



VIEWERS PARA AMBIENTES VIRTUAIS IMERSIVOS: UMA ANÁLISE COMPARATIVA TEÓRICO-PRÁTICA

Felipe Becker Nunes, PPGI, UFSM, nunesfb@gmail.com

Gleizer Bierhalz Voss, PPGI, UFSM, gleizer.voss@gmail.com

Fabrcio Herpich, PPGI, UFSM, fabricio.herpich@gmail.com

Andreia Rosangela Kessler Muhlbeier, PPGI, UFSM, andreiamuhlbeier@yahoo.com.br

Camila Cerezer Possobom, PPGI, UFSM, camila.ccpobbom@gmail.com Roseclea

Duarte Medina, PPGI, UFSM, roseclea.medina@gmail.com

Resumo. Com a evolução tecnol3gica surgiram novas alternativas que auxiliam no complemento das atividades realizadas no 4mbito educacional, tais como os mundos virtuais. Para que estes ambientes imersivos possam ser visualizados pelos usu4rios, 4 necess4rio o uso de um *software* denominado *viewer*. Este artigo apresenta uma an4lise qualitativa sobre os *viewers* dispon4veis para serem utilizados como suporte aos mundos virtuais, em que s4o demonstradas suas caracter4sticas, vantagens e desvantagens, al4m da an4lise do consumo m4dio de processamento. Esta pesquisa visa desta forma auxiliar os usu4rios a selecionarem a aplica3o mais adequada as suas necessidades, baseando-se nos dados fornecidos por este estudo.

Palavras – Chave: *viewers*, mundos virtuais, ambiente imersivos.

VIEWERS FOR IMMERSIVE VIRTUAL ENVIRONMENTS: A THEORY-PRACTICE COMPARATIVE ANALYSIS

Abstract. *With the emergence of new technological alternatives that help to complement the activities carried out in the educational field, such as virtual worlds. For these immersive environments can be displayed by users, it is necessary to use a software called viewer. This paper presents a qualitative analysis on the viewers available for use as a support for virtual worlds, which are shown in their characteristics, advantages and disadvantages, as well as analysis of the average consumption of processing. This research aims thus help users to select the most appropriate application needs, based on the data provided by this study.*

Keywords: *viewers, virtual worlds, immersive environments.*

1. INTRODU3O

A introdu3o de novos paradigmas de aprendizagem e metodologias de ensino modificou o cen4rio tradicional visto nas salas de aula. O uso das TICs (Tecnologia da Informa3o e Comunica3o) na 4rea educacional apresentou novas alternativas de aprendizagem e estendeu o ensino para al4m do ambiente presencial de sala de aula. Dentre as solu3es existentes, est4o os ambientes virtuais de aprendizagem e os mundos virtuais, que podem ser utilizados como suporte para a realiza3o de atividades.

O paradigma da educa3o imersiva surge como uma alternativa interessante para a elabora3o de recursos multidisciplinares, interativos e din4micos, dentro de um metaverso, ou seja, um tipo de mundo virtual que est4 alocado no dispositivo do usu4rio. Os mundos virtuais s4o definidos por Bainbridge (2010) como ambientes

online persistentes gerados por computador, onde as pessoas podem interagir de maneira comparável ao mundo real, seja para o trabalho ou para o lazer.

Com relação à criação dos mundos virtuais, tecnologias têm sido desenvolvidas para dar suporte à construção destes metaversos, visando facilitar e simplificar o processo de criação. De acordo com Reis et al. (2011), diversas plataformas estão disponíveis para a construção de mundos virtuais que preveem o suporte para a modelação 3D, com capacidade de pré-programação dos comportamentos dos objetos.

Podemos citar como exemplos de metaversos o Second Life, OpenSimulator (OpenSim) e OpenWonderland (da Silva, 2012). O OpenSim foi escolhido para ser utilizado nesta pesquisa, por ser um ambiente gratuito, *open source* e deter de uma vasta gama de documentações. Segundo Tarouco et al. (2012), o principal objetivo do OpenSim é disponibilizar um servidor 3D flexível e modular compatível com os *viewers*/clientes do Second Life.

Para a sua visualização é necessária a utilização de uma aplicação computacional denominada *viewer*, que fornece suporte para a visualização, importação e exportação de objetos, e demais recursos relacionados ao mundo virtual OpenSim. A interface gráfica do metaverso é visualizada por meio do *viewer*, no qual o usuário poderá interagir e realizar suas atividades.

