

## PROTÓTIPO DE SIMULADOR DE SINAIS DE ECG COM ESP32 PARA TREINAMENTO E TESTES EM ENGENHARIA CLÍNICA

Lucas Cardoso Madeira<sup>1</sup>  
Patrese Veras Quelemes<sup>2</sup>

**RESUMO:** Este trabalho apresenta o desenvolvimento e a avaliação de um simulador de sinais de eletrocardiograma (ECG) de baixo custo, alta fidelidade e fácil reproduzibilidade, voltado para aplicações em Engenharia Clínica e oficinas de manutenção de equipamentos médico-hospitalares. A solução proposta utiliza o microcontrolador ESP32, responsável pelo processamento digital das amostras provenientes de tabelas previamente definidas, e o conversor digital-analógico MCP4725, de 12 bits, que garante uma saída analógica estável, precisa e com baixa distorção, evitando limitações comuns em sistemas baseados em modulação por largura de pulso (PWM). O simulador é capaz de reproduzir seis ritmos cardíacos amplamente utilizados em testes funcionais: ritmo sinus normal, bradicardia, taquicardia (até 240 bpm), assistolia, extrassístole ventricular (PVC) e fibrilação atrial. Os sinais possuem amplitude e deslocamento ajustáveis, permitindo adequação às entradas diferenciais de monitores multiparamétricos e eletrocardiógrafos. Os resultados mostram que o dispositivo oferece desempenho confiável e adequada compatibilidade elétrica, demonstrando potencial como ferramenta de apoio à capacitação técnica, validação de equipamentos e atividades educacionais.

4121

**Palavras-chave:** Engenharia Clínica. ESP32. MCP4725. Simulador de ECG. Sinais Biomédicos.

**ABSTRACT:** This work presents the development and evaluation of a low-cost, high-fidelity electrocardiogram (ECG) signal simulator designed for Clinical Engineering applications and biomedical equipment maintenance workshops. The proposed solution uses the ESP32 microcontroller to digitally process predefined ECG waveforms and the 12-bit MCP4725 digital-to-analog converter to generate accurate and low-distortion analog output, avoiding limitations commonly found in PWM-based systems. The simulator reproduces six clinically relevant cardiac rhythms: normal sinus rhythm, bradycardia, tachycardia (up to 240 bpm), asystole, premature ventricular contraction, and atrial fibrillation. All signals feature adjustable amplitude and offset, ensuring compatibility with differential inputs of patient monitors and electrocardiographs. The results indicate that the device performs reliably and offers appropriate electrical characteristics, demonstrating its potential as a tool for technical training, equipment validation, and educational environments.

**Keywords:** Biomedical Signals. Clinical Engineering. ECG Simulator. ESP32. MCP4725.4.

<sup>1</sup> Graduando em Engenharia Elétrica, Centro Universitário UNIFSA.

<sup>2</sup> Mestre em Engenharia Elétrica, Universidade Federal de Campina Grande (UFCG). Orientador do curso de Engenharia Elétrica, Centro Universitário Santo Agostinho (UNIFSA).

## I. INTRODUÇÃO

Simuladores de sinais fisiológicos desempenham um papel essencial na Engenharia Clínica, especialmente em atividades de validação funcional, manutenção preventiva e corretiva, além do treinamento de profissionais que operam equipamentos biomédicos. Entre esses sinais, o eletrocardiograma (ECG) se destaca pela ampla utilização em ambientes hospitalares e pela sua relevância para decisões clínicas em tempo real (WEBSTER, 2010; KHANDPUR, 2014).

Os simuladores de ECG permitem testar a integridade de cabos, canais de aquisição, filtros, ganho, alarmes e resposta em frequência de monitores e eletrocardiógrafos, sem expor pacientes a riscos. Entretanto, dispositivos comerciais costumam ter custo elevado e, muitas vezes, dependem de softwares proprietários, o que dificulta sua adoção em hospitais públicos, clínicas populares e laboratórios de ensino.

Nesse contexto, oferecer um simulador didático, acessível e tecnicamente confiável representa uma alternativa de grande valor, especialmente em instituições com recursos limitados. A proposta deste trabalho é a construção de um simulador de ECG baseado em microcontrolador, capaz de gerar ritmos clínicos relevantes com controle de frequência cardíaca, amplitude, ruído e offset, respeitando as faixas de sinal de entrada previstas nas normas internacionais IEC 60601-2-25 e IEC 60601-2-27 (IEC, 2011). 4122

A ausência de simuladores acessíveis prejudica a calibração, o diagnóstico de falhas e a avaliação periódica de equipamentos essenciais para a segurança do paciente. Diante disso, o presente projeto busca desenvolver um dispositivo funcional, reproduzível e de baixo custo, que possa ser utilizado tanto por profissionais de Engenharia Clínica quanto por estudantes em formação.

O ECG é um dos parâmetros mais monitorados em unidades de terapia intensiva, centros cirúrgicos e serviços de emergência, sendo exibido continuamente em monitores multiparamétricos, desfibriladores, ventiladores e centrais de monitoramento (BRUNNER; SUDDARTH, 2018). A confiabilidade desses sistemas é determinante para o diagnóstico e a intervenção clínica imediata. Assim, manter esses equipamentos adequadamente calibrados é fundamental para a segurança do paciente e para o desempenho assistencial (ABNT, 2010).

Em muitos contextos hospitalares — sobretudo em unidades de menor porte — a carência de ferramentas específicas limita a capacidade de manutenção preventiva. Dessa forma, o desenvolvimento de um simulador de ECG simples, robusto e financeiramente acessível surge como solução viável e estratégica, permitindo que técnicos realizem testes e calibrações de forma autônoma e rotineira, ampliando a confiabilidade dos equipamentos biomédicos e democratizando o acesso a práticas adequadas de manutenção.

## 2. REFERENCIAL TEÓRICO

O referencial teórico deste trabalho fundamenta o estudo e o desenvolvimento de um simulador de eletrocardiograma (ECG) microcontrolado, abordando aspectos de eletrofisiologia cardíaca, instrumentação biomédica, segurança elétrica e sistemas embarcados.

A literatura em Engenharia Biomédica descreve um conjunto de requisitos funcionais e de segurança para sistemas geradores de sinais de ECG utilizados em testes de equipamentos médicos. Esses requisitos incluem amplitude, formato de onda, frequência cardíaca, impedância de saída, modos diferenciais, resposta em frequência e conformidade com normas técnicas como a IEC 60601-2-25 (eletrocardiógrafos) e IEC 60601-2-27 (monitores de ECG).

