

PONTÍFICA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS E INFORMÁTICA
UNIDADE EDUCACIONAL PRAÇA DA LIBERDADE
Bacharelado em Engenharia de Software



Ana Luiza Machado Alves
Enzo Caetano Dias Costa
Fabiana Dinelli Salvático de Paula Kraft
João Gabriel Amorim Pádua
Rúbia Coelho de Matos

Sumário

APRESENTAÇÃO:	1
BACKLOG DO PRODUTO:	1
LISTA DE ASSINATURAS DAS FUNÇÕES E PARÂMETROS	10
1. VARIÁVEIS PRÉ-DEFINIDAS:	10
2. VARIÁVEIS GLOBAIS:	10
3. DEFINIÇÃO DOS STRUCTS:	10
4. DEFINIÇÃO DAS FUNÇÕES:	12
Função 1 – Cadastro do cliente	12
Função 2 – Cadastro de funcionário	12
Função 3 – Inicializador do array de structs Quarto quarto[N]	12
Função 4 – Calcula a diferença entre datas de entrada e de saída informadas	12
Função 5 – Cadastro de estadia	12
Função 6 – Dar baixa na estadia	12
Função 7 – Pesquisa um cliente	12
Função 8 – Pesquisa um funcionário	12
Função 9 – Mostra a quantidade de estadias de um cliente	12
Função 10 – Mostra os pontos de fidelidade de um cliente	12
Função 11 – Escreve os dados dos arrays de structs em arquivos correspondentes	12
Função 12 – Lê os dados armazenados nos arquivos	12
5. MAIN	12
TESTES	15
CASOS DE TESTE DE SOFTWARE	15
Função 1 – Cadastro de Clientes	15
Função 2 – Cadastro de funcionário	16
Função 3 – Inicializador do array de structs Quarto quarto[N]	16

Função 4 – Calcula a diferença entre datas de entrada e de saída informadas	17
Função 5 – Cadastro de estadia	17
Função 6 – Dar baixa na estadia	18
Função 7 – Pesquisa um cliente	19
Função 8 – Pesquisa um funcionário	19
Função 9 – Mostra a quantidade de estadias de um cliente	20
Função 10 – Mostra os pontos de fidelidade de um cliente	20
Função 11 – Escreve os dados dos arrays de structs em arquivos correspondentes	20
Função 12 – Lê os dados armazenados nos arquivos	21
RELATÓRIO DE EXECUÇÃO DE TESTES	23
Teste 1 – Menu	23
Teste 2 - Cadastro do cliente	23
Teste 3 - Cadastro do funcionário	24
Teste 4 – Cadastro de Estadia	25
Teste 5 – Pesquisar Cliente	26
Teste 6 – Pesquisar Funcionário	26
Teste 7 - Verificar estadias do cliente	27
Teste 8 - Verificar pontos de fidelidade do cliente	28
Teste 9 – Salvar arquivo	29
Teste 10 – Imprimir arquivo	30

Apresentação:

O objetivo do sistema é auxiliar no gerenciamento e armazenamento de dados do Hotel Descanso Garantido de maneira intuitiva e organizada. O programa foi desenvolvido utilizando a linguagem C aplicando todo o conhecimento técnico abstraído nas aulas de Algoritmos e Estruturas de Dados I em conjunto com a aplicação de tecnologias e práticas de gerência de projetos aprendidas na disciplina de Engenharia de Software.

Backlog do produto:

Apresentação do backlog do produto a cada semana.

Figura 1 – Time e quadros criados no Trello

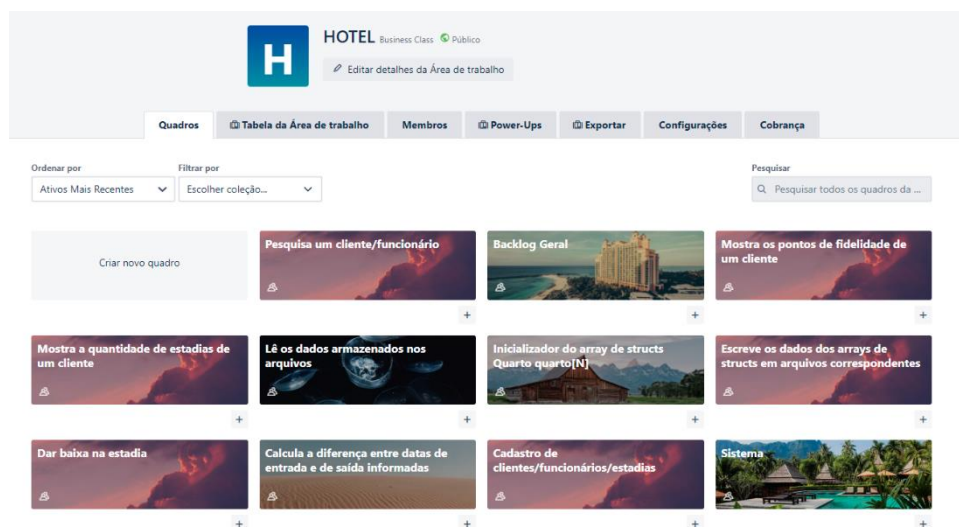


Figura 1: apresenta o time criado no Trello e as divisões dos quadros.

Figura 2 - Backlog geral

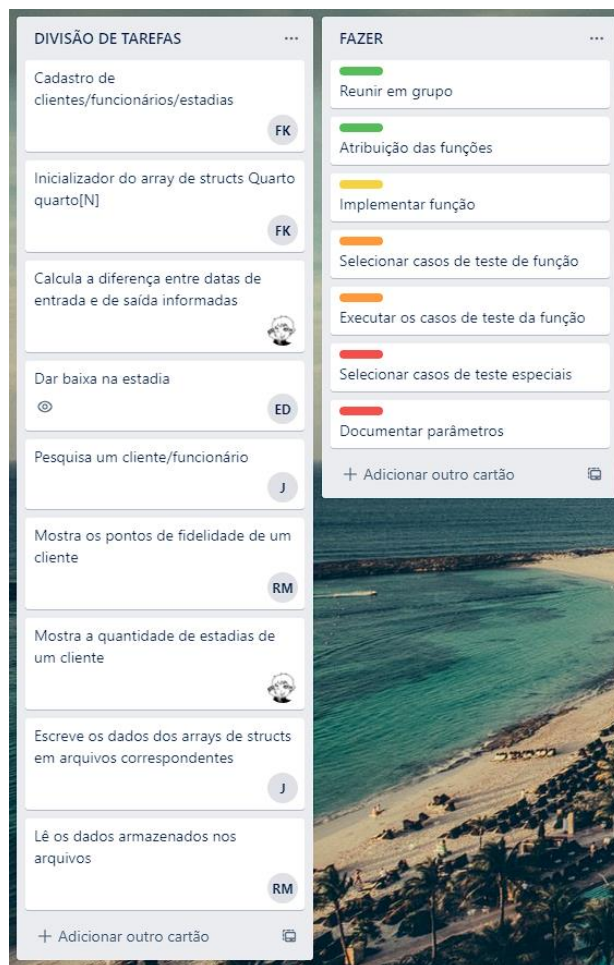


Figura 2: apresenta o backlog geral do projeto, mostrando a divisão das funções para cada membro do grupo e a ordem da entrega de cada tarefa por ordem das sprints.

