

# Hotel Descanso Garantido - Sistema de Gerenciamento

## Introdução

Este projeto é um sistema de gerenciamento para um hotel fictício chamado "Hotel Descanso Garantido". O sistema permite o cadastro e consulta de clientes, funcionários, quartos, e estadias. As principais funcionalidades incluem registrar estadias e dar baixa nelas, mantendo os dados persistentes em arquivos.

## Estrutura do Projeto

O projeto está organizado da seguinte forma:

```
HotelDescansoGarantido/  
├── bin/  
├── include/  
│   ├── cliente.h  
│   ├── funcionario.h  
│   ├── quarto.h  
│   └── estadia.h  
├── src/  
│   ├── cliente.c  
│   ├── funcionario.c  
│   ├── quarto.c  
│   ├── estadia.c  
│   └── main.c  
├── data/  
│   ├── clientes.dat  
│   ├── funcionarios.dat  
│   ├── quartos.dat  
│   └── estadias.dat  
└── docs/
```

```
|— README.md
|— Makefile
```

- **bin/**: Diretório onde o executável final será colocado após a compilação.
- **include/**: Diretório contendo todos os arquivos de cabeçalho ( `.h` ).
- **src/**: Diretório contendo todos os arquivos fonte ( `.c` ).
- **data/**: Diretório onde são armazenados os arquivos de dados ( `.dat` ).
- **docs/**: Diretório para documentação adicional do projeto.
- **README.md**: Arquivo de texto explicativo sobre o projeto, incluindo instruções de uso.
- **Makefile**: Arquivo de construção para facilitar a compilação e limpeza do projeto.

## Arquivos de Cabeçalho ( `include/` )

### `cliente.h`

```
#ifndef CLIENTE_H
#define CLIENTE_H

typedef struct {
    int codigo;
    char nome[100];
    char cpf[12];
    char telefone[15];
} Cliente;

void cadastrarCliente();
void consultarClientes();

#endif // CLIENTE_H
```

### `funcionario.h`

```
#ifndef FUNCIONARIO_H
#define FUNCIONARIO_H
```

```
typedef struct {
    int codigo;
    char nome[100];
    char cpf[12];
    char telefone[15];
    char cargo[50];
} Funcionario;

void cadastrarFuncionario();
void consultarFuncionarios();

#endif // FUNCIONARIO_H
```

#### quarto.h

```
#ifndef QUARTO_H
#define QUARTO_H

typedef struct {
    int numero;
    char tipo[30];
    float valorDiaria;
    char status[10];
} Quarto;

void cadastrarQuarto();
void consultarQuartos();

#endif // QUARTO_H
```

#### estadia.h

```
#ifndef ESTADIA_H
#define ESTADIA_H

typedef struct {
    int codigo;
```

```

    char dataEntrada[11];
    char dataSaida[11];
    int quantidadeDiarias;
    int codigoCliente;
    int numeroQuarto;
} Estadia;

void cadastrarEstadia();
void darBaixaEstadia();
void consultarEstadias();

#endif // ESTADIA_H

```

## Arquivos Fonte ( **src/** )

### **cliente.c**

```

#include <stdio.h>
#include <string.h>
#include "cliente.h"

void cadastrarCliente() {
    Cliente cliente;
    FILE *file = fopen("data/clientes.dat", "ab");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de clientes!\\n");
        return;
    }

    printf("Código: ");
    scanf("%d", &cliente.codigo);
    printf("Nome: ");
    scanf("%s", cliente.nome);
    printf("CPF: ");
    scanf("%s", cliente.cpf);
    printf("Telefone: ");
    scanf("%s", cliente.telefone);
}

```

```

        fwrite(&cliente, sizeof(Cliente), 1, file);
        fclose(file);
        printf("Cliente cadastrado com sucesso!\n\n");
    }

void consultarClientes() {
    Cliente cliente;
    FILE *file = fopen("data/clientes.dat", "rb");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de clientes!\n\n");
        return;
    }

    printf("\n\nLista de Clientes:\n\n");
    while (fread(&cliente, sizeof(Cliente), 1, file)) {
        printf("Código: %d\n", cliente.codigo);
        printf("Nome: %s\n", cliente.nome);
        printf("CPF: %s\n", cliente.cpf);
        printf("Telefone: %s\n", cliente.telefone);
        printf("-----\n");
    }
    fclose(file);
}

```

### funcionario.c

```

#include <stdio.h>
#include <string.h>
#include "funcionario.h"

void cadastrarFuncionario() {
    Funcionario funcionario;
    FILE *file = fopen("data/funcionarios.dat", "ab");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de funcionários!\n\n");
        return;
    }
}

