



Subódulo III. Implementa software de sistemas

1. Escala valorativa: Informe de configuración e software.

Valor: 6.0

Heteroevaluación

Alumno (a):		Grupo: 203 N.L.:				
No.	Criterios a evaluar en actividades teóricas	En proceso	Suficiente	Bueno	Excelente	
		0	de 2 a 1	de 4 a 3	5	
1	Portada (5 puntos)	0 puntos: No incluye portada.	1-2 puntos: Faltan varios elementos o la portada está incompleta.	3-4 puntos: Faltan uno o dos elementos, o la presentación no es del todo clara.	5 puntos: Incluye todos los elementos solicitados (título, nombres de los integrantes, nombre del curso, docente y fecha). Presentación clara y profesional.	
2	Introducción (10 puntos)	0-3 puntos: No cumple con lo solicitado o está incompleta.	4-6 puntos: La introducción es confusa o falta información relevante.	7-9 puntos: La explicación es clara, pero falta profundidad en las generalidades o en el resumen.	10 puntos: Explica de manera clara y concisa las generalidades de la aplicación y resume el contenido del informe.	
Desarrollo						
3	Requisitos de Software (10 puntos)	0-3 puntos: No aborda el tema o lo hace de manera incorrecta.	4-6 puntos: La información es incompleta o confusa.	7-9 puntos: La descripción es clara, pero falta algún detalle importante.	10 puntos: Describe de manera detallada y precisa los requisitos técnicos de software y desarrollo.	
4	Configuración de hardware y software (15 puntos)	0-4 puntos: No cumple con lo solicitado.	5-9 puntos: La explicación es confusa o incompleta.	10-14 puntos: El tutorial es claro, pero falta algún paso o detalle.	15 puntos: Explica paso a paso, en forma de tutorial, cómo se configuró el entorno. Es claro y detallado.	
5	Tabla de compatibilidad de software (5 puntos)	0 puntos: No incluye la tabla.	1-2 puntos: La tabla es confusa o incompleta.	3-4 puntos: La tabla está presente, pero falta claridad o algún detalle.	5 puntos: La tabla es clara, completa y está bien organizada.	
6	Configuración de entorno virtual en Python (10 puntos)	0-3 puntos: No aborda el tema o lo hace de manera incorrecta.	4-6 puntos: La explicación es confusa o incompleta.	7-9 puntos: La explicación es clara, pero falta algún detalle.	10 puntos: Explica de manera clara y detallada cómo se configuró el entorno virtual en Python.	
7	Conclusiones (20 puntos)	0 puntos: No incluye la tabla.	10-14 puntos: Las respuestas son incompletas o poco reflexivas.	15-19 puntos: Las respuestas son claras, pero falta profundidad en alguna de ellas.	20 puntos: Cada integrante responde de manera reflexiva y clara a las preguntas planteadas.	
8	Anexos (10 puntos)	0-3 puntos: No incluye glosario.	4-6 puntos: El glosario es incompleto o confuso.	7-9 puntos: El glosario está presente, pero falta claridad o algunos términos.	10 puntos: El glosario de términos es completo, claro y bien organizado.	Total: (suma/100) * 6
9	Presentación y colaboración (15 puntos)	0-4 puntos: No cumple con los requisitos de presentación o colaboración.	5-9 puntos: El informe está desorganizado o no muestra trabajo en equipo.	10-14 puntos: El informe es claro, pero tiene algunos errores de formato o falta evidencia de colaboración.	15 puntos: El informe está bien estructurado, es claro y muestra evidencia de trabajo colaborativo.	

Excelente: 6.0 Bueno: 5.0 Regular: 3.5 Deficiente: 2.5 En proceso: 1.5

RETROALIMENTACIÓN

Aspecto a mejorar: centrado en la tarea Centrado en el proceso centrado en la autorregulación

Acciones de mejora: Para alumnos que **NO** realizan actividades en casa o que **NO** tienen los materiales para realizar la práctica, se les proporciona una actividad alterna en clase para que se pongan al día, mientras los demás trabajan en algo más avanzado.

Evidencia: Elaboración de la actividad o corrección de la práctica.

