



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Implantando um provedor de serviços educacionais interoperáveis segundo o padrão IMS-LTI

Vitor Pereira Tavares da Costa

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientadora
Prof.a Dr.a Germana Menezes da Nobrega

Brasília
2021



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Implantando um provedor de serviços educacionais interoperáveis segundo o padrão IMS-LTI

Vitor Pereira Tavares da Costa

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Prof.a Dr.a Germana Menezes da Nobrega (Orientadora)
CIC/UnB

Prof. Dr. Fernando W. Cruz Prof. Dr. Marcelo Antonio Marotta
FGA/UnB IE/CIC/UnB

Prof. Dr. Wilson Henrique Veneziano
Coordenador do Curso de Computação — Licenciatura

Brasília, 21 de maio de 2021

Ficha catalográfica elaborada automaticamente,
com os dados fornecidos pelo(a) autor(a)

CC837i Costa, Vitor Pereira Tavares da
Implantando um provedor de serviços educacionais
interoperáveis segundo o padrão IMS-LTI / Vitor Pereira
Tavares da Costa; orientador Germana Menezes da Nobrega. --
Brasília, 2021.
56 p.

Monografia (Graduação - Licenciatura em Computação) --
Universidade de Brasília, 2021.

1. Ecossistemas de aprendizado. 2. Serviços educacionais.
3. Infraestrutura. 4. LTI. I. Nobrega, Germana Menezes da,
orient. II. Título.

Dedicatória

Dedico este trabalho a minha mãe que, trabalhando a vida inteira na área da educação, me ensinou a escutar, a pesquisar, a respeitar, a cuidar e o mais importante, me ensinou que o aprendizado é infinito e que a gente pode sempre estar buscando mais conhecimento. E ao meu pai, meu modelo de dedicação, de corretude e de paciência, e que sem dúvida nenhuma é responsável pela minha paixão por tecnologia.

Agradecimentos

Agradeço aos meus pais, aos meus amigos e a minha namorada por terem dado o espaço para eu lidar com os desafios da minha jornada que culminou neste trabalho, mas que nunca deixaram eu desistir. Agradeço a Professora Doutora Germana Menezes da Nóbrega por propor um trabalho em que eu sabia navegar, mas que apresentou desafios e resultou em descobertas que eu levarei comigo para a vida, e por acompanhar e cobrar sempre que eu precisei. E agradeço a todos os professores que me formaram até aqui, porque ensinar não é fácil não.

Resumo

Este trabalho apresenta um estudo sobre requisitos necessários para a construção de um ecossistema educacional digital, mapeando o panorama atual de implementação dessas plataformas buscando características, padrões, ferramentas e práticas que resultaram em uma implementação de infraestrutura de provedor de serviços. Essa estrutura deve permitir que outros trabalhos sejam desenvolvidos com o propósito final da construção do ecossistema *smartUnB.ECOS*. O padrão LTI, uma plataforma com o propósito de integrar serviços educacionais, é investigado como a plataforma de integração que agregará e cuidará da troca de dados entre as aplicações educacionais do ecossistema. A arquitetura da infraestrutura que permitirá que os serviços educacionais sejam desenvolvidos é desenhada levando em conta as necessidades dos desenvolvedores, bem como os requisitos do padrão para permitir a integração. Com essa infraestrutura construída novos trabalhos dentro do projeto *smartUnB.ECOS* poderão ser desenvolvidos, agregando serviços educacionais e permitindo pesquisas nesse contexto.

Palavras-chave: Ecossistemas de Aprendizado, Serviços Educacionais, Infraestrutura, LTI

Abstract

This research presents a study about some of the requirements to develop a Digital Educational Ecosystem, mapping the current landscape on implementation of learning platforms, researching standards, details, tools and common practices that resulted in the implementation of a server infrastructure in which future researchers will be deployed resulting in the ecosystem smartUnB.ECOS. The LTI, a platform that developed to integrate educational services, is examined as integration tool to aggregate and provide data transfer between the learning tools in the ecosystem. The infrastructure architecture that will allow the development of educational services is drawn considering the developers needs and the LTI requisites. From the infrastructure of smartUnB.ECOS project built in this research, new educational services can be developed, integrated and used to research in digital learning.

Keywords: Educational Ecosystems, Educational Services, Infrastructure, LTI

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Motivação e Justificativa	3
1.3	Questão da pesquisa	3
1.4	Objetivo	3
1.4.1	Objetivo Geral	3
1.4.2	Objetivos Específicos	3
1.5	Estrutura deste documento	4
2	Revisão de Literatura	5
2.1	Ecosistemas Educacionais Apoiados por Computador e Provimento de Ser- viços	6
2.2	Plataformas de integração para ecossistemas	8
2.3	Destaques acadêmicos na última década	9
3	Referencial Teórico para a Proposta	15
3.1	Panorama de provedores LTI certificados	15
3.1.1	O padrão LTI	15
3.1.2	A certificação LTI	16
3.2	Aparato tecnológico e metodológico para criação e manutenção de um pro- vedor LTI	17
3.2.1	Microserviços e Containers	17
3.2.2	Tecnologias para infraestrutura de orquestração de microserviços . . .	18
3.2.3	Protocolo de autorização	18
4	Proposta	20
4.1	O projeto	20
4.1.1	A escolha do LTI	20
4.1.2	O escalonamento	21
4.2	O servidor	22

4.3	Arquitetura da Infraestrutura	23
4.4	Implementação da Infraestrutura	24
4.4.1	Kubernetes com Microk8s	24
4.4.2	Repositório de Código GitLab	26
4.4.3	Controle de acesso	27
4.5	O desenvolvimento	28
4.6	A operação	28
4.6.1	A preparação	28
4.6.2	Viabilizando a execução de um serviço LTI pioneiro na equipe	30
5	Conclusão	32
5.1	Contribuições	32
5.2	Trabalhos futuros e perspectivas	33
	Referências	35
	Apêndice	38
A	Configuração e execução do VisCC no servidor smartUnB.ECOS	39
A.1	O <i>Backend</i>	39
A.2	O <i>Frontend</i>	42
A.3	Meio de acesso	44
A.4	A construção do orquestrador	45
A.5	A submissão para o GitLab do smartUnB.ECOS	46
A.6	Integração no Moodle Sandbox	49
B	Como começar um novo projeto	52
B.1	Criação do projeto	52

Lista de Figuras

1.1	Estrutura do <i>smartUnB.ECOS</i> [1]	2
2.1	Arquitetura Conceitual da Infraestrutura BROAD-ECOS.[2]	7
2.2	Linha temporal de trabalhos voltados para aplicação do padrão LTI	9
2.3	Exemplo de jogo integrado a plataforma usando LTI [3]	10
2.4	Troca de mensagens entre provedor e consumidor [4]	12
2.5	Aumento do número de acessos, do número de páginas acessadas e da duração por sessão durante a pandemia [5]	14
4.1	Diagramas das estruturas com o LRS acoplado ou isolado do AVA(LSM) [6].	21
4.2	Diagrama do papel do LTI e Caliper [7].	22
4.3	Diagrama de Caso de Uso da infraestrutura.	24
4.4	Arquitetura proposta.	25
4.5	Painel de controles do Kubernetes.	25
4.6	Tela de gestão de projeto do GitLab.	26
4.7	Árvore da estrutura da aplicação depois de preparada para ser executada como microsserviços.	29
4.8	Execução do <i>pipeline</i> de lançamento do aplicação.	30
4.9	VisCC em execução dentro do AVA Moodle.	31
4.10	Avaliação do consumo de recurso computacional dos microsserviços no painel de controle do Kubernetes.	31
A.1	Execução local do Backend.	42
A.2	Execução local do Frontend.	44
A.3	Construção do repositório.	47
A.4	Execução do <i>pipeline</i> de lançamento do aplicação.	48
A.5	Aplicação em execução no servidor.	49
A.6	Cadastro da aplicação com LTI no Moodle.	50
A.7	Aplicação integrada ao Moodle Sandbox.	50
B.1	Criação do projeto.	53

B.2	Página do projeto.	54
B.3	Projeto disponível.	56

Lista de Tabelas

2.1 Mapeamento das contribuições dos trabalhos citados [3, 8, 9, 10, 11, 4, 12, 13, 14, 5].	14
3.1 Distribuição de aplicações certificadas por tipo. Dados:[15]	16
3.2 Divisão dos membros da IMS com certificação. Dados:[16]	16
4.1 Especificações do servidor	23
5.1 Trabalhos previstos ou em andamento no projeto smartUnB.ECOS	33

Lista de Abreviaturas e Siglas

API Interface de Programação de Aplicativos (*Application Programming Interface*).

AVA Ambiente Virtual de Aprendizado.

CBL Aprendizado Baseado em Competência.

CI/CD Integração Contínua, Entrega Contínua, Lançamento Contínuo (*Continuous Integration, Continuous Delivery, Continuous Deployment*).

CIC Departamento de Ciência da Computação.

CMI5 Computer Managed Instruction 5.

CNCF Cloud Native Computing Foundation.

HTTP *HyperText Transfer Protocol*.

JSON *JavaScript Object Notation*.

LD Learning Design.

LRS *Learning Record Stores*.

LTI *Learning Tools Interoperability*.

MOOCS Cursos massivos abertos online (*Massive Open Online Courses*).

PCDM Personal Competence Domain Model.

PNLD Programa Nacional de Livros Didáticos.

RBAC Controle de acesso baseado em perfil (*Role-based access control*).

RNP Rede Nacional de Ensino e Pesquisa.

SBC Computador de placa única (*Single Board Computer*).

SOA *Arquitetura Orientada a Serviços*(Service Oriented Architecture).

xAPI Experience API.

Capítulo 1

Introdução

O aprendizado é fruto de experiências de um indivíduo e sendo esse aprendizado formal ou informal é impactado pelo meio em que a pessoa se insere. No contexto atual onde a sociedade está imersa em um meio tecnológico é importante mensurar o potencial da utilização desta para o aprendizado[17]. Ciente disso existe um grande esforço para a melhoria no contexto educacional utilizando ferramentas digitais, e a integração dessas resulta em um ecossistema educacional, plataformas de aprendizado que integra ferramentas com uma abordagem holística e que permite um desenvolvimento colaborativo e incremental, com contribuições de aplicações independentes[2].

1.1 Contextualização

Ecossistema é definido como o conjunto formado pelo ambiente e uma gama de seres que contribuem para um equilíbrio colaborativo [18]. O ambiente prove o necessário para a existência e manutenção dos seres, enquanto os seres são responsáveis por interagir com o ambiente e outros seres e por manter o equilíbrio entre eles. As mesmas características dos ecossistemas naturais podem ser encontradas em ecossistemas digitais. O ambiente é dado por plataformas, ferramentas, aplicações e infraestruturas. Já os seres são usuários e as próprias ferramentas consumidoras. As dinâmicas continuam, usuários produzem e consomem conteúdos, desenvolvem novas ferramentas, integram aplicações, novas plataformas nascem e são utilizadas, substituídas e atualizadas. Administradores proveem o equilíbrio, racionalizando recursos, dando e removendo permissões de acesso, instalando e removendo aplicações. Um ecossistema digital está em evolução constante.

Um ecossistema digital não é apenas um conjunto de aplicações em um lugar. A necessidade da integração, da comunicação entre as diversas aplicações, criadas por desenvolvedores diferentes é um desafio que resulta na existência de diversas soluções com características específicas[2].

O projeto *smartUnB.ECOS* propõe um estudo sobre os ecossistemas educacionais apoiados por computador através de diversos trabalhos integrando ferramentas de aprendizado com uma comunicação via redes sociais descentralizadas.[1] Nesse projeto foi escolhido o padrão LTI, um padrão aberto desenvolvido e mantido pela *IMS Global Learning Consortium*, como plataforma de integração. Esse padrão é proposto na arquitetura de ecossistema do artigo [19], que foi adaptado para a estrutura do *smartUnB.ECOS*. Na Figura 1.1 pode ser observada a estrutura inicial do *smartUnB.ECOS* onde existe uma camada de serviços exemplificada pelo AVA Moodle, pela rede social Friendica e pela plataforma Wiki local, Uma camada de integração com o padrão LTI de um lado e uma proposta de LRS em aberto do outro, e os usuários fazendo interface com a estrutura por meio de seus dispositivos.

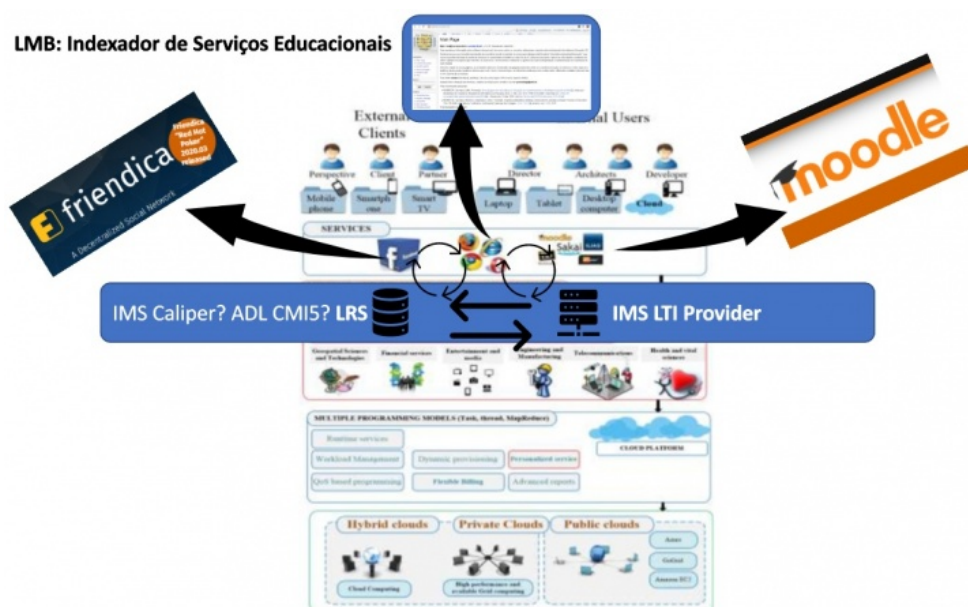


Fig. 3. Ecosystem overview.

Elementos do smartUnB.ECOS, inspirados em [Santamaria-Buitrago, Ballesteros-Ricaurte e Gonzalez-Amarillo, 2019]

Figura 1.1: Estrutura do *smartUnB.ECOS* [1] .

Com esse objetivo em mente foi proposta uma plataforma para que os alunos do CIC possam desenvolver ferramentas para o *smartUnB.ECOS*, como também possam utilizar tais ferramentas ao longo de sua formação. Uma vez escolhida a plataforma que será usada para integrar as aplicações do ecossistema, o próximo passo é desenvolver uma infraestrutura para prover os meios, ferramentas e acessos para que os desenvolvedores possam trabalhar de forma integrada e centralizada, mesmo que os projetos sejam individuais. Este trabalho faz parte do projeto do *smartUnB.ECOS* e apresenta um escopo limitado na abordagem das definições do projeto, outras complementações da estrutura serão desenvolvidas em trabalhos futuros.

1.2 Motivação e Justificativa

A construção de um ecossistema de software é um esforço coletivo. A infraestrutura do sistema provê a fundação onde o desenvolvimento das ferramentas e integrações acontecem e permitirá o lançamento, teste e integração das ferramentas educacionais desenvolvidas. Dentro do ambiente da universidade são desenvolvidos diversos projetos, ferramentas e pesquisas que poderiam ser utilizadas e não são pois não há continuidade no desenvolvimento depois da conclusão do aluno. Fornecer uma estrutura integrada, preparada para armazenar, disponibilizar, testar e integrar os projetos desenvolvidos permite outros desenvolvedores deem continuidade e manutenção e permite que os usuários tenham acesso a essas ferramentas desenvolvidas.

Ter ferramentas avulsas, separadas e dispersas pode gerar desperdício de esforços e desmotivar o usuário. A proposta de um padrão como o LTI é permitir que as ferramentas educacionais sejam integradas, interoperem umas com as outras, troquem informações e permitam que o usuário acesse toda gama de ferramentas através de um ponto de acesso.

1.3 Questão da pesquisa

O trabalho parte do princípio que fazendo parte de um projeto maior, o *smartUnB.ECOS*, com requisitos já definidos e com esses requisitos a questão da pesquisa pode ser enunciada como:

Como estruturar uma infraestrutura de computação preparada para auxiliar o desenvolvimento de serviços educacionais modulares utilizando o padrão LTI e viabilizar o provimento desses serviços?

1.4 Objetivo

1.4.1 Objetivo Geral

O objetivo do trabalho é preparar uma estrutura para que outros desenvolvedores tenham ferramentas em um servidor para trabalhar e integrar seus desenvolvimentos em um ecossistema educacional que será utilizado para outros estudos e pesquisas.

1.4.2 Objetivos Específicos

O objetivo geral do trabalho foi decomposto nos seguintes objetivos específicos:

- Investigar o padrão LTI como plataforma de integração para serviços educacionais.

- Determinar o conjunto de ferramentas necessárias para atender as necessidades de desenvolvedores de software e para atender os requisitos definidos pelo padrão LTI para que as ferramentas possam ser implementadas e operarem corretamente.
- Identificar e elencar os requisitos para atender demandas do padrão LTI, permitindo assim que as aplicações desenvolvidas possam ser integradas utilizando o padrão de uma forma organizada.
- Investigar e explicitar o funcionamento da certificação LTI e o que é necessário para conseguir a certificação.
- Definidas essas ferramentas, preparar o servidor para receber essas ferramentas, instalar as ferramentas, testar a estrutura e entregar o sistema.

1.5 Estrutura deste documento

Este capítulo apresentou uma introdução, o contexto em que este trabalho se insere, a motivação o desenvolvimento do mesmo, a questão levantada, e os objetivos do trabalho. No Capítulo 2 é apresentada a revisão de literatura, que abordou os principais trabalhos acadêmicos no contexto de ecossistemas educacionais digitais no Brasil, os trabalhos que adotam o padrão LTI como plataforma de integração e os destaques acadêmicos dessa área na última década. No Capítulo 3 são apresentados os pressupostos teóricos no campo do LTI e o aparato tecnológico que embasaram a proposta no capítulo seguinte. No Capítulo 4 é documentada a proposta do trabalho, as considerações e escolhas feitas na construção da infraestrutura e a execução da primeira aplicação na infraestrutura construída. No Capítulo 5 são mapeadas as conclusões do trabalho e os trabalhos que poderão acontecer neste mesmo contexto.

Capítulo 2

Revisão de Literatura

Neste capítulo serão apresentadas as pesquisas acadêmicas na área de ecossistemas educacionais digitais que servirão de fundamento para o desenvolvimento do trabalho. Na primeira seção são apresentadas as principais iniciativas nacionais no âmbito de ecossistemas educacionais e serviços educacionais, na segunda é investigado o papel das plataformas de integração e sua aplicação, e no terceiro é apresentado um panorama histórico das pesquisas sobre ecossistemas digitais e aplicação do padrão LTI na última década e é mapeado como essas pesquisas podem contribuir para este trabalho. Foi criado um glossário dos principais termos utilizados nos estudos avaliados.

Glossário

AVA: Ambiente Virtual de Aprendizado, também conhecido pela sigla em inglês LMS, são plataformas desenvolvidas para o acesso do aluno a conteúdos educacionais, podendo apresentar cursos, disciplinas, grupos de pesquisa, avaliações, entre outros serviços. O Moodle, Blackboard, Canvas e Sakai são alguns dos exemplos de AVAs que podem ser encontrados em serviço hoje.

Padrão LTI: É um padrão de integração de serviços educacionais que permite integrar serviços externos a AVAs que seguem este padrão. É o padrão proposto para integrar o projeto *smartUnB.ECOS* e é investigado neste trabalho.

Provedores e Consumidores: Dentro do padrão LTI é a classificação dada para serviços dentro do padrão. Uma aplicação Consumidora acessa os dados de outra aplicação, e por isso na maior parte das vezes o consumidor será um AVA, mas existem exceções, como é mostrado em trabalhos pesquisados.

Blended Learning: Ensino misto ou Ensino híbrido, é um modelo de ensino que com-

bina aulas presenciais e ensino digital online. É proposto como uma forma de alcançar as principais vantagens dos sistemas de ensino presencial e a distância.

Gamificação: É o modelo proposto para engajar o usuário ao conteúdo através de construções lúdicas e interativas. Pode ser presencial ou digital.

Tutores cognitivos: São aplicações construídas para interagir com o aluno durante o processo de aprendizado a fim de mapear o aprendizado, construindo um modelo cognitivo do aluno que poderá ser utilizado para direcionar o ensino ajudando a melhorar o aprendizado.

2.1 Ecossistemas Educacionais Apoiados por Computador e Provedor de Serviços

Existem algumas iniciativas na construção e implementação de ecossistemas educacionais no âmbito nacional. Essas iniciativas apresentam uma maturação do entendimento sobre a necessidade e complexidade desse tipo de ferramenta, motivada, além dos fatores técnicos e metodológicos, pelo impacto dos aspectos culturais e sociológicos [2]. Nesse trabalho são apresentadas necessidades de ecossistemas educacionais que ampliam a noção sobre o impacto dessas ferramentas e a importância da correta implementação delas com impacto no aprendizado. O resultado do trabalho [2] foi a implementação da plataforma BROAD-ECOS, um ecossistema baseado em serviços educacionais e apresentada uma proposta de uma estrutura de integração de serviços a um AVA. A proposta resulta em uma arquitetura que permite a integração e reuso de ferramentas educacionais.

Do ponto de vista de infraestrutura Veiga [2] destaca os requisitos funcionais de não funcionais considerados fundamentais para a construção de um ecossistema educacional. Os requisitos funcionais são:

- Integração de serviços externos;
- Estabelecimento de um modelo comum para o domínio educacional;
- Controle de autorização;
- Uso de metadados educacionais;
- Registro e recuperação de experiências de aprendizagem ;

- Suporte à existência de um canal de distribuição.

E os requisitos não funcionais:

- Extensibilidade;
- Poucas restrições na criação de serviços compatíveis.

Com esses requisitos em mente foi construída a arquitetura conceitual, que pode ser vista na Figura 2.1.

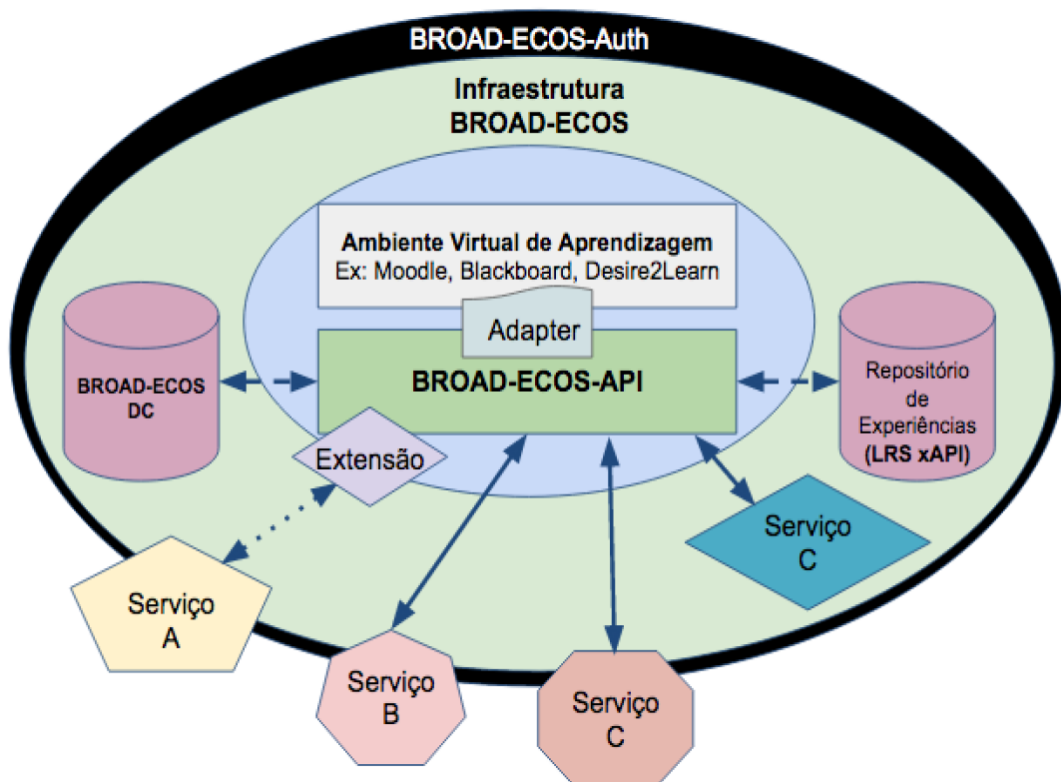


Figura 2.1: Arquitetura Conceitual da Infraestrutura BROAD-ECOS.[2] .

Outra iniciativa nacional é o SOLAR [20], um projeto de *blended learning*, ou ensino misto. O projeto conta com uma API para integração de ferramentas e que contribuiu para a construção do ecossistema. O trabalho mostra que um grande desafio da integração das ferramentas é a padronização, resultando em um alto custo adaptar novas ferramentas. O projeto já é amplamente utilizado e foi usado na implementação inicial da Universidade Aberta do Brasil. Sua estrutura em módulos permite a distribuição das atividades de manutenção entre os desenvolvedores que contribuem para a ferramenta. Embora o artigo [20] não aprofunde as características da infraestrutura, o projeto destaca fortemente a necessidade de modularização e divisão de tarefas no desenvolvimento e

manutenção de um ecossistema educacional, e sendo uma aplicação já amplamente utilizada a sua experiência na forma de planejar e implementar novas funcionalidade em uma aplicação já estabelecida. O trabalho também fala do cuidado na hora de estabelecer o contexto em que o usuário acessa a plataforma. Segundo o IBGE, 98,6% das pessoas que acessam a Internet o fazem por telefone móvel, enquanto apenas 46,2% acessam via computador, dados de 2019 [21]. Esses dados demonstram a necessidade do cuidado com a implementação de uma plataforma educacional, vista a necessidade de garantir que o usuário consiga acessar o conteúdo.

Quando a implementação de ferramentas educacionais esbarra na realidade de um país onde ainda existem muitas escolas e alunos sem acesso a computador ou internet [22], propostas como a de utilizar SBC e smartphones para fazer a interface com a estrutura educacional [23] apresentam uma abordagem sobre o fator adaptativo da proposta de um ecossistema educacional. Com uma proposta de *FOG computing*, onde se tem uma arquitetura distribuída, mas integrada na premissa do usuário, o trabalho tenta fazer face aos desafios de entregar ferramentas educacionais digitais onde não se tem garantia de conectividade.

Ainda no campo de iniciativas nacionais, além de plataformas, podemos destacar a ferramenta TryLogic, uma ferramenta que foi concebida como um serviço LTI, preparada para ser ofertado por plataformas consumidoras certificadas segundo padrão LTI. Nos trabalhos [24, 25], Terremate discute como a ferramenta contribui para o aprendizado do estudante, aborda como é feita a integração da ferramenta a uma plataforma e aborda como o uso de ferramentas complementa o ensino teórico apresentado em classe e como esse complemento possibilita a melhoria no procedimento dedutivo.

2.2 Plataformas de integração para ecossistemas

Em um ambiente tecnológico contemporâneo é comum encontrar diversas soluções que atendem o mesmo problema com soluções diferentes. No caso de plataformas de educação podemos encontrar ferramentas e plataformas desenvolvidas para atender esse propósito com estratégias diferente de como estruturar e integrar os conteúdos. No trabalho do Veiga [2] é feita uma análise qualitativa sobre principais plataformas atuais, com características que os pesquisadores consideraram fundamentais na construção de um ecossistema. Destaca-se que uma plataforma ou padrão não é imutável, mas se adapta às necessidades dos usuários e às demandas de quem desenvolve as ferramentas nesse padrão.

O padrão LTI é uma dessas plataformas e apresenta impactos positivos na iniciativa de facilitar a integração de aplicações educacionais. Ele é proposto no trabalho de [19] como padrão para integrar o ecossistema proposto. O padrão LTI facilita a integração

de AVAs e aplicações que utilizam esse padrão [26] , e com o número de aplicações com esse padrão e muitos com certificação a gama de ferramentas que podem ser acessadas sem precisar de um desenvolvimento dedicado continua crescendo. Com mais de duzentas ferramentas certificadas e mais de vinte plataformas de aprendizado também certificadas o LTI tem destaque como padrão esse tipo de integração [15].

2.3 Destaques acadêmicos na última década

Quando se fala de ferramentas digitais avaliar o comportamento do cenário que envolve a ferramenta é fundamental para avaliar a adoção, maturação ou mesmo a estagnação ou abandono de uma ferramenta. No caso do uso do padrão LTI como plataforma de integração em ambientes de aprendizado digitais podemos ver em trabalhos desenvolvidos na última década o amadurecimento da ferramenta, que teve cinco atualizações de versão nesse período, bem como a adoção desta para desenvolver trabalhos e ferramentas que estão sendo utilizadas atualmente. A Figura 2.2 mostra a linha temporal em que esses trabalhos se encontram.

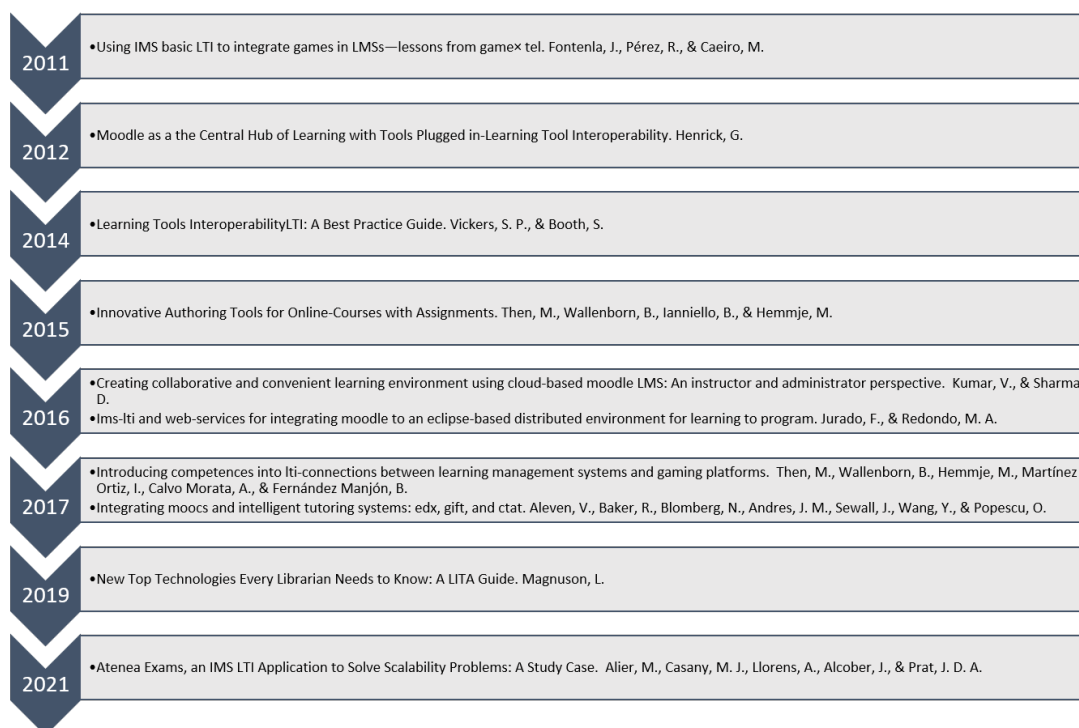


Figura 2.2: Linha temporal de trabalhos voltados para aplicação do padrão LTI .

No trabalho do Fontenla e Caeiro [3] observamos um estudo de caso onde foi proposta a aplicação do LTI na integração de jogos de computador a ferramentas educacionais . Ainda

na primeira versão do LTI, esse projeto traz uma visão compreensiva sobre a vantagem da integração de aplicações externas a plataformas de aprendizado. O trabalho debate a separação entre uma integração direta e indireta (*hard integration* e *soft integration*), desafio esse que deixou de existir nas versões mais novas do LTI. Os desafios apresentados, quando comparados com trabalhos mais atuais, podem exemplificar a maturação que o padrão obteve nessa última década. Na Figura 2.3 podemos ver um exemplo de um jogo digital externo integrado a plataforma do autor.

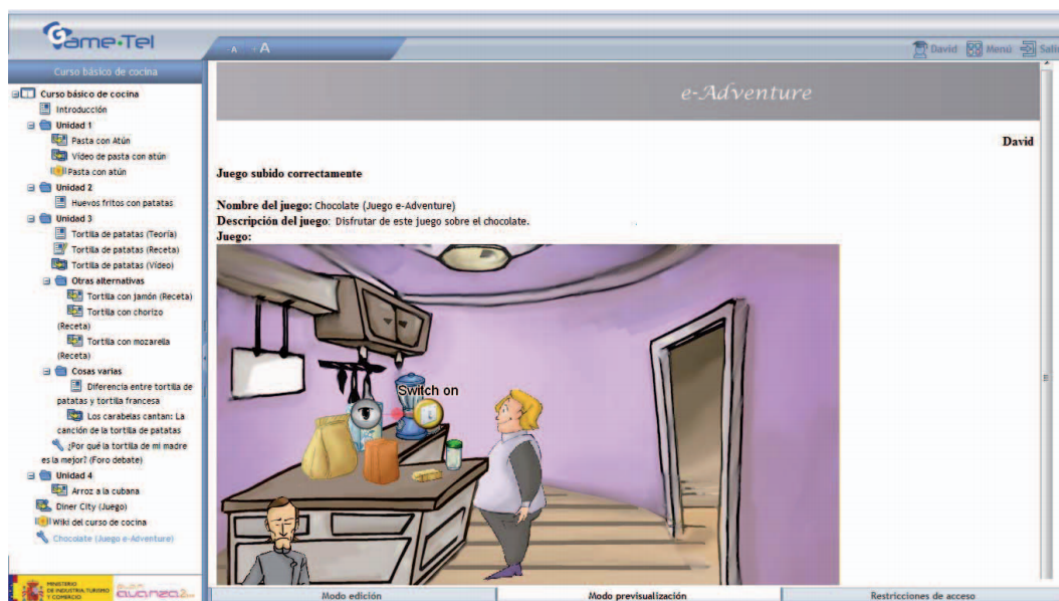


Figura 2.3: Exemplo de jogo integrado a plataforma usando LTI [3] .

Quando se fala de AVAs, o Moodle é citado como um dos mais utilizados. Com mais de 270 milhões de usuários e mais de 184 mil implementações no mundo todo ele é uma referência de aplicação bem sucedida de um modelo de plataforma de aprendizado [27]. No trabalho de Henrick é apresentado o impacto da implementação do padrão LTI no Moodle 2.2 [8]. Ao apresentar formas de aplicar o padrão LTI para integrar ferramentas, bem como exemplificando provedores que relevantes que já estão no padrão, o trabalho mostra o potencial de crescimento que o padrão de integração pode impactar no meio de aprendizado digital.

Enquanto os trabalhos anteriores mostram o potencial do padrão LTI como uma plataforma de integração de ferramentas educacionais, o trabalho de Vickers traz de uma maneira concisa como funciona a estrutura do LTI de uma forma mais clara que as documentações da IMS [9]. O Guia de Boas Práticas apresenta uma visão sistêmica de como estruturar uma aplicação usando o padrão LTI, mas também aponta cuidados que o de-

desenvolvedor deve ter para garantir que o usuário, neste caso focado no administrador de uma plataforma educacional, tenha facilidade para integrar a aplicação a sua plataforma.

