

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en Software y Tecnologías Emergentes
Programación Estructurada

Actividad 12. Archivos de Texto

ALUMNO: Fernando Haro Calvo

MATRICULA: 372106

GRUPO: 932

PROFESOR: Pedro Núñez Yépiz

12 de noviembre del 2023

Funciones Utilizadas:

```
10  /*** PROTOTIPOS DE FUNCIONES *****/
11  int msges(void);
12  void menu(void);
13
14  // Auxiliares
15  Tprogra genPersAlea(void);
16  void imprPersonas(Tprogra vect[], int n, int status);
17  void imprReg(Tprogra pers);
18  void escrArch(char nomArchivo[], Tprogra vect[], int n, bool arch);
19  void leerNomArch(char nomArchivo[]);
20  void actBorrados(char nomArchivo[], Tprogra vect[], int n);
21  int cargarRegistros(Tprogra vect[], int *n);
22  bool cargarArch(char nomArchivo[], Tprogra vect[], int *n);
23  int contarRegArch(char nomArchivo[], int status);
```

Menú:

```
int msges()
{
    int op;
    system("CLS");
    printf("\n  M  E  N  U  \n");
    printf("1.- CARGAR ARCHIVO \n");
    printf("2.- AGREGAR \n");
    printf("3.- ELIMINAR REGISTRO \n");
    printf("4.- BUSCAR \n");
    printf("5.- ORDENAR \n");
    printf("6.- MOSTRAR TABLA (ACTIVOS) \n");
    printf("7.- GENERAR ARCHIVOS \n");
    printf("8.- CANTIDAD DE REGISTROS (ARCHIVO) \n");
    printf("9.- MOSTRAR TABLA (BORRADOS) \n");
    printf("0.- SALIR \n");
    printf("ESCOGE UNA OPCION: ");
    op = valiNum(0, 9);
    return op;
}
```

Ejercicio 1:

```
76 // Cargar archivo
77 case 1:
78     if (!cargado)
79     {
80         leerNomArch(nomArchivo);
81         cargado = cargarArch(nomArchivo, ingenieros, &nPers);
82
83         if (cargado)
84         {
85             printf("Archivo cargado con exito\n");
86         }
87         else
88         {
89             printf("Archivo no encontrado\n");
90         }
91     }
92     else
93     {
94         printf("El archivo ya ha sido cargado\n");
95     }
96     break;
```

```
451 // Carga los registros de un archivo de texto especificado.
452 bool cargarArch(char nomArchivo[], Tprogra vect[], int *n)
453 {
454     FILE *fa;
455     Tprogra reg;
456
457     char temp[30];
458
459     strcpy(temp, nomArchivo);
460     strcat(temp, "_activos.txt");
461
462     fa = fopen(temp, "r");
463
464     if (fa)
465     {
466         while (!feof(fa))
467         {
468             if ((*n) + 1) <= N)
469             {
470                 reg.status = 1;
471                 fscanf(fa, "%s %d %s %s %s %d %s", &temp, &reg.matricula, &reg.nombre, &reg.apmat, &reg.apmat, &reg.edad, &reg.sexo);
472                 reg.key = reg.matricula;
473                 vect[(*n)++] = reg;
474             }
475             else
476             {
477                 printf("Vector llenado\n");
478             }
479         }
480
481         fclose(fa);
482         return true;
483     }
484
485     return false;
486 }
```

Salida 1:

No existe:

```
Ingresa el nombre del archivo (sin extension): datos  
Archivo no encontrado  
  
Presione una tecla para continuar . . . █
```

Existe:

```
Ingresa el nombre del archivo (sin extension): datos  
Archivo cargado con exito  
  
Presione una tecla para continuar . . . █
```

