

Exámen Integral Final de ITI-522

Computación en la Nube

Caso Empresarial: Café Aurora S.R.L.

Fecha Completa: 25/08/2025

Profesor: Andrés Joseph Jiménez

Estudiante 1: Alejandro Cordero Porras

Estudiante 2: Daniel Saborio

Repositorio público: https://github.com/JaimeCordero26/Examen_Nube.git

REGLAS

- Trabajo en parejas. Internet y herramientas de IA permitidas.
- Todo corre localmente en una VM única (Hyper-V/VirtualBox/VMware, a escoger).
- Repositorio público obligatorio y compartido con el profesor (main, CHANGELOG).

ENTORNO OBJETIVO

- **VM:** Ubuntu Server 22.04 LTS (o Debian 12, o RedHat). Si prefiere, Windows Server 2022/2025.
- **Legacy (Legado):** Apache + PHP + BD a elección (MariaDB/MySQL) vía XAMPP/LAMPP.
- **Micro-Servicios:** Docker + Kubernetes local (minikube o k3s), BD (PostgreSQL o MongoDB).
- **Proxy:** Nginx con TLS auto-firmado (HTTPS local).
- **Observabilidad:** Prometheus + cAdvisor o Grafana + Loki.

REQUERIMIENTOS FUNCIONALES

A) Arquitectura y despliegue (Kubernetes)

- 1) Cluster local operativo.
- 2) Micro-servicios (contenedores probes, limits/requests, ConfigMaps/Secrets):
 - **Catalog API:** CRUD producto (id, nombre, precio, stock, descripción).
 - **Orders API:** crear pedido con lista de items (id, id_catalog, orden, cálculo simple de total).
 - **Customers API:** CRUD cliente (id, nombre, email, número de identidad).
 - **Persistencia:** PostgreSQL o MongoDB (a elección para micro-servicios).
- 3) **Service + Ingress:** exponiendo /api/catalog, /api/orders, /api/customers.
- 4) **Seed reproducible:** ≥ 20 productos y ≥ 5 clientes (inserciones de bases de datos).

B) Módulo Legado (XAMPP/LAMPP)

- 1) **Inventario Legado:** Apache + PHP + BD elegida (MariaDB/MySQL).
- 2) **Endpoint** /legacy/inventory (*solo lectura*) para comparar existencias con Catalog.

C) Integración y reverse proxy (HTTPS local)

- 1) Nginx frontal con TLS autofirmado: HTTPS; redirección HTTP \rightarrow HTTPS.
- 2) Rutas:
 - <https://localhost/api/> \rightarrow Ingress del cluster.
 - <https://localhost/legacy/> \rightarrow Apache (legado).
- 3) Compresión (gzip/brotli) y cache de estáticos.

D) Seguridad y privacidad

- 1) Política de Clasificación de Datos (≥ 3 niveles) + matriz mínima de controles.
- 2) Threat model (STRIDE) breve + hardening básico (UFW, permisos, rotación de secretos).
- 3) Cifrado de identidad (clientes):
 - **Dato almacenado cifrado:** visualización en claro en UI/API.
 - **Evidencia:** SELECT + comandos/funciones (SQL o app) y referencia a Secret.
 - **MariaDB/MySQL:** AES_ENCRYPT/AES_DECRYPT (o cifrado en aplicación). **MongoDB:** CSFLE o cifrado en aplicación.

E) SLAs internos y decisión arquitectura

- 1) *SLA interno*: ≥3 métricas (disponibilidad %, p95 latencia, MTTR) con SLO y presupuesto de error.
- 2) Comparativa on-prem vs IaaS vs PaaS vs híbrido (tabla de supuestos y costos).
- 3) Conclusión técnica-económica para MiPyME.

F) Observabilidad, pruebas y respaldo

- 1) Prometheus + cAdvisor activos; Grafana con 1 dashboard mínimo (CPU/Mem por pod, errores HTTP).
- 2) Loki ingestando logs de al menos 1 microservicio; consulta por etiqueta en Grafana.
- 3) *Pruebas de carga*: 2 perfiles (ab/hey) con reporte de p95 y tasa de errores.
- 4) Backup y restore reproducibles de la BD elegida.

G) Frontend mínimo

- 1) Página única (HTML/JS) o panel sencillo con:
 - Listado de productos y creación de pedido.
 - Listado/creación de clientes (visualización de identidad en claro).
 - Consulta a /legacy/inventory.

H) Documentación de Proyecto

- 1) Diagrama lógico + despliegue.
- 2) RunBook de operación (arranque, health-checks, backup/restore, usuarios/roles).
- 3) Decisiones de diseño y supuestos.
- 4) Bitácora de la sesión (hitos, commits, tags y evidencias).

I) Documentación Técnica (*formato APA y, con portada y tabla de contenidos y bibliografía*)

- 1) *Matriz de cumplimiento y privacidad*: Mapa breve que relacione cada dato tratado con leyes/estándares aplicables y controles propuestos (nacional/internacional) + riesgos residuales. Que relacione las “normativas vigentes” y “cumplimiento de leyes y estándares”, con el proyecto creado para el examen.

