



## 2. Estructuras de datos

Los tipos de datos que hemos visto hasta ahora son **tipos simples**: `int`, `float`, `double`... Un tipo de datos simple implica que una variable de dicho tipo puede contener un único dato que, además, no puede estar formado por otros datos. Es decir, un número, una letra o un byte, por mencionar algunos ejemplos.

Existen, sin embargo, otros tipos de datos llamados **estructurados**. Una variable de un tipo estructurado es, en realidad, una **colección de datos** de cierto tipo (en general, el mismo para todos los elementos de la colección) a la que puede accederse mediante un identificador único.



Figura 4.1. Tipo de dato simple y tipo de dato estructurado

Los tipos estructurados suelen tener una estructura definida con ciertas reglas y operaciones que permiten trabajar con su contenido de manera más sencilla que si tuviéramos que manejar cada dato por separado.

Lógicamente, una de las propiedades más importantes de un dato de tipo estructurado es su tamaño. Es decir, la cantidad de datos simples que puede contener. En función de si esa cantidad es fija o variable, podemos hablar de estructuras de datos **estáticas** o **dinámicas**.



### IMPORTANTE

En esta unidad veremos con todo detalle cómo trabajar con estructuras de datos estáticas. Las estructuras dinámicas las estudiaremos más a fondo en siguientes unidades.

### 2.1. Estructuras estáticas

Las estructuras estáticas se llaman así porque tienen un tamaño fijo. Es decir, el espacio que ocupan en la memoria no se puede alterar durante su ejecución. Esto no quiere decir que todo el espacio de la estructura deba necesariamente rellenarse con datos, ya que puede haber posiciones vacías. Por tanto, no hay una cantidad mínima de datos que deba contener la estructura, ya que puede incluso estar vacía, pero sí que hay una cantidad máxima que no puede sobrepasarse.

Una estructura estática se representa en la memoria en posiciones contiguas, de tal forma que es muy sencillo saltar de una posición a la siguiente, o incluso acceder directamente a una posición concreta dentro de la estructura.

Ejemplos de estructuras estáticas son los vectores (Arrays) y las matrices (vectores de dos dimensiones).

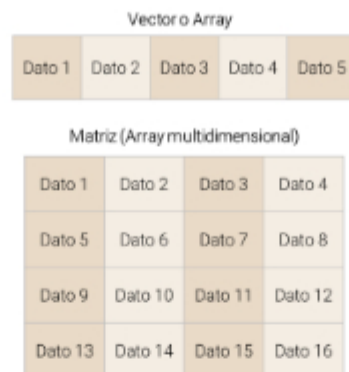


Figura 4.2. Vectores y matrices

En la figura anterior vemos un vector de 5 posiciones y una matriz de 16 posiciones organizadas en 4 filas y 4 columnas (vector bidimensional).

## 2.2. Estructuras dinámicas

Las estructuras dinámicas son aquellas para las que no se determina un tamaño máximo predefinido durante el desarrollo del programa, de manera que pueden crecer o decrecer durante la ejecución de la aplicación.

El tamaño de una estructura dinámica solo está limitado por las reglas del lenguaje, la memoria disponible en el ordenador y la cantidad de esta que se puede manejar usando un único identificador.

Ejemplos habituales de estructuras dinámicas presentes en muchos lenguajes son las pilas, las colas y los árboles. En estas estructuras, en cualquier momento puede incorporarse un dato nuevo: en el caso de la cola, al final; en el caso de la pila, al principio; y en el caso del árbol, como una hoja nueva al final de cualquiera de sus ramas.

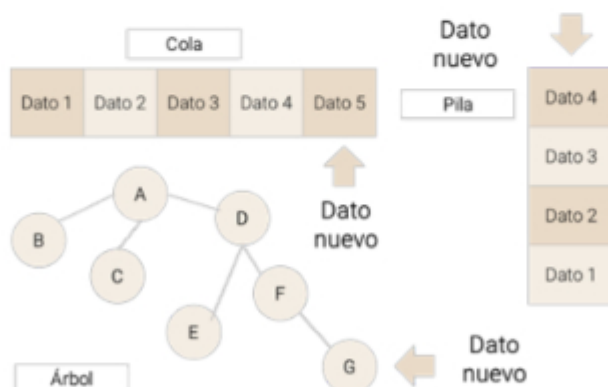


Figura 4.3. Estructuras dinámicas de datos: colas, pilas y árboles

