Resumen ejecutivo

Portal Único de Participación Ciudadana

El **Portal Único de Participación Ciudadana** se concibe como una plataforma digital modular, segura y escalable, diseñada para concentrar y facilitar los principales canales de interacción entre la ciudadanía y el Estado, con trazabilidad, accesibilidad y capacidades de integración con sistemas públicos existentes.

La solución se plantea bajo un enfoque por capas (arquitectura multicapa) para evitar la dependencia de una estructura monolítica difícil de evolucionar. Esta aproximación permite incorporar nuevos módulos (denuncias, consultas públicas, monitoreo de transparencia, visualización de datos, etc.) sin afectar el funcionamiento del portal completo, garantizando mantenibilidad, desempeño y control de riesgos.

Componentes principales:

- **Capa de Presentación (Frontend):** interfaz accesible y responsiva con enfoque Mobile-First; formularios ciudadanos simplificados; paneles de datos (dashboards) y un asistente virtual (chatbot) para guiar trámites y preguntas frecuentes.
- **Capa de Contenidos (CMS):** WordPress como motor para la publicación de información institucional (noticias, guías, páginas informativas), habilitando autonomía editorial y reduciendo dependencia técnica.
- **Capa de Negocio (Backend):** módulos transaccionales críticos (p. ej., denuncias y consultas) con flujos de estados, validaciones, roles y trazabilidad; y conectores para interoperabilidad con plataformas públicas.
- **Capa de Datos:** base de datos optimizada para registros ciudadanos y trazabilidad; almacenamiento seguro de adjuntos y evidencias; separación lógica entre datos públicos (CMS) y transaccionales (módulos críticos).
- **Capa de Integraciones:** interoperabilidad mediante API (REST) y conectores con SAIP, 3-1-1, Datos Abiertos (CKAN), MapaInversiones/DGCP, y servicios de notificación (SMTP/correo transaccional).
- **Infraestructura y Seguridad:** ambientes separados (Desarrollo, QA/Testing y Producción), cifrado TLS/HTTPS, WAF, bitácoras/auditoría, respaldos y controles de acceso con MFA para administración.

Resultado esperado

Un portal que no sea solo informativo, sino una aplicación web institucional robusta, capaz de gestionar participación ciudadana con seguridad, integridad del dato, transparencia operativa y continuidad, reduciendo tiempos de atención y fortaleciendo la confianza pública.

Estructura tecnológica propuesta por capas

Enfoque de arquitectura

La arquitectura se define como modular y multicapa, orientada a:

- Escalabilidad: crecimiento por demanda sin reingeniería total.
- Seguridad y control: aislamiento de componentes críticos y reducción de superficie de ataque.
- Interoperabilidad: integración ordenada con sistemas nacionales existentes.
- Sostenibilidad operativa: mantenimiento simplificado y despliegues controlados.

Capas y componentes

A. Presentación y experiencia ciudadana

- Portal accesible conforme a WCAG 2.1, con diseño responsivo y optimizado para móviles.
- Formularios con validación y guías de llenado, reduciendo errores y abandono.
- Dashboards de información y seguimiento ciudadano.
- Asistente virtual (chatbot) para orientación y consultas frecuentes.

B. Gestión de contenidos

- CMS (WordPress) para contenido informativo institucional, con gobierno editorial (roles, flujos de aprobación y control de cambios).
- **Modelo de implementación recomendado:** híbrido (CMS para contenido; módulos críticos fuera del CMS), por balance de seguridad, rendimiento y sostenibilidad.

C. Lógica de negocio

- **Módulos nativos para participación ciudadana:**
  - Consultas públicas: publicación, recepción, moderación y reporte.
  - Monitoreo de transparencia: carga/cálculo de indicadores y reportes.
  - Denuncias DIGEIG: recepción segura, ticketing, estados y trazabilidad interna.

- **Módulos de integración:**
  - Conectores para consulta, derivación o intercambio de información con plataformas externas.

D. Datos y almacenamiento

- Base de datos para registros ciudadanos y trazabilidad.
- Cifrado en reposo para adjuntos/evidencias sensibles.
- Separación lógica (o física) entre:
  - Datos públicos del CMS.
  - Datos transaccionales de módulos críticos.

E. Interoperabilidad

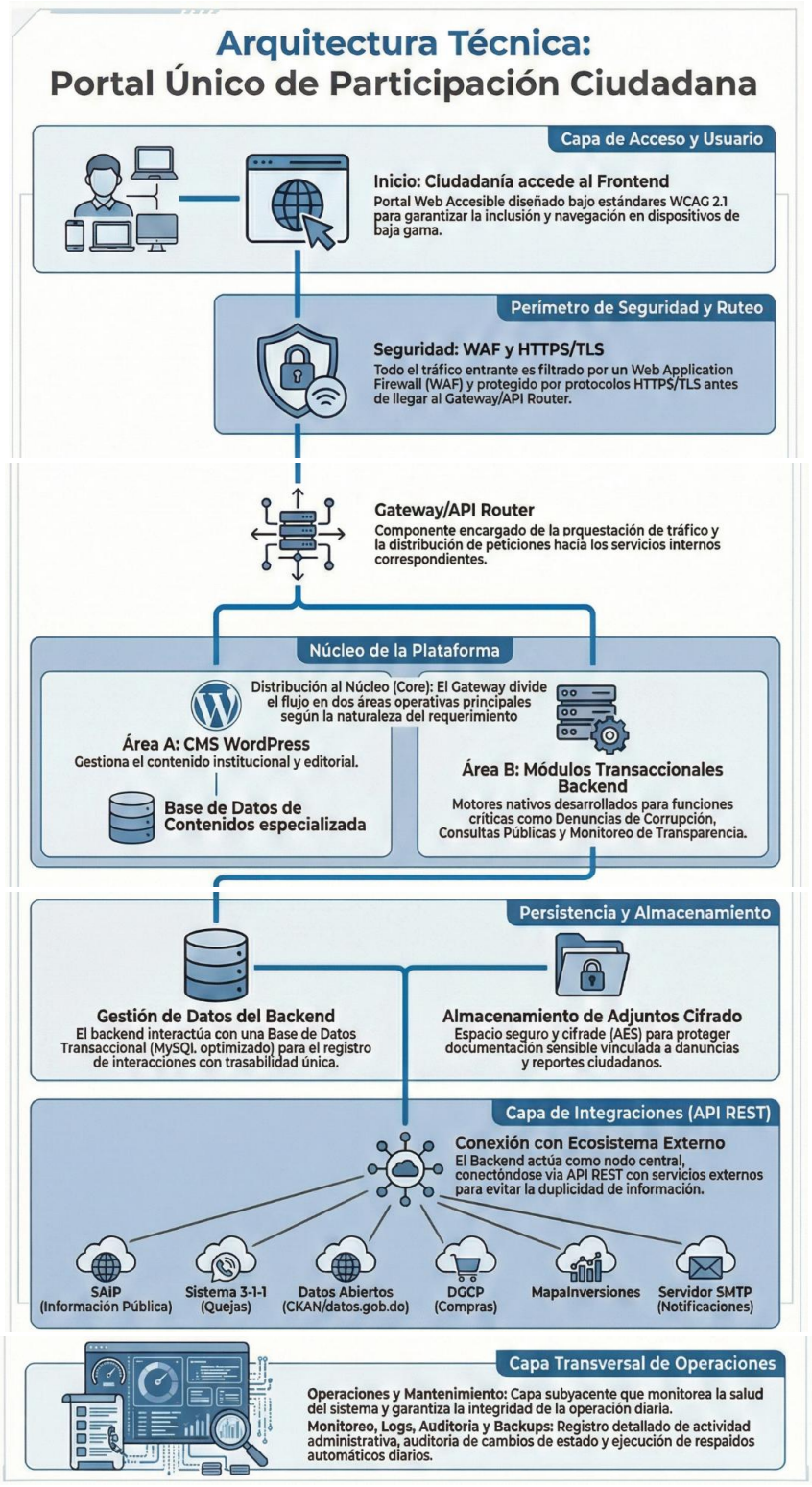
- Integración mediante API REST y conectores seguros con:
  - SAIP, 3-1-1, CKAN/Datos Abiertos, MapaInversiones/DGCP, y notificaciones (SMTP/correo).

