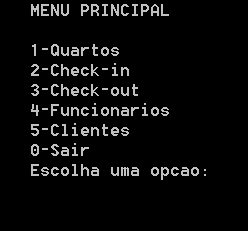
Sistema

de

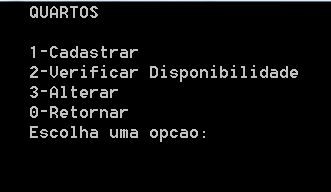
hotelaria

Grupo VemDeZap

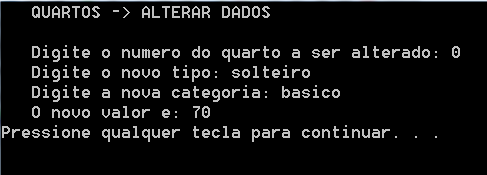
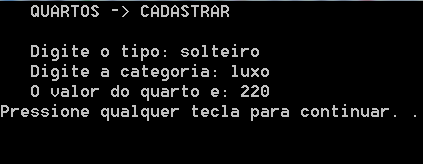
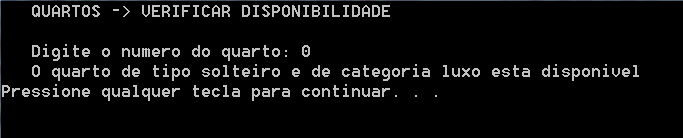
Apresentação dos menus do sistemas

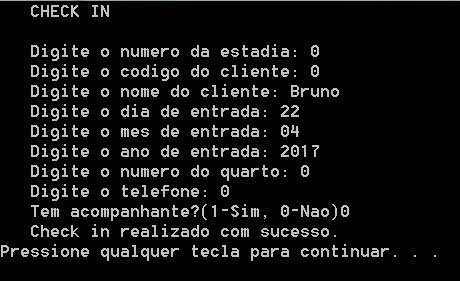


Menu principal do sistemas, mostra as opções de controle do usuário. Deve-se digitar o numero da opção deseja

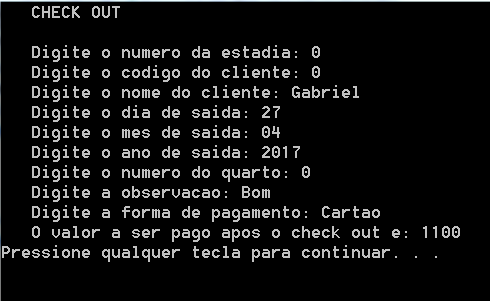
.

Através da opção 1 do menu principal, temos acesso ao menu de quartos, onde podemos ter informações sobre os mesmo, podemos cadastrar, alterar o tipo e a categoria do quarto e também verificar a disponibilidade dos mesmos. Abaixo temos exemplos das telas disponíveis.

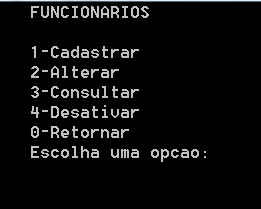




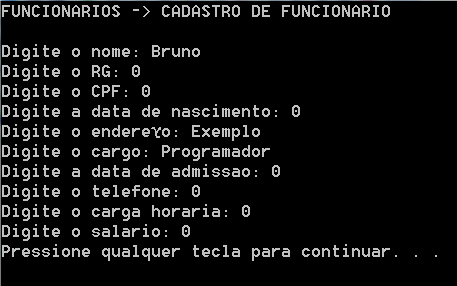
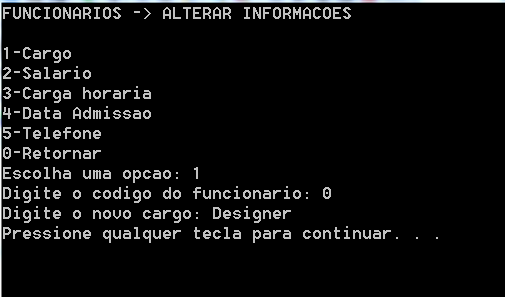
Tela de check in, acessavel através da opção 2 do menu principal. Deve-se ter o cliente cadastrado previamente, informando o numero do cliente, o nome e as demais informações para concluir o check in. Há a opção de se cadastrar um acompanhante, que informará o nome e telefone.

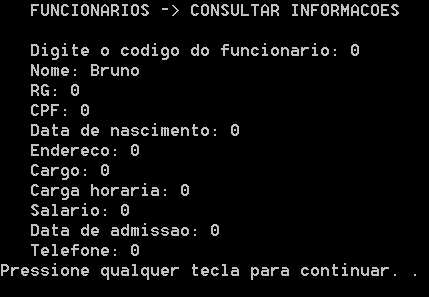


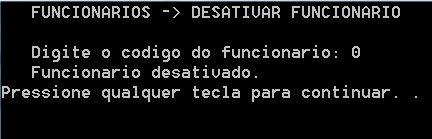
Tela do check out, sendo acessada através da opção 3 do menu principal. Informando o código do cliente, o nome e o quarto, juntamente com a data de saída, a tela retorna o valor a ser pago pelo cliente.

