

# Cardápio digital de restaurante

Neste PDF está o código utilizado para criar um cardápio digital e o resultado

## Código

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>

#define DESCONTO_MINIMO 50.0// Valor mínimo para aplicar desconto
#define DESCONTO 0.10// 10% de desconto
#define MAX_ITENS 100 // Máximo de tipos de itens diferentes no pedido

// Estrutura para armazenar um item do pedido (nome, preço unitário, quantidade)
typedef struct {
    char nome[50];
    float preco_unitario;
    int quantidade;
} ItemPedido;

// Protótipos das funções
void exibirCardapio();
int buscarItem(ItemPedido pedidos[], int qtdPedidos, const char *nomeItem);

// Função para exibir o cardápio
void exibirCardapio() {
    printf("\n===== CARDAPIO DIGITAL =====\n");
    printf("Digite o nome do item exatamente como aparece abaixo:\n\n");
    printf("feijoadá- R$ 30,00\n");
    printf("lasanha- R$ 25,00\n");
    printf("suco- R$ 8,00\n");
    printf("refrigerante- R$ 6,00\n");
    printf("pudim- R$ 12,00\n");
    printf("sorvete- R$ 10,00\n");
    printf("sair- Finalizar pedido\n");
    printf("=====\n");
}

// Função para buscar um item no array de pedidos
```

```

// Retorna o índice do item se encontrado, ou -1 se não encontrado
int buscarItem(ItemPedido pedidos[], int qtdPedidos, const char *nomeItem) {
    for (int i = 0; i < qtdPedidos; i++) {
        // Compara o nome que está no pedido com o nome "limpo" (como no
        // cardápio)
        // Note que tive que usar 'strcasecmp' ou uma adaptação se for case-sensitive,
        // mas aqui vamos comparar com o nome exato que está guardado no struct.
        if (strcmp(pedidos[i].nome, nomeItem) == 0) {
            return i; // Item encontrado
        }
    }
    return -1; // Item não encontrado
}

```

```

int main() {
    setlocale(LC_ALL, "Portuguese"); // habilita acentos no Windows

    char opcao[50];
    float total = 0.0;

    // Array para armazenar os ITENS ÚNICOS do pedido
    ItemPedido pedidos[MAX_ITENS];
    int qtdPedidosUnicos = 0; // Quantidade de TIPOS de itens diferentes

    while (1) {
        exibirCardapio();
        printf("Escolha uma opcao do cardapio: ");
        scanf("%s", opcao);

        char nomeItem[50];
        float precoUnitario = 0.0;
        int itemValido = 1; // Flag para saber se a opcao é válida

        // 1. Identificar o item, seu nome "limpo" e preço unitário
        if (strcmp(opcao, "feijoadada") == 0) {
            strcpy(nomeItem, "Feijoadada");
            precoUnitario = 30.0;
        } else if (strcmp(opcao, "lasanha") == 0) {
            strcpy(nomeItem, "Lasanha");
            precoUnitario = 25.0;
        } else if (strcmp(opcao, "suco") == 0) {
            strcpy(nomeItem, "Suco Natural");
            precoUnitario = 8.0;
        } else if (strcmp(opcao, "refrigerante") == 0) {

```

```

        strcpy(nomItem, "Refrigerante");
        precoUnitario = 6.0;
    } else if (strcmp(opcao, "pudim") == 0) {
        strcpy(nomItem, "Pudim");
        precoUnitario = 12.0;
    } else if (strcmp(opcao, "sorvete") == 0) {
        strcpy(nomItem, "Sorvete");
        precoUnitario = 10.0;
    } else if (strcmp(opcao, "sair") == 0) {
        break; // Sai do loop para finalizar o pedido
    } else {
        printf("Opcao invalida! Tente novamente.\n");
        itemValido = 0;
    }
}

// 2. Processar a adiç o do item (se for v lido)
if (itemValido) {
    // Tenta encontrar o item j  existente no pedido
    int indice = buscarItem(pedidos, qtdPedidosUnicos, nomItem);

    if (indice != -1) {
        // Item encontrado: apenas incrementa a quantidade
        pedidos[indice].quantidade++;
    } else {
        // Item n o encontrado: adiciona como um novo registro
        if (qtdPedidosUnicos < MAX_ITENS) {
            strcpy(pedidos[qtdPedidosUnicos].nome, nomItem);
            pedidos[qtdPedidosUnicos].preco_unitario = precoUnitario;
            pedidos[qtdPedidosUnicos].quantidade = 1;
            qtdPedidosUnicos++;
        } else {
            printf("Limite de itens diferentes no pedido atingido!\n");
        }
    }
    printf("Voc  adicionou mais um(a) %s ao pedido.\n", nomItem);
}
}