Diante deste contexto, este artigo apresenta uma análise de caráter qualitativo sobre os diversos *viewers* suportados pelo OpenSim, buscando identificar os principais recursos fornecidos por eles, suas vantagens e desvantagens. Com isso, este estudo visa auxiliar o pesquisador na escolha do *viewer* mais adequado para suprir as suas necessidades, ou seja, aquele que melhor se adapta ao seu contexto de trabalho, seja para importação ou visualização de objetos, necessidades de processamento, etc.

Assim, devido à grande quantidade de *viewers* disponíveis e suas particularidades, é importante que o desenvolvedor e/ou usuário esteja ciente das características e limitações de cada ferramenta. A escolha de um *viewer* que não atenda determinadas necessidades pode acarretar em diversos problemas (e.g., restrição na importação de arquivos; visualização parcial ou limitada de objetos; ausência de gerenciamento do *grid*; entre outros), impedindo nesses casos a realização das atividades propostas pelos usuários.

2. FUNDAMENTAÇÃO TEÓRICA

Atualmente, questões de como alavancar a educação vem sendo muito debatidas, neste contexto, a implantação de ambientes virtuais imersivos para a aprendizagem vem ganhando força, pois segundo Tarouco et al. (2012), em um ambiente imersivo, estudantes podem se utilizar de ferramentas como o *chat* para juntos discutirem e engajarem-se na resolução dos problemas propostos. Eles interagem entre si e com o mundo virtual através de seus *avatars*, o que lhes proporciona a sensação de compartilharem um ambiente, incentivando o seu envolvimento em atividades colaborativas.

2.1 Mundos Virtuais e educação

Mundos virtuais, também conhecidos como metaversos ou Ambientes Imersivos, vêm sendo explorados para os mais diversos fins, quebrando a barreira do campo do entretenimento para se consolidar na área educacional (Amaral et al., 2012). Nessa área do conhecimento, a tecnologia desses ambientes contribui de diversas formas para a construção da aprendizagem, dentre as quais:

- Permitem a colaboração entre os usuários através de trabalhos em grupos;
- Permitem a realização de experimentos e testes sem o risco inerente que os mesmos teriam caso fossem realizados em um ambiente real;
- Redução de custos com a instalação e manutenção de laboratórios para a execução desses experimentos;
- Podem exercer diferentes funções como jogo, rede social, simulador, entre outros (da Silva, 2012).

2.2 Viewers

O crescimento da utilização de mundos virtuais com o objetivo de apoiar o processo de ensino-aprendizagem é comprovado pelas recentes publicações nos diversos campos de pesquisa (Tarouco et al. (2012), (da Silva, 2012) e (Callaghan et al., 2009)). Este crescimento é em parte devido à flexibilidade do OpenSim, uma plataforma de mundo virtual *open source*, que, pela adoção de protocolos e normas do Second Life criou de fato um padrão de mundos virtuais abertos que são suportados por um número crescente de *viewers* “terceiros”, em sua maioria de código aberto (Allison et al., 2012).

Os ambientes virtuais citados acima são acessados remotamente a partir de um *software* específico, que é denominado *viewer*. Esses visualizadores estabelecem a conexão com o mundo virtual e permitem a interação do usuário, através de seus *avatares* com os objetos e elementos fornecidos pelo metaverso.

A escolha das ferramentas de visualização utilizadas nos testes teve como base os trabalhos de Tarouco et al. (2012) e Reis et al. (2011), e a lista de *viewers* indicados nas páginas oficiais do OpenSim¹ e do Second Life². Em torno de quinze ferramentas foram instaladas, testadas e analisadas, dentre essas, sete se destacaram e foram escolhidas para compor o quadro comparativo (Tabela 1) que embasa os resultados obtidos nesta pesquisa. Os *viewers* selecionados foram os seguintes:

- Imprudence³: é um *software* que oferece interface amigável e de fácil configuração, em consequência disso, é o *viewer* mais utilizado pelos usuários para acessar o metaverso OpenSim (OpenSim, 2013).
- Firestorm Viewer⁴: oferece uma vasta gama de recursos, opções e escolhas de personalização de interface (Firestorm, 2013).
- Hippo OpenSim Viewer⁵: é um visualizador voltado para usuários OpenSim. Suas principais características são: possibilidade de construção de objetos, facilidade na utilização e na configuração do mundo virtual (Second Life, 2013).
- Phoenix Viewer⁶: é um avançado visualizador de código aberto com diversos recursos, opções e flexibilidade da interface (Firestorm, 2013).
- Singularity Viewer⁷: é um visualizador *open source* utilizado tanto para o Second Life quanto para o OpenSim. Procura combinar um visual agradável com a melhor e mais recente tecnologia disponível (Singularity, 2013).
- Cool VL Viewer⁸: combina elementos de vários dos outros visualizadores, bem como recursos extras, tais como, alta estabilidade e confiabilidade (Second Life, 2013).
- Kokua Viewer⁹: é um visualizador *open source* para mundos virtuais, conecta tanto Second Life quanto OpenSim (Second Life, 2013).

3. METODOLOGIA

O desenvolvimento desta análise surgiu da necessidade da escolha de uma ferramenta para visualização do mundo virtual OpenSim durante a elaboração de um trabalho no

grupo de pesquisa dos autores, que envolve a criação de um mundo virtual para o ensino de redes de computadores. Durante esse trabalho, constatou-se a dificuldade em apontar um único *viewer* que se adequasse às necessidades do projeto, dado a grande quantidade de ferramentas disponíveis e a carência de um embasamento específico sobre o estudo em questão.

Diante desse contexto, este estudo vislumbra identificar as principais características de cada aplicação, suas vantagens e desvantagens, níveis de consumo de processamento, demonstrando um comparativo dos principais *viewers*, apontando os que obtiverem melhores resultados após a análise realizada. Para tanto, um conjunto de etapas previamente projetadas foram elaboradas, como segue.

A primeira fase caracterizou-se pelo levantamento do referencial teórico sobre o tema, na qual foram selecionados os *viewers*. Foram utilizados somente aqueles de licença gratuita e que provêm suporte ao metaverso OpenSim (utilizado no projeto do grupo).

Na segunda etapa, para a realização dos testes nos *viewers* selecionados, primeiramente foram definidas as métricas de avaliação a serem utilizadas. Os critérios usados na seleção destes indicadores foram baseados no estudo bibliográfico realizado a respeito deste assunto, conforme citado anteriormente na seção 2.2.

Em um terceiro momento, visto que esta avaliação é de caráter qualitativo, os testes foram realizados no grupo de pesquisa do qual são integrantes os autores deste trabalho, todos com conhecimentos avançados em informática. Foram utilizadas duas máquinas com configurações distintas, um *notebook* e um *desktop*, cujas especificações podem ser visualizadas na Tabela 1. Um mundo virtual foi criado em um servidor local utilizando o OpenSim, no qual foram feitos os acessos com os *viewers*, a partir destas duas máquinas.

Tabela 1 – Especificações técnicas dos equipamentos utilizados nos testes.

Crítérios / Máquinas	Máquina 01 (<i>notebook</i>)	Máquina 02 (<i>desktop</i>)
Processador	Pentium Processor T4300 2.10 GHz	Core 2 Duo E4400 2.00 GHz
Memória	4 GB	4 GB
Disco rígido	320 GB	750 GB
Placa de vídeo	Memória compartilhada	GeForce 9500GT 512 MB
Sistema Operacional	Windows Seven Ultimate	Windows Seven Ultimate

Por fim, a quarta fase constitui-se do preenchimento dos dados da Tabela 2 com os resultados obtidos nos testes, a seguir esses resultados foram analisados com base na escala Likert. Os questionamentos deste tipo de escala servem para indicar o quanto alguém concorda ou discorda, aprova ou reprova, acredita que seja verdadeiro ou falso uma determinada afirmação (Likert, 1932). Não existe nenhuma abordagem perfeita para seguir esta escala, no entanto é utilizada para se ter uma visão equilibrada da situação. A escala enumera um conjunto de declarações e prevê um número determinado de respostas onde cada uma das alternativas recebe um valor específico diferente que serve para quantificar o resultado final do questionamento (Likert, 1932). Também foram elaborados gráficos para a demonstração dos valores obtidos e realizada uma discussão comparativa sobre estes resultados.

