

TP6 INGA GONZALO

Para cada enunciado, indica:

1. Los datos, restricciones y qué se pide.
2. El tipo de dato primitivo necesario.
3. Una frase sobre qué harías para resolverlo.

1. Identifica si el problema requiere iteración conocida (for) o indefinida (while).
2. Declara y define las variables necesarias, como contadores o acumuladores.
3. Diseña una condición clara para detener el ciclo.
4. Si el enunciado parece ambiguo, divídelo en partes para comprender mejor sus requisitos.
5. Prueba con valores simples para asegurarte de que la lógica del ciclo funciona como se espera.

1. Se repite un numero fijo, un for
2. necesito solo un acumulador
3. Llego a la quinta nota, implicito
4. no es necesario
5. Testeo, no es necesario en este caso ya que es una suma simple

- Datos: 5 Estudiantes con calificación
- Restricciones: Si tiene, las notas no pueden ser negativas, ni superar un maximo (agregado)
- Qué se pide: promedio de notas
- Condicion de entrada: Acumulador en 0
- Ciclo: 5 veces pedir nota y acumularla
- Condicion de salida: Llegue a la quinta
- ¿Cómo se le explico a otra persona?
Pedi las 5 notas, las sumas y luego dividis por 5

6.1) Calcular la suma de los primeros 100 números naturales.

n: 100

$$[n * (n+1)] / 2$$

$$[100 * (100+1)] / 2 = 5050$$

6.2) Encontrar el promedio de las calificaciones de 5 estudiantes.

- **Datos:** 5 estudiantes
- **Restricciones:** Las notas no pueden ser negativas, ni superar un máximo establecido, por ejemplo 10/10, No puede haber más de una calificación por alumno.
- **Que me piden:** Un promedio de las notas.
- **Condición de entrada:** un acumulador inicia en 0 (número del primer estudiante)
- **Ciclo:** Pedir ingresar la nota 5 veces, mientras aumenta el contador.
- **Condicion de salida:** Cuando llegue el acumulador a 5

1. Voy a necesitar un ciclo for para iterar por sobre los 5 estudiantes.
2. Necesito un acumulador.
3. El ciclo se detiene cuando llego al número máximo de estudiantes.
4. No necesito dividir el enunciado.
5. No necesito hacer muchas pruebas

- **¿Como le explico a otra persona ?:**
Ingreso las 5 notas, las sumo y las divido por el número de estudiantes

`notasEstudiantesSuma / numeroDeEstudiantes`

6.3) Determinar el número más grande en un conjunto de 10 números enteros.

- **Datos:** 10 números enteros
- **Restricciones:** ¿Se incluirá números negativos?, por el momento usare números enteros positivos.
- **Que me piden:** Determinar el número entero más alto.
- **Condición de entrada:** un acumulador inicia en 0 (número del primer número)
- **Ciclo:** Pedir ingresar un numero 10 veces, mientras aumenta el contador.
- **Condicion de salida:** Cuando llegue el acumulador a 5

1. Voy a necesitar un ciclo for para iterar por sobre los 10 números ingresados.
2. Necesito un acumulador.
3. El ciclo se detiene cuando llego al número máximo de números.
4. Puedo dividir el enunciado en ingreso de datos, y luego comparación de datos, donde voy a comparar el numero con su anterior dato e ir actualizando el número más alto.
5. No necesito hacer muchas pruebas

- **¿Como le explico a otra persona ?:**
Ingreso los 10 números, los comparo entre si y voy determinando cual es el numero mas alto

Ejemplo muy básico de la idea.

```
numeroMasAlto = 99
```

```
numeroMasAlto < numero2 {  
  //si la condición es verdadera  
  numeroMasAlto = numero2  
}  
...  
numeroMasAlto < numero10{  
  //si la condición es verdadera  
  numeroMasAlto = numero10  
}
```

Y así sucesivamente con todos los numero hasta determinar cual es el numero mas alto, en este caso supongamos que el 99 es el mas grande, es el que nos queda como numeroMasAlto.

6.4) Contar cuántos números pares hay entre 1 y 50.

