

En este reporte se presentará sobre las prácticas hechas en clases, en las últimas sesiones hemos visto sobre Fibonacci y Primalidad.

Empezaremos con Fibonacci.

Fibonacci

¿Qué es?

La sucesión Fibonacci se describe como una sucesión infinita de números **naturales** que comienza con el número 1 y por consiguiente el próximo número es 1, y de ahí cada elemento se saca sumando los dos números anteriores.

Está definida por $(n-1) + (n-2)$

Tiene varias aplicaciones en teoría de juegos, ciencias de computación, entre otras.

Lo aplicaremos en códigos en Python.

Fibonacci Recursivo

La recursividad está definida en programación como un método, que consiste cuando se hace una orden a una función de su misma. Entonces quiere decir que se puede repetir continuamente.

Código:

```
cnt=0
```

```
def fibonacci(n):
```

```
    global cnt
```

```
    cnt+=1
```

```
    if n==0 or n==1:
```

```
        return(1)
```

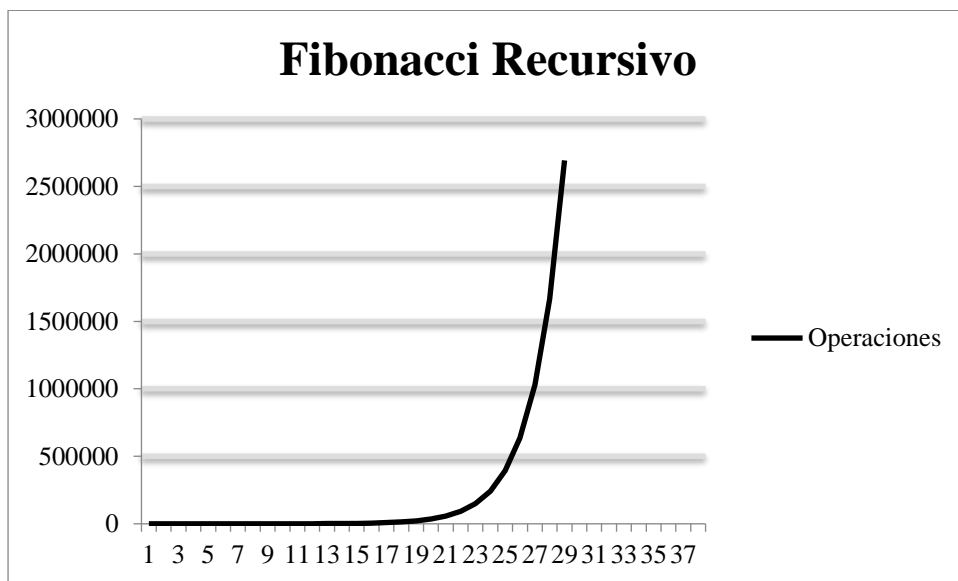
```
    else:
```

```
        return fibonacci(n-1) + fibonacci(n-2)
```

```
    for i in range(2, 30):
```

```
cnt=0
```

```
print(i, fibonacci(i), cnt)
```



Fibonacci con Memoria

Con la memoria es la forma como podemos almacenar contenido, como no se conoce el número de la variable, no se puede definir.

Código

```
arreglo={}
```

```
cnt=0
```

```
def fibonacci(n):
```

```
    global arreglo, cnt
```

```
    cnt+=1
```

```
    if n==0 or n==1:
```

```
        return(1)
```

```
    if n in arreglo:
```

```
        return arreglo[n]
```

```
    else:
```

```
        aux=fibonacci(n-1) + fibonacci(n-2)
```

