

Desenvolvimento e Avaliação do Jogo Sérioo Educacional Coding Hunter: um ambiente para prática de pensamento computacional

Breno Alves Silva¹, Richarlyson Alves D'Emery¹

Unid. Acad. de Serra Talhada (UAST) – Univ. Federal Rural de Pernambuco (UFRPE)
Caixa Postal 063 – 56.909-535 – Serra Talhada – PE – Brasil

brenoalvesesilva@gmail.com, richarlyson.demery@ufrpe.br

Abstract. *Despite the popularization of technology, there is the problem of training students in the areas of Information and Communication Technologies, causing a shortage of professionals for the job market. However, these students are digital natives and serious educational digital games have proven to be an important teaching too, as long as these games undergo evaluations that show their effectiveness in the teaching and learning process (TLP). This paper presents the development and evaluation of "Coding Hunter", a serious game to assist the Computational Thinking TLP. ENgAGED is used and evaluation by System Usability Scale and verification of hypotheses by statistical inference attest to the quality of the game.*

Resumo. *Apesar da popularização da tecnologia, observa-se o problema de formar alunos nas áreas de Tecnologias da Informação e Comunicação, causando déficit de profissionais para o mercado de trabalho. Entretanto, esses alunos são nativos digitais e jogos digitais sérios educacionais têm se mostrado importante ferramenta de ensino, desde que esses jogos passem por avaliações que apontem a efetividade no processo de ensino e aprendizagem (PEA). Este artigo apresenta o desenvolvimento e avaliação do "Coding Hunter", um jogo sério para auxiliar o PEA de Pensamento Computacional. É utilizado ENgAGED e avaliação por System Usability Scale e verificação de hipóteses por inferência estatística atestam a qualidade do jogo.*

1. Introdução

Segundo pesquisas realizadas pelo Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep), entre o período de 2012 e 2022 [Brasil 2023], na área de Computação e Tecnologias da Informação e Comunicação (TIC) referente a cursos de nível superior em universidades públicas e particulares, bem como entre cursos à distância e presenciais, existe relevante diferença entre os números de estudantes ingressantes, concluintes, desvinculados e com o curso trancado. Por exemplo, na área de Computação e TIC em 2012 foram 142.380 ingressantes, mas em 2017 apenas 41.918 se formaram, uma taxa de 29,44% dos ingressos em 2012 (dado que cursos que Computação duram 5 anos em média); taxa semelhante se observa nos anos seguintes, como, por exemplo, dos 150.745 ingressos em 2017, tem-se apenas 61.760 egressos em 2022, taxa de 41% .

O cenário de baixa relação entre o número de formados e ingressantes eleva a demanda no mercado de trabalho em TIC. O Relatório de Inteligência e Informação de 2019 da Associação das Empresas de Tecnologia da Informação e Comunicação e de

Tecnologias Digitais (Brasscom) já apontava alta demanda por profissionais de TIC, com uma necessidade média de quase 70 mil profissionais anuais entre 2019 e 2024 [Brasscom 2019]; e para o período de 2021 a 2025 há uma escalada com média de 159 mil ao ano, com 206.940 mil apenas em 2025 [Brasscom 2021]. Se comparada a média atual de concluintes ao ano, há enorme déficit de profissionais com perfil tecnológico.

Nos últimos anos, a rápida evolução da tecnologia tem transformado significativamente a forma como as pessoas interagem com a informação e adquirem conhecimento. Júnior e Melo (2021) destacam que o surgimento das tecnologias digitais alterou o contexto pedagógico e educacional no Brasil; e novas tecnologias propiciam novo ambiente acadêmico em que alunos têm autonomia no processo de aprendizagem. Oliveira et al. (2015) apontam quanto mais as novas TIC são elemento do cotidiano devem ser incorporadas no processo escolar de aprendizado.

Isso impacta diretamente na vida de ingressantes em cursos de Computação, os quais pertencem a um ambiente em que a tecnologia desempenha importante papel. Esses enfrentam dificuldade relacionada a aspectos de matemática-lógica, como na compreensão de algoritmos, linguagens de programação e raciocínio lógico. Falckembach e Araújo (2013) destacam que, em cursos de graduação ou técnicos com disciplinas de Algoritmo e Programação em sua grade curricular, há dificuldades em compreender os princípios essenciais da lógica de programação. Diante dessa complexidade na assimilação de conceitos lógicos, ainda no início do curso, existe a possibilidade de estudante perder o entusiasmo, podendo levar a desistência do curso que está matriculado [Hoed 2016, Souza et al. 2015].

Um dos fatores é a dificuldade em compreender conceitos básicos de disciplinas voltadas à resolução de problemas com programação e raciocínio lógico. Arimoto e Oliveira (2019) corroboram esse fato a partir da entrevista de 248 estudantes, em que 247 apontaram dificuldade na aprendizagem de programação, lógica de programação e até em definir e usar estruturas mais complexas, devido à complexidade natural destes assuntos. Para alguns, a maior dificuldade não é entender ou aprender uma linguagem de programação em si, mas sim em conseguir entender a lógica de programação, o que leva à desmotivação que contribui com a desistência do curso e ao aumento da evasão.

Esses relatos evidenciam a discussão entre a relação de Pensamento Computacional (PC) e Programação, na qual Wing (2006) aponta que PC não se limita a programação. A partir do instrumento de avaliação de PC do concurso Bebras, aplicado a 105 estudantes brasileiros de cursos de Computação entre ingressantes e outros que já estudaram programação e linguagens, Mélo et al. (2023) evidenciam estudo em que solucionar problemas independem de saber programação. Fato também corroborado por Mooney e Lockwood (2020) entre estudantes de Computação da Irlanda.

Não há consenso universal sobre definição e habilidades de PC [Kalelioglu, Gülbahar e Kukul 2016]. Román-González, Moreno-León e Robles (2019) apontam a divergência sobre como e quando o PC deve aparecer em currículos educacionais. Gouws, Bradshaw e Wentworth (2013) ressaltam que, apesar de o PC ser associado à maneira de pensar típica da Computação, é habilidade que deveria ser exercitada por todos e auxiliar o processo de ensino e aprendizagem (PEA) de diferentes áreas. Essa discussão não é restrita do ensino superior, também abrange a necessidade de introduzir o PC desde a educação básica [Mattos et al. 2018, Schlögl et al. 2017, Wing 2016].

