

hpp lista\_generica.cpp lista\_generica.hpp papaCaliente.cpp lista.h vector.cpp ListaEnt

```
// lista_generica.cpp
// lista_generica

#include "lista_generica.hpp"
#include <iostream>

using std::cin;
using std::cout;
using std::endl;

int main(){
    /**/
    Lista<int> milista;
    cout << "La lista es: " << milista.comoCadena() << endl << endl;

    Lista<int> milista2 = milista;
    cout << endl;

    cout << (milista.estaVacia() ? "true" : "false") << endl << endl;

    milista.adjuntar(12);
    cout << "La lista es " << milista.comoCadena() << endl << endl;
    milista.adjuntar(55);
    cout << "La lista es " << milista.comoCadena() << endl << endl;

    milista.adjuntar(10);
    cout << "La lista es " << milista.comoCadena() << endl << endl;
}
```

```
Obteniendo el tamaño de la lista...
Tamaño de la lista: 5

Comprobando si la lista esta vacia...
Recuperando elemento en el indice 4
94.5

Comprobando si la lista esta vacia...
Removiendo el elemento con indice 2...
Desplazando elemento 365.5 del indice 3 al indice 2
Desplazando elemento 105.3 del indice 4 al indice 3
Reduciendo la cuenta de elementos
365.5

La lista es: [100.1, 107.7, 105.3, 94.5]

Insertando un elemento...
Insertando elemento 104.5 en el indice 4
Incrementando la cuenta de elementos
La lista es: [100.1, 107.7, 105.3, 94.5, 104.5]

Destruyendo la lista...
Destruyendo la lista...
Destruyendo la lista...
Destruyendo la lista...

-----
Process exited after 0.3157 seconds with return value 0
Presione una tecla para continuar . . .
cout << "La lista es " << milista.comoCadena()
```

```

        return rand() % max;
    }

void papaCaliente(const vector<string> &listaNombres){
    int cantidadJugadores = listaNombres.size();
    Cola<string> simulacion;
    for(int i = 0; i < cantidadJugadores; i++){
        simulacion.enqueue(listaNombres[i]);
    }

    for(int i = cantidadJugadores; i > 1; i--){
        cout << "Quien tiene la papa caliente? " << endl;
        for(int pases = numeroAleatorio(cantidadJugadores); pases > 0; pases--){
            string jugador = simulacion.dequeue();
            cout << jugador << endl;
            simulacion.enqueue(jugador);
            this_thread::sleep_for(chrono::milliseconds(10));
        }
        string funado = simulacion.dequeue();
        cout << "Alto!!! " << endl;
        cout << funado << " sale del juego..." << endl << endl;
        this_thread::sleep_for(chrono::milliseconds(2500));
    }
    string ganador = simulacion.dequeue();
    cout << " ***** WINNER ***** " << endl << endl;
    cout << ganador << " gano el juego." << endl;
}

```

```
9.Alberto
Insertando un elemento...
  Insertando elemento 9.Alberto en el indice 1
  Incrementando la cuenta de elementos
Comprobando si la lista esta vacia...
Removiendo el elemento con indice 0...
  Desplazando elemento 15.Carla del indice 1 al indice 0
  Reduciendo la cuenta de elementos
15.Carla
Insertando un elemento...
  Insertando elemento 15.Carla en el indice 1
  Incrementando la cuenta de elementos
Comprobando si la lista esta vacia...
Removiendo el elemento con indice 0...
  Desplazando elemento 9.Alberto del indice 1 al indice 0
  Reduciendo la cuenta de elementos
Alto!!!
9.Alberto sale del juego...

Comprobando si la lista esta vacia...
Removiendo el elemento con indice 0...
  Reduciendo la cuenta de elementos
***** WINNER *****

15.Carla gana el juego.
Destruyendo la lista...

-----
Process exited after 61.24 seconds with return value 0
Presione una tecla para continuar . . . █
```

```

#include <iostream>
#include <vector>

using namespace std;

int main(int argc, char *argv[]){
    vector<int> v;
    v.push_back(2);
    v.push_back(3);
    v.push_back(4);
    v.push_back(5);
    v.push_back(6);
    v.push_back(7);

    cout << v.size() << endl;
    cout << v.front() << endl;
    cout << v.back() << endl;

    for(int i = 0 ; i < v.size() ; i++){
        cout << v[i] << endl;
    }

    return 0;
}

```

```

6
2
7
2
3
4
5
6
7
-----
Process exited after 0.06823 seconds with return value 0
Presione una tecla para continuar . . .

```

```

#include "lista.h"
#include <iostream>

using namespace std;

int main(){

    ListaEnteros lint1(10);

    lint1.adjuntar(12);
    lint1.adjuntar(7);
    lint1.adjuntar(19);
    lint1.imprimirLista();

    lint1.adjuntar(30);
    lint1.imprimirLista();

    lint1.insertar(1, 41);
    lint1.imprimirLista();

    lint1.remover(2);
    lint1.imprimirLista();

    return 0;
}

```

```

| 12 | 7 | 19 | | |
| 12 | 7 | 19 | 30 |
| 12 | 41 | 7 | 19 | 30 |
| 12 | 41 | 19 | 30 |

```

-----  
Process exited after 0.08746 seconds with return value 0  
Presione una tecla para continuar . . .

```

#include <iostream>
#include <string>
#include "pila.hpp"

using std::string;
using std::cin;
using std::cout;
using std::endl;

bool estanLlavesBalanceadas(string expresion){
    Pila<char> unaPila;
    for(size_t i = 0; i < expresion.length(); i++){
        char caracter = expresion[i];
        cout << caracter << endl;
        if(caracter == '{'){
            unaPila.push(caracter);
            cout << unaPila.pilaComoCadena() << endl;
        } else if( caracter == ' '){
            if(unaPila.estaPilaVacia()){
                return false;
            }
            unaPila.pop();
            cout << unaPila.pilaComoCadena() << endl;
        }
    }
    return unaPila.estaPilaVacia();
}

int main(){

```

```

Ingrese una expresion:
hola
Creando una lista vacia...
La lista tiene capacidad para 4 elementos
Creando una lista vacia...
La lista tiene capacidad para 4 elementos
Destruyendo la lista...
h
o
l
a
Comprobando si la lista esta vacia...
Destruyendo la lista...
Las llaves estan balanceadas.
-----
Process exited after 2.923 seconds with return value 0
Presione una tecla para continuar . . . _

```

```

#include <iostream>
#include <array>

using namespace std;

int main(int argc, char *argv[]){
    array<int, 3> a = {8, 9, 10};

    cout << a.size() << endl;
    cout << a.front() << endl;
    cout << a.back() << endl;

    cout << a[0] << a[1] << a[2] << endl;

    return 0;
}

```

```

3
8
10
8910

```

```

-----
Process exited after 0.0695 seconds with return value 0
Presione una tecla para continuar . . . _

```