

**40807 - FUNDAMENTOS DE
PROGRAMACIÓN**

CENTRO: 180 - Escuela de Ingeniería Informática

TITULACIÓN: 4008 - Grado en Ingeniería Informática

ASIGNATURA: 40807 - FUNDAMENTOS DE PROGRAMACIÓN

Vinculado a : (Titulación - Asignatura - Especialidad)

4801-Doble Grado en Ingeniería Informática y - 48107-FUNDAMENTOS DE PROGRAMACIÓN - 00

CÓDIGO UNESCO: 1203

TIPO: Básica de Rama

CURSO: 1

SEMESTRE: 2º semestre

CRÉDITOS ECTS: 6

Especificar créditos de cada lengua:

ESPAÑOL: 6

INGLÉS:

SUMMARY

REQUISITOS PREVIOS

Introducción a al Informática

Plan de Enseñanza (Plan de trabajo del profesorado)

Contribución de la asignatura al perfil profesional:

Fundamentos de programación abre las puertas al mundo de la programación, introduciendo al alumno en las estructuras de programación más complejas, de manera independiente al lenguaje escogido para su implementación. Así mismo, ejemplifica la implementación de algoritmos empleando un lenguaje de programación de propósito general, imperativo, estructurado y con controles de tipado.

Competencias que tiene asignadas:

G1, G2, G3, G4, G5, T1, T5, T6, T8, T9, T10, N1, N2, N3, N4, N5, FB04, FB05.

Objetivos:

(Ob1) Desarrollar las habilidades necesarias para desarrollar algoritmos para resolver problemas de dificultad baja y media, empleando para ello técnicas básicas de programación y estructuras de datos.

(Ob2) Implementar programas de ordenador en un lenguaje de programación de propósito general, que sean reflejo de las soluciones algorítmicas desarrolladas, adaptándose a las características de lenguaje en concreto.

Contenidos:

- 1 Depuración de programas
 - 1.1 Tipos de errores al programar
 - 1.2 Herramientas de depuración
 - 1.3 Estrategias de depuración
- 2 Ristras de caracteres
 - 2.1 Concepto y tipos de ristras
 - 2.2 Operaciones con ristras
 - 2.3 Tratamiento de ristras: recorrido y búsqueda
- 3 Programación modular
 - 3.1 Separación, ocultamiento y encapsulamiento
 - 3.2 Paquetes, clases y objetos
 - 3.3 Visibilidad: interfaz e implementación
- 4 Recursividad
 - 4.1 Subprogramas recursivos
 - 4.2 Caracterización de la recursividad
 - 4.3 Metodología de diseño de soluciones recursivas
- 5 Estructuras encadenadas
 - 5.1 Gestión de la memoria
 - 5.2 Concepto de encadenamiento
 - 5.3 Estructuras encadenadas
 - 5.4 Tratamiento de estructuras encadenadas: recorrido, inserción, extracción y búsqueda
- 6 Excepciones
 - 6.1 Errores durante la ejecución
 - 6.2 Bloques de control
 - 6.3 Ciclo de existencia de las excepciones
 - 6.4 Representación de las excepciones
 - 6.5 Robustez frente a excepciones
- 7 Ficheros
 - 7.1 Organización de la información en memoria secundaria
 - 7.2 Protocolo de transferencia con los ficheros
 - 7.3 Tipos de ficheros, modos de transferencia y formas de acceso
 - 7.4 Operaciones con ficheros
 - 7.5 Tratamiento de ficheros: recorrido, inserción y búsqueda

Para todos los temas se utilizarán como bibliografía los tres libros básicos de la asignatura: [1 Básico] [2 Básico] [3 Básico].

Todos los contenidos son de carácter teórico-práctico. En cada tema se realizarán ejercicios en clase que aplicarán los conocimientos teóricos antes expuestos. Por otra parte, en las clases de laboratorio, se desarrollarán trabajos de curso dirigidos consistentes en el desarrollo de diversos programas, su prueba y depuración que perseguirán que los alumnos adquieran destreza en el uso de un lenguaje de programación y habilidad en el empleo de los conceptos mostrados en clase.

Metodología:

La metodología a desarrollar incluirá:

Sesiones académicas en el aula con un enfoque participativo en las que:

AF1.- Se presentarán los puntos esenciales de la materia, se resolverán dudas y se estudiarán casos de ejemplo.

AF2.- Se desarrollarán ejercicios.

