

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

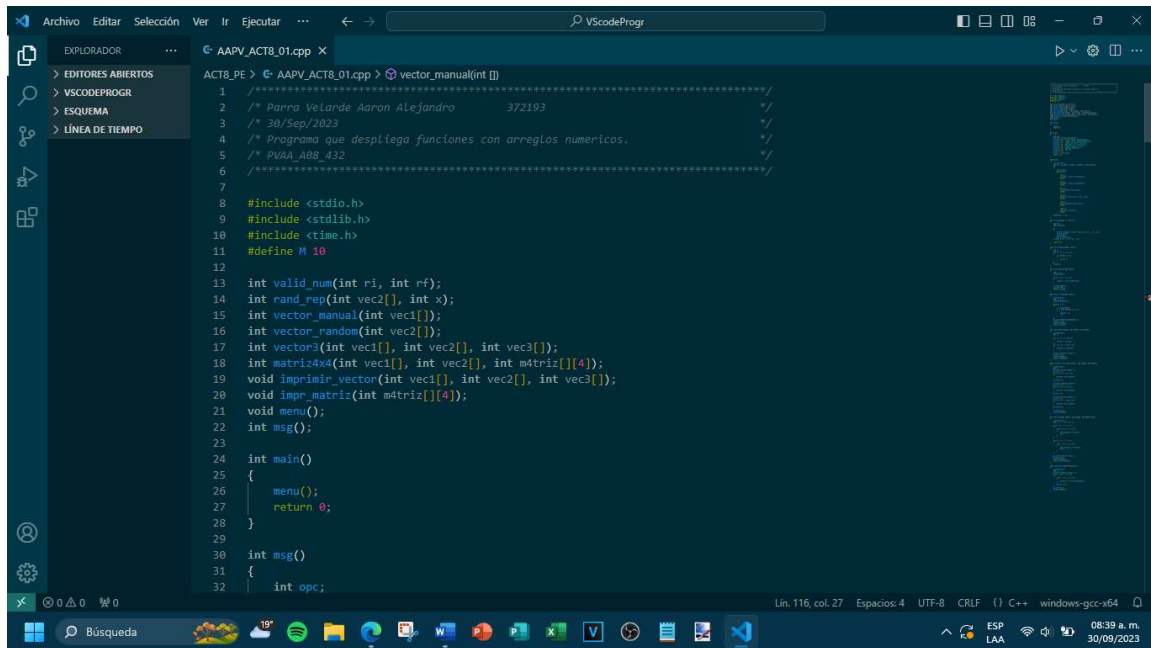


Programación Estructurada
Prof. Pedro Núñez Yepiz
Aarón Alejandro Parra Velarde
Actividad 8 - Anexos

Grupo: 432

Matrícula: 372193

Ensenada, B.C., 30 de Septiembre de 2023

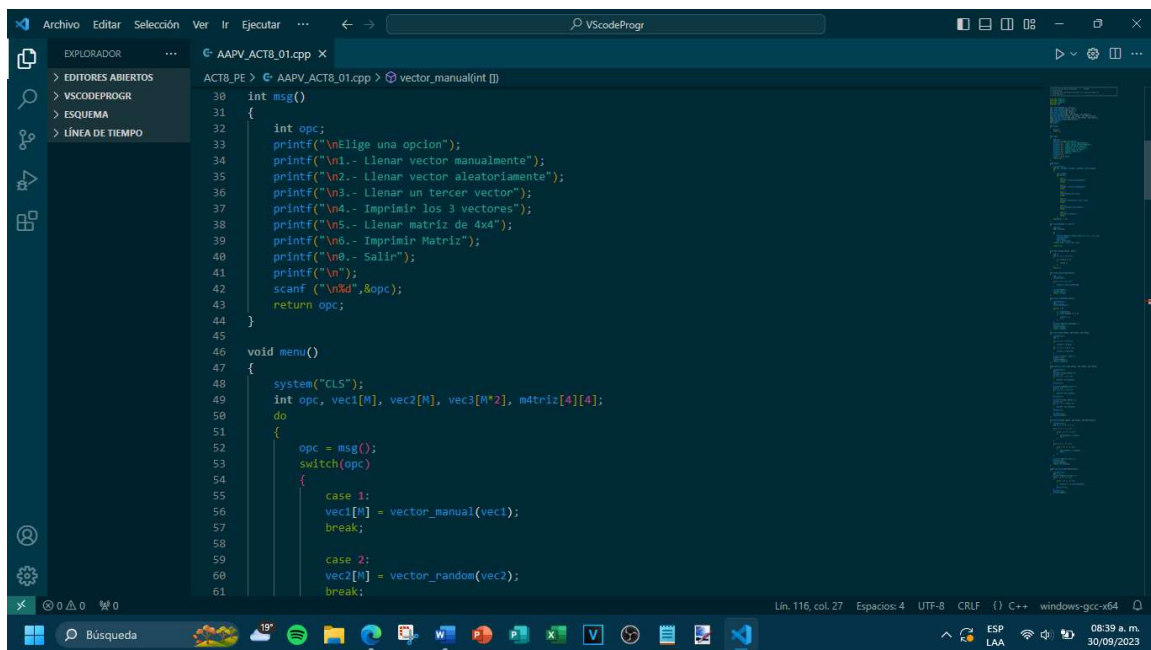


```
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 11
12 12
13 13
14 14
15 15
16 16
17 17
18 18
19 19
20 20
21 21
22 22
23 23
24 24
25 25
26 26
27 27
28 28
29 29
30 30
31 31
32 32
```

