



Algoritmos y Programas



Introducción

- ✦ El objetivo de esta sesión consistirá en explicar los conceptos de algoritmo, programa y lenguaje de programación. Así como explorar otros temas concurrentes.

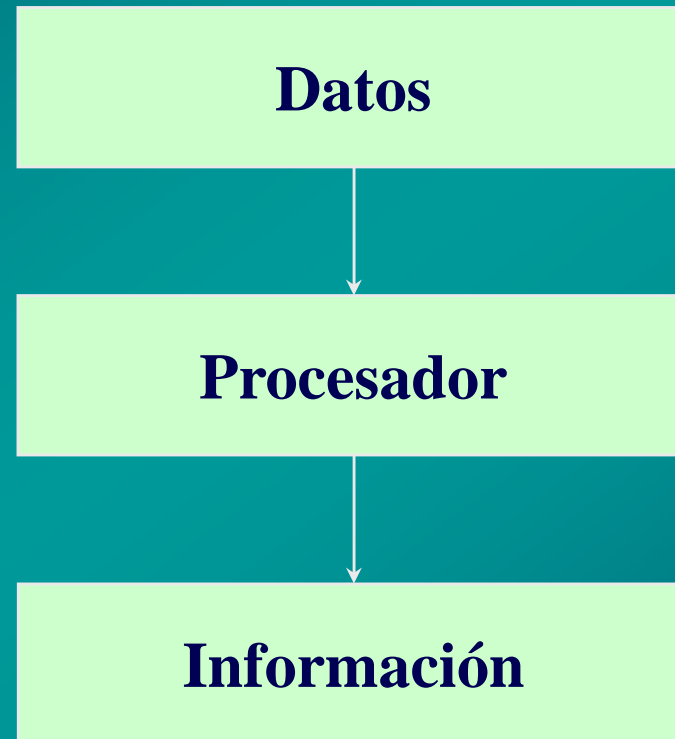


1. Sistemas de procesamiento de la información

- ✱ Un sistema de esta clase se define como un conjunto de componentes interconectados entre sí que transforman datos en información organizada, significativa y útil.



Esquemáticamente





2.2 SOLUCION DE PROBLEMAS

- ★ Fases de fragmentación:
 - ★ **Análisis del problema:** definición concisa a fin de que sea analizado en todo detalle.
 - ★ **Diseño del algoritmo:** procedimiento, paso a paso, para solucionar el problema dado.
 - ★ **Solución del algoritmo con la computadora:** codificación del algoritmo en un lenguaje de programación



