UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO – UFERSA
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS – CMPF
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

TÍTULO DO PROJETO: SIMULT-PDF 2.0

NOME DOS ALUNOS: GEÍSA MORAIS GABRIEL - 2021010372
LEONARDO INÁCIO GUILHERME DANTAS - 2021010757
LÍVIA BEATRIZ MAIA DE LIMA - 2021010871
NOME DO PROFESSOR: ALYSSON FILGUEIRA MILANEZ

Definição do projeto

O SIMULT-PDF é um sistema de monitoramento/cadastramento de multas, desenvolvido inicialmente na linguagem Portugol, com o intuito de aprimorar a lógica de programação. O sistema elaborado na disciplina de Laboratório de Algoritmos e Estrutura de Dados II é simples e consiste na construção de um menu principal para auxiliar nas tomadas de decisões.

Embasado nisso, o SIMULT-PDF 2.0 é resultado do aprimoramento do sistema anterior, visando mais especificamente a parte do administrador. Dessa forma, são acrescidas novas funcionalidades - além da básica geração de multas e relatórios de trânsito - ao novo programa implementado na linguagem de programação Java.

Links

- https://github.com/liviabeatrizml/SIMULT-PDF.v2
- https://www.overleaf.com/read/krgmdddvmjwb

Requisitos

- Funcionais

  [RF001] Criar novos usuário com permissão de administrador -
  - Descrição: O sistema deve permitir a criação de novos usuários administradores para a utilização do sistema.

  [RF002] Administrador pode entrar no sistema -
  - Descrição: O sistema deve permitir que os administradores cadastrados possam entrar no sistema com suas credenciais.

  [RF003] Permitir que o administrador buscar por veículos -
  - Descrição: O sistema deve permitir que o administrador possa realizar a busca de veículos cadastrados no sistema;

  [RF004] O administrador pode visualizar os dados veículos -

- Descrição: O sistema deve permitir que o administrador possa visualizar as informações dos veículos cadastrados no sistema;

[RF005] Permitir que o administrador realizar busca por multas -
- Descrição: O sistema deve permitir que o administrador possa pesquisar por multas cadastrados no sistema;

[RF006] Permitir que o administrador visualize as informações de uma multa -
- Descrição: O sistema deve permitir que o administrador possa visualizar as multas cadastrados no sistema;

[RF007] O administrador pode vincular multas a veículos -
- Descrição: O sistema deve permitir que o administrador possa vincular multas a veículos cadastrados no sistema;

[RF008] Permitir que o administrador excluir multas em veículos -
- Descrição: O sistema deve permitir que o administrador possa desvincular multas a veículos cadastrados no sistema;

[RF009] O administrador pode gerar relatório de multa vinculados no dia -
- Descrição: O sistema deve permitir que o administrador possa gerar um relatório contendo todas as multas vinculadas a veículos por ele;

[RF010] O administrador pode gerar relatório com o histórico de todas as suas ações -
- Descrição: O sistema deve permitir que o administrador possa gerar um relatório contendo todas as ações tomadas pelo mesmo;

- Não Funcionais

[RNF001] Deve haver criptografia da senha dos usuários.
- Descrição: O sistema deve criptografar as senhas do usuário para armazená-la no banco de dados;

