

Hello Food: uma jornada de aprendizagem lúdica em algoritmos, programação e Pensamento Computacional

Jeniffer Macena¹, Fernanda Pires¹, Rafaela Melo²

¹Universidade do Estado do Amazonas (UEA)
ThinkTEd Lab – Amazonas, Brazil

²Instituto de Computação – Universidade Federal do Amazonas (UFAM)
ThinkTEd Lab – Amazonas, Brazil

{jms.lic18, fpires}@uea.edu.br, rmelo@icomp.ufam.edu.br

Abstract. *Much has been discussed about the possibilities of using games for learning computing topics, especially programming, but the creation of mechanics for immersive educational games is complex. This work presents a proposal of gamification, using the four pillars of Computational Thinking, of the themes of introduction to programming from a game entitled “Hello Food”, in which the hero’s journey takes place from the resolution of problems that there is no place for a restaurant. Preliminary results indicate that it was well accepted both in terms of lucidity and learning, however, a greater number of phases is necessary to address a greater number of themes.*

Resumo. *Muito tem se discutido sobre as possibilidades de usar jogos para aprendizagem de temas de Computação, sobretudo programação, mas a criação de mecânicas para jogos educacionais imersivos é complexa. Este trabalho apresenta uma proposta de ludificação, usando os quatro pilares do Pensamento Computacional, dos temas de introdução a programação a partir de um jogo intitulado “Hello Food”, em que a jornada do herói se dá a partir da resolução de problemas que ocorrem no âmbito de um restaurante. Resultados preliminares apontam que ele foi bem aceito tanto no que diz respeito à ludicidade quanto à aprendizagem, entretanto, é necessário estender o número de fases para atender um maior número de temas.*

1. Introdução

Em se tratando de educação em Computação a Lógica e a programação são áreas importantes para representar a comunicação do indivíduo com a máquina por meio de soluções algorítmicas [Krishnamurthi and Fisler 2019], exigindo abstração de conceitos computacionais a fim de propor soluções para problemas do mundo real, tal como preconiza o conceito de Pensamento Computacional [Wing 2006]. Dessa forma, o desenvolvimento de habilidades e competências que permitam aos indivíduos aplicar a teoria no desenvolvimento de soluções de forma prática, é essencial.

Todavia, pesquisas apontam as dificuldades na compreensão dos assuntos em disciplinas voltadas para o exercício da lógica de programação, principalmente em cursos de graduação ou técnicos, o que ocasiona a evasão e reprovação de estudantes [Falckembach and de Araujo 2013]. Um possível fator que interfere nesse aspecto é a

dificuldade em abstrair o problema [Silva et al. 2018], sendo a razão de pesquisadores investigarem abordagens que simplifiquem esse processo.

Desse modo, os jogos vem sendo apontados como possíveis aliados para melhorar o desempenho dos estudantes, considerando sua característica lúdica [Shute and Ke 2012]. Um exemplo disso é a possibilidade do indivíduo estar imerso em um enredo através de dinâmicas que auxiliam na compreensão de um tema, tornando a aprendizagem um processo divertido, o que pode gerar motivação [Macena et al. 2019]. Entretanto, um grande desafio é criar um jogo de propósito educacional que concilie conteúdos curriculares sem perder os elementos lúdicos, que geram motivação e engajamento [Pires et al. 2020].

Para solucionar um problema de programação, é necessário interpretar a questão, identificar variáveis e definir quais estruturas serão utilizadas, o que pode não ser uma tarefa simples. O processo para resolver um problema, aplicando os pilares do Pensamento Computacional, que consistem em: i) decompor ou dividir um problema em partes menores, ii) identificar os padrões dentro dos problemas, iii) abstrair a informação mais importante, iv) criar um conjunto de etapas ou regras a fim de encontrar a solução, também visto como um algoritmo [BBC 2018].

O presente trabalho apresenta o processo de criação de mecânicas de aprendizagem, utilizando os 04 pilares do Pensamento Computacional como base, em um jogo educacional nomeado *Hello Food*. A finalidade desta proposta é contribuir com materiais de *game design* voltados para auxiliar no exercício do raciocínio lógico de estudantes iniciantes em programação. O restante do trabalho está organizado como segue: na Seção 2, o embasamento teórico sobre a carga cognitiva e aprendizagem em jogos; Seção 3, os trabalhos relacionados; Seção 4, o processo da criação de mecânica de aprendizagem para algoritmos; Seção 5 estão os resultados e avaliações, e na Seção 6 as considerações finais.

