



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Nombre del alumno:

Luis Fernando Ochoa Angulo

Matricula:

372746

Semestre:

3ro- Computación

Grupo:

432

Materia:

Programación Estructurada

Actividad:

Actividad 13

Nombre del Profesor:

Pedro Nuñez Yepiz

Fecha: Jueves 23 de Octubre del 2023

```

1 //                               Ensenada, Baja California a 23 de Noviembre del 2023 //
2 // Nombre:                       Luis Fernando Ochoa Angulo //
3 // Matricula:                     372746 //
4 // Programa:                     Crea registros con archivos binarios, desde cargar y crear //
5 //                               archivos de textos //
6 // Nombre del programa: LFOA_ACTV_13 //
7
8
9 #include "8a.h"
10 #define REG 5000
11 void menu(void);
12 int msgs(void);
13
14 //Funciones para la opcion 1
15 int genReg(Tdato vect[],int i);
16 void genRegRand(Tdato vect[],int i);
17 int buscarTelefono(Tdato vect[],int i,double num);
18 void genPuesto(Tdato vect[],int i);
19
20
21 //Funciones para la opcion 2
22 void editarReg(Tdato vect[],int i,int orden);
23 void editMenu(Tdato vect[],int pos);
24 void editPuesto(Tdato vect[],int pos);
25 float validNumF(char txt[],int limi,int lims);
26
27
28 //Funciones para la opcion 3
29 int eliminarEmpleado(Tdato vect[],int i,int orden);
30
31 //Funciones para la opcion 4
32 void buscarEmpleado(Tdato vect[],int i,int orden);
33 void printEmpleadoReg(Tdato empleado);
34 //Funciones para la opcion 5
35 int ordenEmpleado(Tdato vect[],int i,int ordenM);
36 //Funciones para la opcion 6
37 void printEmpleados(Tdato vect[],int i);
38 void printEmpleadoTab(Tdato empleado);
39 //Funcione para la opcion 7
40 void genArchivoTxt(Tdato vect[],int i);
41 //Funcione para la opcion 8
42 void printArchTxt(void);
43 //Funcione para la opcion 9
44 void genArchivoBin(Tdato vect[],int i);
45 //Funcione para la opcion 10
46 int cargarArchivoBin(Tdato vect[],int i);
47 //Funcione para la opcion 11
48 void printDel(void);
49
50 int main()
51 {
52     srand(time(NULL));
53     menu();
54     return 0;
55 }
56
57 int msgs(void)
58 {
59     printf(" Menu\n\n");
60     printf(" 1 Agregar\n");
61     printf(" 2 Editar Reistro\n");
62     printf(" 3 Eliminar Registro\n");
63     printf(" 4 Buscar\n");
64     printf(" 5 Ordenar\n");
65     printf(" 6 Imprimir\n");
66     printf(" 7 Generar Archivo Texto\n");
67     printf(" 8 Ver Archivo Texto\n");
68     printf(" 9 Crear Archivo Binario\n");
69     printf("10 Cargar Archivo Binario\n");
70     printf("11 Mostrar Eliminados\n");
71     printf("12 Salir\n");
72     return (validNum("Escoge una opcion: ",1,12));
73 }
74
75 void menu(void)
76 {

```

```

77 FILE *fa;
78 Tdato empleados[REG];
79 int op;
80 int orden=0,ordenMetodo=0,cargBin=1,del=0;
81 int i=0;
82 /*fa=fopen("registrosbin.dll","rb");
83 if(fa)
84 {
85     fclose(fa);
86     remove("registrosbin.dll");
87 }*/
88
89 do
90 {
91     op=msgsg();
92     switch(op)
93     {
94         case 1:
95             i=genReg(empleados,i);
96             if(i>2500)
97             {
98                 ordenMetodo=1;
99             }
100             orden=0;
101             break;
102         case 2:
103             editarReg(empleados,i,orden);
104             break;
105         case 3:
106             del+=eliminarEmpleado(empleados,i,orden);
107             break;
108         case 4:
109             buscarEmpleado(empleados,i,orden);
110             break;
111         case 5:
112             if(!orden)
113             {
114                 orden=ordenEmpleado(empleados,i,ordenMetodo);
115             }
116             else
117             {
118                 printf("\nLos empleados ya estan ordenados\n");
119             }
120             break;
121         case 6:
122             printEmpleados(empleados,i);
123             break;
124         case 7:
125             genArchivoTxt(empleados,i);
126             break;
127         case 8:
128             printArchTxt();
129             break;
130         case 9:
131             genArchivoBin(empleados,i);
132             break;
133         case 10:
134             if(cargBin)
135             {
136                 i=cargarArchivoBin(empleados,i);
137                 if(i>2500)
138                 {
139                     ordenMetodo=1;
140                 }
141                 orden=0;
142                 cargBin=0;
143             }
144             else
145             {
146                 printf("\nYa se ha cargado el archivo binario!\nRealice otra opcion!\n");
147             }
148             break;
149         case 11:
150             if(del>0)
151             {
152                 printDel();
153             }
154     }

```

```

154         else
155         {
156             printf("\nAun no hay empleados eliminados\n");
157         }
158         break;
159     }
160
161     printf("\n");
162 }while(op!=12);
163
164 /*fa=fopen("datosbin.tmp","rb");
165 if(fa)
166 {
167     fclose(fa);
168     remove("datosbin.tmp");
169 }*/
170 }
171
172
173
174 //case 1
175 int genReg(Tdato vect[],int i)
176 {
177     int j;
178     if(i+100<=REG)
179     {
180         for(j=0;j<100;j++)
181         {
182             genRegRand(vect,i++);
183         }
184         printf("\nSe generaron 100 registros\n");
185     }
186     else
187     {
188         if(i==REG)
189         {
190             printf("\nNo se pueden generar mas registros\n");
191         }
192         else
193         {
194             if(i+100>REG)
195             {
196                 for(i;i<REG;i++)
197                 {
198                     genRegRand(vect,i);
199                 }
200                 printf("\nNo se pueden generar 100 registros\n");
201                 printf("\nSe generaron registros hasta llegar al limite\n");
202             }
203         }
204     }
205     return i;
206 }
207
208 void genRegRand(Tdato vect[],int i)
209 {
210     int s;
211     int j;
212     double num;
213     vect[i].status=1;
214     vect[i].key=genKey(vect,i,300000,399999);
215
216     for(j=0;j<10;j++)
217     {
218         s=rand()%2;
219     }
220
221     if(s)
222     {
223         vect[i].sexo='H';
224     }
225     else
226     {
227         vect[i].sexo='M';
228     }
229
230     if(s)
231     {

