



SISTEMAS OPERATIVOS (1ASI0726)

EXAMEN PARCIAL

2025-10

NRC: 4198

Profesor: Eduardo Vásquez

Duración: 120 minutos

Indicaciones:

1. El examen consta de dos (4) preguntas sobre casos de estudio.
2. El enunciado de la pregunta se encuentra en el archivo 1ASI0726_EA-2025-10_4198.docx
3. La solución deberá adjuntarse únicamente en el aula, en un documento PDF con la siguiente nomenclatura:
1ASI0726_EA-2025-10_4198_Codigo_Apellido.pdf
4. Los profesores en mención solo recibirán correos provenientes de las cuentas UPC, de ninguna manera se recibirán correos de cuentas públicas.
5. Ante problemas técnicos, debe de forma obligatoria adjuntar evidencias de este, como capturas de pantalla, videos, fotos, etc. Siendo requisito fundamental que, en cada evidencia se pueda apreciar claramente la fecha y hora del sistema operativo del computador donde el alumno está rindiendo el examen.
6. Los problemas técnicos se recibirán como máximo 15 minutos culminado el examen.

Caso 1: Pequeña Empresa - Hosting Local vs. Migración a la Nube

Empresa: Startup de venta de ropa online con 10 empleados.

Situación: Actualmente tienen un servidor local con Linux (Ubuntu Server) que aloja su página web desarrollada en WordPress y una base de datos MySQL que se actualiza todos los días y cuya administración está dada por un Community Manager.

Problemas identificados:

El servidor sufre caídas frecuentes sobre todo cuando hay alto tráfico (ej. en fechas donde se publican rebajas de los productos).

Ni el servidor ni los servicios no tienen redundancia ni backups automatizados.

Todos los procesos se hacen manualmente, pero sin manejar un plan de ejecución.

Evaluar:

1. ¿Mantener el hosting local con mejor hardware (más RAM, SSD, balanceador)?
2. ¿Migrar a un servicio cloud (AWS Lightsail, Google Cloud) con autoescalado?
3. ¿Qué sistema operativo (Windows Server vs. Linux) sería más eficiente on premise o en la nube de acuerdo con su evaluación de ubicación?
4. Diseñar una arquitectura basada en hardware y software, completando la Figura 1. Por capas, detalle los sustentos y/o argumentos del porque la elección del software y para hardware realizar el dimensionamiento tomando como referencia lo descrito en la figura 1.

Considerar en sus respuestas los siguientes puntos:

- Capacidad de respuesta (latencia, throughput).
- Tolerancia a fallos (redundancia, backups).
- Escalabilidad (horizontal vs. vertical).
- Seguridad (SO hardened, firewalls, cifrado).
- Costo-beneficio (on-premise vs. cloud).