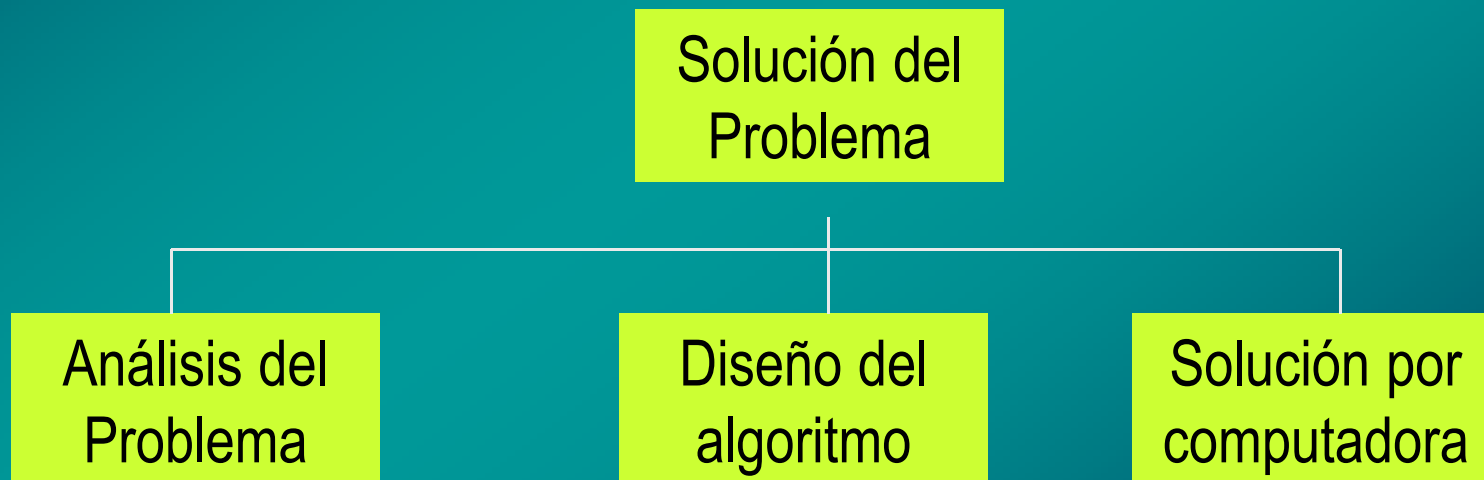
2.3 ANALISIS DEL PROBLEMA

- ✦ El propósito es ayudar al programador a llegar a un cierto grado de comprensión de la naturaleza del problema.
- ✦ Una buena **definición** del problema, junto con una descripción detallada de **entrada** y **salida**, son los requisitos más importantes para una solución eficaz.



2.3 ANALISIS DEL PROBLEMA cont.

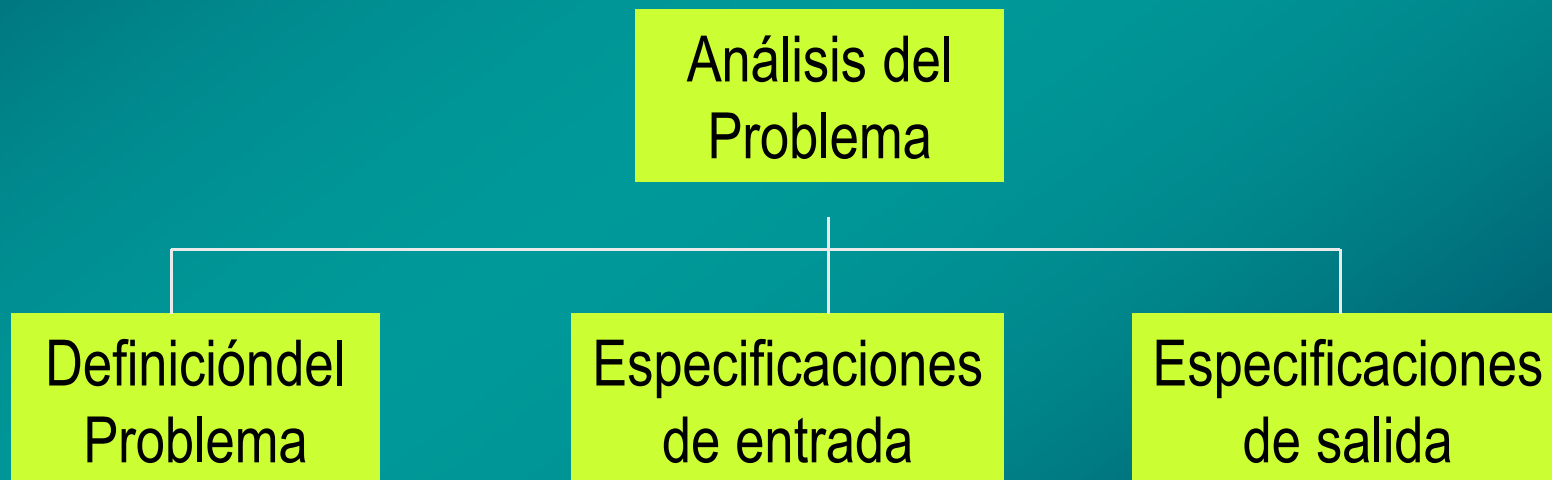
1. Que información debe proporcionar la solución del problema.





2.3 ANALISIS DEL PROBLEMA cont.

2. Que datos se necesitan para resolver el problema.





2.3 ANALISIS DEL PROBLEMA cont.

☀ Ejemplo: Realice un programa que permita hacer el Cálculo del área de un rectángulo e imprima sus valores.

- ✿ Entradas: valores de los lados (variable Altura y variable base).
- ✿ Salida: valor de la base, altura y área (variable área).
- ✿ Variables: Altura, Base, y Área.



