**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN**

**DISEÑO DE SOFTWARE**

**TAREA DE PRINCIPIOS SOLID y UML**

**Objetivos Específicos**

1. Aplicar principios SOLID para garantizar un diseño de software que sea robusto, flexible y fácilmente mantenible.
2. Utilizar diagramas UML para documentar y comunicar el diseño del sistema, especificando casos de uso, clases, y secuencias de interacciones entre componentes.
3. Generar código basado en los diagramas UML de clases y de secuencias, siguiendo el diseño planteado.

**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 de entre 3 a 4 estudiantes, se les asignará la descripción de un sistema para el cual deberán (i) aplicar los principios SOLID en el diseño de las clases del sistema, (ii) generar diagramas UML de Casos de uso, de Clases y de Secuencias, y (iii) escribir el código base a partir de los diagramas generados.

**Especificaciones**

**Sección A: Casos de Uso**

Diseñar el diagrama de Casos de Uso del sistema con al menos 2 refinamientos (includes, extends). Indique las asunciones tomadas para su proyecto. Luego, seleccionar 4 casos de uso que cubran las principales funcionalidades del sistema asignado y para cada caso de uso, detallar: **[20%]**

* Actores involucrados.
* Precondiciones necesarias para que el caso de uso pueda ejecutarse.
* Flujo de Eventos Principal que describe paso a paso la secuencia de acciones.
* Flujos Alternativos para gestionar variaciones en la secuencia.
* Postcondiciones que indican el estado esperado del sistema al completar el caso de uso.

**Sección B: Diagrama de Clases**

Aplicar los principios SOLID al definir las clases, interfaces y relaciones del diagrama de clase del sistema asignado. El diagrama debe representar los atributos, métodos, y asociaciones necesarias para soportar los casos de uso. Justificar la aplicación de cada principio SOLID en las clases y relaciones diseñadas. **[30%]**

**Sección C: Diagrama de Secuencias**

Diseñar los 4 diagramas de secuencia que representen la ejecución de los casos de uso seleccionados en la sección A, detallando los mensajes enviados entre objetos, la creación de instancias y las llamadas a métodos. Sin embargo, debe asegurarse de que los diagramas sean consistentes con el diagrama de clases y muestren cómo se comunican los objetos en respuesta a los eventos de usuario. **[40%]**

**Sección D: Generación de Código Base**

Basándose en los diagramas de clases y de secuencias, desarrollar un prototipo inicial del sistema en Java. El código debe incluir: **[10%]**

* Definición de las clases, interfaces y métodos reflejados en los diagramas.
* Implementación de la lógica mínima para los métodos, según los diagramas.
* Utilizar control de versiones para gestionar el código y registrar el progreso en un repositorio en GitHub.

**Entregables (en un archivo ZIP o RAR)**

1. Un documento (en formato DOCX y PDF) que contenga:

* Una portada con los nombres de los integrantes del equipo.
* Un índice de contenido.
* Los diagramas UML, las asunciones de su proyecto, la descripción de los Casos de Uso seleccionados y la justificación de la aplicación de los principios SOLID.

1. Los archivos de proyecto en Visual Paradigm con los diseños UML.
2. Código Java generado a partir de los diagramas UML, almacenado en un repositorio de GitHub, y evidencia del uso de control de versiones (Todos los integrantes deben aportar al proyecto de manera equitativa).

**Rúbrica de Calificación**

|  |  |
| --- | --- |
| **Descripción** | **Valor** |
| **Sección A** | |
| Diagrama de Casos de Uso (actores, relaciones y casos de uso) | 4 |
| Dos refinamientos de cada tipo (includes, extends) | 8 |
| Descripción de 4 casos de Uso (funcionalidades principales) | 8 |
| **Sección B** | |
| Diagrama de clases con atributos y métodos importantes | 10 |
| Relaciones entre clases usando nombre de la relación, multiplicidad y rol | 5 |
| Justificación de los principios SOLID para cada elemento del diagrama de clases | 15 |
| **Sección C** | |
| Cuatro diagramas de secuencia de los Casos de Uso seleccionados | 10 |
| Uso correcto de los mensajes entre clases | 20 |
| Consistencia con el diagrama de clases | 10 |
| **Sección D** | |
| Implementación del código base Java | 10 |
| **Total** | **100** |
| Si no entrega el documento | -20 |
| Si no entrega los diseños UML en Visual Paradigm | -20 |
| Si no hay un uso adecuado del repositorio en GitHub (Todos deben aportar) | -40 |

**Late Submission Policy**

|  |  |
| --- | --- |
| **Delay (§)** | **Penalty (Ω)** |
| 1 hour or less | loss of 10% |
| 1 to 6 hours | loss of 20% |
| 6 to 24 hours | loss of 30% |
| Over 24 hours: | loss of 100% |

  (§) every clock hour counts including weekends or holidays. (Ω) automatic and non-negotiable penalty.