



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIAS E SAÚDE  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

Roberta Silvano Pereira

**Desenvolvimento da camada três do padrão IEEE 1876-2019 para um Laboratório Remoto de ensino em FPGA com suporte à aprendizagem pelo Moodle, utilizando *Learning Tools Interoperability***

Araranguá  
2022

Roberta Silvano Pereira

**Desenvolvimento da camada três do padrão IEEE 1876-2019 para um Laboratório Remoto de ensino em FPGA com suporte à aprendizagem pelo Moodle, utilizando *Learning Tools Interoperability***

Trabalho de Conclusão de Curso submetida ao Curso de Graduação em Engenharia de Computação da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Engenharia de Computação.  
Orientador: Prof. Antônio Carlos Sobieranski, Dr.  
Coorientador: Prof. Marcelo Daniel Berejuck, Dr.

Araranguá  
2022

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silvano Pereira, Roberta

Desenvolvimento da camada três do padrão IEEE 1876-2019 para um Laboratório Remoto de ensino em FPGA com suporte à aprendizagem pelo Moodle, utilizando Learning Tools Interoperability / Roberta Silvano Pereira ; orientador, Antônio Carlos Sobieranski, coorientador, Marcelo Daniel Berejuck, 2022.

30 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Campus Araranguá,  
Graduação em Engenharia de Computação, Araranguá, 2022.

Inclui referências.

1. Engenharia de Computação. 2. Laboratório Remoto de FPGA. 3. Learning Tools Interoperability. 4. IEEE 1876 2019. 5. Learning Management System. I. Carlos Sobieranski, Antônio. II. Daniel Berejuck, Marcelo. III. Universidade Federal de Santa Catarina. Graduação em Engenharia de Computação. IV. Título.

Roberta Silvano Pereira

**Desenvolvimento da camada três do padrão IEEE 1876-2019 para um Laboratório Remoto de ensino em FPGA com suporte à aprendizagem pelo Moodle, utilizando *Learning Tools Interoperability***

O presente trabalho em nível de bacharel foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

---

Prof<sup>ª</sup>. Luciana Bolan Frigo, Dr<sup>ª</sup>.  
Avaliadora  
Universidade Federal de Santa Catarina

---

Emílio Werner, Bel.  
Avaliador  
Universidade Federal de Santa Catarina

---

Prof. Anderson Luiz Fernandes Perez, Dr.  
Avaliador Suplente  
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão de curso que foi julgado adequado para obtenção do título de Bacharel em Engenharia de Computação.

Araranguá, 25 de Julho de 2022.

---

Prof<sup>a</sup>. Analucia Schiaffino Morales, Dr<sup>a</sup>.  
Coordenadora de Curso

---

Prof. Antônio Carlos Sobieranski, Dr.  
Orientador

# Desenvolvimento da camada três do padrão IEEE 1876-2019 para um Laboratório Remoto de ensino em FPGA com suporte à aprendizagem pelo Moodle, utilizando *Learning Tools Interoperability*

Roberta Silvano Pereira\*      Antônio Carlos Sobieranski†  
Marcelo Daniel Berejuck‡

2022, Julho

## Resumo

Baixo orçamento é um problema comum enfrentado por universidades e outras Instituições de ensino profissionalizante na América do Sul. A qualidade no aprendizado de Linguagem de Descrição de Hardware, como VHDL <sup>1</sup> ou Verilog, pode ser comprometida devido à falta de placas de desenvolvimento com dispositivos lógicos programáveis tipo FPGA <sup>2</sup>, para alunos de graduação dessas instituições. Este é um problema que pode ser contornado com o uso de um laboratório remoto de *hardware*: uma solução que tem sido adotada por instituições que oferecem cursos de graduação via Educação a Distância. Neste contexto, em uma parceria entre o Laboratório de Sinais e Imagens Médicas (LSIM) da Universidade Federal de Santa Catarina e o *Laboratory at Distance (L@D)* da *Université TELUQ* (Canadá), está sendo desenvolvida uma plataforma de ensino de *hardware* remoto, baseado em FPGA, para atender esta demanda. O projeto vem sendo desenvolvido segundo o padrão IEEE SA 1876-2019, proposto pelo *Institute of Electrical and Electronics Engineers* (IEEE), que padroniza e sugere boas práticas para o desenvolvimento deste tipo de laboratório de ensino. Neste trabalho são apresentados os resultados obtidos na implementação de uma das camadas preconizadas por este padrão, a qual integra o projeto ao Ambiente de Aprendizagem Virtual Moodle, adotado por ambas as Instituições de Ensino que patrocinam o projeto, permitindo que estudantes utilizem o Laboratório Remoto de FPGA através do Moodle, garantindo estudos práticos com segurança e troca de dados.

**Palavras-chaves:** Laboratório Remoto de FPGA; Learning Tools Interoperability; IEEE 1876-2019; Learning Management System.

---

\*roberta.silvano@grad.ufsc.br

†a.sobieranski@ufsc.br

‡marcelo.berejuck@ufsc.br

<sup>1</sup> Acrônimo do inglês *VHSIC Hardware Description Language*.

<sup>2</sup> Acrônimo do inglês *Field Programmable Gate Array*.

# Layer three development of the IEEE 1876-2019 standard for an FPGA teaching Remote Laboratory with support for learning through Moodle, using Learning Tools Interoperability

Roberta Silvano Pereira<sup>‡</sup>      Antônio Carlos Sobieranski<sup>§</sup>  
Marcelo Daniel Berejuck<sup>¶</sup>

2022, Julho

## Abstract

Low budget is a common problem faced by universities and other Vocational Education Institutions in South America. The quality of learning Hardware Description Languages, such as VHDL <sup>1</sup> or Verilog, can be compromised due to the lack of development boards with FPGA-type programmable logic devices <sup>2</sup>, for undergraduate students from these institutions. This is a problem that can be overcome with the use of a remote hardware laboratory: a solution that has been adopted by institutions that offer undergraduate courses via Distance Education. In this context, in a partnership between the Laboratório de Sinais e Imagens Médicas (LSIM) of the Federal University of Santa Catarina and the Laboratory at Distance (L@D) of the Université TELUQ (Canada), a remote hardware teaching platform, based on FPGA, was developed to meet this demand. The project has been developed according to the IEEE SA 1876-2019 standard, proposed by the Institute of Electrical and Electronics Engineers (IEEE), which standardizes and suggests good practices for the development of this type of teaching laboratory. In this work, the results obtained in the implementation of one of the layers recommended by this standard are presented, which integrates the project to the Moodle Virtual Learning Environment, adopted by both Educational Institutions that sponsor the project.

**Key-words:** FPGA Remote Lab; Learning Tools Interoperability; IEEE 1876-2019; Learning Management System.

---

<sup>‡</sup>roberta.silvano@grad.ufsc.br

<sup>§</sup>a.sobieranski@ufsc.br

<sup>¶</sup>marcelo.berejuck@ufsc.br

<sup>1</sup> Acronym of *VHSIC Hardware Description Language*.

<sup>2</sup> Acronym of Field Programmable Gate Array.

## 1 Introdução

As dificuldades orçamentárias encontradas por Universidades e Instituições de Ensino Profissionalizante na América Latina dificultam o bom aprendizado dos estudantes de graduação. Segundo (REIS; MACARIO, 2020) os recursos de custeio das universidades federais no Brasil sofreram uma baixa de 27,97% comparando os anos de 2013 e 2019. Já os custos de investimento caíram 95,29% entre 2011 e 2019. Esses valores sofreram ainda mais quedas com a crise sanitária e financeira causada pelo vírus SARS-CoV-2.

Um baixo orçamento dificulta para as universidades investirem em materiais que auxiliem no desenvolvimento de seus estudantes. Um exemplo dessa problemática é o estudo da linguagem de descrição de *hardware*, como VHDL e Verilog, em cursos de tecnologia. Devido a falta de recursos, como placas de lógica programável do tipo FPGA (*Field Programmable Gate Array*), utilizadas nesta disciplina, o aprendizado pode ser comprometido. Por vezes os alunos precisam realizar experimentos práticos para melhor assimilarem os conhecimentos, porém dispõem de poucas placas disponíveis (quando existem na instituição) e só podem fazer uso delas em horário de aula, o que limita o aprendizado do aluno devido a falta de atividades extra classe.

Uma abordagem para este problema é o uso de laboratórios remotos de *hardware*, que são definidos como equipamentos de *hardware* automatizado que funcionam sob o controle de *software* especializado para oferecer acesso para executar experimentos didáticos por meio da Internet (ORDUÑA *et al.*, 2015). Ou seja, um laboratório remoto é um laboratório real, mas que é controlado e monitorado a distância (GARCIA-LORO; SAN CRISTOBAL; CASTRO, 2019). Esses laboratórios geralmente estão disponíveis para o usuário por meio de um Sistema de Gestão de Aprendizagem (LMS), como Moodle, Canvas e Blackboard, que costumam ser utilizados por instituições de ensino para disponibilizar documentos de suporte ao estudo e desenvolvimento dos estudantes, controle de presenças e notas, e permitem integração com outras ferramentas, podendo dar ao usuário acesso ao laboratório remoto. Além disso, o sistema precisa gerenciar recursos como autenticação, autorização, agendamento de uso, rastreamento das atividades do usuário, ferramentas para administração e retorno de notas (ORDUÑA *et al.*, 2013).

Como demonstrado em (VIEGAS *et al.*, 2018) o uso de laboratório remotos tem uma influência positiva e direta no aprendizado, e conseqüentemente nas notas dos estudantes, permitindo que interajam com os conceitos aprendidos em sala de aula de forma prática. Além disso, permite que as instituições reduzam seus custos com laboratórios, visto que eles podem ser compartilhados entre turmas, e até mesmo entre instituições, em tempo integral. Para (POST *et al.*, 2019) tem-se alguns outros benefícios, como a capacidade dos estudantes em definir seus horários de uso do laboratório, criando um sentimento de autonomia e aumentando a sua motivação, permitindo que os professores possam instruir seus alunos de forma mais pessoal. Um outro benefício dos laboratórios remotos é a possibilidade de ensino a distância em situações como a quarentena ocasionada pelo Covid-19.

Neste contexto, por meio de uma parceria entre o Laboratório de Sinais e Imagens Médicas (LSIM) da Universidade Federal de Santa Catarina (UFSC/Brasil) e o *Laboratory at Distance* (L@D) da *Université TÉLUQ* (Canadá) foi proposto o desenvolvimento de um Labora-



tório Remoto de FPGA em (BEREJUCK *et al.*, 2022), utilizando o padrão IEEE 1876-2019. O projeto foi então estruturado em etapas de acordo com as camadas deste padrão, que estão sendo realizadas paralelamente por diferentes participantes dos laboratórios envolvidos, como mostra a figura 1, com o projeto registrado sob o número 202020605 no Sistema Integrado de Gerenciamento de Projetos de Pesquisa e de Extensão da UFSC. Neste documento são apresentados os resultados obtidos com a implementação da camada 3 daquele padrão do IEEE para este projeto. Esta camada é responsável pela integração e comunicação entre um LMS, neste caso o Ambiente Visual de Aprendizagem conhecido como Moodle, e uma plataforma de ensino remoto de FPGA, utilizando as especificações *Learning Tools Interoperability* (LEARNING. . . , 2019), como mostra a Figura 2, onde temos a etapa de pesquisa e documentação inicial do projeto, seguido das implementações de cada especificação LTI, e por fim a validação e implementação da camada como um todo. A partir do desenvolvimento deste estudo, espera-se que os alunos possam utilizar um laboratório remoto de FPGA com segurança, disponibilidade de tempo e recursos, via Moodle. Este trabalho também tem como objetivo ser um facilitador para futuros desenvolvedores que optem por utilizar estas especificações, trazendo exemplos práticos e atuais do seu uso. Outras publicações relacionadas a este projeto estão em (WERNER *et al.*, 2021) e (WERNER *et al.*, 2022).

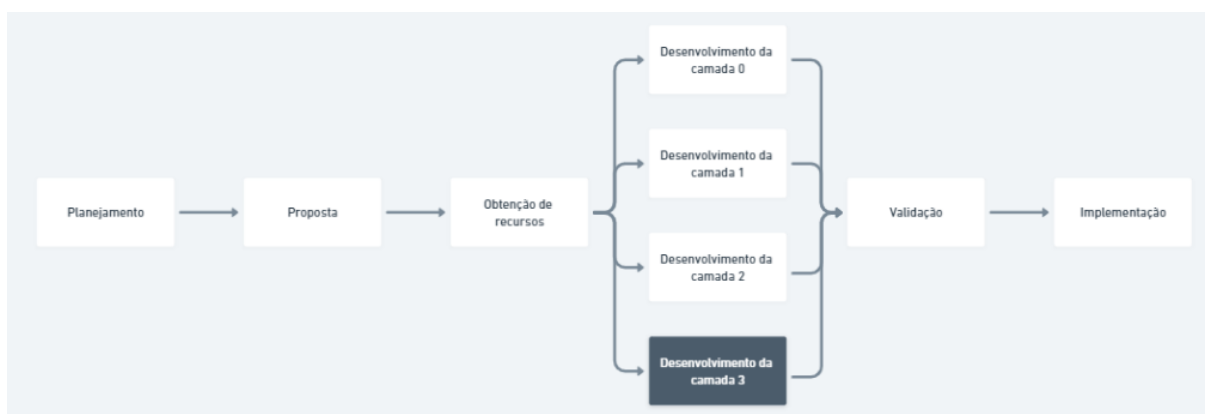


Figura 1 – Fluxo do projeto completo. fonte: autor.

