

# Tema 5

## Datos Estructurados



FACYT

Facultad Experimental de Ciencias y Tecnología

---

Prof. José Henríquez



FACYT

Facultad Experimental de Ciencias y Tecnología

# Datos Estructurados

---

Es una colección de dato que puede ser caracterizado por su organización y las operaciones que se definen en ellos. Los tipos de datos más frecuentes utilizados en los diferentes lenguajes de programación son: definidos por el programador, estáticos y dinámicos. En esta sección sólo nos enfocaremos en los dos primeros.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Definidos por el Usuario

---

- Intervalo
  - $\text{Var} = \text{min} \dots \text{max}$
- Las operaciones que se pueden realizar sobre una variable de tipo Intervalo son iguales que las del tipo a partir del cual está definido.
- Ejemplo :
  - $\text{Nota} = 0 \dots 20$
- Sobre la variable nota se pueden aplicar operaciones definidas para los enteros



FACYT

Facultad Experimental de Ciencias y Tecnología

# Definidos por el Usuario

- Enumerado
  - $\text{Var} = (\text{ident1}, \text{ident2}, \dots, \text{identn})$
- Una variable  $x$  de tipo Enumerado  $T$  sólo puede tomar valores que se encuentren dentro de la lista de identificadores que define a  $T$ .
- Los únicos operadores asociados al tipo Enumerado son el operador de asignación y los operadores de comparación.
- Ejemplo :
  - $\text{Meses} = (\text{'enero'}, \text{'febrero'}, \dots, \text{'diciembre'})$



FACYT

Facultad Experimental de Ciencias y Tecnología

# Estructurado Estático

---

- Homogéneo
  - Vectores Unidimensionales
  - Cadenas
  - Vectores Bidimensionales
- Heterogéneos
  - Registros



FACYT

Facultad Experimental de Ciencias y Tecnología

# Arreglos

- Estructuras de datos conformada por una sucesión de celdas, que permite almacenar en la memoria principal del computador un conjunto finito de elementos (hay un número máximo conocido) que tienen el mismo tipo de dato (son homogéneos).
- Para hacer referencia a cualquiera de las celdas del arreglo es necesario el nombre del arreglo y el valor de uno de los elementos perteneciente al conjunto de **índices** asignado, lo que permite tener acceso aleatorio.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Características básicas de los Arreglos

- **Homogéneo:** los elementos del arreglo son todos del **mismo tipo de dato**
- **Ordenado:** cada elemento del arreglo puede ser identificado por el índice que le corresponde. El índice no es más que un entero que pertenece a un intervalo finito y determina la posición de cada elemento dentro del arreglo.
- **Acceso Secuencial o Directo:** El acceso a cada elemento del arreglo se realiza recorriendo los anteriores según el orden en que están almacenados o de manera directa (operación selectora), indicando el valor del índice del elemento requerido.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Características básicas de los Arreglos

- **Sus elementos son tratados como variables simples:** Una vez seleccionado algún elemento del arreglo este puede ser utilizado en acciones de asignación, como parte de expresiones o como parámetros al llamar a acciones o funciones como si se tratara de una variable del tipo de dato simple declarado para todos los elementos del arreglo.
- **Uso de índice:** El índice es un valor de tipo entero (número entero o carácter con un código entero equivalente) que determina la posición de cada elemento en el arreglo. La cantidad de índices determina las dimensiones del arreglo. Así un arreglo unidimensional tiene un índice, un arreglo bidimensional tiene dos índices y uno n-dimensional tiene n índices





FACYT

Facultad Experimental de Ciencias y Tecnología

# Características básicas de los Arreglos

La sintaxis general para un arreglo de  $r$  dimensiones ( $r \geq 1$ ) es:

- Sintaxis:
- Identificador: arreglo  $[m1..n1, m2..n2, \dots, mr..nr]$  de tipo\_de\_dato

## *Clasificación de los Arreglos:*

- Unidimensionales: también conocidos como vectores. Son arreglos de una dimensión, es decir, sólo utiliza un índice para referenciar a cada uno de los elementos.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Características básicas de los Arreglos

- Bidimensionales: son arreglos de dos dimensiones y también son conocidos como matrices. Para este tipo de arreglos se necesita especificar dos subíndices para identificar cada elemento del arreglo.
- Multidimensionales: un arreglo puede ser definido de tres, cuatro, cinco hasta  $n$  dimensiones. Para este caso, se requiere que los valores de los  $n$  subíndices puedan ser identificados a fin de referenciar un elemento individual del arreglo.