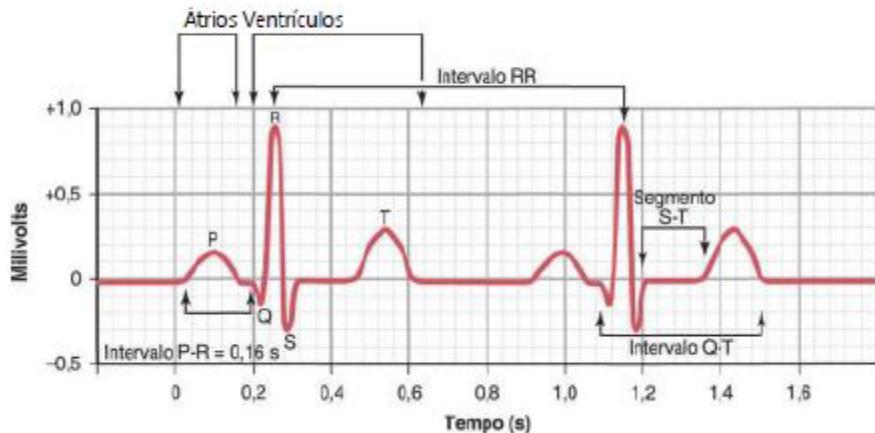
4123

A base teórica visa estabelecer os requerimentos funcionais e de segurança elétrica para a saída de teste de ECG, projetar o hardware (DAC + condicionamento) e firmware para a geração de formas de onda, desenvolver uma biblioteca de ritmos (sinusal, bradi/taqui, assistolia, FA, PVC) com FC chegando a 240 bpm, proporcionar uma interface com o usuário (botões/encoder e display opcional) e uma interface serial para configuração, conferir amplitude, offset e taxa de amostragem, lista de componentes (BOM), esquema e código comentada.

### 2.1 DESENVOLVIMENTO DE UM SIMULADOR DE ELETROCARDIOGRAMA (ECG) MICROCONTROLADO.

O eletrocardiograma é o registro gráfico da atividade elétrica do coração, sendo composto por ondas características: onda P, complexo QRS e onda T. Cada uma dessas componentes fornece informações específicas sobre a despolarização e repolarização das câmaras cardíacas (TORTORA; DERRICKSON, 2017). A Figura 1 mostra o formato de onda de um eletrocardiograma normal.

**Figura 1** – Eletrocardiograma normal



Fonte: (HALL; GUYTON, 2021)

Na engenharia clínica, a utilização de simuladores é fundamental para garantir a qualidade e confiabilidade dos equipamentos médicos. Esses dispositivos permitem a validação de funcionamento de monitores multiparamétricos, além de fornecerem suporte ao treinamento de equipes médicas (WEBSTER, 2010).

O ESP32 é um microcontrolador de alto desempenho, integrando conectividade Wi-Fi, Bluetooth, ADC, DAC, timers de precisão e arquitetura dual-core, sendo amplamente empregado em sistemas embarcados e aplicações biomédicas devido à sua versatilidade e baixo custo (ESPRESSIF SYSTEMS, 2021).

4124

## 2.2 BIOPOTENCIAIS CARDÍACOS E MORFOLOGIA DO ECG

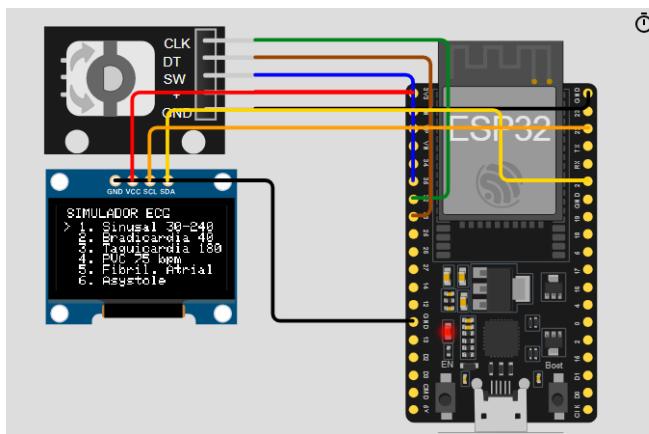
O ECG de superfície é composto por ondas P, complexo QRS e onda T, relacionadas à despolarização e repolarização cardíacas. Em registros diagnósticos padrão, 1 mV corresponde a 10 mm no papel (calibração típica), com amplitudes aproximadas de P de 0,1–0,3mV, QRS de 1–1,5 mV e T 0,2–0,5 mV (WEBSTER, 2010; TORTORA; DERRICKSON, 2017). A banda de interesse para monitores situa-se, em geral, entre 0,05–40 Hz (monitorização) e até 150 Hz (diagnóstico) (IEC, 2011).

## 2.3 SIMULAÇÃO ELETRÔNICA DE ECG

A simulação eletrônica pode ser obtida por reprodução digital com DAC. A abordagem digital oferece maior reprodutibilidade, seleção de ritmos e controle de FC e amplitude (KHANDPUR, 2014; WEBSTER, 2010). A Figura 2 apresenta o diagrama de ligação entre o

microcontrolador ESP32, o display OLED e o encoder rotativo no ambiente de simulação Wokwi, utilizado para validação inicial da lógica do sistema.

**Figura 2** – Diagrama de ligação protótipo no ambiente de simulação Wokwi.

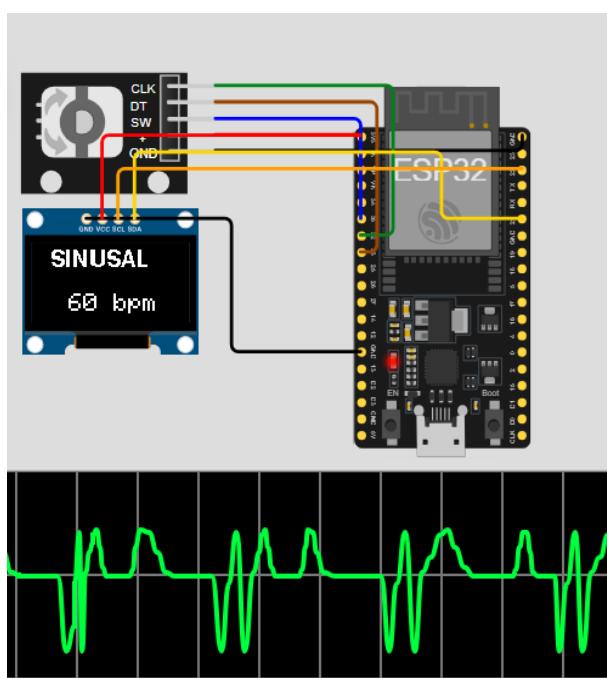


**Fonte:** Elaborada pelo autor, 2025.

Na Figura 3 é exibido o modo sinusal a 60 bpm no display, acompanhado da forma de onda correspondente mostrada no monitor serial simulada no ambiente virtual.

**Figura 3** – Ritmo sinusal 60 bpm no display e forma de onda correspondente gerada no simulador Wokwi..

4125



**Fonte:** Elaborada pelo autor, 2025.

### 3. METODOLOGIA

A metodologia adotada para o desenvolvimento do simulador foi estruturada em etapas sequenciais e complementares, permitindo uma construção sistemática e verificável do protótipo. As fases compreenderam: levantamento bibliográfico, definição do circuito eletrônico, implementação em protoboard, desenvolvimento do firmware e realização de testes funcionais. A Figura 04 sintetiza esse fluxo.

Inicialmente, realizou-se um estudo bibliográfico envolvendo princípios de eletrocardiografia, instrumentação biomédica, conversão digital-analógica, microcontroladores e normas aplicáveis a equipamentos de ECG. As referências possibilitaram definir os requisitos de projeto, bem como as características elétricas e clínicas que os sinais simulados deveriam apresentar.

Na etapa de construção, utilizou-se um conjunto de componentes amplamente disponíveis no mercado nacional: microcontrolador **ESP32**, conversor digital-analógico **MCP4725**, *display OLED*, *encoder* rotativo, bateria 18650 e módulo regulador de tensão de 5 V. A montagem inicial foi feita em *protoboard*, permitindo ajustes rápidos e alterações no circuito conforme resultados preliminares.

