www.revistaft.com.br

Revista FT | ISSN 1678-0817 | v. 30, n. 164, 2026 | DOI: 10.69849/qzazh839

Escape Box com ESP32 para o ensino introdutório de sistemas embarcados: projeto de um jogo educacional orientado por Design Science Research

ESP32-based Escape Box for introductory embedded systems education: design of an educational game guided by Design Science Research

Vinicius Alves da Costa[1]

Mário Sérgio Cavalcante[2]

RESUMO

O artigo apresenta o projeto de uma Escape Box controlada por ESP32 para o ensino introdutório de sistemas embarcados. O ensino desse conteúdo exige que o estudante relacione programação e hardware, o que eleva a carga cognitiva de iniciantes. A pesquisa adotou a Design Science Research, operacionalizada pela metodologia de Peffers *et al.* (2007), e organizou o projeto do artefato pelas seis etapas do framework EscapED. O artefato combina dois microcontroladores: um ESP32 mestre, que controla travas, temporizador, dicas e registro da partida, e uma placa programada pelas equipes. Três desafios progressivos, sobre saída digital, entrada analógica e comunicação serial, só avançam quando o mestre verifica, pelo comportamento elétrico da placa, que o código cumpre a regra da etapa. Os critérios de verificação foram desenhados para resistir a soluções triviais, como manter um pino fixo ou memorizar respostas. O funcionamento foi demonstrado em simulação no Wokwi, por meio de casos de teste funcionais e adversariais; os 14 casos tiveram o resultado esperado depois da correção de três defeitos que os próprios testes revelaram. A construção física e a avaliação com estudantes, para a qual se propõe um plano com o modelo MEEGA+, ficam como trabalhos futuros.

Palavras-chave: jogo educacional; escape room educacional; sistemas embarcados; ESP32; Design Science Research; Ciências Exatas e da Terra.

[1]Graduando em Engenharia de Software pela Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (FUCAPI), Manaus, AM. E-mail: viniciusalvesdacosta.dev@gmail.com.

[2]Doutor em Engenharia Elétrica e de Computação pela Universidade Federal do Rio Grande do Norte (UFRN), professor orientador do Trabalho de Conclusão de Curso na Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (FUCAPI), Manaus, AM.

ABSTRACT

This article presents the design of an ESP32-controlled Escape Box for introductory embedded systems education. Learning this subject requires students to relate programming and hardware, which raises the cognitive load for beginners. The research followed Design Science Research, operationalized by the methodology of Peffers *et al.* (2007), and organized the artifact design according to the six stages of the EscapED framework. The artifact combines two microcontrollers: a master ESP32, which controls locks, timer, hints and game logging, and a board programmed by student teams. Three progressive challenges, on digital output, analog input and serial communication, advance only when the master verifies, from the electrical behavior of the board, that the code meets the rule of the stage. The verification criteria were designed to resist trivial solutions, such as holding a pin at a fixed level or memorizing answers. The artifact was demonstrated in simulation on Wokwi through functional and adversarial test cases; all 14 cases produced the expected result after three defects revealed by the tests were fixed. The physical construction and the evaluation with students, for which an evaluation plan based on the MEEGA+ model is proposed, are left as future work.

Keywords: educational game; educational escape room; embedded systems; ESP32; Design Science Research; Exact and Earth Sciences.

1 INTRODUÇÃO

Sistemas embarcados estão presentes da automação industrial aos dispositivos de Internet das Coisas (IoT), e seu desenvolvimento integra, de forma indissociável, hardware e software (Barr; Massa, 2006; Marwedel, 2021). Para quem está começando, esse ensino é exigente: o estudante precisa mobilizar ao mesmo tempo programação, eletrônica digital e comunicação entre dispositivos, e boa parte do que acontece dentro do sistema não é diretamente observável. Elementos que precisam ser processados simultaneamente aumentam a carga cognitiva da tarefa (Sweller, 1988), o que tende a dificultar a aprendizagem quando o conteúdo é apresentado de forma predominantemente expositiva.

A literatura sobre aprendizagem ativa indica resultados superiores aos da instrução puramente expositiva (Freeman *et al.*, 2014; Prince, 2004). Entre essas estratégias, os escape rooms educacionais têm sido aplicados ao ensino de computação com efeitos positivos sobre engajamento e aprendizagem (Gordillo; López-Fernández, 2024; López-Pernas *et al.*, 2019b). Nesses trabalhos, porém, os enigmas são resolvidos em telas e a verificação das respostas é feita por software. Quando o conteúdo é sistemas embarcados, a resposta correta é um

comportamento físico: um pino que oscila na frequência certa, uma saída que reage a um sensor, uma mensagem trocada entre dois dispositivos.

Este artigo apresenta o projeto de uma Escape Box controlada por ESP32 em que cada etapa só é liberada quando o próprio artefato verifica, pelo comportamento elétrico de uma placa programada pelos estudantes, que o código cumpre a regra do desafio. O trabalho foi conduzido pela Design Science Research (DSR) e se limita ao projeto, à implementação do firmware e à demonstração em simulação. A construção física e a avaliação com estudantes são tratadas como trabalhos futuros, com um plano de avaliação proposto na seção 7.

1.1 Problema e objetivos

O problema que orienta o trabalho é: como projetar um jogo educacional no formato de escape box em que a programação de um microcontrolador seja a própria mecânica de progressão, com verificação automática do comportamento físico do código?

O objetivo geral é projetar e demonstrar uma Escape Box controlada por ESP32 para o ensino introdutório de sistemas embarcados. Os objetivos específicos são:

- **OE1:** selecionar conteúdos introdutórios de sistemas embarcados adequados a desafios verificáveis automaticamente;
- **OE2:** analisar teorias de aprendizagem e elementos de design de jogos que fundamentam o projeto dos desafios;
- **OE3:** projetar uma sequência de desafios progressivos, resistentes a soluções triviais;
- **OE4:** especificar a arquitetura de hardware e implementar o firmware do artefato;
- **OE5:** demonstrar o funcionamento do artefato em simulação, por casos de teste funcionais e adversariais, e propor um plano de avaliação com participantes.

A principal contribuição é uma arquitetura de dois microcontroladores em que o firmware do jogo fica isolado do código dos estudantes e atua como avaliador automático do comportamento físico deste código. O hardware, o firmware e o

roteiro de aplicação estão documentados para reuso em https://github.com/Alvesvnc/escape-box-esp32.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Sistemas embarcados e seu ensino

Um sistema embarcado combina hardware e software para executar funções dedicadas dentro de um sistema maior, sob restrições de processamento, memória, energia e custo (Barr; Massa, 2006). Marwedel (2021) destaca a interação desses sistemas com o mundo físico por meio de sensores e atuadores, o que exige do projetista raciocínio simultâneo sobre código, sinais elétricos e tempo.

No ensino, essa simultaneidade pesa sobre iniciantes. A teoria da carga cognitiva sustenta que tarefas com muitos elementos interdependentes consomem a memória de trabalho e reduzem a capacidade de aprender com a própria resolução (Sweller, 1988). Atividades que tornam visível o efeito de cada linha de código ajudam a fragmentar essa carga.