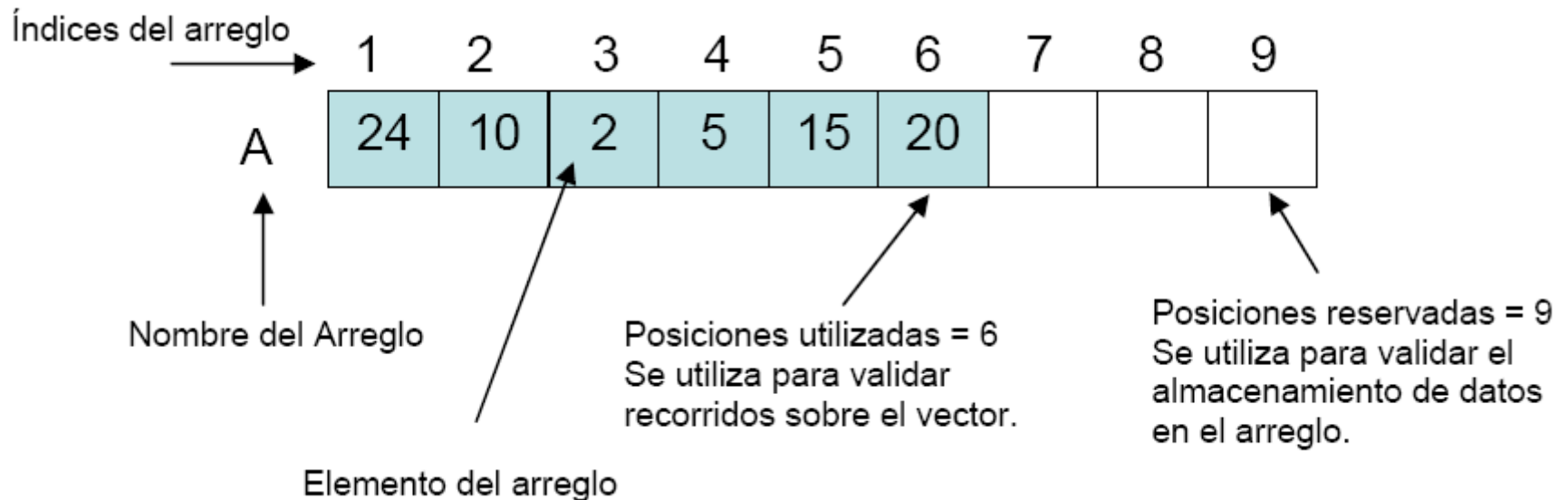


FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores

- Tipos de Datos Estructurados (TDE) donde todos sus elementos pertenecen al mismo tipo y existe una correspondencia uno a uno de cada elemento con un subconjunto de los enteros (índices).



# Vectores

- Cabe destacar que el índice para acceder un arreglo puede ser una constante, variable o expresión , por ejemplo:
  - $A[5]$  acceso al quinto elemento del arreglo, con un índice tipo constante.
  - $A[i]$  accedemos al  $i$ -ésimo elemento del arreglo, con un índice tipo variable. Este tipo de acceso es válido siempre que  $(1 \leq i \leq \text{posiciones reservadas})$
  - $A[(i*2)+1]$  en este caso tenemos un acceso a un elemento del arreglo con un índice tipo expresión. Al igual que el caso anterior este tipo de acceso es válido siempre que  $(1 \leq (i*2) + 1 \leq \text{posiciones reservadas})$ .



