

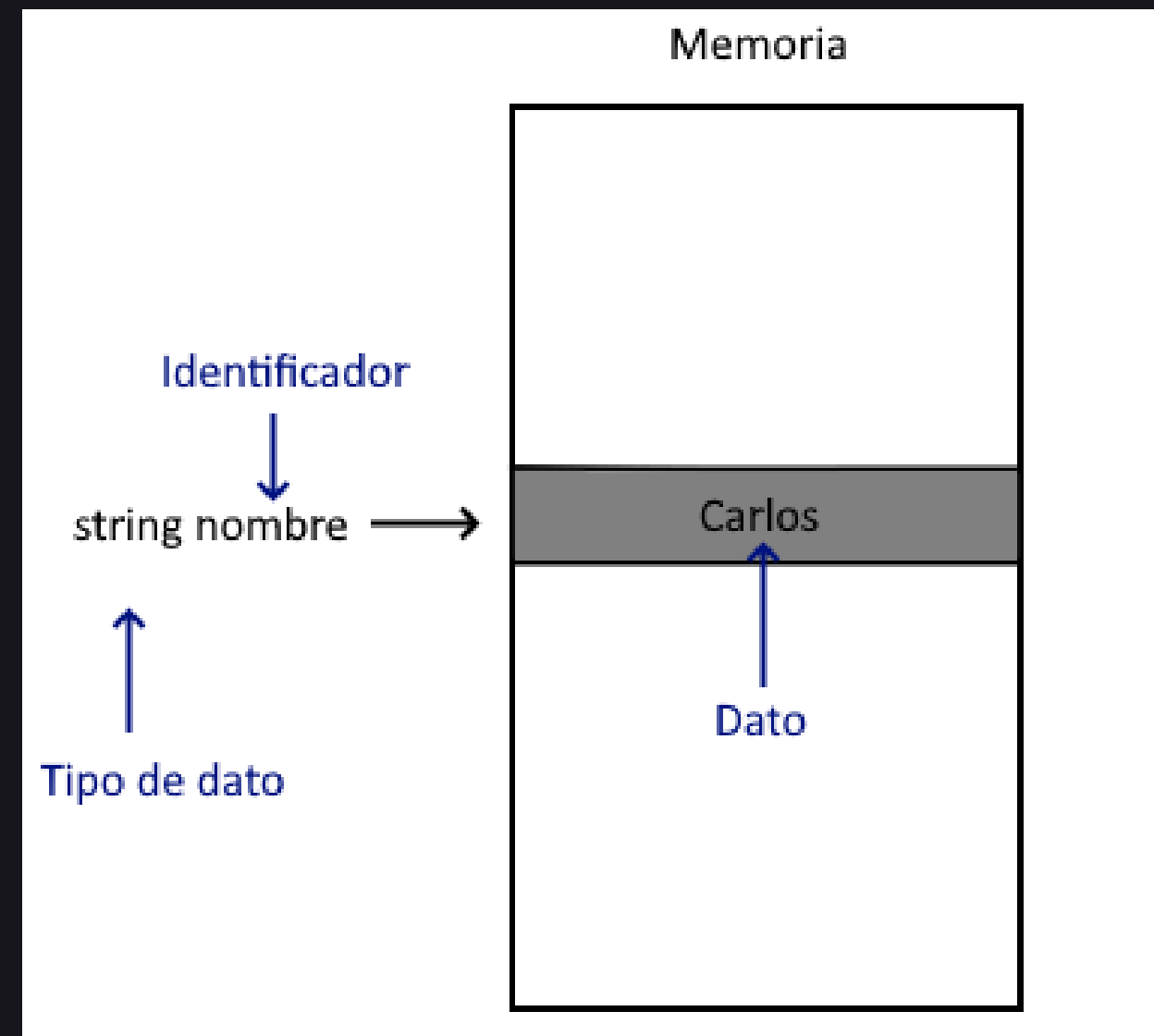
# Conceptos TP N°1

- **Variables**
- **Tipo de datos**
- **Constantes**
- **Operadores**
- **Palabras reservadas**
- **String: Aclaración**



# Variables

**Una variable es un espacio en la memoria de la computadora que permite almacenar temporalmente información (datos) durante la ejecución del algoritmo, y cuyo contenido puede cambiar mientras se ejecuta el algoritmo.**



