

## Resolución de Problemas y Algoritmos

### Clase 5



**Dr. Alejandro J. García**

http://cs.uns.edu.ar/~ajg



Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Bahía Blanca - Argentina

### Conceptos de las clases anteriores

- **Compilador / compilación / ejecución**
- **Diagrama sintácticos**
- **Sentencia condicional IF-THEN-ELSE**
- **Condicionales anidados**

```
PROGRAM preguntas;
VAR respuesta:CHAR;
BEGIN
writeln(' ¿Preguntas? (s/n) ');
Readln(respuesta);
IF (respuesta='s') or (respuesta='S')
THEN write(' OK lo escucho...')
ELSE write(' seguimos entonces..')
END.
```

¿Preguntas? (s/n)

S

OK lo escucho...

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 2

### Hardware [Wikipedia]

**Hardware** corresponde a todas las partes tangibles de un sistema informático; sus componentes son: eléctricos, electrónicos, electromecánicos y mecánicos. Son cables, gabinetes, periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente, el soporte lógico que es intangible y es llamado **software**.

**Hardware** es una palabra inglesa (literalmente: partes duras); como no posee una traducción adecuada, fue admitida por la Real Academia Española que lo define como: «*Conjunto de los componentes que integran la parte material de una computadora*».

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 3

### Software [Wikipedia]

Se conoce como **software** al *equipamiento lógico o soporte lógico* de un sistema informático; comprende el conjunto de los componentes **lógicos** necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados **hardware**.

**Software** es una palabra inglesa (literalmente: partes blandas o suaves); como en español no posee una traducción adecuada, fue admitida por la Real Academia Española que lo define como: «*Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora* ».

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 4

### Software [Wikipedia]

Existen varias definiciones similares aceptadas para software, pero probablemente la más formal sea la siguiente: « **Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación** » [Extraído del estándar 729 del [IEEE](#)]

Considerando esta definición, el concepto de software va más allá de los programas de computación; también su documentación, los datos a procesar e incluso la información de usuario forman parte del software: es decir, *abarca todo lo intangible*, todo lo «no físico» relacionado.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 5

### Historia [Wikipedia]

- **John W. Tukey** usó el termino "Software de Computación" (Computer Software) en un artículo de 1958 en el *American Mathematical Monthly*, aparentemente el primer uso del término.
- Mientras trabajaba con **John von Neumann** en los primeros diseños de computadoras, Tukey introdujo la palabra "**bit**" como contracción de "Dígito binario" (por sus siglas en inglés *Binary Digit*).
- *El concepto* de leer diferentes secuencias de instrucciones (**programa**) desde la **memoria** de un dispositivo para controlar los cálculos fue introducido por **Charles Babbage** alrededor de 1833.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 6

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:  
 “**Resolución de Problemas y Algoritmos. Notas de Clase**”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.

### Ciclo de vida del software

El software surge de una **necesidad, problema** o **idea**.  
 Luego se **desarrolla** e **implementa**.  
 Finalmente se **usa** durante un tiempo y de ser necesario se **modifica**. (según la experiencia el 60% del ciclo de vida lo constituye esta etapa)  
 Es **importante** utilizar **metodologías** y **técnicas** que simplifiquen el desarrollo y faciliten el mantenimiento (algunas de ellas vamos a comenzar a ver en RPA)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 7

### El desarrollo de software y algunas materias

- Definición de requerimientos **ADS**
- Análisis del sistema
- Diseño de sistemas. **IPOO – TP – DDS**
- Implementación de programas. **IPOO-ED-TP**
- Pruebas unitarias (programas). **DDS**
- Pruebas de integración.
- Entrega del sistema y Mantenimiento. **RPA**

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 8

### Pautas para un buen estilo de programación

- Es sumamente importante que un algoritmo o programa sea **claro**, y puede ser **entendido rápidamente** por una **persona** que lo lee.
- Para lograr esto hay que tener en cuenta una serie de **pautas** de "**buen estilo**" para la confección de programas:
  1. **Uso de nombres representativos.**
  2. **Indentación (en inglés indent).**
  3. **Comentarios en el código.**  
 {comentarios en Pascal entre llaves}

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 9

### 1. Uso de nombres representativos

```

IF (x1 > x2)
THEN
IF (x1 > w)
THEN z8 := x1
ELSE z8 := w
ELSE IF (x2 > w)
THEN z8 := x2
ELSE z8 := w
    
```

```

IF (num1 > num2)
THEN
IF (num1 > num3)
THEN máximo := num1
ELSE máximo := num3
ELSE IF (num2 > num3)
THEN máximo := num2
ELSE máximo := num3
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 10

### 2. Indentación

```

IF (num1 > num2)
THEN
IF (num1 > num3)
THEN máximo := num1
ELSE máximo := num3
ELSE IF (num2 > num3)
THEN máximo := num2
ELSE máximo := num3
    
```

```

IF (num1 > num2)
THEN
    IF (num1 > num3)
        THEN máximo := num1
        ELSE máximo := num3
    ELSE
        IF (num2 > num3)
            THEN máximo := num2
            ELSE máximo := num3
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 11