// --- RESUMO E C LCULO FINAL ---

// 1. Recalcula o total a partir dos itens e quantidades
total = 0.0;
for (int i = 0; i < qtdPedidosUnicos; i++) {
    total += pedidos[i].preco_unitario * pedidos[i].quantidade;
}

```

```

}

printf("\n===== RESUMO DO PEDIDO =====\n");
if (qtdPedidosUnicos == 0) {
    printf("Nenhum item selecionado.\n");
} else {
    printf("Itens escolhidos:\n");
    for (int i = 0; i < qtdPedidosUnicos; i++) {
        float subtotalItem = pedidos[i].preco_unitario * pedidos[i].quantidade;
        printf(" - %d x %s (R$ %.2f un.) = R$ %.2f\n",
            pedidos[i].quantidade,
            pedidos[i].nome,
            pedidos[i].preco_unitario,
            subtotalItem);
    }
}

printf("\nSubtotal: R$ %.2f\n", total);

// 2. Aplica desconto se atingir o mínimo
float valorFinal = total;
if (total >= DESCONTO_MINIMO) {
    float valorDesconto = total * DESCONTO;
    printf("Desconto aplicado (10%%): R$ %.2f\n", valorDesconto);
    valorFinal = total - valorDesconto;
} else {
    printf("Nenhum desconto aplicado.\n");
}

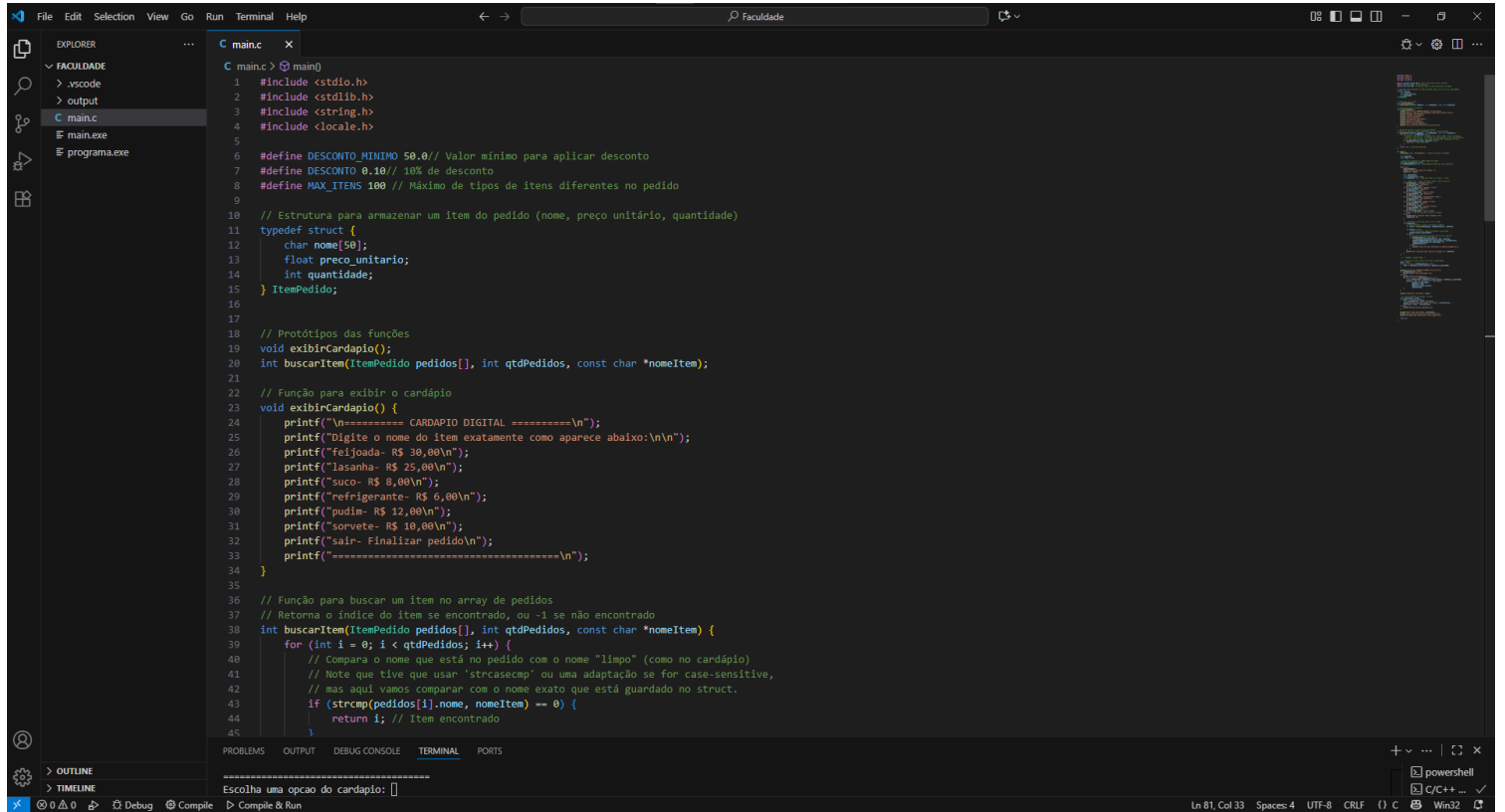
printf("Valor final: R$ %.2f\n", valorFinal);
printf("===== \n");
printf("Obrigado pela preferencia! Volte sempre!\n");

