



Guía docente

340374 - ESIN-I3O23 - Estructura de la Información

Última modificación: 17/05/2023

Unidad responsable: Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú
Unidad que imparte: 723 - CS - Departamento de Ciencias de la Computación.

Titulación: GRADO EN INGENIERÍA INFORMÁTICA (Plan 2018). (Asignatura obligatoria).

Curso: 2023 **Créditos ECTS:** 6.0 **Idiomas:** Catalán

PROFESORADO

Profesorado responsable: Bernardino Casas Fernández, Jordi Esteve Cusiné

Otros: Jordi Esteve Cusiné

CAPACIDADES PREVIAS

Dominio de las técnicas de programación imperativa basada en objetos:

- * Clases
- * Objetos
- * Métodos
- * Paso de parámetros
- * Recursividad

Conocer bien al menos un lenguaje imperativo orientado a objetos, preferentemente C++.

Madurez algorítmica y matemática.

REQUISITOS

Haber aprobado PRO1 o al menos tener los conocimientos que se imparten en PRO1.

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:

1. CEFB3. Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para el tratamiento automático de la información por medio de sistemas computacionales y su aplicación para la resolución de problemas propios de la ingeniería.
2. CEFB4. Conocimiento de los fundamentos del uso y programación de los computadores, los sistemas operativos, las bases de datos y, en general, los programas informáticos con aplicación en ingeniería.
3. CEFB5. Conocimiento de la estructura, funcionamiento e interconexión de los sistemas informáticos, así como los fundamentos de su programación.
4. CEFB6. Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
5. CEFB7. Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
6. CEFB8. Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Transversales:

7. APRENDIZAJE AUTÓNOMO - Nivel 2: Llevar a cabo las tareas encomendadas a partir de las orientaciones básicas dadas por el profesorado, decidiendo el tiempo que se necesita emplear para cada tarea, incluyendo aportaciones personales y ampliando las fuentes de información indicadas.
8. TRABAJO EN EQUIPO - Nivel 1: Participar en el trabajo en equipo y colaborar, una vez identificados los objetivos y las responsabilidades colectivas e individuales, y decidir conjuntamente la estrategia que se debe seguir.

METODOLOGÍAS DOCENTES

El enfoque metodológico consiste en:

- Presentación en el aula, en clases participativas, de conceptos y procedimientos asociados a las materias.
- Realización de ejercicios individualmente o en equipo de forma presencial y no presencial.
- Realización de prácticas individualmente (actividades) o en equipo (proyecto) de forma presencial y no presencial.
- Estudio individual, pruebas y exámenes.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

Proporcionar al alumno la capacidad de especificar, diseñar, implementar y evaluar estructuras de datos y la habilidad de identificar los algoritmos más adecuados sobre estas estructuras. Igualmente, se pretende proporcionar al alumno más experiencia en el campo de la programación mediante la realización de actividades y un proyecto.

1. Especificar: El estudiante deberá aprender algunas técnicas de especificación informal que le permitirán describir el comportamiento de las clases.
2. Diseñar: El estudiante deberá aprender a seleccionar y combinar las técnicas algorítmicas y las estructuras de datos apropiadas para elaborar diseños modulares correspondientes a problemas de mediana dificultad.
3. Implementar: El estudiante aprenderá las principales técnicas de organización de los datos que permitan acceder y modificar la información de manera eficiente respetando la correspondiente especificación.
4. Evaluar: El estudiante será capaz de comparar la eficiencia relativa de soluciones alternativas a un mismo problema y tomar decisiones de diseño en función de criterios objetivos basados en la eficiencia en tiempo de ejecución y espacio de memoria utilizada.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

| 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

1. Programación orientada a objetos

Descripción:

Especificación de clases.
Clases y objetos.
Atributos y métodos.
Paso de parámetros.
Generalización.
Sobrecarga de métodos.

IMPORTANTE: Este tema es de repaso, por tanto no hay docencia presencial del mismo.

Dedicación: 5h

Aprendizaje autónomo: 5h

2. Análisis de la eficiencia de algoritmos

Descripción:

Medidas del coste en espacio y tiempo de algoritmos. Notaciones asintóticas. Coste en caso peor y medio. Reglas de cálculo de la eficiencia de algoritmos iterativos y recursivos.

