Universidad Autónoma de Baja California

Ingeniería en Software y Tecnologías Emergentes.

ACTIVIDAD 9. FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA

Materia: Programación Estructurada.

Maestro: Pedro Yepiz

Alumno: Diego Quiros 372688

Fecha: 10/08/2023

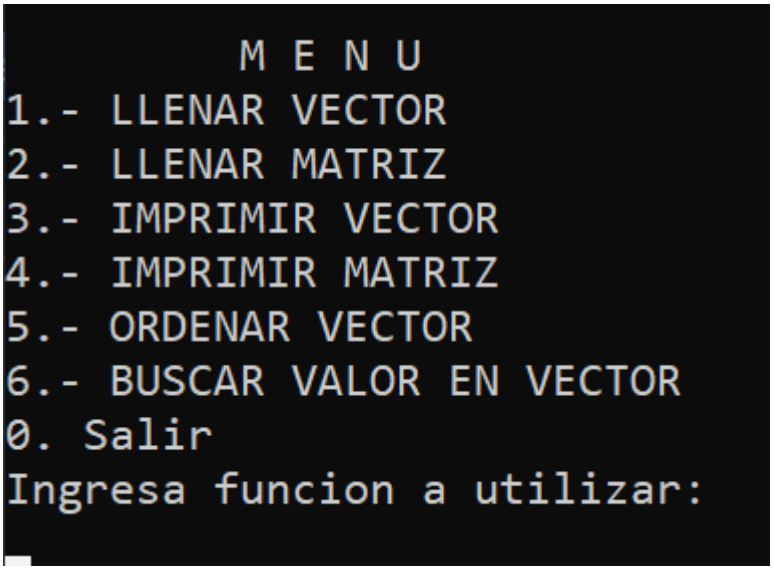
Repositorio.

```
Unset
https://github.com/diegovq12/Programacion_Estructurada_QVDD_932/tree/master/Actividad%209
```

Implementación de todos los programas en un menú:

```c
int msge (void)
{
  int op;
  printf("\n\t M E N U\n");
  printf("1.- LLENAR VECTOR\n");
  printf("2.- LLENAR MATRIZ \n");
  printf("3.- IMPRIMIR VECTOR\n");
  printf("4.- IMPRIMIR MATRIZ\n");
  printf("5.- ORDENAR VECTOR\n");
  printf("6.- BUSCAR VALOR EN VECTOR\n");
  printf("0. Salir\n");
  printf("Ingresa funcion a utilizar:\n");
  op=validInt(0,6);

  return op;
}
```

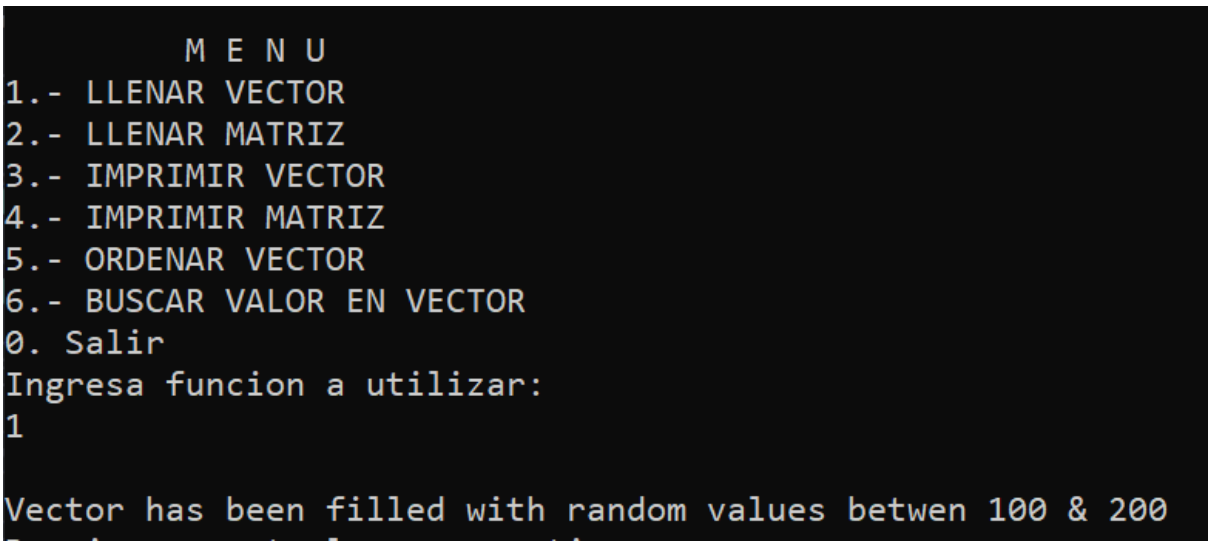


Ejercicio 1.Llenar vector con 15 números, los números generados aleatoriamente, los números entre el rango de 100 al 200 (no repetidos).

C/C++

```
//Funcion que llena un vector con valores aleatorios sin repetirse
//recibe como valor el vector,tamaño y los valores para obtener el
//rango de los valores
//QQVDD_Act9_PE_932
void fill_vectorRand (int *vector,int m,int ri,int rf)
{
  srand(time(NULL));
  int aux,num,i,j;
  for (i = 0; i < m; i++)
  {
    aux=0;
    num=(rand()%(ri-rf))+ri;
    for (j = 0; j<m; j++)
    {
      if (num==vector[j])
      {
        aux=1;
      }
    }

    if (aux==1)
    {
      i--;
    }
    else
    {
      vector[i]=num;
    }
  }
  printf("\nVector has been filled with random values betwen %d &
%d\n",ri,rf);
}
```

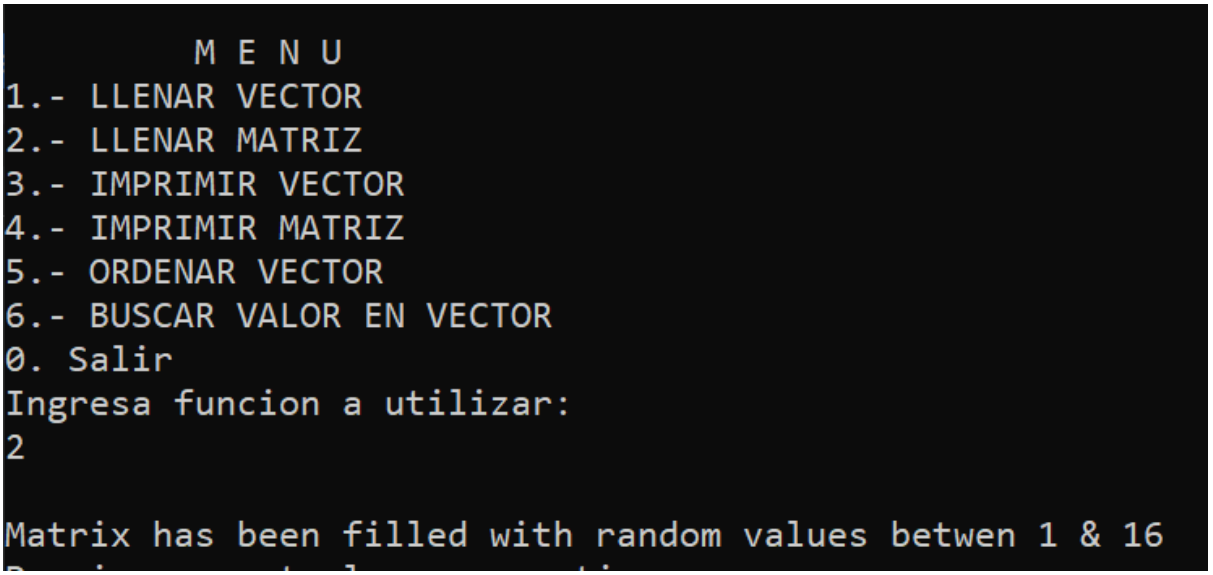


Ejercicio 2.Llenar una matriz de 4x4 con con números generados aleatoriamente, números entre el rango de 1 al 16 (no repetidos).

C/C++

```
//Funcion que llena una matriz 4x4 con valores aleatorios sin repetirse
//recibe como valor la matriz,tamaño y los valores para obtener el
//rango de los valores
////QQVDD_Act9_PE_932
void fill_MatrizRand (int mat[4][4],int m, int n ,int ri,int rf)
{
  srand(time(NULL));
  int i,j,k,l,num;
  int aux;
  for ( i = 0; i < m; i++)
  {
    for ( j = 0; j < n; j++)
    {
      do
      {
        aux=0;
        num=(rand()%(ri-rf))+ri;
        for(k=0 ; k<i ;k++)
        {
          for (l = 0; l < n; l++)
          {
            if (mat[k][l] == num)
            {
              aux=1;
            }
          }
        }
```

