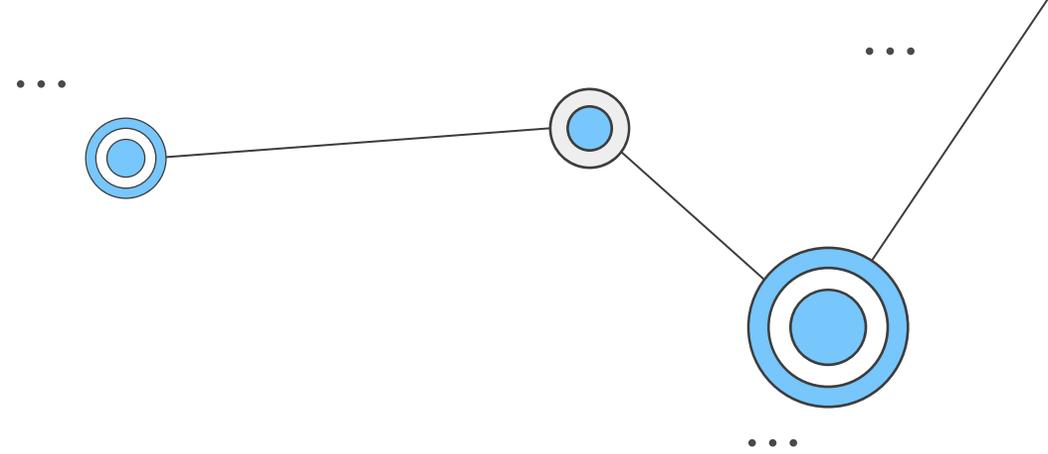
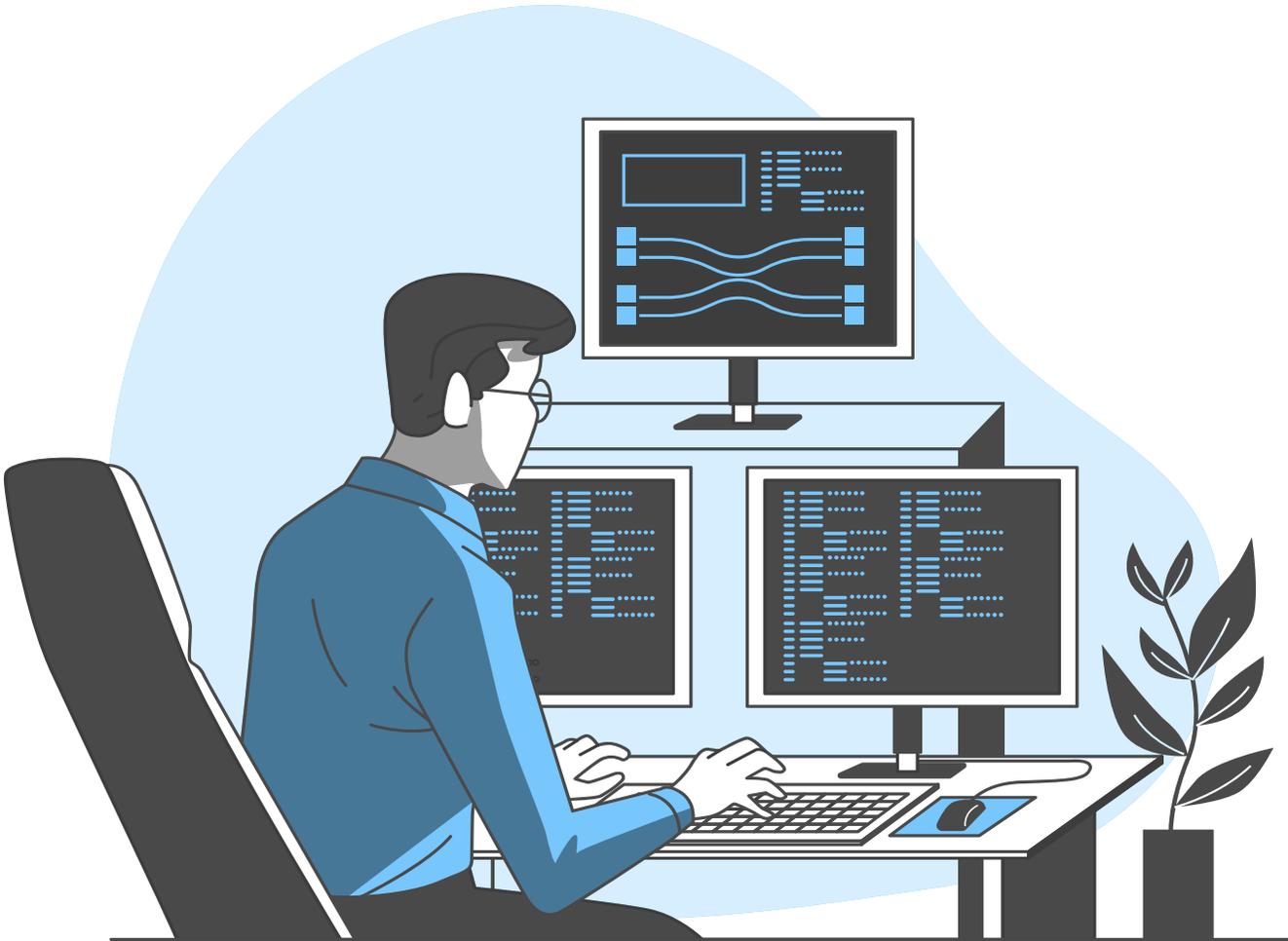