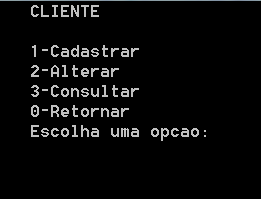


Menu de funcionários, opção 4 do menu principal, há a possibilidade de se cadastrar novos funcionários, alterar os dados dos já existentes e também consultar suas informações. Além de permitir desativar funcionários que já não fazem mais parte da equipe. Abaixo temos exemplos das telas de funcionários

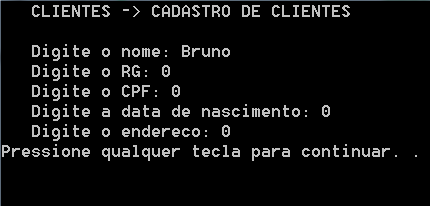


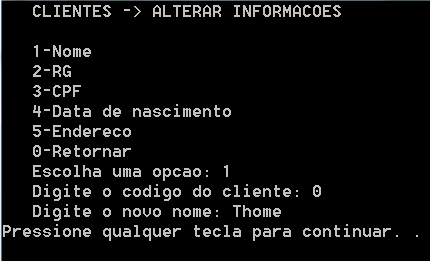
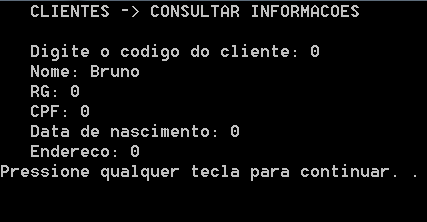






O ultimo menu disponível é o de clientes, que podem ser realizados cadastros, consultas e alterações das informações. Segue padrões parecidos com o de funcionários, com diferença nas informações necessárias. Abaixo temos exemplos das telas





Classe de Controle – Hotelaria

Classe desenvolvida para controlar todo o armazenamento em vetor das classes criadas.

#include "Classes.h"

#include <string>

#include <iostream>

#include <stdio.h>

#ifndef HOTELARIA\_H

#define HOTELARIA\_H

using namespace std;

class Hotelaria{

private:

Estadia vEstadia[100];

Funcionario vFuncionario[100];

Cliente vCliente[100];

Quarto vQuarto[100];

int nEstadia;

int nPessoa;

int num\_quarto;

int cod\_cli;

int cod\_func;

public:

Hotelaria();

~Hotelaria();

void cadastro\_quarto(Quarto quarto);

void cadastro\_func(Funcionario funcionario);

void cadastro\_cli(Cliente cliente);

bool check\_in(int nEstadia,string nome,int cod\_cli,int dia,int mes,int ano,int num\_quarto,string telefone);

Funcionario consulta\_func(int cod\_func);

Quarto consulta\_quarto(int num\_quarto);

Cliente consulta\_cli(int cod\_cli);

Estadia consulta\_estadia(int nEstadia);

float altera\_quarto(int num\_quarto,string tipo,string categoria);

void disponibilidade(int num\_quarto);

float check\_out(int nEstadia,int cod\_cli,string nome,int dia,int mes,int ano,int num\_quarto,string observacao,string forma\_pag);

void acompanhante(int nEstadia,string nome,string telefone);

string recupera\_senha(int cod\_func,string CPF,string cargo);

bool login(int cod\_func, string senha);

bool altera\_cargo(int cod\_func,string cargo);

bool altera\_data\_admissao(int cod\_func,string data\_admissao);

bool altera\_telefone(int cod\_func,string telefone);

bool altera\_carga\_horaria(int cod\_func,int carga\_horaria);

bool altera\_salario(int cod\_func,float salario);

bool desativa\_func(int cod\_func);

bool altera\_RG\_func(int cod\_func,string RG);

bool altera\_CPF\_func(int cod\_func,string CPF);

bool altera\_data\_nasc\_func(int cod\_func,string data\_nasc);

bool altera\_endereco\_func(int cod\_func,string endereco);

bool altera\_nome\_func(int cod\_func,string nome);

bool altera\_RG\_cli(int cod\_cli,string RG);

bool altera\_CPF\_cli(int cod\_cli,string CPF);

bool altera\_data\_nasc\_cli(int cod\_cli,string data\_nasc);

bool altera\_endereco\_cli(int cod\_cli,string endereco);

bool altera\_nome\_cli(int cod\_cli,string nome);

};

E agora a implementação dos métodos

#include "Hotelaria.h"

#include <string>

#include <iostream>

#include <stdio.h>

Hotelaria::Hotelaria()

{

cod\_cli=0;

cod\_func=0;

nEstadia=0;

nPessoa=0;

num\_quarto=0;

}

Hotelaria::~Hotelaria()

{

}

void Hotelaria::cadastro\_cli(Cliente cliente){

vCliente[cod\_cli] = cliente;

cod\_cli++;

nPessoa++;

}

void Hotelaria::cadastro\_func(Funcionario funcionario){

vFuncionario[cod\_func] = funcionario;

cod\_func++;

nPessoa++;

}

void Hotelaria::cadastro\_quarto(Quarto quarto){

vQuarto[num\_quarto] = quarto;

num\_quarto++;

}

bool Hotelaria::check\_in(int nEstadia,string nome,int cod\_cli,int dia,int mes,int ano,int num\_quarto,string telefone){

if(vCliente[cod\_cli].getNome()==nome){

if(vQuarto[num\_quarto].getDisponibilidade()==1){

vEstadia[nEstadia].setNome(nome);

vEstadia[nEstadia].setCodCli(cod\_cli);

vEstadia[nEstadia].setDataEntrada(dia,mes,ano);

vEstadia[nEstadia].setNumQuarto(num\_quarto);

vEstadia[nEstadia].setTelefone(telefone);

vQuarto[num\_quarto].setDisponibilidade();

nEstadia++;

return 1;

}else{

return 0; //ERRO Quarto não disponível

}

}else{

return 0; //ERRO Cliente não cadastrado

}

}

Cliente Hotelaria::consulta\_cli(int cod\_cli){

return vCliente[cod\_cli];

}

Estadia Hotelaria::consulta\_estadia(int nEstadia){

return vEstadia[nEstadia];

}

Funcionario Hotelaria::consulta\_func(int cod\_func){

return vFuncionario[cod\_func];

}

Quarto Hotelaria::consulta\_quarto(int num\_quarto){

return vQuarto[num\_quarto];

}

float Hotelaria::altera\_quarto(int num\_quarto,string tipo,string categoria){

float preco;

preco=vQuarto[num\_quarto].defineValor(tipo,categoria);

return preco;

