

Universidad de Guanajuato División de Ingenierías
Campus Irapuato Salamanca (DICIS)

Algoritmos y estructura de datos
Carlos Hugo García Capulín

Tarea No. 3
Reporte Uso de estrategias recursivas

Jair Chávez Islas
24/Agosto/2021

Problema

En este reporte, retomaremos el código de los 3 programas de los problemas vistos en clase y en el reporte pasado, los cuales son:

1. Calcular la suma de los primeros N números naturales.

Lo que queremos lograr con este programa es que, al poner cierto número, se sumen todos los números desde el primero hasta el número que ingresamos, por ejemplo, si ponemos un número 6, la sumatoria sería $1+2+3+4+5+6 = 21$.

2. Calcular el Factorial del número N .

La factorial de un número es la multiplicación de todos los números desde 1 hasta llegar al número escogido, por ejemplo, si queremos sacar la factorial de 6 sería

$$1 * 2 = 2$$

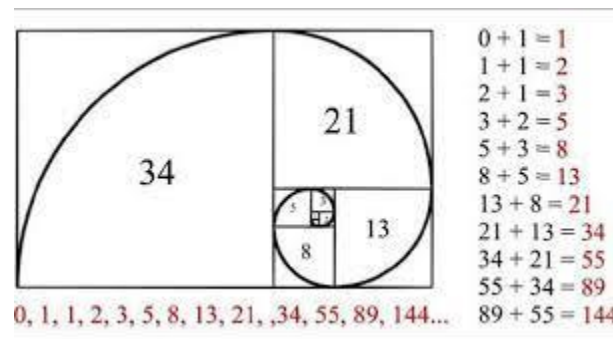
$$2 * 3 = 6$$

$$6 * 4 = 24$$

$$24 * 5 = 120$$

3. Calcular la secuencia de los primeros N números de la serie Fibonacci.

La serie Fibonacci es una secuencia de números en la cual, el siguiente número, es la suma de los últimos 2 números.



Aquí tenemos la serie Fibonacci, entonces tenemos que, por ejemplo, los primeros 6 números de la serie Fibonacci serían 0,1,1,2,3,5.

Solución implementada

Ahora tenemos la misión de hacer los mismos tres problemas, pero con la diferencia de que ahora usamos estrategias recursivas; y para esto ¿Qué es la recursividad?

La recursividad es una técnica usada en la programación en la que una función se llama a sí misma de forma repetida con el fin de obtener un resultado, siendo que cada acción repetida depende del resultado anterior; Para la recursividad, ocupamos que en la función en la que se vaya a aplicar esta técnica, haya un paso base que es donde se termina la función, de no ser así, se abriría un ciclo infinito.

Popularmente para explicar la recursividad se utiliza el problema de sacar el factorial de un número(Problema el cual también va implementado en este reporte).