Unach
UNIVERSIDAD NACIONAL DE CHIMBORAZO
Libres por la Ciencia y el Saber



UNIDAD 1

Introducción a la
Programación

Contenidos



01
...

Primer programa
Estructura básica de un
programa en C++
Buenas prácticas

03
...

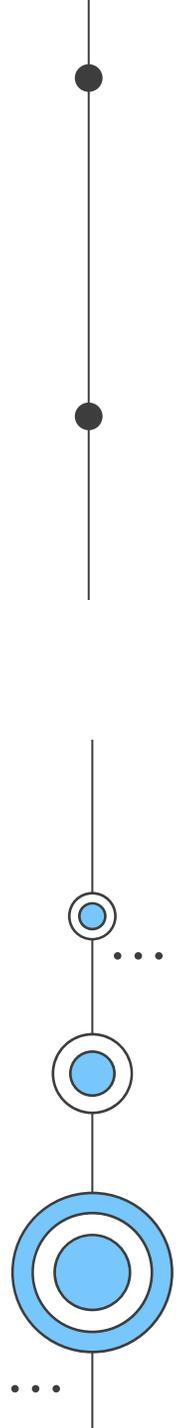
TEMA 1.
Tipos de datos
Variables y constantes



01

Programación en C++

TU PRIMER PROGRAMA EN C++



¿Qué es un programa en C++?



Un programa es un conjunto de instrucciones que le damos a la computadora para que realice una tarea específica.

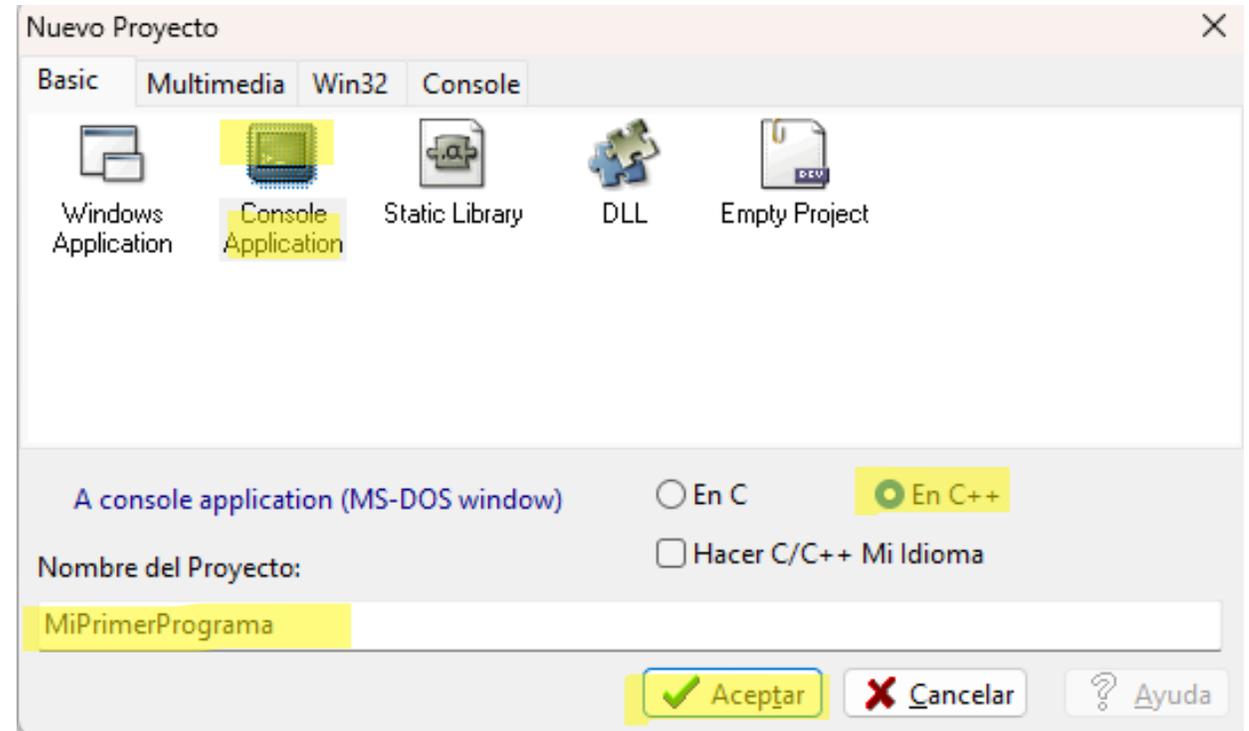
En este caso, conjunto de líneas de código que ingresamos en el IDE de desarrollo para un propósito específico.

