

## Componente de integração entre um ambiente de aprendizagem ubíqua e outras plataformas de educação

Danilo G. Carlos, Igor F. dos Santos, Ramiro de V. dos Santos Júnior, Bruno de S. Monteiro, Francisco M. Mendes Neto

Departamento de Computação (DC)  
Universidade Federal Rural do Semi-Árido (UFERSA) – Mossoró, RN – Brasil

daniologomescarlos@gmail.com, {igor.santos, ramiro.junior,  
brunomonteiro, miltonmendes}@ufersa.edu.br

**Abstract.** *Learning Management Systems (LMS) are software used in didactic activities by courses in the distance modality, as well as in the face-to-face modality. However, the adoption of other digital tools and practices are discouraged because there is no integration with the LMS adopted by the institution. In this context, the present work presents the conception and development of an integration component between Learning Management Systems and a ubiquitous learning environment, Edubi, that uses the Youubi API. Therefore, it is possible to enable a range of new practices, including informal learning. For its validation, a case study was carried out with Moodle LMS.*

**Resumo.** *Sistema de Gestão da Aprendizagem (SGA) são softwares utilizados nas atividades didáticas dos cursos na modalidade a distância, como também na modalidade presencial. Entretanto, a adoção de outras ferramentas digitais e práticas acabam sendo desencorajadas por não haver integração com o SGA adotado pela instituição. Neste sentido, o presente trabalho apresenta a concepção e desenvolvimento de um componente de integração entre Sistemas de Gestão da Aprendizagem e um ambiente de aprendizagem ubíqua, Edubi, que faz uso da Youubi API. Com isso, é possível habilitar um leque de novas práticas, inclusive voltadas ao aprendizado informal. Para sua validação, foi realizado um estudo de caso com o SGA Moodle.*

### 1. Introdução

Sistemas de Gestão da Aprendizagem (SGA), também conhecidos por *Learning Management Systems* (LMS), são softwares utilizados nas atividades didáticas dos cursos na modalidade educação a distância (EaD), como também nas modalidades semipresenciais e presencial. Sistemas desse tipo possuem, normalmente, as seguintes funcionalidades: disponibilização de arquivos e *links web*, normalmente pelo professor ou tutores; disponibilização de atividades, por exemplo, questionários e envio de arquivos; criação de fóruns de discussão; e monitoramento das interações dos alunos [Gomes et al. 2018].

Entretanto, a popularização das tecnologias digitais móveis, a valorização dos ambientes informais nas práticas educacionais e a sensibilidade ao contexto dos indivíduos, viabilizam o desenvolvimento de práticas ainda mais flexíveis, favorecendo

a aprendizagem em qualquer lugar, a qualquer momento, independente do tipo de dispositivo digital disponível e adequado às características do indivíduo [Cobo and Moravec, 2011]. Contudo, nem sempre o SGA adotado pela instituição de ensino possui flexibilidade suficiente para suportar o desenvolvimento de novos modelos de ensino-aprendizagem, visto que é normalmente voltada ao aprendizado formal. No entanto, muitas vezes o aprendizado acontece de maneira informal por meio das atividades realizadas pelo aprendiz no seu cotidiano e fora da plataforma de ensino oficial adotada pela instituição.

Neste sentido, a aprendizagem ubíqua (*ubiquity learning* ou *u-learning*) surge da combinação de modelos de aprendizagem flexíveis e tecnologias digitais com o objetivo de prover ferramentas que ampliam as oportunidades de práticas de ensino e aprendizagem. Ferramentas baseadas nesse conceito permitem oferecer aos aprendizes e professores conteúdos e serviços relevantes com base no contexto no qual os sujeitos estão inseridos [Monteiro et al., 2017].

Contudo, apesar das ferramentas de *u-learning* se apresentarem como soluções inovadoras, algumas características das plataformas de educação tradicionais, especialmente aquelas do tipo SGA, ainda são fundamentais para as instituições que ofertam cursos em EaD. Portanto, este trabalho vislumbra a perspectiva de que ambientes de *u-learning* possam complementar a experiência dos alunos e professores, proporcionando novas formas de interação e práticas de ensino-aprendizagem.

Diante desse cenário, o presente trabalho teve como objetivo conceber, desenvolver e avaliar um componente de integração entre Sistemas de Gestão da Aprendizagem e ambientes de aprendizagem ubíqua, de modo a contribuir com equipes de desenvolvimento que precisam consumir os serviços dessas plataformas, de um modo mais fácil, simples e eficiente.

Para validação, o componente proposto foi integrado ao Edubi [Edubi, 2018], por ser um ambiente de aprendizagem ubíqua composto por quatro aplicações clientes (*Web, Mobile, TV e Watch*) e consome os serviços da Youubi API [Monteiro et al., 2017]. Para viabilizar essa integração e garantir as propriedades de alta coesão e baixo acoplamento, os serviços do componente proposto são disponibilizados por meio de um *web service*. Além disso, no escopo deste trabalho, optou-se pela integração com o ambiente Moodle, por ser bastante difundido e por ser utilizado nos cursos da Universidade Aberta do Brasil. No entanto, a arquitetura do componente foi projetada para ser expandida para outras plataformas (AmadeusLMS, OpenRedu, entre outras).

O presente trabalho está organizado da seguinte forma: a Seção 2 apresenta o ambiente Edubi e a Youubi API; na Seção 3 são descritos aspectos relacionados aos *web services*; a Seção 4 descreve o componente de integração; a Seção 5 mostra a avaliação realizada sobre o componente; e, por fim, a Seção 6 apresenta as considerações finais.

## 2. Trabalhos correlatos

Em Furtado et al. (2015) é apresentado um modelo de integração entre SGAs baseado em arquitetura orientada a serviços. O modelo proposto disponibiliza um *web service* que realiza requisições aos bancos de dados dos sistemas integrados por meio de comandos SQL ou utiliza o protocolo SOAP para comunicação em *web services*.

Em Ouya et al. (2015) foi proposta uma ferramenta de integração entre um ambiente *e-learning* e redes sociais. Os autores desenvolveram um *plugin* para o

Moodle com a ferramenta de comunicação Big Blue Button visando permitir a troca de mensagens com outros usuários em outras redes sociais. Um middleware utiliza as APIs disponibilizadas por essas redes sociais para acessá-las e o protocolo REST através do *plugin* para o Moodle para acessar o Big Blue Button.

Raspopoviü et al. (2016) propõem uma arquitetura para integração de um ambiente de aprendizagem institucional, um sistema de gestão acadêmica e um ambiente de aprendizagem social. Essa integração permite o compartilhamento de objetos de aprendizagem, informações sobre o currículo e interações sociais dos usuários entre os ambientes integrados. A comunicação entre os três sistemas é realizada por meio de *web services*.

O presente trabalho difere-se dos demais por constituir um componente de integração que provê integração de dados entre SGAs e Ambientes de Aprendizagem Ubíqua, por meio de *web service*. O componente é desacoplado dos sistemas integrados permitindo que novos softwares possam ser incluídos. Além disso, qualquer sistema poderá consumir as funcionalidades do componente.

### 3. Ambiente de aprendizagem Ubíqua Edubi e Youubi API

O ambiente Edubi [Edubi, 2018], representado na Figura 1, é composto por quatro aplicações clientes (*Web, Mobile, Watch e TV*), cada uma delas com suas particularidades, e dois componentes de integração onde um deles possibilita a essas aplicações consumirem os serviços disponibilizados por Moodle, e o outro descrito em Bezerra et al. (2017), permite a recomendação de conteúdos da Wikipedia e Youtube. Além desses, os artefatos de software do ambiente Edubi consomem também os serviços da Youubi API [Monteiro et al., 2017], componente central da arquitetura. Dentro desse contexto, o presente trabalho se concentrou especificamente na concepção e desenvolvimento do componente “Integrador de LMS”, responsável por consumir alguns serviços de uma instância Moodle.

