

# Resolución de Problemas

...

Programación  
2024

# Resolución de Problemas

1. Análisis.
2. Diseño.
3. Implementación.
4. Verificación y depuración.
5. Mantenimiento y documentación\*.

# Resolución de Problemas

## 1. Análisis.

2. Diseño.
3. Implementación.
4. Verificación y depuración.
5. Mantenimiento y documentación\*.

¿Qué problema tengo?

- Identificación de datos de entrada (input)
- Definición del resultado esperado (output)
- ¿Qué métodos conozco que puedan producir la salida deseada?
- ¿Cuento con requisitos adicionales y restricciones a la solución (idioma, formato, tecnologías, etc)?



Datos



Proceso



Información

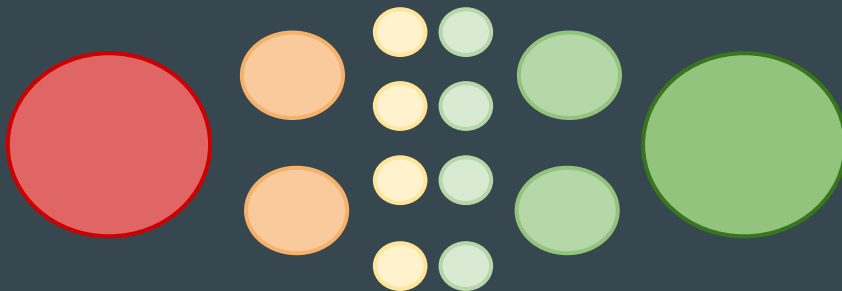
# Resolución de Problemas

1. Análisis.
2. Diseño.
3. Implementación.
4. Verificación y depuración.
5. Mantenimiento y documentación\*.

¿Cómo resuelvo el problema?

- ¿Qué recursos y herramientas dispongo?
- ¿Qué tiempo dispongo?
- ¿Es posible dividir el problema general en subproblemas? => *Divide y vencerás*

Problema

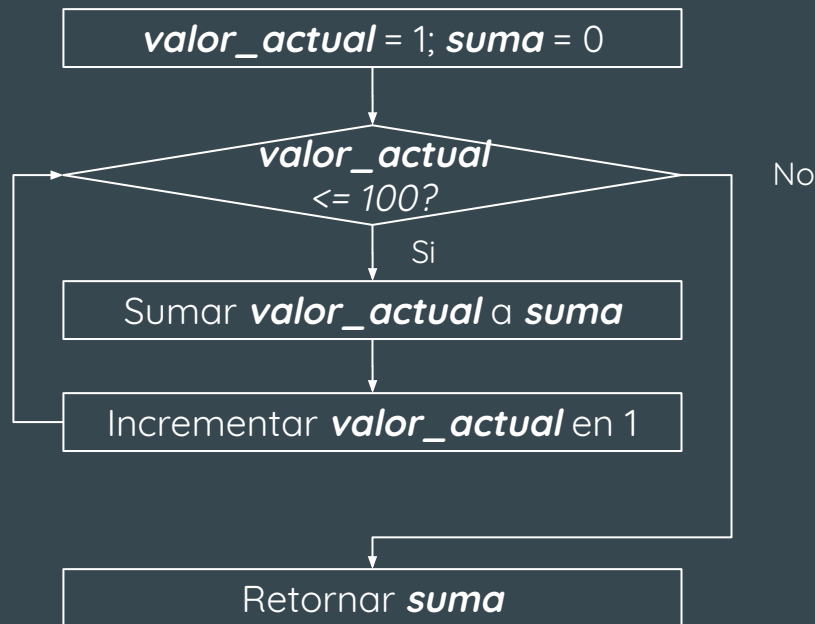


Solución

# Resolución de Problemas

Problema: calcular el valor de la suma  $1+2+3+\dots+100$ .

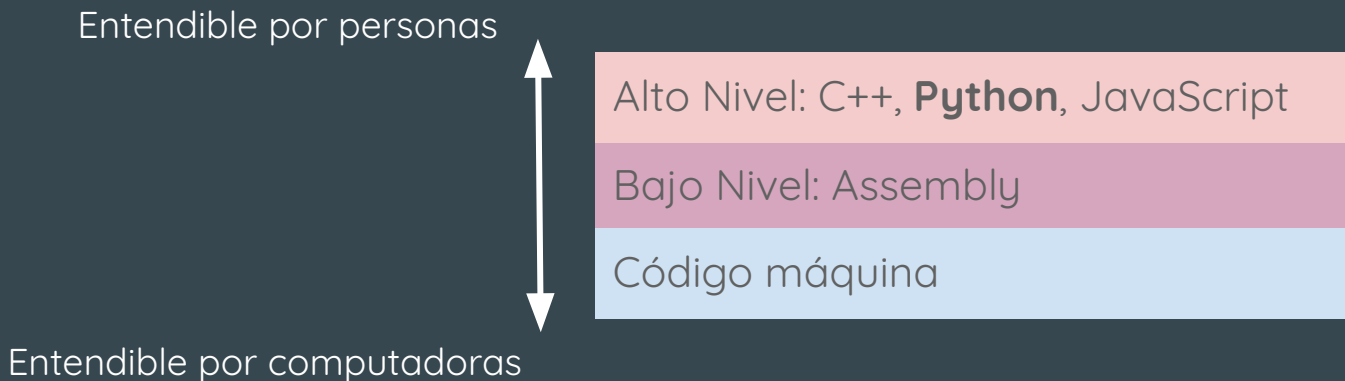
1. Establecer *valor\_actual* a 1
2. Establecer *suma* a 0
3. mientras *valor\_actual*  $\leq 100$  hacer
  - a. Sumar *valor\_actual* a *suma*
  - b. Incrementar *valor\_actual* en 1
4. Retornar *suma*



# Resolución de Problemas

1. Análisis.
2. Diseño.
3. Implementación.
4. Verificación y depuración.
5. Mantenimiento y documentación\*.

- El algoritmo computacional debe ser codificado usando un lenguaje de programación.