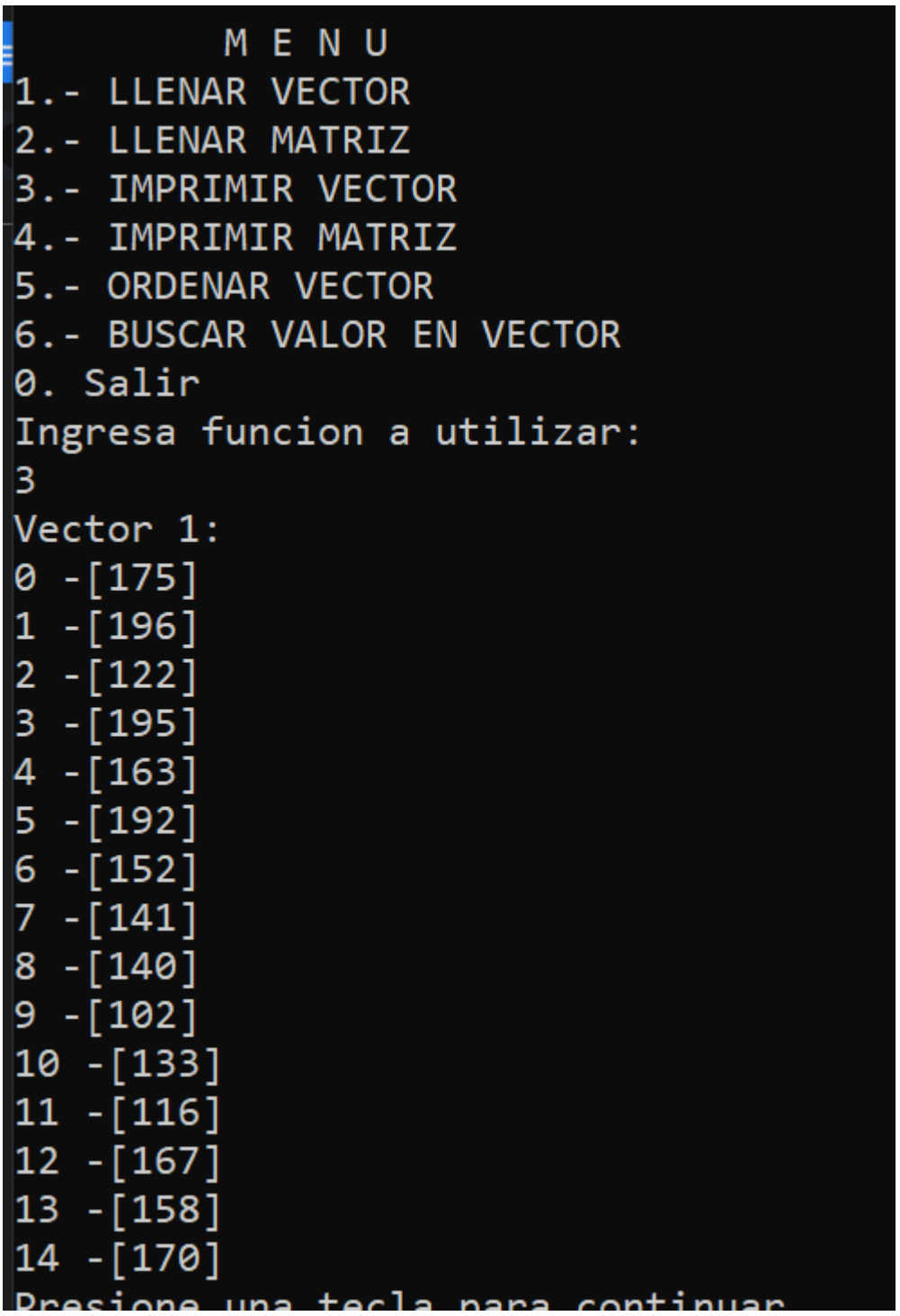
```
      } while (aux==1);
      mat[i][j]=num;
    }
  }
  printf("\nMatrix has been filled with random values betwen %d &
%d\n",ri,rf);
}
```



Ejercicio 3.Imprime el vector que se envíe, donde la función recibe como parámetro el vector,tamaño, nombre del vector.

C/C++

```
//Funcion que Imprime un vector
//recibe como valor el vector y tamaño
////QQVDD_Act9_PE_932
void printVector (int vect[], int m, char name[20])
{
  int i;
  printf("%s: \n",name);
  for ( i = 0; i < m; i++)
  {
    printf("%d -[%2d]\n",i,vect[i]);
  }
}
```



Ejercicio 4. Imprime la matriz sin importar el tamaño de la matriz recibiendo como parámetros la matriz, la cantidad de renglones y columnas, así como nombre que se le dará a la matriz.