}

void Hotelaria::disponibilidade(int num\_quarto){

vQuarto[num\_quarto].setDisponibilidade();

}

float Hotelaria::check\_out(int nEstadia,int cod\_cli, string nome,int dia,int mes,int ano, int num\_quarto,string observacao, string forma\_pag){

if(vCliente[cod\_cli].getNome()==nome){ //Verificar se existe um cliente com esse nome

if((vEstadia[nEstadia].getCodCli()==cod\_cli)&&(vEstadia[nEstadia].getNumQuarto()==num\_quarto)){ // Verificar se existe estadia com esse cliente e esse quarto

vEstadia[nEstadia].setDataSaida(dia,mes,ano);

vEstadia[nEstadia].setObservacao(observacao);

vEstadia[nEstadia].setFormaPag(forma\_pag);

if(vQuarto[num\_quarto].getDisponibilidade()==0){ //Verifica se o quarto foi usado, maneira de verificar se o quarto existe

float valor;

string tipo;

string categoria;

unsigned long dias;

Data entrada;

Data saida;

tipo=vQuarto[num\_quarto].getTipo();

categoria=vQuarto[num\_quarto].getCategoria();

valor=vQuarto[num\_quarto].defineValor(tipo,categoria);

entrada=vEstadia[nEstadia].getDataEntrada();

saida=vEstadia[nEstadia].getDataSaida();

dias=dist\_dias(entrada,saida);

valor\*=dias;

vCliente[cod\_cli].setConta(valor);

vQuarto[num\_quarto].setDisponibilidade();

return valor;

}else{

return 2.0; //ERRO Quarto estava em uso

}

}else{

return 1.0; //ERRO Estadia com esse nome não existe

}

}else{

return 0.0; //ERRO Cliente inexistente

}

}

void Hotelaria::acompanhante(int nEstadia,string nome,string telefone){

vEstadia[nEstadia].setAcompanhante(nome,telefone);

}

string Hotelaria::recupera\_senha(int cod\_func,string CPF,string cargo){

string senha;

senha=vFuncionario[cod\_func].recupera\_senha(CPF,cargo);

return senha;

}

bool Hotelaria::login(int cod\_func,string senha){

string senha\_d;

if(vFuncionario[cod\_func].getAtividade()==1){

senha\_d=vFuncionario[cod\_func].getSenha();

if(senha\_d==senha){

return 1; //Autorizado

}else{

return 0; //ERRO

}

}else{

return 0; //ERRO

}

}

bool Hotelaria::altera\_cargo(int cod\_func,string cargo){

vFuncionario[cod\_func].setCargo(cargo);

return 1;

}

bool Hotelaria::altera\_data\_admissao(int cod\_func,string data\_admissao){

vFuncionario[cod\_func].setDataAdmissao(data\_admissao);

return 1;

}

bool Hotelaria::altera\_telefone(int cod\_func,string telefone){

vFuncionario[cod\_func].setTelefone(telefone);

return 1;

}

bool Hotelaria::altera\_carga\_horaria(int cod\_func,int carga\_horaria){

vFuncionario[cod\_func].setCargaHoraria(carga\_horaria);

return 1;

}

bool Hotelaria::altera\_salario(int cod\_func,float salario){

vFuncionario[cod\_func].setSalario(salario);

return 1;

}

bool Hotelaria::desativa\_func(int cod\_func){

if(vFuncionario[cod\_func].getAtividade()==1){

vFuncionario[cod\_func].setAtividade();

return 1; //Sucesso

}else{

return 0; //ERRO - Funcionário já desativado

}

}

bool Hotelaria::altera\_RG\_func(int cod\_func,string RG){

vFuncionario[cod\_func].setRG(RG);

return 1;

}

bool Hotelaria::altera\_CPF\_func(int cod\_func,string CPF){

vFuncionario[cod\_func].setCPF(CPF);

return 1;

}

bool Hotelaria::altera\_data\_nasc\_func(int cod\_func,string data\_nasc){

vFuncionario[cod\_func].setDataNasc(data\_nasc);

return 1;

}

bool Hotelaria::altera\_endereco\_func(int cod\_func,string endereco){

vFuncionario[cod\_func].setEndereco(endereco);

return 1;

}

bool Hotelaria::altera\_nome\_func(int cod\_func,string nome){

vFuncionario[cod\_func].setNome(nome);

return 1;

}

bool Hotelaria::altera\_RG\_cli(int cod\_cli,string RG){

vCliente[cod\_cli].setRG(RG);

return 1;

}

bool Hotelaria::altera\_CPF\_cli(int cod\_cli,string CPF){

vCliente[cod\_cli].setCPF(CPF);

return 1;

}

bool Hotelaria::altera\_data\_nasc\_cli(int cod\_cli,string data\_nasc){

vCliente[cod\_cli].setDataNasc(data\_nasc);

return 1;

}

bool Hotelaria::altera\_endereco\_cli(int cod\_cli,string endereco){

vCliente[cod\_cli].setEndereco(endereco);

return 1;

}

bool Hotelaria::altera\_nome\_cli(int cod\_cli,string nome){

vCliente[cod\_cli].setNome(nome);

return 1;

}

Criação das Classes – Classes

Foi criado em um documento separado todas as classes que foram trabalhadas, esse mostra a definição dos métodos e objetos:

#include <iostream>

#include <string>

#include <stdio.h>

using namespace std;

#ifndef STRUCTS\_H

#define STRUCTS\_H

typedef struct acompanhante

{

string nome;

string telefone;

}dados\_Acomp;

typedef struct {

int dia;

int mes;

int ano;

} Data;

int bissexto (int ano);

unsigned long dist\_dias (Data inicio, Data fim);

#endif

#ifndef PESSOA\_H

#define PESSOA\_H

class Pessoa

{

public:

Pessoa();

