
LENGUAJES ELECTRÓNICOS

TP 2

DIAGRAMAS DE FLUJO

AUTORES:

GALVAN ABBONDANZA THIAGO

DOBLADEZ MATEO

PROFESOR:

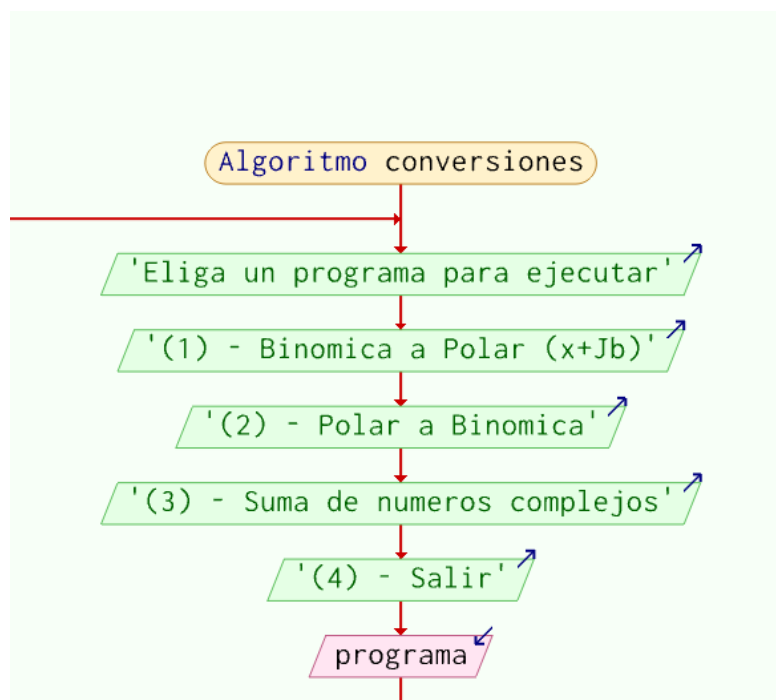
PAVELEK ISRAEL

ACTIVIDAD 1,2,3	2
ACTIVIDAD 4 Y 5.....	7
CONCLUSIONES.....	12

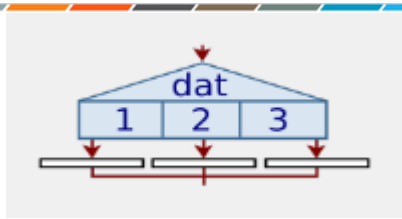
ACTIVIDAD 1,2,3

1. Escribir un programa que calcule el módulo de un número complejo y su ángulo (fase) ingresando su parte real y su parte imaginaria.
2. Escribir un programa que devuelva el número complejo en formato rectangular (su parte real + j su parte imaginaria), ingresando su módulo y ángulo (fase).
3. Escribir un programa que sume dos números complejos en formato rectangular y pregunte al final como desea ver el resultado si en polar o rectangular y lo muestre en el formato que el usuario quiera.

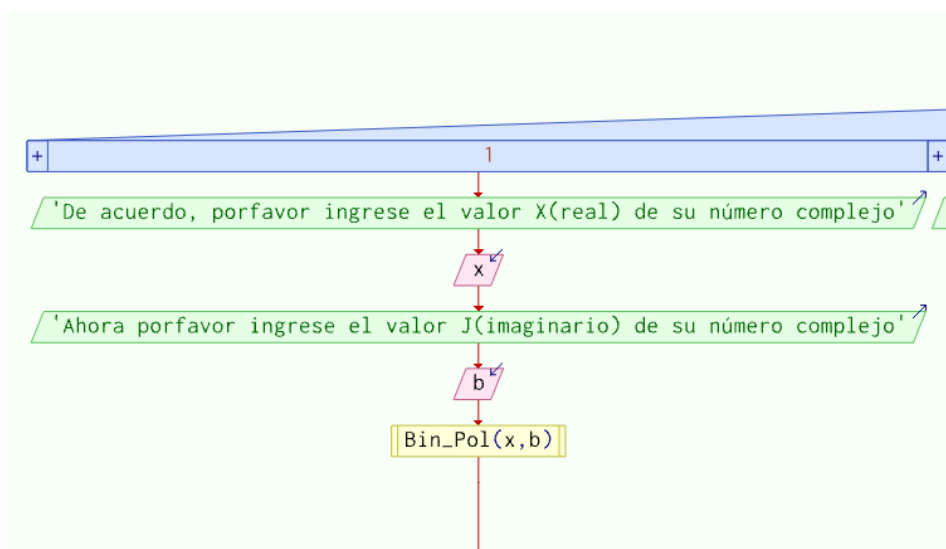
Para el punto 1 arrancamos pidiéndole al usuario que elija entre unos de los 3 programas (1 por punto). Para estos usamos los cuadros verdes que sirven para imprimir texto, entonces el usuario va a poder entender cómo va a funcionar el programa y lo que tiene que ir haciendo.



