

DATOS DE IDENTIFICACIÓN

Titulación:	Grado en Ingeniería Informática		
Rama de Conocimiento:	Ingeniería y Arquitectura		
Facultad/Escuela:	Escuela Politécnica Superior		
Asignatura:	Introducción a la Programación		
Tipo:	Formación Básica	Créditos ECTS:	6
Curso:	1	Código:	5615
Periodo docente:	Primer semestre		
Materia:	Informática		
Módulo:	Formación Básica		
Tipo de enseñanza:	Presencial		
Idioma:	Castellano		
Total de horas de dedicación del alumno:	150		

Equipo Docente	Correo Electrónico
Roberto Rodríguez Galán	roberto.rodriguez@ufv.es
Jorge Antonio Cisneros González	j.cisneros.prof@ufv.es

DESCRIPCIÓN DE LA ASIGNATURA

La Asignatura Introducción a la Programación introduce al alumno en los conceptos de programación básicos, haciendo especial hincapié en la presentación de las técnicas algorítmicas esenciales y la utilización de diferentes paradigmas, lenguajes y entornos de programación para la construcción de aplicaciones informáticas, aplicando metodologías de manera rigurosa que proporcionen al alumno buenos hábitos en el desarrollo de software.

Esta asignatura corresponde al módulo Formación Básica y, dentro de éste, a la materia Informática. Se imparte en el primer semestre del primer curso de los estudios de Grado en Ingeniería Informática, y requiere de una dedicación de 150 horas por parte del alumno.

La asignatura se centra en la resolución de problemas, construyendo programas que resuelvan problemas sencillos. Para ello parte del diseño del algoritmo, utilizando estructuras modulares y distintos métodos de programación, y alcanza tanto la implementación del mismo en un lenguaje de programación estructurada, utilizando estructuras de control fundamentales de la codificación y tipos de datos básicos, como su posterior compilación, depuración y documentación.

OBJETIVO

El objetivo de la asignatura es capacitar al estudiante para resolver problemas de forma algorítmica; es decir, generar el pensamiento algorítmico de un futuro ingeniero informático. Esta resolución sistemática de un problema conlleva el análisis del mismo, el diseño de los pasos a seguir para su resolución (diseño algorítmico) y la codificación de una solución informática siguiendo las reglas básicas de la ciencia de la computación. Aunque el objetivo no es que aprendan un lenguaje de programación, para poder desarrollar los conceptos asociados a la asignatura se utilizará, principalmente, el lenguaje de programación C. El objetivo subyacente es que el alumno se empiece a familiarizar con un lenguaje de propósito general, que reglas estrictas de programación.

Los fines específicos de la asignatura son:

Comprender los conceptos básicos de la programación:

- Programa informático como la suma de Algoritmo y tratamiento de Datos.
- Lenguajes de programación, Entornos de Programación y Paradigmas de Programación.

Ser capaz de resolver un problema aplicando el pensamiento algorítmico. Ser capaz de implementar la solución en una lenguaje de programación. Ser capaz de depurar y mejorar la solución.

Conocer y saber utilizar los elementos básicos de la programación: tipos de datos y alcance, operaciones, variables y operadores.

Conocer y saber utilizar las estructuras de control: selección e iteración.

Conocer y saber utilizar las estructuras de datos básicas y sus combinaciones: arrays, cadenas, diccionarios, etc.

Conocer y saber utilizar otras características básicas de la programación: recursividad, paso de parámetros por referencia o valor, punteros, registros, archivos y sus operaciones básicas, etc.

CONOCIMIENTOS PREVIOS

Se recomienda disponer de conocimiento básico de Cálculo, Álgebra y Aritmética.
Se recomienda tener entrenada la mente para el razonamiento sistemático y detallado.

CONTENIDOS

Tema 1. Introducción a la programación: conceptos básicos.

- Algoritmos + Estructuras de datos = Programas.
- Lenguajes, paradigmas de programación y entornos de desarrollo. Diferencia entre programar y todo lo anterior.
- Que es y por qué aprender C. Sintaxis básica en C.

Tema 2. Fundamentos de la Programación.

- Variables. Declaración, Tipos y Alcance. Nombrando variables.
- Tipos de Datos: números, cadenas, booleanos, tipos especiales.
- Operadores y Operaciones con los tipos de datos: operaciones con números, con cadenas y con booleanos.
- Mostrando el valor de una variable: printf y los formatos.
- Concepto de Constantes y su uso.

Tema 3: Estructuras de control.

- Estructuras de selección.If... else y Case.
- Estructuras de iteración. While Loop, For Loop. Uso de Break / Continue.

Tema 4: Estructuras de datos.

- Arrays y su aritmética: búsqueda, inserción y filtrado. Iteración en un array. Tamaño del array.
- Strings - Diccionarios (hash)
- Combinación de estructuras de datos: arrays de hashes y diccionarios con arrays. Registros y Estructuras en C.
- Punteros. Memoria estática y dinámica.

Tema 5: Funciones y Programas.