Sesiones académicas en el laboratorio, en las que:

AF3.-Se realizarán ejercicios de programación no evaluables con el fin de que los alumnos adquieran destreza en el uso de un lenguaje de programación y habilidad en el empleo de los conceptos mostrados en clase.

AF4.- Se realizarán ejercicios de programación evaluables presenciales con el fin de mostrar la consecución de las habilidades objeto de la materia.

AF5.- Se realizarán ejercicios de programación evaluables, propuestos y a desarrollar de manera no presencial, con el fin de desarrollar y mostrar las habilidades objeto de la materia,

AF6.- Asistencia tutorial cuando sea requerida.

Se hará uso del campus virtual de la ULPGC, procurando materiales, actividades y recursos en la página de la asignatura. En todo momento se potenciará la interacción electrónica entre los alumnos y el profesor.

Evaluación:

Criterios de evaluación

Las actividades de evaluación que se desarrollen tendrán por objeto valorar el grado de consecución de los objetivos de la asignatura y la adquisición por parte de los estudiantes de las competencias de la asignatura: T1, T5, T6, T8, T9, T10, FB04 y FB05. En todo caso, han de ser coherentes con las metodologías de enseñanza/aprendizaje.

Las actividades formativas en las que los estudiantes realicen algún tipo de trabajo tutorizado, individual o en grupo, de contenido teórico, práctico o teórico-práctico, tanto en el aula como en el laboratorio, serán evaluadas a partir del perfil de competencias dado por T1, T5, T6, T8, T9, T10, FB04 y FB05, mediante las siguientes fuentes de evaluación:

FE1.- El desempeño ante los problemas propuestos (a partir de los ejercicios de desarrollo presenciales que se irán evaluando regularmente o de un trabajo alternativo).

FE2.- Evaluación de los trabajos prácticos (a partir de los ejercicios de desarrollo no presencial que se irán evaluando regularmente o de un trabajo alternativo).

FE3.- Los exámenes (sobre los conocimientos tanto teóricos como prácticos de la asignatura).

Todas las fuentes de evaluación están relacionadas con todas las actividades formativas: AF1, AF2, AF3, AF4, AF5 y AF6.

La selección de criterios y fuentes de evaluación se ha realizado teniendo en cuenta las siguientes consideraciones:

Las actividades de evaluación que se desarrollen tendrán por objeto valorar el nivel de alcance de los objetivos de la asignatura y la adquisición por parte de los estudiantes de las competencias de la materia. En todo caso, han de ser coherentes con las metodologías de enseñanza/aprendizaje. Están diseñadas para medir el grado de consecución de los resultados del aprendizaje.

Con la finalidad de llevar a cabo una evaluación continuada y personalizada de la manera más idónea, que tenga en cuenta tanto los aspectos teóricos como los prácticos, se seleccionan los mecanismos más adecuados en función de las características de cada grupo de entre los múltiples recomendados en la amplia bibliografía sobre evaluación continua. En esta evaluación se atenderán aspectos como la asistencia y la participación activa en las sesiones, las pruebas teórico-prácticas y los trabajos individuales dirigidos.

Sistemas de evaluación

----- Convocatoria Ordinaria:

FE1) Evaluación de los trabajos prácticos presenciales y del desempeño con el lenguaje seleccionado. Cada uno de los desarrollos se valorará de 0 a 10 puntos y se efectuará a partir de los ejercicios de laboratorio que se irán planteando regularmente. El enunciado de cada ejercicio incluirá las especificaciones y plazos de entrega.

FE2) Evaluación de los trabajos prácticos no presenciales y del desempeño con el lenguaje seleccionado. Cada uno de los desarrollos se valorará de 0 a 10 puntos y se efectuará a partir de los ejercicios de laboratorio que se irán planteando regularmente. El enunciado de cada ejercicio incluirá las especificaciones y plazos de entrega.

Por cada desarrollo, el profesor tendrá en cuenta todos los aspectos implicados: realización, ejecución (funcionamiento, adecuación a las especificaciones, robustez,...) y estilo (formato, comentarios, elección de identificadores,...). Cuando lo estime conveniente, el profesor podrá citar al alumno para formularle cuestiones que considere relevantes para su valoración.

Convocatorias extraordinaria y especial:

FE3) Un examen de conjunto, en el que las cuestiones podrán ser de carácter teórico, práctico o teórico práctico.

