



Resolución de Problemas y Algoritmos

Clase 2


(أبو عبد الله محمد بن موسى الخوارزمي ابو جعفر)



Dr. Alejandro J. García
[http:// cs.uns.edu.ar /~ajg](http://cs.uns.edu.ar/~ajg)





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



Conceptos de la clase pasada

- Algoritmo
- Traza de un algoritmo
- Primitiva
- Sobre el lenguaje Pascal:
 - Palabras reservadas
 - Constantes y Variables
 - Tipos predefinidos
 - Asignación

¿preguntas?

Más material en la página de la materia
<http://cs.uns.edu.ar/~smd/RPA/>

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

Algoritmo y álgebra: su origen (Wikipedia)

(أبو عبد الله محمد بن موسى الخوارزمي ابو جعفر)

Abu Abdallah Muḥammad ibn Mūsā al-Jwārizmī
 Conocido como **al-Juarismi**, (al-Khwārizmī)

Fue matemático, astrónomo y geógrafo persa musulmán chií, (780 – 850).

Vivió en Bagdad.

La palabra "**Algebra**" viene de **al-jabr**, una de las dos operaciones usadas en su obra: "**Kitab al-jabr wa'l muqabala**".

Allí pretendo enseñar un **álgebra aplicada** a la **resolución de problemas** de la vida cotidiana.



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

Algoritmo y álgebra: su origen (Wikipedia)

Su libro "**Kitab al-jabr wa'l muqabala**": "**Compendio de cálculo por restauración y comparación**", es un manual, destinado a **«hacer más claro lo que era oscuro y [...] facilitar lo que era difícil»** con el **objeto de resolver problemas concretos de cómputo** de herencias, medida de tierra o comercio.

Por ello, **al-Juarismi** es considerado el padre del álgebra y el introductor del sistema de numeración decimal a Persia (luego introducido en Europa en el siglo XII por **Fibonacci**).



1ra. página del libro

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

Expresiones numéricas y lógicas

- Al introducir la primitiva de asignación, se mostró que el lado derecho al símbolo "==" es una **expresión** que da un valor.
- Las expresiones indican ("expresan") como calcular adecuadamente un valor.
- Saber construir correctamente expresiones es muy importante porque:
 - se utilizan de muchas maneras en un algoritmo (no solo en asignaciones)
 - hay expresiones de muchos tipos de valores (no solo numéricos)

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

¿Qué es una expresión?

- Una expresión está formada por "**operadores**" y "**operandos**" (Ej. $3 + 4$)
- Los **operadores** están definidos para un **tipo** específico de **operando** y para una cantidad predeterminada de **operandos**.
- Hay **operadores**:
 - numéricos: $+$, $-$, $*$, etc.
 - lógicos: **y**, **no**, **o**, etc.
 - relacionales: $>$, $<$, $=$, etc.

¿son correctas estas expresiones?

$3 + 1$

$3 +$

$+ 3 1$

$- 4$

$3 + 2 + 1$

$1 + 2 + 3$

$3 ++ 4 1$

$3 + 4 * 2$

$3 * 4 + 2$

$3 > 2 + 1$

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.

Expresiones numéricas (repaso)

Los operadores toman los valores de los operandos y obtienen un nuevo valor. Este valor puede ser el resultado final de la expresión, o un resultado parcial utilizado en la expresión por otro operador.

Ejemplos:

$3 + 4 * 2 - 4$ ¿en que orden se evalúa?

- El orden en que se evalúa la operación depende de la **precedencia de los operadores**.
 $3 + 4 * 2 - 4 = 7$
- Los **paréntesis** permiten cambiar el orden de eval.
 $(3+4) * 2 - 4 = 10$
 $(3+4) * (2 - 4) = -14$

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

Resto de la división entera

•La operación **resto o módulo** será una primitiva:
 $R = X // Q$

representa "R es el resto de dividir X por Q"

Ejemplos $20 // 5=0$ $5 // 2=1$ $8 // 3=2$

Propiedad: Si $R=X//Q$, R está entre 0 y $Q-1$

Nótese también que el resto cicla entre 0 y $Q-1$

$1 // 4=1$ $2 // 4=2$ $3 // 4=3$ $4 // 4=0$
 $5 // 4=1$ $6 // 4=2$ $7 // 4=3$ $8 // 4=0$

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

Operadores y valores

- Operadores numéricos:** (ej. $+$ $-$ $*$ $/$)
Toman números y tienen un número por resultado
- Operadores relacionales:** (ej. $=$ $>$ $<$ \neq \geq \leq)
Relacionan dos datos del mismo tipo y tienen un resultado que es **verdadero** o **falso**.
- Operadores lógicos:** (ej. **y** o **no**)
Toman valores del conjunto { verdadero, falso } y su resultado es un valor verdadero o falso.

