

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en Computación

Anexos Activdad 14

Programación Estructurada

ALUMNO: Arredondo Urbalejo Isai Alexis

MATRÍCULA: 368747

GRUPO: 932

PROFESOR: Pedro Nunez Yepiz



MATERIA



```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
  .vscode
  Actividad2
  Actividad3
  Actividad4
  Actividad5
  Actividad6
  Actividad7
  Actividad8
  Actividad9
  Actividad9.2
  Actividad9.3
  Actividad10
  Actividad11
  Actividad12
  Actividad13
  Actividad13py
  Actividad14
    alexish
    Ejercicio13.1.cpp
    Actividad15
    alexish
    Ejercicio14.1.cpp
    Ejercicio14.1.exe
    registros.dat
  Ejemplo
  Examen
  Juego
  juego2
  Marcelo
  Mates
  revision
  Software
  OUTLINE
  TIMELINE

Actividad15 > Ejercicio14.1.cpp > ...
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <ctype.h>
5 #include <time.h>
6 #include <windows.h>
7 #include "alexis.h"
8 #define N 2700
9
10 bool ordenado = false;
11
12 typedef struct talum{
13     int matricula;
14     char nombre[20];
15     char apmat[20];
16     char apmat2[20];
17     int edad;
18     char sexo[20];
19     int status;
20 }talum;
21
22 typedef struct tindex{
23     long matricula;
24     int indice;
25 }tindex;
26
27 int msges();
28 void menu();
29 talum agregar(talum vect[],int n);
30 void prntbin(talum vect[],int n);
31 void buscar(talum vect[], int n);
32 void cargarind(tindex vect[],int n);
33 void eliminar(tindex vect[], int n);
34 int leebinario(talum vect[],int n);
35 void archtext(talum vect[],tindex vect_index[],int n,char nombreachivo[]);
36 void generabin(void);
37 void buscar_principal(talum vect[],int n);
38 void imprimirtxt(char nombreachivo[]);
39 int validar_matri(int matri, talum vector[], int n);
```

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
  .vscode
  Actividad2
  Actividad3
  Actividad4
  Actividad5
  Actividad6
  Actividad7
  Actividad8
  Actividad9
  Actividad9.2
  Actividad9.3
  Actividad10
  Actividad11
  Actividad12
  Actividad13
  Actividad13py
  Actividad14
    alexish
    Ejercicio13.1.cpp
    Actividad15
    alexish
    Ejercicio14.1.cpp
    Ejercicio14.1.exe
    registros.dat
  Ejemplo
  Examen
  Juego
  juego2
  Marcelo
  Mates
  revision
  Software
  OUTLINE
  TIMELINE

Actividad15 > Ejercicio14.1.cpp > ...
40 void ordenar(tindex vect[], int n);
41 void burbuja(tindex vect[],int n);
42 void buscar(tindex vect[], int n);
43 void quicksort(tindex vect[], int primero, int ultimo);
44 void imprimir_ord(tindex vect[], int n);
45 void empaquetar(int n);
46
47 int main()
48 {
49     srand(time(NULL));
50     menu();
51     return 0;
52 }
53
54 int msges()
55 {
56     int op,menu;
57     printf(" M E N U\n");
58     printf("1.- Agregar\n");
59     printf("2.- Eliminar\n");
60     printf("3.- Buscar\n");
61     printf("4.- Ordenar\n");
62     printf("5.- Imprimir registros archivo original\n");
63     printf("6.- Imprimir registros archivo ordenado\n");
64     printf("7.- Generar archivo de texto\n");
65     printf("8.- Empaquetar\n");
66     printf("9.- Salir\n");
67     op=validum(0,8,"Elige una opcion:");
68     return op;
69 }
70
71 void menu()
72 {
73     int i,op,total,tam=0,j,k=0,l=0;
74     char nom_arch[20] = "Alumnos";
75     char nom_arch_bin[20] = "arch1";
76     char nombreachivo[20];
77     generabin();
78     bool cargado_bin = false;
```



MATERIA

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
Actividad15 > Ejercicio14.1.cpp > ...
79 talun vect[N];
80 tindex vect_index[N];
81 cargarInd(vect_index,0);
82 k = leebinario(vect,k);
83 do{
84     op=meses();
85     switch (op)
86     {
87     case 1:
88         if(k < N-1)
89         {
90             vect[i]=agregar(vect,k);
91             k++;
92             ordenado = false;
93             cargarInd(vect_index,0);
94         }
95         break;
96     case 2:
97         eliminar(vect_index,k);
98         break;
99     case 3:
100         buscar(vect_index,k);
101         break;
102     case 4:
103         if(ordenado==false)
104         {
105             ordenar(vect_index,k);
106             ordenado = true;
107         }
108         else
109         {
110             printf("El vector ya esta ordenado \n");
111             system("PAUSE");
112         }
113         break;
114     case 5:
115         printbin(vect,k);
116         break;
117     case 6:
```