Criterios de calificación

----- Convocatoria ordinaria:

Los trabajos prácticos evaluables presenciales (FE1) y los trabajos prácticos evaluables no presenciales (FE2) serán calificados atendiendo a su correcto funcionamiento y a la calidad del código, de acuerdo con los siguientes criterios:

- Cada ejercicio será calificado inicialmente con 10 puntos de un total de 10.
- Se descontarán 10 puntos si el ejercicio no compila, obteniéndose la calificación de cero (0).
- Se descontarán hasta 10 puntos por fallos de funcionamiento detectados.
- Se descontarán hasta 4 puntos por la inadecuación o excesiva complejidad de los algoritmos utilizados.

- Se descontarán hasta 1 punto por errores en el estilo de codificación.
- La nota mínima que se podrá obtener en un ejercicio será de cero puntos.

Sólo se podrá obtener una puntuación inferior a 2 puntos en uno de los trabajos prácticos evaluables presenciales; en caso contrario, la calificación final máxima (NF) será de 2.

La superación con un mínimo de 5 puntos de media de los trabajos prácticos evaluables presenciales (EVP) es condición para superar la asignatura; en caso contrario, la calificación final máxima será de 2.

En caso de superar los trabajos prácticos evaluables presenciales, la calificación final se obtendrá teniendo además en consideración la nota media obtenida en los trabajos prácticos evaluables no presenciales (EVNP). Se aplicará la siguiente fórmula:

$$NF = \text{MAX}((EVP * 0.8) + (EVNP * 0.2)) ; 5.0$$

Convocatorias extraordinaria y especial:

Un examen (FE3) que se calificará entre 0 y 10 puntos. Debe obtenerse una calificación mínima de 5 para superarlo. Si no se supera, la calificación final máxima será de 2.

Plan de Aprendizaje (Plan de trabajo de cada estudiante)

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

Ta1. Trabajos individuales de carácter práctico dirigidos que se irán planteando regularmente. El objetivo de estas actividades es desarrollar las destrezas de los estudiantes en la implementación de programas que solucionen problemas concretos en un lenguaje de programación.

Ta2. Por otra parte, el estudiante habrá de emplear parte de su tiempo en el estudio de los contenidos de la asignatura y en la preparación de las pruebas objetivas y exámenes que haya de realizar.

Estas tareas contribuyen de manera significativa a la formación del alumno en cualquiera de los contextos en los que pudiera desarrollar su actividad profesional.

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 en el aula y otras 2 a sesiones académicas en el laboratorio.

Cada semana dedicarán 5,5 horas de trabajo no presencial a la consulta y asimilación de materiales documentales (2,5 horas) y a la realización de ejercicios y trabajos (3 horas).

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

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

- Re1. Ordenador
- Re2. Navegador web
- Re3. Entorno Integrado de Desarrollo

Estos recursos redundan en la formación del alumno en cualquiera de los contextos en los que pudiera desarrollar su actividad profesional.

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

RA1. Seleccionar los esquemas algorítmicos básicos de recorrido y búsqueda y aplicarlos de forma adecuada para resolver problemas concretos.

RA2. Describir los mecanismos de almacenamiento de la información en ficheros y usarlos de forma adecuada en el desarrollo de programas.

RA3. Describir los mecanismos de almacenamiento de la información en memoria dinámica y usarlos de forma adecuada en el desarrollo de programas.

RA4. Describir los elementos básicos que proporciona un entorno integrado de desarrollo y usarlos de manera adecuada en la implementación de programas.

RA5. Explicar el concepto de excepción y manejarlo de forma adecuada en el desarrollo de programas.

RA6. Manejar las técnicas y herramientas básicas de depuración y usarlas de forma adecuada para seguir la evolución de la ejecución de un programa.

RA7. Explicar el concepto de recursividad y aplicarlo de forma adecuada para diseñar e implementar soluciones recursivas a problemas simples.

RA8. Describir el concepto de estructura encadenada y usarlo de manera adecuada en el desarrollo de programas.

RA9. Justificar la utilidad de los elementos de estilo y usarlos de forma adecuada en el desarrollo de programas.

Todas las actividades formativas (AF1, AF2, AF3, AF4, AF5 y AF6) contribuyen a adquirir todos y cada uno de estos resultados del aprendizaje.

Plan Tutorial

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

La atención presencial individualizada se realizará mediante el método de cita previa con 48 horas de antelación, preferiblemente mediante las herramientas proporcionadas por el campus virtual, y se usará para aclarar dudas al alumno y controlar y evaluar las actividades individuales dirigidas.