- Declaración de funciones. Parámetros de una función. Ámbito: variables locales y globales.
- Llamadas a funciones y estructura de un programa. Subprogramas. Paso de parámetros
- Funciones recursivas.

Tema 6: Librerías de código.

- Concepto y creación de librerías.
- Ejemplo y uso de librerías.

Tema 7. Operaciones básicas sobre archivos.

- Tipos de archivos. - Operaciones básicas.
- Operaciones adicionales.

Tema 8. Otras características de la programación.

- Algoritmos y Pseudocódigo.
- Entornos de Desarrollo (IDEs), y Control de Versiones (GitHub)
- El futuro de la programación: Copilot.

ACTIVIDADES FORMATIVAS

La metodología seguida en esta asignatura está dirigida a conseguir un aprendizaje significativo por parte del alumno de los conceptos y técnicas fundamentales de la materia. Por ese motivo se combinan lecciones expositivas con clases prácticas, sesiones de laboratorio y presentación de trabajos, de manera que se favorezca la participación del alumno y la interacción alumno-profesor y alumno-alumno como vía para fomentar el aprendizaje colaborativo y la capacidad de auto-aprendizaje, todo ello mediante estrategias de resolución de problemas y metodologías de intervención. Las actividades no presenciales, que pueden ser tanto de tipo individual como colectivo, serán supervisadas por el profesor en clases y tutorías, tanto individuales como de grupo, estando encaminadas a favorecer el aprendizaje autónomo y colaborativo.

El trabajo presencial se completará con trabajo autónomo por parte del alumno, en algunos casos desarrollados en grupo, de manera que se fomente el aprendizaje cooperativo. Las actividades de carácter no presencial previstas incluyen el estudio individual, que permitirá trabajar en la fijación de los conceptos teóricos abordados en las clases expositivas correspondientes a todas las materias del módulo y adquirir la destreza práctica que se persigue con las clases prácticas y los laboratorios, que aplicarán el aprendizaje por descubrimiento. Para el desarrollo de las competencias y habilidades en esta asignatura son igualmente importantes los trabajos individuales y grupales. Todo el estudio y trabajo realizado por el alumno será supervisado y guiado por el profesor mediante tutorías, individuales o en grupo.

Con el fin de facilitar al alumno el acceso a los materiales y la planificación de su trabajo, así como la comunicación con el profesor y el resto de los alumnos, se empleará el Aula Virtual, plataforma de aprendizaje que ofrece diferentes recursos electrónicos para complementar, de forma muy positiva, el aprendizaje del alumno.

Se utilizará como entorno de desarrollo la URL <https://www.onlinegdb.com/> utilizando un usuario de GitHub <https://github.com/>

DISTRIBUCIÓN DE LOS TIEMPOS DE TRABAJO

ACTIVIDAD PRESENCIAL	TRABAJO AUTÓNOMO/ACTIVIDAD NO PRESENCIAL
70 horas	80 horas
Lección expositiva y ejercicios demostrativos. 30h Clase práctica y Laboratorio 30h Tutorías 4h Evaluación 6h	Estudio y trabajo individual 50h Trabajo en grupo 30h

COMPETENCIAS

Competencias básicas

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

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

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

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

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

Competencias generales

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.

Competencias específicas

Capacidad para la resolución de los problemas matemáticos que puedan plantearse en la ingeniería. Aptitud para aplicar los conocimientos sobre: álgebra lineal, cálculo diferencial e integral, métodos numéricos, algorítmica numérica, estadística y optimización.

Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y

programas informáticos con aplicación en ingeniería.

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

RESULTADOS DE APRENDIZAJE

Emplear y comparar los aspectos generales de los diferentes paradigmas y lenguajes de programación así como de cada una de las fases del proceso de desarrollo de software, especialmente las de diseño e implementación.

Analizar e interpretar el problema a resolver y elabora algoritmos utilizando las estructuras adecuadas de control y de datos.

Diseñar e implementar, en el paradigma de programación estructurada, algoritmos que resuelvan problemas sencillos utilizando tipos de datos elementales y estructuras de control básicas.

Demostrar destreza en el uso de las herramientas necesarias para el correcto desarrollo software incluyendo librerías, control de versiones, documentación, etc.

Depurar los errores detectados y analiza las posibles alternativas de resolución del problema.

Analizar y seleccionar la iteración adecuada para cada caso.

Implementar un pseudocódigo detallado, estructurado y conciso sobre un ejemplo dado.

Detectar el uso de funciones recursivas en los casos que proceda.

SISTEMA DE EVALUACIÓN DEL APRENDIZAJE

El sistema de evaluación contempla cuatro tipos de pruebas:

[1] Examen teórico: presenta un peso del 30% en la nota final. Es necesario obtener un mínimo de 5 puntos sobre 10 para poder aprobar la asignatura

[2] Examen práctico: presenta un peso del 30% en la nota final. Es necesario obtener un mínimo de 5 puntos sobre 10 para poder aprobar la asignatura

[3] Prácticas y otros trabajos relacionados con la asignatura: presenta un peso del 35% en la nota final. Es necesario tener un 5 como mínimo sobre 10.