O trabalho de Then mostra um passo importante na maturação do modelo de plataforma de aprendizado com integração de ferramentas externas [10]. Ao abordar a plataforma de aprendizado digital utilizando o modelo de Aprendizado Baseado em Competência (CBL) o trabalho apresenta uma proposta de solução para uma solução de aprendizado híbrido utilizando a integração de ferramentas como ponto importante para facilitar a construção e manutenção do sistema. O trabalho traz outro padrão da IMS, o Learning Design (LD), uma linguagem de modelagem desenhada para descrever cenários educacionais digitais. Enquanto o IMS LD não suporta diretamente o CBL, ele é estruturado o bastante para permitir a extensão do modelo usando o modelo Personal Competence Domain Model (PCDM). O trabalho é bem direto sobre o desafio que o LTI apresenta quando o desenvolvedor tenta implementar um modelo mais complexo que o previsto pelo padrão, mas mostra as possibilidades de adaptação, bem como apresenta os impactos que as atualizações de versão do LTI trazem para aumentar a abrangência da ferramenta. O trabalho apresenta uma análise aprofundada da preparação de uma plataforma para a integração de ferramentas externas, utilizando o Moodle como plataforma escolhida.

O trabalho de Kumar avalia a relação entre as teorias de ensino e aprendizagem e o uso de ferramentas educacionais digitais [11]. Ao avaliar como os AVAs impactam no aprendizado colaborativo e conveniente online, o trabalho traz uma visão pedagógica sobre a aplicação das ferramentas digitais como plataforma de aprendizado. Utilizando novamente o Moodle como ferramenta de referência, ele avalia como os diferentes papéis educacionais (professores, alunos, administradores) se comportam em um modelo digital. O trabalho avalia de maneira breve duas plataformas de integração de ferramentas digitais, o GLUE e o LTI, destacando as possibilidades que as duas ferramentas trazem, mas sem fazer uma distinção clara dos prós e contras das duas.

No trabalho de Jurado e Redondo é apresentada uma visão processual da integração da ferramenta Coala como provedor LTI e o AVA Moodle como consumidor LTI [4]. O trabalho mapeia como as mensagens entre as plataformas são trocadas e tratadas, trazendo uma visão mais descritiva de como o padrão funciona na prática, como pode ser visto na Figura 2.4. O trabalho propõe o uso de plataforma de integração como solução para reduzir o custo de desenvolver ferramentas dedicadas para atender o um modelo de aprendizado digital, reiterando a necessidade de reúso de ferramentas como forma de otimização.

Em um novo trabalho de Then é avaliado o impacto da integração de uma plataforma de jogos digitais em um sistema de aprendizado [12]. Utilizando o RAGE Analytics Environment, uma plataforma de gamificação, e o FUH-WebAssign, uma ferramenta de

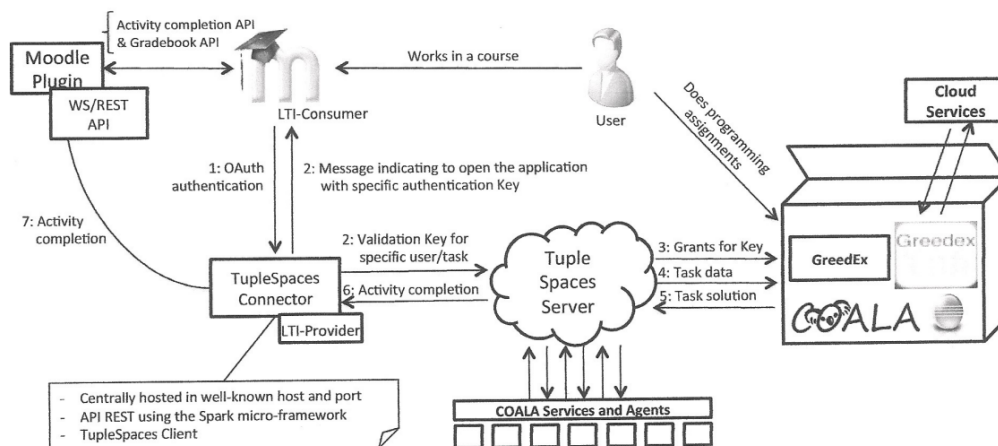


Figura 2.4: Troca de mensagens entre provedor e consumidor [4] .

avaliação do aluno, como provedores, o Moodle como AVA e o LTI como plataforma de integração o trabalho apresenta uma visão prática de como é feita a integração com mais de uma ferramenta, sendo um exemplo mais prático de como a integração pode ser utilizada no dia-a-dia. O trabalho ainda assim destaca alguns desafios que a limitação do LTI apresenta e que requer atenção do desenvolvedor.

Os Cursos massivos abertos online (*Massive Open Online Courses*) (MOOCS) são exemplo da quebra do paradigma do ensino tradicional ao propor uma plataforma de ensino aberta estruturada em cursos. No trabalho [13], é apresentada uma visão prática sobre a integração de ferramentas de aprendizado a uma plataforma de ensino utilizando o LTI. O trabalho apresenta os detalhes de como o padrão LTI foi utilizado para integrar as ferramentas GIFT, uma plataforma orientada a serviços para a criação e gestão de tutores digitais, e o CTAT, uma ferramenta que permite a criação de tutores cognitivos, a plataforma MOOCS edX. O trabalho avalia a adaptabilidade do padrão LTI, ao apresentar a ferramenta GIFT primeiramente como provedor, sendo integrado ao edX e posteriormente avaliando a possibilidade de utilizar o GIFT como consumidor, e utilizando o CTAT como provedor. Essa visão proposta mostra como as ferramentas podem se integrar de forma dinâmica, e com isso a necessidade do cuidado que se deve ter ao definir a arquitetura que a plataforma como um todo deve adotar.

No capítulo 14 do livro *New Top Technologies Every Librarian Needs to Know: A LITA Guide* [14], Varnum mostra ainda outra faceta importante do uso do padrão LTI, que é a integração de ferramentas bibliográficas aos AVAs. Com exemplos de provedores LTI voltados para gestão de materiais educacionais, como o EBSCO Curriculum Builder e o Libris Leganto, duas ferramentas voltadas para gestão e integração de bibliotecas que permitem ao professor escolher os materiais que serão utilizados em determinado curso

e integrar esses materiais ao desenvolver do aprendizado. Já a aplicação Springshare LinGuide permite ao AVAs encontrar os guias de pesquisa mais relevantes ao conteúdo do curso utilizando metadados para cruzar o conteúdo do curso e o das pesquisas. O trabalho apresenta uma discussão sobre o impacto de integrar o processo de pesquisa ao processo de aprendizagem.

O trabalho [5] foi um estudo de caso do impacto que a pandemia de COVID-19 teve nas plataformas de ensino digitais. O trabalho mapeou uma plataforma baseada em Moodle que já operava antes da pandemia, e com a política de confinamento viu um aumento de 800% na carga do servidor onde a plataforma operava. Esse crescimento pode ser visto pelos indicadores de número de acessos, número de páginas acessadas e duração por sessão apresentados nos gráficos da Figura 2.5. Com isso é levantado o problema de escalabilidade e no trabalho é proposta a utilização do padrão LTI como forma de solucionar esse problema. Ao utilizar uma plataforma de integração foi possível distribuir a carga de acessos que estava parcialmente centralizada no AVA para as ferramentas integradas externas, fazendo assim com que o servidor não ficasse sobrecarregado. O trabalho também apresenta como vantagem a não dependência entre as ferramentas externas e o AVA, o que significa que caso uma ferramenta fique sobrecarregada ou pare o AVA não seja impactado diretamente, ou caso seja necessário atualizar o AVA ou a ferramenta, não seja necessário se preocupar com o acoplamento das instalações. Por outro lado o trabalho levanta a preocupação com a privacidade e confidencialidade dos estudantes, tópico não tratado no trabalho presente, mas abordado brevemente nos trabalhos em andamento na equipe.

Esses trabalhos trazem um panorama da evolução da implantação, dos impactos e do reconhecimento do padrão LTI como uma ferramenta que permite atingir o objetivo de integrar serviços educacionais, apresentando vantagens, desvantagens, desafios e descobertas. As contribuições desses documentos neste trabalho foram mapeadas na Tabela 2.1.

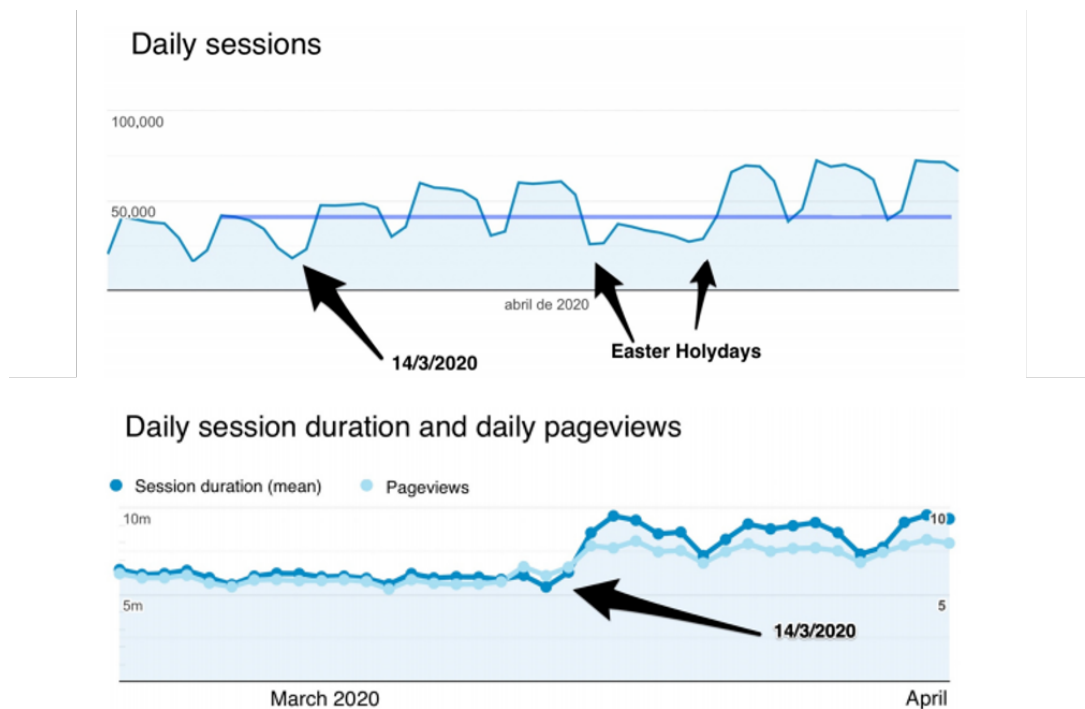


Figura 2.5: Aumento do número de acessos, do número de páginas acessadas e da duração por sessão durante a pandemia [5] .

Ano	Trabalho	Proposta	Impacto neste trabalho
2011	[Fontenla, J., Pérez, R., & Caeiro 2011]	Padrão LTI para integrar jogos em AVA	Desafios de integração do Padrão LTI
2012	[Henrick, G. 2012]	Avalia o impacto da adoção do LTI no Moodle 2.2	Mostra o potencial do Padrão
2014	[Vickers, S. P., & Booth, S. 2014]	Guia de Boas Práticas para implementar LTI	Mais didático e consiso que a documentação oficial do LTI
2015	[Then, Wallenborn, Ianniello, & Hemmje 2015]	CBL implementado em LTI com LD	Mostra o desafio e adaptabilidade dos padrões IMS
2016	[Kumar, V., & Sharma, D. 2016]	Relação entre as teorias de ensino e aprendizagem e o uso de ferramentas educacionais digitais	Apresenta o Glue e LTI como padrões de integração ao Moodle
2016	[Jurado, F., & Redondo, M. A. 2016]	Integração Coala Moodle com LTI	Mapeia a estrutura das mensagens LTI
2017	[Then, Wallenborn, Hemmje, Ortiz, Morata, & Manjón 2017]	Integração de uma plataforma de jogos digitais em um sistema de aprendizagem	Cenário com dois produtores e um consumidor
2017	[Aleven, Baker, Blomberg, Andres, Sewall, Wang, & Popescu 2017, May]	Integração de serviços de tutores cognitivos em MOOCs com LTI	Cenário de integração em cascata
2019	[Magnuson, L. 2019]	Guia sobre tecnologias para biblioteconomia	Propõe o uso do LTI para integrar bibliotecas digitais a disciplinas virtuais
2021	[Alier, Casany, Llorens, Alcober, & Prat 2021]	Abordou um caso de impacto da pandemia no ensino digital	Apresentou o uso do LTI para distribuir carga

Tabela 2.1: Mapeamento das contribuições dos trabalhos citados [3, 8, 9, 10, 11, 4, 12, 13, 14, 5].

Capítulo 3

Referencial Teórico para a Proposta

3.1 Panorama de provedores LTI certificados

3.1.1 O padrão LTI

O LTI é um padrão criado para facilitar a comunicação entre aplicações de aprendizado, como AVAs, repositórios, laboratórios e portais, de uma maneira segura e simplificada [28]. O LTI possui um conjunto de formatos de mensagem, formas de autenticação e padronizações de como informações são transferidas de um sistema para outro. O LTI permite a comunicação entre as aplicações através de mensagens *HTTP* puro ou através de serviços utilizando *JSON*. O processo de autenticação utiliza protocolo *OAuth 2.0* e o LTI requer que as mensagens sejam assinadas com as informações de: quem solicita, de onde solicita, o que solicita e a chave de identificação.

O LTI distingue os sistemas em dois tipos: Consumidor e Provedor. O Consumidor é o sistema que o usuário acessa, normalmente um Ambiente Virtual de Aprendizado, e como Consumidor o LTI permite que esse sistema faça requisição de conteúdos a outra plataforma com o papel de Provedor [28]. O Provedor, por sua vez, possui um conjunto de informações, conteúdos e ferramentas formatados para serem apresentados dentro do Consumidor. Esses conteúdos podem ser customizados para cada usuário, e por isso é necessária a troca de dados e de credenciais de acesso, que são feitas usando o modelo de mensagem do LTI. O LTI apresenta uma integração facilitada das ferramentas, permitindo a centralização dos conteúdos, unificação dos perfis dos usuários e garantindo a compatibilidade entre as plataformas de aprendizado.

A especificação LTI, assim como sua documentação, certificação e gestão são elaborados pelo consórcio IMS Global, uma organização sem fins lucrativos com propósito de unificar e padronizar tecnologias de aprendizado e educação. Seu propósito é prover padrões de interoperabilidade para tecnologias de aprendizado com licença aberta [29].

Plataformas (Consumidor)	24
Plataformas (Consumidor e Provedor)	23
Ferramentas(Provedor)	220
Ferramentas de Supervisão(Provedor)	2
Produtos certificados	269

Tabela 3.1: Distribuição de aplicações certificadas por tipo. Dados:[15]

Membros	618
Não certificados	424
Certificados	194
Fornecedor	182
Instituição de ensino Superior	6
Ensino básico	3
Entes de governo	2
Associações	1

Tabela 3.2: Divisão dos membros da IMS com certificação. Dados:[16]

3.1.2 A certificação LTI

O IMS oferece um conjunto amplo de licenças para cada um dos padrões gerenciados pela IMS, para cada versão. Para LTI especificamente, além da certificação para LTI, dada para cada versão, existem duas certificações agregadoras: LTI Advantage e LTI Advantage Complete. A certificação é dada para um produto e não para a entidade, e tem como função garantir que a ferramenta a ser oferecida seja compatível com as outras ferramentas que possuem o mesmo padrão, dar visibilidade a ferramenta e ao fornecedor e permitir que usuários da ferramenta tenham suporte não só do desenvolvedor, mas também dos técnicos especialistas da IMS. Atualmente existem 269 aplicações certificadas distribuídas conforme a Tabela 3.1 [15]. Para submeter uma ferramenta para ser certificada, o fornecedor precisa ser registrado como membro da IMS. Dos 618 membros atualmente na IMS apenas 194 possuem uma aplicação certificada, e destes a maior parte são desenvolvedores e fornecedores de *software*, como pode ser observado na Tabela 3.2 [16].

Existem três tipos de membro *Contributing Member*, *Affiliate Member* e *Alliance Member*. Os membros do tipo *Contributing Member* são Membros votantes do IMS, podendo atuar em todas e quaisquer iniciativas e trabalho técnico do IMS. Membros *Affiliate Member* tem acesso a maior parte dos trabalhos técnicos da IMS, mas não tem poder de voto. Já os membros *Alliance Member* tem acesso a um subconjunto restrito de trabalhos técnicos da IMS e também não tem poder de voto. [30] Para se tornar um membro da IMS a instituição deverá se inscrever e pagar um anuidade que varia de acordo com o tipo de instituição, divididos entre fornecedores por faturamento anual e instituições de ensino por porte ou orçamento anual.

3.2 Aparato tecnológico e metodológico para criação e manutenção de um provedor LTI

3.2.1 Microsserviços e Containers

Microsserviço é um paradigma de arquitetura de software onde uma aplicação é dividida em diversos componentes e que são executados de forma independente. Essa solução traz diversas vantagens como desenvolvimento simultâneo das ferramentas, melhor gestão de recurso, escalabilidade.[31] Devido a natureza distribuída e modular da implementação do ecossistema educacional proposto, é fácil perceber como esse modelo se adéqua melhor aos requisitos do que um modelo monolítico ou cliente-servidor, pois esses não apresentam uma característica modular, enquanto modelos como Peer-to-Peer não apresentam a hierarquia necessária para um serviço educacional. O modelo SOA, embora apresente uma estrutura modular propõe um modelo de comunicação direcionado a dados que ultrapassa a proposta de comunicação do padrão LTI.

Um container armazena um microsserviço ou aplicação e o conjunto de recursos necessários para executar o mesmo. Um serviço de orquestração de containers permite a execução de várias aplicações ao mesmo tempo compartilhando recursos de um sistema operacional, mas permitindo isolar as execuções de forma a garantir a segurança da aplicação. A inicialização de um novo container é feita rapidamente, permitindo lançamento efêmero para atender a necessidades temporárias ou teste, ou lançamentos persistentes, preparados para armazenar os dados e resultados mesmo após o encerramento do mesmo. O container apresentam um nível de abstração que permite que a mesma imagens seja executada independente do sistema operacional do orquestrador. Containers apresentam um contraste com outras tecnologias de virtualização, como máquinas virtuais, pois estas oferecem uma solução monolítica, onde cada aplicação possui o seu próprio sistema operacional. Uma máquina virtual permite a execução de diversos serviços e funções em uma única instância. Por outro lado as maquinas virtuais exigem mais recursos do ambiente computacional, porque mesmo sistemas operacionais leves são medidos em gigabytes de tamanho. Já o container otimiza o recurso armazenando em suas imagens apenas os componentes do sistema necessários para a execução do serviço alocado nele, assim ocupando menos espaço em disco e levando menos tempo para inicializar o mesmo.

