

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE ENGENHARIA ELÉTRICA

FELIPE GABRIEL BASTOS DOS SANTOS

SISTEMA DE GERENCIAMENTO DE FILA COM PAINEL DE SENHAS

RELATÓRIO DO PROJETO FINAL DA DISCIPLINA DE ELETRÔNICA
DIGITAL

CURITIBA
2025

FELIPE GABRIEL BASTOS DOS SANTOS

SISTEMA DE GERENCIAMENTO DE FILA COM PAINEL DE SENHAS

Relatório do projeto final da disciplina de Sistemas Microcontrolados da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção da aprovação na disciplina no curso de Engenharia Elétrica/Controle e Automação.

Prof. Dr. Delvanei G. Bandeira jr.

CURITIBA

2025

SUMÁRIO

1	INTRODUÇÃO	Error! Indicador Não Definido .
2	DIAGRAMA EM BLOCOS DO PROJETO	3
3	DESCRIÇÃO DA SOLUÇÃO	4
4	LISTA DE MATERIAIS	4
5	CIRCUITO ESQUEMÁTICO DO PROJETO	5
6	MANUAL DE OPERAÇÃO	7
7	PRINCIPAIS DIFICULDADES ENCONTRADAS E SOLUÇÕES UTILIZADAS .	8
8	CONCLUSÕES	12
8.1	TRABALHOS FUTUROS	13
	REFERÊNCIAS	13

1.1.1 1. DESCRIÇÃO DO PROBLEMA E JUSTIFICATIVA

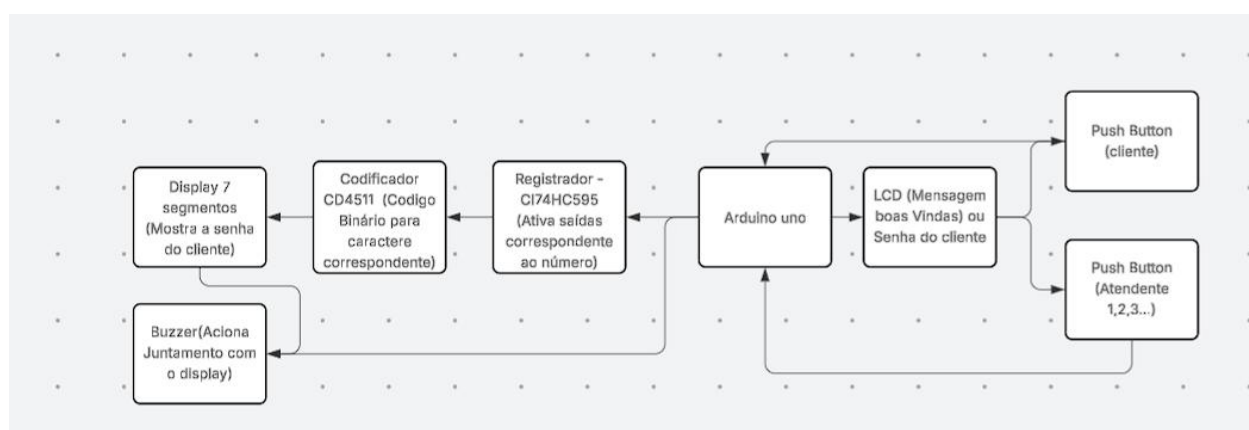
A organização no atendimento ao cliente é um fator essencial para o bom funcionamento de qualquer estabelecimento comercial. Em lojas de pequeno e médio porte, a ausência de um sistema de controle de filas pode gerar desordem, confusão e insatisfação por parte dos clientes, especialmente em horários de pico. Essa desorganização pode comprometer a experiência do consumidor e prejudicar a imagem do negócio.

Diante desse cenário, identificou-se a necessidade de uma solução simples, eficaz e de baixo custo que contribua para melhorar o atendimento, organizando a ordem de chegada dos clientes de forma clara e automatizada.

