

Universidad Autónoma de Baja California  
Facultad de Ingeniería, Arquitectura y Diseño



Programación Estructurada  
Prof. Pedro Núñez Yepiz  
Aarón Alejandro Parra Velarde  
Actividad 13 - Anexos

Grupo: 432  
Matrícula: 372193  
Ensenada, B.C., 21 de Noviembre de 2023

```

1  //////////////////////////////////////////////////
2  /* Pedro Vicente Aaron Alejandro 27/192 */
3  /* 17nov2023 */
4  /* Programa que despliega un menu de estructuras y archivos. */
5  /* Realizado por */
6  //////////////////////////////////////////////////
7
8  #include <string.h>
9  #include <stdio.h>
10 #include <ctype.h>
11 #define N 1000
12
13 typedef struct _alumno
14 {
15     int status;
16     int matric;
17     char nombre[50];
18     char vec[50];
19     char mat[50];
20     char muest[50];
21     char vec[50];
22 }alumno;
23
24 void menu();
25 void mng();
26 void eliminar(char cadena[]);
27 void nom_mujer(char cadena[]);
28 void nom_mujer(char cadena[]);
29 void insertar(alumno vect[], int n);
30 void buscar(alumno vect[], int n);
31 void buscar(alumno vect[], int n, int str);
32 void ordenar(alumno vect[], int n);
33 void agregar(alumno vect[], int n, char name[]);
34 void creararchivo(alumno vect[], int n, char name[]);
35 alumno gen_reg_aut();
36 alumno eliminar(alumno vect[], int n);
37 alumno modifi(alumno vect[], int n);
38 void imprimir(alumno vect[], int n);
39 void menu(char cadena[]);
40 void readutf8(char name[]);
41 void readutf8(alumno vect[], int n);
42 int readbin(alumno vect[], int n);
43
44 int main()
45 {
46     menu();
47     return 0;
48 }
49
50 void mng()
51 {
52     printf("\n1.- Agregar 100 registros (auto.)");
53     printf("\n2.- Editar Registro");
54     printf("\n3.- Eliminar un Registro");
55     printf("\n4.- Borrar Registro");
56     printf("\n5.- Ordenar Registro");
57     printf("\n6.- Imprimir Registro");
58     printf("\n7.- Generar Archivo");
59     printf("\n8.- Ver Archivo de Texto");
60     printf("\n9.- Crear Archivo Binario");
61     printf("\n10.- Cargar Archivo Binario");
62     printf("\n11.- Borrar Alumnos");
63     printf("\n0.- Salir");
64     printf("\n");
65 }
66
67 void menu()
68 {
69     system("CLS");
70     alumno vect[N];
71     char name[50];
72     int opc, i = 0, j, k = 0, u = 0;
73     while(1)
74     {
75         system("CLS");
76         mng();
77         opc = valid_num(0,11,"Elija una opcion");
78         switch(opc)
79         {
80             case 1:
81                 if (i < N-100)
82                 {
83                     for(j = 0; j < 100; j++)
84                     {
85                         vect[i] = gen_reg_aut();
86                         i++;
87                     }
88                     printf("\nRegistro generado.");
89                     printf("\n");
90                     system("pause");
91                 }
92                 break;
93             case 2:
94                 vect[N] = modifi(vect,i);
95                 break;
96             case 3:
97                 vect[N] = eliminar(vect,i);
98                 break;
99             case 4:
100                 borrar(vect,i);
101                 break;
102             case 5:
103                 ordenar(vect,i);
104                 break;
105             case 6:
106                 imprimir(vect,i);
107                 break;
108             case 7:
109                 creararchivo(vect,i,name);
110                 if (i > 0)
111                 {
112                     agregar(vect,i,name);
113                     k++;
114                 }
115                 break;
116             case 8:
117                 if (k != 0)
118                 {
119                     readutf8(name);
120                 }
121                 else
122                 {
123                     printf("\nEl archivo esta vacio.");
124                 }
125                 break;
126             case 9:
127                 readbin(vect,i);
128                 break;
129             case 10:
130                 if (u != 0)
131                 {
132                     i = readbin(vect,i);
133                     u = i;
134                 }
135                 else
136                 {
137                     printf("\nEl archivo solo se puede cargar una vez");
138                     printf("\n");
139                     system("pause");
140                 }
141                 break;
142             case 11:
143                 if (i != 0)
144                 {
145                     imprimir(vect,i);
146                 }
147                 else
148                 {
149                     printf("\nEl vector debe de tener registros.");
150                 }
151                 break;
152             }
153         }
154     }
155 }

```







```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  ...  VSCodeProgr
EXPLORADOR  ACT13_PE > AAPV_ACT13_01.cpp X  CodeSnap
EDITORES ABIERTOS
VSODEPROGR
ESQUEMA
LÍNEA DE TIEMPO

381
382
383
384     printf("\nRegistros Eliminados");
385     printf("\n\tMo.  Matricula  Nombre  A. Paterno  A.Materno  Puesto  Sexo ");
386     for(r = 0; r < u; r++)
387     {
388         printf("\n\t%2d %10d %17s %12s %13s %15s %10s",r+1,ve[r].matric,ve[r].nombre,ve[r].apat,
389             ve[r].amat,ve[r].puesto,ve[r].sexo);
390     }
391
392     else
393     {
394         printf("\nNo hay registros eliminados.");
395     }
396     printf("\n");
397     system("PAUSE");
398 }
399
400 Talum modif(Talum vect[],int n)
401 {
402     int op,h,x,rep,mtr;
403     char modif[30];
404     char sx2[2][8] = {"MUJER","HOMBRE"};
405     mtr = valid_num(1,10000,"Ingresa una matricula: ");
406     x = busreg(vect,n,mtr);
407     if (x != -1)
408     {
409         printf("\nSe ha encontrado el registro:");
410         printf("\n\tMatricula  Nombre  A. Paterno  A.Materno  Puesto  Sexo");
411         printf("\n\t%2d %17s %12s %13s %15s %10s",vect[x].matric, vect[x].nombre, vect[x].apat, vect[x].amat, vect[x].puest
412             printf("\n");
413     }
414 }
415
416 else
417 {
418     printf("\nLa matricula no existe en el registro.");
419 }
420
421 if (x != -1)
422 {
423     do
424     {
425         printf("1-Matricula  ");
426         printf("2-Nombre  ");
427         printf("3-Apellido Paterno  ");
428         printf("4-Apellido Materno  ");
429         printf("5-Puesto  ");
430         printf("6-Sexo  ");
431         printf("0-Salir  ");
432     }
433 }
```

```
Archivo  Editar  Selección  Ver  Ir  Ejecutar  ...  VSCodeProgr
