

Paso 0.

Revisé actualizaciones

```
17 de ago 14:18 nicole@vbox:~  
nicole@vbox:~$ sudo dnf check-update  
[sudo] contraseña para nicole:  
Actualizando y cargando repositorios:  
Repositorios cargados.  
glycin-loaders.x86_64      1.2.3-3.fc42    updates  
librados2.x86_64           2:19.2.3-3.fc42  updates  
librbd1.x86_64              2:19.2.3-3.fc42  updates  
mdadm.x86_64                4.3-8.fc42     updates  
mkpasswd.x86_64             5.6.4-1.fc42   updates  
nspr.x86_64                 4.37.0-1.fc42  updates  
nss.x86_64                  3.114.0-1.fc42  updates  
nss-softokn.x86_64          3.114.0-1.fc42  updates  
nss-softokn-freebl.x86_64   3.114.0-1.fc42  updates  
nss-sysinit.x86_64          3.114.0-1.fc42  updates  
nss-util.x86_64              3.114.0-1.fc42  updates  
python3-boto3.noarch         1.40.9-1.fc42   updates  
python3-botocore.noarch       1.40.9-1.fc42   updates  
whois.x86_64                 5.6.4-1.fc42   updates  
whois-nls.noarch             5.6.4-1.fc42   updates  
nicole@vbox:~$
```

Hice las actualizaciones disponibles

```
17 de ago 14:19 nicole@vbox:~  
+  
whois.x86_64      5.6.4-1.fc42    updates  
whois-nls.noarch   5.6.4-1.fc42    updates  
nicole@vbox:~$ sudo dnf upgrade  
Actualizando y cargando repositorios:  
Repositorios cargados.  
Paquete          Arq.    Versión       Repositor     Tamaño  
Modernizando:  
glycin-loaders   x86_64  1.2.3-3.fc42  updates        10.9 MiB  
    reemplazando glycin-loaders   x86_64  1.2.2-1.fc42  updates        10.7 MiB  
librados2         x86_64  2:19.2.3-3.fc42  updates        17.3 MiB  
    reemplazando librados2       x86_64  2:19.2.3-1.fc42  updates        17.3 MiB  
librbd1           x86_64  2:19.2.3-3.fc42  updates        15.4 MiB  
    reemplazando librbd1        x86_64  2:19.2.3-1.fc42  updates        15.4 MiB  
mdadm             x86_64  4.3-8.fc42    updates        1.0 MiB  
    reemplazando mdadm          x86_64  4.3-7.fc42    624cbc925  1.0 MiB  
mkpasswd          x86_64  5.6.4-1.fc42  updates        39.3 KiB  
    reemplazando mkpasswd        x86_64  5.6.2-1.fc42  updates        39.3 KiB  
nspr              x86_64  4.37.0-1.fc42  updates        315.5 KiB  
    reemplazando nspr           x86_64  4.36.0-10.fc42  updates        315.5 KiB  
nss               x86_64  3.114.0-1.fc42  updates        1.9 MiB  
    reemplazando nss            x86_64  3.113.0-1.fc42  updates        1.9 MiB  
nss-softokn       x86_64  3.114.0-1.fc42  updates        1.9 MiB  
    reemplazando nss-softokn    x86_64  3.113.0-1.fc42  updates        1.9 MiB  
nss-softokn-freebl x86_64  3.114.0-1.fc42  updates        848.4 KiB  
    reemplazando nss-softokn-freebl x86_64  3.113.0-1.fc42  updates        848.5 KiB  
nss-sysinit       x86_64  3.114.0-1.fc42  updates        18.1 KiB
```

