

Calcular el número de impresiones

Código 01

```
//Código 01
for(i=10; i<n*5;i*=2)
    printf("Algoritmos");
```

Se observa que el iterador comienza en 10 y termina hasta que la comparación sea menor a $5n$. Así mismo, el 10 se va multiplicando por potencias de 2, por lo tanto, se debe buscar cuantas veces se puede multiplicar $10 * 2^x$ sin que se pase de $5n$.

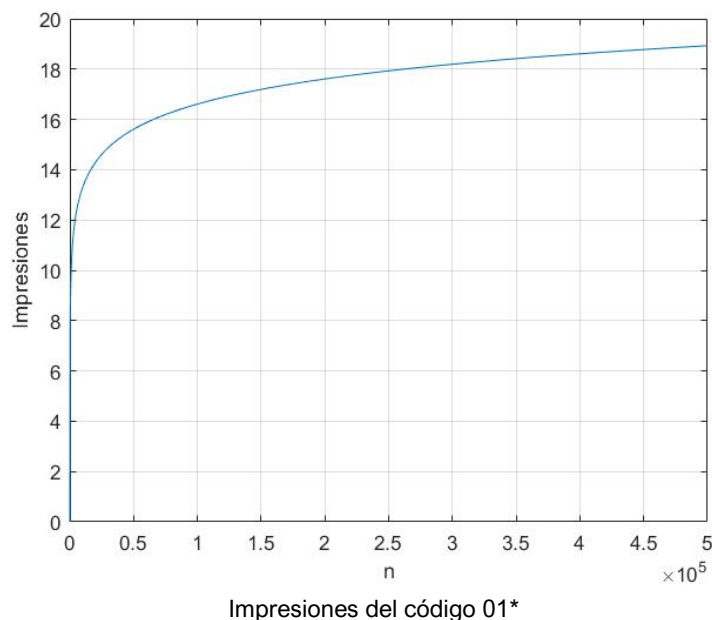
$$10 * 2^x = 5n$$


$$2^x = \frac{5n}{10}$$

$$x = \log_2 \left(\frac{n}{2} \right)$$

De esta forma, se observa que a partir de $n=2$, el hecho que $x = 0$ implica que el único número que cabe antes de pasarse de $5n$ es $i = 10 * 2^0$, es decir, que se imprimió una vez al menos para $i = 10$

$$\text{impresiones} = \text{floor} \left[\log_2 \left(\frac{n}{2} \right) + 1 \right]$$



Código 02


```
//Código 02
for(j=n; j>1; j/=2){
    if(j<(n/2)){
        for(i=0; i<n; i+=2){
            printf("Algoritmos");
        }
    }
}
```

En el for más externo, el iterador se va multiplicando por potencias de $\frac{1}{2}$, es decir, para encontrar cuantas veces se ejecuta es necesario hacer cuantas veces n es divisible entre dos antes de llegar a 1, sin embargo, como no puede ser igual a 1 se elimina esa opción:

$$n \left(\frac{1}{2} \right)^x = 1$$

$$x = \log_{\frac{1}{2}} \left(\frac{1}{n} \right) - 1$$

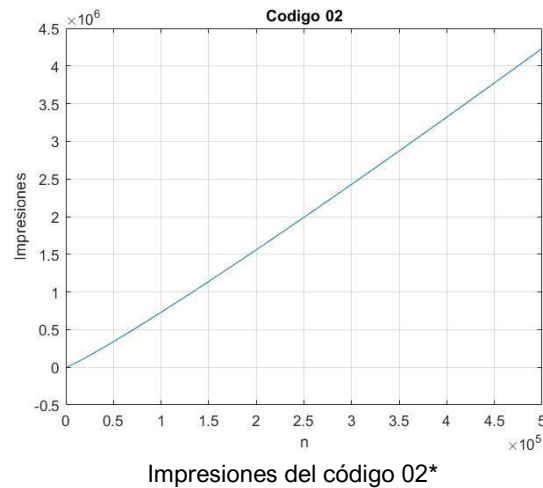
Además, en el primer if se hace una exclusión de la primera división entre 2, por lo tanto, el for más externo se ejecuta:

$$for_{exterior} = \log_{\frac{1}{2}} \left(\frac{1}{n} \right) - 2$$

Finalmente, el for interno se ejecuta $\frac{n}{2}$ veces, así, el número de impresiones está dado por:

$$impresiones = floor \left[\frac{n}{2} \right] * floor \left[\log_{\frac{1}{2}} \left(\frac{1}{n} \right) - 2 \right]$$

En donde es importante notar que antes de realizar la multiplicación a ambos factores se les debe obtener la función piso, en caso contrario se pueden tener variaciones.



