

Documento de referência da Arquitetura Tecnológica Para o Desenvolvimento de Soluções de TIC da SEFAZ/PE

GGTI - Gerência Geral de Tecnologia da informação

Sumário

Introdução.....	4
Aplicações.....	5
Ferramentas:	5
Infraestrutura:	6
Frameworks:.....	6
Linguagens:	6
Softwares de apoio:	6
Padrões de API:.....	6
CI/CD:.....	7
Banco de Dados.....	7
Big Data.....	8
Ingestão	8
Processamento	8
Armazenamento	8
Orquestração	8
Catalogação	8
Qualidade de dados	8
Exploração	8
Disponibilização	9
Machine Learning.....	9
Visualização	9

Introdução

Este documento tem como objetivo orientar equipes internas e externas que pretendam desenvolver soluções para a SEFAZ-PE, destacando as tecnologias, arquitetura, ferramentas e padrões adotados como referência.

A solução desenvolvida deve ser compatível com a infraestrutura computacional interna da SEFAZ-PE, podendo ser absorvida pelo seu corpo técnico de tecnologia da informação e repassada conforme definição da SEFAZ-PE.

Para informações protegidas por sigilo fiscal, a equipe do projeto deverá responsabilizar-se pelas ações recomendadas pela SEFAZ-PE, pelos efeitos delas advindos na execução das atividades do projeto e pelo uso dos acessos, privilégios e informações obtidas em função das atividades executadas.

Qualquer desenvolvimento e implementação de soluções devem ter como fundamentos:

- A legalidade, a imensoalidade, a moralidade, a publicidade e a eficiência, em consonância com o art. 37 da Constituição Federal;
- A proteção de dados objeto de sigilo fiscal, previsto no art. 198 da Lei nº 5.172, de 25 de outubro de 1966 (Código Tributário Nacional);
- A proteção de dados pessoais, previstos na Lei nº 13.709, de 14 de agosto de 2018 (Lei Geral de Proteção de Dados Pessoais – LGPD), e o direito de acesso à informação;

Quanto as soluções de IA, observar o disposto na PORTARIA SF Nº 68, DE 06.05.2025.

O uso dessas tecnologias é recomendado visando garantir padronização, segurança, escalabilidade, observabilidade e facilidade de manutenção. Soluções que se desviem significativamente dessas diretrizes devem ser justificadas tecnicamente e submetidas à avaliação da GGTI.

Aplicações

A SEFAZ-PE adota uma arquitetura de sistemas baseada em microserviços, onde cada componente do sistema é desenvolvido, implantado e escalado de forma independente. Esses serviços são executados em contêineres gerenciados pelo OpenShift, promovendo alta disponibilidade e flexibilidade.

As aplicações são desenvolvidas com princípios de Cloud Native, utilizando práticas de DevOps, como automação de deploys, observabilidade, autenticação federada e integração contínua. O uso de GitOps garante que o estado dos ambientes esteja sempre sincronizado com os repositórios de código.

A "arquitetura mínima" adotada pela SEFAZ-PE é composta por três camadas principais: Front-end (**Angular**), BFF - Back-end for Front-end (**Spring**) e BS - Serviços de Negócio (**Spring**). Essa estrutura promove separação de responsabilidades, facilita a manutenção e permite escalabilidade dos sistemas.

1. Camada Front-end - Angular. Responsável pela interface com o usuário, apresentando dados e recebendo entradas, não contém regras de negócio, toda orquestração é delegada ao BFF, mantendo o front-end focado na experiência visual.
2. Camada Back-end for Front-end (BFF) - Spring. Camada intermediária entre o front-end e os serviços de negócio. Agrega dados, formata respostas e implementa regras de segurança específicas para a interface.
3. Camada de Serviços de Negócio (BS) - Spring. Implementa a lógica de negócio, acessa bancos de dados e fornece APIs. Cada serviço é coeso e independente, promovendo modularidade e reutilização.

Ferramentas:

- **Eclipse** - IDE para desenvolvimento das aplicações legadas
- **VsCode** - IDE para desenvolvimento das novas aplicações
- **Git/GitLab** – Repositório de código e plataforma **DevOps**.
- **Maven** – Gerenciador de dependências e builds.
- **SonarQube / SonarLint** – Inspeção de qualidade e segurança de código.
- **Nexus** – Repositório de artefatos binários.
- **ArgoCD** – Entrega contínua com **GitOps**.
- **Prometheus / Grafana / Kibana / Jaeger** – Observabilidade (métricas, logs, traces).
- **Opengrok** – Indexação e busca de código-fonte.
- **Teams** – Canal de suporte técnico e comunicação com desenvolvedores.
- **WSL**: Ambiente virtualizado Linux integrado ao Windows

Infraestrutura:

- **OpenShift** – Plataforma de orquestração de contêineres (Kubernetes).
- **RHV** (Red Hat Virtualization) – Virtualização de servidores.
- **TrueNAS** – Armazenamento centralizado em rede.
- **ElasticSearch** – Gerenciamento e busca em logs.
- **Istio / Kiali** – Malha de serviço e observabilidade de tráfego.
- **RH AMQ Broker / RH AMQ Streams (Kafka)** – Mensageria e processamento assíncrono (streams).
- **RH SSO** – Autenticação e autorização federada.
- **RH JBoss EAP** – Servidor de aplicação Java EE.
- **IBM Websphere** – Servidor de aplicação Java EE para aplicações legadas.
- **Tomcat** – Servidor de aplicação web Java
-

Frameworks:

- **Angular** - Framework padrão para aplicações frontend.
- **Spring Boot** – Framework padrão para aplicações backend.
- **sefaz-spring-boot-starter-parent** – Projeto base (biblioteca) com configurações de segurança, observabilidade e boas práticas.
- **Arquétipos Maven** – Templates para backend, BFF e jobs.
- **BFF** (Backend for Frontend) – Padrão para integração entre frontend e backend.

Linguagens:

- **Java** – Linguagem principal para backend.
- **JavaScript/TypeScript** – Para desenvolvimento frontend.
- **YAML** – Configuração declarativa de ambientes e aplicações (GitOps).
- **SQL** – Para integração com bancos de dados legados.

Softwares de apoio:

- **MediaWiki** – Portais de conhecimento e documentação.
- **FindBugs / PMD** – Ferramentas de análise estática de código.
- **Assinador Digital** – Validação de documentos e autenticação.
- **SCA / FRW** – Componentes do framework legado do e-Fisco.

Padrões de API:

- **RESTful APIs com formato JSON.**
- **Autenticação via RH SSO (ou GOV.BR)**
- Documentação preferencial via **Swagger/OpenAPI**.
- Integração com sistemas legados via APIs padronizadas e seguras.

- Uso de versionamento e controle de acesso por perfil.

CI/CD:

- **GitLab CI/CD** – Pipelines automatizados para build, testes, deploy e backup.
- **GitOps com ArgoCD** – Deploys automatizados e auditáveis.
- Monitoramento contínuo com **Prometheus**, **Grafana** e alertas.
- Backups automatizados e versionamento de configurações.

Banco de Dados

- **IBM DB2**
- **SQL Server**

Big Data

A infraestrutura de Big Data da SEFAZ/PE atualmente inclui um cluster Hadoop Cloudera CDP Private Base 7.1.9 para armazenamento, tratamento e processamento de dados, abrangendo ambientes de produção e desenvolvimento e os componentes mencionados a seguir:

Ingestão

IBM DataStage - ferramenta de integração de dados responsável por realizar a extração e carga. Sua função principal na arquitetura é conectar-se às diversas fontes de dados, extrair as informações e carregá-las na Camada Landing do HDFS.

Processamento

Apache Spark - Core do processamento de dados na arquitetura. Framework de computação distribuída de alta performance, utilizado para executar todas as transformações de dados em larga escala. Os códigos Spark (ETL) são gerenciados em repositórios GitLab dedicados, seguindo as diretrizes de governança para versionamento e CI/CD.

Armazenamento

Apache Hadoop HDFS - O Apache Hadoop HDFS (Hadoop Distributed File System) é o sistema de arquivos distribuído que serve como base para o datalake, oferecendo armazenamento escalável e tolerante a falhas.

Orquestração

Apache Airflow - Gerencia o ciclo de vida dos pipelines de dados, definindo a sequência de execução, dependências, agendamentos e tratamento de falhas.

Catalogação

Apache Atlas - Catálogo de metadados central, descobrindo e registrando os ativos de dados e sua linhagem.

Qualidade de dados

Great Expectations - utilizada para validar a qualidade e a integridade dos dados.

Exploração

- **DBeaver**: Um cliente SQL universal que permite conectar-se ao Apache Impala para executar consultas ad-hoc.

- **Apache Hue** (Hadoop User Experience): Uma interface web que facilita a interação com o ecossistema Hadoop
- **JupyterLab**: Instâncias de Jupyter Notebooks .

Disponibilização

Apache Impala - Motor de consulta SQL massivamente paralelo (MPP) que atua como a principal camada de disponibilização de dados para análises interativas.

Machine Learning

- **IBM SPSS Modeler**: Uma plataforma visual de data science que permite construir e treinar modelos preditivos através de uma interface gráfica e fluxos de trabalho intuitivos.
- **JupyterLab**: Fornece um ambiente baseado em código para desenvolvimento de Machine Learning, para criar, treinar e avaliar modelos complexos, integrados ao Apache Spark.

Visualização

A camada de visualização é a interface final onde os dados processados se transformam em insights para os usuários de negócios:

- **Angular**: Aplicação web customizada, que oferece visualizações e interações específicas para os processos de negócio da SEFAZ-PE
- **Qlik Sense**: Ferramentas de Business Intelligence (BI), utilizadas para criar dashboards interativos, relatórios gerenciais e análises visuais.