

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**MUSEU VIRTUAL DA HISTÓRIA DA  
COMPUTAÇÃO: UMA ABORDAGEM DE APOIO  
AO PROCESSO DE ENSINO-APRENDIZAGEM DE  
INTRODUÇÃO À COMPUTAÇÃO**

**TRABALHO DE GRADUAÇÃO**

**Pablo Dalsasso**

**Santa Maria, RS, Brasil**

**2013**

**MUSEU VIRTUAL DA HISTÓRIA DA COMPUTAÇÃO: UMA  
ABORDAGEM DE APOIO AO PROCESSO DE ENSINO-  
APRENDIZAGEM DE INTRODUÇÃO À COMPUTAÇÃO**

**Por**

**Pablo Dalsasso**

Trabalho de Graduação apresentado ao Curso de Ciência da Computação da  
Universidade Federal de Santa Maria (UFSM), como requisito parcial para a obtenção  
do grau de  
**Bacharel em Ciência da Computação**

**Orientadora: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Giliane Bernardi**

**Trabalho de Graduação N° 367**

**Santa Maria, RS, Brasil**

**2013**

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

A Comissão Examinadora, abaixo assinada,  
aprova o Trabalho de Graduação

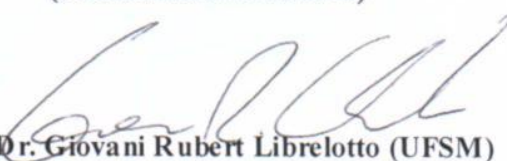
MUSEU VIRTUAL DA HISTÓRIA DA COMPUTAÇÃO: UMA ABORDAGEM  
DE APOIO AO PROCESSO DE ENSINO-APRENDIZAGEM DE  
INTRODUÇÃO À COMPUTAÇÃO

elaborado por  
**Pablo Dalsasso**

como requisito parcial para obtenção do grau de  
**Bacharel em Ciência da Computação**

COMISSÃO EXAMINADORA:

  
Prof.<sup>a</sup> Dr.<sup>a</sup> **Giliane Bernardi**  
(Presidente/Orientadora)

  
Prof. Dr. **Giovani Rubert Librelotto (UFSM)**

  
Prof.<sup>a</sup> Dr.<sup>a</sup> **Ana Trindade Winck (UFSM)**

Santa Maria, 21 de janeiro de 2014.

*“The greater our knowledge increases the more our ignorance unfolds.”*

John F. Kennedy

## **AGRADECIMENTOS**

Primeiramente agradeço ao meu pai Amilton, e minha mãe Araci, pelo apoio e por tudo que sempre fizeram por mim durante os anos de graduação, pois não mediram esforços para que eu chegasse a esta etapa de minha vida.

A minha namorada e grande amiga Anelise, pelo apoio, carinho e compreensão durante estes anos.

A minha professora orientadora, Dr<sup>a</sup>. Giliane, pelo grande auxílio, e pela disponibilidade de tempo e material durante este trabalho. E a todos os professores do curso de Ciência da Computação, que contribuíram e enriqueceram meus conhecimentos durante minha vida acadêmica.

Aos amigos, pela grande parceria, incentivo e apoio constantes.

E a todos que de alguma forma contribuíram para minha chegada nesta etapa.

# RESUMO

Trabalho de Graduação  
Ciência da Computação  
Universidade Federal de Santa Maria

## MUSEU VIRTUAL DA HISTÓRIA DA COMPUTAÇÃO: UMA ABORDAGEM DE APOIO AO PROCESSO DE ENSINO-APRENDIZAGEM DE INTRODUÇÃO À COMPUTAÇÃO

Autor: Pablo Dalsasso  
Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Giliane Bernardi

Diferentes estratégias virtuais e interativas estão sendo cada vez mais exploradas no contexto educacional, buscando aumentar o interesse do estudante, conseqüentemente, melhorando sua aprendizagem. Neste contexto, é apresentado neste trabalho um museu virtual 3D sobre a história da computação, desenvolvido através do ambiente virtual *OpenSimulator*, onde foram utilizadas suas ferramentas de modelagem e desenvolvimento de objetos 3D para a construção e animação dos artefatos do museu. Ainda, foi realizada integração do museu com o AVEA *Moodle*, permitindo ao professor a inserção de materiais e atividades do *Moodle* dentro do museu, além da possibilidade de o professor rastrear os movimentos do aluno durante uma visita. Ao final, foi feita uma avaliação de usabilidade do museu, com uma turma de alunos de Ciência da Computação e Sistema de Informações, observando-se alguns pontos possíveis de melhoria. Com a utilização de mundos virtuais 3D, pretende-se alcançar uma maior interação dos estudantes com o ambiente, neste caso os artefatos da história da computação, buscando contribuir para a construção de novas abordagens educacionais e, principalmente, na motivação dos estudantes pela área tecnológica.

**Palavras-Chave:** Museu da Computação, Mundos Virtuais, Ambiente de Aprendizagem, OpenSimulator.

# **ABSTRACT**

Undergraduate Paper  
Undergraduate Program in Computing Science  
Federal University of Santa Maria

## **VIRTUAL MUSEUM OF COMPUTING HISTORY: AN APPROACH TO SUPPORT THE TEACHING-LEARNING PROCESS OF COMPUTING INTRODUCTION**

Author: Pablo Dalsasso  
Advisor: Prof<sup>ª</sup>. Dr<sup>ª</sup>. Giliane Bernardi

Different virtual and interactive strategies are increasingly being explored in the educational context, seeking to increase student interest, consequently, improving their learning. In this context, this work presented a 3D virtual museum of the computer history, developed in the virtual environment OpenSimulator, where their tools were used for modeling and development of 3D objects and animation for the building of the museum's artifacts. Still, the museum integration with AVEA Moodle was performed, allowing the teacher to enter the Moodle materials and activities inside the museum, besides the possibility of the teacher track the movements of the students during a visitation. At the end, there was a usability evaluation of the museum, with a class of students of Computing Science and Information System, observing some possible points for improvement. With the use of 3D virtual worlds, aims to achieve greater student interaction with the environment, in this case the artifacts of the history of computing, aiming to contribute to the construction of new educational approaches and especially on the motivation of students for the technological area.

**Keywords:** Computing Museum, Virtual Worlds, Learning Environment, OpenSimulator.

## LISTA DE FIGURAS

Figura 1. <i>OpenSimulator</i> modo <i>standalone</i> e modo <i>grid</i> (Fonte: OpenSimulator, 2011). .....	20
Figura 2. Arquitetura <i>OpenSim + Moodle</i> . .....	27
Figura 3. Estrutura do Museu da Computação. ....	28
Figura 4. Entrada do Museu da Computação. ....	28
Figura 5. Balcão de Informações. ....	29
Figura 6. <i>Hall</i> de Entrada. ....	30
Figura 7. Entrada da Galeria da Pré-História da Computação. ....	30
Figura 8. Ábaco. ....	31
Figura 9. Máquina de Diferenças. ....	32
Figura 10. Entrada da Galeria da História das Linguagens de Programação .....	34
Figura 11. Linguagem <i>Assembly</i> . ....	35
Figura 12. Linguagem <i>Fortran</i> . ....	35
Figura 13. <i>Sloodle Quiz Chair</i> . ....	39



## LISTA DE QUADROS

Quadro 1. <i>Script</i> utilizado na Máquina de Escrever. ....	33
Quadro 2. <i>Script</i> utilizado no objeto referente à Linguagem C. ....	37
Quadro 3. <i>Script</i> utilizado no <i>Sloodle Button</i> . ....	40

## LISTA DE ABREVIATURAS E SIGLAS

AVEA	Ambiente Virtual de Ensino e Aprendizagem
SL	<i>Second Life</i>
LSL	<i>Linden Scripting Language</i>
OS	<i>OpenSimulator</i>
FIFO	<i>First In First Out</i>
TIC	Tecnologias de Informação e Comunicação
IHC	Interface Humano-Computador

## SUMÁRIO

1	INTRODUÇÃO .....	12
1.2	Objetivos .....	15
1.2.1	Objetivo Geral .....	15
1.2.2	Objetivos Específicos .....	15
1.3	Estrutura do Texto .....	15
2	MUNDOS VIRTUAIS 3D .....	17
2.1	<i>OpenSimulator</i> .....	19
2.1.1	A Linguagem LSL ( <i>Linden Scripting Language</i> ) .....	21
2.1.2	<i>Sloodle</i> .....	22
2.2	Mundos Virtuais 3D na Educação .....	22
2.3	Museus Virtuais e seu potencial na Educação .....	24
3	O MUSEU DA COMPUTAÇÃO .....	26
3.1	A Estrutura do Museu da Computação .....	27
3.2	A Galeria da Pré-História da Computação .....	30
3.3	A Galeria da História das Linguagens de Programação .....	33
4	INTEGRAÇÃO DO MUSEU COM O <i>MOODLE</i> .....	38
4.1	Implantação do <i>Sloodle</i> .....	38
4.2	O <i>Sloodle Tracker</i> .....	39
5	AValiação e Análise Preliminar dos Resultados.....	42
6	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS .....	46
7	REFERÊNCIAS .....	48
8	APÊNDICES .....	52

## 1 INTRODUÇÃO

O ensino de Computação é um tema que vem sendo tratado em vários locais no mundo. Em alguns países já se utiliza um currículo mínimo para o ensino de Ciência da Computação, é o que acontece em Israel, no Canadá (GAL-EZER, 1999), e nos Estados Unidos, onde o modelo de currículo investe no ensino de conteúdos que explorem a chamada Educação Imperativa, na qual mais importante do que aprender temas ligados às tecnologias está a capacidade de desenvolver o pensamento computacional (CSTA, 2005).

No Brasil, a Sociedade Brasileira de Computação (SBC), através da sua Comissão de Especialistas de Ensino de Computação e Informática e do GT3 (Grupo de Trabalho de Licenciatura em Computação), tem fomentado as discussões sobre a importância de introduzir o pensamento computacional e algorítmico nas universidades, para que se possa desenvolver competências e fomentar o interesse pela área, aumentando o número de profissionais no país (PEREIRA JUNIOR *et al.*, 2005).

Na tentativa de engajar os estudantes e fazê-los conhecer mais sobre a Ciência da Computação, muitos projetos têm procurado esclarecer os seus conceitos fundamentais, através de atividades que possam entreter e desafiar. Alguns países têm conseguido implantar um currículo mínimo nas universidades, conforme cita Ragonis (2007).

Usualmente, tanto cursos de computação quanto áreas afim possuem em seus currículos uma disciplina de introdução à área da computação, que busca explorar a história da computação e sua relação com as principais formas de utilização e aplicação de computadores, apresentando aos estudantes uma visão geral e abrangente do escopo que envolve a computação nos mais diversos segmentos da sociedade. Os principais livros na área de introdução à computação também abordam estes temas, bem como aspectos do funcionamento de um computador e linguagens de programação (FOROUZAN, 2012; FEDELI, 2003).

Uma das primeiras aulas trabalhadas junto aos estudantes trata dos aspectos da história da computação. Segundo Berry *et al.* (2011), o ensino da história da computação é amplamente aceito no que tange a ajudar os estudantes da área a

entenderem e conhecerem seu futuro campo de atuação. Ainda, alguns autores destacam que é fundamental ao estudante de computação possuir uma compreensão dos aspectos técnicos e culturais de sua área, de seu passado e presente, como fonte importante para seus estudos acadêmicos no decorrer do curso (MEDINA, 2004) (GIANGRANDI E MIROLO, 2006).

Porém, do ponto de vista educacional, o que se observa, na maioria dos casos, são apresentações expositivas, por meio de slides, que traçam a trajetória da computação nas diferentes décadas de sua evolução. Estas aulas tendem a se tornar cansativas devido ao seu caráter normalmente expositivo e a falta de interação maior dos estudantes junto ao tema, sem modificar o esquema tradicional de ensino. Uma grande dificuldade que professores enfrentam é a falta de interesse de muitos alunos que, por sua vez, alegam que as aulas são cansativas, especialmente as aulas expositivas. Neste contexto, diferentes abordagens têm sido propostas para agregar às aulas consideradas “tradicionais”, entre elas o uso de simulações, jogos computacionais e ambientes virtuais (CHENGWEI *et al.*, 2011; MATTAR E VALENTE, 2007). O uso de métodos mais eficazes poderá ajudar tanto alunos como professores em uma aula interativa, utilizando recursos de informática e a tecnologia atual poderá ser útil para despertar interesse de alunos e potencializar o trabalho dos professores.