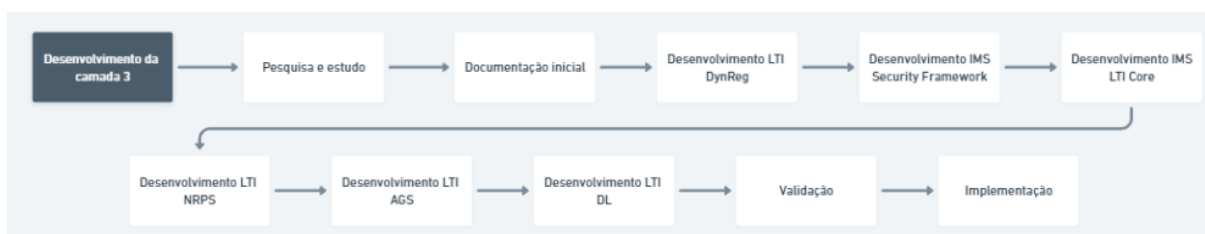


Figura 2 – Fluxo do projeto da camada 3, realizado neste trabalho. fonte: autor.

Este documento foi estruturado da seguinte forma. Na Seção 2 são apresentados os trabalhos relacionados, descrevendo a tomada de decisão por meio do processo de pesquisa, buscando estudos da área de interesse. Na Seção 3 temos a fundamentação teórica, com conceitos e termos iniciais, e uma breve apresentação das especificações utilizadas durante o projeto. Após, na Seção 4 temos a abordagem proposta e o desenvolvimento projeto. Por fim, na seção 5 temos a conclusão e trabalhos futuros.

## 2 Trabalhos relacionados

Todas as buscas da pesquisa foram realizadas na plataforma Google Scholar, considerando estudos dos sistemas Scielo, IEEE Xplore, Research Gate e Elsevier. Foi definido que o intuito inicial seria desenvolver o sistema para o LMS Moodle, por ser um sistema Open-Source, com uma grande variedade de documentações e soluções, ser utilizado na universidade onde o projeto está situado e possuir um conceito modular. Para este propósito, a busca inicial contou com uma combinação dos termos “*Remote laboratories*” e “Moodle”, filtrando a partir de 2018. Neste momento foram selecionados alguns estudos que possuíam relação com o tema proposto.

Entre os estudos encontrados nesse filtro, apenas um deles citava as diferentes tecnologias utilizadas para realizar a integração proposta. Em (RUANO *et al.*, 2020) é explicado a importância de utilizar padrões já difundidos na comunidade, apresentando as diferentes possibilidades de soluções para realizar a integração, como o uso de pacotes SCORM, *Learning Tools Interoperability* (LTI), *Easy Java/JavaScript Simulations* (EJSs), e *IEEE 1876-2019 Standard for Networked Smart Learning Objects for Online Laboratories*.

SCORM é um módulo local que realiza a comunicação entre o laboratório e o LMS, sendo portanto um sistema *client-side*, não permitindo atualizações diretas e gerando limitações quanto a segurança. Por sua vez, o uso de EJSs facilita o desenvolvimento, mas exige que a ferramenta seja desenvolvida utilizando como base a linguagem Javascript ou Java, e tem como foco apenas o LMS Moodle, limitando o projeto. Já o padrão IEEE 1876-2019 estabelece relações entre os componentes utilizados pelo laboratório e o ambiente de aprendizado, sendo utilizado o conceito de *Lab as a Service* (LaaS), mas com funcionalidades limitadas. Já de acordo com (BORJA-JIMÉNEZ *et al.*, 2019) o padrão LTI é uma alternativa segura e confiável que permite a integração de forma transparente, com troca de informações bidirecionais entre o LMS e o laboratório remoto com autenticação implícita, e que permite o escalonamento da ferramenta externa, visto que ela independe do LMS.

Após, foi realizado um aumento no período de buscas, incluindo resultados a partir de 2013, de forma a validar outros métodos para realizar a solução proposta. Tivemos (TORRE *et al.*, 2016) e (TORRE CUBILLO *et al.*, 2015) utilizando o método EJSs, (BORJA-JIMÉNEZ *et al.*, 2019), (SANTAMARÍA-BUITRAGO; BALLESTEROS-RICAURTE; GONZALEZ AMARILLO, 2019) com LTI, e (ORDUÑA *et al.*, 2015) e (ORDUÑA *et al.*, 2013) discutindo um novo método: RLMS. Essa nova proposta define um sistema gerenciador de laboratórios remotos, que funcionaria como uma ferramenta responsável por fazer a conexão entre o LMS e o laboratório, e que na verdade encapsula os métodos do LTI, facilitando e acelerando o processo de desenvolvimento, mas criando a necessidade de utilizar um sistema de terceiros e limitando a escalabilidade. Considerando que o sistema do laboratório já estava em desenvolvimento prévio, e tinha como requisito o uso da linguagem Python e do padrão IEEE 1876-2019, após pesar prós e contras de cada método foi decidido utilizar uma combinação do padrão IEEE com *Learning Tools Interoperability*. A decisão de utilizar essa combinação, e não apenas o primeiro ocorreu devido as limitações cruciais do mesmo, principalmente relacionadas aos métodos de autorização e autenticação, e quantidade de funcionalidades.

A partir disso, com o objetivo de validar e entender o uso do LTI, foi realizada uma

nova busca utilizando o termo “*Learning Tools Interoperability*”, filtrando resultados a partir de 2018. De acordo com (SANTAMARÍA-BUITRAGO; BALLESTEROS-RICAURTE; GONZALEZ AMARILLO, 2019) o LTI é um padrão que busca uma forma de instanciar uma ferramenta de conteúdos externos dentro de um LMS, por meio da troca de informações, como o nome da instituição, usuário que está utilizando a ferramenta, papel do usuário, entre outros, encapsulando estes dados por meio de autenticação e autorização. Segundo (ROY CHOUDHURY; SINGH, 2019) interoperabilidade pode ser definida como a habilidade de sistemas em diferentes redes de computadores se comunicarem, mesmo com cada um rodando em diferentes camadas de tecnologias e protocolos. Ainda segundo este estudo, o LTI é um padrão aberto, desenvolvido pela IMS Global, que provê um formato padrão para que ferramentas possam ser utilizadas dentro de múltiplas plataformas de ensino, com uma experiência totalmente integrada, sendo usada em escala global. Em 2019 havia registro de 18 plataformas de ensino (LMS) que já permitiam o uso e desenvolvimento de sistemas utilizando este padrão, e centenas de ferramentas já implementadas. Com os resultados dessa busca podemos definir que o LTI é realmente a melhor abordagem a ser utilizada, permitindo segurança e aumentando consideravelmente as funcionalidades. Na tabela 1 temos alguns detalhes sobre os trabalhos utilizados durante a pesquisa.

| <b>Estudo</b>  | <b>Contribuições</b>             | <b>Tecnologias</b>   |
|--|----------------------------------|----------------------|
| (RUANO <i>et al.</i> , 2020)   | Vantagens e métodos possíveis    | EJS, SCORM, LTI      |
| (BORJA-JIMÉNEZ <i>et al.</i> , 2019)                                 | Exemplo de integração com LTI    | LTI, Moodle e Canvas |
| (SANTAMARÍA-BUITRAGO; BALLESTEROS-RICAURTE; GONZALEZ AMARILLO, 2019) | Exemplo de integração com RLMS   | Gateway4Labs, Sakai  |
| (ROY CHOUDHURY; SINGH, 2019)   | Benefícios do LTI e dados de uso | LTI                  |
| (TORRE <i>et al.</i> , 2016)   | Exemplo de integração com EjsS   | EjsS, Moodle         |
| (TORRE CUBILLO <i>et al.</i> , 2015)                                 | Exemplo de integração com EjsS   | EjsS, Moodle         |
| (ORDUÑA <i>et al.</i> , 2015)  | Desenvolvimento do RLMS G4L      | Gateway4Labs         |
| (ORDUÑA <i>et al.</i> , 2013)  | Desenvolvimento do RLMS G4L      | Gateway4Labs         |

Tabela 1 – Trabalhos relacionados. fonte: autor.

### 3 Fundamentação teórica

Nesta seção são apresentados, de forma resumida, alguns termos técnicos utilizados ao longo do documento, como forma de equalizar o conhecimento para o bom entendimento do trabalho realizado.

#### 3.1 FPGA

Segundo (OVERVIEW... , 2008) um *Field Programmable Gate Array* (FPGA) é um dispositivo lógico, composto por uma matriz bidimensional de células lógicas genéricas que podem ser configuradas, e chaves programáveis que provêm as interconexões. A configuração dos componentes é realizada através de linguagens de síntese, como VHDL, por meio de

um arquivo *bitstream* que deve ser carregado na placa, sendo possível obter um circuito personalizado.

### 3.2 IEEE Standard for Networked Smart Learning Objects for Online Laboratories

O padrão IEEE 1876-2019 (IEEE... , 2019) estabelece métodos para armazenar, recuperar e acessar laboratórios online definindo-os como um *Lab as a Service* (LaaS), com o objetivo de facilitar o design, implementação e uso destes serviços na educação. Com o uso do paradigma *as a Service* sabe-se que o laboratório deverá ser um serviço oferecido de forma remota por meio de uma *API*, fornecendo uma interface para o usuário final interagir. A partir desta condição, são especificadas 4 camadas, como mostra a Figura 3.

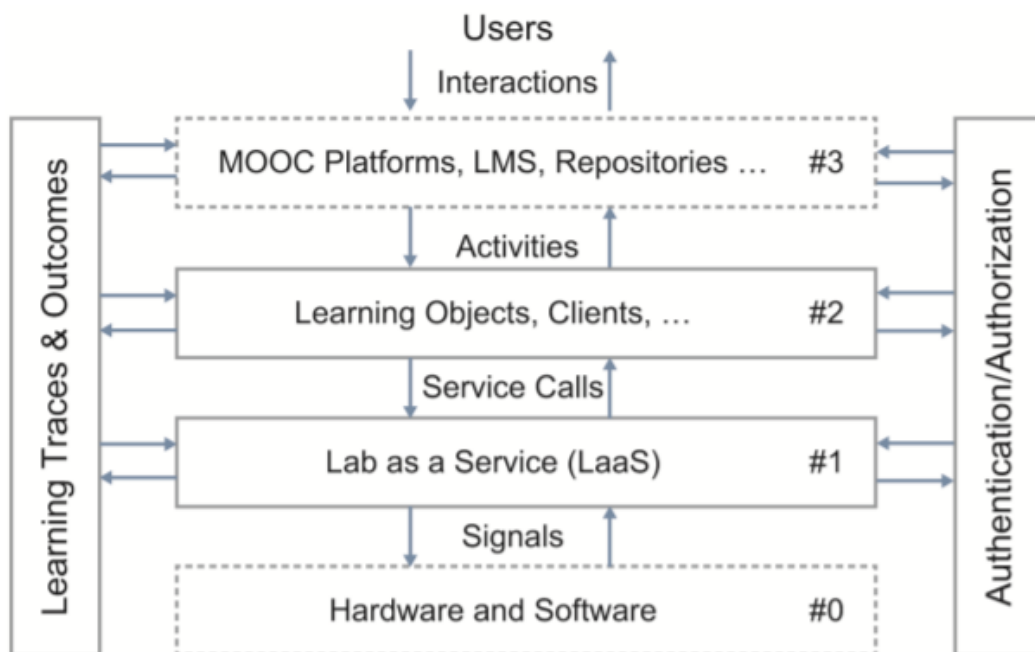
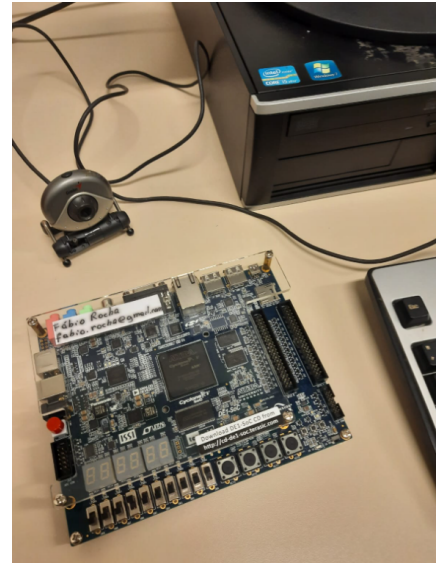


Figura 3 – Camadas do padrão IEEE 1876-2019. fonte: (IEEE... , 2019).

A camada 0 representa o *hardware* e *software* do projeto, sendo neste caso composto pela placa FPGA, *webcam* responsável por transmitir imagens e o servidor, configurados e disponibilizados no laboratório LSIM, como mostra a Figura 4. A camada 1 diz respeito a disponibilização dos serviços do laboratório, que permite que o usuário faça *upload* do *bitstream* e acompanhe a resposta da placa através das imagens fornecidas pela câmera, consulte informações dos pinos da placa, acesse o tutorial e perguntas mais comuns, e visualize quais tarefas deve executar. A camada 2 descreve o laboratório como um *Learning Object* (LO), ou seja, um objeto de aprendizagem que pode ser integrado e disponibilizado como uma atividade em um ambiente de aprendizagem, e que auxilia na educação. A camada 3, que é o foco deste estudo, é composta pela integração com os ambientes, sendo neste caso um *Learning Management System* (LMS), o Moodle.