~Pessoa();

void setNome(string nome);

string getNome();

void setRG(string RG);

string getRG();

void setCPF(string CPF);

string getCPF();

void setDataNasc(string data\_nasc);

string getDataNasc();

void setEndereco(string endereco);

string getEndereco();

private:

string nome;

string RG;

string CPF;

string data\_nasc;

string endereco;

};

#endif

#ifndef QUARTO\_H

#define QUARTO\_H

class Quarto

{

public:

Quarto();

~Quarto();

void setValor(float valor);

float getValor();

void setTipo(string tipo);

string getTipo();

void setCategoria(string categoria);

string getCategoria();

float defineValor(string tipo,string categoria);

void setDisponibilidade();

bool getDisponibilidade();

//float dadosQuarto();

private:

float valor;

string tipo;

string categoria;

bool disponivel;

};

#endif

#ifndef ESTADIA\_H

#define ESTADIA\_H

class Estadia

{

public:

Estadia();

~Estadia();

void setAcompanhante(string nome,string telefone);

dados\_Acomp getAcompanhante();

void setDataEntrada(int dia,int mes,int ano);

Data getDataEntrada();

void setDataSaida(int dia,int mes,int ano);

Data getDataSaida();

void setValor(float valor);

float getValor();

void setFormaPag(string forma\_pag);

string getFormaPag();

void setQualidade(string qualidade);

string getQualidade();

void setObservacao(string observacao);

string getObservacao();

void setTelefone(string telefone);

string getTelefone();

void setQuantQuartos(int quant\_quartos);

int getQuantQuartos();

void setGanhos(float ganhos);

float getGanhos();

void setQuantPessoas(int quant\_pessoas);

int getQuantPessoas();

void setNome(string nome);

string getNome();

void setCodCli(int cod\_cli);

int getCodCli();

void setNumQuarto(int num\_quarto);

int getNumQuarto();

protected:

dados\_Acomp pessoa;

int cod\_cli;

int num\_quarto;

string nome;

Data entrada;

Data saida;

float valor;

string forma\_pag;

string qualidade;

string observacao;

string telefone;

int quant\_quartos;

float ganhos;

int quant\_pessoas;

};

#endif

#ifndef FUNCIONARIO\_H

#define FUNCIONARIO\_H

class Funcionario : public Pessoa

{

public:

Funcionario();

~Funcionario();

string recupera\_senha(string CPF,string cargo);

bool login(string senha);

void setAtividade();

bool getAtividade();

void setCargo(string cargo);

string getCargo();

void setDataAdmissao(string data\_admissao);

string getDataAdmissao();

void setTelefone(string telefone);

string getTelefone();

void setCargaHoraria(int carga\_horaria);

int getCargaHoraria();

void setSalario(float salario);

float getSalario();

void setSenha(string senha);

string getSenha();

protected:

bool ativo;

string cargo;

string data\_admissao;

string telefone;

int carga\_horaria;

float salario;

string senha;

};

#endif

#ifndef CLIENTE\_H

#define CLIENTE\_H

class Cliente : public Pessoa

{

public:

Cliente();

~Cliente();

void setConta(float valor);

float getConta();

void diminuiConta(float valor);

void setCodCli(int cod\_cli);

int getCodCli();

protected:

float Conta;

int cod\_cli;

};

#endif

E agora a definição dos métodos:

#include "Classes.h"

#include <iostream>

#include <string>

#include <stdio.h>

using namespace std;

int bissexto (int ano) {

return (ano % 4 == 0) && ((ano % 100 != 0) || (ano % 400 == 0));

}

int dias\_mes[2][13] = {{0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},

{0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}};

//Função para achar a distancia entre a data de fim e data de início

unsigned long dist\_dias (Data inicio, Data fim) {

unsigned long idias, fdias; /\* guarda qtos dias tem da data \*/

/\* ate o comeco do ano \*/

unsigned long def\_anos = 0; /\* guarda diferenca entre anos das \*/

/\* datas inicio e fim medida em dias \*/

register int i;

int dbissexto;

idias = inicio.dia;

dbissexto = bissexto (inicio.ano);

for (i = inicio.mes - 1; i > 0; --i)

idias += dias\_mes[dbissexto][i];

fdias = fim.dia;

dbissexto = bissexto (fim.ano);

for (i = fim.mes - 1; i > 0; --i)

fdias += dias\_mes[dbissexto][i];

while (inicio.ano < fim.ano)

def\_anos += 365 + bissexto(inicio.ano++);

return def\_anos - idias + fdias;

}

Pessoa::Pessoa()

{

}

Pessoa::~Pessoa()

{

}

void Pessoa::setCPF(string CPF)

{

this->CPF=CPF;

}

string Pessoa::getCPF()

{

return this->CPF;

}

void Pessoa::setRG(string RG)

{

this->RG=RG;

}

string Pessoa::getRG()

{

return this->RG;

}

void Pessoa::setNome(string nome)

{

this->nome=nome;

}

string Pessoa::getNome()

{

return this->nome;

}

void Pessoa::setDataNasc(string data\_nasc)

{

this->data\_nasc=data\_nasc;

}

string Pessoa::getDataNasc()

{

return this->data\_nasc;

}

void Pessoa::setEndereco(string endereco)

{

this->endereco=endereco;

}

string Pessoa::getEndereco()

{

return this->endereco;

}

Funcionario::Funcionario()

{

ativo=1;

}

Funcionario::~Funcionario()

{

}

string Funcionario::recupera\_senha(string CPF,string cargo)

{

string CPFPessoa;

CPFPessoa=getCPF();

if(CPFPessoa==CPF){

if(this->cargo==cargo){

return this->senha;

}else{

return "Cargo inválido";

}

}else{

return "CPF inválido";

}

}

void Funcionario::setCargo(string cargo)

{

this->cargo=cargo;

}

string Funcionario::getCargo()