Nesse sentido, pesquisas em informática na Educação estão sendo alvo de mudanças com a evolução da internet. Conforme Mattar e Valente (2007), ao longo da última década aumentou-se muito o interesse em mundos virtuais em três dimensões (3D) como ferramentas para melhorar o processo de ensino-aprendizagem e estimular o conhecimento e o desenvolvimento do aluno. Segundo Hodge, Collins e Giordano (2011 *apud* SILVA, 2012), o surgimento e o uso de ambientes virtuais para desenvolver e promover a aprendizagem na educação é um fenômeno novo, que está crescendo em ritmo acelerado. Para Mattar e Valente (2007 *apud* SILVA, 2012), os mundos virtuais possibilitam a realização de uma série de atividades, entre elas, atividades de cunho educacional e de treinamento. As qualidades únicas dos mundos virtuais 3D permitem oportunidades para experiências sensoriais imersivas, contextos e atividades para o aprendizado experimental, simulação, modelagem de cenários complexos, entre outros, com oportunidade de colaboração e co-criação que não podem ser facilmente experimentadas em outras plataformas. Assim, instituições acadêmicas e empresas têm explorado os benefícios de ambientes imersivos onde os participantes podem interagir

uns com os outros, bem como com o professor, mesmo que sejam a milhares de quilômetros de distância (WANKEL E KINGSLEY, 2009).

Considerando o foco deste trabalho, que trata da abordagem da história da computação em cursos de graduação, destaca-se como uma possibilidade apresentar de forma mais interativa esta evolução e artefatos de cada década de avanços tecnológicos. Neste contexto, surge a proposta de criar um museu virtual imerso em um mundo virtual 3D, onde os estudantes possam realizar uma visitação virtual pela história da computação interagindo com os diferentes artefatos de cada época.

A escolha do desenvolvimento de um museu deu-se por este ser um campo fértil para as práticas educativas, como citam Gruzman e Siqueira (2007). Segundo Marandino (2003), o papel educacional dos museus sofreu grandes modificações, sendo ampliadas as atividades voltadas à educação. Desde o século XIX, os museus adotam estratégias pedagógicas, que visam facilitar a compreensão dos objetos expostos, por arte dos visitantes. Já, a partir do século XX, os museus voltam-se para exposições de âmbito educativo, levando o público a um maior engajamento com os conceitos apresentados, através de interações com os módulos das exposições (GRUZMAN E SIQUEIRA, 2007).

Para o desenvolvimento do museu será utilizado o ambiente virtual *OpenSimulator*, que dispõe de ferramentas de criação, modelagem e animação de objetos 3D, além do uso de scripts, que torna possível fazer com que os artefatos do museu sejam interativos, tornando o ambiente mais atrativo ao aluno, aumentando sua atenção durante a visita. Também está sendo realizada a integração do museu com o ambiente virtual de ensino e aprendizagem (AVEA) *Moodle*, utilizando o módulo *Sloodle*, que permite a comunicação *OpenSimulator – Moodle*. Tal integração permitirá ao professor expor seus materiais e atividades, importados diretamente do AVEA *Moodle*, tais como apresentações de slides e questionários.

Este projeto encontra-se em desenvolvimento junto ao Grupo de Pesquisa em Redes e Computação Aplicada – GRECA, da UFSM, sendo que este trabalho de conclusão de curso tem como objetivo central o desenvolvimento da estrutura e das galerias correspondentes à pré-história da computação e a história e evolução das linguagens de programação.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Desenvolver a estrutura e duas galerias de um museu virtual 3D para o ensino de introdução à computação, aumentando o interesse do estudante pela área através da interação do mesmo com os objetos do museu.

### 1.2.2 Objetivos Específicos

- Desenvolver a estrutura do museu virtual 3D no ambiente *OpenSimulator*;
- Criar a galeria da pré-história da computação;
- Criar a galeria da história e evolução das linguagens de programação;
- Criar artefatos interativos no museu;
- Possibilitar ao estudante a realização de atividades, como questionários feitos pelo professor, dentro do ambiente do museu, através do *Sloodle*;
- Possibilitar ao professor o acompanhamento das interações realizadas pelos estudantes através do *plugin Sloodle Tracker*;
- Analisar a potencialidade de utilização do museu no processo educacional, através da aplicação do mesmo em turmas da disciplina de introdução a computação da área e áreas afins;

## 1.3 Estrutura do Texto

Este trabalho está estruturado da seguinte maneira: o Capítulo 2 apresenta uma revisão bibliográfica sobre mundos virtuais e as tecnologias envolvidas, especialmente o *OpenSimulator*, que foi utilizado no desenvolvimento do trabalho. Ainda, destaca algumas características sobre museus virtuais e a utilização dos mesmos no âmbito educacional.

No Capítulo 3 é apresentado e descrito o desenvolvimento do Museu da Computação, juntamente com sua estrutura e suas galerias. A integração do museu com

o ambiente virtual de ensino e aprendizagem *Moodle* é descrita no Capítulo 4, além da ferramenta *Sloodle*, que possibilita esta integração.

A avaliação preliminar e a análise dos resultados obtidos com esta, são apresentadas no Capítulo 5.

Por fim, o Capítulo 6 conclui este trabalho, apresentando algumas discussões e contribuições com o desenvolvimento do mesmo, juntamente com possíveis trabalhos futuros para a continuação do projeto.



## 2 MUNDOS VIRTUAIS 3D

Mundos virtuais 3D são ambientes online persistentes gerados por computador, onde os usuários podem interagir de maneira comparável ao mundo real, seja para o trabalho ou lazer, tendo em vista que a partir de uma personificação do indivíduo no mundo virtual, o usuário pode se comunicar via canal de voz, mensagens de texto e exibir expressões corporais (BRAINBRIDGE, 2010). Tais ferramentas fornecem uma interface para um mundo tridimensional, de modo que o usuário acredita estar realmente nesse mundo, passando a interagir intuitivamente com esse ambiente imersivo e dinâmico (MATTAR E VALENTE, 2007).

Segundo Schlemmer (2008), um mundo virtual 3D se torna “material” por suas representações gráficas em três dimensões. Tais mundos são o resultado da ação de seus usuários, os avatares, pois sem a ação destes o mundo não se realiza. Esta ação dos avatares sobre o mundo apresenta resultados em tempo real, ou seja, no momento em que o usuário pratica a ação, o mundo sofre a modificação e se atualiza.

Dentro de um mundo virtual o usuário é representado por um avatar, que possui a forma de uma figura estilizada de acordo com as preferências do usuário, podendo ser humana ou não. O termo avatar surgiu da mitologia hindu, que representava um corpo temporário utilizado pelos deuses para visitar a terra (MACHADO, 2002). Segundo Damer (1997), avatares são representações digitais de si mesmo, que permitem explorar mundos virtuais e quebrar todas as leis da física.

Os mundos virtuais 3D podem ser divididos em dois tipos: os de plataforma proprietária e os de plataforma aberta. Os mundos de plataforma proprietária são de uma empresa controladora que determina as condições de uso do mesmo, autorizando ou não os usuários a realizarem modificações, já os de plataforma aberta têm seus códigos fontes disponibilizados para que usuários possam realizar modificações ou correções no software (CARMO, 2013). Dentre os mundos virtuais 3D existentes, destacam-se como de plataformas proprietárias o *Second Life* e o *Active Worlds*, e como de plataformas abertas o *OpenSimulator* e o *Open Wonderland*.

Desenvolvido em 2003 pela *Linden Lab*, o *Second Life* é um dos principais mundos virtuais 3D. É um espaço onde o usuário pode ser, construir e vender o que quiser, além de poder explorar ambientes virtuais únicos, ouvir apresentações musicais ao vivo, jogar e participar da maior economia virtual do mundo (LINDEN LAB, 2013).

Tal economia é baseada no *Linden Dollar*, que pode ser comprado ou vendido por dinheiro real, que é necessário para a compra de terrenos, casas, roupas e objetos construídos por outros usuários; apesar do usuário poder entrar no mundo virtual gratuitamente é necessária a compra de um terreno para que suas construções sejam permanentes.

Para a animação dos objetos ou comunicações externas ao mundo, o *Second Life* (SL) permite a utilização de *scripts* desenvolvidos na linguagem *Linden Scripting Language* (LSL), que é uma linguagem criada pela *Linden Lab*, baseada em eventos (SECOND LIFE, 2009), com sintaxe semelhante a linguagens como C++ e Java.

O *Active Worlds* é uma plataforma de realidade virtual 3D onde os usuários podem explorar ambientes 3D construídos por outros usuários, conversar uns com os outros e construir estruturas em “terras” particulares ou públicas (ACTIVE WORLDS, 2010). Para entrar no *Active Worlds* é necessário utilizar o navegador, de *download* gratuito, e criar um avatar. O sistema de construção de objetos e estruturas é simples e de fácil utilização, tais objetos podem ser modificados ou deletados por qualquer usuário, a menos que o construtor tenha comprado o *status* de membro. Como membro, além de bloquear seus objetos para que outros usuários não editem, é possível comprar terrenos para utilização própria.

*Open Wonderland* é um conjunto de ferramentas, de código aberto, desenvolvidas totalmente em Java para a criação de mundos virtuais 3D. É completamente extensível, possibilitando aos usuários a adição de funcionalidades para criar novos mundos ou adicionar novos recursos aos mundos existentes (OPEN WONDERLAND, 2010). Apesar de não possuir ferramentas dentro do mundo para a criação de objetos, o *Open Wonderland* suporta a importação de objetos 3D de ferramentas como *3D Studio Max*, *Blender* e do *Google SketchUp*. Também é possível utilizar materiais externos, como imagens e PDFs, facilmente, bastando “arrastá-los” para dentro da janela do *Open Wonderland*.

Para o desenvolvimento deste projeto foi escolhido o mundo virtual 3D *OpenSimulator*, um ambiente 3D de código aberto, compatível com o *Second Life*. Tais características foram consideradas para a sua escolha, pois, além de ser de plataforma aberta, sua compatibilidade com o *Second Life* facilita a busca por modelos 3D e exemplos de *scripts* importantes no desenvolvimento do projeto. Na próxima seção o *OpenSimulator* será abordado com mais detalhes.

## 2.1 OpenSimulator

O *OpenSimulator* (OS), também chamado de *OpenSim*, é um servidor de Mundos Virtuais que pode ser utilizado para criar e desenvolver Ambientes Virtuais em 3D semelhantes ao SL, uma das plataformas pioneiras no desenvolvimento de mundos virtuais 3D. O *OpenSim* é uma solução de código aberto, o que oferece alta compatibilidade com SL e com protocolos de comunicação com outras plataformas (OPENSIMULATOR, 2011).

O *OpenSim* não é a única plataforma de criação de mundos virtuais desenvolvida por comunidades de código aberto. Entretanto, seu projeto se destaca não só pela grande quantidade de usuários e desenvolvedores envolvidos, mas também por possuir uma rica interface gráfica e funcionalidades de comunicação e modelagem simples e interativas (OPENSIMULATOR, 2011).

Uma das grandes vantagens deste ambiente reside na possibilidade de portar objetos desenvolvidos para o *Second Life*, pois é buscada, no seu desenvolvimento, uma compatibilidade com o ambiente comercial *Second Life*. Esta característica se torna importante considerando o grande número de ambientes e artefatos desenvolvidos no *Second Life*, que até recentemente dominava o mercado de mundos virtuais 3D.

O *OpenSimulator* pode funcionar de dois modos: *standalone* ou *grid*. No modo *standalone* um único processo cuida de todo o simulador, rodando o servidor da região e os serviços de dados no mesmo processo, o *OpenSim.exe*. No modo *grid*, os serviços de dados não fazem parte do servidor da região, eles são executados em um executável separado chamado *Robust.exe*. O *Robust* pode executar todos os serviços ou eles podem ser divididos entre várias de instâncias suas. Isso permite, caso necessário, que eles sejam executados em máquinas totalmente separadas, assim, o *OpenSim.exe* funciona apenas como servidor da região. O modo *standalone* é simples de configurar, mas é limitado a um número pequeno de usuários (OPENSIMULATOR, 2011).