3.2.2 Tecnologias para infraestrutura de orquestração de microserviços

Docker

O Docker é uma aplicação para uso de containers localmente. Ele é composto de um conjunto de ferramentas para criação e gestão de imagens, lançamento e execução de containers. Ele é mantido pela empresa Docker Inc, mas como projeto open source ele é aberto para a comunidade de desenvolvimento e passa por melhorias constantes. O Docker utiliza a arquitetura do kernel Linux e suas funcionalidades, como *namespaces*, para executar containers em processos separados, garantindo a segurança da independência entre eles.

Kubernetes

Kubernetes é uma plataforma de orquestração de containers. Diferente do Docker, que é utilizado para executar programas localmente, o Kubernetes permite executar containers em formato de nuvem, distribuído e em clusters de computadores e servidores. O Kubernetes foi desenhado e desenvolvido por engenheiros da Google [32], utilizando os aprendizados com o sistema de orquestração Borg, para atender a demanda de orquestrar bilhões de containers em sistemas distribuídos pelo mundo todo. A Red Hat colaborou com a Google na criação do Kubernetes, se tornando o segundo principal contribuidor. Posteriormente o Kubernetes foi doado ao Cloud Native Computing Foundation (CNCF), um projeto desenvolvido pela fundação Linux para centralizar o desenvolvimento de tecnologias *open source* de componentes críticos para infraestruturas de tecnologias globais, como Kubernetes, Prometheus, e Envoy.[33]

O Kubernetes permite a execução distribuída de containers, controle e automação de lançamentos e updates, escalonamento, gerenciamento de estruturas de armazenamento e manutenção da orquestração.

3.2.3 Protocolo de autorização

Um dos principais braços técnicos do LTI é a centralização das autenticações das diversas ferramentas e para isso o LTI 2.0 conta com o padrão OAuth2 para gerir as trocas de credenciais entre as ferramentas. O OAuth2 permite uma estrutura de escopo para determinar o perfil ou papel do usuário naquele domínio e o LTI utiliza esse escopo para determinar qual o papel do usuário na aplicação específica que ele esteja acessando. Isso porque um usuário pode ter papéis diferentes em ferramentas ou plataformas diferentes e essa distinção é importante durante a integração das mesmas.

O OAuth2 conta com uma estrutura de tokens, que permite ao autenticador validar o acesso do usuário durante o período todo em que o mesmo esteja utilizando a ferramenta, e também permite a revogação de tokens para garantir a segurança do sistema que está utilizando o mesmo.

Capítulo 4

Proposta

O resultado do trabalho é fornecer uma infraestrutura que permitirá o desenvolvimento, manutenção e integração de ferramentas educacionais utilizando o padrão LTI. Para atender essa expectativa foi escolhida a arquitetura de microsserviços utilizando containers. Para orquestrar esses serviços será instalado em um servidor a plataforma Microk8s, que permitirá aos alunos e professores o lançamento de containers com as aplicações necessárias para o desenvolvimento dos serviços LTI do ecossistema, a plataforma GitLab, um repositório de código com integração CI/CD, e a ferramenta OpenID, uma ferramenta OAuth2 para integração de tokens de autenticação entre as ferramentas, plataformas e o LTI.

4.1 O projeto

O projeto *smartUnB.ECOS* propõe um estudo sobre os ecossistemas educacionais apoiados por computador através de diversos trabalhos integrando ferramentas de aprendizado[1]. A proposta não requer apenas a construção de um ambiente de aprendizado, mas a possibilidade de investigar o impacto e o desenvolvimento do ponto de vista formal. Ao escolher o padrão LTI como especificações para integração, escolha essa motivada nos trabalhos acadêmicos na área que aplicaram esse padrão, em destaque [19], é definida uma estratégia de estrutura de implantação limitada pela capacidade da especificações escolhida.

4.1.1 A escolha do LTI

Durante a fase de estudo deste trabalho foi observada que enquanto muitos trabalhos implementavam o padrão LTI como ferramenta de integração [19, 26, 4, 13], existiam outros que argumentavam que o padrão LTI apresentava limitações quanto a capacidade de rastreamento do progresso e interação do usuário com o objeto [6, 17, 2]. Estes trabalhos

apresentavam alternativas como o xAPI e seu subsequente CMI5 como alternativas que permitiam a comunicação com rastreabilidade. Isso porque os desenvolvedores do xAPI propuseram uma nova estrutura de aplicação, chamada de *Learning Record Stores* (LRS) que permite o armazenamento, análise, agregação e consumo dos dados trocados e produzidos por uma plataforma de ensino digital. O LRS no entanto não substitui o AVA, mas complementa a sua funcionalidade e em uma visão de alto nível do projeto pode se propor uma estrutura onde o LRS está acoplado ou isolado do AVA, como mostra a Figura 4.1.

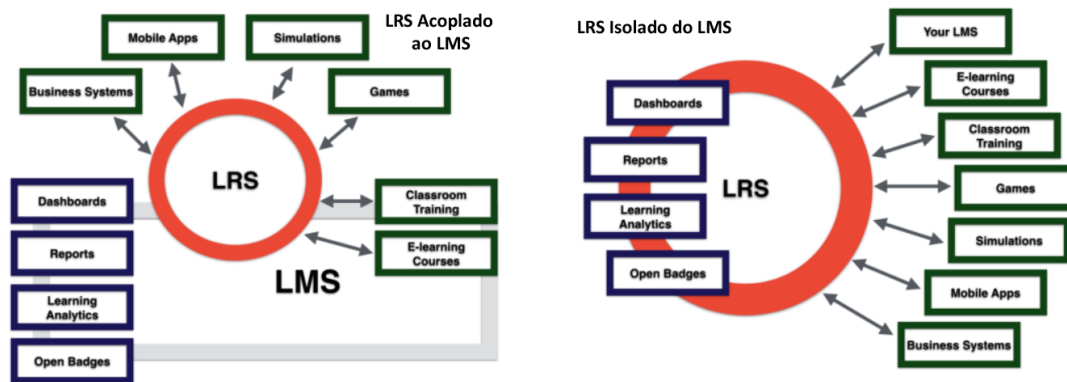


Figura 4.1: Diagramas das estruturas com o LRS acoplado ou isolado do AVA(LSM) [6].

Enquanto os padrões integração xAPI e CMI5 apresentam as funcionalidades de integração e de rastreamento de dados entre AVAs, aplicações e LRSs em um protocolo, a proposta da IMS foi separar os papéis de integrar, com o padrão LTI e de rastrear, com o Caliper, em estruturas diferentes, permitindo controle mais direto de qual funcionalidade será implementada em cada momento. Na Figura 4.2 podemos compreender a estrutura proposta, onde o padrão LTI faz a integração das aplicações e o Caliper cuida da aquisição e gestão de dados.

Dessa forma escolha inicial de seguir os padrões IMS não prejudicará a proposta do projeto *smartUnB.ECOS*, mas não existe impedimento para que trabalhos futuros investiguem outros padrões, podendo até ser proposta uma implementação híbrida, utilizando mais de um padrão operando dentro de uma plataforma.

4.1.2 O escalonamento

Um dos pontos abordados neste trabalho é a gestão de recurso do servidor, e projetar o crescimento na busca pelo serviço e preparar a infraestrutura para acomodar esse crescimento faz parte do planejamento de um projeto de sistema. Neste ponto, como foi mostrado no trabalho [5] os padrões de integração, com destaque para o LTI tem um papel

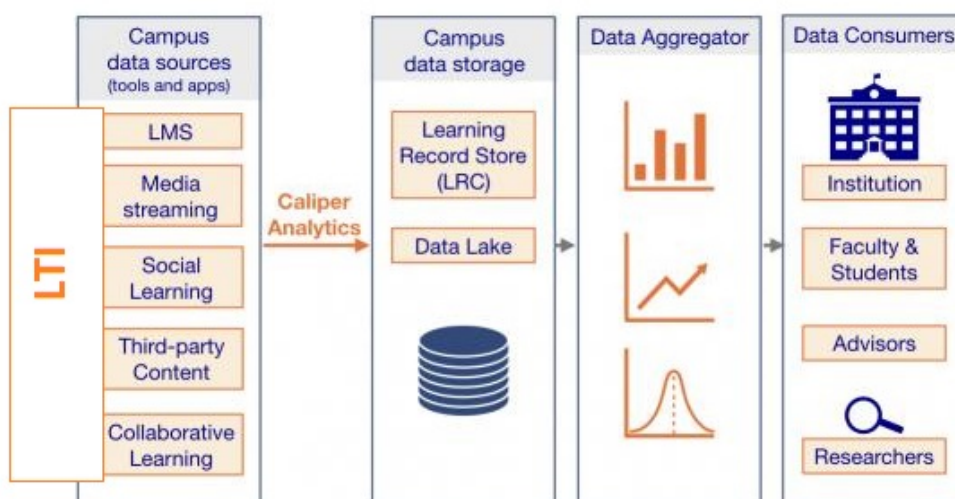


Figura 4.2: Diagrama do papel do LTI e Caliper [7].

importante no escalonamento do sistema. Isso porque ao permitir que as aplicações sejam distribuídas, podendo estar em servidores externos ou mesmo serem geridas e fornecidas por outras entidades, essa integração permite que o tráfego não se concentre no servidor central onde está alocado o ecossistema digital. Também fica aberta a possibilidade de execução temporária de aplicações em nuvem para atender demandas esporádicas, como um aumento de acessos no final de um semestre da universidade ou uma eventualidade como a da pandemia de 2020. No caso do *smartUnB.ECOS*, a possibilidade da integração com o servidor de nuvem Gigacandanga¹ pode ser investigada para a distribuição de carga computacional do projeto.

4.2 O servidor

A proposta inicial do projeto *smartUnB.ECOS* é a instalação das ferramentas em um servidor disponibilizado para o acesso dos desenvolvedores e com acesso para os usuários através da página web. Esse servidor está alocado na UnB e está sendo acessado remotamente para a instalação e administração da estrutura. O servidor apresenta as especificações apresentadas na 4.1, e esses recursos foram considerados na hora de dimensionar a infraestrutura, assim como na escolha das ferramentas que serão inicialmente ofertadas.

¹<https://gigacandanga.net.br/>

Processador	Intel(R) Xeon(R) CPU E31220 @ 3.10GHz
Memória RAM	16GB
Armazenamento	2x250GB
Sistema Operacional	Ubuntu Server 20.04.2 LTS

Tabela 4.1: Especificações do servidor

Nesse servidor está instalado o sistema operacional Linux Ubuntu Server 20.04.2 LTS e o sistema operacional é tratado como requisito para a escolha das ferramentas que serão disponibilizadas na infraestrutura. O servidor já possui algumas aplicações instaladas, a maior parte serviços HTTP e serão avaliadas, descontinuadas e se necessário migradas para o modelo de microsserviços.

4.3 Arquitetura da Infraestrutura

Para definir a arquitetura da infraestrutura foram identificados os casos de uso conforme o diagrama da Figura 4.3. A partir do diagrama e das definições apresentadas no Referencial Teórico foi construída a proposta de infraestrutura apresentada na Figura 4.4.

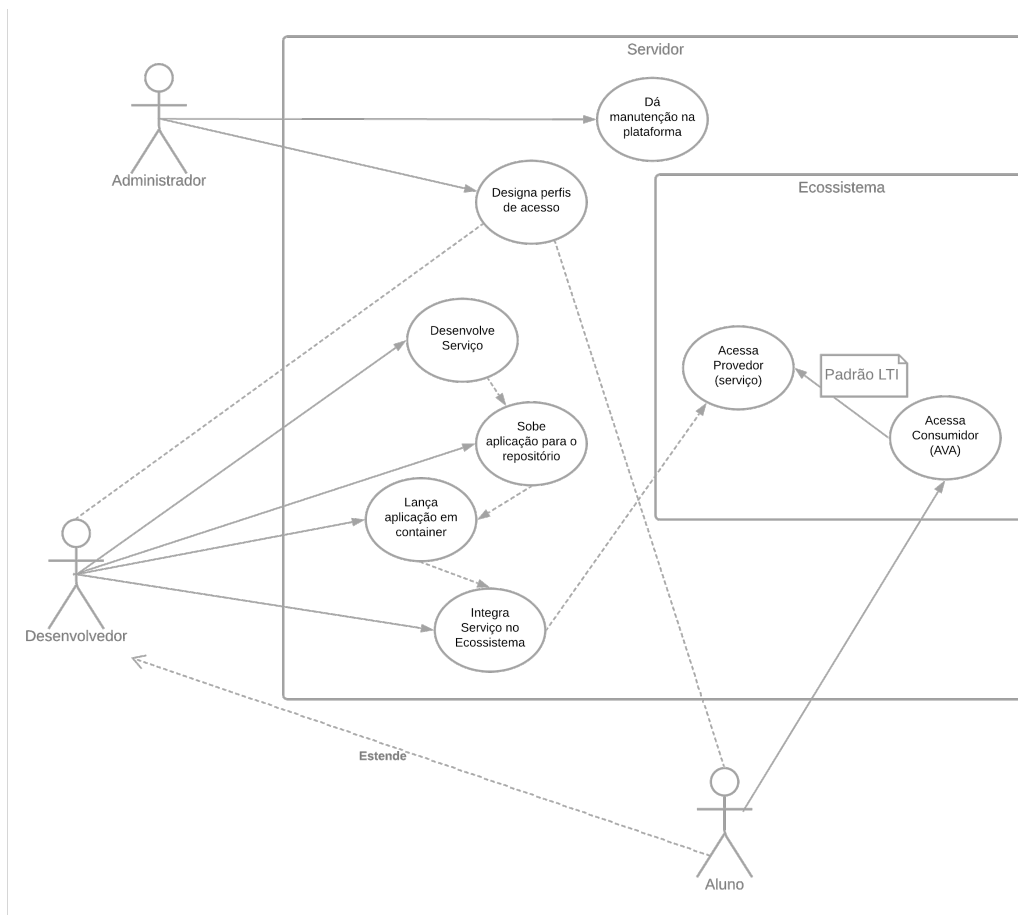


Figura 4.3: Diagrama de Caso de Uso da infraestrutura.

4.4 Implementação da Infraestrutura

A implementação da infraestrutura proposta na Figura 4.4 foi feita mapeando cada módulo a uma aplicação que atendia os requisitos determinados e cada um de seus módulos é descrito nesta seção.

4.4.1 Kubernetes com Microk8s

O Microk8s é uma plataforma Kubernetes distribuída pela Canonical para orquestração de containers. Sua função é armazenar, lançar, monitorar e desativar containers, gerir a rede virtual que permite que os microsserviços conversem entre si e com o usuário, e gerir os nodos, máquinas ligadas ao servidor para executar os serviços de forma distribuída e escalonável. A Figura 4.5 mostra uma das visões que o administrador tem ao gerir os microsserviços.

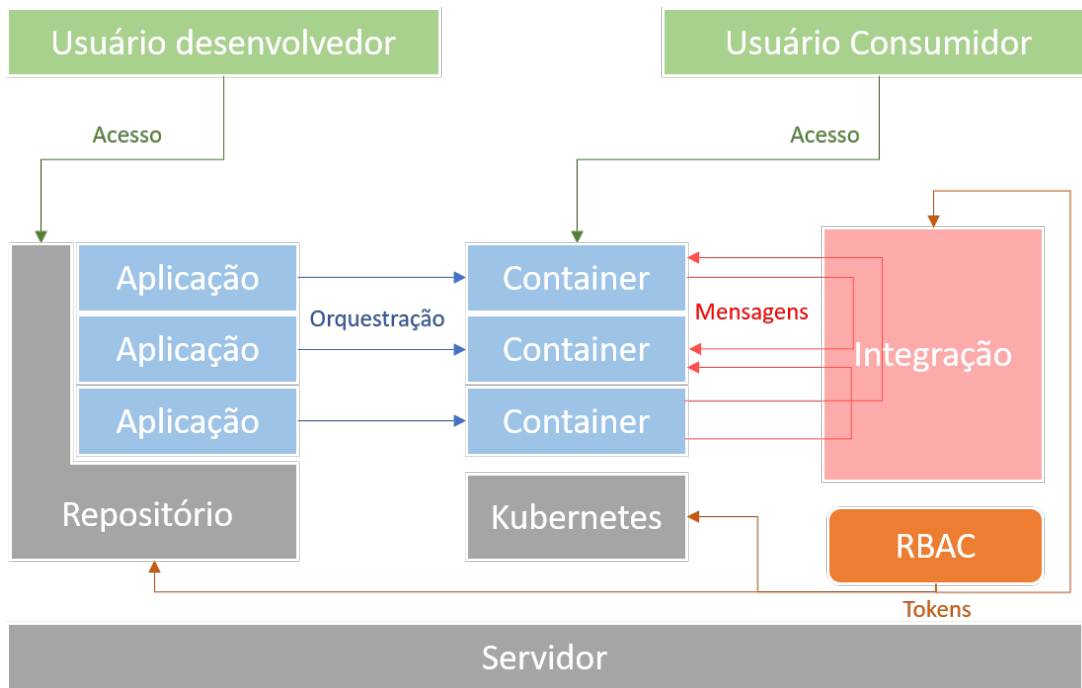


Figura 4.4: Arquitetura proposta.