2. Aprendizagem em jogos e a Carga Cognitiva

Metaforicamente, assim como o computador, o cérebro humano tem uma limitação de memória para armazenar dados, que, quando processados se tornam uma informação [Robins 2019]. Segundo Miller [1956], a memória de trabalho de um indivíduo pode processar entre o intervalo de 7 ± 2 informações (*chunk*). Dessa forma, há uma quantidade de informações que a memória de curta duração é capaz de processar por vez, e se estiver além desse limite, haverá uma sobrecarga cognitiva [Feinberg and Murphy 2000, Sweller 1994]. Então, ao propor materiais com objetivos instrucionais, deve ser levada em consideração a arquitetura cognitiva humana [Kirschner et al. 2008].

Geralmente, as tarefas de programação trabalham com muitas informações relacionadas ao estado atual de dados, representando os processos que estão sendo executados [Robins 2019], tipicamente vistos em linguagem de máquina. O tempo para assimilar e compreender uma questão de programação está diretamente relacionado ao nível de complexidade do problema. Quanto mais complexo for o problema, mais tempo uma pessoa levará para resolvê-lo, o que pode se tornar uma tarefa frustrante [Kirschner et al. 2009].

Considerando a preocupação mencionada e os fundamentos da TCC, é válido discutir sobre os elementos que podem auxiliar estudantes no exercício da lógica de programação, de forma que capture a atenção do aprendiz, percepção e a memória, tal como os jogos permitem, para gerar aprendizado. Uma forma de fazer isso, no caso

de jogos educativos, é na distribuição de conteúdo e um *game design* educacional, podendo criar narrativas imersivas que propõem desafios divertidos e desafiadores, sem desmotivar a pessoa. Isso exigirá também um planejamento adequado que acompanhe o progresso do público-alvo, na qual aumentará as chances de entender um assunto [Kirschner et al. 2009], pois o jogador poderá ser analisado por meio de dados, como testes heurísticos, por exemplo. Combinando programação às características de jogos, podem ser oferecidos diferentes paradigmas quando comparados aos métodos vistos em salas de aula, devido às representações visuais exploradas em *games*, sendo um auxílio na aprendizagem do indivíduo, logo potencializando sua capacidade cognitiva. Sendo assim, é importante destacar a Teoria da Carga Cognitiva (TCC) na qual segue a defesa da existência de uma limitação na memória de longo prazo, para o indivíduo processar as informações e gerar o aprendizado [Pires et al. 2020]. Para Sweller et al. [1998], existem três tipos de carga cognitiva: i) Carga Intrínseca (*Intrinsic Load*); ii) Carga Irrelevante ou Estranha (*Extraneous Load*); e iii) Carga Relevante ou Pertinente (*Germane Load*).

A aplicação desta teoria na proposta do jogo, considera os seguintes pontos: i) a carga é considerada intrínseca quando há muita informação para uma tarefa e estes trazem interatividade entre os elementos; ii) a carga estranha é composta por atividades que não são relevantes ou não contribuem de forma direta a aprendizagem; iii) a carga pertinente está relacionado mais direto a aprendizagem favorecendo os processos cognitivos do indivíduo [Paas et al. 2003].

3. Trabalhos Relacionados

Alguns trabalhos na literatura buscaram explorar as possibilidades de unir jogos e Pensamento Computacional para auxiliar no processo de aprendizagem considerando as estruturas cognitivas humanas.

O trabalho de Michel et al. [2019] apresenta um jogo de estilo *tower defense* que utiliza a dinâmica baseado nos pilares do Pensamento Computacional. A proposta destaca-se pelo seguimento de inspiração em jogos famosos como *Plants vs Zombies* e a criação de um enredo computacional, tal como foi nomeado o local das batalhas (Reino de Turin, fazendo alusão a um dos cientistas famosos da ciência da computação, Alan Turing) onde o jogador conseguirá vencer nas fases se criar ótimas soluções. Outra similaridade, é o conteúdo de ordenação e a ligação dos pilares para executar um algoritmo que ajuda o jogador a vencer na fase, além de ser proposto a pessoas que não tem experiência a estudantes experientes na computação.

Pires et al. [2020] abordam um jogo de língua portuguesa que está fundamentado na Teoria da Aprendizagem Multimídia (TCAM) e em processos da aprendizagem ativa para criação de interfaces e a modelagem do fluxo de aprendizagem aplicado ao jogo, intitulado de “O Livro do Conhecimento”. As similaridades podem ser observadas através do embasamento teórico que há em relação a TCC com os elementos gráficos, incluindo a adaptação das telas demonstradas nos protótipos ao longo do processo para balancear a *gameplay* e evitar a sobrecarga de informação.