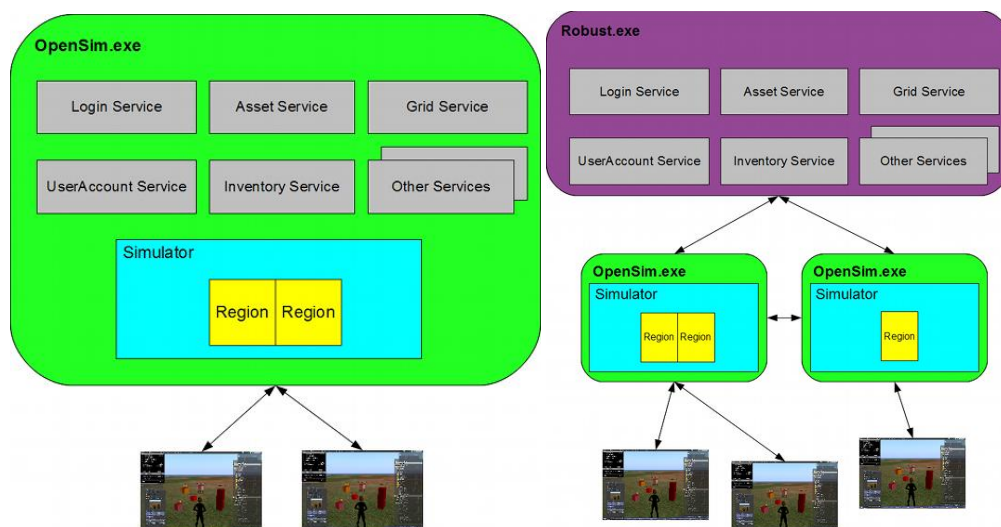


Figura 1. *OpenSimulator* modo *standalone* e modo *grid* (Fonte: OpenSimulator, 2011).

O acesso do usuário ao mundo se dá por meio de visualizadores de mundos virtuais, como o *Imprudence*, o *FireStorm* e o *Singularity*. Devido a compatibilidade do *OpenSim* com o *Second Life*, os visualizadores utilizados no SL também podem ser utilizados no OS.

Dentre as funcionalidades do *OpenSim* encontram-se os diferentes meios de comunicação disponíveis a partir do uso do áudio ou em conversas via *chat*. A construção de objetos gráficos e a disponibilização de recursos em multimídia também são funcionalidades que agregam valor ao ambiente quando este assume a forma de um Laboratório Virtual (TAROUCO, 2012). Para a criação e modelagem de objetos 3D o *OpenSim* fornece ferramentas de fácil utilização, onde o usuário constrói objetos através da edição das primitivas disponíveis (como cubos e pirâmides), também é possível importar objetos de repositórios como o *Google SketchUp* e o *OpenSim Creations*, além de arquivos de áudio e imagens.

Para a animação e interação com objetos são utilizados scripts no *OpenSim*. Tais *scripts* podem ser programados em LSL (*Linden Scripting Language*), mesma linguagem utilizada pelo *Second Life*, em OSSL (*OpenSim Scripting Language*), uma extensão da LSL, em C#, *Java Script* e *Visual Basic(.NET)*. No desenvolvimento dos *scripts* do museu proposto foi utilizada a linguagem LSL, que será detalhada na próxima seção.

### 2.1.1 A Linguagem LSL (*Linden Scripting Language*)

Dentre as linguagens disponíveis para a programação dos *scripts*, foi escolhida a LSL para o desenvolvimento do projeto. Esta linguagem permite que os usuários programem os objetos tridimensionais em torno deles, já que sem a utilização de *scripts* os objetos são imóveis e não-interativos. Adicionar um *script* a um objeto permite que ele interaja com o mundo ao seu redor e os avatares presentes no mesmo; tais *scripts* permitem que objetos como portas, carros, parques de diversão e outros dispositivos de entretenimento interativos sejam criados (HEATON, 2007).

A linguagem LSL é orientada a eventos sendo, em grande parte, semelhante às linguagens de *scripts* comuns, como *Python* e *Java Script*. Quanto aos tipos de dados suportados, a linguagem oferece tipos de dados comuns, como *integer*, *string*, *float* e *vector*, também dando suporte a tipos como *key* (muito utilizado para a comunicação entre objetos) e *rotation* (tipo necessário para a rotação dos objetos no ambiente 3D). Para o controle de fluxo são utilizados *do while*, *if else*, *for* e *state*, que é o estado do objeto. A linguagem ainda apresenta uma enorme lista de funções e constantes, como a função *llGetKey()* que retorna um identificador único do objeto do tipo *key*.

Os eventos que a linguagem oferece podem ser usados para detectar o toque do usuário (*touch\_start*), um objeto começando a se mover (*moving\_start*), a colisão com objetos ou usuários (*collision\_start*), entre outros. Ao todo, existem cerca de 40 eventos suportados pela linguagem, que podem ser utilizados em objetos e avatares dentro do mundo virtual.

No ambiente, se mais de 64 eventos estão esperando, novos eventos são descartados até *slots* livres se tornarem disponíveis, pois estes são registrados em um *unsigned integer* de 64 bits, podendo registrar um máximo de 64 eventos. Tais eventos não interrompem uns aos outros, mas ficam em uma fila FIFO (*First In First Out*) (LINDEN RESEARCH, 2009).

Para maior eficiência, a linguagem LSL compila para um *bytecode* antes da execução, por isso não é tão ineficiente como muitos acreditam, porém não possui um compilador com boa otimização. *Scripts* são limitados a um tamanho de código e dados de 16kb. O gerenciamento de *heap* não é sofisticado, um *scripter* não é capaz de apontar

o quanto *heap* está livre, ou o tamanho do maior bloco contíguo disponível, através do qual é possível determinar o quão perto a pilha e o *heap* estão de colidir (COX, 2009).

O mundo virtual *OpenSimulator* ainda apresenta a possibilidade de integração com o AVEA *Moodle*, para isso é necessária a utilização do módulo *Sloodle*, que será detalhado na próxima seção.

### **2.1.2 Sloodle**

O *Sloodle* é um sistema que integra mundos virtuais 3D desenvolvidos, tanto no *Second Life* quanto no *OpenSim*, com o ambiente virtual de ensino e aprendizagem de código aberto *Moodle*. O *Sloodle* apresenta-se como uma ferramenta com possibilidades variadas no que concerne à dinamização de uma sala de aula virtual tridimensional (OPENSIMULATOR, 2011).

Este módulo foi criado para ser integrado a ambientes virtuais 3D, com o objetivo de dinamizar o ambiente, permitindo que uma atividade disponibilizada via *Moodle* possa se juntar a uma “sala de aula 3D interativa com todos os recursos do *Moodle* disponíveis para seus estudantes no mundo virtual”.

Uma das características interessantes do *Sloodle* é o *Sloodle Tracker*, por ter a capacidade de rastrear os estudantes em suas atividades. Este é um *plugin* de um módulo do *Sloodle*, desenvolvido por uma equipe de pesquisadores da Universidade de Ulster, que permite monitorar e gravar em uma página do *Moodle* as interações ocorridas com objetos nos mundos virtuais *OpenSimulator* e *Second Life* (LIVINGSTONE, 2009). Ele foi desenvolvido para auxiliar no processo de ensino e aprendizagem proporcionando funcionalidades que permitem que as ações dos estudantes sejam monitoradas e gravadas. Ambos, estudantes e professores, têm a visão dos resultados registrados para após poder ter uma análise do que foi realizado com sucesso no ambiente.

Esta integração tem o propósito de auxiliar a utilização de mundos virtuais em ambientes educacionais, que serão abordados na próxima seção.

## 2.2 Mundos Virtuais 3D na Educação

No contexto educacional, Tecnologias de Informação e Comunicação (TICs) podem ampliar as possibilidades de interação e colaboração entre os estudantes e professores, permitindo novas práticas pedagógicas. Uma das Tecnologias de Informação e Comunicação usadas na educação são os mundos virtuais em três dimensões (3D). A utilização desses mundos provê de forma realista o acesso aos ambientes virtuais tridimensionais, oferecendo interatividade e experiências imersivas (PAVLOS E DEMETRIOS, 2011).

Os mundos virtuais 3D propiciam uma sensação de presença ao aluno, devido às características de interatividade e perspectiva em primeira pessoa. Tais mundos diferem dos que tem a interação suportada somente por textos ou diagramas, por serem uma emulação do mundo físico, fornecendo ao aluno um maior sentimento de imersão, que é fundamental para sua motivação (SCHMITT E TAROUÇO, 2008).

Esta possibilidade de imersão em ambientes como *Second Life* e *OpenSimulator* têm sido utilizada para o desenvolvimento de ambientes colaborativos de realidade virtual para uso educacional. Segundo Azevedo e Elia (2011), essas ferramentas computacionais proporcionam aos usuários novas formas de aprendizagem permitindo a construção de cenários e recursos que recriam/simulam com mais fidedignidade o mundo real. Para Osório *et al.* (2004), o paradigma 3D oferece a possibilidade de representar a informação de um modo realístico, organizando-a de uma maneira espacial e tornando sua visualização mais intuitiva por ser mais natural ao ser humano.

Ainda, Schlemmer *et al.* (2008) afirmam que a aprendizagem acontece na interação do sujeito com o objeto de conhecimento e com outros sujeitos, fazendo com que a interação seja um elemento essencial no processo educacional. Ao afirmarem isto, destacam que mundos virtuais 3D podem contribuir significativamente no processo de ensino e aprendizagem mediado por tecnologias, uma vez que estes podem trazer diferentes recursos como telepresença e avatares, diferente da abordagem convencional, que ocorre, quase que basicamente, por meio da linguagem textual.

Analisando as potencialidades descritas sobre mundos virtuais 3D em âmbito educacional, observa-se que tal recurso pode ser utilizado para potencializar a interação dos estudantes com os conteúdos educacionais, tornando os mesmos mais atrativos e

motivacionais sob o ponto de vista pedagógico. Uma das formas de se aproveitar estas possibilidades é o desenvolvimento de museus virtuais, que compreendem uma coleção de artefatos eletrônicos e apresentam como uma das vantagens em relação aos museus tradicionais a reprodução digital de objetos reais, que ainda existem ou não, oferecendo a possibilidade de observar e interagir com as obras e objetos, além de disponibilizar diversos recursos multimídia (CHENGWEI *et al.* 2011). A próxima seção destaca algumas questões relacionadas a museus virtuais e sua possibilidade de uso em âmbito educacional.

### **2.3 Museus Virtuais e seu potencial na Educação**

Museus virtuais podem ser usados no processo de ensino-aprendizagem, proporcionando um novo espaço para socialização e permitindo aos usuários visitarem, interagirem e manipularem seus artefatos de forma mais flexível. Segundo Lévy (2000), museus virtuais trazem possibilidades de serem usados em ambientes educacionais desempenhando um papel fundamental no aumento da motivação e na melhoria do processo de aprendizagem. Além de facilidades oferecidas com o uso de recursos textuais, gráficos em duas dimensões e três dimensões podem oferecer um rico ambiente para encontros sociais dos estudantes.

É possível identificar diversas iniciativas de museus virtuais dedicados a apresentar a história e evolução da computação. No entanto, a maioria destes museus estão organizados através de páginas web que exibem imagens dos artefatos, com pouca interação com os mesmos. São disponibilizadas informações sobre os objetos, na maioria das vezes de forma textual. Em alguns casos, é possível encontrar outros tipos de mídias complementando as exposições, como vídeo e áudio.

O Museu Virtual da Informática (MVI), de Portugal (ALMEIDA, 2006), apresenta a história da informática com base em páginas *web*. Dentro do museu, o estudante pode visualizar seu conteúdo através de diferentes formatos, tais como textos, imagens, sons, animações e vídeos. Na entrada ao museu o estudante dispõe de diversas opções, podendo realizar pesquisas específicas, por temas, ou através de outros sítios da rede.



Outro museu que pode ser citado é o Museu do Computador, da Universidade Estadual de Maringá<sup>1</sup>, desenvolvido pelo Departamento de Informática da universidade. O funcionamento do museu também é baseado em páginas *web*, que apresentam imagens e descrições de personalidades, descobertas e de invenções ao longo da história da computação.

Talvez o mais expressivo exemplo de museu na área de computação seja o *Computer History Museum*<sup>2</sup>, um museu físico que tem como principal objetivo apresentar artefatos e a história da era da informação, contendo mais de 1000 objetos, incluindo desde computadores da década de 40 e 50 até a história da internet e da computação móvel. É considerado o museu com o maior acervo de aparelhos eletrônicos do mundo, bem como vídeos, filmes e gravações de áudio sobre a história da computação. No *site* do museu, é possível visitar o museu virtualmente, podendo visualizar diversos artefatos presentes fisicamente no museu, bem como obter informações sobre o mesmo.