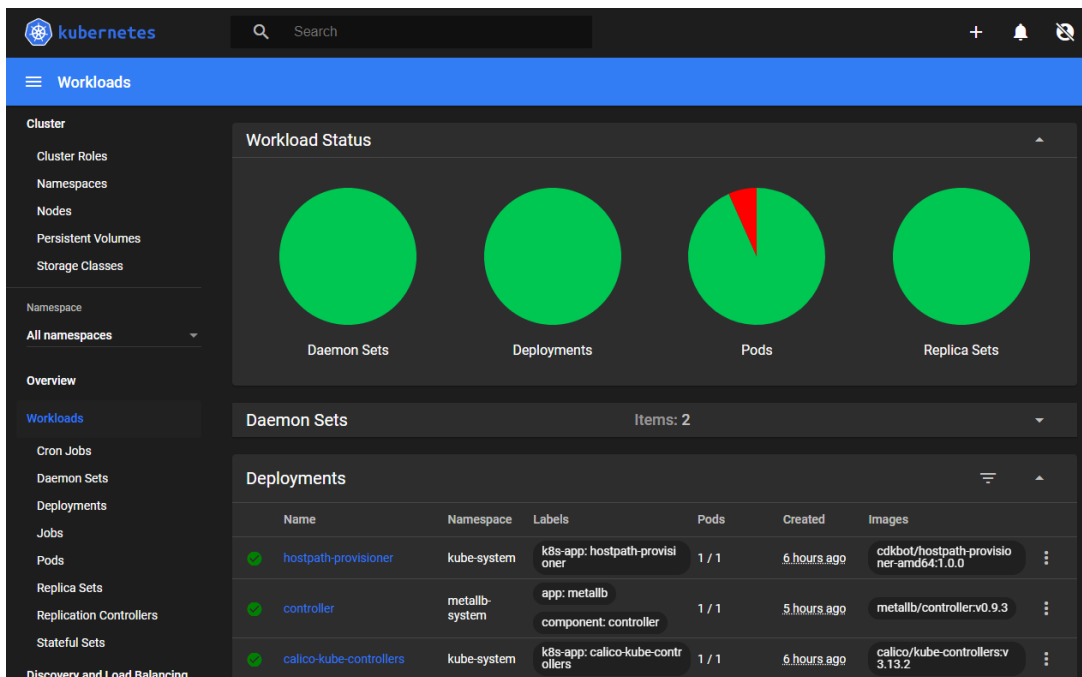


Figura 4.5: Painel de controles do Kubernetes.

A escolha da plataforma Kubernetes deve ser feita considerando a estrutura do servidor em que será instalado, as necessidades de ferramentas e de crescimento da plataforma. O

servidor onde será instalada a infraestrutura utiliza Linux Ubuntu Server, que também é da Canonical e por isso a compatibilidade com Microk8s é garantida. A integração entre o Microk8s e o repositório GitLab está bem documentada tanto pela Canonical quanto pela Gitlab e essa integração será fundamental para o funcionamento da infraestrutura.

Para gerir a plataforma é necessária a configuração do RBAC prevendo perfis de administração e acesso, e deve ser preparada a integração com o GitLab. Essa integração permitirá que o usuário lance containers diretamente do repositório de código e facilitará a manutenção das aplicações.

4.4.2 Repositório de Código GitLab

Uma das necessidades do ambiente proposto é a capacidade de centralizar e armazenar os materiais produzidos, documentações e contribuições dos usuários de forma a permitir a consulta, a manutenção e continuidade dos projetos e aplicações. Quando um aluno produz uma aplicação em sua máquina local, muitas vezes esse trabalho é perdido quando ele se desvincula da universidade. A oferta de um repositório de código-fonte facilita o acesso a outras produções de forma segura, hierárquica, e com controle de versão, reduzindo a perda de material, facilitando a manutenção e atualização e permitindo a continuidade do projeto. Na Figura 4.6 pode ser vista a tela de gestão de projeto do GitLab.

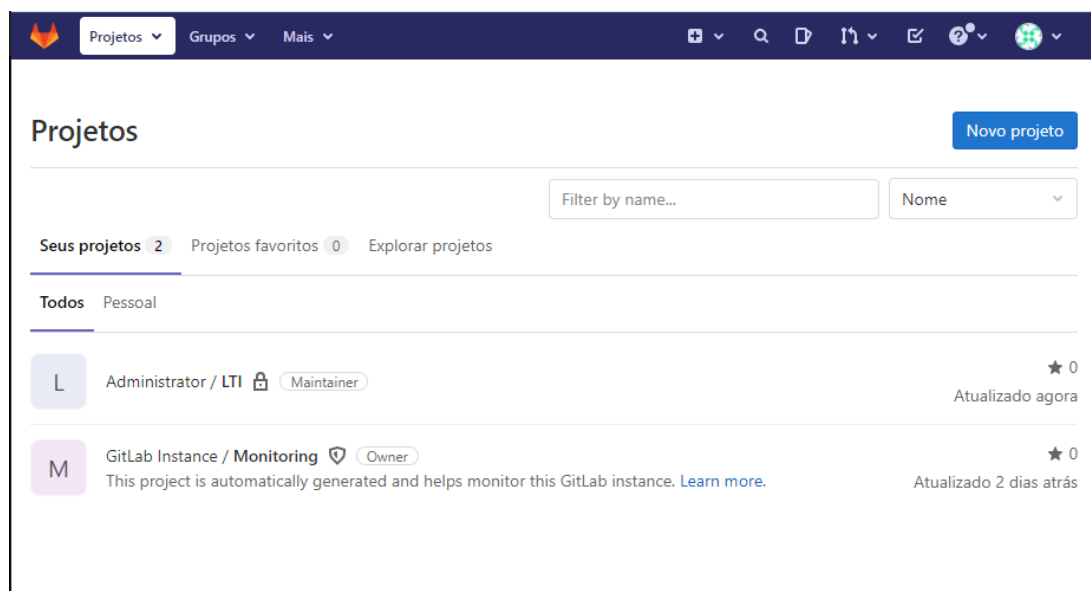


Figura 4.6: Tela de gestão de projeto do GitLab.

Outro ponto importante é a facilidade de integrar uma solução de CI/CD, permitindo um desenvolvimento facilitado, com agilidade de teste e seguro. A solução de repositório

que será instalada é o GitLab, uma ferramenta web de código aberto para armazenamento, controle de versão, revisão de código e integração CI/CD.

Das soluções de repositório de software self hosted, ou seja, instaladas localmente no servidor, o GitLab é a mais utilizada. Gratuito, de código aberto, e com as principais ferramentas de integração as outras estruturas e com uma vasta documentação sobre integração com estruturas de Kubernetes, ele foi escolhido como ferramenta para integrar a infraestrutura.

Na construção da plataforma GitLab foi escolhida uma abordagem de plataforma fechada, onde o usuário terá que ser criado pelo administrador do servidor. Essa decisão tem em mente que a ferramenta está exposta a web e considerando a quantidade de recurso limitada a possibilidade de um visitante criar uma conta e utilizar a ferramenta apresenta um risco para o sistema. Ao escolher uma plataforma fechada, embora se tenha um procedimento burocrático para o primeiro acesso, o usuário e o administrador tem a segurança de que o recurso está distribuído razoavelmente. Foi considerada a possibilidade de integrar ao RBAC da UnB, utilizando o email institucional para o acesso, mas mesmo esse ainda apresenta uma gama maior de potenciais usuários do que o previsto para a estrutura inicial. Destaca-se que esta limitação é para o uso da plataforma de desenvolvimento e não afeta o acesso de usuários às ferramentas educacionais que serão fornecidas no mesmo.

4.4.3 Controle de acesso

Dentro do ambiente educacional da universidade existe uma cultura colaborativa e incremental onde alunos e professores desenvolvem, melhoram e integram aplicações que são disponibilizadas, testadas e utilizadas dentro do meio. Dessa forma é necessário a prover uma infraestrutura aberta, mas que permita uma gestão racional do recurso computacional limitado. Uma solução adequada é a utilização de um RBCA que permita acesso aos diferentes perfis de forma automatizada, mas ofereça ferramentas que para disponibilizar mais recursos de acordo com a necessidade do usuário.

O OpenID permite a integração do LDAP com a sua ferramenta de RBCA, permitindo utilizar os usuários já existentes dentro das plataformas da UnB como forma de autenticação e de hierarquização dos perfis. Dessa forma é possível automatizar o quanto de recurso um usuário padrão, no caso aluno, pode utilizar e permitir aos professores perfis de administrador para gerenciar os recursos, como aumentar ou diminuir o acesso dos alunos conforme a necessidade do projeto. Ele também gere os tokens gerados pelas ferramentas e será integrado ao LTI como ferramenta OAuth2 para a gestão dos perfis.

4.5 O desenvolvimento

Antes da instalação da infraestrutura no servidor, foi criada uma máquina virtual localmente como ambiente de teste onde as ferramentas foram integradas e a interface testada. Durante essa fase foram expostas incompatibilidades de versão, foi avaliado o impacto da ordem da instalação das ferramentas, a necessidade da troca de algumas portas de serviço que causavam conflitos e o teste integrado da infraestrutura como um todo. Depois de quatro versões, onde ao final de cada uma a máquina virtual era revertida para a posição inicial de instalação, foi definido um script de instalação considerado adequado para atender as expectativas da estrutura que foi desenhada e aplicado no servidor.

4.6 A operação

Uma vez implementada a infraestrutura no servidor foi preparado um teste utilizando a aplicação VisCC, desenvolvida como *webapp* com padrão LTI por Cardoso em um trabalho prévio do projeto [34]. A aplicação tem como proposta o aprendizado no campo de tradutores, propondo ofertar o conteúdo de forma interativa aproveitando o modelo de aprendizado virtual. A aplicação apresenta uma estrutura dividida em um *frontend* web em Vue.js em cima de Node.js e um *backend* em Python utilizando o Flask como interface de comunicação HTTP. O código-fonte da aplicação está disponível do GitHub para download.

4.6.1 A preparação

Como a infraestrutura implementada está estruturada em microsserviços, uma vez adquirida a aplicação foi necessário prepará-la para ser executada em um ambiente Kubernetes. Primeiramente foram separados os pré-requisitos do *frontend* e *backend* seguindo a documentação disponibilizada. Com isso foram criados *Dockerfiles*, arquivos que servem como receita para a execução do código criado. Foram criados três arquivos de configuração, sendo o terceiro para a camada HTTP que fará o papel da comunicação entre as duas partes e da aplicação com o usuário. Criados esses arquivos de configuração a estrutura ficou organizada como apresentado na Figura 4.7. O processo completo de preparação e execução pode ser visto no Apêndice A.

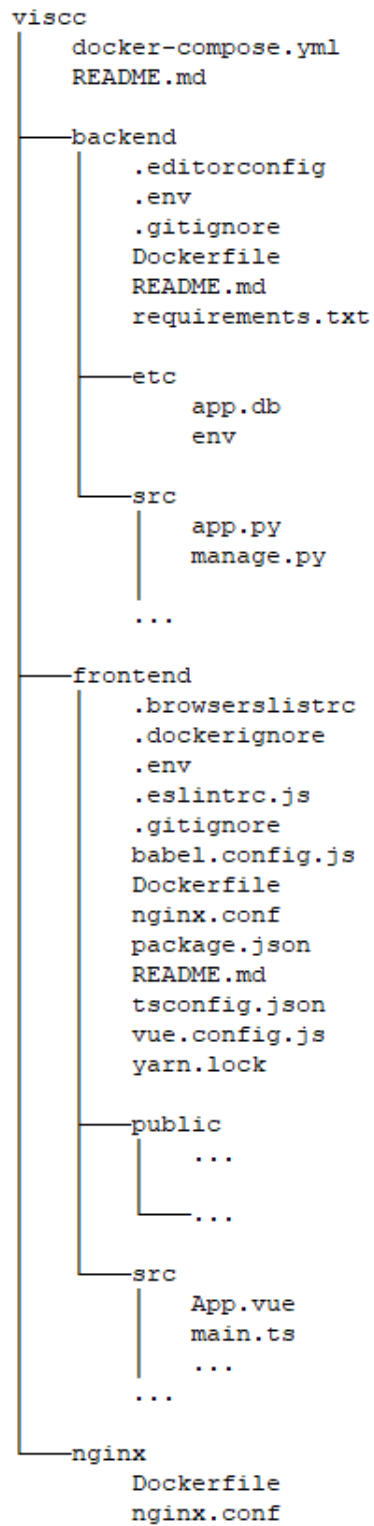


Figura 4.7: Árvore da estrutura da aplicação depois de preparada para ser executada como microsserviços.

4.6.2 Viabilizando a execução de um serviço LTI pioneiro na equipe

Com os arquivos preparados foi criado um projeto no repositório GitLab do *smartUnB.ECOS*, e a ferramenta de CI/CD do repositório já atuou para a construção das imagens, teste e lançamento da aplicação em um container. A construção final da aplicação levou pouco mais de dez minutos até o lançamento, como pode ser observado na Figura 4.8. Uma vez lançada a aplicação pode ser testada de maneira independente acessando através do endereço disponibilizado pelo repositório.

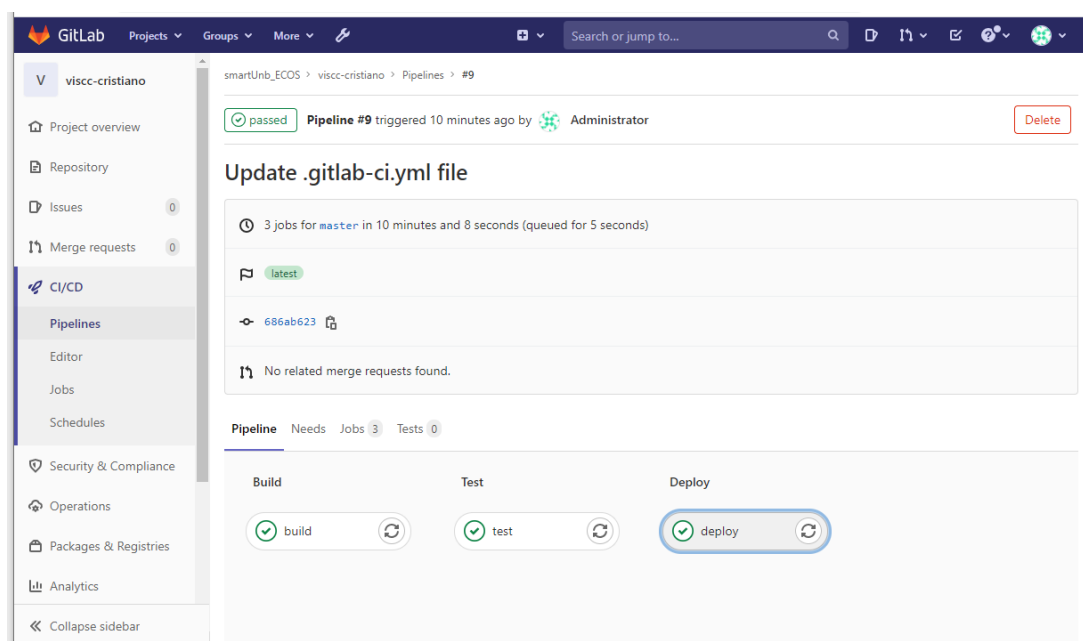


Figura 4.8: Execução do *pipeline* de lançamento do aplicação.

Com a aplicação em execução foi escolhido o AVA Moodle para testar a integração utilizando o padrão LTI. Foi executada uma instância de teste do Moodle, que pode ser encontrada no endereço "<https://sandbox.moodledemo.net/>", seguindo o procedimento documentado no manual do mesmo para a integração de uma aplicação LTI. No caso dessa integração não foi necessária nenhuma configuração adicional, além do procedimento padrão de cadastro, tanto no produtor VisCC quanto no consumidor Moodle, e na Figura 4.9 pode ser vista a integração das ferramentas. Com a aplicação em execução é possível observar também o consumo de recurso do servidor utilizando o Dashboard da plataforma Kubernetes, Figura 4.10, que ficou no patamar de 20% de um núcleo do processador, que tem quatro núcleos, e 240MiB de memória RAM.

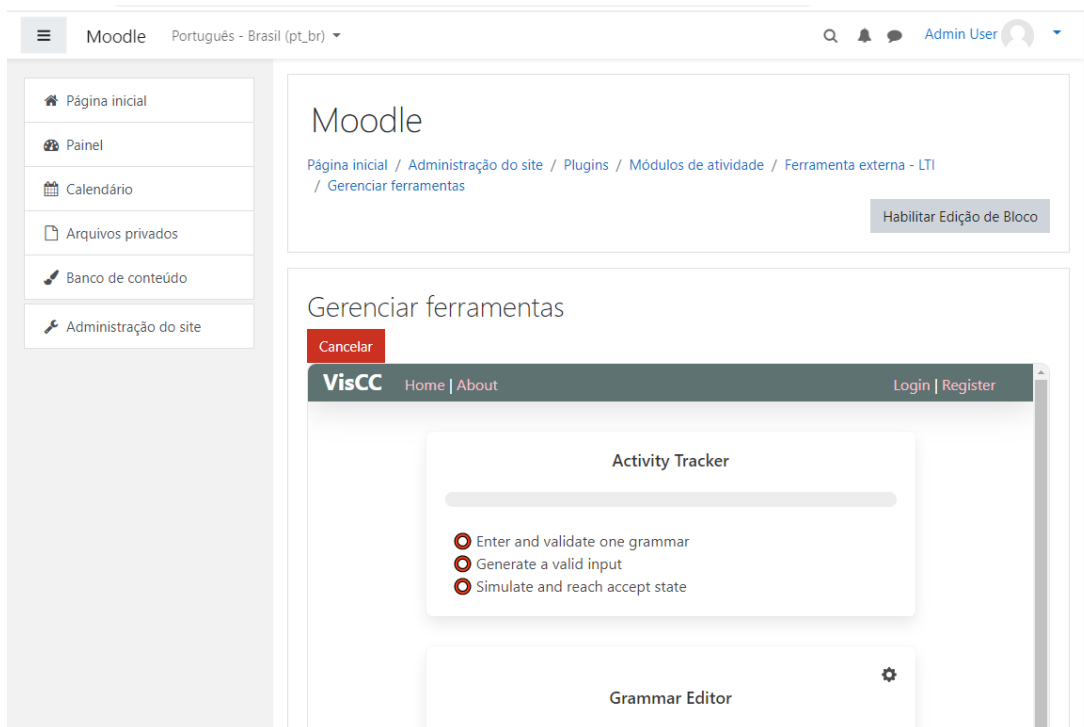


Figura 4.9: VisCC em execução dentro do AVA Moodle.

