

ParsonFlow: Um jogo baseado em Problemas de Parson para auxiliar o processo de ensino-aprendizagem em programação

ParsonFlow: A game based on Parson Problems to support the teaching-learning process in programming

Andressa Mota S. Santos¹, Carlos Eduardo de Souza¹,

Luis Gustavo de Jesus Araujo¹

¹Instituto Federal de Educação, Ciência e Tecnologia da Bahia - Campus Jacobina (IFBA)
Jacobina – BA – Brasil

dhee.andressa@gmail.com, carlos.lcifba@gmail.com,
luis_araujo@ifba.edu.br

Introduction: Learning programming is becoming increasingly essential. However, it can be a challenging task due to the need for knowledge of logic and programming languages. **Objective:** This paper presents the development and design process of an educational puzzle game based on the Parson's Problem to support programming education. **Methodology:** The methodology consisted of five stages: designing the tool (defining the type of game and its rules), researching pedagogical theories, prototyping, validation, and development. **Expected Results:** That this game will assist students in introductory programming courses in understanding and solving Parson's Problems.

Keywords: Programming education. Programming logic. Parson's Problems. Educational technology. Educational tool.

Resumo. Introdução: O aprendizado de programação tem se tornado cada vez mais essencial. No entanto, aprender a programar pode ser uma tarefa desafiadora por necessitar de conhecimentos relacionados à lógica e à linguagem de programação. **Objetivo:** Este artigo apresenta o processo de desenvolvimento e concepção de um jogo educacional no estilo puzzle, baseado em Problema de Parson para ensino de programação. **Metodologia:** A metodologia compreendeu em cinco etapas: Idealização da ferramenta (definição do tipo do jogo e suas regras), Pesquisa de teorias pedagógicas norteadoras, Prototipação, Validação, Desenvolvimento. **Resultados Esperados:** Auxiliar na compreensão dos alunos de turmas iniciais de cursos que envolvam programação a partir de Problemas de Parson.

Palavras-chave: Ensino de programação. Lógica de programação. Problemas de Parson. Educação tecnológica. Ferramentas educacionais

1. Introdução

O aprendizado de programação tem se tornado cada vez mais essencial em um mundo cada vez mais orientado por tecnologias digitais. Seja em áreas diretamente ligadas à computação, como ciência de dados, automação e engenharia, ou em setores tradicionalmente distantes da tecnologia, como saúde e educação, a capacidade de compreender e desenvolver soluções computacionais se mostra uma competência estratégica e amplamente valorizada [Nazir et al., 2020]. No entanto, aprender a programar pode ser uma tarefa desafiadora, principalmente para iniciantes, que frequentemente enfrentam dificuldades relacionadas à lógica de programação, compreensão de sintaxe e estruturação de código.

Nesse cenário, os problemas de Parson [Parsons e Haden, 2006] surgem como uma alternativa pedagógica eficaz para o ensino introdutório de programação. Esse tipo de atividade consiste em apresentar blocos de código desorganizados que devem ser reordenados pelos alunos para formar a solução correta de um problema. Estudos demonstram que essa abordagem reduz a carga cognitiva envolvida na escrita de código do início, sem comprometer a eficácia do aprendizado [Ericson, Margulieux e Rick, 2023]. Além disso, os problemas de Parson promovem o engajamento por meio de uma estrutura semelhante a um quebra-cabeça, podem oferecer feedback imediato e permitir o contato com boas práticas de codificação, contribuindo significativamente para a consolidação dos conceitos fundamentais de programação.

Diante disso, este artigo apresenta o processo de desenvolvimento e concepção de um jogo no estilo puzzle, baseado em problemas de Parson, voltado para o apoio ao processo de ensino-aprendizagem de programação. A ferramenta visa não apenas facilitar o aprendizado de alunos iniciantes, mas também ampliar o repertório de recursos disponíveis para docentes. O presente trabalho está organizado em seis seções, a primeira visa apresentar uma introdução sobre a pesquisa, a segunda seção apresenta a fundamentação teórica sobre Problema de Parsons, na terceira seção, os trabalhos relacionados são apresentados; na quarta, descreve-se a metodologia. O jogo, sua interface e características são apresentados na seção cinco. Por fim, as considerações finais são apresentadas na sexta seção.