return 0;
}

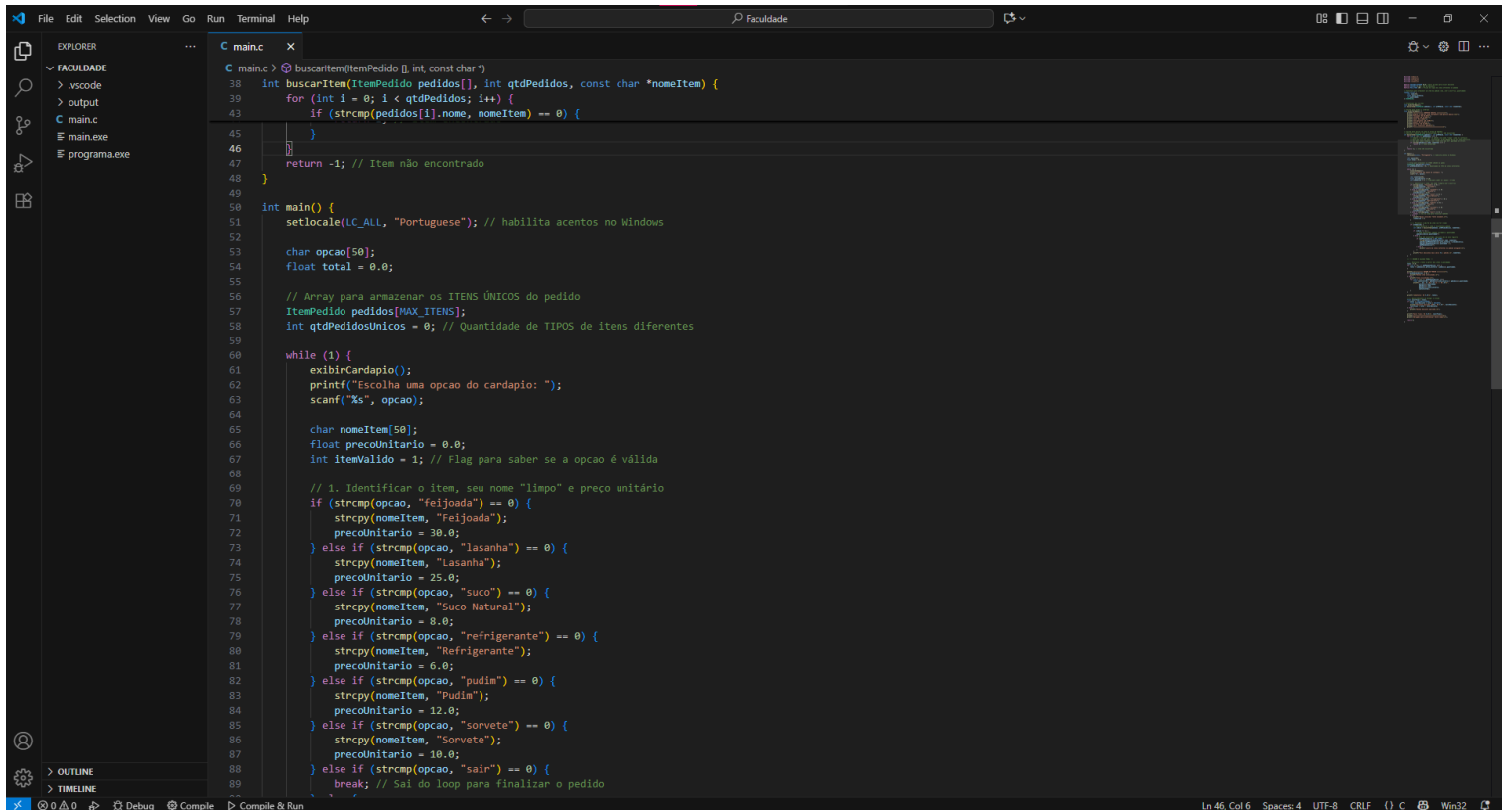
```

---

# Código feito



```
C main.c X
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <locale.h>
5
6 #define DESCONTO_MINIMO 50.0 // Valor mínimo para aplicar desconto
7 #define DESCONTO 0.10 // 10% de desconto
8 #define MAX_ITENS 100 // Máximo de tipos de Itens diferentes no pedido
9
10 // Estrutura para armazenar um item do pedido (nome, preço unitário, quantidade)
11 typedef struct {
12     char nome[50];
13     float preco_unitario;
14     int quantidade;
15 } ItemPedido;
16
17
18 // Protótipos das funções
19 void exibirCardapio();
20 int buscarItem(ItemPedido pedidos[], int qtdPedidos, const char *nomeItem);
21
22 // Função para exibir o cardápio
23 void exibirCardapio() {
24     printf("\n===== CARDAPIO DIGITAL =====\n");
25     printf("Digite o nome do item exatamente como aparece abaixo:\n\n");
26     printf("Feijoadá- R$ 30,00\n");
27     printf("Lasanha- R$ 25,00\n");
28     printf("Suco- R$ 8,00\n");
29     printf("Refrigerante- R$ 6,00\n");
30     printf("Pudim- R$ 12,00\n");
31     printf("Sorvete- R$ 10,00\n");
32     printf("Sair- Finalizar pedido\n");
33     printf("===== \n");
34 }
35
36 // Função para buscar um item no array de pedidos
37 // Retorna o índice do item se encontrado, ou -1 se não encontrado
38 int buscarItem(ItemPedido pedidos[], int qtdPedidos, const char *nomeItem) {
39     for (int i = 0; i < qtdPedidos; i++) {
40         // Compara o nome que está no pedido com o nome "limpo" (como no cardápio)
41         // Note que tive que usar 'strcasecmp' ou uma adaptação se for case-sensitive,
42         // mas aqui vamos comparar com o nome exato que está guardado no struct.
43         if (strcmp(pedidos[i].nome, nomeItem) == 0) {
