

**CENTRO UNIVERSITARIO DE TECNOLOGÍA Y ARTE DIGITAL**



**PLANIFICACIÓN DE LA DOCENCIA  
UNIVERSITARIA**

**GUÍA DOCENTE**

**INTRODUCCIÓN A LA PROGRAMACIÓN I**

# 1. DATOS DE IDENTIFICACIÓN DE LA ASIGNATURA.

|                                |  |
|--------------------------------|--|
| Título:                        | Grado en Ingeniería del Software   |
| Facultad:                      | Centro Universitario de Tecnología y Arte Digital (U-TAD)  |
| Materia:                       | Fundamentos Informáticos   |
| Denominación de la asignatura: | Introducción a la Programación I   |
| Curso:                         | 1  |
| Cuatrimestre:                  | 1  |
| Carácter:                      | Básica   |
| Créditos ECTS:                 | 6  |
| Modalidad/es de enseñanza:     | Híbrido Presencial   |
| Idioma:                        | Castellano   |
| Profesor/a:                    | Alfonso Castro Escudero<br>Marcos Novalbos Mendiguchía   |
| E-mail:                        | <a href="mailto:alfonso.castro@u-tad.com">alfonso.castro@u-tad.com</a><br><a href="mailto:marcos.novalbos@u-tad.com">marcos.novalbos@u-tad.com</a> |
| Teléfono:                      |  |

## 2. DESCRIPCIÓN DE LA ASIGNATURA.

### 2.1 Descripción de la materia

Esta asignatura pertenece a la materia de Fundamentos informáticos, donde se pretende que el estudiante adquiera conocimientos básicos en los que se fundamenta la ingeniería del software.

### 2.2 Descripción de la asignatura

La asignatura de introducción a la programación I es el primer punto de contacto de los alumnos con la creación estructurada de programas. Para ello se ha elegido C como lenguaje de programación, debido a su eficiencia a la hora de crear código con lo que el alumno puede ser capaz de entender las acciones que se desencadenan al escribir una línea de código. Se revisarán los conceptos básicos de la programación como son los tipos de datos, operadores y expresiones y el modularidad.

### 3. COMPETENCIAS

#### Competencias Básicas y Generales

CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio 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.

CB2: 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.

CB3 - Que los estudiantes tengan la capacidad de 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.

CB4: Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía

CG1 - Capacidad para entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas.

CG3 - Conocimiento de los fundamentos científicos aplicables a la resolución de problemas informáticos

CG5 - Gestión de los recursos humanos y tecnológicos para la correcta realización de proyectos informáticos

CG9 - Capacidad para aprender, modificar y producir nuevas tecnologías informáticas

#### Competencias Específicas

CE3 - Conocimiento del álgebra relacional y realización de consultas en lenguajes procedurales para el diseño de esquemas de bases de datos normalizados basados en modelos de entidad-relación

CE9 - Conocimiento de las estructuras de control, variables, sintaxis de programación y gestión del uso de la memoria de manera eficaz en el desarrollo de una aplicación informática

CE11 - Conocimiento de la arquitectura de los Sistemas Operativos así como los distintos mecanismos para la gestión de procesos, comunicación y sincronización de los mismos

CE13 - Conocimiento de los fundamentos de las redes de ordenadores, de las distintas topologías y de sus protocolos de comunicación

CE16 - Conocimiento del funcionamiento

## 4. CONTENIDOS

### **Tema 1. Fundamentos de programación**

- 1.1.Fases de resolución de problemas
- 1.2.Programación estructurada

### **Tema 2. El lenguaje C. Elementos básicos**

- 2.1.Estructura general de un programa en C
- 2.2.Tipos de datos C
- 2.3.Constantes
- 2.4.Variables

### **Tema 3. Operadores y expresiones.**

- 3.1.Operador de asignación
- 3.2.Operadores aritméticos
- 3.3.Operadores de incremento y decremento
- 3.4.Operadores lógicos
- 3.5.Operadores de manipulación de bits
- 3.6.Operador condicional
- 3.7.Conversiones de tipos

### **Tema 4. Estructuras de selección y control**

- 4.1.Estructura de control
- 4.2.If con una alternativa
- 4.3.If con dos alternativas: if-else
- 4.4.If\_else anidados
- 4.5.Sentencia de control: switch
- 4.6.Sentencia while
- 4.7.Repetición: bucle for
- 4.8.Repetición: bucle do .. while

### **Tema 5. Arrays y cadenas**

- 5.1.Arrays unidimensionales
- 5.2.Cadenas
- 5.3.Arrays multidimensionales

### **Tema 6. Punteros**

- 6.1.Variables puntero
- 6.2.Los operadores de punteros
- 6.3.Operaciones con punteros
- 6.4.Punteros y arrays

