

Arquitectura y diseño de sistemas

Guía para el profesor

Clave MTTI2404

Nivel Maestría

Contenido

Datos generales del certificado.....	3
Competencia global del curso.....	3
Introducción al curso.....	3
Información general	3
Calendario de entregas de los aprendedores.....	6
Temario del curso	7
Preguntas más frecuentes	8
Guía general para las sesiones	9
Rúbricas	¡Error! Marcador no definido.5

Datos generales del certificado

Nombre del certificado: Arquitectura y diseño de sistemas

Modalidad: Connect

Clave: MTTI24

Competencia global del curso

Desarrolla la capacidad de diseñar, evaluar y optimizar arquitecturas de sistemas de información, aplicando principios y prácticas modernas para asegurar que sean escalables, seguras, resilientes y eficientes en entornos distribuidos y en la nube, integrando tecnologías emergentes y prácticas de DevOps para crear soluciones robustas que aborden los desafíos actuales en la gestión de sistemas complejos.

Introducción al curso

En el entorno tecnológico actual, la arquitectura de sistemas es clave para garantizar que las aplicaciones sean seguras, modulares y eficientes. En esta experiencia educativa, estudiarás los principios fundamentales que la sustentan; de esta manera, te prepararás para diseñar y gestionar infraestructuras tecnológicas que cumplan con las demandas del mercado laboral contemporáneo. Asimismo, adquirirás conocimientos esenciales sobre atributos de calidad (como escalabilidad, disponibilidad y rendimiento), ya que estos conceptos te ayudarán a destacar en el ámbito profesional.

Con el creciente uso de tecnologías nativas de la nube y sistemas distribuidos, resulta indispensable que adquieras las habilidades adecuadas para diseñar infraestructuras robustas que se adapten a las necesidades de la empresa. Esta experiencia educativa te proporcionará las habilidades necesarias para lograr este objetivo, pues explorarás prácticas ágiles y DevOps, así como la integración de CI/CD para mejorar la eficiencia del ciclo de vida del sistema.

Además, abordarás temas relacionados con la implementación de medidas de seguridad, diseño de sistemas con alta disponibilidad y tolerancia a fallos, los cuales resultan esenciales para garantizar la continuidad operativa; sin embargo, no solo desarrollarás competencias técnicas, también te capacitarás para asumir roles de liderazgo en proyectos complejos. Al finalizar, estarás preparado para ofrecer soluciones innovadoras que optimicen los sistemas de información de las organizaciones, así como para afrontar diversos retos en un entorno tecnológico cada vez más competitivo.

Información general

Metodología

Este curso ha sido diseñado con la finalidad de ser impartido por un **docente líder con experiencia en el ámbito laboral**, quien compartirá contigo su conocimiento, experiencia y las mejores prácticas que realiza en su labor profesional.

La experiencia de curso promueve la interacción entre aprendedores de la Universidad Tecmilenio como una forma de enriquecer tu formación contrastando la realidad con la de otros compañeros.

Durante cada sesión, el docente transmite su experiencia y actúa como guía en el proceso de aprendizaje durante la realización de las actividades.

El curso es tetramestral y tiene una distribución semanal; en cada semana, se lleva a cabo una sesión. La asistencia a estas sesiones es muy importante para el aprendizaje.

Este curso se conforma por 8 temas y su estructura es la siguiente:

Semana	Módulo	Tema	Evaluable	
1	Módulo 1	Tema 1		
		Tema 2	Actividad 1	
Tema 3				
Tema 4		Actividad 2		
2	Módulo 1	Tema 5		
		Tema 6	Evidencia 1	
Tema 7				
Tema 8		Evidencia final		
3		Módulo 1	Tema 1	
			Tema 2	Actividad 1
Tema 3				
Tema 4			Actividad 2	
4	Módulo 1	Tema 5		
		Tema 6	Evidencia 1	
Tema 7				
Tema 8		Evidencia final		

Bibliografía

Bibliografía opcional

- Richards, M., y Ford, N. (2020). Fundamentals of Software Architecture. An Engineering Approach. Estados Unidos: O'Reilly Media. ISBN: 9781492043454.
- Davis, C. (2019). Cloud Native Patterns: Designing change-tolerant software. Estados Unidos: Manning. ISBN: 9781617294297.

Evaluación

A continuación, puedes revisar el detalle de la evaluación:

Semana	Evaluable	Ponderación
1	Actividad I	15
2	Actividad II	15
3	Evidencia I	30
4	Evidencia final	40
Total		100

Estructura de las sesiones

Las sesiones se dividen en dos o tres bloques. Estas son las actividades que se recomienda realizar:

Bloque 1	Bloque 2
Bienvenida y presentación de agenda.	Recapitulación de lo realizado en el bloque previo.
Actividad de bienestar.	Desarrollo de temas de la semana. <ul style="list-style-type: none"> • Explicación de los temas de la semana con ejercicios prácticos. • Cierre de temas.
Desarrollo de temas de la semana. <ul style="list-style-type: none"> • Aplicación en contextos reales (introducción). • Explicación de los temas de la semana con ejercicios prácticos. 	Explicación de las actividades que deberán realizarse en la semana (fuera de la sesión).
Receso.	

Actividades y evidencias

Las actividades y evidencias han sido diseñados para realizarse de manera individual. Por ende, para promover el dinamismo y la interacción de los participantes en distintos formatos, el profesor alternará (durante las sesiones) intervenciones individuales, plenarias y grupales que enriquezcan los puntos de vista del aprendizador.

Todas las actividades y evidencias deberán entregarse por medio de la plataforma tecnológica, para que el profesor pueda hacer la respectiva revisión y evaluación. Es crucial que el aprendizador revise el esquema de evaluación y los criterios que se utilizarán, con el fin de que tenga claro el nivel de complejidad y esfuerzo requerido para realizar las entregas semanales, con lo cual garantizará el éxito en el curso.

