

# Um ambiente extensível para Game Learning Analytics

Adilson Vahldick<sup>1</sup>, Gabriel Dolzan<sup>1</sup>, Marcelo de Souza<sup>1</sup>

<sup>1</sup> Departamento de Engenharia de Software  
Universidade do Estado de Santa Catarina (UDESC)  
Ibirama – SC – Brasil

{adilson.vahldick, marcelo.desouza}@udesc.br, gabriel.dolzan@hotmail.com

**Abstract.** *Game Learning Analytics (GLA) explores techniques for analyzing learning data within the context of educational objectives in serious games. This work presents an extensible Web platform for GLA, developed to collect, analyze, and visualize data from educational games created in Unity. The platform allows users to customize the data to be collected, configure different chart types for visualization, and incorporate new mathematical functions for analysis. Its effectiveness was validated with two games, one focused on physics and the other on history, and an evaluation using NASA-TLX indicated that the platform imposes a low workload on users.*

**Resumo.** *Game Learning Analytics (GLA) explora técnicas de análise de dados de aprendizagem no contexto de objetivos educacionais em jogos sérios. Este trabalho apresenta uma plataforma Web extensível para GLA desenvolvida para coletar, analisar e visualizar dados de jogos educacionais criados no Unity. A plataforma permite personalizar os dados a serem coletados, configurar diferentes tipos de gráficos para visualização e incorporar novas funções matemáticas para análise. Sua eficácia foi validada com dois jogos, um de física e outro de história, e uma avaliação utilizando o NASA-TLX indicou que a plataforma impõe baixa carga de trabalho aos usuários.*

## 1. Introdução

Os Jogos Sérios (JS) implicam sua aplicação em ambientes educacionais, e têm ganhado destaque devido à sua capacidade de criar experiências de aprendizagem envolventes e eficazes [Alonso-Fernandez et al. 2017]. A natureza inerentemente interativa dos jogos gera uma riqueza de dados sobre as ações e comportamentos dos estudantes [Alonso-Fernández et al. 2019]. O *Game Learning Analytics* (GLA) surge da análise desses dados para entender a validade e a eficácia dos JSs em relação aos seus objetivos educacionais.

GLA é definida como a coleta, análise e extração de informações a partir de dados coletados de JSs, baseando-se nos campos de mineração de dados educacionais e *learning analytics* [Loh et al. 2015]. Diferente da análise tradicional de jogos, que foca exclusivamente no jogo em si para fins de desenvolvimento ou financeiros, o GLA incorpora objetivos educacionais, permitindo que educadores explorem o que os jogadores aprenderam e avaliem a eficácia do jogo como ferramenta de ensino [Freire et al. 2023]. A aplicação de técnicas de ciência de dados ao GLA pode melhorar a avaliação dos jogos e permitir decisões baseadas em evidências, aumentando a confiança de professores e instituições no uso de JSs na educação formal [Alonso-Fernández et al. 2019].

Apesar do potencial evidente do GLA, a sua aplicação prática ainda enfrenta desafios significativos [Alonso-Fernandez et al. 2017]. Um desses desafios é a personalização da análise de dados, pois as ferramentas existentes oferecem conjuntos fixos de opções para exibição de dados, restringindo a capacidade de pesquisadores realizarem avaliações detalhadas [Junior e Menezes 2015]. A ausência de mecanismos de extensibilidade impede a adaptação do sistema às necessidades específicas de cada pesquisa ou aplicação educacional. Essa limitação na análise da aprendizagem leva à dependência de avaliações externas, como pré e pós-testes, dificultando um estudo aprofundado dos impactos e benefícios dos JSs [Oliveira et al. 2022].

Este trabalho propõe o desenvolvimento de uma plataforma Web de GLA para jogos sérios que incorpora mecanismos de extensibilidade. O objetivo é fornecer um ambiente onde desenvolvedores possam extrair informações para monitorar o desempenho dos estudantes nas tarefas do jogo. Para isso, é proposta uma API para conexão, envio e armazenamento de informações dos jogos. A aplicação Web permite ao pesquisador a criação de gráficos personalizáveis e a extensão do ambiente com novas funções de cálculo e visualizações.

A plataforma proposta foi integrada a dois jogos sérios, aplicados para o ensino de física [Vahldick et al. 2025] e história [Vahldick e Goedert 2025]. Nessa integração, dados como o tempo gasto pelo estudante em cada nível do jogo e o número de itens coletados foram usados pelos desenvolvedores para avaliar a aprendizagem. Essas informações foram visualizadas em gráficos, permitindo inferências entre o desempenho no jogo e os testes de avaliação externos aplicados aos estudantes. A usabilidade do ambiente foi avaliada utilizando o instrumento NASA-TLX [Hart e Staveland 1988].

O artigo está organizado da seguinte maneira: na próxima seção será descrita a metodologia adotada nesta pesquisa. A terceira seção discute os trabalhos relacionados que serviram de inspiração para o desenvolvimento desta pesquisa. A quarta seção apresenta o ambiente e suas funcionalidades. A quinta seção demonstra as validações desta pesquisa. Na última seção são explanadas as conclusões do trabalho.