2.4 DISEÑO DEL ALGORITMO

- ✱ La información proporcionada al algoritmo constituye su entrada y la información producida constituye su salida.
- ✱ Los problemas complejos se resuelven eficazmente si se fragmentan en subproblemas que sean más sencillos de solucionar que el original. Esta técnica es conocida como divide y vencerás (divide and conquer).

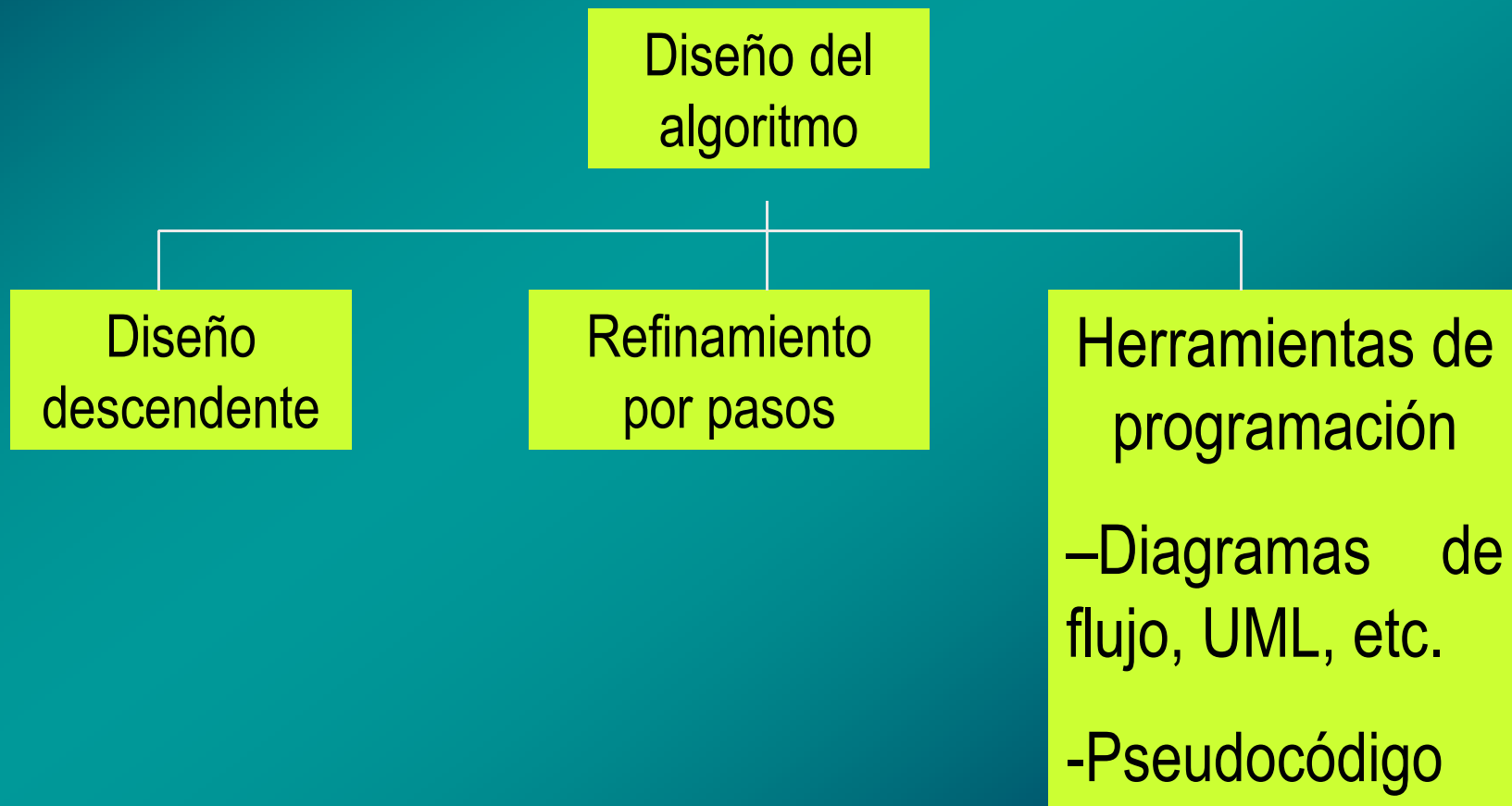


2.4 DISEÑO DEL ALGORITMO cont.

- ✱ La descomposición del problema original en subproblemas cuya solución puede implementarse a través de la computadora se le denomina diseño descendente (top down design)
- ✱ La descripción detallada de la solución mediante pasos específicos se le denomina refinamiento del algoritmo (stepwise refinement).



