

Resolución de Problemas y Algoritmos

Clase 3



John von Neumann



Dr. Alejandro J. García
<http://cs.uns.edu.ar/~ajg>

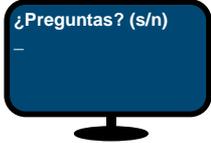


Departamento de Ciencias e Ingeniería de la Computación
 Universidad Nacional del Sur
 Bahía Blanca - Argentina

Conceptos de las clases anteriores

- Algoritmo. Primitiva. Trazo.
- Lenguaje Pascal:
 - Constantes y Variables
 - Tipo de dato (tipos de datos predefinidos)
 - Primitiva de Asignación
 - Primitivas WRITE y WRITELN
 - Primitivas READ y READLN

```
PROGRAM preguntas;
VAR respuesta:CHAR;
BEGIN
writeln(' ¿Preguntas? (s/n) ');
Readln(respuesta);
END.
```



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

Conceptos: Computadora

- Una computadora es un sistema digital con tecnología microelectrónica compuesta por:
 - 1- CPU (Unidad Central de Proceso)
 - 2- Memoria
 - 3- Dispositivos periféricos de Entrada y Salida
 Todo interconectado (por "buses")

Podemos distinguir:

- computadora de propósito general: *PC – notebook – netbook – tablet PC*
- computadora dedicada (a veces empotrada): *celular- rep. MP4 - impresora- video juego- (inyector de combustible de un auto) - placa video*

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

Computadora con arquitectura von Neumann

Una computadora es un sistema digital con tecnología microelectrónica compuesta por:

- 1- CPU (Unidad Central de Proceso)
- 2- Memoria
- 3- Dispositivos de Entrada y Salida

Todo interconectado (por "buses")

CPU

← bus →

Memoria:
programa y datos

Esta arquitectura, y el concepto de programa almacenado en memoria, (ideados en 1945) se le atribuyen al matemático húngaro: John von Neumann

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

Programación de computadoras a bajo nivel






Acme Matrix Red Head Blue Sky

- Cada CPU de una computadora es capaz de ejecutar un único lenguaje llamado lenguaje máquina.
- Cada marca de CPU tiene su propio lenguaje máquina.

¿Tengo que aprender el lenguaje de cada máquina?
Afortunadamente NO (como verá a continuación)

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

Programación de computadoras a alto nivel

- Normalmente hay un compilador para cada marca de CPU.

Compilador Pascal

Compilador Pascal

Compilador Pascal



Programa en Pascal

- Un compilador de Pascal es un programa que traduce los programas escritos en Pascal a instrucciones de lenguaje máquina interpretables por una computadora particular.

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)1998-2010.

Programación de computadoras a alto nivel

- De esta manera, un **mismo programa** puede compilarse y luego puede ejecutarse sobre **distintas computadoras**.

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

Edición / Compilación / Ejecución

Existen en la actualidad aplicaciones que brindan un **entorno de programación**, donde se puede editar, compilar y ejecutar programas en Pascal.

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

Conceptos: compilación

- Un compilador es un **programa** que **traduce un programa** escrito en un **lenguaje de programación** a otro lenguaje de programación, generando un programa equivalente.
- Este proceso de traducción se conoce como **compilación**.
- Un compilador permite traducir el **código fuente** de un programa en **lenguaje de alto nivel**, a otro lenguaje de nivel inferior (típicamente **lenguaje de máquina**).
- De esta manera, un programador puede diseñar un programa en un lenguaje mucho más cercano a como piensa un ser humano, para luego **compilarlo** a un programa más manejable por una computadora.

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

Terminología

Programa Fuente: es el programa escrito en el lenguaje de programación usado (Ej. Pascal).

Programa Ejecutable: Es una secuencia de código que la computadora puede ejecutar **directamente** al ser invocado, (sin necesidad que el compilador esté presente.) Generalmente de extensión EXE o COM.

- La **definición** de un lenguaje es algo **teórico**. A la definición original se lo llama **Pascal Estándar**.
- Numerosas compañías venden la **implementación** de un **compilador** de Pascal para ser usado en diferentes computadoras (esto es algo práctico).
- Estas implementaciones pueden **extender** el estándar, y a veces **modificarlo**.

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

Conceptos: Sintaxis y Diagrama Sintáctico

- La **sintaxis** de un lenguaje es un conjunto de reglas que indica como escribir programas.
- Un **diagrama sintáctico** es una descripción gráfica de la sintaxis de un lenguaje de programación. Permite **describir sin ambigüedad** la sintaxis de un lenguaje de una manera simple y formal.
- Está compuesto por cuatro tipos de elementos.

