

Educação em Algoritmos e Estruturas de Dados no Contexto do Ensino Baseado em Jogos Digitais

Algorithms and Data Structures Education in the context of Digital Game-Based Teaching

Luciane Telinski Wiedermann Agner¹, Marieli Aparecida Ferreira Thomen¹, Gabriel Mauricio Utzig¹, Inali Wisniewski Soares¹

¹ Universidade Estadual do Centro-Oeste (Unicentro), Brasil

lagner@unicentro.br, mari_jawwad@hotmail.com, gabriel.utzig26@gmail.com, inali@unicentro.br

Recibido: 10/12/2020 | Corregido: 01/06/2021 | Aceptado: 27/06/2021

Cita sugerida: L. T. Wiedermann Agner, M. A. Ferreira Thomen, G. M. Utzig, Inali Wisniewski Soares, "Educação em Algoritmos e Estruturas de Dados no Contexto do Ensino Baseado em Jogos Digitais," *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, no. 30, pp. 85-92, 2021. doi: 10.24215/18509959.30.e9

Esta obra se distribuye bajo **Licencia Creative Commons CC-BY-NC 4.0**

Resumo

Em Ciência da Computação a disciplina de Algoritmos e Estruturas de Dados visa fornecer aos alunos a compreensão de tipos de dados abstratos, incluindo sua especificação, implementação e aplicação. Um dos maiores desafios no ensino é demonstrar a natureza dinâmica dos conceitos relacionados. Introduzir o desenvolvimento de jogos em disciplinas que envolvem lógica, algoritmos e programação como tópico para uma tarefa ou projeto é uma abordagem que permite aplicar os conceitos em problemas reais. Os requisitos e as características dos jogos possibilitam a realização de diversas atividades de programação e lógica, bem como o uso de estruturas de dados. Este artigo apresenta um jogo de lógica digital baseado em estruturas de dados. O jogo proposto visa aprimorar habilidades cognitivas de forma lúdica e foi desenvolvido no contexto da Aprendizagem Baseada em Projetos. Como resultado foi possível aplicar um importante conceito da computação em um exemplo real no contexto da gamificação.

Palavras chave: Lógica; Algoritmos; Estrutura de dados; Jogos; Aprendizagem baseada em projetos.

Abstract

In Computer Science the course of Algorithms and Data Structures aims to provide students with an understanding of abstract data types, including their specification, implementation and application. One of the biggest challenges in teaching is to demonstrate the dynamic nature of related concepts. Introducing game development in courses that involve logic, algorithms and programming as a topic for a task or project is an approach that allows applying the concepts to real problems. The requirements and characteristics of the games make it possible to carry out various programming and logic activities, as well as the use of data structures. This article presents a digital logic game based on data structures. The proposed game aims to improve cognitive skills in a playful way and was developed in the context of Project Based Learning. As a result, it was possible to apply an important concept of computing to a real example in the context of gamification.

Keywords: Logic; Algorithms; Data structure; Games; Project-Based learning.

1. Introdução

Em cursos de Ciência da Computação e áreas afins, o estudo de Algoritmos e Estrutura de Dados engloba a organização, a manipulação e o uso dos dados em um computador [1]. Nesse contexto, estruturas de dados lineares permitem representar um conjunto de dados, organizados de maneira sequencial, de forma a preservar a relação de ordem entre seus elementos. Dentre as estruturas lineares mais utilizadas destacam-se a Pilha e a Fila [2].

O ensino de estruturas de dados é desafiador e exige um grande esforço para entendimento dos conteúdos envolvidos, visto que engloba lógica de programação e trata da compreensão de estruturas abstratas, conceitos considerados complexos pela maioria dos alunos [3], [4].

Estudos demonstram que, em geral, os alunos consideram os conteúdos muito abstratos, o que acarreta em falta de interesse e de motivação e, geralmente, resulta em índices significativos de evasão e reprovação [5]. Desse modo, a atualização das didáticas de ensino é necessária para tornar o processo de aprendizagem mais atrativo. Para tal, é preciso transformar processos abstratos em concretos, utilizando abordagens que facilitem a compreensão dos tópicos abordados e possibilitem uma maior interação entre aluno-professor.

