

INTEGRAÇÃO DE UM TERMINAL WEB AO MOODLE

GABRIEL MULLER FISCHER¹; PEDRO IVO KUHN²;
RAFAEL BURLAMAQUI AMARAL³; GERSON GERALDO H. CAVALHEIRO⁴;

¹*Universidade Federal de Pelotas – gmfischer@inf.ufpel.edu.br*

²*Universidade Federal de Pelotas – pikuhn@inf.ufpel.edu.br*

³*Universidade Federal de Pelotas – rafael.amaral@inf.ufpel.edu.br*

⁴*Universidade Federal de Pelotas – gerson.cavalheiro@inf.ufpel.edu.br*

1. INTRODUÇÃO

Os Ambientes Virtuais de Aprendizagem (AVAs) são hoje a plataforma central da educação digital, servindo como o ponto de encontro onde alunos e professores gerenciam conteúdos e avaliações. Contudo, em disciplinas de natureza prática, como as de programação, as funcionalidades padrão de um AVA são muitas vezes insuficientes, criando a necessidade de integrar ferramentas externas especializadas para atividades de experimentação. Neste cenário, o Moodle (MOODLE, 2025) se destaca por sua arquitetura flexível, projetada para se comunicar de forma segura e padronizada com tais soluções, ampliando seu potencial pedagógico. A integração de ferramentas externas no Moodle por meio do protocolo Learning Tools Interoperability (LTI) 1.3 (MOODLEDOCS, 2025) oferece um meio seguro e padronizado de comunicação entre plataformas. O LTI 1.3 utiliza autenticação baseada em OAuth 2.0, OpenID Connect e troca de informações via JSON Web Tokens (JWT), garantindo a confiabilidade e a integridade dos dados compartilhados.

O presente trabalho apresenta o processo de configuração e implementação da integração de um Terminal Web, desenvolvido em trabalho anterior (KUHN, 2025) para provisionamento dinâmico de um cluster virtualizado com acesso via navegador, ao Moodle utilizando o LTI 1.3. O objetivo é permitir que esse terminal, originalmente independente, seja utilizado diretamente dentro do ambiente de aprendizagem, com autenticação automática e transmissão de informações acadêmicas.

O Terminal Web integrado ao Moodle permite que os alunos escrevam, compilem e executem seus códigos diretamente no ambiente virtual de aprendizagem, sem necessidade de configurações adicionais em seus dispositivos. Através do protocolo LTI 1.3, o professor pode acessar, a partir do próprio Moodle, o histórico de execução, saídas de compilação. Quando aplicável, essas informações podem ser enviadas automaticamente para o livro de notas do Moodle por meio dos serviços *Assignment and Grade Services*, garantindo uma avaliação integrada e simplificada.

2. TRABALHOS RELACIONADOS

A integração de ambientes de programação em AVAs não é um problema novo. As soluções existentes podem ser divididas em duas categorias principais: plugins nativos, que se integram profundamente ao Moodle, e plataformas externas, que se comunicam via padrões de interoperabilidade como o LTI.

A abordagem mais tradicional no Moodle é o uso de plugins como o *Virtual Programming Lab (VPL)* (VIRTUAL PROGRAMMING LAB, 2025) e o *CodeRunner* (CODERUNNER, 2025). Concretamente, o VPL oferece ao aluno um editor de código interativo para escrever, compilar e executar programas diretamente no navegador. Já o CodeRunner permite que os professores criem, dentro de um quiz, questões de programação cuja resposta do aluno é avaliada automaticamente contra uma série de casos de teste pré-definidos. Ambas as soluções são robustas, porém, por serem plugins nativos, apresentam um forte acoplamento com a plataforma. Sua instalação e atualização afetam diretamente a base de código do Moodle, e sua arquitetura de execução geralmente depende de servidores dedicados ou *sandboxes* configurados de forma estática, o que pode representar um desafio de escalabilidade.

