Universidad de las Fuerzas Armadas ESPE

Departamento: Ciencias de la computación

Carrera: Ingeniería en Electricidad y Automatización

U2T1: Vectores y funciones

---

1. Información General

- **Asignatura:** Fundamentos de Programación
- **Apellidos y nombres de los estudiantes:** Astudillo Adriana, Muñoz Sarahi, Nero Alan
- **NRC:** 20823
- **Fecha de realización:** 01/06/2025

Problema 2.1.3 Vector con término general dado. Sea la sucesión:

$$k=k^2+3,$$

desarrolle un programa que lea el número n de componentes que se quieren calcular de la sucesión y almacenarlas en un vector vec, tal que vec(i) = ı. Se mostraré vector por pantalla. Puede asumir que n será siempre menor o igual a 100.
Para calcular las componentes del vector se utilizará una iteración con un índice amando valores de 1 a n en diagrama de flujo (de 0 an-1 en C). A la vez, se ira calculando la componente (vec(i) = i^2+3) y mostrándola por pantalla.

- **Tabla de objetos**

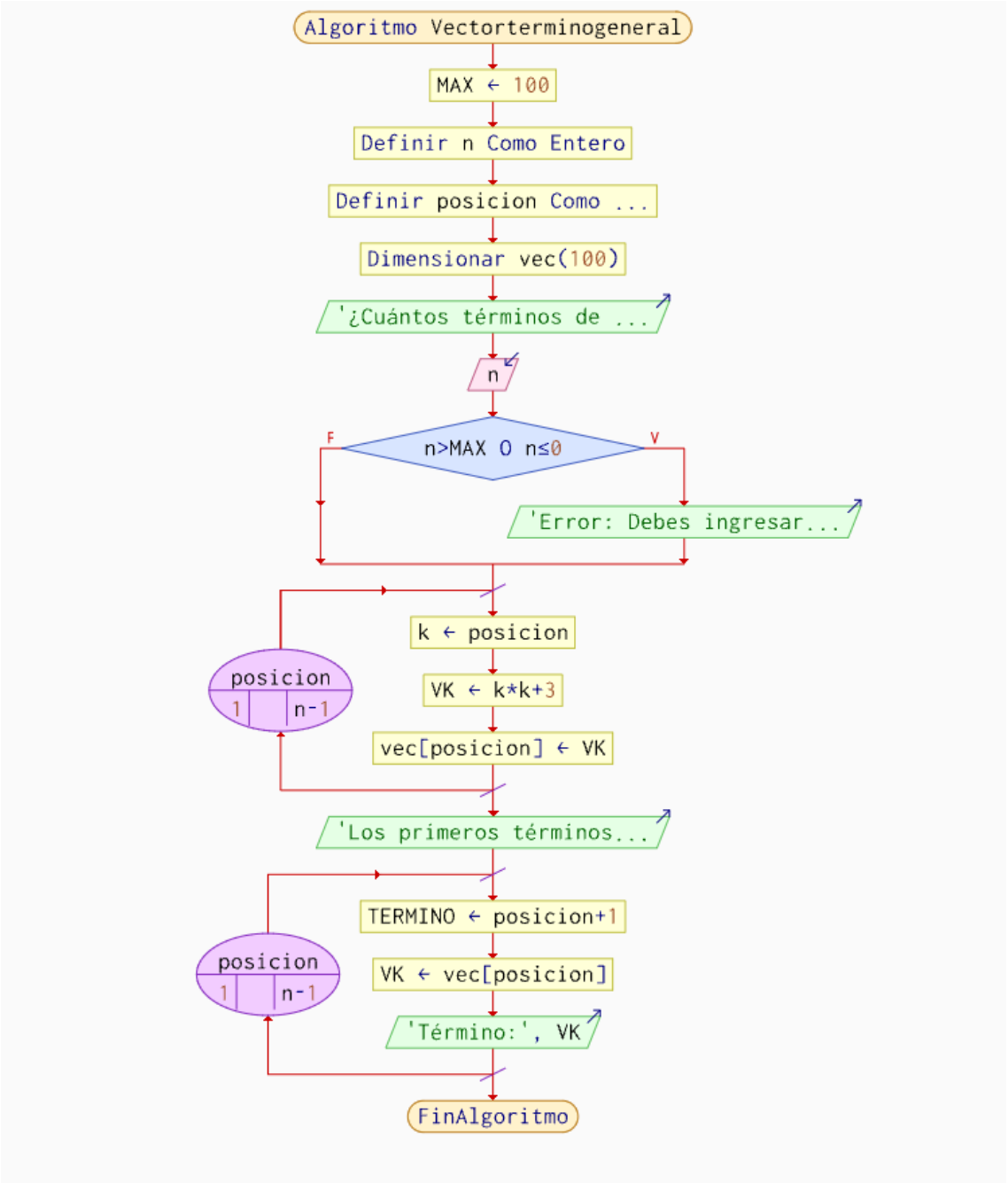
objetos	nombre	valor	Tipo
Maximo	MAX	constante	Entero
Numero de sucesión	n	variable	Entero
posición	posición	constante	entero
Numero de posición en la sucesión	K	Constante	Entero

