PROGRAMACIÓN 2 – Programación Estructurada

Fundamentos de la programación estructurada para construir sistemas robustos y eficientes.

Puntos de la unidad

- Introducción a la Programación Estructurada
- Operadores y Estructuras Condicionales
- Estructuras de Repetición
- Modularidad con Funciones
- Recursividad y Estructuras de Datos Básicas
- Prácticas y Aplicaciones Reales

Objetivos de Aprendizaje

Al finalizar este curso, los estudiantes serán capaces de:

1

Control de Flujo

Utilizar operadores y estructuras condicionales para la toma de decisiones.

2

Automatización de Tareas

Implementar estructuras de repetición para el procesamiento eficiente de datos.

3

Modularidad del Código

Desarrollar programas modulares mediante funciones reutilizables.

4

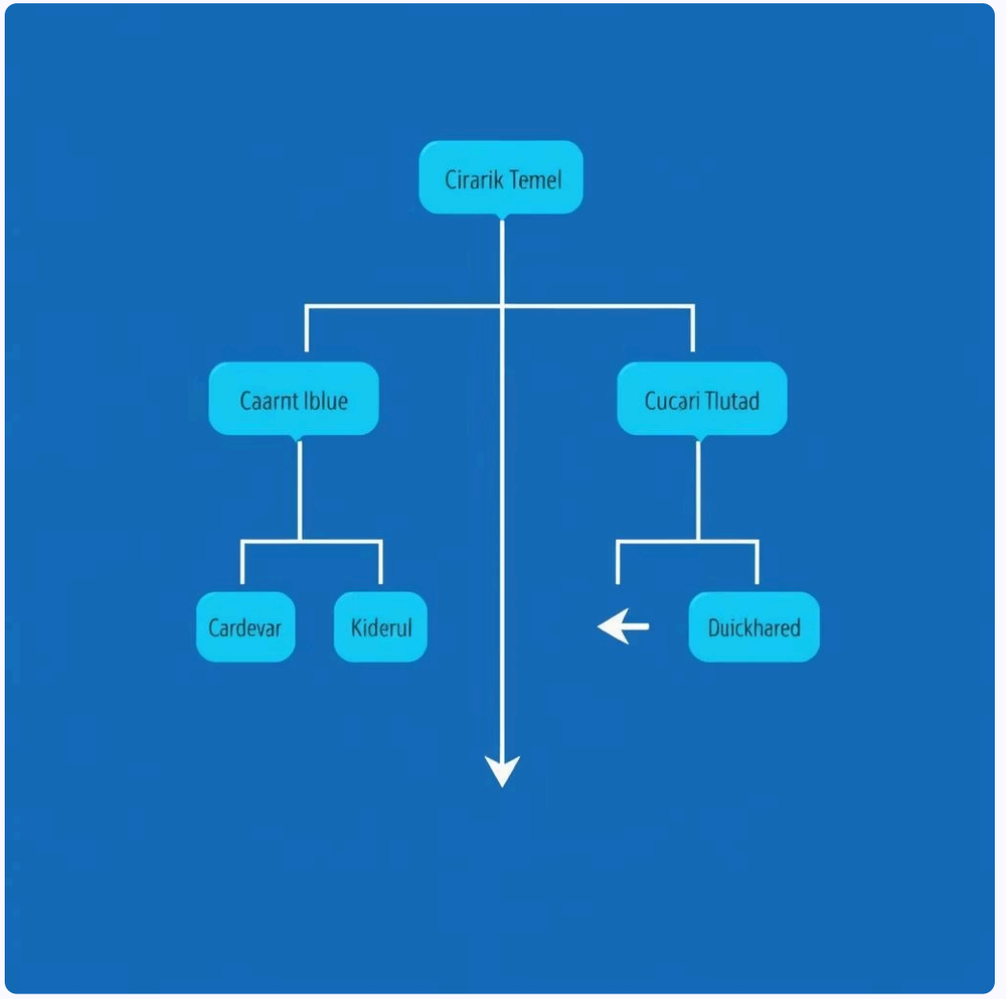
Resolución de Problemas Complejos

Aplicar recursividad y estructuras de datos básicas para la resolución de problemas.

Operadores y Estructuras Condicionales

Conceptos Clave

Dominar el uso de operadores aritméticos, lógicos y relacionales es fundamental para construir la lógica de control. Las estructuras condicionales (`if`, `else if`, `else`, `switch`) permiten al programa tomar decisiones en base a diferentes condiciones.



> Una buena toma de decisiones en el código se traduce en programas más robustos y adaptables a distintos escenarios.

Estructuras de Repetición

Bucles for

Ideales para iterar un número conocido de veces, optimizando tareas repetitivas.

Bucles while

Permiten la ejecución de bloques de código mientras una condición sea verdadera.

Bucles do-while

Aseguran que el bloque de código se ejecute al menos una vez antes de verificar la condición.

La automatización de tareas repetitivas es un pilar de la eficiencia en programación, esencial para el procesamiento de grandes volúmenes de datos.

Desarrollo Modular con Funciones

- **Reusabilidad:** Las funciones permiten reutilizar bloques de código, evitando la duplicación.
- **Organización:** Mejoran la estructura del programa, haciéndolo más legible y fácil de entender.
- **Mantenimiento:** Facilitan la depuración y actualización del código al aislar funcionalidades.
- **Colaboración:** Promueven el trabajo en equipo al permitir que diferentes desarrolladores trabajen en módulos específicos.

Un código bien estructurado con funciones es la base para proyectos de software escalables y de fácil mantenimiento.

Recursividad y Estructuras de Datos Básicas

Recursividad

Diseñar algoritmos eficientes donde una función se llama a sí misma para resolver problemas complejos.

Listas

Manejar colecciones dinámicas de elementos para organizar y acceder a la información.

Pilas (Stacks)

Implementar lógica LIFO (Last-In, First-Out) para diversas aplicaciones.

Colas (Queues)

Aplicar lógica FIFO (First-In, First-Out) para procesar elementos en orden.

Comprender el impacto de estas herramientas en el rendimiento es crucial para la optimización de algoritmos.

Próximos Pasos

¡Estén atentos a las próximas fechas de entrega de trabajos prácticos y al foro de consultas!

Recuerden que la práctica constante es clave para dominar los conceptos de Programación Estructurada.

¡No duden en preguntar y explorar!