En caso de que el aprendizador tenga dudas sobre alguna actividad o contenido del programa, podrá contactar al profesor a través de los medios indicados.

Sesiones virtuales

Para la transmisión de las sesiones se utiliza una herramienta de videoconferencias. Por ende, con el fin de mejorar la calidad de dichas interacciones, se recomienda lo siguiente:



Tutoriales

Para asegurar que el aprendiz aproveche al máximo su experiencia educativa, se le recomienda que siga las indicaciones del docente, así como la revisión de los siguientes tutoriales:

- [¿Cómo ingreso a la plataforma de multipresencia virtual?](#)
- [Tutoriales de Canvas para participantes.](#)
- [¿Cómo evalúo el desempeño de mi red?](#)

Calendario de entregas de los aprendedores

Semanas	Módulos	Temas	Actividades	Proyecto
1	Módulo 1	Tema 1		
		Tema 2	Actividad 1	
Tema 3				
Tema 4		Actividad 2		
2	Módulo 2	Tema 5		
		Tema 6		Evidencia 1
Tema 7				
3		Tema 8		Evidencia final
4				

Temario del curso

Módulo	Tema
Módulo 1	Tema 1. Principios fundamentales de la arquitectura de sistemas Tema 2. Diseño modular y descomposición de sistemas
	Tema 3. Arquitecturas de sistemas distribuidos Tema 4. Arquitectura orientada a servicios (SOA)
Módulo 2	Tema 5. Arquitectura nativa de la nube y contenedores Tema 6. Seguridad en la arquitectura de sistemas
	Tema 7. DevOps y arquitectura Tema 8. Arquitectura de sistemas de alta disponibilidad

Preguntas más frecuentes

¿En dónde o a quién reporto un error detectado en el contenido?

Cualquier incidencia se puede reportar directamente haciendo clic en el botón “Mejora tu curso” que se encuentra en la parte superior derecha de la pantalla en la plataforma de Canvas.

¿Quién me informa de la cantidad de sesiones y tiempo de cada sesión en las semanas?

El coordinador docente te debe proporcionar esta información.

¿En qué semana se aplica examen final?

Consulta con tu coordinador docente los calendarios de acuerdo con la modalidad de impartición.

¿Tengo que capturar las calificaciones en banner y en la plataforma educativa?

Sí, es importante que captures calificaciones en la plataforma para que los aprendedores estén informados de su avance y reciban retroalimentación de tu parte sobre todo lo que realizan en el certificado. El banner es el registro oficial de las calificaciones de los aprendedores.

Guía general para las sesiones

Bloque 1

Actividad	Descripción
Bienvenida y presentación de agenda.	El profesor se presenta ante el grupo y da una breve introducción al certificado.
Práctica de bienestar.	El profesor impartidor seleccionará alguna práctica de bienestar para aplicarla en la sesión. Se recomienda utilizar una diferente por sesión.
Desarrollo de los temas de la semana: <ul style="list-style-type: none">○ Aplicación en contextos reales (introducción).○ Explicación de los temas de la semana con ejercicios prácticos.	El profesor explicará los contenidos con ejercicios prácticos.
Receso.	Se brindará un espacio de receso para que el aprendedor lo utilice en su beneficio.

Bloque 2

Actividad	Descripción
Recapitulación del bloque previo.	De manera dinámica, el profesor recapitulará lo realizado en el bloque previo.
Desarrollo de los temas de la semana: <ul style="list-style-type: none">○ Explicación de los temas de la semana con ejercicios prácticos.○ Cierre de temas.	El profesor explicará los contenidos con ejercicios prácticos y realizará un cierre de los temas correspondientes.
Explicación sobre lo que deberá realizarse fuera de la sesión: <ul style="list-style-type: none">• Actividades, evidencias, exámenes, etc.	Se brindará una breve explicación de las tareas correspondientes a la semana, las cuales se deberán realizar de forma individual.

Semana 1

Notas para el profesor impartidor correspondientes a la explicación del tema 1 (favor de considerar la realización de ejercicios prácticos durante la sesión)

- Introduce el concepto de arquitectura de sistemas y su diferencia con la arquitectura de software. Puedes usar analogías como la construcción de un edificio, donde la arquitectura de sistemas sería el plano completo que incluye la estructura, las instalaciones eléctricas y las tuberías (hardware), mientras que la arquitectura de software sería el diseño interior y la distribución de los espacios (software).
- Explica con ejemplos cómo la arquitectura de sistemas facilita la comunicación entre los stakeholders, guía el desarrollo, mejora la calidad del sistema y facilita su mantenimiento. Puedes usar ejemplos de

proyectos reales donde una buena arquitectura de sistemas haya sido crucial para el éxito, o casos donde su ausencia haya generado problemas.

- Destaca la importancia de los atributos de calidad. Define cada atributo (rendimiento, escalabilidad, disponibilidad, seguridad, mantenibilidad) y explica cómo influyen en la arquitectura. Muestra ejemplos concretos de cómo se aplican estos atributos en sistemas que los estudiantes conocen, como redes sociales, plataformas de *streaming* o aplicaciones móviles. Puedes usar gráficos y diagramas para visualizar las relaciones entre estos elementos.
- Explica los diferentes estilos arquitectónicos (capas, cliente-servidor, SOA, microservicios, etc.) y proporciona ejemplos de aplicaciones reales que los utilizan. Relaciona estos estilos con patrones arquitectónicos como MVC, Observer y Singleton, mostrando cómo se implementan en la práctica. Es importante que los estudiantes puedan conectar estos conceptos con la tecnología que usan a diario.
- Utiliza diagramas y ejemplos visuales para explicar cómo los componentes y conectores interactúan entre sí para formar un sistema. Puedes usar herramientas de diagramación como Lucidchart o Diagrams (<https://app.diagrams.net/>) para crear representaciones visuales que faciliten la comprensión. Introduce conceptos básicos de UML que serán útiles para modelar la arquitectura del sistema durante todo el curso.
- Explica el modelo OSI y sus siete capas, enfatizando cómo este modelo ejemplifica los principios de la arquitectura en capas. Puedes usar una analogía como el envío de una carta por correo para explicar cómo la información se transmite a través de las diferentes capas del modelo.
- Utiliza ejemplos, analogías y preguntas para mantener a los estudiantes involucrados. Fomenta la realización pequeños debates sobre las ventajas y desventajas de diferentes estilos arquitectónicos, o sobre cómo los atributos de calidad pueden entrar en conflicto entre sí. Incorpora actividades prácticas, como el diseño de una arquitectura simple para una aplicación específica, para que los estudiantes apliquen los conceptos aprendidos.

