



MINISTÉRIO DA EDUCAÇÃO – MEC UNIVERSIDADE
FEDERAL RURAL DO SEMI-ÁRIDO – UFERSA
CAMPUS MULTIDISCIPLINAR PAU DOS FERROS – CMPF
CURSO: INTERDISCIPLINAR EM TECNOLOGIA DA
INFORMAÇÃO – TI
COMPONENTE: TESTE DE SOFTWARE
DOCENTE: ALYSSON FILGUEIRA MILANEZ

ANAXÁGORAS LUCAS DE ARAUJO MEDEIROS
NEILLA MARIA DE SOUZA BEZERRA
RUAN PABLO BARBOSA CLAUDINO

RELATÓRIO- PROJETO FINAL

Trabalho desenvolvido no Curso de
Tecnologia da informação da
Universidade Federal Rural do
Semi-Árido, como requisito para
obtenção da nota da 3º unidade da
disciplina de Teste de Software.

1. OBJETIVO PRINCIPAL:

O projeto teve como principal objetivo o desenvolvimento de um sistema de gerenciamento de reservas de hotel, com foco no controle e organização das operações de cadastro de usuário, realização e/ou gerenciamento de reservas, bem como de pagamentos, check-in/check-out, entre outros. Nesse contexto, os testes foram automatizados utilizando o JUnit como ferramenta de teste.

2.ATIVIDADES DE TESTE IMPLEMENTADAS

Teste de cadastro de quarto: CAIXA BRANCA

O mesmo deverá verificar se o sistema irá cadastrar corretamente um novo quarto. Para isso, são fornecidos dados válidos para o número do quarto, comodidades, capacidade, tarifa, disponibilidade e data de disponibilidade. Após o cadastro, é verificado se as informações do quarto foram armazenadas corretamente no sistema.

Teste de exclusão de quarto: CAIXA PRETA

O mesmo avalia a capacidade do sistema de excluir um quarto existente. Primeiramente, um quarto válido é criado e inserido no sistema. Em seguida, é realizada a exclusão do quarto com base no número do quarto. Após a exclusão, o teste verifica se o quarto foi removido corretamente do sistema, verificando se a lista de quartos está vazia.

Teste de cadastro de usuário: CAIXA PRETA

O mesmo verifica se o sistema irá cadastrar corretamente um novo usuário. Primeiramente, é criado um objeto de usuário com informações válidas, como nome, e-mail, senha, telefone e endereço. Após o usuário concluir o seu cadastro, será realizado um teste de visualização do usuário recém-criado, utilizando o nome como parâmetro de busca.

Teste de registro de pagamento: CAIXA PRETA

O mesmo verifica se o sistema é capaz de registrar um novo pagamento corretamente, de forma que os dados de entrada para um novo pagamento sejam simulados, sendo eles: o valor e o método de pagamento. O teste irá verificar que o método de registro de pagamento não lançará nenhuma exceção, indicando que o pagamento foi registrado com sucesso.

Teste de visualização de pagamento: CAIXA PRETA E CAIXA BRANCA

O mesmo verifica se o sistema é capaz de visualizar um pagamento existente. No mais, irá assegurar que nenhuma exceção é lançada durante a visualização do pagamento com um ID válido, e se uma IOException é lançada ao tentar visualizar um pagamento com um ID inválido.

Teste de listagem de pagamentos: CAIXA PRETA E CAIXA BRANCA

O mesmo irá verificar se o sistema será capaz de listar todos os pagamentos registrados. Inicialmente, alguns pagamentos fictícios serão inseridos no sistema. Em seguida, o método de listagem de pagamentos é testado para garantir que a lista de pagamentos não seja nula e que contenha a quantidade correta de pagamentos. Além disso, verifica se os detalhes de cada pagamento na lista correspondem aos valores esperados.

Teste de edição de pagamento: CAIXA PRETA

O mesmo irá verificar se o sistema lança uma exceção ao tentar editar um pagamento. Com isso, novos valores serão simulados para edição de um pagamento existente. O teste verifica se uma IOException é lançada, indicando que a edição de pagamento não é permitida.

Teste de Exclusão de Pagamento: CAIXA PRETA

O mesmo irá verificar se o sistema lança uma exceção ao tentar excluir um pagamento com um ID inválido. O teste verifica se uma `IOException` é lançada, indicando que a exclusão de pagamento não é permitida para IDs inválidos.

