

Guía docente

330219 - TP - Tecnología de Programación

Última modificación: 25/04/2024

Unidad responsable: Escuela Politécnica Superior de Ingeniería de Manresa
Unidad que imparte: 750 - EMIT - Departamento de Ingeniería Minera, Industrial y TIC.

Titulación: GRADO EN INGENIERÍA DE SISTEMAS TIC (Plan 2010). (Asignatura obligatoria).

Curso: 2024 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: MARTA ISABEL TARRÉS PUERTAS

Otros:

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:

1. Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional así como su aplicación al tratamiento automático de la información mediante sistemas computacionales y la aplicación a la resolución de problemas propios de la ingeniería.
2. El conocimiento de las estructuras de datos más habituales y la capacidad de usarlas de forma adecuada en problemas reales. La capacidad de diseñar estructuras de datos específicos cuando los problemas así lo requieran.
3. La capacidad de analizar, diseñar y mantener aplicaciones informáticas así como el conocimiento de los principios y herramientas de la ingeniería del software y su aplicación.
4. El conocimiento y la capacidad de usar las herramientas e instrumentación existentes para el análisis, el diseño, el desarrollo y la verificación de sistemas electrónicos, informáticos y de comunicaciones.

Transversales:

5. COMUNICACIÓN EFICAZ ORAL Y ESCRITA - Nivel 1: Planificar la comunicación oral, responder de manera adecuada a las cuestiones formuladas y redactar textos de nivel básico con corrección ortográfica y gramatical.
7. APRENDIZAJE AUTÓNOMO - Nivel 1: Llevar a cabo tareas encomendadas en el tiempo previsto, trabajando con las fuentes de información indicadas, de acuerdo con las pautas marcadas por el profesorado.

METODOLOGÍAS DOCENTES

La asignatura se estructura en dos clases de dos horas por semana. De estas cuatro horas presenciales semanales una se dedica a presentar los principales contenidos de manera expositiva, la segunda a la resolución de problemas bajo demanda del estudiantado y las dos restantes a resolver problemas prácticos en el laboratorio informático.

Al estudiante se le indican semanalmente los trabajos de estudio y solución de problemas que son necesarios que haga. Estos trabajos se aconseja hacerlos, al menos parcialmente, trabajando en equipo. Periódicamente se evalúa el progreso de cada estudiante individualmente.

La asignatura también incorpora un proyecto de desarrollo de programario de una medida mediana que ha de trabajar en equipo.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Después de superar esta asignatura el estudiante ha de:

1. Saber planificar la comunicación oral, responder de manera adecuada a las cuestiones formuladas y redactar textos de nivel básico con corrección ortográfica y gramatical.
2. Saber identificar las propias necesidades de información y utilizar las colecciones, los espacios y los servicios disponibles para diseñar y ejecutar búsquedas simples adecuadas al ámbito temático.
3. Poder llevar a cabo los trabajos encomendados en el tiempo previsto, trabajando con las fuentes de información indicadas, de acuerdo con las pautas marcadas por el profesorado.
4. Poder aplicar los procedimientos algorítmicos fundamentales a la resolución de problemas haciendo uso de los lenguajes de alto nivel.
5. Conocer las estructuras de datos más habituales y poder usarlas de forma necesaria en problemas reales de complejidad moderada.
6. Estar capacitado para diseñar estructuras de datos específicos de complejidad mediana.
7. Conocer el concepto de complejidad computacional y ser capaz de calcular la complejidad en tiempo y espacio para el peor caso de algoritmos simples.
8. Conocer los principios y algunas herramientas de verificación y validación de programario y ser capaz de aplicarlos a problemas reales.
9. Poder redactar memorias técnicas sencillas, también en una tercera lengua, y presentarlas oralmente.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTE

Tipo	Horas	Porcentaje
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00
Horas grupo grande	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

TEMA 1. CLASES DE OBJETOS

Descripción:

Se inicia al estudiante en el uso de las clases de objetos como forma de organizar el código de las aplicaciones. Esto comporta abordar los aspectos sintácticos y semánticos de estas construcciones. Se presentan de forma muy tímida algunas cuestiones relacionadas con el diseño orientado a objetos y un subconjunto de la notación UML para expresar un diagrama de clases.

