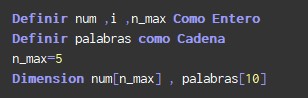
-Vectores/arreglos/Arrays unidimensionales---> Es un tipo de dato que almacena una secuencia de datos del mismo tipo. Los elementos de un vector se almacenan en zonas contiguas de memoria y se puede acceder a ellos de manera directa mediante un índice o posición.

*En este ejemplo hemos definido ”num” como entero por lo que el array num[n\_max] podrá almacenar datos de tipo entero y su dimensión es n\_max(5) , es decir , se podrán almacenar 5 elementos.*

**

*Y también hemos definido “palabras” como cadena por lo que el array palabras[10] podrá almacenar 10 elementos de tipo cadena.*

*para rellenar una posición del array:*

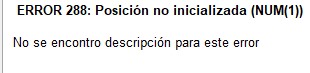
**

*para mostrar el contenido de la posición de un array:*

* *

*¿Qué pasa con el resto de posiciones sin inicializar?*

*Pues resumiendo , en esas posiciones no hay inicializado elemento alguno.*

**

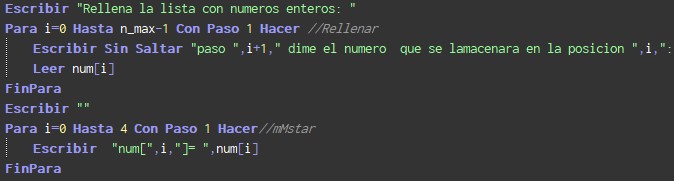
*¿Qué pasa si tratamos de escribir una posición que no ha sido inicializada ?*

**

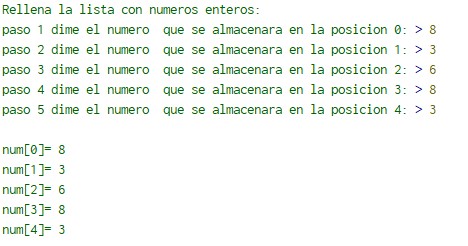
*¿Qué pasa si almacenamos un elemento en una posición que no está dentro de la dimensión?*

**

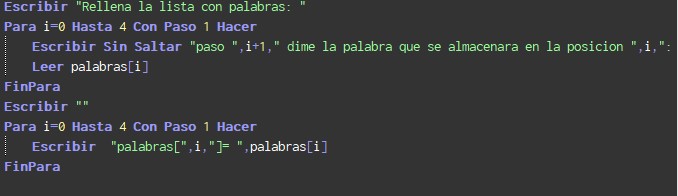
*La mejor manera de manejar un array es mediante el bucle “para”, ya sea para rellenarlo o para mostrar su contenido:*

**

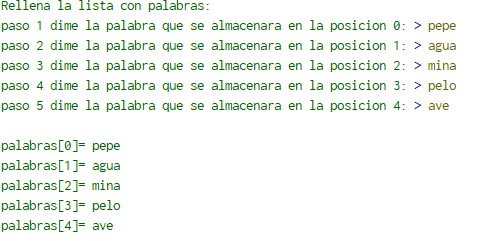
*Como la posición de los arrays comienza en ‘0’ para que se almacenen 5 elementos el bucle “para” deberá ir hasta “n\_max - 1”.*

**

*En ambos casos la dimensión en la cantidad máxima de datos que se podrán almacenar en cada array(reserva memoria para ellos) pero esto no quiere decir que se use todos las posiciones.*

**

*Aquí por ejemplo la cantidad máxima de elementos que se puede almacenar en palabras es 10 (palabras[10]) ,pero nosotros almacenaremos 5*

**

*-Arrays multidimensionales*--->Estructuras de datos que almacenan los valores en más de una dimensión. Es como un contenedor que guardará más valores para cada posición, es decir, como si los elementos del array fueran a su vez otros arrays.

* bidimensional:son tablas de valores en la cual son estructuras de datos que agrupan muchos datos del mismo tipo, en donde cada elemento se puede trabajar individualmente y se puede referenciar con un mismo nombre.



* tridimensional:utiliza tres subíndices. Los dos primeros son como una matriz, pero la tercera dimensión representa páginas u hojas de elementos.



*donde ‘i’ seria las “páginas” y ‘j’ y ‘k‘ la “matriz”*

*(se podría clasificar de la forma que tu quieras pero es mejor tener un orden debido a su complejidad )*

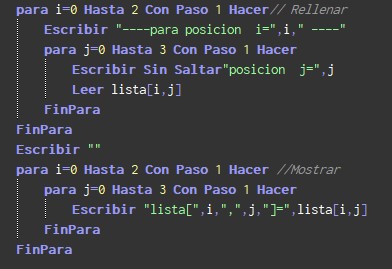
*También se puede interpretar como uno contenedores [i] en que dentro de cada uno de ellos hay otros contenedores[j] que a su vez tienen dentro [k] contenedores.*

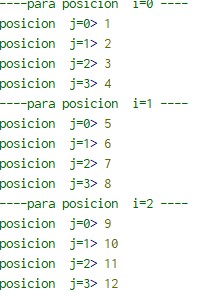
* mayor a 3 dimensiones:vendría a seguir la misma lógica que la tridimensional contenedores que almacenan más contenedores.

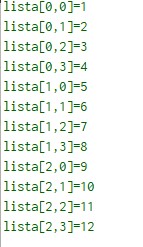
El número de elementos que habrá almacenados dentro del array será la multiplicación de sus dimensiones.

Para rellenar o mostrar los arrays se hace uso de bucles “para” anidados:

(ejemplo con bidimensional)





**

*Se necesitan x iteradores diferentes dependiendo de cuantas dimensiones estemos usando.*

*En este caso como es bidimensional usamos dos iteradores (i y j).*

*Añadiremos bucles “para” anidados dependiendo de lo que necesitemos.*