

CENTRO: 180 - Escuela de Ingeniería Informática

TITULACIÓN: 4008 - Grado en Ingeniería Informática

ASIGNATURA: 40823 - PROGRAMACIÓN III

Vinculado a : (Titulación - Asignatura - Especialidad)

4801-Doble Grado en Ingeniería Informática y - 48124-PROGRAMACIÓN III - 00

CÓDIGO UNESCO: 1203

TIPO: Obligatoria

CURSO: 3

SEMESTRE: 1º semestre

CRÉDITOS ECTS: 6

Especificar créditos de cada lengua:

ESPAÑOL: 6

INGLÉS:

SUMMARY

REQUISITOS PREVIOS

- Introducción a la Informática
- Fundamentos de Programación
- Programación I
- Programación II

Datos identificativos del profesorado que la imparte.

Plan de Enseñanza (Plan de trabajo del profesorado)

Contribución de la asignatura al perfil profesional:

Programación III es una asignatura que proporciona al estudiante técnicas, herramientas y métodos adecuados para la construcción de software de calidad, profundizando fundamentalmente en dos factores: la correctitud y la eficiencia. Este conocimiento aporta criterios para seleccionar la técnica de diseño más adecuada para la solución de un problema o mejorar una propuesta. Además, se ilustra con problemas de dificultad mediana ampliamente estudiados, como son el de ordenación de un vector y el de búsqueda de un patrón en un texto.

Competencias que tiene asignadas:

Competencias generales: G1, G2, G3, G4, G5

Competencias nucleares de la Universidad de Las Palmas de Gran Canaria: N1, N2, N3, N4

Competencias del título de Grado en Ingeniería Informática: T5, T8, T9

Competencias comunes a la Ingeniería Informática: CII01, CII05, CII06, CII08

Objetivos:

- 1) Verificar la correctitud de algoritmos
- 2) Analizar la eficiencia de algoritmos
- 3) Diseñar algoritmos utilizando estrategias básicas conocidas
- 4) Comparar en función de su eficiencia distintos algoritmos que resuelven el mismo problema

Contenidos:

- 1.Verificación formal de algoritmos
 - 1.1.Introducción a la verificación. Lógica de predicados
 - 1.2.Verificación de algoritmos iterativos
 - 1.3.Verificación de algoritmos recursivos
- 2.Análisis de la eficiencia de algoritmos
 - 2.1.Introducción
 - 2.2.Notaciones asintóticas
 - 2.3.Análisis de la eficiencia de algoritmos iterativos
 - 2.4.Resolución de recurrencias
 - 2.5.Análisis de la eficiencia de algoritmos recursivos
- 3.Diseño de algoritmos
 - 3.1.Divide y vencerás
 - 3.2.Programación dinámica
- 4.Algoritmos fundamentales
 - 4.1.Ordenación de un vector
 - 4.2.Búsqueda de un patrón en un texto

Metodología:

La metodología que se va a utilizar está orientada a los estilos de aprendizaje de los alumnos, así se contempla en el Proyecto titulado: Aplicación metodológica orientada a los estilos de aprendizaje con apoyo de material innovador para las actividades formativas. Este Proyecto fue aprobado al grupo TILDE por el Vicerrectorado de Calidad e Innovación Educativa y se va a desarrollar en esta asignatura. Para ello, es necesario conocer la forma de aprender de cada alumno con apreciación de la existencia de diferentes estilos de aprendizaje, permitiendo una atención individualizada por perfil que motive al estudiante, le aporte confianza en sus capacidades y potencie su autoestima y autonomía.

Todas las tareas a realizar en las sesiones académicas se van a desarrollar en equipos. La composición de los equipos se establece teniendo en cuenta diferentes estilos de aprendizaje, que aporten la diversidad requerida para el trabajo colaborativo haciendo que los resultados de aprendizaje obtenidos sean más enriquecedores y completos, debido a la interacción y al diálogo entre estudiantes con perfiles distintos. Es una forma de incentivar la colaboración en una colectividad no competitiva y fortalecer el sentimiento de solidaridad y respeto mutuo, a la vez que disminuye la sensación de aislamiento.

En el horario establecido por el centro para las sesiones académicas teóricas y prácticas, se utilizan dos modalidades para los contextos de las actividades que el profesor y los alumnos desarrollarán en el proyecto: a nivel de grupo y a nivel de equipo.

Los quehaceres diarios durante las sesiones académicas teóricas y de problemas se resumen a continuación:

1. El profesor presenta a todo el grupo una síntesis del contenido de cada tema, publica el material disponible en la página web de la asignatura, indica las tareas individuales y colectivas a realizar y

establece un período adecuado para su finalización.

2. Este lapso de tiempo se consume en las sesiones de trabajo colaborativo, donde cada equipo planifica y organiza su labor, que incluye la ejecución de las tareas colectivas con discusión de aportaciones individuales y resolución de dudas entre iguales; además, debido a la presencia del profesor en el aula comprende la realimentación inmediata.

