

# Integração do Moodle com Sistema de Gestão Acadêmica

**José Martins de Oliveira Junior**  
Universidade Estadual do Maranhão  
São Luís, MA Brasil  
josemartins@nti.uema.br

**Luan Pereira do Nascimento Correa**  
Universidade Estadual do Maranhão  
São Luís, MA, Brasil  
luanpereiranc@gmail.com

**Luís Carlos Costa Fonseca**  
Universidade Estadual do Maranhão  
São Luís, MA, Brasil  
luiscarlos@uema.br

## RESUMO

Este trabalho descreve um software construído para integrar um sistema de gestão acadêmica com um AVA – Ambiente Virtual de Aprendizagem. Demonstra ainda que a conquista deste objetivo é fundamentada em um modelo arquitetônico orientado a serviços, tendo como principal elemento os webservice, e protocolos de integração por serviços. O resultado obtido demonstra que a integração pode se dar de forma bidirecional. Ao final são sugeridos novos trabalhos para permitir não só uma integração ponto a ponto entre Sistemas de Gestão Acadêmica, mas também multiponto permitindo a uma instituição se conectar a múltiplos AVAs (Ambiente Virtual de Aprendizagem).

## ABSTRACT

This paper describes software to integrate an academic management system with the VLE - Virtual Learning Environment. Demonstrated that an conquer this component is fundamentally the web service, having main element the webservice, and protocols of integrated services. The result is that the integration can be bidirectional. In the end, new work is suggested to help not only point-to-point integration between Academic Management Systems, but also multipoint, allowing an institution to connect to multiple VLEs (Virtual Learning Environment).

## INTRODUÇÃO

A integração pode ser definida como a sincronização entre sistemas. Pode-se integrar dados, processos e serviços. A sincronização consiste em garantir que um evento no mundo real seja igualmente registrado em dois sistemas, de modo a permitir que a informação resida nos dois sistemas integrados.

Em particular, quando se trata da integração de um Ambiente Virtual de Aprendizagem, cada registro na aplicação de gestão acadêmica deve ter uma cópia no AVA e vice-versa.

Zacarias e Livongue [20] postulam que a maioria das demandas de integração visa integrar os seguintes dados: (I) Utilizador; (II) Curso; (III) Disciplina; (IV) Matrícula; (V) Nota. No entanto, a integração de aplicações e sistemas não é uma tarefa fácil, nota-se dificuldades em relação ao tratamento de transações e de acompanhamento de processos quando esses processos cruzam a fronteira da empresa em um contexto de colaboração. Dentre outras razões destaca-se o fato bem conhecido no mercado que a maioria dos fornecedores apenas possibilita integração dos seus próprios sistemas e aplicações relacionadas, tornando assim difícil o processo de integração com sistemas diferentes.

Como em quaisquer outros seguimentos de atuação humana, as instituições de ensino também padecem da falta de integração entre sistemas, condição acentuada com a chegada e a proliferação do uso dos Ambientes Virtuais de Aprendizagem (AVA).

O conceito de educação à distância é amplo, compreendendo processos mediados por outros recursos como teleconferência, videoconferência correio, rádio, televisão, vídeo, CD-ROM, telefone, fax e tecnologias similares, Moran [12]. A mediação tecnológica, com aporte da internet, caracteriza a EaD modalidade online. O ambiente virtual de aprendizagem - AVA é o *lôcus* onde as interações acontecem.

Piaget posiciona o centro de formação da aprendizagem no indivíduo, espalhando-se então para o meio social. A experiência de ‘aprender’ visita esquemas conceituais já existentes, mas os renova por meio do despertar ensejado pelas trocas sociais, Moraes [11]. Os ambientes virtuais de aprendizagem, em especial se considerarmos a emergência das redes sociais como pano de fundo, proporcionam um caso particular deste meio social.

Tais condições em que não há interação física entre instrutor e educando, nas quais uma maior autonomia é requerida do discente, e em que, ademais, os investimentos das instituições de ensino são caracterizados por incentivos financeiros em cenários de aprendizagem e

infraestruturas destinadas a mediar à propagação do conhecimento e possibilitar o aprendizado, apenas sublinham a necessidade de maior controle pelos gestores de todos estes processos educacionais tornando necessária a integração dos meios de interação com os sistemas de controle.

## OBJETIVOS

Descrever uma aplicação que integra ambientes de virtuais de aprendizagem, em nosso caso o Moodle com um Sistema de Gestão Acadêmica. Em particular com o SigUema – Sistema de Gestão Integrada da Universidade Estadual do Maranhão. O qual cumpre fazer notar, é uma adaptação do SIGAA (sistema de gestão acadêmica amplamente utilizada nas universidades federais existentes no Brasil).

Demonstrar que uma solução de integração entre o AVA Moodle e um Sistema de Gestão Acadêmica utilizando os padrões SOA (*Service-Oriented Architecture*) ou REST (*Representational State Transfer*) de construção de *Web Service*, pode ser generalizada para a interconexão de mais de um AVA.

## TECNOLOGIAS DE INTERCONEXÃO

A Plataforma Moodle é uma ferramenta de trabalho do professor e permite sobre diversos aspectos gerir o ensino. O seu uso atende as demandas da gestão de atividades pedagógicas do corpo docente. O seu uso permite a implantação de diferentes abordagens de construção do conhecimento. Contudo, não atende as demandas de gestão acadêmica e administrativa.

### Da Arquitetura da Solução

O aplicativo integrador em questão foi construído para permitir que o software integrado de gestão acadêmica e administrativa da Universidade Estadual do Maranhão se conecta online e com a solução AVA que suporta as ações de ensino à distância da universidade, o Moodle.

A arquitetura de software adotada pela universidade tem como seus principais direcionadores os seguintes pontos:

- Sistema orientado a objeto;
- Banco de dados relacional;
- Adoção de Arquitetura Orientada a Serviço; (SOA);
- Adoção de sistemas baseados em *WebServices*;
- Integração de todos os sistemas que atendem à comunidade universitária.

