

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

PROGRAMACIÓN

FACULTAD DE INGENIERÍA

Práctica 5: Programas

Integrantes:

Rodríguez Ruíz Stefanny,

Mendoza Frías Luis

Fernando, Martínez Barras

Alexis, Mandujano Jiménez

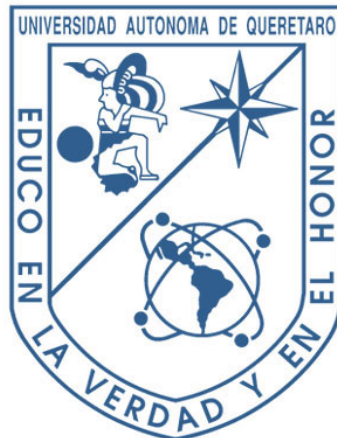
Daniel Cruz

Ingeniería en Nanotecnología

Profesor: José de Jesús

Santana Ramírez

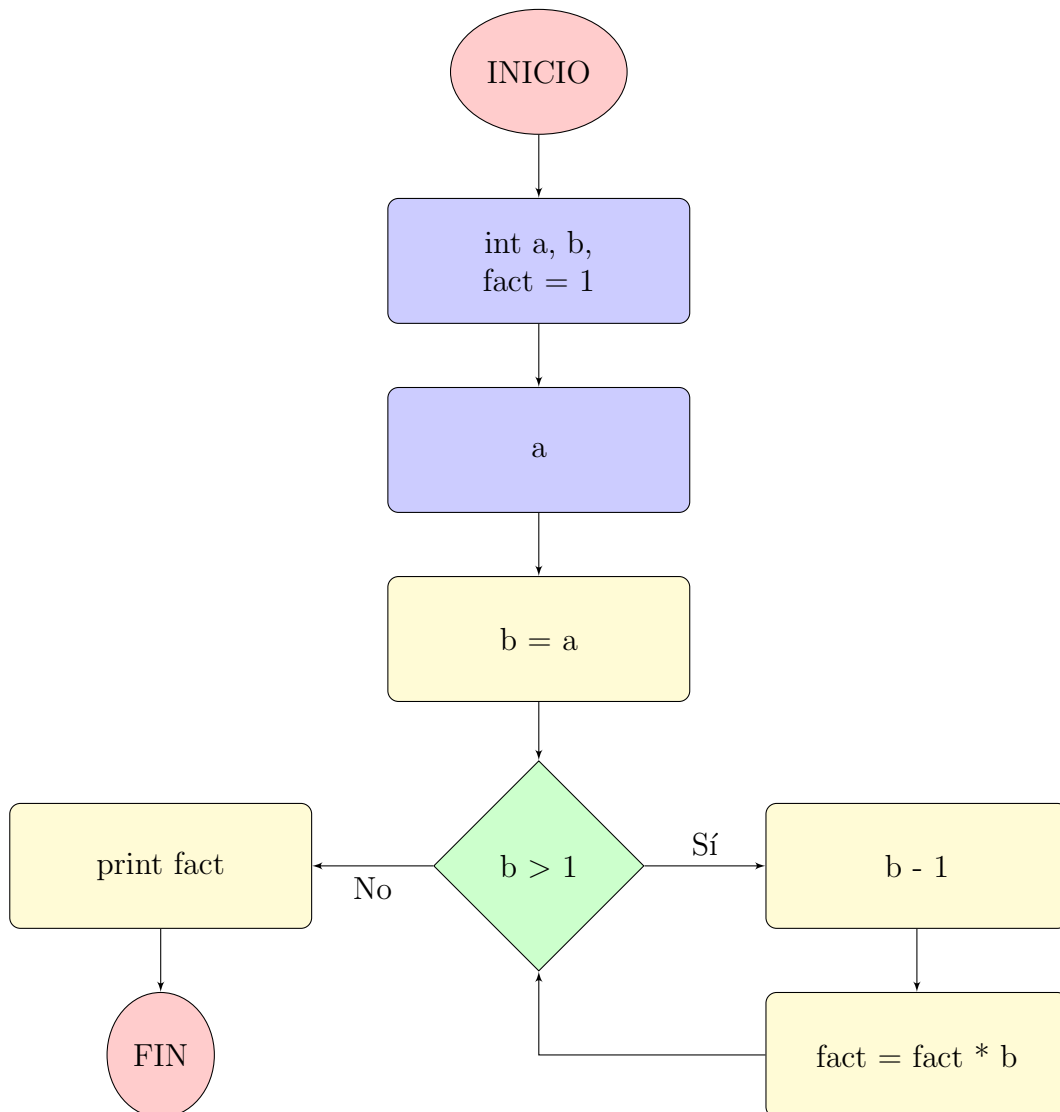
Abril 11, 2019



1 Función factorial

La función factorial es una fórmula matemática representada por el signo de exclamación “!”. En la fórmula Factorial se deben multiplicar todos los números enteros y positivos que hay entre el número que aparece en la fórmula y el número 1.