Termino	Termino	constante	entero

- **Psint**

```
Algoritmo Vectorterminogeneral
    MAX=100
    Definir n como entero
    Definir posicion como entero
    Dimensionar vec[100]
    Escribir "¿Cuántos términos de la sucesión deseas generar?"
    LEER n
    SI n > MAX O n ≤ 0 ENTONCES
        MOSTRAR "Error: Debes ingresar un número entre 1 y 100"
    FIN SI
    PARA posición desde 1 hasta n-1 HACER
        k = posicion
        VK = k * k + 3
        vec[posición]= VK
    FIN PARA
    Escribir "Los primeros términos de la sucesión son:",VK

    PARA posición desde 1 hasta n-1 HACER
        término = posición + 1
        VK = vec[posición]
        MOSTRAR "Término:", VK
    FIN PARA
    FinAlgoritmo
```

- **DF**



- **Código c**

```
#include <stdio.h>
#define MAX 100  // Tamaño máximo del vector

int main() {
```

```
    int n;              // Número de términos a generar
    int vec[MAX];       // Vector para almacenar los términos
    int k, VK;          // Variables auxiliares

    // Solicitar el número de términos
    printf("¿Cuántos términos de la sucesión deseas generar? (1-%d): ", MAX);
    scanf("%d", &n);

    // Validar la entrada
    if (n > MAX || n <= 0) {
        printf("Error: Debes ingresar un número entre 1 y %d\n", MAX);
        return 1;  // Terminar con código de error
    }

    // Generar los términos de la sucesión v_k = k² + 3
    for (int posicion = 0; posicion < n; posicion++) {
        k = posicion + 1;  // Ajuste porque la sucesión comienza en k=1
        VK = k * k + 3;
        vec[posicion] = VK;
    }

    // Mostrar los resultados
    printf("\nLos primeros %d términos de la sucesión son:\n", n);
    for (int posicion = 0; posicion < n; posicion++) {
        printf("Término %d: %d\n", posicion + 1, vec[posicion]);
    }

    return 0;
}
```

Problema 2.1.4 - Comprobar si dos valores pertenecen a un vector:

Realice un algoritmo que lea dos números enteros por teclado y determine si ambos valores forman parte de un vector de enteros previamente definido de dimensión n*n*.
La solución se basa en dos variables bandera, que representan si uno de los números está en el vector. Se inicializan ambas a 0, y se recorre el vector comparando cada componente con los valores leídos por el teclado. Si alguno coincide, se cambia el valor de la bandera asociada a 1. Al finalizar, si ambas valen 1, el resultado será positivo.

