

# UNIVERSIDAD AUTÓNOMA DE SANTO DOMINGO



**Nombre: Luis Miguel Sánchez Cabreja**

**Matrícula: 100613071**

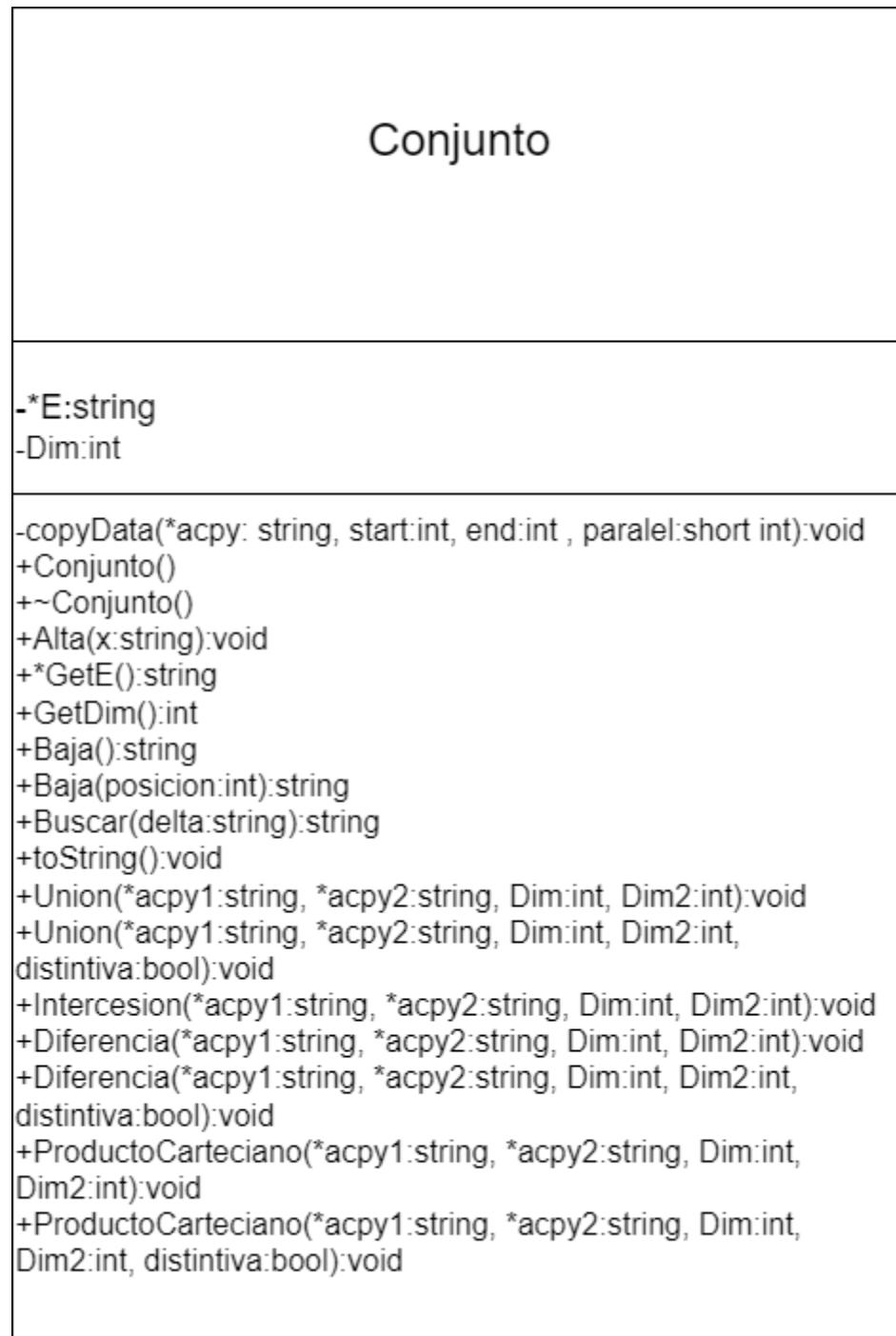
**Asignación: Creación de TAD**

**Clase: Estructura de Datos**

**Profesor: Silverio del Orbe Abad**

**Sección: Z02**

## Diagrama de clase



## Unit Testing de la Interface

```
int main(){//Unit Testing

    Conjunto con1, con2 , con3;

    string elemento = "";

    cout<<"\tCONJUNTO 1"<<endl;

    cout<<"\nIntroduzca su elemento(Para terminar entre ENTER): ";

        getline(cin,elemento);

    con1.Alta(elemento);

    while(elemento!=""){

        cout<<"\nIntroduzca su elemento(Para terminar entre ENTER): ";

        getline(cin,elemento);

        con1.Alta(elemento);

    }

    cout<<"\n";

        cout<<"\tCONJUNTO 2"<<endl;

    elemento = "";

    cout<<"\nIntroduzca su elemento(Para terminar entre ENTER): ";

        getline(cin,elemento);

        con2.Alta(elemento);

    while(elemento!=""){

        cout<<"\nIntroduzca su elemento(Para terminar entre ENTER): ";

        getline(cin,elemento);

        con2.Alta(elemento);

    }

    cout<<"\nConjunto 1: ";con1.toString();cout<<endl;

        cout<<"Conjunto 2: ";con2.toString();cout<<endl;

    system("pause");
```

```

while(elemento!="0"){
    system("cls");
string *Conjunto1 = con1.GetE();
    string *Conjunto2 = con2.GetE();
    int Dim1 = con1.GetDim();
    int Dim2 = con2.GetDim();
    cout<<"\n";
    cout<<"*****INTERFAZ*****"<<endl;
    cout<<"Conjunto 1: ";con1.toString();cout<<endl;
    cout<<"Conjunto 2: ";con2.toString();cout<<endl;
    cout<<"\n0.Salir \n1.Alta \n2.Baja \n3.Buscar elemento \n4.Union \n5.Intercesion
\n6.Diferencia \n7.Producto"<<endl;
    cout<<"Que desea hacer?";getline(cin,elemento);

    if(elemento=="0"){break;//Salida//
    }
    else if(elemento=="1"){//Alta//
        cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
        cout<<"Conjunto 2: ";con2.toString();cout<<endl;
        cout<<"\nCual conjunto quiere manejar?";
        getline(cin,elemento);
        if(elemento=="1"){//Alta Conjunto 1
            cout<<"\nIntroduzca su elemento(Para terminar introduzca ENTER ): ";
            getline(cin,elemento);
            con1.Alta(elemento);
            cout<<"\nNuevo Conjunto 1: ";con1.toString();cout<<endl;
            getline(cin,elemento);
