

**CURSO SUPERIOR DE TECNOLOGIA EM
MANUFATURA AVANÇADA**

**PROJETO EXECUTIVO DE DESENVOLVIMENTO
INTEGRADO DE PRODUTO V - MÁQUINA INTERATIVA**

**JOÃO PEDRO DA SILVA ROSA
JOÃO VITOR MORAES DE SOUZA
GABRIEL ALVES DA SILVA**

São José dos Campos

2025

1. 1 INTRODUÇÃO

Informações contidas na minuta do projeto. O projeto consiste no desenvolvimento de uma máquina interativa, com o objetivo de promover a instituição FATEC. A interação presenteará aqueles que interagirem com brindes relacionados à instituição, que podem ser chaveiros, canetas, adesivos, pirulitos, balas etc.

O projeto será subdividido em 3 partes principais de desenvolvimento, sendo:

Eletrônica

Programação

Estrutural

O processo de desenvolvimento tem início com a montagem do circuito eletrônico do produto, que inicialmente contará com os componentes básicos para sua operação sendo:

1× Arduino Uno

1× Arduino Mega

4× Motores DC

2× Driver Ponte H L293D

1× Display LCD 16×2

4× Botões Push Button

1× Interruptor

A partir destes componentes básicos foram desenvolvidos os cálculos necessários para o desenvolvimento do circuito eletrônico, relativo a suas tensões, corrente e resistências necessárias para sua aplicação.

O objetivo deste circuito primordialmente é ter uma interface de interação com o usuário (LCD) da qual com a interação do cliente se enviará um Input para os motores DC que estarão internamente no produto, fazendo com que se acione alguma alavanca que "derrubará" o brinde ao usuário, sua recompensa pela interação.

O sistema possui dois modos de jogo interativos: um quiz de perguntas verdadeiro/falso e um jogo de memória de sequências. O usuário acumula pontos e ao final escolhe um prêmio (A, B, C ou D) que é dispensado automaticamente.

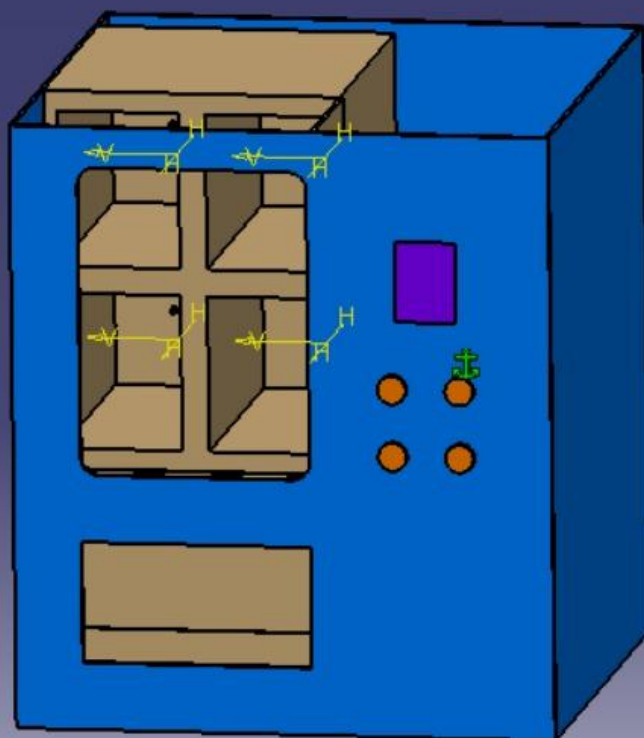
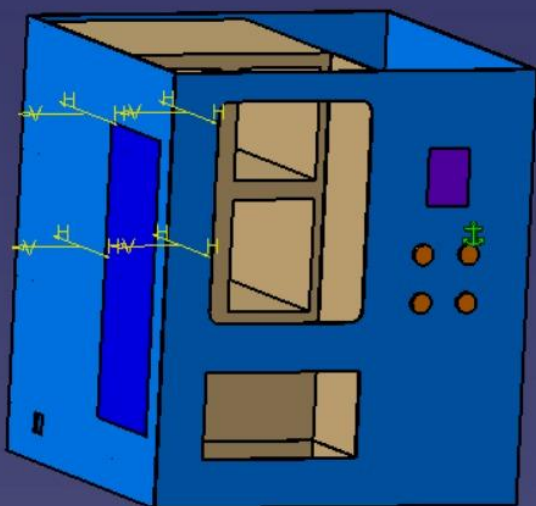
2. 2 PROJETO EXECUTIVO

O relatório do projeto executivo pode ser definido como um relatório as built – como construído – ou como um manual D.I.Y., do it yourself – faça você mesmo – ou ainda como um manual de instruções. Ou seja, o projeto executivo é uma descrição de todas as etapas realizadas no decorrer do projeto onde qualquer pessoa que tenha acesso ao relatório consiga reproduzir a solução que foi desenvolvida.

