

**Aluno: George Lincon Veloso Cruz.**

**PROJETO INTEGRADOR**

**SRP – SISTEMA DE RECLAMAÇÃO POPULAR**

**TADS 22**

**Projeto: SRP – Sistema de Reclamação Popular**

**Disciplina: Projeto Integrado**

**Professor: Diego Hoss**

**Itens da ideia do projeto:**

* O Sistema vai permitir fazer reclamação através do aplicativo. O usuário vai baixar o APP e vai fazer o login. No login vai solicitar dados pessoais para o cadastro;
* O Sistema vai permitir registrar reclamação, colocar o endereço e adicionar um setor e enviar para a Prefeitura;
* O Sistema foca em ajudar o cidadão em visualizar um problema e utilizar o seu celular para relatar a situação;
* O celular é uma ferramenta tecnológica que está presente com as pessoas e possibilita o uso para tal finalidade;
* O cidadão irá relatar o problema social, visando o ato de sanar o problema, comunicando através do sistema as autoridades responsáveis;
* O aplicativo irá permitir que o cidadão seja mais participativo e mostra para o poder público os problemas que muitas vezes não são relatadas e vistas como prioridade;
* O usuário consegue fazer a reclamação e acaba ajudando o poder público a criar medidas sociais mais efetivas;
* Com o tempo o Sistema irá armazenar um grande banco de dados de vários problemas sociais;
* A ideia do APP é ajudar na qualidade de vida das pessoas, dando poder as pessoas e não sendo mais necessário deslocar até a prefeitura da cidade e encarar filas para protocolar um problema.

**Levantamento de Requisitos:**

**Quadro 01 - Requisitos funcionais do sistema:**

|  |  |  |
| --- | --- | --- |
| **Identificação** | **Descrição** | **Regras de Negócio Envolvidas (0..N)** |
| RF001 | O usuário (cidadão) pode fazer a reclamação | RN001 |
| RF002 | O usuário (cidadão) pode informar o endereço |  |
| RF003 | O usuário (cidadão) pode escolher um setor |  |

**Casos de Uso:**

**Caso de uso:** **Registrar reclamação**

**Ator:** **Cidadão**

**Descrição:** O cidadão registra uma reclamação sobre um problema que está enfrentando.

**Fluxo principal:**

1. O cidadão acessa o sistema no aba reclamação.
2. O cidadão faz a sua reclamação no campo reclamar.
3. O cidadão informa o endereço.
4. O cidadão informa o setor.

**Fluxos alternativos:**

* **Caso o cidadão não preencha todos os campos obrigatórios:** O sistema apresenta uma mensagem de erro e solicita que o cidadão preencha os campos obrigatórios.

**Precondições:**

* O cidadão deve ter acesso à internet.
* O cidadão deve ter um dispositivo com um navegador de internet.

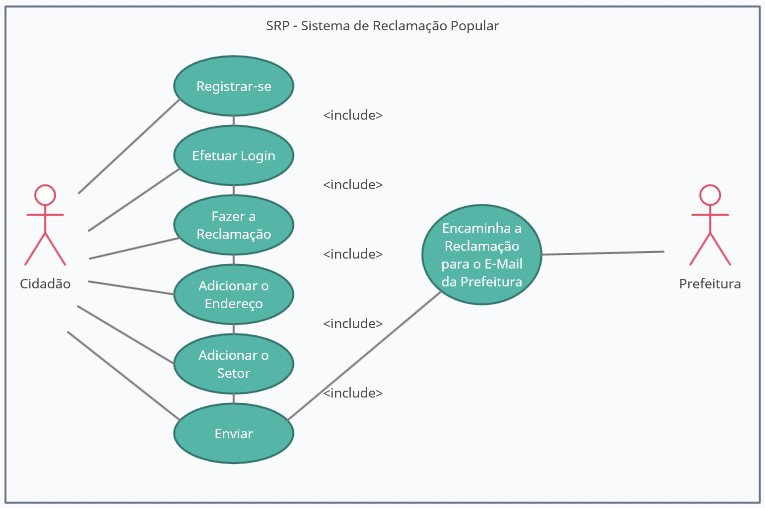
**Pós-condições:**

* A reclamação é registrada no sistema.

