

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

FELIPE AUGUSTO VASCONCELOS E SILVA
Orientador: Saul Emanuel Delabrida Silva

**DESENVOLVIMENTO DE UMA API PARA APLICAÇÃO DE
LEARNING ANALYTICS EM DADOS DE EXPERIÊNCIAS DE
ESTUDANTES EM PLATAFORMAS DE JOGOS EDUCACIONAIS**

Ouro Preto, MG
2023

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

FELIPE AUGUSTO VASCONCELOS E SILVA

**DESENVOLVIMENTO DE UMA API PARA APLICAÇÃO DE LEARNING
ANALYTICS EM DADOS DE EXPERIÊNCIAS DE ESTUDANTES EM
PLATAFORMAS DE JOGOS EDUCACIONAIS**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Saul Emanuel Delabrida Silva

Ouro Preto, MG
2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S586d Silva, Felipe Augusto Vasconcelos E.

Desenvolvimento de uma API para aplicação de Learning Analytics em dados de experiências de estudantes em plataformas de jogos educacionais. [manuscrito] / Felipe Augusto Vasconcelos E Silva. - 2023. 34 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Saul Emanuel Delabrida Silva.

Monografia (Bacharelado). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Graduação em Ciência da Computação .

1. Jogos educativos. 2. Metodologia do ensino. 3. Educação. I. Silva, Saul Emanuel Delabrida. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004

Bibliotecário(a) Responsável: Luciana De Oliveira - SIAPE: 1.937.800



FOLHA DE APROVAÇÃO

Felipe Augusto Vasconcelos e Silva

DESENVOLVIMENTO DE UMA API PARA APLICAÇÃO DE LEARNING ANALYTICS EM DADOS DE EXPERIÊNCIAS DE ESTUDANTES EM PLATAFORMAS DE JOGOS EDUCACIONAIS

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Ciência da Computação

Aprovada em 20 de Março de 2023.

Membros da banca

Saul Emanuel Delabrida Silva (Orientador) - Doutor - Universidade Federal de Ouro Preto
Jadson Castro Gertrudes (Examinador) - Doutor - Universidade Federal de Ouro Preto
Reinaldo Silva Fortes (Examinador) - Doutor - Universidade Federal de Ouro Preto
Robson Santiago Viol (Examinador) - Graduado - Universidade Federal de Ouro Preto

Saul Emanuel Delabrida Silva, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 21/03/2023.



Documento assinado eletronicamente por **Saul Emanuel Delabrida Silva, PROFESSOR DE MAGISTERIO SUPERIOR**, em 21/03/2023, às 11:43, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0493033** e o código CRC **FDE583CA**.

Dedico este trabalho aos meus pais e a minha irmã, por todo apoio e incentivo durante minha trajetória.

Agradecimentos

Agradeço primeiramente aos meus pais, Emar e Jussara, e em especial à minha irmã, Ana Luiza, que sempre me incentivou e me apoiou nos dias difíceis. Agradeço também aos meus irmãos de Ouro Preto, Gabriel Matoso e Frederico Martins, pela amizade, por compartilharem momentos incríveis e por fazerem parte do meu crescimento. Também gostaria de agradecer à minha amiga Flávia Nunes, por todos os conselhos, ensinamentos e troca de experiências que foram essenciais para meu desenvolvimento pessoal e profissional. Agradeço também aos professores por todos os ensinamentos, e principalmente ao professor Saul Delabrida que esteve presente em grande parte da minha trajetória acadêmica.

Resumo

Os dados gerados no mundo nos últimos anos vêm se tornando um assunto muito importante quando falamos de obter informações para tomadas de decisões. Diversos setores empresariais, industriais, dentre outros, buscam cada vez mais solucionar problemas e impulsionar os negócios através de *insights* extraídos de análises de dados. Na educação, a *Learning Analytics* combina diversos métodos de análise de dados para gerar informações, com o propósito de fornecer aos profissionais da educação mais conhecimento sobre os perfis de seus alunos para, assim, realizar melhorias em suas metodologias de ensino. Este trabalho busca contribuir em uma nova área de pesquisa relacionada aos trabalhos de Nunes (2022) e Melo (2022), que tratam da implementação de um jogo educacional e uma plataforma de gerenciamento de dados relacionados ao jogo, respectivamente. Assim, como uma forma de agregar tais trabalhos, foi desenvolvida uma *API* que aplica métodos de análises sobre os dados de experiência dos usuários do jogo, para gerar informações sobre o comportamento e desempenho dos estudantes que utilizarem a plataforma. Dessa forma, a *API* desenvolvida conta com rotas que serão capazes de aplicar estudos dos dados gerados pelos usuários e, assim, fornecer resultados para que os clientes possam gerar gráficos e relatórios que servirão de ferramenta para auxiliarem profissionais da educação a compreender os perfis de seus alunos.

Palavras-chave: *Learning Analytics*. *API*. *xAPI*. Educação.

Abstract

The data generated in the world in recent years has become a very important issue when we talk about obtaining information for decision making. Several business and industrial sectors, among others, increasingly seek to solve problems and sustain business through *insights* extracted from data analysis. In education, *Learning Analytics* combines several methods of data analysis to generate information, with the purpose of providing education professionals with more knowledge about the profiles of their students, thus making improvements in their teaching methodologies. This work seeks to contribute to a new area of research related to the works of Nunes (2022) and Melo (2022), which deal with the implementation of an educational game and a data management platform related to the game, respectively. Thus, as a way of aggregating such works, an *API* was developed that applies analysis methods on the experience data of game users, to generate information about the behavior and performance of students using the platform. In this way, the *API* developed has routes that will be able to apply studies of data generated by users and, thus, provide results so that customers can generate graphs and reports that will serve as a tool to help education professionals to understand your student profiles.

Keywords: Learning Analytics. API. xAPI. Education.

Lista de Ilustrações

Figura 2.1 – Formato de registro de dados pela <i>xAPI</i>	6
Figura 3.1 – Representação do ambiente virtual do jogo (NUNES, 2022).	10
Figura 3.2 – Representação dos componentes disponíveis para interação do aluno no jogo (NUNES, 2022).	11
Figura 3.3 – Representação dos componentes disponíveis para interação do aluno no jogo (NUNES, 2022).	11
Figura 3.4 – Formato <i>JSON</i> construído e armazenado pela <i>xAPI</i>	12
Figura 4.1 – Arquitetura <i>Learning Analytics API</i>	15
Figura 4.2 – Formato de arquivo <i>JSON</i> a ser enviado junto a requisição para <i>LA API</i>	16
Figura 4.3 – Tabela rotas existentes na <i>LA API</i>	18
Figura 4.4 – Dados de configuração para a rota <i>timeExecution</i>	18
Figura 4.5 – Resultado obtido através da requisição à rota <i>timeExecution</i>	18
Figura 4.6 – Gráfico de tempo gasto por tarefas realizadas por um aluno.	19
Figura 4.7 – Dados de configuração para a rota <i>performance</i>	20
Figura 4.8 – Resultado obtido através da requisição à rota <i>performance</i>	20
Figura 4.9 – Gráfico do desempenho do aluno no questionário em relação ao desempenho médio da turma.	21
Figura 4.10–Dados de configuração para a rota <i>behavior</i>	22
Figura 4.11–Resultado obtido através da requisição à rota <i>behavior</i>	22
Figura 4.12–Gráfico do fluxo do comportamento do aluno ao realizar as tarefas.	23
Figura 4.13–Dados de configuração para a rota <i>correlation</i>	24
Figura 4.14–Resultado obtido através da requisição à rota <i>correlation</i>	24
Figura 4.15–Gráfico da correlação entre o desempenho dos alunos e o tempo gasto realizando a tarefa de assistir vídeos.	24
Figura 4.16–Dados de configuração para a rota <i>clustering</i>	25
Figura 4.17–Resultado obtido através da requisição à rota <i>clustering</i>	25
Figura 4.18–Gráfico de agrupamento por perfís semelhantes.	26
Figura 5.1 – Página inicial da documentação da <i>LA API</i>	28
Figura 5.2 – Página do método de comportamento da documentação da <i>LA API</i>	29
Figura 5.3 – Página do método de agrupamento da documentação da <i>LA API</i>	30

Lista de Abreviaturas e Siglas

UFOP	<i>Universidade Federal de Ouro Preto</i>
API	<i>Application Programming Interface</i>
FCM	<i>Fuzzy c-means</i>
IDE	<i>Integrated Development Environment</i>
LA	<i>Learning Analytics</i>
LMS	<i>Learning Management System</i>
LRS	<i>Learning Records Store</i>
xAPI	<i>Experience Application Programming Interface</i>

Sumário

1	Introdução	1
1.1	Justificativa	2
1.2	Objetivos	2
1.3	Organização do Trabalho	3
2	Revisão Bibliográfica	4
2.1	Fundamentação Teórica	4
2.1.1	Learning Analytics	4
2.1.2	API	4
2.1.3	xAPI	5
2.1.4	<i>Clustering</i>	5
2.2	Trabalhos Relacionados	6
2.3	Lições Aprendidas	8
3	Estudo de Caso	9
4	Desenvolvimento	13
4.1	Base de Dados	13
4.2	Learning Analytics API	14
4.2.1	Arquitetura	14
4.2.2	Rotas da <i>LA API</i>	17
4.2.2.1	Método de extração de tempo de execução	17
4.2.2.2	Métodos de extração do Desempenho do Questionário	20
4.2.2.3	Método de extração do comportamento de execução de tarefas	21
4.2.2.4	Método de extração da correlação de atividade e desempenho	23
4.2.2.5	Método de agrupamento por semelhança no perfil de aluno	26
5	Resultados	28
6	Considerações Finais	31
6.1	Conclusão	31
6.2	Trabalhos Futuros	31
	Referências	33

1 Introdução

Nas últimas décadas, muito se tem discutido a respeito da quantidade de dados gerados no mundo em meios digitais. Esses conjuntos de dados possuem grande valor ao auxiliarem nas tomadas de decisões. Por meio da análise desses dados, tornou-se possível prever, cada vez mais com maior confiabilidade, comportamentos e tendências em diversas áreas, como por exemplo do mercado financeiro, indústrias, dentre outros, como discutido em [Cao \(2017\)](#).

O crescimento sobre essa discussão também é notado quando o assunto é dados educacionais. O termo *Learning Analytics* é utilizado para definir a área de aplicação de estudos e técnicas sobre dados adquiridos em ambientes educacionais, com o objetivo de extrair informações relevantes para que os profissionais da educação possam desenvolver planos e estratégias para realizarem melhorias contínuas por meio de uma metodologia de ensino, como foi abordado nos trabalhos de [Lee, Cheung e Kwok \(2020\)](#) e [Gašević, Dawson e Siemens \(2015\)](#).

Diversas instituições de ensino utilizam-se de plataformas de gerenciamento de dados, *Learning Management System (LMS)*, como uma forma reunir, em um só ambiente, a possibilidade de administração de cursos e gestão de dados, definido por [Ifenthaler \(2012\)](#). No entanto, essas plataformas ainda não estão desenvolvidas o bastante para realizar funcionalidades além do que um sistema básico de gerenciamento proporciona [Islam e Mahmud \(2020\)](#). Com isso, torna-se possível explorar esse campo e combiná-lo a aplicações de *Learning Analytics* capazes transformar essa rede mais inteligente.