Portanto, é necessário criar soluções para estimular a prática de PC para que mais pessoas desenvolvam esse tipo de habilidade e, quiçá, se mantenham nos cursos sem a desistência precoce, especialmente em cursos de TIC, e sejam absorvidos no mercado de trabalho. Sendo assim, tem-se o seguinte questionamento: “como colaborar com soluções metodológicas que auxiliem o desenvolvimento de habilidades de PC?”. Para Pinho (2016), jogos sérios são alternativas promissoras para o PEA de PC.

Considerando os desafios na busca por abordagens metodológicas que incentivem a prática de PC, esta pesquisa tem por objetivo desenvolver um jogo sério baseado em blocos para auxiliar estudantes, de educação básica e ingressantes de cursos da área da Computação, a desenvolver habilidades de PC.

O artigo estrutura-se da seguinte forma: a Seção 2 aborda a fundamentação teórica da pesquisa; na Seção 3 há trabalhos relacionados; na Seção 4 tem-se o método com apresentação das etapas do modelo seguido; na Seção 5 discutem-se os resultados; na Seção 6 estão as conclusões; e, por fim, os agradecimentos seguidos das referências.

2. Referencial Teórico

2.1. Jogos Digitais Educacionais e Pensamento Computacional

Para Barreto et al. (2017), a incorporação de tecnologias inovadoras no ambiente escolar que potencializam o PEA está alinhada com os interesses dos estudantes, considerando o quão presente é a tecnologia na sociedade atual. Para os autores, essa abordagem tem o potencial de facilitar a compreensão e a assimilação de conteúdos abstratos.

No contexto do ensino, segundo Medeiros, Falcão e Ramalho (2020), um grande desafio enfrentado por professores em cursos superiores voltados para programação está em manter a motivação dos estudantes, sendo este papel fundamental para o professor. Esta dificuldade está relacionada a um déficit de métodos e ferramentas para o ensino de disciplinas introdutórias de programação que sejam suficientemente efetivas.

Wouters e Oostendorp (2017) apontam que ambientes virtuais educacionais são frequentemente abordados como jogos sérios e, para Dörner et al. (2016), um jogo sério é um jogo digital que busca entreter e atingir pelo menos um segundo objetivo, como, por exemplo, o ensino. Assim, Neto et al. (2013) apontam que a inserção dos jogos sérios na educação traz aspectos como atividade voluntária, aprendizagem em potencial, inovação, motivação e superação, o que facilita o PEA.

O aprendizado de programação, em especial, começa com o entendimento de conteúdo comumente visto no ensino superior, a partir de conceitos que se fundamentam no PC. Para Brennan e Resnick (2012), a aplicação de abordagens baseadas em PC contribui para o aprendizado de programação, especialmente para os jovens, ao empregar ferramentas que os motivem.

Diversas iniciativas presentes na literatura [Macena, Pires e Melo 2022, Silva, Rivero e Santos 2021, Schmitt, Bernardi e Zanki 2019] exploram a prática da programação e de PC de forma lúdica com jogos sérios. No entanto, Aarseth (2003) destaca a necessidade de entender as características dos "bons jogos" e adotar abordagens para uma avaliação adequada antes da distribuição.

2.2. Avaliação de jogos educacionais digitais

São diversas as abordagens presentes na literatura, algumas que se baseiam em verificações de princípios de *design*, como o estudo de Nielsen (2012), no qual aponta que a usabilidade de um sistema está diretamente vinculada à sua sobrevivência. Um sistema que apresenta falhas ou dificuldades acaba por afastar os usuários. Uma falha de usabilidade pode resultar em uma experiência negativa, levando o usuário, conseqüentemente, à desistência de utilizar o sistema devido à frustração.

Segundo Revythi e Tselios (2017), o *System Usability Scale* (SUS) [Brooke 1996] é um dos questionários mais válidos e confiáveis para medir a usabilidade de um sistema por usuários, ainda que utilizadas pequenas amostras. O SUS é tecnologicamente agnóstico e pode ser utilizado para avaliar diversos tipos de produtos, sejam eles hardware ou *software*, inclusive educacionais [Kortum e Bangor 2013].

O questionário SUS possui 10 questões relacionadas a princípios de *design*, alternadas entre afirmativas e negativas, que devem ser respondidas numa escala Likert de 5 pontos que vai de “discordo completamente” à “concordo completamente”. A pontuação de um *software* varia de 0 e 100, calculada da seguinte forma: (i) para questões ímpares, subtrai-se 1 da pontuação atribuída; (ii) para questões pares, subtrai-se a resposta de 5; (iii) soma-se os valores das dez perguntas, multiplica-se por 2,5 para obter a pontuação final de um usuário; e (iv) a média das pontuações de todos os usuários resulta na pontuação final do *software*.

Segundo Bangor, Kortum e Miller (2009), o *software* é classificado como: pior imaginável ($0 \leq \text{sus} \leq 25$), ruim ($25 < \text{sus} \leq 39$), regular ($39 < \text{sus} \leq 52$), bom ($52 < \text{sus} \leq 73,5$), excelente ($73,5 < \text{sus} \leq 85$) e melhor imaginável ($85 < \text{sus} \leq 100$). Um *software* é considerado aceito quando a pontuação SUS é de 70 pontos e tido como bom quando a pontuação é de 73,5 pontos [Brooke 2013]. Se um *software* possui pontuação abaixo desses valores indicam a necessidade de melhorias nos princípios avaliados nas questões do questionário de baixa avaliação Likert.

3. Trabalhos Relacionados

Para evidenciar trabalhos relacionados, utilizou-se de busca automática por Google Scholar¹ para identificar estudos primários atuais, últimos 5 anos, através da *string* de busca: (“Jogo*” AND (“Educação*” OR “Sério*”)) AND (“Ensino de Programação” OR “Pensamento Computacional”). A escolha do motor de busca é por este utilizar mecanismo que indexa estudos publicados em várias bases de dados da Computação e Educação. Dessa forma, tem-se uma busca direta, visto que apenas os resultados de uma busca são analisados. Nesta Seção apresentam-se estudos tidos como relacionados.