        }
        else if(elemento=="2"){//Alta Conjunto 2

```

```

        cout<<"\nIntroduzca su elemento(Para terminar introduzca ENTER ): ";

        getline(cin,elemento);

        con2.Alta(elemento);

        cout<<"\nNuevo Conjunto 2: ";con2.toString();cout<<endl;

        getline(cin,elemento); }

    }

    else if(elemento=="2"){//Baja//

        cout<<"\nConjunto 1: ";con1.toString();cout<<endl;

        cout<<"Conjunto 2: ";con2.toString();cout<<endl;

        cout<<"\nCual conjunto quiere manejar?";

        getline(cin,elemento);

        int posicion;

        if(elemento=="1"){//Baja Conjunto 1//

            cout<<"\nConjunto 1: ";con1.toString();cout<<endl;

            cout<<"\nComo quiere dar de baja? "<<endl;

            cout<<"1. Ultimo elemento \n2. Por posicion"<<endl;

            getline(cin,elemento);

            if(elemento=="1"){//Baja Conjunto 1 Ultima Posicion

                cout<<"Elemento Bajado: "<<con1.Baja()<<endl;

                getline(cin,elemento);

            }

            else if(elemento=="2"){//Baja Conjunto 1 Por Posicion

                cout<<"\nConjunto 1: ";con1.toString();cout<<endl;

                cout<<"\nEn que posicion quiere eliminar? ";cin>>posicion;

                cout<<"Elemento Bajado: "<<con1.Baja(posicion)<<endl;

                system("pause");

```

```

        }
    }
    else if(elemento=="2"){//Baja Conjunto 2

        cout<<"\nConjunto 2: ";con2.toString();cout<<endl;
        cout<<"\nComo quiere dar de baja? "<<endl;
        cout<<"1. Ultimo elemento \n2. Por posicion"<<endl;
        getline(cin,elemento);
        if(elemento=="1"){//Baja Conjunto 2 Ultima Posicion
            cout<<"Elemento Bajado: "<<con2.Baja()<<endl;
            getline(cin,elemento);
        }
        else if(elemento=="2"){//Baja Conjunto 2 Por Posicion

            cout<<"\nConjunto 2: ";con2.toString();cout<<endl;
            cout<<"\nEn que posicion quiere eliminar? ";cin>>posicion;
            cout<<"Elemento Bajado: "<<con2.Baja(posicion)<<endl;
            system("pause");
        }
    }
}

else if(elemento=="3"){//Buscar

    cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
    cout<<"Conjunto 2: ";con2.toString();cout<<endl;
    cout<<"\nCual conjunto quiere manejar?";
    getline(cin,elemento);
    if (elemento=="1"){//Buscar en Conjunto 1

