E AUMENTADA*, 6., 2009, Santos. **Anais** [...]. Santos: UNISANTA, 2009.

PINHEIRO, André; FERNANDES, Paulo; MAIA, Ana; CRUZ, Gonçalo; PEDROSA, Daniela; FONSECA, Benjamim; RAFAEL, Jorge. Development of a mechanical maintenance training simulator in OpenSimulator for F-16 aircraft engines. **Procedia computer science**, v. 15, p. 248-255, 2012.

POTKONJAK, Veljko; GARDNER, Michael; CALLAGHAN, Victor; MATTILA, Pasi; GUETL, Christian; PETROVIĆ, Vladimir M.; JOVANOVIĆ, Kosta. Virtual laboratories for education in science, technology, and engineering: a review. **Computers & education**, v. 95, p. 309-327, 2016.

RAPKIEWICZ, Clevi Elena; FALKEMBACH, Gilse; SEIXAS, Louise; ROSA, Núbia dos Santos; CUNHA, Vanildes Vieira da; KLEMANN, Miriam. Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 4, n. 2, 2006.

RYAN, Richard M.; DECI, Edward L. **Self-determination theory: basic psychological needs in motivation, development, and wellness**. New York: Guilford, 2017.

SAINT-GEORGES, Ingrid de; FILLIETTAZ, Laurent. Situated trajectories of learning in vocational training interactions. **European journal of psychology of education**, v. 23, n. 2, p. 213-233, 2008.

SGOBBI, Fabiana Santiago; NUNES, Felipe Becker; TAROUÇO, Liane Margarida Rockenbach. A utilização de agentes inteligentes no apoio ao autocuidado de idosos. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 12, 2014.

SGOBBI, Fabiana Santiago; TAROUCO, Liane Margarida Rockenbach; MÜHLBEIER, Andreia Rosangela Kessler. Virtual agents' support for practical laboratory activities. *In: EUROPEAN IMMERSIVE EDUCATION SUMMIT*, 5., 2015, Paris, France. **Proceedings** [...]. [S.l.]: Immersive Education Initiative, 2015. v. 1, p. 239-249.

SILVA, Patrícia Fernanda da. **O uso das tecnologias digitais com crianças de 7 meses a 7 anos: como as crianças estão se apropriando das tecnologias digitais na primeira infância?** 2017. 232 f. Tese (Doutorado em Informática na Educação) – Centro de Estudos Interdisciplinares em Novas Tecnologias na Educação, Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

SIMSEK, Irfan; CAN, Tuncer. The design and use of educational games in 3D virtual worlds. *In: INTERNATIONAL CONFERENCE SOCIETY FOR INFORMATION TECHNOLOGY AND TEACHER EDUCATION – SITE*, 27., 2016, Savannah, GA, USA. **Proceedings** [...]. Waynesville, NC: The Association for the Advancement of Computing in Education – AACE, 2016. p. 611-617.

SLOODLE. **SLOODLE: Simulation-Linked Object-Oriented Dynamic Learning Environment**. 2018. Disponível em: <<https://www.sloodle.org/>>. Acesso em: 12 nov. 2018.

SMITH-ROBBINS, Sarah. Are virtual worlds (still) relevant in education? **eLearn Magazine**, ACM, New York, Dec. 2011. Disponível em: <<https://elearnmag.acm.org/featured.cfm?aid=2078479>>. Acesso em: 11 ago. 2019.

TERZIDOU, Theodouli; TSIATSOS, Thrasyvoulos. The impact of pedagogical agents in 3D collaborative serious games. *In: IEEE GLOBAL ENGINEERING EDUCATION CONFERENCE – EDUCON*, 2014, Istanbul, Turkey. **Proceedings** [...]. USA: Institute of Electrical and Electronics Engineers – IEEE, 2014. p. 1175-1182.

TIBOLA, Leandro Rosniak. **Fatores ensejadores de engajamento em ambientes de mundos virtuais**. 2018. 235 f. Tese (Doutorado em Informática na Educação) – Centro de Estudos Interdisciplinares em Novas Tecnologias, Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2018.

TIBOLA, Leandro Rosniak; TAROUCO, Liane Margarida Rockenbach. Rastreamento de interações em laboratórios educacionais nos Mundos Virtuais 3D para identificação de engajamento. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 13, n. 2, 2015. Disponível em: <<https://seer.ufrgs.br/renote/article/view/61440/36327>>. Acesso em: 12 nov. 2018.

TÜZÜN, Hakan; ÖZDİNÇ, Fatih. The effects of 3D multi-user virtual environments on freshmen university students' conceptual and spatial learning and presence in departmental orientation. **Computers & education**, v. 94, p. 228-240, 2016.

VOSS, Gleizer Bierhalz; NUNES, Felipe Becker; HERPICH, Fabrício; MEDINA, Roseclea Duarte. Ambientes virtuais de aprendizagem e ambientes imersivos: um estudo de caso utilizando tecnologias de computação móvel. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE, 24., 2013, Campinas. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2013.

YILMAZ, Rabia M.; BAYDAS, Ozlem; KARAKUS, Turkan; GOKTAS, Yuksel. An examination of interactions in a three-dimensional virtual world. **Computers & education**, v. 88, p. 256-267, 2015.

YOUNG, Jeffrey R. After frustrations in Second Life, colleges look to new virtual worlds. **The chronicle of higher education**, Washington, D.C., Febr. 14, 2010. Disponível em: <<https://www.chronicle.com/article/After-Frustrations-in-Second/64137>>. Acesso em: 11 ago. 2019.

ZIMMER, Josete Maria; VEZZANI, Marco Antônio. Second Life para educação à distância: uma experiência entre estudantes brasileiros e portugueses. *In*: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – WCBIE, 6., 2017, Recife. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2017.

# 7

## Tornando os elementos do MVI interativos

**Bárbara Gorziza Avila - IFRGS**  
bapadoin@gmail.com

**Érico Amaral - UNIPAMPA**  
ericoamaral@unipampa.edu.br

**Fabiana Santiago Sgobbi - PPGIE - UFRGS**  
fabianasgobbi@gmail.com

**Fabrcio Herpich - PPGIE - UFRGS**  
fabricio\_herpich@hotmail.com

**Liane Margarida Rockenbach Tarouco**  
PPGIE - UFRGS - liane@penta.ufrgs.br

**Leandro Rosniak Tibola - PPGIE - UFRGS**  
lrtibola@gmail.com

## 7.1 FERRAMENTAS DE AUTORIA

Quando associados aos objetos colocados no MVI, é o desenvolvimento da programação dos códigos que vai permitir torná-los mais interativos. À medida que os usuários interagem (tocando ou meramente se aproximando de objetos), reagem exibindo os comportamentos previstos em seu *script*. Portanto, o desenvolvimento dessa programação é parte essencial da implantação de um MVI. Essa programação é feita usando a linguagem que foi desenvolvida pela Linden Lab para o ambiente *Second Life* (com pequenas alterações) e, também, complementada pela linguagem de *script* criada pelos desenvolvedores do ambiente *OpenSimulator* e que foi denominada OSSL (*OpenSim Scripting Language*). Ambas são descritas na seção seguinte.

### 7.1.1 Linguagem para a construção dos *scripts*

Os *scripts* podem ser inseridos em qualquer objeto 3D criado no MVI. Nota-se que tanto o avatar quanto o NPC (*Non-Player Character*) são um tipo especial de integrantes dos MVIs. Assim, os NPCs não recebem programação *script* diretamente em sua estrutura, mas são criados e controlados por um objeto com esse fim específico. Os avatares, por sua vez, não aceitam programação *script* diretamente em seu corpo digital e são criados pelo servidor de Mundos Virtuais, tal como o *OpenSim* e o *Second Life*. A customização do avatar e a criação de NPCs são tratadas em mais detalhes no Capítulo 2, “As ferramentas de autoria para criar um Mundo Virtual Imersivo”. Este Capítulo apresenta a programação *script*

direcionada para a manipulação dos objetos primitivos (são as formas geométricas básicas, tal como o cubo e a esfera, que permitem a construção de objetos maiores e mais complexos, como móveis e prédios).

Os Mundos Virtuais são ambientes dinâmicos, que possibilitam a interação entre avatar e objetos e permitem a comunicação entre ambos. A dinâmica, a interação e a comunicação obtidas nos MVIs são possíveis pela *Linden Scripting Language* ou LSL, linguagem originalmente desenvolvida para ser usada nos componentes do *Second Life*, mas também suportada no *OpenSim*. Em virtude da concepção e das características particulares do *OpenSim*, foram desenvolvidas funções que não existiam na LSL original. Assim, foi implementada a linguagem *OpenSimulator Scripting Language*, ou OSSL, objetivando, principalmente, a comunicação HTTP, interação com avatares, manipulação de NPCs, objetos, texturas, *notecards* e recursos do ambiente virtual.

Na LSL, a maioria dos *scripts* fica inativa até receber alguma entrada (como o recebimento de uma mensagem em um canal de comunicação) ou detectar alguma alteração no ambiente (o toque no objeto, a proximidade de um avatar). Sendo assim, o *script* está em algum estado e reagirá a eventos ou entradas de acordo com os objetivos definidos pelo programador. No entanto, um *script* também pode conter dois ou mais estados diferentes e reagir de maneira diferente aos eventos ou entradas, dependendo do estado em que se encontra.

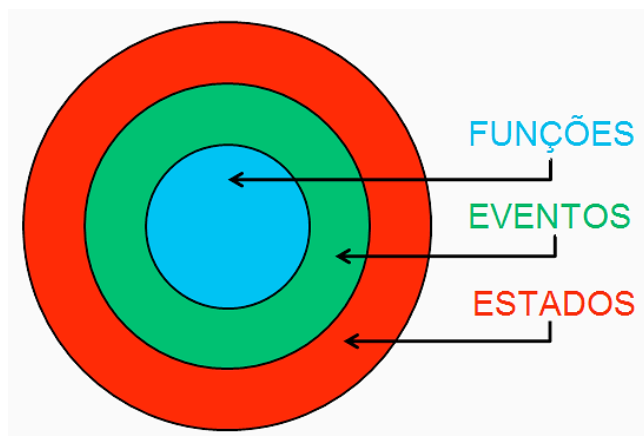
Um exemplo simples de mudança de estado é o de uma lâmpada que está no estado desligada e ignora todas as entradas, exceto a ação de ser tocada. Uma vez tocada, ela vai para o estado ligada, no qual ela aguarda a ação de ser tocada para voltar ao estado desligada.

Na LSL, um estado é uma seção especificada de código dentro da qual todos os eventos são especificados. O estado principal, exigido por todos os *scripts* LSL, é chamado de *default* (padrão). Todos os *scripts* devem ter um estado *default*, e cada estado deve ter pelo menos um evento.

Por sua vez, os eventos são formados por funções. As funções são comandos predefinidos que permitem que o usuário as combine com o seu código, para alterar os valores dos parâmetros e das variáveis do objeto que contém o próprio *script* ou de outro objeto presente no Mundo Virtual.

Assim, os *scripts* LSL são compostos de um ou mais estados. Dentro de cada estado, pode haver um ou vários eventos. Cada evento é formado por uma ou muitas funções (OPENSIM, 2012; LINDEN RESEARCH, 2015; LSL WIKI, 2012). A estrutura hierárquica da programação *script* LSL é apresentada na Figura 7.1.

Figura 7.1 - Estrutura de codificação de um *script* LSL.



Fonte: Adaptado de Guia de iniciación Linden Scripting Language (2018).

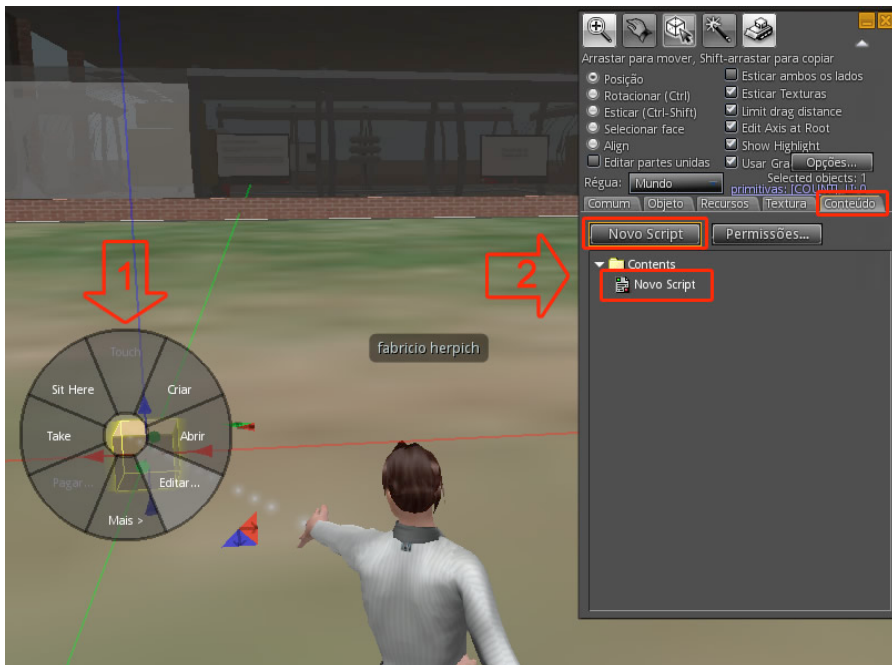


Porém, antes de escrever qualquer trecho de um *script* LSL, é necessário criar um objeto no qual possamos inserir o código. Assim, para criar um objeto no Mundo Virtual *OpenSim*, podem ser seguidos os passos:

- clicar com o botão direito do *mouse* próximo ao terreno;
- no menu circular, clicar em “*Create*”;
- clicar com o botão direito do *mouse* no objeto criado (indicado pelo número 1);
- no menu circular, clicar em *Editar* (ou “*Edit*”), na aba *Conteúdo* (ou “*Content*”) e, por fim, em *Novo Script* (ou “*New Script*”).

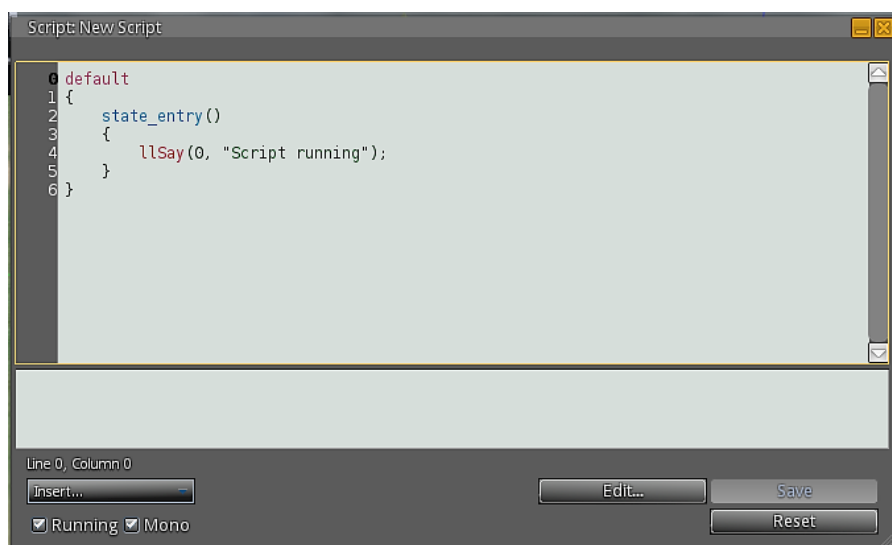
A Figura 7.2 mostra a imagem do resultado dos procedimentos acima.

Figura 7.2 - Criação de um novo *script* em um objeto



Após esses procedimentos, é possível construir a programação LSL para aquele objeto. O *script* LSL pode ser ou não desenvolvido no editor original do *OpenSim*, mas, se forem usados editores externos ao Mundo Virtual, tal como o LSL Editor ou o Notepad++, o código deve ser copiado para o editor do próprio objeto no Mundo Virtual. O editor nativo do *OpenSim* é visto na Figura 7.3.

