

**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA**

**Faculta de Ingeniería, Arquitectura y Diseño**



Ingenieria en Computacion

Programación Estructurada

ALUMNO: Miguel Angel Portillo Attwell

MATRÍCULA: 370097

GRUPO: 432

PROFESOR: Pedro Nunez Yepiz



## Programación Estructurada



```
MAPA_PE_ACT8.cpp
ACTIVIDAD 9 > G- MAPA_PE_ACT8.cpp > ...
1 // MIGUEL ANGEL PORTILLO ATTWELL 370097
2 // FECHA: 3 DE OCTUBRE DEL 2023
3 // DESCRIPCION: PROGRAMA QUE UTILIZE FUNCIONES DE LLENADO DE VECTORES Y MATRIZES Y LAS IMPRIMA
4 // MAPA_PE_ACT8.cpp
5 #include <stdio.h>
6 #include <stdlib.h>
7 #include <time.h>
8 #define tamav 10
9
10 int vector1[tamav];
11 int vector2[tamav];
12 int vector3[20];
13 int matriz[4][4];
14
15 void menu(void);
16 void vector_uno(void);
17 void vector_dos(void);
18 void vector_tres(void);
19 void impri_vect(void);
20 void matriz4x4(void);
21 void impri_matri(void);
22
23 int main()
24 {
25     menu();
26
27     return 0;
28 }
29
30 void menu(void)
31 {
32     int operacion;
33     do
34     {
35         printf("M E N U\n");
36         printf("INGRESE CON UN DIGITO LA OPRACION QUE QUIERE REALIZAR: \n");
37         printf("[1-LLENAR VECTOR 1]\n");
38         printf("[2-LLENAR VECTOR 2]\n");
39         printf("[3-LLENAR VECTOR 3]\n");
40         printf("[4-IMPRIMIR VECTORES]\n");
41         printf("[5-LLENAR MATRIZ]\n");
42         printf("[6-IMPRIMIR MATRIZ]\n");
43         printf("[0-SALIR]\n");
44         scanf("%d", &operacion);
45         switch (operacion)
46         {
47             case 1:
48                 vector_uno();
49                 break;
50             case 2:
51                 vector_dos();
52                 break;
53             case 3:
54                 vector_tres();
55                 break;
56             case 4:
57                 impri_vect();
58                 break;
59             case 5:
60                 matriz4x4();
61                 break;
62             case 6:
63                 impri_matri();
64                 break;
65             default:
66                 break;
67         }
68     } while (operacion != 0);
69 }
70
71
```



```
72 void vector_uno(void)
73 {
74     int i;
75     int num;
76     printf("LLENADO DE VECTOR 1\n");
77     printf("LLENAR SOLO CON NUMEROS QUE ESTE ENTRE EL 30 Y EL 70\n");
78     for (i = 0; i < 10; i++)
79     {
80         printf("INGRESAR EL VALOR DE LA POSICION %d DEL VECTOR: \n", i + 1);
81         scanf("%d", &num);
82         if (num > 30 && num < 70)
83         {
84             vector1[i] = num;
85         }
86         else
87         {
88             i--;
89         }
90     }
91     for (i = 0; i < 10; i++)
92     {
93         printf("VALOR DE LA POSICION %d DEL VECTOR: [%d]\n", i + 1, vector1[i]);
94     }
95     printf("[SE LLENO EL VECTOR 3 CORRECTAMENTE]");
96 }
```

```
98 void vector_dos(void)
99 {
100     int i;
101     int num;
102     printf("LLENADO DE VECTOR 2");
103     srand(time(NULL));
104     for (i = 0; i < 10; i++)
105     {
106         num = rand() % 21;
107         vector2[i] = num;
108     }
109     for (i = 0; i < 10; i++)
110     {
111         printf("EL VALOR DE LA POSICION %d DEL VECTOR: [%d]\n", i + 1, vector2[i]);
112     }
113     printf("[SE LLENO EL VECTOR 3 CORRECTAMENTE]");
114 }
```



```
116 void vector_tres(void)
117 {
118     int i;
119     int j;
120     printf("LLENADO DE VECTOR 3\n");
121     for (i = 0; i < 10; i++)
122     {
123         vector3[i] = vector1[i];
124     }
125     for (i = 10, j = 0; i < 20, j < 10; i++, j++)
126     {
127         vector3[i] = vector2[j];
128     }
129     printf("[SE LLENO EL VECTOR 3 CORRECTAMENTE]");
130 }
131
```

```
132 void impri_vect(void)
133 {
134     int i;
135     printf("\n");
136     printf("VECTOR 1:\n");
137     for (i = 0; i < 10; i++)
138     {
139         printf("%d.-[%d]\n", i + 1, vector1[i]);
140     }
141     printf("\n");
142     printf("VECTOR 2:\n");
143     for (i = 0; i < 10; i++)
144     {
145         printf("%d.-[%d]\n", i + 1, vector2[i]);
146     }
147     printf("\n");
148     printf("VECTOR 3:\n");
149     for (i = 0; i < 20; i++)
150     {
151         printf("%d.-[%d]\n", i + 1, vector3[i]);
152     }
153 }
154
```



```
155 void matriz4x4(void)
156 {
157     int i;
158     int j;
159     int k=0;
160     printf("LLENADO DE MATRIZ 4 X 4\n");
161     for (i = 0; i < 4; i++)
162     {
163         for (j = 0; j < 4; j++)
164         {
165             matriz[i][j] = vector3[k];
166             k++;
167         }
168     }
169 }
```

```
170
171 void impri_matri(void)
172 {
173     int i;
174     int j;
175     int k=0;
176     printf("IMPRESION DE MATRIZ");
177     printf("-----\n");
178     printf("[LUGAR]\t[1]\t[2]\t[3]\t[4]\n");
179     for (i = 0; i < 4; i++)
180     {
181         printf("[%d]", i+1);
182         for (j = 0; j < 4; j++)
183         {
184             matriz[i][j];
185             printf("\t[%d]",matriz[i][j]);
186         }
187         printf("\n");
188     }
189 }
```

GITHUB: [https://github.com/MAIKYPORTILLO67/PROGRAMACION\\_ESTRUCTURADA](https://github.com/MAIKYPORTILLO67/PROGRAMACION_ESTRUCTURADA)