No trabalho de [Bernard et al. \(2019\)](#), é discutido sobre a forma de ensino individualizado e como tal metodologia é capaz de melhorar significativamente o processo de aprendizagem de estudantes. Compreender os diversos perfis de alunos e identificar suas necessidades específicas pode gerar espaços que irão atender melhor cada aluno. Dessa forma é importante entender quais seriam os métodos de ensino capazes de serem mais assertivos, a fim de aumentar os níveis de conhecimentos de um grupo de alunos. Como forma de preencher a lacuna deixada pelo método tradicional de ensino, a tecnologia pode ser uma ferramenta importante para realizar análises que irão transformar a maneira de identificar o perfil de aprendizagem individual de cada estudante, promovendo novas abordagens e melhorias contínuas nas metodologias de ensino.

Dessa forma, este trabalho tem o intuito de desenvolver uma *Application Programming Interface (API)*, que servirá de ferramenta para que educadores possam obter informações sobre o desempenho e comportamento de seus alunos a partir dos dados de experiências coletados das plataformas educacionais. Para isso, foi utilizado, a partir de um estudo de caso, o trabalho de [Nunes \(2022\)](#) e de [Melo \(2022\)](#), que propuseram a criação de uma plataforma de jogos sérios, no qual os usuários são capazes de realizar aulas e atividades avaliativas em um ambiente de Realidade Virtual, combinado a um sistema de LMS, que realiza toda gestão e armazenamento

dos dados relacionados ao jogo.

1.1 Justificativa

Com a individualidade no processo de desenvolvimento na aprendizagem de estudantes, a educação tradicional pode acabar não dando suporte às necessidades específicas de cada aluno, como pode ser observado no trabalho de [Jernstedt \(1976\)](#). O estudo que mostra um experimento ao aplicar metodologias de ensino de forma tradicional e especializada para um grupo de alunos, concluiu-se que alunos que tiveram um tratamento individualizado obtiveram melhor desempenho em comparação aqueles que receberam o tratamento tradicional.

Sendo assim, podemos notar que a educação tradicional pode enfrentar dificuldades em atender uma quantidade significativa de estudantes, visto que existem diversos perfis de aprendizagem e cada aluno possui suas particularidades. Para propor uma metodologia que vai atender um grupo maior dos alunos e abraçar essa individualidade, é necessário primeiro compreender o comportamento dos alunos.

Como uma forma de contribuir aos trabalhos de [Nunes \(2022\)](#) e [Melo \(2022\)](#), os quais combinaram o desenvolvimento de uma plataforma de jogo sério e um sistema de gestão de dados integrado, este trabalho se propõe na criação de uma *API de Learning Analytics* com o propósito de transformar dados em informações úteis para tomadas de decisão. Assim, possibilitando os profissionais da educação um ambiente que os auxiliem em melhorias contínuas ao desenvolver metodologias de ensino com base nos resultados obtidos. Dessa forma, a *Learning Analytics API* tornará possível a transformação de dados de experiência dos alunos em informações que poderão ser utilizadas para gerar relatórios e gráficos.

1.2 Objetivos

O objetivo deste trabalho consiste no desenvolvimento de uma *API de Learning Analytics* que receberá dados de experiências, obtidos através da interação de alunos com uma plataforma digital educativa, com o intuito de aplicar métodos de análises de dados para extração de informações. Dessa forma, com os resultados obtidos, a *Learning Analytis API* poderá fornecer aos clientes dados para que se possa criar relatórios e gráficos. Apresentando como objetivos específicos temos:

- Desenvolver uma base de dados sintética de experiências de alunos. Este projeto utiliza-se como base os trabalhos de [Nunes \(2022\)](#) e [Melo \(2022\)](#). No entanto, tais trabalho ainda não foram utilizados com o intuito de gerar uma base de dados reais. Dessa forma, se tornou necessário o desenvolvimento de uma base de dados sintética para que fosse possível o desenvolvimento da *API* e a realização de prova de conceito.

- Implementar uma API com Linguagem Python que aplicará análises de dados sobre os dados extraídos do jogo. Cada rota implementada na *API* representará um estudo específico que poderá ser aplicado sobre os dados.
- Implementar uma aplicação de interface Web como uma forma de documentação da *Learning Analytis API*, como objetivo de demonstra o valor de cada método desenvolvido, além de realizar uma prova de conceito ao consumir as informações fornecidas pela API desenvolvida.

1.3 Organização do Trabalho

A estrutura proposta para este trabalho de monografia é composta por seis capítulos. No Capítulo 1 são apresentados a introdução do projeto contendo o problema, justificativa, objetivo e organização do trabalho. No Capítulo 2 é apresentada uma revisão bibliográfica composta por uma pesquisa realizada da fundamentação teórica e trabalhos relacionados. No capítulo 3 é apresentado o estudo de caso que motivou este projeto. No Capítulo 4 é apresentado o desenvolvimento do trabalho. No Capítulo 6 são apresentados os resultados obtidos. No Capítulo 6 são apresentadas a conclusão e trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo serão abordados assuntos relevantes para o desenvolvimento deste trabalho. Na Seção 2.1, serão apresentados alguns conceitos como *Learning Analytics*, *APIs*, *xAPI*, Algoritmo de Clusterização. Na Seção 2.2 serão discutidos os Trabalhos Relacionados.

2.1 Fundamentação Teórica

Nesta seção apresentaremos a fundamentação teórica para o desenvolvimento e entendimento deste trabalho.

2.1.1 Learning Analytics

A *Learning Analytics* é uma área na tecnologia que combina diversas ferramentas e métodos, como mineração de dados, aprendizado de máquina e visualização de dados. Para analisar dados educacionais a partir de logs gerados por meio de interações de estudantes com plataformas educacionais é importante desenvolver formas de extrair e gerar informações sobre o desenvolvimento dos estudantes e para tomadas de decisões, como é discutido no trabalho de Gašević, Dawson e Siemens (2015). Em seu processo para gerar informações Chatti et al. (2012) explica que a divisão é feita por meio de coleta de dados extraídos de plataformas educacionais, pré-processamento com o objetivo de aplicação de métodos para limpeza e transformação da base de dados, análise para extração de informações e pós-processamento no qual ocorre atualização dos dados a fim de promover melhorias contínuas dos resultados.

2.1.2 API

No trabalho desenvolvido por Ofoeda, Boateng e Effah (2019), foi realizado um levantamento dos estudos feitos nos últimos anos sobre a evolução das Interfaces de Programação de Aplicações, *APIs*. Em sua definição discutida em seu trabalho, relaciona o termo *API* como uma forma de conectarmos diversas aplicações em uma linguagem de comunicação comum. Assumindo com o objetivo a amplificação de suporte e utilização de funcionalidades computacionais definidas em suas arquiteturas para resolver problemas para diferentes tipos de plataformas.

Para este trabalho, será utilizado o formato de *API REST*. Soni e Ranga (2019) define tal API como um dos serviços web mais utilizados para transferência de informações, além de apresentar como principais características a flexibilidade, a independência entre o cliente e servidor, e utilizar protocolos *HTTP*, como formato de comunicação através rotas, que utilizam-se de métodos, como o método de solicitação *POST*.

O formato de dado *JSON* é utilizado como resposta da *API REST*. Ele se resume em uma estrutura como um dicionário chave-valor muito popular em aplicações e serviços Web (BOURHIS; REUTTER; VRGOČ, 2020). Esse formato possibilita que, independente das linguagens de programação, aplicações sejam capazes de trocar informações entre si, se tornando um meio de comunicação fácil e rápido, já que possui um formato de dados leve.

2.1.3 xAPI

Dessa forma, há diversos tipos de *APIs* para resolver e lidar com problemas. Para o projeto, além do desenvolvimento da *API* responsável por processar e analisar dados, iremos lidar com a *Experience API*, utilizada para lidar com dados extraídos de plataformas educativas sobre informações de desempenho dos estudantes. Como discutido em Kevan e Ryan (2016), a partir das experiências dos usuários, a *xAPI* captura esses dados, de forma estruturada e a partir de um dicionário de ações. Em sua construção, cada formato dos dados é dividido em chaves de identificação importantes. Separadas como: o ator, no qual apresenta informações de identificação de cada aluno que está realizando uma iteração na plataforma; o verbo, que representa qual foi a ação realizada pelo usuário; o objeto, que representa qual foi o item que o aluno realizou a ação; o resultado, que apresenta informações sobre desempenho da ação realizada; além de informações com registro de data e hora da ação realizada. Como pode ser observado na Figura 2.1, que representa um exemplo de como os dados são registrados pela *xAPI*.

2.1.4 Clustering

Neste trabalho, foi utilizado o algoritmo de agrupamento, *Fuzzy c-means*, (*FCM*). A utilização de tal modelo demonstra uma maneira de explorar ainda mais técnicas de extração de informação sobre dados de experiência de alunos. Secretan, Wild e Guest (2019), discute que a área de *Learning Analytics* tem uma necessidade de atenção no aprimoramento das técnicas de análises de dados para alcançar melhores resultados.

Dessa forma, algoritmos de agrupamento podem ser úteis quando se fala em classificação de grupos por características semelhantes. Gaertler (2005) sintetiza a definição do termo *Clustering*, que se pode entender como uma forma não-supervisionada de realizar divisões de grupos por suas características semelhantes. As similaridades são reconhecidas por uma área cujo o ponto central representa o maior grau de similaridade entre os objetos analisados. A partir do cálculo da distância é possível definir o grau de pertencimento de um determinado grupo (XU; TIAN, 2015). Nesse trabalho, exemplifica uma das diversas variações desses algoritmos, como o K-mens que realizará diversas interações, atualizando os valores dos parâmetros até encontrar uma solução ótima, ou seja, que consiste em definir centros em um plano que irá representar cada um dos grupos da melhor forma possível.

Já o algoritmo utilizado em um dos métodos implementados na *API* para *Learning*

```
{
  "actor": {
    "name": "Sally Glider",
    "mbox": "mailto:sally@example.com"
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/experienced",
    "display": { "en-US": "experienced" }
  },
  "object": {
    "id": "http://example.com/activities/solo-hang-gliding",
    "definition": {
      "name": { "en-US": "Solo Hang Gliding" }
    }
  }
}
```

Figura 2.1 – Formato de registro de dados pela xAPI.

Fonte: <<https://xapi.com/statements-101/>>

Analytics foi o FCM. Xu e Tian (2015) descreve o FCM com uma característica de apresentar para cada ponto o grau de pertencimento relacionado a todos os grupos. Ou seja, identificando a probabilidade que cada objeto pode apresentar ao pertencer em todos os grupos. Além disso, o trabalho apresenta como um dos benefícios do FCM é a relação com a realidade, que um objeto mesmo que tenha maior similaridade com um certo grupo, ele ainda possui características que diferentes ao grupo. O algoritmo foi proposto por Bezdek, Ehrlich e Full (1984).

2.2 Trabalhos Relacionados

No trabalho de Kazanidis, Pellas e Christopoulos (2021), foi realizado o trabalho conceitual ao aplicar *Learning Analytics* nos cenários de jogos educativos em Realidade Aumentada. Nele é possível perceber que a utilização dessa ferramenta permite auxiliar de forma significativa o trabalho dos professores e dos orientadores em identificar a evolução da aprendizagem dos estudantes. Isso permite uma melhor visão das estratégias para tomada de decisões com o objetivo de melhorar o ensino através dos dados analisados por meio de mineração de dados e aprendizado de máquina. O projeto proposto também discute a importância do entendimento do cenário a ser aplicado para que os resultados sejam alcançados, como por exemplo a identificação dos perfis dos usuários. Além disso, são discutidas as técnicas para visualização das informações extraídas dos dados coletados, como gráficos e relatórios. Os autores concluíram que essa abordagem gera impactos positivos ao promover informações relevantes para evolução do ensino, no entanto ainda são necessários mais estudos na área.

