**Estrutura de estrutura de dados.**

**src**: diretório de código principal.  
**Configuração**: configuração do banco de dados MySQL usando Sequelize.  
**Manager**: Gerenciador de lógica de negócios (ex.).  
**Middleware**: Funções para validação e autenticação de dados (ex. ValidateCommentary).  
**Model**: Um modelo que representa uma entidade de dados (ex. usuário, comentário).  
**Roteador**: definições de rota API (ex.: routerUser, routerComment).  
**Carregar**: Salve os arquivos carregados (por exemplo, arquivos de imagem).  
**server.js**: Arquivo padrão para iniciar o servidor Express.

**Tecnologia**

* **Node.js**: ambiente de execução JavaScript.
* **Express**: Uma estrutura para construir APIs.
* **Sequelize**: ORM para interagir com MySQL.
* **Multer**: Middleware para transferência de arquivos.
* **JWT**: Autenticação do usuário.
* **CORS**: Permitir solicitações de origem cruzada.

**Funcionalidades**

* **CRUD** para usuários, ações, comentários e anúncios.
* **Validação** dos dados de entrada.
* **Manipulação de Imagens** (upload/download).
* **Resposta** em formato JSON para interação com a interface.

**Prefácio**

**Estrutura de Diretórios**

* **src**: Diretório de código principal.
* **Anterior**: Aplicativo Android (Java).
  + **main/java/br/com/aula/text**: contém funções principais (login, cadastrar, ler, falar, alimentar).
  + **Post.java**: Modelo de postagem.
  + **Comment.java**: Modelo de comentário.
  + **Adaptador**: Adaptador para RecyclerView (por exemplo, FeedAdapter).
  + **ApiClient.java**: Manipula solicitações HTTP para a API.

**Tecnologia**

* **Android SDK**: Desenvolvimento de aplicativos Android.
* **OkHttp**: Uma biblioteca para solicitações HTTP.
* **Flow**: Uma biblioteca para carregar imagens.
* **Material Design**: Componentes de UI responsivos.

**Funcionalidades**

* Interface para registrar, gravar, enviar e comunicar.
* Uso do **RecyclerView** para exibir listas.
* Envio de imagens e criptografia de dados confidenciais.
* Navegação entre telas do aplicativo.

**Visão Geral do Sistema**

**O Sistema**

O sistema é um aplicativo web e móvel que permite interagir com mídia e informações. O **backend** usa Node.js com Express e MySQL, enquanto o **frontend** é um aplicativo Android. O sistema inclui:

* **Autenticação de usuário.**
* **Envio de fotos.**
* **Banco de dados e design de resposta.**

**Estruturas de Dados e Algoritmos**

* **Arrays e Listas**: Utilizados para armazenar dados tanto no front-end (Java) quanto no back-end (JavaScript).
* **Filas e Pilhas**:
  + **Filas** podem ser usadas para processar tarefas, como carregamento de imagens.
  + **Pilhas** são usadas para mover-se entre as telas do Android.

**Objetivo**: Exibir postagens, comentários e usuários, além de lidar com a manipulação e transferência de dados entre o front-end e back-end.

* **Mapa (Dicionário)**: Armazena pares de valores-chave, como localizações e configurações do usuário.
* **Guia**: Estrutura de árvore de histórico ou peça pendurada na frente.

**Protocolo de Comunicação**

* Utiliza **REST** para comunicação do front-end para o back-end.

**Conceitos de Programação e Segurança**

* **Programação Síncrona**: Implementada para sincronizar/esperar com o back-end para otimizar as solicitações.
* **Padrão de Design**: Utiliza o padrão **Model-View-Controller (MVC)** para separar a lógica de negócios, dados e visualização.
* **Validação e Segurança de Dados**:
  + Validação de dados de back-end usando **middleware**.
  + Recursos de segurança, como **criptografia de senha** e uso de **JWT** para autenticação do usuário.