2.4 DISEÑO DEL ALGORITMO cont.





Concepto Inicial

- ★ **Programa:** colección de instrucciones de computador necesarias para resolver un problema determinado.
- ★ **Algoritmo:** método utilizado para resolver el problema
- ★ Un algoritmo puede expresarse de miles de maneras, dependiendo del:
 - Lenguaje de programación
 - Sistema computacional
 - Programador



1. Una Definición más estructurada

- ✱ Un algoritmo consiste en un grupo **finito** de instrucciones **precisas** que **definen la secuencia** de operaciones a realizar para resolver un sistema o problema dado.



Características de los algoritmos

- ✱ Preciso, indicará el orden de cada paso.
- ✱ Definido, el mismo resultado se obtendrá al ejecutar el algoritmo “n” ocasiones.
- ✱ Finito, deberá terminar en algún momento.



Características de los algoritmos

- ☀ Debe constar de 3 partes:
 - ✿ Entrada: información dada al algoritmo
 - ✿ Proceso: Cálculos necesarios para encontrar la solución del problema.
 - ✿ Salida: respuestas dadas por el algoritmo o *resultados finales* de los cálculos.



2. Metodología de la programación

- ✦ Es aquella que es necesaria para resolver problemas mediante programas, su **eje es el algoritmo.**





Como pasar del algoritmo al programa

- ☀ Brecha lingüística



- ☀ Lenguajes de programación ayudan a cerrar la brecha lingüística entre lo que queremos decir y lo que el computador entiende



Pasos para la solución de un problema en función del concepto algoritmo

- ✱ El diseño del algoritmo describe el análisis del problema y desarrollo del algoritmo.
- ✱ Expresar el algoritmo como un programa en un lenguaje de programación.
- ✱ Ejecutar y validar el programa por la computadora.

Ejemplo 2.1 Genere un algoritmo que solucione el siguiente problema:

Un cliente ejecuta un pedido a una fábrica. La fábrica examina en su banco de datos la ficha del cliente, si el cliente es solvente entonces la empresa acepta el pedido, en caso contrario rechaza el pedido.

- ✱ Inicio.
- ✱ Leer el pedido.
- ✱ Examinar el historial crediticio del cliente
- ✱ Si el cliente es solvente, entregar el pedido. En caso contrario, rechazarlo.
- ✱ Fin