1.1 Diagrama de flujo



1.2 Código

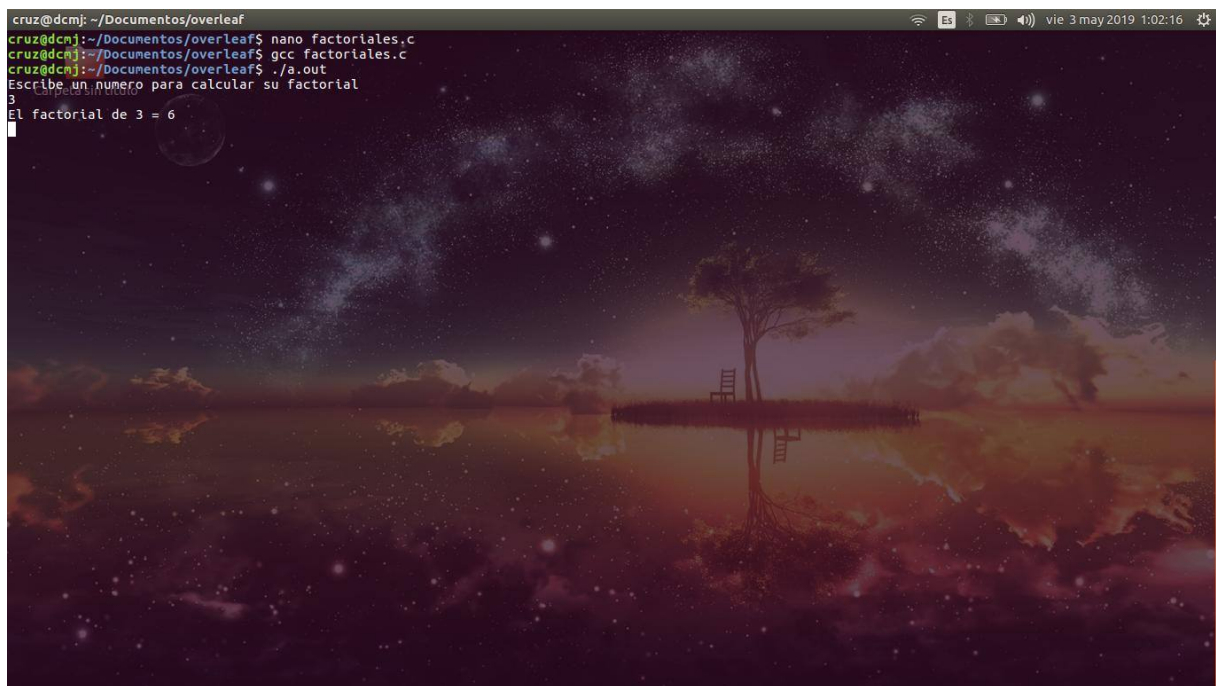
```
#include <stdio.h>
int main()
{
    int a, b, fact = 1;
    printf("Escribe un numero para calcular su factorial\n");
    scanf("%d", &a);
```

```

    for (b = a; b > 1; b--){
        fact = fact * b;
    }
    printf("El factorial de %d = %d\n", a, fact);
    getchar();
    getchar();
    return 0;
}

```

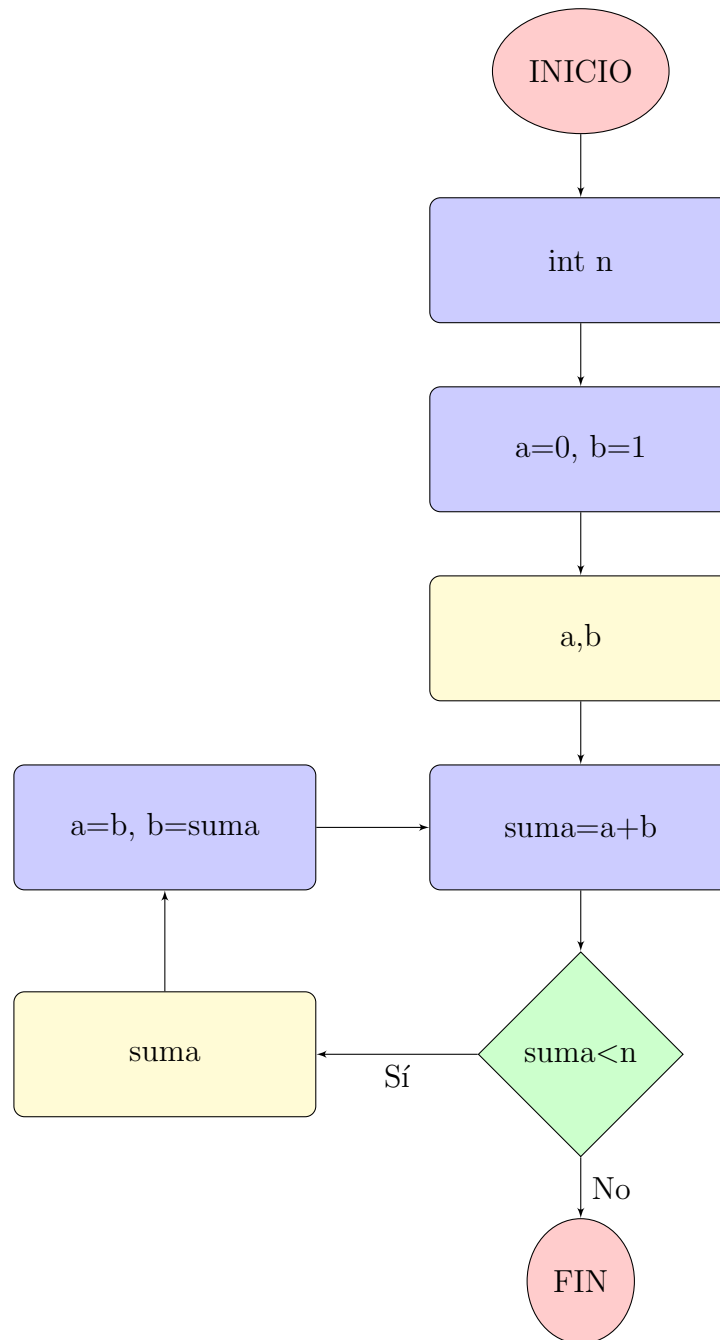
1.3 Terminal



2 Función Fibonacci

La sucesión de Fibonacci es una sucesión definida por recurrencia. Esto significa que para calcular un término de la sucesión se necesitan los términos que le preceden. Se proporcionan los dos primeros términos: $a_0=0$ y $a_1=1$. Los siguientes se calculan con la siguiente fórmula: $a_{n+1} = a_{n-1} + a_n, n \geq 1$

2.1 Diagrama de flujo



2.2 Código

```
#include <stdio.h>
#include <stdlib.h>
int fibonacci(int n)
{
    if (n>2)
        return fibonacci(n-1) + fibonacci(n-2);
}
```

```

    else if (n==2)
        return 1;
    else if (n==1)
        return 1;
    else if (n==0)
        return 0;
}
int main(void)
{
    int num;
    for (num=0; num<=25; num++)
    {
        printf("%d\n", fibonacci(num));
    }
    system("PAUSE");
    return 0;
}

