

UNIVERSITY INTEGRATED PLATFORM FOR PROJECT
MANAGEMENT IN PBL



PLATAFORMA INTEGRADA UNIVERSITÁRIA PARA GERENCIAMENTO DE PROJETOS EM PBL

SILVA, Marcelo Abreu; MARTINS, Paulo Henrique Silva; LOURENÇO, Ramires Oliveira; GONÇALVES, Samuel Pereira; CARVALHO, Marcos Alberto; CARVALHO, Jaqueline Corrêa Silva; SANTOS, Flávia Aparecida Oliveira; BASTOS, Camila; SOUZA, Patrícia Carolina; RAMOS, Celso de Ávila

Marcelo Abreu Silva, UNIFENAS, Brasil

Paulo Henrique Silva Martins, UNIFENAS, Brasil

Ramires Oliveira Lourenço, UNIFENAS, Brasil

Samuel Pereira Gonçalves, UNIFENAS, Brasil

Marcos Alberto Carvalho, UNIFENAS, Brasil

Jaqueline Corrêa Silva Carvalho, UNIFENAS, Brasil

Flávia Aparecida Oliveira Santos, UNIFENAS, Brasil

Camila Bastos, UNIFENAS, Brasil

Patrícia Carolina Souza, UNIFENAS, Brasil

Celso de Ávila Ramos, UNIFENAS, Brasil

Revista Científica da UNIFENAS
Universidade Professor Edson Antônio Velano, Brasil
ISSN: 2596-3481
Publicação: Trimestral
vol. 6, nº. 5, 2024
revista@unifenas.br

Recebido: 08/07/2024
Aceito: 28/08/2024
Publicado: 09/09/2024

URL: <https://revistas.unifenas.br/index.php/revistaunifenas/issue/view/52>

DOI: 10.29327/2385054.6.5-11

ABSTRACT: The integrated university platform for project management in PBL (PLATINUN UNIFENAS) is an initiative of the Prof. Edson Antônio Velano University, being a web platform with the objective of creating a unified environment and focused on academic projects in the field of teaching, research and extension. It was developed using two websites in React for frontend, using the Monorepo architecture that centralizes a set of applications in a single repository. The API was developed in AspNetCore with user management using Keycloak, in addition to MySQL for the database. The platform uses three types of users: Student, Teacher and Administrator, thus having different permissions. The student can only see and send the activities on his page. The teacher can manage activities, in addition to evaluating the activity received from the student, while the administrator records users and disciplines. The platform facilitates the application of academic activities by students and teams in a PBL (Problem-Based Learning) model, helping teachers to communicate and interact with students in assessment ways, stimulating progress and performance. Thus, an innovative and proprietary platform is delivered to the University for continuous use in various academic activities.

KEYWORDS: Api, aspnetcore, keycloak, Monorepo, Mysql, Pbl, React.

RESUMO: A plataforma integrada universitária para gerenciamento de projetos em PBL (PLATINUN UNIFENAS) é uma iniciativa da Universidade Prof. Edson Antônio Velano, sendo uma plataforma web com o objetivo de criar um ambiente unificado e focado em projetos acadêmicos no âmbito do ensino, da pesquisa e da extensão. A plataforma PLATINUN pretende agregar em um único ambiente os projetos acadêmicos realizados sob metodologia inovadora PBL (Problem Based Learning), permitindo a todos os usuários (acadêmicos e professores) acompanharem o desenvolvimento dos projetos, servindo, inclusive como um repositório de projetos para todos os cursos da Universidade. Foi desenvolvida utilizando dois sites em React para frontend, utilizando a arquitetura Monorepo que centraliza um conjunto de aplicações em um único repositório. A API foi desenvolvida em AspNetCore com gerenciamento dos usuários usando o Keycloak, além do

MySQL para banco de dados. A plataforma utiliza três tipos de usuários, aluno, professor e administrador, tendo assim permissões diferentes. O aluno apenas pode ver e enviar as atividades em sua página, o professor pode gerenciar atividades, além de avaliar a atividade recebida do aluno e o administrador, registra os usuários e disciplinas. A plataforma facilita a aplicação de atividades acadêmicas de estudantes e equipes em modelo PBL (Aprendizagem Baseada em Problemas), auxiliando os professores na comunicação e interação com os alunos em meios avaliativos, estimulando o progresso e rendimento, entrega-se à Universidade a que se destina uma plataforma inovadora e proprietária para uso contínuo em diversas atividades acadêmicas.

