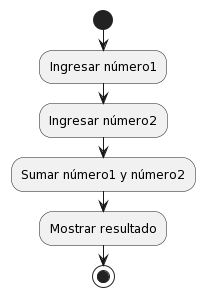
* **Algoritmo para sumar dos números:**



1. **Inicio:** Este es el punto de inicio del algoritmo.

2. **Ingresar número 1:** se solicita al usuario que ingrese el primer número.

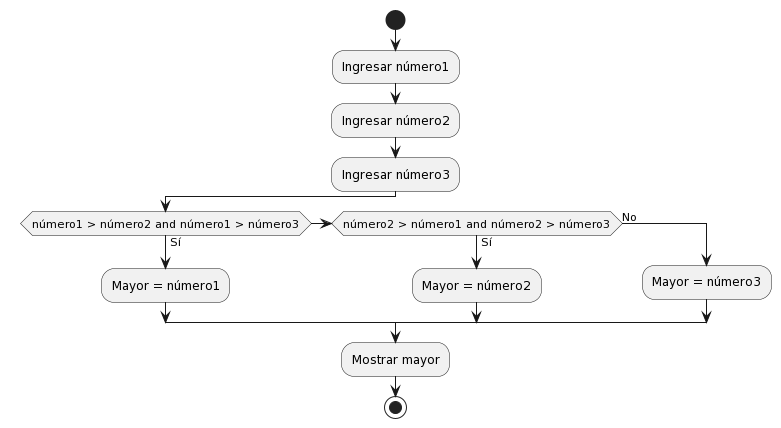
3. **Ingresar número 2:** el segundo número que se desea sumar.

4. **Sumar número1 y número2:** se suma el valor del primer número ingresado (número1) con el valor del segundo número ingresado (número2). Este paso realiza la operación de suma.

5. **Mostrar resultado:** Después de realizar la suma, el resultado se muestra al usuario.

6. **Fin:** Este es el punto final del algoritmo. Después de mostrar el resultado.

* **Algoritmo para encontrar el mayor de tres números:**



1. **Inicio:** El flujo comienza en el inicio del diagrama.

2. **Ingresar número 1:** El usuario ingresa el primer número.

3. **Ingresar número 2:** El usuario ingresa el segundo número.

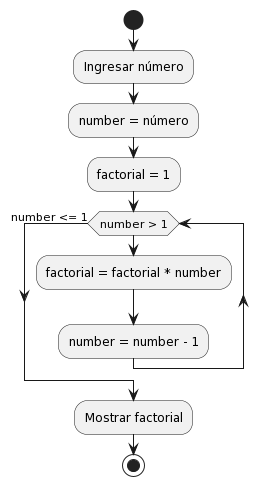
4. **Ingresar número 3:** El usuario ingresa el tercer número.

5. **Comparación:** Se realiza una serie de comparaciones para determinar cuál de los tres números ingresados es el mayor.

6. **Mayor:** Si el primer número es mayor que el segundo y el tercero, se asigna el valor del primer número a la variable "Mayor". Si el segundo número es mayor que el primero y el tercero, se asigna el valor del segundo número a la variable "Mayor". En cualquier otro caso, se asigna el valor del tercer número a la variable "Mayor".

7. **Mostrar mayor:** Se muestra en pantalla el valor del número mayor encontrado.

8. **Fin:** El flujo termina aquí.

* **Algoritmo para calcular el factorial de un número:**

1. **Inicio:** El flujo comienza en este punto.

2. **Ingresar número:** En esta etapa, el usuario ingresa el número del cual se desea calcular la factorial.

3. **number = número:** Se asigna el valor del número ingresado a una variable llamada "number". Esta variable se usará para realizar el cálculo del factorial.

4. **factorial = 1:** Se inicializa una variable llamada "factorial" con el valor de 1. Esta variable se usará para almacenar el resultado del factorial.

5. **while (number > 1):** Se inicia un bucle while que continuará mientras el valor de "number" sea mayor que 1. Este bucle se utilizará para iterar desde el número dado hasta 1.

