

# Eclipse - Frontend

**Eclipse** é um software desenvolvido para a gestão completa do envio, avaliação e acompanhamento de relatórios no âmbito jurídico-administrativo. A plataforma oferece funcionalidades específicas para três tipos de usuários:

- **Defensores:** Criam e submetem triagens e peças processuais.
- **Avaliadores:** Avaliam peças e triagens enviadas.
- **Corregedor:** Gerencia usuários, aprovações e negações, além de supervisionar o sistema.

O sistema também oferece **notas automáticas**, critérios de avaliação e um fluxo para **solicitação de recursos**, promovendo transparência e controle.

---

## Tecnologias

- **Next.js** (App Router)
- **TypeScript**
- **Tailwind CSS**
- **NextAuth (Autenticação)**
- **Docker** (ambiente de desenvolvimento e produção) Usando Docker (ambiente de desenvolvimento)
- **HeroUI:** Biblioteca de componentes visuais modernos e acessíveis (botões, tabelas, modais, etc.), amplamente utilizada na interface do Eclipse para padronizar a experiência do usuário e facilitar/otimizar o desenvolvimento.
- **Tailwind CSS:** Utilizado em conjunto com HeroUI para estilização utilitária e responsiva.

---

## Estrutura de Pastas

```
eclipse/  
├─ app/ # Rotas e páginas (App Router)  
| ── api/auth/[...nextauth]/route.ts # Rota de autenticação (NextAuth)  
| ── registrar/ # Página de registro de usuários
```

— (user-routes)/	# Rotas específicas para usuários logados
— (usuario)/	# OTP e redefinição de senha
— components/	# Componentes utilizados no projeto
— avaliador/	# Componentes de tela do avaliador
— corregedor/	# Componentes de tela do corregedor
— defensor/	# Componentes de tela do defensor
— inicial/	# Componente de gerenciamento da Home
— sidebar/	# Componente de gerenciamento da troca entre telas
— providers/	# Contextos globais (auth, loading, sessão)
— HeroProvider	# Gerenciamento da biblioteca de User Interface que facilita a criação de componentes
— LoadingProvider	# Gerenciamento das funções de loading do software
— ScreenProvider	# Gerenciamento das trocas de componente em tela e seus parâmetros
— SessionProvider	# Gerenciamento das informações de sessão do usuário logado
— public/	# Arquivos estáticos (Imagens e Ícones)
— utils/	# Funções utilitárias
— NextAuthOptions	# Configurações de autenticação de usuário
— *.tsx	# Demais utilitários
— Dockerfile.*	# Dockerfiles para build e produção
— next.config.ts	# Configurações do Next.js
— tailwind.config.js	# Configurações do Tailwind CSS

# Autenticação

A autenticação é gerenciada por **NextAuth** com a configuração definida em:

```
app/api/auth/[...nextauth]/route.ts
```

Suporta autenticação por e-mail/senha com fluxo de redefinição de senha via token e OTP.

# Padrões de Código

- Componentes funcionais com Hooks.
  - Organização baseada em papéis de usuário.
  - ESLint configurado com suporte a TypeScript.
  - Estilização com Tailwind CSS.
  - Uso de **HeroUI** para componentes reutilizáveis.
- 

# Colaboração

## Branches

- `main`: Produção
  - `develop`: Desenvolvimento
  - `feat--*`: Novas funcionalidades
  - `fix--*`: Correções
- 

# Funcionalidades em Destaque

- Triagens e submissões por defensores.
  - Avaliação e notas automáticas.
  - Acompanhamento de histórico.
  - Fluxo de aprovação/negação com recurso e/ou usuários do sistema.
  - Sistema robusto de autenticação e segurança.
- 

# Equipe de Desenvolvimento

- **Desenvolvedores Fullstack:**
    - *João Vitor Viana Chaves*
    - *Gustavo Costa*
    - *Vinícius Jesus*
    - *Guilherme Barbosa*
  - **Coordenação Técnica:**
    - *Guilherme*
    - *Layon*
- 

Revision #4

Created 30 May 2025 11:10:52 by Acua

Updated 3 November 2025 17:54:29 by GuilhermeBDias