



Universidad de las Fuerzas Armadas ESPE

Departamento:

Carrera: Ingeniería en Electrónica y Automatización

Taller académico N°: 7

Parcial Nro:2

1. Información General

- **Asignatura:** Fundamentos de Programación
 - **Apellidos y nombres de los estudiantes:** Alvear Alexander, Campoverde Anthony, Velecela Mateo
 - **NRC:** 20823
 - **Fecha de realización:** 25/06/2025
-

2. Objetivo del Taller y Desarrollo

Objetivo del Taller:

Desarrollar habilidades prácticas en el manejo de requisitos funcionales, pseudocódigo en PseInt, aplicando matrices y lógica para operaciones básicas de recorrido, ordenamiento y manipulación de datos, con énfasis en la eficiencia del código.

Desarrollo:

Aprendimos a manejar la inversa de las matrices, resolviendo ejercicios prácticos para la aplicación intuitiva de conocimientos. Al final, discutimos los resultados y reforzamos la lógica.

U2 T7 VECTORES Y FUNCIONES

EJERCICIO 1 MATRICES:



Problema 2.2.4 Intercambiar las filas i y j de una matriz. Escriba un programa que intercambie las filas i y j de una matriz de enteros de N componentes, siendo i, j dos valores introducidos por teclado.

La solución se muestra en el diagrama de la figura 2.13, junto con su tabla de objetos y codificación. La dificultad del problema reside en intercambiar las filas sin perder información, tal y como pasa aquí al intercambiar el elemento 7 de la fila a y b
`mat[a][6]=mat[b][6]; //Sobrescribe mat[a][6] y se pierde su valor`
`mat[b][6]=mat[a][6]; //Queda mat[a][6] con el mismo valor que mat[b][6]`

Es necesario usar una variable auxiliar para no perder el valor original de `mat[a][6]`
`Aux=mat[a][6]`

```
mat[a][6]=mat[b][6]
mat[b][6]=aux;
```

- **Tabla de Objetos**

OBJETO	NOMBRE	VARIABLE	TIPO
M1	$N, i, j, \text{fila}_j, \text{aux}, k$	variable	Entero
M2	matriz	variable	Entero

- **Requisitos funcionales**

RF1: El sistema debe permitir al usuario ingresar el tamaño de la matriz (número de filas y columnas).

RF2: El sistema debe permitir al usuario ingresar los elementos de la matriz de enteros.

RF3: El sistema debe solicitar al usuario los índices de las dos filas que desea intercambiar (i y j).

RF4: El sistema debe validar que los índices i y j se encuentren dentro del rango válido de la matriz.

RF5: El sistema debe intercambiar correctamente las filas i y j usando una variable auxiliar para evitar la pérdida de datos.

RF6: El sistema debe mostrar en pantalla la matriz original antes del intercambio.

