

Consigna para el Proyecto Final

1. Logro a evaluar:

Al finalizar el curso el estudiante construye soluciones informáticas a problemas específicos utilizando las tecnologías aprendidas hasta el momento, las cuales definirán el alcance de la solución elaborada.

2. Indicación General:

Durante la semana 18, el proyecto final consistirá en aplicar todos los conocimientos adquiridos para construir una solución informática completa a un problema específico. Deberán seguir las instrucciones proporcionadas en cada una de las etapas de avance del proyecto y asegurarse de cumplir con los criterios de evaluación establecidos en la rúbrica. Recuerda que esta evaluación ofrece flexibilidad.

3. Indicaciones específicas:

Consideraciones formales del informe:

- Pruebas de Software:
 - Definición de Pruebas de Software: Identificar los conceptos de Testing en el desarrollo de software.
 - Taller de Testing: Entrega de avance de desarrollo al 60% de los proyectos a desarrollar demostrando el uso de las pruebas de software; Retroalimentación.
- Pruebas de Seguridad:
 - Principales Pruebas de Seguridad: Identificar los conceptos y herramientas de apoyo a las pruebas de seguridad.
 - Taller de Pruebas de Seguridad: Entrega de avance al 70% de los proyectos a desarrollar con observaciones levantadas y reporte de pruebas de seguridad.
- Despliegue:

- Identificar los conceptos del despliegue de aplicaciones Java utilizando Maven y la configuración de servidores para el soporte de la aplicación desarrollada.
- Taller de Despliegue: Entrega de avance al 80% de los proyectos a desarrollar con observaciones levantadas y aplicación desplegada; Retroalimentación.
- Monitoreo:
 - Mejores prácticas para el monitoreo de aplicaciones, incluyendo logs, performance tools y los health tools.
 - Entrega de avance al 90% de los proyectos a desarrollar y elaboración del plan de monitoreo; Retroalimentación.
- Mantenimiento:
 - Mejores prácticas para el mantenimiento de aplicaciones, como cron jobs, backups y scripts.
 - Entrega de proyecto al 100% de los proyectos a desarrollar y elaboración del plan de mantenimiento; Retroalimentación.

Consideraciones para la sustentación oral:

- Preparar una presentación en PowerPoint o equivalente.
- Realizar una exposición oral coherente y fluida, explicando los aspectos más relevantes del proyecto, incluyendo la arquitectura, las pruebas realizadas, el proceso de despliegue y los planes de monitoreo y mantenimiento.
- Demostrar dominio de los conceptos y técnicas aplicadas en el proyecto, así como la capacidad para responder preguntas y comentarios del equipo docente.
- Todos los participantes deben participar en la exposición.
- La duración máxima es de aproximadamente 10 minutos.

Para el desarrollo del avance del proyecto, deben considerar los siguientes aspectos:

- Pruebas de Software y seguridad: Se califica que el estudiante demuestre integración de las pruebas de software y seguridad en el desarrollo de proyectos. Esto implica identificar los conceptos de Testing en el desarrollo de software, así como los conceptos y herramientas de apoyo a las pruebas de seguridad.
- Despliegue del proyecto: Se califica que el estudiante demuestre un conocimiento sólido en el despliegue de aplicaciones Java, haciendo uso de Maven y configurando servidores para garantizar el soporte adecuado de la aplicación desarrollada.

- Monitoreo del proyecto: Se califica que el estudiante demuestre un dominio de las mejores prácticas para el monitoreo de aplicaciones, abarcando el uso de logs, herramientas de rendimiento y herramientas de salud del sistema.
- Mantenimiento del proyecto: Se califica que el estudiante demuestre un dominio de las mejores prácticas para el mantenimiento de aplicaciones, lo cual incluye la implementación efectiva de cron jobs, backups y scripts relevantes.
- Construcción del producto final: Se califica que el estudiante construya una solución informática que cumpla con el alcance comprometido, tomando en cuenta los siguientes criterios: 1) completa (cubre el alcance comprometido), 2) coherente (la documentación y el código están alineados), 3) buenas prácticas (usa librerías adecuadas, patrones de diseño, software de control de versiones, entre otros) y 4) autoría (el código fue hecho por el estudiante o lo domina).
- Sustentación oral: Se califica la sustentación oral del estudiante, la misma que se presenta de manera organizada (siguen una secuencia lógica), en la que desarrolla ideas con coherencia y fluidez (de manera continua, sin pausas prolongadas). Además, demuestra dominio de la sección que sustenta (Analiza el contexto, Analiza alternativas de solución, Diseña la solución, Diseña prototipo, Construye el producto final), pues las vincula con los conceptos estudiados en el curso.