4. AVALIAÇÃO E ANÁLISE DOS RESULTADOS

Para um melhor entendimento das avaliações realizadas a análise foi separada em duas etapas, onde na primeira foram verificados os requisitos definidos para cada *viewer* (Tabela 2). Posteriormente, a medição do consumo médio de processamento utilizado por cada *viewer* foi monitorado com o auxílio da ferramenta Process Lasso¹⁰, uma ferramenta gratuita que faz o sensoriamento dos recursos consumidos por cada *software* na máquina. A seguir são apresentados os dados obtidos e também é feita uma discussão interpretativa a respeito dos resultados.

4.1 Funcionalidades

As funcionalidades apresentadas na Tabela 2 abordam todos os aspectos considerados necessários para estarem presentes em um *viewer*. Foram definidos três tipos de respostas possíveis para a avaliação dos requisitos: sim, parcialmente e não.

Tabela 2 - Comparativo entre os *viewers* analisados.

Critérios/Viewers	Imprudence	Firestorm	Hippo	Phoenix	Singularity	Cool VL	Kokua
Versão utilizada:	1.4.0 beta 2	4.0.1.27000	0.63	1.6.0.1600	1.8.0	1.26.8	3.6.0
Licença:	Open Source	Open Source	Freeware	Open Source	Open Source	Open Source	Open Source
Sistemas Operacionais suportados	Windows, Mac, Linux	Windows, Mac, Linux	Windows, Linux	Windows, Mac, Linux	Windows, Mac, Linux	Windows, Linux	Windows, Linux
Desktop ou Web	Desktop	Desktop	Desktop	Desktop	Desktop	Desktop	Desktop
Selecionar GRID	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Gerenciar GRID	Sim	Parcialmente*	Sim	Sim	Sim	Sim	Sim
Importar objetos de repositórios online (XML)	Sim	Não	Sim	Não	Sim	Sim	Sim
Importar objetos modelados (DAE)	Não	Sim	Não	Sim	Sim	Sim	Sim
Visualizar objetos DAE e XML	Parcialmente**	Sim	Parcialmente**	Sim	Sim	Sim	Sim
Importar objetos multimídia	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Criação de objetos	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Suporta OSSL	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Ajuste do nível de detalhamento dos gráficos	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Ajuste da largura de banda e cache	Sim	Sim	Sim	Sim	Sim	Sim	Sim

* A configuração do Grid é através de um arquivo XML.

** Não visualiza objetos DAE, apenas XML.

Como pode ser visualizado na Tabela 2, a seleção e gerenciamento do *Grid* (denominação dada ao mundo virtual desenvolvido pelo usuário) está presente em todos os *viewers* analisados, com exceção do Firestorm, no qual é necessária a modificação do arquivo *eXtensible Markup Language* (XML) que contém os *Grids* disponíveis para conexão, onde foi adicionado o endereço do mundo virtual criado no OpenSim. Quanto à criação de objetos (portas, janelas, etc.) dentro do mundo virtual e a inserção de *scripts* que utilizam a linguagem padrão do OpenSim (OSSL), todos os aplicativos fornecem essas funcionalidades.

Com relação à importação de objetos prontos (edifícios, computadores, etc.), localizados em repositórios *online* como Zadaroo¹¹ e OpenSim Creations¹², os *viewers* Firestorm e Phoenix não suportam o *upload* destes tipos de arquivos no formato XML, enquanto os demais visualizadores permitem a importação destes objetos. Esse aspecto torna-se importante devido à praticidade fornecida, visto que construções prontas podem ser inseridas diretamente no mundo virtual, otimizando o processo de criação do mundo virtual pelos desenvolvedores.

Com relação aos objetos multimídia (som, vídeo e imagem), é possível realizar a importação em todos os visualizadores. Os arquivos no formato *COLLaborative Design Activity* (COLLADA), extensão de arquivo DAE, criados em modeladores como Skeetchup e AutoCad são suportados na maioria dos *viewers*, com exceção do Imprudence e Hippo, que não reconhecem este tipo de arquivo. Desta forma, limitam consideravelmente a sua utilização, uma vez que o usuário fica impossibilitado de criar e/ou modificar um objeto fora do próprio *viewer*.

Ressalta-se, que apesar do Firestorm e Phoenix não permitirem a inserção de arquivos XML, no momento que o usuário acessar o mundo virtual com um destes *viewers*, ele conseguirá visualizar os objetos adicionados, conforme pode ser visto na Figura 1. Enquanto que, arquivos no formato DAE não são visualizados corretamente em *viewers* como Imprudence e Hippo, conforme visto na Figura 2.

