

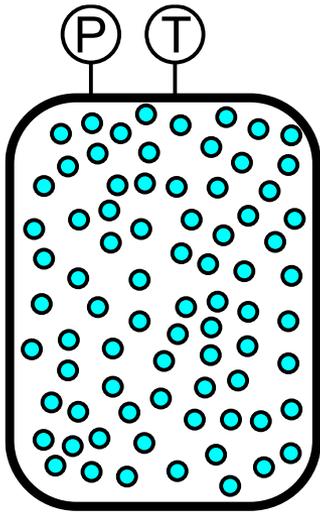
# Modelado Matemático y herramientas para su resolución 2023

Prof.: Dr. Juan Ignacio Manassaldi

JTP: Ing. Amalia Rueda

- El procedimiento metodológico fundamental para resolver un problema en ingeniería consiste en representarlo de una manera adecuada, de tal forma de lograr una **sustitución del sistema real** (equipo, proceso, etc.) por uno más adecuado para el tratamiento formal.
- Las herramientas lógico-matemáticas nos brindan un marco útil para representar mediante un sistema de símbolos y reglas, el comportamiento de los sistemas reales.
- Un **modelo matemático** se define, de manera general, como una formulación o una ecuación que expresa las características esenciales de un sistema físico o de un proceso en términos matemáticos.

EOS  
Ecuaciones de estado



$$Pv = RT \quad \text{Ley de los gases ideales}$$

Ecuación de Van der Waals

$$P = \frac{RT}{v-b} - \frac{a}{v^2}$$

$$a = \frac{0.421875R^2T_c^2}{P_c} \quad b = \frac{RT_c}{8P_c}$$

$$P = \frac{RT}{v-b} - \frac{a\alpha}{(v^2 + 2bv - b^2)}$$

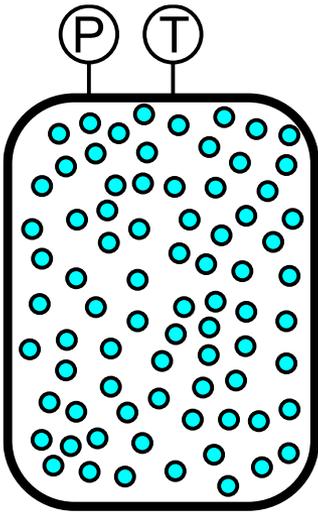
Ecuación Peng Robinson

$$a = \frac{0.45724R^2T_c^2}{P_c} \quad b = \frac{0.07780RT_c}{P_c}$$

$$\alpha(T) = \left[ 1 + (0.37464 + 1.54226\omega - 0.2699\omega^2) \left( 1 - \sqrt{T/T_c} \right) \right]^2$$

EOS

Ecuaciones de estado



$$P = \sum_{n=1}^9 a_n \rho^n + \exp(-\rho_r^2) \sum_{n=10}^{15} a_n \rho^{2n-17}$$

$$a_1 = RT$$

$$a_2 = b_1 T + b_2 T^{1/2} + b_3 + b_4/T + b_5/T^2$$

$$a_3 = b_6 T + b_7 + b_8 T + b_9/T^2$$

$$a_4 = b_{10} T + b_{11} + b_{12}/T$$

$$a_5 = b_{13}$$

$$a_6 = b_{14}/T + b_{15}/T^2$$

$$a_7 = b_{16}/T$$

$$a_8 = b_{17}/T + b_{18} T^2$$

$$a_9 = b_{19}/T^2$$

$$a_{10} = b_{20}/T^2 + b_{21}/T^3$$

$$a_{11} = b_{22}/T^2 + b_{23}/T^4$$

$$a_{12} = b_{24}/T^2 + b_{25}/T^3$$

$$a_{13} = b_{26}/T^2 + b_{27}/T^4$$

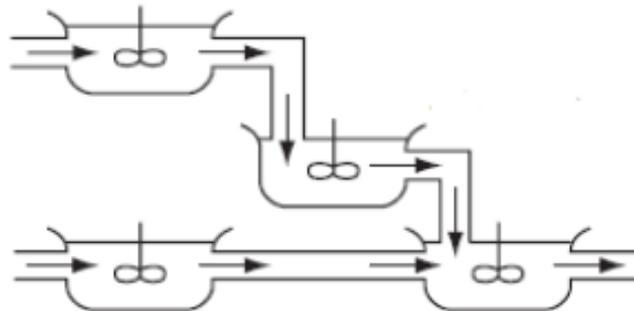
$$a_{14} = b_{28}/T^2 + b_{29}/T^3$$

$$a_{15} = b_{30}/T^2 + b_{31}/T^3 + b_{32}/T^4$$

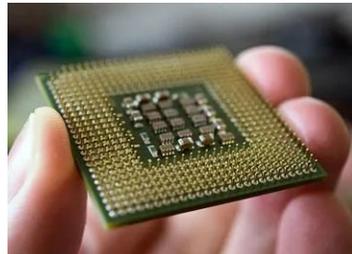
Ecuaciones de Benedict-Webb-Rubin modificadas (mBWR)

- Si bien el sistema real a estudiar es único, puede existir un número muy grande de modelos asociados al mismo.
- Para obtener un modelo que pueda resolverse (es decir que sea útil), resulta necesario adoptar un conjunto de hipótesis.
- No toda ecuación (o sistema de ecuaciones) puede resolverse fácilmente, al menos desde el punto de vista analítico.
- Esto impuso a lo largo de la historia limitaciones importantes al tipo de modelos que podían resolverse, o de otra forma, la necesidad de recurrir a hipótesis inadecuadas o restrictivas (super-simplificaciones) para al menos poder tratar el problema.
- Contar con herramientas potentes para resolver sistemas de ecuaciones, o lo que es lo mismo, relativizar la necesidad de adoptar hipótesis inadecuadas al plantear modelos para resolver problemas complejos, resultó un gran paso adelante.

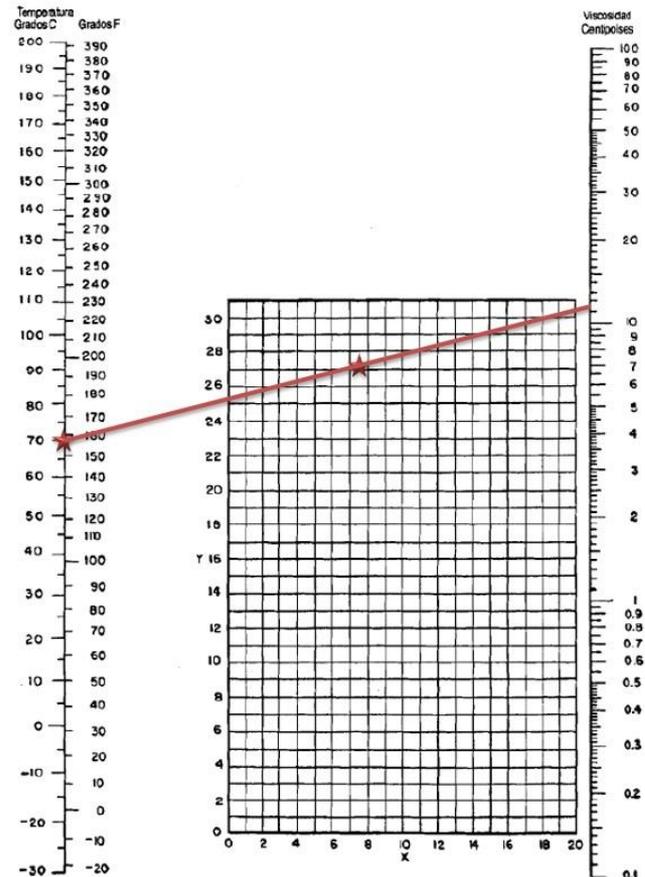
- La simulación es el acto de representar algunos aspectos del mundo real por números o símbolos que pueden manipularse fácilmente para facilitar su estudio.
- La simulación digital por computadora constituye una poderosa herramienta para la resolución de las ecuaciones (modelos matemáticos) que describen a los sistemas en ingeniería química.
- Durante el cursado iremos desarrollando métodos que facilitan la resolución de modelos matemáticos complejos. En su mayoría se agrupan bajo la clasificación de **métodos numéricos**.



