

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN
DISEÑO DE SOFTWARE
TAREA DE PRINCIPIOS SOLID

Objetivos Específicos

1. Diseñar un producto de software aplicando principios de diseño orientado a objetos para que sea robusto, mantenible y escalable.
2. Emplear herramientas informáticas en el control de versiones para la generación de software de calidad en un entorno colaborativo.

Resultado de Aprendizaje

1. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas para la disciplina del programa.
2. Habilidad para aplicar teoría de ciencias computacionales y fundamentos de desarrollo de software para producir soluciones basadas en computación.

Descripción

En equipos de trabajo, conformados por entre tres a cinco estudiantes, se debe (i) seleccionar no más de 3 repositorios públicos en GitHub con código fuente en Java [1] a fin de (ii) rediseñar partes del código fuente en donde sean identificables 3 violaciones a cada principio de diseño SOLID.

Especificaciones

Sección A

Seleccione **no más de 3 repositorios públicos en GitHub, creados al menos un año antes de la fecha de publicación de esta tarea**, en los que pueda identificar violaciones a principios SOLID. Se exceptúan aquellos repositorios que constan en la Lista Negra adjunta. Un repositorio (**incluyendo cualquier fork relacionado**) solo puede ser analizado por un equipo. Por lo tanto, un integrante del equipo debe reportar su reserva en el foro Aula Virtual titulado “Reserva de repositorios Tarea SOLID”. Solo puede haber un máximo de 3 participaciones por equipo en el foro. **Antes de intentar reservar un repositorio, asegúrese que este o algún fork relacionado no haya sido reservado previamente.** [10%]

Sección B

En los repositorios seleccionados, identifique 3 violaciones a cada uno de los principios SOLID. Justifique su argumentación. Además del texto, puede hacer uso de diagramas UML. [45%]

Sección C

Proponga una solución que aplique correctamente cada principio SOLID que se haya violado. Su solución debe ser reportada como incidente en GitHub y debe contener el código fuente propuesto y diagramas UML. [45%]

Entregables

1. Un **documento**, en formato **pdf**, que identifique los integrantes del equipo en la primera página y tenga un índice de contenido en la segunda. El documento debe identificar los repositorios seleccionados, contener su justificación que argumente cada violación de los principios SOLID. Puede hacer uso de cualquier asunción razonable. Se debe incluir capturas de pantalla del código fuente inicial y su estado posterior luego del rediseño. Utilice diagramas UML para ilustrar su solución. Finalmente, el documento debe contener alguna evidencia del reporte de su sugerencia al propietario del repositorio pertinente.

Rúbrica de Calificación

Descripción	Valor
Sección A	
Selección de repositorio(s) único(s)	10
Sección B	
Identificación y justificación de violaciones a principio SRP	09
Identificación y justificación de violaciones a principio OCP	09
Identificación y justificación de violaciones a principio LSP	09
Identificación y justificación de violaciones a principio ISP	09
Identificación y justificación de violaciones a principio DIP	09
Sección C	
Rediseño aplicando principio SRP, incluyendo diagramas UML y reporte con solución al propietario del repositorio	09
Rediseño aplicando principio OCP incluyendo diagramas UML y reporte con solución al propietario del repositorio	09
Rediseño aplicando principio LSP incluyendo diagramas UML y reporte con solución al propietario del repositorio	09
Rediseño aplicando principio ISP incluyendo diagramas UML y reporte con solución al propietario del repositorio	09
Rediseño aplicando principio DIP incluyendo diagramas UML y reporte con solución al propietario del repositorio	09
Total	100
No subir a Aula Virtual los entregables de acuerdo con lo especificado (máximo -50)	-10

Late Submission Policy

Delay (§)	Penalty (Ω)
1 hour or less	loss of 10%
1 to 12 hours	loss of 20%
12 to 48 hours	loss of 30%
Over 48 hours:	loss of 100%

(§) every clock hour counts including weekends or holidays

(Ω) automatic and non-negotiable penalty

Referencias

1. <https://github.com/topics/java>