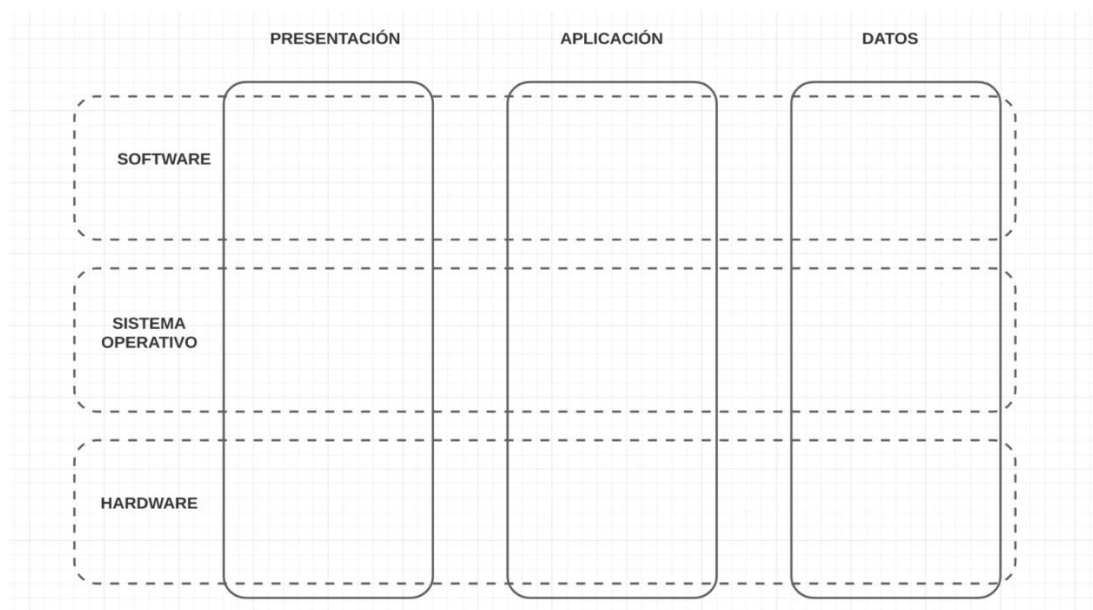
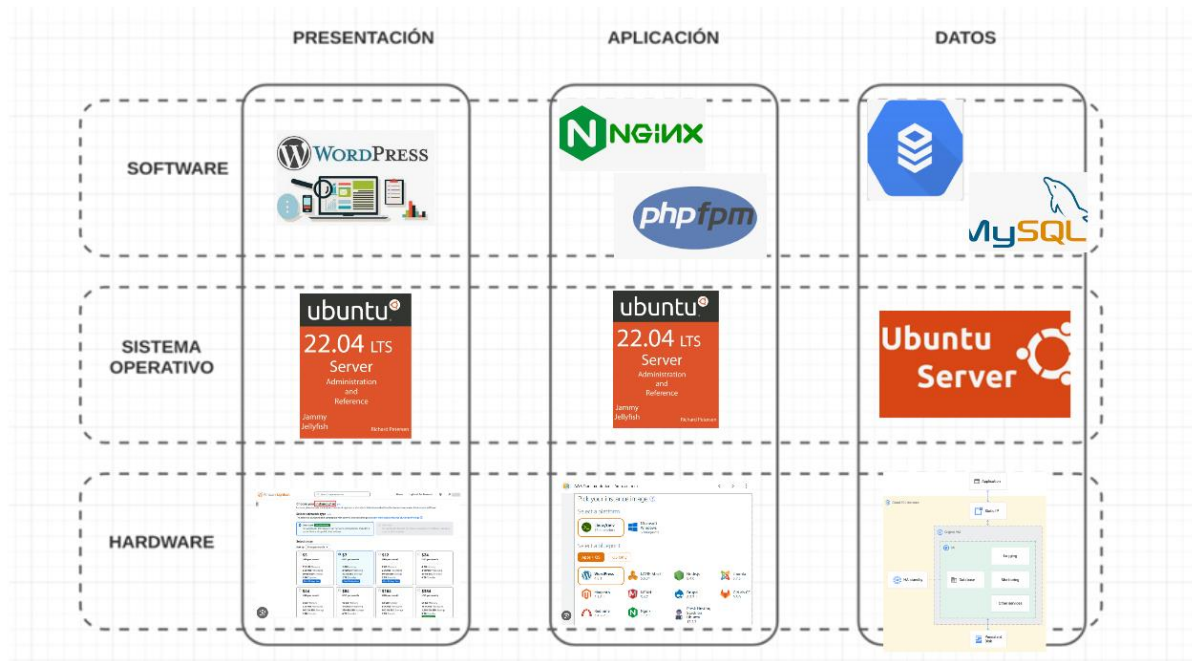


Figura 1: Diagrama de solución basada en 3 capas



Especificaciones:

Capa de Presentación:

Software: Se usará WordPress como software principal, dado que la startup ya lo emplea y es uno de los CMS más utilizados en el mundo por su facilidad de uso. También, permite a Community Manager gestionar el contenido del sitio, subir productos, activar promociones y realizar cambios sin conocimientos técnicos avanzados. WordPress ofrece un gran plan de comunidad de soporte y muchas herramientas de optimización de rendimiento, SEO y seguridad.

