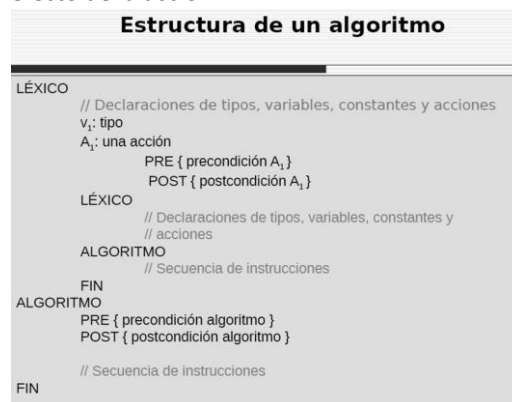


## • Tema 1: Léxico de un algoritmo

### 1. *Léxico de un algoritmo*

- Léxico de un algoritmo: **INFORMACIONES + ACCIONES**
- Informaciones = Variables
- Magnitudes que caracterizan un proceso algorítmico
- **CONSTRUIR UN ALGORITMO CONSISTE EN ELEGIR UN CONJUNTO DE INFORMACIONES Y OTRO DE ACCIONES, Y A CONTINUACIÓN DECIDIR EL MODO DE ORGANIZAR LAS ACCIONES EN EL TIEMPO PARA OBTENER EL RESULTADO DESEADO POR ACUMULACIÓN DE SUS EFECTOS.**
- Necesitamos una notación: **notación algorítmica.**
- La notación algorítmica fija la forma de:
  - Describir las acciones.
  - Describir las informaciones.
  - Organizar las acciones en el tiempo.
  - Incluye acciones elementales.
- **Algoritmo:**
  - Léxico: informaciones u objetos y acciones
  - Control: ordenar en el tiempo cómo actúan las acciones sobre los objetos
- **Abstracción:**
  - Mecanismo fundamental para dominar la complejidad cuando programamos. “Eliminar detalles innecesarios y considerar lo esencial” . El léxico fija el nivel de abstracción.
- **Construcción algoritmos:**
  - Fijar el léxico
  - Organizar las acciones en el tiempo mediante: secuenciación, análisis de casos e iteración (recursión)
- **Dada la especificación de un problema hay que:**
  - Elegir y nombrar las informaciones
  - Asociar un tipo a cada información
  - Elegir y nombrar las acciones
  - Asociar una precondition y una postcondición a cada acción
- Tipo de DATO: dominio de valores y acciones que son posibles realizar sobre esos valores
- Precondición: requerimiento de la acción
- Postcondición: efecto de la acción



### 2. *Tipos de Datos primitivos*

Un tipo de datos especifica un DOMINIO de valores y el conjunto de OPERACIONES que son aplicables a ese dominio.

- Nuestra notación incluye los tipos de Datos: Entero, Real, Booleano, Carácter, Intervalos de enteros, reales y carácter. Así como mecanismos para definir nuevos tipos de datos.
- Ejemplos:
  - total: Entero;
  - i, j: [1,100];
  - letra: Carácter;
  - esúltimo: Booleano
- Para cada tipo es preciso conocer:
  - Dominio de los valores
  - Operaciones definidas
  - Sintaxis de los literales
  - Sintaxis de las expresiones
- **Enteros:** cualquier valor entero positivo o negativo válido
- **Reales:** cualquier valor numérico real positivo o negativo válido. Utilizaremos el símbolo '.' (punto) para separar la parte entera de la parte decimal.
- **Booleanos:** los dos valores lógicos, Verdadero y Falso
- **Caracteres:** el dominio de este tipo está formado por los caracteres de un código válido y un literal se denota como un carácter encerrado entre apóstrofes.

Tipo de dato	Ejemplos de literales
Entero	0, 352, -342, 20050
Real	4.22, -23.44, 341.015
Booleano	Falso, Verdadero
Carácter	'A', 'a', '\$', '1', '+'

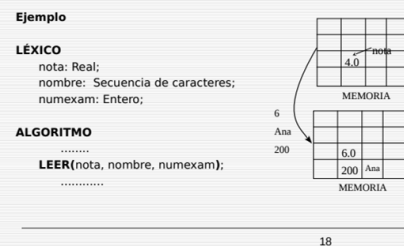
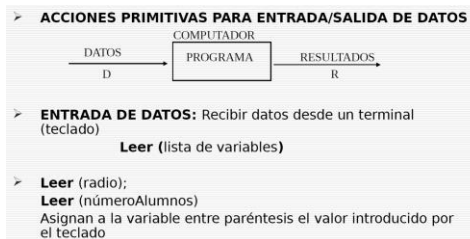
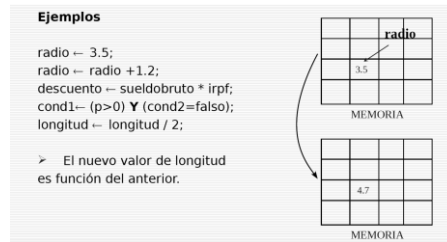
Tipo de dato	Operaciones
Entero	- (Entero $\rightarrow$ Entero)
	+, -, *, DIV, MOD (Entero x Entero $\rightarrow$ Entero)
	/ (Entero x Entero $\rightarrow$ Real)
	<, >, =, ≤, ≥, ≠ (Entero x Entero $\rightarrow$ Booleano)
	Predecesor, Sucesor (Entero $\rightarrow$ Entero)
Real	- (Real $\rightarrow$ Real)
	+, -, *, / (Real x Real $\rightarrow$ Real)
	<, >, =, ≤, ≥, ≠ (Real x Real $\rightarrow$ Booleano)