44             return i; // Item encontrado
45         }
46     }
47     return -1; // Item não encontrado
48 }
49
50 int main() {
51     setlocale(LC_ALL, "Portuguese"); // habilita acentos no Windows
52
53     char opcao[50];
54     float total = 0.0;
55
56     // Array para armazenar os ITENS ÚNICOS do pedido
57     ItemPedido pedidos[MAX_ITENS];
58     int qtdPedidosUnicos = 0; // Quantidade de TIPOS de itens diferentes
59
60     while (1) {
61         exibirCardapio();
62         printf("Escolha uma opcao do cardapio: ");
63         scanf("%s", opcao);
64
65         char nomeItem[50];
66         float precoUnitario = 0.0;
67         int itemValido = 1; // Flag para saber se a opcao é válida
68
69         // 1. Identificar o item, seu nome "limpo" e preço unitário
70         if (strcmp(opcao, "feijoadá") == 0) {
71             strcpy(nomeItem, "Feijoadá");
72             precoUnitario = 30.0;
73         } else if (strcmp(opcao, "lasanha") == 0) {
74             strcpy(nomeItem, "Lasanha");
75             precoUnitario = 25.0;
76         } else if (strcmp(opcao, "suco") == 0) {
77             strcpy(nomeItem, "Suco Natural");
78             precoUnitario = 8.0;
79         } else if (strcmp(opcao, "refrigerante") == 0) {
80             strcpy(nomeItem, "Refrigerante");
81             precoUnitario = 6.0;
82         } else if (strcmp(opcao, "pudim") == 0) {
83             strcpy(nomeItem, "Pudim");
84             precoUnitario = 12.0;
85         } else if (strcmp(opcao, "sorvete") == 0) {
86             strcpy(nomeItem, "Sorvete");
87             precoUnitario = 10.0;
88         } else if (strcmp(opcao, "sair") == 0) {
89             break; // Sai do loop para finalizar o pedido
90         }
91     }
92 }
```