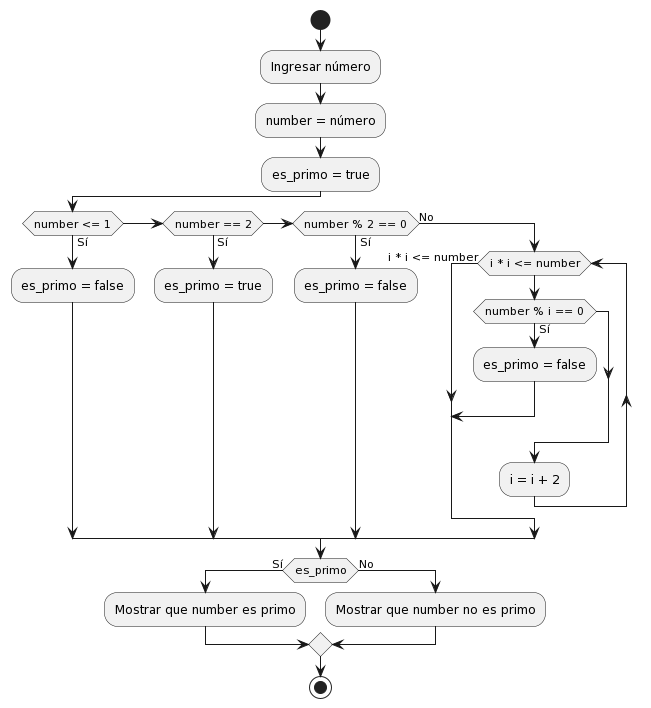
6. **factorial = factorial \* number:** En cada iteración del bucle, se multiplica el valor actual de "factorial" por el valor actual de "number". Esto se hace para calcular el factorial del número.

7. **number = number - 1:** Después de cada multiplicación, el valor de "number" se decrementa en 1 para continuar con la iteración hasta que llegue a 1.

8. **endwhile (number <= 1):** Cuando el valor de "number" llega a 1 o menos, el bucle while termina y se procede al siguiente paso.

9. **Mostrar factorial:** Una vez que se ha calculado el factorial, se muestra el resultado al usuario.

10. **Fin:** El flujo del programa termina aquí.

* **Algoritmo para verificar si un número es primo:**

1. **Inicio:** El flujo comienza en este punto.

2. **Ingresar número:** ingresa el número que se desea verificar si es primo.

3. **number = número:** Se asigna el valor del número ingresado a una variable llamada "number". Esta variable se usará para realizar las verificaciones.

4. **es\_primo = true:** Se inicializa una variable booleana llamada "es\_primo" con el valor verdadero. Esta variable se utilizará para determinar si el número es primo o no.

5. **if (number <= 1):** Se verifica si el número es menor o igual a 1. Si lo es, se asigna el valor falso a la variable "es\_primo", ya que los números menores o iguales a 1 no son primos.

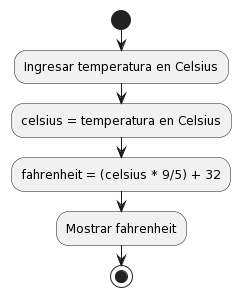
6. **elseif (number == 2):** Se verifica si el número es igual a 2. Si lo es, se asigna el valor verdadero a la variable "es\_primo", ya que el 2 es el único número par primo.

7. **elseif (number % 2 == 0):** Se verifica si el número es divisible por 2. Si lo es, se asigna el valor falso a la variable "es\_primo", ya que ningún otro número par es primo.

8. **else:** Si no se cumple ninguna de las condiciones anteriores, se pasa a este bloque. Aquí se lleva a cabo un bucle while que itera desde 3 hasta la raíz cuadrada del número, verificando si el número es divisible por algún otro número. Si encuentra un divisor, se asigna el valor falso a la variable "es\_primo" y se sale del bucle.

9. i**f (es\_primo):** Después de realizar todas las verificaciones, se verifica el valor de la variable "es\_primo". Si es verdadero, se muestra que el número es primo. Si es falso, se muestra que el número no es primo.

10. **Fin:** El flujo del programa termina aquí.

* **Algoritmo para convertir grados Celsius a Fahrenheit:**

1. **Inicio:** El flujo comienza en este punto.