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
Actividad15 > Ejercicio14.1.cpp > ...
118 imprimir_ord(vect_index,k);
119 break;
120 case 7:
121     printf("Ingrese el nombre para guardar su archivo de texto: ");
122     fflush(stdin);
123     gets(nombreadarchivo);
124     archtext(vect,vect_index,k,nombreadarchivo);
125     break;
126 case 8:
127     empaquetar(k);
128     break;
129 }
130 while (op != 0);
131 system("PAUSE");
132 }
133
134 talun agregar(talun vect[],int n)
135 {
136     talun reg;
137     char nombres_hom[20][20]={"Raymundo", "Jacob", "Jose", "Abraham", "Daniel", "Alexis", "Mercolo", "Roberto", "Ellen", "Francisco"};
138     char nombres_muj[20][20]={"Nancy", "Smartha", "Thanya", "Brianda", "Zharina", "Patricia", "Claudia", "Alexa", "Cosete", "Vanessa"};
139     char apmat[20][20]={"Acosta", "Callejas", "Arista", "Perez", "Guertero", "Evans", "Loyo", "Urbalejo", "Peralta", "Garcia"};
140     char apmat[20][20]={"Hernandez", "Ortiz", "Carrera", "Paniagua", "Beristain", "Marquez", "Quintero", "Cobiza", "Siqueiros", "Navarro"};
141     int sexo,status=1,i;
142     long ri3=300000, rf3=3999999, rango3,matri;
143     strcpy(reg.apmat,apmat[rand()%(sizeof(apmat)/sizeof(char*))]);
144     strcpy(reg.apmat,apmat[rand()%(sizeof(apmat)/sizeof(char*))]);
145     sexo=(rand()%(sizeof(sexo)+1));
146     if (sexo==1)
147     {
148         strcpy(reg.sexo,"HOMBRE");
149         strcpy(reg.nombre,nombres_hom[rand()%(sizeof(nombres_hom)/sizeof(char*))]);
150     }
151     else
152     {
153         strcpy(reg.sexo,"MUJER");
154         strcpy(reg.nombre,nombres_muj[rand()%(sizeof(nombres_muj)/sizeof(char*))]);
155     }
156     reg.edad=(rand()%(sizeof(reg.edad)+1));
```



MATERIA

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
Actividad15
Actividad14.1.cpp
Actividad13.1.cpp
Actividad15
Ejercicio14.1.exe
registros.dat
Ejemplo
Examen
Juego
Juego2
Marcelo
Mates
revision
Software
OUTLINE
TIMELINE

Actividad15 > Ejercicio14.1.cpp > ...
157 rango3 = ( rf3 - ri3 ) + 1;
158 do{
159     matri = (rand()%rango3) + ri3;
160     while(validar_matri(matri,vect,n) != -1);
161     reg.matricula = matri;
162     reg.status=status;
163     FILE *arch;
164     arch = fopen("registros.dat","a+b");
165     if(arch)
166     {
167         fwrite(&reg,sizeof(talum),1,arch);
168         fclose(arch);
169     }
170     printf("Agregado correctamente \n");
171     return reg;
172 }
173
174 int validar_matri(int matri, talum vect[], int n)
175 {
176     for (int i = 0; i < n; i++)
177     {
178         if (vect[i].matricula == matri)
179         {
180             return i;
181         }
182     }
183     return -1;
184 }
185
186 void prntbin(talum vect[],int n)
187 {
188     system("CLS");
189     int i;
190     talum reg;
191     FILE *arch;
192     arch = fopen("registros.dat","r+b");
193     if(arch)
194     {
195         while(fread(&reg,sizeof(talum),1,arch))
196         {
197             printf(" %6d %10s %12s %12s %6d %10s %d \n",reg.matricula,reg.nombre,reg.apmat,reg.apmat,reg.edad,reg.sexo,reg.status);
198         }
199         fclose(arch);
200     }
201     system("PAUSE");
202 }
```