```
C main.c X
38 int buscarItem(ItemPedido pedidos[], int qtdPedidos, const char *nomeItem) {
39     for (int i = 0; i < qtdPedidos; i++) {
40         if (strcmp(pedidos[i].nome, nomeItem) == 0) {
41             return i;
42         }
43     }
44     return -1; // Item não encontrado
45 }
46
47
48 }
49
50 int main() {
51     setlocale(LC_ALL, "Portuguese"); // habilita acentos no Windows
52
53     char opcao[50];
54     float total = 0.0;
55
56     // Array para armazenar os ITENS ÚNICOS do pedido
57     ItemPedido pedidos[MAX_ITENS];
58     int qtdPedidosUnicos = 0; // Quantidade de TIPOS de itens diferentes
59
60     while (1) {
61         exibirCardapio();
62         printf("Escolha uma opcao do cardapio: ");
63         scanf("%s", opcao);
64
65         char nomeItem[50];
66         float precoUnitario = 0.0;
67         int itemValido = 1; // Flag para saber se a opcao é válida
68
69         // 1. Identificar o item, seu nome "limpo" e preço unitário
70         if (strcmp(opcao, "feijoadá") == 0) {
71             strcpy(nomeItem, "Feijoadá");
72             precoUnitario = 30.0;
73         } else if (strcmp(opcao, "lasanha") == 0) {
74             strcpy(nomeItem, "Lasanha");
75             precoUnitario = 25.0;
76         } else if (strcmp(opcao, "suco") == 0) {
77             strcpy(nomeItem, "Suco Natural");
78             precoUnitario = 8.0;
79         } else if (strcmp(opcao, "refrigerante") == 0) {
80             strcpy(nomeItem, "Refrigerante");
81             precoUnitario = 6.0;
82         } else if (strcmp(opcao, "pudim") == 0) {
83             strcpy(nomeItem, "Pudim");
84             precoUnitario = 12.0;
85         } else if (strcmp(opcao, "sorvete") == 0) {
86             strcpy(nomeItem, "Sorvete");
87             precoUnitario = 10.0;
88         } else if (strcmp(opcao, "sair") == 0) {
89             break; // Sai do loop para finalizar o pedido
90         }
91     }
92 }
```