Ejercicio 2:

```
98 // Agregar registros
99 case 2:
100     num = cargarRegistros(ingenieros, &nPers);
101
102     if (num == -1)
103     {
104         printf("Ha llegado al maximo de personas\n");
105     }
106     else
107     {
108         printf("Se han agregado %d personas\n", N_AUTO);
109         ordenado = false;
110     }
111
112     if (strcmp(nomArchivo, "\n") != 0)
113     {
114         escrArch(nomArchivo, ingenieros, nPers, false);
115         escrArch(nomArchivo, ingenieros, nPers, true);
116     }
117     break;
```

```
502 // Carga n cantidad de registros de personas aleatoriamente dentro del vector de personas especificado.
503 int cargarRegistros(Tprogra vect[], int *nPers)
504 {
505     int i;
506     Tprogra pers;
507
508     if ((*nPers + N_AUTO) <= N)
509     {
510         for (i = 0; i < N_AUTO; i++)
511         {
512             pers = genPersAlea();
513
514             while (busqSeq(vect, *nPers, pers.key) != -1)
515             {
516                 pers.matricula = matriAlea();
517                 pers.key = pers.matricula;
518             }
519
520             vect[(*nPers)++] = pers;
521         }
522     }
523     else
524     {
525         return -1;
526     }
527 }
```

Salida 2:

Vacio:

```
Se han agregado 10 personas
```

```
Presione una tecla para continuar . . . █
```

Lleno:

```
Ha llegado al maximo de personas
```

```
Presione una tecla para continuar . . . █
```

Ejercicio 3:

```
120 case 3:
121     printf("Ingrese la matricula del estudiante que desea eliminar: ");
122     num = valiNum(300000, 399999);
123     system("CLS");
124
125     i = busqOpt(ingenieros, nPers, num, ordenado);
126
127     if (i != -1)
128     {
129         if (ingenieros[i].status != 0)
130         {
131             imprReg(ingenieros[i]);
132             printf("\n\nDesea eliminar el registro? (1 - Si, 2 - No) ");
133             op = valiNum(1, 2);
134             system("CLS");
135
136             if (op == 1)
137             {
138                 ingenieros[i].status = 0;
139                 printf("Matricula eliminada con exito\n");
140             }
141             else
142             {
143                 printf("Matricula no eliminada\n");
144             }
145         }
146         else
147         {
148             printf("El alumno ya se encuentra inactivo\n");
149         }
150     }
151     else
152     {
153         printf("Matricula no encontrada\n");
154     }
155     break;
```

Salida 3:

No existe:

```
Ingrese la matricula del estudiante que desea eliminar: 300000
```

```
Matricula no encontrada
```

```
Presione una tecla para continuar . . .
```

Existe:

```
Ingrese la matricula del estudiante que desea eliminar: 394778
```

```
STATUS: ACTIVO  
MATRICULA: 394778  
NOMBRE: DIEGO  
AP. PATERNO: DUARTE  
AP. MATERNO: GARCIA  
EDAD: 26  
SEXO: MASCULINO
```

```
Desea eliminar el registro? (1 - Si, 2 - No)
```

```
Matricula eliminada con exito
```

```
Presione una tecla para continuar . . .
```


Ejercicio 4:

```
157 // Buscar registros
158 case 4:
159     printf("Ingrese la matricula del estudiante que desea buscar: ");
160     num = valiNum(300000, 399999);
161     system("CLS");
162
163     i = busqOpt(ingenieros, nPers, num, ordenado);
164
165     if (i != -1)
166     {
167         printf("Matricula encontrada\n\n", num);
168         imprReg(ingenieros[i]);
169     }
170     else
171     {
172         printf("La matricula %d no se encuentra en el vector\n", num);
173     }
174     break;
```

```
240 // Busca un valor en un arreglo usando el algoritmo optimo
241 int busqOpt(Tprogra vect[], int n, Tkey num, bool band)
242 {
243     int i;
244
245     if (band)
246     {
247         i = busqBin(vect, n, num);
248     }
249     else
250     {
251         i = busqSeq(vect, n, num);
252     }
253
254     return i;
255 }
```

Salida 4:

No existe:

```
Ingrese la matricula del estudiante que desea buscar: 300000
```

```
La matricula 300000 no se encuentra en el vector
```

```
Presione una tecla para continuar . . .
```

Existe:

```
Matricula encontrada
```

```
STATUS: NO ACTIVO
```

```
MATRICULA: 394778
```

```
NOMBRE: DIEGO
```

```
AP. PATERNO: DUARTE
```

```
AP. MATERNO: GARCIA
```

```
EDAD: 26
```

```
SEXO: MASCULINO
```

```
Presione una tecla para continuar . . .
```

Ejercicio 5:

```
176 // Ordenar registros
177 case 5:
178     if (ordenado == false)
179     {
180         ordenado = ordOpt(ingenieros, nPers);
181         printf("El vector ha sido ordenado\n");
182     }
183     else
184     {
185         printf("El vector ya estaba ordenado\n");
186     }
187     break;
```

```
378 // Ordena un vector usando el algoritmo optimo
379 bool ordOpt(Tprogra vect[], int n)
380 {
381     int i;
382
383     if (n <= 200)
384     {
385         ordBubble(vect, n);
386     }
387     else
388     {
389         ordQuick(vect, n);
390     }
391
392     return true;
393 }
```

Salida 5:

Antes de ordenar:

Registros 1 - 40

No	MATRICULA	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
0.-	313695	LUIS	ZAVALA	BAEZ	18	MASCULINO
1.-	340324	PEDRO	PEREDES	ZAVALA	20	MASCULINO
2.-	383421	CESAR	FUENTES	MARTINEZ	21	MASCULINO
3.-	385402	MIRIAM	PENA	GUZMAN	27	FEMENINO
4.-	380951	CLAUDIO	LEON	CHACON	18	MASCULINO
5.-	329890	ALMA	PEREDES	CONTRERAS	20	FEMENINO
6.-	336299	VICTOR	SALAZAR	FRIAS	28	MASCULINO
7.-	327418	JORGE	CALDERON	VELA	20	MASCULINO
8.-	332288	SUSANA	REYES	ESCOBAR	29	FEMENINO

Presione una tecla para continuar . . .

Después de ordenar:

El vector ha sido ordenado

Presione una tecla para continuar . . .

Registros 1 - 40

No	MATRICULA	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
0.-	313695	LUIS	ZAVALA	BAEZ	18	MASCULINO
1.-	327418	JORGE	CALDERON	VELA	20	MASCULINO
2.-	329890	ALMA	PEREDES	CONTRERAS	20	FEMENINO
3.-	332288	SUSANA	REYES	ESCOBAR	29	FEMENINO
4.-	336299	VICTOR	SALAZAR	FRIAS	28	MASCULINO
5.-	340324	PEDRO	PEREDES	ZAVALA	20	MASCULINO
6.-	380951	CLAUDIO	LEON	CHACON	18	MASCULINO
7.-	383421	CESAR	FUENTES	MARTINEZ	21	MASCULINO
8.-	385402	MIRIAM	PENA	GUZMAN	27	FEMENINO

Presione una tecla para continuar . . .

Ejercicio 6:

```
189 // Mostrar registros activos
190 case 6:
191     imprPersonas(ingenieros, nPers, 1);
192     break;
```

```
286 // Imprime todas las personas en un vector dado.
287 void imprPersonas(Tprogra vect[], int n, int status)
288 {
289     int i, activos, op;
290
291     printf("Registros 1 - 40\n\n");
292     printf("-----\n");
293     printf(" No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO \n");
294     printf("-----\n");
295     for (i = 0, activos = 1; i < n; i++)
296     {
297         if (vect[i].status == status)
298         {
299             printf("%4d.- %6d %-10s %-10s %-10s %2d %-7s\n", activos - 1, vect[i].matricula,
300                 activos++,
301
302
303
304             if (activos % 41 == 0 && activos < n)
305             {
306                 printf("\n\n");
307                 printf("Desea continuar? (0 - Si, 1 - No): ");
308                 op = valiNum(0, 1);
309
310                 if (op == 0)
311                 {
312                     system("CLS");
313                     printf("Registros %d - %d\n\n", activos + 1, (activos + 40) > n ? n : (activos + 40));
314                     printf("-----\n");
315                     printf(" No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO \n");
316                     printf("-----\n");
317                 }
318                 else
319                 {
320                     return;
321                 }
322             }
323         }
324     }
325 }
```

Salida 6:

Registros 1 - 40

No	MATRICULA	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
0.-	313695	LUIS	ZAVALA	BAEZ	18	MASCULINO
1.-	327418	JORGE	CALDERON	VELA	20	MASCULINO
2.-	329890	ALMA	PEREDES	CONTRERAS	20	FEMENINO
3.-	332288	SUSANA	REYES	ESCOBAR	29	FEMENINO
4.-	336299	VICTOR	SALAZAR	FRIAS	28	MASCULINO
5.-	340324	PEDRO	PEREDES	ZAVALA	20	MASCULINO
6.-	380951	CLAUDIO	LEON	CHACON	18	MASCULINO
7.-	383421	CESAR	FUENTES	MARTINEZ	21	MASCULINO
8.-	385402	MIRIAM	PENA	GUZMAN	27	FEMENINO
9.-	387495	ELISA	GARCIA	HERRERA	23	FEMENINO
10.-	313772	ANDREA	DIAZ	GARCIA	19	FEMENINO
11.-	312942	CLAUDIO	MEDINA	CARDENAS	19	MASCULINO
12.-	305844	ROSARIO	TOVAR	SERNA	18	FEMENINO
13.-	376632	SUSANA	FERNANDEZ	MEDINA	23	FEMENINO
14.-	395674	ROSARIO	ECHEVERRIA	GARCIA	20	FEMENINO
15.-	397779	DAMIAN	SERNA	CHACON	20	MASCULINO
16.-	323306	AMADEO	PACHECO	ESPINOSA	21	MASCULINO
17.-	356063	EDUARDO	CHACON	ORTEGA	19	MASCULINO
18.-	348704	ROSARIO	ZAVALA	CASTRO	20	FEMENINO
19.-	395204	SILVIA	BAEZ	REYES	31	FEMENINO
20.-	307796	CESAR	BAUTISTA	LUGO	20	MASCULINO
21.-	329471	ISABEL	OCHOA	VELA	19	FEMENINO
22.-	353527	LORENA	CHACON	HIDALGO	18	FEMENINO
23.-	382742	BENJAMIN	ZAVALA	ECHEVERRIA	30	MASCULINO
24.-	317816	CLEMENTE	MARTINEZ	LEON	31	MASCULINO
25.-	327536	LUCIA	GUTIERREZ	FRIAS	26	FEMENINO
26.-	398437	CAROLINA	FRIAS	PADILLA	27	FEMENINO
27.-	306615	ROSARIO	ZAVALA	PERALTA	29	FEMENINO
28.-	364010	MARIA	PEREDES	NAVA	31	FEMENINO
29.-	382541	GABRIEL	MORALES	CALDERON	19	MASCULINO
30.-	373411	BRUNO	NUNEZ	RIVAS	23	MASCULINO
31.-	306112	LUIS	SERNA	CARRILLO	30	MASCULINO
32.-	335494	ALMA	CORDOVA	OSORIO	30	FEMENINO
33.-	386954	MONICA	TOVAR	MIRANDA	26	FEMENINO
34.-	326958	ELISA	ESCOBAR	OSORIO	24	FEMENINO
35.-	386962	RAQUEL	OSORIO	NAVARRO	29	FEMENINO
36.-	371725	JUANA	MONTES	PEREZ	19	FEMENINO
37.-	323739	GISELA	PACHECO	CONTRERAS	30	FEMENINO
38.-	374496	ISABEL	OSORIO	CISNEROS	25	FEMENINO
39.-	300678	ANDREA	AGUILAR	CORDERO	31	FEMENINO

Desea continuar? (0 - Si, 1 - No):

Registros 42 - 70

No	MATRICULA	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
40.-	341085	ALBERTO	TOVAR	ESPINOSA	28	MASCULINO
41.-	398897	RAUL	RIVAS	CHACON	23	MASCULINO
42.-	330975	LUIS	QUINONES	ECHEVERRIA	26	MASCULINO
43.-	338072	IRENE	CASTILLO	GUTIERREZ	20	FEMENINO
44.-	325174	ALFONSO	CHACON	CARRILLO	31	MASCULINO
45.-	323831	MATEO	LEON	RIVAS	18	MASCULINO
46.-	347950	NURIA	PACHECO	TELLEZ	21	FEMENINO
47.-	342534	GABRIEL	PEREDES	VELA	27	MASCULINO
48.-	375858	CLEMENTE	ROMERO	MEDINA	18	MASCULINO
49.-	396061	ALFONSO	RAMIREZ	ZAVALA	21	MASCULINO
50.-	343267	RAQUEL	ORTEGA	ESPINOZA	24	FEMENINO
51.-	317755	FELIPE	LUGO	AGUILAR	26	MASCULINO
52.-	300180	ANDRES	ECHEVERRIA	TOVAR	26	MASCULINO
53.-	329019	DAMIAN	REYES	VARGAS	26	MASCULINO
54.-	322961	ELENA	VELA	CABELLO	29	FEMENINO
55.-	397322	LOURDES	LUGO	MELLENDEZ	21	FEMENINO
56.-	327404	ANDREA	VILLANUEVA	SERNA	31	FEMENINO
57.-	338761	ELISA	PONCE	OCHOA	20	FEMENINO
58.-	329294	AMADEO	ESCOBAR	MELLENDEZ	24	MASCULINO
59.-	345072	ALMA	CONTRERAS	SALAZAR	19	FEMENINO
60.-	326914	ABELARDO	LUGO	LARA	22	MASCULINO
61.-	326397	ARTURO	GUZMAN	LARA	21	MASCULINO
62.-	387462	JUANA	ROSALES	ZAVALA	24	FEMENINO
63.-	313754	ELISA	GUTIERREZ	DUARTE	23	FEMENINO
64.-	328913	BERNARDO	DUARTE	RUEDA	27	MASCULINO
65.-	333702	ISABEL	BAEZ	NAVA	18	FEMENINO
66.-	354201	MARTIN	CABELLO	DUARTE	25	MASCULINO
67.-	359696	ELISA	DAVILA	ALVARADO	20	FEMENINO
68.-	344141	BEATRIZ	PEREZ	SANCHEZ	26	FEMENINO

Presione una tecla para continuar . . .

Ejercicio 7:

```
194 // Generar archivos
195 case 7:
196     leerNomArch(nomArchivo);
197
198     escrArch(nomArchivo, ingenieros, nPers, false);
199     escrArch(nomArchivo, ingenieros, nPers, true);
200     break;
```

```
356 // Escribe los registros de un vector de personas en un archivo de texto especificado.
357 void escrArch(char nomArchivo[], Tprogra vect[], int n, bool arch)
358 {
359     if (n <= 0)
360     {
361         printf("No hay registros para escribir\n");
362         return;
363     }
364
365     int i, cont = 0;
366
367     FILE *fa;
368
369     char temp[30];
370     strcpy(temp, nomArchivo);
371
372     if (!arch)
373     {
374         strcat(temp, "_activos");
375     }
376
377     strcat(temp, ".txt");
378     fa = fopen(temp, "w");
379
380     if (arch)
381     {
382         fprintf(fa, "-----\n");
383         fprintf(fa, " No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO \n");
384         fprintf(fa, "-----\n");
385     }
```



```

387 for (i = 0; i < n; i++)
388 {
389     if (vect[i].status == 1)
390     {
391         fprintf(fa, "%4d.- %6d      %-10s      %-10s      %-10s      %2d      %-7s", cont, vect[i].matricula,
392             vect[i].nombre, vect[i].apellido, vect[i].telefono, vect[i].edad, vect[i].direccion);
393         if (i < (n - 1))
394         {
395             fprintf(fa, "\n");
396         }
397         cont++;
398     }
399 }
400
401 if (arch)
402 {
403     fprintf(fa, "\n-----\n");
404     fprintf(fa, "-- TODOS LOS DERECHOS RESERVADOS @Scalaptia      www.profeyepiz.com      @2023-2      --\n");
405     fprintf(fa, "-----");
406 }
407
408 fclose(fa);
409 }

```

Salida 7:

Ingresa el nombre del archivo (sin extension): datos

datos_activos.txt
datos.txt

datos.txt

ACTIVIDAD12 > datos.txt							
	No	MATRICULA	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
1							
2							
3							
4	0.-	313695	LUIS	ZAVALA	BAEZ	18	MASCULINO
5	1.-	327418	JORGE	CALDERON	VELA	20	MASCULINO
6	2.-	329890	ALMA	PAREDES	CONTRERAS	20	FEMENINO
7	3.-	332288	SUSANA	REYES	ESCOBAR	29	FEMENINO
8	4.-	336299	VICTOR	SALAZAR	FRIAS	28	MASCULINO
9	5.-	340324	PEDRO	PAREDES	ZAVALA	20	MASCULINO
10	6.-	380951	CLAUDIO	LEON	CHACON	18	MASCULINO
11	7.-	383421	CESAR	FUENTES	MARTINEZ	21	MASCULINO
12	8.-	385402	MIRIAM	PENA	GUZMAN	27	FEMENINO
13	9.-	387495	ELISA	GARCIA	HERRERA	23	FEMENINO
14	10.-	313772	ANDREA	DIAZ	GARCIA	19	FEMENINO
15	11.-	312942	CLAUDIO	MEDINA	CARDENAS	19	MASCULINO
16	12.-	305844	ROSARIO	TOVAR	SERNA	18	FEMENINO
17	13.-	376632	SUSANA	FERNANDEZ	MEDINA	23	FEMENINO
18	14.-	395674	ROSARIO	ECHEVERRIA	GARCIA	20	FEMENINO
19	15.-	397779	DAMIAN	SERNA	CHACON	20	MASCULINO
20	16.-	323306	AMADEO	PACHECO	ESPINOSA	21	MASCULINO
21	17.-	356063	EDUARDO	CHACON	ORTEGA	19	MASCULINO
22	18.-	348704	ROSARIO	ZAVALA	CASTRO	20	FEMENINO
23	19.-	395204	SILVIA	BAEZ	REYES	31	FEMENINO
63	59.-	345072	ALMA	CONTRERAS	SALAZAR	19	FEMENINO
64	60.-	326914	ABELARDO	LUGO	LARA	22	MASCULINO
65	61.-	326397	ARTURO	GUZMAN	LARA	21	MASCULINO
66	62.-	387462	JUANA	ROSALES	ZAVALA	24	FEMENINO
67	63.-	313754	ELISA	GUTIERREZ	DUARTE	23	FEMENINO
68	64.-	328913	BERNARDO	DUARTE	RUEDA	27	MASCULINO
69	65.-	333702	ISABEL	BAEZ	NAVA	18	FEMENINO
70	66.-	354201	MARTIN	CABELLO	DUARTE	25	MASCULINO
71	67.-	359696	ELISA	DAVILA	ALVARADO	20	FEMENINO
72	68.-	344141	BEATRIZ	PEREZ	SANCHEZ	26	FEMENINO
73							
74	--	TODOS LOS DERECHOS RESERVADOS @Scalaptia			www.profeyepiz.com	@2023-2	--
75							

datos_activos.txt

HCF_ACT12_932.c		datos.txt 1, M		datos_activos.txt U x		contadorReg.c		alexandria.h	
ACTIVIDAD12 > datos_activos.txt									
1	0.-	313695	LUIS	ZAVALA	BAEZ	18	MASCULINO		
2	1.-	327418	JORGE	CALDERON	VELA	20	MASCULINO		
3	2.-	329890	ALMA	PAREDES	CONTRERAS	20	FEMENINO		
4	3.-	332288	SUSANA	REYES	ESCOBAR	29	FEMENINO		
5	4.-	336299	VICTOR	SALAZAR	FRIAS	28	MASCULINO		
6	5.-	340324	PEDRO	PAREDES	ZAVALA	20	MASCULINO		
7	6.-	380951	CLAUDIO	LEON	CHACON	18	MASCULINO		
8	7.-	383421	CESAR	FUENTES	MARTINEZ	21	MASCULINO		
9	8.-	385402	MIRIAM	PENA	GUZMAN	27	FEMENINO		
10	9.-	387495	ELISA	GARCIA	HERRERA	23	FEMENINO		
11	10.-	313772	ANDREA	DIAZ	GARCIA	19	FEMENINO		
12	11.-	312942	CLAUDIO	MEDINA	CARDENAS	19	MASCULINO		
13	12.-	305844	ROSARIO	TOVAR	SERNA	18	FEMENINO		
14	13.-	376632	SUSANA	FERNANDEZ	MEDINA	23	FEMENINO		
15	14.-	395674	ROSARIO	ECHEVERRIA	GARCIA	20	FEMENINO		
16	15.-	397779	DAMIAN	SERNA	CHACON	20	MASCULINO		
17	16.-	323306	AMADEO	PACHECO	ESPINOSA	21	MASCULINO		
18	17.-	356063	EDUARDO	CHACON	ORTEGA	19	MASCULINO		
19	18.-	348704	ROSARIO	ZAVALA	CASTRO	20	FEMENINO		
20	19.-	395204	SILVIA	BAEZ	REYES	31	FEMENINO		
21	20.-	307796	CESAR	BAUTISTA	LUGO	20	MASCULINO		
22	21.-	329471	ISABEL	OCHOA	VELA	19	FEMENINO		
23	22.-	353527	LORENA	CHACON	HIDALGO	18	FEMENINO		
24	23.-	382742	BENJAMIN	ZAVALA	ECHEVERRIA	30	MASCULINO		
25	24.-	317816	CLEMENTE	MARTINEZ	LEON	31	MASCULINO		
26	25.-	327536	LUCIA	GUTIERREZ	FRIAS	26	FEMENINO		
27	26.-	398437	CAROLINA	FRIAS	PADILLA	27	FEMENINO		
28	27.-	306615	ROSARIO	ZAVALA	PERALTA	29	FEMENINO		
29	28.-	364010	MARIA	PAREDES	NAVA	31	FEMENINO		
30	29.-	382541	GABRIEL	MORALES	CALDERON	19	MASCULINO		
31	30.-	373411	BRUNO	NUNEZ	RIVAS	23	MASCULINO		
32	31.-	306112	LUIS	SERNA	CARRILLO	30	MASCULINO		
33	32.-	335494	ALMA	CORDOVA	OSORIO	30	FEMENINO		
34	33.-	386954	MONICA	TOVAR	MIRANDA	26	FEMENINO		
35	34.-	326958	ELISA	ESCOBAR	OSORIO	24	FEMENINO		
36	35.-	386962	RAQUEL	OSORIO	NAVARRO	29	FEMENINO		
37	36.-	371725	JUANA	MONTES	PEREZ	19	FEMENINO		
38	37.-	323739	GISELA	PACHECO	CONTRERAS	30	FEMENINO		
39	38.-	374496	ISABEL	OSORIO	CISNEROS	25	FEMENINO		
40	39.-	300678	ANDREA	AGUILAR	CORDERO	31	FEMENINO		
41	40.-	341085	ALBERTO	TOVAR	ESPINOSA	28	MASCULINO		
42	41.-	398897	RAUL	RIVAS	CHACON	23	MASCULINO		
43	42.-	330975	LUIS	QUINONES	ECHEVERRIA	26	MASCULINO		
44	43.-	338072	IRENE	CASTILLO	GUTIERREZ	20	FEMENINO		
45	44.-	325174	ALFONSO	CHACON	CARRILLO	31	MASCULINO		
46	45.-	323831	MATEO	LEON	RIVAS	18	MASCULINO		
47	46.-	347950	NURIA	PACHECO	TELLEZ	21	FEMENINO		
48	47.-	342534	GABRIEL	PAREDES	VELA	27	MASCULINO		

Ejercicio 8:

```
202 // Contar registros
203 case 8:
204     printf("Ingrese el status de los registros que desea contar (1 - Borrados, 2 - Activos): ");
205     status = valiNum(1, 2);
206
207     leerNomArch(nomArchivo);
208     num = contarRegArch(nomArchivo, status);
209
210     system("CLS");
211     if (num == -1)
212     {
213         printf("Archivo no encontrado\n");
214     }
215     else
216     {
217         printf("Hay %d registro(s) en el archivo\n", num);
218     }
219     break;
```

```
488 // Cuenta los registros de un archivo de texto especificado desde el ejecutable contadorReg.
489 int contarRegArch(char nomArchivo[], int status)
490 {
491     int cont;
492     char fileName[50];
493     char cmd[50];
494
495     system("gcc contadorReg.c -o contadorReg");
496     sprintf(cmd, "contadorReg.exe %s %d", nomArchivo, status);
497     cont = system(cmd);
498
499     return cont;
500 }
```

contadorReg.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int main(int argc, char *argv[])
6  {
7      if (argc != 3)
8      {
9          printf("Error en los argumentos\n");
10         system("pause");
11         return -1;
12     }
13
14     FILE *fa;
15     char nomArchivo[30];
16     char linea[90];
17     int cont = 0;
18     int status = atoi(argv[2]);
19     strcpy(nomArchivo, argv[1]);
20
21     if (status == 1)
22     {
23         strcat(nomArchivo, "_borrados.txt");
24     }
25     else
26     {
27         strcat(nomArchivo, "_activos.txt");
28     }
29
30     fa = fopen(nomArchivo, "r");
31     if (fa)
32     {
33         while (!feof(fa))
34         {
35             fgets(linea, 90, fa);
36             if (strlen(linea) > 10)
37             {
38                 cont++;
39             }
40         }
41         fclose(fa);
42     }
43     else
44     {
45         return 0;
46     }
47
48     return cont;
49 }
```

Salida 8:

Borrados (antes de crear archivo):

```
Ingrese el status de los registros que desea contar (1 - Borrados, 2 - Activos): 1
```

```
Ingresa el nombre del archivo (sin extension): datos
```

```
Hay 0 registro(s) en el archivo
```

```
Presione una tecla para continuar . . .
```

Activos:

```
Ingrese el status de los registros que desea contar (1 - Borrados, 2 - Activos): 2
```

```
Ingresa el nombre del archivo (sin extension): datos
```

```
Hay 69 registro(s) en el archivo
```

```
Presione una tecla para continuar . . .
```

Borrados (archivo creado):

```
Ingrese el status de los registros que desea contar (1 - Borrados, 2 - Activos): 1
```

```
Ingresa el nombre del archivo (sin extension): datos
```

```
Hay 1 registro(s) en el archivo
```

```
Presione una tecla para continuar . . .
```

Ejercicio 9:

```
221 // Mostrar registros borrados
222 case 9:
223     imprPersonas(ingenieros, nPers, 0);
224     break;
```

```
286 // Imprime todas las personas en un vector dado.
287 void imprPersonas(Tprogra vect[], int n, int status)
288 {
289     int i, activos, op;
290
291     printf("Registros 1 - 40\n\n");
292     printf("-----\n");
293     printf(" No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO \n");
294     printf("-----\n");
295     for (i = 0, activos = 1; i < n; i++)
296     {
297         if (vect[i].status == status)
298         {
299             printf("%4d.- %6d %-10s %-10s %-10s %2d %-7s\n", activos - 1, vect[i].matricula,
300                 activos++,
301
302
303
304             if (activos % 41 == 0 && activos < n)
305             {
306                 printf("\n\n");
307                 printf("Desea continuar? (0 - Si, 1 - No): ");
308                 op = valiNum(0, 1);
309
310                 if (op == 0)
311                 {
312                     system("CLS");
313                     printf("Registros %d - %d\n\n", activos + 1, (activos + 40) > n ? n : (activos + 40));
314                     printf("-----\n");
315                     printf(" No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO \n");
316                     printf("-----\n");
317                 }
318                 else
319                 {
320                     return;
321                 }
322             }
323         }
324     }
325 }
```

Salida 9:

Registros 1 - 40

No	MATRICULA	NOMBRE	APELLIDO P.	APELLIDO MAT.	EDAD	SEXO
0.-	394778	DIEGO	DUARTE	GARCIA	26	MASCULINO

Presione una tecla para continuar . . . █

Salir:

```
226     case 0:
227         if (strcmp(nomArchivo, "\n") != 0)
228         {
229             escrArch(nomArchivo, ingenieros, nPers, false);
230             escrArch(nomArchivo, ingenieros, nPers, true);
231             actBorrados(nomArchivo, ingenieros, nPers);
232         }
233         else
234         {
235             printf("Desea guardar los datos? (1- Si, 2- No): \n");
236             num = valiNum(1, 2);
237             if (num == 1)
238             {
239                 leerNomArch(nomArchivo);
240
241                 escrArch(nomArchivo, ingenieros, nPers, false);
242                 escrArch(nomArchivo, ingenieros, nPers, true);
243                 actBorrados(nomArchivo, ingenieros, nPers);
244             }
245         }
246         break;
247     }
```

Salida:

Archivo nombrado:

```
Presione una tecla para continuar . . . █
```

Archivo no nombrado:

```
Desea guardar los datos? (1- Si, 2- No):  
1 █
```

```
Ingresa el nombre del archivo (sin extension): data  
Presione una tecla para continuar . . . █
```

```
data_activos.txt  
data_borrados.txt  
data.txt
```