

Um Esquema de Sincronismo para Ambientes Virtuais de Ensino e Aprendizagem

Rallyson dos Santos Ferreira, Alberto N. de Castro Jr

Programa de Pós-Graduação em Informática – Universidade Federal do Amazonas
(PPGI/UFAM)

Av. Rodrigo Otávio 3000 – 69080-900 – Manaus – AM – Brazil

rallysonferreira@hotmail.com, alberto@icomp.ufam.edu.br

Abstract. *Difficulties on Internet access, from low quality or unstable connections, have become a main obstacle to LMS use on remote places, specially in geographical challenged regions as is the case of the Amazon. In this scenario, we present a scheme for overcome these limitations, that is based on local hosts and a synchronism technique that makes possible the access to main hosts, keeping local resources even when Internet connection is broken, updating when connection is reestablished. The proposed scheme was developed and tested in typical scenarios, using a real ??? based on Moodle platform where “Recent Activity” was adapted and evaluated with respect to user perception.*

Resumo. *Limitações do acesso à Internet, decorrentes de conexões intermitentes e de baixa qualidade, tem se tornado o principal obstáculo para o uso de AVEAs em localidades fora das capitais, especialmente em regiões geograficamente desafiadoras como é o caso da Amazônia. Nesse cenário, apresentamos um esquema para contornar tais limitações, baseado no uso de servidores locais e de uma técnica de sincronismo que possibilita a intermediação do acesso a servidores centrais, mantendo recursos locais ainda que a conexão a Internet seja interrompida, realizando as atualizações quando tal conexão for restabelecida. O esquema proposto foi desenvolvido e testado em cenários típicos, utilizando uma instalação real baseada na plataforma Moodle onde o módulo Atividade Recente foi adaptado e avaliado com respeito à percepção do usuário.*

1. Introdução

Atualmente idealizamos a *Web* como uma plataforma onipresente de desenvolvimento e disponibilidade de serviços. Entretanto, em algumas regiões isso ainda não é verdade, o que causa um enorme contraste entre aquelas localidades e outras que possuem estrutura adequada para uso da Internet. No topo disso, há domínios onde o acesso aos recursos disponibilizados pela *web* tem se tornado essencial, caso da Educação a Distância (EAD) que comumente utiliza Ambientes Virtuais de Ensino e Aprendizagem (AVEAs) desenvolvidos segundo a arquitetura cliente-servidor para acesso através da *web*.

O esquema apresentado neste artigo possibilita aos AVEAs acesso a recursos locais quando em situações instáveis de acesso a Internet, e uma vez que a conexão esteja restabelecida, possibilita também a sincronização periódica dos recursos globais.

As próximas seções descrevem os elementos constituintes desse esquema, bem como sua implementação utilizando uma instalação real baseada na plataforma Moodle, onde cenários típicos foram testados e o módulo “Atividade Recente” foi adaptado e avaliado segundo a percepção do usuário.

2. Arquitetura do Sincronismo Proposto

A arquitetura cliente-servidor é a mais amplamente adotada por AVEAs e cuja estrutura envolve um servidor principal acessado através da Web por navegadores (clientes). Tal estrutura oferece inúmeras vantagens embora seja dependente das condições de acesso de cada usuário. Essas condições de acesso são bastante instáveis em grande parte da região amazônica, onde a ausência de conexão impossibilita qualquer atividade mediada através do AVEA. Na Figura 1 apresentamos uma configuração alternativa, aderente à realidade de laboratórios de informática encontrados em escolas da rede pública e pólos presenciais da região. Esse esquema tem como elementos centrais:

1. Uso de um Servidor Local que:

- repassa diretamente (*by pass*) as operações entre os clientes e o Servidor Principal em condições normais de acesso a *web*.
- faz um espelhamento das bases de dados atualizando-as constantemente e indica ao usuário, através de auxílio na *interface*, às alterações recentes realizadas no ambiente.

2. Um mecanismo de sincronismo entre o Servidor Local e o Servidor Principal quando as condições de acesso estiverem normais.