Silva, Rivero e Santos (2021) apresentam ProgramSE, que tem a finalidade de apoiar o ensino de programação e o exercício de PC para estudantes ingressantes no ensino superior. O jogo é baseado no estilo *Escape The Room* desenvolvido no *Construct 3*, em que o jogador está preso em um local e precisa escapar. Em seu enredo, são apresentados problemas e desafios que devem ser resolvidos, como, por exemplo, montar pseudocódigos utilizando blocos para executar uma ação e assim resolver o problema. A avaliação se dá pela interação piloto de um instrutor e um aluno com registro por MEEGA+, exceto de interação social por ser *singleplayer*. Após 51 alunos

¹ <https://scholar.google.com/>

avaliarem, teve como pontos fortes a fundamentação, jogabilidade e *design*; mas que precisa ser menos monótono e adicionar novidades.

Macena, Pires e Melo (2022) apresentaram uma proposta de gamificação de temas introdutórios de programação a partir de um jogo educativo chamado Hello Food. A *gameplay* se dá a partir da resolução de problemas dentro da temática de um restaurante, exibindo ao jogador um pseudocódigo que ilustra uma receita de um prato, para que o jogador possa recriar a receita dentro do jogo através da interação com objetos estáticos baseando-se no pseudocódigo. Em pesquisa quantitativa, por SUS e com metodologia GameFlow, constataram bom nível de aceitação e, apesar de um jogo intuitivo, o Controle deve ser melhorado por precisar de tutoriais para seu uso.

O trabalho de Schmitt, Bernardi e Zanki (2019) apresenta o Super Mario Logic, um jogo educacional focado no exercício de PC através de programação em blocos. É baseado na série de jogos Super Mário World, porém com *gameplay* voltada ao ensino, em que o jogador controla o personagem através de blocos que formam uma linha de comandos que o personagem deve seguir. Segue a metodologia Interad para seu desenvolvimento e avaliação por GameFlow, esta última apontando que o jogo tem potencial para prática de PC, mas que deve melhorar a sobrecarga nos jogadores devido ao fluxo de dificuldade dos desafios bem como a imersão; e a quantidade de estímulos é avaliada de maneira mediana.

4. Método

Esta é uma pesquisa com abordagem híbrida (quali-quanti), permitindo uma compreensão mais abrangente do objeto de estudo, ao combinar análises aprofundadas de aspectos qualitativos com a coleta e análise de dados quantitativos. É de natureza aplicada, em torno de problemas presentes nas atividades do público-alvo determinado, é focada na elaboração de diagnósticos, identificação de problemas e busca de soluções.

O modelo ENgAGED [Battistella e von Wangenheim 2016], Figura 1, é adotado com a finalidade de guiar a criação e avaliação do jogo.

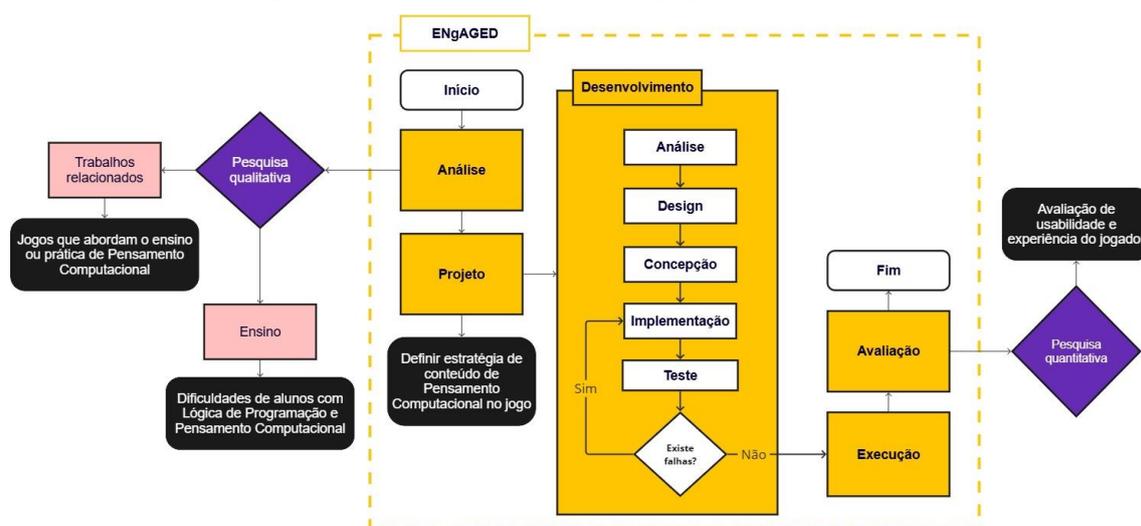


Figura 1. Fluxograma das atividades baseado no modelo ENgAGED.

Fonte: Adaptado de Battistella e von Wangenheim (2016)

Na etapa de “Análise”, destaca-se a pesquisa qualitativa que identificou trabalhos relacionados que abordassem jogos para o ensino ou prática de PC, bem como o mapeamento e coleta de dados que evidenciassem dificuldades atreladas ao aprendizado de programação e interesses de estudantes em cursos da área de TIC.

Em “Projeto” definiu-se a Unidade Instrucional (UI) e sua forma de validação, dos quais há a definição dos desafios de situação-problema que explore PC em um jogo sério através de programação por blocos e validação a partir de mudança em relação à sugerida pelo ENgAGED, respectivamente. Substitui-se a abordagem de foco exclusivo na estatística descritiva através do questionário MEEGA+ [Petri, von Wangenheim e Borgatto 2019] para um novo questionário, apresentado na etapa de “Avaliação”, cujas respostas serão utilizadas em testes de verificação de qualidade.

A etapa de “Desenvolvimento” é dividida em cinco fases: análise, concepção, *design*, implementação e teste. Na fase de “Análise”, distribui-se o conteúdo no ambiente do jogo para que na fase de “Concepção” fossem formados os elementos do jogo, como narrativa, regras e mecânicas. O “*Design*” utiliza técnicas de desenvolvimento de jogos com escolha gráfica de estilo 2D através de *sprites*, *tiles* e *tileMaps*; e a plataforma na qual o jogo será executado. Para este último tem-se que o *software* é *desktop* multiplataformas desenvolvido em Java, organizado por padrão arquitetural *Model-View-Controller* e faz uso de princípios básicos de *design*. “Implementação” e “Teste” são interligadas: a codificação retrata a estratégia em que a programação é construída em forma de blocos e interage diretamente com o cenário, permitindo ao jogador montar seus próprios códigos para resolver situações-problema. A cada incremento, testes internos foram realizados pelos desenvolvedores para descoberta de *bugs*, antes que fosse disponibilizado para avaliações.