```

2.3 Terminal

```

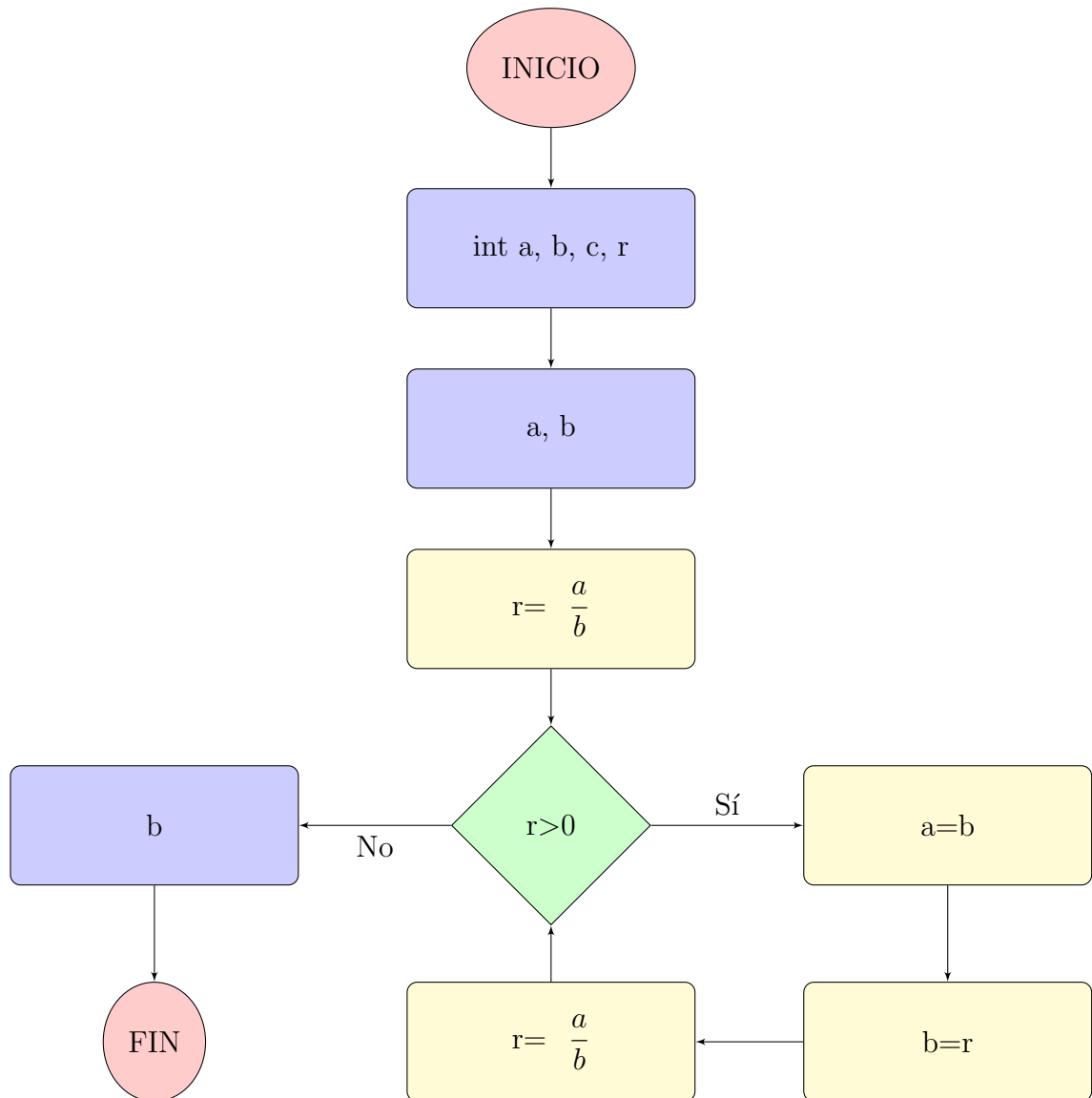
cruz@dcnj: ~/Documentos/overleaf
cruz@dcnj:~/Documentos/overleaf$ nano fabionacci
cruz@dcnj:~/Documentos/overleaf$ nano fabionacci.c
cruz@dcnj:~/Documentos/overleaf$ gcc fabionacci.c
cruz@dcnj:~/Documentos/overleaf$ ./a.out
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
10946
17711
28657
46368
75025
sh: 1: PAUSE: not found
cruz@dcnj:~/Documentos/overleaf$

```

3 Máximo común divisor

El máximo común divisor de dos números a y b es el número más grande que divide a a y divide a b .

