

*UNIVERSIDAD AUTONOMA GABRIEL RENE MORENO*

*FACULTAD DE INGENIERIA Y CIENCIAS DE LA  
COMPUTACION Y TELECOMUNICACIONES*

ESTRUCTURA DE DATOS II

CONTENIDO: Consultas Listas encadenadas simples

PORCENTAJE TERMINADO: 100%

Grupo 14

Garcia Taboada Brayan Albaro

**Fecha de presentación** : Martes , 26 de marzo de 2024

**Fecha Presentada** : Martes , 26 de marzo de 2024

**Días de Atraso** : 0

```

public class ListaEncadenada {

    public Nodo prim;

    public int cantElem;

    public Nodo ult;

    public ListaEncadenada(){

        prim=ult=null;

        cantElem=0;

    }

    public String toString(){

        String s1="[";

        Nodo p=prim;

        while(p!=null){

            s1+=p.elem;

            if(p.prox!=null){

                s1+=", ";

            }

            p=p.prox;

        }

        return s1+"]";

    }


    public Boolean vacia(){

        return prim==null && ult==null;

    }

    public void insertarPrim(int x){

        if(vacia()){

            prim=ult=new Nodo(x,null);

        } else {

            prim=new Nodo(x,prim);

        }

        cantElem++;

    }

}

```

```
}
```

```
public void insertarUlt(int x){  
    if(vacia()){  
        prim=ult=new Nodo(x,null);  
    } else {  
        ult=ult.prox=new Nodo(x,null);  
    }  
    cantElem++;  
}
```

```
public void insertarlesimo(int x,int i){  
    Nodo p=prim;  
    Nodo ap=null;  
    int k=0;  
    while(p!=null && i>k){  
        ap=p;  
        p=p.prox;  
        k++;  
    }  
    insertarNodo(x,ap,p);  
}
```

```
public Boolean iguales(){  
    Nodo p=prim;  
    while(p!=null ){  
        if(p.elem!=p.prox.elem){  
            return false;  
        }  
        p=p.prox;  
    }
```

```
    }  
    return true;  
}
```

```
public int menorElem(){  
    Nodo p=prim;  
    int menor = p.elem;  
    while(p!=null ){  
        p=p.prox;  
        if(menor>p.elem){  
            menor=p.elem;  
        }  
    }  
    return menor;  
}
```

```
public void reemplazar(int x,int y){  
    Nodo p=prim;  
    while(p!=null ){  
        if(p.elem==x){  
            p.elem=y;  
        }  
        p=p.prox;  
    }  
}
```

```
public Boolean seEncuentra(int x){  
    Nodo p=prim;  
    while(p!=null ){  
        if(p.elem==x){  
            return true;  
        }  
    }  
}
```

```

    }
    p=p.prox;
}
return false;
}

```

```

public int frecuencia(int x){
    Nodo p=prim;
    int frec=0;
    while(p!=null ){
        if(p.elem==x){
            frec++;
        }
        p=p.prox;
    }
    return frec;
}

```

```

public void insertarUlt(ListaEncadenada L2){
    Nodo p=L2.prim;
    while(p!=null ){
        insertarUlt(p.elem);
        p=p.prox;
    }
}

```

```

public void insertarLugar(int x){
    Nodo p=prim;
    Nodo ap=null;
    while(p!=null && x>p.elem){
        ap=p;
    }
}

```

```
        p=p.prox;
    }
    insertarNodo(x,ap,p);
}
```

```
public void insertarNodo(int x,Nodo ap,Nodo p){
    if(ap==null) insertarPrim(x);
    else
    if (p==null) insertarUlt(x);
    else{
        ap.prox=new Nodo(x,p);
        cantElem++;
    }
}
}
```

```
class Nodo {
    public int elem;
    public Nodo prox;
    public Nodo (int elem, Nodo prox) {
        this.elem=elem;
        this.prox=prox;
    }
}
```