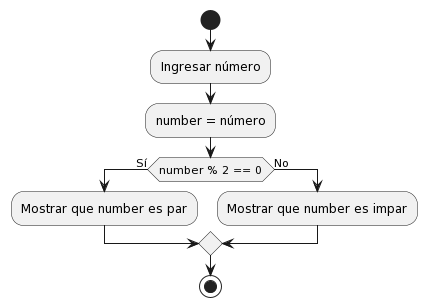
2. **Ingresar temperatura en Celsius:** En esta etapa, el usuario ingresa la temperatura en grados Celsius que se desea convertir a Fahrenheit.

3. **celsius = temperatura en Celsius**: Se asigna el valor de la temperatura en Celsius ingresada por el usuario a una variable llamada "celsius". Esta variable se utilizará para realizar el cálculo de conversión.

4. **fahrenheit = (celsius \* 9/5) + 32:** Se realiza el cálculo de conversión de grados Celsius a Fahrenheit utilizando la fórmula de conversión correspondiente: Fahrenheit = (Celsius \* 9/5) + 32.

5. **Mostrar fahrenheit:** Después de calcular la temperatura en Fahrenheit, se muestra el resultado al usuario.

6. **Fin:** El flujo del programa termina aquí.

* **Algoritmo para determinar si un número es par o impar: **

1. **Inicio:** El flujo comienza en este punto.

2. **Ingresar número:** En esta etapa, el usuario ingresa el número que se desea verificar si es par o impar.

3. **Número = número:** Se asigna el valor del número ingresado por el usuario a una variable llamada "number". Esta variable se utilizará para realizar la verificación.

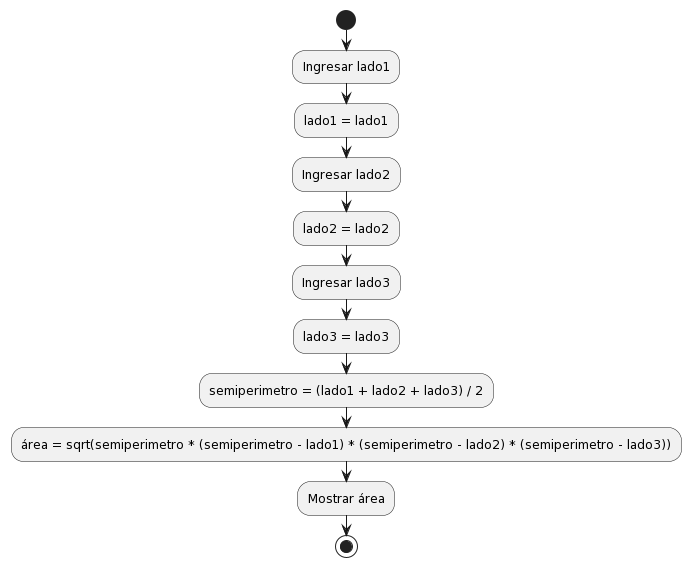
4. **if (number % 2 == 0):** Se verifica si el número es divisible por 2 utilizando el operador módulo (%). Si el residuo de la división es igual a 0, significa que el número es par.

5. **Mostrar que number es par:** Si el número es par (el residuo de la división por 2 es 0), se muestra un mensaje indicando que el número es par.

6. **else:** Si el residuo de la división no es igual a 0 (es decir, no es divisible por 2), se pasa a este bloque.

7. **Mostrar que number es impar:** En este caso, se muestra un mensaje indicando que el número es impar.

8. **Fin:** El flujo del programa termina aquí.

* **Algoritmo para calcular el área de un triángulo:**

1. **Inicio:** El flujo comienza en este punto.

2. **Ingresar lado1:** En esta etapa, el usuario ingresa la longitud del primer lado del triángulo.

3. **lado1 = lado1:** Se asigna el valor del primer lado ingresado por el usuario a una variable llamada "lado1". Este paso se realiza para almacenar la longitud del primer lado del triángulo.

4. **Ingresar lado2:** El usuario ingresa la longitud del segundo lado del triángulo.

5. **lado2 = lado2:** Se asigna el valor del segundo lado ingresado por el usuario a una variable llamada "lado2". Este paso se realiza para almacenar la longitud del segundo lado del triángulo.

6. **Ingresar lado3:** El usuario ingresa la longitud del tercer lado del triángulo.