Sistema Operativo: Se usará Ubuntu Server 22.04 LTS, dado que es una versión de soporte extendido (LTS) asegurando actualizaciones de seguridad durante al menos 5 años. Ubuntu tiene una excelente compatibilidad con WordPress, Nginx y PHP, permitiendo automatizar tareas de administración del servidor con propias herramientas de uso. Además, es ligero. Es bajo en consumo de recursos, ideal para instancias pequeñas (1–2 GB RAM).

Hardware: Se usará una instancia cloud de AWS Lightsail con 1 vCPU, 2 GB de RAM y disco SSD de 80 GB. Al configurar de esta forma es suficiente para alojar una capa, en específico la capa de presentación, especialmente si combinas una red CDN con Cloudflare. Esto reduce la latencia, mejora la experiencia del cliente durante campañas de alto tráfico y protege contra ataques como DDoS.

Capa de Aplicación:

Software: Se propone usar una combinación de Nginx con PHP-FPM. Nginx es un servidor web altamente eficiente y capaz de manejar muchas conexiones simultáneas con bajo consumo de recursos, lo cual es ideal para startups que pueden enfrentar picos de tráfico en momentos

puntuales, como durante campañas de rebajas. PHP-FPM se usará para mejorar el rendimiento, dar con la lógica del servidor web y ejecutar de forma eficiente el código PHP.

Sistema Operativo: Se usará Ubuntu Server 22.04 LTS, hablado anteriormente, esta uniformidad facilita la administración del entorno completo y reduce la complejidad operativa. También. Utilizará herramientas como Fail2Ban, UFW y ClamAV para mejorar la seguridad del entorno de la aplicación.

Hardware: Se usará una instancia de AWS Lightsail con 2 vCPU y 4 GB de RAM, no solo ofrece un buen equilibrio entre costo y rendimiento, sino permite manejar la lógica del sitio y procesos dinámicos de usuarios concurrentes. Llevando a cabo la preparación de futuras ampliaciones mediante escalabilidad vertical u horizontal dependiendo de la empresa, si desea usar servicios más avanzados o no.

Capa de Datos:

Software: Se recomienda usar una base de datos MySQL gestionada en la nube, ya sea mediante Google Cloud SQL o Amazon RDS. Por que ofrece mucha disponibilidad, escalabilidad automática y respaldo diario automatizado con opciones de recuperación ante desastres. Permitiendo tolerancia a fallos y continuidad operativa incluso si una región presenta problemas.

Sistema Operativo: Se ejecuta generalmente las ditribuciones Linux como Ubuntu Server o Debian, ya que están configuradas y optimizadas para cargas de trabajo con MySQL. Asegura un entorno estable y seguro, sin la necesidad de que el equipo técnico se encargue del mantenimiento de bajo nivel.

Hardware: Se recomienda usar máquinas con almacenamiento SSD y capacidad de IOPS ajustable según la carga para las bases de datos gestionadas. Esto nos permitirá mantener un rendimiento óptimo en los momentos de mayor tráfico sin necesidad de sobredimensionar el sistema en tiempos normales.

Caso 2: Mediana Empresa - Virtualización para Aumentar Disponibilidad

Empresa: Cadena de restaurantes con sistema de pedidos online (500 transacciones/hora).

Situación: Tienen dos servidores físicos con Windows Server 2019 ejecutando:

- Servicio web (Node.js).
- Base de datos (SQL Server).
- Sistema de facturación (aplicación legacy en .NET).