```
File Edit Selection View Go Run Terminal Help
Faculdade

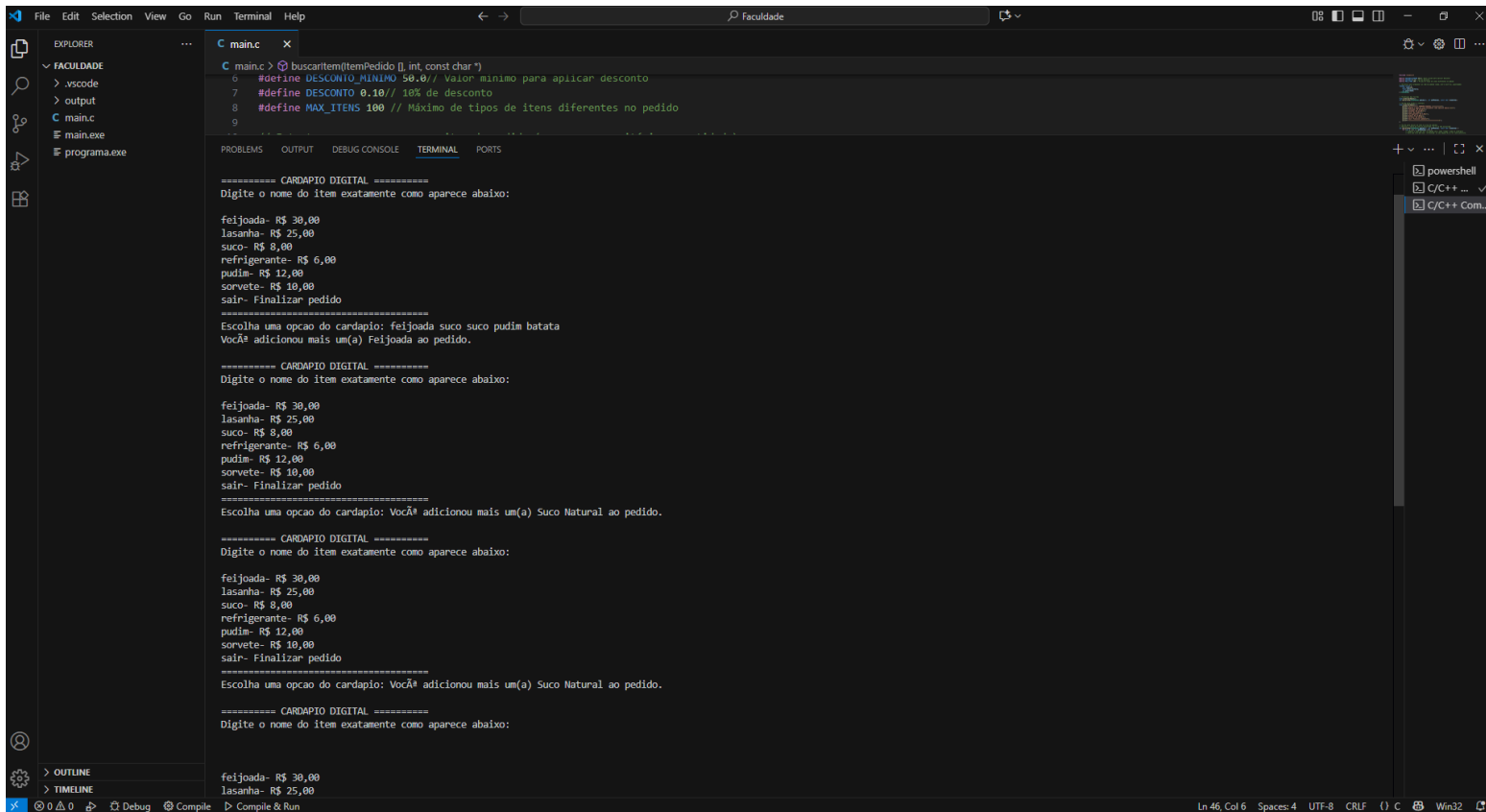
C main.c X
C main.c > buscarItem(ItemPedido [], int, const char *)
58 int main() {
60 while (1) {
88 } else if (strcmp(opcao, "sair") == 0) {
89 break; // Sai do loop para finalizar o pedido
90 } else {
91 printf("Opcao invalida! Tente novamente.\n");
92 itemValido = 0;
93 }
94
95 // 2. Processar a adição do item (se for válido)
96 if (itemValido) {
97 // Tenta encontrar o item já existente no pedido
98 int indice = buscarItem(pedidos, qtdPedidosUnicos, nomeItem);
99
100 if (indice != -1) {
101 // Item encontrado: apenas incrementa a quantidade
102 pedidos[indice].quantidade++;
103 } else {
104 // Item não encontrado: adiciona como um novo registro
105 if (qtdPedidosUnicos < MAX_ITENS) {
106 strcpy(pedidos[qtdPedidosUnicos].nome, nomeItem);
107 pedidos[qtdPedidosUnicos].preco_unitario = precoUnitario;
108 pedidos[qtdPedidosUnicos].quantidade = 1;
109 qtdPedidosUnicos++;
110 } else {
111 printf("Limite de itens diferentes no pedido atingido!\n");
112 }
113 }
114 printf("Você adicionou mais um(a) %s ao pedido.\n", nomeItem);
115 }
116 }
117
118 // --- RESUMO E CÁLCULO FINAL ---
119
120 // 1. Recalcula o total a partir dos itens e quantidades
121 total = 0.0;
122 for (int i = 0; i < qtdPedidosUnicos; i++) {
123 total += pedidos[i].preco_unitario * pedidos[i].quantidade;
124 }
125
126 printf("\n----- RESUMO DO PEDIDO ----- \n");
127 if (qtdPedidosUnicos == 0) {
128 printf("Nenhum item selecionado.\n");
129 } else {
130 printf("Itens escolhidos:\n");
131 for (int i = 0; i < qtdPedidosUnicos; i++) {
132 float subtotalItem = pedidos[i].preco_unitario * pedidos[i].quantidade;
133 printf("  - %d x %s (R$ %.2f un.) = R$ %.2f\n",
134        pedidos[i].quantidade,
135        pedidos[i].nome,
136        pedidos[i].preco_unitario,
137        subtotalItem);
138 }
139 }
140
141 printf("\nSubtotal: R$ %.2f\n", total);
142
143 // 2. Aplica desconto se atingir o mínimo
144 float valorFinal = total;
145 if (total >= DESCONTO_MINIMO) {
146 float valorDesconto = total * DESCONTO;
147 printf("Desconto aplicado (10%%): R$ %.2f\n", valorDesconto);
148 valorFinal = total - valorDesconto;
149 } else {
150 printf("Nenhum desconto aplicado.\n");
151 }
152
153 printf("Valor final: R$ %.2f\n", valorFinal);
154 printf("----- \n");
155 printf("Obrigado pela preferencia! Volte sempre!\n");
156
157 return 0;
158 }
```