Notas para el profesor impartidor correspondientes a la explicación del tema 2 (favor de considerar la realización de ejercicios prácticos durante la sesión)

- Define qué es el diseño modular y cómo ayuda a manejar la complejidad en el desarrollo de sistemas. Resalta los beneficios en términos de flexibilidad, escalabilidad y reutilización. Puedes usar la analogía de construir con bloques de Lego, donde cada bloque es un módulo con una función específica y se pueden combinar para crear estructuras complejas.
- Explica las diferencias entre el enfoque monolítico y el modular, destacando las limitaciones de los primeros en cuanto a mantenimiento y escalabilidad. Puedes mostrar ejemplos de aplicaciones que han migrado de una arquitectura monolítica a una modular y los beneficios que obtuvieron.
- Profundiza en los principios de descomposición y abstracción. Explica cómo estos principios ayudan a simplificar el diseño de sistemas. Describe los enfoques *top-down* y *bottom-up* para la descomposición, y cómo la abstracción permite ocultar la complejidad interna de los módulos. Usa diagramas para ilustrar la descomposición de un sistema en módulos y las relaciones entre ellos.
- Destaca la importancia de la cohesión y el acoplamiento. Define ambos conceptos y explica cómo influyen en la calidad del diseño modular. Muestra ejemplos de código para que los estudiantes puedan identificarlos.
- Introduce los patrones de diseño modular con ejemplos. Explica los patrones estructurales, creacionales y de comportamiento. Proporciona ejemplos concretos de su aplicación en sistemas reales. Puedes usar diagramas UML para representar los patrones y su interacción con los módulos.
- Plantea ejercicios prácticos donde los estudiantes deban diseñar la arquitectura modular de una aplicación simple, aplicando los conceptos de cohesión, acoplamiento y patrones de diseño. Puedes usar herramientas de diagramación para que modelen sus diseños.

Notas para el profesor impartidor correspondientes a la actividad 1

- Antes de orientar la actividad, introduce los conceptos básicos de UML que serán necesarios para la creación de los diagramas (diagramas de clases, diagramas de componentes y diagramas de despliegue).

- Para facilitar la retroalimentación y mantener la coherencia, se recomienda que todos los estudiantes trabajen en el diseño de una aplicación web con una temática específica.
- Muestra ejemplos de arquitecturas de sistemas de aplicaciones web similares, para que los estudiantes tengan una referencia.
- Para la elaboración de diagramas, se recomienda el uso de Lucidchart o Diagrams (<https://app.diagrams.net/>). Ambas herramientas son gratuitas y ofrecen funcionalidades para crear diagramas de arquitectura de sistemas. Es importante que los estudiantes se familiaricen con el uso de una herramienta de diagramación y la utilicen de forma consistente durante el curso.
- Si lo consideras necesario puedes proporcionar una plantilla para la documentación de la arquitectura. Esta plantilla pudiera incluir secciones para la descripción de la aplicación, los requisitos funcionales y no funcionales, la arquitectura general, los componentes, los conectores y la justificación de las decisiones de diseño.

Semana 2

Notas para el profesor impartidor correspondientes a la explicación del tema 3 (favor de considerar la realización de ejercicios prácticos durante la sesión)

- Define qué son los sistemas distribuidos y por qué son tan relevantes en la actualidad. Resalta sus ventajas en términos de escalabilidad, alta disponibilidad y tolerancia a fallos. Puedes usar ejemplos de sistemas distribuidos que los estudiantes usan a diario, como redes sociales, plataformas de streaming y aplicaciones de comercio electrónico.
- Explica las características clave de los sistemas distribuidos como: la transparencia, escalabilidad, concurrencia y tolerancia a fallos. Puedes usar ejemplos concretos de cada característica en sistemas reales.
- Describe los diferentes tipos de arquitectura distribuida. Explica las arquitecturas jerárquica, descentralizada e híbrida, y sus ventajas y desventajas. Puedes usar diagramas para representar la estructura de cada tipo de arquitectura.
- Profundiza en los modelos de comunicación y cómo impactan en la complejidad de los sistemas distribuidos. Puedes usar ejemplos de protocolos de comunicación como HTTP, TCP y UDP para ilustrar los diferentes modelos.
- Aborda los desafíos de la sincronización de tiempo, la coherencia de datos, la latencia en la comunicación y la seguridad. Plantea un debate sobre algunas estrategias que se pueden utilizar para mitigar estos desafíos.
- Compara los modelos cliente-servidor y P2P. Explica las diferencias entre ambos modelos, destacando sus ventajas y desventajas en términos de escalabilidad, seguridad y complejidad. Puedes usar ejemplos de aplicaciones que utilizan cada modelo.
- Explica cómo funcionan las arquitecturas *event-driven* y sus beneficios en términos de eficiencia y desacoplamiento. Describe los componentes clave como productores, consumidores y canales de eventos.
- Describe los mecanismos de sincronización (bloqueos y semáforos) y los algoritmos de consenso (Paxos y Raft). Explica cómo los patrones de diseño como Circuit Breaker y Bulkhead ayudan a aumentar la resiliencia.
- Explica el teorema CAP y sus implicaciones en el diseño de sistemas distribuidos. Discute cómo se debe elegir entre consistencia, disponibilidad y tolerancia a particiones según las necesidades de la aplicación.