Problemas identificados:

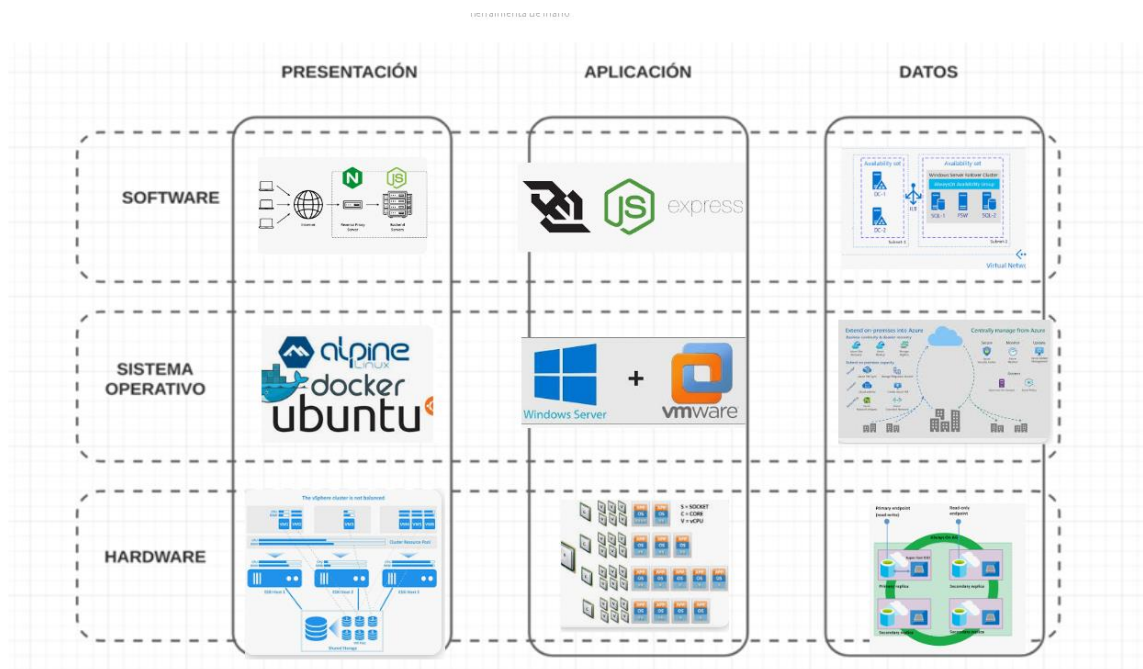
- Si un servidor falla, el servicio se cae.
- Quieren mejorar disponibilidad sin gastar en hardware adicional.

Evaluar:

1. ¿Implementar virtualización (Hyper-V o VMware ESXi) para alta disponibilidad?
2. ¿Mover solo la base de datos a un PaaS (Azure SQL) y dejar el resto on-premise?
3. ¿Conviene usar contenedores (Docker) para el backend en lugar de máquinas virtuales?

Considerar en sus respuestas los siguientes puntos:

- Capacidad de respuesta (latencia, throughput).
- Tolerancia a fallos (redundancia, backups).
- Escalabilidad (horizontal vs. vertical).
- Seguridad (SO hardened, firewalls, cifrado).
- Costo-beneficio (on-premise vs. cloud).



Especificaciones:

Capa de Presentación:

Software: Se usará Nginx como proxy reverso permite manejar múltiples conexiones simultáneas con bajo consumo de recursos. Ya que reduce el tiempo de arranque y se facilita el mantenimiento. Nginx es ideal para una empresa con alta carga como una cadena de restaurantes con pedidos en tiempo real.

Sistema Operativo: Se utilizará Alpine o Ubuntu Server, consumen pocos recursos y tiene buena seguridad. También se integran bien con pipelines de CI/CD. Este enfoque también permite mayor portabilidad entre entornos de prueba y producción.

