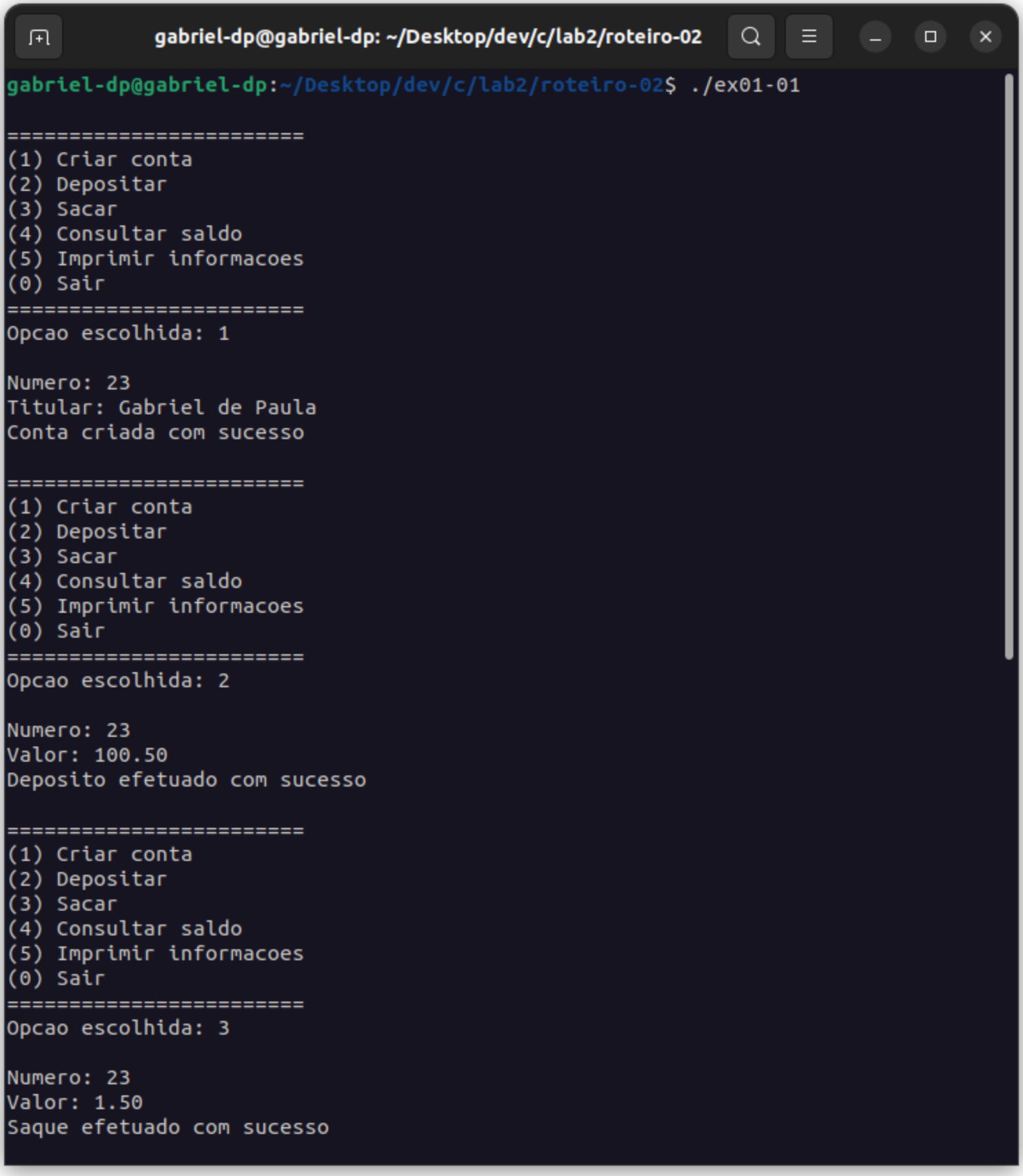
roteiro-02/ex01-01.c

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

#define NOME_CHARS 50
#define CONTAS_MAX 100

typedef struct {
    int numero;
    char titular[NOME_CHARS + 1];
    double saldo;
} ContaBancaria;

typedef struct {
    ContaBancaria contas[CONTAS_MAX];
    int quantidade;
} Banco;

enum OPCOES {
    SAIR = 0,
    CRIAR,
    DEPOSITAR,
    SACAR,
    CONSULTAR,
    IMPRIMIR
};

