

Relatório de Testes

Contextualização:

A aplicação foi desenvolvida inicialmente com foco na funcionalidade, sem a implementação de validações para entradas de dados. Por esse motivo, os testes realizados priorizam cenários simples, com dados corretos, apenas para garantir o funcionamento básico de cada função.

Módulo de Login

Objetivo: Testar as principais funcionalidades de autenticação da aplicação: exibição da tela de login, autenticação de cliente e despachante, falha de login com credenciais inválidas e logout.

1. Rota: /login/ (GET)

Função testada: tela_login

Descrição: Verifica se a página de login é carregada corretamente.

Resultado esperado:

- Status HTTP 200
- Contém a palavra "Login" na resposta.

2. Rota: /login/login (POST) - Login Cliente

Função testada: login

Descrição: Testa o login bem-sucedido de um cliente, com criação de sessão e redirecionamento.

Resultado esperado:

- Redirecionamento (HTTP 302) para /cliente/encontrarServicos.
- Sessão contém cliente_id e cliente_nome.

3. Rota: /login/login (POST) - Login Despachante

Função testada: login

Descrição: Testa o login bem-sucedido de um despachante, com criação de sessão e redirecionamento.

Resultado esperado:

- Redirecionamento (HTTP 302) para /despachante/servicos.
- Sessão contém desp_id e desp_nome.

4. Rota: /login/login (POST) - Falha de Login

Função testada: login

Descrição: Testa o login com credenciais inválidas, sem criação de sessão.

Resultado esperado:

- Resposta HTTP 200 (página recarregada).
- Mensagem "Email ou senha incorretos." exibida.

5. Rota: /login/logout (GET)

Função testada: logout

Descrição: Testa a funcionalidade de logout, removendo dados da sessão.

Resultado esperado:

- Resposta HTTP 200.
- Mensagem "Você saiu da sua conta." exibida.
- Sessão limpa.

Resumo Final: Todos os testes executados obtiveram sucesso, garantindo o funcionamento básico do fluxo de login, autenticação e logout para clientes e despachantes.

Módulo Tela Inicial

Objetivo: Garantir que as principais rotas da tela inicial da aplicação carreguem corretamente, verificando status HTTP e conteúdos esperados.

1. Rota: / (GET)

Função testada: tela_inicial

Descrição: Testa se a página inicial da aplicação é carregada com sucesso.

Resultado esperado:

- Status HTTP 200.

2. Rota: /cliente_desp (GET)

Função testada: cad_cliente_desp

Descrição: Testa o carregamento da página de escolha de cadastro (cliente ou despachante).

Resultado esperado:

- Status HTTP 200.
- Presença das palavras "Cliente" e "Despachante" no conteúdo da página.

Resumo Final: Todos os testes executados foram bem-sucedidos, garantindo que a tela inicial e a página de escolha de cadastro estão acessíveis e retornem os conteúdos esperados.

Módulo Cadastro de Cliente

Objetivo: Garantir que a funcionalidade de cadastro de cliente realiza a inserção no banco de dados corretamente e redireciona para a tela inicial.

1. Rota: /cadastroCliente/cadastro-cliente (POST)

Função testada: cadastrar cliente

Descrição: Testa se, ao enviar os dados corretos de um cliente, a aplicação:

- Insere os dados na tabela cliente.
- Redireciona o usuário para a tela inicial (/).
- Registra a mensagem de sucesso na sessão.

Resultado esperado:

- Redirecionamento (HTTP 302) para /.
- Mensagem de sucesso registrada na sessão (Cadastro realizado com sucesso!).
- Execução do INSERT INTO cliente....
- Commit da transação e fechamento da conexão.

Resumo Final: O teste confirmou que a funcionalidade de cadastro de cliente está operando corretamente, realizando as operações no banco de dados e efetuando o direcionamento adequado.

Módulo Cadastro de Despachante

Objetivo: Garantir que a funcionalidade de cadastro de despachante realiza a inserção no banco de dados corretamente e redireciona para a tela inicial.