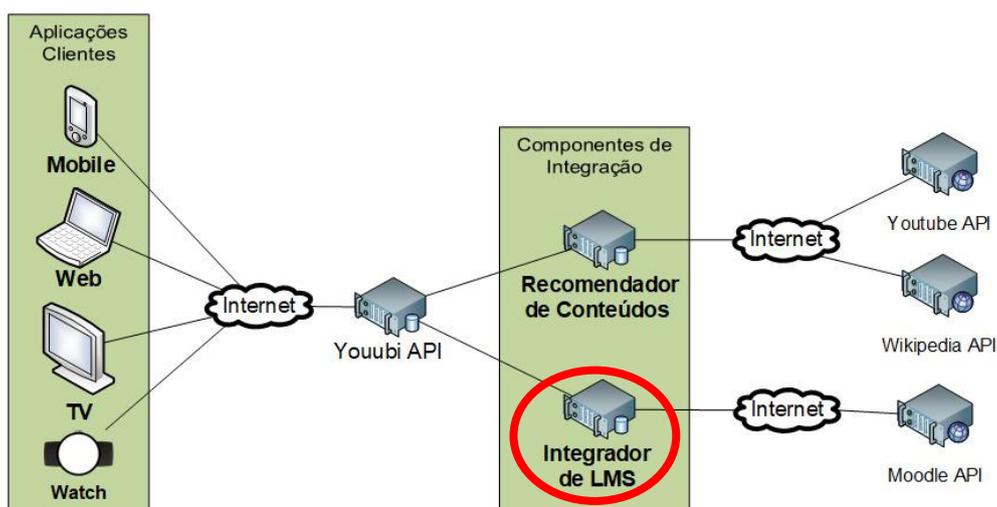


Figura 1. Visão geral do ambiente Edubi [Edubi, 2018]

O Youubi [Monteiro et al., 2017] é uma API (*Application Programming Interface*), desenvolvida com a linguagem Java, que fornece serviços que auxiliam o desenvolvimento de aplicações clientes voltadas à computação ubíqua, mas que pode ser adotada em diversos domínios específicos (educação, saúde, entretenimento, turismo,

comunicação, marketing, etc). Ela permite a criação e manipulação de sete Entidades Elementares (EE): pessoa, postagem, evento, pergunta, local, grupo e missão. Atualmente estão disponíveis 154 serviços (78 requisições post e 76 requisições get), que possibilitam às aplicações clientes implementarem as seguintes funcionalidades: autoria de conteúdo (postagens, eventos, perguntas e locais); recomendação das EE; rede social; comentários; chat; grupo de discussão (por meio da entidade grupo); notificações; gamificação (pontos de experiência, medalhas e rankings); missões (sequência de objetivos); geolocalização; bookmark (lista de conteúdos salvos); monitoramento quantitativo de interações; e moderação.

Conforme pode ser observado na Figura 2, a arquitetura da Youubi API é composta por seis módulos principais: *Communication*: representa a camada de comunicação, pois permite que as aplicações clientes utilizem a Youubi API para requisitar os serviços do Youubi Server. O acesso a API é feito por controle de sessão, ou seja, os serviços da API requerem o ID do usuário e um token de sessão (disponibilizado após o login); *Manager*: é considerado o núcleo do servidor por desempenhar importantes funções: tratar as requisições passadas pela camada de comunicação, gerenciar as regras de domínio definidas na Youubi API, distribuir demandas aos outros módulos, entre outras responsabilidades; *Collector*: é responsável por coletar dados em outros sistemas que disponibilizam uma API de acesso. Os conteúdos coletados automaticamente por esse módulo, em plataformas externas, precisam ser validados por um moderador para que finalmente possam ficar visíveis para os demais usuários. *Recommender*: é responsável por gerar uma lista de objetos recomendados para os usuários, composta por todas as entidades elementares; *Model*: é responsável por representar o modelo de dados das entidades da arquitetura; e *Persistence*: é responsável pela persistência das informações no banco de dados (BD).

