Eusko Jaurlaritzaren Informatika Elkartea
Sociedad Informática del Gobierno Vasco

EJIE-2026-097

Servicio de Centro de Excelencia Cloud para EJIE

Pliego de condiciones técnicas

Junio de 2026

Este documento es propiedad de Eusko Jaurlaritzaren Informatika Elkartea – Sociedad Informática del Gobierno Vasco, S.A. (EJIE). Este documento no puede ser reproducido, en su totalidad o parcialmente, ni mostrado a otros, ni utilizado para otros propósitos que los que han originado su entrega, sin el previo permiso escrito de EJIE. En el caso de ser entregado en virtud de un contrato, su utilización estará limitada a lo expresamente autorizado en dicho contrato. EJIE no podrá ser considerada responsable de eventuales errores u omisiones en la edición del documento.

Versión	Fecha	Resumen de cambios	Elaborado por:	Aprobado por:
	20/04/2026	Primera versión	Javier Cela	
1.0	26/06/2026	Versión Definitiva	Javier Cela	Iratxe Montoya

Índice

1 Introducción 5
2 Antecedentes 6
3 Objeto del contrato 8
4 Alcance del servicio y líneas de trabajo 9
4.1 Gobierno Cloud y mejores prácticas 9
4.2 Capacitación de equipos 9
4.3 Implantación Modelo Operativo Cloud 9
4.4 Plataforma de Ingeniería cloud, Arquitectura cloud, landing zones y building blocks 10
4.5 Infraestructura como Código (IaC) y prácticas GitOps y DevOps 10
4.6 Asistencia Técnica en despliegues de cloud y arquitectura de soluciones 11
4.7 Seguridad Cloud y cumplimiento normativo 11
4.8 FinOps 12
4.9 Site Reliability Engineering (SRE), observabilidad y evolución hacia AIOps 12
4.10 Estrategia de Disaster Recovery y Continuidad de Negocio 12
5 Detalle Modelo Operativo 14
5.1 Modelo Operativo Objetivo 14
5.2 Modelo de adopción a 3 velocidades/niveles DevOps 14
5.2.1 Nivel 1 – “Marco Mínimo” 14
5.2.2 Nivel 2 – “Marco Guiado” 15
5.2.3 Nivel 3 – “DevOps GoldenPaths” 16
5.2.4 Evaluación y Transición entre niveles 18
5.2.5 Resumen Comparativo de los niveles 18
5.2.1 Adaptación de los niveles 19
5.3 Implantación del modelo 20
5.4 Gobernanza Seguridad y Trazabilidad del modelo 21
6 Marco de Seguridad Cloud y Cumplimiento Normativo 22
6.1 Capa 1 — Gobernanza de seguridad cloud 22
6.2 Capa 2 — Gestión de identidades y accesos (IAM) 22
6.3 Capa 3 — Segmentación de red y perímetro 23
6.4 Capa 4 — Seguridad de cargas de trabajo 24
6.5 Capa 5 — Seguridad de datos 25
6.6 Capa 6 — Gestión de postura de seguridad y monitorización continua 26
6.7 Aplicación por velocidades del modelo de adopción 26
6.8 Tecnologías de referencia por hiperescalar 27
7 Infraestructura como Código (IaC), prácticas GitOps y DevOps 28
7.1 Enfoque general y principios rectores 28
7.2 Gobierno y curación de la biblioteca de IaC 28
7.3 Responsabilidades diferenciadas en la creación y mantenimiento de IaC 30

7.4 Herramientas de IaC y selección tecnológica 31
7.5 Prácticas de ingeniería de software aplicadas a la IaC 32
7.6 Policy as Code (PaC) y guardrails 33
7.7 Autoservicio gobernado y catálogo de servicios cloud 33
7.8 Gestión de recursos compartidos 34
7.9 Comunidad de práctica y mejora continua 34
7.10 Entregables del adjudicatario 34

8 FinOps – Gestión Financiera del Cloud 36

8.1 Dominio 1 — Aprender: estándares de coste, retroalimentación y biblioteca de patrones 36
8.2 Dominio 2 — Diseño consciente de costes 37
8.3 Dominio 3 — Estimar: modelo de costes, presupuestos y controles en entregas 38
8.4 Dominio 4 — Seguir y optimizar: monitorización continua y optimización de costes 40
8.5 Dominio 5 — Habilitar: showback, chargeback y cultura de coste 41
8.6 Aplicación por niveles del modelo de adopción 42
8.7 Reporting y Entregables 42

9 Site Reliability Engineering (SRE), observabilidad y evolución hacia AIOps 43

9.1 Estrategia de observabilidad 43
9.2 Práctica SRE: SLOs, error budgets y mejora continua 45
9.3 Paneles, alertas y visualización 46
9.4 Evolución hacia AIOps 46
9.5 Gobernanza de AIOps 48
9.6 Aplicación por niveles del modelo de adopción 48
9.7 Entregables 49

10 Plataforma de Ingeniería Cloud: catálogo de building blocks 50

10.1 Enfoque y Principios 50
10.2 Estructura y Catálogo 50
10.3 Patrones de arquitectura de referencia (composiciones de building blocks) 57
10.4 Publicación en BatArin y modelo de consumo 58
10.5 Hoja de ruta de construcción del catálogo 59
10.6 Mantenimiento y Evolución 59

11 Configuración del Servicio 60

11.1 Composición del Equipo de Trabajo. Roles, responsabilidades y dedicación 60
11.2 Garantía de Disponibilidad del Servicio 62
11.3 Plan de Transición 63

12 Metodología de Trabajo y entorno tecnológico 64

12.1 Metodología del CCoE 64
12.2 Uso de Inteligencia Artificial para potenciar a productividad del equipo 64
12.3 Entorno Tecnológico 67

1 Introducción

EJIE, Eusko Jaurlaritzaren Informatika Elkartea – Sociedad Informática del Gobierno Vasco, es la Empresa pública de servicios de las tecnologías de la información y las comunicaciones (TIC), cuya razón de existir es contribuir a la consecución de un Sector Público Vasco, moderno y eficiente, en el Marco Legal establecido por el Gobierno, con la seguridad y calidad necesarias y con el debido respeto al medio ambiente.

EJIE tiene como meta final la consecución de la satisfacción de sus clientes, siendo el instrumento común de prestación de servicios TIC en el Sector Público Vasco, y comprometiéndose en:

- Construir y mantener con eficiencia y calidad la infraestructura de los Sistemas de Información, posibilitando su continuidad y seguridad.
- Garantizar la interoperabilidad entre las distintas administraciones.
- Servir de apoyo a las necesidades de planificación y realización de la función informática de los Departamentos y Organismos Autónomos del Gobierno, asegurando la cobertura de sus demandas con el compromiso y profesionalidad adecuados a las relaciones contractuales que se establezcan.

Por tanto, EJIE debe ser, un instrumento común de referencia para la prestación de servicios TIC en el Sector Público Vasco:

- Aportando valor añadido.
- Proporcionando soluciones competitivas.
- Transmitiendo confianza a sus clientes.
- Contando con personas cualificadas y comprometidas.

Se puede obtener información más detallada y extensa en nuestra dirección de Internet http://www.ejie.eus

Con fecha 16 de agosto de 2016 se firmó el Acuerdo entre el Ente Público de Derecho Privado Osakidetza y Eusko Jaurlaritzaren Informatika Elkartea—Sociedad Informática del Gobierno Vasco, S.A. (EJIE) para la integración del servicio de infraestructuras y operaciones (I.O.) de OSAKIDETZA con el análogo servicio (I.O.) en EJIE, en el marco del acuerdo adoptado por el Consejo de Gobierno sobre el proceso de convergencia en materia de tecnología de la información y comunicación – TIC/ICPS el 27 de julio de 2015 y del Acuerdo de 21 de junio de 2016 relativo a dicho proceso de convergencia por el que se aprueba el documento ejecutivo “plan general de actuación en materia de convergencia TIC/ICPS”.

2 Antecedentes

En EJIE (Eusko Jaurlaritzaren Informatika Elkartea – Sociedad Informática del Gobierno Vasco) se aspira a impulsar la transformación digital y modernización de las aplicaciones utilizando el cloud público y las arquitecturas cloud-native como aceleradores.

Para ello, se pretende establecer un Centro de Excelencia Cloud (CCoE) que actuará como un centro de recursos y conocimientos, aportando las herramientas y competencias necesarias para maximizar los beneficios de la nube pública/hiperescalares en términos de eficiencia, escalabilidad y agilidad en el desarrollo de aplicaciones y servicios.

Este CCoE se alinea con la Estrategia Cloud del Gobierno Vasco y refuerza el papel de EJIE como habilitador tecnológico en ese contexto. La Dirección de Estrategia Digital del Gobierno Vasco está desplegando una Estrategia Cloud que posiciona a EJIE como gestor tecnológico centralizador de suscripciones cloud públicas, garantizando su gobierno, la aplicación de buenas prácticas de seguridad y la optimización del uso de los servicios cloud de manera coordinada y ordenada.

Fruto de esta apuesta estratégica EJIE y sus clientes han ido consumiendo gradualmente más suscripciones de cloud público, pero además se prevé que ese uso aumente ya que por la evolución del mercado es cada vez más habitual que las soluciones que se buscan para dar respuesta a las necesidades de los clientes tengan su mejor encaje posible en soluciones de este tipo, por lo que en 2023 se licitó el contrato EJIE-082-2023 “Suministro de suscripción multicloud a cloud hiperescalares y solución de analítica de datos sobre cloud”, así como otro que sucedió a éste y actualmente en vigor EJIE-2026-014 "Compra conjunta de suministro de suscripciones a cloud hiperescalares y servicios asociados". En la actualidad en EJIE, de los hiperescalares en el mercado, AWS es el principal ya que se sobre él 3/4 partes del consumo anual.

En la actualidad, AWS constituye el principal proveedor de cloud público de EJIE, concentrando aproximadamente tres cuartas partes del consumo anual realizado sobre plataformas hiperescalares. Esta situación ha propiciado que un volumen muy significativo de cargas de trabajo, arquitecturas, procedimientos operativos, automatizaciones, capacidades de seguridad, mecanismos de monitorización y conocimiento técnico especializado se encuentre alineado con los servicios y tecnologías de AWS. En consecuencia, una eventual migración a otro proveedor hiperescalar supondría un proyecto de transformación de gran complejidad técnica, organizativa y económica, requiriendo la adaptación o reconstrucción de numerosos componentes, la revalidación de arquitecturas y controles de seguridad, la formación de los equipos implicados y la ejecución de procesos de migración con el consiguiente impacto en costes, plazos y riesgos operativos. Por todo ello la sustitución de AWS, total o parcial, no resultaría abordable a corto plazo sin incurrir en importantes esfuerzos, riesgos y costes de transición.

La entidad adjudicataria deberá colaborar estrechamente con el personal de EJIE asumiendo la responsabilidad técnica operativa del CCoE y ejecutando las actividades definidas, con un fuerte carácter prescriptivo, es decir, especificando detalladamente los requisitos, metodologías y entregables para garantizar la consistencia y cumplimiento de la estrategia corporativa de adopción cloud y modernización de aplicaciones. Se espera por tanto que el proveedor aporte experiencia especializada en gobierno cloud, arquitectura, seguridad y automatización para llevar a la práctica todas las funciones del CCoE de manera conforme a las directrices de EJIE.

De forma complementaria a este equipo existirá otro equipo para la operación y soporte de los entornos cloud que contará con una cobertura horario extendida velando por la disponibilidad y que tendrá en su alcance las labores más operativas del día a día, ejecución de tareas, solución de incidencias y problemas,

labores de SRE y remediación, implementar y hacer cumplir las políticas de FinOps, reporting y observabilidad, etc. Estos equipos funcionarán de forma coordinada con el CCoE ya que acogerá en general a las políticas definidas y a la ingeniería de plataforma implementada por este último.

3 Objeto del contrato

El objeto del presente contrato es la contratación de un servicio integral para la creación y gestión operativa de un Centro de Excelencia Cloud (en adelante CCoE) de EJIE. Esto incluye la aportación de un equipo especializado por parte de la entidad adjudicataria, el cual trabajará bajo la coordinación de un Responsable del CCoE designado por EJIE para acompañar y ejecutar las funciones técnicas, de seguridad, de gobierno y de arquitectura definidas en este pliego.

El CCoE se concibe como un centro de recursos y conocimiento destinado a maximizar la eficacia en la adopción de cloud por parte de EJIE y sus clientes, a la vez que minimiza riesgos y garantiza el cumplimiento normativo y de seguridad. Esto se logrará proporcionando herramientas y conocimientos que aceleren la transición a la nube de manera eficiente y segura (por ejemplo, mediante marcos de trabajo estandarizados y best practices).

El CCoE estará integrado como mínimo por los perfiles que se indican en apartado correspondiente a los equipos de trabajo por parte del adjudicatario y por el personal de EJIE que se determine al fin de lograr los fines para los que se impulsa esta iniciativa.

En resumen, el contrato tiene como objetivo la creación, puesta en marcha y gestión operativa del CCoE de EJIE, cubriendo tanto aspectos tecnológicos de ingeniería de plataforma y consultoría (arquitectura cloud, landing zones, automatización, etc.) como organizativos y de gobierno (modelo operativo, formación,marco de desarrollo para IaC y GitOps etc.), tal y como se detalla en los apartados siguientes.

Asimismo, el objeto del presente contrato incluye el mantenimiento, actualización, evolución e impulso del marco de desarrollo estándar para la Infraestructura como código (MD-IaC) para los entornos cloud del que dispone EJIE. Este marco unifica los procesos de desarrollo y despliegue en EJIE y sus proveedores, mejorando la gobernanza, seguridad y trazabilidad mediante el control centralizado de la Infraestructura como Código. El adjudicatario deberá además mantener y evolucionar el conjunto de arquitecturas de referencia y componentes reutilizables (*building blocks*) del que dispone EJIE que sirven como base común para los proyectos software en la nube. Todas las actividades del CCoE se alinearán con este MD-IaC y con las mejores prácticas de la industria (como por ejemplo el AWS Well-Architected Framework), garantizando estándares de eficiencia, seguridad y calidad homogéneos en el desarrollo cloud de EJIE.

Adicionalmente, se contemplará un modelo de adopción por “tres velocidades” o niveles de madurez DevOps, de forma que cada nueva iniciativa o proyecto cloud de EJIE se clasifique formalmente en uno de esos tres niveles y aplique los procesos y exigencias adecuados a su categoría. Este modelo escalonado permitirá incorporar proveedores y desarrolladores con distinto grado de experiencia en DevOps de manera realista, definiendo para cada nivel: alcance mínimo de automatización e IaC, responsabilidades de EJIE vs. proveedor, requisitos de seguridad/observabilidad, y entregables de evidencias de cumplimiento. (Estos niveles se detallan en los apartados posteriores).

La prestación de los servicios solicitados en este pliego conlleva el cumplimiento de los Acuerdos de Nivel de Servicio entre EJIE y la empresa adjudicataria, que tendrá los ajustes necesarios en función de la ejecución de los servicios. Dichos Acuerdos de Nivel de Servicio estarán alineados con las necesidades actuales de EJIE, siendo necesario su constante seguimiento para garantizar el servicio prestado y con la suficiente flexibilidad para que permita su actualización en función de cambios o nuevos requerimientos de EJIE.

4 Alcance del servicio y líneas de trabajo

El alcance del servicio abarca todas las funciones clave necesarias para implantar y consolidar un modelo de gobierno y explotación cloud robusto y alineado con la estrategia de EJIE. El servicio incluye, entre otras, las siguientes líneas de trabajo principales que engloban las funciones y tareas que el CCoE deberá desarrollar.

Se requerirá durante la ejecución del contrato, que estas líneas de trabajo se desarrollarse con un alto nivel de detalle técnico y contractual, especificando qué actividades concretas, entregables y evidencias de cumplimiento.

En particular, para las funcionalidades relacionadas con DevOps y GitOps, se aplicará un modelo de adopción por tres “velocidades” o niveles de madurez (descritos en el siguiente apartado 5) que determinará diferentes exigencias y reparto de responsabilidades en función del nivel de cada proyecto cloud, asegurando así una adopción progresiva y escalonada según la madurez de los equipos implicados.

4.1 Gobierno Cloud y mejores prácticas

Definir, implantar y mantener las políticas, estándares y guardrails para el uso seguro y eficiente del cloud (ej. normativas ENS/Esquema Nacional de Seguridad, directrices internas) y establecer las mejores prácticas en infraestructura y desarrollo sobre la nube. Esto incluye aspectos como control de acceso, seguridad por diseño, cumplimiento regulatorio y adherencia a frameworks prescritos por los hiperescalares.

4.2 Capacitación de equipos

Formar y acompañar a los equipos de desarrollo, ingeniería de otras plataformas y otras áreas de EJIE en la adopción de tecnologías cloud, con programas de formación técnica enfocada en DevOps, SRE, IaC, FinOps, Seguridad Cloud, etc., y actividades de evangelización que impulsen una cultura de mejora continua y uso responsable del cloud tanto en EJIE como en sus clientes. Se espera la creación de planes formativos, talleres prácticos y documentación adaptada a diferentes niveles de madurez.

4.3 Implantación Modelo Operativo Cloud

Diseñar y desarrollar un modelo operativo específico para entornos cloud en EJIE, que defina procesos, flujos de trabajo, DevOps Golden Paths, procedimientos de gestión de demanda y de gobernanza del marco de desarrollo estándar (MD-IaC), asegurando la integración con los órganos y procesos de gestión IT existentes en EJIE, implementados sobre ServiceNOW.

Este modelo deberá detallar cómo se integrarán las iniciativas cloud en la operativa diaria, cómo se priorizarán las demandas y cómo se asegurará la coordinación con otros equipos (ingeniería de otras plataformas, Operaciones, equipos de soporte, Gestión de Iniciativas, SecOps, etc.).

Las labores de operación diaria y soporte específicos de la plataforma cloud y de los entornos desplegados en cloud quedan fuera del alcance de este contrato y se desempeñan por un equipo específico diferenciado.

El adjudicatario no deberá confundir y solapar estas funciones, pero sí prescribir el modelo de relación, reparto de responsabilidades y alcance al establecer, documentar y mantener actualizado el modelo operativo.

4.4 Plataforma de Ingeniería cloud, Arquitectura cloud, landing zones y building blocks

Los recursos cloud se habilitarán para el consumo de los diferentes actores implicados como una plataforma de ingeniería de infraestructura según un paradigma de Infraestructure Platform Engineering (IPE).

El adjudicatario deberá asumiendo esa función de ingeniería de la plataforma tratar las capacidades de plataforma cloud (landing zones, catálogos, building blocks, autoservicio) como un producto interno, con backlog, roadmap y mejora iterativa basada en feedback de los equipos consumidores y otras plataformas de ingeniería que se apoyen en la de cloud con la supervisión del referente que EJIE establezca como "Product Owner" de la misma.

En el diseño y evolución de la arquitectura cloud corporativa se contemplarán principios de multicloud e hibridación, priorizando AWS como nube principal (manteniendo y evolucionando su landing zone ya existente) y GCP y Azure como secundarias.

La empresa adjudicataria deberá por tanto asumir la landing zone de AWS actualmente en producción, así como las landing zone de GCP y Azure, garantizando su consolidación y evolución iterativa en el marco de este contrato, incorporando mejoras de seguridad, automatización e integración según los estándares EJIE. Asimismo, definir y formalizar arquitecturas de referencia en forma de building blocks reutilizables (por ejemplo, patrones de arquitectura multi-tier, contenedores/Kubernetes, serverless, etc.), organizados en un catálogo accesible a través del portal del desarrollador (IDP) de EJIE. Estos building blocks deberán utilizar enfoques de Platform Engineering con IaaS/PaaS en autoservicio para permitir despliegues ágiles, escalables y consistentes tal como se indica anteriormente.

En resumen, por un lado, el CCoE define principios, políticas, guardrails y arquitecturas de referencia, que por otro su función de ingeniería materializa en capacidades reutilizables (building blocks, catálogos, autoservicio controlado) que facilitan, aceleran y reducen la carga cognitiva de los equipos consumidores. Esos equipos consumidores por su parte según su madurez seguirán disciplinas más tradicionales o más DevSecOps para consumiendo la plataforma cloud poder entregar software de forma continua y segura, integrando validaciones de seguridad y calidad en CI/CD y controles en runtime. Estas aplicaciones e iniciativas de negocio finalmente estarán acompañadas, en mayor o menor alcance según el caso, por la cobertura de los equipos de soporte de los que dispone EJIE.

En el apartado 10 se desarrolla y detalla una propuesta inicial de catálogo de building blocks para que sirva como referencia.

4.5 Infraestructura como Código (IaC) y prácticas GitOps y DevOps

Actualizar, impulsar y evolucionar el marco de desarrollo (en adelante MD-IaC) establecido en EJIE que estandariza el uso de la Infraestructura como Código (IaC) en entornos cloud. Este marco integra las plantillas y módulos de IaC (principalmente con Terraform), integrados con los pipelines CI/CD corporativos (GitLab), siguiendo un modelo GitOps.

Las plantillas deberán alinearse con mejores prácticas (p.ej., mantener la separación plan/apply, usar OIDC, asegurar el estado de Terraform). El adjudicatario deberá implantar, aplicar y mantener este marco en la organización, abordando: convenciones de naming y tagging, gestión de secretos, bibliotecas de módulos reutilizables, y CI/CD pipeline con validaciones automáticas de seguridad y calidad en cada push (e.g., linting, análisis SAST/DAST, integration tests, etc.). Esta línea de trabajo incluye la gobernanza del marco: asegurar su adopción por todos los proyectos cloud y actualizarlo continuamente conforme a nuevas mejores prácticas (por ejemplo, integración de recomendaciones del Well-Architected Framework, best practices de AWS y GCP, etc.)

En el apartado 7 se desarrollan las responsabilidades del CCoE en este ámbito y sus principios rectores.

4.6 Asistencia Técnica en despliegues de cloud y arquitectura de soluciones

Acompañar a los proyectos de EJIE y sus clientes en la adopción de cloud, prestando asistencia técnica experta en el diseño y despliegue de soluciones sobre entornos cloud. Esto incluye apoyar tanto nuevos proyectos (diseño de arquitecturas cloud, dimensionamiento, selección de servicios adecuados) como servicios existentes migrados o extendidos a la nube, con especial atención a entornos híbridos (brindando soporte a la integración cloud-on-premises, aplicaciones que se quieran modernizar desde arquitecturas tradicionales y a la arquitectura de la parte cloud de entornos híbridos cuando sea necesario). El adjudicatario actuará como líder técnico de referencia para garantizar que los proyectos cloud se implementen siguiendo los patrones, seguridades y frameworks establecidos por el CCoE.