```
File Edit Selection View Go Run Terminal Help
Faculdade

C main.c X
C main.c > buscarItem(ItemPedido [], int, const char *)
58 int main() {
60 while (1) {
96 if (itemValido) {
115 }
116 }
117
118 // --- RESUMO E CÁLCULO FINAL ---
119
120 // 1. Recalcula o total a partir dos itens e quantidades
121 total = 0.0;
122 for (int i = 0; i < qtdPedidosUnicos; i++) {
123 total += pedidos[i].preco_unitario * pedidos[i].quantidade;
124 }
125
126 printf("\n----- RESUMO DO PEDIDO ----- \n");
127 if (qtdPedidosUnicos == 0) {
128 printf("Nenhum item selecionado.\n");
129 } else {
130 printf("Itens escolhidos:\n");
131 for (int i = 0; i < qtdPedidosUnicos; i++) {
132 float subtotalItem = pedidos[i].preco_unitario * pedidos[i].quantidade;
133 printf("  - %d x %s (R$ %.2f un.) = R$ %.2f\n",
134        pedidos[i].quantidade,
135        pedidos[i].nome,
136        pedidos[i].preco_unitario,
137        subtotalItem);
138 }
139 }
140
141 printf("\nSubtotal: R$ %.2f\n", total);
142
143 // 2. Aplica desconto se atingir o mínimo
144 float valorFinal = total;
145 if (total >= DESCONTO_MINIMO) {
146 float valorDesconto = total * DESCONTO;
147 printf("Desconto aplicado (10%%): R$ %.2f\n", valorDesconto);
148 valorFinal = total - valorDesconto;
149 } else {
150 printf("Nenhum desconto aplicado.\n");
151 }
152
153 printf("Valor final: R$ %.2f\n", valorFinal);
154 printf("----- \n");
155 printf("Obrigado pela preferencia! Volte sempre!\n");
156
157 return 0;
158 }
```

## Resultado

Aqui no resultado podemos ver o preço total, desconto, se a pessoa coloca um item que não tem no catálogo e etc...



```
C main.c > buscarItem(itemPedido [], int, const char *)
6 #define DESCONTO_MINIMO 50.0 // Valor minimo para aplicar desconto
7 #define DESCONTO 0.10 // 10% de desconto
8 #define MAX_ITENS 100 // Máximo de tipos de itens diferentes no pedido
9

===== CARDAPIO DIGITAL =====
Digite o nome do item exatamente como aparece abaixo:

feijoad- R$ 30,00
lasanha- R$ 25,00
suco- R$ 8,00
refrigerante- R$ 6,00
pudim- R$ 12,00
sorvete- R$ 10,00
sair- Finalizar pedido
=====
Escolha uma opcao do cardapio: feijoad suco pudim batata
Você adicionou mais um(a) Feijoad ao pedido.

===== CARDAPIO DIGITAL =====
Digite o nome do item exatamente como aparece abaixo:

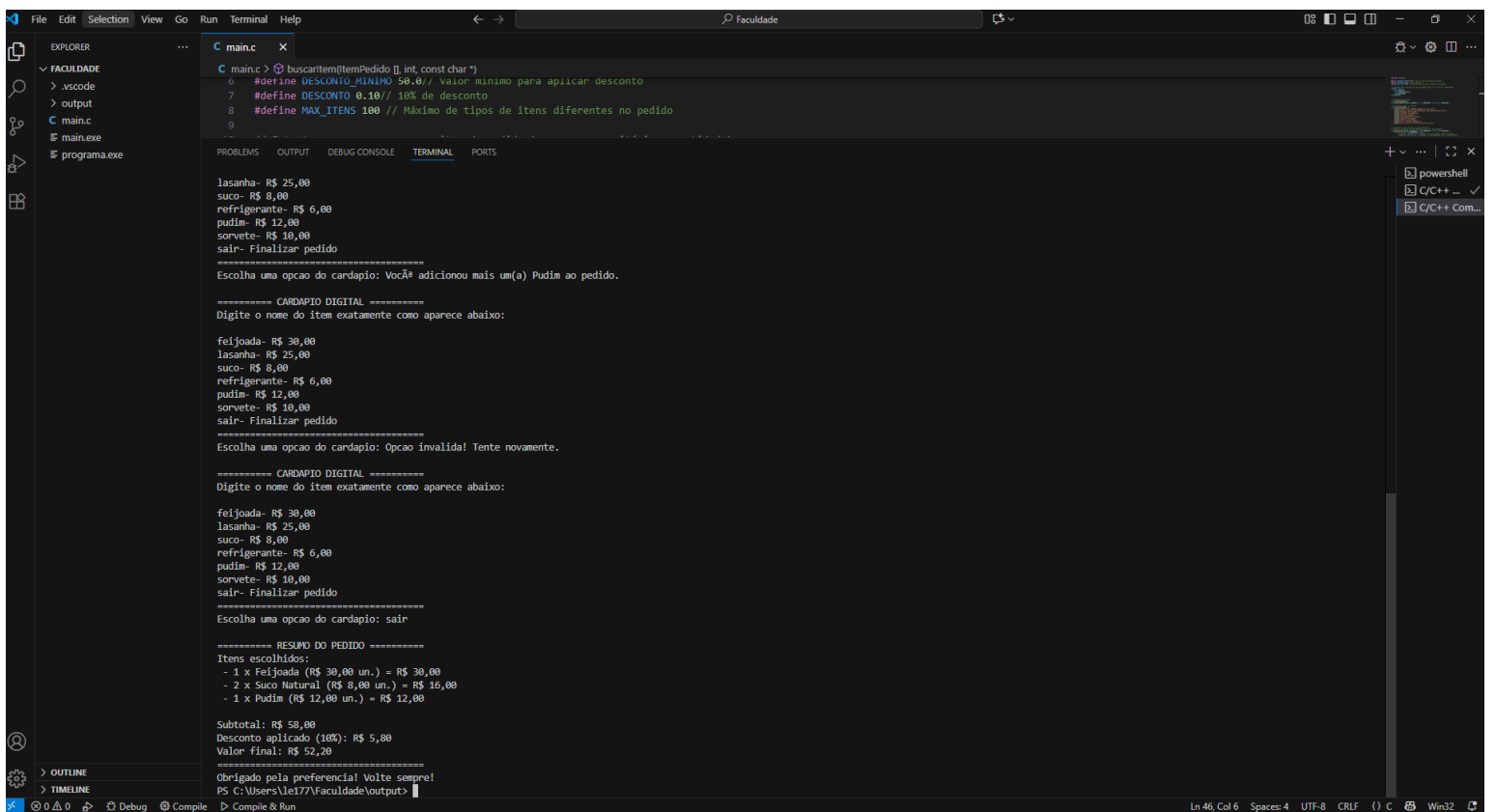
feijoad- R$ 30,00
lasanha- R$ 25,00
suco- R$ 8,00
refrigerante- R$ 6,00
pudim- R$ 12,00
sorvete- R$ 10,00
sair- Finalizar pedido
=====
Escolha uma opcao do cardapio: Você adicionou mais um(a) Suco Natural ao pedido.