1. Rota: /cadastroDespachante/cad-desp (POST)

Função testada: cadastrar_desp

Descrição: Testa se, ao enviar os dados corretos de um despachante e de seu estabelecimento, a aplicação:

- Insere os dados nas tabelas correspondentes.
- Redireciona o usuário para a tela inicial (/).
- Registra a mensagem de sucesso na sessão.

Resultado esperado:

- Redirecionamento (HTTP 302) para /.
- Mensagem de sucesso registrada na sessão (Despachante cadastrado com sucesso!).
- Execução das instruções de inserção no banco.

- Commit da transação e fechamento da conexão.

Resumo Final: O teste confirmou que a funcionalidade de cadastro de despachante está funcionando conforme o esperado, realizando as operações necessárias no banco de dados e efetuando o redirecionamento adequado.

Módulo Cliente

Objetivo: Garantir que as funcionalidades relacionadas ao cliente (visualização e atualização de perfil, visualização de chamados e busca por serviços) estão funcionando corretamente e manipulando adequadamente o banco de dados.

1. Rota: /cliente/encontrarServicos (GET)

Função testada: `encon_desp_serv_get_logado`

Descrição: Testa se, com o cliente logado:

- A página de busca por despachantes e serviços é carregada.
- O banco de dados é consultado corretamente.

Resultado esperado:

- Status HTTP 200 (OK).
- Chamada de execução SQL realizada.

2. Rota: /cliente/chamadoCliente (GET)

Função testada: `chamado_cliente_get_logado`

Descrição: Testa se, com o cliente logado:

- A página de visualização dos chamados do cliente carrega normalmente.

Resultado esperado:

- Status HTTP 200 (OK).

3. Rota: /cliente/perfilCliente (GET)

Função testada: `perfil_cliente_get_logado`

Descrição: Testa se, com o cliente logado:

- A página de perfil do cliente é carregada.
- Os dados pessoais e de endereço são buscados corretamente do banco.

Resultado esperado:

- Status HTTP 200 (OK).
- Os dados do cliente e do endereço estão presentes na resposta (Nome, Teste, endereço).
- Execução das queries de seleção para cliente e endereço.

4. Rota: /cliente/perfilCliente (POST)

Função testada: atualizar_perfil_cliente_post_logado_sucesso

Descrição: Testa se, ao submeter dados válidos para atualização:

- O perfil do cliente é atualizado com sucesso no banco.
- A transação é confirmada.
- A resposta inclui mensagem de sucesso.

Resultado esperado:

- Status HTTP 200 (OK).
- Mensagem: Perfil atualizado com sucesso!.
commit realizado na conexão com o banco.

Resumo Final: Todos os testes do módulo **Cliente** foram bem-sucedidos, garantindo que as principais operações de consulta e atualização de dados estão funcionando conforme o esperado, com proteção de sessão e integração correta com o banco de dados.

Módulo Despachante

Objetivo: Garantir que as funcionalidades relacionadas ao despachante (visualização e atualização de perfil, visualização de chamados, criação de serviços, permitir editar e excluir serviços) estão funcionando corretamente e manipulando adequadamente o banco de dados.

1. Rota: /despachante/servicos (GET)

Função testada: test_servico_desp_ok

Descrição: Exibe a lista de serviços cadastrados pelo despachante.

Testes realizados:

- Simula retorno de serviços mockados (CNH e Placa).
- Verifica status HTTP 200.
- Confirma a presença de conteúdo esperado no HTML (serviços listados).
- Assegura que consultas SQL foram executadas e recursos de banco de dados fechados.

2. Rota: /despachante/criarNovoServico (GET e POST)

Função testada: test_criar_servico_get_ok

Descrição:

- GET: Exibe formulário para criação de novo serviço.
- POST: Realiza inserção de novo serviço e documentos associados.

Testes realizados:

GET:

- Verifica o carregamento correto da página com status 200.
- Confirma que conexão com banco de dados não foi necessária.

POST:

- Simula inserção de novo serviço e documentos.
- Verifica redirecionamento para /despachante/servicos (status 302).
- Confirma a presença de mensagem de sucesso (Serviço e documentos criados com sucesso!).
- Assegura execução de comandos SQL e fechamento de recursos.

3. Rota: /despachante/deletarServico (POST)

Função testada: test_deletar_servico_post_ok

Descrição: Exclui um serviço e seus documentos associados.