Torna-se relevante, portanto, discutir a arquitetura de software adotada e abordar aspectos sobre Arquitetura Orientada a Serviço (SOA).

Posteriormente far-se-á a apresentação de conceitos fundamentais sobre *WebService* incluindo os Protocolos SOAP e REST. Logo após explica-se conceitos que envolvem, Sistema Integrado de Gestão Acadêmica (SigUema Acadêmico), ambiente virtual Moodle. Por fim apresentaremos o Modelo conceitual de Integração entre o Moodle e o SigUema Acadêmico).

A figura a seguir explicita em diagrama de blocos os elementos básicos desta arquitetura dispondo-os em camadas inter-relacionadas.



**Fig 01: Diagrama de Blocos relacionando as camadas de tecnologias e serviços da arquitetura do Integrador.**

### Arquitetura Orientada a Serviço (SOA)

Portanto comecemos por estabelecer que SOA – do inglês *Service-Oriented Architecture* (Arquitetura Orientada a serviço), é um paradigma de desenvolvimento de aplicações cujo objetivo é criar módulos funcionais chamados serviços com baixo acoplamento, permitindo a reutilização de código, Sampaio [18]. SOA estabelece um modelo arquitetônico que visa aprimorar a eficiência, a agilidade e a produtividade de uma empresa através de serviços, Thomas [19]. Em específico, através do compartilhamento de serviço entre sistemas. SOA é, de fato, uma nova maneira de desenvolver aplicações.

Suas origens lançam raízes na orientação a objetos tradicional. A diferença entre esses dois paradigmas consiste no fato que na orientação a objetos, o que se deseja quando se cria uma classe, é caracterizar o objeto em si através de propriedades e métodos. Na orientação a serviço o foco principal não é um objeto em si, mas sim os serviços prestados por um conjunto de objetos.

A computação orientada a serviço representa, portanto, uma nova abordagem para plataforma de computação distribuída na medida em que permite o compartilhamento de serviços entre sistemas diferentes. Um serviço, do ponto de vista da arquitetura SOA, é uma função de um sistema computacional que é disponibilizado para outro

sistema ou para consumo interno ao sistema. Um serviço deve funcionar de forma independente do estado de outros serviços, exceto nos casos de serviços de processos (*process services*), e deve possuir uma interface bem definida. Normalmente, a comunicação entre o sistema cliente e aquele que disponibiliza o serviço é realizada através de *WebServices*.

Os elementos essenciais para alcançar o sucesso em SOA são: Visão conceitual, serviços, tecnologia, governança e estratégia, indicadores e cultura, e comportamento, Marks<sup>[8]</sup>.

Os serviços são os artefatos centrais de SOA. Os serviços devem ser definidos no contexto de um modelo de projeto de serviços que assegure reutilização, interoperabilidade e integração por todos os processos de negócio e plataforma de tecnologias.

A tecnologia deve ser disponibilizada para atingir os seguintes objetivos:

- Permitir que os serviços operem de forma confiável e segura;
- Permitir evoluir na arquitetura de Tecnologia da Informação – TI existente.

## Web Services

Quanto aos *webservices*, segundo Menéndez<sup>[10]</sup>, há uma definição bastante simples: É um componente de aplicação que aceita solicitações de outros sistemas através da Internet. Já segundo Kulchenko<sup>[6]</sup>, *WebServices* são interfaces acessíveis de rede, para as funcionalidades da aplicação, que utilizam em sua construção tecnologias padrões da Internet.

*WebService* é um componente de software que se comunica usando protocolos abertos e que usa como base o padrão XML. Sua principal funcionalidade é prover integração entre aplicações distintas através da World Wide Web utilizando componentes, especificações e padrões tecnológicos que são bem definidas no mercado de TI como um todo.

O funcionamento dos *WebServices* é baseado na interação de três agentes, os quais são: Provedor de Serviços, Consumidor de Serviços e Registro dos Serviços. A interação destes agentes envolve as operações de publicação, pesquisa e ligação. Vejamos a definição de cada uma destas funções, segundo Kreger<sup>[5]</sup>:

- Provedor de serviços: o provedor de serviços é a entidade que cria o Web Service. Ele disponibiliza o serviço para que alguém possa utilizá-lo. Mas, para que isto ocorra, ele precisa descrever o *WebService* em um formato padrão, que seja compreensível para qualquer um que

precise usar esse serviço, também publicar os detalhes sobre seu *WebService* em um registro central que esteja disponível.

- Consumidor de serviços: qualquer um que utilize um *WebService* criado por um provedor de serviços é chamado de consumidor de serviços. Este conhece a funcionalidade do *WebService* a partir da descrição disponibilizada pelo provedor de serviços, recuperando os seus detalhes através de uma pesquisa sobre o registro publicado. Através desta pesquisa, também o consumidor de serviços pode obter o mecanismo para ligação com este *WebService*.
- Registro dos serviços: Um registro de serviços é a localização central onde o provedor de serviços pode relacionar seus *WebServices*, e no qual um consumidor de serviços pode pesquisá-los. O registro dos serviços contém informações como detalhes de uma empresa, quais os serviços que ela fornece e a descrição técnica de cada um deles.

De modo que, em resumo, o provedor de serviços defina a descrição do serviço para o *WebService* e a publica para o consumidor de serviços no registro de serviços. O consumidor de serviços utiliza a descrição do serviço publicada para se ligar ao provedor de serviços e invocar ou interagir com a implementação do *WebService*.

## SOAP e REST - Protocolos de comunicação

### SOAP

A especificação SOAP (Simple Object Access Protocol) é um protocolo de transporte que rege a troca de mensagens entre aplicações em ambientes distribuídos e descentralizados. Seu conteúdo é composto por informações e estruturas de dados [5; 3].