As demais imagens demonstrarão o status do backlog do produto ao fim de cada sprint.

Sprint 1



000 Quadro ▾ **Calcula a diferença entre datas de entrada e de saída informadas** ☆ HOTEL BC

FAZER ...

- Selecione casos de teste de função
- Execute os casos de teste da função
- Selecione casos de teste especiais
- Documentar parâmetros
- + Adicionar outro cartão

FAZENDO ...

- Implementar função
- + Adicionar outro cartão

FEITO ...

- Reunir em grupo
- Atribuição das funções
- + Adicionar outro cartão

000 Quadro ▾ **Dar baixa na estadia** ☆ HOTEL BC Visível à Área de trabalho ED Convidar

FAZER ...

- Selecione casos de teste de função
- Execute os casos de teste da função
- Selecione casos de teste especiais
- Documentar parâmetros
- + Adicionar outro cartão

FAZENDO ...

- Implementar função
- + Adicionar outro cartão

FEITO ...

- Reunir em grupo
- Atribuição das funções
- + Adicionar outro cartão

000 Quadro ▾ **Escreve os dados dos arrays de structs em arquivos correspondentes** ☆ HOTEL BC

FAZER ...

- Selecione casos de teste de função
- Execute os casos de teste da função
- Selecione casos de teste especiais
- Documentar parâmetros
- + Adicionar outro cartão

FAZENDO ...

- Implementar função
- + Adicionar outro cartão

FEITO ...

- Reunir em grupo
- Atribuição das funções
- + Adicionar outro cartão

000 Quadro ▾ **Inicializador do array de structs Quarto quarto[N]** ☆ HOTEL BC Visível à Área de trabalho

FAZER ...

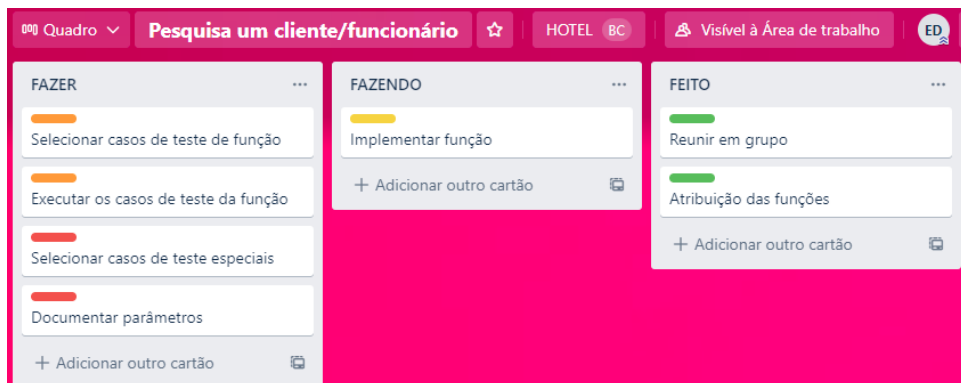
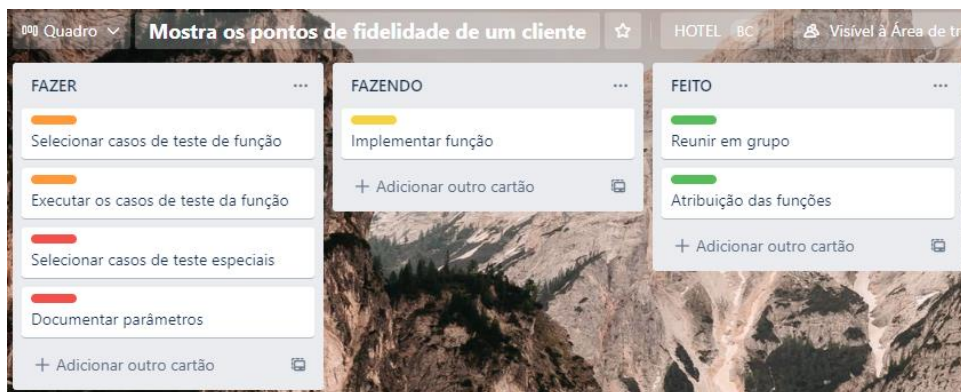
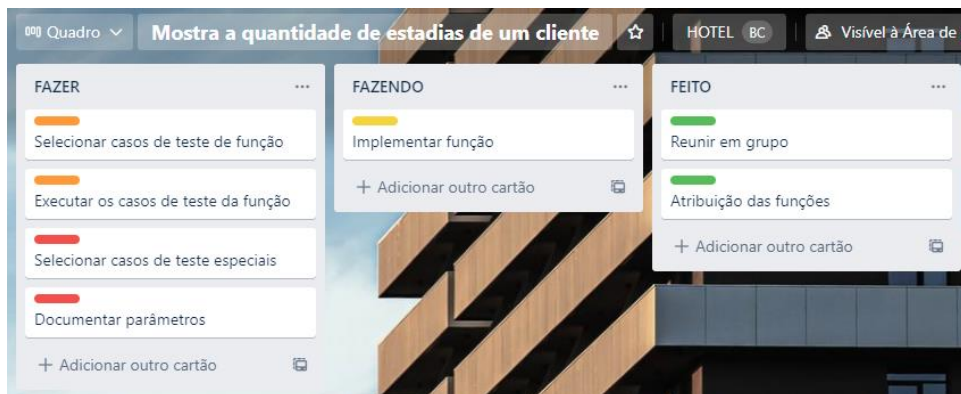
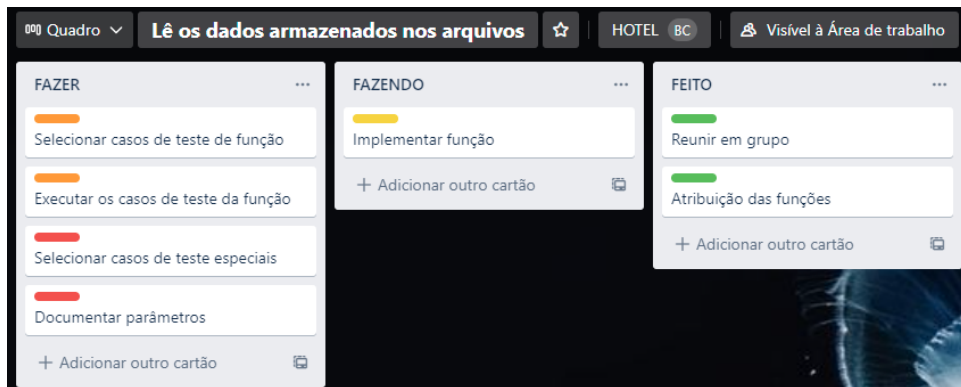
- Selecione casos de teste de função
- Execute os casos de teste da função
- Selecione casos de teste especiais
- Documentar parâmetros
- + Adicionar outro cartão

FAZENDO ...