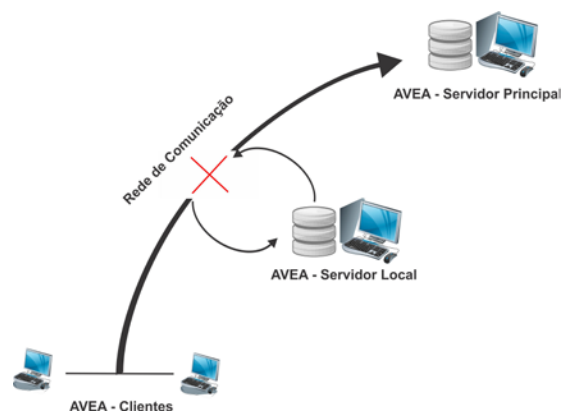


Figura 1. Contornando problemas da arquitetura cliente-servidor em AVEAs.

A Figura 2 ilustra o fluxo de execução *on-line* onde os AVEAs estão instalados em dois domínios distintos (DA e DB). As estações ligadas a esses domínios alimentam o banco de dados localmente, que por sua vez é atualizado quando alguma alteração ocorre no AVEA, mantendo o sincronismo dos dados enquanto houver conexão.

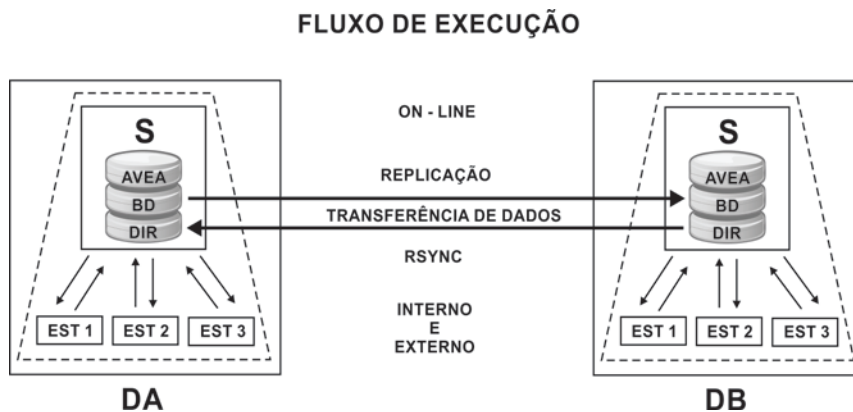


Figura 2. Fluxo de execução do sincronismo *on-line*.

Durante o acesso *on-line*, em que todas as atualizações acontecem em modo síncrono, verificou-se a necessidade de eleger um domínio principal (DA), que será aqui representado pelo domínio mantido no servidor na capital. Isso será necessário para que os acessos sejam direcionados a um só ambiente atualizando-o quando houver conexão com a Internet. Somente depois as atualizações serão replicadas para outros ambientes. Portanto, quando não houver conexão, os acessos serão redirecionados aos ambientes mantidos no servidor local, e as alterações feitas serão enviadas ao ambiente principal quando restabelecer conexão.

No modo de execução *off-line*, onde a conexão é nula (a taxa de transmissão de dados é zero), ou parcialmente limitada (existe uma taxa de transmissão mínima de dados), as estações trabalham localmente ou por agendamento (*scheduling*), em que os domínios tentam manter a comunicação em um determinado intervalo de tempo programado e diferente para cada um, para atualizar o serviço quando houver alterações.

3. Implementação do Modelo de Sincronismo Proposto

A arquitetura proposta concentra-se principalmente na localização onde os dados estão armazenados para que possamos realizar o sincronismo. Mediante isso, construímos uma combinação de dois métodos, observando que o ambiente virtual utilizado, *Moodle*, não armazena suas informações em um único lugar [Bigas, 2009], sendo que os arquivos de *downloads* e *uploads* são armazenados em diretórios de dados e a maior parte das informações, no banco de dados do AVEA.