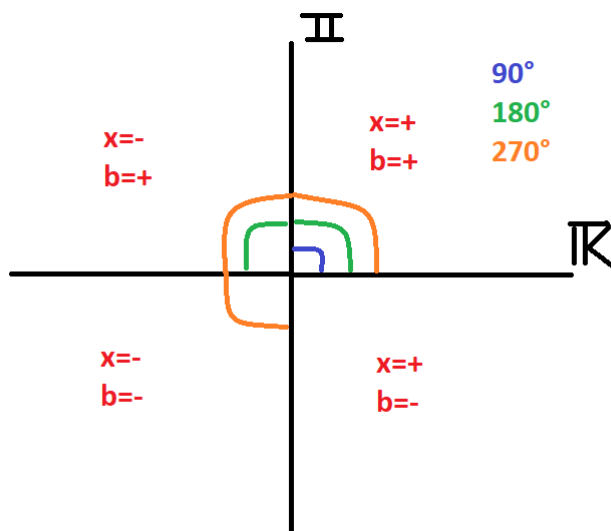
Luego con esta estructura podemos determinar qué fue lo que eligió el usuario y a partir de esa respuesta saber que programa utilizar. Usando como dato la variable “programa” que definimos en previamente (cuadro rosa).



Si el usuario elige el **primer programa (conversión binómica a polar)** le pedimos que ingrese la parte real de su número complejo y la guardamos en la variable "x" debido a que los reales los ubicamos en el eje x. También le pedimos la parte imaginaria y la guardamos en la variable "b". La cual mostramos en la elección del programa que se iba a denominar de esta forma

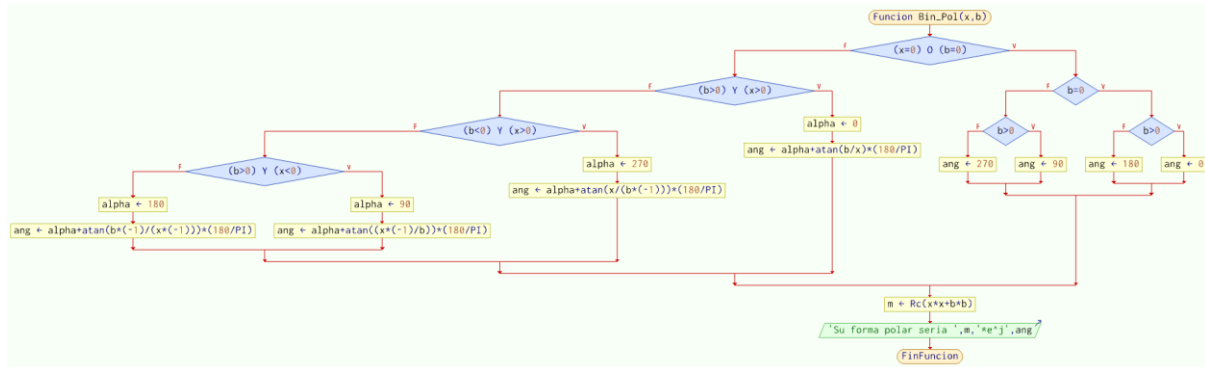


Para hacer la conversión usamos una función para poder tener el diagrama principal mas limpio y organizado. También tenemos que determinar en qué cuadrante se va a encontrar el módulo. Para esto tenemos que ver el signo de la parte real y la imaginaria