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

Expresiones Lógicas

Así como hay infinitos valores numéricos, hay sólo dos valores lógicos: **verdadero** y **falso**

Las **expresiones lógicas** se construyen con los siguientes tres **operadores básicos**:

y (and) conjunción
o (or) disyunción
no (not) negación

A diferencia de los operadores numéricos, **alcanza una tabla** para definir todos los resultados

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

Tablas de verdad

Una tabla de verdad para un operador lógico, muestra explícitamente el **resultado** de **todas** las **operaciones posibles** con ese operador, y por lo tanto define al operador.

Sea A una expresión lógica (esto es, su resultado es verdadero o falso), la tabla de verdad de la negación es la siguiente:

A	no A
verdadero	falso
falso	verdadero

Por ejemplo A podría ser "día = 21" o "día mod 2 = 0"

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

Tabla de Verdad para el "y" (conjunción)

A	B	A y B
verdadero	verdadero	verdadero
verdadero	falso	falso
falso	verdadero	falso
falso	falso	falso

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.

Tabla de Verdad para el "o" (disyunción)

A	B	A o B
verdadero	verdadero	verdadero
verdadero	falso	verdadero
falso	verdadero	verdadero
falso	falso	falso

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

Precedencia en Expresiones Lógicas

El orden de evaluación asociado a los distintos operadores lógicos determina su precedencia. La **precedencia en expresiones lógicas** tiene analogía con la usada en expresiones aritméticas:

(1) no (2) y (3) o	(1) menos unitario (2) producto (3) suma
no A o B y C (no A) o (B y C)	- 3 + 4 x 5 (-3) + (4 x 5)

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

Expresiones Lógicas

- Notemos que el uso de los **paréntesis** ayuda a dar **distintos significados...**

Compare:

(tomo502 o tomo503 o tomo500) y tengo_tarjeta

tomo502 o tomo503 o (tomo500 y tengo_tarjeta)

Calcule el resultado de ambas expresiones con:
 tomo502 = verdadero, tomo503=falso,
 tomo500 =falso, tengo_tarjeta=falso

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

Expresiones Lógicas Equivalentes

- Decimos que dos expresiones lógicas son **equivalentes** si una es verdadera si y solo si la otra también lo es. **Ejemplo:**

(tomo502 o tomo503 o tomo500) y tengo_tarjeta

es equivalente a

(tomo502 y tengo_tarjeta) o (tomo503 y tengo_tarjeta) o (tomo500 y tengo_tarjeta)

¿Por qué es importante poder encontrar expresiones equivalentes?

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

George Boole (1815-1864)

Matemático y filósofo Inglés. Inventor del [álgebra de Boole](#), la base de la [aritmética computacional](#) moderna. Es considerado uno de los fundadores del campo de las [Ciencias de la Computación](#).

En 1854 publicó "An Investigation of the Laws of Thought" en él desarrollaba un sistema de reglas que le permitía expresar, manipular y simplificar, problemas lógicos y filosóficos cuyos argumentos admiten dos estados (verdadero o falso) por procedimientos matemáticos. Se podría decir que es el padre de las operaciones lógicas y gracias a su álgebra hoy en día podemos manipular operaciones lógicas.

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

Expresiones numéricas y lógicas

Ahora si, escriba expresiones para:

- Un número N es mayor a 10
- N es mayor a 10 y menor a 100.
- N tiene a lo sumo 4 dígitos.
- N tiene 4 dígitos (exactamente).
- N tiene dos o cuatro dígitos.
- N es un número impar.
- N es divisible por 7 y divisible por 11 y tiene dos dígitos.

Hay más ejercitación vea el práctico ☺


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.**

El lenguaje de programación Pascal

Pascal fue definido en 1969 por **Niklaus Wirth** (http://en.wikipedia.org/wiki/Pascal_language)

- **Reporte Original:** Jensen - Wirth
 - **Libro de consulta:** "Programación en Pascal" de **Peter Grogono** (1986)



Ambos en Biblioteca Central:
<http://bc.uns.edu.ar> (N. Wirth en 1984)

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

Tipos Predefinidos en Pascal Estándar

Tipo de Dato: define el conjunto de valores posibles que puede tomar una variable, y también define las operaciones que puede usarse.