Dessa forma, o primeiro passo é estabelecer um sincronismo com o BD dos AVEAs onde está armazenado a maior parte das informações. Portanto, seguindo a arquitetura proposta, trabalhamos dois domínios usando uma técnica de replicação de bases de dados para manter os dados sempre disponíveis e atualizados nos outros ambientes. O segundo passo é a aplicação do algoritmo *Rsync* para replicar os diretórios de dados, complementando o sincronismo em paralelo com os outros dados que não estão armazenados no BD do AVEA.

3.1. Aplicação da Replicação da Base de Dados

Os fatores que nos levaram a aplicar a replicação do banco de dados foram principalmente com relação às vantagens que esta técnica proporciona, como: (i) Disponibilidade, (ii) Confiabilidade [Hector *et al.*, 2001], e (iii) Desempenho [Özsu and

Valduriez, 2001]. Na implementação da replicação utilizamos duas máquinas, uma para servir como mestre e outra como escravo. Porém, na configuração é necessário que ambas as partes possam ter liberdade de replicar nas duas direções. Por isso, damos ordem de replicação do mestre para o escravo e do escravo para o mestre.

Segundo [MySQL AB, 2008], o servidor mestre possui um *log* binário de atualizações onde registra as alterações no BD. É mantido também um arquivo de índices dos *logs* binários para manter os registros da rotatividade dos *logs*. Cada escravo, conectado no momento, informa ao mestre onde parou desde a última atualização concretizada, realiza a atualização e então para e espera o mestre informar sobre novas atualizações.

3.2. Aplicação do Algoritmo *Rsync*

O Algoritmo *Rsync* [Tridgell and Mackerras, 1996] é utilizado para copiar diretórios e seus conteúdos da forma mais exata possível (preservando permissões, modos, *timestamps*). Nossa intenção foi manter cópias fiéis de dados entre os ambientes, transmitindo esses dados através da rede utilizando o mínimo de largura de banda. Para isso, utilizamos um sincronismo de forma incremental, ou seja, só é transmitido para o destino o que foi modificado na origem. Nos testes com o *Rsync* utilizamos a opção de comando do SSH [Ylonen and Lonvick, 2006]. A criptografia usada pelo SSH (*Secure Shell*) se destina a garantir a confidencialidade e integridade dos dados através de uma rede não segura, como a *web*.

O *Rsync* é direcionado para ambos os servidores (sincronização bidirecional), assim, todos atuam como mestres e escravos no que se refere à atualização dos dados. Para automação deste trabalho criamos *scripts* para execução do sincronismo dos diretórios. Em cada servidor, descrevemos os dados de endereçamento dos outros servidores que estamos enviando as informações.

3.3. Políticas de Acesso às Atualizações

Quando estamos tratando de um conjunto de solicitações que precisam ter acesso a um determinado local para realizar diferentes ações, precisamos especificar quem tem prioridade naquele momento para obter autorização de acesso. Uma forma de controlar isso é criando regras que especificam todos os aspectos de solicitações (inclusive quem está autorizado a acessar), quando eles podem estabelecer o acesso e o nível de segurança que devem utilizar para efetuar as alterações cabíveis.

Por isso, objetivando manter o esquema de sincronismo entre os AVEAs consistente, e de forma a assegurar a integridade dos dados alterados em domínios distintos, estabelecemos políticas de acesso às atualizações, prevenindo possíveis erros e perdas dos dados, visto que serão utilizados dois métodos de sincronismo diferentes.



Figura 3. Políticas de acesso às atualizações com fluxo contínuo de dados.

Conforme ilustrado na Figura 3, quando houver conexão, todos os acessos são direcionados para o servidor principal localizado na capital. Assim, todas as funcionalidades são acessadas normalmente e as atualizações repassadas automaticamente ao servidor local. No momento em que não existir conexão com a Internet os acessos são redirecionados aos servidores locais. Nesse caso, sendo atualizados respeitando prioridades que evitam conflitar as alterações realizadas nos domínios. Dessa forma, mesmo os dois métodos de sincronismo atuando em paralelo, precisam ser tratados de forma individual devido às atualizações estarem replicando os dados em momentos diferentes.