4.7 Seguridad Cloud y cumplimiento normativo

Implementar las medidas de seguridad en cloud necesarias para asegurar que todos los entornos y prácticas de la nube cumplan con los requisitos del Esquema Nacional de Seguridad (ENS) y estándares ISO/IEC 27001 aplicables. Esto abarca la adopción de DevSecOps (seguridad “shift-left” integrada en el ciclo de vida), la gestión integrada de identidades (IAM), la segregación de entornos, la protección de datos y la conformidad normativa (protección de datos personales, RGPD), con auditorías y logs centralizados. El CCoE definirá controles de seguridad por diseño en la infraestructura y en las pipelines, como autenticación multifactor, niveles de encriptación en datos sensibles, segmentación de red o envío de logs a la plataforma SIEM de EJIE para monitorización y alertas en tiempo real.

El CCoE deberá diseñar, implementar y mantener un marco integral de seguridad para la plataformas cloud de EJIE, siguiendo una arquitectura de defensa en profundidad estructurada en capas lógicas. Este marco garantizará que todos los entornos y prácticas cloud cumplan con los requisitos del Esquema Nacional de Seguridad (ENS), los estándares ISO 27001/27017/27018 y las directrices internas de ciberseguridad de EJIE.

El modelo de seguridad se articulará en torno a seis capas que, en conjunto, proporcionan una protección completa de los entornos IaaS y PaaS: gobernanza de seguridad, gestión de identidades y accesos, seguridad de red y perímetro, seguridad de cargas de trabajo, seguridad de datos, y gestión de postura de

seguridad con monitorización continua. Cada capa se detalla y desarrolla en el apartado 6 con las responsabilidades específicas que asumirá el CCoE.

4.8 FinOps

Supervisar, optimizar y reportar sobre la eficiencia de costes cloud en soluciones desplegadas. El CCoE deberá implementar prácticas de FinOps para asegurar la transparencia, control y optimización de los gastos en cloud, mediante la monitorización continua del consumo, la adopción de estrategias de right-sizing y auto-scaling que minimicen costes, la identificación de recursos infrautilizados o "huérfanos" para su apagado, y la prescripción de medidas de optimización de costes a los equipos de proyectos (por ejemplo, aprovechamiento de instancias reservadas/spot, adopción de arquitecturas serverless cuando sean más económicas, etc.). Deben definirse y cumplirse KPIs financieros, como umbrales de costes por proyecto, y emitir informes regulares, así como canalizar la información relevante a los sistemas de facturación con los que EJIE repercute el consumo a sus clientes.

Actualmente EJIE, para esta labor ya tiene implantada una herramienta específica de gestión financiera (Cloudability) en la que el adjudicatario se apoyará e integrará en su marco de trabajo.En la parte operativa de la gestión financiera de la nube también participará el soporte correspondiente identificado en el punto 4.3.

La práctica FinOps del CCoE se articulará en torno a un ciclo continuo de cuatro dominios que se detallan y desarrollan en el apartado 8 con las responsabilidades específicas que asumirá el CCoE.

4.9 Site Reliability Engineering (SRE), observabilidad y evolución hacia AIOps

Integrar prácticas de SRE y observabilidad en los servicios cloud, asegurando monitorización continua, logging, trazabilidad e indicadores de rendimiento para todas las soluciones desplegadas. Esto implica que, en cada nivel de madurez (ver Velocidades en el apartado 5), se cumplan los niveles mínimos de observabilidad y métricas exigidas. Se implementarán soluciones de monitorización corporativas, integrando logs y métricas de aplicaciones y servicios cloud en la plataforma central (por ejemplo, mediante CloudWatch o Google Operations integrados con herramientas de monitorización o la plataforma de observabilidad corporativa), habilitando alertas y dashboards de rendimiento, disponibilidad, latencia y errores. Se fomentará asimismo la automatización SRE (definición de service-level objectives – SLOs – y error budgets, retroalimentando la mejora continua)

Esta observabilidad no es un fin en sí misma: es la infraestructura de datos sobre la que se sustenta tanto la práctica SRE (para tomar decisiones basadas en datos sobre fiabilidad) como la evolución futura hacia AIOps (para que los modelos de IA puedan analizar, correlacionar y actuar sobre esa telemetría). Por tanto, el diseño de observabilidad que realice el CCoE deberá contemplar desde el inicio los requisitos de calidad, cobertura y normalización de datos que una plataforma AIOps necesitará para funcionar eficazmente.

En el apartado 9 se desarrollan las responsabilidades que deberá asumir en este ámbito.

4.10 Estrategia de Disaster Recovery y Continuidad de Negocio

Potenciar la resiliencia asegurando la viabilidad operativa participando en los planes de recuperación ante desastres y continuidad del negocio. Esto incluye evaluar riesgos, definir objetivos de tiempo de recuperación (RTO), documentar de procedimientos de contingencia y realizar pruebas continuas para minimizar el impacto ante cualquier incidente crítico.

5 Detalle Modelo Operativo

5.1 Modelo Operativo Objetivo

El futuro modelo operativo del servicio deberá consolidar una estructura de gobierno adaptada a entornos nativos, híbridos que podrán estar desplegados en uno o más hiperescalares con EJIE como ente coordinador y el adjudicatario como asistencia técnica especializada. El modelo integrará prácticas de gestión de demanda (recepción y evaluación de solicitudes cloud), escalado de soporte, gestión del pipeline de despliegue de entornos y workflows, y gobierno de la implantación del marco de desarrollo de IaC. Este modelo se articulará en torno a tres niveles de adopción (Velocidad 1, 2 y 3) y definirá cómo los diferentes actores que consuman recursos cloud se integran con el CCoE y la propia plataforma Cloud en función de su madurez DevOps, con procesos diferenciados de entrada y seguimiento según su nivel.

5.2 Modelo de adopción a 3 velocidades/niveles DevOps

Se busca que todos los proyectos e iniciativas de EJIE que se apoyen en el cloud público se puedan incorporar de forma realista conciliando su contexto y la ambición de maximizar los beneficios diferenciales de cloud público en un esquema de mejora continua. Para ello el CCoE definirá un modelo de adopción según tres velocidades, que clasifica, atendiendo a las características y circunstancias del desarrollador/promotor, cada proyecto en uno de tres niveles de madurez DevOps (Velocidad 1, 2 o 3) con exigencias y apoyos diferenciados.

Todos los nuevos proyectos deberán ser por tanto formalmente clasificados en uno de estos niveles antes de su inicio (p.ej., en las fases de evaluación de demanda), en base a criterios como el nivel de automatización y prácticas DevOps actuales del equipo, el nivel de criticidad de la aplicación para el negocio, su complejidad técnica, etc.

En los siguientes subapartados se define cada nivel y sus requisitos, detallando las características y directrices que deben cumplirse en cada nivel de madurez (Vel. 1, 2, 3), abarcando la automatización mínima, prácticas IaC, requisitos de seguridad/observabilidad, entregables exigibles y el punto más importante el reparto de responsabilidades entre los equipos de EJIE (el propio CCoE, la operación cloud, la plataforma de SDLC) y el propio desarrollador/promotor de la iniciativa.

En los casos que el promotor de la iniciativa/desarrollador esté consumiendo cloud pero de forma indirecta porque lo que consume es otra de las plataformas de ingeniería proporcionadas de EJIE como la plataforma de aplicaciones, la plataforma de IA, plataforma de Analítica etc … la velocidad le vendrá predefinida ya en la práctica según el contexto particular de uso de cada una de esas plataformas.

5.2.1 Nivel 1 – “Marco Mínimo”

Pensado para iniciativas de equipos tradicionales con escasa experiencia en DevOps. EJIE/CCoE asumen la responsabilidad principal en las tareas de operación y cloud que trasciendan del desarrollo. En concreto,

EJIE diseña y administra los pipelines CI/CD estándar, gestiona los despliegues y la infraestructura base como código, e implementa los controles de seguridad y monitorización básicos. El consumidor se limita a ejecutar su desarrollo dentro de ese marco predefinido: entrega el código fuente o los artefactos de aplicación al repositorio indicado y consume los módulos/plantillas proporcionadas, siguiendo las indicaciones del CCoE. En este nivel, el énfasis está en garantizar un núcleo mínimo de prácticas:

- En términos RACI: EJIE es Responsable (R) y Ejecutor (Do) de la mayoría de procesos (p.ej. ejecución de Terraform, configuraciones CI), mientras el promotor es Consultado (C) en las decisiones de implementación e Informado (I) de los resultados. No se espera que el promotor proponga cambios al pipeline ni altere procedimientos, salvo adaptar mínimamente su código a las guías recibidas.
- Ejemplo: Migración de una aplicación monolítica N-Tier a la nube usando plantillas estándar, con mínima intervención del equipo en el pipeline.

- *Alcance de automatización/IaC:* Se habilitan componentes básicos de Infraestructura como Código (IaC) para la infraestructura fundamental del proyecto (por ejemplo, aprovisionamiento de entornos base y *networking*), aunque es posible que parte del despliegue se realice manualmente dada la capacidad limitada del equipo. La IaC cubre la capa de landing zone y la infraestructura principal (VPC, VM/contenedores, almacenamiento, etc.) pero puede quedar fuera la automatización de componentes de aplicación no críticos.

- *Pipelines CI/CD:* Se proporcionan plantillas sencillas de pipeline con integración continúa limitada al ciclo básico de construcción y despliegue (p.ej., *build* y *deploy* manual), no exigiendo todavía una pipeline completamente automatizada ni integraciones complejas. En este nivel no se requiere una integración continua avanzada – bastará con un ciclo de build/test simple sin necesidad de despliegues automatizados o *rollbacks* automáticos.

- *Requisitos de seguridad/observabilidad: Mínimos obligatorios*, enfocados en la seguridad básica y la monitorización fundamental. Por ejemplo, asegurar que se integran las cargas en los controles de acceso corporativos, se aplican configuraciones seguras predeterminadas en la infraestructura (p.ej., *security groups*, cifrado por defecto) y se capturan logs y métricas básicas en la plataforma de observabilidad común.

- *Entregables y evidencias para cada iniciativa:* Documentación de la arquitectura e IaC realizada, evidencias de revisiones de seguridad (p. ej., resultados de e*scaneos* de vulnerabilidades) y un informe de adopción DevOps que justifique la clasificación en Velocidad 1 y los planes de mejora para evolucionar a nivel superior.

5.2.2 Nivel 2 – “Marco Guiado”

Dirigido a equipos con cierta experiencia técnica que están dispuestos a adoptar (o disponen ya) prácticas DevOps más completas bajo la supervisión del CCoE, siendo por tanto la responsabilidad de la aplicación en producción más compartida. EJIE proporciona el marco (pautas, herramientas y gobierno) y supervisa/valida las actividades clave, pero el adjudicatario participa activamente en las tareas DevOps del proyecto.

Por ejemplo, la definición detallada de un pipeline CI/CD para la aplicación se hace en colaboración: EJIE aporta la plantilla estándar corporativa que se corresponderá con alguno de los building blocks predefinidos (ej. un modelo de .gitlab-ci.yml con stages predefinidos) y el proveedor la extiende o configura para su aplicación bajo la revisión del CCoE. La gestión de la Infraestructura como Código es dual: EJIE mantiene

los módulos Terraform comunes (building blocks) y el estado backend centralizado, mientras que el proveedor codifica las piezas específicas de la aplicación (p.ej. recursos particulares necesarios) reutilizando esos módulos.

- El promotor debe, por tanto, disponer de personal capacitado que tome el rol Ejecutor (Doer) en tareas como ajustes de pipeline, escritura de código Terraform en repos, configuración de alertas básicas, etc., bajo la supervisión y con la autorización final de EJIE (que retiene la Responsabilidad (R) última de cumplimiento del marco).
- Ejemplos: el proveedor ejecuta despliegues en GitLab pero EJIE aprueba los merge requests críticos; el proveedor propone la estructura de monitorización de logs de su aplicación, EJIE valida que cumpla el estándar mínimo.

Las exigencias son mayores:

- *Alcance de automatización/IaC:* Se extiende la IaC a componentes específicos de la aplicación, no solo la infraestructura base. P. ej., la configuración de servicios *PaaS*, redes, *middleware*, etc., se gestiona con código (Terraform, *Ansible* u otra herramienta) almacenado en repositorios de GitLab, es decir siguiendo enfoque parcialmente GitOps.

- *Pipelines CI/CD:* Pipeline CI/CD completa y adaptada al proyecto, incluyendo etapas integrales de compilación, pruebas, control de calidad y despliegue automatizado en entornos no productivos. Se integran las primeras prácticas de seguridad y calidad en el pipeline (por ejemplo, *escaneos* de seguridad, análisis de código, pruebas unitarias obligatorias), y se comienza a implementar pasos de despliegue automatizado en entornos de prueba o *staging*.

- *Responsabilidades EJIE vs adjudicatario:* Responsabilidad compartida. EJIE (a través del CCoE o el equipo de SDLC) proporciona estándares, plantillas y herramientas (por ejemplo, *blueprints* de pipelines, módulos IaC reutilizables) y soporte activo, mientras el adjudicatario asume gradualmente tareas en la configuración de pipelines y en la gestión de IaC para componentes específicos de la aplicación. Es decir, EJIE define y tutela, pero busca que el adjudicatario impulse la adopción de las prácticas en el equipo del proyecto, a la vez que implementa directamente parte de la solución técnica.

- *Requisitos de seguridad/observabilidad:* Observabilidad mínima obligatoria: se requiere al menos la integración de logs y métricas básicas de la aplicación en las soluciones de monitorización definidas por EJIE. En seguridad, se introducen controles adicionales: revisión de configuraciones vs *best practices*,

- *Entregables y evidencias:* Código IaC y pipelines adaptados al proyecto (repositorios actualizados), propuesta de transición para llevar el equipo al siguiente nivel, informes de incorporación de GitOps parcial (documentando que se gestiona ya por repositorio), evidencias de integración de logs/métricas (capturas de dashboards, alarmas definidas), así como informe RACI de responsabilidades compartidas en este nivel.

5.2.3 Nivel 3 – “DevOps GoldenPaths”

Aplicable a equipos avanzados y pioneros – totalmente orientados a DevOps e inmersos en arquitecturas cloud-native (microservicios, contenedores, serverless, event-driven, etc.). En este nivel se aprovecha al máximo el marco de IaC: todo se define “como código” (infraestructura, configuraciones, pipelines y políticas), implementando GitOps completo (cualquier cambio en infra o config se realiza vía repositorios). Los pipelines CI/CD son altamente automatizados y personalizados según las necesidades del proyecto,

incluyendo despliegues frecuentes, tests automatizados extensivos, integraciones de seguridad desde el inicio (DevSecOps), observabilidad avanzada (trazas distribuidas, alertas personalizadas) y prácticas SRE cuando corresponda.

El proveedor asume aquí un rol predominante en las tareas DevOps del día a día (con mayor autonomía para gestionar despliegues y operaciones), siguiendo los "Golden paths" y consumiendo building blocks establecidos por el CCoE para garantizar que no se aparta de las buenas prácticas. EJIE mantendrá una función de gobierno y auditoría: estableciendo las reglas de funcionamiento, revisando métricas de cumplimiento y apoyando en caso de incidencias, pero siendo el equipo promotor de la iniciativa el que se implicará principalmente en la resolución de las mismas.

- Concretamente, el promotor diseña y desarrolla los pipelines CI/CD específicos de la solución (pudiendo ajustar las herramientas a las necesidades siempre que respete estándares de calidad y seguridad), define la infraestructura como código completa de la aplicación y la despliega, incorpora las métricas y alertas de observabilidad avanzadas, etc.
- EJIE (CCoE) en este nivel actúa como Auditor: valida que el diseño propuesto por el adjudicatario cumple los patrones "Golden Path" documentados (p. ej., que se usan los building blocks y módulos homologados, que se implementa federación de identidad OIDC en lugar de credenciales estáticas en los pipelines, que se incluyen trazas y etiquetas obligatorias, etc.), y realiza seguimiento continuo mediante auditorías automáticas y manuales de los despliegues.
- En resumen, el promotor pasa a ser "Owner" de la operación DevOps de ese apliación, y EJIE provee apoyo muy puntual y control a posteriori. Si en la práctica el promotor no mantuviera el nivel requerido, EJIE se reserva la facultad de degradar el proyecto a Nivel 2 para reforzar el control.

Las exigencias en este caso:

- *Alcance de automatización/IaC:* Todo se define como código: infraestructura, configuraciones, pipelines y políticas, implementando una filosofía GitOps completa. Cualquier cambio en infraestructuras o configuraciones se realiza a través del repositorio Git, con auditoría total. *Landing zones*, entornos de ejecución, redes, seguridad, aprovisionamientos: todo se orquesta mediante IaC.

- *Pipelines CI/CD:* Pipelines CI/CD altamente automatizados y personalizables según necesidades del proyecto. Se alcanzan frecuencias de despliegue elevadas (*continuous deployment*), con baterías de pruebas automáticas exhaustivas, integraciones de seguridad *end-to-end* (SAST, DAST, *compliance scanning*), *chaos testing* y despliegues automáticos en entornos productivos con aprobaciones mínimas (sólo *gates* de *compliance*). La pipeline permite *rollback* automático ante fallos y *feature toggles* para lanzamientos controlados.

- *Responsabilidades EJIE vs adjudicatario:* En este nivel, los equipos de proyecto/desarrolladores son autónomos implementando DevOps, actuando EJIE (CCoE y otros equipos implicados) como facilitador y como proveedora de herramientas y coaching, que se entrega por medio de los golden pathsque se consumen, pero la ejecución recae en los equipos del proyecto, aunque con apoyo experto continuo del CCoE externo para casos complejos. La responsabilidad (R) se desplaza en mayor medida al equipo de desarrollo.

- *Requisitos de seguridad/observabilidad:* Máximos. Todos los principios de *seguridad por diseño* y *observabilidad* se aplican: monitorización integral (métricas de negocio, métricas técnicas y *tracing* distribuido) y seguridad integrada en cada componente (desde *infrastructure scanning* hasta *chaos security testing*). Se define y realiza un *Site Reliability Review* de la aplicación antes de cada *release*, verificando SLOs, resiliencia y fallback.

- *Entregables y evidencias:* Infraestructura y aplicaciones 100% definidas como código (repositorios de Terraform y configuraciones revisables), evidencia de pipelines CI/CD avanzadas (p. ej., flujos de *canary releases*, *blue/green deployments*), pruebas de carga y resiliencia realizadas, y reportes SRE de SLOs cumplidos por el servicio. Una parte crucial es el informe de *Golden Path* de cada proyecto, que refleje cómo la solución adopta las prácticas SRE/DevOps recomendadas.

Nota: Este nivel se espera solo en proyectos selectos durante la vigencia inicial del contrato por como es el entorno de desarrolladores y contratistas de EJIE, dado que implica una madurez muy alta; su adopción más generalizada será objetivo a medio plazo conforme los equipos evolucionen.

5.2.4 Evaluación y Transición entre niveles

Cada nueva iniciativa o proyecto cloud que surja durante la ejecución del contrato deberá ser evaluado conjuntamente por EJIE y el adjudicatario para asignarle uno de los niveles anteriores antes de su inicio. Se tomará en cuenta una checklist objetiva con criterios tales como: tecnología y arquitectura de la aplicación (p.ej. monolítica tradicional vs. microservicios o serverless), experiencia previa del equipo de desarrollo/proveedor en prácticas DevOps, criticidad del servicio y frecuencia esperada de despliegues, y requisitos de seguridad/compliance. En caso de duda, primará el nivel más bajo que garantice controlar los riesgos sin frenar el avance del proyecto. Esta clasificación será vinculante: determinará las obligaciones de cada parte,las expectativas técnicas para ese proyecto y el nivel de implicación que puede esperar por parte de otros equipos de EJIE dedicados al soporte y la operación.

El paso entre niveles deberá estar regulado por criterios claros según una normativa de escalado/degradación: los equipos que demuestren madurez y cumplimiento consistente de los requisitos de su nivel podrán optar a pasar al siguiente nivel al que se les clasificara inicialmente; por el contrario, aquellos que se desvíen persistentemente de las normas mínimas podrán ser reclasificados a un nivel inferior, debiendo implementar acciones correctivas con el apoyo del CCoE. Este enfoque de niveles permite una adopción controlada y progresiva del *DevOps* en la organización, reduciendo riesgos y adaptando los requisitos a las capacidades reales de cada proyecto

5.2.5 Resumen Comparativo de los niveles

Aspecto Clave	Nivel 1 – Básico	Nivel 2 – Guiado	Nivel 3 – Golden Path
Infraestructura como Código (IaC)	Cobertura limitada a infraestructura base; IaC en elementos fundamentales y landing zone (VPC, VM, etc.)	IaC para infra y parte de componentes de aplicación; repositorio centralizado GitOps parcial	Todo como código (infra + config + pipelines + políticas); adopción de GitOps completo
Pipelines CI/CD	Pipeline simplificada (build/test básicos);	Pipeline completa (build/test/deploy); integraciones iniciales de	Pipelines avanzados y personalizados; despliegues frecuentes automáticos, tests

Aspecto Clave	Nivel 1 – Básico	Nivel 2 – Guiado	Nivel 3 – Golden Path
	despliegue manual en Prod	seguridad; algunos despliegues automáticos	extensivos, integraciones *security by design*
Responsabilidades	Promotor/ Desarrollador asume la mayoría de tareas técnicas; EJIE supervisa y valida	Tareas compartidas: EJIE (CCoE) provee estándares y soporte; equipo proyecto configura pipelines, IaC de app	Equipo de proyecto es autónomo; CCoE actúa como habilitador experto y auditor
Seguridad & Observabilidad	Controles básicos de seguridad e integración de logs y métricas fundamentales	Integración de logs y monitoreo de *KPIs* básicos del servicio; controles de seguridad ampliados (ej. IAM, escaneos)	Observabilidad completa (trazas, métricas avanzadas, APM); seguridad embebida a lo largo del ciclo de vida, SLOs y *error budgets*
Entregables clave	Documentación de arquitectura & IaC; reporte de adopción DevOps (planes mejora)	Repositorios de IaC & pipeline adaptados; plan de transición; evidencias de GitOps y observabilidad mínima	Repositorios con todo el *stack* (infra+app) codificados; SLOs del servicio; informes SRE y Golden Path

5.2.1 Adaptación de los niveles

Para asegurar la viabilidad y realismo en la implantación del marco, se establecen mecanismos formales para cambiar el nivel asignado a un proyecto en función de la evolución de su equipo o del cumplimiento (o incumplimiento) de las responsabilidades:

- Escalado de Nivel: Cuando un proyecto clasificado inicialmente en Nivel 1 o 2 muestre una mejora sustancial en la madurez de su equipo y en el cumplimiento de los objetivos del MD-IaC, podrá ser propuesto para escalar al siguiente nivel. Esta propuesta podrá hacerla el adjudicatario (p.ej., el Coordinador del Servicio sugiriendo que un proyecto Nivel 1 pase a Nivel 2 tras un ciclo exitoso) o EJIE, en los comités de seguimiento. Los criterios para escalar incluyen, entre otros: indicadores de DevOps estables al alza (tiempo de entrega reducido, calidad de despliegues alta), automatización completa de procesos actuales, equipo del proyecto adquiriendo competencias (formaciones DevOps completadas, certificaciones obtenidas), y ausencia de incidencias de cumplimiento en un periodo significativo. Si EJIE coincide en la viabilidad, se emitirá un Informe de Reclasificación que modifica el nivel del proyecto, detallando los nuevos requisitos que aplican en adelante. El cambio de nivel implicará una transición planificada: por ejemplo, si se pasa de Nivel 1 a 2, se definirá un plan de adopción de pipeline CI/CD ampliado, capacitación adicional al equipo de desarrollo en las nuevas responsabilidades, etc. Durante un periodo de gracia (p.ej. 1-2 meses) se permitirá ciertos ajustes de procedimientos hasta consolidar el nuevo nivel.
- Degradación de Nivel: Si un proyecto en Nivel 2 o 3 incumple sistemáticamente las responsabilidades asociadas a ese nivel, EJIE podrá degradarlo a un nivel inferior para proteger la calidad y seguridad del servicio. Situaciones que pueden detonar una degradación: reiterados errores graves en despliegues o operación que indiquen falta de control (p.ej., caídas de producción por cambios mal manejados en un proyecto Nivel 3), hallazgos de auditoría críticos (p.ej., brechas de seguridad, o infraestructura fuera de IaC en un nivel donde todo debía ser IaC), incumplimiento

de SLOs continuado, o falta de los perfiles/capacidades comprometidas (p.ej., el equipo del adjudicatario sufre rotación y ya no cuenta con un DevOps senior necesario para Nivel 3). En tales casos, EJIE convocará al adjudicatario en un comité extraordinario para notificar las brechas detectadas y dará un plazo corto de corrección. Si tras ese plazo no se logra reconducir la situación, EJIE emitirá una resolución para reclasificar el proyecto en el nivel inmediatamente inferior (por ejemplo, de 3 a 2). La degradación conllevará medidas de mayor control: se reintroducirán aprobaciones y checkpoints que quizá se habían relajado; por ejemplo, un proyecto degradado de 3 a 2 volverá a requerir aprobación manual de despliegues a producción por parte de EJIE, entren tanto el adjudicatario reestablece la confianza. En casos severos, EJIE podría incluso exigir reemplazo de ciertos miembros clave del equipo adjudicatario por perfiles más adecuados, si eso fue factor del incumplimiento. Impacto contractual: una degradación de nivel persistente podría activar cláusulas de penalización si así estuviera pactado en el contrato (por incumplimiento de niveles de servicio en calidad DevOps), aunque inicialmente se tratará como mecanismo correctivo no punitivo.