Na “Execução” tem-se a distribuição do jogo em arquivo *.jar* de maneira que o usuário consiga adicionar no computador a versão correta do *software*, respeitando características de compatibilidade de arquitetura e sistema operacional.

Por fim, na “Avaliação” utilizou-se distintos métodos, diferentemente do sugerido em ENgAGED, visando atestar a qualidade do jogo por diversas abordagens. A coleta de dados foi realizada por método prospectivo em questionário de perguntas abertas e fechadas, distribuído em formulário da plataforma Google Forms em grupos atrelados a cursos de TIC. Portanto, a escolha dos indivíduos deu-se ao acaso, uma vez que não houve controle da distribuição dos convites, implicando em uma amostragem não-probabilística e por conveniência, podendo participar dessa avaliação alunos, professores, ergonomistas, engenheiros de *software* ou de usabilidade, assim como outros profissionais com habilidades em Computação.

Na avaliação um usuário inspeciona a usabilidade, sendo este um tipo de avaliação de baixo custo e alto benefício em que a interface é examinada verificando as propriedades de usabilidade definidas [Ferreira et al. 2014]. Para inspeção do percurso cognitivo, em que é avaliada a aprendizagem por exploração, antes de responder as questões relacionadas à avaliação do *software*, no formulário foi previamente proposta a realização de algumas tarefas para, por exemplo, fazer o *download* do jogo, assistir vídeo tutorial, realizar comandos e cumprir desafios. A avaliação é desconsiderada em caso de eventuais problemas de execução de tarefas reportados por usuário e que possa comprometer a avaliação.

O formulário é composto de cinco seções: (i) Apresentação e descrição da pesquisa; (ii) Caracterização de indivíduo (Tabela 1); (iii) Tarefas; (iv) Avaliação de usabilidade de *design* (Tabela 2); Avaliação de usabilidade pedagógica (Tabela 3); e (v) Considerações finais (contendo questão para indicação de nota de avaliação entre 0 e 100 a partir da experiência geral (EG) do usuário com o *software* e campo de texto para sugestões e comentários). As questões de usabilidade foram adaptadas, para o contexto da avaliação, sendo elas: Q7 a Q13 de Anjos e Moura (2005), Q14 a Q19 de An et al. (2013), Q20 a Q23 de Almeida et al. (2018), Q24 e Q25 de Fantin (2017) e Q26 de Ferreira et al. (2014). Entende-se que as questões já foram previamente validadas, eliminando a necessidade de realização de pré-testes.

Tabela 1. Questionário para caracterização de indivíduo

Nº	Questões
Q1	Qual sua faixa etária?
Q2	Qual sua escolaridade?
Q3	Qual seu sexo?
Q4	Qual sua área de atuação/estudo?
Q5	Qual seu nível de conhecimento sobre pensamento computacional?
Q6	Com que frequência você costuma utilizar softwares para auxiliar na prática de estudos de pensamento computacional?

Tabela 2. Questionário para avaliação de usabilidade de *design*

Nº	Questões	Heurística
Q7	As alterações de estado na seleção de comandos são claras.	Visibilidade do status do sistema
Q8	O jogo possui ações e opções visíveis sempre que apropriado para o uso.	
Q9	É fácil de operar e controlar o personagem e suas ações.	Operacionalidade
Q10	É fácil alterar os comandos inseridos.	
Q11	O jogo possui funções “Desfazer” facilmente acessíveis.	
Q12	É fácil perceber que botões têm funções diferentes ou semelhantes.	Consistência
Q13	As palavras, situações ou ações disponíveis não geram dúvidas de entendimento ou interpretação.	
Q14	O nível de concentração exigido está de acordo com o público do jogo (pessoas interessadas em programação e pensamento computacional).	Atratividade
Q15	A interface não contém informações irrelevantes.	
Q16	O jogo oferece feedbacks construtivos.	Feedback

Tabela 3. Questionário para avaliação pedagógica

Nº	Questões
Q17	Apesar de necessitar raciocinar para concluir os desafios, não senti grandes dificuldades.
Q18	O jogo promove a criatividade (existem vários caminhos/respostas/soluções).
Q19	Eu alcancei rapidamente os objetivos do jogo.
Q20	O objetivo do jogo está claro.
Q21	O jogo está de acordo com a proposta de um jogo para exercício de pensamento computacional.
Q22	O conteúdo é coerente e contextualizado com a área proposta.
Q23	O conteúdo do jogo está relacionado com coisas que eu já sabia.
Q24	É fácil localizar pontos de interesse no jogo.
Q25	O jogo lhe proporcionou tempo para refletir sobre suas ações e quais decisões tomar.
Q26	O jogo facilita a prática do pensamento computacional.

Realizou-se a verificação de possível existência de *outliers* por gráfico *boxplot* com objetivo de remover dados discrepantes. Também há exclusão de dados a partir de respostas oriundas de variáveis que possam comprometer a qualidade da avaliação, como, por exemplo, dados de usuário que aponte nenhum conhecimento em PC.

A consistência interna das respostas para as questões de avaliação de usabilidade de *design* e pedagógica é dada por coeficiente (αc) de Alpha de Cronbach [Cronbach 1951], calculado a partir da variância dos itens individuais e da variância da soma dos itens de cada avaliador através da equação Eq. 1:

$$\alpha c = \left(\frac{k}{k-1} \right) x \left(1 - \frac{\sum_{i=1}^k s_i^2}{s_t^2} \right) \quad (\text{Eq. 1})$$

Onde: k corresponde ao número de itens do questionário; s_i^2 corresponde a variância de cada item; s_t^2 corresponde a variância total do questionário, determinada como a soma de todas as variâncias.

