

Ingeniería en Sistemas Computacionales

Asignatura:	Tópicos de calidad para el diseño de software			
Grupo:	3A		Fecha:	04/dic/2025
Tipo de Evaluación (Marcar con una "X" según corresponda)	Diagnóstica		Unidad:	Calificación
	Formativa			
	Sumativa	X	Tiempo (minutos):	III 70
	Recuperación			
Nombre del profesor:	Mitsiu Alejandro Carreño Sarabia			

Nombre del alumno: Nataly Guzmán Romo Matrícula: UP240436

Nombre del alumno: Melani Yoselin Castañeda Hernandez Matrícula: UP240219

Nombre del alumno: Gabriela Negrete Álvarez Matrícula: UP240273

Nombre del alumno: Ivanna Danae Recendez Ovalle Matrícula: UP240223

Esta evidencia equivale al 70% de la calificación final en la unidad 3.

Cumple las siguientes instrucciones en el repositorio de git provisto.

1. Todos los colaboradores del equipo deben crear su propia rama de trabajo y subir commits en ella.
2. Las ramas deben someterse a un proceso de peer-review, al menos debe haber un comentario por rama buscando mejorar la legibilidad, calidad, estructura o lógica del código.
3. Una vez que la rama sea aprobada debe ser integrada a la rama principal del equipo.
4. La rama principal del equipo siempre debe cumplir con los requisitos establecidos por checkstyle.
5. El proyecto debe incluir al menos 5 pruebas unitarias.

Criterio	Descripción	Sí/No	Observaciones	Valor (pts)
1. Ramas de trabajo				25
Nombre de rama	Las ramas de trabajo tienen nombres claros y descriptivos.	10		10

Mensajes de commit	Cada commit está debidamente explicado a través de mensajes claros y descriptivos.	10		10
Origen de las ramas	Las ramas de trabajo parten siempre de la rama principal del equipo	5		5
2. Peer-review				25
Peer-review	Todas las ramas son sometidas a un proceso de revisión en el que al menos se deja un comentario	10		10
Criterio de aceptación	Las ramas no se integran hasta que al menos un colaborador acepta los cambios.	10		10
Merge en rama principal	Las ramas de trabajo se integran a la rama principal del equipo	5		5
3. Conflictos				30
Resolución de conflictos	Los conflictos deben resolverse en las ramas de trabajo y la rama principal del equipo siempre debe ser estable.	30		30
4. Checkstyle				10
Seguir convención de checkstyle	La rama principal del equipo siempre debe respetar las convenciones de checkstyle	0	\$ java -jar checkstyle-12.1.1-all.jar -c google_checks.xml *.java Muestra 179 warnings	10
5. Pruebas unitarias				10
Desarrollo de pruebas unitarias	El proyecto debe contar con al menos 5 pruebas unitarias.	4	modificarTarea_modificaCorrectamente() ✗ Unimplemented method 'modificarTarea' modificarTarea_rechazaNúmeroFueraDeRango() ✗ Unimplemented method 'modificarTarea' Esta prueba: https://github.com/ineptos-de-lira/Equipo_7/blob/main/ToDoListAppTest.java#L68 intenta llamar el método "eliminarTarea" que no existe	10
Calificación General	Puntaje Total:	84 / 100	Comentarios Adicionales:	

