

Facultad de Informática y Estadística
Ingeniero en Informmática
Asignatura troncal: Ingeniería del Software II 4º Curso 1998-99

Profesores:

- Víctor Jesús Díaz Madrigal.
- Amador Durán Toro
- Francisco José Galán Morillo (coordinador).

Objetivos:

La asignatura cubrirá cinco objetivos:

1. Metodología y técnicas para diseñar sistemas software. El conocimiento de las técnicas se establecen dentro del marco que representa la metodología.
2. Cubrir aspectos de implementación de sistemas software. Conceptos de diseño traducidos a diferentes lenguajes ejecutables: modelo de objetos traducido a SQL y C++, implementación del control mediante procedimientos en C++, implementación de control de interfaz de usuario mediante eventos, implementación del modelo funcional entendido como colaboraciones entre objetos. Los aspectos no funcionales recogidos en la especificación de requisitos son recogidos en este momento y para ello será necesario "importar" las estructuras de datos y algoritmos vistos en cursos anteriores. La trazabilidad de conceptos queda aún más clara al alumnado al llegar a la implementación.
3. Establecer la importancia de la reutilización, tanto a nivel de diseño como de implementación. La extensión y adaptación de sistemas se verá beneficiada del uso de patrones en el diseño. Diferentes entornos de desarrollo utilizan frameworks para resolver la implementación de sistemas desde el control de la interfaz de usuario. El conocimiento y utilización de algún entorno de desarrollo con estas características se considera importante para cubrir estas necesidades.
4. Integración de los conceptos teóricos en un ejercicio práctico. Localizar la aplicación de todas las técnicas dentro de la metodología se considera necesario para comprender la asignatura.
5. Como último objetivo, introducir el diseño e implementación de sistemas con restricciones expresadas en OCL. La elección de OCL se debe a su simplicidad, precisión y ejecutabilidad. Destacar la importancia de las restricciones en los sistemas no sólo en régimen "estacionario" de ejecución sino incluso como asistencia en la inicialización de los mismos.

Temario:

1. Diseño de sistema: Descomposición en subsistemas, modelo de capas, el problema de la concurrencia "real", diseño de almacenes, recursos globales del sistema, elección del modelo de implementación del control, clasificación de distintas arquitecturas de sistemas y sus diseños asociados como punto de partida en nuestro diseño).
2. Diseño de objetos: Integración de los modelos de objeto, dinámico y funcional obtenidos desde el análisis, clasificación y diseño de algoritmos, optimizaciones en el diseño, ajuste de la herencia y uso de la delegación, diseño de asociaciones y sentidos de navegación.
3. Reutilización mediante Patrones de Diseño: reutilizar en la creación de objetos (creacionales), permitir reutilizar estructuras mediante adaptaciones y composiciones principalmente (patrones estructurales) y reutilizar y extender comportamientos (patrones de conducta). Arquitectura documento/vista. Reutilización usando MFC.
4. Importancia de las restricciones: sistemas anotados con OCL, su diseño e implementación. Reutilización mediante contrato (derechos y obligaciones). Ejecución de aserciones (monitorización de invariantes, ejecución de especificaciones pre/post, ejecuciones de guardas).

Metodología:

El desarrollo de la asignatura se fundamenta en:

- Fundamentos teórico-prácticos: Exposición del contenido temático de la asignatura mediante clases magistrales. Ejercicios prácticos como reforzamiento del contenido teórico.
- Ejercicio de curso: Los alumnos, en grupos reducidos, realizarán de manera opcional, el diseño de un sistema software partiendo de la especificación del mismo.

Este ejercicio se desarrollará durante el cuatrimestre y su entrega será al final del mismo para su evaluación.

Dicho ejercicio tendrá la tutela del profesor consistente en: a) validación de requisitos, b) asistencia y validación del análisis, c) asistencia y validación del diseño y d) a criterio del profesor asistencia en la implementación.

Evaluación:

Distinguiremos dos situaciones:

1. Alumnos que deciden hacer el ejercicio: $0.7 \cdot \text{nota de examen teórico} + 0.3 \cdot \text{nota del ejercicio de curso}$.

2. Alumnos que deciden no hacer el ejercicio de curso: nota del examen teorico.

Bibliografía Básica:

- James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy y William Lorensen. Modelado y Diseño Orientado a Objetos. Ed. Prentice Hall 1996.
- Ehrich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Design Patterns. Elements of Reusable Object-Oriented Software. Ed. Addison-Wesley 1995.

Bibliografía complementaria:

- Bertrand Meyer. Object-Oriented Software Construction. Prentice-Hall, 1988.
- J. Prosise. Programming Windows 95 with MFC. Microsoft Press, 1996.
- D. J. Kuglinski. Programación Avanzada con Visual C++. McGrawHill, 1996.