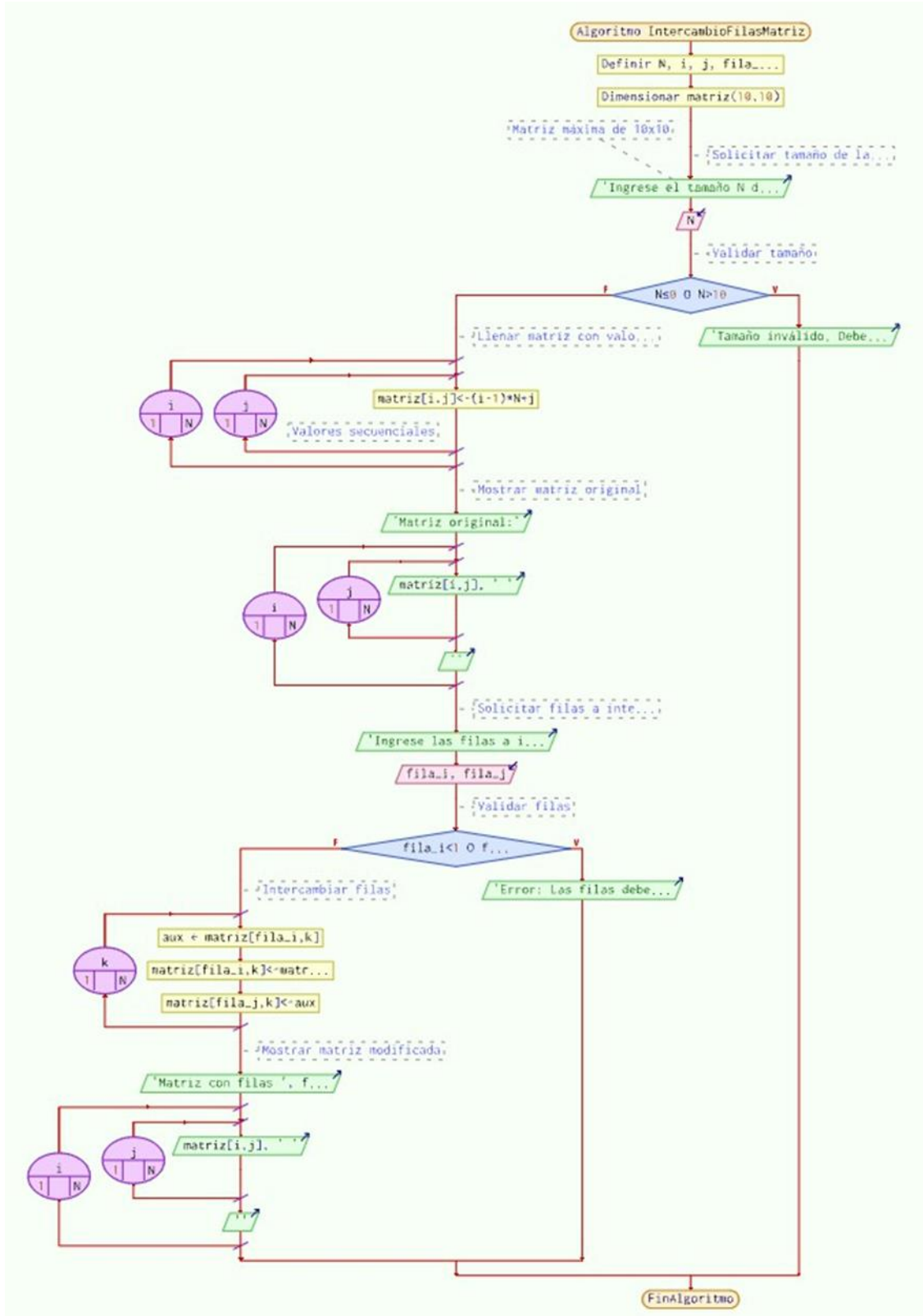
RF7: El sistema debe mostrar en pantalla la matriz actualizada después del intercambio de filas.



- Seudocódigo

```
1  Algoritmo IntercambioFilasMatriz
2  Definir N, i, j, fila_i, fila_j, aux, k Como Entero
3  Dimension matriz[10,10] // Matriz máxima de 10x10
4
5  // Solicitar tamaño de la matriz
6  Escribir 'Ingrese el tamaño N de la matriz cuadrada (N x N, máximo 10): '
7  Leer N
8
9  // Validar tamaño
10 Si N ≤ 0 O N > 10 Entonces
11     Escribir 'Tamaño inválido. Debe ser entre 1 y 10.'
12 Sino
13     // Llenar matriz con valores de ejemplo
14     Para i ← 1 Hasta N Hacer
15         Para j ← 1 Hasta N Hacer
16             matriz[i,j] ← (i-1)*N + j // Valores secuenciales
17         FinPara
18     FinPara
19
20     // Mostrar matriz original
21     Escribir 'Matriz original:'
22     Para i ← 1 Hasta N Hacer
23         Para j ← 1 Hasta N Hacer
24             Escribir Sin Saltar matriz[i,j], ' '
25         FinPara
26         Escribir ''
27     FinPara
28
29     // Solicitar filas a intercambiar
30     Escribir 'Ingrese las filas a intercambiar (i y j, entre 1 y ', N, '): '
31     Leer fila_i, fila_j
32
33     // Validar filas
34     Si fila_i < 1 O fila_i > N O fila_j < 1 O fila_j > N Entonces
35         Escribir 'Error: Las filas deben estar entre 1 y ', N
36     Sino
37         // Intercambiar filas
38         Para k ← 1 Hasta N Hacer
39             aux ← matriz[fila_i, k]
40             matriz[fila_i, k] ← matriz[fila_j, k]
41             matriz[fila_j, k] ← aux
42         FinPara
43
44         // Mostrar matriz modificada
45         Escribir 'Matriz con filas ', fila_i, ' y ', fila_j, ' intercambiadas:'
46         Para i ← 1 Hasta N Hacer
47             Para j ← 1 Hasta N Hacer
48                 Escribir Sin Saltar matriz[i,j], ' '
49             FinPara
50             Escribir ''
51         FinPara
52     FinSi
53 FinSi
54 FinAlgoritmo
55
```

