



Atividade 2

Ocorrências Criminais - Projeto e Modelo de Dados da Aplicação

Grupo:

Carlos Eduardo Dias dos Santos

Jeferson de Oliveira Santos

João Pablo da Paz de Jesus

João Rosa Conceição

Lucas Aragão Damaceno

Marcelo Venicius Almeida Lima

Elaboração do Projeto:

Link do Github: <https://github.com/Pablo-oficial/Atividade-2---Engenharia-de-Software-2.git>

Recapitulando:

O projeto visa aprimorar a acessibilidade e usabilidade na busca e utilização de dados públicos de segurança pública disponíveis no Ministério da Justiça, por meio do portal dados.gov.br. Atualmente, a obtenção desses dados demanda navegação por diversas etapas, como pesquisa em mecanismos de busca e redirecionamento para o site do Ministério. O sistema proposto busca resolver essas questões, fornecendo uma plataforma mais organizada, acessível e fácil de usar.

Principais Pontos de Melhoria Propostos:

Organização das Informações:

Aprimorar a estrutura para facilitar a localização e compreensão dos dados, tornando a experiência do usuário mais intuitiva.

Descrição dos Arquivos:

Proporcionar informações detalhadas sobre os arquivos disponíveis, incluindo descrições claras para auxiliar os usuários na escolha do formato desejado.

Acessibilidade para Pessoas com Deficiência:

Implementar recursos que garantam a acessibilidade para pessoas com deficiência, tornando o portal e os dados compatíveis com tecnologias assistivas.

Diversificação de Formatos de Arquivo:

Além de PDFs e XLSX, oferecer dados em formatos adicionais como CSV, JSON ou APIs, ampliando a versatilidade e a compatibilidade com diversas plataformas.

Melhorias de Pesquisa:

Aprimorar as funcionalidades de pesquisa, incluindo SEO e filtros para facilitar a localização específica de informações desejadas.

Melhorias de Compartilhamento:

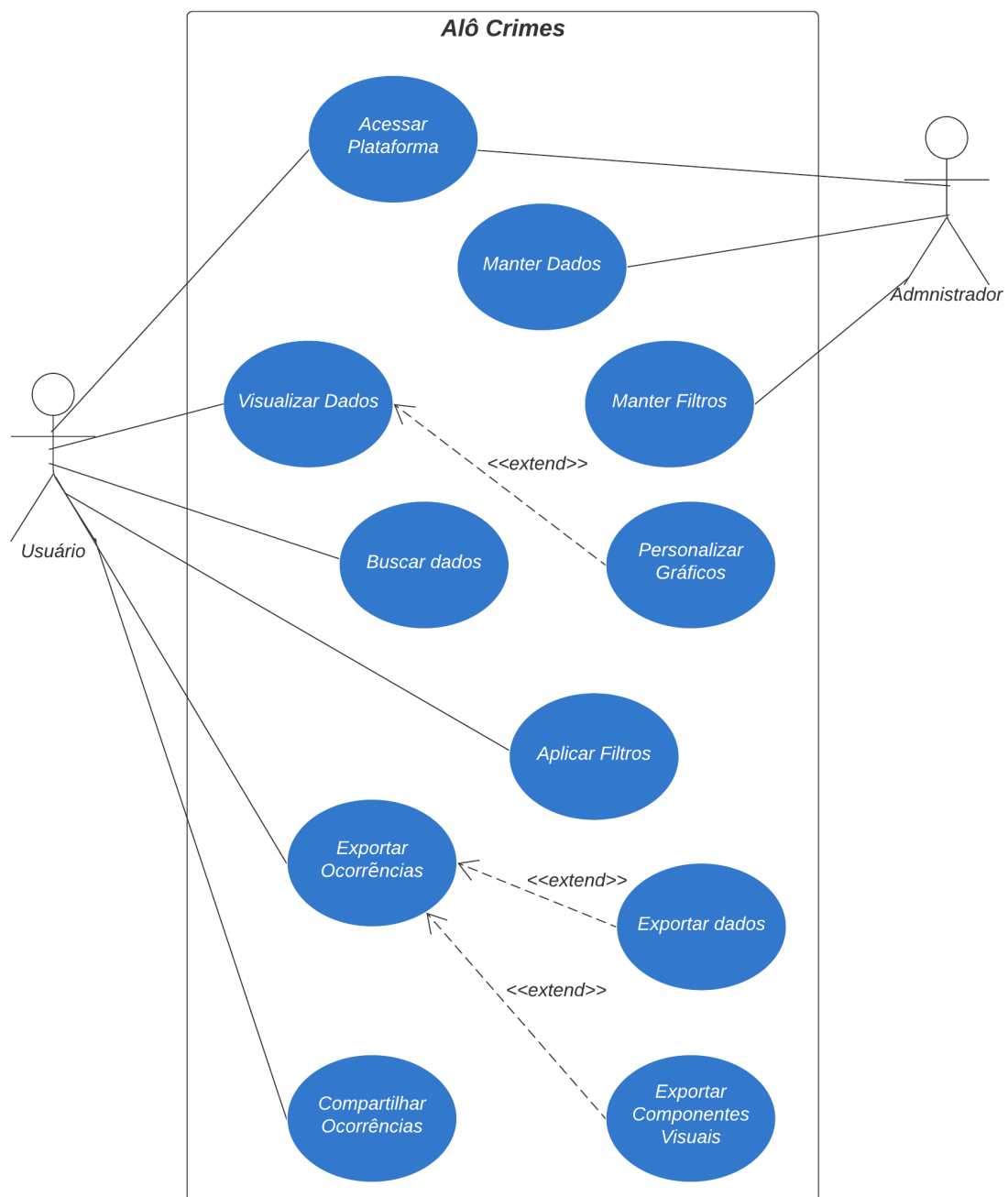
Implementar recursos aprimorados de compartilhamento de dados para facilitar a disseminação e integração em diferentes contextos.