Palabras clave: Clase, método, constructor, herencia, redefinición, método estático, objeto, encajar.

Objetivos específicos:

Una vez logrado el tema:

1. El estudiante ha de ser capaz de usar clases de objetos prefabricados como los que ofrece la propia librería del lenguaje.
2. El estudiante ha de ser capaz de implementar clases de objetos a partir de una especificación incluyendo estructuras con relación de herencia.
3. El estudiante ha de ser capaz de diseñar una solución a un problema mediano utilizando las clases como elemento fundamental de diseño.
4. Un estudiante ha de poder entender un diagrama UML en el que participen los elementos básicos.
5. El estudiante ha de comprender los conceptos clave del tema. Esto comporta saberlos identificar (si es necesario) y comprende lecturas y reflexiones en que estos conceptos participen.

Actividades vinculadas:

Todas las que constan.

Dedicación: 45h

Grupo grande/Teoría: 10h

Grupo pequeño/Laboratorio: 10h

Aprendizaje autónomo: 25h

TEMA 2. RECURSIVIDAD

Descripción:

Se inicia al estudiante en el uso de la recursividad. El objetivo es que el estudiante entienda la recursividad como una forma de expresión más natural para ciertos tipos de algoritmos y sea capaz de aplicarla en casos sencillos. Sin profundizar en el tema, ha de conocer la relación existente entre la expresión recursiva e iterativa de un algoritmo.

Palabras clave: Recursividad, caso base, caso general, diseño recursivo.

Objetivos específicos:

Una vez logrado el tema, el estudiante:

1. Ha de entender el significado de una llamada recursiva y poder hacer un trazo en casos simples.
2. Ha de saber plantear un diseño recursivo de algoritmos sencillos.
3. Ha de conocer las ventajas y limitaciones de la recursividad.

Actividades vinculadas:

Todas las que constan.

Dedicación: 23h

Grupo grande/Teoría: 4h

Grupo pequeño/Laboratorio: 4h

Aprendizaje autónomo: 15h

TEMA 3. ESTRUCTURAS DE DATOS

Descripción:

Este tema se centra en las estructuras de datos más comunes: pilas, colas, listas, diccionarios, conjuntos, árboles y grafos y su utilización en la solución de problemas. El enfoque se orienta inminentemente en el uso de las estructuras habituales para representar datos estructurados y las implicaciones de la representación escogida en los programas que después han de manipularlas.

Palabras clave: Pila, cola, lista, conjunto, diccionario, árbol, grafo, operación, implementación, especificación, recorrido, eficiencia de las operaciones, eficiencia en espacio.

Objetivos específicos:

Una vez logrado el tema, el estudiante:

1. Ha de conocer las principales estructuras de datos y sus operaciones características.
2. Ha de conocer algunos de los principales problemas paradigmáticos que se resuelven con estas estructuras de datos.
3. Ha de ser capaz de usar las estructuras de datos para representar convenientemente las estructuras subyacentes en problemas de abasto medio.
4. Ha de conocer las implicaciones que tiene el uso de una u otra estructura de datos en la resolución de un problema en cuanto a la eficiencia del resultado final.

Actividades vinculadas:

Todas las que constan.

Dedicación: 45h

Grupo grande/Teoría: 10h

Grupo pequeño/Laboratorio: 10h

Aprendizaje autónomo: 25h

TEMA 4. INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE

Descripción:

El desarrollo del software como proceso de ingeniería. Los principales rasgos del proceso de desarrollo. El desarrollo en equipo y las herramientas de soporte. Prueba de software: abasto, técnicas y herramientas.

Palabras clave: Ingeniería de software, problemas en el desarrollo de software, modelos de desarrollo, desarrollo cooperativo, prueba.

Objetivos específicos:

Una vez logrado el tema, el estudiante:

1. Ha de haber adquirido una perspectiva del desarrollo de programario como proceso industrial.
2. Ha de conocer las principales problemáticas del proceso de desarrollo de programario.
3. Ha de conocer y saber cómo gestionar algunos de los escenarios de desarrollo en equipo más habituales.
4. Ha de conocer los principios de la prueba de programario y las herramientas implicadas y saber emplearlas en problemas reales.
5. Ha de tener nociones prácticas de trabajo empleando una metodología ágil.