- Diagrama de Flujo





- Código en C

```
#include <stdio.h>

int main() {
    int N, i, j, fila_i, fila_j, aux;

    int matriz[10][10]; // Matriz máxima 10x10

    // Solicitar tamaño de la matriz
    printf("Ingrese el tamaño N de la matriz cuadrada (N x N, máximo 10): ");
    scanf("%d", &N);

    // Validar tamaño
    if (N <= 0 || N > 10) {
        printf("Tamaño inválido. Debe ser entre 1 y 10.\n");
        return 1; // Salir con error
    }

    // Llenar matriz con valores secuenciales
    for (i = 0; i < N; i++) {
        for (j = 0; j < N; j++) {
            matriz[i][j] = i * N + j + 1;
        }
    }

    // Mostrar matriz original
    printf("Matriz original:\n");
    for (i = 0; i < N; i++) {
        for (j = 0; j < N; j++) {
            printf("%3d ", matriz[i][j]);
        }
        printf("\n");
    }

    // Solicitar filas a intercambiar
    printf("Ingrese las filas a intercambiar (i y j, entre 1 y %d): ", N);
    scanf("%d %d", &fila_i, &fila_j);

    // Validar filas (restar 1 para índices base 0)
    if (fila_i < 1 || fila_i > N || fila_j < 1 || fila_j > N) {
        printf("Error: Las filas deben estar entre 1 y %d.\n", N);
        return 1;
    }
    fila_i--; // Ajustar a índice base 0
    fila_j--;

    // Intercambiar filas
    for (j = 0; j < N; j++) {
        aux = matriz[fila_i][j];
        matriz[fila_i][j] = matriz[fila_j][j];
        matriz[fila_j][j] = aux;
    }

    // Mostrar matriz modificada
    printf("Matriz con filas %d y %d intercambiadas:\n", fila_i + 1, fila_j + 1);
    for (i = 0; i < N; i++) {
        for (j = 0; j < N; j++) {
            printf("%3d ", matriz[i][j]);
        }
        printf("\n");
    }
}
```

Problema 2.2.3 Máximo de una fila.

Escriba un programa que lea una matriz de N filas y N columnas de valores enteros. A continuación, el programa debe pedir el número de una fila y mostrar por pantalla el valor de la mayor componente de esa fila.

Tal como ya se discutió en el problema 1.7, la dificultad de calcular el máximo valor de un vector (en este caso un vector fila de una matriz) reside en decidir qué valor inicial se le da a la variable que va a almacenar el máximo (max). Imagine que se asume que todos los números del vector son positivos y se inicializa $\text{max} = -1000$. Se procede entonces a comparar este valor con todas las componentes del vector y, si alguna es mayor, se actualiza el valor de max con el valor de esa componente. Podría ocurrir, sin embargo...