O uso de jogos na área de educação é uma tendência dentre as abordagens de ensino. Os principais benefícios são maior engajamento, motivação, desenvolvimento da criatividade e da autonomia dos alunos, além de possibilitar aprendizagem de maneira lúdica e a aplicação dos conteúdos em problemas do mundo real [6],[7].

Este artigo apresenta um jogo de lógica digital que visa aperfeiçoar habilidades como raciocínio lógico e concentração dos jogadores. O jogo oferece uma interface atrativa e de fácil uso.

A implementação foi realizada dentro de uma atividade proposta no contexto de Aprendizagem Baseada em Projetos (PBL - *Project Based Learning*) em uma disciplina de Algoritmos e Estruturas de Dados I do curso de Bacharelado em Ciência da Computação da Universidade Estadual do Centro-Oeste, no Brasil. Esta atividade possibilitou visualizar a aplicação de importantes temas da computação em problemas reais. Desse modo, o aprendizado foi reforçado por meio de exemplos práticos e divertidos que apresentaram novas possibilidades de aplicação dos conceitos na resolução de problemas computacionais.

2. O Ensino de Algoritmos e Estruturas de Dados

Disciplinas de Algoritmos e Estruturas de Dados contemplam o ensino de diversas estruturas lineares e não-lineares. Dentre essas, destacam-se a Pilha e a Fila,

estruturas de dados lineares que possibilitam a organização de dados de forma linear [2].

A Pilha é uma lista do tipo LIFO (*Last-in, First-out*), onde o último item inserido será o primeiro a ser removido. A função que insere elementos na Pilha é, geralmente, denominada de *push*. Por sua vez, a função que remove elementos da Pilha é comumente chamada de *pop*. O topo da Pilha é a posição de referência para inserir e remover novos elementos deste tipo de estrutura. A Pilha é largamente utilizada na solução de diversos problemas computacionais, em aplicações como sistemas de gerenciamento de memória, compiladores e interpretadores, por exemplo [1], [2].

A Fila é uma lista FIFO (*First-in, First-out*). O primeiro elemento armazenado na Fila será o primeiro a ser retirado. Diversas aplicações computacionais utilizam este tipo de estrutura, como por exemplo sistemas operacionais, simuladores e sistemas de redes de computadores [1], [2].

Os principais desafios no que tange o aprendizado dessas estruturas é compreender sua forma de funcionamento e o uso da mesma em sistemas computacionais. Desse modo, torna-se importante definir estratégias que permitam aos alunos visualizar a aplicação prática das estruturas estudadas em problemas reais, visando facilitar o entendimento e melhorar a fixação dos conceitos. Nesse contexto, a gamificação pode ser utilizada como estratégia para incentivar o uso das estruturas de dados e aumentar o interesse dos alunos pela disciplina [3], [7].

No contexto da educação, a gamificação consiste no uso de jogos sérios voltados para o ensino de forma lúdica e dinâmica. O desenvolvimento de jogos pode ser utilizado na forma de objeto de aprendizagem, visando melhorar o entendimento dos conceitos por meio da aplicação das estruturas de dados associadas à lógica de implementação [8].

3. Jogo de Lógica Baseado em Estrutura de Dados

Jogos digitais possibilitam realizar atividades lúdicas e estruturadas que envolvem tomada de decisão, regras, desafios e metas [9]. O uso de jogos digitais como ferramenta de ensino permite aprimorar aspectos cognitivos como memória, raciocínio lógico e capacidade de resolução de problemas [10].

O jogo digital implementado, baseado em Pilha, é uma adaptação do 2048, um jogo de lógica que consiste em um tabuleiro 4 x 4 onde os campos são preenchidos com blocos contendo valores de potência de base 2 ou estão em branco [11]. Inicialmente, uma configuração de dois blocos com valores 2s ou 4s é apresentada em duas posições aleatórias do tabuleiro. Quatro movimentos estão disponíveis ao jogador: esquerda, direita, para cima ou para baixo (Figura 1). Um bloco aleatório com o valor 2

ou 4 é adicionado após cada movimento. O objetivo do jogo é atingir um bloco com valor igual a 2048.

A concepção do jogo envolveu a criação de um tabuleiro digital contendo blocos 4 x 4 (4 linhas e 4 colunas). Os movimentos do jogador pelo tabuleiro digital são realizados utilizando as setas do teclado (Figura 1).