Figura 7.3 - Editor de *script* LSL original do *OpenSim*.



### 7.1.1.1 Linden Scripting Language - LSL

A estrutura básica de um *script* LSL é composta pelo estado *default* e pelo evento *state\_entry()*, como mostra o Código 1.

### Código 1: Script LSL básico.

```
// o estado default é o estado padrão do LSL
default
{
// o "evento" state_entry é acionado em qualquer
// transição do estado default e na inicialização do objeto
state_entry()
{
// a função llsay escreve "Alô, avatar!" no MV
llsay(0, "Alô, avatar!");
}
}
```

Um dos eventos mais comuns no LSL é o `touch_start()`, o qual é acionado quando o avatar “inicia” o clique em um objeto. Esse evento possui a estrutura:

- `touch_start( integer total_number ) { ; }`

em que o `total_number` é o número detectado de avatares que tocam o objeto. O Código 2 mostra um exemplo do `touch_start()`:

### Código 2: Evento `touch_start()`.

```
touch_start(integer total_number)
{
// executa ações
}
```

Uma função usada para o envio de mensagens do objeto a outros objetos e/ou para o Mundo Virtual é a `llsay()`. Sua sintaxe é: `llsay (integer channel, string msg)`, em que *channel* é um dos canais de comunicação

privada, numerados de -2.147.483.648 até 2.147.483.647, enquanto as mensagens direcionadas ao canal 0 (zero) são mostradas para todo o Mundo Virtual. O parâmetro msg é a mensagem enviada. Um exemplo dessa função é apresentado no Código 3.

Código 3: Função `lSay()`.

```
touch_start(integer total_number)
{
    // mostra a mensagem "O objeto foi tocado!"
    // no canal de comunicação zero
    lSay(0, "O objeto foi tocado!");
}
```

Em ANEXOS, são descritos os comandos LSL necessários a uma programação básica, os quais podem ser utilizados e adaptados ao desenvolvimento de atividades educacionais em MVI.

### **7.1.2 OpenSimulator Scripting Language - OSSL**

Antes de programar um *script* OSSL, é necessário verificar se essas funções estão habilitadas no *OpenSim*. É necessário editar o arquivo *OpenSim.ini*, situado na pasta em que o *OpenSim* foi instalado no seu computador, algo do tipo "C:\OpenSim-0.9.0.1\bin" (como pode ser visto na Figura 2.2), e identificar se a opção *AllowOSFunctions* está definida como *true* e a função *OSFunctionThreatLevel* está assinalada como *High*.

Alguns comandos OSSL necessitam de parâmetros específicos para a opção *OSFunctionThreatLevel*. Eles podem ser verificados na página oficial do *OpenSim* (2012).

A sintaxe do *osNpcCreate* é: *key osNpcCreate (string firstname, string lastname, vector position, string cloneFrom)*.

Inserido em um objeto, o *script* com o comando *osNpcCreate* cria um NPC com o nome e sobrenome, em uma posição determinada, a partir de um modelo de aparência. O Código 4 apresenta a criação de um NPC. Esse código usa variáveis que são descritas no ANEXO 1.

**Código 4: Criação, movimentação e remoção do NPC com OSSL.**

```
key npc;
vector toucherPos;
default
{

    state_entry()
    {
        IISay(0, "Clique neste objeto para criar, mover e apagar um NPC!");
    }
    touch_start(integer number)
    {
        // recupera a posição do avatar que tocou o objeto
        // e incrementa 1 metro na coordenada X para esta posição
        vector npcPos = IIGetPos() + <1,0,0>;
        // captura a aparência do avatar e a guarda em "forma-roupas"
        osAgentSaveAppearance(IIDetectedKey(0), "forma-roupas");
        // cria o NPC chamado "NPC Clone", na posição registrada em npcPos
        // e com a aparência armazenada em "forma-roupas"
        npc = osNpcCreate("NPC", "Clone", npcPos, "forma-roupas");
        // guarda a posição do avatar que tocou no objeto
        toucherPos = IIDetectedPos(0);
    }
}
```

```

// transfere a execução do script para o estado existeNPC
state existeNPC;
}
} // fim default

state existeNPC
{
state_entry()
{
// move o NPC a 3 metros do avatar na coordenada X
osNPCMoveTo(npc, toucherPos + <3,0,0>);
// o NPC emite uma mensagem para todo o mundo virtual
osNPCSay(npc, "Olá!!! Meu nome é " + llKey2Name(npc));
}
touch_start(integer number)
{
// o NPC emite uma mensagem para todo o mundo virtual
osNPCSay(npc, "Tchau!!");
// remove o NPC
osNPCRemove(npc);
// torna a variável do NPC nula
npc = NULL_KEY;
// retorna para o estado default
state default;
}
} // fim do state existeNPC

```

## 7.2 FERRAMENTAS DE AUTORIA PARA CRIAR O SCRIPT

Embora a criação de *scripts* possa ser realizada usando apenas um editor de textos básico, tal como o Bloco de Notas, recursos adicionais de apoio à formatação, tais como os existentes no editor LSL Editor, podem facilitar o trabalho do desenvolvedor.

O *LSL Editor Community Edition* para *Windows* é um editor, compilador e depurador de *scripts* LSL autônomo. Seu compilador e depurador já estão razoavelmente precisos.

O desenvolvimento do LSL Editor foi feito originalmente por Alphons van der Heijden, autor do LSL Editor. Detalhe: executável e fonte podem ser encontrados em <<http://sourceforge.net/projects/lsleditor/>>.

O LSL Editor, *software* que ajuda os usuários a escrever *scripts* mais facilmente para objetos do *Second Life* e do *OpenSim*, foi atualizado com mais suporte para as funções do *OpenSim*. Com a nova versão 2.56, os usuários também podem criar *scripts* que executam outros *scripts* e a funcionalidade de teste e depuração é muito melhor que a do editor LSL integrado nos visualizadores do *OpenSim*. (KARIUKI, 2018).

A nova versão 2.56.0 tem mais funcionalidades de teste e depuração. Na verdade, pode-se testar seus *scripts* (ou conjuntos de *scripts* e sua interação) em uma extensão muito grande fora da grade (servidores de MVI em *grid*); o suporte de depuração, também, é muito melhor que o editor de *scripts in-world*.

As grades do *OpenSim* podem executar os comandos LSL padrão, mas também possuem funcionalidade adicional por meio da *OpenSimulator Scripting Language* (OSSL). Por exemplo, comandos específicos do *OpenSim* podem forçar o teletransporte, o que é útil para construir porções de teleporte ou armas que imediatamente mandam alguém para um inferno virtual quando são atingidos. Também há vários comandos para criar e gerenciar caracteres não executáveis (NPCs) e para colocar texto e gráficos em um *prim*. As versões anteriores do LSL Editor suportavam apenas funções LSL, o que significa que não se poderia usar o editor com as funções OSSL. Com isso, agora os *scripts* com funções OSSL irão compilar e podem ser testados e depurados.

O LSL Editor, atualizado, também permite aos usuários adicionar novos objetos e itens e executar um verificador de sintaxe, bem como personalizar fontes e cores. Ele ainda permite realçar, localizar e substituir funções, saltar para um número de linha específico, bem como formatar somente a área selecionada ou o documento inteiro.

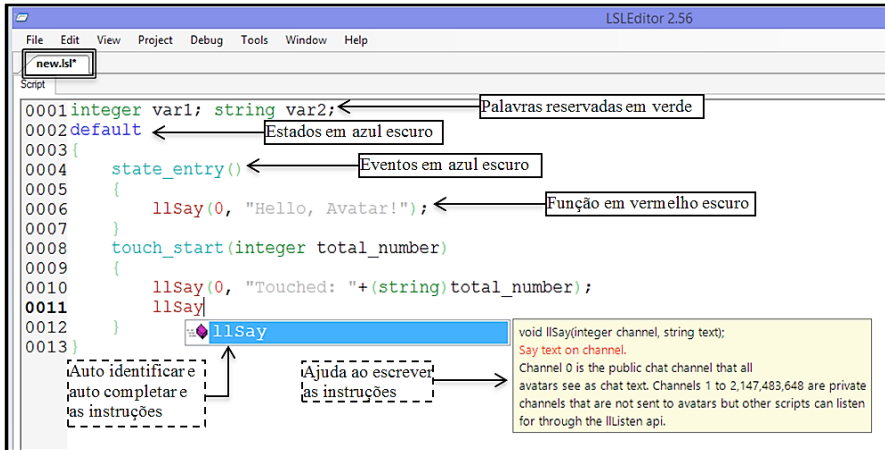
Ele funciona para plataformas *Windows*, incluindo o *Windows 2K*, 2003, XP, Vista, 7 e 8 e está disponível para *download* na página *Source Forge*, enquanto o manual do LSL Editor está disponível na página do *Google Doc* (KARIUKI, 2018).

Dentre as facilidades apresentadas pelo LSL Editor, está a identificação do código *script* por meio de um conjunto de cores, que identificam as palavras reservadas em verde (os declaradores de variáveis *integer* e *string*), estados em azul-escuro (*default*), eventos em azul-claro



(touch\_start() ) e as funções em vermelho-escuro ( llSay() ). A Figura 7.4 apresenta esse esquema de cores para o código LSL nas caixas com borda simples.

Figura 7.4 - Tela do editor LSL Editor.



Essa disposição de cores é útil quando se digita o código ou se procura por um erro de escrita, pois, como a LSL é sensível às letras maiúsculas e minúsculas, a função llSay() é válida, enquanto a função llsay() é inválida e gera um erro durante a compilação.

Outros recursos que auxiliam os iniciantes na programação LSL é o autoidentificar e o autocompletar o código. Assim, enquanto o programador digita as instruções, o LSL Editor verifica a sintaxe e sugere os estados, eventos e funções que se aproximam do texto digitado. Ainda, ao digitar a instrução LSL, o usuário pode verificar a sua sintaxe e descrição na forma de um pequeno quadrado de ajuda. A Figura 7.4 mostra estes recursos nas caixas com a borda pontilhada.

Uma observação pertinente a ser feita é sobre o salvamento do *script*, já que, por padrão, o LSL Editor cria um novo arquivo com o nome de “new.lsl”. Se o usuário não alterar o nome do seu programa, ele será salvo com o mesmo nome, podendo sobrescrever programas escritos anteriormente. Então, é aconselhável que, ao escrever um *script*, ele seja salvo com um nome que indica sua função e seja fácil de lembrar em um momento posterior. Outro detalhe que pode colaborar para manter os códigos sempre salvos é observar o asterisco ( \* ) ao lado do nome do arquivo; ele indica que o *script* foi modificado e ainda não foi salvo. Para manter os *scripts* sempre atualizados e evitar perda de códigos, deve-se clicar em “File - Save as”, digitar o nome e escolher uma pasta para guardar o arquivo. O nome do arquivo-padrão “new.lsl” e o asterisco indicador de alteração podem ser visualizados, no canto superior direito da Figura 7.4, na caixa com a borda dupla.

Adicionalmente, ferramentas que permitam especificar as funções a serem implementadas de um modo mais visual, usando a estratégia de blocos visuais de programação, proposta pelo MIT para o ambiente de autoria Scratch, por exemplo, podem facilitar bastante a criação de *scripts*.

### **7.2.1 ScripTastic (versão *light* e simplificada para gerar LSL)**

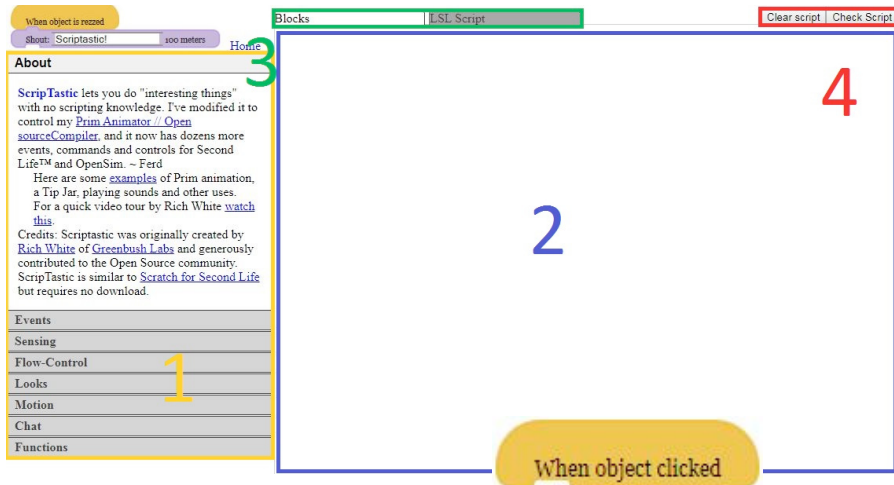
O *ScripTastic* é uma ferramenta básica de conversão de blocos visuais para LSL. O *ScripTastic* consiste em uma versão *light* e simplificada, possuindo menos funcionalidades em relação às outras ferramentas existentes, mas traz uma interface mais simples e intuitiva ao usuário

que está se iniciando no desenvolvimento de *scripts* para Mundos Virtuais. Para acessar a interface *web* do *ScriptTastic*, o usuário pode abrir em seu navegador o endereço: <https://www.outworldz.com/scriptastic/>.

### 7.2.1.1 Interface básica

Conforme a numeração e a coloração mostradas na Figura 7.5, os elementos básicos da interface do *ScriptTastic* são:

Figura 7.5 - Interface inicial do *ScriptTastic*.



1) À exceção da primeira aba, “*About*”, que contém uma breve descrição do *ScriptTastic*, as demais dividem os blocos existentes em categorias, para sua melhor organização, de acordo com sua funcionalidade.

2) É a área de trabalho, para onde serão arrastados os blocos (no caso da visão de blocos, vide item 3) e onde será exibido o *script* LSL resultante (visão LSL).

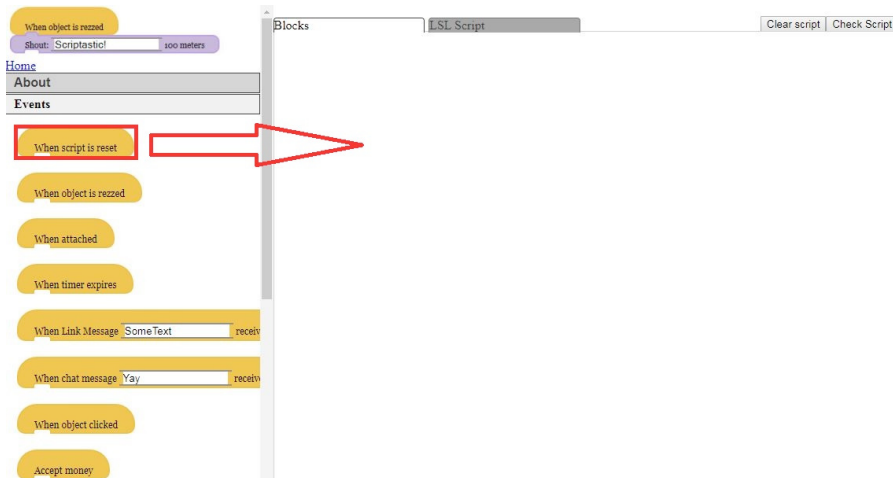
3) A primeira aba, “*Blocks*”, corresponde à visão de blocos e a segunda, “*LSL Script*”, à visão LSL, que conterà o código do *script* gerado a partir dos blocos.

4) O botão “*Clear Script*” limpa a área de trabalho, apagando tanto os blocos quanto o *script*. Já o botão “*Check Script*” varre o *script* em busca de erros e exibe os erros encontrados na própria área de trabalho, com a visão “*LSL Script*” selecionada.

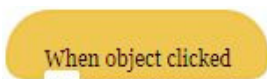
### **7.2.1.2 Blocos de programação**

Os blocos, ao serem arrastados para a área de trabalho a partir do menu lateral esquerdo, imediatamente começam a fazer parte do programa, alterando o *script* resultante, conforme a Figura 7.6.

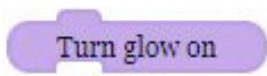
Figura 7.6 - Arrastando bloco para a área de trabalho.