Actividades vinculadas:

Activitat 1
Activitat 2
Activitat 3
Control 1
Control 2
Projecte

Dedicación: 7h

Grupo grande/Teoría: 2h
Aprendizaje autónomo: 5h

3. Estructuras lineales estáticas

Descripción:

Repaso del concepto de secuencia.
Pilas y colas.
Listas.
Aplicaciones.
Implementación en vector y enlazada de estructuras lineales.

IMPORTANTE: Este tema es de repaso, por lo tanto la docencia presencial del mismo será muy breve.

Dedicación: 8h

Grupo grande/Teoría: 1h
Grupo pequeño/Laboratorio: 2h
Aprendizaje autónomo: 5h



4. Estructuras lineales dinámicas

Descripción:

Punteros.
Obtener y liberar memoria.
Pilas, colas y listas enlazadas con memoria dinámica.
Iteradores.
Ordenación por fusión (mergesort).

Actividades vinculadas:

Activitat 1
Control 1
Control 2
Projecte

Dedicación: 22h

Grupo grande/Teoría: 4h
Grupo pequeño/Laboratorio: 6h
Aprendizaje autónomo: 12h

5. Árboles

Descripción:

Repaso del concepto de árbol.
Árboles generales.
Árboles binarios.
Aplicaciones.
Recorridos en preorden, postorden, inorden y por niveles.
Implementación de árboles con vector de punteros a los hijos.
Implementación de árboles con punteros primer-hijo / siguiente-hermano.

Actividades vinculadas:

Control 1
Control 2
Activitat 2
Projecte

Dedicación: 18h

Grupo grande/Teoría: 4h
Grupo pequeño/Laboratorio: 4h
Aprendizaje autónomo: 10h

6. Diccionarios

Descripción:

Especificación de los diccionarios.

Aplicaciones.

Técnicas de implementación sencillas: listas, listas ordenadas, listas autoorganizadas.

Técnicas de implementación avanzadas: árboles binarios de búsqueda (BSTs), árboles binarios de búsqueda balanceados (AVLs), tablas de dispersión (Hash Tables), árboles digitales (Tries).

Ordenaciones Quicksort y Radixsort.

Actividades vinculadas:

Control 2

Activitat 3

Projecte

Dedicación: 27h

Grupo grande/Teoría: 8h

Grupo pequeño/Laboratorio: 4h

Aprendizaje autónomo: 15h

7. Colas de prioridad

Descripción:

Especificación de las colas de prioridad.

Aplicaciones.

Implementación mediante heaps.

Ordenación Heapsort.

Actividades vinculadas:

Control 2

Activitat 3

Projecte

Dedicación: 7h

Grupo grande/Teoría: 2h

Grupo pequeño/Laboratorio: 1h

Aprendizaje autónomo: 4h

8. Grafos

Descripción:

Representaciones con matrices de adyacencia, listas de adyacencia y multilistas de adyacencia.

Recorrido en profundidad (DFS).

Recorrido en anchura (BFS).

Ordenación topológica.

Algoritmo de Dijkstra por caminos mínimos.

Algoritmo de Kruskal para árboles de expansión mínimos.

Actividades vinculadas:

Control 2

Dedicación: 14h

Grupo grande/Teoría: 5h

Grupo pequeño/Laboratorio: 1h

Aprendizaje autónomo: 8h



Proyecto

Descripción:

Proyecto de programación en parejas. El proyecto consistirá en la implementación de un diseño modular con el lenguaje C ++. La tarea del estudiante será implementar diferentes módulos del diseño modular dado en el enunciado. Se dispondrá del enunciado del proyecto y de una guía y normas de programación de ESIN.

Dedicación: 42h

Grupo pequeño/Laboratorio: 8h

Actividades dirigidas: 34h

ACTIVIDADES

Actividad 1

Descripción:

Ver versión catalana.

Material:

Ver versión catalana.

Entregable:

Ver version catalana.

Dedicación: 3h 30m

Grupo pequeño/Laboratorio: 3h

Actividades dirigidas: 0h 30m

Actividad 2

Descripción:

Ver versión catalana

Material:

Ver versión catalana.

Entregable:

Ver versión catalana.

Dedicación: 3h 30m

Grupo pequeño/Laboratorio: 3h

Actividades dirigidas: 0h 30m



Actividad 3

Descripción:

Ver versión catalana.