Este trabalho se posiciona de forma única ao sintetizar os benefícios de ambas as abordagens. Adota-se o padrão de integração moderno e desacoplado LTI 1.3, similar às plataformas externas. Contudo, a solução é auto-hospedada (*self-hosted*), garantindo que todos os dados permaneçam na infraestrutura da instituição, uma vantagem do modelo de plugins. O diferencial inovador reside na camada de execução, que utiliza Kubernetes (KUBERNETES, 2025) para o provisionamento dinâmico de ambientes, oferecendo um nível de escalabilidade e isolamento de recursos superior a ambas as abordagens tradicionais.

3. METODOLOGIA

O presente trabalho foi desenvolvido seguindo um processo sequencial de pesquisa e implementação. O ponto de partida foi o software de Terminal Web desenvolvido por Kuhn (KUHN, 2025), que serviu como base para a integração. A metodologia foi dividida em duas fases principais.

A primeira fase do projeto consistiu na pesquisa e preparação do ambiente. Após a revisão da documentação oficial do Moodle para o entendimento do padrão LTI 1.3, a etapa seguinte foi a criação de um ambiente de validação isolado. Para este fim, o autor instalou e configurou uma instância dedicada do Moodle em um servidor de testes. Este passo foi crucial para garantir o acesso administrativo total, indispensável para o registro de ferramentas LTI 1.3, e para oferecer um ambiente controlado onde os experimentos de integração pudessem ser conduzidos sem qualquer risco ao sistema de produção da instituição. Foi com base nesse ambiente, foi realizado o mapeamento e a configuração manual da ferramenta externa (External Tool) diretamente na interface administrativa do Moodle, acessível ao administrador do sistema pelo menu: *Administração do site* → *Plugins* → *Módulos de atividade* → *Ferramenta externa* → *Gerenciar ferramentas*.

A segunda fase, atualmente em planejamento, consiste no desenvolvimento do backend da ferramenta para suportar o fluxo de autenticação LTI 1.3 e a orquestração dos contêineres.

4. RESULTADOS E DISCUSSÃO

O primeiro resultado concreto é a configuração bem-sucedida do Moodle para atuar como uma plataforma LTI 1.3, tecnicamente apto para iniciar o fluxo de autenticação e comunicação com a ferramenta externa. O segundo e principal

resultado é a arquitetura de integração projetada, que define os fluxos de interação entre o docente, o aluno e o sistema. Ela é dividida em duas operações principais:

O fluxo de configuração do ambiente pelo docente é uma característica central da arquitetura é a flexibilidade. O professor será responsável por preparar e publicar uma imagem Docker (DOCKER, 2025) contendo as ferramentas e bibliotecas necessárias para a sua disciplina em um registro de contêineres, como o Docker Hub. Ao criar a atividade "Ferramenta Externa" no Moodle, o professor utilizará o campo "Parâmetros personalizados" para especificar qual imagem deve ser utilizada, inserindo um valor como (ex: "docker_image=usuário/imagem-disciplina:1.0"). Este mecanismo é crucial, pois mantém o Moodle completamente agnóstico à infraestrutura subjacente; ele apenas transmite esse parâmetro de forma segura para o backend. Para garantir a segurança e o controle, o backend manterá uma lista de permissões de imagens ou repositórios confiáveis, validando o parâmetro recebido antes de prosseguir.

Uma vez configurada a atividade, o fluxo de acesso do aluno seguirá o plano de implementação LTI 1.3. O backend da ferramenta explora os *endpoints* OIDC para receber a requisição de login do Moodle. A etapa central de segurança será a validação criptográfica do token JWT, verificando sua assinatura com a chave pública da plataforma e validando rigorosamente suas claims, como o emissor, o público e o tempo de expiração. Com a sessão autenticada, o fluxo se torna interativo: o backend apresenta uma interface intermediária onde o aluno especifica o número de máquinas (contêineres) que deseja provisionar. Após a submissão, o backend autoriza a requisição e se comunica com a API do Kubernetes para orquestrar a criação do ambiente, utilizando a imagem Docker definida pelo professor. Finalmente, o aluno é direcionado para o terminal de acesso ao ambiente recém-criado, conforme ilustrado na Figura 1.