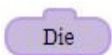
Os blocos se encaixam formando estruturas mais complexas e são executados de forma sequencial, de cima para baixo.



Exemplo de bloco que pode iniciar um *script*. Notam-se a falta de encaixes na parte superior e a presença do encaixe na borda inferior.



Exemplo de bloco intermediário, que deve ser encaixado em algum outro bloco superior.



Exemplo de bloco que só pode ser encaixado no final de um *script*.

### 7.2.1.3 Categorias dos blocos

Todos os blocos disponíveis no *Scriptastic* estão presentes em uma das categorias visíveis na Figura 7.7, devidamente distribuídos de acordo com sua funcionalidade. Os blocos de cada categoria também possuem uma coloração específica.

Figura 7.7 - Categorias dos blocos.


Events
Sensing
Flow-Control
Looks
Motion
Chat
Functions

**EVENTS:** Blocos que definem comportamento quando algum evento for disparado.

**Exemplo:** quando objeto for clicado.

**SENSING:** Blocos relacionados a sensores do ambiente e do *chat*.

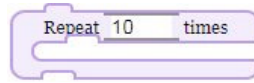
**Exemplo:** escutar mensagens no *chat*.



Listen for chat from anyone

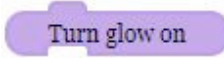
**FLOW-CONTROL:** Blocos com as funções básicas de controle de fluxo: condicionais e repetidores.

**Exemplo:** repetir ações 10 vezes.



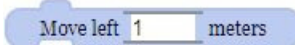
**LOOKS:** Blocos relacionados à aparência do objeto.

**Exemplo:** ligar o brilho.



**MOTION:** Blocos com funcionalidade de movimentação.

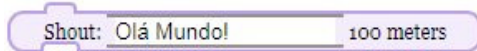
**Exemplo:** mover 1 metro para a esquerda.



**CHAT:** Blocos que enviam mensagens no *chat* do ambiente virtual ou para outros *scripts*.

**Exemplo:** mandar mensagem

“Olá, Mundo!” para todos em um raio de 100 metros.



**FUNCTIONS:** Blocos com funções relacionadas não tão diretamente ao Mundo Virtual, mas que são essenciais dentro da programação dos *scripts*.

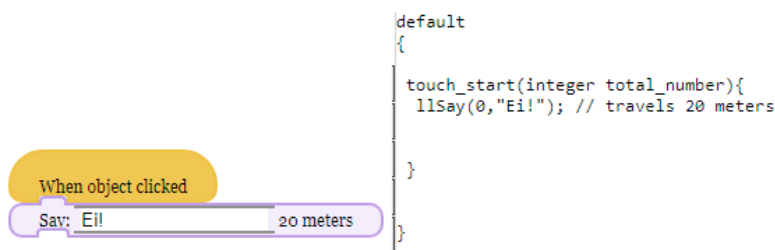
**Exemplo:** criação de uma variável inteira com valor inicial de 1.



## 7.2.1.4 Exemplos de aplicações

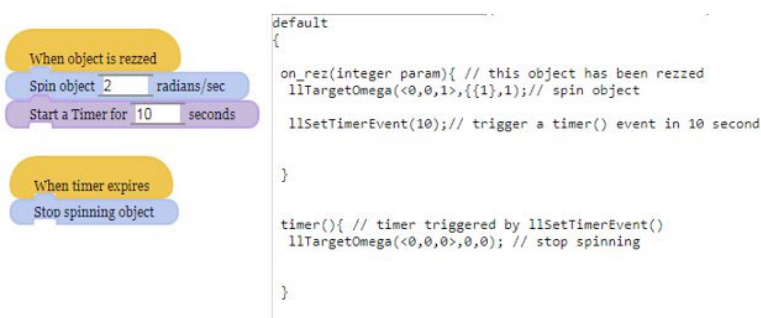
A aplicação abaixo (Figura 7.8), representada à esquerda pelo seu formato em blocos e na direita pelo código LSL gerado, usando o *Scriptastic*, define que, quando o objeto ligado a esse *script* for clicado, enviará a mensagem “Ei!”, no *chat*, para todos num raio de 20 metros.

Figura 7.8 - Codificando um *script* em blocos visuais e sua saída na linguagem LSL.



Já o próximo exemplo (Figura 7.9) faz com que o objeto comece a girar, assim que for criado, a uma velocidade de 2 rad/s. Passados 10 segundos, o objeto para de girar.

Figura 7.9 - Demonstração de *scripts* desenvolvidos com blocos visuais.





Nesse último exemplo (Figura 7.10), o objeto dobra de tamanho sempre que for clicado e é reduzido pela metade quando detecta alguma mensagem do seu dono no *chat*.

Figura 7.10 - Exemplos de aplicação dos blocos visuais para desenvolvimento de MVI.

```
default
{
    touch_start(integer total_number){
        llSetScale(<2.0, 2.0, 2.0>); // grow

        llListen(0, "", llGetOwner(), ""); // Listen for chat from owner
    }
    When object clicked
    Grow x2
    Listen for chat from owner
    sensor(integer num_detected){
        llSetScale(<0.5, 0.5, 0.5>); // shrink
    }
    When something is sensed
    Shrink x2
}
```

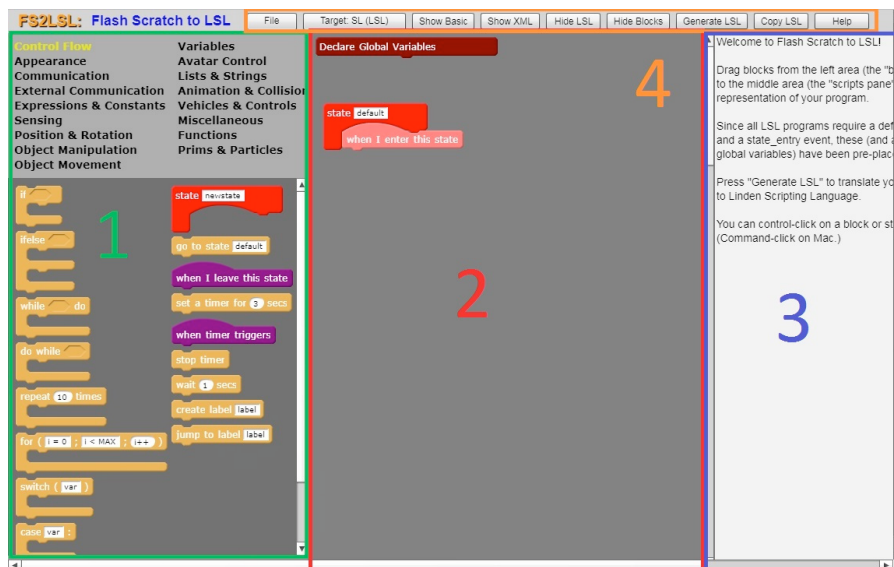
### 7.2.2 Ferramenta FS2LSL (Universidade de Denver)

Disponível em: <<http://inworks.ucdenver.edu/jkb/fs2lsl/>>

O FS2LSL é inspirado diretamente na aplicação de programação em blocos do MIT, o Scratch, utilizando blocos com *design* semelhante. Para usá-lo, é necessário o usar o *plugin Adobe Flash*.

## 7.2.2.1 Interface básica

Figura 7.11 - Interface inicial do FS2LSL.



Conforme a numeração e coloração vistas na Figura 7.11, os elementos básicos da interface do FS2LSL são:

1. Menu com os todos os blocos disponíveis à aplicação, divididos em categorias.
2. Área de trabalho para onde os blocos são arrastados e onde eles são subsequentemente manipulados. Alguns blocos são criados por padrão em um projeto novo.
3. Aqui é exibido o *script* resultante dos blocos presentes na área de trabalho.

#### 4. Barra com diversos botões e comandos:

4.1. FILE abriga as funções de controle de projeto: criar, salvar, carregar.

4.2. TARGET permite a troca entre ambiente *Second Life* e *OpenSim*.

4.3. SHOW(...) alterna entre BASIC ONLY (só serão mostrados blocos simples) e ADVANCED (todos os blocos ficam disponíveis).

4.4. SHOW/HIDE XML alterna a exibição do arquivo XML gerado, além do LSL.

4.5. SHOW/HIDE LSL alterna a exibição do *script* LSL na interface.

4.6. SHOW/HIDE BLOCKS alterna a exibição da área de trabalho de blocos.

4.7. GENERATE LSL gera o código LSL, a partir dos blocos presentes no momento na área de trabalho, e o exibe na tela.

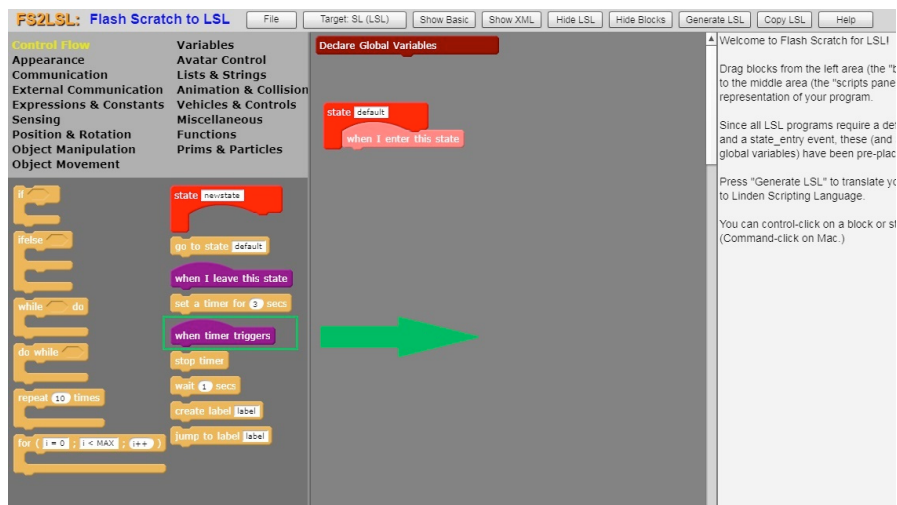
4.8. COPY LSL copia o *script* LSL na área de transferência.

4.9. HELP exibe algumas informações de ajuda sobre o FS2LSL.

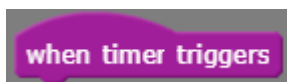
## 7.2.2.2 Blocos de programação

Os blocos a serem usados são arrastados do menu lateral esquerdo para a área de trabalho central. O código LSL é gerado na área à direita, se habilitada, quando o botão “Generate LSL” é pressionado (Figura 7.12).

Figura 7.12 - Inserção dos blocos.



Os blocos se encaixam formando estruturas mais complexas e são executados de forma sequencial, de cima para baixo.



Exemplo de bloco que pode iniciar um *script*. Notam-se a falta de encaixes na parte superior e a presença do encaixe na borda inferior.



Exemplo de bloco intermediário, que deve ser encaixado em algum outro bloco superior.



Exemplo de bloco que recebe um parâmetro booleano.



Exemplo de expressão booleana.

### 7.2.2.3 Categorias dos blocos

Todos os blocos disponíveis no FS2LSL estão presentes em uma das categorias visíveis na Figura 7.13, devidamente distribuídos, de acordo com sua funcionalidade. Os blocos também possuem uma coloração diferenciada, dependendo de sua função.

Figura 7.13 - Categorias.

<b>Control Flow</b>	<b>Variables</b>
<b>Appearance</b>	<b>Avatar Control</b>
<b>Communication</b>	<b>Lists &amp; Strings</b>
<b>External Communication</b>	<b>Animation &amp; Collision</b>
<b>Expressions &amp; Constants</b>	<b>Vehicles &amp; Controls</b>
<b>Sensing</b>	<b>Miscellaneous</b>
<b>Position &amp; Rotation</b>	<b>Functions</b>
<b>Object Manipulation</b>	<b>Prims &amp; Particles</b>
<b>Object Movement</b>	

**CONTROL FLOW:** Blocos com as funções básicas de controle de fluxo, como condicionais e repetidores.

**APPEARANCE:** Blocos relacionados à aparência do objeto.

**COMMUNICATION:** Blocos que tratam de comunicação entre objetos e outros habitantes.

**EXTERNAL COMMUNICATION:** Comunicação com fatores externos.

**EXPRESSIONS & CONSTANTS:** Expressões e constantes matemáticas e boolianas.

**SENSING:** Blocos relacionados a sensores do ambiente e do *chat*.

**POSITION & ROTATION:** Posição e rotação de objetos.

**OBJECT MANIPULATION:** Blocos que modificam características de objetos.

**OBJECT MOVEMENT:** Blocos que movimentam objetos.

**VARIABLES:** Criação e manipulação de variáveis a serem utilizadas no *script*.

**AVATAR CONTROL:** Blocos relacionados ao controle de avatares do Mundo Virtual.

**LISTS & STRINGS:** Criação e manipulação das estruturas de lista e *string*.

**ANIMATION & COLLISION:** Animação e colisão.

**VEHICLES & CONTROLS:** Blocos usados para o controle de veículos.

**MISCELLANEOUS:** Blocos variados.

**FUNCTIONS:** Criação e utilização de funções customizadas pelo usuário do FS2LSL.

**PRIMS & PARTICLES:** Blocos relacionados a primitivas e partículas.

### 7.2.2.4 Exemplos de aplicações

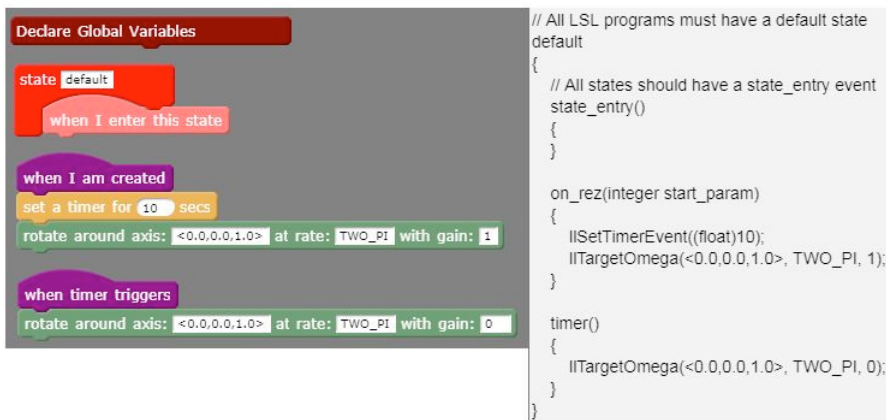
A aplicação abaixo, representada à esquerda pelo seu formato em blocos e à direita pelo código LSL gerado, usando o FS2LSL, define que, quando o objeto ligado a esse *script* for clicado, enviará a mensagem “Ei!” ao canal 0 do *chat*. Nota-se que o FS2LSL exige a existência de um estado-padrão para gerar o código (Figura 7.14).

Figura 7.14 - Codificando um *script* na ferramenta FS2LSL.



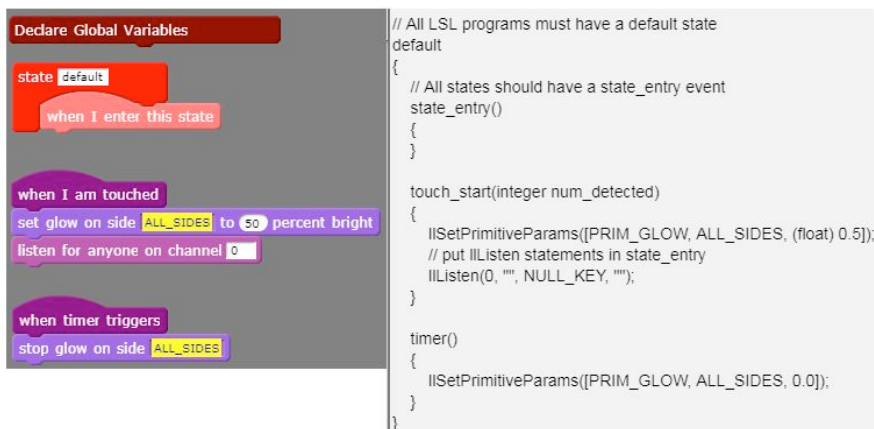
Já o próximo exemplo faz com que o objeto comece a girar, assim que for criado, a uma velocidade a uma taxa de  $2\pi$  em torno do seu eixo Z. Passados 10 segundos, o objeto cessa de girar (Figura 7.15).

