

# Cifrado César

## Cesar.h

```
#pragma once

#include <string>

using namespace std;

class CifradoCesar
{
public:
    CifradoCesar();
    string cifrarMensaje(string mensaje);
    string descifrarMensaje(string mensajeCifrado);
    string abc;

private:
    int clave;
};
```

## Cesar.cpp

```
#include "Cesar.h"

CifradoCesar::CifradoCesar()
{
    abc = "abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ,.";
    clave = 13;
}

string CifradoCesar::cifrarMensaje(string mensaje)
{
    string mensajeCifrado;
    int pos;

    for (int i = 0; i < mensaje.length(); i++)
    {
        pos = abc.find(mensaje[i]) + clave;
        if (pos >= abc.length())
        {
            pos = pos % abc.length();
        }
        mensajeCifrado += abc[pos];
    }
    return mensajeCifrado;
}

string CifradoCesar::descifrarMensaje(string mensajeCifrado)
{
    string mensajeDescifrado;
    int pos;

    for (int i = 0; i < mensajeCifrado.length(); i++)
    {
        pos = abc.find(mensajeCifrado[i]) - clave;
        if (pos < 0)
        {
            pos = abc.length() + pos;
        }
        mensajeDescifrado += abc[pos];
    }
    return mensajeDescifrado;
}
```

## Main.cpp

```
#include <iostream>
#include "Cesar.h"

int main()
{
    CifradoCesar cesar;
    string mensaje, mensajeCifrado;
    int opcion;

    cout << "\nCifrado Cesar: \n\n1. Cifrar mensaje\n2.Descifrar mensaje\n\n";
    cin >> opcion;

    switch (opcion)
    {
    case 1:
        cout << "\nIntroducir mensaje: ";
        getline(cin.ignore(), mensaje);
        cout<<"\nMensaje cifrado: "<<cesar.cifrarMensaje(mensaje);
        break;
    case 2:
        cout << "\nIntroducir mensaje: ";
        getline(cin.ignore(), mensajeCifrado);
        cout<<"\nMensaje descifrado: "<<cesar.descifrarMensaje(mensajeCifrado);
        break;
    default:
        break;
    }
}
```

# Cifrado De Vigenère

## Vigenere.h

```
#pragma once

#include <string>

using namespace std;

class Vigenere
{
public:
    Vigenere();
    string cipherText(string mensaje);
    string decipherText(string mensajeCifrado);
    string abc;

private:
    string clave;
};
```

## Vigenere.cpp

```
#include "Vigenere.h"

Vigenere::Vigenere()
{
    abc = "abcdefghijklmnopqrstuvwxyz -ABCDEFGHIJKLMNOPQRSTUVWXYZ, .0123456789";
    clave = "AlgebraAbstracta";
}

string Vigenere::cipherText(string mensaje)
{
    string mensajeCifrado;
    int posM, posC, k = 0;

    for (int i = 0; i < mensaje.length(); i++)
    {
        posM = abc.find(mensaje[i]);
        posC = abc.find(clave[k]);
        k += 1;

        if (k >= clave.length())
        {
            k = 0;
        }

        posM += posC;

        if (posM >= abc.length())
        {
            posM = posM % abc.length();
        }

        mensajeCifrado += abc[posM];
    }
    return mensajeCifrado;
}

string Vigenere::decipherText(string mensajeCifrado)
{
    string mensaje;
    int posM, posC, k = 0;

    for (int i = 0; i < mensajeCifrado.length(); i++)
    {
        posM = abc.find(mensajeCifrado[i]);
        posC = abc.find(clave[k]);
        k += 1;

        if (k >= clave.length())
        {
            k = 0;
        }

        posM -= posC;

        if (posM < 0)
        {
            posM = posM + abc.length();
        }

        mensaje += abc[posM];
    }
    return mensaje;
}
```

## Main.cpp

```
#include "Vigenere.h"
#include <iostream>

int main()
{
    Vigenere vigenere;

    string mensaje, mensajeCifrado;
    int opcion;

    cout << "\nCifrado De Vigenere: \n\n1. Cifrar mensaje\n2. Descrifrar mensaje\n\n";
    cin >> opcion;

    switch (opcion)
    {
    case 1:
        cout << "Mensaje: \n";
        getline(cin.ignore(), mensaje);
        cout << endl;
        cout << vigenere.cipherText(mensaje) << endl;
        break;
    case 2:

        cout << "Mensaje cifrado: \n";
        getline(cin.ignore(), mensajeCifrado);
        cout << endl;
        cout << vigenere.decipherText(mensajeCifrado) << endl;
        break;

    default:
        break;
    }
}
```