Este trabalho com apresentação do jogo “Hello Food” se diferencia dos demais por meio de algumas particularidades: Em relação ao trabalho (i) a proposta é voltada a questões de jogabilidade, onde os conteúdos educacionais não estão voltados essencialmente a programação ou algoritmos, mas que fazem parte dos mini jogos descritos para

solucionar fases com narrativas diversificadas sendo arbitrário ter tal assunto. Já o trabalho (ii) embora tenha referências externas de jogos de entretenimento, as fontes são diferentes do presente jogo, além da mecânica não ser do mesmo estilo por se tratar do *tower defense* focando mais na parte estratégica para obtenção de itens. Quanto ao (iii), a diferença está na disposição do design de fases pois é um jogo de plataforma, em que as mecânicas de movimentação são essenciais para capturar os itens corretos e exercitar conteúdos gramaticais, logo sendo de narrativas diferentes (gênero de fantasia) e proposta voltada ao ensino básico.

4. Processo de criação de mecânica de aprendizagem para algoritmos

O processo será descrito conforme o desenvolvimento de um jogo educacional baseado no modelo de Aprendizagem Criativa de Pires [2021], visando essencialmente sua mecânica de aprendizagem. O jogo educacional foi criado durante a disciplina de Design Instrucional (DI), por uma estudante de licenciatura em computação sob orientação docente. A seguir serão descritas as etapas para construção de uma mecânica de aprendizagem vinculada a uma narrativa de “cozinha divertida”.

4.1. Levantamento de ideias

No processo de criação do jogo, foram considerados três pontos essenciais para partir para concepção da ideia: i) definição de conteúdo educativo; ii) temática do jogo; iii) mecânica básica para *gameplay*. Assim, parte do levantamento de requisitos foi realizado através de estudo de campo com um grupo de dez estudantes de introdução a programação de um curso técnico profissionalizante em informática. A coleta de dados foi realizada de forma digital, por meio de formulário *online*¹.

A partir do estudo de campo foi possível identificar que os estudantes possuem dificuldades em Lógica de Programação, o que dificultava a execução das atividades práticas de programação. Bem como identificou-se que a instituição na qual estudam seguia algumas indicações do currículo do CIEB (Centro de Inovação para a Educação Brasileira)², essencialmente na Unidade Curricular 10, que trata de um módulo voltado a desenvolvimento de software.

Para definir então quais os conteúdos os mesmos sentiam mais dificuldade (i) foram verificados os resultados da avaliação do formulário em que 60% da turma afirmou ter dificuldades com assuntos de estruturas de dados (Vetores), Laços de Repetição e Condicionais. Logo, a proposta de intervenção considerou tais assuntos, e ao verificar se este público gosta de jogar e quais as plataformas de interesse, notou-se que 100% da turma afirmou gostar e o recurso mais utilizado para isso é o próprio celular.

Dessa forma, foi necessário consultar materiais voltados para aprendizagem em Introdução em Programação, sobretudo o Currículo de Referência em Tecnologia e Computação [Raabe et al. 2018]. O planejamento pedagógico presente no *educational game design* considerou os conhecimentos necessários para o aprendiz compreender sobre o assunto, estes sendo:

¹ todos os estudantes preencheram e assinaram um Termo de Consentimento Livre e Esclarecido (TCLE), autorizando a utilização dos dados para fins científicos.

²Disponível em: <https://curriculo.cieb.net.br/>

- **Lógica de programação:** introdução à lógica. Conceito de algoritmo. Representações visuais de algoritmos, português estruturado ou linguagem algorítmica. Comandos de entrada, processamento e saída de dados. Variáveis e constantes. Expressões e operadores. Teste de mesa. Estrutura condicional simples e composta. Estrutura de repetição. Boas práticas de programação.
- **Linguagem de programação:** tipos de dados. Operadores. Definição de identificadores. Tipos de dados básicos. Declaração e inicialização de variáveis. Definição de constantes. Expressões. Conversão de tipos (implícito e explícito). Estruturas de seleção. Dados e operadores lógicos. Avaliação de expressões lógicas. Estruturas de repetição. Conceitos de laços de repetição. Projeto de programa estruturado. Procedimentos e funções. Devolução de valores. Passagem de parâmetro por valor e por referência. Escopo de variáveis. Coleções (Vetores e matrizes) e conceitos sobre arranjos. Técnicas de depuração de programas e identificação de erros de compilação e execução.

Considerando o tema escolhido para abordar e os conteúdos, a próxima etapa foi a elaboração da temática do jogo (ii). Esta etapa visa a elaboração de um enredo que concilie com os assuntos de programação, de uma forma que se conectem para explicar como funcionam a parte teórica dos assuntos no sentido figurado da narrativa. A partir de *brainstorm* e discussões se chegou a possibilidade de criar um jogo de culinária que envolvesse missões espalhadas por uma cidade fictícia de modo a exercitarem conceitos da computação. Dessa forma, projetou-se uma narrativa contando a seguinte história: Gwen é uma empresária dona do seu próprio restaurante e quer expandir o seu negócio para outras regiões da cidade nomeada de *Otimilândia* através de mais unidades do restaurante, mas que por ser um sonho recém-realizado ainda não conseguiu ganhar o prêmio “5 estrelas” por seus ótimos serviços exercidos pelos funcionários do estabelecimento. Com o alcance dessa conquista, será possível ter o reconhecimento da comunidade gastronômica, oferecendo mais confiança quanto a qualidade do local.