# Las variables deben tener un identificador o etiqueta.

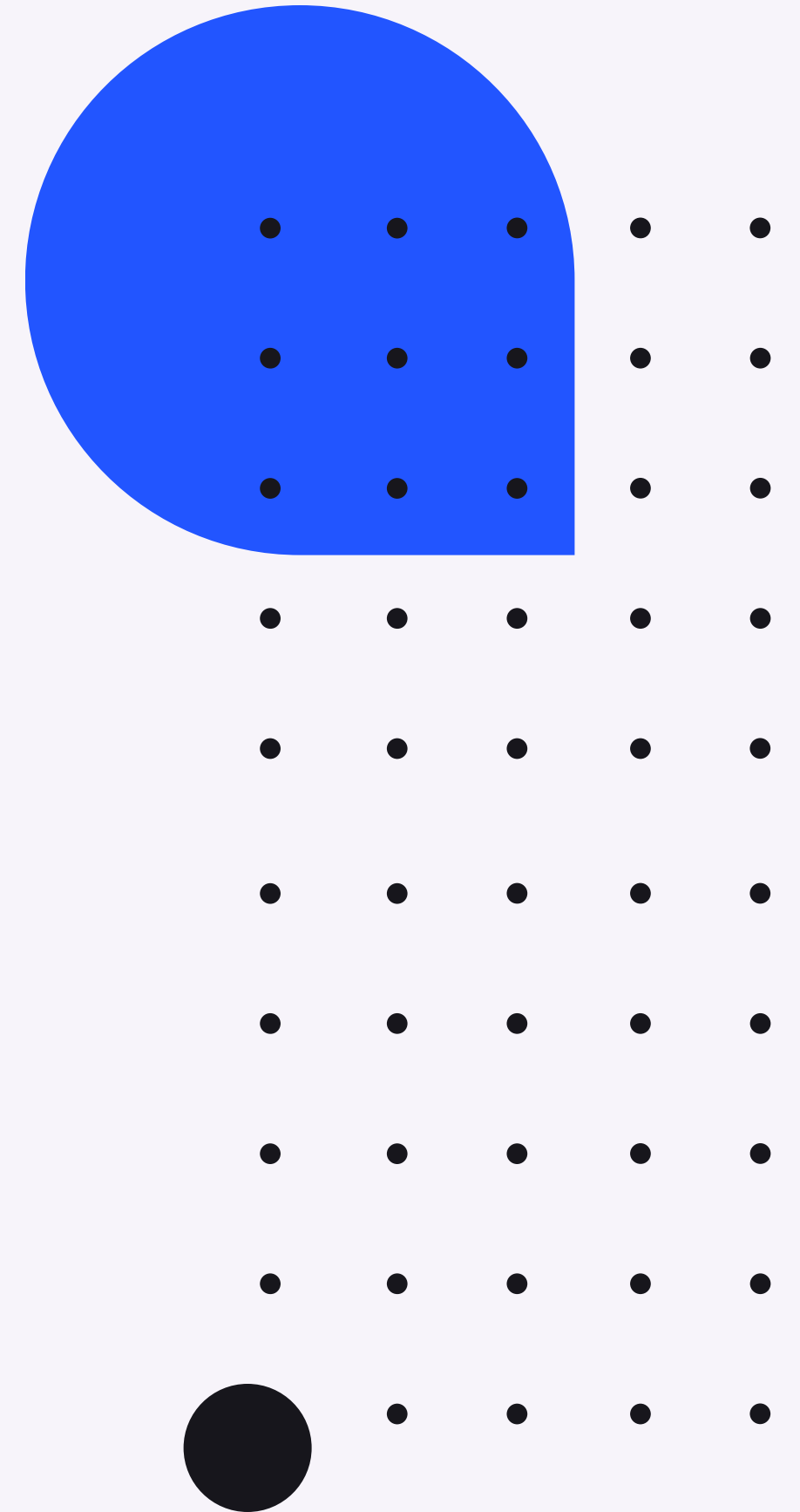
- Deben comenzar por una letra.
- No debe haber espacios.
- No se debe acentuar.
- **Debe ser representativa.**