**Observações:**

* O sistema deve validar os dados informados pelo cidadão.
* O sistema deve armazenar os dados da reclamação de forma segura.

**Representação do Caso de Uso:**



**Especificação dos Casos de Uso:**

**Ator:** **Cidadão**

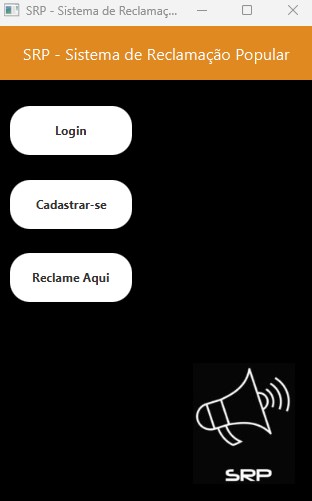
**Descrição:** O cidadão registra a reclamação.

**Fluxo principal:**

* Envia a Reclamação para a Prefeitura.

**Prototipação das Telas**

**Tela 1**



**Tela 2**



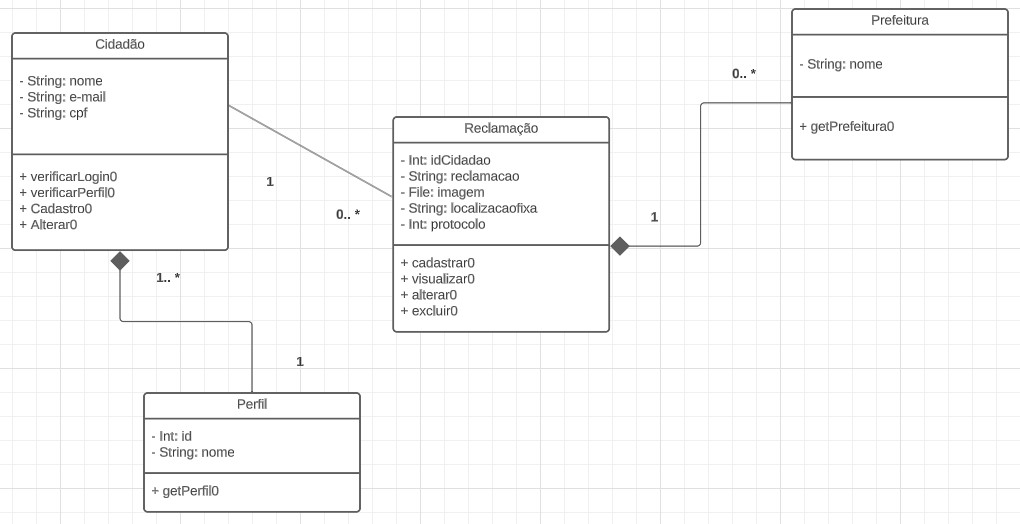
**Tela 3**



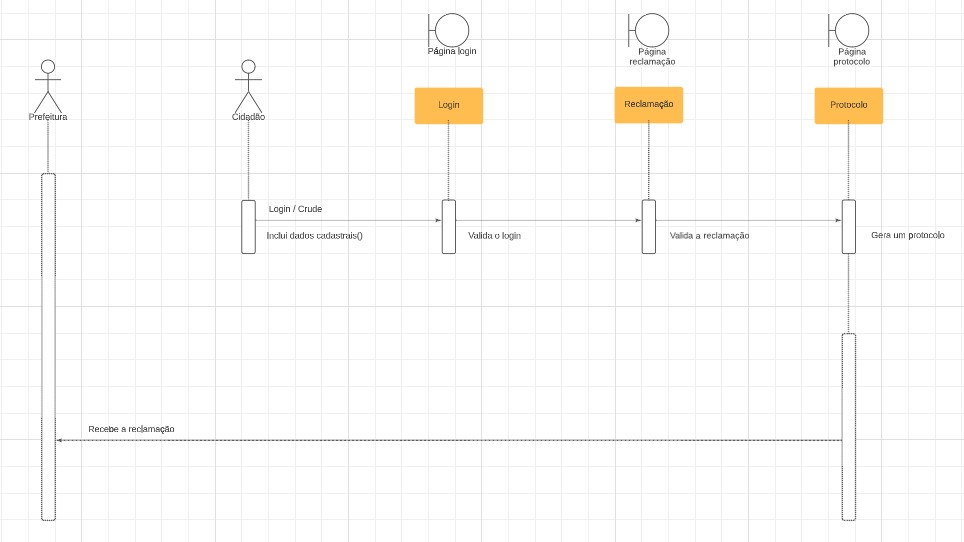
**Tela 4**



**Diagrama de Classes**



**Diagramas de Sequência**



**Projeto: SRP – Sistema de Reclamação Popular**

**Disciplina: Banco de Dados II**

**Professor: Wagner Weinert**

**O sistema proposto:**

**O trabalho consiste em:**

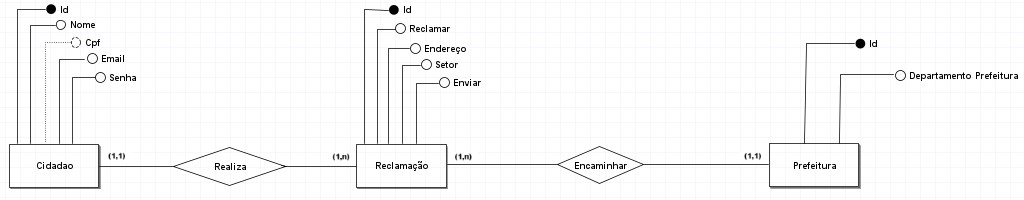
**1º - Definição de um Problema.**

O sistema visa permitir que os usuários registrem reclamações. O processo consiste em:

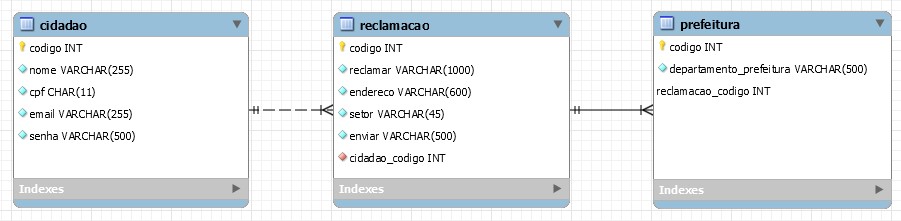
1. Criar uma reclamação.
2. Fornecer informações sobre o endereço relacionado à reclamação.
3. Identificar o setor ao qual a reclamação está associada.
4. Enviar a reclamação para a Prefeitura para análise e resolução.

**2º - Desenvolvimento de um Banco de Dados que represente o problema (Diagrama Entidade Relacionamento e Diagrama Relacional).**

**Diagrama Entidade Relacionamento:**



**Diagrama Relacional:**



**3º - Implementação (Script SQL) e Povoamento do Banco de Dados (Script SQL).**

**Implementação (Script SQL):**

CREATE TABLE `Cidadão` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`nome` varchar(255) NOT NULL,

`cpf` char(11) NOT NULL,

`email` varchar(255) NOT NULL,

`senha` varchar(500) NOT NULL,

PRIMARY KEY (`id`)

);

CREATE TABLE `Reclamação` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`reclamar` varchar(1000) NOT NULL,