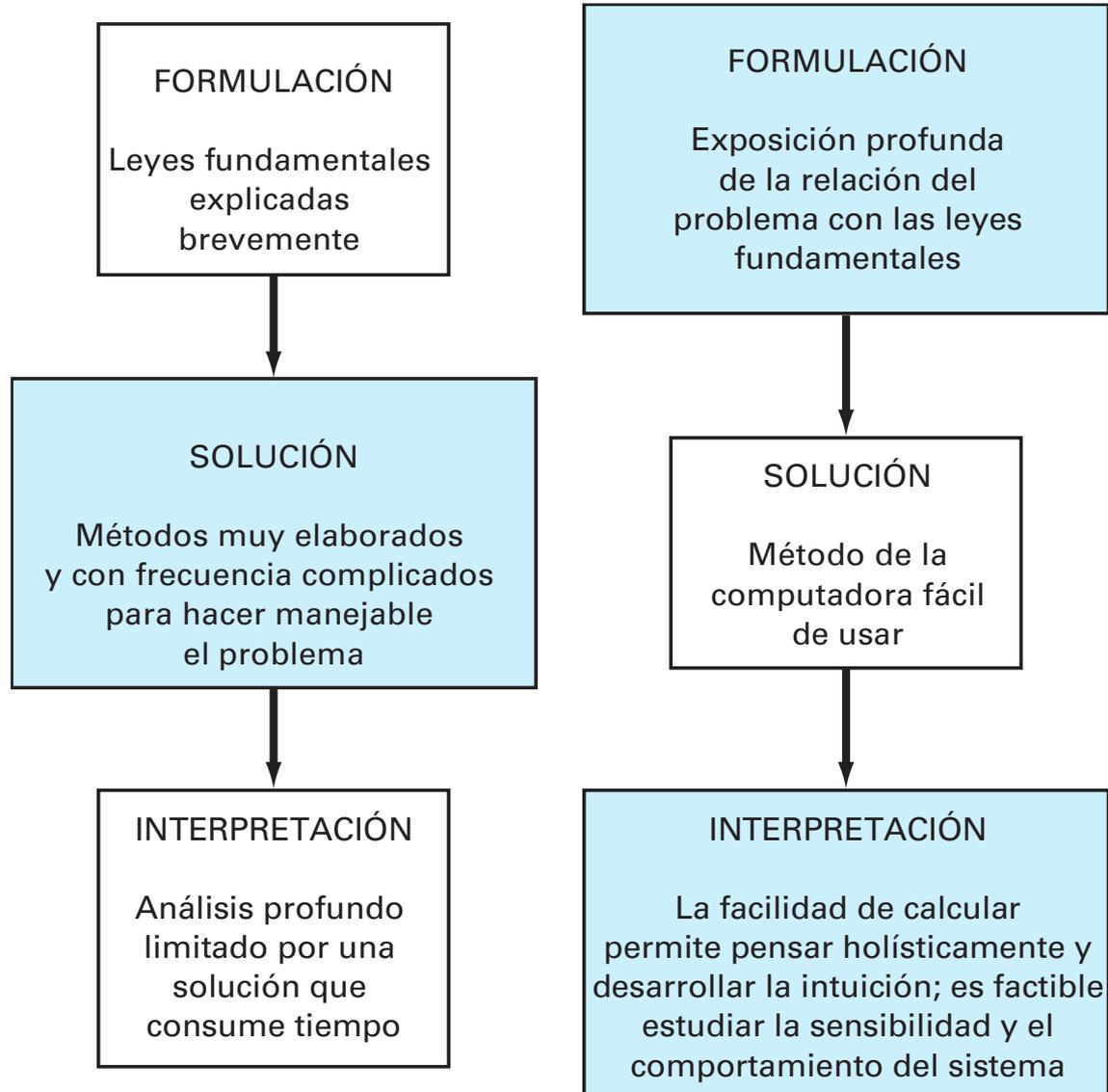
- Los métodos numéricos constituyen técnicas mediante las cuales es posible formular problemas matemáticos, de tal forma que puedan resolverse utilizando operaciones aritméticas.
- Utilizan únicamente operaciones lógicas y aritméticas; por consiguiente, pueden implementarse sobre computadoras digitales.
- Fueron desarrollados hace muchos años pero su uso estuvo restringido hasta el advenimiento de las computadoras personales.