Código 03

```
//Código 03
for(i=0; i<n*5; i+=2){
    for(j=0; j<2*n; j++){
        for(k=j; k<n; k++){
            printf("Algoritmos");
        }
    }
}
```

Como se puede observar, el for exterior se ejecuta hasta que el iterador i sea menor que $5n$, esto va en pasos de dos en dos, entonces su expresión está dada por:

$$\text{floor} \left[\frac{5n}{2} \right]$$

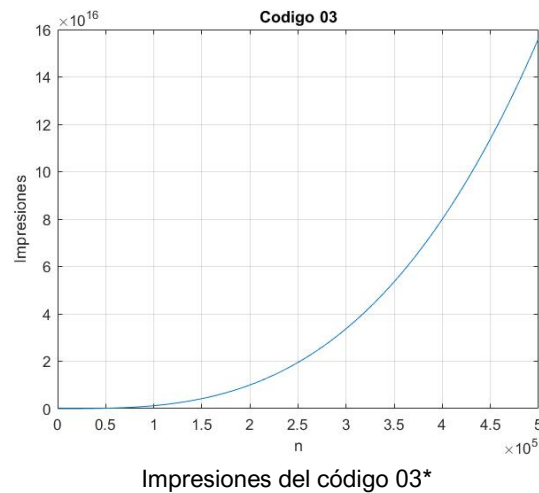
Por otra parte, analizando los for restantes, se tiene que el for con el iterador j se ejecuta $2n$ veces mientras que el for con el iterador k comienza en $k=j$, es decir, siempre en $k=0$ debido a que de igual forma j siempre inicia en 0.

A pesar de que j llega $2n-1$, k solamente puede llegar a un valor menor de n , visto de otra forma, en las iteraciones que van de $n < j < 2n$ no sucede nada porque k no cumpliría la condición de que $k < n$. Con esto se puede concluir que entre los dos for internos, primero se ejecutará n veces, después $n-1$ veces, $n-2$ veces y así sucesivamente. De esta forma, está dada por la expresión

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

Así, el número de impresiones está dado por:

$$\text{impresiones} = \text{floor}\left[\frac{5n}{2}\right] * \frac{n(n+1)}{2}$$



Código 04

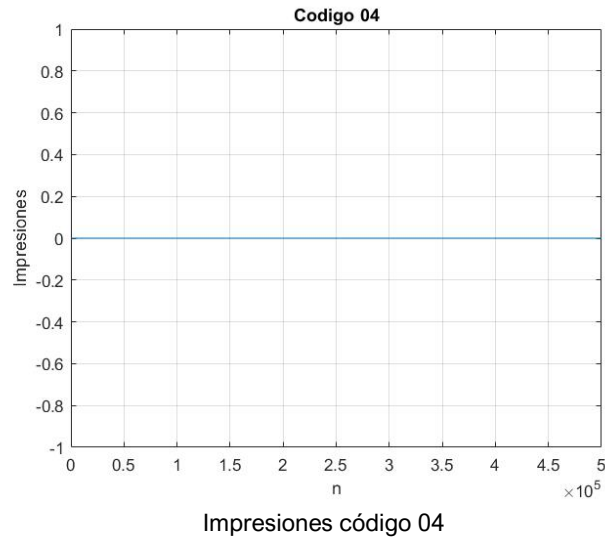
```
//Código 04
i = n;
while(i>=0){
    for(j=n; i<j; i-=2, j/=2){
        printf("Algoritmo");
    }
}
```

Analizando el código, la primera instrucción hace que i tome el valor de n , posteriormente en el for, j toma el valor de n y se hace la comparación de $i < j$, sin embargo, esto nunca será cierto debido a que $i = j = n$, así mismo, j disminuye de manera exponencial mientras que i lo hace de forma lineal, por lo tanto, nunca se entrará al for.

Además, debido a que nunca se entra al for quiere decir que nunca se ejecutará la operación $i -= 2$, por lo tanto, i siempre será mayor que 0 (a menos que se ingrese

una n negativa), cumpliéndose siempre la condición del while y se hará un bucle infinito.

impresiones = 0



Código 05

```
//Código 05
for(i=1; i<4*n; i*=2){
    for(j=i; j<5*n; j+=3){
        printf("Algoritmos");
    }
}
```

En este caso, se tiene que el iterador del for externo va tomando valores de potencia de 2, siempre y cuando no sobrepase $4n$ y considerando la primera iteración cuando $i=1$. De esto, la cantidad de veces que se ejecutará está dada por:

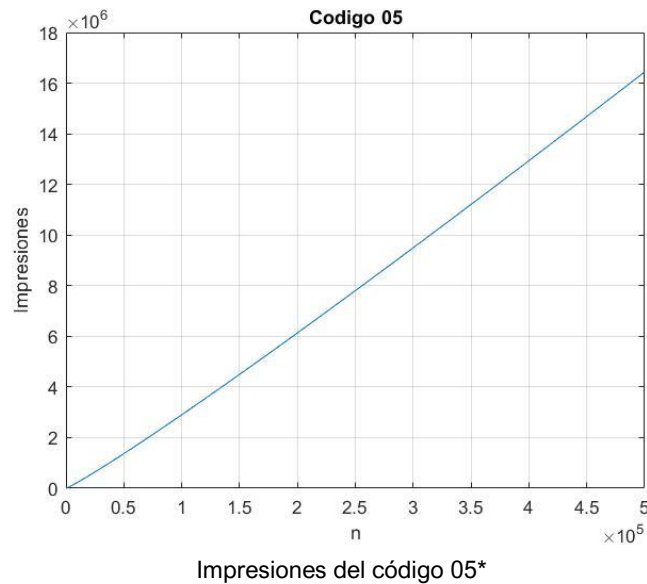
$$2^x = 4n$$

$$x = \log_2(4n) + 1$$

Por su parte, el iterador del for interior va avanzando en pasos de 3 en 3 siempre y cuando no sobrepase $5n$, sin embargo, dicho iterador comienza en $j=i$, es decir, que j va tomando valores de potencias de 2, por lo tanto, se puede tomar como una sumatoria donde el límite superior sería la x calculada anteriormente.