F. Infraestructura, operación y seguridad

- Ambientes separados: Desarrollo, QA/Testing y Producción.
- Controles de protección: WAF, TLS/HTTPS, hardening y gestión de vulnerabilidades.
- Auditoría: bitácoras de acciones administrativas y eventos críticos.
- Continuidad: respaldos cifrados y pruebas periódicas de restauración.

Beneficios institucionales

- Mejora de la experiencia ciudadana y reducción de barreras de acceso.
- Trazabilidad y control de procesos críticos (especialmente denuncias y consultas).
- Integración con el ecosistema digital estatal sin duplicar plataformas.
- Mayor resiliencia operativa y seguridad, con crecimiento ordenado por módulos.


Arquitectura Técnica:
Portal Único de Participación Ciudadana
Capa de Acceso y Usuario
Inicio: Ciudadanía accede al Frontend
Portal Web Accesible diseñado bajo estándares WCAG 2.1 para garantizar la inclusión y navegación en dispositivos de baja gama.
Perímetro de Seguridad y Ruteo
Seguridad: WAF y HTTPS/TLS
Todo el tráfico entrante es filtrado por un Web Application Firewall (WAF) y protegido por protocolos HTTPS/TLS antes de llegar al Gateway/API Router.
Gateway/API Router
Componente encargado de la orquestación de tráfico y la distribución de peticiones hacia los servicios internos correspondientes.
Núcleo de la Plataforma
Distribución al Núcleo (Core): El Gateway divide el flujo en dos áreas operativas principales según la naturaleza del requerimiento
Área A: CMS WordPress
Gestiona el contenido institucional y editorial.
Base de Datos de Contenidos especializada
Área B: Módulos Transaccionales Backend
Motores nativos desarrollados para funciones críticas como Denuncias de Corrupción, Consultas Públicas y Monitoreo de Transparencia.
Persistencia y Almacenamiento
Gestión de Datos del Backend
El backend interactúa con una Base de Datos Transaccional (MySQL optimizado) para el registro de interacciones con trasabilidad única.
Almacenamiento de Adjuntos Cifrado
Espacio seguro y cifrade (AES) para protager documentación sensible vinculada a danuncias y reportes ciudadanos.
Capa de Integraciones (API REST)
Conexión con Ecosistema Externo
El Backend actúa como nodo central, conectóndose via API REST con servicios externos para evitar la duplicidad de información.
SAIP (Información Pública)
Sistema 3-1-1 (Quejas)
Datos Abiertos (CKAN/datos.gob.do)
DGCP (Compras)
MapaInversiones
Servidor SMTP (Notificaciones)
Capa Transversal de Operaciones
Operaciones y Mantenimiento: Capa subyacente que monitorea la salud del sistema y garantiza la integridad de la operación diaria.
Monitoreo, Logs, Auditoria y Backups: Registro detallado de actividad administrativa, auditoria de cambios de estado y ejecución de respaidos automáticos diarios.

Requerimientos mínimos de infraestructura (por ambientes)

Ambientes obligatorios

- Desarrollo (DEV): construcción y pruebas técnicas internas.
- Calidad/Pruebas (QA/UAT): pruebas funcionales, seguridad y validación de usuarios.
- Producción (PROD): servicio público (alta disponibilidad y monitoreo continuo).

Regla operativa: prohibido desarrollar o probar cambios directamente en producción.

Topología mínima recomendada (PROD)

Separación por roles (mínimo recomendado):

1. Capa Web/Frontend (portal público)
2. Capa Aplicación/Backend (módulos transaccionales: denuncias, consultas, monitoreo)
3. Capa Datos (BD transaccional + BD CMS separadas lógicamente)
4. Almacenamiento de adjuntos (repositorio cifrado)
5. Seguridad perimetral (WAF + balanceo/rate limiting)
6. Monitoreo y logs centralizados (auditoría y trazabilidad)

Notas de diseño:

- Si WordPress se usa como CMS, se recomienda aislarlo de los módulos transaccionales.
- Separar BD CMS y BD transaccional reduce riesgo y mejora control.

Dimensionamiento mínimo sugerido (referencial)

Estos mínimos están pensados para un inicio controlado, con posibilidad de escalar vertical u horizontalmente.

DEV (mínimo)

- 1 VM/Servidor:
  - 4 vCPU | 8–16 GB RAM | 150 GB SSD
  - Servicios: Web + App + BD (solo para desarrollo)

QA/UAT (mínimo)

- 2 VM/Servidores:
  - Web/App: 4 vCPU | 8–16 GB RAM | 150 GB SSD
  - BD: 4 vCPU | 16–32 GB RAM | 200–300 GB SSD

PROD (mínimo recomendado)

- **3 VM/Servidores:**
  1. Web/Frontend + WAF/Reverse Proxy
     - 4–8 vCPU | 16 GB RAM | 150–250 GB SSD

2. **App/Backend (módulos transaccionales)**
    - 8 vCPU | 16–32 GB RAM | 200–300 GB SSD
3. **Base de datos**
    - 8–16 vCPU | 32–64 GB RAM | 500 GB SSD (alto IOPS)

Almacenamiento de adjuntos/evidencias (PROD)

- **Repositorio dedicado (NAS/Objeto/FS):**
    - ≥ 1 TB inicial (escalable), con cifrado en reposo y control de acceso.

Software base y componentes

- **Sistema Operativo:** Linux (LTS).
- **Servidor web:** Nginx o Apache (estandarizado).
- **Runtime/App:** PHP 8.x (y/o framework), o stack equivalente si se define React/Vue + API.
- **Base de datos:** MySQL/MariaDB (tuning, backups, replicación si aplica).
- **WAF:** reglas OWASP Top 10 + protección anti-bot + rate limiting.
- **TLS/HTTPS:** certificados válidos, políticas fuertes (HSTS recomendado).
- **Gestión de identidades y accesos (IAM):**
    - Roles por perfil; MFA obligatorio para administración.
- **Logs y auditoría:**
    - Centralización (SIEM/stack de logs) con retención definida.
- **Backups:**
    - BD + adjuntos + configuración + artefactos de despliegue.

Requisitos mínimos de seguridad operativa

- **Segmentación de red:** DMZ (público) / red app / red datos.
- **Principio de mínimo privilegio:** cuentas separadas (admin, operador, publicador, analista).
- **Hardening:** cierre de puertos, deshabilitar servicios no usados, headers seguros.
- **Gestión de parches:** ventanas mensuales + parches críticos inmediatos.
- **Escaneo de vulnerabilidades:** periódico (DAST para portal, revisión plugins/temas WordPress si aplica).
- **Protección contra pérdida de datos (adjuntos):**
    - Antivirus/antimalware en carga de archivos, límites de tamaño/tipo, cuarentena.