2) *Catálogo de métricas y alertas*: Definición formal de métricas (p95 latencia, disponibilidad, error budget, MTTR), umbrales y alertas en Grafana/Loki, más la consulta exacta usada en el exámen. La idea es co-relacionar y definir los “indicadores para medir rendimiento”, el “establecimiento de SLAs”, y los SLOs que ya se solicitan, para que aquí se documente cómo se calculan y monitorean.

3) *Plan de continuidad (BCP/DRP) con RPO/RTO*: Documento corto con supuestos, RPO/RTO, pasos de conmutación/retorno y evidencias (capturas de restore + validación funcional). Alinear con la sección de “asegurar la continua disponibilidad de sistemas y datos”.

4) *Hoja de ruta de migración del legado*: Documento que compare inventario /legacy con catálogo del microservicio, plan de corte, verificación de consistencia y reversa. Enlaza “independencia de modelos” y “uso de aplicaciones en nube vs en sitio”.

CRITERIOS DE ACEPTACIÓN (VERIFICABLES)

- K8s operativo sin CrashLoopBackOff; healthz responde {'status':'ok'}.
- HTTPS operativo (Nginx con certificado autofirmado; redirección HTTP→HTTPS).
- Legacy /legacy/inventory responde JSON válido.
- **Cifrado de identidad**: UI/API en claro; SELECT directo cifrado; comandos/funciones y Secret entregados.
- **Observabilidad**: Prometheus+cAdvisor y Grafana con dashboard; Loki con consulta por etiqueta.
- Backups y restore verificables.
- Repositorio público con ramas, tags y CHANGELOG.md.
- Comprimido .zip con código fuente, y documentación creada.

ENTREGABLES

- Imagen de VM (VHDX/OVA/VMDK) + README de importación y arranque.
- Carpeta source/ (repos y código) y deploy/ (YAML K8s, nginx.conf, scripts).
- Carpeta docs/ (Arquitectura, Runbook, Política de Datos, SLA, Threat model, Bitácora, Documentación Técnica).
- Carpeta evidence/ (*capturas y salidas*: kubect!, curl, ab/hey, panel Grafana, consulta Loki, SELECT cifrado).
- Repositorio público: URL GitHub/GitLab/BitBucket, ramas main, tags por sección, CHANGELOG.md.

RÚBRICA DE EVALUACIÓN (100pts)

Criterio	Descripción	Puntos
Infraestructura K8s	Cluster operativo; probes; limits/requests; ConfigMaps/Secrets; Ingress	18
Microservicios (3 APIs)	Catalog/Orders/Customers; persistencia; seeds consumidos	16
Cifrado de identidad	Almacenado cifrado; UI en claro; evidencia SELECT + comandos + Secret	14
Módulo Legado	Apache+PHP+BD; /legacy/inventory funcional	10
Reverse proxy + HTTPS	Nginx con TLS autofirmado; rutas /api/* y /legacy/*; compresión/cache	10
Observabilidad	Prometheus+cAdvisor; Grafana dashboard; Loki consultas por etiqueta	12
Pruebas & Respaldo	2 perfiles carga; backup/restore reproducibles	8
SLAs & Comparativa	SLOs y presupuesto de error; tabla y conclusión	6
Documentación & Reproducibilidad	Diagrama, runbook, bitácora, scripts básicos, documentación técnica	6
TOTAL		100

CHECKLIST POR SECCIÓN

Sección 1 — Infraestructura

- ☐ VM lista
- ☐ K8s operativo
- ☐ BD contenedor lista
- ☐ Nginx instalado
- ☐ Certificado autofirmado configurado
- ☐ Repo público y permisos

Sección 2 — Datos

- ☐ Esquema/colecciones
- ☐ Seeds (≥ 30 productos, ≥ 15 clientes)

- ☐ Secrets y política de claves
- ☐ Rotación simulada documentada

Sección 3 — Servicios

- ☐ Catalog/Orders/Customers con probes
- ☐ Persistencia funcional
- ☐ Service/Ingress /api/*
- ☐ Healthz OK

Sección 4 — Integración y seguridad

- ☐ Legacy /legacy/inventory
- ☐ Clasificación de datos + STRIDE + hardening

Sección 5 — Observabilidad y calidad

- ☐ Prometheus+cAdvisor
- ☐ Grafana dashboard
- ☐ Loki con consulta en dashboard de Grafana
- ☐ Pruebas de carga y reporte
- ☐ Backup/restore verificados

Sección 6 — Documentación y entrega

- ☐ Diagrama y RunBook
- ☐ SLA/comparativa
- ☐ Evidencias reunidas
- ☐ CHANGELOG actualizado
- ☐ Documentación Técnica

Nota: El equipo debe registrar su tiempo real en la Bitácora de la sesión.