PALAVRAS-CHAVE Api, aspnetcore, keycloak, Monorepo, Mysql, Pbl, React.

1 INTRODUÇÃO

A tecnologia tem contribuído cada vez mais para melhorar a qualidade na educação e na aprendizagem nos ensinamentos acadêmicos, porém as instituições de ensino nem sempre contam com sistemas próprios que possam ter uma integração entre aluno e professores, seja pelo alto custo ou pela falta de requisitos necessários no sistema para cada instituição de ensino.

O objetivo específico foi criar um ambiente unificado e seguro para a manutenção de projetos acadêmicos no âmbito do ensino, da pesquisa e da extensão, sendo a plataforma baseada na metodologia de ensino PBL. A plataforma comporta os módulos e demandas acadêmicas exigidas pela universidade, e tem como sua concepção o marco zero de uma solução própria e autônoma para atuais e futuras atividades desenvolvidas com esta metodologia de ensino na instituição. O ambiente conta com o compartilhamento de arquivos e atividades de forma interativa e de fácil entendimento, abordando o uso da metodologia PBL que emprega técnicas de divisão de atividades em etapas para cada módulo, gerando seu próprio peso de relevância por disciplina.

A metodologia de PBL (Problem Based Learning) resulta na construção do conhecimento de maneira coletiva, quebrando o paradigma do modelo convencional, na qual o professor aplica uma disciplina e os alunos podem demonstrar o quanto tiveram de aproveitamento a partir de avaliações baseadas em perguntas e respostas [1]. Neste processo de aprendizagem, o aluno é exposto a problemas do mundo real que servem de base para estimular o mesmo a desenvolver uma solução que englobe os conhecimentos apresentados por uma disciplina, como por exemplo, conhecimentos que seriam aplicados de forma prática durante a realização das atividades do PBL[2].

Segundo [3], os problemas a serem propostos

durante uma atividade PBL devem ser preferivelmente possíveis de serem resolvidos com múltiplas soluções, não estruturados e na qual os professores terão o papel apenas de auxiliar os alunos a resolverem o problema, não dando exatamente a instrução de como solucionar a atividade.

Um ambiente interativo que sirva como repositório de projetos construídos sob o PBL pode ser desenvolvido como um web site responsivo. Com o avanço da tecnologia no desenvolvimento de novos dispositivos conectados a internet surge a preocupação em relação a como desenvolver websites para dispositivos com diferentes tipos de resoluções, seja o dispositivo mobile ou desktop, assim surge o conceito de website responsivo. Segundo [4], desenvolver uma aplicação web é uma tarefa mutável, pois as metodologias de design e layout duram somente por volta de dois anos, tornando então a escolha do design responsivo muito útil, pois com ela o desenvolvedor pode criar interfaces ajustáveis a diferentes tamanhos e resoluções de tela, tornando mais fácil a navegação no website independente do dispositivo em que for acessado.

No desenvolvimento de layouts responsivos, a utilização de tecnologias específicas é crucial para garantir o correto funcionamento do design em diversos dispositivos. Três principais tecnologias formam a base para o design responsivo: HTML, JavaScript e CSS. HTML (HyperText Markup Language) fornece a estrutura básica das páginas web. JavaScript adiciona interatividade e dinamismo, enquanto CSS (Cascading Style Sheets) controla a aparência e o layout. A combinação dessas tecnologias permite a criação de interfaces web que se adaptam eficientemente a diferentes tamanhos e resoluções de tela, proporcionando uma experiência de usuário consistente e acessível em dispositivos móveis e desktops. [4].

A fim de se disponibilizar as funcionalidades de uma aplicação Web para outras com propósitos semelhantes, a utilização de APIs (Application Programming Interface) é uma solução bastante plausível. Várias são as tecnologias que permitem a construção de APIs.

