Arquitectura Aberta para um Portal do Estado com Acesso Massivo

White paper detalhado sobre plataforma, ferramentas abertas, segurança, actualizações e operação em produção

Versão 1.0 • 17 de Abril de 2026

Este documento propõe uma arquitectura tecnológica aberta, escalável e auditável para um portal digital do Estado, capaz de suportar acesso massivo, integração entre organismos, evolução contínua do software e requisitos exigentes de segurança, resiliência e governação.

Objectivo	**Definir uma plataforma aberta para serviços públicos digitais de grande escala.**
Âmbito	Portal público, área autenticada, APIs, integração, operação, segurança e roadmap.
Abordagem	Tecnologia aberta, arquitectura modular, GitOps, observabilidade e segurança por defeito.

Resumo executivo

Um portal do Estado com tráfego elevado não deve ser concebido como uma aplicação única e rígida, mas como uma plataforma digital modular, observável e continuamente actualizável. A arquitectura recomendada neste white paper assenta em padrões abertos e em componentes amplamente adoptados no ecossistema cloud native: Kubernetes como camada de execução; Gateway API como modelo de entrada e exposição de serviços; Cilium para rede, políticas e observabilidade; Keycloak para identidade; PostgreSQL para dados transaccionais; OpenSearch para pesquisa e análise; Prometheus e Grafana para monitorização; e Argo CD como pilar GitOps para actualizações controladas.

A proposta privilegia software aberto por cinco razões centrais: soberania tecnológica, auditabilidade, independência face a fornecedores, flexibilidade de integração e previsibilidade de custos a longo prazo. Em contexto estatal, estas qualidades não são um luxo: são matéria de estratégia, continuidade do serviço e defesa do interesse público.

A recomendação nuclear é simples: separar claramente a camada pública de apresentação, a camada de APIs e serviços, a camada de dados, a camada de observabilidade e a camada de segurança. Cada uma deve evoluir com ciclos próprios, contratos bem definidos e mecanismos de rollback. O objectivo não é apenas aguentar picos de acesso; é garantir que as actualizações regulares, os novos serviços e as mudanças de política pública possam ser absorvidos sem instabilidade sistémica.

Recomendação de alto nível

Camada	Escolha recomendada	Justificação
Execução	Kubernetes	Escalabilidade, auto-recuperação, rolling updates e forte ecossistema.
Exposição	Gateway API + controlador compatível	Modelo moderno e extensível, mais claro do que Ingress.
Rede/segurança	Cilium	Rede e políticas com eBPF, boa observabilidade e ciclo activo de manutenção.
Identidade	Keycloak	SSO, OIDC, OAuth2, MFA e federação.
Dados	PostgreSQL	Robustez transaccional, HA e maturidade.
Pesquisa	OpenSearch	Pesquisa textual, observabilidade e análise.
Operação	Prometheus + Grafana + Argo CD	Monitorização, alertas e actualizações declarativas.

1. Requisitos estruturantes do portal

- Disponibilidade elevada e capacidade para suportar variações bruscas de carga, incluindo períodos de picos sazonais, campanhas públicas e eventos críticos.
- Separação rigorosa entre front-end, APIs, integração e persistência de dados, evitando acoplamentos que dificultem a evolução.

- Autenticação forte, autorização granular e trilhos de auditoria completos, tanto para cidadãos como para funcionários e integrações máquina-a-máquina.
- Processo de actualização contínua, reversível e auditável, eliminando deploys manuais e alterações não rastreadas.
- Observabilidade completa do sistema, incluindo métricas, logs, tracing e indicadores de experiência do utilizador.
- Portabilidade entre ambientes e neutralidade tecnológica suficiente para reduzir dependências excessivas de um único fornecedor.

2. Princípios de desenho

A plataforma deve ser API-first. Isto significa que os serviços nucleares do portal são expostos de forma consistente por APIs documentadas e versionadas, permitindo reutilização por aplicações web, móveis, balcões digitais, integrações entre organismos e ferramentas internas.

A arquitectura deve ser modular, mas não fragmentada. Modularidade não significa multiplicar micro-serviços sem critério; significa definir fronteiras claras de domínio e separar aquilo que precisa de escalar, ser protegido ou evoluir de forma autónoma.