(a) Servidor, webcam e placa FPGA



(b) Webcam e placa FPGA

Figura 4 – Camada 0. fonte: autor.

### 3.3 IMS LTI

O grupo de especificações *Learning Tools Interoperability* (LTI) foram desenvolvidas pela *IMS Global Learning Consortium*, e tem como objetivo definir um padrão de comunicação entre o LMS e uma Ferramenta Externa, permitindo o gerenciamento dos recursos do laboratório remoto, compartilhamento de informações de forma bidirecional, segurança, disponibilização de atividades específicas, envio de notas, identificação de usuários, entre outras funcionalidades. Uma das grandes vantagens deste padrão é poder ser utilizado em diversas linguagens de programação diferentes, visto que é necessário apenas implementar a validação das mensagens trocadas entre os sistemas. Este grupo é composto por uma série de padrões, detalhados na Tabela 2, e padroniza o termo Plataforma para identificar o LMS, neste caso o Moodle, e o termo Ferramenta para o sistema externo que irá implementar a especificação.

### 3.4 JSON Web Tokens - JWT

O JWT (JONES; BRADLEY; SAKIMURA, 2015) é um protocolo padrão definido no RFC 7519 para transmitir e armazenar partes de informações de forma segura, sendo portanto muito utilizado para autenticação. Esse protocolo faz parte de uma família de especificações, a *JSON Object Signing and Encryption*, que são utilizadas em conjunto. A Figura 5 mostra um JWT, que é o objeto codificado das informações, composto por três partes separadas pelo ponto. A primeira parte é chamada de *Header*, e contém informações sobre o próprio JWT, como o tipo de *token* e o algoritmo de criptografia utilizado. A segunda, o *Payload*, é composto pelos dados a serem transmitidos. Por fim, a *Signature* possui o codificador do *header*, do *payload* e a palavra chave. Dentro de cada parte do JWT temos os pedaços de informações, chamados de *Claims*. O objeto a ser codificado em JWT deve possuir o formato JSON, como mostra a Figura 6. O JWT é o objeto que será transmitido nas mensagens trocadas entre a Ferramenta e o LMS.

| Especificação                                   | Objetivo  |
|---|---|
| IMS LTI Dynamic Registration Specification      | Facilita o processo de registro entre as partes. Esta especificação ainda está disponível apenas para membros (LEARNING. . . , 2021)      |
| IMS Security Framework                          | Define um framework de segurança para todas as especificações da IMS, utilizando OAuth 2 Authorization e OpenID Connect (IMS. . . , 2021) |
| LTI v1.3 Core Specification                     | Define o ecossistema LTI, com princípios da integração entre o LMS e a Ferramenta Externa (LEARNING. . . , 2019)                          |
| LTI Names and Role Provisioning Services        | Define uma forma segura e protegida de transmitir dados sobre os usuários que utilizam o sistema (LEARNING. . . , 2019)                   |
| LTI Assignment and Grade Services Specification | Define o padrão de comunicação entre a Ferramenta Externa e o sistema de notas e atividades do LMS (LEARNING. . . , 2019)                 |
| IMS LTI Deep Linking Specification              | Permite que um LMS consiga integrar atividades específicas de uma Ferramenta Externa de maneira simplificada (IMS. . . , 2019)            |

Tabela 2 – Especificações IMS LTI

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3OTZiIiwiaWF0IjoiMTUxMjM0NTY3OTZiIiwiaXNjb2JlcnRhIjoiZmNpbHZhbm8iLCJpYXQiOiJlMTYyMzkwMjJ9.u9N-hP0-5p4l_IIaRqCV1kGcyv8IJTj29ZYkzx8BG-0
```

Figura 5 – Exemplo de JWT. fonte: autor.

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

(a) Header

```
{
  "sub": "12345",
  "name": "Roberta Silvano",
  "iat": 1516239022
}
```

(b) Payload

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
)  secret base64 encoded
```

(c) Signature

Figura 6 – Exemplos de Claims do JWT em JSON. fonte: autor.

### 3.5 OAuth 2

Desenvolvido pelo *IETF OAuth Working Group* no RFC 6749, o OAuth 2 (HARDT, 2012) é definido como o protocolo padrão da indústria para realizar autorização de aplicações que utilizem o protocolo *HTTP*. Esse padrão tem como foco principal realizar o processo de autorização, permitindo que uma aplicação consiga utilizar recursos protegidos e realizar ações em nome de outra, por meio do fornecimento de um *token* de acesso.

### 3.6 OpenID Connect

O OpenID Connect (OIDC) (OPENID. . . , 2014) é uma camada de identidade que envolve o protocolo OAuth 2, desenvolvida pela *OpenID Foundation*, tendo como foco principal a autenticação. Esse padrão permite que a aplicação que possui o *token* de acesso verifique a identidade do usuário, baseado no processo de autorização, podendo obter informações sobre o perfil, encriptar dados, gerenciar sessões, entre outros. Ao utilizar este padrão, além do *token* de acesso, a aplicação também irá receber um *token id*, que é um JWT com *Claims* de identificação do usuário, como mostra a Figura 7.

```
{
  "iss": "https://server.example.com",
  "sub": "24400320",
  "aud": "s6BhdRkqt3",
  "nonce": "n-0S6_WzA2Mj",
  "exp": 1311281970,
  "iat": 1311280970,
  "auth_time": 1311280969,
  "acr": "urn:mace:incommon:iap:silver"
}
```

Figura 7 – Exemplo de Id *Token*. fonte: (OPENID. . . , 2014).

## 4 Abordagem proposta

### 4.1 Métodos de integração

Com base no padrão do IEEE e a proposta inicial do projeto (BEREJUCK *et al.*, 2022), o desenvolvimento da camada 3 iniciou com a busca pelos possíveis métodos de integração de um laboratório remoto no Moodle. A melhor abordagem para este cenário é a funcionalidade de Ferramenta Externa da Plataforma, onde a partir do cadastro de algumas informações é possível disponibilizar um sistema para os usuários. Como veremos adiante, na seção 4.3.3, essa configuração pode ser realizada de duas maneiras: pelo administrador da Plataforma, ou por usuários que tenham privilégios específicos (sendo geralmente professores).

No primeiro caso o administrador precisará configurar a Ferramenta para todo o sistema da Plataforma, e a partir de então todos os usuários com privilégios específicos podem optar por adicionar a Ferramenta em seus ambientes, que são geralmente representados pelas disciplinas. Esta opção tem como vantagem disponibilizar a Ferramenta de forma global, entretanto é necessário contactar o administrador. Já na segunda opção um usuário que possua privilégios de adicionar atividades em um determinado ambiente irá configurar a Ferramenta de forma manual para todos os ambientes que desejar. O método a ser utilizado depende da instituição de ensino, sendo que neste trabalho foi optado pelo primeiro.

Independente do método, e utilizando apenas a especificação IEEE 1876-2019, a configuração deverá ser realizada disponibilizando o laboratório como um todo, considerando-o como um LO. A partir deste teste inicial foi possível entendermos visualmente seus resultados,

como mostra a Figura 8, onde a ferramenta é adicionada como um novo tópico de atividade em um ambiente de aprendizagem, sendo disponibilizado como um *iFrame*. Com esta primeira versão conseguimos validar que utilizando apenas este padrão já conseguimos fazer uma integração inicial entre as duas partes, porém, existem diversas limitações que só serão solucionadas com a implementação do LTI. As informações necessárias para fazer essa configuração e como obtê-las é descrito na Seção 4.3.

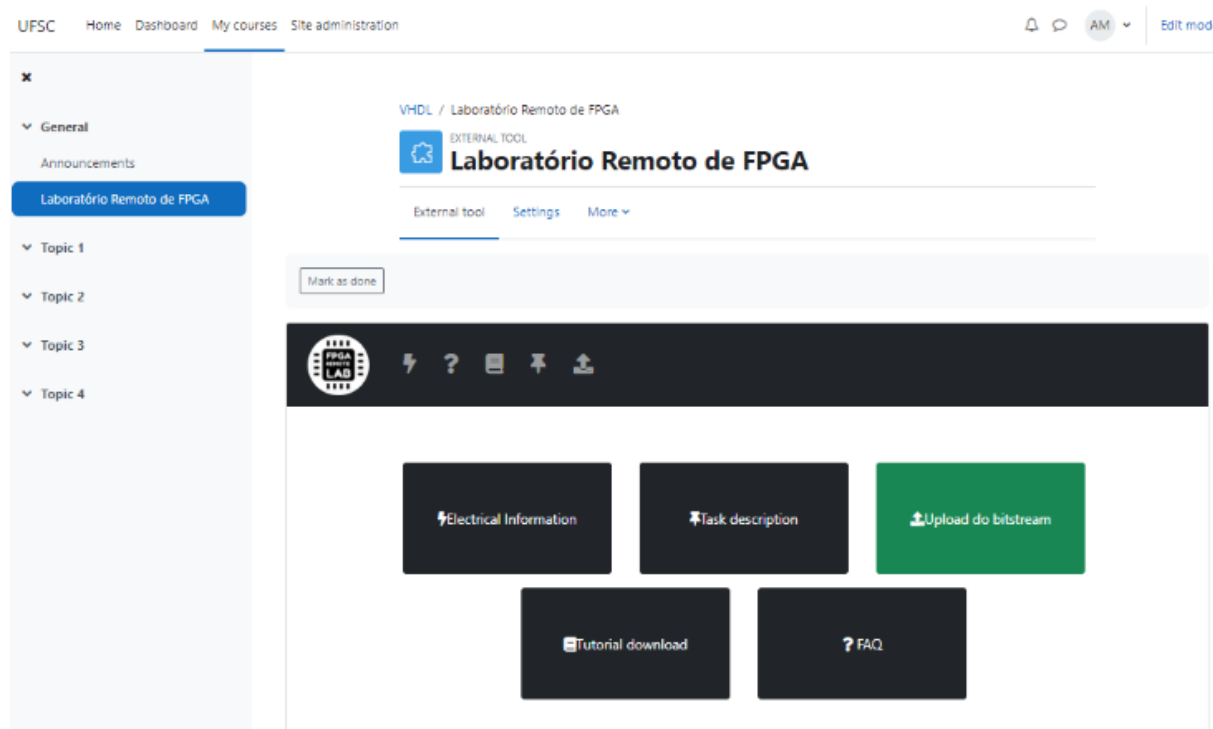


Figura 8 – Integração da Ferramenta no Moodle através da especificação IEEE. fonte: autor.

## 4.2 Arquitetura

Na figura 9 temos a arquitetura do sistema. A figura 9a mostra a estrutura geral do projeto, onde o aluno irá interagir com um sistema LMS, que irá se comunicar através das mensagens LTI implementando as medidas de segurança por meio de OAuth 2 e OpenID Connect, com o laboratório em si. Na Figura 9b temos a comunicação entre as partes do sistema, onde o usuário irá interagir com o LMS, que irá por sua vez comunicar com o servidor, que irá capturar informações da câmera e comunicar com a placa FPGA.

## 4.3 Implementação do LTI

### 4.3.1 Dynamic Registration Specification

A primeira especificação implementada durante o desenvolvimento deste sistema foi a (LEARNING. . . , 2021), que está em fase de homologação, e por enquanto disponível apenas para membros da IMS Global. Esta especificação é a mais atual entre o grupo, e portanto ainda não existem muitos sistemas implementando-a, sendo que as validações feitas para este fluxo são algumas das maiores contribuições deste trabalho. Como citado na Seção



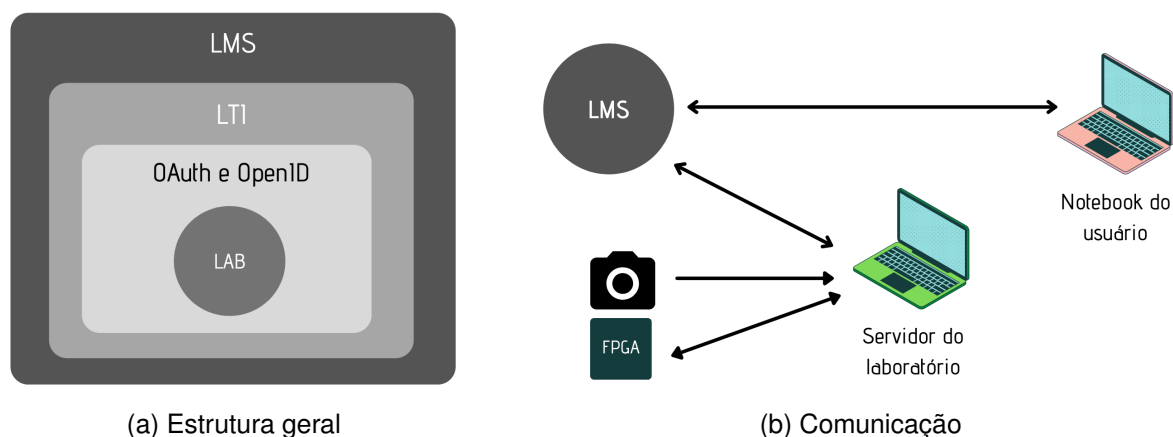


Figura 9 – Arquitetura do sistema. fonte: Laboratório LSIM.

4.1, são necessárias algumas informações para realizar a configuração de uma Ferramenta em uma Plataforma. O método para obtenção dessas informações está fora do escopo das especificações LTI, e para este trabalho optamos por disponibilizar em uma página no nosso sistema, que pode ser acessada fora de um LMS pelo usuário responsável, como mostra a figura 10.