Banco criarBanco() {
    Banco novo;
    novo.quantidade = 0;
    return novo;
}

void criarConta(ContaBancaria* c, int numero, char* titular) {
    c->numero = numero;
    strcpy(c->titular, titular);
    c->saldo = 0.0;
}

void depositar(ContaBancaria* c, double valor) {
    c->saldo += valor;
}

void sacar(ContaBancaria* c, double valor) {
    if (c->saldo >= valor) {
        c->saldo -= valor;
    }
}
```

```c

double consultarSaldo(ContaBancaria* c) {
    return c->saldo;
}

void imprimirInfo(ContaBancaria* c) {
    printf("Numero: %d\n", c->numero);
    printf("Titular: %s\n", c->titular);
    printf("Saldo: %.2lf\n", c->saldo);
}

int getOpcao() {
    int opcao;

    printf("\n========================\n");
    printf("(%d) Criar conta\n", CRIAR);
    printf("(%d) Depositar\n", DEPOSITAR);
    printf("(%d) Sacar\n", SACAR);
    printf("(%d) Consultar saldo\n", CONSULTAR);
    printf("(%d) Imprimir informacoes\n", IMPRIMIR);
    printf("(%d) Sair\n", SAIR);
    printf("========================\n");

    printf("Opcao escolhida: ");
    scanf("%d", &opcao);

    printf("\n");
    return opcao;
}

ContaBancaria* getConta(int numero, Banco* banco) {
    for (int i = 0; i < banco->quantidade; i++) {
        if (banco->contas[i].numero == numero) {
            return &banco->contas[i];
        }
    }
    return NULL;
}

int main() {
    Banco banco = criarBanco();
    int numero;
    double valor;
    char titular[NOME_CHARS + 1];
    ContaBancaria* conta;

    int sair = 0;
    while (!sair) {
        switch (getOpcao()) {
            case SAIR:
                sair = 1;
                break;
```

```
            case CRIAR:
                printf("Numero: ");
                scanf("%d", &numero);
                setbuf(stdin, NULL);
                printf("Titular: ");
                fgets(titular, NOME_CHARS + 1, stdin);
                if (titular[strlen(titular) - 1] == '\n')
                    titular[strlen(titular) - 1] = '\0';
                setbuf(stdin, NULL);

                conta = getConta(numero, &banco);
                if (conta == NULL) {
                    criarConta(&banco.contas[banco.quantidade], numero,
titular);
                    banco.quantidade++;
                    printf("Conta criada com sucesso\n");
                } else {
                    printf("Conta ja existente\n");
                }
                break;
            case DEPOSITAR:
                printf("Numero: ");
                scanf("%d", &numero);
                printf("Valor: ");
                scanf("%lf", &valor);

                conta = getConta(numero, &banco);
                if (conta != NULL) {
                    depositar(conta, valor);
                    printf("Deposito efetuado com sucesso\n");
                } else {
                    printf("Conta nao encontrada\n");
                }
                break;
            case SACAR:
                printf("Numero: ");
                scanf("%d", &numero);
                printf("Valor: ");
                scanf("%lf", &valor);

                conta = getConta(numero, &banco);
                if (conta != NULL) {
                    sacar(conta, valor);
                    printf("Saque efetuado com sucesso\n");
                } else {
                    printf("Conta nao encontrada\n");
                }
                break;
            case CONSULTAR:
                printf("Numero: ");
                scanf("%d", &numero);

