



Análisis de Algoritmos

Ejercicio 01: "Calcular el número de impresiones"

Nombre: Luis Fernando Ramírez Cotonieto

Fecha de entrega: 18 de Marzo del 2021

Grupo: 3CM13



Ejercicios 01: Calcular el número de impresiones

Análisis de Algoritmos

1. Código 01

Se tiene el siguiente programa donde "n" tomará los valores de:

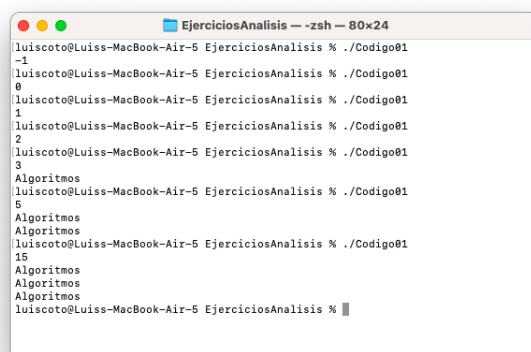
[-1, 0, 1, 2, 3, 5, 15, 20, 100, 409, 500, 593, 1000, 1471, 1500, 2801, 3000, 5000, 10000, 20000]

```
1 //Codigo 1 - Analisis de Algoritmos - 3CM13
2 // Luis Fernando Ramirez Cottonieto
3
4 #include <stdio.h>
5
6 void Algoritmos(int n){
7     int i,j=0;
8     for(i=10;i<n*5;i*=2){
9         printf("Algoritmos\n");
10    }
11 }
12
13 int main (void){
14     int n;
15     scanf("%d", &n);
16     Algoritmos(n);
17     return 0;
18 }
```

Podemos analizar el número de impresiones por medio de la siguiente expresión:

$$f(n)=\log_2 n$$

El código nos da la primera idea que es $i=10$, por lo que tendremos que multiplicar los números dados por 5 hasta obtener esta cualidad. Después obtenemos una ecuación de la forma $a^c = b$, siendo $a=2$ y $b=n$, por lo que al ponerlo en su forma logarítmica obtenemos la expresión anterior. Al realizar las pruebas de escritorio, se obtienen los siguientes resultados:



```
EjerciciosAnalisis -- zsh -- 80x24
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo01
-1
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo01
0
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo01
1
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo01
2
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo01
3
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo01
5
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo01
15
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis %
```

Figura 1: Cuando n es -1, 0, 1, 2, 3, 5 y/o 15

```
EjerciciosAnálisis — -zsh — 80x24
20
Algoritmos
Algoritmos
Algoritmos
Algoritmos
[luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
100
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
[luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
409
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 2: Cuando n es 20, 100 y/o 409

```
EjerciciosAnálisis — -zsh — 80x24
[luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
500
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
[luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
593
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 3: Cuando n es 500 y/o 593

```
EjerciciosAnálisis — -zsh — 80x24
[luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
1000
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
[luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
1471
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 4: Cuando n es 1000 y/o 1471

```
EjerciciosAnálisis — -zsh — 80x24
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
1500
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 5: Cuando n es 1500

```
EjerciciosAnálisis — -zsh — 80x24
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
2801
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 6: Cuando n es 2801

```
EjerciciosAnálisis — -zsh — 80x24
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
3000
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 7: Cuando n es 3000

```
EjerciciosAnálisis — -zsh — 80x24
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
5000
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 8: Cuando n es 5000

```
EjerciciosAnálisis — -zsh — 80x24
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
10000
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 9: Cuando n es 10,000

```
EjerciciosAnálisis — -zsh — 80x24
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis % ./Codigo01
20000
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnálisis %
```

Figura 10: Cuando n es 20,000

n	Valor teórico	Valor empírico
-1	-	0
0	-	0
1	0	0
2	1	0
Debido a la sentencia $i=10$; $i < n*5$, empezamos aquí.		
3	1.584	1
5	2.321	2
15	3.906	3
20	4.321	4
100	6.643	6
409	8.675	8
500	8.965	8
593	9.211	9
1000	9.965	9
1471	10.522	10
1500	10.550	10
2801	11.451	11
3000	11.550	11
5000	12.287	12
10000	13.287	13
20000	14.287	14

Cuadro 1: Tabla de resultados teóricos y empíricos

Al hacer una comparación de resultados teóricos y empíricos, se obtiene una tabla como la que se situa en la parte posterior.