2.2 Aprendizagem ativa

Para Bonwell e Eison (1991), aprendizagem ativa envolve atividades em que os estudantes fazem algo e refletem sobre o que fazem. Em meta-análise com 225 estudos de ciências, tecnologia, engenharia e matemática, Freeman *et al.* (2014) constataram que a aprendizagem ativa eleva o desempenho e reduz a reprovação em relação à instrução expositiva, conclusão convergente com a revisão de Prince (2004). A aprendizagem experiencial de Kolb (1984) descreve um ciclo de experiência concreta, observação reflexiva, conceituação abstrata e experimentação ativa, que a programação de dispositivos físicos reproduz a cada tentativa.

2.3 Jogos educacionais e gamificação

Deterding *et al.* (2011) definem gamificação como o uso de elementos de design de jogos em contextos que não são jogos, e a distinguem dos jogos sérios, que são jogos completos com finalidade além do entretenimento. Por essa definição, uma escape box é um jogo completo: tem objetivo, regras, desafios encadeados, restrição de tempo e condição de vitória. Este trabalho trata, portanto, como **jogo educacional**, e não como gamificação de uma atividade existente.

Os conceitos da gamificação continuam úteis como vocabulário de projeto. Werbach e Hunter (2012) organizam os elementos de jogo em dinâmicas, mecânicas e componentes, e Kapp (2012) associa desafios, níveis e feedback imediato ao envolvimento dos estudantes. A Teoria da Autodeterminação explica por que esses elementos podem motivar: a motivação intrínseca é favorecida pela satisfação das necessidades de autonomia, competência e pertencimento (Ryan; Deci, 2000). As revisões de Hamari, Koivisto e Sarsa (2014) e de Sailer e Homner (2020) mostram, contudo, que os efeitos dependem do desenho da intervenção, o que justifica alinhar cada elemento a um objetivo de aprendizagem.

2.4 Escape rooms e escape boxes educacionais

Escape rooms surgiram como entretenimento comercial: equipes resolvem enigmas em tempo limitado para cumprir um objetivo, geralmente sair de uma sala (Nicholson, 2015). Na versão educacional, os enigmas incorporam conteúdos curriculares, de modo que avançar exige dominá-los (Nicholson; Cable, 2021). Para orientar esse projeto, Clarke *et al.* (2017) propuseram o framework EscapED, com seis etapas: participantes, objetivos, tema, enigmas, equipamentos e avaliação.

O formato admite versões compactas, em que desafios e travas cabem em um único objeto portátil, chamado escape box ou breakout box. O próprio artigo do EscapED relata um protótipo montado dentro de uma maleta, para ser transportado e aplicado em diferentes ambientes (Clarke *et al.*, 2017). É esse o formato adotado aqui.

2.5 Computação física e construcionismo

O construcionismo de Papert (1980) propõe que o conhecimento se constrói melhor quando o estudante produz um artefato concreto que responde às suas ações. A computação física trata de sistemas que sentem e agem sobre o mundo por sensores e atuadores controlados por software (O'Sullivan; Igoe, 2004). Programar um dispositivo que acende um LED ou aciona uma trava dá ao estudante um retorno concreto sobre a correção do código.

O ESP32, da Espressif Systems, é um microcontrolador de baixo custo com Wi-Fi, Bluetooth, GPIO, conversor analógico-digital e múltiplas UARTs (Espressif Systems, 2026). Pode ser programado pelo Arduino IDE, o mesmo ambiente das

placas Arduino, e neste trabalho é usado tanto como controlador da caixa quanto como placa dos estudantes.

3 TRABALHOS RELACIONADOS

Na busca realizada, nenhum trabalho combinou os três traços desta proposta: escape box portátil, programação de microcontrolador como mecânica de jogo e verificação automática do comportamento físico do código. A busca foi exploratória, e não uma revisão sistemática, e seu procedimento está resumido abaixo para que a lacuna possa ser verificada.

- **Data:** novembro de 2026.
- **Fontes:** mecanismo de busca geral da web, com seleção de resultados da SBC OpenLib (SOL), IEEE, arXiv e DOAJ.
- **Termos:** "educational escape room" com "embedded systems", "microcontroller" ou "Arduino"; "escape room educacional" com "Arduino", "computação" ou "MEEGA+"; "escape box" com "micro:bit" ou "Arduino".
- **Período:** publicações de 2017 a 2026, com prioridade para 2022 a 2026.
- **Inclusão:** escape room ou escape box com finalidade educacional no ensino de computação, ou automação de escape rooms com microcontroladores, com descrição suficiente da mecânica.
- **Exclusão:** aplicações fora da computação sem uso de microcontroladores, e materiais sem revisão editorial, como tutoriais e repositórios de código.

O Quadro 1 compara os trabalhos selecionados quanto ao formato,à forma de verificação das respostas e à avaliação.

Quadro 1 – Comparação entre trabalhos relacionados e a proposta

Trabalho	Formato e área	Papel da programação	Verificação das respostas	Avaliação
López-Pernas *et al.* (2019a; 2019b)	Escape room de duas horas em laboratório de informática; programação web no ensino superior	Mecânica: as equipes editam templates, consultas SQL e formulários para obter as pistas	Plataforma web que registra enigmas resolvidos, dicas e tempo	2019a: percepção dos estudantes; 2019b: pré e pós-teste
Gordillo e López-Fernández (2024)	Escape room de 1h30 com cinco enigmas; modelagem UML em engenharia de software	Não há; os enigmas envolvem interpretar e criar diagramas	Plataforma web	Ensaio randomizado com 326 estudantes, pré e pós-teste
Classe, Castro e Oliveira (2025), DBEscape	Escape room digital baseado no EscapED; consultas SQL em banco de dados no ensino superior	Mecânica: as equipes escrevem consultas SQL em um SGBD online	O resultado de cada consulta é o código que abre uma porta	Questionário de percepção com 18 estudantes
Castro, Classe e Oliveira (2026)	Escape room híbrido, com QR codes no prédio e etapa digital, baseado no EscapED; introdução a sistemas de informação	Não há	Senhas e códigos obtidos nas respostas	Questionário adaptado do MEEGA+ com 32 estudantes
Track *et al.* (2026)	Cinco jogos com Arduino em caixas de MDF; algoritmos e eletrônica nos ensinos fundamental e médio técnico	Não há: os estudantes jogam artefatos programados pelos autores	Lógica do próprio jogo embarcado	Questionários pré e pós baseados no MEEGA+, em três turmas
Costa *et al.* (2020), Kalinin II	Escape room físico em duas salas, baseado no EscapED; matrizes, em turma de mestrado em Informática na Educação	Não há	Enigmas em papel e terminal digital	MEEGA+ adaptado
Pfeifer *et al.* (2021)	Curso em que os estudantes constroem um escape room automatizado; sistemas embarcados no ensino superior	Projeto de construção ao longo do curso	Não se aplica: o escape room é o produto do curso	Relato de experiência
Lubyagin, Ivanova e Loginov (2017)	Automação de escape rooms comerciais com microcontroladores	Não há: os microcontroladores são programados pelos criadores	Sensores e travas controlados por microcontrolador	Não se aplica
Esta proposta	**Escape box portátil com dois ESP32; sistemas embarcados introdutório**	**Mecânica central: o código da placa dos estudantes destrava cada etapa**	**O ESP32 mestre verifica o comportamento elétrico da placa**	**Demonstração em simulação; plano com MEEGA+ e log**

Fonte: elaborado pelos autores (2026).