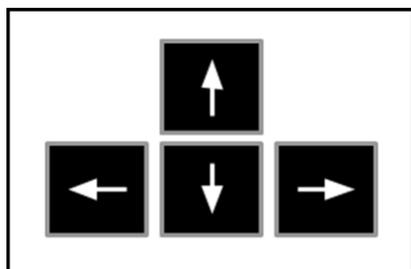


Figura 1. Setas de movimentação do jogo

Os blocos definidos no tabuleiro podem ser deslocados até o limite das bordas do tabuleiro ou, caso já exista um bloco na borda da linha ou coluna, até a posição nula mais próxima do respectivo bloco com base no sentido sinalizado pela seta.

3.1. Âmbito de desenvolvimento do jogo

O jogo foi desenvolvido em uma atividade proposta na disciplina de Algoritmos e Estruturas de Dados I (AEDI) do curso de Bacharelado em Ciência da Computação da Universidade Estadual do Centro-Oeste no ano de 2020, baseada no contexto de abordagens de Aprendizagem Baseada em Projetos (PBL - *Project Based Learning*). A ideia era o desenvolvimento de projetos pelos alunos que permitissem a aplicação, em cenários reais, dos conceitos relacionados com a disciplina.

Dentre os projetos apresentados, destaca-se um jogo de lógica que adaptou o clássico 2048 e utiliza a estrutura de dados Pilha, uma das mais utilizadas na área computacional.

As etapas de desenvolvimento do projeto do jogo, no contexto da disciplina de AEDI, foram as seguintes:

1. Definição do objetivo principal do projeto, sendo ele a criação de um jogo de lógica.
2. Especificação dos requisitos do jogo.
3. Modelagem do jogo.
4. Seleção da estrutura de dados mais adequada para implementação do jogo.
5. Criação da interface gráfica do jogo.
6. Implementação do jogo baseada na estrutura de dados Pilha.

No contexto da disciplina, a etapa 4 é fundamental para atingir o objetivo desta atividade, visto que requer do aluno a capacidade de identificar qual a estrutura de dados melhor se aplica para a solução do problema que está sendo implementado, com base nas características

específicas de cada uma. Desse modo, as habilidades do aluno em reconhecer e identificar situações adequadas para uso dos conceitos são exercitadas. Por sua vez, a etapa 6 exige do aluno a implementação utilizando a estrutura de dados escolhida na etapa 4, o que incentiva o aluno a aplicar o conceito na prática.

3.2. Modelagem do jogo

A modelagem do jogo foi realizada visando a criação de modelos (diagramas) que descrevem os requisitos e funcionalidades do software. Esses modelos possibilitam um maior entendimento do domínio da aplicação. O Diagrama de Casos de Uso do jogo implementado é apresentado na Figura 2.

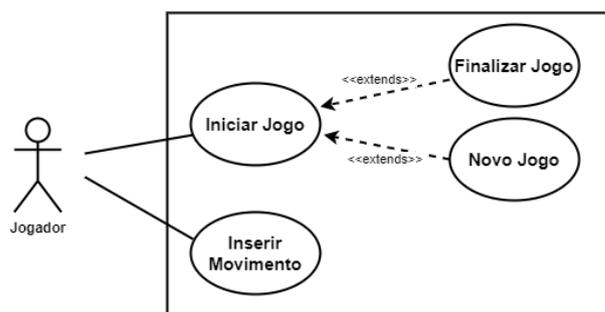


Figura 2. Diagrama de Casos de Uso do jogo

Um Diagrama de Casos de Uso representa graficamente os atores, casos de uso e relacionamentos, ilustrando em alto nível de abstração os elementos que interagem com as funcionalidades levantadas para o *software*. Os atores podem representar usuários, agentes externos e *hardware*, enquanto casos de uso representam as funcionalidades do sistema [12]. Este diagrama é baseado na linguagem de modelagem padrão da indústria de Engenharia de Software atualmente, a UML (*Unified Modeling Language*) [13].