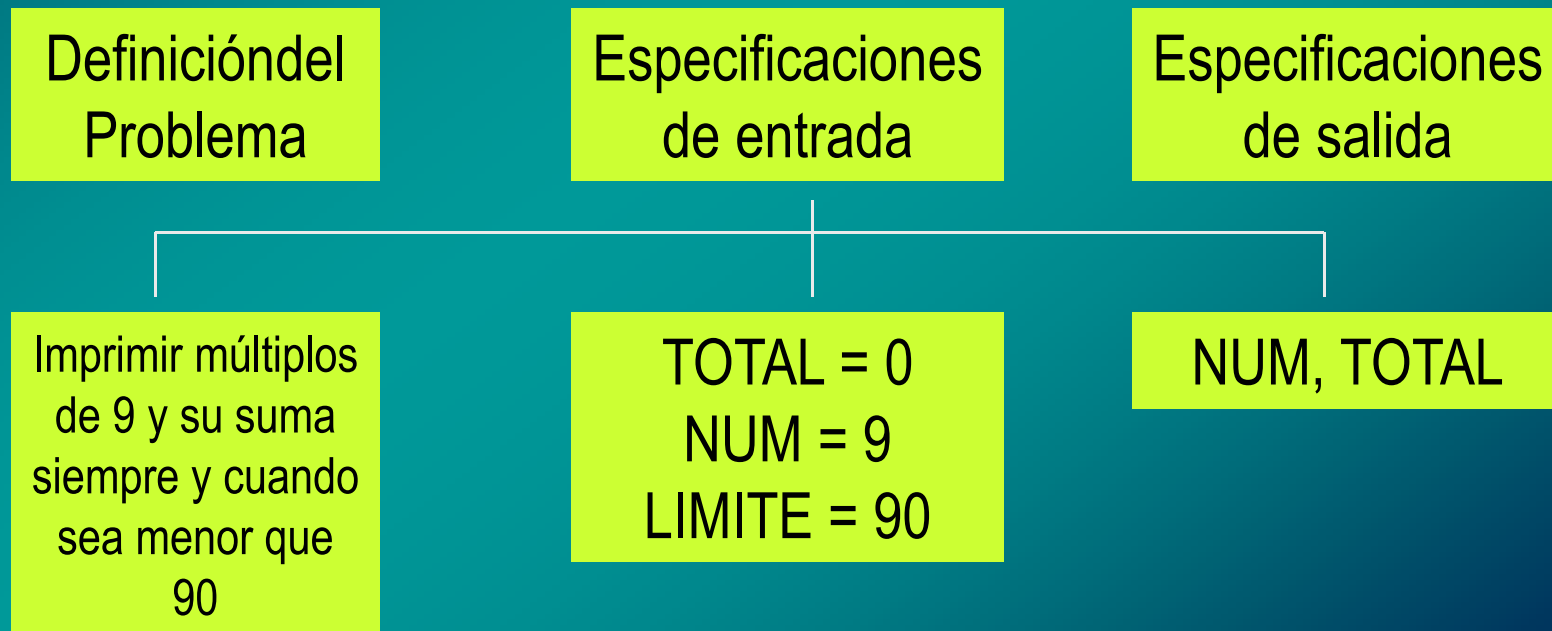
Verificación de algoritmos

- ✱ Es necesario asegurarse que el mismo realiza las tareas para las que se ha diseñado y produce el resultado correcto y esperado.
- ✱ El modo más simple es la ejecución manual usando datos significativos que abarque todo el posible rango de valores y anotando en una hoja de papel los valores que van tomando en las diferentes fases, los datos de entrada o auxiliares y por último los valores de los resultados
- ✱ **PRUEBA DEL ALGORITMO**