Hardware: Se usará una VM de 2 vCPU y 4 GB de RAM sobre un clúster de Hyper-V o ESXi. Esta configuración es económica y suficiente, ya que el contenido estático y la interfaz se sirven rápidamente con ayuda de cache y compresión desde Nginx.

Capa de Aplicación:

Software: Se usará Node.js es ideal para manejar múltiples solicitudes simultáneas con bajo consumo, mientras que la aplicación legacy .NET se mantiene activa sin necesidad de reescritura. Además, cada servicio se puede administrar, escalar o actualizar sin afectar al otro.

Sistema Operativo: Se usará Windows Server 2019, ideal para la compatibilidad con aplicaciones .NET y SQL Server. Usar la versión Core reduce la superficie de ataque, el uso de recursos y mejora la seguridad.

Hardware: Se usará VMs de 4 vCPU y 8 GB de RAM, con snapshots y failover habilitado en el clúster ESXi/Hyper-V. Esto permite que las VMs migren automáticamente a otro, evitando caídas del sistema

Capa de Datos:

Software: Se usará Azure SQL ya que elimina directamente la gestión desde el sistema. Además tiene una escalabilidad automática y backups diarios sin intervención manual. se puede usar SQL Server en modo Always On, que replica datos entre dos nodos para alta disponibilidad.

Sistema Operativo: Se usará Azure SQL, asegura actualizaciones automáticas y parches de seguridad. En la opción on-premise, se debe usar Windows Server con configuración optimizada para SQL, siguiendo buenas prácticas de indexación, particionamiento y cifrado de datos.

Hardware: Se usará Azure SQL en un plan S2 (2 vCore, Gen5), ya que ofrece alto rendimiento con almacenamiento SSD y réplicas geo-redundantes. También, requiere menos inversión que adquirir un nuevo servidor físico, pero aún proporciona tolerancia a fallos y alta disponibilidad.

Caso 3: Gran Empresa - Migración a Cloud Híbrido con Balance de Carga

Empresa: Banco regional con aplicación móvil (100,000 usuarios activos).

Situación: Actualmente usan:

- 10 servidores físicos con Red Hat Enterprise Linux (RHEL) para el backend.
- 3 servidores Windows para autenticación (Active Directory).
- Almacenamiento NAS para registros transaccionales.

Problema:

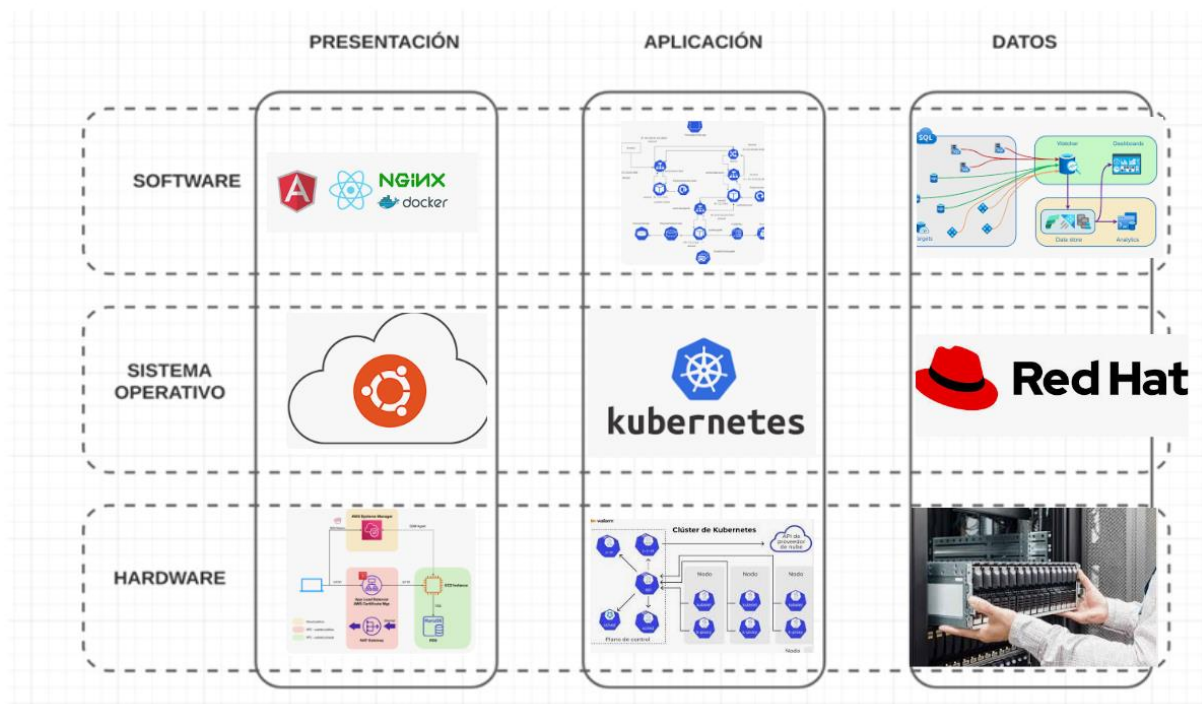
- Se generan picos de tráfico que saturan los servidores con mucha frecuencia.
- Necesitan cumplir regulaciones de seguridad (ISO 27001).