- **Datos:** del 1 a 50
- **Restricciones:** No hay números negativos, números enteros, no se cuenta el cero.
- **Que me piden:** Determinar cuántos números pares hay.
- **Condición de entrada:** un contador inicia en 0 (cantidad de números pares) y un acumulador inicializado en 1 (número del primer número)
- **Ciclo:** Inicia el ciclo, y hasta no terminar con el ciclo, se aumenta el acumulador y vamos contando los números pares y aumentando nuestro contador.
- **Condicion de salida:** Cuando llegue el acumulador a 50

1. Voy a necesitar un ciclo for para iterar desde el 1 hasta el 50.
2. Necesito un acumulador y un contador.
3. El ciclo se detiene cuando llego al 50.
4. Divido el enunciado en la iteración de los números y su comparación para ver si es par o no.
5. No necesito hacer muchas pruebas

- **¿Como le explico a otra persona ?:**

La ejecución de la resolución se dividirá en etapas:

1. Iteración sobre los números
2. Se dividirá ese numero por 2 y se verá si tiene resto
3. Si tiene resto cero se actualizará el contador aumentando en uno
4. Sino se seguirá con el siguiente numero

6.5) Hallar la cantidad de letras en una palabra dada.

- **Datos:** Una palabra.
 - **Restricciones:** La palabra no debe contener espacios si se considera una sola palabra. Se ignoran caracteres especiales o números, se busca contar solo letras.
 - **Qué me piden:** Contar la cantidad de letras en la palabra.
 - **Condición de entrada:** Se recibe una palabra como cadena de texto.
 - **Ciclo:** Recorrer cada carácter de la palabra y contar solo aquellos que sean letras.
 - **Condición de salida:** Terminar cuando se haya revisado toda la palabra.
1. Voy a necesitar un ciclo for, para iterar sobre la palabra
 2. Una variable para almacenar la palabra. Un contador para sumar las letras encontradas.
 3. Se detiene cuando se recorre toda la palabra.
 4. No es necesario, el problema es claro.
 5. Probar con palabras de diferentes longitudes, palabras con números o símbolos para verificar si los excluye correctamente.

- **¿Como le explico a otra persona ?:**

Tomo la palabra ingresada y recorro cada uno de sus caracteres. Si el carácter es una letra, lo cuento. Al final, muestro el total de letras.

6.6) Calcular el total de una factura con un descuento del 10% si supera los \$500.

- **Datos:** Monto total de la factura.
 - **Restricciones:** No puede haber montos negativos, Si el total de la factura es mayor a \$500, se aplica un descuento del 10%, Si es \$500 o menos, no hay descuento.
 - **Qué me piden:** Calcular el total final de la factura considerando el descuento si corresponde.
 - **Condición de entrada:** Se recibe un número positivo que representa el total de la factura.
 - **Ciclo:** No es necesario un ciclo, ya que solo se realiza una evaluación condicional.
 - **Condición de salida:** Mostrar el total final después de aplicar el descuento (si es necesario).
1. No se necesita un ciclo, solo una condición if.
 2. Una variable para almacenar el monto de la factura. Una variable para el total final (puede ser la misma si se sobrescribe el valor).
 3. No hay un ciclo, simplemente se evalúa si el monto supera los \$500.
 4. No es necesario, la consigna es clara.
 5. Facturas menores o iguales a \$500 (no se aplica descuento). Facturas mayores a \$500 (se aplica el 10% de descuento).

- **¿Como le explico a otra persona ?:**

Si el monto de la factura es mayor a \$500, se le descuenta el 10%. Si es \$500 o menos, se mantiene igual.

6.7) Determinar si un número ingresado es primo.

- **Datos:** Un número entero positivo.
 - **Restricciones:** El número debe ser mayor o igual a 2 (los primos comienzan desde el 2). Un número primo solo es divisible por 1 y por sí mismo. No se consideran los números negativos ni el 0 y 1 como primos.
 - **Qué me piden:** Determinar si el número ingresado es primo.
 - **Condición de entrada:** El
 - **Ciclo:** Se debe recorrer desde 2 hasta la raíz cuadrada del número para verificar si tiene divisores.
 - **Condición de salida:** Si encontramos un divisor antes de llegar al final del ciclo, el número no es primo y podemos detener la verificación.
1. Un for, iterando desde 2 hasta la raíz cuadrada del número (esto mejora la eficiencia, ya que nos ahorramos de iterar sobre mayor cantidad de números innecesariamente).
 2. Una variable para almacenar el número ingresado, Una variable booleana (esPrimo) que asumirá True al inicio y cambiará a False si encontramos un divisor.
 3. Si encontramos un divisor distinto de 1 y el propio número, podemos salir del ciclo antes.
 4. Sí, porque debemos manejar casos especiales (números menores a 2 no son primos).
 5. Probar con números primos (2, 3, 5, 7, 11, 13...). Probar con números compuestos (4, 6, 8, 9, 10...). Probar con números grandes para verificar la eficiencia del algoritmo.
- **¿Como le explico a otra persona ?:**

Un número es primo si solo es divisible por 1 y por sí mismo. Para verificarlo, probamos dividiendo el número por todos los valores desde 2 hasta su raíz cuadrada. Si encontramos algún divisor en ese rango, el número no es primo.