A especificação SOAP 1.2 contém três partes principais: (I) **Modelo de empacotamento**, envelope SOAP, define o conteúdo da mensagem SOAP; (II) **Mecanismo de serialização**, conjunto de regras de codificação, define os tipos de dados utilizados na aplicação; (III) **Mecanismo de comunicação**, convenção, define as chamadas e respostas através de procedimentos remotos.

Toda mensagem SOAP é um documento XML. Ela obrigatoriamente contém os elementos <Envelope> e <Body> e, opcionalmente, os elementos <Header> e <Fault>.

O elemento <Envelope> é a raiz do documento XML e representa a mensagem propriamente dita; já o elemento <Body> incorpora a carga de informações (operações e parâmetros) que são entregues ao destinatário da mensagem. Este último, de acordo com a especificação SOAP, pode conter um elemento <Fault>, que, quando presente, pode ser utilizado no processamento de falhas do serviço Web. O elemento <Fault> descreve erros de chamada de métodos remotos ou mantém informações acerca do tipo de erro. Portanto, ele conta com os elementos <FaultCode>, <FaultActor>, <FaultString> e <Detail>.

O elemento <Header> expande uma mensagem SOAP. Ele define algumas características opcionais e acordos negociáveis entre as partes; seu conteúdo deve ser aceito pelas aplicações que estiverem se comunicando.

Newcomer<sup>[15]</sup> considera a especificação SOAP um tipo de extensão do protocolo HTTP, pois ela envia e recebe mensagens XML através das operações requisição e resposta do protocolo HTTP. Para que este processo ocorra, é necessário utilizar um servidor de páginas que seja capaz de atuar como um processador SOAP.

## REST

REST é um estilo de desenvolvimento de *WebServices* que teve origem na tese de doutorado de Roy Fielding<sup>[2]</sup>. Este, por sua vez, é co-autor de um dos protocolos mais utilizados no mundo, o HTTP (*HyperTextTransferProtocol*). De modo que se pode inferir que o protocolo REST é guiado, dentre outros preceitos, pelo que seriam as boas práticas de uso de HTTP.

Serviços baseados no paradigma de transferência de estado representativo, do inglês *Representational State Transfer* (REST), constituem uma implementação parcial de uma arquitetura orientada a serviços (SOA), e encontraram um sucesso ainda maior do que seus irmãos de peso-pesado, que são baseados no WSDL (Web Services Description Language) e SOAP (Simple Object Access Protocol).

Essa transferência de estado representativo é geralmente usada para descrever qualquer interface que transmite dados de um domínio específico sobre HTTP sem uma camada adicional de mensagem como SOAP. No entanto, é possível desenvolver um sistema de software de acordo com as restrições impostas pelo estilo arquitetural REST sem usar HTTP e sem interagir com a Web. Também se torna possível projetar interfaces HTTP + XML que não condizem com os princípios REST de Fielding [2]. Sistemas que

seguem os princípios REST são referenciados também como “RESTful”.

No entanto, REST não é tão simples quanto simplesmente utilizar HTTP – existem regras que devem ser seguidas para se realizar uso efetivo deste protocolo.

Segundo Fielding<sup>[2]</sup>, para que os princípios deste estilo sejam respeitados, um conjunto de restrições deve ser seguido. O primeiro deles é a caracterização como **Cliente-Servidor**. Esta característica é mais comumente encontrada em aplicações Web. Um servidor, com um conjunto de serviços disponíveis, escuta requisições a estes serviços. Um cliente, que deseja que um serviço disponível no servidor seja executado, envia uma requisição para o servidor; o servidor então pode tanto rejeitar como executar o serviço solicitado, e retornar uma resposta ao cliente.

**Stateless (Sem estado)** é outro dos princípios restritivos imposto pelo estilo REST e diz respeito à interação entre cliente e servidor. A comunicação deve ser feita sem o armazenamento de qualquer tipo de estado no servidor, ou seja, cada requisição do cliente para o servidor deve conter todas as informações necessárias para que ela seja entendida. Portanto, estados de sessão, quando necessários, devem ser totalmente mantidos no cliente.

**Cache** é o princípio que estabelece uma forma de diminuir o impacto da desvantagem trazida pela redução de desempenho. O mesmo exige que os dados de uma resposta, vindos de uma requisição ao servidor, sejam marcados como *cacheable* ou *noncacheable* (passíveis ou não de utilização da cache). Se uma resposta é marcada como *cacheable*, então ela será utilizada como resposta para as futuras requisições equivalentes.

A característica central que diferencia o estilo arquitetural REST, a **Interface Uniforme**, é a ênfase em uma interface uniforme entre os componentes (cliente, servidor). Com o objetivo de obter uma interface uniforme, REST define quatro requisitos de interface: (I) identificação de recursos; (II) manipulação de recursos através de representações; (III) mensagens autodescritivas e; (IV) hipermídia como mecanismo de estado da aplicação.

**Multicamada** é outra característica e tem o intuito de aperfeiçoar o requisito de escalabilidade da Internet. Com o princípio Multicamada foi adicionada ao estilo REST a característica de divisão em camadas. Sistemas multicamada utilizam camadas para separar diferentes unidades de funcionalidade. A principal desvantagem deste modelo está na adição de overhead e latência nos dados processados, reduzindo a desempenho. No entanto em um sistema baseado em rede que

suporte cache, esta desvantagem pode ser amenizada.

**Code-On-Demand**, por último, é uma característica opcional. Com ele REST permite que clientes tenham a funcionalidade de baixar e executar diretamente código no lado cliente.

