

Bitsy Tuxedo: uma Ferramenta Aprimorada para Criação de Minijogos e Narrativas Interativas

Bitsy Tuxedo: An Enhanced Tool for Creating Minigames and Interactive Narratives

Pedro Vitor Chagas Bedor¹, Beatriz Loyola Gomes de Vasconcelos¹,
João Euclides Mártires Silva¹, Erick Simões de Matos¹

¹Faculdade do Centro de Estudos e Sistemas Avançados do Recife – CESAR School
Avenida Cais do Apolo, nº 77 – 50.030-220 – Recife – PE – Brasil

{pvcb, blgv, jems2, esm}@cesar.school

Abstract. Introduction: The development of digital games is recognized as a means of exercising Computational Thinking, but low cognitive barrier tools, such as Bitsy, often make creation more accessible at the cost of restricting resources, mechanics, and possibilities for computational elaboration. **Objective:** This work presents Bitsy Tuxedo, a fork of Bitsy Color that implements new tools and features aimed at improving usability and developer experience. **Steps:** Design Science Research was adopted, with requirements gathering conducted among 24 beginner students, mapping and prioritization of the most demanded features, artifact implementation, and controlled technical validation. **Expected Results:** The prioritized features have already been implemented in the artifact's initial version; its controlled testing confirmed the absence of critical bugs and the correct application of the improvements pointed out by the students.

Keywords Bitsy, no-code tools, computational thinking, Design Science Research, educational games.

Resumo. Introdução: O desenvolvimento de jogos digitais é reconhecido como via de exercício do Pensamento Computacional, mas ferramentas de baixa barreira cognitiva, como o Bitsy, frequentemente tornam a criação mais acessível ao custo de restringir recursos, mecânicas e possibilidades de elaboração computacional. **Objetivo:** Este trabalho apresenta o Bitsy Tuxedo, um fork do Bitsy Color que implementa novas ferramentas e recursos voltados a melhorar na usabilidade e experiência do desenvolvedor. **Etapas:** Adotou-se Design Science Research, com levantamento de requisitos junto a 24 estudantes iniciantes, mapeamento e priorização das funcionalidades mais demandadas, implementação do artefato e validação técnica controlada. **Resultados Esperados:** As funcionalidades priorizadas já foram implementadas no artefato em sua versão inicial; tendo seu teste controlado confirmando a ausência de bugs comprometedores e a aplicação correta das melhorias apontadas pelos estudantes.

Palavras-Chave Bitsy, ferramentas no-code, pensamento computacional, Design Science Research, jogos educacionais.

1. Introdução

A crescente inserção da Computação na educação básica brasileira ampliou a necessidade de ferramentas capazes de desenvolver competências dessa área desde

os anos iniciais de escolarização. Esse movimento foi consolidado pela Base Nacional Comum Curricular (BNCC), que reconhece a Computação como área de conhecimento [Brasil. Ministério da Educação 2018], e reforçado pelas Diretrizes Curriculares de Computação da Sociedade Brasileira de Computação (SBC) [Sociedade Brasileira de Computação 2021]. Entre suas competências centrais destaca-se o Pensamento Computacional, definido como o conjunto de processos mentais envolvidos na formulação de problemas e de suas soluções de modo que possam ser executadas por um agente processador de informação [Wing 2006].

Uma forma eficaz de desenvolver essa competência é a criação de jogos digitais, atividade que envolve decomposição de problemas, definição de regras e refinamento contínuo de soluções [Brackmann et al. 2017]. Conforme argumenta Kalmpourtzis [Kalmpourtzis 2018], jogos educacionais bem projetados articulam aprendizagem, motivação e mecânicas de forma indissociável. Essa abordagem dialoga com a perspectiva construcionista de Resnick e Silverman [Resnick e Silverman 2005], que defendem ferramentas capazes de reduzir barreiras de entrada sem restringir a criatividade dos usuários, seguindo a filosofia *low floor, high ceiling, wide walls*.

Entre essas ferramentas destaca-se o Bitsy, desenvolvido para permitir a criação rápida de pequenos jogos e narrativas interativas [Le Doux 2019]. Nele, o usuário cria cenários organizados em salas conectadas, desenha personagens em uma grade de tiles e define diálogos por meio de uma interface visual, sem necessidade de programação. Essa abordagem reduz a barreira de entrada para iniciantes e favorece o foco na criação.