1.1.2 2. DESCRIÇÃO E OBJETIVO DO PROJETO

O presente projeto tem como objetivo desenvolver um sistema eletrônico de controle de senhas para atendimento ao cliente, voltado especialmente para estabelecimentos de pequeno e médio porte. A solução permite que o cliente solicite uma senha ao chegar à loja, sendo informado por meio de um display LCD. O sistema registra a ordem de chegada e, por meio de botões acionados pelos atendentes, atualiza a chamada da próxima senha, exibida em displays de sete segmentos. O projeto visa facilitar o fluxo de atendimento, proporcionar mais organização e melhorar a experiência do cliente no ambiente comercial.

2 DIAGRAMA EM BLOCOS DO PROJETO



3 DESCRIÇÃO DA SOLUÇÃO

O sistema inicia com uma mensagem de boas-vindas exibida no display LCD (16x2 I2C), aguardando que o cliente pressione o botão de chamada. Ao realizar essa ação, o Arduino executa a lógica de controle e envia uma mensagem ao LCD informando a senha atribuída ao cliente, além de orientá-lo a aguardar pelo atendimento.

Quando há clientes na fila, o atendente pode acionar um botão (push-button) localizado em sua respectiva bancada de atendimento. Cada bancada está equipada com um botão exclusivo e um par de displays de sete segmentos (cátodo comum), totalizando seis displays no projeto. Esses displays são responsáveis por mostrar a senha do próximo cliente e são controlados por um codificador (CD4511), que por sua vez é acionado por um registrador de deslocamento (CI 74HC595). Ao pressionar o botão, a senha exibida é atualizada automaticamente na bancada correspondente, e um buzzer é ativado para emitir um sinal sonoro, indicando a chamada do próximo cliente.