Primer programa en C++

Para crear un programa en C++ debe seguir la siguiente ruta.



Observe la imagen y configure su primer programa de acuerdo a lo que se muestra.



Estructura de un programa en C++

Declaración de librerías

Le dice al compilador que utilice el espacio de nombres (namespace) std de forma predeterminada

```
[*] main.cpp
1  #include <iostream>
2  using namespace std;
3  int main(int argc, char** argv)
4  {
5      cout<<"Hola primer semestre de la carrera de Ciencia de Datos";
6      system("pause");
7      return 0;
8  }
```

Punto de entrada de cualquier programa

Instrucción para imprimir por pantalla

Indica que el programa ha finalizado correctamente.

Pausar la ejecución del programa hasta que el usuario presione una tecla.

Buenas prácticas de programación

Documentación y comentarios: Introduzca comentarios en su código para hacerlo más legible tanto para cada uno, como para otros que puedan leer nuestro trabajo.

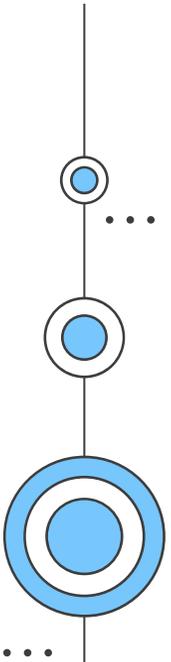
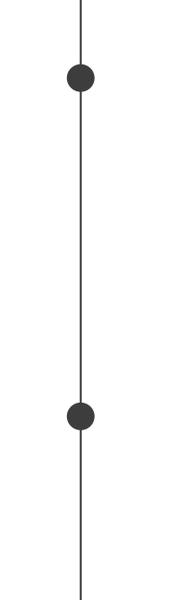
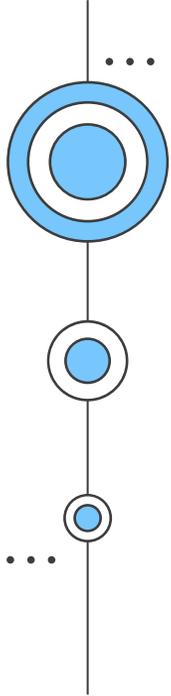
Indentación y formato del código: Establezca desde el principio hábitos para que su código esté bien estructurado y sea fácil de seguir.

```
1  #include <iostream>
2  using namespace std;
3  int main(int argc, char** argv) //entrada del programa
4  {
5      cout<<"Hola primer semestre de la carrera de Ciencia de Datos";
6      /*esto es un
7      comentario de
8      varias lineas*/
9      cout<<"Salto de linea 1"<<endl;
10     cout<<"Salto de linea 2\n";
11     system("pause");
12     return 0;
13 }
```