Essa análise proporciona conhecimento sobre a confiabilidade do instrumento de medida, garantindo a qualidade dos dados obtidos e fortalecendo a validade dos resultados [Hora, Monteiro e Arica 2010]. Segundo George e Mallery (2003), o valor αc pode variar entre 0 e 1, com a seguinte classificação de qualidade: excelente ($\alpha c > 0,9$), bom ($\alpha c > 0,8$), aceitável ($\alpha c > 0,7$), questionável ($\alpha c > 0,6$), ruim ($\alpha c > 0,5$); e inaceitável ($\alpha c < 0,5$).

Em seguida, tem-se: (i) análise descritiva dos dados; (ii) avaliação da qualidade de *software* por SUS; e (iii) inferência estatística a partir de testes de hipóteses.

Na avaliação adotada, as perguntas do questionário de usabilidade são formuladas de maneira positivo-afirmativa, por os pesquisadores entender que assimilar concordância e discordância seja mais natural. Sendo assim, antes de continuar a avaliação, as respostas pares do questionário foram ajustadas, de maneira que: resposta com valor 5 é convertida em 1; 4 em 2; 3 em 3; 2 em 4; e 1 em 5, refletindo a opinião inversa caso o questionamento estivesse escrito de maneira positivo-negativa.

Para a análise por inferência estatística, têm-se testes de hipóteses definidos a partir de duas situações distintas. Na primeira situação (S1) é analisado se há diferença significativa entre as médias das avaliações a partir de: (i) hipótese nula (H0-S1) – as médias das notas para EG e SUS são consideradas estatisticamente iguais; e (ii) hipótese alternativa (H1-S1) – as médias das notas para EG e SUS diferem estatisticamente.

Na segunda situação (S2) verifica-se a contribuição do jogo proposto. Para isso, é observado se há relação de dependência entre os níveis de aceitação em relação à avaliação do jogo e, se confirmada essa dependência, verificam-se hipóteses definidas quanto à contribuição do jogo no PEA de PC: (i) hipótese nula (H0-S2) – o jogo não contribui significativamente; e (ii) hipótese alternativa (H1-S2) – o jogo contribui.

Para responder a S1 utiliza-se a ANOVA a partir do Teste F para verificação das hipóteses [Vieira 2006]; enquanto para S2 tem-se o Qui-Quadrado (χ^2) [Plackett 1983 apud [Levin e Fox 2004]].

Destaca-se que, conforme Siegel (1975), a aplicação do teste de χ^2 é restrita a tabelas que possuam várias linhas e colunas, devendo o pesquisador observar que o número de frequências esperadas com valores inferiores a 5 seja inferior a 20% do total das frequências esperadas, e que nenhum desses valores seja inferior a 1. Se essas

condições não forem atendidas, é recomendável que o pesquisador, ao conduzir o teste, proceda com o rearranjo dos dados. Isso implica combinar categorias nas tabelas de contingência a partir das variáveis, de modo a aumentar as frequências esperadas.

5. Resultados e Discussões

Como resultado tem-se o jogo sério Coding Hunter² (Figura 2), um *software* lúdico baseado em programação em blocos para auxiliar o PEA que possibilite desenvolver habilidades de PC.



Figura 2. Tela do jogo sério Coding Hunter.

O objetivo do jogo é montar uma sequência de comandos que leve o personagem do lobo guará caçar sua presa (coelho). Estratégias de gerenciamento de recurso devem ser pensadas, ou seja, o lobo cansa e/ou tem fome completando a temática de sobrevivência do jogo. Caçar no jogo reflete um algoritmo a ser criado pelo jogador. A sequência lógica movimenta o lobo através dos comandos avançar ou virar (esquerda ou direita) em um plano 2D (com recurso de grade para identificar a quantidade de movimentos) para alcançar o coelho e assim comê-lo (comando comer). Sequências erradas podem ser removidas parcialmente (comando reverter) ou totalmente (comando reverter tudo). O lobo cansa ao passar o tempo e deve-se utilizar o comando de descansar para recuperar seu fôlego, porém ao fazê-lo a sua fome aumenta. Sempre que o jogador alcança o objetivo, uma nova presa aparece em um lugar aleatório do mapa, fazendo o jogo durar até o jogador cometer algum erro ou o lobo estar saciado.

No jogo também é possível identificar características de ambiente de programação como comandos e console de execução. Também há recursos de gamificação de jogos digitais como som e *life* de recursos.

Para avaliação da qualidade do jogo sério Coding Hunter, participaram um total de 22 pessoas (18 do sexo masculino e 4 feminino); de faixa etária: 23% entre 16 e 19 anos, 68% de 20 a 29 anos, e 9% com 30 anos ou mais; escolaridade: 4% possui ensino

² <https://github.com/LEIA-UFRPE/Coding-Hunter>

médio incompleto, 23% ensino médio completo, 55% ensino superior incompleto, 9% superior completo e 9% possui especialização; área de formação/atuação: 73% é de Computação e 27% de outras áreas; e todos possuem algum nível de conhecimento em PC. Todas as avaliações foram consideradas por não apresentarem *outliers* (Figura 3) nas avaliações EG ou SUS, bem como não possuir usuário sem conhecimento em PC.

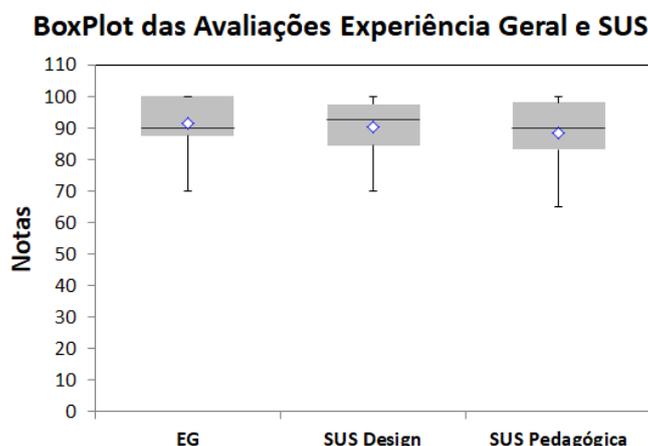


Figura 3. *BoxPlot* das avaliações de Experiência Geral e SUS.

Na Tabela 4 tem-se o resultado da confiabilidade das respostas dos participantes para o questionário de avaliação do jogo. Após constatar a confiabilidade dos dados, a Figura 4 apresenta a distribuição de aceitação, em pontos percentuais, por usabilidade.