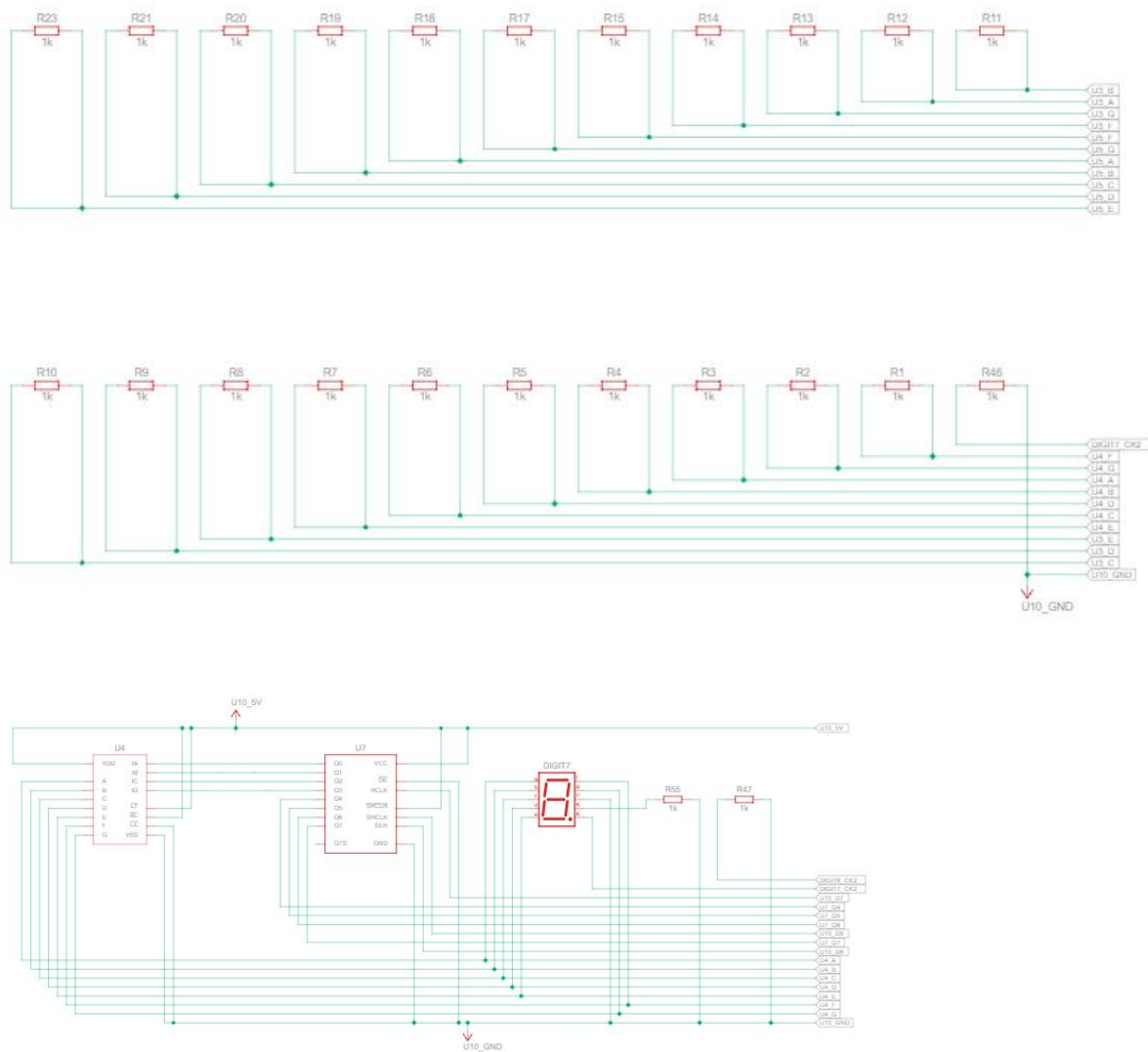
4 LISTA DE MATERIAIS

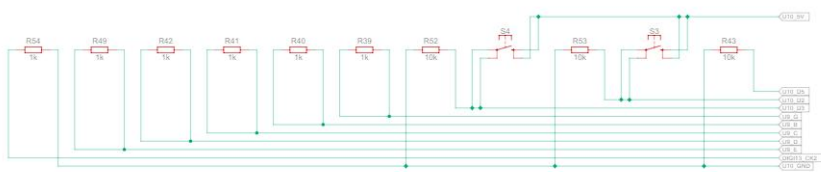
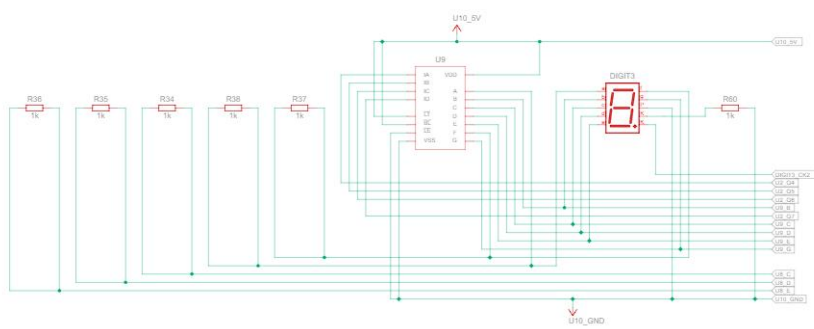