Teste de verificação de disponibilidade de quarto: CAIXA PRETA

O mesmo tem como o objetivo verificar se o sistema está corretamente identificando a disponibilidade de um quarto. Com isso, o método de verificação de disponibilidade é testado com o número de um quarto especificado. O teste verifica se o método retorna `false` quando o quarto especificado não existe no arquivo de quartos, o que significa que o quarto não está disponível para reserva.

Teste de criação de reserva: CAIXA PRETA

O mesmo irá verificar se o sistema está criando corretamente uma reserva. Com isso, é fornecida uma entrada simulada contendo as datas de início e fim da reserva. O teste verifica se a reserva foi criada com sucesso.

Teste de visualização de reserva: CAIXA PRETA

O mesmo irá verificar se o sistema está visualizando corretamente uma reserva. Dessa forma, o método `visualizarReserva()` é testado com o ID de uma reserva específica. O teste verifica se a mensagem "Reserva não encontrada." é retornada quando a reserva não é encontrada.

Teste de check-in: CAIXA PRETA

O mesmo irá verificar a eficácia do sistema quanto à correta execução da operação de check-in. Dessa forma, o método `checkin()` é chamado com o ID de uma reserva específica e a data de check-in. O teste verifica se a data de check-in foi atualizada corretamente.

Teste de check-out: CAIXA PRETA

O mesmo irá verificar a eficácia do sistema quanto à correta execução da operação de check-out. Dessa forma, o método `checkout()` é chamado com o ID de uma reserva específica e a data de check-out. O teste verifica se a data de check-out foi atualizada corretamente.

Teste de cancelamento de reserva:

O mesmo irá verificar se o procedimento de cancelamento de reserva será efetuado com êxito. Dessa forma, o método `cancelarReserva()` é chamado com o ID de uma reserva específica. O teste verifica se a reserva foi cancelada corretamente usando o método `verificaReserva()`.

Teste de criação de quarto: CAIXA PRETA

O mesmo irá verificar se o sistema está criando um quarto corretamente. Uma entrada simulada é fornecida contendo os detalhes do quarto, como número, comodidades, capacidade, tarifa, disponibilidade, entre outros. Com isso, o método `inserirQuarto()` é chamado com as entradas fornecidas. O teste então verifica se o quarto foi criado com sucesso usando o método `listarQuartos()`.

Teste de falha ao criar quarto com número existente: CAIXA PRETA

O mesmo contará com uma entrada simulada, contendo os detalhes de um quarto com um número que já existe no sistema. Com isso, o método `inserirQuarto()` é chamado com as entradas fornecidas. O teste verifica se o quarto não foi criado com sucesso usando o método `listarQuartos()`.

Teste de Falha ao Criar Quarto com Data Inválida: CAIXA PRETA

O mesmo irá verificar se o sistema está corretamente lidando com a inserção de um quarto com uma data inválida. Nesse caso, uma entrada simulada é fornecida contendo os detalhes do quarto, incluindo uma data inválida. Com isso, o método ``inserirQuarto()`` é chamado com as entradas fornecidas. O teste verifica se o quarto não foi criado com sucesso usando o método ``listarQuartos()``.

Teste de autenticação do administrador: CAIXA PRETA

O mesmo irá verificar se o sistema está corretamente autenticando o administrador. Uma entrada simulada contendo o nome de usuário e a senha do administrador é fornecida. Com isso, o método ``autenticarAdmin()`` é chamado com as entradas fornecidas. O teste verifica se a autenticação é bem-sucedida, retornando *true*.

Teste de autenticação do usuário: CAIXA PRETA

O mesmo irá verificar se o sistema está corretamente autenticando um usuário comum. Uma entrada simulada contendo o nome de usuário e a senha do usuário é fornecida. Dessa forma, o método ``autenticarUser()`` é chamado com as entradas fornecidas. O teste verifica se a autenticação é bem-sucedida, retornando *true*.

Teste de falha na autenticação do administrador: CAIXA PRETA

O mesmo irá verificar se o sistema está corretamente lidando com a tentativa de autenticação do administrador com uma senha incorreta. Uma entrada simulada contendo o nome de usuário do administrador e uma senha incorreta é fornecida. Com isso, o método ``autenticarAdmin()`` é chamado com as entradas fornecidas. O teste verifica se a autenticação falha, retornando *false*.

Teste de falha na autenticação do usuário: CAIXA PRETA

O mesmo irá verificar se o sistema está corretamente lidando com a tentativa de autenticação de um usuário comum com uma senha incorreta. Uma entrada simulada contendo o nome do usuário comum e uma senha incorreta é fornecida. Com isso, o método ``autenticarUser()`` é chamado com as entradas fornecidas. O teste verifica se a autenticação falha, retornando *false*.