===== CARDAPIO DIGITAL =====
Digite o nome do item exatamente como aparece abaixo:

feijoad- R$ 30,00
lasanha- R$ 25,00
suco- R$ 8,00
refrigerante- R$ 6,00
pudim- R$ 12,00
sorvete- R$ 10,00
sair- Finalizar pedido
=====
Escolha uma opcao do cardapio: Você adicionou mais um(a) Suco Natural ao pedido.

===== CARDAPIO DIGITAL =====
Digite o nome do item exatamente como aparece abaixo:

feijoad- R$ 30,00
lasanha- R$ 25,00
suco- R$ 8,00
refrigerante- R$ 6,00
pudim- R$ 12,00
sorvete- R$ 10,00
sair- Finalizar pedido
=====
Escolha uma opcao do cardapio: Você adicionou mais um(a) Suco Natural ao pedido.
```



```
C main.c > buscarItem(itemPedido [], int, const char *)
6 #define DESCONTO_MINIMO 50.0 // Valor minimo para aplicar desconto
7 #define DESCONTO 0.10 // 10% de desconto
8 #define MAX_ITENS 100 // Máximo de tipos de itens diferentes no pedido
9

lasanha- R$ 25,00
suco- R$ 8,00
refrigerante- R$ 6,00
pudim- R$ 12,00
sorvete- R$ 10,00
sair- Finalizar pedido
=====
Escolha uma opcao do cardapio: Você adicionou mais um(a) Pudim ao pedido.

===== CARDAPIO DIGITAL =====
Digite o nome do item exatamente como aparece abaixo:

feijoad- R$ 30,00
lasanha- R$ 25,00
suco- R$ 8,00
refrigerante- R$ 6,00
pudim- R$ 12,00
sorvete- R$ 10,00
sair- Finalizar pedido
=====
Escolha uma opcao do cardapio: Opcao invalida! Tente novamente.

===== CARDAPIO DIGITAL =====
Digite o nome do item exatamente como aparece abaixo:

feijoad- R$ 30,00
lasanha- R$ 25,00
suco- R$ 8,00
refrigerante- R$ 6,00
pudim- R$ 12,00
sorvete- R$ 10,00
sair- Finalizar pedido
=====
Escolha uma opcao do cardapio: sair

===== RESUMO DO PEDIDO =====
Itens escolhidos:
- 1 x Feijoad (R$ 30,00 un.) = R$ 30,00
- 2 x Suco Natural (R$ 8,00 un.) = R$ 16,00
- 1 x Pudim (R$ 12,00 un.) = R$ 12,00

Subtotal: R$ 58,00
Desconto aplicado (10%): R$ 5,80
Valor final: R$ 52,20

Obrigado pela preferencial! Volte sempre!
PS C:\Users\le177\Faculdade\output>
```