Tabela 4. Alfa de Cronbach para os dados da avaliação

Tipo de questão	Nº de questões	Nº de indivíduos	Total de respostas	α	Confiabilidade
Usabilidade de <i>design</i>	10	22	220	0,720	Aceitável
Usabilidade pedagógica	10	22	220	0,731	Aceitável
Todas as questões	20	22	440	0,838	Bom

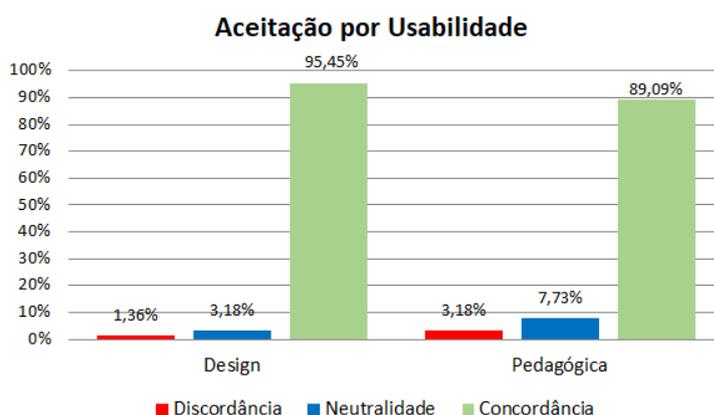


Figura 4. Aceitação de critérios de usabilidades de *design* e pedagógica.

Na Figura 4 vê-se a alta aceitação nos critérios de usabilidade de *design* e pedagógica da avaliação de Coding Hunter, resultando em pontuações SUS de 90,11 e 88,52, respectivamente, o que reflete na qualidade “melhor imaginável” de SUS.

Visando confrontar a avaliação SUS com a percepção dos usuários a partir da EG, as Tabelas 5 e 6 apresentam a análise das hipóteses definidas em S1. Para a análise será definido o nível de significância (α) de 5%, sendo esta a probabilidade de rejeição

da hipótese nula (H0) quando ela é verdadeira, ou seja, $\alpha = 0,05$ indica um risco de 5% de concluir que existe uma diferença quando não há diferença real.

Tabela 5. ANOVA para os valores médios das avaliações de Experiência Geral e SUS *design*

Estatística Descritiva				
Grupo	Contagem	Soma	Média	Variância
EG	22	2010	91,36	98,05
SUS (<i>Design</i>)	22	1982,5	90,11	80,05

ANOVA: fator único						
Fonte da variação	SQ	gl	MQ	F	valor-p	F crítico
Entre grupos	17,19	1	17,19	0,19	0,66	4,07
Dentro dos grupos	3740,06	42	89,05			
Total	3757,24	43				

Tabela 6. ANOVA para os valores médios das avaliações de Experiência Geral e SUS pedagógica

Estatística Descritiva				
Grupo	Contagem	Soma	Média	Variância
EG	22	2010	91,36	98,05
SUS (Pedagógica)	22	1945	88,41	115,21

ANOVA: fator único						
Fonte da variação	SQ	gl	MQ	F	valor-p	F crítico
Entre grupos	96,02	1	96,02	0,90	0,35	4,07
Dentro dos grupos	4478,41	42	106,63			
Total	4574,43	43				

Na comparação entre EG e SUS para as usabilidades *design* e pedagógica, tem-se em ambos os casos valores-p muito superiores a α ($p > 0,05$). Portanto, (H0-S1) não é refutada, ou seja, as médias das avaliações obtidas não diferem significativamente. A análise demonstra a homogeneidade dos resultados que consideram a ótima qualidade da avaliação de Coding Hunter.

Para proceder com as situações por inferência estatística, na Tabela 7 são apresentadas as frequências observadas (f_o), ou seja, as respostas dos questionamentos para a avaliação de usabilidades de *design* e pedagógica, enquanto na Tabela 8 decorre-se a análise estatística do teste de χ^2 .

Tabela 7. Frequências observadas das avaliações de usabilidade

Métricas	Usabilidade		Total
	<i>Design</i>	Pedagógica	
Discordância	3	7	10
Neutralidade	7	17	24
Concordância	210	196	406
Total	220	220	440

Para definir as dependências estatísticas entre os níveis de aceitação dos usuários e as usabilidade, foram realizadas as seguintes combinações da escala Likert, a fim de satisfazer o pressuposto de Siegel (1975) para realização do teste χ^2 , a saber:

discordância com a união das opções “discordo totalmente” e “discordo”; e concordância para “concordo totalmente” e “concordo”.

Tabela 8. Cálculo de χ^2 para as usabilidades

Métricas	f_o	f_e	χ^2_{UD}	χ^2_{UP}
Discordância (Usabilidade de <i>design</i>)	3	5	3,125	3,125
Discordância (Usabilidade pedagógica)	7	5		
Neutralidade (Usabilidade de <i>design</i>)	7	12		
Neutralidade (Usabilidade pedagógica)	17	12		
Concordância (Usabilidade de <i>design</i>)	210	203		
Concordância (Usabilidade pedagógica)	196	203		
χ^2			6,25	
gl			4	
valor-p			0,04	
Legenda: f_o – Frequência observada; f_e – Frequência esperada; χ^2 – Qui-Quadrado; gl – Graus de liberdade.				

Para a análise de inferência estatística da S2, obteve-se $\chi^2_{calculado} = 6,25$. Como $\chi^2_{tabelado} = 5,991$ (gl = 2 e $\alpha = 0,05$) e valor-p = 0,04, então se refuta H0-S2, pois $\chi^2_{calculado} > \chi^2_{tabelado}$. Portanto, a usabilidade oferecida por Coding Hunter fornece contribuição significativa para o PEA de PC (H1-S2), tendo em vista a dependência ($p < 0,05$) entre as variáveis de usabilidade e o nível de aceitação elevado (Figura 4) para as afirmações definidas no questionário de avaliação.

6. Conclusões

Este estudo permitiu apresentar o jogo sério Coding Hunter, contribuindo com uma solução que poderá auxiliar o PEA com o objetivo de desenvolver habilidade de PC. A avaliação do *software* permitiu atestar sua qualidade, destaca-se o fato dos participantes terem avaliado o *design* do jogo positivamente em todos os questionamentos. Aponta-se a possibilidade de outros estudos, relacionados em produzir e avaliar a usabilidade de jogos sérios, a partir desse artigo, em uma perspectiva de substituição de métodos tradicionais de avaliação de jogos em modelos com a finalidade de guiar a criação e avaliação de jogos, como, por exemplo, o EngAGED.