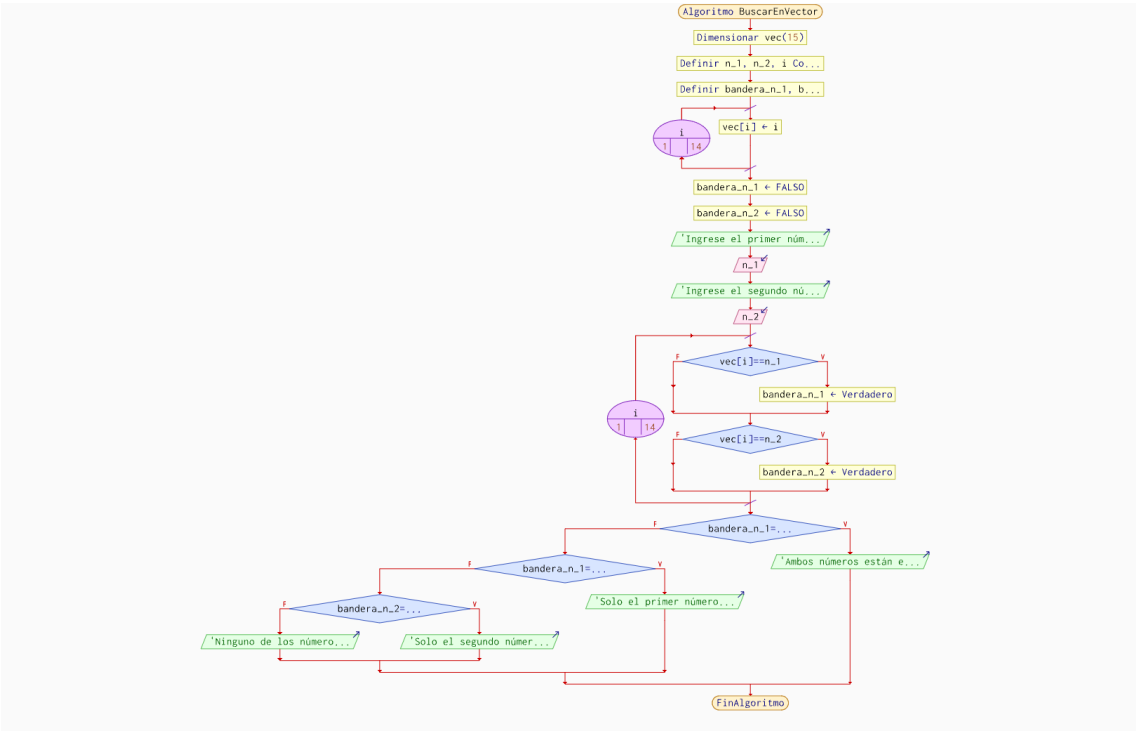
- **Tabla de objetos**

Objeto	nombre	valor	Tipo
Numero 1	n_1	Variable	Entero
Número 2	n_2	Variable	Entero
Vector	vec	constante	Entero
Bandera n_1	bandera n_1	Variable	Entero
Banderan_2	bandera n_2	Variable	Entero

- **Psint**

```
Algoritmo BuscarEnVector
Dimensionar vec[15]
Definir n_1, n_2, i como entero
Definir bandera_n_1, bandera_n_2 como logico
Para i ← 1 Hasta 14 Hacer
    vec[i] ← i
FinPara
    bandera_n_1 = FALSO
    bandera_n_2 = FALSO
    Escribir "Ingrese el primer número a buscar: "
    Leer n_1
    Escribir "Ingrese el segundo número a buscar: "
    Leer n_2
    Para i desde 1 hasta 14 Hacer
        Si vec[i] == n_1 Entonces
            bandera_n_1= Verdadero
        FinSi

        Si vec[i] == n_2 Entonces
            bandera_n_2 = Verdadero
        FinSi
    FinPara
    Si bandera_n_1 == Verdadero Y bandera_n_2 == Verdadero Entonces
        Escribir "Ambos números están en el vector"
    Sino Si bandera_n_1 == Verdadero Entonces
            Escribir "Solo el primer número está en el vector"
        Sino Si bandera_n_2 == Verdadero Entonces
                Escribir "Solo el segundo número está en el vector"
            Sino
                Escribir "Ninguno de los números está en el vector"
            FinSi
        FinSi
    FinSi

FinAlgoritmo

```

- **DF**



- **Código c**
- #include <stdio.h>
-
- int main() {
- // Definición del vector de 15 elementos
- int vector[15] = {4, 8, 15, 16, 23, 42, 3, 9, 12, 5, 7, 11, 13, 17, 19};
- int num1, num2;
- int bandera_num1 = 0, bandera_num2 = 0;
-
- // Solicitar números al usuario
- printf("Ingrese el primer número a buscar: ");
- scanf("%d", &num1);
- printf("Ingrese el segundo número a buscar: ");
- scanf("%d", &num2);
-
- // Buscar los números en el vector
- for(int i = 0; i < 15; i++) {
- if(vector[i] == num1) {
- bandera_num1 = 1;
- }

```
        if(vector[i] == num2) {
            bandera_num2 = 1;
        }
    }

    // Mostrar resultados
    if(bandera_num1 && bandera_num2) {
        printf("Ambos números %d y %d están en el vector.\n", num1, num2);
    } else if(bandera_num1) {
        printf("Solo el número %d está en el vector.\n", num1);
    } else if(bandera_num2) {
        printf("Solo el número %d está en el vector.\n", num2);
    } else {
        printf("Ninguno de los números %d y %d está en el vector.\n", num1, num2);
    }

    return 0;
}
```