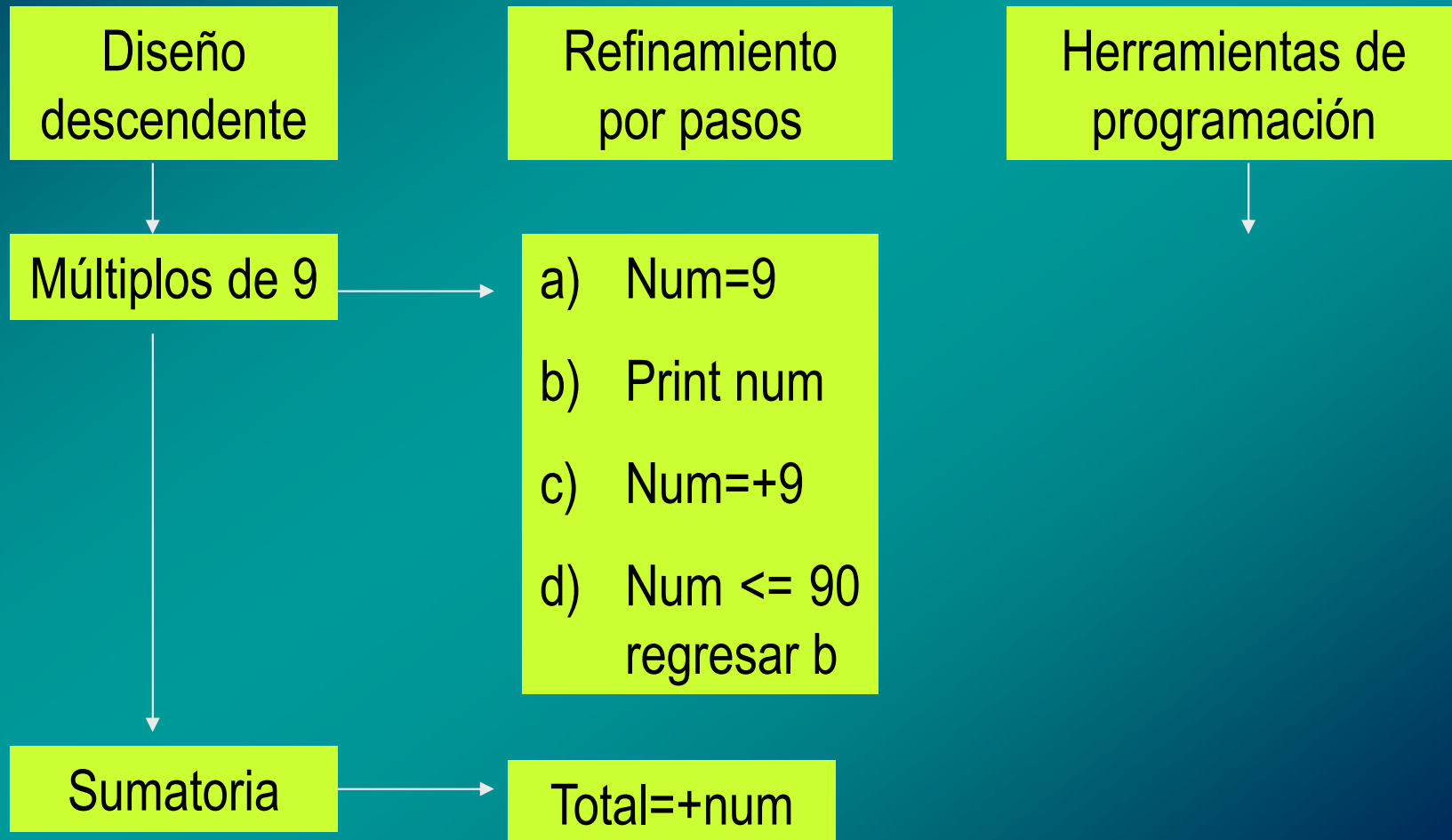
2.4 DISEÑO DEL ALGORITMO cont.

- ✦ Ejemplo: Determine a los números que sean múltiplos de 9 hasta 90 , así como su sumatoria.





2.4 DISEÑO DEL ALGORITMO cont.





2.4 DISEÑO DEL ALGORITMO cont.

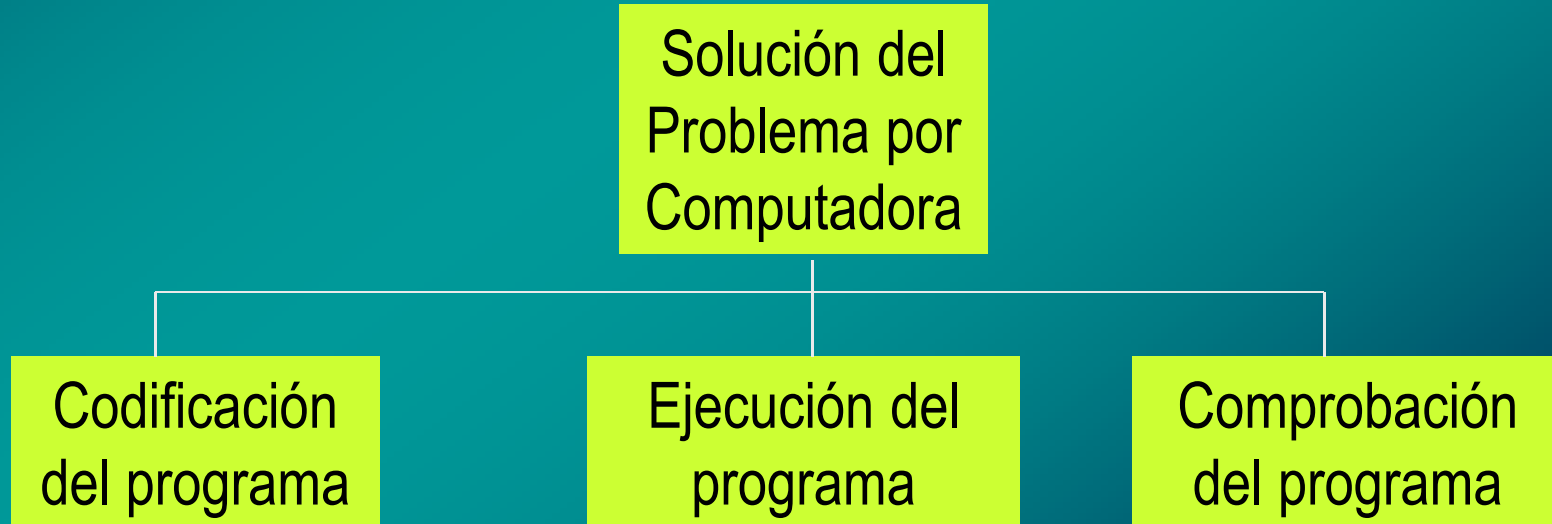
1. Inicio
2. $Total=0$, $Num=9$, $Limite=90$
3. Print Num
4. $Total=Total+Num$
5. Si $Num < Limite$ entonces
 $Num=Num+9$ y regresar a 3
6. Print Total
7. Fin