Figura 7.15 - Exemplo de aplicação de blocos visuais para desenvolvimento de um MVI.



Nesse último exemplo (Figura 7.16), o objeto começa a brilhar sempre que for clicado e deixa de brilhar quando detecta alguma mensagem de seu dono no canal 0 do *chat*.

Figura 7.16 - Demonstração de blocos visuais no desenvolvimento de *scripts* LSL.





## 7.3 PROCESSO DE CARGA E DEPURAÇÃO DOS SCRIPTS

Nesta seção, primeiro será apresentado como carregar um *script* em um *prim* (um objeto no Mundo Virtual Imersivo) utilizando soluções escritas na linguagem LSL (*Linden Scripting Language*). Na próxima subseção, será apresentado o relatório de erros de depuração.

*OpenSimulator*, frequentemente chamado de *OpenSim*, é um servidor *open source* para hospedagem de Mundos Virtuais similares ao *Second Life*. *OpenSimulator* utiliza o *libsecondlife* para cuidar da comunicação entre o cliente e o servidor, portanto é possível conectar a um servidor *OpenSim* utilizando o cliente *Second Life* da Linden Lab. Outros clientes, para o *Second Life*, também, podem ser utilizados, uma vez que o *Second Life* e o *OpenSim* utilizam os mesmos protocolos de comunicação.

Tudo pode ser programado nos Mundos Virtuais por meio do LSL; pode-se fazer coisas engraçadas, como interagir um avatar com um simples objeto (cubo, por exemplo).

O LSL é uma linguagem de *scripting* que faz uso de eventos. Aqui tudo são eventos.

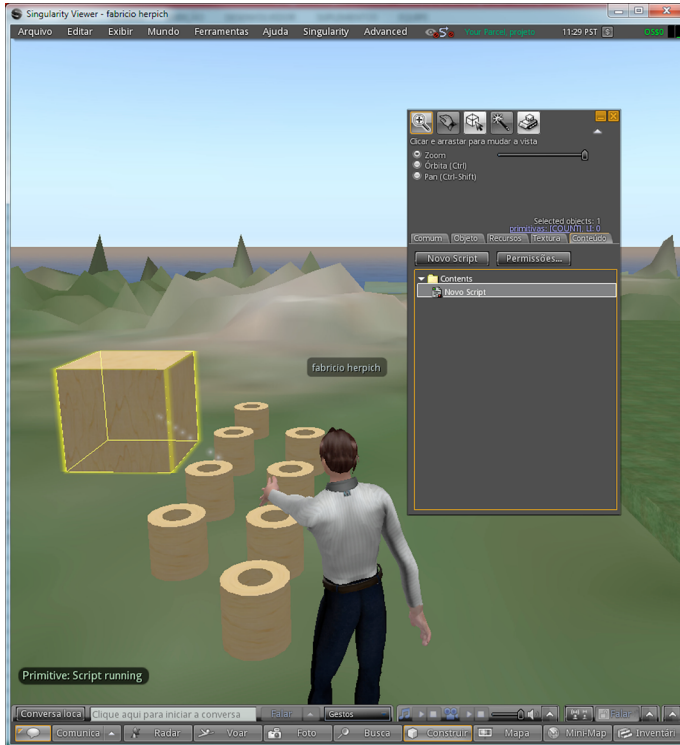
Todo o *script* segue a estrutura:

```
default
{
state_entry()
{
  IISay(0, "Script running");
}
touch_start(integer total_number)
{
  IISay(0, "Hello Avatar");
}
}
```

Há um estado *default* e dentro desse estado podem-se ter outros. Nesse caso, será apresentado o estado *state\_entry* e no *touch\_start*.

Para se iniciar a escrever o *script*, é preciso colocar um cubo no chão, *right-click* (clique com o botão direito do *mouse*) e escolher a opção opção Editar ("*Edit*"). Em seguida, ir para o separado Conteúdo ("*Content*") e escolher Novo Script ("*New script*").

Figura 7.17 - Escrita de script.



Abrindo o *script*, está pronto para programar.

Para começar, será apresentada uma programação tal que, quando o avatar formular algo, o nosso cubo responda com o típico “*Hello World!*”!

Para isso, precisa-se colocar no *state\_entry* uma função que permita ouvir o que o nosso avatar diz. O *state\_entry* é logo executado quando o *script* é iniciado.

Essa função é o *llListen* (*integer channel, string name, key id, string msg*).

- *Channel* é o canal em que vamos falar, 0 (zero), se for um canal público, em que todos ouvem o que é dito, ou outro número inteiro para comunicações privadas.
- *Name* é usado quando queremos especificar o nome do avatar ou objeto. Caso contrário, deixa-se em branco: usa-se “ ”.
- ID é semelhante ao de cima, mas usando o UUID – identificação do avatar/Objeto.
- *msg* é a mensagem que queremos transmitir.

Dessa forma, com esse desempenho, o objeto irá receber a nossa mensagem. Essa função é complementada por:

```
listen( integer channel, string name, key id, string message )
{
  ISay(0,"Hello "+message);
}
```

Dessa forma, o que se escreve no *chat* (*World*, neste caso) será ouvido pelo objeto, guardado na variável *message* e concatenado à palavra *Hello*, formando o *Hello World*.

A função *ISay* (*channel*, *message*) é utilizada para comunicar entre avatares/Objetos. Existem outras funções desse tipo.

Figura 7.18 - Comunicação avatar/objetos.



O `touch_start` permite executar instruções quando o avatar clica no objeto, ou seja, quando este evento é disparado. Nesse caso, ao ser clicado, o objeto diz apenas *“Hello Avatar”*.

## 7.3.1 Relatório de erros e depuração

Pode-se usar o mesmo conceito para rastrear problemas depois que o código for concluído da seguinte forma:

```
notify ( string genus, string message )
{
    criador chave = "a822ff2b-ff02-461d-b45d-dcd10a2de0c2" ;
    // linha próxima do comentário para desativar as notificações
    // genus = "off"
    if ( genus == "dbg" )
    {
        // comentou a próxima linha para desativar a depuração
        IInstantMessage ( criador, "DEBUG:" + mensagem ) ;
    }
    else if ( genus == "err" )
    {
        IOwnerSay ( "ERRO:" + mensagem ) ;
        IInstantMessage ( criador, "ERRO:" + mensagem + "no objeto" + IGetObjectName
        () + "que pertence a" + IKey2Name ( IGetOwner ( ) ) + "na região" + IGetRegion-
        Name ( ) ) ;
    }
    else if ( genus == "info" )
    {
        // no prefixar porque é uma mensagem normal
        IOwnerSay ( mensagem ) ;
    }
    outro
    {
        ; // não faz nada, as notificações foram desativadas
    }
}
```

Aqui se faz uma série de coisas: lidar com diferentes mensagens informativas à execução normal, relatório de erros para o proprietário e criador e mensagens de depuração. Para desativar a depuração, simplesmente comente a linha de depuração, conforme mostrado no código, e descomente a linha `genus = "off"` para desativar todas as notificações.

Usa-se o `IIOwnerSay` para o proprietário atual e o `IIInstantMessage` para entrar em contato com o criador do *script* que continuará a funcionar, depois que a propriedade for aprovada. Basta inserir sua própria chave na parte superior.

Outros mecanismos podem ser usados e incluem `IIEmail` ou `IIHTTPRequest`, para postar uma mensagem, em um sistema de *log* externo, talvez.

Definem-se os próprios tipos de mensagem personalizados e se comunica usando o método mais apropriado.

### **7.3.2 Mais reutilização**

Pode-se estender ainda mais a reutilização, tornando-a um *script* discreto por si próprio dentro do objeto da forma:

```

notify ( string genus, string message )
{
    criador chave = "a822ff2b-ff02-461d-b45d-dcd10a2de0c2" ;
    // desmarque esta linha para desativar as notificações
    // genus = "off"
    if ( genus == "dbg" )
    {
        // comentou esta linha para desativar a depuração
        IInstantMessage ( criador, "DEBUG:" + mensagem ) ;
    }
    else if ( genus == "err" )
    {
        IOwnerSay ( "ERRO:" + mensagem ) ;
        IInstantMessage ( criador, "ERRO:" + mensagem + "no objeto" + IGetObject Name
( ) + "que pertence a"
            + IKey2Name ( IGetOwner ( ) ) + "na região" + IGetRegionName ( ) ) ;
    }
    else if ( genus == "info" )
    {
        // no prefixar porque é uma mensagem normal
        IOwnerSay ( mensagem ) ;
    }
    outro
    {
        ;// não faz nada, as notificações foram desativadas
    }
}
padrão
{
state_entry ( )
{
    notificar ( "info", "sistema de notificação pronto" ) ;
}
link_message ( integer sender_num, integer num, string str, id da chave )
{
    // estamos usando a chave como um segundo parâmetro de string
    notificar ( str, id ) ;
}
}
}

```



Para enviar uma notificação, simplesmente use sua implementação -padrão:

```
{
    touch_start ( integer num_detected )
    {
        chave id = IIDetectedKey ( 0 );
        string message = IKey2Name ( id ) + "tentou me roubar !";
        // isso pode parecer estranho, mas é válido usar o parâmetro chave de IMessageLinked
        // como um segundo recurso de passagem de string se não for realmente usado por
        // uma chave
        IMessageLinked ( LINK_SET, 0, "info", mensagem );
    }
}
```

## 7.4 FORMAÇÃO PARA AUTORIA EM LSL E OSSL

Nesta seção, serão abordados temas de relevância para a formação de indivíduos capacitados a construir soluções em ambientes imersivos. Na primeira subseção (7.4.1), é apresentada a formação inicial em programação, a qual demonstra, por um conjunto de paradigmas distintos, como é possível capacitar programadores iniciantes. Já na seção seguinte (7.4.2), é abordada uma concepção para a formação de docentes, a qual visa qualificá-los para a utilização dos Mundos Virtuais como ferramenta no processo de ensino e de aprendizagem.

## 7.4.1 A formação inicial em programação

Atualmente o uso de tecnologias é considerado um dos alicerces do novo cenário mundial, em que o indivíduo está constantemente exposto a ambientes computacionais. Sua capacidade de utilizar tais recursos habilita-o a tomar decisões diárias, as quais necessitam de aptidão e domínio básico da lógica de programação. Essas são mudanças que exigem adaptação e, desta forma, possibilitam a formação do cidadão do século XXI.

Identificando esse momento de transformação, denota-se a importância da aprendizagem da Lógica de Programação, a qual já se encontra presente em diferentes meios, das escolas às universidades. Compreender conceitos de raciocínio lógico é condição mínima para que os alunos alcancem um desempenho desejável em cursos de nível superior com disciplinas de programação em seus projetos pedagógicos, conforme apontado por Santos e Costa (2006). Alinhado a esse conceito, observa-se a necessidade da construção do Pensamento Computacional por esses indivíduos, que pode ser identificado como o conjunto de competências e habilidades básicas relacionadas à competência de aplicar estratégias algorítmicas para a solução de problemas (WING, 2006).

Em vários estudos, Silva, Silva e Santos (2014), Mattos e Vahldick (2008), Silva e Raabe (2008) e Falckembach e Araújo (2013) apontam que a dificuldade da construção do conhecimento pelos estudantes sobre programação é atribuída à metodologia de ensino dita tradicional, visto que esse modelo, segundo Petry (2005), muitas vezes não atende às expectativas do estudante, não atraindo o seu interesse pelos conteúdos.

Ao aprofundar os estudos sobre programação, Borges (2000) acrescenta que o ensino de algoritmos comumente aborda um conjunto de ações comuns, conforme apresentado na Figura 7.19, iniciando-se por informações teóricas, exemplos, exercícios e execução de projetos mais avançados.

Figura 7.19 - Ações para o ensino de algoritmos e programação.



Fonte: Borges (2000).

Uma tendência que aponta resultados pertinentes é a adoção do ensino de algoritmos voltado à construção do Pensamento Computacional para crianças. Essa é uma estratégia adotada pelo Departamento de Educação Britânico (UNITED KINGDOM, 2013), tendo como foco disponibilizar um recurso educacional que possibilite ao aluno utilizar o Pensamento Computacional para resolver de forma criativa diferentes problemas. Dessa forma, habilita-se o estudante a reconhecer, interpretar e sugerir soluções computacionais para os problemas a ele apresentados.

Além da proposta Britânica, encontram-se, na literatura, diferentes práticas pedagógicas que vislumbram alcançar um grau elevado de efetividade no ensino de programação, como o uso de realidade virtual (BARBAS; LOPES, 2013), adoção de dispositivos móveis como ferramenta de apoio (BARCELOS; SILVEIRA, 2012), prática com o uso de animações, simulações e *web* (BECK *et al.*, 2014), intervenções complexas (BROWN, 1992), ambientes interativos – projeto Alice (CONWAY; PAUSCH, 1997), ensino de programação por demonstração (FERREIRA *et al.*, 2010), robótica como instrumento de aprendizagem e a utilização de redes sociais com prática em algoritmos (MIRANDA, 2014).

A partir da concepção de que o modelo tradicional de ensino de programação não atende à demanda dos principiantes e reconhecendo a evolução das tecnologias, é possível apontar a necessidade de mudanças no processo de ensino e de aprendizagem para iniciantes. Assim sendo, as práticas atuais devem ser apoiadas por novas tecnologias (ambientes/plataformas), buscando, dessa forma, disponibilizar recursos multimídia (animações, interatividade, vídeos) ao estudante.

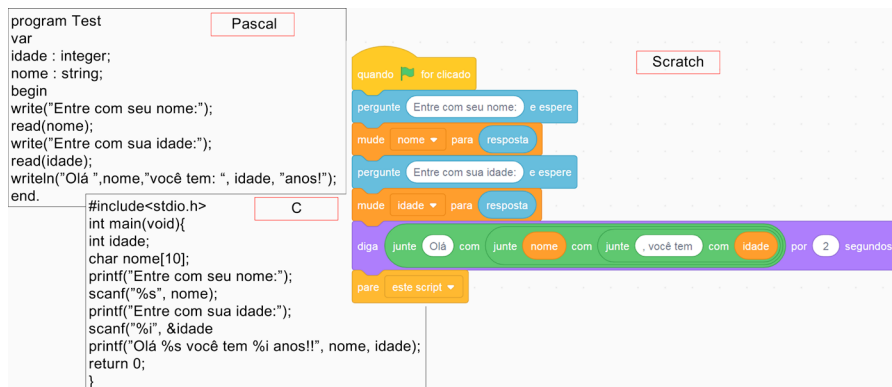
Um ponto relevante nesse contexto é o fato de que, para a construção do conhecimento sobre programação, o aluno precisa ser capaz de definir padrões e ordená-los de forma sistemática, além de reconhecer problemas e lidar com eles, a fim de resolvê-los. Porém, a habilidade do estudante, ao visualizar uma solução e mapeá-la no espaço e conceber o código relacionado, o auxilia na compreensão de todos os aspectos associados à programação, o que está diretamente pautado sobre a teoria de inteligência espacial. Ao entender essa concepção, é possível afirmar que a utilização de recursos visuais, como a programação em blocos vi-

suais, beneficia o estudante no processo de aprendizagem da lógica e programação – teoria já comprovada pelos estudos sobre a linguagem LOGO de Abelson e Disessa (1981) e Papert (1985).