A segurança deve ser assumida como propriedade de desenho, não como camada posterior. A rede, as imagens de container, as credenciais, os segredos, a autenticação e as pipelines de entrega devem ser tratados como elementos de segurança desde o início.

3. Arquitectura de referência proposta

A arquitectura recomendada organiza-se em seis planos: experiência digital, entrada e entrega, serviços de negócio, integração e eventos, dados e pesquisa, e operação e segurança.

Plano	Componentes	Função
Experiência digital	Portal web, SPA/SSR, design system, CDN	Disponibilizar serviço público digital rápido, acessível e coerente.
Entrada e entrega	Gateway API, load balancers, certificados, rate limiting, WAF	Receber tráfego e aplicar políticas de exposição e segurança.
Serviços de negócio	APIs, serviços de domínio, filas, workflows	Executar lógica funcional e processos administrativos.
Integração e eventos	Mensageria, adaptadores, barramento de eventos	Articular organismos, sistemas legados e tarefas assíncronas.
Dados e pesquisa	PostgreSQL, Redis, OpenSearch, object storage	Persistência transaccional, cache, pesquisa e documentos.
Operação e segurança	Prometheus, Grafana, logs, tracing, Keycloak, Vault, Argo CD	Monitorizar, proteger e actualizar a plataforma.

3.1 Diagrama lógico textual

```
Cidadão / empresa / funcionário

   ↓

CDN / anti-DDoS / WAF / TLS termination
```

```
   ↓
Gateway API / políticas de tráfego / autenticação preliminar
   ↓
Front-end web  ⇔  APIs de serviços  ⇔  serviços de integração / eventos
   ↓                     ↓                     ↓
Keycloak            PostgreSQL / Redis     Mensageria / adaptadores
   ↓                     ↓                     ↓
Auditoria         OpenSearch / object storage / relatórios
   ↓
Prometheus / Grafana / alertas / GitOps / SIEM
```

4. Stack tecnológica recomendada

A stack abaixo foi seleccionada por maturidade, ecossistema, clareza operacional e adequação a contexto estatal. Não é a única possível, mas oferece um ponto de equilíbrio robusto entre flexibilidade, segurança e capacidade de evolução.

Área	Escolha principal	Notas	Prioridade
Front-end	React + Next.js + TypeScript	Bom desempenho, SSR/ISR, ecossistema vasto, forte reutilização de componentes.	Alta
Back-end core	Java + Spring Boot	Adequado a equipas grandes, integrações complexas e domínios críticos.	Alta
Serviços rápidos	Python + FastAPI	Excelente produtividade para APIs satélite, automação e serviços leves.	Média
Execução	Kubernetes	Orquestração, escalabilidade, isolamento e resiliência.	Alta
Exposição	Gateway API	Modelo moderno de entrada e políticas de tráfego.	Alta
Rede	Cilium	Rede, políticas e observabilidade com eBPF.	Alta
Identidade	Keycloak	SSO, MFA, OIDC, OAuth2, federação.	Alta
Dados	PostgreSQL	Base transaccional principal.	Alta
Cache	Redis	Sessões, cache e filas curtas.	Média
Pesquisa	OpenSearch	Pesquisa full-text, logs e análise.	Alta

Observabilidade	Prometheus + Grafana	Métricas, dashboards e alertas.	Alta
Entrega	Argo CD	GitOps, rastreabilidade, rollback.	Alta
Segredos	Vault ou equivalente	Gestão segura de credenciais e segredos.	Alta

5. Segurança por defeito

Num portal estatal, a segurança não pode depender apenas de perímetro. A plataforma deve assumir que haverá tráfego malicioso, configurações erradas, credenciais expostas, dependências vulneráveis e tentativas de abuso da própria lógica funcional. Por isso, a arquitectura deve combinar defesa em profundidade com mínima confiança implícita.

A gestão de identidade deve centralizar autenticação, MFA, federação e autorização. A gestão de segredos deve ser separada do código e das imagens. A rede entre serviços deve ser regida por políticas explícitas. As imagens de container devem ser assinadas e validadas. As pipelines devem gerar provas mínimas de proveniência. A observabilidade deve incluir eventos de segurança e trilhos de auditoria utilizáveis em investigação.