Ejecuté comando para instalar GCC

```
17 de ago 14:22 nicole@vbox:~
[13/32] Actualizando python3-boto3-0:1.40.9-1. 100% | 34.3 MiB/s | 2.2 MiB | 00m00s
[14/32] Actualizando mkpasswd-0:5.6.4-1.fc42.x 100% | 1.8 MiB/s | 40.9 KiB | 00m00s
[15/32] Actualizando whois-0:5.6.4-1.fc42.x86_ 100% | 3.6 MiB/s | 174.6 KiB | 00m00s
[16/32] Actualizando mdadm-0:4.3-8.fc42.x86_64 100% | 10.1 MiB/s | 1.0 MiB | 00m00s
[17/32] Actualizando glycin-loaders-0:1.2.3-3. 100% | 129.6 MiB/s | 10.9 MiB | 00m00s
[18/32] Eliminando nss-0:3.113.0-1.fc42.x86_64 100% | 2.7 KiB/s | 19.0 B | 00m00s
[19/32] Eliminando nss-softokn-0:3.113.0-1.fc4 100% | 1.4 KiB/s | 26.0 B | 00m00s
[20/32] Eliminando nss-sysinit-0:3.113.0-1.fc4 100% | 500.0 B/s | 7.0 B | 00m00s
[21/32] Eliminando librbd1-2:19.2.3-1.fc42.x86 100% | 1.6 KiB/s | 18.0 B | 00m00s
[22/32] Eliminando python3-boto3-0:1.40.8-1.fc 100% | 15.2 KiB/s | 187.0 B | 00m00s
[23/32] Eliminando nss-softokn-freebl-0:3.113. 100% | 1.6 KiB/s | 13.0 B | 00m00s
[24/32] Eliminando nss-util-0:3.113.0-1.fc42.x 100% | 400.0 B/s | 6.0 B | 00m00s
[25/32] Eliminando whois-0:5.6.2-1.fc42.x86_64 100% | 375.0 B/s | 15.0 B | 00m00s
[26/32] Eliminando mkpasswd-0:5.6.2-1.fc42.x86 100% | 769.0 B/s | 10.0 B | 00m00s
[27/32] Eliminando whois-nls-0:5.6.2-1.fc42.no 100% | 2.2 KiB/s | 16.0 B | 00m00s
[28/32] Eliminando python3-botoocore-0:1.40.8-1 100% | 106.4 KiB/s | 2.9 KiB | 00m00s
[29/32] Eliminando nspr-0:4.36.0-10.fc42.x86_6 100% | 1.7 KiB/s | 12.0 B | 00m00s
[30/32] Eliminando librados2-2:19.2.3-1.fc42.x 100% | 625.0 B/s | 20.0 B | 00m00s
[31/32] Eliminando mdadm-0:4.3-7.fc42.x86_64 100% | 1.0 KiB/s | 46.0 B | 00m00s
[32/32] Eliminando glycin-loaders-0:1.2.2-1.fc 100% | 12.0 B/s | 30.0 B | 00m02s
¡Completado!
nicole@vbox:~$ sudo dnf install gcc
Actualizando y cargando repositorios:
Repositorios cargados.
Paquete          Arq.      Versión           Repositorio      Tamaño
Instalando:
```

17 de ago 14:23

nicole@vbox:~

Resumen de la transacción:

Instalando: 5 paquetes

El tamaño total de paquetes entrantes es 42 MiB. Se necesita descargar 42 MiB.

Después de esta operación, 122 MiB extra serán utilizados (instalar 122 MiB, eliminar 0 B).

Is this ok [y/N]: y

Paquete	Tamaño	Tasa	Tamaño	Tiempo
[1/5] make-1:4.4.1-10.fc42.x86_64	100%	396.1 KiB/s	587.0 KiB	00m01s
[2/5] glibc-devel-0:2.41-10.fc42.x86_64	100%	409.3 KiB/s	624.6 KiB	00m02s
[3/5] libxcrypt-devel-0:4.4.38-7.fc42.x86_64	100%	40.7 KiB/s	29.4 KiB	00m01s
[4/5] kernel-headers-0:6.15.3-200.fc42.x86_64	100%	1.2 MiB/s	1.7 MiB	00m01s
[5/5] gcc-0:15.2.1-1.fc42.x86_64	100%	4.1 MiB/s	39.4 MiB	00m10s