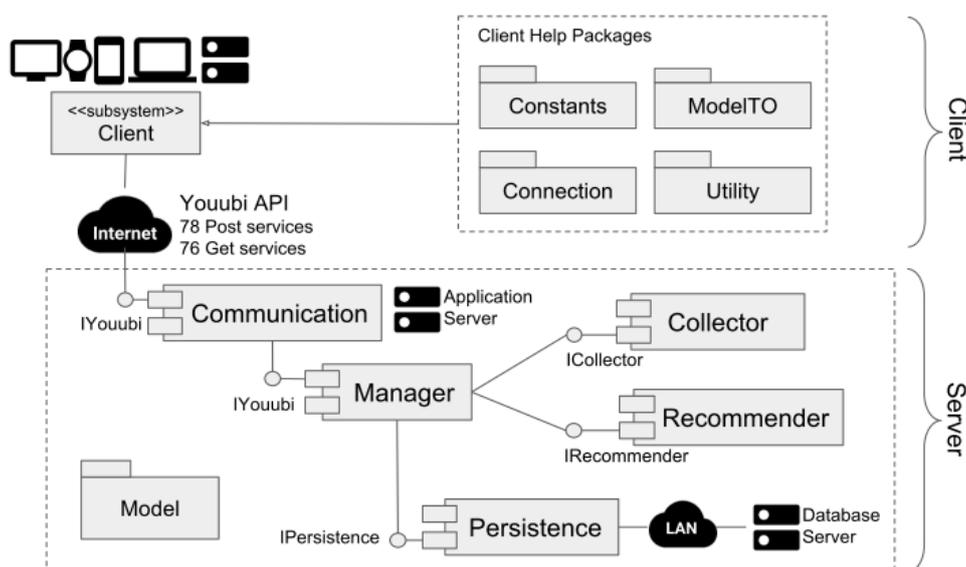


Figura 2. Arquitetura da Youubi API (adaptado de [Monteiro et al., 2017]).

#### 4. Web Services

Enquanto a *web* fornece uma solução para uma interação flexível entre humanos e máquinas, os *web services* visam uma interação flexível entre máquinas [Teknillinen, 2008]. De acordo com a W3C (2004), um *web service* é um software projetado para suportar interações entre máquinas através de uma rede. Na prática, um *web service*

fornece um padrão de interoperabilidade permitindo que diferentes softwares possam consumir seus serviços, independente da plataforma na qual foi desenvolvido. Em um nível técnico, existem diversas maneiras de se implementar um *web service*, no entanto, existem dois tipos de que se destacam: os *Big Web Services* e os *RESTful Web Service*.

Os *Big Web Services* utilizam uma pilha de tecnologias (SOAP, WSDL, entre outras) conhecida como “WS-\*”, para prover interoperabilidade tanto por troca de mensagens quanto por chamadas a procedimentos. Os *RESTful Web Services* se apresentam como alternativa mais “leve” em comparação aos *Big Web Services*. O REST foi designado originalmente como um estilo arquitetônico para a construção de sistemas hipermídia distribuídos em larga escala. Sua simplicidade é atribuída devido a utilização de padrões já conhecidos na *web* e a sua infraestrutura necessária já ter se tornado difundida. Essa infraestrutura possui um custo reduzido e, portanto, tem pouca resistência em sua utilização [Pautasso, Zimmermann and Leymann, 2008]. Portanto, devido suas características, este foi o modelo considerado mais adequado para o contexto do corrente projeto.

## 5. Resultados: Desenvolvimento do Componente LMS e *Plug-in Moodle*

O componente desenvolvido neste trabalho consiste de um subsistema da arquitetura do ambiente Edubi (Figura 1), que tem como objetivo receber uma solicitação da Youubi API e realizar requisições a serviços disponibilizados por outros ambientes de aprendizagem por meio da Internet. Sua codificação foi feita na linguagem de programação *Java*, e para garantir as propriedades de alta coesão e baixo acoplamento, seus serviços são disponibilizados por meio de um *web service*, conforme representado pelo “Integrador de LMS” da Figura 1. Desta forma, o componente LMS pode consumir os serviços da Youubi API, como também o Youubi pode consumir os serviços oferecidos pela componente LMS.

A Youubi API possui um módulo *Controller* (Figura 2), que é responsável por solicitar informações a serviços externos (por exemplo, Google Place, Youtube e Wikipedia). Portanto, da mesma forma, este módulo solicita ao componente LMS a execução de uma determinada função previamente criada, envia as informações sobre o serviço que deseja requisitar e os parâmetros disponíveis. O Componente LMS identifica os parâmetros necessários para invocar o serviço e valida os parâmetros enviados pelo *Controller*. Em seguida, monta uma requisição ao serviço com as informações enviadas pelo *Controller* e solicita sua invocação. Em posse dos resultados, o Componente LMS encapsula os dados de resposta em uma entidade elementar do Youubi e retorna ao *Controller*, que fará o devido uso da entidade gerada.

Para armazenar as informações de como acessar os serviços, foram utilizadas as classes do Youubi: *InstitutionTO*, que representa uma instituição de ensino; e *AcountInstitutionTO*, que possui informações sobre as plataformas disponibilizadas pela instituição e como seus serviços podem ser consumidos. A manipulação desses dados pode ser feita no painel de administração do cliente Edubi-Web, mas somente por um usuário cadastrado na Youubi API com papel de administrador.

Uma adversidade encontrada foi o mapeamento entre os diferentes modelos de dados. Isto ocorre porque cada plataforma possui características específicas e diferentes maneiras de abordar um conceito. Por exemplo, a entidade *Person* (pessoa) no Edubi mapeia de forma diferente da entidade *User* do Moodle os atributos: data de nascimento e *username*. A solução proposta para contornar este problema foi criar um controlador

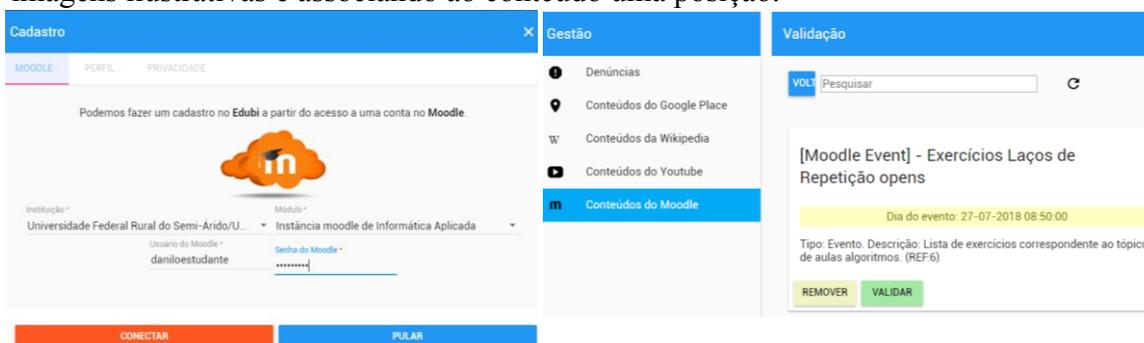
genérico para automatizar a construção das requisições, as quais independem do sistema integrado. O controlador recebe informações relacionadas à comunicação com o provedor do serviço, como a URL do serviço e *token* de acesso. Outra característica do controlador é a possibilidade de criar composições de serviços. Desse modo, é possível a montagem de novos serviços a partir da composição de outros serviços já existentes. O controlador é ligado a clientes que implementam os modelos de dados de cada plataforma, e os mapeiam em entidades elementares do Youubi.

### 5.1. Integração com o Moodle

Inicialmente, foram avaliadas as formas possíveis de integração com os serviços do *Moodle*, pois este possui suporte tanto a serviços baseados na pilha WS-\*, quanto a serviços REST. Enfim, optou-se por utilizar a API REST por sua simplicidade.

A etapa seguinte foi configurar o Moodle para permitir o acesso do Youubi e selecionar os serviços que podem ser consumidos. Com os serviços criados, é gerado aleatoriamente um *token* de acesso pelo Moodle. Em posse do *token*, já é possível realizar requisições e consumir os serviços do Moodle. É importante ressaltar que cada plataforma possui suas particularidades e, portanto, configurações específicas para permitir o acesso de ambientes externos a seus serviços. Uma necessidade encontrada na qual o Moodle não provê nenhum tipo de serviço, é o de autenticação para ambientes externos. Por conta disto, foi criado um *plug-in*, codificado na linguagem PHP, que implementa um provedor de autenticação baseado no padrão *OAuth* (*Open Authorization Protocol*). Em seguida, o *plug-in* foi adicionado ao Moodle e disponibilizado como um *web service*.

Por fim, após o esforço de integração, o Youubi pôde consumir do componente LMS os seguintes serviços: autenticação de usuário Moodle; obtenção do perfil de usuário Moodle; e lista de conteúdos e eventos criados nos cursos Moodle que o usuário faz parte. No contexto do ambiente Edubi, esses serviços são consumidos nos seguintes cenários (Figura 3): para cadastrar um novo usuário Edubi é preciso antes ser autenticado como usuários Moodle; professores e tutores podem importar seus conteúdos e eventos do Moodle para o Edubi por meio do painel de administração do Edubi-Web. Vale salientar que durante essa importação, o professor ou tutor podem enriquecer seus conteúdos Moodle com mais detalhes, por exemplo, adicionando imagens ilustrativas e associando ao conteúdo uma posição.



a) Autenticação Moodle para cadastrar usuário no Edubi      b) Importação e melhoria de conteúdos Moodle para o Edubi

**Figura 3. Exemplos de cenários de uso do Componente LMS**

## 6. Resultados: Avaliação do Componente LMS

O componente de integração foi avaliado com uma amostra de seis participantes (4 programadores, 1 testador, 1 gerente e 1 designer) que o utilizaram (direta ou indiretamente) durante o desenvolvimento dos artefatos de *software* do Edubi, no departamento de educação à distância de uma universidade federal no Brasil.