3.1 Diagrama de flujo



3.2 Código

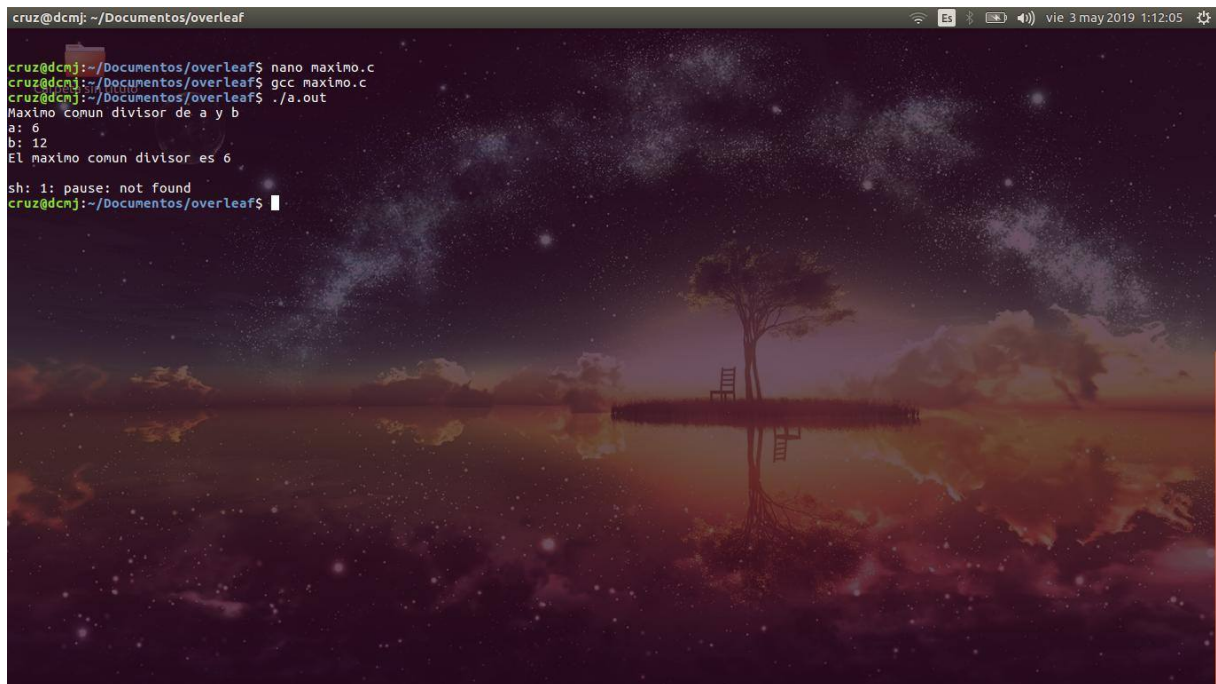
```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int a, b, c, r;
    printf("Maximo comun divisor de a y b\n");
    printf("a: ");
    scanf("%d", &a);
    printf("b: ");
    scanf("%d", &b);
    r=a%b;
    while(r>0)
```

```

{
    fflush(stdin);
    a=b;
    b=r;
    r=a%b;
}
printf("El maximo comun divisor es %d", b);
printf("\n\n");
system("pause");
}

```

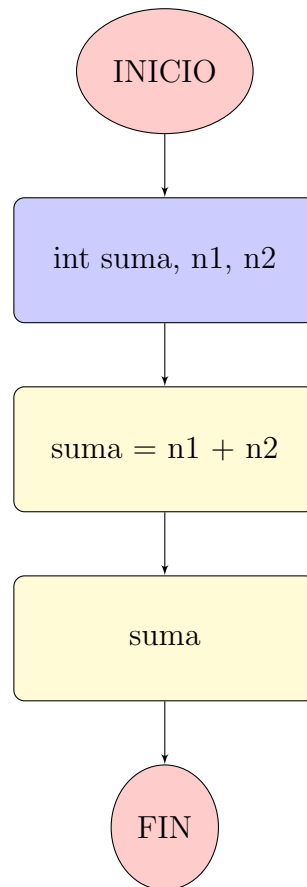
3.3 Terminal



4 Suma de dos enteros

Es la operación matemática que consiste en combinar o añadir dos números o más para obtener una cantidad total. Por otro lado, repetir la acción de sumar varias veces es la forma más común de contar.

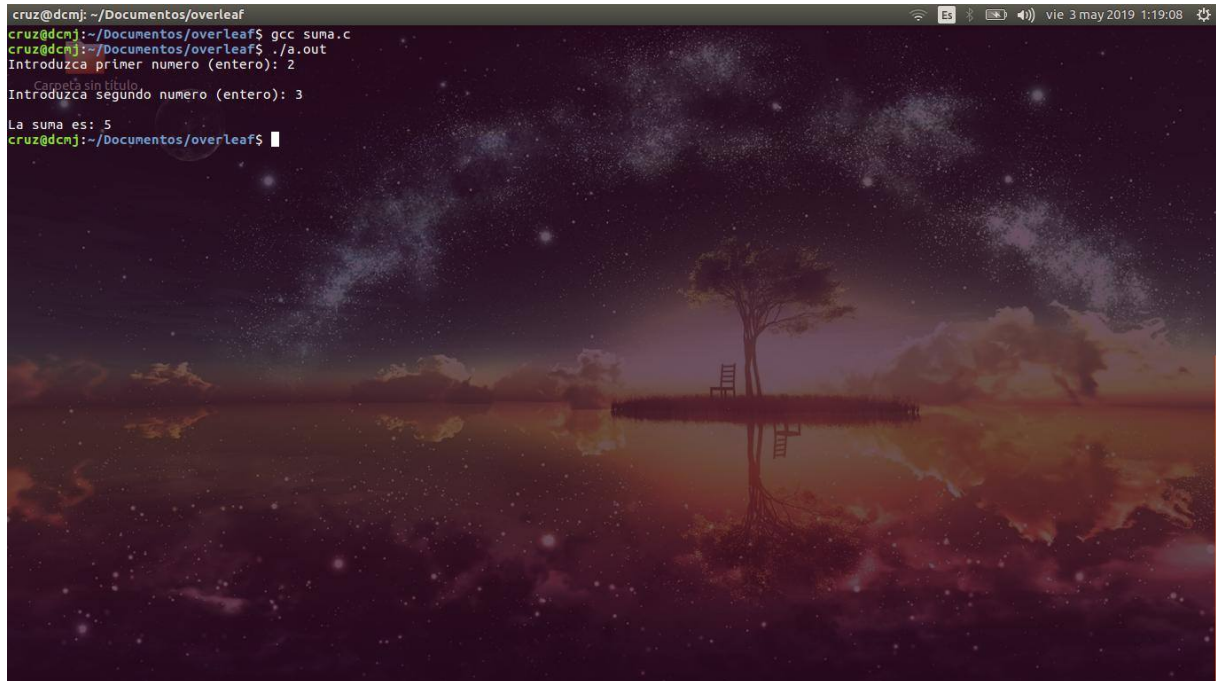
4.1 Diagrama de flujo



4.2 Código

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n1, n2, suma;
    printf( "Introduzca primer numero (entero): " );
    scanf( "%d", &n1 );
    printf( "\nIntroduzca segundo numero (entero): " );
    scanf( "%d", &n2 );
    suma = n1 + n2;
    printf( "\nLa suma es: %d", suma );
    getch(); /* Pausa */
    return 0;
}
```


4.3 Terminal



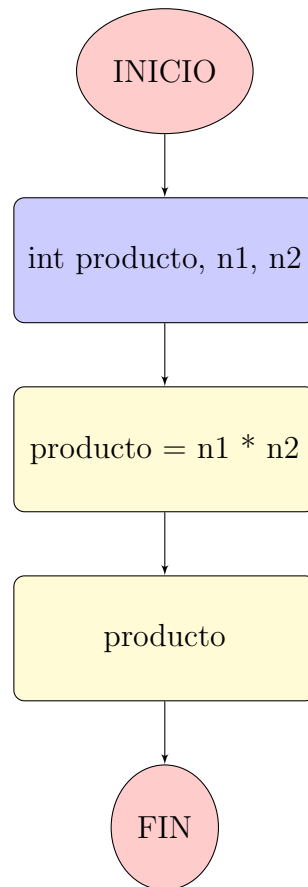
```
cruz@dcnj: ~/Documentos/overleaf
cruz@dcnj:~/Documentos/overleaf$ gcc suma.c
cruz@dcnj:~/Documentos/overleaf$ ./a.out
Introduzca primer número (entero): 2
Introduzca segundo número (entero): 3
La suma es: 5
cruz@dcnj:~/Documentos/overleaf$
```

The terminal window is overlaid on a background image of a lone tree on a small, floating island in a vast, starry space. The sky is filled with a dense field of stars and a prominent band of the Milky Way galaxy. The island is a small, dark, circular patch of land with a single, leafy tree growing from its center. A small wooden chair is placed on the island, and a ladder extends from the tree's roots down into the surrounding space. The overall scene is surreal and evocative, suggesting a sense of isolation and contemplation in the cosmos.

5 Multiplicación de dos enteros

Es una operación aritmética de composición que consiste en sumar reiteradamente la primera cantidad tantas veces como indica la segunda. El resultado de la multiplicación de varios números se llama producto.

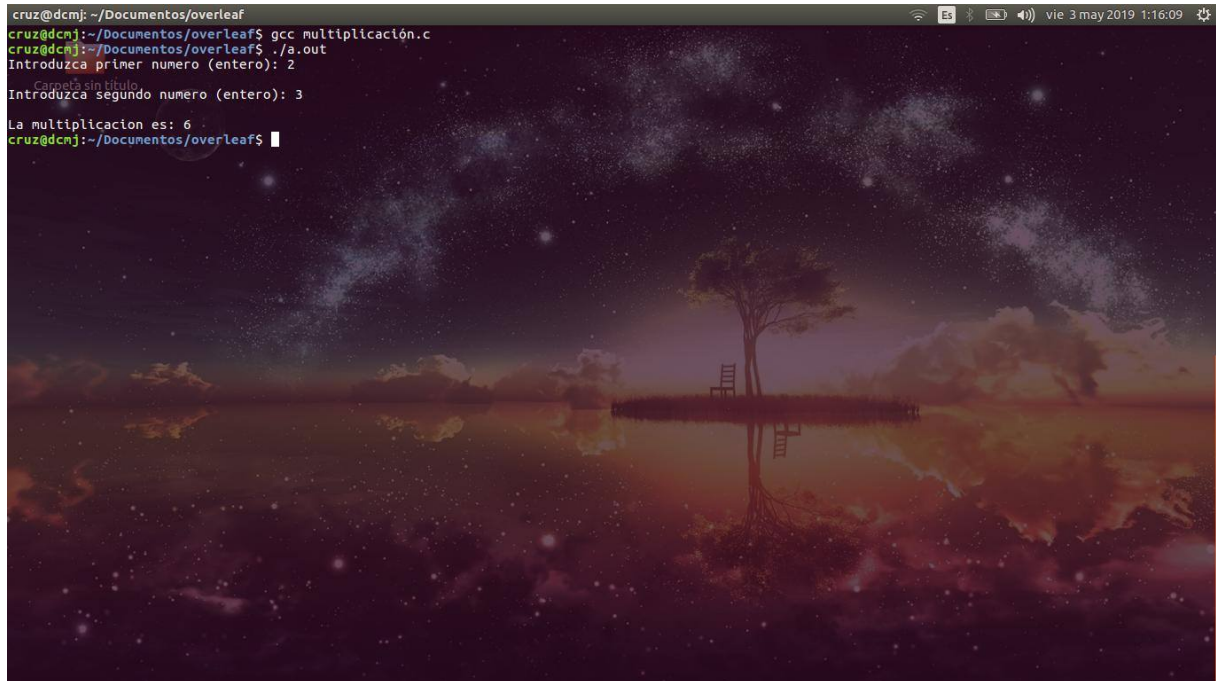
5.1 Diagrama de flujo



5.2 Código

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n1, n2, producto;
    printf( "Introduzca primer numero (entero): " );
    scanf( "%d", &n1 );
    printf( "\nIntroduzca segundo numero (entero): " );
    scanf( "%d", &n2 );
    producto = n1 * n2;
    printf( "\nLa multiplicacion es: %d", producto );
    getch(); /* Pausa */
    return 0;
}
```

