

## Procedimiento – Programa 1

### Primera parte:

Se nos dio una serie de instrucciones para hacer un programa en Python que determinara si el número ingresado era positivo, negativo o cero. Para ello, se realizó el siguiente procedimiento:

- **Definir número:** Se define el módulo del programa como `def num_enteros()` para poder ejecutar el programa.
- **Ingresar número:** Mediante las funciones `input` e `int`, se guarda el número ingresado en una variable, en este caso `num`.
- **Definir si es positivo, negativo o cero:** Mediante las estructuras `if`, `elif` y `else`, se establecen las condiciones si el número es menor que 0, mayor que 0 o igual a 0, y posteriormente se muestra en la pantalla del usuario si es negativo, positivo o igual a 0, respectivamente.
- **Final:** Se finaliza el programa.

### Parte dos:

Después de desarrollar la primera parte, se nos dieron dos instrucciones adicionales: determinar si el número ingresado era un número primo y si pertenecía al conjunto de números de Fibonacci.

- **Fibonacci:**
  1. **Definiciones:** Mediante una lista se agregan los casos base de Fibonacci, en este caso 0 y 1. Si el número ingresado es 0 o 1, automáticamente se determina que sí pertenece a la sucesión mediante una variable booleana igualada a `true`.
  2. **Desarrollo:** Mediante un ciclo `while`, se verifica si el último número de la lista es menor que el número ingresado. Al sumar los dos últimos números de la lista, se obtiene y agrega el siguiente número de la sucesión.
  3. **Final:** Se determina si el número ingresado es igual al último número de la lista mediante un `if`. Si existe coincidencia, con un `print` se indica que el número pertenece a la sucesión de Fibonacci; de lo contrario, el programa continúa.
- **Primos:**
  1. **Definiciones:** Se define una lista vacía para los divisores del número y se determina que el número es primo mediante una variable llamada `primo`, la cual se iguala a `true`.
  2. **Determinación del número:** Si el número es menor que 2, por definición no es primo, por lo que la variable `primo` se iguala a `false`. En caso contrario, se determina un límite mediante la raíz entera del número más 1 para, seguidamente, realizar un bucle `for` con un rango desde 1 hasta dicho límite y verificar si el número es divisible entre los valores del rango. Si lo es, estos se agregan a la lista de divisores.
  3. **Final:** Mediante un `if` y la función `len()`, se determina la longitud de la lista de divisores. Si esta es mayor que 2, la variable `primo` se iguala a `false`. A continuación, con otro `if`, si la variable `primo` es `true`, se indica que el número es primo.