A utilização de *Learning Analytics* por Secretan, Wild e Guest (2019), combinada com a aplicação de realidade aumentada tem com foco em inovações tecnológicas na área da educação.

Um recurso importante abordado em seu trabalho, foi o uso da *Experience API*, *xAPI*, que registra as informações coletadas pelas interações dos usuários ao realizarem as tarefas. A configuração dos dados armazenados segue o formato: quem realizou a ação, qual foi a ação e qual o objeto que houve interação. Dessa forma, podemos obter uma informação de desempenho de um usuário ao analisar a sua ação e competência ao executar as tarefas. No entanto, não podemos dizer que uma ação gera uma competência, mas sim que uma competência só é desenvolvida a partir de uma sequência de ações. Tal fato, torna-se um grande desafio ao identificar se o usuário atingiu um determinado nível de competência que irá resultar em uma melhora no aprendizado. Os autores também acreditam que deve haver mais estudos nessa área, como por exemplo o uso de técnicas de ciência de dados e *machine learning* e seu grande potencial para atingir resultados cada vez mais assertivos.

Para tornar o fluxo de aprendizado cada vez mais próximos da realidade dos estudantes em disciplinas de programação, [Ardimento et al. \(2019\)](#) apresentou um estudo sobre o desenvolvimento do ambiente *CodeMiner*. A partir disso, foi utilizado de algoritmos de inteligência artificial como a Lógica de *Fuzzy*, que tem como objetivo de capturar dados durante o uso de *IDEs*, ambiente de desenvolvimento integrado, para acompanhar, analisar e gerar informações relevantes na tomada de decisão durante o processo de aprendizagem, a partir dos *logs* capturados. Os *logs* são minerados para extrair informações a respeito das habilidades e desempenhos. Apresentando como resultados pontos positivos sobre a abordagem, como por exemplo um potencial para se tornar uma ferramenta que auxiliará a compreensão dos professores em como se dá o comportamento e os resultados dos alunos.

No trabalho de [Pandolfo et al. \(2020\)](#), uma maneira de tornar mais eficiente o processo de gerenciamento da triagem de um hospital, é a utilização de algoritmo com lógica *Fuzzy* para classificar pacientes em seus respectivos graus de urgências. Foi discutido que o atendimento da triagem não poderia ser apenas um processo automatizado, mas também deveria apresentar uma abordagem analítica sobre os dados analisados para fornecer cada vez mais resultados assertivos. Dessa forma, o atendimento se torna mais dinâmico evitando uma possível superlotação nos hospitais.

Para prever o desempenho dos alunos em instituições de ensino superior, o trabalho de [Sohail, Khanum e Alvi \(2018\)](#) utilizaram o modelo estatístico *Fuzzy* híbrido. A utilização de ferramentas online para disponibilização de materiais didáticos e aplicação de atividades avaliativas, como por exemplo os Sistemas de Gestão de Aprendizagem, *LMS*, podem fornecer grandes quantidades de bases de dados para a utilização de *Learning Analytics*. Essa abordagem pode implicar em previsões que auxiliarão no acompanhamento educacional dos estudantes por meio de feedbacks personalizados. A utilização do método da lógica de *fuzzy* para a realização das previsões se dá pelo fato de que os dados utilizados são variáveis linguísticas e que modelos matemáticos não seriam adequados para resolver este problema. Em sua conclusão, o modelo utilizado apresentou resultados eficientes e de fácil implementação, no entanto, um fator limita-

dor discutido foi a dificuldade de se lidar com previsões estatísticas levando em consideração diferentes disciplinas.

2.3 Lições Aprendidas

Uma área capaz de realizar impactos positivos nas metodologias de ensino vem se tornando muito popular nos últimos anos. A *Learning Analytics* combina diversos métodos de análises de dados de experiências de alunos, ao realizarem tarefas em plataformas educacionais, com o objetivo de gerar *insight* para tomadas de decisão com foco na melhoria do ensino. Em diversos trabalhos é notável os resultados gerados no cenário educacional. Entender cada vez mais o perfil de cada aluno para que se possa criar metodologias menos generalizadas vem se tornando cada vez mais discutido. Dessa forma, muitos estudos apontam como a área de *Learning Analytics* tem muito a ser explorada, como por exemplo a utilização de algoritmos mais robustos para aumentar o poder ação e revolucionar o meio educacional com ajuda da tecnologia.

3 Estudo de Caso

Neste capítulo será realizado um estudo de caso, cujo objetivo é analisar e utilizar como base para o desenvolvimento deste trabalho, os trabalhos realizados por Nunes (2022) e por Melo (2022).

No trabalho de Nunes (2022) foi proposto o desenvolvimento de uma plataforma educativa, classificada pelo termo de jogos sérios em Realidade Virtual (VR). O projeto elaborado tem como função dar suporte às metodologias de ensino, como aulas introdutórias de laboratórios para estudantes de biologia. Além disso, outro trabalho que compõem tal projeto foi desenvolvido por Melo (2022), no qual foi proposta e aplicado uma arquitetura de Sistema de Gestão da Aprendizagem (LMS) capaz de fornecer aos professores uma plataforma de gerenciamento das tarefas inseridas ao jogo, além de aplicar um sistema de *Learning Records Store (LRS)*, que realiza a coleta e armazenamento dos dados dos alunos ao interagirem com a plataforma.

O jogo desenvolvido por Nunes (2022), busca a simulação de aulas virtuais imersivas utilizando-se da tecnologia de VR, como pode ser ilustrado pela Figura 3.1. A aula que compõem uma das fases do jogo permite que o aluno seja capaz de realizar atividades preparatórias relacionadas ao processo de limpeza e manuseio de um microscópio ótico. As atividades podem ser divididas em assistir vídeos, ver slides, ouvir áudios, ler textos, além de interagir com objetos virtuais. Por fim, como uma forma de validar e avaliar o desempenho do aluno nos estudos ao interagirem com a plataforma, também é aplicada uma atividade avaliativa, como forma de questionário, para que a fase seja concluída. Alguns dos exemplos dos componentes de interação para realizar as atividades no jogo podem ser visualizadas na Figura 3.2

Cada interação realizada por um aluno nesse jogo pode implicar em registros de suas experiências. Tais registros são compostos por dados que podem representar, por exemplo, qual atividade está sendo realizada, qual aluno está interagindo com determinada tarefa, qual o desempenho que o aluno obteve ao executar o questionário, dentre várias outras. A partir disso, o trabalho realizado por Melo (2022), por sua vez, teve como objetivo integrar um sistema LRS. Essa integração utiliza-se da *xAPI* para realiza a coleta e armazenamento, em um sistema LMS, dos dados de experiências dos usuários ao interagirem com a plataforma.

A *xAPI* é capaz de armazenar os dados de experiência dos usuários em um formato *JSON* previamente definido. A estrutura desse formato de dado pode ser observada na Figura 3.4. No instante que o usuário realiza uma ação no jogo, como exemplificada na Figura 3.3, um *log* de registro é construído e armazenado, representando que no instante momento o aluno realizou a ação de visualizar um slide.

O formato da configuração em *JSON* pode ser dividida em chaves. A chave “*id*” representa o código de identificação da aula. A chave “*timestamp*” representa o momento em que a tarefa foi



Figura 3.1 – Representação do ambiente virtual do jogo (NUNES, 2022).

executada, além disso a chave “*actor*” representa as informações do aluno que está executando a tarefa. Outra parte da estrutura é composta pelas informações a respeito de qual foi a interação que o aluno realizou. Como por exemplo, a chave “*verb*” que armazena a identificação da ação a qual o aluno executou sobre um objeto, como por exemplo: assistir, ler, visualizar, ouvir ou responder. A chave “*object*” representa qual foi o componente ou material da aula em que o aluno realizou a ação. Como por exemplo se o aluno respondeu uma questão do questionário. Por fim, a chave “*result*” representa qual a situação que se encontra a atividade, se está completa ou não e qual a pontuação o aluno obteve ao realiza-la.

A partir dos estudos propostos, somos capazes de explorar diversos aspectos que podem contribuir a essa linha de pesquisa. Uma forma de explorar, sendo a que está sendo executado neste trabalho, é utilizar dos dados coletados no formato *xAPI* para aplicar *Learning Analytics* nos dados de experiências dos alunos. Dessa forma podemos transformar dados em informações relevantes para os profissionais da educação com foco na melhoria contínua em métodos de aprendizagem.



Figura 3.2 – Representação dos componentes disponíveis para interação do aluno no jogo (NUNES, 2022).

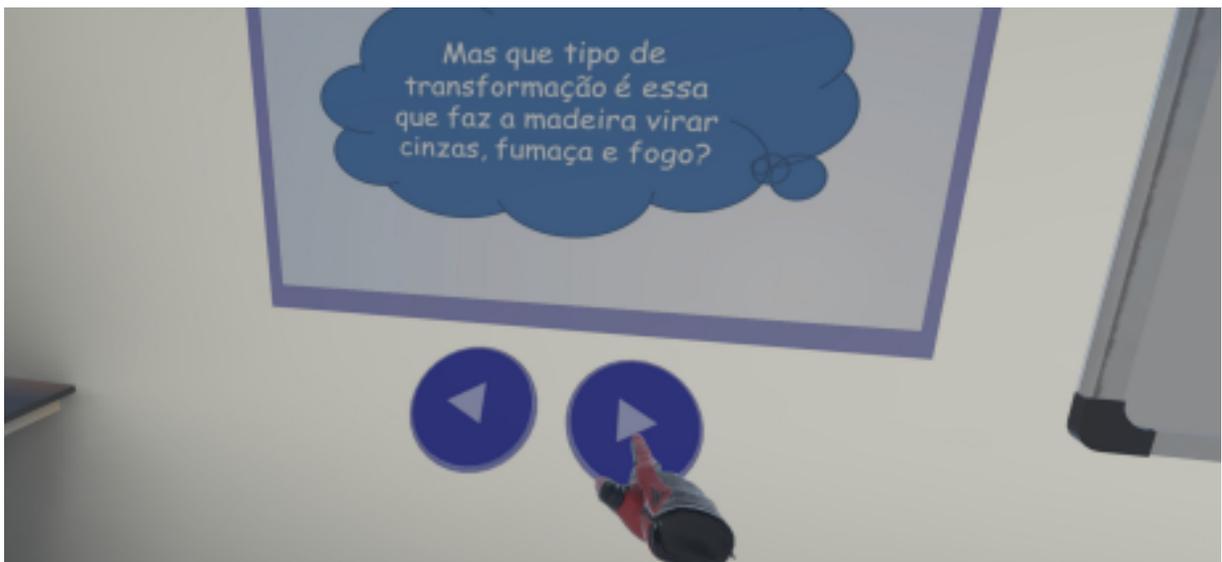


Figura 3.3 – Representação dos componentes disponíveis para interação do aluno no jogo (NUNES, 2022).

```
{
  "version": "1.0.3",

  "timestamp": "2022/05/08T00:02:37",
  "stored": "2022/05/08T00:02:37",

  "id": "dd99da4f-47ba-4a74-9456-3f4dc44fa9cb",
  "actor": {
    "objectType": "Agent",
    "mbox": "mail:student01@gmail.com",
    "name": "student01"
  },

  "verb": {
    "id": "http://url/verb/viewed",
    "display": { "en-US": "viewed" }
  },

  "object": {
    "id": "http://url/1.0/viewed",
    "objectType": "Activity"
  },

  "result": {
    "score": { "raw": 0 },
    "success": false,
    "completion": false
  }
},
```

Figura 3.4 – Formato *JSON* construído e armazenado pela *xAPI*.