INTEGER: es un subconjunto de los números enteros
REAL: es un subconjunto de los números reales.
BOOLEAN: es el conjunto {true, false} (valores lógicos)
CHAR: es el conjunto de los 256 caracteres del código ASCII (letras, dígitos, y símbolos)

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

El código ASCII

American Standard Code for Information Interchange

Está formado por 256 símbolos, aquí se muestran algunos:

32	33	!	34	"	35	#	36	\$	37	%	38	&	39	'					
40	(41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0	49	1
50	2	51	3	52	4	53	5	54	6	55	7	56	8	57	9	58	:	59	;
60	<	61	=	62	>	63	?	64	@	65	A	66	B	67	C	68	D	69	E
70	F	71	G	72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X	89	Y
90	Z	91	[92	\	93]	94	^	95	_	96	~	97	a	98	b	99	c
100	d	101	e	102	f	103	g	104	h	105	i	106	j	107	k	108	l	109	m
110	n	111	o	112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	{	123		124	}	125	~	127		128	Ç	129	ü	

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

Operaciones del Tipo de Dato Simple INTEGER

Operador	operación	operandos	resultado
+	suma	Integer	integer
-	resta	Integer	integer
*	producto	Integer	integer
div	división entera	Integer	integer
mod	resto	Integer	integer

Función SQR: devuelve el cuadrado (square) de un entero. **Ejemplo:** SQR(3) = 9.
Constante predefinida MAXINT: Es el máximo valor del tipo INTEGER.

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

Realice una traza y luego pase a la máquina

PROGRAM Ejemplo1;
{Algunos ejemplos para el tipo entero}
VAR N1,N2,N3,N4,N5,N6:INTEGER;
BEGIN
 N1 := 1+ 2000 mod 2;
 N2 := SQR(9);
 N3 := SQR(SQR(3));
 N3 := N3+N2;
 N4 := MAXINT;
 N5 := 1+ N4 ; *{¿qué valor toma N5?}*
 N3 := N6; *{error de programación: N6 no tiene valor}*
END.

Recuerde: Usar una variable sin valor es un error de programación.

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

Tipo de Dato Simple REAL

Tipo REAL : Corresponde a un subconjunto de los números reales. Pascal permite reales con **punto decimal** (ej: 3.5459), o en **notación científica**:
 Ej: $3.5 \cdot 10^{-3} = 0.0035$ en Pascal es **3.5E-3**
 Ej: $1.28 \cdot 10^8 = 128000000$ en Pascal es **1.28E8**

Operador	Operación	Operandos	Resultado
+	suma	real o integer	real (1)
-	resta	real o integer	real (1)
*	mult.	real o integer	real (1)
/	división real	real o integer	real

(1) El resultado es real cuando al menos un operando es real

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.

Algunas funciones predefinidas para REAL

Funciones trigonométricas: SIN, COS y TAN. Dado un valor de un ángulo (en radianes), devuelven su seno, coseno o tangente. Ejemplos: SIN(0) = 0, COS(0) = 1

Función raíz cuadrada (square root) **SQRT**
Ejemplo: SQRT(4) = 2.0

Función de redondeo ROUND: dado un valor real, devuelve el entero más cercano.
Ejemplos: ROUND(2.9) = 3 ROUND(2.3) = 2

Función truncado TRUNC: dado un valor real, devuelve el entero que resulta de eliminar la parte decimal.
Ejemplos: TRUNC (2.9) = 2 TRUNC(2.3) = 2 (p)

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

Realice una traza y luego pase a la máquina

PROGRAM Ejemplo2;
{Algunos ejemplos para el tipo real}
VAR N1,N2:INTEGER;
 R1,R2: REAL;
BEGIN
 R1 := 20 mod 2;
 R2 := MAXINT + 1;
 N1 := TRUNC(2.5);
 N2 := ROUND(2.5);
END.

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

Tipo de Dato Simple BOOLEAN

Este tipo de dato tiene únicamente dos valores asociados, **true** y **false** (corresponden a **verdadero** y **falso**).

Los operadores booleanos **y**, **o** y **no** se representan en Pascal con las palabras reservadas **and**, **or** y **not**.

- Puedo utilizar los símbolos = > y <
- El símbolo ≠ se escribe <>
- El símbolo ≥ se escribe >= o >=

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 27

Realice una traza y luego pase a la máquina

PROGRAM Ejemplo3;
VAR R1: REAL;
 es_par, positivo, mayor_a_maxint:BOOLEAN;
 condicion:BOOLEAN;
BEGIN
 R1 := 23;
 Es_par := (TRUNC(R1) mod 2) <> 1;
 positivo := R1 >= 0;
 Mayor_a_maxint := R1 > MAXINT;
 Condicion:= es_par and es_positivo
 and not mayor_a_maxint;
END.

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 28

Tipo CHAR :

es el conjunto de los 256 símbolos ASCII.

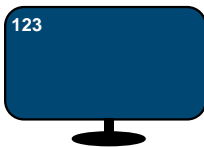
¿Cómo se diferencia entre una variable cuyo identificador es A y el símbolo ASCII A ?