Domínio	Medidas recomendadas	Resultado esperado
Autenticação	Keycloak, MFA, políticas de sessão, OIDC, passkeys onde aplicável	Redução do risco de abuso de contas.
Autorização	RBAC/ABAC, privilégios mínimos, segregação de funções	Menor blast radius e melhor auditoria.
Segredos	Vault, rotação, não inclusão em imagens nem repositórios	Diminuição do risco de exposição de credenciais.
Rede	Network policies, segmentação, mTLS em fluxos sensíveis	Contenção lateral e controlo do tráfego leste-oeste.
Supply chain	Assinatura de imagens, SBOM, scans de vulnerabilidades	Maior confiança em artefactos e actualizações.
Auditoria	Logs imutáveis, correlação de eventos, retenção adequada	Capacidade de análise forense e conformidade.

6. Modelo de actualização e entrega contínua

As actualizações do portal devem ser declarativas, repetíveis e reversíveis. GitOps é a abordagem mais consistente para este cenário: o estado desejado do sistema é descrito em repositórios versionados; as alterações passam por revisão, validação e aprovação; e a sincronização com produção é efectuada por um agente dedicado, com trilho completo de auditoria.

A plataforma deve privilegiar imagens imutáveis, configurações versionadas e estratégias de rollout como blue/green, canary e feature flags. Sempre que possível, as migrações de base de dados devem ser compatíveis com versões adjacentes, evitando janelas de indisponibilidade e acoplamento excessivo entre deploy e mudança de esquema.

Etapa	Ferramenta/processo	Prática	Controlo
Código	Git, PRs, revisão	Mudanças pequenas e auditáveis	Aprovação e CI
Build	Pipeline CI/CD	Testes, linting, scans, SBOM	Bloqueios automáticos
Empacotamento	Containers assinados	Artefactos imutáveis	Verificação de

			assinatura
Entrega	Argo CD	Sincronização declarativa	Histórico, drift detection, rollback
Produção	Canary/blue-green	Exposição gradual	SLOs, métricas e abortos

7. Operação, resiliência e capacidade

Para um portal do Estado, disponibilidade elevada significa mais do que redundância técnica. Exige ensaio regular de recuperação, simulação de falhas, monitorização baseada em SLOs e governação de capacidade. A plataforma deve ser desenhada para falhar graciosamente: serviços não críticos podem degradar, mas os serviços nucleares devem manter-se operacionais.

Recomenda-se separar ambientes de desenvolvimento, integração, pré-produção e produção com controlos rigorosos de promoção. A produção deve dispor de múltiplos nós de controlo, múltiplos nós de trabalho, PostgreSQL em alta disponibilidade, backups testados, object storage redundante e observabilidade centralizada.

7.1 Métricas mínimas de governação operacional

- Disponibilidade por serviço e por jornada crítica do utilizador.
- Latência percentil (p95/p99) para páginas, APIs e autenticação.
- Taxa de erro por domínio funcional e por organismo integrado.
- Tempo médio de detecção e de recuperação de incidentes.
- Tempo de rollout, taxa de rollback e falhas por release.
- Consumo de capacidade por ambiente, por serviço e por janela temporal.

8. Governação tecnológica

Um portal estatal não se sustenta apenas com boa tecnologia. Precisa de governação arquitectural. Deve existir um comité técnico com mandato claro para definir padrões de API, identidade, observabilidade, gestão de dependências, requisitos de segurança, níveis de suporte e ciclos de actualização. Sem isso, a plataforma tenderá a fragmentar-se em excepções e 'soluções provisórias' — essa praga administrativa que em muitos sítios dura mais do que a pedra.

A governação deve incluir um catálogo de componentes aprovados, uma política de versões suportadas, um processo de excepções documentado e indicadores de dívida técnica. O objectivo é permitir flexibilidade sem cair em entropia tecnológica.

9. Roadmap de implementação (0–24 meses)

Fase	Horizonte	Objectivos principais	Resultado
Fase 0	0–3 meses	Definição da arquitectura, landing zone, identidade, observabilidade base, padrões CI/CD e segurança.	Fundação comum
Fase 1	3–6 meses	Portal base, autenticação, design system, primeiros serviços públicos, GitOps e pipelines	Primeira operação

		controladas.	
Fase 2	6–12 meses	Integrações prioritárias, OpenSearch, eventos, dashboards operacionais, DR ensaiado.	Escala funcional
Fase 3	12–18 meses	Maturação de SLOs, multi-zona, optimização de custo, reforço de auditoria e conformidade.	Resiliência robusta
Fase 4	18–24 meses	Automação avançada, self-service interno, revisão de arquitectura e expansão de serviços.	Plataforma madura