2.1 Lista de Materiais e Componentes Utilizados

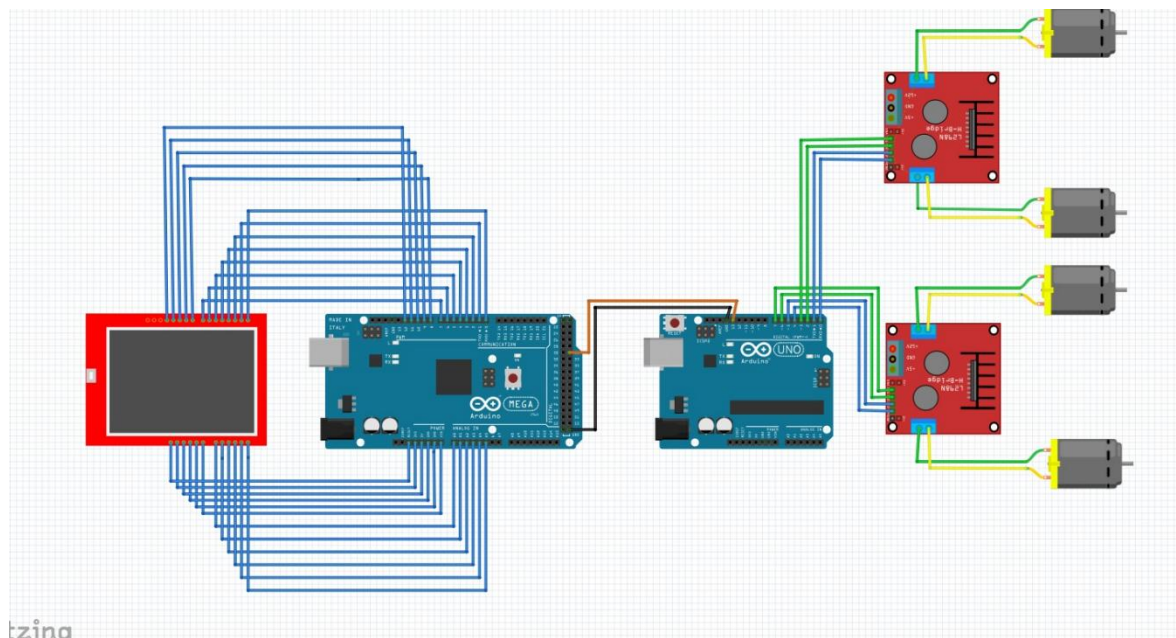
Estimativa de Materiais - Maquinas Interativa					
Materiais	Quantidade Necessária	Tamanho	Preço Unitário	Preço Total	Link
Arduino Uno	1	68,6 mm × 53,4 mm	R\$ 49,90	R\$ 49,90	Arduino UNO
Motor DC	4	70 mm × 22 mm × 18 mm	R\$ 7,90	R\$ 31,60	Motor DC
Driver A4988	2	15,3 mm × 20,3 mm	R\$ 16,90	R\$ 33,80	Driver Motor de Passo A4988 com Dissipador de Calor
Servomotor SG90	2	23x12.1x21.4mm	R\$ 12,50	R\$ 25,00	Servomotor
Tela Display TFT 2.4	1	71,5mm x 52mm.	R\$ 59,90	R\$ 59,90	Tela Display TFT
Sensor IR	2	32mm x 14mm	R\$ 6,90	R\$ 13,80	Sensor IR
Teclado matricial 4x4	1	77 mm × 70 mm × 0,8 mm	R\$ 6,50	R\$ 6,50	Teclado Matricial
Arduino Mega	1	53.36 × 101,86mm	R\$ 189,90	R\$ 189,90	Arduino Mega
Ponte H L298N	2	43x43x27mm	R\$ 18,90	R\$ 37,80	https://www.eletragate.com/ponte-h-dupla-l298n?srsId=AfmBOooxFKuecVTfZwfj0Jnjk-0DZSt2XCxmcY2mr0Rf7FqVgvFyY
Estrutura (MDF/Acrílico/Papelão)	1	-	-	-	-
Total Estimado		R\$	448,20		

2.2 Desenhos Técnicos da Solução



2.3 Circuitos Eléctricos

DIAGRAMA DE BLOCOS



Conexões do Display LCD 16x2 (Arduino Interface)

Pino Display	Pino Arduino	Função
VSS	GND	Terra
VDD	5V	Alimentação
V0	Potenciômetro	Contraste
RS	12	Register Select
RW	GND	Read/Write (Write)
E	11	Enable
D4	5	Data Bit 4
D5	4	Data Bit 5
D6	3	Data Bit 6
D7	2	Data Bit 7
A	5V (+ resistor 220Ω)	Backlight +
K	GND	Backlight -

Conexões dos Botões (Arduino Interface)

Botão	Pino Arduino	Função	Resistor Pull-down
Botão A	6	Quiz V / Menu	10kΩ
Botão B	7	Quiz F / Memória	10kΩ
Botão C	8	Próxima / Final	10kΩ
Botão D	9	Opção D	10kΩ

Conexões dos Drivers L293D (Arduino Motor)

Pino Driver	Pino Arduino	Função
EN1	2	Enable Motor 1 (PWM)
M1A	14 (A0)	Motor 1 Input A
M1B	15 (A1)	Motor 1 Input B
EN2	5	Enable Motor 2 (PWM)
M2A	3	Motor 2 Input A
M2B	4	Motor 2 Input B
VCC1	5V	Lógica
VCC2	12V	Motores
GND	GND	Terra comum

Interruptor

Componente	Pino Arduino Motor	Configuração
Switch	6	INPUT_PULLUP

Comunicação Serial entre Arduinos

Arduino Interface	Arduino Motor	Função
TX (1)	RX (0)	Transmissão
RX (0)	TX (1)	Recepção
GND	GND	Terra comum

Protocolo: 9600 baud

Mensagens:

- '0' = Sistema OFF (Motor → Interface)
- '1' = Sistema ON (Motor → Interface)
- 'A' = Acionar Motor 1 (Interface → Motor)
- 'B' = Acionar Motor 2 (Interface → Motor)
- 'C' = Acionar Motor 3 (Interface → Motor)
- 'D' = Acionar Motor 4 (Interface → Motor)

Cálculos Elétricos

Consumo de Corrente:

Arduinos: $2 \times 50\text{mA} = 100\text{mA}$

Display LCD: 20mA

Motores DC: $4 \times 200\text{mA} = 800\text{mA}$ (máximo)

Drivers: $2 \times 50\text{mA} = 100\text{mA}$

Total máximo: $1020\text{mA} + \text{margem } 100\% = \sim 2\text{A}$

Fonte especificada: 12V, 3A

Componentes de Proteção:

Resistores pull-down 10kΩ nos botões

Resistor 220Ω para backlight do LCD

Capacitores de desacoplamento 100nF (recomendados)

2.4 Algoritmos e Códigos

Descrição do Sistema

O software possui:

Estado OFF: Sistema aguardando ativação via interruptor

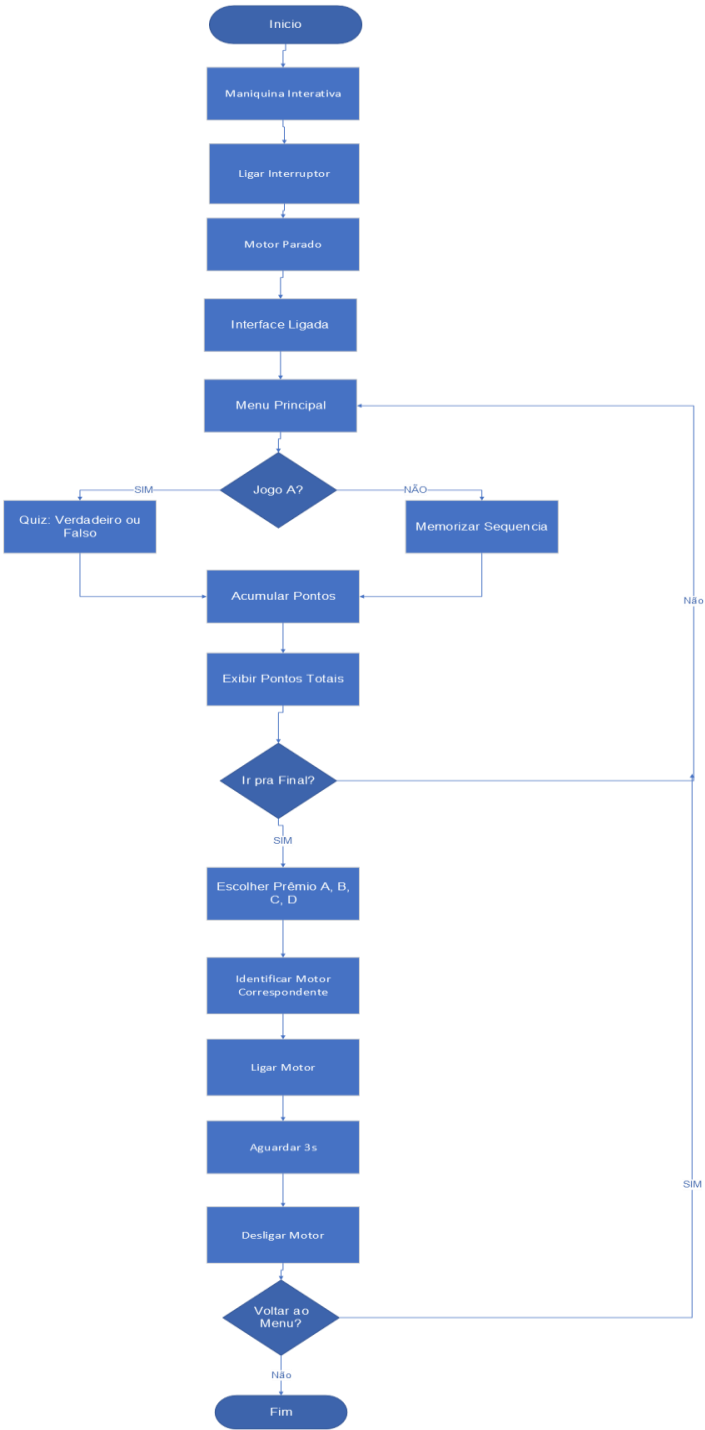
Estado MENU: Escolha entre Quiz ou Jogo de Memória