Embora a LOGO seja uma linguagem simples, não teve muito êxito em sua finalidade de ser uma maneira fácil de programar por inúmeros motivos, segundo Resnick *et al.* (2009). Nesse sentido, Papert (1980) argumentou que o entendimento de programação deva ser fácil para começar (*low floor*), permita a construção de projetos elevados ao longo do tempo (*high ceiling*) e apoie diferentes projetos para indivíduos com diferentes interesses (*wide walls*). Ou seja: para que uma linguagem de programação seja aceita, ela deve possuir piso baixo, teto alto e paredes largas, características inerentes ao ambiente gráfico de programação Scratch, desenvolvido pelo *Lifelong Kindergarten Group* (LKG, 2014). Constitui-se de uma interface baseada em ícones e blocos com a finalidade de facilitar o processo de aprendizagem de conceitos de computação, por meio do pensamento criativo, trabalho colaborativo e raciocínio sistemático (SCRATCH, 2014).

A fim de demonstrar a facilidade da programação com o Scratch, na Figura 7.20, são demonstrados três códigos com a mesma funcionalidade, porém em linguagens diferentes (Pascal, C e Scratch). É visível a diferença de sintaxe e complexidade de implementação de cada aplicação.

Figura 7.20 - Comparação da programação em blocos visuais e outras linguagens.



Fonte: Adaptado de Scaico et al. (2012).

A partir da necessidade de uma solução, para o ensino de programação para iniciantes, baseada em recursos computacionais atuais e que permita ao leitor iniciar seus primeiros passos para o desenvolvimento de *scripts* para Mundos Virtuais, será apresentada, nesta etapa do livro, uma adaptação da metodologia proposta por Amaral, Medina e Tarouco (2016), na qual sugerem a adoção de uma abordagem didática, calcada em novos paradigmas, para o processo de ensino e de aprendizagem de algoritmos, buscando efeitos positivos. Objetiva-se, dessa forma, a utilização de uma estrutura didática de ensino baseada no entendimento sobre raciocínio lógico, construção de fluxogramas, programação com a linguagem *Scratch*, desenvolvimento com o *Scratch4OS* e construção de *scripts* para Mundos Virtuais com o *LSL*, conforme apresentado na Figura 7.21.

Figura 7.21 - Metodologia proposta para ensino de iniciantes em programação.



Fonte: Adaptado de Amaral, Medina e Tarouco (2016).

A aplicação dessa proposta pelos autores em sua pesquisa visou a estimular o aluno a desenvolver soluções de seu interesse, focando nos níveis motivacionais que poderiam ser alcançados com o uso desses sistemas. A motivação gerada pelo interesse desses estudantes, ao interagirem com as aplicações Scratch, Scratch4OS e *OpenSim*, fomentou a busca por um alto padrão de envolvimento. A motivação necessária ocorreu pela possibilidade de simulação aberta inerente a essas aplicações, que possibilitou, por meio da simulação, o desenvolvimento de hipóteses, testes e ajustes dos conceitos observados, assim como proposto por Valente (1997).

No modelo proposto, tem-se como primeira etapa a construção do raciocínio e da lógica por parte dos iniciantes, visto que as principais causas dos problemas de programação, percebidas por programadores novatos, segundo Proulx (2000) e Faria e Adán-Coello (2004), baseiam-se no desconhecimento de padrões de raciocínio lógico na solução de problemas. Sugere-se como recurso a utilização de problemas de raciocínio lógico, como os disponibilizados no Portal Plastelina (<<http://www.plastelina.net/>>). Essas atividades buscam desenvolver a capacidade do estudante para seguir uma sequência lógica a fim de solucionar os desa-

fos propostos. Como exemplo, cita-se o problema da Ovelha, Frutos e Lobo, apresentado na Figura 7.22, em que o estudante deve transportar esses três elementos de uma margem à outra do rio, um de cada vez, respeitando a regra de que a ovelha não pode permanecer sozinha na mesma margem que os frutos nem permanecer com o lobo.

Figura 7.22 - Prática de raciocínio lógico (ovelha, frutos e lobo).

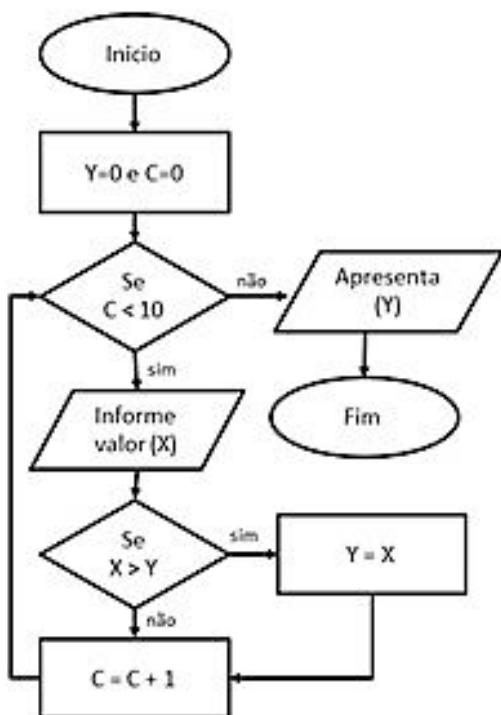


Fonte: <<http://www.plastelina.net/>>.

Como segunda etapa no processo de aprendizagem de programação, é proposta a prática de resolução de problemas pelo uso de fluxogramas. Entendendo que os indivíduos apresentam uma maior capacidade de reconhecer algoritmos por meio de recursos visuais, a representação gráfica por fluxogramas (BRUSILOVSKY, 1994) pode ser considerada uma solução, a fim de que o aluno implemente respostas estruturadas e teste conceitos de maneira fácil e intuitiva, a partir da organização de símbolos com um fluxo e sequência lógica formalizados. Um exemplo de fluxograma é demonstrado na Figura 7.23.



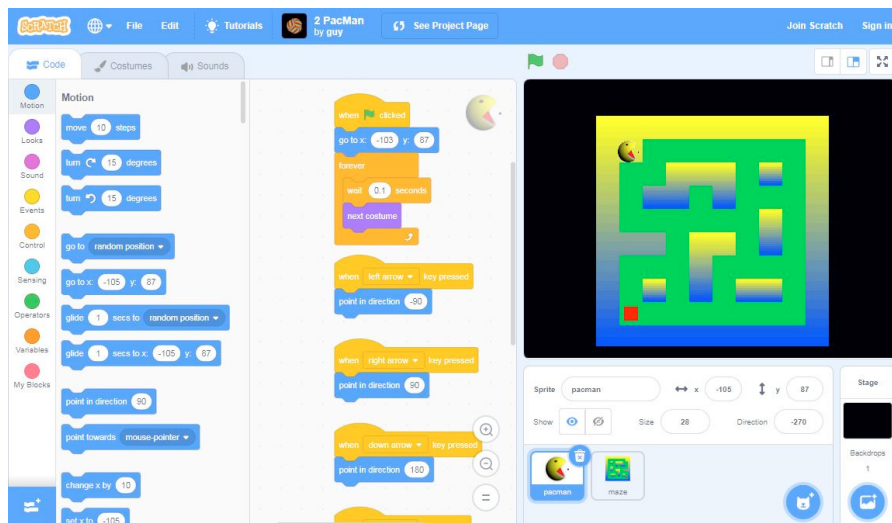
Figura 7.23 - Exemplo da resolução de problema com fluxograma.



Fonte: Amaral, Medina e Tarouco (2016).

Com base na construção do raciocínio lógico, é possível ter os primeiros contatos com uma solução de programação. Nesse sentido, a fim de proporcionar um ambiente amigável e de simples entendimento, propõe-se a utilização do Scratch (Figura 7.24), uma linguagem de programação em blocos visuais de fácil utilização, conforme apontado por Resnick *et al.* (2009). Busca-se desenvolver a capacidade dos indivíduos de implementar programas simples utilizando os recursos dessa linguagem.

Figura 7.24 - Exemplo de aplicação (jogo) desenvolvida com Scratch.



Fonte: <<https://scratch.mit.edu/>>.

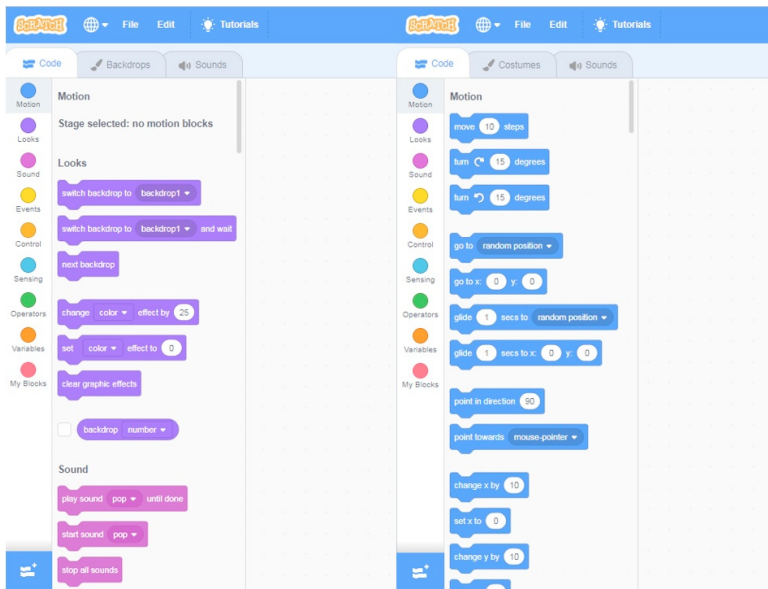
Nessa etapa, reconhecendo que o indivíduo tenha passado pela experiência de programar com o Scratch e, dessa forma, desenvolvido habilidades necessárias à construção de soluções algorítmicas para problemas, é proposto o contato com outra ferramenta, o Scratch4OS, um *software* baseado na linguagem Scratch que também trabalha com o conceito de programação pela movimentação de blocos visuais (AVILA *et al.*, 2013). O diferencial do Scratch4OS está no fato de permitir a tradução das construções visuais em *scripts* para Mundos Virtuais, funcionalidade que possibilita a programação de diferentes ações para objetos em um metaverso.

A utilização do Scratch4OS permite ao estudante ter os primeiros contatos com a sintaxe da linguagem LSL, pela construção e transporte de códigos desenvolvidos no Scratch4OS para os objetos (também denominados *prisms*) do próprio Mundo Virtual *OpenSim*. Essa atividade

possibilita ao programador reconhecer que estruturas criadas pelos blocos visuais podem ser convertidas em código (*scripts* LSL). Em suma, essas intervenções visam promover uma estrutura de ligação entre imagem e texto, a fim de que o aluno reconheça e compreenda a sintaxe dos *scripts* em LSL, relacionando o seu funcionamento com a base de conhecimento construída com os estudos sobre programação no Scratch.

A Figura 7.25 demonstra a semelhança dos ambientes de desenvolvimento Scratch e Scratch4OS.

Figura 7.25 - Ambientes de desenvolvimento Scratch e Scratch4OS.



Fonte: Amaral, Medina e Tarouco (2016).

Nesse ponto, o método descrito pelos autores pressupõe que o indivíduo já tenha construído o conhecimento necessário sobre lógica de programação, algoritmos e resolução de problemas e que, dessa forma, esteja apto a construir soluções utilizando a linguagem de programação LSL. Observa-se que elementos como sintaxe e formalidade, no desenvolvimento com a linguagem, serão obstáculos a serem ultrapassados. Contudo, a base sólida construída até o momento viabiliza e reduz o impacto gerado por estes elementos.

Por fim, com o intuito de comprovar a viabilidade desta metodologia, são apresentados resultados alcançados com a sua aplicação em cursos de nível superior na Universidade Federal do Pampa, Campus Bagé. Os dados na íntegra podem ser obtidos em Amaral, Medina e Tarouco (2016).

A aplicação dessa metodologia teve como foco o aprimoramento do processo de ensino e de aprendizagem da lógica, algoritmos e programação. Os diferentes paradigmas adotados estimularam as habilidades dos indivíduos, com os quais conseguiu-se propor soluções computacionais calcadas sobre o pensamento computacional e o raciocínio lógico efetivo. Além dessas observações, analisou-se o desempenho dos estudantes com base nas avaliações realizadas durante três semestres, com turmas distintas (Tabela 7.1). Salienta-se que, nos dois primeiros semestres (2013/2 e 2014/1), a ordem dos recursos (Lógica, Scratch e LSL) foi alterada e, no semestre 2014/2, adotou-se exatamente a sequência descrita neste capítulo.

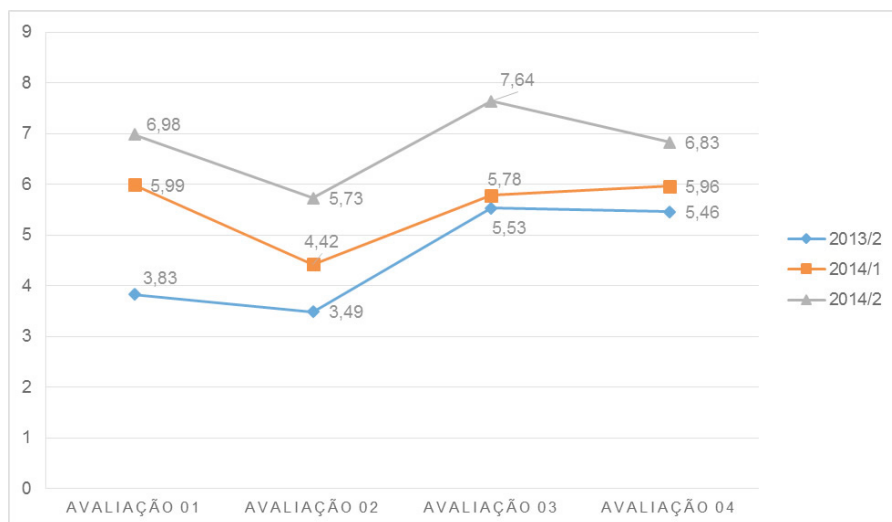
**Tabela 7.1 - Participantes dos experimentos com a metodologia proposta.**

Semestre x Matrículas		
Semestre	Matriculados	Frequentes
2013/2	60	21
2014/1	60	25
2014/2	60	22

Fonte: Amaral, Medina e Tarouco (2016).

A evolução dos alunos é evidenciada, na Figura 7.26, pelo gráfico que apresenta a comparação do desempenho dos estudantes nas diferentes atividades avaliativas realizadas durante os respectivos semestres. O progresso é percebido em todas as quatro avaliações, com um aumento de 0,96 ponto na média de 2014/1 em relação a 2013/2, 1,26 de 2014/2 para 2014/1 e um avanço significativo de 2,22 pontos alcançados pelos alunos de 2014/2 sobre o grupo de 2013/1.

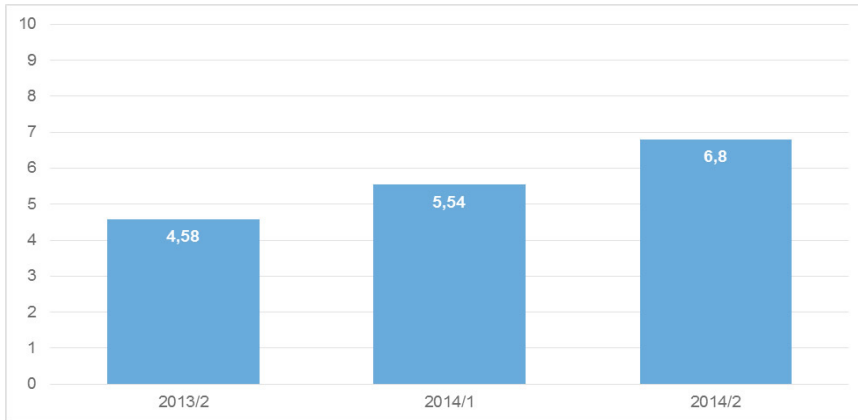
Figura 7.26 - Desempenho em atividades avaliativas por semestre.



Fonte: Amaral, Medina e Tarouco (2016).

A diferença entre as médias finais de todos os alunos nos semestres observados é mostrada na Figura 7.27. Assim, denota-se o gradual incremento dos resultados alcançados pelos estudantes nas disciplinas de algoritmos. A maior margem de intervalo de notas é percebida no grupo geral, exibindo um acréscimo médio entre os semestres de 1,11 ponto.

