

## Algoritmo para determinar si un número es primo.

¿Qué es?

Se sabe que para que un número sea primo solo es divisible entre 1 y entre sí mismo, además se sabe que si un número  $n$  no es divisible entre números primos menores o iguales a la raíz cuadrada de  $n$ , entonces  $n$  es primo.

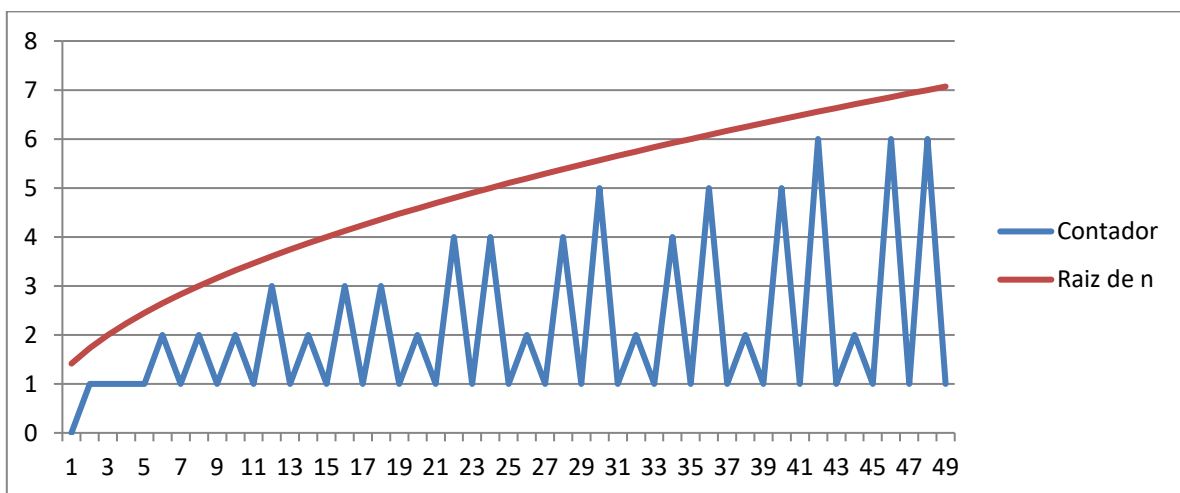
¿Cómo lo programamos?

Bueno definimos la función  $\text{primo}(n)$  usamos un ciclo for que va desde 2 hasta la raíz cuadrada de  $n$  y luego le sumamos 1 después se pone una condición para ver si el número es divisible entre un numero diferente de 1 y así determinar si es primo o no.

Seudocódigo

```
primo(n):  
    para cada i en un rango de (2 hasta  $n^{1/2}+1$ )  
        si  $n \% i = 0$ :  
            regresar (no es primo)|  
  
    regresar(si es primo)
```

Grafica para los números de 2 hasta 49.



## Algoritmo de fibonacci recursivo.

¿Qué es?

La sucesión de Fibonacci nos dice que el n-ésimo término de la sucesión es igual a la suma de los 2 términos anteriores de la sucesión. Los primeros términos de la sucesión son 1, 1, 2, 3, 5, 8, 13, 21 etc.

¿Cómo lo programamos?

Primero creamos una función, donde se tenga limite el n-ésimo termino que se quiere obtener como se sabe que en los términos 0 y 1 son iguales a 1 entonces siempre que se ingrese el valor de 0 y 1 debe ser igual a 1 y la función va a regresar el valor de 1, para esto se pondrá una condición para la verificación del término, ya que si n es diferente de 0 y 1 entonces el n-ésimo termino debe ser mayor que 2 y para esto usamos la ecuación  $\text{fibonacci}(n-2) + \text{fibonacci}(n-1)$ .

Seudocódigo.

```
contador=0
fibonacci(n):
    global contador
    contador+=1
    Si n=0 ó n=1:
        regresar (1)
    Si no:
        regresar fibonacci(n-2)+fibonacci(n-1)
```

Gráfica de las operaciones que realiza el algoritmo para los términos de 2 hasta 30.



## Algoritmo de Fibonacci iterativo.

¿Cómo lo programamos?