Ademais para interagir com as URL's utilizamos os métodos HTTP. A regra de ouro para esta interação é considerar que URL's são substantivos e métodos HTTP são verbos. Isto quer dizer que os métodos HTTP são os responsáveis por provocar alterações nos recursos identificados pelas URL's. O HTTP provê uma interface uniforme com quatro métodos básicos para as quatro operações mais comuns. **GET** para recuperar uma representação de um recurso, **PUT** para criar um novo recurso ou modificar um existente, **DELETE** para deletar um recurso e **POST** utilizado para criação de um novo recurso.

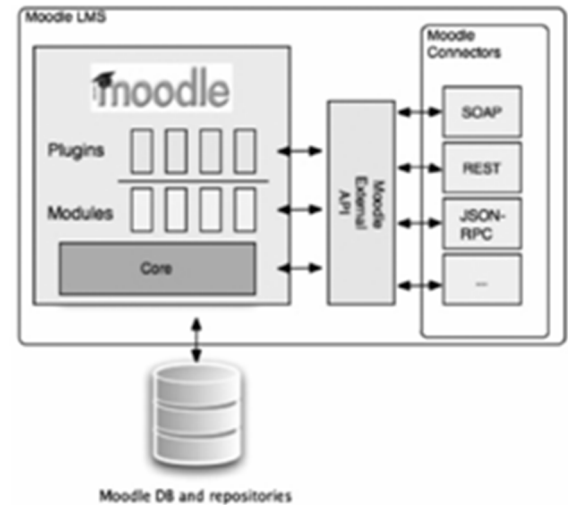
Note-se que os quatro métodos principais podem ser diretamente relacionados a operações de bancos de dados. E, no entanto, todas estas operações são lógicas. Isto quer dizer que utilizar um método não significa necessariamente excluir o dado do banco de dados; significa, apenas, uma exclusão lógica.

Outro dos pilares de REST é o uso de media types para alterar as representações de um mesmo conteúdo, sob perspectivas distintas. Esta ótica fica evidente quando se responde à pergunta: - Qual lado do recurso que se está procurando se quer enxergar? Precisa-se de uma foto, ou da descrição deste? Assim, ao realizar uma busca por um objeto no sistema, pode-se tanto desejar um XML com os dados deste objeto, quanto um JSON quanto uma foto. Com REST diferentes representações podem ser feitas utilizando-se a mesma URL.

Em uma abordagem pragmática, Richardson e Rubys [17] defendem que é interessante oferecer distinções entre esses tipos de dados na própria URL, a partir de extensões (similar às extensões de arquivos no sistema operacional). Observa-se então que esta abordagem possui a vantagem de que a realização de testes é simplificada, já que podemos testar os resultados pelo navegador; ao passo que a implementação pode se tornar mais complexa, além de induzir diferenciações nas implementações acarretando lógicas distintas para buscar dados semelhantes.

### CONCEITO GERAL DO INTEGRADOR

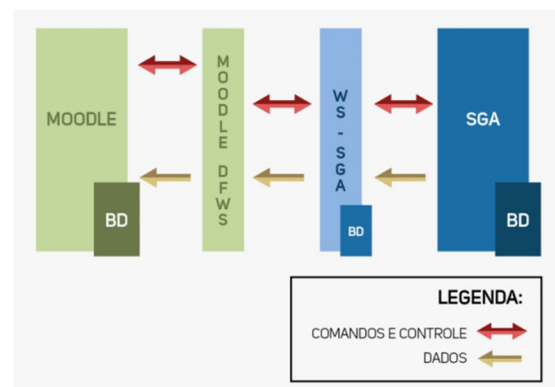
Já em 2008 os responsáveis pelo projeto Moodle propuseram uma arquitetura de conexão com outros sistemas que no momento de execução do projeto aqui descrito pode ser representada da seguinte maneira:



**Fig 02: Moodle: Proposta de Interconexão**

*Citado por Moura e Bernardino (2010)*

Dada a importância dos Sistemas Integrados de Gestão (SIG) para as instituições de ensino, Moura e Bernardino [14] propuseram um modelo genérico de integração onde são identificados quatro componentes distintos: (I) Moodle, LMS – Learning Management System; (II) SGA – Sistema de Gestão Acadêmica; (III) Moodle-DFWS – infraestrutura de *WebServices* do Moodle; (IV) WS-SGA – *WebServices* de acesso ao Sistema de Gestão Acadêmica.



**Fig 03: Modelo Genérico de interconexão proposto por Moura e Bernardino (2010)**

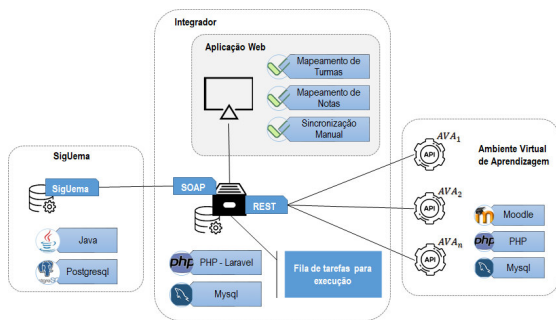
Nesta proposição genérica, Moura e Bernardino [14], prescrevem “A infraestrutura de *WebServices* do SGA permite proteger o SGA e simultaneamente compatibilizar o modelo com qualquer SGA”.

A modelagem do integrador desenvolvido na Uema parte deste modelo geral e a ele aplica duas restrições, a saber: (I) Devido questões arquitetônicas nativas do SigUema foi necessário

optar por uma estratégia híbrida de tecnologias de serviços. A comunicação com o integrador ocorre utilizando o protocolo SOAP. E além do SOAP, o integrador utiliza o REST para encaminhar as ações recebidas aos ambientes virtuais. (II) Devido a conflitos existentes entre as estruturas de dados do SigUema e do Moodle se fez necessário criar no integrador condições para corrigir por operação realizada pelo usuário estas distorções.

