

Fundamentos de
informática
**Grado en Ingeniería
Mecánica**



GUÍA DOCENTE

Asignatura: Fundamentos de Informática

Titulación: Grado en Ingeniería Mecánica

Carácter: Obligatorio

Idioma: Castellano

Modalidad: Presencial

Créditos: 6

Curso: 1º

Semestre: 1º

Profesores/Equipo Docente: D. Carmelo Martínez Morales y D. Elam Uceda Herrero

1. COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

1.1. Competencias

CGT1 Análisis y síntesis.

CGT2 Resolución de problemas.

CGT3 Comunicación oral y escrita de conocimientos en lengua propia.

CGT4 Comunicación oral y escrita de conocimientos en lengua extranjera.

CGT7 Organización y planificación.

CGS2 Aprendizaje y trabajo autónomos.

CGS3 Planificar cambios que mejoren sistemas globales.

CGP4 Trabajo en un contexto internacional.

1.2. Resultados de aprendizaje

- Que los estudiantes conozcan los fundamentos de la informática, los conocimientos relativos a la introducción a la Informática, arquitectura básica de ordenadores y una introducción a un lenguaje de programación.
- Que los estudiantes desarrollen la capacidad de pensamiento computacional y que puedan aplicarla a la solución de problemas de diversa índole que se puedan presentar en otras materias propias de los estudios de su grado. Especialmente la capacidad de análisis del problema, búsqueda de posibles soluciones, elaboración del algoritmo de la solución elegida, formalización del algoritmo en un lenguaje de programación.
- Que los estudiantes sean capaces de valorar la eficiencia de la solución informática desarrollada y en el caso de errores, sean capaces de corregir y volver a desarrollar.
- Que los estudiantes sean capaces de reunir los datos necesarios para el desarrollo de aplicaciones informáticas básicas aplicando los juicios y criterios que garanticen su buen funcionamiento.
- Que los estudiantes puedan transmitir estas soluciones informáticas, empleando con soltura los conceptos e ideas adquiridos en esta materia.
- Que los estudiantes desarrollen habilidades de aprendizaje en el campo de la Informática que les permitan aplicarlas a asignaturas posteriores, así como a proyectos en su vida profesional, con un alto grado de autonomía.

2. CONTENIDOS

2.1. Requisitos previos

Es necesario que el alumno tenga conocimientos elementales de un sistema operativo (Windows, OSx o Linux) y de Internet.

Será necesario consultar bibliografía y analizar textos en inglés, por lo que es conveniente tener conocimientos de este idioma.

2.2. Descripción de los contenidos

Dentro de cualquier estudio de ingeniería es necesario hacer frente a la solución de problemas y hacerlo de forma eficiente. Para ello es necesario seguir una serie de pasos y una metodología que permita llevarlo a cabo.

El pensamiento computacional a través de la formalización de algoritmos en un lenguaje de programación como Java permite introducir una metodología para abordar la solución de problemas.

Para ello es necesario desarrollar una forma de trabajo sistemática que pasa por el análisis del problema, la búsqueda de posibles soluciones o algoritmos que permitan resolverlo y la escritura de la solución elegida en un lenguaje de programación.

También permite desarrollar habilidades relacionadas con la detección de errores, análisis de estos y su corrección

Se parte inicialmente de una introducción de la informática básica y la arquitectura básica de un ordenador y de los lenguajes binarios que utiliza la mayoría de los sistemas digitales. Posteriormente, se realiza una introducción de la programación con diagramas de flujo y en pseudocódigo para pasar a la programación mediante el lenguaje Java.

2.3. Contenido detallado

El objetivo de la asignatura es que el alumno conozca los fundamentos de la informática, aunque la mayor parte de la asignatura consistirá en el diseño de algoritmos y su programación en Java. El aprendizaje de este lenguaje servirá para, posteriormente, extrapolar los conocimientos adquiridos a otros lenguajes de programación según vayan siendo necesarios.