```

```

        cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
        cout<<"\nDigite que elemento quiere buscar: ";
        getline(cin,elemento);
        cout<<"Elemento buscado: "<<con1.Buscar(elemento)<<endl;
        getline(cin,elemento);
    }

    else if (elemento=="2"){//Buscar en Conjunto 2

        cout<<"\nConjunto 2: ";con2.toString();cout<<endl;
        cout<<"\nDigite que elemento quiere buscar: ";
        getline(cin,elemento);
        cout<<"Elemento buscado: "<<con2.Buscar(elemento)<<endl;
        getline(cin,elemento);
    }

}

else if(elemento=="4"){//Union
    cout<<"Como quiere unir los conjuntos?"<<endl;
    cout<<"\n1.Union normal \n2.Union distintiva"<<endl;
    getline(cin,elemento);
    if(elemento=="1"){//Union Normal
        cout<<"\nConjunto Unido 1-2: ";con3.Union(Conjunto1,Conjunto2,Dim1,Dim2);
        getline(cin,elemento);
    }

    else if(elemento=="2"){//Union Distintiva
        cout<<"\nConjunto Unido distintivamente 1-2:
";con3.Union(Conjunto1,Conjunto2,Dim1,Dim2,true);
        getline(cin,elemento);
    }
}

```

```

    }
}
else if(elemento=="5"){//Intercesion
    cout<<"\nConjunto Intercedido: ";con3.Intercesion(Conjunto1,Conjunto2,Dim1,Dim2);
    getline(cin,elemento);
}
else if(elemento=="6"){//Diferencia
    cout<<"Que conjunto quiere diferenciar de cual?"<<endl;
    cout<<"\n1.Diferenciar Conjunto 1 de 2 \n2.Diferenciar Conjunto 2 de 1"<<endl;
    getline(cin,elemento);
    if(elemento=="1"){//Conjunto 1 diferenciado de 2
        cout<<"\nConjunto 1 diferenciado de 2: ";con3.Diferencia(Conjunto1,Conjunto2,Dim1,Dim2);
        getline(cin,elemento);
    }
    else if(elemento=="2"){//Conjunto 2 diferenciado de 1
        cout<<"\nConjunto 2 diferenciado de 1: ";con3.Diferencia(Conjunto1,Conjunto2,Dim1,Dim2,true);
        getline(cin,elemento);
    }
}
else if(elemento=="7"){//Producto
    cout<<"\nQue tipo de Producto quiere?";
    cout<<"\n1.Producto Carteciano \n2.Producto Natural"<<endl;
    getline(cin,elemento);
    if(elemento=="1"){//Producto Carteciano
        cout<<"\nQue operacion quiere realizar?";
        cout<<"\n1.Conjunto 1 X Conjunto 2 \n2.Conjunto 2 X Conjunto 1"<<endl;
        getline(cin,elemento);
        if(elemento=="1"){

```



```

        cout<<"Producto Carteciano (Conjunto 1 X Conjunto 2):
";con3.ProductoCarteciano(Conjunto1,Conjunto2,Dim1,Dim2);

        getline(cin,elemento);

        }

        else if(elemento=="2"){

                cout<<"Producto Carteciano (Conjunto 2 X Conjunto 1):
";con3.ProductoCarteciano(Conjunto1,Conjunto2,Dim1,Dim2,true);

                getline(cin,elemento);

        }

        }

        else if(elemento=="2"){

                cout<<"Producto Natural (Conjunto 1 X Conjunto 2): ";

                getline(cin,elemento);

        }

        }

        }

}

```

## Código Fuente

```

#include <iostream>

#include <string>

#include "Conjunto.h"

using namespace std;

Conjunto::Conjunto(){

E = NULL;

this->Dim = 0;

}

Conjunto::~~Conjunto(){

delete E;

```

```

}

string* Conjunto::GetE(){
return E;
}

int Conjunto::GetDim(){
return Dim;
}

void Conjunto::Alta(string x){
// por cada alta hay que volver a resim el E
if(x!=""){
string *acopy = E;
Dim++;
this->E = new string[Dim];
copyData(acopy, 0,Dim-1, 0);
this->E[Dim - 1] = x; // nueva alta en posicion n-1
}
else;
}

void Conjunto::copyData(string *acpy, int start, int end, short int paralel){
for (int k = start; k < end; k++)
this->E[k] = acpy[k+paralel];
}

string Conjunto::Baja(){
// por cada alta hay que volver a resim el E
string *acopy = E;
string bajado = acopy[Dim - 1]; // retira el ultimo
Dim--;
this->E = new string[Dim];
copyData(acopy, 0, Dim,0);

```

```

return bajado;
}

string Conjunto::Baja(int posicion){
// por cada alta hay que volver a resim el E
if (posicion < 0 || posicion > Dim - 1)
return "";

string *acopy = E;
string bajado = acopy[posicion];
Dim--;

this->E = new string[Dim];
copyData(acopy, 0, posicion, 0);
copyData(acopy, posicion, Dim, 1);
return bajado;
}

```

```

string Conjunto::Buscar(string delta){
// por cada alta hay que volver a resim el E
string buscado = "";
for (int k = 0; k < Dim; k++)
if (E[k] == delta){
buscado = E[k];
break;
}
return buscado;
}