Notas para el profesor impartidor correspondientes a la explicación del tema 4 (favor de considerar la realización de ejercicios prácticos durante la sesión)

- Introduce el concepto de SOA y explica sus principios clave: interoperabilidad, descubrimiento de servicios, reutilización, contratos definidos y bajo acoplamiento. Puedes usar ejemplos de servicios web como los de IBM, Amazon o Google para ilustrar estos conceptos.

- Define que son los microservicios y explica cómo se diferencian de SOA. Resalta sus características clave: separación en componentes, organización alrededor de capacidades de negocio, despliegue automático y descentralización.
- Explica las diferencias de SOA y los microservicios en términos de granularidad, comunicación, despliegue y organización. Puedes usar un cuadro comparativo para visualizar las diferencias.
- Profundiza en el rol de los servicios web en SOA y en los microservicios. Describe algunos protocolos como SOAP, HTTP y REST, y cómo se utilizan en la comunicación entre servicios.
- Este es un buen punto para introducir algunas herramientas y tecnologías. Menciona *frameworks* como Node.js y Spring Boot o tecnologías de contenedores y orquestadores como Docker y Kubernetes. Explica cómo estas herramientas facilitan la implementación de SOA y microservicios basados en tu experiencia práctica.
- Discute las ventajas de cada enfoque en términos de escalabilidad, flexibilidad y mantenibilidad. También aborda los desafíos de la complejidad, la gestión de datos distribuidos y el monitoreo.

Notas para el profesor impartidor correspondientes a la actividad 2

- El caso de estudio proporcionado es un buen ejemplo de un problema que requiere una arquitectura distribuida. Se puede complementar con información adicional sobre la empresa, como el tipo de productos que vende, el volumen de transacciones esperado y la ubicación geográfica de sus clientes.
- Sugiere a los estudiantes que utilicen la misma herramienta de dibujo de la Actividad 1 para mantener la consistencia.
- El informe debe incluir una descripción detallada de la arquitectura propuesta, justificando las decisiones tomadas en cada etapa del diseño. Se recomienda que el informe siga una estructura similar a la de la Actividad 1, incluyendo secciones para la descripción del problema, la arquitectura general, los componentes, los conectores y la justificación de las decisiones.
- Proporciona ejemplos de arquitecturas distribuidas de aplicaciones de e-commerce, para que los estudiantes tengan una referencia.

Semana 3

Notas para el profesor impartidor correspondientes a la explicación del tema 5 (favor de considerar la realización de ejercicios prácticos durante la sesión)

- Define la arquitectura nativa de la nube y repasa sus principios fundamentales: microservicios, contenedorización, orquestación, infraestructura inmutable, DevOps y observabilidad. Puedes usar ejemplos de aplicaciones nativas de la nube como Netflix o Spotify.
- Explica cómo esta arquitectura permite escalabilidad bajo demanda, alta resiliencia, velocidad de entrega y optimización de costos. Puedes usar ejemplos concretos de empresas que han adoptado la arquitectura nativa de la nube y los beneficios que han obtenido.
- Profundiza en los conceptos de escalabilidad, elasticidad y resiliencia y cómo se aplican en la arquitectura nativa de la nube. Puedes usar diagramas para ilustrar cómo se escalan las aplicaciones en la nube.
- Explica las herramientas Docker y Kubernetes. Repasa su rol en la contenedorización y orquestación de aplicaciones. Puedes mostrar ejemplos prácticos de cómo se definen y despliegan aplicaciones sencillas con Docker y Kubernetes en IBM Cloud.
- Discute los desafíos de la complejidad operacional, la seguridad, la optimización de costos y la independencia del proveedor. Propón un breve debate sobre el uso de buenas prácticas para superar estos desafíos.

Notas para el profesor impartidor correspondientes a la explicación del tema 6 (favor de considerar la realización de ejercicios prácticos durante la sesión)

- Introduce la importancia de la seguridad en la arquitectura de sistemas, especialmente en entornos distribuidos y de microservicios. Puedes mencionar ejemplos de ataques y brechas de seguridad que han tenido un gran impacto.
- Explica los principios clave de la seguridad: confidencialidad, integridad y disponibilidad. Describe con ejemplos prácticos en que consiste la autenticación y autorización, el cifrado y los principios de mínimo privilegio y no repudio.
- Describe las estrategias de seguridad y su implementación en el diseño, la defensa en profundidad, la gestión de riesgos, el cumplimiento normativo y la gestión de identidades y accesos (IAM).
- Explica cómo la superficie de ataque ampliada, la seguridad en la comunicación interna, la gestión de secretos y la automatización presentan desafíos específicos en arquitecturas como los microservicios y los sistemas distribuidos.
- Introduce prácticas de seguridad para microservicios. Describe el principio de confianza cero, la segmentación de red, el uso de API gateways y el monitoreo/detección de amenazas. Utiliza ejemplos de tu experiencia personal en este campo.