La dotación de medios en el aula es esencial para aplicar la metodología propuesta y poder usar la tecnología para buscar información, consultar fuentes y disponer de recursos didácticos de forma inmediata. Se requieren un portátil para cada estudiante y una pizarra por equipo.

El trabajo práctico en equipo está compuesto de actividades dirigidas por el profesor, que suponen la realización de programas en lenguaje Java y la elaboración de informes, que les permiten aplicar los conocimientos adquiridos en las sesiones académicas teóricas. El profesor presenta la actividad en las sesiones académicas prácticas, publica el enunciado en la página web de la asignatura, indica las tareas a realizar de forma individual y en equipo, establece un plazo de entrega y el porcentaje que aporta a la nota del trabajo práctico. El trabajo colaborativo se desarrolla en las sesiones académicas prácticas comprendidas en este período de tiempo.

El tiempo que dedica el profesor a cada equipo durante las diferentes sesiones académicas será aproximadamente el mismo, intentando optimizar su presencia en cada uno para poder ocuparse de todos los equipos. Los quehaceres que realiza son la resolución de dudas, la valoración del grado de participación y de las aportaciones, tanto individuales como colectivas, y la evaluación del progreso de cada estudiante.

El medio principal para suministrar información referente a la asignatura es la página web de la asignatura.

Criterios y fuentes para la evaluación:

1) Evaluación teórica

Criterios relativos a los exámenes escritos:

- Identificación de los elementos de la verificación formal de algoritmos y demostración de la correctitud.
- Identificación de los elementos que definen la eficiencia de un algoritmo y deducción del tiempo de ejecución.
- Estudio del mejor y peor caso de la eficiencia de un algoritmo.
- Adaptación de una técnica de diseño de algoritmos general a la solución de un problema concreto observando los detalles del funcionamiento.

2) Trabajo práctico en equipo

Criterios relativos a una práctica:

- El profesor someterá los programas e información entregados a los análisis y pruebas oportunos para formular una valoración global que tenga en cuenta todos los aspectos implicados en su realización, con especial énfasis en la ejecución (funcionamiento, adecuación a las especificaciones, robustez, ...) y en el estilo (formato, comentarios, elección de identificadores, ...). Cuando lo estime conveniente, el profesor podrá citar a los alumnos para formularles cuestiones que considere relevantes para la valoración global.

Criterios relativos a los informes:

- Valoración adecuada a la documentación estudiada
- Aplicación de los criterios de eficiencia para comparar y justificar la selección de un algoritmo entre varios que resuelven el mismo problema.

3) Coevaluación

Cada miembro del equipo valora a sus compañeros teniendo en cuenta su grado de:

- Colaboración
- Respeto

- Integración
- Distribución del trabajo
- Motivación
- Compromiso
- Conciliación

4) Examen final

Los criterios para la evaluación del examen escrito final son los mismos que los establecidos para la evaluación teórica. Los criterios para la evaluación del examen práctico final son los mismos que los establecidos para la evaluación del trabajo práctico.

Sistemas de evaluacion:

1) Evaluación continua

* Teórica

La escala que se utilizará para valorar la teoría es de 0 a 10 puntos y su evaluación se efectuará a partir de cinco exámenes escritos que constarán de ejercicios relativos a los temas tratados. Cada examen aportará el mismo porcentaje a la evaluación teórica y se valorará de 0 a 10 puntos. La preparación de cada examen se realizará por medio de las actividades formativas en equipo.

* Trabajo práctico en equipo

La escala que se utilizará para valorar el trabajo práctico es de 0 a 10 puntos y su evaluación se efectuará a partir de las prácticas de laboratorio que se irán planteando regularmente y de los informes solicitados. El enunciado de cada práctica y la solicitud de cada informe incluirá las especificaciones, plazos de entrega y porcentaje que aporta a la evaluación del trabajo práctico. Cada equipo deberá entregar, dentro de los plazos establecidos, los ficheros conteniendo el código fuente y demás información requerida, usando el medio que se establezca. La nota obtenida en cada práctica y en cada informe es la que le corresponderá a cada alumno componente del equipo que realiza la actividad.

* Coevaluación

Cada alumno interviene en la evaluación al calificar la participación activa en el trabajo colaborativo de sus compañeros de equipo por medio de un cuestionario que se valorará de 0 a 10 puntos. La nota de un alumno por cada coevaluación realizada será la media de la otorgada por sus compañeros de equipo. La nota final de esta parte será la media de los resultados obtenidos en las coevaluaciones efectuadas a lo largo del semestre y se valorará de 0 a 10 puntos. Requiere la asistencia a las sesiones académicas, con un mínimo del 85%, debido a la metodología orientada a los estilos de aprendizaje y a la coevaluación. Se realizarán controles de asistencia a las sesiones académicas.

2) Examen final

Aquellos alumnos que no superen o no completen la evaluación continua, teórica o práctica, podrán optar a un examen donde deberán demostrar sus conocimientos de la asignatura. El examen estará dividido en ejercicios, cada uno de los cuales aportará un porcentaje especificado a la nota. Se valorará de 0 a 10 puntos.