- Implementar função
- + Adicionar outro cartão

FEITO ...

- Reunir em grupo
- Atribuição das funções
- + Adicionar outro cartão



Sprint 2

Quadro ▾ **Cadastro de clientes/funcionários/estadias** ☆ HOTEL BC Visível à Área de trabalho

FAZER	FAZENDO	FEITO
<div><div></div>Documentar parâmetros</div>	<div><div></div>Executar os casos de teste da função</div>	<div><div></div>Reunir em grupo</div>
<div>+ Adicionar outro cartão</div>	<div><div></div>Selecionar casos de teste especiais</div>	<div><div></div>Atribuição das funções</div>
	<div>+ Adicionar outro cartão</div>	<div><div></div>Implementar função</div>
		<div><div></div>Selecionar casos de teste de função</div>
		<div>+ Adicionar outro cartão</div>

Quadro ▾ **Calcula a diferença entre datas de entrada e de saída informadas** ☆ HOTEL BC

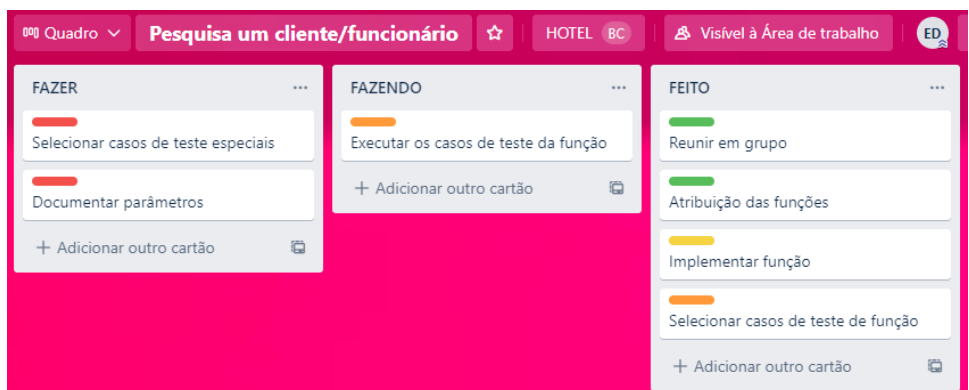
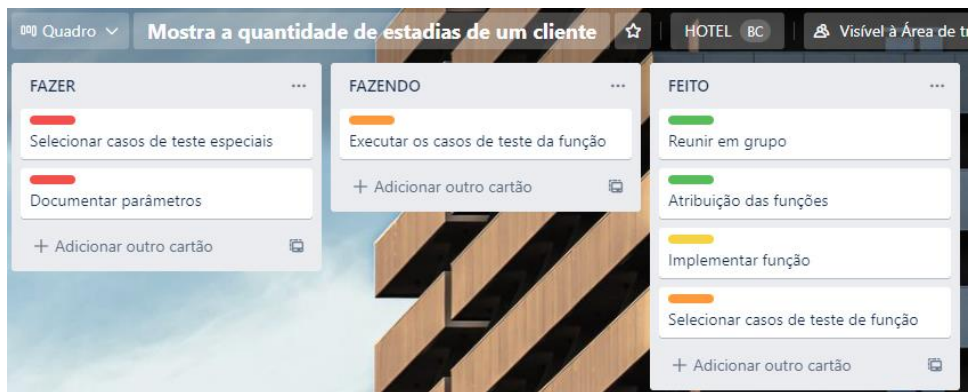
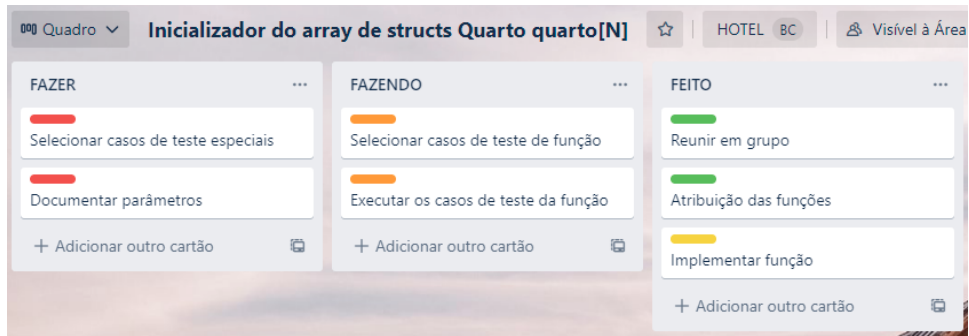
FAZER	FAZENDO	FEITO
<div><div></div>Selecionar casos de teste especiais</div>	<div><div></div>Selecionar casos de teste de função</div>	<div><div></div>Reunir em grupo</div>
<div><div></div>Documentar parâmetros</div>	<div><div></div>Executar os casos de teste da função</div>	<div><div></div>Atribuição das funções</div>
<div>+ Adicionar outro cartão</div>	<div>+ Adicionar outro cartão</div>	<div><div></div>Implementar função</div>
		<div>+ Adicionar outro cartão</div>

Quadro ▾ **Dar baixa na estadia** ☆ HOTEL BC Visível à Área de trabalho ED Convidar