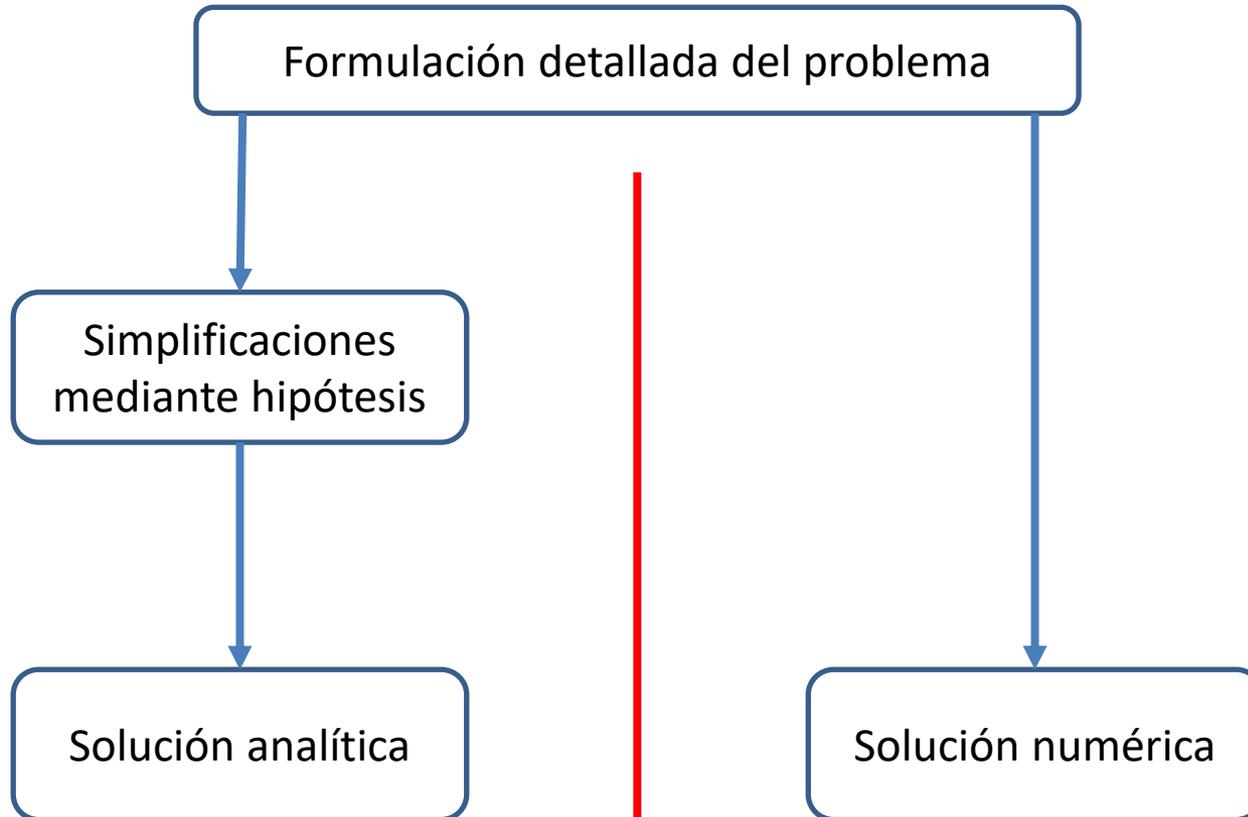


- Se encontraban las soluciones de algunos problemas usando métodos exactos o analíticos.
- Para analizar el comportamiento de los sistemas se usaban soluciones gráficas, las cuales tomaban la forma de gráficas o nomogramas.
- Para implementar los métodos numéricos se utilizaban calculadoras y reglas de cálculo.
- Se gastaba bastante energía en la técnica misma de solución en lugar de usarla en la definición del problema y su interpretación.



- Las computadoras y los métodos numéricos ofrecen una alternativa para los cálculos complicados.
- Se obtienen soluciones directamente, de esta manera se pueden aproximar los cálculos sin tener que recurrir a consideraciones de simplificación o a técnicas muy lentas.
- Los métodos numéricos representan opciones que aumentan, en forma considerable, la capacidad para enfrentar y resolver los problemas.
- Los métodos numéricos son herramientas muy poderosas para la solución de problemas. Son capaces de manipular sistemas de ecuaciones grandes, manejar no linealidades y resolver geometrías complicadas, comunes en la práctica de la ingeniería y, a menudo, imposibles de resolver en forma analítica.





**Solución exacta de un modelo simplificado**

**Solución aproximada de un modelo detallado**

Tendencia actual



$$F = ma$$

$$F = F_D - F_U$$

$$F_D = mg$$

¿ $F_U$ ? Arrastre o fricción

$$F_U = cv$$

$$F_U = c'v^2$$

$$F_U = \alpha v + \chi v^{\gamma}$$

$$\frac{dv}{dt} = \frac{F}{m} = \frac{F_D - F_U}{m}$$

$$\frac{dv}{dt} = g - \frac{F_U}{m}$$

$$\rightarrow \frac{dv}{dt} = g - \frac{c}{m}v$$

$$\rightarrow \frac{dv}{dt} = g - \frac{c'}{m}v^2$$

$$\rightarrow \frac{dv}{dt} = g - \frac{1}{m}(\alpha v + \chi v^{\gamma})$$

Modelo cada vez  
mas detallado



$$\frac{dv}{dt} = g - \frac{c}{m} v$$

$$\rightarrow v = \frac{gm}{c} (1 - e^{-(c/m)t})$$

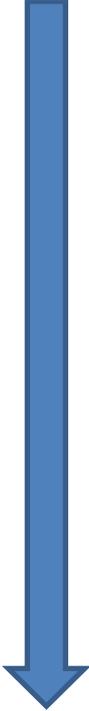
$$\frac{dv}{dt} = g - \frac{c'}{m} v^2$$

$$\rightarrow v = \sqrt{\frac{gm}{c'}} \tanh\left(\sqrt{\frac{gc'}{m}} t\right)$$

$$\frac{dv}{dt} = g - \frac{1}{m} (\alpha v + \chi v^r)$$

$$\rightarrow v = ?$$

Solución analítica cada  
vez mas compleja



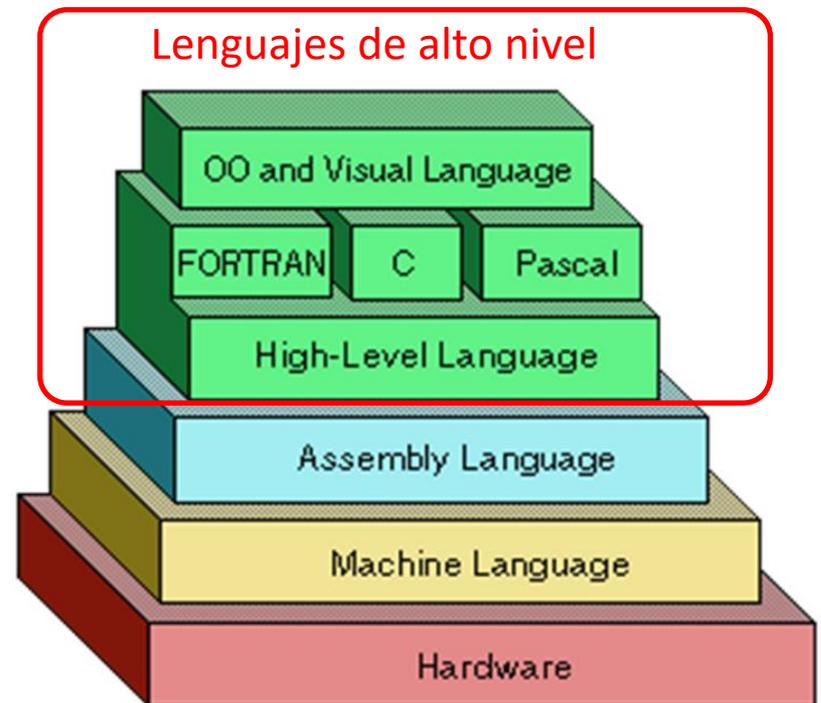
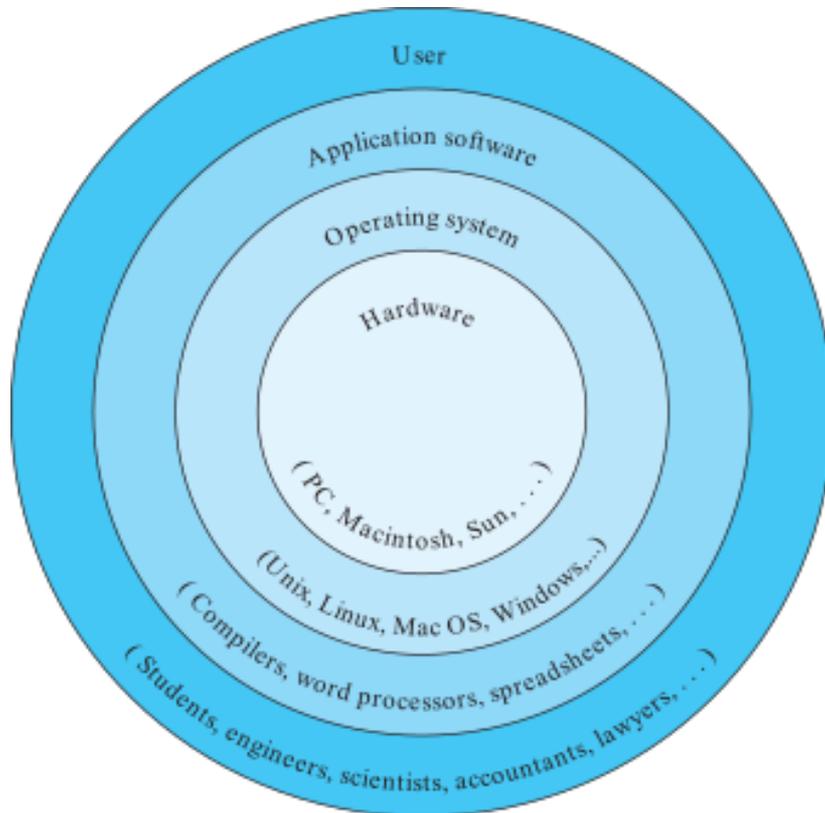
Los métodos numéricos amplían el abanico de problemas a resolver.  
Como contrapartida su solución no es exacta (tiene asociado un error).  
Nuestro objetivo es controlar o acotar ese error dentro de valores tolerables