4126

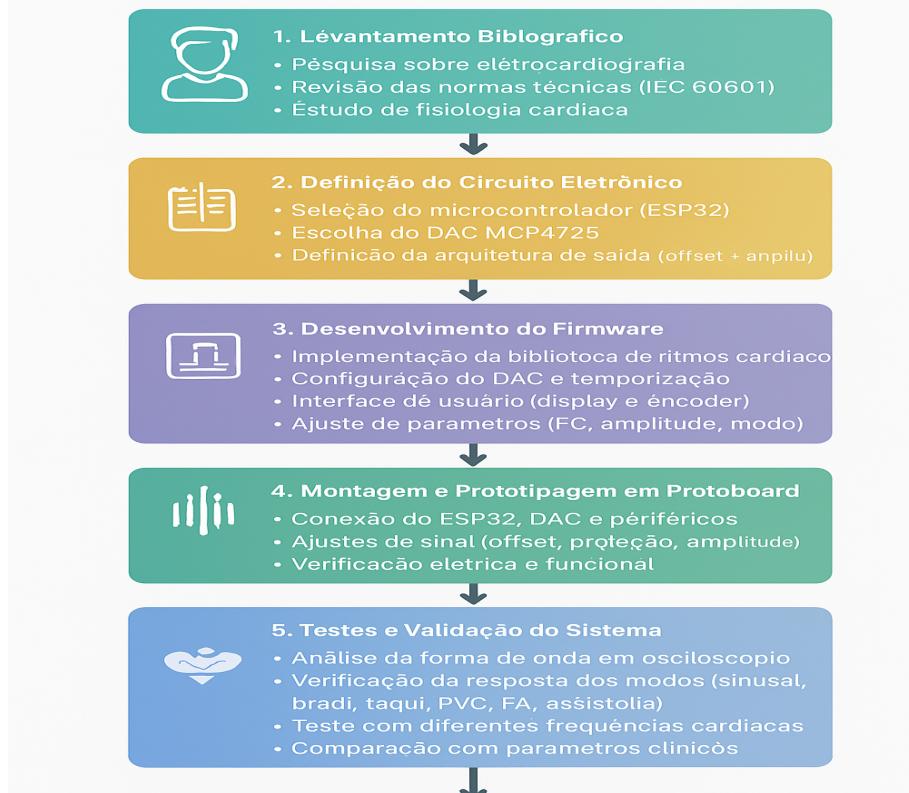
Após a integração dos componentes, iniciou-se o desenvolvimento do firmware para o **ESP32**, responsável pela geração dos sinais amostrados, controle de temporização, estrutura de menus e seleção dos ritmos cardíacos. O conversor **MCP4725** operou como principal meio de saída analógica, garantindo baixo ruído e alta resolução.

A validação funcional dos ritmos simulados foi realizada por meio de um **osciloscópio**, permitindo observar a forma de onda, amplitude, offset e estabilidade temporal dos sinais. Foram avaliados diferentes modos de operação, como: ritmo sinusal, bradicardia, taquicardia, extrassístole ventricular, fibrilação atrial e assistolia, além de variações de frequência cardíaca e ajustes de amplitude. Cada ensaio foi devidamente registrado, incluindo capturas de tela, medidas e observações relevantes.

O processo metodológico seguido na construção do simulador está sintetizado na Figura 4, que ilustra as etapas de levantamento bibliográfico, definição do circuito, montagem e validação do protótipo.

**Figura 4** – Fluxograma das etapas sequenciais do desenvolvimento do simulador.

## Fluxograma do Desenvolvimento do Simulador de ECG

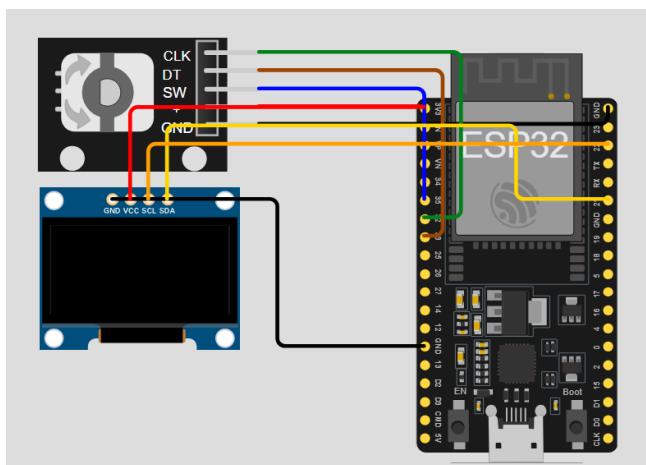


4127

**Fonte:** Elaborada pelo autor, 202

A Figura 5 apresenta a disposição prática dos periféricos conectados ao ESP32, indicando os pontos de alimentação, comunicação e interface utilizados no projeto.

**Figura 5** – Ligação dos periféricos ao microcontrolador ESP32.



**Fonte:** Elaborada pelo autor, 2025.

### 3.1 REQUISITOS DE ENTRADA DOS EQUIPAMENTOS

As entradas de ECG presentes em monitores multiparamétricos e eletrocardiógrafos apresentam alta impedância de entrada (tipicamente superior a  $2,5\text{ M}\Omega$ ), tolerância a polarização DC e circuitos destinados à rejeição de modo comum, como o Right Leg Drive (RLD).

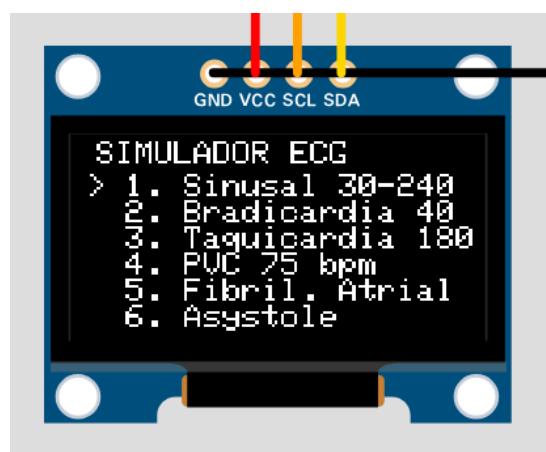
Para ensaios práticos com simuladores, recomenda-se fornecer ao equipamento um sinal diferencial de baixa amplitude (da ordem de 1 mV a alguns milivolts), com offset seguro e proteção via resistores em série, conforme diretrizes da IEC 60601-2-27 e princípios descritos por Webster (2010). Esses cuidados garantem compatibilidade elétrica adequada e evitam saturação dos amplificadores de entrada do monitor.

### 3.2 REQUISITOS DO SISTEMA

As especificações técnicas e funcionais propostas para o desenvolvimento de um simulador de Eletrocardiograma (ECG) microcontrolado, com foco na geração de ritmos, ajustes de parâmetros, topologia de saída e sistema de alimentação. O simulador deverá implementar uma biblioteca essencial de ritmos cardíacos, abrangendo estados normais e patológicos comuns, cruciais para o teste de monitores e eletrocardiógrafos. A usabilidade e flexibilidade do simulador são garantidas por meio de ajustes de frequência e uma interface de seleção intuitiva. As saídas elétricas devem emular as conexões padrão de um paciente a um equipamento de ECG, garantindo compatibilidade e segurança. A segurança elétrica é crítica, especialmente no que diz respeito à alimentação e ao carregamento. A Figura 6 mostra o menu de seleção no *display*, onde o operador pode escolher o ritmo a ser simulado no ambiente wokwi.

