UNIVERSIDAD POPULAR DEL CESAR
ESPECIALIZACION DE INGENIEIRA DE SOFTWARE
GUÍA DE ACTIVIDADES PARA LA CONSTRUCCIÓN DE CONOCIMIENTO

NOMBRE DE LA ASIGNATURA	Patrones De Diseño de Software							
NOMBRE DE LA ACTIVIDAD	Introducción a UML y Diseño Orientado a Objetos							
TIPO DE ACTIVIDAD	Sincrónica		Asincrónica	x	Individual	x	Grupal	

TEMÁTICA REQUERIDA PARA LA ACTIVIDAD	**OBJETIVOS**
Unidad 2. Introducción a patrones de diseño. ○ Principios universales de diseño de software ○ Principios SOLID	Evaluar la capacidad para refactorizar el diseño de una solución de software, implementando mejoras a partir de la aplicación de principios de diseño SOLID, con el propósito de obtener un diseño más eficiente y flexible.
COMPETENCIAS	**INSUMOS PARA EL DESARROLLO DE LA ACTIVIDAD / REFERENCIAS BIBLIOGRÁFICAS**
• Aplicación de principios de diseño SOLID	• Taller No 1 Elaborado. • Material educativo y material complementario de la asignatura "**Unidad 2.**" • Fuentes bibliográficas del módulo.

CONOCIMIENTOS PREVIOS REQUERIDOS

Conceptos fundamentales de POO, UML, POO en Java

ESPECIFICACIONES DE LA ACTIVIDAD

Requerimientos de la evaluación:

- Refactorización de diagrama de clases (nueva versión de diagrama de clases)
- Implementación de código
- Sustentación de cambios (justificación)

Cada grupo de estudiantes, a partir de los conocimientos adquiridos en su análisis sobre los principios universales de diseño y/o principios SOLID, deberá elaborar una nueva versión del diseño UML elaborado y del código desarrollado en el primer taller, incorporando las mejoras que considere necesarias, aplicando algunos de los principios SOLID.

Así mismo, deberán realizar un video explicando los principales cambios incorporados tanto en el UML y el Código, indicando a partir de qué (o cuales) principio(s) SOLID se generó dicho cambio.

Problema planteado:

A partir del desarrollo del Taller No 1, se solicita la revisión del diseño elaborado y entregado. Durante la revisión, se debe identificar aspectos que permitan mejorar del diseño inicial, en busca de mayor flexibilidad, eficiencia, desacoplamiento y extensibilidad. Así mismo, se debe realizar la refactorización del diseño, en la cual deberá incluir las mejoras correspondientes, a partir de los diferentes principios de diseño SOLID.

Del mismo modo, deberá implementar dichos cambios en código.

Aspectos que se deben tener en cuenta para la nueva versión de su solución:

- Es posible que en un corto tiempo se ofrezca un nuevo servicio de envío.
- Es posible que por eficiencia se requiera almacenar las guías en mapas (Map) o conjuntos (Set) y no en listas (List)
- Es posible que las tarifas y valor base de liquidación de los servicios puedan variar en un futuro
- Las que usted considere (las deberá explicar).

RECOMENDACIONES / OBSERVACIONES	Para el diseño UML del Diagrama de clases se sugiere utilizar cualquiera de las siguientes herramientas: StartUML, PlantUML, Draw.io, GenMyModel o Visual Paradigm. Puede utilizar el lenguaje de programación de su preferencia, preferiblemente Java, es opcional el desarrollo de interfaces Graficas de Usuario.

Elaboro: Ing. Jairo Seoanes, Msc Ingeniería de Sistemas y Computación