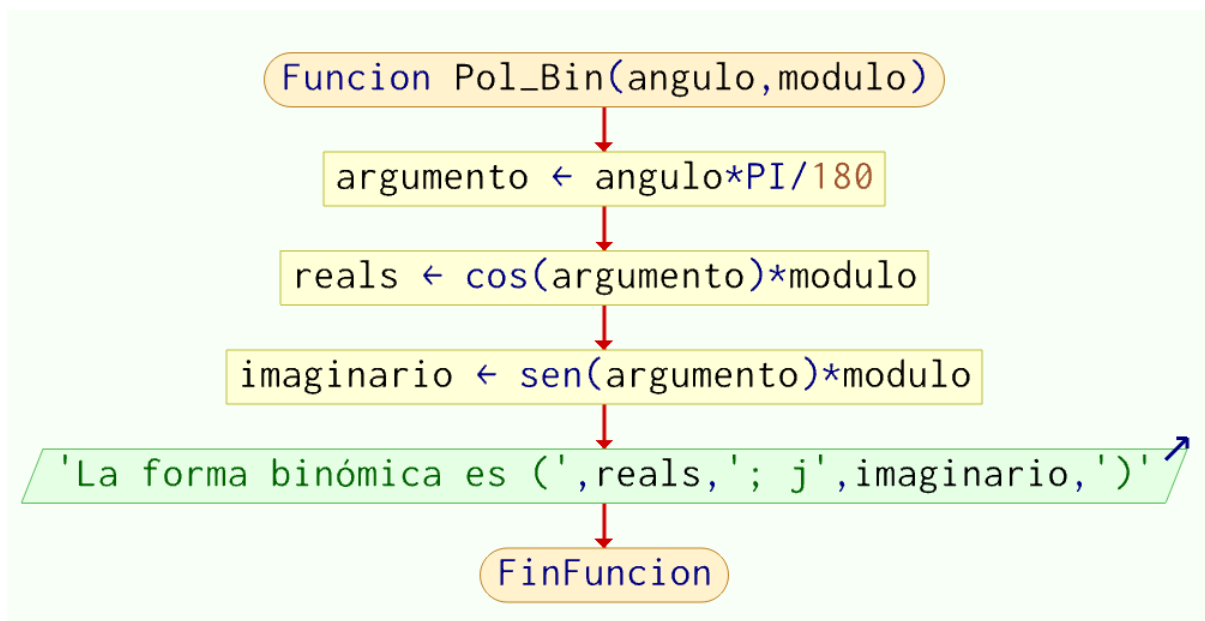


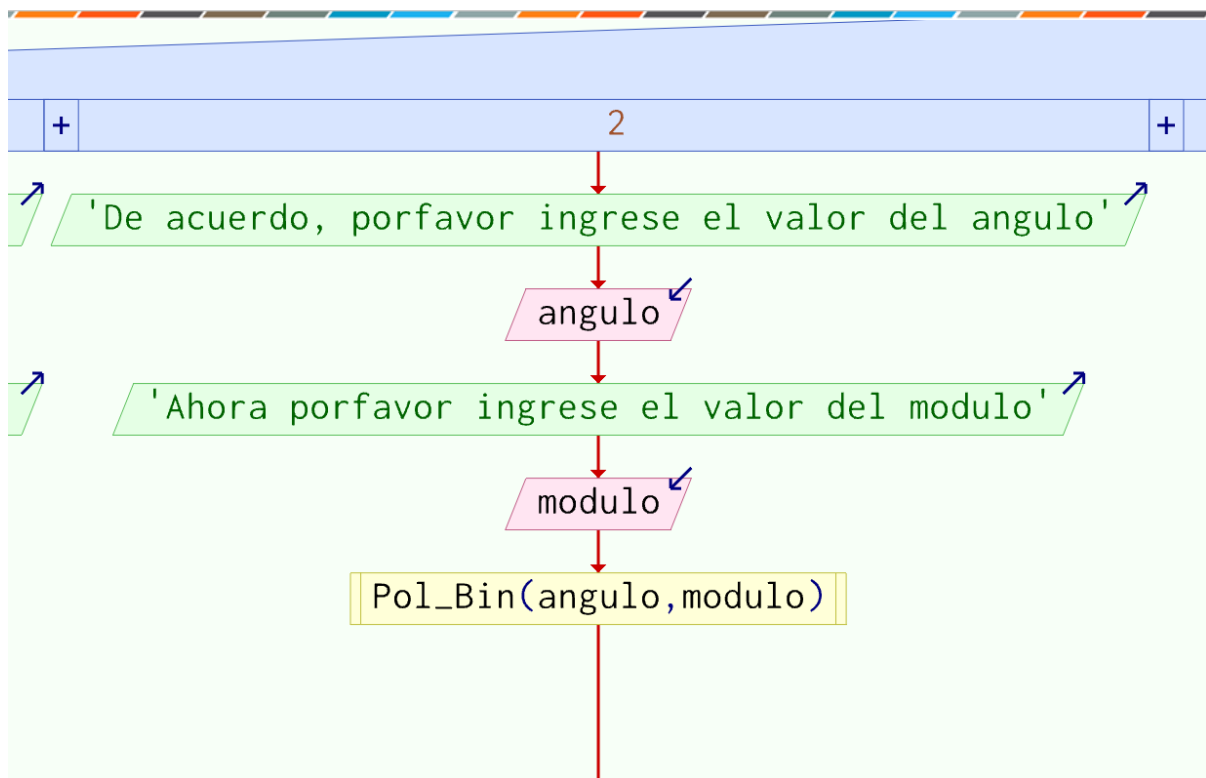
Una vez determinado esto calculamos el ángulo por trigonometría usando arco tangente.

Y para calcular el módulo usamos teorema de Pitágoras.



