

Fundamentos de
informática
**Grado en Ingeniería
del Automóvil**



UNIVERSIDAD
NEBRIJA

GUÍA DOCENTE

Asignatura: Fundamentos de informática

Titulación: Grado en Ingeniería del Automóvil

Carácter: Obligatorio

Idioma: Castellano

Modalidad: Presencial

Créditos: 6

Curso: 1º

Semestre: 1º

Profesores/Equipo docente: D. Carlos López Lázaro, Dña. Ibtissam Azaoum Zahoum

1. COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

1.1. Competencias

Competencias específicas

- CE3. Conocimientos básicos sobre el uso y la programación de ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicaciones en ingeniería

Competencias instrumentales

- CGI1. Capacidad de análisis y síntesis
- CGI2. Capacidad de organizar y planificar
- CGI3. Conocimientos generales básicos
- CGI5. Comunicación oral y escrita en la lengua nativa
- CGI7. Conocimientos de informática
- CGI8. Capacidad de gestión de la información
- CGI9. Resolución de problemas
- CGI10. Capacidad para la toma de decisiones

Competencias sistémicas

- CGS1. Capacidad de aplicar los conocimientos en la práctica
- CGS2. Capacidad de aprender
- CGS7. Habilidad para trabajar de forma autónoma
- CGS10. Preocupación por la calidad
- CGS11. Motivación de logro

1.2. Resultados de aprendizaje

Los efectos que cabe asociar a la realización por parte de los estudiantes de las actividades formativas anteriormente indicadas son los conocimientos de la materia, la aplicación con criterio los métodos de análisis y técnicas descritos en ella, redactar utilizando un lenguaje preciso y adecuado a la misma, y aprender por sí mismo otros conocimientos relacionados con la materia, que se demuestran:

- En la realización de los exámenes: parcial, final y extraordinario en su caso.
- En sus intervenciones orales en clase.
- En la memoria y detalle del trabajo de programación propuesto que el estudiante entrega, utilizando los programas informáticos de la asignatura.

2. CONTENIDOS

2.1. Requisitos previos

Ninguno.

2.2. Descripción de los contenidos

- Arquitectura de ordenadores.
- Sistemas operativos.
- Redes de ordenadores e Internet.
- Bases de datos.
- HTML y páginas Web.
- Introducción a Java. Manejo del entorno de desarrollo.
- Entradas y salidas básicas: pantalla y teclado.
- Tipos de datos y operadores.
- Algorítmica y sentencias de control.
- Clases y métodos.
- Excepciones y paquetes.
- Herencia y agregación.

2.3. Contenido detallado

Presentación de la asignatura

Explicación de la **guía docente**

1. Fundamentos de informática

- Arquitectura de ordenadores
- Sistemas operativos
- Redes de ordenadores e Internet
- Bases de datos
- HTML y páginas Web
- Entradas y salidas básicas: pantalla y teclado

2. Entorno de desarrollo

- Presentación
- Descarga e instalación del intérprete y del IDE
- Elementos básicos de un lenguaje de programación. Scripts y programas

3. Algorítmica y sentencias de control

- Algoritmia básica

4. Elementos de un lenguaje de programación

- Guía de estilo
- Comentarios
- Tipos de datos simples. Variables
- Operaciones aritméticas
- Creación, manipulación y operaciones con strings
- Salida de strings por consola. Salida formateada

5. Control de flujo. Instrucciones condicionales

- Instrucciones condicionales
- Operadores relacionales y operadores lógicos
- Variables booleanas

6. Bucles

- Bucle While. Condición del bucle. Bucles infinitos

- Bucle For. Rango de números
- Instrucciones break y continue

7. Listas

- Trabajar con listas. Acceso, modificación, adicción y eliminación de elementos de una lista. Iteración por los elementos de una lista
- Listas de listas (listas bidimensionales)
- Métodos principales para el manejo de listas

8. Diccionarios y JSON

- Diccionarios. Sintaxis: elementos y claves
- Acceso, adición, modificación y supresión de elementos de un diccionario. Iteración por diccionarios

9. Funciones

- Definición de funciones. Sintaxis. Llamadas a funciones. Parámetros y argumentos
- Funciones sin retorno y funciones que devuelven valores
- Programación funcional. Funciones anónimas: Lambda

10. Entrada y salida con ficheros.

- Abrir ficheros. Modos de apertura del fichero
- Operaciones con ficheros: escribir y leer. Cerrar ficheros.

11. Desarrollo de software estructurado

- Paquetes. Importación de módulos. Importación genérica, importación de funciones individuales e importación universal
- Introducción a la OOP
- Errores de programación. Tipos de errores. Estrategias para depuración de errores
- Control de errores con excepciones
- Módulos y librerías de uso interesante

12. JAVA: Introducción a Java

2.4. Actividades dirigidas

Los estudiantes, de manera individual, tendrán que realizar los ejercicios prácticos que se les solicite a lo largo del cuatrimestre relacionados con el temario de la asignatura. Algunas de estas prácticas deberán entregarse a petición del profesor o se expondrán en clase.

Durante el curso se desarrollará la siguiente actividad dirigida:

- Actividad dirigida (AD1). Desarrollar un trabajo de programación 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. Incluirá:
 - 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...).

El estudiante debe entregar el trabajo de programación al profesor de la asignatura.

2.5. Actividades formativas

Clases de teoría y prácticas: (1,8 ECTS, 45h, 100%presencialidad). 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, 15h, 100% presencialidad). 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, 90h, 0% presencialidad). 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
Asistencia, participación y trabajo de programación	25%
Examen parcial	15%
Examen final	60%

Convocatoria extraordinaria

Sistemas de evaluación	Porcentaje
Asistencia, participación y trabajo de programación	10%
Examen final	90%

3.3. Restricciones

Calificación mínima

Calificación mínima

En la convocatoria ordinaria, la nota final de la asignatura será la media ponderada obtenida tras la aplicación de los porcentajes anteriores. Esta ponderación solo se aplicará si el estudiante obtiene al menos 4,5 puntos en el examen final ordinario. Una nota inferior a 4,0 puntos en el trabajo de programación supone el suspenso automático de la asignatura en la convocatoria ordinaria. Además, 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 5,0 puntos en el examen final correspondiente.

En la convocatoria extraordinaria, la nota final de la asignatura será la media ponderada obtenida tras la aplicación de los porcentajes anteriores. Esta ponderación solo se aplicará si el estudiante obtiene al menos 4,5 puntos en el examen final extraordinario.

Es imprescindible la entrega de las prácticas, tanto para la convocatoria ordinaria, como para la extraordinaria y obtener al menos 4,5 puntos.

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, 4ª edición” Osborne-McGraw-Hill, 2001.
- J. Galve, y otros, “Algoritmica 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
- P. Barry: Head First Python. O'Reilly 2013

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