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
Actividad15
Actividad14.1.cpp
Actividad13.1.cpp
Actividad15
Ejercicio14.1.exe
registros.dat
Ejemplo
Examen
Juego
Juego2
Marcelo
Mates
revision
Software
OUTLINE
TIMELINE

Actividad15 > Ejercicio14.1.cpp > ...
196 i = n;
197 while(fread(&reg,sizeof(talum),1,arch))
198 {
199     printf(" %6d %10s %12s %12s %6d %10s %d \n",reg.matricula,reg.nombre,reg.apmat,reg.apmat,reg.edad,reg.sexo,reg.status);
200 }
201 fclose(arch);
202 system("PAUSE");
203 }
204
205 void cargarind(tindex vect[],int n)
206 {
207     int i, index;
208     long matricula;
209     talum temp;
210     tindex reg;
211     FILE *arch;
212     arch = fopen("registros.dat","r+b");
213     if(arch)
214     {
215         i = n;
216         while(fread(&temp,sizeof(talum),1,arch))
217         {
218             reg.matricula = temp.matricula;
219             reg.indice = i;
220             vect[i] = reg;
221             i++;
222         }
223         fclose(arch);
224     }
225 }
226
227 void eliminar(tindex vect[], int n)
228 {
229     system("CLS");
230     int mati,i,j,com=0,op;
231     int indicebuscado;
232     long desplazamiento;
233     mati=validar(300000,399999,"Ingrese la Matricula que desea eliminar:");
234     while(mati != -1)
235     {
236         for (j = 0; j < n; j++)
237         {
238             if (vect[j].matricula == mati)
239             {
240                 com = j;
241                 break;
242             }
243         }
244         if (com != 0)
245         {
246             for (i = com; i < n; i++)
247             {
248                 vect[i] = vect[i+1];
249             }
250             vect[n] = vect[com];
251             n--;
252         }
253         mati = validar(300000,399999,"Ingrese la Matricula que desea eliminar:");
254     }
255 }
```



MATERIA

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
Actividad15 > Ejercicio14.1.cpp > ...
235 for(i=0; i<n; i++)
236 {
237     if (vect[i].matricula==matl)
238     {
239         printf("Matricula %ld encontrada en el indice %d \n",vect[i].matricula,vect[i].indice);
240         com=1;
241         FILE *arch;
242         arch = fopen("registros.dat","r+b");
243         if(arch)
244         {
245             indicebuscado = vect[i].indice;
246             desplazamiento = (long)(sizeof(talun)) * (long)(indicebuscado);
247             fseek(arch, desplazamiento, SEEK_SET);
248             talun reg;
249             fread(&reg, sizeof(talun), 1, arch);
250             printf("\n");
251             printf(" MATRICULA NOMBRE APELLIDO PAT APELLIDO MAT EDAD SEXO \n");
252             printf(" %6ld %8s %14s %14s %5d %7s \n",reg.matricula,reg.nombre,reg.apmat,reg.apmat,reg.edad,reg.);
253             com=1;
254             printf("Seguro que desea eliminar la matricula? \n");
255             op=validum(1,2,"1.-Si \n 2.-No \n");
256             if(op == 1)
257             {
258                 reg.status = 0;
259                 fseek(arch, desplazamiento, SEEK_SET);
260                 fwrite(&reg, sizeof(talun), 1, arch);
261                 printf("\n Matricula eliminada correctamente \n");
262             }
263             fseek(arch, desplazamiento, SEEK_END);
264             fclose(arch);
265         }
266     }
267     if (com==0)
268     {
269         printf("..\n");
270     }
271     system("PAUSE");
272 }
273
```

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
Actividad15 > Ejercicio14.1.cpp > ...
274 }
275 void buscar(tindex vect[], int n)
276 {
277     system("CLS");
278     int mati,i,j,com=0;
279     int indicebuscado;
280     long desplazamiento;
281     mati=validum(300000,399999,"Ingrese la Matricula que desea buscar: \n");
282     for(i=0; i<n; i++)
283     {
284         if (vect[i].matricula==mati)
285         {
286             printf("Matricula %ld encontrada en el indice %d \n",vect[i].matricula,vect[i].indice);
287             com=1;
288             FILE *arch;
289             arch = fopen("registros.dat","r+b");
290             if(arch)
291             {
292                 indicebuscado = vect[i].indice;
293                 desplazamiento = (long)(sizeof(talun)) * (long)(indicebuscado);
294                 fseek(arch, desplazamiento, SEEK_SET);
295                 talun reg;
296                 fread(&reg, sizeof(talun), 1, arch);
297                 printf("\n");
298                 printf(" MATRICULA NOMBRE APELLIDO PAT APELLIDO MAT EDAD SEXO STATUS\n");
299                 printf(" %6ld %8s %14s %14s %5d %7s [%d] \n",reg.matricula,reg.nombre,reg.apmat,reg.apmat,reg.edad,reg.);
300                 com=1;
301                 fseek(arch, desplazamiento, SEEK_END);
302                 fclose(arch);
303             }
304         }
305     }
306     if (com==0)
307     {
308         printf("Matricula no encontrada \n");
309     }
310     system("PAUSE");
311 }
312
```



MATERIA

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
> .vscode
> Actividad2
> Actividad3
> Actividad4
> Actividad5
> Actividad6
> Actividad7
> Actividad8
> Actividad9
> Actividad9.2
> Actividad9.3
> Actividad10
> Actividad11
> Actividad12
> Actividad13
> Actividad13.py
> Actividad14
> Ejercicio14.1.cpp
> Ejercicio13.1.cpp
> Ejercicio14.1.exe
> Ejercicio14.1.cpp
> Ejercicio14.1.exe
> registros.dat
> Ejemplo
> Examen
> Juego
> Juego2
> Marcelo
> Mates
> revision
> Software
> OUTLINE
> TIMELINE
Actividad15 > Ejercicio14.1.cpp > ...
312 system("PAUSE");
313 }
314
315 int leebinario(talum vect[],int n)
316 {
317     int i;
318     talum reg;
319     FILE *arch;
320     arch = fopen("registros.dat","r+b");
321     if(arch)
322     {
323         i = n;
324         while(fread(&reg,sizeof(talum),1,arch))
325         {
326             vect[i] = reg;
327             i++;
328             if(i == n)
329             {
330                 return i;
331             }
332         }
333         fclose(arch);
334     }
335     return i;
336 }
337
338 void ordenar(tindex vect[], int n)
339 {
340     int i, j;
341     tindex burb;
342     if (vect[i].matricula > vect[i+1].matricula)
343     {
344         quicksort(vect,0,n);
345     }
346     else
347     {
348         burbuja(vect,n);
349     }
350     printf("Vector ordenado correctamente \n");