                conta = getConta(numero, &banco);
```

```
153                 if (conta != NULL) {
154                     printf("Saldo = %2lf\n", consultarSaldo(conta));
155                 } else {
156                     printf("Conta nao encontrada\n");
157                 }
158                 break;
159             case IMPRIMIR:
160                 printf("Numero: ");
161                 scanf("%d", &numero);
162 
163                 conta = getConta(numero, &banco);
164                 if (conta != NULL) {
165                     imprimirInfo(conta);
166                 } else {
167                     printf("Conta nao encontrada\n");
168                 }
169                 break;
170             default:
171                 printf("Opcao invalida - Tente novamente\n");
172                 break;
173         }
174     }
175 
176     return 0;
177 }
```


gabriel-dp@gabriel-dp: ~/Desktop/dev/c/lab2/roteiro-02
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-02$ ./ex01-01
========================
(1) Criar conta
(2) Depositar
(3) Sacar
(4) Consultar saldo
(5) Imprimir informacoes
(0) Sair
========================
Opcao escolhida: 1
Numero: 23
Titular: Gabriel de Paula
Conta criada com sucesso
========================
(1) Criar conta
(2) Depositar
(3) Sacar
(4) Consultar saldo
(5) Imprimir informacoes
(0) Sair
========================
Opcao escolhida: 2
Numero: 23
Valor: 100.50
Deposito efetuado com sucesso
========================
(1) Criar conta
(2) Depositar
(3) Sacar
(4) Consultar saldo
(5) Imprimir informacoes
(0) Sair
========================
Opcao escolhida: 3
Numero: 23
Valor: 1.50
Saque efetuado com sucesso

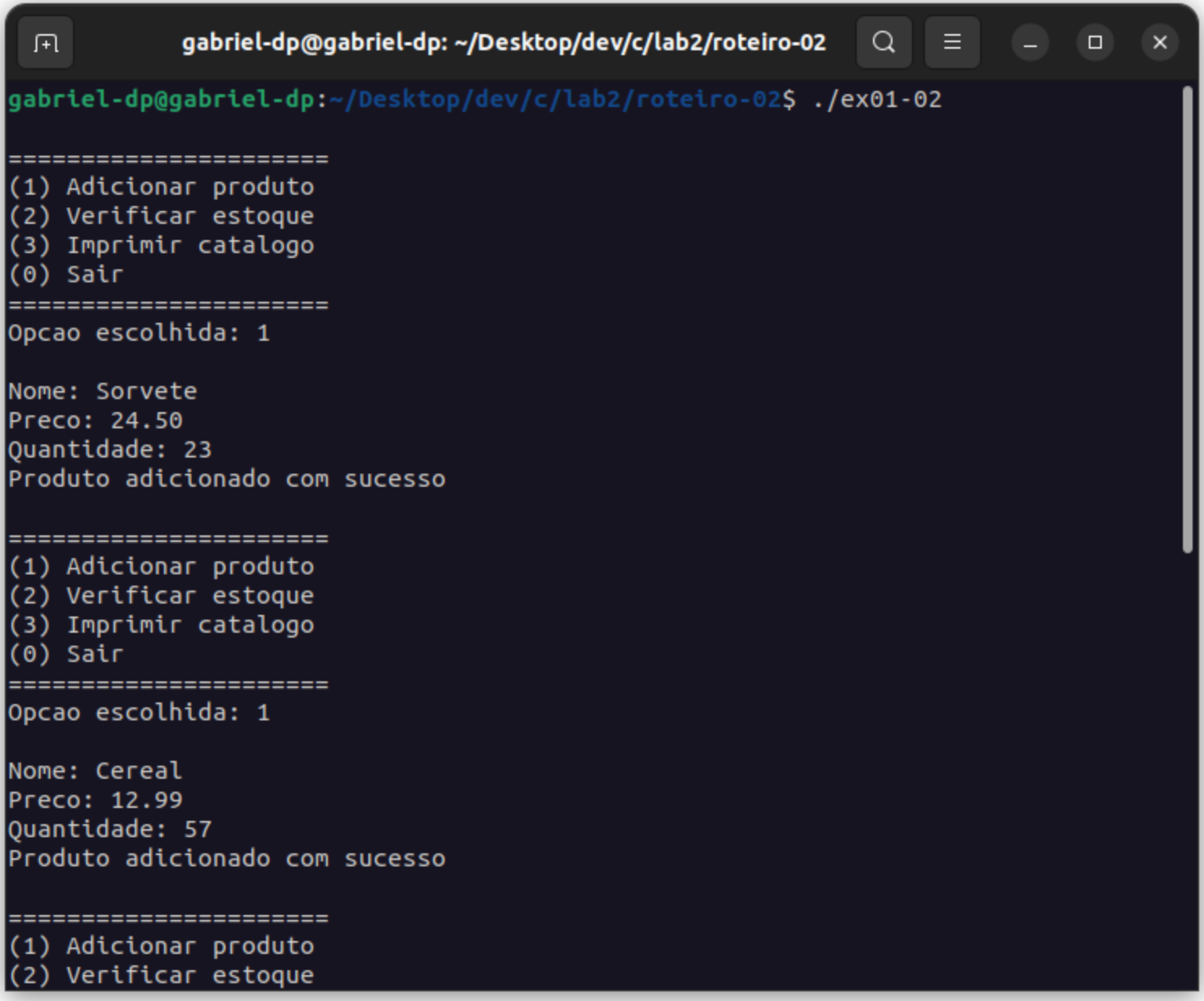
```
Numero: 23
Valor: 1.50
Saque efetuado com sucesso