```

```

void Conjunto::Union(string *acpy1, string *acpy2, int Dim, int Dim2){//Union Normal
    cout<<"[";
    for (int i = 0; i < Dim; i++)

```

```

{
    cout << acpy1[i] << ",";
}

for (int i = 0; i < Dim2; i++)

{
    cout << acpy2[i] << ",";
}
cout<<"]";
}

void Conjunto::Union(string *acpy1,string *acpy2,int Dim, int Dim2, bool distintiva){//Union Distintiva
    cout<<"[";
    for(int l=0;l<Dim;l++){
        cout<<acpy1[l]<<" ";
    }
    for(int k=0;k<Dim2;k++){
        int contador=0;
        for(int i=0;i<Dim;i++){
            if(acpy1[i]==acpy2[k]){
                contador++;
            }

        }

        if(contador==0){
            if(k==Dim2-1){
                cout<<acpy2[k];
                break;
            }

```

```

        cout<<acpy2[k]<<" ";
    }
    else if(contador>0){

    }

}

cout<<"]"<<endl;
}

void Conjunto::toString() {

    cout<<"[";

    for (int i = 0; i < Dim; i++)

    {
        cout << E[i] <<" ";
    }
    cout<<"]";
}

void Conjunto::Intercesion(string *acpy1,string *acpy2,int Dim, int Dim2){

    cout<<"[";

    for(int k=0;k<Dim2;k++){
        for(int i=0;i<Dim;i++){
            if(acpy1[i]==acpy2[k]){
                cout<<acpy2[k]<<" ";
            }
        }
    }
}

```

```

        }
    }
    cout<<"]"<<endl;

}

void Conjunto::Diferencia(string *acpy1,string *acpy2,int Dim, int Dim2){//Diferencia Conjunto 1 de
Conjunto 2
cout<<"[";

    bool roptura = false;
    for(int k=0;k<Dim;k++){
        roptura = false;
        for(int i=0;i<Dim2;i++){

            if(acpy1[k]==acpy2[i]){
                roptura = true;
                break;
            }
        }
        if(roptura!=true){
            cout<<acpy1[k]<<" ";
        }
    }

    cout<<"]"<<endl;
}

```

```

void Conjunto::Diferencia(string *acpy1,string *acpy2,int Dim, int Dim2, bool distintiva){//Diferencia
Conjunto 2 de Conjunto 1
cout<<"[";

```

```

bool roptura = false;

for(int k=0;k<Dim2;k++){

    roptura = false;

    for(int i=0;i<Dim;i++){

        if(acpy2[k]==acpy1[i]){

            roptura = true;

            break;

        }

    }

    if(roptura!=true){

        cout<<acpy2[k]<<" ";

    }

}

cout<<"]"<<endl;

}

void Conjunto::ProductoCarteciano(string *acpy1,string *acpy2, int Dim, int Dim2){//Conjunto1 x
Conjunto2

    cout<<"[";

    for(int k=0;k<Dim;k++){

        for(int i=0;i<Dim2;i++){

            cout<<"("<<acpy1[k]<<","<<acpy2[i]<<"),";

        }

    }

    cout<<"]"<<endl;

}

```

```
void Conjunto::ProductoCarteciano(string *acpy1,string *acpy2, int Dim, int Dim2,bool distintiva){  
    cout<<"[";  
  
    for(int k=0;k<Dim2;k++){  
        for(int i=0;i<Dim;i++){  
            cout<<"("<<acpy2[k]<<" "<<acpy1[i]<<" "<<acpy1[i]<<" "<<acpy2[k]<<")";  
        }  
    }  
    cout<<"]"<<endl;  
}
```



## Creación de Conjuntos

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

CONJUNTO 1

Introduzca su elemento(Para terminar entre ENTER): Jugo
Introduzca su elemento(Para terminar entre ENTER): Pera
Introduzca su elemento(Para terminar entre ENTER): Lima
Introduzca su elemento(Para terminar entre ENTER): Uva
Introduzca su elemento(Para terminar entre ENTER):

CONJUNTO 2

Introduzca su elemento(Para terminar entre ENTER ): Carro
Introduzca su elemento(Para terminar entre ENTER ): Motor
Introduzca su elemento(Para terminar entre ENTER ): Muffler
Introduzca su elemento(Para terminar entre ENTER ): Capó
Introduzca su elemento(Para terminar entre ENTER ):

Conjunto 1: [Jugo Pera Lima Uva ]
Conjunto 2: [Carro Motor Muffler Capó ]
Presione una tecla para continuar . . .
```

## Interfaz

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima Uva ]
Conjunto 2: [Carro Motor Muffler Capó ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?
```

## Alta

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima Uva ]
Conjunto 2: [Carro Motor Muffler Capó ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?1