López-Pernas *et al.* (2019a) já usavam a programação como mecânica de jogo: para avançar, as equipes corrigem e completam código de uma aplicação web. O trabalho seguinte dos mesmos autores, com o mesmo desenho, mediu ganhos de aprendizagem com pré e pós-teste (López-Pernas *et al.*, 2019b). Gordillo e López-Fernández (2024) encontraram, em ensaio randomizado, desempenho superior do escape room em relação a uma aula expositiva de mesma duração.

Entre os trabalhos brasileiros recentes, Classe, Castro e Oliveira (2025) também usam código como mecânica: no DBEscape, o resultado de cada consulta SQL é a chave que abre uma porta da mansão virtual. Track *et al.* (2026) se aproximam no hardware, com jogos embarcados em Arduino e montados em caixas de MDF, mas os estudantes jogam artefatos já programados. Castro, Classe e Oliveira (2026) aplicaram o EscapED e um questionário adaptado do MEEGA+ a um escape room híbrido, e Castro *et al.* (2026) e Vasconcelos *et al.* (2026) apresentam jogos digitais com mecânica de escape room para portas lógicas e lógica proposicional.

A diferença desta proposta, portanto, não está em programar para avançar, e sim em três pontos. Primeiro, o código produz um comportamento físico, e não texto em uma tela. Segundo, esse comportamento é verificado automaticamente pelo próprio artefato, sem plataforma web. Terceiro, o formato projetado é o de uma caixa portátil, pensada para caber em uma aula. Pfeifer *et al.* (2021) são os mais próximos no conteúdo, mas seus estudantes constroem o escape room ao longo de um curso, em vez de jogar. Lubyagin, Ivanova e Loginov (2017) mostram a viabilidade de microcontroladores no controle de travas, sem finalidade didática. Costa *et al.* (2020), embora fora da computação, mostram que o MEEGA+ pode ser adaptado a escape rooms físicos, opção adotada no plano de avaliação.

4 METODOLOGIA

A pesquisa é aplicada, de caráter exploratório e descritivo (Gil, 2017; Prodanov; Freitas, 2013), e foi conduzida pela Design Science Research, abordagem voltada à construção e à avaliação de artefatos que resolvem problemas práticos (Dresch; Lacerda; Antunes Júnior, 2015; Hevner *et al.*, 2004). Hevner (2007) descreve a DSR em três ciclos: o de relevância, que liga o trabalho ao problema do

ensino de sistemas embarcados; o de rigor, que fundamenta o artefato na literatura da seção 2; e o de design, que alterna construção e avaliação.

O processo seguiu as seis atividades da Design Science Research Methodology (DSRM) de Peffers *et al.* (2007), relacionadas aos objetivos específicos no Quadro 2. Para a avaliação, adotou-se a distinção de Venable, Pries-Heje e Baskerville (2016) entre avaliação artificial, feita sem usuários reais, e avaliação naturalista, feita com usuários no contexto de uso. Este trabalho realiza a avaliação artificial, por testes em simulação, e propõe a naturalista como etapa seguinte.

Quadro 2 – Etapas da DSRM, objetivos e atividades realizadas

Etapa da DSRM	Objetivo	Atividade	Seção
1. Identificação do problema e motivação	Problema	Caracterização das dificuldades do ensino introdutório de sistemas embarcados e busca de trabalhos relacionados	1 a 3
2. Definição dos objetivos da solução	OE1	Seleção dos conteúdos e definição dos requisitos	5.1, 5.2 e 5.5
3. Projeto e desenvolvimento	OE2, OE3 e OE4	Projeto dos desafios e dos elementos de jogo; especificação do hardware; implementação do firmware	5.3 a 5.5
4. Demonstração	OE5	Execução de partidas completas em simulação no Wokwi	6
5. Avaliação	OE5	Avaliação artificial por casos de teste funcionais e adversariais; plano de avaliação naturalista	6 e 7
6. Comunicação	-	Este artigo e a publicação do código em repositório aberto	-

Fonte: elaborado pelos autores (2026), com base em Peffers *et al.* (2007).

O trabalho percorreu dois ciclos curtos de construção e avaliação em simulação: o artefato foi projetado, implementado e testado, e os defeitos encontrados nos testes foram corrigidos e testados de novo (seção 6.3). Ciclos seguintes, com o protótipo físico e com participantes, estão previstos na seção 7.

5 O ARTEFATO

O artefato projetado é uma caixa com três compartimentos trancados, controlada por um ESP32 mestre, em que cada compartimento abre quando uma segunda placa, programada pela equipe, apresenta o comportamento elétrico esperado. O projeto seguiu as seis etapas do EscapED (Clarke *et al.*, 2017), apresentadas a seguir.

5.1 Participantes

O público-alvo são estudantes de graduação em cursos de computação, maiores de 18 anos, que já cursaram introdução à programação (variáveis, condicionais, laços e funções), conhecem o básico da linguagem C++, usada para programar as placas pelo Arduino IDE, e não têm experiência com microcontroladores. O jogo é cooperativo, para equipes de três estudantes, com uma caixa por equipe e 45 minutos de partida. A dificuldade cresce a cada etapa, e o desafio 3 recebe apoio extra, descrito em 5.4, por combinar comunicação serial e manipulação de caracteres.

5.2 Objetivos

Os conteúdos foram escolhidos por três critérios: pertencer ao núcleo introdutório da disciplina, permitir progressão de complexidade e produzir um comportamento elétrico verificável automaticamente (OE1). O resultado são três funções básicas de qualquer sistema embarcado: gerar uma saída, ler uma entrada e trocar dados com outro dispositivo (Quadro 3). Como objetivos transversais, a atividade exercita depuração e trabalho em equipe.

Quadro 3 – Conteúdos e conceitos de cada desafio

Desafio	Conteúdo	Conceitos mobilizados
1	Saída digital e temporização	`pinMode`, `digitalWrite`, `delay`, frequência e período
2	Entrada analógica e estruturas condicionais	Conversor analógico-digital, `analogRead`, faixas de valores, `if/else`
3	Comunicação serial (UART)	Recepção e envio de linhas, laços, manipulação de caracteres

Fonte: elaborado pelos autores (2026).

5.3 Tema e narrativa

O jogo usa o modo fuga do EscapED, em partida avulsa. A narrativa é regional: a estação de monitoramento de um rio foi bloqueada, e a equipe precisa restaurá-la antes da cheia. Cada desafio corresponde a uma função real de um sistema embarcado de monitoramento. No desafio 1, o LED de status precisa voltar a piscar para indicar que o módulo está ativo. No desafio 2, o alarme do sensor de nível só pode disparar na faixa de alerta. No desafio 3, a equipe decifra a mensagem da estação vizinha e confirma o recebimento.

5.4 Enigmas

Os enigmas são tarefas de programação na placa da equipe (Quadro 4). O critério de projeto central foi que nenhum desafio pudesse ser superado por uma solução trivial, condição para que o progresso no jogo dependa de implementar o conceito (OE3).

Quadro 4 – Sequência de desafios e critério de verificação

Desafio	O que a equipe faz	Como o mestre verifica	Recompensa
1. Restaurar o sinal	Faz o pino 23 piscar a 2 Hz	Mede por interrupção o período entre bordas de subida e exige seis ciclos seguidos entre 450 e 550 ms	Abre o compartimento 1, com a faixa de alerta
2. Calibrar o sensor	Lê o potenciômetro e ativa o sinal só dentro da faixa	Lê o mesmo potenciômetro a cada 50 ms e exige acerto dentro e fora da faixa	Abre o compartimento 2, com a chave da cifra
3. Decifrar a mensagem	Completa a função que decifra a mensagem recebida pela UART	Envia um código cifrado aleatório e exige três respostas corretas seguidas, cada uma em até 3 s	Abre a tampa final

Fonte: elaborado pelos autores (2026).

Cada critério fecha um atalho específico. Um pino fixo em nível alto não gera bordas e não conclui o desafio 1. No desafio 2, um pino sempre alto falha nas amostras fora da faixa, e uma zona de tolerância de 150 unidades nas bordas compensa diferenças de leitura entre os conversores das duas placas. No desafio 3, a cifra de César é aplicada a um código de cinco letras sorteado a cada mensagem, com cerca de 11,9 milhões de combinações, o que torna inútil memorizar respostas.

O prazo de 3 segundos vale para o programa responder, e não para a equipe, que tem o tempo da partida para escrevê-lo. Para reduzir a carga do desafio mais difícil, o código-modelo entregue à equipe já traz a leitura e o envio pela UART prontos; falta apenas a função que decifra o texto. As instruções aparecem no display e em cartões dentro dos compartimentos, e um botão libera até três dicas por partida. O EscapED recomenda planejar como as dicas serão dadas quando a equipe travar (Clarke *et al.*, 2017); a penalidade de 3 minutos por dica é decisão deste projeto, para que pedir ajuda tenha custo no tempo da partida.

O Quadro 5 relaciona os elementos de jogo à hierarquia de Werbach e Hunter (2012) e à necessidade da Teoria da Autodeterminação que cada um procura sustentar (OE2).

Quadro 5 – Elementos de jogo do artefato

Elemento	Nível	Implementação	Necessidade (TAD) ou efeito pretendido
Narrativa	Dinâmica	Estação do rio bloqueada antes da cheia	Imersão e significado da tarefa
Progressão	Dinâmica	Três etapas de dificuldade crescente	Competência
Restrição de tempo	Dinâmica	Temporizador regressivo de 45 minutos	Desafio e foco
Desafios	Mecânica	Tarefas de programação verificadas pelo artefato	Competência
Feedback imediato	Mecânica	Display, LEDs e buzzer a cada tentativa	Competência
Cooperação	Mecânica	Equipes de três estudantes	Pertencimento
Dicas com custo	Mecânica	Até três dicas, com penalidade de 3 minutos	Autonomia
Desbloqueio	Componente	Compartimentos físicos com pistas	Curiosidade e recompensa
Placar	Componente	Tempo restante e dicas usadas ao final	Competência

Fonte: elaborado pelos autores (2026), com base em Werbach e Hunter (2012) e Ryan e Deci (2000).

5.5 Equipamentos

Os requisitos do artefato estão no Quadro 6, cada um ligado à forma como é verificado. Os casos de teste citados estão no Quadro 8, na seção 6.

Quadro 6 – Requisitos e forma de verificação

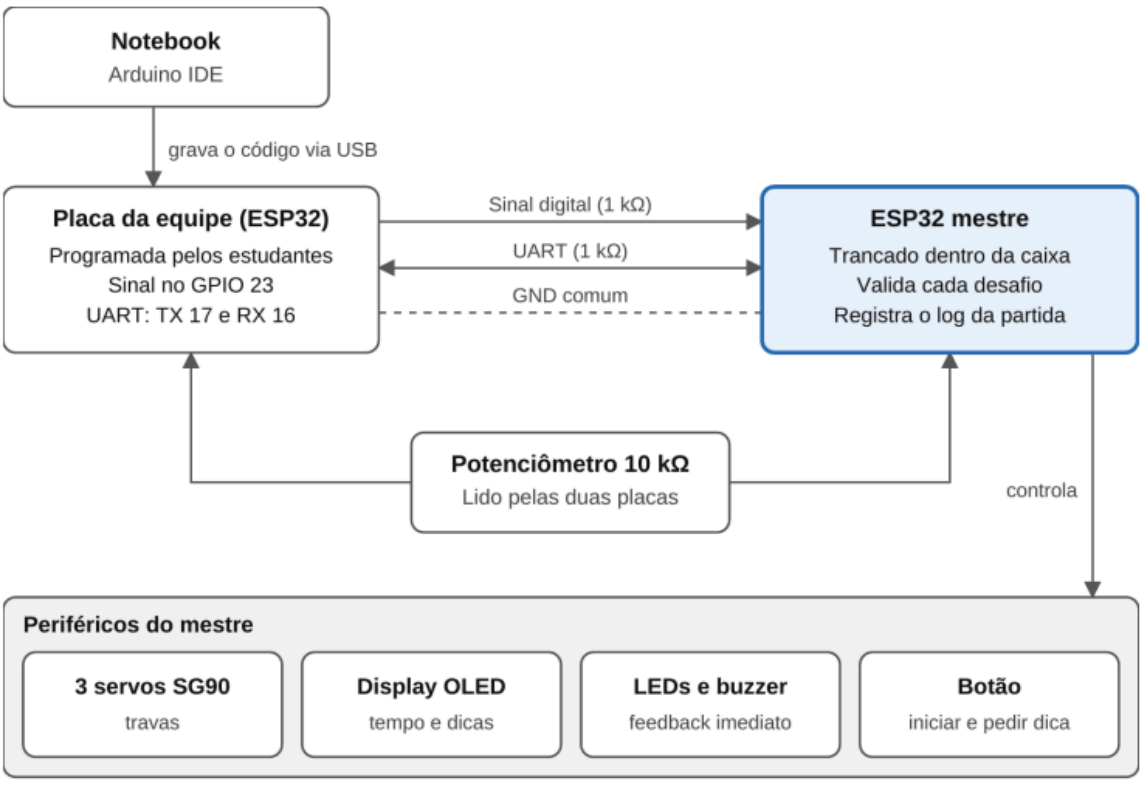
Código	Requisito	Verificação
RF01	Controlar três travas, liberadas em sequência	TF02, TF04, TF06
RF02	Validar cada desafio automaticamente, pelo comportamento elétrico da placa da equipe	TF02 a TF07, TA03, TA04
RF03	Exibir temporizador regressivo de 45 minutos	TF09
RF04	Oferecer até três dicas, com penalidade de 3 minutos	TF08
RF05	Dar feedback visual a cada tentativa e sonoro a cada erro detectado	TF03, TF05
RF06	Registrar um log persistente de cada partida	TA05
RF07	Permitir modo de manutenção e reinício pelo professor	TF01, TF10
RNF01	Usar componentes comuns, vendidos no varejo de eletrônica	Inspeção da lista de materiais (Quadro 7)
RNF02	Ser portátil e exigir apenas tomada e notebook	Protótipo físico (trabalho futuro)

Código	Requisito	Verificação
RNF03	Resistir a erros de programação e de ligação sem danos ao hardware	TA01 (lógica) em simulação; TA02 (proteção elétrica) no protótipo físico
RNF04	Permitir uma partida completa em uma aula de 60 a 90 minutos	Avaliação com participantes (seção 7)
RNF05	Ser reprodutível, com código e roteiro publicados	Inspeção do repositório

Fonte: elaborado pelos autores (2026).