{

return this->cargo;

}

void Funcionario::setDataAdmissao(string data\_admissao)

{

this->data\_admissao=data\_admissao;

}

string Funcionario::getDataAdmissao()

{

return this->data\_admissao;

}

void Funcionario::setTelefone(string telefone)

{

this->telefone=telefone;

}

string Funcionario::getTelefone()

{

return this->telefone;

}

void Funcionario::setCargaHoraria(int carga\_horaria)

{

this->carga\_horaria=carga\_horaria;

}

int Funcionario::getCargaHoraria()

{

return this->carga\_horaria;

}

void Funcionario::setSalario(float salario)

{

this->salario=salario;

}

float Funcionario::getSalario()

{

return this->salario;

}

void Funcionario::setSenha(string senha)

{

this->senha=senha;

}

string Funcionario::getSenha()

{

return this->senha;

}

void Funcionario::setAtividade()

{

if(ativo==1){

ativo=0;

}else{

ativo=1;

}

}

bool Funcionario::getAtividade()

{

if(ativo==1){

return 1;

}else{

return 0;

}

}

Cliente::Cliente()

{

this->Conta=0.0;

}

Cliente::~Cliente()

{

}

void Cliente::setConta(float valor)

{

this->Conta+=valor;

}

float Cliente::getConta()

{

return this->Conta;

}

void Cliente::diminuiConta(float valor)

{

this->Conta-=valor;

}

void Cliente::setCodCli(int cod\_cli)

{

this->cod\_cli=cod\_cli;

}

int Cliente::getCodCli()

{

return this->cod\_cli;

}

Quarto::~Quarto()

{

}

Quarto::Quarto()

{

disponivel=1;

}

void Quarto::setValor(float valor)

{

this->valor=valor;

}

float Quarto::getValor()

{

return this->valor;

}

void Quarto::setTipo(string tipo)

{

this->tipo=tipo;

}

string Quarto::getTipo()

{

return this->tipo;

}

void Quarto::setCategoria(string categoria)

{

this->categoria=categoria;

}

string Quarto::getCategoria()

{

return this->categoria;

}

float Quarto::defineValor(string tipo, string categoria)

{

float valor;

this->tipo=tipo;

this->categoria=categoria;

if(tipo=="solteiro"){

valor+=70.00;

}else{

if(tipo=="duplo\_solteiro"){

valor+=120.00;

}else{

if(tipo=="casal"){

valor+=150.00;

}

}

}

if(categoria=="basico"){

valor+=0.00;

}else{

if(categoria=="luxo"){

valor+=150.00;

}else{

if(categoria=="sweet\_dreams"){

valor+=300.00;

}

}

}

return valor;

}

void Quarto::setDisponibilidade()

{

if(disponivel==0)

disponivel=1;

else

disponivel=0;

}

bool Quarto::getDisponibilidade(){

return this->disponivel;

}

Estadia::Estadia()

{

}

Estadia::~Estadia()

{

}

void Estadia::setAcompanhante(string nome,string telefone)

{

int i;

for(i=0;i<quant\_pessoas;i++){

pessoa.nome=nome;

pessoa.telefone=telefone;

}

pessoa.nome=nome;

pessoa.telefone=telefone;

}

dados\_Acomp Estadia::getAcompanhante()

{

return this->pessoa;

}

void Estadia::setDataEntrada(int dia,int mes,int ano)

{

this->entrada.dia=dia;

this->entrada.mes=mes;

this->entrada.ano=ano;

}

Data Estadia::getDataEntrada()

{

return this->entrada;

}

void Estadia::setDataSaida(int dia,int mes,int ano)

{

this->saida.dia=dia;

this->saida.mes=mes;

this->saida.ano=ano;

}

Data Estadia::getDataSaida()

{

return this->saida;

}

void Estadia::setValor(float valor)

{

this->valor=valor;

}

float Estadia::getValor()

{

return this->valor;

}

void Estadia::setFormaPag(string forma\_pag)

{

this->forma\_pag=forma\_pag;

}

string Estadia::getFormaPag()

{

return this->forma\_pag;

}

void Estadia::setQualidade(string qualidade)

{

this->qualidade=qualidade;

}

string Estadia::getQualidade()

{

return this->qualidade;

}

void Estadia::setObservacao(string observacao)

{

this->observacao=observacao;

}

string Estadia::getObservacao()

{

return this->observacao;

}

void Estadia::setTelefone(string telefone)

{

this->telefone=telefone;

}

string Estadia::getTelefone()

{

return this->telefone;

}

void Estadia::setQuantQuartos(int quant\_quartos)

{

this->quant\_quartos=quant\_quartos;

}

int Estadia::getQuantQuartos()

{

return this->quant\_quartos;

}

void Estadia::setGanhos(float ganhos)

{

this->ganhos=ganhos;

}

float Estadia::getGanhos()

{

return this->ganhos;

}

void Estadia::setQuantPessoas(int quant\_pessoas)

{

this->quant\_pessoas=quant\_pessoas;

}

int Estadia::getQuantPessoas()

{

return this->quant\_pessoas;

}

void Estadia::setNome(string nome)

{

this->nome=nome;

}

string Estadia::getNome()

{

return this->nome;

}

void Estadia::setCodCli(int cod\_cli)

{

this->cod\_cli=cod\_cli;

}

int Estadia::getCodCli()

{

return this->cod\_cli;

}

void Estadia::setNumQuarto(int num\_quarto)

{

this->num\_quarto=num\_quarto;

}

int Estadia::getNumQuarto()

{

return this->num\_quarto;

}

Interface Textual

E aqui está o arquivo em que foi implementada a interface textual de contato com o usuário:

#include <string>

#include <iostream>

#include <stdio.h>

#include "Classes.h"

#include "Hotelaria.h"

#include "ClassesDAO.h"

#include <windows.h>

using namespace std;

int main(int argc,char\*\* argv)