4128

Figura 6 – Menu de seleção simulado em ambiente wokwi.



Fonte: Elaborada pelo autor, 2025.

## 4. PROJETO DE HARDWARE

### 4.1 ESQUEMÁTICO

#### 4.1.1 Fonte e referência

Na Figura 7 temos a bateria de lítio modelo 18650 de 3,7 V. A Figura 8 mostra o módulo bms (*battery management system*), um componente que monitora, protege e otimiza o desempenho da bateria, prevenindo sobrecarga, descarga profunda, superaquecimento e garantindo sua segurança, eficiência e vida útil. A Figura 9 mostra o módulo *step-up/step-down* com saída usb que aumenta a tensão de 3,7V da bateria para 5V e alimenta o protótipo.

**Figura 7 – Bateria 18650.**



**Fonte:** WJ Componentes, 2025.

**Figura 8- Módulo BMS .**



**Fonte:** RS Robótica, 2025.

**Figura 9 - Step-up/down 5 V.**



**Fonte:** Casa da Robótica, 2025.

4129

#### 4.1.2 Microcontrolador

A arquitetura central do simulador utiliza o microcontrolador ESP32, responsável pelo processamento digital e controle de temporização, que pode ser observado na **Figura 10**.

**Figura 10 – Microcontrolador ESP32.**

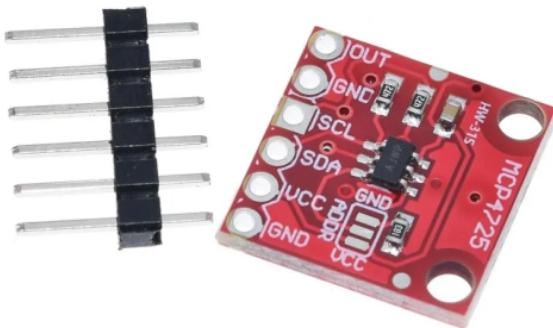


**Fonte:** Mercado Livre, 2025.

#### 4.1.3 DAC (Conversor Digital Analógico)

A conversão do sinal digital gerado pelo ESP32 em forma de onda analógica é realizada pelo DAC MCP4725. A Figura 11 apresenta a visualização do módulo, evidenciando seus pinos de alimentação e a interface I<sup>2</sup>C utilizada na comunicação com o microcontrolador.

**Figura 11 – MCP4725**



**Fonte:** Mercado Livre, 2025.

#### 4.1.3 Interface de usuário

A interface visual do sistema é exibida em um display OLED de 0,96", comunicação I<sup>2</sup>C, 4130 que apresenta menus, modo selecionado e frequência cardíaca configurada. A Figura 12 ilustra o módulo SSD1306 empregado no projeto.

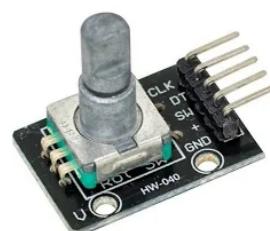
A seleção de modos e o ajuste de parâmetros do simulador são realizados por meio de um encoder rotativo com botão integrado. A Figura 13 apresenta o módulo utilizado, destacando seus três sinais principais (CLK, DT e SW) conectados às GPIOs 34, 35 e 32 do ESP32.

**Figura 12 – Display OLED 0.96"**



**Fonte:** Mercado Livre, 2025

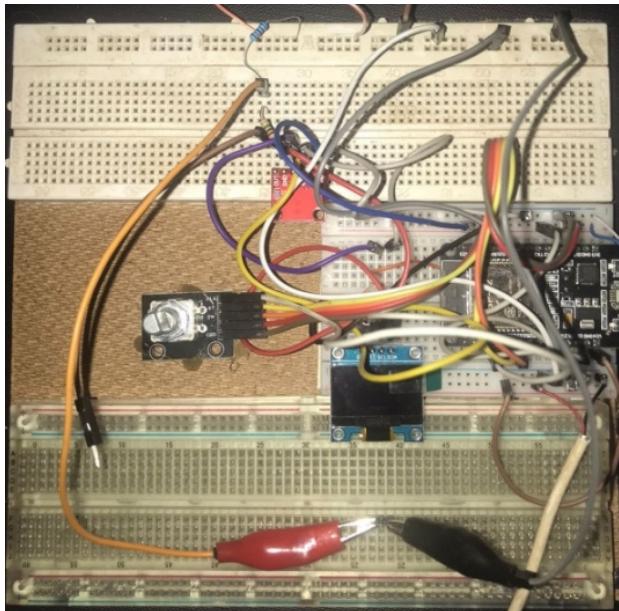
**Figura 13 – Encoder rotativo com push button.**



**Fonte:** Mercado Livre, 2025

A montagem experimental do simulador foi inicialmente realizada em protoboard, permitindo validar o funcionamento do ESP32, do DAC MCP4725 e dos demais periféricos antes da implementação definitiva. A Figura 14 apresenta o protótipo completo durante a fase de testes.

**Figura 14** – Protótipo montado em *protoboard*



4131

**Fonte:** Elaborada pelo autor, 2025.

#### 4.2 LISTA DE MATERIAL

Para viabilizar a reprodução do projeto e garantir clareza na seleção dos componentes utilizados, foi elaborada uma lista completa de materiais necessários para a construção do simulador. A Tabela 1 apresenta todos os itens empregados, incluindo o microcontrolador, o conversor DAC, periféricos, conectores e elementos auxiliares.

Tabela 1 – Lista de materiais utilizados na construção do simulador de ECG.

QUANTIDADE	PEÇA
1	ESP32 DevKit
1	MCP4725 DAC 12-bit
2	Módulos de bateria 18650

I	Suporte para 4 baterias 18650
I	Módulo BMS
I	Regulador <i>step-up/step-down</i> 5 V
I	Encoder rotativo com <i>push button</i>
I	Display OLED 0,96" I <sup>2</sup> C
VÁRIOS	Cabos, conectores, protoboard