Com estas considerações a vista a arquitetura de integração do sistema de gestão acadêmica, SigUema, com o ambiente virtual de aprendizagem, Moodle, foi desenvolvida orientada a serviços com uma camada intermediária que orquestra as atualizações nas duas pontas. Esta camada, o integrador, recebe requisições do SigUema conforme alimentação de dados operacionalizados no sistema acadêmico pela equipe responsável. Essa alimentação pode ser criação de turma, matrícula, cancelamento, dentre outros. Uma vez recebida estas requisições, o integrador realiza algumas validações dos dados e organiza-os em fila para processamento. Por fim, os dados são enviados para a API do Moodle onde são interpretados e executados.

Além disso, o integrador possui uma aplicação Web com algumas parametrizações que auxiliarão na operação de integração: configuração de ambientes virtuais, mapeamento de turma associando o AVA com a turma em questão, e o mapeamento de notas onde será informado os identificadores de notas para sua devida recuperação da informação. Um painel com as requisições e seus status de execução são exibidos para acompanhamento dos processos. Por este painel será possível identificar as execuções com falhas e poder tratá-las. Na Figura abaixo é possível observar a arquitetura definida do processo de integração entre os sistemas SigUema e o Moodle com suas tecnologias distintas.



**Fig 04: Arquitetura do Integrador desenvolvidos neste trabalho.**

## AMBIENTE VIRTUAL DE APRENDIZAGEM

Buscando uma conceituação, por ambientes podemos entender o conjunto das pessoas, natureza ou coisas, inclusive objetos técnicos. Já o virtual vem do latim *medieval virtualis*, derivado por sua vez de *virtus*, força, potência. Utiliza-se a expressão virtual para que signifique alguma coisa que não tem materialidade.

Segundo Lyceum [7] um Ambiente Virtual de Aprendizagem (AVA) é um software que utiliza técnicas pedagógicas que possam se expressar através de ferramentas online, como a internet, para o ensino.

Outro termo utilizado para o AVA é o LMS (Learning Management System), estes dois termos são considerados sinônimos. Muitos autores utilizam também o termo EaD (Educação à Distância) para definir um AVA. No entanto, o EAD engloba todos os meios que aproximam as figuras de professores e alunos, que estejam fisicamente separados, utilizando qualquer meio, que não presencial, para a tarefa de ensino-aprendizagem, tais como: CD, DVD, TV, correspondências e também através da internet. No AVA, a distância física entre alunos e professores é aproximada somente através da Internet.

Neste sentido pode-se afirmar que um ambiente virtual é um *locus* onde seres humanos e objetos técnicos interagem potencializando assim, a construção de conhecimentos, logo a aprendizagem.

Neste trabalho referenciaremos sempre o AVA, suportado apenas pela a internet, como meio de auxílio ao Ensino-Aprendizagem.

Um Ambiente Virtual de Aprendizagem pode ser inserido em um projeto pedagógico para ampliar o espaço da aula, mas ele é absolutamente necessário quando se trata de Educação a Distância.

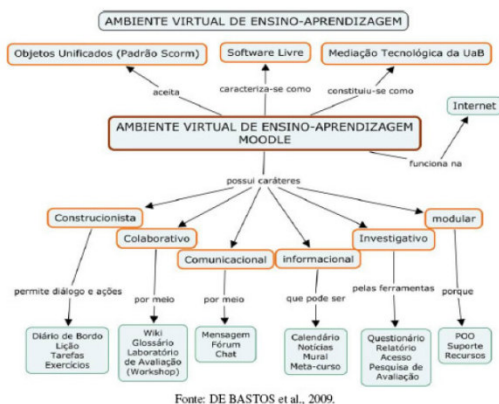
Algumas vantagens dos AVAs na perspectiva de vários autores como Duggleby [4] são: (I) **Flexibilidade temporal**, onde os cursos e serviços estão disponíveis 24 horas, proporcionando ao educando adaptar seu processo de aprendizagem à sua disponibilidade de horário. (II) **Flexibilidade no acesso a aprendizagem**, caracterizada pela possibilidade de o aluno aprender a um ritmo próprio, aprofundar os conteúdos, associar a teoria às múltiplas atividades práticas e escolher o método que melhor se adapta ao seu estilo e possibilidades. (III) **Independência Geográfica** é possível por meio da educação à distância, em particular o *e-learning*. Derrubam-se, nesta perspectiva, fronteiras de espaço abrindo-se caminhos de construção do conhecimento para as

peças que tenham dificuldades de frequentar o ensino presencial. (IV) Estimular a **Autoaprendizagem**, permitindo o desenvolvimento pessoal e contínuo dos indivíduos, conferindo-lhes maior autonomia.

No entanto, para Duggleby [4], as desvantagens apresentadas pelo *e-learning* são: (I) **Ausência de Relação Humana** entre educador e educando a qual pode traduzir-se em isolamento, desmotivação, baixa autodisciplina levando a resultados nada satisfatórios. (II) **Foco em Conteúdos Genéricos**, onde a componente prática é menor. Pois se constata, na prática, que o investimento em simuladores tem custo elevado. (III) **Dificuldade de Adaptação** a todos os perfis de destinatários. O EaD é uma das modalidades de ensino que exige muita responsabilidade e maturidade, por parte do aluno. Tais fatores podem ser mais essenciais para a eficácia dessa estratégia de aprendizagem do que a formação ou experiências prévias no uso de computadores e de utilização de Internet. (IV) **Exigência de Conhecimentos Tecnológicos**. A falta de habilidade com as tecnologias e Internet por parte dos alunos, torna muito difícil a adaptação do AVA, para estes, o uso de AVA como estratégia ideal para aprimorar os seus conhecimentos.

Para além destes conceitos introdutórios, há um leque muito grande de ofertas de Ambientes Virtuais de Aprendizagem voltados para diferentes públicos. Barboza, Sales Filho e Silva [1] citam como plataformas em uso no contexto brasileiro o Moodle, o Blackboard, o TelEduc, o WEB Ensino, o Web Course Tools – WeBCT, e o e-Proinfo. Dando relevância ao Moodle e ao Blackboard. O integrador descrito neste artigo foi executado tendo o Moodle como um dos lados integrados tendo em vista a escolha feita em tempos mais remotos pela Universidade Estadual do Maranhão.