Actividades vinculadas:

Todas las que constan.

Dedicación: 23h

Grupo grande/Teoría: 4h

Grupo pequeño/Laboratorio: 4h

Aprendizaje autónomo: 15h

TEMA 5: COMPLEJIDAD ALGORÍTMICA

Descripción:

Noción de complejidad en tiempo y espacio. Medida de la complejidad y parámetros que intervienen. Complejidad en peor caso. Jerarquía de complejidades. La tiranía de la complejidad. Problemas difíciles y concepto de dificultad. El papel de la constante en la realidad. Medida estadística del coste.

Palabras clave: Coste en tiempo y espacio. Medida del coste. Complejidad en peor caso. Orden de una función y notación Big O. Jerarquía de complejidades.

Objetivos específicos:

Una vez logrado el tema, el estudiante:

1. Entenderá la noción del coste de un programa y conocerá cuales son los factores que afectan al coste.
2. Conocerá el concepto de complejidad en peor caso y sabrá calcularla en el caso de algoritmos sencillos.
3. Entenderá las implicaciones que supone para un algoritmo tener una determinada complejidad en peor caso.

Actividades vinculadas:

Todas las que constan.

Dedicación: 14h

Grupo grande/Teoría: 2h

Grupo pequeño/Laboratorio: 2h

Aprendizaje autónomo: 10h

ACTIVIDADES

ACTIVIDAD 1. EXAMEN

Descripción:

La asignatura contempla un examen final que consiste en un conjunto de ejercicios a resolver individualmente sobre papel sin soporte de ningún tipo de material y en un tiempo ajustado.

Entregable:

Se entrega la solución individual del examen y se evalúa ésta.

Dedicación: 2h

Grupo grande/Teoría: 2h

ACTIVIDAD 2. ESTUDIO DE CONTENIDOS

Descripción:

El estudio de contenidos es la actividad individual o colectiva que conduce a entender y asumir los conocimientos, vocabulario y técnicas que forman parte de los contenidos de la asignatura.

Material:

Los materiales de soporte son:

- Referencia principal de la asignatura (libre en formato web).
- Colección de problemas de la asignatura.

Dedicación: 25h

Aprendizaje autónomo: 25h

ACTIVIDAD 3. CLASE EXPOSITIVA

Descripción:

Son clases presenciales específicamente dedicadas a la comprensión de los contenidos de la asignatura. Son clases con baja participación de los estudiantes.

Material:

Los materiales de soporte son:

- Referencia principal de la asignatura (libre en formato web).
- Bibliografía básica.
- Colección de problemas de la asignatura.

Dedicación: 12h

Grupo grande/Teoría: 12h

ACTIVIDAD 4. CLASES DE PROBLEMAS

Descripción:

Son clases presenciales específicamente dedicadas a la resolución de problemas. Se hacen en una clase ordinaria y son complementarias a la actividad del laboratorio. Son clases que requieren la participación de los estudiantes.

Material:

Los materiales de soporte son:

- Referencia principal de la asignatura (libre en formato web).
- Bibliografía básica.
- Colección de problemas de la asignatura.

Dedicación: 12h

Grupo grande/Teoría: 12h

ACTIVIDAD 5. CLASE DE LABORATORIO

Descripción:

El estudiante tiene como objetivo la solución de pequeños ejercicios que complementan los contenidos y colaboran en la mejor comprensión de éstos. Los ejercicios se realizan en el laboratorio y comportan la implementación real de programas sobre el computador y su comprobación.

La actividad puede comportar la finalización de los ejercicios en tiempo de aprendizaje autónomo.

Material:

Los materiales de soporte son:

- Referencia principal de la asignatura (libre en formato web).
- Colección de problemas de la asignatura.
- Manuales de programario utilizado.
- Bibliografía básica.

Entregable:

Quincenalmente, el estudiante, de manera individual hace un pequeño ejercicio entregable que evalúa sus progresos en esta actividad. El ejercicio se realiza dentro del tiempo de la misma actividad.