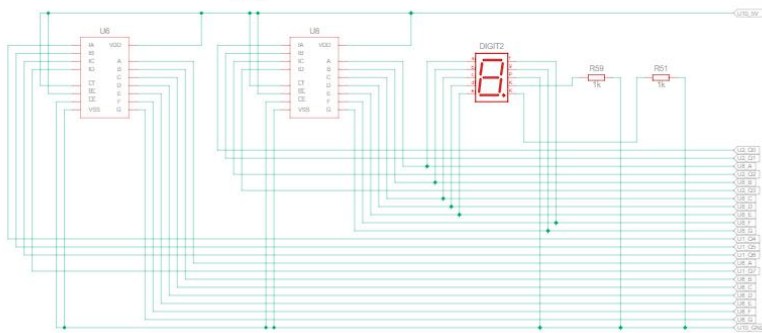
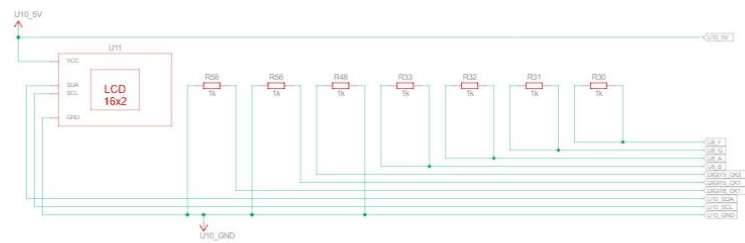
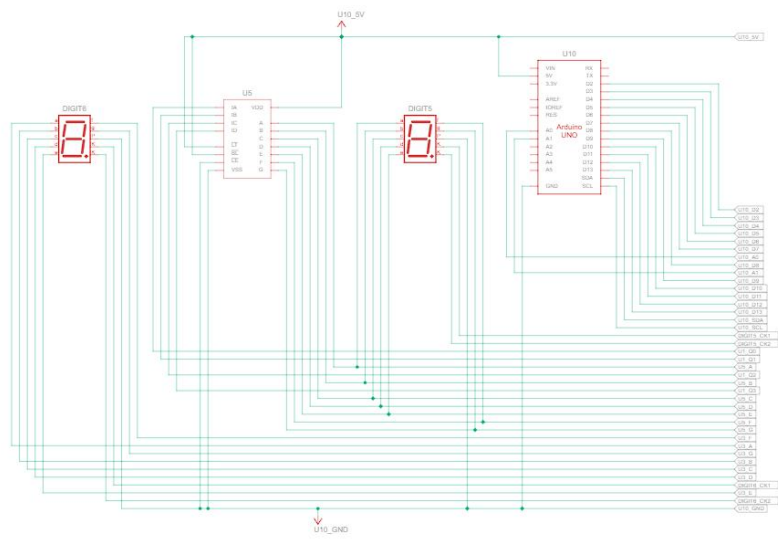
Tabela 1 – Lista de materiais.

