



UNIVERSIDAD DE LAS PALMAS  
DE GRAN CANARIA

GUÍA DOCENTE

CURSO: 2012/13

## 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

**Datos identificativos del profesorado que la imparte.**

**Plan de Enseñanza (Plan de trabajo del profesorado)**

### Contribución de la asignatura al perfil profesional:

Fundamentos de programación es la asignatura que abre las puertas al mundo de la programación, utilizando un lenguaje de programación de propósito general con tipado fuerte.

### Competencias que tiene asignadas:

La asignatura contribuirá a lograr las siguientes competencias:

G1. Poseer y comprender conocimientos en un área de estudio (Ingeniería Informática) que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio;

G2. 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;

G3. Reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética;

G4. Transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado;

G5. Desarrollar aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

N1. Comunicarse de forma adecuada y respetuosa con diferentes audiencias (clientes, colaboradores, promotores, agentes sociales, etc.), utilizando los soportes y vías de comunicación más apropiados (especialmente las nuevas tecnologías de la información y la comunicación) de modo que pueda llegar a comprender los intereses, necesidades y preocupaciones de las personas y organizaciones, así como expresar claramente el sentido de la misión que tiene encomendada y la forma en que puede contribuir, con sus competencias y conocimientos profesionales, a la satisfacción de esos intereses, necesidades y preocupaciones.

N2. Cooperar con otras personas y organizaciones en la realización eficaz de funciones y tareas propias de su perfil profesional, desarrollando una actitud reflexiva sobre sus propias competencias y conocimientos profesionales y una actitud comprensiva y empática hacia las competencias y conocimientos de otros profesionales.

N3. Contribuir a la mejora continua de su profesión así como de las organizaciones en las que desarrolla sus prácticas a través de la participación activa en procesos de investigación, desarrollo e innovación.

N4. Comprometerse activamente en el desarrollo de prácticas profesionales respetuosas con los derechos humanos así como con las normas éticas propias de su ámbito profesional para generar confianza en los beneficiarios de su profesión y obtener la legitimidad y la autoridad que la sociedad le reconoce.

N5. Participar activamente en la integración multicultural que favorezca el pleno desarrollo humano, la convivencia y la justicia social.

T1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.

T5. Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad.

T6. Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes.

T8. Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

T9. Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

T10. Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática.

FB04. Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación de la ingeniería.

FB05. Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de la programación, y su aplicación para la resolución de problemas propios de la ingeniería.

## Objetivos:

Se pretende el alumno que complete esta asignatura sea capaz de:

- Seleccionar los esquemas algorítmicos básicos de recorrido y búsqueda y aplicarlos de forma adecuada para resolver problemas concretos.
- Describir los mecanismos de almacenamiento de la información en ficheros y usarlos de forma adecuada en el desarrollo de programas.
- Describir los mecanismos de almacenamiento de la información en memoria dinámica y usarlos de forma adecuada en el desarrollo de programas.
- Describir los elementos básicos que proporciona un entorno integrado de desarrollo y usarlos de manera adecuada en la implementación de programas.

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

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

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

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

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

## Contenidos:

### 1 DEPURACIÓN DE PROGRAMAS

- 1.1 introducción
- 1.2 tipos de errores
- 1.3 herramientas de depuración
- 1.4 depurador
- 1.5 estrategia de depuración

### 2 PROGRAMACIÓN MODULAR

- 2.1 separación
- 2.2 ocultamiento
- 2.3 encapsulamiento
- 2.4 interfaz e implementación

### 3 RISTRAS DE CARACTERES

- 3.1 introducción
- 4.2 concepto y tipos de ristra
- 4.3 operaciones con ristras
- 4.4 entrada y salida de ristras de caracteres
- 4.5 tratamiento de ristras: recorrido y búsqueda

### 5 RECURSIVIDAD

- 5.1 introducción

- 5.2 caracterización de la recursividad
- 5.3 subprogramas recursivos
- 5.4 metodología de diseño de soluciones recursivas
- 5.5 inmersión

## 6 ESTRUCTURAS ENCADENADAS

- 6.1 introducción
- 6.2 gestión dinámica de la memoria
- 6.2 concepto de encadenamiento
- 6.3 listas encadenadas
- 6.4 tratamiento de listas: recorrido, inserción, extracción y búsqueda

## 7 EXCEPCIONES

- 7.1 introducción
- 7.2 representación de las excepciones
- 7.3 bloques de control
- 7.4 ciclo de existencia de las excepciones
- 7.5 robustez frente a excepciones

## 7 FICHEROS

- 7.1 introducción
- 7.2 organización de la información en memoria secundaria
- 7.3 protocolo de transferencia con los ficheros
- 7.4 tipos de ficheros
- 7.5 modos de transferencia y formas de acceso
- 7.6 operaciones con ficheros
- 7.7 excepciones relacionadas con el manejo de ficheros
- 7.8 tratamiento de ficheros: recorrido, inserción y búsqueda

### **Metodología:**

La metodología a desarrollar incluirá sesiones académicas teóricas y prácticas, realización de ejercicios, y asistencia tutorial cuando sea requerida. Se fomentará el aprendizaje cooperativo a través de grupos de trabajo.

### **Criterios y fuentes para la evaluación:**

La evaluación se basará en las soluciones de los ejercicios de programación propuestos por el profesor y entregados por los estudiantes como culminación de las distintas actividades enumeradas en el apartado de metodología.

Como criterio general, se distinguirá entre los ejercicios realizados dentro de una sesión académica, con la presencia del profesor (ejercicios presenciales) y los realizados con libertad horaria, dentro de unos plazos amplios (ejercicios no presenciales).