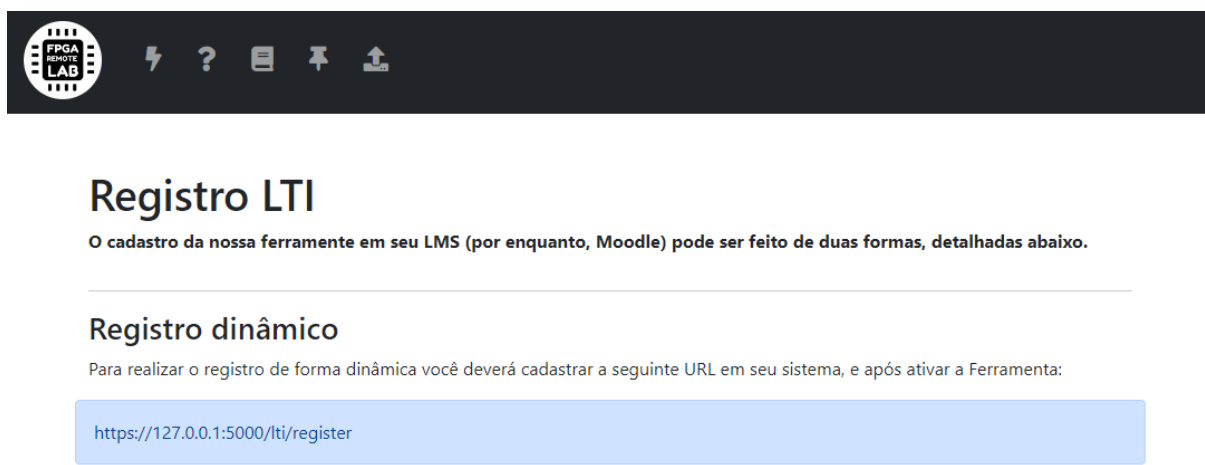


Figura 10 – Obtenção das informações para cadastro. fonte: autor.

Existem dois métodos para realizar o registro: manual, como era feito pelas especificações mais antigas, e dinâmico, como especificado neste novo fluxo. Nosso sistema disponibiliza as duas formas, mas focaremos principalmente no segundo. Para isso, o usuário responsável pelo cadastro (que pode ser o administrador e/ou o usuário com privilégios) deverá obter a URL de início de registro na página pública da Ferramenta, adicioná-la ao Moodle, como mostra a figura 11, e por fim ativá-la. Ao realizar este processo, o sistema implementado por este estudo irá acionar o fluxo desta especificação, que será apresentado mais adiante. Mas antes, é necessário definir as principais *Claims* que serão trocadas entre Ferramenta e Plataforma durante a execução deste fluxo, segundo as tabelas 3 e 4. Outras *Claims* necessárias são detalhadas na especificação.

Após definir estes valores, podemos começar a implementar o fluxo, segundo a Figura 12. Na primeira etapa temos o cenário onde o usuário responsável faz a solicitação de cadastro

## Universidade Federal de Santa Catarina

Search

### Manage tools

Add tool

Add LTI Advantage
Add Legacy LTI

Alternatively, you can [configure a tool manually](#).

[Manage preconfigured tools](#)  
[Manage external tool registrations](#)

Figura 11 – Adicionando a Ferramenta ao Moodle. fonte: autor.

| <b>Claim</b>           | <b>Informações</b>   | <b>Valor</b>   |
|------------------------|--|--|
| issuer                 | URL que endereça a Plataforma                                      | https://localhost/moodle                                 |
| authorization_endpoint | URL do <i>endpoint</i> de autorização do OAuth 2                   | https://localhost/moodle/mod/lti/auth.php                |
| registration_endpoint  | URL do <i>endpoint</i> de registro                                 | https://localhost/moodle/mod/lti/openid-registration.php |
| token_endpoint         | URL do <i>endpoint</i> para a Ferramenta solicitar um <i>token</i> | https://localhost/moodle/mod/lti/token.php               |
| jwtks_uri              | URL do <i>endpoint</i> do conjunto de chaves públicas (JWK Set)    | https://localhost/moodle/mod/lti/certs.php               |
| scopes_supported       | Identificação dos serviços suportados                              | openid (pode ter outros)                                 |

Tabela 3 – *Claims* da Plataforma para o fluxo de registro. fonte: autor.

utilizando a URL de início de registro, via a *UI* da Plataforma, abrindo a *UI* da Ferramenta em um *iFrame*. Essa ação irá acionar a etapa 2, chamando a rota de registro da Ferramenta, passando as informações da Plataforma como mostra a Figura 13. A partir do parâmetro *openid\_configuration* podemos fazer uma requisição das *Claims* da Plataforma, como mostra a Figura 14, realizando diversas validações nos valores, de forma a garantir que a conexão é segura. Na etapa 3 é realizado o processo em si, enviando as informações da Ferramenta, segundo a Figura 15, e recebendo as informações que a Plataforma gera para identificar o registro na Figura 16. Na etapa 4 é retornado para a *UI* da Plataforma o resultado do registro e o usuário pode fazer a ativação da Ferramenta.

| <b>Claim</b>       | <b>Informações</b>  | <b>Valor</b>  |
|--------------------|---|---|
| initiate_login_uri | URI utilizada para iniciar o <i>LTI Launch</i> , fazendo login  | https://127.0.0.1:5000/lti/oidc/init  |
| redirect_uris      | URI utilizada para redirecionar o <i>LTI Launch</i> após login  | https://127.0.0.1:5000/lti/oidc/ launch   |
| hwks_uri           | URL do <i>endpoint</i> do conjunto de chaves públicas (JWK Set) | https://127.0.0.1:5000/lti/.well-known/jwks.json  |
| client_name        | Nome da Ferramenta  | Laboratório Remoto de FPGA - LSIM   |
| scope              | Identificação dos serviços que a Ferramenta solicita acesso     | openid https://purl.imsglobal.org/spec/lti-nrps/scope/context<br>membership.readonly<br>https://purl.imsglobal.org/spec/lti-ags/scope/lineitem<br>https://purl.imsglobal.org/spec/lti-ags/scope/lineitem.readonly<br>https://purl.imsglobal.org/spec/lti-ags/scope/result.readonly<br>https://purl.imsglobal.org/spec/lti-ags/scope/score<br>https://purl.imsglobal.org/spec/lti-dl/claim/deep_linking_settings |

Tabela 4 – *Claims* da Ferramenta para o fluxo de registro. fonte: autor.

Todas essas informações de identificação de registro e da Plataforma precisam ser guardadas, para que a Ferramenta consiga identificar posteriormente. Portanto, durante este processo, optamos por salvar estes dados em uma base SQLite, seguindo a estrutura apresentada em 17. Com isso, temos o registro finalizado.

#### 4.3.2 IMS Security Framework

Enquanto as outras especificações da IMS definem como deve ocorrer a troca de dados de cada serviço e quais informações enviar, esta tem como objetivo principal definir os métodos de segurança e privacidade que devem ser utilizados, de forma que sejam padronizados e simplifique o desenvolvimento. Através desta especificação iremos promover autenticação, autorização, encriptação e registro dinâmico dos usuários, por meio do uso de dois padrões já consolidados do mercado: OAuth 2 e OpenID Connect.

Existem dois tipos de arquitetura de segurança, que dependem do cenário de implementação e de como é realizada a troca de mensagens. Para as especificações LTI, apesar de o usuário utilizar um *WEB browser*, a Ferramenta é externa, e portanto devemos seguir a arquitetura baseada em serviços não *web*, segundo a Figura 18.

Neste cenário um usuário irá ter a experiência do nosso sistema via *WEB*, onde ele irá interagir com a Plataforma via o *browser*, que será responsável por fazer a troca de mensagens assinadas com a Ferramenta, sendo necessário realizar autenticação. Para este contexto, o servidor de autorização pode ou não ser o mesmo sistema da Plataforma, sendo que para o Moodle será.

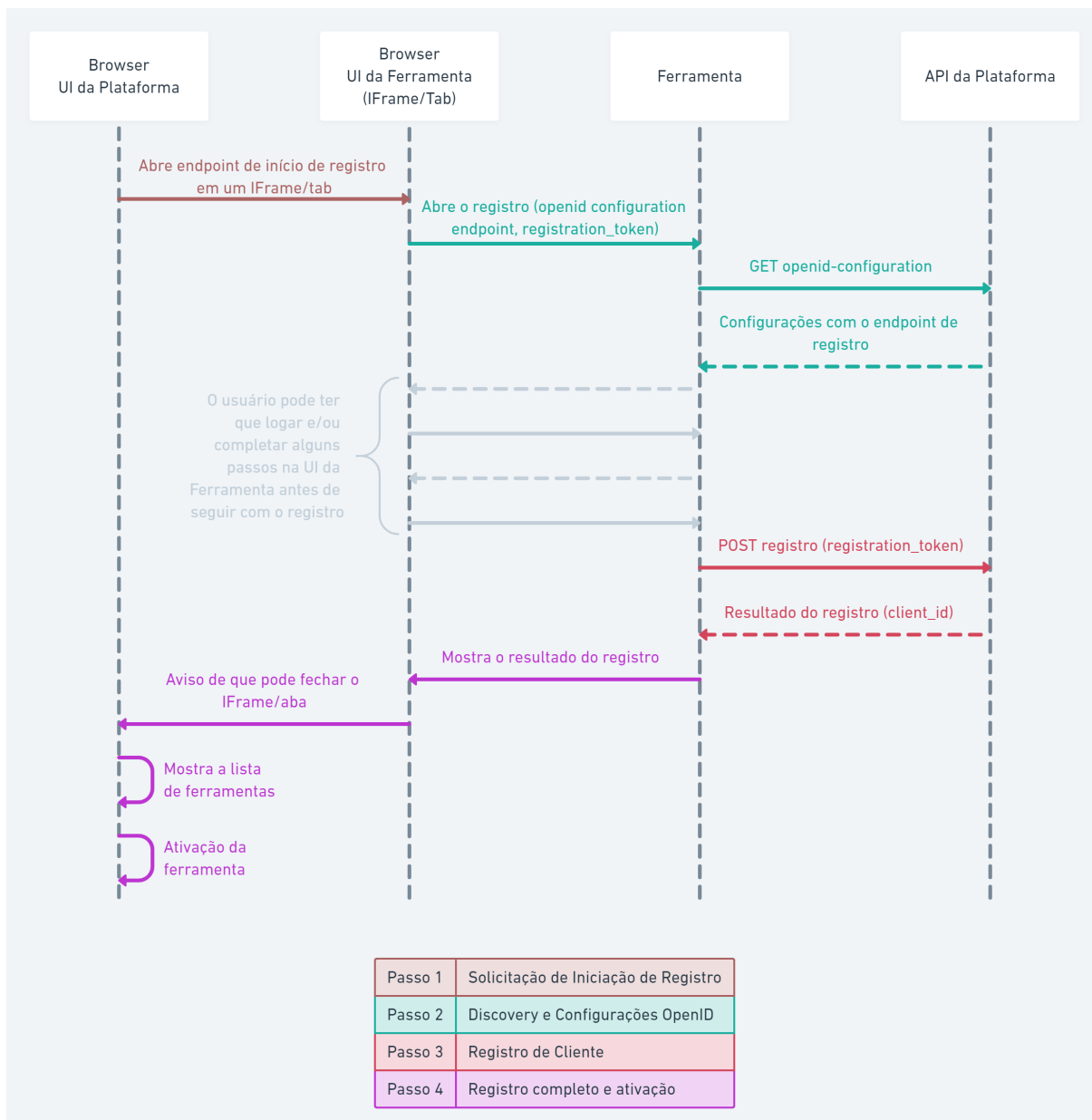


Figura 12 – Fluxo de registro dinâmico. fonte: adaptado de (LEARNING..., 2021)

Essa experiência via *browser* requer que as partes sempre enviem mensagens assinadas com sua chave privada como um JWT, e a parte que recebe a mensagem deve então verificar a assinatura com a chave pública, que foi trocada durante o processo de registro 4.3.1. A parte que estará enviando a mensagem irá depender do contexto e da especificação utilizada para aquele serviço.

As mensagens originadas pela Plataforma devem sempre enviar informações relacionadas a identidade do usuário. Isso é realizado utilizando uma adaptação do fluxo do OpenID Connect, de forma que o usuário consiga se autenticar na Ferramenta fazendo login, como mostra o diagrama da Figura 19. Desta forma, todas as mensagens serão tratadas como uma tentativa de login de terceiros.

