

Manual de **Informações Sistêmicas do SADI**

Responsável: PD Case Informática



iperon

RONDÔNIA
★
Governo do Estado



Sumário

1. Visão Geral da Aplicação	3
2. Banco de Dados	3
3. APIs / Endpoints	4
4. Configurações do Back-End.....	4
5. Configurações do Front-End.....	5
6. Arquitetura da Solução	5
7. Fluxo resumido da aplicação.....	6



Levantamento de Informações Sistêmicas SADI

Este documento apresenta o levantamento das informações sistêmicas referentes ao SADI, incluindo bancos de dados, APIs, configurações de front-end e back-end, além da arquitetura da solução.

1. Visão Geral da Aplicação

O SADI tem como objetivo disponibilizar arquivos em layout específico para importação no SISPREVWEB. A aplicação é composta por uma arquitetura moderna, baseada em .NET no back-end e React no front-end, com dados armazenados em SQL Server.

2. Banco de Dados

- Tipo: SQL Server
- Finalidade: Leitura da base de dados governa-esocial.
- Observações:
 - Integração direta com o back-end .NET
 - Suporta operações somente de leitura no banco SQL Server
 - Requer políticas de backup e restore para garantir continuidade do serviço

3. APIs / Endpoints

- Descrição: API desenvolvida em .NET, documentada via Swagger, responsável por disponibilizar os dados do portal.

1. Homologação/QA : <https://api-sadi.qa.iperon.ro.gov.br/swagger/index.html>
2. Desenvolvimento/DEV : <https://api-sadi.dev.iperon.ro.gov.br/swagger/index.html>

- Exemplos de endpoints:

1. <https://api-sadi.qa.iperon.ro.gov.br/exportacao-arquivo?MesCompetencia=07&AnoCompetencia=2025> → Consulta lista de arquivos.

4. Configurações do Back-End

- Tecnologia: .NET
- Padrão de API: REST, documentado com Swagger
- Integrações: Comunicação direta com o banco de dados SQL Server
- Configurações importantes:
 - Suporte a ambiente de desenvolvimento (DEV), homologação (QA) e produção
 - Logs de processamento disponíveis no HangFire, os arquivos são gerados em background e armazenados no S3 para download.
 - Bucket do S3 (sadi-dev)
 - Autenticação/Autorização pode ser configurada via JWT utilizando o padrão OAuth2



5. Configurações do Front-End

- Tecnologia: ReactJS
- Comunicação: Consome dados do back-end por meio das APIs REST
- Deploy: Aplicação front-end é hospedada separadamente do back-end
- Ambientes:
 - QA: conectado ao endpoint de homologação
 - Produção: conectado ao endpoint oficial (em fase de definição)

6. Arquitetura da Solução

- Camada de Front-End: Desenvolvida em React, responsável pela interface do usuário e exibição de dados
- Camada de API Back-End: Desenvolvida em .NET, expõe APIs REST para consumo de dados
- Camada de Dados: Banco de dados SQL Server, onde as informações do governa são consultadas



7. Fluxo resumido da aplicação

- Usuário (navegador) → Front-End (React) → Back-End (API .NET) → SQL Server:

