



Universidad Autónoma de Baja California
Facultad de Ingeniería Arquitectura y Diseño



Ingeniería en Software y Tecnologías Emergentes.

Programación estructurada

Actividad 8

Funciones e introducción a arreglos C

Danna Guadalupe Sandez Islas

373080

1.- LLENAR VECTOR 1 (MANUALMENTE)

Desarrollo:

```
/* Funcion para llenar un vector con datos dados por el usuario */
void llenar(int vector1[])
{
    // variables locales
    int num;
    int i;
    /* desarrollo de la funcion */
    for (i = 0; i < 10; i++)
    {
        printf(" Posicion [%d] del vector\n", i);
        num = validar("Ingresa un numero entre 30 y 70 por favor: ", 30, 70);
        vector1[i] = num;
        system("CLS");
    }
}
```

2.- LLENAR VECTOR 2 ALEATORIAMENTE

Función auxiliar:

```
/* Funcion que verifica si el numero aleatorio dado ya se encuentra dentro del vector*/
int repetido(int num, int i, int vector2[])
{
    for (int j = 0; j < i; j++)
    {
        if (num == vector2[j])
        {
            return 1; // si el numero ya esta dentro del vector, retorna 1
        }
    }
    return 0; // si el numero no se encuentra dentro del vector, retorna 0
}
```

Desarrollo de función:

```
/* Funcion para llenar un vector de manera aleatoria sin numeros repetidos */
void llenar2(int vector2[])
{
    // variables locales
    int num;
    int i;
    /* Desarrollo de funcion */
    for (i = 0; i < 10; i++)
    {
        do
        {
            num = rand() % 20 + 1;
        } while (repetido(num, i, vector2));
        vector2[i] = num;
    }
    printf("El vector se lleno con numeros aleatorios entre 1-20 exitosamente\n");
}
```

3.- LLENAR VECTOR 3 (CON VECTOR1 Y VECTOR2)

Desarrollo de función:

```
/* Funcion para llenar un vector con otros dos vectores */
void llenar3(int vector3[], int vect2[], int vect1[])
{
    for (int i = 0; i < 20; i++)
    {
        if (i < 10) // Se llenan las primeras 10 posiciones con el vector 1
        {
            vector3[i] = vect1[i];
        }
        else // se llenan las ultimas 10 posiciones con el vector 2
        {
            vector3[i] = vect2[i - 10];
        }
    }
    printf("Vector de 20 espacios completo exitosamente!\n");
}
```

4.- IMPRIMIR VECTORES

Desarrollo de función:

```
/* Funcion para imprimir los vectores */
void imprimir(int vect1[], int vect2[], int vect3[])
{
    int i;
    printf("--Vector 1--\n");
    for (i = 0; i < 10; i++)
    {
        printf("Vector[%d] --> %d\n", i, vect1[i]);
    }
    system("PAUSE");
    system("CLS");
    printf("--Vector 2--\n");
    for (i = 0; i < 10; i++)
    {
        printf("Vector[%d] --> %d\n", i, vect2[i]);
    }
    system("PAUSE");
    system("CLS");
    printf("--Vector 3--\n");
    for (i = 0; i < 20; i++)
    {
        printf("Vector[%d] --> %d\n", i, vect3[i]);
    }
}
```

Salida:

```
--Vector 1--  
Vector[0] --> 30  
Vector[1] --> 30  
Vector[2] --> 30  
Vector[3] --> 30  
Vector[4] --> 30  
Vector[5] --> 30  
Vector[6] --> 30  
Vector[7] --> 30  
Vector[8] --> 30  
Vector[9] --> 30  
Presione una tecla para continuar . . .
```

```
--Vector 2--  
Vector[0] --> 5  
Vector[1] --> 13  
Vector[2] --> 15  
Vector[3] --> 19  
Vector[4] --> 8  
Vector[5] --> 7  
Vector[6] --> 4  
Vector[7] --> 6  
Vector[8] --> 9  
Vector[9] --> 12  
Presione una tecla para continuar . . .
```

```
--Vector 3--  
Vector[0] --> 30  
Vector[1] --> 30  
Vector[2] --> 30  
Vector[3] --> 30  
Vector[4] --> 30  
Vector[5] --> 30  
Vector[6] --> 30  
Vector[7] --> 30  
Vector[8] --> 30  
Vector[9] --> 30  
Vector[10] --> 4  
Vector[11] --> 16  
Vector[12] --> 12  
Vector[13] --> 18  
Vector[14] --> 6  
Vector[15] --> 2  
Vector[16] --> 14  
Vector[17] --> 17  
Vector[18] --> 19  
Vector[19] --> 10  
Presione una tecla para continuar . . .
```

5.- LLENA MATRIZ 4 X 4

Desarrollo de función:

```
/* Funcion para llenar una matriz con 2 vectores */
void matriz4x4(int matriz[][4], int vect1[], int vect2[])
{
    // Variables locales
    int i, j, k = 0;
    /* Desarrollo de funcion */
    for (i = 0; i < 4; i++)
    {
        for (j = 0; j < 4; j++)
        {
            if (k < 10) // Se llenan los primeros 10 espacios con el vector 1
            {
                matriz[i][j] = vect1[k];
            }
            else // Se llenan los ultimos 6 espacios con el vector 2
            {
                matriz[i][j] = vect2[k - 6];
            }
            k++;
        }
    }
    printf("La matriz 4x4 se lleno exitosamene!\n");
}
```

6.- IMPRIMIR MATRIZ

Desarrollo de función:

```
/* Funcion para imprimir la matriz */
void imprimirMatriz(int matriz[][4])
{
    printf("~~~ Matriz 4x4 ~~~\n");
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            printf("%d ", matriz[i][j]);
        }
        printf("\n");
    }
}
```

Salida:

```
~~~ Matriz 4x4 ~~~
30 30 30 30
30 30 30 30
30 30 6 2
14 17 19 10
Presione una tecla para continuar . . .
```