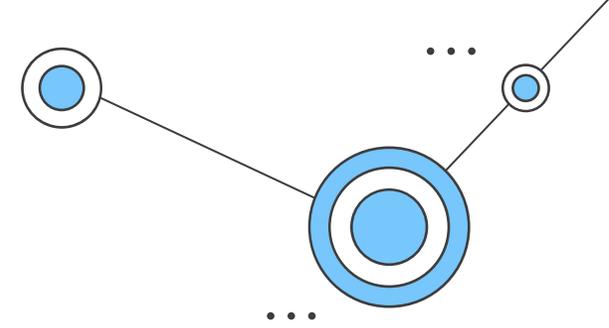
02

Tema

TIPOS DE DATOS, VARIABLES Y
CONSTANTES



Tipos de Datos en C++



...

Enteros

int: Almacena números enteros.

short: Almacena números enteros más pequeños.

long: Almacena números enteros más grandes.

long long: Almacena enteros más grandes que long.

unsigned int, unsigned short, unsigned long, unsigned long long: Enteros sin signo, es decir, solo números positivos o cero.

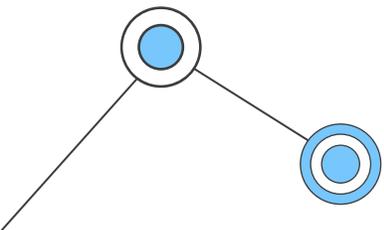
...

Flotantes

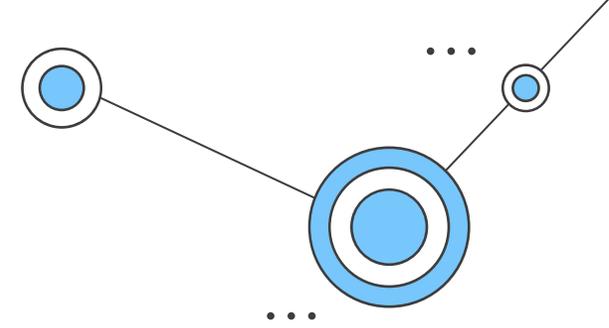
float: Almacena números en coma flotante.

double: Almacena números en coma flotante de mayor precisión.

long double: Almacena números en coma flotante con aún más precisión que double.



Tipos de Datos en C++



...

Caracter

char: Almacena un solo carácter.

string: cadenas de caracteres.

...

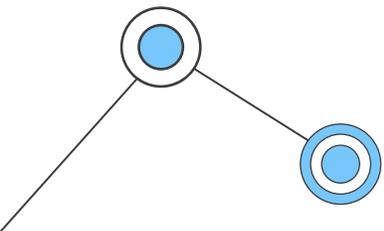
Booleano

bool: Almacena valores lógicos: true o false.

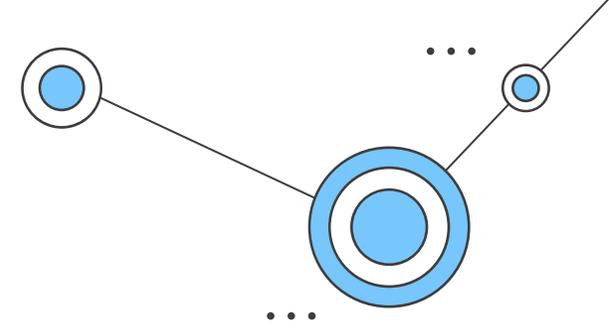
...

Vacío

void: No es un tipo de dato que almacene información, sino que se usa en funciones que no devuelven valor.



Variables definidas para datos enteros

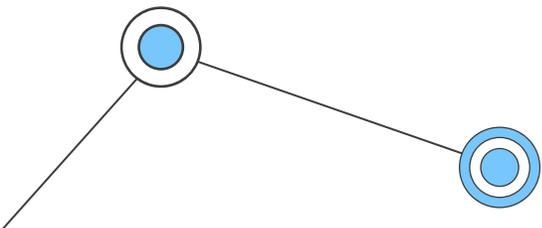


Int

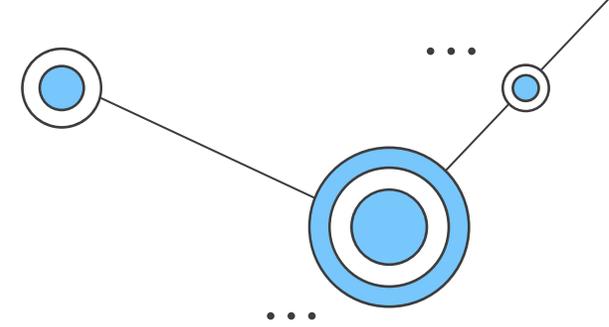
```
#include <iostream>
using namespace std;
int main()
{
    int edad = 25;
    cout << "Tu edad es: " << edad << endl;
    system("pause");
    return 0;
}
```

