

Ejercicio 16. a

INSTRUCTOR



Escribir el código flowgorithm XML para
Generar las secuencias de números capturando n
para n=1 generar 1
para n=2 generar 1, 3
para n=3 generar 1, 3, 5
para n=4 generar 1, 3, 4, 7
para n=m generar 1, 3, 5, 7 ... $m*2-1$
para $n \leq 0$ imprimir error



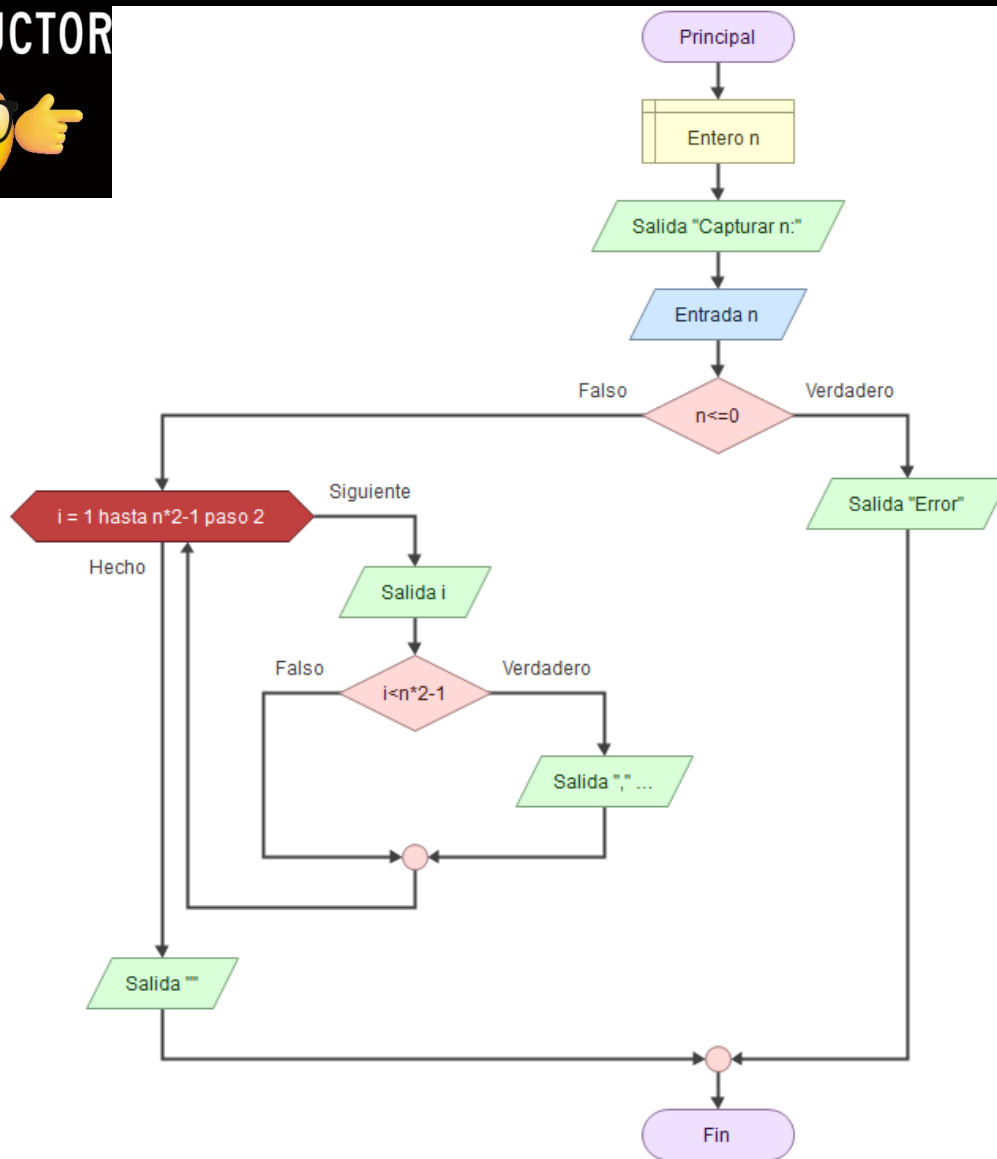
Aquí tienes el archivo XML completo para Flowgorithm que genera la secuencia de números impares desde 1 hasta $n*2-1$ (inclusive), y muestra "Error" si $n \leq 0$.

```
<?xml version="1.0"?>
<flowgorithm fileversion="3.0">
  <function name="Main" type="None">
    <parameter/>
    <body>
      <declare name="n" type="Integer" array="False"/>
      <output text="Capturar n: "/>
      <input variable="n"/>

      <if expression="n <= 0">
        <then>
          <output text="Error"/>
        </then>
        <else>
          <for variable="i" start="1" end="n * 2 - 1" step="2"/>
            <output expression="i" newline="False"/>
            <if expression="i <= n * 2 - 1">
              <then>
                <output text="," newline="False"/>
              </then>
            </if>
          </for>
          <output text=" " newline="True"/>
        </else>
      </if>
    </body>
  </function>
</flowgorithm>
```