A técnica utilizada foi o Modelo de Aceitação da Tecnologia (*Technology Acceptance Model* - TAM). De acordo com Davis (1989), ela tem como objetivo esclarecer o comportamento dos usuários a partir da utilidade e da facilidade de utilização percebida por eles. A utilidade percebida é o quanto um indivíduo acha que a tecnologia melhorará seu desempenho no trabalho. A facilidade de utilização percebida consiste em avaliar se utilizar essa tecnologia vai exigir pouco esforço. A junção desses dois construtos resulta nas intenções de uso.

Os dados foram coletados através de um questionário on-line, utilizando a ferramenta Google Forms. O questionário possui 8 questões do tipo escala *Likert* com opções que variam de 1 a 5, assumindo que os valores 1 e 2 representam que o entrevistado discorda da afirmação, o valor 3 representa indecisão e os valores 4 e 5 assumem que o entrevistado concorda com a afirmação. Além das duas questões de múltipla escolha, foram definidas 3 questões subjetivas para coletar opiniões sobre o componente. A Tabela 1 apresenta as questões objetivas e suas respectivas variáveis.

**Tabela 1 – Variáveis, Conceitos e Dimensões**

Dimensões e Conceitos	Variáveis
<b>Facilidade de Uso Percebida</b>	FUP1. Aprendi rapidamente a utilizar as funcionalidades relacionadas ao <i>Moodle</i> .
	FUP2. A utilização das funcionalidades relacionadas ao <i>Moodle</i> é de fácil compreensão.
	FUP3. A utilização das funcionalidades relacionadas ao <i>Moodle</i> não requer muito esforço.
	FUP4. No geral, considero fácil a utilização do conteúdo <i>Moodle</i> gerado pela lista de recomendações.
<b>Utilidade Percebida</b>	UP1. Usar o componente de integração com o <i>Moodle</i> foi útil para o meu trabalho.
	UP2. Usar o componente de integração melhorou a forma de coletar conteúdos vindos do <i>Moodle</i> .
	UP3. Usar o componente de integração com o <i>Moodle</i> contribuiu para geração de conteúdos no <i>Youubi</i> .
	UP4. No geral, utilizar o componente de integração com o <i>Moodle</i> é útil para o <i>Youubi</i> .
<b>Intenção de Uso</b>	IU1. Você utilizaria o componente em outras ocasiões?
<b>Variável Externa</b>	VE1. Você já teve alguma experiência com integração de software?

A análise dos dados coletados teve como objetivo testar empiricamente as hipóteses no TAM para medir os seus construtos. Pretendeu-se chegar ao objetivo de medir a influência de cada construto, testando as seguintes hipóteses:

- Hipótese 1: a experiência do usuário influencia na facilidade de uso percebida;
- Hipótese 2: a facilidade de uso percebida influencia na utilidade percebida;

- Hipótese 3: a utilidade percebida influencia a intenção de uso do componente;
- Hipótese 4: a facilidade de uso percebida influencia a intenção de uso.

Os cálculos estatísticos foram realizados utilizando o software Microsoft Excel. Para avaliar as hipóteses, foi realizado o cálculo do Coeficiente de Correlação Linear de *Pearson* (R). Esse método é usualmente utilizado para medir a correlação entre duas variáveis. Seu valor varia de -1 a 1, onde -1 indica uma perfeita correlação inversa entre as variáveis, e 1 indica uma perfeita correlação positiva entre as variáveis. O valor 0 indica que não há correlação linear entre as variáveis. No entanto, os valores podem variar entre -1 e 1, como apresentado a seguir: de 0,00 a 0,29 existe correlação linear fraca entre as variáveis; entre 0,30 e 0,59 existe correlação linear moderada entre as variáveis; de 0,60 a 0,89 existe correlação linear forte entre as variáveis; e para valores de 0,90 a 1,00 existe correlação linear muito forte entre as variáveis. Já o Coeficiente de Variação (CV) explica o quanto a variação de uma variável influencia na variação de outra. Para obter o CV, basta calcular R elevado ao quadrado. Ainda é possível ser expresso na forma de percentagem se multiplicado por 100.