**Fonte:** Elaborada pelo autor, 2025.

## 5. FIRMWARE

Abaixo, firmware para ESP32. Implementa: tabela base de ECG sinusal, geração *sample-accurate* via Timer, seleção de ritmos, ajuste de bpm e amplitude, e modos: Sinusal, **Assistolia, Bradicardia, Taquicardia, PVC e FA simulada.**

```
// HARDWARE (ESP32 + MCP4725)
#include <Wire.h>
#include <Adafruit_MCP4725.h>
#include <Adafruit_SSD1306.h>
#include <Fonts/FreeSansBold9pt7b.h>
#include "esp32-hal-timer.h"
#include "freertos/task.h"

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define SCREEN_ADDRESS 0x3C
#define DAC_ADDR 0x60
#define CLK_PIN 34
#define DT_PIN 35
#define SW_PIN 32

hw_timer_t * timer = NULL;
const uint32_t TIMER_FREQ_HZ = 1000000;

#define MODO_MENU 0
#define MODO_SINUSAL 1
#define MODO_BRADI 2
#define MODO_TAQUI 3
#define MODO_PVC 4
#define MODO_FA 5
#define MODO_ASYSTOLE 6

volatile uint8_t modo = MODO_MENU;
volatile int bpm_current = 60;
volatile int menuItem = 0;
volatile bool inMenu = true;
volatile bool write_dac_flag = false;
```

```
int lastCLK = HIGH;
long lastTurnTime = 0;
const long debounce = 3;

const int ARRAY_SIZE = 200;
volatile int ecg_index = 0;
float pvcCounter = 0;
bool insertPVC = false;

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
Adafruit_MCP4725 dac;

// MODEL
struct Peak { float phase; float amp; float sigma; };
const Peak peaks[5] = {
    {0.12f, 0.15f, 0.03of},
    {0.28f, -0.08f, 0.01of},
    {0.3of, 1.0of, 0.006f},
    {0.32f, -0.12f, 0.01of},
    {0.55f, 0.3of, 0.04of}
};

float baselineAmp = 6.0f;
float noiseLSB = 4.0f;
float gainScale = 1024.0f;

void atualizaTimer();
void showMenu();
void showStatus(const char* texto, int valorBPM);

void IRAM_ATTR onTimer() {
if (modo != MODO_MENU && modo != MODO_ASYSTOLE) write_dac_flag = true;
}

float model_ecg(float theta) {
    float z = 0.0f;
    for (int i=0;i<5;i++){
        float d = theta - peaks[i].phase;
        if (d > 0.5f) d -= 1.0f;
        if (d < -0.5f) d += 1.0f;
        float s = peaks[i].sigma;
        z += peaks[i].amp * expf(-0.5f * (d*d)/(s*s));
    }
    return z;
}

void writeDacHardware(int value) {
    value = constrain(value, 0, 4095);
    dac.setVoltage((uint16_t)value, false);
}

int scaleModelToDAC(float modelVal){
    float out = 2048.0f + modelVal * gainScale;
    return (int)constrain(round(out), 0, 4095);
}
```

```
void generateSinusal(){
    float theta = (float)ecg_index / (float)ARRAY_SIZE;
    float v = model_ecg(theta);
    float tsec = (float)millis() / 1000.0f;
    float baseline = baselineAmp * sinf(2.0f * 3.14159265f * 0.15f * tsec);
    float noise = ((float)random(-100,100)/100.0f) * noiseLSB;
    float combined = v + (baseline/ gainScale) + (noise/ gainScale);
    int dacv = scaleModelToDAC(combined);
    writeDacHardware(dacv);
}

void generatePVC(){
pvcCounter += 1.0f;
if (pvcCounter >= ARRAY_SIZE * 8) { insertPVC = true; pvcCounter = 0; }
if (insertPVC&&ecg_index< 12) {
    float theta = (float)ecg_index / (float)ARRAY_SIZE;
    float v = model_ecg(theta) * 1.6f + 0.3f;
    int dacv = scaleModelToDAC(v);
    writeDacHardware(dacv);
    if (ecg_index>= 11) insertPVC = false;
} else {
    generateSinusal();
}
}

void generateFA(){
float jitter = ((float)random(-50,50)) / 1000.0f;
float theta = (float)ecg_index / (float)ARRAY_SIZE + jitter;
if (theta > 1.0f) theta -= 1.0f;
if (theta < 0.0f) theta += 1.0f;
float v = model_ecg(theta);
v += ((float)random(-300,300))/4096.0f;
int dacv = scaleModelToDAC(v);
writeDacHardware(dacv);
}

void dacTask(void *pvParameters) {
    while (true) {
        if (write_dac_flag) {
            write_dac_flag = false;
            switch (modo) {
                case MODO_SINUSAL: generateSinusal(); break;
                case MODO_BRADI: generateSinusal(); break;
                case MODO_TAQUI: generateSinusal(); break;
                case MODO_PVC: generatePVC(); break;
                case MODO_FA: generateFA(); break;
                case MODO_ASYSTOLE: writeDacHardware(2048); break;
                default: writeDacHardware(2048); break;
            }
            ecg_index++;
            if (ecg_index>= ARRAY_SIZE) ecg_index = 0;
        }
        vTaskDelay(1 / portTICK_PERIOD_MS);
    }
}
```

}

```
// encoder/menu/display - mantidos
void handleEncoder() {
    int clk = digitalRead(CLK_PIN);
    if (clk != lastCLK&&clk == LOW) {
        if (millis() - lastTurnTime > debounce) {
            lastTurnTime = millis();
            if (digitalRead(DT_PIN) == LOW) {
                if (inMenu) { menuItem = (menuItem + 1) % 6; showMenu(); }
                else if (modo == MODO_SINUSAL) { bpm_current = constrain(bpm_current + 5, 30, 240); atualizaTimer(); }
                showStatus("SINUSAL", bpm_current); }
            } else {
                if (inMenu) { menuItem = (menuItem + 5) % 6; showMenu(); }
                else if (modo == MODO_SINUSAL) { bpm_current = constrain(bpm_current - 5, 30, 240); atualizaTimer(); }
                showStatus("SINUSAL", bpm_current); }
            }
        }
    lastCLK = clk;

    if (digitalRead(SW_PIN) == LOW) {
        delay(20);
        if (digitalRead(SW_PIN) == LOW) {
            while(digitalRead(SW_PIN) == LOW);
            delay(20);
            if (inMenu) {
                inMenu = false;
                modo = menuItem + 1;
                atualizaTimer();
                switch (modo) {
                    case MODO_SINUSAL: bpm_current = 60; showStatus("SINUSAL", bpm_current); break;
                    case MODO_BRADI: bpm_current = 40; showStatus("BRADIC", 40); break;
                    case MODO_TAQUI: bpm_current = 180; showStatus("TAQUIC", 180); break;
                    case MODO_PVC: bpm_current = 75; showStatus("PVC", 75); break;
                    case MODO_FA: bpm_current = 0; showStatus("FIB.ATR", 0); break;
                    case MODO_ASYSTOLE: bpm_current = 0; writeDacHardware(2048); showStatus("ASYSTOLE", 0);
                }
            }
        }
    }
}

void showMenu() {
    display.clearDisplay();
    display.setFont();
    display.setTextSize(1);
    display.setTextColor(SSD1306_WHITE);
    display.setCursor(8, 2);
```

```
display.println("SIMULADOR ECG");
display.setTextSize(1);
int startY = 12;
int lineHeight = 8;
const char* itens[] = {
    "1. Sinusal 30-240",
    "2. Bradicardia 40",
    "3. Taquicardia 180",
    "4. PVC 75 bpm",
    "5. Fibril. Atrial",
    "6. Asystole"
};
for (int i = 0; i < 6; i++) {
    display.setCursor(5, startY + i * lineHeight);
    if (i == menuItem) display.print("> ");
    else display.print(" ");
    display.println(itens[i]);
}
display.display();
}

void showStatus(const char* texto, int valorBPM) {
display.clearDisplay();
display.setFont(&FreeSansBold9pt7b);
display.setTextSize(1);
display.setTextColor(SSD1306_WHITE);
display.setCursor(15,18);
display.println(texto);
display.setFont();
display.setTextSize(2);
if (valorBPM > 0) {
display.setCursor(30,40);
display.print(valorBPM);
display.print(" bpm");
} else {
display.setCursor(10,40);
display.println("SEM PULSO");
}
display.display();
}

void atualizaTimer() {
int targetBPM = bpm_current;
if (modo == MODO_BRADI) targetBPM = 40;
elseif (modo == MODO_TAQUI) targetBPM = 180;
else if (modo == MODO_PVC) targetBPM = 75;
elseif (modo == MODO_FA || modo == MODO_ASYSTOLE) return;
if (targetBPM > 0) {
    long interval_us = (60000L * 1000L) / (targetBPM * ARRAY_SIZE);
    if (timer != NULL) {
        timerWrite(timer, (uint64_t)interval_us);
        timerAlarm(timer, interval_us, true, 0);
    }
}
}
```

```
void setup() {  
Serial.begin(115200);  
Wire.begin(21, 22);  
if (!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {  
Serial.println("ERRO: Display não encontrado");  
}  
pinMode(CLK_PIN, INPUT_PULLUP);  
pinMode(DT_PIN, INPUT_PULLUP);  
pinMode(SW_PIN, INPUT_PULLUP);  
lastCLK = digitalRead(CLK_PIN);  
display.clearDisplay();  
display.setFont(&FreeSansBold9pt7b);  
display.setTextSize(1);  
display.setTextColor(SSD1306_WHITE);  
display.setCursor(22,28);  
display.println(" LTECH ");  
display.setCursor(10,55);  
display.println("SOLUCOES");  
display.setFont();  
display.setTextSize(1);  
display.setCursor(8,58);  
display.println("Simulador ECG 2025");  
display.display();  
delay(1200);  
// start DAC hardware  
dac.begin(DAC_ADDR);  
dac.setVoltage(2048, true);  
// Timer  
timer = timerBegin(TIMER_FREQ_HZ);  
timerAttachInterrupt(timer, &onTimer);  
xTaskCreatePinnedToCore(  
dacTask,  
    "DACTask",  
    4096,  
    NULL,  
    5,  
    NULL,  
    0  
);  
showMenu();  
}  
void loop() {  
handleEncoder();  
delay(5);  
}
```

