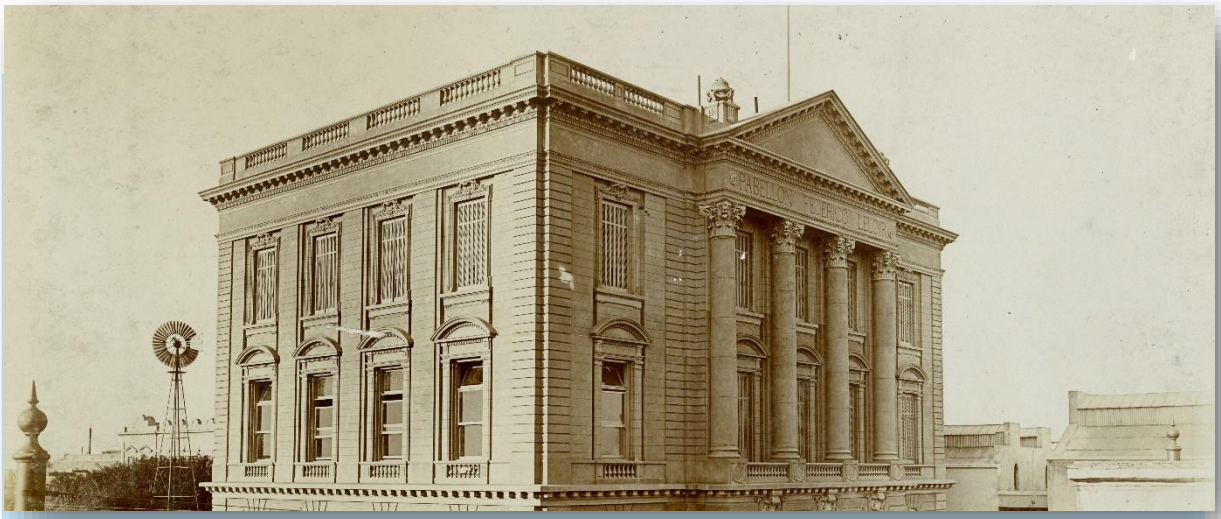
UTN

INGRESO 2025

TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN A DISTANCIA

UNIVERSIDAD TECNOLÓGICA NACIONAL



CUADERNILLO 11

Integrando técnicas

CURSO COMPLETO

UNIDAD I FUNDAMENTOS LOGICOMATEMÁTICOS

CUADERNILLO 1 – Teoría de conjuntos, números y sus tipos

CUADERNILLO 2 – Sistema Binario

CUADERNILLO 3 – Introducción a la lógica

CUADERNILLO 4 – Operaciones aritméticas

CUADERNILLO 5 – Números Enteros

CUADERNILLO 7 – Más de números

UNIDAD II RESOLUCIÓN DE PROBLEMAS

CUADERNILLO 6 – Análisis verbal

CUADERNILLO 8 – Método iterativo

CUADERNILLO 9 – Analogía y Patrones

CUADERNILLO 10 – Divide y conquistarás

CUADERNILLO 11 – Integración

CUADERNILLO 12 – Ensayo y Error

11: INTEGRACION

REPASO DE LAS TECNICAS

INICIO

Cuando encaramos la resolución de un problema para resolver con un programa, seguimos unos pasos lógicos:

ENUNCIADO

ANALISIS: Con análisis verbal de enunciado detectamos las posibles entradas o inputs (datos), salidas o outputs (que nos piden) y restricciones de nuestro software.

PSEUDOCÓDIGO: Hacemos una lista de pasos del proceso que relaciona los inputs con los outputs, en un lenguaje que llamamos pseudocódigo. Esta primera versión es muy sencilla y en general en lenguaje coloquial.

REFINAMIENTO: En forma iterativa vamos refinando el primer esbozo hasta llegar a un pseudocódigo refinado con los pasos lógicos y sistemáticos de nuestra solución. Para ellos usamos todas las heurísticas que conocemos.

CODIFICACIÓN: Con el pseudocódigo refinado vamos "traduciendo" cada línea a una sentencia en el lenguaje que vamos a usar. O sea, codeamos.

SOFTWARE

INTRODUCCION

Para realizar los pasos que nos lleva a un programa utilizamos en forma intensiva las técnicas que te enseñamos:

1. **Análisis verbal de enunciados**: Desmenuzar el problema en palabras clave y significados.
2. **Método iterativo**: Atacar el problema en sucesivas aproximaciones, afinando y corrigiendo en cada vuelta.
3. **Analogía y patrones**: Buscar semejanzas con problemas resueltos previamente y reconocer patrones para aplicarlos en nuevos contextos. Nos acelera el proceso de refinamiento si ya trabajamos en el mismo tipo de problema.
4. **Divide y conquistarás**: fraccionar el problema en partes manejables que se puedan abordar de forma organizada.

El propósito de esta lección es ofrecerte tips que te ayuden a leer, analizar y resolver enunciados de manera clara y efectiva.