# Resolución de Problemas

1. Análisis.
2. Diseño.
3. Implementación.
4. Verificación y depuración.
5. Mantenimiento y documentación\*.

- Errores de compilación: uso incorrecto de las reglas del lenguaje de programación (errores de sintaxis).

```
>>> suma = 3 + 4)  
SyntaxError: unmatched ')'
```

- Errores de ejecución: instrucciones que la computadora puede comprender pero no ejecutar.

```
>>> div = 7 / 0  
ZeroDivisionError: division by zero
```

- Errores lógicos: errores en el diseño del algoritmo.

# Resolución de Problemas

1. Análisis.
2. Diseño.
3. Implementación.
4. Verificación y depuración.
5. Mantenimiento y documentación\*.

- El mantenimiento incluye la corrección de errores y el agregado de nuevas funcionalidades.
- La documentación incluye manuales de instrucciones y comentarios internos del funcionamiento del código.
- Los comentarios facilitan la identificación y depuración de errores.

```
# Calculo de un promedio
s = 14 + 21 + 6 # En s se guarda la suma de los valores
c = 3           # c es la cantidad de valores
p = s / c
```



# Ejemplo 1

Para aprobar una materia es necesario obtener un promedio de al menos 6.0. Si un alumno obtuvo las calificaciones 8.0, 5.0 y 5.5, logró aprobar la materia?

# Ejemplo 1

Para aprobar una materia es necesario obtener un promedio de al menos 6.0. Si un alumno obtuvo las calificaciones 8.0, 5.0 y 5.5, logró aprobar la materia?

## Planteo:

Entradas: notas;

Salidas: “aprobó” / “no aprobó”

- 1) Sumar notas.
- 2) Dividir por el número de notas.
- 3) Si promedio  $\geq 6.0$ , devolver “aprobó”;  
sino devolver “no aprobó”

## Resolución:

- 1)  $8.0 + 5.5 + 5.0 = 18.5$
- 2)  $18.5 / 3 = 6.16$
- 3)  $6.16 > 6.0?$  => “aprobó”

# Ejemplo 2

A partir de la siguiente lista de palabras, obtenga las 2 más largas y las 3 más cortas:

- Orquesta
- Mudos
- Sol
- Amordazar
- Vocales
- Gema
- Envoltura
- Mantel
- Limpiar
- Mostachos
- Citar
- Lento
- Panda
- Mosquito
- Trinchera
- Pagar
- Tortuga
- Significado
- Billetes
- Venecia
- Te
- Experimental

# Ejemplo 2

## Planteo

Entrada: lista de palabras

Salida: 3 palabras más cortas; 2 más largas

- 1) Por cada palabra: Calcular largo
- 2) Ordenar palabras por largo (menor -> mayor).
- 3) Tomar 3 primeras palabras de la lista.
- 4) Tomar 2 últimas palabras de la lista.

Te	2
Sol	3
Gema	4
Mudos	5
Citar	5
Lento	5
Panda	5
Pagar	5
Mantel	6
Vocales	7
Limpiar	7

Tortuga	7
Venecia	7
Orquesta	8
Mosquito	8
Billetes	8
Amordazar	9
Envoltura	9
Mostachos	9
Trinchera	9
Significado	11
Experimental	12

# Ejemplo 3

¿Cuántas palabras hay en las siguientes oraciones?

- “Esta es una oración.”
- “Esta también es una oración.”
- “A riesgo de parecer reiterativo, esta es otra oración.”
- “La última oración.”

Entrada: oracion (serie de letras y espacios);

Salida: número de palabras

# Ejemplo 3

Alternativa 1: contar espacios y sumar 1

- 1) Recorrer caracter por caracter y preguntar si es un espacio.
- 2) Almacenar el número de espacios.
- 3) Sumar 1

Resolución:

“E” es un espacio? No (número de espacios = 0)

“s” es un espacio? No (número de espacios = 0)

“t” es un espacio? No (número de espacios = 0)

“a” es un espacio? No (número de espacios = 0)

“ ” es un espacio? No (número de espacios = 1)

...

número de espacios: 3 => número de palabras: 4

# Ejemplo 3

Alternativa 2: contar espacios que estén antes de una letra y sumar 1

- 1) Recorrer caracter por caracter y preguntar si es un espacio o una letra.
- 2) Si es una letra y la anterior es un espacio, contar una palabra
- 3) Sumar 1

Resolución:

“L” es una letra y la anterior un espacio? No

“a” es una letra y la anterior un espacio? No

“ ” es una letra y la anterior un espacio? No

“ ” es una letra y la anterior un espacio? No

“ ” es una letra y la anterior un espacio? No

“u” es una letra y la anterior un espacio? Si

...