4137

## Notas:

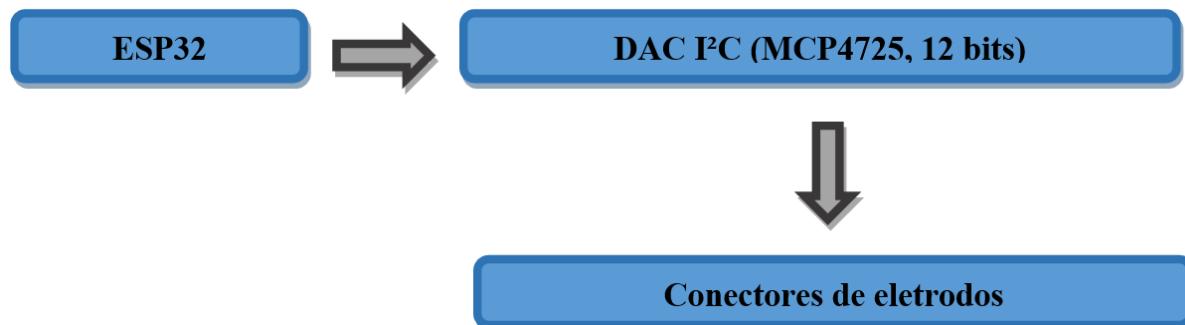
- A assistolia produz linha reta no valor médio; pode-se adicionar ruído branco de baixa amplitude para teste de detecção de artefatos.
- A FA simulada remove P e introduz variabilidade de RR (jitter) simples; para maior realismo, usar distribuição não uniforme de períodos.

## 6. DIAGRAMAS

### 6.1 ARQUITETURA GERAL (BLOCOS)

A arquitetura funcional do simulador foi organizada em três blocos principais: (1) processamento e geração digital de formas de onda no ESP32, (2) conversão digital-analógica por meio do DAC MCP4725 e (3) condicionamento e disponibilização do sinal nos conectores de saída. A Figura 15 apresenta o fluxograma dessa arquitetura, evidenciando o fluxo completo desde a seleção do ritmo até a saída destinada aos equipamentos de teste.

Figura 15 – Arquitetura geral do simulador de ECG



Fonte: Elaborado pelo autor, 2025.

4138

O microcontrolador ESP32 é empregado como o núcleo de processamento e geração de dados. A estratégia centraliza-se no armazenamento de perfis de formas de onda (ritmos biológicos) em tabelas de consulta (*lookup tables* - LUTs). Cada LUT contém um conjunto discreto de pontos que definem o formato do sinal desejado.

Para garantir a fidelidade e a suavidade da forma de onda gerada, mesmo com um número limitado de pontos nas LUTs, é aplicada a técnica de interpolação linear. Isso permite que o ESP32 calcule valores intermediários entre os pontos amostrados nas tabelas, gerando uma taxa de amostragem mais elevada e, consequentemente, uma reconstrução de sinal mais precisa e contínua no domínio analógico.

Para a conversão dos dados digitais gerados pelo ESP32 em um sinal analógico utilizável, é utilizado o Conversor Digital-Analógico (DAC) MCP4725. Este componente é escolhido por sua capacidade de fornecer uma saída analógica precisa e por sua simplicidade de integração via interface de comunicação I<sup>2</sup>C.

O MCP4725 opera com uma faixa de tensão de saída de até 3,3 V em escala completa (*full-scale*), compatível com a lógica de tensão da plataforma ESP32. Esta característica é crucial para garantir que o sinal analógico de saída represente com exatidão a amplitude definida digitalmente. O sinal analógico de saída do DAC é processado para criar uma saída diferencial, o que é essencial para aplicações onde a rejeição de ruído de modo comum é prioritária. A combinação do poder de processamento do ESP32 para interpolação de dados de LUTs com a precisão analógica do MCP4725 fornece uma solução eficiente e de alta fidelidade para a simulação de formas de onda biológicas em ambientes de teste e pesquisa.