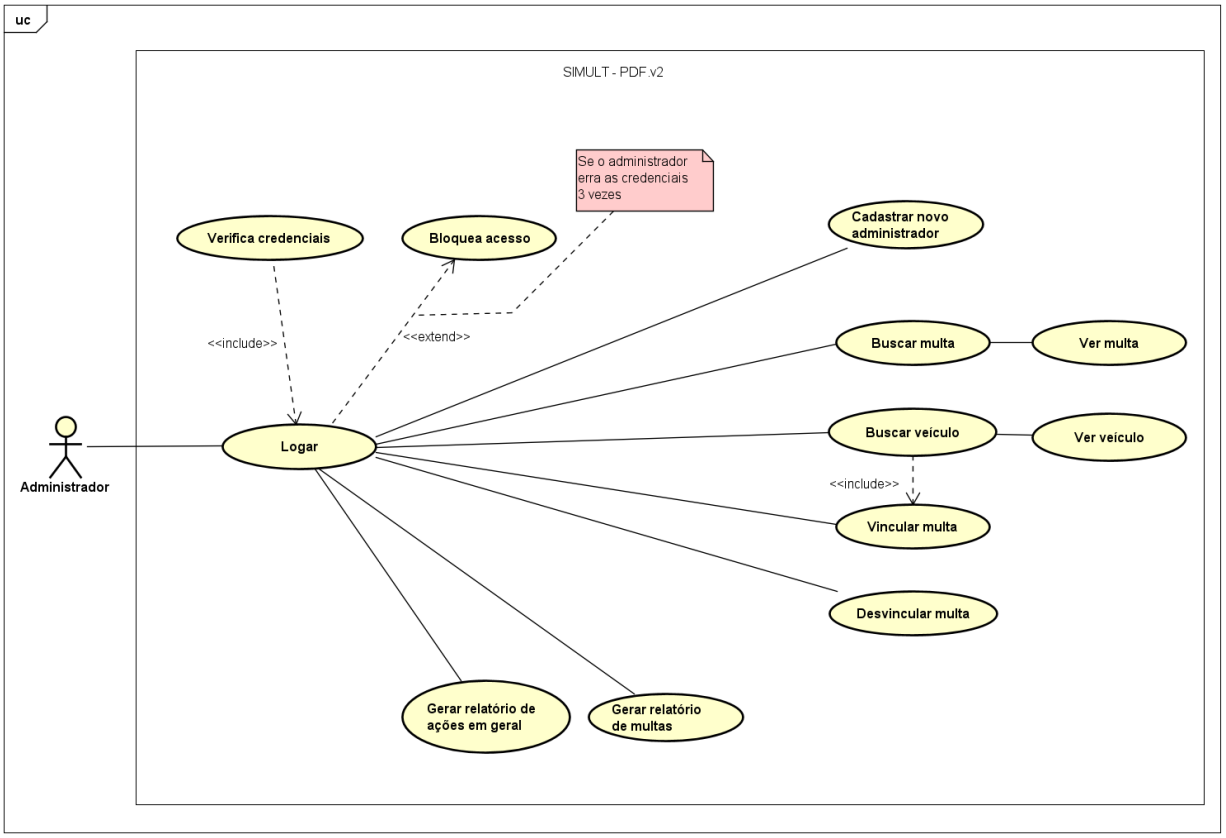
[RNF002] Deve ser bloqueado acesso ao sistema caso as credenciais tenham sido colocadas 3 vezes seguidas erradas.
- Descrição: O sistema deve bloquear o acesso e encerrar o processo quando for inserida credenciais incorretas 3 vezes seguidas durante o login do administrador;

[RNF003] O administrador deve ser notificado do sucesso ou fracasso da execução de uma operação;
- Descrição: O sistema deve retornar uma mensagem no terminal afirmando o sucesso ou o fracasso ao finalizar qualquer operação para melhor visualização do administrador.