A figura a seguir faz uma caracterização conceitual do Moodle.



**Fig 05: Moodle: Características e funcionalidades**

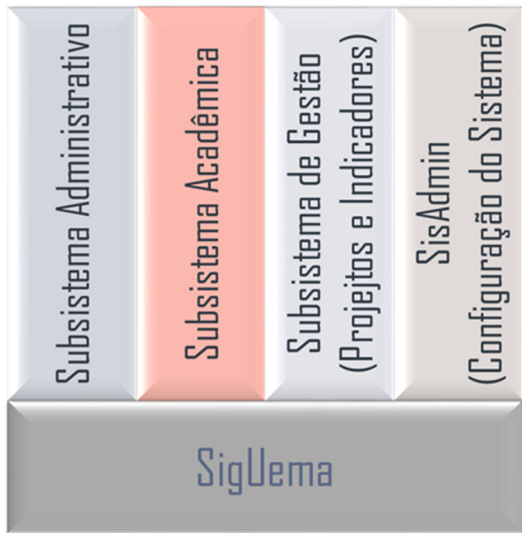
### SISTEMAS DE GESTÃO ACADÊMICA

Os sistemas de gestão acadêmica são sistemas de informação que, normalmente, disponibilizam funcionalidades de controle quanto a dados sobre registros de discentes, docentes, cursos, perfis curriculares: disciplinas, requisitos, equivalências, associações; oferta de turmas a cada período letivo; pré-matricula e matrícula de discentes; lançamento de notas on-line pelos docentes; histórico escolar; registro de diplomas, entre outras funcionalidades que permitem gestão administrativa e acadêmica das instituições de ensino.

Tais sistemas de gestão acadêmica propõem-se a registrar, gerenciar e tornar ágeis os processos da instituição, possibilitando a consolidação de informações importantes para a gestão, por meio da análise de dados, tais como matrículas, aproveitamento acadêmico, frequência, evasão, entre outros indicadores.

Estão disponíveis num sistema de gestão acadêmica de uma instituição de ensino todas as informações referentes à vida acadêmica do aluno e do professor. Isso permite que diretores contem com uma radiografia completa das atividades da instituição permitindo monitorar o desempenho acadêmico e financeiro, além de permitir planejar a evolução futura e tomar decisões, Lyceuem [7].

Existem várias instituições de ensino que desenvolveram sistemas próprios de gestão acadêmica. Em particular a Universidade Estadual do Maranhão, que optou por adquirir um software amplamente utilizado em universidades públicas no Brasil para lhe servir de sistema integrado de gestão (SIG). O sistema tem diversos subsistemas integrados entre si através da arquitetura definida anteriormente como mostra o quadro a seguir.



**Fig 06: SigUema: Subsistemas que o compõe, dentre eles o SGA.**

O subsistema de gestão acadêmica inclui o módulo de gestão do ensino à distância. A integração dos diversos módulos e subsistemas do SigUema é caracterizada adoção do SOA utilizando protocolos SOAP para a construção de *WebServices* internos ao sistema.

Neste ponto cumpre-nos fazer notar que a utilização de um AVA externo não é mandatório no SigUema, ou em seus sistemas irmãos SIGAA. Pois é este SGA dotado de um mini AVA interno, a Turma Virtual. Contudo na Uema, para os alunos do ensino a distância, fez-se a escolha por um AVA mais completo deixando para os alunos presenciais o uso da Turma Virtual.

### O INTEGRADOR DESENVOLVIDO

Um projeto de integração de sistemas de natureza tão diversa, embora necessário, certamente é um projeto de alta complexidade e com importantes riscos a serem considerados já em seu nascedouro. Com base nesta constatação buscou-se identificar os principais atores e um consenso entre eles sobre quais requisitos seriam incorporados ao projeto de software do integrador.

Obtido o consenso estavam listados os seguintes requisitos associados às condições gerais da instituição de ensino, às normas e procedimentos associados a gestão acadêmica, às necessidades específicas da gestão do ensino à distância, e aos condicionantes tecnológicas de implementação do software.

As condições gerais da instituição de ensino apontaram para: (I) a necessidade de a universidade possuir e operar um sistema integrado e centralizado que lhe permita registrar as atividades meio e finalísticas necessárias ao

cumprimento de sua missão institucional; (II) obrigação de registrar as operações abrangidas pelo integrador de modo seguro, confiável e com o as automatizações necessárias a atender as premissas de operação das diversas estruturas gerenciais que a comunidade universitária; (III) o dever de construir os objetos de software que integrarão as estruturas de dados de tal maneira que atendam às necessidades de mineração de dados que comporão as ferramentas *business intelligence* projetadas para a Uema.

### O Integrador

Quanto a sincronicidade, a integração entre o SigUema e o Ambiente Virtual de Aprendizagem é realizada de forma pervasiva e operando em períodos de tempo determinados de forma cíclica. São sincronizados todos os dados necessários para a disponibilidade do ambiente virtual. Dotado de códigos que compõe a interface do lado do SGA, de outra interface do lado AVA, e de um conjunto de funções para gerenciamento e configuração de si próprio, este sistema foi intitulado como Integrador.

O Integrador, embora bidirecional, é ativo do lado SigUema. Ou seja, é a base de dados central que aciona todos os serviços referentes às informações acadêmicas por meio do processo de integração. No entanto, no AVA, passivo, o processo de sincronização de notas ouvirá as mudanças ocorridas no ambiente virtual disponibilizando-as para transferência quando requeridas pelo lado ativo do SigUema. O integrador possui o papel de orquestrador sendo ele responsável por executar todos os processos de sincronização.