Primero creamos una función igual que en el anterior donde se tenga limite el n-ésimo termino que se quiere obtener como se sabe que en los términos 0 y 1 son iguales a 1 entonces siempre que se ingrese el valor de 0 y 1 debe ser igual a 1 y la función va a regresar el valor de 1, para esto se pondrá una condición para la verificación del término, pero ahora usaremos 3 variables donde la primer es igual a 0 y las otras dos variables sin igual a 1 y esto para usar la ecuación que usamos en el Fibonacci anterior luego las variables que están igualadas a es porque son los términos 0 y 1 luego se usa el ciclo for que va a iterar desde 2 hasta el término que buscamos .

Seudocódigo.

```
fibo(n):  
    contador=0  
    Si n=0 o n=1:  
        regresar(1)  
    v=0  
    v1=1  
    v2=1  
    Para i en el rango(2,n+1):  
        contador+=1  
        v=v1 + v2  
        v2=v1  
        v1=v  
    regresar v, contador
```

Gráfica con las operaciones que realiza el algoritmo desde los términos de 2 hasta 30.



## Algoritmo de Fibonacci con memoria

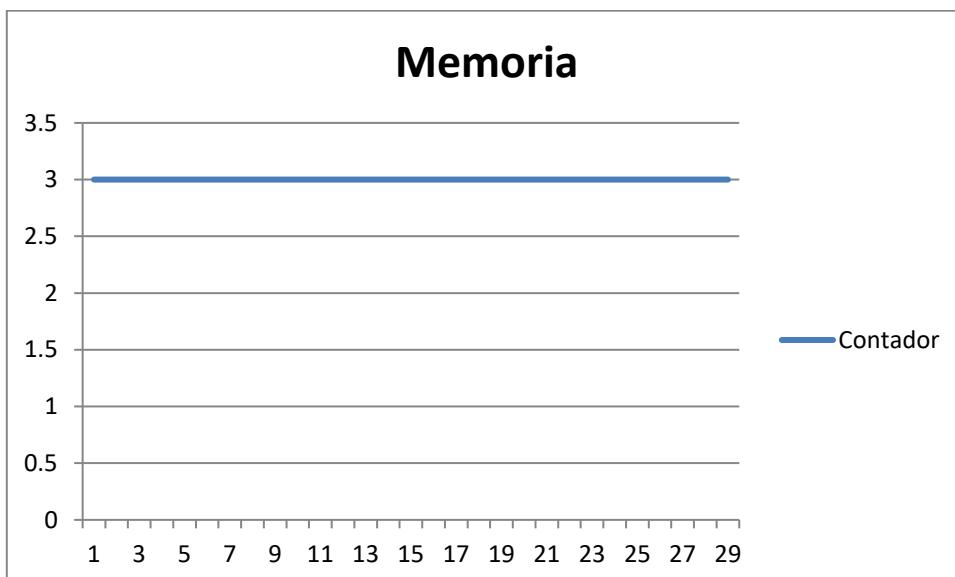
¿Cómo lo programamos?

Bueno en este algoritmo usamos un diccionario que este hace una reserva de espacio de memoria para un conjunto de datos. Mediante una condición se va a verificar que que el n-ésimo termino ya esté en el diccionario si ya esta va a regresar el valor del n-ésimo termino y si no esta hacemos uso de la ecuación  $\text{fibonacci}(n-2) + \text{fibonacci}(n-1)$  lo que nos ayuda en este algoritmo es que si el termino que pedimos ya se calculo anteriormente el valor va a salir directamente del diccionario gracias a que ya esta guardado.

Seudocódigo.

```
memo={}
contador=0
fibonacci(n):
    global memo, contador
    contador+=1
    Si n=0 o n=1:
        regresar(1)
    Si n esta en memo:
        regresat memo[n]
    Si no:
        Valor=fibonacci(n-2)+fibonacci(n-1)
        memo[n]=Valor
        regresar Valor
```

Gráfica con las operaciones que realiza el algoritmo desde los términos de 2 hasta 30 y nos damos cuenta que siempre hace las mismas operaciones y solo cambia cuando nosotros le damos un valor a n y los dos valores de los términos anteriores no estén en la memoria.



**Ruiz Fraser Francisco Guillermo 1837506**