- Tabla de Objetos

OBJETO	NOMBRE	VARIABLE	TIPO
M1	N, fila, i, j, maximo	variable	Entero
M2	matriz	variable	Entero

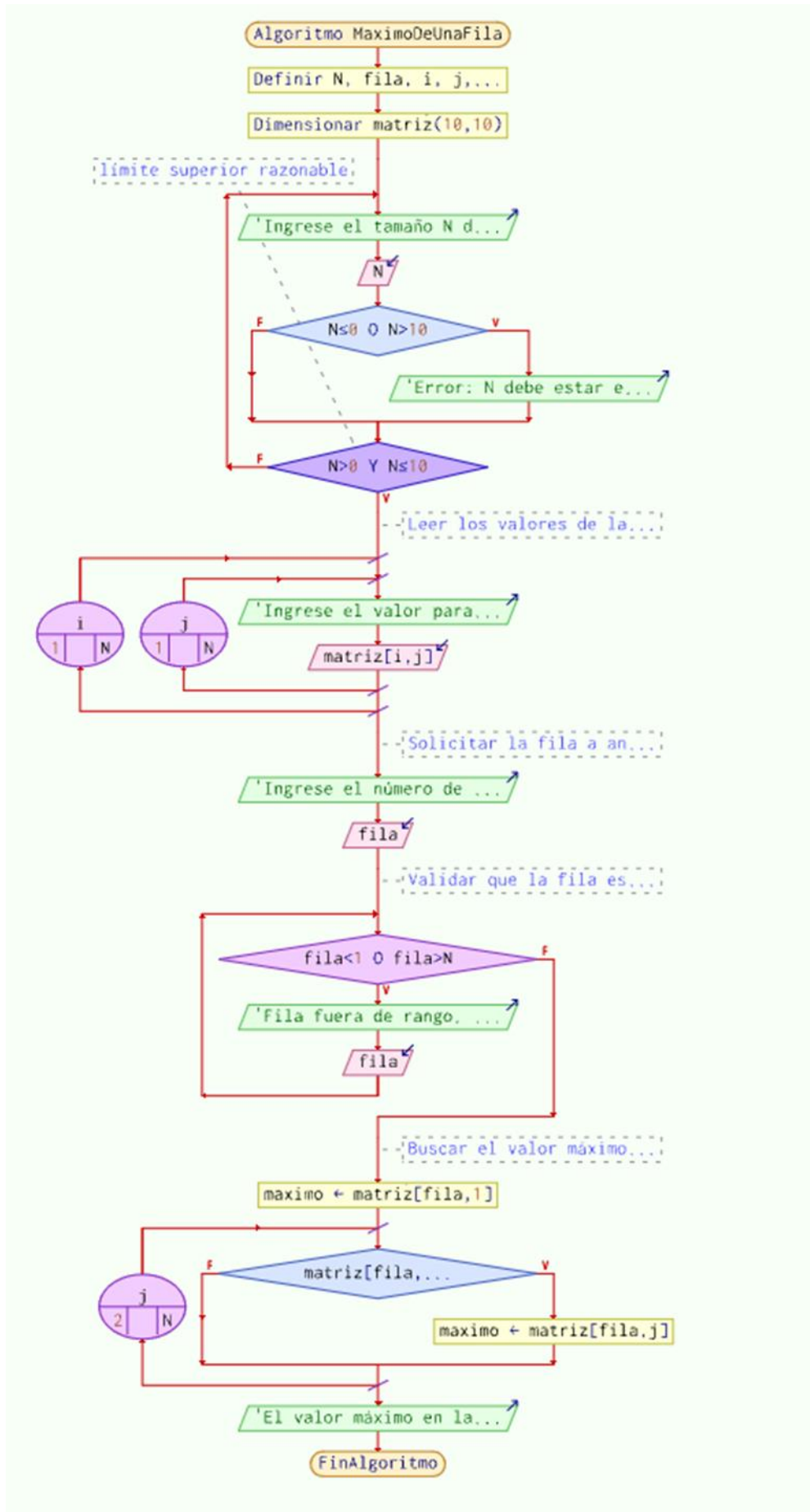
- Seudocódigo

```

1  Algoritmo MaximoDeUnaFila
2
3      Definir N, fila, i, j, maximo Como Entero
4      Dimension matriz[10, 10] // límite superior razonable
5
6      Repetir
7          Escribir "Ingrese el tamaño N de la matriz NxN (entre 1 y 10):"
8          Leer N
9
10         Si N ≤ 0 O N > 10 Entonces
11             Escribir "Error: N debe estar entre 1 y 10, intente nuevamente."
12         FinSi
13
14     Hasta Que N > 0 Y N ≤ 10
15
16     // Leer los valores de la matriz
17     Para i ← 1 Hasta N
18         Para j ← 1 Hasta N
19             Escribir "Ingrese el valor para la posición [", i, ", ", j, "]: "
20             Leer matriz[i, j]
21         FinPara
22     FinPara
23
24     // Solicitar la fila a analizar
25     Escribir "Ingrese el número de fila (entre 1 y ", N, ") para buscar el máximo:"
26     Leer fila
27
28     // Validar que la fila esté en el rango correcto
29     Mientras fila < 1 o fila > N
30         Escribir "Fila fuera de rango. Intente nuevamente:"
31         Leer fila
32     FinMientras
33
34     // Buscar el valor máximo en esa fila
35     maximo ← matriz[fila, 1]
36     Para j ← 2 Hasta N
37         Si matriz[fila, j] > maximo Entonces
38             maximo ← matriz[fila, j]
39         FinSi
40     FinPara
41
42     Escribir "El valor máximo en la fila ", fila, " es: ", maximo
43
44 FinAlgoritmo