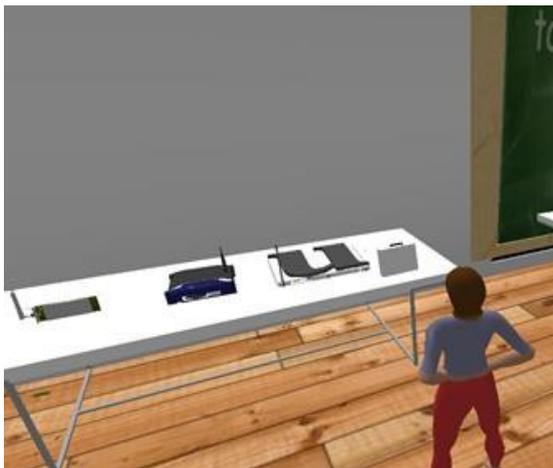


Figura 1 – Objeto XML adicionado pelo Imprudence e visualizado no Firestorm



Figura 2 – Objeto DAE adicionado pelo Firestorm e visualizado incorretamente no Imprudence

Na Figura 1 é possível visualizar uma maleta branca inserida no Imprudence a partir de um arquivo no formato XML, sendo que o objeto é visto em seu perfeito estado. Além disso, é visto uma mesa contendo uma placa de rede e dois roteadores sem fio, objetos adicionados a partir de arquivos no formato DAE. Enquanto que, na Figura 2 é visto apenas a maleta adicionada do arquivo XML, já os demais objetos com extensão DAE ficaram desconfigurados, não permitindo identificá-los. Ou seja, para a correta visualização é necessário utilizar um *viewer* compatível.

Por fim, com relação ao ajuste do nível de detalhamento dos gráficos no mundo virtual e da largura de banda e *cache* para a rede, todos os visualizadores permitem modificar estas configurações de formas semelhantes. A licença dos *viewers* analisados é gratuita e em todos os casos se faz necessária a instalação do aplicativo na máquina do usuário.

A avaliação destes dados foi realizada utilizando a escala Likert, na qual foram definidas pontuações para os três tipos possíveis de respostas: (0) Não, (5) Parcialmente

e (10) Sim. A Figura 3 apresenta o gráfico com as pontuações máximas atingidas por cada *viewer*, resultante da soma de todas as respostas para cada um.

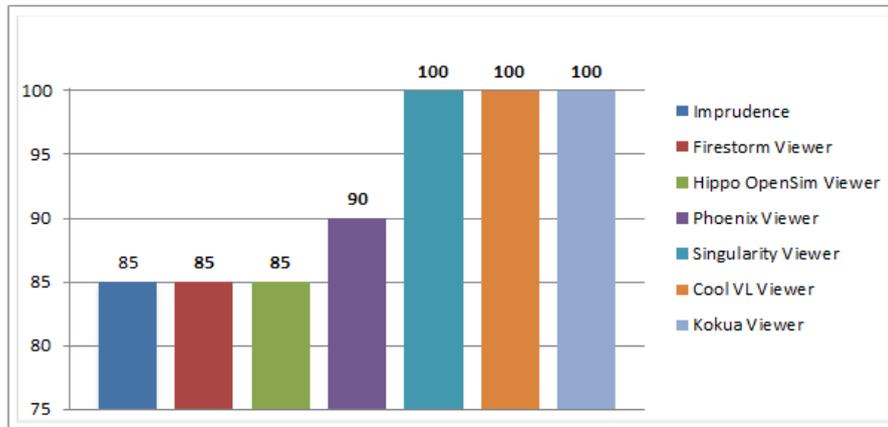


Figura 3 – Pontuação total sobre as funcionalidades em cada *viewer*

Conforme visto na Figura 3, os visualizadores que obtiveram os resultados mais elevados foram o Singularity, o Cool VL e o Kokua (100), seguidos pelo Phoenix (90), enquanto que as pontuações mais baixas foram registradas no Imprudence, Firestorm e Hippo (85). Isto demonstra que todos os requisitos julgados necessários de serem suportados em um *viewer*, foram satisfeitos pelo Singularity, o Cool VL e o Kokua, tornando-os os mais completos em relação a todos os visualizadores analisados.

