

# Arquitetura e Stack Tecnológico para o Portal de Licitação

## Sumário

1. [Introdução](#)
2. [Visão Geral da Arquitetura](#)
3. [Stack Tecnológico](#)
4. [Componentes do Sistema](#)
5. [Integração com Sistemas Externos](#)
6. [Segurança](#)
7. [Plano de Desenvolvimento](#)
8. [Alternativas Consideradas](#)
9. [Conclusão](#)

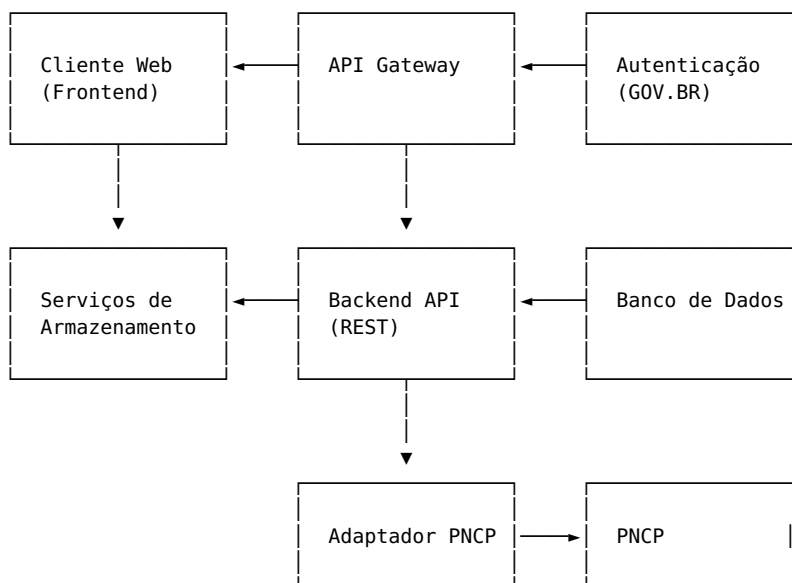
## Introdução

Este documento apresenta a arquitetura e o stack tecnológico propostos para o desenvolvimento do portal de licitação destinado a câmaras de vereadores e prefeituras da região oeste da Bahia. A escolha das tecnologias e da arquitetura foi baseada nos requisitos legais estabelecidos pela Lei nº 14.133/2021, nos requisitos funcionais identificados na análise de portais existentes e nos requisitos técnicos para integração com o Portal Nacional de Contratações Públicas (PNCP).

O prazo de desenvolvimento de 30 dias foi um fator determinante nas escolhas tecnológicas, priorizando soluções que permitam um desenvolvimento rápido sem comprometer a qualidade, segurança e conformidade legal do sistema.

## Visão Geral da Arquitetura

A arquitetura proposta para o portal de licitação segue o padrão de aplicação web moderna, com separação clara entre frontend e backend, comunicação via API REST, e integração com sistemas externos. A arquitetura é composta pelos seguintes componentes principais:



Esta arquitetura proporciona:

1. **Separação de Responsabilidades:** Frontend e backend separados, facilitando o desenvolvimento paralelo e a manutenção.
2. **Escalabilidade:** Componentes podem ser escalados independentemente conforme a demanda.
3. **Segurança:** Camadas de autenticação e autorização bem definidas.
4. **Interoperabilidade:** Integração com sistemas externos através de APIs padronizadas.
5. **Manutenibilidade:** Estrutura modular que facilita atualizações e correções.

