

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA -  
CAMPUS CAMPINA GRANDE  
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

ANDERSON LUCAS  
DIOGO SANTOS  
GABRIEL LOPES  
JACKSON PLATINY

**SISTEMAS EMBARCADOS**  
**PROJETO FINAL**

Campina Grande  
2025

## INTRODUÇÃO

Este projeto consiste no desenvolvimento de um sistema embarcado de jogos interativos utilizando um sensor de movimento MPU6050, display OLED SSD1306, botões físicos para navegação e um módulo SD Card para armazenamento persistente de dados. O sistema integra múltiplos recursos de entrada e saída, comunicação I2C e SPI, além de processamento em tempo real no microcontrolador ESP32.

O objetivo principal é criar uma plataforma com quatro minigames controlados por inclinação, oferecendo interação física e experiência de jogo intuitiva. O sistema inicia com um menu de seleção e permite que o usuário navegue entre os jogos e selecione a opção desejada utilizando dois botões físicos. Os recordes de cada jogo são armazenados no cartão SD, garantindo persistência mesmo após desligar o sistema.

O projeto também explora conceitos de sistemas embarcados como multitarefas, debounce de botões, PWM para sons simulados e comunicação com múltiplos periféricos.

Funcionamento:

O sistema embarcado inicia exibindo um menu principal no display OLED com as quatro opções de jogos. O Botão 1 é utilizado para navegar entre as opções, enquanto o Botão 2 confirma a seleção.

Cada jogo implementa uma lógica independente, controlada por inclinação, e possui um sistema de pontuação. Ao final da partida, se o jogador atingir um recorde, este é salvo no SD Card.

Lista de Jogos:

1. Dodge the Blocks – Desviar de obstáculos com movimento lateral.
2. Tilt Maze – Conduzir uma esfera virtual por um labirinto.
3. Snake Tilt – Versão do clássico Snake controlada por inclinação.
4. Paddle Pong – Rebater a bola com uma raquete controlada por inclinação.

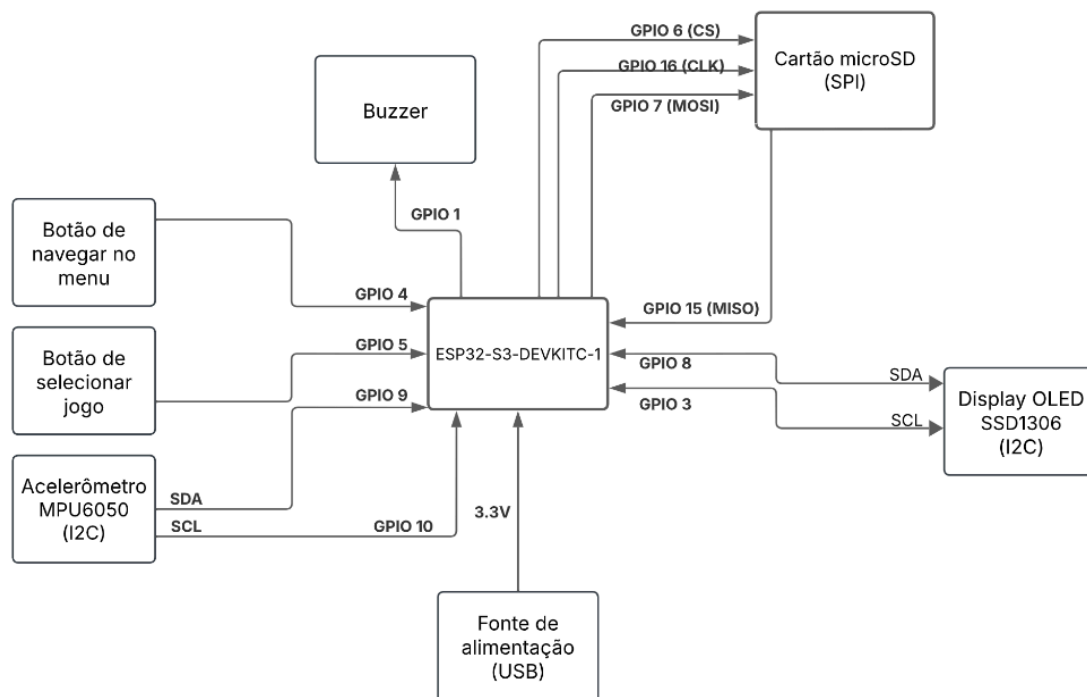
Componentes:

| Descrição      | Função                          | Quantidade |
|----------------|---------------------------------|------------|
| ESP32          | MCU Principal                   | 1          |
| MPU6050        | Controle por movimento (I2C)    | 1          |
| SSD1306        | Interface gráfica (I2C)         | 1          |
| Módulo SD Card | Armazenamento persistente (SPI) | 1          |
| Buzzer         | Efeitos sonoros (PWM)           | 1          |

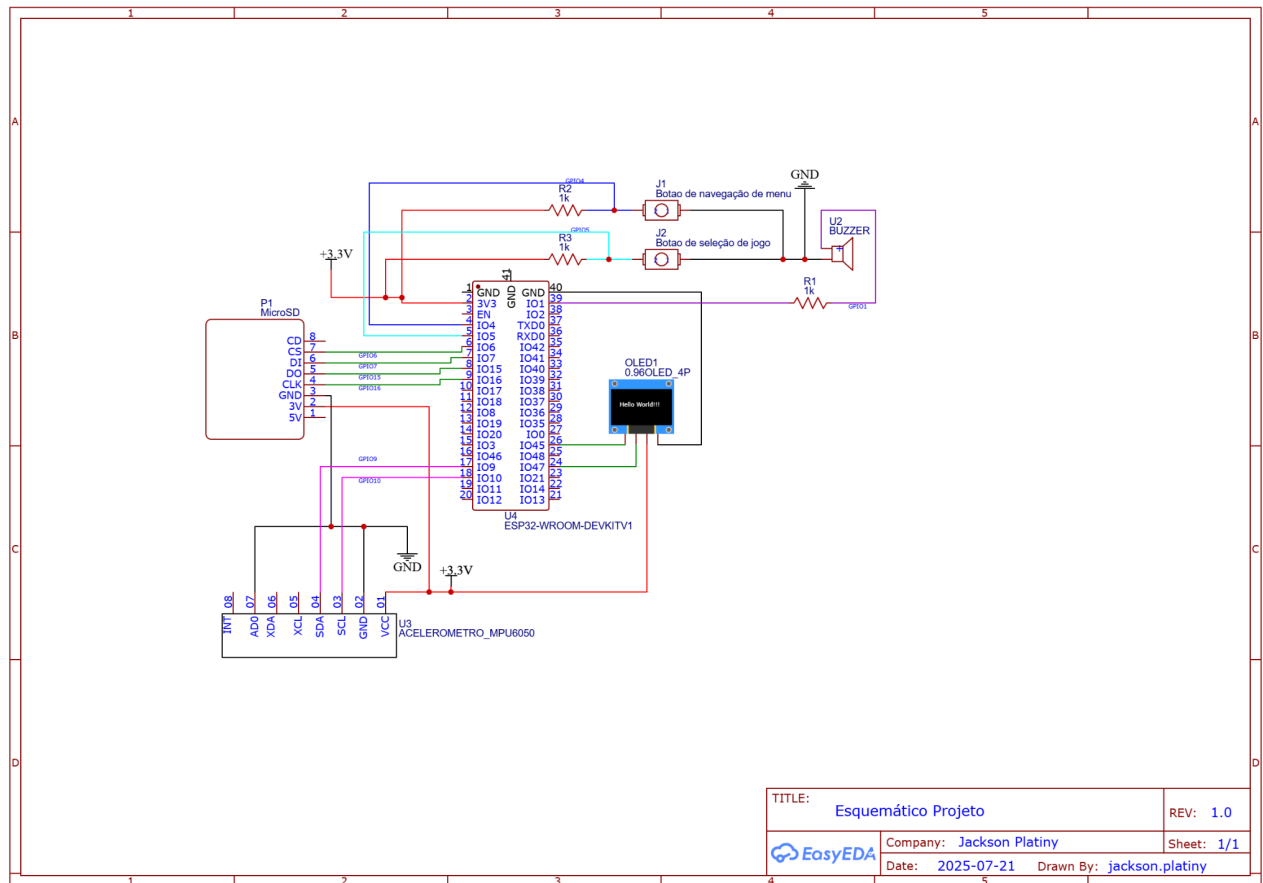
|        |  |   |
|--------|--|---|
| Botões | Navegação no menu e seleção de jogo (GPIO) | 2 |
|--------|--|---|