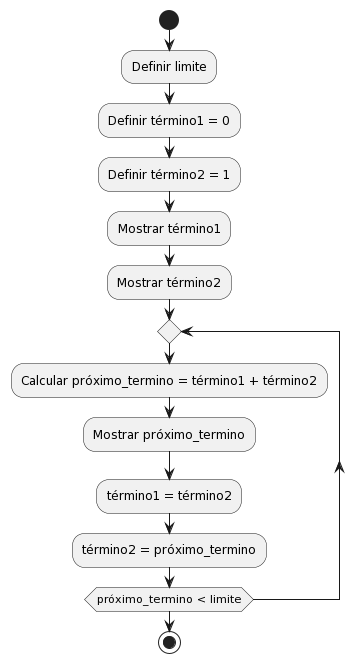
7. **lado3 = lado3:** Se asigna el valor del tercer lado ingresado por el usuario a una variable llamada "lado3". Este paso se realiza para almacenar la longitud del tercer lado del triángulo.

8. **semiperímetro = (lado1 + lado2 + lado3) / 2:** Se calcula el semiperímetro del triángulo sumando los tres lados y dividiendo el resultado por 2. El semiperímetro se utiliza en la fórmula de Herón para calcular el área del triángulo.

9. **área = sqrt(semiperímetro \* (semiperímetro - lado1) \* (semiperímetro - lado2) \* (semiperímetro - lado3)):** Se aplica la fórmula de Herón para calcular el área del triángulo utilizando el semiperímetro y las longitudes de los lados. La función sqrt representa la raíz cuadrada.

10. **Mostrar área:** Finalmente, se muestra el área calculada del triángulo.

11. **Fin:** El flujo del programa termina aquí.

* **Algoritmo para generar la serie Fibonacci:**

1. **Inicio:** El flujo comienza en este punto.

2. **Definir límite:** En esta etapa, se define el límite hasta el cual se generará la serie Fibonacci. Este límite determina cuántos términos de la serie se mostrarán.

3. **Definir término1 = 0:** Se inicializa el primer término de la serie Fibonacci como 0.

4. **Definir término2 = 1:** Se inicializa el segundo término de la serie Fibonacci como 1.

5. **Mostrar término1:** Se muestra en pantalla el valor del primer término de la serie Fibonacci.

6. **Mostrar término2:** Se muestra en pantalla el valor del segundo término de la serie Fibonacci.

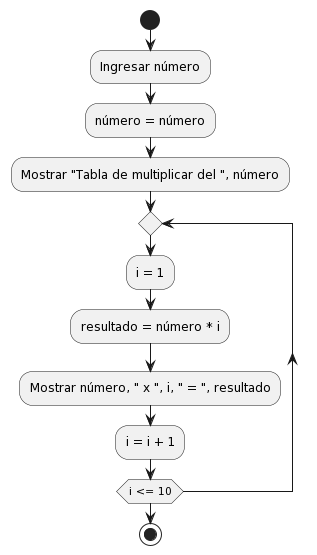
7. **Repetir mientras próximo término < límite:** Se inicia un bucle que se repetirá hasta que el valor del próximo término sea mayor o igual al límite establecido. Dentro del bucle:

7.1.**Calcular próximo término = término1 + término2:** Se calcula el siguiente término de la serie Fibonacci sumando los dos términos anteriores.

7.2.**Mostrar próximo término:** Se muestra en pantalla el valor del próximo término calculado.

7.3.**Actualizar términos:** Se actualizan los valores de los términos para preparar el siguiente cálculo. El valor de término1 se actualiza con el valor actual de término2, y el valor de término2 se actualiza con el valor calculado del próximo término.

8. **Detener:** El flujo del programa se detiene aquí después de que se haya mostrado la serie Fibonacci hasta el límite establecido.

* **Algoritmo para generar una tabla de multiplicar:**

1. **Inicio:** El flujo comienza en este punto.

2. **Ingresar número:** En esta etapa, el usuario ingresa el número para el cual desea generar la tabla de multiplicar.

3. **Número = número:** Se asigna el valor del número ingresado por el usuario a una variable llamada "número". Este paso se realiza para almacenar el número para el cual se generará la tabla de multiplicar.

4. **Mostrar "Tabla de multiplicar del ", número:** Se muestra un encabezado indicando para qué número se está generando la tabla de multiplicar.

