

APRESENTANDO O

INDIESTREAM

Integrantes da equipe

Manoel Ramos - 2317037 Mario Caeté - 2312640 Felipe Martins - 2312673 Luis Kaio - 2312648 Matheus Moura -2422792

INTRODUÇÃO

IndieStream é uma plataforma de conexão entre profissionais do cinema brasileiro dedicada a conteúdos independentes, como filmes, músicas e roteiros.

Nosso objetivo é fornecer uma alternativa às grandes plataformas de streaming, promovendo a diversidade cultural e dando visibilidade a artistas independentes. Promovendo o networking e gerando mais oportunidades aos profissionais da área.



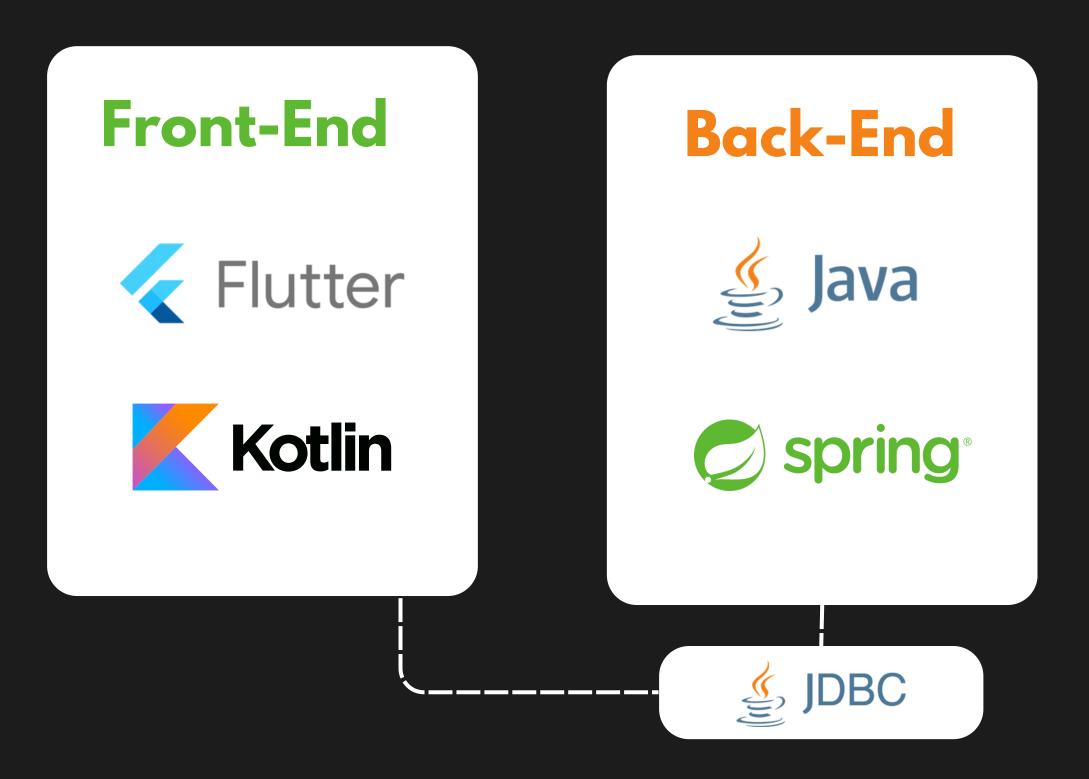
PRINCIPAIS RECURSOS

 Criação de Perfis: Permite que usuários criem perfis detalhados com suas funções, projetos, e obras favoritas.

 Gestão de Projetos: Ferramentas para criar, gerenciar e divulgar projetos com status atualizados. • Inscrições em Projetos: Funcionalidade para profissionais se inscreverem em projetos abertos.

 Busca Avançada: Sistema de filtragem por localização, função, nome, idade, entre outros critérios.

Tecnologias Ultilizadas



Data Base



✓ INDIESTREAM-APP

- > 🖷 android
- > @ assets
- > 🔳 Documentação
- > 📑 ios
- ∨ 📹 lib
 - AppColors.dart
 - auth_service.dart
 - cadastro_screen.dart
 - config_page.dart
 - nedicao-perfil_screen.dart
 - nome_screen.dart
 - login_screen.dart
 - nain.dart
 - onboarding_screen.dart
 - nessoas_screen.dart
 - nlanos_page.dart
 - projeto_create_screen.dart
 - nrojeto_detalhes_screen.dart
 - projetos_screen.dart
 - sobre_nos.dart
 - theme_provider.dart
 - user_detalhes_screen.dart
- > **=** seeder_script
- > 🧖 test
- > 臧 web
- > **ii** windows
- .gitignore
- .metadata
- analysis_options.yaml

Estruturação do Front-end

Arquitetura em Camadas (Layered Architecture)

main (Ponto de Entrada)

Inicializa o aplicativo, configura rotas e define o tema principal da aplicação.

auth_service (Serviço de Autenticação)

Gerencia a lógica de login, registro e sessão do usuário, separado da camada visual.

AppColors (Configuração de Estilo)

Define paleta de cores reutilizável em toda a aplicação, promovendo consistência visual.

