

```
1 //Compilação//
2 // gcc cliente_tcp.c -o cliente_tcp
3
4 //Execução//
5 // ./cliente_tcp <ip do servidor>
6
7 //-----
8 //Includes do programa//
9 #include <stdio.h>
10 #include <stdlib.h>
11 #include <string.h>
12 #include <unistd.h>
13 #include <sys/socket.h>
14 #include <arpa/inet.h>
15 #include <pthread.h>
16 #include <termios.h>
17 #include <fcntl.h>
18
19 //-----
20 //Variáveis globais//
21 unsigned short PORTA_ALARME = 4034;
22 unsigned short PORTA_GERAL = 8034;
23
24
25 //-----
26 //Menu do Cliente//
27
28 int menu(int opcao, int recursao)
29 {
30     int escolha;
31
32     //Opções de Escolha do Cliente//
33     switch(opcao)
34     {
35         //Menu geral
36         case 0:
37             printf("[1] - Temperatura\n");
38             printf("[2] - Presença\n");
39             printf("[3] - Portas/Janelas\n");
40             printf("[4] - Ar-condicionado\n");
41             printf("[5] - Lampadas\n");
42             printf("[6] - Alarme\n");
43             scanf("%d", &escolha);
44
45             if((escolha < 1) || (escolha > 6))
46                 return 0;
47
48             return menu(escolha, 0); //Retorna a propria função
49
50         //Menu temperatura
51         case 1:
52             printf("[1] - Externa\n");
53             printf("[2] - Interna da Sala\n");
54             printf("[3] - Interna do quarto\n");
55             scanf("%d", &escolha);
56
57             if((escolha < 1) || (escolha > 3))
58                 return 0;
59
60             return (1000 + 10*escolha);
61
62         //Menu Presenca
63         case 2:
64             printf("[1] - Entrada Principal\n");
65             printf("[2] - Entrada de Serviço\n");
66             printf("[3] - Garagem\n");
67             scanf("%d", &escolha);
```

```
68
69         if((escolha < 1) || (escolha > 3))
70             return 0;
71
72
73         return (2000 + 10*escolha);
74
75 //Menu Portas/Janelas
76 case 3:
77     printf("[1] - Porta principal\n");
78     printf("[2] - Porta de serviço\n");
79     printf("[3] - Porta garagem\n");
80     printf("[4] - Janela da sala\n");
81     printf("[5] - Janela do quarto\n");
82     printf("[6] - Janela da cozinha\n");
83     scanf("%d", &escolha);
84
85     if((escolha < 1) || (escolha > 6))
86         return 0;
87
88     return (3000 + 10*escolha);
89
90 //Menu Ar-condicionado
91 case 4:
92     printf("[1] - Sala\n");
93     printf("[2] - Quarto\n");
94     scanf("%d", &escolha);
95
96     if((escolha < 1) || (escolha > 2))
97         return 0;
98
99     return menu(7, (4000 + 10*escolha)); //Retorna o menu, na opção
estado (Ligado/Desligado)
100
101 //Menu Lampadas
102 case 5:
103     printf("[1] - Entrada principal\n");
104     printf("[2] - Entrada de serviço\n");
105     printf("[3] - Garagem\n");
106     printf("[4] - Sala\n");
107     printf("[5] - Quarto\n");
108     printf("[6] - Cozinha\n");
109     scanf("%d", &escolha);
110
111     if((escolha < 1) || (escolha > 6))
112         return 0;
113     return menu(7, (5000 + 10*escolha));
114
115 //Menu Alarme
116 case 6:
117     printf("[1] - Setor 1 (Sensores de Presença)\n");
118     printf("[2] - Setor 2 (Sensores de Portas/ Janelas)\n");
119     scanf("%d", &escolha);
120
121     if((escolha < 1) || (escolha > 2))
122         return 0;
123     return menu(7, (6000 + 10*escolha));
124
125 //Menu Estado Ligado ou Desligado
126 case 7:
127     printf("[1] - Liga\n");
128     printf("[0] - Desliga\n");
129     scanf("%d", &escolha);
130
131     if((escolha < 0) || (escolha > 1))
132         return 0;
133
```

```
134         return (escolha + recursao);
135
136     default:
137         printf("Opção Inválida.\n");
138         return 0;
139     }
140 }
141
142 //-----
143 //Funções de envio e recebimento de dados//
144
145 //Recebe float do servidor
146 void recebe_float(int *clienteSocket)
147 {
148     float temperatura;
149
150     if((recv(*clienteSocket, &temperatura, sizeof(float), 0)) < 0)
151         printf("Erro no recv()\n");
152
153     printf("Temperatura %f\n", temperatura);
154
155     close(*clienteSocket);
156 }
157
158 //Recebe char do servidor
159 void recebe_char(int *clienteSocket)
160 {
161     char mensagem;
162
163     if((recv(*clienteSocket, &mensagem, sizeof(char), 0)) < 0)
164         printf("Erro no recv()\n");
165
166     printf("Estado %c\n", mensagem);
167
168     close(*clienteSocket);
169 }
170
171 //Envia um request de para o servidor
172 void request(int *clienteSocket, int mensagem)
173 {
174     int bytesRecebidos;
175     int buffer;
176
177     if(send(*clienteSocket, &mensagem, sizeof(int), 0) != sizeof(int))
178         printf("Erro no envio: numero de bytes enviados diferente do esperado\n");
179 }
180
181 //-----
182
183 //Abertura do socket//
184 void socket_open(int *clienteSocket, struct sockaddr_in *servidorAddr, char *IP_Servidor, unsigned
short servidorPorta)
185 {
186     // Criar Socket
187     if((*clienteSocket = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
188         printf("Erro no socket()\n");
189
190     // Construir struct sockaddr_in
191     memset(servidorAddr, 0, sizeof(*servidorAddr)); // Zerando a estrutura de dados
192     servidorAddr->sin_family = AF_INET;
193     servidorAddr->sin_addr.s_addr = inet_addr(IP_Servidor);
194     servidorAddr->sin_port = htons(servidorPorta);
195
196     // Connect
197     if(connect(*clienteSocket, (struct sockaddr *) servidorAddr, sizeof(*servidorAddr)) < 0)
198         printf("Erro no connect()\n");
199 }
```

```
200 }
201 }
202
203
204 //-----
205 //Função principal//
206
207 int main(int argc, char *argv[]) {
208     //Variaveis locais//
209     int clienteSocket;
210     int escolha;
211     struct sockaddr_in servidorAddr;
212     unsigned short servidorPorta = PORTA_GERAL;
213     char *IP_Servidor;
214
215     if ((argc < 2) | (argc > 3))
216     {
217         printf("Uso: %s <IP do Servidor>\n", argv[0]);
218         exit(1);
219     }
220
221     //Ip via linha de comando
222     IP_Servidor = argv[1];
223
224
225     while(1)
226     {
227         if((escolha = menu(0, 0)) == 0)
228             printf("Escolha invalida\n");
229
230
231         else
232         {
233             system("clear");
234
235             //Define a porta a ser conectado//
236             if(escolha > 6000)
237                 servidorPorta = PORTA_ALARME;
238
239             else
240                 servidorPorta = PORTA_GERAL;
241
242             //Mostra a porta utilizada e a opção escolhida
243             printf("|Escolha %d, Porta %d|\n", escolha, servidorPorta);
244
245             //Abrimos a conexão
246             socket_open(&clienteSocket, &servidorAddr, IP_Servidor, servidorPorta);
247
248             //Enviamos um dado do menu
249             request(&clienteSocket, escolha);
250
251             //Recebemos o retorno do sensor, de acordo com o dado requerido
252             if((escolha > 1000) && (escolha < 2000))
253                 receive_float(&clienteSocket);
254
255             if((escolha > 2000) && (escolha < 4000))
256                 receive_char(&clienteSocket);
257
258             //Fechamos a conexão com o servidor
259             close(clienteSocket);
260
261         }
262     }
263
264 }
265
266 exit(0);
```

267 }