

Arrays nativos en C++

C++ tiene arrays nativos heredados del C que se pueden utilizar sin definir clases. Además permite una fácil utilización de la gestión dinámica de memoria mediante los operadores *new* y *delete*. Vamos a verlos para los casos de vectores y matrices.

Vectores

Vamos a definir un vector *a* de dimensión *n* y rellenarlo con 5.

```
double *a = new double [n];  
for (int i=0; i<n ; i++)  
    a[i]=5.;
```

Cuando hemos acabado de utilizar el vector liberamos la memoria con

```
delete [a];
```

Matrices

Vamos a definir una matriz de *n* filas y *m* columnas. Definimos un vector de punteros cuyos elementos son vectores que corresponden a las filas de la matriz

```
int ** mx;  
mx = new int* [n];  
for (int i=0; i<n; i++) mx[i] = new int[m];
```

Tenemos que las filas son los *mx[i]* y las columnas de la fila *mx[i]* son *mx[i][j]*. Se empieza a contar por 0. Tenemos que la matriz definida es

$$\begin{array}{cccc} mx[0][0] & mx[0][1] & \cdots & mx[0][m] \\ mx[1][0] & mx[1][1] & \cdots & mx[1][m] \\ \vdots & \vdots & \vdots & \vdots \\ mx[n][0] & mx[n][1] & \cdots & mx[n][m] \end{array}$$

Para liberar la memoria después de utilizarla se emplea el comando

```
delete [] mx;
```