Problema 2.1.5 - Vector de factoriales:

Dado un vector Vec, que contiene los primeros 15 números naturales, calcule un vector fact con sus factoriales y mostrarlo por pantalla.
Nota: ¿Por qué a partir del número 12 no funciona correctamente el cálculo del factorial? ¿Qué se podría hacer para evitarlo?
El algoritmo consiste en dos bucles anidados, uno externo, que da valores a la variable i*i* entre 1 y 15, y otro interno, que calcula el factorial de la componente i*i*-ésima. El algoritmo deja de funcionar para el valor 12 porque en la codificación en C se han empleado variables de tipo entero. El valor de 13! excede la capacidad de almacenamiento de una variable entera, y el cálculo se corrompe. Una posible solución es emplear variables de tipo *unsigned long int* (entero largo sin signo), que incrementan la capacidad de almacenamiento.

- **Tabla de objetos**

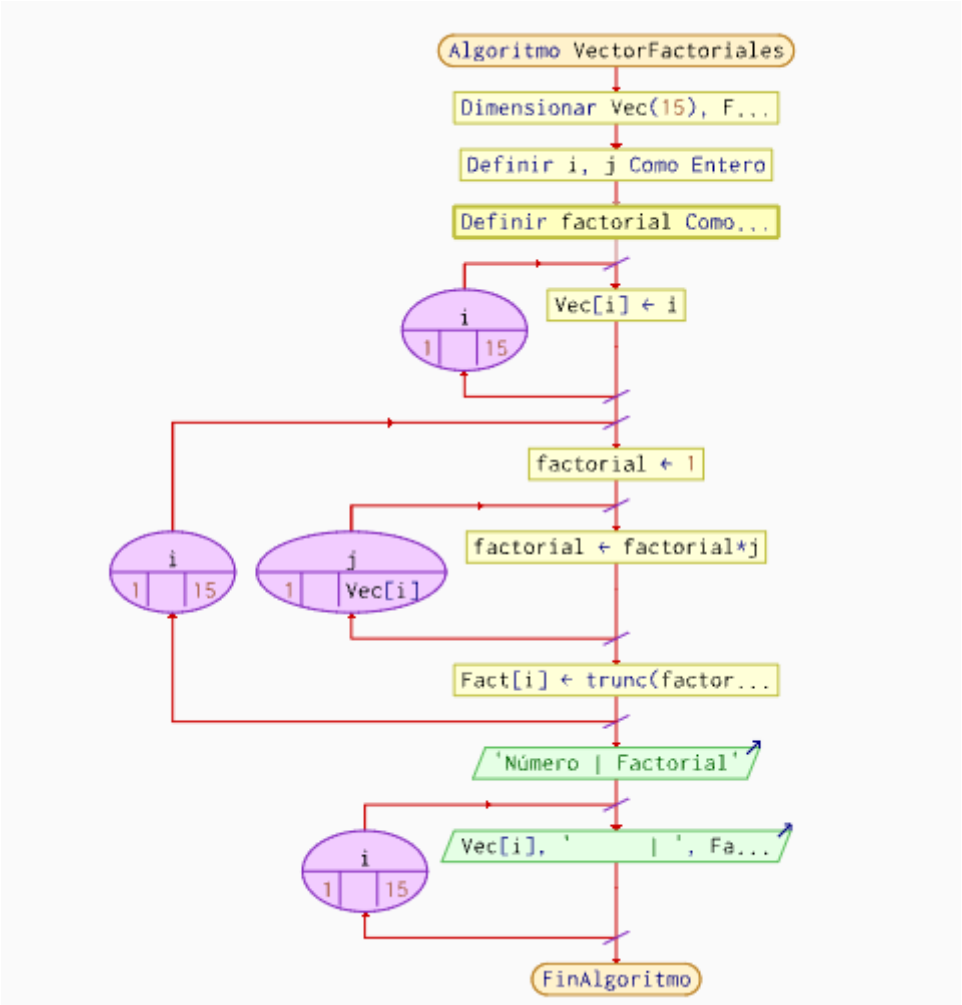
Objeto	Nombre	Valor	Tipo
Vector original	Vec	[1, 2, 3, ..., 15] Constante	Enteros
Vector factorial	Fact	[1!, 2!, 3!, ..., 15!] Variable	Enteros

Contador externo	i	1 a 15 Variable	Entero
Contador interno	j	1 a Vec[i] Variable	Entero
Valor factorial	factorial	Temporal para cálculo (1, 2, 6,..) Variable	Entero
Límite máximo	n	15 Constante	Entero

- **Psint**

```
Algoritmo VectorFactoriales
    Dimension Vec[15], Fact[15]
    Definir i, j Como Entero
    Definir factorial Como Real
    Para i ← 1 Hasta 15 Hacer
        Vec[i] ← i
    FinPara

    Para i ← 1 Hasta 15 Hacer
        factorial ← 1

        Para j ← 1 Hasta Vec[i] Hacer
            factorial ← factorial * j
        FinPara

        Fact[i] ← trunc(factorial)
    FinPara

    Escribir "Número | Factorial"
    Para i ← 1 Hasta 15 Hacer
        Escribir Vec[i], "      | ", Fact[i]
    FinPara
FinAlgoritmo
```

- **DF**



- **Código c**

```
#include <stdio.h>
#include <limits.h> // Para INT_MAX

int main() {
    const int n = 15;
    int Vec[n];
    unsigned long long Fact[n];

    for(int i = 0; i < n; i++) {
        Vec[i] = i + 1;
    }

    for(int i = 0; i < n; i++) {
        Fact[i] = 1;
```

```
    for(int j = 1; j <= Vec[i]; j++) {
        Fact[i] *= j;
    }
}

printf("Número | Factorial\n");
printf("------------------\n");
for(int i = 0; i < n; i++) {
    printf("%6d | %20llu\n", Vec[i], Fact[i]);
}

return 0;
}
```

Problema 2.1.6 - Ordenación de un vector:
Desarrolle un programa que ordene un vector de 10 componentes de mayor a menor valor. Asuma que el vector está ya leído y almacenado en memoria.
Existen varios métodos de ordenación. Se expondrán dos:

- **Ordenación iterativa:** Esta solución emplea dos bucles anidados para comparar cada elemento del vector con los que le siguen, intercambiando las parejas de elementos fuera de orden. Por ejemplo, para ordenar un vector de n*n* elementos numerados de 0 a n−1*n*−1 (en DF de 1 a n*n*), se comienza comparando el elemento 0 con los que le siguen, es decir, las parejas de elementos (0-1), (0-2), ...(0-n-1). Si alguna pareja presenta un orden inverso al deseado, se intercambian las posiciones de sus elementos en el vector. Tras comparar la última pareja, se tiene la garantía de que el elemento de mayor valor está en la posición 0. El procedimiento se repite a continuación con el elemento 1, esto es, se estudian las parejas (1-2), (1-3), ...(1-n-1), intercambiando aquellas fuera de orden. El procedimiento se repite hasta llegar al elemento n−2*n*−2, que debe ser comparado con el último, es decir, n−2*n*−2 con n−1*n*−1.