Diagrama del primer problema usando recursividad

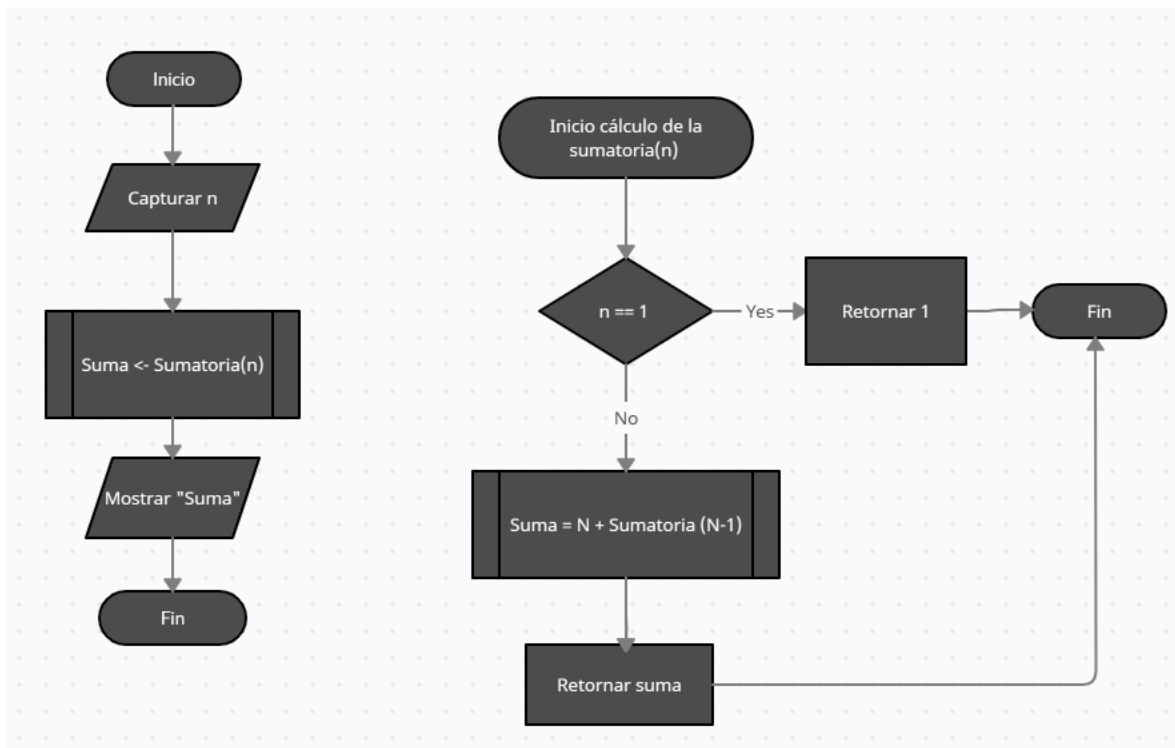


Diagrama del segundo problema usando recursividad

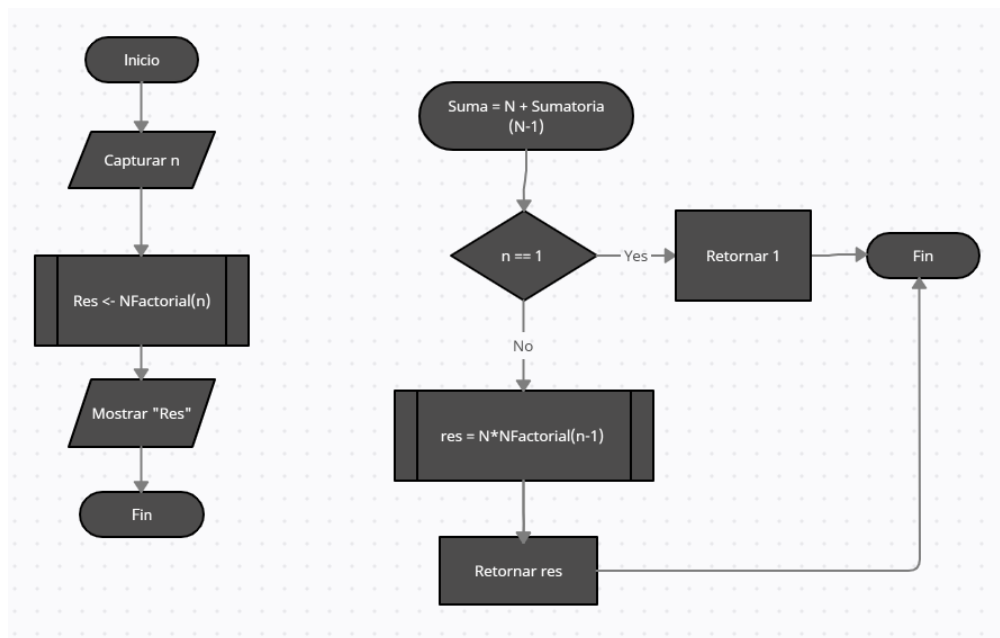
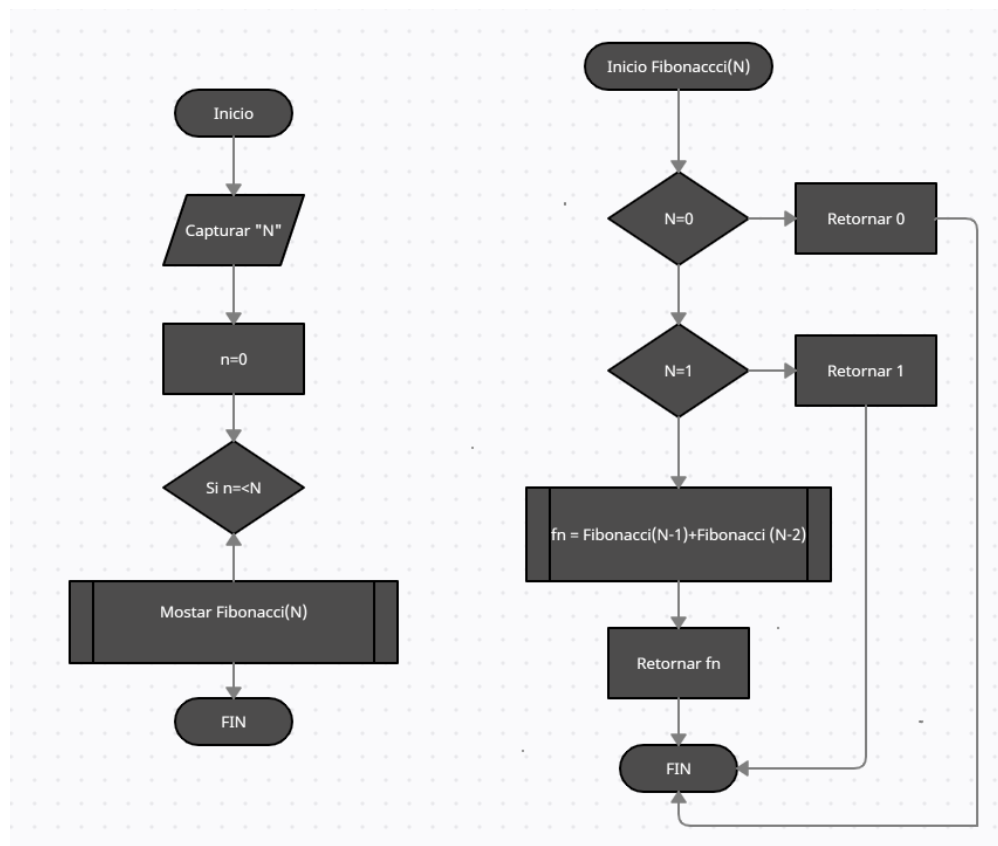