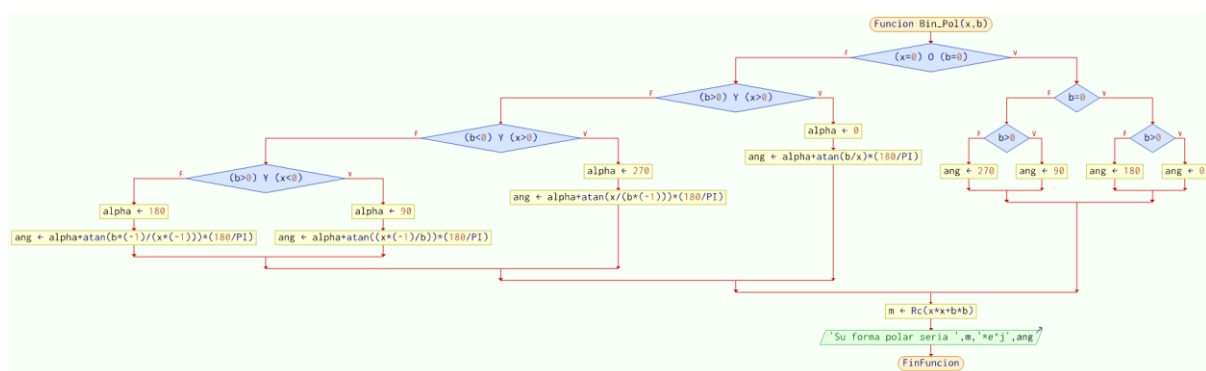
Si el usuario elige el **segundo programa(conversion polar a binomica)**, le pedimos que ingrese el valor del ángulo y módulo de su número complejo. Y luego usamos nuevamente una función en la cual hacemos la conversión de polar a binómica. Antes que nada necesitamos convertir el valor del ángulo que nos dio el usuario y convertirlo a radianes ya que es la unidad que utiliza Pseint para los cálculos. Luego simplemente por trigonometría calculamos la parte real y la parte imaginaria.



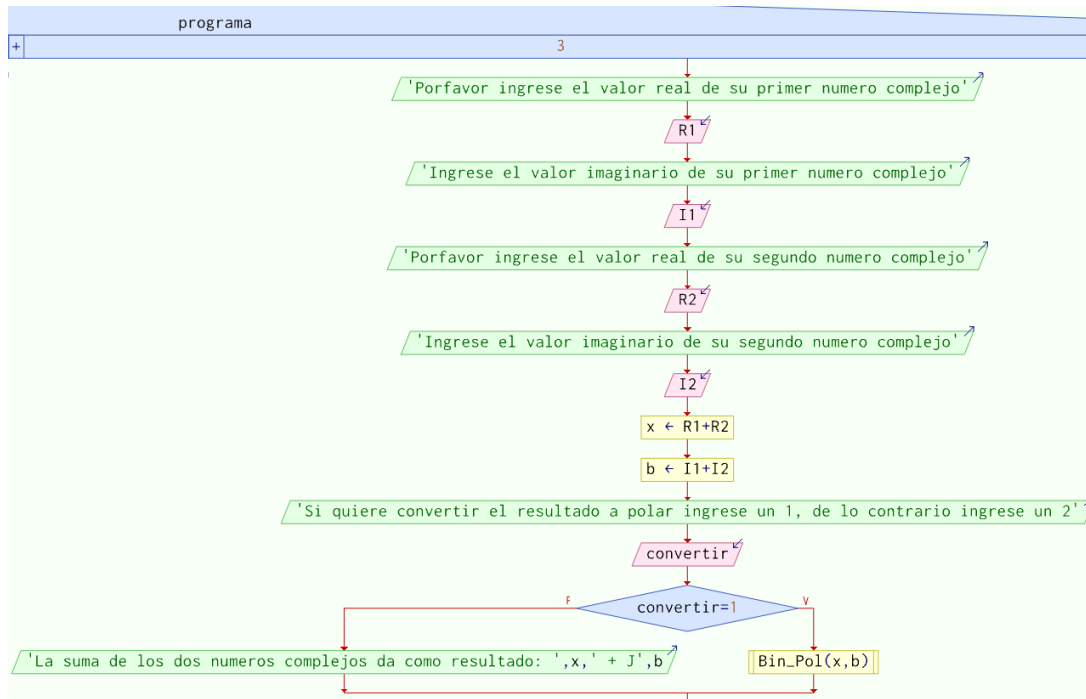


Si el seleccionado es el **tercero (suma de numeros complejos)** le pedimos al usuario que ingrese los valores imaginarios y reales de ambos números complejos a sumar. Luego hacemos la suma, parte real con parte real y parte imaginaria con parte imaginaria. Una vez hecha la suma le preguntamos al usuario de que forma quiere visualizar su numero complejo, polar o binomica.