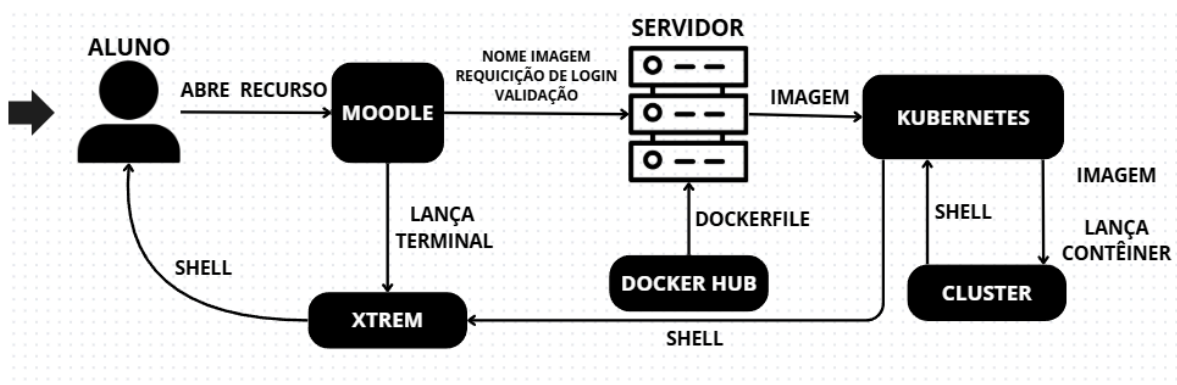


Figura 1 - Fluxo de interação do aluno

A discussão sobre os resultados aponta para os benefícios esperados da arquitetura planejada. A integração via LTI 1.3, por ser puramente baseada em configuração, promete maior manutenção e estabilidade em comparação com plugins nativos como VPL e CodeRunner. Adicionalmente, o provisionamento de *pods* dinâmicos via Kubernetes oferecerá um ambiente isolado e efêmero, representando um avanço significativo em segurança e escalabilidade frente às arquiteturas estáticas. Por fim, o modelo auto-hospedado se posiciona como uma

alternativa robusta às plataformas comerciais, buscando combinar a flexibilidade do padrão LTI com a soberania de dados de um sistema interno.

5. CONCLUSÕES

Este artigo apresentou e validou a viabilidade de uma arquitetura para a integração de um Terminal Web ao Moodle. A solução proposta se destaca por combinar três elementos estratégicos: a interoperabilidade do padrão LTI 1.3, o controle de uma solução auto-hospedada e a escalabilidade de uma infraestrutura em Kubernetes. Conclui-se que este modelo híbrido é uma alternativa robusta e segura aos plugins nativos, como VPL, e às plataformas LTI comerciais, oferecendo um caminho para que instituições de ensino modernizem seu ecossistema de ferramentas educacionais com eficiência e autonomia. A implementação do backend e a validação em um projeto piloto são os próximos passos para a consolidação do trabalho.

6. REFERÊNCIAS BIBLIOGRÁFICAS

MOODLEDOCS. LTI External tools. Disponível em: https://docs.moodle.org/en/LTI_External_tools. Acesso em: 21 ago. 2025.

MOODLE. Open-source learning platform. Moodle Pty Ltd, 2025. Disponível em: <https://moodle.org>. Acesso em: 28 ago. 2025.

KUHN, Pedro Ivo Kuhn: Terminal Web. 2025. Disponível em: <https://github.com/PedroIvoKuhn/webTerminal>. Acesso em: 21 ago. 2025.

VIRTUAL PROGRAMMING LAB (VPL). Virtual Programming Lab for Moodle. University of Las Palmas de Gran Canaria, 2025. Disponível em: https://moodle.org/plugins/mod_vpl. Acesso em: 21 ago. 2025.

CODERUNNER. CodeRunner Question Type for Moodle. University of Canterbury, 2025. Disponível em: https://moodle.org/plugins/qtype_coderunner. Acesso em: 21 ago. 2025.

KUBERNETES. Production-Grade Container Orchestration. The Kubernetes Authors / CNCF, 2025. Disponível em: <https://kubernetes.io/>. Acesso em: 28 ago. 2025.

DOCKER. Accelerated, Containerized Application Development. Docker, Inc., 2025. Disponível em: <https://www.docker.com/>. Acesso em: 28 ago. 2025.