Continuidad (parámetros recomendados)

- **RPO objetivo**: 24 h (mínimo) / ideal 4–8 h (si la criticidad lo exige).
- **RTO objetivo:** 8–24 h (mínimo) / ideal 4–8 h.
- **Backups:**

- BD: diario + copias incrementales (si aplica).
- Adjuntos: diario o por lotes; con verificación de integridad.
- Prueba de restauración: mensual (obligatoria como evidencia).

Arquitectura Sugeridas Entregables Consultoría

La arquitectura se ha diseñado bajo un enfoque **modular y por capas**, lo que permite que el sistema sea escalable, seguro y fácil de mantener, evitando que se convierta en una estructura "monolítica" difícil de actualizar.

Principios de diseño (base de la arquitectura)

- **Modularidad:** cada módulo (denuncias, consultas públicas, monitoreo, etc.) puede evolucionar sin afectar a los demás.
- **Separación por capas:** presentación, contenido, negocio, datos, integraciones e infraestructura.
- **Seguridad por diseño (Security-by-Design):** control de acceso, cifrado, auditoría y hardening desde el inicio.
- **Escalabilidad y mantenibilidad:** posibilidad de crecer por demanda (usuarios, formularios, integraciones) sin rediseño total.

1) Capa de Presentación (Frontend y Experiencia Ciudadana)

Objetivo: ser la "cara" del portal, reducir la brecha digital y maximizar la accesibilidad.

Componentes y requisitos clave

- **Interfaz accesible (WCAG 2.1):**
  - Compatibilidad con lectores de pantalla.
  - Navegación por teclado (sin depender del mouse).
  - Alto contraste, tipografías legibles, foco visible, textos alternativos.
- **Diseño responsivo (Mobile-First):**
  - Componentes ligeros, optimización de imágenes, carga progresiva.
  - Tolerancia a conexiones lentas (minificación, lazy-loading).
- **Elementos funcionales del frontend:**
  - **Dashboards ciudadanos (ej.:** indicadores de transparencia, resultados de consultas).
  - **Formularios simplificados (denuncias/solicitudes):**
    - Validaciones en tiempo real.
    - Guardado parcial y recuperación (opcional).

- Adjuntos controlados (tamaño/tipo), con guía clara.

- **Chatbot ciudadano / asistente:**
  - FAQ, guía de trámites, enrutamiento a módulos (SAIP/311, etc.).
  - Escalamiento a "contacto humano" cuando aplique (correo/ticket).

2) Capa de Gestión de Contenidos (CMS)

Objetivo: permitir que el equipo institucional publique y gestione contenido sin depender de desarrolladores.

Motor propuesto: WordPress para contenido informativo (noticias, guías, preguntas frecuentes, páginas institucionales).

Escenarios de implementación

1. **Opción 1 – CMS Monolítico (rápida y económica)**
   - WordPress maneja contenido + formularios + lógica.
   - Ventajas: menor complejidad inicial.
   - Riesgo: crecimiento difícil, mayor superficie de ataque, rendimiento condicionado.
2. **Opción 2 – Híbrida (equilibrio recomendado)**
   - WordPress solo para contenido.
   - Módulos transaccionales (denuncias, consultas, monitoreo) en backend independiente (PHP/Framework).
   - Ventajas: más seguridad, mejor rendimiento, separación de responsabilidades.
   - Permite endurecer WordPress como "capa pública" y aislar lo crítico.
3. **Opción 3 – Headless / Desacoplada (moderna y robusta)**
   - WordPress expone contenido vía API (REST/GraphQL).
   - Frontend en React/Vue consumiendo APIs.
   - Ventajas: mejor UX, control total del frontend, aislamiento fuerte del CMS.
   - Requiere mayor madurez técnica (DevOps/CI-CD, monitoreo, API gateway).

3) Capa de Lógica de Negocio (Backend)

Objetivo: contener la "inteligencia" del portal, reglas de negocio y procesos institucionales.

3.1 Módulos nativos (desarrollo propio)

- **Consultas Públicas**
  - Publicación de borradores, recepción de comentarios, moderación, reporte de resultados.
  - Trazabilidad de versiones y participación.
- **Monitoreo de Transparencia**

  - Carga de evaluaciones/mediciones.
  - Cálculo de indicadores y generación de rankings/reportes.
- **Denuncias DIGEIG (módulo crítico)**
  - Recepción segura, número de seguimiento (ticket), gestión de estados.
  - Flujo interno: recibido → validación → investigación → cerrado/archivado.
  - Canal de notificación al ciudadano (sin exponer datos sensibles).

3.2 Módulos de integración (conectores / orquestación)

- Integraciones sin duplicar sistemas externos.
- El portal consulta/consume datos o redirige a plataformas oficiales cuando corresponda.
- Permite una experiencia "hub" sin crear islas de información.

4) Capa de Datos y Almacenamiento

Objetivo: persistencia segura y ordenada de la información.

Tecnología base sugerida

- Base de datos: MySQL/MariaDB optimizada (índices, particiones si crece, tuning).
- Almacenamiento de archivos: repositorio de adjuntos con controles (objeto/FS), idealmente separado del servidor web.

Seguridad del dato

- Cifrado en reposo para adjuntos sensibles (denuncias, evidencias).
- Cifrado en tránsito (TLS) entre servicios internos.
- Separación lógica o física:
  - BD del CMS (pública) separada de BD transaccional (privada).
- Política de retención y clasificación:
  - Definir tiempos de conservación (legal/operativo) y anonimización cuando aplique.

5) Capa de Servicios e Integraciones (Interoperabilidad)

Objetivo: conectar el portal con el ecosistema estatal, garantizando interoperabilidad y coherencia.

Mecanismos

- API REST (principal), tokens, mTLS si aplica, validación de esquemas.
- Conectores seguros y colas/asíncrono cuando se requiera robustez (notificaciones, sincronizaciones).

Integraciones críticas identificadas

- SAIP: solicitudes de acceso a la información (consulta/derivación).
- Sistema 3-1-1: canalización de quejas/servicios públicos.

- Datos Abiertos (CKAN): visualización/consumo de datasets.
- MapaInversiones / DGCP: obras, compras, procesos (según disponibilidad de APIs/datos).
- SMTP/Servicio de correo: notificaciones automáticas (acuse de recibo, cambios de estado, etc.).

6) Infraestructura, Operación y Seguridad (Hosting / Plataforma)

Objetivo: garantizar disponibilidad, rendimiento, continuidad y protección.

Entornos obligatorios

- Desarrollo (construcción)
- QA/Testing (pruebas funcionales y de seguridad)
- Producción (público)

Regla: nunca se desarrolla directamente en producción.

Plataforma recomendada

- Linux + PHP 8.x (si el backend es PHP) + Nginx/Apache según estándar institucional.
- WAF (Firewall de Aplicaciones Web):
  - Protección ante SQLi, XSS, CSRF, bots, rate limiting.
- HTTPS/TLS extremo a extremo (certificados válidos, HSTS recomendado).
- Gestión de identidades y accesos (IAM):
  - Roles por perfil (publicador, moderador, analista, supervisor, admin).
  - MFA para accesos administrativos.
- Auditoría y bitácoras (Logs):
  - Registro de acciones administrativas (quién / qué / cuándo / desde dónde).
  - Retención de logs para auditoría y respuesta a incidentes.
- Backups y continuidad:
  - Backups cifrados, pruebas de restauración periódicas.
  - (Deseable) Plan de contingencia/DR: RPO/RTO definidos.

7. Controles transversales recomendados (aplican a todas las capas)

Estos controles son **obligatorios como práctica de madurez** porque aseguran que, independientemente de la tecnología elegida (monolítico, híbrido o headless), el Portal opere con **seguridad, continuidad, trazabilidad y calidad**.