Ainda, é possível encontrar trabalhos desenvolvidos por instituições de ensino que agregam metodologias ao processo educacional com o uso da tecnologia, como o SciCentr.org que é um museu de ciências online do Centro Teórico de Cornell (CTC), um centro de pesquisa de computação interdisciplinar de alto desempenho na Universidade de Cornell Ithaca, em Nova York (EDUNY, 2013).

O que se observa nos museus analisados, é sua estrutura através de páginas *web*, com apresentação estática dos objetos, usualmente representados apenas por imagens em duas dimensões e textos explicativos acompanhando os mesmos. Poucas iniciativas possuem integração com algum tipo diferente de mídia, como vídeo, animações, simulações, entre outros. Ainda, não possuem qualquer tipo de integração com ambientes virtuais de ensino e aprendizagem, não são construídos em ambientes 3D e não privilegiam a interação entre seus visitantes ou com os artefatos do museu.

No próximo Capítulo será abordado o desenvolvimento do museu, o qual é tratado neste projeto.

---

<sup>1</sup> Museu do Computador - <http://www.din.ue.br/museu>

<sup>2</sup> *Computer History Museum* - <http://www.computerhistory.org/>

### 3 O MUSEU DA COMPUTAÇÃO

O museu virtual 3D abordado neste trabalho tem como finalidade auxiliar o processo de ensino-aprendizagem de disciplinas de introdução à computação dos cursos da área. Com esta finalidade, o museu foi desenvolvido focando na interação do estudante com os artefatos do museu, fazendo deste ambiente virtual algo mais atrativo, prendendo a atenção do estudante durante a visita.

Primeiramente, foi realizada a instalação do ambiente virtual *OpenSimulator*, sendo que detalhes desta instalação podem ser vistos no apêndice A. Para a escolha dos conteúdos e objetos a serem inseridos no museu, foi feito um levantamento de requisitos, através de pesquisas nos programas das disciplinas de introdução à computação de universidades<sup>34</sup>, *sites* de museus virtuais vistos na seção anterior e livros de introdução à computação, como os livros “Ciência da Computação: Uma visão abrangente” de J. Glenn Brookshear, “Introdução à Informática” de Capron e Johnson e “Fundamentos da Ciência da Computação” de Forouzan e Mosharraf; tais fontes também foram utilizadas na criação dos artefatos, e são de uso livre. Através desta pesquisa foi definida a divisão do museu em quatro galerias, que apresentarão a pré-história da computação, as gerações dos computadores, as personalidades na história da computação e a evolução das linguagens de programação.

Após, foi realizado o projeto arquitetural do museu, para organizar fisicamente a disposição das galerias e salas do mesmo, assim como a disposição de seus artefatos, conteúdos e atividades propostas. Esta organização física tomou como referência os museus virtuais analisados, bem como *layouts* de alguns museus reais (físicos), como o Museu do Ar e Espaço de Washington<sup>5</sup> e o Museu de História Natural de Londres<sup>6</sup>, com proposta semelhante à deste projeto.

---

<sup>3</sup> UFSM - <http://www.inf.ufsm.br/index/graduacao/cc/ELC%201085%20PRG%20Introducao%20a%20Computacao%20A.pdf>

<sup>4</sup> USP - <https://uspdigital.usp.br/jupiterweb/obterDisciplina?sglid=SSC0101&codcur=55041&codhab=0>

<sup>5</sup> Museu do Ar e Espaço de Washington - <http://airandspace.si.edu/>

<sup>6</sup> Museu de História Natural - <http://www.nh.m.ac.uk/>

Ainda, foi definida a integração do museu virtual com ambiente virtual de ensino e aprendizagem *Moodle*, através do módulo *Sloodle*, de forma a permitir que o professor, ao final da visitação possa trazer atividades complementares aos estudantes, integrando o museu ao ambiente educacional. Esta integração permite a utilização de objetos, dentro do *OpenSimulator*, para exibição de conteúdos existentes no *Moodle*. A mesma é representada, no museu, através do que foi denominada sala de estudos, sendo que é previsto que a mesma seja acessada pelos estudantes ao final da visitação. A arquitetura do sistema com a integração do *Moodle* pode ser vista na figura 2.

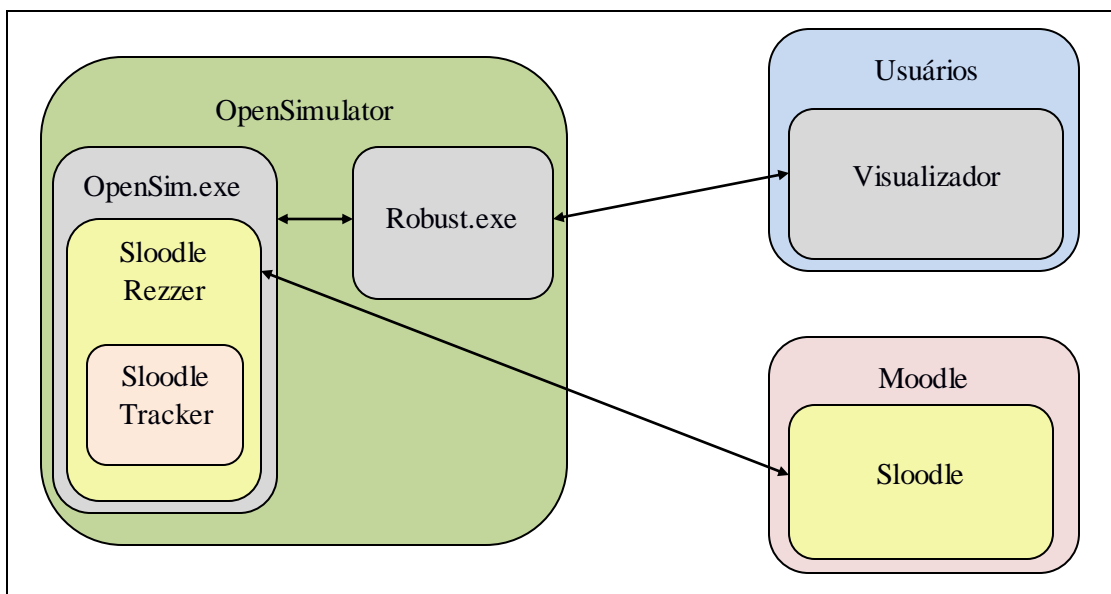


Figura 2. Arquitetura *OpenSim* + *Moodle*.

Nas próximas seções será abordado com mais detalhes o desenvolvimento da estrutura do museu e das galerias da pré-história da computação e da evolução das linguagens de programação, que são os objetivos principais deste trabalho.

### 3.1 A Estrutura do Museu da Computação

A estrutura do museu (figura 3) está dividida em um hall de entrada, quatro galerias de exposições e duas salas. No *hall* de entrada, o estudante tem acesso a um mapa do museu para orientar-se e a um ambiente para a visualização de um vídeo introdutório sobre o museu, apresentando sua estrutura e suas principais atrações.

Também é oferecida a possibilidade de se teletransportar direto para uma determinada galeria ou sala de sua preferência, clicando sobre o mapa da entrada.

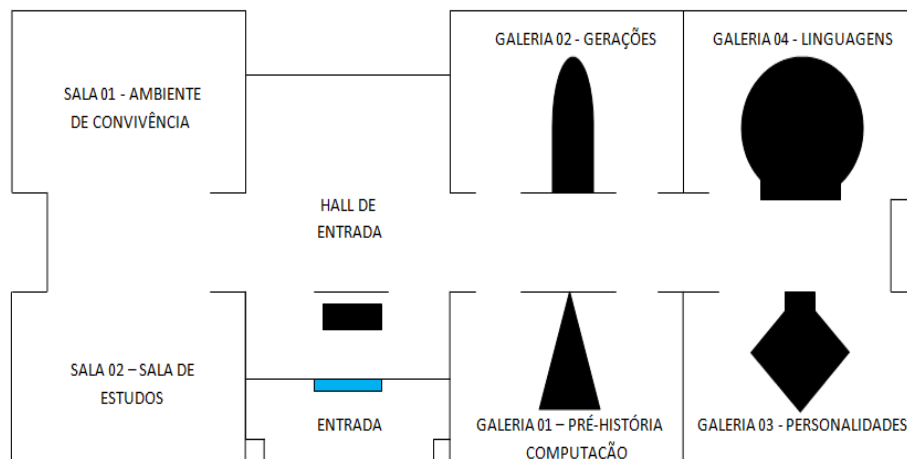


Figura 3. Estrutura do Museu da Computação.

A estrutura foi desenvolvida utilizando as ferramentas de criação de modelagem de objetos 3D do *OpenSim* e seu banco de texturas, além da utilização de *scripts* para algumas partes interativas da estrutura, como a porta de entrada (figura 4), que é acionada com a presença próxima de algum avatar.



Figura 4. Entrada do Museu da Computação.

O estudante, ao entrar no museu, pode dirigir-se até o balcão de informações (figura 5) e interagir com um dos guias virtuais do museu. Os mesmos são representados por avatares de *software* inteligentes que tem como objetivo comunicarem-se com os estudantes potencializando a interação e colaboração durante a visita. O desenvolvimento dos agentes não faz parte do escopo deste trabalho, portanto não será apresentado.

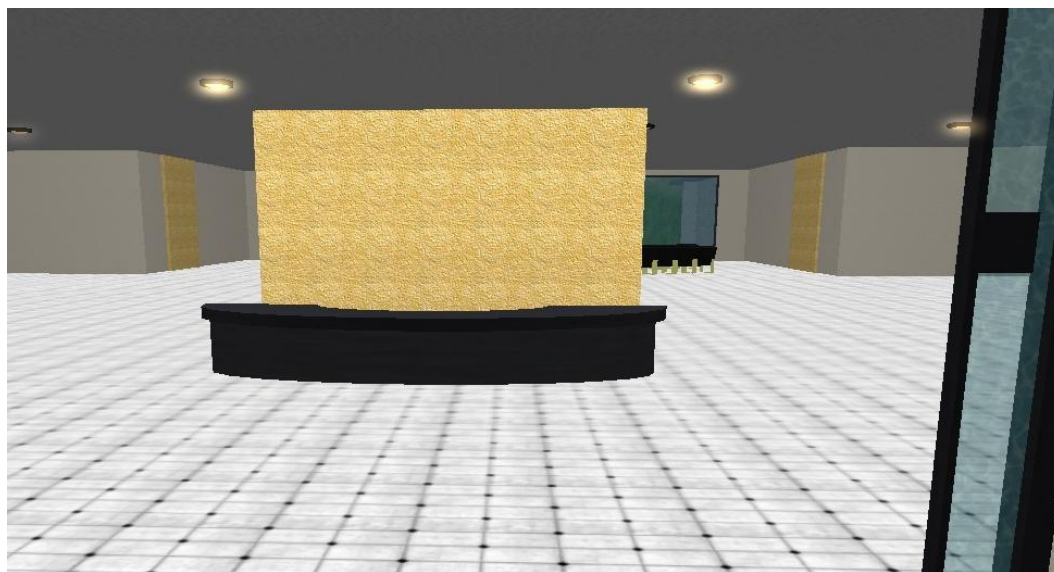


Figura 5. Balcão de Informações.

A utilização de *scripts* também foi necessária para o desenvolvimento do *hall* de entrada (figura 6), para dar a possibilidade ao aluno de reproduzir o vídeo introdutório sobre o museu e de sentar-se. Nesta parte do museu foram utilizados objetos importados do repositório 3D *OpenSim Creations*<sup>7</sup>, como as cadeiras e a estrutura da televisão onde o vídeo é reproduzido.

<sup>7</sup>

OpenSim Creations - <http://opensim-creations.com/>

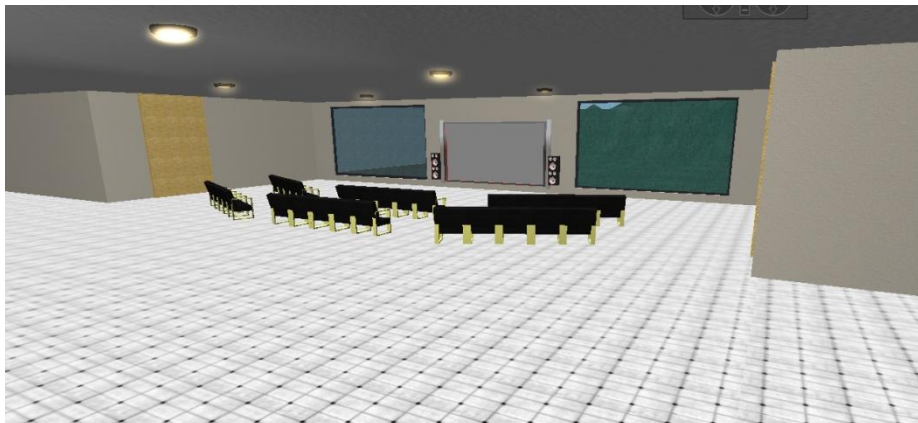


Figura 6. *Hall* de Entrada.