4.2 Consumo médio de processamento

Um aspecto relevante na hora da escolha da aplicação a ser utilizada é o consumo que ela terá na máquina do usuário, visto que conforme o percentual de processamento consumido, isto pode afetar o desempenho da máquina no momento da inicialização ou durante a interação do usuário com o mundo virtual. Sendo que, caso esse processo seja demasiadamente afetado, pode desestimular os usuários, inibindo a utilização do mundo por parte desses.

Conforme explicado na seção 3, foi utilizada a ferramenta Process Lasso¹¹ para monitorar o consumo médio de processamento por cada *viewer*, durante um período definido de 30 minutos. Cada teste foi realizado separadamente, ou seja, somente um visualizador foi iniciado na máquina por vez para efetuar o registro dos dados. Durante o período de avaliação dos *viewers*, estes foram iniciados na máquina e realizada a conexão ao mundo virtual criado no OpenSim. A partir disto, eles ficaram rodando sem haver interação por parte do usuário, ou seja, não foram feitas atividades no mundo durante o processo de análise, com o intuito de manter um padrão de avaliação.

A Figura 4 apresenta os percentuais de consumo médio de processamento para cada *viewer*, obtidos da máquina 01 (*notebook*). Percebe-se que a maior parte dos visualizadores manteve uma média de consumo similar, variando entre 26% e 32%, com exceção do Cool VL e o Imprudence que atingiram percentuais mais baixos.

Levando em consideração os valores mais elevados e baixos, percebe-se que o Imprudence obteve o menor percentual de processamento, ou seja, exigiu menos trabalho da máquina. Enquanto que o Phoenix atingiu o valor mais elevado de consumo de processamento, o que significou uma maior exigência de processamento por parte do computador 01 para rodar este *viewer*.

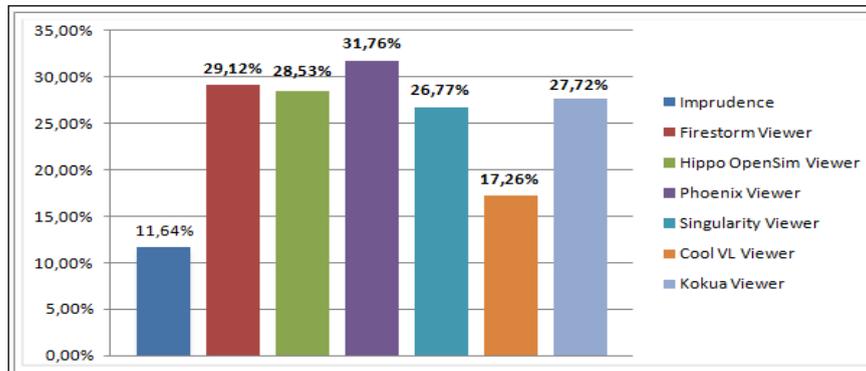


Figura 4 – Consumo médio de processamento pelos *viewers* na máquina 01 (notebook)

A Figura 5 apresenta os percentuais de consumo médio de processamento para cada *viewer*, obtidos da máquina 02 (*desktop*). Percebe-se que houve uma maior disparidade do que em relação ao observado na Figura 4. Os visualizadores Firestorm, Singularity e Kokua obtiveram os resultados mais elevados, respectivamente. Já os demais *viewers* registram valores de consumo mais baixos.

Levando em consideração os valores mais elevados e baixos, percebe-se que o Hippo obteve o menor percentual de processamento, ou seja, exigiu menos trabalho da máquina. Enquanto que o Singularity atingiu o valor mais elevado de consumo de processamento, o que significou uma maior exigência de processamento por parte do computador 02 para rodar este *viewer*.

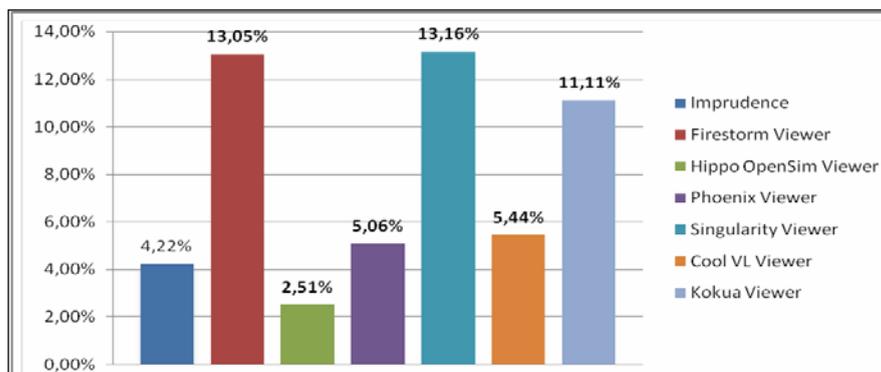


Figura 5 - Consumo médio de processamento pelos *viewers* na máquina 02 (*desktop*)