Dedicación: 41h

Grupo pequeño/Laboratorio: 26h

Aprendizaje autónomo: 15h

ACTIVIDAD 6. RESOLUCIÓN DE PROBLEMAS

Descripción:

Es una actividad que hace el estudiante autónomamente y que consiste en la solución de problemas de programación, generalmente sin ser necesario el soporte del computador.

Material:

Los materiales de soporte son:

- Referencia principal de la asignatura (libre en formato web).
- Colección de problemas de la asignatura.

Entregable:

La actividad comporta la entrega durante el curso de algunos problemas que se corrigen convenientemente y forman parte de la evaluación de la asignatura.

Dedicación: 30h

Aprendizaje autónomo: 30h

ACTIVIDAD 7. PROYECTO

Descripción:

La asignatura requiere realizar un proyecto de programación de medida mediana. El proyecto consiste en la implementación y prueba de un programa el diseño del cual viene dado por el enunciado. Esta actividad se realiza en grupo y comporta, además, la escritura de un informe técnico sobre el programa. Esta actividad tiene naturaleza de síntesis de todos los conocimientos de la asignatura.

Material:

Los materiales de soporte son:

- Servicio de laboratorio informático del CCEPSEM.
- Enunciado y guión del proyecto.
- Ejemplo de informe.
- Apuntes personales y resto de material de soporte del curso.

Entregable:

Como resultado de la actividad se entregan:

1. El informe del proyecto.
2. El código fuente resultado del proyecto.

La entrega se realiza con la presencia de todo el equipo de trabajo. Se evalúa el informe y el resultado a que se ha llegado en la confección del proyecto.

Dedicación: 28h

Grupo grande/Teoría: 4h

Grupo pequeño/Laboratorio: 4h

Aprendizaje autónomo: 20h

SISTEMA DE CALIFICACIÓN

La calificación se realiza en base a 3 elementos:

1. La evaluación del trabajo autónomo del estudiante (A). Este componente contiene tanto el progreso hecho en los aspectos teóricos como en los prácticos. Su medida se realiza en base a ejercicios obligatorios entregados durante el curso.
2. La evaluación del proyecto (P). Se realiza a partir de una entrega presencial del proyecto en curso que puede comportar una presentación pública y la confección de una memoria.
3. La evaluación final (F). Se hace a través de un examen final que tiene naturaleza global e integra todos los conocimientos y destrezas adquiridas durante el curso.

A partir de estos elementos se calcula la nota final con las siguientes ponderaciones:

$$\text{Final} = 0.35A + 0.25P + 0.40F$$

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Las actividades se realizarán siguiendo los usos y las costumbres del trabajo académico y, particularmente, se respetarán las siguientes pautas:

1. Aquellas actividades que sean explícitamente declaradas como individuales, sean de naturaleza presencial o no, se realizarán sin ninguna colaboración por parte de otras personas.
2. Las fechas, formatos y otras condiciones de entrega que se fijen serán de obligado cumplimiento.
3. El uso del laboratorio informático se reservará exclusivamente para las actividades académicas y en ningún caso se podrá hacer un uso abusivo.

BIBLIOGRAFÍA

Básica:

- Downey, A. Python for software design: how to think like a computer scientist [en línea]. Cambridge: Cambridge University, 2009 [Consulta: 09/11/2020]. Disponible a: <http://openbookproject.net/thinkcs/python/english3e/>. ISBN 9780521725965.
- Miller, Bradley N.; Ranum, David L. Problem solving with algorithms and data structures using Python. Wilsonville, OR: Franklin, Beedle & Associates, 2006. ISBN 9781590280539.
- Necaie, Rance D. Data structures and algorithms using Python. Hoboken: Wiley, 2011. ISBN 9780470618295.

Complementaria:

- Metha, D. P.; Sahni, S., ed. Handbook of data structures and applications. Boca Raton, Fla: Chapman and Hall/CRC, 2005. ISBN 1584884355.
- Sommerville, I. Software engineering [en línea]. 9th ed. Boston: Pearson / Addison Wesley, 2011 [Consulta: 01/06/2022]. Disponible a: <https://ebookcentral-proquest-com.recursos.biblioteca.upc.edu/lib/upcatalunya-ebooks/detail.action?docID=5185655>. ISBN 9780137053469.