7.1 Observabilidad (Monitoreo integral y telemetría)

Objetivo: detectar fallas antes de que afecten al ciudadano, medir desempeño real y asegurar trazabilidad operativa.

Alcance mínimo recomendado:

- **Monitoreo de disponibilidad (Uptime):**
  - HTTP/HTTPS (portal y endpoints críticos).
  - Health checks para módulos transaccionales (denuncias/consultas).
- **Monitoreo de rendimiento (Performance):**
  - Latencia promedio y percentiles (p95/p99) por endpoint.
  - Tiempos de respuesta por formulario y carga de páginas.
  - Uso de recursos (CPU, RAM, disco, IOPS, red).
- **Monitoreo de errores y calidad:**
  - Tasa de errores 4xx/5xx, timeouts, fallos de integraciones.
  - Errores de aplicación (exceptions) con trazabilidad del flujo.
- **Trazas distribuidas (Tracing) y correlación:**
  - ID de correlación por solicitud para unir frontend ↔ backend ↔ BD ↔ integraciones.
- **Alertas y umbrales (Alerting):**
  - Alertas por caída, degradación, saturación, anomalías de tráfico.
  - Alarmas por fallos de jobs/colas y por retrasos de notificaciones.

Evidencias clave para auditoría:

- Dashboard mensual con uptime, latencia, errores y disponibilidad por módulo.
- Bitácora de alertas + tickets generados + acciones correctivas.

7.2 DevSecOps y CI/CD (despliegues controlados con gates)

Objetivo: garantizar que todo cambio pase por validaciones técnicas, funcionales y de seguridad, evitando despliegues improvisados.

Buenas prácticas mínimas:

- **Control de versiones y gestión de ramas (Git):**
  - Flujo con PR/MR obligatorio y revisiones.
- **Pipelines automatizados CI/CD:**
  - Compilación/build (si aplica), pruebas unitarias, pruebas de integración.
- **Gates obligatorios antes de producción:**

1. **QA técnico** (pruebas automáticas y regresión)
2. **Seguridad** (SAST/DAST + revisión dependencias + escaneos contenedores si aplica)
3. **UAT** (validación funcional por el área dueña)
4. **Go/No-Go** (aprobación final con plan de rollback)

- **Despliegues con control:**
  - Ventana de mantenimiento, checklist post-deploy (smoke test).
  - **Rollback documentado** y probado (volver a la versión anterior).

Evidencias clave:

- Historial de despliegues (versión/fecha/responsable).
- Reporte de pipeline (resultados de pruebas, escaneos).
- Acta Go/No-Go + checklist UAT.

7.3 Hardening y gestión de vulnerabilidades (prevención y respuesta)

Objetivo: reducir la superficie de ataque y gestionar vulnerabilidades de forma continua.

Hardening mínimo por componente:

- **Servidor / SO:**
  - Cierre de puertos, servicios mínimos, firewall, SSH seguro, segregación por redes.
  - Cuentas separadas (admin vs operación) + MFA donde aplique.
- **Aplicación:**
  - Validación server-side, protección CSRF, gestión segura de sesiones.
  - Headers de seguridad (CSP, HSTS, X-Frame-Options, etc.).
- **WAF y perímetro:**
  - Reglas OWASP Top 10, rate limiting, anti-bot, reputación IP.
- **Base de datos:**
  - Usuarios con mínimo privilegio, acceso solo desde red interna, auditoría de consultas críticas.
- **WordPress (si aplica):**
  - Lista blanca de plugins, actualización controlada, deshabilitar edición desde panel, hardening wp-config.
  - Separación CMS vs módulos críticos (ideal).

Gestión de vulnerabilidades (ciclo continuo):

- **SAST (código fuente):** al menos en cada release.
- **DAST (portal en QA/PROD):** mensual o por cambios críticos.
- **Escaneo de dependencias (SCA):** librerías, paquetes, CVEs.
- **Gestión de parches:** mensual + parches críticos inmediatos.
- **Remediación con SLA:** priorización por severidad (Crítica/Alta/Media/Baja).

Evidencias clave:

- Reportes de escaneo + plan de remediación + cierre de hallazgos.
- Registro de parches + acta de mantenimiento.

7.4 Gobierno de datos (clasificación, acceso, trazabilidad y ciclo de vida)

Objetivo: asegurar que la información se gestione con criterios de privacidad, integridad y transparencia, según su sensibilidad.

Elementos mínimos del gobierno de datos:

- **Clasificación de la información:**
  - Pública / Uso Interno / Confidencial / Sensible (ej. denuncias y evidencias).
- **Acceso mínimo necesario (Least Privilege):**
  - RBAC por rol (operador, analista, supervisor, admin).
  - Segregación de funciones: quien aprueba no debe ser quien ejecuta cambios críticos.
- **Trazabilidad (auditoría):**
  - Bitácora de accesos y acciones sobre datos sensibles (lectura, modificación, descarga).
  - Auditoría reforzada en denuncias (cadena de custodia digital).
- **Ciclo de vida y retención:**
  - Retención por normativa/operación.
  - Eliminación segura o anonimización cuando aplique.
- **Calidad del dato:**
  - Reglas de validación y deduplicación (instituciones, RAI, etc. si integran catálogos).
- **Protección criptográfica:**
  - Cifrado en tránsito y en reposo, con control de llaves (y rotación cuando aplique).

Evidencias clave:

- Matriz de clasificación y retención.
- Registro de accesos a datos sensibles.
- Políticas de privacidad y tratamiento de datos publicadas y aplicadas.

7.5 Gestión de configuración y secretos (Secrets Management)

Objetivo: evitar exposición de credenciales y asegurar configuración consistente entre ambientes.

Requisitos mínimos:

- Secretos fuera del código (variables de entorno / vault).
- Rotación de credenciales (BD, SMTP, APIs).
- Separación de secretos por ambiente (DEV/QA/PROD).
- Auditoría de cambios de configuración.

Evidencias:

- Inventario de secretos (sin revelar valores) + política de rotación.
- Registro de cambios de configuración.

7.6 Continuidad, backups y recuperación (resiliencia operativa)

Objetivo: garantizar recuperación ante fallas, errores humanos o incidentes de seguridad.

Requisitos mínimos:

- Backups cifrados (BD + adjuntos + configuración).

- Pruebas de restauración mensuales.
- Definición formal de RPO/RTO.
- Procedimiento de recuperación documentado y probado.

Evidencias:

- Logs de backups + acta de restore test + resultados (RPO/RTO medidos).

7.7 Gestión de incidentes y respuesta (CSIRT operativo)

Objetivo: actuar rápido ante incidentes, minimizar impacto y dejar trazabilidad.

Requisitos mínimos:

- Flujo: detección → contención → erradicación → recuperación → lecciones aprendidas.
- Clasificación por severidad (S1–S4) y tiempos SLA.
- Comunicación interna y, si aplica, comunicación pública (status page).
- Preservación de evidencias (logs, trazas, snapshots) ante incidentes críticos.

Evidencias:

- Ticket del incidente + timeline + post-mortem + plan de mejora.

Procedimientos mínimos de operación y mantenimiento (O&M)

Objetivo

Asegurar la **disponibilidad, seguridad, integridad de datos, trazabilidad y continuidad operativa** del Portal Único de Participación Ciudadana, mediante rutinas estandarizadas de monitoreo, mantenimiento, control de cambios y gestión de incidentes.