### 3. Comentarios en el código

```

IF (num1 > num2)
THEN
IF (num1 > num3)
THEN máximo := num1
ELSE máximo := num3
ELSE
IF (num2 > num3)
THEN máximo := num2
ELSE máximo := num3
    
```

```

{ calcula el máximo entre 3
números: num1, num2 y num3}
IF (num1 > num2)
THEN
    IF (num1 > num3)
        THEN máximo := num1
        ELSE máximo := num3
    ELSE {... caso num1 <= num2...}
        IF (num2 > num3)
            THEN máximo := num2
            ELSE máximo := num3
        {... la variable máximo ahora tiene
el mayor valor de los 3....}
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 12

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:  
 "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.

### Compare



```

IF (x1 > x2)
THEN
IF (x1 > w)
THEN z8 := x1
ELSE z8 := w
ELSE IF (x2 > w)
THEN z8 := x2
ELSE z8 := w
                    
```



```

{ calcula el máximo entre 3
números: num1, num2 y num3}
IF (num1 > num2)
THEN
IF (num1 > num3)
THEN máximo := num1
ELSE máximo := num3
ELSE { ... caso num1 <= num2...}
IF (num2 > num3)
THEN máximo := num2
ELSE máximo := num3
{ ... la variable máximo ahora tiene
el mayor valor de los 3....}
                    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 13

### Cuando escribe en papel también puede usar líneas para agrupar bloques



```

IF (num1 > num2)
THEN
IF (num1 > num3)
THEN máximo := num1
ELSE máximo := num3
ELSE
IF (num2 > num3)
THEN máximo := num2
ELSE máximo := num3
                    
```



```

IF (num1 > num2)
THEN
IF (num1 > num3)
THEN máximo := num1
ELSE máximo := num3
ELSE
IF (num2 > num3)
THEN máximo := num2
ELSE máximo := num3
                    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 14

```

PROGRAM CantDiasMes;
VAR mes, anio, cant_dias: INTEGER;
BEGIN { ... dado mes y año obtiene cantidad de días...}
write(' Ingrese mes (1 a 12) y año: '); readln(mes, anio);
IF (mes<1) OR (mes>12) THEN write(' valor de mes inválido ')
ELSE
begin { ...mes está entre 1 y 12...}
IF (mes = 2) { ... febrero depende si es año bisiesto...}
THEN IF (anio mod 4=0) and (anio mod 100<>0)
or (anio mod 400=0)
THEN cant_dias := 29 {cuando es bisiesto}
ELSE cant_dias := 28 {cuando no es bisiesto}
ELSE IF (mes = 11) OR (mes = 4) OR (mes = 6) OR (mes = 9)
THEN cant_dias := 30 { para nov., abril, junio y sep.}
ELSE cant_dias := 31; { cuando es 1,3,5,7,8,10 o 12 }
writeln('La cantidad de días para mes ', mes, ' es ', cant_dias);
end; { ... fin del ELSE (mes está entre 1 y 12)...}
END.
                    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 15

### Especificación de un algoritmo

- Escriba un algoritmo para obtener exactamente 1 litro de agua en un bidón de 100 litros. Se dispone de una botella de medio litro y de las primitivas vaciar, llenar y trasvasar.
- Escriba otro par obtener obtener exactamente 10 litros de agua en un bidón de 100 litros. Se dispone de una botella de medio litro.

**Algoritmo 2:**  
**Vaciar(bidón)**  
**Repetir 20 veces lo que sigue:**

- Llenar (botellita, canilla)
- Trasvasar (botellita, bidón)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 16

### Especificación de repetición en un algoritmo

- Escriba otro algoritmo para obtener exactamente 80 litros de agua en un bidón de 100 litros. Se dispone de una botella de medio litro.
- Escriba otro algoritmo para obtener exactamente N litros (0<N<100) litros de agua en un bidón de 100 litros. Se dispone de una botella de medio litro.

**Algoritmo 4:**  
**Vaciar(bidón)**  
**Obtener el valor de N**  
**Repetir 2\*N veces lo que sigue:**

- Llenar (botellita, canilla)
- Trasvasar (botellita, bidón)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 17

### Problema propuesto

- Escriba un algoritmo y un programa para calcular “num” elevado a la “pot” (num y pot naturales).
- **Ejemplos:** 2<sup>3</sup>=8 3<sup>2</sup>=9 1<sup>8</sup>=1
- **Solución:** multiplicar “num” “pot” veces

**Algoritmo:**

```

Leer num y pot
Potencia ← 1
Repetir pot veces
    potencia ← potencia * num
Mostrar potencia
                    
```

**Tarea:** Escriba el programa en Pascal.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 18

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente:  
 “Resolución de Problemas y Algoritmos. Notas de Clase”. Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.

### Repetición incondicional en Pascal

Las sentencias de un ciclo **FOR-TO** se ejecutan CERO o más veces dependiendo de *valor\_inicial* y *valor\_final*.

```
FOR V:= valor_inicial TO valor_final
DO 1 sentencia simple
   o compuesta ;
Otra sentencia siguiente;
```

La sentencia se ejecuta un número fijo de veces y luego continua en:

- La **variable V** se suele llamar **variable de control**.
- Al comenzar a **V** se le asigna *valor\_inicial*.
- Luego, **V** es **incrementada automáticamente de a uno** en cada repetición (hasta llegar a *valor\_final*).

