

```

Declarar las variables P Q R como Booleano
Declarar las variables A B C como Real
Declarar la Variable Z como Cadena
Asignar a la variable P true
Asignar a la variable Q true
Asignar P&&Q a la variable R
imprimir el valor de R
Asignar a la variable P true
Asignar a la variable Q false
Asignar P&&Q a la variable R
Imprimir el valor de R
Asignar 5.5 a la variable A
Asignar 10.0 a la variable B
Realizar la siguiente asignacion: C=A*2+B
Imprimir el valor de C
Asignar Z="Prueba"
Imprimir el valor de Z

```



Aquí tienes el contenido completo del archivo `variables_demo.fprg` .

Guárdalo y ábrelo en Flowgorithm; ejecuta paso a paso para ver los valores en la ventana *Watch*.

```
<?xml version="1.0"?>
```

```

<flowgorithm fileversion="4.5">
  <algorithm name="variables_demo">
    <attributes>
      <attribute key="isMain" value="true"/>
    </attributes>
    <function name="Main" type="None">
      <parameters/>
      <body>
        <!-- Declaraciones -->
        <declare name="P" type="Boolean" array="False"/>
        <declare name="Q" type="Boolean" array="False"/>
        <declare name="R" type="Boolean" array="False"/>
        <declare name="A" type="Real" array="False"/>
        <declare name="B" type="Real" array="False"/>
        <declare name="C" type="Real" array="False"/>
        <declare name="Z" type="String" array="False"/>

        <!-- Prueba lógica AND -->
        <assign variable="P" expression="True"/>
        <assign variable="Q" expression="True"/>
        <assign variable="R" expression="P && Q"/>
        <output expression=""P && Q (True,True) = " & R"
newLine="True"/>
      </body>
    </function>
  </algorithm>
</flowgorithm>

```

```

    <assign variable="P" expression="True"/>
    <assign variable="Q" expression="False"/>
    <assign variable="R" expression="P && Q"/>
    <output expression="&quot;P && Q (True,False) = &quot; && R"
newline="True"/>

    <!-- Prueba aritmética real -->
    <assign variable="A" expression="5.5"/>
    <assign variable="B" expression="10.0"/>
    <assign variable="C" expression="A * 2 + B"/>
    <output expression="&quot;C = &quot; && C" newline="True"/>

    <!-- Prueba cadena -->
    <assign variable="Z" expression="&quot;Prueba&quot;"/>
    <output expression="&quot;Z = &quot; && Z" newline="True"/>
  </body>
</function>
</algorithm>
</flowgorithm>

```

Salida esperada en consola:

📋 复制

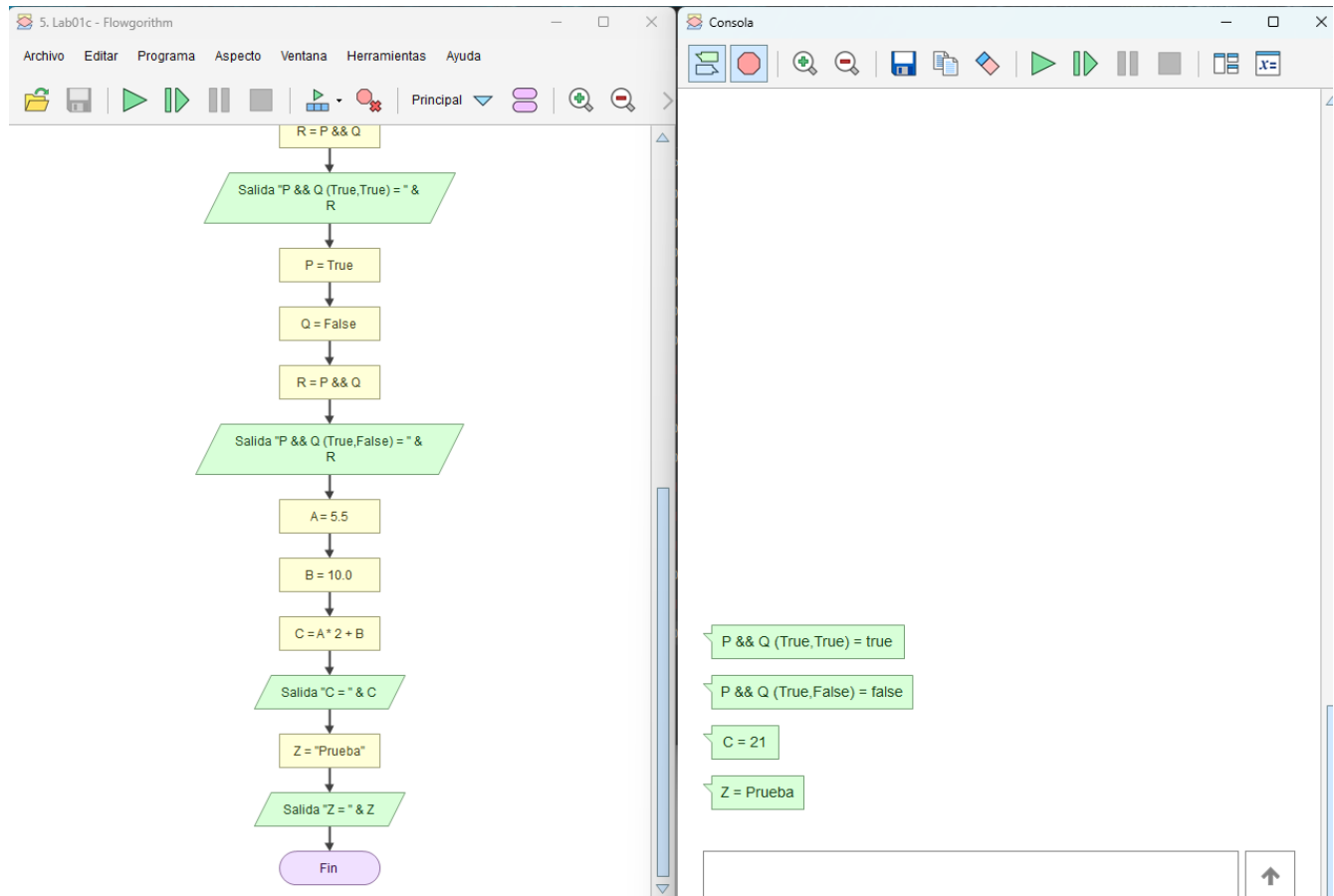
```

P && Q (True,True) = True
P && Q (True,False) = False
C = 21
Z = Prueba

```



Utilizando la misma forma que los dos programas anteriores



¿Sabes hacer simulación manual de la ejecución del programa anterior utilizando tablas?



¡Claro! A continuación tienes la **simulación manual** paso a paso del programa `variables_demo.fprg` usando una tabla de estados.

Cada fila representa una línea ejecutada y muestra el valor actual de **todas las variables** (solo cambian las que se modifican).

Simulación manual del programa

Paso	Línea ejecutada	P	Q	R	A	B	C	Z	Salida en consola
1	Declarar variables?	?	?	?	?	?	?	?	—
2	P = True	True	?	?	?	?	?	?	—
3	Q = True	True	True	?	?	?	?	?	—
4	R = P && Q	True	True	True	?	?	?	?	P && Q (True,True) = True
5	P = True	True	True	True	?	?	?	?	—
6	Q = False	True	False	True	?	?	?	?	—
7	R = P && Q	True	False	False	?	?	?	?	P && Q (True,False) = False
8	A = 5.5	True	False	False	5.5	?	?	?	—
9	B = 10.0	True	False	False	5.5	10.0	?	?	—
10	C = A * 2 + B	True	False	False	5.5	10.0	21.0	?	—
11	Z = "Prueba"	True	False	False	5.5	10.0	21.0	"Prueba"	—
12	output Z	—	—	—	—	—	—	—	Z = Prueba

Observaciones didácticas

- Las variables no inicializadas aparecen como `?` (Flowgorithm las pone en 0, 0.0, False, "", pero aquí usamos `?` para destacar el cambio).
- Los valores en **negrita** indican la asignación que ocurre en ese paso.
- La columna *Salida* muestra exactamente lo que aparece en la consola de Flowgorithm.

Puedes entregar esta tabla a los alumnos para que **completen la simulación a mano** y comprueben paso a paso.