[5/5] Total 100% 3.8 MiB/s 42.3 MiB 00m11s

Ejecutando transacción

Acción	Paquete	Tamaño	Tasa	Tamaño	Tiempo
[1/7]	Verificar archivos de paquete	100%	40.0 B/s	5.0 B	00m00s
[2/7]	Preparar transacción	100%	15.0 B/s	5.0 B	00m00s
[3/7]	Instalando kernel-headers-0:6.15.3-200.f	100%	32.8 MiB/s	6.7 MiB	00m00s
[4/7]	Instalando libxcrypt-devel-0:4.4.38-7.fc	100%	2.9 MiB/s	33.1 KiB	00m00s
[5/7]	Instalando glibc-devel-0:2.41-10.fc42.x8	100%	23.8 MiB/s	2.3 MiB	00m00s
[6/7]	Instalando make-1:4.4.1-10.fc42.x86_64	100%	42.9 MiB/s	1.8 MiB	00m00s
[7/7]	Instalando gcc-0:15.2.1-1.fc42.x86_64	100%	67.0 MiB/s	111.3 MiB	00m02s

;Completado!

nicole@vbox:~\$

Finalicé la instalación de GCC

Paso 1.

Hice el código de Fibonacci con recursión de pila

The screenshot shows a terminal window with the following details:

- Top bar: Date and time: 17 de ago 14:39.
- File menu: Abrir (Open) dropdown, New (+), File (i), View (≡), Close (x).
- Current file: lab_interrupciones.c, located at ~/Documentos.

The code itself is a C program that calculates the nth Fibonacci number using a stack-based recursive approach. It includes input validation for non-negative integers and handles errors for negative inputs. The code uses standard C libraries and defines two functions: fibonacci_pila() and main().

```
#include <stdio.h>

//Fibonacci Recursion de Pila
long fibonacci_pila(long n){
    if (n <= 1){
        return n;
    } else {
        return fibonacci_pila(n-1) + fibonacci_pila(n-2);
    }
}

int main(){
    long n;
    long res;
    double tiempo_transcurrido;

    printf("Ingrese un numero entero positivo: ");
    scanf("%d", &n);

    if (n < 0){
        printf("Error: Solo se aceptan numeros enteros positivos");
        return 1;
    }
}
```