2. Problemas de Parson

Os autores do artigo *“Solving Parson's Problems Versus Fixing and Writing Code”*, Ericson, Margulieux e Rick [2023], analisam as dificuldades que os alunos enfrentam ao resolver problemas de Parson, que são desafios de programação nos quais os alunos precisam reorganizar trechos de código embaralhados para formar uma solução correta. Um dos principais desafios apontados é a necessidade de compreender a lógica do programa sem um contexto prévio estruturado, o que exige habilidades de raciocínio lógico e conhecimento de sintaxe da linguagem de programação. Além disso, muitos alunos enfrentam dificuldades para lidar com os possíveis erros conceituais nos fragmentos de código, tornando o processo ainda mais complexo. Essas barreiras são evidentes principalmente para iniciantes, que podem ter dificuldades em identificar a sequência correta das instruções e a relação entre os blocos de código.

Os autores investigaram diferentes abordagens para mitigar essas dificuldades e analisaram como os problemas de Parson podem ser usados para melhorar a aprendizagem da programação. Eles compararam a resolução desses desafios com a correção de erros em códigos e a escrita de programas do zero, observando que os problemas de Parson reduzem a carga cognitiva dos alunos ao apresentar o código sem a necessidade de lidar com a sintaxe das linguagens. Os resultados indicaram que os alunos que resolveram os problemas de

Parson revelaram um desempenho melhor na compreensão dos conceitos de programação em comparação com aqueles que apenas corrigiram códigos com erros. Isso sugere que essa abordagem pode ser uma ferramenta eficaz para ensinar lógica de programação de forma mais acessível, ajudando os alunos a focar na estrutura e na lógica do código antes de se preocuparem com erros na linguagem propriamente dita [Ericson, Margulieux e Rick, 2023].

Reis [2023] denomina *os parson problems* como problemas de Parson em seu texto, e o define como uma técnica de ensino de programação que apresenta trechos de código embaralhados, desafiando os alunos a reorganizá-los na sequência correta para formar um programa, algo como “exercícios de programação baseados em arrastar e soltar blocos”. Ele utiliza essa abordagem no ensino de programação introdutória como uma alternativa para reduzir a carga cognitiva dos alunos, permitindo que eles se concentrem na lógica do código sem se preocupar inicialmente com a sintaxe. O seu estudo indica que essa metodologia melhora os resultados da compreensão dos conceitos fundamentais de programação, especialmente entre iniciantes, ao proporcionar uma experiência de aprendizagem mais guiada e interativa. Além disso, assim como outros autores, seus resultados demonstraram que os alunos que praticaram com problemas de Parson demonstraram maior engajamento e confiança na resolução de problemas computacionais em comparação com aqueles que apenas escrevem o código desde o início.

3. Trabalhos Relacionados

Izidoro [2024], em seu trabalho intitulado “Extensão de uma plataforma web para criação de parsons puzzles de apoio ao ensino de programação para iniciantes”, traz o mesmo conceito, porém agora denominado *Parsons Puzzles*, no qual o autor aplica esse conceito por meio de exercícios que incluem blocos de código propositalmente incorretos, incentivando a análise crítica e a identificação de falhas. Os *Parsons Puzzles* auxiliam na assimilação da lógica de programação ao promover o aprendizado ativo, fornecendo feedback imediato. De acordo com o autor, sua popularidade tem crescido, sendo utilizada tanto em plataformas online quanto em atividades presenciais, demonstrando sua efetividade como ferramenta pedagógica.

Ihantola et al. [2013], em seu trabalho, “*MobileParsons: bringing Parson's Problems to mobile devices*”, apresentam uma ferramenta chamada *MobileParsons*, focada em problemas de Parson para dispositivos móveis. O trabalho ressalta a importância pedagógica dos problemas de Parson no ensino de programação, destacando como a ferramenta auxilia na construção do raciocínio lógico e na compreensão da estrutura dos programas, especialmente por meio de exercícios de ordenação de blocos de código. O estudo reforça o potencial dessa metodologia como ferramenta eficaz de aprendizagem ativa, promovendo o desenvolvimento do pensamento computacional de forma interativa. No entanto, diferente do ParsonFlow, os dois trabalhos apresentados limitam-se no quesito portabilidade, não permitindo o seu uso em navegadores e diferentes sistemas. Consideramos que o uso do jogo no computador com uma interface amigável e que ofereça suporte adequado pode potencializar o processo de ensino e aprendizagem.