Para superar la asignatura se exigirá alcanzar una nota mínima en los ejercicios presenciales, tal como se detalla en el apartado de criterios de calificación.

La participación activa del estudiante, concretada por la participación en cuestionarios y encuestas junto con las intervenciones relevantes en los foros de la asignatura, será tomada en cuenta para estimar la excelencia. Se consideran intervenciones relevantes aquellas que susciten debate o den respuestas originales y precisas.

## Sistemas de evaluación:

La evaluación será continua, a lo largo de todo el semestre, no existiendo ningún examen ni prueba para completarla en el periodo reservado para la realización de la convocatoria ordinaria al final del semestre.

En las convocatorias extraordinaria y especial se realizará una prueba, consistente en la realización en presencia del profesor de un conjunto de ejercicios de programación. Los resultados de esta prueba sustituirán a la nota correspondiente a los ejercicios presenciales realizados durante el semestre. La nota correspondiente a los ejercicios no presenciales será la obtenida durante el periodo de impartición de la asignatura.

Los estudiantes a tiempo parcial que por circunstancias especiales no puedan seguir la asignatura en los términos del proyecto docente y tengan un modelo de evaluación diferenciado, tal y como se recoge en el artículo 17 del "Reglamento de evaluación de los resultados de aprendizaje y de las competencias adquiridas por el alumnado en los títulos oficiales, títulos propios y de formación continua de la ULPGC" tendrán derecho a un examen global de la asignatura en los términos del citado artículo.

Se facilitará a los estudiantes con diversidad funcional la realización de pruebas y exámenes en condiciones acordes con sus necesidades específicas, con el asesoramiento del órgano o unidad competente en materia de diversidad funcional, en los términos establecidos en el artículo 16 del "Reglamento de evaluación de los resultados de aprendizaje y de las competencias adquiridas por el alumnado en los títulos oficiales, títulos propios y de formación continua de la ULPGC"

## Criterios de calificación:

Los ejercicios de programación 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 que serán testeados mediante pruebas de unidad de caja negra.
- Se descontarán hasta 4 puntos por la inadecuación o excesiva complejidad de los algoritmos utilizados.
- Se descontarán hasta 2 puntos por errores en el estilo de codificación, atendiendo a las recomendaciones de estilo establecidas y publicadas al inicio del semestre.
- La nota mínima que se podrá obtener en un ejercicio será de cero puntos.

Los ejercicios presenciales aportarán hasta 6 puntos de los 10 que se pueden obtener como máximo. Para superar la asignatura se han de obtener al menos 3,5 puntos con los ejercicios presenciales.

Siempre que se hayan obtenido al menos 3,5 puntos con los ejercicios presenciales, se sumará a la calificación la nota obtenida con los ejercicios no presenciales, ponderada en el rango 0-4. Si no se ha obtenido el mínimo de 3,5 puntos con los ejercicios presenciales, no se sumará nada más para obtener la calificación final.

A aquellos estudiantes que obtengan una calificación final igual o superior a 9,5 puntos se les hará una valoración global del trabajo realizado para estimar la posibilidad de otorgarles la mención de "Matricula de Honor", dentro de los límites establecidos por la ULPGC. En esta valoración global se incluirán factores cualitativos, como la creatividad de las soluciones desarrolladas o la participación activa; estos factores no aportarán puntuación adicional sino que

se usarán exclusivamente para clasificar a los estudiantes destacados cara al reconocimiento de la excelencia.

## **Plan de Aprendizaje (Plan de trabajo de cada estudiante)**

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

Asistencia a sesiones académicas.  
Consulta y asimilación de materiales documentales.  
Realización de ejercicios y trabajos.  
Tutorías.

### **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 y solución problemas de programación (60 horas en 15 semanas).

En promedio, realizarán cada semana 5,5 horas de trabajo no presencial dedicados a la consulta y asimilación de materiales documentales (2,5 horas en promedio) y la realización de ejercicios y trabajos (3 horas en promedio), ello supone 82,5 horas en 15 semanas.

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.**

Ordenador  
Navegador web  
Entorno Integrado de Desarrollo

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

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

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

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

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

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

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

Explicar el concepto de recursividad y aplicarlo de forma adecuada para diseñar e implementar

soluciones recursivas a problemas simples.

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

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

## Plan Tutorial

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

La atención personal individualizada se verificará en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas. Se favorecerá la cita previa utilizando los mecanismos disponibles en el Campus Virtual de la ULPGC. La atención personal individualizada se verificará en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas. Se favorecerá la cita previa utilizando los mecanismos disponibles en el Campus Virtual de la ULPGC.

### Atención presencial a grupos de trabajo

La atención personal a grupos de trabajo se verificará en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas. Se favorecerá la cita previa utilizando los mecanismos disponibles en el Campus Virtual de la ULPGC.

### Atención telefónica

La atención telefónica se verificará en los teléfonos del profesorado publicados en la web del Departamento de Informática y Sistemas, en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas.

### Atención virtual (on-line)

La atención virtual (online) se llevará a cabo usando las herramientas disponibles en el Campus Virtual de la ULPGC. Se responderá a las consultas virtuales en el horario de tutorías establecido para los profesores de la asignatura y publicado en la web del Departamento de Informática y Sistemas.

### Bibliografía

---

#### [1 Recomendado] Java 7 /

*Herbert Schildt.*

*Anaya Multimedia,, Madrid : (2011)*

*978-84-415-3067-6*

---

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

*Jesús J. García Molina, ...[et al.].*

*Thomson,, Madrid : (2005)*

*8497321855*

---

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

*Luis Joyanes Aguilar, Ignacio Zahonero Martínez.*

84-481-2042-6

---

**[4 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