

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

Nome dos integrantes do grupo

Áulus Arcanjo Alves Batista

Nome do sistema

Hotel Descanso Garantido

Apresentação:

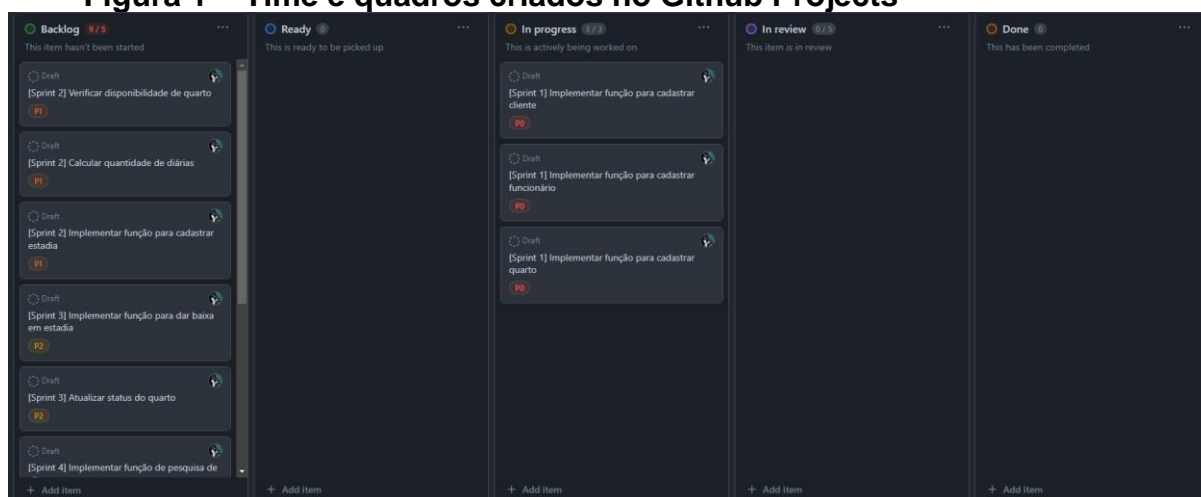
O sistema desenvolvido tem como objetivo automatizar e otimizar a gestão do Hotel Descanso Garantido, substituindo métodos manuais por uma solução digital. Através da integração de funções para cadastro de clientes, funcionários e estadias, o sistema busca garantir a eficiência operacional do hotel, evitando duplicidades de registros, gerenciando a disponibilidade de quartos e facilitando o processo de reserva e baixa em estadias. Com a capacidade de armazenar todas as informações em arquivos, o sistema visa proporcionar uma experiência mais organizada e ágil tanto para os clientes quanto para os funcionários, contribuindo para a fidelização dos clientes e o sucesso do empreendimento.

Backlog do produto:

Apresente o backlog do seu produto a cada semana.