O ASP.NET Core tem como seus benefícios uma ampla comunidade de desenvolvedores e diversas bibliotecas que podem auxiliar em atividades complexas, como a conectividade com serviços de cloud, capacitação para testes, facilidade de codificação por ser uma linguagem fortemente tipada, ótimo desempenho além de ser um software livre [5]. Apesar do modelo de estrutura do ASP.NET Core adequar-se muito bem tanto para server-side quando para client-side, a utilização apenas da estrutura de serviços de API Server Side foi atribuído por seu suporte a serviço de redes HTTP, tendo um grande apoio atual que torna ele uma ótima opção para domínio da regra de negócio e suas diretrizes.

2 METODOLOGIA

A plataforma integrada universitária para gerenciamento de projetos Aprendizagem Baseada em Problemas, ou em PBL, PLATINUN UNIFENAS, foi desenvolvida através de uma iniciativa da Universidade Prof. Edson Antônio Velano para ser uma plataforma web onde reúne atividades acadêmicas de estudantes e equipes de trabalhos inovadores, atividades de extensão, grupos de pesquisa etc., buscando melhorar a comunicação e a colaboração entre os integrantes, de forma simples e intuitiva. Trata-se de uma plataforma com o intuito

de agregar todos os trabalhos acadêmicos construídos sob a metodologia inovadora de ensino Problem Based Learning em um só local, permitindo aos usuários (acadêmicos e professores) acompanharem a evolução dos projetos, servindo ainda como um repositório para pesquisa.

Este tópico abordará elementos práticos e tecnológicos do projeto, analisando a forma de uso e escolha de componentes, recursos e bibliotecas que foram utilizadas para o desenvolvimento do sistema.

2.1 Planejamento

Em relação à classificação quanto à abordagem adotada, trata-se de uma pesquisa qualitativa e experimental, pois durante o estudo foram realizados testes para uma conclusão concreta. Quanto à classificação de objetivo o estudo tem empregado a metodologia de pesquisa exploratória, pois estão sendo utilizados levantamentos bibliográficos para aprofundamento no tema e também análises de exemplos já existentes e durante o desenvolvimento do trabalho foram feitas as etapas de análise bibliográfica, levantamento de requisitos, desenvolvimento da plataforma e testes.

2.2 Análise bibliográfica

Na Análise Bibliográfica, foi feito o embasamento teórico do assunto e o levantamento de requisitos para a construção da plataforma. Para a pesquisa de publicações relevantes sobre o assunto, foram usadas as seguintes palavras-chave nos mecanismos de busca: PBL, plataforma PBL, react, software acadêmico. Em seguida, na etapa de desenvolvimento, foram feitas as prototipagens do software, e sua implementação finalizado então na etapa de Testes. Foram realizadas as avaliações dos resultados obtidos com os testes feitos com o software em um ambiente experimental.

2.3 Levantamento de requisitos

Na fase de levantamento de requisitos, foi feita a coleta de todos os requisitos necessários para a construção da plataforma, de acordo com as especificações solicitadas pela Universidade de forma que o sistema atenda as suas demandas. A coleta desses dados foi realizada em reuniões semanais entre o time de desenvolvimento, professores e gestores da universidade, seguindo um cronograma pré-estabelecido, assim realizou-se a metodologia ágil para acelerar o desenvolvimento do software e os requisitos funcionais foram:

- Elaborar um ambiente online para desenvolvimento e aplicação de atividades baseado no modelo PBL.
- Criar um ambiente unificado que traga dinamismo e uma maior proximidade entre aluno e professor.
- Tornar a plataforma divertida e atrativa para que gere interesse na execução das atividades propostas

aos alunos.

- Poder criar projetos segmentados por objetivo para cada grupo de alunos
- O professor poder criar atividade para entrega de forma simples e rápida
- O aluno precisa poder enviar suas atividades para o professor de forma prática
- Os usuários precisam receber e-mails sobre atualizações relacionadas a ele:

o Notificação sobre novas atividades para correção. (Professor)

o Notificação sobre nova etapa de entrega de atividade marcada. (Aluno)

o Notificação sobre atividades corrigidas pelo professor. (Aluno)

Os requisitos não-funcionais foram:

- Criar um serviço em que o professor possa criar um projeto e definir quais alunos podem ter visão daquele projeto.
- Incluir um bloqueio para que um aluno não possa participar de um projeto a menos que um professor libere o acesso para ele.
- Sempre que uma nova etapa é criada pelo professor, todos os alunos que fazem parte do projeto em que a etapa foi criada devem ser notificados por e-mail.
- Sempre que um aluno enviar uma atividade, todos os projetos que administram o professor deverão ser notificados por e-mail.
- Quando o professor corrigir uma atividade enviada pelo aluno, o aluno dono da atividade deve ser informado por e-mail sobre a nota e sobre qual etapa se trata a atividade corrigida.
- Criar um sistema de gamificação para que o professor possa atribuir medalhas por atividade entregue pelo aluno onde a medalha será recebida de acordo com a nota da atividade do aluno, as medalhas poderão ser visíveis pelo aluno no seu perfil como troféu de mérito.

2.4 Utilização da Plataforma

A plataforma é utilizada por 3 tipos de usuários, sendo o administrador, professor e aluno, sendo que cada usuário tem diferentes responsabilidades dentro da plataforma, onde o professor pode por exemplo realizar o cadastro de projetos e avaliar alunos e projetos, enquanto o aluno pode consultar etapas de envio de atividades de projetos em que ele faz parte e consultar suas notas das atividades anteriores enviadas por ele e corrigidas por um de seus professores, ficando para o administrador a parte de gerenciamento do cadastro dos usuários no sistema, o diagrama de caso de uso a seguir exemplifica se forma visual e clara este modelo. Na Figura 1, observa-se o diagrama do uso de cada usuário no software.

Figura 1. Diagrama de casos de uso (elaborado pelos autores)



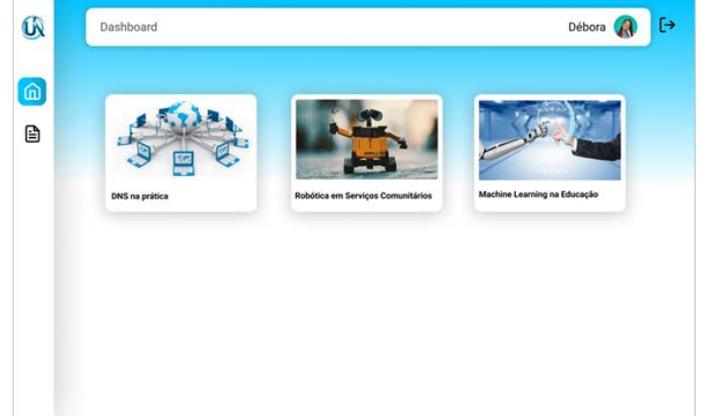
2.5 Desenvolvimento da plataforma

Na etapa de desenvolvimento, foi realizada a construção do software com as tecnologias selecionadas a partir da análise dos requisitos solicitados pela universidade e no desenvolvimento da plataforma foi escolhido o estilo de Web Site Responsivo, pois dessa maneira o sistema não terá a necessidade de ser instalado localmente no aparelho, como no caso de desenvolvimento em aplicativos nativos e/ou híbridos que requerem instalação. Desta maneira, o sistema se torna flexível e abrangente, pois para realizar o acesso será necessário que o dispositivo contenha apenas conexão à internet e um navegador, onde conseguirá acessar a plataforma que ficará disponível online a partir de sua hospedagem em servidores na nuvem.

2.6 Modelagem de Layout

O layout do software tem como base o esquema de cores padrão da universidade que predomina com Azul, Branco, Cinza e Chumbo, e foi desenvolvido pensando na interação entre alunos e professores durante o andamento dos projetos. O layout é representado por uma prototipação feita com todas as telas da plataforma e tendo como tela inicial de todos os 3 layouts a tela de dashboard. Todo o protótipo foi desenvolvido na ferramenta Figma, sendo utilizado como base para a codificação da interface de usuário. Na Figura 2, observa-se o detalhe da tela de dashboard do aluno.

