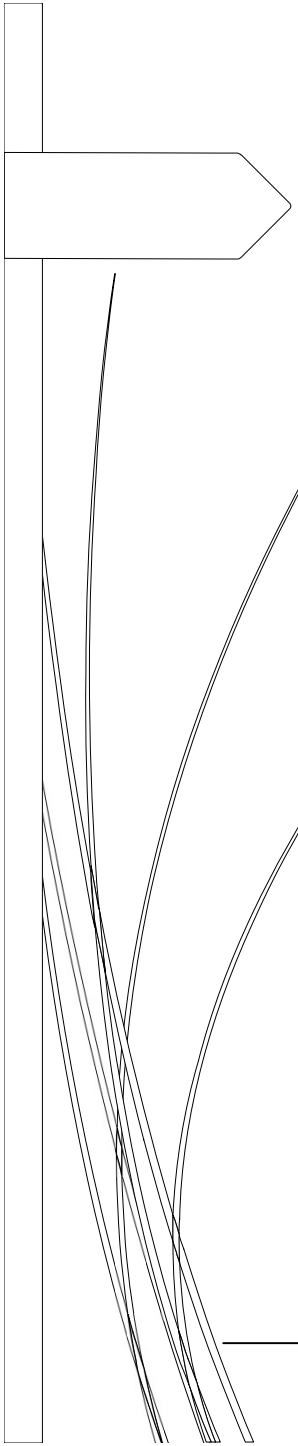

Listas Simplemente Enlazadas

A partir de “nodos” dinámicos, implementados como variables de tipo registro, en el que un campo de tipo puntero a registro contiene la dirección del próximo nodo, es posible implementar Pilas, Colas y Listas tomando como enlaces los vínculos en memoria dinámica.

Cada inserción implica crear y vincular nodos y cada eliminación desvincularlos devolviendo al heap los bytes que ocupa el nodo.

La forma en que se modifican los enlaces o vínculos depende de la forma de operar de la estructura.

No se considera la posibilidad de estructura llena, ya que ésta crece y decrece de acuerdo a los requerimientos.

Listas SE - Recorrido

Dada una lista simplemente enlazada de cadenas, retornar la cantidad que tienen longitud par

```
typedef struct nodo {
  char dato[15];
  struct nodo * sig;} nodo;
typedef struct nodo * TLista;

int CantPares (TLista L){
TLista aux;
int cont = 0;
aux = L;
while (aux != NULL) {
    if (!(strlen(aux-> dato) % 2))
       cont++;
    aux = aux -> sig;
}
return cont;
}
```

Listas SE - Búsqueda

Dada una lista simplemente enlazada de cadenas, verificar si X está

```
int esta (TLista L, char *x){
TLista aux;

aux = L;
while (aux != NULL && strcmp(x,aux->dato) !=0)
    aux = aux -> sig;
return aux != NULL;
```

Dada una lista simplemente enlazada ordenada de cadenas, verificar si X está

```
int estaOrd (TLista L, char *x){
TLista aux  = L;
while (aux != NULL && strcmp(x, aux->dato) > 0)
    aux = aux -> sig;

return aux != NULL && strcmp(x, aux->dato) == 0;
}
```

Listas SE - Inserción

Insertar un dato en una lista ordenada simplemente enlazada de enteros

```
void insertaOrd (TListaE * L, int x){
TListaE aux, ant, act;
aux = (TListaE) malloc (sizeof(nodo));
aux->dato= x;
if (*L == NULL || x < (*L)->dato) {
    aux->sig = *L;
    *L = aux;
}
else{
    ant=NULL;
    act=*L;
    while (act != NULL && x> act->dato){
        ant=act;
        act=act->sig;
    }
    aux->sig=act;
    ant->sig=aux;
}
}
```

Listas SE - Eliminación

Dada una lista ordenada simplemente enlazada de enteros, eliminar X

```
void eliminaE(TListaE *L, int x){
TListaE ant,act;
if (*L !=NULL)
  if ((*L)->dato==x){
    act=*L;
    *L=(*L)->sig;
    free(act);
  }
  else {
    act = *L;
    while (act != NULL && x>act->dato){
      ant=act;
      act = act -> sig;
    }
    if (act != NULL && x==act->dato) {
       ant->sig=act->sig;
       free(act);
    }
   }
}
```