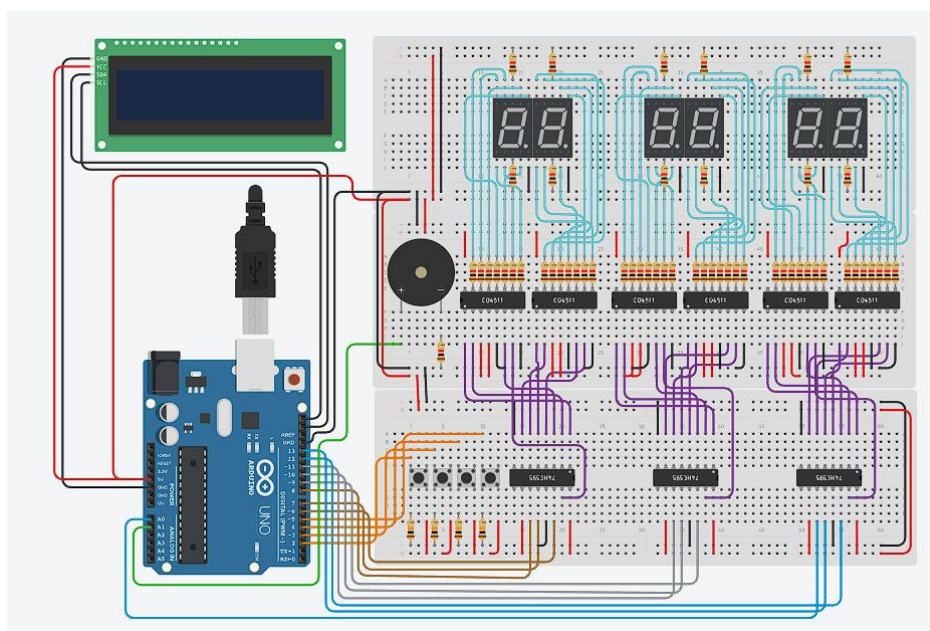
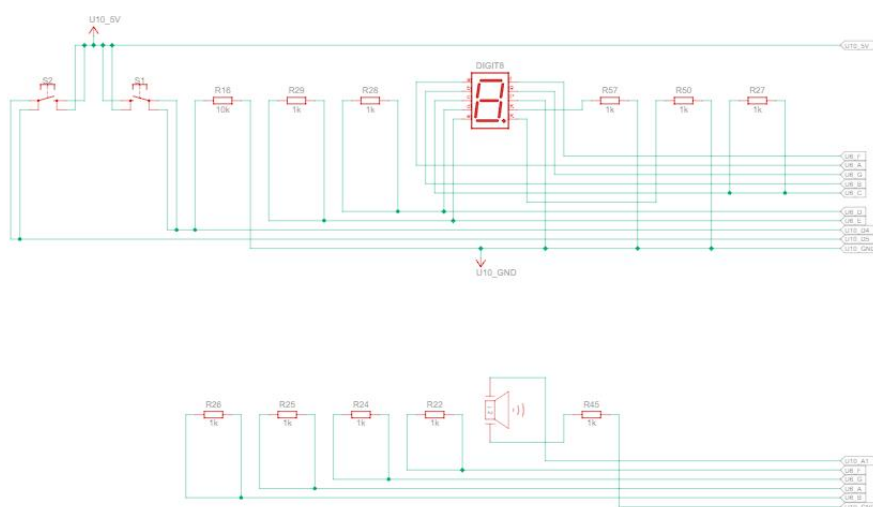
Item	Quantidade	Fonte
Display de LCD (I2C)	1	BETA Comercial Eletrônica
Display 7 segmentos (cátodo comum)	6	BETA Comercial Eletrônica
Arduino UNO	1	BETA Comercial Eletrônica
Buzzer	1	BETA Comercial Eletrônica
Protoboard	3	BETA Comercial Eletrônica
Resistor 1K Ohm	55	BETA Comercial Eletrônica
Resistor 10K Ohm	4	BETA Comercial Eletrônica
Registrador CI74HC595	3	BETA Comercial Eletrônica
Codificador CD4511	6	BETA Comercial Eletrônica
Push Button	4	BETA Comercial Eletrônica