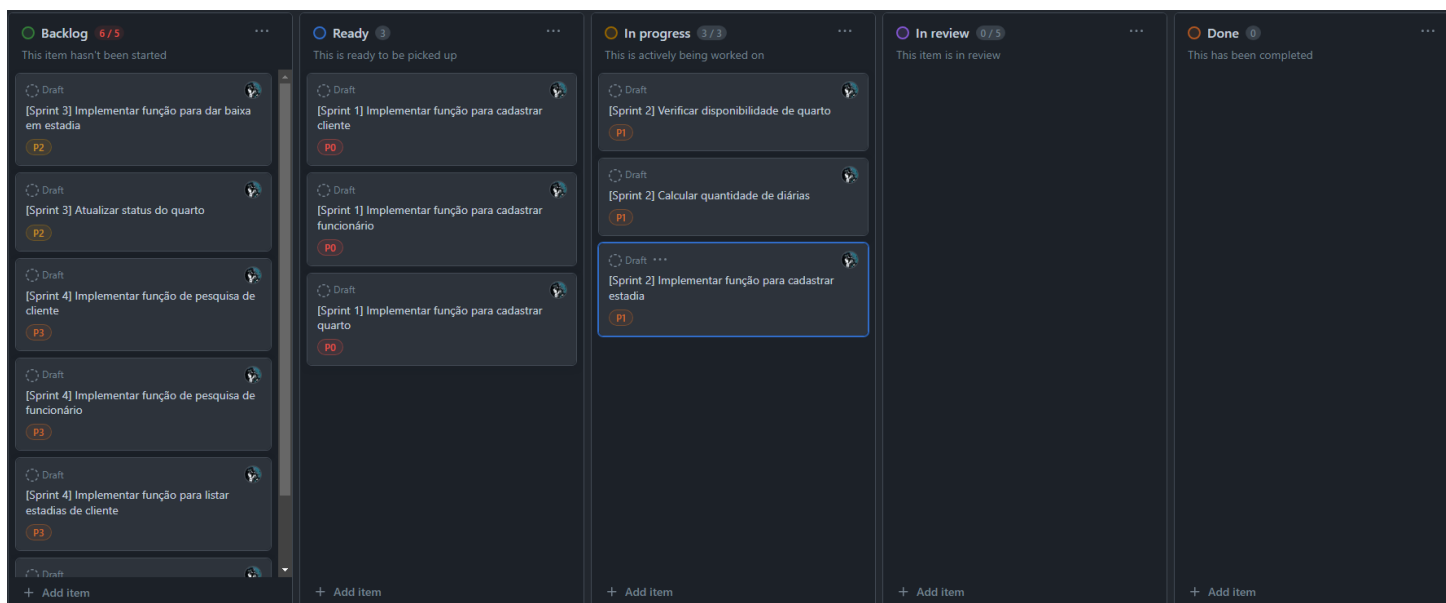
A Figura 1 apresenta o time criado no github projects com as divisões dos quadros. O quadro “Backlog geral” mostrará as divisões das funções entre os integrantes e quais tarefas deverão ser entregues por cada integrante por ordem de sprints. Os demais quadros são os *backlogs* de cada integrante, uma vez que cada um ficou responsável por uma função do sistema.

