

Documentação de Sistema

Sistema de Cadastro e Reservas Hotel Descanso Garantido L.L.

Autores:
Luan Henrique Goiana de Melo
Lamôni Leal Pereira

Belo Horizonte
Junho de 2024

Documentação do Sistema

Documentação do Software	2
Introdução	3
Escopo do software.....	3
Nome do sistema e de seus componentes principais	3
Missão ou objetivo do software	3
Arquitetura do Sistema	4
Funcionalidades do produto.....	4
Usuários e sistemas externos	4
Descrição	4
Documentação do código	6
Documentação da Estrutura de dados geral do software	6
Função <nome da função>	6
Função <nome da função>	7
Testes do software	8
Casos de testes do software: função <nome da função>.....	8
Casos de testes do software: função <nome da função>.....	8

Introdução

Escopo do Sistema

Nome do sistema e de seus componentes principais

NOME: Descanso_Garantido_Hotel

DESCRIÇÃO: Sistema de controle de reservas e cadastros do Hotel “Descanso Garantido”

Classes Utilizadas:

Pessoa – Cliente / Funcionário

Quarto

Estadia

Principais Procedimentos:

void menu() – //Impressão de um menu escrito e descritivo para o usuário, expecificando opções de escolha.

void registrarDiaria(Quarto quartos[], int numQuartos) – //Procedimento de pesquisa e comparação dentro do sistema para registro de estadia do cliente

void pesquisarCliente() – //Procedimento para pesquisar clientes pelo código gerado no sistema

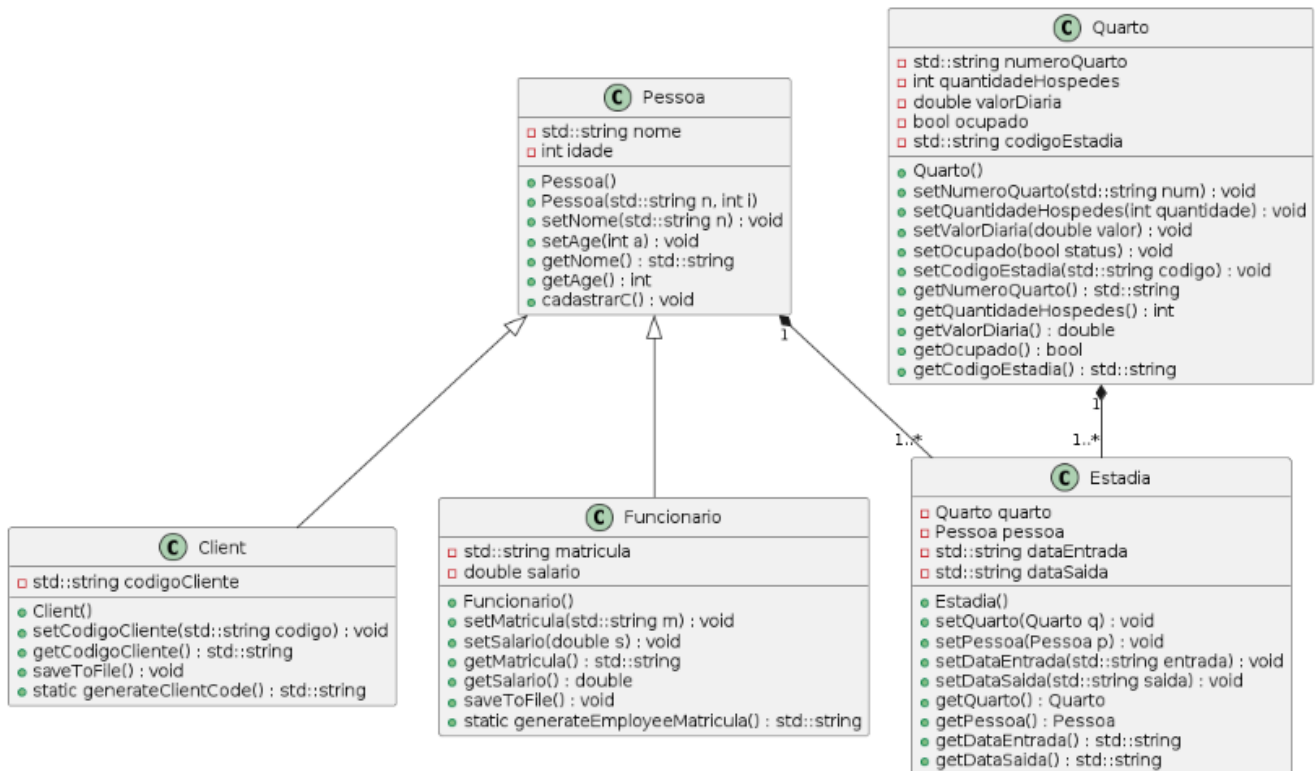
void pesquisarFuncionario() – //Procedimento para pesquisar funcionário pelo código gerado no sistema