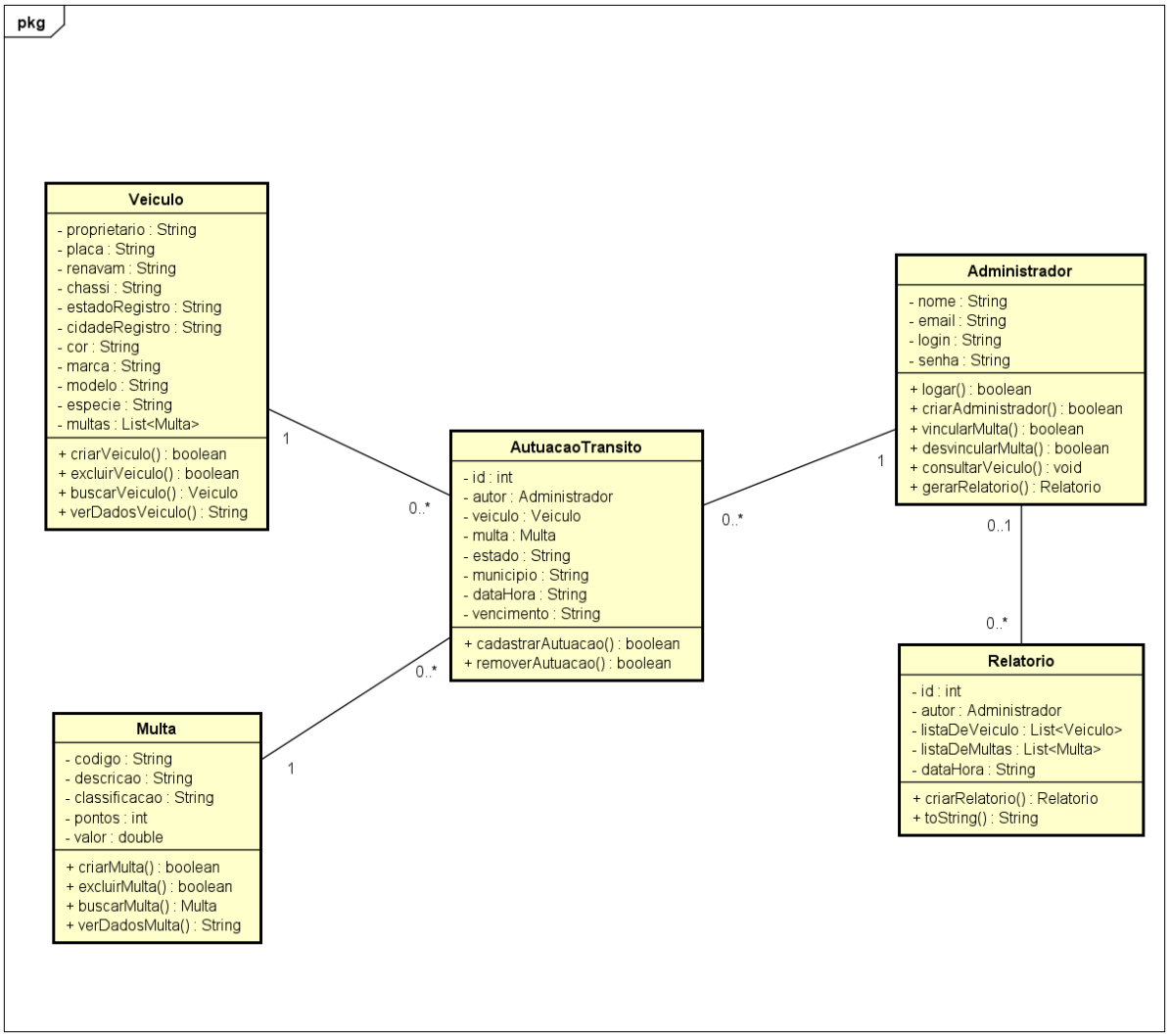
[RNF004] O sistema deve ser implementado em linguagem Java.