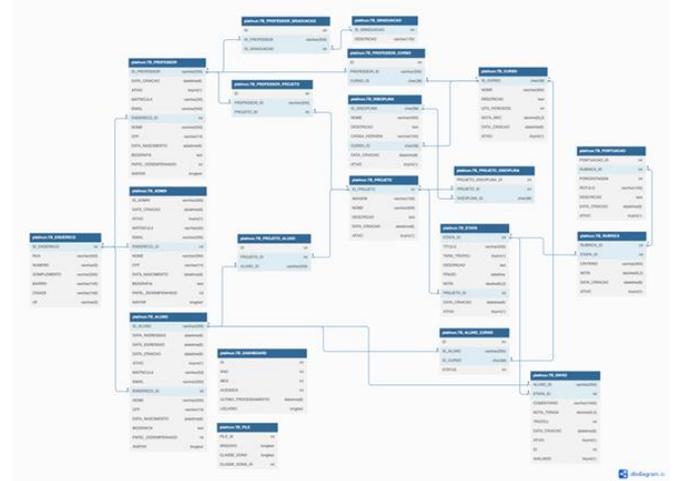
Figura 2. Tela de dashboard (aluno) (elaborado pelos autores)



2.7 Modelagem de Classes

Foi construída uma estrutura de armazenamento para o banco de dados que se encaixe neste requisito, onde há tabelas para o controle de arquivos, atividades, armazenamento de informações sobre os professores e também sobre formas de avaliação, conforme descrito na figura 3 a seguir, onde foi detalhado o diagrama de classes.

Figura 3. Diagrama de classes (elaborado pelos autores)



2.8 Containers Docker

Alguns dos recursos que detalhados nos tópicos a seguir foram configurados para serem executados por containers Docker, sendo que a utilização consiste em tornar quase todo tipo de aplicação compatível em multi plataforma, possibilitando utilizar a aplicação em Windows, Linux ou MacOS.

O principal motivo para a decisão de utilizar essa metodologia foi a facilidade de instalação e utilização das ferramentas, evitando, por exemplo, que seja necessário ter noção de comandos da linguagem em questão ou em outros casos fazer a instalação manual do SDK da linguagem, como em exemplos do .Net, Java, Node e Python. Logo, a utilização do Docker não existe a necessidade da instalação de nenhum pacote adicional, o container Docker foi auto-sustentável.

2.9 Backend - API RestFul

O desenvolvimento do Backend (server-side) da plataforma foi feito utilizando a IDE (Interface Development Environment) Visual Studio da Microsoft e a estrutura de controle de tráfego de requisições e manipulação de dados foi feita através de uma API no padrão RestFul, que foi desenvolvida na linguagem C# utilizando o framework Asp Net Core 6 [5]. O sistema foi desenvolvido com a arquitetura de projetos MVC (Model-View-Controller) por ser uma opção fortemente apoiada pelas empresas devido sua estrutura robusta, alto nível de confiabilidade na tipagem dos dados, além de suporte nativo pelas plataformas e ferramentas que foram utilizadas durante a construção do software. Então com a finalidade de se alcançar a alta disponibilidade multi-plataforma a API foi configurada para ser utilizada em um container Docker.

2.10 Gerenciador de usuários – Keycloak

Para todo o processo de gerenciamento, manipulação e gestão de usuários foi utilizado a ferramenta chamada Keycloak. O Keycloak é uma ferramenta open-source em Java de autoria da Red Hat que foi criada para ser um recurso gratuito para controle de usuários, semelhante a ferramentas hoje disponíveis no mercado como Amazon Cognito, OAuth2 e Azure AD. O Keycloak contém integração com os Social Login mais utilizados atualmente como Google, Instagram, Twitter entre muitos outros, porém para esse projeto foi utilizada apenas a integração com Google para utilização do email institucional. O Suporte utilizado nesse projeto foi apenas o controle de usuários, tela de Login e redefinição de senha.

2.11 Hangfire - Processador Programado Automatizado

No requisito do dashboard dos Administradores, foi solicitado um painel, no qual é possível visualizar o número de acessos anual dos usuários da plataforma, além do top 5 usuários com mais acesso no mês atual, porém os dados de acessos (login) não são de responsabilidade da API .Net. Eles foram controlados e armazenados pelo serviço de Gerenciamento de Usuários Keycloak, por esse motivo foi utilizada a biblioteca Hangfire que consiste em um automatizador de processos em segundo plano totalmente independente para que de forma programada a API captura os dados de login do Keycloak de 1 em 1 hora para alimentar a base de dados sobre acessos consumidos pelo Painel Administrativo. A biblioteca é um módulo da API .Net e contém um banco de dados dedicado para armazenar dados dos processos executados, como quando foi executado, quanto tempo demorou a execução e caso tenha recebido erros, tem a missão de detalhá-lo.