➢ **Tabla de objetos.**

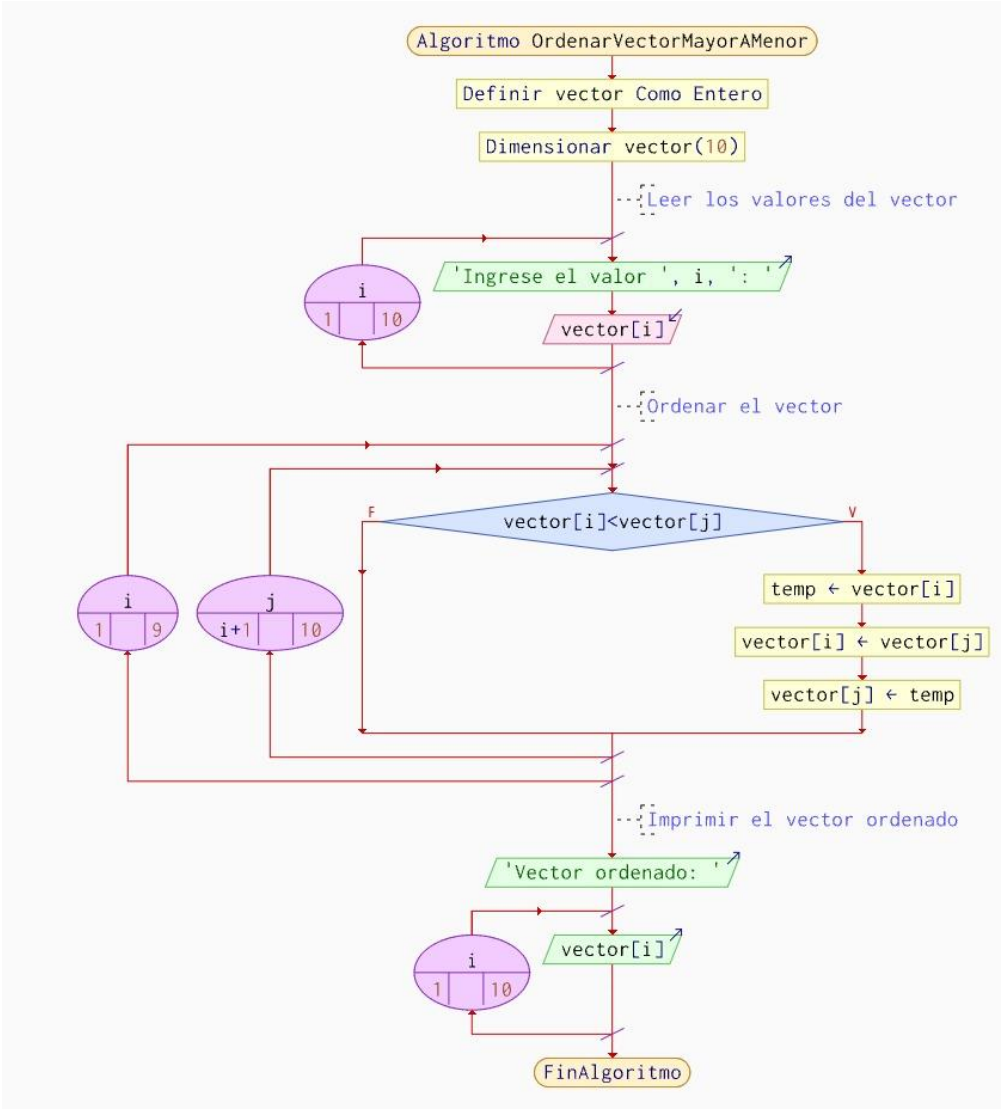
Objetos	Nombre	Valor	Tipo
Vector	**vector**	**constante**	**Entero**
Valor i	**i**	**Variable**	**Entero**
Valor j	**j**	**Variable**	**Entero**

➢ **PsInt.**

```
Algoritmo OrdenarVectorMayorAMenor
    Definir vector Como Entero
    Dimension vector[10]

    // Leer los valores del vector
```

```
    Para i <- 1 Hasta 10 Hacer
        Escribir "Ingrese el valor ", i, ": "
        Leer vector[i]
    Fin Para

    // Ordenar el vector
    Para i <- 1 Hasta 9 Hacer
        Para j <- i + 1 Hasta 10 Hacer
            Si vector[i] < vector[j] Entonces
                temp <- vector[i]
                vector[i] <- vector[j]
                vector[j] <- temp
            Fin Si
        Fin Para
    Fin Para

    // Imprimir el vector ordenado
    Escribir "Vector ordenado: "
    Para i <- 1 Hasta 10 Hacer
        Escribir vector[i]
    Fin Para
FinAlgoritmo.
```

- **DF**



- **Código en c.**

```
#include <stdio.h>
int main() {
    int i;
    int j;
    int temp;
    int vector[10];

    /* Leer los valores del vector */
    for (i = 0; i < 10; i++) {
        printf("Ingrese el valor %d: \n", i + 1);
        scanf("%d", &vector[i]);
```

```
    }

    /* Ordenar el vector */
    for (i = 0; i < 9; i++) {
        for (j = i + 1; j < 10; j++) {
            if (vector[i] < vector[j]) {
                temp = vector[i];
                vector[i] = vector[j];
                vector[j] = temp;
            }
        }
    }
    /* Imprimir el vector ordenado */
    printf("Vector ordenado: \n");
    for (i = 0; i < 10; i++) {
        printf("%d\n", vector[i]);
    }
    return 0;
}
```