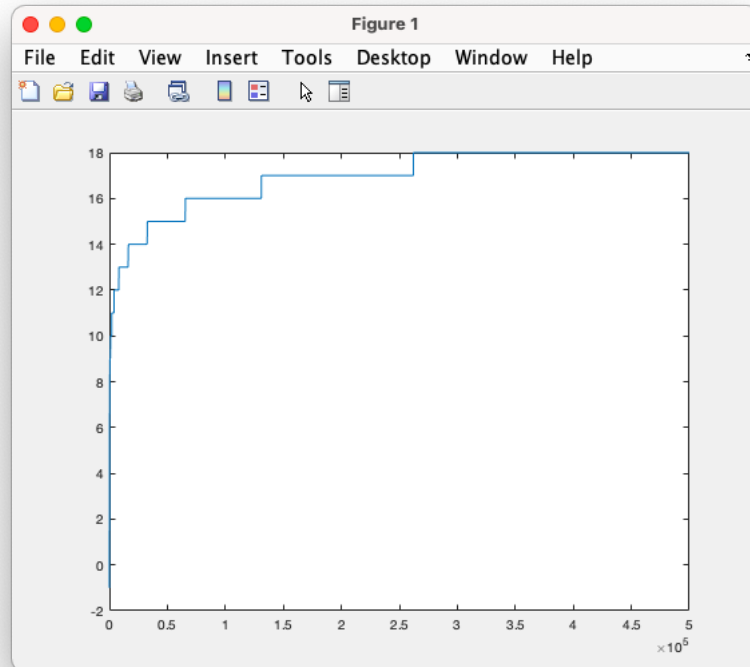


Figura 11: Gráfica del código 1 [MATLAB]

2. Código 02

Se tiene el siguiente programa donde "n" tomará los valores de:

[-1, 0, 1, 2, 3, 5, 15, 20, 100, 409, 500, 593, 1000, 1471, 1500, 2801, 3000, 5000, 10000, 20000]

```
1 //Codigo 2 - Analisis de Algoritmos - 3CM13
2 // Luis Fernando Ramirez Cotonieto
3
4 #include <stdio.h>
5
6 void Algoritmos(int n){
7     int i,j;
8     for(j=n;j>1;j/=2){
9         {
10             if(j<(n/2))
11             {
12                 for(i=0;i<n;i+=2)
13
14                 {
15                     printf("Algoritmos\n");
16                 }
17             }
18         }
19     }
20 }
21
22 int main (void){
23     int n;
24     scanf("%d", &n);
25     Algoritmos(n);
26     return 0;
27 }
```

Podemos analizar el número de impresiones por medio de la siguiente expresión:

$$f(n) = \text{piso}(\log_2(n-2)) \cdot \text{techo}\left(\frac{n}{2}\right)$$

Esta expresión un tanto más compleja se obtiene de un razonamiento similar al anterior, el segundo for nos denota el $\frac{n}{2}$ y de este desprendemos el conocer si son funciones techo o funciones suelo para mayor exactitud del programa.

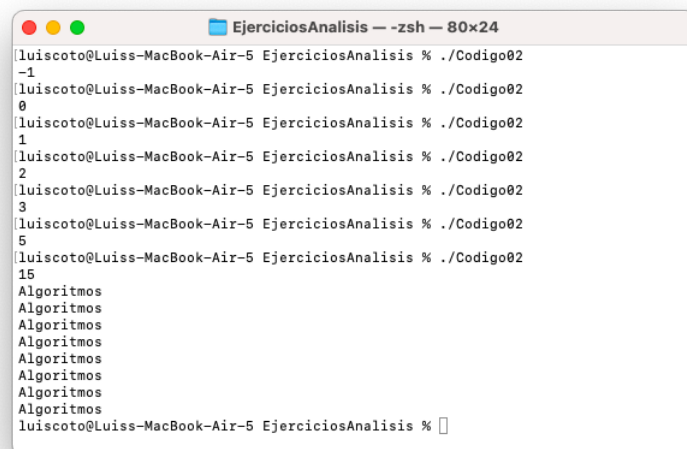


Figura 12: Cuando n es 1, 3, 5 y 15

n	Valor teórico	Valor empírico
-1	-	0
0	-	0
1	-	0
2	-	0
3	-	0
5	-	0
15	8	8
20	20	20
100	200	200
409	1230	1230
500	1500	1500
593	2079	2079
1000	3500	3500
1471	5888	5888
1500	6000	6000
2801	12,609	12,609
3000	13,500	13,500
5000	25,000	25,000
10000	55,000	55,000
20000	120,000	120,000

Cuadro 2: Tabla de resultados teóricos y empíricos

De los resultados anteriores obtenemos una tabla que se muestra en la parte superior y una gráfica característica que se muestra en la parte superior.