Missão ou objetivo do sistema

Sistema para cadastrar, organizar e controlar numero de clientes, funcionários e quartos, também disponibilizar e vincular clientes e quartos a suas respectivas estadias. Controlar o acesso e agenda disponível para realizar cadastros e reservas, assim como disponibilizar por meio do cadastro salvo e uso regular, pontos de fidelidade aos clientes.

Arquitetura do Sistema

Diagrama de Classes:



Funcionalidades do produto

Número	Funcionalidade do sistema
1	Menu Interativo para uso do sistema
2	Controle de agenda e lotação dos quartos
3	Arquivamento de dados de funcionários e clientes
4	Controle de pontos de fidelidade de clientes repetidos
5	

Usuários e sistemas externos

Descrição

Número	Usuários	Definição
1	GITHUB	Plataforma Online de repositórios para atualização do código
2	CODEBLOCKS	Plataforma IDE para montagem do código
3	PLANTUML	Sistema modelador de diagramas UML via linguagem de código

Anexo – código para diagrama de classes plantUML

@startuml

```
class Hotel {
    + menu(): void
    + cadastrarCliente(): void
    + cadastrarFuncionario(): void
    + registrarDiaria(quartos: Quarto[], numQuartos: int): void
    + pesquisarCliente(): void
    + pesquisarFuncionario(): void
}
```

```
class Client {
    - nome: std::string
    - codigo: std::string
    - pontosFidelidade: int
    + cadastrarCliente(): void
    + saveToFile(filename: std::string): void
    + searchClient(keyword: std::string): bool
    + showInfo(): void
    + addPontosFidelidade(pontos: int): void
    + readFromFile(filename: std::string): void
}
```

```
class Funcionario {
    - nome: std::string
    - matricula: std::string
    + cadastrarFuncionario(): void
    + saveToFile(filename: std::string): void
    + searchFuncionario(keyword: std::string): bool
    + showInfo(): void
}
```

```
class Quarto {
    - numeroQuarto: std::string
    - quantidadeHospedes: int
    - valorDiaria: float
    - ocupado: bool
    - codigoEstadia: std::string
    + setNumeroQuarto(num: std::string): void
    + setQuantidadeHospedes(qtd: int): void
    + setValorDiaria(valor: float): void
    + getNumeroQuarto(): std::string
    + getOcupado(): bool
    + setOcupado(ocupado: bool): void
    + setCodigoEstadia(codigo: std::string): void
}
```

```
class Estadia {
    - dataEntrada: std::string
    - dataSaida: std::string
    - codigoCliente: std::string
    - quantidadeDiarias: int
    - codigoEstadia: std::string
    + generateCodigoEstadia(): void
    + setDataEntrada(data: std::string): void
    + setDataSaida(data: std::string): void
    + setCodigoCliente(codigo: std::string): void
    + setQuantidadeDiarias(qtd: int): void
    + getCodigoEstadia(): std::string
    + saveToFile(filename: std::string): void
}
```

Documentação do código

Documentação da Estrutura de dados geral do software

```
/*H*****  
****
```

```
* FILENAME : Descanso_Garantido_Hotel DESIGN
```

```
*
```

```
* DESCRIPTION :
```

```
* Sistema de controle de cadastros e reservas do Hotel "Descanso Garantido"
```

```
*
```

```
* PUBLIC FUNCTIONS :
```

```
* void menu()
```

```
* void cadastrarCliente()
```

```
* void cadastrarFuncionario();
```

```
* void registrarDiaria(Quarto quartos[], int numQuartos);
```

```
* void pesquisarCliente();
```

```
* void pesquisarFuncionario();
```

```
*
```

```
* NOTES :
```

```
* Sistema de controle e cadastro de clientes funcional e dinâmica.
```

```
* Não utilizamos Funções dentro do código principal, apenas nas classes
```

```
*
```

```
* AUTHOR(s) :
```

```
* LUAN HENRIQUE GOIANA de MELO
```

```
* LAMÔNÍ LEAL PEREIRA
```

```
* Other START DATE : 17 de JUN de 2024
```

```
*
```

```
* CHANGES :
```

```
* 17Jun2024 - Inicializando construção do código e das classes
```

```
* 25Jun2024 - Continuando construção das classes
```

```
* 26Jun2024 - Finalizando código de classes
```

```
* 28Jun2024 - Construindo Menu e procedimentos do código
```

```
* 30Jun2024 - Finalizando código, reparando erros e documentando código
```

```
*H*/
```