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    19

### Repetición incondicional en Pascal

```
FOR V:= valor_inicial TO valor_final
DO sentencia
```

- *valor\_inicial* y *valor\_final* son expresiones cuyo valor debe pertenecer al mismo tipo que **V**.
- La *sentencia* (que puede ser compuesta), se repetirá ( $valor\_final - valor\_inicial + 1$ ) veces.

```
FOR V:= 1 TO 5 DO writeln(V); { 5 veces}
```

- Si *valor\_final* es menor estricto a *valor\_inicial* entonces se repetirá 0 veces.

```
FOR V:= 5 TO 1 DO write(V); {0 veces}
```

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    20

### Repetición incondicional

- Al comenzar a **V** se le asigna el valor inicial y luego, **V** se incrementada automáticamente de a uno hasta llegar al valor final.

FOR V:= 1 TO 100 DO <sentencia>	Aquí <sentencia> se repite 100 veces
FOR V:= 100 TO 199 DO <sentencia>	Aquí <sentencia> se repite 100 veces
FOR V:= -10 TO -1 DO <sentencia>	Aquí <sentencia> se repite 10 veces
FOR V:= 1 TO -2 DO <sentencia>	Aquí <sentencia> se repite 0 veces

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    21

### Problema Propuesto

Escriba un programa para mostrar por pantalla los números de 1 a un tope ingresado y sus potencias de 2.

```
PROGRAM Cuadrados;
{muestra los números de 1 a un tope y sus cuadrados}
VAR n,tope:INTEGER;
BEGIN
write('Ingrese tope: '); readln(tope);
writeln('Potencias de 2');
FOR N:=1 TO tope DO
BEGIN
write(N); write(' ');
write(N*N); writeln;
END;
END.
```

**Ingrese tope: 5**

**Potencias de 2**

1:	1
2:	4
3:	9
4:	16
5:	25

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    22

### Sentencia FOR-TO

```
FOR V:= exp1 TO exp2 DO sentencia
```

- Tanto *exp1* como *exp2* pueden ser valores, variables, o expresiones siempre que sean del mismo tipo ordinal que **V** (**no puede ser tipo real**).
- Al comenzar a **V** se le asigna el valor de evaluar *exp1* y luego, **V** se incrementada automáticamente de a uno hasta llegar al valor de *exp2*.

```
N:=2;
FOR V := 1+1 TO N DO writeln(N);
FOR V := N TO N+N DO writeln(N);
FOR V := N div 2 TO (N+10) - N + 2 DO writeln(N);
```

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    23

### FOR-TO: ejemplo con CHAR

```
PROGRAM letras_y_codigos;
{muestra los códigos ASCII de las mayúsculas}
VAR LETRA: CHAR;
BEGIN
FOR LETRA:= 'A' TO 'Z' DO
BEGIN
WRITE(LETRA, ' = ');
WRITELN(ORD(LETRA));
END;
END.
```

A =	65
B =	66
C =	67
D =	68
...	

Resolución de Problemas y Algoritmos    Dr. Alejandro J. García    24

El uso total o parcial de este material está permitido siempre que se haga mención explícita de su fuente: "Resolución de Problemas y Algoritmos. Notas de Clase". Alejandro J. García. Universidad Nacional del Sur. (c)1998-2012.

**Problema propuesto**

Factorial de un número N se denota con **N!**

y se define:  **$N! = 1 * 2 * 3 * 4 * \dots * N$**

**$0! = 1$**

**Ejemplos:**

**$1! = 1$      $2! = 2$      $3! = 6$      $4! = 24$**

**$5! = 120$**

**$6! = 720$**

**$7! = 5.040$**

**$8! = 40.320$**

**$9! = 362.880$**

**$10! = 3.628.800$**

Escriba un programa para calcular el factorial de un número ingresado por el usuario.

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

25

**Problema propuesto**

**Problema:** Escriba un programa para calcular el Factorial de un número N ingresado por el usuario.

**Solución:** si  $N=0$   $0! = 1$ ; si  $N>0$   $N! = 1 * 2 * 3 * 4 * \dots * N$

**Algoritmo:**

Leer N

factorial  $\leftarrow 1$

factor  $\leftarrow 1$

repetir N veces:

    factorial  $\leftarrow$  factorial x factor

    factor  $\leftarrow$  factor + 1

Mostrar factorial en pantalla

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

26

```

PROGRAM  CalcularFactorial;
        { calcula factorial de un número leído}
VAR  numero, factor, factorial: INTEGER;
BEGIN
  writeln('ingrese número >= 0');
  readln(numero);
  factorial:=1;
  FOR factor:=1 TO numero
    DO factorial:=factorial * factor;
  Writeln(' El factorial de ',numero, ' es ', factorial);
  Readln; {espera que presione enter ....}
END.

```

Resolución de Problemas y Algoritmos

Dr. Alejandro J. García

27