Atención presencial a grupos de trabajo

La atención presencial a grupos de trabajo se realizará mediante el método de cita previa con 48 horas de antelación, preferiblemente mediante las herramientas proporcionadas por el campus virtual, y se usará para aclarar dudas y para la monitorización y evaluación de los trabajos de grupo dirigidos.

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 o mediante el correo electrónico institucional.

Datos identificativos del profesorado que la imparte.

Datos identificativos del profesorado que la imparte

D/Dña. Juan de Dios Duque Martín de Oliva (COORDINADOR)

Departamento: 260 - INFORMÁTICA Y SISTEMAS

Ámbito: 570 - Lenguajes Y Sistemas Informáticos

Área: 570 - Lenguajes Y Sistemas Informáticos

Despacho: INFORMÁTICA Y SISTEMAS

Teléfono: 928458735 **Correo Electrónico:** juandedios.duque@ulpgc.es

D/Dña. Miguel Ángel Pérez Aguiar

Departamento: 260 - INFORMÁTICA Y SISTEMAS

Ámbito: 570 - Lenguajes Y Sistemas Informáticos

Área: 570 - Lenguajes Y Sistemas Informáticos

Despacho: INFORMÁTICA Y SISTEMAS

Teléfono: 928458741 **Correo Electrónico:** miguelangel.perez@ulpgc.es

Dr./Dra. José Fortes Gálvez

Departamento: 260 - INFORMÁTICA Y SISTEMAS

Ámbito: 570 - Lenguajes Y Sistemas Informáticos

Área: 570 - Lenguajes Y Sistemas Informáticos

Despacho: INFORMÁTICA Y SISTEMAS

Teléfono: 928458724 **Correo Electrónico:** jose.fortes@ulpgc.es

D/Dña. José Daniel González Domínguez

Departamento: 260 - INFORMÁTICA Y SISTEMAS

Ámbito: 570 - Lenguajes Y Sistemas Informáticos

Área: 570 - Lenguajes Y Sistemas Informáticos

Despacho: INFORMÁTICA Y SISTEMAS

Teléfono: 928458751 **Correo Electrónico:** josedaniel.gonzalez@ulpgc.es

D/Dña. Gustavo Rodríguez Rodríguez

Departamento: 260 - INFORMÁTICA Y SISTEMAS

Ámbito: 570 - Lenguajes Y Sistemas Informáticos

Área: 570 - Lenguajes Y Sistemas Informáticos

Despacho: INFORMÁTICA Y SISTEMAS

Teléfono: 928458734 **Correo Electrónico:** gustavo.rodriguez@ulpgc.es

Dr./Dra. Octavio Santana Suárez

Departamento: 260 - INFORMÁTICA Y SISTEMAS

Ámbito: 570 - Lenguajes Y Sistemas Informáticos

Área: 570 - Lenguajes Y Sistemas Informáticos

Despacho: INFORMÁTICA Y SISTEMAS

Teléfono: 928458730 **Correo Electrónico:** octavio.santana@ulpgc.es

Bibliografía

[1 Básico] Java 7 /

Herbert Schildt.

Anaya Multimedia,, Madrid : (2011)

978-84-415-3067-6

[2 Básico] The Java tutorials [

Oracle and /or its affiliates.

Oracle,, [s.l.] : (1995)

[3 Básico] Fundamentos de estructuras de datos: soluciones en Ada, Java y C++ /

Zenón José Hernández Figueroa, Juan Carlos Rodríguez del Pino, José Daniel González Domínguez, Margarita Díaz Roca, José Rafael Pérez Aguilar, Gustavo Rodríguez Rodríguez ... [et al.].

Thomson,, Australia, España [etc.] : (2005)

84-9732-358-0

[4 Recomendado] Piensa en Java /

Bruce Eckel.

Prentice Hall,, Madrid [etc.] : (2007) - (4ª ed.)

9788489660342

[5 Recomendado] Una introducción a la programación: un enfoque algorítmico /

Jesús J. García Molina, ...[et al.].

Thomson,, Madrid : (2005)

8497321855

[6 Recomendado] Estructura de datos: algoritmos, abstracción y objetos /

Luis Joyanes Aguilar, Ignacio Zahonero Martínez.

84-481-2042-6

[7 Recomendado] Iniciación a la programación Ada 2005 como primer lenguaje /

Zenón

J. Hernández Figueroa, Francisco J. Carreras Riudavets, Gustavo Rodríguez Rodríguez, José Daniel González Domínguez.

Lulu.com,, [s.l.] : (2009) - (2ª ed.)

9781409268000