4. Recomendaciones:

- Designar un responsable de cada etapa del proyecto para garantizar su correcta ejecución y seguimiento.
- Mantener una comunicación constante para coordinar tareas y resolver posibles inconvenientes.
- Utilizar metodologías ágiles para la gestión del desarrollo del proyecto y ajustar el plan según sea necesario.
- Consulta la bibliografía recomendada (libros, artículos de revistas, sitios web relevantes, entre otros).
- Evita el plagio. El trabajo debe ser original, evita copiar información de cualquier fuente.
- Solicitar retroalimentación periódica al docente y realizar ajustes en función de las sugerencias recibidas.

5. Criterios de evaluación:

A continuación, podrás encontrar la rúbrica de evaluación con la que será evaluada la actividad. Recuerda que también puedes encontrarla en la plataforma virtual de aprendizaje. Asegúrate de leerla antes de realizar la actividad.

6. Anexos:

El trabajo debe incluir la siguiente información y estructura:

- Diagramas de Arquitectura: Se incluyen diagramas que representan la arquitectura del proyecto, incluyendo la estructura MVC, la implementación de patrones SOLID y la integración de capas DAO.
- Documentación de Código: Se adjunta la documentación técnica generada con herramientas como Javadoc, que describe la estructura del código y el funcionamiento de las principales clases y métodos.
- Informes de Pruebas: Se incluyen los informes de las pruebas de software y seguridad realizadas, así como las observaciones levantadas y los reportes correspondientes.
- Plan de Despliegue, Monitoreo y Mantenimiento: Se adjuntan los planes elaborados para el despliegue, monitoreo y mantenimiento de la aplicación desarrollada, que incluyen la configuración de servidores, las estrategias de monitoreo y las prácticas de mantenimiento a seguir.
- Referencias bibliográficas consultadas para la elaboración del avance de portafolio final.
- Cualquier otro material relevante que contribuya a la comprensión y evaluación de tu trabajo.

A tomar en cuenta en caso de plagio:

“Todo acto de copiar, intentarlo o dejar copiar, durante una prueba, examen, práctica, trabajo o cualquier asignación académica, usando tanto el medio físico como el electrónico, se encuentra normado en el Reglamento de Estudios y el Reglamento de Disciplina del Estudiante vigentes en el Portal de Transparencia y/o en el Portal del Estudiante”

RÚBRICA DE PROYECTO FINAL

Criterio	Definición de criterio	Estándar Esperado	En Proceso 2	En Proceso 1	Inicial
Pruebas de Software y seguridad	Se califica que el estudiante demuestre integración de las pruebas de software y seguridad en el desarrollo de proyectos. Esto implica identificar los conceptos de Testing en el desarrollo de software, así como los conceptos y herramientas de apoyo a las pruebas de seguridad.	Se califica que el estudiante demuestre integración de las pruebas de software y seguridad en el desarrollo de proyectos. Esto implica identificar los conceptos de Testing en el desarrollo de software, así como los conceptos y herramientas de apoyo a las pruebas de seguridad. En el Taller de Testing, se espera que el estudiante entregue avances al 60% de los proyectos, demostrando el uso efectivo de las pruebas de software y recibiendo retroalimentación. En el Taller de Pruebas de Seguridad, se espera una entrega de avances al 70% de los proyectos, junto con observaciones levantadas y un reporte detallado de las pruebas de seguridad realizadas.	Se califica que el estudiante demuestre integración de las pruebas de software y seguridad en el desarrollo de proyectos. Esto implica identificar los conceptos de Testing en el desarrollo de software, así como los conceptos y herramientas de apoyo a las pruebas de seguridad. En el Taller de Testing, se espera que el estudiante entregue avances al 40% de los proyectos, demostrando el uso efectivo de las pruebas de software y recibiendo retroalimentación. En el Taller de Pruebas de Seguridad, se espera una entrega de avances al 60% de los proyectos, junto con observaciones levantadas y un reporte detallado de las pruebas de seguridad realizadas.	Se califica que el estudiante demuestre integración de las pruebas de software y seguridad en el desarrollo de proyectos. Esto implica identificar los conceptos de Testing en el desarrollo de software, así como los conceptos y herramientas de apoyo a las pruebas de seguridad. En el Taller de Testing, se espera que el estudiante entregue avances al 25% de los proyectos, demostrando el uso efectivo de las pruebas de software y recibiendo retroalimentación. En el Taller de Pruebas de Seguridad, se espera una entrega de avances al 50% de los proyectos, junto con observaciones levantadas y un reporte detallado de las pruebas de seguridad realizadas.	Se califica que el estudiante demuestre integración de las pruebas de software y seguridad en el desarrollo de proyectos. Esto implica identificar los conceptos de Testing en el desarrollo de software, así como los conceptos y herramientas de apoyo a las pruebas de seguridad. En el Taller de Testing, se espera que el estudiante entregue avances al 10% de los proyectos, demostrando el uso efectivo de las pruebas de software y recibiendo retroalimentación. En el Taller de Pruebas de Seguridad, se espera una entrega de avances al 40% de los proyectos, junto con observaciones levantadas y un reporte detallado de las pruebas de seguridad realizadas.
		2	1.5	1	0.5
Despliegue del proyecto	Se califica que el estudiante demuestre un conocimiento	Se califica que el estudiante demuestre un conocimiento sólido en el	Se califica que el estudiante demuestre un conocimiento sólido en el despliegue de	Se califica que el estudiante demuestre un conocimiento sólido en el despliegue de	Se califica que el estudiante demuestre un conocimiento sólido en el