Si desea verlo en forma polar tenemos que usar una de las funciones que hicimos previamente en la actividad 1. La funcion es Bin_Pol(x,b)



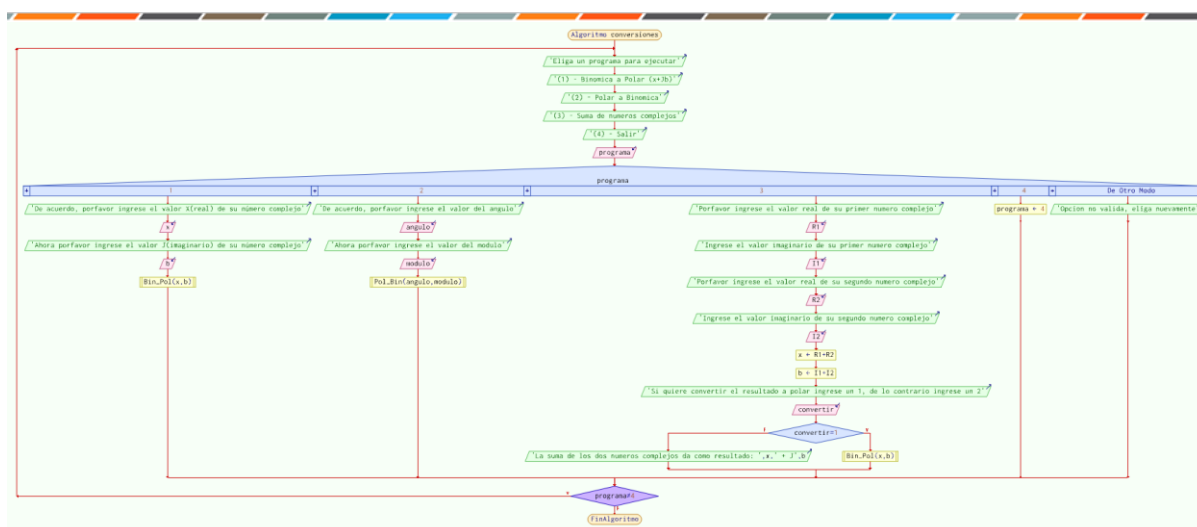
Como antes realizamos la Funcion $\text{Bin_Pol}(x, b)$, vamos a volver a utilizarla tan solo escribiendo el nombre de la funcion y la agregamos con bloque de V o F donde antes preguntamos al usuario si desea hacer el pasaje o dejarlo en forma binomica



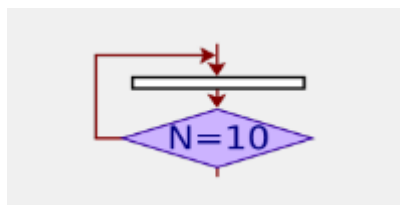
Como ya sabemos, para realizar una suma de dos números complejos es simplemente sumar sus dos partes reales o Resistencia en caso de calcular impedancia y sumar sus partes imaginarias o reactancias en caso de impedancias.

Luego de esto asignamos como “x” a la suma de sus partes reales y “b” a la suma de sus partes imaginarias para finalmente preguntar si se desea realizar el pasaje a forma polar y escribir el resultado en pantalla en la forma deseada.

Aunque no lo pedia el ejercicio, agregamos algunos detalles más. Por ejemplo, si el usuario quiere salir del programa puede apretar 4, y si por alguna razon no pone uno de los valores proporcionados, se le pedira de vuelta ingresar su elección.



Para eso usamos la estructura **repetir-mientras**, entonces la condicion que le pusimos es que repita mientras la variable programa $\neq 4$ por lo que cuando el usuario ingrese 4, el programa finalizara.