3.3.1 Sincronia Durante a Atualização dos Diretórios de Dados

A sincronia dos diretórios de dados com o *Rsync* é realizada no modo assíncrono, pois, o *script* criado para automatizar o sincronismo com o algoritmo é programado para ser executado somente no tempo definido para sua atualização, independentemente de se ter conexão ou não com a Internet no momento. Se não houver conexão de dados no momento programado, ele ficará aguardando a próxima execução até conseguir estabelecer o sinal novamente. O tempo para execução do sincronismo foi definido em uma hora. Em cada um dos domínios foi programado um horário diferente dos demais para executar o *script*, deixando uma margem de trinta minutos para que pudéssemos atualizar os dados sem que houvesse conflitos na sincronização, formando um ciclo de atualização.

O algoritmo *Rsync* foi programado apenas para adicionar as informações no repositório de dados, e não faz a exclusão das informações quando elas não constarem no destino. Dessa forma, não perdemos dados quando os dois domínios tiverem informações diferentes adicionadas antes do momento das atualizações acontecerem.

No banco de dados MySQL a replicação é feita principalmente em modo síncrono, ou seja, é realizada no exato instante em que a modificação acontece. Portanto, quando é realizada alguma alteração no banco de dados mestre, havendo conexão, no mesmo instante são replicadas as alterações no banco de dados escravo, e dessa forma, acontecerá também do escravo para o mestre.

3.3.2 Sincronia Durante a Atualização do Banco de Dados

Durante uma replicação assíncrona no BDs, quando os acessos estiverem acontecendo localmente, as cópias dos dados podem ficar fora de sincronia entre os

bancos durante algum tempo. Quando alguma informação é incluída, a modificação é registrada nos *logs* e aguarda a leitura dos BDs em outros servidores. Nesse momento, nos dois servidores podem ser incluídas novas informações e na hora de atualizarem podem acontecer conflitos na replicação que comprometam a estrutura dos BDs.

Dessa forma, é importante que a replicação possa tratar esses conflitos de forma a assegurar a integridade dos dados. Com isso, há a necessidade de sincronizar essas atualizações, de forma a manter a coerência e consistência do sistema como um todo. Portanto, as atualizações no BD do ambiente foram definidas por ordem de chegada registrada de acordo com o momento em que aconteceram as ocorrências, ou seja, os ambientes manterão o registro das alterações de ambos os domínios e atualizarão as informações de acordo com o horário em que ocorreram as mudanças.

4. Adaptação do Módulo “Atividade Recente”

A importância de introduzir uma ferramenta que possibilite aos professores e alunos que utilizam AVEAs, a percepção dos dados que são introduzidos no ambiente virtual é, principalmente, quando algum dos domínios estiver sem comunicação. Dessa forma, tendo informações gerais sobre as últimas postagens, todos os participantes poderão identificar também, quando foram realizadas as últimas atualizações do ambiente, tornando perceptível quando foi possível ocorrer o último sincronismo dos dados.

Portanto, adaptamos o módulo “Atividade Recente”, que tem como objetivo registrar as últimas alterações realizadas pelos participantes no ambiente virtual, a partir dos resultados de sua avaliação através dos métodos da Engenharia Semiótica.

O módulo “Atividade Recente”, que está disponível a partir da versão do *Moodle* 1.9, foi adaptado para descrever as alterações somente quando os usuários estiverem acessando informações dentro da área referente ao curso participante, não sendo de interesse as informações dos cursos em que não é matriculado.

4.1. Avaliação do Módulo “Atividade Recente” Usando o MAC

Para avaliar os princípios de comunicação e colaboração que a introdução do módulo “Atividade Recente” efetiva aos participantes dos cursos, analisando a metacomunicação da *interface* com os usuários, aplicamos as estratégias da Engenharia Semiótica [De Sousa, 2005], [De Souza *and* Leitão, 2009], para medir o nível de percepção desta ferramenta quanto ao sincronismo proposto neste trabalho.