	sólido en el despliegue de aplicaciones Java, haciendo uso de Maven y configurando servidores para garantizar el soporte adecuado de la aplicación desarrollada.	despliegue de aplicaciones Java, haciendo uso de Maven y configurando servidores para garantizar el soporte adecuado de la aplicación desarrollada. En el Taller de Despliegue, se espera que el estudiante entregue avances al 80% de los proyectos asignados, desplegando la aplicación de manera efectiva y recibiendo retroalimentación, además de aplicar observaciones levantadas para mejorar el proceso de despliegue.	aplicaciones Java, haciendo uso de Maven y configurando servidores para garantizar el soporte adecuado de la aplicación desarrollada. En el Taller de Despliegue, se espera que el estudiante entregue avances al 60% de los proyectos asignados, desplegando la aplicación de manera efectiva y recibiendo retroalimentación, además de aplicar observaciones levantadas para mejorar el proceso de despliegue.	aplicaciones Java, haciendo uso de Maven y configurando servidores para garantizar el soporte adecuado de la aplicación desarrollada. En el Taller de Despliegue, se espera que el estudiante entregue avances al 40% de los proyectos asignados, desplegando la aplicación de manera efectiva y recibiendo retroalimentación, además de aplicar observaciones levantadas para mejorar el proceso de despliegue.	despliegue de aplicaciones Java, haciendo uso de Maven y configurando servidores para garantizar el soporte adecuado de la aplicación desarrollada. En el Taller de Despliegue, se espera que el estudiante entregue avances al 20% de los proyectos asignados, desplegando la aplicación de manera efectiva y recibiendo retroalimentación, además de aplicar observaciones levantadas para mejorar el proceso de despliegue.
		2	1.5	1	0.5
Monitoreo del proyecto	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el monitoreo de aplicaciones, abarcando el uso de logs, herramientas de rendimiento y herramientas de salud del sistema.	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el monitoreo de aplicaciones, abarcando el uso de logs, herramientas de rendimiento y herramientas de salud del sistema. En el contexto del Taller de Monitoreo, se espera que el estudiante entregue avances al 90% de los proyectos asignados y elabore un plan de monitoreo exhaustivo.	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el monitoreo de aplicaciones, abarcando el uso de logs, herramientas de rendimiento y herramientas de salud del sistema. En el contexto del Taller de Monitoreo, se espera que el estudiante entregue avances al 70% de los proyectos asignados y elabore un plan de monitoreo exhaustivo.	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el monitoreo de aplicaciones, abarcando el uso de logs, herramientas de rendimiento y herramientas de salud del sistema. En el contexto del Taller de Monitoreo, se espera que el estudiante entregue avances al 50% de los proyectos asignados y elabore un plan de monitoreo exhaustivo.	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el monitoreo de aplicaciones, abarcando el uso de logs, herramientas de rendimiento y herramientas de salud del sistema. En el contexto del Taller de Monitoreo, se espera que el estudiante entregue avances al 30% de los proyectos asignados y elabore un plan de monitoreo exhaustivo.

