



# PRÁCTICA N°2

## Tablas, vectores y matrices. Listas.

### 1.- TABLAS Y MATRICES.

Nos interesará saber como se declara una variable como matriz, vector o tabla, y además veremos como introducir la información en las mismas, como consultarla y como mostrarla en pantalla. Para ello vamos a usar dos instrucciones:

Utilizaremos:

**vector=Table[valor\_por\_defecto, {j,n}]**

para definir vectores n-dimensionales. Donde *valor\_por\_defecto* será el valor salvo que se diga lo contrario que toma el vector en todas sus coordenadas. Y:

**matriz=Table[valor\_por\_defecto, {j\_1,n\_1}, {j\_2,n\_2}]**

para definir tablas o matrices  $n_1 \times n_2$ . Donde *valor\_por\_defecto* será el valor salvo que se diga lo contrario que toma la matriz en todas sus coordenadas.

Para acceder o modificar la información contenida en un vector o matriz nos referiremos a sus coordenadas utilizando la siguiente fórmula:

1.- **vector[[i]]** para referirnos a la coordenada i-ésima.

2.- **matrix[[i,j]]** para referirnos a la posición (i,j) de la matriz.

*Ejemplo 1:*

*In[1]:=*

```
vector=Table[0,{i,4}];  
vector[[1]]=A;  
vector[[2]]=B;  
vector[[3]]=C;  
vector[[4]]=D;  
vector
```

*Out[1]=*

```
{A, B, C, D}
```

Aquí definimos un vector 4-dimensional, y asignamos a cada coordenada una letra.

Utilizaremos:

**MatrixForm[lista]**

para que la información se muestre como una matriz y no como una lista de vectores.

Y utilizaremos:

**TableForm[lista]**

Para que la información se muestre como una tabla y no como una lista de vectores.

*Ejemplo 2:*

Si introducimos:

*In[1]:=*

**matriz=Table[0,{i,3},{j,3}]**

*Out[1]=*

**{{0,0,0},{0,0,0},{0,0,0}}**

definimos una matriz 3x3 que contiene 0 en todas sus coordenadas. El *Matemática* la muestra como una lista de vectores, si queremos que aparezca como una matriz, escribimos:

*In[2]:=*

**matriz=Table[0,{i,3},{j,3}];**

**MatrixForm[matriz]**

*Out[2]=*

si queremos que se muestre como una tabla escribiremos:

*In[3]:=*

**TableForm[matriz]**

*Out[3]=*

0	0	0
0	0	0
0	0	0

para terminar, escribimos:

*In[4]:=*

**matriz=Table[0,{i,3},{j,3}];**

**matriz={{1,2,3},{4,5,6},{7,8,9}};**

**MatrixForm[matriz]**

**TableForm[matriz]**

*Out[4]=*

Out[5]=

1	2	3
4	5	6
7	8	9

y así comprobamos claramente como interpreta Matemática las listas de vectores si queremos representarlas como matrices o tablas.

### Ejercicio 1:

- Crear un vector cuyos coeficientes sean los números menores que 50 y múltiplos de 7.
- Crear una tabla  $2 \times 4$  cuya primera fila sean los números pares menores de 10 y la segunda los cuadrados de números de la primera fila.

### Solución:

## 2.- LISTAS: REPRESENTACIÓN DE CONJUNTOS.

Nos limitaremos a conjuntos o listas con finitos elementos, es decir de la forma:  $\{a_1, a_2, \dots, a_n\}$ . Una lista de elementos cualesquiera la representaremos:

Las listas son objetos muy generales que representan colecciones de objetos, en particular, recordemos que, la lista  $\{a, b, c\}$  representa un vector y la lista  $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$  representa una matriz.

En la siguiente tabla podemos ver las operaciones más habituales que podemos hacer para manejar una lista:

<b>Funciones para usar listas</b>	
<b>lista[[i]]</b>	El i-ésimo elemento de la lista.
<b>First[lista]</b>	Primer elemento de la lista
<b>Last[lista]</b>	Último elemento de la lista
<b>Length[lista]</b>	Longitud de la lista.
<b>Append[lista, elemento]</b>	No añade el elemento a la lista, pero devuelve una nueva lista resultado de añadir el elemento al final de la lista.
<b>AppendTo[lista,elemento]</b>	Añade el elemento a la lista (al final de la lista).
<b>Prepend[lista,elemento]</b>	No añade el elemento a la lista, pero devuelve una nueva lista resultado de añadir el elemento al principio de la lista.
<b>PrependTo[lista,elemento]</b>	Añade el elemento a la lista (al principio de la lista).
<b>Join[lista1,lista2,...]</b>	Devuelve una nueva lista resultado de unir lista1, lista 2,...
<b>Insert[lista,elemento,n]</b>	Devuelve una nueva lista resultado de añadir el elemento en la posición n de la lista.
<b>Delete[lista,n]</b>	Devuelve una nueva lista resultado de eliminar el elemento de la posición n de la lista.

*Ejemplo 3:*

```

lista={1,2,3,5,4,6,7}
Last[lista]
First[lista]
Length[lista]
AppendTo[lista,8]
PrependTo[lista,0]
lista>Delete[lista,6]
lista=Insert[lista,4,5]
lista

```

Ejercicio 2:

Utilizando las funciones del cuadro anterior realizar el apartado (a) del ejercicio 1.

**Solución:**

### 3.- OTRAS FUNCIONES NECESARIAS.

Además de los conceptos ya reseñados necesitaremos conocer algunas otras funciones más.

La función **StringJoin** se utiliza para concatenar cadenas alfanuméricas, se utilizará de la siguiente forma:

**StringJoin["s1", "s2", ... ]**

dará como resultado la cadena unión de todos los 'si'. Por ejemplo,

**StringJoin["el", "coche", "rojo" ]**

dará como salida: 'elcocherojo'.

La función **ToString[variable ]** convierte en una cadena alfanumérica el valor de **variable**.

La función **Print[expr1, expr2, ...]** imprime en pantalla en una nueva línea expr1, expr2, ...

La función **ToExpression["cadena"]** hace lo contrario que la función **ToString**.