Short

```
#include <iostream>
using namespace std;
int main()
{
    short num_hijos = 3;
    cout << "Tu tienes " << num_hijos << " hijos " << endl;
    system("pause");
    return 0;
}
```



Variables definidas para datos enteros

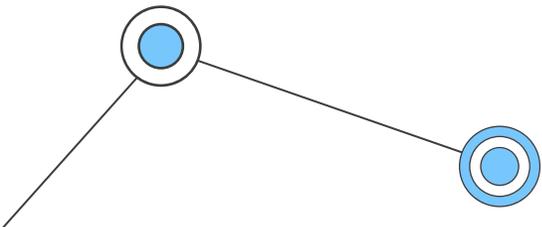


long/long long

```
#include <iostream>
using namespace std;
int main()
{
    long distancia = 1000000;
    long long distanciaGrande = 10000000000;
    cout << "Distancia (long): " << distancia << endl;
    cout << "Distancia grande (long long): " << distanciaGrande << endl;
    system("pause");
    return 0;
}
```

unsigned

```
#include <iostream>
using namespace std;
int main()
{
    unsigned int positivo = 300;
    cout << "Número positivo sin signo: " << positivo << endl;
    system("pause");
    return 0;
}
```



Variables definidas para datos decimales

float

```
#include <iostream>
using namespace std;
int main()
{
    float pi = 3.14;
    cout << "El valor de pi es: " << pi << endl;
    system("pause");
    return 0;
}
```

double

```
#include <iostream>
#include <iomanip>

using namespace std;
int main()
{
    double extension_terreno = 1236546.789123;
    cout << fixed << setprecision(6); //cuando se desee mostrar todos los decimales
    cout << "La extensión es: " << extension_terreno << endl;
    system("pause");
    return 0;
}
```

long double

```
#include <iostream>
using namespace std;
int main()
{
    long double valorPreciso = 2.7182818284590452353602874L;
    cout << "Valor preciso de e: " << valorPreciso << endl;
    system("pause");
    return 0;
}
```

Variables definidas para caracteres

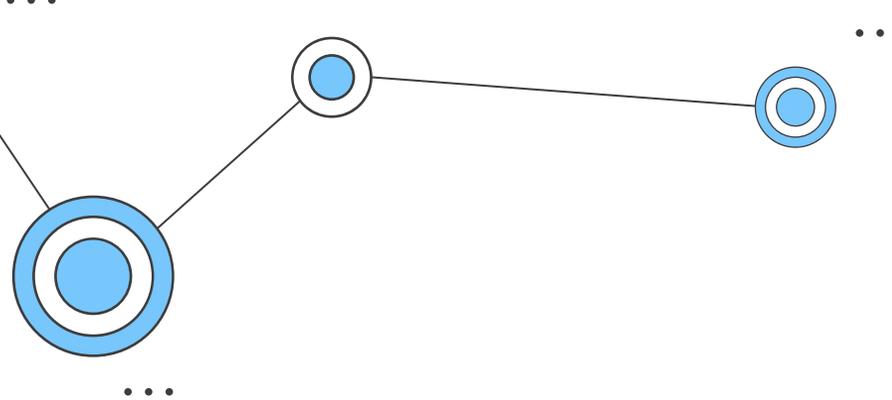
char

```
#include <iostream>
using namespace std;
int main()
{
    char letra = 'A';
    cout << "La letra es: " << letra << endl;
    system("pause");
    return 0;
}
```

