TecnoCampus Escola Superior Politècnica
Centre adscrit a la upf. Universitat Pompeu Fabra Barcelona

DOBLE GRADO EN INFORMÁTICA DE GESTIÓN Y SISTEMAS DE INFORMACIÓN/GRADO EN DISEÑO Y PRODUCCIÓN DE VIDEOJUEGOS

107131 - ESTRUCTURA DE DATOS Y ALGORITMOS

Información general

- Tipo de asignatura : Obligatoria
- Coordinador : Alfons Palacios Gonzàlez
- Curso: Primero
- Trimestre: Tercero
- Créditos: 4
- Profesorado:
  - Enric Sesa Nogueras <sesa@tecnocampus.cat>

Idiomas de impartición

- Inglès
- Catalán

Enunciados de prácticas y / o exámenes pueden ser en inglés.

Competencias que se trabajan

Básica

- B2_Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio
- B5_Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

Común

- CIN1_Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y la legislación y normativa vigente.
- CIN3_Capacidad para comprender la importancia de la negociación, los hábitos de trabajo efectivos, el liderazgo y las habilidades de comunicación en todos los entornos de desarrollo de software
- CIN5_Conocimiento , administración y mantenimiento sistemas , servicios y aplicaciones informáticas
- CIN6_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.
- CIN7_Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
- CIN8_Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
- CIN13_Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los sistemas de información, incluidos los basados en web.

Específica

- EFB3_Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería.
- EFB4_Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.

Transversal

- T1_Que los estudiantes conozcan un tercer idioma, que será preferentemente inglés, con un nivel adecuado de forma oral y por escrito y de acuerdo con las necesidades que tendrán las graduadas y los graduados en cada titulación

Descripción

Esta asignatura cierra el bloque de las asignaturas de programación del primer curso. Está dividida en cuatro partes: en la primera parte se estudia el uso de los almacenes de datos volátiles (en memoria) más frecuentes: colas y pilas, listas, conjuntos, maps, ... continuando la aproximación OO iniciada a las asignaturas previas. La segunda parte es una breve introducción a las técnicas de programación recursiva. La tercera parte muestra la utilización de archivos como herramienta de almacenamiento no volátil. La última parte es una introducción a la complejidad algorítmica.

Si por motivos relacionados con la Covid-19 la asignatura tuviera que impartise en modalidad no presencial o en alguna modalidad que representara una disminución sensible de la presencialidad, la metodología de las sesiones se adaptará a estas circunstancias. El objetivo último de esta adaptación será el de preservar al máximo la consecución de los conocimientos y de las competencias especificadas en este plan docente.

Resultados de aprendizaje

A nivel general, esta asignatura contribuye a los siguientes resultados de aprendizaje especificados para la materia a la que pertenece (Algorítmica y programación)

- Demostrar conocimiento y comprensión de hechos esenciales, conceptos, principios y teorías relativas a la informática y en sus disciplinas de referencia.
- Aplicar técnicas básicas de descomposición modular de problemas
- Diseñar la arquitectura de los programas utilizando técnicas de orientación a objetos, de modularización y de especificación e implementación de tipos abstractos de datos
- Construir algoritmos correctos y eficientes para problemas de dificultad pequeña.
- Implementar algoritmos sencillos en un lenguaje de programación imperativo de referencia para el nivel inicial
- Identificar las soluciones algorítmicas más adecuadas para resolver problemas de dificultad mediana.
- Razonar sobre la corrección y la eficiencia de una solución algorítmica.
- Diseñar, escribir, probar, depurar, documentar y mantener código en un lenguaje de alto nivel para resolver problemas de programación aplicando esquemas algorítmicos y usando estructuras de datos.
- Usar las herramientas de un entorno de desarrollo de software para crear y desarrollar aplicaciones.

En un nivel más concreto, al finalizar la asignatura el estudiante debe ser capaz de:

- RA1: Aplicar técnicas recursivas para la resolución de problemas. Codificar programas de naturaleza recursiva.
- RA2: Conocer la especificación de alto nivel de colecciones de datos como las pilas, las colas, las listas, los conjuntos y los diccionarios (colecciones asociativas). Hacer uso de estas colecciones en programas, aplicando algoritmos adecuados.
- RA3: Seleccionar las colecciones de datos más adecuados para un problema dado
- RA4: Almacenar y recuperar información de archivos.
- RA5: mostrar conocimiento sobre las diferentes clases de complejidad computacional

Metodología de trabajo

Todos los conceptos teóricos de la materia expondrán en clases de teoría (grupos grandes). En estas clases, y a discreción de los docentes impartidores, también se resolverán ejercicios y problemas de carácter más práctico. Asimismo, y siempre a discreción de los impartidores, se podrá pedir a los estudiantes que resuelvan, de forma individual o en grupo, problemas y / o ejercicios breves. Estas actividades, breves y optativas, servirán al estudiante como instrumento de autoevaluación de su asimilación de los contenidos de la materia

Los estudiantes deberían asistir a todas las clases con un ordenador portátil con la capacidad de ejecutar el software apropiado para la asignatura. Los docentes impartidores informarán de cual es este software y cómo se puede obtener.