Tipo de dato	Operaciones
Booleano	Y, O (Booleano x Booleano $\rightarrow$ Booleano)
	NO (Booleano $\rightarrow$ Booleano)
Carácter	Car (Entero $\rightarrow$ Carácter)
	Ord (Carácter $\rightarrow$ Entero)
	<, >, =, ≤, ≥, ≠ (Carácter x Carácter $\rightarrow$ Booleano)
	Predecesor, Sucesor (Carácter $\rightarrow$ Carácter)

### 3. Acciones primitivas

- La acción de asignación: la asignación es la acción primitiva que caracteriza a los lenguajes imperativos
- Sintaxis: <nombre de la variable>  $\leftarrow$  <expresión>

- Semántica: Acción elemental de asignar a la variable cuyo nombre aparece a la izquierda del símbolo  $\leftarrow$  el resultado de evaluar la expresión de la derecha.  
La acción:  
númeroAlumnos  $\leftarrow$  200;  
Asigna a la variable númeroAlumnos el valor 200.

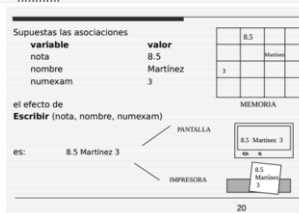


### 1.3 Acciones primitivas

- **SALIDA DE DATOS:** enviar al terminal de salida los valores obtenidos como resultado de evaluar una lista de expresiones  
**Escribir** (lista de expresiones)

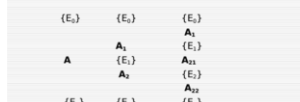
**LÉXICO**  
nota: Real;  
nombre: Secuencia de caracteres;  
numexam: Entero;

**ALGORITMO**  
.....  
**Escribir** (nota, nombre, numexam);  
.....



### 1.4 Organización de las acciones

**Composición secuencial:**  
Introducción de estados intermedios para reducir la complejidad.  
Para ello descomponemos el problema inicial en subproblemas más simples que se pueden resolver de forma independiente

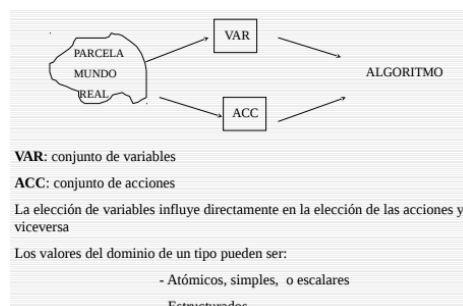


## 4. Organización de las acciones: análisis de casos

- Técnica de descomposición

- Se basa en la partición del dominio de datos en subdominios (casos). Cada subproblema es la restricción del problema inicial al del subdominio considerado
- La descomposición puede estar guiada por la estructura de los datos o de los resultados
- La postcondición de cada subproblema debe cumplir la postcondición del problema inicial y la unión de las precondiciones de los subproblemas debe cubrir la precondición del problema inicial
- **Enunciado:** Dados dos números enteros calcular el mayor
- **Especificación:**
  - $x, y, z$ : entero
  - Precondición:  $\{x = Y = Y\}$
  - Postcondición:  $\{z = \text{Max}(X, Y)\}$
- **Lectura de la especificación:** Dados tres enteros  $x, y, z$ , tal que  $x$  contiene un valor  $X$ , e  $y$  un valor  $Y$ , después de la acción máximo obtenemos en  $z$  el máximo de los valores  $X$  e  $Y$ .
- **Análisis:** existen 2 posibilidades. A)  $x \geq y$  el máximo es  $x$ ; B)  $x < y$  el máximo es  $y$ ;
- La composición secuencial no nos da la posibilidad de tomar decisiones en función de los datos. NECESITAMOS UNA NUEVA COMPOSICIÓN: **composición alternativa, o composición condicional.**
- **Problema:** dados tres enteros diferentes, ordénense de menor a mayor
- **Especificación:** A)  $a, b, c, p, s, t$ : entero. B) Precondición  $\{(a=X \wedge b=Y \wedge c=Z) \wedge (X \neq Y \neq Z \neq X)\}$ . C) Postcondición  $\{(p, s, t) \in \text{perm}(X, Y, Z) \wedge p < s < t\}$

## 5. Tipos de datos no primitivos



- Las notaciones algorítmicas incluyen mecanismos para definir tipos estructurados. Los constructores de tipos más usuales son: tablas, registro (o producto de tipos) y secuencias.
- **Producto de tipos o registro:** Sus valores son una enumeración o agregación de otros tipos ya definidos.
- **Dominio:**  $n$ -tuplas de los tipos constituyentes
- **Definición:**  $\text{nombre\_del\_tipo} = \text{TIPO} < a1: T1; a2: T2; \dots aN: TN >$
- $a_x$ : denota el nombre de cada uno de los campos o elementos del registro
- $T1, T2, \dots TN$ : deben ser tipos ya existentes
- **Cardinalidad:**  $Ct = Ct1 * Ct2 * \dots * CtN$