ANALISIS VERBAL DE ENUNCIADOS

Antes de empezar a diseñar cualquier solución, hay que entender el problema al detalle. Estos son algunos pasos y consejos que te pueden ayudar:

1. Leer detenidamente el enunciado

Haz una primera lectura para comprender la idea general.

Subraya o anota términos clave y datos importantes (fechas, cantidades, relaciones lógicas, etc.).

Identifica si el problema te pide un resultado numérico, un texto, un procedimiento específico o alguna combinación.

2. Reformular el enunciado

Después de la primera lectura, explica con tus propias palabras qué se te está pidiendo.

Si hay partes confusas, intenta reescribirlas o preguntar.

Asegúrate de distinguir lo que es dato de entrada (input) de lo que es resultado esperado (output).

3. Identificar restricciones o condiciones

¿Hay límites claros para las variables o datos?

¿Existen condiciones que deban cumplirse para considerar la respuesta como válida?

¿Se establecen plazos de tiempo, cantidad máxima/mínima de elementos, formatos específicos, etc.?

4. Ejemplo concreto o caso de uso

Cuando sea posible, imagina un ejemplo sencillo para verificar tu comprensión del enunciado.

Revisar un caso con datos concretos, incluso inventados, puede ayudarte a ver la estructura del problema.

Tips:

Practica el “análisis verbal” en enunciados simples para adquirir soltura.

Hacete preguntas del estilo: “¿Qué se sabe?”, “¿Qué me piden?”, “¿En qué condiciones?”, “¿Qué debo calcular o programar exactamente?”.

METODO ITERATIVO

El método iterativo sugiere que abordemos el problema en varios ciclos, refinando en cada paso:

1. Borrador de la solución

A partir de tu comprensión inicial, crea un primer intento de solución.

No te preocupes todavía por detalles como la optimización del código o la elegancia de la lógica.

Crea un esquema general, puede ser un pseudocódigo.

2. Probar y ajustar

Aplica tu borrador a un ejemplo básico preferentemente, uno que tú mismo hayas construido durante el análisis del enunciado.

Observa si los resultados son los esperados.

Si aparecen errores o comportamientos inesperados, corrígelos en esta etapa.

Repite la prueba cuantas veces sea necesario.

3. Refinar la solución

Una vez que el esqueleto básico funciona, añade detalles o mejoras como manejo de excepciones, validaciones, mejoras en la eficiencia, etc.

Documenta los cambios o justifica por qué tu solución final es la adecuada.

Tips:

Usar el método iterativo te ayuda a no abrumarte con la complejidad inicial del problema. Trabajar gradualmente, paso a paso, previene errores y acelera el proceso de desarrollo.

ANALOGIA Y PATRONES

Los programadores suelen encontrarse con problemas similares una y otra vez en diferentes contextos. Reconocer patrones y aplicar analogías te ayudará a desarrollar soluciones de manera más rápida y efectiva.

1. Buscar problemas similares

Piensa en otros desafíos resueltos previamente.

Pregúntate: “¿Este problema se parece a uno que ya resolví?”, “¿Hay alguna estructura o patrón que se repite?”.

2. Aplicar la analogía

Si existen similitudes, aprovecha la lógica y la estructura de la solución anterior.

Ajusta los detalles como nombres de variables, tipos de datos, límites, de acuerdo con el nuevo problema.

3. Identificar patrones comunes

Algunos patrones frecuentes en programación son: búsqueda lineal o binaria, ordenamientos, recursividad, manejo de estructuras de datos como pilas, colas, listas, árboles, etc.

Dominar estos patrones es clave: te ahorran tiempo y te dan un marco de referencia.

Tips:

Lleva un “repositorio mental” o real como notas, repositorios de código en GitHub, de soluciones previas.

Revisitar tus códigos y ejemplos es una excelente fuente de ideas para enfrentar nuevos retos.

DIVIDE Y CONQUISTARAS

Muchos problemas complejos se hacen más simples si se dividen en partes más pequeñas que se resuelven por separado. Posteriormente, se integran todas las piezas.

1. Descomponer el problema

Haz un listado de sub-problemas o tareas.

Ordena esas tareas en función de sus dependencias, primero resuelve las partes que son indispensables para seguir avanzando.

2. Resolver cada parte de forma independiente

Toma una tarea y resuélvela o codifícala sin preocuparte aún por el resto.

Haz pruebas sobre esa pequeña parte para asegurar que funciona por sí sola.

3. Integrar y verificar

Una vez que cada parte funciona, ve ensamblando todas las piezas.

Revisa la comunicación entre los módulos o funciones (pasaje de parámetros, resultados intermedios, etc.).

Realiza pruebas de conjunto para encontrar errores de integración.

Tips:

Cada sub-problema debería ser lo suficientemente pequeño para poder entenderlo y resolverlo en un intervalo de tiempo razonable. Si sigue siendo complejo, ¡subdivide aún más!

CHECKLIST

A continuación, se propone un checklist que integra las cuatro estrategias:

1. Leer y re-leer el enunciado (Análisis verbal)

Identificar variables de entrada, resultados esperados, restricciones.