- Proceso de decisión y comunicación: Cualquier cambio de nivel, sea escalado o degradación, debe ser aprobado por escrito por EJIE. El adjudicatario tendrá oportunidad de presentar sus argumentos o plan de remediación antes de una degradación definitiva. Una vez decidida, se documentará en acta la nueva clasificación, las razones, y la fecha de efecto. Se actualizarán todos los documentos del proyecto (ficha de proyecto, plan de trabajo) y se comunicarán a todos los equipos involucrados las nuevas formas de trabajo y responsabilidades vigentes tras el cambio.
- Frecuencia de revisión de nivel: Independientemente de incidencias, anualmente (o al finalizar cada fase importante del proyecto) EJIE y adjudicatario evaluarán si el nivel asignado sigue siendo apropiado, pudiendo voluntariamente acordar un cambio de nivel si las circunstancias han variado. Esto garantiza la flexibilidad del marco para adaptarse a la realidad evolutiva de los proyectos cloud.

En conclusión, estas reglas aseguran que el modelo de tres niveles/velocidades no es rígido, sino que acompaña la transformación progresiva de los equipos y proyectos: permite acelerar cuando están preparados para asumir más autonomía, y frenar si se detectan riesgos por falta de cultura DevOps. Así, EJIE garantiza la realista implantación del marco en toda su diversidad organizativa, cumpliendo con los objetivos de estandarización, gobernanza, agilidad y reutilización sin comprometer la calidad ni la seguridad en ningún momento.

5.3 Implantación del modelo

El CCoE adoptará mecanismos específicos para aplicar el MD-IaC en cada nivel:

- **Procedimiento de Encaje de Nivel**: Se establecerá un procedimiento estándar para, al inicio de cada iniciativa, aplicar un checklist de nivel y emitir una valoración de Clasificación de Nivel aprobado por el responsable de EJIE del CCoE y comunicado al proveedor/promotor de la iniciativa. Este informe determinará el nivel asignado y enumerará las condiciones particulares de actuación para ese proyecto (p.ej., si es Nivel 1: “pipelines centralizados por EJIE, IaC limitado a X, entregables Y…”).

- **Coordinación y Soporte según Nivel**: El Coordinador del Servicio (CCoE) deberá ajustar la asignación de recursos y funciones en función del nivel. Por ejemplo, en proyectos Nivel 1 se designará un recurso del CCoE que actúe como apoyo principal ejecutando las tareas clave, mientras el proveedor actúa como soporte funcional; en Nivel 2 se conformará un equipo mixto EJIE–proveedor para co-gestionar los pipelines y la infraestructura; en Nivel 3 se esperará que el

equipo del proveedor incluya perfiles DevOps dedicados, con el CCoE dando mentorización y auditoría.

- **Cadencia de Revisión y Mejora Continua**: Independientemente del nivel, el modelo prevé reuniones de revisión periódicas (por ejemplo, retrospectives trimestrales) para evaluar el grado de adopción del MD-IaC en cada iniciativa. El adjudicatario deberá proponer mejoras para que proyectos en Nivel 1 o 2 puedan aspirar a subir de nivel en el futuro si procede, fomentando la maduración progresiva de los proveedores (ej.: formación adicional, incremento paulatino de automatizaciones, etc.). Igualmente, EJIE evaluará si algún proyecto en Nivel 3 requiere retrotraerse a Nivel 2 por cambios en el equipo o desviaciones detectadas – aplicando mecanismos de control para no comprometer la calidad.

5.4 Gobernanza Seguridad y Trazabilidad del modelo

El adjudicatario deberá implementar, en colaboración con EJIE, un **marco operativo de gobierno** que garantice la seguridad y la trazabilidad en todos los niveles del MD-IaC:

- **Políticas y Controles Comunes**: Se definirán políticas corporativas de DevSecOps aplicables a cualquier pipeline o infraestructura desplegada. Por ejemplo, será obligatorio integrar barreras de seguridad (guardrails) automáticas en los flujos de CI/CD e infra, de modo que ningún despliegue que las viole pueda proceder. Estos guardrails incluirán: validaciones de conformidad (ej. análisis de código estático, escaneo de vulnerabilidades de contenedores), chequeos de IaC (linting de Terraform, comprobación de etiquetas obligatorias), y pruebas de seguridad (ej. tests OWASP) en las etapas definidas del pipeline antes de pasar a producción.

- **Gestión de Identidades y Accesos**: Siguiendo las mejores prácticas cloud, se eliminará el uso de credenciales estáticas en los procesos automatizados. El proveedor deberá implementar mecanismos y medidas como las que se detallan en el apartado 6.2.

- **Trazabilidad Completa**: En alineación con el MD-IaC, toda acción sobre la infraestructura o el pipeline debe ser auditable. El adjudicatario configurará las herramientas para que se conserven historiales de cambios (versioning de Terraform state, registro de cada deployment realizado, logs de actividad de usuarios y servicios en GitLab, etc.) y para que se envíen estos registros a los sistemas centrales de EJIE (p.ej., CloudTrail, CloudWatch, o plataformas SIEM) para su monitoreo. Adicionalmente, cada recurso cloud desplegado deberá estar etiquetado según la política de tagging de EJIE (indicando proyecto, propietario, entorno, nivel de criticidad, entre otros), facilitando así la trazabilidad y el FinOps.

- **Revisiones de Seguridad y Cumplimiento**: El CCoE, con participación del adjudicatario, realizará auditorías periódicas (por ejemplo, semestrales) del cumplimiento del marco de seguridad en los proyectos. Se generarán informes donde se verifique: porcentaje de pipelines con OIDC implementado, porcentaje de recursos sin etiquetar, incidencias de seguridad detectadas y resueltas, etc. Cualquier desviación significativa deberá ser corregida en un plazo acordado. Asimismo, se documentarán lecciones aprendidas de incidentes o casi-incidentes, actualizando el marco MD-IaC para prevenir recurrencias (mejora continua).

6 Marco de Seguridad Cloud y Cumplimiento Normativo

El modelo de seguridad se articulará en torno a seis capas que, en conjunto, proporcionan una protección completa de los entornos. Cada capa se detalla a continuación con las responsabilidades específicas que asumirá el CCoE.

6.1 Capa 1 — Gobernanza de seguridad cloud

La gobernanza de seguridad constituye la base sobre la que se sustenta todo el marco. El CCoE será responsable de definir y mantener los controles y guardrails de seguridad que se aplicarán de forma estandarizada a todos los entornos cloud de EJIE.

Responsabilidades del CCoE:

El adjudicatario deberá diseñar y mantener las landing zones de cada hiperescalar (AWS como principal, GCP como secundario, Azure según alcance definido) como fundamento técnico de la gobernanza de seguridad, garantizando que cada nueva cuenta, suscripción o proyecto parta de una configuración segura preestablecida. Esto incluye la definición de la estructura organizativa cloud (jerarquía de cuentas AWS Organizations, carpetas y proyectos GCP, management groups en Azure) con segmentación por unidades de negocio, entornos (desarrollo, preproducción, producción) y requisitos de cumplimiento, de forma que se establezcan fronteras de seguridad claras y se limite el radio de impacto ante incidentes.

El CCoE definirá y aplicará políticas de seguridad a nivel de organización utilizando los mecanismos nativos de cada proveedor (Service Control Policies en AWS, Organization Policies en GCP, Azure Policy), que restrinjan las acciones permitidas en cada nivel de la jerarquía sin conceder acceso per se, actuando como guardrails preventivos. Estas políticas se codificarán como infraestructura como código (IaC) para asegurar su reproducibilidad, auditabilidad y protección frente a la deriva de configuración.

El adjudicatario deberá mantener un catálogo documentado de políticas y guardrails de seguridad, clasificados por nivel de criticidad y ámbito de aplicación, y someterlos a revisión semestral con el equipo de ciberseguridad de EJIE.

Entregables esperados:

El adjudicatario producirá y mantendrá actualizado un documento de diseño de la estructura organizativa cloud con la justificación de la segmentación adoptada, el catálogo de políticas organizativas codificadas como IaC con su correspondiente versionado en GitLab, y un informe semestral de revisión de guardrails que incluya las políticas activas, las excepciones concedidas y las acciones correctivas aplicadas.

6.2 Capa 2 — Gestión de identidades y accesos (IAM)

La gestión de identidades y accesos constituye el perímetro lógico principal de los entornos cloud. El CCoE deberá garantizar que solo las entidades autorizadas — personas, servicios y cargas de trabajo — accedan a los recursos cloud, con los permisos mínimos necesarios.

Responsabilidades del CCoE:

El adjudicatario deberá implementar y mantener la federación de identidades entre el proveedor de identidad corporativo de EJIE (OKTA) y los servicios de identidad de cada hiperescalar (AWS IAM Identity Center, Google Cloud Identity, Microsoft Entra ID según corresponda), asegurando que la gestión de identidades sea centralizada y no se creen cuentas locales en los entornos cloud. Se habilitará Single Sign-On (SSO) para el acceso a las consolas y APIs cloud, eliminando la autenticación separada por servicio y reduciendo el riesgo de reutilización de credenciales.

Para las identidades de cargas de trabajo (identidades máquina), el CCoE implementará un modelo basado en credenciales temporales de corta duración emitidas por los servicios nativos de cada plataforma (AWS STS, tokens OAuth 2.0 en GCP y Azure). Se eliminará de forma progresiva el uso de credenciales estáticas (access keys de larga duración, service account keys almacenadas) en los procesos automatizados, sustituyéndolas por mecanismos de federación OIDC entre la plataforma CI/CD (GitLab) y los hiperescalares, de forma que los pipelines se autentiquen sin secretos almacenados.

El CCoE deberá implementar autenticación multifactor (MFA) obligatoria para todas las cuentas con acceso a la consola cloud, con especial exigencia en cuentas privilegiadas (root, administradores de organización). Se definirán políticas de acceso personalizadas (custom policies) que sustituyan progresivamente las políticas gestionadas por defecto del proveedor, que típicamente conceden permisos más amplios de lo necesario, por políticas granulares adaptadas a los roles reales de los usuarios y servicios de EJIE, siguiendo el principio de mínimo privilegio.

El adjudicatario deberá realizar revisiones trimestrales de los permisos efectivos de todas las identidades (humanas y máquina), identificando y remediando permisos excesivos, cuentas inactivas y roles sin uso. Para ello se apoyará en las herramientas nativas de análisis de acceso (AWS IAM Access Analyzer, GCP IAM Recommender) y, si se justifica, en capacidades CIEM (Cloud Infrastructure Entitlement Management) para una visión consolidada multicloud.

Entregables esperados:

Se entregarán los procedimientos documentados de federación de identidades y SSO, el plan de eliminación de credenciales estáticas con cronograma y métricas de avance, la matriz de roles y permisos personalizados por entorno y perfil de usuario, y los informes trimestrales de revisión de accesos con las acciones correctivas ejecutadas.

6.3 Capa 3 — Segmentación de red y perímetro

La segmentación de red protege tanto las zonas públicas como privadas de los entornos cloud y controla el tráfico entre cargas de trabajo. El CCoE deberá asegurar que la arquitectura de red cloud de EJIE siga un modelo segmentado que minimice la superficie de ataque.

Responsabilidades del CCoE:

El adjudicatario deberá mantener la arquitectura de red basada en el modelo hub-and-spoke, donde los servicios compartidos de seguridad y conectividad (DNS, firewalls, sistemas de prevención de intrusiones) se centralicen en VPCs/VNets de tipo hub, y las cargas de trabajo se desplieguen en spokes aislados entre sí. Cada spoke estará aislado por defecto, y la conectividad entre spokes y con el hub se establecerá

exclusivamente a través de los mecanismos controlados del hub, de modo que todo el tráfico norte-sur sea inspeccionado.

Dentro de cada spoke, el CCoE establecerá microsegmentación mediante los controles nativos de políticas de red (Security Groups y NACLs en AWS, NSGs en Azure, Firewall Rules y Service Perimeters en GCP) para controlar el tráfico este-oeste entre cargas de trabajo y servicios, limitando el movimiento lateral en caso de compromiso.

El adjudicatario implementará controles de seguridad perimetral utilizando los servicios nativos de protección en el borde (AWS WAF y Shield Advanced, Azure WAF y DDoS Protection, Cloud Armor en GCP según corresponda), configurados para proteger las aplicaciones expuestas a internet frente a ataques DDoS y amenazas a nivel de aplicación (capa 7).

Se priorizará el uso de endpoints privados (AWS PrivateLink, Azure Private Link, Private Google Access) para el acceso a los servicios cloud gestionados, evitando la exposición al internet público siempre que sea técnicamente viable. La conectividad híbrida con la infraestructura on-premise de EJIE (CPD y cloud privado Batera) se realizará a través de conexiones cifradas dedicadas (Direct Connect/VPN en AWS, Interconnect/VPN en GCP), y el CCoE deberá documentar y mantener actualizado el diagrama de flujos de red y las reglas de firewall aplicadas.

Entregables esperados:

Se producirán los documentos de diseño de la arquitectura de red hub-and-spoke por hiperescalar, los diagramas de flujos de red actualizados trimestralmente, el inventario de reglas de firewall y security groups con justificación de cada regla, y el informe de configuración de controles de seguridad perimetral.

6.4 Capa 4 — Seguridad de cargas de trabajo

Las cargas de trabajo desplegadas en cloud (máquinas virtuales, contenedores, funciones serverless) requieren protección a lo largo de todo su ciclo de vida: construcción, despliegue y ejecución. El CCoE incorporará controles de seguridad en cada fase, siguiendo el enfoque shift-left.

Responsabilidades del CCoE:

En la fase de construcción (build), el adjudicatario establecerá un proceso de hardening de imágenes que garantice que todas las cargas de trabajo se desplieguen a partir de imágenes base preaprobadas y endurecidas, con software innecesario eliminado, puertos cerrados y configuraciones alineadas con los estándares de seguridad de EJIE. Las imágenes (tanto de VMs como de contenedores) se almacenarán en registros controlados y se someterán a análisis de vulnerabilidades automatizado antes de su aprobación para despliegue, utilizando los servicios nativos de escaneo (Amazon Inspector, Azure Defender for Containers, GCP Artifact Analysis). El objetivo es implantar un modelo de cargas de trabajo inmutables, donde las imágenes se construyen de forma segura, se validan antes del despliegue y no se modifican en tiempo de ejecución.

En la fase de despliegue (deploy), el CCoE integrará validaciones de seguridad en los pipelines CI/CD de GitLab, incluyendo análisis de configuración de IaC (escaneo de plantillas Terraform antes de aplicar), escaneo de dependencias, y verificación del cumplimiento de las políticas de seguridad definidas. Ningún despliegue a producción se ejecutará sin haber superado las validaciones de seguridad automatizadas del pipeline.

En la fase de ejecución (run), el adjudicatario implementará mecanismos de protección en tiempo de ejecución (runtime security) que detecten comportamientos anómalos como ejecución de procesos no autorizados, intentos de escalada de privilegios o tráfico de red sospechoso desde las cargas de trabajo, utilizando servicios como AWS GuardDuty, Microsoft Defender o GCP Security Command Center según el hiperescalar. Las cargas de trabajo que muestren comportamiento anómalo podrán ser aisladas automáticamente siguiendo procedimientos de contención predefinidos.

El CCoE definirá y mantendrá un proceso de gestión de parches y actualizaciones que asegure la aplicación oportuna de correcciones de seguridad: para VMs mediante herramientas de gestión automatizada (AWS Systems Manager Patch Manager, equivalentes en GCP/Azure), para contenedores mediante la reconstrucción y redespliegue de imágenes con las capas base actualizadas, y para funciones serverless mediante la actualización de dependencias y runtimes.

Entregables esperados:

El adjudicatario entregará el catálogo de imágenes base endurecidas y aprobadas con el procedimiento de su mantenimiento, la documentación de las validaciones de seguridad integradas en los pipelines CI/CD (shift-left), los procedimientos de respuesta ante detección de anomalías en tiempo de ejecución, y el informe mensual de estado de parcheado y vulnerabilidades pendientes.

6.5 Capa 5 — Seguridad de datos

La protección de los datos en los entornos cloud de EJIE requiere un enfoque data-centric que asegure la confidencialidad, integridad y disponibilidad de la información independientemente del servicio donde resida.

Responsabilidades del CCoE:

El adjudicatario deberá garantizar que el cifrado esté habilitado en todos los servicios cloud utilizados por EJIE, tanto en reposo como en tránsito. Para los datos en reposo, se distinguirá entre cifrado gestionado por el proveedor (SSE-PMK) y cifrado con claves gestionadas por el cliente (SSE-CMK), adoptando CMK (Customer-Managed Keys) mediante los servicios de gestión de claves nativos (AWS KMS, GCP Cloud KMS, Azure Key Vault) para aquellos entornos y datos clasificados como sensibles o de nivel alto según el ENS. El CCoE definirá qué entornos y tipos de datos requieren CMK frente a claves gestionadas por el proveedor, documentando los criterios de decisión. Para los datos en tránsito se exigirá TLS en todas las comunicaciones entre servicios.

El CCoE implementará una gestión centralizada de secretos utilizando los servicios nativos (AWS Secrets Manager, GCP Secret Manager, Azure Key Vault) para almacenar y distribuir de forma segura las claves de cifrado de datos, credenciales de bases de datos, tokens de API y cualquier otro secreto requerido por las aplicaciones. Se prohibirá la inclusión de secretos en el código fuente, ficheros de configuración o imágenes de contenedores, y los pipelines CI/CD se configurarán para inyectar secretos en tiempo de ejecución desde el gestor centralizado.

Cuando el volumen y la sensibilidad de los datos lo justifique, el CCoE evaluará la implantación de capacidades de prevención de pérdida de datos (DLP) utilizando los servicios nativos (AWS Macie, GCP Cloud DLP, Microsoft Purview), comenzando por la detección y clasificación automatizada de datos sensibles almacenados en los servicios de almacenamiento cloud (S3, GCS, Azure Storage).

Entregables esperados:

Se entregarán la política de cifrado cloud de EJIE con la clasificación de entornos por nivel de cifrado requerido, los procedimientos de gestión de claves (rotación, revocación, control de acceso a las claves), el informe de cumplimiento de la política de cifrado (porcentaje de recursos cifrados con CMK vs. PMK), y las recomendaciones sobre implantación de DLP con hoja de ruta priorizada.

6.6 Capa 6 — Gestión de postura de seguridad y monitorización continua

La naturaleza dinámica de la infraestructura cloud, donde los recursos se aprovisionan, modifican y eliminan continuamente, genera un riesgo constante de deriva de configuración y desviación respecto a los estándares de seguridad establecidos. El CCoE deberá implementar mecanismos de evaluación continua de la postura de seguridad y monitorización centralizada.

Responsabilidades del CCoE:

El adjudicatario implementará las capacidades de Cloud Security Posture Management (CSPM) utilizando los servicios nativos de cada hiperescalar (AWS Security Hub y Config, Microsoft Defender for Cloud, GCP Security Command Center) para proporcionar visibilidad continua y automatizada del estado de configuración de los recursos cloud, detectar desviaciones respecto a las políticas de seguridad definidas y comparar la postura de los entornos con los marcos regulatorios aplicables (ENS, ISO 27001, CIS Benchmarks). El CSPM deberá estar habilitado en todas las cuentas, suscripciones y proyectos y en todas las regiones relevantes.

Se implementarán capacidades de Cloud Workload Protection Platform (CWPP) para la detección de vulnerabilidades en las imágenes de máquinas virtuales y contenedores a lo largo de su ciclo de vida, complementando el hardening de imágenes (capa 4) con análisis continuos en los registros de imágenes y en tiempo de ejecución.

El CCoE diseñará e implementará un repositorio centralizado de logs de seguridad (security data lake) donde se agregarán los logs de seguridad de todas las cuentas y entornos cloud: logs de auditoría de API (CloudTrail en AWS, Cloud Audit Logs en GCP), logs de flujo de red, logs de acceso y eventos de los servicios de seguridad. Este repositorio se configurará con políticas de escritura única (write-only) e inmutabilidad para asegurar que los logs no puedan ser modificados ni eliminados por usuarios no autorizados, y se integrará con el SIEM corporativo de EJIE para correlación y análisis.

El adjudicatario activará y configurará las capacidades de detección de amenazas (threat detection) nativas de cada hiperescalar (Amazon GuardDuty, Microsoft Defender for Cloud, GCP SCC) para identificar en tiempo real amenazas como cuentas comprometidas, movimiento lateral, exfiltración de datos y actividad de malware. Las alertas de alta severidad se integrarán con los flujos de respuesta a incidentes del SOC/equipo de ciberseguridad de EJIE, definiendo los procedimientos de escalado, tiempos de respuesta y canales de comunicación.

El CCoE definirá y mantendrá un cuadro de mando de seguridad cloud con indicadores clave que incluyan, como mínimo, el número de hallazgos críticos y altos abiertos por antigüedad, el porcentaje de recursos conformes con las políticas de seguridad (compliance rate), el porcentaje de cuentas con MFA habilitado, el porcentaje de pipelines con OIDC implementado (sin credenciales estáticas), el tiempo medio de remediación de hallazgos de seguridad (MTTR), el número de imágenes con vulnerabilidades críticas sin parchear, y el porcentaje de recursos accesibles públicamente sin justificación.

Entregables esperados:

El adjudicatario entregará el diseño del security data lake con la arquitectura de ingesta y retención de logs, los procedimientos de respuesta ante alertas de seguridad integrados con el SOC de EJIE, el cuadro de mando de seguridad cloud con las métricas definidas (actualización mensual), el informe mensual de postura de seguridad con hallazgos, tendencias y acciones correctivas, y un informe semestral de auditoría de cumplimiento ENS/ISO en los entornos cloud.

6.7 Aplicación por velocidades del modelo de adopción

Las exigencias de seguridad se aplicarán de forma gradual según el nivel de madurez del modelo de adopción a tres velocidades:

Velocidad 1 — Marco Mínimo: Se aplicarán los controles fundamentales de forma centralizada por el CCoE: integración en la landing zone segura, federación de identidades con MFA obligatorio, segmentación de red básica (aislamiento del entorno de producción), cifrado por defecto con claves gestionadas por el proveedor, habilitación de logs de auditoría y monitorización básica (integración en CSPM corporativo). El CCoE será responsable de la ejecución de estos controles, con participación mínima del equipo del proyecto.

Velocidad 2 — Marco Guiado: Se añaden controles compartidos entre el CCoE y el equipo de proyecto: validaciones de seguridad integradas en los pipelines CI/CD (shift-left parcial), uso de imágenes base endurecidas del catálogo corporativo, cifrado con CMK para entornos con datos sensibles, monitorización de KPIs de seguridad del servicio (logs, métricas básicas de observabilidad de seguridad), y revisiones de seguridad periódicas con el CCoE. El equipo de proyecto asume la responsabilidad de aplicar las plantillas y estándares proporcionados.

Velocidad 3 — DevOps Golden Paths: Se aplica el marco completo de seguridad: DevSecOps integral con validaciones automatizadas en cada fase del pipeline, modelo de cargas de trabajo inmutables, cifrado con CMK generalizado, gestión centralizada de secretos integrada en el flujo de despliegue, observabilidad de seguridad completa (trazas, métricas avanzadas, correlación de eventos), runtime security activo con capacidad de contención automática, y cumplimiento continuo verificable contra ENS e ISO. El equipo de proyecto es autónomo en la implementación pero el CCoE audita y valida el cumplimiento.

6.8 Tecnologías de referencia por hiperescalar

A continuación se resumen los servicios nativos de seguridad de referencia para cada capa y hiperescalar, que el CCoE deberá conocer, configurar y operar:

- **Gobernanza de seguridad:** AWS Control Tower y Service Control Policies; GCP Cloud Foundation Toolkit, Resource Manager y Organization Policy Service; Azure Landing Zones y Azure Policy.
- **Gestión de identidades y accesos:** AWS IAM Identity Center, IAM Policies, IAM Access Analyzer; GCP Cloud Identity, IAM Bindings, IAM Recommender; Microsoft Entra ID, Azure RBAC/ABAC.
- **Seguridad de red y perímetro:** AWS Security Groups, NACLs, WAF, Shield, PrivateLink, Direct Connect; GCP Firewall Rules, Service Perimeters, Cloud Armor, Private Google Access, Cloud Interconnect; Azure NSG, WAF, DDoS Protection, Private Link, ExpressRoute.
- **Seguridad de cargas de trabajo:** AWS Inspector, GuardDuty, Systems Manager Patch Manager, CodeArtifact; GCP Artifact Registry, Artifact Analysis, Security Command Center, OS Config; Azure Defender for Containers/Servers, Azure Update Management, Azure Artifacts.
- **Seguridad de datos:** AWS KMS, CloudHSM, Secrets Manager, Macie; GCP Cloud KMS, Cloud HSM, Secret Manager, Cloud DLP; Azure Key Vault, Managed HSM, Microsoft Purview.
- **Postura de seguridad y monitorización:** AWS Security Hub, Config, CloudTrail, GuardDuty, S3 (data lake); GCP Security Command Center, Cloud Audit Logs, GCS (data lake); Microsoft Defender for Cloud, Azure Data Lake Storage.

7 Infraestructura como Código (IaC), prácticas GitOps y DevOps

7.1 Enfoque general y principios rectores

El CCoE asumirá la responsabilidad de mantener, evolucionar e impulsar el marco de desarrollo estándar para la Infraestructura como Código (en adelante MD-IaC) de EJIE, actuando como gestor de producto (product manager) de la IaC en los entornos cloud. Este rol implica definir la visión, la estrategia y la hoja de ruta para el uso de IaC, así como gobernar la biblioteca de patrones reutilizables, la adopción de prácticas GitOps y la integración de IaC en los flujos CI/CD corporativos.

La IaC es una tecnología habilitadora cuyo objetivo final no es el código en sí mismo, sino los resultados que permite alcanzar: gobernanza consistente, autoservicio gobernado, productividad de los equipos, reducción del riesgo operativo y aceleración de la adopción cloud. El adjudicatario deberá aplicar los siguientes principios en toda su actuación sobre el MD-IaC:

- **Gobernanza adaptativa sobre control restrictivo.** La IaC se utilizará para comunicar buenas prácticas y prevenir malos resultados (guardrails), no para restringir el uso del cloud de forma innecesariamente rígida. Las políticas de seguridad y cumplimiento se expresarán como Policy as Code (PaC) integrado en los pipelines CI/CD, no como plantillas inflexibles que limiten artificialmente las opciones de los equipos de proyecto. Los controles de acceso nativos de cada hiperescalar (IAM, SCPs, Organization Policies) complementarán la PaC para una gobernanza en profundidad.
- **Patrones reutilizables y componibles sobre plantillas rígidas.** Los artefactos de IaC se diseñarán como módulos componibles, configurables y bien documentados que puedan adaptarse a las necesidades de cada solución, no como plantillas cerradas para uso verbatim. Se evitará el antipatrón de construir cientos de plantillas rígidas que nadie consume; en su lugar, se priorizará la creación progresiva de patrones para las arquitecturas más comunes en EJIE.
- **Enfoque "application-down" sobre "infrastructure-up".** La IaC conceptualizará las aplicaciones como unidades que arrastran consigo su infraestructura y servicios necesarios, en lugar de provisionar una base de infraestructura sobre la que luego se colocan las aplicaciones. Cada solución cloud se desplegará mediante IaC que cubra todo su stack, desde los recursos de infraestructura base hasta la configuración de servicios PaaS y los componentes de observabilidad.
- **IaC como software.** Todo el código de infraestructura se gestionará con la misma disciplina de ingeniería de software que el código de aplicación: control de versiones en GitLab, revisión de código (code review), testing automatizado, documentación y gestión del ciclo de vida. La IaC producida deberá ser usable (documentada y protegida contra errores del usuario), mantenible (legible, adaptable y fácilmente modificable) y testeable (con criterios de corrección bien definidos y verificables).

7.2 Gobierno y curación de la biblioteca de IaC

Rol del CCoE como gestor de producto de la IaC

El adjudicatario, en su función de CCoE, será responsable de la gestión de producto de la IaC cloud de EJIE. Esto implica las siguientes funciones:

- El CCoE definirá la visión y hoja de ruta de la IaC, priorizando qué patrones, módulos y building blocks se desarrollarán o evolucionarán en función de las necesidades de los proyectos activos y previstos. Esta priorización se gestionará como un backlog de producto, revisado en cada sprint y alineado con EJIE.
- El CCoE será responsable de la curación del repositorio centralizado de IaC en GitLab, incluyendo la decisión sobre qué módulos se publican como reutilizables, la revisión y aprobación de contribuciones de la comunidad de práctica cloud de EJIE, y el mantenimiento de la calidad y vigencia de los artefactos publicados.
- El CCoE establecerá las normas y estándares para la creación de IaC en el ámbito cloud de EJIE, incluyendo convenciones de nombrado, estructura de repositorios, reglas de versionado semántico, requisitos de documentación y criterios de aceptación para la publicación de nuevos módulos.

Biblioteca de patrones reutilizables

El adjudicatario construirá y mantendrá de forma progresiva una biblioteca de patrones de IaC reutilizables, organizada en los siguientes niveles de abstracción:

- **Módulos base (building blocks de infraestructura).** Módulos Terraform que encapsulan la provisión y configuración de recursos cloud individuales o grupos de recursos estrechamente acoplados, con las opciones de configuración parametrizadas y los guardrails de seguridad incorporados. Ejemplos: módulo de VPC con subnets y reglas de seguridad, módulo de bucket S3 con cifrado y políticas de acceso, módulo de base de datos RDS con configuración de backup y monitorización. Estos módulos se alinearán con la landing zone y las políticas corporativas.
- **Patrones de arquitectura de referencia.** Composiciones de módulos base que expresan arquitecturas de solución completas para los tipos de aplicación más comunes en EJIE. El CCoE identificará entre tres y cinco patrones arquitectónicos iniciales — por ejemplo, aplicación web N-Tier, arquitectura de microservicios sobre contenedores, solución de datos/analytics, aplicación serverless event-driven — y los expresará como código IaC reutilizable y bien documentado. Estos patrones servirán como punto de partida que los equipos de proyecto adaptan a sus necesidades, no como plantillas inflexibles.
- **Plantillas de catálogo de servicios.** Para soluciones repetitivas con pocas opciones de configuración (por ejemplo, entornos de desarrollo estándar, sandboxes, entornos analíticos preconfigurados), se crearán plantillas de IaC consumibles desde la plataforma de desarrollo interno (IDP basada en Backstage) en modo autoservicio. Estas plantillas se reservarán para casos donde la estandarización total aporte valor, evitando la sobreingeniería de catálogos que los equipos no consumen.
- **Snippets y ejemplos.** Fragmentos de código documentados que muestren cómo resolver problemas específicos (integración con un servicio concreto, patrón de naming, configuración de un provider) y que sirvan como referencia y material de aprendizaje para los equipos de proyecto.

Descubribilidad y consumo

Todos los artefactos de IaC reutilizables se publicarán en la plataforma IDP (Backstage) con documentación asociada que incluya la descripción del patrón y caso de uso, diagrama de arquitectura, instrucciones de uso con ejemplos, variables de entrada y salida documentadas, dependencias y requisitos previos, y la versión del módulo y changelog. El objetivo es que cualquier equipo de proyecto pueda descubrir, evaluar y consumir los building blocks sin necesidad de soporte directo del CCoE.

Mantenimiento y evolución

El adjudicatario será responsable de mantener la biblioteca actualizada frente a la evolución de los servicios cloud de los hiperescalares, las actualizaciones de Terraform y las nuevas necesidades de EJIE. Se establecerá una cadencia de revisión trimestral de los módulos publicados para verificar su vigencia, compatibilidad y alineación con las políticas de seguridad vigentes. Los módulos obsoletos se deprecarán con un periodo de aviso y un plan de migración hacia su sucesor.

7.3 Responsabilidades diferenciadas en la creación y mantenimiento de IaC

La división de responsabilidades sobre la IaC entre el CCoE y los equipos de proyecto variará según el nivel de madurez del modelo de adopción a tres velocidades:

Velocidad 1 — Marco Mínimo

En este nivel, el CCoE asume la responsabilidad principal de la creación, ejecución y mantenimiento de la IaC, que aplicará en el contexto de la plataforma cloud. Los equipos de proyecto tienen participación mínima en el código de infraestructura, que les viene dado.

El CCoE creará la IaC necesaria para cada proyecto utilizando los módulos base y patrones de la biblioteca corporativa, configurándolos según los requisitos de la solución. La ejecución de Terraform (plan/apply) será responsabilidad del CCoE, que controlará los despliegues y gestionará el estado (state) de Terraform de forma centralizada. El equipo de proyecto será consultado en las decisiones de diseño de infraestructura, pero no escribirá ni ejecutará código IaC directamente. La IaC se limitará a la cobertura de la infraestructura fundamental: provisión de entornos base, networking, seguridad básica y servicios compartidos. El CCoE documentará toda la IaC generada para facilitar una futura transición del equipo hacia niveles superiores.

Velocidad 2 — Marco Guiado

En este nivel, la responsabilidad de la IaC es compartida entre el CCoE y el equipo de proyecto, con el CCoE actuando como guía y revisor.

El equipo de proyecto, con el apoyo del CCoE, comenzará a crear y mantener IaC para los componentes específicos de su aplicación (servicios PaaS, middleware, configuración de aplicación), extendiendo la cobertura más allá de la infraestructura base. El CCoE proporcionará los módulos base, las plantillas de referencia y la formación necesaria para que el equipo de proyecto pueda trabajar con autonomía progresiva. Se fomentará la creación colaborativa de IaC mediante pair programming entre ingenieros del CCoE y del equipo de proyecto, de forma que el equipo de proyecto adquiera las competencias necesarias mientras produce código de calidad. Toda la IaC del proyecto se gestionará en repositorios GitLab con enfoque parcialmente GitOps, y se someterá a code review por parte del CCoE antes de su integración. El equipo de proyecto ejecutará despliegues en entornos no productivos, mientras el CCoE aprobará los merge requests y supervisará los despliegues a producción.

Velocidad 3 — DevOps Golden Paths

En este nivel, el equipo de proyecto es autónomo en la creación, mantenimiento y ejecución de IaC, con el CCoE actuando como habilitador experto y auditor.

El equipo de proyecto diseña, desarrolla y mantiene toda la IaC de su solución, siguiendo la filosofía de "todo como código": infraestructura, configuraciones, pipelines, políticas y procedimientos operativos. El equipo consume los building blocks y módulos corporativos del CCoE, pudiendo extenderlos y personalizarlos siempre que respete los estándares de calidad y seguridad. Se aplica GitOps completo: cualquier cambio en infraestructura o configuración se realiza exclusivamente mediante commits en el repositorio, con reconciliación automática entre el estado declarado en código y el estado real de los entornos. El CCoE audita la IaC producida por el equipo de proyecto para verificar el cumplimiento de los estándares corporativos, las políticas de seguridad y las buenas prácticas, pero no interviene en la ejecución operativa. El equipo de proyecto es responsable de mantener su IaC actualizada, de contribuir mejoras al repositorio corporativo cuando identifique patrones generalizables, y de documentar sus decisiones arquitectónicas.

7.4 Herramientas de IaC y selección tecnológica

Herramienta principal: Terraform (HCL)

El MD-IaC de EJIE se basa en Terraform como herramienta principal de IaC, como lenguaje de definición. El adjudicatario deberá mantener y evolucionar toda la base de código Terraform existente y nueva bajo esta elección tecnológica. La decisión de usar una herramienta multicloud como Terraform, en lugar de las herramientas nativas de cada proveedor (CloudFormation/CDK en AWS, Deployment Manager en GCP, ARM/Bicep en Azure), se justifica por la necesidad de mantener un conjunto de competencias unificado en el equipo, dado que EJIE opera en un entorno multicloud.

El adjudicatario tendrá en cuenta que Terraform multicloud no implica portabilidad de código entre proveedores: el código es específico de los recursos y servicios de cada hiperescalar. La ventaja reside en la uniformidad de la herramienta, los flujos de trabajo y las prácticas de ingeniería, no en la portabilidad del código.

Herramientas complementarias

Además de Terraform, el MD-IaC podrá incorporar las siguientes herramientas complementarias dentro del ecosistema tecnológico de EJIE:

GitLab CI/CD como orquestador de pipelines que ejecutará los flujos de plan, validación y apply de Terraform, así como las verificaciones de PaC y los tests automatizados de IaC. Ansible u otras herramientas de automatización de servidores para la configuración dentro de instancias de cómputo (OS, paquetes, middleware) cuando los servicios no sean PaaS gestionados. Open Policy Agent (OPA), Checkov, tfsec u otras herramientas de PaC para la validación automatizada de las configuraciones de Terraform contra las políticas de seguridad y cumplimiento de EJIE antes de cada despliegue. Terratest u otros frameworks de testing de IaC para la validación automatizada de los módulos Terraform.

El CCoE evaluará periódicamente la evolución del ecosistema de herramientas de IaC y propondrá a EJIE la incorporación de nuevas herramientas cuando aporten un valor demostrable, documentando la justificación y el plan de adopción.

7.5 Prácticas de ingeniería de software aplicadas a la IaC

Gestión de código fuente y versionado

Toda la IaC se almacenará en repositorios GitLab bajo control de versiones. El adjudicatario definirá y aplicará una estructura de repositorios coherente que distinga entre los módulos base reutilizables (repositorio centralizado de módulos corporativos), los patrones de arquitectura de referencia, la IaC específica de cada proyecto (repositorios dedicados por proyecto o solución) y la configuración de las landing zones.

Se aplicará versionado semántico (semver) a los módulos reutilizables, de forma que los equipos de proyecto puedan fijar la versión consumida y actualizar de forma controlada. Los cambios en módulos reutilizables seguirán un proceso formal de release que incluya changelog, documentación de breaking changes y periodo de deprecación para versiones anteriores.

Separación plan/apply y flujos de aprobación

Los pipelines de IaC en GitLab respetarán la separación estricta entre las fases plan y apply de Terraform. La fase plan generará un plan de ejecución que será revisable antes de su aplicación. Ningún apply en entornos productivos se ejecutará sin revisión previa del plan y aprobación explícita mediante merge request aprobado por el responsable designado. En entornos no productivos, el nivel de aprobación requerido se adaptará según la clasificación del entorno (abierto, restringido o producción).

Gestión del estado de Terraform

El estado de Terraform (state) se gestionará de forma remota y segura, utilizando backends remotos con bloqueo de estado (state locking) para prevenir ejecuciones concurrentes y corrupción del estado. El acceso al state se restringirá mediante políticas de IAM específicas, dado que el state puede contener información sensible. Se establecerán procedimientos documentados para la recuperación del estado en caso de corrupción, la migración de estado entre backends y la importación de recursos existentes.

Testing y validación automatizada

El adjudicatario implementará un marco de testing para la IaC que incluya los siguientes niveles:

Validación estática: comprobaciones de sintaxis, formato (terraform fmt) y validación de configuración (terraform validate) ejecutadas en cada commit. Análisis de políticas (PaC): verificación automatizada contra las políticas de seguridad y cumplimiento de EJIE utilizando herramientas como OPA, Checkov o tfsec, integradas en el pipeline CI. Estas verificaciones actuarán como gates que impidan el despliegue de configuraciones no conformes. Tests de integración: para los módulos base críticos, se implementarán tests

automatizados que desplieguen el módulo en un entorno efímero, verifiquen que los recursos se crean correctamente con la configuración esperada, y destruyan el entorno al finalizar. Revisión de coste estimado: cuando sea viable, se incorporarán estimaciones de coste basadas en el plan de Terraform (mediante herramientas como Infracost) para alertar sobre incrementos de coste significativos antes de aplicar cambios, complementando las prácticas de FinOps.

Code review

Todo cambio en la IaC de módulos reutilizables requerirá code review por al menos un ingeniero del CCoE distinto del autor. Para la IaC de proyectos específicos, el nivel de revisión requerido se adaptará a la velocidad del proyecto: en Velocidad 1 el CCoE revisa internamente, en Velocidad 2 el CCoE revisa la IaC del equipo de proyecto, y en Velocidad 3 el equipo de proyecto realiza su propia revisión interna y el CCoE audita periódicamente.

7.6 Policy as Code (PaC) y guardrails

El adjudicatario implementará un marco de Policy as Code que exprese las políticas de gobernanza cloud de EJIE como código verificable automáticamente, complementando los guardrails nativos de cada hiperescalar (SCPs, Organization Policies, Azure Policy).

Las políticas se codificarán como reglas evaluables en el pipeline CI/CD, de forma que toda IaC sea validada contra ellas antes de su despliegue. Las políticas cubrirán como mínimo los siguientes ámbitos: restricciones de servicios y regiones permitidos, requisitos de cifrado (tipo de cifrado, gestión de claves), requisitos de etiquetado (tags obligatorios para la trazabilidad y el FinOps), configuraciones de red prohibidas (por ejemplo, prohibición de acceso público no justificado), requisitos de logging y monitorización, y estándares de naming.

Las políticas se gestionarán en un repositorio dedicado en GitLab con el mismo rigor de control de versiones y testing que la IaC, dado que cualquier cambio en una política de PaC equivale efectivamente a un cambio en la configuración de producción.

El adjudicatario distinguirá entre políticas bloqueantes (que impiden el despliegue si no se cumplen) y políticas informativas (que generan advertencias pero no bloquean), y documentará el criterio de clasificación para cada política.

7.7 Autoservicio gobernado y catálogo de servicios cloud

El adjudicatario diseñará un modelo de autoservicio gobernado que permita a los equipos de proyecto y desarrolladores de EJIE consumir recursos cloud de forma autónoma dentro de los límites de gobernanza establecidos.

Catálogo de servicios en “BatArin” (Backstage, IDP)

Los building blocks y patrones de IaC reutilizables se expondrán en la plataforma IDP “BatArin” (basada en Backstage) como elementos de un catálogo de servicios cloud. El catálogo incluirá plantillas de autoservicio para los casos de uso más estandarizados (provisión de entornos de desarrollo, sandboxes, entornos analíticos), así como documentación y guías de consumo para los módulos y patrones más complejos que requieran adaptación por parte del equipo de proyecto.

Entornos según nivel de criticidad

La gobernanza de los entornos cloud se diferenciará según su clasificación:

Entornos abiertos (desarrollo, experimentación): los equipos de proyecto podrán provisionar recursos con máxima flexibilidad dentro de los guardrails de coste y seguridad básica, pudiendo interactuar directamente con la consola y CLI del proveedor. La IaC será recomendable pero no obligatoria en estos entornos. Entornos restringidos (preproducción, UAT, entornos con datos sensibles): se requerirá el uso de IaC y el paso por el pipeline de validación con PaC. La provisión manual estará restringida. Entornos de producción: todo despliegue y cambio de configuración se realizará exclusivamente mediante IaC gestionada en GitLab, con pipeline completo de validación, aprobación de merge request y trazabilidad completa. Se prohíbe la provisión o modificación manual de recursos en producción.

7.8 Gestión de recursos compartidos

Cuando múltiples aplicaciones compartan recursos cloud (por ejemplo, clústeres Kubernetes, balanceadores de carga, bases de datos compartidas), el CCoE diseñará servicios internos que encapsulen la lógica de provisión compartida y expongan APIs consumibles desde la IaC de cada proyecto. De esta forma, la IaC de un proyecto invocará el servicio interno de provisión en lugar de gestionar directamente el recurso compartido, evitando conflictos de estado y duplicaciones. Estos servicios de provisión compartida podrán publicarse en Backstage para su descubrimiento y consumo por los equipos de proyecto.

7.9 Comunidad de práctica y mejora continua

El adjudicatario fomentará la creación y animación de una comunidad de práctica cloud (Cloud Community of Practice) dentro de EJIE, orientada a la mejora continua de la IaC y las prácticas DevOps. Las acciones concretas incluirán la solicitud activa de contribuciones de los equipos de proyecto al repositorio de módulos corporativos, con un proceso de curación y revisión por parte del CCoE; la identificación de patrones emergentes en los proyectos de EJIE que merezcan generalizarse como módulos reutilizables; el aprovechamiento de las comunidades open-source de Terraform (Terraform Registry, módulos verificados) para enriquecer la biblioteca corporativa cuando existan módulos maduros y mantenidos; y la realización de sesiones periódicas de compartición de conocimiento (divulgaciones técnicas, demos de nuevos módulos, retrospectivas de IaC).

7.10 Entregables del adjudicatario

El adjudicatario producirá y mantendrá actualizados los siguientes entregables en relación con la línea de trabajo de IaC y prácticas GitOps/DevOps:

- La biblioteca de módulos Terraform reutilizables en GitLab, versionados, documentados y publicados en Backstage.
- Los patrones de arquitectura de referencia expresados como IaC con documentación de diseño asociada.
- El marco de PaC con las políticas codificadas, clasificadas y documentadas. Los pipelines CI/CD de referencia en GitLab para la ejecución de IaC (plan/validate/apply) con las integraciones de PaC y testing.
- La documentación de estándares de IaC: convenciones de nombrado, estructura de repositorios, guía de desarrollo de módulos y proceso de contribución.
- El informe trimestral de estado del MD-IaC que incluya métricas de adopción (porcentaje de proyectos que usan IaC, porcentaje de cobertura de IaC por proyecto), métricas de calidad (tasa de fallos en validación de PaC, tiempo medio de resolución de fallos de IaC), métricas de reutilización (número de módulos en la biblioteca, frecuencia de consumo de cada módulo, contribuciones de la comunidad de práctica) y hoja de ruta de evolución con las prioridades del próximo trimestre.
- La propuesta semestral de evolución tecnológica del MD-IaC, evaluando nuevas versiones de Terraform, nuevas herramientas del ecosistema y oportunidades de mejora.

8 FinOps – Gestión Financiera del Cloud

El CCoE asumirá la función de habilitación financiera del cloud dentro de la práctica FinOps de EJIE, actuando como equipo central que asiste, forma y acompaña a los equipos de proyecto (consumidores cloud) en la planificación, seguimiento y optimización de los costes de sus cargas de trabajo en la nube. El objetivo va más allá de que el CCoE controle centralmente los costes, sino que capacite a los consumidores cloud para que asuman la responsabilidad de sus propias decisiones de gasto, fomentando una cultura de concienciación y rendición de cuentas financiera (cost accountability) en toda la organización.

La práctica FinOps del CCoE se articulará en torno a un ciclo continuo de cuatro dominios: planificar (estimar costes antes de desplegar), seguir (monitorizar el gasto y compararlo con las expectativas), optimizar (reducir el gasto sin perder valor) y empoderar (que los consumidores cloud gestionen sus propios costes). Este ciclo se aplicará de forma iterativa, de modo que los aprendizajes de cada iteración refinen las siguientes.

El adjudicatario se apoyará en la herramienta de gestión financiera cloud ya implantada en EJIE (Cloudability) como plataforma principal de visibilidad, análisis y reporting de costes, integrándola plenamente en su marco de trabajo FinOps.

8.1 Dominio 1 — Aprender: estándares de coste, retroalimentación y biblioteca de patrones

Estándares de coste y buenas prácticas

El adjudicatario desarrollará y mantendrá un conjunto de estándares de coste cloud corporativos que establezcan las reglas y principios que deben aplicarse a toda carga de trabajo desplegada en la nube de EJIE. Estos estándares reflejarán la estrategia de EJIE para mitigar el riesgo de sobregasto y operar entornos cloud eficientes, y se revisarán periódicamente a medida que la organización madure en su gestión financiera cloud.

Los estándares de coste cubrirán, como mínimo, los siguientes ámbitos: la obligación de disponer de un presupuesto precalculado y aprobado antes de desplegar cualquier carga de trabajo; el cumplimiento de la estrategia de etiquetado (tagging) establecida, como requisito previo para la atribución de costes; los plazos máximos para revisar y actuar sobre recomendaciones de optimización de costes identificadas por las herramientas; la obligación de implementar autoescalado horizontal cuando sea técnicamente viable, permitiendo el sobredimensionamiento estático solo con justificación de negocio documentada; la selección de tiers de almacenamiento basada en necesidades reales de rendimiento y frecuencia de acceso, priorizando siempre el tier más frío que satisfaga los requisitos; y las directrices sobre modelos de precios (PAYG como base, compromisos solo tras la confirmación de patrones de uso reales).

El CCoE incorporará estos estándares en la formación obligatoria que deberán completar los equipos de proyecto antes de obtener acceso a entornos cloud de EJIE.

Retroalimentación de gobernanza

El adjudicatario establecerá un bucle formal de retroalimentación entre los consumidores cloud (que aplican los estándares) y el equipo de gobernanza (que los mantiene). Este mecanismo servirá para que los equipos de proyecto comuniquen obstáculos creados por estándares excesivamente restrictivos, propongan mejoras basadas en nuevas técnicas o servicios cloud, y señalen estándares ineficaces o contradictorios. El CCoE recogerá también retroalimentación de los equipos de seguridad, operaciones y finanzas para mantener el equilibrio entre habilitación y control. Las conclusiones de este bucle se incorporarán en las revisiones semestrales de los estándares de coste.

Biblioteca de arquitecturas de referencia con datos de coste

El adjudicatario enriquecerá la biblioteca de patrones de IaC reutilizables (descrita en el apartado de IaC) con información de coste específica para cada patrón, creando una biblioteca de arquitecturas de referencia con valor financiero. Para cada patrón documentado se registrarán dos tipos de información:

Patrones de diseño con datos de coste predespliegue: la estimación de coste calculada antes del despliegue utilizando las calculadoras de precios del hiperescalar o las herramientas de estimación de IaC, adjunta al patrón junto con los supuestos utilizados.

Patrones de despliegue con datos de coste postdespliegue: los costes reales acumulados tras el despliegue de instancias concretas del patrón, medidos en los mismos intervalos que las estimaciones previas (por ejemplo, a los tres y seis meses), para permitir la comparación directa entre estimación y realidad.

La comparación sistemática entre estimaciones y resultados reales permitirá identificar componentes con alta variabilidad de coste, calcular factores de corrección para futuras estimaciones, y priorizar los patrones con menor desviación para proyectos que requieran alta predictibilidad de costes. La biblioteca se mantendrá actualizada y se revisará periódicamente para retirar patrones obsoletos y documentar la evolución de las prácticas.

8.2 Dominio 2 — Diseño consciente de costes

Proceso de asistencia al diseño

El adjudicatario implementará un proceso formal de asistencia al diseño consciente de costes, mediante el cual el CCoE acompañará a los equipos de proyecto en las fases iniciales de diseño de cada nueva solución o migración cloud, con el objetivo de que las implicaciones financieras de las decisiones arquitectónicas se evalúen antes de que comience la facturación.

Este proceso no consiste en que el CCoE diseñe las soluciones ni apruebe las arquitecturas como un comité de revisión bloqueante. Su función es la de asesor proactivo que desafía las decisiones con mayor impacto financiero, propone alternativas más eficientes y transfiere conocimiento a los equipos de proyecto. La responsabilidad última del diseño permanece en el equipo de proyecto, lo que refuerza su sentido de accountability financiera.

La asistencia al diseño incluirá las siguientes actividades:

- **Análisis de requisitos con impacto en costes.** El CCoE ayudará al equipo de proyecto a cuestionar los requisitos que más impactan al coste: disponibilidad (¿es realmente necesaria una arquitectura multiregión o basta con multi-AZ?), resiliencia (¿qué nivel de RPO/RTO se necesita y cuál es su coste?), rendimiento (¿están sobredimensionados los recursos?), escalabilidad (¿se puede usar autoescalado en lugar de provisionar para el pico?), monitorización y logging (¿qué volumen de telemetría es necesario y cuál es su coste de almacenamiento?), seguridad y

cumplimiento (¿qué controles son regulatoriamente obligatorios frente a los que son opciones de sobreingeniería?), transferencia de datos (¿se han considerado los costes de egress y de peering entre VPCs/regiones?) y residencia de datos (¿la región seleccionada es la más eficiente en coste para el caso de uso?).

- **Análisis específico para migraciones.** Cuando la solución sea una migración desde on-premise, el CCoE aplicará un análisis específico que aproveche los datos históricos del entorno origen (uso real de CPU, memoria, almacenamiento, red) para dimensionar el entorno destino de forma adecuada, evitando la tendencia habitual a sobredimensionar que se arrastra de los entornos on-premise donde los recursos son limitados. Se contrastarán las recomendaciones de rightisizing de las herramientas de migración con las necesidades reales del equipo de proyecto, y se considerará el impacto de la estrategia de migración (rehost, revise, rearchitect) en la precisión de las estimaciones de coste.
- **Recomendaciones de optimización en fase de diseño.** El CCoE propondrá activamente optimizaciones durante el diseño, tales como: simplificar la arquitectura de alta disponibilidad cuando la criticidad del servicio lo permita, implementar autoescalado horizontal para cargas variables en lugar de provisionar para el pico, sustituir servicios serverless por servicios con capacidad provisionada (o viceversa) según el patrón de consumo previsto, reducir el número de VPCs para minimizar costes de peering, y seleccionar tiers de almacenamiento y tipos de instancia apropiados al perfil de carga.

Arquitectura suficiente

El CCoE guiará a los equipos de proyecto hacia una "arquitectura suficiente" que equilibre la optimización de costes en fase de diseño con la velocidad de entrega. No se perseguirá un diseño perfecto en coste antes del despliegue, ya que muchos supuestos solo pueden validarse con datos reales de producción. En su lugar, se dedicará un tiempo razonable al diseño consciente de costes (el CCoE definirá directrices sobre el esfuerzo mínimo esperado, por ejemplo, un número mínimo de iteraciones de revisión de costes) y se planificará la optimización post-despliegue como parte natural del ciclo.

8.3 Dominio 3 — Estimar: modelo de costes, presupuestos y controles en entregas

Estimación de costes y modelo de coste por carga de trabajo

El adjudicatario implementará un proceso de estimación de costes para cada nueva carga de trabajo o proyecto cloud de EJIE, que se completará antes del despliegue inicial. La estimación partirá de dos elementos: la lista de materiales cloud (cloud BOM), que detalla todos los servicios, recursos y configuraciones necesarios, y el factor de crecimiento, que refleja la previsión de evolución del consumo en un horizonte de tres a seis meses.

Para la estimación se utilizarán las siguientes herramientas, seleccionando la más apropiada para cada caso:

- **Calculadoras de precios del proveedor (herramienta principal).** Las calculadoras nativas de AWS (AWS Pricing Calculator), GCP (Google Cloud Pricing Calculator) y Azure (Azure Pricing Calculator) según el hiperescalar de destino. Estas herramientas se utilizarán como mecanismo principal de estimación, dada su cobertura completa del catálogo de servicios y su actualización continua con las listas de precios vigentes.

- **Cloudability Workload Planning.** La funcionalidad de planificación de cargas de trabajo de Cloudability (ya implantada en EJIE) se utilizará para estimaciones comparativas multicloud y para enriquecer las estimaciones con datos históricos de consumo de patrones similares ya desplegados.
- **Herramientas de estimación de costes basadas en IaC.** Para las cargas de trabajo con arquitectura mantenida como IaC (Terraform), el CCoE integrará herramientas como Infracost o la funcionalidad equivalente de HCP Terraform en los pipelines de GitLab, de forma que la estimación de coste se calcule automáticamente a partir de la plantilla de Terraform. Estas herramientas eliminan la necesidad de modelar la carga de trabajo por separado para la estimación financiera, aumentan la concienciación de costes entre los ingenieros que mantienen el código IaC, y habilitan los controles de coste en el pipeline CI/CD (cost-related release gates).
- **Hojas de cálculo personalizadas.** Cuando sea necesario incorporar elementos no cloud al modelo de coste (licencias de software, costes laborales, activos on-premise), se complementará la estimación con hojas de cálculo que integren las exportaciones de las calculadoras del proveedor.

Selección del modelo de precios

El adjudicatario aplicará como regla general la estimación en precios PAYG (pago por uso) para todas las estimaciones iniciales, evitando la tentación de utilizar modelos de compromiso (reserved instances, savings plans) en las estimaciones pre-despliegue, ya que los descuentos de compromiso distorsionan la estimación cuando no se dispone de datos reales de uso que los justifiquen. La migración hacia modelos de compromiso se producirá solo después de que los patrones de uso se hayan confirmado con datos de producción reales, y siguiendo un proceso de análisis que el CCoE documentará (análisis de coverage, break-even y escenarios de flexibilidad). Para cargas de trabajo tolerantes a interrupciones (batch processing, trabajos no críticos), el CCoE evaluará el uso de instancias preemptible/spot cuando el perfil de carga lo permita.

Aprobación de presupuestos

El adjudicatario implementará un proceso formal de aprobación de presupuesto cloud por carga de trabajo, como paso previo obligatorio a la provisión de recursos en entornos productivos. El proceso funcionará de la siguiente manera:

- El equipo de proyecto (consumidor cloud), asistido por el CCoE si es necesario, calcula la estimación de coste de su carga de trabajo. El equipo de proyecto solicita la aprobación del presupuesto al responsable presupuestario del proyecto en EJIE, presentando la estimación junto con su justificación de negocio. Una vez aprobado, el presupuesto se configura como alerta en Cloudability para la monitorización continua de desviaciones. El CCoE no impone presupuestos a los equipos de proyecto; los equipos de proyecto proponen los presupuestos que consideran necesarios para alcanzar sus objetivos de negocio, reforzando su sentido de responsabilidad financiera.
- La aprobación de presupuesto será un proceso único por carga de trabajo al inicio del despliegue, complementado por un proceso ágil de revisión presupuestaria que se active cuando las alertas de Cloudability señalen desviaciones significativas respecto al presupuesto aprobado.

Controles de coste en releases (cost-related release gates)

Para las aplicaciones entregadas de forma continua mediante CI/CD, el adjudicatario integrará controles de coste automatizados en los pipelines de GitLab, de forma que la gestión financiera no se limite al despliegue inicial, sino que acompañe cada release posterior.

El mecanismo funcionará de la siguiente manera: las herramientas de estimación basadas en IaC (Infracost u equivalente) calcularán automáticamente el coste estimado de cada nueva release a partir de la plantilla Terraform incluida en el merge request. La estimación se comparará con la de la release anterior (no con un valor fijo absoluto, que quedaría obsoleto rápidamente), y si la diferencia supera un umbral configurable (por ejemplo, un incremento superior al 10%), se activará un gate que requerirá la aprobación manual del responsable de la aplicación o del responsable presupuestario antes de proceder al despliegue. Los resultados de la estimación se publicarán como comentario en el merge request de GitLab, proporcionando visibilidad inmediata al equipo de proyecto sobre el impacto financiero de sus cambios de infraestructura.

Este mecanismo no bloquea la entrega de forma rutinaria; su función es detectar cambios con impacto financiero significativo e inesperado y asegurar que se autorizan conscientemente.

8.4 Dominio 4 — Seguir y optimizar: monitorización continua y optimización de costes

Visibilidad y atribución de costes

El adjudicatario configurará y mantendrá en Cloudability los dashboards y reportes necesarios para proporcionar visibilidad completa sobre el gasto cloud de EJIE, con los siguientes niveles de detalle:

Visión consolidada del gasto total cloud de EJIE, con desglose por hiperescalar (AWS, GCP, Azure según corresponda). Visión por unidad organizativa, proyecto o aplicación, basada en la estrategia de etiquetado (tagging). Visión por línea de servicio cloud (cómputo, almacenamiento, red, bases de datos, etc.). Análisis de tendencias temporales (mensual, trimestral, interanual). Identificación de anomalías de coste (consumos inusuales, picos inesperados).

Estrategia de etiquetado (tagging)

El adjudicatario definirá y mantendrá una estrategia de etiquetado de recursos cloud que sea la base de la atribución de costes. La estrategia definirá como mínimo las etiquetas obligatorias (proyecto, responsable, entorno, centro de coste, fecha de creación), las convenciones de nomenclatura y valores permitidos, el mecanismo de enforcement (políticas que impidan la creación de recursos sin las etiquetas obligatorias), y las métricas de cumplimiento (porcentaje de recursos correctamente etiquetados, monitorizado mensualmente). El CCoE realizará auditorías mensuales de cumplimiento de la estrategia de etiquetado y reportará el porcentaje de recursos no etiquetados o incorrectamente etiquetados, trabajando con los equipos de proyecto para alcanzar un objetivo de cumplimiento superior al 95%.

Optimización continua de costes

El adjudicatario implementará un proceso continuo de optimización de costes que identifique y materialice oportunidades de ahorro, incluyendo las siguientes prácticas:

- **Rightsizing.** Análisis periódico (mensual) de los recursos provisionados frente a su utilización real, identificando instancias sobredimensionadas que puedan reducirse sin impacto en el rendimiento. Se utilizarán las recomendaciones nativas de los hiperescalares (AWS Compute Optimizer, GCP Recommender) complementadas con los análisis de Cloudability.

- **Recursos infrautilizados o huérfanos.** Detección de recursos sin uso (volúmenes no adjuntos, IPs elásticas sin asignar, snapshots antiguos, load balancers sin tráfico, entornos de desarrollo/test activos fuera de horario) y propuesta de eliminación o programación de apagado.
- **Autoescalado y scheduling.** Revisión de las oportunidades de implementar autoescalado en cargas con variabilidad de demanda y scheduling (encendido/apagado programado) en entornos no productivos.
- **Selección de tiers de almacenamiento.** Revisión periódica de los tiers de almacenamiento utilizados frente a los patrones de acceso reales, con recomendaciones de migración a tiers más fríos cuando proceda (S3 Intelligent-Tiering, GCS Nearline/Coldline).
- **Gestión de compromisos.** Una vez confirmados los patrones de uso estable con al menos tres meses de datos de producción, el CCoE evaluará la conveniencia de contratar compromisos (Reserved Instances, Savings Plans en AWS, Committed Use Discounts en GCP) mediante un análisis de coverage, break-even y escenarios de flexibilidad. La decisión de contratar compromisos será siempre de EJIE, con la recomendación técnica del CCoE.
- **Optimización de transferencia de datos.** Análisis de los costes de transferencia de datos (egress, peering inter-región, tráfico entre AZs) e identificación de oportunidades de reducción mediante cambios arquitectónicos (consolidación de VPCs, uso de endpoints privados, caching).

8.5 Dominio 5 — Habilitar: showback, chargeback y cultura de coste

Modelo de showback/chargeback

El adjudicatario diseñará, en colaboración con EJIE, un modelo de reparto y visibilización de costes cloud adaptado a la estructura organizativa de EJIE. El modelo podrá ser de showback (visibilización informativa del coste por unidad/proyecto sin facturación interna) o de chargeback (imputación real de costes a cada unidad/proyecto), según la decisión de EJIE. En ambos casos, el CCoE configurará en Cloudability los reportes necesarios para que cada responsable de proyecto pueda ver el coste de sus recursos cloud desglosado y comparado con su presupuesto aprobado.

Formación y concienciación financiera

El adjudicatario diseñará e impartirá un programa de formación en economía cloud dirigido a los equipos de proyecto de EJIE, como requisito previo a la obtención de acceso a entornos cloud. El programa cubrirá los fundamentos de facturación cloud (cómo se acumulan los cargos en los diferentes servicios), los modelos de precios disponibles y sus *trade-offs* (PAYG, compromisos, spot), las herramientas de estimación y monitorización de costes disponibles (calculadoras, Cloudability, Infracost), los estándares de coste corporativos de EJIE y su aplicación, y las técnicas básicas de optimización que cada equipo de proyecto puede aplicar. Se ofrecerán dos modalidades: una formación inicial para equipos que se incorporen al cloud por primera vez, y sesiones de actualización periódicas para consumidores experimentados cuando se actualicen los estándares de coste o se incorporen nuevos servicios o herramientas.

Comunidad de práctica y compartición de aprendizajes

El CCoE canalizará la compartición de aprendizajes de FinOps a través de la comunidad de práctica cloud de EJIE, incluyendo la publicación de casos de éxito de optimización (cuánto se ahorró, cómo se hizo, qué se aprendió), la difusión de nuevas técnicas o servicios con impacto en costes, y la discusión de incidentes de coste (picos inesperados, anomalías) y las lecciones extraídas.

8.6 Aplicación por niveles del modelo de adopción

Las prácticas FinOps se aplicarán de forma gradual según el nivel de madurez del modelo de tres niveles:

Nivel 1 — Marco Mínimo. El CCoE asume la responsabilidad principal de la planificación y estimación de costes. Se aplican los estándares de coste básicos de forma centralizada. El CCoE calcula la estimación de coste del proyecto y la presenta al equipo de proyecto para su validación y solicitud de aprobación presupuestaria. Se configura la monitorización básica de presupuesto en Cloudability (alertas de umbral). La optimización de costes la realiza el CCoE, con participación mínima del equipo de proyecto.

Nivel 2 — Marco Guiado. La responsabilidad de la planificación de costes es compartida. El equipo de proyecto, con asistencia del CCoE, calcula sus propias estimaciones de coste y solicita la aprobación presupuestaria. El CCoE proporciona asistencia al diseño consciente de costes (design assistance) durante la fase de arquitectura. Se integran controles de coste básicos en los pipelines CI/CD (estimaciones Infracost informativas, no bloqueantes). El equipo de proyecto recibe reportes de showback de sus recursos y es responsable de revisar las recomendaciones de optimización dentro de los plazos establecidos.

Nivel 3 — DevOps Golden Paths. El equipo de proyecto es autónomo en la gestión financiera de sus cargas de trabajo. El equipo calcula, presenta y gestiona su presupuesto cloud de forma independiente. Los controles de coste en CI/CD están plenamente activos (release gates bloqueantes para incrementos significativos). El equipo de proyecto gestiona directamente sus dashboards de coste en Cloudability y ejecuta sus propias optimizaciones. El CCoE actúa como auditor y asesor experto, interviniendo solo cuando se detectan desviaciones significativas o cuando el equipo solicita ayuda en optimizaciones complejas (compromisos, renegociación de tiers, cambios arquitectónicos de alto impacto).

8.7 Reporting y Entregables

El adjudicatario producirá y mantendrá actualizados los siguientes entregables en relación con la línea de trabajo de FinOps:

- El documento de estándares de coste cloud corporativos, actualizado semestralmente.
- Los dashboards configurados en Cloudability con las vistas descritas (consolidada, por proyecto, por servicio, tendencias, anomalías).
- El informe mensual de costes cloud que incluirá el gasto total y su evolución respecto al mes anterior y al mismo mes del año anterior, el desglose por proyecto/aplicación y comparación con presupuestos aprobados, el porcentaje de cumplimiento de la estrategia de etiquetado, las recomendaciones de optimización identificadas con ahorro estimado por cada una, las acciones de optimización ejecutadas en el periodo con ahorro materializado, y las alertas de anomalía de coste producidas y su resolución.
- El informe trimestral de FinOps ampliado que incluirá además el análisis de tendencias y proyección a tres meses, la evaluación de oportunidades de compromiso (reserved instances, savings plans), la comparación entre estimaciones pre-despliegue y costes reales por proyecto (retroalimentación de la biblioteca de patrones), el análisis de costes de transferencia de datos y recomendaciones arquitectónicas, y la propuesta de actualización de los estándares de coste si procede.
- La propuesta anual de estrategia FinOps con objetivos de ahorro, iniciativas de optimización prioritarias y hoja de ruta de madurez para el siguiente periodo.

9 Site Reliability Engineering (SRE), observabilidad y evolución hacia AIOps

El CCoE integrará prácticas de Site Reliability Engineering (SRE) y observabilidad como pilares fundamentales de la operación cloud de EJIE, con el doble objetivo de garantizar la fiabilidad y el rendimiento de los servicios desplegados en la nube y de construir, de forma progresiva, la base de datos de telemetría, los procesos y las competencias necesarias para implantar capacidades de AIOps (Artificial Intelligence for IT Operations) que permitan evolucionar hacia operaciones cada vez más inteligentes y automatizadas.

Tal como se adelantaba en el apartado anterior de líneas de trabajo, la observabilidad no es un fin en sí misma: es la infraestructura de datos sobre la que se sustenta tanto la práctica SRE (para tomar decisiones basadas en datos sobre fiabilidad) como la evolución futura hacia AIOps (para que los modelos de IA puedan analizar, correlacionar y actuar sobre esa telemetría). Por tanto, el diseño de la plataforma de observabilidad que realice el CCoE deberá contemplar desde el inicio los requisitos de calidad, cobertura y normalización de datos que una plataforma AIOps necesitará para funcionar eficazmente.

9.1 Estrategia de observabilidad

El adjudicatario diseñará e implantará una estrategia de observabilidad basada en los tres pilares fundamentales — métricas, logs y trazas distribuidas — complementados con perfilado continuo cuando la criticidad del servicio lo justifique. La combinación de estos pilares proporcionará a EJIE la capacidad de comprender el comportamiento interno de sus sistemas cloud a partir de los datos que producen, no solo de responder a alertas predefinidas.

Métricas: Se implementará la recolección de métricas a tres niveles:

- Métricas de infraestructura: utilización de CPU, memoria, disco, red, estado de instancias, rendimiento de servicios gestionados (bases de datos, colas, caches), recogidas de los servicios nativos de monitorización de cada hiperescalar (CloudWatch en AWS, Cloud Monitoring en GCP, Azure Monitor) e integradas en la plataforma de monitorización corporativa de EJIE.
- Métricas de aplicación: tiempos de respuesta, tasas de error, throughput, latencia por endpoint, colas de procesamiento, estado de conexiones. Estas métricas se instrumentarán desde las propias aplicaciones utilizando estándares abiertos.
- Métricas de negocio: indicadores directamente vinculados al valor que el servicio aporta (transacciones procesadas, usuarios activos, pedidos completados, expedientes tramitados). Estas métricas son fundamentales para la práctica SRE porque conectan la fiabilidad técnica con el impacto en el negocio, y serán esenciales para que los futuros modelos de AIOps prioricen los incidentes por impacto real.

El CCoE definirá estándares de cardinalidad y convenciones de etiquetado para las métricas, evitando el uso de valores de alta cardinalidad (IDs de usuario, IDs de request, timestamps) como etiquetas de métricas, que incrementan exponencialmente los costes de almacenamiento y procesamiento y degradan el rendimiento de las consultas. El control de la cardinalidad es un requisito previo para que los modelos de AIOps puedan consumir las métricas de forma eficiente.

Logs: implementará una arquitectura de logging centralizado que agregue los logs de todas las fuentes relevantes: logs de auditoría de la API cloud (CloudTrail, Cloud Audit Logs), logs de aplicación, logs de servicios gestionados, logs de red y logs de seguridad.

El adjudicatario definirá y aplicará estándares de formato de logging estructurado (JSON preferentemente) con campos obligatorios normalizados (como por ejemplo timestamp, severity, service, trace_id, span_id, message, contexto de negocio, etc.). La normalización del formato de logs es un requisito crítico para la futura implantación de AIOps: los modelos de IA no pueden correlacionar eventos de forma fiable si los logs de diferentes servicios utilizan formatos, campos y convenciones de severidad inconsistentes.

Se establecerán políticas de retención diferenciadas por tipo de log y criticidad, equilibrando la necesidad de retención para auditoría y análisis histórico (alimentar modelos de AIOps con datos de largo plazo) con los costes de almacenamiento. El CCoE definirá qué logs se habilitan por defecto (no todos los servicios cloud los activan automáticamente), priorizando los que aportan mayor valor diagnóstico y de seguridad.

Se aplicarán técnicas de optimización como el muestreo inteligente (tail-based sampling) para logs de alto volumen: conservando el 100% de los registros asociados a errores y un porcentaje configurable de los registros de operaciones exitosas, reduciendo costes sin perder la visibilidad sobre los fallos.

Trazas Distribuidas: Se entiende por este tipo de trazas registrar el camino completo de una petición a través de los diferentes servicios y componentes de una arquitectura distribuida (microservicios, funciones serverless, servicios gestionados). El CCoE implementará tracing distribuido para todas las soluciones desplegadas en niveles 2 y 3, y de forma selectiva en nivel 1 cuando la complejidad de la solución lo justifique.

Las trazas proporcionarán la visibilidad necesaria para diagnosticar problemas de latencia, identificar cuellos de botella entre servicios, comprender las dependencias reales entre componentes (no solo las documentadas), y alimentar los modelos de AIOps con datos de topología de servicio para el análisis de causa raíz.

Instrumentación con OpenTelemetry: El CCoE adoptará OpenTelemetry (OTel) como estándar de instrumentación para la generación y recolección de telemetría (métricas, logs y trazas). OpenTelemetry es el estándar abierto y vendor-neutral que define APIs, SDKs y formatos de datos unificados para los tres pilares de la observabilidad, y su adopción garantiza la independencia de EJIE respecto a cualquier backend de observabilidad concreto.

El adjudicatario desplegará y operará los OpenTelemetry Collectors necesarios para recibir, procesar, muestrear y exportar la telemetría hacia los backends de observabilidad de EJIE (la plataforma de monitorización corporativa y los servicios nativos de cada hiperescalar). Los Collectors actuarán como capa de abstracción que desacopla la instrumentación de las aplicaciones del backend de almacenamiento, facilitando la evolución futura de la plataforma de observabilidad sin requerir cambios en las aplicaciones instrumentadas.

El CCoE proporcionará a los equipos de proyecto guías de instrumentación OTel adaptadas a los lenguajes y frameworks más comunes en EJIE, así como módulos de IaC para el despliegue automatizado de los Collectors.

9.2 Práctica SRE: SLOs, error budgets y mejora continua

Service Level Indicators (SLIs) y Service Level Objectives (SLOs): El adjudicatario implantará un modelo formal de SLIs y SLOs para los servicios cloud gestionados por el CCoE y, progresivamente, para las soluciones de los equipos de proyecto.

Los SLIs se definirán como medidas cuantitativas enfocadas en la experiencia del usuario final, no en métricas internas de infraestructura. Los SLIs típicos incluirán la disponibilidad (proporción de peticiones exitosas sobre el total), la latencia (proporción de peticiones servidas por debajo del umbral de tiempo objetivo, por ejemplo el percentil 99), la corrección (proporción de respuestas que devuelven el resultado correcto) y la frescura (para servicios de datos, el tiempo transcurrido desde la última actualización). Los SLIs se calcularán a partir de las métricas recolectadas por la plataforma de observabilidad, cerrando el vínculo entre observabilidad y fiabilidad.

Los SLOs se establecerán como objetivos internos de fiabilidad, ligeramente por debajo del rendimiento típico observado, de forma que sean alcanzables pero exigentes. Por ejemplo, si un servicio mide típicamente una disponibilidad del 99,97%, un SLO del 99,9% proporciona margen operativo sin ser trivial. Los SLOs son decisiones estratégicas, no técnicas: el CCoE asesorará a EJIE y a los equipos de proyecto en la definición de SLOs que reflejen el equilibrio adecuado entre fiabilidad, velocidad de innovación y coste.

Error budgets y políticas de error Budget: Para cada servicio con SLO definido, el CCoE calculará y monitorizará el error budget: la cantidad de incumplimiento que el servicio puede "gastar" antes de violar su SLO. El error budget transforma la fiabilidad de un concepto abstracto en un recurso cuantificable que tanto el equipo de desarrollo como el de operaciones pueden gestionar conjuntamente.

El adjudicatario definirá una política de error budget que establezca las acciones a tomar según el consumo del presupuesto de error. Cuando el servicio opera dentro de su SLO (error budget positivo), las releases continúan según la cadencia normal y el equipo puede priorizar nuevas funcionalidades. Cuando el error budget se aproxima al agotamiento, se activan medidas preventivas: se incrementan los requisitos de testing, se reduce la frecuencia de releases y se priorizan las mejoras de fiabilidad. Cuando el error budget se agota, se congelan las releases (salvo correcciones de incidencias críticas y parches de seguridad) hasta que el servicio recupere su SLO. Se implementarán alertas de burn rate que detecten consumos acelerados del error budget (por ejemplo, si se consume el 10% del presupuesto mensual en menos de 24 horas), como señal temprana de degradación.

Gestión de incidentes y postmortems: El CCoE alineado con el proceso de gestión de incidencias de EJIE determinará las peculiaridades que se deban introducir para contemplar los servicios cloud. En todo caso se deberá incluir la clasificación por severidad, los procedimientos de escalado, los tiempos de respuesta objetivo y la comunicación durante el incidente.

Tras cada incidente significativo, el adjudicatario de manera conjunta con el equipo responsable de la operación conducirá un postmortem documentado que incluya la cronología del incidente, el análisis de causa raíz, el impacto real en los SLOs y el error budget, las acciones correctivas comprometidas con responsable y plazo, y las lecciones aprendidas. Los postmortems se

almacenarán en una base de conocimiento accesible, alimentando el aprendizaje organizativo y, a futuro, los modelos de AIOps con datos históricos de incidentes y sus causas raíz.

Runbooks y automatización de respuesta: El adjudicatario creará y mantendrá runbooks operativos documentados para los escenarios de incidencia más comunes en los servicios cloud gestionados por el CCoE, incluyendo los procedimientos de diagnóstico paso a paso, las acciones de mitigación y recuperación, los criterios de escalado, y las dependencias y contactos relevantes. Los runbooks se diseñarán con un formato estructurado que facilite su futura automatización: cada paso del runbook identificará si es automatizable (ejecutable sin intervención humana) o si requiere decisión humana, de modo que cuando se implanten capacidades de AIOps, los runbooks puedan convertirse progresivamente en flujos de remediación automatizada.

9.3 Paneles, alertas y visualización

El adjudicatario diseñará e implementará un sistema de paneles y alertas estructurado en niveles:

- **Dashboards ejecutivos**: visión de alto nivel del estado de fiabilidad de los servicios cloud de EJIE (SLOs, error budgets, disponibilidad global, tendencias), orientados al comité de seguimiento y a la dirección.
- **Dashboards operativos**: visión detallada por servicio/aplicación con las métricas de los tres pilares, el estado de los SLIs, el consumo de error budget y las alertas activas, orientados a los equipos de operación y proyecto.
- **Dashboards de servicio**: visión específica de cada solución desplegada, con las métricas de infraestructura, aplicación y negocio relevantes, orientados al equipo propietario del servicio.

Las alertas se diseñarán con criterios de reducción de ruido, priorizando las alertas basadas en SLOs (alerta cuando el servicio consume error budget de forma anómala) sobre las alertas basadas en umbrales estáticos de infraestructura (por ejemplo alerta cuando la CPU supera el 80%). Las alertas basadas en SLOs son más significativas porque reflejan el impacto real en el usuario y generan menos falsos positivos, lo que a su vez mejora la calidad de los datos disponibles para los futuros modelos de AIOps.

9.4 Evolución hacia AIOps

El adjudicatario diseñará la observabilidad y las prácticas SRE con la visión explícita de que constituyen la base sobre la que EJIE implantará capacidades de AIOps de forma progresiva. AIOps no es una herramienta que se despliega de golpe, sino una capacidad que se construye sobre tres prerrequisitos: datos de telemetría de calidad (cobertura, normalización, retención), procesos operativos maduros (gestión de incidentes, postmortems, runbooks) y competencias del equipo (comprensión de los modelos, capacidad de configuración y supervisión).

Esta labor se realizará con la colaboración y supervisión del CoE de Inteligencia Artificial del que dispone EJIE.

El adjudicatario elaborará una hoja de ruta de evolución hacia AIOps con tres fases de madurez:

Fase 1 — Observabilidad inteligente (desde el inicio del contrato)

En esta fase se establece la base de datos de telemetría y los procesos SRE, e incorpora las primeras capacidades de inteligencia sobre los datos:

- **Detección de anomalías** estadísticas en las métricas de los servicios cloud, utilizando las capacidades nativas de los hiperescalares (Amazon DevOps Guru, CloudWatch Anomaly Detection, GCP Cloud Monitoring alerting con ML) y de la plataforma de monitorización corporativa de EJIE. A diferencia de los umbrales estáticos, la detección de anomalías aprende los patrones normales de comportamiento de cada métrica y alerta cuando el comportamiento se desvía significativamente, adaptándose a la estacionalidad y las tendencias.
- **Correlación básica de eventos**: agrupación automática de alertas relacionadas (alert grouping) para reducir el ruido y presentar a los operadores una visión consolidada de los incidentes en lugar de decenas de alertas individuales. La reducción de ruido de alertas es uno de los primeros beneficios tangibles de AIOps, con reducciones de hasta el 85% del volumen de alertas reportadas en implementaciones maduras.
- **Análisis predictivo de capacidad**: uso de las tendencias históricas de métricas de consumo (CPU, memoria, almacenamiento, costes) para proyectar cuándo se alcanzarán los umbrales de capacidad, permitiendo planificar proactivamente el escalado o la optimización en lugar de reaccionar a los picos.

Fase 2 — Análisis de causa raíz asistido (a partir de los 12 meses)

Una vez establecida la base de telemetría normalizada y los procesos SRE, se incorporan capacidades más avanzadas:

- **Análisis de causa raíz asistido por IA**: uso de las trazas distribuidas, los mapas de dependencias entre servicios y las métricas correlacionadas para sugerir automáticamente la causa raíz probable de un incidente, reduciendo significativamente el tiempo medio de diagnóstico (MTTD). Los modelos utilizarán los datos de topología de servicio recopilados por OpenTelemetry y los datos históricos de incidentes (postmortems) para mejorar progresivamente sus sugerencias.
- **Recomendaciones inteligentes de remediación**: basándose en los runbooks documentados y en el historial de resolución de incidentes similares, el sistema sugerirá al operador las acciones de remediación más probables. En esta fase las recomendaciones requieren aprobación humana antes de ejecutarse.
- **Correlación cross-domain avanzada**: correlación de eventos entre los diferentes dominios de telemetría (infraestructura, aplicación, seguridad, costes) y entre los diferentes hiperescalares, proporcionando una visión unificada que ninguna herramienta nativa de un solo proveedor puede ofrecer.
- **Predicción de degradación de SLOs**: modelos que anticipen la violación de SLOs basándose en las tendencias de los SLIs y en el consumo del error budget, alertando antes de que el usuario se vea afectado.

Fase 3 — Remediación automatizada gobernada (a partir de los 24 meses)

La fase más avanzada, donde los modelos de AIOps no solo detectan y diagnostican, sino que actúan de forma controlada:

- **Remediación automatizada para escenarios de bajo riesgo**: ejecución automática de acciones de remediación predefinidas y aprobadas (reinicio de servicios, escalado de instancias, rotación de pods, limpieza de caché) cuando el sistema detecta condiciones conocidas que históricamente se

han resuelto con estas acciones. La automatización se limita a los escenarios incluidos en los runbooks clasificados como "automatizables" y opera bajo guardrails estrictos (límites de frecuencia, verificación post-acción, rollback automático si la acción empeora el estado).

- **Autonomía progresiva y gobernada**: el alcance de la remediación automatizada se ampliará de forma gradual, comenzando por los incidentes de menor riesgo (entornos no productivos, acciones reversibles) y expandiéndose a producción solo tras la validación exitosa y la aprobación explícita de EJIE para cada tipo de escenario.
- **Optimización continua y autoajuste**: los modelos de AIOps retroalimentan los procesos de optimización de costes (FinOps) y de capacidad, sugiriendo o ejecutando ajustes de right-sizing, scheduling y selección de tiers de almacenamiento basados en el análisis continuo de los patrones de consumo real.

9.5 Gobernanza de AIOps

El adjudicatario deberá alinear las prácticas de AIOps alineado con la política de gobernanza de IA implantada en EJIE y que por tanto asegure la rendición de cuentas, la trazabilidad y el control/supervisión humana.

9.6 Aplicación por niveles del modelo de adopción

Nivel 1 — Marco Mínimo

Se aplican los requisitos mínimos de observabilidad de forma centralizada por el CCoE: integración de logs de infraestructura y métricas básicas en la plataforma de monitorización corporativa, habilitación de los logs de auditoría cloud, dashboards de monitorización estándar proporcionados por el CCoE, y alertas basadas en umbrales estáticos para los indicadores críticos (disponibilidad, errores). No se requiere instrumentación OTel de la aplicación. El CCoE proporciona los SLIs básicos (disponibilidad y latencia) sin definición formal de SLOs ni error budgets. Se aplica la detección de anomalías nativa del hiperescalar sobre las métricas de infraestructura (fase 1 de AIOps).

Nivel 2 — Marco Guiado

Se amplía la observabilidad con participación del equipo de proyecto: integración de logs de aplicación con formato estructurado normalizado, instrumentación OTel básica para métricas de aplicación y tracing, dashboards específicos del servicio con métricas de infraestructura, aplicación y los KPIs de negocio más relevantes, y alertas evolucionadas que combinan umbrales estáticos con detección de anomalías. Se definen formalmente SLIs y SLOs para el servicio, con cálculo y monitorización del error budget. Se generan postmortems para incidentes significativos. Los runbooks del servicio se documentan y clasifican según su potencial de automatización. Se aplica correlación básica de eventos y análisis predictivo de capacidad (fase 1-2 de AIOps).

Nivel 3 — DevOps Golden Paths

Se aplica el marco completo de observabilidad y SRE: instrumentación OTel completa (métricas, logs, trazas distribuidas), observabilidad integral incluyendo métricas de negocio, profiling continuo para servicios críticos, dashboards multinivel (ejecutivo, operativo, servicio) con SLOs y error budgets en tiempo real, y alertas basadas en SLOs (burn rate alerts) como mecanismo primario de alerta. Los SLOs están plenamente operativos y la política de error budget gobierna las decisiones de release del equipo de proyecto. Los postmortems son práctica habitual. Los runbooks están

automatizados (o en vía de automatización). Se aplican las capacidades AIOps de las fases 2 y 3: análisis de causa raíz asistido, recomendaciones de remediación, y remediación automatizada para los escenarios aprobados por EJIE.

9.7 Entregables

El adjudicatario producirá y mantendrá actualizados los siguientes entregables:

- La estrategia de observabilidad cloud de EJIE, con el diseño de la arquitectura de recolección, procesamiento y almacenamiento de telemetría (métricas, logs, trazas) y los estándares de instrumentación OTel.
- Los estándares de logging estructurado y de cardinalidad de métricas, como documentos normativos de obligado cumplimiento para los equipos de proyecto.
- Las guías de instrumentación OTel para los lenguajes y frameworks más comunes en EJIE.
- El catálogo de SLIs y SLOs definidos para cada servicio, con las políticas de error budget asociadas.
- Los runbooks operativos documentados y clasificados por potencial de automatización.
- Los dashboards configurados a los tres niveles (ejecutivo, operativo, servicio) en la plataforma de monitorización corporativa.
- La hoja de ruta de evolución hacia AIOps con las tres fases, los criterios de avance entre fases, y la evaluación semestral de progreso.
- El informe mensual de observabilidad y fiabilidad que incluirá el estado de los SLOs de todos los servicios monitorizados, el consumo de error budget y tendencias, los incidentes producidos y su tiempo de resolución (MTTR), las alertas generadas y la ratio señal/ruido, y las acciones de mejora derivadas de los postmortems.
- El informe semestral de AIOps que incluirá las capacidades implantadas y su eficacia (reducción de ruido de alertas, reducción de MTTR, incidentes detectados proactivamente), las acciones de remediación automatizadas ejecutadas y sus resultados, la evaluación de nuevas herramientas y capacidades disponibles en el mercado, y la propuesta de ampliación de alcance para el siguiente semestre.

10 Plataforma de Ingeniería Cloud: catálogo de building blocks

10.1 Enfoque y Principios

El CCoE de EJIE, en su función de equipo de ingeniería de plataforma cloud (Infrastructure Platform Engineering), tratará las capacidades de la plataforma cloud como un producto interno cuyo consumidor son los equipos de proyecto y desarrollo de EJIE y sus clientes. Siguiendo los principios de platform engineering, el CCoE diseñará, construirá y mantendrá un catálogo de building blocks: capacidades reutilizables, componibles y gobernadas que abstraen la complejidad de los hiperescalares y permiten a los equipos de proyecto desplegar soluciones cloud de forma autónoma, rápida y segura, dentro de los guardrails establecidos.

Los building blocks se materializarán como módulos Terraform versionados, documentados y publicados en la plataforma IDP (BatArin) para su descubrimiento y consumo. Cada building block encapsulará las buenas prácticas, las políticas de seguridad y los estándares corporativos de EJIE, de forma que el equipo de proyecto que los consume no necesite ser experto en la configuración de bajo nivel del servicio cloud subyacente.

Los principios que regirán el catálogo son los siguientes: cada building block será modular y componible (se podrá usar de forma independiente o combinado con otros), seguro por defecto (incorporará los guardrails de seguridad corporativos), documentado y descubrible (publicado en Backstage con documentación completa), versionado semánticamente (cambios controlados con changelog y período de deprecación), multi-hiperescalar cuando corresponda (con implementación para AWS como mínimo y para GCP o Azure cuando el servicio equivalente sea apropiado), y verificable (con validaciones automatizadas en el pipeline de CI).

El catálogo no pretende cubrir exhaustivamente todos los servicios de cada hiperescalar desde el primer día. Se construirá progresivamente, priorizando los building blocks que aporten mayor valor a los proyectos activos de EJIE, siguiendo el principio de "empezar con tres a cinco patrones comunes y ampliar según la demanda". Lo que se presenta aquí es una propuesta inicial tal como se decía en el apartado de líneas de trabajo para ir adaptando y actualizando.

Alguno de los elementos del catálogo o patrones pueden estar delegados para que su creación o mantenimiento se haga por otros equipos de EJIE diferentes a la ingeniería de la plataforma cloud, que tengan la madurez suficiente como por ejemplo los específicos de datos y analítica por el equipo de BatData, los específicos de IA por el equipo de la plataforma IA, BatSDLC para CI/CD etc. Pero el CCoE velará por la consistencia del catálogo.

10.2 Estructura y Catálogo

El catálogo se organiza en nueve dominios funcionales. Para cada building block se indica su descripción funcional, los servicios cloud subyacentes por hiperescalar (AWS como principal, GCP como secundario), el nivel de prioridad para la construcción inicial del catálogo, y los guardrails de seguridad incorporados por defecto.

Dominio 1 - Networking				
Nombre	**Descripción**	**Servicios Subyacentes**	**Guardrails**	**Prioridad**
BB-NET-01: Red virtual corporativa (VPC/VNet)	Provisión de una red virtual aislada con la estructura de subnets estándar de EJIE (públicas, privadas para aplicación, privadas para datos, privadas para gestión), tablas de rutas, gateways NAT y conectividad con la landing zone corporativa.	AWS VPC, subnets, route tables, NAT Gateway, Internet Gateway; GCP VPC, subnets, Cloud NAT, Cloud Router.	Segmentación obligatoria por capas (aplicación, datos, gestión), prohibición de acceso directo desde internet a subnets de datos, tagging obligatorio, CIDR ranges alineados con el plan de direccionamiento corporativo.	Alta
BB-NET-02: Conectividad híbrida	Configuración de la conectividad entre el entorno cloud y la infraestructura on-premise de EJIE (CPD y cloud privado Batera), incluyendo las conexiones dedicadas y/o VPN, el peering con la VPC de tránsito de la landing zone, y las reglas de firewall asociadas.	AWS Direct Connect, Site-to-Site VPN, Transit Gateway attachments; GCP Cloud Interconnect, Cloud VPN, Network Connectivity Center.	Cifrado obligatorio en tránsito, acceso restringido a rangos IP corporativos, integración con la arquitectura hub-and-spoke de la landing zone.	Alta
BB-NET-03: Endpoints privados	Habilitación de acceso privado a los servicios gestionados del hiperescalar (almacenamiento, bases de datos, registros de contenedores, servicios de gestión de secretos) sin pasar por internet público.	AWS PrivateLink, VPC Endpoints (gateway e interface); GCP Private Google Access, Private Service Connect.	Endpoint privado obligatorio para servicios de datos en entornos de producción y restringidos	Alta
BB-NET-04: Balanceo de carga y distribución de tráfico	Provisión de balanceadores de carga para exponer aplicaciones de forma controlada, con terminación TLS, reglas de enrutamiento y protección WAF integrada	AWS Application Load Balancer (ALB), Network Load Balancer (NLB); GCP Cloud Load Balancing (HTTP/S, TCP/SSL).	Certificados TLS gestionados obligatorios, integración con WAF corporativo, logging de acceso habilitado	Alta

BB-NET-05: DNS y gestión de dominios	Gestión de zonas DNS para la resolución de nombres de los servicios cloud, tanto públicos como privados, integrada con la estrategia DNS corporativa de EJIE	AWS Route 53 (zonas públicas y privadas); GCP Cloud DNS	Zonas privadas por defecto para servicios internos, registro de auditoría de cambios DNS, TTLs estandarizados	Media
Dominio 2 – Cómputo y Contenedores				
BB-COM-01: Clúster Kubernetes gestionado	Provisión de un clúster Kubernetes gestionado con la configuración estándar de EJIE: node groups con autoescalado, integración con IAM para control de acceso, networking con CNI nativo, add-ons corporativos (CoreDNS, monitorización, ingress controller) y políticas de seguridad de pods.	AWS EKS con managed node groups; GCP GKE con node pools	Acceso al API server restringido a redes corporativas, RBAC configurado según los roles corporativos, imágenes de nodo endurecidas, integración con la plataforma de observabilidad, políticas de red (network policies) habilitadas	Alta
BB-COM-02: Máquina Virtual (VM)	Provisión de instancias de cómputo virtual con la configuración estándar de EJIE: imagen base endurecida del catálogo corporativo, integración con el servicio de gestión de parches, monitorización y backup.	AWS EC2 con launch templates; GCP Compute Engine con instance templates	Imagen base obligatoria del catálogo corporativo (no se permiten AMIs/imágenes públicas no aprobadas), disco cifrado por defecto, acceso SSH/RDP restringido a bastión corporativo, agent de monitorización preinstalado, integración con AWS Systems Manager / GCP OS Config para gestión de parches	Alta
BB-COM-03: Registro de imágenes de contenedores	Registro privado de imágenes de contenedores para almacenar, versionar y escanear las imágenes Docker utilizadas por las cargas de trabajo en Kubernetes.	AWS ECR; GCP Artifact Registry	Escaneo de vulnerabilidades automático en cada push, políticas de inmutabilidad de tags, retención configurable de imágenes antiguas, acceso restringido por IAM	Alta
BB-COM-04: Función serverless	Provisión de funciones serverless para cargas de trabajo event-driven	AWS Lambda; GCP Cloud Functions /	Permisos IAM de mínimo privilegio por función, timeout máximo	Media

	con configuración estandarizada de permisos, timeout, memoria, logging y alertas.	Cloud Run (modo serverless	configurado, integración con la plataforma de observabilidad, secretos inyectados desde el gestor centralizado (nunca en variables de entorno en claro)	
Dominio 3 - Almacenamiento				
BB-STO-01: Almacenamiento de objetos	Bucket de almacenamiento de objetos con la configuración de seguridad, cifrado, versionado y ciclo de vida estándar de EJIE	AWS S3; GCP Cloud Storage	Acceso público bloqueado por defecto, cifrado obligatorio (SSE-S3/SSE-KMS en AWS, cifrado con CMEK en GCP para datos sensibles), versionado habilitado, políticas de ciclo de vida para transición a tiers fríos (S3 Intelligent-Tiering, GCS Nearline/Coldline), logging de acceso habilitado, tagging obligatorio para FinOps.	Alta
BB-STO-02: Almacenamiento en bloque	Volúmenes de almacenamiento en bloque para instancias de cómputo y nodos de Kubernetes, con cifrado y políticas de snapshot.	AWS EBS; GCP Persistent Disk	Cifrado obligatorio, snapshots automáticos según política de retención, tipo de volumen seleccionado según perfil de rendimiento (no sobreprovisionar IOPS)	Media
BB-STO-03: Sistema de ficheros compartido	Sistema de ficheros compartido para cargas de trabajo que requieren acceso concurrente a un volumen de datos desde múltiples instancias o pods	AWS EFS; GCP Filestore	Cifrado en reposo y en tránsito, acceso restringido a subnets específicas, throughput dimensionado según los requisitos (evitar sobredimensionamiento)	Baja
Dominio 4 – Bases de Datos y Datos				
BB-DAT-01: Base de datos relacional gestionada	Provisión de una instancia de base de datos relacional gestionada con alta disponibilidad, backup, cifrado y monitorización estándar.	AWS RDS (PostgreSQL, MySQL, Oracle, SQL Server) y Aurora; GCP Cloud SQL y AlloyDB	Despliegue multi-AZ obligatorio en producción, cifrado con CMK para datos sensibles, backup automatizado con retención mínima de 30 días, acceso restringido a subnets privadas de datos (no accesible desde internet),	Alta

			monitorización de rendimiento (Performance Insights / Query Insights) habilitada, parámetros de seguridad del motor según los estándares corporativos.	
BB-DAT-02: Base de datos NoSQL gestionada	Provisión de un servicio de base de datos NoSQL gestionado para cargas de trabajo que requieren esquemas flexibles, baja latencia y alta escalabilidad.	AWS DynamoDB; GCP Firestore / Cloud Bigtable	Cifrado obligatorio, modo de capacidad (on-demand vs. provisionado) seleccionado según patrón de acceso, backup habilitado, acceso restringido por IAM y endpoint privado.	Media
BB-DAT-03: Caché en memoria	Provisión de un servicio de caché gestionado para mejorar el rendimiento de aplicaciones que requieren acceso frecuente a datos con baja latencia.	AWS ElastiCache (Redis, Memcached); GCP Memorystore.	Cifrado en tránsito y en reposo, despliegue en subnet privada, autenticación habilitada, monitorización de uso de memoria y tasa de aciertos.	Media
BB-DAT-04: Cola de mensajería y eventos	Provisión de servicios de mensajería asíncrona y streaming de eventos para la comunicación desacoplada entre componentes de aplicación	AWS SQS, SNS, EventBridge, MSK (Kafka gestionado); GCP Pub/Sub, Eventarc	Cifrado obligatorio, dead letter queue configurada, monitorización de profundidad de cola y latencia de procesamiento, políticas de reten	Media
Dominio 5 – Identidad, Accesos y Secretos				
BB-IAM-01: Rol de servicio y políticas de acceso	Creación de roles IAM de servicio con políticas de mínimo privilegio para aplicaciones y cargas de trabajo, incluyendo la federación OIDC con GitLab para pipelines CI/CD	AWS IAM Roles, Policies, OIDC Identity Provider; GCP IAM Service Accounts, Workload Identity Federation	Prohibición de políticas gestionadas por defecto del proveedor (sustituidas por custom policies granulares), no se permiten credenciales estáticas (access keys) — solo credenciales temporales vía STS/OIDC, revisión trimestral de permisos no utilizados	Alta
BB-IAM-02: Gestión de secretos	Almacenamiento y distribución segura de secretos (credenciales de bases de datos, tokens de API, claves	AWS Secrets Manager; GCP Secret Manager.	Rotación automática de secretos cuando el servicio lo soporte, acceso restringido por IAM a los consumidores	Alta

	de cifrado de datos) para su consumo por aplicaciones y pipelines		autorizados, prohibición de secretos en código o ficheros de configuración, integración con los pipelines de GitLab para inyección en tiempo de ejecución, registro de auditoría de acceso a secretos	
BB-IAM-03: Gestión de claves de cifrado	Provisión de claves de cifrado gestionadas por el cliente (CMK) para el cifrado de datos sensibles en reposo	AWS KMS; GCP Cloud KMS	Rotación anual automática de claves, políticas de acceso que separen el rol de administrador de claves del rol de consumidor de datos (separación de funciones), registro de auditoría de uso de claves	Alta
Dominio 6 – Seguridad y Cumplimiento				
BB-SEC-01: Configuración de seguridad de cuenta/proyecto	Configuración de seguridad base para cada nueva cuenta AWS o proyecto GCP dentro de la estructura organizativa de la landing zone, incluyendo los guardrails, la configuración de logging, los controles de seguridad y la integración con la plataforma de postura de seguridad.	AWS Security Hub, Config, CloudTrail, GuardDuty; GCP Security Command Center, Cloud Audit Logs, Organization Policy Service.	Habilitación obligatoria de CloudTrail/Audit Logs, activación de Security Hub/SCC con los estándares CIS y ENS, habilitación de GuardDuty para detección de amenazas, configuración de alertas de hallazgos de alta severidad hacia el equipo de seguridad	Alta
BB-SEC-02: Firewall y protección de aplicaciones web	Configuración de Web Application Firewall con las reglas base corporativas para la protección de aplicaciones expuestas a internet	AWS WAF (reglas gestionadas + custom rules); GCP Cloud Armor	conjunto de reglas base corporativas (OWASP Top 10, protección contra bots, rate limiting), integración con los balanceadores de carga (BB-NET-04), logging de peticiones bloqueadas	Media
Dominio 7 – Observabilidad y Monitorización				
BB-OBS-01: Stack de monitorización y alertas	Configuración del stack de monitorización estándar para una carga de trabajo, incluyendo la recolección de	AWS CloudWatch (métricas, alarmas, dashboards); GCP Cloud	métricas base habilitadas para todos los recursos (CPU, memoria, disco, red, errores), alertas estándar para umbrales	Alta

	métricas de infraestructura, la integración con la plataforma de monitorización corporativa, y las alertas base.	Monitoring. Integración con la plataforma corporativa de EJIE vía OpenTelemetry Collector.	críticos, dashboards generados automáticamente, retención de métricas según política corporativa	
BB-OBS-02: Centralización de logs	Configuración de la recolección y envío de logs de la carga de trabajo al sistema de logging centralizado de EJIE, con el formato estructurado normalizado.	AWS CloudWatch Logs, Kinesis Data Firehose; GCP Cloud Logging, Log Router. Exportación a la plataforma corporativa vía OpenTelemetry Collector.	Formato de log estructurado obligatorio (JSON con campos normalizados), políticas de retención diferenciadas por tipo de log, filtrado de datos sensibles en logs antes del envío.	Alta
BB-OBS-03: Tracing distribuido	Instrumentación y recolección de trazas distribuidas para aplicaciones desplegadas en arquitecturas de microservicios o multi-componente.	AWS X-Ray; GCP Cloud Trace. Instrumentación vía OpenTelemetry SDK y Collector.	Muestreo configurado según volumen y criticidad (tail-based sampling), correlación con logs y métricas vía trace_id, integración con la plataforma de observabilidad corporativa	Media (obligatorio en Niveles 2 y 3)
Dominio 8 – CI/CD y Automatización				
BB-CIC-01: Pipeline CI/CD estándar para IaC	Plantilla de pipeline de GitLab CI/CD para la ejecución de flujos de IaC Terraform (init, validate, plan, apply) con las validaciones de seguridad (PaC), estimación de costes y aprobaciones integradas.	GitLab CI/CD, Terraform CLI, herramientas de PaC (Checkov, OPA), Infracost.	Separación estricta plan/apply, aprobación de merge request obligatoria para apply en producción, validación de PaC como gate bloqueante, autenticación con hiperescalares vía OIDC (sin credenciales estáticas en el pipeline).	Alta
BB-CIC-02: Pipeline CI/CD para aplicaciones contenerizadas	Plantilla de pipeline de GitLab CI/CD para el ciclo de vida de aplicaciones contenerizadas: build de imagen, escaneo de vulnerabilidades, push al registro, despliegue en Kubernetes.	GitLab CI/CD, Docker, registro de contenedores (BB-COM-03), herramientas de escaneo (Trivy, Amazon Inspector, GCP Artifact Analysis)	Escaneo de vulnerabilidades obligatorio antes del push (gate bloqueante para vulnerabilidades críticas), imagen base del catálogo corporativo, tag de imagen inmutable, despliegue en producción con aprobación.	Alta

BB-CIC-03: Automatización de entornos efímeros	Provisión automatizada de entornos completos de desarrollo o testing a partir de un pipeline, con destrucción automática tras un período configurable de inactividad.	Composición de múltiples building blocks (BB-NET-01, BB-COM-01/02, BB-DAT-01) orquestados por GitLab CI/CD y Terraform.	TTL (time to live) obligatorio con destrucción automática, costes acotados mediante instancias de menor tamaño y scheduling de apagado, aislamiento de red respecto a producción.	Media
Dominio 9 – FinOps y Gobernanza de Costes				
BB-FIN-01: Etiquetado corporativo (tagging)	Módulo Terraform que aplica de forma obligatoria y consistente el conjunto de etiquetas corporativas de EJIE a todos los recursos provisionados por cualquier otro building block.	Sistema de tags nativo de AWS y GCP (labels)	Etiquetas obligatorias (proyecto, responsable, entorno, centro de coste, fecha de creación, velocidad del modelo de adopción), validación de valores permitidos, integración con Cloudability para la atribución de costes.	Alta
BB-FIN-02: Presupuesto y alertas de coste	Configuración de presupuestos y alertas de coste por cuenta/proyecto, con umbrales de aviso configurables y notificación automática al responsable.	AWS Budgets; GCP Budget Alerts. Integración con Cloudability.	Presupuesto obligatorio para cada cuenta/proyecto productivo, alertas al 50%, 80% y 100% del presupuesto, notificación al responsable y al CCoE cuando se supera el 100%	Alta

10.3 Patrones de arquitectura de referencia (composiciones de building blocks)

Además de los building blocks individuales, el CCoE construirá y mantendrá composiciones que combinen múltiples building blocks para expresar las arquitecturas de solución más comunes en EJIE. Los patrones (arquetipos) iniciales a desarrollar serán:

Patrón	Descripción	Building Blocks Utilizados
PAT-01: Aplicación web N-Tier	Composición para desplegar una aplicación web clásica de tres capas (frontend, backend, base de datos) con alta disponibilidad y seguridad	BB-NET-01 (VPC), BB-NET-04 (ALB con WAF), BB-COM-02 (instancias de cómputo) o BB-COM-01 (EKS), BB-DAT-01 (RDS), BB-STO-01 (S3 para activos estáticos), BB-IAM-01 (roles de servicio), BB-OBS-01 y BB-OBS-02 (monitorización y logs), BB-FIN-01 (tagging).

Patrón	Descripción	Building Blocks Utilizados
PAT-02: Microservicios sobre Kubernetes	Composición para desplegar una arquitectura de microservicios sobre Kubernetes gestionado con ingress, service mesh, registro de contenedores, mensajería y observabilidad completa. Combina: BB-NET-01 (VPC con subnets dimensionadas para pods)	BB-COM-01 (EKS/GKE), BB-COM-03 (registro de contenedores), BB-NET-04 (ALB/ingress controller), BB-DAT-01 o BB-DAT-02 (base de datos), BB-DAT-04 (mensajería), BB-IAM-01 (IRSA/Workload Identity), BB-OBS-01/02/03 (observabilidad completa con tracing), BB-CIC-02 (pipeline de contenedores), BB-FIN-01 (tagging).
PAT-03: Solución de datos y analytics	Composición para desplegar un entorno de datos con almacenamiento de objetos como data lake, base de datos para consultas analíticas y pipeline de procesamiento	BB-STO-01 (S3/GCS como data lake), BB-DAT-01 (base de datos analítica), BB-NET-03 (endpoints privados para acceso a datos), BB-IAM-01/03 (roles y cifrado CMK para datos sensibles), BB-OBS-01/02 (monitorización y logs), BB-FIN-01 (tagging con etiquetas de proyecto de datos).
PAT-04: Aplicación serverless event-driven	Composición para desplegar una aplicación basada en funciones serverless y eventos, con almacenamiento y APIs gestionadas.	BB-COM-04 (funciones Lambda/Cloud Functions), BB-DAT-02 (DynamoDB/Firestore), BB-DAT-04 (EventBridge/Pub/Sub, SQS), BB-STO-01 (S3/GCS), BB-IAM-01/02 (roles y secretos), BB-OBS-01/02 (monitorización y logs), BB-FIN-01 (tagging).
PAT-05: Entorno de desarrollo/sandbox	Composición ligera para provisionar un entorno de desarrollo o experimentación con costes acotados. Guardrails mínimos de seguridad y un presupuesto máximo configurado.	BB-CIC-03 (entorno efímero)

10.4 Publicación en BatArin y modelo de consumo

Cada building block y patrón de arquitectura se publicará en BatArin (Backstage de EJIE) como una entrada del catálogo de servicios cloud con la siguiente información: nombre y código del building block, descripción funcional y caso de uso, diagrama de arquitectura (para los patrones), variables de entrada con descripción, tipo, valores por defecto y restricciones, outputs producidos, dependencias con otros building blocks, versión actual y changelog, nivel de velocidad mínimo requerido para su consumo autónomo, enlace al repositorio GitLab con el código fuente, y ejemplo de uso completo.

Los equipos de proyecto podrán consumir los building blocks de dos formas: autoservicio directo (Niveles 2 y 3) — referenciando el módulo Terraform desde su propio código IaC y pasando las variables de configuración, con el pipeline de CI/CD validando automáticamente el cumplimiento de políticas; o solicitud asistida (Nivel 1) — solicitando al CCoE o al equipo de implantación designado el despliegue de la solución, que se realizará utilizando los mismos building blocks del catálogo.

10.5 Hoja de ruta de construcción del catálogo

El adjudicatario construirá el catálogo de building blocks de forma progresiva, alineada con las necesidades reales de los proyectos de EJIE. La priorización propuesta es la siguiente:

- **Fase 1:** todos los building blocks marcados como prioridad "alta" y el patrón PAT-01 (aplicación web N-Tier). Esto cubre los fundamentos de red, cómputo, almacenamiento de objetos, bases de datos relacionales, identidad/secretos/cifrado, seguridad de cuenta, monitorización y logs, pipelines CI/CD y tagging. Este conjunto mínimo viable permite desplegar la mayoría de cargas de trabajo típicas de EJIE con los guardrails corporativos incorporados.
- **Fase 2**: building blocks de prioridad "media", patrones PAT-02 (microservicios) y PAT-04 (serverless). Se amplía el catálogo con tracing distribuido, bases de datos NoSQL, caché, mensajería, funciones serverless, entornos efímeros, DNS y protección DDoS.
- **Fase 3**: building blocks de prioridad "baja", patrones PAT-03 (datos) y PAT-05 (sandbox), y nuevos building blocks que surjan de la demanda de los proyectos. Se aborda almacenamiento compartido y se evalúan nuevos building blocks para servicios emergentes si la demanda de EJIE lo justifica.

El adjudicatario revisará trimestralmente la priorización del backlog de building blocks con EJIE, ajustándola según los proyectos previstos y la retroalimentación de los equipos de proyecto.

10.6 Mantenimiento y Evolución

El adjudicatario será responsable de mantener el catálogo actualizado frente a la evolución de los servicios de los hiperescalares, las actualizaciones de Terraform y las nuevas necesidades de EJIE. Esto incluye la actualización de los módulos cuando los providers de Terraform o los servicios cloud introduzcan cambios relevantes, la deprecación ordenada de building blocks obsoletos (con período de aviso de al menos un sprint y plan de migración), la incorporación de nuevos building blocks propuestos por la comunidad de práctica cloud de EJIE (con curación y review por el CCoE antes de su publicación), y el informe trimestral del estado del catálogo que incluirá el número de building blocks disponibles por dominio, la tasa de adopción de cada building block (proyectos que lo consumen), las contribuciones recibidas de la comunidad de práctica, y la hoja de ruta de nuevos building blocks para el siguiente trimestre.

11 Configuración del Servicio

La Composición del equipo aportado por el adjudicatario deberá cubrir todas las funciones definidas en las líneas de trabajo mencionadas (apartado 3), con un equipo multidisciplinar. El equipo contará al menos con los roles clave siguientes, junto con la dedicación anual estimada para cada uno, expresada en horas de dedicación por año:

11.1 Composición del Equipo de Trabajo. Roles, responsabilidades y dedicación

Rol / Perfil	Descripción de responsabilidades principales	Dedicación Fija (horas/año)	Dedicación Variable (horas/año)
Coordinador del CCoE (Responsable del Servicio)	Responsable global del contrato y punto de contacto principal con EJIE. Coordina el servicio en su conjunto, asegurando el cumplimiento de objetivos, calidad y plazos. Supervisión de compromisos, reporting a órganos de gobierno, gestión de riesgos, planificación de actividades y alineación estratégica.	1.200	
Arquitecto Cloud (AWS principal / GCP secundario)	Diseña arquitecturas de referencia cloud y define estrategias para evolución de landing zones en AWS (y GCP). Asegura la alineación con la arquitectura corporativa, last best-prectice y requisitos de seguridad, integrando soluciones on-premises con nube pública. Participa en la definición de building blocks y blueprint de Infraestructura como Código.	1.700	
Especialista Cloud / DevOps (*Cloud Engineer / SRE*)	Implementa arquitecturas de referencia y mantiene las automatizaciones (pipelines CI/CD, flujos GitOps) y módulos IaC, asegurando su eficacia y seguridad. Presta soporte técnico a los equipos de proyectos en integraciones de CI/CD, resolviendo incidencias y optimizando pipelines. Lidera la adopción del marco de desarrollo IaC en los proyectos, adaptando las plantillas a casos particulares.	1.700	800
Ingeniero de Seguridad Cloud	Establece controles de seguridad cloud y cumplimiento normativo. Gestiona integración de la nube con sistemas corporativos de identidad (IAM), aplica políticas de seguridad (cifrado, segmentación de redes, gestión de claves) y monitorización de logs y auditoría, garantizando la	1.700	

Rol / Perfil	Descripción de responsabilidades principales	Dedicación Fija (horas/año)	Dedicación Variable (horas/año)
	adherencia al ENS y RGPD. Colabora en las revisiones de seguridad de IaC y pipelines DevSecOps.		
Consultor FinOps	Encargado de monitorizar y optimizar costes en la nube. Define y sigue KPIs financieros, implementa mecanismos FinOps (etiquetado de recursos, *showback/chargeback*, optimización de dimensionamiento) y propone medidas de ahorro. Entrega informes periódicos de costes y *reportings* de eficiencia financiera.	800	
Especialista Modernización / Cloud Native	Perfil complementario al del especialista Cloud / DevOps centrado en la ingeniería de software enfocada en transformar aplicaciones legacy en sistemas modulares y escalables, diseñados para ejecutarse de manera óptima en entornos de nube pública, privada o híbrida. Proponer estrategias de migración: Evaluar y rediseñar aplicaciones monolíticas tradicionales. Apoyar la visión de plataforma de aplicaciones sobre la plataforma de ingeniería cloud. Acompañar a equipos específicos dedicados de modernización de aplicaciones.	800	
Formador/Consultor de Adopción Cloud	Diseña y ejecuta programas de formación y evangelización en *DevOps, IaC, SRE, Cloud-native development*, etc., adaptados a los distintos niveles de madurez (Velocidades 1-3). Apoya a los equipos en su transición cultural hacia DevOps, promueve comunidades de práctica internas y difunde casos de éxito. Documenta y actualiza los manuales y guías operativas del CCoE.		1.000
Estrategia y Consultoría	Especialización de nivel muy alto y experiencia en el ámbito de las tecnologías cloud definidas en el entorno tecnológico, prestando labores de asistencia técnica y complementando al resto de roles. Estos perfiles intervendrán bajo demanda, previa valoración por jornadas aceptada expresamente por EJIE y se facturarán al precio ofertado en el mes siguiente en el que se hayan prestado. El nivel de dedicación de la tabla se debe interpretar por tanto para este caso como un máximo.		4.300

Adicionalmente las horas de cada perfil, se aportarán otras **4.300 horas anuales para un posible servicio de consultoría adicional** que se prestará mediante los recursos especializados que proporcione el adjudicatario. Para la ejecución de estas horas, se seguirá la siguiente metodología:

- Mensualmente EJIE establecerá tareas de consultoría que se ejecutarán bajo petición para resolver una necesidad concreta que surja en el ámbito objeto del contrato.
- Una vez recibida la petición, la empresa adjudicataria presentará un plan detallado de recursos y horas para ejecutar dichas tareas, que tendrá que ser aprobado por EJIE previamente a su ejecución.
- La empresa adjudicataria asumirá los retrasos que se deriven de la ejecución de las tareas de consultoría acordadas
- La dedicación se facturará al precio ofertado y por servicio efectivamente ejecutado con el visto bueno de EJIE en el mes siguiente en el que se hayan prestado. El nivel de dedicación de la tabla se debe interpretar, por tanto, para este caso, como un máximo.

Estos volúmenes son los que se han utilizado para calcular el presupuesto máximo

***Nota**:* Los roles anteriores pueden ser cubiertos por uno o varios profesionales, siempre que se garantice una cobertura completa de las competencias y dedicaciones mínimas señaladas. Por ejemplo, algunos roles pueden ser asumidos por una misma persona con perfil mixto, siempre que esta posea la experiencia acreditada en ambas áreas y la dedicación total sea la suma indicada. En cualquier caso, los licitadores deberán proponer una organización del equipo que incluya al menos estos roles y justifique las horas anuales asignadas a cada uno, de modo de asegurar la disponibilidad del conocimiento y la capacidad de respuesta necesarias para cumplir con las obligaciones del pliego.

Responsabilidades diferenciadas por nivel: Además de las funciones generales, cada rol deberá tener responsabilidades específicas según el nivel del modelo de adopción. Por ejemplo, en Nivel 1, el Especialista Cloud tendrá la mayoría de las tareas de implementación (implementando IaC/pipelines), mientras que en Nivel 3 actuará más como coach y auditor de buenas prácticas para un equipo ya avanzado. Este reparto detallado se incluirá por parte del adjudicatario en su oferta como propuesta inicial que se podrá ir adaptando durante el periodo de vigencia del contrato.

11.2 Garantía de Disponibilidad del Servicio

En todo caso y bajo cualquier circunstancia, la adjudicataria deberá garantizar la disponibilidad continua del servicio objeto del contrato.

A tal efecto, el adjudicatario deberá definir y mantener una estrategia de continuidad del servicio que contemple las políticas, procedimientos y mecanismos necesarios para hacer frente a contingencias materiales o humanas que puedan afectar a la correcta prestación del servicio (desastres, emergencias, indisponibilidad de recursos, incumplimiento de plazos, etc.). Con carácter obligatorio, el adjudicatario deberá cumplir los siguientes requisitos:

- Disponer de un Plan de Continuidad del Servicio que garantice la provisión del servicio en modo contingencia ante cualquier situación que impida su ejecución normal. Dicho plan deberá identificar

claramente los roles, responsabilidades y recursos necesarios para su correcta activación y ejecución.

- Contar con un plan de pruebas periódicas (simulacros) que cubra de forma integral los escenarios de continuidad del servicio, incluyendo la generación de evidencias documentales que permitan su verificación a solicitud de EJIE.
- Definir e implantar planes de mejora derivados de los resultados de dichas pruebas, cuando proceda.
- Disponer de un canal de comunicación específico para informar a EJIE de forma inmediata sobre cualquier incidente que active el plan de continuidad, manteniendo una comunicación periódica mientras dure la situación de contingencia, así como informando del restablecimiento del servicio en condiciones normales.

11.3 Plan de Transición

En la oferta, el licitador deberá presentar de forma clara y detallada tanto el plan de transición de entrada al servicio como el plan de transición de salida al finalizar el contrato o cualquiera de sus prórrogas.

Al finalizar la vigencia del contrato, el adjudicatario estará obligado a prestar el apoyo necesario y a realizar la transferencia completa del conocimiento, documentación, información técnica y operativa al nuevo adjudicatario, en caso de que éste sea distinto, garantizando la continuidad del servicio.

El plazo máximo para la ejecución de cada uno de los planes de transición será de un (1) mes.

12 Metodología de Trabajo y entorno tecnológico

El adjudicatario deberá integrarse en la metodología de trabajo definida por EJIE para el CCoE, que combina principios ágiles (equipos pluridisciplinares, *sprints* y backlog priorizado por EJIE) con prácticas *ITIL/ITSM* para la gestión eficiente y la mejora continua de los servicios. Esto implica la participación en rituales ágiles (p.ej., *dailies*, *retrospectives*, *planning meetings* para ajustar las prioridades con EJIE) y simultáneamente la adherencia a procedimientos formales para gestión de cambios, incidencias y *releases* según la normativa interna de EJIE y su sistema de gestión de calidad. Cualquier incidencia de nivel crítico relacionada con el servicio deberá tratarse según los protocolos de soporte de EJIE (ServiceNow) y ser comunicada, gestionada y resuelta de acuerdo a los acuerdos de nivel de servicio definidos.

12.1 Metodología del CCoE

La **metodología del CCoE** se articulará alrededor de los siguientes principios:

- Colaboración estrecha EJIE-adjudicatario: el equipo adjudicatario trabajará *codo con codo* con los referentes de EJIE, integrándose en la operativa diaria (incluso se podrán solicitar puntualmente sesiones de trabajo presenciales además de virtuales según necesidades) para asegurar transferencia de conocimiento y alineación continua.
- Estandarización y reutilización: se impulsará la generación de artefactos reutilizables (p. ej., módulos de código Terraform, plantillas de pipeline) para evitar duplicidad de esfuerzos y lograr eficiencias en el tiempo. Todos los proyectos cloud deberán emplear estos artefactos standard.
- Automatización extensiva: se promoverá la automatización de tareas manuales para reducir errores y acelerar la entrega (por ej., mediante IaC, *scripts* de gestión, *CI/CD pipelines*), siguiendo la premisa de *"lo que se pueda automatizar, se automatiza"*. La Infraestructura como Código (IaC) se considerará requisito fundamental para la agilidad y control de costes en la nube, evitándose configuraciones manuales que impidan la trazabilidad o repetibilidad.
- Seguridad y calidad integradas: se aplicará el enfoque de DevSecOps, integrando actividades de validación y prueba en cada fase del ciclo de vida de despliegue (pruebas unitarias, integraciones en pipeline, revisiones de seguridad antes de producir). Ningún despliegue en producción se realizará sin pasar por los *gates* de calidad y seguridad definidos.
- Transparencia y reporting continuo: se habilitarán métricas e indicadores para monitorizar la eficacia del servicio (adopción de DevOps por niveles, compliance de ANS, grado de reutilización de building blocks, ahorros de costes por FinOps, etc.) con reportes periódicos (ej. informes mensuales, comités de seguimiento) hacia EJIE y hacia los órganos de gobierno pertinentes.

12.2 Uso de Inteligencia Artificial para potenciar a productividad del equipo

Se espera que el adjudicatario integre herramientas de IA como aceleradoras de la productividad del equipo del CCoE, aplicándolas de forma gobernada y alineada con los requisitos de seguridad y confidencialidad de EJIE. La IA no sustituirá el criterio experto de los profesionales del equipo, sino que potenciará su capacidad de entrega, reducirá el tiempo dedicado a tareas repetitivas y permitirá que el equipo concentre su esfuerzo en las decisiones de diseño, arquitectura y gobernanza que requieren conocimiento humano.

Ámbitos de Aplicación

El adjudicatario aplicará herramientas de IA para potenciar la productividad del equipo del CCoE en los siguientes ámbitos de trabajo, priorizando aquellos donde la ganancia de productividad es más demostrable y el riesgo más controlable:

- **Generación y aceleración de código IaC:** Los asistentes de codificación con IA se utilizarán para acelerar la escritura de código Terraform (HCL), reduciendo el tiempo necesario para generar módulos, recursos y configuraciones. Esto incluye la generación de borradores iniciales de módulos Terraform a partir de especificaciones de alto nivel o lenguaje natural, la autocompletación contextualizada de bloques de recursos y configuraciones, la conversión de patrones de arquitectura documentados en código IaC funcional, y la generación de código boilerplate para nuevos building blocks, permitiendo al ingeniero concentrarse en la lógica específica. El adjudicatario deberá documentar las herramientas de IA que utilizará para este fin (por ejemplo, GitHub Copilot, Amazon Q Developer, asistentes integrados en el IDE, u otras) y las políticas de uso que aplicará.

- **Revisión de código y detección de problemas:** La IA se empleará como capa complementaria de revisión de código que refuerce los procesos de code review humano existentes, incluyendo la detección automatizada de errores comunes, antipatrones y configuraciones potencialmente inseguras en el código Terraform antes de la revisión humana, la identificación de desviaciones respecto a los estándares corporativos de IaC de EJIE (convenciones de nombrado, estructura de módulos, buenas prácticas), el análisis de los planes de Terraform (terraform plan) para generar resúmenes legibles del impacto de los cambios propuestos, y la asistencia en la resolución de errores de ejecución y fallos de pipeline, reduciendo el tiempo de diagnóstico.

- **Seguridad y cumplimiento** El CCoE integrará capacidades de IA en los flujos de validación de seguridad del código IaC, en complemento de las herramientas de Policy as Code (OPA, Checkov, tfsec) ya descritas en otros apartados del pliego. Esto abarcará el análisis inteligente de configuraciones para detectar misconfiguraciones de seguridad que las reglas estáticas de PaC podrían no cubrir (permisos excesivamente amplios, recursos expuestos inadvertidamente, patrones de red inseguros), la priorización de hallazgos de seguridad basada en el contexto y el riesgo real frente a la simple aplicación de reglas binarias, y la generación asistida de remediaciones para los hallazgos de seguridad, produciendo propuestas de corrección que el ingeniero revisa y aplica.

- **Documentación y transferencia de conocimiento:** La IA se utilizará para acelerar y mejorar la calidad de la documentación técnica producida por el CCoE, incluyendo la generación automática de documentación de módulos Terraform (descripción de variables, outputs, dependencias, ejemplos de uso), la creación de borradores de runbooks, guías de operación y ADRs (Architecture Decision Records) a partir de las decisiones técnicas del equipo, la generación de resúmenes técnicos de cambios para los merge requests que faciliten la revisión, y la asistencia en la preparación de materiales de formación y capacitación para los equipos de EJIE.

- **Operaciones y troubleshooting:** En el ámbito operativo del CCoE, la IA se aplicará para el análisis asistido de logs y trazas para reducir el tiempo de diagnóstico de incidentes en los entornos cloud, la generación de consultas y filtros para las herramientas de monitorización y observabilidad, y la asistencia en la interpretación de alertas de costes y recomendaciones de optimización de Cloudability.

Gobernanza del uso de IA

El adjudicatario establecerá un marco de gobernanza para el uso de herramientas de IA dentro del equipo del CCoE, que asegure que los beneficios de productividad se obtienen sin comprometer la seguridad, la calidad ni la confidencialidad de los activos de EJIE.

Principio fundamental: IA como asistente, no como ejecutor autónomo: Todo código, configuración o artefacto generado o modificado con asistencia de IA será tratado exactamente igual que el código producido manualmente: se someterá a los mismos procesos de code review, validación de PaC, testing automatizado y aprobación de merge request definidos en el MD-IaC. En ningún caso una herramienta de IA tendrá acceso directo a ejecutar cambios en los entornos cloud de EJIE (por ejemplo, ejecutar terraform apply de forma autónoma). La IA genera propuestas; el ingeniero humano decide, revisa y ejecuta.

Confidencialidad y protección de datos: El adjudicatario garantizará que las herramientas de IA utilizadas no envíen a servicios externos información sensible de EJIE, incluyendo credenciales, tokens, claves de acceso, datos de configuración de producción, código propietario o información de arquitectura que pueda comprometer la seguridad de los entornos. Para ello, el adjudicatario priorizará el uso de herramientas de IA que puedan ejecutarse en entornos privados o que ofrezcan garantías contractuales de no retención ni uso de los datos para el entrenamiento de modelos. Antes de implantar cualquier herramienta de IA, el adjudicatario presentará a EJIE un análisis de la herramienta que incluya su modelo de procesamiento de datos (dónde se procesan los prompts, si los datos se retienen, si se utilizan para entrenamiento), las certificaciones de seguridad y cumplimiento del proveedor, y la evaluación de riesgos de confidencialidad en el contexto de EJIE. EJIE deberá aprobar el uso de cada herramienta de IA antes de su implantación en el equipo del CCoE.

Calidad y prevención de deuda técnica: El uso de IA acelera la generación de código, pero también puede amplificar la propagación de errores o antipatrones si no se supervisa adecuadamente. Para mitigar este riesgo, el adjudicatario aplicará las siguientes prácticas: todo código generado con IA se validará contra las mismas reglas de PaC, linting y testing automatizado que el código escrito manualmente; los ingenieros del CCoE que utilicen IA serán responsables de comprender y poder explicar todo el código que produzcan con asistencia de IA, evitando la aceptación acrítica de sugerencias; se realizarán auditorías periódicas de la calidad del código generado con IA para detectar patrones de deuda técnica (uso de APIs deprecadas, configuraciones de seguridad deficientes, código duplicado o innecesariamente complejo); y el CCoE mantendrá guías internas de prompting efectivo para IaC, documentando los patrones de prompt que producen resultados de mayor calidad en el contexto de los estándares de EJIE.

Trazabilidad: Las contribuciones asistidas por IA serán trazables en el repositorio de GitLab. El adjudicatario establecerá una convención para identificar en los commits o merge requests cuándo se ha utilizado asistencia de IA en la generación del código, de forma que las revisiones de calidad puedan considerar este factor.

Capacitación del equipo en el uso efectivo de IA

El adjudicatario asegurará que los profesionales que integran el equipo del CCoE poseen las competencias necesarias para utilizar las herramientas de IA de forma efectiva y segura. Esto incluye la formación en técnicas de ingeniería de prompts (prompt engineering) aplicadas a IaC y DevOps, el conocimiento de las limitaciones y riesgos de los modelos de IA (alucinaciones, código inseguro, APIs deprecadas, sesgos en las sugerencias), la capacidad de evaluar críticamente las sugerencias de IA antes de aceptarlas, y la familiarización con las políticas de gobernanza de IA establecidas para el CCoE.

Evolución y mejora continua

El adjudicatario evaluará periódicamente (al menos semestralmente) la eficacia de las herramientas de IA implantadas, midiendo su impacto real en la productividad del equipo y proponiendo a EJIE la incorporación de nuevas herramientas o capacidades cuando aporten un valor demostrable. Esta evaluación incluirá métricas de productividad (reducción del tiempo de desarrollo de módulos IaC, reducción del tiempo de resolución de incidentes, volumen de código revisado con asistencia de IA), métricas de calidad (tasa de defectos en código generado con IA frente a código manual, hallazgos de seguridad en código asistido por IA), evaluación de nuevas herramientas disponibles en el mercado relevantes para el contexto del CCoE (asistentes especializados en Terraform, herramientas de revisión de IaC con IA, asistentes de operaciones cloud), y análisis de riesgos emergentes (nuevas vulnerabilidades asociadas al uso de IA en la cadena de entrega de software).

El CCoE compartirá con EJIE los aprendizajes y buenas prácticas derivados de su uso de IA, de forma que estos conocimientos puedan beneficiar a otros equipos de desarrollo y operaciones de EJIE que consideren adoptar herramientas similares.

Entregables

El adjudicatario producirá y mantendrá actualizados los siguientes entregables en relación con el uso de IA en el CCoE:

- El catálogo de herramientas de IA autorizadas para el CCoE, con el análisis de seguridad y confidencialidad de cada una y la aprobación de EJIE.
- Las políticas de uso de IA del CCoE, incluyendo las directrices de gobernanza, confidencialidad, trazabilidad y calidad descritas en este apartado.
- La guía interna de prompting efectivo para IaC y DevOps, actualizada con los patrones que producen mejores resultados en el contexto de EJIE.
- El informe semestral de impacto de la IA en la productividad del CCoE, con las métricas de productividad y calidad, los riesgos identificados y las propuestas de evolución.

12.3 Entorno Tecnológico

El alcance tecnológico del CCoE es **amplio y multicloud**. Se resume a continuación:

- **Nubes públicas:** el cloud de referencia será AWS (hiperescalar principal en EJIE) para el despliegue de la mayoría de las soluciones. Adicionalmente, en Google Cloud Platform (GCP), al ser la nube secundaria en esta fase, se contemplará su integración y evolución (p.ej., para casos de uso específicos). El adjudicatario deberá asumir las landing zones de AWS, GCP y Azure existentes en EJIE – la de AWS en un estado mayor de madurez; (la de GCP y Azure más básicas) – y evolucionarlas tecnológicamente durante el contrato, incorporando mejoras que las equiparen en robustez y funcionalidad. Según la estrategia de sourcing se podría también incorporar OCI, u alguno de los otros hiperescalares relevantes en el mercado.

- **Infraestructura On-premise:** El modelo es híbrido, por lo que la nube pública deberá integrarse con entornos on-premises (infraestructura tradicional del CPD de EJIE y cloud privado Batera construido sobre vCF). El adjudicatario deberá contemplar la interoperabilidad con las soluciones on-premises, por ejemplo a través de conectividad dedicada (MPLS, VPN, Direct Connect, etc.) y mecanismos de integración de identidades y redes entre nubes y CPDs on-premises.

- **Herramientas de DevOps**: La pila de herramientas corporativa incluye GitLab (como sistema de control de versiones y orquestador de pipelines CI/CD) y Terraform para la definición de Infrastructure as Code. Se seguirá un modelo GitOps (gestión de infra/config mediante repositorio). Además, se usarán tecnologías de contenedores (Kubernetes) y su ecosistema para aplicaciones en la nube.

- **Plataforma de desarrollo interno (IDP)**: se aprovechará la plataforma de entregables interna de EJIE (IDP) basada de Backstage llamada BatArin, para publicar y exponer los building blocks y servicios cloud a los equipos de desarrollo. Esto permitirá consumo en modo autoservicio de los componentes validados por el CCoE, facilitando la adopción de *Golden Paths* y evitando desviaciones no controladas.

- **Monitorización y gestión**: Todos los entornos cloud y recursos serán integrados en la plataforma de monitorización y gestión de EJIE, permitiendo a EJIE visualizar alertas, métricas y controlar la calidad del servicio. Por ejemplo, los logs y métricas de las aplicaciones en AWS se remitirán a las herramientas corporativas (p.ej., Grafana, ELK stack), y las cuentas/subscripciones cloud se integrarán con las herramientas de gestión de inventario y service management de EJIE (ServiceNow).

- **Seguridad y compliance**: se usarán los servicios nativos de seguridad de AWS/GCP (ej. AWS Config, CloudTrail, Security Hub, GCP Cloud Security Command Center), u otros hiperescalares implicados según el caso, integrados con la estrategia de ciberseguridad corporativa (p.ej., volcando logs al SIEM, cumpliendo las directrices de GureSek y del Delegado de Protección de Datos). Además, se considerarán los lineamientos de ENS, la NIS2 y los marcos de compliance relevantes a la nube pública.