No Diagrama de Casos de Uso, apresentado na Figura 2, o jogador representa o ator denominado "Jogador" e seu relacionamento com os seguintes casos de uso:

- "Iniciar jogo": caso de uso que possibilita ao jogador começar o jogo;
- "Finalizar jogo": fluxo alternativo do caso de uso "Iniciar jogo" que permite ao jogador fechar a janela do software pressionando a tecla ESC do teclado ou clicando no 'X' localizado no canto superior direito da janela para encerrar o jogo;
- "Novo jogo": fluxo alternativo de "Iniciar jogo" que possibilita ao jogador iniciar um novo jogo quando o jogador pressiona a tecla N. Nesse caso, o placar do jogador retorna ao valor zero e o tabuleiro apresenta o estado inicial do jogo, com dois blocos alocados aleatoriamente com valor 2 ou 4. O valor 2 possui maior probabilidade de ocorrer (60%), com base na lógica implementada, contra 40% de probabilidade de ocorrência do valor 4;

- “Inserir movimento”: caso de uso que representa como os movimentos devem ser realizados durante o jogo, usando as setas do teclado. Esse caso de uso também define um fluxo de exceção para garantir que teclas diferentes de “ESC”, “N” ou setas de movimentação, se pressionadas, não gerem nenhuma forma de interferência na partida.

3.3. Funcionamento do jogo

O funcionamento do jogo é detalhado no fluxograma apresentado na Figura 3. O jogo inicia com dois blocos alocados aleatoriamente no tabuleiro digital 4 x 4 (com valores 2s ou 4s) e com o placar zerado. A cada jogada o usuário realiza um movimento utilizando uma seta do teclado e toda movimentação adiciona um novo bloco ao tabuleiro. Conforme os requisitos definidos para o sistema, o valor 2 possui maior probabilidade (60%) com relação ao valor 4 (40%).

Quando o tabuleiro apresenta blocos com valores iguais e localizados em posições adjacentes com relação a qualquer uma das extremidades (acima, abaixo, esquerda ou direita), a soma desses blocos é realizada.

Os blocos iguais são somados e substituídos por um bloco com o resultado da soma final, o qual ficará na posição determinada pelo sentido da seta pressionada e dentro do limite das bordas do tabuleiro. Caso já exista um bloco na respectiva linha ou coluna, o valor da soma será armazenado na posição nula mais próxima ao respectivo bloco.

O processo de soma dos blocos acontece de forma a evitar somas em cadeia. Desse modo, quando dois blocos com valores iguais estão posicionados ao lado de um terceiro com o mesmo valor resultante da adição dos anteriores, a soma em cadeia não acontece. Por exemplo, supondo que existem dois blocos com valor 2 posicionados ao lado de um bloco 4 no tabuleiro. Ao unir os blocos com valor 2, a soma resulta em um novo bloco 4 que será inserido ao lado de outro bloco 4. Entretanto, não será realizada a soma para essa última combinação, evitando assim que o processo de soma seja repetido.

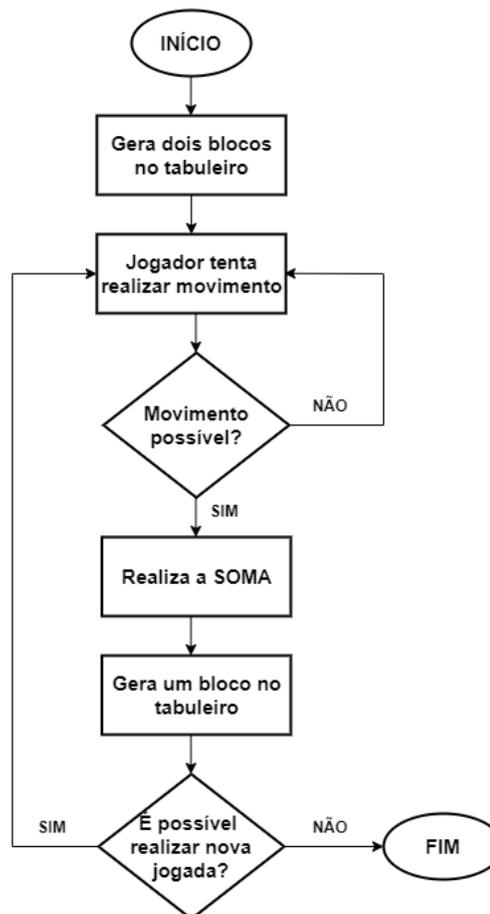


Figura 3. Fluxograma do jogo

O processo de movimentações e soma continua até que o objetivo do jogo seja atingido, ou seja, o valor 2048 seja alcançado. Quando isso ocorre, o jogador tem a opção de continuar jogando ou finalizar o jogo. Ainda, conforme as regras definidas pelo jogo, o jogador perde quando não é possível realizar a movimentação de blocos por falta de espaço livre no tabuleiro digital ou por impossibilidade de realizar um movimento. O objetivo principal não é atingir uma meta específica e finalizar a jogada, mas estimular o jogador na criação de táticas e estratégias para gerar a combinação do número 2048 (alvo deste jogo).