Diagrama del tercer problema usando recursividad



Código comentado del primer problema

```
p04.c x p05.c x p06.c
1 #include <stdio.h>//Agregamos la libreria stdio.h que contiene las funciones básicas del lenguaje de programación C.
2
3 unsigned int Sumatoria(unsigned int N);//Aqui Se ponen los prototipos de las funciones que vayamos utilizando.
4
5 int main();//Inicia función Principal.
6 {
7     unsigned int N,suma; //Declaracion de las variables.
8
9     printf("\nDigite un numero natural");//pedimos un numero mediante un texto que se muestra al usuario.
10    scanf("%u",&N);//guardamos el numero en la variable "N".
11
12    suma = Sumatoria(N);//Aplicamos la funcion "Sumatoria" al numero N y lo guardamos en suma.
13
14    printf("El resultado de la sumatoria de los primeros %i numeros es: %i",N,suma);//Imprimimos el resultado de la suma.
15    printf("\n");//Aqui solo se imprime un salto de linea.
16
17    return 0;//se retorna el valor de 0 para verificar que haya finalizado sin problemas.
18 }
19
20 unsigned int Sumatoria(unsigned int N)//Aqui comienza la funcion "Sumatoria".
21 {
22     unsigned int suma;//Aqui se declaran las o en este caso la variable local de la funcion.
23
24     if(N==1)//
25         return 1;//Para evitar un bucle infinito, implementamos el caso base, que nos dice que si N es igual a 1, retorna 1.
26
27     suma = N+Sumatoria(N-1);//Aqui llamamos a la misma funcion de donde estamos trabajando para seguir la recursividad.
28     return suma;//Al final de la suma, retornamos ese valor.
29 }
```