4.3 Discussão dos resultados

Posteriormente a análise das funcionalidades e do consumo médio de processamento em cada *viewer* ter sido realizada, é possível estabelecer algumas observações gerais a respeito desses. Quanto ao consumo médio de processamento, visto em duas máquinas com configurações distintas, observa-se que devido à máquina 02 deter de maiores recursos computacionais, o processamento exigido pela aplicação não foi tão elevado como o exigido na máquina 01.

Após a análise dos dois gráficos, pode-se perceber que nas duas máquinas houve um consumo médio elevado de processamento por parte do visualizador Firestorm, demonstrando que esta aplicação exige um maior nível de processamento em relação aos demais *viewers* analisados. Já o Imprudence foi o visualizador que menos exigiu processamento das máquinas, sendo, portanto a aplicação mais leve para rodar no computador do usuário.

Levando em consideração os valores obtidos nos três gráficos, verificou-se que os *viewers* que obtiveram uma melhor classificação de forma geral foram o Imprudence, o Cool VL, indicando uma melhor relação custo-benefício que os demais *viewers*. Ainda neste sentido, os *viewers* que obtiveram resultados inferiores foram o Firestorm e Singularity, sugerindo uma pior relação custo-benefício.

Este comparativo buscou esclarecer e demonstrar os aspectos relacionados ao funcionamento dos *viewers*, suas vantagens e desvantagens, com o intuito de apontar as principais características de cada um. Desta forma, o usuário poderá analisar mais aprofundadamente para decidir qual visualizador utilizar, levando em consideração aspectos de processamento, recursos disponibilizados e tipos de arquivos suportados.

5. CONCLUSÃO

Com a utilização das tecnologias no âmbito educacional, novos paradigmas e metodologia de ensino foram introduzidos. O uso dos mundos virtuais pode auxiliar os usuários no complemento do processo de aprendizagem, passando a sensação de imersão em um ambiente, maior interatividade e compartilhamento de informações. Para que isto seja possível, os mundos virtuais precisam ser visualizados através de visualizadores, denominados de *viewers*.

Este artigo apresentou uma análise comparativa teórico-prática, a respeito dos diversos *viewers* disponíveis para serem utilizados como suporte ao mundo virtual OpenSim. Foram explanadas as principais funcionalidades existentes em cada um, suas vantagens e desvantagens, além do consumo médio de processamento exigido pelos visualizadores na máquina.

A escolha de um *viewer* que não atenda determinadas necessidades pode acarretar em diversos problemas: no caso de um *viewer* que exige mais processamento, o usuário deverá optar por um *hardware* mais robusto ou trabalhar com desempenho limitado, podendo causar a insatisfação do mesmo; ou ainda no caso de uma visualização incorreta dos objetos importados, que irá conseqüentemente limitar as ações deste no mundo, impedindo a realização de determinadas atividades. Portanto, o estudo realizado analisou diversos aspectos relacionados aos *viewers*, com o intuito de auxiliar os usuários na seleção do visualizador mais adequado a cada situação.

Como resultado geral foi possível observar que os *viewers* Imprudence e Cool VL atingiram uma melhor classificação dentre todos. Já o Firestorm e o Singularity apresentaram um nível de classificação menor. Ressalta-se que este estudo não visa determinar que somente os *viewers* que obtiveram melhores classificações sejam utilizados, pois a seleção da ferramenta a ser utilizada varia muito de acordo com a necessidade do usuário, como no caso de ter que importar arquivos DAE, não suportados pelo Imprudence.

Neste sentido, o contexto da aplicação torna-se ainda mais importante para a escolha do *viewer* mais adequado. De uma forma geral, o ideal seria utilizar ao menos dois *viewers*, um complementando ao outro. Assim, é possível usufruir dos recursos de cada um desses *viewers* e ao mesmo tempo superar determinadas limitações apresentadas pelos mesmos para o desenvolvimento de algumas atividades.