Notas para el profesor impartidor correspondientes a la evidencia 1

- Dedicar un pequeño espacio para repasar los conceptos clave de microservicios, arquitectura orientada a servicios, descomposición de sistemas y comunicación entre componentes. Puedes usar ejemplos de sistemas reales que utilicen microservicios (Netflix, Amazon, Uber) para ilustrar los conceptos.
- Guía a los estudiantes en la selección de un problema adecuado que se pueda resolver con una arquitectura de microservicios. Asegúrate de que el problema sea lo suficientemente complejo para que puedan aplicar los conceptos aprendidos, pero no tan complejo que les resulte abrumador.
- Supervisa el proceso de diseño de la arquitectura, prestando atención a la descomposición en microservicios, la definición de las responsabilidades de cada uno y la comunicación entre ellos. Fomenta el uso de diagramas UML para representar la arquitectura.
- Explica las ventajas de usar contenedores y herramientas de orquestación. Introduce Docker y Kubernetes mostrando ejemplos de cómo se utilizan para desplegar microservicios.
- Introduce los conceptos de CI/CD y la importancia de la automatización en el desarrollo de software. Muestra ejemplos de herramientas de CI/CD como Jenkins, GitHub Actions o Azure DevOps.
- Asegúrate de que los estudiantes comprendan la importancia de familiarizarse con la plataforma IBM Cloud. Puedes dedicar un espacio a explorar la plataforma y sus servicios.

Semana 4

Notas para el profesor impartidor correspondientes a la explicación del tema 7 (favor de considerar la realización de ejercicios prácticos durante la sesión)

- Define que es DevOps y explica cómo su integración en el diseño de sistemas permite alcanzar agilidad, calidad y estabilidad. Puedes mencionar ejemplos de empresas que han implementado DevOps con éxito.
- Describe la automatización, la transformación organizacional y la descomposición modular como aspectos clave de la integración de DevOps en el diseño de sistemas.
- Profundiza en las prácticas esenciales en CI/CD. Ejemplifica como se utiliza el repositorio de código, la automatización de pruebas, el pipeline de CI/CD y los entornos replicables. Realiza un ejemplo práctico en la clase de este proceso con el apoyo de IBM Cloud.
- Define IaC y describe sus enfoques declarativos e imperativos. Menciona herramientas como Terraform y Ansible. Destaca los beneficios de IaC en términos de consistencia, escalabilidad y control de versiones. Realiza un ejemplo práctico en la clase de estas herramientas con el apoyo de IBM Cloud.
- Explica la importancia del monitoreo continuo y las prácticas de observabilidad. Describe los componentes clave: logs, métricas y trazas.

Notas para el profesor impartidor correspondiente a la explicación del tema 8 (favor de considerar la realización de ejercicios prácticos durante la sesión)

- Introduce la importancia de la alta disponibilidad y la recuperación ante desastres. Explica por qué la continuidad y disponibilidad de los sistemas son esenciales en escenarios críticos. Puedes mencionar ejemplos de sistemas donde la alta disponibilidad es crucial, como sistemas bancarios o de atención médica.
- Explica los principios relacionados con estos conceptos. Describe la redundancia, el *failover* y las estrategias de recuperación ante desastres como el almacenamiento de copias de seguridad y la replicación geográfica. Puedes utilizar ejemplos de cómo IBM Cloud implementa estas funciones en su plataforma.
- Explica las técnicas de tolerancia a fallos, como la redundancia, la replicación y los algoritmos de consenso. Describe los patrones de diseño como Circuit Breaker y Bulkhead.
- Describe las configuraciones de replicación maestro-esclavo y maestro-maestro. Explica los modelos de consistencia fuerte y eventual.
- Plantea las estrategias de failover y redundancia, explicando la redundancia activa-activa, activa-pasiva y las configuraciones N+1 / N+M. Puedes apoyarte en gráficos y ejemplos que permitan comprender estos conceptos de forma rápida y sencilla.

Notas para el profesor impartidor correspondientes a la evidencia final

- Guía a los estudiantes en la implementación de los microservicios, prestando atención a la elección del lenguaje de programación, el uso de frameworks o librerías y la aplicación de buenas prácticas de desarrollo.
- Supervisa el proceso de creación de las imágenes de Docker y el despliegue de los contenedores en IBM Cloud. Resuelve dudas sobre el uso de estas herramientas.
- Ayuda a los estudiantes a configurar el pipeline de CI/CD, integrando las herramientas de IBM Cloud. Explica la importancia de las pruebas automatizadas en el proceso de CI/CD.
- Guía a los estudiantes en el desarrollo de la prueba de concepto, asegurándote de que demuestren la funcionalidad de la arquitectura, la comunicación entre microservicios, la escalabilidad y las medidas de seguridad.
- Insiste en la importancia de la documentación del proceso de desarrollo, tanto para la comprensión del sistema como para el mantenimiento futuro.
- Fomenta la reflexión sobre el proceso de desarrollo, los desafíos encontrados y las soluciones implementadas. Esto les ayudará a consolidar su aprendizaje y a mejorar sus habilidades de resolución de problemas.

Arquitectura y Diseño de Sistemas

Rúbrica de evaluación

Actividad I

Nivel de desempeño				
Criterios de evaluación	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	%
1. Definición del contexto y requisitos del sistema	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	20
	Realiza una definición clara del contexto y requisitos del sistema.	Realiza una definición del contexto, pero no menciona claramente los requisitos del sistema.	No realiza una definición clara del contexto y requisitos del sistema.	
2. Aplicación de los principios de arquitectura y selección de estilos	25 – 23 puntos	22 – 20 puntos	19 – 0 puntos	25
	Aplica correctamente los principios de arquitectura y selección de estilos.	Presenta algunos errores al aplicar los principios de arquitectura y selección de estilos.	Aplica incorrectamente los principios de arquitectura y selección de estilos.	
3. Selección y justificación de los componentes y conectores del sistema	25 – 23 puntos	22 – 20 puntos	19 – 0 puntos	25
	Selecciona y justifica correctamente los componentes y conectores del sistema.	Muestra problemas al seleccionar y justificar los componentes y conectores del sistema.	Presenta una selección y una justificación inadecuada sobre los componentes y conectores del sistema.	
4. Diagrama conceptual del sistema	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	20
	Elabora de manera precisa y coherente el diagrama conceptual del sistema.	Elabora un diagrama conceptual del sistema, pero está incompleto.	No elabora un diagrama conceptual correcto.	
5. Informe y justificación	10 – 8 puntos	7 – 5 puntos	4 – 0 puntos	10
	Realiza un informe con justificaciones coherentes y acertadas.	Realiza un informe, pero no presenta justificaciones.	Realiza un informe incorrecto.	
TOTAL				100%