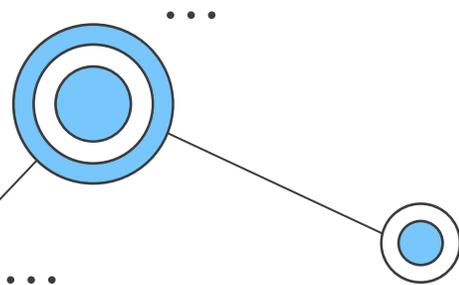
string

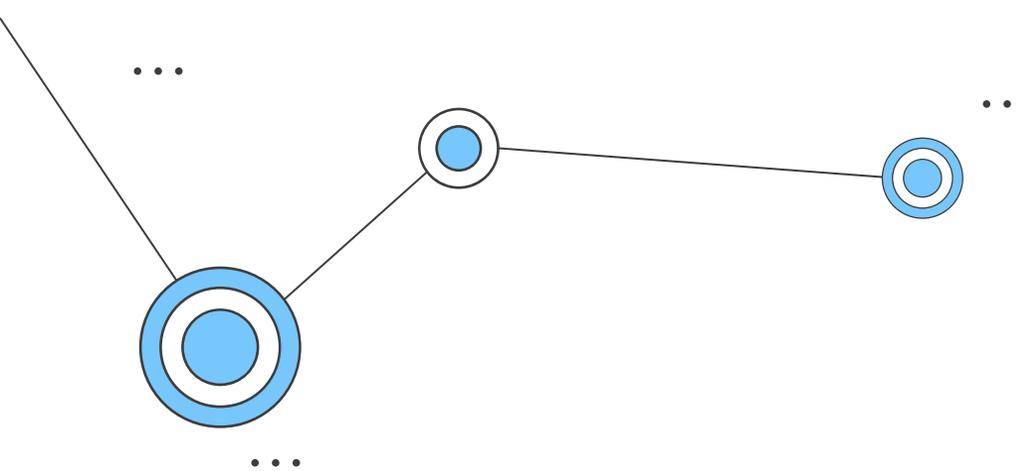
```
#include <iostream>
using namespace std;
int main()
{
    string nombres_completos = "Fernando Perez Silva";
    cout << "Su nombre es: " << nombres_completos << endl;
    system("pause");
    return 0;
}
```

Variables booleanas



```
#include <iostream>
using namespace std;
int main()
{
    bool esMayorDeEdad = true;
    cout << "Es mayor de edad: " << esMayorDeEdad << endl; // true se imprime como 1
    system("pause");
    return 0;
}
```

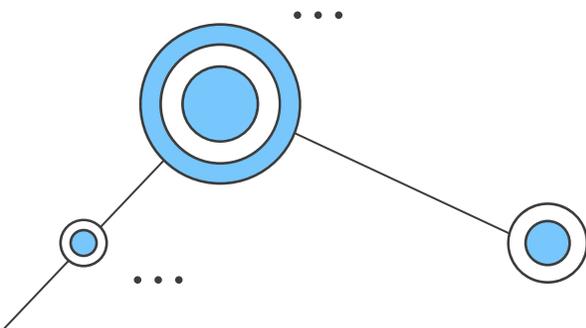




Variables contadoras y acumuladoras

- EJEMPLO: USO DE VARIABLES CONTADORAS Y ACUMULADORAS EN UN EJERCICIO SENCILLO.

```
#include <iostream>
using namespace std;
int main()
{
    int contadorPositivos = 0; // Variable contadora para números positivos
    int sumaTotal = 1;        // Variable acumuladora para sumar números
    // Primer número
    int numero = 5; // Ejemplo: 5 es positivo
    sumaTotal = sumaTotal+numero; // Acumula el valor del número
    contadorPositivos=contadorPositivos+1; // Incrementa el contador
    cout<<sumaTotal<<endl;
    cout<<contadorPositivos<<endl;
    system("pause");
    return 0;
}
```



Constantes

- EJEMPLO: CÁLCULO DEL ÁREA DE UN CÍRCULO UTILIZANDO UNA CONSTANTE Y VARIABLES.

```
#include <iostream>
using namespace std;
int main()
{
    // Declaración de una constante
    const double PI = 3.14159;
    // Declaración de variables
    double radio, area;
    // Pedir al usuario el radio del círculo
    cout << "Ingresa el radio del círculo: ";
    cin >> radio;
    // Calcular el área del círculo
    area = PI * radio * radio; // Usamos la constante PI
    // Mostrar el resultado
    cout << "El área del círculo con radio " << radio << " es: " << area << endl;
    // Intentar modificar PI causaría un error, ya que PI es una constante:
    return 0;
    system("pause");
    return 0;
}
```



FIN