Durante todo esse fluxo são utilizadas as informações trocadas entre a Ferramenta e a Plataforma durante o processo de registro, como especificado anteriormente nas Tabelas 3

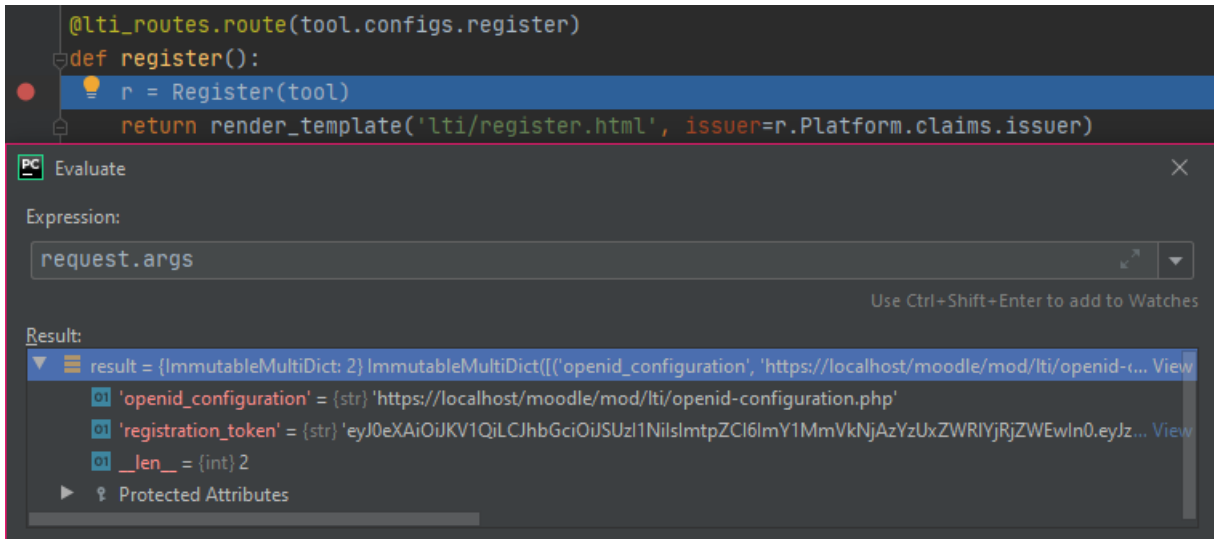


Figura 13 – Etapa 2: Abre o registro. fonte: autor.

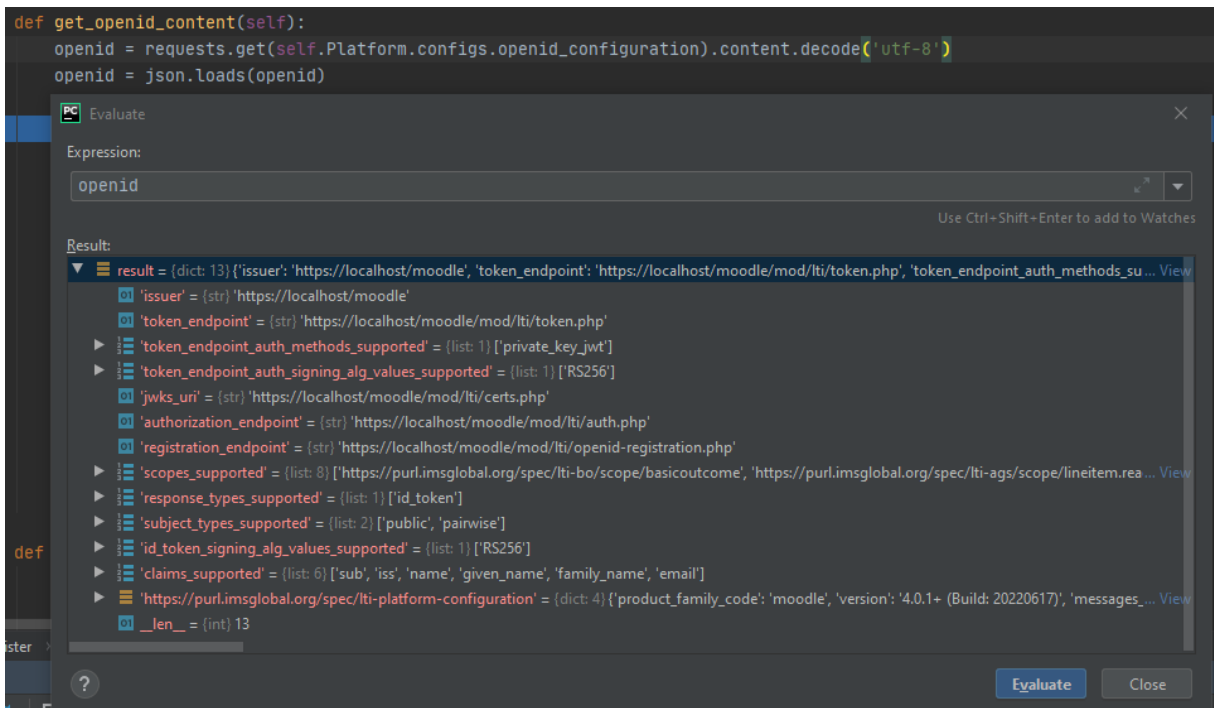


Figura 14 – Etapa 2: Configurações com o *endpoint* de registro. fonte: autor.

e 4. O fluxo inicia quando o usuário requisita acesso a recursos da Ferramenta, de forma que a Plataforma cria uma solicitação de login, redirecionando o *browser* para o *endpoint* de login iniciado por terceiros *initiate\_login\_uri*, enviando informações para identificação do contexto, e qual recurso o usuário está tentando acessar, como mostra a Figura 20. A Ferramenta irá então validar essa solicitação, incluindo alguns parâmetros necessários, de forma a requisitar a autenticação do usuário, redirecionando o *browser* para o *endpoint* de autorização da Plataforma (*authorization\_endpoint*). A Plataforma então realiza a validação dessa requisição, e retorna o *id\_token*, contendo as informações da identidade do usuário, permitindo que o login seja realizado, segundo a figura 21. Por fim, a Ferramenta exibe os recursos que haviam sido solicitados. Para a implementação desta etapa foram preparadas as rotas e informações

```
r = requests.post(self.Platform.claims.registration_endpoint, json=tool_body, headers=header) r: <Response [200]>
```

Evaluate

Expression: tool\_body

Result:

```
{dict: 15} {'application_type': 'web', 'grant_types': ['client_credentials', 'implicit'], 'response_types': ['id_token'], 'redirect_uris': ['https://127.0.0.1:5000... View
  application_type = (str) 'web'
  grant_types = (list: 2) ['client_credentials', 'implicit']
  response_types = (list: 1) ['id_token']
  redirect_uris = (list: 1) ['https://127.0.0.1:5000/lti/oidc/launch']
  initiate_login_uri = (str) 'https://127.0.0.1:5000/lti/oidc/init'
  client_name = (str) 'Laboratório Remoto de FPGA - LSIM'
  jwks_uri = (str) 'https://127.0.0.1:5000/lti/.well-known/jwks.json'
  logo_uri = (str) 'https://lsim.ufsc.br/files/2022/04/v3condensed.png'
  token_endpoint_auth_method = (str) 'private_key_jwt'
  contacts = (list: 1) ['robertaspb@gmail.com']
  client_uri = (str) 'https://lsim.ufsc.br'
  tos_uri = (NoneType) None
  policy_uri = (NoneType) None
  scope = (str) 'openid https://purl.imsglobal.org/spec/lti-nrps/scope/contextmembership.readonly https://purl.imsglobal.org/spec/lti-ags/scope/lineit... View
  https://purl.imsglobal.org/spec/lti-tool-configuration = (Dotdict: 8) {'messages': [{'type': 'LtiDeepLinkingRequest', 'target_link_uri': None, 'label': None... View
    messages = (list: 1) [{'type': 'LtiDeepLinkingRequest', 'target_link_uri': None, 'label': None, 'icon_uri': None, 'custom_parameters': None, 'placements': Non... View
      domain = (str) '127.0.0.1:5000'
      secondary_domains = (NoneType) None
      deployment_id = (NoneType) None
      target_link_uri = (str) 'https://127.0.0.1:5000/'
      custom_parameters = (NoneType) None
      description = (str) 'Aplicação desenvolvida pelo Laboratório de Sinais e Imagens Médicas da UFSC Araranguá para o aprendizado de FPGA'
    claims = (list: 5) ['iss', 'sub', 'name', 'given_name', 'family_name']
      __len__ = (int) 8
    __len__ = (int) 15
```

Figura 15 – Etapa 3: POST registro. fonte: autor.

```
registration = Dotdict(json.loads(r.text)) registration: {'client_id': 'wGkjRvg1pErjOKv', 'response_types': ['id_t...
```

Evaluate

Expression: registration

Result:

```
(Dotdict: 12) {'client_id': 'wGkjRvg1pErjOKv', 'response_types': ['id_token'], 'jwks_uri': 'https://127.0.0.1:5000/lti/.well-known/jwks.json', 'initiate_logi... View
  client_id = (str) 'wGkjRvg1pErjOKv'
  response_types = (list: 1) ['id_token']
  jwks_uri = (str) 'https://127.0.0.1:5000/lti/.well-known/jwks.json'
  initiate_login_uri = (str) 'https://127.0.0.1:5000/lti/oidc/init'
  grant_types = (list: 2) ['client_credentials', 'implicit']
  redirect_uris = (list: 1) ['https://127.0.0.1:5000/lti/oidc/launch']
  application_type = (str) 'web'
  token_endpoint_auth_method = (str) 'private_key_jwt'
  client_name = (str) 'Laboratório Remoto de FPGA - LSIM'
  logo_uri = (str) 'https://lsim.ufsc.br/files/2022/04/v3condensed.png'
  scope = (str) 'https://purl.imsglobal.org/spec/lti-ags/scope/score https://purl.imsglobal.org/spec/lti-ags/scope/result.readonly https://purl.imsglobal... View
  https://purl.imsglobal.org/spec/lti-tool-configuration = (dict: 7) {'version': '1.3.0', 'deployment_id': '29', 'target_link_uri': 'https://127.0.0.1:5000/', 'dom... View
    __len__ = (int) 12
```

Figura 16 – Etapa 3: Resultado do registro. fonte: autor.

necessárias. Porém, foi possível facilitar o desenvolvimento utilizando um módulo da linguagem, o *pylti1p3*, com algumas pequenas adaptações.

Com esse fluxo obtemos o *id token*, o JWT com informações sobre a conexão, como o nome do usuário que solicitou o recurso e qual seu cargo, de qual Plataforma foi realizada a solicitação e em qual contexto, quais *endpoints* utilizar para os outros serviços LTI, entre outros,

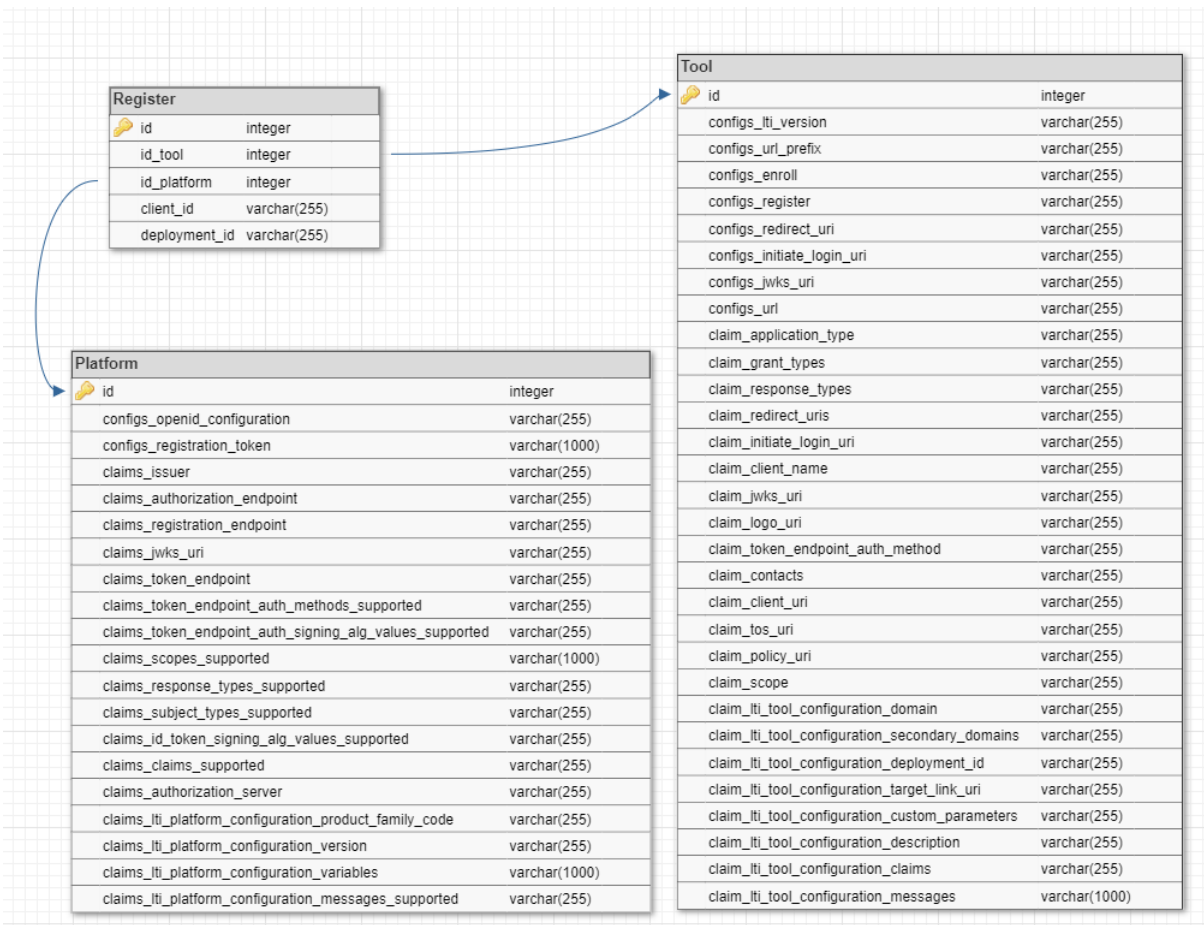


Figura 17 – Estrutura da base de dados. fonte: autor.

