

Fundamentos de Ingeniería del Software

Código: 102708
Créditos ECTS: 3

Titulación	Tipo	Curso	Semestre
2500895 Ingeniería Electrónica de Telecomunicación	OB	1	2
2500898 Ingeniería de Sistemas de Telecomunicación	OB	1	2

Contacto

Nombre: Daniel Ponsa Mussarra
Correo electrónico: Daniel.Ponsa@uab.cat

Uso de idiomas

Lengua vehicular mayoritaria: catalán (cat)
Algún grupo íntegramente en inglés: No
Algún grupo íntegramente en catalán: Sí
Algún grupo íntegramente en español: No

Prerequisitos

La asignatura no tiene prerrequisitos.

La asignatura se sustenta en conocimientos básicos de programación en C. Hay que llevar al día pues el temario sobre programación en C dado en la asignatura Fundamentos de Informática.

Objetivos y contextualización

La asignatura corresponde a una materia básica de introducción a los fundamentos de la ingeniería del software, el análisis y diseño orientado a objeto, y la programación orientada a objeto.

Los objetivos formativos básicos de la asignatura son:

- Introducir al alumno en los conceptos clave de la ingeniería del software, viendo las singularidades y diferencias respecto la ingeniería civil.
- Introducir los conceptos clave respecto al análisis y diseño de software orientado a objeto.
- Aprender un lenguaje de programación orientado a objeto. Específicamente el C++.

Competencias

Ingeniería Electrónica de Telecomunicación

- Aprender nuevos métodos y tecnologías en base a sus conocimientos básicos y tecnológicos, con gran versatilidad de adaptación a nuevas situaciones.
- Hábitos de pensamiento
- Hábitos de trabajo personal
- Trabajo en equipo

Ingeniería de Sistemas de Telecomunicación

- Aprender nuevos métodos y tecnologías en base a sus conocimientos básicos y tecnológicos, con gran versatilidad de adaptación a nuevas situaciones.
- Hábitos de pensamiento
- Hábitos de trabajo personal
- Trabajo en equipo

Resultados de aprendizaje

1. Aplicar los fundamentos de la ingeniería del software al desarrollo de aplicaciones software.
2. Conocer y utilizar los fundamentos de la programación en redes, sistemas y servicios de telecomunicación
3. Desarrollar el pensamiento científico.
4. Desarrollar estrategias de aprendizaje autónomo.
5. Desarrollar la capacidad de análisis y de síntesis.
6. Evaluar de forma crítica el trabajo realizado.
7. Gestionar el tiempo y los recursos disponibles
8. Gestionar el tiempo y los recursos disponibles. Trabajar de forma organizada.
9. Trabajar cooperativamente.
10. Trabajar de forma autónoma.
11. Utilizar los fundamentos de la arquitectura y metodología de diseño, verificación y validación del software.

Contenido

A. INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE

A1. Introducción: Hay una ingeniería del software? conceptos Básicos

- Presentación de la asignatura.
- Ingeniería del software vs Ingeniería Civil.
- Etapas del ciclo de desarrollo del software.
- Procesos de desarrollo del software.
- Planificación de proyectos Software.

A2. Análisis y Diseño orientado a objetos

- Modularización efectiva de un software.
- Diseño orientado a objeto.
- UML: Diagrama de Clases.

B. PROGRAMACIÓN ORIENTADA A OBJETOS EN C++

B1. Clases, Atributos y Métodos

- Declaración y definición de los elementos básicos.
- Clase vs Objeto.
- Accesibilidad: atributos y métodos públicos y privados.
- Sobrecarga de funciones

B2. Enumerados. Mecanismo de llamadas a función. memoria Dinámica

- Enumerados.
- Mecanismo de llamadas a función.
- Funciones inline
- Paso por referencia / Paso por referencia constante.
- Reserva / Liberación de memoria.
- Aritmética de apuntadorss

B3. Constructores y Destrucción. operadores

- Tipo de constructores.
- Destrucción.
- Operadores binarios.
- El operador de asignación.
- Objetos y el apuntador this.
- El concepto friend.

- Streams. Operadores >> y <<.

B4. Herencia, Polimorfismo y Genericidad

- Jerarquía. Generalización vs Especificación.
- Accesibilidad a atributos y métodos 'protegidos'.
- Ligadura estática y dinámica.
- Funciones virtuales.
- Funciones template.
- Clases template.

Metodología

Las diferentes actividades que se llevarán a cabo en la asignatura se organizan de la siguiente manera:

Clases de teoría:

Se exponen los conceptos básicos de la asignatura y se dan indicaciones de cómo completar y profundizar en este contenidos. Se realizan actividades para poner en práctica los contenidos de la sesión. En la medida que los recursos de la escuela lo permitan, estas actividades se realizarán utilizando un entorno de programación.