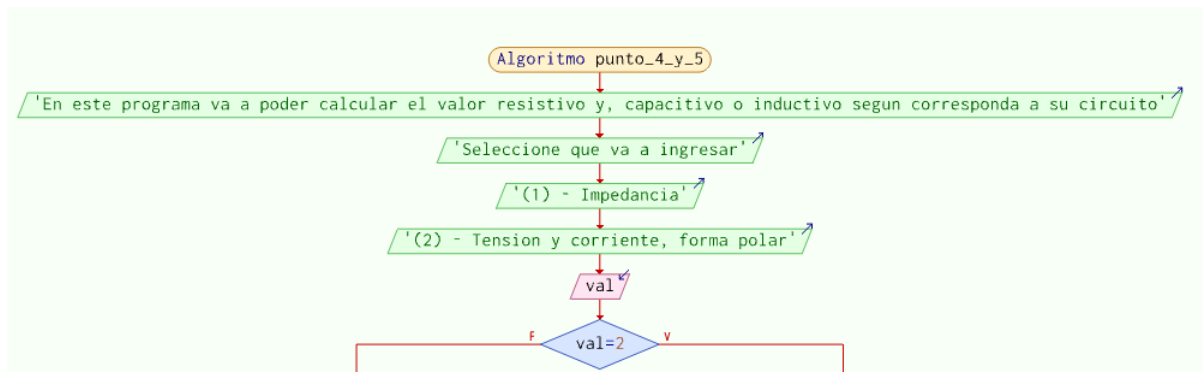


ACTIVIDAD 4 Y 5

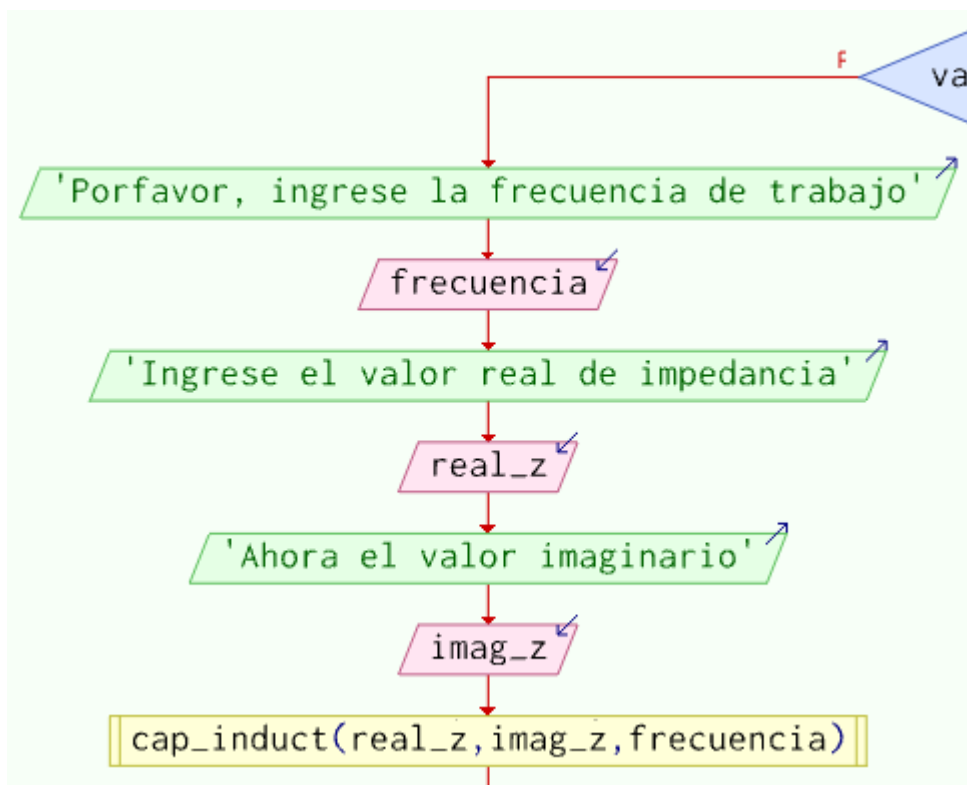
4. Escribir un programa que al recibir una impedancia (parte real y parte imaginaria) y la frecuencia de trabajo, indique el valor de la resistencia y el capacitor o inductor según corresponda.
5. Escribir un programa que ingresada una tensión y una corriente (con su modulo y ángulo) y la frecuencia de trabajo, informe el valor de la impedancia (parte real y parte imaginaria, y luego el valor de los componentes asociados, resistor y capacitor o inductor según corresponda. Para este programa deberá utilizar lo realizado en el punto anterior como subprograma.

Ambas actividades son muy parecidas, con la diferencia que en el programa 1 lo que vamos a ingresar es una impedancia con su debida frecuencia y en el segundo programa lo que vamos a hacer es primero calcular la impedancia con los valores de tensión y corriente proporcionados por el usuario.

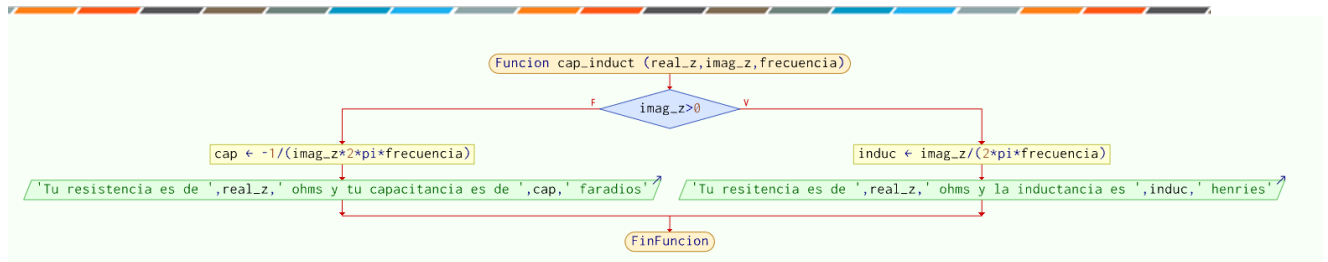
Antes que nada, preguntamos que valores se van a ingresar, si una impedancia o una tensión y una corriente. Cada uno correspondiendo a cada actividad, 4 o 5.