- 1. Introducción a la asignatura**
- 2. Arquitectura de ordenadores**
 - Componentes de un sistema informático
 - Hardware y software
 - Sistema operativo y software de aplicación
- 3. Representación de la información**
 - Códigos para representación numérica
 - Códigos para representación de caracteres
 - Cambios de base
 - Sistemas de numeración
- 4. Algoritmos.**
 - Representación mediante diagramas de flujo
 - Representación mediante pseudocódigo
- 5. Introducción a Java. El entorno eclipse. Estructura de un programa en Java**
- 6. Datos. Tipos de datos y variables en Java.**
 - Operadores y variables

- Expresiones
 - La clase String
 - Entrada/salida básica en Java
- 7. Estructuras de control: Condicionales**
- if-then
 - if-then-else
 - switch
- 8. Estructuras de control: bucles**
- a. While
 - b. Do-while
 - c. For
- 9. Arrays y matrices**
- 10. Subprogramas y métodos**
- 11. Introducción OOP**

2.4. Actividades Dirigidas

Durante el curso se podrán desarrollar algunas de las actividades, prácticas, memorias o proyectos siguientes, u otras de objetivos o naturaleza similares:

Prácticas de la asignatura: Los alumnos, de manera individual, tendrán que realizar los ejercicios prácticos que se le vayan solicitando relacionados con las partes que se van desarrollando del temario a lo largo del cuatrimestre. Algunas de estas prácticas deberán entregarse a petición del profesor o se expondrán en clase.

Actividad Dirigida (AD1): Desarrollar una práctica que recoja los conceptos adquiridos durante el curso sobre programación, conjuntándolos para crear una pequeña aplicación para repasar y poner en práctica los conocimientos aprendidos a lo largo del curso.

- Diseñar una aplicación sencilla a partir de las especificaciones y requisitos.
- Analizar y descomponer en pasos un problema generando un algoritmo que permita su solución.
- Implementar mediante programación en Java el algoritmo generado.
- Elaborar la documentación necesaria para la adecuada utilización de la aplicación.
- Enviar por correo electrónico en un único fichero comprimido todos los archivos que formen parte de la aplicación (ficheros fuente java, documentación, manuales, ...).

2.5 Actividades Formativas

Clases de teoría y prácticas: (1,8 ECTS) Apoyándose en transparencias el profesor explica los conceptos y plantea los ejemplos y problemas. El material presentado en las transparencias de clase no constituirá la totalidad del contenido. El alumno deberá completar sus conocimientos con referencias bibliográficas, enlaces por Internet, tutoriales y ayuda del entorno integrado de desarrollo. El profesor propondrá a los alumnos el desarrollo de un programa de ordenador, iniciado y desarrollado parcialmente en las propias sesiones presenciales en aula informática y completado por el alumno en su trabajo individual.

Tutorías: (0,6 ECTS) Consulta al profesor por parte de los alumnos sobre la materia en los horarios de tutorías o empleando mecanismos de tutoría telemática (correo electrónico y uso del campus virtual de la Universidad).

Estudio individual y trabajo de asignatura: (3,6 ECTS) Estudio individual del alumno utilizando los apuntes y programas explicados en clase, libros de la biblioteca, o apuntes del profesor o módulos de software disponibles en el campus virtual. Para facilitar el aprendizaje y la realización de las prácticas de programación, el alumno puede acceder, en un horario amplio, a las aulas

informáticas donde dispondrá de los entornos de desarrollo y ordenadores con todos los programas informáticos de la asignatura. Deben entregar el programa final al profesor para ser evaluado.

3. SISTEMA DE EVALUACIÓN

3.1. Sistema de calificaciones

El sistema de calificaciones finales se expresará numéricamente del siguiente modo:

- 0 - 4,9 Suspenso (SS)
- 5,0 - 6,9 Aprobado (AP)
- 7,0 - 8,9 Notable (NT)
- 9,0 - 10 Sobresaliente (SB)

La mención de "matrícula de honor" podrá ser otorgada a alumnos que hayan obtenido una calificación igual o superior a 9,0.