```


- Diagrama de Flujo





Requisitos Funcionales

- Ingreso del tamaño de la matriz

El programa debe solicitar al usuario un valor entero N que represente el tamaño de una matriz cuadrada $N \times N$.

- Ingreso de los elementos de la matriz

El programa debe solicitar al usuario que ingrese un número entero para cada una de las posiciones $[i, j]$ de la matriz de tamaño $N \times N$.

- Selección de la fila a analizar

El programa debe solicitar al usuario que ingrese un número de fila dentro del rango de 1 a N . El programa debe validar que el número de fila ingresado esté dentro del rango permitido.

El programa debe volver a solicitar el número de fila si este está fuera del rango válido.

- Cálculo del valor máximo en una fila

El programa debe recorrer los elementos de la fila seleccionada por el usuario. El programa debe identificar y almacenar el valor máximo presente en esa fila.

- Visualización del resultado

El programa debe mostrar en pantalla el valor máximo encontrado en la fila seleccionada.

Código en C



```
1  #include <stdio.h>
2
3  int main() {
4      int N, fila, i, j, maximo;
5      int matriz[10][10];
6
7      // Validar N
8      do {
9          printf("Ingrese el tamaño N de la matriz NxN (entre 1 y 10): ");
10         scanf("%d", &N);
11
12         if (N <= 0 || N > 10) {
13             printf("Error: N debe estar entre 1 y 10. Intente nuevamente.\n");
14         }
15     } while (N <= 0 || N > 10);
16
17     // Ingreso de valores de la matriz
18     for (i = 0; i < N; i++) {
19         for (j = 0; j < N; j++) {
20             printf("Ingrese componente de la matriz [%d][%d]: ", i, j);
21             scanf("%d", &matriz[i][j]);
22         }
23     }
24
25     // Pedir número de fila
26     printf("Ingrese el número de fila (entre 1 y %d) para buscar el máximo: ", N);
27     scanf("%d", &fila);
28
29     // Validar fila
30     while (fila < 1 || fila > N) {
31         printf("Fila fuera de rango. Intente nuevamente: ");
32         scanf("%d", &fila);
33     }
34
35     // Ajustar índice de fila (porque el usuario usa 1 a N, pero C usa 0 a N-1)
36     fila = fila - 1;
37
38     // Buscar el máximo en esa fila
39     maximo = matriz[fila][0];
40     for (j = 1; j < N; j++) {
41         if (matriz[fila][j] > maximo) {
42             maximo = matriz[fila][j];
43         }
44     }
45
46     // Mostrar resultado
47     printf("El valor máximo en la fila %d es: %d\n", fila + 1, maximo);
48
49     return 0;
50 }
51
52
```

Problema 2.1.5 Vector de factoriales.

Problema 2.2.2 Escritura de matriz en sentido inverso. Dada una matriz de $N \times N$ elementos, realice un algoritmo que recorra la por filas desde la última a la primera y cada fila en sentido inverso, y de la columna a la primera, de modo que se vaya mostrando cada elemento.

