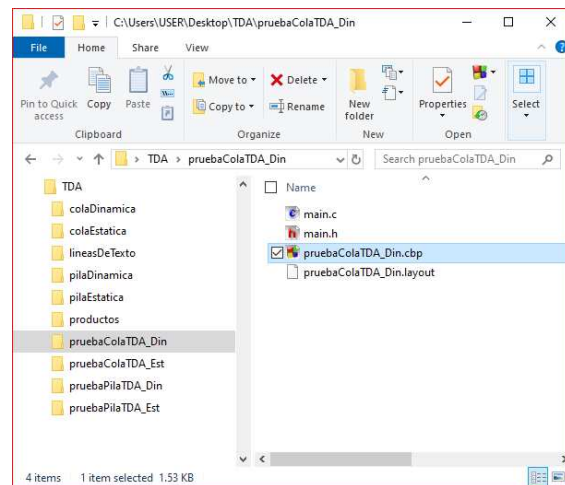
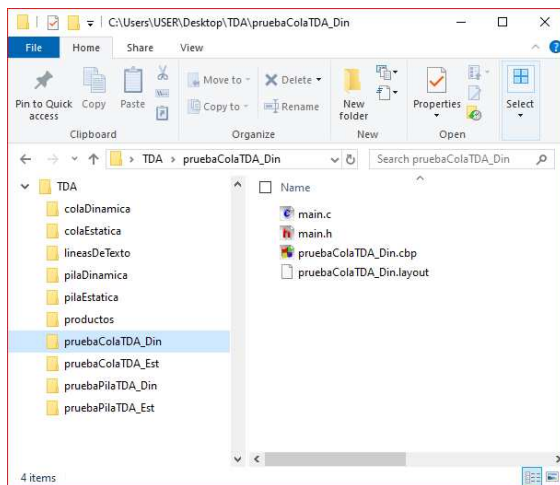


Tipos de Datos Abstractos (TDA) - COLA.

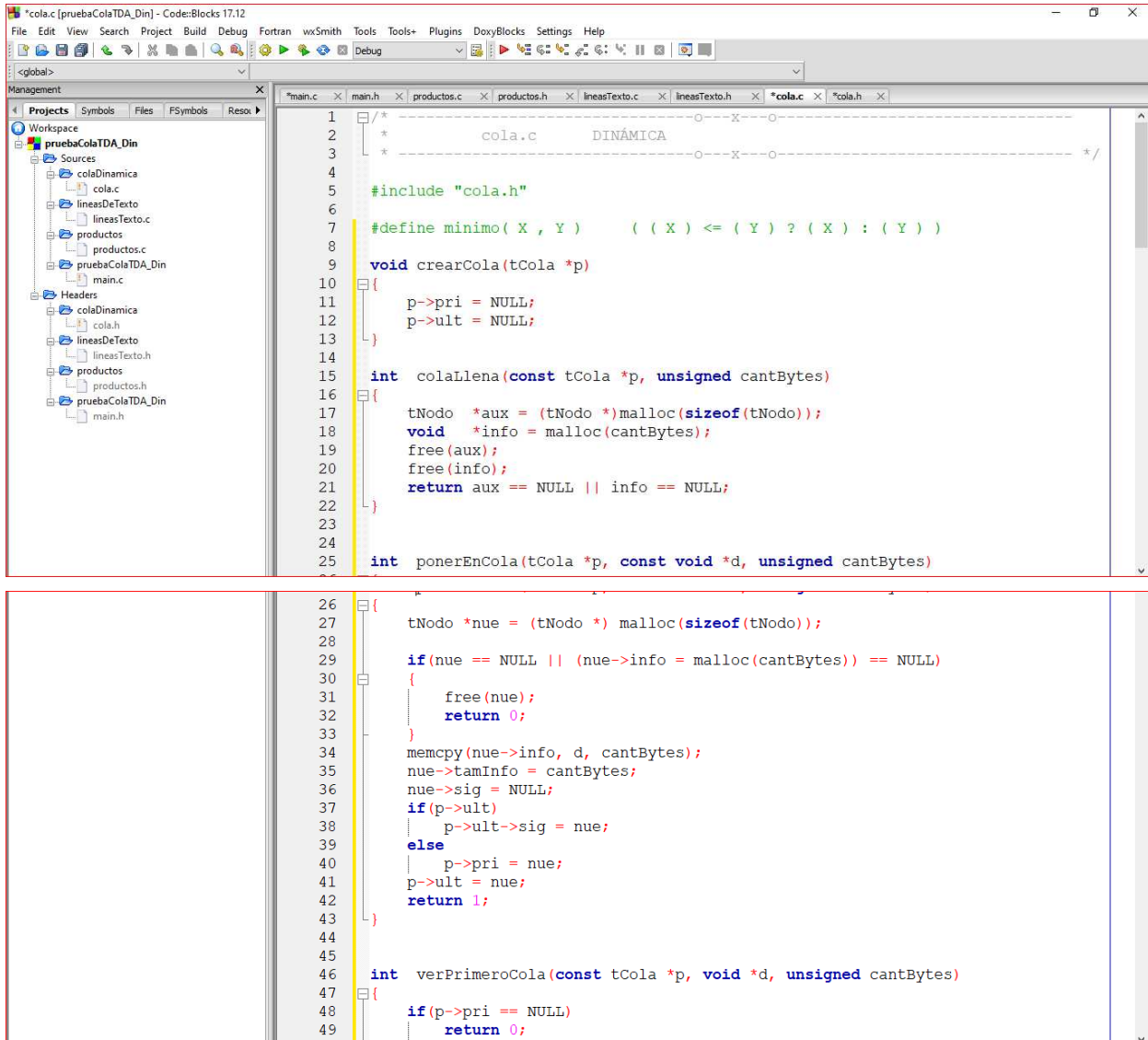
Codificar las primitivas con memoria estática y dinámica.

Cuando generamos los proyectos "pruebaColaTDA_Din" y "pruebaColaTDA_Est", se dejó el "raquítico esqueleto" de "cola.c" y "cola.h" en las carpetas "colaDinamica" y en "colaEstatica".

Una vez entendido el problema, y habiendo elegido las respectivas estrategias, y que ya se ha comenzado con un esbozo de solución, pase a codificar y probar. Comenzando por ejemplo con la implementación dinámica, vaya a su carpeta de trabajo (que en mi caso está en la carpeta TDA en el escritorio), y abra el proyecto . . .



Una vez abierto complete las definiciones de "cola.c" . . .



```

1  /*
2  *      cola.c      DINÁMICA
3  *
4  */
5
6  #include "cola.h"
7
8  #define minimo( X , Y )    ( ( X ) <= ( Y ) ? ( X ) : ( Y ) )
9
10 void crearCola(tCola *p)
11 {
12     p->pri = NULL;
13     p->ult = NULL;
14 }
15
16 int colaLlena(const tCola *p, unsigned cantBytes)
17 {
18     tNodo *aux = (tNodo *)malloc(sizeof(tNodo));
19     void *info = malloc(cantBytes);
20     free(aux);
21     free(info);
22     return aux == NULL || info == NULL;
23 }
24
25 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
26 {
27     tNodo *nue = (tNodo *) malloc(sizeof(tNodo));
28
29     if(nue == NULL || (nue->info = malloc(cantBytes)) == NULL)
30     {
31         free(nue);
32         return 0;
33     }
34     memcpy(nue->info, d, cantBytes);
35     nue->tamInfo = cantBytes;
36     nue->sig = NULL;
37     if(p->ult)
38         p->ult->sig = nue;
39     else
40         p->pri = nue;
41     p->ult = nue;
42     return 1;
43 }
44
45 int verPrimeroCola(const tCola *p, void *d, unsigned cantBytes)
46 {
47     if(p->pri == NULL)
48         return 0;
49
50     tNodo *nue = p->pri;
51     while(nue->sig != NULL)
52         nue = nue->sig;
53     memcpy(d, nue->info, cantBytes);
54     return 1;
55 }
56
57 int verUltimoCola(const tCola *p, void *d, unsigned cantBytes)
58 {
59     if(p->ult == NULL)
60         return 0;
61     tNodo *nue = p->ult;
62     while(nue->sig != NULL)
63         nue = nue->sig;
64     memcpy(d, nue->info, cantBytes);
65     return 1;
66 }

```



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```

50     memcpy(d, p->pri->info, minimo(cantBytes, p->pri->tamInfo));
51     return 1;
52 }
53
54
55 int colaVacia(const tCola *p)
56 {
57     return p->pri == NULL;
58 }
59
60
61 int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
62 {
63     tNodo *aux = p->pri;
64     if(aux == NULL)
65         return 0;
66     p->pri = aux->sig;
67     memcpy(d, aux->info, minimo(aux->tamInfo, cantBytes));
68     free(aux->info);
69     free(aux);
70     if(p->pri == NULL)
71         p->ult = NULL;
72     return 1;
73 }
74
75
76 void vaciarCola(tCola *p)
77 {
78     while(p->pri)
79     {
80         tNodo *aux = p->pri;
81         p->pri = aux->sig;
82         free(aux->info);
83         free(aux);
84     }
85     p->ult = NULL;
86 }
87
88

```

Continúe con las declaraciones de "cola.h"