Um destaque interessante dessa etapa é a forma de brincar com referências em filmes e jogos para criar um enredo atrativo, motivo pela qual ao longo dos níveis do jogo haverá trocadilhos para relacionar o aspecto de aprendizagem com *easter-eggs* de computação e jogos famosos tal como é o caso de *Overcooked*³. Por exemplo, o *storytelling* teve como inspiração o filme *Ratatouille*⁴.

Então ao tratar da mecânica, foi projetado um jogo *topdown* em que o personagem precisa se locomover pela cozinha para resolver o problema de acordo com um algoritmo apresentado, ou seja, será dado uma missão ao jogador em que ele precisará seguir os passos das instruções e quando solicitado elaborar seu próprio algoritmo durante o atendimento de um pedido (Figura 1(b)). Durante as fases, pretende-se trabalhar também com os pilares do Pensamento Computacional (1) para auxiliar a pessoa a escolher o problema (saber priorizar), se organizar para resolver dividindo em partes (decompor), reconhecer os passos que poderá otimizar na cozinha (padrão) e elaborar toda a sequência da solução passo a passo (algoritmo).

Para auxiliar no processo criativo, foi pesquisado na literatura jogos desenvolvi-

³Loja do jogo: <https://store.steampowered.com/app/448510/Overcooked/>

⁴Disponível: <https://www.disneyplus.com/pt-br/movies/ratatouille/4zRnUvYGbUZG>

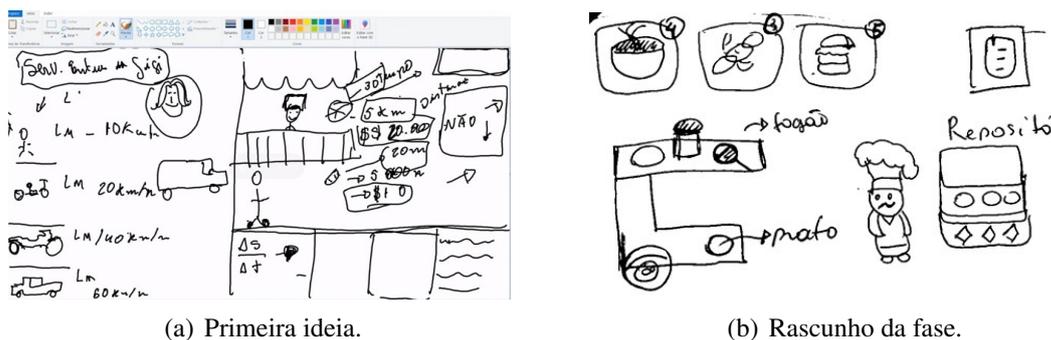


Figura 1. Esboço da ideia inicial de mecânica de jogo.

Tabela 1. Pensamento Computacional no Jogo

Pilar	Elemento	Relação com Pilar
Abstração		O jogador terá que compreender quais pedidos precisam ser atendidos logo de acordo com o valor e tempo para realizar a preparação do prato.
Decomposição		O jogador terá que separar os valores que abstraiu de mais urgência e com menos urgência, para as mesas que estão livres.
Reconhecimento de Padrão		O jogador terá que ver os tipos de pedidos que irão ser realizados, se há algo em comum em questão de categoria no cardápio e preço (urgência).
Algoritmo		O jogador terá que construir um passo a passo para fazer o pedido. Neste percurso, poderá identificar as condições de sucesso para concluir o prato.

dos nesse âmbito. Dentre eles estão: (i) Doce Sort [Alencar et al. 2020] e (ii) Operação Lovelace [Macena et al. 2020], na qual apresentam as características na O jogo (i) aborda sobre conceitos de árvores, tipos de configuração (pré-ordem, pós-ordem e em ordem) e temática de uma doceria. Quanto ao jogo (ii), trabalha com algoritmos e estruturas de dados através da execução de blocos e mecânicas de aprendizagem baseadas na movimentação da personagem. Contribuíram no sentido de oferecer alternativas para compor um jogo com múltiplas mecânicas, visando maior possibilidade para estimular aprendizagem dos assuntos.

