UNIVERSIDAD NACIONAL DE INGENIERÍA

SÍLABO

INFORMACIÓN GENERAL

ASIGNATURA	:INTRODUCCIÓN A LA COMPUTACIÓN
DOCENTE	:GERSON GARRIDO
HORAS POR SEMANA	:04 (TEORÍA: 02, LABORATORIO: 02)

SUMILLA

El curso es de naturaleza teórico-práctica y es la primera mirada a la computación: qué es un algoritmo, cómo se convierte en un programa y qué hace la computadora para ejecutarlo. Se organiza en cuatro partes: **fundamentos** (algoritmos, pseudocódigo, variables, representación de datos, terminal y Git); **programación en C** (decisiones, ciclos, números aleatorios, funciones y pruebas); **memoria y datos** (funcionamiento de la computadora, arreglos, cadenas, matrices, estructuras, archivos, búsqueda y eficiencia); y **computación moderna** (una mirada a Python, internet, inteligencia artificial y ética). El laboratorio se desarrolla en C, en un entorno común.

COMPETENCIAS

- Describe un problema como un algoritmo, en pseudocódigo y con diagramas de flujo, y lo sigue a mano.
- Explica cómo se representan números y texto en bits, y prevé errores como el overflow y la imprecisión de los decimales.
- Escribe en C programas con variables, decisiones, ciclos, funciones, arreglos, cadenas, matrices, estructuras y archivos.
- Prueba y depura sus programas, y usa la terminal y lo básico de Git.
- Explica a grandes rasgos cómo funcionan el hardware, el sistema operativo e internet.
- Compara algoritmos de búsqueda y ordenamiento según su eficiencia, con mediciones reales.
- Explica qué hace un modelo de machine learning y reconoce los límites y efectos éticos de la inteligencia artificial.

UNIDADES DE APRENDIZAJE

Unidad	Contenido	Semanas
I	Fundamentos	1–3
II	Programación en C	4–7
III	Memoria y datos	8–13
IV	Computación moderna	14–16

PROGRAMACIÓN SEMANAL

Cada semana de clases comprende una sesión de teoría de 2 horas y una sesión de laboratorio de 2 horas. El capítulo de los apuntes de cada semana es lectura previa. Los temas marcados con (*) pueden pasar a lectura autónoma si se pierden clases por feriados.

Sem.	Teoría	Laboratorio
1	Presentación. Algoritmos y pseudocódigo. Del programa al ejecutable	Entorno, terminal y primer programa
2	Variables, tipos, operadores, entrada y salida	Fórmulas en C
3	Bits: binario, hexadecimal, enteros, decimales y texto	Bits a mano y primeros pasos con Git
4	Decisiones y lógica booleana. Compuertas (*)	Decisiones. *Práctica 1*
5	Ciclos: `while`, `do while` y `for`	Ciclos
6	Ciclos anidados. Números aleatorios	Ciclos anidados y una simulación
7	Funciones. Pruebas y depuración	Funciones con pruebas y el debugger. *Práctica 2*
8	**Examen parcial**	
9	Dentro de la computadora. Direcciones y arreglos	Arreglos
10	Arreglos y funciones. Cadenas	Cadenas y errores de memoria
11	Biblioteca de cadenas. Matrices	Cadenas y matrices. *Práctica 3*
12	`struct` y archivos de texto	Procesar un archivo de datos
13	Búsqueda, ordenamiento y eficiencia. Recursión (*)	Medir búsquedas y ordenamientos
14	Internet y la web. Una mirada a Python	Ver HTTP por dentro y comparar C con Python. *Práctica 4*
15	Inteligencia artificial y ética	Actividad guiada con un asistente. Repaso
16	**Examen final**	

METODOLOGÍA

- **Teoría:** clases expositivas con ejercicios resueltos en clase y programas seguidos paso a paso. El estudiante lee el capítulo de los apuntes antes de cada semana.
- **Laboratorio:** prácticas guiadas en un entorno común para todos los estudiantes, donde escriben, prueban y depuran sus programas.
- **Prácticas:** cuatro prácticas individuales, en las semanas 4, 7, 11 y 14, con pruebas automáticas. Esa semana, la sesión de laboratorio es para la práctica y las actividades de la guía quedan como trabajo autónomo.
- Los términos técnicos se presentan en español cuando así se dicen (variable, arreglo, función) y en inglés cuando la traducción suena forzada (bit, commit, router).

NORMAS DEL CURSO

- **Trabajo:** individual. Copiar soluciones de otras personas o de internet cuenta como plagio.

HERRAMIENTAS PERMITIDAS

Las herramientas de IA generativa están prohibidas en el curso, salvo en la actividad guiada de la semana 15. El resto depende de la herramienta:

Herramienta	¿Se permite?
Mensajes del compilador y del intérprete, `-Wall -Wextra` y los sanitizers (`-fsanitize`)	Sí, siempre. Son parte de lo que se enseña
Documentación oficial (páginas `man`, cppreference, docs.python.org) y búsquedas en la web	Sí, para consultar cómo funciona algo. No para copiar soluciones
Autocompletado del editor que completa nombres o una línea, sin generar soluciones enteras	Sí
Correctores de ortografía, traductores y herramientas de accesibilidad (lectores de pantalla, dictado)	Sí
Formateadores de código y linters, como el formateador del editor	Sí
IA generativa para pedir explicaciones, sin que genere código	No, salvo en la actividad guiada de la semana 15
IA generativa que escribe código, incluidos los asistentes del editor	No, salvo en la actividad guiada de la semana 15, donde se revisa el código que propone

Cómo se declara el uso. En la actividad de la semana 15, el estudiante anota en una hoja lo que pidió al asistente, qué hizo bien y qué error encontró. Fuera de esa actividad, si una idea o un fragmento del código sale de una página de documentación o de una búsqueda, se anota en un comentario junto a ese código, con el enlace. El compilador, el editor y los correctores no se declaran. En cualquier laboratorio, práctica o examen, el docente puede pedir que el estudiante explique su trabajo de forma oral.

BIBLIOGRAFÍA

- Apuntes del curso, un capítulo por semana, en el sitio del curso. No se requiere comprar ningún libro.
- Sitio del curso: guías semanales, apuntes, diapositivas, código y enunciados.