Evaluar:

1. ¿Migrar a un modelo híbrido (nube pública AWS/Azure para frontend y privada para datos sensibles)?
2. ¿Implementar Kubernetes (K8s) para orquestar contenedores y autoescalar?
3. ¿Qué sistema operativo (RHEL vs. Ubuntu Server) es más estable para Kubernetes en la nube?

Considerar en sus respuestas los siguientes puntos:

- Capacidad de respuesta (latencia, throughput).
- Tolerancia a fallos (redundancia, backups).
- Escalabilidad (horizontal vs. vertical).
- Seguridad (SO hardened, firewalls, cifrado).
- Costo-beneficio (on-premise vs. cloud).



Especificaciones:

Capa de Presentación:

Software: Se usará API Gateway (como AWS API Gateway o Azure Front Door) junto con Nginx permite controlar el tráfico, aplicar políticas de seguridad, y proteger los endpoints backend.

Sistema Operativo: Se usará Ubuntu ya que es liviano y compatible con servicios cloud. Permitiendo despliegues continuos y configuraciones automatizadas sin licencia adicional, ya que su integración con herramientas de DevOps y Kubernetes es superior.

Hardware: Se usará instancias cloud (EC2, App Services) de tipo medio (2–4 vCPU), tiene balances globales y distribuye bien tráfico automáticamente entre regiones. Garantizando una alta disponibilidad y una experiencia fluida para el usuario.

Capa de Aplicación:

Software: Se usará Kubernetes, ya que permite escalar cada componente importante según la necesidad ya sea login, pagos o reportes. Por otro lado, los pods se reinician al fallar y se distribuyen en múltiples nodos, garantizando alta disponibilidad y resiliencia.

Sistema Operativo: Se usará Ubuntu Server ya que funciona muy bien como sistema base para clústeres Kubernetes en nube pública. Soportando autoescalado e integración con herramientas como Helm y Prometheus.

Hardware: Se usará clúster K8s, ya que permite aumentar o reducir nodos según la demanda. Así, el banco puede soportar picos sin sobredimensionar infraestructura todo el tiempo.

Capa de Datos:

Software: Usaremos SQL Server con Always On, y Active Directory garantiza identidad centralizada. Para que los registros de transacciones se almacenen en NAS local o en servicios Azure Blob Storage.

Sistema Operativo: Usaremos RHEL, cumpliendo los estándares ISO 27001 y SELinux. Util para los bancos ya que tiene estabilidad y un buen soporte que le convierte en el mejor sistema operativo para datos financieros críticos.