ACT8_PE > AAPV_ACT8_01.cpp > vector_manual(int I)

```
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 11
12 12
13 13
14 14
15 15
16 16
17 17
18 18
19 19
20 20
21 21
22 22
23 23
24 24
25 25
26 26
27 27
28 28
29 29
30 30
31 31
32 32
```

Lin. 116, col. 27 Espacios: 4 UTF-8 CRLF {} C++ windows-gcc-x64

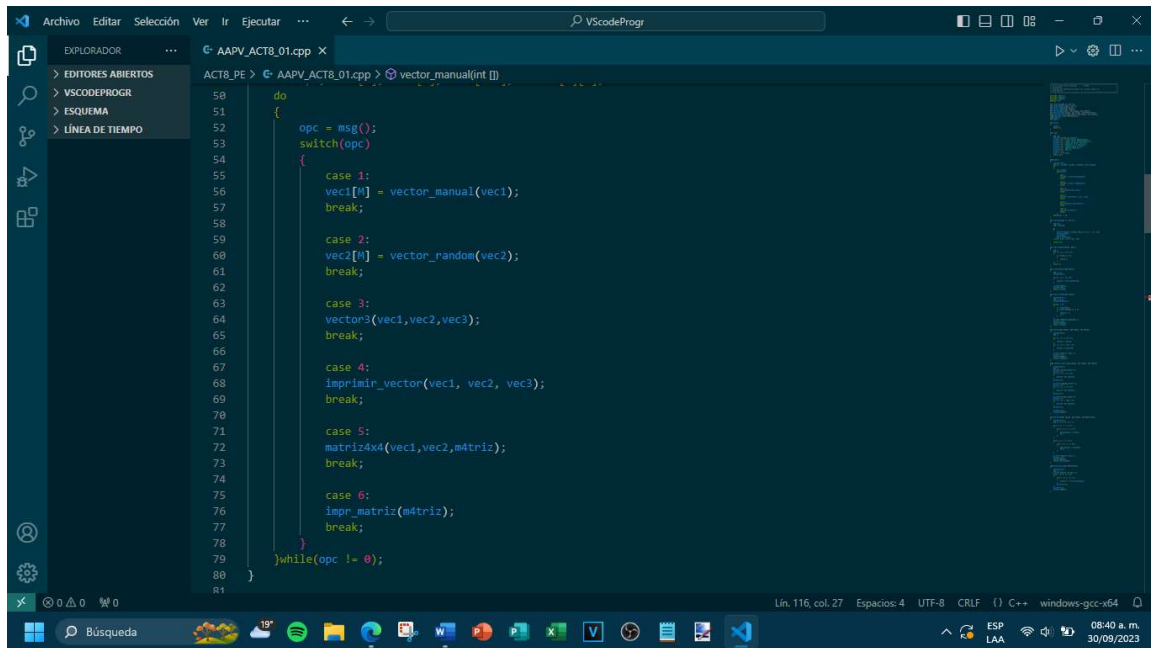


```
30 30
31 31
32 32
33 33
34 34
35 35
36 36
37 37
38 38
39 39
40 40
41 41
42 42
43 43
44 44
45 45
46 46
47 47
48 48
49 49
50 50
51 51
52 52
53 53
54 54
55 55
56 56
57 57
58 58
59 59
60 60
61 61
```