Arquitectura y Diseño de Sistemas

Rúbrica de evaluación

Actividad II

Nivel de desempeño				
Criterios de evaluación	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	%
1. Diseño arquitectónico	30 – 28 puntos	27 – 25 puntos	24 – 0 puntos	30
	Presenta un diseño arquitectónico adecuado y de excelente calidad.	Presenta, pero no de manera clara, el diseño arquitectónico.	Muestra un diseño arquitectónico incorrecto.	
2. Estrategias de tolerancia a fallos y escalabilidad	25 – 23 puntos	22 – 20 puntos	19 – 0 puntos	25
	Utiliza adecuadamente estrategias de tolerancia a fallos y escalabilidad.	Presenta problemas para utilizar estrategias de tolerancia a fallos y escalabilidad.	No presenta estrategias adecuadas de tolerancia a fallos y escalabilidad.	
3. Diagrama de arquitectura	25 – 23 puntos	22 – 20 puntos	19 – 0 puntos	25
	Presenta un diagrama de arquitectura detallado y de manera clara.	Presenta un diagrama de arquitectura incompleto.	Realiza un diagrama incorrecto.	
4. Informe y justificación	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	20
	Redacta un informe con justificaciones coherentes y acertadas.	Realiza un informe, pero no presenta justificaciones.	Realiza un informe incorrecto.	
TOTAL				100%

Evidencia 1 Arquitectura y diseño de sistemas Rúbrica de evaluación

Nivel de desempeño

Criterios de evaluación	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	%
1. Claridad en la definición del sistema y sus objetivos	10 – 8 puntos	7 – 5 puntos	4 – 0 puntos	10
	Describe de manera clara y detallada el sistema propuesto, incluyendo su funcionalidad y objetivos. Además, identifica correctamente los requerimientos funcionales y no funcionales.	Describe el sistema y sus objetivos de forma general, pero carece de algunos detalles específicos. Además, identifica la mayoría de los requerimientos principales.	La definición del sistema y sus objetivos son confusos o están incompletos. No identifican adecuadamente los requerimientos principales.	
2. Desarrollo de la arquitectura de microservicios	30 – 27 puntos	26 – 23 puntos	22 – 0 puntos	30
	Descompone el sistema en microservicios de forma lógica y detallada, definiendo claramente las responsabilidades de cada uno. Elabora diagramas completos y precisos que muestran la interacción y comunicación entre ellos.	Realiza una descomposición adecuada del sistema, pero falta claridad en algunas responsabilidades o detalles en los diagramas de arquitectura.	La descomposición es insuficiente o inadecuada. Los diagramas son incompletos, erróneos o están ausentes.	
3. Planificación de contenedorización y orquestación	20 – 17 puntos	16 – 14 puntos	13 – 0 puntos	20
	Planifica detalladamente el uso de contenedores para cada microservicio, definiendo imágenes, versiones y tecnologías. Investiga y selecciona la herramienta de orquestación más adecuada, justificando su elección.	Realiza una planificación básica de contenedorización y orquestación, pero carece de algunos detalles o justificaciones en la selección de herramientas.	La planificación es insuficiente o carece de detalles importantes. No investiga adecuadamente las herramientas de orquestación.	
4. Integración de prácticas de DevOps en el diseño	20 – 17 puntos	16 – 14 puntos	13 – 0 puntos	20
	Diseña un pipeline de CI/CD detallado, que facilita el desarrollo y despliegue continuo, incluyendo pruebas automatizadas y despliegue en la nube. Demuestra una comprensión profunda de las prácticas DevOps.	Diseña un pipeline de CI/CD básico, pero omite algunos elementos como pruebas automatizadas o detalles del despliegue en la nube.	No integra adecuadamente las prácticas de DevOps en el diseño. Además, el pipeline de CI/CD es incompleto o ineficaz.	
5. Familiarización con las herramientas de IBM Cloud	10 – 7 puntos	6 – 4 puntos	3 – 0 puntos	10
	Completa todos los módulos solicitados del curso IBM Cloud Essentials y muestra	Completa solo uno de los módulos solicitados del curso IBM Cloud Essentials, pero muestra	No realiza ningún módulo del curso IBM Cloud Essentials ni muestra evidencias.	

	correctamente las evidencias requeridas.	las evidencias requeridas.		
5. Redacción de la documentación de la arquitectura	10 – 7 puntos	6 – 4 puntos	3 – 0 puntos	10
	El informe está bien estructurado, es claro y coherente, incluye toda la documentación requerida con redacción profesional y sin errores ortográficos o gramaticales.	El informe incluye la mayoría de la documentación, pero presenta problemas menores de estructura, claridad o algunos errores de redacción.	El informe es incompleto, desorganizado o tiene serios problemas de redacción. Falta documentación importante.	
TOTAL				100%

