

Carlos Vinícius Monteiro Costa

LTlaaS: Transformando aplicações em ferramentas de aprendizagem integráveis

São Luís - MA

2021

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Monteiro Costa, Carlos Vinícius.

LTIIaaS: Transformando aplicações em ferramentas de
aprendizagem integráveis / Carlos Vinícius Monteiro Costa.

- 2021.

61 f.

Orientador(a): Carlos de Salles Soares Neto.

Curso de Ciência da Computação, Universidade Federal do
Maranhão, Online, 2021.

1. Educação a distância. 2. Integração. 3.
Interoperabilidade. 4. LMS. 5. LTI. I. Soares Neto,
Carlos de Salles. II. Título.

Carlos Vinícius Monteiro Costa

LTlaaS: Transformando aplicações em ferramentas de aprendizagem integráveis

Trabalho de conclusão de curso apresentado ao curso de Ciência da Computação da Universidade Federal do Maranhão como parte dos requisitos necessários para obtenção do grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Carlos de Salles Soares Neto

São Luís - MA
2021

Carlos Vinícius Monteiro Costa

LTlaaS: Transformando aplicações em ferramentas de aprendizagem integráveis

Trabalho de conclusão de curso apresentado ao curso de Ciência da Computação da Universidade Federal do Maranhão como parte dos requisitos necessários para obtenção do grau de bacharel em Ciência da Computação.

Trabalho aprovado em 30 de abril, São Luís - MA, 2021:

Prof. Dr. Carlos de Salles Soares Neto
Orientador
Universidade Federal do Maranhão

Prof. Dr. Carlos Eduardo Portela Serra de Castro
Examinador
Universidade Federal do Maranhão

Prof. Dr. Francisco Glaubos Nunes Clímaco
Examinador
Universidade Federal do Maranhão

São Luís - MA
2021

À minha família e meus amigos.

Agradecimentos

A minha mãe e toda minha família por sempre terem me incentivado e me apoiado em tudo.

A Gaby, por ser a minha namorada, companheira e por me dar forças e alegria para continuar.

Ao meu orientador, Salles, pela sua dedicação mesmo que em tão pouco tempo.

A Dilson, por ser um mentor brilhante sempre que eu precisei.

Aos professores Anselmo, Geraldo, Simara, Greiciane, e todos os outros professores que fizeram parte dessa jornada acadêmica.

A Pedro, por ser um amigo incrível e por ter me ajudado imensamente nas últimas etapas da graduação.

A Wesley, Welton e Vinícius, pelos inúmeros momentos compartilhados de diversão.

A todos os meus amigos do CodeBuilders por me acompanharem nessa escalada.

A Rodrigo e Danilo, pela sua amizade e companheirismo de muitos anos.

A Júnior e Vinícius, por estarem ao meu lado durante os melhores e piores momentos.

A Camila por ter sempre apoiado e acreditado em mim.

A Felipe e Talita, por serem amigos inesquecíveis.

A você que tirou um pouco do seu tempo para prestigiar este trabalho.

Mostly void, partially stars

Resumo

A educação a distância é uma modalidade de ensino que tem como objetivo promover um processo de aprendizagem completo através de recursos tecnológicos, sendo aplicada, normalmente, através de um sistema de gestão de aprendizagem (LMS). No ano de 2020, devido a pandemia de coronavírus SARS-Cov2, houve um forte avanço na adesão à educação a distância, a medida que o ensino presencial se tornou inviável. O aumento na demanda e a necessidade de atender à uma maior variedade de casos de uso trouxe à tona as limitações das funcionalidades oferecidas nativamente pelos LMSs. Para preencher estas lacunas e possibilitar a utilização de uma maior diversidade de recursos educacionais, o protocolo de interoperabilidade de ferramentas de aprendizagem (LTI) pode ser utilizado para integrar recursos externos no ambiente de ensino. Porém, devido a sua complexidade, o protocolo LTI ainda demanda muito tempo e muitos recursos para ser implementado, o que dificulta sua adoção pelo público geral. Dado este contexto, o presente trabalho tem por objetivo apresentar duas contribuições centrais: a biblioteca LtijS e o sistema LTIaaS, que foi desenvolvido como uma solução capaz de abstrair a complexidade do protocolo LTI, facilitando a criação de recursos de aprendizagem integráveis, de uma maneira desacoplada e agnóstica de linguagem.

Palavras-chaves: Educação a distância, interoperabilidade, integração, LTI, LMS.

Abstract

Distance education is a teaching modality that aims to promote a complete learning process through technological resources, being applied, usually, through a learning management system (LMS). In the year 2020, due to a SARS-Cov2 coronavirus pandemic, there was a strong increase in adhering to distance education, as face-to-face education became unviable. The increase in demand and the need to attend to a greater variety of use cases brought to light the limitations of the features offered natively by LMSs. To fill these gaps and enable the use of a greater diversity of educational resources, the learning tools interoperability protocol (LTI) can be used to integrate external resources in the teaching environment. However, due to its complexity, the LTI protocol still requires a lot of time and resources to be fully implemented, which makes it difficult for the general public to adopt it. Given this context, the present work aims to present two central contributions: the Ltijis library and the LTIaaS system, which was developed as a solution capable of abstracting the complexity of the LTI protocol, facilitating the creation of integrable learning resources, in an uncoupled and language agnostic way.

Keywords: Distance education, interoperability, integration, LTI, LMS.

Lista de ilustrações

Figura 1 – Estrutura do protocolo LTI 1.3 e LTI <i>Advantage</i> . Fonte: (IMS Global Learning Consortium Inc.,)	17
Figura 2 – Método de comunicação baseado em mensagens. Fonte: Autor do trabalho	18
Figura 3 – Método de comunicação baseado em serviços. Fonte: Autor do trabalho	19
Figura 4 – Fluxo de funcionamento da biblioteca Ltjjs. Fonte: Autor do trabalho	25
Figura 5 – Representação da arquitetura LTIaaS. Fonte: Autor do trabalho . .	29
Figura 6 – Fluxo de funcionamento do sistema LTIaaS. Fonte: Autor do trabalho	31
Figura 7 – Powtoon acessado pelo <i>placement Global navigation</i> . Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.	39
Figura 8 – Botão Powtoon acessível através da caixa de edição de texto. Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.	39
Figura 9 – Tela de seleção de powtoons. Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.	40
Figura 10 – Powtoon sendo visualizado no <i>placement Assignment selection</i> . Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor. . .	40
Figura 11 – Tela de seleção de powtoons no <i>placement Homework submission</i> . Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor. . .	41
Figura 12 – Powtoon sendo avaliado em <i>Speed Grader</i> . Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.	41
Figura 13 – Controle de acesso pela atividade Biometria facial. Fonte: (Mobilidade Segura, 2021). Captura de tela criada pelo autor.	43
Figura 14 – Tela de biometria facial. Fonte: (Mobilidade Segura, 2021). Captura de tela criada pelo autor.	44
Figura 15 – Arquitetura do tipo de caso de uso Repositório de aplicações. Fonte: Autor do trabalho	44
Figura 16 – Tela de seleção de recursos na plataforma SAITE AVA. Fonte: (UNA-SUS/UFMA, 2021). Captura de tela criada pelo autor.	45
Figura 17 – Recurso externo acessado dentro da plataforma SAITE AVA. Fonte: (UNA-SUS/UFMA, 2021). Captura de tela criada pelo autor.	46
Figura 18 – Arquitetura do tipo de caso de uso Moodle como Ferramenta LTI 1.3. Fonte: Autor do trabalho	48
Figura 19 – Recurso publicado como Ferramenta LTI 1.3. Fonte: (BLENDING EDUCATION, LLC. U.S.A, 2021). Captura de tela criada pelo autor.	48
Figura 20 – Tela de seleção de recursos publicados como Ferramenta LTI 1.3. Fonte: (BLENDING EDUCATION, LLC. U.S.A, 2021). Captura de tela criada pelo autor.	49
Figura 21 – Recurso acessado como Ferramenta LTI 1.3. Fonte: (BLENDING EDUCATION, LLC. U.S.A, 2021). Captura de tela criada pelo autor.	49

Lista de listagens

1	Exemplo de corpo gerador de JWT	16
2	Exemplo de JWT assinado	16
3	Exemplo de <i>ID Token</i> gerado no processo de lançamento	20
4	Exemplo de Content Item do tipo <i>litiResourceLink</i>	22
5	Exemplo de um objeto gerador de <i>Itik</i>	24
6	Exemplos de métodos de acesso a serviços	26
7	Exemplo de método de registro de plataformas	26
8	Exemplo de criação de rota de lançamento	27
9	Exemplo de uso do objeto <i>res.locals.token</i>	27
10	Exemplo de arquivo de configuração <i>dev.config.json</i>	30
11	Exemplo de cabeçalho de autorização <i>LTIK-AUTH-V1</i>	31
12	Exemplo de geração de cabeçalho de autorização <i>LTIK-AUTH-V1</i>	32
13	Exemplo de cabeçalho de autorização <i>Bearer Token</i>	32
14	Exemplo de geração de cabeçalho de autorização <i>Bearer Token</i>	32
15	Exemplo de código de acesso ao <i>endpoint ID Token</i>	33
16	Exemplo de estrutura do objeto <i>ID Token</i>	33
17	Exemplo de campo <i>user</i> do objeto <i>ID Token</i>	34
18	Exemplo de campo <i>platform</i> do objeto <i>ID Token</i>	34
19	Exemplo de campo <i>launch</i> do objeto <i>ID Token</i>	35
20	Exemplo de URL gerada pelo processo de seleção de <i>powtoons</i>	42
21	Exemplo de objeto gerador de evento	47
22	Exemplo de objeto gerador de evento com informações de contexto	47
23	Exemplo de código de acesso ao <i>endpoint Memberships</i>	56
24	Exemplo de estrutura do objeto retornado pelo <i>endpoint Memberships</i>	56
25	Exemplo de código de recuperação de linhas de nota	57
26	Exemplo de estrutura do objeto retornado pelo <i>endpoint Line Items</i>	57
27	Exemplo de código de criação de linha de nota	58
28	Exemplo de código de atualização de linha de nota	58
29	Exemplo de código de envio de nota	59
30	Exemplo de código de recuperação de notas	59
31	Exemplo de objeto retornado pela sub-rota <i>scores</i> do <i>endpoint Line Items</i>	60
32	Exemplo de código de geração de formulário <i>endpoint Deep Linking</i>	60
33	Exemplo de objeto retornado pelo <i>endpoint Deep Linking</i>	61
34	Exemplo de código de criação de registro de Plataforma	61
35	Exemplo de código de recuperação de registros de Plataforma	61
36	Exemplo de estrutura do objeto retornado pelo <i>endpoint Platforms</i>	62
37	Exemplo de código de atualização de um registro de Plataforma	62

Lista de abreviaturas e siglas

API	Application Programming Interface
CPF	Cadastro de Pessoas Físicas
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JWS	JSON Web Signature
JWT	JSON Web Token
LMS	Learning Management System
LTI	Learning Tools Interoperability
LTIaaS	LTI as a Service
SaaS	Software as a Service
SaaS	Software as a Service
SSO	Single Sign-On
SUS	Sistema Único de Saúde
UFMA	Universidade Federal do Maranhão
UNASUS	Universidade Aberta do SUS
URL	Uniform Resource Locator

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	14
1.1.1	Objetivo geral	14
1.1.2	Objetivos específicos	15
1.2	Organização do trabalho	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	Sistemas de gestão da aprendizagem	16
2.2	JSON Web Tokens	16
2.3	Protocolo LTI	17
2.3.1	LTI 1.3 Core	19
2.3.2	Deep Linking	21
2.3.3	Names and Roles Provisioning	22
2.3.4	Assignment and Grades	22
2.3.5	Extensões	23
3	BIBLIOTECA LTIJS	24
3.1	Funcionamento	24
3.2	Usabilidade	25
3.3	Extensões	27
4	LTI AS A SERVICE	29
4.1	Configuração	29
4.2	Funcionamento	30
4.3	Autenticação	31
4.3.1	Esquema de autenticação <i>LTIK-AUTH-V1</i>	31
4.3.2	Esquema de autenticação <i>Bearer Token</i>	32
4.4	Endpoints de serviço	32
4.4.1	ID Token	33
4.4.2	Memberships	35
4.4.3	Line Items	36
4.4.4	Deep Linking	37
4.4.5	Platforms	37
5	CASOS DE USO	38
5.1	Aplicação única	38
5.1.1	Powtoon	38
5.1.2	Mobilidade Segura	43
5.2	Repositório de aplicações	44
5.2.1	UNA-SUS/UFMA	45
5.3	Moodle como Ferramenta LTI 1.3	47
5.3.1	Blending Education	48

6	CONCLUSÃO	51
6.1	Contribuições	51
6.2	Trabalhos futuros	52
	REFERÊNCIAS	53
	APÊNDICES	55
	APÊNDICE A – LISTAGENS REFERENTES AOS ENDPOINTS CITADOS NA SEÇÃO 4.4	56

1 Introdução

A educação a distância é uma modalidade de ensino que tem como objetivo promover um processo de aprendizagem completo através de recursos tecnológicos, enquanto professores e alunos estão separados, física ou temporalmente. Sendo aplicada, normalmente, através de um sistema de gestão de aprendizagem (LMS, do inglês *Learning Management System*).

No ano de 2020, devido à pandemia de coronavírus SARS-Cov2, houve um forte avanço na adesão à educação a distância, a medida que o ensino presencial se tornou inviável do ponto-de-vista sanitário (UNESCO, 2020). Neste cenário de adoção forçada, os problemas da abordagem remota de ensino se tornaram ainda mais evidentes. O aumento na demanda e a necessidade de atender à uma maior variedade de casos de uso trouxe à tona as limitações das funcionalidades oferecidas nativamente pelos LMSs, que não permitem o total aproveitamento das tecnologias disponíveis no meio.

Para preencher as lacunas nas funcionalidades dos LMSs, o protocolo de interoperabilidade de ferramentas de aprendizagem (LTI, do inglês *Learning Tools Interoperability*) (VERVOORT; MILLS, 2019) pode ser usado para integrar recursos externos no ambiente de ensino. Isso possibilita o aumento na diversidade dos recursos educacionais utilizados, e permite que estes recursos possam se comunicar com o LMS, criando uma experiência mais unificada para alunos e professores, onde, se aplicado corretamente, a barreira entre plataforma e recurso externo é imperceptível.

Devido à sua complexidade, embora existam recursos que visem facilitar a sua utilização, o protocolo LTI ainda demanda muito tempo e muitos recursos para ser implementado completamente, o que dificulta sua adoção pelo público geral. A utilização destes recursos também limita as tecnologias que podem ser utilizadas na criação de integrações. Esta barreira técnica limita o impacto que a tecnologia poderia ter no meio educacional.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo do trabalho consiste em apresentar e descrever o sistema *LTI as a Service* (LTIaaS), capaz de abstrair a complexidade do protocolo LTI enquanto atuando de maneira desacoplada da aplicação, facilitando sua utilização tanto na criação de ferramentas LTI do zero, quanto na adaptação de aplicações já consolidadas, e permitindo que a especificação tenha um alcance e impacto maior dentro do meio educacional.

1.1.2 Objetivos específicos

- Apresentar aspectos relevantes da biblioteca LtijS, contribuição deste trabalho que serviu de base para o desenvolvimento do sistema LTiaaS.
- Apresentar o sistema LTiaaS, suas características, seu processo de configuração e funcionamento.
- Descrever casos de uso reais que exemplificam a utilização do sistema LTiaaS na criação de integrações LTI.

1.2 Organização do trabalho

Além do Capítulo 1, este trabalho está organizado em mais 5 capítulos, sendo o Capítulo 2 a fundamentação teórica do trabalho, descrevendo o protocolo LTI e outras tecnologias relevantes.

O Capítulo 3 apresenta a biblioteca LtijS, descrevendo suas características e funcionalidades relevantes para o sistema LTiaaS.

O Capítulo 4 apresenta o sistema LTiaaS, descrevendo seus processos de configuração e usabilidade, e suas funcionalidades. O Capítulo 5 descreve diversos casos de uso reais onde o sistema LTiaaS foi aplicado na criação de integrações LTI, e por fim, o Capítulo 6 contém as considerações finais sobre este trabalho e uma breve apresentação de trabalhos futuros.

2 Fundamentação teórica

Neste capítulo são apresentadas as principais tecnologias e conceitos utilizados no desenvolvimento deste trabalho.

2.1 Sistemas de gestão da aprendizagem

Sistemas de gestão da aprendizagem (LMS, do inglês *Learning Management System*), ou plataformas de ensino, são aplicações que oferecem funcionalidades focadas no gerenciamento de atividades educacionais de forma colaborativa. As funcionalidades oferecidas podem abranger aspectos de administração e controle de usuários e cursos, manipulação, criação e armazenamento de recursos educacionais, e auxílio na comunicação entre alunos e professores.