A engenharia semiótica propõe dois métodos para avaliar os efeitos da comunicabilidade em sistemas computacionais: o Método de Inspeção Semiótica (MIS) [De Souza *et al.*, 2008] e o Método de Avaliação da Comunicabilidade (MAC) [Prates *et al.*, 2000]. Dessa forma o método de avaliação realizado foi o MAC, pelo fato de realizar uma abordagem centrada na utilização de um artefato pelo usuário no que diz respeito aos aspectos de *interface*, e ressaltando que não avaliaremos o ambiente num todo, mas, somente a ferramenta em questão.

Na definição do cenário de uso definimos dois tipos de usuários diferentes: o ‘Usuário Comum’ que classificamos como aqueles alunos que não utilizam frequentemente AVEAs; e o ‘Usuário Experiente’ que classificamos como conhecedores praticantes de AVEAs. Dessa forma, convidamos cinco usuários para cada tipo de

classificação, o que representa um número adequado para que fosse realizado esse teste [Nielsen, 1993].

A Tabela 1 mostra o resultado da soma dos registros das etiquetas do MAC, em que a maior ocorrência foi da etiqueta identificada como Cadê?, devido aos usuários tentarem em um primeiro instante identificar as informações na área central do ambiente e não através da ferramenta.

Tabela 1. Registro das Etiquetas do MAC.

Etiquetas	Usuário Comum	Usuário Experiente	Total
Cadê?	3	2	5
E agora?	1	-	1
Ué, o que houve?	-	-	-
Epa!	2	-	2
Assim não dá	-	-	-
Onde estou?	2	-	2
O que é isto?	1	1	2
Por que não funciona?	-	-	-
Socorro!	-	-	-
Vai de outro jeito	1	-	1
Não, obrigado	1	1	2
Para mim está bom...	1	-	1
Desisto	-	-	-

Fizemos o cruzamento das informações obtidas através dos métodos avaliados, criando um perfil semiótico e reconstruindo a metamsagem feita para o usuário.

“Segundo minha interpretação você é um usuário experiente da área de ciência da computação e encontra problemas em utilizar AVEAs em lugares com a conexão de banda da Internet limitada ou nula, por isso adotou um ambiente que utiliza uma técnica de sincronismo automático podendo trabalhar também em modo off-line, postando suas informações mesmo se não tiver conexão nenhuma com a Internet. Mas, possui dificuldades para identificar as últimas atualizações do ambiente.

Eu entendo que você precisa de uma ferramenta que identifique todas as alterações no AVEA lhe informando quem postou, que momento ocorreu a alteração e o que foi alterado, reforçando o intuito de colaboração e comunicação da ferramenta com relação ao sincronismo.

Desse modo, adaptei um módulo no ambiente virtual em que você poderá visualizar todas as ocorrências e, além disso, saberá identificar em que instante aconteceram às últimas atualizações no AVEA. O Módulo estará localizado do lado direito superior do ambiente, e as informações serão apresentadas com uma fonte adequada para que esteja visível a todos.”

5. Estudo de Caso

Na elaboração do estudo de caso desse projeto foi organizado um modelo de cenário de uso em dois laboratórios do Instituto de Computação da UFAM, com o propósito de

gerar situações condizentes com a realidade enfrentada pelos alunos que utilizam os AVEAs na modalidade EAD.

Em um primeiro contato, alguns alunos descreveram ter conhecimento sobre outros ambientes, usadas em disciplinas dos cursos de Ciência da Computação, como é o caso do ColabWeb. Outros relataram ainda não ter muito contato com AVEAs.

5.1. Descrição do Cenário

Para a implementação da técnica de sincronismo, foram configuradas duas máquinas utilizadas como servidor mestre e servidor escravo. O servidor mestre foi utilizado como o servidor principal, e o servidor escravo como servidor local e serviu para controle da coleta dos dados de acesso dos estudantes.

A rede utilizada foi do tipo LAN (*Local Area Network*), padrão IEEE 802.3 própria da UFAM. Não existem *nobreaks* em ambos os locais, o que ocasiona falhas no *link* de Internet. Esses fatos nos proporcionaram um cenário adequado para realizar os testes, podendo assim verificar a consistência da replicação implementada.