Si elige ingresar una impedancia, le pedimos que ingrese la parte real e imaginaria de la misma. También le pedimos la frecuencia de trabajo. Luego usamos una función para poder aclarar el código.



Dentro de la Funcion lo que tenemos es el siguiente código:



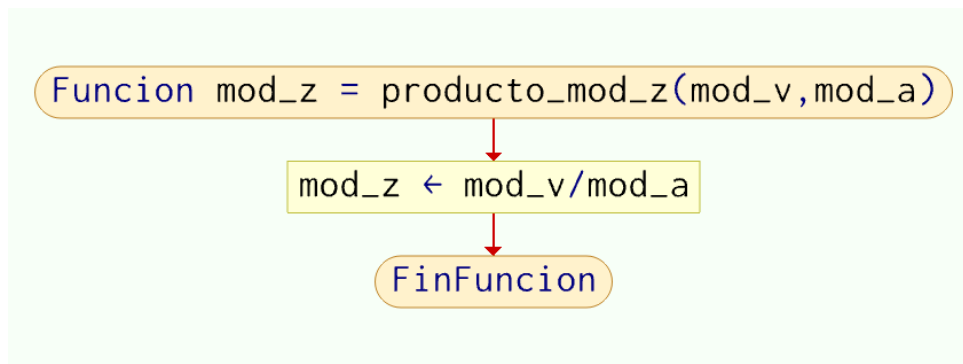
Lo que hacemos dentro de la función es evaluar primero si el valor de la reactancia (imaginario) de la impedancia es negativo o positivo ya que esto nos dirá si se trata de un capacitor o de un inductor.

Si el valor es mayor a 0 sabemos que se trata de un inductor. La fórmula para calcular la impedancia de un inductor es $Z_L = j\omega L$ siendo ω =omega y L =inductancia. Pero la inductancia ya la tenemos y lo que necesitamos calcular es la inductancia, por lo que despejando llegamos que $L = Z_L / \omega$. Vale aclarar que $\omega = 2\pi \cdot f$

En cambio, si el valor de la reactancia es menor a uno se trataría de un capacitor y por ende la cuenta es diferente. La fórmula de impedancia de un capacitor es $Z_C = -j/\omega C$, lo que necesitamos es C =Capacitancia, entonces despejando llegamos a que $C = -1/\omega Z_C$.

Luego de cada caso se imprime la impedancia, la parte real va a ser la resistencia y lo calculado, capacitancia(faradios) o inductancia(henries) según corresponda.

Si el usuario elige ingresar una tensión y corriente en forma polar, le pedimos los módulo y ángulos de cada uno respectivamente. También la frecuencia de trabajo. Como sabemos, $V/I = Z$ entonces eso es lo que vamos a hacer, dividir ambos fasores. Como sabemos, para hacer esto, dividimos los módulo y restamos los ángulos. Esto es lo que hicimos en las funciones.



En esta calculamos el módulo de la impedancia, haciendo la división de ambos módulo de tensión y corriente. Y además hacemos que la función nos de un valor como salida, en este caso `mod_z` para luego poder usarlo en otras cuentas. Para hacerlo es muy simple, ponemos el nombre del valor que queremos sacar, un `=` y nuestra función. Esto mismo hicimos en la función donde calculamos el ángulo de la impedancia.

```
Funcion ang_z = producto_ang_z(ang_v,ang_a)
```

```
ang_z ← ang_v-ang_a
```

```
FinFuncion
```

En esta función restamos ambos angulos y tenemos como salida a ang_z, que seria el ángulo de la impedancia.

Una vez conseguidos el modulo y angulo (forma polar) de la impedancia, tenemos que hacer la conversión para tenerla en forma rectangular y asi poder calcular el valor de la resistencia y capacitancia/inductancia. Como hicimos en el punto 4.

Para esto usamos los valores conseguidos previamente y por trigonometria calculamos.

Como pseint trabaja con radianes hacemos la conversion y calculamos la parte real usando coseno. Esto es lo que usamos en la funcion polar_binomica_real. Y tenemos como valor de salida, real_z que la utilizaremos para mas tarde determinar los valores de los componente.

```
Funcion real_z = polar_binomica_real(mod_z,ang_z)
```

```
argumento ← ang_z*pi/180
```

```
real_z ← cos(argumento)*mod_z
```

```
FinFuncion
```

Para calcular la parte imaginaria hacemos lo mismo que la real. Pasamos los grados a radianes y en este caso calculamos la coordenada imaginaria con seno. Tambien vamos a tener una salida, que la llamamos imag_z.

```
Funcion imag_z = polar_binomica_imag(mod_z,ang_z)
```