La solución a este problema consiste en recorrer la matriz invirtiendo el sentido habitual de los bucles. Observe cómo, en este caso, los bucles de filas y columnas las variables i y j comienzan en la última fila/columna de la matriz. La condición de permanencia en los bucles es ahora $i \geq 1$ y $j \geq 1$ (en C $i \geq 0$ y $j \geq 0$) variables se decrementan en cada iteración.

A continuación se muestra el diagrama de flujo de la solución en la figura 2004 como la tabla de objetos y codificación en C3.



Tabla de Objetos

OBJETO	NOMBRE	VARIABLE	TIPO
M1	N, i, j	Variable	Entero
M2	matriz	Variable	Entero

Requisitos funcionales

RF1: El sistema debe permitir al usuario ingresar el tamaño N de la matriz cuadrada ($N \times N$).

RF2: El sistema debe permitir al usuario ingresar los elementos de la matriz manualmente o generar una matriz con valores predefinidos para pruebas.

RF3: El sistema debe recorrer la matriz desde la última fila a la primera (es decir, de $i = N-1$ a $i = 0$).

RF4: El sistema debe recorrer cada fila desde la última columna a la primera (es decir, de $j = N-1$ a $j = 0$).

RF5: El sistema debe mostrar en pantalla cada elemento de la matriz en el orden inverso especificado.

RF6: El sistema debe mantener el formato de impresión para que se comprenda la estructura de la matriz (opcional: separadores, saltos de línea por fila).

RF7: El sistema debe validar que el tamaño de la matriz sea un número entero positivo.