Atualização Regular e Manutenção:

Garantir a regularidade na atualização e revisão dos dados, informando os usuários sobre quaisquer modificações nos conjuntos de dados para manter a confiança na qualidade e precisão das informações.

Diagrama Casos de Uso:

Diagrama Casos de Uso



Os casos de uso envolvem ações dos atores que são:

Usuário: ações relacionadas a filtrar, visualizar e exportar dados da plataforma.

Administrador: ações relacionadas a manter o sistema em funcionamento e implementar melhorias.

Diagrama de Classes:

O diagrama de classes é uma representação visual da estrutura de um sistema orientado a objetos, mostrando as classes no sistema, seus atributos, métodos e os relacionamentos entre elas. No contexto do projeto de ocorrências criminais, o diagrama de classes pode ser representado da seguinte forma:

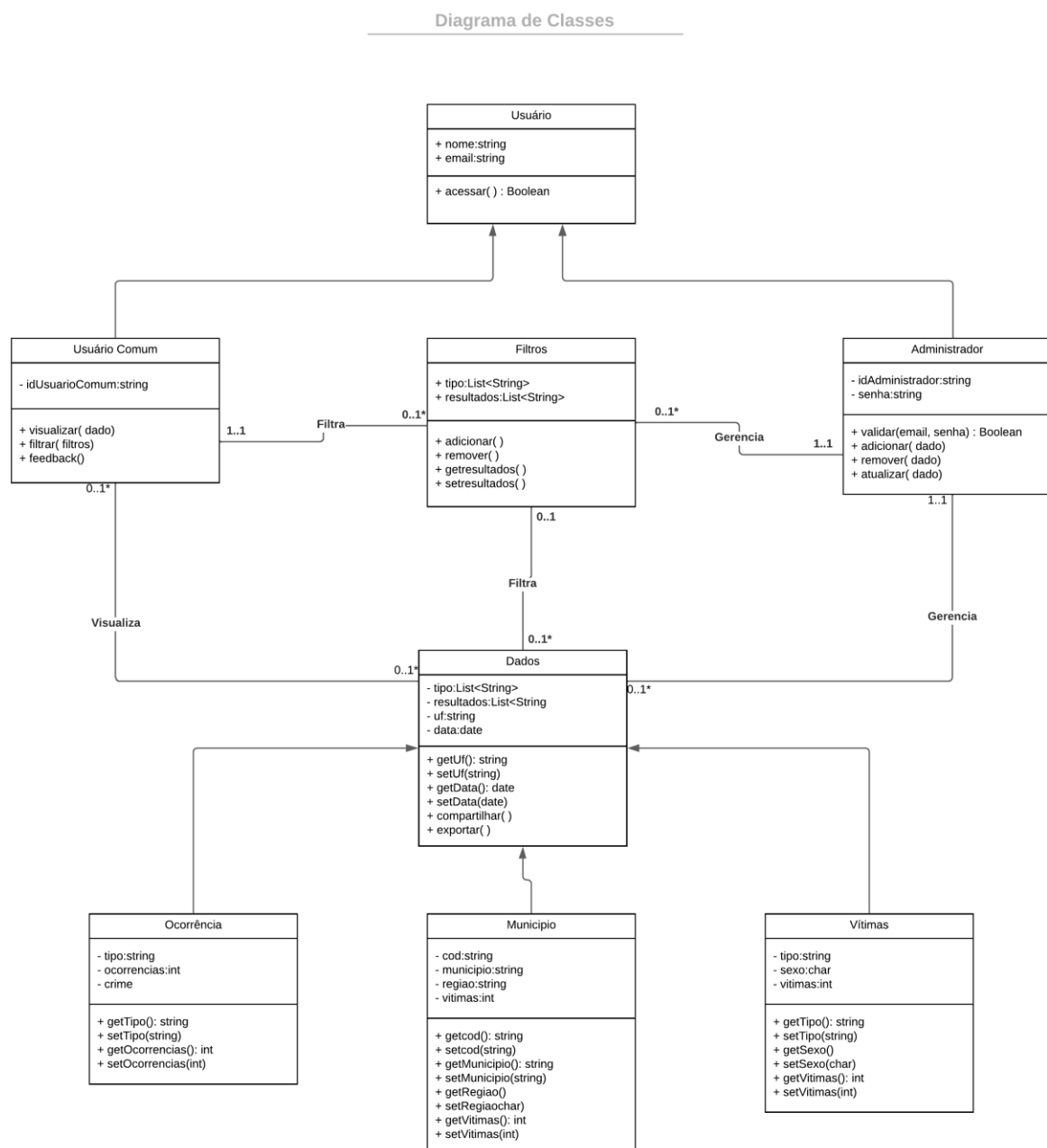
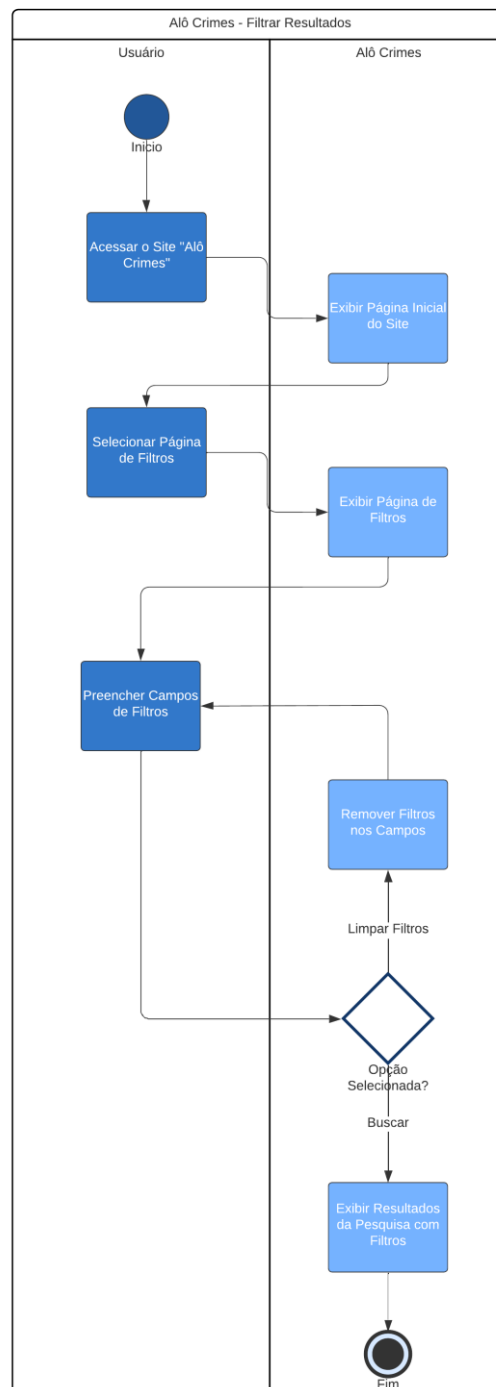