Subrayar palabras clave.

Reformular el problema con tus propias palabras.

2. Dividir en partes (Divide y conquistarás)

Esbozar un listado de sub-problemas o módulos.

Priorizar el orden en que se abordarán.

3. Relacionar con problemas conocidos (Analogía y patrones)

Preguntarte si hay un enfoque, un algoritmo o un patrón que ya hayas visto que aplique.

Revisar ejemplos o soluciones similares.

4. Desarrollar un primer borrador (Método iterativo)

Crear pseudocódigo o un diagrama de flujo que abarque el problema completo.

Ignora la optimización de momento.

5. Probar el borrador en un caso sencillo (Método iterativo)

Usar datos de prueba simples para verificar la lógica.

Anotar qué funciona y qué no.

6. Refinar y optimizar (Método iterativo)

Corregir errores hallados en la fase de prueba.

Ajustar detalles de implementación (estructura de datos, validaciones, etc.).

Añadir comentarios o documentación.

7. Prueba con más casos (Análisis verbal / Método iterativo)

Introducir nuevos ejemplos, incluso casos límite o poco comunes.

Asegurate de que la solución se mantenga estable y robusta.

8. Conclusión y comunicación

Elaborar una explicación final de la solución, describiendo cada paso o módulo del programa.

Si fuera un trabajo práctico o examen, detallar la parte conceptual y la implementación.

CONCLUSION

La resolución de problemas en programación no es un proceso lineal ni único; suele involucrar prueba y error, búsqueda de patrones, descomposición de tareas y, sobre todo, comprensión profunda del enunciado.

Usar estas cuatro herramientas te ayudará a entender los enunciados y generar soluciones más sólidas y eficientes.

¡Ánimo con la práctica! Recuerda que la mejor forma de interiorizar estas estrategias es resolviendo muchos problemas, aunque parezcan sencillos. De cada uno se aprende algo nuevo y relevante para tu formación como programador.

APLICACIÓN INTEGRAL

ENUNCIADO 1

"Determinar si un número n es par o impar."

Análisis verbal del enunciado

Datos (inputs): Un número.

Restricciones: Ninguna.

Qué se pide (outputs): Si es par o impar

Idea de solución

Sabemos que, si un número es divisible por 2, o sea el resto es 0, es par. Si no es par.

Pseudocódigo

1. Tomamos el número.
2. Dividimos por 2.
3. Si el residuo es 0, informamos que el número es par.
4. Si no, informamos que es impar.

ENUNCIADO 2

"Calcular el promedio de un conjunto de n números."

Análisis verbal del enunciado

Datos (inputs): Un conjunto de números.

Restricciones: al menos un número.

Qué se pide (outputs): Promedio

Idea de solución

Sumar uno a uno todos los números y dividir el total por la cantidad de números para obtener el promedio. (usaremos método iterativo)

Pseudocódigo

1. Tomamos un conjunto de números.
2. Inicializamos suma en 0.
3. Recorremos el conjunto de números acumulándolos (acumulador).
4. Contamos los números que vamos sumando (contador).
5. Divide acumulador por contador para obtener promedio.
6. Informamos el promedio

ENUNCIADO 3

"Resolver la ecuación lineal: a ∗ x + b = 0."

Análisis verbal del enunciado

Datos (inputs): Coeficientes a y n.

Restricciones: a no puede ser 0.

Qué se pide (outputs): El valor de x.

Idea de solución

Dividir el problema en tres: Validar que a sea distinto de 0. Despejar x de la ecuación lineal. (Divide y conquistarás).

Pseudocódigo

1. Validar a.
    1. Si no es igual a 0 continuar
    2. Si es, informar que las soluciones son infinitas y salir
2. Usar la fórmula x = −b/a.

ENUNCIADO 4

"Convertir una temperatura de grados Celsius a Fahrenheit."

Análisis verbal del enunciado

Datos (inputs): Un valor en grados Celsius.

Restricciones: Ninguna.

Qué se pide (outputs): El equivalente en grados Fahrenheit.

Idea de solución

Relacionado con otros problemas de conversión de unidades entre sistemas de medidas diferentes, hay que buscar la fórmula. (Analogía)

Pseudocódigo

1. Tomamos el valor en Celsius.
2. multiplica ese valor 9 y divide por 5, luego suma 32.
3. Muestra la temperatura en Fahrenheit.

ENUNCIADO 5

"Calcular el perímetro de un rectángulo dado su base b y altura h"

Análisis verbal del enunciado

Datos (inputs): Toma base b y altura h.

Restricciones: Base b y altura h deben ser mayores a 0.

Qué se pide (outputs): perímetro.

Idea de solución

Multiplicamos base y altura por 2 y luego sumas ambos resultados (Analogía con otros problemas de geometría).

Pseudocódigo

1. Tomamos base b y altura h.
2. Sumamos base b más altura h y luego multiplicamos por 2.
3. Informamos el perímetro.