[4] Participación en clase e implicación en la asignatura: presenta un peso del 5% en la nota final, siendo requisito imprescindible haber asistido como mínimo al 80% de las sesiones. En caso contrario este tipo de prueba se calificará con 0 puntos.

Aquellos alumnos que estén exentos de la obligación de asistir a clase, bien por segunda o sucesivas matrículas en la asignatura, bien por contar con autorización expresa de la Dirección del Grado, serán evaluados por el mismo tipo de pruebas. El 10% de la participación en clase podrán obtenerlo asistiendo al menos a tres tutorías con el profesor responsable de la asignatura.

Recuperación en convocatoria ordinaria:

Los alumnos que no hayan alcanzado la nota mínima en el examen parcial teórico y práctico podrán optar a una recuperación al final del semestre.

Recuperación en convocatoria extraordinaria:

La convocatoria extraordinaria tiene sentido para los alumnos que habiendo asistido a un número significativo de clases (más del 80%) no han conseguido superar una parte de esta. La convocatoria extraordinaria no tiene

sentido para un alumno que no haya asistido a clase ni haya participado en las entregas de prácticas de la asignatura salvo que tenga autorización expresa de la Dirección del Grado bajo alguno de los aspectos pautados para estos casos.

Los alumnos que no hayan alcanzado la nota mínima en el examen teórico y práctico y/o prácticas, habiendo suspendido por tanto en la convocatoria ordinaria, podrán optar a una recuperación en la convocatoria extraordinaria.

En ambas recuperaciones (ordinaria y extraordinaria) el alumno se presentará solo a las partes que tenga evaluadas por debajo de 5.

La nota ponderada de la evaluación continua será un valor entre 0 y 10 y se calculará como sigue: $0,30 * [1] + 0,30 * [2] + 0,35 * [3] + 0,05 * [4]$.

A efecto de cómputo de convocatorias en la asignatura, se seguirá la normativa de la universidad para estos casos.

[1a] Exámenes a mitad de cuatrimestre de carácter teórico y práctico. Cada parte de este examen puntuará sobre el 30% de la calificación final (un 60% en total) y evaluará la primera mitad del temario. El examen se puntuará de 0 a 10, repartiendo esta puntuación de manera equitativa entre todos los ejercicios y apartados, salvo que se indique lo contrario.

[1b] Exámenes de carácter teórico y práctico a final de curso que puntuara sobre el 30% de la calificación final. Se evaluará la segunda mitad del temario, si bien, debido a la relación de todos los conceptos vistos en la asignatura, se recomienda encarecidamente repasar los contenidos de la primera parte. También se puntuará de 0 a 10, repartiendo esta puntuación de manera equitativa entre todos los ejercicios y apartados, salvo que se indique lo contrario. Se evaluará el planteamiento de los problemas, así como la corrección, presentación e interpretación de los resultados obtenidos. Ambas pruebas [1a] y [1b] se realizarán sin transparencias, apuntes, libros ni cualquier otro material relacionado con la asignatura.

[2] Realización de prácticas de programación, y defensa oral o escrita, en un lenguaje estructurado y otros trabajos (35% de la calificación final). Tareas individuales o en grupo, de diversa índole, incluyendo prácticas de programación y otros ejercicios relacionados con la asignatura. En el caso que el profesor estime oportuno, la calificación quedará afectada por la defensa oral o escrita del trabajo, al alza o a la baja, para asegurar la autoría de los trabajos.

[3] Participación e implicación: 5% de la calificación final. Se evaluarán los ejercicios y otras actividades en grupo, el interés mostrado por el alumno, concretamente se computará el índice de asistencia a tutorías tanto individuales o grupales, el grado de participación activa en las clases mediante la respuesta a preguntas del profesor, el estudio de temas avanzados no vistos en clase, la recopilación y presentación en clase de noticias aparecidas en los medios de comunicación relacionadas con la asignatura, etc. La calificación de este apartado será un valor numérico entre 0 y 10. Aunque esta nota sea inferior a 5, no se podrá optar a recuperación.

Cualquier tipo de fraude o plagio por parte del alumno en una actividad evaluable, será sancionado según se recoge en la Normativa de Convivencia de la UFV. A estos efectos, se considerará "plagio" cualquier intento de defraudar el sistema de evaluación, como copia en ejercicios, exámenes, prácticas, trabajos o cualquier otro tipo de entrega, bien de otro compañero, bien de materiales o dispositivos no autorizados, con el fin de hacer creer al profesor que son propios.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

MCCONNELL, Steve. Code complete: a practical handbook of software construction / Washington :Microsoft Press,1993.

Complementaria

Carlos Pes Lenguaje C para principiantes 2021
www.abrirllave.com

Ignacio Alvarado Aldea, Jose Maria Maestre Torreblanca, Carlos Vivas Venegas, Ascensión Zafra Cabeza 100 Problemas resueltos de programación en lenguaje C para ingeniería 2017 Paraningo Universidad