2.2 JSON Web Tokens

JSON Web Token (JWT) (JONES; BRADLEY; SAKIMURA, 2015b) define uma forma compacta de representar informações passadas entre duas partes através de um objeto JSON, contendo um cabeçalho. Um JWT pode ser assinado através de vários algoritmos criptográficos (JONES, 2015), gerando um *JSON Web Signature* (JWS) (JONES; BRADLEY; SAKIMURA, 2015a), que pode ser utilizado para assegurar a integridade e origem das informações passadas. As listagens 1 e 2 apresentam, respectivamente, um exemplo do corpo gerador de um JWT, e o JWS gerado pela sua assinatura.

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

Listagem 1 – Exemplo de corpo gerador de JWT

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwiaXNjaWkiOiJkbmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MzkwMjQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

Listagem 2 – Exemplo de JWT assinado

2.3 Protocolo LTI

O protocolo de Interoperabilidade de ferramentas de aprendizagem (LTI) foi criado pela IMS Global Learning Consortium com o intuito de padronizar a comunicação entre plataformas de ensino e ferramentas de aprendizagem, e permitir que as duas partes troquem informações através de fluxos descritos como serviços. Esta seção apresentará alguns conceitos e características do protocolo LTI. Aspectos mais técnicos da especificação, como modelos de autenticação, não serão detalhados, pois não são relevantes no escopo do trabalho proposto.

O protocolo LTI passou por diversas atualizações desde sua concepção, a versão relevante para o escopo do trabalho proposto é a 1.3 (VERVOORT; MILLS, 2019), juntamente com o conjunto de serviços que a acompanham denominado de *LTI Advantage*. A atualização 1.3 modificou os modelos de autenticação utilizados e adicionou novas funcionalidades, ambos serão descritos nesta seção. A Figura 1 ilustra a estrutura do protocolo LTI 1.3 e *LTI Advantage*.

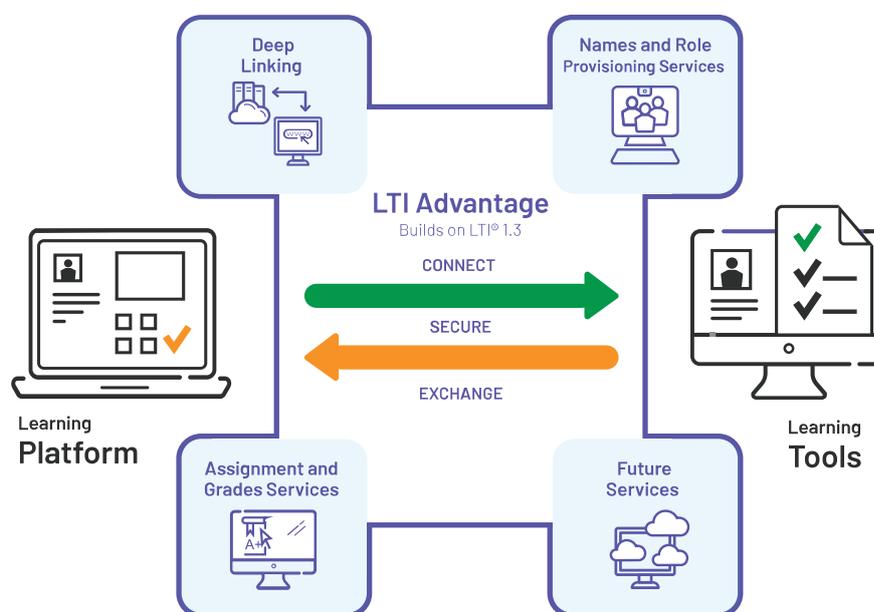


Figura 1 – Estrutura do protocolo LTI 1.3 e *LTI Advantage*. Fonte: (IMS Global Learning Consortium Inc.,)

São definidos dois tipos de sistemas, Plataformas e Ferramentas. Plataformas são quaisquer sistemas através dos quais um usuário final pode obter acesso a uma ferramenta de ensino lançada remotamente, comumente esse papel é realizado por sistemas de gestão de aprendizagem (LMSs). Ferramentas são quaisquer sistemas que expõem interfaces para um ou mais recursos externos que podem ser acessados através de uma Plataforma.

Para que Plataformas e Ferramentas possam se comunicar, é preciso que antes ocorra um processo de registro, onde ambas as partes trocam identificadores que serão utilizados pelos diversos fluxos de autenticação descritos a seguir. O protocolo LTI descreve dois métodos de comunicação entre Plataformas e Ferramentas. Baseada em mensagens, utilizado quando a comunicação é intermediada por um navegador, e baseada em serviços, utilizado quando a comunicação entre as duas

partes é direta. É descrito um protocolo de segurança (SMYTHE et al., 2019b) englobando os dois métodos de comunicação implementados.

Para mensagens originadas na Plataforma é implementado um modelo de autenticação baseado no protocolo *OpenID Connect Core* (SAKIMURA et al., 2014). O resultado desse fluxo de autenticação é um objeto *ID Token*, que será descrito com mais detalhes a seguir. Mensagens originadas na Ferramenta devem ter o seu conteúdo enviado dentro de um JWT assinado, como descrito na Seção 2.2. A Figura 2 ilustra o método de comunicação baseado em mensagens.

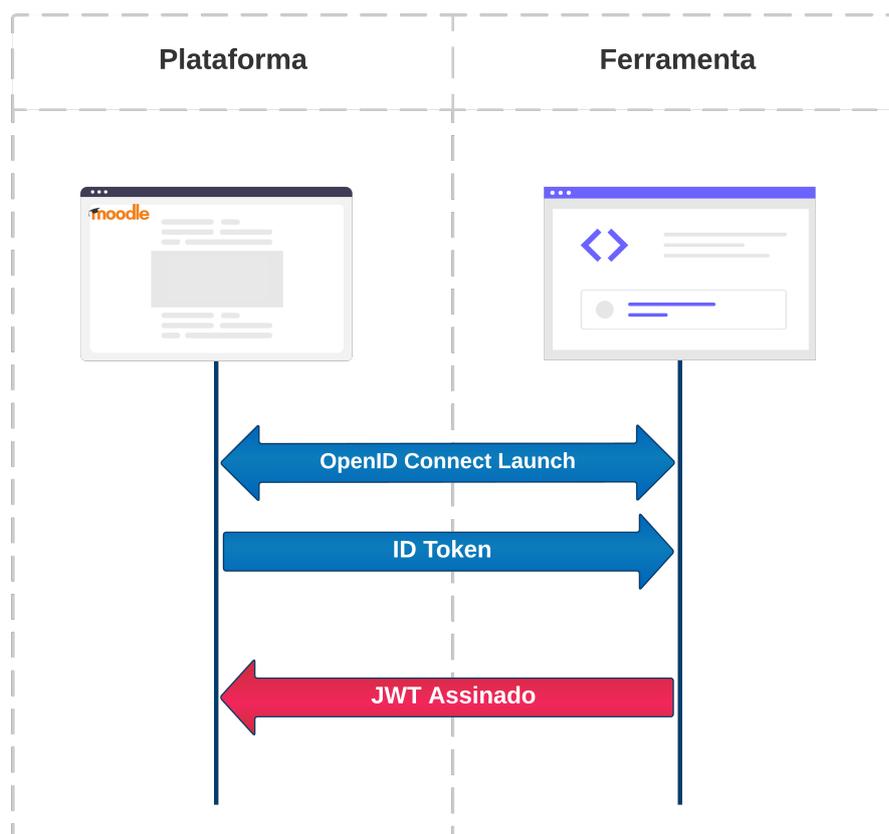


Figura 2 – Método de comunicação baseado em mensagens. Fonte: Autor do trabalho

Para a comunicação baseada em serviços é implementado um modelo de autenticação baseado no fluxo *Client-Credentials* do protocolo *OAuth 2.0* (HARDT, 2012). O resultado desse fluxo de autenticação é um *token* de acesso que pode ser utilizado pela Ferramenta para acessar os serviços oferecidos pela Plataforma. A Figura 3 ilustra o método de comunicação baseado em serviços.

As possíveis interações entre Plataforma e Ferramenta são denominadas de serviços, que, apesar do nome, podem ser baseados em qualquer um dos métodos de comunicação descritos anteriormente. O protocolo LTI 1.3 implementa um serviço base chamado de *Core* e um conjunto de serviços adicionais chamado de *LTI Advantage*.

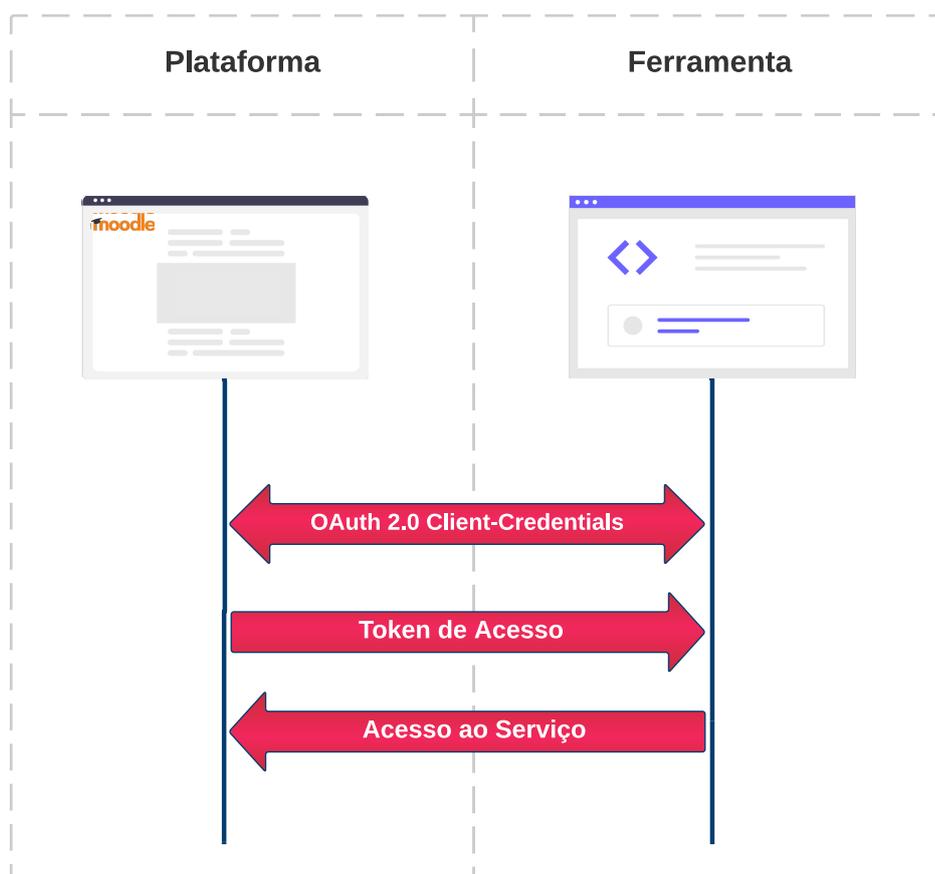


Figura 3 – Método de comunicação baseado em serviços. Fonte: Autor do trabalho

2.3.1 LTI 1.3 Core

O serviço LTI 1.3 *Core* (VERVOORT; MILLS, 2019) descreve a interação principal entre Plataformas e Ferramentas, através da qual uma Ferramenta pode ser acessada a partir do contexto de uma Plataforma, comumente dentro de um *Iframe*¹. Este processo é chamado de lançamento LTI.

Lançamentos LTI podem ocorrer a partir de diversas localizações dentro de uma Plataforma, chamadas comumente de *placements*, cada Plataforma é livre para decidir os *placements* que deseja suportar e como os deseja nomear, normalmente dando para cada um deles uma utilidade única. O *placement* mais comum é o que utiliza Ferramentas como recursos educacionais dentro de um curso.

São definidos dois tipos de lançamento LTI com funções distintas, *Core* e *Deep Linking*, ambos seguindo o método de comunicação baseado em mensagens originadas na Plataforma.

Ao fim de um processo de lançamento LTI, a Ferramenta terá acesso a um objeto *ID Token*, gerado pelo fluxo de autenticação *OpenID Connect Launch*, contendo informações sobre o contexto geral do lançamento e sobre o acesso de cada um dos serviços LTI disponibilizados por aquela Plataforma. A Listagem 3 contém um exemplo de objeto *ID Token*.

¹ Elemento HTML que permite incorporação de documentos e mídias interativas em uma página.

```
1 {
2   "iss": "https://platform.example.edu",
3   "sub": "a6d5c443-1f51-4783-ba1a-7686ffe3b54a",
4   "aud": ["962fa4d8-bcbf-49a0-94b2-2de05ad274af"],
5   "exp": 1510185728,
6   "iat": 1510185228,
7   "azp": "962fa4d8-bcbf-49a0-94b2-2de05ad274af",
8   "nonce": "fc5fdc6d-5dd6-47f4-b2c9-5d1216e9b771",
9   "name": "Ms Jane Marie Doe",
10  "given_name": "Jane",
11  "family_name": "Doe",
12  "middle_name": "Marie",
13  "picture": "https://platform.example.edu/jane.jpg",
14  "email": "jane@platform.example.edu",
15  "locale": "en-US",
16  "https://purl.imsglobal.org/spec/lti/claim/deployment_id":
17    "07940580-b309-415e-a37c-914d387c1150",
18  "https://purl.imsglobal.org/spec/lti/claim/message_type":
19    "LtiResourceLinkRequest",
20  "https://purl.imsglobal.org/spec/lti/claim/version": "1.3.0",
21  "https://purl.imsglobal.org/spec/lti/claim/roles": [
22    "Student",
23    "Learner"
24  ],
25  "https://purl.imsglobal.org/spec/lti/claim/context": {
26    "id": "c1d887f0-a1a3-4bca-ae25-c375edcc131a",
27    "label": "ECON 1010",
28    "title": "Economics as a Social Science",
29    "description": "An introduction.",
30    "type": ["CourseOffering"]
31  },
32  "https://purl.imsglobal.org/spec/lti/claim/tool_platform": {
33    "guid": "ex/48bbb541-ce55-456e-8b7d-ebc59a38d435",
34    "contact_email": "support@platform.example.edu",
35    "description": "An Example Tool Platform",
36    "name": "Example Tool Platform",
37    "url": "https://platform.example.edu",
38    "product_family_code": "ExamplePlatformVendor-Product",
39    "version": "1.0"
40  },
41  "https://purl.imsglobal.org/spec/lti/claim/target_link_uri":
42    "https://tool.example.com/lti/48320/ruix8782rs"
43 }
```

Listagem 3 – Exemplo de *ID Token* gerado no processo de lançamento

Lançamentos LTI Core são utilizados para acessar um determinado recurso disponibilizado pela Ferramenta através da Plataforma, independente do *placement*. Esse lançamento é a base do protocolo LTI e permite que ferramentas sejam integradas à plataformas de ensino. O *ID Token* gerado por esse tipo de lançamento terá dois campos adicionais:

- https://purl.imsglobal.org/spec/lti/claim/message_type: Tipo de lançamento *LtiResourceLinkRequest*.
- https://purl.imsglobal.org/spec/lti/claim/resource_link: Informações do recurso acessado.

Lançamentos LTI *Deep Linking* são utilizados para iniciar o fluxo do serviço *Deep Linking*, que será descrito da Seção 2.3.2. O *ID Token* gerado por esse lançamento terá dois campos adicionais:

- https://purl.imsglobal.org/spec/lti/claim/message_type: Tipo de lançamento *LtiDeepLinkingRequest*.
- https://purl.imsglobal.org/spec/lti-dl/claim/deep_linking_settings: Informações utilizadas pelo serviço *Deep Linking*.

Adicionalmente, o objeto *ID Token* também deverá conter campos específicos contendo informações sobre os demais serviços suportados:

- <https://purl.imsglobal.org/spec/lti-nrps/claim/roleservice>: Informações utilizadas pelo serviço *Names and Roles Provisioning*, descrito na Seção 2.3.3.
- <https://purl.imsglobal.org/spec/lti-ags/claim/endpoint>: Informações utilizadas pelo serviço *Assignment and Grades*, descrito na Seção 2.3.4.

2.3.2 Deep Linking

O serviço LTI 1.3 *Deep Linking* (VERVOORT; PRESTON, 2019) pode ser utilizado para permitir que uma Plataforma integre mais facilmente o conteúdo disponibilizado por uma Ferramenta.

Usando o lançamento LTI *Deep Linking*, descrito na Seção 2.3.1, os usuários da Plataforma podem acessar uma tela de seleção de recursos disponibilizada pela Ferramenta e, em seguida, selecionar o recurso específico apropriado para seu uso. Por fim, é utilizado o modelo de comunicação baseada em mensagens originadas na Ferramenta para retornar o recurso escolhido, que poderá ser utilizado posteriormente na Plataforma.