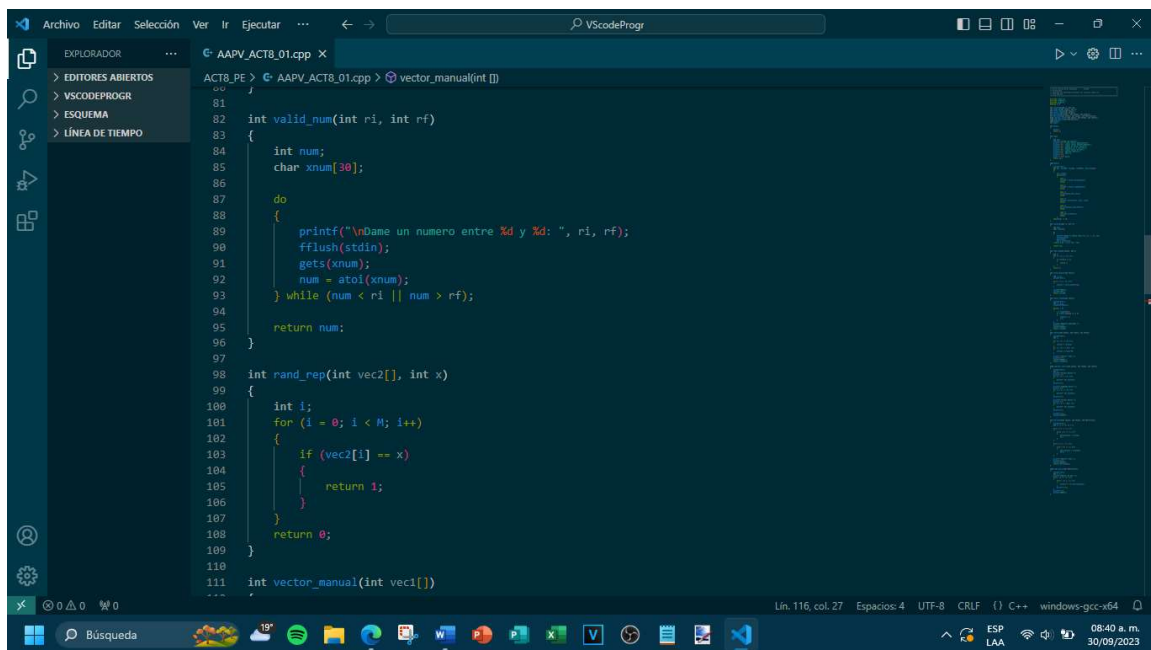
ACT8_PE > AAPV_ACT8_01.cpp > vector_manual(int I)

```
30 30
31 31
32 32
33 33
34 34
35 35
36 36
37 37
38 38
39 39
40 40
41 41
42 42
43 43
44 44
45 45
46 46
47 47
48 48
49 49
50 50
51 51
52 52
53 53
54 54
55 55
56 56
57 57
58 58
59 59
60 60
61 61
```

Lin. 116, col. 27 Espacios: 4 UTF-8 CRLF {} C++ windows-gcc-x64



```
ACT8_PE > AAPV_ACT8_01.cpp > vector_manual(int I)
50 do
51 {
52     opc = msg();
53     switch(opc)
54     {
55         case 1:
56             vec1[M] = vector_manual(vec1);
57             break;
58         case 2:
59             vec2[M] = vector_random(vec2);
60             break;
61         case 3:
62             vector3(vec1,vec2,vec3);
63             break;
64         case 4:
65             imprimir_vector(vec1, vec2, vec3);
66             break;
67         case 5:
68             matriz4x4(vec1,vec2,m4triz);
69             break;
70         case 6:
71             impr_matriz(m4triz);
72             break;
73     }
74 }while(opc != 0);
75
76
77
78
79
80 }
```



```
ACT8_PE > AAPV_ACT8_01.cpp > vector_manual(int I)
80
81
82 int valid_num(int ri, int rf)
83 {
84     int num;
85     char xnum[30];
86
87     do
88     {
89         printf("\nDame un numero entre %d y %d: ", ri, rf);
90         fflush(stdin);
91         gets(xnum);
92         num = atoi(xnum);
93     } while (num < ri || num > rf);
94
95     return num;
96 }
97
98 int rand_rep(int vec2[], int x)
99 {
100     int i;
101     for (i = 0; i < M; i++)
102     {
103         if (vec2[i] == x)
104         {
105             return 1;
106         }
107     }
108     return 0;
109 }
110
111 int vector_manual(int vec1[])
112 ...
```

```
110
111 int vector_manual(int vec1[])
112 {
113     int i = 0;
114     system("CLS");
115
116     for(i = 0; i < M; i++)
117     {
118         vec1[i] = valid_num(30, 70);
119     }
120
121     system("PAUSE");
122     system("CLS");
123     return vec1[M];
124 }
125
126 int vector_random(int vec2[])
127 {
128     system("CLS");
129     int i = 0, x;
130     srand(time(NULL));
131
132     while(i < M)
133     {
134         x = rand()%20+1;
135         if (rand_rep(vec2, x) == 0)
136         {
137             vec2[i] = x;
138             i++;
139         }
140     }
141     printf("\nNumeros generados.");
```

```
140     }
141     printf("\nNumeros generados.");
142     printf("\n");
143     system("PAUSE");
144     return vec2[M];
145 }
146
147 int vector3(int vec1[], int vec2[], int vec3[])
148 {
149     system("CLS");
150     int i;
151
152     for (i = 0; i < M; i++)
153     {
154         vec3[i] = vec1[i];
155     }
156     for (i = M; i < M*2; i++)
157     {
158         vec3[i] = vec2[i-M];
159     }
160
161     printf("\nVector llenado.");
162     printf("\n");
163     system("PAUSE");
164     return vec3[M*2];
165 }
166
167 void imprimir_vector(int vec1[], int vec2[], int vec3[])
168 {
169     system("CLS");
170     int i;
171     printf("\nVector 1: ");
```