- La matemática clásica, aún en manos de los más ingeniosos matemáticos aplicados no puede resolver problemas con el nivel requerido por la tecnología actual.
- La combinación de métodos numéricos y computadoras digitales constituye una poderosa herramienta para el análisis matemático.
- Debido a la cantidad de cálculos que deben realizarse, los métodos numéricos resultan eficientes solo al implementarse en computadoras digitales (ganamos rapidez).

- Existen dos tipos de usuarios de software:
  - **Usuario tradicional o limitado:** Se limita a las capacidades que encuentran en el modo estándar de operación del software existente.  
**El problema debe adaptarse al software.**
  - **Usuario avanzado o potente:** Amplía las capacidades nativas del software mediante herramientas de programación.  
**El software se adapta al problema.**

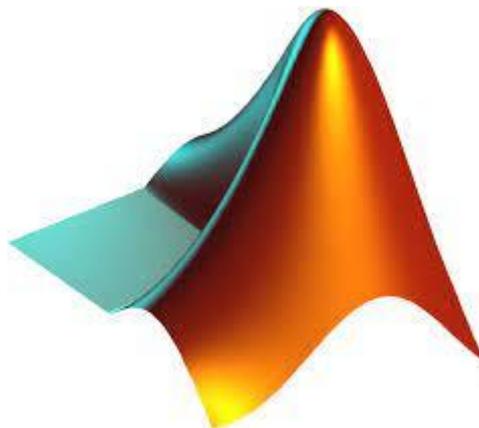
**Un usuario avanzado puede comportarse como un usuario tradicional si el problema lo amerita.**

- Los programas computacionales son únicamente conjuntos de instrucciones que dirigen a la computadora para realizar una cierta tarea.

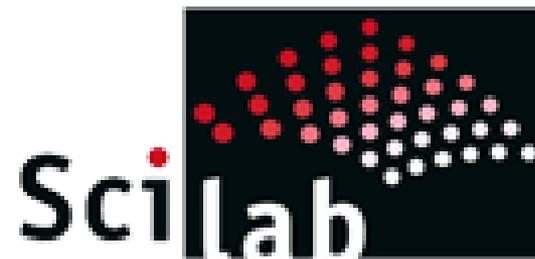




Excel + VBA



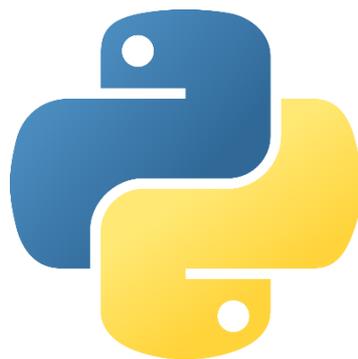
MATLAB



Scilab



GNU Octave



Python

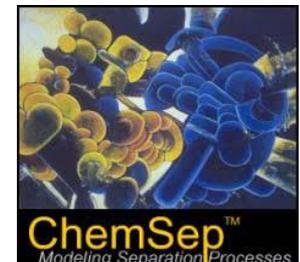


C y C++

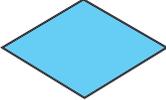
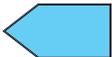
- El listado anterior corresponde a softwares de propósito general y sirven para la resolución de problemas numéricos de la mayoría de las ingenierías.
- Por otro lado, cada especialidad tiene también softwares para la resolución puntual de sus problemas específicos (software enlatado).
- En este tipo de herramientas informáticas la resolución ya viene programada internamente.
- Un ejemplo clásico de ingeniería química son los **simuladores de procesos**:



DWSIM

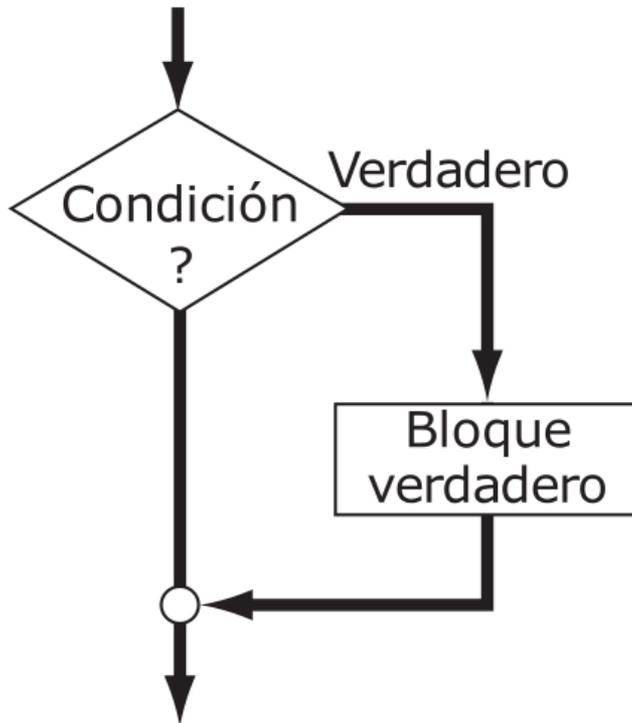


- Es un conjunto de reglas que desarrollan en el programador los hábitos para lograr un buen estilo.
- La idea clave detrás de la programación estructurada es que cualquier algoritmo numérico requiere tan sólo de tres estructuras de control fundamentales: **secuencia, selección y repetición**.
- Un diagrama de flujo es una representación visual o gráfica de un **algoritmo**.
- Otra manera de expresar algoritmos, y que constituye un puente de unión entre los diagramas de flujo y el código de la computadora, es el **pseudocódigo**. En esta técnica se utilizan expresiones semejantes a las del código, en lugar de los símbolos gráficos del diagrama de flujo.