Hecho en DF



2.5 SOLUCION DE PROBLEMAS MEDIANTE LA COMPUTADORA

- Una vez diseñado el algoritmo y representado mediante una herramienta de programación se debe:





2.6 REPRESENTACION GRAFICA DE LOS ALGORITMOS

- ☀ Los métodos usuales para representarlo son:
 1. Diagramas de flujo
 2. Diagramas UML
 1. Caso de Uso
 2. Clases
 3. Lenguaje de especificación de algoritmo
 1. Pseudocódigo



2.7 PSEUDOCODIGO

- ✱ Es un lenguaje de especificación de algoritmos que permite una codificación rápida y simple.
- ✱ Su ventaja radica en que el programador no debe preocuparse por la sintaxis de un lenguaje de programación en particular.
- ✱ Utiliza palabras sencillas de entender para codificar programas.



PSEUDOCODIGO cont.

- ✱ Se pueden usar palabras de nuestro lenguaje español, o en inglés.
- ✱ Eso queda a criterio de la persona que realice el **algoritmo**.
- ✱ No hay forma exacta para realizar un **algoritmo**, sino que se entienda fácilmente tanto para el que lo realiza, como para el que lo tenga que revisar.



Ejemplo en pseudocódigo

```
★ [codigo]valor = 0
  Mientras (valor sea menor a
    20) {
      mostrar valor
      valor = valor + 1
      Si (valor es igual a
        10) {
          mostrar "me
            gusta el 10"
        }
    }
[/codigo]
```

```
★ [codigo]Inicio Programa
  valor = 0
  Mientras (valor sea menor a 20)
    Mostrar valor
    valor++
    Si (valor es igual a 10)
      Mostrar "me gusta
        el 10"
    Fin Si
  Fin Mientras
Fin Programa
[/codigo]
```



Ejemplo 2

- ✱ Una persona ha trabajado 40 horas en una semana dada y gana a razón de \$10 la hora. El impuesto de renta es de 13.5% de su pago bruto. Se desea saber cuál es el pago bruto, el monto del impuesto y el pago neto del trabajador.



Ejemplo 2 cont.

- ✱ La información dada es:
- ✱ Horas trabajadas: 40
- ✱ Impuestos: 13.5%
- ✱ Paga/Hora: \$10

✱ ENTRADA:

- ✱ Hr.Tra:40
- ✱ Impuest.:13.5%
- ✱ Paga/Hr: \$10

SALIDA:

Pago Bruto
Tot. Imp. A pagar
Pago Neto



Ejemplo 2.2 Genere un algoritmo que sume los números entre el 3 y el 30 y presente el resultado de suma.

1. Inicio.
2. Hacer SUMA igual cero.
3. Establecer NUMERO igual a 3.
4. Sumar NUMERO a SUMA.
5. Incrementar NUMERO en 1
6. Si NUMERO es menor o igual que 30 ir a 4; si no imprimir a SUMA.
7. Fin