Criterios de calificacion:

Para superar la asignatura habrá que obtener una nota mínima de 5 puntos en la evaluación continua de la teoría, del trabajo práctico en equipo y en la coevaluación. La nota final se calcula con la siguiente fórmula:

$$NF = NT * 0.5 + NTP * 0.3 + NC * 0.2$$

Donde NF representa la nota final, NT la nota de teoría, NTP la nota del trabajo práctico en equipo y NC la nota de la coevaluación. NT y NTP serán sustituidas en la fórmula por las notas correspondientes obtenidas mediante el examen final para aquellos alumnos que opten por este sistema de evaluación. La nota final de los alumnos que no cumplan las condiciones establecidas para superar la asignatura será como máximo de 4 puntos.

Plan de Aprendizaje (Plan de trabajo de cada estudiante)

Tareas y actividades que realizará según distintos contextos profesionales (científico, profesional, institucional, social)

Estudio de la documentación específica y técnica de la materia.

Actividades formativas, que se irán planteando regularmente, consistentes en la realización de ejercicios en equipo que requieren solución individual para una posterior discusión colectiva.

Trabajo práctico consistente en la realización en equipo de las prácticas de laboratorio y de los informes que requieren solución individual para una posterior discusión colectiva.

Uso del entorno de desarrollo Eclipse.

Temporalización semanal de tareas y actividades (distribución de tiempos en distintas actividades y en presencialidad - no presencialidad)

Los estudiantes dedicarán 2 horas semanales a asistir a sesiones académicas teóricas y otras dos a sesiones académicas prácticas, que incluirán la realización de ejercicios de aplicación de las técnicas introducidas en la teoría y la solución problemas de programación (60 horas en 15 semanas). Los exámenes que se realizarán a lo largo del semestre consumen 5 horas presenciales en total. Además, cada equipo tendrá media hora presencial semanal de tutoría (7,5 horas en 15 semanas).

En promedio, realizarán cada semana 5 horas de trabajo no presencial dedicado a la consulta y asimilación de materiales documentales (2 horas en promedio) y la realización de ejercicios y trabajos (3 horas en promedio), ello supone 75 horas en 15 semanas.

Las horas restantes las distribuirá el estudiante, en función de sus necesidades particulares, entre atención tutorial individual, tiempo adicional de estudio y realización de ejercicios y trabajos.

Recursos que tendrá que utilizar adecuadamente en cada uno de los contextos profesionales.

Bibliografía, documentación específica sobre el contenido de la asignatura y documentación técnica.

Herramientas de desarrollo general para Java.

Herramientas del campus virtual de la ULPGC.

Resultados de aprendizaje que tendrá que alcanzar al finalizar las distintas tareas.

Aplicar las técnicas apropiadas para demostrar la correctitud de algoritmos iterativos y recursivos simples.

Aplicar técnicas adecuadas para analizar la eficiencia de los algoritmos

Explicar las estrategias básicas de diseño de algoritmos, seleccionar la más adecuada y adaptarla para resolver un problema concreto.

Explicar y distinguir los principales algoritmos de ordenación y de búsqueda de patrones en un texto.

Aplicar los conocimientos adquiridos a la resolución de problemas.

Plan Tutorial

Atención presencial individualizada (incluir las acciones dirigidas a estudiantes en 5ª, 6ª y 7ª convocatoria)

Se usará para aclarar dudas y asesorar al alumno en las tareas individuales.

Atención presencial a grupos de trabajo

Se programarán tutorías para cada equipo. Se usarán para aclarar dudas, orientar las actividades colectivas dirigidas, mejorar el rendimiento académico, hacer un seguimiento del trabajo realizado y modelar las habilidades sociales.

Atención telefónica

Los profesores podrán atender consultas telefónicas del alumnado en su horario de tutorías.

Atención virtual (on-line)

La atención virtual on-line se realizará haciendo uso de las herramientas del campus virtual de la ULPGC.

Bibliografía

[1 Básico] Metodología y tecnología de la programación II /

Margarita Díaz Roca, Juan Carlos Rodríguez del Pino.

Universidad de Las Palmas de Gran Canaria, Vicerrectorado de Calidad e Innovación Educativa,, Las Palmas de Gran Canaria : (2007) - (2ª ed.)

9788488412157

[2 Recomendado] Programación orientada a objetos con Java /

Francisco Durán, Francisco Gutiérrez, Ernesto Pimentel.

Thomson Paraninfo,, Madrid : (2007)

9788497325721

[3 Recomendado] Fundamentos de algoritmia /

G. Brassard, P. Bratley.

Prentice Hall,, Madrid : (1998)

848966000X

[4 Recomendado] Introducción al diseño y análisis de algoritmos: un enfoque estratégico /

R.C.T. Lee ... [et al.] ; revisión técnica: Miguel A. Orozco Malo, Jorge Valeriano Assem, Carlos Villegas Quezada.

McGraw-Hill,, México, D.F : (2007)

978-970-10-6124-4

[5 Recomendado] Diseño de programas. Formalismo y abstracción /

Ricardo Peña Marí.

..T250:

Prentice Hall,, MadridMadrid : (2004)

84-205-4191-5