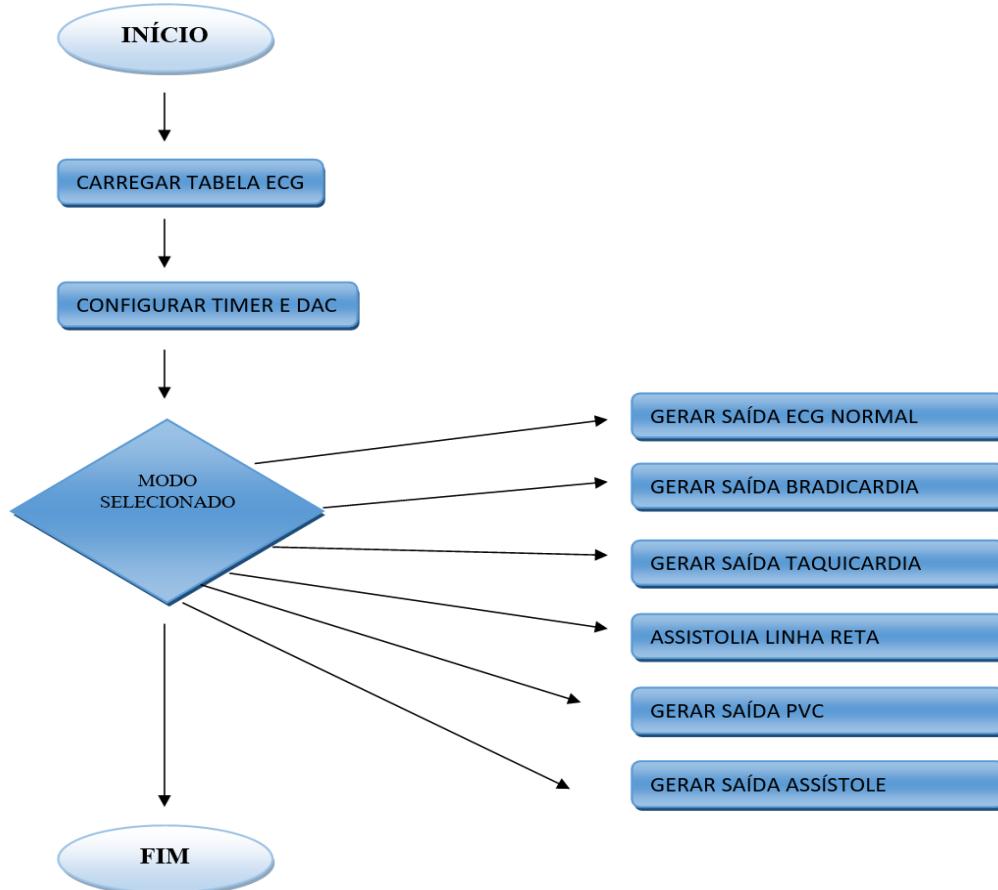
## 6.2 FLUXO DE SOFTWARE

O fluxograma do firmware do simulador de ECG ilustra a lógica principal utilizada para geração dos sinais cardíacos. O processo inicia-se com a etapa de inicialização geral do sistema, responsável por preparar o microcontrolador e os periféricos essenciais. Em seguida, ocorre o carregamento da tabela de amostras de ECG, que contém os valores pré-definidos utilizados como referência para compor as formas de onda. Após essa etapa, o sistema executa a configuração do temporizador (timer) de alta precisão e do DAC MCP4725, que garantem a taxa de amostragem correta e a conversão digital-analógica necessária. Com o hardware inicializado, o firmware avança para a verificação do modo selecionado pelo usuário no menu — Sinusal, Bradicardia, Taquicardia, PVC, Fibrilação Atrial ou Assistolia. A lógica interna direciona o fluxo para a rotina específica correspondente a cada ritmo, aplicando as variações de frequência, amplitude e dinâmica conforme a escolha realizada. Após gerar o conjunto de amostras, o ciclo é encerrado e reiniciado continuamente, garantindo a produção ininterrupta do sinal até uma nova seleção de modo ou desligamento.

4139

O funcionamento interno do firmware desenvolvido para o simulador pode ser compreendido por meio do fluxograma apresentado na Figura 16, que descreve a sequência lógica de inicialização, processamento e geração dos sinais simulados.

**Figura 16 – Fluxo de software**



**Fonte:** Elaborada pelo autor, 2025

4140

## 7. PROCEDIMENTOS DE MONTAGEM E VERIFICAÇÃO DE RESULTADOS

Este relatório detalha as etapas de montagem inicial e a verificação funcional do circuito de geração de sinais em protoboard, com foco na operação do Conversor Digital-Analógico (DAC) MCP4725. A placa foi montada em uma protoboard, interligando o microcontrolador ESP32 e o DAC MCP4725. Uma etapa crucial na montagem foi o desacoplamento da alimentação do DAC. Um capacitor cerâmico de 100 nF foi instalado entre os pinos de alimentação (3.3V Vcc) e terra (GND) do MCP4725. O posicionamento do capacitor foi feito o mais próximo possível dos pinos do CI para garantir a máxima eficácia na filtragem de ruídos de alta frequência presentes na linha de alimentação.

A verificação elétrica inicial foi realizada para confirmar a correta tensão de operação do sistema. Utilizando um multímetro, foi confirmada a tensão de 3.3 V (Volts) na trilha de

alimentação da *protoboard*. Essa tensão é fornecida a partir das baterias 18650, garantindo que o DAC MCP4725 opere com seu *full-scale* projetado e que os demais componentes digitais recebam a tensão nominal.

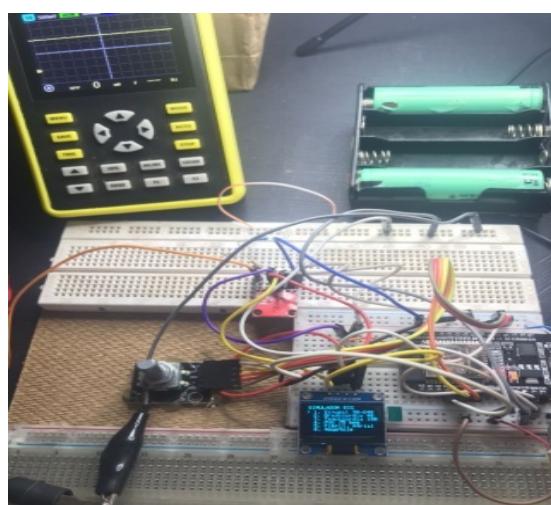
Após a confirmação da alimentação, procedeu-se ao teste funcional do sinal de saída do DAC utilizando um osciloscópio. A ponta de prova do osciloscópio foi conectada diretamente ao pino de saída analógica (Vout) do MCP4725. O *firmware* do microcontrolador foi configurado para gerar um sinal de teste no modo Sinusal com uma frequência de 60 batimentos por minuto (bpm). Osciloscópio configurado para escala vertical ajustada para 1 V por divisão (50mV/div) e escala horizontal ajustada na faixa de 500 ms por divisão (ms/div).

Ao observar a tela do osciloscópio, foi verificada a onda. O sinal apresentado correspondeu ao formato de uma onda senoidal (ou de ECG se este for o formato sinusual utilizado) com período de 1 segundo, confirmando que o DAC estava recebendo corretamente os dados digitais e realizando a conversão digital-analógica (DAC) com sucesso. Este teste inicial assegurou a integridade da montagem, a estabilidade da alimentação e o funcionamento básico da conversão de sinal, validando a plataforma para os testes de simulação de ritmos mais complexos.

4141

A interface inicial do dispositivo, utilizada para escolha dos ritmos cardíacos disponíveis, é apresentada na Figura 17, onde o simulador exibe o menu de seleção contendo todas as opções de modos de operação.

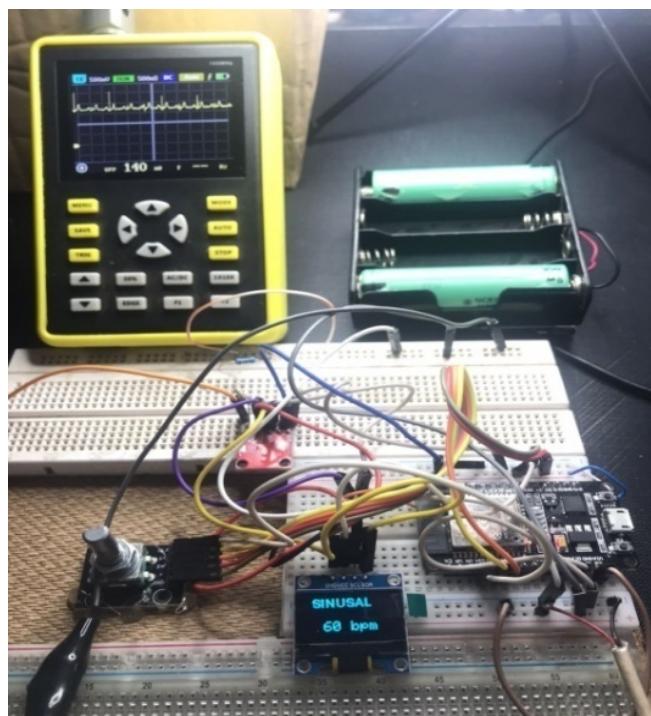
**Figura 17** – Protótipo de simulador de sinal de ECG no menu de seleção



**Fonte:** Elaborada pelo autor, 2025.

A Figura 18 ilustra o simulador em funcionamento no modo ritmo sinusal a 60 bpm, permitindo verificar o comportamento do display e do encoder durante a geração da forma de onda padrão, este é o único modo em que o usuário pode utilizar o encoder rotativo para alterar o valor de bpm e assim simular a onda desejada nas freqüências de 30 a 240 bpm. Os outros modos tem valores fixos para simular anomalias.