Conjunto 1: [Jugo Pera Lima Uva ]
Conjunto 2: [Carro Motor Muffler Capó ]

Cual conjunto quiere manejar?1

Introduzca su elemento(Para terminar introduzca ENTER ): Zapote

Nuevo Conjunto 1: [Jugo Pera Lima Uva Zapote ]
```

## Baja (Ultima Posición)

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima Uva Zapote ]
Conjunto 2: [Carro Motor Muffler Capó ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?2

Conjunto 1: [Jugo Pera Lima Uva Zapote ]
Conjunto 2: [Carro Motor Muffler Capó ]

Cual conjunto quiere manejar?2

Conjunto 2: [Carro Motor Muffler Capó ]

Como quiere dar de baja?
1. Ultimo elemento
2. Por posicion
1
Elemento Bajado: Capó
```

## Baja (Por posición)

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe
Conjunto 1: [Jugo Pera Lima Uva Zapote ]
Conjunto 2: [Carro Motor Muffler ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?2

Conjunto 1: [Jugo Pera Lima Uva Zapote ]
Conjunto 2: [Carro Motor Muffler ]

Cual conjunto quiere manejar?1

Conjunto 1: [Jugo Pera Lima Uva Zapote ]

Como quiere dar de baja?
1. Ultimo elemento
2. Por posicion
2

Conjunto 1: [Jugo Pera Lima Uva Zapote ]

En que posicion quiere eliminar? 3
Elemento Bajado: Uva
Presione una tecla para continuar . . .
```

## Buscar

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima Zapote ]
Conjunto 2: [Carro Motor Muffler ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?3

Conjunto 1: [Jugo Pera Lima Zapote ]
Conjunto 2: [Carro Motor Muffler ]

Cual conjunto quiere manejar?2

Conjunto 2: [Carro Motor Muffler ]

Digite que elemento quiere buscar: Muffler
Elemento buscado: Muffler
```

## Unión normal

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima Zapote Motor ]
Conjunto 2: [Carro Motor Muffler Lima Capó ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?4
Como quiere unir los conjuntos?

1.Union normal
2.Union distintiva
1

Conjunto Unido 1-2: [Jugo,Pera,Lima,Zapote,Motor,Carro,Motor,Muffler,Lima,Capó,]
```

## Unión Distintiva

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima Zapote Motor ]
Conjunto 2: [Carro Motor Muffler Lima Capó ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?4
Como quiere unir los conjuntos?

1.Union normal
2.Union distintiva
2

Conjunto Unido distintivamente 1-2: [Jugo Pera Lima Zapote Motor Carro Muffler Capó]
```

## Intercesión

C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

```
*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima Zapote Motor ]
Conjunto 2: [Carro Motor Muffler Lima Capó ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?5

Conjunto Intercedido: [Motor Lima ]
```

## Diferencia

C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

```
*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima Zapote Motor ]
Conjunto 2: [Carro Motor Muffler Lima Capó ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?6
Que conjunto quiere diferenciar de cual?

1.Diferenciar Conjunto 1 de 2
2.Diferenciar Conjunto 2 de 1
1

Conjunto 1 diferenciado de 2: [Jugo Pera Zapote ]
```

## Producto Carteciano

```
C:\Users\User\Documents\Semestre 2023-1\Estructura de Datos\Programas\Unidad 1\Conjunto.exe

*****INTERFAZ*****
Conjunto 1: [Jugo Pera Lima ]
Conjunto 2: [Carro Motor Muffler ]

0.Salir
1.Alta
2.Baja
3.Buscar elemento
4.Union
5.Intercesion
6.Diferencia
7.Producto
Que desea hacer?7

Que tipo de Producto quiere?
1.Producto Carteciano
2.Producto Natural
1

Que operacion quiere realizar?
1.Conjunto 1 X Conjunto 2
2.Conjunto 2 X Conjunto 1
2

Producto Carteciano (Conjunto 2 X Conjunto 1): [(Carro,Jugo),(Carro,Pera),(Carro,Lima),(Motor,Jugo),(Motor,Pera),(Motor,Lima),(Muffler,Jugo),(Muffler,Pera),(Muffler,Lima),]
```

## **Conjunto.h(Interface)**

```
#include <iostream>

#include <string>

using namespace std;

class Conjunto{

private:

    string *E;

    int Dim; /*Cantidad*/

    void copyData(string *acpy, int start, int end, short int paralel);

public:

    Conjunto();

    ~Conjunto();

    void Alta(string x);

    string* GetE();

    int GetDim();

    string Baja();

    //overload 1 de baja

    string Baja(int posicion);

    string Buscar(string delta);

    void toString();

    void Union(string *acpy1,string *acpy2,int Dim, int Dim2);//Union normal

    void Union(string *acpy1,string *acpy2,int Dim, int Dim2, bool distintiva);//Union Distintiva

    void Intercesion(string *acpy1,string *acpy2,int Dim, int Dim2);

    void Diferencia(string *acpy1,string *acpy2,int Dim, int Dim2);//Diferencia Conjunto 1 de Conjunto 2

    void Diferencia(string *acpy1,string *acpy2,int Dim, int Dim2, bool distintiva);//Diferencia Conjunto 2 de Conjunto 1

    void ProductoCarteciano(string *acpy1,string *acpy2,int Dim, int Dim2);//Conjunto 1 X Conjunto2
```

```
void ProductoCarteciano(string *acpy1,string *acpy2,int Dim, int Dim2, bool distintiva);//Conjunto 1 X
Conjunto2
};
```