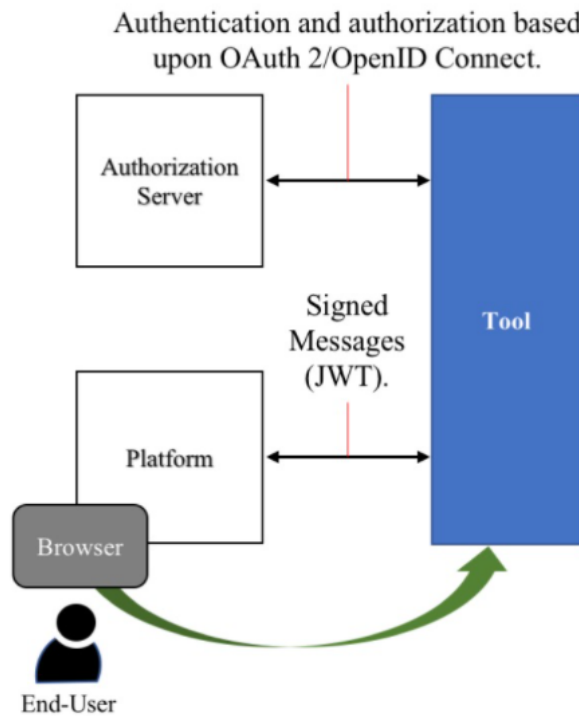


Figura 18 – Arquitetura de segurança para o LTI. fonte: (IMS... , 2021)

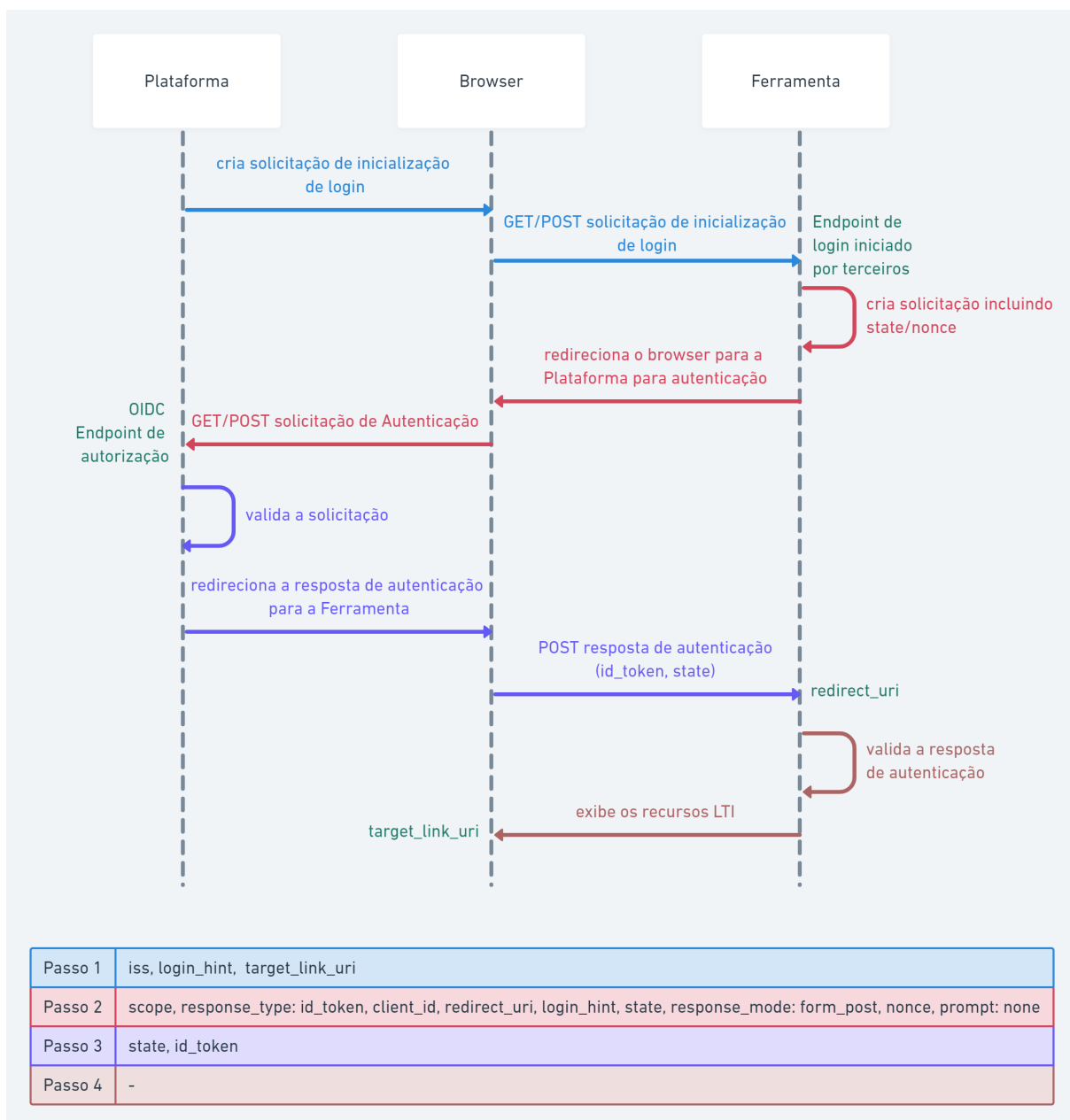


Figura 19 – Fluxo de inicialização OpenID Connect. fonte: adaptado de (IMS. . . , 2021)

e deve ser validado utilizando a chave pública da Plataforma, garantindo que a requisição é segura.

Já as mensagens originadas pela Ferramenta não possuem um usuário para autenticar, e portanto devem somente assinar as mensagens utilizando sua chave pública, informando o *client\_id* do registro.

Essa especificação também define questões relacionadas as chaves utilizadas para assinar as mensagens. Para este sistema optamos por utilizar chaves assimétricas, disponibilizadas via *Key Sets*, ou seja, um *endpoint* no sistema responsável por expor as chaves públicas, que possuem o formato JSON Web Key, como mostra a Figura 22, com as quais a Plataforma pode validar se as mensagens são originadas da Ferramenta, e permitindo que a adição de novas chaves seja dinâmica.



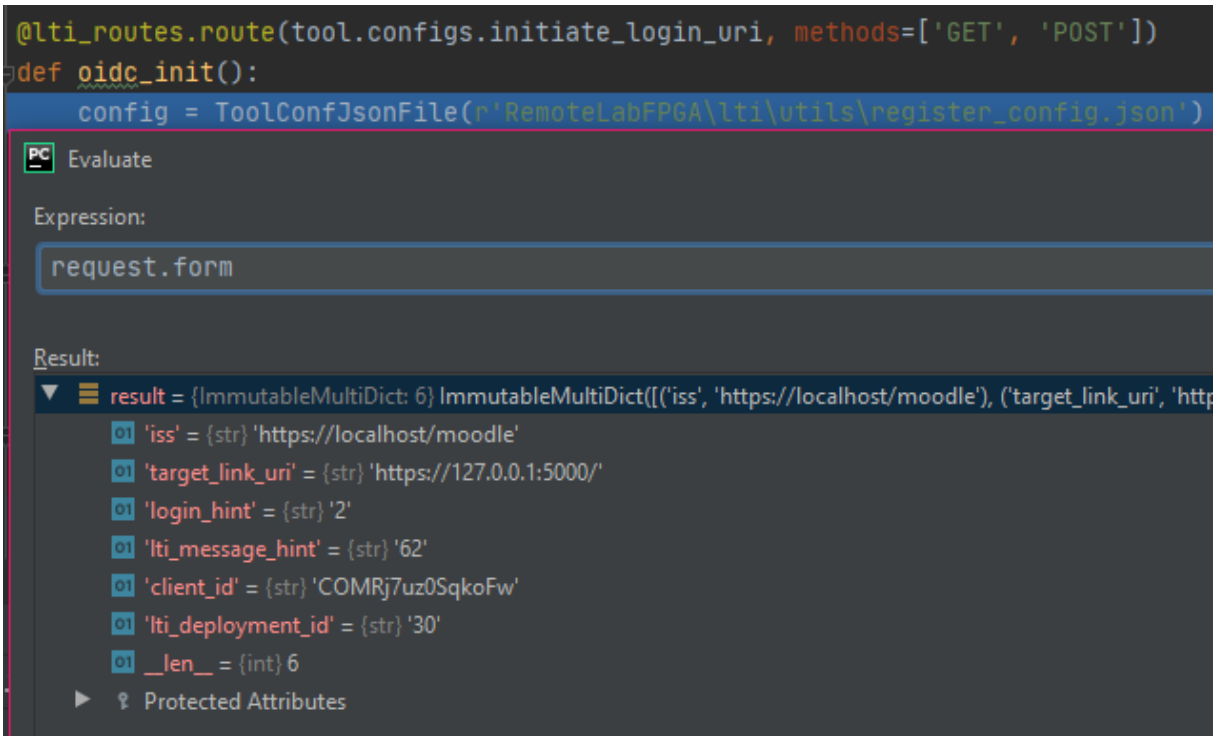


Figura 20 – Inicialização de login. fonte: autor.



Figura 21 – Resposta de autenticação. fonte: autor.

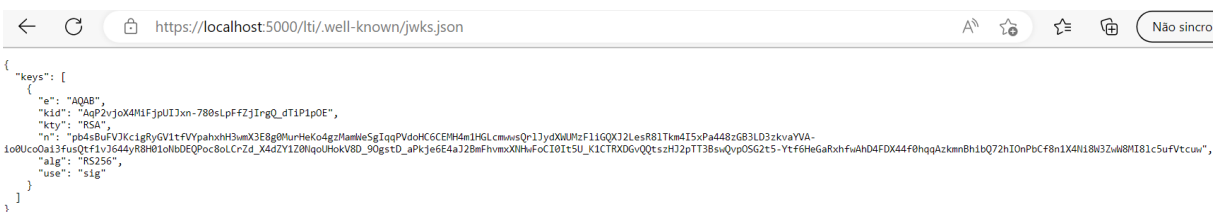


Figura 22 – Key set da Ferramenta. fonte: autor.

### 4.3.3 LTI v1.3 Core Specification

Essa especificação é responsável pela definição do ecossistema LTI, que é composta por uma Plataforma, ou seja, um sistema que precisa delegar funcionalidades para ferramentas externas, e uma Ferramenta, que é a aplicação externa, provendo serviços para a Plataforma. Temos nesta seção a definição de alguns conceitos importantes para o desenvolvimento do sistema.

O primeiro deles diz respeito aos diferentes métodos de integração, comentados anteriormente na seção 4.1, que são: *multi-tenant* e *single-tenant*. O primeiro ocorre quando é feito o registro uma única vez pelo administrador do LMS, e a Ferramenta pode ser utilizada em diferentes contextos daquela Plataforma. Já o segundo ocorre quando cada usuário com privilégios precisa fazer o registro em cada contexto que quiser utilizar a Ferramenta. Independente do método, cada integração deve ser identificada através da *Claim* imutável e única *deployment\_id*, que deve ser utilizada em todas as mensagens entre as duas partes, junto com a *Claim client\_id*. Essa identificação é gerada pela Plataforma e obtida no final do processo de registro. No caso deste trabalho utilizamos o modelo *multi-tenant*, onde a Plataforma Moodle é identificada pelo *client\_id*, e cada contexto é identificado pela *deployment\_id*, que representam as diferentes turmas que utilizam o laboratório remoto, como mostra a figura 23.

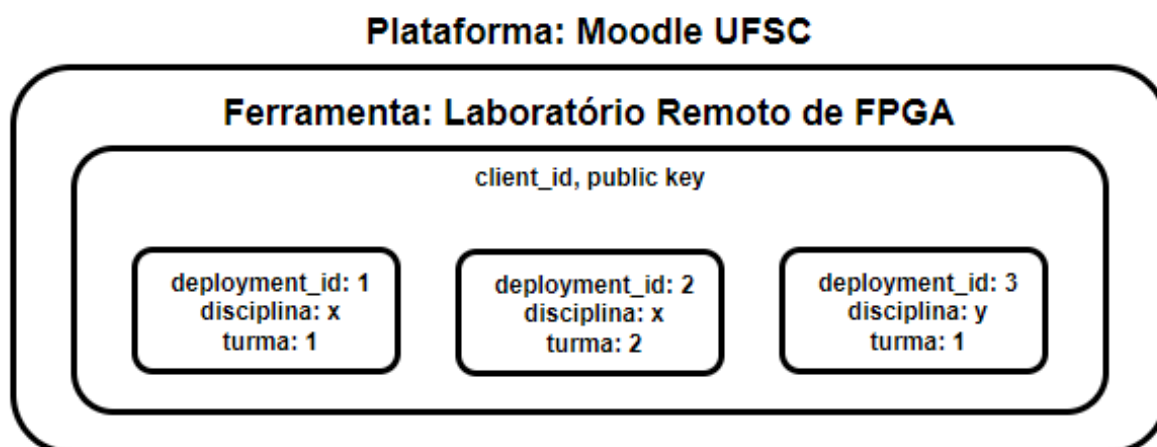


Figura 23 – Modelo multi-tenant. fonte: adaptado de (LEARNING. . . , 2019)

Outro termo importante está relacionado aos diferentes recursos que um usuário pode utilizar da Ferramenta externa, que devem ser disponibilizados através de *LTI Links*, redirecionando para diferentes páginas, podendo conter metadados. Neste trabalho temos quatro *LTI Links* importantes, como mostra a Figura 24, sendo o primeiro a disponibilização do sistema como um todo, o segundo e terceiro exemplos de utilização dos serviços LTI que serão explicados nas Seções 4.3.5 e 4.3.4, e um utilizando ambos serviços. Cada *LTI Link* deve possuir um *deployment\_id* único. Caso o *LTI Link* esteja relacionado a um recurso específico, como é o caso do segundo e terceiro, deverá possuir um *resource\_link\_id*, sendo nomeado como *Resource Link*. Quando o usuário interage com algum *LTI Link*, é acionado o Fluxo de inicialização OpenID Connect, explicado na seção 4.3.2, chamado de *LTI Launch*. Por fim, na Figura 25 temos as diferentes *Claims*, e como elas se relacionam durante o desenvolvimento e

uso dessa especificação.