O Hangfire disponibilizou um dashboard para

visualização, monitoramento e acompanhamento dos processos que ele executa através de um gráfico temporal sobre o número de processos sucedidos/falhos por dia ou mês, um gráfico de execução em tempo real, além da listagem de todos os processos executados desde o início.

2.12 Frontend – React

A tecnologia client-side escolhida foi o Framework React por contar com uma comunidade muito ampla e bibliotecas que facilitaram os requisitos de software como usabilidade, clareza visual e funcionalidades limpas, sendo que o React, foi possível construir toda a interface do usuário do website de forma responsiva, devido a facilidade de manipulação de dados a partir dos componentes e também por contar com bibliotecas que contém templates disponíveis que auxiliaram na rapidez para a construção da parte visual do website e na construção da interface foi utilizado o editor de textos Visual Studio Code da Microsoft.

Para garantir um nível de segurança maior entre o poder administrativo e as permissões dos demais usuários foi utilizado a arquitetura Monorepo que consiste em possuir mais de uma código-fonte, em um mesmo repositório. A utilização dessa arquitetura, possibilitou elaborar dois websites, um deles dedicado apenas aos administradores, tendo o privilégio de possuir um subdomínio único, enquanto o outro dedicado aos professores e alunos sendo hospedado no mesmo website. Sem a arquitetura Monorepo seria muito trabalhoso possuir dois sites com funcionalidades semelhantes, componentes reutilizáveis e layout compatíveis. Na utilização do Monorepo foi possível ter três repositórios hospedados em um, sendo: área do admin; área dos demais usuários e componentes compartilháveis, sendo que todos utilizam os mesmos componentes, recursos e configurações e apenas é desenvolvido nas áreas as especificações únicas de cada website.

O acesso a usuários que não possuem um e-mail institucional foi bloqueado para a segurança da Wiki.

2.13 Azure Devops

O Software Azure Devops é um recurso disponibilizado pela Microsoft que tem como principal objetivo oferecer aos desenvolvedores uma plataforma capaz de centralizar funcionalidades úteis no processo de desenvolvimento de um produto. Alguns dos principais recursos oferecidos pelo Software: Repositórios de código fonte, Processos de Pipelines, Ferramentas de Metodologia Ágil e nesse projeto foram utilizados os repositórios de código fonte e os processos de pipelines.

No repositório de código fonte foram armazenados todo conteúdo resultante dos projetos como o site administrador, site usuário (professor/aluno), Keycloak, Wiki, API .Net Core e configuração do servidor. Por fim, o processo de Pipeline se baseia em uma sequência de tarefas que foram executadas após uma atualização acontecer em um dos códigos fonte, sendo possível configurar mais de um ambiente para um produto, muito conhecido como ambientes de desenvolvimento, homologação e produção e assim possível que cada ambiente tenha configurações diferentes. Para isso, é utilizado o Azure Library que foi um recurso adicional do Pipeline, muito utilizado para testes manuais antes de disponibilizar as novas funcionalidades para o

usuário final.

2.14 Banco de dados RDS

O RDS (Relational Database Service), é uma instância de um servidor dedicado a hospedar um banco de dados relacional, essa instância é responsável por oferecer armazenamento dos dados dos serviços: API .Net Core; Hangfire; Keycloak e Wiki.

3 RESULTADOS E DISCUSSÃO

A plataforma integrada universitária para gerenciamento de projetos em PBL desenvolvida reúne atividades acadêmicas de estudantes e equipes, buscando melhorar a comunicação e a colaboração entre os integrantes, de forma simples e intuitiva. A plataforma facilita o gerenciamento das atividades utilizando o modelo PBL, auxiliando os professores na comunicação e interação com os alunos em meios avaliativos, estimando o progresso e rendimento dos mesmos durante o seu desenvolvimento acadêmico.