Puntaje al 80%:

Firma del Alumno (a)

Sello o firma Docente



Subódulo III. Implementa software de sistemas informáticos

Docente: MTE. Lizbeth Alcántara Blas

Semestre: Segundo

Academia: Programación

2. Lista de Cotejo: Simulación de roles

Valor: 2.0

Autoevaluación

Alumno (a): _____ Grupo: 203 N.L.: _____

No.	Criterio	Sí	No	En proceso	Observaciones
1. Investigación y Preguntas Guía (20 puntos)					
1	Explican qué roles son esenciales en un proyecto de software y por qué.				
2	Identifican situaciones adecuadas para usar el modelo en cascada.				
3	Describen las diferencias entre el modelo en cascada y el ágil.				
Actividad 1: Cascada vs. Ágil					
1	Presentan una descripción clara de la metodología asignada.				
2	Mencionan ejemplos de proyectos donde sería útil la metodología.				
3	Enumeran ventajas y desventajas de la metodología.				
4	Defienden su postura de manera convincente y con argumentos válidos.				
Actividad 2: Simulación de Roles					
1	Asignan roles claros (analista, diseñador, programador, tester).				
2	Explican cómo colaborarían en cada metodología (cascada o ágil).				
3	Muestran trabajo en equipo y coordinación durante la simulación.				
3. Puesta en Común y Reflexión					
1	Presentan conclusiones claras sobre las metodologías.				
2	Reflexionan sobre la simulación de roles y su experiencia.				
3	Responden a las preguntas de reflexión: - ¿Qué metodología les parece más flexible y por qué? - ¿Qué rol les gustaría desempeñar en un proyecto real y por qué?				
Tota=(Suma/13) * 2:					

3. Escala valorativa: Estudio de caso 1

Valor: 2.0

Coevaluación

Alumno (a) evaluador(a): _____ Grupo: 203 N.L.: _____

Criterio	Excelente	Bueno	Regular	Deficiente	Nulo
1. ¿Qué causó el fallo en el sistema de Target durante el Black Friday?					
1.1. Identifica de manera clara y precisa la causa del fallo.	1	0.8	0.6	0.4	0
1.2. Explica la causa, pero falta profundidad o detalle.	1	0.8	0.6	0.4	0
1.3. La explicación es confusa o no está basada en información relevante.	1	0.8	0.6	0.4	0
2. ¿Qué consecuencias tuvo este fallo para la empresa y los clientes?					
2.1. Describe de manera clara y completa las consecuencias para la empresa y los clientes.	1	0.8	0.6	0.4	0
2.2. Menciona algunas consecuencias, pero falta profundidad o detalle.	1	0.8	0.6	0.4	0
2.3. La explicación es incompleta o no está bien fundamentada.	1	0.8	0.6	0.4	0
3. ¿Qué medidas podrían haberse tomado para evitar este problema?					
3.1. Propone medidas concretas, viables y bien fundamentadas para evitar el problema.	1	0.8	0.6	0.4	0
3.2. Sugiere medidas, pero falta claridad o viabilidad en algunas.	1	0.8	0.6	0.4	0
3.3. Las medidas propuestas son poco claras o no están relacionadas con el problema.	1	0.8	0.6	0.4	0
4. ¿Cómo crees que Target podría mejorar su sistema para evitar fallos en el futuro?					
4.1. Presenta propuestas claras, innovadoras y bien fundamentadas para mejorar el sistema.	1	0.8	0.6	0.4	0
4.2. Sugiere mejoras, pero falta claridad o profundidad en la explicación.	1	0.8	0.6	0.4	0
4.3. Las propuestas son poco claras o no están relacionadas con el problema.	1	0.8	0.6	0.4	0
Tota=(Suma/12) * 2:					

Excelente: 2.0 Bueno: 1.5 Regular: 1.0
Deficiente: 0.5 En proceso: 0

RETROALIMENTACIÓN

Aspecto a mejorar: centrado en la tarea Centrado en el proceso centrado en la autorregulación

Acciones de mejora: Para alumnos que **NO** realizan actividades en casa o que **NO** tienen los materiales para realizar la práctica, se les proporciona una actividad alterna en clase para que se pongan al día, mientras los demás trabajan en algo más avanzado.

Evidencia: Elaboración de la actividad o corrección de la práctica.