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores

Sintaxis para declarar un arreglo unidimensional:

- *Declaración Directa:*

Ejemplo: Persona: arreglo [1..30] de Cadena

- *Declaración Indirecta:*

Ejemplo:

Tipo

Vector\_cad : arreglo [1..30] de Cadena

Variables

Profesores, obreros, administrativos: Vector\_cad



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores

## Ejercicios:

- Declare un arreglo llamado números que almacene 10 enteros máximo
- Declare un arreglo llamado letras que almacene 200 caracteres máximo
- Declare un arreglo llamado Calificaciones que almacene 1000 notas.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Lectura

**Primer Caso:** número de elementos conocidos

Leer (m) //Cantidad de elementos a almacenar  
en el arreglo

Para  $i \leftarrow 1$  [incrementado] hasta m Hacer

Leer (A(i))

Fpara



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Lectura

**Segundo Caso:** se desconoce el número de elementos

Leer(valor\_max)

$m \leftarrow 1$  //contador de elementos que se ingresarán

cent  $\leftarrow$  's'

Mientras (cent = 's'  $\wedge$   $m \leq \text{valor\_max}$ ) Hacer

Leer (A(m) )

Escribir ('Desea ingresar más datos? s / n')

Leer (cent)

$m \leftarrow m + 1$

Fmientras

$m \leftarrow m - 1$  //Al salir del ciclo m representa el número de posiciones utilizadas dentro del vector





FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Escritura

---

La escritura de vectores se representa de un modo similar:

Para  $i \leftarrow 1$  [incrementado] hasta  $m$

    Escribir ( $A(i)$ )

Fpara



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Asignación

- Para asignar valores a todos elementos de un vector, se debe recurrir a estructuras repetitivas.
- Ejemplo: crear un arreglo con los primeros 20 números pares

Para  $i \leftarrow 1$  [incrementado] hasta 10

$$A(i) \leftarrow i * 2$$

Fpara



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores

## Ejercicios:

- Elabore un algoritmo que lea un arreglo de enteros de 50 elementos y luego imprima aquellas posiciones donde existen elementos negativos.
- Dado un arreglo de caracteres y una letra, determinar si un elemento dado pertenece al arreglo e indicar el número de veces que aparezca.
- Se tiene un vector con 20 elementos, elabore un algoritmo que permita saber si un elemento X se encuentra en ese vector y en que lugar.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Operaciones

- **Operación Constructora:** Permite asociarle al nombre de un arreglo un valor estructurado de las mismas dimensiones del arreglo, y con componentes del mismo tipo. Esto sólo se puede hacer en la declaración.

- Ejemplos:

A : Arreglo[1..5] de Entero = {10,15,20,25,30};

A : Arreglo[1..2] de Cadena = {"Primero", "Segundo"};



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Operaciones

---

- **Operación Selectora:** permite hacer referencia a un elemento individual del arreglo, mediante el índice unívoco del elemento. El índice es evaluado, por lo que puede ser una constante, variable o expresión.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Operaciones

**N, i: entero**

**A : Arreglo[1..5] de Entero**

**B : Arreglo[1..10] de Entero**

**N <- 10**

**i <- 3**

**A[1] <- 100      // se le asigna 100 al elemento de índice 1**

**A[i] <- 30      // se le asigna 30 al elemento de índice i=3**

**A[i\*2-1] <- 1      // se le asigna 1 al elemento i\*2-1 (5) del arreglo**

**Para i <- 1 hasta N en 1 hacer    // inicializa el arreglo con la letra x**

**B[i] <- 'x'**

**FPara**

**A[1] <- 3**

**A[2] <- Factorial(A[1])**

**A[3] <- A[1] + A[2] + 1**

**A[ A[1] ] <- 8**



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Operaciones

- **Recorrido Secuencial:** Esta operación se realiza cuando se utiliza una estructura de control iterativa para tratar todos y cada uno de los elementos del arreglo de acuerdo al orden en que se almacenan. El tratamiento es el mismo para todo el arreglo, y se aplica tanto cuando se desea leer el arreglo, buscar un elemento en un arreglo, listar todo el contenido del mismo, y muchos otros.
- El recorrido puede ser de manera ascendente (el más utilizado), partiendo del límite inferior hasta el superior e incrementando en uno; o de manera descendente, partiendo del límite superior hasta el inferior y decrementando en uno.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Operaciones

---

- La estructura de control **Para** es la más conveniente para realizar el recorrido, indicando solo el límite inferior y superior, pues el incremento en uno del índice es automático. Sin embargo, si existiera alguna condición adicional o si se necesita usar una condición lógica para finalizar el recorrido, no basta con llegar al límite superior del índice, hay que cambiar de la estructura iterativa Para, y utilizar un **Mientras** o un **Repetir**.





FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Operaciones

---

Ejemplo: Copia de un arreglo a otro

Const N <- 250

Tipo Arreglo Numeros [1..N] de Entero

A, B: Numeros    #declara dos arreglos del mismo tipo

// llena un arreglo pasado como parámetro de limite superior N

Para i <- 1 hasta N en 1 hacer // lectura de los valores del arreglo

Escribir ("suministre el valor para el elemento" + i + "del arreglo")

Leer(A[i])

FPara;



FACYT

Facultad Experimental de Ciencias y Tecnología

# Vectores - Operaciones

// acción que copia dos arreglos de un mismo tipo

Para i <- 1 hasta N hacer // lectura del arreglo

B[i] ← A[i]

Fpara;

// acción que imprime un arreglo pasado como parámetro de límite superior N

Escribir("A continuación se escriben los valores almacenados en el arreglo")

Para i <- 1 hasta N en 1 hacer // se escriben los valores del arreglo

Escribir(B[i], " -- ")

FPara;



FACYT

Facultad Experimental de Ciencias y Tecnología

# Actualización de Vectores

---

- **Añadir Elemento:** es la operación de añadir un nuevo elemento al final del vector. La única condición necesaria para esta operación consistirá en la comprobación de espacio de memoria suficiente para el nuevo elemento.

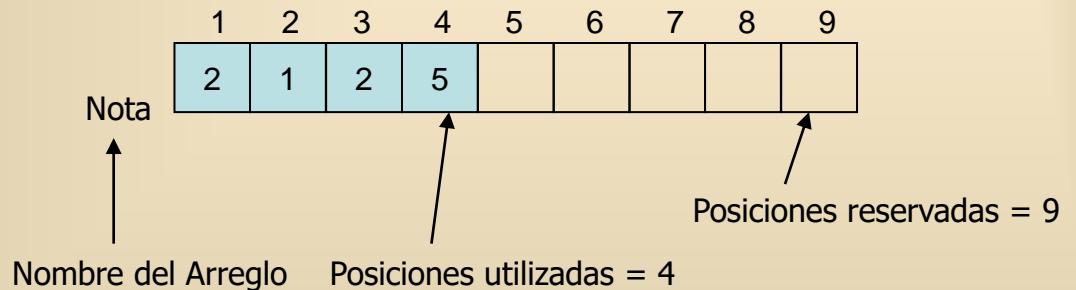


FACYT

Facultad Experimental de Ciencias y Tecnología

# Actualización de Vectores

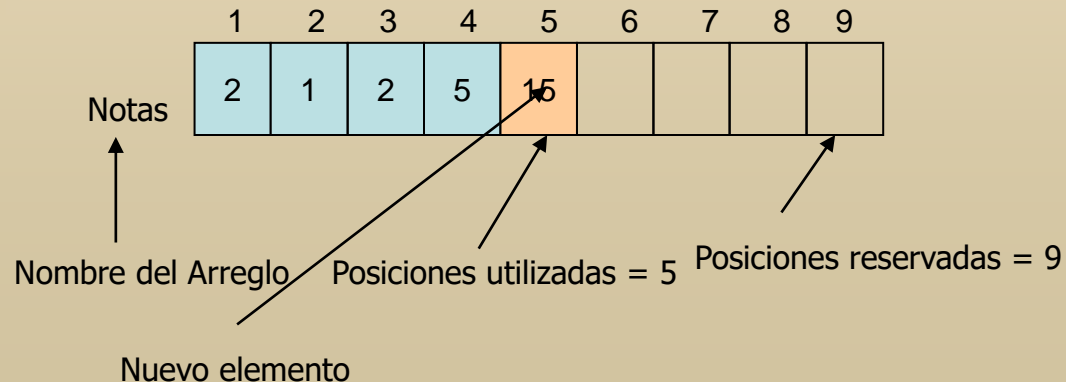
## ■ Añadir Elemento:



Para añadir un nuevo elemento deberíamos:

Nota  $(4+1) \leftarrow 15$  // 15 es el nuevo elemento a añadir en el vector

Y quedaría





FACYT

Facultad Experimental de Ciencias y Tecnología

# Actualización de Vectores

---

- **Insertar un elemento a un vector:** esta operación consiste en introducir dicho elemento en el interior del vector en una posición intermedia. En este caso se necesita un desplazamiento previo hacia abajo para insertar el nuevo elemento en su posición. El bloque algoritmo que resuelve esto es:



FACYT

Facultad Experimental de Ciencias y Tecnología

# Actualización de Vectores

//Donde n es el número de posiciones  
utilizadas y k la posición donde se desea  
insertar el nuevo elemento

Para  $i \leftarrow n$  [Decrementando] hasta k  
 $B(i+1) \leftarrow B(i)$

Fpara

$B(k) \leftarrow p$  //p es el Nuevo valor

$n \leftarrow n+1$  // Incremento de las posiciones utilizadas



FACYT

Facultad Experimental de Ciencias y Tecnología

# Actualización de Vectores

---

- **Eliminar un elemento de un vector:** consiste en eliminar un elemento en una posición intermedia dentro del vector. Lo que provocaría un movimiento hacía arriba de los elementos inferiores a la posición de interés, para reorganizar el vector. El conjunto de acciones que resuelven eso es:



FACYT

Facultad Experimental de Ciencias y Tecnología

# Actualización de Vectores

---

Para  $i \leftarrow [\text{incrementando}]$  hasta  $n-1$

$B(i) \leftarrow B(i+1)$

Fpara

$n \leftarrow n-1$  //Actualizar posiciones utilizadas





FACYT

Facultad Experimental de Ciencias y Tecnología

# Actualización de Vectores

---

- **Ordenamiento de un vector**
- Proceso de organizar datos en algún orden o secuencia específica, tal como creciente o decreciente, o alfabéticamente para datos de tipo carácter. Existen varios tipos de métodos de ordenamiento, la elección de un determinado algoritmo depende del tamaño del vector a clasificar, el tipo de dato y la cantidad de memoria.
- Los métodos más populares son: El de intercambio, Inserción, Selección, Shell, Quick Sort.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Actualización de Vectores

---

- **Búsqueda en un Vector**
- Operación de encontrar la posición de un elemento entre un conjunto de elementos. Existen diferentes algoritmos de búsqueda. Elegir un algoritmo va a depender de la forma en que se encuentren organizados los datos. Los principales tipos de búsqueda son: Secuencial o lineal, Binaria, Hash.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Strings (Cadenas)

- **String** (cadena de caracteres): es una sucesión de caracteres que se encuentran delimitados por doble comilla según el lenguaje de programación, por ejemplo, "Ciudad de Caracas".
- Acotación: Un dato que sea de tipo string no podrá almacenar más de una cadena de caracteres, pero sí podríamos descomponerlo en caracteres o substring más pequeños.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Strings (Cadenas)

- La **Longitud** de una cadena es el numero de caracteres que la componen.
- La cadena que no contiene ningún carácter se denomina cadena vacía o nula, y es de longitud cero.
- Ejemplos:
  - Cad1 = "" diferente de Cad2="" "



FACYT

Facultad Experimental de Ciencias y Tecnología

# Strings (Cadenas)

---

- Longitud de la cadena
- Comparación
- Concatenación
- Extracción de Sub-cadenas
- Búsqueda de Información.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

- Para que un vector de caracteres pueda ser considerado como una cadena de caracteres, el último de los elementos útiles del vector debe ser el carácter nulo (código ASCII 0). Según esto, si se quiere declarar una cadena formada por N caracteres, deberá declararse un vector con  $N + 1$  elementos de tipo carácter. Por ejemplo, la declaración arreglo cadena de caracter [1..5], reserva suficiente espacio en memoria para almacenar una cadena de 4 caracteres, como la palabra "tres":