```

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
> .vscode
> Actividad2
> Actividad3
> Actividad4
> Actividad5
> Actividad6
> Actividad7
> Actividad8
> Actividad9
> Actividad9.2
> Actividad9.3
> Actividad10
> Actividad11
> Actividad12
> Actividad13
> Actividad13.py
> Actividad14
> Ejercicio14.1.cpp
> Ejercicio13.1.cpp
> Ejercicio14.1.exe
> Ejercicio14.1.cpp
> Ejercicio14.1.exe
> registros.dat
> Ejemplo
> Examen
> Juego
> Juego2
> Marcelo
> Mates
> revision
> Software
> OUTLINE
> TIMELINE
Actividad15 > Ejercicio14.1.cpp > ...
351 system("PAUSE");
352 }
353
354 void burbuja(tindex vect[], int n)
355 {
356     int i, j;
357     tindex burb;
358     for (i = 0; i < n; i++)
359     {
360         for (j = i+1; j < n; j++)
361         {
362             if (vect[i].matricula > vect[j].matricula)
363             {
364                 burb = vect[i];
365                 vect[i] = vect[j];
366                 vect[j] = burb;
367             }
368         }
369     }
370 }
371
372 void quicksort(tindex vect[], int primero, int ultimo)
373 {
374     int i, j, pivote_n;
375     tindex pivote,temp;
376     if (primero < ultimo)
377     {
378         pivote = vect[ultimo];
379         i = primero - 1;
380
381         for (j = primero; j <= ultimo-1; j++)
382         {
383             if (vect[j].matricula < pivote.matricula)
384             {
385                 i++;
386                 temp = vect[i];
387                 vect[i] = vect[j];
388                 vect[j] = temp;
389             }

```



MATERIA

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
Actividad15 > Ejercicio14.1.cpp > ...
390 }
391 temp = vect[i+1];
392 vect[i+1] = vect[ultimo];
393 vect[ultimo] = temp;
394 pivote_n = i + 1;
395 quicksort(vect, primero, pivote_n - 1);
396 quicksort(vect, pivote_n + 1, ultimo);
397 }
398 }
399
400 void imprimir_ord(tindex vect[], int n)
401 {
402     system("CLS");
403     int mati,i,j;
404     long desplazamiento;
405     int indicebuscado;
406     FILE *arch;
407     arch = fopen("registros.dat","rb");
408     if(arch)
409     {
410         printf(" MATRICULA NOMBRE APELLIDO PAT APELLIDO MAT EDAD SEXO STATUS\n");
411         printf("\n");
412         for(i=0;i<n;i++)
413         {
414             indicebuscado = vect[i].indice;
415             desplazamiento = (long)(sizeof(talum)) * (long)(indicebuscado);
416             fseek(arch, desplazamiento, SEEK_SET);
417             talum reg;
418             fread(&reg, sizeof(talum), 1, arch);
419             printf("%6d %8s %14s %14s %5d %7s [%d] \n",reg.matricula,reg.nombre,reg.apmat,reg.apmat,reg.edad,reg.
420             );
421             fseek(arch, desplazamiento, SEEK_END);
422         }
423         fclose(arch);
424         system("PAUSE");
425     }
426 }
427 void archtext(talum vect[],tindex vect_index[],int n,char nombreadchivo[])
428 {
```

```
File Edit Selection View Go Run ... Programacion
EXPLORER
PROGRAMACION
Actividad15 > Ejercicio14.1.cpp > ...
429 int i,op;
430 long desplazamiento;
431 int indicebuscado;
432 char nombre_arch[24];
433 strcpy(nombre_arch,nombreadchivo);
434 strcat(nombre_arch,".txt");
435 printf("Desea imprimir el archivo ordenado? \n");
436 op=validum(1,2,"1.-Si \n 2.- No \n");
437 FILE *arch;
438 arch = fopen(nombre_arch,"w");
439 FILE *arch_bin;
440 arch_bin = fopen("registros.dat","rb");
441 if(arch)
442 {
443     talum reg;
444     fprintf(arch," No. MATRICULA NOMBRE APELLIDO P. APELLIDO MAT. EDAD SEXO \n");
445     if(op == 1)
446     {
447         for(i=0;i<n;i++)
448         {
449             indicebuscado = vect_index[i].indice;
450             desplazamiento = (long)(sizeof(talum)) * (long)(indicebuscado);
451             fseek(arch_bin, desplazamiento, SEEK_SET);
452             fread(&reg, sizeof(talum), 1, arch_bin);
453             fprintf(arch,"%6d %8s %10s %12s %12s %6d %10s \n",i,reg.matricula,reg.nombre,reg.apmat,reg.apmat,reg.
454             );
455         }
456     }
457     else
458     {
459         i = 0;
460         while(fread(&reg,sizeof(talum),1,arch_bin))
461         {
462             fprintf(arch,"%6d %8s %10s %12s %12s %6d %10s \n",i,reg.matricula,reg.nombre,reg.apmat,reg.apmat,reg.
463             );
464             i++;
465         }
466         fclose(arch);
467     }
468     fseek(arch_bin, desplazamiento, SEEK_END);
469     fclose(arch_bin);
470 }
```



MATERIA

```
File Edit Selection View Go Run ... Programacion
EXPLORER
  PROGRAMACION
    .vscode
    > Actividad2
    > Actividad3
    > Actividad4
    > Actividad5
    > Actividad6
    > Actividad7
    > Actividad8
    > Actividad9
    > Actividad9.2
    > Actividad9.3
    > Actividad10
    > Actividad11
    > Actividad12
    > Actividad13
    > Actividad13.py
    > Actividad14
    > alexish
    > Ejercicio13.1.cpp
    > Actividad15
    > alexish
    > Ejercicio14.1.cpp
    > Ejercicio14.1.exe
    > registros.dat
    > Ejemplo
    > Examen
    > Juego
    > Juego2
    > Marcelo
    > Mates
    > revision
    > Software
    > OUTLINE
    > TIMELINE
  Ejercicio14.1.cpp
  Ejercicio14.1.exe
  registros.dat
  Ejemplo
  Examen
  Juego
  Juego2
  Marcelo
  Mates
  revision
  Software
  OUTLINE
  TIMELINE

468 }
469 }
470
471 void generarbin(void)
472 {
473     int i;
474     talum reg;
475     FILE *arch;
476     arch = fopen("registros1.dat", "r+b");
477     FILE *copia_arch;
478     copia_arch = fopen("registros.dat", "w+b");
479     if(arch)
480     {
481         while(fread(&reg, sizeof(talum), 1, arch))
482         {
483             fwrite(&reg, sizeof(talum), 1, copia_arch);
484         }
485         fclose(arch);
486         fclose(copia_arch);
487     }
488 }
489
490 void empaquetar(int n)
491 {
492     int i;
493     talum reg;
494     FILE *arch;
495     arch = fopen("registros.dat1", "wb");
496     FILE *copia_arch;
497     copia_arch = fopen("registros.dat", "r+b");
498     if(arch)
499     {
500         for(i=0; i<n; i++)
501         {
502             fread(&reg, sizeof(talum), 1, copia_arch);
503             fwrite(&reg, sizeof(talum), 1, arch);
504         }
505         fclose(arch);
506         fclose(copia_arch);
507     }
508 }
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Alexis\OneDrive\Documentos\ESTRUCTURADA 2023 1\Programacion\Actividad15> .\Ejercicio14.1
M E N U
1.- Agregar
2.- Eliminar
3.- Buscar
4.- Ordenar
5.- Imprimir registros archivo original
6.- Imprimir registros archivo ordenado
7.- Generar archivo de texto
8.- Empaquetar
0.- Salir
Elige una opcion: 1
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
304234 Samantha Peralta Hernandez 16 HOMBRE 1
324553 Brianda Siqueiros 16 MUJER 1
308641 Josue Hernandez 19 MUJER 1
332749 Patricia Acosta Beristain 21 HOMBRE 1
302626 Patricia Urbalejo Ortiz 21 MUJER 1
Presione una tecla para continuar . . .
```