Ejemplo:

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

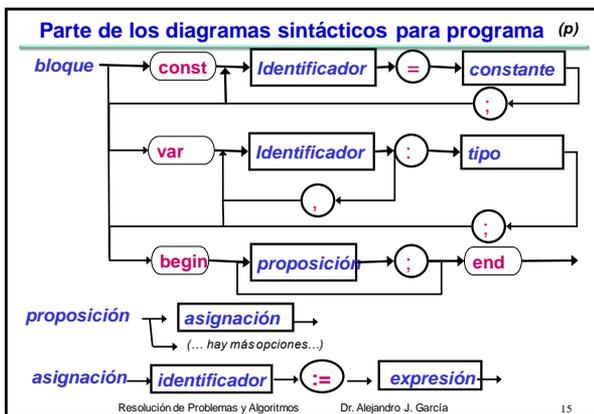
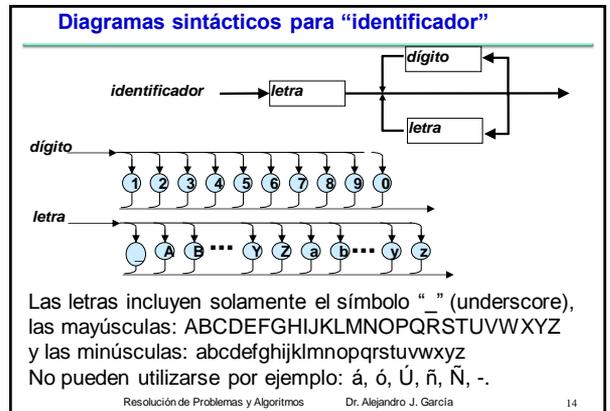
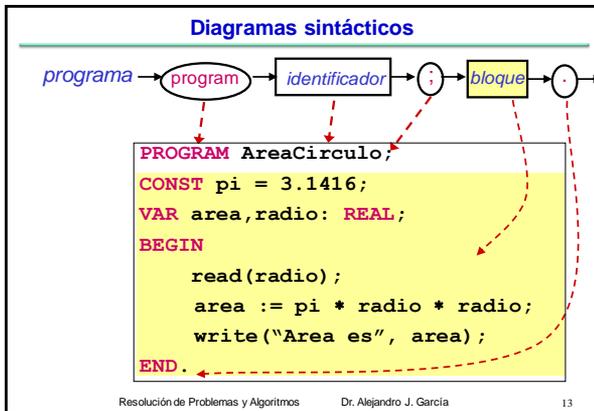
(p) Elementos de un diagrama sintáctico

- Un **nombre** y **flecha** indican el comienzo de un diagrama para la definición de **nombre**.
- Las **figuras "redondeadas"** indican que **texto** se debe **incluir tal cual** como aparece.
- Los **rectángulos** indican que **nombre** está **definido en algún otro diagrama** sintáctico.
- Las **flechas** indican el **orden** de lectura en el diagrama.

Ejemplo:

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)1998-2010.



Problema simple propuesto

Problema: Escriba un programa en Pascal para obtener el valor absoluto de un número.

Solución: Si el número es positivo, el valor absoluto es el mismo número y sino es el número multiplicado por -1.

Algoritmo:
 Leo el número Num
 Si Num > 0
 entonces val_abs es Num
 Sino val_abs es Num * -1
 Muestro val_abs en pantalla

Verificación:
 ejemplos significativos 3, 0 y -3

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

Uso de punto y coma

Para separar sentencias en Pascal se utiliza el símbolo “;” (**punto y coma**).

Observación: puede haber varias sentencias en un mismo renglón y una sentencia puede tener varios renglones.

i := 1; j:=2; k:= 3;
B:= (i>0) and (k = j);

↔

i := 1;
j:=2;
k:= 3;
B:= (i>0)
and (k=j);

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

Sentencias en Pascal

Las sentencias en Pascal pueden ser simples o compuestas. Ejemplos:

Sentencia
o
proposición

{

simple { a:=1

compuesta { BEGIN
PrecioBase := 200;
Iva:= Precio * 0.20;
PrecioFinal:= PrecioBase+ iva;
END

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)1998-2010.

Sentencia Compuesta en Pascal

Una sentencia compuesta comienza con **BEGIN** y termina con **END** y permite definir una secuencia de sentencias como si fuera una única sentencia.