6.8) Verificar si dos cadenas de texto son iguales.

- **Datos:** Dos cadenas de texto.
- **Restricciones:** Se deben comparar exactamente, respetando mayúsculas y minúsculas. No se deben considerar espacios adicionales a menos que se especifique lo contrario.
- **Qué me piden:** Determinar si dos cadenas son iguales.
- **Condición de entrada:** Se reciben dos cadenas de texto.
- **Ciclo:** No es necesario un ciclo, ya que la comparación se hace de una vez.

- **Condición de salida:** Mostrar si las cadenas son iguales o no.
1. No se necesita un ciclo, ya que se puede comparar directamente ambas cadenas.
 2. Dos variables para almacenar las cadenas ingresadas. Una variable booleana (sonIguales) que será True si las cadenas coinciden y False si no.
 3. Se evalúa la igualdad con ==, lo que devuelve True o False.
 4. No es necesario, ya que la comparación es directa.
 5. Probar con cadenas idénticas, probar con diferencias de mayúsculas/minúsculas y probar con espacios adicionales al inicio o final.

- **¿Como le explico a otra persona ?:**

Tomamos dos cadenas de texto y verificamos si son exactamente iguales, incluyendo mayúsculas, minúsculas y espacios. Si son iguales, devolvemos True, o decimos que la expresión es verdadera; si no, False, o que las palabras no son idénticas.

6.9) Contar las vocales en una frase dada.

- **Datos:** Una cadena de texto (frase).
 - **Restricciones:** Solo se cuentan las vocales (a, e, i, o, u), tanto mayúsculas como minúsculas. Se ignoraran los espacios y otros caracteres que no sean vocales.
 - **Qué me piden:** Contar cuántas vocales hay en la frase.
 - **Condición de entrada:** Se recibe una cadena de texto.
 - **Ciclo:** Se recorre la cadena caracter por caracter y se cuentan las vocales.
 - **Condición de salida:** Mostrar el número total de vocales encontradas.
1. Se necesita un for para recorrer cada carácter de la frase.
 2. Una variable para almacenar la frase. Un contador (contadorVocales) que inicia en 0 y aumenta cuando encontramos una vocal.
 3. Cuando se ha revisado toda la frase.
 4. No es necesario, la consigna es clara.
 5. No necesito hacer muchas pruebas

- **¿Como le explico a otra persona ?:**

Recorremos la frase letra por letra y contamos cuántas son vocales (a, e, i, o, u). Para que funcione con mayúsculas y acentos, convertimos todo a minúsculas antes de comparar. Si la letra es una vocal (sin importar si es mayúscula o minúscula), sumamos 1 al contador.

6.10) Determinar cuántos días quedan para fin de mes a partir de una fecha dada.

- **Datos:** Una fecha dada (día, mes y año).
- **Restricciones:**
 - Se debe considerar cuántos días tiene el mes en curso.
 - Tener en cuenta los meses con diferente cantidad de días (28, 29, 30 o 31).
 - Si es febrero, verificar si el año es bisiesto (29 días en vez de 28).
- **Qué me piden:** Calcular cuántos días faltan para que termine el mes.
- **Condición de entrada:** Se recibe una fecha (día, mes, año).
- **Ciclo:** No es necesario, se puede resolver con cálculos directos.
- **Condición de salida:** Devolver el número de días restantes.

1. No se necesita un ciclo, solo cálculos con fechas.
2. día_actual , mes_actual , año_actual (fecha dada). días_en_mes para saber cuántos días tiene el mes actual. $\text{días_restantes} = \text{días_en_mes} - \text{día_actual}$ (cálculo final).
3. Se calcula el total de días del mes y se resta el día actual.
4. Sí, primero determinar la cantidad de días del mes y luego hacer la resta.
5.
 - a. Fechas de meses con 30 días (abril, junio, septiembre, noviembre).
 - b. Fechas de meses con 31 días (enero, marzo, mayo, julio, agosto, octubre, diciembre).
 - c. Febrero en un año común (28 días) y en un año bisiesto (29 días).
 - d. Último día del mes (debe dar 0).

- **¿Como le explico a otra persona ?:**

Primero determinamos cuántos días tiene el mes en la fecha ingresada. Luego, restamos el día actual a ese total para saber cuántos días quedan hasta fin de mes. Usamos una tabla de días por mes y verificamos si el año es bisiesto para febrero.