========================
(1) Criar conta
(2) Depositar
(3) Sacar
(4) Consultar saldo
(5) Imprimir informacoes
(0) Sair
========================
Opcao escolhida: 4

Numero: 23
Saldo = 99.000000

========================
(1) Criar conta
(2) Depositar
(3) Sacar
(4) Consultar saldo
(5) Imprimir informacoes
(0) Sair
========================
Opcao escolhida: 5

Numero: 23
Numero: 23
Titular: Gabriel de Paula
Saldo: 99.00

========================
(1) Criar conta
(2) Depositar
(3) Sacar
(4) Consultar saldo
(5) Imprimir informacoes
(0) Sair
========================
Opcao escolhida: 0

gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-02$
```

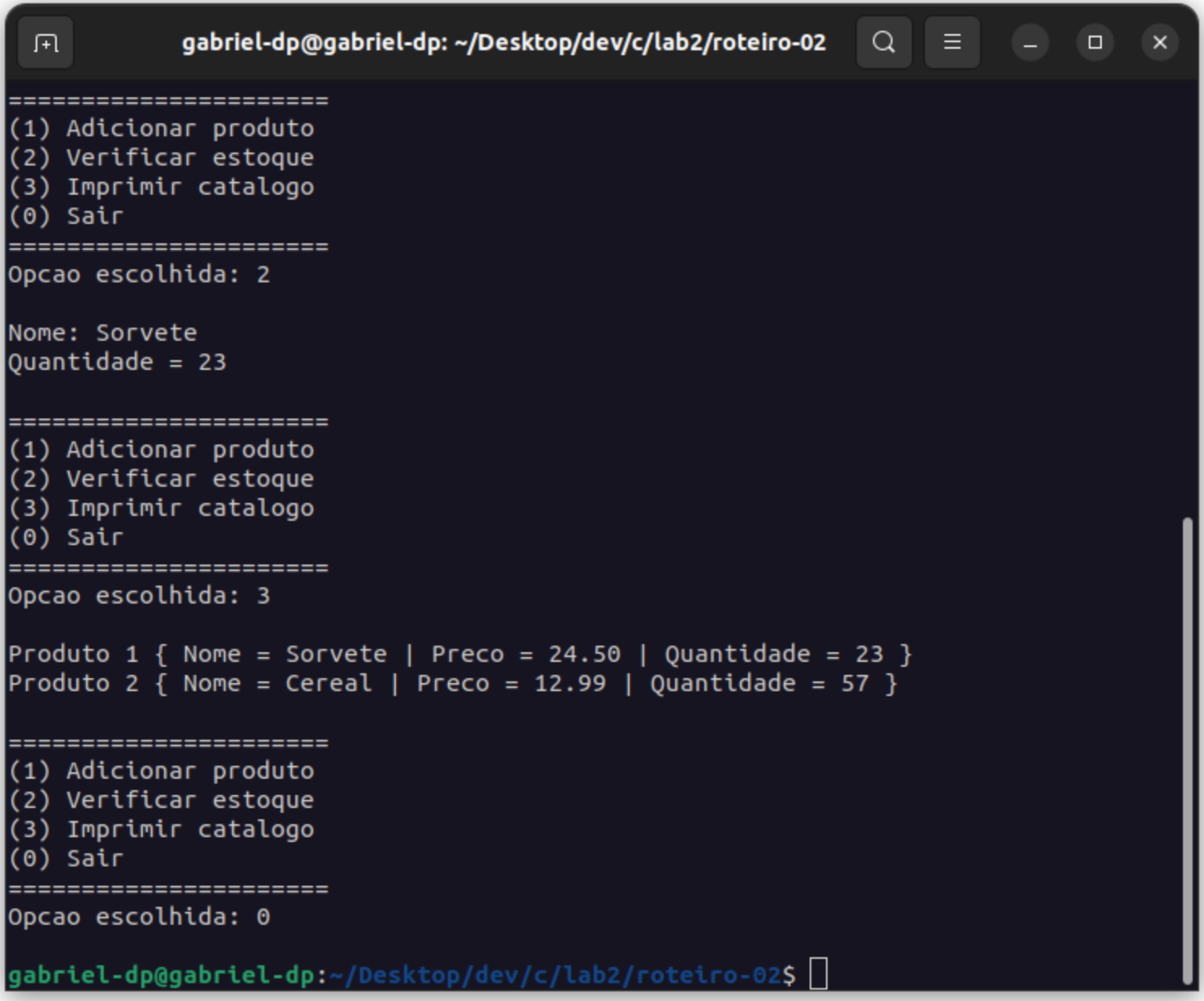
roteiro-02/ex01-02.c

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

#define NOME_CHARS 50

typedef struct {
    char nome[NOME_CHARS];
    double preco;
    int quantidade;
} Produto;

typedef struct {
    Produto produtos[100];
    int produtos_total;
} CatalogoProdutos;

enum OPCOES {
    SAIR = 0,
    ADICIONAR,
    VERIFICAR,
    IMPRIMIR
};

void criarCatalogo(CatalogoProdutos *c) {
    c->produtos_total = 0;
}

void adicionarProduto(CatalogoProdutos *c, char *nome, double preco, int quantidade) {
    Produto novo;
    strcpy(novo.nome, nome);
    novo.preco = preco;
    novo.quantidade = quantidade;

    c->produtos[c->produtos_total] = novo;
}

int verificarEstoque(CatalogoProdutos *c, char *nome) {
    for (int i = 0; i < c->produtos_total; i++) {
        if (strcmp(nome, c->produtos[i].nome) == 0) {
            return c->produtos[i].quantidade;
        }
    }

    return 0;
}

void imprimirCatalogo(CatalogoProdutos *c) {
    for (int i = 0; i < c->produtos_total; i++) {
        printf("Produto %d { Nome = %s | Preco = %.2lf | Quantidade = %d }\n", i + 1, c->produtos[i].nome, c->produtos[i].preco, c->produtos[i].quantidade);
    }
}

```

```c
int getOpcao() {
    int opcao;

    printf("\n======================\n");
    printf("(%d) Adicionar produto\n", ADICIONAR);
    printf("(%d) Verificar estoque\n", VERIFICAR);
    printf("(%d) Imprimir catalogo\n", IMPRIMIR);
    printf("(%d) Sair\n", SAIR);
    printf("======================\n");
    printf("Opcao escolhida: ");