4.2. Mapear exercícios em base de dados

Com a finalidade de ludificar os exercícios de programação de acordo com a ideia do jogo desenhada, foi necessário verificar as bases de dados cujo assunto esteja relacionado à condicionais, repetições e vetores. Neste intuito, buscou-se por plataformas de juízes on-

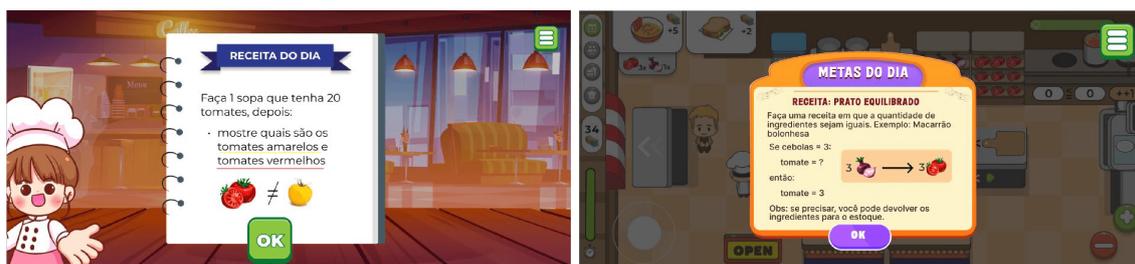


Figura 2. Comparação entre Designs para as Metas de fase.

line e atividades propostas por docentes em instituições de ensino, seja de nível superior ou técnico. Essa verificação de conteúdos não foi tão trivial para obter as informações disponibilizadas devido à falta de recursos para filtragem de assuntos considerando o material (exercícios e desafios) apresentado nas mídias de sites e soluções implementadas. Então, como critério de seleção de questões de programação, a prioridade foi dada para materiais que mostravam explicitamente tais assuntos, seja por título de listas de exercícios ou perguntas afins. Assim, foi criado um documento⁵ que contém um pseudocódigo para o jogador seguir de acordo com o desafio da fase, gerado a partir de interpretação de soluções implementadas na Linguagem de Programação C.

4.3. Aplicando conteúdo no *Level Design*

No intuito de conciliar os elementos de aprendizagem com o jogo, foram criados pseudocódigos para demonstrar a aplicação de um algoritmo (selecionado na base de dados) no formato de receitas culinárias. Dessa forma, o jogador poderá seguir as missões solicitadas pela *Chef* no decorrer das fases, servindo de orientação para resolver os desafios e praticar o raciocínio lógico por meio de comandos conforme as especificações do enunciado. Assim, o próximo passo foi construir o Design de Interface do Usuário (UI) e a modelagem das fases (*Level Design*) em que esses assuntos poderiam ser distribuídos. Essa parte foi essencial para equilibrar a parte didática dos conteúdos com o fluxo de jogo, visando não sobrecarregar o usuário e manter os princípios da Carga Cognitiva.

Tendo em vista a aplicação de conteúdos no *game*, o design gráfico precisou de ajustes para conciliar com uma identidade visual do contexto criado e a clareza dos objetivos educacionais previstos. Por isso, será demonstrado o processo criativo e mudanças realizadas para adaptar os itens da interface, tal como Pop-Ups, HUD (*Head Up Display*) e até animações criadas no *Figma*. O motivo desses ajustes são decorrentes da avaliação de outros universitários e professores ao comentarem sobre o excesso de elementos ou a falta deles na UI durante as amostras de software e testes, na qual era questionado sobre o que interpretavam com os símbolos apresentados na UI.

Na Figura 2 é exibido a versão inicial e a atual para o jogo. A modificação principal é a adição do exemplo na missão do jogo, chamada de “Metas do dia” onde o jogador precisa construir um prato, cuja quantidade de itens sejam iguais. Para isso foi criado a codificação no intuito de esclarecer uma conexão do assunto de algoritmo com o que precisa resolver no *game*. Partindo desse ponto, foi possível alterar a própria exibição da receita transcrita na versão atual, à direita na figura citada.

⁵Material complementar: <https://bitly.com/OXWdJeD>



Figura 3. Versões desenvolvidas para o Nível 1.

Desse modo, ao iniciar a *gameplay*, o jogador terá o primeiro contato com a *Chef* do restaurante e serão acrescentadas essas metas para cada fase, por sua vez, cada uma contém 2 níveis. O primeiro nível trata-se da organização do restaurante em que o usuário será representado por uma gerente (ou anfitriã) e precisará conduzir o personagem à sua respectiva mesa, de acordo com a regra criada para os vetores. Essa regra refere-se ao critério para deslocar o cliente e adicioná-lo à uma mesa vazia, podendo ser por ordenação crescente ou decrescente, por exemplo. A Figura 3 apresenta o versionamento do layout da fase já com o design aplicado até a presente fase projetada (versão 3). Na versão 1, a projetista preocupou-se mais com o design que seria utilizado, onde precisou recorrer a sites com artes licenciadas para comprar devido ao tempo para desenvolver todos os componentes do jogo. Mas, ao simplificar o cenário para o testador experimentar a mecânica básica, notou-se a necessidade de reduzir os itens do restaurante na parte externa (hall). Já a versão 3 é resultado da adição de tickets de reserva e os ícones de mesa reservados, mostrando que ainda precisa adicionar alguém lá.