3.2. Criterios de evaluación

Convocatoria ordinaria

Sistemas de evaluación	Porcentaje
Presentación de trabajos y proyectos (Prácticas individuales y trabajo en equipo)	20%
Prueba parcial (escrita/presentación trabajo)	15%
Examen final o trabajo final presencial	65%

Convocatoria extraordinaria

Sistemas de evaluación	Porcentaje
Presentación de trabajos y proyectos (Prácticas individuales y trabajo en equipo)	10%
Examen final o trabajo final presencial	90%

3.3. Restricciones

Calificación mínima

La nota final de la asignatura será la media ponderada obtenida tras la aplicación de los porcentajes anteriores. Solo se aplicará si el alumno obtiene al menos un 5 en el examen final. En caso contrario, el alumno se considerará suspenso con la nota del Examen Final. Una nota inferior a 4 en el trabajo de programación supone el suspenso automático de la asignatura en la convocatoria ordinaria. Se considera que la asignatura está aprobada si la nota final es igual o superior a 5 puntos.

Para poder hacer la suma ponderada de las calificaciones anteriores, es necesario: la asistencia a las clases como mínimo del 75% de las horas presenciales y obtener al menos un cinco en el examen final correspondiente. El alumno con nota inferior se considerará suspenso.

Es imprescindible la entrega de las prácticas, tanto para la convocatoria ordinaria, como para la extraordinaria y obtener al menos un 5.

El examen parcial no libera material y es obligatorio.

Asistencia

El alumno que, injustificadamente, deje de asistir a más de un 25% de las clases presenciales podrá verse privado del derecho a examinarse en la convocatoria ordinaria.

Normas de escritura

Se prestará especial atención en los trabajos, prácticas y proyectos escritos, así como en los exámenes tanto a la presentación como al contenido, cuidando los aspectos gramaticales y ortográficos. El no cumplimiento de los mínimos aceptables puede ocasionar que se resten puntos en dicho trabajo.

3.4. Advertencia sobre plagio

La Universidad Antonio de Nebrija no tolerará en ningún caso el plagio o copia. Se considerará plagio la reproducción de párrafos a partir de textos de auditoría distinta a la del estudiante (Internet, libros, artículos, trabajos de compañeros...), cuando no se cite la fuente original de la que provienen. El uso de las citas no puede ser indiscriminado. El plagio es un delito.

En caso de detectarse este tipo de prácticas, se considerará Falta Grave y se podrá aplicar la sanción prevista en el Reglamento del Alumno.

3.5 Uso de la inteligencia artificial (IA) generativa en las actividades formativas

La adopción de herramientas de IA en la docencia debe basarse en un enfoque transparente, responsable, ético y seguro, que fomente el desarrollo de competencias digitales en el estudiantado:

- El profesor incluirá en cada actividad formativa si tiene previsto el uso de IA Generativa, con qué objetivo y los requisitos de aplicación de esta.
- Es responsabilidad del estudiante mostrar una conducta transparente, ética y responsable con el uso de IA Generativa, y adaptarse a los criterios de aplicación dictados por el profesor en cada actividad.
- La detección de cualquier conducta fraudulenta con respecto al uso de IA Generativa, no atendiendo a las indicaciones del profesorado, aplicará las sanciones previstas en el Reglamento Disciplinario.

4. BIBLIOGRAFÍA

Bibliografía básica

H. Schildt, “Java 2. Manual de Referencia, 4^a edición” Osborne-McGraw-Hill, 2001.
J. Galve, y otros, “Algorítmica diseño y análisis de algoritmos funcionales e imperativos”. Ra-Ma.
A. Gómez Vieites y M. Veloso Espiñeira, “Redes de ordenadores e Internet”, Ra-Ma, 2003.
Prieto: “Introducción a la informática”. McGraw-Hill, 2002.
De Miguel: “Fundamentos de los computadores”. Paraninfo 1990

Bibliografía complementaria

C. Horstmann y G. Cornell, “Java 2. Vol I. Fundamentos”, Prentice Hall, 2003.
Programación en Java 2. Serie Schaum. Jesús Sánchez Allende y otros. McGraw-Hill. 2005.
Javier Moldes, “Java 9 (Manuales Imprescindibles)”. Anaya. 2017

Otros recursos

Presentaciones y documentación disponible en el campus, así como hojas de problemas.