## 2. Metodologia

Primeiramente, foram analisados trabalhos correlatos que possam prover inspiração com base nas limitações identificadas. A partir dessas lacunas, foram definidos os requisitos da plataforma. O *frontend* da plataforma foi implementado usando VueJS e Chart.js para a construção dos gráficos. O *backend* foi desenvolvido usando Python, Django e o banco de dados PostgreSQL.

Dois jogos foram desenvolvidos e validados usando a plataforma: um jogo para dispositivos móveis usado no ensino de física [Vahldick et al. 2025] e outro jogo para computador usado no ensino de história local [Vahldick e Goedert 2025]. Após a validação dos jogos, cada desenvolvedor respondeu ao instrumento NASA-TLX [Hart e Staveland 1988], que fornece uma avaliação quantitativa global da carga de trabalho, baseada na média ponderada da avaliação de seis dimensões listadas na Tabela 1. Esse instrumento é utilizado para (1) avaliar a carga de trabalho associada a tarefas ou sistemas específicos [Torres e Barwäldt 2021]; (2) identificar aspectos de tarefas ou sistemas que talvez contribuam para uma excessiva carga de trabalho [Silva et al. 2019]; e (3) comparar uma tarefa em diferentes condições [Millais et al. 2018]. Além disso, foram

**Tabela 1. Dimensões do NASA-TLX [Cardoso e Gontijo 2012].**

<b>Dimensão</b>	<b>Descrição</b>
Demanda mental	Quantidade da atividade mental e perspectiva que a tarefa necessita (pensar, decidir, calcular, lembrar, olhar, procurar, etc.).
Demanda física	Quantidade da atividade física que a tarefa necessita (puxar, empurrar, girar, deslizar, etc.).
Temporal	Nível de pressão temporal sentida. Razão entre o tempo necessário e o disponível.
Desempenho	Até que ponto o indivíduo se sente satisfeito com o nível de rendimento e desempenho no trabalho.
Esforço	Grau de esforço mental e físico que o sujeito tem que realizar para obter seu nível de rendimento.
Frustração	Até que ponto o sujeito se sente inseguro, estressado, irritado, descontente, etc. durante a realização da atividade.

solicitadas aos desenvolvedores sugestões de melhorias.

### 3. Trabalhos correlatos

Com a intenção de identificar ideias e lacunas para a implementação desse ambiente extensível para GLA, foram encontradas na literatura dois ambientes com a proposta de gerenciar e apresentar dados dos jogos.

Em [Silva et al. 2022] são enviados os dados dos jogos por meio de uma API, e o sistema disponibiliza a análise desses dados usando *dashboards* na Web. Esses dados seguem um padrão que limita os tipos de informações a serem armazenados. Essa limitação acontece porque o *dashboard* é formado por análises específicas desses dados, e.g. análises gerais dos dados (quantidade de jogadores, jogadores por gênero, tempo total em jogo, quantidade de tentativas por fase, etc.), análise individual por jogador (as mesmas informações da análise geral, mas por jogador) e análises de grupos (avaliação de relações, como entre tempo e tentativas, entre derrotas e tentativas, etc.). Existem três perfis de usuários para acesso ao sistema Web e, por consequência, aos *dashboards*: desenvolvedor, professor e jogador. Foi desenvolvido um pacote de integração com a Unity.

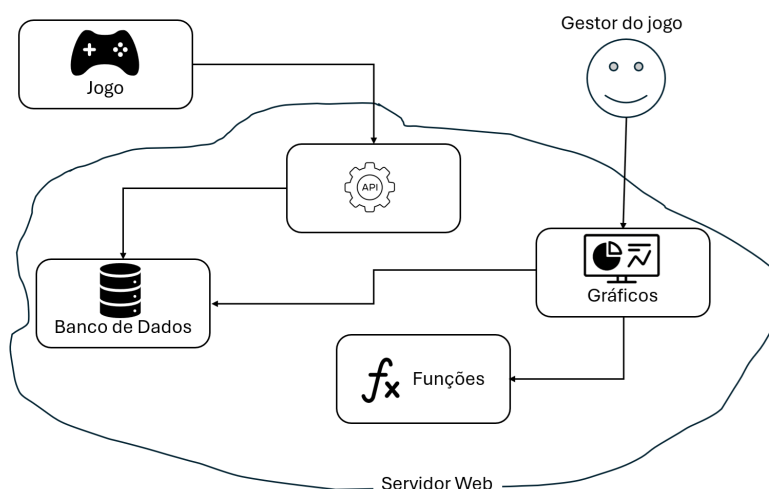
A proposta de [Calvo-Morata et al. 2025] tem suporte mais abrangente para a GLA. Além da coleta de dados de interação dos jogos, a plataforma suporta dados da validação experimental (pré e pós-testes, e testes A/B). Os dados são enviados no padrão IEEE xAPI<sup>1</sup>, com pacotes de integração desenvolvidos com a Unity e em Javascript. O sistema gera gráficos padrões, que podem demonstrar análises gerais (itens acessados, pontos totais no jogo, pontos em determinada fase, tempo para completar determinada fase, etc.) e essas mesmas análises por jogador. Além disso, existem modelos de predição que identificam a relevância de variáveis. São testados seis modelos de predição sobre um conjunto de variáveis escolhido pelo usuário, e com base na maior acurácia das previsões, é selecionado um modelo para avaliar a relevância de cada variável. Existem quatro perfis de usuário para acesso a diferentes ferramentas no sistema: professor, pesquisador, desenvolvedor e cientista da educação.