Entretanto, essa simplicidade também impõe limitações. A versão original do Bitsy carece de funcionalidades básicas, como desfazer ações, preenchimento de áreas, importação de imagens e suporte a resoluções maiores. Embora parte dessas limitações seja suprida pela comunidade por meio de *hacks* [LeBlanc 2026], trechos de código ou versões derivadas que estendem a ferramenta, sua utilização frequentemente exige editar código HTML ou recorrer a versões específicas para garantir compatibilidade. Como consequência, as exigências técnicas recaem justamente sobre o público que mais se beneficiaria da ferramenta: estudantes iniciantes.

Diante desse cenário, este trabalho apresenta o Bitsy Tuxedo, um *fork* do Bitsy Color, uma versão derivada do projeto original desenvolvida de forma independente, que integra, de maneira estável, documentada e traduzida, as funcionalidades mais demandadas pela comunidade. O artefato é direcionado a estudantes em seus primeiros contatos com o desenvolvimento de jogos digitais, buscando preservar a baixa barreira de entrada do Bitsy enquanto amplia seu potencial para o desenvolvimento do Pensamento Computacional.

2. Fundamentação e Trabalhos Relacionados

Entre diferentes trabalhos relacionados identificados (como o Bitsy HD e o Bitsy 3D), o Bitsy Color amplia a paleta de cores da ferramenta original, enquanto o Borksy facilita a aplicação de *hacks* sem edição direta de HTML. Entretanto, ambos resolvem problemas específicos e não oferecem documentação integrada. O Bitsy Tuxedo diferencia-se por integrar, em um único ambiente, funcionalidades priorizadas a partir de um levantamento sistemático com usuários.

Tabela 1. Tabela de ferramentas relacionadas ao Bitsy.

Ferramenta	Pontos Positivos	Pontos Negativos
Bitsy Color [AuryFae 2024]	• Suporte a paletas de cores maiores • Não altera a curva de aprendizagem	• Instabilidade • Ausência de documentação oficial
Bitsy 3D [aloelazoe 2020]	• Adiciona ferramentas dedicadas ao modo 3D	• Curva de aprendizagem maior • Instabilidade com a Câmera Universal
Bitsy HD [Bednarski 2018]	• Duplicação na quantidade de pixels por tile	• Baixa compatibilidade com <i>hacks</i> • Sem manutenção
Borksy [AYolland 2018]	• Interface para aplicar <i>hacks</i> sem edição direta de HTML	• Dependência do conhecimento de <i>hacks</i> independentes • Ausência de documentação

3. Metodologia e Desenvolvimento

O desenvolvimento seguiu os princípios de Design Science Research (DSR), abordagem orientada à construção de artefatos que resolvam problemas relevantes e explícitos de uma classe de usuários [Hevner et al. 2004]. O ciclo adotado compreendeu quatro etapas: (1) levantamento de requisitos com usuários; (2) mapeamento e priorização de funcionalidades; (3) implementação do artefato; e (4) validação técnica.

3.1. Levantamento de requisitos

O levantamento foi conduzido por meio de formulário aplicado a 24 estudantes de duas turmas da disciplina de Introdução à Computação, após período de uso livre do Bitsy ou de suas variantes (Bitsy Color, Bitsy HD, Bitsy 3D) em atividade prática de criação de minijogos. O instrumento solicitava, em campos abertos, a principal dificuldade enfrentada, o que fez falta durante o uso e quais funcionalidades os estudantes gostariam de ver em uma versão aprimorada. As respostas foram organizadas por afinidade temática, agrupando queixas semanticamente equivalentes em categorias, técnica qualitativa comum em etapas exploratórias de DSR.

A Tabela 2 resume as categorias de dor mais reportadas, com a contagem de respostas em que cada categoria foi identificada. Instabilidade e travamentos durante a edição foram a queixa mais recorrente, mencionada por metade da amostra (12/24), frequentemente associada a perdas de trabalho não salvo. A importação limitada de mídia apareceu em quase um terço dos relatos (7/24), seguida pela ausência de uma função de desfazer (5/24). Documentação insuficiente, dificuldade de pintura célula a célula e interface pouco informativa foram mencionadas por poucos participantes, mas de forma consistente entre si (2 a 3 respostas cada).