Puntaje al 80%:

Firma del Alumno (a)

Sello o firma Docente



Submódulo III. Implementa software de sistemas informáticos

Docente: MTE. Lizbeth Alcántara Blas

Semestre: Segundo

Academia: Programación

3. Escala valorativa: Estudio de caso 2

Valor: 2.0

Coevaluación

Alumno (a):	Grupo: 203 N.L.:
Alumno (a) evaluador(a):	Grupo: 203 N.L.:

Criterio	Excelente	Bueno	Regular	Deficiente	Nulo
1. Revisión en Plenaria: Preguntas Generales					
1.1. Identifica y explica normas y estándares de calidad relevantes en el desarrollo de software.	1	0.8	0.6	0.4	0
1.2. Describe cómo ITIL y DevOps contribuyen a la calidad del software.	1	0.8	0.6	0.4	0
1.3. Explica la importancia de cumplir con regulaciones y políticas en el desarrollo de software.	1	0.8	0.6	0.4	0
2. Estudio de Caso 2: Facebook y el GDPR					
Pregunta 1: ¿Qué cambios implementó Facebook para cumplir con el GDPR?					
2.1. Describe de manera clara y completa los cambios implementados por Facebook.	1	0.8	0.6	0.4	0
2.2. Menciona algunos cambios, pero falta profundidad o detalle.	1	0.8	0.6	0.4	0
2.3. La explicación es incompleta o no está bien fundamentada.	1	0.8	0.6	0.4	0
Pregunta 2: ¿Cómo mejoró la calidad del software de Facebook como resultado de estos cambios?					
2.4. Explica de manera clara y detallada cómo mejoró la calidad del software.	1	0.8	0.6	0.4	0
2.5. Menciona algunas mejoras, pero falta claridad o profundidad.	1	0.8	0.6	0.4	0
2.6. La explicación es confusa o no está relacionada con la calidad.	1	0.8	0.6	0.4	0
Pregunta 3: ¿Qué impacto tuvo el cumplimiento del GDPR en la experiencia del usuario?					
2.7. Describe de manera clara y completa el impacto en la experiencia del usuario.	1	0.8	0.6	0.4	0
2.8. Menciona algunos impactos, pero falta profundidad o detalle.	1	0.8	0.6	0.4	0
2.9. La explicación es incompleta o no está bien fundamentada.	1	0.8	0.6	0.4	0
Pregunta 4: ¿Qué lecciones se pueden extraer de este caso para otros proyectos de software?					
2.10. Presenta lecciones claras, relevantes y bien fundamentadas.	1	0.8	0.6	0.4	0
2.11. Sugiere algunas lecciones, pero falta claridad o profundidad.	1	0.8	0.6	0.4	0
2.12. Las lecciones son poco claras o no están relacionadas con el caso.	1	0.8	0.6	0.4	0
Tota=(Suma/15) * 2:					

Excelente: 2.0
Bueno: 1.5
Regular: 1.0
Deficiente: 0.5
En proceso: 0

3. Escala valorativa: Estudio de caso 3

Valor: 2.0

Coevaluación

Criterio	Excelente	Bueno	Regular	Deficiente	Nulo
1. Revisión en Plenaria: Preguntas Generales					
1.1. Explica qué son las pruebas de funcionalidad y su importancia.	1	0.8	0.6	0.4	0
1.2. Describe cómo se realizan las pruebas de rendimiento y las métricas utilizadas.	1	0.8	0.6	0.4	0
1.3. Identifica los aspectos evaluados en las pruebas de seguridad.	1	0.8	0.6	0.4	0
1.4. Explica cómo se realizan las pruebas de usabilidad y sus beneficios.	1	0.8	0.6	0.4	0
1.5. Describe qué se evalúa en las pruebas de compatibilidad.	1	0.8	0.6	0.4	0
Pregunta 1: ¿Qué tipos de pruebas realizó Microsoft para garantizar la calidad de Windows 10?					
2.1. Describe de manera clara y completa los tipos de pruebas realizadas.	1	0.8	0.6	0.4	0
2.2. Menciona algunos tipos de pruebas, pero falta profundidad o detalle.	1	0.8	0.6	0.4	0
2.3. La explicación es incompleta o no está bien fundamentada.	1	0.8	0.6	0.4	0
Pregunta 2: ¿Cómo impactan estas pruebas en la calidad del producto final?					
2.4. Explica de manera clara y detallada el impacto de las pruebas en la calidad.	1	0.8	0.6	0.4	0
2.5. Menciona algunos impactos, pero falta claridad o profundidad.	1	0.8	0.6	0.4	0
2.6. La explicación es confusa o no está relacionada con la calidad.	1	0.8	0.6	0.4	0
Pregunta 3: ¿Qué lecciones se pueden extraer de este caso para otros proyectos de software?					
2.7. Presenta lecciones claras, relevantes y bien fundamentadas.	1	0.8	0.6	0.4	0
2.8. Sugiere algunas lecciones, pero falta claridad o profundidad.	1	0.8	0.6	0.4	0
2.9. Las lecciones son poco claras o no están relacionadas con el caso.	1	0.8	0.6	0.4	0
Pregunta 4: ¿Cómo podrían aplicarse estas prácticas de prueba en un proyecto de menor escala?					
2.10. Propone aplicaciones prácticas y viables para proyectos de menor escala.	1	0.8	0.6	0.4	0
2.11. Sugiere algunas aplicaciones, pero falta claridad o viabilidad.	1	0.8	0.6	0.4	0
2.12. Las propuestas son poco claras o no están relacionadas con el caso.	1	0.8	0.6	0.4	0
Tota=(Suma/17) * 2:					