A Figura 4 apresenta a interface do jogo desenvolvido. O lado direito da interface exibe o tabuleiro digital 4 x 4. É possível observar que as informações sobre o jogo atual são apresentadas na lateral esquerda, sendo elas:

- “RECORDE”, mostra a maior pontuação atingida dentre as partidas jogadas;
- “PLACAR”, apresenta a pontuação atual;
- “NOVO JOGO”, apresenta a mensagem informando que ao pressionar a tecla N uma nova jogada será iniciada.



Figura 4. Interface do jogo.

Um dos recursos da interface implementada é a variação das paletas de cores. Toda vez que um novo jogo é iniciado, novas cores são apresentadas ao usuário. Para tal, a estrutura do jogo define comandos para gerar números aleatórios que combinados com o valor das células formam modelos de cores RGB (*Red, Green, Blue*). Isto possibilita dar uma aparência diferente para cada partida. Além disso, foram acrescentados ao jogo retornos sonoros a cada movimento realizado e também para indicar que o jogador atingiu o objetivo de criar uma peça com o valor 2048.

A Figura 5 demonstra uma etapa de execução do jogo, onde (a) e (b) representam dois momentos distintos do tabuleiro. Desse modo, o tabuleiro exibido em (b) é o resultado do tabuleiro representado em (a) após a seta “para cima” ser pressionada para o valor 2. Pode-se perceber que o bloco com valor 2 no fragmento (a) da Figura 5 é deslocado para uma posição acima no tabuleiro em (b), bem como um novo bloco é adicionado ao tabuleiro em (b). O novo bloco com o valor 2 é acrescentado no canto inferior esquerdo do tabuleiro (apresentado em destaque na figura). Isso ocorre de acordo com a regra do jogo que define que um bloco aleatório aparece no tabuleiro após a realização de um movimento.

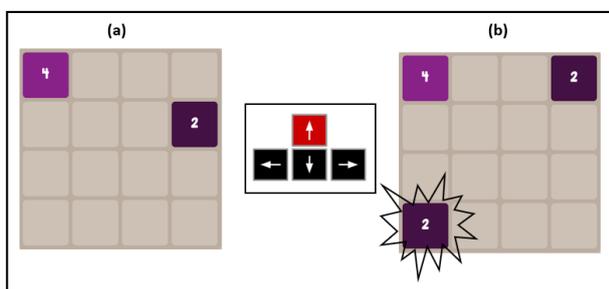


Figura 5. Exemplo do funcionamento do jogo ao pressionar a seta para cima

3.4. Mecânica do jogo

A mecânica do jogo é baseada no sentido indicado pelas setas do teclado (Figura 1). Desse modo, o código foi estruturado em módulos, onde cada módulo é responsável pelas ações a serem realizadas em cada um dos sentidos. Por exemplo, o módulo que trata da seleção da seta “para

baixo” define que todos os blocos do tabuleiro devem ser movimentados até a máxima posição abaixo quando esse sentido é selecionado. O mesmo acontece para as outras setas de movimentação, sempre com base no sentido da seta.

Cada módulo utiliza estruturas de repetição aninhadas para realizar os movimentos necessários. Essas estruturas visam percorrer as dimensões da matriz de 4 linhas por 4 colunas que representa o tabuleiro digital. Além disso, estruturas de seleção realizam o controle dos limites das bordas do tabuleiro.

Outra importante função do jogo é realizar os deslocamentos necessários no tabuleiro depois de selecionar cada movimento. Para tal, é preciso percorrer o tabuleiro representado por meio de uma matriz (4 linhas x 4 colunas). A Figura 6 apresenta um exemplo da ordem de acesso a cada elemento da matriz para realizar o deslocamento dos blocos para a possível posição abaixo, depois de selecionado um movimento de seta “para baixo”.