$$impresiones = \sum_{i=0}^{\log_2(4n)+1} \frac{5n - 2^i}{3}$$

Donde tanto el límite de la suma como el resultado de la división son funciones piso.



*Sin usar función floor.

Código	Tipo \ n	-1	0	1	2	3	5	15	20	100	409	500	593	1000	1471	1500	2801	3000	5000	10000	20000
01	Teórico	*	*	0	1	1	2	3	4	6	8	8	9	9	10	10	11	11	12	13	14
	Empírico	0	0	0	0	1	2	3	4	6	8	8	9	9	10	10	11	11	12	13	14
02	Teórico	*	*	0	-1	-1	0	7	20	200	1,224	1,500	2,072	3,500	5,880	6,000	12,600	13,500	25,000	55,000	120,000
	Empírico	0	0	0	0	0	0	8	20	200	1,230	1,500	2,079	3,500	5,888	6,000	12,609	13,500	25,000	55,000	120,000
03	Teórico	0	0	2	15	42	180	4,440	10,500	1,262,500	85,689,590	156,562,500	261,011,322	1,251,250,000	3,980,926,112	4,221,562,500	27,477,255,402	33,761,250,000	156,281,250,000	1,250,125,000,000	10,000,500,000,000
	Empírico	0	0	3	15	48	195	4,560	10,500	1,262,500	85,773,435	156,562,500	261,187,443	1,251,250,000	3,982,008,768	4,221,562,500	27,481,179,603	33,761,250,000	156,281,250,000	#	#
04	Teórico	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Empírico	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
05	Teórico	*	*	2	7	12	27	129	179	1,324	6,813	8,631	10,114	18,934	28,860	29,532	59,098	64,070	111,483	239,640	512,613
	Empírico	0	0	3	8	17	32	132	192	1,333	6,820	8,486	10,497	18,639	29,146	29,776	59,898	64,546	114,080	244,827	522,979

*La función no está definida.

#Como se observa con el modelo teórico, la salida es muy grande y requiere de mucho procesamiento.

Conclusión

Como se puede observar, es posible dar una aproximación al número de impresiones que se ejecutarán e incluso obtener algún modelo matemático. Sin embargo, no siempre es posible calcular el número exacto, ya que dependiendo del algoritmo este análisis puede ser sencillo o complicado, hasta el grado de necesitar software que permita obtener el resultado de las operaciones, aunque aún así la función obtenida sirve para obtener una idea cercana del comportamiento del algoritmo.

Códigos

Código 01

```
#include <stdio.h>

int main(){
    int i, n, impresiones=0;
    printf("Ingrese n\n");
    scanf("%d", &n);

    for(i=10; i<n*5; i*=2){
        impresiones++;
    }

    printf("para n=%d tuvo impresiones=%d", n, impresiones);
}
```


Código 02

```
#include <stdio.h>

int main(){
    int i, j, n, impresiones=0;
    printf("Ingrese n\n");
    scanf("%d", &n);

    //Código 02
    for(j=n; j>1; j/=2){
        if(j<(n/2)){
            for(i=0; i<n; i+=2){
                impresiones++;
            }
        }
    }

    printf("para n=%d tuvo impresiones=%d", n, impresiones);
}
```

Código 03

```
#include <stdio.h>

int main(){
    int i, j, k, n;
    double impresiones=0;
    printf("Ingrese n\n");
    scanf("%d", &n);

    //Código 03
    for(i=0; i<n*5; i+=2){
        for(j=0; j<2*n; j++){
            for(k=j; k<n; k++){
                impresiones++;
            }
        }
    }

    printf("para n=%d tuvo impresiones=%f", n, impresiones);
}
```

Código 04

```
#include <stdio.h>

int main(){
    int i, j,k, n;
    double impresiones=0;
    printf("Ingrese n\n");
    scanf("%d", &n);

    //Código 04
    i = n;
    while(i>=0){
        printf("%d", i);
        for(j=n; i<j; i-=2, j/=2){
            impresiones++;
        }
    }

    printf("para n=%d tuvo impresiones=%f", n, impresiones);
}
```

Código 05

```
#include <stdio.h>

int main(){
    int i, j,k, n, impresiones=0;
    printf("Ingrese n\n");
    scanf("%d", &n);

    //Código 05
    for(i=1; i<4*n; i*=2){
        for(j=i; j<5*n; j+=3){
            impresiones++;
        }
    }

    printf("para n=%d tuvo impresiones=%d", n, impresiones);
}
```