Fonte: Autoria própria.

5 CIRCUITO ESQUEMÁTICO DO PROJETO







6 MANUAL DE OPERAÇÃO

1. Início do Sistema

Ao ser ligado, o sistema exibe automaticamente uma mensagem de boas-vindas no LCD, indicando que está pronto para uso.

2. Geração de Senhas para Novos Clientes

Para emitir uma nova senha, pressione o **primeiro botão** localizado à esquerda do painel. Esse botão é responsável por gerar senhas sequenciais para os clientes que estão chegando à loja. A senha será exibida no LCD, iniciando a fila de atendimento.

3. Chamadas nas Bancadas de Atendimento

Os **outros três botões** correspondem às **bancadas de atendimento**. Cada botão está vinculado a uma bancada específica. Quando houver clientes aguardando, o atendente deve pressionar o botão de sua bancada para chamar a **próxima senha da fila**.

6.1 LINK PARA O VÍDEO DEMONSTRATIVO NO YOUTUBE

7 CÓDIGO DE PROGRAMAÇÃO DO PROJETO DO ARDUINO

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3
4  // DEFINIÇÕES
5  #define endereco 0x27 // Endereços comuns: 0x27, 0x3F
6  #define colunas 16
7  #define linhas 2
8
9  // INSTANCIANDO OBJETOS
10 LiquidCrystal_I2C lcd(endereco, colunas, linhas);
11
12
13 // Pino conectado ao pino de trava (ST_CP) de 74HC595
14 const int latchPin_1 = 7;
15 const int latchPin_2 = 10;
16 const int latchPin_3 = 13;
17
18 // Pino conectado ao pino de clock (SH_CP) de 74HC595
19 const int clockPin_1 = 8;
20 const int clockPin_2 = 11;
21 const int clockPin_3 = A0;
22
23 //Pino conectado ao pino de data (DS) of 74HC595
24 const int dataPin_1 = 6;
25 const int dataPin_2 = 9;
26 const int dataPin_3 = 12;
27
28 // pinos dos Push buttons
29 const int PbCliente = 2;
30 const int PbAtendente_1 = 3;
31 const int PbAtendente_2 = 4;
32 const int PbAtendente_3 = 5;
```

```

33
34 // pino Buzzer
35 const int pinBuzzer = A1;
36
37 //Numero do cliente
38 int adCliente = 0;
39 int numCliente = 0;
40
41 //Buffers para os botões
42 volatile bool botao2 = false;
43 volatile bool botao3 = false;
44 volatile bool botao4 = false;
45 volatile bool botao5 = false;
46
47 volatile uint8_t ultimo_estado;
48
49 int seven_seg_digits[10][4] = { { 0,0,0,0 }, // = 0
50                                  { 0,0,0,1 }, // = 1
51                                  { 0,0,1,0 }, // = 2
52                                  { 0,0,1,1 }, // = 3
53                                  { 0,1,0,0 }, // = 4
54                                  { 0,1,0,1 }, // = 5
55                                  { 0,1,1,0 }, // = 6
56                                  { 0,1,1,1 }, // = 7
57                                  { 1,0,0,0 }, // = 8
58                                  { 1,0,0,1 } // = 9
59                                  };
60
61 void bemVindo() {
62     lcd.init(); // INICIA A COMUNICAÇÃO COM O DISPLAY
63     lcd.backlight(); // LIGA A ILUMINAÇÃO DO DISPLAY
64     lcd.clear(); // LIMPA O DISPLAY

```

```

65
66     lcd.print("Bem vindo a Loja!");
67     lcd.setCursor(0, 1); // POSICIONA O CURSOR NA PRIMEIRA COLUNA DA LINHA 2
68     lcd.print("Retire sua senha");
69 }
70
71 void senhaCliente(int numeroCliente) {
72     lcd.clear(); // LIMPA O DISPLAY
73
74     lcd.print("Aguarde chamada");
75     lcd.setCursor(0, 1); // POSICIONA O CURSOR NA PRIMEIRA COLUNA DA LINHA 2
76     lcd.print("Sua senha eh ");
77     lcd.print(numeroCliente);
78 }
79
80 void atendente(int* numeroCliente, int latchPin, int dataPin, int clockPin) {
81     int numero = *numeroCliente;
82
83     if (++numero <= adCliente) {
84         int num1=0, num2=0;
85         *numeroCliente = numero;
86
87         if((numero < 0) || (numero > 99)){
88             numero = 0;
89         }
90
91         num1=numero/10;
92         num2=numero%10;
93         // os bits que você deseja enviar
94         byte enviarBits = 0;
95         // desligue a saída para que os pinos não acendam
96         // enquanto você está trocando bits:
97         digitalWrite(latchPin, LOW);

```