EXPLORADOR  ACT13_PE > AAPV_ACT13_01.cpp X  CodeSnap
EDITORES ABIERTOS
VSODEPROGR
ESQUEMA
LÍNEA DE TIEMPO

381
382
383
384     printf("\nRegistros Eliminados");
385     printf("\n\tMo.  Matricula  Nombre  A. Paterno  A.Materno  Puesto  Sexo ");
386     for(r = 0; r < u; r++)
387     {
388         printf("\n\t%2d %10d %17s %12s %13s %15s %10s",r+1,ve[r].matric,ve[r].nombre,ve[r].apat,
389             ve[r].amat,ve[r].puesto,ve[r].sexo);
390     }
391
392     else
393     {
394         printf("\nNo hay registros eliminados.");
395     }
396     printf("\n");
397     system("PAUSE");
398 }
399
400 Talum modif(Talum vect[],int n)
401 {
402     int op,h,x,rep,mtr;
403     char modif[30];
404     char sx2[2][8] = {"MUJER","HOMBRE"};
405     mtr = valid_num(1,10000,"Ingresa una matricula: ");
406     x = busreg(vect,n,mtr);
407     if (x != -1)
408     {
409         printf("\nSe ha encontrado el registro:");
410         printf("\n\tMatricula  Nombre  A. Paterno  A.Materno  Puesto  Sexo");
411         printf("\n\t%2d %17s %12s %13s %15s %10s",vect[x].matric, vect[x].nombre, vect[x].apat, vect[x].amat, vect[x].puest
412             printf("\n");
413         system("PAUSE");
414     }
415
416     else
417     {
418         printf("\nLa matricula no existe en el registro.");
419     }
420
421     if (x != -1)
422     {
423         do
424         {
425             printf("1-Matricula  ");
426             printf("2-Nombre  ");
427             printf("3-Apellido Paterno  ");
428             printf("4-Apellido Materno  ");
429             printf("5-Puesto  ");
430             printf("6-Sexo  ");
431             printf("0-Salir  ");
432         }
433     }
```





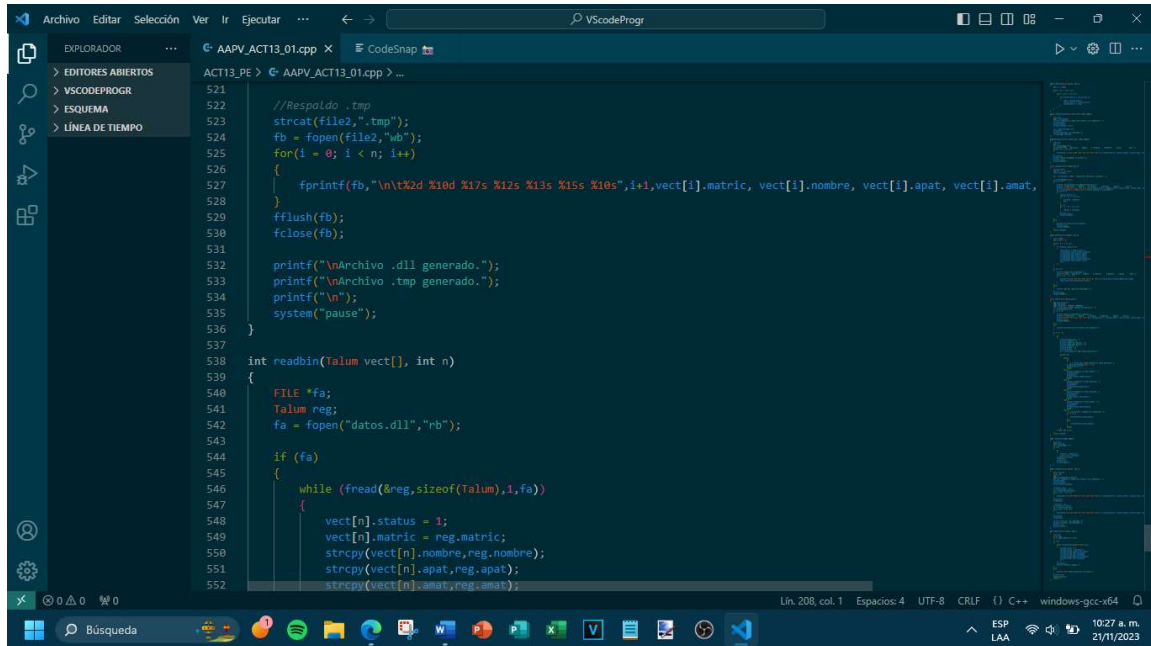
This screenshot shows the Visual Studio Code editor with the file `AAPV_ACT13_01.cpp` open. The editor is displaying the `readtxtfile` function, which reads a text file and prints its contents. The function signature is `void readtxtfile(char name[])`. The code includes file opening, character reading, and printing. The status bar at the bottom indicates the file is in UTF-8 encoding with 4 spaces and CRLF line endings.

```
469     {
470         strcpy(vect[x].sexo, sx2[1]);
471     }
472     else
473     {
474         strcpy(vect[x].sexo, sx2[0]);
475     }
476     break;
477 }
478 } while (op != 0);
479 }
480 return vect[N];
481 }
482
483 void readtxtfile(char name[])
484 {
485     FILE *fa;
486     char caracter;
487     fa = fopen(name, "r");
488     if (fa)
489     {
490         do
491         {
492             caracter = fgetc(fa);
493             printf("%c", caracter);
494             while (!feof(fa));
495             fclose(fa);
496             printf("\n");
497             system("pause");
498         }
499     }
500 }
```

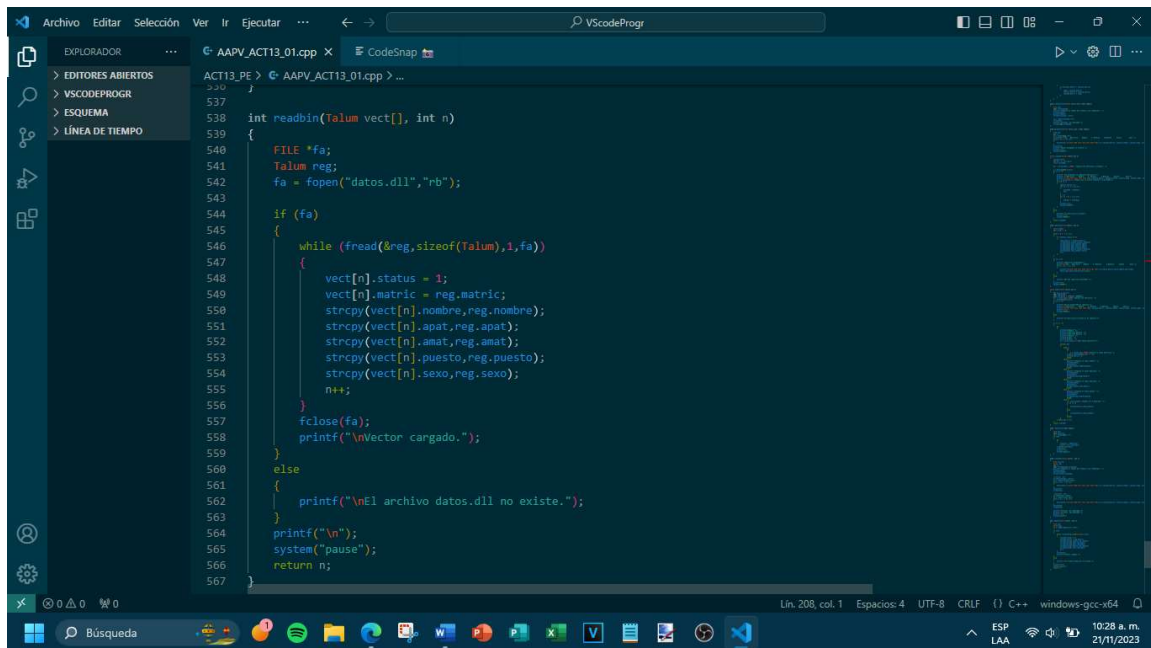
This screenshot shows the Visual Studio Code editor with the file `AAPV_ACT13_01.cpp` open. The editor is displaying the `createbin` function, which creates a binary file from a vector of data. The function signature is `void createbin(Talun vect[], int n)`. The code includes file opening, data writing, and file closing. The status bar at the bottom indicates the file is in UTF-8 encoding with 4 spaces and CRLF line endings.