```

```

    printf("Código: ");
    scanf("%d", &funcionario.codigo);
    printf("Nome: ");
    scanf("%s", funcionario.nome);
    printf("CPF: ");
    scanf("%s", funcionario.cpf);
    printf("Telefone: ");
    scanf("%s", funcionario.telefone);
    printf("Cargo: ");
    scanf("%s", funcionario.cargo);

    fwrite(&funcionario, sizeof(Funcionario), 1, file);
    fclose(file);
    printf("Funcionário cadastrado com sucesso!\n\n");
}

void consultarFuncionarios() {
    Funcionario funcionario;
    FILE *file = fopen("data/funcionarios.dat", "rb");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de funcionários!\n\n");
        return;
    }

    printf("\n\nLista de Funcionários:\n\n");
    while (fread(&funcionario, sizeof(Funcionario), 1, file)) {
        printf("Código: %d\n", funcionario.codigo);
        printf("Nome: %s\n", funcionario.nome);
        printf("CPF: %s\n", funcionario.cpf);
        printf("Telefone: %s\n", funcionario.telefone);
        printf("Cargo: %s\n", funcionario.cargo);
        printf("-----\n");
    }
    fclose(file);
}

```

## quarto.c

```
#include <stdio.h>
#include <string.h>
#include "quarto.h"

void cadastrarQuarto() {
    Quarto quarto;
    FILE *file = fopen("data/quartos.dat", "ab");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de quartos!\n\n");
        return;
    }

    printf("Número: ");
    scanf("%d", &quarto.numero);
    printf("Tipo: ");
    scanf("%s", quarto.tipo);
    printf("Valor da Diária: ");
    scanf("%f", &quarto.valorDiaria);
    strcpy(quarto.status, "livre");

    fwrite(&quarto, sizeof(Quarto), 1, file);
    fclose(file);
    printf("Quarto cadastrado com sucesso!\n\n");
}

void consultarQuartos() {
    Quarto quarto;
    FILE *file = fopen("data/quartos.dat", "rb");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de quartos!\n\n");
        return;
    }

    printf("\n\nLista de Quartos:\n\n");
    while (fread(&quarto, sizeof(Quarto), 1, file)) {
        printf("Número: %d\n", quarto.numero);
    }
}
```

```

        printf("Tipo: %s\\n", quarto.tipo);
        printf("Valor da Diária: %.2f\\n", quarto.valorDiaria);
        printf("Status: %s\\n", quarto.status);
        printf("-----\\n");
    }
    fclose(file);
}

```

### estadia.c

```

#include <stdio.h>
#include <string.h>
#include "estadia.h"
#include "quarto.h"

void cadastrarEstadia() {
    Estadia estadia;
    FILE *file = fopen("data/estadias.dat", "ab");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de estadias!\\n");
        return;
    }

    printf("Código: ");
    scanf("%d", &estadia.codigo);
    printf("Data de entrada (DD/MM/YYYY): ");
    scanf("%s", estadia.dataEntrada);
    printf("Data de saída (DD/MM/YYYY): ");
    scanf("%s", estadia.dataSaida);
    printf("Quantidade de diárias: ");
    scanf("%d", &estadia.quantidadeDiarias);
    printf("Código do cliente: ");
    scanf("%d", &estadia.codigoCliente);
    printf("Número do quarto: ");
    scanf("%d", &estadia.numeroQuarto);

    // Atualiza o status do quarto para "ocupado"
}

```



```

    Quarto quarto;
    FILE *fileQuarto = fopen("data/quartos.dat", "rb+");
    if (fileQuarto != NULL) {
        while (fread(&quarto, sizeof(Quarto), 1, fileQuarto)) {
            if (quarto.numero == estadia.numeroQuarto) {
                strcpy(quarto.status, "ocupado");
                fseek(fileQuarto, -sizeof(Quarto), SEEK_CUR);
                fwrite(&quarto, sizeof(Quarto), 1, fileQuarto);
                break;
            }
        }
        fclose(fileQuarto);
    }

    fwrite(&estadia, sizeof(Estadia), 1, file);
    fclose(file);
    printf("Estadia cadastrada com sucesso!\n");
}

void dar

BaixaEstadia() {
    int codigo;
    printf("Código da estadia para dar baixa: ");
    scanf("%d", &codigo);

    Estadia estadia;
    FILE *file = fopen("data/estadias.dat", "rb+");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de estadias!\n");
        return;
    }

    int found = 0;
    while (fread(&estadia, sizeof(Estadia), 1, file)) {