Código comentado del segundo problema

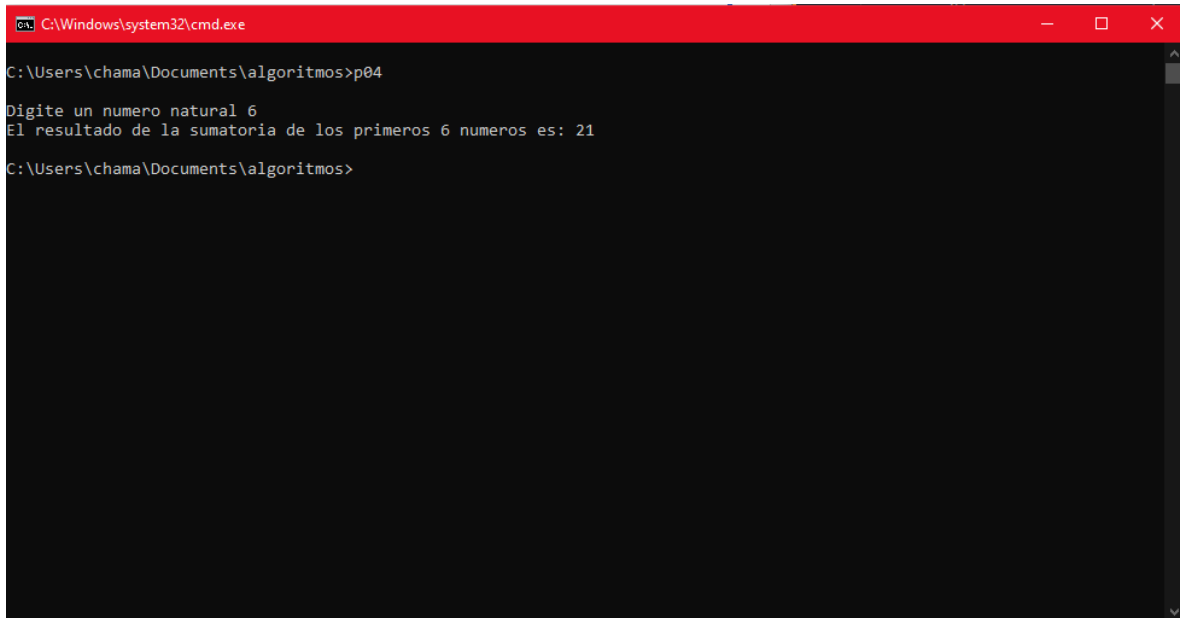
```
p04.c x p05.c x p06.c
1 #include <stdio.h>//Agregamos la libreria stdio.h que contiene las funciones básicas del lenguaje de programación C.
2
3 unsigned int Factorial(unsigned int N);//Aqui Se ponen los prototipos de las funciones que vayamos utilizando.
4
5 int main();//Inicia función Principal
6 {
7     unsigned int N,fact;//Declaracion de las variables.
8
9     printf("\nDigite un numero natural");//pedimos un numero mediante un texto que se muestra al usuario.
10    scanf("%u",&N);//guardamos el numero en la variable ya declarada anteriormente "N".
11
12    fact = Factorial(N);//Aplicamos la funcion "Factorial" al numero N y lo guardamos en la variable fact.
13
14    printf("El resultado del Factorial de %i es: %i",N,fact);//Imprimimos el resultado del factorial.
15    printf("\n");//Aqui solo se imprime un salto de linea.
16
17    return 0;//se retorna el valor de 0 para verificar que haya finalizado sin problemas
18 }
19
20 unsigned int Factorial(unsigned int N)//Aqui comienza la funcion "Factorial".
21 {
22     unsigned int productos;//Aqui se declaran las o en este caso la variable local de la funcion.
23     if(N==1) return 1;//Para evitar un bucle infinito, implementamos el caso base, que nos dice que si N es igual a 1, retorna 1.
24     return N*Factorial(N-1);//De no ser N=1, aplicamos la misma funcion a N-1 para seguir la recursividad y retornamos el resultado.
25 }
```