Diversos tipos de recursos também podem ser retornados pelo processo de *Deep Linking*, como imagens e fragmentos de HTML. Recursos são retornados através um objeto *Content Item*, que deve possuir um campo *type*, determinando o seu tipo e campos adicionais específicos para cada tipo de recurso suportado. O

tipo de *Content Item ltiResourceLink* retorna uma URL que poderá ser utilizada pela Plataforma como base para lançamentos LTI Core. A Listagem 4 exemplifica um objeto *Content Item* do tipo *ltiResourceLink*.

```
1  {
2    "type": "ltiResourceLink",
3    "title": "A title",
4    "text": "This is a link to an activity that will be graded",
5    "url": "https://lti.example.com/launchMe",
6    "icon": {
7      "url": "https://lti.example.com/image.jpg",
8      "width": 100,
9      "height": 100
10   }
11 }
```

Listagem 4 – Exemplo de Content Item do tipo ltiResourceLink

2.3.3 Names and Roles Provisioning

O serviço de *Names and Roles Provisioning* (SMYTHE et al., 2019a) permite que a Ferramenta acesse informações de usuários em um determinado contexto da Plataforma, o conjunto de informações de um determinado usuário é denominado de *membership*. O acesso a este serviço utiliza o método de comunicação baseado em serviços.

Após a realização de uma chamada bem sucedida ao serviço, a Plataforma retorna um objeto contendo os seguintes campos:

- *id*: Identificador do grupo de *memberships* retornado.
- *context*: Informações sobre o contexto no qual as *memberships* estão inseridas.
- *members*: Lista de *memberships* retornados.

Cada *membership* contém informações sobre o usuário e o seu papel no contexto em que está inserido. Os resultados podem ser limitados através de um parâmetro de URL *limit*, e filtrados pelo papel do usuário, através de um parâmetro *role*.

O objeto retornado pelo serviço *Names and Roles Provisioning* é chamado de *application/vnd.ims.lti-nrps.v2.membershipcontainer+json*.

2.3.4 Assignment and Grades

O serviço *Assignment and Grades* (VERVOORT et al., 2019) pode ser utilizado pela Ferramenta para manipular linhas de notas na Plataforma através de três

subserviços, *Line Items*, *Score* e *Result*. Todos acessados utilizando o método de comunicação baseado em serviços.

O subserviço *Line Items* é utilizado para buscar, criar, modificar e deletar linhas de nota em um contexto específico da Plataforma. São definidos dois tipos de objeto utilizados por este subserviço:

- *application/vnd.ims.lis.v2.lineitem+json*: Representa uma linha de nota.
- *application/vnd.ims.lis.v2.lineitemcontainer+json*: Representa uma lista de linhas de nota.

Os subserviços *Score* e *Result* são utilizados, respectivamente, para enviar e receber notas da Plataforma. São definidos dois tipos objeto utilizados por este subserviço:

- *application/vnd.ims.lis.v1.score+json*: Representa uma nota.
- *application/vnd.ims.lis.v2.resultcontainer+json*: Representa uma lista de nota.

O subserviço *Score* também pode ser utilizado para realizar submissões sem nota para serem avaliadas posteriormente. Plataformas podem expandir o subserviço para aceitarem tipos proprietários de submissões, como URLs e arquivos.

2.3.5 Extensões

O conjunto de serviços LTI *Advantage*, formado inicialmente pelos 3 serviços descritos anteriormente, permite que extensões sejam desenvolvidas na forma de novos serviços, à medida que novas funcionalidades se tornam necessárias.

Um dos serviços de extensão é o serviço *Dynamic Registration*, que torna o processo de registro entre Plataformas e Ferramentas totalmente automático. Devido a sua utilidade, este serviço foi rapidamente adotado por diversos LMSs, como Moodle², D2L³ e Sakai⁴.

² <https://moodle.org/>

³ <https://www.d2l.com/>

⁴ <https://www.sakailms.org/>

3 Biblioteca LtijS

Logo após o lançamento da versão 1.3 do protocolo LTI em 2019, havia uma falta evidente de tutoriais e outros recursos sobre a atualização, o que dificultava sua adoção pelo público geral. Com o objetivo de abstrair a complexidade da especificação, lidando com os protocolos de autenticação e formatação de mensagens, e fornecendo métodos através dos quais desenvolvedores pudessem utilizar suas funcionalidades de maneira simples, foi criado o projeto LtijS¹, uma biblioteca de código livre para *Node JS*².

Para se entender certos aspectos do projeto apresentado, são detalhados o fluxo de funcionamento, o acesso aos serviços LTI e a capacidade de extensão da biblioteca, entretanto, os detalhes da sua implementação e utilização não fazem parte do escopo do trabalho proposto.

3.1 Funcionamento

Utilizando o *framework web Express*³, foi implementada uma *API REST* encarregada de receber, validar e executar todas as etapas do processo de lançamento LTI, de acordo com a especificação do protocolo descrito na Seção 2.3. Após a conclusão de um lançamento LTI bem sucedido, a biblioteca salva o *ID Token* em um banco de dados.

Posteriormente, um JWT assinado *ltik* é gerado para servir de identificador único para o contexto do lançamento, contendo identificadores de Plataforma, de Usuário e de Recurso. A Listagem 5 contém um exemplo de objeto utilizado para gerar um *token ltik*.

```
1  {
2    "platformUrl": "http://localhost/moodle",
3    "clientId": "RlaFd4JpxEMKcYu",
4    "deploymentId": "29",
5    "contextId": "2_23",
6    "user": "2"
7  }
```

Listagem 5 – Exemplo de um objeto gerador de *ltik*

Por último, a requisição é redirecionada para a rota alvo, com o *token ltik* sendo passado como parâmetro de URL adicional. A Figura 4 ilustra o fluxo de funcionamento descrito nesta seção.

¹ <https://cvmcosta.me/ltijs/>

² <https://nodejs.org/>

³ <https://expressjs.com/>

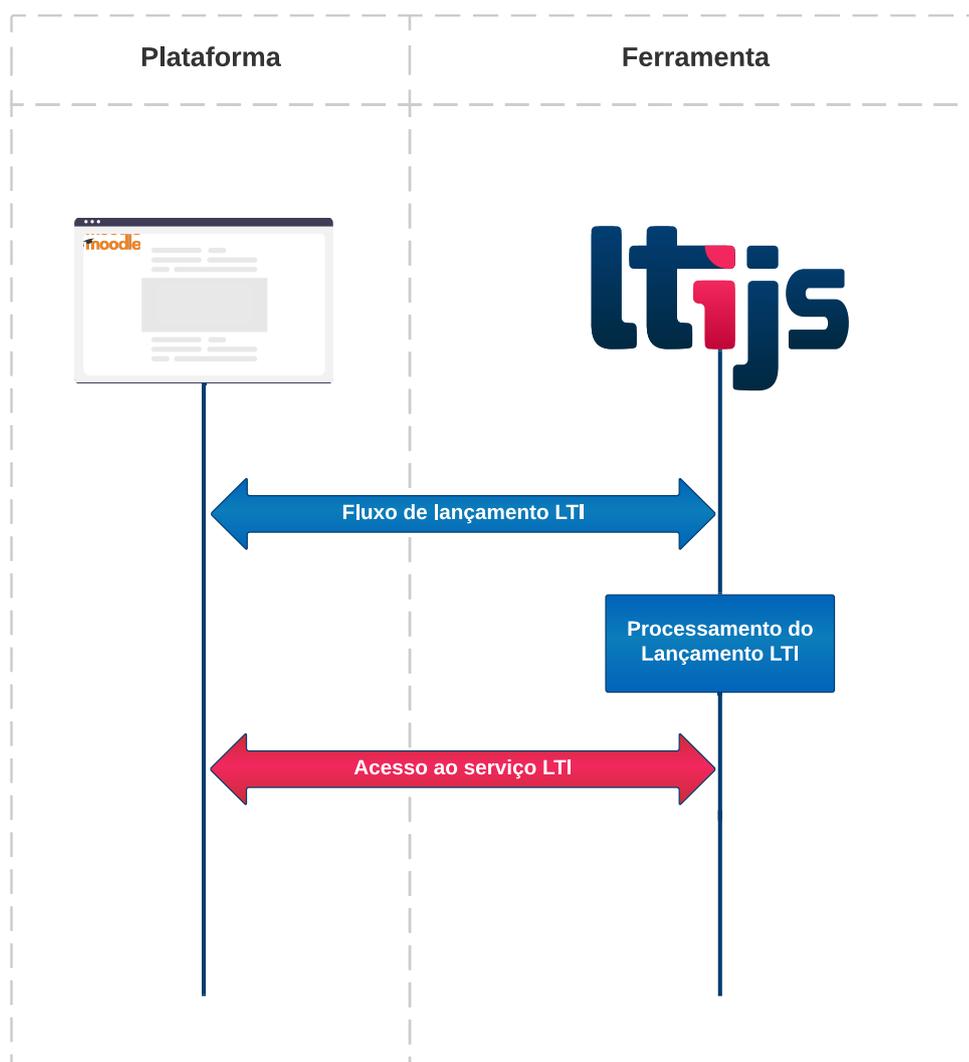


Figura 4 – Fluxo de funcionamento da biblioteca Ltijis. Fonte: Autor do trabalho

Além de se encarregar dos lançamentos LTI, a biblioteca Ltijis também implementa todos os *endpoints* necessários para a utilização da Ferramenta. Os *endpoints* implementados e suas funções são, respectivamente:

- */login*: *Endpoint* de inicialização do fluxo de login *OpenID Connect*.
- */keys*: *Endpoint* de coleção de chaves públicas que serão utilizados pela Plataforma para validar mensagens assinadas vindas da Ferramenta.
- */register*: *Endpoint* de inicialização do fluxo de registro *Dynamic Registration*.

3.2 Usabilidade

A biblioteca Ltijis oferece suporte aos três serviços LTI principais: *Deep Linking*, *Names and Roles* e *Assignment and Grades*, descritos, respectivamente, nas seções 2.3.2, 2.3.3 e 2.3.4. O serviço de extensão *Dynamic Registration*, descrito na Seção 2.3.5, também é suportado. Todos esses serviços podem ser acessados através de métodos disponibilizados pela biblioteca.

```
1 // Serviço Names and Roles
2 ltjjs.NamesAndRoles.getMembers(idToken)
3 // Serviço Assigment and Grades
4 ltjjs.Grades.getScores(idToken, lineitem)
5 // Serviço Deep Linking
6 ltjjs.DeepLinking.createDeepLinkingForm(idToken, contentItems)
7 // Serviço Dynamic Registration
8 ltjjs.DynamicRegistration.register(configuration, token)
```

Listagem 6 – Exemplos de métodos de acesso a serviços

Como exemplificado na Listagem 6, os métodos de acesso a serviços recebem um objeto *ID Token* como parâmetro, este é utilizado para montar as requisições adequadas a cada serviço, e gerar o *token* de acesso à Plataforma, quando necessário.

Adicionalmente, também são disponibilizados métodos para a criação e manipulação de registros de Plataformas, necessários para o correto funcionamento do protocolo LTI, como especificado na Seção 2.3. A Listagem 7 contém um exemplo do método `ltjjs.registerPlatform`, utilizado para criar novos registros de Plataforma.

```
1 ltjjs.registerPlatform({
2   url: 'https://platform.url',
3   name: 'Platform Name',
4   clientId: 'TOOLCLIENTID',
5   authenticationEndpoint: 'https://platform.url/auth',
6   accesstokenEndpoint: 'https://platform.url/token',
7   authConfig: {
8     method: 'JWK_SET',
9     key: 'https://platform.url/keyset'
10  }
11 })
```

Listagem 7 – Exemplo de método de registro de plataformas

O objeto Plataforma implementado pela biblioteca Ltjjs é composto pelos seguintes campos:

- `url`: URL da Plataforma. Utilizada como identificador em mensagens.
- `name`: Nome da Plataforma.
- `clientId`: *Client ID* fornecido pela Plataforma. Utilizado pelo protocolo *OAuth 2.0*.
- `authenticationEndpoint`: Endereço de autenticação de usuários da Plataforma. Utilizado para iniciar o fluxo de autenticação *OpenID Connect*.

- `accessTokenEndpoint`: Endereço do servidor de autorização da Plataforma. Utilizado para gerar *tokens* de acesso de acordo com o protocolo *OAuth 2.0*.
- `authConfig.method`: Método de autenticação de mensagens da Plataforma.
- `authConfig.key`: Chave pública ou endereço de coleção de chaves públicas. Utilizado para validar mensagens vindas da Plataforma.

3.3 Extensões

A biblioteca Ltij expõe, através de um objeto `ltijs.app`, a instância do servidor *Express* usada para construir a API base. Esse objeto pode ser utilizado para estender a API e criar rotas adicionais para acessar serviços específicos, ou receber lançamentos LTI, como exemplificado na Listagem 8.

```
1 // Cria um rota HTTP GET para receber lançamentos
2 ltij.app.get('/launch', async (req, res) => {
3   // Retornando arquivo HTML
4   return res.sendFile(path.join(__dirname, './index.html'))
5 })
```

Listagem 8 – Exemplo de criação de rota de lançamento

Toda rota criada pela biblioteca Ltij tem seu acesso controlado por um *middleware* de autenticação que checa a presença do *token ltik*, podendo ser passado como parâmetro de URL, no corpo da requisição ou em um cabeçalho HTTP de autorização do tipo *Bearer*. Caso o *token* não seja encontrado, o *middleware* de autenticação retorna um erro com o *status* HTTP 401 Unauthorized (Não autorizado).

Após a autenticação, o *token ltik* é utilizado para recuperar o *ID Token* relacionado ao lançamento, e disponibilizá-lo dentro da rota acessada através de um objeto `res.locals.token`. A Listagem 9 exemplifica como esse objeto pode ser utilizado para chamar o método de acesso ao serviço *Names and Roles Provisioning*.

```
1 // Cria um rota HTTP GET para buscar lista de membros
2 ltij.app.get('/members', async (req, res) => {
3   const idToken = res.locals.token
4   const members = await ltij.NamesAndRoles.getMembers(idToken)
5   return res.send(members)
6 })
```

Listagem 9 – Exemplo de uso do objeto `res.locals.token`

Caso uma rota precise ser acessada fora do contexto de um lançamento LTI, ou seja, sem acesso a um *token ltik*, o método `ltij.whitelist(rota)` pode ser utilizado

para adicionar uma rota específica a uma lista branca para que seu acesso não seja controlado pelo *middleware* de autenticação.

Utilizando os métodos e possíveis extensões citados acima, a biblioteca Ltjs pode ser utilizada para criar Ferramentas LTI desprendendo-se dos pormenores da implementação do protocolo, como fluxos de autenticação e troca de mensagens.

4 LTI as a Service

À medida que o uso empresarial da biblioteca Ltijts começou a crescer, surgiu a necessidade da criação de uma solução mais completa, que pudesse ser utilizada juntamente a aplicações já consolidadas, requerendo mínimas alterações ao código da aplicação e sendo agnóstica de linguagem. Com o objetivo de suprir esses requisitos, o projeto *LTI as a Service* (LTiaaS)¹ foi criado.

O sistema LTiaaS funciona como uma *API REST* posicionada entre uma aplicação alvo e as Plataformas com as quais ela deseja se comunicar, encarregando-se de mediar a comunicação entre as duas partes e cuidando da validação e formatação das informações pertinentes aos fluxos de dados que compõem o protocolo LTI. Essa união entre LTiaaS e aplicação alvo deve resultar em uma Ferramenta LTI completa, como ilustrado na Figura 5.



Figura 5 – Representação da arquitetura LTiaaS. Fonte: Autor do trabalho

Esta seção descreve aspectos relevantes do processo de configuração e funcionamento do sistema LTiaaS e as suas funcionalidades, implementadas através de extensões da biblioteca Ltijts, para atingir os objetivos descritos acima.

4.1 Configuração

Embora o nome *LTI as a Service* remeta a um sistema *Software as Service* (SaaS)², devido a sua maior complexidade de implementação, o modelo de distribuição utilizado inicialmente foi o modelo *Software as a Product* (SaaP)³, portanto, a configuração inicial deve ser realizada pelo usuário do sistema.

A configuração é realizada através de um arquivo JSON `dev.config.json` ou `prod.config.json`, dependendo do ambiente de implantação. Para o escopo deste

¹ <https://ltiaas.cvmcosta.com/>

² Software as a Service é um modelo de distribuição de software que consiste em fornecer ao cliente um serviço online, constante, em troca de um pagamento periódico.

³ Software as a Product é um modelo de distribuição de software que consiste em entregar uma cópia do sistema para cliente, podendo oferecer uma licença de pagamento único ou periódico.

trabalho, é relevante descrevermos apenas três dos campos de configuração possíveis, *toolUrl*, *toolDl* e *apiKey*. Os demais campos ou permanecem imutáveis a maioria das vezes, ou são campos referentes à configuração da biblioteca Ltajs.

Os campos *toolUrl* e *toolDl* são os endereços alvo na aplicação que serão acessados pelo sistema LTaaS após receber lançamentos LTI *Core* e *Deep Linking* válidos respectivamente.

Além dos endereços alvo, também deve ser declarado o campo *apiKey*, uma chave de API secreta que será utilizada para validar requisições feitas pela aplicação. A Listagem 10 mostra um exemplo de arquivo de configuração *dev.config.json*.

```
1  {
2    "toolUrl": "http://app.example.com/launch",
3    "toolDl": "http://app.example.com/deeplink",
4    "apiKey": "APIKEY",
5    "ltiKey": "EXAMPLEKEY",
6    "devMode": true,
7    "port": 3000,
8    "dbHost": "localhost",
9    "dbName": "database",
10   "dbUser": "databaseUser",
11   "dbPass": "databasePass"
12 }
```

Listagem 10 – Exemplo de arquivo de configuração *dev.config.json*

O sistema LTaaS pode ser inicializado através de dois comandos, o comando *npm start*, inicia o sistema no modo de desenvolvimento, usando o arquivo de configuração *dev.config.json*, enquanto o comando *npm run start:prod* inicia o sistema em modo de produção, usando o arquivo de configuração *prod.config.json*.

4.2 Funcionamento

Como ilustrado na Figura 6, o sistema LTaaS se encarrega de receber os lançamentos LTI direcionados à aplicação e, ao fim do processo de validação, redireciona a requisição para o endereço alvo correspondente ao tipo do lançamento, passando o *token* de contexto *ltik*, descrito na Seção 3.1, nos parâmetros de URL.

Para que a aplicação tenha acesso às informações contidas no *ID Token* gerado por um lançamento LTI e aos serviços disponibilizados pelo protocolo, é disponibilizada uma *API REST* com cinco *endpoints* de acesso a serviço, descritos na Seção 4.4. Para assegurar que esses *endpoints* sejam acessados somente por aplicações autorizadas, são implementados os esquemas de autenticação descritos na Seção 4.3.

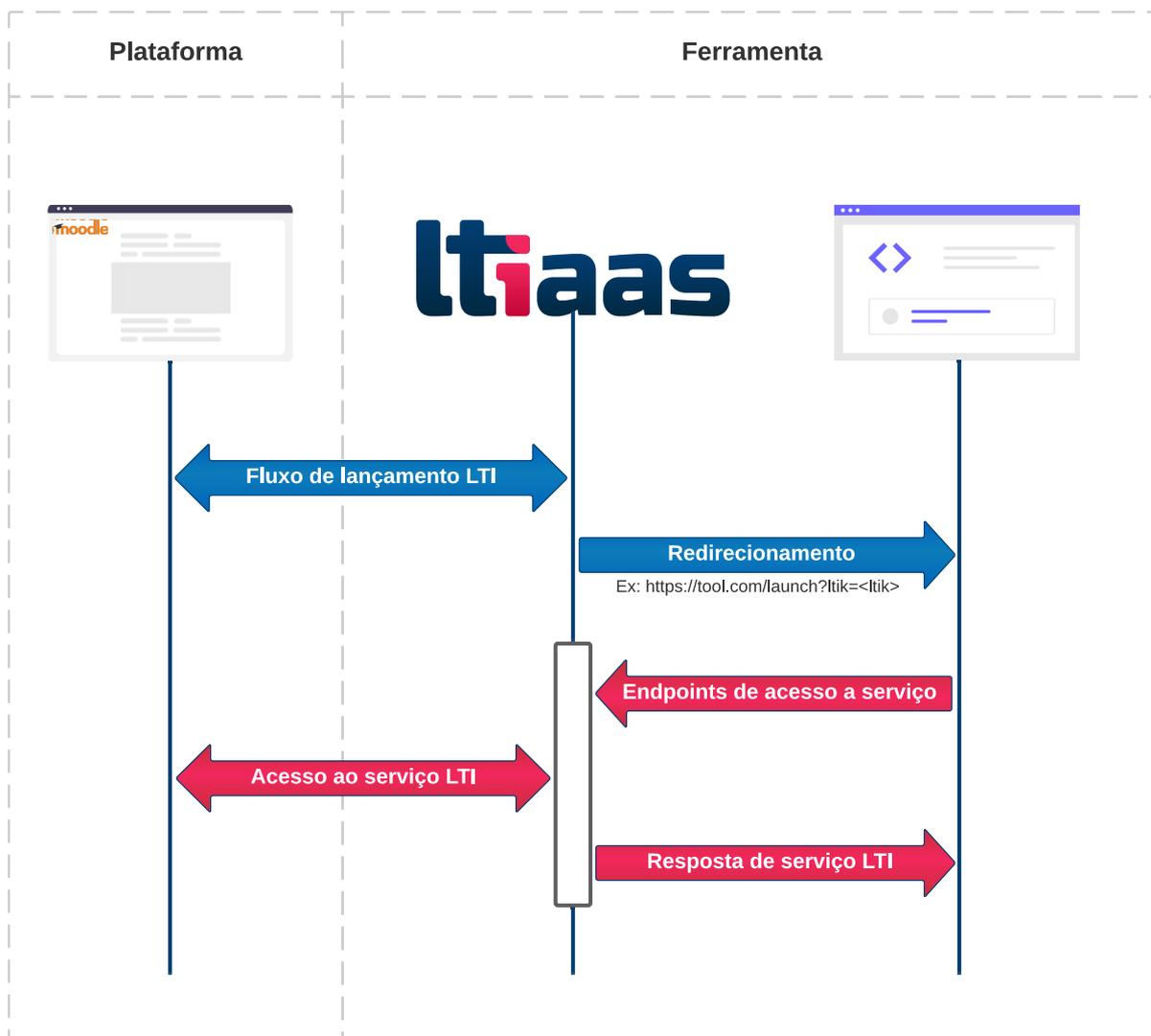


Figura 6 – Fluxo de funcionamento do sistema LTiaaS. Fonte: Autor do trabalho

4.3 Autenticação

Todos os *endpoints* de acesso a serviço implementados pelo sistema LTiaaS são protegidos através de dois possíveis esquemas de autenticação, *LTIK-AUTH-V1* e *Bearer Token*, ambos utilizando a chave de API configurada previamente.

4.3.1 Esquema de autenticação *LTIK-AUTH-V1*

Caso o *endpoint* precise de acesso às informações contidas no *ID Token*, ele será protegido através do esquema de autenticação *LTIK-AUTH-V1*. O esquema foi criado como uma extensão do *middleware* de autenticação padrão da biblioteca LtijS, descrito na Seção 3.3, e exige que o *token ltik* e a chave de API sejam passados através de um cabeçalho HTTP *Authorization* como representado na Listagem 11:

```
Authorization: LTIK-AUTH-V1 Token=<ltik>, Additional=Bearer <API_KEY>
```

Listagem 11 – Exemplo de cabeçalho de autorização *LTIK-AUTH-V1*

Endpoints que implementarem o esquema de autenticação *LTIK-AUTH-V1* terão o cabeçalho *HTTP Authorization* gerado como descrito na Listagem 12.

```
1 // Busca ltik nos parâmetros de URL
2 const ltik = req.query('ltik')
3
4 // Monta cabeçalho da requisição
5 const ltikAuth = `LTIK-AUTH-V1 Token=${ltik}`
6 const bearerAuth = `Additional=Bearer ${API_KEY}`
7 const headers = { Authorization: `${ltikAuth}, ${bearerAuth}` }
```

Listagem 12 – Exemplo de geração de cabeçalho de autorização *LTIK-AUTH-V1*

4.3.2 Esquema de autenticação *Bearer Token*

Os *endpoints* que não precisam de acesso às informações contidas no *ID Token* são adicionados à lista branca da biblioteca *Ltijs*, descrita na Seção 3.3, e são protegidos através do esquema de autenticação *Bearer token*. O esquema exige que a chave de API seja passada através de um cabeçalho *Authorization* como representado na Listagem 13:

```
Authorization: Bearer <API_KEY>
```

Listagem 13 – Exemplo de cabeçalho de autorização *Bearer Token*

Endpoints que implementarem o esquema de autenticação *Bearer token* terão o cabeçalho *HTTP Authorization* gerado como descrito na Listagem 14.

```
1 const headers = { Authorization: `Bearer ${API_KEY}` }
```

Listagem 14 – Exemplo de geração de cabeçalho de autorização *Bearer Token*

A natureza dos esquemas de autenticação descritos acima faz com que a comunicação entre a aplicação e o sistema *LTIaaS* deva ser realizada de Servidor para Servidor, para evitar que a chave de API seja exposta no navegador através de ferramentas como o inspetor de rede.

4.4 *Endpoints* de serviço

Para permitir que a aplicação tenha acesso aos serviços *LTI* suportados pela biblioteca *Ltijs*, foram implementados cinco *endpoints*: *ID Token*, *Memberships*, *Line Items*, *Deep Linking* e *Platforms*. Cada *endpoint* representa um serviço ou funcionalidade necessário para a construção de uma Ferramenta *LTI*. Esta seção irá descrever cada um dos *endpoints* citados acima e fornecer exemplos de sua utilização.

4.4.1 ID Token

O *endpoint ID Token* pode ser utilizado pela aplicação para recuperar o *ID Token* gerado por um determinado lançamento. O acesso se dá através de uma requisição *HTTP GET* para a rota */api/idtoken* obedecendo o esquema *LTIAAS-V1* de autenticação. A Listagem 15 contém um exemplo de código utilizado para acessar o *endpoint ID Token*.

```
1 // Monta URL da requisição
2 const url = LTIAAS_URL + '/api/idtoken'
3 // Realiza a requisição
4 const response = await request.get(url,
5   { headers: headers }).json()
6 return response
```

Listagem 15 – Exemplo de código de acesso ao *endpoint ID Token*

Como ilustrado na Listagem 16, as informações contidas no *ID Token* são formatadas e reorganizadas pelo sistema *LTIAAS* em três campos principais para tornar o acesso à informação mais direto e intuitivo, *user*, *platform* e *launch*. Campos presentes no *ID Token* original utilizados especificamente para acessar serviços *LTI* não são relevantes à aplicação nesse contexto, portanto, são omitidos.

```
1 {
2   user: {
3     // Informações sobre Usuário
4     ...
5   },
6   platform: {
7     // Informações sobre Plataforma
8     ...
9   },
10  launch: {
11    // Informações sobre lançamento
12    ...
13  }
14 }
```

Listagem 16 – Exemplo de estrutura do objeto *ID Token*

O campo *user*, exemplificado na Listagem 17, contém informações sobre o usuário que iniciou o lançamento, como ID do usuário na Plataforma, e-mail, nome e papel no contexto do lançamento. Esses dados podem ser utilizados para criação de fluxos de *Single Sign-On* (SSO)⁴ e customização da interface apresentada.

⁴ Processo de autenticação única onde o usuário tem acesso a vários sistemas precisando se autenticar somente em um. Neste caso, o *LMS*.

Informações pessoais como nome e e-mail podem ser omitidas de acordo com a política de segurança de cada Plataforma.

```
1  {
2    user: {
3      id: '2',
4      given_name: 'Admin',
5      family_name: 'User',
6      name: 'Admin User',
7      email: 'admin@lms.example.com',
8      roles: [
9        'Administrator',
10       'Instructor',
11       'Administrator'
12     ]
13   },
14   ...
15 }
```

Listagem 17 – Exemplo de campo *user* do objeto *ID Token*

O campo *platform*, exemplificado na Listagem 18, contém informações sobre a Plataforma onde o lançamento se originou, como ID, nome, URL e *Client ID*.

```
1  {
2    ...,
3    platform: {
4      id: '0c41aa0849215449d4298e58f7626c68',
5      url: 'https://lms.example.com',
6      clientId: 'CLIENTID',
7      deploymentId: '1',
8      product_family_code: 'canvas',
9      version: '2020073000',
10     name: 'LMS',
11     description: 'LMS'
12   },
13   ...
14 }
```

Listagem 18 – Exemplo de campo *platform* do objeto *ID Token*

Por fim, o campo *launch*, exemplificado na Listagem 19, contém informações sobre o contexto do lançamento, como curso, recurso e parâmetros customizados. Relacionando as informações de plataforma e contexto, contidas nos campos *platform* e *launch*, respectivamente, pode ser criada uma identificação universal do contexto de um lançamento.

```
1  {
2  ...
3  launch: {
4    type: 'LtiResourceLinkRequest',
5    target: 'http://tool.example.com?resource=value1',
6    context: {
7      id: '2',
8      label: 'course',
9      title: 'Course',
10     type: [
11       'CourseSection'
12     ]
13   },
14   resource: {
15     title: 'Activity',
16     id: '1'
17   },
18   custom: {
19     locale: 'en',
20     placement: 'assignment-selection'
21   }
22 }
23 }
```

Listagem 19 – Exemplo de campo *launch* do objeto *ID Token*

4.4.2 Memberships

O *endpoint Memberships* pode ser utilizado pela aplicação para acessar o serviço de *Names and Roles Provisioning*, descrito na Seção 2.3.3, e recuperar uma lista dos usuários inseridos no contexto de um determinado lançamento. O acesso se dá através de uma requisição *HTTP GET* para a rota */api/memberships* obedecendo o esquema *LTIK-AUTH-V1* de autenticação. Os filtros suportados pelo serviço podem ser passados para o *endpoint* como parâmetros de URL, como exemplificado na Listagem 23, Apêndice A.

O objeto retornado pela requisição segue o padrão *application/vnd.ims.lti-nrps.v2.membershipcontainer+json*, com um campo adicional *next* contendo a URL retornada pela Plataforma como cabeçalho *Link* para auxiliar na navegação entre as páginas de usuários caso a lista retornada não esteja completa. Isso é exemplificado na Listagem 24, Apêndice A.

As informações retornadas pelo *endpoint Memberships* podem ser utilizadas para realizar o cadastro automático de usuários na aplicação após um primeiro contato, e manter essa lista de usuários atualizada através de subseqüentes consultas.

4.4.3 Line Items

O *endpoint Line Items* pode ser utilizado pela aplicação para acessar o serviço de *Assignment and Grades* e seus subserviços, descritos na Seção 2.3.4, e efetuar manipulação das linhas de notas pertencentes ao contexto de um determinado lançamento. O acesso ao *endpoint* e todas as suas sub-rotas é controlado pelo esquema de autenticação *LTIK-AUTH-V1*.

Para buscar linhas de notas, a aplicação deve realizar uma requisição *HTTP GET* para a rota */api/lineitems*. Os filtros suportados pelo subserviço *Line Items* podem ser passados como parâmetros de URL, como exemplificado na Listagem 25, Apêndice A.

O objeto retornado pela requisição, representado na Listagem 26, Apêndice A, contém um campo *lineItems* e campos adicionais *next*, *first* e *last* contendo as URLs retornadas pela Plataforma como cabeçalho *Link* para auxiliar na navegação entre as páginas de linhas de nota caso a lista retornada não esteja completa. O campo *lineItems* contém um objeto do tipo *application/vnd.ims.lis.v2.lineitemcontainer+json*.

Para criar uma linha de nota, a aplicação deve realizar uma requisição *HTTP POST* para a rota */api/lineitems*. O corpo da requisição deve conter um objeto do tipo *application/vnd.ims.lis.v2.lineitem+json*. A Listagem 27, Apêndice A contém um exemplo de código utilizado para realizar a criação de uma linha de nota.

Para buscar, modificar, ou deletar uma linha de nota, a aplicação pode realizar, respectivamente, requisições *HTTP GET*, *PUT* ou *DELETE* para a rota */api/lineitems/:id*, passando o ID da linha de nota como sub-rota. O ID de uma linha de nota é uma URL, portanto seus caracteres especiais precisam ser escapados para ser utilizado como parte da rota, como exemplificado pela Listagem 28, Apêndice A.

Os subserviços *Score* e *Result*, podem ser utilizados pela aplicação para recuperar e enviar notas para uma determinada linha de nota da Plataforma. Ambos podem ser acessados através da rota */api/lineitems/:id/score*.

Para enviar notas para uma linha de nota, a aplicação deve realizar uma requisição *HTTP POST*. O corpo da requisição deve conter um objeto do tipo *application/vnd.ims.lis.v1.score+json*. A Listagem 29, Apêndice A contém um exemplo de código utilizado para realizar o envio de nota.

Para receber notas de uma linha de nota, deve ser realizada uma requisição *HTTP GET*. Os filtros suportados pelo subserviço *Result* podem ser passados como parâmetros de URL. Isso pode ser observado na Listagem 30, Apêndice A. O objeto retornado pela requisição, que pode ser visto na Listagem 31, Apêndice A, contém um campo *scores* e campos adicionais *next*, *first* e *last* contendo as URLs retornadas pela Plataforma como cabeçalho *Link* para auxiliar na navegação entre as páginas de notas caso a lista retornada não esteja completa. O campo *scores* contém um objeto do tipo *application/vnd.ims.lis.v2.resultcontainer+json*.

4.4.4 Deep Linking

Como descrito na Seção 2.3.2, ao fim do fluxo de *Deep Linking*, a Ferramenta deve submeter à Plataforma, através de um formulário, os recursos selecionados pelo usuário formatados em um JWT assinado. A criação do JWT e formulário de auto-submissão pode ser realizada através do *endpoint Deep Linking*. A aplicação deve realizar uma requisição *HTTP POST* para a rota */api/deeplinking/form* obedecendo o esquema *LTIK-AUTH-V1* de autenticação. No corpo da requisição deve haver um objeto com o campo *contentItems* contendo uma lista de *Content Items* válidos, especificados na Seção 2.3.2. A Listagem 32, Apêndice A, contém um exemplo do código utilizado para acessar o *endpoint Deep Linking*.

O objeto retornado pela requisição, exemplificado na Listagem 33, Apêndice A, contém em um campo *form* o formulário de auto-submissão que deve ser inserido no HTML da página para finalizar o processo de *Deep Linking*.

4.4.5 Platforms

Para que a comunicação entre Ferramenta e Plataforma seja estabelecida, é necessário que ambas as partes tenham conhecimento de diversos identificadores uma da outra, como dito na Seção 2.3. Através do *endpoint Platforms*, a aplicação pode criar e manipular registros de Plataforma. O acesso ao *endpoint* e todas as suas sub-rotas é controlado pelo esquema de autenticação *Bearer Token*, visto que não há necessidade de nenhuma informação vinda de um lançamento LTI.

Para criar um registro de Plataforma, a aplicação deve realizar uma requisição *HTTP POST* para a rota */api/platforms*. O corpo da requisição deve conter um objeto do tipo Plataforma, como exemplificado na Seção 3.2. A Listagem 34, Apêndice A, contém um exemplo de código utilizado para realizar a criação de um registro de Plataforma.

Para buscar plataformas, a aplicação deve realizar uma requisição *HTTP GET* para a rota */api/platforms*. Como exemplificado na Listagem 35, Apêndice A. O objeto retornado pela requisição, exemplificado na Listagem 36, Apêndice A, contém, no campo *platforms*, uma lista de Plataformas. Os campos adicionais *id* e *publicKey*, representam, respectivamente, o ID e a chave pública gerados automaticamente para a Plataforma na hora de seu registro.

Para buscar, modificar, ou deletar um registro de Plataforma específico, a aplicação pode realizar, respectivamente, requisições *HTTP GET*, *PUT* ou *DELETE* para a rota */api/platforms/:id*, passando o ID da Plataforma como sub-rota. A Listagem 37, Apêndice A, contém um exemplo de código utilizado para atualizar um registro de Plataforma.

Através das funcionalidades descritas acima, o sistema LTIaaS pode ser utilizado para transformar aplicações consolidadas em Ferramentas LTI completas. No Capítulo 5 serão apresentados exemplos de tipos de integrações construídas em cima da estrutura apresentada neste capítulo.

5 Casos de uso

Utilizando os serviços disponibilizados pelo protocolo LTI, existem diversas maneiras através das quais aplicações externas podem estender as funcionalidades de uma plataforma de ensino. Para demonstrar a capacidade do sistema LTIaaS de atender as necessidades de tipos distintos de integração, este capítulo irá descrever três tipos de caso de uso, Aplicação única, Repositório de aplicações e Moodle como Ferramenta LTI 1.3, apresentando exemplos reais de Ferramentas baseadas em cada tipo.

Ao descrevermos as diversas Ferramentas apresentadas neste capítulo, não será citada a utilização do *endpoint Platforms*, descrito na Seção 4.4.5, pois se entende que sua utilização está implícita, já que, como especificado na Seção 2.3, Plataformas e Ferramentas precisam ter conhecimento de identificadores umas das outras para que possam se comunicar.

5.1 Aplicação única

O caso de uso mais simples e direto do sistema LTIaaS consiste em utilizar as suas funcionalidades para transformar uma única aplicação consolidada em uma Ferramenta LTI.

Nesta seção serão descritas duas Ferramentas, Powtoon e Mobilidade Segura, cuja implementação se encaixa no caso de uso Aplicação única.

5.1.1 Powtoon

Powtoon ([Powtoon Ltd, 2021](#)) é uma plataforma de comunicação visual fundada em 2012, atendendo mais de 30 milhões de usuários entre diversos setores. Seu principal produto é um estúdio de criação de conteúdo visual que pode ser utilizado para criar facilmente apresentações animadas conhecidas popularmente como powtoons.

Com o intuito de adquirir mais espaço no mercado da tecnologia educacional, surgiu a ideia de transformar a plataforma Powtoon em uma Ferramenta LTI fornecendo todos os meios necessários para que os powtoons criados possam ser utilizados por alunos e professores como objetos de aprendizagem.

Como descrito na Seção 2.3, diferentes LMSs podem suportar lançamentos LTI de diferentes localizações, chamadas *placements*, o que deve ser levado em consideração quando se planeja a construção de uma nova integração. Com isso em mente, o LMS escolhido para servir de base para a criação da Ferramenta LTI da plataforma Powtoon foi o Canvas, pois este suporta *placements* que possibilitam uma integração bem mais completa ([Instructure, Inc, 2019](#)).

A integração LTI Powtoon no Canvas tem como alvo os *placements Global navigation, Editor button, Assignment Selection e Homework Submission*. Ao acessar

qualquer um destes, o usuário será logado automaticamente em uma conta criada ou acessada com base no e-mail informado pelo Canvas.

O *placement Global navigation* é utilizado para que a plataforma Powtoon como um todo seja acessível a partir da barra de navegação lateral do Canvas, como visto na Figura 7. A partir deste *placement* usuários podem criar, editar, e visualizar powtoons que poderão ser utilizados em outros pontos do sistema.

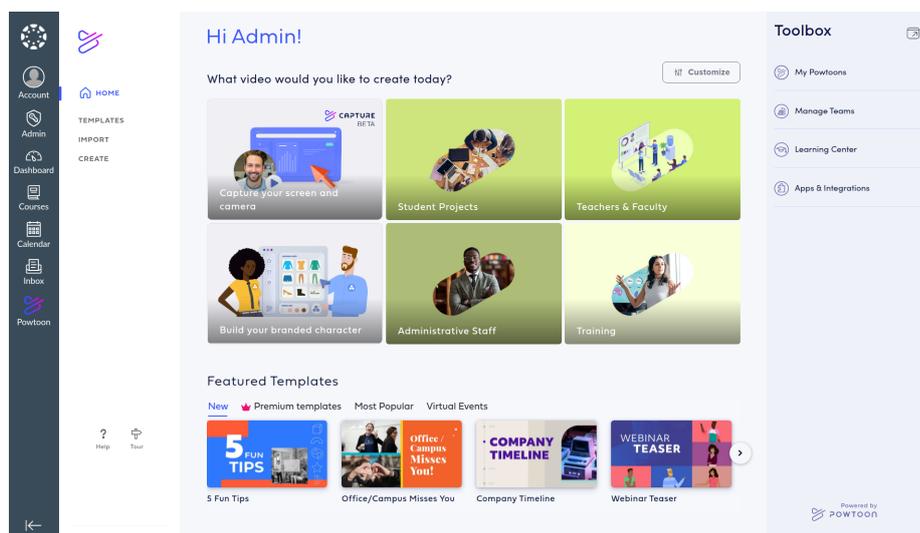


Figura 7 – Powtoon acessado pelo *placement Global navigation*. Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.

Para incorporar powtoons nas atividades criadas, professores podem utilizar os *placements Editor button* e *Assignment selection*. O *placement Editor button* pode ser acessado a partir de qualquer caixa de edição de texto no Canvas ao se clicar no botão representando a Ferramenta LTI Powtoon. O professor poderá selecionar um powtoon e este será adicionado ao texto como um link que quando clicado abrirá uma tela de visualização com o powtoon escolhido em uma nova aba. As figuras 8 e 9, mostram, respectivamente, a caixa de edição de texto com o link gerado e a tela de seleção de powtoons utilizada pelo professor.

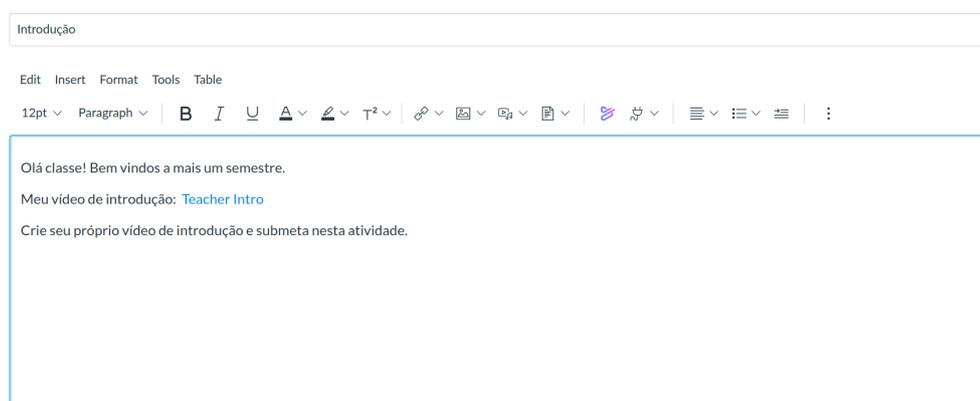


Figura 8 – Botão Powtoon acessível através da caixa de edição de texto. Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.

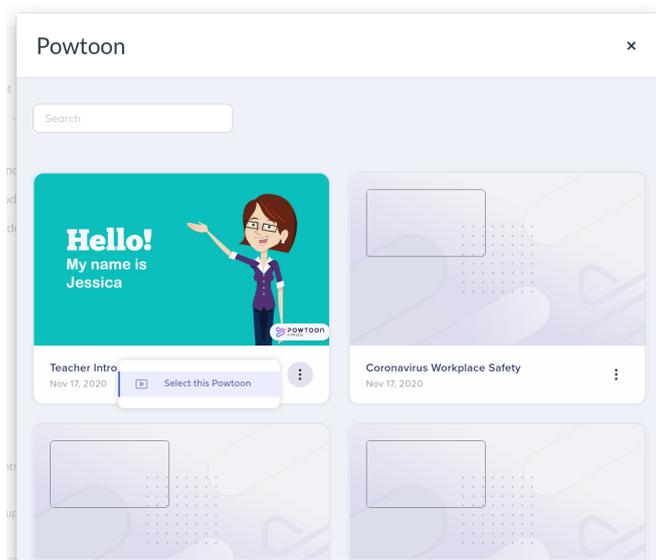


Figura 9 – Tela de seleção de powtoons. Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.

O *placement Assignment selection* pode ser utilizado para incorporar um powtoon específico, escolhido através de uma tela de seleção muito semelhante à mostrada na Figura 9, no corpo de uma atividade do tipo Ferramenta externa, de forma que ao acessar a atividade o aluno se depara diretamente com o visualizador de powtoons com o powtoon escolhido. Como mostrado na Figura 10, também estará disponível um botão "Submit Your Work Via Powtoon", através do qual um aluno pode submeter um powtoon como resposta para a atividade.

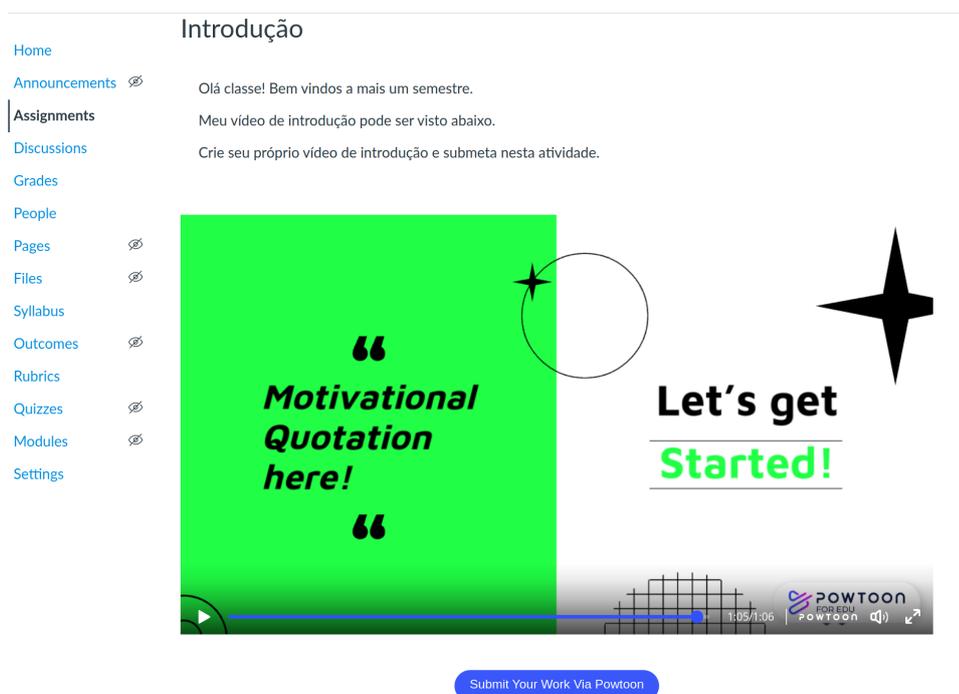


Figura 10 – Powtoon sendo visualizado no *placement Assignment selection*. Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.

Powtoons podem ser utilizados como respostas para atividades caso estas sejam do tipo Ferramenta externa ou Submissão online. Em atividades do tipo Ferramenta externa, utilizando o *placement Assignment selection*, ao clicar no botão "Submit Your Work Via Powtoon", o aluno é redirecionado para a plataforma Powtoon em uma nova aba, onde é possível criar, editar ou selecionar um powtoon para ser submetido como resposta. Atividades do tipo Submissão online usam o *placement Homework submission*, onde é disponibilizada uma tela de seleção de powtoons através da qual o aluno pode escolher o powtoon a ser submetido. A Figura 11 mostra a tela de seleção de powtoons disponível para os alunos através do *placement Homework submission*.

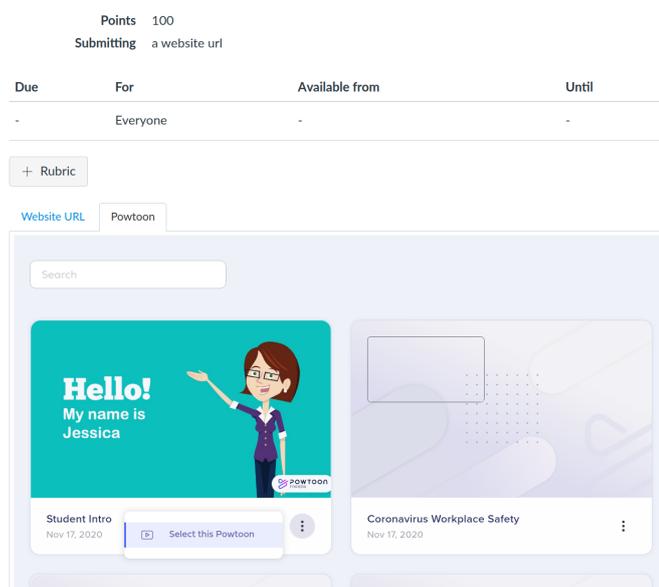


Figura 11 – Tela de seleção de powtoons no *placement Homework submission*. Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.

Por fim, professores podem visualizar e avaliar os powtoons submetidos pelos alunos através de uma ferramenta chamada *Speed Grader*, como visto na Figura 12.

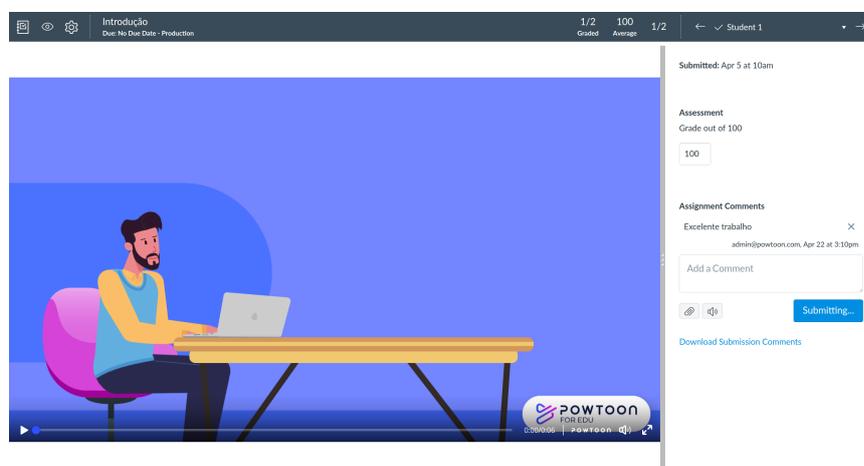


Figura 12 – Powtoon sendo avaliado em *Speed Grader*. Fonte: (Powtoon Ltd, 2021). Captura de tela criada pelo autor.

Os fluxos de uso descritos acima foram implementados utilizando os *endpoints* e demais funcionalidades disponibilizadas pelo sistema LTIaaS.

A Ferramenta LTI Powtoon implementa um sistema de Single Sign-On (SSO) construído utilizando o *endpoint ID Token*, descrito na Seção 4.4.1. O e-mail do usuário, retornado no objeto *ID Token*, é utilizado para criar uma conta nova ou, caso já exista uma conta com o mesmo e-mail, logar o usuário automaticamente na sua conta, este segundo caso exige uma etapa adicional de confirmação durante a primeira ocorrência.

As telas de seleção de powtoon encontradas nos *placements Editor button*, *Assignment selection* e *Homework submission* são acessadas após lançamentos LTI Deep Linking válidos, através do endereço alvo *toolDL*, como especificado na Seção 4.1, e utilizam o *endpoint Deep Linking*, descrito na Seção 4.4.4, para submeter uma URL unindo o endereço padrão da Ferramenta LTI Powtoon com um parâmetro de URL *embed* cujo valor é o ID do powtoon selecionado. A listagem 20 exemplifica uma URL gerada pelo processo de seleção de powtoons.

`https://lti.powtoon.com?embed=e9SvOTaHXM`

Listagem 20 – Exemplo de URL gerada pelo processo de seleção de powtoons

As demais telas são acessadas após lançamentos LTI *Core* válidos, através do endereço alvo *toolUrl*. Lançamentos efetuados de diferentes *placements* são diferenciados a partir do parâmetro customizado *placement*, enviado em cada lançamento e acessado através campo *launch.custom* do *ID Token*. A partir desta distinção a Ferramenta LTI Powtoon decide qual tela mostrar ao usuário.

A visualização de powtoons é implementada de modo que após um lançamento LTI *Core* válido é realizado, o sistema LTIaaS repassa o parâmetro *embed*, gerado pelo processo de *Deep Linking* na tela de seleção de powtoons, para o endereço alvo *toolUrl*, que se encarrega de redirecionar o usuário para o visualizador de powtoons com o powtoon correspondente.

Finalmente, a submissão de powtoons como resposta para atividades é realizada de duas maneiras diferentes. No *placement Homework submission*, a submissão é realizada através de uma tela de seleção de powtoons, portanto o *endpoint Deep Linking* é utilizado para submeter a URL contendo o ID do powtoon escolhido.

Alternativamente, no *placement Assignment selection*, onde os powtoons são submetidos através da plataforma Powtoon acessada pelo botão "*Submit Your Work Via Powtoon*", é utilizado o *endpoint Line Items*, descrito na Seção 4.4.3, para criar ou recuperar uma linha de nota e então submeter o powtoon escolhido. Como descrito na Seção 2.3.4, o serviço de *Assignment and Grades* pode ser expandido para receber submissões não numéricas. O LMS Canvas recebe, através de um campo customizado `https://canvas.instructure.com/lti/submission`, uma URL que pode ser utilizada para realizar lançamentos LTI, neste caso é submetida a URL contendo o ID do powtoon escolhido, como descrito anteriormente.

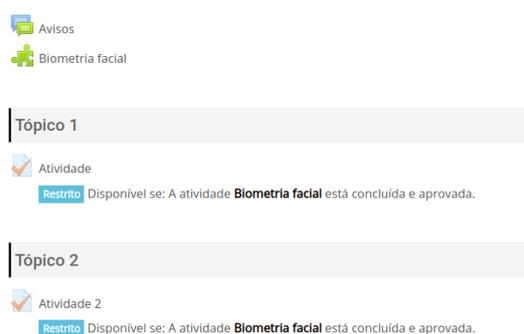
5.1.2 Mobilidade Segura

Mobilidade Segura ([Mobilidade Segura, 2021](#)) é uma empresa de soluções em mobilidade pública especializada em gestão de comportamento de condução. A empresa possui uma plataforma de ensino chamada CNH Cursos, construída utilizando o LMS Moodle, que oferece diversos cursos nas áreas de mobilidade urbana, trânsito e prevenção de acidentes.

Para controlar o acesso de usuários aos cursos oferecidos, foi criada uma aplicação que utiliza a API de biometria facial da Serpro, Datavalid ([Serpro, 2021](#)), para relacionar as informações fornecidas com uma foto do usuário tirada no momento do acesso. O sistema LTIIaaS foi utilizado para integrar esta aplicação com a plataforma CNH Cursos.

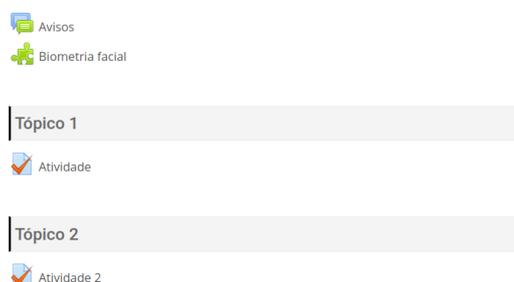
A aplicação de controle de acesso de usuários, chamada na plataforma simplesmente de Biometria facial, funciona internamente como uma Ferramenta LTI, sendo o primeiro recurso cadastrado em cada curso. As funcionalidades de restrição de acesso do Moodle são então utilizadas para garantir que todos os outros recursos daquele contexto só se tornem acessíveis caso o usuário tenha uma nota igual ou superior a 80 na atividade de biometria. A Figura 13 mostra recursos de um curso com o acesso controlado pela atividade Biometria facial.

Painel > Cursos > Miscellaneous > Curso



(a) Acesso restrito

Painel > Cursos > Miscellaneous > Curso



(b) Acesso liberado

Figura 13 – Controle de acesso pela atividade Biometria facial. Fonte: ([Mobilidade Segura, 2021](#)). Captura de tela criada pelo autor.

Ao clicar na atividade Biometria facial, o usuário é redirecionado para a tela de validação, onde deve inserir seu CPF e tirar uma foto do seu rosto, e submeter para análise. Caso a API Datavalid detecte um grau de similaridade maior ou igual a 80% entre a foto submetida e o rosto cadastrado junto ao CPF inserido, o usuário ganhará acesso ao restante dos recursos do curso. A Figura 14 mostra a tela de biometria facial.

A integração descrita acima, apesar de simples, demonstra a diversidade de casos de uso que podem ser atingidos através do protocolo LTI. A utilização do sistema LTIIaaS nesta integração consistiu em configurar o *endpoint* alvo *toolUrl*, descrito na Seção 4.1, para que o usuário fosse redirecionado para a aplicação de biometria facial após um lançamento LTI *Core*, e então utilizar o *endpoint Line Items*,

descrito na Seção 4.4.3, para submeter o percentual de similaridade gerado pela API Datavalid como nota para a atividade.

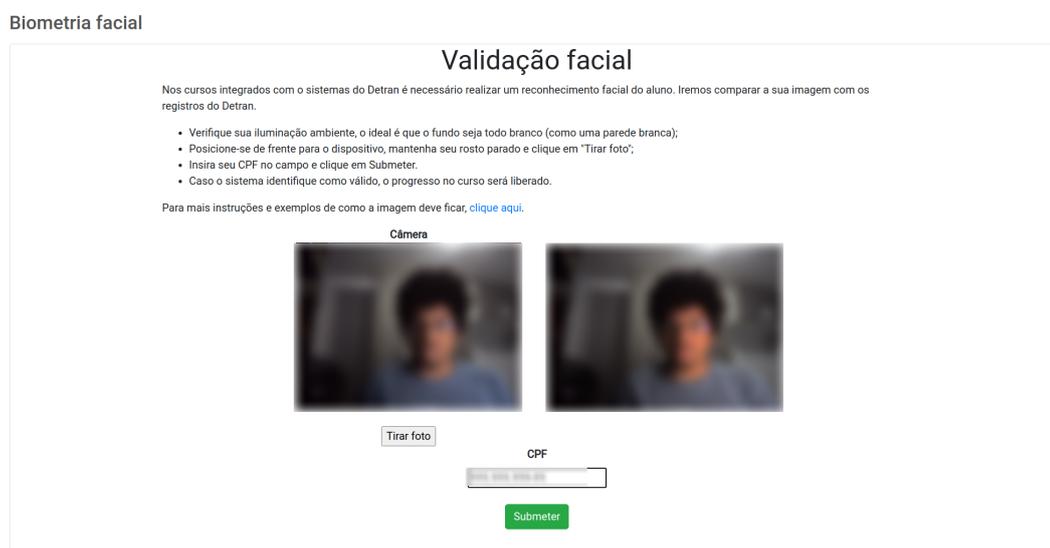


Figura 14 – Tela de biometria facial. Fonte: (Mobilidade Segura, 2021). Captura de tela criada pelo autor.

5.2 Repositório de aplicações

O sistema LTiaaS pode ser utilizado juntamente à uma simples camada de controle para permitir e organizar o acesso à um repositório de aplicações, sem que estas aplicações sejam relacionadas entre si. Desta maneira é possível servir, de uma mesma Ferramenta LTI, inúmeras aplicações distintas, fornecendo para cada uma delas todas as funcionalidades do protocolo LTI. A Figura 15 ilustra a arquitetura do tipo de caso de uso Repositório de aplicações.

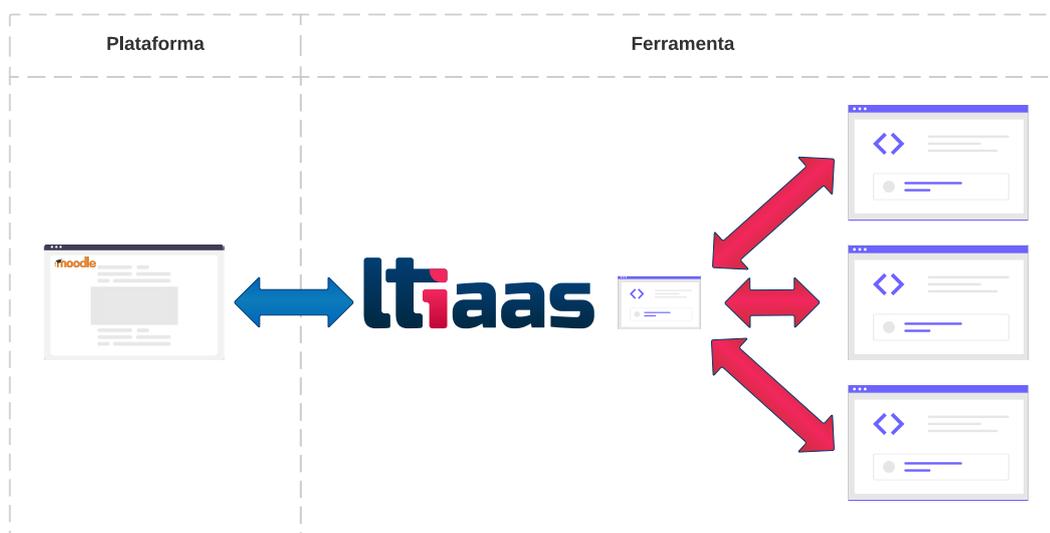


Figura 15 – Arquitetura do tipo de caso de uso Repositório de aplicações. Fonte: Autor do trabalho

Nesta seção a Ferramenta LTI UNA-SUS/UFMA será apresentada, e sua implementação será utilizada para descrever o caso de uso descrito acima.

5.2.1 UNA-SUS/UFMA

A Universidade Aberta do SUS (UNA-SUS) (UNA-SUS, 2021) foi criada em 2010 para atender às necessidades de capacitação e educação dos profissionais de saúde. Em 2019, a Universidade Federal do Maranhão (UFMA) aderiu à rede colaborativa de instituições de ensino superior que fazem parte do projeto, formando assim a UNA-SUS/UFMA (UNA-SUS/UFMA, 2021), desde então a instituição tem oferecido cursos gratuitos de especialização e aperfeiçoamento na área da saúde através do LMS Moodle, atualmente na forma da plataforma de ensino SAITE AVA.

Os cursos oferecidos pela UNA-SUS/UFMA dispõem de diversos tipos de recursos de aprendizagem, como e-books e infográficos. Para permitir que esses recursos externos possam ser integrados na plataforma SAITE AVA, foi criada uma integração LTI utilizando o sistema LTIIaaS.

Cada recurso externo criado é cadastrado em um banco de dados, onde se torna disponível para utilização como ferramenta LTI. Um professor ou administrador pode então utilizar uma tela de seleção de recursos para adicioná-lo a um curso, como mostrado na Figura 16.

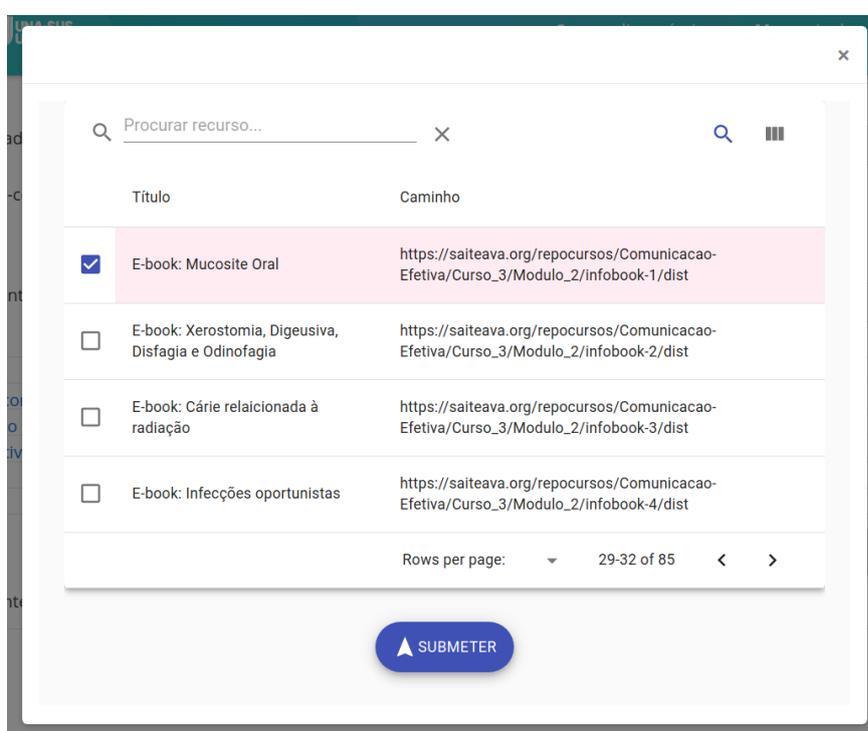


Figura 16 – Tela de seleção de recursos na plataforma SAITE AVA. Fonte: (UNA-SUS/UFMA, 2021). Captura de tela criada pelo autor.

O recurso selecionado pode então ser acessado como parte do curso, como mostrado na Figura 17. Todos os recursos externos acessados como Ferramentas LTI possuem acesso às funcionalidades disponibilizadas pelo sistema LTIIaaS, o que expande o escopo do que pode ser realizado dentro de cada uma dessas aplicações.

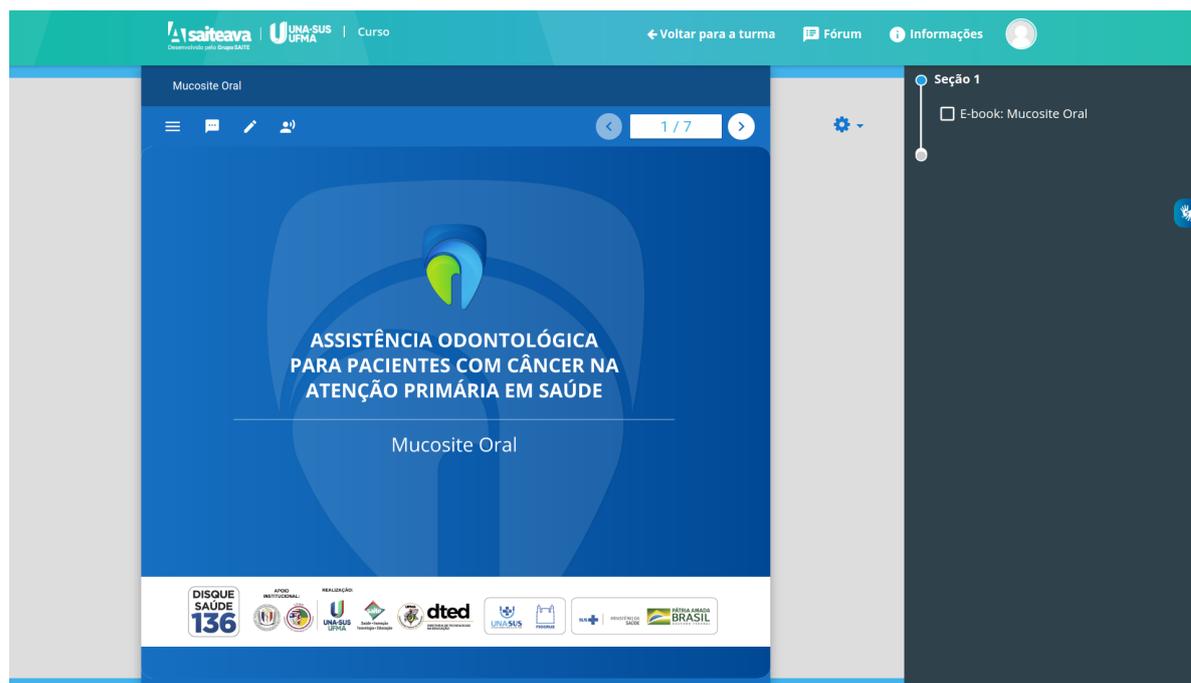


Figura 17 – Recurso externo acessado dentro da plataforma SAITE AVA. Fonte: (UNA-SUS/UFMA, 2021). Captura de tela criada pelo autor.

A integração LTI UNA-SUS/UFMA foi criada implementando uma camada de controle responsável por realizar a comunicação com o sistema LTIaaS e controlar o acesso aos recursos externos. Durante a configuração do sistema, ambos os endereços alvo, *toolUrl* e *toolDL*, descritos na Seção 4.1, foram apontados para a camada de controle, que se encarrega de exibir a tela de seleção de recursos ou redirecionar o usuário para o recurso externo adequado, após lançamentos LTI *Deep Linking* ou *Core*, respectivamente.

Comumente, no tipo de caso de uso Repositório de aplicações, nada impede que as aplicações acessem o sistema LTIaaS diretamente, porém, devido a natureza diversa dos recursos externos oferecidos pela UNA-SUS/UFMA, e a possibilidade da utilização de recursos que funcionam inteiramente do lado do cliente, foi necessária a criação de uma solução que permitisse aos recursos acesso ao sistema sem seguir os esquemas de autenticação baseados em chave de API, pois como especificado na Seção 4.3, requisições feitas no lado do cliente podem deixar a chave de API exposta no navegador, comprometendo a segurança do sistema.

Por esse motivo, a camada de controle implementa diversos *endpoints* que podem ser utilizados pelos recursos externos para acessar as funcionalidades do sistema LTIaaS passando somente o *token ltik*, descrito na Seção 3.1. Em cada um desses *endpoints*, o *token ltik* é utilizado para implementar funcionalidades que geram dados específicos a um único contexto sem que esse contexto precise ser conhecido diretamente pelos recursos externos. A camada de controle então se encarrega de realizar as chamadas aos *endpoints* do sistema LTIaaS e retornar a resposta para o recurso externo.

Um bom exemplo de *endpoint* implementado pela camada de controle é

o *endpoint* utilizado para o registro de eventos, acessível para qualquer recurso externo. Através deste, é possível gerar qualquer tipo de evento que se deseje salvar no banco de dados. O seu acesso é realizado através de uma simples requisição *HTTP POST* para a rota */api/events*, contendo um objeto com o campo *key* utilizado para descrever o tipo do evento. A Listagem 21 mostra um exemplo de objeto passado para o *endpoint* de registro de eventos.

```
1 {  
2   key: 'click_menu'  
3 }
```

Listagem 21 – Exemplo de objeto gerador de evento

O *endpoint* de registro de eventos, ao receber o objeto descrito acima, utiliza o *token Itik* passado para realizar uma chamada para o *endpoint LTIaaS ID Token*, descrito na Seção 4.4.1, e então utiliza as informações retornadas para incrementar o objeto recebido com os dados específicos do contexto em questão antes de salvá-lo no banco de dados. A Listagem 22 mostra um exemplo de objeto de evento contendo informações de contexto.

```
1 {  
2   key: 'click_menu',  
3   platform: 'https://saiteava.org',  
4   context: '8',  
5   resource: '25',  
6   user: '2',  
7 }
```

Listagem 22 – Exemplo de objeto gerador de evento com informações de contexto

Utilizando o sistema LTIaaS da maneira descrita acima, foi possível criar recursos externos para a plataforma SAITE AVA que podem ter suas funcionalidades facilmente expandidas através de *endpoints* que utilizam os serviços disponibilizados pelo protocolo LTI.

5.3 Moodle como Ferramenta LTI 1.3

O LMS Moodle pode ser utilizado nativamente como uma Ferramenta LTI 1.1, permitindo que cursos e atividades possam ser publicadas como Ferramentas LTI individuais. Devido às limitações do protocolo LTI 1.1, para serem utilizadas, cada Ferramenta precisa ser registrada manualmente em Plataformas compatíveis, o que pode consumir muito tempo de administradores.

Buscando melhorar o uso do Moodle como Ferramenta LTI, foi desenvolvido um *plugin* chamado *ltiadv* que quando utilizado juntamente ao sistema LTIaaS

transforma o Moodle em uma Ferramenta LTI 1.3, podendo usufruir dos serviços atualizados do protocolo, em especial o serviço de *Deep Linking*, descrito na Seção 2.3.2.

Nesta seção, a Ferramenta LTI Blending Education será apresentada e seu funcionamento será utilizado para descrever o funcionamento do plugin Moodle *ltiadv*. A Figura 18 ilustra a arquitetura do tipo de caso de uso Moodle como Ferramenta LTI 1.3.

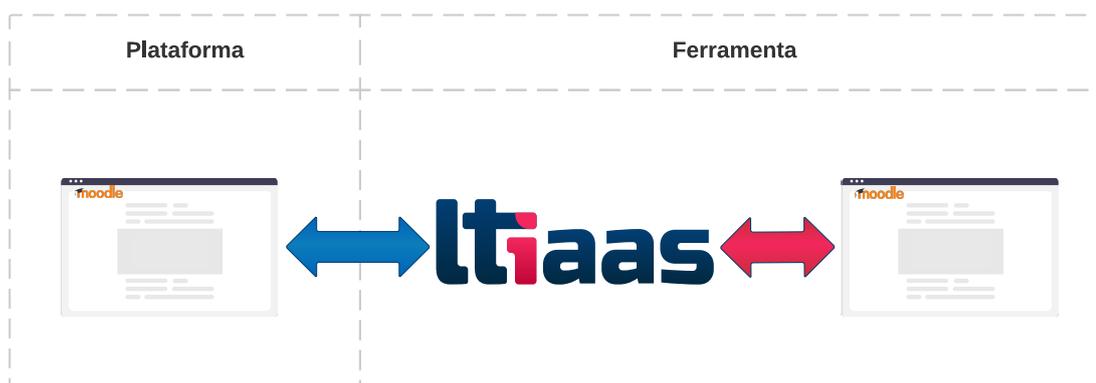


Figura 18 – Arquitetura do tipo de caso de uso Moodle como Ferramenta LTI 1.3. Fonte: Autor do trabalho

5.3.1 Blending Education

Blending Education ([BLENDING EDUCATION, LLC. U.S.A, 2021](#)) é uma instituição que oferece pacotes de recursos educacionais em diversas áreas de conteúdo e níveis de ensino, através de uma instância do LMS Moodle chamada de Microlearning Library App. Utilizando o *plugin ltiadv*, recursos educacionais criados nesta plataforma podem ser publicados como Ferramentas LTI, como visto na Figura 19.

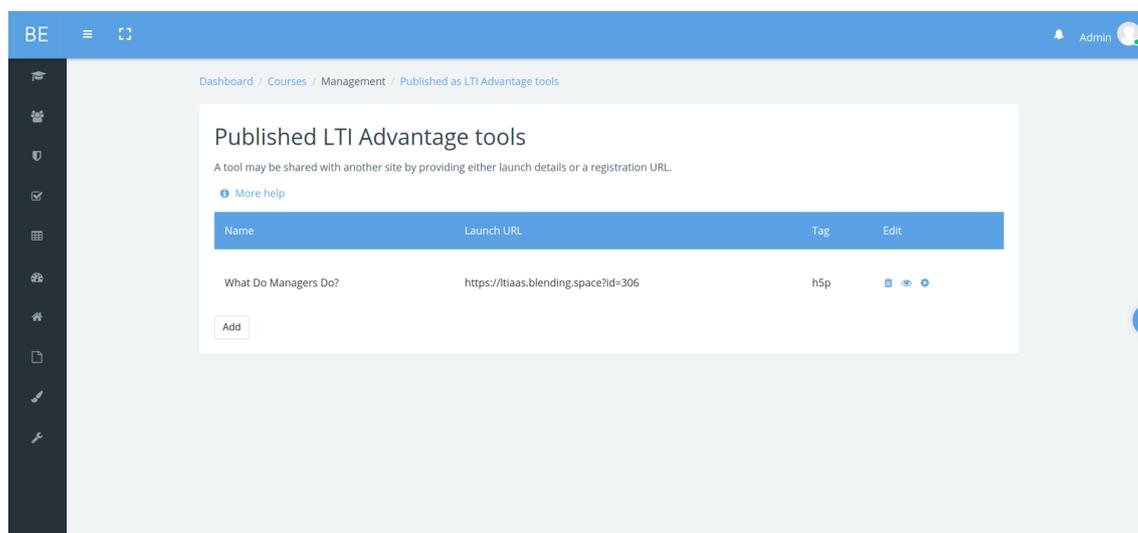


Figura 19 – Recurso publicado como Ferramenta LTI 1.3. Fonte: ([BLENDING EDUCATION, LLC. U.S.A, 2021](#)). Captura de tela criada pelo autor.

Utilizando o Moodle como Ferramenta LTI 1.1, os recursos publicados tinham que ser registrados e acessados individualmente em uma Plataforma, por outro lado, através do *plugin ltiadv*, estes recursos podem ser acessados através de uma única Ferramenta centralizadora que fornece uma tela de seleção, mostrada na Figura 20, através da qual um recurso pode ser selecionado e então acessado a partir da Plataforma.

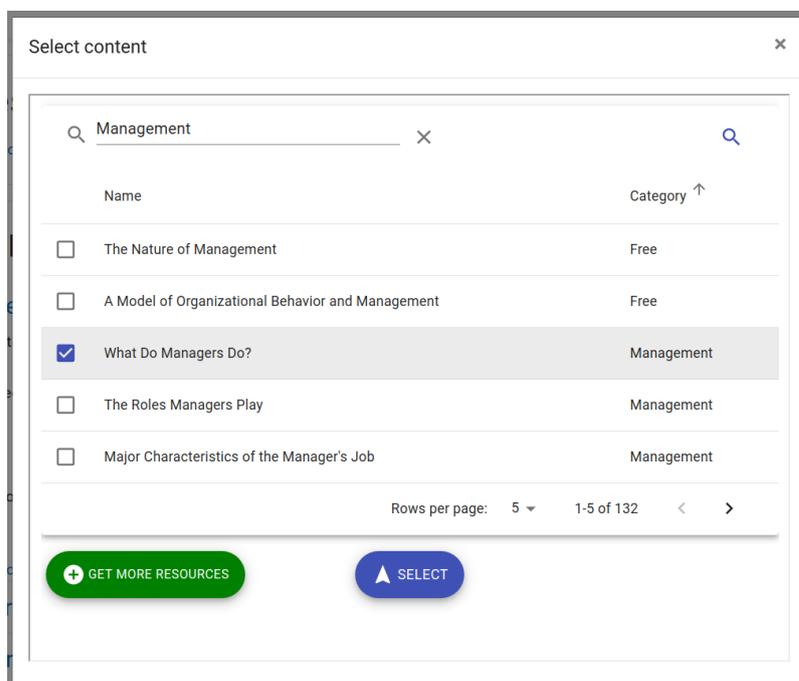


Figura 20 – Tela de seleção de recursos publicados como Ferramenta LTI 1.3. Fonte: (BLENDING EDUCATION, LLC. U.S.A, 2021). Captura de tela criada pelo autor.

Como pode ser visto na Figura 21, ao acessarmos um recurso através de uma Plataforma, a interface do Moodle agindo como Ferramenta LTI é simplificada para mostrar somente a área do recurso em si.

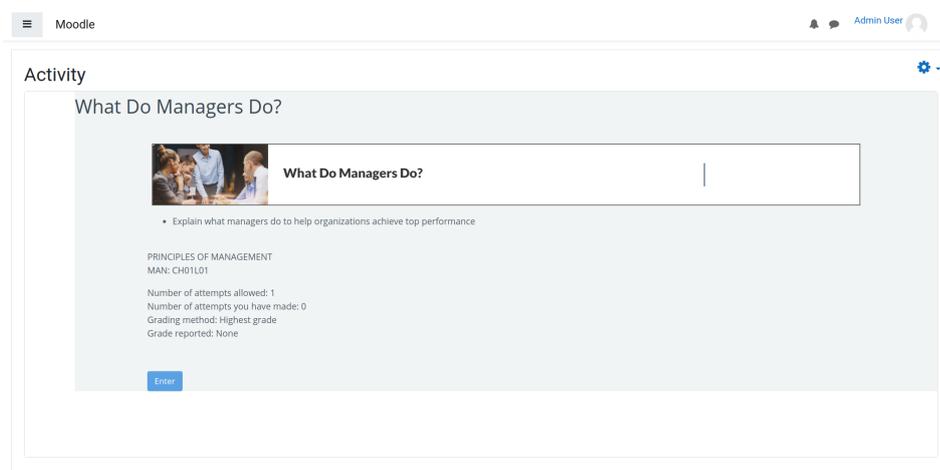


Figura 21 – Recurso acessado como Ferramenta LTI 1.3. Fonte: (BLENDING EDUCATION, LLC. U.S.A, 2021). Captura de tela criada pelo autor.

O usuário acessando o recurso é automaticamente cadastrado no Moodle agindo como Ferramenta LTI, e o seu papel no contexto do acesso também sendo transferido, permitindo que professores possam acessar funcionalidades específicas dos recursos disponibilizados, como manipular o número permitido de tentativas de envio.

O *plugin ltiadv* se encarrega de realizar a comunicação com o sistema LTIaaS, recebendo lançamentos LTI através de uma rota `/enrol/ltiadv/tool.php` configurada como o endereço alvo *toolUrl*, como especificado na Seção 4.1. Essa rota recebe um parâmetro de URL *id*, contendo o identificador do recurso publicado que deve ser mostrado.

Para flexibilizar a aparência e funcionamento da tela de seleção de recursos, foi utilizada uma pequena aplicação externa, responsável por receber os lançamentos LTI *Deep Linking* e concluir a seleção de recursos através do *endpoint LTIaaS Deep Linking*, descrito na Seção 4.4.4. O acesso à esta aplicação tem seu acesso configurado através do endereço alvo *toolDI*. Informações sobre recursos publicados são obtidas através de uma rota `/enrol/ltiadv/tools.php`, fornecida pelo *plugin ltiadv*.

O sistema de *Single Sign-On* (SSO) foi implementado utilizando o *endpoint LTIaaS ID Token*, descrito na Seção 4.4.1, e consiste em replicar a conta do usuário da Plataforma dentro do Moodle agindo como Ferramenta LTI.

Por fim, notas geradas pelos recursos publicados são repassadas para a Plataforma através do *endpoint LTIaaS Line Items*, descrito na Seção 4.4.3.

Através do sistema LTIaaS, juntamente com o *plugin ltiadv*, foi possível permitir que a plataforma Moodle Microlearning Library App funcione como uma Ferramenta LTI 1.3, e, como descrito acima, utilize os serviços atualizados do protocolo para facilitar a utilização dos recursos educacionais ofertados.

6 Conclusão

O sistema LTIIaaS foi implementado com o objetivo de abstrair a complexidade do protocolo LTI, fornecendo uma solução capaz não somente de criar novas Ferramentas LTI facilmente, mas também de possibilitar a adaptação de aplicações já consolidadas em Ferramentas LTI, de uma maneira desacoplada e agnóstica de linguagem. Os casos de uso apresentados no Capítulo 5 demonstram a capacidade do sistema de suprir as necessidades de diferentes tipos de aplicações e criar integrações LTI complexas através das funcionalidades definidas.

A integração LTI da plataforma Powtoon, descrita na Seção 5.1.1, é a mais completa entre os casos de uso apresentados. A forma como powtoons podem ser utilizados como objetos de aprendizagem reitera a capacidade do protocolo LTI de diversificar o processo de aprendizagem em plataformas de ensino. E sua implementação demonstra como os pontos principais de integração, como SSO e envio de notas, podem ter sua complexidade abstraída através do sistema LTIIaaS.

O caso de uso da plataforma CNH Cursos e seu sistema de biometria, descrito na seção 5.1.2, foi incluído para demonstrar a gama de funcionalidades que podem ser implementadas através dos serviços disponibilizados pelo protocolo LTI, indo além de simples recursos educacionais.