Diagrama de Atividades - Filtrar:

Diagrama de Atividades - Filtrar



O diagrama de atividades para a funcionalidade de filtrar resultados da busca no sistema "Alô Crimes" é projetado para otimizar a experiência do usuário da seguinte maneira:

1. Ingresso no Sistema:

O usuário acessa o sistema "Alô Crimes" e é automaticamente direcionado para a página inicial.

2. Navegação até a Página de Filtros:

O usuário navega para a seção de "Filtros" na plataforma.

3. Exibição da Página de Filtros:

O sistema exibe a página de "Filtros", apresentando campos de preenchimento para auxiliar o usuário na definição de seus critérios de pesquisa.

4. Preenchimento dos Campos de Filtros:

O usuário preenche os campos de acordo com os critérios específicos de sua pesquisa.

5. Condição de Decisão:

Após o preenchimento, o usuário é direcionado para uma condição de decisão que consiste em dois caminhos distintos:

5.1. Limpar Filtros:

Se o usuário optar por limpar os filtros, o sistema remove os campos preenchidos e retorna à página de "Filtros".

5.2. Buscar Resultados:

Se o usuário optar por buscar, o sistema utiliza os critérios estabelecidos nos campos de filtros para realizar a pesquisa.

6. Exibição dos Resultados da Pesquisa:

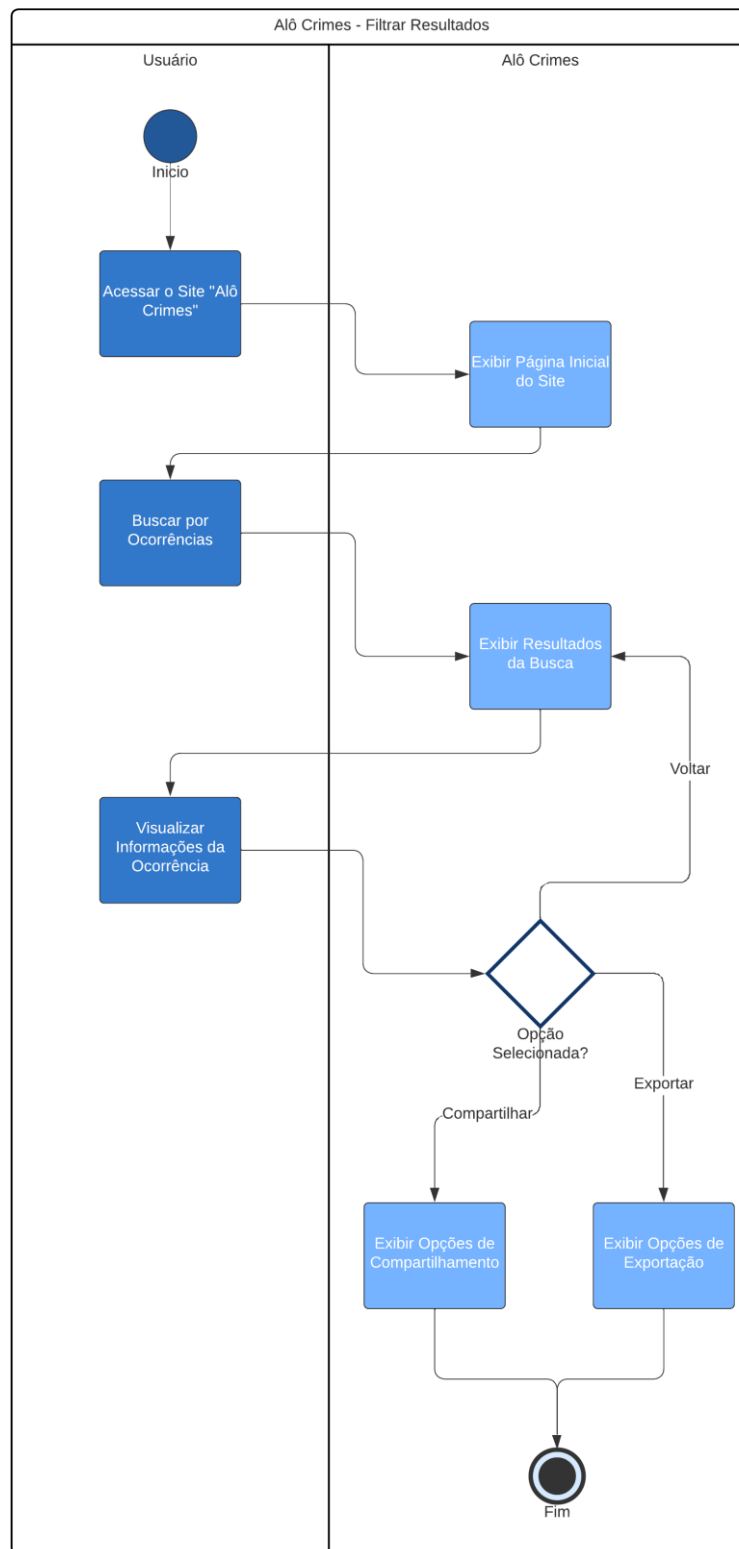
Os resultados da pesquisa são apresentados ao usuário de acordo com os critérios definidos anteriormente.