```

```

232     genH_Name(vect,i);
233 }
234 else
235 {
236     genM_Name(vect,i);
237 }
238
239 genAp(vect,i);
240
241 do
242 {
243     num=6460000000+(rand()%100000)+(rand()%99)*100000;
244 }while(buscarTelefono(vect,i,num)!=-1);
245 vect[i].telefono=num;
246
247 genPuesto(vect,i);
248
249 }
250
251 void genPuesto(Tdato vect[],int i)
252 {
253     char pst[12][L]={
254         "OPERADOR","SUPERVISOR","MANTENIMIENTO","INGENIERO DE PROCESOS","INGENIERO DE CALIDAD",
255         "OPERADOR DE MAQUINARIA", "ALMACENISTA", "JEFE DE ALMACEN", "INSPECTOR DE CALIDAD",
256         "GERENTE DE PRODUCCION", "TECNICO EN SEGURIDAD", "RECURSOS HUMANOS"
257     };
258     strcpy(vect[i].puesto,pst[(rand()%12)]);
259 }
260
261 int buscarTelefono(Tdato vect[],int i,double num)
262 {
263     int j;
264     for(j=0;j<i;j++)
265     {
266         if(vect[j].telefono==num)
267         {
268             return j;
269         }
270     }
271     return -1;
272 }
273
274 //case 2
275 void editarReg(Tdato vect[],int i,int orden)
276 {
277     int pos;
278     int edit;
279     if(i>0)
280     {
281         printf("Editar empleado\n\n");
282         pos=validNum("Ingrese el numero de emplado que desea editar: ",300000,399999);
283         if(orden)
284         {
285             pos=buscarTBin(vect,i,pos);
286             if(pos!=-1)
287             {
288                 if(vect[pos].status)
289                 {
290                     printEmpleadoReg(vect[pos]);
291                     editMenu(vect,pos);
292                 }
293                 else
294                 {
295                     printf("\nEse empleado fue eliminado de los registros\n");
296                 }
297             }
298             else
299             {
300                 printf("\nEse empleado no existe\n");
301             }
302         }
303         else
304         {
305             pos=buscarTSec(vect,i,pos);
306             if(pos!=-1)
307             {
308                 if(vect[pos].status)

```

```

309         {
310             printEmpleadoReg(vect[pos]);
311             editMenu(vect,pos);
312         }
313         else
314         {
315             printf("\nEse empleado fue eliminado de los registros\n");
316         }
317     }
318     else
319     { //322010
320         printf("\nEse empleado no existe\n");
321     }
322 }
323 }
324 else
325 {
326     printf("\nAun no hay empleados registrados\n");
327 }
328 }
329
330 void editMenu(Tdato vect[],int pos)
331 {
332     int op;
333     int s;
334     int j;
335     char num[11];
336     printf("\nEditar\n\n");
337     printf("1) Puesto\n");
338     printf("2) Nombre\n");
339     printf("3) Apellido Paterno\n");
340     printf("4) Apellido Materno\n");
341     printf("5) Telefono\n");
342     printf("6) Sexo\n");
343     printf("7) Salir\n");
344     op=validNum("Esoge una opcion: ",1,7);
345     switch(op)
346     {
347         case 1:
348             printf("Editar puesto\n\n");
349             editPuesto(vect,pos);
350             printf("\nSe edito el campo deseado\n");
351             break;
352         case 2:
353             printf("Editar nombre\n\nIngresa el nuevo nombre: ");
354             validStr(vect[pos].nombre);
355             printf("\nSe edito el campo deseado\n");
356             break;
357         case 3:
358             printf("Editar apellido paterno\n\nIngresa el nuevo apellido: ");
359             validStr(vect[pos].apPat);
360             printf("\nSe edito el campo deseado\n");
361             break;
362         case 4:
363             printf("Editar apellido materno\n\nIngresa el nuevo apellido: ");
364             validStr(vect[pos].apMat);
365             printf("\nSe edito el campo deseado\n");
366             break;
367         case 5:
368             printf("Editar telefono\n\n");
369             vect[pos].telefono+=validNumF("Ingresa el telefono: 646",0,9999999);
370             printf("\nSe edito el campo deseado\n");
371             break;
372         case 6:
373             printf("Editar sexo\n\n");
374             s=validNum("(1:Hombre 2:Mujer) Elige una opcion: ",1,2);
375             if(s==1)
376             {
377                 vect[pos].sexo='H';
378             }
379             else
380             {
381                 vect[pos].sexo='M';
382             }
383             printf("\nSe edito el campo deseado\n");
384             break;
385     }
386 }