17 de ago 14:40

• lab_interrupciones.c
~/Documentos

```
Abrir v + • lab_interrupciones.c
Abrir v + ~/Documentos
    return n;
} else {
    return fibonacci_pila(n-1) + fibonacci_pila(n-2);
}
}

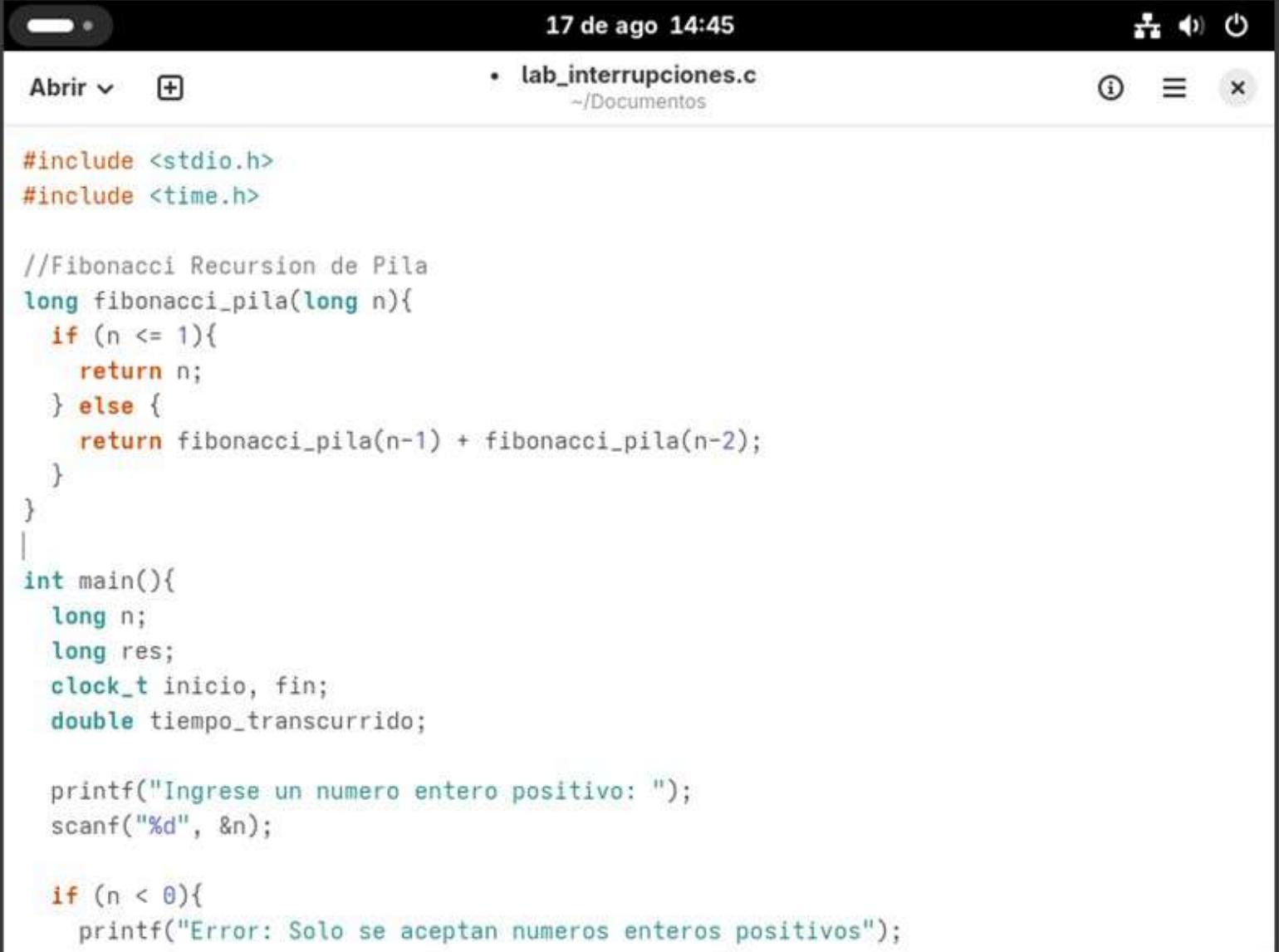
int main(){
    long n;
    long res;
    double tiempo_transcurrido;

    printf("Ingrese un numero entero positivo: ");
    scanf("%d", &n);

    if (n < 0){
        printf("Error: Solo se aceptan numeros enteros positivos");
        return 1;
    }
    res = fibonacci_pila(n);
    printf("El valor de fibonacci de %d es %ld\n", n, res);
    return 0;
}
```

Paso 2.

Agregué la biblioteca time para poder calcular el tiempo que tarda cada ejecución



```
17 de ago 14:45
Abrir + • lab_interrupciones.c ~/Documentos ⓘ ⌂ ×

#include <stdio.h>
#include <time.h>

//Fibonacci Recursion de Pila
long fibonacci_pila(long n){
    if (n <= 1){
        return n;
    } else {
        return fibonacci_pila(n-1) + fibonacci_pila(n-2);
    }
}

int main(){
    long n;
    long res;
    clock_t inicio, fin;
    double tiempo_transcurrido;

    printf("Ingrese un numero entero positivo: ");
    scanf("%d", &n);

    if (n < 0){
        printf("Error: Solo se aceptan numeros enteros positivos");
    }
}
```

17 de ago 14:46

• lab_interrupciones.c
~/Documentos

```
int main(){
    long n;
    long res;
    clock_t inicio, fin;
    double tiempo_transcurrido;

    printf("Ingrese un numero entero positivo: ");
    scanf("%d", &n);

    if (n < 0){
        printf("Error: Solo se aceptan numeros enteros positivos");
        return 1;
    }
    inicio = clock();
    res = fibonacci_pila(n);
    fin = clock();
    tiempo_transcurrido = (double)(fin - inicio)/CLOCKS_PER_SEC;
    printf("El valor de fibonacci de %d es %ld\n", n, res);
    printf("El calculo tardó: %f segundos\n", tiempo_transcurrido);
    return 0;
}
```

Paso 3.

