

# Creación y uso de matrices:

Rust tiene varios tipos de datos compuestos que se pueden utilizar para agrupar varios valores en un solo tipo.

En otro módulo, hemos revisado el tipo de dato **tuple**, que resulta útil cuando se quieren combinar tipos diferentes en un solo valor. En esta sección, se explorará el tipo de dato **array**.

Una matriz es una **colección de objetos del mismo tipo**, que se almacenan secuencialmente en memoria. **La longitud o tamaño de una matriz es igual al número de elementos que esta contiene**. El tamaño de una matriz se puede especificar en el código o calcularse mediante el compilador.

Definición de una matriz:

Una matriz se puede definir de dos maneras:

- Una lista de valores separados por comas, donde no se especifica la longitud.
- El valor inicial, seguido de punto y coma y, después, la longitud de la matriz.

En ambos casos, el contenido se pone entre corchetes [].

Rust

 Copiar

```
// Declare array, initialize all values, compiler infers length = 7
let days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];

// Declare array, initialize all values to 0, length = 5
let bytes = [0; 5];
```

## Lectura de la firma de la matriz:

En tiempo de compilación, la firma de una matriz se define como `[T; size]`:

- **T**: Es el tipo de dato de todos los elementos de la matriz.
- **Size**: Es un entero no negativo que representa la longitud de la matriz.

La firma revela dos características importantes sobre las matrices:

- Todos los elementos de una matriz tienen el mismo tipo de dato. El tipo de dato nunca cambia.
- El tamaño de una matriz es fijo. La longitud nunca cambia.

Solo hay una cosa de una matriz que puede cambiar con el tiempo: los valores de los elementos que contiene. El tipo de dato sigue siendo constante y el número de elementos (longitud) permanece invariable. Solo los valores pueden cambiar.

## Indexación en una matriz;

Los elementos de una matriz se enumeran implícitamente desde 0. Usamos la indexación para acceder a los elementos de una matriz con la expresión `<array>[<index>]`. Por ejemplo, `my_array[0]` accede al elemento en el índice 0 de la variable `my_array`. La expresión devuelve el valor del elemento de matriz en esa ubicación de índice.

Echemos un vistazo a una matriz denominada `days` con siete elementos:

```
// Days of the week
let days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];

// Get the first day of the week
let first = days[0];

// Get the second day of the week
let second = days[1];
```

Para acceder a los elementos de la matriz `days`, usamos un índice que va de 0 a la longitud de la matriz: 1 o 6. El primer elemento en el índice 0 tiene el valor "Sunday". El séptimo elemento del índice 6 tiene el valor "Saturday".

Para asignar un valor a la variable `first`, usamos la expresión `days[0]` a fin de obtener el valor del primer elemento de la matriz `days`, que es Sunday. Para la variable `second`, usamos una expresión `days[1]` similar a fin de obtener el valor del segundo elemento de la matriz `days`, que es Monday.


## Búsqueda de valores de índice fuera de los límites:

Si intentamos acceder a un elemento de nuestra matriz con un índice que no está en el intervalo permitido, el compilador devuelve un error. Una expresión como `days[7]` está fuera de los límites porque nuestra matriz solo tiene siete elementos. El intervalo de índice válido es de 0 a 6.

Cualquier índice que sea igual o mayor que la longitud de la matriz está fuera de los límites. Cualquier índice que sea un número negativo también está fuera de los límites.

En el código siguiente se muestra el error del compilador fuera de los límites:


Rust

 Copiar

```
// Get the seventh day of the week using the wrong index - should be 6
let seventh = days[7];
```

Salida del error:

Resultados

 Copiar

```
error: this operation will panic at runtime
--> src/main.rs:19:42
19 |         let seventh = days[7];
    |                        ^^^^^^^ index out of bounds: the length is 7 but the index is 7
```

Dado que la longitud de la matriz se conoce en tiempo de compilación, Rust hace imposible compilar cualquier programa que intente acceder a la matriz con un índice fuera de sus límites.