

```
int eliminaryMostrarUnicos_MIO(tLista *p, FILE *fpPant,  
                               int comparar(const void *, const void *),
```

```

        void mostrar(const void *, FILE *)
    {
        tLista *q, *inicio = p;
        tNodo *aux;
        int iguales, cont=0;

        mostrar(NULL, fpPant);

        while(*p)
        {
            iguales = 0;
            q = inicio;
            while(*q)
            {
                if(!comparar((*p)->info, (*q)->info))
                    iguales++;

                q=&(*q)->sig;
            }
            if(iguales<2)
            {
                mostrar((*p)->info, fpPant);
                aux=*p;
                *p=aux->sig;
                free(aux);
                cont++;
            }
            else
                p=&(*p)->sig;
        }

        return cont;
    }

int eliminarYMostrarRepetidos_MIO(tLista *p, FILE *fpPant,
                                int comparar(const void *, const void *),
                                void mostrar(const void *, FILE *))
{
    tLista *q, *inicio = p;
    tNodo *aux;
    int f, cont=0;

    mostrar(NULL, fpPant);

    while(*p)
    {
        // la comparacion comienza a partir del siguiente
        // nodo donde esta parado el puntero hasta el final de la lista
        f=0;
        q=&(*p)->sig;

        while(*q)
        {
            // en caso de encontrar un nodo igual
            // muestra, borra y cambia al nodo que es igual
            // para asi seguir comparando
            if(!comparar((*p)->info, (*q)->info))
            {
                f=1;
                mostrar((*p)->info, fpPant);
                aux=*p;
                *p=aux->sig;
                free(aux);
                cont++;
                p=q;
            }
            q=&(*q)->sig;
        }
    }
}

```

```
}  
  
// si hay repetidos, al llegar al final de la lista  
// faltara mostrar uno, para eso es la bandera f.  
if(f)  
{  
    mostrar((*p)->info, fpPant);  
    aux=*p;  
    *p=aux->siga;  
    free(aux);  
    cont++;  
    p=inicio;  
}  
else  
    p=&(*p)->siga;  
  
return cont;  
  
}
```

/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*/\*\*