

# 1. Introdução



Este documento descreve a arquitetura do sistema **AcadMap**, abordando suas principais decisões estruturais, visões, preocupações dos stakeholders e justificativas técnicas. Ele está baseado nos princípios básicos da ISO/IEC 42010, com foco em clareza, rastreabilidade e evolução futura do sistema.

## 1.1 Objetivo

Apresentar a estrutura e organização da solução, documentar decisões arquiteturais e comunicar de forma clara entre todos os envolvidos.

## 1.2 Escopo

Sistema web para permitir o armazenamento, consulta, classificação e geração de relatórios sobre eventos e periódicos científicos, com base em regras atualizadas da CAPES, substituindo a antiga consulta ao Qualis e eliminando a necessidade de verificações manuais, com front-end React, back-end Spring Boot e banco de dados PostgreSQL.

## 1.3 Definições

- **AcadMap**: Nome do sistema representado nesta arquitetura. Plataforma para análise de produção científica baseada em periódicos e eventos.
- **MVP** (*Minimum Viable Product*): Versão mínima funcional do sistema, com as funcionalidades essenciais para validar a proposta.
- **API REST**: Interface de comunicação baseada em HTTP com operações sobre recursos, utilizando padrões como GET, POST, PUT e DELETE.
- **Stakeholder (ST)**: Pessoa, grupo ou organização com interesse direto ou indireto na definição, uso ou evolução do sistema. Referenciado por códigos **STxx** (ex: **ST01** – Desenvolvedor Backend).
- **Preocupação Arquitetural (P)**: Necessidade ou exigência relevante de um ou mais stakeholders que deve ser considerada na arquitetura. Referenciada por **Pxx** (ex: **P01** –

Clareza de código).

- **Ponto de Vista (PV):** Perspectiva sob a qual a arquitetura é descrita, tratando um conjunto de preocupações. Referenciado por `PVxx` (ex: `PV01` – Visão Lógica).
- **DAS:** Documento de Arquitetura de Software.
- **COMP-BACK-01, SEQU-MTRC-02**, etc.: Códigos de identificação dos diagramas utilizados ao longo do documento, seguindo o padrão `[TIPO] - [OBJETO] - [NÚMERO]`.
- **Controller / Service / Repository / Model:** Camadas da arquitetura backend seguindo o padrão MVC, responsáveis respectivamente por controlar requisições, encapsular regras de negócio, acessar dados e representar entidades de domínio.
- **Pages / Components / Services (frontend):** Diretórios lógicos do frontend React. `pages` contém telas principais, `components` agrupa componentes reutilizáveis, e `services` abstrai chamadas à API.

## Equipes

Gestão de Projeto



GESTÃO DE PROJETO

Arquitetura



ARQUITETURA

Requisitos



REQUISITOS

Desenvolvimento



DESENVOLVIMENTO

Testes



TESTES

---