C/C++

```
//Funcion que Imprime una Matriz
//recibe como valor la matriz y tamaño
////QQVDD_Act9_PE_932
void printMatrix (int m, int n, int mat[m][n],char name[20])
{
  int i,j;

  printf("%s:\n",name);

  for ( i = 0; i < m; i++)
  {
    printf("%d -",i);
   for (j = 0; j < n; j++)
   {
     printf("[%2d]",mat[i][j]);
   }
   printf("\n");
  }
}
```

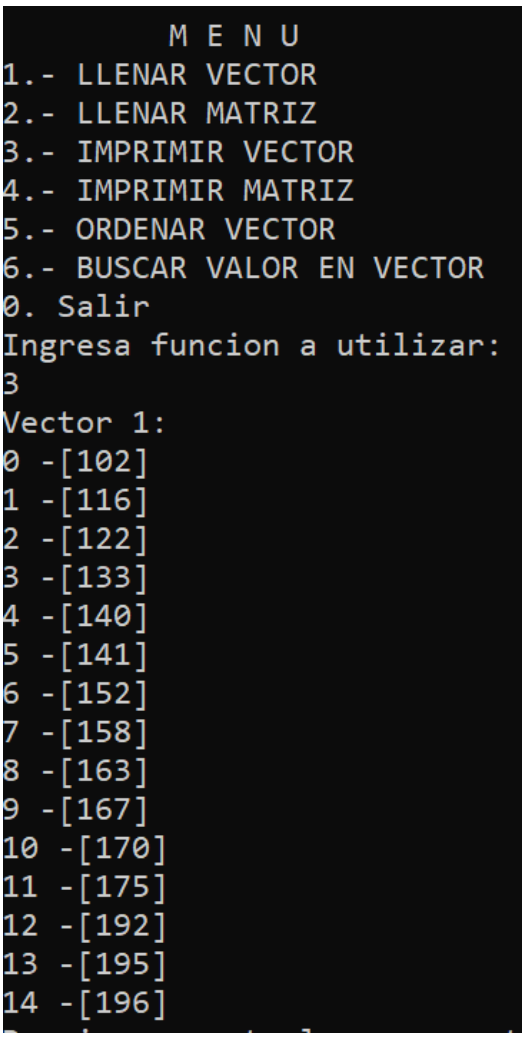
```
         M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0. Salir
Ingresa funcion a utilizar:
4
Matriz 1:
0 -[14][ 4][ 3][ 4]
1 -[ 9][12][11][ 8]
2 -[15][ 6][ 5][ 1]
3 -[ 7][13][ 2][10]
Presione una tecla para continuar . . .
```

Ejercicio 5. Usar función que ordene el vector por el método de ordenación de la Burbuja mejorada.

C/C++

```
//Funcion que ordena un vector de mayor
//a menor mediante el metodo de la
//burbuja-Solicita el vector y su tamano
void bubbleSort (int vector[], int m)
{
  int i,j;
  int aux;

  for ( i = 0; i < m-1; i++)
  {
    for (j = i+1; j < m; j++)
    {
      if(vector[i]>vector[j])
      {
        aux=vector[i];
        vector[i]=vector[j];
        vector[j]=aux;
      }
    }
  }
}
```



Ejercicio 6. Buscar un valor en el vector usando el método de búsqueda secuencial.

C/C++

```
//Funcion que busca un valor pedido por el usuario dentro
//de un vector
//solicita vector, tamaño y valor a buscar
////QQVDD_Act9_PE_932
void findVectorValue (int vector[], int m, int val)
{
  int i;
  int aux=0;

  for (i = 0; i < m; i++)
  {
    if (vector[i]==val)
    {
      printf("El Valor se encontro en el espacio %d\n",i);
      aux=1;
```

```
      }
    }
    if (aux==0)
    {
      printf("El valor %d no se ha encontrado en el vector\n",val);
    }
  }
```

```
         M E N U
1.- LLENAR VECTOR
2.- LLENAR MATRIZ
3.- IMPRIMIR VECTOR
4.- IMPRIMIR MATRIZ
5.- ORDENAR VECTOR
6.- BUSCAR VALOR EN VECTOR
0. Salir
Ingresa funcion a utilizar:
6
Que valor deseas buscar: 147
El valor 147 no se ha encontrado en el vector
Quisieras buscar otro valor?
        [ Yes[1]/No[2] ]1
Que valor deseas buscar: 148
El valor 148 no se ha encontrado en el vector
Quisieras buscar otro valor?
        [ Yes[1]/No[2] ]1
Que valor deseas buscar: 158
El Valor se encontro en el espacio 7
Quisieras buscar otro valor?
        [ Yes[1]/No[2] ]2
Presione una tecla para continuar . . .
```