A principal decisão de arquitetura foi usar dois microcontroladores. Se a equipe gravasse seu código no mesmo ESP32 que controla o jogo, apagaria o firmware da caixa a cada tentativa. Por isso, a lógica do jogo fica em um ESP32 mestre, trancado na caixa, e a equipe programa uma segunda placa, exposta em um painel externo. As duas se ligam por uma linha de sinal digital, um par UART e a derivação do potenciômetro, com terra comum (Figura 1). O mestre atua, assim, como avaliador da placa da equipe.

Figura 1 – Arquitetura da Escape Box



Fonte: elaborado pelos autores (2026).

Os componentes estão no Quadro 7. Resistores de 1 kΩ em série nas linhas entre as placas limitam a corrente caso a equipe configure como saída um pino que deveria ser entrada (RNF03). Os servos são desligados do sinal após cada

movimento, e um capacitor de 470 µF está previsto para absorver os picos de corrente que poderiam reiniciar o mestre.

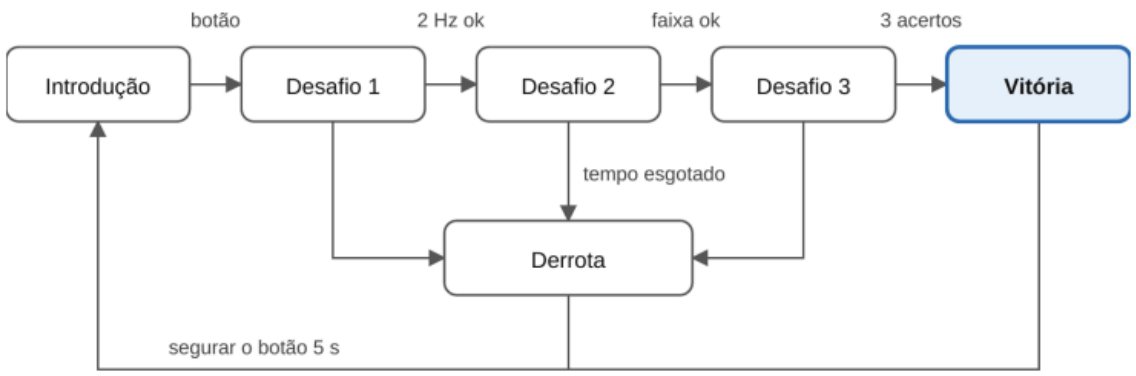
Quadro 7 – Componentes de hardware

Componente	Qtd.	Função
ESP32 DevKit V1	2	Mestre e placa da equipe
Servomotor SG90	3	Travas dos compartimentos
Display OLED 0,96" I2C	1	Temporizador, objetivo, status e dicas
Buzzer ativo	1	Feedback sonoro
LEDs verde e vermelho	2	Feedback visual
Resistores 220 Ω	2	Limitação de corrente dos LEDs
Potenciômetro 10 kΩ	1	Entrada analógica do desafio 2, lida pelas duas placas
Botão táctil	1	Iniciar, pedir dicas e reiniciar
Resistores 1 kΩ	4	Proteção das linhas entre as placas
Capacitor 470 µF	1	Estabilização da alimentação dos servos
Fonte 5 V / 2 A	1	Alimentação do mestre e dos servos
Protoboard e jumpers	1 conjunto	Ligações entre as placas e os periféricos
Cabo USB	1	Programação da placa da equipe pelo notebook
Caixa em MDF	1	Estrutura com três compartimentos

Fonte: elaborado pelos autores (2026).

O firmware do mestre foi escrito em C++ no Arduino IDE e organizado como máquina de estados: introdução, desafios 1 a 3, vitória e derrota (Figura 2). Em qualquer desafio, o esgotamento do tempo leva à derrota, e o botão exibe uma dica e desconta 3 minutos. Vitória e derrota só voltam à introdução quando o professor mantém o botão pressionado por 5 segundos, o que evita reinícios acidentais.

Figura 2 – Máquina de estados do firmware do mestre



Fonte: elaborado pelos autores (2026).

O mestre registra cada partida em CSV, com tempo decorrido, evento e detalhe, no monitor serial e na memória flash. A gravação na flash resolve um problema do hardware: nas placas DevKit, abrir o monitor serial reinicia o ESP32, então o log não pode depender do monitor aberto durante a partida. Ao final, o professor recupera o log por um comando serial. Erros são registrados por episódio, uma vez quando começam, e o episódio só se encerra quando a equipe volta a acertar de forma sustentada; assim, a amostragem de 50 ms do desafio 2 não gera dezenas de linhas por segundo. O log descreve o comportamento da equipe com o artefato e, sozinho, não é evidência de aprendizagem individual.

5.6 Avaliação e reinício

A última etapa do EscapED pede testar o jogo antes da aplicação, refletir com os jogadores, avaliar os objetivos de aprendizagem, ajustar o projeto e manter uma lista de reinício (Clarke *et al.*, 2017). O teste antes da aplicação corresponde à demonstração em simulação (seção 6), e as demais atividades compõem o plano da seção 7. Para o reinício, o modo de manutenção abre todas as travas ao ligar a caixa com o botão pressionado, para recolocar as pistas, e a partida seguinte só começa com as travas fechadas.

6 DEMONSTRAÇÃO EM SIMULAÇÃO

Os 14 casos de teste executados em simulação tiveram o resultado esperado na versão final do firmware, depois da correção de três defeitos que a primeira execução revelou. Nos casos com código correto, a placa da equipe rodou o próprio gabarito publicado no repositório. A demonstração corresponde à avaliação artificial da DSR (Venable; Pries-Heje; Baskerville, 2016): verifica a lógica do artefato sem usuários reais.

6.1 Configuração

A simulação foi feita no Wokwi (Wokwi, 2026), simulador online que reproduz o ESP32 com GPIO, interrupções, ADC e UART, além de servos, display OLED, LEDs, buzzer, botão e potenciômetro. Como o simulador não executa dois microcontroladores no mesmo projeto, o firmware do mestre e o código da equipe rodaram em um único ESP32 DevKit V1 simulado: o mestre no laço principal do programa e a placa da equipe em uma tarefa separada do FreeRTOS. Fios externos

no diagrama ligam as duas partes como no hardware real: o pino 23 da equipe ao pino 18 do mestre, o par UART cruzado entre os pinos 4 e 5 da equipe e os pinos 16 e 17 do mestre, e o potenciômetro às entradas analógicas das duas partes (Figura 3).

O código do mestre é o mesmo do artefato, acrescido de dois comandos de teste pelo monitor serial. Os dígitos 0 a 8 escolhem o comportamento simulado da equipe, entre soluções corretas e incorretas de cada desafio, e a letra T reduz o tempo restante para 10 s, para testar a derrota sem esperar 45 minutos. Nos casos corretos, a tarefa da equipe executa o código do gabarito publicado no repositório, sem alteração no corpo dos desafios; a única adaptação é a porta serial, porque na simulação os pinos 16 e 17 pertencem ao mestre. Os comportamentos errados, como piscar a 1 Hz ou responder sempre o mesmo texto, foram escritos para os testes. Para que o log em LittleFS funcionasse, o projeto incluiu uma tabela de partições com área de arquivos, ausente na configuração padrão do simulador. Essa configuração verifica a lógica dos desafios, mas não o isolamento elétrico entre as placas. O projeto da simulação está disponível em https://wokwi.com/projects/480636863090165761.