    scanf("%d", &opcao);

    printf("\n");
    return opcao;
}

Produto *getProduto(CatalogoProdutos *catalogo, char *nome) {
    for (int i = 0; i < catalogo->produtos_total; i++) {
        if (strcmp(catalogo->produtos[i].nome, nome) == 0) {
            return &catalogo->produtos[i];
        }
    }
    return NULL;
}

int main() {
    CatalogoProdutos catalogo;
    criarCatalogo(&catalogo);

    int quantidade;
    double preco;
    char nome[NOME_CHARS + 1];

    int sair = 0;
    while (!sair) {
        switch (getOpcao()) {
            case SAIR:
                sair = 1;
                break;
            case ADICIONAR:
                setbuf(stdin, NULL);
                printf("Nome: ");
                fgets(nome, NOME_CHARS + 1, stdin);
                if (nome[strlen(nome) - 1] == '\n')
                    nome[strlen(nome) - 1] = '\0';
                setbuf(stdin, NULL);
                printf("Preco: ");
                scanf("%lf", &preco);
                printf("Quantidade: ");
                scanf("%d", &quantidade);

                if (getProduto(&catalogo, nome) == NULL) {
                    adicionarProduto(&catalogo, nome, preco, quantidade);
                    catalogo.produtos_total++;
                    printf("Produto adicionado com sucesso\n");
                } else {
                    printf("Produto ja existente\n");
                }
                break;
```

```c
            case VERIFICAR:
                setbuf(stdin, NULL);
                printf("Nome: ");
                fgets(nome, NOME_CHARS + 1, stdin);
                if (nome[strlen(nome) - 1] == '\n')
                    nome[strlen(nome) - 1] = '\0';
                setbuf(stdin, NULL);

                if (getProduto(&catalogo, nome) != NULL) {
                    printf("Quantidade = %d\n", verificarEstoque(&catalogo, nome));
                } else {
                    printf("Produto nao encontrado\n");
                }
                break;
            case IMPRIMIR:
                imprimirCatalogo(&catalogo);
                break;
            default:
                printf("Opcao invalida - Tente novamente\n");
                break;
        }
    }

    return 0;
}
```