Agregué el print("."); para cada vez que se hace una llamada a la función de Fibonacci

The screenshot shows a terminal window titled "lab_interrupciones.c" located in the "/Documentos" directory. The title bar also displays the date and time: "17 de ago 14:46". The window includes standard OS X-style controls for "Abrir", "Nuevo", "Información", "Minimizar", "Maximizar", and "Cerrar". The code itself is a C program that defines a recursive function "fibonacci_pila" and a "main" function. The "fibonacci_pila" function prints a dot (".") before each recursive call to track the execution flow. The "main" function prompts the user for a positive integer, reads it, and then checks if it's negative, printing an error message if so.

```
#include <stdio.h>
#include <time.h>

//Fibonacci Recursion de Pila
long fibonacci_pila(long n){
    printf(".");
    if (n <= 1){
        return n;
    } else {
        return fibonacci_pila(n-1) + fibonacci_pila(n-2);
    }
}

int main(){
    long n;
    long res;
    clock_t inicio, fin;
    double tiempo_transcurrido;

    printf("Ingrese un numero entero positivo: ");
    scanf("%d", &n);

    if (n < 0){
        printf("El numero debe ser positivo.");
    }
}
```

Paso 4.

Corrí el programa sin punto con los números: 5, 10, 15, 20, 30, 40, 50, 100

```
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 5
El valor de fibonacci de 5 es 5
El calculo tardó: 0.000001 segundos
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 10
El valor de fibonacci de 10 es 55
El calculo tardó: 0.000002 segundos
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 15
El valor de fibonacci de 15 es 610
El calculo tardó: 0.000006 segundos
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 20
El valor de fibonacci de 20 es 6765
El calculo tardó: 0.000055 segundos
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 30
El valor de fibonacci de 30 es 832040
El calculo tardó: 0.006528 segundos
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 40
El valor de fibonacci de 40 es 102334155
El calculo tardó: 0.800849 segundos
nicole@vbox:~/Documentos$
```

17 de ago 15:07



nicole@vbox:~/Documentos
~/Documentos



```
nicole@vbox:~/Documentos$ ./lab_interrupciones
```

Ingrese un numero entero positivo: 30

El valor de fibonacci de 30 es 832040

El calculo tardo: 0.006528 segundos

```
nicole@vbox:~/Documentos$ ./lab_interrupciones
```

Ingrese un numero entero positivo: 40

El valor de fibonacci de 40 es 102334155

El calculo tardo: 0.800849 segundos

```
nicole@vbox:~/Documentos$ ./lab_interrupciones
```

Ingrese un numero entero positivo: 50

El valor de fibonacci de 50 es 12586269025

El calculo tardo: 99.167858 segundos

```
nicole@vbox:~/Documentos$ ./lab_interrupciones
```

Ingrese un numero entero positivo: 51

El valor de fibonacci de 51 es 20365011074

El calculo tardo: 161.442491 segundos

```
nicole@vbox:~/Documentos$ ./lab_interrupciones
```

Ingrese un numero entero positivo: 52

El valor de fibonacci de 52 es 32951280099

El calculo tardo: 260.153015 segundos

```
nicole@vbox:~/Documentos$ ./lab_interrupciones
```

Ingrese un numero entero positivo: 53

El valor de fibonacci de 53 es 53316291173

El calculo tardo: 428.375605 segundos

```
nicole@vbox:~/Documentos$ █
```

Paso 5

Ejecuté el programa con punto con los números: 5, 10, 15, 20, 30, 40, 50, 100

```
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 5
.....El valor de fibonacci de 5 es 5
El calculo tardó: 0.000004 segundos
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 10
.....El valor de fibonacci de 10 es 55
El calculo tardó: 0.000005 segundos
nicole@vbox:~/Documentos$ ./lab_interrupciones
Ingrese un numero entero positivo: 15
```

17 de ago 15:08



nicole@vbox:~/Documentos
~/Documentos



```
.....El valor de fibonacci de 15  
es 610  
El calculo tardó: 0.000039 segundos  
nicole@vbox:~/Documentos$
```