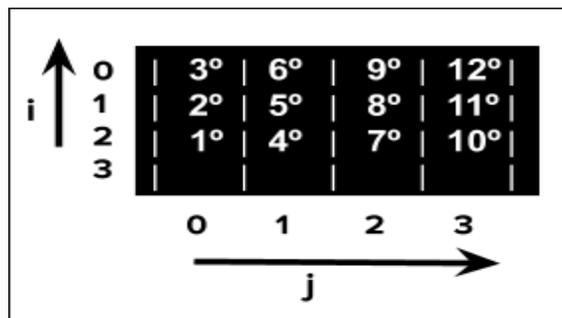


Figura 6. Ordem para percorrer o tabuleiro

3.5. Uso de estrutura de dados na criação do jogo

A seleção da estrutura de dados a ser utilizada na implementação do jogo deve refletir as características de manipulação dos valores a serem armazenados. A escolha da Pilha na implementação do jogo é justificada pela relação entre a característica LIFO (*Last-in, First-out*) desta estrutura e o problema de recuperar os últimos valores somados, resultantes das movimentações do jogador.

Quando uma movimentação realizada pelo jogador resulta em dois blocos com valores iguais localizados em posições adjacentes, a soma desses valores é realizada e o resultado é armazenado na Pilha, utilizando a função *push*. O respectivo valor somente é retirado da Pilha, utilizando a função *pop*, depois de realizar as verificações necessárias relacionadas aos limites do tabuleiro e ao critério de evitar somas em cadeia. Assim, o elemento retirado da Pilha é o último que foi inserido nesta estrutura, estabelecendo desta forma uma relação com o conceito de lista LIFO.

4. Trabalhos Relacionados

Diversos trabalhos da literatura tratam do ensino baseado no desenvolvimento de jogos digitais em cursos de Ciência da Computação. Alguns desses trabalhos serão apresentados a seguir.

Um projeto para ensino de Estrutura de Dados baseado no desenvolvimento de jogos combinado com programação competitiva é apresentado em [8]. Nessa abordagem os alunos competem para criar recursos em um jogo já existente. Segundo Lawrence [8] a concorrência aumenta o esforço do estudante na realização das atividades por meio de uma competição amigável que resulta em maior motivação dos alunos.

Uma análise do ensino de Ciência da Computação baseado no desenvolvimento de jogos digitais com o objetivo de desenvolver competências e habilidades requeridas pelas principais disciplinas do curso é apresentada por Silva e Martins [14]. Segundo os autores, atividades de aprendizagem que utilizam a construção de jogos trazem inúmeros benefícios como a melhora no engajamento e na motivação dos alunos para realização das atividades.

Um estudo sobre o uso de desenvolvimento de jogos como abordagem de ensino nas disciplinas de Lógica de Programação, Programação Orientada a Objetos e Estruturas de Dados é apresentado em [15]. Os resultados desse estudo apontam que essa abordagem estimula o aprendizado dos conceitos, aumenta a confiança e o interesse dos alunos, além de possibilitar o desenvolvimento das competências exigidas pelas disciplinas avaliadas.

Um levantamento sistemático sobre o uso de jogos no contexto de educação em disciplinas da área de Engenharia de Software é apresentado em [16], e inclui abordagens onde jogos pedagógicos são utilizados no aprendizado, estratégias onde os estudantes desenvolvem jogos como parte de um projeto, além de ambientes e ferramentas que oferecem suporte a gamificação.

5. Resultados e Discussões

O jogo digital proposto oferece desafios que possibilitam exercitar aspectos como raciocínio lógico, criatividade, atenção e concentração. Jogos digitais quando utilizados em práticas pedagógicas podem auxiliar no desenvolvimento cognitivo dos alunos. O software foi concebido buscando oferecer uma interface gráfica atrativa aos alunos e de fácil uso, por meio da representação de um tabuleiro digital e movimentações realizadas utilizando as setas do teclado. Além disso, a mecânica do jogo envolve estratégias, regras pré-definidas e estabelece relações entre as movimentações realizadas e as posições no tabuleiro. O jogo foi implementado utilizando a linguagem de programação C [17] e a biblioteca Allegro [18].