Figura 1 – Time e quadros criados no Github Projects



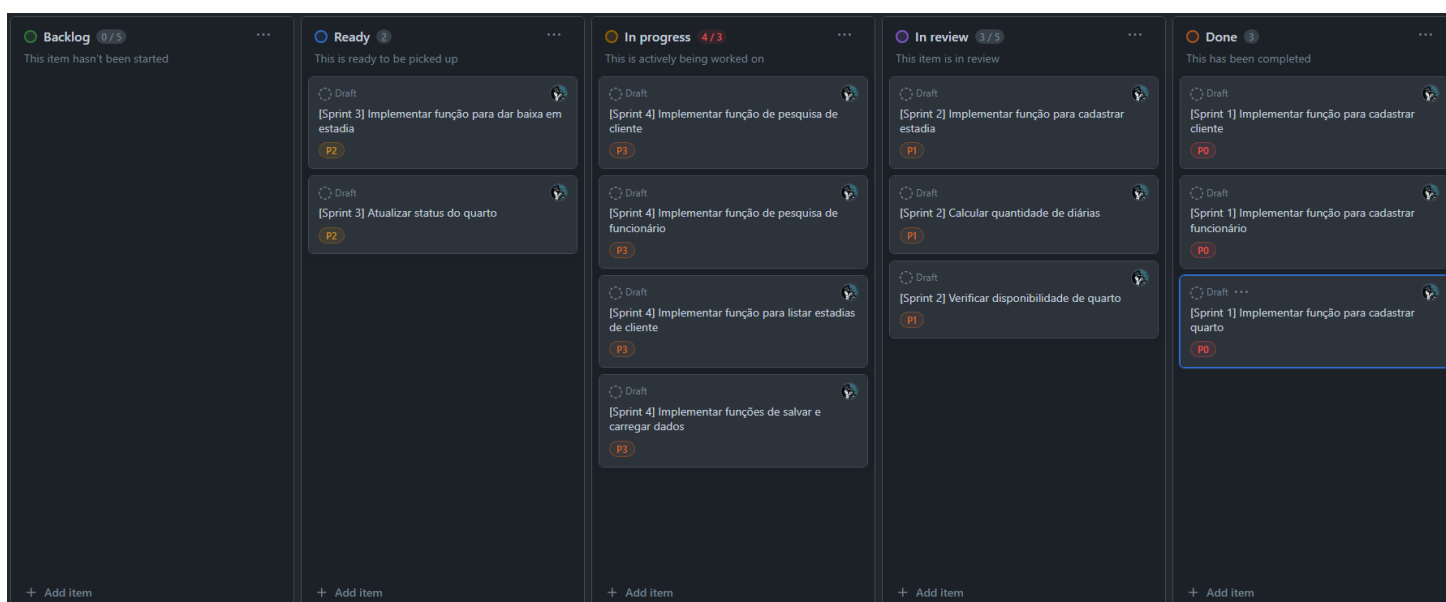
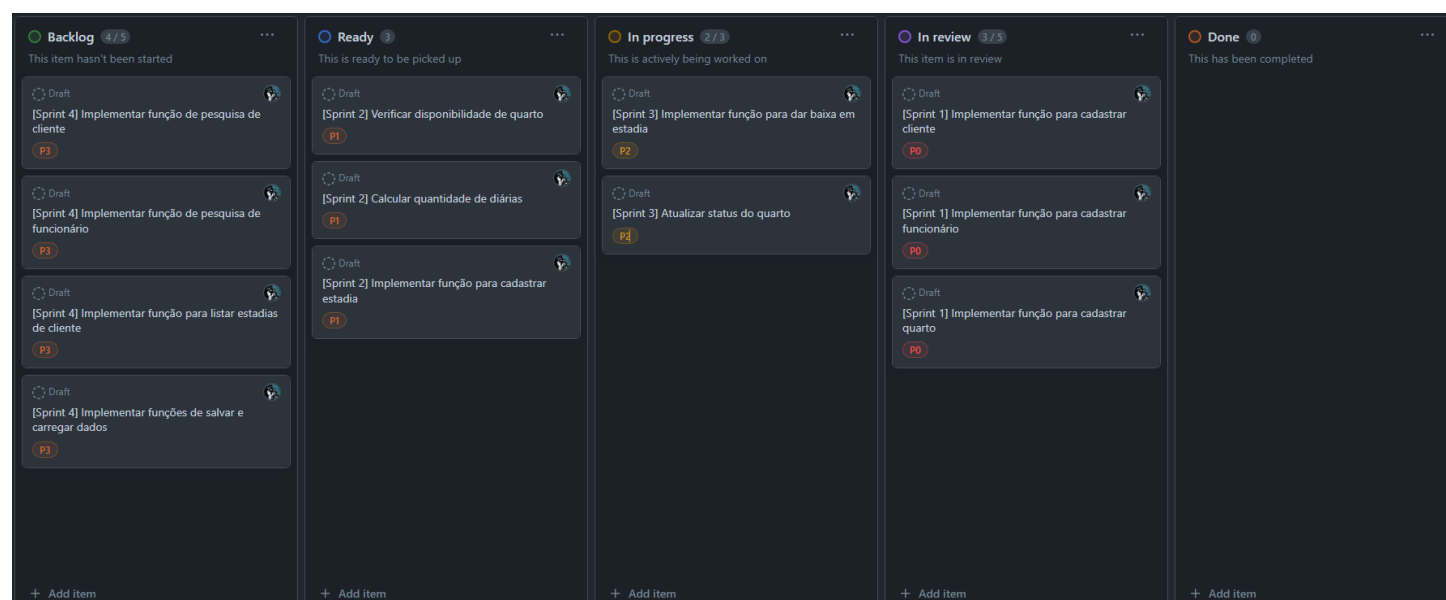
Fonte: Elaborado pelo autor