```

```

---
387 }
388 //308895 312946
389 void editPuesto(Tdato vect[],int pos)
390 {
391     char pst[12][L]={
392         "OPERADOR","SUPERVISOR","MANTENIMIENTO","INGENIERO DE PROCESOS","INGENIERO DE CALIDAD",
393         "OPERADOR DE MAQUINARIA","ALMACENISTA","JEFE DE ALMACEN","INSPECTOR DE CALIDAD",
394         "GERENTE DE PRODUCCION","TECNICO EN SEGURIDAD","RECURSOS HUMANOS"
395     };
396
397     printf("1) Operador\n");
398     printf("2) Supervisor\n");
399     printf("3) Mantenimiento\n");
400     printf("4) Ingeniero de Procesos\n");
401     printf("5) Ingeniero de Calidad\n");
402     printf("6) Operador de Maquinaria\n");
403     printf("7) Almacenista\n");
404     printf("8) Jefe de Almacen\n");
405     printf("9) Inspector de Calidad\n");
406     printf("10) Gerente de Produccion\n");
407     printf("11) Tecnico en Seguridad\n");
408     printf("12) Recursos Humanos\n");
409     strcpy(vect[pos].puesto,pst[validNum("Escoge una opcion: ",1,12)-1]);
410 }
411
412 float validNumF(char txt[],int limi,int lims)
413 {
414     char r[STR];
415     int rfloat;
416     do
417     {
418         printf("%s",txt);
419         fflush(stdin);
420         gets(r);
421         rfloat=atof(r);
422         if(rfloat<limi || rfloat>lims)
423         {
424             printf("Valor invalido, ingresalo de nuevo\n");
425         }
426     }while(rfloat<limi || rfloat>lims);
427     return rfloat;
428 }
429
430 //case 3
431 int eliminarEmpleado(Tdato vect[],int i,int orden)
432 {
433     int pos;
434     int del;
435     if(i>0)
436     {
437         printf("Eliminar empleado\n\n");
438         pos=validNum("Ingresa el numero de empleado que desea eliminar: ",300000,399999);
439         if(orden)
440         {
441             pos=buscarTBin(vect,i,pos);
442             if(pos!=-1)
443             {
444                 if(vect[pos].status)
445                 {
446                     printEmpleadoReg(vect[pos]);
447                     printf("\nDesea eliminar al empleado? (1:Si 2:No)\n");
448                     del=validNum("Escoge una opcion: ",1,2);
449                     if(del==1)
450                     {
451                         vect[pos].status=0;
452                         return 1;
453                     }
454                 }
455                 else
456                 {
457                     printf("\nEse empleado ya fue eliminado de los registros\n");
458                 }
459             }
460             else
461             {
462                 printf("\nEse empleado no existe\n");
463             }
464         }
465     }

```

```

464     }
465     else
466     {
467         pos=buscarTSec(vect,i,pos);
468         if(pos!=-1)
469         {
470             if(vect[pos].status)
471             {
472                 printEmpleadoReg(vect[pos]);
473                 printf("\nDesea eliminar al empleado? (1:Si 2:No)\n");
474                 del=validNum("Escoge una opcion: ",1,2);
475                 if(del==1)
476                 {
477                     vect[pos].status=0;
478                     return 1;
479                 }
480             }
481             else
482             {
483                 printf("\nEse empleado ya fue eliminado de los registros\n");
484             }
485         }
486         else
487         {
488             printf("\nEse empleado no existe\n");
489         }
490     }
491 }
492 else
493 {
494     printf("\nAun no hay empleados registrados\n");
495 }
496 return 0;
497 }
498
499 //case 4
500 void buscarEmpleado(Tdato vect[],int i,int orden)
501 {
502     int pos;
503     if(i>0)
504     {
505         printf("Buscar empleado\n\n");
506         pos=validNum("Ingrese el numero de empleado que desea buscar: ",300000,399999);
507         if(orden)
508         {
509             pos=buscarTBin(vect,i,pos);
510             if(pos!=-1)
511             {
512                 if(vect[pos].status)
513                 {
514                     printEmpleadoReg(vect[pos]);
515                 }
516                 else
517                 {
518                     printf("\nEse empleado fue eliminado de los registros\n");
519                 }
520             }
521             else
522             {
523                 printf("\nEse empleado no existe\n");
524             }
525         }
526         else
527         {
528             pos=buscarTSec(vect,i,pos);
529             if(pos!=-1)
530             {
531                 if(vect[pos].status)
532                 {
533                     printEmpleadoReg(vect[pos]);
534                 }
535                 else
536                 {
537                     printf("\nEse empleado fue eliminado de los registros\n");
538                 }
539             }
540             else
541             {

```



```

542         printf("\nEse empleado no existe\n");
543     }
544 }
545 }
546 else
547 {
548     printf("\nAun no hay empleados registrados\n");
549 }
550 }
551
552 void printEmpleadoReg(Tdato empleado)
553 {
554     printf("Status: %d\n",empleado.status);
555     printf("\nNo. Empleado: %d\n",empleado.key);
556     printf("Puesto: %s\n",empleado.puesto);
557     printf("Apellido Paterno: %s\n",empleado.apPat);
558     printf("Apellido Materno: %s\n",empleado.apMat);
559     printf("Nombre: %s\n",empleado.nombre);
560     printf("Telefono: %.0f\n",empleado.telefono);
561     printf("Sexo: ");
562     if(empleado.sexo=='H')
563     {
564         printf("MASCULINO");
565     }
566     else
567     {
568         printf("FEMENINO");
569     }
570     printf("\n");
571 }
572
573 //case 5
574 int ordenEmpleado(Tdato vect[],int i,int ordenM)
575 {
576     if(i>0)
577     {
578         if(ordenM)
579         {
580             burbuja(vect,i);
581         }
582         else
583         {
584             seleccion(vect,i);
585         }
586         printf("\nSe ordenaron los empleados\n");
587         return 1;
588     }
589     else
590     {
591         printf("\nAun no hay empleado registrados\n");
592         return 0;
593     }
594 }
595
596
597 //case 6
598 void printEmpleados(Tdato vect[],int i)
599 {
600     int j,k=0;
601     if(i>0)
602     {
603         printf("| No |STATUS|No. Empleado| Puesto | Apellido Paterno | Apellido Materno | Nombre | Telefó
604         for(j=0;j<i;j++)
605         {
606             printf("| %4d ",k);
607             printEmpleadoTab(vect[j]);
608             k++;
609         }
610     }
611     else
612     {
613         printf("\nAun no hay empleado registrados\n");
614     }
615 }
616 }
617
618 void printEmpleadoTab(Tdato empleado)