## Stack Tecnológico

Após análise dos requisitos, prazos e considerações técnicas, recomendamos o seguinte stack tecnológico:

### Linguagem de Programação e Framework Backend

#### Python com Django REST Framework

Justificativa: - Desenvolvimento rápido e produtivo, ideal para o prazo de 30 dias - Ecossistema maduro com bibliotecas para todas as necessidades do projeto - Excelente suporte para APIs REST - ORM poderoso que simplifica operações com banco de dados - Bibliotecas disponíveis para integração com GOV.BR e APIs externas - Comunidade ativa e ampla documentação - Facilidade de encontrar desenvolvedores com experiência

Componentes específicos: - Django 4.2+ como framework web principal - Django REST Framework para criação de APIs - Django Channels para funcionalidades em tempo real (chat, notificações) - Celery para processamento assíncrono e tarefas agendadas

### Framework Frontend

#### React com Next.js

Justificativa: - Renderização do lado do servidor (SSR) para melhor SEO e performance - Desenvolvimento rápido com componentes reutilizáveis - Excelente experiência do usuário com atualizações dinâmicas - Ampla disponibilidade de bibliotecas de componentes UI - Facilidade de integração com APIs REST - Suporte a TypeScript para maior segurança de tipos

Componentes específicos: - Next.js 13+ como framework React - Material UI ou Chakra UI para componentes de interface - React Query para gerenciamento de estado e cache - React Hook Form para formulários

### Banco de Dados

#### PostgreSQL

Justificativa: - Sistema de banco de dados relacional robusto e maduro - Excelente suporte para transações ACID, essencial para operações financeiras - Suporte nativo a JSON e JSONB para dados semiestruturados - Extensões geográficas para funcionalidades de localização - Excelente integração com Django ORM - Licença open-source sem restrições comerciais - Ampla disponibilidade de hospedagem em provedores cloud

### Infraestrutura e Implantação

## **Docker + Docker Compose**

Justificativa: - Ambiente de desenvolvimento consistente - Facilidade de implantação em diferentes ambientes - Isolamento de componentes - Escalabilidade horizontal simplificada - Integração com CI/CD

## **Serviços de Armazenamento**

- Amazon S3 ou equivalente para armazenamento de documentos e arquivos
- Redis para cache e filas de mensagens

# **Componentes do Sistema**

O sistema será dividido nos seguintes componentes principais:

## **1. Módulo de Autenticação e Autorização**

- Integração com GOV.BR para autenticação de usuários
- Gerenciamento de perfis e permissões
- Controle de acesso baseado em papéis (RBAC)

## **2. Módulo de Gestão de Licitações**

- Criação e gerenciamento de processos licitatórios
- Suporte a todas as modalidades previstas na Lei 14.133/2021
- Fluxos de aprovação e publicação
- Gestão de documentos e anexos

## **3. Módulo de Fornecedores**

- Cadastro e gestão de fornecedores
- Envio de propostas
- Participação em sessões públicas
- Gestão de contratos

## **4. Módulo de Integração com PNCP**

- Sincronização de dados com o PNCP
- Gerenciamento de tokens JWT
- Validação e transformação de dados
- Logs de operações

## **5. Módulo de Relatórios e Estatísticas**

- Geração de relatórios gerenciais
- Dashboards de indicadores
- Exportação de dados em diferentes formatos

## **6. Módulo de Gestão de Contratos**

- Elaboração e registro de contratos
- Controle de vigência e valores
- Gestão de aditivos e apostilamentos
- Avaliação de fornecedores

# **Integração com Sistemas Externos**

## **Integração com GOV.BR**

A integração com o GOV.BR para autenticação será implementada utilizando o protocolo OAuth 2.0/OpenID Connect. O fluxo de autenticação seguirá os seguintes passos:

1. Redirecionamento do usuário para o portal GOV.BR
2. Autenticação do usuário no GOV.BR
3. Redirecionamento de volta ao portal com código de autorização
4. Troca do código por token de acesso
5. Validação do token e obtenção de informações do usuário

## **Integração com PNCP**

A integração com o PNCP será implementada conforme os requisitos técnicos identificados no documento de requisitos de integração. Principais aspectos:

1. **Autenticação:** Implementação de mecanismo para obtenção e renovação de tokens JWT
2. **Sincronização de Dados:** Envio automático de informações sobre licitações, contratos e documentos
3. **Validação de Dados:** Garantia de conformidade com os requisitos do PNCP
4. **Tratamento de Erros:** Mecanismos robustos para lidar com falhas de comunicação
5. **Logs:** Registro detalhado de todas as operações para auditoria