### 3.2 A Galeria da Pré-História da Computação

A primeira galeria do museu é a da pré-história da computação, onde se encontram vários objetos de diferentes tempos, que representam as primeiras invenções humanas para contagem, exibição e manipulação de números, palavras, sons e imagens. Para o desenvolvimento desta galeria foram utilizados os museus virtuais pesquisados, dos quais também foram utilizadas as informações acerca de cada objeto e as texturas dos mesmos. Uma visão geral desta galeria pode ser observada na figura 7.



Figura 7. Entrada da Galeria da Pré-História da Computação.

Os artefatos da galeria são interativos, apresentando animações, movimentos e sons que simulam seu funcionamento; além de possuírem uma placa com seu nome, que ao ser clicada, exibe a descrição do artefato em questão. Todos os objetos da galeria

foram modelados utilizando as ferramentas do próprio *OpenSim*, além de texturas obtidas no sites dos museus e livros citados anteriormente, e do próprio banco de texturas do *OpenSim*.

Um dos artefatos da galeria é o ábaco (figura 8). Para a interação do usuário com o objeto foi utilizado um *script* em LSL que funciona da seguinte maneira: ao clicar no objeto, o usuário aciona um evento que envia uma mensagem para o objeto que contém a animação. O objeto que recebe a mensagem possui duas texturas, uma invisível e outra com uma animação, e outro *script*. A animação é uma imagem .gif animada<sup>8</sup> que reproduz o funcionamento do ábaco.

O objeto inicialmente está configurado com a textura invisível, mas ao receber a mensagem do artefato ábaco, a textura da animação é setada. Após 10 segundos, é dado o comando para pausa na reprodução da animação e resetada a textura do objeto para a textura invisível.

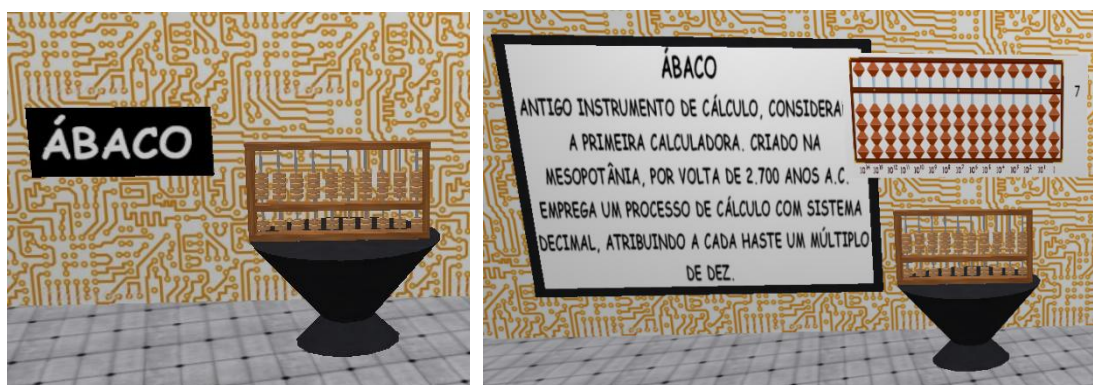


Figura 8. Ábaco.

Os artefatos referentes à escrita cuneiforme e à televisão possuem funcionamento semelhante ao do ábaco, cada um disparando a sua própria animação condizente com o artefato.

Outro artefato da galeria é a máquina de diferenças (figura 9). Quando o estudante interage com o artefato, este reproduz um movimento, simulando o seu funcionamento. Outros artefatos da galeria apresentam funcionamento semelhante, como a régua de cálculo e a relógio de sol; cada um reproduzindo seus movimentos específicos.

<sup>8</sup> <http://www.ptc.dcs.edu/Moody/comphistory/cavemanwriting.gif>

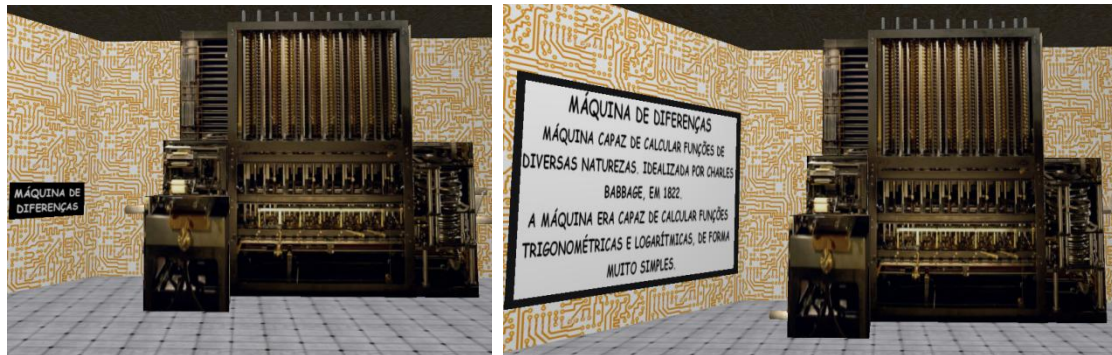


Figura 9. Máquina de Diferenças.

Alguns dos artefatos da galeria também reproduzem sons; a máquina de escrever, por exemplo, além de reproduzir movimentos que simulam seu funcionamento, dispara um arquivo de áudio referente ao som de alguém pressionando suas teclas. Tal simulação é feita utilizando-se o script do quadro 1. Primeiramente, o *script* executa o arquivo de som denominado “*typewriter*”, então envia uma mensagem pelo canal -002 para o objeto que representa a folha da máquina; este realiza um movimento de rotação enquanto o som é reproduzido, simulando a ação de digitar na máquina de escrever.

```

default {
    touch_start( integer vIntNul ){
        llPlaySound("typewriter", 1.0);
        llRegionSayTo(id1, -002, "GO!");
    }
}

...

default {
    state_entry() {
        llListen(-002, "", NULL_KEY, "");
        llSay(0, llGetKey());
    }
    listen(integer channel, string name, key id, string message) {
        rot = llGetRot();
        drot = llEuler2Rot( <-5.0, 0.0, 0.0> * DEG_TO_RAD );
        integer i = 0;
    }
}
  
```



```
while(i < 25){
    llSetPrimitiveParams([PRIM_ROTATION, drot * llGetRot()]);
    i++;
}
llSetPrimitiveParams([PRIM_ROTATION, rot]);
}
```

Quadro 1. *Script* utilizado na Máquina de Escrever.

Além dos artefatos citados, a galeria apresenta ainda os objetos referentes ao tear programável, que é representado por um cubo que gira, mostrando diferentes imagens do artefato; o telefone *bell*, que reproduz o som de um telefone; o diodo, que acende e apaga com a interação do estudante; além dos artefatos referentes à máquina de pascal, ao *arithometer* e ao telegrafone. Na próxima seção será apresentada a galeria com a história das linguagens de programação.

### 3.3 A Galeria da História das Linguagens de Programação

A galeria das linguagens de programação tem como objetivo disponibilizar um breve histórico sobre as principais linguagens que fazem parte da evolução das linguagens de programação, vistas nos livros de introdução já citados e estudadas durante o curso de Ciência da Computação da UFSM, apresentando algumas características principais e o foco de utilização das mesmas, além de possibilitar ao estudante a compilação, ou interpretação, e execução de um pequeno exemplo da linguagem. Tais exemplos são interativos e propiciam uma ideia básica da estrutura e sintaxe da linguagem, onde o estudante, além de compilar e executar o código exemplo, interage com o programa durante a sua execução. A figura 10 apresenta a entrada da galeria.



Figura 10. Entrada da Galeria da História das Linguagens de Programação.

Os objetos da galeria possuem um modelo padrão, semelhante a um computador, que foi modelado com as ferramentas do *OpenSim*, mudando apenas os exemplos e os textos para cada linguagem diferente. Estes objetos possuem duas telas, uma representando um editor de texto, onde é exibido um exemplo de código da linguagem, e outro representando o terminal do computador, onde é simulada a execução do código compilado. Também estão presentes nos objetos dois botões, um para compilar o exemplo e outro para executá-lo; caso o estudante tente executar o código antes de compilar, o objeto enviará uma mensagem pelo chat avisando que o código não está compilado. Os textos foram retirados dos livros de introdução à computação e museus citados nas seções anteriores, assim como a base para os exemplos.

Todas as linguagens da galeria apresentam exemplos de códigos comentados e interativos, além de uma placa interativa com o nome e o histórico da linguagem. O primeiro objeto da galeria é referente à linguagem *Assembly* (figura 11), o código-exemplo utilizado no objeto serve para imprimir na tela a mensagem “*Hello World*”; após compilar e executar o código, a mensagem é exibida na tela referente ao terminal do computador, simulando a execução real do código.

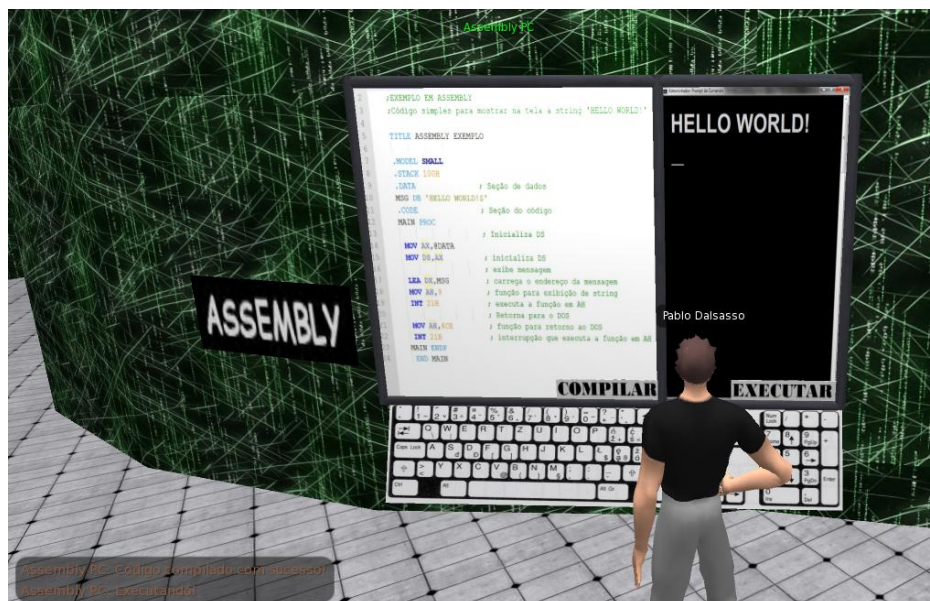


Figura 11. Linguagem *Assembly*.

Para a linguagem *Fortran* (figura 12), por exemplo, foi utilizado um código que informa ao usuário se o número digitado por ele é par ou é ímpar. O estudante visualiza esse código, que possui comentários explicando seu funcionamento, e pode compilá-lo e executá-lo, simulando a execução de um programa criado a partir daquele código. Durante a execução, o estudante também interage com o programa pelo chat do *OpenSim*, desempenhando o papel de usuário do mesmo. Ao final da execução o programa é reiniciado, necessitando ser compilado e executado pelo próximo estudante para nova simulação.



Figura 12. Linguagem *Fortran*.