## Forma correcta

- nombre
- carrera1
- ciudad\_natal
- cancion

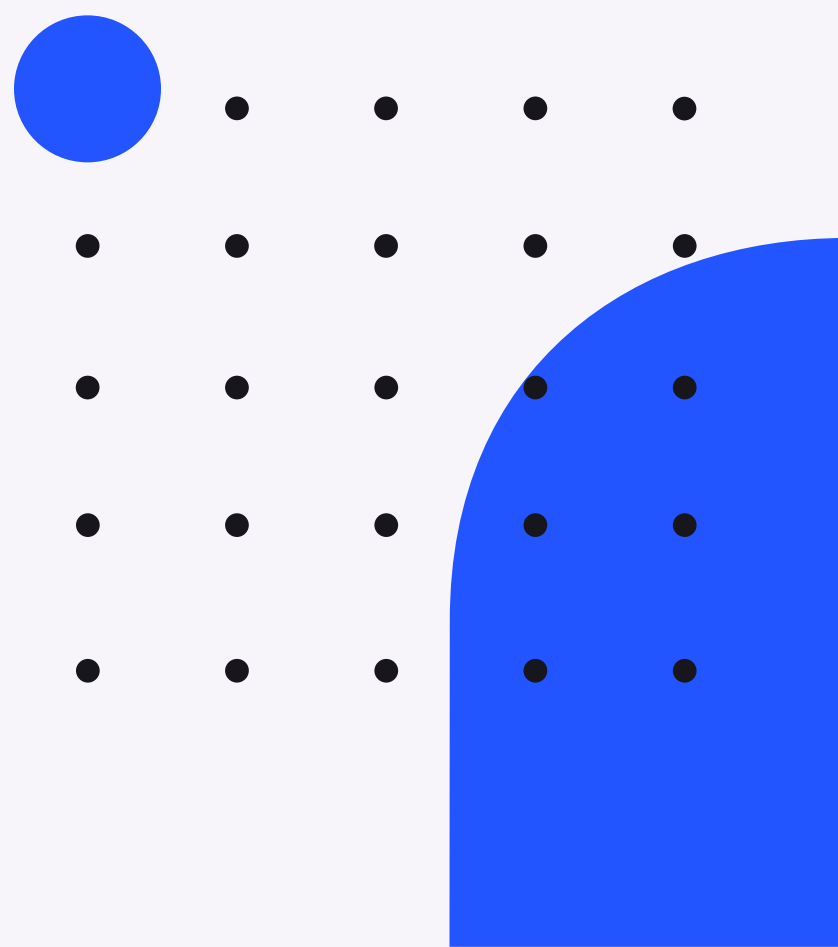
## Forma incorrecta

- **n**
- 1carrera
- ciudad natal
- canción



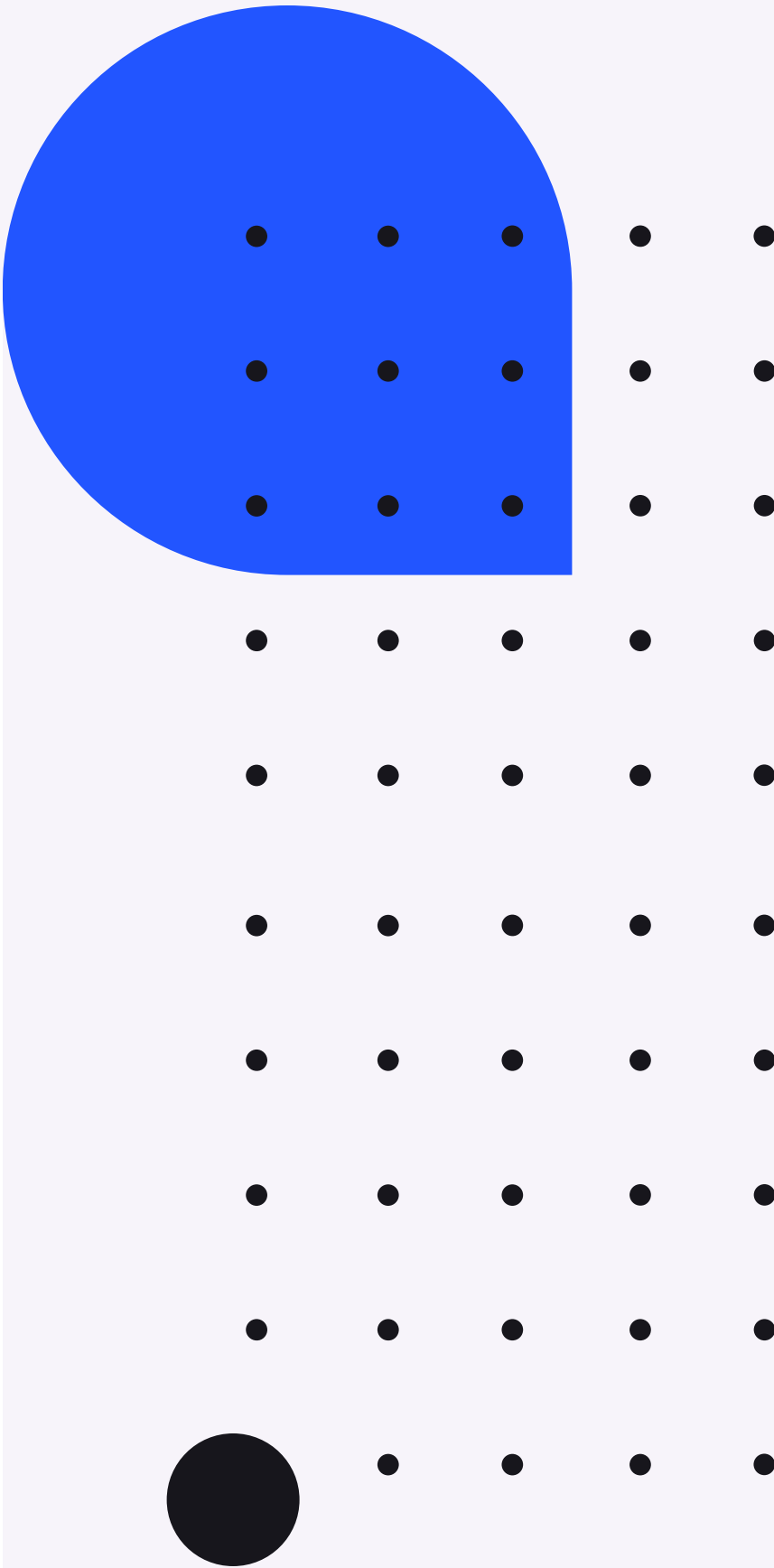
# Tipo de datos

**Al declarar una variable debemos indicar el tipo de dato que es permitido almacenar en dicha variable.**



# Tipos de datos

Tipo	Tam. Bits	Dígitos de precisión	Rango	
			Min	Max
● Bool	8	0	0	1
● Char	8	2	-128	127
Signed char	8	2	-128	127
unsigned char	8	2	0	255
short int	16	4	-32,768	32,767
unsigned short int	16	4	0	65,535
● Int	32	9	-2,147,483,648	2,147,483,647
unsigned int	32	9	0	4,294,967,295
long int	32	9	-2,147,483,648	2,147,483,647
unsigned long int	32	9	0	4,294,967,295
long long int	64	18	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
unsigned long long int	64	18	0	18,446,744,073,709,551,615
● Float	32	6	1.17549e-38	3.40282e+38
● Double	64	15	2.22507e-308	1.79769e+308

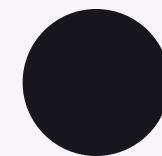


# Ejemplo

```
#include <iostream>
using namespace std;

int main() {
    int num_1, num_2;
    num_1 = 15;
    num_2 = 3;
    cout << "Suma y resta respectivamente \n";
    cout << num_1 + num_2<<endl;
    cout << num_1 - num_2;
    return 0;
}
```

```
Suma y resta respectivamente
18
12
```



# Constantes

**Representa un espacio de memoria RAM que guarda un valor que servirá para algún proceso en particular, dicho valor permanece fijo, es decir no puede cambiarse en la ejecución del programa**



**Las constantes tienen al igual que las variables un identificador (nombre) y un tipo de dato.**



# Ejemplo

```
int main() {  
    //Declaro la constante PI y le doy un valor  
    const float PI=3.14159;  
    //Intento cambiar el valor  
    PI=3.14;  
    return 0;  
}
```

```
prog.cpp: In function 'int main()':  
prog.cpp:8:4: error: assignment of read-only variable 'PI'
```

```
8 | PI=3.14;  
  | ~~~~~
```