Excelente: 2.0
Bueno: 1.5
Regular: 1.0
Deficiente: 0.5

RETROALIMENTACIÓN

Aspecto a mejorar: centrado en la tarea Centrado en el proceso centrado en la autorregulación

Acciones de mejora: Para alumnos que **NO** realizan actividades en casa o que **NO** tienen los materiales para realizar la práctica, se les proporciona una actividad alterna en clase para que se pongan al día, mientras los demás trabajan en algo más avanzado.

Evidencia: Elaboración de la actividad o corrección de la práctica.

Puntaje al 80%:

Firma del Alumno (a)

Sello o firma Docente



Subódulo III. Implementa software de sistemas informáticos

5. Escala valorativa: Mapa conceptual

Valor: 2.0

Heteroevaluación

Alumno (a):	Grupo: 203 N.L.:
-------------	------------------

Criterio		Excelent e	Buen o	Regula r	Deficient e	Nul o
1. Trabajo en Equipo: Discusión de Preguntas Guía						
1.1	Participación activa y colaborativa en la discusión.	5	4	3	2	0
1.2	Respuestas claras y precisas a las preguntas guía.	5	4	3	2	0
1.3	Uso de ejemplos o casos prácticos para apoyar las respuestas.	5	4	3	2	0
2. Mapa Conceptual Individual						
2.1	Organización: El mapa conceptual está bien estructurado y es fácil de seguir.	5	4	3	2	0
2.2	Contenido: Incluye todos los temas clave (herramientas de pruebas automatizadas, monitoreo y alertas, CI/CD).	5	4	3	2	0
2.3	Profundidad: Explica los conceptos con claridad y detalle.	5	4	3	2	0
2.4	Conexiones: Muestra relaciones claras y lógicas entre los conceptos.	5	4	3	2	0
2.5	Creatividad: Utiliza colores, imágenes o diseños que mejoran la comprensión.	5	4	3	2	0
Tota=(Suma/40) * 2:						

RETROALIMENTACIÓN

Aspecto a mejorar: centrado en la tarea Centrado en el proceso centrado en la autorregulación

Acciones de mejora: Para alumnos que **NO** realizan actividades en casa o que **No** tienen los materiales para realizar la práctica, se les proporciona una actividad alterna en clase para que se pongan al día, mientras los demás trabajan en algo más avanzado.

Evidencia: Elaboración de la actividad o corrección de la práctica.

Puntaje al 80%:

Firma del Alumno (a)

Sello o firma Docente



Subodulo III. Implementa software de sistemas informáticos

Docente: MTE. Lizbeth Alcántara Blas

Semestre: Segundo

Academia:

6. Rúbrica: Documentación del sistema

Valor: 3.0

Heteroevaluación

Alumno (a):	Grupo: 203		N.L.:	
	Nivel de desempeño			
	Excelente (5)	Bueno (4)	Regular (3)	Deficiente (2)
Descripción de criterios				
1. Planificación				
1.1. Distribución equitativa y clara de tareas entre los integrantes.				
1.2. Definición de roles y responsabilidades para cada tarea.				
1.3. Establecimiento de un cronograma realista para la ejecución del proyecto.				
1.4. Identificación de recursos y herramientas necesarias para el proyecto.				
2. Configuración de Visual Studio Code				
2.1. Instalación y configuración de extensiones útiles para el proyecto.				
2.2. Configuración del editor para seguir estándares de codificación.				
2.3. Implementación de herramientas de formato automático.				
3. Documentación Técnica				
3.1. Archivo README				
3.1.1. Descripción clara y completa del proyecto.				
3.1.2. Instrucciones de instalación y configuración del entorno.				
3.2. Manuales Técnicos y Guías de Referencia				
3.2.1. Explicación detallada de los módulos, clases o funciones.				
3.2.2. Inclusión de ejemplos prácticos y casos de uso.				
3.3. Diagramas de Flujo y Arquitectura				
3.3.1. Diagramas claros y bien estructurados que representan el flujo del sistema.				
3.3.2. Representación adecuada de la arquitectura del proyecto.				
3.4. Descripción General, Parámetros y Retornos				
3.4.1. Explicación clara del propósito de cada módulo, clase o función.				
3.4.2. Detalle preciso de los parámetros de entrada y salida.				
3.5. Ejemplos de Uso y Notas Adicionales				
3.5.1. Ejemplos prácticos y bien documentados de cómo utilizar el código.				
3.5.2. Inclusión de notas adicionales (limitaciones, dependencias, etc.).				
Total(suma/85)*3 :				

RETROALIMENTACIÓN

Aspecto a mejorar: centrado en la tarea Centrado en el proceso centrado en la autorregulación

Acciones de mejora: Corregir diseño del sistema basandose en las observaciones y comentarios.

Periodo a completar: La próxima sesión, fecha: _____

Evidencia: Diseño del sistema corregido.

Puntaje al 80%:

Firma del Alumno (a)

Sello o firma Docente



Subódulo III. Implementa software de sistemas informáticos

Docente: MTE. Lizbeth Alcántara Blas

Semestre: Segundo

Academia:

6. Rúbrica: Documentación del sistema

Valor: 3.0

Heteroevaluación

Alumno (a):		Grupo: 203		N.L.:	
Descripción de criterios	Nivel de desempeño				
	Excelente (5)	Bueno (4)	Regular (3)	Deficiente (2)	
1. Estructura del Manual					
1.1. Portada: Incluye título, autores, fecha y logotipo (si aplica).					
1.2. Índice: Organizado y con enlaces (si es digital).					
1.3. Introducción: Explica generalidades y resume el contenido.					
1.4. Desarrollo: Incluye requisitos, configuración, tutoriales y ejemplos.					
1.5. Conclusiones: Reflexiones claras sobre el proceso.					
1.6. Anexos: Glosario, referencias y otros elementos adicionales.					
2. Formato					
2.1. Lenguaje: Claro, sencillo y adecuado para el público objetivo.					
2.2. Imágenes, diagramas y tablas: Facilitan la comprensión del contenido.					
2.3. Interfaz amigable: Diseño atractivo y fácil de navegar (si es digital).					
3. Investigación					
3.1. Identificación de dudas comunes: Basada en necesidades reales de los usuarios.					
3.2. Definición del público objetivo: Técnicos, usuarios finales o desarrolladores.					
4. Diseño					
4.1. Tutorial paso a paso: Incluye capturas de pantalla y explicaciones claras.					
4.2. Consistencia visual: Uso de colores, tipografías y estilos uniformes.					
4.3. Iconos e ilustraciones: Guían al usuario de manera efectiva.					
5. Desarrollo					
5.1. Precisión técnica: Contenido técnicamente correcto y actualizado.					
5.2. Tabla de compatibilidad: Incluye versiones de Python y sus requisitos.					
6. Revisión Cruzada y Publicación					
6.1. Revisión cruzada: Sugerencias claras y constructivas para mejorar.					
6.2. Corrección de errores: Errores técnicos y de redacción corregidos.					
Total(suma/90)*3 :					
RETROALIMENTACIÓN					
Aspecto a mejorar: <input type="checkbox"/> centrado en la tarea <input type="checkbox"/> Centrado en el proceso <input type="checkbox"/> centrado en la autorregulación Acciones de mejora: Corregir diseño del sistema basandose en las observaciones y comentarios. Periodo a completar: La próxima sesión, fecha: _____ Evidencia: Diseño del sistema corregido. Puntaje al 80%:					

Firma del Alumno (a)

Sello o firma Docente