Para o desenvolvimento da simulação dos exemplos, foram utilizados *scripts* em LSL que fazem a comunicação entre as partes dos objetos e a comunicação com o estudante, através do *chat* do *OpenSim*. No objeto referente à linguagem C, por exemplo, um dos *scripts* utilizados tem um trecho apresentado no quadro 2. Com este *script* o objeto é setado para “escutar” o canal -004, que é por onde as partes do objeto se comunicam. Quando o estudante clica no botão “executar”, este manda uma mensagem para o *script* anterior com o *id* do estudante; então é setada a textura no painel de execução com a mensagem “Entre com três números:” e o objeto é setado para “escutar” o canal do *chat* global. Por meio deste canal o objeto recebe os três números digitados pelo estudante no *chat* e verifica qual é o maior, escolhendo a textura correta para ser exibida. Então, após 20 segundos desde a execução do exemplo, a textura e as configurações do objeto são resetadas às iniciais, estando pronto para nova interação com os estudantes.

```

default {
    state_entry() {
        llListen(-004, "", NULL_KEY, "");
    }

    listen(integer channel, string name, key id, string message) {
        if(idA != id) {
            if(message == "ReIn") {
                llSetTexture("DOS", ALL_SIDES);
                cont = 0;
            } else {
                llSetTexture("CResult1", ALL_SIDES);
                idA = message;
                aux = llListen(0, "", idA, "");
            }
        }

        ...

        llSetTexture("CResult2", ALL_SIDES);
    } else { if(n2 > n3 && n2 > n1) {
        llSetTexture("CResult3", ALL_SIDES);
    } else { if(n3 > n1 && n3 > n2) {
        llSetTexture("CResult4", ALL_SIDES);
    } else {
        llSetTexture("CResult5", ALL_SIDES);
    } } }

```

```
cont = 0;  
idA = "op";  
llListenRemove(aux);
```

Quadro 2. *Script* utilizado no objeto referente à Linguagem C.

Além das linguagens *Fortran* e *C* já citadas, na galeria estão representadas as linguagens *Assembly*, *Lisp*, *Pascal*, *C++*, *Haskell* e *Java*. Elas estão organizadas cronologicamente, seguindo suas datas de criação existentes nas fontes pesquisadas. Cada uma das linguagens apresenta um exemplo diferente, mas com funcionamento semelhante, onde as partes dos objetos se comunicam entre si e com o estudante, exibindo as texturas adequadas de acordo com a simulação da execução do código exemplo. No próximo Capítulo será detalhada a integração do museu virtual com o AVEA Moodle.

## 4 INTEGRAÇÃO DO MUSEU COM O *MOODLE*

O *Moodle* é um ambiente virtual de ensino e aprendizagem (AVEA) *web* livre e gratuito, utilizado por professores para disponibilizarem seus materiais de forma online para os alunos (MOODLE, 2007). Por ser um dos ambientes mais utilizados no mundo, conforme citam Rodrigues e Brandão (2009), e por ser o AVEA adotado formalmente pela UFSM como ambiente de apoio ao ensino presencial e a distância, nos cursos de graduação e pós-graduação da universidade, surgiu a ideia de tentar integrá-lo com o museu.

Para tal integração foi utilizado o *Sloodle*, permitindo ao professor a possibilidade de importar seus materiais e atividades criadas no Moodle para dentro do museu. Assim, o estudante pode visualizar os materiais do professor e realizar suas atividades, sem a necessidade de sair do ambiente do museu.

### 4.1 Implantação do *Sloodle*

Para a implantação do *Sloodle* foi instalada a sua versão 2.1 e seus componentes dentro da pasta destino do Moodle. Após, foi baixado e importado para dentro do ambiente do museu o *Sloodle Rezzer 2.1*, objeto através do qual é feita a importação dos materiais do Moodle em forma de objetos para dentro do museu. Por exemplo, um conjunto de *slides* ou imagens postadas pelo professor no Moodle é importado pelo *Sloodle* como o objeto *Presenter*, que tem a aparência de um quadro e permite a exibição dos *slides* de forma interativa, trocando de *slide* ao toque.

Outra possibilidade é a importação de questionários do Moodle, que são acessíveis dentro do museu pelo objeto *Quiz Chair* (figura 13), que é uma cadeira onde o estudante sentará e responderá ao questionário proposto pelo professor. Ao final, pode ser mostrada a pontuação do aluno e quais foram seus erros e acertos. O resultado do questionário é importado para o Moodle, onde o professor tem acesso a ele.

Para a geração deste relatório, primeiramente o estudante deve atribuir o seu usuário do Moodle ao avatar que ele está utilizando para a visitação; este processo é feito através do objeto *Sloodle RegEnroll Booth*, uma cabine que permite o estudante

acessar o *site* do *Moodle* para entrar com o seu usuário e senha. Assim, quando o aluno realizar as atividades, os resultados serão atribuídos no *Moodle* ao seu usuário.

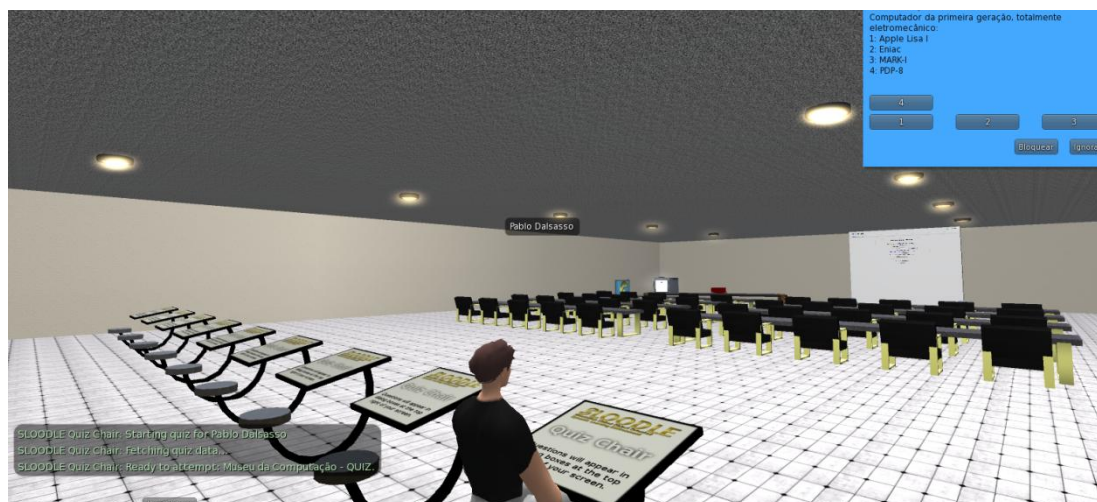


Figura 13. *Sloodle Quiz Chair*.

Tais objetos estão dispostos na sala de estudos do museu, onde o aluno visualizará as apresentações de *slides* criadas pelo professor e outros materiais, além de responder questionários referentes ao conteúdo visto durante a visita às galerias do museu.

## 4.2 O Sloodle Tracker

Outra ferramenta do *Sloodle* é o *Sloodle Tracker*, que permite que a interação com os objetos do Museu seja monitorada e gravada em uma página *Moodle*. Esta ferramenta possibilita um maior controle do professor sobre as atividades e os objetos educacionais com os quais o estudante interage ao longo da visitação, principalmente em visitas não acompanhadas pelo professor. Seu funcionamento baseia-se na interação do estudante com os objetos do museu e na presença de seu avatar em suas galerias.

Para a detecção da interação do estudante, o *Sloodle Tracker* disponibiliza o objeto *Sloodle Button* que, ao ser tocado pelo estudante, se comunica com o ambiente *Moodle* registrando a interação do mesmo com o objeto. Para a detecção da presença do estudante, é utilizado o *Sloodle Scanner*, objeto que envia uma mensagem ao *Moodle* quando o estudante se aproxima, registrando a presença do mesmo em determinado

local, instrumento que foi utilizado nas galerias do museu para registrar quais delas foram visitadas pelos estudantes. O professor tem acesso a tais registros, permitindo-lhe observar os estudantes que realizaram as atividades e visitaram os locais rastreados pelo *Sloodle Tracker*. Também, caso o professor tenha instruído os estudantes a realizarem um determinado roteiro durante a visita ao museu, o relatório do *Sloodle Tracker* permitirá ao professor a análise de como o estudante cumpriu o roteiro proposto.

No museu foi disposto um *Sloodle Scanner* em cada entrada e saída das galerias, além de ter o *Sloodle Button* integrado com vários objetos do museu, não em todos, pois deixaria o servidor muito lento durante a aplicação com uma turma. Para integrar o *Sloodle Button* com os objetos foram necessárias certas modificações em seus scripts (representadas no quadro 3), fazendo com que ele “escute” as mensagens enviadas pelos objetos, quando estes forem tocados. Assim, o objeto registra a interação do estudante quando este clica sobre os objetos do museu, e não quando ele clica sobre o objeto *Sloodle Button* em si.

```

default
{
  state_entry() {
    llListen(-101, "", NULL_KEY, "");
    llSay(0, llGetKey());
  }

  listen(integer channel, string name, key id, string message){
    llMessageLinked(LINK_THIS, CHANNEL,
"INTERACTION|"+MY_TYPE+"|0", (key)message);
  }

  link_message(integer sender_num, integer num, string sval, key kval)
  {
    if (sval == "INTERACTION_RESPONSE" && kval != NULL_KEY)
    {
      llSay(0, llKey2Name(kval) + " touched this button.");
    }
  }
}

```

Quadro 3. *Script utilizado no Sloodle Button.*

Por exemplo, durante uma visita ao museu um estudante entra na galeria da pré-história da computação e interage com o objeto “escrita cuneiforme” e com o “ábaco”. No relatório do *Sloodle Tracker* constarão como “completadas” as atividades com as



quais ele interagiu e sua presença na galeria, registrando a data e a hora da interação/presença, sendo que constará como “não completados” os objetos com os quais o aluno não interagiu e as galerias ou salas pelas quais ele não passou.

Ao final do desenvolvimento do museu e da integração com o Moodle, foi realizada a avaliação do mesmo, que é o assunto do próximo Capítulo.

## 5 AVALIAÇÃO E ANÁLISE SUCINTA DOS RESULTADOS

Com as etapas de modelagem, construção e integração do Museu da Computação finalizadas, o foco passou a ser a avaliação do ambiente. Baseando-se no livro de Barbosa e Silva (2010), “Interação Humano-Computador”, foi iniciado o planejamento da avaliação sob o ponto de vista da usabilidade do museu, no que se refere ao ambiente em si e aos artefatos presentes nele.

Segundo os autores, a avaliação de interfaces envolve as seguintes etapas: *preparação*, onde são definidos os participantes da avaliação e suas tarefas a serem executadas, além da preparação do material para observação e registro durante a atividade; *coleta de dados*, onde é observada e registrada a opinião dos participantes, é a etapa onde os participantes realizam as atividades propostas, enquanto os avaliadores coletam os dados; *interpretação e consolidação dos dados*, que serve para reunir, contabilizar e sumarizar os dados coletados dos participantes; e a etapa de *relato dos resultados*, onde é feito o relato do desempenho e opinião deles.

O método escolhido para avaliação foi o de teste de usabilidade, por melhor se encaixar no objetivo da avaliação do museu, que era tentar mensurar quais os pontos fortes e fracos, e quais poderiam ser melhorados, na interação do usuário com sistema.

Com o método definido, partiu-se para a preparação do teste de usabilidade. Primeiramente, foram definidos os usuários-alvo da avaliação, estudantes dos cursos de Ciência da Computação e de Sistemas de Informação da UFSM, que cursavam a disciplina de Interface Humano-Computador (IHC); a escolha deu-se devido ao fato dos estudantes estarem estudando métodos de avaliação de IHC, podendo fornecer um *feedback* mais criterioso, com respostas mais substanciais durante o teste.

Após, foi definido como seria a visita ao museu; em um primeiro momento, os alunos seriam deixados “livres”, para se adaptarem ao ambiente (teclas de navegação, possibilidades de interação, etc.), após, eles seriam orientados para as galerias do museu e, ao final, seriam direcionados para a sala de estudos para responderem um pequeno questionário de conteúdos associados aos artefatos do museu, questionário este que foi feito no *Moodle* e importado para o museu por meio do *Sloodle* (este questionário pode ser encontrado no apêndice B).

Cabe salientar que o objetivo da avaliação era analisar a usabilidade do ambiente e não o processo de ensino e aprendizagem dos conteúdos apresentados ao longo da visita. No entanto, como uma das etapas da visita compreende a utilização da sala de estudos e seus recursos, optou-se por simular uma visita que incluía o preenchimento de um questionário sobre os conteúdos apresentados. Esta estratégia permitiu avaliar a usabilidade da sala e dos recursos de integração com o ambiente *Moodle*. No entanto, a partir desta atividade, também foi possível inferir algumas possibilidades, que poderão ser analisadas no futuro, com relação a uma avaliação sob o ponto de vista pedagógico.