10. Riscos típicos e como mitigá-los

Risco	Impacto	Mitigação
Micro-serviços em excesso desde o início	Complexidade operacional e cognitiva	Começar por serviços de domínio bem delimitados.
Dependência excessiva de fornecedor	Custos e rigidez estratégica	Preferir APIs abertas, standards e portabilidade.
Deploys manuais e urgentes	Erros, drift e falta de auditoria	Adoptar GitOps e pipelines bloqueantes.
Segurança tratada no fim	Superfície de ataque elevada	Shift-left security, gestão de segredos e validação contínua.
Baixa disciplina de versões	Acumulação de dívida técnica	Política clara de suporte e upgrades regulares.

11. Decisão recomendada

Para um novo portal do Estado com acesso massivo, a recomendação final é adoptar uma plataforma aberta assente em Kubernetes, Gateway API, Cilium, Keycloak, PostgreSQL, OpenSearch, Prometheus/Grafana e Argo CD, combinando front-end moderno em React/Next.js com serviços core em Spring Boot e serviços satélite em FastAPI, quando a rapidez de desenvolvimento e a leveza do serviço o justificarem.

Esta escolha oferece um equilíbrio forte entre soberania tecnológica, escalabilidade, auditabilidade, segurança e capacidade de evolução. Mais importante ainda: permite que o portal seja tratado como uma plataforma pública viva, não como um produto fechado condenado a envelhecer mal.

12. Referências técnicas e fontes oficiais

1. Kubernetes releases e estado da série 1.35: https://kubernetes.io/releases/
2. Kubernetes 1.35 overview: https://kubernetes.io/releases/1.35/
3. Kubernetes Gateway API documentation: https://kubernetes.io/docs/concepts/services-networking/gateway/
4. Gateway API introduction: https://gateway-api.sigs.k8s.io/
5. Gateway API v1.4.0 GA: https://kubernetes.io/blog/2025/11/06/gateway-api-v1-4/
6. Ingress NGINX retirement statement: https://kubernetes.io/blog/2026/01/29/ingress-nginx-statement/
7. Ingress2Gateway 1.0 release: https://kubernetes.io/blog/2026/03/20/ingress2gateway-1-0-release/

8. Cilium project and stable releases: https://github.com/cilium/cilium
9. Keycloak homepage and releases: https://www.keycloak.org/
10. Keycloak 26.6.1 released: https://www.keycloak.org/2026/04/keycloak-2661-released
11. PostgreSQL homepage and latest releases: https://www.postgresql.org/
12. PostgreSQL release notes: https://www.postgresql.org/docs/release/
13. OpenSearch downloads and release line 3.6.0: https://opensearch.org/downloads/
14. OpenSearch version history: https://docs.opensearch.org/latest/version-history/
15. Prometheus downloads: https://prometheus.io/download/
16. Prometheus security model: https://prometheus.io/docs/operating/security/
17. Argo CD release process and cadence: https://argo-cd.readthedocs.io/en/latest/developer-guide/release-process-and-cadence/
18. Argo CD releases and supply-chain verification notes: https://github.com/argoproj/argo-cd/releases

Nota metodológica

As recomendações deste documento combinam boas práticas de arquitectura para plataformas digitais de grande escala com validação de versões, políticas de suporte e estado recente dos projectos, consultados em fontes oficiais em Abril de 2026.

Anexo A — Evidência técnica sintética

- Kubernetes 1.35 encontra-se activamente suportado, com a release 1.35.3 publicada a 19 de Março de 2026.
- A comunidade Kubernetes anunciou a retirada do Ingress NGINX em Março de 2026, pelo que uma plataforma nova deve privilegiar Gateway API.
- Gateway API é apresentado como a próxima geração de Ingress e de APIs de service networking em Kubernetes.
- Keycloak 26.6.1 foi publicado a 15 de Abril de 2026.
- OpenSearch 3.6.0 foi publicado a 7 de Abril de 2026.
- Prometheus apresenta releases actuais activas em Abril de 2026.
- Argo CD mantém apenas as três minor versions mais recentes para patch releases.
- Cilium mantém as três minor versions estáveis mais recentes.