**Figura 18** – Simulador de sinal de ECG no modo sinusal 60 bpm



4142

**Fonte:** Elaborada pelo autor

## 8. DISCUSSÃO

A implementação do simulador de ECG utilizando o microcontrolador ESP32 demonstrou ser uma solução viável, de baixo custo e tecnicamente robusta para aplicações de ensino e manutenção em engenharia clínica. O uso do DAC MCP4725 mostrou-se adequado ao objetivo do projeto, especialmente pela capacidade de conversão de 12 bits, pela estabilidade da saída analógica e pela simplicidade de comunicação via protocolo I<sup>2</sup>C. Isso permitiu gerar sinais suaves, com baixo jitter e com controle preciso de amplitude e offset, características essenciais para testes de monitores multiparamétricos.

Em termos de desempenho, observou-se que o método baseado em tabelas (*lookup tables*) e modelos matemáticos de forma de onda foi eficiente para sintetizar ritmos cardíacos fisiológicos, como ritmo sinusal, bradicardia e taquicardia. Os modos patológicos (PVC, fibrilação atrial e assistolia) foram reproduzidos com boa consistência visual e compatibilidade com equipamentos reais. A validação com osciloscópio confirmou a integridade da forma de onda e a estabilidade temporal do gerador, reforçando a confiabilidade do sistema.

Apesar da eficiência do método, algumas limitações foram identificadas. A simulação da morfologia do ECG, embora satisfatória para testes funcionais, não corresponde fielmente às exigências normativas para instrumentos diagnósticos, especialmente as normas IEC 60601-2-25 e IEC 60601-2-27, que especificam requisitos rigorosos para amplitude, filtragem e linearidade. As formas de onda geradas são adequadas para validação básica de cabos, alarmes e aquisição de sinais, mas não devem ser utilizadas para calibrações oficiais de equipamentos diagnósticos.

Além disso, o projeto não contempla uma saída diferencial real com circuito de *right leg drive* (RLD), frequentemente utilizado em equipamentos de ECG clínico para aumentar a rejeição de ruído de modo comum. Embora não seja essencial para o propósito didático, sua ausência limita o uso do simulador em testes mais avançados.

4143

Ainda assim, a arquitetura escolhida, ESP32 + MCP4725, se mostrou altamente estável e expansível. O microcontrolador possui recursos suficientes para futuras implementações, como geração de sinais mais complexos, comunicação sem fio e integração com plataformas de calibração. Dessa forma, os resultados obtidos indicam que o simulador cumpre plenamente sua finalidade educacional e de apoio à manutenção preventiva, oferecendo um meio acessível e confiável para testes práticos.

## 9. CONCLUSÃO

O presente trabalho desenvolveu um simulador de sinais de ECG baseado no microcontrolador ESP32 associado ao DAC MCP4725, capaz de reproduzir diferentes ritmos cardíacos fisiológicos e patológicos. O sistema atendeu às necessidades de um instrumento acessível e robusto para uso em engenharia clínica, permitindo a verificação de cabos, canais, alarmes e parâmetros de equipamentos médicos de maneira segura e eficiente.

A implementação demonstrou ser tecnicamente sólida, combinando baixo custo, simplicidade de montagem e alta flexibilidade. A geração dos sinais se mostrou estável e compatível com o uso em monitores multiparamétricos, validada por meio de análise em osciloscópio. O menu intuitivo, aliado à interface por *encoder* e *display* OLED, proporciona uma experiência de uso clara tanto para estudantes quanto para profissionais.

Com uma arquitetura modular e expansível, o simulador estabelece uma base para futuras melhorias, podendo evoluir para aplicações mais avançadas de treinamento, automação de testes e protocolos de calibração. Deste modo, o trabalho contribui para a democratização de ferramentas essenciais para a engenharia clínica, promovendo acessibilidade, autonomia e aprimoramento técnico em instituições educacionais e de saúde.

## 10. TRABALHOS FUTUROS

Para aprimorar o desempenho e expandir as funcionalidades do simulador, são propostas as seguintes evoluções:

### 1. Comunicação sem fio avançada

- 
- 4144
- Implementação de aplicativo móvel (*Android/iOS*) para ajuste remoto de ritmos, amplitude e BPM;
  - Controle via Wi-Fi usando protocolo MQTT ou WebSocket;
  - Sincronização com monitores de teste em laboratório;
  - Desenvolver *firmware* e *software* capaz de gerar relatórios automáticos para calibração funcional;
  - Integração *bluetooth* para emissão de certificados de teste e manutenção preventiva.

### 2. Nova geração de formas de onda

- Implementação de algoritmos paramétricos baseados no modelo de McSharry para ECG sintético;
- Inclusão de ruídos fisiológicos configuráveis (EMG, respiração, *baseline wander*);
- Geração de interferência 50/60 Hz ajustável para testes de rejeição de modo comum.

### 3. Versão em PCB e carcaça profissional

- Desenvolvimento de placa dedicada para reduzir ruídos e aumentar estabilidade elétrica;
- Inclusão de filtros RC passa-baixa e estágio amplificador analógico;
- Projeto de carcaça impressa em 3D com conectores padrão ECG.

Essas melhorias podem transformar o protótipo atual em um simulador profissional de baixo custo, ampliando sua aplicação em laboratórios, empresas de engenharia clínica e instituições de ensino.

## II. REFERÊNCIAS

**ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR IEC 60601-1: Equipamento eletromédico – Parte 1: Requisitos gerais para segurança básica e desempenho essencial.** Rio de Janeiro, 2010.

BRUNNER, L. S.; SUDDARTH, D. S. **Tratado de enfermagem médica-cirúrgica.** 14. ed. Rio de Janeiro: Guanabara Koogan, 2018.

ESPRESSIF SYSTEMS. **ESP32 Series Datasheet.** Shanghai: Espressif Systems, 2021. Disponível em:  
[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf). Acesso em: 06 dez. 2025.

HALL, J. E.; GUYTON, A. C. **Tratado de fisiologia médica.** 14. ed. Rio de Janeiro: Elsevier, 2021.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 60601-2-25: Medical electrical equipment – Part 2-25: Particular requirements for the basic safety and essential performance of electrocardiographs.** Geneva, 2011.

4145

INTERNATIONAL ELECTROTECHNICAL COMMISSION. **IEC 60601-2-27: Medical electrical equipment – Part 2-27: Particular requirements for the basic safety and essential performance of electrocardiographic monitoring equipment.** Geneva, 2011.

KHANDPUR, R. S. **Handbook of Biomedical Instrumentation.** 3. ed. New Delhi: Tata McGraw-Hill, 2014.

TORTORA, G. J.; DERRICKSON, B. H. **Princípios de anatomia e fisiologia.** 14. ed. Rio de Janeiro: Guanabara Koogan, 2017.

WEBSTER, J. G. (ed.). **Medical instrumentation: application and design.** 4. ed. Hoboken: John Wiley & Sons, 2010.