Por ejemplo, la siguiente es una sentencia (compuesta, a su vez, por tres sentencias simples)

```

BEGIN
  PrecioBase := 200;
  Iva:= Precio * 0.20;
  PrecioFinal:= PrecioBase+ iva;
END
    
```

Las **sentencias** que forman una sentencia compuesta se separan una de otra con **punto y coma**

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

Estructura de control condicional (1)

IF expresión lógica

THEN Sentencia 1 (simple o compuesta)

ELSE Sentencia 2 (simple o compuesta)

Ejemplo (valor absoluto de un número):
 Write ('Ingrese...'); read(numero);
IF numero > 0
THEN val_abs := numero
ELSE val_abs := (-1) * numero;
 writeln(' Valor absoluto: ', val_abs);

Obs: no lleva “;” antes del ELSE

- **Sintaxis:** ver el diagrama sintáctico.
- **Semántica:** Si la evaluación de la expresión lógica da verdadero, entonces se ejecuta solamente la sentencia que sigue al “THEN”, Si en cambio la evaluación de la expresión lógica da falso, entonces se ejecuta sólo la sentencia que sigue al “ELSE”

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

Problema: Escriba un programa que lea un caracter (CHAR) y diga si se trata de una letra mayúscula, minúscula, o un dígito.

Solución: de ‘A’ a la ‘Z’ es una mayúscula, de ‘a’ a la ‘z’ es una minúscula y de ‘0’ a ‘9’ un dígito.

Algoritmo:

- leer el caracter
- Si está entre ‘A’ y ‘Z’ entonces es una mayúscula
- Si está entre ‘a’ y ‘z’ entonces es una minúscula
- Si está entre ‘0’ y ‘9’ entonces es un dígito

Verificación:
ejemplos significativos ‘G’, ‘g’, ‘3’, ‘\$’

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

Estructura de control condicional (2)

IF Exp. lógica

THEN

Sent. 1 (simple o compuesta)

```

program leer_char;
var ch: char;
begin
write('Ingrese un caracter:');
readln(ch);
IF (ch >= 'A') and (ch <= 'Z')
  then writeln('mayúscula.');
```

- Este es un caso particular del IF-THEN-ELSE.
- **Semántica:** Si la evaluación de la expresión lógica da verdadero, entonces se ejecuta solamente la sentencia que sigue al “THEN”. Si da falso, sigue con la sentencia siguiente.

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

Programa del problema anterior

```

program leer_char;
var ch: char;
begin
write('Ingrese un caracter:');
readln(ch);
IF (ch >= 'A') and (ch <= 'Z')
  then writeln(ch, ' es una mayúscula.');
```

```

IF (ch >= 'a') and (ch <= 'z')
  then writeln(ch, ' es una minúscula.');
```

```

IF (ch >= '0') and (ch <= '9')
  then writeln(ch, 'es un dígito. ');
end.
```

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

Tarea para el hogar ☺

Problema propuesto:

- Escriba un programa en Pascal que lea un carácter (CHAR) y diga si se trata de una letra mayúscula, minúscula, un dígito, o un símbolo distinto a los anteriores.

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)1998-2010.

• En el problema de calcular la cantidad de litros para pintar un aula, ¿cómo calcular la cantidad de latas de 4 litros a comprar?

```

PROGRAM pintura_aula;
CONST alto = 2.60; puerta = 3.20; ventanas = 2;
      cubrelitro = 8; litroslata = 4;
VAR ancho, largo, a_no_pintar, cant_litros:REAL;
    cant_ventanas, cant_latas:INTEGER;
BEGIN
ancho := 5;
largo := 10; cant_ventanas := 2;
a_no_pintar := 2*puerta + 2 * cant_ventanas ;
a_pintar := 2*(ancho*alto)+2*(largo*alto)-a_no_pintar;
cant_litros:= a_pintar / cubrelitro;
cant_latas:= cant_litros / litroslata;
END.
    
```

Solución:
SI cant_litros dividido cubrelitro da justo
ENTONCES cant_latas es la división
SINO tengo que considerar una lata mas

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 25

Dos soluciones posibles

```

...
IF (cant_litros / litroslata) = TRUNC (cant_litros / litroslata)
THEN cant_latas:= TRUNC(cant_litros / litroslata)
ELSE cant_latas:= TRUNC(cant_litros/ litroslata) +1;
writeln(' Latas a usar: ', cant_latas);
    
```

```

...
cant_latas:=TRUNC(cant_litros/litroslata);
IF cant_latas < (cant_litros / litroslata)
THEN cant_latas:= cant_latas + 1;
writeln(' Latas a usar: ', cant_latas);
    
```

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 26

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)1998-2010.