```

1  /*
2  *      cola.c      DINÁMICA
3  *
4  */
5  #ifndef COLA_H_
6  #define COLA_H_
7
8  #include <stdlib.h>
9  #include <string.h>
10
11
12 typedef struct sNodo
13 {
14     void          *info;
15     unsigned      tamInfo;
16     struct sNodo  *sig;
17 } tNodo;
18
19 typedef struct

```



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```

20  {
21      tNodo  *pri,
22      *ult;
23  } tCola;
24
25  void crearCola(tCola *p);
26
27  int colaLlena(const tCola *p, unsigned cantBytes);
28
29  int ponerEnCola(tCola *p, const void *d, unsigned cantBytes);
30
31  int verPrimeroCola(const tCola *p, void *d, unsigned cantBytes);
32
33  int colaVacia(const tCola *p);
34
35  int sacarDeCola(tCola *p, void *d, unsigned cantBytes);
36
37  void vaciarCola(tCola *p);
38
39  #endif
40
41

```

Continúe con "main.h"

```

1  /*
2   *      main.h      prueba del TDA COLA con asignación dinámica de memoria
3   *      -----
4   *
5   *
6   *
7   *
8   *
9   *
10  *
11  *
12  *
13  *
14  *
15  *
16  *
17  *
18  *
19  *
20  *
21  *
22  *
23  *
24  */

```

```

1  #ifndef MAIN_H_
2  #define MAIN_H_
3
4  #include <stdio.h>
5
6  #include "../productos/productos.h"
7  #include "../lineasDeTexto/lineasTexto.h"
8  #include "../colaDinamica/cola.h"
9
10 void probarIngresarYMostrarProd(void);
11
12 void probarIngresarYMostrarTexto(void);
13
14 void probarPonerYSacarDeCola(void);
15
16 #endif

```



Y finalmente en **"main.c"**

The screenshot shows the Code::Blocks IDE with the following components:

- Project Manager (Left):** Displays the project structure for 'pruebaColaTDA_Din'. It includes source files like 'colaDinamica.c', 'lineasDeTexto.c', 'productos.c', and 'pruebaColaTDA_Din/main.c', as well as header files like 'cola.h', 'lineasDeTexto.h', and 'productos.h'.
- Toolbar (Top):** Contains standard IDE icons for file operations, compilation, and debugging.
- Main Editor (Right):** Shows the source code for 'main.c'. The code is as follows:


```

1  /*-----X-----*/
2  *      main.c      prueba del TDA COLA con asignación dinámica de memoria
3  *
4  *-----O-X-O-----*/
5  #include "main.h"
6
7
8  int main(void)
9  {
10     probarIngresarYMostrarProd();
11
12     probarIngresarYMostrarTexto();
13
14     probarPonerYSacarDeCola();
15
16     return 0;
17 }
18
19
20 void probarIngresarYMostrarProd(void)
21 {
22     tProd  prod;
23     int    cant = 0;
24
25     puts("Probando ingresar productos y mostrar productos.\n"
26
27         | "===== = =====");
28     if (ingresarProducto(&prod))
29         mostrarProducto(NULL);
30     do
31     {
32         mostrarProducto(&prod);
33         cant++;
34     } while (ingresarProducto(&prod));
35     fprintf(stdout, "Se mostraron %d productos.\n\n", cant);
36 }
37
38 void probarIngresarYMostrarTexto(void)
39 {
40     char    linea[90];
41     int     cant = 0;
42
43     puts("Probando ingresar lineas de texto mostrandolas.\n"
44         | "===== = =====");
45     while (ingresarTexto(linea, sizeof(linea)))
46     {
47         cant++;
48         printf("\n%s\n", linea);
49     }
50     fprintf(stdout, "Se mostraron %d lineas de texto.\n\n", cant);

```



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```

50 }
51
52 int _probarLlenaYEncolar(tCola *cola)
53 {
54     tProd prod;
55     int cant = 0;
56
57     puts("Probando cola llena y poner en cola.");
58     mostrarProducto(NULL);
59     while(!colaLlena(cola, sizeof(prod)) && ingresarProducto(&prod))
60     {
61         if(!ponerEnCola(cola, &prod, sizeof(prod)))
62         {
63             fprintf(stderr, "ERROR - inesperado - cola llena.\n");
64             puts("no se pudo cargar la informacion y"
65                 " habria que tomar alguna decision drastica.");
66         }
67         mostrarProducto(&prod);
68         cant++;
69     }
70     return cant;
71 }
72
73

```

```

74 void _probarVerTope(tCola *cola)
75 {
76     tProd prod;
77
78     puts("Probando ver el primero de la cola.");
79     if(verPrimeroCola(cola, &prod, sizeof(prod)))
80     {
81         mostrarProducto(NULL);
82         mostrarProducto(&prod);
83     }
84     else
85     {
86         puts("La cola estaba vacia.");
87         puts("");
88     }
89
90 void _probarVaciarYDesacolarN(tCola *cola, int canti)
91 {
92     tProd prod;
93
94     printf("Probando cola vacia y sacar de cola %d productos (mostrandolos.\n",
95         canti);
96     if(colaVacía(cola))
97     {
98         puts("La cola esta vacia.");
99     }
100

```

```

101     else
102     {
103         mostrarProducto(NULL);
104         while(canti > 0 && sacarDeCola(cola, &prod, sizeof(prod)))
105         {
106             canti--;
107             mostrarProducto(&prod);
108         }
109         puts("");
110     }
111
112 int _probarVaciarColaYColaVacía(tCola *cola)
113 {
114     puts("Probando vaciar cola y cola vacia.");
115     vaciarCola(cola);
116     if(!colaVacía(cola))
117     {
118         return 0; // fprintf(stderr, "ERROR - la cola debia estar vacia\n\n");
119     }
120     printf("Vaciar cola funciona!\n\n");
121     puts("");
122     return 1;
123 }
124
125 void _probarLlenaYEncolarTexto(tCola *cola)
126 {

```




UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```

122  {
123      char    linea[70];
124      int      cant = 0;
125
126      puts("Probando cola llena y poner en cola Texto.");
127      while(!colaLlena(cola, sizeof(linea)) &&
128            ingresarTexto(linea, sizeof(linea)))
129      {
130          if(!ponerEnCola(cola, linea, strlen(linea) + 1))
131          {
132              fprintf(stderr, "ERROR - inesperado - cola llena.\n");
133              puts("no se pudo cargar la informacion y"
134                  " habria que tomar alguna decision drastica.");
135          }
136          printf("\n%s\n", linea);
137          cant++;
138      }
139      printf("se pusieron %d lineas de texto en la cola.\n\n", cant);
140      printf("Probando sacar de cola con las lineas de texto.\n");
141  }
142
143
144  void _probarSacarDeColaTexto(tCola *cola)
145  {

```

```

146      char    linea[70];
147      int      cant = 0;
148
149      while(sacarDeCola(cola, linea, sizeof(linea)))
150      {
151          cant++;
152          printf("\n%s\n", linea);
153      }
154      printf("Se sacaron y mostraron %d lineas de texto\n\n", cant);
155  }
156
157  void probarPonerYSacarDeCola(void)
158  {
159      tCola    cola;
160      int      cant;
161
162      crearCola(&cola);
163
164      puts("Probando primitivas de cola con productos.\n"
165          "===== \n");
166      cant = _probarLlenaYEncolar(&cola);
167      printf("se pusieron %d productos en la cola.\n\n", cant);
168
169

```

```

170      _probarVerTope(&cola);
171
172      _probarVaciarDesacolarN(&cola, cant - 2);
173
174      _probarVerTope(&cola);
175
176      if(!_probarVaciarColaYColaVacía(&cola) != 1)
177          fprintf(stderr, "ERROR - inesperado, la cola NO QUEDO vacia\n");
178
179      puts("Probando primitivas de cola con lineas de texto.\n"
180          "===== \n");
181
182      _probarLlenaYEncolarTexto(&cola);
183
184      _probarSacarDeColaTexto(&cola);
185      cant = 0;
186
187      puts("ATENCIÓN: se mostro el uso de una cola, en la que se pusieron"
188          " productos, se pro\n"
189          "baron todas las primitivas. Una vez que se la dejo vacia "
190          "se encolaron lineas\n"
191          "de texto y luego se desencolaron. Lo mas importante de esto "
192          "no es utilizar la\n"
193          "misma cola, lo que mas importa es que con las mismas primiti"

```



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```

194         "vas se pueden enco-\n"
195         "lar distintos tipos de objetos, incluyendo lineas de texto d"
196         "e distinto tamano.\n");
197     }
198
199

```

Para luego asegurarse que lo hecho compila y ejecuta sin problemas . . .