### **Tema 7. Funciones**

- 7.1.Concepto de una función
- 7.2.Prototipo de una función
- 7.3.Parámetros formales y reales
- 7.4.Instrucción return
- 7.5.Paso de argumentos a funciones
  - Paso por valor
  - Paso por referencia
- 7.6.Ámbito y clases de almacenamiento
- 7.7.Recursividad

## 5. ACTIVIDADES FORMATIVAS Y MODALIDADES DE ENSEÑANZAS

### 5.1 Modalidades de enseñanza

La asignatura se desarrollará a través de los siguientes métodos y técnicas generales, que se aplicarán diferencialmente según las características propias de la asignatura:

- **Método expositivo/Lección magistral:** el profesor desarrollará, mediante clases magistrales y dinámicas los contenidos recogidos en el temario.
- **Estudio de casos:** análisis de casos reales relacionados con la asignatura.
- **Resolución de ejercicios y problemas:** los estudiantes desarrollarán las soluciones adecuadas aplicando procedimientos de transformación de la información disponible y la interpretación de los resultados.
- **Aprendizaje basado en problemas:** utilización de problemas como punto de partida para la adquisición de conocimientos nuevos.
- **Aprendizaje orientado a proyectos:** se pide a los alumnos que, en pequeños grupos, planifiquen, creen y evalúen un proyecto que responda a las necesidades planteadas en una determinada situación.
- **Aprendizaje cooperativo:** Los estudiantes trabajan en grupo para realizar las tareas de manera colectiva.

### 5.2 Actividades formativas

| Actividad Formativa                                     | Horas | Presencialidad |
|---|-------|----------------|
| AF1 Clases teóricas / Expositivas                       | 45    | 100%           |
| AF2 Clases Prácticas                                    | 36    | 100%           |
| AF3 Tutorías  | 6     | 50%            |
| AF4 Estudio independiente y trabajo autónomo del alumno | 57,5  | 0%             |
| AF5 Elaboración de trabajos (en grupo o individuales)   | 28,5  | 0%             |
| AF6: Actividades de Evaluación                          | 9     | 100%           |

## 6. DESARROLLO TEMPORAL

| UNIDADES DIDÁCTICAS / TEMAS   | PERÍODO TEMPORAL |
|---|------------------|
| Tema 1. Fundamentos de programación<br>Tema 2. El lenguaje C. Elementos básicos<br>Tema 3. Operadores y expresiones<br>Tema 4. Estructuras de selección y control | 5 semanas        |
| Tema 5. Arrays y cadenas  | 3 semanas        |
| Tema 6. Punteros  | 3 semanas        |
| Tema 7. Funciones.  | 3 semanas        |

## 7. SISTEMA DE EVALUACIÓN

| ACTIVIDAD DE EVALUACIÓN   | VALORACIÓN MÍNIMA<br>RESPECTO A LA CALIFICACIÓN<br>FINAL (%) | VALORACIÓN MÁXIMA<br>RESPECTO A LA CALIFICACIÓN<br>FINAL (%) |
|---|--|--|
| SE1 Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura | 0%   | 30%  |
| SE2 Evaluación de trabajos, proyectos, informes, memorias                                 | 30%  | 80%  |
| SE3 Prueba Objetiva   | 10%  | 60%  |

## 8. CRITERIOS DE EVALUACIÓN

| ACTIVIDAD DE EVALUACIÓN   | VALORACIÓN RESPECTO A LA CALIFICACIÓN FINAL (%) |
|---|---|
| Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura | 10%   |
| Evaluación de trabajos, proyectos, informes, memorias                                 | 30%   |
| Prueba Objetiva   | 60%   |

- La evaluación de la participación en clase, en prácticas o en proyectos de la asignatura se realizará a partir de la asistencia y la participación activa en clase y en el resto de las actividades desarrolladas durante el curso. Este aspecto representará el 10% de la calificación final de la asignatura en la convocatoria ordinaria.
- A lo largo del curso se plantearán actividades, ejercicios y problemas que deberán ser entregadas antes de la fecha indicada a través de la plataforma virtual. Este trabajo se evaluará a través de la propia plataforma virtual y supondrá un 30% de la calificación final de la asignatura en la convocatoria ordinaria. No se admitirán trabajos fuera de forma y fecha sin causa justificada. En caso de que la nota de un ejercicio no alcance el aprobado, se admitirá una nueva entrega que si tiene la suficiente calidad tendrá una calificación de 5.0.
- Para aprobar la asignatura en la convocatoria ordinaria, es imprescindible que la nota final sea al menos 5.0 (sobre 10). Además de ese requisito, es necesario que la media de los trabajos planteados sea al menos 5.0 (sobre 10), y que la nota del examen final sea al menos de 4.0 (sobre 10). **En caso de no cumplirse alguno de estos requisitos, la asignatura se considerará automáticamente suspensa independientemente del resto de calificaciones.**
- En caso de no conseguir el aprobado en la convocatoria ordinaria, el alumno podrá presentarse a la convocatoria extraordinaria. Se realizará un examen final que representará el 70% de su calificación en dicha convocatoria, y en el que formará parte de la materia exigible al alumno todo el contenido de la asignatura visto en clase. El otro 30% se calculará a partir de un ejercicio o ejercicios práctico entregable que será planteado al comienzo del comienzo de segundo cuatrimestre. Para aprobar la asignatura en la convocatoria extraordinaria, es imprescindible que la nota final sea al menos 5.0 (sobre 10). Además de ese requisito, es necesario que la media de los trabajos planteados sea al menos 5.0 (sobre 10), y que la nota del examen final sea al menos de 4.0 (sobre 10).
- En los exámenes no se permite el uso de apuntes ni de calculadoras científicas programables, para lo que el alumno debe remitirse a las instrucciones específicas del profesor sobre este tema.