{

Hotelaria hotel;

Funcionario funcionario;

Cliente cliente;

int opcao=0;

int opcao2=0;

int opcao3 = 0;

//cout << "Sistema de hotelaria VemDeZap" << endl;

cout << " \_\_ \_\_ " << endl;

cout << " /\\ \\/\\ \\ " << endl;

cout << " \\ \\ \\ \\ \\ \_\_ \_\_\_ \_\_\_ " << endl;

cout << " \\ \\ \\ \\ \\ /'\_\_`\\ /' \_\_` \_\_`\\ " << endl;

cout << " \\ \\ \\\_/ \\/\\ \_\_/ /\\ \\/\\ \\/\\ \\ " << endl;

cout << " \\ `\\\_\_\_/\\ \\\_\_\_\_\\\\ \\\_\\ \\\_\\ \\\_\\ " << endl;

cout << " `\\/\_\_/ \\/\_\_\_\_/ \\/\_/\\/\_/\\/\_/ " << endl;

cout << " " << endl;

cout << " " << endl;

cout << " \_\_\_\_ " << endl;

cout << " /\\ \_`\\ " << endl;

cout << " \\ \\ \\/\\ \\ \_\_ " << endl;

cout << " \\ \\ \\ \\ \\ /'\_\_`\\ " << endl;

cout << " \\ \\ \\\_\\ \\/\\ \_\_/ " << endl;

cout << " \\ \\\_\_\_\_/\\ \\\_\_\_\_\\ " << endl;

cout << " \\/\_\_\_/ \\/\_\_\_\_/ " << endl;

cout << " " << endl;

cout << " " << endl;

cout << " \_\_\_\_\_\_\_\_ " << endl;

cout << " /\\\_\_\_\_\_ \\ " << endl;

cout << " \\/\_\_\_\_//'/' \_\_ \_\_\_\_\_ " << endl;

cout << " //'/' /'\_\_`\\ /\\ '\_\_`\\ " << endl;

cout << " //'/'\_\_\_ /\\ \\L\\.\\\_\\ \\ \\L\\ \\ " << endl;

cout << " /\\\_\_\_\_\_\_\_\\\\ \\\_\_/.\\\_\\\\ \\ ,\_\_/ " << endl;

cout << " \\/\_\_\_\_\_\_\_/ \\/\_\_/\\/\_/ \\ \\ \\/ " << endl;

cout << " \\ \\\_\\ " << endl;

cout << " \\/\_/ " << endl;

cout << " " << endl;

system("pause");

do{

system("cls");

cout << " MENU PRINCIPAL" << endl << endl;

cout << " 1-Quartos" << endl;

cout << " 2-Check-in" << endl;

cout << " 3-Check-out" << endl;

cout << " 4-Funcionarios" << endl;

cout << " 5-Clientes" << endl;

cout << " 0-Sair" << endl;

cout << " Escolha uma opcao: ";

cin >> opcao3;

cout << endl;

string nome\_pessoa = "";

string rg = "";

string cpf = "";

string data\_nasc = "";

string endereco = "";

string cargo = "";

string data\_adm = "";

string telefone = "";

int carga\_h = 0;

float salario = 0;

string senha = "";

int cod\_func = 0;

string tipo = "";

string categoria = "";

string nome\_cliente = "";

string observacao = "";

string forma\_pagamento = "";

string nome\_acompanhante = "";

string tel\_acompanhante = "";

bool acompanhante;

int cod\_cliente = 0;

int dia\_saida = 0;

int mes\_saida = 0;

int ano\_saida = 0;

int dia\_entrada = 0;

int mes\_entrada = 0;

int ano\_entrada = 0;

int nEstadia = 0;

Quarto quarto;

float valor;

int num\_quarto = 0;

switch (opcao3){

case 1: // Quartos

do{

system("cls");

cout << " QUARTOS" << endl << endl;

cout << " 1-Cadastrar" << endl;

cout << " 2-Verificar Disponibilidade" << endl;

cout << " 3-Alterar" << endl;

cout << " 0-Retornar" << endl;

cout << " Escolha uma opcao: ";

cin >> opcao2;

switch(opcao2){

case 1:

system("cls");

cout << " QUARTOS -> CADASTRAR" << endl << endl;

cout << " Digite o tipo: ";

cin >> tipo;

quarto.setTipo(tipo);

cout << " Digite a categoria: ";

cin >> categoria;

quarto.setCategoria(categoria);

valor=quarto.defineValor(tipo,categoria);

quarto.setValor(valor);

cout << " O valor do quarto e: " << valor << endl;

hotel.cadastro\_quarto(quarto);

system("pause");

break;

case 2:

system("cls");

cout << " QUARTOS -> VERIFICAR DISPONIBILIDADE" << endl << endl;

cout << " Digite o numero do quarto: ";

cin >> num\_quarto;

quarto = hotel.consulta\_quarto(num\_quarto);

if(quarto.getDisponibilidade() == 1){

cout << " O quarto de tipo " << quarto.getTipo() << " e de categoria " << quarto.getCategoria() << " esta disponivel" << endl;

system("pause");

}else{

cout << " O quarto de tipo " << quarto.getTipo() << " e de categoria " << quarto.getCategoria() << " esta ocupado" << endl;

system("pause");

}

break;

case 3:

system("cls");

cout << " QUARTOS -> ALTERAR DADOS" << endl << endl;

cout << " Digite o numero do quarto a ser alterado: ";

cin >> num\_quarto;

cout << " Digite o novo tipo: ";

cin >> tipo;

cout << " Digite a nova categoria: ";

cin >> categoria;

valor = hotel.altera\_quarto(num\_quarto, tipo, categoria);

cout << " O novo valor e: " << valor << endl;

system(" pause");

break;

}

}while(opcao2!=0);

break;

case 2:

system("cls");

cout << " CHECK IN" << endl << endl;

cout << " Digite o numero da estadia: ";

cin >> nEstadia;

cout << " Digite o codigo do cliente: ";

cin >> cod\_cliente;

cout << " Digite o nome do cliente: ";

cin >> nome\_cliente;

cout << " Digite o dia de entrada: ";

cin >> dia\_entrada;

cout << " Digite o mes de entrada: ";

cin >> mes\_entrada;

cout << " Digite o ano de entrada: ";

cin >> ano\_entrada;

cout << " Digite o numero do quarto: " ;

cin >> num\_quarto;

cout << " Digite o telefone: " ;

cin >> telefone;

hotel.check\_in(nEstadia, nome\_cliente, cod\_cliente, dia\_entrada, mes\_entrada, ano\_entrada, num\_quarto, telefone);

cout << " Tem acompanhante?(1-Sim, 0-Nao)";

cin >> acompanhante;

if(acompanhante==1){

cout << " Digite o nome do acompanhante: ";

cin >> nome\_acompanhante;

cout << " Digite o telefone do acompanhante: ";

cin >> tel\_acompanhante;

hotel.acompanhante(nEstadia, nome\_acompanhante, tel\_acompanhante);

cout << " Check in realizado com sucesso." << endl;

system("pause");

break;

}

else{

cout << " Check in realizado com sucesso." << endl;

system("pause");

break;

}

case 3: //Check out

system("cls");

cout << " CHECK OUT" << endl << endl;

cout << " Digite o numero da estadia: ";

cin >> nEstadia;

cout << " Digite o codigo do cliente: ";

cin >> cod\_cliente;

cout << " Digite o nome do cliente: ";

cin >> nome\_cliente;

cout << " Digite o dia de saida: ";

cin >> dia\_saida;

cout << " Digite o mes de saida: ";

cin >> mes\_saida;

cout << " Digite o ano de saida: ";

cin >> ano\_saida;

cout << " Digite o numero do quarto: " ;

cin >> num\_quarto;

cout << " Digite a observacao: " ;

cin >> observacao;

cout << " Digite a forma de pagamento: " ;

cin >> forma\_pagamento;

valor = hotel.check\_out(nEstadia, cod\_cliente, nome\_cliente, dia\_saida, mes\_saida, ano\_saida, num\_quarto, observacao, forma\_pagamento);

if(valor == 0.0){

cout << " O cliente informado nao existe." << endl;

system("pause");

}

else if(valor == 1.0){

cout << " O check in nao foi realizado." << endl;

system("pause");

}

else if(valor == 2.0){

cout << " O quarto informado nao confere com o cliente." << endl;

system("pause");

}

else{

cout << " O valor a ser pago apos o check out e: " << valor << endl;

system("pause");

break;

}

case 4:

do{

system("cls");

cout << "FUNCIONARIOS" << endl << endl;

cout << "1-Cadastrar" << endl;

cout << "2-Alterar" << endl;

cout << "3-Consultar" << endl;

cout << "4-Desativar" << endl;

cout << "0-Retornar" << endl;

cout << "Escolha uma opcao: ";

cin >> opcao2;

switch(opcao2){

case 1:

system("cls");

cout << "FUNCIONARIOS -> CADASTRO DE FUNCIONARIO" << endl << endl;

cout << "Digite o nome: " ;

cin >> nome\_pessoa;

funcionario.setNome(nome\_pessoa);

cout << "Digite o RG: " ;

cin >> rg;

funcionario.setRG(rg);

cout << "Digite o CPF: " ;

cin >> cpf;

funcionario.setCPF(cpf);

cout << "Digite a data de nascimento: " ;

cin >> data\_nasc;

funcionario.setDataNasc(data\_nasc);

cout << "Digite o endereço: " ;

cin >> endereco;

funcionario.setEndereco(endereco);

cout << "Digite o cargo: " ;

cin >> cargo;

funcionario.setCargo(cargo);

cout << "Digite a data de admissao: " ;

cin >> data\_adm;

funcionario.setDataAdmissao(data\_adm);

cout << "Digite o telefone: " ;

cin >> telefone;

funcionario.setTelefone(telefone);

cout << "Digite o carga horaria: " ;

cin >> carga\_h;

funcionario.setCargaHoraria(carga\_h);

cout << "Digite o salario: " ;

cin >> salario;

funcionario.setSalario(salario);

hotel.cadastro\_func(funcionario);

system("pause");

break;

case 2:

do{

system("cls");

cout << "FUNCIONARIOS -> ALTERAR INFORMACOES" << endl << endl;

cout << "1-Cargo" << endl;

cout << "2-Salario" << endl;

cout << "3-Carga horaria" << endl;

cout << "4-Data Admissao" << endl;

cout << "5-Telefone" << endl;

cout << "6-Nome" << endl;

cout << "7-RG" << endl;

cout << "8-CPF" << endl;

cout << "9-Endereço" << endl;

cout << "0-Retornar" << endl;

cout << "Escolha uma opcao: ";

cin >> opcao;

switch (opcao){

case 1:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite o novo cargo: ";

cin >> cargo;

hotel.altera\_cargo(cod\_func,cargo);

system("pause");

break;

case 2:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite o novo salario: " ;

cin >> salario;

hotel.altera\_salario(cod\_func,salario);

system("pause");

break;

case 3:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite a nova carga horaria: " ;

cin >> carga\_h;

hotel.altera\_carga\_horaria(cod\_func,carga\_h);

system("pause");

break;

case 4:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite a nova data de admissao: " ;

cin >> data\_adm;

hotel.altera\_data\_admissao(cod\_func,data\_adm);

system("pause");

break;

case 5:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite o novo telefone: " ;

cin >> telefone;

hotel.altera\_telefone(cod\_func,telefone);

system("pause");

break;

case 6:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite o novo nome: " ;

cin >> nome\_pessoa;

hotel.altera\_nome\_func(cod\_func,nome\_pessoa);

system("pause");

break;

case 7:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite o novo RG: " ;

cin >> rg;

hotel.altera\_RG\_func(cod\_func,rg);

system("pause");

break;

case 8:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite o novo CPF: " ;

cin >> cpf;

hotel.altera\_CPF\_func(cod\_func,cpf);

system("pause");

break;

case 9:

cout << "Digite o codigo do funcionario: ";

cin >> cod\_func;

cout << "Digite o novo endereco: " ;

cin >> endereco;

hotel.altera\_endereco\_func(cod\_func,endereco);

system("pause");

break;