Los conceptos de carácter más práctico y todo lo que en esencia se pueda considerar la aplicación práctica de los conceptos teóricos serán trabajados de manera más intensiva en grupos pequeños (de laboratorio). En las sesiones que se programen al efecto se darán las herramientas adecuadas para resolver las actividades programadas aunque se espera que estas se alarguen desde el punto de vista temporal, más allá de las horas de laboratorio y que, en consecuencia , los estudiantes deban finalizarlas durante el tiempo de aprendizaje autónomo.

Se pondrá a disposición de los estudiantes actividades de tipo totalmente opcional que le ayuden a preparar y prepararse para las de tipo obligatorio.

Contenidos

1.- Colecciones de datos

- Colecciones secuenciales (pilas, colas, listas...)
- Conjuntos
- Colecciones asociativas (maps o "diccionarios")

2.- Recursivitad

- Algoritmos recursivos vs. algoritmos iterativos
- Tratamiento recursivo de secuencias. estrategias

3.- Almacenes permanentes: archivos

- E / S hacia y desde archivos. Canales
- Archivos de texto
- Archivos de datos (binarios)

4.- Introducción a la complejidad algorítmica

Actividades de aprendizaje

Se pone a disposición de los estudiantes una serie de actividades de carácter eminentemente práctico (ejercicios cortos, problemas ...) que son la base de las actividades de aprendizaje de la asignatura. Estas actividades los estudiantes las tendrán que resolver, a menudo de manera no presencial, siguiendo las indicaciones de los docentes y también serán trabajadas en clase, ya sea como ejemplos en las sesiones de teoría, ya sea en las sesiones de laboratorio. Si bien estas actividades tendrán carácter optativo (los docentes no verificarán de manera individualizada su realización por parte de los estudiantes), serán imprescindibles para alcanzar los conocimientos teórico-prácticos de la asignatura.

Con el objetivo de recoger evidencia del logro de los resultados de aprendizaje esperados se realizarán las siguientes actividades de carácter evaluativo:

Dos pruebas escritas (utilizando, en su caso, una herramienta IDE)

- La primera prueba versará sobre el primer contenido (colecciones de datos) y dará evidencia de los resultados de aprendizaje RA2 y RA3. Esta prueba pide a los estudiantes el desarrollo correcto de software a partir de una base preestablecida. Esto tiene que ver con las siguientes competencias comunes y específicas (entre paréntesis los aspectos más relevantes de cada competencia a los que la asignatura contribuye)
  - CIN1 (desarrollar software),
  - CIN5 (mantener software),
  - CIN6 (aplicar procedimientos algorítmicos),
  - CIN7 (usar estructuras de datos),
  - CIN8 (construir y mantener aplicaciones),
  - EFB3 (comprender y dominar los conceptos de la algorítmica) y
  - EFB4 (uso de programas informáticos)

- La segunda prueba incluirá todos los contenidos y dará evidencia de todos los resultados de aprendizaje previstos. Esta prueba pide a los estudiantes el desarrollo correcto de software a partir de una base preestablecida e incluye el uso de la recursividad y los archivos. A nivel de competencias comunes y específicas serán relevantes todas las mencionadas para la primera prueba añadiendo:
  - CIN13 (conocimiento y aplicación de las herramientas necesarias para el almacenamiento)

Se realizarán también cuatro prácticas:

- En las dos primeras prácticas, los estudiantes deberán codificar, probar, depurar y documentar programas que hagan uso de colecciones de datos. Con ello se podrá recoger evidencia de los resultados de aprendizaje RA2 y RA3
- En la tercera práctica se trabajará, sobre todo, la codificación, prueba, depuración y documentación de programas de naturaleza recursiva. Con ello se podrá recoger evidencia del resultado de aprendizaje RA1
- En la cuarta práctica se trabajará, sobre todo, la codificación, prueba, depuración y documentación de programas que hagan uso de colecciones de datos y de archivos, recogiendo así evidencia de los resultados de aprendizaje RA2, RA3 y RA4.

A nivel competencial las cuatro prácticas, que mayoritariamente se deberán desarrollar en tiempo no presencial, cubren todas las competencias comunes y específicas de la asignatura:

- CIN1 (desarrollar software),
- CIN3 (hábitos de trabajo efectivos en los entornos de desarrollo de software)
- CIN5 (mantener software),
- CIN6 (aplicar procedimientos algorítmicos),
- CIN7 (usar estructuras de datos),
- CIN8 (construir y mantener aplicaciones),
- CIN13 (conocimiento y aplicación de las herramientas necesarias para el almacenamiento)
- EFB3 (comprender y dominar los conceptos de la algorítmica) y
- EFB4 (uso de programas informáticos)

Para superar (aprobar) las actividades evaluativas, los estudiantes deberán demostrar