5.3 Terminal



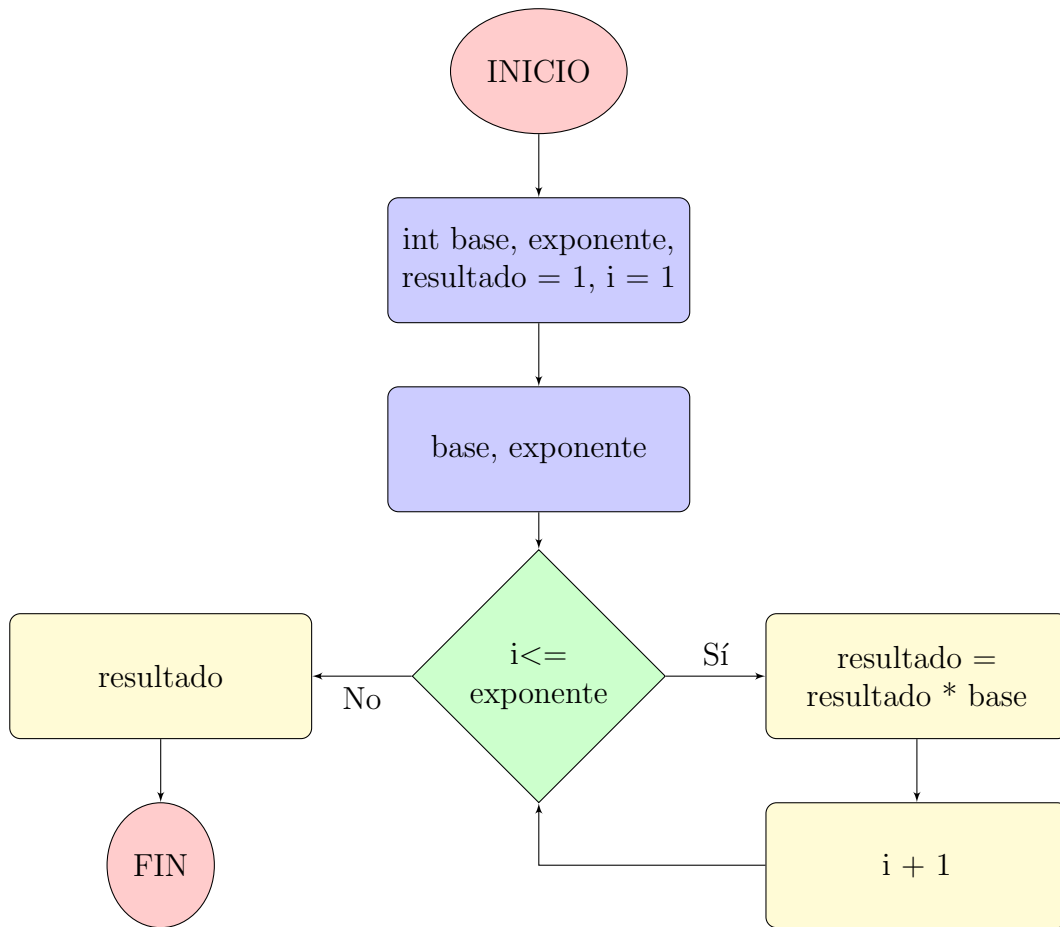
```
cruz@dcnj: ~/Documentos/overleaf
cruz@dcnj:~/Documentos/overleaf$ gcc multiplicación.c
cruz@dcnj:~/Documentos/overleaf$ ./a.out
Introduzca primer numero (entero): 2
Introduzca segundo numero (entero): 3
La multiplicacion es: 6
cruz@dcnj:~/Documentos/overleaf$
```

The terminal window shows the execution of a C program that calculates the product of two numbers. The user enters 2 and 3, and the program outputs 6. The background of the terminal is a dark, starry space scene with a small tree and a chair on a floating platform.

6 Como elevar un entero a una potencia entera

Las potencias son una manera abreviada de escribir una multiplicación formada por varios números iguales. Son muy útiles para simplificar multiplicaciones donde se repite el mismo número. Las potencias están formadas por la base y por el exponente. La base es el número que se está multiplicando varias veces y el exponente es el número de veces que se multiplica la base.

6.1 Diagrama de flujo

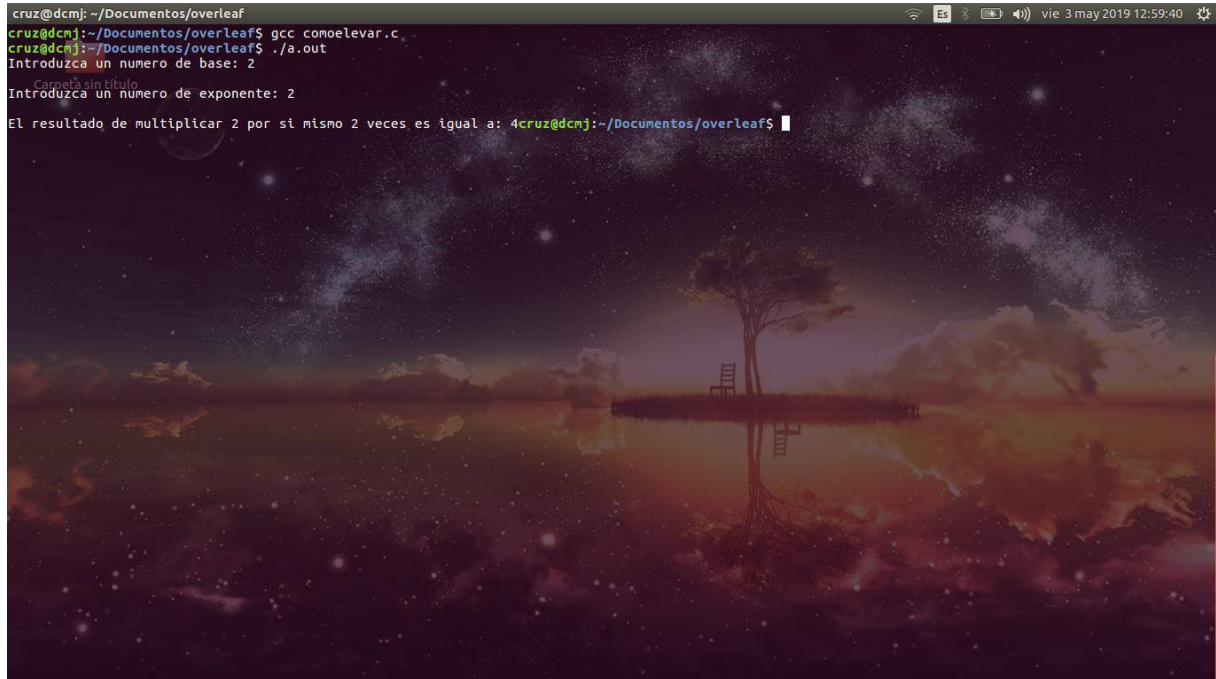


6.2 Código

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int base, exponente, resultado, i;
    i=1;
    resultado = 1;
    printf("Introduzca un numero de base: ");
    scanf("%d", &base);
    puts("");
    printf("Introduzca un numero de exponente: ");
    scanf("%d", &exponente);
    puts("");
    while(i<=exponente){
        resultado = resultado * base;
        i++;
    }
    printf("El resultado de multiplicar %d por si mismo %d veces es igual a: %d", base,exponente, resultado);
}
```

