LECCION 7
SEGUNDA PARTE

(1)

```
template <class T>          //FILE: VD.h
class VD{
  private:
    T* datos;
    int n;
    int reservados;
    void resize(int nuevo_tam);
    void Copiar(const VD<T> &V);
    void Liberar();
  public:
    VD(int tam=10);
    VD(const VD<T> &original);
    ~VD();
    VD<T> & operator=(const VD<T> &v);
    int size() const { return n; }
    T & operator[](int i) { return datos[i]; }
    const T & operator[](int i) const { return datos[i]; }
    void Insertar(const T &d, int pos);
    void Borrar(int pos);

};
#include "VD.cpp"
```

LECCION 7
segunda parte

(2)

```
//VD.cpp
template <class T>
void VD<T>::Borrar(int pos){

    for(int i=pos; i<n-1; i++)
        datos[i]=datos[i+1];

    n--;
    if (n < reservados/4)
        resize(reservados/2);

}
```

```
//Conjunto.h
#include <VD.h>
class Conjunto{
 private:
   VD<int> d;

 public:
   int size() const{return d.size();}
   bool Esta(int x)const;
   void Insertar(int x);
   void Borrar(int x);
};
```

EJEMPLO: T.D.A Conjunto de Enteros.

ESPECIFICACIÓN - Un objeto del tipo de dato conjunto de enteros es una coleccion ordenada de elementos de tipo de entero en la que no existen elementos repetidos

$\{a_0, a_1, \ldots, a_n\}$ $\forall i<j \rightarrow a_i < a_j$

Al n= de elementos del conjunto se le denomina cardinal. Si el cardinal es 0 - entonces el conjunto es el conjunto vacio.

TIPO_REP

```
class Conjunto{
   private:
     VD<int> d;
```

FUNCION de ABSTRACCION

$f_A(r) = \{r.d[0], r.d[1], \ldots, r.d[r.d.size()-1]\}$

INVARIANTE de la REP.

$\forall i,j \quad i<j \quad r.d[i] < r.d[j]$.

```
//Conjunto.cpp
#include "Conjunto.h" //no es template
int Conjunto::Esta(int x) const{
    int n= d.size();
    //Busqueda Binaria.

    int inicio=0, fin=n;
    while (inicio<fin){
      int mitad= (inicio+fin)/2;
      if (d[mitad]==x)
          return true; //return mitad;

      else
          if (d[mitad]<x)
              inicio=mitad+1;

          else
              if (d[mitad]>x)
                  fin=mitad;
    }return false; //return inicio;
}
```

③ LECCION 7
segunda parte

```
void Conjunto::Insertar (int x){
        .
        .
        .
    int pos = Esta(x);
    if (d[pos] == x)
            return;
    else
        d.Insertar(x, pos);

}
void Conjunto::Borrar(int x){
    int pos = Esta(x);
    if (d[pos] == x)
            d.Borrar(pos);
    else return;
}
```

3.