O BD utilizado foi o MySQL na versão 5.1.67 e a plataforma foi o Ubuntu 11.10. O ambiente *Moodle* utilizado foi a versão 2.2.9, que na época que se iniciou os testes era a versão estável.

5.2. Aplicação dos Conceitos ao Estudo de Caso

Para os testes do estudo de caso, relacionamos algumas situações que foram criadas simulando diferentes contextos do sincronismo nos quais, professores e alunos deveriam enfrentar durante o acesso ao ambiente virtual.

Um exemplo das situações avaliadas foi elaborada da seguinte forma:

Situação 1 - Tudo ON (O sistema funciona perfeitamente e posteriormente um dos Servidores fica OFF)

- (1) Professor envia o trabalho e a data de entrega.
- (2) Os alunos visualizam o envio do trabalho.
- (3) Professores e alunos postam no fórum.

Nota: Neste momento o Servidor Local (PC) fica sem conexão.

- (4) Alunos respondem o trabalho em tempo hábil de entrega.

Nota: Após a data de entrega o Servidor Local (PC) restabelece a conexão. Neste momento o sincronismo dos dados é efetivado.

- (5) O Professor verifica quem respondeu o trabalho na data correta.

A Figura 4 ilustra o cenário elaborado para os testes, em que no servidor principal o Coordenador do curso representado pelo Usuário Administrador é quem controla e observa as alterações realizadas, validando em ambos os domínios o sincronismo entre os ambientes. No servidor local os alunos acessam o ambiente postando e respondendo as tarefas enviadas pelo Coordenador.

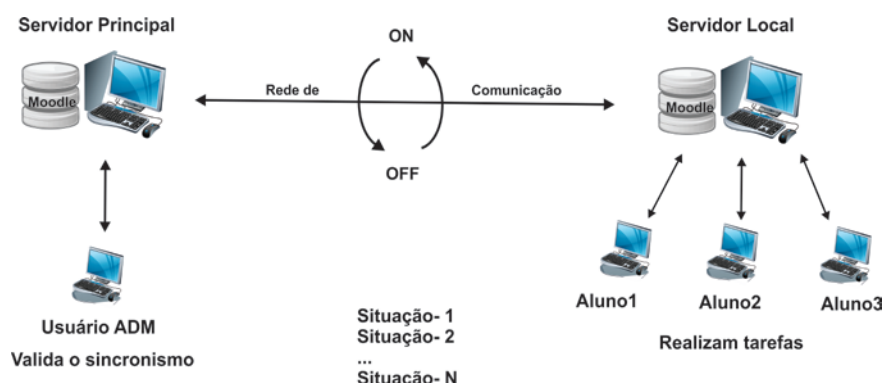


Figura 4. Representação do estudo de caso.

A cada situação avaliada, foram discutidos com os alunos os passos executados com o único objetivo de detectar possíveis falhas durante o uso dos ambientes, como também das técnicas implementadas.

No momento em que os alunos eram designados a acessar o ambiente criado para o sincronismo automático, eles também eram orientados a acessar outro AVEA baseado no Moodle (o *ColabWeb*) utilizado nas disciplinas do IComp/UFAM, mesmo sem realizar nenhuma tarefa. Esse acesso ao mesmo tempo, aos dois ambientes de aprendizagem, foi importante para que pudéssemos verificar e comparar a disponibilidade dos AVEAs mediante os diferentes contextos em que a conexão com a Internet local poderia se encontrar. Dividimos os alunos em três grupos, um grupo para cada turno em que realizamos os testes: manhã, tarde e noite. Cada grupo realizou todas as dez tarefas impostas mediante as situações criadas para os testes.

Tabela 2. Registro de disponibilidade de acesso dos AVEAs.