Como trabalhos futuros, pretende-se realizar uma análise comparativa sobre os mundos virtuais disponíveis atualmente, como OpenSim e OpenWonderland. Serão abordados aspectos como o consumo de processamento, memória, dados trafegados na rede pela aplicação, de forma a demonstrar uma visão geral dos recursos consumidos por cada metaverso.

- ¹ Disponível em: < http://opensimulator.org/wiki/Main_Page>
² Disponível em: < <http://wiki.secondlife.com/wiki>>
³ Disponível em: < <http://wiki.kokuaviewer.org/wiki/Imprudence:Downloads>>
⁴ Disponível em: < <http://www.firestormviewer.org/>>
⁵ Disponível em: < <http://hippo-opensim-viewer.sharewarejunction.com/>>
⁶ Disponível em: < <http://www.firestormviewer.org/>>
⁷ Disponível em: < <http://www.singularityviewer.org/>>
⁸ Disponível em: < <http://sldev.free.fr/>>
⁹ Disponível em: < <http://wiki.kokuaviewer.org/wiki/Kokua/Downloads>>
¹⁰ Disponível em: < <http://bitsum.com/prolasso.php>>
¹¹ Disponível em: < <http://zadaroo.com/New/>>
¹² Disponível em: < <http://opensim-creations.com/>>

REFERÊNCIAS BIBLIOGRÁFICAS

- ALLISON, C.; CAMPBELL, A.; DAVIES, C.J.; DOW, L.; KENNEDY, S.; MCCAFFERY, J.; MILLER, A.; OLIVER, I.; PERERA, I. Growing the Use of Virtual Worlds in Education: an OpenSim Perspective. In: **European Immersive Education Summit - E-iED 2012**, 2., 2012, p.1-13, Paris.
- AMARAL, E; ÁVILA, B. G; TAROUCO, L. M. R. Aspectos teóricos e práticos da implantação de um laboratório virtual no OpenSim. In: **Simpósio Brasileiro de Informática na Educação**, 23., 2012, Rio de Janeiro.
- BAINBRIDGE, W.S. (ed.), *Online Worlds: Convergence of the Real and the Virtual*, Human-Computer Interaction Series. **Springer-Verlag London Limited**, 2010.
- CALLAGHAN, M. J.; MCCUSKER, K.; LOSADA, J. L.; HARKIN, J. G.; WILSON, S. Teaching Engineering Education using Virtual Worlds and Virtual Learning Environments. In: **International Conference on Advances in Computing, Control, and Telecommunication Technologies**. 2009.
- da SILVA, T. G. **Jogos Sérios em Mundos Virtuais: uma abordagem para o ensino-aprendizagem de teste de Software**. Santa Maria: Universidade Federal de Santa Maria, 2012. 84p. Dissertação de Mestrado.
- FIRESTORM**. Página oficial da ferramenta. Disponível em <<http://www.firestormviewer.org/>>. Acesso em: 11 abr. 2013.
- KOKUA**. Página oficial da ferramenta. Disponível em <<http://wiki.kokuaviewer.org/wiki/Kokua/Downloads>>. Acesso em: 27 jun. 2013.
- LIKERT, R. A technique for the measurement of attitudes. **Archives of Psychology**, [S.l.], v.22, n.140, p.1-55, 1932.
- OPENSIM**. Página oficial da ferramenta. Disponível em <<http://opensimulator.org/>>. Acesso em: 10 abr. 2013.
- REIS, R.; FONSECA, B.; ESCUDEIRO, P. Comparative analysis of virtual worlds. In: Iberian Conference, 6., **Information Systems and Technologies (CISTI)**, 2011, p.1-7.
- SECOND LIFE**. Página oficial da ferramenta. Disponível em <<http://wiki.secondlife.com/>>. Acesso: em 18 abr. 2013.
- SINGULARITY**. Página oficial da ferramenta. Disponível em <<http://www.singularityviewer.org/>>. Acesso em: 13 abr. 2013.
- TAROUCO, L.; ÁVILA, B.; AMARAL, É.; ZEDNIK, H. VEGA - Implementando um Laboratório Virtual Imersivo no OpenSim. **Revista Novas Tecnologias na Educação (RENOTE)**, v. 10, n1, 2012.