Figura 3 – Simulação no Wokwi: circuito e telas do display



Fonte: elaborado pelos autores (2026).

6.2 Casos de teste

O Quadro 8 lista os casos funcionais (TF), que conferem o comportamento esperado, e os adversariais (TA), que tentam superar os desafios por atalhos ou provocar falhas.

Quadro 8 – Casos de teste e resultados

ID	Caso	Resultado esperado	Obtido
TF01	Ligar com o botão pressionado	Abre as três travas; novo clique trava	Conforme: tela de manutenção com as três travas abertas; o clique seguinte as fechou
TF02	Gabarito do desafio 1 na placa da equipe	Etapa 1 concluída após cerca de 3 s	Conforme: etapa concluída 3,5 s após o início do sinal
TF03	Piscar a 1 Hz ou manter o pino fixo em nível alto	Etapa 1 não conclui; feedback de erro; um episódio no log	Conforme: o pino fixo gerou o registro de sinal parado e a tela indicou “Sem oscilacao”, sem bip; o 1 Hz acendeu o LED vermelho e foi registrado no mesmo episódio
TF04	Gabarito do desafio 2, girando o potenciômetro	Etapa 2 concluída	Conforme: etapa concluída após 20 amostras corretas fora da faixa e 20 dentro
TF05	Pino fixo em nível alto no desafio 2	Progresso zerado; feedback de erro	Conforme: erro_d2 fora_high; progresso zerado
TF06	Gabarito do desafio 3	Três acertos seguidos; tampa abre	Conforme: três respostas decifradas em sequência; tampa aberta e vitória registrada
TF07	Resposta errada ou após 3 s	Contador de acertos zerado; erro com código esperado e recebido no log	Conforme: resposta errada registrada com os códigos esperado e recebido; resposta após 3,5 s registrada como atraso
TF08	Pedir dica	Exibe a dica e desconta 3 minutos	Conforme: dica registrada; o tempo restante final, 2455 s, corresponde a 2700 s menos 64,9 s de partida e 180 s de penalidade
TF09	Esgotar o tempo, com tempo reduzido para o teste	Tela de derrota e evento no log	Conforme: tela de derrota e evento derrota 10 s após o comando T
TF10	Segurar o botão 5 s na tela final	Travas fecham; volta à introdução	Conforme: três travas fechadas e tela de introdução
TA01	Pino de sinal configurado como entrada, sem código	Etapa 1 não avança	Conforme: etapa 1 não avançou e nada foi registrado
TA03	Desafio 2 com o pino alternando rápido, sem ler o potenciômetro	Progresso zera; um episódio de erro por vez no log	Conforme após correção: nenhum episódio novo, só continuações a cada 5 s; antes, um registro a cada 0,5 s
TA04	Desafio 3 respondendo sempre o mesmo texto	Nenhum acerto registrado	Conforme: nenhum acerto em 11 s
TA05	Após a partida, enviar o comando de leitura do log	Log completo da partida impresso	Conforme: o comando L imprimiu a partida atual e o comando A, a anterior

Fonte: elaborado pelos autores (2026).

O caso TA02, que liga os dois pinos como saída em níveis opostos para conferir a

limitação de corrente pelos resistores, depende do hardware real e fica para o protótipo físico.

6.3 Resultados

A primeira execução dos testes revelou três defeitos no firmware do mestre, todos ligados ao registro de erros. No desafio 1, quando o sinal voltava a oscilar depois de uma pausa, a primeira medida cobria a pausa inteira e gerava um erro falso, com frequência calculada de 0,06 Hz; o defeito apareceu na passagem do pino fixo do TF03 para o sinal correto do TF02. Ainda no TF03, o pino fixo em nível alto não gerava registro no log, ao contrário do esperado. No desafio 2, um pino que alternava sem ler o potenciômetro acertava parte das amostras, e cada acerto encerrava o episódio de erro, o que produzia um registro a cada 0,5 s no caso TA03.

As correções foram pontuais. Períodos acima de 2 s passaram a reiniciar a medida, em vez de contar como erro. Um sinal que oscilou e depois ficou parado por mais de 2 s passou a gerar um registro de sinal parado, sem alarme sonoro, porque no início da etapa o sinal parado é normal. Um episódio passou a se encerrar só com acerto sustentado: dois ciclos corretos seguidos no desafio 1 e dez amostras corretas seguidas, ou 0,5 s, no desafio 2. Um erro que persiste volta ao log no máximo a cada 5 s, marcado como continuação, sem contar como episódio novo. Na segunda execução, com os casos corretos rodando o código do gabarito, os 14 casos do Quadro 8 tiveram o resultado esperado. Esse retorno da avaliação ao projeto é o ciclo previsto na DSRM (Peffers *et al.*, 2007).

A simulação também levou a um ajuste de interface: mensagens de status com mais de 21 caracteres quebravam linha no display e foram encurtadas. O Quadro 9 mostra o log de uma partida de teste vencida, com erros provocados nos três desafios e uma dica. Os tempos refletem o roteiro de teste, e não o desempenho de uma equipe. O resultado de cada caso e os logs completos estão repositório.

Quadro 9 – Log de uma partida de teste simulada

Tempo (s)	Evento	Detalhe
0,0	inicio	
6,1	erro_d1	sem_oscilacao
11,2	erro_d1	1.00;continua

Tempo (s)	Evento	Detalhe
16,2	erro_d1	1.00;continua
20,7	dica	1-1
21,2	erro_d1	1.00;continua
26,2	etapa_concluida	etapa=1;duracao_s=26;episodios_erro=1
27,2	erro_d2	fora_high
32,3	erro_d2	fora_high;continua
37,3	erro_d2	dentro_low;continua
43,2	etapa_concluida	etapa=2;duracao_s=16;episodios_erro=1
43,9	erro_d3	esperado=ZXEHA;recebido=ABCDE
53,9	erro_d3	esperado=HENLT;recebido=ABCDE;continua
62,4	erro_d3	atraso;continua
64,1	etapa_concluida	etapa=3;duracao_s=20;episodios_erro=1
64,9	vitoria	restante_s=2455;dicas=1

Fonte: dados da simulação (2026).

7 PLANO DE AVALIAÇÃO COM PARTICIPANTES

A avaliação naturalista, prevista para quando o protótipo físico estiver construído, terá a forma de um teste-piloto com equipes de três estudantes do público-alvo definido em 5.1. O Quadro 10 resume as questões, os instrumentos e os critérios. Os indicadores e critérios foram definidos antes de qualquer coleta, para que os resultados não orientem a escolha do que medir.

Quadro 10 – Questões, instrumentos e critérios da avaliação planejada

Questão	Instrumento	Indicador e critério
O jogo cabe em uma aula? (RNF04)	Log do artefato	Proporção de equipes que concluem em 45 minutos; meta de pelo menos metade
Onde as equipes travam?	Log e roteiro de observação	Duração, episódios de erro e dicas por desafio, descritos por equipe
A experiência é bem avaliada?	MEEGA+ (Petri; Gresse von Wangenheim; Borgatto, 2019)	Mediana dos itens de experiência do jogador e usabilidade; meta de concordância (mediana de 4 ou mais na escala de 5 pontos)
Os estudantes percebem aprendizagem?	Itens de percepção de aprendizagem do MEEGA+ e perguntas abertas	Mediana por objetivo de aprendizagem e temas recorrentes nas respostas
Há ganho de aprendizagem?	Pré e pós-teste curto sobre os três conteúdos	Diferença entre as notas, como em López-Pernas *et al.* (2019b) e Gordillo e López-Fernández (2024)

Fonte: elaborado pelos autores (2026).