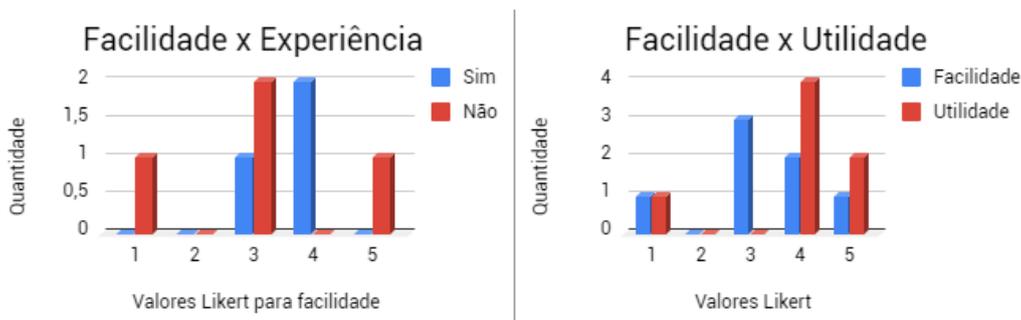
### 6.1. Análise dos dados

Para realizar a análise dos resultados foi construída uma tabela com base nas respostas obtidas pelo questionário descrito na Tabela 2.

**Tabela 2 - Conjunto dos dados obtidos do questionário**

Indivíduo	Facilidade	Utilidade	Intenção	Experiência
Indivíduo #1	3	4	Sim	Sim
Indivíduo #2	3	5	Sim	Não
Indivíduo #3	1	1	Sim	Não
Indivíduo #4	5	5	Sim	Não
Indivíduo #5	4	4	Sim	Sim
Indivíduo #6	3	4	Sim	Não
Indivíduo #7	4	4	Sim	Sim

A coluna Facilidade contém a média das respostas das questões FUP1, FUP2, FUP3 e FUP4. Já a coluna Utilidade, contém os resultados das médias das questões UP1, UP2, UP3 e UP4. A coluna Intenção contém as respostas relacionadas à questão IU1. E, por fim, a coluna Experiência contém as respostas relacionadas a questão VE1. A seguir, serão exibidos os resultados e as análises dos testes de hipóteses baseados nos dados da Tabela 2.



**Figura 4. Gráficos dos resultados obtidos na avaliação.**

**Hipótese 1: a experiência do usuário influencia na facilidade de uso percebida:**

Conforme o Gráfico (a) da Figura 4, os entrevistados que não tiveram experiência com integração de software divergiram de forma significativa em suas respostas. Desse modo, não foi possível identificar relação entre experiência com integração de software e a facilidade de uso.

**Hipótese 2: a facilidade de uso percebida influencia na utilidade percebida:**

Conforme o Gráfico (b) da Figura 4, observa-se que a maioria dos entrevistados consideraram que o componente é útil. Com relação a facilidade, a maioria das respostas se concentraram em fácil e moderado. Porém, para avaliar melhor a relação entre essas variáveis, foi calculado o Coeficiente de Correlação Linear de *Pearson*, cujo valor resultou em aproximadamente 0,8189. Esse valor indica uma correção forte entre as variáveis e implica que a facilidade de uso influencia em cerca de 67% na utilidade.

**Hipótese 3: a utilidade percebida influencia a intenção de uso do componente:** Por meio do Gráfico (b) da Figura 4, observa-se que a maioria dos entrevistados considerou o componente útil. Além disso, todos demonstraram interesse em utilizar o componente em outras ocasiões. Percebe-se então que há uma relação positiva entre a utilidade percebida pelos entrevistados e sua intenção de uso.

**Hipótese 4: a facilidade de uso percebida influencia a intenção de uso:** Conforme o Gráfico (b) da Figura 4, percebe-se uma divergência nas respostas com relação percepção dos usuários na facilidade de uso do componente. Sendo assim, com todos os desenvolvedores demonstrando interesse em utilizar o componente em outras ocasiões, não é possível afirmar que há relação entre a facilidade de uso e a intenção de uso por parte dos desenvolvedores. Portanto, mesmo com dificuldades, os desenvolvedores têm a intenção de utilizar a ferramenta novamente.

## 7. Considerações Finais

Neste trabalho, foi apresentado e avaliado o componente de integração do Moodle utilizado no ambiente Edubi. O componente é capaz de consumir *web services* oferecidos pelo Moodle e retornar resultados compatíveis com a Youubi API.

