

Parámetros (Teoría)

En programación, un **procedimiento** es un conjunto de instrucciones que realizan una acción definida. Cuando ese procedimiento necesita trabajar con datos variables, se le pueden pasar parámetros. Un **parámetro** es un valor que no está escrito de forma fija en la definición del procedimiento, sino que se especifica al momento de llamarlo.

La **definición** del procedimiento indica qué pasos seguirá y qué parámetros recibirá. En cambio, la **llamada** ocurre cuando se ejecuta el procedimiento, indicando los valores concretos para esos parámetros. A esos valores concretos se los llama **argumentos**. De este modo, la lógica se escribe una sola vez y se puede reutilizar en diferentes situaciones, cambiando solo los datos que se le entregan.

Ejemplo 1: Videojuegos – Carrera con pista variable

En un juego de carreras, la acción “*correr una carrera*” siempre sigue la misma secuencia: largar, avanzar, doblar en las curvas y llegar a la meta. Esa parte no cambia. Lo que sí puede cambiar es la **pista**: puede ser más corta o más larga, con muchas curvas o casi recta. Ese dato —la pista elegida— es el **parámetro**.

En la programación del juego, la acción “correr una carrera” se escribe una sola vez, dejando un lugar vacío donde irá el dato de la pista. Cuando el jugador elige una pista, el programa coloca ese valor en el lugar correspondiente y ejecuta la acción.

Así, con una sola acción podemos correr en cualquier pista. Lo único que cambia es el dato que le damos, y el programa adapta el recorrido usando ese valor.

Ejemplo 2: Reproducir música

Escuchar música en el celular siempre implica lo mismo: tocar “*play*” y dejar que suene.

Lo que cambia es **qué canción elegimos**: puede ser de rock, cumbia, rap o cualquier otro género. Ese nombre de la canción que seleccionamos es el **parámetro**.

Si no existieran los parámetros, necesitaríamos un botón distinto para cada canción de nuestra lista. Con parámetros, un solo botón sirve para todas: lo único que cambia es el dato que le pasamos, en este caso, la canción elegida.

En Pilas Bloques

En Pilas Bloques, algunos bloques incluyen un espacio para escribir un valor antes de que se ejecuten. Por ejemplo: el **largo del lado** que se va a dibujar, los **grados** que se va a girar o la **distancia** que se va a saltar.

La acción que realiza el bloque siempre es la misma, pero el resultado cambia según el valor que escribamos en ese espacio. Esto permite usar el mismo bloque en muchas situaciones distintas, simplemente cambiando ese dato.

También es posible **crear procedimientos propios con parámetros**. Esto significa que podemos diseñar una secuencia de acciones que reciba un valor y lo utilice dentro de sus instrucciones. De esta forma, en lugar de escribir varias versiones casi idénticas de un procedimiento, podemos tener uno solo que funcione con diferentes valores.

¿Para qué sirven los parámetros?

1. Usar una sola acción para muchos casos

Con parámetros, se puede **reutilizar** una misma acción en distintas situaciones. En lugar de escribirla de nuevo para cada caso, se cambia el **valor del parámetro** en cada **llamada**, adaptando la ejecución a los datos recibidos.

2. Controlar el comportamiento con datos externos

Un **parámetro** es una variable definida en la acción que, al ejecutarse, recibe un valor concreto llamado **argumento**. La definición fija las instrucciones, pero el argumento **determina** datos específicos para esa ejecución, permitiendo que un mismo procedimiento se ajuste a distintos casos.

3. Reducir errores y ahorrar tiempo

Al centralizar la lógica en **una sola acción con parámetros**, cualquier cambio se refleja en todas sus ejecuciones. Esto evita duplicar código, disminuye posibles errores y favorece la **reutilización**, clave para mantener el programa ordenado y eficiente.