# Operadores

**Son los que permiten realizar los cálculos entre valores fijos y variables.**

**Los operadores se clasifican por:**

- Operadores aritméticos
- Operadores relacionales
- Operadores lógicos
- Operadores cadena



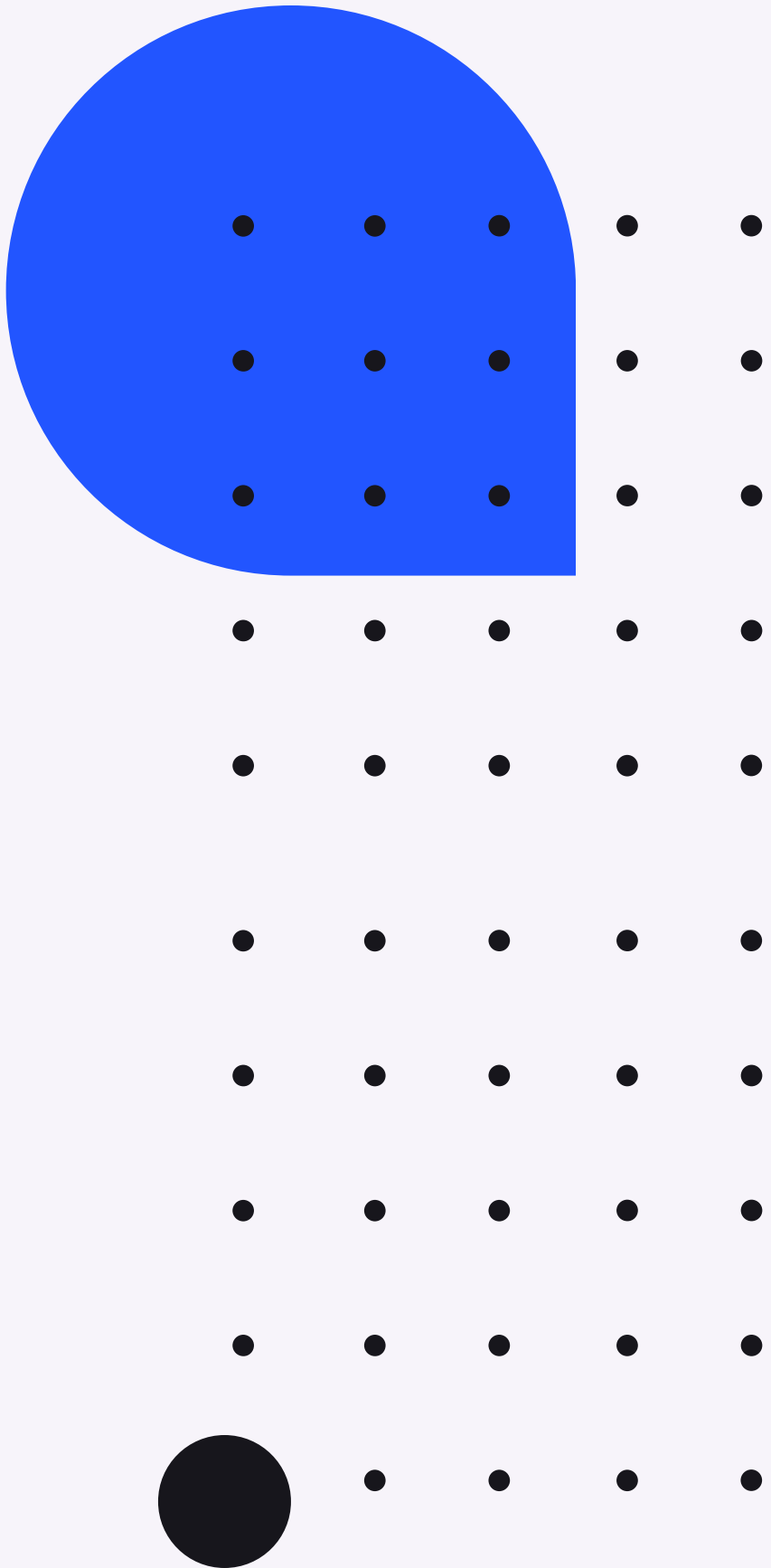
**Operadores Aritméticos:** Son aquellos operadores que permiten realizar las operaciones aritméticas, de la misma forma como se utilizan en las matemáticas.

