

EVALUACIÓN	Obligatorio	GRUPO	75170 N6A	FECHA	21/10/2025
MATERIA	Certificado en DevOps				
CARRERA	Analista en Tecnologías de la Información Actualización profesional				
CONDICIONES	<ul style="list-style-type: none"> - Puntaje máximo: 55 puntos - Puntaje mínimo: 1 punto - Fecha de entrega: 01/12/2025 hasta las 21:00 horas en gestion.ort.edu.uy (max. 40Mb en formato zip, rar o pdf) 				
<p>Uso de material de apoyo y/o consulta</p> <p><u>Inteligencia Artificial Generativa</u></p> <ul style="list-style-type: none"> - Seguir las pautas de los docentes: Se deben seguir las instrucciones específicas de los docentes sobre cómo utilizar la IA en cada curso. - Citar correctamente las fuentes y usos de IA: Siempre que se utilice una herramienta de IA para generar contenido, se debe citar adecuadamente la fuente y la forma en que se utilizó. - Verificar el contenido generado por la IA: No todo el contenido generado por la IA es correcto o preciso. Es esencial que los estudiantes verifiquen la información antes de usarla. - Ser responsables con el uso de la IA: Conocer los riesgos y desafíos, como la creación de “alucinaciones”, los peligros para la privacidad, las cuestiones de propiedad intelectual, los sesgos inherentes y la producción de contenido falso - En caso de existir dudas sobre la autoría, plagio o uso no atribuido de IAG, el docente tendrá la opción de convocar al equipo de obligatorio a una defensa específica e individual sobre el tema <p>IMPORTANTE:</p> <ol style="list-style-type: none"> 1) Inscribirse 2) Formar grupos de hasta 3 personas del mismo dictado 3) Subir el trabajo a Gestión antes de la hora indicada (ver hoja al final del documento: "RECORDATORIO") <p>Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails alamon@ort.edu.uy y rodriguez_mb@ort.edu.uy, o telefónicamente al 29021505 - int 1156 u 1138</p>					

1. Presentación del problema

Una empresa de comercio electrónico en rápido crecimiento ha adoptado *StockWiz* como base para su plataforma de ventas. A pesar de contar con una arquitectura moderna de microservicios, la empresa enfrenta desafíos críticos que amenazan su capacidad de escalar y mantener la calidad del servicio durante períodos de alto tráfico como Black Friday, Cyber Monday y temporadas festivas.

El análisis de incidentes recientes revela problemas sistémicos profundos: durante el último evento de ventas, la aplicación experimentó múltiples caídas que resultaron en pérdida de transacciones, carritos de compra abandonados y una experiencia de usuario deteriorada. Las investigaciones posteriores identificaron despliegues manuales inconsistentes, falta de pruebas automatizadas de integración entre microservicios, configuraciones diferentes entre ambientes, y ausencia de monitoreo proactivo que permitiera detectar problemas antes de impactar a los clientes.

La raíz del problema trasciende lo técnico: existe una brecha cultural y organizacional entre los equipos. Los desarrolladores crean nuevas funcionalidades para microservicios sin comprender completamente sus interdependencias y requisitos operacionales. Por su parte, el equipo de operaciones gestiona la infraestructura de manera reactiva, con procesos manuales que no escalan con la velocidad de desarrollo requerida por el negocio.

La dirección ejecutiva ha reconocido que el crecimiento sostenible del negocio depende de resolver estos problemas fundamentales. Se ha aprobado una iniciativa estratégica para transformar la organización mediante la implementación de prácticas DevOps modernas, asegurando que *StockWiz* pueda desplegarse de manera confiable, escalar elásticamente según demanda, y mantener los más altos estándares de calidad y seguridad.

2. Entrega

Se detallan los siguientes aspectos a considerar:

- a. Planificación y Seguimiento
 - i. Utilizar un *Kanban* o *Scrum* para realizar la planificación y seguimiento de las tareas identificadas a realizar por el equipo.
 - ii. Documentar (screenshots) el flujo de trabajo y estado de tareas en al menos 3 momentos distintos del proyecto.
- b. Integración y Entrega Continua (CI/CD)
 - i. Desarrollar pipelines de CI/CD para al menos tres ambientes: Dev, Test y Prod.
 - ii. Cada pipeline debe contener como mínimo tres etapas.
 - iii. Implementar *quality gates* entre ambientes.
- c. Containerización y Despliegue
 - i. Optimizar la creación de los containers utilizando buenas prácticas.
 - ii. Desplegar dichos containers en un servicio de orquestación en AWS (ECS o EKS) u otra nube equivalente.
- d. Testing y Calidad
 - i. Implementar uno de los siguientes tipos de testing:
 - 1. Pruebas funcionales (Postman, Selenium, etc.)
 - 2. Pruebas de carga/rendimiento (JMeter, k6, etc.)
 - ii. Integrar alguna herramienta de análisis de código estático (SonarQube, ESLint, o similares) en el pipeline.
 - iii. Elaborar un informe con los resultados obtenidos y recomendaciones de mejora con hallazgos significativos.
- e. Control de Versiones
 - i. Almacenar todo el código en repositorios *Git*.
 - ii. Implementar una estrategia de ramificación (Git Flow, trunk based, feature branch, etc.) claramente definida. Justificar la elección.
 - iii. Demostrar colaboración efectiva mediante PRs con revisiones de código entre miembros del equipo.
- f. Infraestructura como Código (IaC)
 - i. Implementar toda la infraestructura como código usando Terraform.
 - ii. El código de IaC debe incluir:
 - 1. Modularización adecuada
 - 2. Variables parametrizadas por ambiente
 - 3. Outputs relevantes documentados
 - 4. Manejo seguro de secretos
 - iii. Servicios Serverless
 - 1. Implementar al menos uno de los siguientes servicios serverless:
 - a. AWS Lambda para automatizaciones y/o procesamiento de eventos (por ejemplo: alertas, backups, análisis de logs, etc.).
 - b. API Gateway para gestión de acceso y seguridad a APIs
 - 2. El servicio debe integrarse con el resto de la arquitectura y tener un propósito claramente definido (a elección).