5. **Repetir mientras i <= 10:** Se inicia un bucle que se repetirá hasta que el valor de i sea menor o igual a 10. Este bucle se utiliza para generar la tabla de multiplicar del número del 1 al 10. Dentro del bucle:

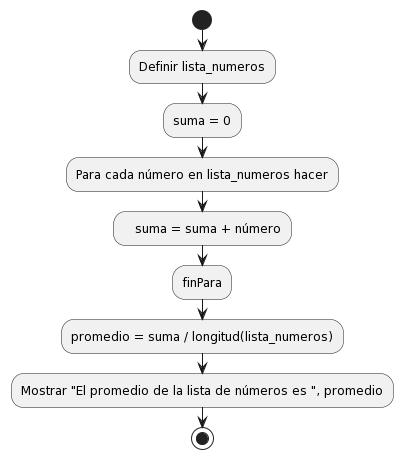
5.1.**i = 1:** Se inicializa una variable llamada "i" con el valor de 1, que se utilizará como el multiplicador en la tabla de multiplicar.

5.2.**Resultado = número \* i:** Se calcula el resultado de la multiplicación entre el número ingresado por el usuario y el multiplicador actual (i). Este resultado representa el producto de la tabla de multiplicar para el número actual.

5.3.**Mostrar número, " x ", i, " = ", resultado:** Se muestra en pantalla la multiplicación realizada en forma de expresión de tabla de multiplicar.

5.4.**i = i + 1:** Se incrementa el valor de i en 1 para pasar al siguiente multiplicador.

6. **Detener:** El flujo del programa termina aquí después de que se hayan generado todas las multiplicaciones del 1 al 10 para el número dado.

* **Algoritmo para calcular el promedio de una lista de números:**

1. **Inicio:** El flujo comienza en este punto.

2. **Definir lista\_numeros:** En esta etapa, se define la lista de números para la cual se calculará el promedio. Pueden ser números enteros o decimales, según sea necesario.

3. **suma = 0:** Se inicializa una variable llamada "suma" con el valor cero. Esta variable se utilizará para almacenar la suma de todos los números en la lista.

4. **Para cada número en lista\_numeros hacer:** Se inicia un bucle que recorre cada número en la lista de números. Dentro del bucle:

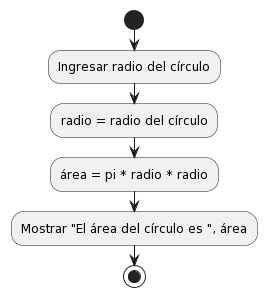
4.1.**suma = suma + número:** Se suma el valor del número actual a la variable "suma", lo que permite acumular la suma total de todos los números en la lista.

5. **finPara:** Se señala el final del bucle, lo que indica que se han sumado todos los números de la lista.

6. **promedio = suma / longitud(lista\_numeros):** Después de calcular la suma total de los números en la lista, se calcula el promedio dividiendo la suma por la cantidad total de números en la lista, que se obtiene utilizando la función "longitud(lista\_numeros)".

7. **Mostrar "El promedio de la lista de números es ", promedio:** Se muestra en pantalla el resultado del promedio calculado para la lista de números.

8. **Fin:** El flujo del programa termina aquí después de mostrar el resultado del promedio.

* **Algoritmo para calcular el área de un círculo: **

1. **Inicio:** El flujo comienza en este punto.

2. **Ingresar radio del círculo:** En esta etapa, el usuario ingresa el radio del círculo para el cual se calculará el área.

3. **radio = radio del círculo:** Se asigna el valor del radio del círculo ingresado por el usuario a una variable llamada "radio". Este paso se realiza para almacenar el radio del círculo para su posterior uso en el cálculo del área.

4. **área = pi \* radio \* radio:** Se calcula el área del círculo utilizando la fórmula matemática: área = π \* radio^2, donde π (pi) es una constante matemática aproximada a 3.14159. Se multiplica el valor del radio por sí mismo y por π para obtener el área.

5. **Mostrar "El área del círculo es ", área:** Después de calcular el área del círculo, se muestra un mensaje que indica el valor del área calculada.

6. **Fin:** El flujo del programa termina aquí después de mostrar el resultado del área del círculo.