```
    return 0;  
}
```

6.3 Terminal

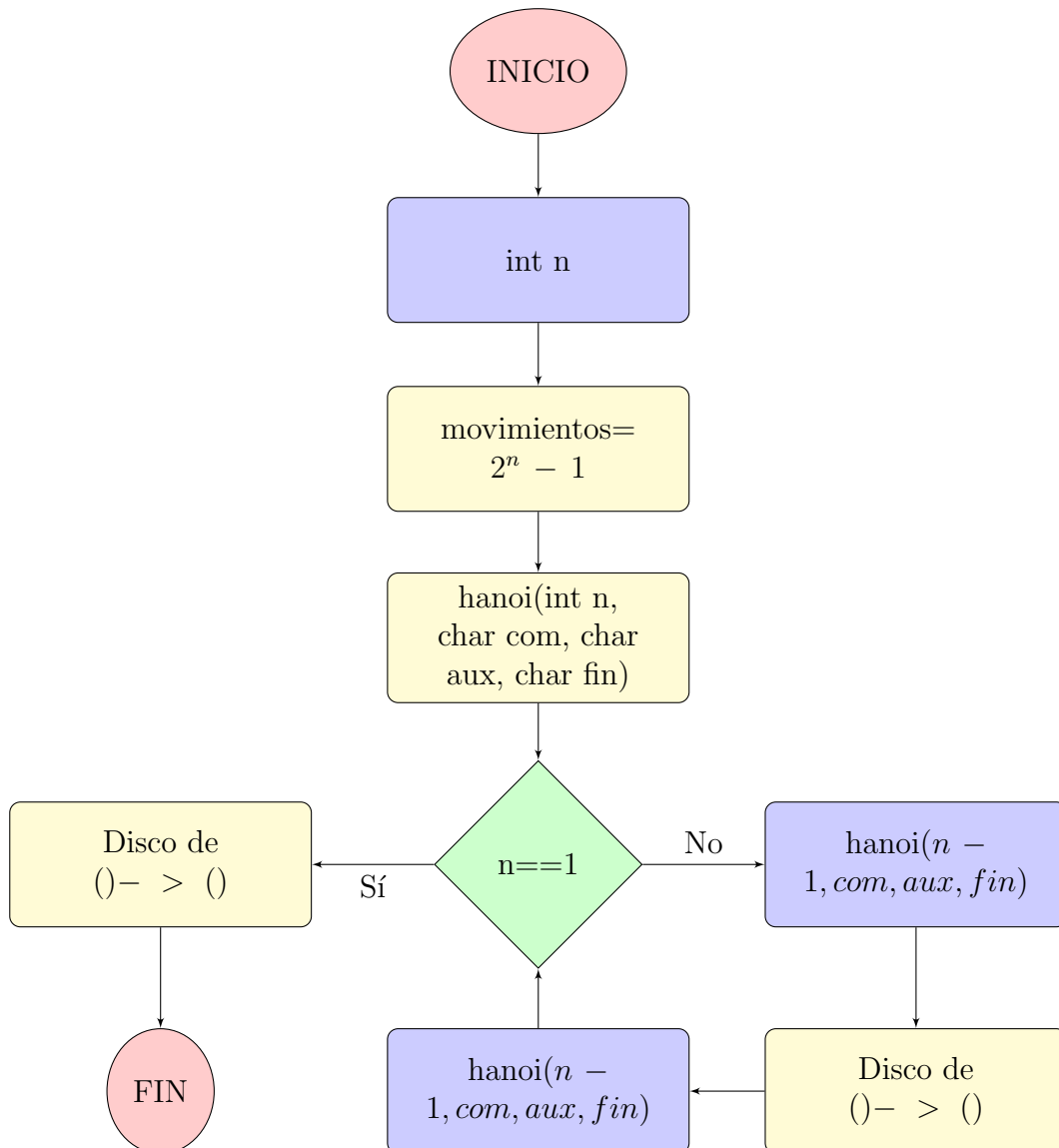


```
cruz@dcnj: ~/Documentos/overleaf  
cruz@dcnj:~/Documentos/overleaf$ gcc comoelevar.c  
cruz@dcnj:~/Documentos/overleaf$ ./a.out  
Introduzca un numero de base: 2  
Introduzca un numero de exponente: 2  
El resultado de multiplicar 2 por si mismo 2 veces es igual a: 4cruz@dcnj:~/Documentos/overleaf$
```

7 Torres de Hanoi

Te dan un conjunto de tres varillas y n discos, con cada disco de un tamaño diferente. Llamemos a las varillas A, B y C, y numeremos los discos desde 1, el disco más pequeño, hasta n , el disco más grande. Al principio, todos los n discos están en la varilla A, en orden de tamaño decreciente de la parte inferior a la parte superior, de modo que el disco n está en la parte inferior y el disco 1 está en la parte superior. La finalidad del juego es pasar todos los discos de la varilla A hasta la varilla C.

7.1 Diagrama de flujo



7.2 Código