`endereco` varchar(600) NOT NULL,

`setor` varchar(45) NOT NULL,

`enviar` varchar(500) NOT NULL,

`cidadao\_id` bigint NOT NULL,

`setor\_id` bigint NOT NULL,

PRIMARY KEY (`id`)

);

CREATE TABLE `Prefeitura` (

`id` bigint NOT NULL AUTO\_INCREMENT,

`departamento\_prefeitura` varchar(500) NOT NULL,

PRIMARY KEY (`id`)

);

**Povoamento do Banco de Dados (Script SQL):**

-- Inserindo dados na tabela Cidadão

INSERT INTO `Cidadão` (`nome`, `cpf`, `email`, `senha`)

VALUES

('João Silva', '12345678901', 'joao@email.com', 'senha123'),

('Maria Souza', '98765432109', 'maria@email.com', 'senha456');

-- Inserindo dados na tabela Prefeitura

INSERT INTO `Prefeitura` (`departamento\_prefeitura`)

VALUES

('Departamento de Obras'),

('Departamento de Saúde');

-- Inserindo dados na tabela Reclamação

INSERT INTO `Reclamação` (`reclamar`, `endereco`, `setor`, `enviar`, `cidadao\_id`, `setor\_id`)

VALUES

('Buraco na rua', 'Rua A, 123', 'Obras', 'Problema urgente', 1, 1),

('Falta de medicamentos', 'Rua B, 456', 'Saúde', 'Medicamentos essenciais em falta', 2, 2);

**4º - Implementação de 2 Triggers e 2 Funções.**

**1ª Trigger:**

DELIMITER //

CREATE TRIGGER CriptografarSenhaBeforeInsert

BEFORE INSERT ON Cidadão

FOR EACH ROW

BEGIN

SET NEW.senha = CriptografarSenha(NEW.senha);

END //

DELIMITER ;

**2ª Trigger:**

DELIMITER //

CREATE TRIGGER VerificarLimiteReclamacoes

BEFORE INSERT ON Reclamação

FOR EACH ROW

BEGIN

DECLARE total\_reclamacoes INT;

SET total\_reclamacoes = ContarReclamacoesCidadao(NEW.cidadao\_id);

IF total\_reclamacoes >= 10 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Limite de reclamações atingido para este cidadão.';

END IF;

END //

DELIMITER ;

**1ª Função:**

DELIMITER //

CREATE FUNCTION CriptografarSenha(input\_senha VARCHAR(500)) RETURNS VARCHAR(500)

BEGIN

DECLARE hashed\_password VARCHAR(500);

SET hashed\_password = SHA2(input\_senha, 256); -- Pode-se usar uma função de hash mais segura como SHA256

RETURN hashed\_password;

END //

DELIMITER ;

**2ª Função:**

DELIMITER //

CREATE FUNCTION ContarReclamacoesCidadao(citizen\_id BIGINT) RETURNS INT

BEGIN

DECLARE count\_reclamacoes INT;

SELECT COUNT(\*) INTO count\_reclamacoes

FROM Reclamação

WHERE cidadao\_id = citizen\_id;

RETURN count\_reclamacoes;

END //

DELIMITER ;

**5º - Implementação de 3 Relatórios utilizando os conceitos de Store Procedure, Cursores e Transação.**

**Relatório 1: Listar todas as reclamações de um cidadão específico:**

**Stored Procedure para listar reclamações de um cidadão:**

DELIMITER //

CREATE PROCEDURE ListarReclamacoesCidadao(IN cidadão\_id BIGINT)

BEGIN

SELECT \*

FROM Reclamação

WHERE cidadao\_id = cidadão\_id;

END //

DELIMITER ;

**Relatório 2: Listar todas as reclamações de um determinado setor:**

**Stored Procedure para listar reclamações de um setor específico:**

DELIMITER //

CREATE PROCEDURE ListarReclamacoesSetor(IN setor\_nome VARCHAR(45))

BEGIN

SELECT \*

FROM Reclamação r

WHERE r.setor = setor\_nome;