Material:

Ver versión catalana.

Entregable:

Ver versión catalana.

Dedicación: 3h 30m

Grupo pequeño/Laboratorio: 3h

Actividades dirigidas: 0h 30m

Control 1

Descripción:

Ver versión catalana.

Entregable:

Ver versión catalana.

Dedicación: 2h

Grupo grande/Teoría: 2h

Control 2

Descripción:

Ver version catalana.

Entregable:

Ver versión catalana.

Dedicación: 3h

Grupo grande/Teoría: 3h

Proyecto

Descripción:

Ver versión catalana.

Material:

Ver versión catalana.

Entregable:

Ver versión catalana.

Dedicación: 27h

Grupo pequeño/Laboratorio: 7h

Aprendizaje autónomo: 20h

SISTEMA DE CALIFICACIÓN

C1 = Control 1 (parcial). Prueba individual (2 horas).
C2 = Control 2 (final). Prueba individual (3 horas) de carácter global.
Act = Nota resultado de las 3 actividades realizadas.
Pro = Nota resultado del proyecto.

Calificación Teoría: $QT = \text{valor máximo} (0,40 \cdot C1 + 0,60 \cdot C2, C2)$

Calificación Final: $QF = 0,5 \cdot QT + 0,2 \cdot Act + 0,3 \cdot Pro$

Proyecto (Pro): Se evalúa a partir de: la ejecución del programa realizado, el código presentado, y una prueba de validación (PV). La Prueba de Validación puede realizarse mediante una entrevista presencial, o bien en el Control Final de la asignatura.

La realización y presentación del proyecto será condición necesaria para la superación de la asignatura. En caso contrario, la calificación final de toda la asignatura será de 'No Presentado'.

Se puede reevaluar el Control 2 (final).

NORMAS PARA LA REALIZACIÓN DE LAS PRUEBAS.

Las pruebas escritas (Control 1 y 2) y las Actividades son presenciales e individuales, principalmente realizadas sobre la plataforma jutge.org.

El proyecto se realiza en equipos de dos personas. Se entrega de forma no presencial y se evalúa tanto de forma presencial (Prueba de Validación) como de forma no presencial a partir de la documentación presentada.

BIBLIOGRAFÍA

Básica:

- Bernardino Casas, Jordi Esteve. Apunts d'ESIN. Transparències pels alumnes. Campus virtual,
- Bernardino Casas, Jordi Esteve. Col·lecció de problemes d'ESIN. Campus virtual,
- Bernardino Casas, Jordi Esteve. Manual de laboratori d'ESIN. Campus virtual,
- Hernández Orallo, Enrique; Hernández Orallo, José; Juan Lizandra, M^a Carmen. C++ estándar : [programación con el estándar ISO y la biblioteca de plantilla (STL)]. Madrid: Paraninfo, 2002. ISBN 8497320409.
- Guardati Buemo, Silvia. Estructura de datos orientada a objetos : algoritmos con C++ [en línea]. Naucalpan de Juárez (México): Pearson, 2007 [Consulta: 16/02/2024]. Disponible a : https://www-ingebook-com.recursos.biblioteca.upc.edu/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=3887. ISBN 9702607922.
- Martí Oliet, Narciso; Ortega Mallén, Yolanda; Verdejo López, José Alberto. Estructuras de datos y métodos algorítmicos. Ejercicios resueltos [en línea]. Madrid [etc.]: Pearson Educación, 2001 [Consulta: 06/03/2024]. Disponible a : https://www-ingebook-com.recursos.biblioteca.upc.edu/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=1316. ISBN 8420538493.

Complementaria:

- Stroustrup, Bjarne. El lenguaje de programación C++. Madrid [etc.]: Addison Wesley, 2002 (reimpresió 2007). ISBN 847829046X.
- Joyanes Aguilar, Luis. Programación en C++ : algoritmos, estructuras de datos y objetos [en línea]. 2a ed. Madrid: McGraw-Hill, 2006 [Consulta: 12/02/2024]. Disponible a : https://www-ingebook-com.recursos.biblioteca.upc.edu/ib/NPcd/IB_BooksVis?cod_primaria=1000187&codigo_libro=4071. ISBN 844814645X.