Roles mínimos (gobernanza operativa)

- **Propietario del Servicio (Service Owner):** responsable funcional y de priorización (Alta Dirección/área dueña).
- **Administrador de Plataforma (SysAdmin/DevOps):** servidores, despliegues, hardening, disponibilidad.
- **Administrador de Aplicación (App Admin):** configuración de módulos, parámetros, colas, jobs.
- **DBA/Administrador de Datos:** base de datos, backups, performance, restauración.
- **Seguridad (Ciberseguridad):** WAF, vulnerabilidades, monitoreo de eventos, respuesta a incidentes.
- **Gestor de Contenidos (Comunicaciones/Transparencia):** publicación en CMS y flujos editoriales.
- **Mesa de Ayuda/Soporte:** recepción y escalamiento de tickets, comunicación con usuarios internos.

Requisito: accesos administrativos con **MFA**, roles segregados y bitácora/auditoría habilitada.

Rutinas operativas por frecuencia (checklist)

A) Rutina diaria (operación básica)

Responsable: Admin Plataforma + Seguridad + DBA (según tarea)

Objetivo: detectar fallas temprano y asegurar continuidad diaria.

1. **Disponibilidad (Uptime)**
   - Verificar estado del portal (HTTP/HTTPS), latencia y errores 4xx/5xx.
   - **Evidencia:** captura o export de dashboard + log de incidentes (si aplica).
2. **Alertas de seguridad (WAF/IDS/Logs)**
   - Revisar eventos críticos: fuerza bruta, SQLi/XSS, picos de tráfico, bloqueos.
   - **Evidencia:** reporte diario (resumen) o ticket de investigación.
3. **Backups (éxito y consistencia)**
   - Confirmar ejecución correcta de backups (BD + adjuntos + configs).
   - **Evidencia:** log de backup + checksum/registro de verificación.
4. **Colas/Jobs/Notificaciones**
   - Confirmar envío de correos (SMTP), colas y tareas programadas sin errores.
   - **Evidencia:** log de job scheduler + muestra de notificaciones.

B) Rutina semanal (control y prevención)

Responsable: Admin Plataforma + App Admin + Seguridad + DBA

1. **Revisión de capacidad**
   - CPU/RAM/Disco/IOPS, crecimiento de BD y almacenamiento de adjuntos.
   - **Evidencia:** reporte semanal de métricas y tendencias.
2. **Revisión de cuentas y accesos**
   - Altas/bajas; revisar roles, intentos fallidos, accesos fuera de horario.
   - **Evidencia:** export de auditoría IAM + acta de revisión.
3. **Revisión de integraciones (SAIP/311/CKAN/DGCP/otros)**
   - Health-check de endpoints, timeouts, reintentos, cambios en respuestas.
   - **Evidencia:** bitácora de pruebas + dashboard de integraciones.
4. **Cambios menores controlados**
   - Aplicar ajustes de configuración de bajo riesgo en QA antes de PROD.
   - **Evidencia:** ticket de cambio + validación en QA + aprobación.

C) Rutina mensual (cumplimiento, calidad y seguridad)

Responsable: Seguridad + Admin Plataforma + App Admin + DBA + Propietario del Servicio

1. **Parches y actualizaciones**
    - SO, servidor web, runtime (PHP), DB engine, librerías, dependencias.
    - WordPress (si aplica): core, plugins y temas **solo desde QA → PROD**.
    - **Evidencia:** registro de parches + ventana de mantenimiento + acta de cambio.
2. **Escaneo de vulnerabilidades**
    - DAST (portal público) y revisión de CVEs de componentes.
    - **Evidencia:** reporte de escaneo + plan de remediación + cierre de hallazgos.
3. **Prueba de restauración (restore test)**
    - Restaurar BD y al menos una muestra de adjuntos en ambiente controlado.
    - **Evidencia:** acta de restore test + tiempos (RTO/RPO) + resultado.
4. **Revisión de logs y auditoría**
    - Confirmar retención, integridad, eventos críticos y trazabilidad de acciones admin.
    - **Evidencia:** reporte mensual de auditoría + confirmación de retención.
5. **Reporte mensual de operación (SLA/indicadores)**
    - Uptime, incidentes, tiempos de respuesta, volumen de formularios/denuncias (agregado), rendimiento, seguridad.
    - **Evidencia:** informe mensual firmado o aprobado por el Service Owner.

D) Rutina trimestral (resiliencia y mejora continua)

Responsable: Seguridad + Admin Plataforma + Propietario del Servicio

1. **Simulacro de incidente**
    - Ej.: caída de integración, saturación, bloqueo WAF, cuenta comprometida.
    - **Evidencia:** post-mortem / lecciones aprendidas + acciones preventivas.
2. **Prueba de continuidad/DR (si aplica)**
    - Conmutación o recuperación parcial según el plan definido.
    - **Evidencia:** acta de prueba + resultados y ajustes al plan.

Gestión de cambios (estándar mínimo)

- Todo cambio debe tener: **ticket**, **evaluación de riesgo**, **plan de rollback**, **ventana**, **aprobación** y **evidencia en QA/UAT**.
- Cambios en producción solo con criterio **Go/No-Go** y monitoreo post-despliegue (smoke test).

Evidencias mínimas por cambio

- Checklist QA/UAT completado
- Acta Go/No-Go (aprobación)
- Registro de despliegue (versión/fecha/responsable)
- Resultado de smoke test + monitoreo 24–48h

Gestión de incidentes (estándar mínimo)

Flujo: Detección → Registro → Clasificación → Contención → Erradicación → Recuperación → Post-mortem

- **Clasificación por severidad:** Crítica / Alta / Media / Baja
- **Tiempos objetivo:** definidos por el SLA (ej.: crítica atención inmediata)
- **Comunicación:** plantilla de aviso interno y, si aplica, aviso público (status page).

Evidencia obligatoria

- Ticket de incidente + línea de tiempo
- Evidencias técnicas (logs, capturas, métricas)
- Informe post-incidente con acciones correctivas/preventivas

Anexo operativo: formato de registro (plantilla rápida)

Bitácora mensual O&M

- Mes/Año:
- Uptime:
- Incidentes (cantidad / severidad):
- Cambios en producción (cantidad):
- Parches aplicados:
- Vulnerabilidades (abiertas/cerradas):
- Restore test (sí/no, resultado, RTO/RPO medidos):
- Observaciones / Riesgos:
- Aprobación (Service Owner):

Propuesta SLA operativo mínimo

A. Definiciones rápidas

- **Disponibilidad (Uptime):** porcentaje de tiempo en que el portal está accesible.
- **Tiempo de respuesta:** tiempo hasta primera atención/confirmación del incidente.
- **Tiempo de resolución:** tiempo hasta restablecer servicio o aplicar mitigación aceptable.
- **Severidad:** impacto en ciudadanía, datos, seguridad o continuidad.

B. Niveles de servicio recomendados

Disponibilidad objetivo (Producción)

- Portal público (Frontend): **≥ 99.5% mensual**

- Módulos críticos (Denuncias/Consultas): **≥ 99.5% mensual**
- Integraciones externas (SAIP/311/CKAN/DGCP): depende del tercero; se mide "disponibilidad de integración" separada.

RPO / RTO (continuidad)

- **RPO objetivo:** 8–24 horas (según criticidad del dato).
- **RTO objetivo:** 4–24 horas (según severidad e infraestructura).