### A Estratégia

O integrador realiza consultas periódicas na base de dados do SigUema que contém os dados a serem sincronizados e a ação que deverá ser realizada. Por exemplo, cadastrar aluno ou incluí-lo no curso ou disciplina. Esses dados são reflexos de ações realizadas pelo usuário na aplicação do sistema acadêmico que será abordado mais adiante. Essa base de dados possui a seguinte estrutura:

Nome do campo	Descrição
id	Identificador único de registro
entidade	Descrição do recurso ao qual os dados se referem
id_siguema	Id de registro do dado no siguema
id_moodle	Id de registro do dado no AVA
data_criacao	Data de criação do registro de equivalência
data_alteracao	Data de alteração do registro de equivalência

**Tabela 01 – Assinatura de dados do webservice**

Para alimentar essa base, foi utilizado a abordagem de *triggers* (recurso de programação executado sempre que o evento associado ocorrer). Elas capturaram o momento que os recursos foram modificados, adicionados ou excluídos na aplicação do sistema acadêmico pelo usuário e preenchem a base de ações com as informações manipuladas e as ações que devem ser tomadas.

Uma vez de posse dos dados, o integrador identifica a ocupação do recurso com o campo `flag_processamento` evitando assim que outro conjunto de operações que se executam simultaneamente o ocupe. Em seguida, verifica a ação que deve ser realizada e solicita a API – *Application Programming Interface* do AVA para realizar tal ação. Na consulta de tarefas para serem executadas, o integrador fixa uma quantidade de registros da base restando outras tuplas para outros processos paralelos. Essa solução foi implantada para corrigir problemas de *deadlock* (situação em que ocorre um impasse, e dois ou mais processos ficam impedidos de continuar suas execuções) encontrados durante a implantação e contornado por meio do campo `flag_processamento`.

Ao registrar o recurso no AVA, os dados são retornados para o integrador. Assim, o sistema poderá associar o registro do SigUema com o

registro do ambiente virtual, para futuras manipulações. Esse registro é realizado na base de dados de equivalência. A tabela da equivalência possui a seguinte estrutura:

Nome do campo	Descrição
Id	Identificador único do registro
Entidade	Descrição do recurso aos quais os dados se referem. Por exemplo, professor, aluno, mediador EaD, etc.
Dados	Conjunto de dados relacionados à entidade.
Ação	Ação que deve ser feita no AVA (criar, atualizar ou excluir).
Data_hora_registro	Data e hora que o registro foi adicionado na base para sincronização.
Data_hora_sincronizacao	Data e hora que o registro foi sincronizado pelo integrador.
Flag_processamento	<i>Flag</i> que indica que esta ação está na fila de processamento de uma instância do <i>job</i> . Resolver problema de concorrência.

**Tabela 02 - Registro da equivalência de dados**

## CONCLUSÃO

O cenário tecnológico atual das instituições de ensino, e em especial as instituições de ensino superior, caracterizado por processos de constante inovação, aponta uma preocupação crescente para integrar os ambientes de gestão acadêmica com os ambientes virtuais de aprendizagem. Tal integração visa, na esfera administrativa das instituições de ensino, facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório; e, na esfera acadêmica destas mesmas instituições de ensino, obter dados que possam qualificar e quantificar os processos de aprendizagem dos discentes abrangidos. Demonstrou-se em nosso trabalho, em que pese as complexidades do projeto, que a integração bidirecional entre SGA e AVA é factível e que acarreta benefícios e auxilia o docente em sua prática, seja de planejamento, ensino, avaliação ou em sua ação reflexiva. Adicionalmente investigações posteriores haverão de comprovar que a integração dos Sistemas de Gestão Acadêmica com o Ambiente Virtual de Aprendizagem contribuirá para melhor integrar os

discentes na comunidade universitária. Também foi observado, durante a construção de nosso software integrador que o uso do SOA, através de *WebService*, possibilitou obter, entre maior interoperabilidade, maior agilidade organizacional e menor carga de trabalho do TI. Ficou também patente que, respeitada a mesma arquitetura de software, é possível construir a integração com outras plataformas, distintas do Moodle. Integrações futuras, a serem viabilizadas em outros trabalhos, permitiriam a uma instituição de ensino utilizar simultaneamente mais de um Ambiente Virtual de Aprendizagem sem acarretar impactos nas condições de gestão das instituições.

Por último a construção do software integrador mostra que em um ambiente de cooperação entre instituições de ensino e desenvolvedores de plataformas virtuais de aprendizagem seria possível construir interfaces de integração baseadas em arquitetura orientada a serviços genéricos o suficiente para minimizar o esforço de programação envolvido na construção de projetos semelhantes de integração a partir da generalização e padronização dos serviços web.