Figura 24 – LTI Links. fonte: autor.

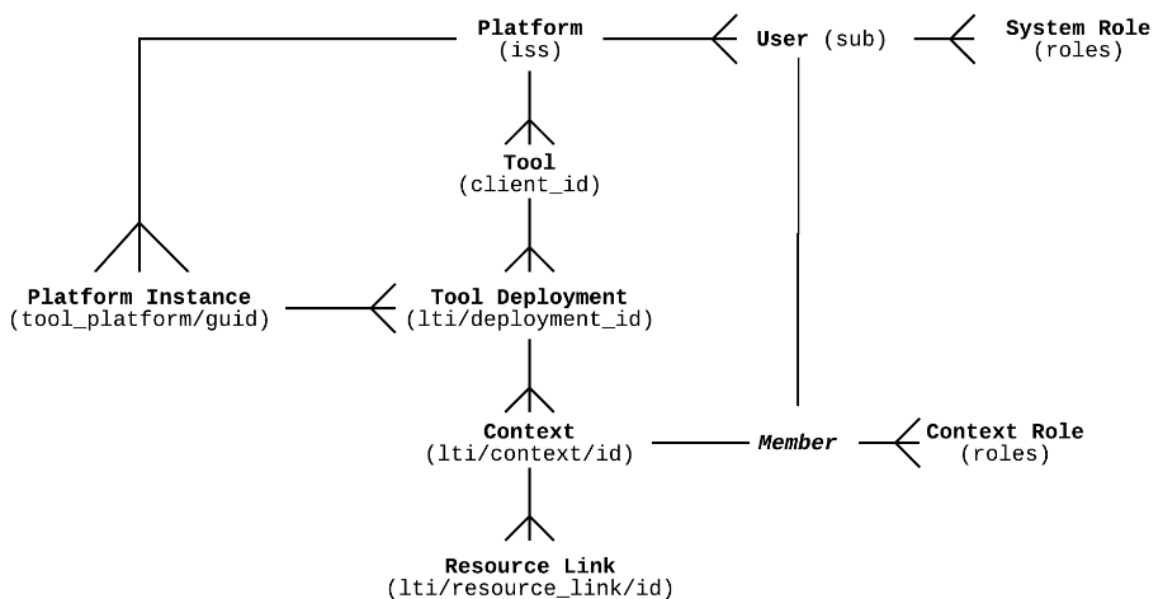


Figura 25 – Modelo de domínio LTI. fonte: (LEARNING... , 2019)

#### 4.3.4 LTI Names and Role Provisioning Services (NRPS)

Essa especificação permite que a Ferramenta obtenha acesso aos dados de usuários e seus respectivos cargos dentro de um contexto, bem como as atividades realizadas por cada um, de forma segura. Para utilizar este serviço é necessário primeiramente solicitar acesso a este recurso, informando durante a etapa de registro na *Claim scope*, para que o *Access Token* gerado possua esta permissão. Após, deve-se realizar uma requisição do tipo *HTTP GET* para

o *endpoint* informado pela Plataforma durante o fluxo de inicialização. Para o desenvolvimento desta etapa também utilizamos a biblioteca *pylti1p3*, onde o usuário faz uma solicitação para o *Resource Link*, e então retornamos as informações dos usuários, como mostra a Figura 26.

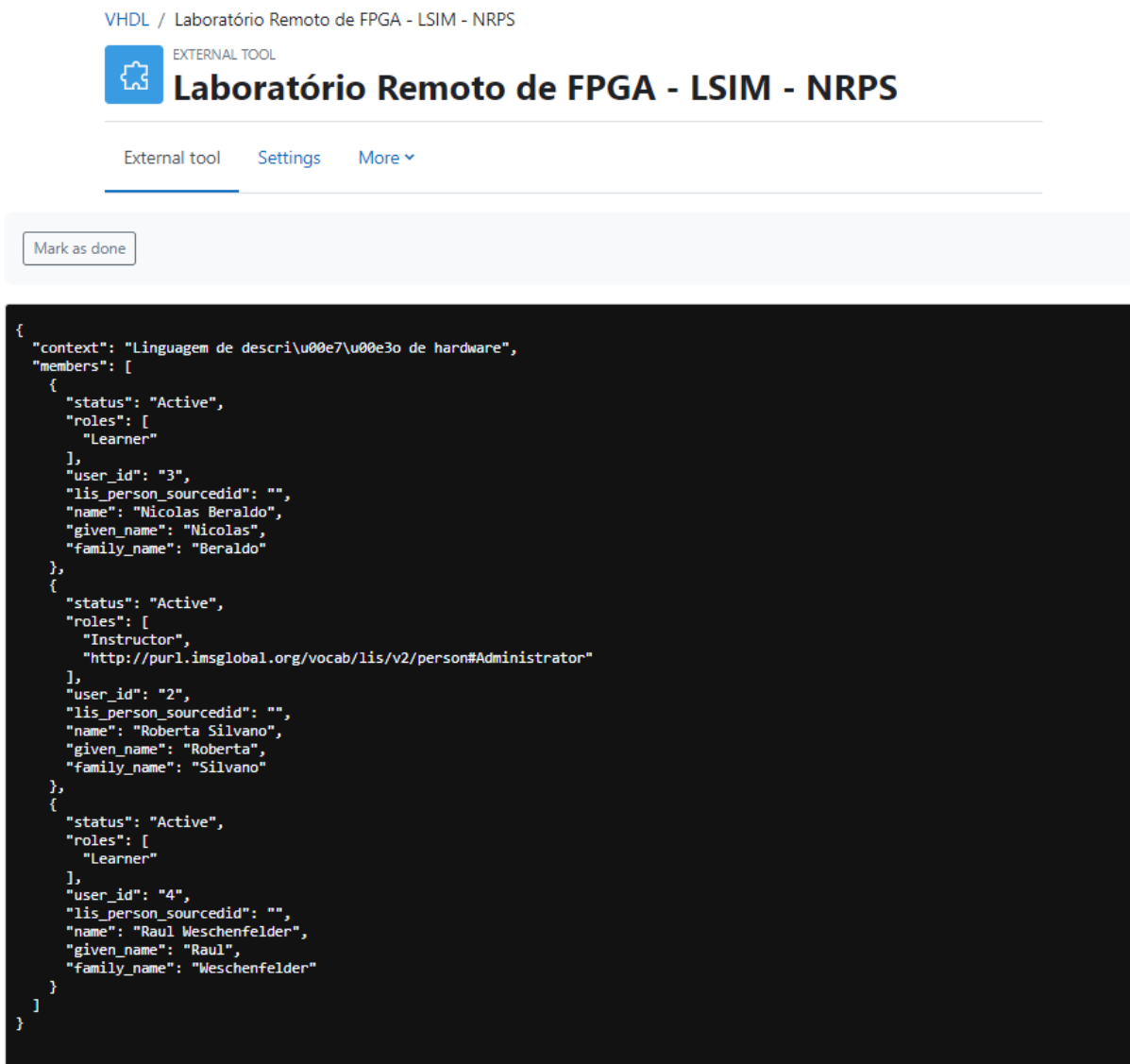


Figura 26 – Uso da NRPS. fonte: autor.

#### 4.3.5 LTI Assignment and Grade Services Specification (AGS)

Uma das grandes vantagens de utilizar o LTI neste projeto vem do uso desta especificação, que gera a possibilidade de implementar atividades específicas para os usuários, definindo as notas de forma automática. Neste caso, o *Resource Link* deverá ser a tarefa, que estará associada a uma coluna da tabela de notas, também conhecido como *Line Item*. Esse documento define três serviços: *Line Item Service*, *Score Service* e *Result Service*. O primeiro permite o gerenciamento dos *Line Items*, o segundo que a Ferramenta envie notas para a Plataforma, e portanto é somente de escrita, e o terceiro, que é somente leitura, obtém as notas atuais da Plataforma. A Figura 27 mostra alguns detalhes deste padrão.

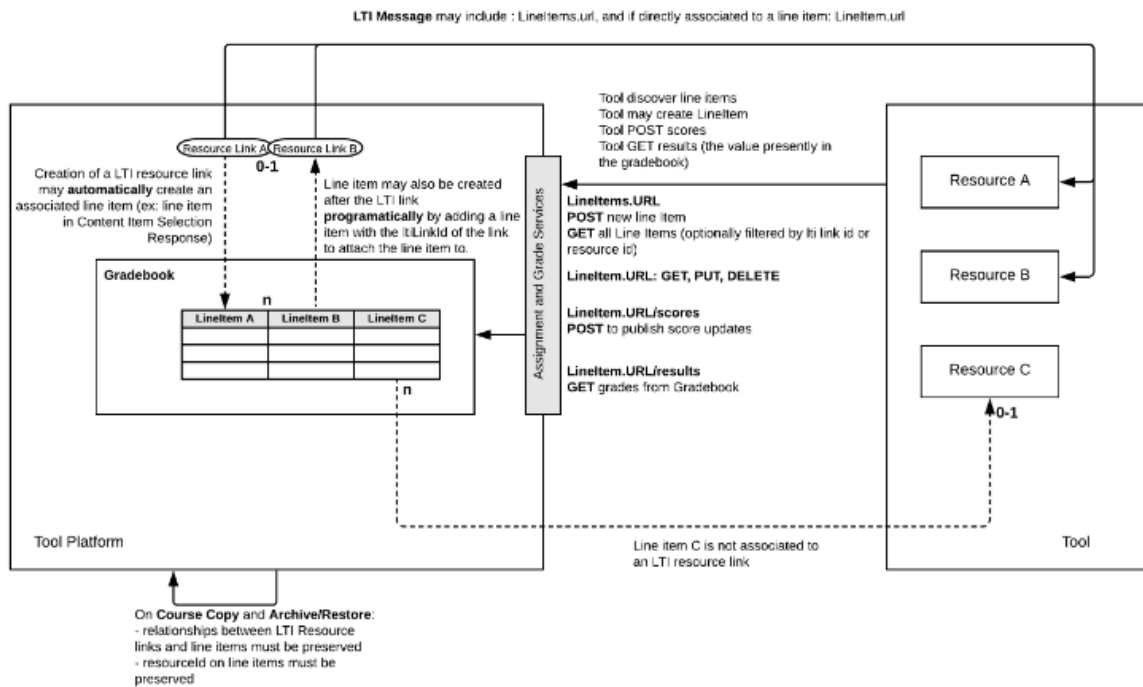


Figura 27 – Uso do AGS. fonte: (LEARNING. . . , 2019)

Assim como no NRPS, para fazer uso desses serviços é necessário adicionar essa solicitação na *Claim scope*, e com isso as mensagens geradas pela Plataforma irão possuir uma *Claim* para informar o *endpoint* do serviço. Existem dois tipos de *Line Items*: declarativos, que são criados já com a criação do *Resource Link*, e programáticos, que são criados via código. Para este projeto estamos utilizando o segundo modelo. Na Figura 28 temos o *Resource Link* responsável por gerar e preencher um *Line Item*, postando essas informações na Plataforma. Já na Figura 29 temos o quadro de notas, já preenchido com o resultado. Um ótimo exemplo de uso combinado das especificações NRPS e AGS é para montar um *scoreboard* com os nomes e notas de diferentes usuários simultaneamente, como mostra a Figura 30.

VHDL / Laboratório Remoto de FPGA - LSIM - AGS



EXTERNAL TOOL

**Laboratório Remoto de FPGA - LSIM - AGS**

Mark as done

```
{
  "student": "Roberta Silvano",
  "score given": 95,
  "maximum score": 100,
  "activity progress": "Completed",
  "timestamp": "2022-07-11T23:02:50+0000",
  "lineitem tag": "ags"
}
```

Figura 28 – *Resource Link* responsável por gerar e preencher um *Line Item*. fonte: autor.

## Linguagem de descrição de hardware: View: User report

[Course](#) [Participants](#) [Grades](#) [Question bank](#) [Competencies](#)

User report

### User report - Roberta Silvano

| Grade item                           | Calculated weight | Grade | Range | Percentage | Feedback | Contribution to course total |
|--------------------------------------|-------------------|-------|-------|------------|----------|------------------------------|
| ■ Linguagem de descrição de hardware |                   |       |       |            |          |                              |
| <input type="checkbox"/> ags         | -                 | 95,00 | 0-100 | 95,00 %    | -        | -                            |
| Σ Course total                       | -                 | -     | 0-100 | -          | -        | -                            |

Figura 29 – Resultado das notas. fonte: autor.

VHDL / Laboratório Remoto de FPGA - LSIM - SCOREBOARD



EXTERNAL TOOL

### Laboratório Remoto de FPGA - LSIM - SCOREBOARD

[External tool](#) [Settings](#) [More](#)

Mark as done

```
{
  "ags": [
    {
      "grade": 95,
      "name": "Roberta Silvano"
    },
    {
      "grade": 90,
      "name": "Nicolas Beraldo"
    },
    {
      "grade": 90,
      "name": "Raul Meschenfelder"
    }
  ]
}
```

Figura 30 – Exemplo de *Scoreboard*. fonte: autor.