Registro de Disponibilidade de Acesso dos Ambientes Virtuais de Aprendizagem				
Ambientes	Horários de Acesso			Total de Acessos
	Manhã	Tarde	Noite	
Ambiente de Sincronismo	10	10	10	30
ColabWeb	8	10	7	25

Observando a Tabela 2 verificamos que todos os momentos em que os alunos tentaram acessar o ambiente criado para o sincronismo, eles conseguiram realizar as alterações desejadas no AVEA. Mesmo ocorrendo falhas na replicação os alunos conseguiram acessar o ambiente, visto que esse problema não afeta o acesso diretamente ao AVEA.

Por outro lado, constatamos que o acesso ao ambiente ColabWeb não alcançou êxito em algumas tentativas, conseguindo uma margem de 83,4% de um total de 30 acessos. Em 4 das 5 ocasiões em que o ColabWeb não estava disponível para acesso, o motivo foi uma interrupção no sinal com a Internet e em 1 ocasião o motivo foi falta no abastecimento de energia no prédio onde o servidor hospeda o ambiente virtual.

O estudo de caso representou uma fonte valiosa de avaliação de nosso esquema de sincronismo implementado. Isso se deve, também, pelo fato de que o estudo de caso

é criado para fazer o detalhamento dos testes a partir de múltiplos pontos de vista dos participantes envolvidos em uma situação.

6. Conclusão

Este artigo descreveu a concepção e implementação de um esquema alternativo para uso de ambientes virtuais em situações instáveis de conexão com a *web*. As tarefas definidas para os testes constantes na Seção 5 foram executadas com sucesso, sendo que mesmo com interrupções na replicação o acesso aos ambientes não foi afetado, uma vez que estavam instaladas no servidor local. Através do estudo de caso, comprovamos 100% de disponibilidade do AVEA, mediante o esquema de sincronismo proposto neste, evidenciando a contribuição central da investigação.

Através deste projeto, estabelecemos um fluxo de trabalho diferenciado para as ações de EAD mediadas por ambientes virtuais de ensino e aprendizagem, abrindo novas oportunidades para utilização de AVEAs contornando as limitações impostas pela arquitetura cliente-servidor em tais ambientes.

Referências

- Azevedo, C. E. F. and Elia, M. F. (2011). *Proposta de uma arquitetura de acesso a mundos virtuais para escolas com baixo requisito tecnológico*. 8º E-TIC – 2011 – Relação das Comunicações. Rio de Janeiro.
- Bigas, J. M. C. (2009) “*Canal PBX Voip para E-learning*”. Dissertação de Mestrado em Informática – Universidade do Minho. Braga, Portugal.
- De Souza, C. S. (2005) *The Semiotic Engineering of Human-Computer Interaction*. The MIT Press, Cambridge, MA.
- De Souza, C. S., Leitão, C. F., Prates, R. O., Bim, S. A. and Da Silva, E. J. (2008) *Using the Semiotic Inspection Method in Scientific Research Contexts*.
- De Souza, C. S. and Leitão, C. F. (2009) *Semiotic Engineering Methods for Scientific Research in HCI*. Princeton: NJ. Morgan & Claypool.
- Hector, G. M., Jeffrey, D., and Jennifer, W. (2001). *Implementação de Sistemas de Bancos de Dados*. Rio de Janeiro: Campus.
- MySQL AB, Sun Microsystems. (2008) *Manual de Referência do MySQL 4.1*. Disponível em: <<http://ftp.nchu.edu.tw/MySQL/doc/refman/4.1/pt/index.html>>. Acesso em 26/11/2012.
- Özsu, M. T. and Valduriez, P. (2001) *Princípios de Sistemas de Bancos de Dados Distribuídos*. Editora Campus. ISBN 85-352-0713-9.
- Prates, R. O., De Souza, C. S. and Barbosa, S. D. J. (2000) *Communicability Evaluating Method for User Interfaces*. ACM Interactions, Nova York: v. 7, n. 1, 33–38.
- Tridgell, A. and Mackerras, P. (1996). *The Rsync Algorithm*. Imagine (5).
- Ylonen, T. and Lonvick, C. (2006) *The Secure Shell (SSH) Protocol Architecture*. Standards Track. RFC 4251.