END //

DELIMITER ;

**Relatório 3: Atualizar o endereço de uma reclamação específica:**

**Stored Procedure para atualizar o endereço de uma reclamação:**

DELIMITER //

CREATE PROCEDURE AtualizarEnderecoReclamacao(IN reclamacao\_id BIGINT, IN novo\_endereco VARCHAR(600))

BEGIN

DECLARE exit handler for sqlexception

BEGIN

ROLLBACK;

END;

DECLARE exit handler for sqlwarning

BEGIN

ROLLBACK;

END;

START TRANSACTION;

UPDATE Reclamação

SET endereco = novo\_endereco

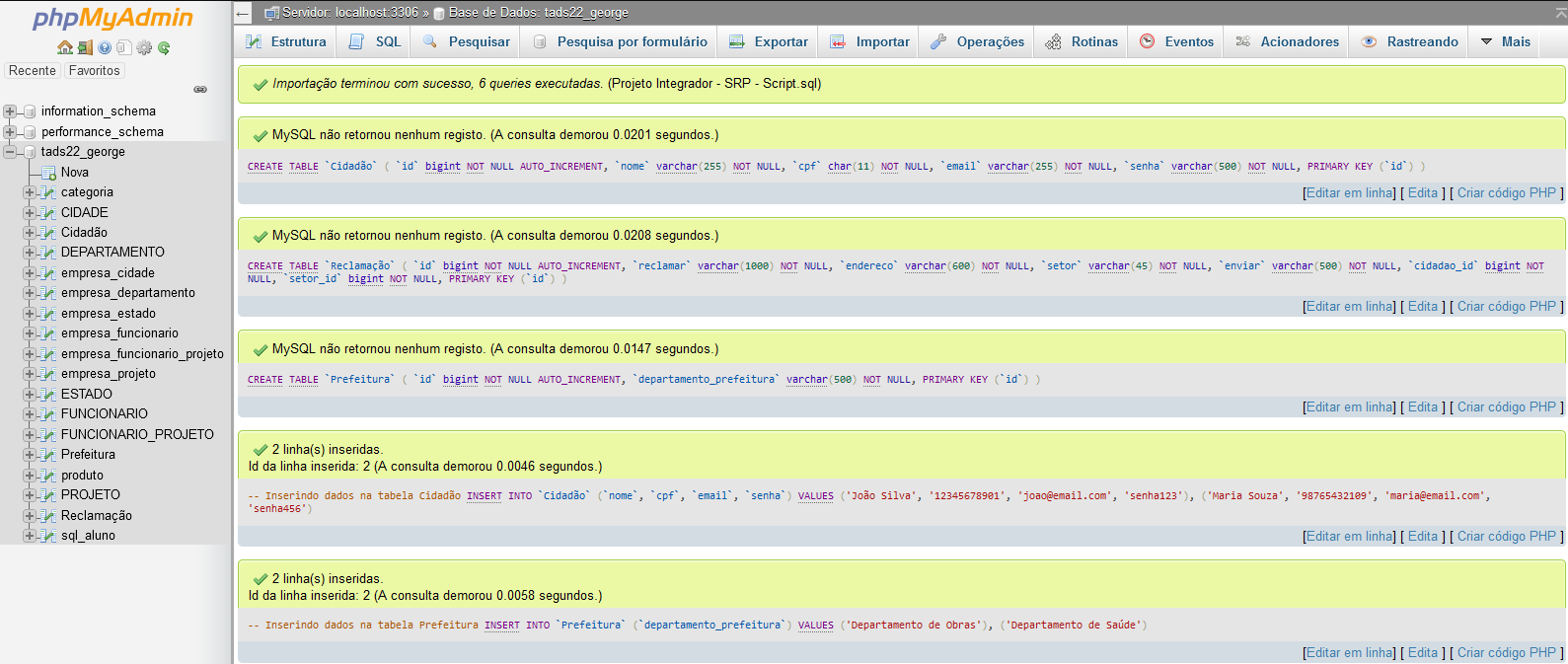
WHERE id = reclamacao\_id;