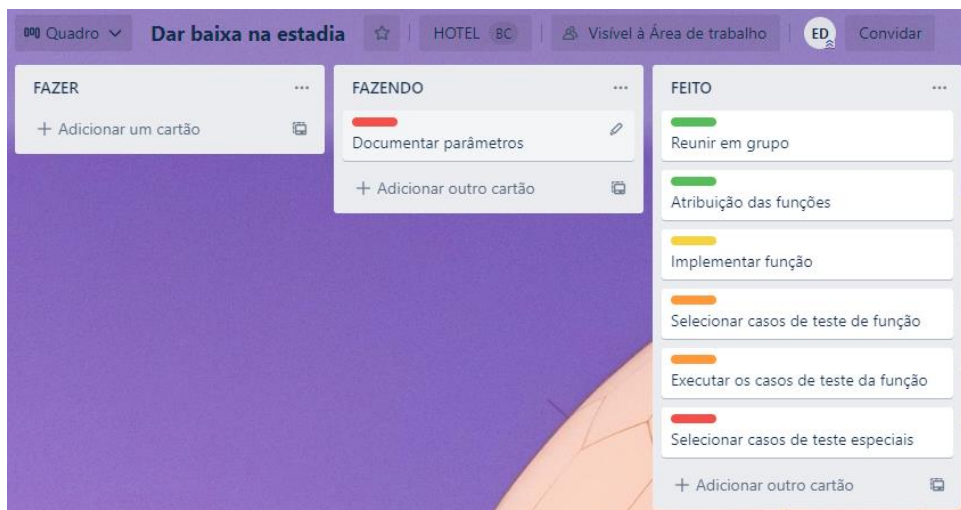
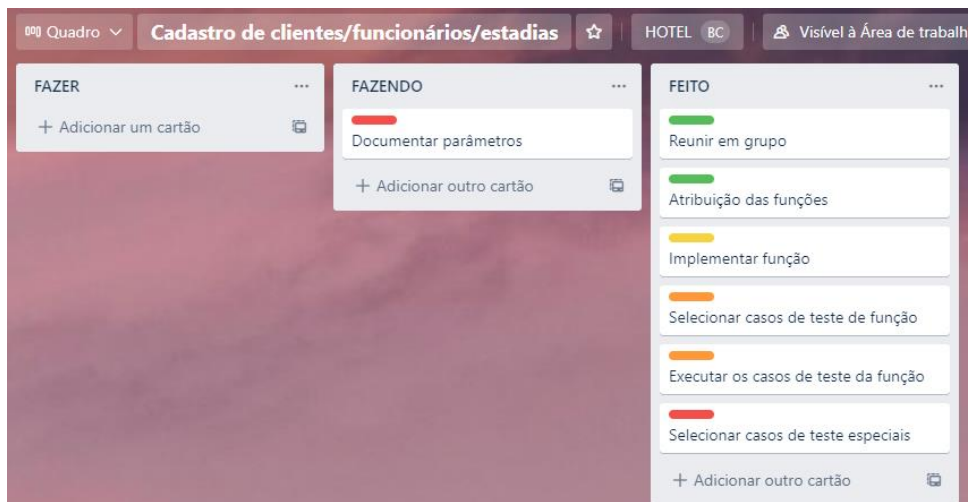
FAZER	FAZENDO	FEITO
<div><div></div>Documentar parâmetros</div>	<div><div></div>Executar os casos de teste da função</div>	<div><div></div>Reunir em grupo</div>
<div>+ Adicionar outro cartão</div>	<div><div></div>Selecionar casos de teste especiais</div>	<div><div></div>Atribuição das funções</div>
	<div>+ Adicionar outro cartão</div>	<div><div></div>Implementar função</div>
		<div><div></div>Selecionar casos de teste de função</div>
		<div>+ Adicionar outro cartão</div>

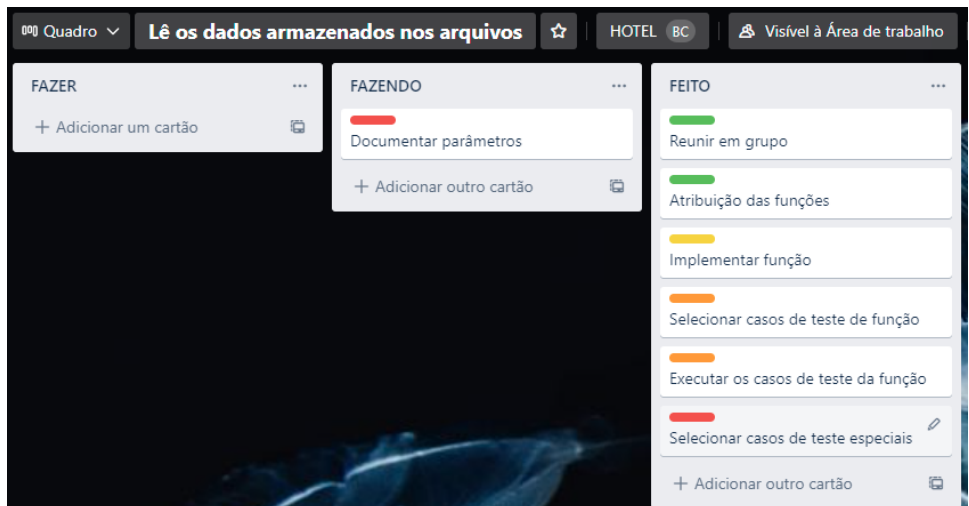
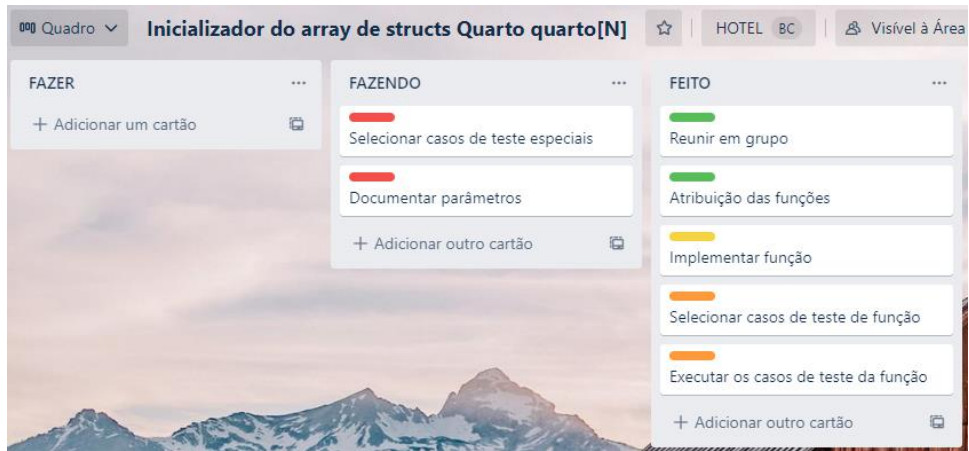
Quadro ▾ **Escreve os dados dos arrays de structs em arquivos correspondentes** ☆ HOTEL BC

