

Fatec Praia Grande

Hackathon Técnicas De Programação II

Desafio de Desenvolvimento - 2 Dias

Introdução

Bem-vindo ao Hackathon de Java! Neste evento de dois dias, desafiaremos vocês alunos do curso DSM3 Fatec Praia Grande a desenvolverem dois projetos em Java utilizando boas práticas de programação orientada a objetos, interface gráfica e persistência de dados. Os projetos propostos são:

1. **Sistema de Reservas de Passagens Aéreas**
2. **Sistema de Pedidos em Restaurante**

Requisitos

- Utilização de boas práticas de programação Orientado a Objetos em Java.
- Utilização de ArrayLists
- Interface gráfica para interação com o usuário.
- Persistência de dados para armazenar informações relevantes.

Entrega

Os participantes devem criar um repositório no GitHub contendo os códigos-fonte e uma documentação personalizada, explicativa sobre o projeto. Certifique-se de incluir instruções claras sobre como executar e testar os sistemas.

Boa sorte e que vençam os melhores desenvolvedores!

ENTREGA NO GITHUB ATÉ 20/02/24 ÀS 18H00.

Descrição dos Projetos

1. Sistema de Reservas de Passagens Aéreas

Regras de Negócio:

1. A reserva só pode ser feita se houver assentos disponíveis no voo.
2. O pagamento deve ser realizado após a reserva, sendo necessário escolher entre opções de pagamento (somente ida ou ida e volta, com taxa para pontos turísticos).
3. O número do voo deve ser único para cada voo.
4. A impressão da passagem deve conter todas as informações relevantes do voo.

Requisitos Funcionais:

1. Cadastro de Voos:

- O sistema deve permitir o cadastro de voos com número, origem, destino e quantidade de assentos disponíveis.

2. Reserva de Assentos:

- Os usuários devem ser capazes de realizar reservas de assentos em voos específicos.

3. Pagamento:

- Após a reserva, os usuários devem escolher o tipo de pagamento (somente ida ou ida e volta, com taxa para pontos turísticos).

4. Impressão de Passagem:

- O sistema deve possibilitar a impressão da passagem contendo todas as informações do voo.

Casos de Uso:

1. Cadastrar Voo:

- Ator: Administrador
- Fluxo:
 - O administrador insere os detalhes do voo.
 - O sistema valida e armazena as informações.

2. Realizar Reserva:

- Ator: Cliente
- Fluxo:
 - O cliente escolhe o voo desejado.
 - O sistema verifica a disponibilidade.

- Se houver assentos disponíveis, a reserva é realizada.

3. Realizar Pagamento:

- Ator: Cliente
- Fluxo:
 - Após a reserva, o cliente escolhe a forma de pagamento.
 - O sistema aplica as regras de pagamento.

4. Imprimir Passagem:

- Ator: Cliente
- Fluxo:
 - O cliente solicita a impressão da passagem.
 - O sistema gera a passagem com todas as informações do voo.

Exemplo Classe Voo:

- Atributos:
 - numeroVoo (String): número do voo.
 - origem (String): cidade de origem.
 - destino (String): cidade de destino.
 - assentosDisponiveis (int): número de assentos disponíveis.
- Métodos:
 - realizarReserva(int quantidadeAssentos): realiza a reserva de uma determinada quantidade de assentos.
 - verificarDisponibilidade(int quantidadeAssentos): verifica se há assentos disponíveis.
 - realizarPagamento(): aplica regras de pagamento (somente ida ou ida e volta, com taxa para pontos turísticos).
 - imprimirPassagem(): imprime a passagem.

Exemplo Classe Aeroporto

- Utiliza a classe Voo para armazenar e gerenciar os voos disponíveis.

2. Sistema de Pedidos em Restaurante

Regras de Negócio:

1. O pedido só pode ser calculado se houver pelo menos um item no pedido.
2. A remoção de itens só é permitida se houver mais de um item no pedido.
3. A reserva de mesa só pode ser feita para pedidos com itens.

Requisitos Funcionais:

1. Cadastro de Pratos:

- O sistema deve permitir o cadastro de pratos com nome, preço e detalhes.

2. Realizar Pedido:

- Os clientes devem ser capazes de criar pedidos, adicionando itens ao pedido.

3. Calcular Total do Pedido:

- O sistema deve calcular o valor total do pedido, incluindo o preço de todos os itens e a taxa de entrega.

4. Adicionar e Remover Itens:

- Os clientes podem adicionar e remover itens do pedido.

5. Reservar Mesa:

- Os clientes podem reservar uma mesa ao fazer um pedido.

Casos de Uso:

1. Cadastrar Prato:

- Ator: Administrador
- Fluxo:
 - O administrador insere os detalhes do prato.
 - O sistema valida e armazena as informações.

2. Fazer Pedido:

- Ator: Cliente
- Fluxo:
 - O cliente seleciona os pratos desejados.
 - O sistema cria um pedido associado ao cliente.

3. Calcular Total do Pedido:

- Ator: Cliente
- Fluxo:

- O cliente solicita o cálculo do valor total do pedido.
- O sistema realiza o cálculo considerando itens e taxa de entrega.

4. Adicionar Item ao Pedido:

- Ator: Cliente
- Fluxo:
 - O cliente adiciona um item ao pedido.
 - O sistema atualiza o pedido.

5. Remover Item do Pedido:

- Ator: Cliente
- Fluxo:
 - O cliente remove um item do pedido.
 - O sistema atualiza o pedido se houver mais de um item.

6. Reservar Mesa:

- Ator: Cliente
- Fluxo:
 - O cliente reserva uma mesa ao fazer o pedido.
 - O sistema associa a reserva ao pedido realizado.

Exemplo Classe Pedido

- Atributos:
 - numero (int): número do pedido.
 - itens (ArrayList<ItemPedido>): lista de itens no pedido, onde ItemPedido possui informações como nome do prato, quantidade e preço unitário.
- Métodos:
 - calcularTotalPedido(): double: calcula o valor total do pedido somando os preços de todos os itens e a taxa de entrega.
 - adicionarItem(ItemPedido item): adiciona um item ao pedido.
 - removerItem(): remove um item do pedido.
 - reservarMesa(): reserva uma mesa.

Exemplo Classe Restaurante

- Utiliza a classe Pedido para armazenar e gerenciar os pedidos realizados.