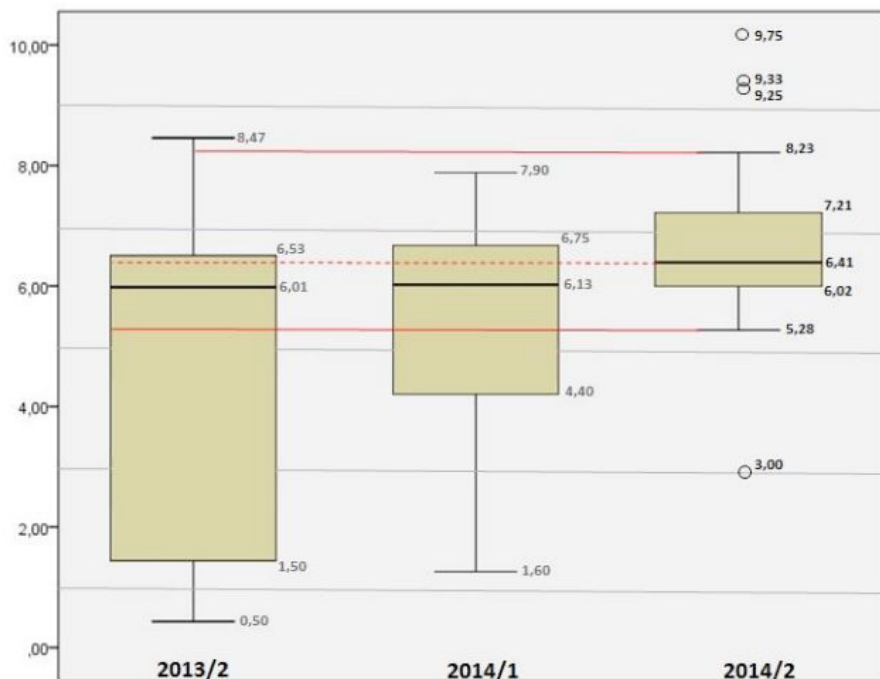
Figura 7.27 - Média final dos alunos.



Fonte: Amaral, Medina e Tarouco (2016).

A seguir, é apresentado o gráfico de comparação entre as tendências das médias dos estudantes para o período (Figura 7.28) considerando as notas individuais. Destaca-se, na representação dos *outliers* dos dados de 2014/2, referentes às notas de três acadêmicos (9,25/9,33/9,75), o bom nível de habilidade de programação desses estudantes. O limite superior no BoxPlot de 2013/1 encontra-se acima dos demais semestres, contudo a mediana para esse semestre é a menor observada, sobressaindo novamente 2014/2, com um valor de média 7,21 para o terceiro quartil e 6,41 como mediana, sendo esses dados superiores aos demais períodos. Como informação relevante nesta observação, distingue-se o limite inferior denotado no segundo semestre de 2014, apontando um valor de 5,28, caracterizando que a predisposição para a melhora no desempenho ocorreu de forma linear e equânime, tanto pelos aprovados quanto pelos discentes com notas reduzidas (abaixo da média).

Figura 7.28 - Estudo comparativo entre médias.



Fonte: Amaral, Medina e Tarouco (2016).

Por fim, esta seção buscou apresentar uma proposta à formação inicial de programadores, vislumbrando que eles construam o conhecimento e desenvolvam habilidades necessárias à implementação de soluções algorítmicas em ambientes imersivos. Reconhecendo que o ensino de algoritmos e programação não é uma tarefa simples, foi proposta a adoção de uma metodologia que comprovadamente teve bons resultados como estratégia para os primeiros passos do iniciante no mundo da programação.



## 7.4.2 A formação de docentes

A constante evolução dos dispositivos eletrônicos, em termos de memória, interface gráfica e capacidade de processamento, tem propiciado maior acesso a *softwares* que demandam um intenso uso desses componentes em função da sua alta complexidade.

Esse é o caso dos Mundos Virtuais, ambientes imersivos que simulam, com alto grau de detalhamento, aspectos do mundo real. Salienta-se que o uso dos Mundos Virtuais como ferramentas de réplica não necessariamente fica restrito à realidade. Ambientes dessa natureza são comumente utilizados em jogos eletrônicos, simulando fantasias derivadas da realidade concreta. Como bem coloca Wang (2011, p. 4), “[...] computadores podem simular um cavalo, mas eles também podem facilmente simular um Pegasus”.

Em função destas características, Mundos Virtuais tornam-se ambientes muito propícios para o desenvolvimento de situações de ensino e aprendizagem calcadas na contextualização do conhecimento e na realização de tarefas autênticas. Tarefas autênticas são definidas por Jonassen (1999) como atividades que apresentam relevância para o estudante. Trata-se de desafios cognitivos com os quais o estudante deve se deparar no intuito de explorar, na prática, a aplicabilidade dos conhecimentos escolares em situações da vida real. Van Merriënboer e Kirschner (2013) esclarecem que o pleno domínio do conhecimento requer o desenvolvimento de habilidades para manipulá-lo e adequá-lo a diferentes situações.

Nos Mundos Virtuais, o estudante encontra a possibilidade de imergir em cenários contextualizados (ou até mesmo de construí-los) e, dentro desses contextos, pode realizar investigações de acordo com as propostas pedagógicas estabelecidas, participar de simulações, interagir com objetos 3D (nem sempre facilmente construídos no ambiente concreto). Enfim, há uma diversidade de possibilidades pedagógicas alinhadas à ideia de construção de um conhecimento contextualizado que podem ser exploradas a partir de ambientes imersivos.

O aprimoramento de atributos como a capacidade de processamento e interfaces gráficas das tecnologias disponíveis no mercado vem criando um cenário propício à profusão de ambientes imersivos como os Mundos Virtuais. Esses ambientes, com características semelhantes aos jogos eletrônicos, simulam espaços complexos e podem ser explorados como réplicas virtuais do mundo real, ensejando o oferecimento de novas situações de ensino e aprendizagem que oportunizam a contextualização do conhecimento e o desenvolvimento de tarefas autênticas. Entretanto, poucos docentes têm domínio sobre esta tecnologia, ainda considerada de alta complexidade. Com vista a promover uma aproximação do docente da Educação Básica com os ambientes imersivos, foi elaborado um conjunto de estratégias para compor um programa de formação voltado ao desenvolvimento da prática docente nos Mundos Virtuais, envolvendo a produção de artefatos educacionais para tais ambientes (AVILA, 2016).

No intuito de delinear como poderia ocorrer o processo de autoria de futuros docentes no Mundo Virtual, foi desenvolvido e testado um programa curricular destinado à produção de artefatos educacionais para ambientes imersivos.

O planejamento da atividade de formação envolveu constantes passagens pelo Mundo Virtual, no qual os estudantes tinham a missão de construir seus próprios laboratórios virtuais de aprendizagem, em suas respectivas áreas do conhecimento. Os laboratórios deveriam ser compostos por materiais educacionais produzidos em ferramentas de autoria exploradas na disciplina, pela importação de conteúdos 3D e animações disponíveis em repositórios *on-line* e pela adição de *scripts* a objetos do Mundo Virtual, construídos com o apoio das ferramentas *Scratch for OpenSim* e *ScripTastic*.

Na atividade inicial da formação, solicitou-se aos alunos que explorassem o Mundo Virtual, fizessem leituras relacionadas ao uso de ambientes imersivos no contexto educacional e, por fim, idealizassem um tema dentro de sua área de conhecimento o qual deveria nortear os artefatos que seriam produzidos. Com base no tema escolhido, seria construído, pouco a pouco, um laboratório virtual de aprendizagem dentro do Mundo Virtual. Para tanto, foi organizado o planejamento (Tabela 7.2).

**Tabela 7.2 - Atividades de Planejamento.**

Planejamento extensão	
Semana 1 (4h)	<p><b>Conhecendo os Mundos Virtuais</b>            Discussão sobre Mundos Virtuais na Educação.            Exploração inicial do Mundo Virtual.</p> <ul style="list-style-type: none"> <li>• Baixar e configurar visualizador para acesso.</li> <li>• Cadastro no OSGRID.</li> <li>• Visita a mundos no OSGRID.</li> <li>• Cadastro no <i>grid Second Life</i>.</li> <li>• Visita aos mundos do <i>Second Life</i>.</li> </ul>
Semana 2 (4h)	<p><b>Desenvolvendo atividades básicas no Mundo Virtual</b></p> <ul style="list-style-type: none"> <li>• Construção e edição de objetos.</li> <li>• Comunicação via bate-papo.</li> <li>• Adição de imagens de páginas <i>web</i>.</li> </ul>
Semana 3 (4h)	<p><b>Definição de um tema a ser explorado no Mundo Virtual</b> (discussão inicial com ideias para diferentes áreas do conhecimento).</p> <ul style="list-style-type: none"> <li>• Importação de objetos a partir do repositório <i>OpenSim Creations</i> e do <i>Sketchup</i>.</li> <li>• Animação de avatares.</li> </ul>
Semana 4 (4h)	<p><b>Construção de <i>scripts</i> para o Mundo Virtual</b></p> <ul style="list-style-type: none"> <li>• Apresentação da estrutura de programação da linguagem LSL. Idealização de uma animação simples a ser incorporada ao laboratório de cada professor. Utilização das ferramentas <i>ScriptTastic</i> e <i>Scratch for OpenSim</i>. Utilização dos repositórios de <i>script</i> das linguagens LSL e OSSL:            &lt;<a href="http://wiki.secondlife.com/wiki/LSL_Portal/pt">http://wiki.secondlife.com/wiki/LSL_Portal/pt</a>&gt;,            &lt;<a href="http://OpenSimulator.org/wiki/OSSL_Implemented">http://OpenSimulator.org/wiki/OSSL_Implemented</a>&gt;.</li> </ul>
Semana 5 (4h)	<p><b>Instalando um servidor na máquina do usuário</b></p> <ul style="list-style-type: none"> <li>• Instalar e configurar um servidor de <i>OpenSim</i> na máquina do usuário.</li> <li>• Baixar e importar bibliotecas para o Mundo Virtual.</li> <li>• Salvar diferentes versões no mundo e abri-las por meio do servidor.</li> </ul>
Semana 6 (4h)	<p><b>Apresentação dos laboratórios e avaliação por parte dos grupos</b></p> <ul style="list-style-type: none"> <li>• Discussão coletiva sobre possíveis melhorias a serem implementadas em cada um dos laboratórios produzidos.</li> </ul>

A cada encontro presencial, novas ferramentas eram apresentadas e atividades de teste eram realizadas. Após a apresentação inicial, os docentes realizavam suas próprias produções, primeiramente, testando as ferramentas e, posteriormente, contribuindo para a construção de

materiais que deveriam compor os laboratórios de aprendizagem de seus respectivos grupos. Ao todo, na busca por contemplar o conteúdo descrito no planejamento do curso, seis tipos de atividades foram desenvolvidas nos encontros presenciais:

1. Modificar aparência do avatar.
2. Uso da ferramenta *chat*.
3. Criação e importação de objetos.
4. Inserção de mídias no Mundo Virtual.
5. Uso de animações.
6. Construção de *scripts*.

Com este conjunto de atividades, buscou-se oferecer aos docentes um domínio básico sobre ferramentas que compõem os Mundos Virtuais e que podem ser exploradas de forma simplificada por usuários leigos em *design* e programação. A partir da proposta de uso dos Mundos Virtuais como espaços para a construção de laboratórios virtuais de aprendizagem, foi definido um conjunto inicial de estratégias que buscou promover tanto a autoria dos docentes sobre o uso de tais ferramentas quanto a sua reflexão sobre o potencial pedagógico dos Mundos Virtuais no âmbito de suas respectivas áreas de conhecimento.

## REFERÊNCIAS

ABELSON, Harold; DISESSA, Andrea A. **Turtle geometry**: computations as a medium for exploring mathematics. Cambridge, MA: MIT, 1981.

AMARAL, Érico Marcelo Hoff do; MEDINA, Roseclea Duarte; TAROUCO, Liane Margarida Rockenbach. Processo de ensino e aprendizagem de algoritmos integrando ambientes imersivos e o paradigma de blocos de programação visual. *In: WORKSHOPS DO CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – WCBIE, 5., 2016, Uberlândia. Anais [...].* Porto Alegre: Sociedade Brasileira de Computação – SBC, 2016.

AVILA, Bárbara Gorziza. **Formação docente para a autoria dos mundos virtuais**: uma aproximação do professor às novas demandas tecnológicas. 2016. 231 f. Tese (Doutorado em Informática na Educação) – Centro de Estudos Interdisciplinares em Novas Tecnologias, Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2016.

AVILA, Bárbara Gorziza; AMARAL, Érico Marcelo Hoff do; TAROUCO, Liane Margarida Rockenbach. Implementação de laboratórios virtuais no metaverso OpenSim. **Renote – Revista novas tecnologias na educação**, Porto Alegre, v. 11, n. 1, 2013.

BARBAS, Maria; LOPES, Nuno. Introdução à programação: (re)construção de espaços educacionais em realidade aumentada. **Revista da UIIPS**, Santarém, Portugal, v. 1, n. 3, p. 40-55, 2013.

BARCELOS, Thiago Schumacher; SILVEIRA, Ismar Frango. Pensamento computacional e educação matemática: relações para o ensino de computação na educação básica. *In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO – WEI, 20.; CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO – CSBC, 32., 2012, Curitiba. Anais [...].* Porto Alegre: Sociedade Brasileira de Computação – SBC, 2012.

BORGES, Marcos Augusto F. Avaliação de uma metodologia alternativa para a aprendizagem de programação. *In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO – WEI, 8.; CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO – CSBC, 20., 2000, Curitiba. Anais [...].* Porto Alegre: Sociedade Brasileira de Computação – SBC, 2000.

BRONDANI, Matheus Beck; MOZZAQUATRO, Patricia Mariotto; ANTONIAZZI, Rodrigo Luiz. Ambiente de simulação e animação para o ensino de programação. **Revista interdisciplinar de ensino, pesquisa e extensão**, v. 1, n. 1, 2013.

BROWN, Ann L. Design experiments: theoretical and methodological challenges in creating complex interventions in classroom settings. **The journal of the learning sciences**, v. 2, n. 2, p. 141-178, 1992.

BRUSILOVSKY, Peter. Program visualization as a debugging tool for novices. *In*: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS – INTERCHI 93, 1993, Amsterdam, Netherlands. **Proceedings [...]**. New York: ACM Press, 1994. p. 29-30.

CONWAY, Matthew J.; PAUSCH, Randy. Alice: easy to learn interactive 3D graphics. **Computer & graphics**, v. 31, n. 3, p. 58-59, 1997.

FALKEMBACH, Gilse Antoninha Morgental; ARAÚJO, Fabrício Viero de. Aprendizagem de algoritmos: dificuldades na resolução de problemas. *In*: CONGRESSO SUL-BRASILEIRO DE COMPUTAÇÃO – SULCOMP, 2013, Criciúma. **Anais [...]**. [S.l.: s.n.], 2013. v. 2, n. 2.

FARIA, Eustáquio São José de; ADÁN-COELLO, Juan Manuel. Detectando diferenças significativas entre programas como auxílio ao aprendizado colaborativo de programação. *In*: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO – WEI, 12., CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO – CSBC, 24., 2004, Salvador. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2004.

FERREIRA, Cláudia Caroline; GONZAGA, Flávio Barbieri; SANTOS, Rodrigo Pereira dos. Um estudo sobre a aprendizagem de lógica de programação utilizando programação por demonstração. *In*: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO – WEI, 18., CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO – CSBC, 30., 2010, Belo Horizonte. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2010.

GUÍA de iniciación Linden Scripting Language. 2018. Disponível em: <<https://docplayer.es/32535209-Guia-de-iniciacion-linden-scripting-language.html>>. Acesso em: 23 maio 2018.

JONASSEN, David H. Designing constructivist learning environments. *In*: REIGELUTH, Charles M. **Instructional-design theories and models**. New Jersey: Lawrence Erlbaum Associates, 1999. p. 215-239.

KARIUKI, David. New LSL Editor released. Blog Hypergrid Business, [s.l.], 2018. Disponível em <<https://www.hypergridbusiness.com/2018/03/new-lsl-editor-released/>>. Acesso em: 10 nov. 2018.

