

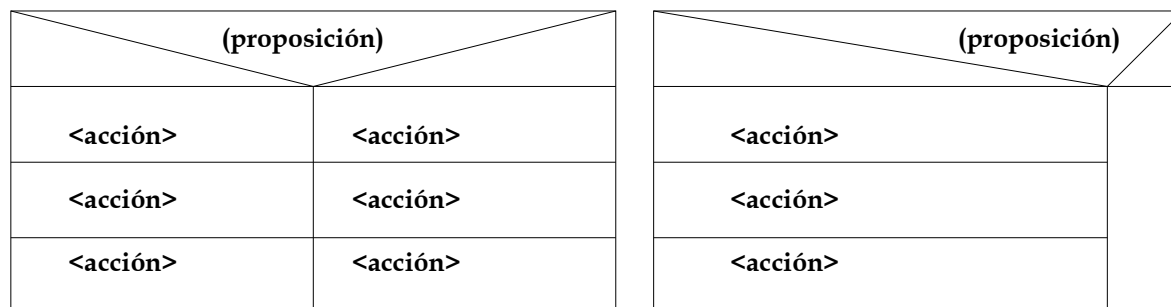
Algoritmos Condicionales

Es habitual que, llegado a un punto de la resolución de un problema, existan diferentes acciones que pueden llegar a ejecutarse.

Pero la elección de las acciones a ejecutar no se realiza al azar, sino que existe una **condición** según la cual ejecutaremos unas u otras acciones..

Esa **condición** se expresa como una **proposición**, cuyo valor de verdad determina el camino a seguir.

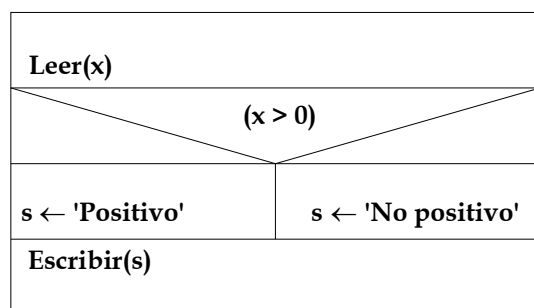
Es aquí donde el **PSD** cobra especial importancia como herramienta de diseño de algoritmos.



El **PSD** de la izquierda representa una estructura condicional **binaria**. Las acciones que aparecen a la izquierda se ejecutarán si el valor de verdad de la proposición es verdadero, y las situadas a la derecha, si es falso.

El **PSD** de la derecha representa una estructura condicional **unaria**. En ella, solo se ejecutarán acciones si el valor de verdad de la proposición es verdadero.

Puede suceder que para uno o ambos valores de verdad de la proposición se deba ejecutar una sola acción:



```

Comienzo
Leer(x);
Si (x > 0)
    entonces s ← 'Positivo'
    si no      s ← 'No positivo';
Escribir(s);
Fin.
    
```

Las acciones que se ejecutan condicionalmente, en este caso las dos asignaciones, y que ocupan la mitad del ancho del PSD, se denominan **acciones de segundo nivel**. Las que se ejecutan siempre, son **acciones de primer nivel**.

En el **pseudocódigo**, las acciones de segundo nivel se escriben dejando una sangría mayor, como una forma de mostrar la estructura.

Una visión liviana del algoritmo, podría sugerir la eliminación de las acciones de asignación y reemplazarlas por **Escribir('Positivo')** y **Escribir('No positivo')**, con lo cual se eliminaría la última acción y la necesidad de utilizar la variable **s**.

Pero eso alteraría la forma que debe tener un algoritmo para ser fácilmente entendible, con vistas a su posible modificación o corrección.

Un algoritmo debe tener la estructura formal:



Esa estructura se muestra claramente en el algoritmo que hemos escrito.

```
Comienzo
Leer(x);
Si (x > 0)
    entonces s ← 'Positivo'
    si no      s ← 'No positivo';
Escribir(s);
Fin.
```