## 1. Hardware:

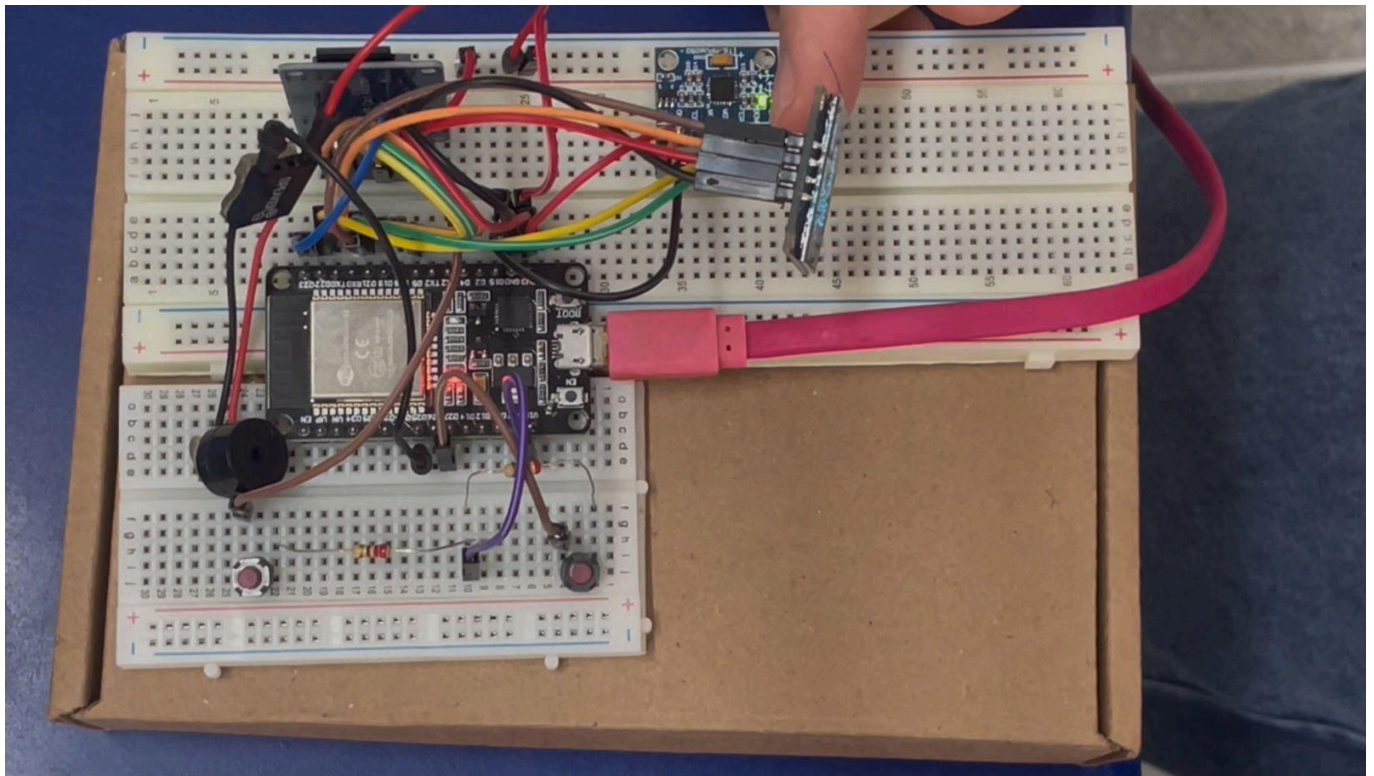
Diagrama de bloco do dispositivo:



Esquema elétrico:



Protótipo do dispositivo:



Demonstração de funcionamento do dispositivo: [Link do video](#)

## 2. Bibliotecas

|                                       |   |
|---------------------------------------|---|
| stdio.h                               | Biblioteca padrão de entrada/saída em C.  |
| string.h                              | Biblioteca padrão de manipulação de strings em C.   |
| stdlib.h                              | Biblioteca de funções utilitárias de propósito geral (ex.: rand() para posições aleatórias nos jogos).  |
| stdbool.h                             | Usada para definição do tipo booleano bool e valores true/false.  |
| stdint.h                              | Biblioteca de tipos de dados inteiros com tamanho fixo.   |
| freertos/FreeRTOS.h e freertos/task.h | Bibliotecas relacionadas ao FreeRTOS, um sistema operacional em tempo real para sistemas embarcados. Elas são usadas para criar tarefas e gerenciar o sistema de tempo real (multithread no ESP32). |
| driver/gpio.h                         | Biblioteca para operações GPIO no ESP-IDF. Usada para configurar e controlar pinos GPIO.  |
| driver/ledc.h                         | Biblioteca para controle de PWM via módulo LEDC para gerar tons no buzzer.  |
| driver/i2c.h                          | Biblioteca para comunicação I2C com MPU6050 (acelerômetro) e SSD1306 (display OLED).  |
| driver/sdmmc_types.h                  | Biblioteca que contém estruturas e definições para interface com cartão SD.   |

|             |   |
|-------------|---|
| esp_log.h   | Biblioteca de registro de logs para depuração durante a execução.   |
| esp_err.h   | Biblioteca para a manipulação de códigos de erro e status de funções da ESP-IDF.                                    |
| "font5x7.h" | Biblioteca de fonte bitmap personalizada desenvolvida pela equipe do projeto para renderizar caracteres no display. |

|   |   |
|---|---|
| "display.h"   | Biblioteca desenvolvida pela equipe do projeto que contém as funções próprias para inicializar e desenhar no display OLED SSD1306.                  |
| "mpu6050.h"   | Biblioteca desenvolvida pela equipe do projeto que contém as funções próprias para inicialização e leitura de dados do sensor de movimento MPU6050. |
| "sdcard.h"  | Biblioteca desenvolvida pela equipe do projeto que contém as funções próprias para inicializar, ler e gravar dados no cartão SD (registros).        |
| Essas bibliotecas são essenciais para o desenvolvimento do sistema embarcado de jogos no ESP32 utilizando o framework ESP-IDF, proporcionando funcionalidades como manipulação de GPIO, comunicação I2C e SPI, controle de PWM para geração de sons, gerenciamento de tarefas com FreeRTOS, exibição gráfica no display OLED, leitura de dados do sensor MPU6050 e armazenamento persistente de informações no cartão SD. |   |

## DESCRIÇÃO DAS ATIVIDADES

Este projeto visa o desenvolvimento de um sistema embarcado de jogos com quatro minigames controlados via sensor de movimento MPU6050, com interface gráfica em display OLED SSD1306, controle de menu por botões físicos e armazenamento persistente de dados em SDCard. O sistema deve iniciar com um menu de seleção dos jogos, permitindo ao usuário navegar e selecionar nas opções de minigames utilizando dois botões.

Atividades Realizadas:

Configuração do Ambiente de Desenvolvimento:

- Instalação e configuração do ambiente de desenvolvimento ESP-IDF (Espressif IoT Development Framework).