Evidencia 2
Arquitectura y diseño de sistemas
Rúbrica de evaluación

Nivel de desempeño				
Criterios de evaluación	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	%
1. Implementación correcta de los microservicios	25 – 22 puntos	21 – 18 puntos	17 – 0 puntos	25
	Implementa al menos dos microservicios esenciales de forma eficiente, siguiendo buenas prácticas de programación. Cada microservicio funciona de manera independiente y cumple con las funcionalidades requeridas.	Implementa los microservicios esenciales, pero presenta pequeñas fallas o inconsistencias. Los microservicios funcionan, pero requieren ajustes.	La implementación es incompleta o tiene errores significativos. Los microservicios no funcionan adecuadamente.	
2. Contenedorización y despliegue en la nube exitoso	25 – 22 puntos	21 – 16 puntos	15 – 0 puntos	25
	Realiza correctamente la contenedorización utilizando scripts de Docker y despliega exitosamente los contenedores en IBM Cloud con una herramienta de orquestación.	Implementa los contenedores y despliega los microservicios, pero enfrenta problemas menores durante el proceso. El despliegue es funcional, pero requiere optimización.	No logra implementar los contenedores o desplegar adecuadamente los microservicios en la nube. Presenta errores significativos.	
3. Configuración efectiva del pipeline CI/CD	20 – 17 puntos	16 – 14 puntos	13 – 0 puntos	20
	Configura un pipeline de CI/CD que automatiza eficazmente la construcción, prueba y despliegue de los microservicios, integrando herramientas sugeridas por IBM. Documenta	Configura un pipeline de CI/CD básico que automatiza parte del proceso, pero omite algunos elementos o detalles en la documentación.	No configura un pipeline de CI/CD funcional. Falta automatización en etapas clave y la documentación es insuficiente.	

	detalladamente el proceso.			
4. Realización de la prueba de concepto y aplicación de medidas de seguridad	20 – 17 puntos	16 – 14 puntos	13 – 0 puntos	20
	Desarrolla una prueba de concepto que demuestra plenamente la funcionalidad de la arquitectura, incluyendo comunicación entre microservicios, escalabilidad y medidas de seguridad básicas. El video muestra claramente el funcionamiento.	Realiza una prueba de concepto funcional, pero omite algunas pruebas o medidas de seguridad. El video es adecuado, pero puede mejorar en claridad.	La prueba de concepto es incompleta o no demuestra adecuadamente la funcionalidad. El video es insuficiente o inexistente.	
5. Informe final y reflexión sobre el proceso	10 – 7 puntos	6 – 4 puntos	3 – 0 puntos	10
	El informe es detallado y bien estructurado, describiendo el proceso de implementación y despliegue. La reflexión final es profunda y analiza los desafíos y soluciones adoptadas.	El informe cubre el proceso general, pero carece de detalles en algunos aspectos. La reflexión final es superficial o incompleta.	El informe es desorganizado o incompleto. La reflexión final es insuficiente o está ausente.	
TOTAL				100%

Administración de Proyectos de TI
Rúbrica de evaluación
Evidencia 1

Nivel de desempeño				
Criterios de evaluación	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	%
1. Descripción de la oportunidad de negocio, justificación y declaración de la visión	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	20
	Identifica una oportunidad de negocio para desarrollar un nuevo producto o servicio de TI; además, elabora una justificación de media cuartilla acerca del negocio, es decir, explica por qué considera que su realización es factible. Finalmente, define la declaración de la visión basada en la problemática identificada.	Identifica una oportunidad de negocio para desarrollar un nuevo producto o servicio de TI; sin embargo, no logra justificar de manera clara el negocio. Finalmente, define la declaración de la visión basada en la problemática identificada.	Identifica una oportunidad de negocio para desarrollar un nuevo producto o servicio de TI; sin embargo, no logra justificar de manera clara el negocio. Tampoco define la declaración de la visión basada en la problemática identificada.	
2. Estrategia de implementación de Scrum	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	20
	Define y explica la estrategia para implementar el proyecto con el uso de la metodología Scrum; para ello, toma en cuenta todos los puntos	Define y explica la estrategia para implementar el proyecto con el uso de la metodología Scrum; sin embargo, no incluye algún punto mencionado en las instrucciones.	Define, pero no explica la estrategia para implementar el proyecto con el uso de la metodología Scrum; además, no incluye algunos	

	mencionados en las instrucciones.		puntos mencionados en las instrucciones.	
3. Creación del <i>product backlog</i>	30 – 28 puntos	27 – 25 puntos	24 – 0 puntos	30
	Identifica los requerimientos detallados del proyecto y crea la lista priorizada de productos (product backlog).	Identifica los requerimientos detallados del proyecto, pero presenta errores en la creación de la lista priorizada de productos (product backlog).	Crea una lista incorrecta de productos (product backlog).	
4. Elaboración del plan de lanzamiento y cronograma de Gantt	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	20
	Genera un cronograma con una planeación de lanzamiento del producto final. También elabora un cronograma de Gantt con la planeación del lanzamiento a alto nivel; asimismo, incluye las fases generales del proceso Scrum. El cronograma de Gantt incluye todos los sprint, desde el inicio hasta el final del proyecto estimado.	Genera un cronograma con una planeación de lanzamiento del producto final. También elabora un cronograma de Gantt con la planeación del lanzamiento a alto nivel, pero no incluye las fases generales del proceso Scrum.	Genera un cronograma incompleto con una planeación de lanzamiento del producto final. De igual manera, elabora un cronograma de Gantt incompleto, pues no incluye las fases generales del proceso Scrum.	
5. Investigación sobre <i>burndown chart</i> , métricas de rendimiento y conclusión acerca de las herramientas de TI	10 – 7 puntos	6 – 4 puntos	3 – 0 puntos	10
	Realiza una investigación donde responde las siguientes preguntas: - ¿Qué es un gráfico de trabajo pendiente (burndown chart)? - ¿Qué ventajas tiene el uso de un gráfico de trabajo pendiente en Scrum? - Con un ejemplo, explica cómo se construye y utiliza un gráfico de trabajo pendiente. - Describe los indicadores clave y métricas que se pueden utilizar para revisar el estatus del rendimiento de un proyecto en Scrum. También elabora una conclusión de media	Realiza una investigación, pero no responde en su totalidad las siguientes preguntas. - ¿Qué es un gráfico de trabajo pendiente (burndown chart)? - ¿Qué ventajas tiene el uso de un gráfico de trabajo pendiente en Scrum? - Con un ejemplo, explica cómo se construye y utiliza un gráfico de trabajo pendiente. - Describe los indicadores clave y métricas que se pueden utilizar para revisar el estatus del rendimiento de un proyecto en Scrum. También elabora una conclusión de media	Realiza una investigación, pero no responde en su totalidad las siguientes preguntas; - ¿Qué es un gráfico de trabajo pendiente (burndown chart)? - ¿Qué ventajas tiene el uso de un gráfico de trabajo pendiente en Scrum? - Con un ejemplo, explica cómo se construye y utiliza un gráfico de trabajo pendiente. - Describe los indicadores clave y métricas que se pueden utilizar para revisar el estatus del rendimiento de un proyecto en Scrum.	

