



TRABAJO PRÁCTICO Nº 2

ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE MODELADO ORIENTADO A OBJETOS CON UML

1 Objetivos

- Reconocer la importancia de la documentación y del modelado en el proceso de desarrollo de software.
- Comprender el enfoque del Modelado Orientado a Objetos utilizando UML.
- Conocer técnicas y herramientas para modelar soluciones de software para los sistemas de información.

2 Consignas

- Trabajen colaborativamente en grupo de hasta 4 estudiantes.
- Considere que los requerimientos que plantea el usuario como tales pueden no ser los únicos. Analice detenidamente el planteo del problema en busca de requerimientos implícitos.
- Incluyan los requerimientos del sistema en la Especificación de Requerimientos de Software (ERS).
- Elaboren los diferentes diagramas utilizando UML.
- Incluya los diagramas en la ERS.
- Tenga en cuenta las características que debe reunir una ERS para que se considere "buena".

3 Conceptos Básicos

La ingeniería de software debe dar solución a problemas inmersos en la complejidad del ambiente en que se desenvuelven las organizaciones; donde los sistemas software deben considerar esa complejidad, según Grady Booch (1996) "... la *complejidad* de la que se habla parece ser una propiedad esencial de todos los sistemas de software de gran tamaño; con esencial quiere decirse que esa complejidad puede dominarse, pero nunca eliminarse."; esto ha forzado a los analistas de sistemas software a examinar alternativas para resolver los problemas, y encontrar otras formas más eficientes de manejar los procesos complejos. La tecnología Orientada a Objetos (en adelante OO) se ha convertido en uno de los motores clave de la industria del software para abordar la complejidad del software, implica la creación de modelos del mundo real y la construcción de software basados en esos modelos.

EL Paradigma Orientado a Objetos es una disciplina de ingeniería de desarrollo y modelado de software que permite construir más fácilmente sistemas complejos a partir de componentes individuales. Se basa en cuatro principios:

1. *Abstracción*,
2. *Encapsulamiento*, la
3. *Modularidad*
4. *Herencia*.

El desarrollo OO tiene fases genéricas para generar sistemas software, las cuales son: análisis OO, diseño OO, implementación OO, validación y evolución.

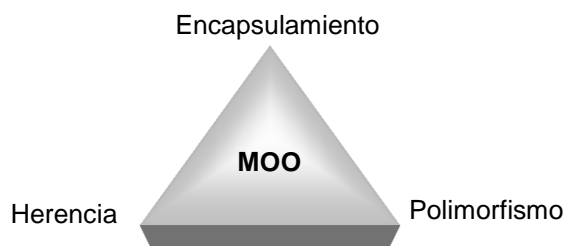
El *análisis* OO es un enfoque de la ingeniería de software que modela un sistema como un grupo de objetos que interactúan entre sí. Según Grady Booch (1996) es un método de análisis que examina los requerimientos desde la perspectiva de clases y objetos que se encuentran en el vocabulario del dominio del problema, de los objetos Booch dice "son

entidades tangibles que muestran un comportamiento bien definido". Esto quiere decir que el análisis orientado a objetos parte de entidades tangibles halladas en el problema, son elementos reales que forman parte del problema de forma directa. El modelado de análisis tiene por objetivo determinar *que necesita el cliente y modelar el comportamiento deseado*.

Conceptos Básicos

La *metodología orientada a objetos* modela la realidad tomando como concepto fundamental al objeto, combinando operaciones y datos en una sola entidad; los valores de los atributos del objeto no pueden manipularse directamente, sino a través del envío de mensajes del *objeto cliente* al *objeto servidor*; el cliente conoce qué hace el servidor, no cómo lo hace (*encapsulamiento*). Cuando existen objetos que poseen características comunes, éstas se describen en clases; si estas clases a su vez presentan similitudes, se puede definir una clase específica que contenga el comportamiento y las estructuras de información comunes a todas ellas, de manera que las clases originales sólo especifican su comportamiento exclusivo, heredando de la nueva clase el comportamiento común a todas ellas. Esta clasificación define una jerarquía de clases, donde las superclases (antecesoras) heredan a sus subclases (descendientes) estas características comunes (*herencia*). Los objetos invocan el comportamiento de otros objetos sin tener en cuenta la clase específica de los mismos (*polimorfismo*).

El *encapsulamiento*, la *herencia* y el *polimorfismo* son los tres conceptos clave que constituyen la génesis de los métodos.



4 Recursos

• Bibliográficos

- Larman, Craig: *UML y Patrones*. Prentice Hall. Madrid, 2003.
- O'Docherty, Mike: *Object-Oriented Analysis and Design – Understanding System Development with UML 2.0*. John Wiley & Sons Ltd. England, 2005.
- Ferré Grau, Xavier – Sánchez Segura, María Isabel: *Desarrollo Orientado a Objetos con UML*. Facultad de Informática – UPM
- Pressman, Roger S.: *Ingeniería del Software - Un enfoque práctico*. 7^{ma} Edición McGraw-Hill. Mexico, 2010.
- Sommerville, Ian : *Ingeniería del Software*. 7^a Edición. Pearson Educación. Madrid, 2005.
- IEEE *Recommended Practice for Software Requirements Specifications*. IEEE Std 830-1998 (Revision of IEEE Std 830-1993). U.S.A., 1998.

• Software

- Herramienta CASE: Ideas Modeler, ArgoUML u otra a elección.
- Archivo: IS1-2018-Plantilla-ERS-UML.dot (contiene la plantilla del documento de ERS). Disponible en el aula virtual.

5 Resultado Esperado

- El documento de ERS, preparado en equipo, siguiendo las "Prácticas Recomendadas" y el esquema que se presentan en el estándar IEEE Std 830-1998.
- El equipo deberá subir el documento de ERS al aula virtual hasta el **viernes 16/11/2018**.



6 Evaluación del Trabajo Práctico

| Concepto | Valor |
|---|-------|
| Formulación adecuada de los requerimientos del sistema | 1,50 |
| Grado de aplicación de los conceptos en el modelado de la solución del problema planteado | 3,00 |
| Aplicación de las herramientas de modelado. | 1,00 |
| Calidad del documento ERS de acuerdo con las normas establecidas en el estándar IEEE Std 830-1998. | 1,50 |
| Presentación escrita del documento ERS (deberá ser clara, ordenada, concisa y minuciosa en lo referente a errores ortográficos y gramaticales). | 1,00 |
| Presentación en tiempo y forma del documento ERS. | 1,00 |
| Trabajo colaborativo: participación de todos los miembros, respeto y colaboración. | 1,00 |

Escala de valoración: en todos los casos es de 0 (cero) a 10 (diez). El Trabajo Práctico se aprueba con 6,00 (seis).

7 Planteo del Problema

Ídem al planteo del Trabajo Práctico N° 1.