4 Desenvolvimento

Esse capítulo será dividido em duas seções que apresentarão o processo de desenvolvimento deste trabalho, sendo elas: a Seção 4.1, construção da base de dados e a Seção 4.2, que consiste na definição da arquitetura da plataforma de *Learning Analytics* e no desenvolvimento das rotas e métodos que irão compor a *API*.

4.1 Base de Dados

Este trabalho utiliza como base o projeto desenvolvido por Nunes (2022) e Melo (2022), no qual foi desenvolvido um jogo sério que se utiliza da *xAPI* para coletar os dados de experiência dos alunos ao utilizarem a plataforma, como discutido em na seção 3. Porém, como o projeto de base se encontra em uma fase inicial de desenvolvimento, ainda não foi possível a realização de testes com o usuário. Dessa forma, foi necessário a implementação de um gerador de base dados sintéticos que representasse diferentes perfis de interações de alunos ao interagir plataforma.

O jogo utiliza-se da *xAPI* para registrar os dados de experiências de cada interação dos estudantes em uma estrutura de arquivo no formato *JSON*, discutida na Seção 3. A divisão dessa estrutura se dá por três chaves principais, sendo elas: ator, verbo e objeto, como representada na Figura 3.4.

A aula presente no jogo representa um conjunto de atividades que deve ser realizada pelos alunos. Dentre as atividades, os alunos terão a sua disposição tarefas preparatórias, como vídeos, slides, textos e áudios. Além disso, para conclusão das atividades propostas, os alunos deverão responder um questionário referente aos materiais de apoio fornecidos.

Dessa forma, utilizando o cenário de possibilidades, foi implementado na linguagem de programação *Python* um gerador de execução de atividades, simulando o formato de registros de dados realizado pela *xAPI*. Assim, para a construção da base de dados, foi levado em consideração uma estratégia que se baseou em diferentes perfis de aprendizagem e comportamento de alunos.

A estratégia trata-se da utilização de um sistema de nível de conhecimento dos alunos, que é incrementado à medida que os alunos realizam tarefas preparatórias. Dessa forma, o desempenho dos estudantes ao responderem o questionário se dará pelo nível do seu conhecimento, ou seja, alunos que não realizarem as atividades preparatórias terão menor probabilidade de acertar uma questão quando comparados aos que realizaram todas as tarefas.

O tamanho da amostra gerada possui um total de 3.636 execuções de atividades realizadas por um grupo de 100 alunos. Esse grupo é subdividido e classificado em:

- Alunos que realizam totalmente as atividades preparatórias antes de realizarem o questio-

nário, representando 40% da base de dados;

- Alunos que realizam parcialmente as atividades preparatórias antes de realizarem o questionário, representando 30% da base de dados;
- Alunos que realizam as atividades preparatórias intercalando com a realização do questionário, representando 20% da base de dados;
- Alunos que realizam apenas o questionário, negligenciando totalmente as atividades preparatórias, representando 10% da base de dados.

4.2 Learning Analytics API

O objetivo deste trabalho consiste no desenvolvimento de uma API para aplicação de *Learning Analytics* em dados extraídos de uma plataforma de jogo com foco educacional, por meio de dados estruturado pela *xAPI*. Nessa seção será apresentado o processo de desenvolvimento de uma API capaz de transformar dados em informações. Por meio da análise de dados e aplicação de algoritmo de agrupamento será possível identificar o desempenho e comportamento dos alunos ao interagirem com a plataforma.

Para isso, foi utilizado a linguagem de programação em *Python*, já que possui diversas bibliotecas que fornecem uma gama de ferramentas para aplicação de pré-processamento e análise de dados, utilizando as bibliotecas *Numpy*¹ e *Pandas*², como é discutido no trabalho de Stančín e Jović (2019). Além disso, foi utilizado o *Flask*³, que proporcionam um ambiente fácil e rápido para desenvolvimento de aplicações web por meio da linguagem *Python*⁴ (MUFID et al., 2019).

4.2.1 Arquitetura

A arquitetura proposta para o desenvolvimento da *LA API* é ilustrada na Figura 4.1. A estrutura da *API* é dividida em três camadas importantes, sendo ela: camada de controle, camada de pré-processamento e a camada de métodos de *Learning Analytics*. Dessa forma, será possível ao cliente realizar requisições, utilizando o método *HTTP POST*, à *API* através de um endereço eletrônico, como pode ser referenciado na interação 1 da Figura 4.1.

A camada de controle representa todos os métodos desenvolvidos. Para esse projeto existem cinco métodos que são rotas para executar algum estudo específico sobre os dados que serão enviados no corpo da requisição, ou seja, através do endereço eletrônico deve-se enviar junto ao corpo da requisição os dados de configuração, que fornecerão dados importantes para a

¹ Numpy <<https://numpy.org/>>.

² Pandas <<https://pandas.pydata.org/>>.

³ Flask <<https://flask.palletsprojects.com/en/2.2.x/>>.

⁴ Python <<https://www.python.org/>>.

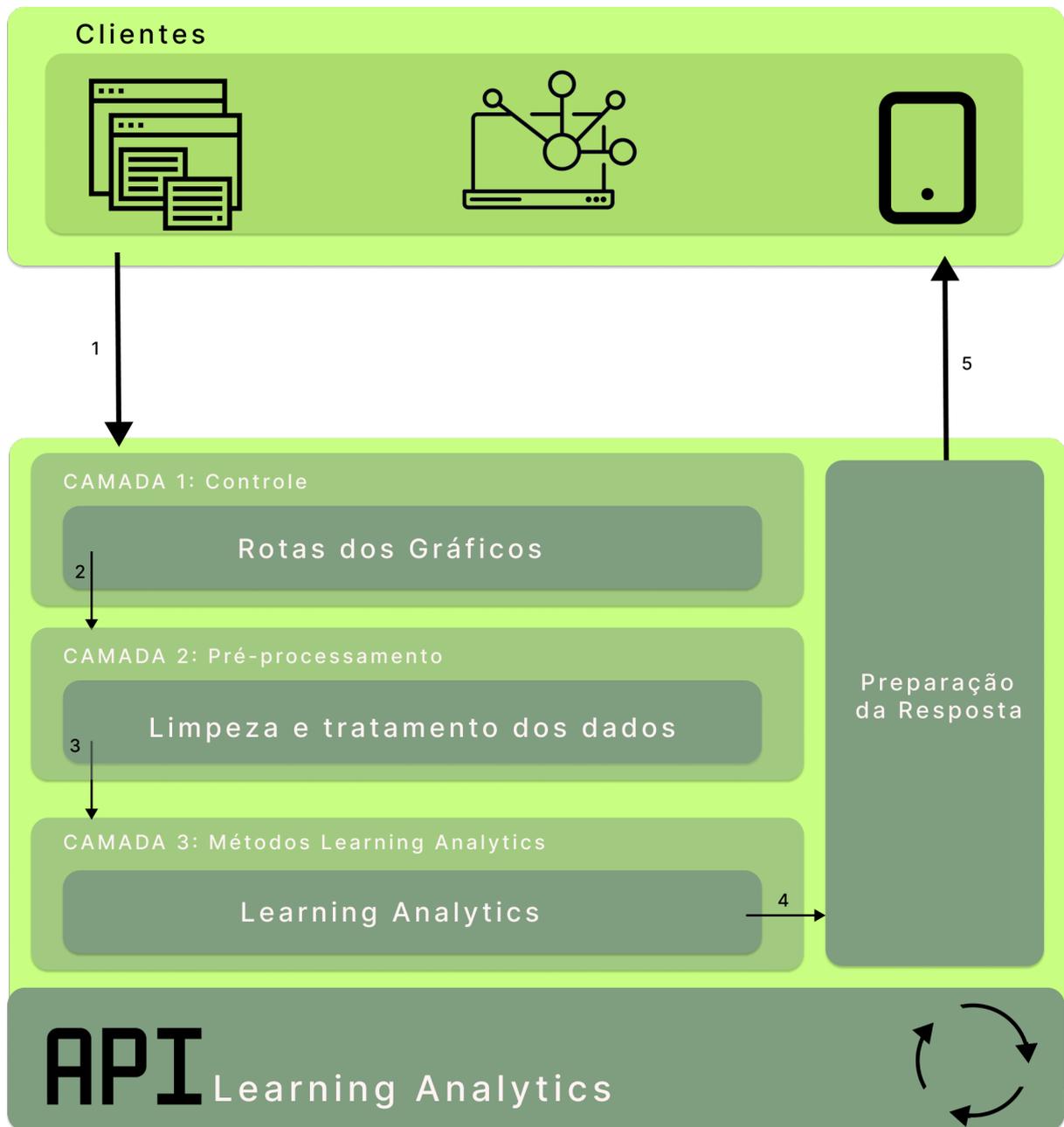


Figura 4.1 – Arquitetura *Learning Analytics API*.

```
{
  "id": "1",

  "config": {
    "studentID": "7faf0071-e6cd-4db2-9846-f0edbfdd742e",
    "startDate": "2022/05/01T00:00:00",
    "endDate": "2022/05/30T23:59:59",
  },

  "source": {
    "statements": [
      {
        "version": "1.0.3",
        "id": "e999c1a8-e602-42b8-9e5f-72cc35ab8e68",
        "actor": {
          "objectType": "Agent",
          "mbox": "mailto:student01@gmail.com",
          "name": "student01"
        },
        "timestamp": "2022/05/20T05:35:55",
        "stored": "2022/05/20T05:35:55",
        "verb": {
          "id": "http://url/verb/answered",
          "display": { "en-US": "answered" }
        },
        "object": {
          "id": "http://url/1.0/question",
          "definition": {
            "extensions": {
              "http://url/1.0/question": {
                "question": "question_01",
                "answer": "A"
              }
            }
          }
        }
      }
    ]
  }
}
```

Figura 4.2 – Formato de arquivo *JSON* a ser enviado junto a requisição para *LA API*

execução dos métodos, e o conjunto dos dados de experiências dos alunos extraídos do jogo pela *xAPI*, após a interação dos estudantes com a plataforma, como discutido no capítulo de 3.

Para essa *API*, foram desenvolvidas cinco rotas diferentes que realizarão extração de informações sobre o desempenho e comportamento dos alunos em diferentes casos de análises. Nas próximas seções serão explicados com mais detalhes cada uma das rotas com exemplos demonstrando quais serão os dados necessários e os resultados produzidos pela *API* após o processamento dos dados dos alunos.

Após a requisição a uma determinada rota, o caminho para a execução do método de *Learning Analytics* será estabelecido, mas antes, como representado na interação 2 da Figura 4.1, os dados de experiência dos alunos passarão para uma série de etapas que envolverão métodos de limpeza e tratamentos dos dados como uma forma de prepará-los para a aplicação dos estudos. Essa etapa representa a camada de Pré-processamento.

A interação 3 representada, retratada na Figura 4.1, demonstra o próximo passo sendo a execução da camada de Métodos *Learnig Analytics*. Nessa camada, cada método desenvolvido, são realizadas uma sequência de cálculos e execução de algoritmos com o objetivo de extrair informações sobre os dados de entrada. Como por exemplo identificar comportamento do aluno ao realizar as tarefas, classificar grupos de perfis semelhantes de usuários, ou até mesmo correspondências entre atividades e desempenho. A partir disso, ao obter os resultados do método executado, a interação 4 da figura 4.1 iniciará o processo para estruturar a resposta em um formato *JSON*, que será retornado ao cliente na interação 5.