## **Segurança**

A segurança do sistema será garantida através das seguintes medidas:

### **1. Autenticação e Autorização**

- Integração com GOV.BR para autenticação segura
- Implementação de controle de acesso baseado em papéis (RBAC)
- Tokens JWT com tempo de expiração curto
- Renovação segura de tokens

### **2. Proteção de Dados**

- Criptografia de dados sensíveis em repouso e em trânsito
- Implementação de HTTPS em todas as comunicações
- Sanitização de inputs para prevenção de injeção SQL e XSS
- Validação rigorosa de dados de entrada

### **3. Auditoria e Logs**

- Registro detalhado de todas as operações críticas
- Logs de acesso e alterações
- Trilhas de auditoria para operações sensíveis
- Monitoramento de atividades suspeitas

### **4. Conformidade com LGPD**

- Implementação de mecanismos para consentimento de uso de dados
- Funcionalidades para exclusão e portabilidade de dados
- Documentação clara sobre tratamento de dados pessoais

## **Plano de Desenvolvimento**

Considerando o prazo de 30 dias, propomos o seguinte plano de desenvolvimento:

## Semana 1: Configuração e Estrutura Básica

- Configuração do ambiente de desenvolvimento
- Implementação da estrutura básica do backend (Django)
- Implementação da estrutura básica do frontend (Next.js)
- Configuração do banco de dados e migrações iniciais
- Implementação da integração com GOV.BR

## Semana 2: Funcionalidades Essenciais

- Implementação do módulo de gestão de licitações
- Implementação do módulo de fornecedores
- Desenvolvimento das interfaces de usuário principais
- Implementação de fluxos básicos de licitação

## Semana 3: Integração e Funcionalidades Avançadas

- Implementação da integração com PNCP
- Desenvolvimento do módulo de gestão de contratos
- Implementação de funcionalidades avançadas (diligências, sessões públicas)
- Testes de integração

## Semana 4: Finalização e Implantação

- Testes finais e correção de bugs
- Otimização de performance
- Documentação do sistema
- Implantação em ambiente de produção
- Treinamento inicial de usuários

## Alternativas Consideradas

### PHP com Laravel

**Prós:** - Desenvolvimento rápido - Ampla disponibilidade de hospedagem - Ecossistema maduro - Eloquent ORM poderoso

**Contras:** - Menor robustez para aplicações complexas - Desempenho inferior em comparação com Python/Django para operações intensivas - Menor suporte para processamento assíncrono

### Node.js com Express

**Prós:** - JavaScript em toda a stack - Excelente desempenho para operações I/O - Amplo ecossistema de pacotes

**Contras:** - Maior complexidade na estruturação de aplicações grandes - Callback hell e problemas de manutenção em projetos complexos - Menor maturidade de ORMs em comparação com Django

### Bancos de Dados Alternativos

**MySQL:** - Amplamente utilizado, mas com menos recursos avançados que PostgreSQL - Menor suporte para JSON e dados geoespaciais

**MongoDB:** - Excelente para dados não estruturados - Menos adequado para dados relacionais complexos e transações ACID

## Conclusão

A arquitetura e stack tecnológico propostos (Python/Django + React/Next.js + PostgreSQL) oferecem o melhor equilíbrio entre velocidade de desenvolvimento, robustez e conformidade com os requisitos legais e técnicos do portal de licitação. A escolha dessas tecnologias permitirá o desenvolvimento do sistema no prazo de 30 dias, garantindo a qualidade, segurança e usabilidade necessárias.

Recomendamos a adoção desta arquitetura e stack tecnológico, com foco inicial nas funcionalidades essenciais para garantir a conformidade com a Lei nº 14.133/2021 e a integração com o PNCP. Funcionalidades adicionais podem ser implementadas em fases posteriores, após a entrega inicial do sistema.

A implementação deve seguir as melhores práticas de desenvolvimento, com ênfase em testes automatizados, documentação clara e código limpo, para garantir a manutenibilidade e evolução do sistema a longo prazo.