	cuartilla sobre las herramientas de TI de apoyo para gestionar proyectos bajo la metodología Scrum. De igual manera, justifica sus planteamientos con fuentes bibliográficas en formato APA.	herramientas de TI de apoyo para gestionar proyectos bajo la metodología Scrum; sin embargo, justifica vagamente sus planteamientos, aunque sí agrega fuentes bibliográficas en formato APA.	Sin embargo, no presenta una conclusión adecuada ni justifica sus planteamientos con fuentes bibliográficas.	
TOTAL				100%

**Administración de Proyectos de TI
Rúbrica de evaluación
Evidencia final**

Nivel de desempeño				
Criterios de evaluación	Altamente competente 100%-86%	Competente 85%-70%	Aún sin desarrollar la competencia 69%-0%	%
1. Análisis y refinamiento del <i>product backlog</i>	15 – 13 puntos	12 – 10 puntos	9 – 0 puntos	15
	Realiza un análisis detallado sobre el refinamiento del product backlog y lo muestra actualizado; en caso de que no haya actualización, justifica por qué.	Realiza un análisis sobre el refinamiento del producto backlog, pero no presenta una actualización o, en su defecto, no explica por qué no se produce ninguna actualización.	Realiza un análisis incorrecto.	
2. Preparación y estimación del <i>sprint backlog</i>	25 – 22 puntos	21 – 19 puntos	18 – 0 puntos	25
	Prepara una lista de pendientes e identifica las historias de usuario (sprint backlog) de manera correcta. También completa todas las tareas mencionadas en las instrucciones	Prepara una lista de pendientes, pero no logra identificar las historias de usuario (sprint backlog) de manera correcta. De igual manera, solo completa algunas de las tareas mencionadas en las instrucciones.	Realiza una preparación incorrecta y no identifica las historias de usuario (sprint backlog). Tampoco completa las tareas mencionadas en las instrucciones.	
3. Ejecución de la simulación del <i>sprint 0</i>	25 – 22 puntos	21 – 19 puntos	18 – 0 puntos	25
	Ejecuta la simulación del <i>sprint 0</i> de manera completa; además, cumple con todos los puntos que se encuentran en las instrucciones.	Ejecuta la simulación <i>sprint 0</i> , pero cumple solo con algunos puntos que se encuentran en las instrucciones.	Ejecuta de manera incorrecta la simulación del <i>sprint 0</i> .	
	20 – 18 puntos	17 – 15 puntos	14 – 0 puntos	

4. Resultados, revisión y retrospectiva del sprint	Presenta los resultados y describe a detalle cómo se ejecuta la ceremonia de revisión del sprint (<i>sprint review</i>); además, contesta correctamente todas las preguntas de las instrucciones.	Presenta los resultados y describe a detalle cómo se ejecuta la ceremonia de revisión del sprint (<i>sprint review</i>); sin embargo, contesta solo algunas de las preguntas de las instrucciones.	Presenta los resultados incorrectos; además, solo contesta algunas de las preguntas de las instrucciones.	20
5. Propuesta de mejora	<p>15 – 13 puntos</p> <p>Realiza una propuesta para mejorar el desempeño del equipo del proyecto, a partir de la implementación de tablero Kanban. De igual manera, justifica su estrategia desarrollando cada uno de los puntos mencionados en las instrucciones:</p> <ul style="list-style-type: none"> • ¿Cuáles son las principales ventajas de incorporar el tablero Kanban al proyecto? ¿Por qué? • Elabora el tablero Kanban propuesto e identifica lo siguiente: <ul style="list-style-type: none"> ○ Columnas o estados del tablero y justificación de su uso. ○ Definición de los límites de trabajo (WIP, del inglés <i>F</i>) y justificación de tu propuesta. ○ Ejemplo del tablero gráfico con la muestra de algunas actividades. <p>Finalmente, elabora una conclusión adecuada para el cierre de la evidencia.</p>	<p>12 – 10 puntos</p> <p>Realiza una propuesta para mejorar el desempeño del equipo del proyecto, a partir de la implementación de tablero Kanban. Además, no justifica su propuesta en alguno de los siguientes puntos contemplados en las instrucciones:</p> <ul style="list-style-type: none"> • ¿Cuáles son las principales ventajas de incorporar el tablero Kanban al proyecto? ¿Por qué? • Elabora el tablero Kanban propuesto e identifica lo siguiente: <ul style="list-style-type: none"> ○ Columnas o estados del tablero y justificación de su uso. ○ Definición de los límites de trabajo (WIP, del inglés <i>F</i>) y justificación de tu propuesta. ○ Ejemplo del tablero gráfico con la muestra de algunas actividades. <p>Finalmente, elabora una conclusión para el cierre de la evidencia, pero con algunas inconsistencias.</p>	<p>9 – 0 puntos</p> <p>Realiza una propuesta insustancial; además, no justifica su estrategia.</p>	15
TOTAL				100%