Como ameaça à validade do estudo, aponta-se a seleção dos participantes como aspecto a ser considerado. Mas, acredita-se que esta tenha sido mitigada, a partir da exclusão de eventuais *outliers* e do conhecimento em diferentes níveis de PC. Portanto, o tamanho da amostra de participantes não permite generalizar os resultados de contribuição do *software* em todos os ambientes de ensino, entretanto, é indicativo de sua qualidade e, por isso, pode ser tratado como recomendação de iniciativa a ser utilizada em cursos de TIC que possuam disciplinas introdutórias de PC.

A partir das sugestões, em trabalhos futuros pretende-se criar novos cenários ao jogo de maneira que o personagem possa se deslocar com regras predefinidas, a exemplo de mover apenas na diagonal; e da criação de novos personagens, como, por exemplo, animais e frutas comumente encontrados na alimentação de um lobo guará.

Agradecimentos

Os autores agradecem ao grupo de pesquisa Laboratório de Estudos em Informática (LEIA) da UAST–UFRPE.

Referências

- Aarseth, E. (2003). Playing research: Methodological approaches to game analysis. In: Australia DAC. [S.l.: s.n.].
- Almeida, A., Gomes, T., Leal, V., Gomes, R., & Leal, L. (2018). Indicadores para Avaliação de Software Educacional com base no guia GDSM (Goal Driven Software Measurement). In: *Anais do XXIX Simpósio Brasileiro de Informática na Educação - SBIE*, v. 29, n., p. 21-30.
- An, D., da Silva, C., Ribeiro, D., da Rocha, P., Maltinti, C., Nunes, V. & Fávero, R. (2013). Digita – um Jogo Educativo de Apoio ao Processo de Alfabetização Infantil. In: *Anais do XXIV Simpósio Brasileiro de Informática na Educação*, v. 24. n.1, p. 154-163.
- Anjos, L. A. M. & Moura, H. P. (2005). Um Modelo para Avaliação de Produtos de Software. Centro de Informática - Universidade Federal de Pernambuco (UFPE).
- Arimoto, M. & Oliveira, W. (2019). Dificuldades no Processo de Aprendizagem de Programação de Computadores: um Survey com Estudantes de Cursos da Área de Computação. In *Anais do XXVII Workshop sobre Educação em Computação*, pp. 244-254. Porto Alegre: SBC.
- Bangor, A., Kortum, P. and Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. In: *Journal of Usability Studies (JUS)*. v. 4, issue 3, pp. 114-123.
- Barreto, G. S. N., Xavier, J. L., Santos, J. D. & Mesquita, N. A. da S. (2017). O Processo de Criação de um Software Educacional para o Ensino e aprendizagem de Química. In: *Revista Eletrônica Ludus Scientiae (RELuS)*, v.1, n. 2.
- Battistella, P. E. & von Wangenheim, C. G. (2016). ENgAGED: Um processo de desenvolvimento de jogos para ensinar computação. In: *Anais do XXVII Simpósio Brasileiro de Informática na Educação*. 27, n. 1, p. 380-389.
- Brasil. (2023). Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira-Inep. Censo da Educação Superior, <https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-da-educacao-superior/resultados>.
- Brasscom. (2019). Formação Educacional e Empregabilidade em TIC Achados e Recomendações, <https://brasscom.org.br/wp-content/uploads/2021/10/BRI2-2019-010-P02-Formacao-Educacional-e-Empregabilidade-em-TIC-v83.pdf>.
- Brasscom. (2021). Relatório Setorial 2021 – Macrossetor em TIC, <https://brasscom.org.br/pdfs/relatorio-setorial-de-tic/>
- Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, Vancouver, Canada.

- Brooke, J. (1996). SUS: A quick and dirty usability scale. In: P.W. Jordan, B. Thomas, B. A. Weerdmeester and I. L. McClelland (Eds.), *Usability Evaluation in Industry*, pp. 189-194. London: Taylor & Francis
- Brooke, J. (2013). SUS: A retrospective. In: *Journal of Usability Studies (JUS)*. v. 8, issue 2, February, pp. 29-40.
- Cronbach, J. L. (1951). Coefficient alpha and the internal structure of tests. In: *Psychometrika*. v. 16, n. 3, pp. 297-334.
- Dörner, R., Göbel, S., Effelsberg, W. and Wiemeyer, J. (2016). *Serious Games*. In: *Springer International Publishing*.
- Falckembach, G. A. M. & de Araujo, F. V. (2013). Aprendizagem de algoritmos: dificuldades na resolução de problemas. In: *Anais do Sulcomp*, 2.
- Fantin, K. (2017). Metodologia de Avaliação de Software Educacional. In: *Informática na Educação: teoria e prática*, v. 22, n. 2, Porto Alegre.
- Ferreira, B. M., Rivero, L., Lopes, A., Marques, A. & Conte, T. (2014). UsabiliCity: Um Jogo de Apoio ao Ensino de Propriedades de Usabilidade de Software Através de Analogias. In: *Anais do III CBIE*, p.1273-1282.
- George, D. and Mallery, P. (2003). *SPSS for Windows step by step: A simple guide and reference*. 11.0 update (4th ed.). Boston: Allyn & Bacon. In: Gliem, J. A. and Gliem, R. R. (2003). Calculating, Interpreting, and Reporting Cronbach's Alpha Reliability Coefficient for Likert-Type Scales, Midwest Research-to-Practice Conference in Adult, Continuing, and Community Education, The Ohio State University, Columbus, OH, p. 82-88.
- Gouws, L. A., Bradshaw, K. and Wentworth, P. (2013). Computational Thinking in Educational Activities: an evaluation of the educational game Light-Bot. In: *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, pp. 10-15. ACM.
- Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação. Dissertação (Mestrado - Mestrado Profissional em Computação Aplicada) - Universidade de Brasília.
- Hora, H. R. M., Monteiro, G. T. R. & Arica, J. (2010). Confiabilidade em Questionários para Qualidade: Um Estudo com o Coeficiente Alfa de Cronbach. In: *Produto & Produção*, v. 11, n. 2, p. 85-103, jun.
- Júnior, I. L. V. & Melo, J. C. (2021). Utilizando as tecnologias na educação: possibilidades e necessidades nos dias atuais. In *Brazilian Journal of Development*, v. 7, n. 4, p. 34301-34313.
- Kalelioglu, F., Gülbahar, Y. and Kukul, V. (2016). A Framework for Computational Thinking Based on a Systematic Research Review. In: *Baltic Journal of Modern Computing*, v.4 n. 3, p. 583-596.
- Kortum, P. T. and Bangor A. (2013). Usability ratings for everyday products measured with the System Usability Scale. In: *International Journal of Human-Computer Interaction*, v. 29, n 2. p. 67-76.