- Para indicar un **valor** de tipo CHAR, se utilizan las comillas simples. Ej: 'a', '?', '+', ' ', etc.
- Por lo tanto, en Pascal: 'A' es una letra, y en cambio A es un identificador.


Función CHR: permite obtener un caracter cualquiera a partir de su código ASCII.
Ejemplos: chr(65) = 'A'; chr(33) = '!'.
Función ORD: dado un caracter cualquiera, devuelve su código ASCII.
Ejemplos: ord('A') = 65, ord('!') = 33. (p)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 29

Primitivas para mostrar en pantalla



```
write(1);
write(2);
write(3);
```



```
writeln(1);
writeln(2);
writeln(3);
```

WRITE: muestra valores en la pantalla
WRITELN: muestra valores en pantalla y baja de línea (LN)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 30

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.

¿Qué puede imprimir write/ln?

- La primitivas write y writeln permiten mostrar por pantalla los siguientes elementos:
 - Valores de los 4 tipos real, integer, char y boolean
ej. Write(1, 2.30, 'A', true);
 - El valor de una variable de alguno de esos 4 tipos
ej. Write(N1, R1, CH, es_par);
 - El valor de evaluar una expresión de esos 4 tipos
ej. Write(2+N1, SQRT(5), CHR(64), true or false);
 - Textos encerrados entre comillas simples
ej. Write(' este es un texto a mostrar en pantalla');
- Ambas primitivas permiten recibir una cantidad arbitraria de elementos separados por comas (,)

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 31

Equivalencias y diferencias

WRITE: muestra valores en la pantalla
WRITELN: muestra valores en pantalla y baja de línea (LN)

WRITE(1);
WRITELN;

WRITELN(1);

mismo efecto

WRITE('ho');
WRITE('la ');
WRITE('chel!');
WRITELN;

WRITELN('hola che !');

m. efecto

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 32

```
PROGRAM pintura_aula;
CONST alto = 2.60;
cubrelitro=8; puerta=3.20;
VAR ancho, largo, a_no_pintar: REAL;
    cant_litros , a_pintar :REAL; cant_ventanas : INTEGER;
BEGIN
ancho := 5; largo:= 10; cant_ventanas :=2;
writeln(' Para un aula de ', largo, 'x', ancho,' con ',
cant_ventanas,' ventanas');
a_no_pintar:= 2*puerta + 2 * cant_ventanas ;
a_pintar := 2*(ancho*alto)+2*(largo*alto)-a_no_pintar;
Writeln( 'voy a pintar: ', a_pintar,' m2');
cant_litros:= a_pintar / cubrelitro;
Write(' Litros a usar: '); writeln(cant_litros);
END.
```

Para un aula de 10 x 5 con 2 ventanas voy a pintar 67.60 m2 Litros a usar: 8.45

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 33

Primitiva para la lectura o ingreso de datos

READ: obtiene (lee) valores ingresados por teclado
READLN: (read-line) idem a read pero espera por un ENTER

•Ambas tienen como argumentos una o varias variables que pueden ser integer, real, o char.
Ej: VAR N:integer; R: real; CH: char;

READ(N);
READLN(CH);
READ(R,N,CH)

READ(CH);
READLN;

READLN(CH);

mismo efecto

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 34

```
PROGRAM pintura_aula;
CONST alto = 3; puerta = 3.20;
ventanas = 2; cubrelitro = 8;
VAR
ancho, largo, a_no_pintar: REAL;
cant_litros , a_pintar :REAL;
cant_ventanas:INTEGER;
BEGIN
writeln('Ingrese ancho del aula');
readln(ancho);
writeln('Ingrese largo del aula'); readln(largo);
writeln(' ¿Cuántas ventanas?'); readln(cant_ventanas);
a_no_pintar:= 2*puerta + 2 * cant_ventanas ;
a_pintar := 2*(ancho*alto)+2*(largo*alto)-a_no_pintar;
cant_litros:= a_pintar / cubrelitro;
writeln('Litros a usar =', cant_litros);
END.
```

Ingrese ancho del aula
5
Ingrese largo del aula
10
¿Cuántas ventanas?
2
Litros a usar = 8.45

Resolución de Problemas y Algoritmos Dr. Alejandro J. García 35


Herramientas a disposición de los alumnos

Herramientas brindadas por la cátedra:

- clase teórica
- clase práctica
- ejercicios en los prácticos
- horarios de laboratorio
- evaluación en máquina y parciales (obligatorio)

Extras

- material en línea (<http://cs.uns.edu.ar/~smd/RPA/>)
- horarios de consulta individual
- Tutores (tutoria@cs.uns.edu.ar)



Resolución de Problemas y Algoritmos Dr. Alejandro J. García 36

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.