Nombre del arreglo



num

t	r	e	s	\0
---	---	---	---	----

1

2

3

4

5



FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

- En pseudoformal el tipo de dato cadena (string) será considerado una secuencia de caracteres, sin límite predefinido, es decir, de longitud variable. Las cadenas de caracteres se tratan como un tipo de dato estándar pero estructurado (arreglo de caracteres).
- En el ejemplo anterior la declaración e inicialización de la variable sería:

<i><b>Pseudoformal</b></i>	<i><b>Lenguaje C</b></i>
<i>var cadena num <math>\leftarrow</math> "tres"</i>	<i>var char num [5] = "tres"</i>



FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

- Las operaciones básicas entre cadenas son: asignación, concatenación y comparación.
- El **operador de asignación** ( $\leftarrow$ ) nos permite asignar una cadena a otra, o una cadena constante a una variable. Es importante recordar que una cadena constante es una secuencia de caracteres delimitados por comillas, como "palabra"; a diferencia de una constante de tipo carácter (char) la cual va delimitada por comillas simples, como 'x'.
- *nombre*  $\leftarrow$  "Leonardo Miguel"
- *nombre2*  $\leftarrow$  *nombre*





FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

- *El **operador de concatenación (+)** se utiliza para unir varias cadenas.*
- *Ejemplo de concatenación:*
  - *Cad1 ← "este es un ejemplo"*
  - *Cad2 ← "de concatenación"*
  - *Cad ← Cad1 + Cad2*
  - *Escribir(Cad)*
- *Se visualizaría: este es un ejemplo de concatenación*



FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

- Las **comparaciones de las cadenas** de caracteres se hacen según el orden de los caracteres en el código ASCII y con los operadores de relación.

*Ejemplo:*

- ***merced*** < ***mercurio***      *porque e < u*
- *perro* < *zorro*      *porque p < z*
- ***señor*** < ***señora***      *tienen el mismo prefijo y la más corta es menor*
- ***ANA*** < *ana*      *las mayúsculas tienen menor valor en ASCII (A < a)*



FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

- Para conocer la longitud de una cadena, es decir, conocer su número de caracteres se utiliza la **función longitud**.

*Ejemplo:*

- *Cad1 ← "este es un ejemplo"*
- *Escribir("El nro. de caracteres de la cadena es: "*  
*Longitud(Cad1))*
- *Se visualizaría: El nro. de caracteres de la cadena es:*  
*18*
- ***Nota: Si la cadena es vacía, su longitud es cero***



FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

- La **función posición** nos permite determinar si una cadena está contenida en otra. En este caso, indicará la posición donde comienza la cadena buscada en la cadena fuente (la primera aparición). Si la cadena no existe el resultado será 0

*Ejemplo:*

- *cad1 ← "este **es** un ejemplo de datos **estructurados**"*
- *subc1 ← "es"*
- *subc2 ← "os"*
- *escribir("La palabra <" subc1 "> está en la posición: " posicion(cad1,subc1)*
- *escribir("La palabra <" subc2 "> está en la posición: " posicion(cad1,subc2)*
- *Se visualizaría: La palabra <es> está en la posición: 1*  
*La palabra <os> está en la posición: 26*



FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

---

- Por último, tenemos la función **subcadena**, la cual retorna una subcadena de la cadena original formada por todos los caracteres a partir de una posición dada. Además si se le indica la longitud de la nueva subcadena, sólo retornará los primeros caracteres a partir de la posición indicada.



FACYT

Facultad Experimental de Ciencias y Tecnología

# Cadenas de Carácter

- *Ejemplo:*
- *cad* ← "algunos ejemplos de cadenas"
- *// retorna la cadena a partir de la posición 9 de longitud 8 caracteres*
- *subc1* ← *subcadena(cad, 9)* *subc2* ← *subcadena(cad, 9, 8)*
- *escribir("Nueva subcadena: " subc1)*
- *escribir("Tiene la palabra: " subc2)*
- *Se visualizaría: Nueva subcadena: ejemplos de cadenas*  
*Tiene la palabra: ejemplos*