Operador	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
%	Módulo (resto de una división)



**Operadores Relacionales:** Llamados también operadores de comparación y permiten evaluar si dos valores guardan alguna relación entre si.

Operador	Descripción
==	Igualdad
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que
!=	Diferente a

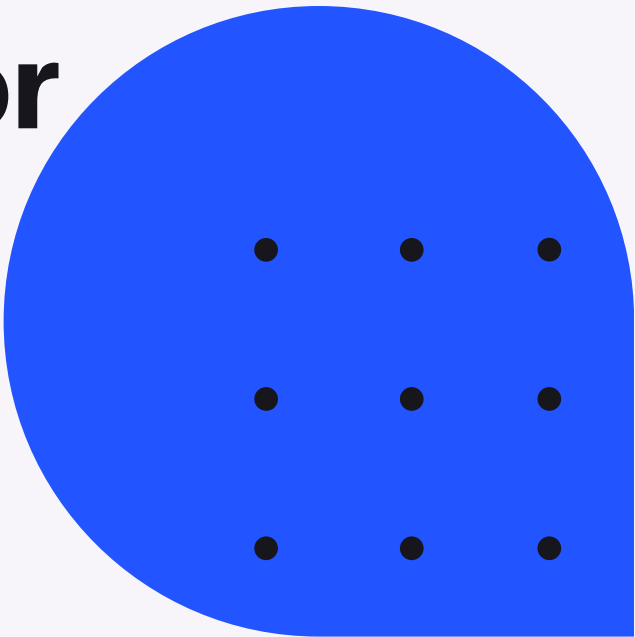


**Operadores Lógicos:** Son aquellos operadores que se utilizan en combinación con los operadores de relación.



Operador	Descripción
&&	Y Lógico
	O Lógico
!	No Lógico

**Operadores de Cadena:** Son aquellos operadores que permiten realizar operaciones con cadenas, por lo general permiten unir cadena llamado también concatenar.



Operador	Descripción
+	Unir cadenas
&	Unir Cadenas

**Ejemplo:**

"Ricardo" + " " + "Marcelo"	Ricardo Marcelo
"ricardomarcelo" & "@" & "hotmail.com"	ricardomarcelo@hotmail.com



# Palabras reservadas

Son identificadores predefinidos que tienen significados especiales y no pueden usarse como identificadores creados por el usuario en los programas.

## Palabras heredadas de C      Palabras que solo utiliza C++, no provienen de C

```
auto  const  double  float  int  
short  struct  unsigned  break  continue  
else  for  long  signed  switch  
void  case  default  enum  goto  
register  sizeof  typedef  volatile  char  
do  extern  if  return  static  
union  while
```

```
asm  dynamic_cast  namespace  reinterpret_cast  try  bool  explicit  
new  static_cast  typeid  catch  false  operator  template  
typename  class  friend  private  this  using  const_cast  inline  
public  throw  virtual  delete  mutable  protected  true  wchar_t
```

## Alternativas para algunos operadores de C++

```
and  bitand  compl  not_eq  or_eq  xor_eq  and_eq  bitor  not  or  >xor
```

## String / Aclaración

La cadena de caracteres, realmente no es un tipo de dato, sino que el string, que representa una cadena de caracteres es una clase.

Al crear una variable de una clase, a esta variable se le llama objeto, y ese objeto además de guardar información puede ejecutar funciones o métodos que trabajan sobre los datos que tienen guardados.

**En este curso no vamos a ver programación orientada a objetos. Por lo tanto, usaremos la clase string como tipo de dato.**



# Ejemplo uso string:

```
#include <iostream>
#include <string>    //incluimos la biblioteca
using namespace std;

int main() {
    string ciudad, provincia; //declaramos la "variable" tipo string
    ciudad = "Bahia Blanca";
    provincia = "Buenos Aires";

    cout << "La ciudad de " << ciudad << " queda en la provincia de "
    << provincia;
    return 0;
}
```

La ciudad de Bahia Blanca queda en la provincia de Buenos Aires