4.2.2 Rotas da *LA API*

As rotas implementadas para realizar as requisições à *LA API* se dá através de cinco *endpoin*s. Sendo eles: *timeExecution*, *performance*, *behavior*, *correlation* e *clustering*. Por se tratar de uma *API REST*, foi utilizado o formato de método *POST* para as requisições, ou seja, é necessário enviar junto à requisição um arquivo de protocolo no formato *JSON* contendo os dados que serão analisados, no formato construído pela *xAPI*, além dos dados de configuração, tais dados representam informações úteis para execução dos métodos solicitados. As Figuras 4.3 e 4.2 representam uma descrição breve das rotas disponíveis e um exemplo do formato de arquivo que deve ser enviado junto a requisição, respectivamente.

Cada uma das rotas tem como finalidade retornar dados processados sobre um estudo que se pode aplicar sobre eles. A seguir serão descritos e exemplificados cada uma dos estudos.

4.2.2.1 Método de extração de tempo de execução

Com os dados de experiência dos alunos podemos aplicar cálculos com o objetivo de encontrar o tempo de execução em minutos que cada aluno apresentou ao realizar cada uma as atividades propostas no jogo. Dessa forma, com os resultados, uma das possibilidades de uso

Rota	Método	Função
/performance	POST	Capaz de gerar informação sobre o desempenho de um aluno em atividades
/correlation	POST	Capaz de gerar informação sobre a correlação entre duas atividades e o desempenho do aluno
/clustering	POST	Capaz de subdividir grupos de alunos com perfis semelhantes
/behavios	POST	Capaz de identificar o comportamento de fluxo do aluno ao realizar tarefas
/timeExecution	POST	Capaz de gerar informação do tempo que o aluno gasta em cada tarefa

Figura 4.3 – Tabela rotas existentes na *LA API*

```

"config": {
  "studentID": "2d609279-9ee3-4afd-aab2-369838689361",
  "startDate": "2022/05/01T00:00:00",
  "finishDate": "2022/05/30T23:59:59",
},
"source": {
  "statements": [...]
}
}

```

Figura 4.4 – Dados de configuração para a rota *timeExecution*.

```

"data": {
  "data": {
    "studentTime": [ 81.0, 117.0, 36.0, 145.0, 28.0],
    "classTime": [ 99.24, 27.98, 13.51, 34.61, 9.03],
    "labelTaks": [
      "answer", "read", "view", "watch", "listen"
    ]
  },
  "titulo": "Time Per Task Chart",
  "type": "TimePerTaskChart"
  "id": "2d609279-9ee3-4afd-aab2-369838689361",
}

```

Figura 4.5 – Resultado obtido através da requisição à rota *timeExecution*.

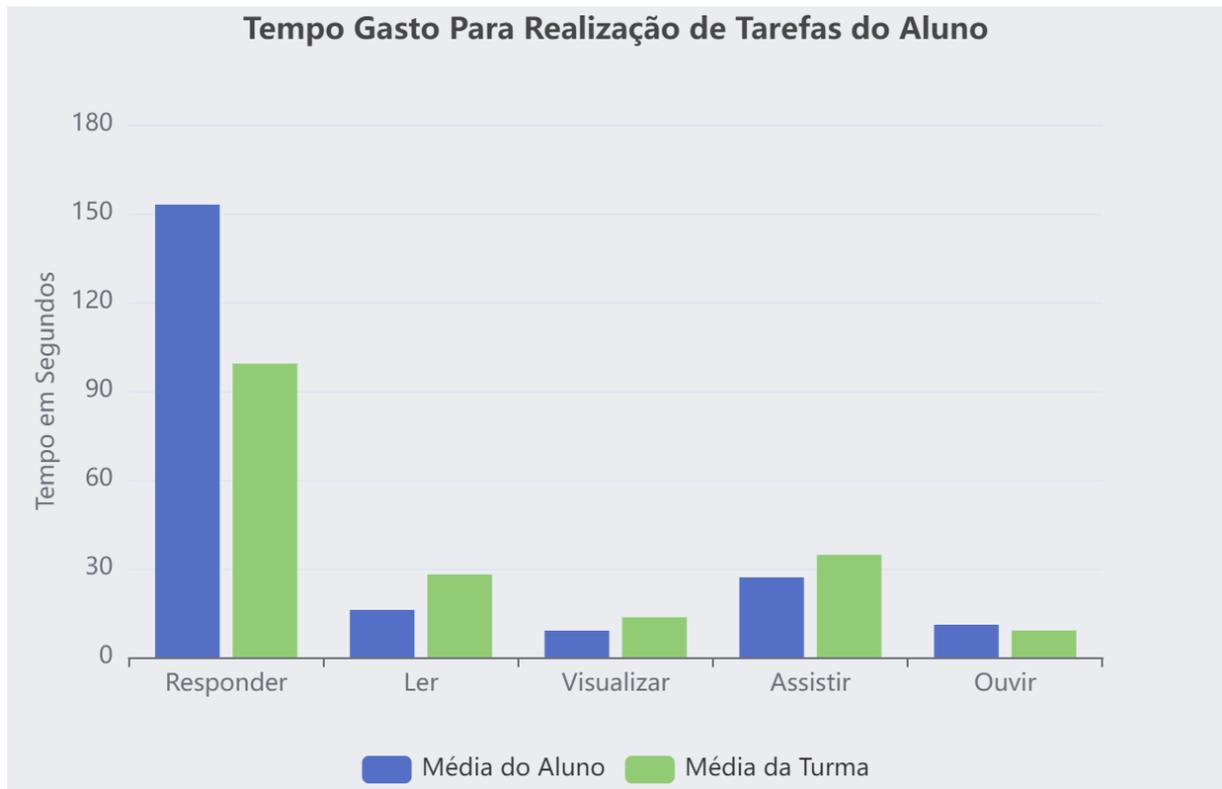


Figura 4.6 – Gráfico de tempo gasto por tarefas realizadas por um aluno.

do retorno dos dados processados, será utilizá-los para a construção de um gráfico que dará ao profissional da educação uma visão clara sobre quais tarefas o aluno teve menor ou maior interação na plataforma.

Assim, pode ser possível a identificação de diferentes situações e combiná-las para entender o processo de aprendizagem dos alunos, ou seja, identificar se o aluno está negligenciando a tarefa ou até mesmo se a atividade está sendo um gargalo nesse processo, tomando tempo demais do aluno.

Para isso, ao realizar a requisição POST à rota *timeExecution* deve ser enviado juntamente os dados de configuração do método, que pode ser representado na Figura 4.4. Os dados de configuração no formato *JSON* são divididos em chave-valor, sendo elas: a chave *studentID* que deverá conter o número de identificação do aluno a qual se busca realizar a extração de informação; e as chaves *startDate* e *finishDate* representando o intervalo de tempo que se deseja extrair a informação.

O método *timeExecution* é dividido em duas partes. Na primeira serão realizadas uma filtragem dos dados do aluno solicitado para calcular o tempo de execução de cada tarefa, utilizando a data e hora que se iniciou cada uma das tarefas. Já na segunda parte do método, será realizado o cálculo médio de tempo gasto em cada atividade. Assim, os resultados obtidos poderão ser representados por um *JSON*, mostrado na Figura 4.5. A estrutura retornada apresenta a chave *data*, sendo dividido em três chaves-valores. A primeira chave é denominada pela chave

```
"config": {
  "studentID": "2d609279-9ee3-4afd-aab2-369838689361",
  "startDate": "2022/05/01T00:00:00",
  "finishDate": "2022/05/30T23:59:59",
},
"source": {
  "statements": [...]
}
}
```

Figura 4.7 – Dados de configuração para a rota *performance*.

```
"data": {
  "data": {
    "studentPerformance": 0.5263157894736842,
    "classPerformance": 0.7190464805456729,
  },
  "titulo": "Performance Chart",
  "type": "PerformanceChart",
  "id": "2d609279-9ee3-4afd-aab2-369838689361",
}
```

Figura 4.8 – Resultado obtido através da requisição à rota *performance*.

studentTime, que representará uma lista do somatório de tempo gasto em cada uma das atividades, em minutos. A segunda chave, *classTime*, representa uma lista de valores médios, em minutos, do tempo gasto pelos alunos da turma ao realizarem cada uma das atividades. E, por fim, a chave *labelTasks* que representa uma lista de ações das atividades disponíveis no jogo.

Como uma das formas de utilizar os dados resultantes do método, realizado pela API, pode se construir gráficos com o objetivo de transformar os dados brutos em dados mais fáceis de serem visualizados pelos professores. A Figura 4.6 ilustra esta forma.

4.2.2.2 Métodos de extração do Desempenho do Questionário

Outra possibilidade de analisar os dados de experiência é calcular qual o desempenho que os alunos obtiveram ao realizarem o questionário avaliativo. Dessa forma, através da requisição POST à rota *performance*, junto aos dados de configuração, que pode ser exemplificado na Figura 4.7. Assim como no método anterior, os dados possuem a mesma estrutura de chave valor, com a diferença do nome do método, que para esse caso é *performanceChart*. Os dados de configuração no formato *JSON* são compostos das chaves *studentID*, que deverá conter o número de identificação do aluno a qual se busca realizar a extração de informação, e as chaves *startDate* e *finishDate* representando o intervalo de tempo que se deseja extrair a informação.

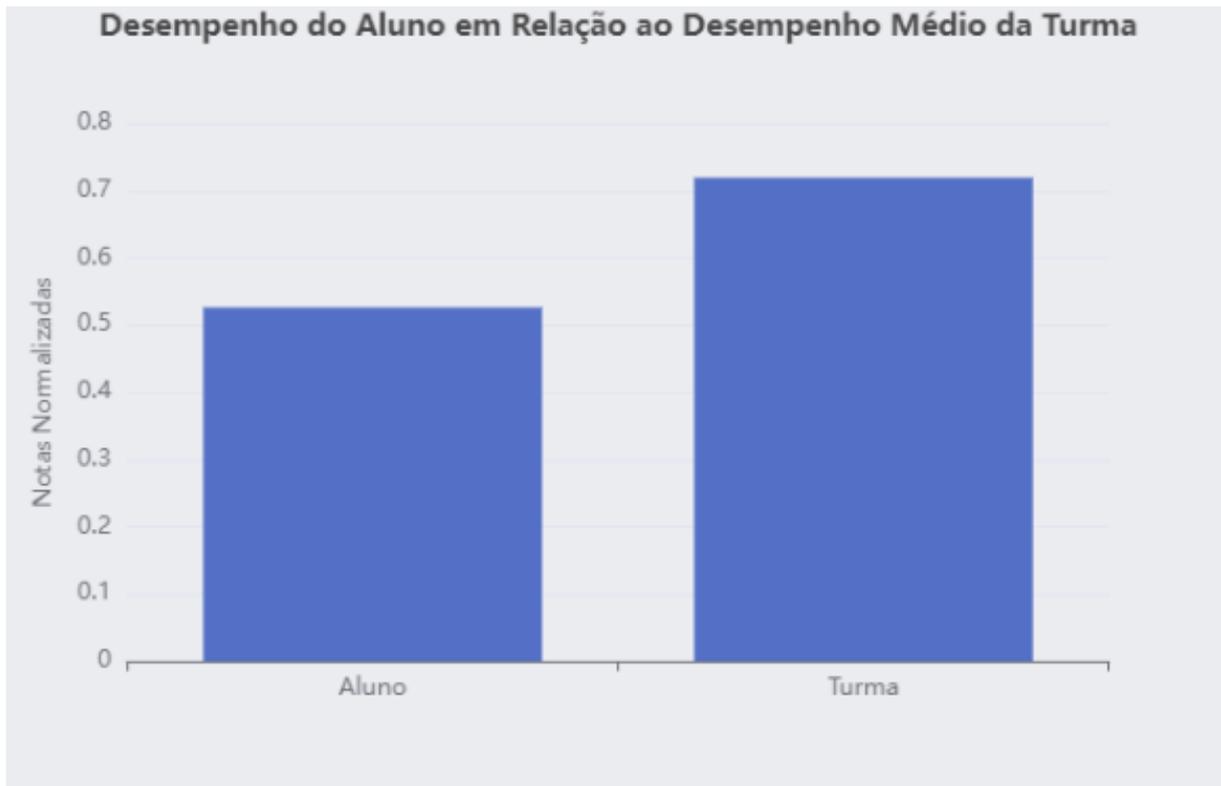


Figura 4.9 – Gráfico do desempenho do aluno no questionário em relação ao desempenho médio da turma.

O método *performance* primeiramente realiza a filtragem dos dados referente ao aluno definido pelo número de identificação. A atividade de questionário permite que os alunos refaçam as questões quando houver respostas erradas, dessa forma, para que o aluno conclua a atividade do questionário, por exemplo, ele deverá responder todas as questões de forma correta. Então, o desempenho do aluno é dado pelo número de vezes que questões foram registradas como erradas, ou seja, quanto maior os números de erros cometidos pior será o desempenho dos alunos. Além disso, o método também realiza o cálculo do desempenho médio dos alunos da turma, para que se possa fornecer mais informação de comparar o desempenho do aluno em relação a sua turma. Por fim, ao finalizar a execução do método, o retorno dos dados no formato *JSON* é representado pela Figura 4.8. O resultado contém em estrutura de chave-valor a chave *data*, sendo ela dividida em duas chaves, *studentPerformance*, que representa o valor normalizado do desempenho do aluno, e *classPerformance*, que representa o valor normalizado da média do desempenho dos alunos da turma.

A Figura 4.9 demonstra uma forma de utilizar os dados retornados pelo método *performance*.

4.2.2.3 Método de extração do comportamento de execução de tarefas

Como uma forma de identificar a ordem da execução das atividades que foram realizadas por um aluno, o método *behavior* é capaz de extrair essa informação além da calcular quantidade

```
"config": {
  "studentID": "2d609279-9ee3-4afd-aab2-369838689361",
  "startDate": "2022/05/01T00:00:00",
  "finishDate": "2022/05/30T23:59:59",
},
"source": {
  "statements": [...]
}
}
```

Figura 4.10 – Dados de configuração para a rota *behavior*.

```
"data":
{
  "data": {
    "links": [
      { "source": "start", "target": "readed" },
      { "source": "readed", "target": "watched" },
      ...
      { "source": "answered", "target": "answered" },
      { "source": "answered", "target": "answered" }
    ],
    "values": {
      "answered": 29, "listened": 1, "readed": 1,
      "viewed": 1, "watched": 1
    }
  },
  "titulo": "FlowTaskGraphChart",
  "type": "FlowTaskGraphChart",
  "id": "2d609279-9ee3-4afd-aab2-369838689361",
}
}
```

Figura 4.11 – Resultado obtido através da requisição à rota *behavior*.

de vezes que um aluno realizou cada tarefas. Com essa informação, pode-se transformar em um grafo que será capaz de representar o comportamento que o aluno tem ao realizar as tarefas, como por exemplo se o aluno prefere executar primeiro todas as atividades preparatórias antes de responder o questionário, ou se prefere intercalar as atividades com o questionário.

Para isso, através da rota *behavior*, deve se passar como dados de configuração as informações que representarão o número de identificação do aluno, pela chave *studentID*, e o intervalo de tempo que se deseja extraír a informação, pelas chaves *startDate* e *finishDate*. A Figura 4.10 ilustra o formato da configuração para o método.

Como resultado, as informações que podem ser retornadas após a execução do método *behavior* são representadas na Figura 4.11. O formato de dados JSON resultante contém a chave *data* que são divididas em duas chaves. A primeira chave, *links*, apresenta como valor uma lista



Figura 4.12 – Gráfico do fluxo do comportamento do aluno ao realizar as tarefas.

de objetos que representarão o fluxo de execução das tarefas do aluno, como por exemplo os valores que definem a atividade de origem, pela chave */emphsource*, e a atividade de destino, pela chave *target*. A segunda chave, *values*, é um objeto dividido em chaves, sendo cada chave o nome da atividade executada e seus respectivos representando a quantidade de vezes que houve interação por parte do aluno.

Os dados retornados poderão ser utilizados para construção de grafos, no qual os vértices são os verbos que foram executados e as arestas representarão o fluxo realizado pelo aluno. Uma forma de representar o resultado em forma de um gráfico, pode ser visualizado na Figura 4.12.

4.2.2.4 Método de extração da correlação de atividade e desempenho

Uma informação importante que podemos extrair utilizando o método *correlation* é encontrar uma correlação entre uma atividade preparatória e desempenho dos alunos ao realizarem o questionário. Sendo capaz de fornecer ao educador a informação de quais atividades podem estar gerando impactos positivos ou negativos nos resultados do desempenho dos estudantes.

Dessa maneira, através da requisição à rota *correlation*, deve enviar os dados de configuração para execução do método descrito. Os dados de configuração no formato *JSON* deverá conter as chaves *startDate* e *finishDate* representando o intervalo de tempo que se deseja extrair a informação e qual atividade deseja correlacionar ao desempenho médio da turma, pela chave *task*. Figura 4.13.

```

"config": {
  "task": "watched",
  "startDate": "2022/05/01T00:00:00",
  "finishDate": "2022/05/30T23:59:59",
},
"source": {
  "statements": [...]
}
}

```

Figura 4.13 – Dados de configuração para a rota *correlation*.

```

"data": {
  "xValues": [0.0, 0.0, ..., 0.190, 0.183],
  "yValues": [0.169, 0.348, ..., 0.169, 0.108],
  "yValuesPredict": [0.311, 0.312, ..., 0.2.489, 0.478],
  "taskLabel": "watched",
  "titulo": "Task Performance Correlation Graph",
  "type": "taskPerformanceCorrelationGraph"
}

```

Figura 4.14 – Resultado obtido através da requisição à rota *correlation*.

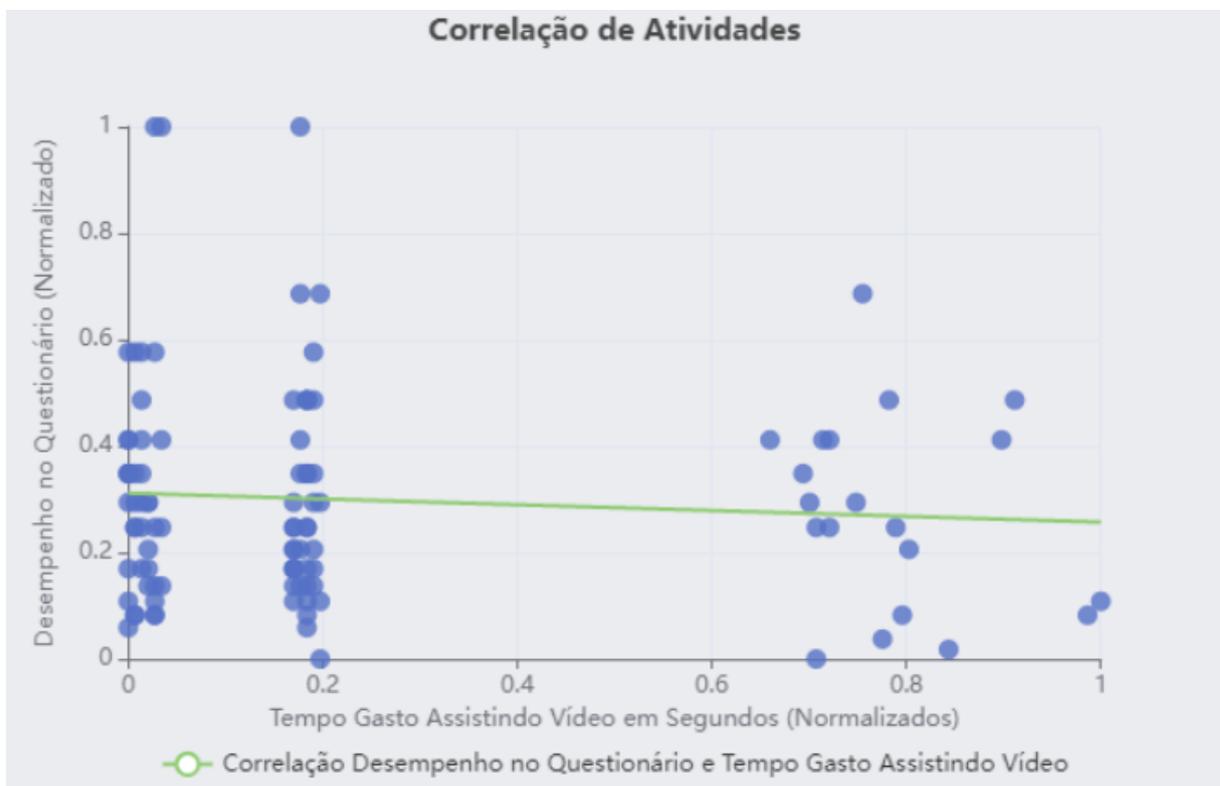


Figura 4.15 – Gráfico da correlação entre o desempenho dos alunos e o tempo gasto realizando a tarefa de assistir vídeos.

```
"config": {
  "nGroup": 4,
  "startDate": "2022/05/01T00:00:00",
  "finishDate": "2022/05/30T23:59:59",
},
"source": {
  "statements": [...]
}
}
```

Figura 4.16 – Dados de configuração para a rota *clustering*.

```
"data": {
  "data": [
    {
      "xValues": [0.078, ..., 0.023],
      "yValues": [0.588, ..., 0.588],
      "id": 0,
    },
    ...
    {
      "xValues": [1.0, ..., 0.850],
      "yValues": [0.764, ..., 0.882],
      "id": 3,
    }
  ],
  "titulo": "Student Profiles Clustering",
  "type": "studentProfilesClustering"
}
```

Figura 4.17 – Resultado obtido através da requisição à rota *clustering*.

Assim, como resultado do método será possível identificar se, por exemplo, o grau de relevância ao realizar a atividade de assistir vídeo impactou em relação ao resultado final do usuário ao realizar o questionário. Na Figura 4.14 é possível observar que o resultado é dado pela chave *data* que é dividido em duas chaves, sendo elas a chave *xValues*, que representa uma lista de valores de tempo em que cada aluno gastou ao realizar a tarefa, e a chave *yValues*, que representa o desempenho de cada aluno ao realizar o questionário. Além disso, chave *yValuesPredict* representa uma lista de valores que representará a reta de correlação entre as duas atividades em uma regressão linear. A inclinação dessa reta irá fornecer ao analisador se a atividade que ele propôs aos alunos impactou na preparação dos alunos para a execução do questionário.

Como resultado para esse métodos, podemos utilizar os dados para gerar um gráfico como mostrado na Figura 4.15

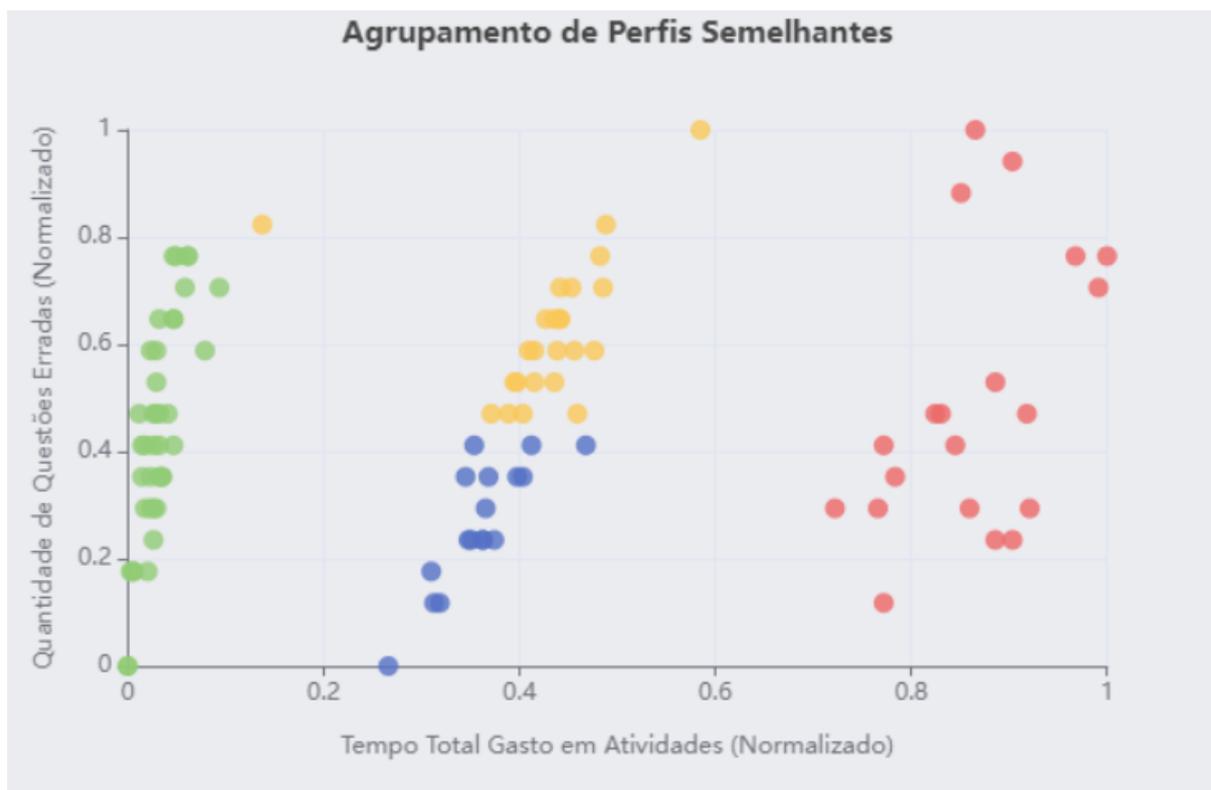


Figura 4.18 – Gráfico de agrupamento por perfis semelhantes.

4.2.2.5 Método de agrupamento por semelhança no perfil de aluno

O método de *clustering* é capaz analisar os dados dos alunos por meio de suas características no comportamento em seus processos de aprendizagem, como por exemplo levar em consideração o tempo que um aluno passa estudando e o resultado que adquire no questionário, sendo possível evidenciar grupos de alunos que tiveram sucesso ao utilizarem os materiais de apoio durante o processo de aprendizado ou grupos de alunos que apresentaram dificuldades mesmo utilizando os materiais. Para isso, ao utilizar a rota *clustering*, os dados de configuração deverão ser enviados com as seguintes informações, sendo elas: as chaves *startDate* e *finishDate* representando o intervalo de tempo que se deseja extrair a informação; o número de grupos que se deseja representar a turma, através da chave *nGroup*. A Figura 4.16 exemplifica os dados de configuração para este método.

Como uma forma de classificar e identificar grupos de alunos com perfis de comportamento semelhante, foi realizado primeiramente um tratamento no arquivo de dados enviados, contido nos dados de configuração por meio da chave *source*. Esses dados são filtrados e processados para extrair as informações de quais atividades foram realizadas por cada aluno, qual o tempo gasto na realização da atividade preparatória e avaliativas e qual o desempenho os alunos obtiveram ao realizar o questionário.

Para realizar o agrupamento foi utilizado o algoritmo de clusterização *Fuzzy c-mean*, da

biblioteca `scikit-fuzzy` *scikit-fuzzy*⁵. A base de dados utilizada representa quatro tipos de estudantes, dentre eles, existem aqueles que realizam todas as tarefas e respondem o questionário, aqueles que ignoram as atividades preparatórias e realizaram o questionário, aqueles que realizaram o questionário intercalando com as atividades preparatórias, e aqueles que realizaram parcialmente as atividades preparatórias e depois responderam o questionário. Então como parâmetro de quantidade de clusters definido previamente foi a geração de 4 grupos.

Na Figura 4.17 é possível observar que o resultado é dado pela chave *data* que representa uma lista de objetos de cada grupo identificado. Cada grupo irá conter as seguintes informações, sendo elas: o *id*, para a identificação do grupo; as listas de valores do desempenho do alunos e tempo gasto com atividades, ambos normalizados, nas respectivas chaves *xValues* e *yValues*.

O resultado para esse método pode ser ilustrado na Figura 4.18. O Algoritmo *FCM*, foi capaz de subdividir a turma de alunos em grupos de quatro perfis diferentes. Dessa forma, pode-se observar que quando o tempo de estudo aumenta, ocorre maior dispersão dos dados em relação a quantidade de questões erradas. Isso permite identificar com maior clareza os alunos que necessitam de maior atenção, como por exemplo o grupo vermelho, no qual alguns alunos utilizaram mais tempo realizando atividades preparatórias e mesmo assim obtiveram um desempenho ruim. Com isso o professor poderá compreender melhor as dificuldades desses alunos e assim aplicar metodologias alternativas.

⁵ `scikit-fuzzy` <<https://pypi.org/project/scikit-fuzzy/>>.

5 Resultados

Como uma forma de demonstrar os resultados e possibilidades e utilização da *API*, foi implementado uma documentação em formato web para que além de apresentar a descrição de cada método desenvolvido, também serviu como uma demonstração de um sistema que está atuando como cliente da *Learning Analytics API*. O sistema foi implementado utilizando as linguagens *HTML* e *JavaScript*, através de uma biblioteca do *Python* chamada de *MkDocs*¹, que possibilita criar de forma fácil e customizável documentações de projetos.

A documentação é dividida entre a página de introdução e as páginas que descrevem cada um dos métodos desenvolvidos. Cada uma das páginas de descrição dos métodos apresenta, em sua estrutura, um exemplo de quais serão os dados de entrada necessários para serem enviados no corpo da requisição à rota. Além disso, apresenta um exemplo do formato dados de saída, que são retornados após a execução da rota da *API*.

Como uma forma de demonstrar o uso da *API*, cada método possui em sua página um gráfico que foi gerado a partir dos resultados retornados após realizar uma requisição à *Learning Analytics API*. Para os gráficos, foi utilizada a biblioteca de visualização de dados do *JavaScript*, conhecida como *Echarts*².

Como forma de ilustrar a documentação desenvolvida, as Figuras 5.1, 5.2 e 5.3 demonstra respectivamente um exemplo das páginas de Introdução e as páginas dos métodos de comporta-

¹ MkDocs <<https://www.mkdocs.org/>>.

² Apache Echarts <<https://echarts.apache.org/en/index.html>>.

Learning Analytics API

A Learning Analytics API, LA API, consiste em uma aplicação que realizará estudos matemáticos e estatísticos sobre dados coletados de plataformas digitais educacionais por meio da xAPI.

Por meio das rotas desenvolvidas, é possível solicitar através de uma requisição POST, passando os dados do formato xAPI, diferentes formatos de estudos com o objetivo de extrair informações úteis sobre desempenho, comportamento, análises de perfis, dentre outros.

Rotas

Há diferentes rotas que podem ser usadas para extrair informações dos dados. Sendo elas:

Simple table with header

Rota	Função
/performance	Capaz de gerar informação sobre o desempenho de um aluno em atividades
/correlation	Capaz de gerar informação sobre a conexão entre duas atividades
/clustering	Capaz de gerar informação subdividindo perfis de alunos
/behaviors	Capaz de gerar identificar o comportamento do aluno ao realizar tarefas
/timeExecution	Capaz de gerar informação o tempo que o aluno gasta em cada tarefa

Next

Built with MkDocs using a theme provided by Read the Docs.

Figura 5.1 – Página inicial da documentação da *LA API*

Learning Analytics para xAPI

Search docs

- Início
- Gráficos de Desempenho
- Gráficos de Correlação
- Gráficos de Agrupamento
- Gráficos de Comportamento**
 - Gráficos de Tempo de Execução

« Previous

Gráficos de Comportamento

Comportamento

Como uma forma de identificar a ordem de atividades que foram realizadas por um aluno, a rota Behavior é capaz de extrair essa informação além da calcular quantidade de vezes que um aluno realizou cada tarefa.

Essa informação pode ser capaz de representar o comportamento que o aluno tem ao realizar as tarefas, como por exemplo se o aluno prefere executar primeiro todas as atividades preparatórias antes de responder o questionário ou se prefere intercalar as atividades com o questionário.

Rota Behavior

A rota é nomeada pela url/behaviors por meio de uma requisição POST. Ao realizar a solicitação à rota enviando os dados de configuração em formato JSON, o sistema irá realizar a preparação inicial dos, ou seja, o processo de limpeza e pre-processamento e a partir disso é realizado a etapa de calcular o tempo em que um aluno especificado por um ID gastou ao executar cada uma das tarefas do jogo.

```

@app.route('/behavior', methods=['POST'], strict_slashes=False)
def analysis_student():
    data = Prepare_data.Prepare_data()
    data.build_data(request.json)
    behavior = Behavior_Analysis.Behavior_analysis(data)
    result = behavior.behavior()
    return result
    
```

Formato dos Dados de Requisição

Os dados de configuração no formato JSON são divididos chave-valor, sendo elas:

- studentID:** Contendo o número de identificação do aluno
- startDate:** Período de início para se analisar
- finishDate:** Período de fim para se analisar

```

"config": {
  "studentID": "25089279-9ee3-4af9-ab02-36983689361",
  "startDate": "2022/05/01 00:00",
  "finishDate": "2022/05/31 23:59:59",
  "source": {
    "statements": [...]
  }
}
    
```

Formato dos Dados de Resposta

Os dados de resultantes são retornados no formato JSON são divididos chave-valor, sendo elas:

- links:** Lista de objetos, no qual cada item corresponde em objeto o fluxo de trocas de atividades realizadas pelo aluno, origem e destino. A primeira posição da lista representa a primeira atividade realizada pelo aluno e a última posição representa a última atividade realizada pelo aluno.
- values:** Objeto que representa quantas vezes houve uma interação em cada atividade pelo aluno

```

"data": {
  "data": {
    "links": [
      { "source": "start", "target": "readed" },
      { "source": "readed", "target": "watched" },
      { "source": "watched", "target": "answered" },
      { "source": "answered", "target": "answered" }
    ],
    "values": {
      "answered": 20, "listened": 1, "readed": 1,
      "viewed": 1, "watched": 1
    }
  },
  "titulo": "FlowDiagramChart",
  "type": "FlowDiagramChart",
  "id": "25089279-9ee3-4af9-ab02-36983689361"
}
    
```

Exemplo

Como uma forma de utilização dos dados de retornos, podemos construir gráficos como:

```

graph TD
    start((start)) --> Ler((Ler))
    start --> Visualizar((Visualizar))
    start --> Assistir((Assistir))
    Ler --> Assistir
    Visualizar --> Assistir
    Assistir --> Ouvir((Ouvir))
    Assistir --> Responder((Responder))
    Ouvir --> Responder
    Responder --> Leitura((Leitura))
    
```

Built with MkDocs using a theme provided by Read the Docs.

Figura 5.2 – Página do método de comportamento da documentação da LA API

mento e agrupamento.

Learning Analytics para xAPI

Search docs

- Início
- Gráficos de Desempenho
- Gráficos de Correlação
- Gráficos de Agrupamento
- Gráficos de Comportamento
- Gráficos de Tempo de Execução

« Previous Next »

» Gráficos de Agrupamento

Agrupamento

O método de agrupamento de alunos desenvolvido na LA API visa em analisar os dados dos alunos por meio de suas características no comportamento em seus processos de aprendizagem, sendo possível dividir os alunos em grupos.

Com isso é possível comprar classificar diferentes tipos de comportamento dos alunos ao realizarem as tarefas preparatórias e como esse comportamento pode afetar no desempenho dos alunos ao realizarem o questionário avaliativo, ou seja, identificar um grupo de alunos no qual as atividade não estão sendo eficazes para atingirem bons desempenhos no questionário.

Rota Clustering

A rota é nomeada pela url/clustering por meio de uma requisição POST. Ao realizar a solicitação à rota enviando os dados de configuração em formato JSON, o sistema irá realizar a preparação inicial dos, ou seja, o processo de limpeza e pre-processamento e a partir disso é realizado a etapa de calcular o tempo em que um aluno especificado por um ID gastou ao executar cada uma das tarefas do jogo.

```

@app.route('/clustering', methods=['POST'], strict_slashes=False)
def clustering():
    data = Prepare_data.Prepare_data()
    data.build_dataset(request.json)
    clustering = Clustering_analysis.Clustering_analysis(data)
    result = Clustering.clustering()
    return result
    
```

Formato dos Dados de Requisição

Os dados de configuração no formato JSON são divididos chave-valor, sendo elas:

- nGroup:** Quantidade de grupos no qual se deseja subdividir a turma
- startDate:** Período de início para se analisar
- finishDate:** Período de fim para se analisar

```

"config": {
  "nGroup": 4,
  "startDate": "2022/05/01T00:00:00",
  "finishDate": "2022/05/31T23:59:59",
},
"source": {
  "statements": [...]
}
    
```

Formato dos Dados de Resposta

Os dados de resultantes são retornados no formato JSON são divididos chave-valor, sendo elas:

- xValues:** Lista de valores normalizados sobre o tempo gasto em atividades preparatórias
- yValues:** Lista de valores normalizados sobre o numero de questões erradas no questionário
- id:** Identificador do grupo

```

"data": {
  "data": [
    {
      "xvalues": [0.078, ..., 0.023],
      "yvalues": [0.588, ..., 0.588],
      "id": 0,
    },
    ...
    {
      "xvalues": [1.0, ..., 0.050],
      "yvalues": [0.799, ..., 0.002],
      "id": 3,
    }
  ],
  "titulo": "Student Profiles Clustering",
  "type": "studentProfilesClustering"
}
    
```

Exemplo

Como uma forma de utilização dos dados de retornos, podemos construir gráficos como:

« Previous Next »

Built with MkDocs using a theme provided by Read the Docs.

Figura 5.3 – Página do método de agrupamento da documentação da LA API

6 Considerações Finais

6.1 Conclusão

Neste trabalho foi realizado o desenvolvimento de uma *API* de *Learning Analytics* capaz de realizar extração de informações de dados estruturados por uma *API* de experiência com o objetivo de contribuir com uma nova área de pesquisa ao utilizar como base o trabalho de Nunes (2022) e Melo (2022). Dessa forma, foi construído uma base de dados sintética, foi desenvolvido a *Learning Analytics API* e foi desenvolvido a documentação da *API* em formato web.

Em um primeiro momento, como o trabalho desenvolvido por Nunes (2022) e Melo (2022) ainda não possui uma base de dados reais de experiência de alunos, ao interagirem com o jogo, foi necessário desenvolver um gerador de dados, utilizando a linguagem *Python*. O gerador desenvolvido foi capaz de criar um conjunto de dados estruturados no formato dos resultados entregue pela *xAPI*, de diferentes grupos de perfis de alunos em relação ao comportamento e resultados. Tal objetivo, possibilitou realizar testes para o desenvolvimento da *Learning Analytics API*.

A partir disso, foi desenvolvida uma *API* para aplicação de análises de dados de experiência. A plataforma apresenta um ambiente no qual cada rota desenvolvida, representa um estudo específico que pode ser aplicado sobre o conjunto de dados, com o intuito de extrair informações sobre os dados brutos. Esses estudos que podem ser exemplificados como cálculos de desempenho de alunos, identificação de comportamento ao executarem as tarefas, identificação de perfis subdividindo a turma em grupos, identificar a correlação de impacto de quais tarefas podem influenciar no desempenho dos alunos ao realizarem a atividade avaliativa e, por fim, calcular o tempo que cada aluno gasta ao realizar cada tarefa e comparar com a média dos tempos de outros alunos.

Como uma forma de documentação da *Learning Analytics API* foi desenvolvida uma página *Web*, que além de trazer descrições sobre cada rota desenvolvida, proporcionou uma prova de conceito. Ou seja, cada gráfico que foi renderizado na página de cada rota da documentação está consumindo a *API*, mostrando possibilidades de uso das informações retornadas por cada requisição.

6.2 Trabalhos Futuros

Para trabalhos futuros desse projeto, pode se criar diversos caminhos para expandir ainda mais a área de pesquisa de *Learning Analytics* e evoluções no poder da *API* desenvolvida. Um exemplo seria realizar um estudo para compreender quais tipos de análises serão mais eficientes

quando se trata de extração de informações sobre comportamentos e tendências nos processos de criação de metodologias de ensino. Além disso, cabe explorar ainda mais utilizando-se de algoritmos de *Machine Learning* e inteligência artificial com o objetivo de criar novos métodos de extração de informações mais robustos, bem como a exploração de novas camadas na arquitetura, como por exemplo sistemas de autenticação, na *Learning Analytics API* . Outro trabalho poderá ser a realização de testes com base de dados reais.

Referências

- ARDIMENTO, P.; BERNARDI, M. L.; CIMITILE, M.; RUVO, G. D. Learning analytics to improve coding abilities: a fuzzy-based process mining approach. In: IEEE. *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. [S.l.], 2019. p. 1–7.
- BERNARD, R. M.; BOROKHOVSKI, E.; SCHMID, R. F.; WADDINGTON, D. I.; PICKUP, D. I. Twenty-first century adaptive teaching and individualized learning operationalized as specific blends of student-centered instructional events: A systematic review and meta-analysis. *Campbell Systematic Reviews*, ERIC, v. 15, 2019.
- BEZDEK, J. C.; EHRLICH, R.; FULL, W. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, Elsevier, v. 10, n. 2-3, p. 191–203, 1984.
- BOURHIS, P.; REUTTER, J. L.; VRGOČ, D. Json: Data model and query languages. *Information Systems*, Elsevier, v. 89, p. 101478, 2020.
- CAO, L. Data science: a comprehensive overview. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 50, n. 3, p. 1–42, 2017.
- CHATTI, M. A.; DYCKHOFF, A. L.; SCHROEDER, U.; THÜS, H. A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, Inderscience Publishers, v. 4, n. 5-6, p. 318–331, 2012.
- GAERTLER, M. Clustering. *Network analysis: Methodological foundations*, Springer, p. 178–215, 2005.
- GAŠEVIĆ, D.; DAWSON, S.; SIEMENS, G. Let's not forget: Learning analytics are about learning. *TechTrends*, Springer, v. 59, n. 1, p. 64–71, 2015.
- IFENTHALER, D. Learning management system. In: _____. *Encyclopedia of the Sciences of Learning*. Boston, MA: Springer US, 2012. p. 1925–1927. ISBN 978-1-4419-1428-6. Disponível em: <https://doi.org/10.1007/978-1-4419-1428-6_187>.
- ISLAM, S.; MAHMUD, H. Integration of learning analytics into learner management system using machine learning. In: *Proceedings of the 2020 2nd International Conference on Modern Educational Technology*. [S.l.: s.n.], 2020. p. 1–4.
- JERNSTEDT, G. C. The relative effectiveness of individualized and traditional instruction methods. *The Journal of Educational Research*, JSTOR, p. 211–218, 1976.
- KAZANIDIS, I.; PELLAS, N.; CHRISTOPOULOS, A. A learning analytics conceptual framework for augmented reality-supported educational case studies. *Multimodal Technologies and Interaction*, Multidisciplinary Digital Publishing Institute, v. 5, n. 3, p. 9, 2021.
- KEVAN, J. M.; RYAN, P. R. Experience api: Flexible, decentralized and activity-centric data collection. *Technology, knowledge and learning*, Springer, v. 21, n. 1, p. 143–149, 2016.
- LEE, L.-K.; CHEUNG, S. K.; KWOK, L.-F. Learning analytics: current trends and innovative practices. *Journal of Computers in Education*, Springer, v. 7, p. 1–6, 2020.

- MELO, K. G. F. Uma arquitetura de integração de learning management system e learning record system para jogos sérios. 2022.
- MUFID, M. R.; BASOFI, A.; RASYID, M. U. H. A.; ROCHIMANSYAH, I. F. et al. Design an mvc model using python for flask framework development. In: IEEE. *2019 International Electronics Symposium (IES)*. [S.l.], 2019. p. 214–219.
- NUNES, A. J. d. A. Jogos sérios em realidade virtual para ensino: um caso de estudo com aulas laboratoriais em ciências biológicas. 2022.
- OFOEDA, J.; BOATENG, R.; EFFAH, J. Application programming interface (api) research: A review of the past to inform the future. *International Journal of Enterprise Information Systems (IJEIS)*, IGI Global, v. 15, n. 3, p. 76–95, 2019.
- PANDOLFO, G.; D'AMBROSIO, A.; CANNAVACCIUOLO, L.; SICILIANO, R. Fuzzy logic aggregation of crisp data partitions as learning analytics in triage decisions. *Expert Systems with Applications*, Elsevier, v. 158, p. 113512, 2020.
- SECRETAN, J.; WILD, F.; GUEST, W. Learning analytics in augmented reality: Blueprint for an ar/xapi framework. In: IEEE. *2019 IEEE International Conference on Engineering, Technology and Education (TALE)*. [S.l.], 2019. p. 1–6.
- SOHAIL, S.; KHANUM, A.; ALVI, A. Hybrid fuzzy-statistical system for learning analytics. In: IEEE. *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. [S.l.], 2018. p. 989–994.
- SONI, A.; RANGA, V. Api features individualizing of web services: Rest and soap. *International Journal of Innovative Technology and Exploring Engineering*, v. 8, n. 9, p. 664–671, 2019.
- STANČIN, I.; JOVIĆ, A. An overview and comparison of free python libraries for data mining and big data analysis. In: IEEE. *2019 42nd International convention on information and communication technology, electronics and microelectronics (MIPRO)*. [S.l.], 2019. p. 977–982.
- XU, D.; TIAN, Y. A comprehensive survey of clustering algorithms. *Annals of Data Science*, Springer, v. 2, p. 165–193, 2015.