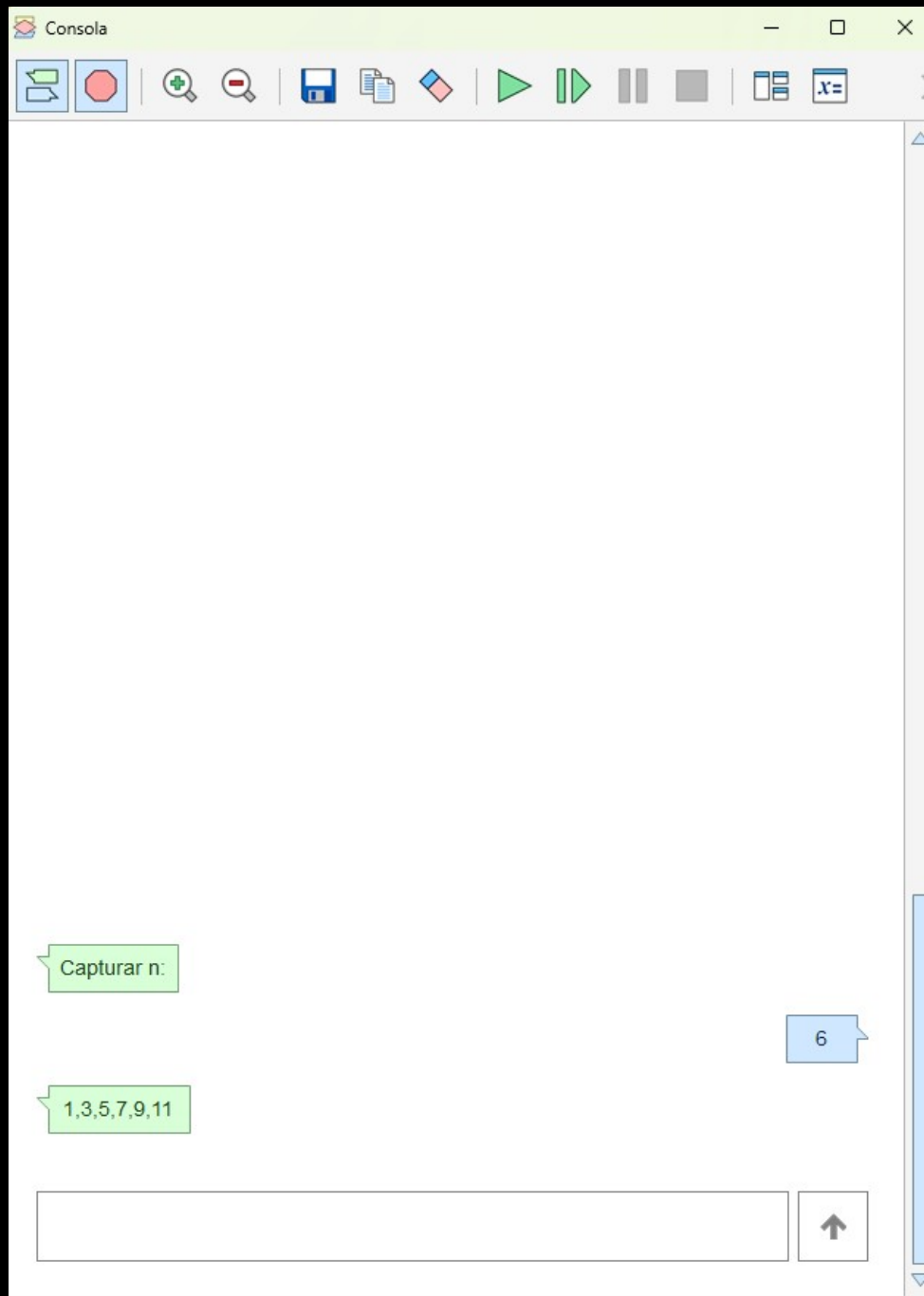
El diagrama de flujo del código anterior es:

INSTRUCTOR



Obviamente tiene un error. Modificar el diagrama de flujo anterior para que genere lo que se pide.