SÍMBOLO	NOMBRE	FUNCIÓN
	Terminal	Representa el inicio o el final de un programa.
	Líneas de flujo	Representan el flujo de la lógica. Los arcos en la flecha horizontal indican que ésta pasa sobre las líneas de flujo verticales y no se conecta con ellas.
	Proceso	Representa cálculos o manipulación de datos.
	Entrada/Salida	Representa entrada o salida de datos e información.
	Decisión	Representa una comparación, una pregunta o una decisión que determina los caminos alternativos a seguir.
	Unión	Representa la confluencia de líneas de flujo.
	Conexión de fin de página	Representa una interrupción que continúa en otra página.
	<i>Ciclo</i> de cuenta controlada	Se usa para <i>ciclos</i> que repiten un número predeterminado de iteraciones.

- **Secuencia.** Expresa la trivial idea de que, a menos que se indique otra cosa, el código debe realizarse instrucción por instrucción.
- **Selección.** La selección nos ofrece un medio de dividir el flujo del programa en ramas considerando el resultado de una condición lógica.
  - Una sola alternativa: IF/THEN
  - Dos alternativas: IF/THEN/ELSE
  - Múltiples alternativas.

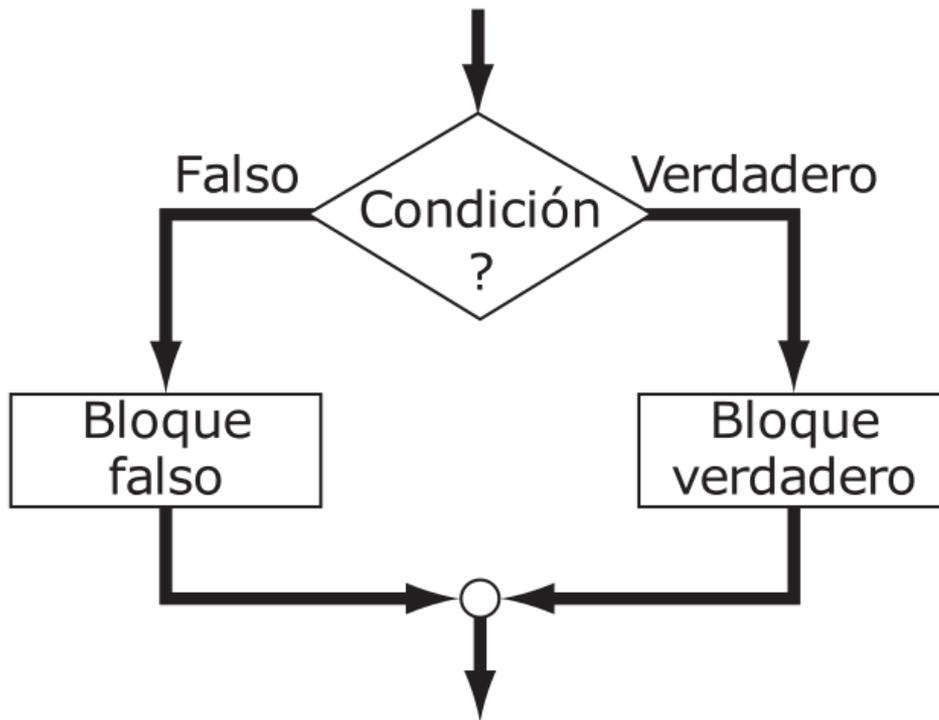
Diagrama de flujo



Pseudocódigo

```
IF condición THEN  
Bloque verdadero  
ENDIF
```

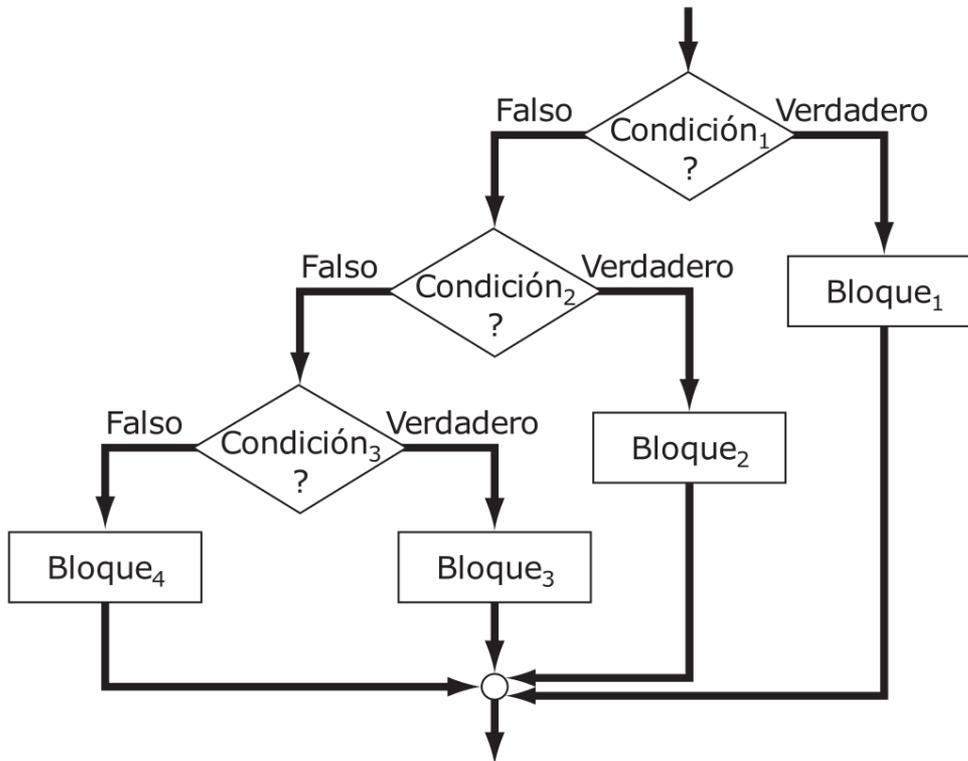
Diagrama de flujo



Pseudocódigo

```
IF condición THEN  
Bloque verdadero  
ELSE  
Bloque falso  
ENDIF
```

Diagrama de flujo

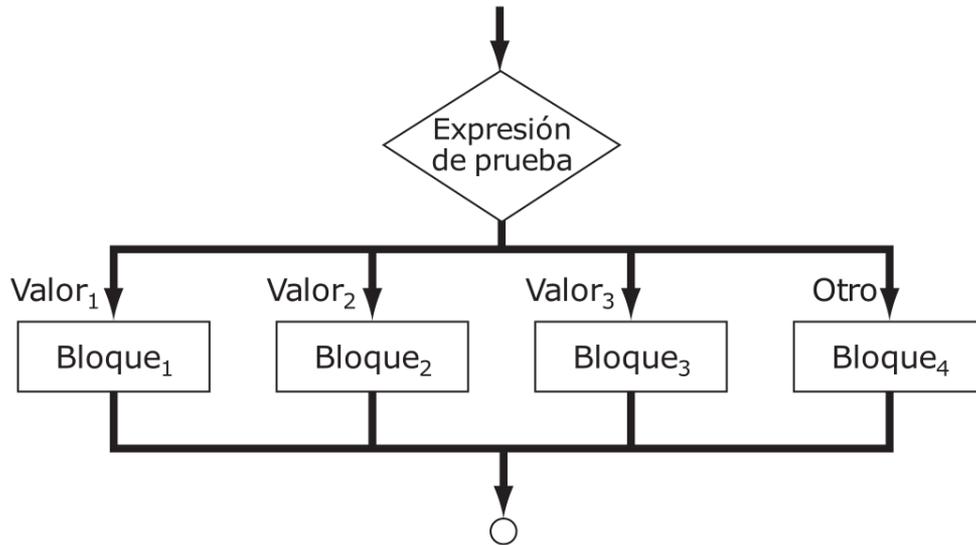


Pseudocódigo

```

IF condición1 THEN
    Bloque1
ELSEIF condición2
    Bloque2
ELSEIF condición3
    Bloque3
ELSE
    Bloque4
ENDIF
    
```

Diagrama de flujo

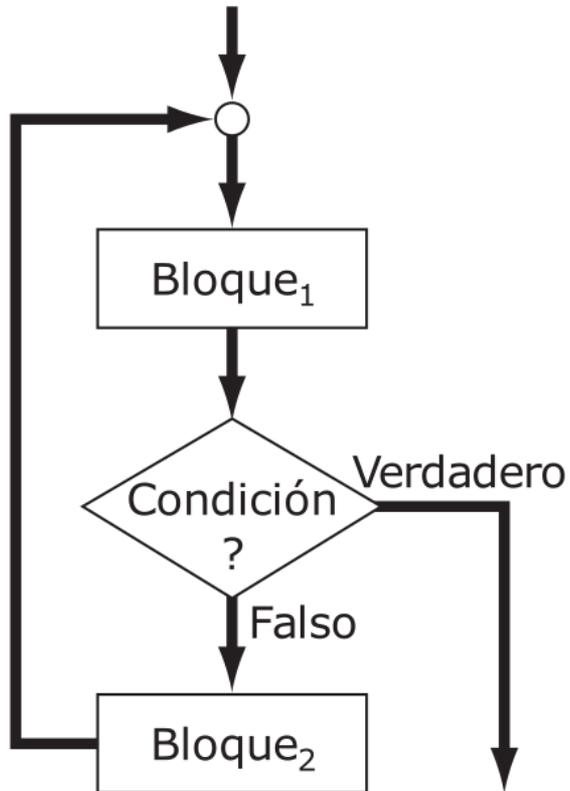


Pseudocódigo

```
SELECT CASE Expresión de prueba  
CASE Valor1  
Bloque1  
CASE Valor2  
Bloque2  
CASE Valor3  
Bloque3  
CASE ELSE  
Bloque4  
END SELECT
```

- **Repetición.** La repetición nos proporciona una manera de llevar a cabo instrucciones repetidamente. Las estructuras resultantes, llamadas **loops** o **ciclos**, se presentan en dos formas distintas que se diferencian por la manera en que terminan.
  - El primer tipo, y el fundamental, es el llamado **loop de decisión** (o loop *DOEXIT*) debido a que termina basándose en el resultado de una condición lógica.
  - El segundo tipo es el **loop controlado por contador** (o loop *DOFOR*) que realiza un número determinado de repeticiones o iteraciones.

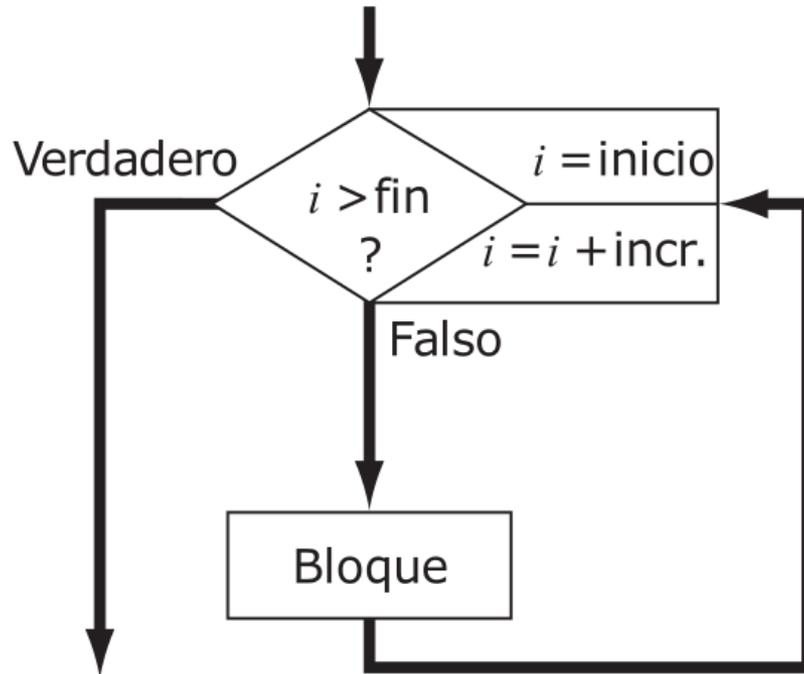
Diagrama de flujo



Pseudocódigo

```
DO  
Bloque1  
IF condición EXIT  
Bloque2  
ENDDO
```

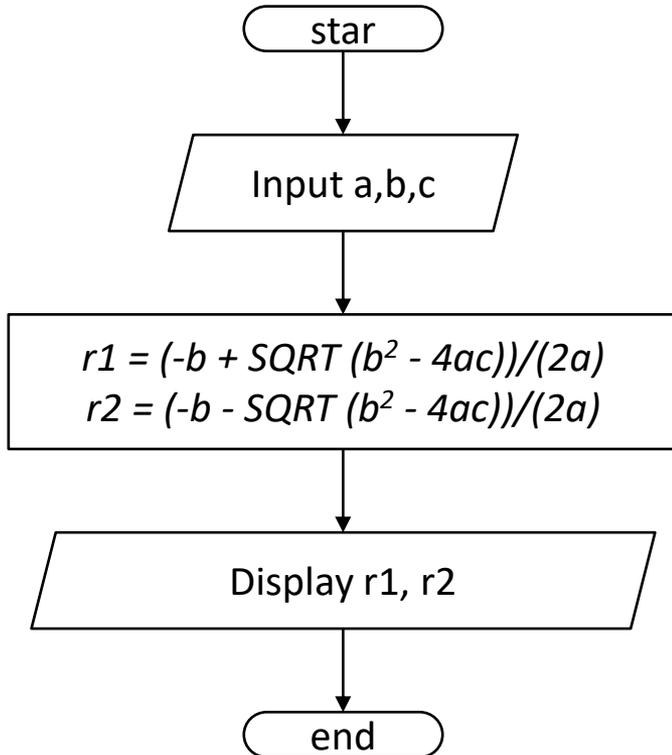
Diagrama de flujo



Pseudocódigo

***DOFOR***  $i = inicio, incremento, fin$   
*Bloque*  
***ENDDO***

Diagrama de flujo



Pseudocódigo

**INPUT**  $a, b, c$   
 $r1 = (-b + \text{SQRT}(b^2 - 4ac))/(2a)$   
 $r2 = (-b - \text{SQRT}(b^2 - 4ac))/(2a)$   
**DISPLAY**  $r1, r2$

```
INPUT  $a, b, c$   
IF  $a = 0$  THEN  
    IF  $b \neq 0$  THEN  
         $r1 = -c/b$   
        DISPLAY  $r1$   
    ELSE  
        DISPLAY "No hay solución"  
    ENDIF  
ELSE  
     $discr = b^2 - 4 * a * c$   
    IF  $discr \geq 0$  THEN  
         $r1 = (-b + \text{Sqrt}(discr)) / (2 * a)$   
         $r2 = (-b - \text{Sqrt}(discr)) / (2 * a)$   
        DISPLAY  $r1, r2$   
    ELSE  
        DISPLAY "Raíces Complejas"  
    ENDIF  
ENDIF
```

```
INPUT  $\underline{v}$ ,  $n$   
 $suma = 0$   
DOFOR  $i = 1, 1, n$   
 $suma = suma + v(i)$   
ENDDO  
DISPLAY  $suma$ 
```

```
INPUT  $\underline{v}$ ,  $n$   
 $suma = v(1)$   
DOFOR  $i = 2, 1, n$   
 $suma = suma + v(i)$   
ENDDO  
DISPLAY  $suma$ 
```

$\underline{v}$ : vector de trabajo  
 $n$ : tamaño del vector

```
DOFOR  $i = inicio, incremento, fin$   
Bloque  
ENDDO
```

- Los programas de computación se dividen en subprogramas más pequeños, o módulos que pueden desarrollarse y probarse por separado. A esta forma de trabajar se le llama programación modular.
- La principal cualidad de los módulos es que son tan independientes y autosuficientes como sea posible. Además, en general, están diseñados para llevar a cabo una función específica y bien definida, y tienen un punto de entrada y un punto de salida.
- Finalmente combinando programación estructurada y modular es posible resolver la mayoría de los problemas de ingeniería.
- **A lo largo del año iremos programando “Módulos” o “Funciones” que aplican diferentes algoritmos de resolución. Luego, la combinación de estos nos permitirá resolver los problemas planteados.**
- **Obviamente también se aprovecharán los módulos ya elaborados e incluidos de manera nativa en el software.**

```
INPUT  $\underline{v}$ , n  
 $suma = v(1)$   
DOFOR  $i = 2, 1, n$   
 $suma = suma + v(i)$   
ENDDO  
DISPLAY  $suma$ 
```

```
INPUT  $\underline{v}$   
 $n = SIZE(\underline{v})$   
 $suma = v(1)$   
DOFOR  $i = 2, 1, n$   
 $suma = suma + v(i)$   
ENDDO  
DISPLAY  $suma$ 
```

```
FUNCTION  $suma = SUM(\underline{v})$   
 $n = SIZE(\underline{v})$   
 $suma = v(1)$   
DOFOR  $i = 2, 1, n$   
 $suma = suma + v(i)$   
ENDDO  
ENDFUNCTION
```

Modulo o función para realizar la sumatoria de un conjunto de valores

```
INPUT  $\underline{u}$   
 $promedio = SUM(\underline{u}) / SIZE(\underline{u})$   
DISPLAY  $promedio$ 
```

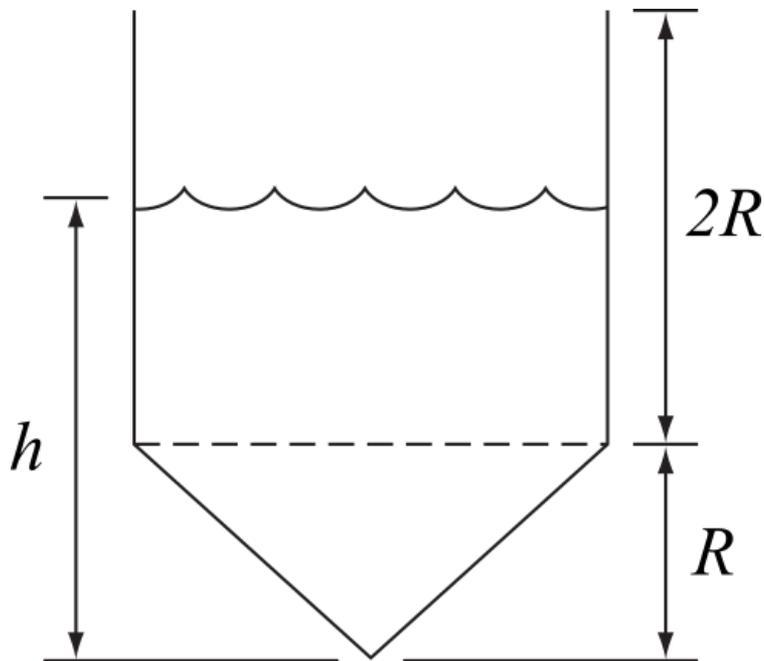
Pseudocódigo para el calculo de un promedio aprovechando los módulos disponibles

```
FUNCTION promedio=PROM(v)  
n=SIZE(n)  
suma = v(1)  
DOFOR i = 2, 1, n  
  suma = suma + v(i)  
ENDDO  
promedio = suma/n  
ENDFUNCTION
```

Modulo o función para calcular el promedio de un conjunto de valores

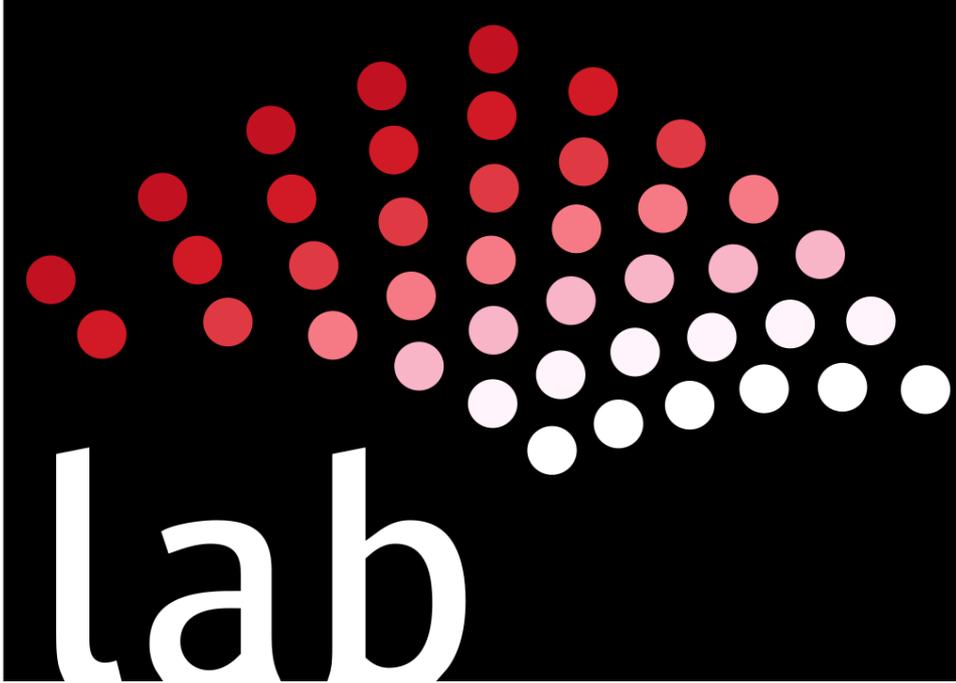
```
FUNCTION [r1 r2]=CUADRATICA(a,b,c)
IF a = 0 THEN
    IF b  $\neq$  0 THEN
        r1 = -c/b; r2 = []
    ELSE
        DISPLAY "No hay solución"
        r1 = []; r2 = []
    ENDIF
ELSE
    discr =  $b^2 - 4 * a * c$ 
    IF discr  $\geq$  0 THEN
        r1 = (-b + Sqrt(discr))/(2 * a)
        r2 = (-b - Sqrt(discr))/(2 * a)
    ELSE
        DISPLAY "Raíces Complejas"
        r1 = []; r2 = []
    ENDIF
ENDIF
ENDFUNCTION
```