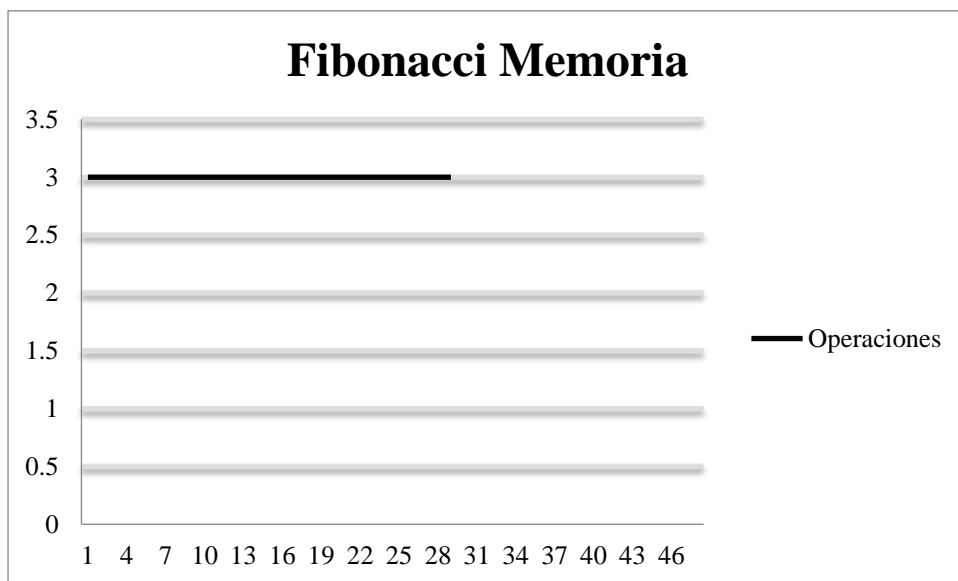
```
arreglo[n]=aux
```

```
return aux
```

```
for i in range(2, 30):
```

```
    cnt=0
```

```
    print(i, fibonacci(i), cnt)
```



Fibonacci Iterado

Iterado en programación se define como la acción de estar repitiendo un método hasta llegar al objetivo del algoritmo, cada repetición es la iteración. Entonces para esto se utiliza ciclos, en el código utilizamos 'for' como ciclo.

Código

```
def fibonacci(n):
```

```
    global cnt
```

```
    cnt=0
```

```
    if n==0 or n==1:
```

```
        return(1)
```

```
    r,r1,r2=0,1,1
```

```
for i in range(2,n+1):
```

```
    cnt+=1
```

```
    r=r1+r2
```

```
    r2=r1
```

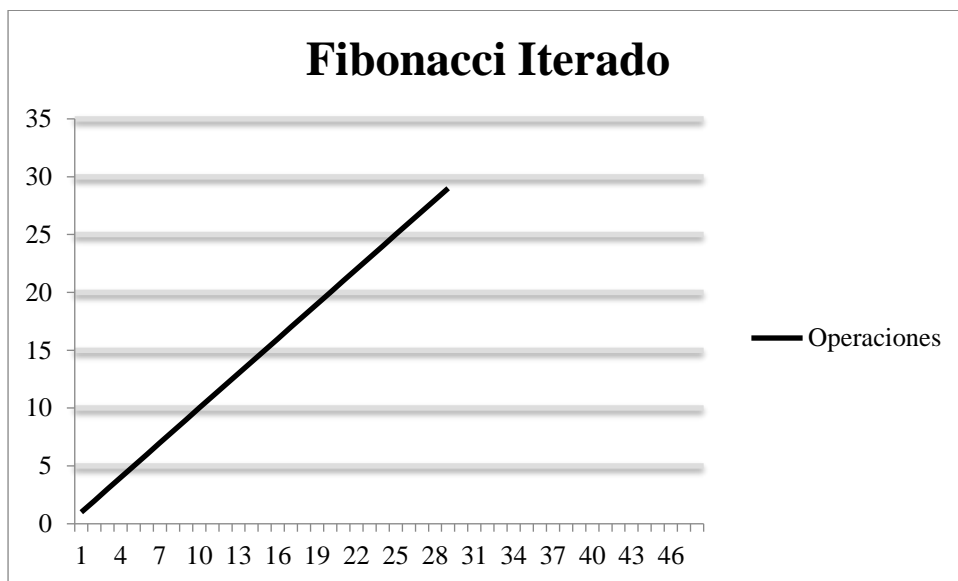
```
    r1=r
```

```
return r, cnt
```

```
for i in tange(2,30):
```

```
    cnt=0
```

```
    print(i, fibonacci(i), cnt)
```



Primos

Este código determina si un número es primo o no, por medio de ciclo 'for', primero pones el 'n' número, se saca la raíz de 'n' sumando 1 para después dividirlo por la raíz del número 'n', si es divisible "no es primo", si no es divisible "si es primo".

```
def primo(n):
```

```
    global cnt
```

```
    for i in range(2, round(n**(1/2)+1)):
```

```
    cnt=cnt+1  
    if ((n%i)==0):  
        return ("No primo")  
    return ("Es primo")  
for i in range (2,40):  
    cnt=0  
    primo(i)  
    print(i, cnt)
```