No geral, caso o usuário realize uma ação incorreta, o tempo será decrementado diminuindo as chances do jogador cumprir a meta do dia. Caso essa variável fique com zero, precisará reiniciar a fase. Por isso, foi adicionado power-ups que a cada conquista do personagem, ele poderá ganhar “relógios” que aumentam as chances de vitória. Partindo para o nível 2 (Figura 4), o jogador precisará construir pratos de comida vendo qual o pedido mais urgente (posição na fila de pedidos), aplicar as condicionais para ver se a quantidade de elemento segue a instrução (utilizando operadores relacionais) do *Chef* e após isso adicioná-lo para cozimento no “Forgão” (trocadilho para o Laço de repetição For). Nesta última estrutura há parâmetros modificáveis (1º e 2º) que referem-se ao parâmetro inicial (tempo de início) e final de tempo (quantidade total de alimentos) de preparação.



Figura 4. Versões inicial e atual para o Nível 2.

5. Resultados e discussão

No processo de construção do jogo, houveram algumas dificuldades para a consolidação de uma mecânica de múltiplos desafios devido ao excesso de elementos incrementados

para tornar o jogo divertido e adequado a diferentes níveis de aprendizado. Além disso, inicialmente, a ideia era de aplicar mais questões com equações numéricas para tornar desafiador, mas verificando os testes, notou-se um desvio do assunto de programação, tornando frustrante para alguns testadores a resolução do jogo. Neste sentido, foi sugerido como intervenção a separação das fases por níveis, e quanto aos algoritmos, deveriam ser seguidos conforme era pedido na fase pela *Chef*. Sendo assim, desde as ilustrações de algoritmos (receitas dos jogadores para cumprir tarefas) a *level design* precisaram serem refeitos e submetido a avaliação da turma de DI.

Com a finalidade de validar a proposta do jogo educacional *Hello Food*, criado a partir dos processos anteriormente citados foram realizados dois estudos experimentais: um com 04 professores especialistas em programação, para validar a disposição de conteúdo e sequencia didática implícita ludificada; e o outro com estudantes, sendo 75% do Ensino Superior e 25% Ensino Médio Técnico. Algumas limitações advindas da pandemia Sars-Cov 2019 permitiram que os testes fossem realizados de forma presencial e *online*. Foi disponibilizado aos educadores o *mockup* funcional digital, automatizado, para que pudessem simular a jogatina e deixar as suas impressões. Os mesmos puderam realizar perguntas e fazer sugestões durante esse tempo. A Tabela 2 explicita os *feedbacks* obtidos.

Em relação ao perfil dos participantes estudantes ou egressos de computação, 26,7% é do sexo feminino e 73,3% é do sexo masculino. Quanto a faixa etária variou entre 17 a 28 anos. Outra característica do público é que apenas 1 pessoa trabalhou efetivamente como professor sendo formado no curso de LiComp, os demais egressos (N=3) estão atuando como desenvolvedores e são mestrandos no campo de Computação. Em relação ao tempo dos testes, a média foi de 40 minutos, já incluindo os comentários adicionais de sugestão dos testadores.

5.1. Avaliação com estudantes

Os avaliadores são 86,7% de acadêmicos ou egressos de Licenciatura em Computação e 13% corresponde a estudantes de Ensino Médio ou graduação em outras áreas. Os estudantes realizaram o teste em um protótipo digital, assim como os professores e ao fim responderam a um questionário *online* inspirado no SUS (*System Usability Scale*).

As respostas as perguntas foram dispostas em escala *Likert* na qual 1 significa “Discordo completamente” e 5 “Concordo completamente”. Uma possível ameaça a validade, que pode influenciar no flow do jogo é fato do teste ter sido realizado com um protótipo. Como pode ser observado na Figura 5, 60% dos estudantes tiveram facilidade de uso, bem como 53% indicaram sentirem-se confiantes ao usar o sistema, e 73,33% sentiram-se confiantes ao fazer uso do artefato, entretanto, alguns estudantes tiveram dificuldades com a abordagem dada aos conhecimentos, o que indicam precisar de ajuda em alguns momentos, 53,1% deram essa indicação.

De forma geral, a avaliação foi positiva, ao se considerar a colocação dos professores, entretanto, foi possível identificar que alguns elementos referentes ao *game design*, como mais *feedback*, telas de ajuda e tutoriais precisam ser implementados. Observou-se também, a partir do discurso dos estudantes e professores, que os primeiros apresentam mais facilidade de compreender a mecânica proposta, enquanto que os professores tiveram mais dúvidas. Ambos concordam que a aprendizagem de programação é complexa e

Tabela 2. Avaliação de professores de programação

<p>P1 – Sugestões: Ajustes nas missões de condicionais; Observação sobre posição das mesas e a Leitura em “Z”; Visualizou algoritmos no jogo, inclusive o Selection Sort; Considerou o nível 2 mais complicado de entender; Design: Cor das setas nas condicionais; Elementos visuais para explicar o que são as condicionais no jogo e laços de repetição; Fala do professor: “O processo está implícito, porque é um processo lógico (algoritmos são lógicos) e isso não acho uma coisa ruim.”</p>
<p>P2 – Sugestões: Diferenciar melhor ordenação e enfatizar sobre <i>Arrays</i>; Identificou todos os conceitos a uma parte do jogo devido ao constante contato com a linguagem; Recomenda o teste com público-alvo; Fala do professor: “O design cartoon ficou bem legal, mas acho que o jogo vai precisar de mais animações para combinar com estilo dele.”</p>
<p>P3 – Sugestões: Adicionar caminho alternativo no jogo para condicionais; Explicação no jogo para os parâmetros do laço de repetição; Achou a proposta interessante para criar desafios divertidos e eficazes; Gostou da referência de outros jogos e acha que deve ser aumentado o tamanho de itens de comida; As metas do dia do jogo parecem estar implícitas; Fala do professor: “Acho que fica mais fácil de entender sobre essas soluções em algoritmos no jogo, e também não sei quão complexo será no jogo conforme as questões de programação vão aumentando o grau de complexidade, mas vale o desafio. ”</p>
<p>P4 – Sugestões: Adicionar explicação sobre as estruturas e como funcionam; Discorda sobre a regra de pedidos para condicionais; Questionamento sobre pseudocódigo parecer estranho de uma linguagem; <i>Joystick</i> precisa ser mais evidente na interface e instruir quanto movimentação; Está implícito, por ser um jogo mais intuitivo. Poderia melhorar a usabilidade, reduzindo a quantidade de cliques para “arrastar” itens para uma cesta para os alimentos. Fala do professor: “Raciocínio lógico pode ser trabalhado no jogo, agora para envolver programação precisa de melhorias na interface”</p>

necessita de abordagens mais lúdicas e que os jogos podem ser importantes aliados nesse processo. Como trabalhos futuros pretende-se implementar um sistema de avaliação automático utilizando técnicas de *learning analytics* para prover um feedback de curva de aprendizagem para o jogador.

6. Considerações Finais

Este artigo apresentou a proposta de ludificação de conteúdos relacionados a lógica de programação, especificamente, com estruturas condicionais, laços de repetição e vetores aplicados a um jogo educacional 2D, utilizando os pilares do Pensamento Computacional como estratégia para projetar um design de aprendizagem divertido. O artigo envolveu a descrição de um processo criativo para transformar exercícios de programação localizados em bases de dados on-line em uma mecânica de aprendizagem que concilie com uma narrativa criada para explicar conceitos de algoritmos.

Quanto à avaliação, considerou-se compreender também a perspectiva de docentes da área para avaliar a usabilidade, jogabilidade e a proposta didática aplicada no jogo, questionando-os se é válida para quem está iniciando na área de computação. Ao sintetizar os registros obtidos por meio de uma entrevista semiestruturada, constatou-se que todos os professores acham a proposta desafiadora, válida para testar com o próprio público a qual se destina o jogo, mas que para isso, será necessário simplificar os desafios nas

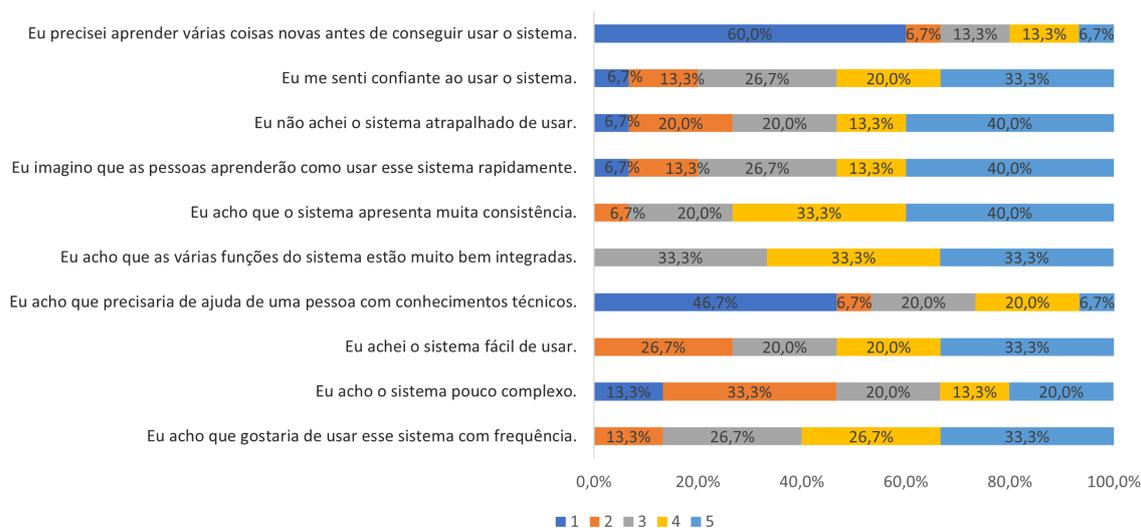


Figura 5. Dados do questionário respondido pelos estudantes.

