

Clase 2: Tipos de Datos y Condiciones

Repaso

`cin`

```
string nombre;  
cin >> nombre;
```

`cout`

```
cout << "¿Cómo te  
llamas?";
```

Tipos de datos básicos

Imaginemos que tenemos cajas de distintos tipos y tamaños que utilizamos para guardar objetos. Cada caja tiene pegada una etiqueta, donde sale el **nombre** de lo que tiene adentro y el **tipo** de objeto que guarda.

Siguiendo la analogía, una **variable** corresponde una de estas cajas. Una **variable** es un nombre simbólico asociado a un valor que queremos guardar. El valor de una variable puede cambiar dependiendo de las condiciones e información que recibe el programa. Algunos tipos de variables son los `int` (números enteros), `double` (números decimales), `string` (letras).

Por ejemplo, si Josefa guarda su puntaje PSU en una caja y después da la PSU de nuevo, tendría que cambiar el valor que guardó en su caja **puntajePSU**.





¿Cómo creo una variable?

Para crear una variable, es necesario asignarle un nombre y decir cuál es su tipo. Ejemplo:

```
int puntajePSU;  
string nombreDeMiGato;
```

¿Cómo le doy un valor a una variable?

Una vez creada la variable, para darle un valor basta con utilizar el símbolo =. Ejemplo:

```
string nombreDeMiGato;  
nombreDeMiGato = "Señor bigotes";
```

Asignación de variables

Otra forma de crear variables es declarando su tipo, nombre y valor en una misma línea.

```
int numeroDeCelular = 87654321;  
string usuarioInstagramDeMiGato = "bigotes";  
double decimal = 8.9993920;
```

Tipos de datos básicos

| Tipo | Ejemplos |
|------|----------|
|------|----------|



| | |
|-------------------------------------|-----------------------------|
| Números enteros (int) | 1, 32, 298, -54, -700 |
| Números decimales (double) | 1.5, -3.89, 5.0, -23.9876 |
| Caracteres (char) | 'A', 'z' |
| Palabras (string) | "Hola", "maria", "Un barco" |
| Valores de verdad (bool) | true, false |

Operaciones básicas

Operaciones matemáticas básicas

- Suma +
- Resta -
- Multiplicación *
- División /
- Módulo o resto de una división %

$$5 : 2 = 2$$

1 $\xrightarrow{\text{Resto}}$

```
int modulo = 5 % 2; //el  
resultado de esto es 1
```

Operaciones más avanzadas

Para trabajar con operadores matemáticos más avanzados debes escribir en la parte superior de tu programa **#include <math.h>**. Ejemplo:

```
#include<cmath>  
int potencia= pow(2,3); //el resultado es 8  
float raizCuadrada= sqrt(9); // el resultado es 3
```

Operaciones con string

Para trabajar con string debes escribir en la parte superior de tu programa **#include <string>** y **using namespace std;**

```
#include <string>  
using namespace std;  
string saludo= "Hola mundo";
```



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

int main(){
    string saludo= "Hola mundo";
    cout<<saludo;
    return 0;
}
```

Suma

Para unir string puedes utilizar el operador +

```
#include <string>
using namespace std;
string saludo= "Hola ";
string amigas= "Ada, Frida, Marie";
string saludoAmigas= saludo + amigas;

#include <string>
#include <iostream>
using namespace std;
int main(){
    string saludo = "Hola ";
    string amigas = "Ada, Frida, Marie";
    string saludoAmigas = saludo + amigas;
    cout << saludoAmigas;
    return 0;}
```

Tamaño de un texto

Para saber cuál es el tamaño de un texto puedes usar **length()** y **size()**

```
#include <string>
using namespace std;
string saludo = "Hola";
int tamanoSaludo1 = saludo.length() //el resultado es 4
int tamanoSaludo2 = saludo.size() //el resultado es 4
```



Operadores Condicionales

Operadores de comparación

`a==b`

¿a es igual a b?

`a!=b`

¿a es distinto de b?

Desigualdades

<

Menor que

>

Mayor que

<=

Menor o igual que

>=

Mayor o igual que

Ejemplos

| <code>!= ==</code> | <code>< ></code> | <code><= >=</code> |
|--------------------|------------------------|--------------------------|
| ¿3==3? Verdadero | 3<5 | 3<=5 |
| ¿"a"=="A"? Falso | a>b | 5>=9 |
| ¿5!=3? Verdadero | -3>-1 | -3<=-1 |

Operadores lógicos

!

Invierte el valor de verdad

&&

Evalúa dos condiciones e indica **si ambas son ciertas**

||

Evalúa dos condiciones e indica si **alguna de ellas es cierta**

Controles de flujo: condiciones

Si mides más de 1.40 **entonces**:
 puedes subir al juego
Si no:
 no puedes subir

```
if (estatura >= 1.40)  
    puedes subir al juego  
else  
    no puedes subir
```

```
if (condición)  
{  
    /*Código que se ejecuta si se cumple la condición*/  
}
```

```
if (condición)  
{  
    /*Código que se ejecuta si se cumple la condición*/  
}  
else  
{  
    /*Código que se ejecuta si no se cumple la condición*/  
}
```

```
if (condición1)  
{  
    /*Código que se ejecuta si se cumple la condición 1*/  
}  
else if (condición2)  
{  
    /*Código que se ejecuta si no se cumple la condición 2*/  
}  
else  
{  
    /*Código que se ejecuta si no se cumple ninguna condición*/  
}
```

Ejemplo: **Entrada al cine**



Las condiciones:



```
if (dia==miercoles){
    La entrada está a la mitad del precio normal
}
else if (tiene descuento y la pelicula no tiene restriccion){
    Se le aplica el descuento al precio de la entrada
}
else {
    Se mantiene el precio normal
}
```

El programa es el siguiente:

```
#include <string>
#include <iostream>
using namespace std;
int main(){

    // Las variables
    int precioNormal=5600;
    string dia= "lunes";
    int descuento=10;
    bool restriccionPelicula=true;
    int precioEntradaFinal;

    // Las condiciones
    if (dia == "miercoles"){
        precioEntradaFinal=precioNormal/2;
    }
    else if (descuento>0 && !restriccionPelicula ){
        precioEntradaFinal=precioNormal-(precioNormal*descuento/100);
    }
    else {
        precioEntradaFinal=precioNormal;
    }

    // Mostrar el resultado
    cout<<precioEntradaFinal;
    return 0;
}
```





Dado que es sábado y
la película que
queremos ver tiene
restricción
tendremos que pagar
el precio normal