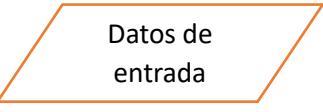
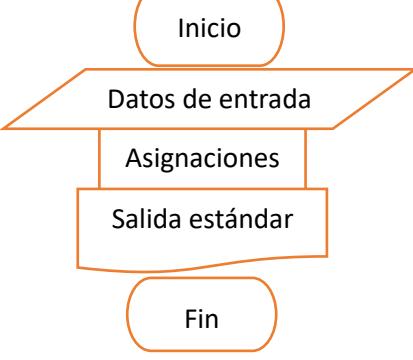
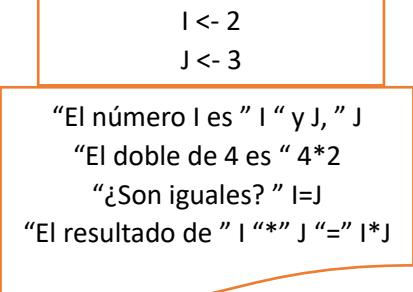
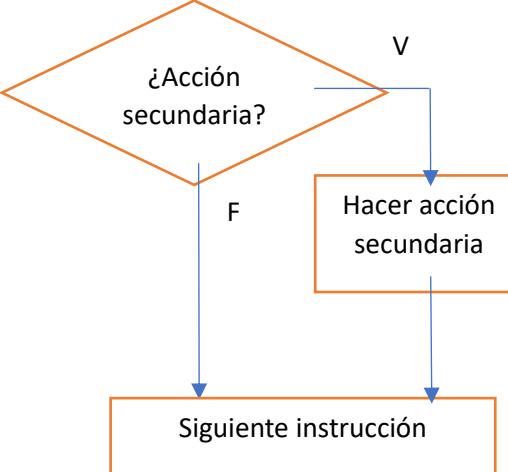


Diagramas de flujo

 	 <p>Se escriben separados por comas los datos que se desean recibir por entrada estándar.</p>
<p>1. TODO diagrama de flujo tiene como mínimo principio y fin, y, a excepción del algoritmo que no hace nada, un bloque de asignación, en el que van una o más asignaciones.</p>	<p>2. Todos los datos se leen desde memoria interna a no ser que se indique explícitamente. P.ej. "entrada estándar", "se lee por el teclado", "el usuario introduce"</p>
	
<p>3. La regla anterior aplica también para los datos de salida. A no ser que se indique "salida estándar", "se imprime en pantalla", "se muestra", estos se escriben en memoria</p>	<p>4. Así, todo diagrama de flujo que requiera de interacción con el usuario, tiene como mínimo principio y final y una serie de bloques de asignación y de entrada o salida de datos que pueden intercalarse a lo largo del flujo del algoritmo</p>
 <p><u>Resultado:</u> El número I es 2 y J, 3 El doble de 4 es 8 ¿Son iguales? false El resultado de 2*3=6</p>	

Furro Computero
TRUCOS DEL ALMENDRUZO

<p>5. Si se quieren imprimir expresiones literales, estas deben indicarse entre comillas para no confundirlas con variables, números u operandos. De manera extraordinaria, puede no cumplirse esta regla si no hay ambigüedad. Recuerde también no confundir asignación ($<-$) con comparación igual ($=$)</p>	<p>6. Si en el problema se pide ejecutar una acción determinada solo si se cumplen requisitos, en nuestro diagrama debe haber un bloque de decisión que desvíe el flujo hacia dicha acción. Independientemente de si esas asignaciones se han procedido o no, pasamos a la siguiente instrucción</p>
<pre> graph TD V((V)) --> A1[Acción A] F((F)) --> A2[Acción B] A1 --- D1{¿Se ejecuta A?} A2 --- D1 </pre>	<pre> graph TD D2{Variables de condición} --> A3[A] D2 --> B3[B] D2 --> C3[C] D2 --> D3[D] </pre>
<p>7. Si en el problema se debe decidir entre dos acciones distintas, se usa el bloque de decisión, y en dicho bloque, debe haber una condición que devuelva verdadero o falso que nos permita decidir si se ejecuta la acción A o la acción B</p>	<p>8. Si hay más de dos acciones a decidir las cuales depende cada una de un requisito, se usa un bloque de condición. Ese bloque contiene a las variables separadas por comas con las que se estipulan los requisitos, y cada flujo, viene condicionado por un rango de valores de dichas variables</p>
<pre> graph TD A4[Inicialización de variables] --> A5[Instrucciones a ejecutar en cada iteración] A5 --> D3{¿Condición de salida?} D3 --> V4[V: Salir del bucle] D3 --> F4((F)) F4 --> A4 </pre>	<p>9. Los problemas donde se requiera “buscar”, “recorrer”, “contar” y/o “acumular” suelen ser tratados con un bucle, ya que para cada elemento, se repiten las mismas instrucciones. Un bucle queda definido por una inicialización de variables, una serie de instrucciones a ejecutar en cada iteración y una condición de salida del bucle.</p> <p>Hay cinco tipos principales de bucle dependiendo de la disposición de los bloques dichos y las condiciones de salida.</p> <p>Practicaremos bastantes problemas de este tipo a lo largo del curso, ya que son los más comunes en el mundo de la programación.</p>