Estado QUIZ: 10 perguntas V/F, +10 pontos por acerto

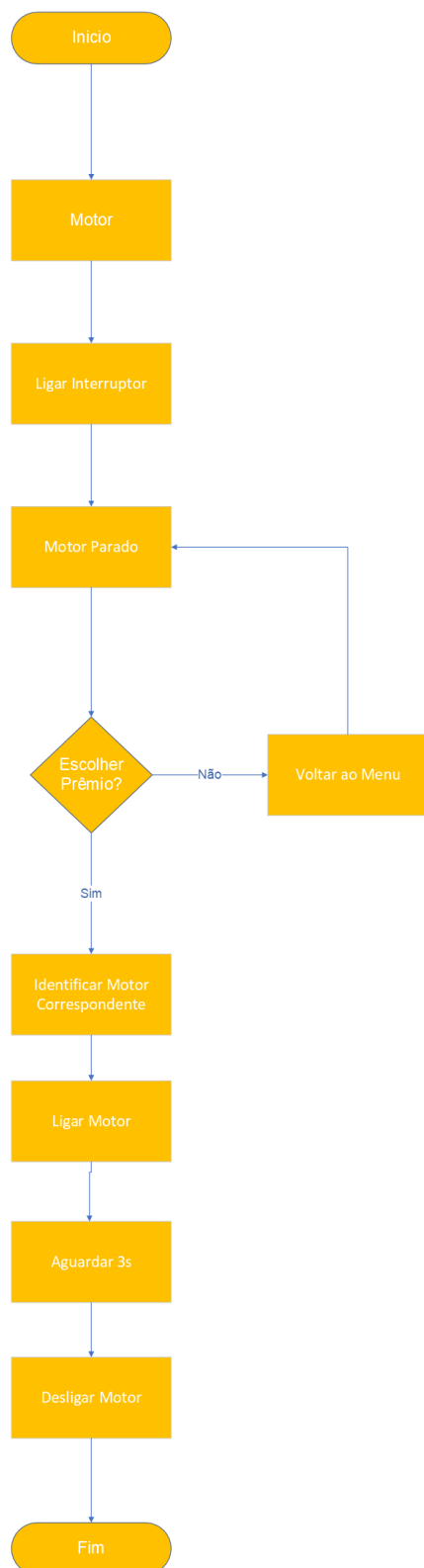
Estado REFLEXO: Jogo de memória, 5 níveis, +10×N pontos

Estado FINAL: Escolha de prêmio e acionamento de motor

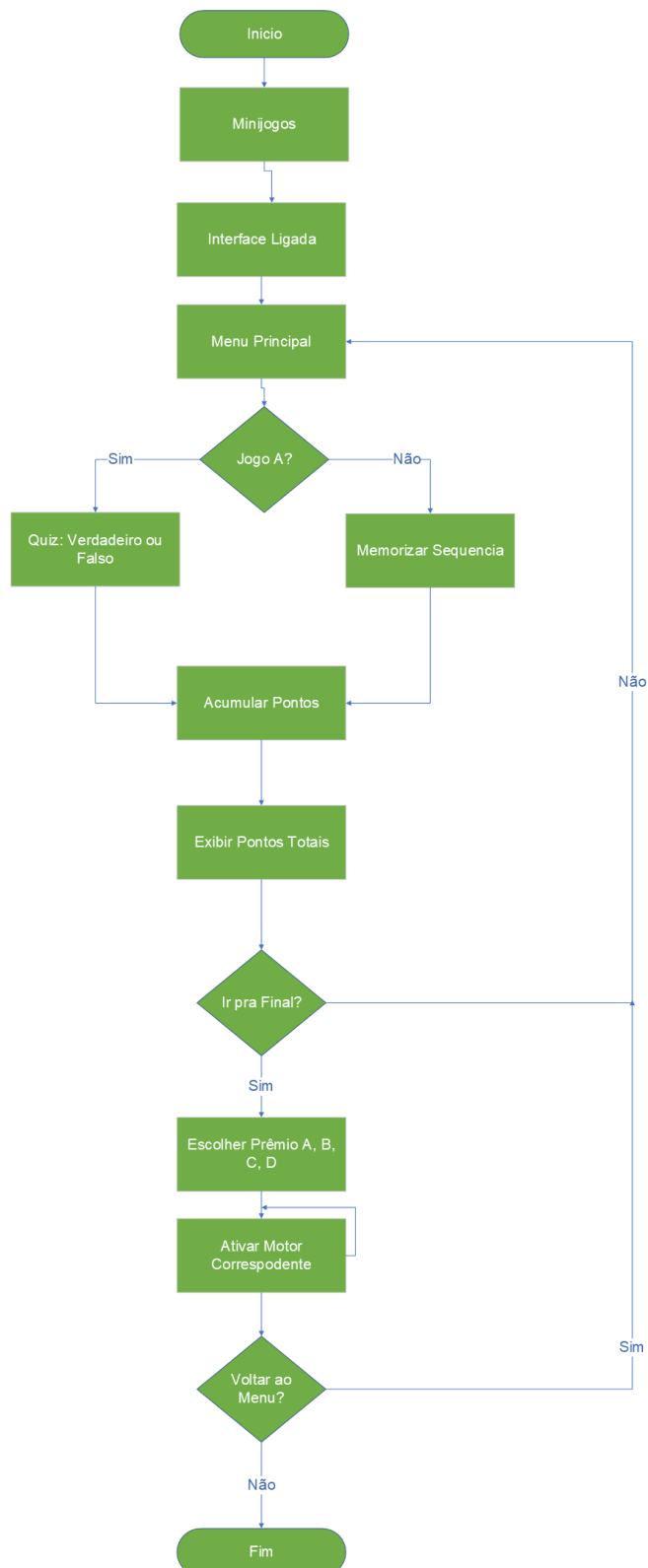
Fluxograma Principal



Fluxograma Motor



Fluxograma Interface



Código Arduino Interface

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Pinos
const byte BTN[4] = {6, 7, 8, 9}; // A, B, C, D

// Estados
enum Estado : byte { OFF, MENU, QUIZ, REFLEXO, FINAL };
Estado estado = OFF;
bool ligado = false;

// Variáveis globais
int pontos = 0;
byte idx = 0;
bool flag = false;

// Quiz - armazenado em PROGMEM (Flash) para economizar RAM
const char q0[] PROGMEM = "O Brasil tem 5";
const char q1[] PROGMEM = "regioes?";
const char q2[] PROGMEM = "A agua ferve a";
const char q3[] PROGMEM = "100 graus C?";
const char q4[] PROGMEM = "O Sol e uma";
const char q5[] PROGMEM = "estrela?";
const char q6[] PROGMEM = "Existem 12";
const char q7[] PROGMEM = "meses no ano?";
const char q8[] PROGMEM = "A Terra e";
const char q9[] PROGMEM = "plana?";
const char q10[] PROGMEM = "Os peixes";
```

```

const char q11[] PROGMEM = "respiram agua?";
const char q12[] PROGMEM = "O gelo e mais";
const char q13[] PROGMEM = "pesado q agua?";
const char q14[] PROGMEM = "Ha 60 minutos";
const char q15[] PROGMEM = "em 1 hora?";
const char q16[] PROGMEM = "Os morcegos";
const char q17[] PROGMEM = "sao cegos?";
const char q18[] PROGMEM = "A velocidade da";
const char q19[] PROGMEM = "luz e constante?";

const char* const perguntas[] PROGMEM = {
q0, q1, q2, q3, q4, q5, q6, q7, q8, q9,
q10, q11, q12, q13, q14, q15, q16, q17, q18, q19
};

const bool respostas[] PROGMEM = {1,1,1,1,0,0,0,1,0,1};

// Reflexo/Memória
byte seq[10], pos, nivel, ptsTemp;
bool mostrando, esperando, fim;
char buffer[17];

void setup() {
  lcd.begin(16, 2);
  for(byte i=0; i<4; i++) pinMode(BTN[i], INPUT);
  Serial.begin(9600);
  randomSeed(analogRead(A0));
  delay(2000);
  lcd.noDisplay();
}

void loop() {
  if (Serial.available()) {

```

```
char c = Serial.read();
if (c == '1' && !ligado) ligar();
else if (c == '0' && ligado) desligar();
}

if (ligado) {
  switch(estado) {
    case MENU: loopMenu(); break;
    case QUIZ: loopQuiz(); break;
    case REFLEXO: loopReflexo(); break;
    case FINAL: loopFinal(); break;
  }
}
delay(50);
}

void ligar() {
  ligado = true;
  lcd.display();
  lcd.clear();
  lcd.print(F("SISTEMA LIGADO!"));
  delay(1500);
  pontos = 0;
  menu();
}

void desligar() {
  ligado = false;
  estado = OFF;
  pontos = 0;
  lcd.clear();
  lcd.noDisplay();
}
```

```

void menu() {
    estado = MENU;
    lcd.clear();
    lcd.print(F("ESCOLHA O JOGO:"));
    lcd.setCursor(0, 1);
    lcd.print(F("A=Quiz B=Reflexo"));
}

void loopMenu() {
    if (lerBtn(0)) initQuiz();
    if (lerBtn(1)) initReflexo();
}

// ===== QUIZ =====
void initQuiz() {
    estado = QUIZ;
    idx = 0;
    ptsTemp = 0;
    flag = false;
    fim = false;
    lcd.clear();
    lcd.print(F("INICIANDO QUIZ"));
    delay(1000);
    mostrarQ();
}

void mostrarQ() {
    lcd.clear();
    lcd.print(F("Q"));
    lcd.print(idx + 1);
    lcd.print(F("/10:"));
    lcd.setCursor(0, 1);

```



```

strcpy_P(buffer, (char*)pgm_read_word(&(perguntas[idx*2]]));
lcd.print(buffer);
delay(1500);

lcd.clear();
strcpy_P(buffer, (char*)pgm_read_word(&(perguntas[idx*2+1]]));
lcd.print(buffer);
lcd.setCursor(0, 1);
lcd.print(F("A=V B=F"));
flag = false;
}

void loopQuiz() {
if (Serial.available() && Serial.read() == '0') {
desligar();
return;
}

if (!fim) {
if (!flag && lerBtn(0)) verificarResp(true);
if (!flag && lerBtn(1)) verificarResp(false);
if (flag && lerBtn(2)) proxQ();
}
}

void verificarResp(bool r) {
flag = true;
lcd.clear();

if (r == pgm_read_byte(&respostas[idx])) {
ptsTemp += 10;
lcd.print(F("CORRETO! +10pts"));
}
}

```

```

} else {
  lcd.print(F("ERRADO!"));
  lcd.setCursor(0, 1);
  lcd.print(F("Resp: "));
  lcd.print(pgm_read_byte(&respostas[idx]) ? 'V' : 'F');
}

delay(1500);
lcd.clear();
lcd.print(F("Pontos: "));
lcd.print(ptsTemp);
lcd.setCursor(0, 1);
lcd.print(F("C = Proxima"));
}

void proxQ() {
  if (++idx >= 10) finQuiz();
  else mostrarQ();
}

void finQuiz() {
  fim = true;
  pontos += ptsTemp;
  lcd.clear();
  lcd.print(F("Quiz Completo!"));
  lcd.setCursor(0, 1);
  lcd.print(F("Pontos: "));
  lcd.print(ptsTemp);
  delay(2000);
  pergProx();
}

// ===== REFLEXO/MEMÓRIA =====

```

```

void initReflexo() {
    estado = REFLEXO;
    nivel = 1;
    ptsTemp = 0;
    fim = false;
    lcd.clear();
    lcd.print(F("JOGO DE MEMORIA"));
    lcd.setCursor(0, 1);
    lcd.print(F("Decorar sequenc"));
    delay(2000);
    initNivel();
}

void initNivel() {
    for(byte i=0; i<nivel; i++) seq[i] = random(1, 5);
    mostrarSeq();
}

void mostrarSeq() {
    mostrando = true;
    lcd.clear();
    lcd.print(F("NIVEL "));
    lcd.print(nivel);
    lcd.setCursor(0, 1);
    lcd.print(F("Observe..."));
    delay(1000);

    for(byte i=0; i<nivel; i++) {
        lcd.clear();
        lcd.print(F("Memorize:"));
        lcd.setCursor(0, 1);
        lcd.print(F(">>> "));
        lcd.print((char)('A' + seq[i] - 1));
    }
}

```

```

lcd.print(F(" <<<"));
delay(800);
lcd.clear();
delay(300);
}

lcd.clear();
lcd.print(F("Sua vez!"));
lcd.setCursor(0, 1);
lcd.print(F("Repita sequenc"));
delay(1500);

pos = 0;
esperando = true;
mostrando = false;
mostrarProg();
}

void mostrarProg() {
lcd.clear();
lcd.print(F("Nivel "));
lcd.print(nivel);
lcd.print(F(" ("));
lcd.print(pos + 1);
lcd.print(F("/"));
lcd.print(nivel);
lcd.print(F(")"));
lcd.setCursor(0, 1);
lcd.print(F("A B C D"));
}

void loopReflexo() {
if (Serial.available() && Serial.read() == '0') {

```

```

desligar();
return;
}

if (esperando && !fim) {
byte b = 0;
for(byte i=0; i<4; i++) {
if (digitalRead(BTN[i])) {
b = i + 1;
break;
}
}

if (b) {
while(digitalRead(BTN[b-1])) delay(10);
verificarSeq(b);
}
}
}

void verificarSeq(byte b) {
if (b == seq[pos]) {
if (++pos >= nivel) {
ptsTemp += nivel * 10;
lcd.clear();
lcd.print(F("CORRETO! +"));
lcd.print(nivel * 10);
lcd.setCursor(0, 1);
lcd.print(F("Pontos: "));
lcd.print(ptsTemp);
delay(2000);

if (++nivel > 5) finReflexo();

```

```

else initNivel();
} else mostrarProg();
} else {
lcd.clear();
lcd.print(F("ERROU!"));
lcd.setCursor(0, 1);
lcd.print(F("Fim do jogo"));
delay(2000);
finReflexo();
}
}

void finReflexo() {
fim = true;
pontos += ptsTemp;
lcd.clear();
lcd.print(F("Jogo Completo!"));
lcd.setCursor(0, 1);
lcd.print(F("Pontos: "));
lcd.print(ptsTemp);
delay(2000);
pergProx();
}

// ===== TRANSIÇÃO =====
void pergProx() {
lcd.clear();
lcd.print(F("Total: "));
lcd.print(pontos);
lcd.print(F(" pts"));
lcd.setCursor(0, 1);
lcd.print(F("A=Menu C=Final"));

```

```

while(true) {
if (Serial.available() && Serial.read() == '0') {
desligar();
return;
}
if (lerBtn(0)) { menu(); return; }
if (lerBtn(2)) { telaFinal(); return; }
delay(50);
}
}

// ===== FINAL =====
void telaFinal() {
estado = FINAL;
lcd.clear();
lcd.print(F("PONTUACAO FINAL"));
lcd.setCursor(0, 1);
lcd.print(pontos);
lcd.print(F(" pontos!"));
delay(2500);

lcd.clear();
lcd.print(F("ESCOLHA PREMIO:"));
lcd.setCursor(0, 1);
lcd.print(F("A B C D"));
}

void loopFinal() {
if (Serial.available() && Serial.read() == '0') {
desligar();
return;
}
}

```

```

for(byte i=0; i<4; i++) {
if (lerBtn(i)) ativarMotor('A' + i);
}
}

void ativarMotor(char c) {
lcd.clear();
lcd.print(F("OPCAO "));
lcd.print(c);
lcd.setCursor(0, 1);
lcd.print(F("Ativando motor.."));

Serial.write(c);
Serial.flush();
delay(3000);

lcd.clear();
lcd.print(F("Concluido!"));
lcd.setCursor(0, 1);
lcd.print(F("A=Menu"));

while(true) {
if (Serial.available() && Serial.read() == '0') {
desligar();
return;
}
if (lerBtn(0)) {
pontos = 0;
menu();
return;
}
delay(50);
}
}

```



```
}  
  
// ===== UTILITÁRIOS =====  
bool lerBtn(byte b) {  
  if (digitalRead(BTN[b])) {  
    while(digitalRead(BTN[b])) delay(10);  
    return true;  
  }  
  return false;  
}
```

Código Arduino Motor

```
// ARDUINO MOTOR - L293D Control

#define SW 6

// L293D #1 - Motores 1 e 2
#define EN1 2
#define M1A 14
#define M1B 15
#define EN2 5
#define M2A 3
#define M2B 4

// L293D #2 - Motores 3 e 4
#define EN3 10
#define M3A 8
#define M3B 9
#define EN4 13
#define M4A 11
#define M4B 12

const byte EN[] = {EN1, EN2, EN3, EN4};
const byte MA[] = {M1A, M2A, M3A, M4A};
const byte MB[] = {M1B, M2B, M3B, M4B};

byte vel = 255;
bool prev = false;
bool on = false;

void setup() {
  Serial.begin(9600);
```

```

// Config pines
for(byte i=0; i<4; i++) {
    pinMode(EN[i], OUTPUT);
    pinMode(MA[i], OUTPUT);
    pinMode(MB[i], OUTPUT);
    digitalWrite(MA[i], LOW);
    digitalWrite(MB[i], LOW);
}

pinMode(SW, INPUT_PULLUP);
delay(100);

// Teste inicial enables
for(byte i=0; i<2; i++) {
    analogWrite(EN[i], 255);
    delay(50);
    analogWrite(EN[i], 0);
}

parar();
Serial.write('0');
Serial.flush();

Serial.println(F("=== PRONTO ==="));
Serial.println(F("Motores 1-2: L293D #1"));
Serial.println(F("Motores 3-4: L293D #2"));
}

void loop() {
    // Checa interruptor
    bool curr = (digitalRead(SW) == LOW);

    if (curr != prev) {

```

```

on = curr;
Serial.write(on ? '1' : '0');
Serial.flush();

if (!on) parar();

Serial.println(on ? F("ON") : F("OFF"));
prev = curr;
delay(50);
}

// Comandos serial
if (Serial.available()) {
  char c = Serial.read();

  if (on && c >= 'A' && c <= 'D') {
    byte m = c - 'A';
    Serial.print(F("Motor "));
    Serial.println(m + 1);
    ativar(m);
  }
}

// Estado periódico
Serial.write(on ? '1' : '0');
delay(200);
}

void ativar(byte m) {
  digitalWrite(MA[m], HIGH);
  digitalWrite(MB[m], LOW);
  analogWrite(EN[m], vel);
}

```

```
delay(3000);

analogWrite(EN[m], 0);
digitalWrite(MA[m], LOW);
digitalWrite(MB[m], LOW);
}

void parar() {
  for(byte i=0; i<4; i++) {
    analogWrite(EN[i], 0);
    digitalWrite(MA[i], LOW);
    digitalWrite(MB[i], LOW);
  }
}
```

2.5 Manual de Operação Básico

Para Ligar o Sistema:

- Conectar fonte à tomada
- Acionar interruptor
- Aguardar mensagem "SISTEMA LIGADO!"
- Sistema entra no Menu Principal automaticamente

Para Jogar Quiz:

- No menu, pressionar botão A
- Ler pergunta no LCD (mostrada em 2 partes)
- Pressionar A para Verdadeiro ou B para Falso
- Pressionar C para próxima pergunta
- Ao final de 10 perguntas, escolher entre Menu ou Final

Para Jogar Memória:

- No menu, pressionar botão B
- Observar sequência mostrada
- Repetir sequência pressionando botões
- Avançar pelos níveis até errar ou completar 5 níveis
- Ao final, escolher entre Menu ou Final

Para Receber Prêmio:

- Na tela final, verificar pontuação
- Pressionar botão do prêmio desejado (A, B, C ou D)
- Aguardar 3 segundos (motor acionando)
- Retirar brinde da bandeja inferior
- Pressionar A para voltar ao menu ou desligar

Para Desligar:





- Desligar interruptor
- Display apaga
- Sistema desliga

2.6 CRONOGRAMA MACRO

Entrega 3

Tarefas	Colaborador	Início	Fim
✓ Desenvolver o Fluxograma da Máquina	João Pedro	10/10/25	06/11/25
✓ Visualização e Testes da eletrônica	Gabriel e João Vitor	16/10/25	23/10/25
✓ Visualização e testes de programação	Equipe	21/10/25	30/10/25
✓ Finalização do Projeto Executivo	João Pedro	24/10/25	06/11/25
✓ Documentação e Apresentação	Equipe	03/11/25	06/11/25

2.7 CRONOGRAMA MACRO

Cronograma Micro - Entrega 3			
 Data	 Tarefa	 Detalhamento	 Responsável
10/10/2025	Início do desenvolvimento do fluxograma da máquina	Estruturação inicial do fluxograma e definição dos processos principais da máquina	João Pedro
14/10/2025	Refinamento do fluxograma	Ajustes lógicos e revisão da sequência operacional	João Pedro
16/10/2025	Início dos testes da eletrônica	Montagem dos componentes e primeiros testes de ligações	Gabriel e João Vítor
18/10/2025	Validação dos circuitos	Correção de falhas e conferência de funcionamento geral	Gabriel e João Vítor
21/10/2025	Início dos testes de programação	Implementação e depuração inicial do código	Equipe
25/10/2025	Testes integrados	Testes combinando software e hardware	Equipe
27/10/2025	Ajustes no código	Correção de bugs e melhoria de performance	Equipe
30/10/2025	Conclusão da etapa de programação	Validação completa e salvamento das versões finais	Equipe
24/10/2025	Início da finalização do projeto executivo	Revisão de medições e integração dos subsistemas	João Pedro
28/10/2025	Consolidação dos arquivos técnicos	Conferência e unificação dos desenhos e diagramas	João Pedro
03/11/2025	Início da documentação e da apresentação	Estruturação dos tópicos e definição do roteiro de apresentação	Equipe
05/11/2025	Redação da documentação final	Escrita e formatação do relatório e slides	Equipe
07/11/2025	Revisão geral da documentação	Correções finais e preparação dos anexos	Equipe
09/11/2025	Apresentação e entrega	Ensaio final e submissão da entrega 3	Equipe

3 REFERÊNCIAS

ARDUINO. **Arduino Uno Rev3**. Arduino Official Documentation, 2024.

Disponível em: <https://www.arduino.cc/en/Main/ArduinoBoardUno>.

ARDUINO. **LiquidCrystal Library Reference**. Arduino Official

Documentation, 2024. Disponível em:

<https://www.arduino.cc/en/Reference/LiquidCrystal>

ATMEL. **ATmega328P Datasheet**. Atmel Corporation, 2015. Disponível em:

<http://www.atmel.com>.

BANZI, Massimo; SHILOH, Michael. **Primeiros passos com o Arduino**. 2.

ed. São Paulo: Novatec, 2015.

INSTRUCTABLES. **Otto DIY - Build You Own Robot in Two Hours**.

Disponível em: <https://www.instructables.com/Otto-Build-You-Own-Robot-in-Two-Hours/>.

MARGOLIS, Michael. **Arduino cookbook**. 3. ed. Sebastopol: O'Reilly

Media, 2020.

MCROBERTS, Michael. **Arduino básico**. São Paulo: Novatec, 2011.

MONK, Simon. **Programação com Arduino: começando com sketches**.

2. ed. Porto Alegre: Bookman, 2017.

TEXAS INSTRUMENTS. **L293D Quadruple Half-H Driver Datasheet**.

Texas Instruments, 2016. Disponível em: <http://www.ti.com>.

Material didático do Curso Superior de Tecnologia em Manufatura

Avançada. **FATEC São José dos Campos**, 2023-2024.