Matriz SLA por severidad (respuesta y resolución)

Severidad	Criterio	Ejemplos	Tiempo de respuesta	Tiempo de resolución	Comunicación
Crítica (S1)	Portal caído, riesgo alto de seguridad/datos	Caída total, fuga de datos, compromiso admin	**≤ 30 min**	**≤ 4 h** (mitigación) / **≤ 24 h** (total)	Aviso inmediato interno + status si aplica
Alta (S2)	Afectación parcial significativa	Denuncias no registran, formularios fallan masivo	**≤ 1 h**	**≤ 8 h**	Aviso interno + actualización periódica
Media (S3)	Incidencia funcional sin impacto masivo	Error en sección no crítica, lentitud moderada	**≤ 4 h**	**≤ 3 días hábiles**	Ticket y notificación al solicitante
Baja (S4)	Mejora o fallo menor	Ajuste contenido, cambio menor UI	**≤ 1 día hábil**	**≤ 10 días hábiles**	Gestión estándar por backlog

Indicadores mínimos (para reporte mensual)

- Uptime (%)
- Cantidad de incidentes por severidad
- MTTA (tiempo promedio de atención) y MTTR (tiempo promedio de recuperación)
- Cambios en producción (cantidad + % exitosos)
- Vulnerabilidades abiertas/cerradas (por severidad)
- Resultado de restore test (sí/no + RTO/RPO medidos)

Checklist UAT + Go/No-Go

A. Checklist UAT (Validación funcional por el área usuaria)

1) Alcance validado

- Requerimientos/alcance del cambio confirmados (ticket + nota de versión)
- Criterios de aceptación definidos y entendidos

2) Pruebas funcionales

- Navegación general sin errores
- Formularios principales (denuncias/consultas) operan de inicio a fin
- Validaciones: campos obligatorios, formatos, adjuntos, límites
- Notificaciones (correo) enviadas correctamente (si aplica)
- Dashboards/visualizaciones cargan correctamente (si aplica)
- Roles y permisos: cada perfil solo ve lo que corresponde

3) Pruebas de datos

- Registros se guardan correctamente en BD
- No hay duplicidad ni corrupción de registros
- Exportes/reportes (si existen) funcionan

4) Pruebas de integraciones

- SAIP / 3-1-1 / CKAN / DGCP / otros: endpoints responden como se espera
- Manejo de caída: mensaje al usuario y reintentos (si aplica)

5) Accesibilidad y UX (mínimo)

- Flujo usable en móvil (pantallas clave)
- Textos claros y consistentes
- No hay bloqueos por teclado (si aplica)

Resultado UAT

- **APROBADO**
- APROBADO con observaciones
- NO APROBADO (requiere corrección).

Go/No-Go (control final antes de Producción)

Pre-requisitos técnicos

- Cambio probado en QA/UAT y aprobado por el área usuaria
- Plan de despliegue documentado (pasos + responsable + ventana)
- Plan de rollback probado o definido (cómo volver atrás)
- Backups verificados antes del despliegue (BD + adjuntos)
- Revisión de seguridad: WAF/logs/DAST (si aplica al cambio)
- Monitoreo activo y umbrales configurados (alertas)

Ejecución

- Despliegue completado sin errores
- Smoke test post-deploy (10–15 min):
  - Portal carga
  - Formulario principal envía correctamente
  - Login admin (si aplica) ok con MFA
  - Integración crítica responde (si aplica)

Decisión

- **GO** (publicar)
- GO con mitigación
- **NO-GO** (rollback)

Registro (obligatorio)

- Versión/Release:
- Fecha/hora:
- Responsable:
- Aprobación Service Owner:
- Evidencias adjuntas (capturas/logs/ticket):

Plan de implementación por fases

Enfoque general

Implementación incremental para entregar valor temprano, reducir riesgos y asegurar control de calidad:

- **Fases con entregables verificables**
- **Gates obligatorios:** QA + UAT + Seguridad + Go/No-Go
- **Evidencias:** actas, reportes, bitácoras, dashboards

Fase 0 — Preparación y gobierno (base de control)

Objetivo: dejar listo el marco operativo y técnico para construir sin improvisación.

Alcance

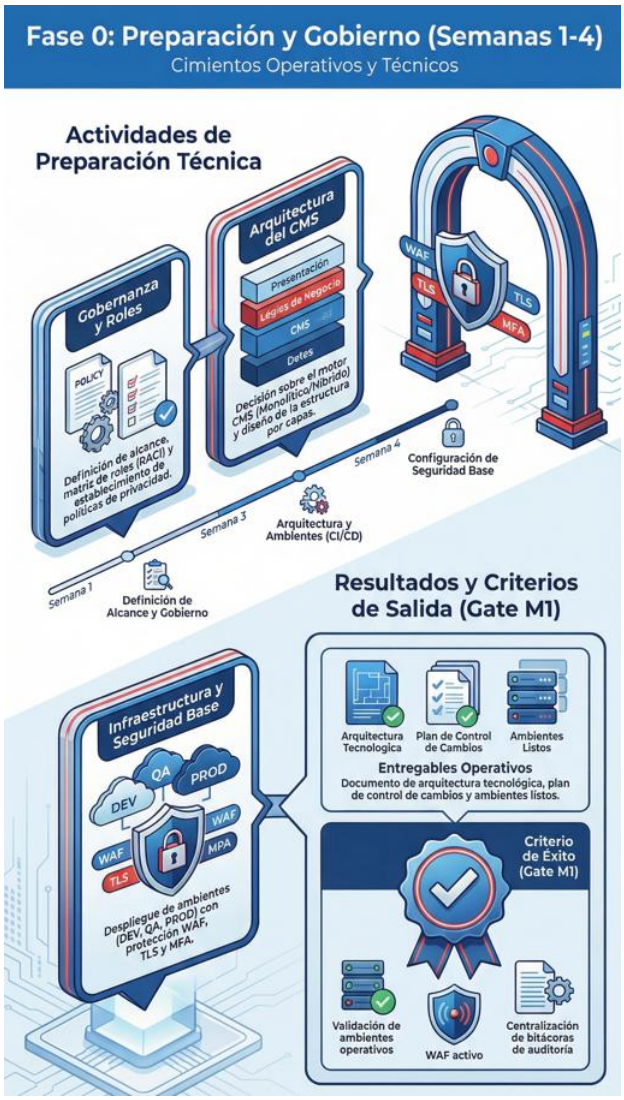
- Gobernanza (roles, RACI, flujos de aprobación)
- Lineamientos de seguridad, privacidad, retención, logging
- Diseño de arquitectura objetivo (capas) + decisión CMS (monolítico/híbrido/headless)
- Preparación de ambientes DEV/QA/PROD + CI/CD + WAF base

Entregables

- Documento de arquitectura + decisión tecnológica
- Plan de control de cambios + gestión de incidentes
- Plantillas: UAT + Go/No-Go + reporte SLA
- Ambientes operativos con monitoreo y backups

Gate de salida

- Ambientes listos + WAF activo + MFA admin + bitácoras centralizadas.



Fase 1: Portal informativo + CMS

Objetivo: publicar el portal como canal oficial con contenido institucional y estructura navegable.