A criação do jogo está inserida no contexto de uma atividade proposta em uma disciplina de Algoritmos e Estruturas de Dados I que possibilitou aplicar os conceitos em projetos de interesse dos alunos, como é o caso do desenvolvimento de jogos digitais, resultando em aumento da motivação e do engajamento. Essa estratégia também proporcionou uma maior interação entre os alunos por meio da troca de experiências baseadas no uso de estruturas de dados no desenvolvimento de software.

A experiência aqui relatada visa incentivar o aluno a buscar soluções computacionais utilizando os conceitos abordados pela disciplina. Foi possível aos alunos trabalharem ativamente, de forma colaborativa, na produção de conhecimento sobre os temas envolvidos. O desenvolvimento do projeto permitiu aos alunos aperfeiçoar competências como autonomia, criatividade e iniciativa. Os resultados também levaram a inovação por meio da proposição de novas ideias.

Em metodologias tradicionais, o ensino geralmente acontece por transferência e absorção do conhecimento. Na experiência relatada neste artigo o aluno também é responsável pelo processo de aprendizagem e necessita se movimentar na direção de aprender, buscando soluções para problemas e cenários reais baseadas no uso de estruturas de dados. Também, nos métodos tradicionais as avaliações geralmente são padronizadas e seguem um formato comum. Por sua vez, na abordagem aqui descrita o aluno precisa desenvolver habilidades e competências para atingir seus objetivos.

Finalmente, em metodologias tradicionais é o professor quem detém o conhecimento sobre o tema, no entanto, na abordagem utilizada o professor agrega o papel de mediador, orientando os alunos no desenvolvimento de seus projetos.

Os resultados obtidos com este projeto podem ser utilizados como exemplo na proposição de novas estratégias que visem diminuir os problemas e demandas no ensino de disciplinas que envolvem lógica, entendimento de estruturas abstratas e requisitos de programação.

Conclusões

Abordagens de Aprendizagem Baseadas em Projetos apresentam-se como um recurso didático que possibilita experimentar os conceitos em problemas reais, visando melhorar a taxa de retenção dos conhecimentos por parte dos alunos [19]. A experiência descrita neste artigo une a abordagem PBL com técnicas de gamificação.

A gamificação aplicada ao ensino de Ciência da Computação é uma tendência que oferece diversos benefícios e pode ser descrita como o processo de usar a mecânica e o pensamento de jogos em áreas que não são de jogos, como meio para resolver problemas e engajar pessoas [20]. Entretanto, esta abordagem deve ser adotada de maneira planejada, visando fornecer novas experiências

para professores e alunos, além de fortalecer os conhecimentos obtidos.