### **Conjunto.cpp(Method Implement)**

```
#include <iostream>

#include <string>

#include "Conjunto.h"

using namespace std;

Conjunto::Conjunto(){
    E = NULL;
    this->Dim = 0;
}

Conjunto::~~Conjunto(){
    delete E;
}

string* Conjunto::GetE(){
    return E;
}

int Conjunto::GetDim(){
    return Dim;
}

void Conjunto::Alta(string x){
    // por cada alta hay que volver a resim el E
    if(x!=""){
        string *acopy = E;
        Dim++;
        this->E = new string[Dim];
        copyData(acopy, 0,Dim-1, 0);
        this->E[Dim - 1] = x; // nueva alta en posicion n-1
```



```

}
else;
}

void Conjunto::copyData(string *acpy, int start, int end, short int paralel){
for (int k = start; k < end; k++)
this->E[k] = acpy[k+paralel];
}

string Conjunto::Baja(){
// por cada alta hay que volver a resim el E
string *acopy = E;
string bajado = acopy[Dim - 1]; // retira el ultimo
Dim--;
this->E = new string[Dim];
copyData(acopy, 0, Dim,0);
return bajado;
}

string Conjunto::Baja(int posicion){
// por cada alta hay que volver a resim el E
if (posicion < 0 || posicion > Dim - 1)
return "";
string *acopy = E;
string bajado = acopy[posicion];
Dim--;
this->E = new string[Dim];
copyData(acopy, 0, posicion, 0);
copyData(acopy, posicion, Dim, 1);
return bajado;
}

```

```

string Conjunto::Buscar(string delta){
// por cada alta hay que volver a resim el E
string buscado = "";
for (int k = 0; k < Dim; k++)
if (E[k] == delta){
buscado = E[k];
break;
}
return buscado;
}

void Conjunto::Union(string *acpy1,string *acpy2, int Dim, int Dim2){//Union Normal
    cout<<"[";
    for (int i = 0; i < Dim; i++)

    {
        cout << acpy1[i] <<" ";
    }

    for (int i = 0; i < Dim2; i++)

    {
        cout << acpy2[i] <<" ";
    }
    cout<<"]";
}

void Conjunto::Union(string *acpy1,string *acpy2,int Dim, int Dim2, bool distintiva){//Union Distintiva
    cout<<"[";
    for(int l=0;l<Dim;l++){
        cout<<acpy1[l]<<" ";
    }

```

```

    }
    for(int k=0;k<Dim2;k++){
        int contador=0;
        for(int i=0;i<Dim;i++){
            if(acpy1[i]==acpy2[k]){
                contador++;
            }

        }

        if(contador==0){
            if(k==Dim2-1){
                cout<<acpy2[k];
                break;
            }
            cout<<acpy2[k]<<" ";
        }
        else if(contador>0){

        }

    }
    cout<<"]"<<endl;
}

void Conjunto::toString() {

    cout<<"[";

    for (int i = 0; i < Dim; i++)

    {

```

```

        cout << E[i] << " ";
    }
    cout<<"]";
}

```

```

void Conjunto::Intercesion(string *acpy1,string *acpy2,int Dim, int Dim2){

```

```

    cout<<"[";

```

```

        for(int k=0;k<Dim2;k++){
            for(int i=0;i<Dim;i++){
                if(acpy1[i]==acpy2[k]){
                    cout<<acpy2[k]<<" ";
                }
            }
        }
        cout<<"]"<<endl;

```

```

}

```

```

void Conjunto::Diferencia(string *acpy1,string *acpy2,int Dim, int Dim2){//Diferencia Conjunto 1 de
Conjunto 2

```

```

    cout<<"[";

```

```

    bool roptura = false;

```

```

        for(int k=0;k<Dim;k++){
            roptura = false;
            for(int i=0;i<Dim2;i++){

                if(acpy1[k]==acpy2[i]){
                    roptura = true;

```

```

                break;
            }
        }
        if(roptura!=true){
            cout<<acpy1[k]<<" ";
        }
    }
    cout<<"]"<<endl;
}

```

void Conjunto::Diferencia(string \*acpy1,string \*acpy2,int Dim, int Dim2, bool distintiva){//Diferencia  
Conjunto 2 de Conjunto 1

```

cout<<"[";
    bool roptura = false;
    for(int k=0;k<Dim2;k++){
        roptura = false;
        for(int i=0;i<Dim;i++){

            if(acpy2[k]==acpy1[i]){
                roptura = true;
                break;
            }
        }
        if(roptura!=true){
            cout<<acpy2[k]<<" ";
        }
    }
}

```

```

        cout<<"]"<<endl;

    }

    void Conjunto::ProductoCarteciano(string *acpy1,string *acpy2, int Dim, int Dim2){//Conjunto1 x
    Conjunto2

        cout<<"[";

        for(int k=0;k<Dim;k++){

            for(int i=0;i<Dim2;i++){

                cout<<"("<<acpy1[k]<<","<<acpy2[i]<<"),";

            }

        }

        cout<<"]"<<endl;

    }

    void Conjunto::ProductoCarteciano(string *acpy1,string *acpy2, int Dim, int Dim2,bool distintiva){

        cout<<"[";

        for(int k=0;k<Dim2;k++){

            for(int i=0;i<Dim;i++){

                cout<<"("<<acpy2[k]<<","<<acpy1[i]<<"),";

            }

        }

        cout<<"]"<<endl;

    }

    int main(){//Unit Testing

        Conjunto con1, con2 , con3;

        string elemento = "";

        cout<<"\tCONJUNTO 1"<<endl;

```

```

cout<<"\nIntroduzca su elemento(Para terminar entre ENTER): ";
    getline(cin,elemento);
con1.Alta(elemento);
while(elemento!=""){
    cout<<"\nIntroduzca su elemento(Para terminar entre ENTER): ";
    getline(cin,elemento);
con1.Alta(elemento);
}

cout<<"\n";
    cout<<"\tCONJUNTO 2"<<endl;
elemento = "";
cout<<"\nIntroduzca su elemento(Para terminar entre ENTER ): ";
    getline(cin,elemento);
    con2.Alta(elemento);
while(elemento!=""){
    cout<<"\nIntroduzca su elemento(Para terminar entre ENTER ): ";
    getline(cin,elemento);
    con2.Alta(elemento);
}

cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
    cout<<"Conjunto 2: ";con2.toString();cout<<endl;
    system("pause");

while(elemento!="0"){
    system("cls");
string *Conjunto1 = con1.GetE();
    string *Conjunto2 = con2.GetE();
    int Dim1 = con1.GetDim();

```

```

int Dim2 = con2.GetDim();

    cout<<"\n";

cout<<"*****INTERFAZ*****"<<endl;

    cout<<"Conjunto 1: ";con1.toString();cout<<endl;

    cout<<"Conjunto 2: ";con2.toString();cout<<endl;

        cout<<"\n0.Salir \n1.Alta \n2.Baja \n3.Buscar elemento \n4.Union \n5.Intercesion
\n6.Diferencia \n7.Producto"<<endl;

        cout<<"Que desea hacer?";getline(cin,elemento);


if(elemento=="0"){break;//Salida//
    }

else if(elemento=="1"){//Alta//

        cout<<"\nConjunto 1: ";con1.toString();cout<<endl;

        cout<<"Conjunto 2: ";con2.toString();cout<<endl;

        cout<<"\nCual conjunto quiere manejar?";

        getline(cin,elemento);

        if(elemento=="1"){//Alta Conjunto 1

            cout<<"\nIntroduzca su elemento(Para terminar introduzca ENTER ): ";

            getline(cin,elemento);

            con1.Alta(elemento);

            cout<<"\nNuevo Conjunto 1: ";con1.toString();cout<<endl;

            getline(cin,elemento);

            }

            else if(elemento=="2"){//Alta Conjunto 2

                cout<<"\nIntroduzca su elemento(Para terminar introduzca ENTER ): ";

                getline(cin,elemento);

                con2.Alta(elemento);

                cout<<"\nNuevo Conjunto 2: ";con2.toString();cout<<endl;

                getline(cin,elemento); }

```



```

}

else if(elemento=="2"){//Baja//

    cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
cout<<"Conjunto 2: ";con2.toString();cout<<endl;

    cout<<"\nCual conjunto quiere manejar?";

    getline(cin,elemento);

    int posicion;

    if(elemento=="1"){//Baja Conjunto 1//

        cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
cout<<"\nComo quiere dar de baja? "<<endl;
cout<<"1. Ultimo elemento \n2. Por posicion"<<endl;

        getline(cin,elemento);

        if(elemento=="1"){//Baja Conjunto 1 Ultima Posicion

            cout<<"Elemento Bajado: "<<con1.Baja()<<endl;
            getline(cin,elemento);

        }

        else if(elemento=="2"){//Baja Conjunto 1 Por Posicion

            cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
cout<<"\nEn que posicion quiere eliminar? ";cin>>posicion;
cout<<"Elemento Bajado: "<<con1.Baja(posicion)<<endl;
system("pause");

        }

    }

    else if(elemento=="2"){//Baja Conjunto 2

        cout<<"\nConjunto 2: ";con2.toString();cout<<endl;