gabriel-dp@gabriel-dp: ~/Desktop/dev/c/lab2/roteiro-02
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-02$ ./ex01-02
======================
(1) Adicionar produto
(2) Verificar estoque
(3) Imprimir catalogo
(0) Sair
======================
Opcao escolhida: 1
Nome: Sorvete
Preco: 24.50
Quantidade: 23
Produto adicionado com sucesso
======================
(1) Adicionar produto
(2) Verificar estoque
(3) Imprimir catalogo
(0) Sair
======================
Opcao escolhida: 1
Nome: Cereal
Preco: 12.99
Quantidade: 57
Produto adicionado com sucesso
======================
(1) Adicionar produto
(2) Verificar estoque


gabriel-dp@gabriel-dp: ~/Desktop/dev/c/lab2/roteiro-02
==========
(1) Adicionar produto
(2) Verificar estoque
(3) Imprimir catalogo
(0) Sair
==========
Opcao escolhida: 2
Nome: Sorvete
Quantidade = 23
==========
(1) Adicionar produto
(2) Verificar estoque
(3) Imprimir catalogo
(0) Sair
==========
Opcao escolhida: 3
Produto 1 { Nome = Sorvete | Preco = 24.50 | Quantidade = 23 }
Produto 2 { Nome = Cereal | Preco = 12.99 | Quantidade = 57 }
==========
(1) Adicionar produto
(2) Verificar estoque
(3) Imprimir catalogo
(0) Sair
==========
Opcao escolhida: 0
gabriel-dp@gabriel-dp:~/Desktop/dev/c/lab2/roteiro-02$

2) Análise de Complexidade

2.1

$8n^2 > 64n\log n$

$n > 8\log n$

$\frac{n}{8} > \log_2 n$

$[44, \infty[$ //

2.2

$100n^2 < 2^n$

$\log_2(100n^2) < n$

$\log_2(100) + 2\log_2(n) < n$

$6.6 + 2\log_2(n) < n$

15 //

2.3

A notaçao "O" define que o limite superior de $g(n)$ é $f(n)$.

2.4

A notação "Ω" define que o limite inferior de $g(n)$ é $f(n)$.

2.5

Não é possível que o mínimo seja $O(n^2)$, já que a notação define o limite superior (máximo).

2.6

$a(n) = n^2 - n + 500$

$b(n) = 47n + 47$

$n^2 - n + 500 < 47n + 47$

$n^2 - 48n + 453 < 0$

$\Delta = \sqrt{48^2 - 4 \cdot 453}$

$\Delta \approx 22,2$

$\frac{48 \pm 22,2}{2} = 24 \pm 11,1$

$12,9 < n < 35,1$ //

$[13, 35]$

2.7

$\sum_{j=1}^{N-1} \sum_{k=1}^{j} k$ $\quad O(n^3)$

$\frac{n^3 - 3n^2 + 2n}{3}$

2.8

$n-1$ passos para encontrar o maior. Se é Ω e O, logo é Θ.

$C=1$ em $\Omega(n)$ e $C=2$ em $O(n)$