-
- iv. Observabilidad
 - 1. Implementar una solución básica de monitoreo y alerta.
 - 2. Configurar *al menos* 1 dashboard para métricas críticas.
 - 3. Establecer *al menos* 2 alertas para condiciones relevantes.
 - g. Documentación
 - i. Crear README.md completo con instrucciones de despliegue y configuración.
 - ii. Generar diagramas de arquitectura, flujos CI/CD y estrategia de Git.
 - iii. Documentar decisiones de diseño importantes y lecciones aprendidas.

Algunas consideraciones / aclaraciones

- 1. Las aplicaciones a desplegar se encuentran en el siguiente repositorio:
 - a. [StockWiz](#)
- 2. Queda a consideración del equipo la elección de las herramientas y/o servicios a utilizar para desplegar la solución propuesta.
- 3. Se deberá exponer el trabajo realizado en una instancia de defensa. Para esto, recomendamos usar una presentación de venta a modo de guía. La presentación de venta consta en la venta de la solución/producto, **por ende, no es una explicación técnica.**
- 4. La documentación entregada será valorada en prolíjidad, calidad y organización. Queda en total libertad la definición del formato de la misma.
- 5. El equipo deberá entregar todo el contenido en el formato establecido por el sistema de gestión (.rar o .zip). Este archivo contendrá todo lo utilizado: repositorio git, documentación, imágenes.
- 6. Registrar equipo de obligatorio:  [Listado de Alumnos DevOps - 2025 \(Agosto\)](#)
- 7. Se podrán realizar TODAS las consultas que quieran, mientras estas sean CLARAS, CONCISAS y tengan un mínimo de investigación previa. **No se podrán enviar documentos para ser corregidos o medir el avance fuera de las fechas de entrega previstas.**

3. Defensa

Como defensa deberán realizar una presentación del proyecto realizado mediante Teams / Zoom en una sesión grupal de no más de 20 minutos. Es necesario tener la cámara encendida y el micrófono abierto. Podremos llamar de forma individual, solo si es necesario, a un alumno para complementar lo expuesto en la defensa grupal.

4. Rúbrica

SE PIDE	PUNTAJE MÁXIMO	SATISFACTORIO
Implementación en base a los puntos detallados en la parte de entrega	7	La solución resuelve el problema propuesto de forma completa, es innovadora y demuestra entendimiento de DevOps.
Gestión de proyecto	5	Se evidencia uso consistente de la metodología seleccionada con un progreso bien documentado.
Documentación técnica	5	Documentación completa, bien estructurada y técnicamente precisa en formato Markdown.
Containerización y despliegue	5	Containers correctamente definidos y desplegados en servicio de orquestación en la nube.
Gestión de código	5	Repositorio/s bien organizado/s con separación adecuada de componentes.
Estrategias de ramificación	4	Evidencia clara de trabajo colaborativo con un uso correcto de las estrategias de ramificación y pull requests.
Implementación serverless	4	Servicio serverless implementado con propósito claro y bien integrado.
Infraestructura como código	4	IaC bien implementada, modular y parametrizada.
Análisis de código estático	4	Informe detallado con hallazgos significativos y recomendaciones accionables.
Testing	4	Integración de uno de los tipos de testing mencionados.
Pipelines CI/CD	4	Pipelines implementados para todos los ambientes con etapas claramente definidas.
Observabilidad	4	Puntos de observabilidad detallados

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. Ingresá al sistema de Gestión.
2. En el menú, seleccioná el ítem “Evaluaciones” y la instancia de evaluación correspondiente, que figura bajo el título “Inscripto”.
3. Para iniciar la entrega hacé clic en el ícono: 
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en “Agregar”. El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en “Crear equipo”.

Cualquier integrante podrá:

- **Modificar la integración del equipo.**
- **Subir el archivo de la entrega.**

5. Seleccioná el archivo que deseás entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en “Subir” para iniciar la entrega. Cada equipo (hasta 2 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar). El archivo a subir debe tener **un tamaño máximo de 40mb**

Cuando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.

6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. **La hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc.).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con la Coordinadora o Coordinación adjunta antes de las 20:00hs. del día de la entrega, a través de los mails, alamon@ort.edu.uy o rodriguez_mb@ort.edu.uy; o telefónicamente al 29021505 - int 1156 u 1138