## REFERÊNCIAS

1. BARBOZA, E.J.S.; A.G.S., FILHO, SILVA, M.T. *Comparação entre os principais avas quanto a interatividade* Publicado em 2016. Disponível em: <[https://www.researchgate.net/publication/268078497\\_Ambientes\\_Virtuais\\_de\\_Aprendizagem\\_e\\_Ambientes\\_Imersivos\\_um\\_estudo\\_de\\_caso\\_utilizando\\_tecnologias\\_de\\_computacao\\_movel](https://www.researchgate.net/publication/268078497_Ambientes_Virtuais_de_Aprendizagem_e_Ambientes_Imersivos_um_estudo_de_caso_utilizando_tecnologias_de_computacao_movel)>. Acessado em 28/12/2018.
2. FIELDING, R.T. *Architectural Styles and the Design of Network-based Software Architectures*. Publicado em : 2000. Disponível em: <<https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>. Acessado em 21 de dezembro de 2017.
3. FRANKLIN COYLE. 2002. *XML, Web Services and data Revolution*. Boston.
4. JULIA DUGGLEBY. *Como ser um tutor online*. Universidade Federal do Paraná
5. KREGER, H. *Web services conceptual architecture (WSCA 1.0)* 2001. Disponível em: <<http://www.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>>. Acessado em 30 de Janeiro de 2018.
6. KULCHENKO, P.; TIDWELL, D.; SNELL, J. *Programming Web Services with SOAP*. Publicado em: 2001. Disponível em: <<https://www.oreilly.com/library/view/programming-web-services/0596000952/>>. Acessado em 22 de fevereiro de 2018.
7. LYCEUEM, L. Publicado em 2013. Disponível em: <<http://www.teche.com.br/pr odutos.asp?id=8>>. Acessado em 02 de Fevereiro de 2018.
8. MARKS, E.; MICHAEL B. Publicado em: 2006. *Service-Oriented Architecture: a planning and implementation guide for business and technology*. Disponível em: <<https://onlinelibrary.wiley.com/doi/book/10.1002/9781119201700>>. Acessado em 02 de Setembro de 2018.
9. MAXUEL DE PAIVA LUCENA. 2013. *Plataforma de Integração Moodle-Sistema de Gestão Acadêmica Baseada em Webservice*.
10. MENÉNDEZ, A. Publicado em: 2002. *Uma ferramenta de apoio ao desenvolvimento de Web Services*. Disponível em: <[http://docs.computacao.ufcg.edu.br/posgraduacao/dissertacoes/2002/Dissertacao\\_AndresInacioMar-tinezMenendez.pdf](http://docs.computacao.ufcg.edu.br/posgraduacao/dissertacoes/2002/Dissertacao_AndresInacioMar-tinezMenendez.pdf)>. Acessado em 21 de Agosto de 2018.
11. MORAES, R. Publicado em: 2000. *É possível ser construtivista no ensino de Ciências*. Disponível em: <[http://www.aedmoodle.ufpa.br/pluginfile.php?file=%2F238925%2Fmod\\_folder%2Fcontent%2F0%2FTEXT0%2003\\_ENSINO%20DE%20CI%C3%80NCIAS%20CONSTRUTIVISTA.pdf&forcedownload=1](http://www.aedmoodle.ufpa.br/pluginfile.php?file=%2F238925%2Fmod_folder%2Fcontent%2F0%2FTEXT0%2003_ENSINO%20DE%20CI%C3%80NCIAS%20CONSTRUTIVISTA.pdf&forcedownload=1)>. Acessado em 01 de Setembro de 2018.

12. MORAN, J. Publicado em :2002. *O que é educação a distância*..Disponível em: <<http://www2.eca.usp.br/mo-ran/wp-content/uploads/2013/12/dist.pdf>>. Acessado em 02 de Setembro de 2018.
13. MORAN, J.M. 2003. *Contribuições para uma pedagogia da educação online*. In: OLIVEIRA, E.G. *Educação a distância na transição paradigmática*. São Paulo: Papyrus, 2003.
14. MOURA,R.;BERNARDINO,J. Publicado em : 2010.*Um modelo para a integração de serviços: Moodle e Sistemas de Gestão Acadêmica*. Disponível em: <[https://www.researchgate.net/publication/258994898\\_Um\\_modelo\\_para\\_a\\_integracao\\_de\\_servicos\\_Moodle\\_e\\_Sistemas\\_de\\_Gestao\\_Academica](https://www.researchgate.net/publication/258994898_Um_modelo_para_a_integracao_de_servicos_Moodle_e_Sistemas_de_Gestao_Academica)>. Acessado em 03 de Janeiro de 2018.
15. NEWCOMER, E. Understanding web services: XML, WSDL, SOAP and UDDI. Boston: Addison-Wesley, 2002.
16. RICE, W. H. *Moodle: E-Learning Course Development - A complete guide to* SAMPAIO, C. *SOA e Web Service em Java*.Rio de Janeiro: Brasport, 2006.
17. RICHARDSON, L.; RUBY,S. *RESTful Web Services* . Publicado em :2007. Disponível em: <[https://www.crummy.com/writing/RESTful-Web-Services/RESTful\\_Web\\_Services.pdf](https://www.crummy.com/writing/RESTful-Web-Services/RESTful_Web_Services.pdf)>. Acessado em 05 de Janeiro de 2018.
18. SAMPAIO,C. Publicado em: 2006. *SOA e Web services em Java. Brasport*.Disponível em: <[https://books.google.com.br/books/about/SOA\\_e\\_Web\\_Services\\_em\\_Java.html?id=Y2btOCiUptQC&redir\\_esc=y](https://books.google.com.br/books/about/SOA_e_Web_Services_em_Java.html?id=Y2btOCiUptQC&redir_esc=y)>. Acessado em 12 de setembro de 2018.
19. THOMAS,ERL. *SOA: Princípios de design de serviços*. Publicado em:2009. Disponível em <<http://ebookbit.com/book?k=Soa+-+Princ%C3%ADpios+de+Design+de+Servi%C3%A7os&isbn=978857605189-3&lang=pt&source=firebaseapp.com#pdf>>. Acessado em 03 de janeiro de 2018.
20. ZACARIAS,A.;LIVONGUE.E.*Integração do Moodle com Sistemas de Gestão Acadêmica*. Publicado em 2018. Disponível em:[https://www.researchgate.net/profile/Edson\\_Livongue/publication/305463108\\_Integracao\\_do\\_Moodle\\_com\\_Sistemas\\_de\\_Gestao\\_Academica/links/578f92d708aec23468b65e84/Integracao-do-Moodle-com-Sistemas-de-Gestao-Academica.pdf](https://www.researchgate.net/profile/Edson_Livongue/publication/305463108_Integracao_do_Moodle_com_Sistemas_de_Gestao_Academica/links/578f92d708aec23468b65e84/Integracao-do-Moodle-com-Sistemas-de-Gestao-Academica.pdf). Acessado em 27/12/2017.