

Actividad 07 – Arreglo Dinámico

David Madrid Nápoles

Estructura de datos I

Lineamientos de evaluación

- El programa corre sin errores.
- Se creo la clase arregloDinamico.
- En la clase arregloDinamico se implementaron los metodos, agregar_inicio, agregar_final, size.
- Se implementó el uso de operador de sobrecarga.
- Se implementó el destructor para borrar la memoria.

Desarrollo

Programa principal (salida main.exe)

- Insertar final

```
madri@PCerda MINGW64 ~/OneDrive/Escritorio/Escuela/Boites/actividad07
$ g++ *.cpp -std=c++11 -o main.exe

madri@PCerda MINGW64 ~/OneDrive/Escritorio/Escuela/Boites/actividad07
$ ./main.exe
1 2 3 4 5 6 7 8
madri@PCerda MINGW64 ~/OneDrive/Escritorio/Escuela/Boites/actividad07
$ |
```

- Insertar inicio

```
madri@PCerda MINGW64 ~/OneDrive/Escritorio/Escuela/Boites/actividad07
$ g++ *.cpp -std=c++11 -o main.exe

madri@PCerda MINGW64 ~/OneDrive/Escritorio/Escuela/Boites/actividad07
$ ./main.exe
-1 0 1 2 3 4 5 6 7 8
madri@PCerda MINGW64 ~/OneDrive/Escritorio/Escuela/Boites/actividad07
```

Conclusiones

Me pareció muy útil el uso de arreglos dinámicos, se me hizo muy útil y muy practico que para aumentar el tamaño de este se pueda crear uno nuevo y pasarle la información del anterior.

Con esta modificación a los arreglos les da muchas posibilidades más, y se les pueden dar muchas más aplicaciones.

Con el video fui fácil aplicar los mismos arreglos dinámicos, pero para strings.

Referencias

<https://www.youtube.com/watch?v=tozHXqXI3Js>, Memoria dinámica, Michel Davalos Boites.

Código

```
//main.cpp

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

using namespace std;

int main () {
    ArregloDinamico arr;

    arr.insertar_final("1");
    arr.insertar_final("2");
    arr.insertar_final("3");
    arr.insertar_final("4");
    arr.insertar_final("5");
    arr.insertar_final("6");
    arr.insertar_final("7");
    arr.insertar_final("8");

    arr.insertar_inicio("0");
    arr.insertar_inicio("-1");

    for(size_t i = 0; i < arr.size();i++){
        cout << arr[i] << " ";
    }

    return 0;
}

//arregloDinamico.h

#ifndef ARREGLODINAMICO_H
#define ARREGLODINAMICO_H

#include <iostream>
```

```

using namespace std;

class ArregloDinamico
{
private:
    string *arreglo;
    size_t tam;
    size_t cont;
    const static size_t MAX = 5;

    void expandir();
public:
    ArregloDinamico(/* args */);
    ~ArregloDinamico();

    void insertar_final(const string &v);
    void insertar_inicio(const string &v);

    size_t size();

    string operator[](size_t p) {
        return arreglo[p];
    }
};

#endif

//arregloDinamico.cpp

#include "arregloDinamico.h"
#include <iostream>
using namespace std;

ArregloDinamico::ArregloDinamico(/* args */)
{
    arreglo = new string[MAX];
    cont = 0;
    tam = MAX;
}

```

```

}
ArregloDinamico::~ArregloDinamico(/* args */)
{
    delete[] arreglo;
}

void ArregloDinamico::insertar_final(const string &v) {
    if(cont == tam){
        //Expandir el arreglo
        expandir();
    }
    arreglo[cont] = v;
    cont++;
}

void ArregloDinamico::insertar_inicio(const string &v) {
    if(cont == tam){
        //Expandir el arreglo
        expandir();
    }
    for(size_t i = cont; i > 0; i--){
        arreglo[i] = arreglo[i-1];
    }
    arreglo[0] = v;
    cont++;
}

void ArregloDinamico::expandir() {
    string *nuevo = new string[tam+MAX];

    for(size_t i = 0; i < cont; i++){
        nuevo[i] = arreglo[i];
    }

    delete[] arreglo;
    arreglo = nuevo;
    tam = tam + MAX;
}

```

```
size_t ArregloDinamico::size() {  
    return cont;  
}
```