Artefatos

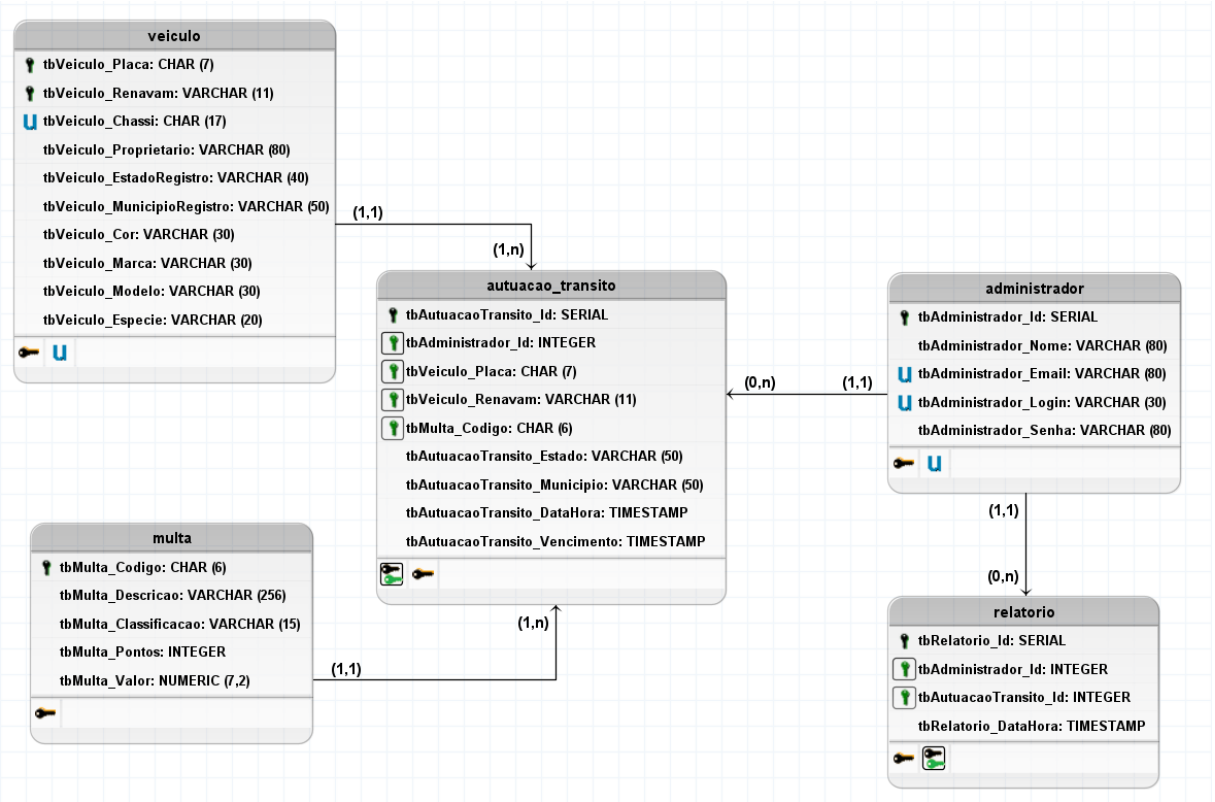
- Diagrama de Caso e Uso



- Diagrama de Classe



- Banco de Dados Modelo Lógico



Testes

A realização de casos de teste são essenciais no processo de desenvolvimento de um software, na qual, irá garantir a qualidade de um produto, sendo ele consistente e coerente com o que foi especificado. O foco dos testes é evitar e prevenir possíveis falhas durante o processo até a entrega do software, sendo assim, consideramos como base os testes estruturais e funcionais:

- **Teste Estrutural**
  - Fluxo de Controle: é utilizado principalmente para verificar os caminhos lógicos presente em um código, o testador irá pôr em execução cada caminho de uma instrução a outra no código. Assim, para evitar falhas e corrigir caminhos impossíveis, analisaremos os grafos de fluxo de controle do código do SIMULT-PDF.v2.

  - Fluxo de Dados: tem como objetivo detectar o uso impróprio de valores de dados devido ao erro na codificação, dessa forma, visa a contratação das entradas de valores das variáveis e em que lugar esses valores estão sendo utilizados. Nessa orientação, o sistema faz a definição e inicialização das credenciais logo no início das funções, estabelecendo seu uso na interação, quando necessária.

- **Teste Funcional**
  - Partição por Equivalência: esse método divide o domínio de entrada de um programa em classes de dados de onde podem ser criados casos de teste. A partir dessa conceituação, o emprego do método no projeto está relacionado à divisão das entradas do programa no que faz referência, por exemplo, a inserção de valores ligados à placa de um veículo. Nesse sentido, os testes baseados em partição por equivalência devem verificar a conformidade da entrada com o modelo de referência para as placas.

  - Análise do valor limite: principal objetivo desse tipo de teste é verificar as extremidades do software. Nesse sentido, tendo em vista as especificações do sistema SIMULT_PDF 2.0, bem como os requisitos não funcionais, o teste de análise do valor limite é utilizado para garantir precisão e segurança na validação das senhas, por exemplo. Dessa forma, os limites impostos nas condições de entrada são examinados conforme o que é especificado.

  - Grafo de Causa-Efeito: esse método, baseado em lógica, para ser implementado deve obedecer a uma série de especificações pré-estabelecidas. Assim, cada causa é relativa a uma condição. Já o efeito pode ser representado como um estado ou o produto de uma combinação de causas. Logo, partindo do programa em construção, entende-se que para cada ação realizada pelo usuário pode ser construído o grafo de causa-efeito contendo os possíveis efeitos correspondentes.

Devido à complexidade dos testes e aos erros associados, divide-se a atividade de teste em diversas fases visando objetivos distintos. Essas fases são essenciais para garantir a identificação de erros em diferentes níveis de abstração. Perante o exposto, o SIMULT-PDF 2.0 aborda as seguintes fases de teste: unidade, integração e aceitação.