O MEEGA+ é um modelo validado para avaliar jogos educacionais de computação pela percepção dos estudantes, com itens em escala Likert de cinco pontos (Petri; Gresse von Wangenheim; Borgatto, 2019), e já foi adaptado a escape rooms físicos (Costa *et al.*, 2020). Os dados ordinais serão resumidos por frequências e medianas. O log e o pré e pós-teste respondem a perguntas diferentes: o primeiro descreve o comportamento da equipe com o artefato, e apenas o segundo mede aprendizagem individual.

Quanto à ética, a participação será voluntária, restrita a maiores de 18 anos, sem vínculo de avaliação com o pesquisador e sem coleta de dados que identifiquem os participantes. Todos assinarão um Termo de Consentimento Livre e Esclarecido que autorize também a divulgação dos resultados em publicações científicas. O protocolo será submetido ao sistema CEP/Conep antes da coleta. Como o plano inclui um pré e pós-teste de conhecimento, e não apenas uma pesquisa de opinião, não se assume a dispensa prevista na Resolução CNS nº 510/2016 para pesquisas de opinião com participantes não identificados (Brasil, 2016).

Mesmo com esse plano, um piloto de sessão única, sem grupo de comparação, não separa o efeito da programação como mecânica de outros fatores, como a novidade, o trabalho em equipe ou a computação física em si. Suas conclusões deverão se limitar à viabilidade, à aceitação e à aprendizagem percebida, com o pré e pós-teste como indicativo inicial.

8 CONSIDERAÇÕES FINAIS

Este artigo apresentou o projeto de uma Escape Box controlada por ESP32 em que programar um microcontrolador é a mecânica de progressão e o próprio artefato verifica o comportamento físico do código. Retomando os objetivos específicos:

- **OE1** foi atendido com a seleção de saída digital, entrada analógica e comunicação serial (seção 5.2).
- **OE2** foi atendido com a fundamentação na aprendizagem ativa, no construcionismo e na Teoria da Autodeterminação, e com o mapeamento dos elementos de jogo (seções 2, 5.3 e 5.4).

- **OE3** foi atendido com três desafios progressivos, cujos critérios de verificação resistem às soluções triviais testadas (seção 5.4).
- **OE4** foi atendido com a especificação da arquitetura de dois microcontroladores e a implementação do firmware (seção 5.5).
- **OE5** foi atendido com a demonstração em simulação, em que os 14 casos de teste tiveram o resultado esperado após a correção de três defeitos (seção 6), e com o plano de avaliação da seção 7.

A principal contribuição é a arquitetura em que um microcontrolador isolado do código dos estudantes atua como avaliador automático do comportamento elétrico desse código. Ela leva a programação como mecânica de jogo, já usada em escape rooms de programação web, para o domínio dos sistemas embarcados, em um formato projetado para ser portátil e caber em uma aula, o que ainda depende do protótipo físico e da avaliação com participantes.

8.1 Limitações

- O artefato não foi construído fisicamente; a demonstração em simulação verifica a lógica, mas não o comportamento elétrico real, a temporização do hardware nem a durabilidade das travas.
- A simulação reuniu as duas lógicas em um único ESP32 e não verificou o isolamento elétrico entre as placas nem a limitação de corrente pelos resistores (caso TA02).
- Como as duas partes leram o mesmo conversor analógico-digital, a zona de tolerância de 150 unidades do desafio 2, que compensa diferenças entre os conversores de duas placas, não foi testada.
- Os comportamentos errados da equipe foram escritos pelos autores, e a cobertura adversarial se limita aos atalhos previstos; cada caso rodou uma vez na versão final do firmware. O código-modelo entregue aos estudantes não foi executado, porque depende do que cada equipe escreve.
- O desenho pedagógico dos desafios não foi validado por professores ou especialistas da área.
- Não houve avaliação com estudantes, então não há evidência sobre engajamento ou aprendizagem proporcionados pelo artefato.

- A busca por trabalhos relacionados foi exploratória, e não uma revisão sistemática.

8.2 Trabalhos futuros

- Construir o protótipo físico e repetir os casos de teste em bancada, incluindo os elétricos e a calibração da zona de tolerância do desafio 2 com as duas placas.
- Validar o desenho dos desafios com professores da disciplina antes do piloto.
- Aplicar o plano de avaliação da seção 7 e, depois, um estudo com grupo de comparação.
- Ampliar os desafios para PWM, interrupções e I2C.
- Usar o Wi-Fi do ESP32 para um placar entre equipes.

REFERÊNCIAS

BARR, Michael; MASSA, Anthony. **Programming embedded systems**: with C and GNU development tools. 2. ed. Sebastopol: O'Reilly Media, 2006.

BONWELL, Charles C.; EISON, James A. **Active learning**: creating excitement in the classroom. Washington, D.C.: George Washington University, 1991. (ASHE-ERIC Higher Education Report, n. 1).

BRASIL. Conselho Nacional de Saúde. **Resolução nº 510, de 7 de abril de 2016**. Dispõe sobre as normas aplicáveis a pesquisas em Ciências Humanas e Sociais. Diário Oficial da União: seção 1, Brasília, DF, p. 44-46, 24 maio 2016.

CASTRO, Felipe *et al.* LogicScape: um jogo educacional baseado em escape room para o ensino de portas lógicas. *In*: SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL, 25., 2026, Goiânia. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2026. p. 2427-2437. DOI: 10.5753/sbgames.2026.26359.

CASTRO, Ronney Moreira de; CLASSE, Tadeu Moreira de; OLIVEIRA, Eduardo Gomes de. Escape room educacional como estratégia de aprendizagem ativa em sistemas de informação: percepção dos estudantes. *In*: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO, 34., 2026, Gramado. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2026. p. 698-710. DOI: 10.5753/wei.2026.20162.

CLARKE, Samantha *et al.* EscapED: a framework for creating educational escape rooms and interactive games for higher/further education. **International Journal of Serious Games**, v. 4, n. 3, p. 73-86, 2017. DOI: 10.17083/ijsg.v4i3.180.

CLASSE, Tadeu Moreira de; CASTRO, Ronney Moreira de; OLIVEIRA, Eduardo Gomes de. Escape room educacional digital como ferramenta de aprendizado na disciplina de banco de dados. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 36., 2025, Curitiba. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2025. p. 550-563. DOI: 10.5753/sbie.2025.12533.

COSTA, Guilherme S. *et al.* Kalinin II: relato de experiência de um escape room aplicado no ambiente educacional. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 31., 2020, Online. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 481-490. DOI: 10.5753/cbie.sbie.2020.481.

DETERDING, Sebastian *et al.* From game design elements to gamefulness: defining "gamification". *In*: INTERNATIONAL ACADEMIC MINDTREK CONFERENCE, 15., 2011, Tampere. **Proceedings** [...]. New York: ACM, 2011. p. 9-15. DOI: 10.1145/2181037.2181040.