7. Encerramento da Atividade:

Após a exibição dos resultados, a atividade é encerrada, proporcionando ao usuário uma conclusão clara da interação.

Diagrama de Atividades – Compartilhar/Exportar:

Diagrama de Atividades -
Exportar/Compartilhar



O diagrama de atividades Exportar/Compartilhar engloba dois requisitos funcionais de uma vez só, que serão as duas opções oferecidas ao usuário, nesse caso, ao

selecionar uma ocorrência, o mesmo poderá compartilhar a ocorrência com outro usuário, ou escolher um formato e exportar o arquivo. O caso de uso ocorre da seguinte forma:

Ingresso no Sistema:

O usuário acessa o sistema "Alô Crimes" e é automaticamente direcionado para a página inicial.

Buscar Ocorrência:

O usuário realiza a busca por uma ocorrência de interesse na plataforma.

Exibição da Resultados da Busca:

O sistema exibe os resultados da busca com base na descrição dos critérios de pesquisa do usuário.

Visualizar Ocorrência:

O usuário seleciona ocorrência de preferência para visualização.

Condição de Decisão:

Após o preenchimento, o usuário é direcionado para uma condição de decisão que consiste em dois caminhos distintos:

1 – Compartilhar Dados:

Se o usuário optar por compartilhar dados, o sistema exibe as opções de compartilhamento, via Link, E-mail, Whatsapp , etc...

2 – Exportar Dados:

Se o usuário optar por exportar dados, o sistema exibe as opções de exportação, via CSV, XLSX, PDF, PNG, etc...