}

}while(opcao != 0);

break;

case 3:

system("cls");

cout << " FUNCIONARIOS -> CONSULTAR INFORMACOES" << endl << endl;

cout << " Digite o codigo do funcionario: ";

cin >> cod\_func;

funcionario = hotel.consulta\_func(cod\_func);

cout << " Nome: " << funcionario.getNome() << endl;

cout << " RG: " << funcionario.getRG() << endl;

cout << " CPF: " << funcionario.getCPF() << endl;

cout << " Data de nascimento: " << funcionario.getDataNasc() << endl;

cout << " Endereco: " << funcionario.getEndereco() << endl;

cout << " Cargo: " << funcionario.getCargo() << endl;

cout << " Carga horaria: " << funcionario.getCargaHoraria() << endl;

cout << " Salario: " << funcionario.getSalario() << endl;

cout << " Data de admissao: " << funcionario.getDataAdmissao() << endl;

cout << " Telefone: " << funcionario.getTelefone() << endl;

if(funcionario.getAtividade()==1){

cout << " Status: Ativo" << endl;

}else{

cout << " Status: Desativado" << endl;

}

system("pause");

break;

case 4:

system("cls");

cout << " FUNCIONARIOS -> DESATIVAR FUNCIONARIO" << endl << endl;

cout << " Digite o codigo do funcionario: ";

cin >> cod\_func;

hotel.desativa\_func(cod\_func);

cout << " Funcionario desativado." << endl;

system("pause");

break;

}

}while(opcao2!=0);

break;

case 5:

do{

system("cls");

cout << " CLIENTE" << endl << endl;

cout << " 1-Cadastrar" << endl;

cout << " 2-Alterar" << endl;

cout << " 3-Consultar" << endl;

cout << " 0-Retornar" << endl;

cout << " Escolha uma opcao: ";

cin >> opcao2;

switch(opcao2){

case 1:

system("cls");

cout << " CLIENTES -> CADASTRO DE CLIENTES" << endl << endl;

cout << " Digite o nome: " ;

cin >> nome\_pessoa;

cliente.setNome(nome\_pessoa);

cout << " Digite o RG: " ;

cin >> rg;

cliente.setRG(rg);

cout << " Digite o CPF: " ;

cin >> cpf;

cliente.setCPF(cpf);

cout << " Digite a data de nascimento: " ;

cin >> data\_nasc;

cliente.setDataNasc(data\_nasc);

cout << " Digite o endereco: " ;

cin >> endereco;

cliente.setEndereco(endereco);

cliente.setConta(0);

hotel.cadastro\_cli(cliente);

system("pause");

break;

case 2:

do{

system("cls");

cout << " CLIENTES -> ALTERAR INFORMACOES" << endl << endl;

cout << " 1-Nome" << endl;

cout << " 2-RG" << endl;

cout << " 3-CPF" << endl;

cout << " 4-Data de nascimento" << endl;

cout << " 5-Endereco" << endl;

cout << " 0-Retornar" << endl;

cout << " Escolha uma opcao: ";

cin >> opcao;

switch (opcao){

case 1:

cout << " Digite o codigo do cliente: ";

cin >> cod\_cliente;

cout << " Digite o novo nome: ";

cin >> nome\_pessoa;

hotel.altera\_nome\_cli(cod\_cliente,nome\_pessoa);

system("pause");

break;

case 2:

cout << " Digite o codigo do cliente: ";

cin >> cod\_cliente;

cout << " Digite o novo RG: " ;

cin >> rg;

hotel.altera\_RG\_cli(cod\_cliente,rg);

system("pause");

break;

case 3:

cout << " Digite o codigo do cliente: ";

cin >> cod\_cliente;

cout << " Digite o novo CPF: " ;

cin >> cpf;

hotel.altera\_CPF\_cli(cod\_cliente,cpf);

system("pause");

break;

case 4:

cout << " Digite o codigo do cliente: ";

cin >> cod\_cliente;

cout << " Digite a nova data de nascimento: " ;

cin >> data\_nasc;

hotel.altera\_data\_nasc\_cli(cod\_cliente,data\_nasc);

system("pause");

break;

case 5:

cout << " Digite o codigo do cliente: ";

cin >> cod\_cliente;

cout << " Digite o novo endereco: " ;

cin >> endereco;

hotel.altera\_endereco\_cli(cod\_cliente,endereco);

system("pause");

break;

}

}while(opcao != 0);

break;

case 3:

system("cls");

cout << " CLIENTES -> CONSULTAR INFORMACOES" << endl << endl;

cout << " Digite o codigo do cliente: ";

cin >> cod\_cliente;

cliente = hotel.consulta\_cli(cod\_cliente);

cout << " Nome: " << cliente.getNome() << endl;

cout << " RG: " << cliente.getRG() << endl;

cout << " CPF: " << cliente.getCPF() << endl;

cout << " Data de nascimento: " << cliente.getDataNasc() << endl;

cout << " Endereco: " << cliente.getEndereco() << endl;

cout << " Conta: " << cliente.getConta() << endl;

system("pause");

break;

}

}while(opcao2 != 0);

break;

}

}while(opcao3!=0);

return 0;

}