Hardware: Usaremos servidores físicos para los datos sensibles, teniendo conexión cifrada con la nube pública. Además, el almacenamiento puede combinar NAS local y Blob Storage en nube para backups y recuperación ante desastres.

Caso 4: Empresa en Crecimiento - Optimización de Costos en la Nube

Empresa: Plataforma de streaming (similar a Netflix) con 50,000 usuarios accediendo de manera simultánea.

Situación:

- Usan Google Cloud con máquinas virtuales (Debian) para streaming.
- El almacenamiento se encuentra en Cloud Storage.
- Se tiene una base de datos en Firebase.

Problema:

- Los costos de computación son altos por uso ineficiente de recursos.
- Algunas instancias están subutilizadas.

Evaluar:

1. ¿Reemplazar VMs por servicios serverless (Cloud Functions) para procesamiento de video?
2. ¿Usar un SO optimizado para multimedia (como Alpine Linux en contenedores)?
3. ¿Implementar un sistema de caché distribuido (Redis) para reducir cargas a la BD?

Considerar en sus respuestas los siguientes puntos:

- Capacidad de respuesta (latencia, throughput).
- Tolerancia a fallos (redundancia, backups).
- Escalabilidad (horizontal vs. vertical).
- Seguridad (SO hardened, firewalls, cifrado).
- Costo-beneficio (on-premise vs. cloud).



Especificaciones:

Capa de Presentación:

Software: Usaremos React Native y web responsive, ya que nos permite llegar a más usuarios con una sola base de código. Además, tenemos una entrega de contenido estático que se realiza a través de un CDN de Google, lo que garantiza baja latencia global y mejor experiencia para los 50,000 usuarios concurrentes.

Sistema Operativo: Se usará Cloud Functions o Cloud Run, son contenedores ligeros que pueden usar Alpine Linux, donde los tiempos de inicio son más rápidos y de menor consumo.

Hardware: Se usará un balanceador de carga global, que distribuye peticiones entre regiones y minimiza los tiempos de espera. Ya que, se mejora la disponibilidad y la capacidad de respuesta del sistema, incluso durante picos de tráfico.

Capa de Aplicación:

Software: Se usará FFmpeg, dentro de contenedores orquestados por Cloud Run o Kubernetes permite un balance entre costo y rendimiento. Donde Node.js maneja APIs y controladores que procesan tareas livianas mediante funciones serverless que escalan automáticamente.

Sistema Operativo: Se usará Alpine Linux, más que nada por su eficiencia en contenedores pequeños. Pero se requiere estabilidad y compatibilidad para tareas multimedia pesadas. Donde también se puede mantener a Debian en VMs de uso específico.

Hardware: Se usará Cloud Run y Kubernetes Engine, lo que nos permite tener un autoescalado en función de tráfico. Además, se mantiene algunas VMs que son optimizadas, como: 8 vCPU, GPU si se usa encoding. Donde se puede lograr una eficiencia sin sacrificar rendimiento.

Capa de Datos:

Software: Se integrará Redis, su uso nos ayudará para chequear resultados frecuentes. Como listas de reproducción o configuraciones. Además se integrará una herramienta Cloud Storage donde este almacena archivos de video de una forma distribuida y redundante.

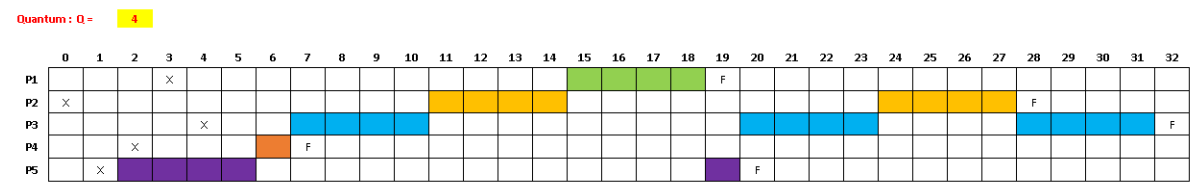
Sistema Operativo: Se usará Firebase y Cloud Storage, ya que son servicios gestionados donde el sistema operativo está encapsulado. Nos ayuda a reducir las preocupaciones operativas donde se mejora la seguridad y garantiza alta disponibilidad y compliance sin configuración adicional.

Hardware: Se usará Cloud Storage, ya que tiene una garantía de durabilidad del 99.999999999% y bajo costo para archivos estáticos como los videos. Para acceso ultra rápido, Redis en Memorystore permite mantener en RAM la información más consultada mejorando su rendimiento general evitando cuello de botella.

Caso 5.Gestión por procesos– Incrustar las capturas de pantalla de la plantilla usadas en el presente documento.Para gestionar procesos del computador aplicar el algoritmo de Round Robin, con un quantum=4 para la siguiente tabla de procesos:

Process	Arrive Time	Burst Time	(WT) Waiting Time	(CT) Complete Time
P1	3 ns	4 ns		
P2	0 ns	8 ns		
P3	4 ns	12 ns		
P4	2ns	1ns		
P5	1ns	5ns		

- Realizar el diagrama de Gannt.



- Calcular el AWT y ACT.

	Tiempo Llegada		Duración			
Process	Arrive Time	Burst Time	(WT) Waiting Time		(CT) Complete Time	
P1	3 ns	4 ns	19 menos 7	12	19 menos 3	16
P2	0 ns	8 ns	28 menos 8	20	28 menos 0	28
P3	4 ns	12 ns	32 menos 16	16	32 menos 4	28
P4	2 ns	1 ns	7 menos 3	4	7 menos 2	5
P5	1 ns	5 ns	20 menos 6	14	20 menos 1	19
			AWT	13.2	ACT	19.20

Rúbrica de preguntas 1 a 4

Criterio de calificación	Excelente	Promedio	Deficiente
C01. Software	Define y sustenta el uso de software para las capas que corresponden	Define y sustenta parcialmente el uso de software para las capas que corresponden	No define ni sustenta el uso de software para las capas que corresponden
	1.0 puntos	0.5 punto	0 puntos
C02. Sistema Operativo	Define y sustenta el uso de Sistema Operativo para las capas que corresponden	Define y sustenta parcialmente el uso de Sistema Operativo para las capas que corresponden	No define ni sustenta el uso de Sistema Operativo para las capas que corresponden
	1.0 puntos	0.5 puntos	0 puntos
C03. Hardware	Define y sustenta el uso de hardware para las capas que corresponden	Define y sustenta parcialmente el uso de hardware para las capas que corresponden.	No define ni sustenta el uso de hardware para las capas que corresponden
	1.0 puntos	0.5 punto	0 puntos
C04. Herramientas o componentes basadas en hardware o software para la Integración, seguridad de la información de la solución	Define y sustenta el uso de herramientas o componentes basados en hardware o software para las capas que corresponden	Define y sustenta parcialmente el uso de herramientas o componentes basados en hardware o software para las capas que corresponden	No define ni sustenta el uso de herramientas o componentes basados en hardware o software para las capas que corresponden
	1.0 puntos	0.5 punto	0 puntos
Total	4 puntos	2.5 puntos	0 puntos

Rúbrica de la pregunta 5

Criterio de calificación	Excelente	Promedio	Deficiente
C01. Diagrama de Gantt	Lista correctamente el orden de los procesos	Lista incorrectamente el orden de los procesos	No lista el orden de los procesos
	2.0 puntos	1 puntos	0 puntos
C02. Cálculo de AWT y ACT	Calcula correctamente el AWT y ACT	Calcula incorrectamente el AWT y ACT	No calcula el AWT y ACT
	2.0 puntos	1.0 puntos	0 puntos
Total	4.0 puntos	2.0 puntos	0 puntos