- Todo el código y trabajos entregados por los alumnos deberán ser ORIGINALES. Quiere decir que deberán haber sido desarrollados por los alumnos a lo largo de la asignatura, sin ayuda externa. En caso de usar código/librerías externas a lo suministrado por el profesor, deberá estar debidamente documentado y justificado. Se permite consultar documentación externa a la asignatura, pero el código entregado por el alumno deberá respetar las leyes de copyright y licencias software vigentes. En todo caso, el alumno deberá ser capaz de explicar el código usado y entregado durante el curso.
- **Copias entre trabajos:** Se entenderá como copia de trabajo aquellos proyectos que contengan partes iguales o muy similares, que no cumplan las reglas establecidas en los párrafos anteriores. Las copias de trabajos conllevarán la completa suspensión de la asignatura, sin posibilidad de recuperación en la convocatoria actual. Será el profesor el que decida la gravedad de la copia, y la decisión final podrá ser consultada y revocada por el resto del equipo docente en caso de necesitar una segunda opinión.
- No se conservarán calificaciones de ningún tipo entre distintos cursos académicos, ni entre distintas convocatorias.
- No está permitido el uso de teléfonos móviles en el aula durante el período de evaluación continua, excepto indicación expresa en sentido contrario del profesor. Los ordenadores portátiles podrán utilizarse únicamente para actividades relacionadas con la asignatura. El profesor podrá retirar el derecho al uso del ordenador a aquellos alumnos que lo utilicen para actividades que no estén relacionadas con la asignatura (consulta de correos, noticias o redes sociales, consulta o elaboración de actividades de otras asignaturas, etc.).
- No está permitido consumir bebidas ni comidas en el aula. Tampoco está permitida la presencia de cualquier tipo de bebida en las mesas, incluso en envases cerrados.
- Se demandará del alumno una participación activa, necesaria para el desarrollo de las clases.
- Se exigirá al alumno un buen comportamiento en todo momento durante el desarrollo de las clases. El mal comportamiento que impida el normal desarrollo de la clase puede conllevar la expulsión del aula por un tiempo a determinar por el profesor.

## 9. BIBLIOGRAFÍA / WEBGRAFÍA

### Bibliografía Básica:

- David Griffiths, Dawn Griffiths, Head First C, O'Reilly Media, ISBN: 978-1-4493-9991-7
- Tony Gaddis, Starting Out with C++. Addison-Wesley. ISBN-13: 978-0132576253
- Jesse Liberty, C++ (Programación). Anaya Multimedia. ISBN-13: 978-844159793
- Bjarne Stroustrup, Programming: Principles and Practice Using C++. Addison-Wesley Professional. ISBN-13: 978-0321543721
- Nell B. Dale, Programming and Problem Solving with C++: Comprehensive Edition. Jones Bartlett Publishers. ISBN-13: 978-0763771560
- Bjarne Stroustrup, El lenguaje de programación C++. Addison-Wesley. ISBN-13: 9788478290468
- Luis Joyanes, Fundamentos de programación. McGraw-Hill Interamericana de España S.L. ISBN-13: 978-8448161118

### Bibliografía Recomendada:

- Brian W. Kernighan, Dennis M. Ritchie, El lenguaje de Programación C. Prentice Hall, ISBN 9789688802052
- Bruce Eckel, Thinking in C++. Prentice Hall. ISBN-13: 978-0139798092
- Andrei Alexandrescu, Modern C++ Design: Generic Programming and Design Patterns Applied. Addison-Wesley Professional. ISBN-13: 978-0201704310
- Herb Sutter, More Exceptional C++: 40 New Engineering Puzzles, Programming Problems, and Solutions. Addison-Wesley Professional. ISBN-13: 978-0201704341
- Scott Erich Gamma, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional. ISBN-13: 978-0201633610
- Meyers, Effective C++: 55 Specific Ways to Improve Your Programs and Designs. Addison-Wesley Professional. ISBN-13: 978-0321334879.

## 6. MATERIAL, SOFTWARE Y HERRAMIENTAS NECESARIAS

Cygwin: Emulador de Linux en Windows  
Editor de texto: Notepad ++