```

```

cout<<"\nComo quiere dar de baja? "<<endl;
cout<<"1. Ultimo elemento \n2. Por posicion"<<endl;

    getline(cin,elemento);

    if(elemento=="1"){//Baja Conjunto 2 Ultima Posicion
        cout<<"Elemento Bajado: "<<con2.Baja()<<endl;
        getline(cin,elemento);
    }

    else if(elemento=="2"){//Baja Conjunto 2 Por Posicion

        cout<<"\nConjunto 2: ";con2.toString();cout<<endl;

        cout<<"\nEn que posicion quiere eliminar? ";cin>>posicion;
        cout<<"Elemento Bajado: "<<con2.Baja(posicion)<<endl;
        system("pause");
    }

}

}

else if(elemento=="3"){//Buscar

    cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
    cout<<"Conjunto 2: ";con2.toString();cout<<endl;
    cout<<"\nCual conjunto quiere manejar?";
    getline(cin,elemento);

    if (elemento=="1"){//Buscar en Conjunto 1

        cout<<"\nConjunto 1: ";con1.toString();cout<<endl;
        cout<<"\nDigite que elemento quiere buscar: ";
        getline(cin,elemento);

        cout<<"Elemento buscado: "<<con1.Buscar(elemento)<<endl;
        getline(cin,elemento);
    }
}
}

```

```

    }

    else if (elemento=="2"){//Buscar en Conjunto 2

        cout<<"\nConjunto 2: ";con2.toString();cout<<endl;
        cout<<"\nDigite que elemento quiere buscar: ";
        getline(cin,elemento);
        cout<<"Elemento buscado: "<<con2.Buscar(elemento)<<endl;
        getline(cin,elemento);
    }

}

else if(elemento=="4"){//Union
    cout<<"Como quiere unir los conjuntos?"<<endl;
    cout<<"\n1.Union normal \n2.Union distintiva"<<endl;
    getline(cin,elemento);
    if(elemento=="1"){//Union Normal
        cout<<"\nConjunto Unido 1-2: ";con3.Union(Conjunto1,Conjunto2,Dim1,Dim2);
        getline(cin,elemento);
    }

    else if(elemento=="2"){//Union Distintiva
        cout<<"\nConjunto Unido distintivamente 1-2:
";con3.Union(Conjunto1,Conjunto2,Dim1,Dim2,true);
        getline(cin,elemento);
    }
}

}

else if(elemento=="5"){//Intercesion
    cout<<"\nConjunto Intercedido: ";con3.Intercesion(Conjunto1,Conjunto2,Dim1,Dim2);
    getline(cin,elemento);
}

```

```

    }

    else if(elemento=="6"){//Diferencia

        cout<<"Que conjunto quiere diferenciar de cual?"<<endl;

        cout<<"\n1.Diferenciar Conjunto 1 de 2 \n2.Diferenciar Conjunto 2 de 1"<<endl;

        getline(cin,elemento);

        if(elemento=="1"){//Conjunto 1 diferenciado de 2

            cout<<"\nConjunto 1 diferenciado de 2: ";con3.Diferencia(Conjunto1,Conjunto2,Dim1,Dim2);

            getline(cin,elemento);

        }

        else if(elemento=="2"){//Conjunto 2 diferenciado de 1

            cout<<"\nConjunto 2 diferenciado de 1:

";con3.Diferencia(Conjunto1,Conjunto2,Dim1,Dim2,true);

            getline(cin,elemento);

        }

    }

    else if(elemento=="7"){//Producto

        cout<<"\nQue tipo de Producto quiere?";

        cout<<"\n1.Producto Carteciano \n2.Producto Natural"<<endl;

        getline(cin,elemento);

        if(elemento=="1"){//Producto Carteciano

            cout<<"\nQue operacion quiere realizar?";

            cout<<"\n1.Conjunto 1 X Conjunto 2 \n2.Conjunto 2 X Conjunto 1"<<endl;

            getline(cin,elemento);

            if(elemento=="1"){

                cout<<"Producto Carteciano (Conjunto 1 X Conjunto 2):

";con3.ProductoCarteciano(Conjunto1,Conjunto2,Dim1,Dim2);

                getline(cin,elemento);

            }

            else if(elemento=="2"){

```

```
        cout<<"Producto Carteciano (Conjunto 2 X Conjunto 1):  
";con3.ProductoCarteciano(Conjunto1,Conjunto2,Dim1,Dim2,true);  
        getline(cin,elemento);  
    }  
}  
else if(elemento=="2"){  
    cout<<"Producto Natural (Conjunto 1 X Conjunto 2): ";  
    getline(cin,elemento);  
}  
}  
}  
}
```