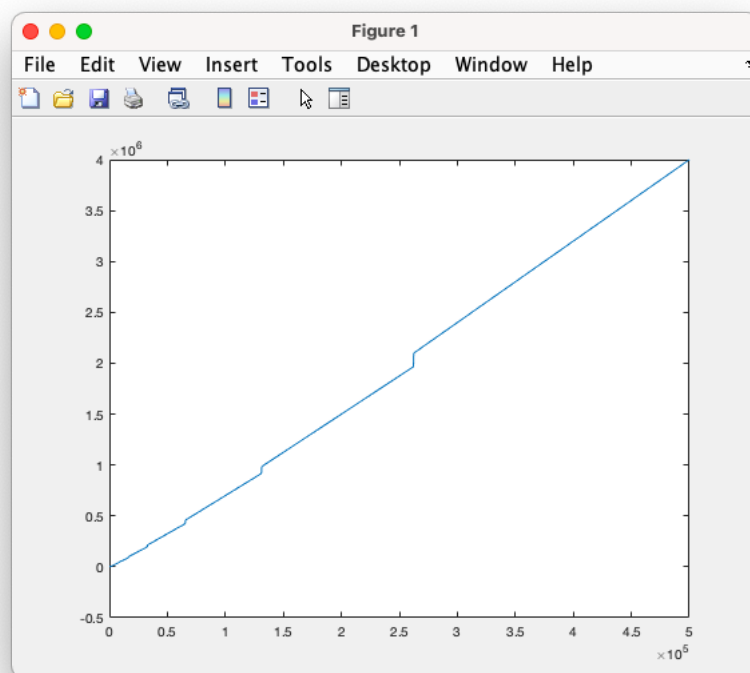


Figura 15: Gráfica del código 2 [MATLAB]

3. Código 03

Se tiene el siguiente programa donde "n" tomará los valores de:

[-1, 0, 1, 2, 3, 5, 15, 20, 100, 409, 500, 593, 1000, 1471, 1500, 2801, 3000, 5000, 10000, 20000]

```
1 //Codigo 3 - Analisis de Algoritmos - 3CM13
2 // Luis Fernando Ramirez Cotonieto
3
4 #include <stdio.h>
5
6 void Algoritmos(int n){
7     int i,j, k;
8     unsigned long int cont=0;
9
10    for(i=0; i<n*5; i+=2){
11        {
12            for(j=0; j<2*n; j++)
13            {
14                for(k=j; k<n; k++)
15
16                {
17                    printf("Algoritmos\n", cont);
18                }
19            }
20        }
21    }
22 }
23
24 int main (void){
25     int n;
26     scanf("%d", &n);
27     Algoritmos(n);
28     return 0;
29 }
```

Podemos analizar el número de impresiones por medio de la siguiente expresión:

$$f(n) = ((\frac{n^2+n}{2})) \cdot techo(\frac{5 \cdot n}{2})$$

Se obtiene desde el primer for el cual tiene como función techo $\frac{5 \cdot n}{2}$, los otros dos for los encontramos interactuando entre ellos por lo que los podemos tomar como uno solo similar a $f(n) = (n(\frac{n+1}{2}))$, donde observamos los comportamientos de j con relación a n y a 3.



Figura 16: Cuando n es 3

n	Valor teórico	Valor empírico
-1	0	0
0	0	0
1	3	3
2	15	15
3	48	48
5	195	195
15	4560	4560
20	10500	10500
100	1262500	1262500
409	85773435	85773435
500	156562500	156562500
593	261187443	261187443
1000	1251250000	1251250000
1471	3982008768	-
1500	4221562500	-
2801	27481179603	-
3000	33761250000	-
5000	156281250000	-
10000	1250125000000	-
20000	10000500000000	-

Cuadro 3: Tabla de resultados teóricos y empíricos

De los resultados anteriores obtenemos una tabla que se muestra en la parte superior y una gráfica característica que se muestra en la parte inferior.