Alcance

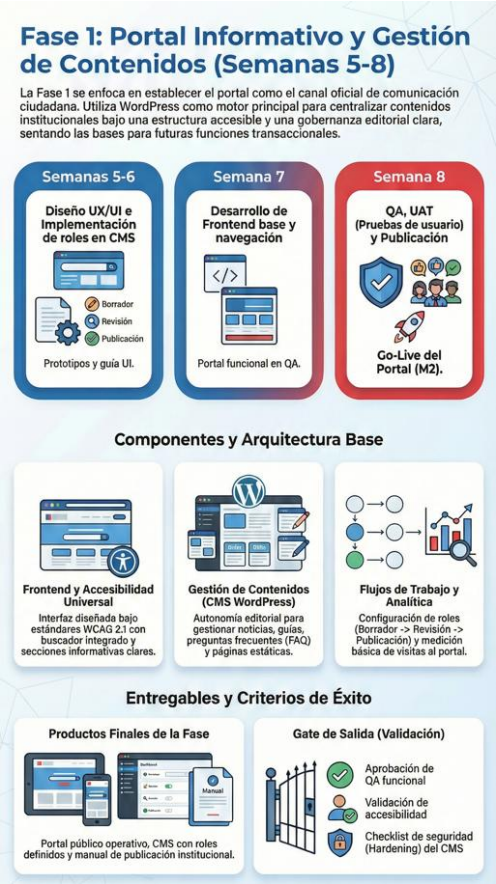
- Frontend base (home, secciones, buscador, accesibilidad)
- CMS WordPress (contenido, noticias, guías, FAQ)
- Flujos editoriales (borrador → revisión → publicación)
- Analítica básica (visitas, páginas más consultadas)

Entregables

- Portal público informativo
- CMS gobernado (roles y permisos)
- Manual corto de publicación (1–2 páginas)

Gate de salida

- Accesibilidad mínima validada + QA funcional + checklist seguridad CMS (plugins permitidos, hardening).



Fase 2: Módulos transaccionales críticos (denuncias + consultas públicas)

Objetivo: habilitar participación real con trazabilidad y control.

Alcance

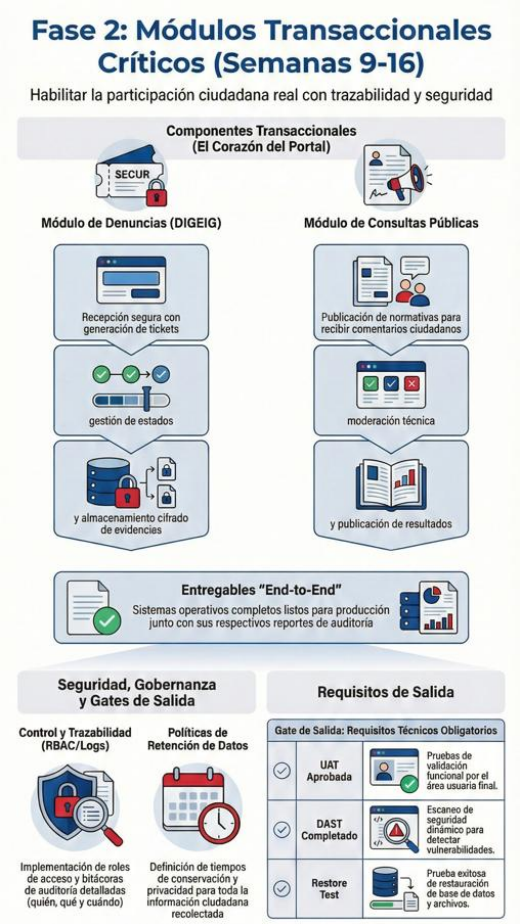
- **Denuncias DIGEIG:** formulario, ticket, estados, adjuntos cifrados, notificaciones
- **Consultas públicas:** publicación de borradores, comentarios, moderación, resultados
- Control de Acceso Basado en Roles (RBAC) interno (roles operativos) + auditoría de acciones
- Políticas de retención y privacidad en formularios

Entregables

- Módulo de denuncias con trazabilidad end-to-end
- Módulo de consultas con moderación y reportes
- Bitácoras y reportes de auditoría (quién/qué/cuándo)

Gate de salida

- UAT aprobada por área usuaria + DAST/validación seguridad + restore test básico (BD/adjuntos).



Fase 3: Integraciones e interoperabilidad (Hub estatal)

Objetivo: conectar el portal al ecosistema sin duplicar sistemas.

Alcance

- Integración/derivación con **SAIP**, **3-1-1**, **CKAN**, **DGCP/MapaInversiones** (según disponibilidad)
- Manejo de "modo degradado" (mensajes claros cuando un tercero falle)
- Panel de estado de integraciones (health-check)

Entregables

- Conectores/API y documentación técnica
- Monitoreo de integraciones + bitácora de fallas
- Evidencia de pruebas de integración

Gate de salida

- Pruebas integrales + acuerdos operativos (SLA del tercero y plan de contingencia).

Fase 4 — Analítica, transparencia operativa y madurez (optimización)

Objetivo: elevar confianza y control institucional mediante datos, indicadores y mejora continua.

Alcance

- Dashboards avanzados (uso del portal, tiempos, volumen agregado, calidad)
- Optimización performance (caching, tuning BD, WAF reglas finas)
- Fortalecimiento DevSecOps (SAST/DAST recurrente, pipeline con gates)
- Simulacros trimestrales de incidentes y continuidad

Entregables

- Tablero ejecutivo mensual (SLA + seguridad + operación)
- Plan de mejora continua con backlog priorizado
- Evidencias de auditoría y simulacros

Gate de salida

- KPI operativos estables + reducción de incidentes recurrentes + cumplimiento de evidencias

Métricas de éxito recomendadas (para BID / Dirección)

- **Disponibilidad mensual** ≥ 99.5%
- **% despliegues exitosos** ≥ 95% (sin rollback)
- **MTTR** (recuperación) medible y en mejora
- **UAT aprobadas** vs rechazadas
- **Vulnerabilidades críticas**: 0 abiertas (o con mitigación documentada)
- **Satisfacción usuaria** (encuesta corta) + reducción de abandono de formularios.


CIERRE Y SOSTENIBILIDAD
Paso de Proceso
Entregables de Gestión Estratégica
Entrega del Tablero Ejecutivo Mensual (SLA) y el Plan de Mejora Continua priorizado.
Statístico
Criterios del Gate de Salida
Estabilización de KPIs operativos, reducción de incidentes y aprobación formal del informe final.
Disponibilidad (Uptime)
≥ 99.5%
Mensual
Despliegues Exitosos
≥ 95%
(sin rollback)
95%
Vulnerabilidades Críticas
0
Abiertas
Secured

Cronograma por fases sugerido para la Implementación de las fases.

Plan de trabajo (Cronograma Gantt)

Leyenda

- S = Semana
- Dependencia = tarea que debe estar completada antes de iniciar (ruta crítica).

Fase / Actividad	Inicio	Fin	Dependencia	Entregable principal	Hito
Fase 0 – Preparación y gobierno	S1	S4	—	Arquitectura + Gobierno + Ambientes + CI/CD + WAF base	**M1**
0.1 Definición de alcance, roadmap y criterios de aceptación	S1	S2	—	Backlog priorizado + criterios de aceptación	
0.2 Arquitectura objetivo (capas) + decisión CMS (mono/híbrido/headless)	S1	S3	0.1	Documento arquitectura	
0.3 Gobierno y controles (RACI, cambios, incidentes, logs, privacidad)	S2	S4	0.1	Políticas mínimas + plantillas (UAT/Go-NoGo/SLA)	
0.4 Ambientes DEV/QA/PROD + CI/CD + repositorios	S2	S4	0.2	Ambientes listos con pipeline inicial	
0.5 Seguridad base: WAF, TLS, MFA, hardening inicial	S3	S4	0.4	Controles base activos	
Fase 1 – Portal informativo + CMS	S5	S8	**M1**	Portal informativo + CMS gobernado	**M2**
1.1 Diseño UX/UI + accesibilidad (WCAG 2.1 mínimo)	S5	S6	M1	Prototipos + guía UI	
1.2 Implementación CMS WordPress + roles editoriales + flujos	S5	S7	M1	CMS operativo (QA/PROD)	
1.3 Frontend base (home, secciones, navegación, buscador)	S6	S8	1.1	Portal informativo funcional	
1.4 Contenido inicial (páginas, FAQ, guías, política privacidad)	S6	S8	1.2	Contenido mínimo publicado	
1.5 QA + UAT Fase 1 + Go/No-Go y publicación	S8	S8	1.3,1.4	Acta UAT + publicación	**M2**
Fase 2 – Módulos transaccionales críticos	S9	S16	**M2**	Denuncias + Consultas Públicas + trazabilidad	**M3**