As categorias identificadas indicam que instabilidade, limitações na importação de mídia e ausência de mecanismos de desfazer constituem os principais obstáculos enfrentados pelos estudantes, orientando diretamente a priorização das funcionalidades implementadas.

Tabela 2. Categorias de dor reportadas pelos estudantes (n=24, respostas abertas; uma resposta pode mencionar mais de uma categoria).

Categoria	Respostas (n/24)	Exemplo de relato
Instabilidade/travamentos	12	“Travava com frequência, e cheguei a perder vários arquivos”
Importação limitada de mídia	7	“Não ter como colocar áudio”; “a importação de arquivos não funciona direito”
Ausência de desfazer	5	“Senti muita falta do Ctrl+Z, chegamos a apagar até a paleta de cores sem querer”
Falta de documentação/tutoriais	3	“Não tem fórum de ajuda online, não tem wiki detalhada”
Pintura célula a célula	2	“Ter que editar quadrado por quadrado, devagar”
Interface pouco informativa	2	“A interface de usuário deixa a desejar. Pouquíssimos botões”

3.2. Mapeamento e priorização

As categorias identificadas foram cruzadas com o catálogo de *hacks* já disponíveis na comunidade, distinguindo problemas que exigiam apenas integração de soluções existentes daqueles que exigiam implementação original. Cinco funcionalidades concentraram a maior parte das menções e foram priorizadas para a primeira versão do artefato. São elas: sistema histórico de ações, balde de tinta para preenchimento de área, importação local de imagem e áudio, temas customizáveis incluindo modo escuro, e tradução completa da interface para português brasileiro. Esta última funcionalidade foi motivada por relatos de estudantes que recorriam a ferramentas externas de tradução automática para compreender a interface original em inglês.

3.3. Implementação

O Bitsy Tuxedo foi implementado como *fork* do Bitsy Color em JavaScript e HTML5, sob licença MIT, preservando a mecânica original da ferramenta e incorporando as funcionalidades priorizadas no levantamento de requisitos. Sobre essa base, o sistema de desfazer foi implementado por meio do uso de diferentes estados que registram alterações na grade de desenho, nos mapas de salas e no editor de diálogos. Também foi implementado um balde de tinta baseado no algoritmo de preenchimento por adjacência (flood fill), reduzindo o número de cliques necessários para colorir áreas contínuas.

Além disso, foram adicionadas funcionalidades de importação local de imagens e áudios, eliminando a dependência de *hacks* externos, bem como temas de interface customizáveis, incluindo modo escuro. A interface também foi integralmente traduzida para o português brasileiro, reduzindo a barreira linguística identificada no levantamento. A Figura 1 apresenta a interface do Bitsy Tuxedo com as principais funcionalidades implementadas.

O código-fonte e a documentação do Bitsy Tuxedo foram disponibilizados em repositório público através do Github [Loyola e Bedor 2026]. Além de documentar cada

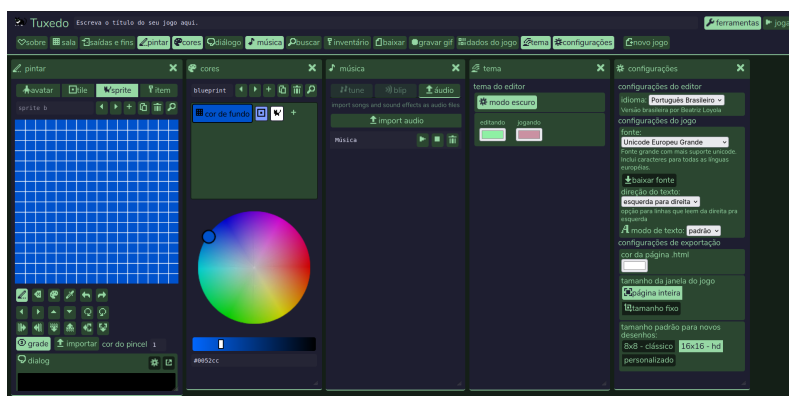


Figura 1. Captura de Tela da Interface Principal do Bitsy Tuxedo.

funcionalidade nova, o repositório preenche lacunas de documentação já presentes no Bitsy original, respondendo diretamente à queixa de falta de documentação identificada no levantamento (Tabela 2).

3.4. Validação

Antes da disponibilização do artefato, foi realizado um teste interno para verificar a estabilidade da ferramenta e o correto funcionamento das funcionalidades implementadas. Foram exercitadas as principais operações do editor, incluindo criação de sprites, preenchimento de áreas, histórico de ações, importação de mídia e alternância de temas. Os testes não identificaram falhas críticas que comprometessem o uso do artefato, indicando que as funcionalidades implementadas operam conforme o esperado. A avaliação do impacto pedagógico do Bitsy Tuxedo permanece como trabalho futuro e será conduzida com estudantes em ambiente controlado.

4. Considerações Finais

Este trabalho apresentou o Bitsy Tuxedo, um *fork* do Bitsy Color desenvolvido a partir de um diagnóstico sistemático das necessidades de estudantes iniciantes. Sua principal contribuição consiste na integração de funcionalidades antes distribuídas em hacks da comunidade e no desenvolvimento de novos recursos voltados à melhoria da usabilidade da ferramenta, preservando sua proposta de baixa barreira de entrada.

O levantamento de requisitos baseou-se em relatos de 24 estudantes de duas turmas de uma única disciplina introdutória, o que limita a generalização das necessidades identificadas para outros contextos. Além disso, a validação realizada restringiu-se a testes técnicos conduzidos pela equipe de desenvolvimento, destinados a verificar a ausência de falhas críticas e o funcionamento das funcionalidades implementadas.

Como trabalhos futuros, pretende-se realizar uma validação com usuários em ambiente controlado, aplicando o Bitsy Tuxedo em disciplinas introdutórias e workshops de iniciação. Em etapa posterior, pretende-se avaliar seu impacto no desenvolvimento do Pensamento Computacional e da agência dos estudantes por meio de instrumentos apropriados.

Referências

- aloelazoe (2020). Bitsy 3d beta. <https://aloelazoe.itch.io/bitsy-3d>. Extensão do editor Bitsy para criação de jogos em ambiente 3D. Acesso em: 6 jul. 2026.
- AuryFae (2024). Bitsy color. <https://auryfae.itch.io/multicolorbitsy>. Modificação do motor Bitsy com suporte a múltiplas cores. Acesso em: 6 jul. 2026.
- AYolland (2018). Borksy game hacker. <https://ayolland.itch.io/borksy>. Ferramenta para personalização e extensão de jogos desenvolvidos com Bitsy. Acesso em: 6 jul. 2026.
- Bednarski, F. (2018). Bitsy hd. <https://vonbednar.itch.io/bitsy-x2>. Versão modificada do Bitsy com suporte a sprites de 16x16 pixels. Acesso em: 6 jul. 2026.
- Brackmann, C. P., Román-González, M., Robles, G., Moreno-León, J., Casali, A., e Barone, D. (2017). Development of computational thinking skills through unplugged activities in primary school. In *Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE)*, pages 65–72.
- Brasil. Ministério da Educação (2018). *Base Nacional Comum Curricular*. MEC.
- Hevner, A. R., March, S. T., Park, J., e Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.
- Kalmpourtzis, G. (2018). *Educational Game Design Fundamentals: A Journey to Creating Intrinsically Motivating Learning Experiences*. CRC Press.
- Le Doux, A. (2019). Bitsy. <https://ledoux.itch.io/bitsy>. Acesso em: 6 jul. 2026.
- LeBlanc, S. S. (2026). bitsy-hacks. <https://github.com/seleb/bitsy-hacks>. GitHub repository. Accessed: 2026-07-06.
- Loyola, B. e Bedor, P. (2026). Bitsy tuxedo. <https://github.com/FORJAGame/bitsy-tuxedo>. Acesso em: 5 de agosto de 2026.
- Resnick, M. e Silverman, B. (2005). Some reflections on designing construction kits for kids. In *Proceedings of the Interaction Design and Children (IDC)*, pages 117–122.
- Sociedade Brasileira de Computação (2021). *Referenciais de Formação em Computação: Educação Básica*. SBC.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.