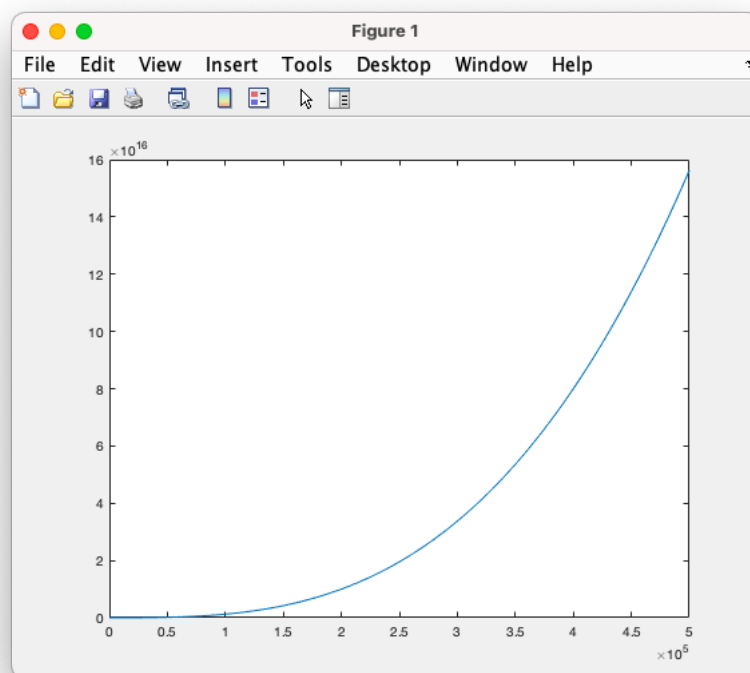


Figura 17: Gráfica del código 3 [MATLAB]

4. Código 04

Se tiene el siguiente programa donde "n" tomará los valores de:

[-1, 0, 1, 2, 3, 5, 15, 20, 100, 409, 500, 593, 1000, 1471, 1500, 2801, 3000, 5000, 10000, 20000]

```
1 //Codigo 4 - Analisis de Algoritmos - 3CM13
2 // Luis Fernando Ramirez Cottonieto
3
4 #include <stdio.h>
5
6 void Algoritmos(int n){
7     int i=n;
8     int j;
9
10    while(i>=0){
11        for(j=n;i<j;i-=2, j/=2)
12        {
13
14            printf("Algoritmos\n");
15        }
16    }
17 }
18
19 int main (void){
20     int n;
21     scanf("%d", &n);
22     Algoritmos(n);
23     return 0;
24 }
```

Este código tiene una curiosidad, ya que tanto i como j terminan teniendo un valor de "n", por lo que estamos haciendo un ciclo infinito que no nos conduce a ningun resultado realmente, es decir, no termina. Podemos analizar el número de impresiones por medio de la siguiente expresión:

$$f(N) = \infty$$

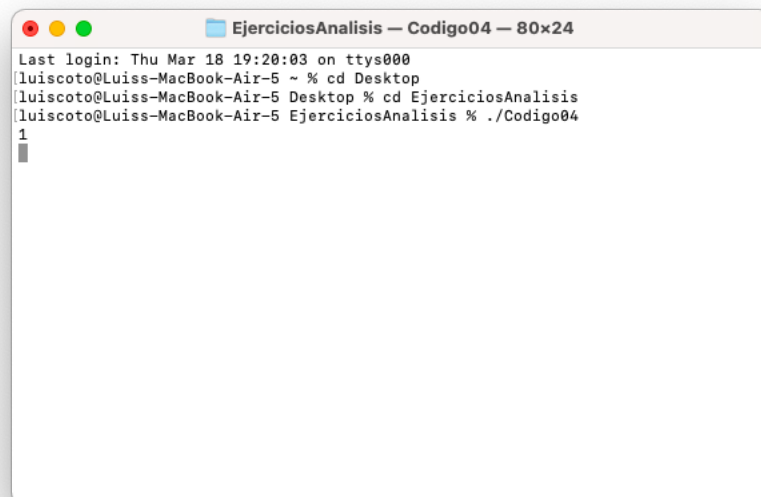


Figura 18: Compilación del código 4

5. Código 05

Se tiene el siguiente programa donde "n" tomará los valores de:

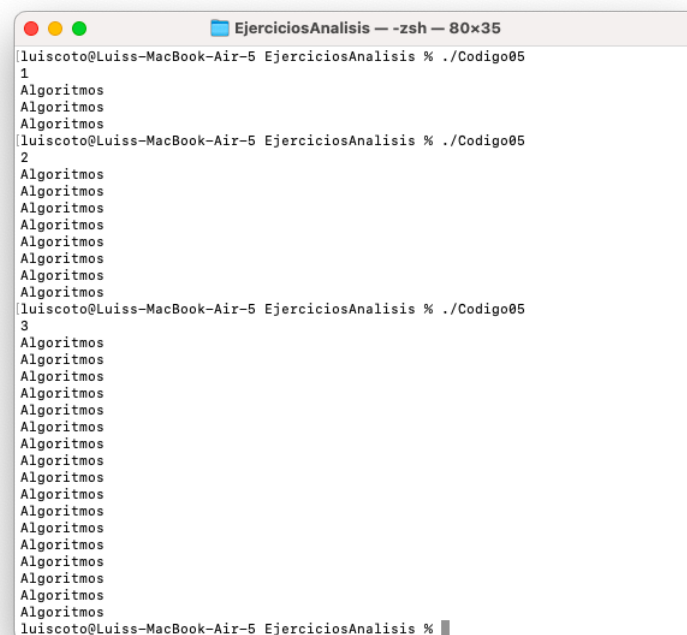
[-1, 0, 1, 2, 3, 5, 15, 20, 100, 409, 500, 593, 1000, 1471, 1500, 2801, 3000, 5000, 10000, 20000]

```
1 //Codigo 5 - Analisis de Algoritmos - 3CM13
2 // Luis Fernando Ramirez Cottonieto
3
4 #include <stdio.h>
5
6 void Algoritmos(int n){
7     int i,j=0;
8     for(i=1;i<4*n;i*=2){
9         for(j=i;j<5*n;j+=3){
10            printf("Algoritmos\n");
11        }
12    }
13 }
14 int main (void){
15     int n;
16     scanf("%d", &n);
17     Algoritmos(n);
18     return 0;
19 }
20 }
```

Podemos analizar el número de impresiones por medio de la siguiente expresión:

$$f(n)=\sum_{i=0}^n \frac{5n-i}{3}$$

El razonamiento de este ejercicio parte desde su ciclo interno donde encontramos $\frac{5n-1}{3}$, como la sentencia anterior nos delimita las condiciones, se puede intuir automáticamente que estamos hablando de una sumatoria de valores que puede llegar hasta el número que lo deseemos.



```
EjerciciosAnalisis -- zsh -- 80x35
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo05
1
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo05
2
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis % ./Codigo05
3
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
Algoritmos
luiscoto@Luiss-MacBook-Air-5 EjerciciosAnalisis %
```

Figura 19: Cuando n es 1,2 y 3

De los resultados anteriores obtenemos una tabla y una gráfica característica de la siguiente manera.

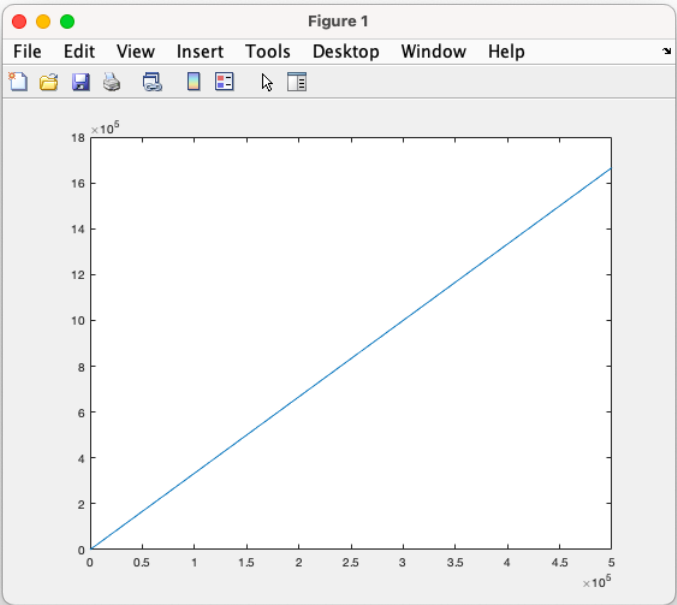


Figura 22: Gráfica del código 5 [MATLAB]