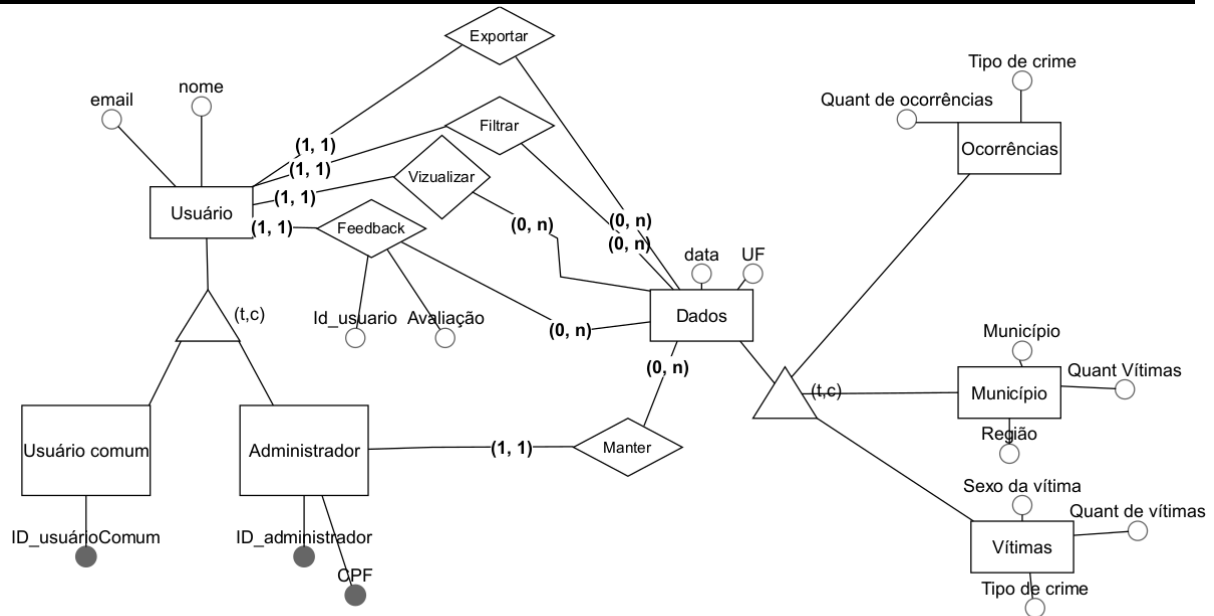
Encerramento da Atividade:

Após a exibição dos resultados, a atividade é encerrada, proporcionando ao usuário uma conclusão clara da interação.

Modelagem de Dados:

Projeto

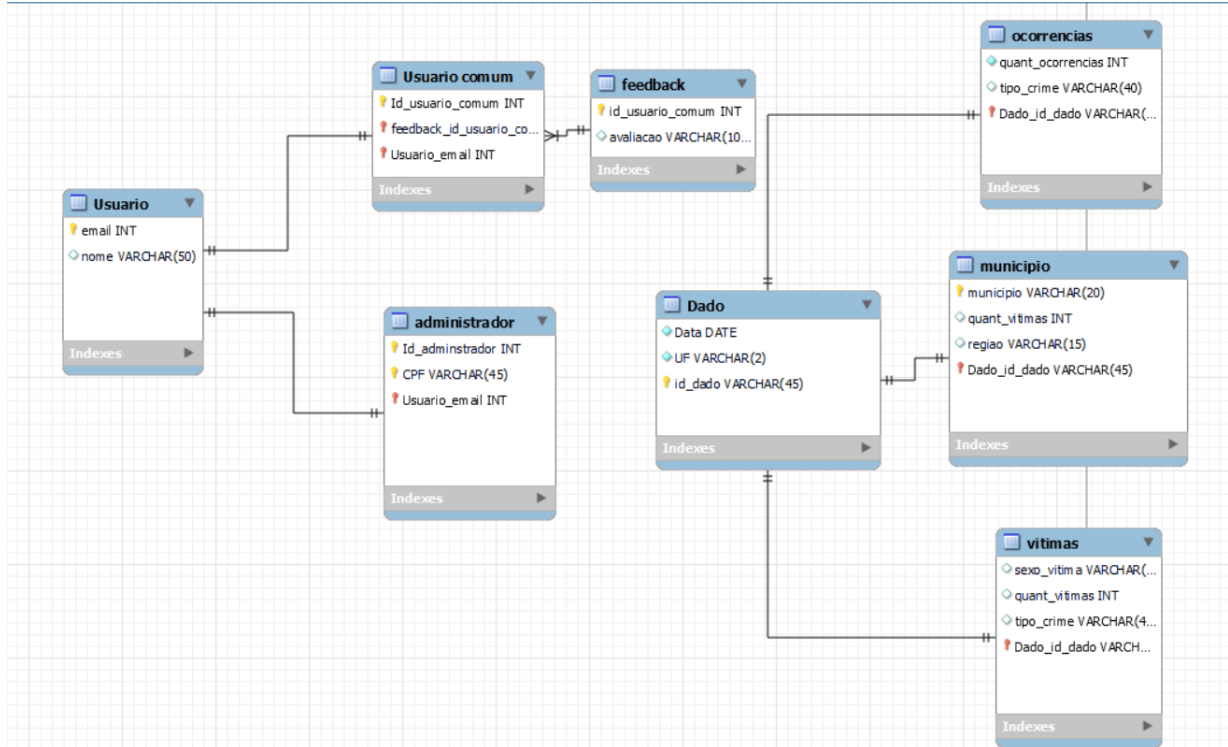
Conceitual:



O projeto conceitual de banco de dados foi escolhido pois permite uma fácil e intuitiva avaliação inicial de como será modelada a database, esse modelo possui entidades, relacionamentos e também atributos que pertencem a essas entidades, no caso do nosso projeto conceitual existem as entidades Usuário com os atributos nome e Email, essa entidade possui duas entidades que herdam ela sendo elas usuário comum e administrador, os quais possuem chaves primarias (ID_usuarioComum, ID_administrador e CPF), a entidade administrador possui uma relação única com a entidade de dados que é a de manter os dados(fazer o CRUD dos dados no banco), as outras relações dos dados são realizadas pela entidade usuário geral mesma e consistem em um feedback que avaliará a experiência do usuário na aplicação, visualizar que fará com que o usuário consiga visualizar os dados de um jeito mais limpo e intuitivo do que no site do gov, filtrar que permitirá que o usuário filtre e organize os dados do jeito que ele desejar e por último exportar que será responsável por permitir que o usuário exporte os dados em formatos .png, .pdf e/ou .xls.

Modelo Físico:

Do modelo conceitual foi possível gerar o modelo físico a seguir:



Esta modelagem foi realizada utilizando o aplicativo MySQL Workbench, que oferece a capacidade de desenvolver diagramas de modelagem física. Os elementos principais deste modelo são evidenciados através de cores distintas: chaves primárias são destacadas em amarelo, atributos definidos como NOT NULL são apresentados em azul claro, e os atributos normais estão identificados em branco.

A entidade "usuário" desempenha um papel central, atuando como progenitor de uma hierarquia. Nesta estrutura, "usuario_comum" e "administrador" são subentidades que herdam características da entidade "usuário". O "usuario_comum" tem a capacidade de enviar feedback, sendo essencial o seu registro no sistema. Enquanto isso, o "administrador" é designado para realizar eventuais manutenções no sistema.

Adicionalmente, uma outra hierarquia é estabelecida entre a entidade "dado" e suas entidades descendentes: "municipio", "ocorrencias" e "vitimas". Esses relacionamentos são cruciais para a inserção dos dados obtidos do Ministério da Justiça.