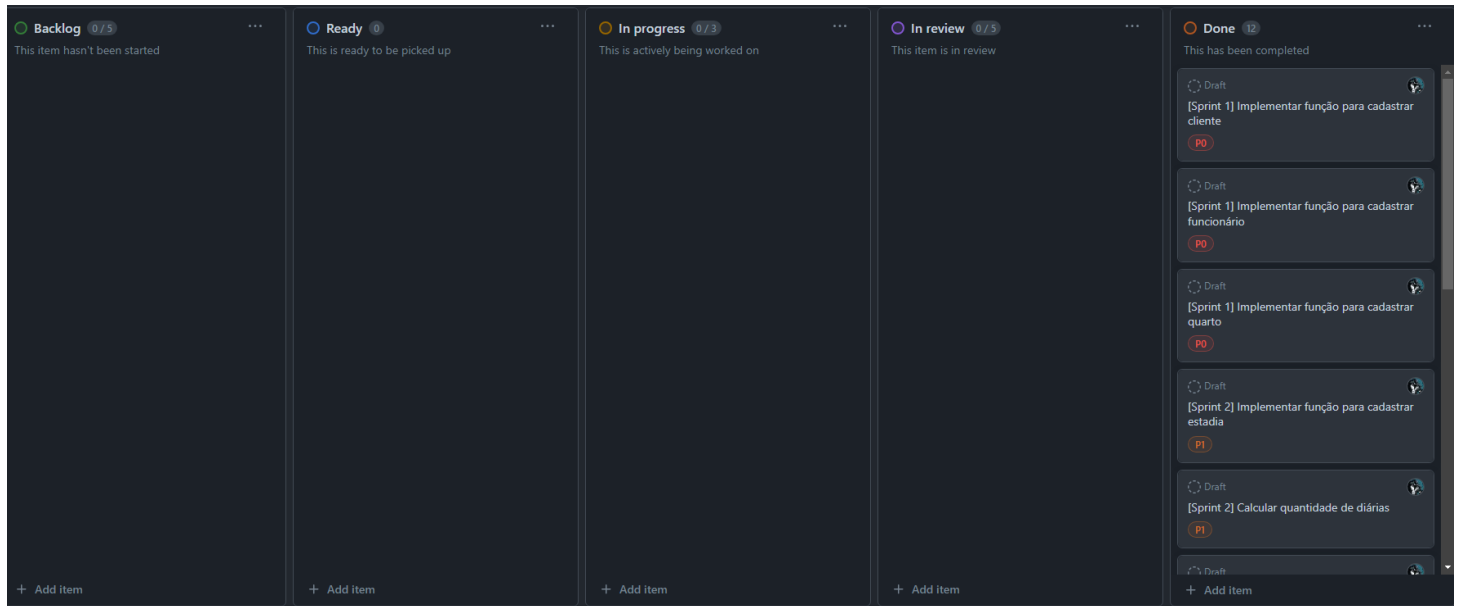
A Figura 2 apresenta o protótipo de tela realizado no início do projeto, destacando o backlog geral e a divisão de tarefas para cada sprint. Na sprint 2, as atividades atribuídas ao integrante Áulus Batista incluem a implementação da função para cadastrar estadia, a verificação da disponibilidade de quarto e o cálculo da quantidade de diárias. Essas tarefas são representadas na figura pela cor amarela, indicando que devem ser finalizadas durante esta etapa do desenvolvimento. O progresso dessas atividades é crucial para garantir o avanço do projeto de forma eficiente e dentro do prazo estabelecido



Fonte: Elaborado pelo autor

As Figuras 3, 4, e 5 demonstram o status do backlog do produto de cada integrante no final do trabalho





Fonte: Elaborado pelo autor

Lista de assinaturas das funções e parâmetros

Explicação da estrutura de dados principal do programa. Apresentação das assinaturas das funções

As funções e parâmetros utilizados no programa foram:

1. `void cadastrar_cliente()`

1-Verificação do limite de clientes: O início da função verifica se o número de clientes já atingiu o limite máximo definido (`MAX_CLIENTES`). Se o limite foi alcançado, a função exibe uma mensagem informando que o limite foi atingido e retorna, interrompendo o processo de cadastro.

2-Obtenção dos dados do cliente: Se o limite de clientes não foi atingido, a função prossegue criando uma instância da estrutura `Cliente`. Ela solicita ao usuário que digite o nome, endereço e telefone do cliente utilizando a função `scanf()`, armazenando esses dados na estrutura `cliente`. O código do cliente é atribuído automaticamente, sendo calculado como o número atual de clientes mais um.

3-Armazenamento dos dados do cliente: Após coletar os dados do cliente, a função armazena esses dados no array `clientes[]` na posição indicada pelo número atual de clientes (`num_clientes`). Em seguida, incrementa o contador de clientes (`num_clientes`) para refletir o novo cadastro.

4-Gravação dos dados em arquivo: A função abre o arquivo `"clientes.txt"` em modo de anexação ("`a`") utilizando a função `fopen()`. Se ocorrer algum erro ao abrir o arquivo, uma mensagem de erro é exibida e a função retorna. Caso contrário, os dados do cliente são escritos no arquivo utilizando a função `fprintf()`, com o formato `"código, nome, endereço, telefone"`. Por fim, o arquivo é fechado com a função `fclose()`.

5-Feedback ao usuário: Após completar todas as etapas com sucesso, a função exibe uma mensagem indicando que o cliente foi cadastrado com êxito.