Optou-se por dividir a visita em duas etapas: na primeira, os estudantes deveriam cumprir o “roteiro” do museu, passando pelas galerias, para que ao final pudessem responder as questões relacionadas aos conteúdos do museu; na segunda etapa seriam deixados livres para visitarem as salas e galerias na ordem que julgassem melhor. Para que os estudantes pudessem avaliar o ambiente, foi elaborado um questionário para os mesmos responderem, se baseando na experiência durante a visita ao museu, quais suas observações, dificuldades e pontos encontrados que poderiam ser melhorados.

Pelo questionário foi buscado fazer o levantamento do grau de sucesso, da quantidade e tipos de erros cometidos e do tempo necessário para a realização de cada tarefa feita durante a visita, como o *login* no sistema, a utilização das ferramentas de interação do visualizador, a interação com os artefatos do museu, a localização das salas e galerias e a resolução do questionário referente aos artefatos do museu. O questionário aplicado para o teste de usabilidade pode ser encontrado ao final do trabalho, no apêndice C. Com relação ao uso dos visualizadores necessários para navegação pelo museu, foi desenvolvido um manual para *download*, instalação e configuração dos mesmos, a fim de auxiliar os estudantes durante estes processos.

Após a preparação, partiu-se para a visita da turma ao museu, da qual nove estudantes participaram. Inicialmente, não houve problemas na parte de configuração do visualizador por parte dos estudantes, devido ao auxílio do manual de configuração, bem como pelo fato do público-alvo ser formado por alunos da área de computação, o que já os caracteriza como usuários familiarizados com o uso de tecnologias.

Após o acesso, os estudantes demonstraram dominar as ferramentas básicas do ambiente, como as teclas de movimentação, a utilização do *chat* do *OpenSim* e a

interação com os objetos. Porém, ficaram “perdidos” no ambiente, sem saber por onde começar a visita, ou o que fazer primeiro. Como o objetivo da primeira etapa era guiar os mesmos pelo museu, foram dadas instruções aos estudantes de como deveria ser a visita, que serviram para guiá-los pelas galerias e salas do museu.

Dentro das galerias percebeu-se que poucos estudantes prestavam atenção nas descrições dos objetos, interagindo somente com os artefatos em si, para visualizar suas animações. Após passarem por todas as galerias, os estudantes foram orientados até a sala de estudos para responderem o questionário baseado nas descrições dos objetos do museu. O resultado geral do questionário foi em torno de 60% de acerto e uma análise inicial dos resultados do *Sloodle Tracker* mostrou que poucos alunos interagiram com as placas dos objetos e que passaram rapidamente pelas galerias.

Durante a segunda visitação, com a participação de sete estudantes da turma, tinha-se como abordagem uma interação livre, onde os estudantes não foram orientados a refazerem as atividades na ordem da primeira visita. Foi possível observar que estudantes optaram por abordagens diferentes quanto ao rumo da visitação. Muitos foram direto aos questionários sobre o museu, para ver as perguntas e “caçar” as respostas pelas galerias do museu, em uma espécie de jogo “caça-tesouro”.

O resultado do questionário melhorou, chegando próximo dos 90% de acerto. Shu e Ming-Shen (2009) afirmam que o viés de jogos “caça-tesouro”, no processo de ensino-aprendizagem, aumenta o interesse do aluno, o ajudando a aprender, afirmação esta que pode ajudar a confirmar os dados obtidos. Ainda, Silva (2012) afirma que jogos voltados para educação profissional ou voltados para o ensino superior, denominados, usualmente, de jogos sérios (*serious games*), podem unir a finalidade educacional explícita e formal com a ludicidade associada a jogos, sendo que inseridos em mundos virtuais 3D podem ter seus objetivos educacionais ampliados.

Considerando as visitações ao museu, outra análise dos relatórios do *Tracker* no *Moodle*, revelou que a interação com as placas dos objetos foi maior, assim como o tempo que cada aluno permaneceu em cada galeria.

Após o término da visita, foi aplicado o questionário do teste de usabilidade, para que os estudantes o respondessem. O questionário foi enviado via *e-mail* para um total de doze estudantes, que participaram em pelo menos uma das visitas; destes, sete responderam as questões e as enviaram com as respostas de volta. Baseando-se em suas

respostas, os estudantes descreveram o sistema de navegação como intuitivo, mas apontaram algumas melhorias que poderiam ser feitas.

No geral os únicos problemas apontados foram os de localização dentro do museu, pois sem um guia, ou uma melhor sinalização, os estudantes se encontraram “perdidos” no primeiro momento dentro do museu, com algumas dificuldades para encontrarem as salas desejadas. Cabe salientar que os questionários respondidos e os dados obtidos com o rastreamento do *Sloodle Tracker* ainda não passaram por um processo de análise detalhada, sendo que as informações descritas aqui são baseadas em percepções dos avaliadores e em uma rápida leitura das respostas. Neste estágio do desenvolvimento do projeto, os questionários estão sendo avaliados para cada uma de suas questões, e análises e possibilidades de melhorias estão sendo estudadas detalhadamente.

## 6 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo apresentar o desenvolvimento do Museu Virtual da Computação, desenvolvido no mundo virtual *OpenSimulator*, que visa apoiar o processo de ensino-aprendizagem de introdução à computação. Toda a estrutura, galerias e artefatos do museu foram criados a partir das ferramentas de modelagem do *OpenSim* e alguns importados do repositório *OpenSim Creations*.

Durante o desenvolvimento do trabalho, as ferramentas e recursos utilizados, como mundos virtuais, ambientes virtuais de aprendizagem e ambientes 3D, se mostraram com grande potencial no desenvolvimento de projetos educacionais. Primeiramente, com relação aos mundos virtuais 3D, são poucas as iniciativas encontradas, tendo a grande maioria o foco no mundo virtual *Second Life*.

A integração com o AVEA *Moodle* foi possível através da utilização do *Sloodle*, recurso que pode auxiliar o professor na utilização do ambiente, pois propicia ferramentas para importação de seus materiais para o ambiente 3D, fazendo de um ambiente prazeroso e interessante ao aluno, um ambiente de auxílio ao processo de ensino-aprendizagem.

A partir da instalação do *Sloodle* foi possível a utilização do *Tracker*, que dá a possibilidade de o professor monitorar os passos do aluno neste ambiente. Apesar da utilização deste recurso não poder afirmar que o estudante cumpriu com as atividades propostas e nem que teve alcançado o nível de aprendizagem desejado, pode traçar indícios com base nos caminhos percorridos pelo estudante, possibilitando inferências sobre as preferências do mesmo, verificando-se a dedicação em determinados pontos do museu e o tempo gasto em cada galeria.

Com relação à avaliação realizada com a turma de estudantes de IHC, pode-se perceber a naturalidade com que os mesmos utilizaram o sistema, sem problemas de adaptação ou configuração do ambiente. Vale salientar que o fato de serem estudantes da área de computação pode ter contribuído para que este processo fosse mais simples. Para obter informações mais precisas sobre a adaptação ao sistema, e mesmo se o processo de instalação e configuração deste é feito com facilidade, é necessária a avaliação junto com estudantes de introdução à computação de outras áreas. Ainda, realizando esta avaliação, com estudantes de introdução à computação, será possível

avaliar o ambiente sob o ponto de vista pedagógico, por serem estudantes cursando uma disciplina com conteúdos relacionados aos apresentados no museu.

Também percebeu-se o maior empenho por parte dos estudantes, quando eles fizeram do questionário sobre o museu um “caça-tesouros”, aumentando sua interação com os artefatos do museu, uma vez que deveriam procurar pelas galerias as respostas.

Para o prosseguimento do projeto, é possível se fazer uma análise detalhada da avaliação realizada, propondo e implementando as melhorias necessárias que surgirem da avaliação. Ainda, conforme mencionado anteriormente, é necessário avaliar o museu com estudantes de introdução à computação, para que se possa fazer uma análise sob o ponto de vista pedagógico deste.

Outra proposta futura é a finalização da integração de agentes inteligentes no museu, que serviriam como guias para os estudantes, os orientando durante as visitas. Outro caminho a seguir seria o de criar jogos dentro do museu, semelhantes com a proposta do “caça-tesouros”, aumentando o interesse dos estudantes no ambiente, fazendo com estes procurem interagir mais com os artefatos, aumentando o nível de aprendizagem durante a visitação. Pois foi possível observar que a união da ludicidade com um ambiente virtual e a integração de jogos ao museu, possibilita maior motivação por parte do estudante, sem descaracterizar o objetivo educacional do mesmo.

## 7 REFERÊNCIAS

- Active Worlds** (2010). Disponível em: <https://www.activeworlds.com/>.
- Almeida, J. M. F. (2006). **Museu Virtual de Informática**. Universidade do Minho, Escola de Engenharia, Departamento de Sistemas de Informação.
- Azevedo, C. E. F.; Elia, M. F. (2011). Proposta de uma Aplicação de Mundos Virtuais na Educação usando o OpenSimulator com diferentes requisitos tecnológicos. Anais do **XXII SBIE - XVII WIE**.
- Bainbridge, W. S. (editor). (2010). **Online Worlds: Convergence of the Real and the Virtual**. London: Springer.
- Barbosa, Simone Diniz Junqueira e Silva, Bruno Santana da (2010). **Interação Humano-Computador**. Editora Campus. 2010.
- Berry, G; Sheard, J; Quartly, M. (2011). A virtual museum of computing history: an educational resource bringing the relationship between people and computers to life. In: **Proceedings of the Thirteenth Australasian Computing Education Conference - Volume 114 (ACE '11)**. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 79-86.
- Brookshear, J. Glenn (2013). **Ciência da Computação: Uma visão abrangente**. 11ª Edição. Editora Bookman. 2013.
- Capron, H. e Johnson, J. (2004). **Introdução À Informática**. 8ª Edição. Editora Pearson Prentice Hall. 2004.
- Carmo, Fábio Martins do (2013). Mundo virtual 3D em plataforma aberta como interface para ambientes de aprendizagem. **Dissertação de Mestrado na Escola Politécnica da Universidade de São Paulo**.
- Chengwei, Y., Chengle, Y., Shijun, L., Xiangxu, M. e Rui, W. (2011). An Approach of Personalized 3D Scene Customization Based on Multimedia Resources. In **International Conference on Multimedia and Signal Processing (CMSP)**, p. 131-135.



- Cox, Robert J. e Crowther, Patricia S. (2009). **A Review of Linden Scripting Language and Its Role in Second Life. Lecture Notes in Computer Science** Volume 5322, pp 35-47. 2009.
- CSTA – Computer Science Teacher Association** (2005). Disponível em: <https://csta.acm.org/>.
- Damer, Bruce (1997). **Avatars!; Exploring and Building Virtual Worlds on the Internet. Peachpit Press Berkeley, CA, USA. 1997**
- Fedelil, R. M., Giulio, E., Polloni, F. PERES, F. (2003). **Introdução à Ciência da Computação**. São Paulo: Cengage Learning.
- Forouzan, B., Mosharraf, F. (2012). **Fundamentos da Ciência da Computação - Tradução da 2ª Edição Internacional**. São Paulo: Cengage Learning.
- Freire, Adriano; Rolim, Cledja e Bessa, Wladia (2010). Criação de um ambiente virtual de ensino-aprendizagem usando a plataforma OpenSimulator. **V CONNEPI**. Maceió. 2010.
- Gal-Ezer, J.; Harel D.; Yehudai, A. (1995). A High-School Program in Computer Science. In: **Journal Computer**. Volume 28 Issue 10.
- Giangrandi, P. and Mirolo, C. (2006). Numeri e Macchine - A virtual museum to learn the history of computing. In proceedings of the **ITicSE 2006**, Bologna, Italy.
- Guzman, Carla e Siqueira, Vera H. F. (2007). O papel educacional do Museu de Ciências: desafios e transformações conceituais. **Revista Electrónica de Enseñanza de las Ciencias** Vol. 6, N° 2, 402-423. 2007
- Heaton, Jeff (2007). Introduction to Linden Scripting Language for Second Life. **Heaton Research, Inc. 2007**.
- Lévy, P. **Cibercultura**. Lisboa : Instituto Piaget, 2000. p. 281.
- Linden Lab: Makers of Shared Creative Spaces** (2013). Disponível em: <http://lindenlab.com/>.
- Livingstone, Sonia (2009). **Maximising opportunities and minimising risks for children online**. From evidence to policy *Intermedia*, 37 (4). 50-53. ISSN 0309-118X.