Segue abaixo a apresentação dos scripts elaborados para a criação das tabelas no banco de dados, consolidando assim a representação física destas entidades e relacionamentos.

```
-- Schema aloCrimes
```

```
CREATE SCHEMA IF NOT EXISTS `aloCrimes` ;
```

```
USE `aloCrimes` ;
```

```
-- -----
```

```
-- Table `aloCrimes`.`feedback`
```

```
-- -----
```

```
CREATE TABLE IF NOT EXISTS `aloCrimes`.`feedback` (  
  `id_usuario_comum` INT NOT NULL,  
  `avaliacao` VARCHAR(100) NULL,  
  PRIMARY KEY (`id_usuario_comum`))
```

```
-- -----
```

```
-- Table `aloCrimes`.`Usuario`
```

```
-- -----
```

```
CREATE TABLE IF NOT EXISTS `aloCrimes`.`Usuario` (  
  `email` INT NOT NULL,  
  `nome` VARCHAR(50) NULL,  
  PRIMARY KEY (`email`),  
  UNIQUE INDEX `nome_UNIQUE` (`nome` ASC))
```

```
-- -----
```

```
-- Table `aloCrimes`.`Usuario_comum`
```

```
-- -----
```

```
CREATE TABLE IF NOT EXISTS `aloCrimes`.`Usuario_comum` (  
  `Id_usuario_comum` INT NOT NULL,  
  `feedback_id_usuario_comum` INT NOT NULL,  
  `Usuario_email` INT NOT NULL,  
  PRIMARY KEY (`Id_usuario_comum`, `feedback_id_usuario_comum`,  
  `Usuario_email`),  
  INDEX `fk_Usuario_comum_feedback1_idx` (`feedback_id_usuario_comum` ASC),
```

```

INDEX `fk_Usuario comum_Usuario1_idx` (`Usuario_email` ASC),
CONSTRAINT `fk_Usuario comum_feedback1`
  FOREIGN KEY (`feedback_id_usuario_comum`)
  REFERENCES `aloCrimes`.`feedback` (`id_usuario_comum`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_Usuario comum_Usuario1`
  FOREIGN KEY (`Usuario_email`)
  REFERENCES `aloCrimes`.`Usuario` (`email`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)

```

-- Table `aloCrimes`.`administrador`

```

CREATE TABLE IF NOT EXISTS `aloCrimes`.`administrador` (
  `Id_administrador` INT NOT NULL,
  `CPF` VARCHAR(45) NOT NULL,
  `Usuario_email` INT NOT NULL,
  PRIMARY KEY (`Id_administrador`, `CPF`, `Usuario_email`),
  INDEX `fk_administrador_Usuario1_idx` (`Usuario_email` ASC),
  CONSTRAINT `fk_administrador_Usuario1`
    FOREIGN KEY (`Usuario_email`)
    REFERENCES `aloCrimes`.`Usuario` (`email`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

-- Table `aloCrimes`.`Dado`

```
-- -----  
CREATE TABLE IF NOT EXISTS `aloCrimes`.`Dado` (  
  `Data` DATE NOT NULL,  
  `UF` VARCHAR(2) NOT NULL,  
  `id_dado` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`id_dado`))
```

```
-- -----  
-- Table `aloCrimes`.`ocorrencias`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `aloCrimes`.`ocorrencias` (  
  `quant_ocorrencias` INT NOT NULL,  
  `tipo_crime` VARCHAR(40) NULL,  
  `Dado_id_dado` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Dado_id_dado`),  
  CONSTRAINT `fk_ocorrencias_Dado1`  
    FOREIGN KEY (`Dado_id_dado`)  
    REFERENCES `aloCrimes`.`Dado` (`id_dado`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
-- -----  
-- Table `aloCrimes`.`municipio`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `aloCrimes`.`municipio` (  
  `municipio` VARCHAR(20) NOT NULL,  
  `quant_vitimas` INT NULL,  
  `regiao` VARCHAR(15) NULL,  
  `Dado_id_dado` VARCHAR(45) NOT NULL,
```

```
PRIMARY KEY (`municipio`, `Dado_id_dado`),  
INDEX `fk_municipio_Dado1_idx` (`Dado_id_dado` ASC),  
CONSTRAINT `fk_municipio_Dado1`  
  FOREIGN KEY (`Dado_id_dado`)  
  REFERENCES `aloCrimes`.`Dado` (`id_dado`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION)
```

```
-- -----  
-- Table `aloCrimes`.`vitimas`  
-- -----
```

```
CREATE TABLE IF NOT EXISTS `aloCrimes`.`vitimas` (  
  `sexo_vitima` VARCHAR(15) NULL,  
  `quant_vitimas` INT NULL,  
  `tipo_crime` VARCHAR(45) NULL,  
  `Dado_id_dado` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Dado_id_dado`),  
  CONSTRAINT `fk_vitimas_Dado1`  
    FOREIGN KEY (`Dado_id_dado`)  
    REFERENCES `aloCrimes`.`Dado` (`id_dado`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```