Código comentado del tercer problema

```
p04.c x p05.c x p06.c
1 #include <stdio.h>//Agregamos la libreria stdio.h que contiene las funciones básicas del lenguaje de programación C.
2
3 unsigned int Fibonacci(unsigned int n);//Aqui Se ponen los prototipos de las funciones que vayamos utilizando.
4
5 int main();//Inicia función Principal
6 {
7     unsigned int N,n;//Declaracion de las variables.
8     printf("Calculo de los primeros N numeros de la serie fibonacci\n");//Aqui solo se imprime al usuario la funcion del codigo
9     printf("Escriba un numero natural\n");//pedimos un numero mediante un texto que se muestra al usuario.
10    scanf("%u",&N);//guardamos el numero en la variable ya declarada anteriormente "N".
11    for(n=0; n<=N; n++)//Iniciamos un ciclo for con el contador de n que ira incrementando hasta ser igual que N(valor escrito por el usuario).
12    {
13        printf("%i ",Fibonacci(n));//Cada numero de este contador, se le aplica la funcion Fibonacci para luego imprimirlo en pantalla.
14    }
15    printf("\n");//Aqui solo se imprime un salto de linea.
16    return 0;//se retorna el valor de 0 para verificar que haya finalizado sin problemas
17 }
18
19 unsigned int Fibonacci(unsigned int n)//Aqui comienza la funcion "Fibonacci".
20 {
21     unsigned int fn;//Aqui se declara la variable local de la funcion.
22
23     if(n==0) return 0;//En este problema en particular tenemos dos casos base.
24     if(n==1) return 1;//Bonde si n=0, se retorna 0 y si n=1, se retorna 1.
25
26     fn = Fibonacci(n-1)+Fibonacci(n-2);//De no ser 0, ni 1, pero mayor a estos, y para seguir con la recursividad, aplicamos la funcion Fibonacci a n-1 y n-2 y lo guardamos en fn
27     return fn;//la funcion retorna el valor de fn.
28 }
```

Pruebas y resultados

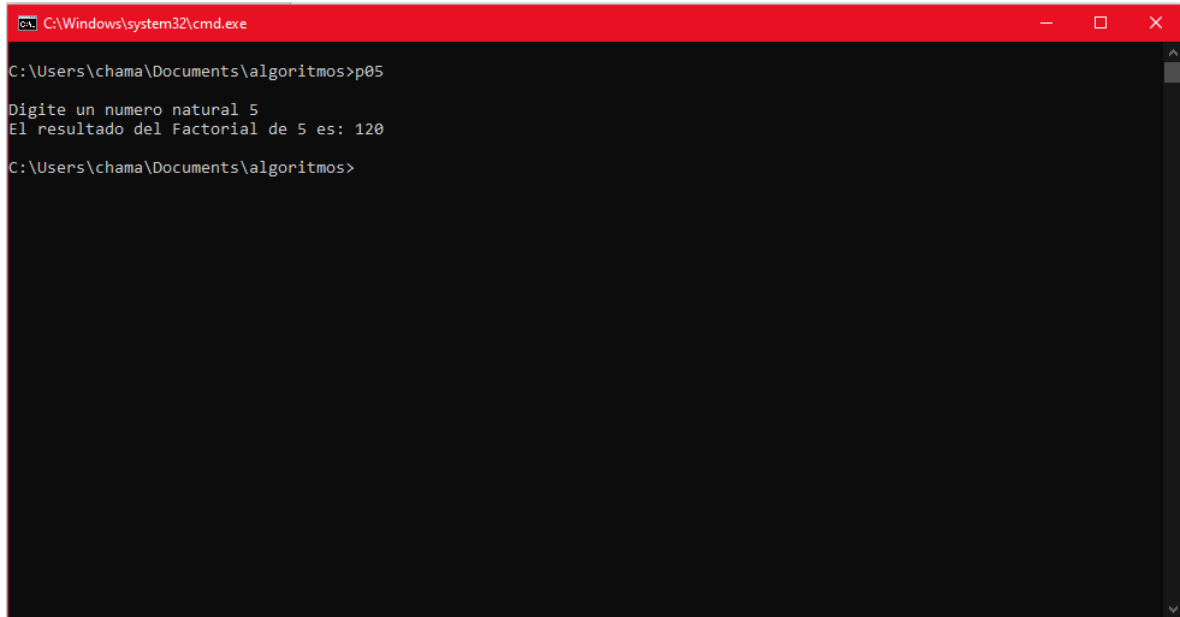
Evidencia del primer problema



```
C:\Windows\system32\cmd.exe

C:\Users\chama\Documents\algoritmos>p04
Digite un numero natural 6
El resultado de la sumatoria de los primeros 6 numeros es: 21
C:\Users\chama\Documents\algoritmos>
```