Testes realizados:

- Simula exclusão com serviço mockado.
Verifica redirecionamento para /despachante/servicos (status 302).
- Confirma presença de mensagem de sucesso (Serviço excluído com sucesso.).
Garante execução de comandos de banco e fechamento dos recursos.

4. Rota: /despachante/editarServico (POST)

Função testada: test_editar_servico_get_ok

Descrição: Exibe página de edição de um serviço existente, carregando dados do serviço e documentos.

Testes realizados:

- Simula retorno de dados de serviço e documentos.
- Confirma o carregamento correto da página com status 200.
- Verifica a presença dos dados mockados no HTML.
- Garante que comandos SQL foram executados e recursos fechados.

5. Rota: /despachante/atualizarServico (POST)

Função testada: test_atualizar_servico_post_ok

Descrição: Atualizar os dados de um serviço e seus documentos.

Testes realizados:

- Simula atualização de serviço e documentos.
- Verifica redirecionamento para /despachante/servicos (status 302).
- Confirma presença de mensagem de sucesso (Serviço atualizado com sucesso!).
- Assegura execução e commit de comandos SQL e fechamento de recursos.

6. Rota: /despachante/chamados (GET)

Função testada: test_chamado_desp_get_ok

Descrição: Exibe a lista de chamados do despachante.

Testes realizados:

- Verifica carregamento da página com status 200.
- Confirma que nenhuma conexão com banco foi necessária.

7. Rota: /despachante/agendamentos (GET)

Função testada: test_agenda_desp_get_ok

Descrição: Exibe a agenda de agendamentos do despachante.

Testes realizados:

- Confirma o carregamento da página com status 200.
- Verifica que não houve interação com banco de dados.

8. Rota: /despachante/perfil (GET)

Função testada: test_perfil_desp_get_ok

Descrição: Exibe dados do perfil do despachante e seu estabelecimento.

Testes realizados:

- Simula retorno de dados completos do despachante e estabelecimento.
- Confirma status HTTP 200.
- Verificar a presença das informações corretas na resposta.
- Assegura a execução dos comandos SQL e fechamento adequado de recursos.

Conclusão:

Os testes desenvolvidos nesta etapa do projeto tiveram como foco a criação de cenários mais simples, com o objetivo principal de validar o funcionamento das funcionalidades implementadas até o momento. Priorizamos verificar aspectos essenciais como o correto redirecionamento entre páginas e o comportamento esperado das funcionalidades disponíveis, mesmo que nem todas as telas tenham sido finalizadas.

Esta abordagem inicial permitiu estabelecer uma base sólida para a evolução dos testes, garantindo que os fluxos principais do sistema estejam funcionando conforme o esperado.

Futuramente, planejamos ampliar a cobertura dos testes, incluindo validações mais complexas, como a verificação de vulnerabilidades relacionadas a SQL Injection e outros problemas de segurança. Além disso, há a intenção de implementar testes de ponta a ponta

(E2E) utilizando ferramentas como o Selenium, de modo a simular interações completas do usuário e validar o sistema de forma mais abrangente.

Assim, este trabalho representa o início de uma cultura de testes contínua, que será progressivamente aprimorada conforme o desenvolvimento do projeto avança.

Modificações:

Link do repositório remoto no GitHub:

<https://github.com/theraissa/projeto-sistem-web-desp-transito.git>

Link do DrawSQLI com os diagramas do banco de dados:

<https://drawsql.app/teams/raissa-3/diagrams/sistema-web-para-despachantes-de-transito>

Contextualização:

Além dos testes que foram adicionados, a nossa aplicação teve um crescimento, pois foi criado rotas que além de redirecionar para os caminhos corretos, também é possível criar serviço como despachante, deletar e editar. É possível visualizar as informações do perfil e atualizar, tanto como cliente ou despachante. O cliente consegue filtrar por nome e cidade, dessa forma encontrando os despachantes disponíveis no sistema. E ambos conseguem se cadastrar e logar.

Com tudo, ainda falta muito a ser feito, porque não foi aprimorado em relação a segurança e autenticação, falta implementar algumas funcionalidades e realizar o gerenciamento do documentos.