3.1 Testes e Usabilidade

Com o objetivo de ter comprovações do funcionamento das funcionalidades da plataforma, foi elaborado um relatório evidenciando testes de sistema onde foram abordadas todas as funções que compõem os requisitos do sistema. Para avaliar a usabilidade e a coesão foi elaborado um formulário para captura de opiniões, levantamento dos dados de satisfação e considerações de usuários na aplicação. Previamente, foram liberados acessos de professor e aluno para um número limitado de usuários, a fim de realizar os testes.

Nas figuras de 4 e 5 foram detalhadas as métricas que indicaram o nível de satisfação dos usuários, após a utilização da plataforma, além de considerações sobre falhas e/ou sugestões de melhorias da plataforma.

Figura 4. Desempenho Geral da Plataforma (Elaborado pelos Autores)

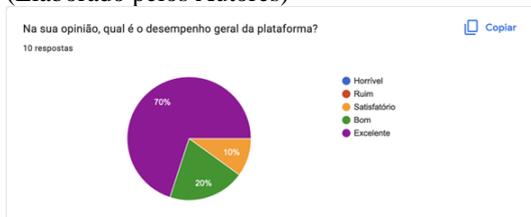


Figura 5. Avaliação da Funcionalidade da Plataforma (Elaborado pelos Autores)



2.16 Desempenho

Foi desenvolvido um teste de desempenho utilizando a ferramenta JMeter, esta ferramenta tem como objetivo simular múltiplas requisições para endpoint e gerar relatórios de disponibilidade, para o teste devemos levar em conta as seguintes configurações.

- 03 requisições de login (1 para cada tipo de usuário)
- 25 usuários alunos, 5 requisições cada um
- 25 usuários professores, 5 requisições cada um
- 05 usuários administradores, 5 requisições cada um

Cada um dos pacotes de requisição foi feito para mais de um endpoint por vez e totalizaram 903 requisições durante o teste completo. Na Figura 6, a seguir observa-se o detalhe do resultado do teste de desempenho.

Figura 6. Resultados do teste de desempenho (elaborado pelos autores)

Rótulo	# Amostra	Média	Min	Max	Desvio padrão	Erro %	Taxa de transferência	KB/sec Recebidos	KB/sec Enviados	Média Bytes
Get Token Login Professor	1	1039	1039	1039	0.00	0.000%	.96246	2.94	0.33	3131.0
Get Token Login Aluno	1	923	923	923	0.00	0.000%	108.342	3.27	0.37	3094.0
Get Token Login Admin	1	1157	1157	1157	0.00	0.000%	.86430	2.73	0.29	3233.0
Dashboard Aluno	125	299	184	792	198.06	0.000%	3.011.323	57.43	52.23	1953.0
Dashboard Professor	125	299	184	831	197.82	0.000%	3.065.980	58.48	54.28	1953.0
Perfil Professor	125	199	182	326	23.43	0.000%	3.614.806	28.17	63.75	798.0
Perfil Aluno	125	228	206	356	25.12	0.000%	3.493.572	34.73	60.35	1018.0
Get Projeto by Id	250	198	181	330	21.26	0.000%	7.028.395	42.83	123.17	624.0
Dashboard Admin	25	1328	973	2356	443.86	0.000%	217.770	0.96	4.07	453.0
Listagem Alunos	25	216	210	223	3.34	0.000%	267.838	3.76	5.00	1437.0
Listagem Disciplinas	25	217	208	293	15.65	0.000%	268.097	0.95	5.02	361.0
Listagem Cursos	25	214	211	218	1.50	0.000%	268.269	0.96	5.01	368.0
Listagem Professores	25	217	213	227	3.46	0.000%	268.298	4.14	5.02	1579.0
Perfil Admin	25	190	186	206	3.72	0.000%	269.078	2.07	5.01	788.0
TOTAL	903	266	181	2356	229.52	0.000%	6.498.273	70.65	114.70	1113.4