```
500
501 void createbin(Talun vect[], int n)
502 {
503     FILE *fa, *fb;
504     Talun reg;
505     int i;
506     char filename[40], file2[40];
507     printf("\nIngrese el nombre del archivo (sin extension): ");
508     fflush(stdin);
509     gets(filename);
510     strcpy(file2, filename);
511
512     // Archivo .dll
513     strcat(filename, ".dll");
514     fa = fopen(filename, "wb");
515     for(i = 0; i < n; i++)
516     {
517         fprintf(fa, "\n\t%2d\t%10d\t%17s\t%12s\t%13s\t%15s\t%10s", i+1, vect[i].matric, vect[i].nombre, vect[i].apat, vect[i].amat,
518     }
519     fflush(fa);
520     fclose(fa);
521
522     // Respaldo .tmp
523     strcat(file2, ".tmp");
524     fb = fopen(file2, "wb");
525     for(i = 0; i < n; i++)
526     {
527         fprintf(fb, "\n\t%2d\t%10d\t%17s\t%12s\t%13s\t%15s\t%10s", i+1, vect[i].matric, vect[i].nombre, vect[i].apat, vect[i].amat,
528     }
529     fflush(fb);
530     fclose(fb);
531 }
```





```
521 //Respaldo .tmp
522 strcat(file2, ".tmp");
523 fb = fopen(file2, "wb");
524 for(i = 0; i < n; i++)
525 {
526     fprintf(fb, "\n\t%2d %10d %17s %12s %13s %15s %10s", i+1, vect[i].matric, vect[i].nombre, vect[i].apat, vect[i].amat,
527 }
528 fflush(fb);
529 fclose(fb);
530
531 printf("\nArchivo .dll generado.");
532 printf("\nArchivo .tmp generado.");
533 printf("\n");
534 system("pause");
535 }
536
537 int readbin(Talun vect[], int n)
538 {
539     FILE *fa;
540     Talun reg;
541     fa = fopen("datos.dll", "rb");
542
543     if (fa)
544     {
545         while (fread(&reg, sizeof(Talun), 1, fa))
546         {
547             vect[n].status = 1;
548             vect[n].matric = reg.matric;
549             strcpy(vect[n].nombre, reg.nombre);
550             strcpy(vect[n].apat, reg.apat);
551             strcpy(vect[n].amat, reg.amat);
552             strcpy(vect[n].amat, reg.amat);
553         }
554     }
555 }
```



```
537 }
538
539 int readbin(Talun vect[], int n)
540 {
541     FILE *fa;
542     Talun reg;
543     fa = fopen("datos.dll", "rb");
544
545     if (fa)
546     {
547         while (fread(&reg, sizeof(Talun), 1, fa))
548         {
549             vect[n].status = 1;
550             vect[n].matric = reg.matric;
551             strcpy(vect[n].nombre, reg.nombre);
552             strcpy(vect[n].apat, reg.apat);
553             strcpy(vect[n].amat, reg.amat);
554             strcpy(vect[n].puesto, reg.puesto);
555             strcpy(vect[n].sexo, reg.sexo);
556             n++;
557         }
558         fclose(fa);
559         printf("\nVector cargado.");
560     }
561     else
562     {
563         printf("\nEl archivo datos.dll no existe.");
564     }
565     printf("\n");
566     system("pause");
567     return n;
568 }
```