```

Logs & others
Code::Blocks  Search results  Cccc  Build log  Build messages  CppCheck/Vera++  CppCheck/Vera++ messages  Cscs
----- Clean: Debug in pruebaColaTDA_Din (compiler: GNU GCC Compiler) -----
Cleaned "pruebaColaTDA_Din - Debug"
----- Build: Debug in pruebaColaTDA_Din (compiler: GNU GCC Compiler) -----
mingw32-gcc.exe -Wall -DDINAMICA -g -o C:\Users\USER\Desktop\TDA\pruebaColaTDA_Din\cola.o -c obj\Debug\colaDinamica\cola.o
mingw32-gcc.exe -Wall -DDINAMICA -g -o C:\Users\USER\Desktop\TDA\pruebaColaTDA_Din\lineasDeTexto.o -c obj\Debug\lineasDeTexto\lineasDeTexto.o
mingw32-gcc.exe -Wall -DDINAMICA -g -o C:\Users\USER\Desktop\TDA\pruebaColaTDA_Din\productos.o -c obj\Debug\productos\productos.o
mingw32-gcc.exe -Wall -DDINAMICA -g -o C:\Users\USER\Desktop\TDA\pruebaColaTDA_Din\main.o -c obj\Debug\pruebaColaTDA_Din\main.o
mingw32-g++.exe -o bin\Debug\pruebaColaTDA_Din.exe obj\Debug\colaDinamica\cola.o obj\Debug\lineasDeTexto\lineasDeTexto.o obj\Debug\productos\productos.o obj\Debug\pruebaColaTDA_Din\main.o
Output file is bin\Debug\pruebaColaTDA_Din.exe with size 41.91 KB
Process terminated with status 0 (0 minute(s), 1 second(s))

```

```

Select C:\Users\USER\Desktop\TDA\pruebaColaTDA_Din\bin\Debug\pruebaColaTDA_Din.exe
Probando ingresar productos y mostrar productos.
=====
Cod. Produ Descripcion del producto ...
clavoro3/4 Clavo de oro 24 kilates de 3/4 de pulgada ...
martillo3K Martillo bolita con saca clavos de 3 kilos ...
alamesol Alambre de yeso de un milimetro de espesor ...
rem-vid15 Remache de vidrio de 1,5 milímetros ...
plom-telgo Plomada de poliestireno expandido ...
limagoma17 Lima de goma de 17 pulgadas ...
Se mostraron 6 productos.

Probando ingresar lineas de texto mostrandolas.
=====
"Se necesita un amigo - Fragmento del Poema de Vinicius de Moraes"
""
"Debe tener un ideal, y miedo de perderlo,"
"y en caso de no ser asi,"
"debe sentir el gran vacio que esto deja."
"Tiene que tener resonancias humanas,"
"su principal objetivo debe ser el del amigo."
"Debe sentir pena por las personas tristes"
"y comprender el inmenso vacio de los solitarios."
"Se busca un amigo para gustar"
"de los mismos gustos,"
"que se conmueva cuando es tratado de amigo."
""
Se mostraron 13 lineas de texto.

Probando primitivas de cola con productos.
=====

```




UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```
Select C:\Users\USER\Desktop\TDA\pruebaColaTDA_Din\bin\Debug\pruebaColaTDA_Din.exe
Probando cola llena y poner en cola.
Cod. Produ Descripción del producto      ...
clavoro3/4 Clavo de oro 24 kilates de 3/4 de pulgada ...
martillo3K Martillo bolita con saca clavos de 3 kilos ...
alamesol Alambre de yeso de un milimetro de espesor ...
rem-vid15 Remache de vidrio de 1,5 milímetros ...
plom-telgo Plomada de poliestireno expandido ...
limagomal7 Lima de goma de 17 pulgadas ...
se pusieron 6 productos en la cola.

Probando ver el primero de la cola.
Cod. Produ Descripción del producto      ...
clavoro3/4 Clavo de oro 24 kilates de 3/4 de pulgada ...

Probando cola vacia y sacar de cola 4 productos (mostrandolos.
Cod. Produ Descripción del producto      ...
clavoro3/4 Clavo de oro 24 kilates de 3/4 de pulgada ...
martillo3K Martillo bolita con saca clavos de 3 kilos ...
alamesol Alambre de yeso de un milimetro de espesor ...
rem-vid15 Remache de vidrio de 1,5 milímetros ...

Probando ver el primero de la cola.
Cod. Produ Descripción del producto      ...
plom-telgo Plomada de poliestireno expandido ...

Probando vaciar cola y cola vacia.
Vaciar cola funciona!

Probando primitivas de cola con lineas de texto.
=====
```

```
Select C:\Users\USER\Desktop\TDA\pruebaColaTDA_Din\bin\Debug\pruebaColaTDA_Din.exe
Probando cola llena y poner en cola Texto.
"Se necesita un amigo - Fragmento del Poema de Vinicius de Moraes"
""
"Debe tener un ideal, y miedo de perderlo,"
"y en caso de no ser asi,"
"debe sentir el gran vacio que esto deja."
"Tiene que tener resonancias humanas,"
"su principal objetivo debe ser el del amigo."
"Debe sentir pena por las personas tristes"
"y comprender el inmenso vacio de los solitarios."
"Se busca un amigo para gustar"
"de los mismos gustos,"
"que se conmueva cuando es tratado de amigo."
""
se pusieron 13 lineas de texto en la cola.

Probando sacar de cola con las lineas de texto.
"Se necesita un amigo - Fragmento del Poema de Vinicius de Moraes"
""
"Debe tener un ideal, y miedo de perderlo,"
"y en caso de no ser asi,"
"debe sentir el gran vacio que esto deja."
"Tiene que tener resonancias humanas,"
"su principal objetivo debe ser el del amigo."
"Debe sentir pena por las personas tristes"
"y comprender el inmenso vacio de los solitarios."
"Se busca un amigo para gustar"
"de los mismos gustos,"
"que se conmueva cuando es tratado de amigo."
""
```

```

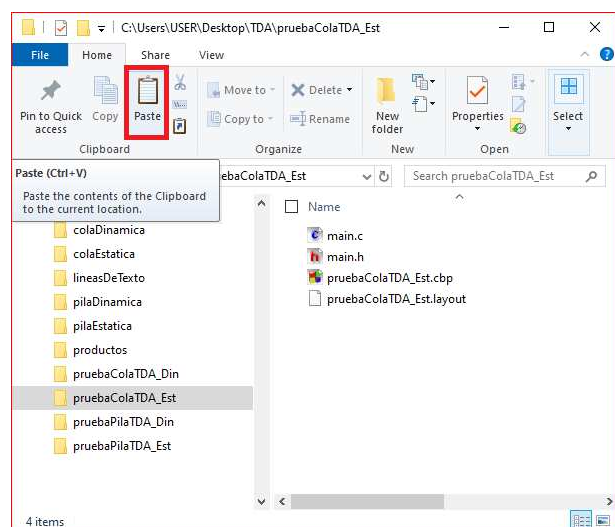
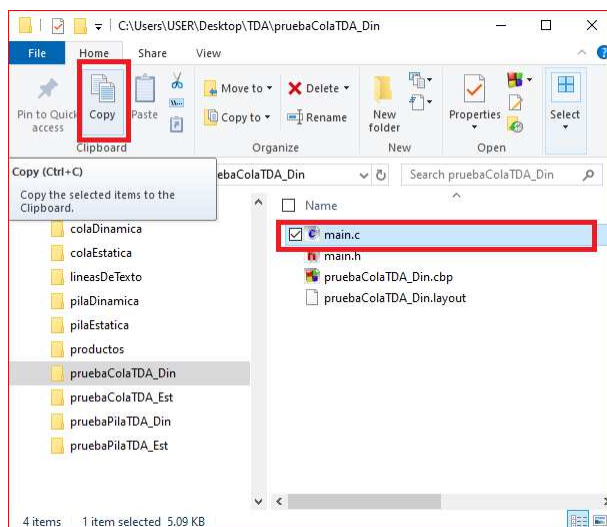
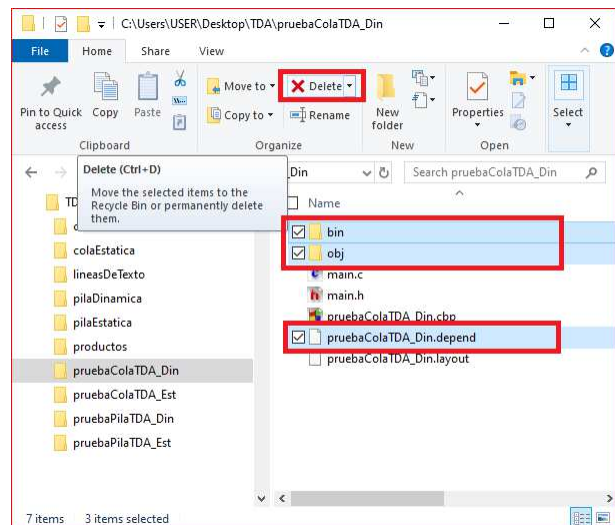
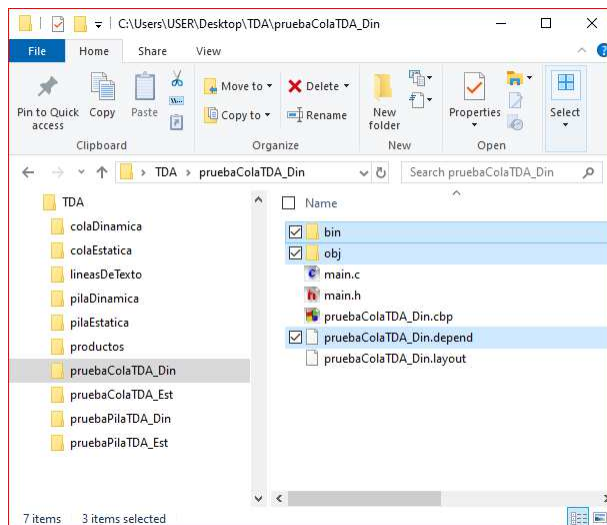
Select C:\Users\USER\Desktop\TDA\pruebaColaTDA_Din\bin\Debug\pruebaColaTDA_Din.exe
Se sacaron y mostraron 13 lineas de texto

ATENCIÓN: se mostro el uso de una cola, en la que se pusieron productos, se pro
baron todas las primitivas. Una vez que se la dejo vacia se encolaron lineas
de texto y luego se desencolaron. Lo mas importante de esto no es utilizar la
misma cola, lo que mas importa es que con las mismas primitivas se pueden enco
lar distintos tipos de objetos, incluyendo lineas de texto de distinto tamano.

Process returned 0 (0x0)   execution time : 0.094 s
Press any key to continue.

```

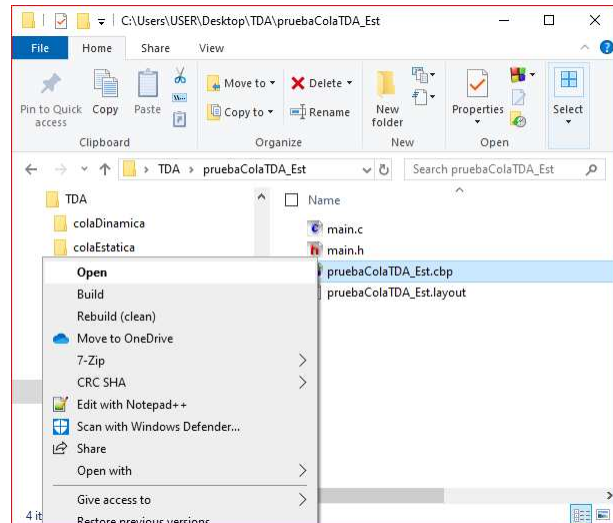
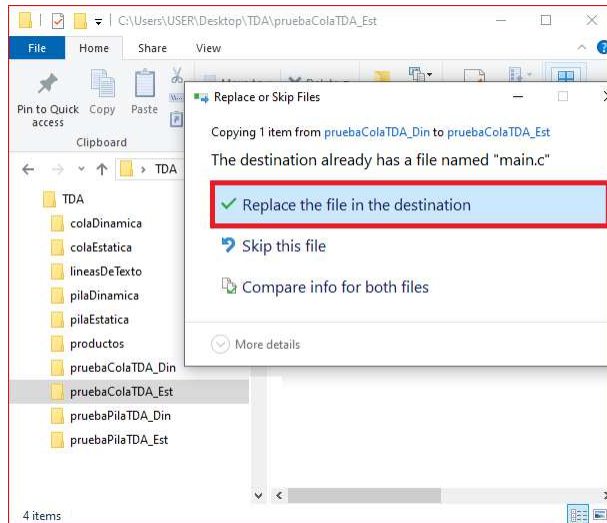
Pasemos al proyecto de prueba de cola estática. Copie de "pruebaColaTDA_Din" a "pruebaColaTDA_Est" el archivo "main.c"



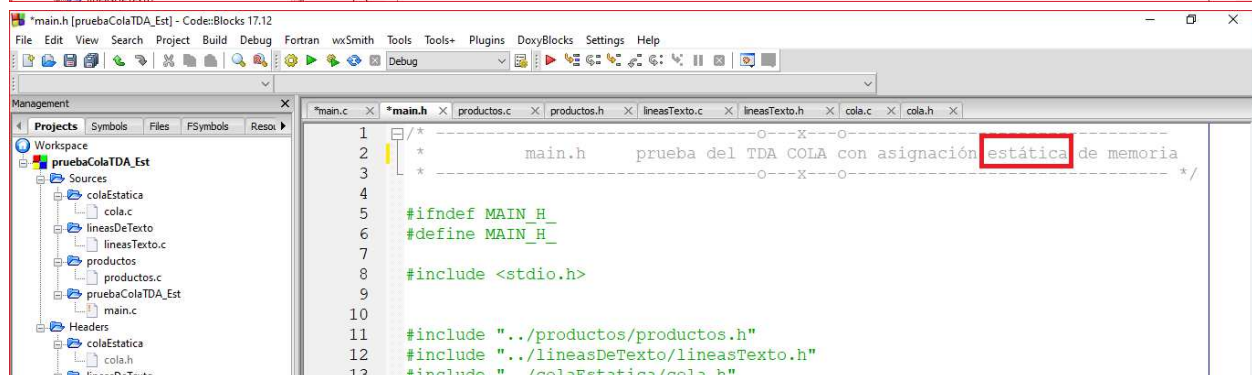
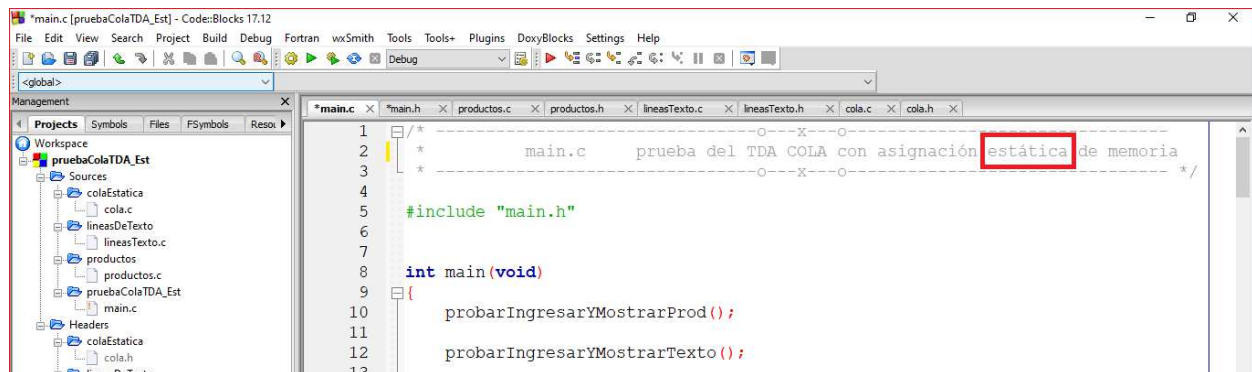


UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas



Abra el proyecto de prueba de cola estática y en "main.c" y "main.h", modifique el comentario de documentación . . .





UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Completar en "cola.c" la definición de las primitivas.

```

*cola.c [pruebaColaTDA_Est] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
Workspace
pruebaColaTDA_Est
Sources
  colaEstatica
  cola.c
  lineasDeTexto.c
  lineasTexto.h
  productos.c
  productos.h
  pruebaColaTDA_Est
  main.c
Headers
  colaEstatica
  cola.h
  lineasDeTexto.h
  productos.h
  pruebaColaTDA_Est
  main.h

1  /*
2  *      cola.c      ESTATICA
3  *
4  */
5  #include "cola.h"
6
7  #define minimo( X , Y )      ( ( X ) <= ( Y ) ? ( X ) : ( Y ) )
8
9
10 void crearCola(tCola *p)
11 {
12     p->pri      = TAM_COLA - 70;
13     p->ult      = TAM_COLA - 70;
14     p->tamDisp  = TAM_COLA;
15 }
16
17 int colaLlena(const tCola *p, unsigned cantBytes)
18 {
19     return p->tamDisp < cantBytes + sizeof(unsigned);
20 }
21
22 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes)
23 {
24     unsigned    ini,
25                fin;
26
27     if(p->tamDisp < sizeof(unsigned) + cantBytes)
28         return 0;
29     p->tamDisp += sizeof(unsigned) + cantBytes;
30     if((ini = minimo(sizeof(cantBytes), TAM_COLA - p->ult)) != 0)
31         memcpy(p->cola + p->ult, &cantBytes, ini);
32     if((fin = sizeof(cantBytes) - ini) != 0)
33         memcpy(p->cola, ((char *)&cantBytes) + ini, fin);
34     p->ult = fin ? fin : p->ult + ini;
35     if((ini = minimo(cantBytes, TAM_COLA - p->ult)) != 0)
36         memcpy(p->cola + p->ult, d, ini);
37     if((fin = cantBytes - ini) != 0)
38         memcpy(p->cola, ((char *)d) + ini, fin);
39     p->ult = fin ? fin : p->ult + ini;
40     return 1;
41 }
42
43 int verPrimeroCola(const tCola *p, void *d, unsigned cantBytes)
44 {
45     unsigned    tamInfo,
46                ini,
47                fin,
48                pos = p->pri;
49

```



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

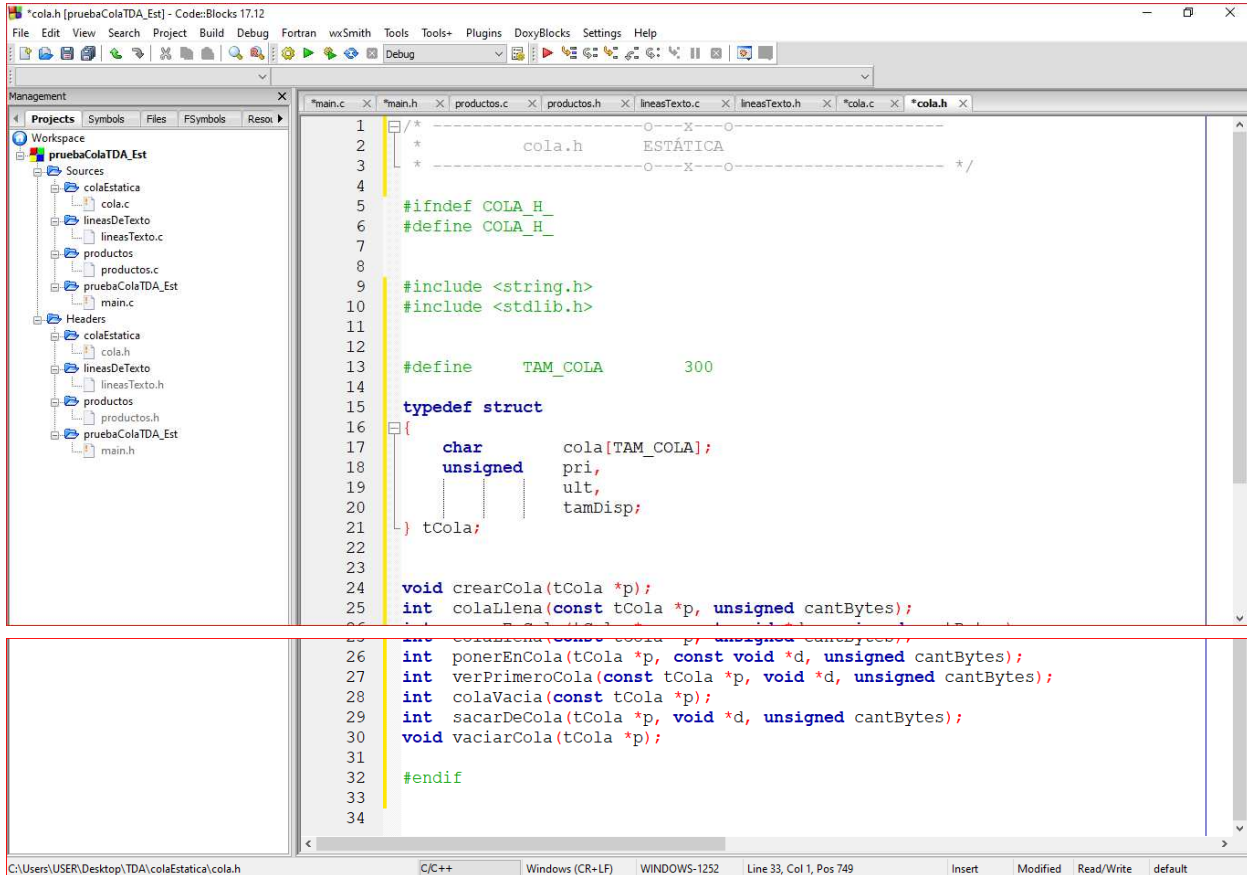
```

50     if(p->tamDisp == TAM_COLA)
51         return 0;
52     if((ini = minimo(sizeof(unsigned), TAM_COLA - pos)) != 0)
53         memcpy(&tamInfo, p->cola + pos, ini);
54     if((fin = sizeof(unsigned) - ini) != 0)
55         memcpy(((char *)&tamInfo) + ini, p->cola, fin);
56     pos = fin ? fin : pos + ini;
57     tamInfo = minimo(tamInfo, cantBytes);
58     if((ini = minimo(tamInfo, TAM_COLA - pos)) != 0)
59         memcpy(d, p->cola + pos, ini);
60     if((fin = tamInfo - ini) != 0)
61         memcpy(((char *)d) + ini, p->cola, fin);
62     return 1;
63 }
64
65 int colaVacia(const tCola *p)
66 {
67     return p->tamDisp == TAM_COLA;
68 }
69
70 int sacarDeCola(tCola *p, void *d, unsigned cantBytes)
71 {
72     unsigned    tamInfo,
73               ini,
74               fin;
75
76     if(p->tamDisp == TAM_COLA)
77         return 0;
78     if((ini = minimo(sizeof(unsigned), TAM_COLA - p->pri)) != 0)
79         memcpy(&tamInfo, p->cola + p->pri, ini);
80     if((fin = sizeof(unsigned) - ini) != 0)
81         memcpy(((char *)&tamInfo) + ini, p->cola, fin);
82     p->pri = fin ? fin : p->pri + ini;
83     p->tamDisp += sizeof(unsigned) + tamInfo;
84     tamInfo = minimo(tamInfo, cantBytes);
85     if((ini = minimo(tamInfo, TAM_COLA - p->pri)) != 0)
86         memcpy(d, p->cola + p->pri, ini);
87     if((fin = tamInfo - ini) != 0)
88         memcpy(((char *)d) + ini, p->cola, fin);
89     p->pri = fin ? fin : p->pri + ini;
90     return 1;
91 }
92
93 void vaciarCola(tCola *p)
94 {
95     p->ult = p->pri;
96     p->tamDisp = TAM_COLA;
97 }
98
99

```

C:\Users\USER\Desktop\TDA\colaEstatica\cola.c C/C++ Windows (CR+LF) WINDOWS-1252 Line 99, Col 1, Pos 2925 Insert Modified Read/Write default

En "cola.h" complete las correspondientes declaraciones . . .



```
1  /*
2  *      cola.h      ESTATICA
3  *
4  */
5  #ifndef COLA_H_
6  #define COLA_H_
7
8
9  #include <string.h>
10 #include <stdlib.h>
11
12
13 #define      TAM_COLA      300
14
15 typedef struct
16 {
17     char      cola[TAM_COLA];
18     unsigned  pri,
19             ult,
20             tamDisp;
21 } tCola;
22
23
24 void crearCola(tCola *p);
25 int colaLlena(const tCola *p, unsigned cantBytes);
26
27 int ponerEnCola(tCola *p, const void *d, unsigned cantBytes);
28 int verPrimeroCola(const tCola *p, void *d, unsigned cantBytes);
29 int colaVacía(const tCola *p);
30 int sacarDeCola(tCola *p, void *d, unsigned cantBytes);
31 void vaciarCola(tCola *p);
32
33 #endif
34
```




UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Compile, asegurándose que no hay errores ni "warnings" luego ejecútelo y vea que el proyecto hace lo que debe . . .

```

C:\Users\USER\Desktop\TDA\pruebaColaTDA_Est\bin\Debug\pruebaColaTDA_Est.exe
Probando ingresar productos y mostrar productos.
=====
Cod. Produ Descripcion del producto    ...
clavoro3/4 Clavo de oro 24 kilates de 3/4 de pulgada    ...
martillo3K Martillo bolita con saca clavos de 3 kilos    ...
alamyeso1 Alambre de yeso de un milimetro de espesor    ...
rem-vid15 Remache de vidrio de 1,5 milímetros    ...
plom-telgo Plomada de poliestireno expandido    ...
limagoma17 Lima de goma de 17 pulgadas    ...
Se mostraron 6 productos.

Probando ingresar lineas de texto mostrandolas.
=====
"Se necesita un amigo - Fragmento del Poema de Vinicius de Moraes"
""
"Debe tener un ideal, y miedo de perderlo,"
"y en caso de no ser asi,"
"debe sentir el gran vacio que esto deja."
"Tiene que tener resonancias humanas,"
"su principal objetivo debe ser el del amigo."
"Debe sentir pena por las personas tristes"
"y comprender el inmenso vacio de los solitarios."
"Se busca un amigo para gustar"
"de los mismos gustos,"
"que se conmueva cuando es tratado de amigo."
""
Se mostraron 13 lineas de texto.

Probando primitivas de cola con productos.
=====

```

```

C:\Users\USER\Desktop\TDA\pruebaColaTDA_Est\bin\Debug\pruebaColaTDA_Est.exe
Probando cola llena y poner en cola.
Cod. Produ Descripcion del producto    ...
clavoro3/4 Clavo de oro 24 kilates de 3/4 de pulgada    ...
martillo3K Martillo bolita con saca clavos de 3 kilos    ...
alamyeso1 Alambre de yeso de un milimetro de espesor    ...
rem-vid15 Remache de vidrio de 1,5 milímetros    ...
se pusieron 4 productos en la cola.

Probando ver el primero de la cola.
Cod. Produ Descripcion del producto    ...
clavoro3/4 Clavo de oro 24 kilates de 3/4 de pulgada    ...

Probando cola vacia y sacar de cola 2 productos (mostrandolos).
Cod. Produ Descripcion del producto    ...
clavoro3/4 Clavo de oro 24 kilates de 3/4 de pulgada    ...
martillo3K Martillo bolita con saca clavos de 3 kilos    ...

Probando ver el primero de la cola.
Cod. Produ Descripcion del producto    ...
alamyeso1 Alambre de yeso de un milimetro de espesor    ...

Probando vaciar cola y cola vacia.
Vaciar cola funciona!

Probando primitivas de cola con lineas de texto.
=====

Probando cola llena y poner en cola Texto.
"Se necesita un amigo - Fragmento del Poema de Vinicius de Moraes"
""

```



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

```
cola.h [pruebaColaTDA_Est] - Cola
File Edit View Search Project
Select C:\Users\USER\Desktop\TDA\pruebaColaTDA_Est\bin\Debug\pruebaColaTDA_Est.exe
"Debe tener un ideal, y miedo de perderlo,"
"y en caso de no ser asi,"
"debe sentir el gran vacio que esto deja."
"Tiene que tener resonancias humanas,"
Se pusieron 6 lineas de texto en la cola.

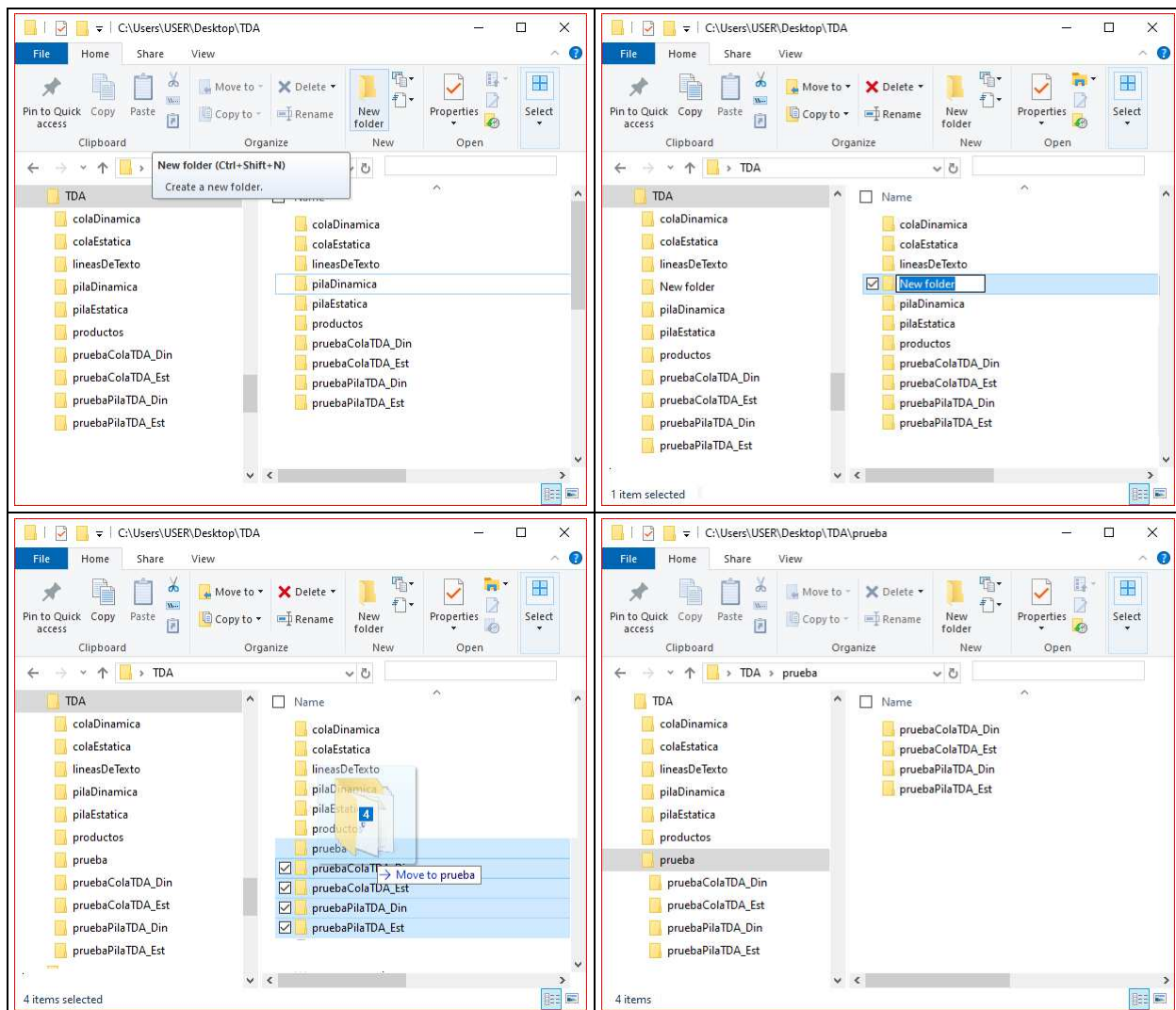
Probando sacar de cola con las lineas de texto.
"Se necesita un amigo - Fragmento del Poema de Vinicius de Moraes"
""
"Debe tener un ideal, y miedo de perderlo,"
"y en caso de no ser asi,"
"debe sentir el gran vacio que esto deja."
"Tiene que tener resonancias humanas,"
Se sacaron y mostraron 6 lineas de texto

ATENCIÓN: se mostro el uso de una cola, en la que se pusieron productos, se probaron todas las primitivas. Una vez que se la dejo vacia se encolaron lineas de texto y luego se desencolaron. Lo mas importante de esto no es utilizar la misma cola, lo que mas importa es que con las mismas primitivas se pueden encolar distintos tipos de objetos, incluyendo lineas de texto de distinto tamaño.

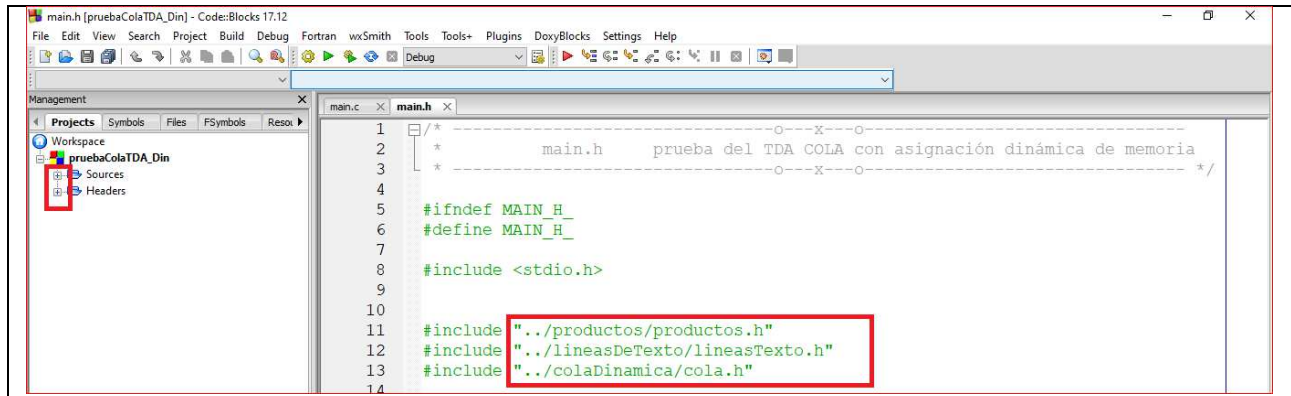
Process returned 0 (0x0)   execution time : 0.062 s
Press any key to continue.
```

Una última observación.