Os resultados da avaliação apontam que os entrevistados divergiram sobre o componente ser de fácil utilização ou não. Com relação à utilidade, a maioria afirmou que a ferramenta é útil para integrar o Moodle com o Youubi. Além disso, todos os entrevistados demonstraram interesse em utilizar o componente em outras ocasiões. Apenas um dos entrevistados respondeu negativamente a avaliação. No geral, as opiniões sobre o componente desenvolvido foram positivas. Um dos entrevistados afirmou que “o componente serve como um facilitador para um desenvolvedor de aplicações que deseja integrar conteúdos do Moodle em seu sistema”. Outro afirma que “o componente consegue auxiliar na percepção do usuário sobre os conteúdos presentes no Moodle, porém não tem a pretensão de prover todo o conjunto de conteúdos do Moodle com todos os seus detalhes”. Com relação às dificuldades encontradas pelos desenvolvedores, apenas um declarou dificuldade em configurar o Moodle para habilitar o acesso externo. Os demais não mencionaram dificuldades na manipulação do componente. Um deles afirmou ainda que “a utilização dos serviços do componente ficou bem clara por parte da API”. Já as sugestões de melhorias foram direcionadas ao fornecimento de documentação mais detalhada e mais informações no conteúdo fornecido pelo Moodle. Com base nesses resultados, principalmente nos aspectos de utilidade percebida e intenção de uso, observaram-se indícios de que a utilização do Componente LMS contribuiu para facilitar a integração entre a Youubi API e o Moodle.

Com relação aos trabalhos futuros, pode-se incluir a melhoria do mapeamento entre os modelos de dados e a inclusão de novos sistemas SGA.

## Referências

- Bezerra, S. D. F. et al. (2017). Sistema de Recomendação Ubíquo Integrando Hipermídias Baseadas em Ontologias. In: XXVIII Brazilian Symposium on Computers in Education (XXVIII Simpósio Brasileiro de Informática na Educação-SBIE). p. 1793.
- Callegari-Jacques, S. M. Bioestatística: princípios e aplicações. Porto Alegre: Artemed, 2003. 255p.
- Cobo, C. and Moravec, J. W. Aprendizaje invisible. Hacia una nueva ecología de la educación. Universitat de Barcelona; Universidad Internacional de Andalucía, Barcelona, Sevilla, 2011.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, Minneapolis (MN), v.13, n.3, p.319-339.
- Edubi. (2018). Disponível: <<http://edubi.ufersa.edu.br/>> Acesso: 13 de junho de 2018.
- Furtado, U. de M.; De Lima, R. W.; De Farias, A. M. G. Modelo de integração adaptável entre ambientes virtuais de aprendizagem e sistemas de gestão acadêmica baseado em arquitetura orientada a serviços. *Revista de Educação a Distância*, v.2, n.2. 2015.
- Gomes, A. S. et al. (2018). Amadeus: Novo modelo de sistema de gestão de aprendizagem. *Revista Brasileira de Aprendizagem Aberta e a Distância*, v. 8, 2018.
- Monteiro, B. S. et al. (2017). “Youubi: Ambiente de Aprendizagem Ubíqua.” *Revista Brasileira de Informática na Educação*, v. 25, p. 94, 2017.
- Oliveira, P. R. D. (2015). Uma avaliação de metodologias para validação de software. 68 p. Monografia (Graduação em Ciência da Computação) - Universidade Federal Rural do Semi-Árido, Mossoró, 2015.
- Ouya, S.; Mbacke, A. B.; Mendy, G.; Diouf, P. W.; Sy, K. Social Network Integration To E-Learning Environment. 2015 Ieee/Acs 12th International Conference of Computer Systems and Applications (Aiccsa). 2015.
- Pautasso, C., Zimmermann, O. and Leymann, F. (2008). "RESTful Web Services versus 'Big' Web Services: Making the Right Architectural Decision", Proc. 17th Int'l World Wide Web Conf. (WWW '08).
- Raspopoviü, M.; Cvetanoviü, S.; Stanojeviü, D.; Opaüiü, M. Software Architecture for Integration of Institutional and Social Learning Environmentss, *Sci. Comput. Program*. 2016.
- Teknillinen, A. (2008). Semantic Web Services Survey. Technical Report. Helsinki University of Technology, Laboratory of Software Technology.
- Vinoski, S. (2002). Putting the "Web" into Web services: Interaction models, part 1: Current practice. *IEEE Internet Computing*, 6(3):89–91.
- W3C. (2004). Architecture of the World Wide Web. Disponível em: <<https://www.w3.org/TR/webarch/>>. Acesso em: junho de 2018.