		6	4	2	1
Mantenimiento del proyecto	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el mantenimiento de aplicaciones, lo cual incluye la implementación efectiva de cron jobs, backups y scripts relevantes.	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el mantenimiento de aplicaciones, lo cual incluye la implementación efectiva de cron jobs, backups y scripts relevantes. En el contexto del Taller de Mantenimiento, se espera que el estudiante entregue los proyectos completos al 100% de su desarrollo y elabore un plan de mantenimiento integral.	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el mantenimiento de aplicaciones, lo cual incluye la implementación efectiva de cron jobs, backups y scripts relevantes. En el contexto del Taller de Mantenimiento, se espera que el estudiante entregue los proyectos completos al 80% de su desarrollo y elabore un plan de mantenimiento integral.	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el mantenimiento de aplicaciones, lo cual incluye la implementación efectiva de cron jobs, backups y scripts relevantes. En el contexto del Taller de Mantenimiento, se espera que el estudiante entregue los proyectos completos al 60% de su desarrollo y elabore un plan de mantenimiento integral.	Se califica que el estudiante demuestre un dominio de las mejores prácticas para el mantenimiento de aplicaciones, lo cual incluye la implementación efectiva de cron jobs, backups y scripts relevantes. En el contexto del Taller de Mantenimiento, se espera que el estudiante entregue los proyectos completos al 30% de su desarrollo y elabore un plan de mantenimiento integral.
		5	3	2	0.5
Construcción del producto final	Se califica que el estudiante construya una solución informática que cumpla con el alcance comprometido, tomando en cuenta los siguientes criterios: 1) completa (cubre el alcance comprometido), 2) coherente (la documentación y el código están alineados), 3) buenas prácticas (usa librerías adecuadas, patrones de diseño, software de control de versiones, entre otros) y 4) autoría (el código fue hecho por el estudiante o lo domina).	El estudiante construye una solución informática considerando todos los siguientes criterios: 1) completa (cubre el alcance comprometido), 2) coherente (la documentación y el código están alineados), 3) buenas prácticas (usa librerías adecuadas, patrones de diseño, software de control de versiones, entre otros) y 4) autoría (el código fue hecho por el estudiante o lo domina).	El estudiante construye una solución informática considerando solo 3 de siguientes criterios: 1) completa (cubre el alcance comprometido), 2) coherente (la documentación y el código están alineados), 3) buenas prácticas (usa librerías adecuadas, patrones de diseño, software de control de versiones, entre otros) y 4) autoría (el código fue hecho por el estudiante o lo domina).	El estudiante construye una solución informática considerando solo 2 de siguientes criterios: 1) completa (cubre el alcance comprometido), 2) coherente (la documentación y el código están alineados), 3) buenas prácticas (usa librerías adecuadas, patrones de diseño, software de control de versiones, entre otros) y 4) autoría (el código fue hecho por el estudiante o lo domina).	El estudiante construye una solución informática considerando sólo uno de los siguientes criterios: 1) completa (cubre el alcance comprometido), 2) coherente (la documentación y el código están alineados), 3) buenas prácticas (usa librerías adecuadas, patrones de diseño, software de control de versiones, entre otros) y 4) autoría (el código fue hecho por el estudiante o lo domina).
		3	2	1	0.5

Sustentación oral	Se califica la sustentación oral del estudiante, la misma que se presenta de manera organizada (siguen una secuencia lógica), en la que desarrolla ideas con coherencia y fluidez (de manera continua, sin pausas prolongadas). Además, demuestra dominio de la sección que sustenta (Analiza el contexto, Analiza alternativas de solución, Diseña la solución, Diseña prototipo, Construye el producto final), pues las vincula con los conceptos estudiados en el curso.	Las ideas siguen una secuencia lógica acorde a la estructura de presentación establecida. Además, las ideas se desarrollan de manera continua, sin pausas (fluidez) y son coherentes con el desarrollo del tema. Asimismo, demuestra dominio de la sección que sustenta, pues vincula sus ideas con los conceptos abordados en el curso.	Las ideas siguen una secuencia lógica acorde a la estructura de presentación establecida. Además, las ideas se desarrollan de manera continua, sin pausas (fluidez) y son coherentes con el desarrollo del tema. Sin embargo, no demuestra dominio de la sección que sustenta, pues no vincula sus ideas con los conceptos abordados en el curso.	Las ideas siguen una secuencia lógica acorde a la estructura de presentación establecida. Además, las ideas se desarrollan de manera continua, sin pausas (fluidez) pero son poco coherentes con el desarrollo del tema (hay ciertos vacíos de información y por momentos se desvía del tema). Asimismo, no demuestra dominio de la sección que sustenta, pues no vincula sus ideas con los conceptos abordados en el curso.	Las ideas siguen una secuencia lógica acorde a la estructura de presentación establecida. Sin embargo, las ideas se desarrollan de manera discontinua, hay pausas o interrupciones. Las ideas son poco coherentes con el desarrollo del tema (hay ciertos vacíos de información y por momentos se desvía del tema). Asimismo, no demuestra dominio de la sección que sustenta, pues no vincula sus ideas con los conceptos abordados en el curso.
		2	1.5	1	0.5