```

```

        if (estadia.codigo == codigo) {
            found = 1;
            break;
        }
    }

    if (found) {
        fseek(file, -sizeof(Estadia), SEEK_CUR);
        fwrite(&estadia, sizeof(Estadia), 1, file);

        // Atualiza o status do quarto para "livre"
        Quarto quarto;
        FILE *fileQuarto = fopen("data/quartos.dat", "rb
+");
        if (fileQuarto != NULL) {
            while (fread(&quarto, sizeof(Quarto), 1, fileQu
arto)) {
                if (quarto.numero == estadia.numeroQuarto)
                {
                    strcpy(quarto.status, "livre");
                    fseek(fileQuarto, -sizeof(Quarto), SEEK
_CUR);
                    fwrite(&quarto, sizeof(Quarto), 1, file
Quarto);
                    break;
                }
            }
            fclose(fileQuarto);
        }

        printf("Baixa na estadia realizada com sucesso!
\\n");
    } else {
        printf("Estadia não encontrada!\\n");
    }
    fclose(file);
}

```

```

void consultarEstadias() {
    Estadia estadia;
    FILE *file = fopen("data/estadias.dat", "rb");
    if (file == NULL) {
        printf("Erro ao abrir o arquivo de estadias!\n\n");
        return;
    }

    printf("\n\nLista de Estadias:\n\n");
    while (fread(&estadia, sizeof(Estadia), 1, file)) {
        printf("Código: %d\n", estadia.codigo);
        printf("Data de Entrada: %s\n", estadia.dataEntrada);
        printf("Data de Saída: %s\n", estadia.dataSaida);
        printf("Quantidade de Diárias: %d\n", estadia.quantidadeDiarias);
        printf("Código do Cliente: %d\n", estadia.codigoCliente);
        printf("Número do Quarto: %d\n", estadia.numeroQuarto);
        printf("-----\n");
    }
    fclose(file);
}

```

### main.c

```

#include <stdio.h>
#include "cliente.h"
#include "funcionario.h"
#include "quarto.h"
#include "estadia.h"

int main() {
    int opcao;

    do {
        printf("\n\nMenu:\n\n");

```

```

printf("1. Cadastrar Cliente\\n");
printf("2. Consultar Clientes\\n");
printf("3. Cadastrar Funcionario\\n");
printf("4. Consultar Funcionarios\\n");
printf("5. Cadastrar Quarto\\n");
printf("6. Consultar Quartos\\n");
printf("7. Cadastrar Estadia\\n");
printf("8. Dar Baixa em Estadia\\n");
printf("9. Consultar Estadias\\n");
printf("0. Sair\\n");
printf("Opcao: ");
scanf("%d", &opcao);

switch (opcao) {
    case 1:
        cadastrarCliente();
        break;
    case 2:
        consultarClientes();
        break;
    case 3:
        cadastrarFuncionario();
        break;
    case 4:
        consultarFuncionarios();
        break;
    case 5:
        cadastrarQuarto();
        break;
    case 6:
        consultarQuartos();
        break;
    case 7:
        cadastrarEstadia();
        break;
    case 8:
        darBaixaEstadia();
        break;
}

```

```

        case 9:
            consultarEstadias();
            break;
        case 0:
            printf("Saindo...\n");
            break;
        default:
            printf("Opcao invalida!\n");
    }
} while (opcao != 0);

return 0;
}

```

## Makefile

```

CC = gcc
CFLAGS = -Iinclude -Wall
SRC = src/cliente.c src/funcionario.c src/quarto.c src/estadia.c src/main.c
OBJ = $(SRC:.c=.o)
EXEC = bin/HotelDescansoGarantido

all: $(EXEC)

$(EXEC): $(OBJ)
    $(CC) -o $@ $^

%.o: %.c
    $(CC) $(CFLAGS) -c -o $@ $<

clean:
    rm -f $(OBJ) $(EXEC)

```

## Instruções de Compilação e Execução

1. Navegue até o diretório do projeto:

```
cd HotelDescansoGarantido
```

2. Compile o projeto usando o Makefile:

```
make
```

3. Execute o programa:

```
./bin/HotelDescansoGarantido
```

## Considerações Finais

Este projeto oferece uma base para o gerenciamento de um hotel fictício. As funcionalidades básicas incluem cadastro e consulta de clientes, funcionários, quartos e estadias, bem como a atualização do status dos quartos. A estrutura modular do código facilita a manutenção e expansão do sistema, permitindo adicionar novas funcionalidades conforme necessário.