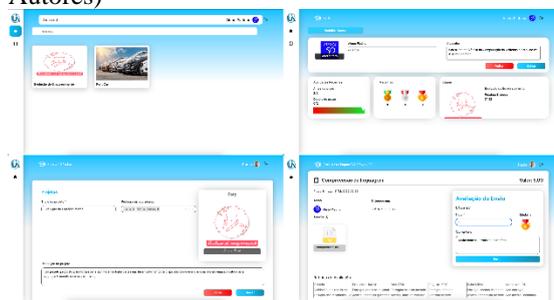
O resultado final testado da plataforma conta com três tipos de usuários, sendo o aluno, professor e o administrador. O aluno tem acesso aos projetos e atividades disponibilizados a ele, sempre que for necessário um novo projeto em que o aluno trabalhará será colocado em sua seção de Dashboard, o aluno pode subir arquivos para a plataforma, sendo arquivos de texto, imagens e ou arquivos compactados. Além dos projetos o aluno também conta com métricas e pontuações através de medalhas (ou badges), sempre que o aluno completar e entregar um projeto ou trabalho ele é avaliado pelo professor e recebe uma medalha de acordo com os critérios atingidos ao final de cada projeto finalizado e entregue. O Professor pode criar e alterar os projetos e trabalhos nos quais ele administra, e diferentemente do aluno, que pode ter vários projetos e trabalhos de diferentes professores, o professor só terá acesso aos trabalhos que ele administra.

O professor, ao verificar o envio da atividade por parte do aluno, irá atribuir a nota e comentar sobre o trabalho enviado. O valor da nota será seguindo a rubrica de avaliação se o

aluno cumprir com todos os critérios da avaliação receberá 100% (cem por cento da nota), se cumprir com apenas metade receberá 50% (cinquenta por cento da nota) e se cumprir com apenas um critério receberá 25% (vinte e cinco por cento da nota). Assim, o professor enviará para o aluno o seu resultado da atividade junto à sua nota, comentário e medalha.

O administrador é responsável por cadastrar e regularizar as configurações tanto dos alunos quanto dos professores, ele é quem configura as permissões e funções dos usuários da plataforma, atribuindo as disciplinas e projetos cadastrados no sistema pelos professores para cada aluno cadastrado em determinado projeto corretamente

Figura 7. Telas do Sistema (Elaborado pelos Autores)



CONCLUSÃO

A Plataforma Integrada Universitária para Gerenciamento de Projetos em PBL (PLATINUN) facilita e ajuda na aplicação de atividades acadêmicas de estudantes e equipes em modelo PBL, auxilia os professores na comunicação e interação com os alunos em meios avaliativos, estima o progresso e rendimento dos mesmos durante todo o seu desenvolvimento acadêmico, possuindo flexibilidade para a incorporação de novas funcionalidades e atividades relacionadas à disciplina gerenciada. O trabalho desenvolvido entrega à Universidade Prof. Edson Antônio Velano uma plataforma inovadora e proprietária para uso contínuo em diversos cursos e atividades acadêmicas.

REFERÊNCIAS

[1] GNAUR, Dorina; HYTTEL, Hans. A Platform for Developing and Maintaining Competences in PBL Supervision. Livro Emerging Technologies for Education, v.1, n.1, p. 1-8, Jan/Jan. 2017.

[2] HASIBUAN, Siti Rohaniah; DAMANIK, Marini; NASUTION, Hafni Indriati. Differences in Learning Models of Problem Based Learning and NHT Cooperative Type with Card Media Assistance to Student Learning Outcomes and Activities in Naming Chemical Compounds. Journal of Transformative Education and Educational Leadership, v. 1, n. 1, p. 18-22, 2019.

[3] MALMIA, Wa et al. Problem-Based Learning As An Effort To Improve Student Learning Outcomes. International Journal of Scientific e Technology Research Volume. [s.l.], v.8, n.9, p. 1-4, Set/Set. 2019.

[4] CAMPOS, Alyson Wegilles de Souza. O USO DO DESIGN RESPONSIVO NO DESENVOLVIMENTO DE WEBSITES: aplicação dos métodos na construção de um website. Repositório da Universidade Federal de Pernambuco. Caruaru, v.1, n.1, p. 1-62, Jan/Dez. 2017.

[5] ROTH, Daniel; ANDERSON, Rick; LUTTIN, Shaun. Documentação do ASP.NET Core. Disponível em: <<https://docs.microsoft.com/pt-br/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-5.0>>. Acesso em: 8 out. 2021.