

# Controle de Processos Judiciais - Frontend (Angular)

---

Frontend em Angular para gerenciamento de processos judiciais integrado com a API .NET/C#.



## Funcionalidades

- ☒ Listar todos os processos judiciais com paginação e filtros
- ☒ Cadastrar novos processos com validação de campos
- ☒ Editar processos existentes
- ☒ Deletar processos
- ☒ Atualizar status do processo diretamente na listagem
- ☒ Busca por número do processo, autor ou réu
- ☒ Filtro por status
- ☒ Interface responsiva com Tailwind CSS
- ☒ Validação de campos no frontend
- ☒ Mensagens de sucesso e erro



## Tecnologias

- **Angular:** 18.0.0
- **TypeScript:** 5.4
- **Tailwind CSS:** 3.4.0
- **RxJS:** 7.8.0
- **Node.js:** 18+ (recomendado)



## Pré-requisitos

- Node.js 18.x ou superior
- npm 9.x ou superior
- Angular CLI 18.x
- O backend (.NET/C#) rodando em <http://localhost:5000>



## Instalação e Execução

### 1. Instalar dependências

```
cd controle-processos-client  
npm install
```

### 2. Configurar a URL da API (se necessário)

Abra o arquivo [src/app/services/processo.service.ts](#) e ajuste a URL da API:

```
private apiUrl = 'http://localhost:5000/processos';
```

### 3. Servir a aplicação

```
npm start
```

A aplicação estará disponível em <http://localhost:4200>

### 4. Build para produção

```
npm run build
```

Os arquivos otimizados serão gerados em [dist/controle-processos-client](#)

## Estrutura do Projeto

```
controle-processos-client/  
├── src/  
│   ├── app/  
│   │   ├── components/  
│   │   │   ├── processo-list/  
│   │   │   │   ├── processo-list.component.ts  
│   │   │   │   ├── processo-list.component.html  
│   │   │   │   └── processo-list.component.css  
│   │   ├── services/  
│   │   │   └── processo.service.ts  
│   │   ├── models/  
│   │   │   ├── processo.ts  
│   │   │   └── e-processo-status.ts  
│   │   └── app.component.ts  
│   ├── main.ts  
│   ├── index.html  
│   └── styles.css  
├── angular.json  
├── tsconfig.json  
├── package.json  
├── tailwind.config.js  
└── postcss.config.js
```

## Componentes

### ProcessoListComponent

Componente principal que implementa:

- **Listagem:** Exibe todos os processos em uma tabela responsiva
- **Busca:** Filtra por número, autor ou réu em tempo real
- **Filtro de Status:** Filtra processos por status
- **CRUD:** Formulário modal para criar/editar processos
- **Status Inline:** Dropdown para alterar status diretamente na tabela
- **Validação:** Validação de campos obrigatórios
- **Feedback:** Mensagens de sucesso e erro

## Integração com a API

O `ProcessoService` fornece os seguintes métodos:

```
// Listar todos os processos
getProcessos(): Observable<Processo[]>

// Obter processo por ID
getProcessoById(id: number): Observable<Processo>

// Criar novo processo
createProcesso(request: CreateProcessoRequest): Observable<Processo>

// Atualizar processo
updateProcesso(id: number, request: UpdateProcessoRequest):
Observable<Processo>

// Deletar processo
deleteProcesso(id: number): Observable<void>
```

## Modelos

### Processo

```
interface Processo {
  id: number;
  numeroProcesso: string;
  autor: string;
  reu: string;
  status: EProcessoStatus;
  descricao?: string;
  dataAjuizamento?: Date;
  dataInclusao: Date;
  historicos?: Historico[];
}
```

### EProcessoStatus

```
enum EProcessoStatus {  
  EmAndamento = 'Em Andamento',  
  Finalizado = 'Finalizado',  
  Arquivado = 'Arquivado',  
  Suspenso = 'Suspenso'  
}
```

## Usando com Docker

A aplicação pode ser servida dentro de um container Docker. Veja o [Dockerfile](#) na raiz do projeto para configuração completa.

## Resolução de Problemas

### Erro CORS

Se receber erro de CORS:

1. Certifique-se de que o backend está rodando em <http://localhost:5000>
2. Verifique se o backend tem CORS habilitado para <http://localhost:4200>

### API não responde

1. Verifique se o backend está rodando: `curl http://localhost:5000/processos`
2. Confirme a URL da API em `processo.service.ts`

### Tailwind CSS não funciona

1. Execute `npm install` novamente
2. Verifique se `tailwind.config.js` está correto
3. Limpe o cache: `rm -rf node_modules/.vite` e `npm run build`

## Documentação Adicional

- [Angular Documentation](#)
- [Tailwind CSS](#)
- [RxJS](#)

## Desenvolvedor

Desenvolvido como solução do desafio de Controle de Processos Judiciais.

## Licença

Projeto educacional - 2025