

# Tipos de Datos mutables e inmutables

Valery Triana

April 4, 2024

## 1 Java

### 1.1 Introducción

A continuación se presentan los tipos de datos primitivos en Java y sus características.

Java	
Tipo	Mutable
String	X
byte	X
short	X
int	X
long	X
float	X
double	X
ArrayList	X
char	X

### 1.2 Código de ejemplo

Listing 1: Ejemplo de código en Java

```
import java.util.ArrayList;

public class MutableExample {
    public static void main(String[] args) {
        // Creamos un ArrayList mutable de cadenas
        ArrayList<String> mutableList = new ArrayList<>();

        // A adimos elementos al ArrayList
        mutableList.add("Hola");
        mutableList.add("Mundo");
    }
}
```

```

mutableList.add("!");

// Modificamos el primer elemento
mutableList.set(0, "Saludos");

// Eliminamos el segundo elemento
mutableList.remove(1);

// Imprimimos el ArrayList modificado
System.out.println("ArrayList-modificado:-" + mutableList);
}

```

En este ejemplo, la complejidad temporal es  $O(n)$ , donde  $n$  es la cantidad de elementos de la lista. Esta complejidad se debe al método `remove`, el cual, al retirar un elemento de la lista, debe reacomodar los que quedan.

## 2 Golang

### 2.1 Introducción

A continuación se presentan los tipos de datos primitivos en Golang y sus características.

golang	
Tipo	Mutable
Slices	X
Maps	X
Arrays	
String	

### 2.2 Código de ejemplo

Listing 2: Ejemplo de código en Golang

```

package main

import "fmt"

func main() {
    // Creamos un slice mutable
    mutableSlice := []int{1, 2, 3}

    // Modificamos el slice
    mutableSlice[0] = 10
}

```

```

mutableSlice = append(mutableSlice , 4)

fmt.Println(" Slice mutable modificado:" , mutableSlice)
}

```

En este ejemplo, la complejidad temporal es  $O(1)$ . Esto se debe a que crear un slice, modificar una posición específica y añadir un elemento al final del slice requieren un tiempo constante.

## 3 Python

### 3.1 Introducción

A continuación se presentan los tipos de datos primitivos en Python y sus características.

python	
Tipo	Mutable
Listas	X
Diccionarios	X
sets	X
String	
Tuples	
Frozens	

### 3.2 Código de ejemplo

Listing 3: Ejemplo de código en Python

```

# Creamos una lista mutable
mutable_list = [1, 2, 3]

# Modificamos la lista
mutable_list[0] = 10
mutable_list.append(4)

```

En este ejemplo, la complejidad temporal es  $O(1)$ , ya que tanto la creación de una lista, la modificación de un elemento en una posición dada y la adición de un elemento al final toman un tiempo constante.

## 4 C++

### 4.1 Introducción

A continuación se presentan los tipos de datos primitivos en C++ y sus características.

C++	
Tipo	Mutable
Arrays	X
Vectores	X
Constantes	
Objetos	X
Punteros constantes	

### 4.2 Código de ejemplo

Listing 4: Ejemplo de código en C++

```
#include <vector>
#include <iostream>

int main() {
    std::vector<int> mutableVector = {1, 2, 3}; // Inicializamos un vector mutable
    mutableVector.push_back(4); // Agregar un nuevo elemento al final
    mutableVector[1] = 5; // Modificar el segundo elemento
    std::cout << "Vector mutable modificado:";
    for (int num : mutableVector) {
        std::cout << " " << num;
    }
    std::cout << std::endl;
    return 0;
}
```

En este ejemplo, la complejidad es  $O(n)$ , donde  $n$  representa la cantidad de elementos en el vector. Aunque la creación de un vector, la adición de un elemento y la edición de un elemento en una posición dada toman un tiempo constante, el recorrido de cada elemento para imprimirlo implica una complejidad  $O(n)$ , donde  $n$  es la cantidad de elementos.