Escriba un procedimiento bien estructurado de función para calcular el volumen del tanque como función de los valores dados de  $R$  (*radio del tanque*) y  $h$  (*altura de liquido*). Utilice estructuras de control de decisiones (como If/Then, Elself, Else, End If). Diseñe la función de modo que produzca el volumen en todos los casos y además genere un mensaje de error (“Sobrepasado”) si se rebasa la altura del tanque.



```

IF condición1 THEN
  Bloque1
ELSEIF condición2
  Bloque2
ELSEIF condición3
  Bloque3
ELSE
  Bloque4
ENDIF
  
```



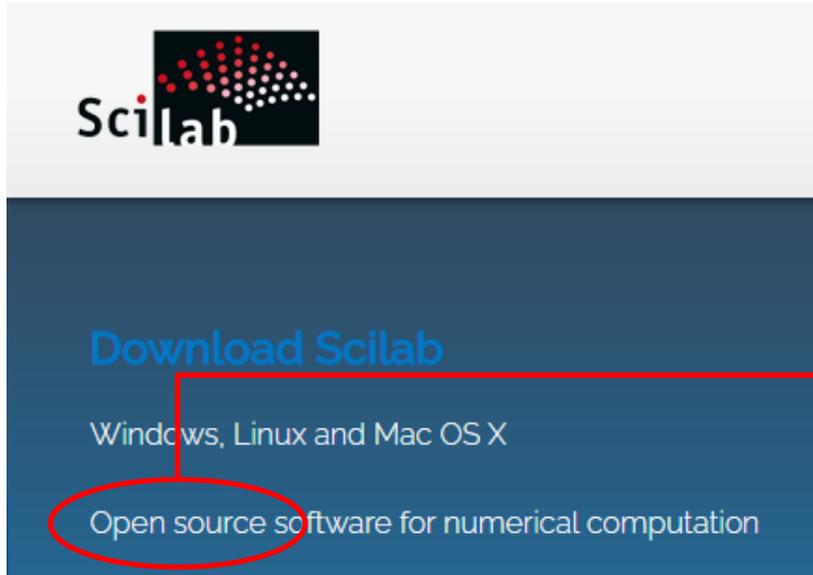
**Sci** **lab**

2023

Prof.: Dr. Juan Ignacio Manassaldi

JTP: Ing. Amalia Rueda

- Scilab es un software para análisis numérico, con un lenguaje de programación de alto nivel para cálculo científico.
- Es desarrollado por Scilab Enterprises, bajo la licencia CeCILL, compatible con la GNU General Public License.
- Las características de Scilab incluyen:
  - Análisis numérico
  - Visualización 2-D y 3-D
  - Optimización
  - Análisis estadístico
  - Diseño y análisis de sistemas dinámicos
  - Procesamiento de señales
  - Interfaces con Fortran, Java, C y C++.
  - Xcos, un editor gráfico para diseñar modelos de sistemas dinámicos.



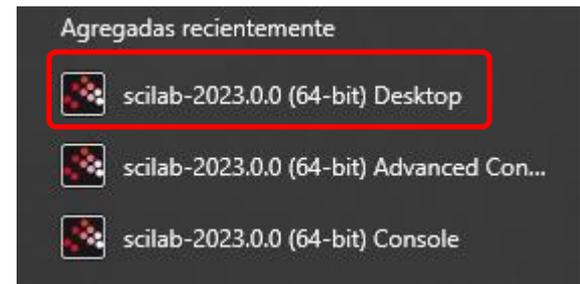
<https://www.scilab.org/>

El código abierto es un modelo de desarrollo de software basado en la colaboración abierta.

**Scilab 2023.0.0**

**Windows 8, 10, 11**

Scilab 2023.0.0 - Windows 64 bits (exe)



*IF condición THEN*  
*Bloque verdadero*  
*ENDIF*

```
if b  $\approx$  0 then  
r1 = -c / b;  
end
```

*IF condición THEN*  
*Bloque verdadero*  
*ELSE*  
*Bloque falso*  
*ENDIF*

```
if a < 0 then  
b = sqrt(abs(a));  
else  
b = sqrt(a);  
end
```

*IF condicion<sub>1</sub> THEN*

*Bloque<sub>1</sub>*

*ELSEIF condicion<sub>2</sub>*

*Bloque<sub>2</sub>*

*ELSEIF condicion<sub>3</sub>*

*Bloque<sub>3</sub>*

*ELSE*

*Bloque<sub>4</sub>*

*ENDIF*

if class == 1 then

x = x + 8;

elseif class < 1 then

x = x - 8;

elseif class < 10 then

x = x - 32;

else

x = x - 64;

end

*SELECT CASE Expresión de prueba*

*CASE Valor<sub>1</sub>*

*Bloque<sub>1</sub>*

*CASE Valor<sub>2</sub>*

*Bloque<sub>2</sub>*

*CASE Valor<sub>3</sub>*

*Bloque<sub>3</sub>*

*CASE ELSE*

*Bloque<sub>4</sub>*

*END SELECT*

```
select a + b
```

```
case 1
```

```
    x = -5;
```

```
case 2
```

```
    x = -5 - (a + b) / 10;
```

```
case 3
```

```
    x = (a + b) / 10;
```

```
else
```

```
    x = 5;
```

```
end
```

*DO*

*Bloque<sub>1</sub>*

*IF condicion EXIT*

*Bloque<sub>2</sub>*

*ENDDO*

```
while (1)
    i = i + 1
    if i >= 10
        break
    end
    j = i
end
```

*DO*

*Bloque<sub>1</sub>*

*IF condicion EXIT*

*Bloque<sub>2</sub>*

*ENDDO*

*while*  $i < 10$

$i = i + 1$

$j = i$

*end*

*DOFOR i = inicio, fin, incremento*  
*Bloque*  
*ENDDO*

```
for i = 1:2:10  
    x = x + i;  
end
```