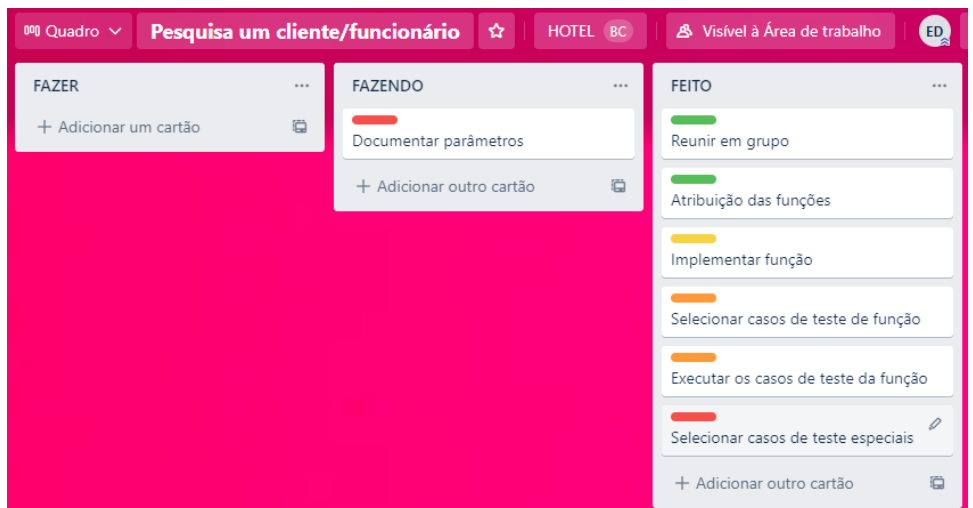
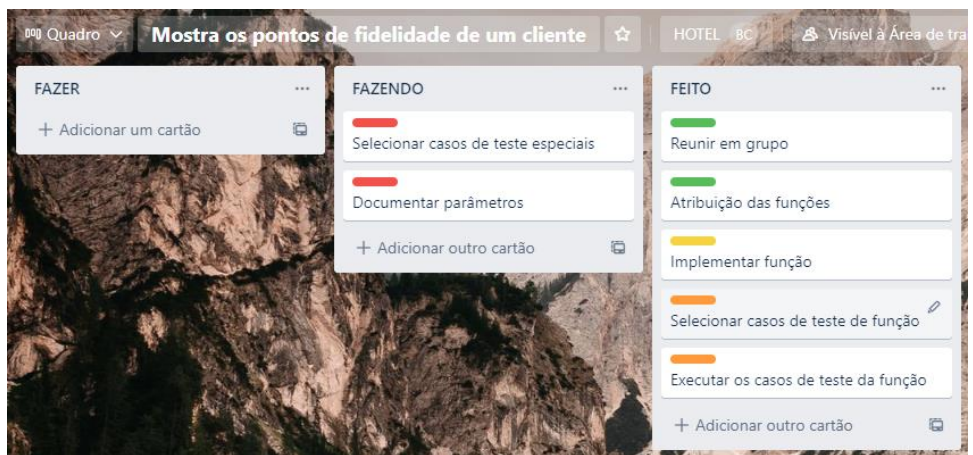
FAZER	FAZENDO	FEITO
<div><div></div>Selecionar casos de teste especiais</div>	<div><div></div>Executar os casos de teste da função</div>	<div><div></div>Reunir em grupo</div>
<div><div></div>Documentar parâmetros</div>	<div>+ Adicionar outro cartão</div>	<div><div></div>Atribuição das funções</div>
<div>+ Adicionar outro cartão</div>		<div><div></div>Implementar função</div>
		<div><div></div>Selecionar casos de teste de função</div>
		<div>+ Adicionar outro cartão</div>



Sprint 3







Lista de assinaturas das funções e parâmetros

1. Variáveis pré-definidas:

- N = valor utilizado para definir os tamanhos dos arrays de structs; tipo: int
- MAX_H = valor máximo de hóspedes permitidos no hotel; tipo: int
- MIN_H = valor mínimo de hóspedes permitidos no hotel; tipo: int

2. Variáveis Globais:

- i_cliente = valor da próxima posição a ser preenchida no array do struct de clientes; tipo: int
- i_funcionario = valor da próxima posição a ser preenchida no array do struct de funcionários; tipo: int
- i_estadia = valor da próxima posição a ser preenchida no array do struct de estadias; tipo: int

3. Definição dos Structs:

- Cliente = contém os dados a serem armazenados dos clientes:
 - codigo = valor aleatório e único entre 0 e N representando o id do cliente → tipo: int
 - nome = nome do cliente cadastrado → tipo: string (char[100])
 - endereco = dados de moradia do cliente → tipo: string (char[100])
 - Telefone = número de contato telefônico do cliente → tipo: string (char[20])
- Funcionario = contém os dados a serem armazenados sobre os funcionários que trabalham no hotel
 - codigo = valor aleatório e único entre 0 e N representando o id do funcionário → tipo: int
 - nome = nome do funcionário cadastrado → tipo: string (char[100])

- cargo = ocupação do funcionário no hotel → tipo: string (char[100])
 - telefone = número de contato telefônico do cliente → tipo: string (char[20])
 - Salario = valor monetário mensal recebido pelo funcionário do hotel → tipo: real (double)
- Estadia = contém os dados sobre a estadia cadastrada para determinado cliente
 - codigo = valor único correspondente ao id da estadia → tipo: int (no presente trabalho, com intuito de simplificação, optou-se por utilizar o código igual à posição do elemento no array de struct de estadias)
 - data_entrada = data em formato dia/mês/ano referente ao check-in (data de entrada) do cliente → tipo: string (char[11])
 - data_saida = data em formato dia/mês/ano referente ao check-out (data de saída) do cliente → tipo: string (char[11])
 - quantidade_diarias = valor numérico obtido por meio do cálculo da diferença entre data de saída e data de entrada → tipo: int
 - codigo_cliente = id do cliente, obtido do struct Cliente → tipo: int
 - numero_quarto = número do quarto do hotel obtido do struct Quarto → tipo: int
- Quarto = contém os dados referentes aos quartos presentes no hotel
 - numero = número único correspondente a cada quarto do hotel → tipo: int (no presente trabalho, com intuito de simplificação, optou-se por utilizar o número igual à posição do elemento no array de struct de quartos)
 - quantidade_hospedes = valor entre MAX_H e MIN_H, representando o número de hóspedes hospedados no quarto → tipo: int
 - valor_diaria = valor em reais referente ao custo de uma diária no quarto → tipo: double
 - status = definição se o quarto em questão está ocupado (true) ou desocupado (false) → tipo: booleano

4. Definição das funções:

Função 1 – Cadastro do cliente

Função 2 – Cadastro de funcionário

Função 3 – Inicializador do array de structs Quarto quarto[N]

Função 4 – Calcula a diferença entre datas de entrada e de saída informadas

Função 5 – Cadastro de estadia

Função 6 – Dar baixa na estadia

Função 7 – Pesquisa um cliente

Função 8 – Pesquisa um funcionário

Função 9 – Mostra a quantidade de estadias de um cliente

Função 10 – Mostra os pontos de fidelidade de um cliente

Função 11 – Escreve os dados dos arrays de structs em arquivos correspondentes

Função 12 – Lê os dados armazenados nos arquivos

5. Main

Declaração de variáveis:

- Array de structs do tipo Cliente, com nome “cliente”, de tamanho N;
- Variáveis para input: strings nome (representando o nome do cliente), endereco (representando o endereço do cliente), e telefone (representando o telefone do cliente), respectivamente com tamanhos 100, 1000 e 20;
- Array de structs do tipo Funcionario, com nome “funcionario”, de tamanho N;
- Variáveis para input: strings nome_f (representando o nome do funcionario), telefone_f (representando o telefone do funcionario), cargo_f (representando o cargo do funcionario), de tamanhos respectivos 100, 20 e 100; variável real do tipo double chamada salario_f (representando o salário do funcionario), com valor inicial 0.0;

- Array de structs do tipo Estadia, com nome “estadia”, de tamanho N;
- Variáveis para input: inteiros chamados diaria (representando quantidade de diárias), d_e, d_s, m_e, m_s, a_e, a_s (representando dia, mês e ano de entrada e saída), quantidade_hospedes (representando quantidade de hóspedes), codigo_estadia (representando código da estadia); strings nome_cliente (representando o nome do cliente), data_entrada (representando a data de entrada), data_saida (representando a data de saída), de tamanhos 100, 11 e 11, respectivamente.
- Inteiro x, correspondente ao input da opção a ser colocada no menu switch-case pelo usuário.

Estrutura switch-case:

- Case 0: sai do menu e encerra a execução do programa;
- Case 1: cadastro do cliente: input pelo usuário do das variáveis nome, endereco e telefone. Chamada da função cadastro_cliente com os parâmetros cliente, nome, endereco e telefone;
- **Case 2:** cadastro do funcionario: input pelo usuário do das variáveis nome_f, telefone_f, cargo_f, salario_f. Teste com switch-case para verificar se o cargo do funcionário digitado encontra-se dentro das opções permitidas. Chamada da função cadastro_funcionario com os parâmetros funcionario, nome_f, telefone_f, cargo_f, salario_f;
- **Case 3:** cadastro da estadia: input pelo usuário do da variável nome_cliente. Declaração da variável local do tipo booleano chamada cadastrado, com valor inicial false. Teste se o cliente está cadastrado, se sim, cadastrado recebe true e a posição do cliente no array é salva na variável j; caso contrário, é necessário digitar novamente o nome do cliente, até que cadastrado torne-se true. Input pelo usuário da variável quantidade_hospedes; declaração da variável viavel do tipo booleano com valor inicial false; enquanto viavel for false, é feita a leitura da quantidade de hóspedes, sendo que, se o valor estiver dentro do permitido, ele é aceito e, ao mudar viavel para true, sai-se do loop. Input pelo usuário das variáveis referentes aos dias, meses e anos de entrada e saída; teste se são valores válidos; senão, permanece no loop para redigitar os valores. Chamada da função diarias para calcular a quantidade de diarias e armazenamento na

variável diaria; Passagem para strings contendo as datas no formato dd/mm/aaa. Por fim, cadastro da estadia, por meio da chamada da função cadastro_estadia

- **Case 4:** dar baixa na estadia: leitura da variável codigo_estadia e teste se é existente. Se existe, chama a função dar_baixa. Senão, permanece no loop para releitura do valor
- **Case 5:** pesquisa do cliente: leitura do input nome do cliente, salvo na variável nome. Chamada da função pesquisa_cliente.
- **Case 6:** pesquisa do funcionário: leitura do input nome do funcionário, salvo na variável nome_f. Chamada da função pesquisa_funcionario.
- **Case 7:** Verifica as estadias de um determinado cliente. Input do nome do cliente, salvo na variável nome. Teste se existem clientes e estadias cadastradas. Se sim, chama a função estadias_cliente; Senão, dá mensagem de erro e sai da estrutura do case.
- **Case 8:** Verifica os pontos de fidelidade de um cliente a partir do nome do mesmo informado pelo usuário, na variável nome. Chamada da função fidelidade.
- **Case 9:** escreve os arquivos, usando a função escrever_arquivo, passando como parâmetros os arrays de structs cliente, funcionario, estadia e quarto. Necessário chamar a função uma vez apenas para escrever os 4 arquivos
- **Case 10:** leitura dos arquivos com os nomes salvos anteriormente dentro da função escrever_arquivo, por meio da função ler_arquivo. Chamada de função para cada arquivo;

Testes

Casos de teste de software

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Função 1 – Cadastro de Clientes				
Nome	Letras. Exemplo: Ana Carolina Mendonça	Inserir Endereço	Números e Caracteres não letras	Saída não saiu como esperado
Endereço	Números, letras, vírgulas, hífen e barra. Exemplo: Rua Claudio Manoel, 125 – Funcionários – Belo Horizonte/MG – 31.303-205	Inserir Telefone	Não Existe	Não Existe
Telefone	Números de onze dígitos, dois parênteses e 1 hífen. (31) 9 9758-4063	Cadastro feito com sucesso, usuário volta para o menu	Letras	Saída não saiu como esperado

Função 2 – Cadastro de funcionário				
Nome	Letras e espaço. Exemplo: Beatriz Fernandes	Inserir Telefone	Números e Caracteres	Saída não saiu como esperado
Telefone	Números de onze dígitos, parênteses e hífen. (31) 9 8545-8796	Informe o cargo: 1 - Recepcionista 2 - Auxiliar de limpeza 3 - Garcom 4 - Gerente	Letras	Saída não saiu como esperado
Cargo	Números de 1 a 4. Exemplo: 3 - Garçom	Inserir o Salário	Números maiores que 4 e menores que 1, letras e caracteres	Cargo não existe!
Sálario	Números de 8 dígitos, vírgula. Exemplo: 1500,00	Cadastrado efetuado com sucesso, usuário volta para o menu	Letras, caracteres não numéricos exceto.	Saída não saiu como esperado
Função 3 – Inicializador do array de structs Quarto quarto[N]				
Pelo Código	Struts vazio	Quantidade de quartos necessários	Struts preenchido com dados iniciais	Saída não saiu como esperado

Função 4 – Calcula a diferença entre datas de entrada e de saída informadas				
Pelo código	Entrada: 01/02/2020 Saída: 02/03/2020	Tempo exato da diária	Entrada: 01/02/2020 Saída: 02/01/2020	Saída não saiu como esperado
Função 5 – Cadastro de estadia				
Menu	Número 3	Inserir Nome do Cliente	Diretamente no 3 sem antes cadastrar cliente	ERRO: Nao ha clientes cadastrados
Nome do Cliente	Letras. Exemplo: Ana Carolina Mendonça	Inserir a quantidade de hospedes	Números e Caracteres	ERRO: Nome do cliente nao encontrado, favor repetir a pesquisa:
Quantidade de hospedes	Números. Exemplo: 3	Inserir dia de entrada	Letras e caracteres não numéricos	ERRO: quantidade de hospedes invalida
Dia de Entrada	Números de 1 a 31. Exemplos: 04	Inserir mês de entrada	Letras e caracteres não numéricos, números maiores que 31 e menores que 1	ERRO: dia invalido

Mês de Entrada	Números de 1 a 12. Exemplos: 07	Inserir ano de entrada	Letras e caracteres não numéricos, números maiores que 12 e menores que 1	ERRO: mes invalido
Ano de Entrada	Números de 4 dígitos. Exemplos: 2020	Inserir dia de saída	Letras e caracteres não numéricos, números maiores que 3000 e menores que 1900	ERRO: ano invalido
Dia de Saída	Números de 1 a 31. Exemplos: 30	Inserir mes de saída	Letras e caracteres não numéricos, números maiores que 31 e menores que 1	ERRO: dia invalido
Mês de Saída	Números de 1 a 12. Exemplos: 07	Inserir ano de saída	Letras e caracteres não numéricos, números maiores que 12 e menores que 1	ERRO: mes invalido
Ano de Saída	Números de 4 dígitos. Exemplos: 2020	Usuário volta para o menu	Letras e caracteres não numéricos, números maiores que 3000 e menores que 1900	ERRO: ano invalido
Função 6 – Dar baixa na estadia				
Menu	4	Inserir nome	Diretamente no 3 sem antes cadastrar cliente e cadastrar a estadia	Não ha estadias cadastradas

Codigo	Número de 0 e (nº de hospedes cadastrados). Exemplo: 1	Valor total a ser pago=	Número menores de 0 e maiores que o número de hospedes cadastrados	ERRO: favor inserir codigo entre 0 e (nº de hospedes cadastrados)
Função 7 – Pesquisa um cliente				
Nome Do Cliente A Ser Pesquisado	Letras. Exemplo: Ana Carolina Mendonça	Codigo= 48 Endereço= Rua Claudio Manoel, 125 – Funcionários – Belo Horizonte/MG – 31.303-205 Telefone= (31) 9 9758-4063 Usuário volta para o menu	Números e caracteres	Nao foi encontrado cliente com o nome (nome do Cliente pesquisado)
Função 8 – Pesquisa um funcionário				
Nome Do Funcionário A Ser Pesquisado	Letras e espaço. Exemplo: Beatriz Fernandes	Codigo= 45 Telefone=(31) 9 8545-8796 Cargo=Garcom Salario=1500.00 Usuário volta para o menu	Números e caracteres	ERRO: Nao ha funcionarios

Função 9 – Mostra a quantidade de estadias de um cliente				
Nome	Letras. Exemplo: Ana Carolina Mendonça	Estadias do cliente Ana Carolina Mendonça Codigo da estadia:0 Data de entrada:4/7/2020 Data de saida:30/7/2020 Quantidade de diarias:26 Numero do quarto:8 Usuário volta para o menu	Números e caracteres	Nao ha estadias para o cliente (nome do Cliente pesquisado)
Função 10 – Mostra os pontos de fidelidade de um cliente				
Nome	Letras. Exemplo: Ana Carolina Mendonça	Pontos de fidelidade do cliente Ana Carolina Mendonça Pontos de fidelidade do cliente= 260 Usuário volta para o menu	Usuário sem estadias cadastradas	Nao ha estadias para o cliente (nome do Cliente pesquisado)
Função 11 – Escreve os dados dos arrays de structs em arquivos correspondentes				
Menu	Número inteiro 9	Dados salvos com sucesso	Erro de código	Saída não saiu como esperado

Função 12 – Lê os dados armazenados nos arquivos

Menu	Número 10	<p>Clientes:</p> <p>codigo= 48</p> <p>nome= Ana Carolina Mendonça</p> <p>endereco= Rua Claudio Manoel, 125 - Funcionários - Belo Horizonte/MG - 31.303-2 05</p> <p>telefone= (31) 9 9758-4063</p> <p>Funcionarios:</p> <p>codigo= 45</p> <p>nome= Beatriz Fernandes</p> <p>telefone= (31) 9 8568-5846</p> <p>cargo= Garcom salario= R\$1500.00</p>	Não tem os arquivos no sistema	ERRO de abertura de arquivo
------	-----------	---	--------------------------------	-----------------------------

		Estadias: codigo= 0 data entrada= 12/5/2020 data saida= 22/5/2020 quantidade de diarias= 10 codigo do cliente= 48 numero do quarto= 8 Quartos ocupados:		
--	--	--	--	--

Relatório de execução de testes

Teste 1 – Menu

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Números Inteiro	Números entre 0 e 9	Entrada na função correspondente à opção escolhida	Números menores de 0 ou maiores de 9	Solicitar a entrada na opção válida
Relatório de Execução de Testes				
Entradas		Resultado	Aprovado?	
0		Sair do sistema	Sim	
1		Cadastrar cliente	Sim	
2		Cadastrar funcionário	Sim	
3		Cadastrar a estadia	Sim	
4		Fazer Checkout	Sim	
5		Pesquisar Cliente	Sim	
6		Pesquisar Funcionário	Sim	
7		Verificar estadia de cliente	Sim	
8		Verificar pontos de fidelidade do cliente	Sim	
9		Salvar arquivos	Sim	
10		Imprimir arquivos	Sim	
a		default	Sim	

Teste 2 - Cadastro do cliente

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Nome Endereço telefone	Números, letras e símbolos	Código do cliente Nome endereço telefone	N/A	Cadastro com sucesso
Relatório de Execução de Testes				

Entradas	Resultado	Aprovado?
FULANO; Rua x 123; (31)3888-4444	Nome do cliente = FULANO Código = 48 Endereço = Rua x 123 Telefone = (31)3888-4444	SIM
777; abc; 12x	Nome do cliente = 777 Código = 45 Endereço = abc Telefone = 12x	SIM

Teste 3 - Cadastro do funcionário

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Nome Telefone Cargo Salário	Números, letras e símbolos Salário valor real	Código Telefone Cargo Salário	Salário com letras e símbolos diferentes de "." ou mais de um "."	Erro de execução do programa
Relatório de Execução de Testes				
Entradas	Resultado		Aprovado?	
Paulo, abc123, 3, 123.123	Nome do funcionario= Paulo Codigo= 13 Telefone=abc123 Cargo=Garcom Salario=123.12		SIM	
JOAQUIM DOS MILAGRES; 3333-3333; 4; 1000.0000	Nome do funcionario= JOAQUIM DOS MILAGRES Codigo= 8 Telefone=3333-3333 Cargo=Gerente		SIM	

	Salario=1000.00	
j, abc123, 3, a	Fica no loop e pula nome do funcionario, inicia o input a partir do próximo valor. Ao chamar a função 6, imprime o salário como 0	NÃO

Teste 4 – Cadastro de Estadia

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Nome, quantidade de hóspedes, dia/mês/ano de entrada, dia/mês/ano de saída	Números e letras (apenas para o nome)	Código, data de entrada, data de saída, quantidade de diárias, código do cliente, número do quarto	Símbolos e números decimais, nomes de clientes não cadastrados, número de hóspedes < 1 e >10, dia/mês/ano inválidos	Cadastro realizado e armazenado em um arquivo com sucesso
Relatório de Execução de Testes				
Entradas			Resultado	Aprovado?
Cláudio Rodrigues; 06, 07, 2021, 09, 07, 2021;			Código=0 Data de entrada=06/07/2021 Data de saída=09/07/2021 Quantidade de diárias:=3 Código do cliente:=48 Número do quarto=3	SIM
J; 06, 07, 2021, 06, 07, 2021;			Código=1 Data de entrada=06/07/2021 Data de saída=06/07/2021 Quantidade de diárias=0 Código do cliente=48 Número do quarto=3	SIM
JOSE GUILHERME DA SILVA; ABC;			Código=3 Data de entrada=03/01/2021	NÃO

	Data de saída=01/02/2021 Código do cliente=8 Número do quarto=12	
--	--	--

Teste 5 – Pesquisar Cliente

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Nome do cliente	Nome dos clientes já cadastrados	Nome Código Endereço Telefone	Nomes de clientes não cadastrados ou se não houve ainda cadastro	Mensagem de erro
Relatório de Execução de Testes				
Entradas		Resultado	Aprovado?	
Pedro (CADASTRADO)		Nome do cliente= Pedro Codigo= 48 Endereco=Rua da Bahia Telefone=(31)99494-5060	SIM	
João (não cadastrado)		Nao foi encontrado cliente com o nome Joao	SIM	
Lucas(não houve cadastro ainda realizado para nenhum cliente)		ERRO: Nao ha clientes cadastrados	SIM	

Teste 6 – Pesquisar Funcionário

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Nome do funcionário	Nomes de funcionários já cadastrados	Nome Código Telefone Cargo Salário	Nomes de funcionários não cadastrados ou se não houve ainda cadastro	Mensagem de erro
Relatório de Execução de Testes				

Entradas	Resultado	Aprovado?
Paulo (já cadastrado)	Nome do funcionario= Paulo Codigo= 13 Telefone=abc123 Cargo=Garcom Salario=123.12	SIM
Pessoa (não cadastrado)	Nao foi encontrado funcionario com o nome Pessoa	SIM
Abc (não houve cadastro ainda realizado para nenhum funcionário)	ERRO: Nao ha funcionarios cadastrados	SIM

Teste 7 - Verificar estadias do cliente

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Nome do Cliente	Cliente já cadastradas	Codigo da estadia: Data de entrada: Data de saída: Quantidade de diárias: Numero do quarto:	Cliente não cadastrado	Mensagem de Erro
Relatório de Execução de Testes				
Entradas		Resultado	Aprovado?	
Ana Carolina Mendonça		Codigo da estadia: 45 Data de entrada: 01/07/2020 Data de saída: 07/07/2020 Quantidade de diárias: 7 Numero do quarto:	Sim	
Joaquim Fernandes		Codigo da estadia: 48 Data de entrada: 01/04/2020	Sim	

	Data de saída: 15/04/2020 Quantidade de diárias: 15 Numero do quarto: 05	
Bernardo Martins	Codigo da estadia: 46 Data de entrada: 20/05/2020 Data de saída: 10/05/2020 Quantidade de diárias: 10 Numero do quarto: 03	Sim
Cristina Bittencourt	Codigo da estadia: 47 Data de entrada: 25/06/2020 Data de saída: 30/06/2020 Quantidade de diárias: 5 Numero do quarto: 02	sim

Teste 8 - Verificar pontos de fidelidade do cliente

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Nome do cliente	Letras	Mostrar pontos que o cliente possui	Números e Caracteres	Pedir para digitar novamente o nome do cliente
Relatório de Execução de Testes				
Entradas		Resultado	Aprovado?	
Ana Carolina Mendonça		Pontos de fidelidade do cliente= 50	Sim	
Joaquim Fernandes		Pontos de fidelidade do cliente= 260	Sim	
Beatriz Guimarães		Pontos de fidelidade do cliente= 150	Sim	

Cristina Bittencourt	Pontos de fidelidade do cliente= 200	Sim
Bernardo Martins	Pontos de fidelidade do cliente= 350	Sim
Maria Eduarda Napolitano	Pontos de fidelidade do cliente= 260	Sim
Jorge Magalhães	Pontos de fidelidade do cliente= 90	Sim
Lucas Amorim	Pontos de fidelidade do cliente= 70	Sim

Teste 9 – Salvar arquivo

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
cliente, funcionario, estadia , quarto	Array de Structs preenchidos ou não	Geração de um arquivo para cada array	-	-
Relatório de Execução de Testes				
Entradas			Resultado	Aprovado?
Array preenchido de clientes			Clientes: codigo= 48 nome= ABC endereco= rua q telefone= 123456	SIM
			codigo= 45 nome= DEF endereco= rua 1 telefone= 111	
			codigo= 13 nome= FGH endereco= rua 12 telefone= 111111	

Array não preenchido de clientes	Clientes:	SIM
----------------------------------	-----------	-----

Teste 10 – Imprimir arquivo

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Arquivos com dados salvos	Arquivo.txt	Dados de cliente, funcionários, estadia e pontos de fidelidade	Arquivos de outros extensões	

Relatório de Execução de Testes

Entradas	Resultado	Aprovado?
Clientes.txt	Documentos exibindo	Sim
Funcionarios.txt	Documentos exibindo	Sim
Estadias.txt	Documentos exibindo	Sim
Quartosocupados.txt	Documentos exibindo	Sim