- **Seudocódigo**



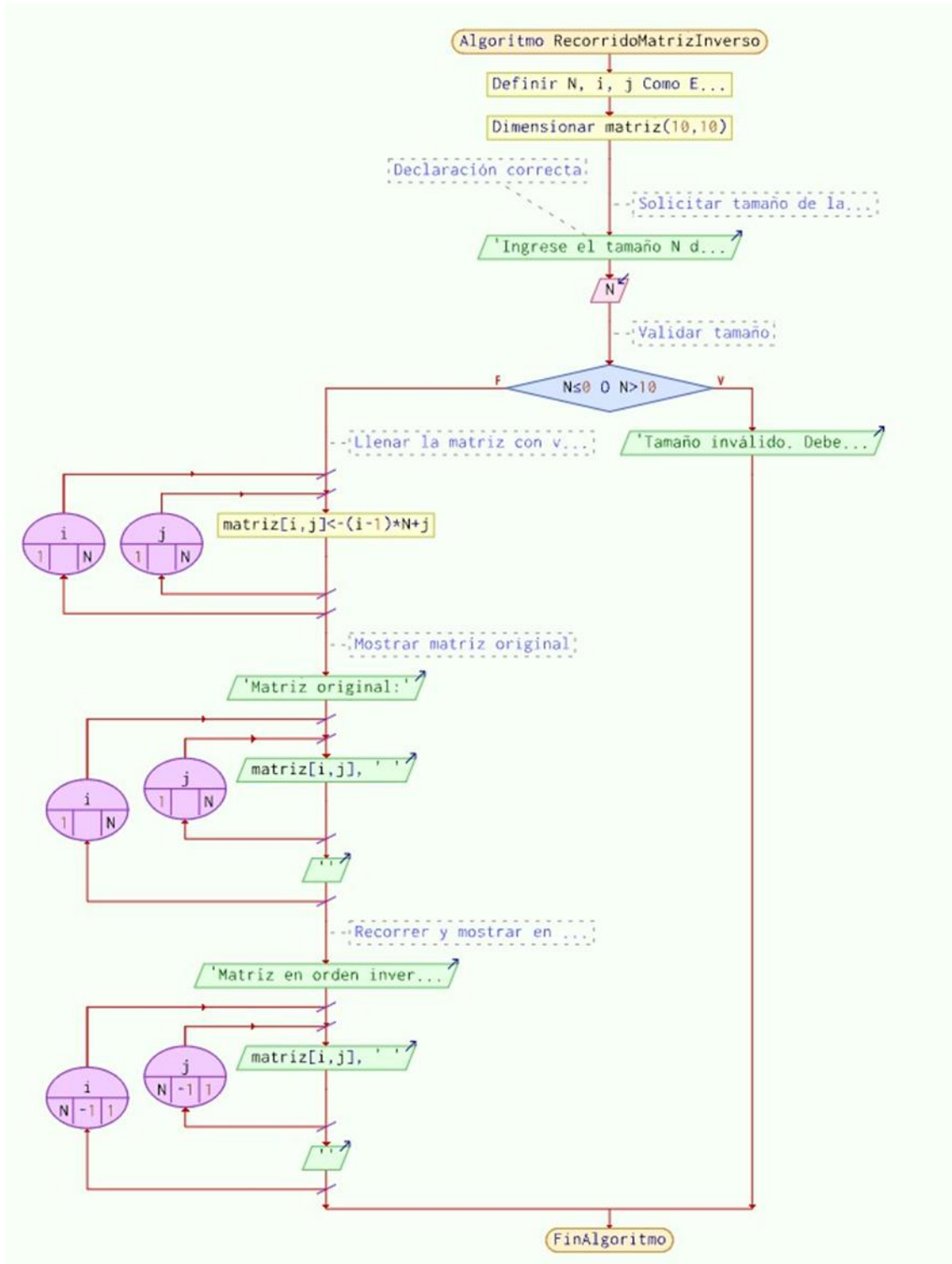
ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



```
1  Algoritmo RecorridoMatrizInverso
2      Definir N, i, j Como Entero
3      Dimension matriz[10,10] // Declaración correcta
4
5      // Solicitar tamaño de la matriz
6      Escribir "Ingrese el tamaño N de la matriz cuadrada (N x N, máximo 10): "
7      Leer N
8
9      // Validar tamaño
10     Si N ≤ 0 O N > 10 Entonces
11         Escribir "Tamaño inválido. Debe ser entre 1 y 10."
12     Sino
13         // Llenar la matriz con valores de ejemplo
14         Para i ← 1 Hasta N Hacer
15             Para j ← 1 Hasta N Hacer
16                 matriz[i,j] ← (i - 1) * N + j
17             FinPara
18         FinPara
19
20         // Mostrar matriz original
21         Escribir "Matriz original:"
22         Para i ← 1 Hasta N Hacer
23             Para j ← 1 Hasta N Hacer
24                 Escribir Sin Saltar matriz[i,j], " "
25             FinPara
26             Escribir ""
27         FinPara
28
29         // Recorrer y mostrar en orden inverso
30         Escribir "Matriz en orden inverso:"
31         Para i ← N Hasta 1 Con Paso -1 Hacer
32             Para j ← N Hasta 1 Con Paso -1 Hacer
33                 Escribir Sin Saltar matriz[i,j], " "
34             FinPara
35             Escribir ""
36         FinPara
37     FinSi
38 FinAlgoritmo
39
```

- Diagrama de Flujo





Código en C

```
#include <stdio.h>

int main() {
    int N, i, j;
    int matriz[10][10]; // Tamaño máximo de 10x10

    // Solicitar tamaño de la matriz
    printf("Ingrese el tamaño N de la matriz cuadrada (N x N, máximo 10): ");
    scanf("%d", &N);

    // Validar tamaño
    if (N <= 0 || N > 10) {
        printf("Tamaño inválido. Debe ser entre 1 y 10.\n");
        return 1; // Terminar el programa con error
    }

    // Llenar la matriz con valores de ejemplo
    for (i = 0; i < N; i++) {
        for (j = 0; j < N; j++) {
            matriz[i][j] = i * N + j + 1;
        }
    }

    // Mostrar matriz original
    printf("Matriz original:\n");
    for (i = 0; i < N; i++) {
        for (j = 0; j < N; j++) {
            printf("%3d ", matriz[i][j]);
        }
        printf("\n");
    }

    // Mostrar matriz en orden inverso
    printf("Matriz en orden inverso:\n");
    for (i = N - 1; i >= 0; i--) {
        for (j = N - 1; j >= 0; j--) {
            printf("%3d ", matriz[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA



