

## ACTA DE CIERRE DE PROYECTO

Proyecto: Modernización Serverless de Acme Inc. – Arquitecturas AWS

Células involucradas: 1 a la 9

Fecha de cierre: 27/11/2025

Responsable de documento: Jorge Sánchez Girón

Participantes del cierre:

**GusOdioso**, Inversionista del proyecto

**Christian**, Product Owner

**Carlos**, Líder Técnico

**Patricio**, Responsable de Gobernanza de Nube

### 1. Objetivo del acta

El presente documento tiene como objetivo dejar constancia formal de la conclusión del proyecto “Modernización Serverless de Acme Inc.”, desarrollado en 9 células de trabajo, así como del cumplimiento del alcance definido, la entrega de los artefactos comprometidos, los principales resultados obtenidos, las incidencias relevantes, las lecciones aprendidas y los pendientes recomendados para futuras iteraciones.

### 2. Resumen ejecutivo del proyecto

El proyecto tuvo como propósito diseñar e implementar un ecosistema de soluciones 100% serverless en AWS, aplicando buenas prácticas de arquitectura, seguridad, automatización (IaC) y CI/CD, tomando como caso ficticio a la empresa Acme Inc.

Cada célula desarrolló un módulo funcional específico (reconocimiento de imágenes, e-commerce, procesamiento paralelo, votaciones en tiempo real, dashboards de órdenes, SPA de catálogo, etc.), utilizando de forma consistente servicios como Amazon API Gateway, AWS Lambda, Amazon DynamoDB, Amazon S3, Amazon Cognito, AWS KMS, Amazon VPC y AWS CodePipeline/CodeBuild, siguiendo los lineamientos de Ciberseguridad y del Documento Maestro.

### 3. Alcance final del proyecto

De acuerdo con el Documento Maestro y los SOW de cada célula, el alcance global del proyecto abarcó:

- Definición de una arquitectura de referencia serverless multi-cuenta, con controles de seguridad centralizados.
- Implementación de 9 soluciones especializadas, una por célula, integradas lógicamente al ecosistema Acme.
- Uso sistemático de infraestructura como código (CloudFormation) en todas las cuentas objetivo.
- Implementación de pipelines CI/CD automáticos basados en GitHub, AWS CodePipeline y AWS CodeBuild.
- Generación de datos sintéticos con GenAI para pruebas funcionales en cada módulo.
- Elaboración de estimaciones de costos en AWS Pricing Calculator por célula.

Al cierre del proyecto, el alcance se considera cumplido, con ligeros ajustes operativos documentados en las secciones de incidencias y lecciones aprendidas.

### 4. Entregables finales

A la fecha del cierre, se confirman los siguientes entregables globales:

#	Entregable	Responsable / Célula	Ubicación / Referencia	Estado
1	Repositorios GitHub por célula (src/, iac/, data/, README)	Células 1–9	[URL repositorios]	Entregado
2	Plantillas de CloudFormation (infraestructura completa)	Todas	Carpeta iac/ en cada repo	Entregado
3	Código Python de Lambdas y lógica de negocio	Todas	Carpeta src/ en cada repo	Entregado
4	Datos sintéticos generados con GenAI	Todas	Carpeta data/ en cada repo	Entregado

5	Pipelines CI/CD (CodePipeline + CodeBuild)	Todas	Consola AWS - cuentas sandbox/preprod/prod	Operativo
6	Diagramas de arquitectura (draw.io / imágenes)	Todas	Anexos / carpeta de diagramas compartida	Entregado
7	Pruebas funcionales documentadas (curl y flujos end-to-end)	Todas	README y documentación de cada célula	Entregado
8	Estimación de costos por célula (AWS Pricing Calculator)	Todas	PDFs / capturas adjuntas al documento maestro	Entregado

Nota: Cada célula mantiene además su propio listado de entregables específicos (por ejemplo, reconocimiento de caricaturas, scheduling de órdenes, e-commerce, Ubertoons, SPA de vehículos, etc.), los cuales se consideran entregados y probados conforme a las especificaciones de su SOW.

## 5. Resultados y validación

- Todas las APIs se encuentran protegidas mediante Amazon Cognito (JWT) y API Gateway Authorizers.
- Las funciones Lambda operan dentro de VPC con subredes privadas, utilizando VPC Endpoints y KMS para cifrado de datos en S3, DynamoDB y Secrets/variables de entorno.
- Se ejecutaron pruebas funcionales con curl para autenticación, invocación de endpoints principales y validación de flujos end-to-end donde aplica.
- Se generaron estimaciones de costo razonables por célula, con supuestos documentados (volumen de requests, cantidad de datos, usuarios, jobs procesados, etc.).
- No se detectaron incidentes críticos que impidan el uso académico/demostrativo de las soluciones.

## 6. Incidencias y desviaciones relevantes

Durante la ejecución del proyecto se identificaron y atendieron las siguientes incidencias relevantes (ejemplos, ajustar a tu caso real):

- Ajustes en límites de tiempo y memoria de algunas funciones Lambda para soportar cargas mayores de procesamiento de imágenes.

- Corrección de permisos IAM en determinados roles para alinearse a las políticas de Ciberseguridad (principio de mínimo privilegio).
- Modificaciones menores en algunos endpoints y esquemas de DynamoDB para simplificar pruebas y demo.

Todas las incidencias fueron resueltas o mitigadas y documentadas en los repositorios correspondientes.

## 7. Lecciones aprendidas

- La definición clara de estándares de carpeta (src/, iac/, data/) y de IaC desde el inicio facilita enormemente el trabajo entre células.
- La integración temprana con Ciberseguridad (roles, KMS, VPC, endpoints) evita retrabajos en etapas avanzadas.
- Mantener un pipeline CI/CD único por célula reduce errores manuales y asegura consistencia entre sandbox, preprod y prod.
- El uso de datos sintéticos bien diseñados permite validar funcionalidades sin depender de información real.

Estas lecciones se recomiendan como base para futuros proyectos de arquitectura serverless en la organización.

## 8. Pendientes y trabajo futuro recomendado

- Profundizar en observabilidad (traces distribuidos, dashboards de métricas).
- Afinar más las estimaciones de costo con escenarios de carga adicionales.
- Extender la integración entre células (por ejemplo, usar datos reales del e-commerce en más módulos analíticos).
- Desplegar entornos adicionales (por ejemplo, ambientes de demo externos).

Estos puntos quedan como recomendaciones, no como compromisos del presente proyecto.

## 9. Cierre y aprobaciones

Con base en lo anterior, se declara que:

- Las células 1 a la 9 han concluido las actividades planificadas.
- Los entregables definidos en el Documento Maestro y en los SOW individuales han sido generados y puestos a disposición en los repositorios de GitHub y en las cuentas de AWS correspondientes.
- El sistema cumple con el alcance y los criterios de aceptación acordados para fines académicos y de demostración.

**Inversionista - GusOdioso**

Firma:

Fecha: 27/Noviembre/2025**Product Owner - Christian**

Firma:

Fecha: 27/Noviembre/2025**Líder Técnico - Carlos**

Firma:

Fecha: 27/Noviembre/2025**Gobernanza de Nube - Patricio**

Firma:

Fecha: 27/Noviembre/2025