Implementações e Configurações Relativas a Comunicação dos Dispositivos:

- Implementação da comunicação I2C para o MPU6050 e SSD1306.
- Implementação da comunicação SPI para o módulo SD Card.
- Desenvolvimento de bibliotecas para leitura do acelerômetro, interface gráfica, controle do buzzer e manipulação do SD Card.

Implementação dos jogos:

- Implementação dos quatro jogos independentes, cada um com sua lógica de controle por inclinação.
- Criação do menu inicial e navegação com botões físicos.

- Salvamento e recuperação de recordes de pontuação no SD Card.

Realização de Testes para Comprovação e Validação:

- Testes no hardware real com ajuste fino da jogabilidade.

### 3. Código ([Link do Repositório](#))

## RESULTADOS

Durante o desenvolvimento do projeto, obtivemos resultados positivos em várias frentes, destacando conquistas significativas em áreas-chave. A seguir, apresento uma visão geral dos resultados obtidos:

Configurações iniciais do Framework e dos dispositivos:

A configuração do framework apesar de inicialmente ter sido um pouco conturbada devido a diferenças de versão e erros de instalação, terminou sendo bem sucedida ao optar por uma versão robusta e estável, permitindo dar continuidade no desenvolvimento das ferramentas e garantindo o correto funcionamento da comunicação entre dispositivos.

Implementação dos jogos:

A implementação dos jogos teve momentos desafiadores e momentos amenos, onde alguns jogos exigiram mais tempo e esforço que outros, como foi o caso do Tilt Maze, mas de maneira geral conseguimos alcançar os objetivos estabelecidos. A interface dos jogos foi projetada seguindo à risca o esperado de cada jogo, atendendo aos requisitos do projeto. Em alguns jogos foi necessário realizar experimentações e testes quanto a velocidade de resposta e velocidade do jogo em si, para fins de encontrar a melhor experiência de jogatina para o usuário final do sistema embarcado.

Implementação das bibliotecas dos dispositivos:

A implementação das bibliotecas dos dispositivos foi uma etapa tranquila e atendeu às expectativas. Essa etapa foi essencial para garantir o correto funcionamento da comunicação entre dispositivos, conforme as necessidades do projeto.

Aplicação de testes no ambiente de simulação Wokwi:

Nesta etapa, fizemos os testes iniciais para validar a construção do nosso código, reparando erros e ajustando parâmetros para garantir que o código estivesse adequado com as diretrizes e objetivos traçados no documento do projeto e rodando sem erros. Encontramos algumas dificuldades para chegar na versão final do código, mas nada fora do esperado. Como resultado final, conseguimos desenvolver um código robusto e funcional, que possui todas as funcionalidades esperadas para o projeto.

Aplicação de testes no Hardware definitivo:

Devido a limitações na ferramenta Wokwi com relação a simulação da escrita e leitura de arquivos no cartão SD, acabamos tendo um pouco de dificuldade em acertar no hardware real as configurações adequadas que permitiam o correto salvamento dos recordes dos jogos no cartão SD, porém, após algumas modificações e testes, foi possível realizar o salvamento da maneira esperada, garantindo assim o cumprimento dos requisitos solicitados para este projeto e o funcionamento esperado desse sistema embarcado.

De maneira geral, o projeto atendeu aos objetivos propostos, resultando em um sistema funcional com os quatro minigames operacionais. A interação via MPU6050 apresentou boa responsividade e a interface gráfica no SSD1306 foi clara e objetiva. O salvamento e recuperação de recordes no SD Card, apesar das dificuldades iniciais encontradas, após pequenos ajustes acabou funcionando conforme esperado também. Os efeitos sonoros simulados com PWM complementam a experiência de jogo do usuário.

## CONSIDERAÇÕES FINAIS

Este projeto demonstrou a viabilidade de integrar múltiplos dispositivos e protocolos em um sistema embarcado compacto. O projeto pode ser expandido com novos jogos, melhorias gráficas e maior complexidade na lógica de jogabilidade. A abordagem modular adotada facilita futuras atualizações e reutilização de componentes em outros projetos de sistemas embarcados.