Referências

- [1] A. M. Tenenbaum, Y. Langsam and M. J. Augenstein, *Estruturas de Dados usando C*. São Paulo: Makron Books, 1995.
- [2] D. Piva Junior, G. S. Nakamiti, F. Bianchi, R. L. Freitas and L. A. Xastre, *Estrutura de Dados e Técnicas de Programação*. Rio de Janeiro: Elsevier, 2014.
- [3] N. M. Bigolin et al., "Metodologias Ativas de Aprendizagem: um relato de experiência nas disciplinas de programação e estrutura de dados," *Research, Society and Development*, vol. 9, no. 1, e74911648, 2020.
- [4] J. I. Deters, J. M. C. da Silva, E. M. de Miranda and A. M. da R. Fernandes, "O desafio de trabalhar com alunos repetentes na disciplina de Algoritmos e Programação," in *Workshop de Ambientes de Apoio à Aprendizagem de Algoritmos e Programação*, Fortaleza, CE, Brasil, 2008.
- [5] R. Santos and H. Costa, "Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos Iniciantes em Computação e Informática," *INFOCOMP - Journal of Computer Science*, vol. 5, no. 1, pp. 41-50, 2006.
- [6] J. Lee and J. Hammer, "Gamification in education: What, how, why bother?," *Academic Exchange Quarterly*, vol. 15, no. 2, 2011.
- [7] D. Dicheva and A. Hodge, "Active Learning Through Game Play in a Data Structures Course," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, Baltimore, Maryland, USA. New York, NY, USA: ACM, pp. 834-839, 2018.
- [8] R. Lawrence, "Teaching data structures using competitive games," *IEEE Transactions on Education*, vol. 47, no. 4, pp. 459-466, 2004.
- [9] P. Schuytema, *Design de Games: Uma abordagem prática*. 1ª ed. São Paulo: Cengage Learning, 2008.
- [10] D. K. Ramos, "Jogos Cognitivos Eletrônicos: Contribuições à Aprendizagem no Contexto Escolar," *Ciências & Cognição*, vol. 18, no. 1, 2013.
- [11] V. Kaundinya et al., "Game Playing Agent for 2048 using Deep Reinforcement Learning," in *Proceedings of the 3rd National Conference on Image Processing, Computing, Communication, Networking and Data Analytics (NCICCND 2018)*, pp. 363-370, 2018. doi: <https://doi.org/10.21467/proceedings.1.57>
- [12] E. Bezerra, *Princípios de Análise e Projeto de Sistemas com UML*. 2ª ed. Rio de Janeiro: Elsevier, 2007.
- [13] D. Budgen, A. J. Burn, O. P. Brereton, B. A. Kitchenham and R. Pretorius, "Empirical evidence about the UML: a systematic literature review," *Software: Practice and Experience*, vol. 41, no. 4, pp. 363-392, 2011.
- [14] R. E. Silva and S. W. Martins, "Ensino de Ciência da Computação através do Desenvolvimento de Jogos," in *VII Congresso Iberoamericano de Informática na Educação*, Monterrey, pp. 1285-1295, 2004.
- [15] H. P. Pontes, "Desenvolvimento de Jogos no Processo de Aprendizado em Algoritmos e Programação de Computadores," in *XII Simpósio Brasileiro de Games e Entretenimento Digital (SBGames)*, São Paulo, 2013.
- [16] M. Kosa, M. Yilmaz, R. O'connor and P. Clarke, "Software engineering education and games: a systematic literature review," *Journal of Universal Computer Science*, vol. 22, no. 12, pp. 1558-1574, 2016.
- [17] P. Feofiloff, *Algoritmos em linguagem C*. Rio de Janeiro: Elsevier, 2009.
- [18] Allegro.c.c. Game developing community network. [Online]. Available: <https://www.allegro.cc/>. Accessed: 10 de Novembro de 2020.
- [19] M. J. O'Grady, "Practical Problem-Based Learning in Computing Education," *ACM Transactions On Computing Education*, vol. 12, no. 3, 2012, Art.10. doi: <https://doi.org/10.1145/2275597.2275599>.
- [20] G. Zichermann and J. Linder, *Game-based marketing: inspire customer loyalty through rewards, challenges, and contests*. Wiley, 2010.

Información de Contacto de los/as Autores/as:

Luciane Telinski Wiedermann Agner

Guarapuava, Paraná
Brasil

lagner@unicentro.br

ORCID ID: <https://orcid.org/0000-0001-9490-0808>

Marieli Aparecida Ferreira Thomen

Guarapuava, Paraná
Brasil

mari_jawwad@hotmail.com

ORCID ID: <https://orcid.org/0000-0001-6099-4375>

Gabriel Mauricio Utzig

Guarapuava, Paraná
Brasil

gabriel.utzig26@gmail.com

ORCID ID: <https://orcid.org/0000-0001-9354-6560>

Inali Wisniewski Soares

Guarapuava, Paraná
Brasil

inali@unicentro.br

ORCID ID: <https://orcid.org/0000-0002-5033-1815>

Luciane Telinski Wiedermann Agner

Professora de Departamento de Ciência da Computação da Universidade Estadual do Centro-Oeste (Unicentro). Doutora em Engenharia Elétrica e Informática Industrial (CPGEI) pela Universidade Tecnológica Federal do Paraná.

Marieli Aparecida Ferreira Thomen

Bacharelada em Ciência da Computação pela Universidade Estadual do Centro-Oeste (Unicentro).

Gabriel Mauricio Utzig

Bacharelado em Ciência da Computação pela Universidade Estadual do Centro-Oeste (Unicentro)

Inali Wisniewski Soares

Professora de Departamento de Ciência da Computação da Universidade Estadual do Centro-Oeste (Unicentro). Doutora em Engenharia Elétrica e Informática Industrial (CPGEI) pela Universidade Tecno