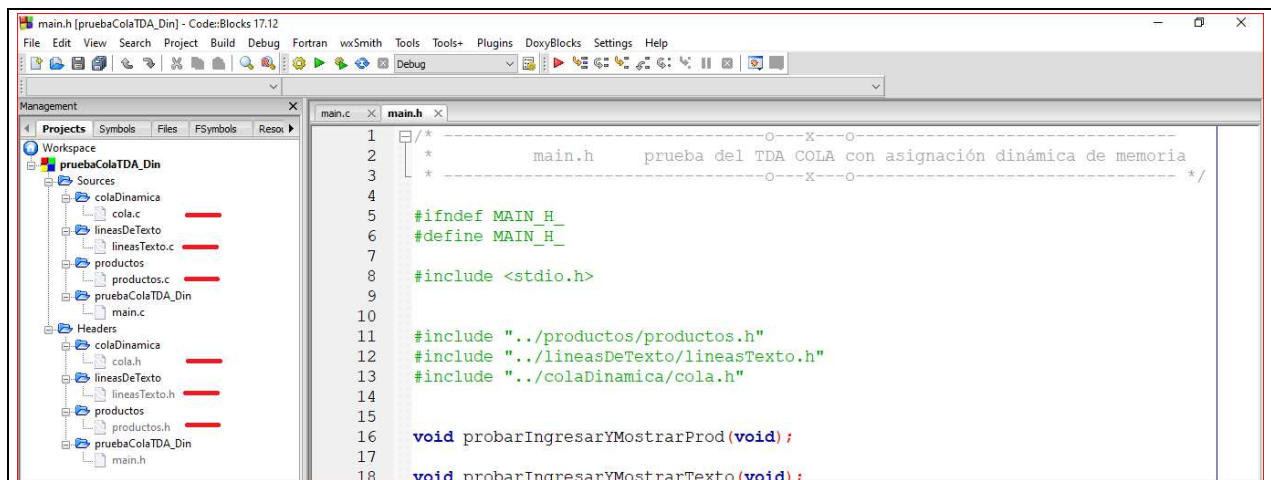
Hasta ahora con estos dos tipos de datos hemos generado dentro de la *carpeta* TDA, cuatro *carpetas* para probar ambas implementaciones. Para tener agrupadas las pruebas, vamos a generar una *carpeta* "**prueba**" y a mover las *carpetas* de los proyectos:



Ahora, en cada proyecto procederemos a hacer las correspondientes correcciones . . .



Las directivas "#include" para el preprocesador de C ya no direccionan los .h de las bibliotecas que hicimos. Por eso, como los archivos fuente no están disponibles (por la diferencia provocada en la *ruta o path*), no los abre en la ventana de edición. Expanda, haciendo "click" en el [+] de la vista del "Workspace" (Espacio de trabajo), y nuevamente en cada [+] que le muestra al expandir **Sources** y **Headers**.

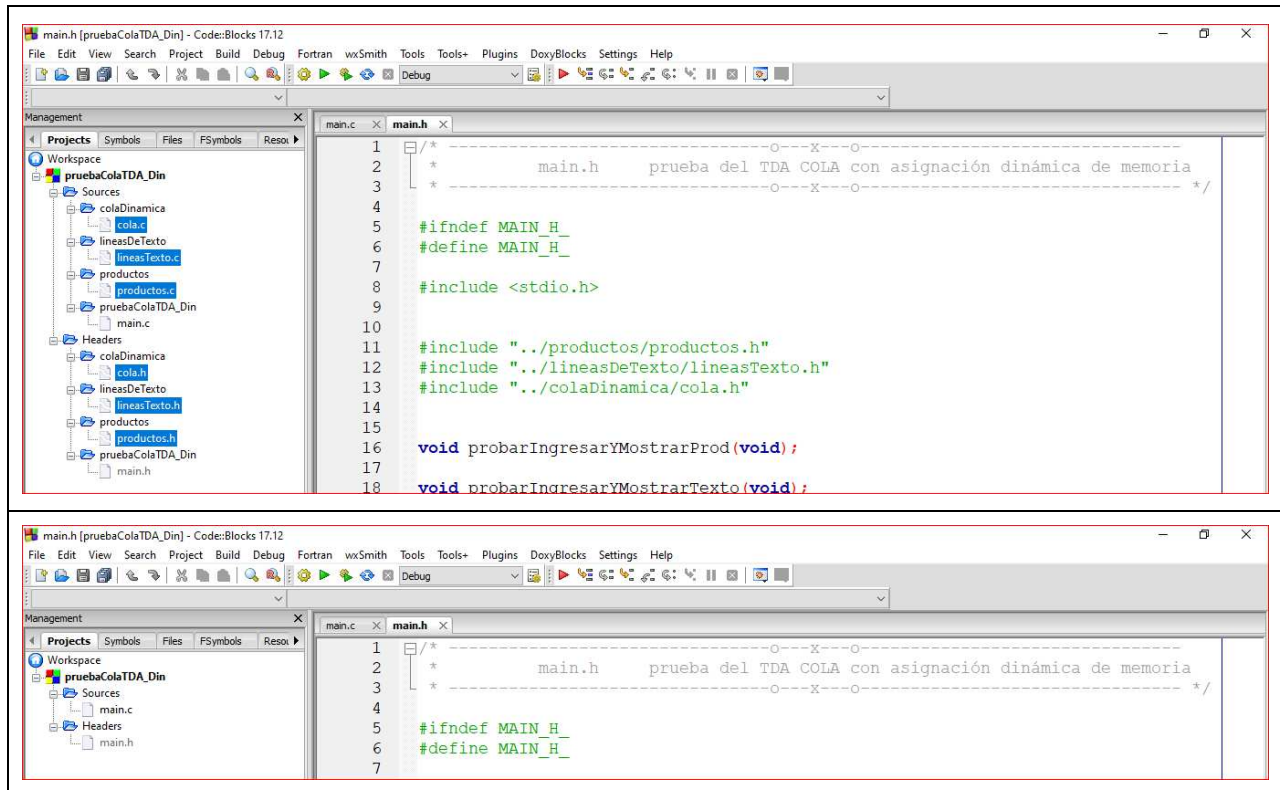


Seleccione haciendo "click" con el mouse en cada uno de los archivos fuente: "cola.c", "lineasTexto.c", "productos.c", "cola.h", "lineasTexto.h" y "productos.h", y presione la tecla [Del] (o [Supr]), para eliminarlos del proyecto.

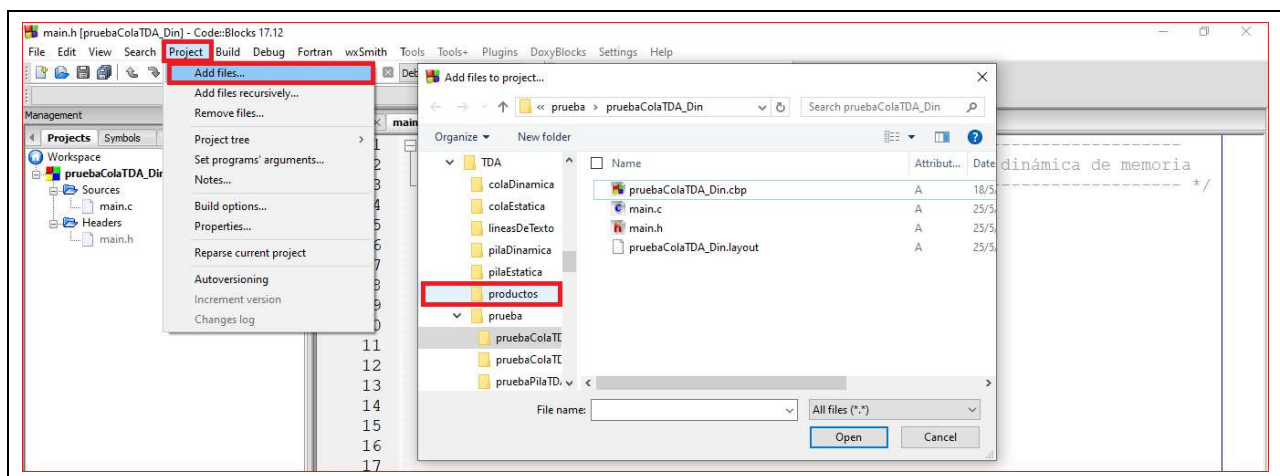


UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas



Entrando por el menú **[Project] / [Add files ...]** se despliega la ventana emergente **[Add files to project...]**. En su parte izquierda seleccione la carpeta **"productos"**

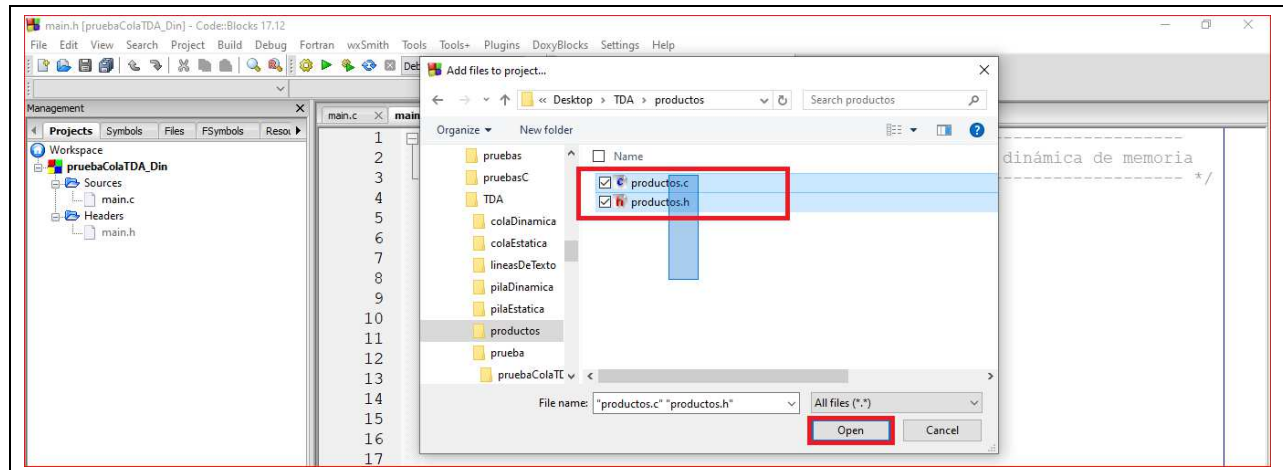




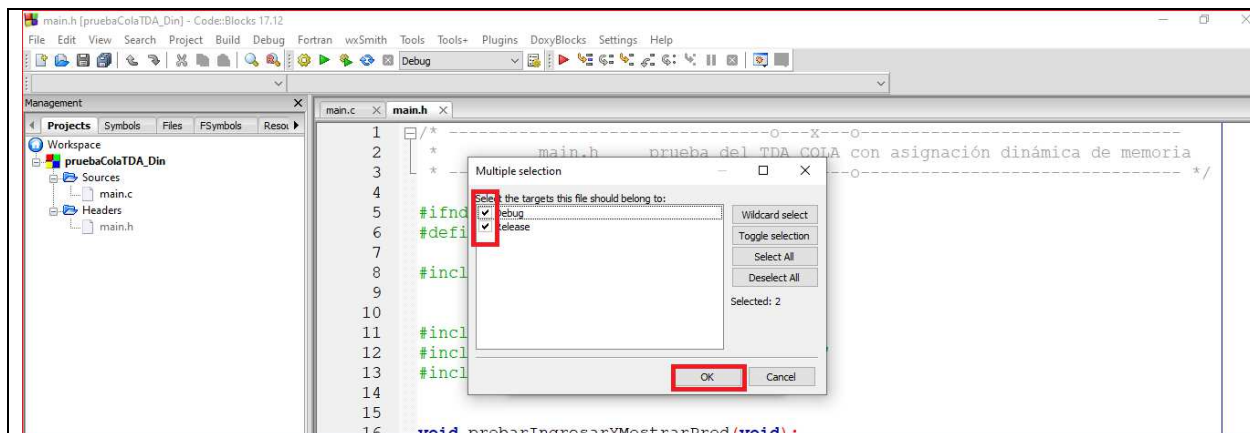
UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Al hacerlo le mostrará los archivos de esa *carpeta*, seleccione ambos y pulse en **[open]**

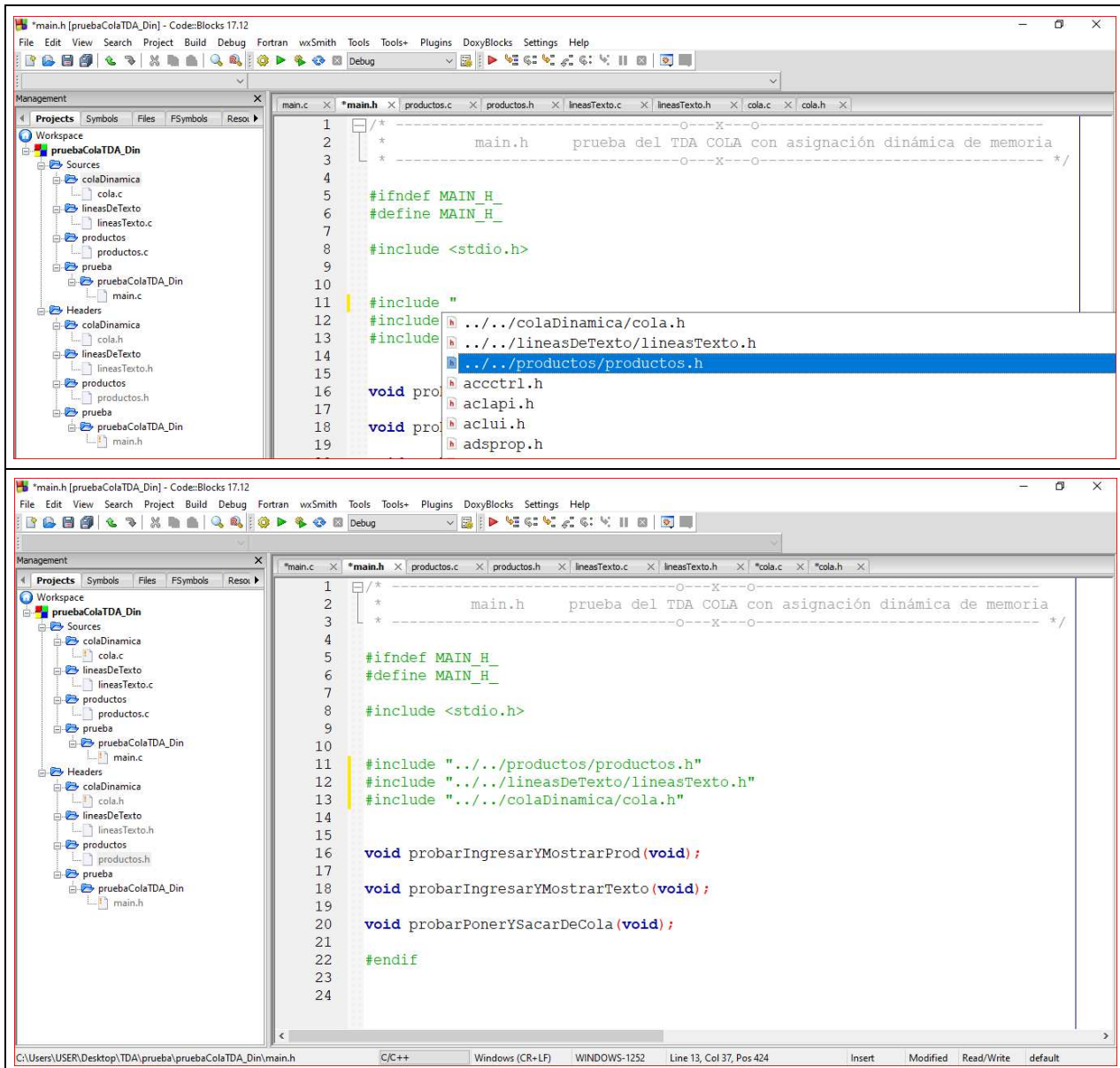


Asegúrese que en la nueva ventana emergente (**[Multiple selection]**), estén seleccionadas las opciones indicadas y pulse en **[ok]**.



Se contraerá el árbol del "Workspace". Una vez expandido verá que "productos.c" y "productos.h" ya forman parte del proyecto. Hágales "doble click" a ambos para abrirlos en la ventana de edición del IDE. Proceda del mismo modo con "lineasTexto.c", "lineasTexto.h", "cola.c" y "cola.h" (estos últimos de la *carpeta* "colaDinamica").

Finalmente, en "main.h", corrija las directivas "#include" para el preprocesador de C.



```
1  /*-----X-----*/
2  *      main.h      prueba del TDA COLA con asignación dinámica de memoria
3  *      *-----X-----*/
4
5  #ifndef MAIN_H
6  #define MAIN_H
7
8  #include <stdio.h>
9
10
11 #include "
12 #include ../../colaDinamica/cola.h
13 #include ../../lineasDeTexto/lineasTexto.h
14 #include ../../productos/productos.h
15
16 void probarIngresarYMostrarProd(void);
17 void probarIngresarYMostrarTexto(void);
18 void probarPonerYSacarDeCola(void);
19 #endif
```

Compile, ejecute, asegúrese que todo sigue andando correctamente. Al terminar elimine el archivo ".depend" y las carpetas "bin" y "obj".

Proceda del mismo modo con los otros proyectos de prueba.