

1. Llevar a cabo un programa que permita cargar mediante el teclado una matriz de 3 x 3 (3 filas, 3 columnas) con razas de perros. Una vez cargada, se debe permitir al usuario ingresar por teclado una raza de perro que quiera buscar; el programa deberá de poder encontrarla y mostrar un mensaje con la posición (fila, columna) donde la encontró. En caso de que la raza de perro no se encuentre en la matriz deberá informar la situación mediante un mensaje en pantalla.

```
1  Algoritmo matrizRaza
2      //Entradas: raza como caracter
3      //fila , columna como entero
4      //bandera como logico
5      //Salidas. "Encontramos la raza, está en la posición " fila, columna
6      //"La raza ingresada no se encuentra en la matriz"
7      //casos de prueba= pastor aleman, mestizo , husky, pug, xolo, dalmata, pitbull, corgi , boxer
8      //raza= pug "Raza encontrada. Está en la posición (1 , 0)
9
10     //Definicion de variables
11     Definir raza Como Caracter;
12     Definir fila, columna como entero;
13     Definir bandera como logico;
14     Definir matriz Como caracter;
15     Dimensionar matriz[3,3];
16
17     //Inicializacion
18     raza← " ";
19     fila← 0;
20     columna← 0;
21     bandera← falso;
22
23     //Ingreso de datos
24     Para fila← 0 Hasta 2 Hacer
25         Para columna← 0 Hasta 2 Hacer
26             Escribir "Ingrese la raza que desea guardar";
27             Leer matriz[fila, columna];
28             matriz[fila, columna]← Mayusculas(matriz[fila, columna]);
29             Escribir " dato en (" fila,", " columna ")"; //Para que cada que ingrese un dato vea donde se guardó
30         Fin Para
31     Fin Para
32
33     //Pido al usuario que ingrese la raza que desea buscar
34
35     Escribir "Ingrese la raza que desea buscar";
36     Leer raza;
37     raza← Mayusculas(raza);
38
39     //recorro la matriz para encontrar la raza solicitada
40     Para fila← 0 Hasta 2 Hacer
41         Para columna← 0 Hasta 2 Hacer
42             Si matriz[fila, columna]==raza Entonces
43                 bandera← verdadero;
44                 Escribir "Raza encontrada en la posición (" fila , " , " columna, " )";
45             FinSi
46         Fin Para
47     Fin Para
48
49     Si bandera==falso Entonces
50         Escribir "La raza ingresada no se encuentra en la matriz";
51     FinSi
52
53 FinAlgoritmo
```

### Algoritmo matrizRaza

```
//Entradas: raza como caracter
//fila , columna como entero
//bandera como logico
//Salidas. "Encontramos la raza, está en la posición " fila, columna
//"La raza ingresada no se encuentra en la matriz"
//casos de prueba= pastor aleman, mestizo , husky, pug, xolo, dalmata, pitbull, corgi , boxer
//raza= pug "Raza encontrada. Está en la posición (1 , 0)
```

  

```
//Definicion de variables
Definir raza Como Caracter;
Definir fila, columna como entero;
Definir bandera como logico;
Definir matriz Como caracter;
Dimensionar matriz[3,3];

//Inicializacion
raza← " ";
fila← 0;
columna← 0;
bandera← falso;

//Ingreso de datos
Para fila← 0 Hasta 2 Hacer
    Para columna← 0 Hasta 2 Hacer
        Escribir "Ingrese la raza que desea guardar
        Leer matriz[fila, columna];
        matriz[fila, columna]← Mayusculas(matriz[f
        Escribir " dato en (" fila," , " columna "
```

  

```
Fin Para
Fin Para
```

▶ PSeInt - Ejecutando proceso MATRIZRAZA

```
dato en (1, 2)
Ingrese la raza que desea guardar
> pitbull
dato en (2, 0)
Ingrese la raza que desea guardar
> corgi
dato en (2, 1)
Ingrese la raza que desea guardar
> boxer
dato en (2, 2)
Ingrese la raza que desea buscar
> pug
Raza encontrada en la posición (1 , 0 )
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible

Razón de mi elección.

Decidí incluir este programa en mi portafolio, ya que fue el primero que realicé usando matrices y me sirvió de referencia para entender un poco mejor su funcionamiento