4. Metodologia

A metodologia adotada neste trabalho apresenta o processo de concepção da ferramenta e sua implementação.

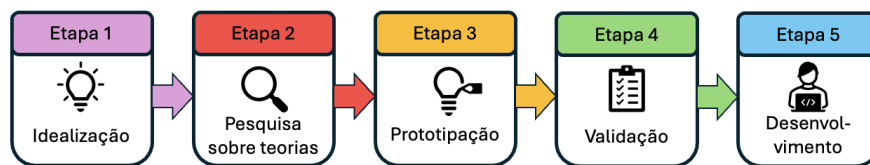


Figura 1. Etapas Metodológicas

4.1 Desenvolvimento do Jogo

Na primeira etapa, foi idealizada a ferramenta, com a definição dos objetivos e a mecânica principal: puzzle. Um jogo puzzle pode ser entendido como um jogo que exige do jogador habilidades de análise do cenário, abstração e pensamento lógico para resolver o problema proposto. A regra central estipulada foi norteadora pela própria dinâmica dos problemas de Parson: a organização de linhas de código para a resolução de um problema específico. Na etapa seguinte foram realizadas pesquisas sobre o caráter pedagógico do Problema de Parson e em qual linha teórica o jogo poderia seguir. Esta etapa contribuiu para a idealização das mecânicas do jogo. Assim, utilizou-se a técnica de prototipação para a validação da ideia. O protótipo pode ser visto na Figura 1.

Após a etapa de validação da ideia, iniciou-se o desenvolvimento da ferramenta e a definição das regras do jogo. O jogo foi implementado com tecnologia web utilizando a linguagem JavaScript, HTML e CSS, em conjunto com o framework React. A interface foi projetada para ser intuitiva e responsiva, visando promover a acessibilidade em diferentes dispositivos, como computadores, tablets e smartphones. Abaixo segue uma tabela demonstrativa sobre as etapas de desenvolvimento e seu status atual. O jogo possui 2 integrantes em sua equipe de desenvolvimento, um homem e uma mulher, ambos alunos do curso de licenciatura em computação do IFBA Campus Jacobina, em seu último ano de curso, os quais dividem as etapas de desenvolvimento igualmente, além de contar com o auxílio de um dos professores do campus.

5. ParsonFlow

Esta seção apresenta o jogo ParsonFlow, sua interface, suas características de jogo segundo Juul [2003] e aspectos pedagógicos segundo a Teoria da Carga Cognitiva e o Construcionismo. Na Figura 2 é possível visualizar, na parte superior, a indicação dos pontos. Na parte central, encontram-se linhas de código desorganizadas. Ao seu lado, um campo que permite arrastar as listas para organizá-las. Do lado oposto, o jogador tem acesso a informações do nível atual e próximos níveis, assim como à sua progressão.

A mecânica do jogo inclui uma lista progressiva de exercícios com níveis de dificuldade crescentes, uma barra de progresso para acompanhamento do desempenho e a liberação gradual das atividades, condicionada à resolução correta das anteriores. Baseado no que propõe Juul [2003], o jogo, além das regras, possui outras características, como o resultado variável – dependendo do desempenho do jogador; esforço do jogador – o jogador precisa se esforçar para realizar a tarefa correta em tempo hábil; consequências negativas – caso o jogador demore para resolver ou execute códigos incorretamente; e valorização dos resultados – pontuação com base no tempo e tentativas para gerar um ranking.