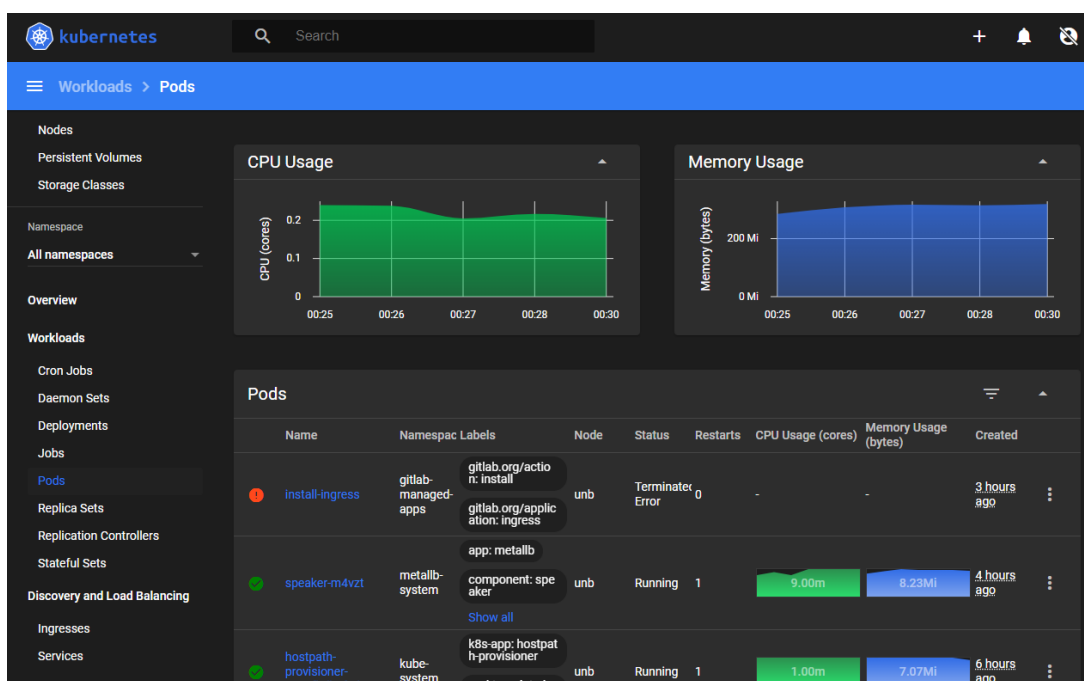


Figura 4.10: Avaliação do consumo de recurso computacional dos microsserviços no painel de controle do Kubernetes.

Capítulo 5

Conclusão

O meio em que o indivíduo é inserido tem um papel essencial no aprendizado do mesmo[17]. A natureza incremental e colaborativa de um ecossistema educacional requer um conjunto de cuidados na sua concepção e implementação para garantir que a usabilidade, o acesso, a facilidade de integração de ferramentas, o escalonamento, permitindo assim que o usuário tenha acesso às ferramentas necessárias para o seu aprendizado, mas permitindo que a manutenção e o desenvolvimento das ferramentas não sejam prejudicados pela natureza independente de seus desenvolvedores.

5.1 Contribuições

A construção de um ecossistema educacional é um trabalho colaborativo e incremental, e considerando essas características foram mapeados os requisitos, necessidades, funcionalidades e ferramentas necessárias para iniciar o desenvolvimento do *smartUnB.ECOS* com as características inicialmente definidas [1]. Com a proposta de preparar uma estrutura para que outros desenvolvedores trabalhem e integrem ferramentas no ecossistema que será utilizado para outros estudos e pesquisas, o trabalho abordou como as escolhas de ferramentas e padrões tem impacto no resultado da ferramenta para os usuários.

Para atender a necessidade de um desenvolvimento centralizada em um servidor, permitindo o acesso dos desenvolvedores, mas garantindo a segurança e o controle de acesso as ferramentas, foi escolhida uma estrutura que permita ao desenvolvedor o acesso a um repositório de código com integração a uma plataforma Kubernetes que permite executar as aplicações que estão sendo desenvolvidas no servidor sem acesso direto ao mesmo, em uma estrutura de microsserviço.

Com a ferramenta LTI escolhida, foi identificada a necessidade da implantação e configuração do servidor OAuth2. Também foram disponibilizados no repositório os templates do padrão LTI que podem ser usados na construção e integração das aplicações.

Serviço	Usuário	Status	Consumidor potencial
Aprendizagem de Análise Sintática Descendente em Compiladores	Discente	Versão 1 finalizada [Cardoso,2020]	LMS
Aprendizagem de Análise Sintática Ascendente em Compiladores	Discente	Iniciado	LMS
Atribuição, apreciação e visualização de <i>feedback</i> de atividades avaliativas	Docente Discente	Em andamento	LMS
Manutenção de portfólio de competências para cursos de graduação	Docente-Admin (NDE)	Em andamento	LMS
Elaboração colaborativa de Plano de Ensino em IMS-LD	Docente	Previsto	LMS
Questionamento discente colaborativo e gamificado	Discente	Previsto (versão wiki finalizada em 2011)	LMS
Grupos informais de conversas sobre disciplinas	Discente Docente	Em andamento	Rede Social LMS

Tabela 5.1: Trabalhos previstos ou em andamento no projeto smartUnB.ECOS

Foi apontado que a certificação é dada para uma ferramenta e não para a entidade, e que para submeter a certificação é necessário ser membro do IMS, com um custo associado ao porte da entidade. Com isso ficou aberto para um possível trabalho futuro onde uma ferramenta completa seja desenvolvida a submissão para a certificação.

O servidor está disponibilizado no acesso *ecos.cic.unb.br*, inicialmente com uma interface para o repositório de códigos Gitlab. Quando as ferramentas do provedor estiverem em produção, o mesmo poderá ser apontado para uma página inicial. No Apêndice B existe um passo-a-passo para iniciar um projeto no GitLab.

5.2 Trabalhos futuros e perspectivas

Como a proposta do trabalho foi a construção da infraestrutura para a construção do provedor de serviços educacionais *smartUnB.ECOS*, com o propósito de ser um ecossistema para estudos e pesquisas, existem incontáveis iniciativas que poderão ser construídas em cima deste trabalho [1]. Um dos trabalhos iniciais é o estudo de Redes Sociais em contexto educacional, já com uma plataforma escolhida e sendo testada. Outros trabalhos de integração de ferramentas também estão sendo desenvolvidos concomitantemente com a construção da infraestrutura. Os serviços que estão previstos ou em andamento no *smartUnB.ECOS* podem ser vistos na Tabela 5.1.

No quesito de certificação e participação nas atividades da IMS Global, pode ser avaliado o impacto da participação de uma instituição brasileira no consórcio, visto que muito além do padrão LTI, o consórcio discute ações, padrões, estruturas e direcionamentos para ferramentas digitais educacionais e o seu trabalho tem impacto direto em moldar como serão as ferramentas que serão fornecidas no futuro.

Vale ainda destacar que o governo brasileiro, por meio do Fundo Nacional de Desenvolvimento da Educação (FNDE), lançou no edital de 2023 [35] do Programa Nacional de Livros Didáticos (PNLD), para processo de aquisição de obras didáticas, literárias e pedagógicas destinadas aos estudantes, professores e gestores das escolas dos anos iniciais do ensino fundamental da educação básica pública, das redes federal, estaduais, municipais e do Distrito Federal, a possibilidade da aquisição de materiais virtuais e plataformas digitais de literatura, e para isso foi lançado um Relatório recomendatório sobre oferta de livros digitais pelo PNLD [36], produzido pela Rede Nacional de Ensino e Pesquisa (RNP), que inclui entre as ferramentas recomendadas o padrão LTI como um dos padrões de integração, mostrando com isso uma tendência do crescimento da adoção dessas ferramentas no Brasil que pode ser investigada em trabalhos futuros.

No campo de perspectivas existe uma preocupação cada vez maior com a privacidade no meio digital. Dessa preocupação estão partindo iniciativas para dar controle ao usuário de seus dados pessoais, tanto no campo legislativo quanto no campo de ferramentas tecnológicas. Entre essas iniciativas destaca-se o Solid, uma especificação de armazenamento descentralizado de dados segura que permite o usuário controlar quais e quem pode acessar que tipo de dado, dando mais controle sobre a sua privacidade [37]. A iniciativa está sendo liderada pelo Professor Tim Berners-Lee, criador da rede mundial de computadores, que em suas palestras e entrevistas demonstra sua preocupação com a forma que as informações pessoais das pessoas estão sendo usadas na Internet [38] e está sendo desenvolvida dentro em laboratórios acadêmicos no MIT, Oxford University Computer Science Lab, BBC R&D, KNowledge on Web Scale, University of Oviedo, CERN e Open University. A discussão sobre privacidade e acesso digital no campo da educação está começando e terá impacto direto na forma que as ferramentas educacionais deverão evoluir e dessa forma recomenda-se a investigação sobre o padrão Solid relativamente à especificação LTI.

Referências

- [1] Fernando W. Cruz, Germana M. da Nobrega e: *Rumo a um ecossistema educacional apoiado por computador e socialização em rede descentralizada*. Anais do SBSC 2021, 2021. ix, 2, 20, 32, 33
- [2] Veiga Silva, Welington de: *Broad-ecos: Ecossistema de e-learnign baseado em serviços educacionais*. Tese de Mestrado, Universidade Federal de Juiz de Fora), 2016. ix, 1, 6, 7, 8, 20
- [3] Fontenla, Jorge, Roberto Pérez e Manuel Caeiro: *Using ims basic lti to integrate games in lmss — lessons from game×tel*. Em *2011 IEEE Global Engineering Education Conference (EDUCON)*, páginas 299–306, 2011. ix, xi, 9, 10, 14
- [4] Jurado, F. e M. Duque: *Ims-lti and web-services for integrating moodle to an eclipse-based distributed environment for learning to program*. *International Journal of Engineering Education*, 32:1007–1014, 2016. ix, xi, 11, 12, 14, 20
- [5] Alier, Marc, María José Casany, Ariadna Llorens, Jesús Alcober e Joana d’Arc Prat: *Atenea exams, an ims lti application to solve scalability problems: A study case*. *Applied Sciences*, 11(1), 2021. <https://www.mdpi.com/2076-3417/11/1/80>. ix, xi, 13, 14, 21
- [6] Matta Furniel, Ana Cristina da, Ana Paula Bernardo Mendonça e Rosane Mendes da Silva: *Recursos educacionais abertos: Conceitos e princípios*. Guia, 2019. <https://campusvirtual.fiocruz.br/portal/guiaarea/assets/files/Guia1.pdf>, acesso em 2021-05-15. ix, 20, 21
- [7] Consortium, IMS Global Learning: *Caliper metric profiles: Common explanations*, 2021. <https://www.imsglobal.org/caliper-11-metric-profiles>, acesso em 2021-03-15. ix, 22
- [8] Henrick, Gavin: *Moodle as a the central hub of learning with tools plugged in - learning tool interoperability*. 1st Moodle Research Conference (MRC2012), setembro 2012. xi, 10, 14
- [9] Vickers, S. P. e S. Booth: *Learning tools interoperability(lti): A best practice guide*. Guia, 2014. http://ltiapps.net/guide/LTI_Best_Practice.pdf, acesso em 2021-05-15. xi, 10, 14
- [10] Then, Matthias, Benjamin Wallenborn, Birgit Ianniello, Binh Vu, Michael Fuchs e Matthias Hemmje: *Innovative authoring tools for online-courses with assignments -*

- integrating heterogeneous tools of e-learning platforms into hybrid application solutions*. International Journal of Emerging Technologies in Learning (iJET), 11:12, fevereiro 2016. xi, 11, 14
- [11] Kumar, Vikas: *Creating collaborative and convenient learning environment using cloud-based moodle lms: An instructor and administrator perspective*. International Journal of Web-Based Learning and Teaching Technologies, 11:35–50, fevereiro 2016. xi, 11, 14
- [12] Then, Matthias, Ivan Martinez-Ortiz, Antonio Calvo-Morata, Benjamin Wallenborn, Baltasar Fernández-Manjón e Matthias Hemmje: *Introducing competences into lti-connections between learning management systems and gaming platforms*. junho 2017. xi, 11, 14
- [13] Baker, Ryan, Vincent Aleven, Nathan Blomberg, Miggy Andres-Bray, Jonathan Sewall, Yuan Wang e Octav Popescu: *Integrating moocs and intelligent tutoring systems: edx, gift, and ctat*. janeiro 2017. xi, 12, 14, 20
- [14] Varnum, Kenneth J.: *New Top Technologies Every Librarian Needs to Know: A LITA Guide*. American Library Association, 2019. xi, 12, 14
- [15] Consortium, IMS Global Learning: *Ims product directory*, 2021. <https://site.imsglobal.org/certifications>, acesso em 2021-03-15. xi, 9, 16
- [16] Consortium, IMS Global Learning: *Ims all members*, 2021. <https://site.imsglobal.org/membership/members>, acesso em 2021-03-15. xi, 16
- [17] Kawagoe, Akemi Leandra: *O que aprendemos em silêncio: Aprendizagem informal e ecossistemas de aprendizagem*. Tese de Mestrado, Universidade de Brasília(UnB), 2019. 1, 20, 32
- [18] Chang, E. e M. West: *Digital ecosystems a next generation of the collaborative environment*. Em *iiWAS*, 2006. 1
- [19] Santamaría-Buitrago, Frey, Javier Antonio Ballesteros-Ricaurte e Angela Gonzalez Amarillo: *Ecosystem for the deployment and management of virtual laboratories based on the standard ims lti*. Revista Facultad de Ingeniería, 28:79–99, outubro 2019. 2, 8, 20
- [20] Coutinho, Emanuel Ferreira e Carla Ilane Moreira Bezerra: *Um estudo sobre a variabilidade de aspectos dinâmicos no ecossistema de software educacional solar*. Em *Anais do I Workshop em Modelagem e Simulação de Sistemas Intensivos em Software*, páginas 59–68, Porto Alegre, RS, Brasil, 2019. SBC. <https://sol.sbc.org.br/index.php/mssis/article/view/7560>. 7
- [21] IBGE: *Pesquisa nacional por amostra de domicílios contínua - pnad contínua - acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal 2019*. Pesquisa, IBGE, 2019. 8
- [22] NIC.BR|CETIC.BR: *Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras - tic educação 2019*. Pesquisa, CETIC.BR, 2020. 8

- [23] Kuss, Fabiano Sardenberg: *Ecossistema educacional apoiado por computadores : um modelo para uso de novas tecnologias no processo de ensino e aprendizagem*. 2020. <https://www.acervodigital.ufpr.br/handle/1884/68836>. 8
- [24] Terrematte, Patrick e Joao Marcos: *Ensino de lógica através de estratégias de demonstração e refutação: A integração do tutorial interativo trylogic via ims learning tools interoperability*. novembro 2013. 8
- [25] Terrematte, Patrick e Joao Marcos: *Trylogic tutorial: an approach to learning logic by proving and refuting*. julho 2015. 8
- [26] Sá, Saulo Rufino de: *Arquitetura de comunicação entre avas e objetos de aprendizagem dinâmicos utilizando a especificação ims lti*. 2017. <https://repositorio.ufrn.br/jspui/handle/123456789/24335>. 9, 20
- [27] Moodle: *Moodle statistics*, 2021. <https://stats.moodle.org/>, acesso em 2021-03-15. 10
- [28] Consortium, IMS Global Learning: *Ims lti 1.3 and lti advantage*, 2021. <https://www.imsglobal.org/activity/learning-tools-interoperability>, acesso em 2021-03-15. 15
- [29] Consortium, IMS Global Learning: *About ims global learning consortium*, 2021. <http://www.imsglobal.org/aboutims.html>, acesso em 2021-03-15. 15
- [30] Consortium, IMS Global Learning: *Ims membership*, 2021. <https://www.imsglobal.org/imsmembership.html>, acesso em 2021-03-15. 16
- [31] Douglis, F. e J. Nieh: *Microservices and containers*. IEEE Internet Computing, 23(6):5–6, 2019. 17
- [32] Burns, Brendan, Brian Grant, David Oppenheimer, Eric Brewer e John Wilkes: *Borg, omega, and kubernetes: Lessons learned from three container-management systems over a decade*. Queue, 14(1):70–93, janeiro 2016, ISSN 1542-7730. <https://doi.org/10.1145/2898442.2898444>. 18
- [33] The Linux Foundation: *Open container initiative*, 2020. <https://www.cncf.io/>, acesso em 2021-03-04. 18
- [34] Cardoso, Cristiano: *Viscc: uma abordagem pedagógica para o ensino de tradutores*. 2020. 28
- [35] Educação, Fundo Nacional de Desenvolvimento da: *Edital pnld 2023*, 2021. <https://www.fnde.gov.br/index.php/programas/programas-do-livro/consultas/editais-programas-livro/item/14094-edital-pnld-2023>, acesso em 2021-03-15. 34
- [36] Pesquisa, Rede Nacional de Ensino e: *Relatório recomendatório sobre oferta de livros digitais pelo pnld*. Relatório, PNDE, 2020. https://www.gov.br/fnde/pt-br/aceso-a-informacao/acoes-e-programas/programas/programas-do-livro/consultas-editais/editais/edital-pnld-2023-1/PNLDDIG_Rel_de_recomendacoes_v3_2020.pdf, acesso em 2021-05-15. 34

- [37] Solid Project: *Solid: Your data, your choice. advancing web standards to empower people.* <https://solidproject.org/>, acesso em 2021-05-03. 34
- [38] Ling, Thomas: *Interview: Web inventor tim berners-lee thinks his creation is out of control. here's his plan to save it.* <https://www.sciencefocus.com/future-technology/interview-web-inventor-tim-berners-lee-thinks-his-creation-is-out-of-control-heres-his-plan-to-save-it/>, acesso em 2021-05-03. 34

Apêndice A

Configuração e execução do VisCC no servidor smartUnB.ECOS

Para a instalação do VisCC, aplicação que foi utilizada como modelo de teste deste trabalho, foi necessário construir a estrutura Docker e Docker Compose que em um processo normal de desenvolvimento de uma aplicação deveria ser acontecer junto com a construção da mesma, mas neste caso, como é a aplicação de uma ferramenta já existente, foi necessária a construção posterior, o que demandou um processo investigativo de como a aplicação está construída. Neste documento será apresentada por partes: A estruturação do Backend, a estruturação do Frontend, a configuração do meio de acesso, a construção do orquestrador Docker Compose, a submissão para o GitLab, a configuração do CI/CD e o teste e por fim a integração no Moodle Sandbox.

A.1 O *Backend*

O documento README.MD do *Backend* nos trás a direção em que devemos estruturar o requisito da aplicação para a construção do Dockerfile usado para configurar ela como microserviço.

```
### Installing

Create a new Python virtual environment, install dependencies and bootstrap
the
application

'''bash
# make sure you're in root directory
test -f app.py && echo "you're in the server-side root directory!"
# create env
```

```
python3 -m venv venv
# switch to it
source venv/bin/activate
# install deps
python3 -m pip install -r requirements.txt
'''

#### Dependencies

This project use the following packages:

- SQLAlchemy: [homepage](https://www.sqlalchemy.org/), [repository](https://github.com/sqlalchemy/sqlalchemy), License: MIT
- Flask: [homepage](https://flask.palletsprojects.com/), [repository](https://github.com/pallets/flask), License: BSD 3-Clause
```

O README.MD propõe a criação de um ambiente virtual Python onde serão instalados os requisitos para a execução da aplicação, o que para desenvolvimento em Python é uma boa prática pois evita conflitos entre bibliotecas de projetos diferentes em um ambiente, mas como a aplicação será instalada em um container estamos garantindo esse isolamento no ambiente de execução, e a não ser em casos esporádicos a utilização do ambiente virtual é desaconselhada por causar *overhead* que pode causar falhas de execução no container. Como imagem base do container foi escolhida a imagem "python:3-slim" disponível no DockerHub por ser uma imagem mais limpa, contendo apenas o necessário para a execução do Python, pois a imagem completa "python:3" acaba ocupando mais espaço com bibliotecas que não precisaremos. O arquivo requirements.txt contém as bibliotecas Python necessárias para a execução do Backend:

```
#requirements.txt

# serve our website with Flask
Flask==1.1.2
# ORM with sqlalchemy
SQLAlchemy==1.3.20
# quick-cli framework
click==7.1.2
# encryption package
py-bcrypt==0.4
# json web token
PyJWT==1.7.1
```

Esse arquivo é passado dentro do Dockerfile para o Python Pip, aplicação que gerencia a instalação e manutenção de bibliotecas Python com a instrução:

```
RUN python3 -m pip install --no-cache-dir -r requirements.txt
```

Na primeira execução do Docker foi constatada que o "py-bcrypt" é uma biblioteca abandonada, que foi substituída pela biblioteca "bcrypt". Como o propósito do trabalho não é dar manutenção na aplicação VisCC, foi optado pela instalação dos complementos que permitirão a execução do mesmo da forma que foi construído. Para isso foram instaladas algumas bibliotecas pendentes utilizando a instrução Debian "apt-get". Também foi necessária a migração do arquivo ".venv" para a pasta onde está o "app.py" que será executado. Esses ajustes mapeados foi construído o Dockerfile do *Backend*, que foi colocado na raiz da aplicação e ficou com a seguinte configuração:

```
FROM python:3-slim
RUN apt-get update \
    && apt-get install -y python3-dev build-essential libc6-dev && \
    rm -rf /var/lib/apt/lists/*

WORKDIR /usr/src/app
RUN mkdir backend

WORKDIR /usr/src/app/backend

COPY ./requirements.txt ./

RUN python3 -m pip install --no-cache-dir -r requirements.txt

COPY . .
COPY ./env ./src/

WORKDIR /usr/src/app/backend/src/

CMD [ "python3", "./app.py" ]

EXPOSE 8081
```

Ao executar o container no Docker local é possível constatar que a aplicação é construída corretamente e está atuando no endereço local <http://172.17.0.2:8081/>

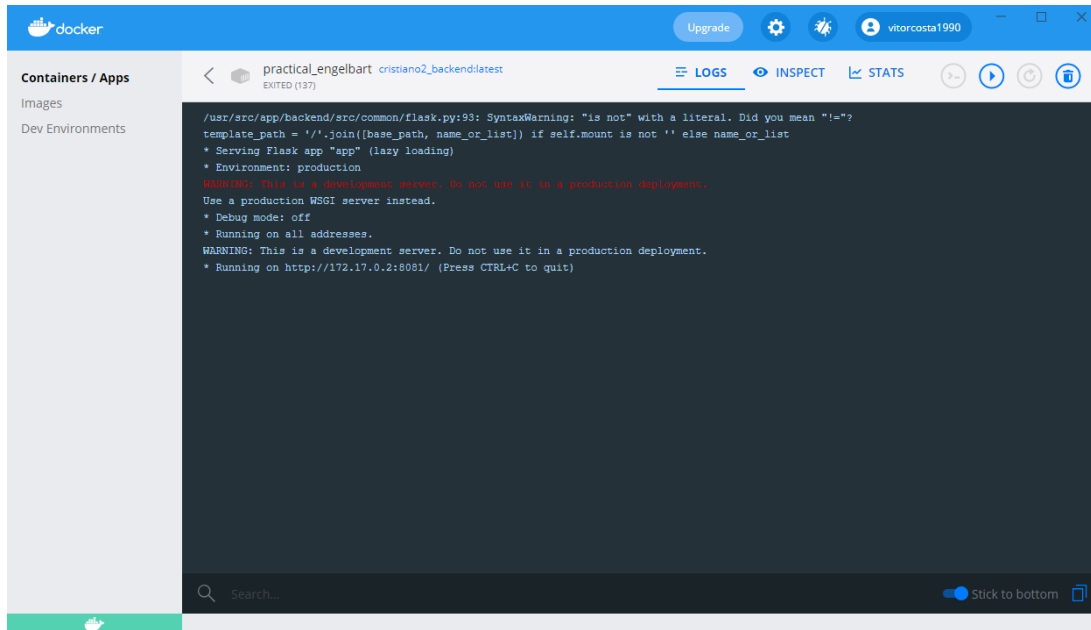


Figura A.1: Execução local do Backend.

A.2 O *Frontend*

O documento README.MD do *Frontend* nos trás a direção em que devemos estruturar o requisito da aplicação para a construção do Dockerfile usado para configurar ela como microsserviço.

```
# viscc-vue2-ts

## Project setup
'''
yarn install
'''

### Compiles and hot-reloads for development
'''
yarn serve
'''

### Compiles and minifies for production
'''
yarn build
'''

### Lints and fixes files
```



```
“““
yarn lint
“““

### Customize configuration
See [Configuration Reference](https://cli.vuejs.org/config/).
```

Pela estrutura do documento podemos ver que é uma aplicação em Vue.js, um (framework) construído em Node.js que utiliza o gerenciador de pacotes Yarn. Com isso foi escolhida a imagem "node:16-slim" para o nosso Dockerfile, foi instalado o "vue/cli-service", e foi executado a instalação conforme a instrução, do README.MD. Embora o Node.js permita a execução local para teste utilizando o "yarn serve", a própria documentação avisa que ele não deve ser utilizado para a aplicação final por questão de segurança. Dessa forma após a construção da aplicação foi criada uma nova fase no Dockerfile para servir a aplicação web. Para isso foi utilizada a imagem "nginx", um servidor web leve muito bem documentado. A estrutura final do Dockerfile pode ser vista a baixo:

```
FROM node:16-slim as build-stage

WORKDIR /usr/src/app/

COPY ./package.json ./package.json
COPY ./yarn.lock ./yarn.lock

RUN ["yarn", "install"]
RUN npm install -g @vue/cli-service

COPY . .
RUN ["yarn", "build"]

FROM nginx as production-stage
RUN mkdir /app
COPY --from=build-stage /usr/src/app/dist /app

COPY nginx.conf /etc/nginx/nginx.conf
```

O nginx foi configurado para expor a porta interna 80 e pode ser vista a execução local na o endereço *http://172.17.0.2:80/*

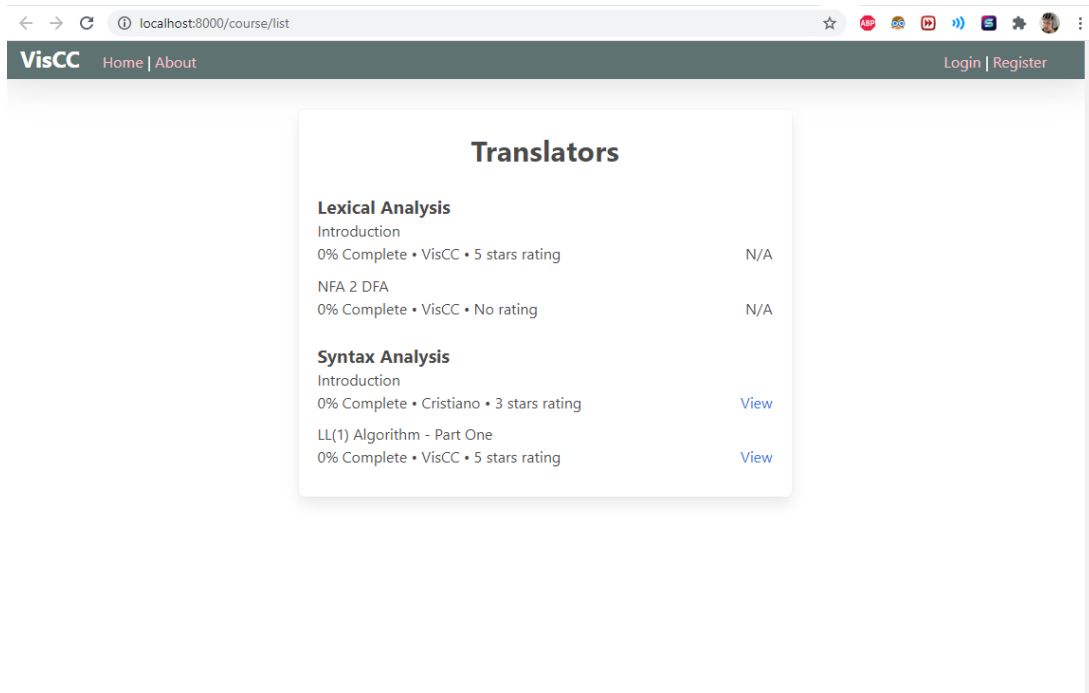


Figura A.2: Execução local do Frontend.

A.3 Meio de acesso

No caso de uma execução local tanto o Frontend quanto o Backend estão servidos no "localhost" e portanto tem acesso um ao outro diretamente. Quando a aplicação for executada no Kubernetes não haverá mais essa exposição, portanto na hora de construir a aplicação foi necessária a construção de uma aplicação de servidor web que terá como papel garantir que todos consigam se falar. Para isso foi escolhida novamente uma imagem "nginx" simplesmente com:

```
FROM nginx
RUN rm /etc/nginx/conf.d/default.conf
COPY nginx.conf /etc/nginx/nginx.conf
```

e no arquivo "nginx.conf" são apontadas as portas do *Frontend* e *Backend*:

```
server {
    listen 80;

    location / {
        proxy_pass http://frontend:8080;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
```

```

    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}

location /api {
    include uwsgi_params;
    uwsgi_pass backend:8081;
}

```

A.4 A construção do orquestrador

Com as três micro-aplicações construídas foi hora de unificar as três em uma aplicação só, que será posteriormente servida no Kubernetes. Para isso foi criada uma configuração do orquestrador Docker Compose "docker-compose.yml":

```

# check your version: https://docs.docker.com/compose/compose-file/
version: "3.7"

services:
# first container
  backend:
    # look for dockerfile and build the container
    build: ./backend
    container_name: backend
    restart: always
    # set environment variables
    environment:
      - APP_NAME=FlaskApp
    # mount the volumes so we can change code and hot reload
    volumes:
      - './backend:/backend'
    # expose port on the container, only to other containers
    expose:
      - 8081

  nginx:
    build: ./nginx
    container_name: nginx
    restart: always
    # port allows the host machine to link to container
    # pass 80 from host machine to 80 of the image filesystem
    ports:

```

```
    - "80:80"

frontend:
  build: ./frontend
  container_name: frontend
  restart: always
  # mount the volumes so we can change code and hot reload
  volumes:
    - './frontend:/frontend'

  # port allows the host machine to link to container
  # 8000 for vue ui, 5000 for our vue app
  ports:
    - "8000:8000"
```

Nota-se que essa solução resultará em uma situação chamada "Docker-in-Docker", onde você tem duas camadas de virtualização acontecendo uma dentro da outra. Em um cenário onde a aplicação será desenvolvida do zero uma solução mais adequada seria servir cada uma das partes no "gitlab-ci.yml", resultando em três containers separados se comunicando através da infraestrutura.

A.5 A submissão para o GitLab do smartUnB.ECOS

Para a submissão para o repositório foi utilizadas as seguintes instruções:

```
git config --global user.name "Vitor Costa"
git config --global user.email "140165584@aluno.unb.br"

cd viscc
git init
git remote add origin http://ecos.cic.unb.br/gitlab/smartunb_ecos/viscc-cristiano.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

Resultando no repositório:

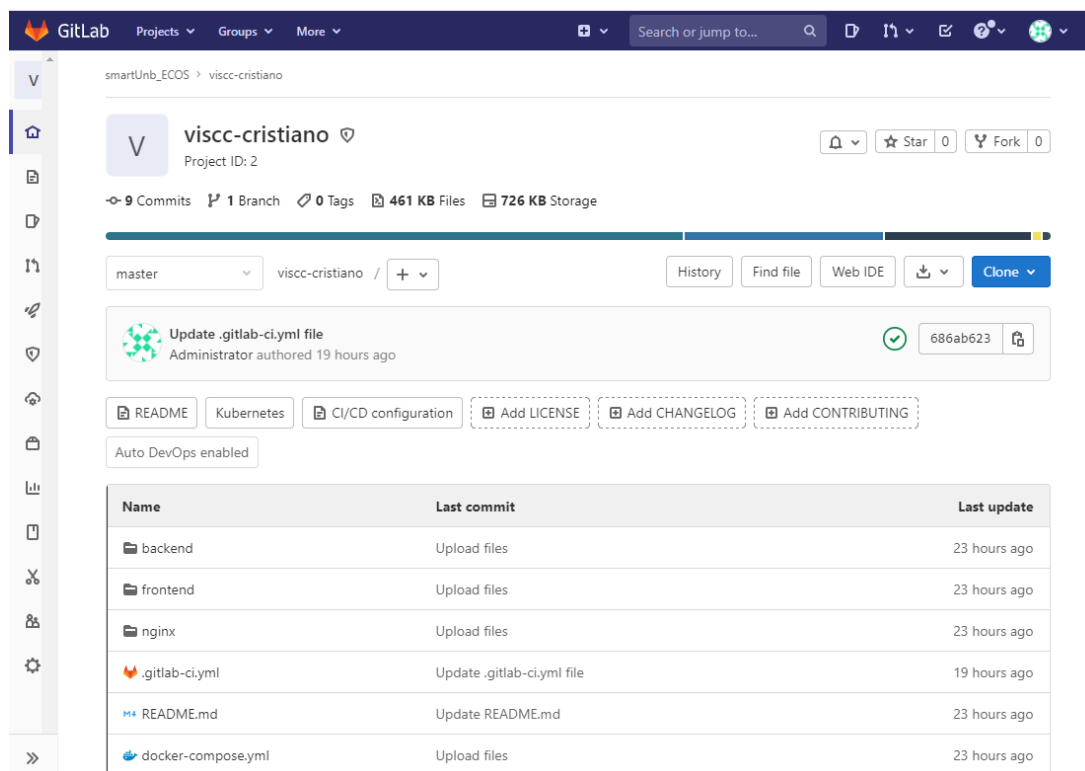


Figura A.3: Construção do repositório.

Ao entrar na aba CI/CD>Pipeline podemos ver que a execução do repositório está travada por falta de "Runner", aplicação no Kubernetes que executa as imagens das aplicações em containers. Para isso foi criado um arquivo ".gitlab-ci.yml" com os requisitos para a construção da aplicação. Como a nossa aplicação já está preparada para ser executada como uma orquestração de imagens, foi escolhido o modelo de "docker-compose", mas vale a pena destacar que o ".gitlab-ci.yml" funciona como orquestrador dentro do Kubernetes, dessa forma, na criação de uma aplicação do zero, a construção da estrutura poderia ter sido feita toda dentro do ambiente CI/CD.

