**Documentação de Testes CRUD - Classes Denúncia, Usuário e Contato**

#### **1. Objetivo do Teste**

* Validar a implementação das operações CRUD nas classes **Denúncia**, **Usuário** e **Contato**, garantindo o correto funcionamento dos métodos e a integridade dos dados, focando no acesso e operações a nível de administrador.

#### **2. Ambiente de Teste**

* **Ambiente**: Desenvolvimento.
* **Banco de Dados**: SQL Server.
* **Framework**: ASP.NET Core com Razor Pages.
* **Data:** 13/11/2024 às 16:30.

#### **3. Casos de Teste para Operações CRUD**

##### **3.1. Classe Denúncia**

* **Create**: Inserção de nova denúncia com dados válidos. O formulário foi enviado e os dados foram corretamente inseridos no banco sem gerar exceções.
* **Read**: As denúncias foram listadas e exibidas com sucesso. A pesquisa e paginação funcionaram conforme o esperado.
* **Update**: Modificações em uma denúncia existente foram salvas corretamente e os dados atualizados foram refletidos na interface.
* **Delete**: O soft delete (marca de status "inativo") foi executado corretamente, mantendo o registro no banco mas ocultando-o da listagem ativa.

##### **3.2. Classe Contato**

* **Create**: O formulário de criação foi preenchido e enviado com sucesso, sem gerar erros.
* **Read**: A listagem de contatos e a funcionalidade de busca retornaram os resultados corretamente.
* **Update**: A edição de informações de um contato já existente foi realizada sem problemas.
* **Delete**: Contatos foram marcados como inativos conforme o esperado.

##### **3.3. Classe Usuário**

* **Create**: Ao tentar criar um novo usuário, a página recarregava sem lançar uma exceção, mas sem criar o usuário. Após análise, o campo de senha estava ausente na Create.cshtml. **Correção**: A inclusão do campo de senha no formulário solucionou o problema.
* **Read**: A listagem de usuários foi realizada corretamente, com a exibição de informações válidas.
* **Update**: Ao tentar editar um usuário existente, a aplicação lançou a exceção System.NullReferenceException: 'Object reference not set to an instance of an object.' Microsoft.AspNetCore.Mvc.Razor.RazorPage<TModel>.Model.get retornou null.. **Correção**: Após adicionar o campo de senha ausente na view Edit.cshtml, a operação de edição funcionou como esperado.
* **Delete**: Exclusão de usuários funcionou, com o campo Ativo atualizado conforme esperado para marcar usuários como inativos.

#### **4. Erros Identificados e Resoluções Aplicadas**

|  |  |  |  |
| --- | --- | --- | --- |
| **Classe** | **Operação** | **Erro** | **Resolução** |
| Usuário | Create | Página recarregava sem criar o usuário após enviar o formulário. | Adicionado o campo de senha na Create.cshtml. |
| Usuário | Update | Exceção System.NullReferenceException ao enviar o formulário de edição (Edit.cshtml). | Adicionado o campo de senha ausente na view Edit.cshtml. |

#### **5. Considerações Finais**

* As operações de **Create**, **Read**, **Update** e **Delete** para **Denúncias** e **Contatos** estão funcionando conforme esperado.
* Para **Usuário**, os problemas observados foram solucionados ao garantir que todos os campos necessários estivessem presentes nas views de criação e edição.
* Esses testes mostraram que campos obrigatórios, como a senha do usuário, precisam estar sempre definidos e corretamente vinculados na view para evitar exceções e falhas silenciosas, como o recarregamento da página sem erro explícito.