```
#include <stdio.h>
#include <conio.h>
void hanoi(int n, int com, int aux, int fin);
int main(void){

    char com='A';
    char aux='B';
    char fin='C';
    int n;
    printf("\nNumero de discos: ");
    scanf("%d",&n);
    fflush(stdin);
```

```

printf("\n\nLos movimientos a realizar son: \n");
hanoi(n,com,aux,fin);
}
void hanoi(int n,int com, int aux, int fin){
if(n==1){
printf("%c->%c",com,fin);
}
else{
hanoi(n-1,com,fin,aux);
printf("\n%c->%c\n",com,fin);
hanoi(n-1,aux,com,fin);
}
}
}

```

7.3 Terminal

```

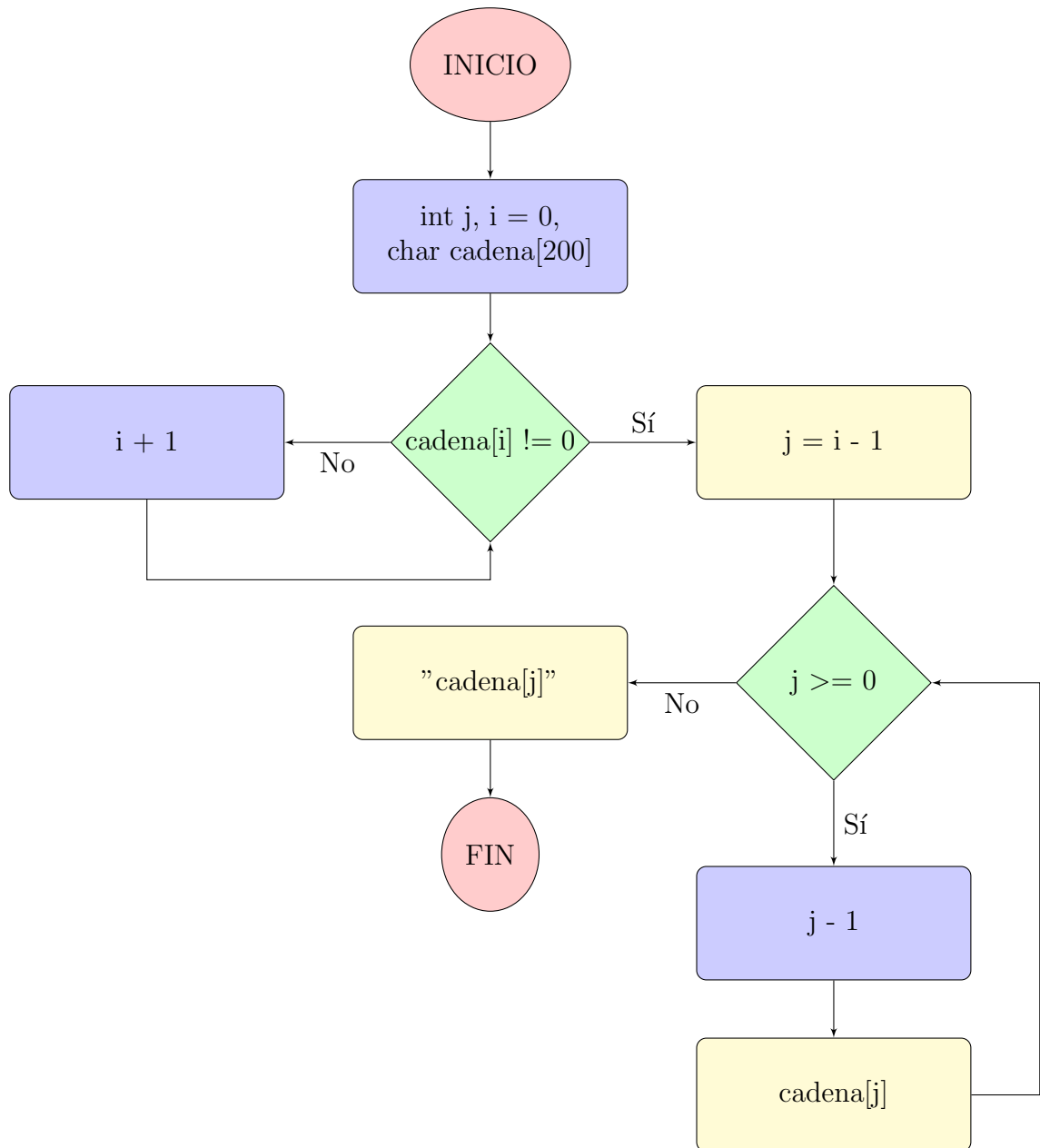
cruz@dcnj: ~/Documentos/overleaf
cruz@dcnj:~/Documentos/overleaf$ nano torres.c
cruz@dcnj:~/Documentos/overleaf$ gcc torres.c
cruz@dcnj:~/Documentos/overleaf$ ./a.out
Numero de discos: 3
Los movimientos a realizar son:
A->C
A->B
C->B
A->C
B->A
B->C
A->C
cruz@dcnj:~/Documentos/overleaf$

```

8 Impresión inversa de entrada desde el teclado

Las funciones declaradas en `string.h` se han hecho muy populares, por lo que están garantizadas para cualquier plataforma que soporte C. Sin embargo, existen algunos problemas de seguridad con estas funciones, como el desbordamiento de buffer (buffer overflow), que hacen que algunos programadores prefieran opciones más seguras frente a la portabilidad que estas funciones ofrecen. Además, las funciones para cadenas de caracteres sólo trabajan con conjuntos de caracteres ASCII o extensiones ASCII compatibles.

8.1 Diagrama de flujo



8.2 Código

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char cadena[200];
    int i = 0;
    int j;
    printf("Introduce una cadena de texto: ");
```



```

    gets(cadena);
    printf("\nLa cadena introducida es: %s\n\n", cadena);
    while(cadena[i]!='\0'){
        i++;
    }
    printf("La cadena al rev\x82s es: \t");
    for (j=i-1; j>=0; j--){
        printf("%c", cadena[j]);
    }
    puts("");
    return 0;
}

```

8.3 Terminal