```

98
99 //Escreve num1 (digito mais significativo)
100 bitWrite(enviarBits, 0, seven_seg_digits[num1][3]);
101 bitWrite(enviarBits, 1, seven_seg_digits[num1][2]);
102 bitWrite(enviarBits, 2, seven_seg_digits[num1][1]);
103 bitWrite(enviarBits, 3, seven_seg_digits[num1][0]);
104
105 //Escreve num2 (digito menos significativo)
106 bitWrite(enviarBits, 4, seven_seg_digits[num2][3]);
107 bitWrite(enviarBits, 5, seven_seg_digits[num2][2]);
108 bitWrite(enviarBits, 6, seven_seg_digits[num2][1]);
109 bitWrite(enviarBits, 7, seven_seg_digits[num2][0]);
110 // desloque os bits para fora:
111 shiftOut(dataPin, clockPin, MSBFIRST, enviarBits);
112 // ligue a saída para que os LEDs possam acender:
113 digitalWrite(latchPin, HIGH);
114 digitalWrite(pinBuzzer, HIGH);
115 delay(1000);
116 digitalWrite(pinBuzzer, LOW);
117 }
118
119 }
120
121
122 void setup() {
123 // Habilita o grupo de interrupções PCINT2 (PORTD → pinos D0 a D7)
124 PCICR |= (1 << PCIE2);
125
126 // Habilita as interrupções individuais nos pinos D2 a D5
127 PCMSK2 |= (1 << PCINT18); // D2
128 PCMSK2 |= (1 << PCINT19); // D3
129 PCMSK2 |= (1 << PCINT20); // D4
130 PCMSK2 |= (1 << PCINT21); // D5

```

```

130 PCMSK2 |= (1 << PCINT21); // D5
131
132 // inicializa com o estado real atual dos pinos D0-D7
133 ultimo_estado = PIND;
134
135 // Saídas para os displays de sete segmentos
136
137 pinMode(latchPin_1, OUTPUT);
138 pinMode(dataPin_1, OUTPUT);
139 pinMode(clockPin_1, OUTPUT);
140
141 pinMode(latchPin_2, OUTPUT);
142 pinMode(dataPin_2, OUTPUT);
143 pinMode(clockPin_2, OUTPUT);
144
145 pinMode(latchPin_3, OUTPUT);
146 pinMode(dataPin_3, OUTPUT);
147 pinMode(clockPin_3, OUTPUT);
148
149 //Saída buzzer
150 pinMode(pinBuzzer, OUTPUT);
151
152 //Estradas dos push buttons
153
154 pinMode(PbCliente, INPUT);
155 pinMode(PbAtendente_1, INPUT);
156 pinMode(PbAtendente_2, INPUT);
157 pinMode(PbAtendente_3, INPUT);
158
159 bemVindo();
160
161 Serial.begin(9600);
162

```

```

161 | Serial.begin(9600),
162 | }
163
164 void loop() {
165     //Se o cliete pressionou o botão
166     if (botao2) {
167         botao2 = false;
168
169         adCliente++;
170
171         if (adCliente > 99) {
172             adCliente = 0;
173         }
174
175         senhaCliente(adCliente);
176         delay(3000); //tempo de 3 segundos
177         bemVindo(); // mostra bem vindo para o proximo cliente
178     }
179
180     if (botao3) {
181         botao3 = false;
182         atendente(&numCliente, latchPin_1, dataPin_1, clockPin_1);
183     }
184
185     if (botao4) {
186         botao4 = false;
187         atendente(&numCliente, latchPin_2, dataPin_2, clockPin_2);
188     }
189
190     if (botao5) {
191         botao5 = false;
192         atendente(&numCliente, latchPin_3, dataPin_3, clockPin_3);
193     }

```