```
# .gitlab-ci.yml
stages:
  - build
  - test
  - deploy

variables:
  DOCKER_HOST: tcp://docker:2375/
  DOCKER_DRIVER: overlay2

# Official docker compose image.
image:
```

```

name: docker/compose:latest

services:
  - docker:dind

before_script:
  - docker version
  - docker-compose version

build:
  stage: build
  tags:
    - cluster
  script:
    - docker-compose down
    - docker-compose build
    - docker-compose up -d

```

Com a configuração criada podemos ver que o pipeline é executado corretamente:

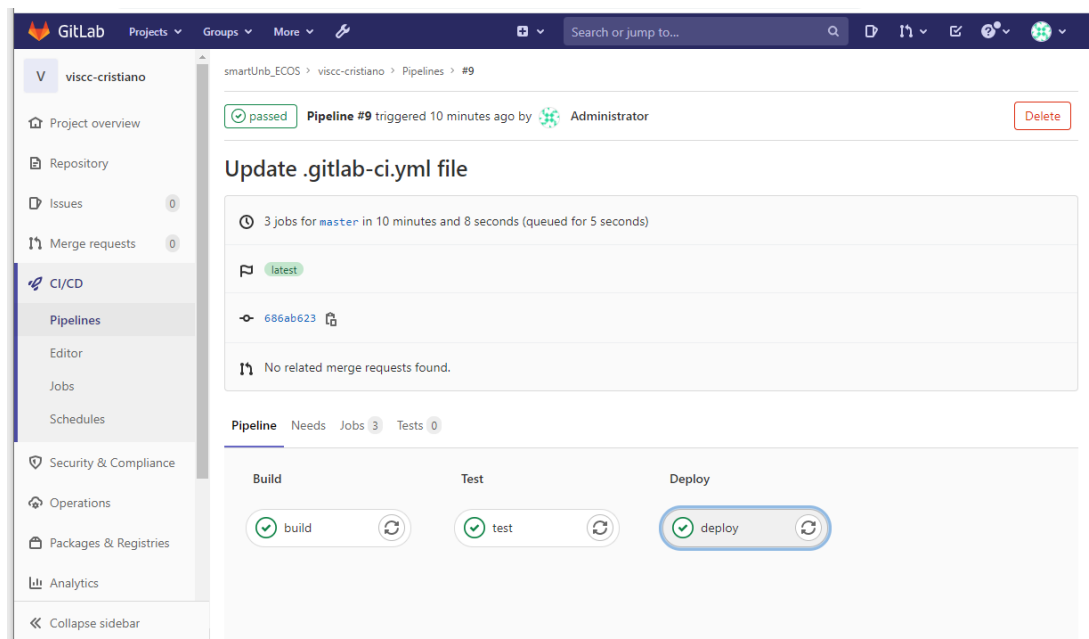


Figura A.4: Execução do *pipeline* de lançamento do aplicação.

A sua aplicação deverá estar disponível em "ecos.cic.unb.br/microk8s/[nome-do-projeto]".

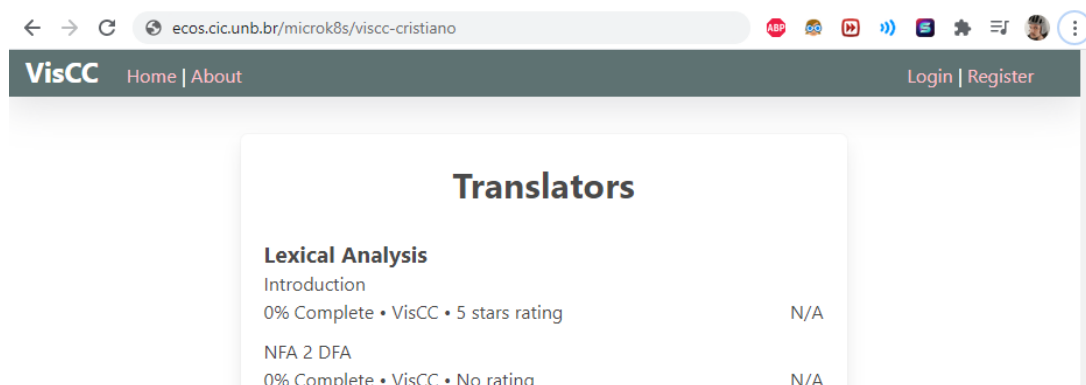


Figura A.5: Aplicação em execução no servidor.

A.6 Integração no Moodle Sandbox

O teste foi feito utilizando o Moodle Sandbox através do endereço "<https://sandbox.moodledemo.net/>". O Moodle Sandox é uma instância temporária do Moodle servida na Web e é limpa a cada uma hora, servindo adequadamente para fazer testes de aplicações e integrações, ou mesmo para explorar como o Moodle funciona. Para o nosso teste nós entramos no Moodle Sandbox e fazemos o Login como administrados com as credenciais:

Login:admin Senha:sandbox

Trocamos a linguagem para Português Brasil apenas para ficar mais claro nesta documentação. Para testar uma integração LTI nós utilizaremos o caminho Administração do site > Plugins > Módulos de atividade > Ferramenta externa - LTI > Gerenciar ferramentas. Nessa página é possível cadastrar uma aplicação LTI e ver o comportamento dela.

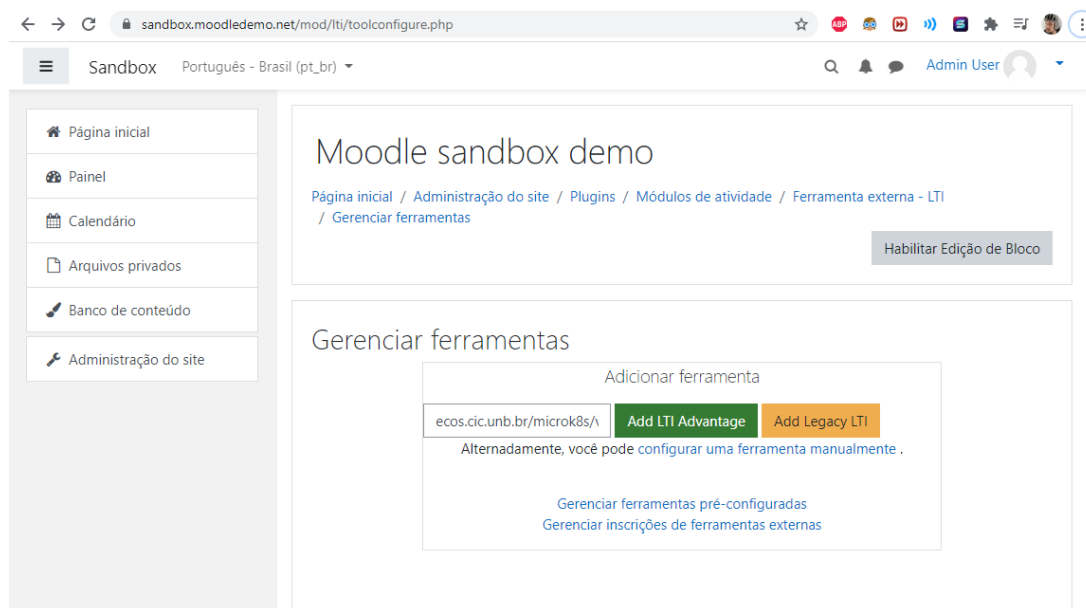


Figura A.6: Cadastro da aplicação com LTI no Moodle.

Colocamos o endereço da nossa aplicação de teste no campo URL da Ferramenta e apertamos o botão "Add LTI Advantage". Imediatamente vemos a execução local da nossa aplicação.

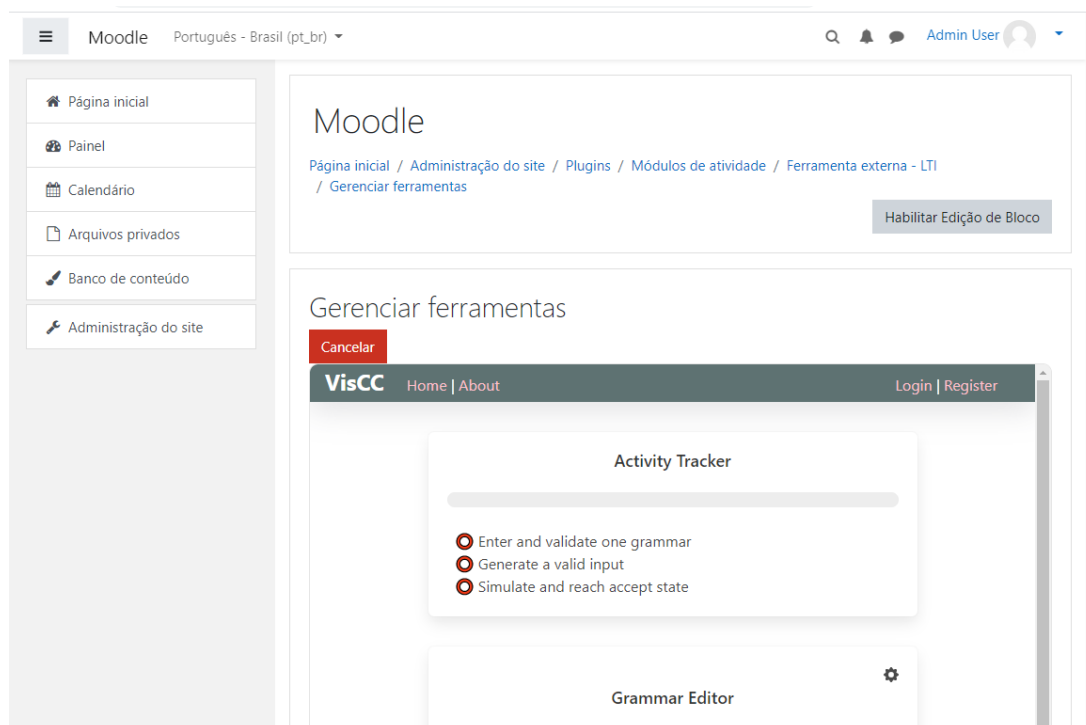


Figura A.7: Aplicação integrada ao Moodle Sandbox.

Com a aplicação funcionando podemos cadastrar ela no Moodle para poder ser servida dentro de Cursos através do botão "Configurar ferramenta manualmente", onde é necessário preencher os dados da ferramenta que você está adicionando, como Nome, URL, Descrição, Versão do LTI e outros dados. Esse preenchimento dependerá do tipo de serviço provedor que você está adicionando.

Apêndice B

Como começar um novo projeto

No processo de desenvolvimento de uma nova aplicação dentro do ambiente do *smarUnB.ECOS* é proposto a utilização da metodologia *DevOps* com o processo de Integração Contínua (*Continuous Integration - CI*). Isso porque o servidor está preparado para a execução do código desenvolvido dentro do repositório de códigos. Como o repositório escolhido foi o GitLab, existe uma documentação completa no GitLab Doc (<https://docs.gitlab.com/ee/ci/>), mas neste apêndice vai ser apresentado um guia de inicialização rápida.

B.1 Criação do projeto

Para a criação do projeto faça o login no GitLab do smarUnB.ECOS com a credencial gerada pelo seu orientador. Selecione novo projeto e escolha projeto em branco. Selecione um nome para o projeto e coloque uma descrição. Inicialmente o projeto poderá ser privado, mas é recomendado que em algum momento do desenvolvimento ele seja migrado para interno ou público (O GitLab no SmarUnB está configurado para acesso exclusivo com login, então as duas opções tem o mesmo efeito).

GitLab Projetos Grupos Mais Pesquise ou pule para...

Novo projeto > Create blank project

Create blank project
Create a blank project to house your files, plan your work, and collaborate on code, among other things.

Nome do projeto
Teste1

URL do projeto Slug do projeto

Want to house several dependent projects under the same namespace? [Create a group.](#)

Project description (optional)
Descrição do projeto

Visibility Level ⓘ

- Privado
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.
- Interno
The project can be accessed by any logged in user except external users.
- Público
O projeto pode ser acessado sem a necessidade de autenticação.

Initialize repository with a README
Permite que você clone imediatamente o repositório deste projeto. Pule esta etapa se você planeja fazer o push de um repositório existente.

Figura B.1: Criação do projeto.

Uma vez criado o projeto o GitLab apresenta um conjunto de exemplos de instruções para a sincronização da sua pasta local de projeto com a do repositório.

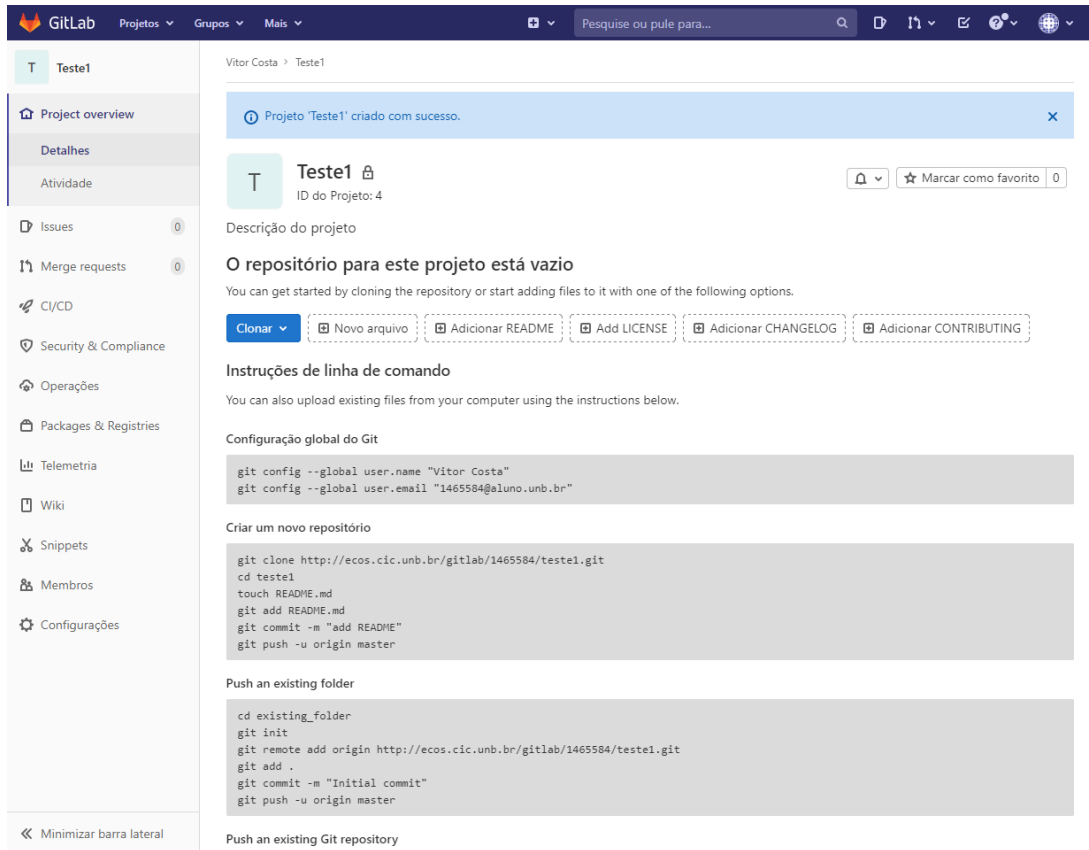


Figura B.2: Página do projeto.

Crie uma pasta local no seu computador para o desenvolvimento do projeto e crie os arquivos iniciais para a pasta. Como exemplo será criado um arquivo "index.html" simples.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Teste</title>
  </head>
  <body>

    <h1>Hello World!</h1>

    <p>
      Teste1.
    </p>
  </body>
</html>
```

um arquivo "README.MD"que deverá conter uma descrição breve do projeto e as instruções para executa-lo.

```
Arquivo README.md inicial. Devera ser atualizado com informacoes do projeto
```

E um arquivo ".gitlab-ci.yml"com instruções para o GitLab de como construir o projeto. O arquivo ".gitlab-ci.yml"tem funcionamento semelhante a um arquivo Dockerfile em uma execução local de um container Docker. Ele deve conter uma imagem inicial do container e as instruções para a construção do projeto.

```
image: alpine:latest

pages:
  stage: deploy
  script:
  - echo 'Nothing to do...'
  artifacts:
    paths:
    - public
  only:
  - master
```

Uma vez criados você deverá fazer o lançamento inicial do projeto no GitLab. Você poderá seguir as instruções da página inicial do projeto. Note que você deverá ter a aplicação Git instalada na máquina.

```
> git config --global user.name "Usuario"
> git config --global user.email "xxx@aluno.unb.br"

> git init
Initialized empty Git repository in C:/xxx/teste1/.git/

> git remote add origin http://ecos.cic.unb.br/gitlab/usuario/teste1.git
> git add .

> git commit -m "Lancamento inicial"
[master (root-commit) c0782bb] Lancamento inicial
3 files changed, 27 insertions(+)
create mode 100644 .gitlab-ci.yml
create mode 100644 README.md
create mode 100644 index.html

> git push -u origin master
Enumerating objects: 5, done.
```

```
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 613 bytes | 306.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To http://ecos.cic.unb.br:8080/gitlab/usuario/teste1.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

E com isso o projeto já está no GitLab.

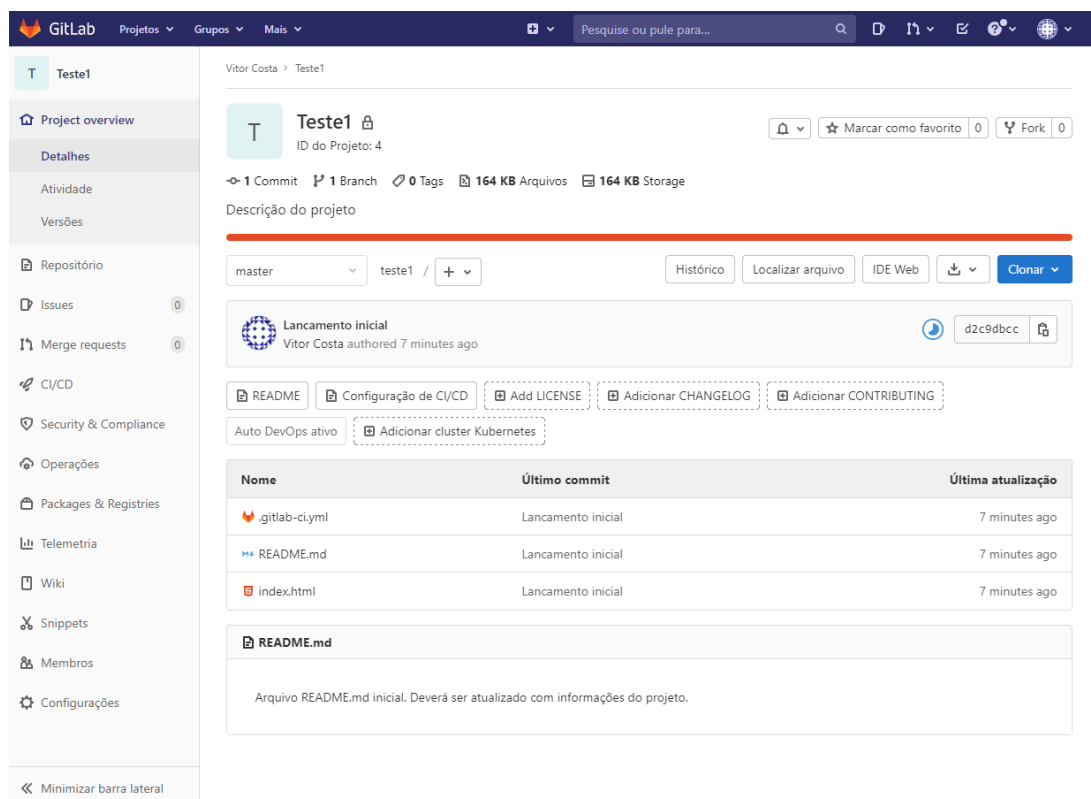


Figura B.3: Projeto disponível.

A partir deste ponto pode ser iniciado o desenvolvimento.