

Nombre: Laila Romano

Procedimiento – Prueba 1

Primera parte: Las indicaciones dadas fueron

- **Definir número como entero:** Para definir un numero como entero en Python utilice la función **int**
- **Escribir “Ingrese el número”:** Para preguntar al usuario el número “que ingresara” utilice la función **input** la cual nos permite que el usuario digite el valor de la variable “n” (en este caso, nuestro número) y no tener que definir desde el principio del código el valor de este.
- **Condición 1:** Si es un número mayor que cero, para esto utilice como condicional **if** y luego tome como condición **0<**, para luego si este número cumple con este condicional le muestre al usuario “El número es positivo”.
- **Condición 2:** Si es un número menor a cero, para esta segunda condición utilice el condicional **elif** para no tener que utilizar **if- else** anidados y así obtener menos líneas de código (simple estética) dando como condición que **n<0** y si este cumple con la condición le muestre al usuario “El número es negativo.”
- **Condición 3:** Como tercera y última condición debemos probar si es número es cero, esto lo hice con el condicional **else** y simplemente este se ejecutará en caso de que el número no cumpla con las condiciones anteriores.

Segunda parte:

- **Validar si el número ingresado es primo:** Para esta condición, utilicé un bucle **while** que genera una secuencia a partir de una lista en donde la inicialicé con dos datos **[0,1]** los cuales son los casos bases de la función de recurrencia de Fibonacci como no se generan a partir de la función debemos tenerlos en cuenta siempre para validar si el número pertenece al conjunto de números de Fibonacci. Siendo así, mientras el último elemento de la lista **caso_base[-1]** sea menor al numero que ingreso el usuario, el programa calcula el siguiente valor sumando los dos últimos datos que se encuentran en la lista principal de casos base, que también nos esta sirviendo como una lista acumuladora de números generados por la función de Fibonacci **casos_base[-1] + casos_base[-2]**. Para finalizar y así verificar que el numero ingresado si cumple, seria una vez terminada la secuencia alcanza o supera el numero ingresa por el usuario, utilizando un condicional **if** junto con **in** para verificar que dentro de la lista que se genero en la secuencia se encuentra el numero ingresado, si es así, le mostrara al usuario “Es un número Fibonacci.”
- **Validar si el valor ingresado pertenece al conjunto de números Fibonacci:** Para validar si el numero es primo de manera eficiente implemente un bucle **while** que busca divisores. Como filtro inicial incluí condiciones para descartar números menos o iguales a 1, ya que por definición estos no cumplen con ser números primos. Luego de esto el bucle **while** solo busca divisores hasta la raíz cuadrada del número **divisor*divisor*<=n** haciendo así que se reduzcan el numero de iteraciones que hará el bucle, solo llegara hasta donde los divisores sean dos y enseguida dirá que es un numero primo, ya que así sabríamos que ese numero es divisible por si mismo y por 1. Como resultado utilicé una variable **es_primo** como verdadero, esta solo cambiara si al momento de la división el residuo es cero **n%divisor == 0**, haciendo que cambie a falso e indicando que ya no cumple con ser primo. Si este valor de verdad no cambia, el usuario tendrá como mensaje en la pantalla “El número es primo.”