17 de ago 15:08

nicole@vbox:~/Documentos
~/Documentos



```
65
El calculo tardó: 0.000557 segundos
nicole@vbox:~/Documentos$
```

.....El valor de fibonacci de 20 es 67

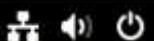
17 de ago 15:09

nicole@vbox:~/Documentos
~/Documentos



```
.....El valor de fibonacci de 3
0 es 832040
El calculo tardó: 0.050895 segundos
nicole@vbox:~/Documentos$ █
```

17 de ago 15:09



nicole@vbox:~/Documentos
~/Documentos



```
.El valor de fibonacci de 40 es 102334155
El calculo tardó: 6.193818 segundos
nicole@vbox:~/Documentos$
```

17 de ago 15:35



nicole@vbox:~/Documentos
~/Documentos



```
.....El valor de fibonacci de 50 es 12586269025  
El calculo tardó: 781.886037 segundos
```

nicole@vbox:~/Documentos\$ █

17 de ago 16:14

nicole@vbox:~/Documentos
~/Documentos



```
..... El valo
r de fibonacci de 51 es 20365011074
El calculo tardo: 1258.069343 segundos
nicole@vbox:~/Documentos$
```

17 de ago 17:08

nicole@vbox:~/Documentos
~/Documentos



```
...El valor de fibonacci de 52 es 32951280099
El calculo tardó: 2032.626511 segundos
nicole@vbox:~/Documentos$
```

```

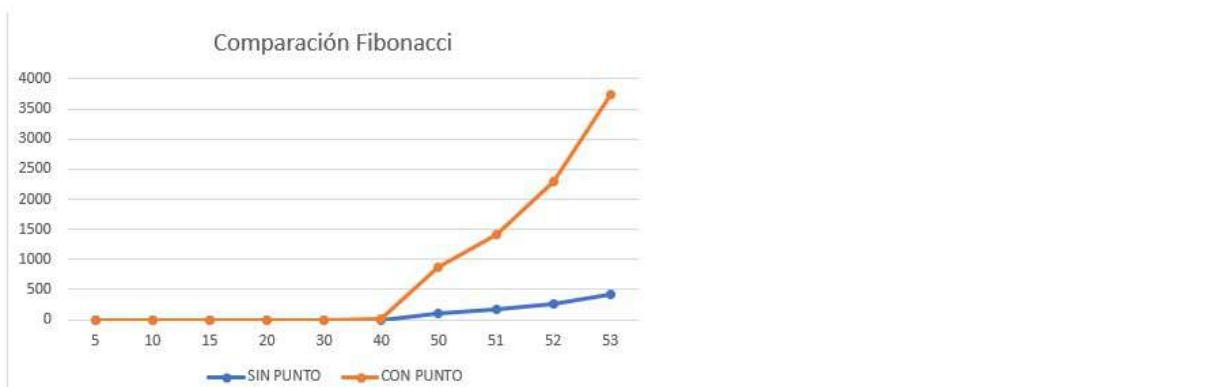
valor de fibonacci de 53 es 53316291173
El calculo tarde: 3302.513765 segundos
nicole@vbox:~/Documentos$ █

```

5 10 15 20 30 40 50 51 52 53

SIN PUNTO 0.000001 0.000002 0.000006 0.000055 0.006528 0.800849 99.167858 161.442491 260.153015 428.375605

CON PUNTO 0.000004 0.000005 0.000039 0.000557 0.050895 6.193818 781.886037 1258.069343 2032.626511 3302.513765



Tanto la curva con punto como la sin punto tienen un comportamiento exponencial en el tiempo dado un mayor número de Fibonacci, pero la curva con punto tiene un mayor crecimiento que empieza a tener una diferencia evidente a partir del Fibonacci de 50. Con esta gran diferencia podemos concluir que el print es más demandante en el tiempo que el resto de operaciones del código, con una muestra pequeña parece insignificante para los humanos, pero cuando se va acumulando podemos llegar a estos resultados.