#### 4.3.6 IMS LTI Deep Linking Specification (DL)

O objetivo principal desta especificação é permitir um método mais simples para a Plataforma integrar conteúdos específicos da Ferramenta, adicionando múltiplos *Resources Links* simultaneamente, como mostra a Figura 31, modificando a *Claim target\_link\_uri* alvo. Para isto é utilizado o fluxo da figura 32, que faz uso do fluxo de inicialização OpenID Connect, mas gerando um formulário onde o usuário poderá escolher quais conteúdos serão adicionados no seu ambiente.

## 5 Conclusão e Trabalhos Futuros

Com o uso das especificações *Learning Tools Interoperability* foi possível desenvolver a terceira camada do padrão IEEE 1876-2019, aplicando funcionalidades cruciais para a ad-

# Linguagem de descrição de hardware

Course Settings Participants Grades Reports More ▾

## ✚ Adding a new External tool <sup>?</sup>

The following items will be added to your course:

- Laboratório Remoto de FPGA - LSIM - AGS
- Laboratório Remoto de FPGA - LSIM - NRPS
- Laboratório Remoto de FPGA - LSIM - SCOREBOARD

Save and return to course

Cancel

Figura 31 – Uso do *Deep Linking*. fonte: autor.

ministração do sistema, atingindo o objetivo proposto neste trabalho, tendo sido efetuado todo o planejamento proposto na Figura 2, realizando a integração entre o Laboratório Remoto de FPGA e o Moodle. As duas principais contribuições com certeza foram o desenvolvimento do fluxo de registro dinâmico, e uma fonte detalhada de como utilizar estas especificações, facilitando e incentivando o uso para futuros projetos. Outro benefício deste trabalho é garantir a segurança do laboratório e dos usuários através do LTI, que era um dos pontos de cuidado do projeto como um todo, como podemos ver em (WERNER *et al.*, 2021).

O próximo passo é finalizar o desenvolvimento das outras camadas do padrão IEEE, para que seja possível disponibilizar o sistema como um todo, implementando inicialmente nas duas universidades envolvidas. A camada um possui uma importância maior relacionada a este trabalho, visto que a partir dela será possível utilizar as implementações dos serviços desenvolvidos aqui de forma mais aplicável no mundo real, criando diversos *Resources Links*, que integrem com as rotas do NRPS, AGS e DL.

Para trabalhos futuros é recomendado o desenvolvimento de um *template*, com rotas pré definidas, e que permita a configuração direta das partes envolvidas, de forma que desenvolvedores possam focar no sistema do laboratório, apenas envolvendo-o com o *template*, mas sem depender de sistemas terceiros. Também é interessante pensar em combinar a implementação feita aqui do fluxo de registro com a biblioteca já existente em Python, facilitando o uso. Outra melhoria importante é validar a conformidade deste sistema com a especificação (LEARNING. . . , 2019), de forma a obter certificação e divulgação do nosso sistema.

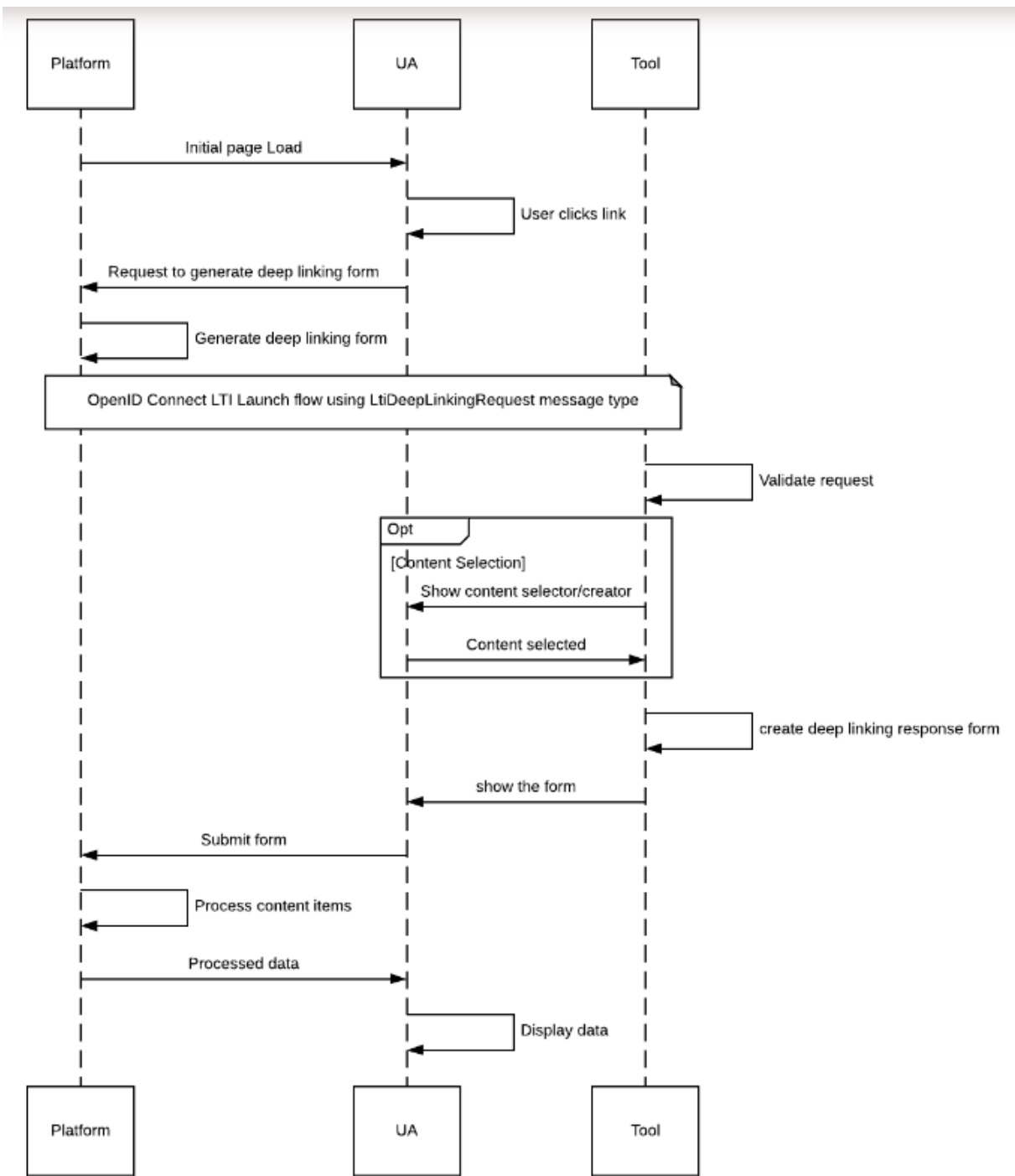


Figura 32 – Fluxo do *Deep Linking*. fonte: (IMS..., 2019)



## Referências

- BEREJUCK, Marcelo Daniel; MATIAS, Jhennifer Cristine; WERNER, Emilio; SALIAH-HASSANE, Hamadou. Work-in Progress: Development of an E-learning FPGA Platform Following the IEEE SA Std. 1876 - 2019 Standard for Networked Smart Learning Objects for Online Laboratories. *In: AUER, Michael E.; BHIMAVARAM, Kalyan Ram; YUE, Xiao-Guang (Ed.). **Online Engineering and Society 4.0**. Cham: Springer International Publishing, 2022. P. 3–9.*
- BORJA-JIMÉNEZ, Romario Marcelo; SARANGO-LAPO, Celia Paola; JARA-ROA, Dunia Inés; RÍOS-CASTILLO, Luis Eduardo. LTI Solution as an Integrating Element between an LMS and a Remote Laboratory (LR). *In: 2019 XIV Latin American Conference on Learning Technologies (LACLO). [S.l.: s.n.], 2019. P. 123–128.*
- GARCIA-LORO, Felix; SAN CRISTOBAL, Elio; CASTRO, Manuel. PILAR: A Federation of VISIR Systems for Analog Electronics. *In: IEEE. 2019 5th Experiment International Conference (exp. at'19). [S.l.: s.n.], 2019. P. 260–261.*
- HARDT, Dick. **The OAuth 2.0 Authorization Framework**. [S.l.]: RFC Editor, out. 2012. RFC 6749. (Request for Comments, 6749).
- IEEE Standard for Networked Smart Learning Objects for Online Laboratories. **IEEE Std 1876-2019**, p. 1–57, 2019.
- IMS Learning Tools Interoperability (LTI)® Deep Linking Specification. **IMS Global Learning Consortium**, 2019.
- IMS Security Framework. **IMS Global Learning Consortium**, 2021.
- JONES, Michael; BRADLEY, John; SAKIMURA, Nat. **JSON Web Token (JWT)**. [S.l.]: RFC Editor, 2015. RFC 7519. (Request for Comments, 7519).
- LEARNING Tools Interoperability (LTI) Assignment and Grade Services Specification. **IMS Global Learning Consortium**, 2019.
- LEARNING Tools Interoperability (LTI) Dynamic Registration Specification. **IMS Global Learning Consortium**, 2021.
- LEARNING Tools Interoperability Advantage Conformance Certification Guide. **IMS Global Learning Consortium**, 2019.
- LEARNING Tools Interoperability Core Specification. **IMS Global Learning Consortium**, 2019.

LEARNING Tools Interoperability Names and Role Provisioning Services. **IMS Global Learning Consortium**, 2019.

OPENID Connect Core 1.0. **OpenID Foundation**, 2014.

ORDUÑA, Pablo *et al.* An extensible architecture for the integration of remote and virtual laboratories in public learning tools. **IEEE Revista Iberoamericana de Tecnologías del Aprendizaje**, IEEE, v. 10, n. 4, p. 223–233, 2015.

ORDUÑA, Pablo *et al.* Generic integration of remote laboratories in learning and content management systems through federation protocols. *In*: 2013 IEEE Frontiers in Education Conference (FIE). [S.l.: s.n.], 2013. P. 1372–1378.

OVERVIEW of FPGA and EDA Software. *In*: FPGA Prototyping by VHDL Examples. [S.l.]: John Wiley Sons, Ltd, 2008. cap. 2, p. 11–34. ISBN 9780470231630.

POST, Lysanne S.; GUO, Pengyue; SAAB, Nadira; ADMIRAAL, Wilfried. Effects of remote labs on cognitive, behavioral, and affective learning outcomes in higher education. **Computers Education**, v. 140, p. 103596, 2019. ISSN 0360-1315.

REIS, Luiz Fernando; MACARIO, Epitácio. DÍVIDA PÚBLICA E FINANCIAMENTO DAS UNIVERSIDADES FEDERAIS E DA CIÊNCIA E TECNOLOGIA NO BRASIL (2003-2020). **Práxis Educacional**, v. 16, n. 41, p. 20–46, jul. 2020.

ROY CHOUDHURY, Barnali; SINGH, Rupinder. Learning Tools Interoperability: Paradigm shift of Open Learning. *In*.

RUANO, Ildefonso Ruano; ESTÉVEZ, Elísabet Estévez; GARCÍA, Alejandro Sánchez; GARCÍA, Javier Gámez; ORTEGA, Juan Gómez. Integration of online Laboratories in Learning Platforms. *In*: 2020 XIV Technologies Applied to Electronics Teaching Conference (TAEE). [S.l.: s.n.], 2020. P. 1–8.

SANTAMARÍA-BUITRAGO, Frey; BALLESTEROS-RICAURTE, Javier Antonio; GONZALEZ AMARILLO, Angela. Ecosystem for the deployment and management of virtual laboratories based on the standard IMS LTI. **Revista Facultad de Ingeniería**, v. 28, p. 79–99, out. 2019.

TORRE, Luis de la; SÁNCHEZ, José; ANDRADE, Tiago Faustino; RESTIVO, Maria Teresa. Easy creation and deployment of Javascript remote labs with EjsS and Moodle. *In*: 2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV). [S.l.: s.n.], 2016. P. 260–261.

TORRE CUBILLO, Luis de la; GUINALDO, Maria; HERADIO, Ruben; DORMIDO, S. The Ball and Beam System: A Case Study of Virtual and Remote Lab Enhancement With Moodle. **IEEE Transactions on Industrial Informatics**, v. 11, ago. 2015.

VIEGAS, Clara *et al.* Impact of a remote lab on teaching practices and student learning. **Computers Education**, v. 126, p. 201–216, 2018. ISSN 0360-1315.

WERNER, Emilio; MATIAS, Jhennifer Cristine; BEREJUCK, Marcelo Daniel; SALIAH-HASSANE, Hamadou. Main Attacks and Ways to Keep Security Guidelines Updated in Remote Laboratories. *In*: AUER, Michael E.; BHIMAVARAM, Kalyan Ram; YUE, Xiao-Guang (Ed.). **Online Engineering and Society 4.0**. Cham: Springer International Publishing, 2022. P. 115–121.

WERNER, Emilio; MATIAS, Jhennifer Cristine; DANIEL BEREJUCK, Marcelo; SALIAH-HASSANE, Hamadou. Evaluation of blockchain techniques to ensure secure access on remote FPGA laboratories. *In*: 2021 9th International Symposium on Digital Forensics and Security (ISDFS). [S.l.: s.n.], 2021. P. 1–6.