Função/Procedimento <void registrarDiaria(Quarto quartos[], int numQuartos)>

```
/* * NAME: void registrarDiaria(Quarto quartos[], int numQuartos)
```

* *DESCRIPTION: Procedimento de pesquisa e comparação dentro do sistema para registro de estadia do cliente*

* *INPUTS : PARAMETERS:*

* *Quarto quartos[] Vetor da classe quarto*

*

* *int numQuartos Valor inteiro para pesquisa de quartos*

* *PROCESS:*

* *[1] Solicita dados p/ armazenar em variáveis auxiliares e no objeto da classe Estadia*

* *[2] Repetir por - Pesquisa de todos quartos - "i < numQuartos"*

* *[3] Procurar por quarto selecionado*

* *[4] Verificar ocupação do quarto (Ocupado/Desocupado)*

* *[5] Gravar e salvar quarto junto ao numero de dias da estadia*

* *[6] Fechar Repetição*

* *[7] Gravar e salvar cliente junto a estadia + anexar pontos de fidelidade*

* *NOTES:*

* *Procedimento mais extenso do código que vincula a maioria dos objetos utilizados;*

* *Procedimento central para funcionamento do sistema de cadastro e funcionalidade do hotel.*

*/

Função/Procedimento < void pesquisarFuncionario()>

*/** NAME: void pesquisarFuncionario()*

** DESCRIPTION: Procedimento para pesquisar funcionário pelo código gerado no sistema*

** INPUTS:*

* *string keyword String solicitada dentro do procedimento*

* *gerada por código de cadastro de funcionarios*

** PROCESS:*

* *[1] Solicitação de Código ao usuário*

* *[2] Criação de objeto Funcionario*

* *[3] Pesquisa por meio de método implementado no objeto da classe Funcionário - "searchFuncionario"*

* *[4] Impressao do resultado(Encontrado + Nome/Não encontrado)*

*/

Função/Procedimento < void pesquisarCliente() >

*/** NAME: void pesquisarCliente()*

* *DESCRIPTION: Procedimento para pesquisar clientes pelo código gerado no sistema*

* *INPUTS:*

* *string keyword* *String solicitada dentro do procedimento*

* *gerada por código de cadastro de Cliente*

* *PROCESS:*

* *[1] Solicitação de Código ao usuário*

* *[2] Criação de objeto cliente*

* *[3] Pesquisa por meio de método implementado no objeto da classe Clientes - "searchClient"*

* *[4] Impressao do resultado(Encontrado + Nome/Não encontrado)*

* /

Testes do sistema

Casos de testes do sistema: função < void registrarDiaria(Quarto quartos[], int numQuartos)>

Númer o	Varáveis de Entrada	Valores válidos	Resultado Esperado	Valores inválidos	Resultado Espe
1	Data de entrada, Data de saída	01/01/2024, 05/01/2024	Estadia registrada com sucesso	01/01/2024, 32/01/2024	sobre a data de saída inválida
2	Código do cliente	CL12345	Estadia registrada com sucesso	CL99999	cliente não encontrado
3	Quantidade de diárias	5	Estadia registrada com sucesso	-1	quantidade de diárias inválida
4	Número do quarto	Q1	Estadia registrada com sucesso	Q100	número do quarto não encontrado
...	Dados completos	CL12345, 01/01/2024 05/01/2024, Q1, 5 diárias	Estadia registrada com sucesso e pontos de fidelidade atualizados	CL12345, 01/01/2024 32/01/2024 Q1, -1	data e quantidade de diárias inválida

Casos de testes do sistema: função <nome da função>

Número	Varáveis de Entrada	Valores válidos	Resultado Esperado	Valores inválidos	Resultado Esperado
1	Código do cliente	CL12345	Cliente encontrado	CL99999	cliente não encontrado

			e informações exibidas		
2	Nome do cliente	João Silva	Cliente encontrado e informações exibidas	Nome não registrado	Mensagem de cliente não encontrado
3	Código do cliente	CL98765	Cliente encontrado e informações exibidas		cliente não encontrado
4	Código inválido	"" (vazio)	Mensagem de cliente não encontrado	CL* & ^%\$#@!	Mensagem de erro e cliente não encontrado
5	Código do cliente	CL12345	Cliente encontrado e informações exibidas	CL00000	Mensagem de cliente não encontrado