- Machado, Arlindo (2002). Regimes de Imersão e Modos de Agenciamento. **INTERCOM – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação XXV Congresso Brasileiro de Ciências da Comunicação**. Salvador/BA. 2002
- Marandino, Martha (2003). Enfoques de educação e comunicação nas bioexposições de museus de ciências. **Revista Brasileira de Pesquisa em Educação em Ciências**. v. 3, n. 1. 2003.
- Mattar, J.; Valente, C. (2007). Second Life e Web 2.0 na educação: o potencial revolucionário das novas tecnologias. São Paulo: **Novatec**.
- Medina, E. M. (2004). Beyond the ballot box: Computer science education and social responsibility. **ACM SIGCSE Bulletin** inroads 36(4), 7-10.
- Moodle.org: open-source community-based tools for learning** (2007). Disponível em: <https://moodle.org/>
- Open Wonderland: Open source 3D virtual collaboration toolkit**, (2010). Disponível em: [openwonderland.org/](http://openwonderland.org/).
- OpenSimulator** (2011). Disponível em: [opensimulator.org/](http://opensimulator.org/).
- Osório, F. S.; Musse, S. R.; Santos, C. T.; Heinen, F.; Braun, A. e Silva, A.T. S. (2004). Ambientes Virtuais Interativos e Inteligentes: Fundamentos, Implementação e Aplicações Práticas. **XXIV Congresso da SBC – JAI 2004 (Jornadas de Atualização em Informática)**. Tutorial. Salvador, Bahia.
- Pavlos, K., Demetrios, G.S. (2011). A 3D virtual classroom simulation for supporting school teachers' training based on synectics - "making the strange familiar". In **11th IEEE International Conference on Advanced Learning Technologies**, Athens, 2011.
- Pereira Júnior, J.; Rapkiewicz, C.E.; Delgado, C.; Xexeo, J.A.M. (2005). Ensino de Algoritmos e Programação: Uma Experiência no Nível Médio. **XIII Workshop de Educação em Computação (WEI'2005)**. São Leopoldo, RS, Brasil.
- Ragonis, N. (2007). Computing Pre-University: Secondary Level Computing Curricula. In **Encyclopedia of Computer. Proc. 12th Australasian Computing**

**Education Conference (ACE 2010)**, Brisbane, Australia 21Science, 4th Edition, Ralston.

Rodrigues, Patricia A. e Brandão, Leônidas O. (2009). Tarefa Interativa: uma proposta flexível de interatividade para o Moodle. **Anais do SBIE 2009**.

**Second Life** (2009). Disponível em: <http://secondlife.com/index.php?lang=pt-BR>

Schlemmer, Eliane; Trein, Daiana; Oliveira, Christoffer (2008). Metaverso: a telepresença em Mundos Digitais Virtuais 3D por meio do uso de avatares. In. **XIX Simpósio Brasileiro de Informática Educativa – SBIE 2008**. Fortaleza – CE.

Schlemmer, Eliane e Backes, Luciana (2008). Metaversos: novos espaços para construção de conhecimento. **Revista Diálogo Educacional**. Universidade Puntífica do Paraná. 2008.

Schmitt, Marcelo Augusto Rauh e Tarouco, Liane Margarida Rockenbach (2008). Metaversos e laboratórios virtuais - possibilidades e dificuldades. Anais da **Renote - XII Ciclo de Palestras Novas Tecnologias na Educação**. UFRGS. 2008.

Shu, Hui Hsu e Ming-Shen, Jian (2009). Digital and Modular Design Scheme based on education theory in RPG learning game. **Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on (Volume:03)**. 2009.

Silva, Tarcila Gesteira da (2012). Jogos sérios em mundos virtuais: uma abordagem para o ensino-aprendizagem de teste de software. **Dissertação de Mestrado na UFSM**. 2012.

Tarouco, Liane. Ávila, Barbara, Amaral Erico, Zednik, Herik (2012). VEGA - Implementando um Laboratório Virtual Imersivo no OpenSim. **RENOTE, Revista Novas Tecnologias na Educação**. V. 10 N° 1, julho, 2012.

Wankel, C.; Kingsley, J. (2009) **Higher Education in Virtual Worlds**. Bingley: Emerald.

## 8 APÊNDICES

### APÊNDICE A - Instalação do OpenSim

Para o desenvolvimento deste projeto, o OpenSimulator foi instalado em um servidor do laboratório do GRECA. O servidor tem como sistema operacional o Debian GNU/Linux e está configurado com o pacote LAMP (Linux, Apache, MySQL e PHP). No servidor também foram instaladas algumas ferramentas necessárias, como o Mono<sup>1</sup>, o Nant<sup>2</sup>, o Vim<sup>3</sup>, o Git<sup>4</sup> e o Screen<sup>5</sup>.

Para a instalação e configuração do OpenSimulator foram utilizados os tutoriais disponíveis no seu site (<http://opensimulator.org>). Inicialmente foi feito o *download* do OpenSimulator com o comando:

```
git clone git://opensimulator.org/git/opensim
```

Em seguida, para a compilação do código foi utilizado o *Nant*, através dos comandos:

```
./runprebuild.sh
```

```
nant
```

Após feito o *download* e compilado o código, foi necessária a criação de duas bases de dados no banco MySQL (uma para ser utilizada pelo processo Robust.exe e outra pelo processo OpenSim.exe) e a edição de alguns arquivos do OpenSimulator para que este funcionasse no modo grid. Primeiramente, no diretório opensim/bin, foi necessário renomear os arquivos OpenSim.ini.example e Robust.ini.example, para OpenSim.ini e Robust.ini respectivamente. Então foi utilizado o editor de texto Vim para a edição do arquivo OpenSim.ini, mudando a opção de arquitetura do modo *standalone* para o modo grid, e do arquivo Robust.ini, especificando a base de dados a ser usada, o IP externo e a porta do servidor (<http://servgreca01.ct.ufsm.br:8002>) utilizadas pelo processo Robust.exe.

Já no diretório opensim/bin/config-include foi renomeado o arquivo GridCommon.ini.example para GridCommon.ini. Neste arquivo foi especificada a base de dados a ser utilizada pelo processo OpenSim.exe, além da substituição das ocorrências dos endereços “*myassetserver.com*”, “*myinventoryserver.com*” e

“mygridserver.com” que aparecerem no arquivo inteiro, pelo IP interno do servidor (http://10.0.1.252).

Após estas configurações bastou executar os processos Robust.exe e OpenSim.exe, utilizando o gerenciador de janelas *Screen*. Primeiramente foi executado o processo Robust.exe através dos seguintes comandos, onde o primeiro comando foi utilizado para executar o processo e o segundo para visualizar a janela do processo:

```
screen -S robust -d -m mono Robust.exe
```

```
screen -r -d robust
```

Na janela do processo Robust.exe foi utilizado o comando *create user* para a criação de um nome, sobrenome e senha de usuário, que foi utilizado posteriormente nas configurações iniciais no processo OpenSim.exe, que foi executado em seguida com o comando:

```
screen -S opensim -d -m mono OpenSim.exe
```

Na janela do OpenSim.exe foi configurado, além do nome e sobrenome de usuário, o IP externo e a porta do servidor, que são utilizados para acessar o ambiente virtual através dos visualizadores.

**APÊNDICE B – Questões sobre os artefatos do museu**

Instrumento considerado a primeira calculadora:

- a. Máquina de diferenças
- b. Ábaco
- c. Régua de cálculo
- d. Arithometer

Foi criada pelos sumérios, por volta de 3.500 anos a.c.:

- a. Ábaco
- b. Máquina de escrever
- c. Escrita cuneiforme
- d. Relógio de sol

Dispositivo capaz de detectar, ou não, passagem de corrente elétrica:

- a. Diodo
- b. Máquina de Pascal
- c. Telegrafone
- d. Tear programável

Dispositivo de gravação magnética, usado para gravar sobre um condutor magnético:

- a. Telegrafone
- b. Diodo
- c. Tear programável
- d. Máquina de diferenças

Primeira calculadora comercializada com sucesso, capaz de efetuar as quatro operações básicas:

- a. Ábaco
- b. Arithometer
- c. Máquina de Pascal
- d. Telegrafone

Computador da primeira geração, totalmente eletromecânico:

- a. Apple Lisa I
- b. PDP-8
- c. MARK-I
- d. Eniac

Primeiro supercomputador da segunda geração, permitia um milhão de operações por segundo:

- a. PDP-8

- b. IBM 7030
- c. IBM 360
- d. MARK-I

Um dos minicomputadores mais conhecidos da segunda geração, uma versão básica dos supercomputadores:

- a. IBM 360
- b. Eniac
- c. PDP-8
- d. Apple Lisa I

Computador da terceira geração, permitia processamento em batch multiprogramado e processamento em tempo real:

- a. IBM 7030
- b. IBM 360
- c. Apple Lisa I
- d. DECsystem-10

Computador da quarta geração, primeiro a ter um mouse e uma interface gráfica:

- a. PDP-8
- b. DECsystem-10
- c. Apple Lisa I
- d. Eniac

Autor de uma das primeiras calculadoras mecânicas e de estudos sobre o método científico:

- a. Charles Babbage
- b. Blaise Pascal
- c. John Napier
- d. James Martin

Conhecido por constituir junto com Paul Allen a Microsoft, considerada como a maior e mais famosa empresa de software do mundo.

- a. Willian Henry Gates
- b. Steven Paul Jobs
- c. Blaise Pascal
- d. Alan Mathison Turing

Influenciou o desenvolvimento da Ciência da Computação e proporcionou uma formalização do conceito de algoritmo e computação:

- a. Alan Mathison Turing
- b. John Napier

- c. Seymour Roger Cray
- d. Steven Paul Jobs

Empresário e magnata americano da informática, destacou-se como co-fundador, presidente e diretor executivo da Apple:

- a. John von Neumann
- b. William Henry Gates
- c. Alan Mathison Turing
- d. Steven Paul Jobs

Conhecido como o inventor que projetou o primeiro computador, utilizando apenas a mecânica:

- a. John von Neumann
- b. Blaise Pascal
- c. Alan Mathison Turing

Provavelmente a primeira linguagem de programação da história, surgida na década de 50:

- a. Java
- b. Assembly
- c. Haskell
- d. Fortran

Linguagem projetada primariamente para o processamento de dados simbólicos, é uma linguagem formal matemática:

- a. C++
- b. Assembly
- c. Lisp
- d. Haskell

A linguagem foi criada simultaneamente para ensinar programação estruturada e para ser utilizada em uma fábrica de software:

- a. Pascal
- b. Fortran
- c. C
- d. C++

Linguagem de programação compilada de propósito geral, estruturada, imperativa, procedural, padronizada pela ISO.

- a. C++
- b. Java
- c. C
- d. Lisp



### APÊNDICE C – Questionário do teste de usabilidade

- 1- Sobre o processo de *download*, instalação e configuração do visualizador (*FireStorm*), qual o nível de dificuldade encontrado utilizando o tutorial (fácil, médio, difícil)? E quais as principais dificuldades nestes processos?
- 2- Para realizar o *login* no ambiente, foi necessário gastar quanto tempo? Quais dificuldades foram encontradas?
- 3- Durante a visita ao museu, descreva quais foram as dificuldades para se localizar dentro do ambiente.
- 4- O sistema de movimentação e interação do avatar, com outros usuários e objetos, é de fácil aprendizagem? Quais melhorias poderiam ser feitas no sistema?
- 5- A visualização das descrições dos objetos e das animações foi satisfatória? Quais problemas foram encontrados?
- 6- O sistema de respostas do questionário é intuitivo? Foram encontradas algumas dificuldades?
- 7- Descreva qualquer dificuldade extra, ou sugestão para melhora, encontrada desde o processo de *download* até a realização do questionário no ambiente.
- 8-“Quantos erros os usuários cometem nas primeiras sessões de uso?”;
- 9-“Quantos usuários conseguiram completar com sucesso determinadas tarefas?”;
- 10-“Quantas vezes os usuários consultaram a ajuda online ou o manual do usuário?”
- 11- Para identificar tais rupturas durante o teste de interação este método possui algumas classes de problemas:
  - Orientação (Onde estou? Onde está? Onde estava?)
  - Navegação (Aonde posso/devo ir?)
  - Identificação de objetos (O que é isto?)
  - Interpretação da simbologia (O que isto representa?)
- 12- Os usuários conseguem reconhecer e utilizar todos os controles para a navegação?
- 13 - Os usuários conseguem navegar através dos modelos 3D de forma produtiva?

14- Os usuários conseguem navegar através dos modelos 3D sem perderem sua localização?

15- Os usuários conseguem reconhecer a funcionalidade dos principais controles?