La salida debe ser (pero con diagrama actual no es) parecida a lo siguiente:



Ejercicio 16. b

INSTRUCTOR



i n=1	1,1
si n=2	2,1,1,4
si n=3	3,2,1,1,2,9
si n=4	4,3,2,1,1,2,3,16
si n=5	5,4,3,2,1,1,2,3,4,25

El último número generado es n^2



Entonces el patrón es:

1. Descendente: $n, n-1, \dots, 1$
 2. Ascendente: $1, 2, \dots, n$
 3. Último término = n^2
- Longitud total = $2n$.



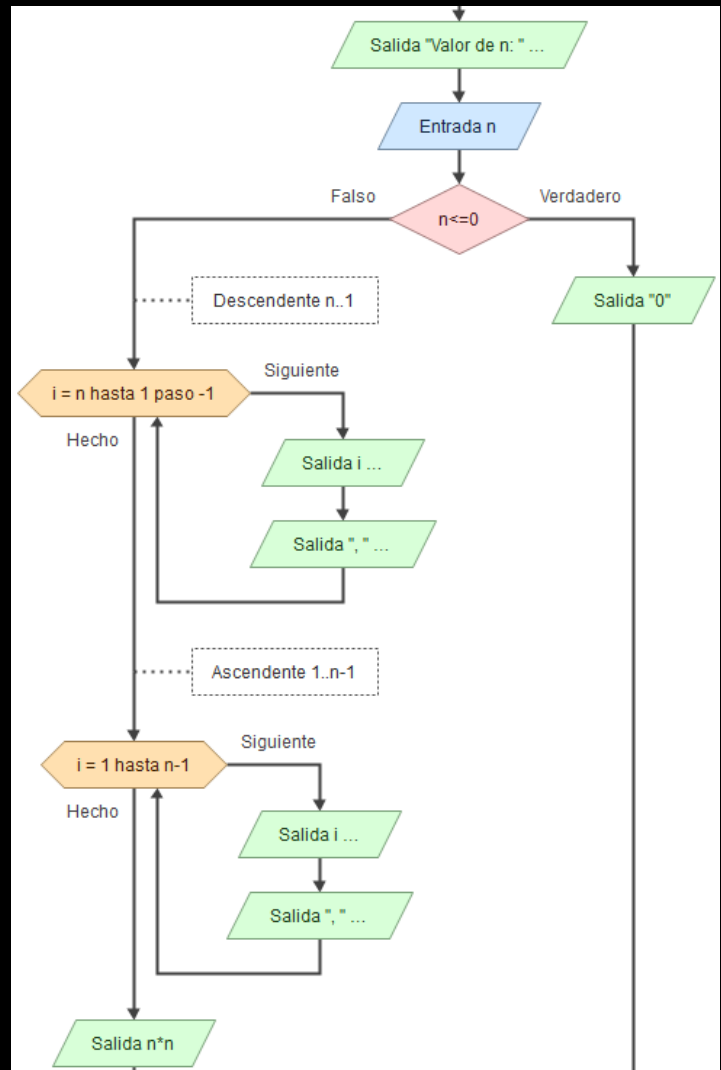
Aquí tienes el mismo algoritmo ya montado en Flowgorithm (archivo `.fprg` + imagen + pseudocódigo).

Solo importa el archivo o copia el pseudocódigo en un nuevo diagrama.

```
<flowgorithm fileversion="3.0">
  <function name="Main" type="None">
    <parameter/>
    <body>
      <declare name="n" type="Integer" array="False"/>
      <output text="Valor de n: "/>
      <input variable="n"/>
      <if expression="n <= 0">
        <then>
          <output text="0"/>
        </then>
        <else>
          <!-- Descendente n..1 -->
          <for variable="i" start="n" end="1" direction="down"/>
            <output expression="i" newline="False"/>
            <output text="," newline="False"/>
          </for>
          <!-- Ascendente 1..n-1 -->
          <for variable="i" start="1" end="n-1"/>
            <output expression="i" newline="False"/>
            <output text="," newline="False"/>
          </for>
          <!-- Último término n^2 -->
          <output expression="n * n" newline="True"/>
        </else>
      </if>
    </body>
  </function>
</flowgorithm>
```

Pero al construir el diagrama de flujo a partir del código anterior está mal.

INSTRUCTOR



Modificar el código anterior para que funcione de manera correcta

Ejercicio 16. c Solicitar a Kimi que construya el siguiente programa en flowgorithm:

INSTRUCTOR



Escribir código flowgorithm que realice lo siguiente:

si $n=1$ 1,1
si $n=2$ 2,1,1,4
si $n=3$ 3,2,1,1,2,9
si $n=4$ 4,3,2,1,1,2,3,16
si $n=5$ 5,4,3,2,1,1,2,3,4,25 El último número generado es n^2

Usar únicamente un lazo para generar todos los números

Igual que el anterior *pero únicamente mediante* un lazo y modificar el código generado por Kimi si es que contiene errores.

Ejemplo de corrida:

