## 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 •
      #define tamav 10
      int vector1[tamav];
      int vector3[20];
      int matriz[4][4];
      void vector_dos(void);
     void vector_tres(void);
      void impri_vect(void);
     void matriz4x4(void);
      void impri_matri(void);
      void menu(void)
            int operacion;
                 printf("M E N U\n");
printf("INGRESE CON UN DIGITO LA OPRACION QUE QUIERE REALIZAR: \n");
printf("[1-LLENAR VECTOR 1]\n");
printf("[2-LLENAR VECTOR 2]\n");
printf("[3-LLENAR VECTOR 3]\n");
printf("[4-IMPRIMIR VECTORES]\n");
                 printf("[6-LIENAR MATRIZ]\n");
printf("[6-LMPRIMIR MATRIZ]\n");
printf("[6-SALIR]\n");
scanf("%d", &operacion);
                 switch (operacion)
                 case 1:
                     vector_uno();
                       vector_dos();
                       vector_tres();
                 case 4:
                    impri_vect();
                 case 5:
                      matriz4x4();
                       impri_matri();
             } while (operacion != 0);
```



## Programación Estructurada



```
void vector_uno(void)
{
    int i;
    int num;
    printf("LLENADO DE VECTOR 1\n");
    printf("LLENAR SOLO CON NUMEROS QUE ESTE ENTRE EL 30 Y EL 70\n");
    for (i = 0; i < 10; i++)
    {
        printf("INGRESAR EL VALOR DE LA POCISION %d DEL VECTOR: \n", i + 1);
        scanf("%d", &num);
        if (num > 30 && num < 70)
        {
            vector1[i] = num;
        }
        else
        {
            i i--;
        }
        }
        printf("VALOR DE LA POCISION %d DEL VECTOR: [%d]\n", i + 1, vector1[i]);
        }
        printf("[SE LLENO EL VECTOR 3 CORRECTAMENTE]");
}</pre>
```



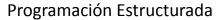
## Programación Estructurada



```
116 ∨ void vector tres(void)
117
118
          int i;
119
          int j;
120
          printf("LLENADO DE VECTOR 3\n");
          for (i = 0; i < 10; i++)
121 🗸
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("\n");
149     for (i = 0; i < 20; i++)
150     {
151         printf("%d.-[%d]\n", i + 1, vector3[i]);
152     }
153     }
154</pre>
```







GITHUB: https://github.com/MAIKYPORTILLO67/PROGRAMACION ESTRUCTURADA