**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 see 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

Text

Description automatically generated

* Insertar inicio

Text

Description automatically generated

# 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;

}