LINDEN RESEARCH. **Second Life**. Getting started with LSL. Linden Research, Boulder, CO, USA, 2015. Disponível em: <[http://wiki.secondlife.com/wiki/Help:Getting\\_started\\_with\\_LSL](http://wiki.secondlife.com/wiki/Help:Getting_started_with_LSL)>. Acesso em: 13 ago. 2018.

LKG – LIFELONG KINDERGARTEN GROUP. **Homepage**. MIT, Cambridge, MA, USA, 2014. Disponível em: <<http://llk.media.mit.edu/>>. Acesso em: 14 set. 2014.

LSL WIKI. **Homepage**. Disponível em: <<http://lslwiki.digiworldz.com/>>. Acesso em: 13 ago. 2018.

MATTOS, Mauro Marcelo; VAHL DICK, Adilson. Relato de uma experiência no ensino de algoritmos e programação utilizando um *framework* lúdico. *In: WORKSHOP DE AMBIENTES DE APOIO À APRENDIZAGEM DE ALGORITMOS E PROGRAMAÇÃO, 2., 2008. Anais [...]. [S.l.: s.n.], 2008.*

MIRANDA, Fábio Neves de. **Facebook e algoritmos: um estudo de caso do uso do Facebook como suporte pedagógico na disciplina de algoritmos e técnicas de programação do curso de sistemas de informação em IES**. Disponível em: <[http://www.faminasbh.edu.br/upload/downloads/20130627151026\\_83686.pdf](http://www.faminasbh.edu.br/upload/downloads/20130627151026_83686.pdf)>. Acesso em: 20 jun. 2014.

OPENSIM. **Scripting languages**. USA, 2012. Disponível em: <[http://OpenSimulator.org/wiki/Scripting\\_Languages](http://OpenSimulator.org/wiki/Scripting_Languages)>. Acesso em: 9 ago. 2018.

PAPERT, Seymour M. **LOGO: computadores e educação**. São Paulo: Brasiliense, 1985.

PAPERT, Seymour M. **Mindstorms: children, computers, and powerful ideas**. New York: Basic Books, 1980.

PETRY, Patrícia Gerent. **Um sistema para o ensino e aprendizagem de algoritmos utilizando um companheiro de aprendizagem colaborativo**. 2005. 94 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2005.

PROULX, Viera K. Programming patterns and design patterns in the introductory computer science course. **ACM SIGCSE Bulletin**, v. 32, n. 1, p. 80-84, Mar. 2000.

RESNICK, Mitchel *et al.* Scratch: programming for all. **Communications of the ACM**, v. 52, n. 11, p. 60-67, Nov. 2009.



SANTOS, Rodrigo Pereira dos; COSTA, Heitor Augustus Xavier. Análise de metodologias e ambientes de ensino para algoritmos, estruturas de dados e programação aos iniciantes em computação e informática. **INFOCOMP – Journal of computer science**, v. 5, n. 1, p. 41-50, 2006.

SCAICO, Pasqueline Dantas; LIMA, Anderson Alves de; SILVA, Jefferson Barbosa Belo da; AZEVEDO, Sílvia; PAIVA, Luiz Fernando; RAPOSO, Ewerton Henning Souto; ALENCAR, Yugo; MENDES, João Paulo. Programação no ensino médio: uma abordagem de ensino orientado ao design com Scratch. *In: WORKSHOP DE INFORMÁTICA NA ESCOLA – WIE*, 18., 2012, Rio de Janeiro. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2012.

SCRATCH. **Acerca do Scratch**. MIT, Cambridge, MA, USA, 2014. Disponível em: <<https://scratch.mit.edu/about>>. Acesso em: 18 jul. 2014.

SGOBBI, Fabiana Santiago; TAROUÇO, Liane Margarida Rockenbach; REATEGUI, Eliseo. The pedagogical use of the Internet of Things in virtual worlds to encourage a behavior change in obese individuals. *In: IEEE INTERNATIONAL CONFERENCE ON INTERNET OF THINGS, 2017*, Exeter, UK. **Proceedings [...]**. USA: Institute of Electrical and Electronics Engineers – IEEE, 2017. v. 10, p. 676-682.

SILVA, Ítalo Fernandes Amorim da; SILVA, Ivanda Maria Martins; SANTOS, Marizete Silva. Análise de problemas e soluções aplicadas ao ensino de disciplinas introdutórias de programação. *In: JORNADA DE ENSINO, PESQUISA E EXTENSÃO – JEPEX*, 9., 2009, Recife. **Anais [...]**. Recife: UFRPE, 2009. Disponível em: <<http://www.eventosufrpe.com.br/jepex2009/cd/resumos/R1479-1.pdf>>. Acesso em: 18 abr. 2014.

SILVA, R. F.; RAABE, André Luís Alice. Modelagem cognitiva dos problemas de aprendizagem de algoritmos. *In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO – SBIE*, 19., 2008, Fortaleza. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação – SBC, 2008.

UNITED KINGDOM. Department for Education. Statutory Guidance. **National curriculum in England**: computing programmes of study. Published 11 Sept. 2013. Disponível em: <<https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>>. Acesso em: 10 set. 2014.

VALENTE, José Armando. **Informática na educação**: instrucionismo x construcionismo. NIED/UNICAMP, 1997. Manuscrito não publicado.

VAN MERRIËNBOER, Jeroen J. G.; KIRSCHNER, Paul A. **Ten steps to complex learning**. New York: Routledge, 2013.

WANG, Tsung Juang. Educating avatars: on virtual worlds and pedagogical intent. **Teaching in higher education**, London, v. 16, n. 6, p. 617-628, May/Dec. 2011.

WING, Jeannette M. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.

# ANEXOS

## ANEXO 1 - VARIÁVEIS

Tipos de variáveis usadas para os diferentes objetivos que se pretendem alcançar:

- integer: armazena números inteiros (1, 2, 5 e 7);
- float: guarda números reais (1.235 ou 10.0001);
- string: armazena texto (“Olá Avatar!”);
- key: variável especial usada para identificar características próprias de elementos do Mundo Virtual, como, por exemplo, um avatar, um objeto ou uma textura;
- vector: uma variável composta por três números de tipo float (<1.0,1.0,1.0>);
- rotation: variável composta por 4 números de tipo float, usada para dar movimento aos objetos.

## ANEXO 2 - OPERADORES

Para realizar cálculos e operações lógicas, o LSL utiliza estes operadores:

$X=Y$  : X **recebe** o valor de Y;

$X==Y$  : a comparação é verdadeira se o valor de X é **igual** ao valor de Y;

$X!=Y$  : a comparação é verdadeira se o valor de X é **diferente** do valor de Y;

$X < Y$  : a comparação é verdadeira se o valor de X é **menor** do que valor de Y;

$X > Y$  : a comparação é verdadeira se o valor de X é **maior** do que valor de Y;

$X<=Y$  : a comparação é verdadeira se o valor de X é **menor ou igual** ao valor de Y;

$X>=Y$  : a comparação é verdadeira se o valor de X é **maior ou igual** ao valor de Y;

$+$  : mais (soma);

$-$  : menos (subtração);

$*$  : multiplicação;

$/$  : divisão;

$++$  : incremento de 1 em 1;

-- : decremento de 1 em 1;

% : módulo (resto de divisão);

&& : operador lógico e (and)

|| : operador lógico ou (or)

## ANEXO 3 - COMANDOS LSL

O formato do texto podem ser alteradas com a função `llSetText`, que possui a sintaxe [`llSetText( string text, vector color, float alpha );`]. O primeiro parâmetro, `text`, define o texto que vai ser apresentado acima do objeto. O parâmetro `color` indica a cor do texto, sendo formado por 3 valores que representam a formatação RGB (Red, Green, Blue), e a intensidade da cor é definida pelo número, que varia de 0.0 a 1.0; desta forma, `<1.0, 0.0, 0.0>` = vermelho, `<0.0, 1.0, 0.0>` = verde, `<0.0, 0.0, 1.0>` = azul, `<0.0, 0.0, 0.0>` = preto e `<1.0, 1.0, 1.0>` = branco. O parâmetro `alpha` indica a transparência do texto, variando de 0 (completamente transparente) a 1 (completamente opaco). O Código 3.1 apresenta uma aplicação do `llSetText`.

**Código Anexo 3.1: Função `llSetText()`.**

```
touch_start(integer total_number)
{
    // mostra o texto "NOME" em vermelho e sólido
    llSetText("NOME", <1.0,0.0,0.0>, 1.0);
}
```

De forma semelhante, a função `IISetColor` muda a cor do objeto nas respectivas faces do objeto. A função possui a seguinte composição: `[ IISetColor( vector color, integer face ); ]`. O parâmetro *color* é semelhante ao da função `IISetText`. Já o parâmetro *face* é o “lado” da figura geométrica que será alterado. O Código Anexo 3.2 mostra a função `IISetColor`.

**Código Anexo 3.2: Função `IISetColor()`.**

```
touch_start(integer total_number)
{
    // altera todas as faces do objeto para verde
    IISetColor(<0.0,1.0,0.0>, ALL_SIDES);
}
```

Além da cor, a posição do objeto pode ser verificada e alterada pelas funções `IIGetPos` e `IISetPos`, respectivamente. A função `IIGetPos()` apresenta o vetor com as coordenadas do objeto na região em que ele se encontra. Um retorno para a função `IIGetPos()` poderia ser `<123.5, 25.3, 86.2>`, em que  $X=123.5$ ,  $Y=25.3$  e  $Z=86.2$ .

Já a função `IISetPos(vector pos)` coloca o objeto em uma nova posição pelo parâmetro o qual é um vetor com as coordenadas `<X,Y,Z>`. Código Anexo 3.3 mostra essas funções.

### Código Anexo 3.3: Funções `llGetPos()` e `llSetPos()`.

```
touch_start(integer total_number)
{
// escreve a posição do objeto capturada com llGetPos()
llSay(0,"A posição do objeto é "+(string)llGetPos());
// a partir da captura da posição do objeto,
// altera sua posição em 1 metro com llSetPos()
llSetPos (llGetPos ()+<0, 0, 1>);
llSay (0,"A nova posição é "+(string)
llGetPos());
}
```

Muitas vezes, precisa-se de que determinada ação seja realizada de tempos em tempos. A linguagem LSL permite fazer isto com a função `llSetTimerEvent`. Essa função faz com que o evento do temporizador seja acionado, no máximo, uma vez a cada X segundos. Assim:

```
llSetTimerEvent( float sec );
```

O parâmetro *sec* é a quantidade de segundos que o evento deve esperar. Por sua vez, a função `llFrnd(float mag)` retorna um valor do tipo *float* entre 0,0 e o número informado. Neste exemplo, ele irá variar entre 0 e 1. O Código Anexo 3.4 mostra um exemplo destas funções.

#### Código Anexo 3.4: Função `llSetTimerEvent`.

```
default
{
    state_entry ()
    {
        // define em 5 segundos o tempo de espera
        llSetTimerEvent(5);
    }

    timer ()
    {
        float x= llFrand(1); // gera um nro aleatorio para x
        float y= llFrand(1); // gera um nro aleatorio para y
        float z= llFrand(1); // gera um nro aleatorio para x
        llSay(0, "x= "+ (string)x +" y= "+ (string)y + " z= "+ (string)z);
        llSetColor (<x, y, z>, ALL_SIDES);
    }
} // fim de default
```

Como descrito anteriormente, o *OpenSim* possui um grande número de canais para comunicação. Esses canais são úteis quando se deseja que um objeto execute certas instruções sem que seja necessário tocá-lo. A função `llListen` faz com que um objeto “escute” os canais de mensagens. Sua sintaxe é: `llListen(integer channel, string name, key id, string msg)`, em que *channel* é o canal que o objeto deve escutar, *name* identifica um avatar ou objeto específico pelo nome, *id* identifica um avatar ou objeto específico pela identificação numérica e *msg* é a mensagem recebida. A função `llListen` executa o evento `Listen`, cujas instruções LSL são codificadas.



Para enviar uma mensagem por um canal no *viewer*, é necessário digitar o canal para o qual se quer enviar a mensagem, seguido da mensagem, assim, [/33 vermelho], por exemplo. A Figura 7.29 mostra o *chat* local, que fica no canto esquerdo inferior dos *viewers*. O Código 3.5 apresenta um exemplo de uso dos canais de comunicação.

Figura 7.29 - Local chat para comunicação no Mundo Virtual.



### Código Anexo 3.5: Função IListen().

```
default
{
    state_entry()
    {
        IISay(0, "Escutando a porta 33.");
        // escuta o canal 33 para qualquer emissor
        // com qualquer mensagem
        IISetListen(33, "", "", "");
    }

    listen(integer channel, string name, key id, string message)
    {
        IISay(0, "Cor recebida: " + message);
        if ( message == "vermelho" )
            {
                IISetColor( < 1.0, 0.0, 0.0 >, ALL_SIDES);
            }
        else if ( message == "verde" )
            {
                IISetColor( <0, 1, 0>, ALL_SIDES);
            }
        else if (message == "azul" )
            {
                IISetColor( <0, 0, 1>, ALL_SIDES);
            }
        else
            {
                IISay (0, "Informe vermelho, verde ou azul");
            }
    }
} // fim listen
} // fim default
```

Além da comunicação por canais, o *OpenSim* possui a função `IIDialog()`, a qual possibilita a criação de menus, que são caixas de diálogo apresentadas no canto superior esquerdo da tela ou inferior direito, os quais podem ser usados como recursos de seleção e fluxo de controle das ações do usuário no ambiente virtual. A função `IIDialog()` possui a estrutura: `IIDialog(key avatar, string message, list buttons, integer channel)`, em que *avatar* é o usuário que verá o menu, *message* é o texto apresentado na caixa de diálogo, *buttons* são os textos para cada um dos botões e *channel* é o canal de comunicação usado por este menu. O Código Anexo 3.6 mostra a função `IIDialog()`.

**Código Anexo 3.6: Função `IIDialog()`.**

```
// define 100 como o canal de comunicação
integer canal=100;
default
{
    state_entry ()
    {
        // escuta o canal de comunicação definido
        IIListen(canal, "", "", "");
        // ao iniciar, altera a cor do objeto para preta
        IISetColor(<0,0,0>, ALL_SIDES);
    }

    touch_start (integer total_number)
    {
        // mostra o menu para o avatar que tocar no
        // objeto através da IIDetectedKey(0)
        IIDialog(IIDetectedKey(0), "Escolha uma cor", ["Vermelho", "Verde", "Azul"], canal);
    }
}
```

```

listen (integer channel, string name, key id, string message )
{
    // escreve a cor escolhida na tela
    llSay(0, "Você escolheu: " + message);
    // muda a cor do objeto de acordo com a escolha
    if ( message == "Vermelho")
    {
        // altera a cor para Vermelho
        llSetColor(<1,0,0>, ALL_SIDES);
    }
    else if ( message == "Verde" )
    {
        // altera a cor para Verde
        llSetColor(<0,1,0>, ALL_SIDES);
    }
    else if ( message == "Azul" )
    {
        // altera a cor para Azul
        llSetColor(<0,0,1>, ALL_SIDES);
    }
} // fim listen
} // fim default

```

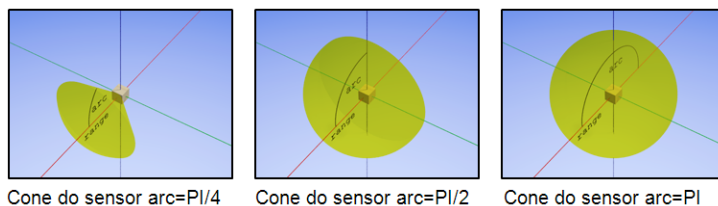
Por vezes, almeja-se que determinado *script* LSL seja executado sem que o usuário toque em um objeto. Isto pode ser feito por meio da função `llSensor()`. Sua sintaxe é: `llSensor(string name, key id, integer type, float radius, float arc)`.