DRESCH, Aline; LACERDA, Daniel Pacheco; ANTUNES JÚNIOR, José Antônio Valle. **Design Science Research**: método de pesquisa para avanço da ciência e tecnologia. Porto Alegre: Bookman, 2015.

ESPRESSIF SYSTEMS. **ESP32 series datasheet**. Versão 5.3. Shanghai: Espressif Systems, 2026. Disponível em: https://documentation.espressif.com/esp32_datasheet_en.pdf. Acesso em: 14 nov. 2026.

FREEMAN, Scott *et al.* Active learning increases student performance in science, engineering, and mathematics. **Proceedings of the National Academy of Sciences**, v. 111, n. 23, p. 8410-8415, 2014. DOI: 10.1073/pnas.1319030111.

GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 6. ed. São Paulo: Atlas, 2017.

GORDILLO, Aldo; LÓPEZ-FERNÁNDEZ, Daniel. Are educational escape rooms more effective than traditional lectures for teaching software engineering? A randomized controlled trial. **IEEE Transactions on Education**, v. 67, n. 5, p. 660-668, 2024. DOI: 10.1109/TE.2024.3403913.

HAMARI, Juho; KOIVISTO, Jonna; SARSA, Harri. Does gamification work? A literature review of empirical studies on gamification. *In*: HAWAII INTERNATIONAL CONFERENCE ON SYSTEM SCIENCES, 47., 2014, Waikoloa. **Proceedings** [...]. Washington, D.C.: IEEE, 2014. p. 3025-3034. DOI: 10.1109/HICSS.2014.377.

HEVNER, Alan R. A three cycle view of design science research. **Scandinavian Journal of Information Systems**, v. 19, n. 2, p. 87-92, 2007.

HEVNER, Alan R. *et al.* Design science in information systems research. **MIS Quarterly**, v. 28, n. 1, p. 75-105, 2004. DOI: 10.2307/25148625.

KAPP, Karl M. **The gamification of learning and instruction**: game-based methods and strategies for training and education. San Francisco: Pfeiffer, 2012.

KOLB, David A. **Experiential learning**: experience as the source of learning and development. Englewood Cliffs: Prentice-Hall, 1984.

LÓPEZ-PERNAS, Sonsoles *et al.* Examining the use of an educational escape room for teaching programming in a higher education setting. **IEEE Access**, v. 7, p. 31723-31737, 2019a. DOI: 10.1109/ACCESS.2019.2902976.

LÓPEZ-PERNAS, Sonsoles *et al.* Analyzing learning effectiveness and students' perceptions of an educational escape room in a programming course in higher education. **IEEE Access**, v. 7, p. 184221-184234, 2019b. DOI: 10.1109/ACCESS.2019.2960312.

LUBYAGIN, Igor; IVANOVA, Anna; LOGINOV, Sergei. Microcontrollers in automation systems of escape rooms. *In*: HUMAN. ENVIRONMENT. TECHNOLOGIES: STUDENTS INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE, 21., 2017, Rēzekne. **Proceedings** [...]. Rēzekne: Rezekne Academy of Technologies, 2017. p. 164-169. DOI: 10.17770/het2017.21.3559.

MARWEDEL, Peter. **Embedded system design**: embedded systems foundations of cyber-physical systems, and the Internet of Things. 4. ed. Cham: Springer, 2021.

NICHOLSON, Scott. **Peeking behind the locked door**: a survey of escape room facilities. [*S. l.*: *s. n.*], 2015. Disponível em: http://scottnicholson.com/pubs/erfacwhite.pdf. Acesso em: 20 maio 2026.

NICHOLSON, Scott; CABLE, Liz. **Unlocking the potential of puzzle-based learning**: designing escape rooms and games for the classroom. London: Corwin, 2021.

O'SULLIVAN, Dan; IGOE, Tom. **Physical computing**: sensing and controlling the physical world with computers. Boston: Thomson Course Technology, 2004.

PAPERT, Seymour. **Mindstorms**: children, computers, and powerful ideas. New York: Basic Books, 1980.

PEFFERS, Ken *et al.* A design science research methodology for information systems research. **Journal of Management Information Systems**, v. 24, n. 3, p. 45-77, 2007. DOI: 10.2753/MIS0742-1222240302.

PETRI, Giani; GRESSE VON WANGENHEIM, Christiane; BORGATTO, Adriano Ferreti. MEEGA+: um modelo para a avaliação de jogos educacionais para o ensino de computação. **Revista Brasileira de Informática na Educação**, v. 27, n. 3, p. 52-81, 2019. DOI: 10.5753/rbie.2019.27.03.52.

PFEIFER, Marc *et al.* Teaching embedded systems by constructing an escape room. *In*: ACM TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION, 52., 2021, Online. **Proceedings** [...]. New York: ACM, 2021. p. 1103-1109. DOI: 10.1145/3408877.3432485.

PRINCE, Michael. Does active learning work? A review of the research. **Journal of Engineering Education**, v. 93, n. 3, p. 223-231, 2004. DOI: 10.1002/j.2168-9830.2004.tb00809.x.

PRODANOV, Cleber Cristiano; FREITAS, Ernani Cesar de. **Metodologia do trabalho científico**: métodos e técnicas da pesquisa e do trabalho acadêmico. 2. ed. Novo Hamburgo: Feevale, 2013.

RYAN, Richard M.; DECI, Edward L. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. **American Psychologist**, v. 55, n. 1, p. 68-78, 2000. DOI: 10.1037/0003-066X.55.1.68.

SAILER, Michael; HOMNER, Lisa. The gamification of learning: a meta-analysis. **Educational Psychology Review**, v. 32, n. 1, p. 77-112, 2020. DOI: 10.1007/s10648-019-09498-w.

SWELLER, John. Cognitive load during problem solving: effects on learning. **Cognitive Science**, v. 12, n. 2, p. 257-285, 1988. DOI: 10.1207/s15516709cog1202_4.

TRACK, Mateus Bonfim *et al.* Jogos educativos com Arduino no ensino básico e técnico: uma avaliação de engajamento e aprendizado. *In*: SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL, 25., 2026, Goiânia. **Anais** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2026. p. 2073-2084. DOI: 10.5753/sbgames.2026.25904.

VASCONCELOS, Beatriz Loyola Gomes de *et al.* EXIT: um jogo de puzzle como recurso de apoio ao ensino de lógica proposicional. *In*: WORKSHOP MAGICA: GAMES NA GRADUAÇÃO E NA EDUCAÇÃO BÁSICA, SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL, 25., 2026, Goiânia. **Anais Estendidos** [...]. Porto Alegre: Sociedade Brasileira de Computação, 2026. p. 651-656. DOI: 10.5753/sbgames_estendido.2026.32769.

VENABLE, John; PRIES-HEJE, Jan; BASKERVILLE, Richard. FEDS: a framework for evaluation in design science research. **European Journal of Information Systems**, v. 25, n. 1, p. 77-89, 2016. DOI: 10.1057/ejis.2014.36.

WERBACH, Kevin; HUNTER, Dan. **For the win**: how game thinking can revolutionize your business. Philadelphia: Wharton Digital Press, 2012.

WOKWI. **Wokwi**: ESP32, STM32, Arduino simulator. [*S. l.*], 2026. Disponível em: https://wokwi.com. Acesso em: 15 nov. 2026.