Clases de problemas:

Se extienden de manera práctica temas vistos tangencialmente a las clases magistrales. Se resuelven problemas y se discuten casos prácticos. Con las actividades planteadas se promueve la capacidad de análisis y síntesis, el razonamiento crítico, y se entrena al estudiante en la resolución de problemas.

En la medida que los recursos de la escuela lo permitan, se promoverá la resolución de ejercicios propuestos utilizando un entorno de programación.

Prácticas:

Durante el curso se realizará una práctica a lo largo de distintas sesiones. Siempre que la capacidad de las aulas lo permita, los alumnos trabajarán en grupos de dos personas.

La práctica consistirá en el desarrollo de un programa en C ++ que implemente una aplicación solicitada.

Para desarrollar el programa los grupos trabajarán de forma autónoma. Las sesiones de prácticas se dedicarán principalmente a resolver dudas con el profesor, y entregar hitos del programa a desarrollar, para su posterior evaluación. Los grupos deberán explicar el trabajo hecho al profesor, y éste planteará cuestiones a los miembros del grupo para valorar la contribución de cada uno.

Consideraciones generales

Para diseminar información al estudiante se utilizará la plataforma Campus Virtual. Las fechas de evaluación continua y entrega de trabajos se publicarán a través de este medio, y pueden estar sujetos a posibles cambios de programación por motivos de adaptación a posibles incidencias. Siempre se informará mediante el Campus Virtual sobre estos posibles cambios ya que esta es la plataforma de intercambio de información entre el profesorado y el estudiantado.

Actividades

Título	Horas	ECTS	Resultados de aprendizaje
Tipo: Dirigidas			
Clases de problemas	6	0,24	1, 6, 3, 5, 10, 11
Clases de teoría	11,5	0,46	2, 3, 5, 11
Clases de prácticas	6	0,24	1, 6, 4, 5, 8, 9, 10, 11

Pruebas de evaluación	2,5	0,1	2, 3, 5, 8, 11
Tipo: Supervisadas			
Preparación de las sesiones de prácticas	15	0,6	1, 3, 7, 8, 9, 10
Resolución de problemas fuera de clase	9	0,36	1, 2, 4, 5, 7, 10
Tipo: Autónomas			
Estudio	23	0,92	2, 4, 10
Tutorías y consultas	2	0,08	4, 10

Evaluación

a) Proceso y actividades de evaluación programadas

La evaluación de la asignatura se llevará a cabo de manera continua a partir de las evidencias de aprendizaje recogidas en los siguientes procesos:

- [E1]. Realización de pruebas escritas (exámenes).
- [E2]. Resolución y entrega de cuestionarios y ejercicios planteados en las sesiones de teoría y problemas.
- [E3]. Realización de una práctica, evaluada a partir de diferentes actividades y entregas.

La asignatura consta de las siguientes actividades de evaluación, cada una evaluada con una nota entre 0 y 10 (ambos inclusive):

- [E1]-Exp1, examen parcial 1, 18% sobre la calificación final.
- [E1]-Exp2, examen parcial 2, 42% sobre la calificación final.
- [E2]-Prob, resolución de ejercicios planteados en las sesiones de teoría y problemas, 10% sobre la calificación final.
- [E3]-Prac, práctica de programación en C ++, 30% sobre la calificación final.

Para poder aprobar la asignatura mediante la evaluación continua, habrá sacar una nota mínima de 5 en las siguientes 2 expresiones.

- $(0,3 * \text{Nota [E1]-Exp1}) + (0,7 * \text{Nota [E1]-Exp2}) + (0,1 * \text{Nota [E2]-Prob})$
- $(0,18 * \text{Nota [E1]-Exp1}) + (0,42 * \text{Nota [E1]-Exp2}) + (0,1 * \text{Nota [E2]-Prob}) + (0,3 * \text{Nota [E3]-Prac})$

Hay que tener en cuenta que:

- Si no se supera la primera condición para aprobar, se asignará el resultado de su expresión como calificación final de la asignatura.
- los ejercicios que integran la actividad [E2]-Prob deberán entregarse dentro de un plazo establecido, y se evaluarán con una nota entre 0 y 10 (ambos inclusive). Los ejercicios no entregados dentro de su plazo se evaluarán con una nota de 0, y no se podrán recuperar.
- la actividad [E3]-Prac evaluará en base a diferentes subactividades planteadas, que tendrán un plazo de entrega establecido. Cada subactividad evaluará con una nota entre 0 y 10 (ambos inclusive). Las subactividades no realizadas o entregadas fuera de su plazo se evaluarán con una nota de 0, y no se podrán recuperar.