O caso de uso da plataforma SAITE AVA, descrita na Seção 5.2.1, demonstra a facilidade na integração de múltiplos recursos educacionais através de uma única Ferramenta LTI, e como todos esses recursos podem ter acesso às funcionalidades do protocolo utilizando o sistema LTIIaaS.

Por fim, o caso de uso da plataforma Blending Education, demonstra a capacidade do sistema LTIIaaS de ser utilizado com aplicações complexas, como o LMS Moodle, de uma maneira desacoplada, tendo em vista que a integração é inteiramente realizada através do sistema de plugins do LMS.

Utilizando o sistema LTIIaaS, o processo de adaptação e criação de recursos de aprendizagem LTI pode ser facilitado, potencialmente aumentando a adesão ao protocolo e contribuindo para uma maior diversificação no ambiente da educação a distância.

6.1 Contribuições

A biblioteca Ltjjs, contribuição do presente trabalho, conseguiu as certificações IMS LTI Advantage Complete ([IMS Global Learning Consortium Inc., 2020b](#)), sendo o única biblioteca pública à receber a certificação até então, e IMS TrustEd Apps™ Seal ([IMS Global Learning Consortium Inc., 2021](#)). O projeto tem cerca de 5 mil downloads por mês no repositório de pacotes NPM.

Foi apresentada uma palestra "Ltjjs, um pacote *open-source* para NodeJS que busca facilitar ao máximo a criação de provedores de ferramentas LTI" no evento

Moodle Moot Brasil em 2019.

Como parte do desenvolvimento da biblioteca Ltjjs, foi implementado o serviço *Dynamic Registration* no LMS Moodle.

O sistema LTaaS, contribuição central deste trabalho, está sendo utilizado por diversas empresas e instituições nacionais e internacionais. Alguns exemplos notáveis são Powtoon¹, DawnSign Press², Dynos.io³, WGU Labs⁴, CPM Education Program⁵ e a Pontifícia Universidade Católica de Minas Gerais⁶. A plataforma da empresa Dynos.io também conseguiu a certificação IMS LTI Advantage Complete (IMS Global Learning Consortium Inc., 2020a).

6.2 Trabalhos futuros

Como dito na Seção 4.1, o sistema LTaaS é distribuído em um formato *Software as a Product*, porém, o sistema está sendo atualmente adaptado para aderir ao modelo *Software as a Service*.

O sistema LTaaS também está sendo adaptado para ser utilizado na criação de Plataformas de ensino, buscando facilitar a adoção ao protocolo LTI 1.3 por parte de LMSs.

¹ <https://powtoon.com/>

² <https://www.dawnsign.com/>

³ <https://www.dynos.io/>

⁴ <https://wgulabs.org/>

⁵ <https://cpm.org/>

⁶ <https://www.pucminas.br/>

Referências

BLENDING EDUCATION, LLC. U.S.A. *Blending Education, We help educators with blended learning*. 2021. Disponível em: <<https://blending.education/about/>>. Acesso em: 2021-04-25. Citado 3 vezes nas páginas 9, 48 e 49.

HARDT, E. D. *The OAuth 2.0 Authorization Framework*. 2012. Disponível em: <<https://tools.ietf.org/html/rfc6749>>. Acesso em: 2021-04-24. Citado na página 18.

IMS Global Learning Consortium Inc. *IMS Global Learning Tools Interoperability®, howpublished = Disponível em: <<https://www.imsglobal.org/activity/learning-tools-interoperability>>, year = 2019, note = Acesso em: 2021-04-25. Citado 2 vezes nas páginas 9 e 17.*

IMS Global Learning Consortium Inc. *Dynos.io - IMS LTI Advantage Complete Certification*. 2020. Disponível em: <<https://site.imsglobal.org/certifications/greenos-llc/dynosio/>>. Acesso em: 2021-04-25. Citado na página 52.

IMS Global Learning Consortium Inc. *Ltijs - IMS LTI Advantage Complete Certification*. 2020. Disponível em: <<https://site.imsglobal.org/certifications/coursekey/ltijs/>>. Acesso em: 2021-04-25. Citado na página 51.

IMS Global Learning Consortium Inc. *Ltijs - IMS TrustEd Apps™ Seal*. 2021. Disponível em: <<https://site.imsglobal.org/certifications/coursekey/ltijs/>>. Acesso em: 2021-04-25. Citado na página 51.

Instructure, Inc. *Canvas LTI documentation*. 2019. Disponível em: <https://canvas.instructure.com/doc/api/file.tools_intro.html>. Acesso em: 2021-04-24. Citado na página 38.

JONES, M. *JSON Web Algorithms (JWA)*. 2015. Disponível em: <<https://tools.ietf.org/html/rfc7518>>. Acesso em: 2021-04-24. Citado na página 16.

JONES, M.; BRADLEY, J.; SAKIMURA, N. *JSON Web Signature (JWS)*. 2015. Disponível em: <<https://tools.ietf.org/html/rfc7515>>. Acesso em: 2021-04-24. Citado na página 16.

JONES, M.; BRADLEY, J.; SAKIMURA, N. *JSON Web Token (JWT)*. 2015. Disponível em: <<https://tools.ietf.org/html/rfc7519>>. Acesso em: 2021-04-24. Citado na página 16.

Mobilidade Segura. *Mobilidade Segura*. 2021. Disponível em: <<https://mobilidadesegura.com.br>>. Acesso em: 2021-04-24. Citado 3 vezes nas páginas 9, 43 e 44.

Powtoon Ltd. *Powtoon, the visual communication platform*. 2021. Disponível em: <<https://www.powtoon.com/>>. Acesso em: 2021-04-24. Citado 5 vezes nas páginas 9, 38, 39, 40 e 41.

- SAKIMURA, N.; BRADLEY, J.; JONES, M.; MEDEIROS, B.; MORTIMORE, C. *OpenID Connect Core 1.0 incorporating errata set 1*. 2014. Disponível em: <http://openid.net/specs/openid-connect-core-1_0.html>. Acesso em: 2021-04-24. Citado na página 18.
- Serpro. *Datavalid*. 2021. Disponível em: <<https://www.loja.serpro.gov.br/datavalid?lang=en>>. Acesso em: 2021-04-24. Citado na página 43.
- SMYTHE, C.; VERVOORT, C.; MCKELL, M.; MILLS, N. *IMS Global Learning Tools Interoperability® Names and Role Provisioning Services v2.0*. 2019. Disponível em: <<https://www.imsglobal.org/spec/lti-nrps/v2p0>>. Acesso em: 2021-04-24. Citado na página 22.
- SMYTHE, C.; VERVOORT, C.; MCKELL, M.; MILLS, N. *IMS Global Security Framework v1.0*. 2019. Disponível em: <<https://www.imsglobal.org/spec/security/v1p0/>>. Acesso em: 2021-04-24. Citado na página 18.
- UNA-SUS. *Universidade aberta do SUS*. 2021. Disponível em: <<https://www.unasus.gov.br/>>. Acesso em: 2021-04-25. Citado na página 45.
- UNA-SUS/UFMA. *Universidade Aberta do SUS - Universidade Federal do Maranhão*. 2021. Disponível em: <<https://www.unasus.ufma.br/>>. Acesso em: 2021-04-25. Citado 3 vezes nas páginas 9, 45 e 46.
- UNESCO. *National education responses to COVID-19: summary report of UNESCO's online survey*. 2020. Disponível em: <<https://unesdoc.unesco.org/ark:/48223/pf0000373322>>. Acesso em: 2021-04-27. Citado na página 14.
- VERVOORT, C.; MILLS, N. *IMS Global Learning Tools Interoperability® Core Specification v1.3*. 2019. Disponível em: <<https://www.imsglobal.org/spec/lti/v1p3/>>. Acesso em: 2021-04-24. Citado 3 vezes nas páginas 14, 17 e 19.
- VERVOORT, C.; PRESTON, E. *IMS Global Learning Tools Interoperability® Deep Linking 2.0*. 2019. Disponível em: <<https://www.imsglobal.org/spec/lti-dl/v2p0/>>. Acesso em: 2021-04-24. Citado na página 21.
- VERVOORT, C.; PRESTON, E.; MCKELL, M.; RISSLER, J. *IMS Global Learning Tools Interoperability® Assignment and Grade Services*. 2019. Disponível em: <<https://www.imsglobal.org/spec/lti-ags/v2p0/>>. Acesso em: 2021-04-24. Citado na página 22.

Apêndices

APÊNDICE A – Listagens referentes aos endpoints citados na Seção 4.4

```
1 // Monta URL da requisição
2 const url = LTIAAS_URL + '/api/memberships'
3 // Monta parâmetros de URL
4 const query = {
5   role: 'Learner',
6   limit: 10
7 }
8 // Realiza a requisição
9 const response = await request.get(url,
10   { searchParams: query, headers: headers }).json()
11 return response.members
```

Listagem 23 – Exemplo de código de acesso ao *endpoint Memberships*

```
1 {
2   id : 'https://lms.example.com/sections/2/memberships',
3   context: {
4     id: '2923-abc',
5     label: 'CPS 435'
6   },
7   next: 'https://lms.example.com/memberships/2/pages/2',
8   members : [
9     {
10      status : 'Active',
11      name: 'Jane Q. Public',
12      given_name : 'Jane',
13      family_name : 'Doe',
14      email: 'jane@platform.example.edu',
15      user_id : '0ae836b9-7fc9-4060-006f-27b2066ac545',
16      roles: [
17        'Instructor'
18      ]
19    }
20 ]
21 }
```

Listagem 24 – Exemplo de estrutura do objeto retornado pelo *endpoint Memberships*

```
1 // Monta URL da requisição
2 const url = LTIAAS_URL + '/api/lineitems'
3 // Monta parâmetros de URL
4 const query = {
5   tag: 'grade_line',
6   limit: 1,
7   resourceId: 'a-9334df-33'
8 }
9 // Realiza a requisição
10 const response = await request.get(url,
11   { searchParams: query, headers: headers }).json()
12 return response.lineItems
```

Listagem 25 – Exemplo de código de recuperação de linhas de nota

```
1 {
2   next: 'https://lms.example.com/sections/lineitems/2/pages/2',
3   first: 'https://lms.example.com/sections/lineitems/2/pages/1',
4   last: 'https://lms.example.com/sections/lineitems/2/pages/3',
5   lineItems: [
6     {
7       id: 'https://lms.example.com/2/lineitems/1',
8       scoreMaximum: 60,
9       label: 'Grade Line',
10      resourceId: 'a-9334df-33',
11      tag: 'grade',
12      resourceLinkId: '1g3k4dlk49fk',
13      endDateTime: '2018-04-06T22:05:03Z'
14    }
15  ]
16 }
```

Listagem 26 – Exemplo de estrutura do objeto retornado pelo endpoint *Line Items*

```
1 // Monta URL da requisição
2 const url = LTIAAS_URL + '/api/lineitems'
3 // Monta corpo da requisição
4 const lineitem = {
5   scoreMaximum: 60,
6   label: 'Grade line',
7   resourceId: 'quiz-231',
8   tag: 'grade',
9   startDateTime: '2021-03-06T20:05:02Z',
10  endDateTime: '2021-04-06T22:05:03Z'
11 }
12 // Realiza a requisição
13 const response = await request.post(url,
14   { json: lineitem, headers: headers }).json()
15 return response
```

Listagem 27 – Exemplo de código de criação de linha de nota

```
1 // Prepara ID da linha de nota
2 const id = 'https://lms.example.com/2/lineitems/1'
3 const cleanId = encodeURIComponent(id)
4 // Monta URL da requisição
5 const url = LTIAAS_URL + '/api/lineitems/' + cleanId
6 // Monta corpo da requisição
7 const lineitem = {
8   scoreMaximum: 60,
9   label: 'Grade line',
10  resourceId: 'quiz-231',
11  tag: 'grade',
12  startDateTime: '2021-03-06T20:05:02Z',
13  endDateTime: '2021-04-06T22:05:03Z'
14 }
15
16 // Realiza a requisição
17 const response = await request.put(url,
18   { json: lineitem, headers: headers }).json()
19 return response
```

Listagem 28 – Exemplo de código de atualização de linha de nota

```
1 // Prepara ID da linha de nota
2 const id = 'https://lms.example.com/2/lineitems/1'
3 const cleanId = encodeURIComponent(id)
4 // Monta URL da requisição
5 const url = LTIAAS_URL + '/api/lineitems/' + cleanId + '/scores'
6 // Monta corpo da requisição
7 const score = {
8   userId: '5323497',
9   activityProgress: 'Completed',
10  gradingProgress: 'FullyGraded',
11  scoreGiven: 83,
12  comment: 'This is exceptional work.',
13 }
14 // Realiza a requisição
15 const response = await request.post(url,
16   { json: score, headers: headers }).json()
17 return response
```

Listagem 29 – Exemplo de código de envio de nota

```
1 // Prepara ID da linha de nota
2 const id = 'https://lms.example.com/2/lineitems/1'
3 const cleanId = encodeURIComponent(id)
4 // Monta URL da requisição
5 const url = LTIAAS_URL + '/api/lineitems/' + cleanId + '/scores'
6 // Monta parâmetros de URL
7 const query = {
8   userId: 2,
9   limit: 10
10 }
11 // Realiza a requisição
12 const response = await request.get(url,
13   { searchParams: query, headers: headers }).json()
14 return response.scores
```

Listagem 30 – Exemplo de código de recuperação de notas

```
1  {
2    next: 'https://lms.example.com/sections/2/scores/pages/2',
3    first: 'https://lms.example.com/sections/2/scores/pages/1',
4    last: 'https://lms.example.com/sections/2/scores/pages/3',
5    scores: [
6      {
7        id: 'https://lms.example.com/2/lineitems/1/results?u=2',
8        userId: '2',
9        resultScore: 100,
10       resultMaximum: 100,
11       timestamp: '2020-06-02T10:51:08-03:00'
12     }
13   ]
14 }
```

Listagem 31 – Exemplo de objeto retornado pela sub-rotina *scores* do *endpoint Line Items*

```
1  // Monta URL da requisição
2  const url = LTIAAS_URL + '/api/deeplinking/form'
3  // Monta corpo da requisição
4  const link = {
5    contentItems: [{
6      type: 'ltiResourceLink',
7      url: 'https://tool.com?resourceid=123456',
8      title: 'Resource'
9    }]
10 }
11 // Realiza a requisição
12 const response = await request.post(url,
13   { json: link, headers: headers }).json()
14 return response.form
```

Listagem 32 – Exemplo de código de geração de formulário *endpoint Deep Linking*

```
1 {
2   form: `
```

Listagem 33 – Exemplo de objeto retornado pelo endpoint *Deep Linking*

```
1 // Monta URL da requisição
2 const url = LTIAAS_URL + '/api/platforms'
3 // Monta corpo da requisição
4 const platform = {
5   url: 'https://lms.example.com',
6   clientId: '10000000000001',
7   name: 'Platform 1',
8   authenticationEndpoint: 'https://lms.example.com/auth',
9   accesstokenEndpoint: 'https://lms.example.com/token',
10  authConfig: {
11    key: 'https://lms.example.com/jwks',
12    method: 'JWK_SET'
13  }
14 }
15 // Realiza a requisição
16 const response = await request.post(url,
17   { json: platform, headers: headers }).json()
18 return response
```

Listagem 34 – Exemplo de código de criação de registro de Plataforma

```
1 // Monta URL da requisição
2 const url = LTIAAS_URL + '/api/platforms'
3 // Realiza a requisição
4 const response = await request.get(url,
5   { headers: headers }).json()
6 return response.platforms
```

Listagem 35 – Exemplo de código de recuperação de registros de Plataforma

```
1 {
2   platforms: [
3     {
4       id: 'f33d6c450b7632c1ff87586c4f150eee',
5       url: 'https://lms.example.com',
6       clientId: '10000000000001',
7       name: 'Platform 1',
8       authenticationEndpoint: 'https://lms.example.com/auth',
9       accesstokenEndpoint: 'https://lms.example.com/token',
10      authConfig: {
11        key: 'https://lms.example.com/jwks',
12        method: 'JWK_SET'
13      },
14      publicKey: '-----BEGIN PUBLIC KEY-----...'
15    }
16  ]
17 }
```

Listagem 36 – Exemplo de estrutura do objeto retornado pelo *endpoint Platforms*

```
1 // Monta URL da requisição
2 const id = 'f33d6c450b7632c1ff87586c4f150eee'
3 const url = LTIAAS_URL + '/api/platforms/' + id
4 // Monta corpo da requisição
5 const platform = {
6   name: 'Platform 2',
7   authenticationEndpoint: 'https://lms.example.com/auth2',
8   accesstokenEndpoint: 'https://lms.example.com/token2',
9   authConfig: {
10    key: 'https://lms.example.com/jwks2',
11    method: 'JWK_SET'
12  }
13 }
14 // Realiza a requisição
15 const response = await request.put(url,
16   { json: platform, headers: headers }).json()
17 return response
```

Listagem 37 – Exemplo de código de atualização de um registro de Plataforma