This screenshot shows the VS Code editor with the file `AAPV_ACT8_01.cpp` open. The code defines a function `vector_manual(int I)` which calls `imprimir_vector`. The `imprimir_vector` function prints three vectors: `vec1`, `vec2`, and `vec3`. Each vector is printed on a new line, with a label like `"\nPrimer Vector:"` followed by the vector elements separated by spaces. The function also includes `system("CLS");` to clear the screen and `system("PAUSE");` to keep the window open.

```
167 void imprimir_vector(int vec1[], int vec2[], int vec3[])
168 {
169     system("CLS");
170     int i;
171     printf("\nPrimer Vector:");
172     printf("\n");
173     for (i = 0; i < M; i++)
174     {
175         printf("%d", vec1[i]);
176     }
177     printf("\n");
178     printf("\nSegundo Vector:");
179     printf("\n");
180     for (i = 0; i < M; i++)
181     {
182         printf("%d", vec2[i]);
183     }
184     printf("\n");
185     printf("\nTercer Vector:");
186     printf("\n");
187     for (i = 0; i < M*2; i++)
188     {
189         printf("%d", vec3[i]);
190     }
191     printf("\n");
192     printf("\n");
193     printf("\n");
194     printf("\n");
195     system("PAUSE");
196 }
```

This screenshot shows the VS Code editor with the file `AAPV_ACT8_01.cpp` open. The code defines a function `matriz4x4(int vec1[], int vec2[], int matriz[][4])` which fills a 4x4 matrix. The matrix is filled with values from `vec1` and `vec2` in a specific pattern. The function also includes `system("CLS");` to clear the screen and `system("PAUSE");` to keep the window open.

```
198 int matriz4x4(int vec1[], int vec2[], int matriz[][4])
199 {
200     system("CLS");
201     int i, j, k = 0, k2 = 0;
202     for(i = 0; i < 2; i++)
203     {
204         for(j = 0; j < 4; j++)
205         {
206             matriz[i][j] = vec1[k];
207             k++;
208         }
209     }
210     for(i = 2; i < 4; i++)
211     {
212         for(j = 0; j < 4; j++)
213         {
214             matriz[i][j] = vec2[k2];
215             k2++;
216         }
217     }
218     printf("\nMatriz Llena.");
219     printf("\n");
220     system("PAUSE");
221     return matriz[4][4];
222 }
223 void impr_matrix(int matriz[][4])
224 {
225 }
```

```
ACT8_PE > C:\AAPV_ACT8_01.cpp > vector.manual(int D)
217     matriz[i][j] = vec2[k2++];
218     }
219 }
220 }
221
222 printf("\nMatriz llena.");
223 printf("\n");
224 system("PAUSE");
225 return matriz[4][4];
226 }
227
228 void impr_matriz(int matriz[][4])
229 {
230     system("CLS");
231     int i, j;
232     printf("\nMatriz de 4x4: ");
233     for(i = 0; i < 4; i++)
234     {
235         for(j = 0; j < 4; j++)
236         {
237             printf("\t %d",matriz[i][j]);
238         }
239         printf("\n");
240     }
241     printf("\n");
242     system("PAUSE");
243 }
```

Dame un numero entre 30 y 70: 53
Dame un numero entre 30 y 70: 35
Dame un numero entre 30 y 70: 66
Dame un numero entre 30 y 70: 34
Dame un numero entre 30 y 70: 31
Dame un numero entre 30 y 70: 57
Dame un numero entre 30 y 70: 61
Dame un numero entre 30 y 70: 68
Dame un numero entre 30 y 70: 47
Dame un numero entre 30 y 70: 42
Presione una tecla para continuar . . .

Primer Vector:[,53,35,66,34,31,57,61,68,47,42]
Segundo Vector:[,14,19,1,17,6,13,8,11,9,7]
Tercer Vector:[,53,35,66,34,31,57,61,68,47,42,14,19,1,17,6,13,8,11,9,7]
Presione una tecla para continuar . . .

Matriz de 4x4: 53 35 66 34
 31 57 61 68
 14 19 1 17
 6 13 8 11

Presione una tecla para continuar . . .

Matriz de 4x4:
 53 35 66 34
 31 57 61 68
 18 14 10 11
 19 5 2 4

Presione una tecla para continuar . . .