

Manual do Desenvolvedor - IGRP Workflow Studio

Este manual destina-se a desenvolvedores que pretendem contribuir, entender a arquitetura ou modificar o IGRP Workflow Studio.

1. Visão Geral do Projeto

O IGRP Workflow Studio é uma aplicação moderna para desenho e gestão de fluxos de trabalho, construída com React, TypeScript e BPMN.js.

1.1. Funcionalidades Principais

- Designer visual de processos BPMN
- Organização hierárquica com Aplicações, Áreas e SubÁreas
- UI moderna e responsiva construída com React e Tailwind CSS
- Desenvolvimento seguro com TypeScript (Type-safe)
- Arquitetura modular com estrutura monorepo

2. Começar (Ambiente de Desenvolvimento)

Siga estes passos para configurar o ambiente de desenvolvimento local:

1. **Instalar dependências:** `bash npm install`
2. **Iniciar o servidor de desenvolvimento:** `bash npm run dev`
3. **Abrir no navegador:** Acesse a <http://localhost:5173>.

3. Estrutura do Projeto

O projeto está organizado da seguinte forma:

```
packages/
├── igrp-wf-engine/      # Biblioteca principal do motor de workflow
│   ├── src/
│   │   ├── core/       # Lógica de negócio principal
│   │   ├── types/      # Definições de tipos TypeScript
│   │   └── utils/      # Funções utilitárias
│   └── package.json
└── igrp-wf-studio-ui/  # Aplicação UI baseada em React
    ├── src/
    │   ├── components/ # Componentes UI reutilizáveis
    │   ├── pages/      # Páginas da aplicação
    │   └── lib/         # Funções utilitárias
    └── package.json
```

4. Arquitetura Detalhada

4.1. Pacote do Motor (igrp-wf-engine)

O motor de workflow principal (igrp-wf-engine) é responsável por:

- Gestão de Aplicações
- Manuseamento de definições de processos

- Operações de sistema de ficheiros (inferido, pode necessitar de confirmação)
- Definições de tipos TypeScript

Este pacote contém a lógica central e as abstrações necessárias para o funcionamento dos fluxos de trabalho.

4.2. Pacote da UI (igrp-wf-studio-ui)

A interface do utilizador (igrp-wf-studio-ui) é construída com as seguintes tecnologias:

- **React:** Para a arquitetura de componentes.
- **Tailwind CSS:** Para estilização.
- **BPMN.js:** Para a modelação de processos BPMN.
- **React Router:** Para navegação entre páginas.

Este pacote foca-se na apresentação e interação com o utilizador, consumindo os serviços fornecidos pelo igrp-wf-engine.

5. Funcionalidades Centrais (Perspetiva Técnica)

5.1. Aplicações

- Permite a criação e gestão de aplicações de workflow.
- Organiza processos numa estrutura hierárquica.
- Rastreia o estado e metadados da aplicação.
- Do ponto de vista técnico, envolve modelos de dados para aplicações, APIs para CRUD (Create, Read, Update, Delete) e lógica para versionamento e ciclo de vida.

5.2. Áreas & SubÁreas

- Agrupam processos relacionados em áreas lógicas.
- Permitem a criação de uma organização aninhada com subáreas.
- Gerem configurações e permissões específicas da área.
- Tecnicamente, isto implica relações entre entidades (Aplicação -> Área -> SubÁrea -> Processo) e mecanismos de herança de permissões ou configurações.

5.3. Designer de Processos

- Designer visual de processos BPMN 2.0 integrado.
- Validação de processos em tempo real.
- Painel de propriedades para configuração de elementos BPMN.
- Funcionalidade de importação/exportação de diagramas BPMN.
- A integração com BPMN.js é crucial aqui, envolvendo a sua API para manipulação de diagramas, serialização para XML (BPMN) e a ligação de eventos do modelador com a lógica da aplicação (e.g., guardar alterações, popular painel de propriedades).

6. Desenvolvimento e Contribuições

6.1. Processo de Desenvolvimento

- O desenvolvimento é feito nos pacotes igrp-wf-engine e igrp-wf-studio-ui.
- Utilize os scripts NPM definidos no package.json raiz e nos package.json dos pacotes para construir, testar e executar a aplicação.
- Siga as convenções de código e estilo existentes.

6.2. Como Contribuir

1. Faça um Fork do repositório.
2. Crie uma branch para a sua funcionalidade (`git checkout -b feature/nova-funcionalidade`).
3. Faça commit das suas alterações (`git commit -am 'Adiciona nova funcionalidade'`).
4. Faça push para a branch (`git push origin feature/nova-funcionalidade`).
5. Crie um Pull Request.

7. Licença

Este projeto está licenciado sob a Licença MIT.