```

```

619 {
620     printf("| %4d |%11d | %-22s | %-16s | %-17s| %-20s | %.0f |", empleado.status, empleado.key, empleado.puesto, empleado.apPat, empleado.apMi;
621     if(empleado.sexo=='H')
622     {
623         printf(" MASCULINO |\n");
624     }
625     else
626     {
627         printf(" FEMENINO  |\n");
628     }
629 }
630
631 //case 7
632 void genArchivoTxt(Tdato vect[],int i)
633 {
634     FILE *fa;
635     int j,k;
636     int len;
637     char name[L];
638     printf("Generar archivo\n\n");
639     if(i>0)
640     {
641         printf("Ingresa el nombre del archivo de texto: ");
642         gets(name);
643         len=strlen(name);
644         name[len++]='.';
645         name[len++]='t';
646         name[len++]='x';
647         name[len++]='t';
648         name[len]='\0';
649
650         fa=fopen(name,"r");
651         if(!fa)
652         {
653             fa=fopen(name,"a");
654             fprintf(fa,"| No  |STATUS| No.Empleado |          Puesto          | Apellido Paterno | Apellido Materno | Nombre
655             k=0;
656             for(j=0;j<i;j++)
657             {
658                 if(vect[j].status)
659                 {
660                     fprintf(fa,"| %4d ",k);
661                     fprintf(fa,"| %4d | %11d | %-22s | %-16s | %-17s | %-20s | %.0f |", vect[j].status, vect[j].key, vect[j].puesto, vect[j].apl
662                     if(vect[j].sexo=='H')
663                     {
664                         fprintf(fa," MASCULINO |\n");
665                     }
666                     else
667                     {
668                         fprintf(fa," FEMENINO  |\n");
669                     }
670                     k++;
671                 }
672             }
673             printf("\nSe genero el archivo: %s\n",name);
674             fclose(fa);
675         }
676         else
677         {
678             fclose(fa);
679             printf("\nYa existe un archivo con ese nombre\n");
680         }
681     }
682     else
683     {
684         printf("\nAun no hay empleados registrados\n");
685     }
686 }
687
688 //case 8
689 void printArchTxt(void)
690 {
691     FILE *fa;
692     char name[L];
693     int len;
694     char c;
695     printf("Mostrar archivo\n\n");

```

```

696     printf("ingresa el nombre del archivo de texto: ");
697     gets(name);
698     len=strlen(name);
699     name[len++]='.';
700     name[len++]='t';
701     name[len++]='x';
702     name[len++]='t';
703     name[len]='\0';
704
705     fa=fopen(name, "r");
706     if(fa)
707     {
708         printf("\n %s\n\n",name);
709         do
710         {
711             c=fgetc(fa);
712             printf("%c",c);
713         }while(!feof(fa));
714         fclose(fa);
715     }
716     else
717     {
718         printf("\nEl nombre no coincide con un archivo existente\n");
719     }
720 }
721
722 //case 9
723 void genArchivoBin(Tdato vect[],int i)
724 {
725     FILE *fa;
726     int j;
727     if(i>0)
728     {
729         fa=fopen("datosbin.dll", "rb");
730         if(fa)
731         {
732             fclose(fa);
733             fa=fopen("datosbin.tmp", "rb");
734             if(fa)
735             {
736                 remove("datosbin.tmp");
737                 rename("datosbin.dll", "datosbin.tmp");
738                 fa=fopen("datosbin.dll", "wb");
739                 for(j=0;j<i;j++)
740                 {
741                     fwrite(&vect[j],sizeof(Tdato),1,fa);
742                 }
743                 fclose(fa);
744             }
745             else
746             {
747                 rename("datosbin.dll", "datosbin.tmp");
748                 fa=fopen("datosbin.dll", "wb");
749                 for(j=0;j<i;j++)
750                 {
751                     fwrite(&vect[j],sizeof(Tdato),1,fa);
752                 }
753                 fclose(fa);
754             }
755         }
756     }
757     else
758     {
759         fa=fopen("datosbin.dll", "ab");
760         for(j=0;j<i;j++)
761         {
762             fwrite(&vect[j],sizeof(Tdato),1,fa);
763         }
764         fclose(fa);
765     }
766     printf("\nSe genero el archivo binario\n");
767 }
768 else
769 {
770     printf("\nAun no hay empleados registrados\n");
771 }
772
773 //case 10

```

```

774 int cargarArchivoBin(Tdato vect[],int i)
775 {
776     FILE *fa;
777     int j;
778     fa=fopen("datosbin.dll","rb");
779     if(fa)
780     {
781         j=0;
782         while(!feof(fa))
783         {
784             fread(&vect[j],sizeof(vect[j]),1,fa);
785             j++;
786         }
787         printf("\nSe cargo el archivo binario datos.dll a los registros\n");
788         fclose(fa);
789         return j;
790     }
791     else
792     {
793         printf("\nEl archivo binario a cargar no existe\n");
794     }
795     return 0;
796 }
797
798 //case 11
799 void printDel(void)
800 {
801     FILE *fa;
802     Tdato reg;
803     int k;
804     fa=fopen("datosbin.dll","rb");
805     if(fa)
806     {
807         printf("\nEmpleados eliminados\n\n");
808         printf("| No |STATUS| No.Empleado | Puesto | Apellido Paterno | Apellido Materno | Nombre | Tele
809         k=0;
810         while(!feof(fa))
811         {
812             fread(&reg,sizeof(Tdato),1,fa);
813             if(reg.status != 1)
814             {
815                 printf("| %4d ",k);
816                 printEmpleadoTab(reg);
817                 k++;
818             }
819         }
820         fclose(fa);
821     }
822     else
823     {
824         printf("\nAun no hay un archivo binario\n");
825     }
826 }//
827 // Nombre: Luis Fernando Ochoa Angulo //
828 // Matricula: 372746 //
829 // Programa: Crea registros con archivos binarios, desde cargar y crear //
830 // archivos de textos //
831 // Nombre del programa: LFOA_ATCV_13
832
833
834 #include "8a.h"
835 #define REG 5000
836 void menu(void);
837 int msgs(void);
838
839 //Funciones para la opcion 1
840 int genReg(Tdato vect[],int i);
841 void genRegRand(Tdato vect[],int i);
842 int buscarTelefono(Tdato vect[],int i,double num);
843 void genPuesto(Tdato vect[],int i);
844
845
846 //Funciones para la opcion 2
847 void editarReg(Tdato vect[],int i,int orden);
848 void editMenu(Tdato vect[],int pos);
849 void editPuesto(Tdato vect[],int pos);
850 float validNumF(char txt[],int limi,int lims);

```

```

851
852
853 //Funciones para la opcion 3
854 int eliminarEmpleado(Tdato vect[],int i,int orden);
855
856 //Funciones para la opcion 4
857 void buscarEmpleado(Tdato vect[],int i,int orden);
858 void printEmpleadoReg(Tdato empleado);
859 //Funciones para la opcion 5
860 int ordenEmpleado(Tdato vect[],int i,int ordenM);
861 //Funciones para la opcion 6
862 void printEmpleados(Tdato vect[],int i);
863 void printEmpleadoTab(Tdato empleado);
864 //Funcione para la opcion 7
865 void genArchivoTxt(Tdato vect[],int i);
866 //Funcione para la opcion 8
867 void printArchTxt(void);
868 //Funcione para la opcion 9
869 void genArchivoBin(Tdato vect[],int i);
870 //Funcione para la opcion 10
871 int cargarArchivoBin(Tdato vect[],int i);
872 //Funcione para la opcion 11
873 void printDel(void);
874
875 int main()
876 {
877     srand(time(NULL));
878     menu();
879     return 0;
880 }
881
882 int msgs(void)
883 {
884     printf(" Menu\n\n");
885     printf(" 1 Agregar\n");
886     printf(" 2 Editar Reistro\n");
887     printf(" 3 Eliminar Registro\n");
888     printf(" 4 Buscar\n");
889     printf(" 5 Ordenar\n");
890     printf(" 6 Imprimir\n");
891     printf(" 7 Generar Archivo Texto\n");
892     printf(" 8 Ver Archivo Texto\n");
893     printf(" 9 Crear Archivo Binario\n");
894     printf("10 Cargar Archivo Binario\n");
895     printf("11 Mostrar Eliminados\n");
896     printf("12 Salir\n");
897     return (validNum("Escoge una opcion: ",1,12));
898 }
899
900 void menu(void)
901 {
902     FILE *fa;
903     Tdato empleados[REG];
904     int op;
905     int orden=0,ordenMetodo=0,cargBin=1,del=0;
906     int i=0;
907     /*fa=fopen("registrosbin.dll","rb");
908     if(fa)
909     {
910         fclose(fa);
911         remove("registrosbin.dll");
912     }*/
913
914     do
915     {
916         op=msgs();
917         switch(op)
918         {
919             case 1:
920                 i=genReg(empleados,i);
921                 if(i>2500)
922                 {
923                     ordenMetodo=1;
924                 }
925                 orden=0;
926                 break;
927             case 2:
928                 editarReg(empleados,i,orden);

```

```

929         break;
930     case 3:
931         del+=eliminarEmpleado(empleados,i,orden);
932         break;
933     case 4:
934         buscarEmpleado(empleados,i,orden);
935         break;
936     case 5:
937         if(!orden)
938         {
939             orden=ordenEmpleado(empleados,i,ordenMetodo);
940         }
941         else
942         {
943             printf("\nLos empleados ya estan ordenados\n");
944         }
945         break;
946     case 6:
947         printEmpleados(empleados,i);
948         break;
949     case 7:
950         genArchivoTxt(empleados,i);
951         break;
952     case 8:
953         printArchTxt();
954         break;
955     case 9:
956         genArchivoBin(empleados,i);
957         break;
958     case 10:
959         if(cargBin)
960         {
961             i=cargarArchivoBin(empleados,i);
962             if(i>2500)
963             {
964                 ordenMetodo=1;
965             }
966             orden=0;
967             cargBin=0;
968         }
969         else
970         {
971             printf("\nYa se ha cargado el archivo binario!\nRealice otra opcion!\n");
972         }
973         break;
974     case 11:
975         if(del>0)
976         {
977             printDel();
978         }
979         else
980         {
981             printf("\nAun no hay empleados eliminados\n");
982         }
983         break;
984     }
985
986     printf("\n");
987 }while(op!=12);
988
989 /*fa=fopen("datosbin.tmp","rb");
990 if(fa)
991 {
992     fclose(fa);
993     remove("datosbin.tmp");
994 }*/
995 }
996
997
998
999 //case 1
1000 int genReg(Tdato vect[],int i)
1001 {
1002     int j;
1003     if(i+100<=REG)
1004     {
1005         for(j=0;j<100;j++)

```

```

1006     {
1007         genRegRand(vect,i++);
1008     }
1009     printf("\nSe generaron 100 registros\n");
1010 }
1011 else
1012 {
1013     if(i==REG)
1014     {
1015         printf("\nNo se pueden generar mas registros\n");
1016     }
1017     else
1018     {
1019         if(i+100>REG)
1020         {
1021             for(i;i<REG;i++)
1022             {
1023                 genRegRand(vect,i);
1024             }
1025             printf("\nNo se pueden generar 100 registros\n");
1026             printf("\nSe generaron registros hasta llegar al limite\n");
1027         }
1028     }
1029 }
1030 return i;
1031 }
1032
1033 void genRegRand(Tdato vect[],int i)
1034 {
1035     int s;
1036     int j;
1037     double num;
1038     vect[i].status=1;
1039     vect[i].key=genKey(vect,i,300000,399999);
1040
1041     for(j=0;j<10;j++)
1042     {
1043         s=rand()%2;
1044     }
1045
1046     if(s)
1047     {
1048         vect[i].sexo='H';
1049     }
1050     else
1051     {
1052         vect[i].sexo='M';
1053     }
1054
1055     if(s)
1056     {
1057         genH_Name(vect,i);
1058     }
1059     else
1060     {
1061         genM_Name(vect,i);
1062     }
1063
1064     genAp(vect,i);
1065
1066     do
1067     {
1068         num=6460000000+(rand()%100000)+(rand()%99)*100000;
1069     }while(buscarTelefono(vect,i,num)!=-1);
1070     vect[i].telefono=num;
1071
1072     genPuesto(vect,i);
1073
1074 }
1075
1076 void genPuesto(Tdato vect[],int i)
1077 {
1078     char pst[12][L]={
1079         "OPERADOR","SUPERVISOR","MANTENIMIENTO","INGENIERO DE PROCESOS","INGENIERO DE CALIDAD",
1080         "OPERADOR DE MAQUINARIA","ALMACENISTA","JEFE DE ALMACEN","INSPECTOR DE CALIDAD",
1081         "GERENTE DE PRODUCCION","TECNICO EN SEGURIDAD","RECURSOS HUMANOS"
1082     };
1083     genPuesto(vect,i);

```

```

1083     strcpy(vect[i].puesto,psc1(1*anu(1/12)));
1084 }
1085
1086 int buscarTelefono(Tdato vect[],int i,double num)
1087 {
1088     int j;
1089     for(j=0;j<i;j++)
1090     {
1091         if(vect[j].telefono==num)
1092         {
1093             return j;
1094         }
1095     }
1096     return -1;
1097 }
1098
1099 //case 2
1100 void editarReg(Tdato vect[],int i,int orden)
1101 {
1102     int pos;
1103     int edit;
1104     if(i>0)
1105     {
1106         printf("Editar empleado\n\n");
1107         pos=validNum("Ingrese el numero de emplado que desea editar: ",300000,399999);
1108         if(orden)
1109         {
1110             pos=buscarTBin(vect,i,pos);
1111             if(pos!=-1)
1112             {
1113                 if(vect[pos].status)
1114                 {
1115                     printEmpleadoReg(vect[pos]);
1116                     editMenu(vect,pos);
1117                 }
1118                 else
1119                 {
1120                     printf("\nEse empleado fue eliminado de los registros\n");
1121                 }
1122             }
1123             else
1124             {
1125                 printf("\nEse empleado no existe\n");
1126             }
1127         }
1128         else
1129         {
1130             pos=buscarTSec(vect,i,pos);
1131             if(pos!=-1)
1132             {
1133                 if(vect[pos].status)
1134                 {
1135                     printEmpleadoReg(vect[pos]);
1136                     editMenu(vect,pos);
1137                 }
1138                 else
1139                 {
1140                     printf("\nEse empleado fue eliminado de los registros\n");
1141                 }
1142             }
1143             else
1144             {
1145                 printf("\nEse empleado no existe\n");
1146             }
1147         }
1148     }
1149     else
1150     {
1151         printf("\nAun no hay empleados registrados\n");
1152     }
1153 }
1154
1155 void editMenu(Tdato vect[],int pos)
1156 {
1157     int op;
1158     int s;
1159     int j;
1160     char num[11];

```



```

1161     printf("\nEditar\n\n");
1162     printf("1) Puesto\n");
1163     printf("2) Nombre\n");
1164     printf("3) Apellido Paterno\n");
1165     printf("4) Apellido Materno\n");
1166     printf("5) Telefono\n");
1167     printf("6) Sexo\n");
1168     printf("7) Salir\n");
1169     op=validNum("Esoge una opcion: ",1,7);
1170     switch(op)
1171     {
1172         case 1:
1173             printf("Editar puesto\n\n");
1174             editPuesto(vect,pos);
1175             printf("\nSe edito el campo deseado\n");
1176             break;
1177         case 2:
1178             printf("Editar nombre\n\nIngresa el nuevo nombre: ");
1179             validStr(vect[pos].nombre);
1180             printf("\nSe edito el campo deseado\n");
1181             break;
1182         case 3:
1183             printf("Editar apellido paterno\n\nIngresa el nuevo apellido: ");
1184             validStr(vect[pos].apPat);
1185             printf("\nSe edito el campo deseado\n");
1186             break;
1187         case 4:
1188             printf("Editar apellido materno\n\nIngresa el nuevo apellido: ");
1189             validStr(vect[pos].apMat);
1190             printf("\nSe edito el campo deseado\n");
1191             break;
1192         case 5:
1193             printf("Editar telefono\n\n");
1194             vect[pos].telefono+=validNumF("Ingresa el telefono: 646",0,9999999);
1195             printf("\nSe edito el campo deseado\n");
1196             break;
1197         case 6:
1198             printf("Editar sexo\n\n");
1199             s=validNum("(1:Hombre 2:Mujer) Elige una opcion: ",1,2);
1200             if(s==1)
1201             {
1202                 vect[pos].sexo='H';
1203             }
1204             else
1205             {
1206                 vect[pos].sexo='M';
1207             }
1208             printf("\nSe edito el campo deseado\n");
1209             break;
1210     }
1211 }
1212 }
1213 //308895 312946
1214 void editPuesto(Tdato vect[],int pos)
1215 {
1216     char pst[12][L]={
1217         "OPERADOR","SUPERVISOR","MANTENIMIENTO","INGENIERO DE PROCESOS","INGENIERO DE CALIDAD",
1218         "OPERADOR DE MAQUINARIA", "ALMACENISTA", "JEFE DE ALMACEN", "INSPECTOR DE CALIDAD",
1219         "GERENTE DE PRODUCCION", "TECNICO EN SEGURIDAD", "RECURSOS HUMANOS"
1220     };
1221
1222     printf("1) Operador\n");
1223     printf("2) Supervisor\n");
1224     printf("3) Mantenimiento\n");
1225     printf("4) Ingeniero de Procesos\n");
1226     printf("5) Ingeniero de Calidad\n");
1227     printf("6) Operador de Maquinaria\n");
1228     printf("7) Almacenista\n");
1229     printf("8) Jefe de Almacen\n");
1230     printf("9) Inspector de Calidad\n");
1231     printf("10) Gerente de Produccion\n");
1232     printf("11) Tecnico en Seguridad\n");
1233     printf("12) Recursos Humanos\n");
1234     strcpy(vect[pos].puesto,pst[validNum("Escoge una opcion: ",1,12)-1]);
1235 }
1236
1237 float validNumF(char txt[],int limi,int lims)
1238 {

```

```

1238 {
1239     char r[STR];
1240     int rfloat;
1241     do
1242     {
1243         printf("%s",txt);
1244         fflush(stdin);
1245         gets(r);
1246         rfloat=atof(r);
1247         if(rfloat<limi || rfloat>lims)
1248         {
1249             printf("Valor invalido, ingresalo de nuevo\n");
1250         }
1251     }while(rfloat<limi || rfloat>lims);
1252     return rfloat;
1253 }
1254
1255 //case 3
1256 int eliminarEmpleado(Tdato vect[],int i,int orden)
1257 {
1258     int pos;
1259     int del;
1260     if(i>0)
1261     {
1262         printf("Eliminar empleado\n\n");
1263         pos=validNum("Ingrese el numero de emplado que desea eliminar: ",300000,399999);
1264         if(orden)
1265         {
1266             pos=buscarTBin(vect,i,pos);
1267             if(pos!=-1)
1268             {
1269                 if(vect[pos].status)
1270                 {
1271                     printEmpleadoReg(vect[pos]);
1272                     printf("\nDesea eliminar al empleado? (1:Si 2:No)\n");
1273                     del=validNum("Escoge una opcion: ",1,2);
1274                     if(del==1)
1275                     {
1276                         vect[pos].status=0;
1277                         return 1;
1278                     }
1279                 }
1280                 else
1281                 {
1282                     printf("\nEse empleado ya fue eliminado de los registros\n");
1283                 }
1284             }
1285             else
1286             {
1287                 printf("\nEse empleado no existe\n");
1288             }
1289         }
1290         else
1291         {
1292             pos=buscarTSec(vect,i,pos);
1293             if(pos!=-1)
1294             {
1295                 if(vect[pos].status)
1296                 {
1297                     printEmpleadoReg(vect[pos]);
1298                     printf("\nDesea eliminar al empleado? (1:Si 2:No)\n");
1299                     del=validNum("Escoge una opcion: ",1,2);
1300                     if(del==1)
1301                     {
1302                         vect[pos].status=0;
1303                         return 1;
1304                     }
1305                 }
1306                 else
1307                 {
1308                     printf("\nEse empleado ya fue eliminado de los registros\n");
1309                 }
1310             }
1311             else
1312             {
1313                 printf("\nEse empleado no existe\n");
1314             }
1315         }

```

```

1316     }
1317     else
1318     {
1319         printf("\nAun no hay empleados registrados\n");
1320     }
1321     return 0;
1322 }
1323
1324 //case 4
1325 void buscarEmpleado(Tdato vect[],int i,int orden)
1326 {
1327     int pos;
1328     if(i>0)
1329     {
1330         printf("Buscar empleado\n\n");
1331         pos=validNum("Ingrese el numero de empleado que desea buscar: ",300000,399999);
1332         if(orden)
1333         {
1334             pos=buscarTBin(vect,i,pos);
1335             if(pos!=-1)
1336             {
1337                 if(vect[pos].status)
1338                 {
1339                     printEmpleadoReg(vect[pos]);
1340                 }
1341                 else
1342                 {
1343                     printf("\nEse empleado fue eliminado de los registros\n");
1344                 }
1345             }
1346             else
1347             {
1348                 printf("\nEse empleado no existe\n");
1349             }
1350         }
1351         else
1352         {
1353             pos=buscarTSec(vect,i,pos);
1354             if(pos!=-1)
1355             {
1356                 if(vect[pos].status)
1357                 {
1358                     printEmpleadoReg(vect[pos]);
1359                 }
1360                 else
1361                 {
1362                     printf("\nEse empleado fue eliminado de los registros\n");
1363                 }
1364             }
1365             else
1366             {
1367                 printf("\nEse empleado no existe\n");
1368             }
1369         }
1370     }
1371     else
1372     {
1373         printf("\nAun no hay empleados registrados\n");
1374     }
1375 }
1376
1377 void printEmpleadoReg(Tdato empleado)
1378 {
1379     printf("Status: %d\n",empleado.status);
1380     printf("\nNo. Empleado: %d\n",empleado.key);
1381     printf("Puesto: %s\n",empleado.puesto);
1382     printf("Apellido Paterno: %s\n",empleado.apPat);
1383     printf("Apellido Materno: %s\n",empleado.apMat);
1384     printf("Nombre: %s\n",empleado.nombre);
1385     printf("Telefono: %.0f\n",empleado.telefono);
1386     printf("Sexo: ");
1387     if(empleado.sexo=='H')
1388     {
1389         printf("MASCULINO");
1390     }
1391     else
1392     {

```

```

1393     printf("FEMENINO");
1394 }
1395 printf("\n");
1396 }
1397
1398 //case 5
1399 int ordenEmpleado(Tdato vect[],int i,int ordenM)
1400 {
1401     if(i>0)
1402     {
1403         if(ordenM)
1404         {
1405             burbuja(vect,i);
1406         }
1407         else
1408         {
1409             seleccion(vect,i);
1410         }
1411         printf("\nSe ordenaron los empleados\n");
1412         return 1;
1413     }
1414     else
1415     {
1416         printf("\nAun no hay empleado registrados\n");
1417         return 0;
1418     }
1419 }
1420
1421
1422 //case 6
1423 void printEmpleados(Tdato vect[],int i)
1424 {
1425     int j,k=0;
1426     if(i>0)
1427     {
1428         printf("| No |STATUS|No. Empleado|          Puesto          | Apellido Paterno | Apellido Materno | Nombre          | Telefo
1429         for(j=0;j<i;j++)
1430         {
1431             printf("| %4d ",k);
1432             printEmpleadoTab(vect[j]);
1433             k++;
1434         }
1435     }
1436     else
1437     {
1438         printf("\nAun no hay empleado registrados\n");
1439     }
1440 }
1441 }
1442
1443 void printEmpleadoTab(Tdato empleado)
1444 {
1445     printf("| %4d |%11d | %-22s | %-16s | %-17s| %-20s | %.0f |",empleado.status,empleado.key,empleado.puesto,empleado.apPat,empleado.apMa
1446     if(empleado.sexo=='H')
1447     {
1448         printf(" MASCULINO |\n");
1449     }
1450     else
1451     {
1452         printf(" FEMENINO |\n");
1453     }
1454 }
1455
1456 //case 7
1457 void genArchivoTxt(Tdato vect[],int i)
1458 {
1459     FILE *fa;
1460     int j,k;
1461     int len;
1462     char name[L];
1463     printf("Generar archivo\n\n");
1464     if(i>0)
1465     {
1466         printf("Ingresa el nombre del archivo de texto: ");
1467         gets(name);
1468         len=strlen(name);
1469         name[len++]='.';
1470         name[len++]='t'.

```

```

1470     name[len++] = c;
1471     name[len++] = 'x';
1472     name[len++] = 't';
1473     name[len] = '\0';
1474
1475     fa=fopen(name,"r");
1476     if(!fa)
1477     {
1478         fa=fopen(name,"a");
1479         fprintf(fa,"| No   |STATUS| No.Empleado |      Puesto      | Apellido Paterno | Apellido Materno | Nombre
1480         k=0;
1481         for(j=0;j<i;j++)
1482         {
1483             if(vect[j].status)
1484             {
1485                 fprintf(fa,"| %4d ",k);
1486                 fprintf(fa,"| %4d | %11d |%-22s | %-16s | %-17s| %-20s | %.0f |",vect[j].status,vect[j].key,vect[j].puesto,vect[j].api
1487                 if(vect[j].sexo=='H')
1488                 {
1489                     fprintf(fa," MASCULINO |\n");
1490                 }
1491                 else
1492                 {
1493                     fprintf(fa," FEMENINO  |\n");
1494                 }
1495                 k++;
1496             }
1497         }
1498         printf("\nSe genero el archivo: %s\n",name);
1499         fclose(fa);
1500     }
1501     else
1502     {
1503         fclose(fa);
1504         printf("\nYa existe un archivo con ese nombre\n");
1505     }
1506 }
1507 else
1508 {
1509     printf("\nAun no hay empleados registrados\n");
1510 }
1511 }
1512
1513 //case 8
1514 void printArchTxt(void)
1515 {
1516     FILE *fa;
1517     char name[L];
1518     int len;
1519     char c;
1520     printf("Mostrar archivo\n\n");
1521     printf("Ingresa el nombre del archivo de texto: ");
1522     gets(name);
1523     len=strlen(name);
1524     name[len++]='.';
1525     name[len++]='t';
1526     name[len++]='x';
1527     name[len++]='t';
1528     name[len]='\0';
1529
1530     fa=fopen(name,"r");
1531     if(fa)
1532     {
1533         printf("\n %s\n\n",name);
1534         do
1535         {
1536             c=fgetc(fa);
1537             printf("%c",c);
1538         }while(!feof(fa));
1539         fclose(fa);
1540     }
1541     else
1542     {
1543         printf("\nEl nombre no coincide con un archivo existente\n");
1544     }
1545 }
1546
1547 //case 9

```

```

1548 void genArchivoBin(Tdato vect[],int i)
1549 {
1550     FILE *fa;
1551     int j;
1552     if(i>0)
1553     {
1554         fa=fopen("datosbin.dll","rb");
1555         if(fa)
1556         {
1557             fclose(fa);
1558             fa=fopen("datosbin.tmp","rb");
1559             if(fa)
1560             {
1561                 remove("datosbin.tmp");
1562                 rename("datosbin.dll","datosbin.tmp");
1563                 fa=fopen("datosbin.dll","wb");
1564                 for(j=0;j<i;j++)
1565                 {
1566                     fwrite(&vect[j],sizeof(Tdato),1,fa);
1567                 }
1568                 fclose(fa);
1569             }
1570         }
1571         else
1572         {
1573             rename("datosbin.dll","datosbin.tmp");
1574             fa=fopen("datosbin.dll","wb");
1575             for(j=0;j<i;j++)
1576             {
1577                 fwrite(&vect[j],sizeof(Tdato),1,fa);
1578             }
1579             fclose(fa);
1580         }
1581     }
1582     else
1583     {
1584         fa=fopen("datosbin.dll","ab");
1585         for(j=0;j<i;j++)
1586         {
1587             fwrite(&vect[j],sizeof(Tdato),1,fa);
1588         }
1589         fclose(fa);
1590     }
1591     printf("\nSe genero el archivo binario\n");
1592 }
1593 else
1594 {
1595     printf("\nAun no hay empleados registrados\n");
1596 }
1597 }
1598 //case 10
1599 int cargarArchivoBin(Tdato vect[],int i)
1600 {
1601     FILE *fa;
1602     int j;
1603     fa=fopen("datosbin.dll","rb");
1604     if(fa)
1605     {
1606         j=0;
1607         while(!feof(fa))
1608         {
1609             fread(&vect[j],sizeof(vect[j]),1,fa);
1610             j++;
1611         }
1612         printf("\nSe cargo el archivo binario datos.dll a los registros\n");
1613         fclose(fa);
1614         return j;
1615     }
1616     else
1617     {
1618         printf("\nEl archivo binario a cargar no existe\n");
1619     }
1620     return 0;
1621 }
1622 }
1623 //case 11
1624 void printDel(void)
1625 {

```

```

1625 {
1626     FILE *fa;
1627     Tdato reg;
1628     int k;
1629     fa=fopen("datosbin.dll","rb");
1630     if(fa)
1631     {
1632         printf("\nEmpleados eliminados\n\n");
1633         printf("| No   |STATUS| No.Empleado |      Puesto      | Apellido Paterno | Apellido Materno | Nombre      | Tele"
1634         k=0;
1635         while(!feof(fa))
1636         {
1637             fread(&reg,sizeof(Tdato),1,fa);
1638             if(reg.status != 1)
1639             {
1640                 printf("| %4d ",k);
1641                 printEmpleadoTab(reg);
1642                 k++;
1643             }
1644         }
1645         fclose(fa);
1646     }
1647     else
1648     {
1649         printf("\nAun no hay un archivo binario\n");
1650     }
1651 }

```