<sup>1</sup><https://standards.ieee.org/ieee/9274.1.1/7321/>

#### 4. Dolzanlytics

A plataforma proposta é um ambiente Web, que permite coletar dados de jogos desenvolvidos com Unity e exibi-los como gráficos, chamada *Dolzanlytics*. A Figura 1 ilustra a visão geral da aplicação, onde pode-se visualizar as três propriedades de extensibilidade da plataforma:

1. Dados a serem coletados pelos jogos: para cada jogo, é possível definir os artefatos a serem guardados nas sessões de jogos. Esses artefatos são definidos por jogo e não há limitação quanto ao tipo ou natureza da informação;
2. Informações a serem exibidas em um gráfico: refere-se a um tipo de visualização para um agrupamento de informações. Existem seis tipos de gráficos que resumem os dados de acordo com os artefatos e/ou com os jogadores. O usuário configura quais artefatos deseja visualizar em qual tipo de gráfico;
3. Funções matemáticas a serem usadas no gráfico: os gráficos necessitam de funções para sumarizar dados. Estão disponíveis funções de soma e contagem. Servindo de um mecanismo de programação, é possível adicionar novas funções na plataforma.



**Figura 1. Visão geral da aplicação.**

Na Figura 2 são apresentadas as tecnologias utilizadas no desenvolvimento. Para o desenvolvimento do *frontend* foi utilizado o *framework* JavaScript VueJS para criar interfaces de usuário em aplicações web de maneira fácil e flexível. Além disso, para a montagem dos gráficos, foi utilizada a biblioteca JavaScript Chart.js, que oferece uma maneira simples e flexível de representar dados através de uma variedade de tipos de gráficos, facilitando a exibição e interpretação visual de informações. As páginas HTML fazem requisições assíncronas usando a biblioteca AXIOS. Já no desenvolvimento *backend*, primeiro foi necessário configurar o servidor web, e nesse caso foi utilizado o Nginx, que recebe as requisições e as direciona para o Gunicorn. O Gunicorn é um servidor de aplicação que executa código em Python. Os scripts em Python utilizam o Django como *framework* para desenvolvimento de aplicações web. Todos os dados são salvos em um banco de dados PostgreSQL. Os jogos desenvolvidos em Unity enviam requisições para o servidor web, e não espera que sejam respondidas.

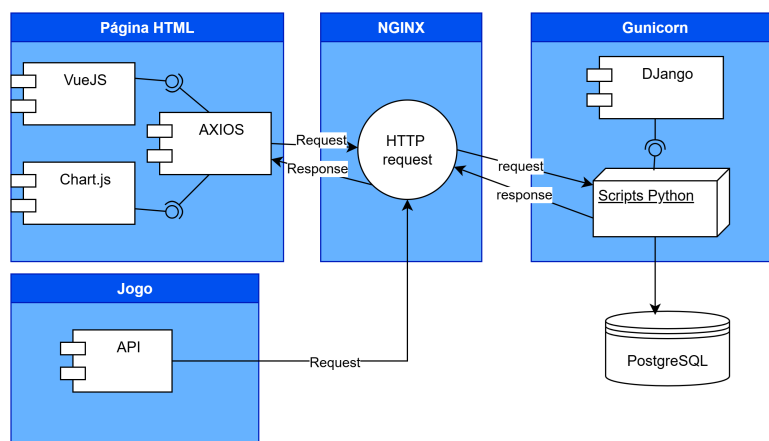


Figura 2. Arquitetura do sistema.

#### 4.1. Dados a serem coletados pelos jogos

O desenvolvedor acessa o sistema e cadastra o nome do seu jogo. Após o cadastro, o sistema informa um número inteiro que corresponde à chave a ser usada para acessar o sistema a partir do jogo desenvolvido na Unity.

Para cada jogo podem ser adicionados os artefatos, que correspondem aos dados a serem armazenados. Para cada artefato deve ser definido o tipo: numérico, data/hora, intervalo, texto e verdadeiro/falso. Esses artefatos podem estar relacionados às tarefas no jogo, como a quantidade de tentativas, tempo para a tentativa ser concluída, ou a resposta fornecida em uma tarefa. Logo, um artefato é composto de um identificador que deve ser fornecido com a chave do jogo, seu tipo, e uma descrição usada pelo usuário do sistema para identificar o artefato na geração dos gráficos.

A Figura 3 ilustra parte da classe a ser usada na Unity para envio dos dados ao servidor. O atributo `apiUrl` deve apontar para o endereço do servidor. A Figura 4 demonstra quatro exemplos com valores fixos, mas que podem ser obtidos de variáveis, quando implementados no jogo. O código mostra o fornecimento de dados referentes ao jogo de código 1. Duas linhas referem-se ao jogador com *nick* “fulano” e duas para o “ciclano”. Os artefatos exemplificados têm o nome de “tentativas\_modlex3” e “tempo\_modlex3”.

```

1  ...
2  api.SendDataToAPI(1, "fulano", "tentativas_modlex3", "1");
3  api.SendDataToAPI(1, "ciclano", "tentativas_modlex3", "4");
4
5  api.SendDataToAPI(1, "fulano", "tempo_modlex3", "73");
6  api.SendDataToAPI(1, "ciclano", "tempo_modlex3", "259");
7  ...

```

Figura 4. Exemplo de uso no Unity.

#### 4.2. Informações a serem exibidas em um gráfico

Não existem gráficos prontos no ambiente. Todos eles são criados a partir das necessidades dos usuários. Quando o usuário elabora um gráfico, ele inicia selecionando o jogo, e

```

1  public class API : MonoBehaviour {
2      private string apiUrl = "http://localhost/api/data/";
3
4      public void SendDataToAPI(int game, string player,
5                               string artifact, string value) {
6          ApiData data = new ApiData(game, player, artifact, value);
7          StartCoroutine(SendReq(JsonUtility.ToJson(data)));
8      }
9      private IEnumerator SendReq(string jsonData) {
10         ...
11     }
12
13 }

```

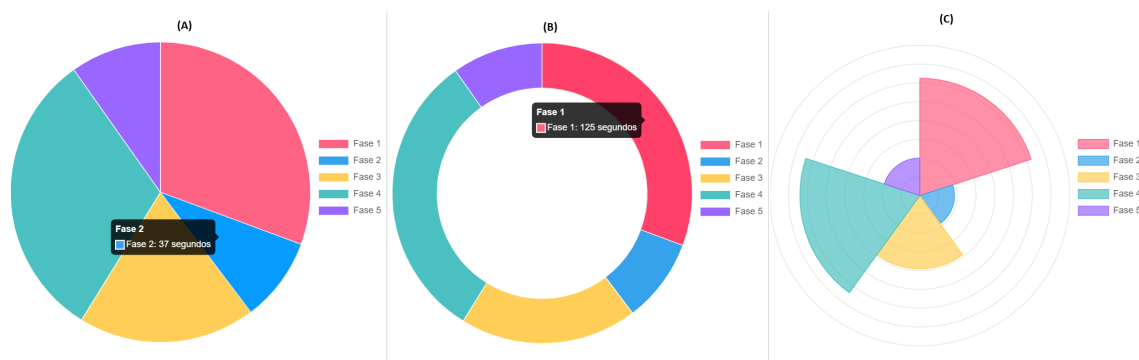
**Figura 3. Classe da Unity responsável por enviar a requisição.**

o sistema lhe fornece a identificação de todos os jogadores e artefatos do jogo. O usuário seleciona os artefatos desejados e pode selecionar um ou mais jogadores. Em seguida, ele seleciona uma das cinco funções disponíveis, três somadoras e duas contadoras:

- Somatório (todos os artefatos): soma os valores de todos os artefatos selecionados. Se houver jogadores selecionados, considera somente os artefatos desses jogadores;
- Somatório (por identificador): soma os valores de todos os artefatos selecionados agrupado por jogador;
- Somatório (por artefato): soma os valores dos artefatos agrupando por artefato. Se houver jogadores selecionados, considera somente os artefatos desses jogadores;
- Ocorrência (quantidade por valor): conta quantas vezes aparece cada um dos valores dos artefatos selecionados. Se houver jogadores selecionados, considera somente os artefatos desses jogadores;
- Ocorrência (quantidade por artefato): conta quantas vezes aparece cada um dos artefatos selecionados. Se houver jogadores selecionados, considera somente os artefatos desses jogadores.

Em seguida, o usuário seleciona o tipo de gráfico. Existem seis tipos de gráficos: barra vertical, linha, pizza, rosca, polar e radar. Os gráficos de barra, linha e pizza são amplamente conhecidos. O gráfico de rosca é uma variação do gráfico de pizza, com uma diferença visual: enquanto o gráfico de pizza é um círculo completo, o gráfico de rosca tem um espaço vazio no centro, formando um anel. A versão em rosca tende a ser mais limpa visualmente, especialmente quando há muitas categorias, pois reduz a sobrecarga de informações no centro. As Figuras 5 (A) e 5 (B) ilustram a diferença entre ambas. Elas exemplificam os valores dos artefatos *TempoFase1*, *TempoFase2*, *TempoFase3*, *TempoFase4* e *TempoFase5* de um jogador.

O gráfico de área polar é outra variação do gráfico de pizza, porém todos os segmentos possuem o mesmo ângulo e o raio de cada segmento difere dependendo do valor. Ele é útil para apresentar a variação dos valores entre as categorias, sem se preocupar com relações entre eles. A Figura 5 (C) apresenta esse gráfico com o mesmo exemplo dos tempos em cada fase. O tamanho de cada “fatia” mostra claramente que as fases 1 e 4



**Figura 5. Diferença entre os gráficos de pizza (A) e rosca (B); gráfico de área polar (C).**

consumiram mais tempo.

O gráfico de radar é uma forma de apresentar múltiplos pontos de dados e a variação entre eles. Esse gráfico é útil para comparar dois ou mais conjuntos de dados. A Figura 6 exemplifica esse gráfico sendo usado para comparar os tempos entre as fases de dois jogadores. Por exemplo, ele mostra claramente que o Jogador 2 levou mais tempo na Fase 4. Uma última configuração possível na criação dos gráficos é determinar um período (data inicial a data final) em que os dados foram coletados.

### 4.3. Funções matemáticas a serem usadas no gráfico

Para permitir que os desenvolvedores incorporem novas funções de cálculo, foi desenvolvida uma classe abstrata denominada `FunctionBase` (Figura 7) que possui dois métodos abstratos. O primeiro é `get_descricao`, que retorna a descrição da função de cálculo que será exibida na tela. O segundo método é o `get_data`, que retorna as informações necessárias para montar o gráfico. Esse último método recebe por parâmetros as configurações de exibição. Todas as funções se encontram no arquivo `functions.py` e são mapeadas automaticamente sempre que uma nova classe é incluída e esta implementa a classe `FunctionBase`. As cinco funções existentes, que servem para compreender a implementação de outras, estão ilustradas em <http://bit.ly/3JyrrjU>.

## 5. Validação e discussão dos resultados

Dois jogos foram desenvolvidos e validados usando a plataforma Dolzanlytics: um jogo para dispositivos móveis usado no ensino de física [Vahldick et al. 2025] e outro jogo para computador usado no ensino de história local [Vahldick e Goedert 2025].

### 5.1. Jogo de Física

A amostra do jogo foi de 55 alunos e foram coletados 18 artefatos. Os participantes foram divididos em duas turmas (uma com 32 alunos e outra com 23 alunos), e então foi aplicado um estudo quase-experimental com delineamento cruzado. A Figura 8 apresenta os tempos gastos em cada um dos oito níveis por aluno em uma das turmas. Desse gráfico, é possível observar: (1) existiram alguns alunos atípicos (que gastaram mais tempo que os demais em alguns níveis); (2) foi consumido mais tempo no nível 2; (3) dois alunos

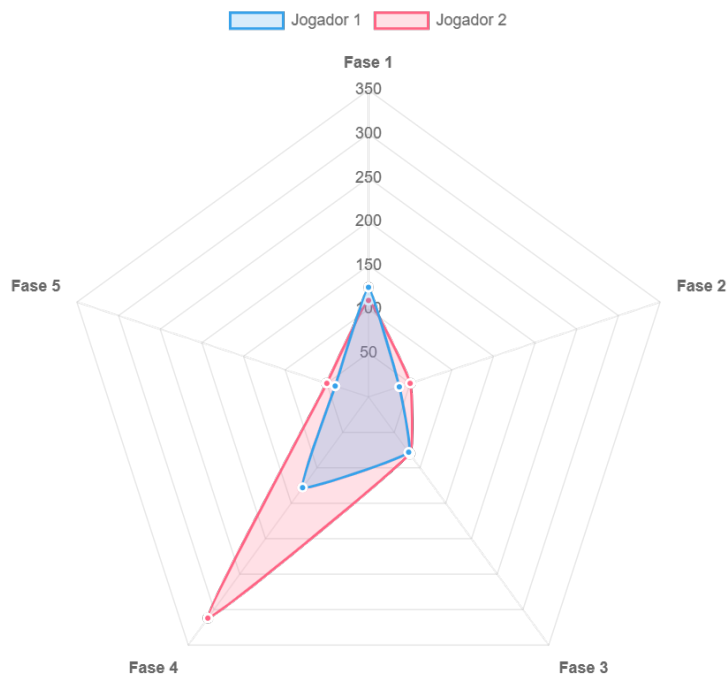


Figura 6. Gráfico de radar.

```

1  class FunctionBase(ABC):
2
3      @abstractmethod
4      def get_descricao(self):
5          pass
6
7      @abstractmethod
8      def get_data(self, tipo: int, game: int, artefato: list,
9                  id: list, inicio: str, final: str):
10         pass

```

Figura 7. Classe abstrata `FunctionBase` para implementação de novas funções de cálculo.

deixaram de jogar a partir do nível 2, e mais quatro não jogaram o nível 8; e (4) de forma geral, os tempos não aumentaram progressivamente, havendo variação entre os níveis.

Outro gráfico utilizado (Figura 9) tem relação com itens de bônus coletados pelos jogadores. Pode-se notar que a prioridade dos jogadores foi coletar ampliadores de tempo, que estendem o tempo final da conclusão do nível, e assim permitem que os alunos continuem jogando com mais tranquilidade.

## 5.2. Jogo de História Local

A amostra do jogo foi de 17 alunos e foram coletados seis artefatos. Diferente do jogo anterior, este experimento contou com somente uma turma. A Figura 10 apresenta os tempos gastos em cada um dos cinco níveis por aluno. Nota-se que no primeiro nível



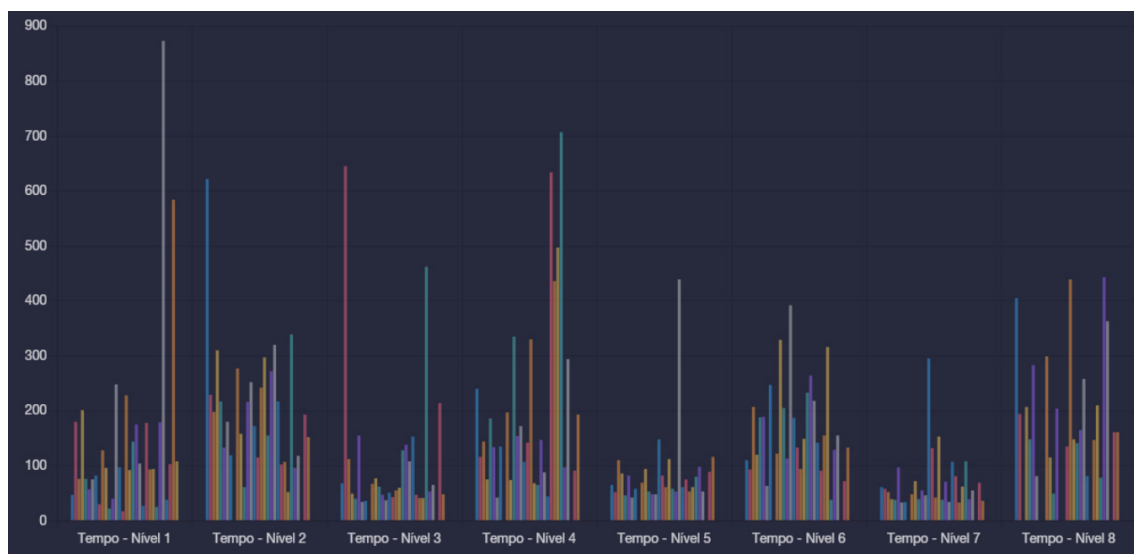


Figura 8. Gráfico de tempo por jogador em cada fase.

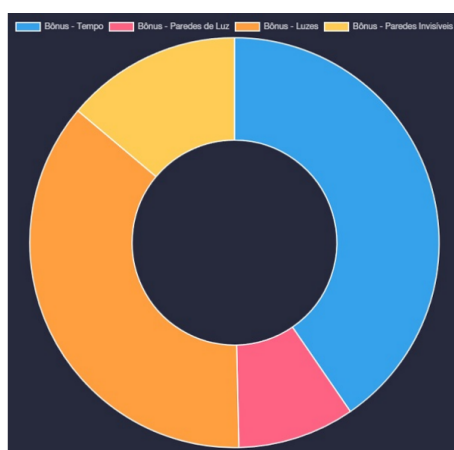


Figura 9. Gráfico de modificadores coletados.

foram gastos consideravelmente mais tempo do que nos demais níveis. Diferente do jogo anterior, não houve alunos atípicos em nenhum dos níveis.

### 5.3. Avaliação da Carga de Trabalho

Um dos objetivos no desenvolvimento dessa plataforma é diminuir o esforço para coletar e disponibilizar os dados do *gameplay*, e facilitar sua visualização. Para avaliar esse esforço, os dois pesquisadores, que foram responsáveis pela implementação dos jogos, responderam ao instrumento NASA-TLX, usando uma aplicação Web <sup>2</sup> que coleta e apresenta os resultados já calculados.

Primeiramente, é preciso responder a um valor entre 5 e 100 para as seis dimensões. Com exceção do desempenho, quanto menor o valor da resposta, menor a carga de trabalho sentida naquele aspecto. Quanto ao desempenho, quanto menor o valor, mais satisfeito o usuário está com seus objetivos. Em seguida, são mostrados 15 pares de

<sup>2</sup><https://www.keithv.com/software/nasatlx/nasatlx.html>

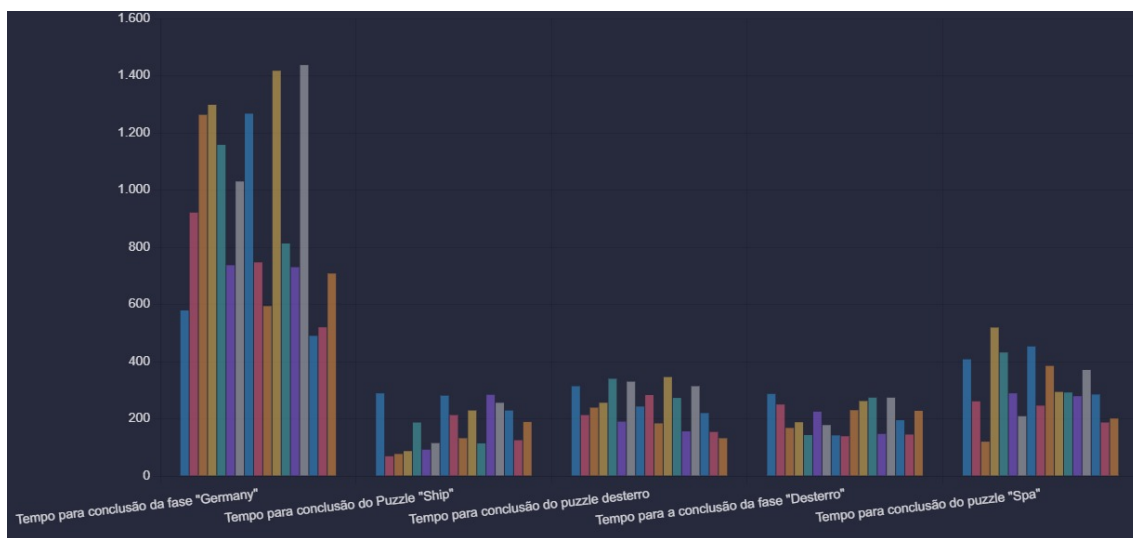


Figura 10. Gráfico de tempo por jogador em cada fase.

dimensões, e o usuário precisa selecionar qual das duas representa o fator mais relevante para determinar a carga de trabalho da tarefa que está sendo avaliada. Ao final, com base nas respostas desses 16 (1+15) formulários, o sistema calcula a soma ponderada da carga de trabalho. A Tabela 2 apresenta o resumo dessas duas avaliações.

Tabela 2. Respostas do NASA-TLX do desenvolvedor do jogo de física.

Dimensão	Física			História		
	Avaliação	Contagem	Peso	Avaliação	Contagem	Peso
Demanda Mental	30	5	0,333	40	4	0,266
Demanda Física	15	1	0,066	10	0	0,000
Demanda Temporal	15	2	0,133	25	3	0,200
Desempenho	5	2	0,133	20	5	0,333
Esforço	20	4	0,266	20	1	0,067
Frustração	5	1	0,066	15	2	0,133
<b>Soma ponderada</b>		<b>19,33</b>			<b>25,66</b>	

A dimensão com maior contribuição para a carga de trabalho no desenvolvimento dos dois jogos foi a *demanda mental*. Apesar dos valores dessa dimensão em ambas as tabelas serem bem menores que 100, representando que nela não houve carga de trabalho expressiva, era esperado que ela fosse a maior neste trabalho, pois é uma tarefa de programação e decisão de quais seriam as informações a serem coletadas e armazenadas.

Além disso, para o jogo de física, o desenvolvedor apontou o *esforço* como a dimensão com segunda maior relevância. Para o segundo jogo, pode-se notar que o desenvolvedor apontou a *demanda temporal* nesse quesito. Isso pode ser explicado pelo prazo de entrega e aplicação dos jogos: o desenvolvedor do jogo de história estava mais atrasado com seu trabalho e sentiu maior pressão em relação ao tempo.

De qualquer forma, em uma escala de 5 a 100, a soma ponderada de ambos foi de 19,33 (jogo de física) e 25,66 (jogo de história). Isso demonstra que os desenvolvedores concluíram que usar o Dolzanlytics requer pouca carga de trabalho, pois assim são

classificados valores abaixo de 30 [Hart e Staveland 1988].

#### 5.4. Feedback dos participantes

Ao final da experiência, os desenvolvedores foram questionados sobre sugestões para a melhoria da plataforma. Eles sugeriram (1) adicionar mais funções matemáticas como soma dos artefatos por identificador, média, mediana, desvio padrão e correlação; (2) por consequência, para visualizar as correlações serão necessários novos gráficos de dispersão e mapa de calor; (3) implementar uma interface de criação de *dashboards* personalizados, onde o usuário possa combinar múltiplos gráficos e resultados das funções em uma única visualização interativa.

### 6. Conclusão

Este trabalho apresentou o Dolzanlytics, um ambiente extensível para análise de *Game Learning Analytics* (GLA), com o objetivo de facilitar a coleta, processamento e visualização de dados de jogos educacionais. A plataforma foi desenvolvida para ser flexível, permitindo a definição personalizada de artefatos de dados, tipos de gráficos e funções matemáticas, atendendo às necessidades específicas de desenvolvedores e pesquisadores. Sua arquitetura aberta permite evoluções contínuas, alinhando-se às demandas crescentes por análises personalizadas em jogos educacionais.

Os resultados da validação, realizada com dois jogos (um de física e outro de história local), demonstraram a eficácia da plataforma. Os gráficos gerados permitiram identificar padrões de jogabilidade, como a variação de tempo entre níveis e a preferência por certos itens. Além disso, a avaliação da carga de trabalho com o NASA-TLX mostrou que os desenvolvedores consideraram o uso do Dolzanlytics de baixa demanda (somas ponderadas de 19,33 e 25,66), destacando sua usabilidade.

Em comparação com os trabalhos correlatos, o Dolzanlytics se diferencia pela extensibilidade. Enquanto as outras ferramentas limitam-se a gráficos e análises pré-definidas, a proposta deste trabalho permite adaptações dinâmicas, com a personalização dos gráficos.

Uma das sugestões apresentadas pelos desenvolvedores que utilizaram o Dolzanlytics está alinhada com as abordagens encontradas nos trabalhos correlatos: a implementação de *dashboards* interativos que integrem múltiplos gráficos em uma única visualização. Além das sugestões feitas pelos desenvolvedores, outras inspirações nessas pesquisas relacionadas são a adoção de técnicas de aprendizagem de máquina, e a expansão dos pacotes de integração com ferramentas de desenvolvimento de jogos como Godot e RPG Maker, bem como com linguagens de programação como JavaScript, Python e Java.

### 7. Agradecimentos

Os autores agradecem o apoio financeiro na infraestrutura do Grupo de Pesquisa Engenharia e Desenvolvimento de Tecnologia da Informação pela Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC) através do Termo de Outorga nº 2023TR000246.

## Referências

- Alonso-Fernandez, C., Calvo, A., Freire, M., Martinez-Ortiz, I., e Fernandez-Manjon, B. (2017). Systematizing game learning analytics for serious games. In *2017 IEEE global engineering education conference (EDUCON)*, pages 1111–1118. IEEE.
- Alonso-Fernández, C., Calvo-Morata, A., Freire, M., Martínez-Ortiz, I., e Fernández-Manjón, B. (2019). Applications of data science to game learning analytics data: A systematic literature review. *Computers & Education*, 141:103612.
- Calvo-Morata, A., Santilario-Berthilieres, J., Alonso-Fernández, C., Freire, M., Martínez-Ortiz, I., e Fernández-Manjón, B. (2025). Full software support for game learning analytics. In *Learning Analytics Summer Institute Spain 2025*.
- Cardoso, M. d. S. e Gontijo, L. A. (2012). Avaliação da carga mental de trabalho e do desempenho de medidas de mensuração: Nasa tlx e swat. *Gestão & Produção*, 19:873–884.
- Freire, M., Serrano-Laguna, Á., Manero Iglesias, B., Martínez-Ortiz, I., Moreno-Ger, P., e Fernández-Manjón, B. (2023). Game learning analytics: Learning analytics for serious games. In *Learning, design, and technology: An international compendium of theory, research, practice, and policy*, pages 3475–3502. Springer.
- Hart, S. G. e Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, 52(C):139–183.
- Junior, H. e Menezes, C. (2015). Modelo para um framework computacional para avaliação formativa da aprendizagem em jogos digitais. *XIV Simpósio Brasileiro de Games e Entretenimento Digital SBGames, Trilha da Cultura*, pages 819–828.
- Loh, C. S., Sheng, Y., e Ifenthaler, D. (2015). Serious games analytics: Theoretical framework. *Serious games analytics: Methodologies for performance measurement, assessment, and improvement*, pages 3–29.
- Millais, P., Jones, S. L., e Kelly, R. (2018). Exploring data in virtual reality: Comparisons with 2d data visualizations. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 5–10.
- Oliveira, R., Goya, D., e Rocha, R. (2022). Proposta de metodologia para avaliação da aprendizagem de alunos em jogos sérios. In *Congresso Brasileiro de Informática na Educação (CBIE)*, pages 180–185. SBC.
- Silva, D., Pires, F., Melo, R., e Pessoa, M. (2022). Glboard: um sistema para auxiliar na captura e análise de dados em jogos educacionais. In *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 959–968. SBC.
- Silva, E., Franco, N., Ferro, M., e Fidalgo, R. (2019). Mental Workload Impact of a Visual Language on Understanding SQL Queries. In *Anais do XXX Simpósio Brasileiro de Informática na Educação (SBIE 2019)*, pages 239–248.
- Torres, M. J. R. e Barwäldt, R. (2021). Workspace Awareness Acessível: estratégias de sonificação para projetar interfaces colaborativas acessíveis aos cegos. In *Anais do XXXII Simpósio Brasileiro de Informática na Educação (SBIE 2021)*, pages 619–629.

Vahldick, A. e Goedert, M. (2025). A história da vinda dos imigrantes alemães para santa catarina experienciada num jogo digital. In *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*. SBC.

Vahldick, A., Silva, L. F. d., e Marquez, V. A. G. (2025). Um jogo digital para explorar o aprendizado de buracos negros e da radiação hawking. *Revista Brasileira de Ensino de Física*, 47:e20250026.