

Actividad de Reflexión Inicial

Título: *¿Y si tu software fuera un puente? La calidad no se improvisa*

Descripción de la actividad

Imaginen que han sido contratados para desarrollar un sistema de reservas para un hospital rural con recursos limitados. El software se entregó según lo acordado, pero una vez en funcionamiento surgieron problemas graves:

- La interfaz está en inglés, pero los usuarios son adultos mayores que solo leen español.
- El sistema se bloquea con más de 10 usuarios simultáneos.
- No puede usarse desde celular, y los médicos solo tienen datos móviles.
- No se documentaron ni anticiparon estos fallos en ninguna parte del proceso.

Ante estas fallas, la empresa contratante amenaza con cancelar el contrato y denunciar por negligencia profesional. Frente a esta situación, deberán asumir el rol de profesionales comprometidos con la calidad del software y reflexionar sobre sus prácticas, criterios éticos y compromisos.

Desarrollo de la actividad

1. Análisis individual escrito

Redacten de forma personal, en su cuaderno o archivo digital, respuestas argumentadas a las siguientes preguntas:

- ¿Cuál fue, a tu juicio, la falla más crítica del proyecto y por qué?
- ¿Qué evidencia demostraría que el equipo no trabajó con control de calidad?
- ¿Qué acciones propondrías para evitar estos errores en futuros proyectos?
- ¿Qué responsabilidad ética tienen los desarrolladores en este tipo de contextos?

Importante: Las respuestas deben referirse a condiciones reales de su entorno (por ejemplo, características de hospitales rurales locales, experiencias propias o casos conocidos). No se aceptan textos generados por IA. Durante la sesión, podré verificar sus respuestas a través de preguntas espontáneas y retroalimentación en grupo.

2. Trabajo colaborativo – “Decálogo del software responsable”

Formen equipos de tres personas y redacten un *decálogo de compromisos* para garantizar el desarrollo ético y de calidad del software en contextos sensibles como el hospital descrito.

- Denle un **título creativo** al decálogo.
- Incluyan una **metáfora visual** que represente el desarrollo de software (por ejemplo: un puente, una receta, una cirugía).
- Incluyan compromisos claros en relación con: control de calidad, pruebas, documentación, validación de usuarios, y responsabilidad social.

La presentación final debe realizarse en cartel o formato digital personalizado, con diseño gráfico o dibujo que refleje la metáfora elegida. La entrega debe estar **firmada por todos los integrantes** del grupo.

3. Galería crítica y debate

Cada grupo presentará su decálogo frente a sus compañeros y defenderá sus ideas ante una ronda de preguntas abiertas.

Promoveremos un espacio de análisis colectivo donde se destaquen argumentos sólidos, compromiso ético y creatividad técnica. Al final, reflexionaremos sobre los aprendizajes obtenidos a partir del trabajo de los demás.

4. Reflexión personal final

Escriban de manera personal, conectando su experiencia durante la actividad:

- ¿Qué aprendizajes significativos me deja esta experiencia?
- ¿Qué relación puedo establecer entre la calidad del software y el impacto social?
- ¿Qué compromisos personales asumo para mejorar mis prácticas como futuro tecnólogo en desarrollo?

Nota: Esta reflexión será revisada por su conexión emocional, argumentativa y ética. No se aceptarán respuestas copiadas, ni impersonales.

Material de apoyo sugerido

- Lectura breve: “¿Qué pasa cuando falla el software?” (fragmento adaptado por el instructor, ver abajo)

Fragmento adaptado: ¿Qué pasa cuando falla el software?

Por: un desarrollador con cicatrices de código

“Cuando falla una app de domicilios, un usuario se molesta. Pero cuando falla un sistema médico, alguien puede perder la vida. He aprendido que desarrollar software no es solo picar código o cumplir fechas. Es asumir que, del otro lado, hay personas reales confiando en que haremos bien nuestro trabajo.

La calidad no es un añadido; es el puente invisible que permite cruzar del diseño a la confianza. Un puente sin vigas puede parecer listo... hasta que se cae. Cada vez que omites una prueba, que no documentas un error o que asumes que el usuario entenderá algo 'porque a ti te parece obvio', estás poniendo una piedra floja en ese puente.

Me ha tocado ver proyectos cancelados, datos perdidos y gente frustrada. He aprendido que no hay código tan brillante que justifique la negligencia. Hoy, mi primer paso antes de programar no es abrir el editor, es pensar: *¿para quién estoy haciendo esto? ¿y qué pasará si falla?*"