

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - UTFPR  
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ARTHUR BASSANI, GABRIELA KOLBEN, JOÃO FACCIN, JOÃO ZILIO, JOSÉ  
METZDORF, RAFAEL VIER

**PROPOSTA DE ARQUITETURA TECNOLÓGICA**

MEDIANEIRA  
2025

## SUMÁRIO

1. INTRODUÇÃO	3
2. OBJETIVOS	3
3. STACK TECNOLÓGICA	3
3.1 BACK-END	3
3.2 FRONT-END	4
3.3 BANCO DE DADOS	4
4. AMBIENTE DE HOSPEDAGEM	5
4.1 FRONT-END	5
4.2 BACK-END	6
4.3 BANCO DE DADOS	6
4.4 DOMÍNIO	6
5. PROPOSTA FINAL	7
5.1 CUSTOS PREVISTOS	7
6. CONCLUSÃO	8

## 1. INTRODUÇÃO

Este documento tem como finalidade apresentar uma proposta de arquitetura tecnológica para o desenvolvimento e implantação do projeto da equipe. A proposta busca atender critérios de eficiência operacional, escalabilidade, viabilidade técnica e otimização de custos, considerando o cenário atual da equipe e as metas do projeto.

## 2. OBJETIVOS

O objetivo central é propor uma stack tecnológica e um plano de hospedagem que contemple:

- Eficiência operacional durante o desenvolvimento e a execução;
- Escalabilidade para sustentar o crescimento do sistema;
- Custos reduzidos e controlados, priorizando ferramentas gratuitas ou acessíveis;
- Viabilidade técnica, respeitando o nível de conhecimento da equipe.

## 3. STACK TECNOLÓGICA

### 3.1 BACK-END

#### Tecnologia Principal: Java Spring Boot

- **Nível de complexidade:** Médio
- **Justificativa:**
  - Framework robusto, amplamente utilizado para desenvolvimento de APIs RESTful.
  - Integração nativa com bancos de dados relacionais através do JPA/Hibernate.
  - Suporte avançado à arquitetura de microsserviços.
  - Comunidade ativa e extensa documentação.

#### Tecnologia Alternativa: Python Flask

- **Nível de complexidade:** Baixo/Médio
- **Justificativa:**
  - Framework minimalista e flexível, com curva de aprendizado reduzida.
  - Recomendado para projetos menores e MVPs.
  - Possui diversas extensões e boa integração com bibliotecas Python.

## 3.2 FRONT-END

### Tecnologia Principal: Next.js

- **Nível de complexidade:** Médio
- **Justificativa:**
  - Otimização nativa para SEO, ideal para projetos com exposição pública.
  - Excelente performance com renderização híbrida (SSR/SSG).
  - Forte integração com o ecossistema React.

### Tecnologia Alternativa: React.js + Vite

- **Nível de complexidade:** Baixo/Médio
- **Justificativa:**
  - Rápida inicialização e alto desempenho no desenvolvimento.
  - Compatibilidade com o código React existente.
  - Ideal para aplicações SPA simples e de fácil manutenção.

## 3.3 BANCO DE DADOS

### Tecnologia Principal: PostgreSQL

- **Nível de complexidade:** Médio
- **Justificativa:**
  - Banco de dados relacional robusto e open-source.
  - Suporte a dados estruturados e semiestruturados (JSON).
  - Alta escalabilidade em ambientes cloud.
  - Amplamente adotado em sistemas corporativos.

### **Tecnologia Alternativa: MongoDB**

- **Nível de complexidade:** Baixo/Médio
- **Justificativa:**
  - Banco de dados NoSQL com estrutura flexível.
  - Alta performance para operações de leitura e escrita.
  - Facilidade de integração com diversas linguagens, inclusive com Spring Boot.

## **4. AMBIENTE DE HOSPEDAGEM**

### **4.1 FRONT-END**

- **Provedor Principal:** Vercel
  - **Custo:** Gratuito (até 100GB/mês)
  - **Vantagens:** Deploy contínuo, integração direta com GitHub, suporte a Next.js.
- **Provedor Alternativo:** Netlify
  - **Custo:** Gratuito (até 100GB/mês)

- **Vantagens:** Integração com Git, facilidade de uso e documentação acessível.

## 4.2 BACK-END

- **Provedor Principal:** Hostinger
  - **Custo:** R\$ 36,99/mês após período gratuito
  - **Vantagens:** Interface amigável, bom suporte a aplicações Java.
- **Provedor Alternativo:** Railway
  - **Custo:** A partir de US\$ 20/mês (~R\$ 120,00)
  - **Vantagens:** Deploy automatizado com GitHub, escalável, integração fácil com bancos de dados.

## 4.3 BANCO DE DADOS

- **Provedor Principal:** Supabase
  - **Custo:** Gratuito (500MB de dados + 2 bancos)
  - **Limitações:** 500MB de armazenamento, 50 mil linhas por projeto gratuito
  - **Vantagens:** Interface moderna, REST API pronta, autenticação integrada.
- **Provedor Alternativo:** Elephant SQL
  - **Custo:** R\$ 75,00/mês no plano padrão
  - **Vantagens:** Solução gerenciada de PostgreSQL com backups automáticos e segurança robusta.

## 4.4 DOMÍNIO

- **Domínio Registrado:** sanem.com.br

- **Custo:** R\$29,00 no primeiro ano; renovação anual por R\$59,99.

## 5. PROPOSTA FINAL

### Stack Tecnológica Recomendada

- Front-end: Next.js (React)
- Back-end: Java Spring Boot
- Banco de Dados: PostgreSQL

### Hospedagem

- Front-end: Vercel (gratuito)
- Back-end: Hostinger (R\$ 36,99/mês)
- Banco de Dados: Supabase (gratuito)
- Domínio: Registro BR (R\$ 29,00 no primeiro ano)

### 5.1 CUSTOS PREVISTOS

Componente	Custo Mensal Estimado
Front-end	Gratuito
Back-end	R\$36,99
Banco de Dados	Gratuito
Domínio (.com.br)	R\$2,40 (média mensal)
Total Aproximado	R\$39,40/mês

## **6. CONCLUSÃO**

A arquitetura tecnológica proposta oferece um equilíbrio estratégico entre custo, viabilidade técnica e escalabilidade. A escolha de tecnologias robustas e modernas, aliada a serviços de hospedagem acessíveis e com boa documentação, permite à equipe implementar tanto um MVP funcional quanto uma aplicação escalável para ambientes de produção. O uso de ferramentas com planos gratuitos atende às necessidades iniciais, com possibilidade de evolução conforme a demanda do projeto.