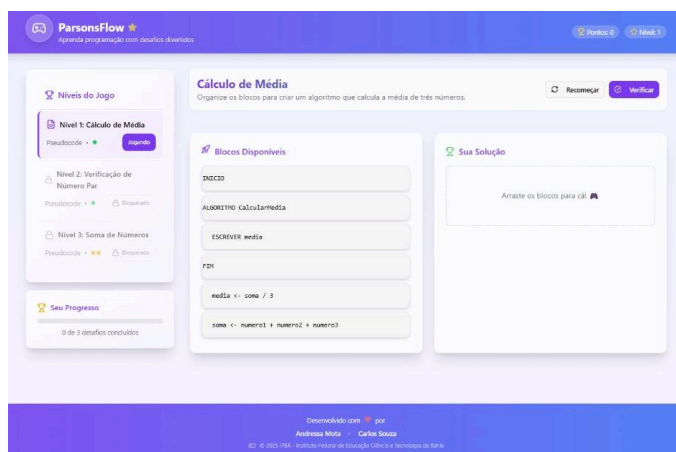


Figura 2. Tela do protótipo do jogo

Do ponto de vista pedagógico, elegeram-se dois paradigmas norteadores. Em primeiro lugar, o construcionismo idealizado por Seymour Papert. O construcionismo caracteriza-se pela construção do conhecimento pelos alunos, com foco na criação de algo, neste caso, a criação de soluções para os problemas propostos. Ao contrário da visão dominante da época, que tratava o computador como uma máquina de ensinar baseada nas respostas dos alunos, Papert propõe um uso mais criativo e ativo da tecnologia. Para ele, o computador deveria servir como ferramenta que favorece a construção do conhecimento, criando contextos mais ricos e apropriados para a aprendizagem [Papert, 1986].

Outro paradigma que norteou a construção do jogo descrito neste trabalho foi a teoria da carga cognitiva proposta por John Sweller na década de 1980. Esta teoria considera que a memória de trabalho processa novas informações e, em seguida, as adiciona à representação do conhecimento (esquemas) que existem na memória de longo prazo. No entanto, essa memória de trabalho é limitada, e essa limitação deve ser considerada no processo de ensino-aprendizagem. Este trabalho leva em consideração essa teoria, ao reduzir a carga cognitiva sobre a sintaxe correta e ao estabelecer um conjunto mínimo de instruções a serem utilizadas, em oposição à criação do código desde o início [Sweller, 1988].

6. Conclusão

Este trabalho apresentou o processo de desenvolvimento e as características do jogo puzzle FlowParson para o auxílio no processo de ensino e aprendizagem em programação. Como resultado, percebe-se que a estrutura dos problemas de Parson é adequada para a utilização em jogos, configurando-se como um puzzle. Em trabalhos futuros, o jogo será aplicado em turmas do ensino médio integrado ao técnico em informática, em turmas iniciais do IFBA – campus Jacobina, sendo utilizado pelos professores das disciplinas de programação como recurso pedagógico complementar. A proposta é que o FlowParson atue como uma ferramenta lúdica e interativa para apoiar o desenvolvimento do raciocínio lógico, a compreensão da estrutura de algoritmos e a prática de montagem de código, aspectos fundamentais no processo de aprendizagem de programação.

Essa integração ao contexto escolar permitirá avaliar não apenas a eficácia da ferramenta no ensino técnico, mas também identificar melhorias para sua aplicação em diferentes cenários educacionais.

Referências

- Ericson, Barbara J.; Margulieux, Lauren E.; Rick, Jochen. (2023) “Solving Parson's Problems Versus Fixing and Writing Code”. *ACM Transactions on Computing Education* , v. 23, n. 4, p. 1-28.
- Ihantola, Petri et al. (2013) “MobileParsons: bringing Parson's Problems to mobile devices.” In: *Proceedings of the 13th Annual Conference on Information Technology Education*. Calgary, Alberta, Canada: ACM. p. 161–166.
- Izidoro, Ramón Godoy. (2024) “Extensão de uma plataforma web para criação de Parsons Puzzles de apoio ao ensino de programação para iniciantes”. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Federal de Santa Maria, Santa Maria, RS.
- Juul, Jesper. (2003) “The game, the player, the world: Looking for a heart of gameness”. In: *DIGRA CONFERENCE: LEVEL UP, Utrecht. Proceedings...* [S.l.: s.n.].
- Parsons, D.; Haden, P. (2006) “Parson's programming puzzles: A fun and effective learning tool for first programming courses”. In: *Proceedings of the 8th Australasian Conference on Computing Education*. Hobart: Australian Computer Society. p. 157–163.
- Papert, S. (1986). *LOGO: Computadores e Educação*. São Paulo: Brasiliense.
- Reis, Anilton Carlos de Lima. (2023) “Uso de problemas de Parson no ensino de programação introdutória”. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação) – Universidade Federal do Amazonas.
- Sweller, John. (1988) “Cognitive load during problem solving: Effects on learning”. *Cognitive Science*, v. 12, n. 2, p. 257-285.