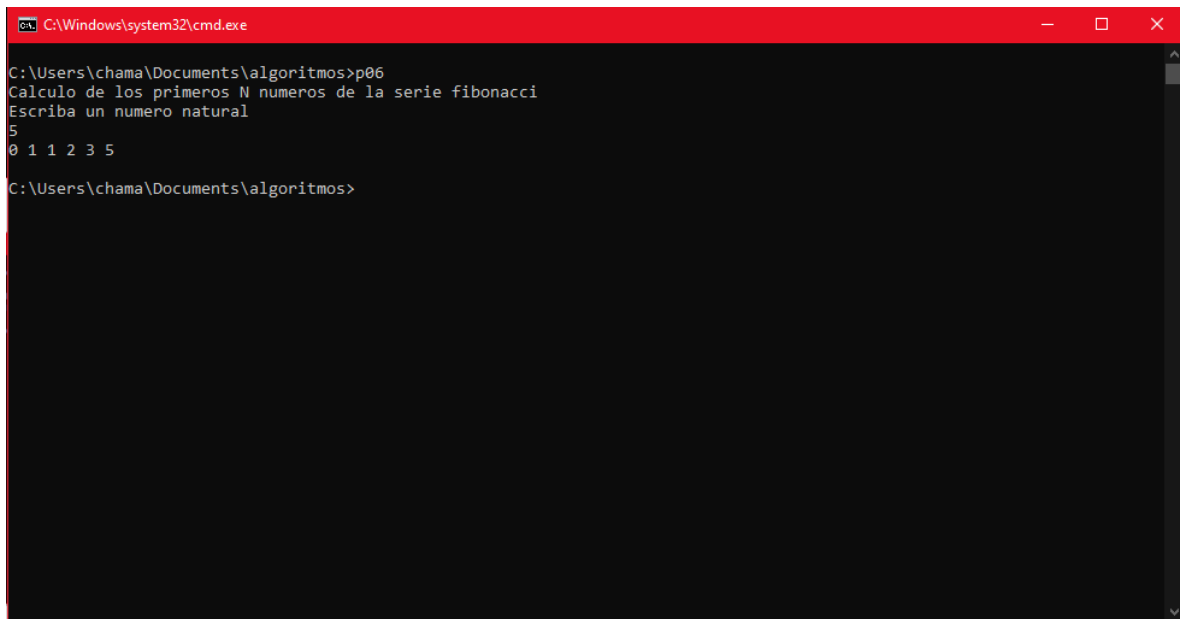
Evidencia del segundo problema



```
C:\Windows\system32\cmd.exe

C:\Users\chama\Documents\algoritmos>p05
Digite un numero natural 5
El resultado del Factorial de 5 es: 120
C:\Users\chama\Documents\algoritmos>
```

Evidencia del tercer problema



```
C:\Windows\system32\cmd.exe
C:\Users\chama\Documents\algoritmos>p06
Calculo de los primeros N numeros de la serie fibonacci
Escriba un numero natural
5
0 1 1 2 3 5
C:\Users\chama\Documents\algoritmos>
```

Programa 1

Justo como lo habíamos planteado en el problema, poniéndolo a prueba en el programa que hicimos, y por el resultado obtenido en la evidencia, tenemos que la sumatoria de los primeros 6 números es precisa con el resultado de 21.

Programa 2

En el segundo problema tenemos también los resultados precisos que previamente habíamos puesto en el problema siendo que la factorial de 5 es 120.

Programa 3

Según vimos en la imagen de la serie Fibonacci que puse en el problema, tenemos que efectivamente los primeros 6 números son 0,1,1,2,3,5 confirmando que el siguiente número de la serie es la suma de los últimos 2.

Al principio no me quedó del todo claro este tema de la recursividad y tuve un poco de problemas al analizar los códigos para ver exactamente cómo funciona, pero al final, la recursividad es un recurso muy útil ya que podemos recurrir a ella cuando no haya una solución iterativa simple y una vez repasándolo un par de veces, no es difícil de entender.