

## EJERCICIO 4

### ===Torre.h===

```
#pragma once
#include <iostream>

class Hanoi {
public:
    void moverDiscos(int n, char origen, char destino, char auxiliar);
};

#include "Torre.tpp"
```

### ===Torre.tpp===

```
#include "Torre.h"
#include <iostream>
#include <stdexcept>

void Hanoi::moverDiscos(int n, char origen, char destino, char auxiliar)
{
    if (n <= 0)
    {
        throw std::invalid_argument("El número de discos debe ser mayor que cero.");
    }
    else if (n == 1) // Caso base: si solo hay un disco, se mueve directamente del
origen al destino.
    {
        std::cout << "Mover disco 1 de " << origen << " a " << destino << std::endl;
    }
    else
    {
        //Hay 2 casos recursivos que debemos tomar en la torre de hanoi
        moverDiscos(n - 1, origen, auxiliar, destino); //Mover n-1 discos del origen al
auxiliar usando destino como apoyo
        std::cout << "Mover disco " << n << " de " << origen << " a " << destino <<
std::endl;
        moverDiscos(n - 1, auxiliar, destino, origen); //Mover los n-1 discos del
auxiliar al destino usando origen como apoyo
    }
}

/* ¿CUÁNDO ESTO SERÍA INFINITO?
    La recursión se volvería infinita si si se usa 'n' en lugar de 'n - 1'
    lo que provocaría un ciclo sin fin y un desbordamiento de pila.
    Tambien sería infinita si no se definene de forma correcta las 3 torres, por lo
que también
    se generaría un desbordamiento.
*/
```

```
/* ¿POR QUÉ ES UNA SOLUCIÓN NATURAL?
```

```
La Torre de Hanoi se define de forma recursiva:
```

```
Para mover n discos, primero se mueven n-1 discos a un soporte auxiliar,  
luego se mueve el disco más grande, y finalmente se mueven los n-1 discos al  
destino.
```

```
Esta estructura se adapta perfectamente a la recursión, ya que cada paso es una  
versión más pequeña del mismo problema.
```

```
Aunque puede resolverse iterativamente con estructuras más complejas, la  
recursión ofrece una solución clara y elegante.
```

```
*/
```

```
===main.cpp===
```

```
#include <iostream>
```

```
#include "Torre.h"
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    Hanoi torre;
```

```
    int disco;
```

```
    do
```

```
    {
```

```
        cout << "Ingrese el número de discos: ";
```

```
        cin >> disco;
```

```
        if (disco > 10)
```

```
        {
```

```
            cout << "Número de discos demasiado grande, por favor ingrese un número menor  
o igual a 10." << endl;
```

```
        }
```

```
        else
```

```
        {
```

```
            try
```

```
            {
```

```
                torre.moverDiscos(disco, 'A', 'C', 'B');
```

```
            }
```

```
            catch (const std::exception &e)
```

```
            {
```

```
                std::cerr << "Error a la hora de movimientos" << e.what() << '\n';
```

```
            }
```

```
        }
```

```
    } while (disco > 10);
```

```
}
```