2. void cadastrar_funcionario()

A função `cadastrar_funcionario()` adiciona um novo funcionário ao sistema, limitado pelo máximo permitido. Ela coleta e armazena informações sobre o nome, telefone, cargo e salário do funcionário. Em seguida, os detalhes são registrados em um arquivo "funcionarios.txt", e uma mensagem de sucesso é exibida ao usuário.

3. void cadastrar_estadia()

Essa função adiciona uma nova estadia ao sistema, limitada pelo máximo permitido. Ela solicita o código do cliente e verifica se o cliente está registrado. Em seguida, pede a quantidade de hóspedes e encontra um quarto disponível para eles. As datas de entrada e saída são inseridas e a quantidade de diárias é calculada. Os detalhes da estadia são registrados em um arquivo "estadias.txt", e uma mensagem de sucesso é exibida.

4. void dar_baixa_estadia()

Esta função permite dar baixa em uma estadia do sistema, solicitando o código da estadia. Em seguida, verifica se a estadia está registrada e, se encontrada, atualiza o status do quarto para "desocupado". Calcula o valor total a ser pago pela estadia com base na quantidade de diárias e no valor da diária do quarto. Essa função fornece o valor total a ser pago e uma mensagem de confirmação, ou informa se a estadia não foi encontrada.

5. void pesquisar_cliente()

Esta função permite pesquisar um cliente pelo código. Ela solicita o código do cliente e verifica se está registrado. Se o cliente for encontrado, exibe suas informações, incluindo código, nome, endereço e telefone. Se o cliente não for encontrado, exibe uma mensagem informando que o cliente não foi encontrado.

6. void pesquisar_funcionario()

Essa função permite pesquisar um funcionário pelo código. Solicita o código do funcionário e verifica se está registrado. Se o funcionário for encontrado, exibe suas informações, incluindo código, nome, telefone, cargo e salário. Se o funcionário não for encontrado, exibe uma mensagem informando que o funcionário não foi encontrado..

7. void mostrar_estadias_cliente()

Solicita o código do cliente ao usuário.

Exibe um cabeçalho indicando as estadias do cliente com o código fornecido.

Itera sobre todas as estadias registradas.

Para cada estadia, verifica se o código do cliente associado corresponde ao código fornecido.

Se uma estadia pertencer ao cliente, imprime as informações relevantes, incluindo o código da estadia, o número do quarto, as datas de entrada e saída, e a quantidade de diárias.

8. void carregar_dados()

A função `carregar_dados()` tem como objetivo ler os dados dos arquivos de texto correspondentes aos clientes, funcionários, estadias e quartos e armazená-los nas respectivas estruturas de dados. Aqui está uma descrição resumida do seu funcionamento:

Abre o arquivo "clientes.txt" para leitura e, se o arquivo existir, lê os dados do arquivo linha por linha usando `fscanf()`. Para cada linha lida, os dados são atribuídos à estrutura de cliente correspondente e o contador `num_clientes` é incrementado.

Repete o mesmo processo para os arquivos "funcionarios.txt", "estadias.txt" e "quartos.txt", lendo e armazenando os dados nas estruturas de funcionários, estadias e quartos, respectivamente, e incrementando os contadores num_funcionarios, num_estadias e num_quartos.

Essa função é essencial para carregar os dados previamente armazenados nos arquivos de texto quando o programa é iniciado, garantindo que as informações persistem entre diferentes execuções do sistema.

TESTES

Casos de teste do software:

Entradas	Classes Válidas	Resultado Esperado	Classes Inválidas	Resultado Esperado
Cliente: código do cliente	Cliente existente, número com um dígitos, exemplo 1, 2, 3, 4	Cliente encontrado	Cliente inexistente, número com mais de quatro dígitos, ou caracteres, exemplos 10005, 100c, números negativos, número com pontos e/ou vírgulas	Cliente não encontrado, é solicitado ao usuário que informe novamente o dado do cliente
Funcionario:codigo do funcionario	Funcionário existente, número com um dígitos, exemplo 1, 2, 3, 4	Funcionário encontrado	Funcionário inexistente, número com mais de quatro dígitos, ou caracteres, exemplos 10005, 100c, números negativos, número com pontos e/ou vírgulas	Funcionário não encontrado, é solicitado ao usuário que informe novamente o dado do cliente
Estadia:código de estadia	Estadia: data de entrada e saída	Quarto com estadia encontrado	Sem nenhuma data de entrada e nem de saída	Estadia nao encontrado