- Que han adquirido los conocimientos teóricos relativos a los contenidos de la asignatura y que su comprensión les permite llevarlos a la práctica [MECES-2 punto a, punto c]
- Que pueden desarrollar soluciones a problemas que, si bien son similares a otros vistos anteriormente, presentan aspectos que son nuevos [MECES-2 punto f]

En relación a las competencias básicas asignadas a la asignatura, éstas quedan cubiertas especialmente en cuanto a los aspectos que se explicitan:

- CB2: resolución de problemas dentro de su área de estudio.
- CB5: desarrollo de habilidades de aprendizaje necesarias para estudios posteriores (acceso autónomo a documentación, hábitos de trabajo efectivos)

Finalmente, y en cuanto a la competencia transversal asociada a la asignatura (CT1: conocimiento tercera lengua) esta se trabaja a partir de las fuentes documentales que los estudiantes deben consultar, dado que todas ellas se encuentran en inglés. En algunas ocasiones también se suministran en inglés (solamente) los enunciados de algunas de las prácticas de la asignatura.

Para cada actividad, los docentes informarán de las normas y condiciones particulares que las rijan.

Las actividades unipersonales presuponen el compromiso del estudiante de realizarlas de manera individual y sin ningún tipo de colaboración con otras personas. Se considerarán suspendidas (calificación 0) todas aquellas actividades en que el estudiante no se ajuste a este compromiso de individualidad, independientemente de su papel (emisor o receptor) y sin que ello excluya la posible aplicación de otras sanciones de acuerdo con el Régimen Disciplinario vigente.

Igualmente, las actividades que se deban realizar en grupo presuponen el compromiso por parte de los estudiantes que lo integran de realizarlas en el seno del grupo y sin ningún tipo de colaboración con otros grupos o personas que sean ajenas (individualidad grupal). Se considerarán suspendidas (calificación 0) todas aquellas actividades en las que el grupo no haya respetado este compromiso con independencia de su papel (emisor o receptor) y sin que ello excluya la posible aplicación de otras sanciones de acuerdo con el Régimen Disciplinario vigente.

En el caso particular de las prácticas, cuando en alguna de ellas no se respete el compromiso de individualidad y / o utilicen medios fraudulentos en su realización, la calificación de prácticas (Pr) será, para todos los miembros del grupo, de 0 puntos (Pr = 0) con independencia de la calificación de las otras prácticas y sin que ello excluya la posible aplicación de otras sanciones de acuerdo con el Régimen Disciplinario vigente.

Cualquier actividad no entregada se considerará puntuada con cero puntos

Es potestativo de los docentes aceptar o no entregas fuera de los plazos que se indiquen. En caso de que estas entregas fuera de plazo se acepten, es potestativo del docente decidir si aplica alguna penalización y la cuantía de la misma.

Sistema de evaluación

La calificación final de la asignatura (QF) se calculará según se detalla a continuación

Pe1: calificación de la primera prueba escrita

Pe2: calificación de la segunda prueba escrita

Teo: calificación de teoría

Pr_**i**: calificación de la i-ésima práctica

Pr: calificación de prácticas

QPT: calificación ponderada teoría + prácticas

QF: calificación final de la asignatura

Teo = max((Pe1+Pe2)/2, Pe2)

Pr = media(Pr_i)

QPT = Pr·**0.35** + Teo·**0.65**

QF = Teo si Teo < 4

QF = QPT si Teo ? 4

Es responsabilidad del alumno evitar el plagio en todas sus formas. En el caso de detectar un plagio, independientemente de su alcance, en alguna actividad, corresponderá a tener una nota de 0. Además, el profesor comunicará a la Jefa de Estudios la situación para que tome medidas aplicables en materia de régimen sancionador. En el contexto de esta asignatura, plagio también significa utilizar y / o adaptar código que no se haya desarrollado de manera totalmente individual (o en el seno del grupo en el caso de actividades grupales). Facilitar el código que da lugar al plagio es también una forma de plagio y será tratado de la misma manera. De forma resumida podemos decir que las actividades evaluativas deben resolverse de manera estrictamente no colaborativa (en el caso de actividades en grupo la colaboración no puede trascender el seno del grupo).

Recuperación

Para poder realizar el examen de recuperación es necesario que la calificación final (QF) sea diferente de "no presentado".

La parte de prácticas de la asignatura (calificación Pr) NO es recuperable.

Para los estudiantes que asistan al examen de recuperación su calificación Teo será la obtenida en esta prueba y su calificación final se calculará con las fórmulas anteriormente detalladas, y en ningún caso será superior a 5.

Recursos

Básicos

Bibliografías

- Sesa i Nogueras. EDA: notes de classe, exemples i exercicis. Publicació interna del TCM.

Enlaces web

- Java 8 API Documentation. https://docs.oracle.com/javase/8/docs/api/
- Tutorial Oracle Java Collections https://docs.oracle.com/javase/tutorial/collections/