```

194 }
195
196 // Função chamada quando qualquer pino D0 a D7 muda de estado
197 ISR(PCINT2_vect) {
198     uint8_t estado_atual = PIND;
199
200     // Verifica transições de HIGH para LOW (borda de descida)
201     if ((estado_atual & (1 << PIND2)) == 0 && (ultimo_estado & (1 << PIND2)) != 0) {
202         botao2 = true;
203     }
204     if ((estado_atual & (1 << PIND3)) == 0 && (ultimo_estado & (1 << PIND3)) != 0) {
205         botao3 = true;
206     }
207     if ((estado_atual & (1 << PIND4)) == 0 && (ultimo_estado & (1 << PIND4)) != 0) {
208         botao4 = true;
209     }
210     if ((estado_atual & (1 << PIND5)) == 0 && (ultimo_estado & (1 << PIND5)) != 0) {
211         botao5 = true;
212     }
213
214     // Atualiza o último estado dos pinos
215     ultimo_estado = estado_atual;
216 }
217
218
219
220

```

8 SIMULAÇÃO DO PROJETO

A simulação do projeto pode ser visualizada no seguinte link:
<https://www.tinkercad.com/things/jX2mPzyTOi0-cool-elzing>

9 PRINCIPAIS DIFICULDADES ENCONTRADAS E SOLUÇÕES UTILIZADAS

Durante o desenvolvimento do sistema, a principal dificuldade encontrada esteve relacionada ao funcionamento do display LCD. Inicialmente, o componente apresentava apenas a retroiluminação ativa, sem exibir qualquer informação, o que indicava que a comunicação entre o Arduino e o módulo LCD não estava sendo estabelecida corretamente.

Após uma análise detalhada e diversas pesquisas, foi identificado que o problema estava associado à biblioteca inicialmente utilizada e à ausência da especificação correta do endereço I2C do módulo. A solução adotada envolveu a substituição da biblioteca por uma versão mais compatível com o modelo de display em uso, além da definição explícita do endereço I2C correspondente.

Com essas modificações, a comunicação foi devidamente estabelecida, e o display passou a exibir corretamente as informações do sistema.

10 CONCLUSÕES

O desenvolvimento deste projeto permitiu simular de forma eficiente o funcionamento de um sistema eletrônico de controle de senhas para atendimento ao cliente, voltado especialmente para estabelecimentos de pequeno e médio porte. A solução proposta demonstrou ser prática, acessível e funcional, contribuindo para a organização do fluxo de atendimento e melhorando a experiência do usuário final.

Durante a execução, foi possível aplicar conceitos de eletrônica digital, automação e programação com microcontroladores, integrando diversos componentes como botões, displays de sete segmentos, display LCD, buzzer, codificadores e registradores de deslocamento. A utilização do Arduino como unidade central de controle facilitou a implementação lógica e a comunicação entre os dispositivos.

Além de cumprir os requisitos propostos, o projeto evidenciou a importância da automação em contextos cotidianos, mostrando que soluções simples podem trazer melhorias significativas em termos de organização e eficiência. Dessa forma, este sistema se apresenta como uma base viável para aplicações reais, com possibilidade de futuras expansões, como integração com sistemas de banco de dados, controle remoto ou interfaces mais modernas de usuário.

10.1 TRABALHOS FUTUROS

Como proposta de continuidade deste projeto, sugere-se a integração do sistema com tecnologias sem fio, como Wi-Fi ou Bluetooth, com o objetivo de eliminar a necessidade de cabeamento entre os módulos. Além disso, pode-se considerar a implementação de um aplicativo móvel que permita ao cliente retirar a senha de forma remota, otimizando ainda mais o fluxo de atendimento. Também é possível substituir os displays convencionais por interfaces mais modernas, como telas OLED ou touch screen, proporcionando uma experiência mais interativa e intuitiva ao usuário.

REFERÊNCIAS

TEXAS INSTRUMENTS. *SN74HC595 – 8-bit serial-in, parallel-out shift register with output latches*. [S.l.], 2020. Disponível em: <https://www.ti.com/lit/ds/symlink/sn74hc595.pdf>. Acesso em: 10 jul. 2025.

TEXAS INSTRUMENTS. *CD4511B – BCD to 7-segment latch decoder driver*. [S.l.], 2017. Disponível em: <https://www.ti.com/lit/ds/symlink/cd4511b.pdf>. Acesso em: 10 jul. 2025.

KINGBRIGHT. *SC56-11 Series – 7 Segment Display Datasheet*. [S.l.], 2012. Disponível em: <https://www.kingbrightusa.com/images/catalog/spec/SC56-11GWA.pdf>. Acesso em: 10 jul. 2025.

HITACHI. *HD44780U – LCD Controller/Driver Datasheet*. [S.l.], 2000. Disponível em: <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>. Acesso em: 10 jul. 2025.