fases iniciais e remover os *assets* que não interferem na fase. O design visual foi aprovado por todos, mas com restrições e sugestões de animações que indiquem quais passos o personagem precisa fazer. Este aspecto técnico foi uma dificuldade enfrentada devido aos recursos disponíveis na ferramenta de prototipação, mas que servirá de incentivo nos próximos passos deste estudo, a implementação e programação do jogo em uma *engine* de jogos.

Em relação a ameaça à validade, é importante destacar que o jogo foi testado também com um público aberto devido a escassez de testadores com os protótipos iniciais, sendo compartilhado por redes sociais neste período de pandemia de COVID-19. Como trabalhos futuros, pretende-se compartilhar uma nova versão desse material com outras universidades para contribuição de feedbacks e melhorias. E, será desenvolvido mais fases considerando as sugestões de todos avaliadores envolvidos para ser testado com pessoas de ensino técnico profissional e educação básica, visto que houveram comentários nas pesquisas de que o jogo possa atrair mais o público infantil devido aos traços artísticos.

Referências

- Alencar, L., Pessoa, M., and Pires, F. (2020). Um jogo educacional para exercitar propriedades de árvores binárias de busca. In *Anais dos Workshops do IX Congresso Brasileiro de Informática na Educação*, pages 226–231. SBC.
- BBC (2018). Introduction to computational thinking. <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>.
- Falckembach, G. A. M. and de Araujo, F. V. (2013). Aprendizagem de algoritmos: dificuldades na resolução de problemas. *Anais Sulcomp*, 2.
- Feinberg, S. and Murphy, M. (2000). Applying cognitive load theory to the design of web-based instruction. In *18th Annual Conference on Computer Documentation. ipcc sigdoc 2000. Technology and Teamwork. Proceedings. IEEE Professional Communication Society International Professional Communication Conference an*, pages 353–360. IEEE.

- Kirschner, F., Paas, F., and Kirschner, P. A. (2009). A cognitive load approach to collaborative learning: United brains for complex tasks. *Educational psychology review*, 21(1):31–42.
- Kirschner, P. A., Beers, P. J., Boshuizen, H. P., and Gijsselaers, W. H. (2008). Coercing shared knowledge in collaborative learning environments. *Computers in human behavior*, 24(2):403–420.
- Krishnamurthi, S. and Fisler, K. (2019). 13 programming paradigms and beyond. *The Cambridge handbook of computing education research*, page 377.
- Macena, J., Melo, G., Lais, R., Pires, F., and Pessoa, M. (2019). Gramágica: um jogo educativo para praticar classificaç ao silábica através do pensamento computacional.
- Macena, J., Pires, F., and Pessoa, M. (2020). Operação lovelace: uma abordagem lúdica para introdução de aprendizagem em algoritmos.
- Michel, F., Pires, F., and Pessoa, M. (2019). Walgor: um jogo de tower defense para o desenvolvimento do pensamento computacional e apresentação de algoritmos computacionais. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 8, page 514.
- Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81.
- Paas, F., Tuovinen, J. E., Tabbers, H., and Van Gerven, P. W. (2003). Cognitive load measurement as a means to advance cognitive load theory. *Educational psychologist*, 38(1):63–71.
- Pires, F. G., Pessoa, M. S. P., Ferreira, R. M., Bernardo, J. R. S., and de Lima, F. M. M. (2020). O livro do conhecimento: um serious game educacional para aprendizagem de ortografia da língua portuguesa. *Revista Brasileira de Informática na Educação*, 28:436.
- Pires, F. G. d. S. (2021). Thinkted lab, um caso de aprendizagem criativa em computação no nível superior.
- Raabe, A. L. A., Brackmann, C. P., and Campos, F. R. (2018). Currículo de referência em tecnologia e computação: da educação infantil ao ensino fundamental. *Centro de Inovação para a Educação Básica-CIEB*.
- Robins, A. V. (2019). 12 novice programmers and introductory programming. *The Cambridge handbook of computing education research*, page 327.
- Shute, V. J. and Ke, F. (2012). Games, learning, and assessment. In *Assessment in game-based learning*, pages 43–58. Springer.
- Silva, R. R., Fernandes, J., and Santos, R. (2018). Panorama da utilização de jogos digitais no ensino de programação no nível superior na última década: Uma revisão sistemática da literatura. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 29, page 535.
- Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, 4(4):295–312.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.