✓ src > iii db ✓ iii main ✓ iii java\com\unifor\indiestream ✓ iii config

- SecurityConfig.java
- - GdadeController.java
 - ComentarioController.java
 - EstadoController.java
 - HabilidadeController.java
 - LinhaDoTempoController.java
 - ProfissaoController.java
 - A ProjetoController.java
 - UsuarioController.java
- - AtualizarUsuarioDTO.java
 - GidadeDTO.java
 - ComentarioDTO.java
 - EstadoDTO.java
 - # HabilidadeDTO.java
 - LinhaDoTempoDTO.java
 - PessoaEnvolvidaDTO.java
 - ProfissaoDTO.java
 - ProjetoDTO.java
 - UsuarioDTO.java
- 😕 🐝 mode
 - 🚣 Cidade.java
 - 🐇 Comentario.java
 - Estado.java
 - Habilidade.java
 - LinhaDoTempo.java
 - Profissao.java
 - A Projeto.java
 - Usuario java
- v 📹 repository
 - CidadeRepository.java
 - ComentarioRepository.java
 - EstadoRepository.java

Estruturação do Back-end

Arquitetura em Camadas (Layered Architecture)

Config (Configuração de Segurança)

Responsável por definir configurações globais da aplicação, como autenticação, autorização e rotas protegidas.

Controller (Controlador de API)

Recebe as requisições do cliente, valida os dados e repassa para a camada de serviço. Atua como ponto de entrada da API.

Repository (Acesso a Dados)

Isola a lógica de acesso ao banco de dados. Realiza consultas, inserções e atualizações de forma organizada.

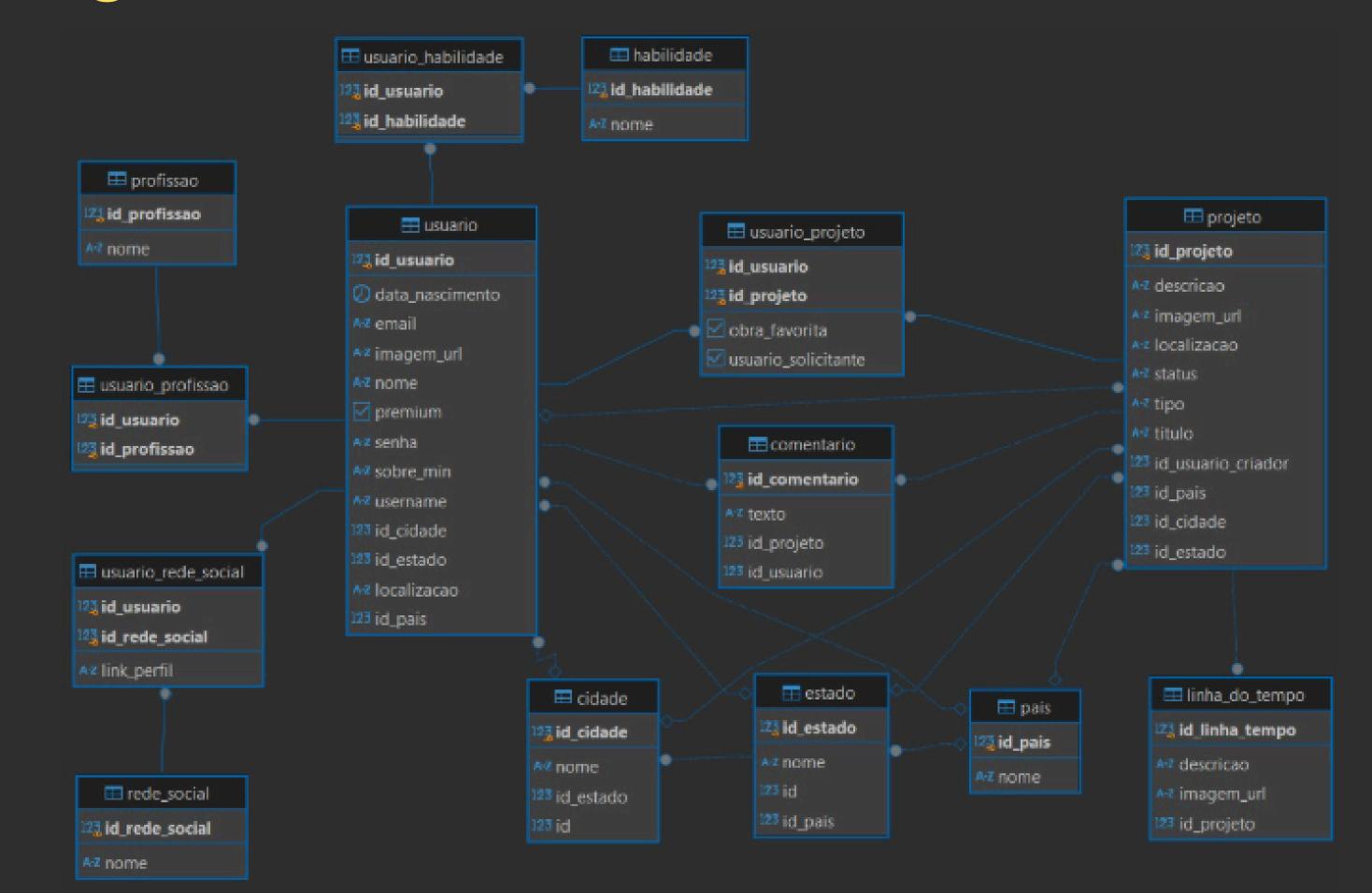
Model (Modelo de Dados)

Define a estrutura e as regras dos dados usados na aplicação, representando entidades do sistema.

Service (Camada de Serviço)

Contém a lógica de negócio da aplicação. Processa os dados e orquestra as operações entre as outras camadas.

Modelagem do Banco de Dados



VAMOS A APLICAÇÃO

OBRIGADOIII