- Levin, J. & Fox, J. A. (2004). “Estatística para Ciências Humanas”, São Paulo: Prentice Hall, 9 ed.
- Macena, J., Pires, F. & Melo, R. (2022). Hello Food: uma jornada de aprendizagem lúdica em algoritmos, programação e Pensamento Computacional, In: *Anais do XXXIII Simpósio Brasileiro de Informática na Educação (SBIE)*, v. 33, p. 561-572, Manaus: SBC.
- Mattos, M., Araújo, L., Silveira, H., Schlögl, L., Giovanella, G., Santos, B., Fronza, L., Zucco, F., Hein, N., Oliveira, G., Cunha, K. & Sartori, A. (2018). Uma pesquisa-ação sobre o desenvolvimento do pensamento computacional com crianças. In *Anais do XXIV Workshop de Informática na Escola*, p. 421-429. Porto Alegre: SBC.
- Medeiros, R. P., Falcão, T. P. & Ramalho, G. L. (2020). Ensino e Aprendizagem de Introdução à Programação no Ensino Superior Brasileiro: Revisão Sistemática da Literatura. In: *Anais do XXVIII Workshop sobre Educação em Computação*, p. 186-190, Porto Alegre: SBC
- Mélo, Á., Ferreira, S., Oliveira, A. & Andrade, W. (2023). Avaliação do Pensamento Computacional em Graduandos de Cursos de Computação: uma Disciplina de Programação faz Diferença?. In *Anais do XXXIV Simpósio Brasileiro de Informática na Educação (SBIE 2023)*, p. 47-57. Porto Alegre: SBC.
- Mooney, A. and Lockwood, J. (2020). The Analysis of a Novel Computational Thinking Test in a First Year Undergraduate Computer Science Course. In: *Journal of Teaching and Learning in Higher Education (AISHE-J)*, v. 12 , n. 1, p. 1-26 .
- Neto, S. R. S., Santos, H. R. M., Souza, A. A. & Santos, W. O. (2013). Jogos Educacionais como Ferramenta de Auxílio em Sala de Aula. In: *Proceedings of XIX Workshop de Informática na Escola (CBIE/WIE)*.
- Nielsen, J. (2012). Usability 101: Introduction to usability.
- Oliveira, C., Moura, S. P. & Sousa, E. R. (2015). TIC's na educação: a utilização das tecnologias da informação e comunicação na aprendizagem do aluno. In *Pedagogia em Ação*, v. 7, n. 1, p. 75-95.
- Petri, G., von Wangenheim, C. G. & Borgatto, A. F. (2019). Meega+: Um modelo para a avaliação de jogos educacionais para o ensino de computação. In: *Revista Brasileira de Informática na Educação*, v. 27, n. 03, p. 52-81.
- Pinho, G., Weissshahn, Y., Brum, C., Cavalheiro, G. & Cavalheiro, S. (2016). Proposta de Jogo Digital para Dispositivos Móveis: Desenvolvendo Habilidades do Pensamento Computacional. In: *Anais do XXVII Simpósio Brasileiro de Informática na Educação*, v.27, n. 1, p. 100-109. Uberlândia: SBC.
- Revythi, A. and Tselios, N. (2017). Extension of technology acceptance model by using system usability scale to assess behavioral intention to use e-learning. In: *Education and Information Technologies*, v. 24, p. 2341-2355.
- Román-González, M., Moreno-León, J. and Robles, G. (2019). Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions. Springer Singapore, Singapore, p. 79-98.
- Schlogl, L. E., Oliveira, G. C., Giovanella, G. C., Bizon, A., Santos, B., Kruger, N., Bursoni, P., Neumann, C. B., Huber, E. E., Araújo, L. P., Mattos, M. M., Zucco, F.

- D., Zendron, K. & Hein, N. (2017). Ensino do pensamento computacional na educação básica. In *Revista de Sistemas e Computação*, v. 7, n. 2, p. 304-322.
- Schmit, P. F., Bernardi, G. & Zanki C. A. (2019). Super Mario Logic: um jogo sério para auxiliar no processo de ensino e aprendizagem de lógica de programação, In: *Revista Novas Tecnologias na Educação*, Porto Alegre, v. 17, n. 1, p. 244-253.
- Siegel, S. (1975). *Estatística Não-paramétrica para as Ciências do Comportamento*. São Paulo: McGraw-Hill.
- Silva, R. R., Rivero, L. and Santos, R. P. (2021). ProgramSE: A Game for Learning Programming Logic Concepts (ProgramSE: Um Jogo para Aprendizagem de Conceitos de Lógica de Programação), In: *Brazilian Journal of Computers in Education (Revista Brasileira de Informática na Educação - RBIE)*, v. 29, p. 301-330.
- Souza, O., Morais, P. & Silva Júnior, F. (2015). Um Estudo sobre a Evasão no Curso de Licenciatura em Informática do IFRN – Campus Natal – Zona Norte. In *Anais do XXIII Workshop sobre Educação em Computação*, p. 216-225. Porto Alegre: SBC.
- Vieira, S. (2006). “Análise de variância: ANOVA”. São Paulo: Atlas.
- Wing, J. M. (2006). Computational Thinking. In: *Communications of the ACM*, v. 49, n. 3, p. 33-35.
- Wouters, P. and Van Oostendorp, H. (2017). Overview of instructional techniques to facilitate learning and motivation of serious games. In: *Instructional techniques to facilitate learning and motivation of serious games*, p. 1-16. Springer, Cham.