COMMIT;

END //

DELIMITER ;

**\* Print da Implementação – Banco de Dados:**



**Banco de Dados: http://wagnerweinert.com.br/phpmyadmin**

**Projeto: SRP – Sistema de Reclamação Popular**

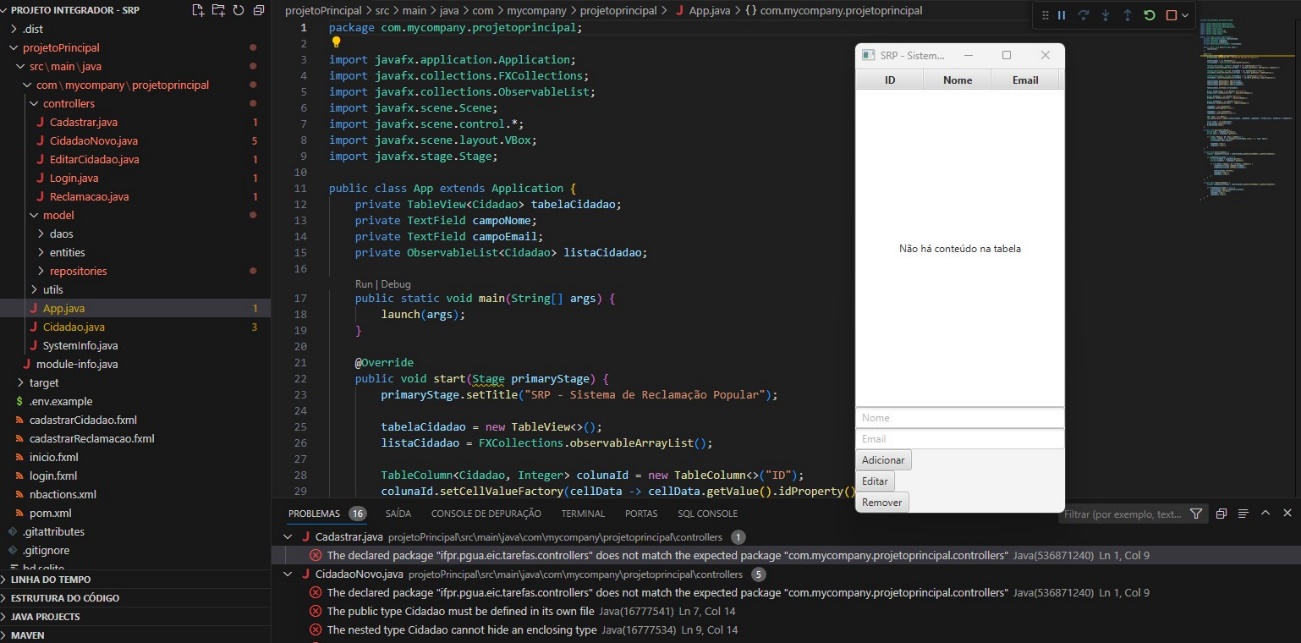
**Disciplina: Orientação a Objetos II**

**Professor: Hugo Perlin**

**Requisitos Orientação a Objetos:**

Os requisitos da disciplina de OO que deverão constar no projeto são:

* Deve implementar no **mínimo três** casos de uso. Casos de uso manter incluem as seguintes operações (cadastrar, editar, listar e remover (marcar como inativo)).
* Deve implementar controle de acesso (caso de uso separado).
* Deve ser desenvolvido em Java.
* Deve conter interface gráfica utilizando JavaFX.
* Deve utilizar Git e Github para fazer o controle de versão.
* Deve utilizar os padrões de projetos abordados (DAO, Repository, Factory, MVC).



**(Projeto Integrador - SRP - Demonstração do Cadastro do Cidadão)**