O parâmetro **name** é o nome do avatar ou do objeto procurado, **id** é a identificação única do grupo, avatar ou **object** na região, **type** é a máscara de identificação para o grupo, avatar ou objeto, **radius** é o raio do centro do objeto até o limite definido (de 0.0 até 96.0 metros do Mundo

Virtual), *arc* é o ângulo máximo entre o eixo X dos objetos e o eixo X do avatar ou objeto detectáveis, variando de 0.0 até  $\pi$ . A Figura 7.30 mostra algumas opções de construção destes ângulos.

Além da detecção dentro do *radius*, é possível atribuir um *script* LSL para quando o avatar ou objeto ultrapassarem o limite da distância. O evento `no_sensor()` é responsável por essa condição. Na execução do Código Anexo 3.7, quando o avatar se aproxima a menos de 10 metros e toca no objeto, uma mensagem é apresentada, mas, se a distância for maior do que 10 metros, outra mensagem é apresentada.

Figura 7.30 - Representação dos ângulos para o parâmetro *arc*.



### Código Anexo 3.7: Função IISensor().

```
default
{
    touch_start (integer total_number)
    {
        // varre o ambiente buscando qualquer nome e id de avatar
        // para o tipo AGENT, na distância de 10 mt, com ângulo PI
        IISensor("", "", AGENT, 10, PI);
    }

    sensor (integer num)
    {
        // escreve o nome do avatar detectado a partir
        //da função IIDetectedName()
        IISay(0, "Avatar "+ IIDetectedName(0) + " detectado!");
    }

    no_sensor()
    {
        IISay(0, "Nenhum avatar detectado!");
    }
} // fim default
```

Como descrito anteriormente, o LSL permite realizar a mudança entre estados, como foi relatado no exemplo da lâmpada. O Código Anexo 3.8 mostra a transição entre estados.

**Código Anexo 3.8: Mudança de estados no LSL.**

```
// no estado default, a lâmpada inicia "desligada"
default
{
    state_entry()
    {
        ISay(0, "Lâmpada desligada.");
    }

    touch_end(integer num_detected)
    {
        ISay(0, "Ligando .....");
        state ligada;
    }
} // fim estado default

state ligada
{
    state_entry()
    {
        ISay(0, "Lâmpada ligada.");
    }

    touch_end(integer num_detected)
    {
        ISay(0, "Desligando .....");
        state default;
    }
} // fim state ligada
state ligada
```





# INFORMAÇÕES SOBRE OS AUTORES

## **Anita Raquel da Silva**

Doutoranda em Informática na Educação pelo PPGIE/UFRGS. Mestre em Informática na Educação, pelo Instituto Federal do Rio Grande do Sul (IFRS). Especialista em Informática na Educação (ESPIE2005) e em Mídias na Educação (MA3-2014), ambas realizadas no CINTED/UFRGS. Graduada em Pedagogia Magistério para Educação Infantil pela Universidade Federal do Rio Grande do Sul (2004). Fez parte da Equipe Multidisciplinar do Curso de Especialização em Mídias na Educação 4<sup>a</sup>, 2<sup>a</sup> e 1<sup>a</sup> eds. no CINTED/UFRGS. Professora na Sala I: Introdução ao Ambiente Moodle e ao Curso do curso de Pós-Graduação em Gestão Escolar 2<sup>a</sup> ed. no CINTED/UFRGS. Bolsista de Iniciação Científica PIBIC/CNPq.

## **Bárbara Gorziza Avila**

Licenciada em Matemática (UFRGS), Mestre em Educação (FACED/UFRGS) e Doutora em Informática na Educação (PGIE/UFRGS). Atualmente encontra-se no Instituto Federal Farroupilha, no setor de apoio pedagógico do campus avançado Uruguaiana. Desenvolve pesquisa na área de ensino de Programação e Matemática através do uso da robótica.

## **Clóvis da Silveira**

Licenciado em Computação (FEEVALE), Especialista em Educação a Distância (PUCRS), Mestre em Diversidade Cultural e Inclusão Social na linha de pesquisa em Linguagens e Tecnologias (FEEVALE), e Doutorando em Informática na Educação (UFRGS/PPGIE). Atualmente é professor do Centro Universitário Cenecista de Osorio, onde leciona nos cursos de Sistemas de Informação e Análise e Desenvolvimento de Sistemas na modalidade EaD. Coordenou Curso de Graduação e Pós-Graduação presencial e EaD. Tem interesse em pesquisas na área de Informática na Educação, principalmente nos seguintes temas: informática na educação, agentes conversacionais, educação a distância, inclusão digital, acessibilidade em interfaces digitais.

## **Érico Marcelo Hoff do Amaral**

Possui graduação em Ciência da Computação pela Universidade Federal de Santa Maria (2006) é mestre em Engenharia de Produção (UFSM) e doutor pelo Programa de Pós Graduação em Informática na Educação na UFRGS. Atualmente é professor do curso de Engenharia de Computação na Universidade Federal do Pampa (Unipampa). Tem experiência na área de Redes de Computadores, Gestão de Segurança da Informação, Sistemas Operacionais, Arquitetura de Computadores, Governança de TI e Informática na Educação.

## **Fabiana Santiago Sgobbi**

Graduada em Tecnologia de Processamento de dados, Licenciatura plena para docência em tecnologia de Processamento de dados, graduada em Pedagogia educacional e administrativa, pós-graduada em Psicopedagogia, Mestrado em Educação, Doutorado em Informática na Educação (UFGRS) e atualmente é Pós-Doutoranda em Informática na educação (UFRGS). Atuou como professora concursada por 13 anos no CETEEPS (Centro de Educação Tecnológica Paula Souza), atualmente é professora substituta no Instituto Federal do Rio Grande do Sul. Desenvolveu trabalho em indústria como programadora de computador na área de automação. Sua área de pesquisa é mundos virtuais e *Internet* das coisas.

## **Fabício Herpich**

Doutor em Informática na Educação no Programa de Pós-Graduação em Informática na Educação da Universidade Federal do Rio Grande do Sul (2019). Mestre em Ciência da Computação no Programa de Pós-Graduação em Informática, pela Universidade Federal de Santa Maria (2015). Especialista em Aplicações para *Web* pela Universidade Federal do Rio Grande (2015). Graduado em Sistemas para *Internet* pela Universidade Federal de Santa Maria (2013). Atualmente realiza atividades de Pós-Doutorado (PPGIE/UFRGS) desenvolvendo recursos educacionais em realidade virtual e aumentada no âmbito do Projeto AVATAR. Atua como Professor Colaborador do Programa de Pós-Graduação em Informática na Educação no Centro Interdisciplinar de Novas Tecnologias na Educação (CINTED/UFRGS).

## **Felipe Becker Nunes**

Doutor em Informática na Educação pelo Programa de Pós-Graduação em Informática na Educação (PPGIE) na Universidade Federal do Rio Grande do Sul (UFRGS). Mestre em Ciência da Computação pelo Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal (UFSM) de Santa Maria (2014). Bacharel em Sistemas de Informação na Universidade Luterana do Brasil (ULBRA - Santa Maria) no ano de 2012. Professor na Antonio Meneghetti Faculdade. Pesquisador do Grupo de Redes de Computadores e Computação Aplicada e do Grupo de Trajetórias de Aprendizagem em Hiperdocumentos Ubíquo. Tem experiência com Mundos Virtuais, Agentes Pedagógicos, AVA, Educação a Distância, Computação Móvel e Ubíqua, Contexto e Qualidade do Contexto.

## **Leandro Rosniak Tibola**

Possui doutorado em Informática na Educação e mestrado em Ciência da Computação pela UFRGS. cursou as graduações de Administração de Empresas e de Ciência da Computação pela Universidade Regional Integrada do Alto Uruguai e das Missões - Campus de Frederico Westphalen. Foi professor e coordenador do curso de Ciência da Computação da Universidade Regional Integrada do Alto Uruguai e das Missões - Campus de Frederico Westphalen. Possui interesse em desenvolvimento *Web* e móvel, gerenciamento de redes, tecnologias e modelos de gestão, liderança e motivação, empreendedorismo e startups, novos modelos de educação, jogos educacionais, laboratórios virtuais, *academic* e *learning analytics* e educação a distância.

## **Liane Margarida Rockenbach Tarouco**

Licenciada em Física (UFRGS), Mestre em Ciência da Computação (UFRGS) e Doutora em Engenharia Elétrica/Sistemas Digitais (USP). Atualmente é professora titular da Universidade Federal do Rio Grande do Sul. Atua como pesquisadora e docente junto ao Programa de Pós-Graduação em Informática na Educação onde assumiu, em 2017, a função de Coordenadora. Desenvolve também atividade docente e de pesquisa na área de Ciência da Computação, com ênfase em Redes de Computadores e em Gerência de Rede.

## **Patrícia Fernanda da Silva**

Licenciada em Ciências Exatas com habilitação integral em Matemática Física e Química (Centro Universitário Univates); Mestre em Ensino de Ciências Exatas (Centro Universitário Univates); Doutora em Informática na Educação (UFRGS) e Pós-Doutoranda (UFRGS). Atualmente participa como membro do Comitê Gestor da Comissão Especial de Informática na Educação (SBC), desenvolve atividades de pós-doutorado atuando como professora colaboradora junto ao Programa de Pós-Graduação em Informática na Educação (PPGIE - UFRGS). Desempenha atividades no Projeto AVATAR e também no Projeto AVAECIM (Ambiente Virtual de Aprendizagem Experimental em Ciências e Matemática) onde pesquisa sobre a eficácia do uso de ambientes virtuais, laboratórios reais e virtuais por adolescentes nas disciplinas de Ciências e Matemática.

# APOIO

Anderson Sousa Dufech Castilhos

Augusto Timm do Espirito Santo

Barbara Lopes Neres do Nascimento

Cassiane Gross de Vargas

Fernanda Stenert

Giovani Gheno Bombardieri

Grégori Francisco Barros

Janaina de Abreu Guimarães

# ÍNDICE REMISSIVO

## A

acomodação 50, 51, 202  
Active Worlds 22, 23  
AGATA 153, 154, 156, 157, 158, 171  
Agentes Conversacionais 60, 61, 139,  
140, 141, 142, 143, 145, 149, 153,  
157, 158, 161, 169, 173  
agentes virtuais conversacionais 60  
AIML 119, 123, 124, 140, 141, 142, 146,  
149, 150, 151, 152, 153, 156, 157  
A.L.I.C.E. 142, 146  
Alicebot 140  
ALICEbot 126, 145  
Anita Raquel da Silva 139  
aprendizagem experiencial 13, 23, 32,  
55, 257  
assimilação 50, 51, 202  
Atena 126, 129, 130, 132, 133  
ATENA 145, 157  
avatar 23, 103, 104, 105, 106, 118, 119,  
122, 125, 129, 130, 131, 132, 133,  
135, 144, 145, 146, 147, 149, 150,  
151, 152, 154, 155, 156, 157, 159,  
190, 193, 194, 202, 212, 216, 217,  
219, 220, 221, 226, 227, 228, 230,  
250, 251, 252, 253, 257, 258, 259,  
268, 272, 273, 277, 279, 280  
AVATAR 103, 105, 106, 118, 126, 144,  
145, 192, 201, 203, 217, 218,  
224, 296

## B

Bárbara Gorziza Avila 239  
BVH 133, 134

## C

carga cognitiva 32, 161  
chatbots 140, 141, 149, 150, 158, 172  
Chatbots 64  
chatterbot 118, 121, 123, 125, 130, 153,  
155, 168, 169, 170, 247, 249  
Chatterbot 124, 126  
Ciclo de Kolb 35, 55, 203  
Clóvis da Silvaiera 139  
cognição e aprendizagem 47, 199  
conceituação abstrata 35, 55, 56

## D

desenvolvimento cognitivo 44, 52

## E

engajamento 40, 61, 164, 167, 174, 175,  
185, 197, 200, 201  
Engajamento 186  
equilíbrio 51, 52  
Érico Amaral 271  
estilos de aprendizagem 56, 57  
experiência concreta 35, 55, 56, 257  
experimentação ativa 35, 55, 56, 257

## F

Fabiana Santiago Sgobbi 17, 69, 239,  
271  
Fabrício Herpich 17, 69, 271  
FastAIML 149, 150, 151, 152, 158, 171  
feedback 52, 53, 173, 263  
Felipe Becker Nunes 17, 60, 69, 207, 239  
Firestorm 22, 29

FS2LSL 196, 291, 292, 293, 295, 297

## G

Grid 70, 71, 225, 281

## H

HotPotatoes 228, 229, 252, 253, 254

## I

Imprudence 22, 29

## K

Kolb 35, 49, 55, 56, 57, 64, 203, 257

## L

Leandro Rosniack Tibola 207, 239, 271

Liane Margarida Rockenbach Tarouco  
17, 139, 179, 239, 271

Linden Scripting Language 118, 273,  
274, 276, 299

LSL Editor 276, 281, 282, 283, 284

## M

METIS 145, 146, 147, 148, 149, 152,  
162, 163, 164, 166, 169, 170, 171,  
173, 174, 175

motivação 18, 40, 58, 59, 60, 61, 143,  
145, 218, 261, 262, 263, 264,  
313, 350

multimídia 30, 39, 43, 147, 148, 150,  
151, 153, 155, 163, 179, 180, 184,  
185, 187, 188, 190, 192, 198, 199,  
200, 201, 204, 208, 216, 241, 246,  
247, 310

Mundos Virtuais Imersivos 17, 18, 19,  
22, 26, 27, 58, 60, 120, 144, 158,  
172, 207

## N

Non-Player Character 118, 136, 158,

195, 272

Non-Player Characters 31, 144, 247

NPC 31, 118, 119, 120, 121, 122, 123,  
126, 127, 128, 129, 130, 131, 132,  
133, 136, 145, 158, 159, 195, 247,  
249, 263, 272, 273, 279, 280, 282

## O

OAR 26, 107, 109

observação reflexiva 55, 56

OpenSim 24, 25, 26, 28, 29, 30, 70, 72,  
73, 74, 75, 76, 78, 79, 80, 81, 82,  
83, 84, 86, 87, 89, 90, 91, 94, 98,  
99, 100, 101, 102, 107, 108, 109,  
117, 118, 120, 121, 124, 126, 134,  
180, 189, 204, 211, 220, 221, 222,  
223, 224, 225, 228, 253, 255,  
259, 272, 273, 275, 276, 278, 279,  
281, 282, 293, 299, 313, 316, 325,  
338, 341

OpenSim Scripting Language 118, 272

Open Simulator 22

Open Wonderland 22, 24, 26, 28, 29

## P

Pandorabots 119, 120, 121, 123, 125, 142

Patrícia Fernanda da Silva 17

PHP 129, 150, 154, 210, 211, 223, 253,  
254

primis 134, 135, 189, 197, 216, 226

Program-O 119, 126, 128, 129, 130, 147

Python 154

## Q

QR Code 190, 191, 192, 196

## R

reequilibração 51, 53

## S

Sansar 22, 24, 26



ScripTastic 196, 197, 284, 285, 286,  
288, 325  
scripts 24, 25, 30, 31, 36, 71, 107, 109,  
110, 111, 116, 118, 121, 144,  
158, 196, 198, 210, 211, 216,  
217, 223, 247, 272, 273, 274,  
281, 282, 284, 285, 289, 290,  
298, 312, 316, 317, 325, 327  
Second Life 22, 23, 24, 25, 26, 28, 29,  
31, 255, 272, 273, 281, 293,  
299  
simulações 20, 41, 43, 46, 47, 48, 58,  
208, 242, 243, 244, 246, 310,  
324  
Singularity 22, 77, 78, 103, 105, 106,  
109, 110  
SLOODLE 31, 255, 256  
SQL 253  
StandAlone 70, 72, 76

## **T**

Taxonomia 32, 35, 186  
Teoria da Autodeterminação 58, 262  
Trajetória de Aprendizagem Situacional  
214

## **V**

viewer 22, 29, 30, 76, 182, 339

## **Z**

ZDP 160  
Zona de Desenvolvimento Proximal  
160