2.1 Diseño funcional detallado (flujos, estados, roles, datos)	S9	S10	M2	Especificación funcional + reglas	
2.2 Modelo de datos + retención/clasificación + cifrado adjuntos	S9	S11	2.1	Modelo BD + políticas de datos	
2.3 Desarrollo módulo Denuncias (ticket, estados, adjuntos, notifs)	S11	S14	2.2	Denuncias end-to-end (QA)	
2.4 Desarrollo módulo Consultas Públicas (publicación, comentarios, mod)	S11	S14	2.2	Consultas end-to-end (QA)	
2.5 Auditoría/logs + RBAC/MFA admin + hardening app	S12	S15	2.3/2.4 (parcial)	Trazabilidad completa	
2.6 QA integral + DAST (QA) + UAT + Go/No-Go	S15	S16	2.3,2.4,2.5	Acta UAT + publicación módulos	**M3**
Fase 3 – Integraciones e interoperabilidad	S17	S20	**M3**	Conectores SAIP/311/CKAN/DGCP + monitoreo	**M4**
3.1 Diseño técnico de integraciones (APIs, timeouts, modo degradado)	S17	S18	M3	Especificación técnica integraciones	
3.2 Implementación conectores (según disponibilidad de terceros)	S18	S19	3.1	Integraciones en QA	
3.3 Health-check + monitoreo integraciones + bitácora fallas	S18	S20	3.2	Dashboard de integraciones	
3.4 QA + pruebas integrales + UAT + Go/No-Go	S20	S20	3.2,3.3	Publicación de integraciones	**M4**
Fase 4 – Analítica, optimización y madurez	S21	S24	**M4**	Dashboards ejecutivos + optimización + cierre	**M5**
4.1 Analítica y tableros (uso, SLA, volumen agregado, tiempos)	S21	S22	M4	Dashboard mensual ejecutivo	
4.2 Optimización performance (caching, tuning BD, WAF reglas finas)	S21	S23	M4	Mejoras de rendimiento	
4.3 DevSecOps maduro (SAST/DAST recurrente, SCA, pipeline gates)	S22	S24	M4	Seguridad continua operativa	
4.4 Simulacro incidente + restore test formal + cierre	S24	S24	4.1–4.3	Informe cierre + plan mejora continua	**M5**

2) Hitos (milestones) y "gates"

- **M1 (Fin S4):** Ambientes DEV/QA/PROD listos + CI/CD inicial + WAF/TLS/MFA base + documentos de gobierno.
- **M2 (Fin S8):** Go-Live del **Portal informativo + CMS** (contenido mínimo publicado) con UAT aprobada.
- **M3 (Fin S16):** Go-Live de **Módulos transaccionales críticos** (Denuncias + Consultas) con auditoría/logs y UAT.
- **M4 (Fin S20):** Go-Live de **Integraciones** (SAIP/311/CKAN/DGCP/MapaInversiones según aplique) + monitoreo.
- **M5 (Fin S24): Analítica + optimización + cierre** (dashboard ejecutivo, simulacro, restore test, plan mejora continua).

3) Dependencias (ruta crítica resumida)

1. **Ambientes + CI/CD + Seguridad base (M1)**
2. **CMS/Portal informativo (M2)**
3. **Módulos críticos (M3)**
4. **Integraciones (M4)**
5. **Analítica/Optimización/Cierre (M5)**

4) Mini-Gantt ASCII

Cada bloque "[redacted]" representa 1 semana.

Semanas: 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

- Fase 0 (Prep+Ambientes): [redacted]
- Fase 1 (CMS+Portal): [redacted]
- Fase 2 (Módulos): [redacted]
- Fase 3 (Integraciones): [redacted]
- Fase 4 (Analítica+Optim.): [redacted]

Matriz de riesgos y controles

Riesgo	Escenario	Impacto	Controles preventivos	Controles detectivos	Controles correctivos	Evidencia requerida
Ataques a la aplicación (OWASP Top 10)	SQLi/XSS/CSRF, fuerza bruta	Alto	WAF, rate limiting, validación server-side, hardening	Alertas WAF, DAST, logs HTTP	Bloqueo IP/reglas, parche urgente, hotfix	Reporte WAF, resultados DAST, bitácora incidente
Acceso no autorizado a módulos internos	Credenciales comprometidas	Alto	MFA, RBAC, mínimo privilegio, rotación credenciales	Alertas IAM, detección de anomalías	Revocar sesiones, reset credenciales, investigación	Logs IAM, acta revocación, reporte de análisis
Exposición de datos sensibles (denuncias/adjuntos)	Descarga/filtración	Alto	Cifrado en reposo, control de acceso por rol, segmentación	Alertas de acceso a repositorio, auditoría	Revocar accesos, rotación llaves, notificación interna	Evidencia cifrado, auditoría accesos, informe
Integraciones fallidas (SAIP/311/CKAN)	API caída o cambios	Medio/Alto	Timeouts, reintentos, colas, versionado API	Monitoreo de endpoints, health checks	Failover/"modo degradado", corrección con proveedor	Dashboard monitoreo, bitácora de fallas
Indisponibilidad del portal	Sobrecarga/DoS/falla infraestructura	Alto	WAF anti-bot, caching, balanceo, capacidad mínima	APM/uptime, alertas CPU/RAM	Escalamiento, rollback, DR si aplica	Reporte uptime, métricas, acta de restauración
Vulnerabilidad por plugins/temas (CMS)	WordPress desactualizado	Alto	Política de plugins, listas permitidas, staging, parches	Escaneo CVE, revisión mensual	Actualización controlada, desinstalar plugin	Inventario plugins, reporte parches, acta cambios
Pérdida/corrupción de base de datos	Error humano o falla disco	Alto	Backups, replicación (si aplica), controles de cambios	Monitoreo BD, alertas integridad	Restauración probada, reconstrucción índices	Evidencia backups, prueba restore, reporte BD
Manipulación de contenido público	Defacement / edición no autorizada	Medio/Alto	Roles editoriales, aprobación, MFA, WAF	Auditoría CMS, alertas cambios	Revertir versión, bloqueo usuario, hardening	Registro de cambios CMS, capturas, acta
Falta de trazabilidad/auditoría	No se registra "quién hizo qué"	Alto	Logs centralizados, auditoría obligatoria	Revisiones periódicas, alertas sin logs	Ajuste configuración, retención reforzada	Política logs, muestra de eventos, retención
Brecha de privacidad por formularios	Recolección excesiva o exposición	Alto	Minimización de datos, campos obligatorios limitados, aviso privacidad	Revisiones de cumplimiento, pruebas	Ajustes formularios, anonimización/retención	Política privacidad, versionado formularios

Errores en despliegues	Cambios rompen producción	Alto	CI/CD con gates, QA/UAT, Go/No-Go	Monitoreo post-deploy, smoke tests	Rollback inmediato, corrección y re-deploy	Checklist UAT, acta Go/No-Go, evidencia rollback
Dependencia de un solo proveedor	Bloqueo por licencias/soporte	Medio	Estándares abiertos, documentación, contratos SLA	Indicadores SLA, revisiones	Plan alterno/dual-sourcing	SLA, informes mensuales, plan contingencia