En caso de irregularidades en las actividades evaluativas, se aplicará lo que se detalla en el apartado f).

Es importante tener en cuenta que no se realizarán actividades de evaluación en una fecha o momento diferente al establecido, salvo causas justificadas, debidamente informadas con antelación al profesorado.

b) Programación de actividades de evaluación

El calendario de las diferentes actividades de evaluación se explicará en las clases de teoría. Las fechas de realización de las pruebas escritas (actividades [E1] -Exp1 y [E1] -Exp2) también se harán públicas en la web de la Escuela de Ingeniería, en el apartado de exámenes.

c) Proceso de recuperación

Las únicas actividades de evaluación recuperables son las pruebas escritas [E1]-Exp1 y [E1]-Exp2.

El/La estudiante puede presentarse a recuperar o mejorar las notas de estas pruebas (una de ellas, o ambas) siempre que se haya presentado a un conjunto de actividades que representen al menos dos terceras partes de la calificación total de la asignatura.

De cara a la calificación final de la asignatura, la nota que se obtenga en la recuperación sustituirá la nota de la prueba correspondiente realizada dentro de la evaluación continua.

De acuerdo con la coordinación del Grado y la dirección de la Escuela de Ingeniería las siguientes actividades no se podrán recuperar:

- [E2] -Prob, 10% sobre la calificación final.
- [E3] -Prac, 30% sobre la calificación final.

d) Procedimiento de revisión de las calificaciones

Para las actividades de evaluación basadas en pruebas escritas ([E1] -Exp1 y [E1] -Exp2) indicará un lugar, fecha y hora de revisión en la que el estudiante podrá revisar la actividad con el profesorado. En este contexto, se podrán hacer reclamaciones sobre la nota de la actividad, que serán evaluadas por el profesorado responsable de la asignatura. Asimismo, se podrá concertar con el profesorado responsable de la asignatura la revisión del resto de actividades de evaluación hasta dos semanas antes de la prueba de recuperación.

e) Calificaciones especiales

Si el alumno no ha realizado ninguna de las pruebas [E1] -Exp1 y [E1] -Exp2 se le asignará la nota de "No Evaluable". Cabe recordar pero que, según normativa vigente, las calificaciones de "No Evaluable" agotan también convocatoria.

Sólo podrán obtener una MH los estudiantes que tengan una nota igual o superior a 9 puntos. Como el número de MH no puede superar el 5% de los estudiantes matriculados, se concederán a los estudiantes que tengan las notas finales más altas. En caso de empate, los resultados de las pruebas parciales serán tomados en consideración.

f) Irregularidades por parte del estudiante, copia y plagio.

Sin perjuicio de otras medidas disciplinarias que se estimen oportunas, se calificarán con un cero las irregularidades cometidas por el estudiante que puedan conducir a una variación de la calificación de un acto de evaluación. Por lo tanto, la copia, el plagio, el engaño, dejar copiar, etc. en cualquiera de las actividades de evaluación implicará suspenderla con un cero.

g) Evaluación de los estudiantes repetidores

A partir de la segunda matrícula, el estudiante repetidor puede solicitar convalidar la evaluación de las actividades en [E3]-Prac, tomando la nota obtenida la primera vez que el estudiante se ha matriculado de la asignatura. Para poder optar a esta evaluación diferenciada, el estudiante repetidor debe pedir al profesorado mediante un correo electrónico.

Actividades de evaluación

Título	Peso	Horas	ECTS	Resultados de aprendizaje
--------	------	-------	------	---------------------------

Prueba de evaluación ExP1	18%	0	0	6, 2, 3, 4, 5, 7, 8, 10, 11
Prueba de evaluación ExP1	42%	0	0	1, 6, 2, 3, 4, 5, 7, 8, 10, 11
Práctica	30%	0	0	1, 2, 4, 7, 8, 9, 10, 11
Resolución de Problemas y Actividades Propuestas	10%	0	0	1, 6, 3, 4, 5, 10, 11

Bibliografía

- R.S. Pressman, Ingeniería del software, un enfoque práctico. Mc Grah-Hill, 7a. edición , 2010.
- A. Álvarez García, R. de las Heras del Dedo, C. Lasa Gómez, Métodos Ágiles y Scrum. Anaya Multimedia, 2012.
- T.A. Pender, UML Weekeng Crash Course, Wiley Publishing, Inc. 2002.
- P. Bustamante, I. Aguinaga, M. Aybar, L. Olaizola, I. Lazacano, Aprenda C++ Básico como si estuviera en primero, Universidad de Navarra, 2004.
- P. Bustamante, I. Aguinaga, M. Aybar, L. Olaizola, I. Lazacano, Aprenda C++ Avanzado como si estuviera en primero, Universidad de Navarra, 2004.