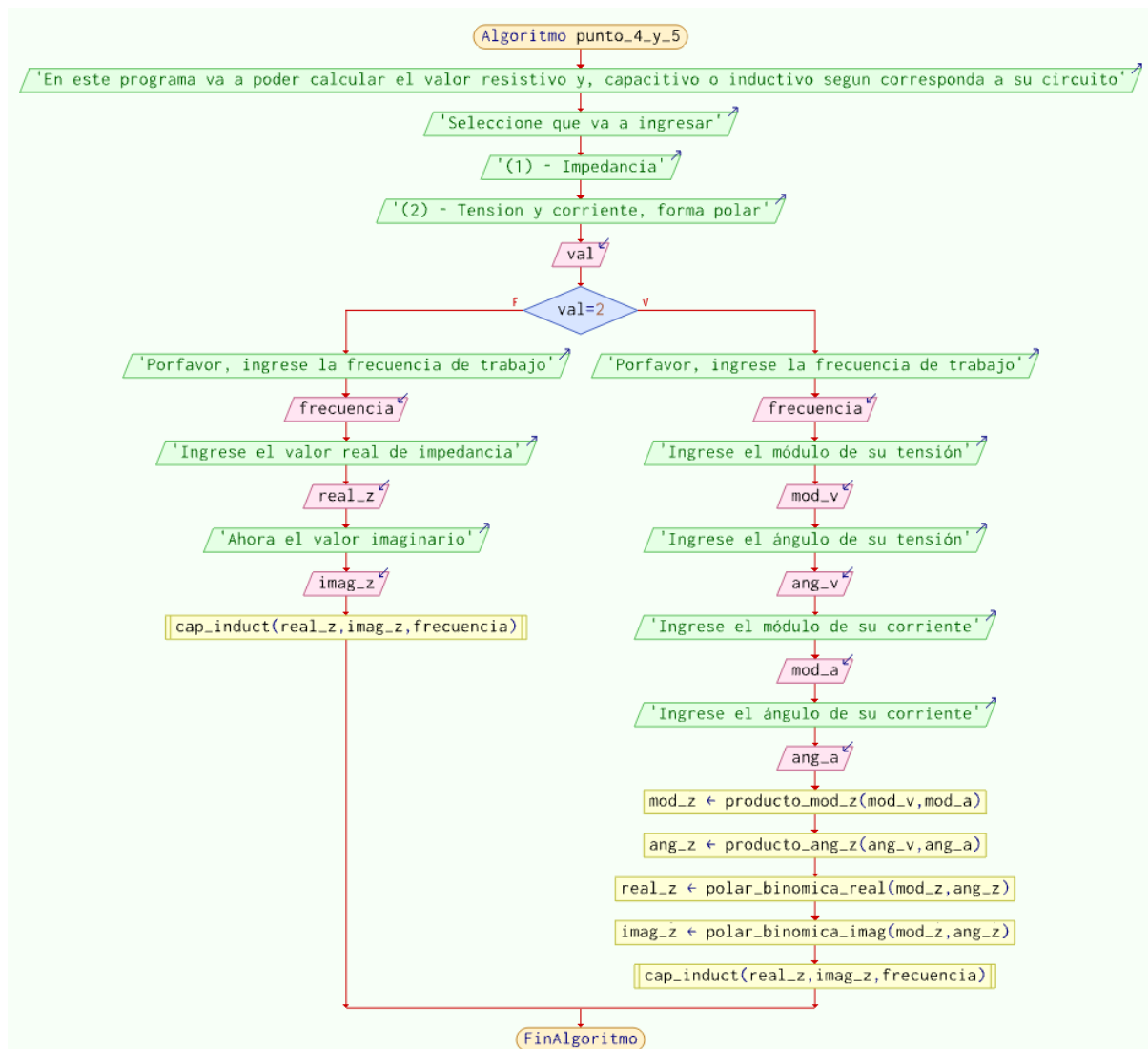
```
argumento ← ang_z*(pi/180)
```

```
imag_z ← sen(argumento)*mod_z
```

```
FinFuncion
```

Ahora que calculamos ambas coordenadas tenemos que determinar los valores de los componentes, a partir de acá, lo que tenemos que hacer es lo mismo del punto 4. Debido que tenemos una impedancia en forma rectangular. Por lo que usamos la misma funcion que previamente utilizamos, `cap_induct(real_z,imag_z,frecuencia)` . Su funcionamiento ya lo explicamos en el punto 4.

Al final, nuestro diagrama de flujo nos quedó así, consideramos que es bastante compacto y simple de entender.



CONCLUSIONES

Pudimos entender más a profundidad el funcionamiento de pseint y los diagramas de flujo. También la gran cantidad de cosas que se pueden crear a partir de la programación. Consideramos que esta aplicación nos fue muy útil para poder entender de forma visual el funcionamiento de la programación.